

*République Algérienne Démocratique et Populaire*  
*Ministère de l'Enseignement Supérieure et de la Recherche*  
*Scientifique*

*Université A. MIRA Bejaia*  
*Faculté de Technologie*  
*Département de Génie Electrique*

# Mémoire de fin d'études

En vue de l'obtention du diplôme master 2  
En automatisme industriel

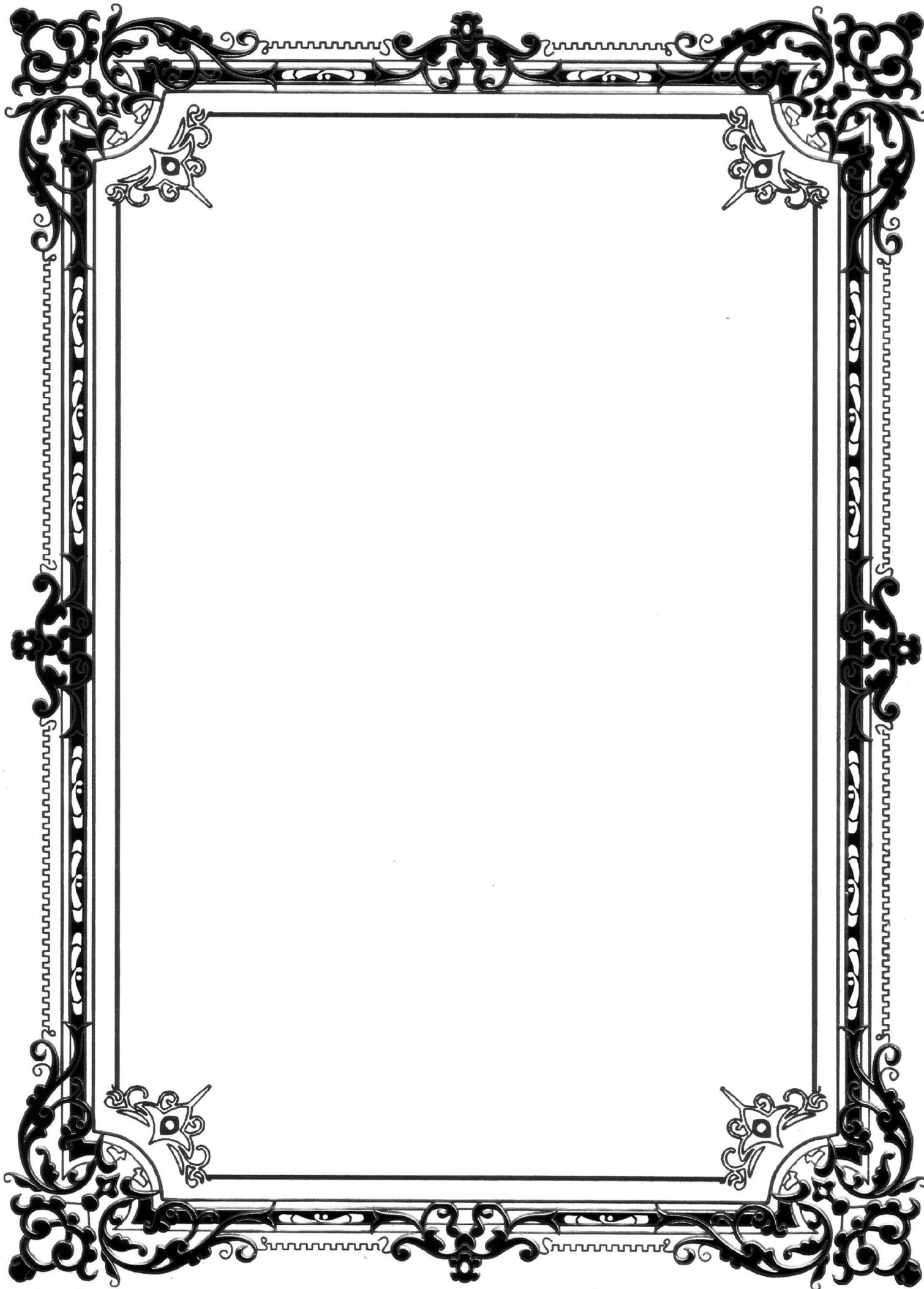
## Thème

**Conception et Réalisation**  
**d'une Carte d'Acquisition**  
**de Donnée via le Protocole TCP/IP**

**Encadré par :**  
Mr. METIDJI B.  
Mr. TAIB N.

**Présenté par :**  
DERGUINI NASSIM  
SAHLI FARID

Promotion 2010/2011



# *Remerciements*

*On tient à remercier vivement nos deux promoteurs Mr : METTIDJI ainsi Mr : TAIB pour leurs dévouements et leurs aides précieuses qu'ils nous ont apporté .on tient également à remercier les professeurs qui ont contribué de près ou de loin à notre formation.*

# *Dédicaces*

*Je dédie ce modeste travail :*

*À mes chers parents*

*À mes frères et mes sœurs*

*À tous mes amis (es) et toutes les personnes qui m'ont encouragé tout au long de mes études*

*À tous les êtres chers dont le soutien m'a été indispensable*

*Nassim*

*Je dédie ce modeste travail :*

*À mes chers parents SURTOUT MA MAMAN*

*À mes frères et mes sœurs MAMOU ET JIJA EN EXCIPTION*

*À tous mes amis (es) et toutes les personnes qui m'ont encouragé tout au long de mes études surtout Ahmed*

*À tous les êtres chers dont le soutien m'a été indispensable.*

*Farid*

---

|                                    |   |
|------------------------------------|---|
| <b>Introduction générale</b> ..... | 1 |
|------------------------------------|---|

## **Chapitre I : Acquisition de données**

|  |    |
|--|----|
| I.1 Acquisition .....                                      | 2  |
| I.2 Structure d'une chaîne d'acquisition .....             | 2  |
| I.2.1 Capteurs .....                                       | 3  |
| I.2.2 Conditionnement de signaux .....                     | 4  |
| I.2.3 Les échantillonneurs.....                            | 5  |
| I.2.4 Convertisseurs analogique/numérique (A/D) .....      | 6  |
| I.2.4.1 Quelques convertisseurs analogique/numérique ..... | 6  |
| I.3 Généralités sur la transmission .....                  | 8  |
| I.3.1 Représentation des données .....                     | 8  |
| I.3.2 Transmission de données .....                        | 8  |
| I.3.3 Codage des signaux de transmission .....             | 8  |
| I.3.4 Codage en bande de base.....                         | 8  |
| I.3.4.1 Codage On-Off ou NRZ (non retour à zéro).....      | 9  |
| I.3.4.2 Codage RZ (retour à zéro).....                     | 9  |
| I.3.4.3 Codage Manchester.....                             | 10 |
| I.3.4.4 Codage Manchester différentiel .....               | 10 |
| I.3.5 Support de transmission des données .....            | 11 |
| I.3.6 Transmission série et parallèle .....                | 13 |
| I.3.7 Transmission synchrone et asynchrone .....           | 14 |
| I.4. Conclusion .....                                      | 12 |

## **Chapitre II : protocole de communication**

|  |    |
|--|----|
| II.1. Protocoles de communication .....                          | 14 |
| II.2. Modèle de référence OSI (Open System Interconnexion) ..... | 14 |
| II.2.1 Origine du modèle .....                                   | 14 |

---

|        |                                       |    |
|--------|---------------------------------------|----|
| II.2.2 | Système ouvert .....                  | 15 |
| II.2.3 | Couches du modèle OSI .....           | 15 |
| II.3.  | Model de TCP/IP .....                 | 17 |
| II.4.  | La technologie Ethernet .....         | 18 |
| II.4.1 | Propriétés d'un réseau Ethernet ..... | 18 |
| II.4.2 | Variantes de l'Ethernet .....         | 19 |
| II.4.3 | L'adressage Ethernet .....            | 20 |
| II.4.4 | Structure des trames Ethernet .....   | 20 |
| II-5.  | Conclusion .....                      | 21 |

## **Chapitre III : Microcontrôleur ATMEGA28J60 & ENC28J60E**

|         |  |    |
|---------|--|----|
| III.1.  | Description interne d'un microcontrôleur .....       | 23 |
| III.2.  | Les différentes architectures des microprocesseurs c |    |
| III.2.1 | L'architecture Von-Neumann .....                     | 24 |
| III.2.2 | L'architecture Harvard .....                         | 24 |
| III.3.  | Présentation Physique .....                          | 26 |
| III.4.  | Synoptique .....                                     | 27 |
| III .5  | Descriptions des broches .....                       | 28 |
| III.6   | CPU du corps AVR .....                               | 29 |
| III.7   | Le Plan Mémoire .....                                | 30 |
| III.7.1 | La mémoire programme(FLASH) .....                    | 30 |
| III.7.2 | La mémoire de donnée(SRAM) .....                     | 30 |
| III.7.3 | La mémoire morte(EEPROM) .....                       | 30 |
| III.8.  | L'Horloge du Système et Option .....                 | 30 |
| III.9.  | Les interruptions .....                              | 31 |
| III.10. | Les Entrées/Sorties (PORTx) .....                    | 32 |

---

|  |    |
|--|----|
| III.11. LE TIMER/COMPTEUR(TIMER) .....                             | 32 |
| III.11.1 Timer 0 .....   | 33 |
| III.11.2 TIMER1 .....  | 34 |
| III.11.3 Timer2 .....  | 35 |
| III.12. Le watchdog .....  | 36 |
| III.13 L'interface Série Synchrone SPI .....                       | 36 |
| III.14. L'interface Série USART .....                              | 38 |
| III-15. L'interface I2C (TWI) .....                                | 38 |
| III-16. Le comparateur analogique .....                            | 38 |
| III.17. Le convertisseur analogique numérique ADC .....            | 40 |
| III.17.1 Synoptique de l'ADC .....                                 | 41 |
| III.18. Présentation général de contrôleur Ethernet ENC28J60 ..... | 41 |
| III.18.1 Présentation physique .....                               | 42 |
| III.18.2 CONNEXIONS EXTERNES .....                                 | 43 |
| III.18.2.1 Oscillateur .....                                       | 44 |
| III.18.2.2 Oscillateur Start-up Timer .....                        | 44 |
| III.18.2.3 Horloge externe (CLKOUT) .....                          | 44 |
| III.18.2.4 LED .....   | 44 |
| III.19. Mémoire .....  | 45 |
| III.20. Interface SPI .....  | 45 |
| III.20.1 Fonctionnement .....                                      | 46 |
| III.21 Conclusion .....  | 46 |
| <b>Chapitre IV : Conception et réalisation</b>                     |    |
| IV.1. Représentation synoptique .....                              | 47 |
| IV-2. Description des différents blocs .....                       | 48 |

|  |    |
|--|----|
| IV-2.1 L'alimentation .....  | 48 |
| IV.2. 2 Le microcontrôleur ATMEGA32 .....                                | 49 |
| IV.2.3 Le microcontrôleur ENC28J60 .....                                 | 49 |
| IV.2.3 Capteurs LM35 .....   | 56 |
| I V. 2.3.1 Choi du capteur .....   | 51 |
| IV. 2.3.2 Propriétés du LM35 .....                                       | 51 |
| IV.2.3.3 Conversion et contrôle de l'ADC .....                           | 53 |
| IV.2.3.4 Paramétrage du convertisseur simple .....                       | 55 |
| IV.3. Les outils de programmation et simulation du microcontrôleur ..... | 55 |
| IV.3.1 L'environnement de développement AVR studio .....                 | 55 |
| IV.3.2 Création d'un projet .....  | 57 |
| I V.3.3 Transfert du programme vers le microcontrôleur.....              | 60 |
| IV.4. Simulation .....   | 61 |
| IV.4.1 Présentation de Proteus ISIS7.....                                | 62 |
| IV4.2 Création d'un projet .....   | 62 |
| IV.5. Protocole de transmission .....                                    | 63 |
| IV.5.1 Configuration d'un poste client .....                             | 63 |
| IV.6 Organigramme fonctionnel .....                                      | 65 |
| IV.6.1 Appel a la lecture des données du capteur .....                   | 66 |
| IV.6.2 Transfert et Contrôle de l'arriver l'information .....            | 67 |
| IV.7 Schéma électrique globale .....                                     | 68 |
| IV.8 Circuit imprimé .....   | 69 |
| IV.9 Conclusion .....  | 70 |

**Conclusion générale**

**Bibliographie**

**Annexe**

**Introduction**

**générale**

La plus grande difficulté de mise en œuvre d'une chaîne d'acquisition de données est de réaliser l'adéquation la plus parfaite possible entre les caractéristiques des données du procédé externe et les éléments de la chaîne d'acquisition, cette adéquation doit être réalisée selon différents critères :

- Obtenir des données numériques les plus proches possible des données réelles.
- Minimiser le coût de cette réalisation.
- Posséder un matériel standard pour répondre à l'évolutivité et à la réutilisation.

En outre la démarche du choix d'une chaîne d'acquisition de données doit suivre une méthodologie aussi précise que possible ayant comme objectifs la détermination des paramètres importants de son application et le choix parmi les matériels existants correspondants à cette application.

Les ordinateurs sont actuellement des plates formes privilégiées pour les applications d'acquisition, de traitement et de restitution des données. Ils offrent en effet une très grande variété d'outils logiciels pour le développement des programmes d'acquisition.

Dans ce travail, nous allons présenter l'étude et la réalisation d'une carte d'acquisition de données via le protocole TCP/IP. Nous avons pris un capteur de température LM35 est utilisé comme exemple.

A cet effet notre document est subdivisé en quatre chapitres. Dans le premier, on a abordé des généralités sur la transmission de données ainsi que l'acquisition.

Le deuxième chapitre est consacré à la présentation des fondements d'un protocole de communication.

Puis, au troisième chapitre, est consacré à la description détaillée du microcontrôleur ATMEGA32 et du contrôleur Ethernet ENC28J60 sera présentée.

Le dernier chapitre concernera la présentation de la partie réalisation de notre carte d'acquisition.

Enfin on termine par une conclusion générale.

# Chapitre I

## I.1 Acquisition [1]

La nature qui nous environne est perceptible par nos sens. Mais l'archivage de ces perceptions (la mémoire) et son traitement (la pensée) ne sont pas des moyens adaptés respectivement au *partage* de ces informations, non plus qu'à leur traitement *massif*. Pour ce faire, nous utilisons des outils numériques. Mais ces outils, par leur nature même, requièrent une interface avec le monde réel, qui est lui fondamentalement analogique, pour autant que nous le sachions. La réalisation de cette interface est un problème complexe, dont nous n'allons aborder.

On peut distinguer plusieurs composants nécessaires à cette réalisation :

- capteur.
- échantillonneur.
- convertisseur analogique/numérique.
- système numérique de traitement.
- convertisseur numérique/analogique éventuellement

## I.2 Structure d'une chaîne d'acquisition

Une chaîne d'acquisition suit la synoptique suivante :

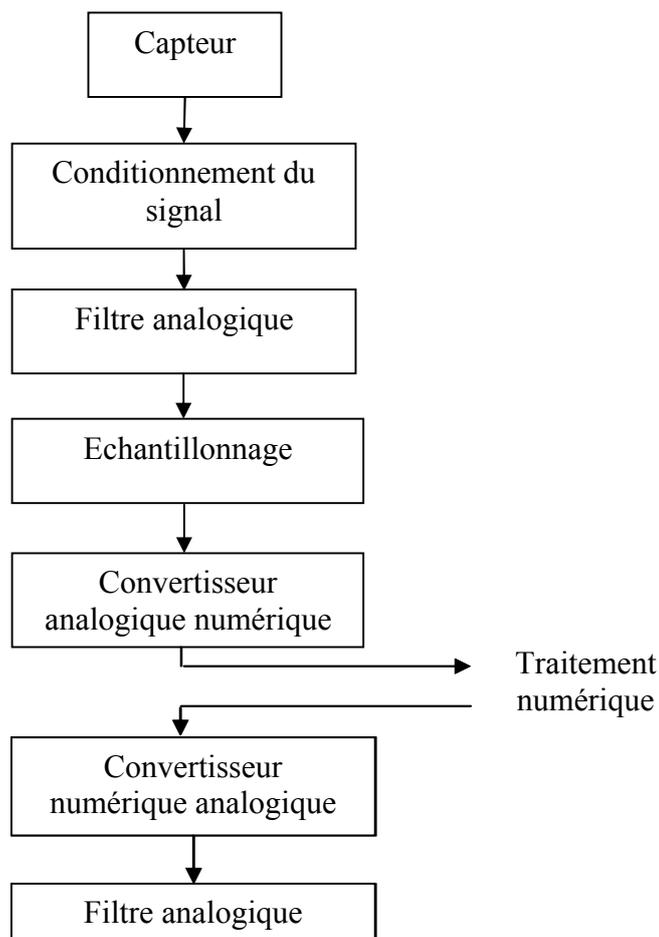


Fig.1 chaîne d'acquisition

Le capteur est le premier maillon de la chaîne. Il conditionne toutes les performances en ce sens que le reste de la chaîne de mesure ne pourra fournir plus d'information que ce que lui-même permet.

### I.2.1 Capteurs

Un **capteur** est un dispositif transformant l'état d'une grandeur physique observée en une grandeur utilisable exemple : une tension électrique, une hauteur de mercure, une intensité, la déviation d'une aiguille.

Le capteur se distingue de l'instrument de mesure par le fait qu'il ne s'agit que d'une simple interface entre un processus physique et une information manipulable. Par opposition, l'instrument de mesure est un appareil autonome se suffisant à lui-même, disposant d'un affichage ou d'un système de stockage des données. Le capteur, lui, en est dépourvu.

Les capteurs sont les éléments de base des systèmes d'acquisition de données. Leur mise en œuvre est du domaine de l'instrumentation.

Les capteurs et leurs conditionneurs peuvent aussi faire l'objet d'une classification par type de sortie.

La sortie est une grandeur électrique dont la valeur est proportionnelle à la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peuvent être du type :

- ✓ sortie tension
- ✓ sortie courant
  
- On distingue plusieurs modes de classifications
  - ✓ Capteurs passifs
  - ✓ Capteurs actifs (ou capteurs directs)
  
  - ✓ Capteurs numériques
  
  - ✓ Capteurs logiques

## I.2.2 Conditionnement de signaux

Le conditionnement de signaux transforme votre matériel d'acquisition de données en un système complet en facilitant la connexion directe à un large éventail de capteurs et de types de signaux, depuis les thermocouples jusqu'aux signaux haute tension. Des technologies majeures de conditionnement de signaux vont jusqu'à décupler les performances globales et la précision du système d'acquisition de données.

De plus, On a vu qu'un signal électrique pouvait être traité de manière analogique et/ou numérique. Cette relative facilité de manipulation peut être exploitée quand il s'agit de transmettre et d'analyser des informations. Mais le nombre de celles-ci et surtout la grande diversité de leurs supports (ondes radio, ondes lumineuses, courants électriques, etc.) impliquent qu'il soit procédé à leur prétraitement, un codage, avant leur émission, et donc à un décodage à leur réception.

Produits de conditionnement de signaux de National Instruments offrent sept technologies majeures qui simplifient la connectivité au capteur, éliminent le bruit, assurent la sécurité et améliorent la précision.

- **Amplification**

Les amplificateurs élèvent le niveau du signal d'entrée pour mieux correspondre à la gamme du convertisseur analogique/numérique (ADC), augmentant ainsi la résolution et la sensibilité de la mesure. De plus, l'utilisation de conditionneurs de signaux externes, situés plus près de la source du signal, ou du capteur, améliore le rapport signal/bruit de la mesure en élevant le niveau du signal avant qu'il puisse être affecté par le bruit ambiant.

- **Isolation**

Les matériels de conditionnement de signaux isolés transmettent le signal de sa source vers le matériel de mesure sans connexion physique grâce à des techniques de couplage par transformateur, optiques ou capacitives. En plus d'interrompre les boucles de terre, l'isolation bloque les transitoires hautes tensions et rejette les tensions élevées en mode commun. Elle protège ainsi les opérateurs et les équipements de mesure onéreux.

- **Multiplexage**

En multiplexant, le système de mesure peut router plusieurs signaux de manière séquentielle vers un seul numériseur, offrant ainsi une manière économique d'augmenter fortement le nombre de voies du système. Le multiplexage est nécessaire pour toute application à nombre élevé de voies.

- **Filtrage**

Les filtres rejettent le bruit indésirable dans une certaine gamme de fréquences.

Presque toutes les applications d'acquisition de données sont soumises à un certain niveau de bruit à 50 ou 60 Hz provenant du secteur ou du matériel. La plupart des conditionneurs de signaux incluent des filtres passe-bas spécifiquement conçus pour offrir une réjection maximum du bruit de 50 à 60 Hz.

- **Excitation**

L'excitation est nécessaire pour certains capteurs. Par exemple, les jauges de contrainte, les thermistances et les RTD nécessitent des signaux d'excitation externes de tension ou de courant. Les mesures par RTD et par thermistance sont habituellement effectuées à l'aide d'une source de courant qui convertit la variation de la résistance en tension mesurable. Les jauges de contrainte, qui sont des matériels très basse résistance, sont typiquement utilisées dans une configuration en pont Wheatstone avec une source d'excitation en tension.

- **Compensation de soudure froide**

La compensation de soudure froide est une technologie nécessaire pour des mesures précises par thermocouple. Lorsqu'un thermocouple est connecté à un système d'acquisition de données, il faut connaître la température au point de connexion (cette jonction représente un autre "thermocouple" dans le circuit de mesure et introduit typiquement un décalage de la mesure) pour calculer la température réelle mesurée par le thermocouple.

### 1.2.3 Les échantillonneurs

L'échantillonneur est un composant essentiel destiné à relier le (monde analogique) au (monde numérique). Il prend en entrée un signal analogique, et est branché en sortie sur un système numérique de traitement de données via un convertisseur analogique/numérique. On distingue deux types de circuits échantillonneurs :

-Echantillonneur simple.

-Echantillonneur bloqueur.

## I.2.4 Convertisseurs analogique/numérique (A/D)

Un convertisseur analogique/numérique (A/D), comme son nom l'indique, convertit un signal analogique en un signal numérique. Il est à noter qu'on perd de la précision dans l'opération : alors qu'en analogique on peut espérer avoir une précision infinie, en numérique on est limité par le *pas de l'échantillonnage*. En règle générale, on aura à établir un compromis entre la précision désirée et la rapidité de conversion.

- Un A/D aura une sortie série ou parallèle. Il sera nécessairement composé :
  - ✓ d'une horloge, car les bits à la sortie du composant auront une durée déterminée.
  - ✓ de signaux de contrôle (début de conversion, fin de conversion, reset...) pour lui permettre de dialoguer avec le système numérique situé en aval.
  - ✓ d'un système ((producteur de bits)) : registre, compteur ou simple circuit de logique combinatoire.
- De plus un convertisseur AD est caractérisé par :
  - ✓ sa résolution, définie comme étant la plus petite variation de l'entrée entraînant une variation d'un bit en sortie.
  - ✓ son temps de conversion.

### I.2.1 Quelques convertisseurs analogique/numérique

#### - Convertisseurs à rampe.

A la valeur de la tension d'entrée on fait correspondre une impulsion dont la largeur est proportionnelle à cette tension. Cette impulsion vient contrôler l'autorisation à s'incrémenter d'un compteur. On génère ainsi le code binaire de sortie en comptant plus ou moins longtemps en fonction de l'amplitude du signal à convertir.

Il est Simple et peu coûteux. Mais lent car nécessite 2 puissance N cycles d'horloges pour effectuer une conversion. Comme il n'y a pas de synchronisme entre l'horloge et le RAZ (remise a zéro), cela induit une imprécision de 1 période au début et à la fin de la conversion soit une erreur moyenne de 1,5 quantum.

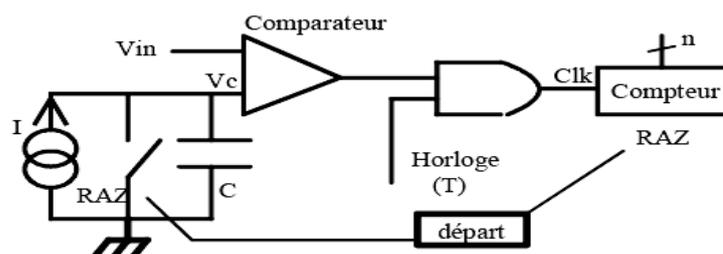


Fig.3 Convertisseurs rampe.

### - Convertisseur à approximations successives.

On détermine les valeurs des différents bits l'un après l'autre en commençant par le bit le moins significative. Le signal est comparé à une tension de référence  $V_0/2$ . S'il est supérieur, on lui retranche cette valeur et on met le bit de comparaison à '1', sinon on met le bit de comparaison à '0' et on le compare à la tension suivante.

On effectue ainsi un encadrement progressif de plus en plus fin. Ainsi pour un CAN N bits, en N coups on obtient la conversion. Il est lent, ainsi pour 16 bits, il lui faut en moyenne un temps de conversion de 10  $\mu$ s.

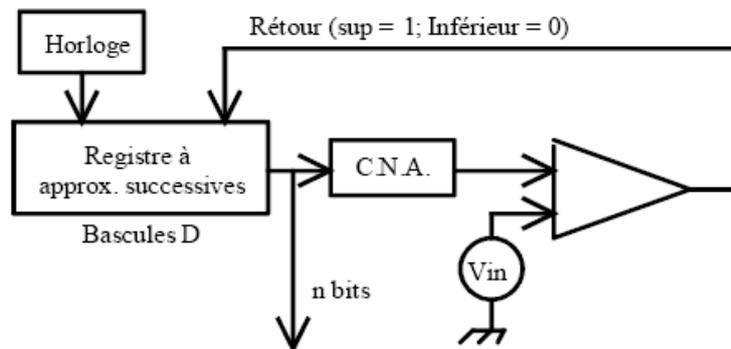


Fig.4 Convertisseur à approximations successives.

Lorsqu'un capteur est associé à un conditionneur les deux délivrent des informations sous forme de tension. L'information doit donc être transmise dans des conditions qui n'entraînent ni sa dégradation pas des parasites ni sa pertes. Cette transmission peut se faire de plusieurs manières.

## I.3. Généralités sur la transmission [1]

### I.3.1 Représentation des données

Le but d'un réseau ou d'une ligne de transmission est de transmettre des informations d'un ordinateur à une carte ou d'un ordinateur à un autre. Pour cela il faut dans un premier temps décider du type de codage de la donnée à envoyer, c'est-à-dire sa représentation informatique. Celle-ci sera différente selon le type de données, car il peut s'agir de :

- Données sonores.
- Données textuelles.
- Données graphiques.
- Données vidéo.

### **I.3.2 Transmission de données [2]**

La transmission de données est une technique qui transfère des informations codées d'un émetteur à un récepteur grâce à un support de transmission physique. Dans tout système informatique, l'unité d'information est représentée par un octet. Lors de chaque traitement dans ce système informatique, tous les bits sont soit déplacés simultanément d'un endroit mémoire à un autre, soit traités l'un à la suite de l'autre. La transmission de données est "simple" lorsque seules deux machines sont en communication, ou lorsque l'on envoie une seule donnée. Dans le cas contraire il est nécessaire de mettre en place plusieurs lignes de transmission ou bien de partager la ligne entre les différents acteurs de la communication. Ce partage est appelé multiplexage.

La transmission est caractérisée par :

- le sens des échanges.
- le mode de transmission: il s'agit du nombre de bits envoyés simultanément.
- la synchronisation: il s'agit de la synchronisation entre émetteur et récepteur.

### **I.3.3 Codage des signaux de transmission**

Pour qu'il puisse y avoir un échange de données, un codage des signaux de transmission doit être choisi, celui-ci dépend essentiellement du support physique utilisé pour transférer les données, ainsi que de la garantie de l'intégrité des données et de la vitesse de transmission.

### **I.3.4 Codage en bande de base**

C'est le plus utilisé pour toutes les transmissions par conducteurs électriques ou par fibre optique. Chaque station est ajustée sur les points suivants :

Fréquence, respectivement durée de chaque bit.

Niveaux du signal correspondant à un 1 et à un 0.

Les stations les stations doivent travailler sur le même mode de codage :

- NRZ : non retour à zéro
- RZ : retour à zéro (ou codage On-Off)
- Manchester
- Manchester différentiel

### I.3.4.1 Codage On-Off ou NRZ (non retour à zéro)

Un 1 logique est signalé par un niveau du signal. Un 0 logique est signalé par un autre niveau.

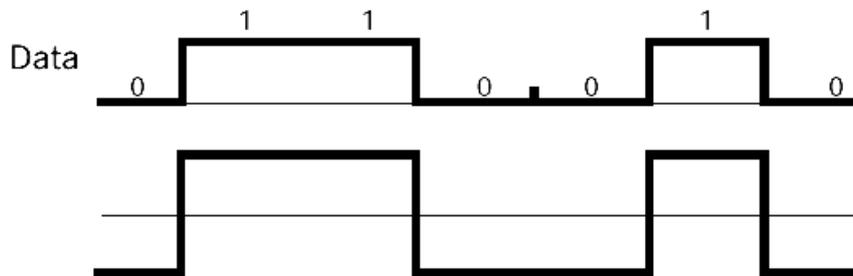


Fig.5 Codage en NRZ.

#### Exemple :

+5V, resp 0V

+12V, resp - 12V

#### Avantage :

- Simplicité

#### Inconvénients :

- Grande bande de fréquences
- Difficulté d'assurer la synchronisation lors d'une longue suite de 0 ou de 1, car il n'y a alors plus de transitions.
- Valeur moyenne du signal très variable, d'où difficulté d'isoler par des transformateurs.

### I.3.4.2 Codage RZ (retour à zéro)

Similaire au codage NRZ, sauf que le signal revient toujours à 0 à la moitié du bit.

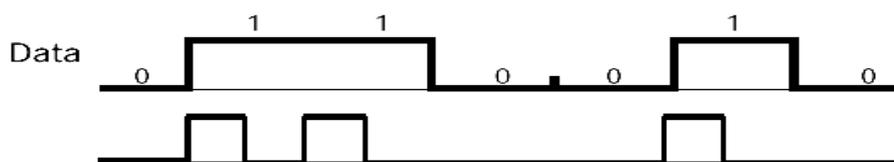


Fig.6 Codage RZ.

#### Avantage :

- Simplicité.
- Il y a toujours une double transition pour transmettre un 1.

#### Inconvénient :

- Grande bande de fréquences.
- Difficulté d'assurer la synchronisation lors d'une longue suite de 0.
- Valeur moyenne du signal très variable.

### I.3.4.3 Codage Manchester

Le signal commute toujours à mi-période :

Un bit à 0 sera codé en une transition de 0 à 1.

Un bit à 1 sera codé en une transition de 1 à 0.

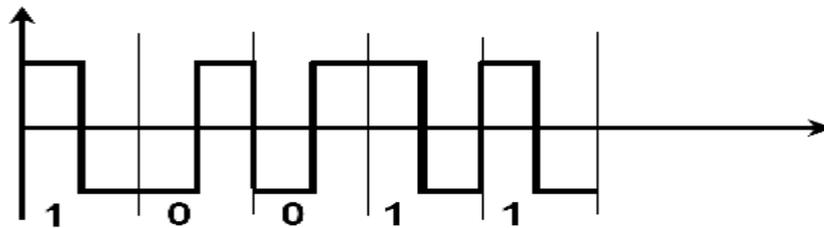


Fig.7 Codage Manchester.

#### Avantage :

- La transition à chaque demi-période est garantie.
- Synchronisation de l'horloge faible et précise.
- Valeur moyenne constante quel que soit le message.

#### Inconvénient :

- Signal riche en harmonique, doublement de la fréquence.

### I.3.4.4 Codage Manchester différentiel

Le signal commute toujours à mi-période, mais tient compte de valeur précédente.

Si le bit est à 0 alors il y aura une transition initiale.

Si le bit est à 1 alors il n'y aura pas de transition initiale.

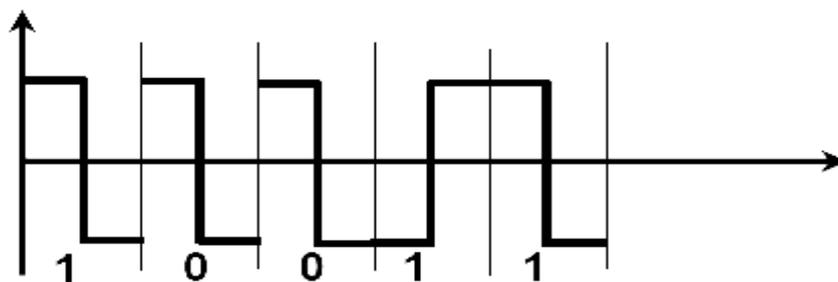


Fig.8 Codage Manchester différentiel.

#### Avantage :

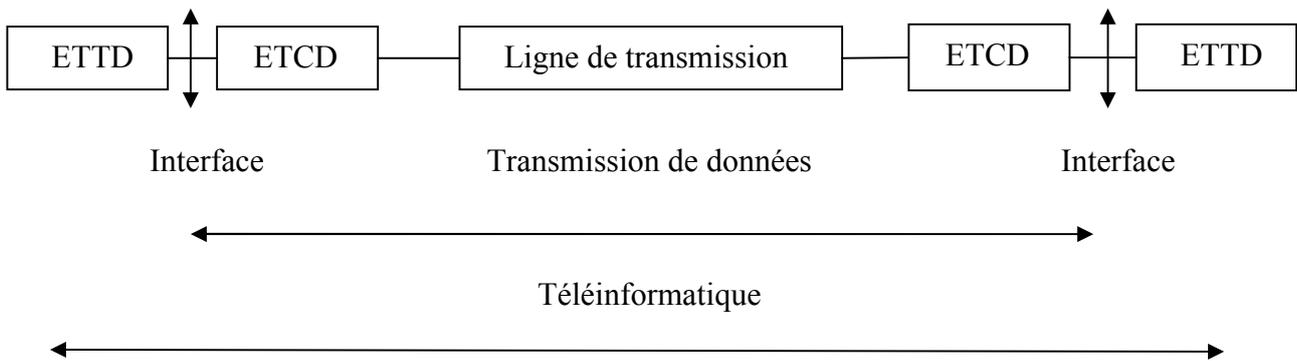
- Idem code Manchester.
- Une inversion de polarité (erreur de câblage) n'inverse pas le signal.

#### Inconvénient :

- Idem code Manches

### I.3.5 Support de transmission des données

Pour que la transmission de données puisse s'établir, il doit exister une ligne de transmission, appelée aussi voie de transmission ou canal, entre les deux machines. Ces voies de transmission sont constituées de plusieurs tronçons permettant de faire circuler les données sous forme électriques, lumineuses ou même acoustiques.

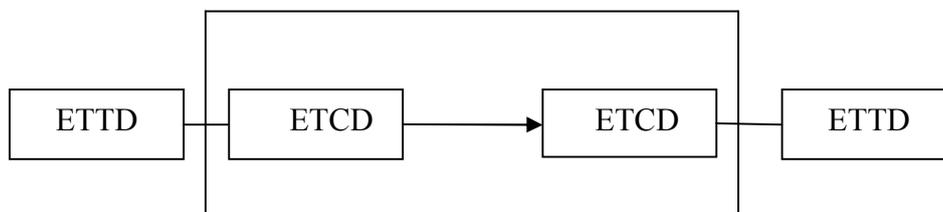


**Fig.9 Synoptique fonctionnelle d'une liaison**

- L'ETTD (équipement terminal de traitement de données)
- L'ETCD (Équipement Terminal de Circuit de Données).

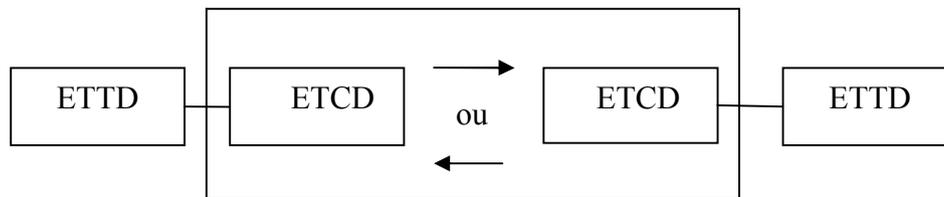
Selon le sens des échanges, on distingue 3 modes de transmission :

- **Transmission simplex** caractérise une liaison (transmission) dans laquelle les données circulent dans un seul sens, c'est-à-dire de l'émetteur vers le récepteur. Ce genre de liaison est utile lorsque les données n'ont pas besoin de circuler dans les deux sens (par exemple de votre ordinateur vers l'imprimante ou de la souris vers l'ordinateur...).



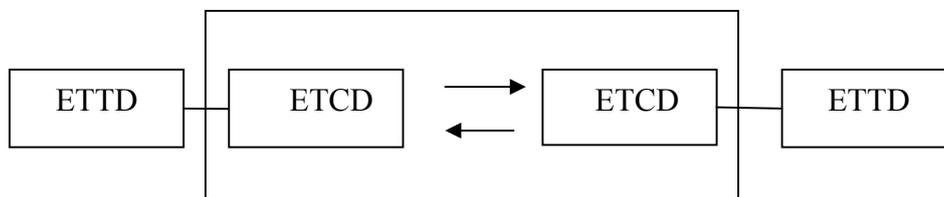
**Fig.10 Transmission simplex**

- **Transmission half-duplex** (parfois appelée liaison à l'alternat ou semi duplex) caractérise une liaison dans laquelle les données circulent dans un sens ou l'autre, mais pas les deux simultanément. Ainsi, avec ce genre de liaison chaque extrémité de la liaison émet à son tour. Ce type de liaison permet d'avoir une liaison bidirectionnelle utilisant la capacité totale de la ligne.



**Fig. 11 Transmission half-duplex**

- **Transmission full duplex** (appelée aussi duplex intégral) caractérise une liaison dans laquelle les données circulent de façon bidirectionnelle et simultanément. Ainsi, chaque extrémité de la ligne peut émettre et recevoir en même temps.



**Fig. 12 Transmission full-duplex**

La transmission de donnée sur une voie de communication peut s'effectuer de différentes manières. Nous différencions deux systèmes de transmission :

- la transmission parallèle.
- la transmission sérielle.

### I.3.6 Transmission série et parallèle [3]

#### a- Transmission parallèle

Si un octet doit être transmis, tous les bits du même octet sont émis simultanément. Cela signifie qu'à chaque bit correspond une liaison conductrice séparée, c'est-à-dire un fil conducteur. Pour la transmission d'un octet, huit liaisons parallèles doivent être présentes entre l'émetteur et le récepteur. Dans ce cas, on parle d'une transmission parallèle.

C'est une transmission rapide mais sensible à l'environnement électrique. De plus, son coût et le nombre de fils la rend inapte pour les longues distances.

#### b. Transmission série

Une transmission série est l'envoi des bits l'un après l'autre, sur le même support physique, à l'inverse de la transmission de données en parallèle où l'envoi des bits est simultané. A l'autre extrémité, côté réception, la démarche est inverse, les bits sont réassemblés pour reconstituer la donnée d'origine ce qui nécessite un montage décodeur (UART : Universal Asynchronous Receiver Transmitter) pour retransformer les données série.

C'est une transmission plus lente mais moins sensible à l'environnement que la transmission parallèle. Elle permet un grand nombre d'applications avec une faible contrainte d'encombrement ; la transmission peut s'effectuer avec trois fils, elle est donc plus économique pour de longues distances. Plus difficile à mettre en œuvre que la transmission parallèle, la transmission en série se révèle cependant plus puissante et plus universelle.

Etant donné les problèmes que pose la liaison de type parallèle, c'est la liaison série qui est la plus utilisée. Toutefois, puisqu'un seul fil transporte l'information, il existe un problème de synchronisation entre l'émetteur et le récepteur, c'est-à-dire que le récepteur ne peut pas a priori distinguer les caractères (ou même de manière plus générale les séquences de bits) car les bits sont envoyés successivement. Il existe donc deux types de transmission permettant de remédier à ce problème :

### I.3.7 Transmission synchrone et asynchrone

#### a- Transmission asynchrone

Dans laquelle chaque caractère est émis de façon irrégulière dans le temps (par exemple un utilisateur envoyant en temps réel des caractères saisis au clavier). Ainsi, imaginons qu'un seul bit soit transmis pendant une longue période de silence le récepteur ne pourrait savoir s'il s'agit de 00010000, ou 10000000 ou encore 00000100. Afin de remédier à ce problème, chaque caractère est précédé d'une information indiquant le début de la transmission du caractère (l'information de début d'émission est appelée *bit START*) et terminé par l'envoi d'une information de fin de transmission (appelée *bit STOP*, il peut éventuellement y avoir plusieurs bits STOP).

#### b- Transmission synchrone

Dans laquelle émetteur et récepteur sont cadencés à la même horloge. Le récepteur reçoit de façon continue (même lorsque aucun bit n'est transmis) les informations au rythme où l'émetteur les envoie. C'est pourquoi il est nécessaire qu'émetteur et récepteur soient cadencés à la même vitesse. De plus, des informations supplémentaires sont insérées afin de garantir l'absence d'erreurs lors de la transmission.

## I.4. Conclusion

Dans ce présent chapitre, nous avons présenté des généralités sur l'acquisition de données telles que les capteurs et le conditionnement de leurs signaux.

Les ordinateurs sont actuellement des plates formes privilégiées pour les applications des acquisitions, de traitement et de restitution des données. Ils offrent en effet une très grande variété d'outils logiciels pour le développement des programmes d'acquisition.

# Chapitre II

Afin de transmettre les informations dans de bonnes conditions il faut suivre des règles et se référer à des normes.

## II.1 Protocoles de communication [4]

Un protocole est un langage commun utilisé par l'ensemble des acteurs de la communication pour échanger des données. Toutefois son rôle ne s'arrête pas là. Un protocole permet aussi :

- L'initiation de la communication.
- L'échange de données.
- Le contrôle d'erreur.
- Une fin de communication.

On classe généralement les protocoles en deux catégories selon le niveau de contrôle des données que l'on désire :

- **Les protocoles orientés connexion:** Il s'agit des protocoles opérant un contrôle de transmission des données **pendant** une communication établie entre deux machines. Dans un tel schéma, la machine réceptrice envoie des accusés de réception lors de la communication, ainsi la machine émettrice est garante de la validité des données qu'elle envoie. Les données sont ainsi envoyées sous forme de flot.
- **Les protocoles non orientés connexion:** Il s'agit d'un mode de communication dans lequel la machine émettrice envoie des données sans prévenir la machine réceptrice, et la machine réceptrice reçoit les données sans envoyer d'avis de réception à la première. Les données sont ainsi envoyées sous forme de blocs (datagrammes).

## II.2. Modèle de référence OSI (Open System Interconnexion) de l'ISO (international organisation standardisation)

### II.2.1 Origine du modèle

Les communications entre systèmes ne sont possibles que si chaque système comprend son destinataire. Il a donc été nécessaire de définir une norme pour permettre à chacun de communiquer avec un réseau existant.

Afin de simplifier la définition des normes de communication, en les situant les unes par rapport aux autres, l'organisation internationale de normalisation (ISO) a lancé en 1977 un projet de définition d'un modèle de référence pour

l'interconnexion de systèmes ouverts, appelé simplement « **modèle de référence OSI** ». La version finale du modèle OSI date de 1984. Il a été défini à partir des expériences dans les réseaux publics, mais a dû ultérieurement être adapté aux réseaux locaux.

## II.2.2 Système ouvert

Un système est dit **ouvert** lorsqu'il permet la communication entre équipements de types différents, pouvant provenir de constructeurs différents, pourvu que ces équipements respectent les règles de communication dans un environnement OSI. Les règles de communication sont publiques.

Et inversement, un système est dit **privé**, lorsqu'il ne permet la communication qu'entre des équipements d'un même type, ou d'un même constructeur, en utilisant des protocoles qui sont propre à lui.

Le modèle OSI constitue un cadre de référence pour l'**interconnexion de systèmes ouverts hétérogènes**. Il s'agit d'un modèle pour élaborer des normes d'interconnexion et de coopération de systèmes répartis. Il est construit selon une **structure en sept couches** qui correspondent à un type de préoccupation ou à un type de problème à résoudre pour pouvoir communiquer.

## II.2.3 Couches du modèle OSI

- Le **protocole de la couche physique** assure le transport de l'information sous forme binaire.
- Le **protocole de liaison** est responsable d'acheminement de blocs d'informations sur le support physique avec un taux d'erreurs qui doit rester négligeable par rapport aux besoins de l'application.
- Le **protocole réseau** est responsable de l'acheminement des paquets de données qui transitent à l'intérieur du système.
- Le **protocole de la couche transport** est responsable du contrôle de l'acheminement des informations de bout en bout, au travers du réseau.
- La **couche session** est responsable de la mise en place et du contrôle du dialogue entre processus distants.
- La **couche présentation** est responsable de la présentation des données échangées par les applications ; cela pour avoir une compatibilité entre tous les matériels raccordés au réseau.

- La **couche application** se préoccupe de la sémantique de l'information et complète la partie syntaxe prise en charge par la **couche 6**.

Pour communiquer, chaque carte réseau doit être informée de toute demande de communication lui étant destinée. Il est également nécessaire d'informer les autres utilisateurs du réseau d'une volonté de communiquer avec un tiers.

Il a donc été décidé de mettre au point une méthode d'adressage logique (donc informatique et non matériel), qui permettrait à un ou plusieurs organismes de régulation de distribuer des adresses à ceux qui en feront la demande. Cette normalisation des adresses se nomme **Internet Protocol (IP)**, et connaît aujourd'hui sa version 4, alors que la version 6 s'apprête à remplacer la version 4 aujourd'hui obsolète.

L'Internet Protocol (IP) se place donc comme couche réseau, puisqu'elle fournit les fonctions et les informations nécessaires au routage et adressage. Un équipement IP n'a pas connaissance nécessairement de la topologie du réseau auquel il est relié, mais sait comment router les paquets (connaissance de la prochaine passerelle pour une destination donnée). Ce processus se résume à « la donnée émise vers la destination **xxx.xxx.xxx.xxx** existe dans ma table de routage et doit aller sur telle interface, où n'existe pas est doit aller sur l'interface par défaut (si elle existe) ».

Deux protocoles viennent combler ce manque : **UDP** (User Datagramme Protocol) et **TCP** (Transmission Control Protocol).

L'**UDP** utilise un mode sans connexion, ne contrôle pas la perte des paquets, chaque paquet et émis sans numérotation vers le destinataire.

Les protocoles **TCP** et **UDP** se placent donc comme couche de transport dans la pile IP, car ce sont eux qui assurent la transmission des données d'un point à l'autre d'un réseau, en gérant les nécessaires réémissions (ou non) des paquets perdus ou altérés, etc.

C'est en cela que TCP/IP est appelé réseau ouvert. Les protocoles utilisés sont normalisés et disponibles pour le monde entier. Chacun peut donc adapter son système propriétaire pour communiquer en TCP/IP, en écrivant les différents composants logiciels répondant aux normalisations TCP/IP.

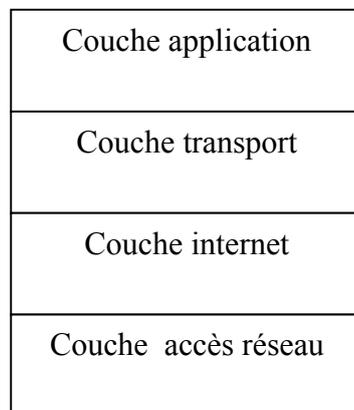
On voit donc que pour une seule communication entre deux systèmes, il est nécessaire d'utiliser plusieurs protocoles.

## II.3 Model de TCP/IP

Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules ou couches) mais en contient uniquement quatre.

TCP/IP définit une architecture en couches qui inclut également, sans qu'elle soit définie explicitement, une interface d'accès au réseau.

Les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.



**Fig. 9** Architecture en couche du model TCP/IP

- **Couche Accès réseau** : elle spécifie la forme sous laquelle les données doivent être acheminées quel que soit le type de réseau utilisé (FDDI, PPP, **Ethernet**, Anneau à jeton (Tok en ring))
- **Couche Internet** : elle est chargée de fournir le paquet de données (datagramme) (**IP**, ARP, RARP)
- **Couche Transport** : elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaître l'état de la transmission (**TCP**, UDP)
- **Couche Application** : elle englobe les applications standard du réseau (Telnet, SMTP, FTP, ...)

Un réseau est un moyen de communication permettant d'interconnecter des équipements informatiques et de partager certaines ressources (de calcul, de stockage, etc.)

Aujourd'hui, les réseaux constituent l'axe autour duquel s'organise l'ensemble des services informatiques. Parmi eux on trouve le réseau Ethernet utilisé dans le domaine de l'informatique et des télécommunications.

En résumé, TCP est un protocole de niveau transport en mode connexion, très complet, permettant de supporter des applications qui souhaitent une bonne qualité de transport de leur information.

## II.4 La technologie Ethernet

La technologie Ethernet utilise la commutation de paquets ; elle a été inventée par Xerox, au début des années 70, puis normalisée en 1978 par Xerox, Intel et Digital Equipment.

Les réseaux Ethernet peuvent être étendus au moyen de répéteurs. Deux répéteurs au plus peuvent être placés entre deux machines quelconques, ce qui limite la longueur d'un réseau Ethernet à environ 1500 mètres. Les connexions sont réalisées au moyen de robinets « taps », appelés également vampires lorsqu'ils mordent le câble. Chaque connexion comporte deux composants électroniques majeurs : un « transceiver » et un coupleur.

Chaque machine connectée à un réseau Ethernet possède un identificateur unique (un entier sur 48 bits) qui constitue son adresse Ethernet.

### II.4.1 Propriétés d'un réseau Ethernet

Un réseau Ethernet est un bus à contrôle distribué, d'une capacité nominale de 10 Mbit/s, basé sur la notion de remise pour le mieux (« best effort delivery »). Les stations sont en compétition les unes avec les autres et l'accès au support n'est pas garanti, des données peuvent donc être perdues. Les informations sont diffusées (« broadcast »), les transceivers ne filtrent pas les données, ce sont les coupleurs qui éliminent les paquets non désirés. Il n'y a pas d'autorité centrale responsable de l'allocation du support, la méthode d'accès est dite CSMA/CD (« Carrier Sense Multiple Access with Collision Detect »).

Le principe de CSMA est que plusieurs machines peuvent accéder simultanément au câble, chaque machine détermine si le câble est disponible en y détectant une onde

porteuse. En l'absence de transmission, l'interface de la machine commence à transmettre pour une durée limitée (afin de ne pas monopoliser le support).

CD : (détection de collision) Normalement, une station n'émet que s'il n'y a pas de trafic sur le réseau. Lorsqu'une station commence à émettre, les bits émis mettent un certain temps pour arriver aux autres stations (du fait du temps de propagation du signal sur le câble). Supposons qu'une deuxième station veuille émettre avant que le message de la première soit parvenu à sa hauteur. Elle considère que la voie est libre et émet son message. Il y a alors collision des deux messages. Le standard Ethernet prévoit cette situation : chaque nœud émetteur est capable de détecter le changement de niveau d'énergie et de l'interpréter comme une collision. Lorsqu'une collision se produit, les deux postes cessent immédiatement d'émettre. Ils essaient ensuite à nouveau, après un délai librement choisi.

## II.4.2 Variantes de l'Ethernet

La technologie Ethernet a été améliorée de multiples manières :

- **10Base5 : le support coaxial**

10Base5 est le support physique natif d'Ethernet qui supporte une vitesse de transmission de 10 Mbit/s sur le fameux câble coaxial épais, L'identifiant « 10Base5 » est une contraction de la vitesse de 10 Mbit/s, une transmission du type bande de base, et de la longueur maximale de segment de 500 m. La connexion des stations sur le câble coaxial est réalisée au moyen d'un transceiver,( transceiver est une contraction de transmitter-receiver, émetteur-récepteur.)

- **10Base2 : câble coaxial fin**

10Base2 supporte les vitesses de transmission de 10 Mbit/s sur un câble coaxial fin le support 10Base2 est en plusieurs points semblable au 10Base5 : même signal, même encodage et même mécanisme de détection de collision mais sur un câble coaxial plus fin qui lui permet d'être plus flexible, moins lourd, moins cher et plus facile à installer. La connexion des stations ne nécessite pas de transceiver mais un simple connecteur BNC en T suffit. En revanche, le câble coaxial fin ne présente pas des caractéristiques de transmission de très bonne qualité et est de ce fait limité en distance à 185 m avec un nombre de stations de 30 maximum par segment.

- **10BaseT : paire torsadée**

10BaseT supporte les vitesses de transmission de 10 Mbit/s sur deux paires de catégorie 3 ou plus. L'utilisation largement diffusée de la paire torsadée a fait de 10BaseT le support le plus populaire d'Ethernet.

10BaseT utilise une paire de fils pour transmettre les données et une autre paire pour recevoir les données. Le câble se termine à chaque extrémité par un connecteur RJ45 à 8 pôles .

Toutes les connexions 10BaseT sont des liaisons point à point. Cela implique que le câble ne peut avoir qu'un maximum de deux transceivers Ethernet, un à chaque extrémité. Une extrémité du câble est typiquement reliée à un hub 10BaseT, l'autre étant directement reliée à la station au moyen de la carte réseau sur laquelle est enfiché le connecteur RJ45 sans ajout d'autre composant supplémentaire.

### II.4.3 L'adressage Ethernet

L'interface filtre les paquets qui ne sont pas destinés à la machine. Chaque machine (en fait chaque coupleur) connecté à un réseau Ethernet possède un identificateur unique (un entier sur 48 bits) qui constitue son adresse Ethernet.

Les adresses Ethernet sont partagées en 3 types :

- ✓ adresse d'une interface de réseau
- ✓ adresse de diffusion sur un réseau
- ✓ adresse de diffusion de groupe

Par convention, l'adresse de diffusion est réservée à l'émission vers toutes les machines du réseau.

### II.4.4 Structure des trames Ethernet

Le réseau Ethernet est une connexion entre machines au niveau liaison, les données échangées sont structurées en paquets appelés trames (« frames »). Les trames ont une longueur variable entre 64 octets et 1518 octets.

| Préambule | Destination | Source  | Type    | Données,      | CRC     |
|-----------|-------------|---------|---------|---------------|---------|
| 64 bits   | 48bits      | 48 bits | 16 bits | 64 à 1500bits | 32 bits |

Le préambule contient une alternance de 64 bits à 1 et 0, terminée par 2 bits à 1. Le type de trame permet l'utilisation simultanée de plusieurs protocoles, le CRC (Cyclic Redundancy Check) contrôle la validité des informations.

La structure de la trame Ethernet a été normalisée par l'IEEE (Institute of Electrical and Electronics Engineers), après avoir été défini à l'origine par le triumvirat d'industriels Xerox, Digital et Intel. Deux trames Ethernet coexistent donc, la version primitive du triumvirat fondateur et celle de la normalisation par l'IEEE.

## **II.5 Conclusion**

Le protocole TCP quant à lui assure un transfert en ouvrant une session de communication avant tout dialogue, puis en numérotant les paquets pour la reconstruction, il se place donc comme couche de transport dans la pile IP, car c'est lui qui assure la transmission des données d'un point à l'autre d'un réseau, en gérant les nécessaires réémissions (ou non) des paquets perdus ou altérés

A rectangular area with rounded corners, filled with a light brown wood-grain texture. This area is framed by a dark, solid border. The text 'Chapitre III' is centered within the wood-grain area.

# Chapitre III

Un microcontrôleur est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique (quelques milliwatts en fonctionnement, quelques nano watts en veille), une vitesse de fonctionnement plus faible (quelques mégahertz à quelques centaines de mégahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Il est conçu pour se suffire à lui-même possède sa propre RAM, sa ROM et le programme pour exécuter les tâches, tel un système d'exploitation.

### **III.1. Description interne d'un microcontrôleur [6]**

Un microcontrôleur se présente sous la forme d'un circuit intégré réunissant tous les éléments d'une structure à base de microprocesseur. Voici généralement ce que l'on trouve à l'intérieur d'un tel composant :

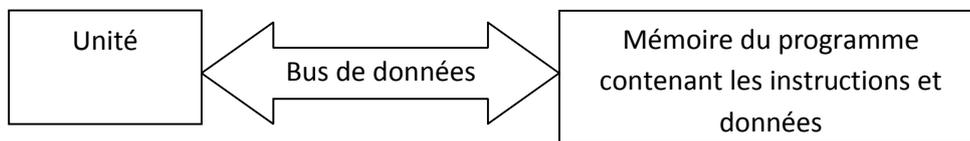
- Un microprocesseur (C.P.U.),
- De la mémoire de donnée (RAM et EEPROM),
- De la mémoire programme (ROM, OTPROM, UVPROM ou EEPROM),
- Des interfaces parallèles pour la connexion des entrées / sorties,
- Des interfaces séries (synchrone ou asynchrone) pour le dialogue avec d'autres unités,
- Des timers pour générer ou mesurer des signaux avec une grande précision temporelle,
- Des convertisseurs analogique / numérique pour le traitement de signaux analogiques.

## III.2. Les différentes architectures des microprocesseurs

IL existe deux différentes architectures, à savoir l'architecture dite de Von-Neumann, et l'architecture Harvard.

### III.2.1 L'architecture Von-Neumann

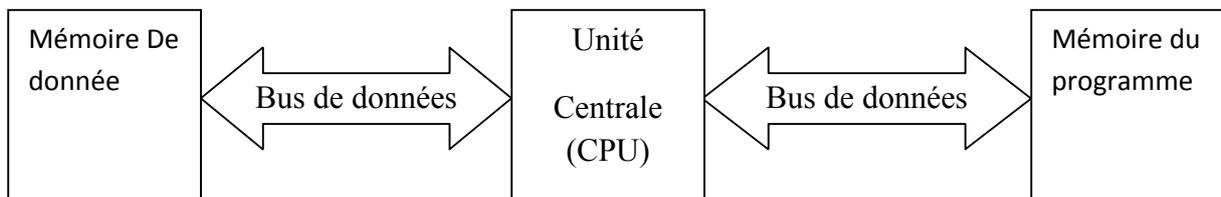
La majorité des microprocesseurs et microcontrôleurs actuels emploient une architecture interne dite de Von-Neumann. C'est en fait une architecture commune à celle que l'on rencontre dans les microcontrôleurs. Cette architecture est basée sur un bus de données unique. Celui-ci véhicule tour à tour les instructions et les données qui leur sont associées,



**Fig.1 Architecture Von-Neumann**

### III.2.2 L'architecture Harvard

C'est en fait dans cette architecture que les instructions et les données sont clairement différenciées et sont véhiculées sur deux bus différents comme dans le cas de la figure.



**Fig.2 Architecture Harvard**

On peut noter qu'il existe 2 catégories de microcontrôleurs : les CISC et les RISC.

- CISC (Complex Instruction Set Computer) : Ces microcontrôleurs possèdent un nombre important d'instructions. Chacune d'elles s'exécute en plusieurs périodes d'horloges
- RISC (Reduced Instruction Set Computer) : Ces microcontrôleurs possèdent un nombre réduit d'instructions. Chacune d'elles s'exécute en une période d'horloge. L'ALU (unité arithmétique et logique) dispose en effet en une seule fois de toutes les informations nécessaires à l'exécution de l'instruction.

Un circuit de type RISC présente aussi un certain nombre de particularités propres à accroître sa vitesse de fonctionnement.

Les microcontrôleurs à architecture RISC utilisent des instructions codées sur un seul mot (de 12 ou 14 bits de long), ce qui présente deux principaux avantages. Tout les emplacements de la mémoire du programme contiennent une instruction, c'est le premier avantage, le second est qu'un seul cycle machine suffit pour lire le code complet d'une instruction, d'où un gain en vitesse d'exécution.

Les circuits RISC utilisent une structure de type pipe-line qui leur permet d'exécuter une instruction tout en recherchant la suivante en mémoire d'où, là encore, accroissement de la vitesse d'exécution.

Les instructions d'un circuit RISC peuvent être exécutées sur tous les registres avec tous les modes d'adressage. Cela simplifie le travail du programmeur.

- Avantage :
  - Encombrement réduit,
  - Circuit imprimé peu complexe,
  - Faible consommation,
  - Coût réduit.
- Inconvénient :
  - Système de développement onéreux,
  - Programmation nécessitant un matériel adapté.

Il existe trois statuts physiques différents pour les ports, c'est à dire que c'est la fabrication électronique des ports qui décide dans quel mode il sera capable de fonctionner. Certains microcontrôleurs possèdent des ports dont certaines broches sont

en entrée et d'autres en sortie. Enfin, il est possible sur certains modèles de définir logicielle ment dans quel mode le Port va fonctionner;

- Port en entrée : il est capable de recevoir des informations en provenance de l'extérieur.
- Port en sortie : il émet des informations du microcontrôleur vers l'extérieur.
- Port bidirectionnel : il est capable, pour toutes ses broches ou certaines d'entre elles, de recevoir ou d'émettre des données.

Les microcontrôleurs AVR d'Atmel sont développés et optimisés pour une programmation en C. Une puissance et une rapidité d'exécution peu communes pour leur coût, une mémoire flash reprogrammable électriquement en quelques secondes sans qu'ils soient démontés de l'application où ils sont installé

### III.3. Présentation Physique [7]

L'ATMEGA ce présente sous la forme d'un circuit intégré, famille CMOS (Complementary Metal-Oxide-semiconducteur) à 40 broches pour le modèle ATMEGA32 en boîtier PDIP ou le boîtier TQFP/MLF à 44 broches.

Comme vous pouvez le voir, les ports sont composés de 8 broches, ce qui forme un octet. Chaque broche a son propre poids ; on nomme généralement chaque broche en prenant le nom du port et le poids d'une broche.

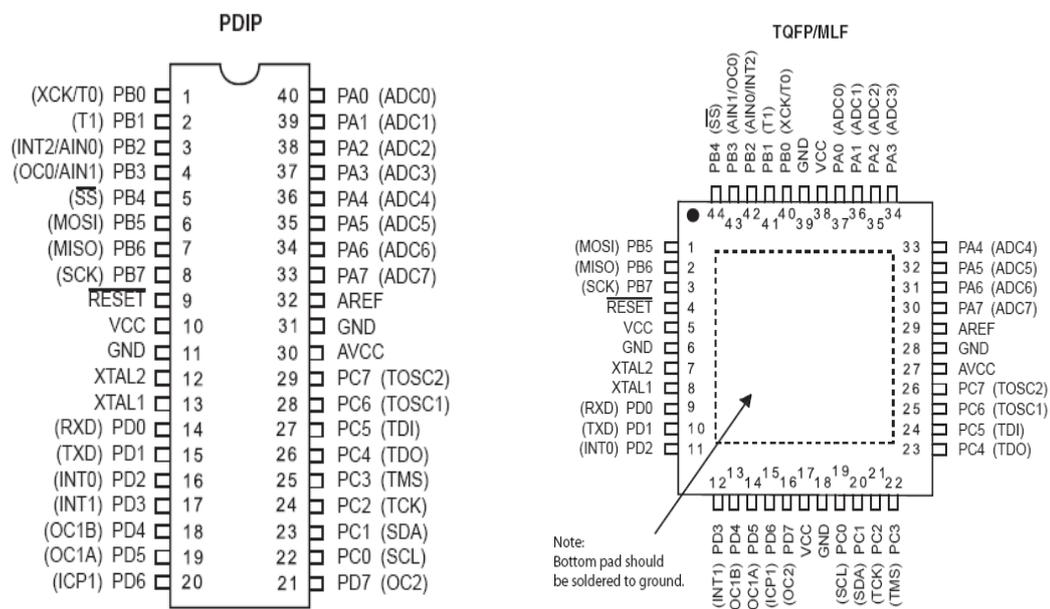


Fig.3 Brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC.

### III.4. Synoptique

L'AVR Tmega32 d'Atmel est un microcontrôleur 8 bits basé sur l'architecture RISC AVR est de technologie CMOS (Complementary Metal-Oxide-semiconducteur) basse consommation.

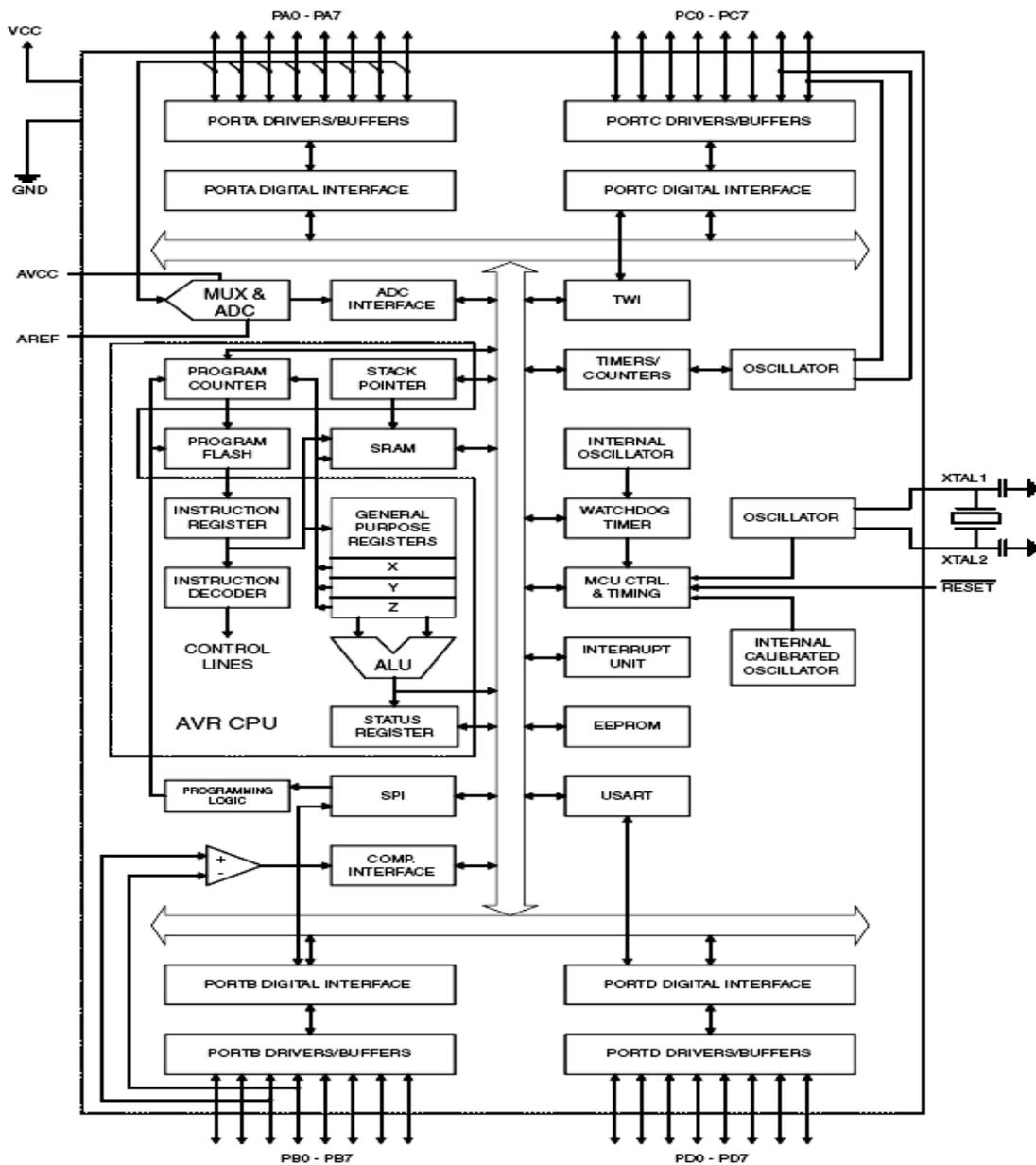


Fig.4 Synoptique générale d'un ATMEGA.

### III.5. Descriptions des broches

- Port A (PA7... PA0) le Port A est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi pour les entrées analogiques du convertisseur A/D. Le Port A (comme le B, C et D) est en position trois états quand une condition de reset devient active, même si l'horloge ne court pas.
- Port B (PB7... PB0) le Port B est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi de comparateur analogique (sortie sur PB2, PB3), ou de SPI.
- Port C (PC7... PC0) le Port C est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi comme oscillateur pour le Timer/Compteur2 et d'interface I2C.
- Port D (PD7... PD0) le Port D est un port d'entrée-sortie à 8 bits bidirectionnel avec des résistances internes de tirage (choisi pour chaque bit). Il sert aussi d'USART et d'entrées pour les interruptions externes.
- RESET déclenché par un front descendant maintenue plus de 50 ns il produira le Reset du microcontrôleur, même si l'horloge ne court pas.
- XTAL1 Entrée de l'oscillateur externe ou libre pour l'horloge interne.
- XTAL2 Production de l'amplificateur d'oscillateur.
- AVCC est une broche de tension d'alimentation pour le Convertisseur A/D qui doit être connectée à
- VCC via un filtre passe-bas pour éviter les parasites.
- AREF est l'entrée de référence analogue pour le Convertisseur A/D avec une tension dans la gamme de 2 V à AVCC avec filtre passe bas.
  
- AGND masse Analogique. Si la masse analogique est séparée de la masse générale, brancher cette broche sur la masse analogique, sinon, connecter cette broche à la masse générale GND.
- VCC broches d'alimentation du microcontrôleur (+3 à +5V).
- GND masse de l'alimentation.

### III.6. CPU du corps AVR

Cette section traite l'architecture en général du Corps AVR d'Atmel. La fonction principale du corps de la CPU est de veiller à une exécution correcte du programme. La CPU doit donc être capable d'accéder aux mémoires, d'effectuer des calculs, contrôle des périphériques, et interruptions. Elle est constituée des éléments suivants :

- D'Unité Arithmétique et logique (ALU) qui fonctionne en liaison directe avec tous les 32 registres de travail d'usage général divisés en trois catégories principales l'arithmétique, logique et bit-fonctions.
- Le registre d'état SREG (stat register) contient les informations sur le résultat de l'instruction arithmétique récemment exécutée, Il indique et autorise aussi le fonctionnement des interruptions.
- Le registre de gestion de la pile (Stack Pointer) est principalement utilisé pour stocker : temporairement des données, les variables locales et les adresses de retour après interruptions.

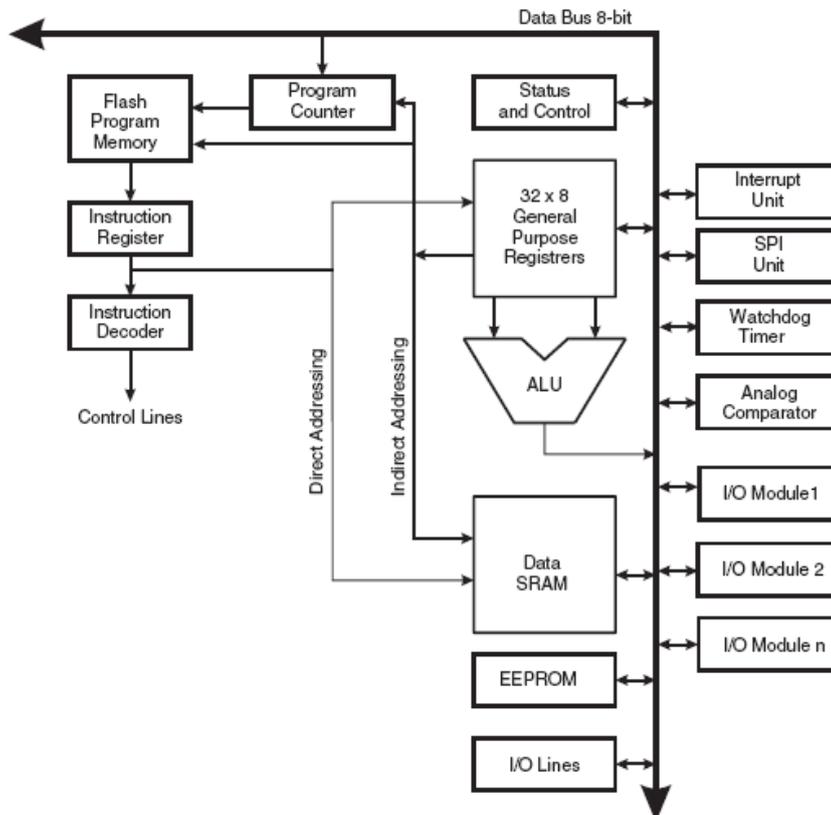


Fig.5 Architecture générale des circuits AVR d'Atmel

### **III.7. Le Plan Mémoire**

Trois types de mémoire sont utilisés dans la série ATMEGA, la mémoire programme FLASH, la mémoire de donnée SRAM et la mémoire morte de type EEPROM.

#### **III.7.1 La mémoire programme(FLASH)**

La mémoire programme permet de stocker et de faire fonctionner le microcontrôleur, il contient de 4 à 256 Ko de programme selon le modèle du microcontrôleur. Toutes les instructions AVR sont sur 16 bits ou 32 bits de large. Pour plus de sécurité, l'espace mémoire (Flash) est divisé en deux sections, section Programme de démarrage et section Programme d'application. Le nombre d'écriture sur cette mémoire est limité à 10.000, largement suffisant pour la majorité des applications. La figure6 donne un exemple de l'adressage de la mémoire FLASH du modèle ATMEGA 32.

#### **III.7.2 La mémoire de donnée(SRAM)**

La mémoire de donnée contient les 32 registres de travail, les 64 registres de commande et la mémoire SRAM pour les variables du programme de 2048 octets pour le modèle ATMEGA 32.

#### **III.7.3 La mémoire morte(EEPROM)**

La mémoire morte est de type EEPROM d'accès plus complexe contiendra la configuration du programme et les données importantes qui seront sauvé pendant l'absence de courant électrique. On peut écrire et effacer jusqu'à 100.000 fois dans l'EEPROM. La taille de l'EEPROM est fonction du modèle de microcontrôleur ATMEGA (de 256 bits à 8 Ko). L'ATMEGA32 contient 1024 octets de données dans la mémoire EEPROM.

### **III.8. L'Horloge du Système et Option**

L'horloge peut être soit interne, soit externe. Le choix du type d'horloge est déterminé lors de la programmation de la mémoire FLASH du microcontrôleur.

Pour une horloge externe un oscillateur à quartz est connecté sur la pin XTAL1 et la pin XTAL2 du microcontrôleur. La fréquence du quartz est de 4 à 16 MHz de type cristal ou un résonateur céramique. Pour conduire un dispositif externe avec la source d'horloge interne, la pin XTAL2 doit être laissée en l'air, tandis que l'on connecte la pin XTAL1 au dispositif à piloter.

Pour l'oscillateur du Timer temps réel, les pins TOSC1 et TOSC2 (PC6 & PC7) peuvent recevoir un cristal de 32 768 Hz directement connecté entre ces pins sans condensateur externe. L'application d'une source d'horloge externe sur la pin TOSC1 n'est pas recommandée. Ce système permet de cadencer le Timer avec des valeurs de temps en sous-multiple de la seconde.

### III.9. Les interruptions

Afin d'éviter les boucles d'attente sans fin et les risques de blocage, il est assez simple de mettre en œuvre les interruptions du microcontrôleur qui dispose d'un grand nombre de possibilité d'activation

La gestion des interruptions paraît souvent difficile et pourtant ce n'est pas plus compliquer en premier lieu lever le drapeau d'autorisation d'interruption général et il faut donner au périphérique concerné la possibilité d'interrompre provisoirement le programme principal en réglant les bits d'interruption dans les registres respectifs.

Les sources d'interruptions possibles:

- RESET : Reset Externe, Début d'Alimentation, Reset Erreur d'Alimentation, Watch dog Reset, et Reset JTAG AVR.
- INT0 ,1 ,2 : Interruption Externe 0, 1,2.
- TIMER2 COMP : Comparaison Timer/Compteur2.
- TIMER2 OVF : Débordement Timer/Compteur2.
- TIMER1 CAPT : Evénement de Capture du Timer/Compteur1.
- TIMER1 COMPA : Comparaison A Timer/Compteur1.
- TIMER1 COMPB : Comparaison B Timer/Compteur1.
- TIMER1 OVF : Débordement Timer/Compteur1.
- TIMER0 COMP : Comparaison Timer/Compteur0.
- TIMER0 OVF : Débordement Timer/Compteur0.
- SPI, STC : Transfert Série Complet.
- USART, RXC USART : Réception Rx Complet
- USART, UDRE USART : Registre de Donnée Vide.
- USART, TXC USART : Transmission Tx Complet.
- ADC : Conversion Complet.
- EE\_RDY EEPROM : Prêt.
- ANA\_COMP : Comparateur Analogique.
- TWI : Interface Série I2C (TWI).
- SPM\_RDY : Programme de Stockage de la Mémoire Prêt.

### III.10. Les Entrées/Sorties (PORTx)

Les microcontrôleurs ATMEGA32 dispose de quatre ports d'entrées/sorties numériques (A, B, C, D) pour communiquer avec l'extérieur. Ces ports sont multidirectionnels et configurable broche à broche soit en entrée, soit en sortie.

D'autres modes sont aussi utilisables comme des entrées analogiques, des fonctions spéciales de comparaison, de communication synchrone, ...

Pour éviter le flottement des valeurs en entrée d'un port, il est possible de mettre des résistances de tirage entre la broche du port et VCC (+3 ou +5 V), chaque port est configurable avec ou sans résistance de tirage.

Pour contrôler le mode de fonctionnement de chaque broche d'un port, 3 registres sont mis à notre disposition :

- DDRx : Registre de direction (Mode Entrée, Sortie)
  - PORTx : Registre de données (Donnée à lire ou à écrire)
  - PINx : Registre d'état (Donnée disponible)
- X : Représente le nom de port (A, B, C ou D).

Chaque registre est configurable bit à bit, c'est à dire que l'on peut utiliser sur le même port des fonctions en entrée et/ou en sortie simultanément. Par exemple, on peut avoir les quatre premiers bits en entrée et les quatre derniers bits en sortie sur un même port

SFIOR : est un registre spécial permet de modifier le fonctionnement des entrées/sorties pour les ports.

### III.11. LE TIMER/COMPTEUR(TIMER)

Un Timer/Compteur est un ensemble logique qui permet d'effectuer du comptage et permet de réaliser les fonctions suivantes :

- Génération d'un signal périodique modulé ou non en largeur d'impulsion,
- Génération d'une impulsion calibrée,
- Temporisation,
- Comptage d'événements.

Plusieurs registres associés au Timer permettent de configurer les différents modes.

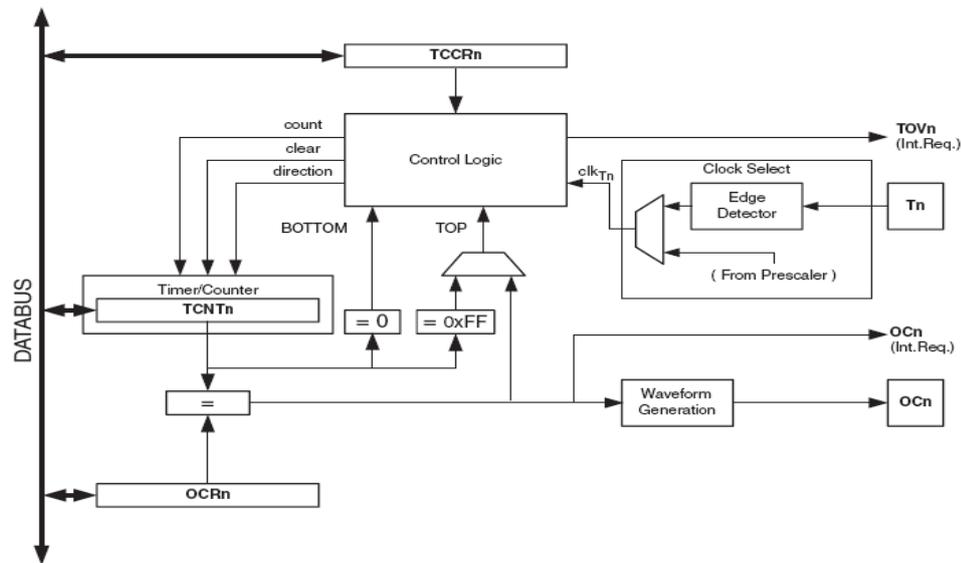


Fig.6 Synoptique du Timer/Compteur.

### III.11.1 Timer 0

Le Timer/Compteur0 est un compteur à 8 bits avec les particularités principales :

- ✓ Remise à 0 du compteur sur Comparaison (Rechargement Automatique)
- ✓ Générateur de Modulation de Phase en Largeur d'Impulsion (PWM)
- ✓ Générateur de Fréquence
- ✓ Compteur d'Événement Externe
- ✓ Horloge à 10 bits avec pré diviseur
- ✓ Interruption sur Débordement et Comparaison (TOV0 et OCF0).

Le Timer0 utilise 3 registres spécifiques et deux registres d'interruption.

- Registre TCCR0 (Timer/Counter Control Register): Registre de contrôle du Timer 0, sélection du facteur de pré division de l'horloge de comptage ou incrémentation par signaux externe sur broche T0.
- Registre TCNT0 (Timer/Counter Register): Le registre TCNT0 du Timer/Compteur0 est accessible en lecture / écriture en permanence.
- Registre OCR0 (Output Compare Register): Le registre OCR0 du Timer/Compteur0 est accessible en lecture / écriture en permanence.
- Registre TIFR (Timer/Counter Interrupt Flag) : Le registre TIFR indique l'état des interruptions internes du Timer0 et du Comparateur. Les interruptions

seront prises en charge si le bit correspondant dans le registre TIMSK a été activé (mis à 1).

- Registre TIMSK (Timer/Counter Interrupt Mask) : Le registre TIMSK définit individuellement le masque pour les modules Timer, Compteurs et Comparateurs. Le masque autorise les interruptions s'il est à 1, s'il est à 0 l'interruption ne sera pas prise en compte.

### III.11.2 TIMER1

Le Timer/Compteur1 est un compteur à 16 bits, permet le cadencement précis des programmes, la gestion d'événement, la génération d'onde, etc. 15 modes sont disponibles avec le Timer et les particularités principales sont :

- ✓ Une Conception 16 bits totale.
- ✓ Deux Unités de Comparaison Indépendante.
- ✓ Double Comparateur.
- ✓ Une Unité de Capture d'Entrée à faible Bruit.
- ✓ Comparateur avec remise à 0 Automatique.
- ✓ Générateur de Modulation de Phase en Largeur d'Impulsion Correct (PWM).
- ✓ Générateur d'onde PWM Périodique.
- ✓ Générateur de Fréquence.
- ✓ Compteur d'Événement Externe.
- ✓ Quatre Sources Indépendant d'Interruption (TOV1, OCF1A, OCF1B et ICF1).

Le Timer1 utilise 5 registres spécifiques et deux registres d'interruption.

- Registre TCCR1A & B (Timer/Counter1 Contrôle Register A & B): Le registre de control du Timer/Compteur1 partie A & partie B,
- Registre TCNT1H & TCNT1L (Timer/Counter1 Register): Les deux registres TCNT1H et TCNT1L, combiné dans TCNT1 donnent l'accès direct au compteur à 16 bits du Timer1.
- Registre OCR1AH and OCR1AL (Output Compare Register 1 A): Les deux registres OCR1AH et OCR1AL, combiné dans OCR1A donnent l'accès direct au comparateur A à 16 bits du Timer1.
- Registre OCR1BH and OCR1BL (Output Compare Register 1 B): Les deux registres OCR1BH et OCR1BL, combiné dans OCR1B donnent l'accès direct au comparateur B à 16 bits du Timer1.

- Registre TIFR (Timer/Counter Interrupt Flag): Le registre TIFR indique l'état des interruptions internes des modules Timer et Comparateur.
- Registre TIMSK (Timer/Counter Interrupt Mask): Le registre TIMSK définit individuellement le masque pour les modules Timer, Compteurs et Comparateurs.

Le Timer/Compteur0 et le Timer/Compteur1 partagent le même module de pré division, mais les Timer/Compteurs peuvent avoir des options différentes de pré diviseur.

Le Timer/Compteur peut être cadencé directement par l'horloge interne du système permettant une fréquence d'horloge maximale ( $F_{clk\_I/O}$ ). Autrement, l'un des quatre bits du pré diviseur peut être employé comme la source d'horloge avec un facteur de pré division équivalent à :  $F_{clk\_I/O}/8$ ,  $F_{clk\_I/O}/64$ ,  $F_{clk\_I/O}/56$ , ou bien  $F_{clk\_I/O}/1024$ .

Comme on peut appliquer une source d'horloge externe à la broche T0/T1 peut être employée comme horloge de Timer/Compteur ( $clkT1/clkT0$ ). Le signal sur T0/T1 est échantillonné une fois par cycle d'horloge système par la logique de synchronisation.

### III.11.3 Timer2

Le Timer/Compteur2 à usage général à 8 bits avec les particularités principales :

- ✓ Simple Compteur à 8 bits
- ✓ Comparaison à Rechargement Automatique.
- ✓ Générateur de Modulation de Phase en Largeur d'Impulsion Correct (PWM).
- ✓ Générateur de Fréquence.
- ✓ Horloge à 10 bits de Pré diviseur.
- ✓ Source d'Interruption sur Débordement et Comparaison (TOV2 et OCF2).
- ✓ Horloge Externe 32 768 Hz par Quartz Indépendant.

Le Timer2 utilise 4 registres spécifiques et deux registres d'interruption.

- Registre TCCR2 (Timer/Counter2 Control Register A): Le registre de contrôle du Timer/Compteur2.
- Registre TCNT2 (Timer/Counter2 Register): Le registre TCNT2 donne l'accès direct au compteur à 8 bits du Timer2.
- Registre OCR2 (Output Compare Register 2): Le registre OCR2 donne l'accès direct au comparateur à 8 bits du Timer2.

- Registre ASSR (Asynchronous Status Register): Le registre permet de sélectionner le mode RTC, soit l'utilisation du quartz connecté sur TOSC1/2 pour créer une horloge basé sur le temps calendaire (à la seconde).
- Registre TIFR (Timer/Counter Interrupt Flag): Le registre TIFR indique l'état des interruptions internes des modules Timer et Comparateur.
- Registre TIMSK (Timer/Counter Interrupt Mask): Le registre TIMSK définit individuellement le masque pour les modules Timer, Compteurs et Comparateurs.

### III.12. watchdog

Le Watchdog est un compteur dont la source d'horloge est indépendante du quartz du microcontrôleur, mais pas de l'alimentation. La fréquence de fonctionnement est de 1 MHz pour une tension d'alimentation de 5 V et 350 KHz pour 3 V.

Le compteur part de 0, incrémente de 1 à chaque top d'horloge et revient à 0 après la valeur 255 et on recommence, tout débordement du compteur (255 -> 0) génère un Reset automatique. En pratique, le watchdog intervient et effectue le reset du microcontrôleur si celui-ci n'est pas effectué par l'instruction "WDR" (WatchDog Reset) dans le temps établi par les broches 0, 1 et 2 du registre "WDTCR".

Les registres qu'utilise le watchdog

- Registre WDCTR : C'est le registre de contrôle du Watchdog
- WDE (Watchdog Enable): Marche/Arrêt du Watchdog.
- WDP2, 1, 0 (Watch dog Timer Prescaler2, 1, 0) : Sélection du facteur de pré division du signal d'horloge, en fait vous définissez l'intervalle maximum de temps pour la remise à 0 RAZ

### III.13. L'interface Série Synchrones SPI

L'interface SPI est l'abréviation Serial Peripheral Interfacel Synchronous, soit Interface Série Synchrones. Ce type de périphérique génère les signaux d'horloge de synchronisation, arbitré par un Maître.

L'interface SPI permet le transfert de données ultrarapide synchrone entre l'ATMEGA et des périphériques ou entre plusieurs dispositifs. L'interface SPI inclut les propriétés suivantes :

- ✓ Transfert de Données Full Duplex à Quatre Fils Synchrone.
- ✓ Maître ou Esclave
- ✓ Transfert de Données avec LSB d'abord ou MSB en premier
- ✓ Sept Taux de Transfert Programmables.
- ✓ Drapeau de Fin de Transmission pour Interruption.
- ✓ Protection de Drapeau de Collision.
- ✓ Mode Inoccupé.
- ✓ SPI Mode Maître à Vitesse Double (CK/2).

Le SPI comporte quatre broches, appeler ligne de communication :

- MISO : Master In Slave Out entrée des données série sur le Maître, sorties sur l'Esclave.
- MOSI : Master Out Slave sortie des données série pour le Maître, entrée pour l'esclavage.
- SCK : Serial Clock signaux d'horloge de synchronisation des circuits connectés sur le Bus. Ces signaux d'horloges sont générés par le Maître.
- SS : Slave Select sélection du mode Maître de la liaison, le Maître la met à l'état haut.

L'interface SPI utilise 4 registres, ils sont au nombre de 3 :

- Registre SPCR (SPI Control Register): Le registre de contrôle de l'interface SPI.
- Registre SPSR (SPI Status Register): Le registre de Statut du SPI.
- Registre SPDR (SPI Data Register): Le registre de données.

### III.14. L'interface Série USART

L'USART est l'abréviation de Universal Synchronous and Asynchronous Receiver and Transmitter, soit Interface Série Synchrone et Asynchrone avec les caractéristiques suivantes :

- Générateur interne de fréquence de cadencement.
- Asynchrone et Synchrone opération.
- Horloge Maître ou Esclave.
- Echanges de données sur 5 à 9 bits et 1 ou 2 bits de stop.
- Gestion des parités.
- Communication en Full duplex (peut émettre et recevoir en même temps).
- Protection des débordements.
- Détection de faux départs de transmission.
- Filtrage de l'entrée.
- Plusieurs interruptions programmables sur le mode émission et réception.
- Mode double vitesse de communication.

L'application principale de ce périphérique est la communication entre le microcontrôleur et un ordinateur via le port série RS232

### III.15. L'interface I2C (TWI)

L'interface à deux conducteurs périodique TWI est une dérivé de l'interface I2C de Philips, c'est une nouveauté du modèle ATMEGA, il n'existe pas dans les versions précédentes AT89 et AT90.

Le protocole TWI permet de connecter jusqu'à 111 systèmes différents employant seulement deux lignes de bus bidirectionnelles, une pour l'horloge SCL et une pour les données SDA. Le seul matériel externe nécessaire pour la mise en œuvre du bus est un simple jeu de résistance pour chacune des lignes (R1, R2).

Tous les systèmes connectés au bus ont des adresses individuelles et les mécanismes de control sont inhérents au protocole I2C.

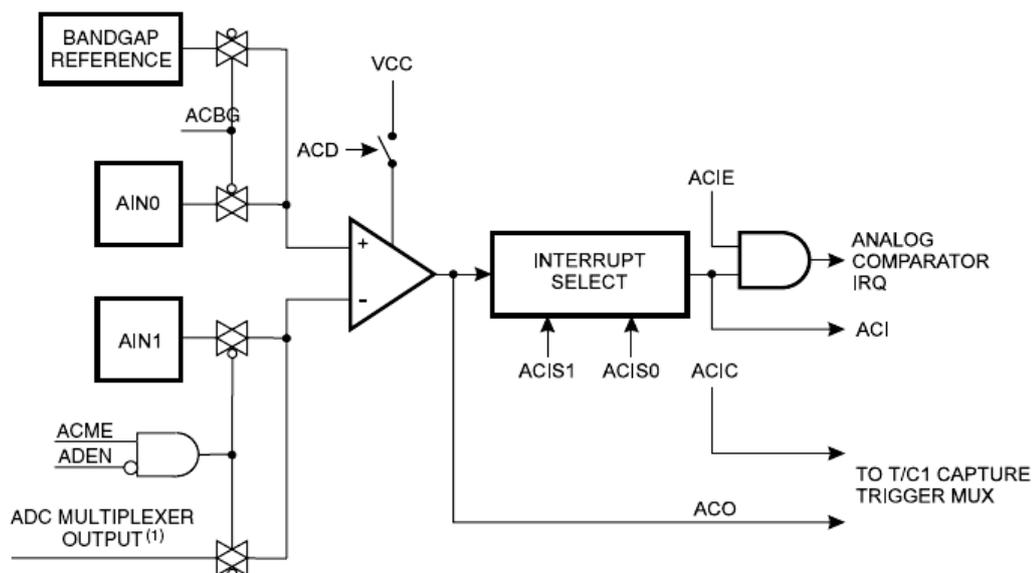
### III.16. Le comparateur analogique

Le comparateur analogique compare les valeurs d'entrée sur la broche positive AIN0 (PB2) et la broche négative AIN1 (PB3). Quand la tension sur la broche AIN0 est plus haute que la tension sur la broche AIN1, le comparateur analogique ACO est

mis à 1. Le comparateur analogique peut être utilisé pour déclencher la fonction de capture d'entrée du Timer/Compteur1. De plus, le comparateur analogique peut déclencher une interruption.

L'utilisateur peut choisir le front montant ou descendant pour déclencher la sortie ACO. L'entrée négative du comparateur peut être l'une des 8 entrées analogiques de l'ADC. Il est possible de choisir chacune des broches du convertisseur analogique/numérique ADC7:0 pour remplacer l'entrée négative AIN1 du comparateur analogique. Le multiplexeur de l'ADC est employé pour choisir cette entrée et par conséquent l'ADC doit être éteint pour utiliser cette possibilité (ADEN dans ADCSRA est à 0).

Le choix entre l'entrée AIN1 et l'entrée multiplexé de l'ADC est déterminé par le bit ADEM du registre SFIOR.



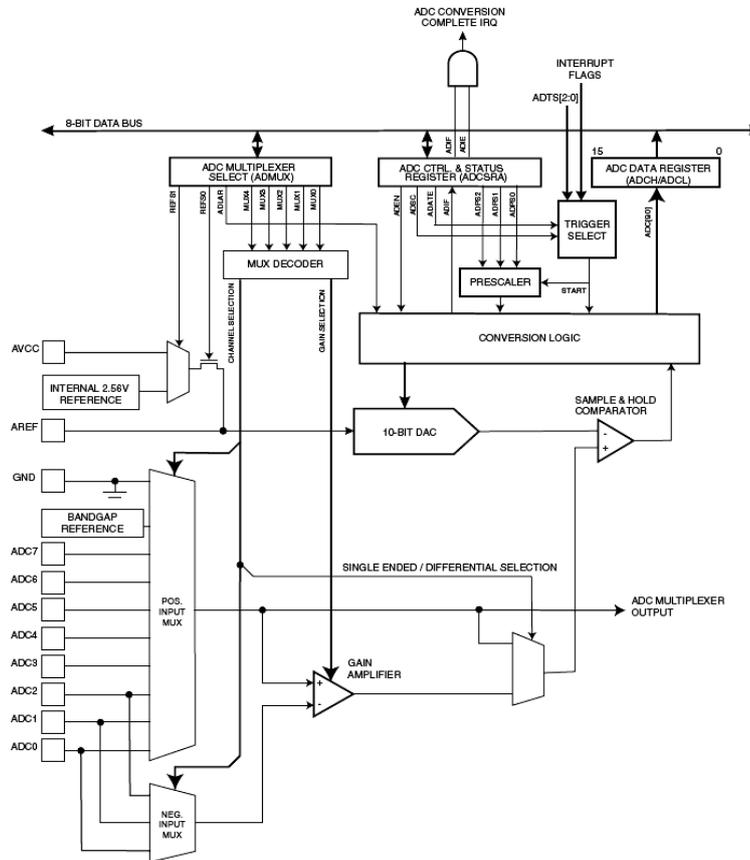
**Fig.7 Synoptique du comparateur analogique.**

### III.17. Convertisseur analogique numérique ADC

L'ADC intégré dans les microcontrôleurs est du type "Approximations successives". Il est accessible via les broches du port A. Il est doté de caractéristique très intéressante avec :

- ✓ Une résolution sur 10 bits,
- ✓ Huit entrées multiplexées,
- ✓ Sept Canaux d'entrées différentiels,
- ✓ Deux Canaux d'entrées différentiels avec un gain Optionnelle de 10x et 200x ;
- ✓ Une non linéarité inférieur à  $\frac{1}{2}$  LSB,
- ✓ Une erreur à 0 V inférieur à 1 LSB,
- ✓ Le temps de conversion est réglable de 13 à 260  $\mu$ s (plus le temps est long, plus le résultat est précis).
- ✓ Une tension de référence sélec table (peut être externe 0 à Aréf où peut être interne avec la tension de référence de 2,56 V,
- ✓ La possibilité de limiter le bruit en mode sommeil,
- ✓ L'ADC à une interruption sur conversion complète,
- ✓ Ajustement optionnel gauche pour le résultat extrait de l'ADC,
- ✓ Gamme de Tension d'entrée de l'ADC est de 0-Vcc ,
- ✓ L'Auto Déclenchement du Début de La Conversion de l'ADC sur Sources d'Interruption,
- ✓ Une résolution maximum Jusqu'à 15kSPS (15000 échantillons/second),

### III.17.1 Synoptique de l'ADC



**Fig.8 convertisseur analogique numérique.**

Les registres utilisés par l'ADC sont aux nombres de quatre.

- Registre ADMUX (ADC Multiplexer Selection Register)  
Sélection la voie de conversion (sur le port A, configuration des broches en entrées sans résistance de rappel).
- ADLAR ADC Left Adjust Result Ajustement à gauche à 1 ou à droite à 0 du résultat dans le registre ADCL et ADCH.
- Registre ADCSRA (ADC Control and Status Register A)  
Le registre de contrôle et statut de l'ADC.
- Registre ADCH et ADCL (ADC Data Register)  
Registres de résultats de la conversion analogique / digital.
- Registre SFIOR (Special Function I/O Register)

Le registre spécial SFIOR permet de modifier le déclenchement du convertisseur A/D (ADTSx).

Pour éviter qu'un éventuel blocage du programme qui est non prévue par le programmeur qui peut être d'ordre logiciel (retour impossible, mis en boucle infinie ou tout simplement erreur de structuration), soit matériel (parasites, chute de tensions) ; dans les deux cas, le blocage du programme peut avoir des conséquences très embêtantes, on utilise un chien de garde ou le watchdog qui est un compteur utilisé dans les systèmes à microprocesseur comme sécurité.

Avec notre microcontrôleur, le contrôleur Ethernet et quelques composants passifs sont tout ce qu'il faut pour relier un microcontrôleur à un réseau Ethernet.

### **III.18. Présentation général de contrôleur Ethernet ENC28J60 [5]**

L'ENC28J60 est un contrôleur Ethernet autonome conçu par la firme Micochip pour servir le réseau Ethernet avec une interface SPI (interface de périphérique série) il est industriellement compatible avec la norme IEEE802.3. Il est incorporé d'un certain nombre de filtre de paquet de tel sort à limiter et contrôlé les paquets entrants. Il est également doté d'un module interne DMA (Direct Memory Access) un accès direct à la mémoire pour le flux de données rapide, qui est employé dans divers protocoles de réseaux, en l'absence de ce dernier, le système est presque bloqué pendant les transferts de données. La communication avec le contrôleur hôte est mise en application par l'intermédiaire d'une pine d'interruption et du SPI, avec des fréquences de base allant jusqu'à 20 mégahertz. Deux pins sont liés aux LED pour indiquer l'activité du réseau.

L'ENC28J60 se compose de sept principaux blocs fonctionnels :

- ✓ Une interface SPI qui sert de canal de communication entre le contrôleur hôte et ENC28J60.
- ✓ Registres de contrôles d'instructions qui sont employés pour commander et surveillez l'ENC28J60.
- ✓ Un double accès de RAM pour réception et Emissions de paquets de données.
- ✓ Un arbitre pour le contrôler l'accès à la RAM lorsque la DMA le demande.
- ✓ Le bus de l'interface SPI s'occupe de l'interprétation des données et la réception des commandes reçues.
- ✓ Le module du MAC (Medium Access Controller) est conforme à la norme IEEE 802.3.
- ✓ L'interface du module PHY (couche physique) qui code et décode les données analogiques ce présentes sous forme de paire torsadée.

- ✓ Le dispositif contient également d'autres blocs de soutien, tels que l'oscillateur, le régulateur de tension.....

### III.18.1 Présentation physique

L'ENC28J60 se présente sous la forme d'un circuit intégré à 28 broches pour le modèle en boîtier SPDIP/SSOP/SOIC ou le boîtier QFN à 28 broches.

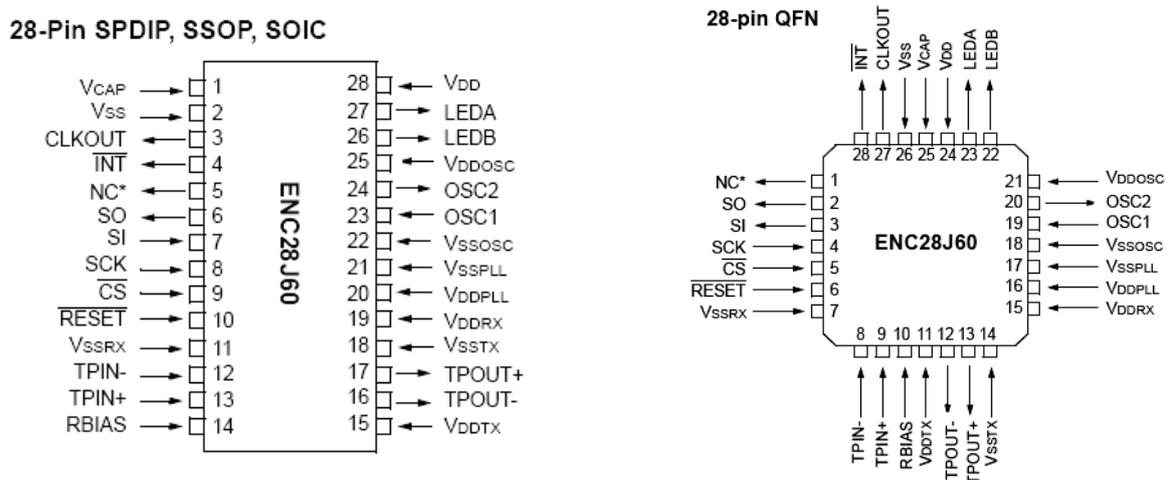


Fig.9 Brochage typique de l'ATMEGA32J60 en boîtier SPDIP et QFN.

L'architecture de l'ENC28J60 est la suivante :

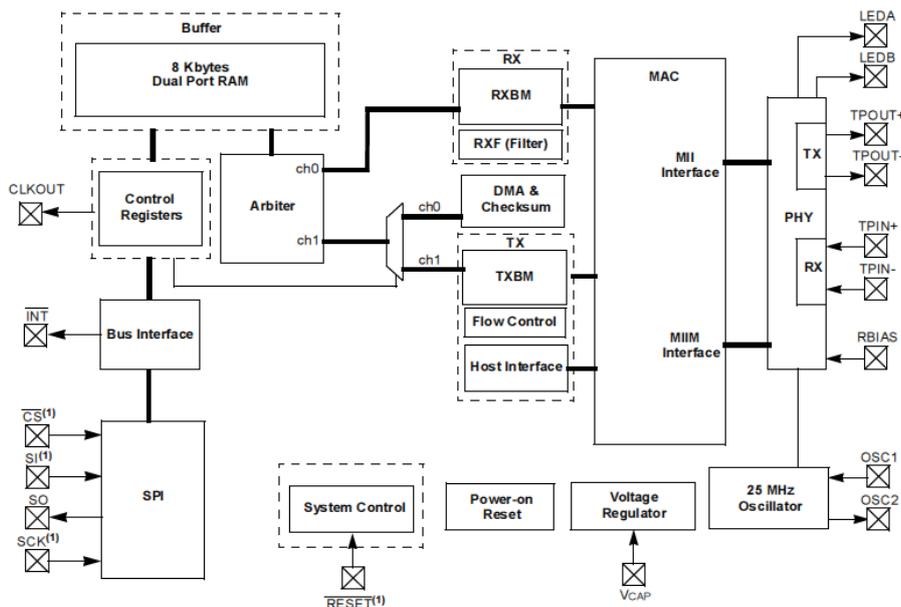


Fig.10 architecture interne de l'ENC28J60

## III.18.2 CONNEXIONS EXTERNES

### III.18.2.1 Oscillateur

L'ENC28J60 est conçu pour fonctionner à une fréquence de 25 MHz avec un cristal connecté aux broches OSC1 et OSC2. Comme elle peut être également piloté par une source d'horloge externe connectée à la broche OSC1.

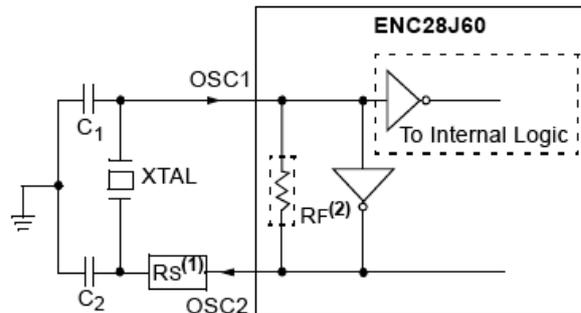


Fig.11 Montage du Quartz.

### III.18.2.2 Oscillateur Start-up Timer

L'ENC28J60 contient un oscillateur Start-up Timer (OST) afin de s'assurer la stabilité de l'oscillateur avant leur utilisation. L'OST n'expire qu'après le passage de 7500 cycles d'horloge (300 ms) après la mise sous tension ou la réinitialisation.

### III.18.2.3 Horloge externe (CLKOUT)

La broche de l'horloge externe est fournie pour le concepteur du système pour les utiliser comme l'horloge du contrôleur hôte ou comme une source d'horloge pour les autres appareils dans le système. Le CLKOUT a un pré-diviseur interne qui peut diviser la sortie en 1, 2, 3, 4 ou 8. La fonction CLKOUT est activée et le pré-diviseur est sélectionné via le registre ECOCON (EXTERNAL CLOCK OUTPUT CONTROL).

### III.18.2.4 LED

Lors de la connections des deux Leds, l'ENC28J60 va détecter leur présence est commencer à les conduire par défaut à l'aide du registre PHLCON (PHY MODULE LED CONTROL REGISTER). Si la polarité de la LED est modifiée alors que l'ENC28J60 est en fonctionnement, la nouvelle polarité ne sera pas détectée jusqu'à ce que la réinitialisation du système se produit.

### III.19. Mémoire

On trouve trois types de mémoire dans l'ENC28J60 qui sont des RAM statique :

- ✓ Registre de contrôle,
- ✓ Ethernet Buffer,
- ✓ Registre physique,

La mémoire du registre de contrôle contient des registres utiles pour la configuration et contrôle de l'ENC28J60, ou le registre de contrôle est lu directement par interface SPI. Le registre de contrôle est l'interface principale entre le contrôleur hôte et l'Ethernet contrôleur. La mémoire du registre de contrôle est divisée en quatre banques, Chaque banque est d'une longueur de 32 octets sélectionnables par les bits de sélections, BSEL1: BSEL0 (Bank Select), dans le registre ECON (ETHERNET CONTROL).

Ethernet buffer contient un espace mémoire pour la transmission et la réception utilisé par le contrôleur Ethernet. Les tailles des zones mémoire est programmables par le contrôleur hôte via l'interface SPI. La mémoire de l'Ethernet buffer est uniquement accessible par la commande SPI soit en lecture ou écriture dans la mémoire buffer.

Les registres PHY sont utilisés pour la configuration, le contrôle et l'état du module PHY. Les registres ne sont pas directement accessibles via l'interface SPI, ils ne sont accessibles que par le biais de Media Management Interface indépendants (MIIM) mis en œuvre dans le MAC.

### III.20. Interface SPI

L'ENC28J60 est conçu pour communiqué directement avec l'interface SPI (Serial Peripheral interface). Une liaison SPI est un bus de donnée série synchrone baptisé ainsi par Motorola, et qui opère en Full Duplex. Les circuits communiquent selon un schéma maître-esclaves, où le maître s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un bus, la sélection du destinataire se fait par une ligne dédiée entre le maître et l'esclave appelée « chip select ».

Le bus SPI contient 4 signaux logiques

- SCLK : (slave clock) Horloge généré par le maître
- MOSI : (Master Output, Slave Input), généré par le maître
- MISO : (Master Input, Slave Output), généré par l'esclave
- SS : (Slave Select), Actif à l'état bas, généré par le maître

### **III.20.1 Fonctionnement**

Une transmission SPI typique est une communication simultanée entre un maître et un esclave.

- Le maître génère l'horloge et sélectionne l'esclave avec qui il veut communiquer
- L'esclave répond aux requêtes du maître

A chaque coup d'horloge le maître et l'esclave s'échangent un bit. Après huit coups d'horloges le maître a transmis un octet à l'esclave et vice-versa. La vitesse de l'horloge est réglée selon des caractéristiques propres aux périphériques.

### **III.20 Conclusion**

Dans ce présent chapitre dédié au microcontrôleur ATMEGA23 et au contrôleur Ethernet, nous avons essayé de rassembler et d'évoqué les informations nécessaires pour leurs bon fonctionnement on donnant une description générale.

L'offre en microcontrôleurs s'est développée d'autant plus des périphériques intégrés, capacité de calcul accrue, plus de mémoire embarquée. Il y a maintenant de nombreux fabricants de microcontrôleurs, possédant chacun plusieurs gamme, véritable couteau suisse électronique.

# Chapitre IV

Dans ce chapitre nous allons présenter les différentes parties pour la mise en œuvre de notre carte d'acquisition de données.

## IV.1 Représentation synoptique

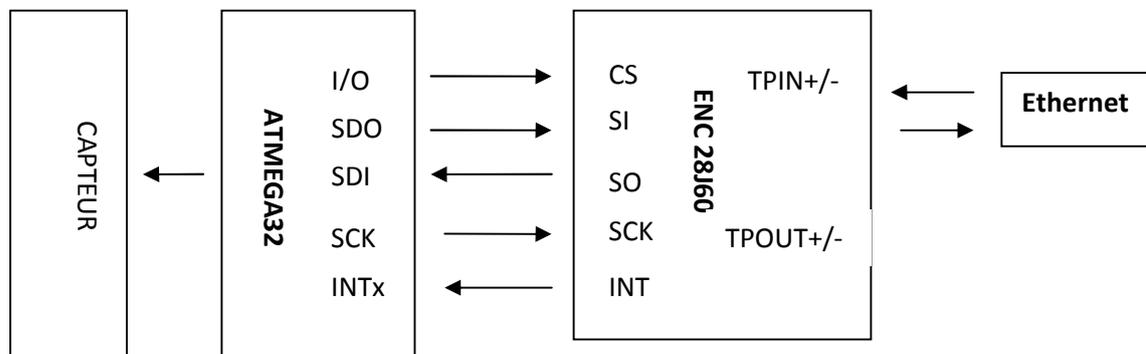
Le schéma synoptique montre en gros les différentes parties essentielles constituant notre carte d'acquisition de données ou cette dernière est divisés en deux étage.

On trouve dans le premier étage:

- ✓ Circuit d'alimentation.
- ✓ Un microcontrôleur ATMEGA32.
- ✓ Un bloc pour l'emplacement du capteur.

Dans le second étage :

- ✓ Un contrôleur Ethernet ENC28J60.
- ✓ Un transformateur Ethernet.
- ✓ Un connecteur RJ45.



**Fig. 1 Représentation synoptique global**

## IV.2 Description des différents blocs

### IV.2.1 L'alimentation

Pour tout circuit ou montage électrique, il faut une alimentation pour son fonctionnement. La tension d'alimentation du microcontrôleur ATMEGA32 est de 2,7 à 5,5 V et celle de l'ENC28J60 est de 3,1 à 3,6V.

Notre circuit d'alimentation se constitue :

- ✓ D'un connecteur.
- ✓ D'une diode en série pour la protection du montage.
- ✓ D'une led d'indication.
- ✓ De deux régulateurs LM 317 et 7805 pour fournir respectivement une tension fixe et stable de +3V et +5V.
- ✓ De condensateurs 100 nF regroupés en parallèle deux avant et après le régulateur pour effectuer un filtrage et réduire les parasites hautes fréquences qui pourraient nuire le microcontrôleur.

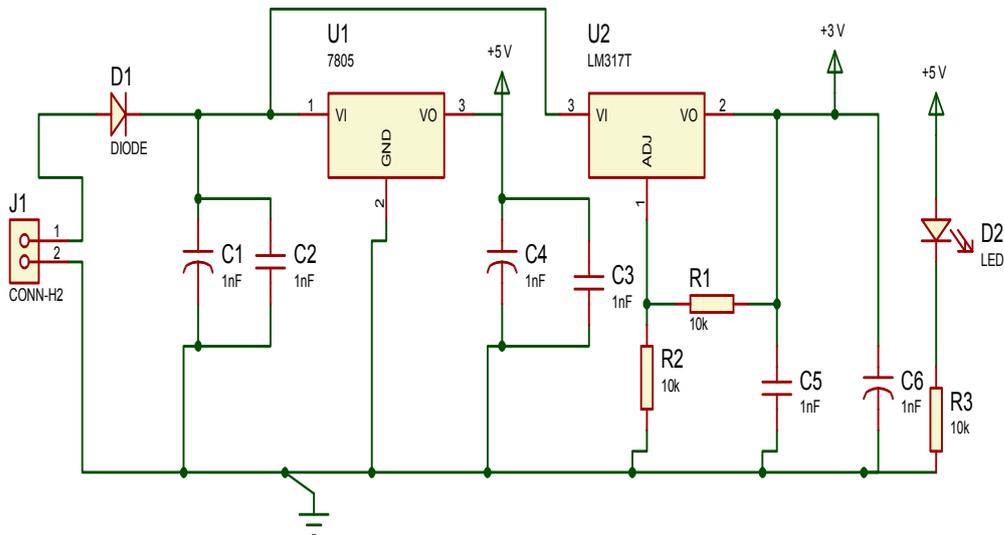


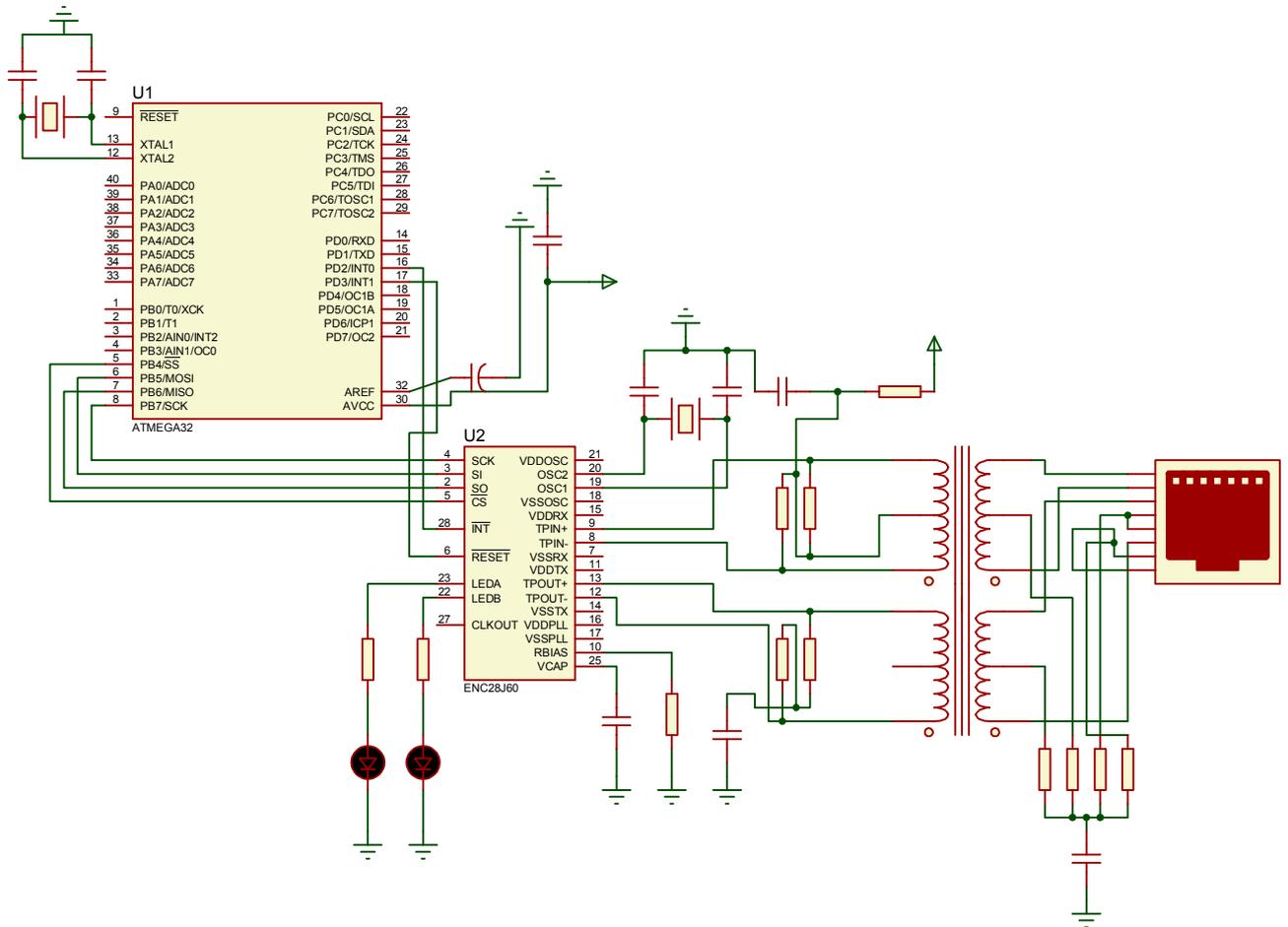
Fig.2 circuit d'alimentation

## **IV.2.2 Le microcontrôleur ATMEGA32**

Le microcontrôleur est la pièce maîtresse de notre carte car il s'occupe du contrôle et du traitement des données. Les signaux issus du capteur sont transmis au microcontrôleur pin (PA0), ce dernier est relié, pin (PB4, PB5, PB6, PB7) à l'ENC28J60 qui est là pour transmettre les données issues des capteurs vers un éventuel réseau. D'une manière générale si ma carte est connectée à un réseau internet par exemple, donc je peux accéder directement aux données du capteur.

## **IV.2.3 Le microcontrôleur ENC28J60**

L'ENC28J60 est un contrôleur Ethernet associé à 2 LEDs d'état qui sont connectées aux pins (23,22) et à un connecteur RJ45 isolé par l'intermédiaire d'un transformateur d'impulsion connecté aux pins (13,12). La communication avec le contrôleur hôte se fait via l'interface SPI connectée aux pins (4, 3, 2,5) de l'ENC sans oublier de connecter les pins d'interruption et le RESET. Il est conçu pour fonctionner avec une fréquence de 25 MHz fournie par un quartz externe connecté aux pins OSC1 et OSC2 (pin 19 et 20) comme on peut aussi connecter OSC1 à une source d'horloge externe. Une capacité de filtrage 10  $\mu$ F connectée à la pin 29 VCAP vers la masse.



**Fig.3 Montage de l'ENC28J60**

La plus part des applications a base de microcontrôleur communiquent avec le milieu extérieur.

### IV.2.3 Capteurs LM35

Le capteur est un maillon de la chaine d'acquisition il est censés nous délivre les informations du milieu extérieur, les signaux délivrés par le capteur sont généralement de nature électrique.

Notre choix c'est porter sur un capteur de température qui est le LM35 qui sera l'exemple de notre application pour les éventuelles essais du fonctionnement de notre carte.

Ce capteur est destiné à prélever de la température du milieu extérieur, la série LM35 sont a base de circuit intégré formant un capteur de température de précision, dont la tension de sortie est linéairement proportionnelle au Celsius de la température (centigrade), de plus l'avantage qu'il a sur les capteurs de température linéaire calibrée en ° Kelvin, c'est que

L'utilisateur n'est pas obligé de soustraire une tension constante de ses grandes de sorties pour obtenir l'échelle centigrade pratique, puis il n'a besoin aucune calibration externe. Il peut être utilisé avec une alimentation simple, ne consomme que 60  $\mu\text{A}$  de son alimentation, il a un très faible auto-échauffement, moins de 0,1  $^{\circ}\text{C}$  en air ambiant. Le LM35 est prévu pour fonctionner sur une plage de température qui varie de  $^{\circ}\text{-55}$  à  $^{\circ}\text{+150}$   $^{\circ}\text{C}$ .

### IV. 2.3.1 Choix du capteur

Les capteurs présentent deux parties distinctes, une première partie qui a pour rôle de détecter un événement, une seconde partie qui a pour rôle de traduire l'évènement en un signal compréhensible.

Pour choisir un capteur il faudra définir tout d'abord :

- le type et la nature de l'évènement à détecter ;
- la grandeur de l'évènement ;
- l'environnement de l'évènement ;

D'autres éléments peuvent permettre de cibler précisément le capteur à utiliser :

- ses performances ;
- son encombrement ;
- sa fiabilité ;
- la nature du signal délivré par le capteur et son prix ;

### IV. 2.3.2 Propriétés du LM35

- Calibré directement en  $^{\circ}\text{Celsius}$  (centigrade) ;
- Facteur d'échelle linéaire  $+ 10,0 \text{ mV} / ^{\circ}\text{C}$  ;
- Précision de 0.5  $^{\circ}\text{C}$  à  $+ 25^{\circ}\text{C}$  ;
- Plage de fonctionnement ( $T_{\text{min}}$ ,  $T_{\text{max}}$ )  $-55^{\circ}$  à  $+150^{\circ}\text{C}$  ;
- Convient pour des applications à distance ;
- Tension de fonctionnement 4 à 30 volts ;
- Consommation du courant moins de 60  $\mu\text{A}$ .



Le microcontrôleur ATMEGA32 possède un convertisseur analogique/numérique intégré au sein du microcontrôleur qui s'en charge de convertir les signaux analogiques issu du capteur à des données numériques.

### IV.2.2.3 Conversion et contrôle de l'ADC

Notre capteur étant connecté au convertisseur qui se dernier est chargé de convertir tension analogique issu du capteur en résultat numérique (digital) codé sur 10 bits par approximation successive ou la valeur minimale est représenté par **GND** et la valeur maximale est la tension sur la broche **AREF**, nous pouvons écrire l'équation suivante :

Résultat numérique = (Tension d'entrée / tension de référence **AREF**) x 1024 - 1.

Chacune des 8 broches d'entrées de l'ADC ainsi que **GND** et la référence de tension, peuvent être choisie comme des entrées simples de l'ADC, comme elles peuvent être choisies comme des entrées positives et négatives de l'amplificateur de gain différentiel. Si des canaux différentiels sont choisis, l'étage de gain différentiel amplifie la tension différence par le facteur de gain choisi. Cette valeur amplifiée devient alors le résultat de l'entrée analogique de l'ADC. Si des canaux simples sont employés l'amplificateur de gain est contourné.

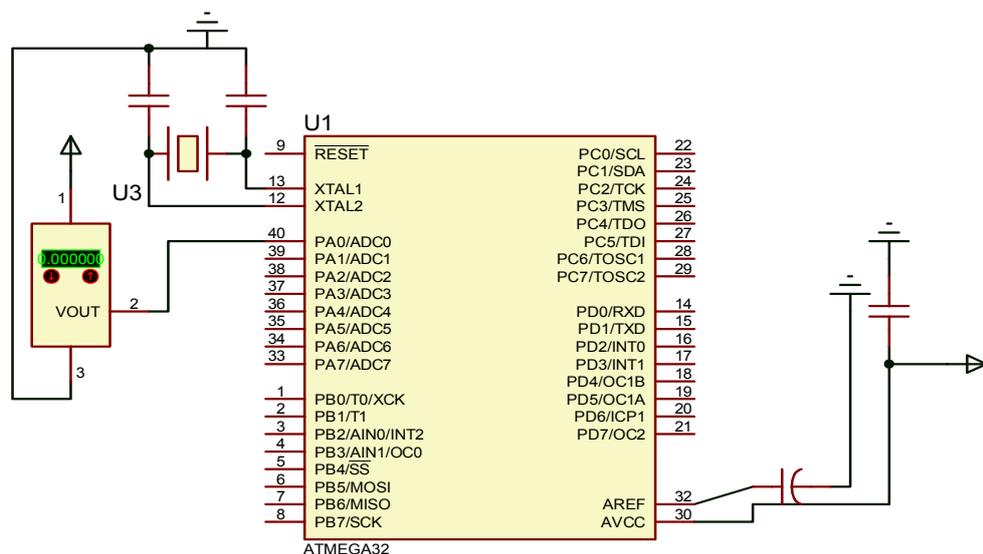


Fig.5 Montage du capteur

L'ADC est actif lorsque le bit **ADEN** est mis à 1 dans le registre **ADCSRA**. La référence de tension et le choix du canal d'entrée n'entrera pas en vigueur quand **ADEN** est mis à 1, il faut d'abord désactiver **ADEN**. L'ADC ne consomme pas de puissance quand **ADEN** est à zéro, donc on recommande d'éteindre l'ADC avant l'entrée en mode sommeil.

L'ADC produit un résultat sur 10 bits qui est présenté dans les registres **ADCH** et **ADCL**. Par défaut, le résultat est présenté ajusté à droite, mais peut facultativement être présenté ajusté à gauche en mettant le bit **ADLAR** dans **ADMUX**. Pour un résultat sur 8 bits et un ajustement à gauche, la lecture du registre **ADCH** est suffisante. Autrement, **ADCL** doit être lu en premier puis **ADCH** pour assurer la cohérence des données qui appartiennent à la même conversion. Une fois **ADCL** lu, l'accès aux registres de commandes est bloqué afin d'empêcher une nouvelle conversion tant que **ADCH** n'est pas lu. Quand **ADCH** est lu, l'ADC peut à nouveau être opérationnelle.

L'ADC a sa propre interruption qui peut être déclenché quand une conversion est achevée. Quand l'accès de l'ADC aux registres de commandes est interdit par la lecture d'**ADCL** et d'**ADCH**, l'interruption est quant même déclenché et le résultat sera perdu, il faut donc faire attention à lire rapidement le résultat d'une conversion.

Une simple conversion commence en mettant le bit **ADSC** à 1 dans **ADCSRA** puis en attendant le front montant suivant du cycle d'horloge **ADC**. Le bit est maintenu en état haut tant que la conversion se réalise et sera remis à 0 quand la conversion est achevée. Si un canal de données différent est choisi tandis qu'une conversion est en cours, l'ADC finira la conversion actuelle avant l'exécution du changement de canal.

Une conversion normale prend 13 cycles d'horloge. La première conversion après l'allumage de l'ADC prend 25 cycles d'horloge pour initialiser le circuit analogique.

Une conversion peut être déclenchée automatiquement par des sources diverses. Le déclenchement automatique se fait en programmant le bit **ADATE** dans le registre **ADCSRA**. La source de déclenchement est choisie en mettant les bits de déclenchement **ADTS** dans **SFIOR**.

Par défaut, le circuit d'approximation successif exige une fréquence d'horloge d'entrée entre 50 **kHz** et 200 **kHz** pour obtenir la résolution maximale. Si une résolution inférieure à 10 bits est nécessaire, la fréquence d'horloge d'entrée de l'ADC peut être supérieure à 200 **kHz** pour augmenter le nombre de conversion à la seconde.

Le module **ADC** contient un pré diviseur qui produit une fréquence d'horloge acceptable pour l'ADC. La mise en service du pré diviseur est obtenu par le bit **ADPS** dans **ADCSRA**.

### IV.2.3.4 Paramétrage du convertisseur simple

- ✓ Mise en marche du convertisseur **ADEN = 1.**
- ✓ Sélection du facteur de pré division **ADPS2...ADPS0.**
- ✓ Sélection de l'entrée à convertir **MUX4...MUX0.**
- ✓ Enclenchement d'une acquisition **ADSC = 1.**
- ✓ Attente fin de conversion **ADSC = 0.**
- ✓ Lecture du résultat sur **ADCH** et **ADCL.**

## IV.3. Les outils de programmation et simulation du microcontrôleur

### IV.3.1 L'environnement de développement AVR studio

L'ATMEGA32 est un composant programmable, il doit être programmé avant de l'utiliser dans un circuit électronique, la programmation consiste à programmer en assembleur ou en compilateur ou en un autre langage évolué qui sera traduit en un langage machine que l'en injecte dans la mémoire du microcontrôleur.

L'AVR studio est un logiciel fourni par la firme ATMEL utilisable sous Windows. Couplé à Win AVR, on dispose d'un environnement de développement puissant intégré pour debugger les programmes AVR, il permet la simulation de puces et de circuits d'émulation pour la famille de microcontrôleurs AVR.

L'interface utilisateur est spécialement conçue pour être facile à utiliser et à donner des informations complètes, l'AVR utilise la même interface utilisateur à la fois pour la simulation et l'émulation.

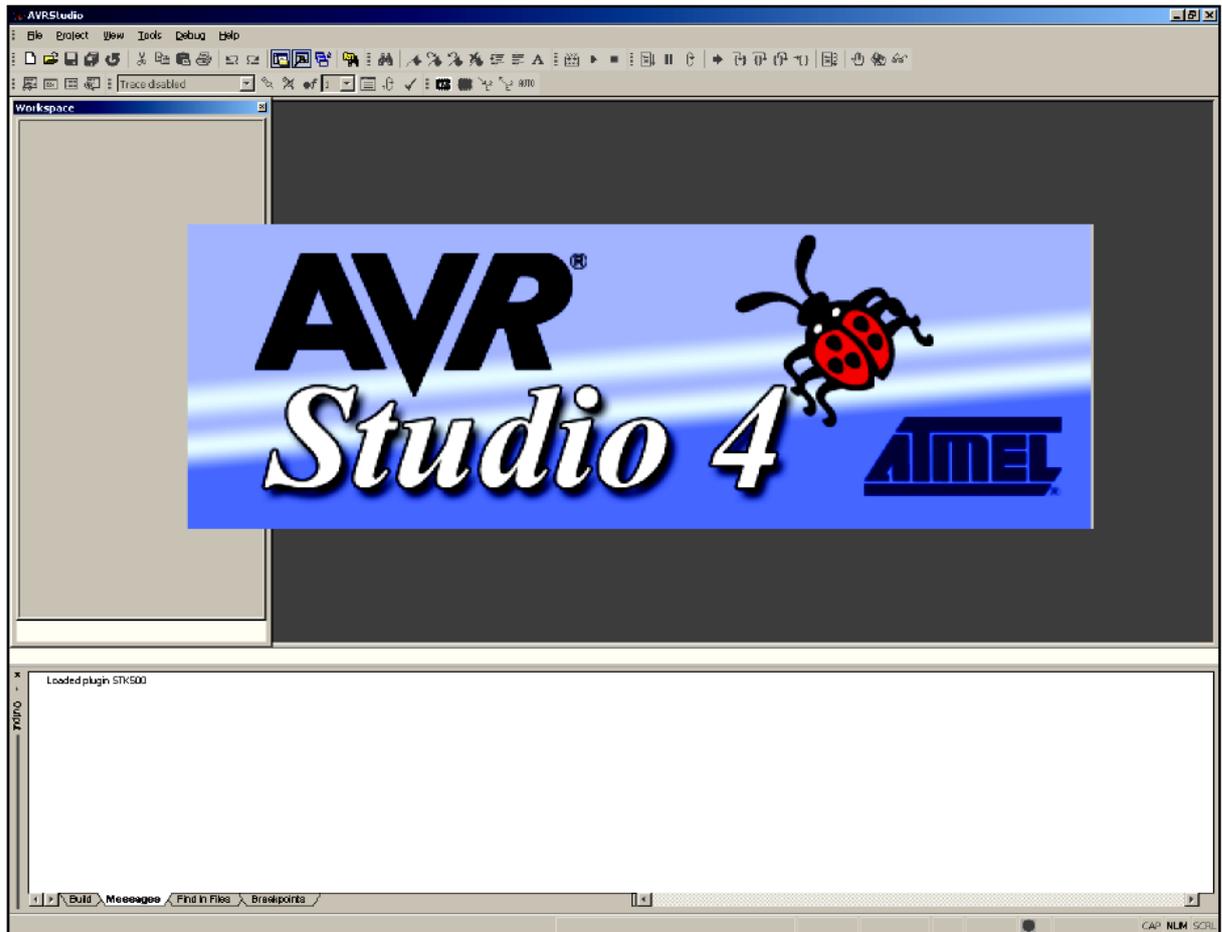


Fig.1 AVR studio4

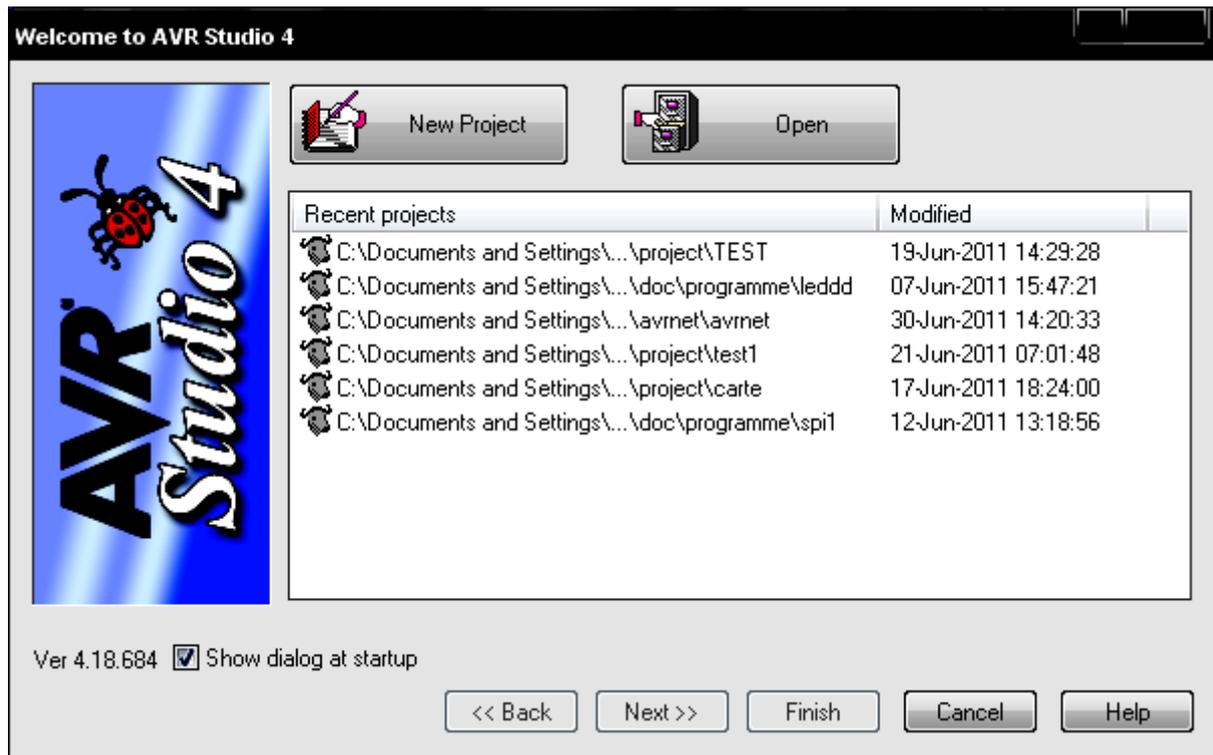
L'AVR studio utilise le C comme langage de programmation qui est un langage de haut niveau mais qui permet d'effectuer facilement des opérations de bas niveau et d'accéder aux fonctionnalités du système, de plus c'est le langage le plus répandu dans le domaine des systèmes embarqués. Il convient parfaitement au développement des pilotes de périphériques spécialisés tels que les convertisseurs analogiques/numériques, les capteurs, etc. Les outils de développement modernes possèdent des fonctions d'optimisation performantes et le recours au langage assembleur est devenu inutile.

Le cycle de développement se résumerait aux 3 étapes suivantes :

- ✓ Écrire le code source,
- ✓ le compiler,
- ✓ lancer le programme.

## IV-3.2 Création d'un projet

Lorsque en click sur l'icône de l'AVR studio, une petite fenêtre apparie nous donnons un choix, soit exécuté un programme ou créer un nouveau projet.



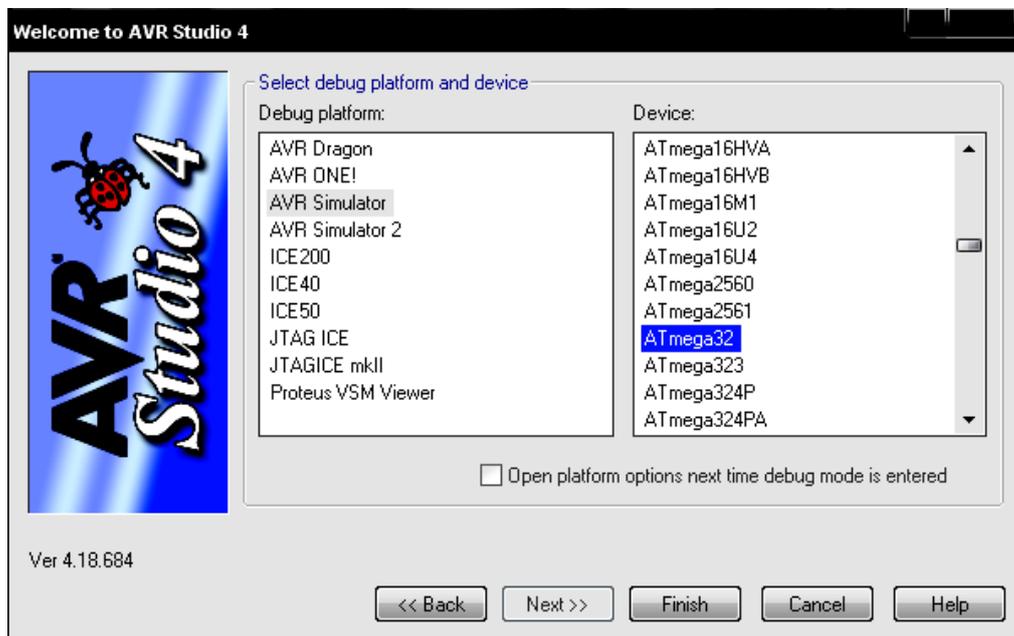
**Fig.2 création ou ouverture d'un fichier**

Pour créer un nouveau projet il suffi de click sur New Project, ensuite une autre fenêtre apparie afin de choisir de programmer soit en assembleur ou en compilateur, donnée un nom au projet et choisir son emplacement.



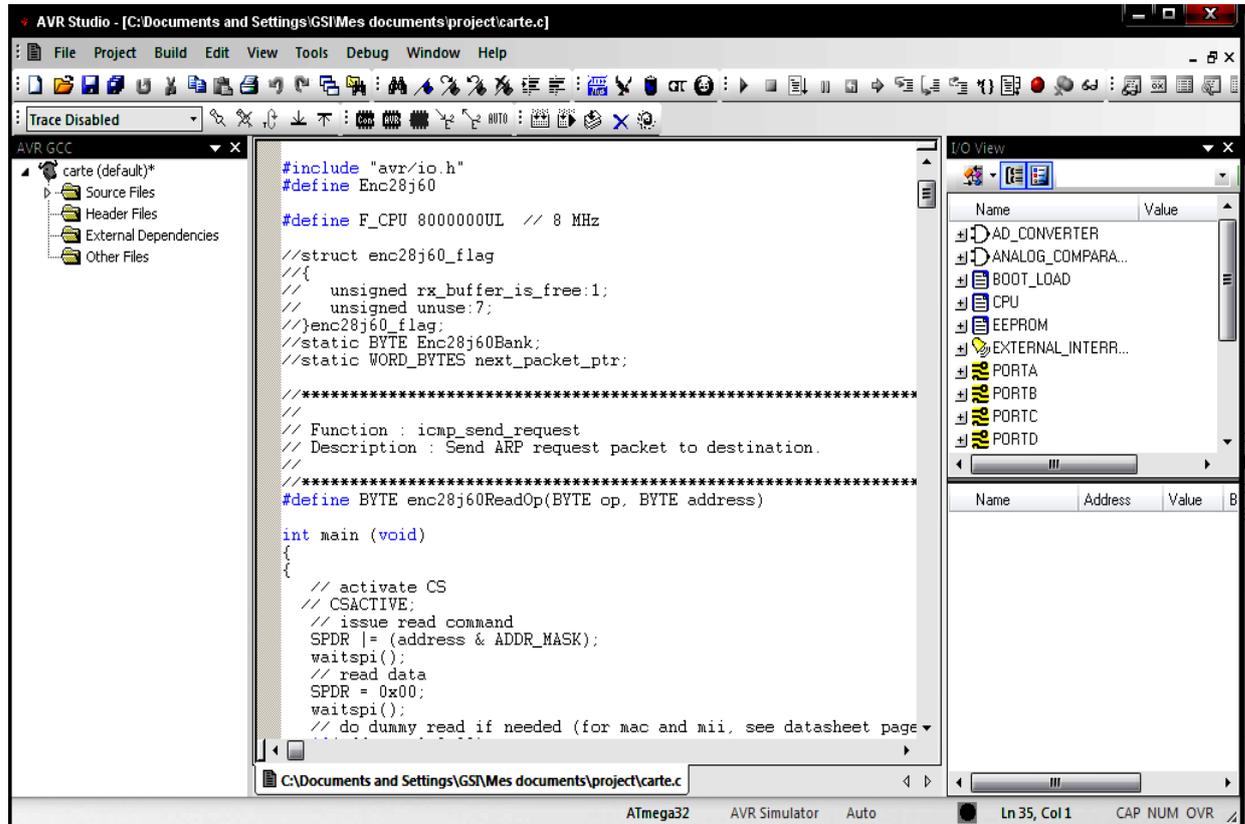
**Fig.3 configuration**

Ensuite vient l'étape pour le choix du microcontrôleur qu'en desire travailler et c'est l'étape la plus essentielle, pour notre cas c'est l'ATMEGA32.



**Fig.4 choix du microcontrôleur**

Après toutes ces étapes d'initialisations apparie l'éditeur de programme ou va être saisie notre code source puis effectuer de réelle simulation et voir le déroulement du programme.



**Fig.5 Editeur Windows**

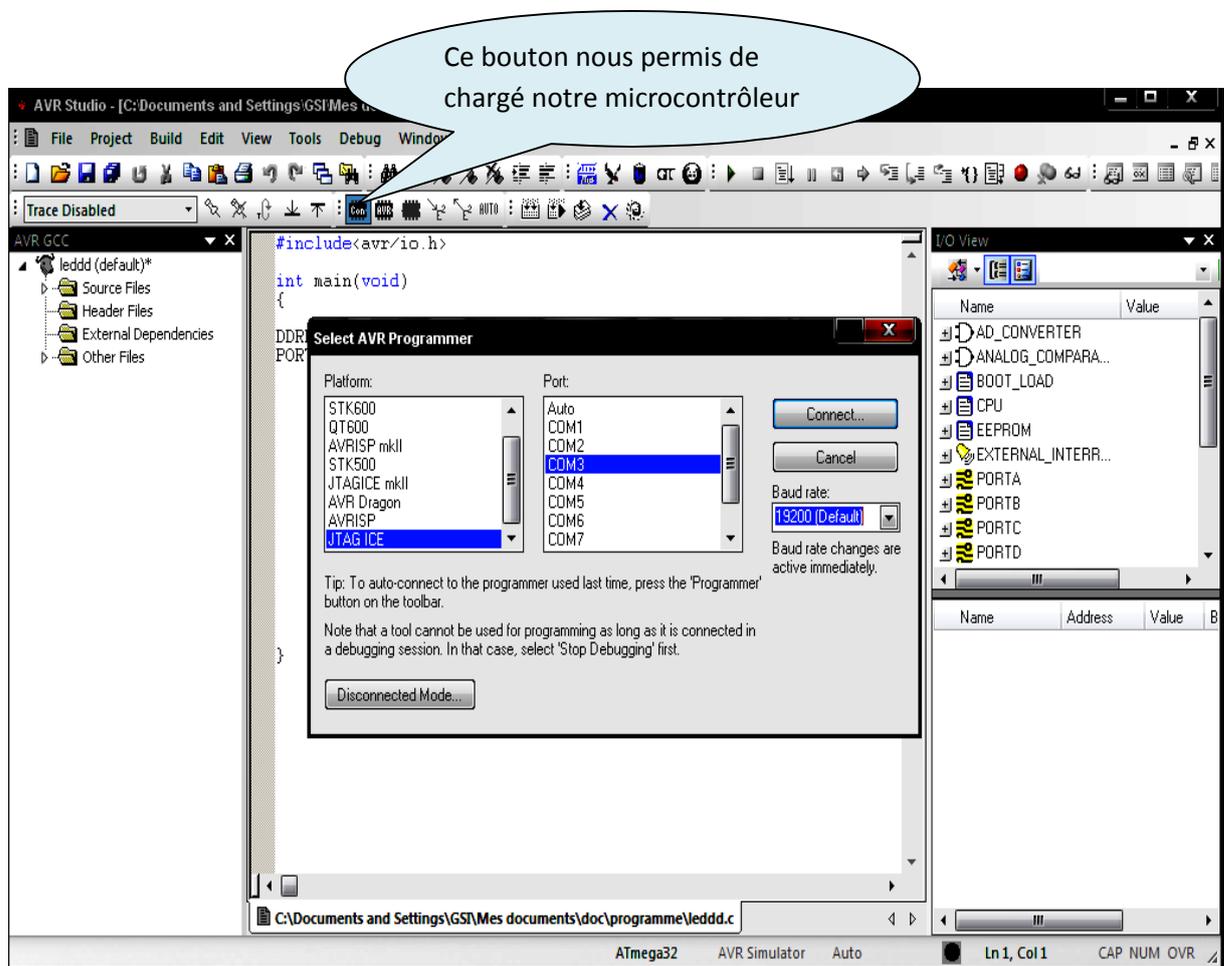
Afin d'identifier facilement et rapidement les erreurs fonctionnelles, il peut être intéressant d'utiliser un débogueur qui permet par exemple, d'exécuter pas à pas l'application, de poser des points d'arrêts ou d'observer en temps réel l'évolution du contenu d'une variable ou d'une zone mémoire.

Avant de télécharger le programme dans la mémoire FLASH du microcontrôleur, il est nécessaire d'effectuer un certain nombre d'opérations :

- ✓ Création et configuration d'un projet
- ✓ Création d'un fichier .c contenant le code source du programme.
- ✓ Ajout de fichiers sources au projet.
- ✓ Compilation du projet et correction éventuelle des erreurs de syntaxe jusqu'à avoir un compte rendu du compilateur sans erreur.
- ✓ Transfert du fichier .hex qui est généré automatiquement contenant le code exécutable dans la mémoire FLASH sur le microcontrôleur.
- ✓ Exécution et débogage du programme.

### IV.3.3 Transfert du programme vers le microcontrôleur

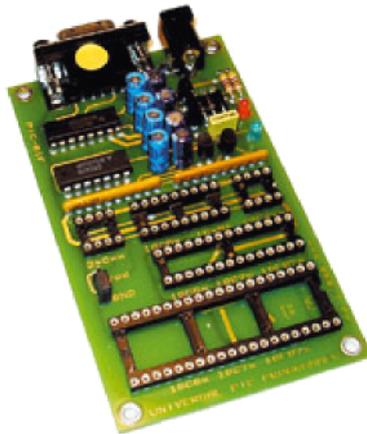
Après avoir testés les programme par simulation, nous pouvons transférer le programme vers le microcontrôleur pour procéder aux tests et aux essais pratiques. Le programmeur de transfert qui se charge du transfert est inclus dans le logiciel AVR studio, le transfert ce fait à l'aide d'un port série RS232. On appuie sur le « Display the Connect Dialog » il apparie un terminal a fin de dialoguer sur la liaison série avec la carte cible pour choisir le numéro et le type du port ainsi la vitesse de transmission en baud rate.



**Fig.6 Interface du chargement de programme dans la mémoire du microcontrôleur ATMEGA32**

Lorsque le transfert du programme dans l'ATMEGA est effectué, il y a une procédure de vérification. Si un message d'erreur apparaît, il peut s'agir d'une mauvaise connexion du ou d'une mauvaise alimentation du programmeur.

Le but de cette étape est d'obtenir un microcontrôleur ayant un programme est prêt à l'emploi, sans oublier de tester son fonctionnement dans le montage auquel il est destiné. Une fois qu'on est sûr du transfert, mettre le programmeur hors tension, puis sortir délicatement le l'ATMEGA.



**Fig. 7 Exemple d'un programmeur**

Mais avant de passer à la pratique c'est-à-dire la réalisation de la carte, il convient de faire des essais virtuelles pour mieux optimiser notre conception et voir d'éventuelle modification côté soft et hard (programme, emplacement des composants ...) s'il est nécessaire, donc on dispose d'un logiciel Proteus ISIS qui nous permet de concevoir et simuler virtuellement notre dispositif électronique.

## **IV.4 Simulation**

### **IV.4.1 Présentation de Proteus ISIS7**

Proteus est un logiciel de simulation de montage en logique combinatoire, séquentiel, en électronique analogique et création de circuits imprimés.

ISIS est au cœur du système Proteus, il est bien plus qu'un simple paquet de schémas. Il combine un puissant environnement de conception avec la possibilité de définir la plupart des aspects de l'apparence de dessin, il fonctionne sous Windows xp,

une apparence d'un fichier Word, il dispose d'une bibliothèque de composants qui nous permet de créer et d'étudier des circuits électroniques, logiques de toute sorte, ISIS est l'outil pour le travail.

### IV.4.2 Création d'un projet

Dans cette partie, pour la création d'un projet dans ISIS, On va suivre la présente méthodologie présentée ci-dessus :

1. Lancer ISIS ;
2. Aller à la bibliothèque, choisir les composants nécessaires ;
3. Disposer les composants à leurs places puis les relier ;
4. Importer les programmes nécessaires aux fonctionnements de certains composants ;
5. Lancer la simulation en appuyant sur le bouton Play ;

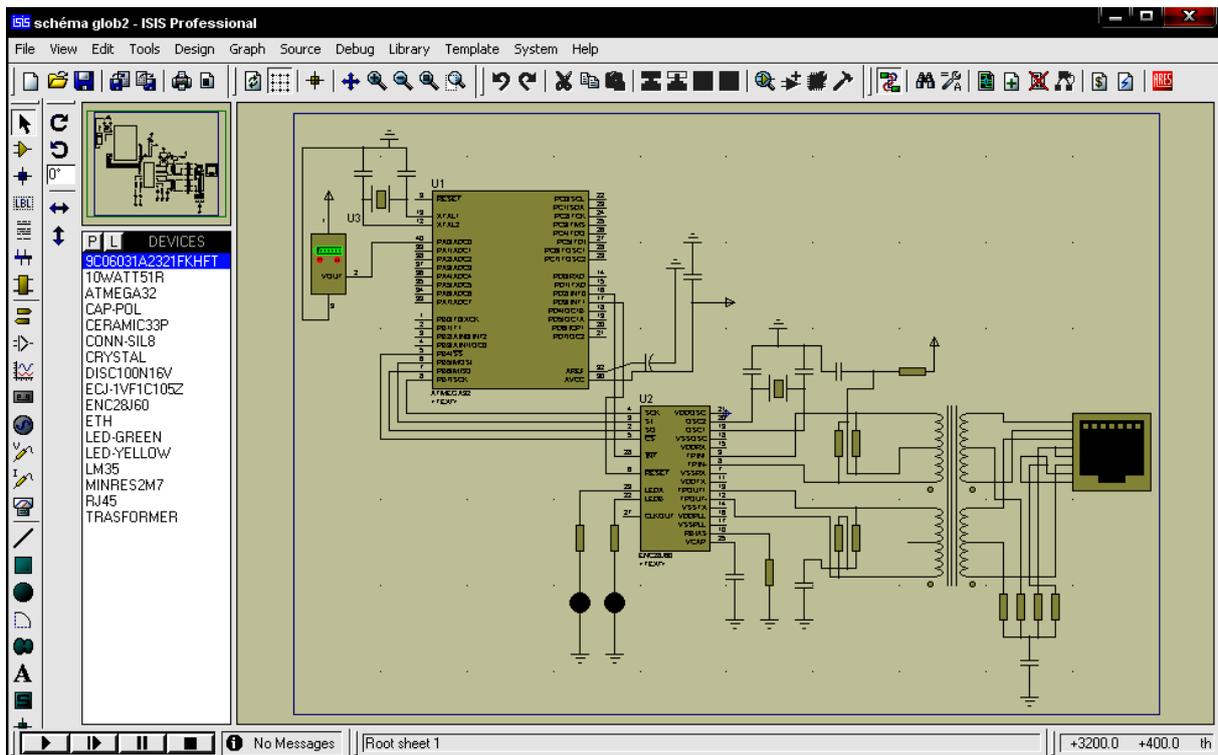


Fig.8 Interface du logiciel Proteus ISIS

## IV.5 Protocole de transmission

A fin que notre carte puisse être connectée a un réseau et transmettre des données, il est nécessaire d'établir un protocole de transmission. Ce protocole devra être le même afin que la transmission fonction correctement.

Un protocole de communication permet aux machines de dialoguer. Le protocole que utilisée notre carte est le protocole TCP/IP (Transmission Control Protocol) et (Internet Protocol), ce sont deux suites de protocoles. C'est ce protocole qui permet la gestion des mécanismes des paquets d'information et leur transition du réseau à l'application.

Donc :

Le protocole IP assure le transport des paquets de données numériques.

Le protocole TCP assure le contrôle de la transmission des données.

TCP est un protocole de transport, orienté connexion, qui fournit un flux d'octets fiable assurant l'arrivée des données sans altérations et dans l'ordre, avec retransmission en cas de perte, et élimination des données dupliquées tout en veillent aux respects des règles de communications sur internet et se base sur l'adresse IP. Les machines sont identifiées sur le réseau par une adresse IP unique sur quatre octets.

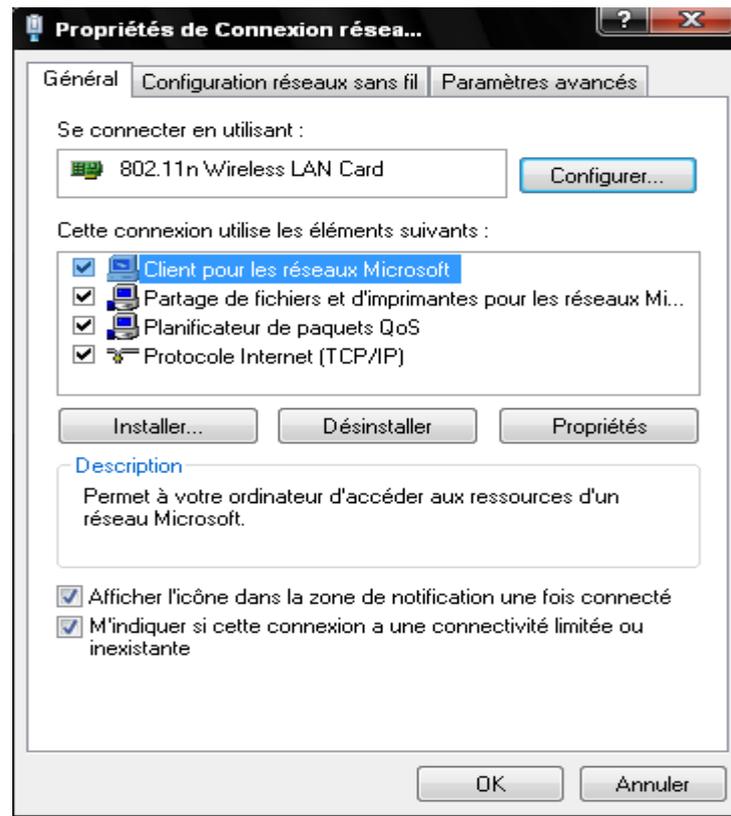
Exemple : **255.255.255.0**

L'adresse IP peut être attribuée de façon fixe à chaque équipement ou attribuée de façon dynamique au moment de la connexion au réseau par un serveur DHCP (Dynamique Host Configuration Protocol). La machine demande une adresse sur le réseau à sa mise sous tension et le serveur DHCP présent lui en attribue une.

### IV.5.1 Paramètre de configuration

Pour la configuration il faut choisir :

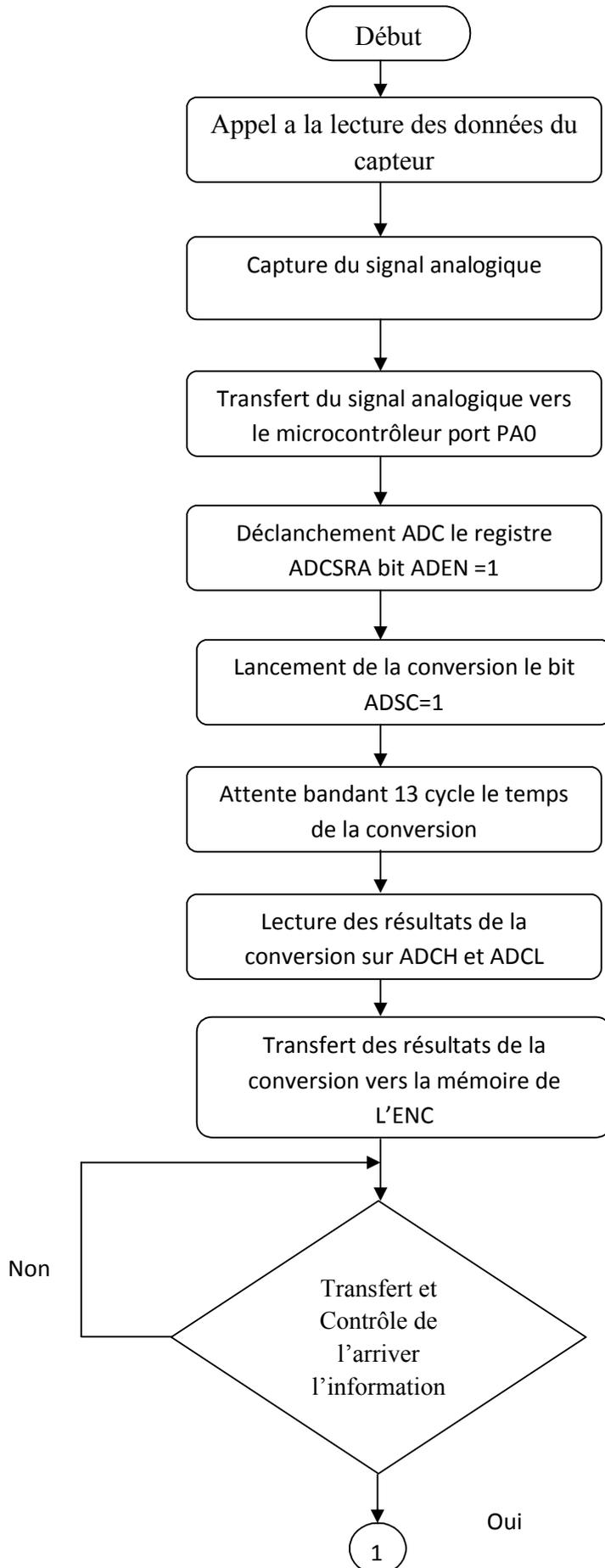
- Une carte réseau (carte d'acquisition)
- Un protocole de communication (TCP/IP)

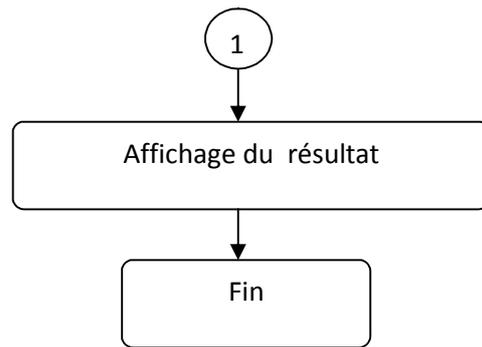


### Configuration du protocole TCP/IP

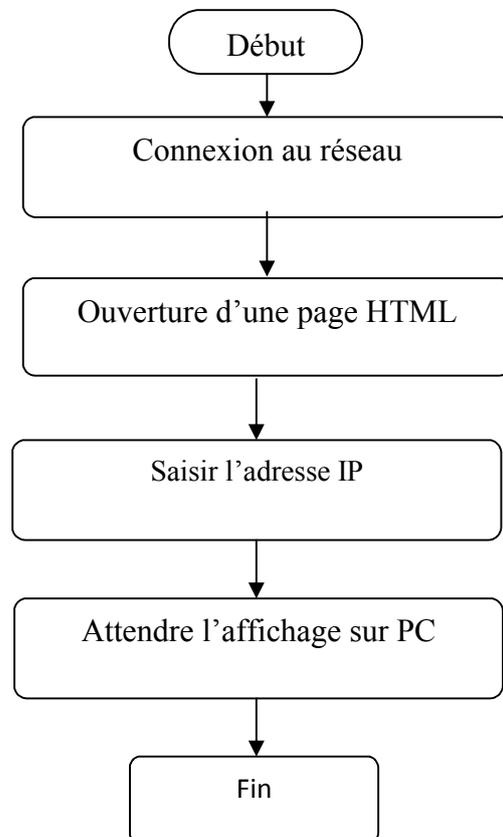


## IV.6 Organigramme fonctionnel

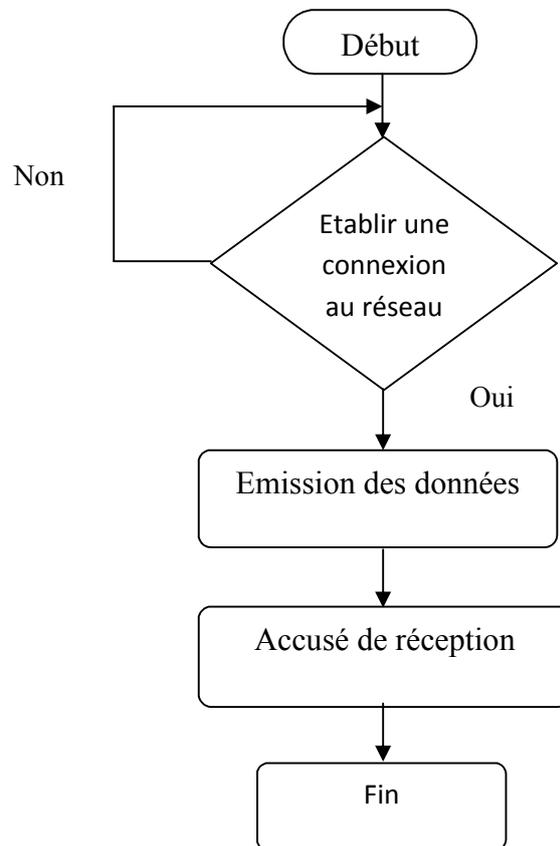




### IV.6.1 Appel a la lecture des données du capteur



## IV.6.2 Transfert et Contrôle de l'arrivée l'information



## **Schéma globale**

## IV.8 Circuit imprimé

Le circuit imprimé de notre carte est réalisé à l'aide du logiciel(ARES).

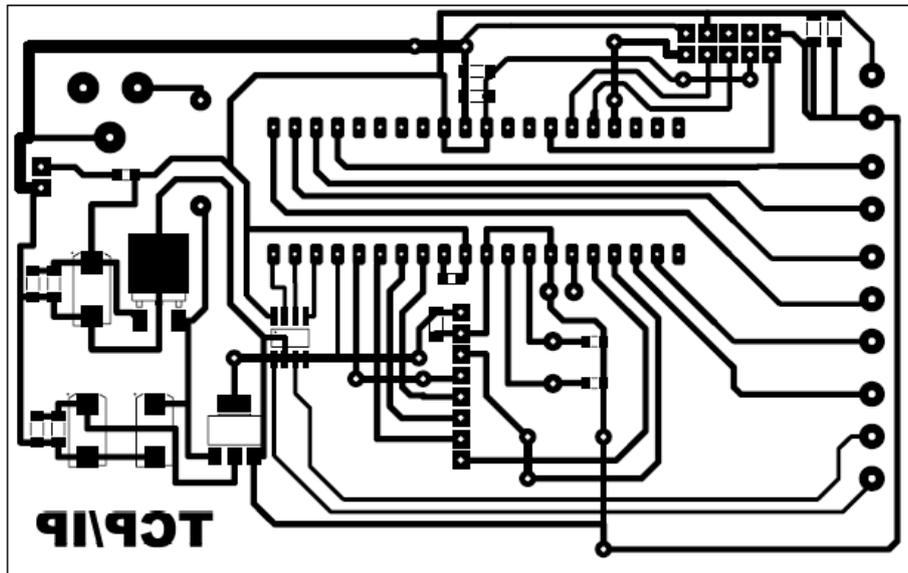


Fig.9 Circuit imprimé du TCP/IP

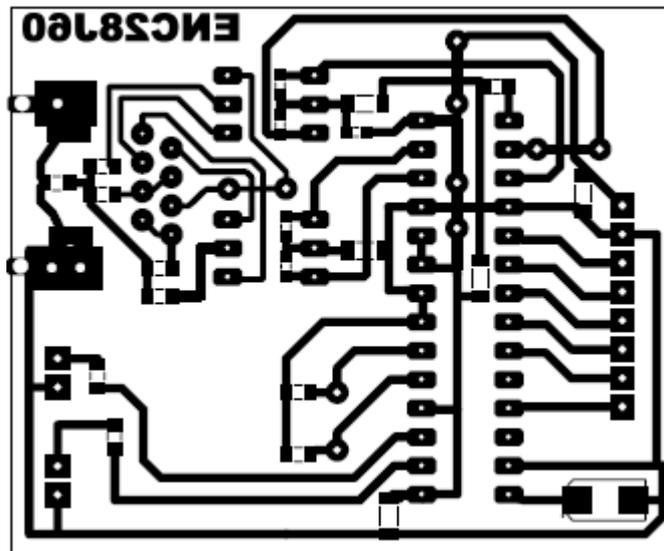


Fig.9 Circuit imprimé de l'ENC28J60 et l'aimantation

## **IV-9 Conclusion**

Dans ce chapitre nous avons cité l'utilité et le fonctionnement des deux logiciels qui sont essentiels à l'élaboration de notre carte d'acquisition. De plus dans ce chapitre nous a permis de ce familiariser avec outils de développement même aussi nous avons appris la manière comment programmé un microcontrôleur ainsi quelque aspect technique de conception et de réalisation.





**Conclusion  
générale**

### Conclusion générale

L'objectif de notre travail est la réalisation d'une carte d'acquisition de données à l'aide des composants électroniques programmable a base du microcontrôleur ATMEGA 32 et du contrôleur ENC28J60 communiquant via le protocole TCP/ IP.

Dans un premier temps, il était intéressant de présenter les bases fondamentales des protocoles de communication avant d'aborder la réalisation de notre carte. A cet effet notre carte peut être accessible depuis le réseau internet ainsi on peut accéder aux informations délivré par nos capteur de import tous.

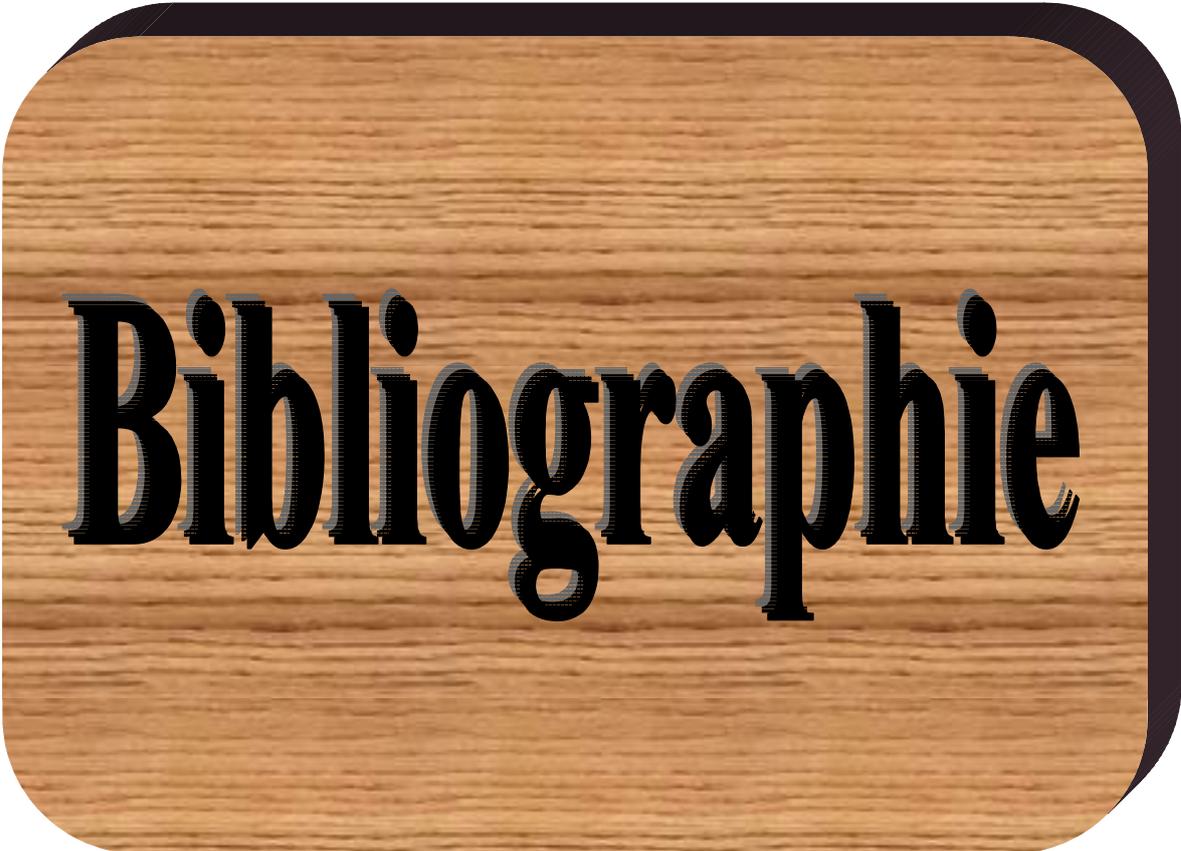
Dans le but d'exploiter quelques avantages d'une famille de microcontrôleur, notre choix a été porté sur un AVR (ATMEGA 32). En ce qui concerne le contrôleur Ethernet ENC28J60 c'est l'outil qui nous permet de ce connecté à un éventuel réseau. Le capteur utilisé est le LM35 qui est un capteur de température fonctionnant sur une plage de température qui varie de ° -55 à +150 ° C.

Une phase assez importante avant de passé à une réalisation est celle du développement du circuit. Pour ce, un ensemble d'outils a été exploiter pour atteindre notre objectif.

Après avoir réalisé notre carte, des testes de simulation virtuelle ont été effectués afin de vérifier le fonctionnement de notre carte.

En perspectives,

- ✓ On peut étendre la carte sur d'autres types de capteurs.
- ✓ Ajouter un afficheur pour une lecture directe.
- ✓ Exploiter d'autres types de protocole de communication (sans fils).



# Bibliographie

## BIBLIOGRAPHIE

[1] **Mr FANCIS COTTET**, « Traitement des signaux et acquisition de donnée » édition DUNOD. novembre 2005

[2] **Mr GEORGE SEBEK**, « Transmission de données », article **E7810**, technique d'ingénieur.

[3] **Mr Jacques TICHON** « Communication avec les périphériques », article **S8590** technique d'ingénieur.

[4] **Mr GUY PUJOLLE** « Les réseaux », édition **EYROLLES**. septembre 2006.

[5] **Document technique** ENC28J60 (stand-alone Ethernet controller with SPI), [www.microchip.com](http://www.microchip.com)

[6] **Christian Tavernier**, « Les Microcontrôleurs PIC, Description et mise en œuvre », 2<sup>ème</sup> Editions : Dunod, Paris, 2002.

[7] **Document technique** ATMEGA32 ( Microcontroller with 32Kbytes In-System Programmable Flash) [www.microchip.com](http://www.microchip.com)



# **Annexes**

## Annexe A

### Le tableau des composants

|                         |   |
|-------------------------|---|
| U1                      | Microcontrôleur ATMEGA 32               |
| U2                      | Ethernet contrôleur ENC28J60            |
| U3                      | Capteur de température LM35             |
| J1                      | connecteur d'alimentation               |
| U4                      | 7805 (régulateur 5 volts)               |
| U5                      | LM317T (régulateur 3 volts)             |
| R1,R2                   | 180 ohm                                 |
| R3                      | 2.32K ohm 1%                            |
| R5,R6,R7,R8             | 49.9k ohm 1%                            |
| R9,R10,R11,R12          | 75 ohm                                  |
| R13, R14                | 10K ohm                                 |
| C1,C2,C5,C6             | 33 pF                                   |
| C9                      | 10 $\mu$ F                              |
| C10,C11,C12,C13,C14,C15 | 10 nF                                   |
| C3,C7,C8                | 100 nF                                  |
| C4                      | 47 $\mu$ F                              |
| 1 :1 CT                 | Transformateur d'impulsion Rapport 1 :1 |
| RJ 45                   | Connecteur Ethernet                     |
| X1                      | Quartz 16 MHz                           |
| X2                      | Quartz 25 MHz                           |
| Led A                   | diode électroluminescente rouge 3mm     |
| Led B                   | diode électroluminescente jaune 3mm     |
| D2                      | LED lumineuse                           |

---

## Annexe B

### Quelques définitions

- **Nœud** : Equipement (ordinateur, concentrateur, imprimante etc.) du réseau informatique possédant une adresse physique.
- **Nœud TCT/IP** : Equipement fonctionnant avec le protocole TCT/IP.
- **Adresse physique** : adresse unique attribuée par le fabricant, gérée par un organisme international et inscrite de façon définitive dans l'équipement. Appelée aussi *adresse Ethernet* ou *adresse MAC* (Media Access Control).
- **Adresse logique** : Adresse unique attribuée par les répondants informatiques locaux et gérée par le NIC pour les adresses IP (organisme international).
- **Paquet, trame (*frame*)** : Information électrique transitant sur le réseau et représentant une suite de bits. Il s'agit de l'unité de base du réseau.
- **Protocole** : Description des mécanismes permettant la gestion des paquets d'information et leur transition du réseau à l'application. Par extension, logiciel (software) fonctionnant sur une machine et permettant cette **gestion interne**.
- **Suite de protocoles** : Ensemble de protocoles communiquant entre eux.
- **Couches réseau** : Composante protocole fonctionnant sur un équipement et lui permettant de communiquer à travers le réseau.
- **Media** : Support permettant le passage de l'information : différents types de câbles, ondes hertziennes, laser etc.
- **Bande passante**: Comme en téléphonie ou en hi-fi, la bande passante détermine la quantité d'informations capables de transiter par seconde sur un media donné.
- **Backbone** : Réseau physique (câble et matériel actif) servant de colonne vertébrale au reste du réseau. En général composée de fibres optiques est assurant parfois une plus grande bande passante (155Mbits /s).
- **Interconnexion** : Ensemble matériel et logiciel permettant la connexion d'équipements utilisant des média et des protocoles différents.
- **Segment physique** : Câble ou ensemble de câbles reliés sans équipement électronique d'interconnexion (il s'agit d'un seul domaine de collision).
- **Segment logique** : Sous-réseau regroupant des nœuds correspondant à un même groupe d'adressage ; un segment logique peut englober plusieurs segments physiques.



---

## Annexe C

### Quelques programmes

#### Configuration IP

```
//*****
//*****
//
// File: ip.h implement for Internet Protocol
//
//*****
//*****
//
// Copyright (C) 2007
//
// This program is free software; you can redistribute it and/or modify it
// under
// the terms of the GNU General Public License as published by the Free
// Software
// Foundation; either version 2 of the License, or (at your option) any
// later
// version.
// This program is distributed in the hope that it will be useful, but
//
// WITHOUT ANY WARRANTY;
//
// without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
// PARTICULAR
// PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
// with
// this program; if not, write to the Free Software Foundation, Inc., 51
// Franklin St, Fifth Floor, Boston, MA 02110, USA
//
// http://www.gnu.de/gpl-ger.html
//
//*****
//*****
#define IP_HEADER_LEN          20

#define IP_PROTO_ICMP_V        0x01
#define IP_PROTO_TCP_V         0x06
#define IP_PROTO_UDP_V         0x11
#define IP_V4_V                 0x40
#define IP_HEADER_LENGTH_V     0x05

#define IP_P                     0x0E
#define IP_HEADER_VER_LEN_P    0x0E
#define IP_TOS_P                 0x0F
#define IP_TOTLEN_H_P           0x10
#define IP_TOTLEN_L_P           0x11
#define IP_ID_H_P                0x12
#define IP_ID_L_P                0x13
#define IP_FLAGS_H_P            0x14
```

```

#define IP_FLAGS_L_P          0x15
#define IP_TTL_P              0x16
#define IP_PROTO_P            0x17
#define IP_CHECKSUM_H_P       0x18
#define IP_CHECKSUM_L_P       0x19
#define IP_SRC_IP_P           0x1A
#define IP_DST_IP_P           0x1E

/*****
**/
//
// Prototype function
//
/*****
**/
//void ip_fill_ip_address( unsigned char *buf, unsigned char *avr_ip,
unsigned char *dest_ip );
//void ip_fill_hdr_checksum( unsigned char *buf );

extern BYTE ip_packet_is_ip ( BYTE *rxtx_buffer );
extern void ip_generate_header ( BYTE *rxtx_buffer, WORD_BYTES
total_length, BYTE protocol, BYTE *dest_ip );

```

## Configuration TCP

```

/*****
**/
//
// File : tcp.h implement for Transmission Control Protocol
//
/*****
**/
//
// Copyright (C) 2007
//
// This program is free software; you can redistribute it and/or modify it
under
// the terms of the GNU General Public License as published by the Free
Software
// Foundation; either version 2 of the License, or (at your option) any
later
// version.
// This program is distributed in the hope that it will be useful, but
//
// WITHOUT ANY WARRANTY;
//
// without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR
// PURPOSE. See the GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License along
with
// this program; if not, write to the Free Software Foundation, Inc., 51
// Franklin St, Fifth Floor, Boston, MA 02110, USA
//
// http://www.gnu.de/gpl-ger.html

```

```
//
//*****
*****
#define TCP_HEADER_LEN      20
#define TCP_OPTION_LEN     4

#define TCP_FLAG_FIN_V      0x01
#define TCP_FLAG_SYN_V      0x02
#define TCP_FLAG_RST_V      0x04
#define TCP_FLAG_PSH_V      0x08
#define TCP_FLAG_ACK_V      0x10
#define TCP_FLAG_URG_V      0x20
#define TCP_FLAG_ECE_V      0x40
#define TCP_FLAG_CWR_V      0x80

#define TCP_SRC_PORT_H_P    0x22
#define TCP_SRC_PORT_L_P    0x23
#define TCP_DST_PORT_H_P    0x24
#define TCP_DST_PORT_L_P    0x25
#define TCP_SEQ_P           0x26 // the tcp seq number is 4 bytes
0x26-0x29
#define TCP_SEQACK_P        0x2A // 4 bytes
#define TCP_HEADER_LEN_P    0x2E
#define TCP_FLAGS_P         0x2F
#define TCP_WINDOWSIZE_H_P  0x30 // 2 bytes
#define TCP_WINDOWSIZE_L_P  0x31
#define TCP_CHECKSUM_H_P    0x32
#define TCP_CHECKSUM_L_P    0x33
#define TCP_URGENT_PTR_H_P  0x34 // 2 bytes
#define TCP_URGENT_PTR_L_P  0x35
#define TCP_OPTIONS_P       0x36
#define TCP_DATA_P          0x36

//*****
*****
//
// Prototype function
//
//*****
*****
extern WORD tcp_get_dlength ( BYTE *rxtx_buffer );
extern BYTE tcp_get_hlength ( BYTE *rxtx_buffer );
extern WORD tcp_puts_data ( BYTE *rxtx_buffer, BYTE *data, WORD offset );
extern WORD tcp_puts_data_p ( BYTE *rxtx_buffer, PGM_P data, WORD offset );

extern void tcp_send_packet (
    BYTE *rxtx_buffer,
    WORD_BYTES dest_port,
    WORD_BYTES src_port,
    BYTE flags,
    BYTE max_segment_size,
    BYTE clear_seqack,
    WORD next_ack_num,
    WORD dlength,
    BYTE *dest_mac,
    BYTE *dest_ip );
```

## Annexe D

Le tableau suivant donne la liste des 66 registres existant dans la gamme ATMEGA :

| Catégorie                     | Désignation                             | Registre                    | Adresse |
|-------------------------------|---|-----------------------------|---------|
| Interface deux fils I2C (TWI) | Contrôle du Bus                         | <a href="#">TWBR</a>        | \$00    |
|                               | Status                                  | <a href="#">TWSR</a>        | \$01    |
|                               | Adresse                                 | <a href="#">TWAR</a>        | \$02    |
|                               | Données à transmettre ou reçus          | <a href="#">TWDR</a>        | \$03    |
| Convertisseur analogique      | Poids Faible Résultat                   | <a href="#">ADCL</a>        | \$04    |
|                               | Poids fort Résultat                     | <a href="#">ADCH</a>        | \$05    |
|                               | Contrôle et statut                      | <a href="#">ADCSR</a>       | \$06    |
|                               | Sélection de la voie à échantillonner   | <a href="#">ADMUX</a>       | \$07    |
| Comparateur analogique        | Contrôle et statut du Comparateur       | <a href="#">ACSR</a>        | \$08    |
| USART                         | Vitesse de communications (Bauds)       | <a href="#">UBRR</a>        | \$09    |
|                               | Contrôle                                | <a href="#">UCR</a>         | \$0A    |
|                               | Status                                  | <a href="#">USR</a>         | \$0B    |
|                               | I/O donnée                              | <a href="#">UDR</a>         | \$0C    |
| SPI                           | Contrôle                                | <a href="#">SPCR</a>        | \$0D    |
|                               | Status                                  | <a href="#">SPSR</a>        | \$0E    |
|                               | I/O donnée                              | <a href="#">SPDR</a>        | \$0F    |
| Port parallèle D              | Adresse des broches (Pin)               | <a href="#">PIND</a>        | \$10    |
|                               | Direction des broches                   | <a href="#">DDR D</a>       | \$11    |
|                               | Données des broches                     | <a href="#">PORT D</a>      | \$12    |
| Port parallèle C              | Adresse des broches (Pin)               | <a href="#">PINC</a>        | \$13    |
|                               | Direction des broches                   | <a href="#">DDRC</a>        | \$14    |
|                               | Données des broches                     | <a href="#">PORTC</a>       | \$15    |
| Port parallèle B              | Adresse des broches (Pin)               | <a href="#">PINB</a>        | \$16    |
|                               | Direction des broches                   | <a href="#">DDRB</a>        | \$17    |
|                               | Données des broches                     | <a href="#">PORTB</a>       | \$18    |
| Port parallèle A              | Adresse des broches (Pin)               | <a href="#">PINA</a>        | \$19    |
|                               | Direction des broches                   | <a href="#">DDRA</a>        | \$1A    |
|                               | Données des broches                     | <a href="#">PORTA</a>       | \$1B    |
| EEPROM                        | Contrôle                                | <a href="#">EECR</a>        | \$1C    |
|                               | Donnée                                  | <a href="#">EEDR</a>        | \$1D    |
|                               | Poids faible adresse                    | <a href="#">EEARL</a>       | \$1E    |
|                               | Poids Fort adresse                      | <a href="#">EEARH</a>       | \$1F    |
| USART                         | Vitesse de communications (Bauds)       | <a href="#">UBRRH/UCSRC</a> | \$20    |
| Watchdog                      | Chien de Garde                          | <a href="#">WDTCR</a>       | \$21    |
| Timer / compteur 2 (8 bits)   | Statut du mode asynchrone               | <a href="#">ASSR</a>        | \$22    |
|                               | Comparaison                             | <a href="#">OCR2</a>        | \$23    |
|                               | Compteur                                | <a href="#">TCNT2</a>       | \$24    |
|                               | Contrôle                                | <a href="#">TCCR2</a>       | \$25    |
| Timer / compteur 1 (16 bits)  | Entrée poids faible de capture          | <a href="#">ICR1L</a>       | \$26    |
|                               | Entrée poids fort de capture            | <a href="#">ICR1H</a>       | \$27    |
|                               | Timer B sortie comparaison poids faible | <a href="#">OCR1BL</a>      | \$28    |
|                               | Timer B sortie comparaison poids fort   | <a href="#">OCR1BH</a>      | \$29    |
|                               | Timer A sortie comparaison poids faible | <a href="#">OCR1AL</a>      | \$2A    |
|                               | Timer A sortie comparaison poids fort   | <a href="#">OCR1AH</a>      | \$2B    |
|                               | Timer/compteur 1 poids Faible           | <a href="#">TCNT1L</a>      | \$2C    |
|                               | Timer/compteur 1 poids Fort             | <a href="#">TCNT1H</a>      | \$2D    |

| <b>Catégorie</b>            | <b>Désignation</b>                         | <b>Registre</b>    | <b>Adresse</b> |
|-----------------------------|--|--------------------|----------------|
| Port I/O<br>Oscillateur     | Contrôle Timer/compteur 1 poids Faible     | <b>TCCR1B</b>      | \$2E           |
|                             | Contrôle Timer/compteur 1 poids Fort       | <b>TCCR1A</b>      | \$2F           |
|                             | Registre Spéciale des fonctions <b>I/O</b> | <b>SFIOR</b>       | \$30           |
|                             | Calibration d'Oscillateur                  | <b>OSCCAL/OCDR</b> | \$31           |
| Timer / compteur 0 (8 bits) | Timer/compteur 0                           | <b>TCNT0</b>       | \$32           |
|                             | Contrôle Timer/compteur 0                  | <b>TCCR0</b>       | \$33           |
| MCU                         | Statuts Général                            | <b>MCUSR</b>       | \$34           |
|                             | Contrôle Général                           | <b>MCUCR</b>       | \$35           |
| Interface I2C (TWI)         | Contrôle Général                           | <b>TWCR</b>        | \$36           |
| MCU 'Boot Loader'           | Contrôle mémoire programme                 | <b>SPMCR</b>       | \$37           |
| Interruptions               | Drapeau d'interruption Compteur/Timer      | <b>TIFR</b>        | \$38           |
|                             | Masque d'interruption Compteur/Timer       | <b>TIMSK</b>       | \$39           |
|                             | Drapeau d'interruption Généraux            | <b>GIFR</b>        | \$3A           |
|                             | Masque d'interruption Généraux             | <b>GICR</b>        | \$3B           |
|                             | Timer/compteur comparaison en Sortie       | <b>OCR0</b>        | \$3C           |
| Pointeur de pile            | Pointeur de pile poids faible              | <b>SPL</b>         | \$3D           |
|                             | Pointeur de pile poids fort                | <b>SPH</b>         | \$3E           |
| Registre SREG               | Registre d'Etat et de Statut               | <b>SREG</b>        | \$3F           |

# IV.7 Schéma électrique globale

