

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : électronique  
Option : télécommunication

Réf: .....

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

Mise en œuvre du protocole I2C  
dans environnement à microcontrôleur  
Atmel (ATMEGA 32)

Présenté par :  
**Chennoufi Hicham**  
Soutenu le : Juin 2013

Devant le jury composé de :

Mr. A.Benakcha

MCA

Président

Mr. A.Debilou

MCA

Encadreur

Mme. M .Handaoui

MAA

Examineur

**Année universitaire : 2012 / 2013**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : électronique  
Option : télécommunication

Mémoire de Fin d'Etudes  
En vue de l'obtention du diplôme:

**MASTER**

*Thème*

Mise en œuvre du protocole I2C  
dans environnement à microcontrôleur  
Atmel (ATMEGA 32)

Présenté par :

*Chennoufi Hicham*

Avis favorable de l'encadreur :

*Debilou Abderrazak*

*signature*

**Avis favorable du Président du Jury**

*Dr. A .Benakcha*

*Signature*

**Cachet et signature**

الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي و البحث العلمي  
Ministère de l'enseignement Supérieur et de la Recherche scientifique



Université Mohamed Khider Biskra  
Faculté des Sciences et de la Technologie  
Département de Génie Electrique  
Filière : électronique  
Option : télécommunication

## *Thème :*

# Mise en œuvre du protocole I2C dans environnement à microcontrôleur Atmel (ATMEGA 32)

Proposé par : **Debilou Abderrazak**

Dirigé par : **Debilou Abderrazak**

## **RESUMES (Français et Arabe)**

Les protocoles de communication sont dévers et varient entre les appareils électroniques et se diffère selon leur utilisation.

Dans notre projet, nous avons utilisé le protocole I2C dans un environnement à microcontrôleur Atmel (ATMEGA 32) pour mesurer la température par le circuit DS1621 et afficher par un afficheur graphique LCD.

تتعدد بروتوكولات الاتصال بين الاجهزة الالكترونية وتختلف حسب استعمالاتها .  
في مشروعنا هذا استخدمنا البروتوكول I2C في بيئة المتحكم (ATMEGA 32) من اجل قياس درجة  
الحرارة باستعمال الدارة (DS1621) وعرضها في عارض بياني (LCD) .

*Au nom de dieu clément et miséricordieux*

# *Dédicace*

*Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage  
et la volonté nécessaire pour affronter les différentes difficultés de la vie,  
Je dédie ce modeste travail*

*A ma mère.*

*A mon père.*

*A mes frères et mes sœurs.*

*A toute ma famille.*

*A mes amis et mes collègues de la promotion 2013  
« Electronique ».*

*A mes amis (chaque un à son nom).*

*A tous les musulmans dans le monde entier*

*H. CHENNOUFI*

## Remerciements

*Nous rendons nos profondes gratitude à dieu qui nous a aidés à réaliser ce modeste travail.*

*Nous exprimons nos profondes gratitude à nos parents pour leurs encouragements, leurs soutiens et pour les sacrifices qu'ils ont enduré.*

*Je remercie mon encadreur Dr. Dehilou Abderrazak pour les efforts qu'ils ont déployés, pour nous aider, conseiller, encourager et corriger.*

*Nous tenons à remercier les membres de jury d'avoir accepté d'examiner notre travail.*

*Nous remercions aussi tout le corps enseignant et administratif qui a contribué à notre formation universitaire.*

*Sans oublier tous nos amis.*

*H.Chennoufi*

## *Résumé*

Les protocoles de communication sont dévers et varient entre les appareils électroniques et se diffère selon leur utilisation.

Dans notre projet, nous avons utilisé le protocole I2C dans un environnement à microcontrôleur Atmel (ATMEGA 32) pour mesurer la température par le circuit (DS162) et afficher par un afficheur graphique (LCD).

## ملخص

تتعدد بروتوكولات الاتصال بين الاجهزة الالكترونية وتختلف حسب استعمالاتها .  
في مشروعنا هذا استخدمنا البروتوكول I2C في بيئة المتحكم (ATMEGA 32) من اجل قياس درجة الحرارة باستعمال الدارة (DS1621) وعرضها في عارض بياني (LCD) .

***Liste De Tableau***

Tableau II.1 : Comparaison des technologies RISC et CISC .....18

Tableaux III.1 : Les bibliothèques du Mikro C..... 40

***Liste De Figures***

Figure I.1 : Architecture matérielle d'un système embarqué : .....3

Figure I.2 : La loi G. Moore pour les processeurs Intel : .....6

Figure I.3 : radar ACC 77Ghz : .....9

Figure I.4 : Fonctionnalité de radar ACC : .....11

Figure II.1 : Architecture de Von Neumann : .....13

Figure II.2 : Architecture de Harvard : ..... 14

Figure II.3 : Structure de Harvard modifiée : .....15

Figure II.4: organisation Processeur pipeline : ..... 18

Figure II.5: La Famille AVR : .....20

Figure II.6 : Synoptique générale d'un ATMEGA 32 : .....22

Figure II.7: brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC : ..... 23

Figure III.1: Structure d'E/S d'un module I2C : .....27

Figure III.2: Exemple de condition de départ et d'arrêt : ..... 28

Figure III.3: Exemple de transmission réussie : ..... 29

Figure III.4: Exemple d'octet d'adresse : .....29

Figure III.5: Ecriture d'une donnée (ds1307) : .....30

Figure III.6: le chronogramme correspondant des lignes SDA et SCL : .....31



## Liste Des Figures

---

Figure III.7: Lecture d'une donnée (ds1307) :.....	32
Figure III.8: Interface du logiciel MikroC :.....	35
Figure III.9 : Création d'un projet en Mikro C :.....	36
Figure III.10 : Les configurations de projet :.....	37
Figure III.11 : Fenêtre de saisie de programme :.....	38
Figure IV.1 : Schéma synoptique du projet :.....	42
Figure IV.2: Interface du logiciel ISIS :.....	43
Figure IV.3: Organisation de la boîte à outils ISIS :.....	45
Figure IV.4: Le logiciel ISIS 7 en mode simulation du projet :.....	46
Figure IV.5: configuration le port c (SDA et SCL) :.....	47
Figure IV.6:Brochage d'afficheur LCD :.....	48
Figure IV.7:Brochage du capteur DS 1621 :.....	49

## ***Liste Des Abréviations***

**A/N** : Convertisseur analogique – numérique  
**N/A** : Convertisseurs Numériques – Analogiques  
**FPGA** : Circuit logique programmable  
**E/S** : Entré Sortie  
**CPU** : Unité centrale de traitement  
**ACC** : Adaptive Cruise Control  
**DSP** : Digital Signal Processor  
**RISC** : Reduced Instruction Set Computer  
**CISC** : Complex Instruction Set Computer  
**PIC** : Programmable Interface Contrôler  
**ALU** : Unité arithmétique et logique  
**USART** : Port série universel mode asynchrone  
**SSP** : Port série synchrone  
**ICSP** : In Circuit Serial Programming  
**MCLR** : Reset  
**I2C** : (IIC) Inter Integrated Circuit  
**SDA** : Signal de donnée  
**SCL** : Signal d'horloge  
**ACK** : Acknowledge (reconnaître)  
**ISIS** : Logicielle de simulation  
**DS** : Dallas  
**LCD** : cristaux liquide display

---

# Sommaire

---

# *Sommaire*

---

## *Sommaire*

Introduction générale :.....	1
------------------------------	---

### *Chapitre I : les systèmes embarqués*

I.1 Introduction :.....	2
I.2 Système embarqué typique :.....	3
I.3 Circuit logique programmable « FPGA » :.....	5
I.4 Caractéristiques principales d'un système embarqué :.....	5
I.5 Le langage C pour l'embarqué :.....	8
I.6 Exemple un système embarqué (radar ACC 77Ghz) :.....	9
I.6.1 Organigramme de radar ACC :.....	10
I.6.2 Fonctionnalité de radar ACC :.....	11
I.7 Conclusion :.....	11

### *Chapitre II : Les microcontrôleurs dans systèmes embarqués*

II.1 Introduction :.....	12
II.2 Organisation du plan mémoire :.....	13
II.2.1 Architecture de Von Neumann et architecture de Harvard :.....	13
II.2.2 Architecture de Harvard modifiée DSP (Digital Signal Processor) :.....	15
II.3 Les type de processeur :.....	15
II.3.1 Architecture RISC (Reduced Instruction Set Computer) :.....	15
II.3.2 Caractéristiques des machines RISC :.....	16
II.3.3 Architecture CISC (Complex Instruction Set Computer) :.....	16
II.3.4 Comparaison des technologies RISC et CISC :.....	17
II.4 Processeur pipeline :.....	18
II.5 Le microcontrôleur ATMEL AVR (ATMEGA) :.....	19

# Sommaire

---

II.5.1 Définition:.....	19
II.5.2 La Famille AVR :.....	19
II.6 Les caractéristique du µc ATMEGA32 :.....	20
II.7 L 'Architecture interne du ATMEGA32 :.....	21
II.8 Présentation Physique :.....	23
II.11 Descriptions des broches :.....	23
II.12 :Le bus I2C et l'atmega 32: .....	25
II.13 Conclusion :.....	25

## *Chapitre II : le protocole bus I2C et micro C*

III.1 Introduction :.....	26
III.2 Présentation et caractéristiques du bus I2C :.....	26
III.3 Le protocole I2C :.....	28
III.3.1 La prise contrôle du bus :.....	28
III.3.2 La transmission d'un octet :.....	28
III.3.3 La transmission d'une adresse :.....	29
III.3.4 Écriture d'une donnée :.....	30
III.3.5 Lecture d'une donnée :.....	31
III.4 Présentation du MikroC :.....	34
III.4.1 Création d'un projet :.....	36
III.4.2 Les bibliothèques du MikroC :.....	39
III.5 Bus I2C et MikroC :.....	40
III.5.1 Fonction d'écriture :.....	40
III.5.2 Fonction de lecture :.....	41
III.6 Conclusion:.....	41

## *Chapitre IV : conception de la carte et simulation*

# *Sommaire*

---

IV.1 Introduction :	42
IV.2 Schéma synoptique du projet:	42
IV.3 Présentation du logiciel ISIS:	43
IV.3.1 Fenêtre d'ensemble (Vue d'ensemble):	44
IV.3.2 Fenêtre d'édition :	44
IV.3.3 La boîte à outils :	44
IV.3.4 Organisation de la boîte à outils :	45
IV.4 Simulation par ISIS :	46
IV.4.1 descriptif des composants utilisés :	47
IV.4.1.2 Unité de commande :	47
IV.5 L'afficheur LCD :	48
IV.5.1 Brochage LCD 2x16:	48
IV.5.2 Mode 4 bits :	49
IV.6 Le Capteur de Temperature DS 1621:	49
IV.6.1 Definition:	49
VI.6.2 Mot de contrôle :	50
VI.6.3 Commande :	50
IV.6 Organigramme de système :	51
IV.7 Conclusion:	54
Conclusion générale:	55
Bibliographie :	56

---

# Introduction générale

---

## ***Introduction générale***

Les nouvelles applications industrielles de contrôle, de commande et de communication sont devenues des systèmes embarqués qui reposent sur l'utilisation de circuits intégrés complexes incluant matériel et logiciels (systèmes d'exploitation et logiciel d'application) implanté dans une même puce électronique, FPGA, ASIC etc.

Que ce soit en génie électriques, dans les entraînements industriels, dans les systèmes de télécommunication ou même dans le domaine de l'instrumentation médicale les systèmes embarqués sont fortement présents.

Notre projet a pour objectif la conception et la mise en œuvre d'un système électronique embarqué capable de piloté via un bus I2C l'affichage d'information sur matrice a diode LED. Le système a concevoir et flexible extensible et programmable en langage C . L'ensemble de système est piloté par le microcontrôleur pic.ce dernier est conçu par Microchip pour les applications similaires.

Dans le premier chapitre nous avons étudié le système embarqué en générale en termes caractéristiques les architecteur. Ces élément sont rassemblés dans un circuit intelligent le microcontrôleur qui est programmé en un langage c .en fin nous avons présenté un l'exemple de système embarqué.

Dans le deuxième chapitre nous avons vu une étude détaillée sur les microcontrôleurs et leur architecteurs et l'organisation de leur mémoire, après comparaison des technologies CISC et RISC on s'est intéressé au microcontrôleur AVR ATMEGA 32 conçu en technologie RISC.

Dans le troisième chapitre nous avons étudié le protocole de bus I2C qui un protocole de communication rapide et facile à mettre en œuvre, Il permet de créer des bus contenant plusieurs maîtres et esclaves. Le logiciel de programmation mikroC et utilisé pour l'implantation de ce protocole.

Dans le dernier chapitre nous présentons les modèles et résultats de simulation en environnement à microcontrôleur AVR ATMEGA 32 pour la mise en œuvre de notre application.





---

# Chapitre I

---

---

---

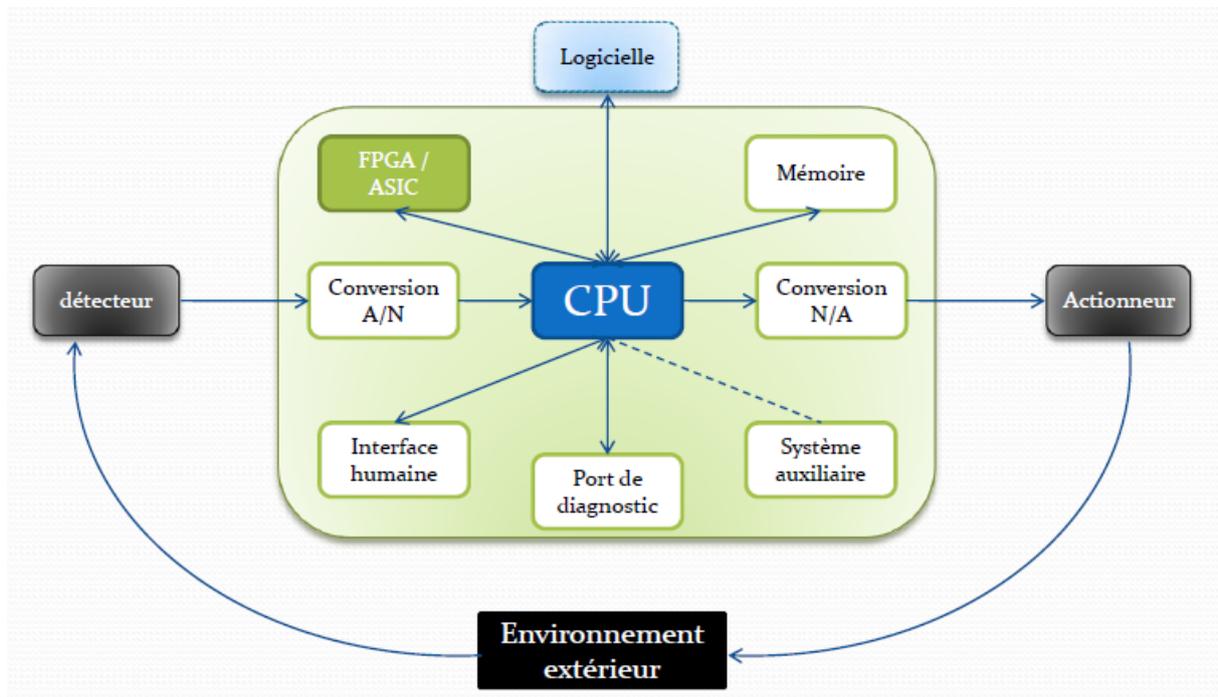
## **I.1 Introduction :**

Un système embarqué est un système électronique et informatique autonome ne possédant pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur. Il est piloté dans la majorité des cas par un logiciel, qui est complètement intégré au système qu'il contrôle. On peut aussi définir un système embarqué comme un système électronique soumis à diverses contraintes.

Les concepteurs des systèmes électroniques sont aujourd'hui confrontés à la complexité croissante des algorithmes mis en œuvre et à la variété des cibles potentielles FPGAs et/ou DSPs. Actuellement, il n'est pas rare que ces systèmes intègrent plusieurs douzaines voire des centaines de processeurs. A l'origine, ce sont des systèmes matériels et logiciels intégrés dans des avions militaires ou des missiles. Ensuite dans le civil: avions, voitures, machine à laver...! La machine et le logiciel sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC. On mentionne les contraintes physiques fortes: dimensions, poids, taille, autonomie, consommation, fiabilité, contraintes temporelles (temps réels).

Un système embarqué peut être défini comme un système électronique et informatique autonome, qui est dédié à une tâche bien précise. Le matériel et le logiciel ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur PC.

## I.2 Système embarqué typique :



**Figure I.1 : Architecture matérielle d'un système embarqué**

On retrouve en entrée des capteurs généralement analogiques couplés à des convertisseurs A/N. On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs N/A. Au milieu, on trouve le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S. Il est à noter qu'il est complété généralement d'un circuit FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur.

On retrouve en fait un système d'asservissement entre les entrées et les sorties, Il est à noter que l'expression la plus simple de cette figure (I.1) est de considérer comme capteurs des interrupteurs et comme actionneurs des LED Sur ce schéma théorique se greffe un paramètre important : le rôle de l'environnement extérieur. Un système embarqué doit faire face à des environnements plus hostiles. Il doit faire face à un ensemble de paramètres agressifs :

- Variations de la température.
- Vibrations, chocs.
- Variations des alimentations.
- Corrosion.
- Eau, feu, radiations.

L'environnement dans lequel opère le système embarqué n'est pas contrôlé ou contrôlable. Cela suppose donc de prendre en compte ce paramètre lors de sa conception. On doit par exemple prendre en compte les évolutions des caractéristiques électriques des composants en fonction de la température, des radiations..., les systèmes embarqués sont aujourd'hui fortement communicants. Cela est possible grâce aux puissances de calcul offertes par les processeurs pour l'embarqué (32 bits en particulier) et grâce aussi à l'explosion de l'usage de la connectivité Internet. La connectivité internet permet fondamentalement de contrôler à distance un système embarqué par Internet. Ce n'est en fait que l'aboutissement du contrôle à distance d'un système électronique par des liaisons de tout type : liaisons RS.232, RS.485, bus de terrain...

Cela permet l'emploi des technologies modernes d'internet pour ce contrôle à distance par l'utilisateur : il suffit d'embarquer un serveur internet dans son équipement électronique pour pouvoir le contrôler ensuite à distance, de n'importe où, à l'aide d'un simple navigateur. Il n'y a plus d'interface humaine spécifique à concevoir pour cela, ce rôle étant rempli par le navigateur Internet. Cela est une réalité : les chauffagistes proposent maintenant des chaudières pouvant être pilotées par le web ! Il faut aussi noter la montée en puissance des communications sans fil dans l'embarqué au détriment des communications filaires pour limiter le câblage et faciliter la mise en place du système embarqué. Le wifi et toutes les normes de réseaux sans fil IEEE 802.15 comme Zigbee dans l'embarqué et surtout en domotique (réseaux de capteurs sans fil par exemple). Mais ne nous méprenons pas sur ces facilités et commodités, cela a bien sûr un revers : la sécurité du système embarqué, puisque connecté à Internet. [1]

Un système embarqué composé de :

- Unité centrale de traitement (Central processing unit « CPU »)
- Les Convertisseurs analogiques – numériques (CAN)

- Les Convertisseurs Numériques – Analogiques (CNA)
- Les mémoires
- Les Circuits logiques programmables « FPGA »

### **I.3 Circuit logique programmable « FPGA » :**

Un circuit logique programmable, ou réseau logique programmable, est un circuit intégré logique qui peut être reprogrammé après sa fabrication. Il est composé de nombreuses cellules logiques élémentaires librement assemblables.

La plupart des grands FPGA modernes sont basés sur des cellules SRAM aussi bien pour le routage du circuit que pour les blocs logiques à interconnecter. FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur [1]

Les circuits spécifiques sont spécialement conçu au spécialement adapté au système embarqué les microcontrôleurs.

### **I.4 Caractéristiques principales d'un système embarqué :**

Les principales caractéristiques d'un système embarqué sont les suivantes :

- C'est un système principalement numérique.
- Il met en œuvre généralement un processeur.
- Il exécute une application logicielle dédiée pour réaliser une fonctionnalité précise et n'exécute donc pas une application scientifique ou grand public traditionnelle.
- Il n'a pas réellement de clavier standard (Bouton Poussoir, clavier matriciel...). L'affichage est limité (écran LCD...) ou n'existe pas du tout.
- Ce n'est pas un PC en général mais des architectures similaires (x86) basse consommation sont de plus en plus utilisées pour certaines applications embarquées. De ce constat, on peut voir.
- Qu'un PC standard peut exécuter tout type d'applications car il est généraliste alors qu'un système embarqué n'exécute qu'une application dédiée.

- Que des circuits numériques ou des circuits analogiques sont utilisés en plus pour augmenter les Performances du système embarqué ou sa fiabilité.

L'omniprésence des systèmes embarqués dans notre vie est liée à la révolution numérique opérée dans les années 1970 avec l'avènement des processeurs. Les processeurs, de plus en plus rapides, puissants et bon marché ont permis cette révolution et aussi le boom du marché de l'embarqué. Ceci se confirme au travers de la loi empirique de Gordon Moore, cofondateur d'Intel, qui stipule que pour une surface de silicium donnée, on double le nombre de transistors intégrés tous les 18 mois ! La figure 1 montre cette évolution inexorable. [1]

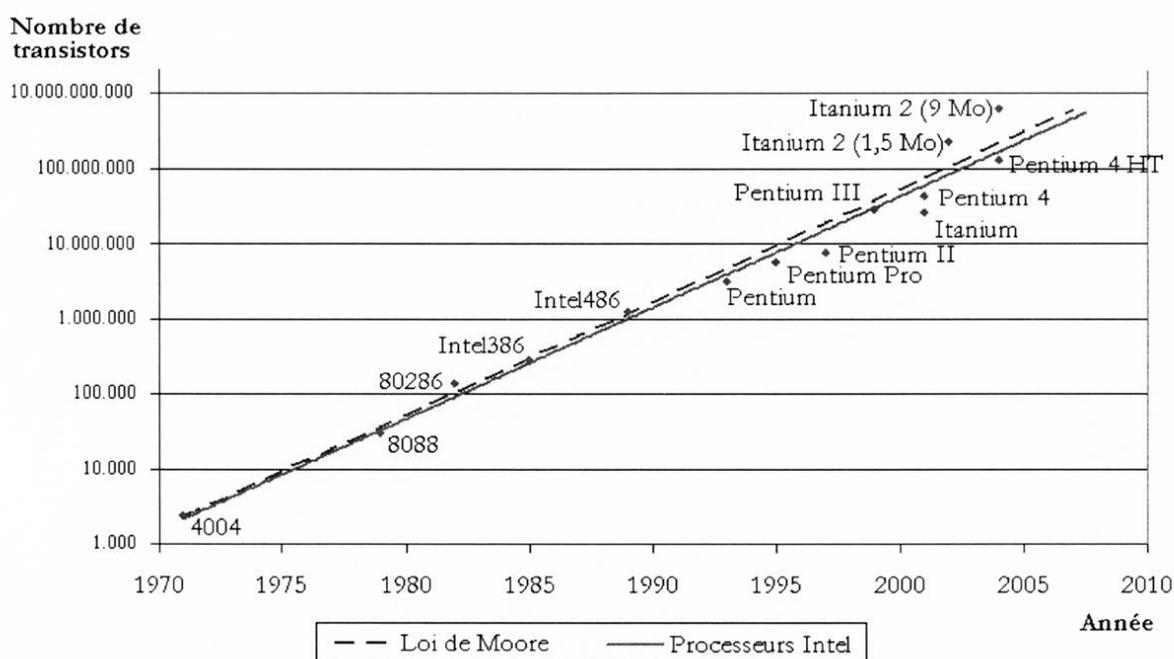


Figure I.2 : La loi G. Moore pour les processeurs Intel

En 1999, il a été vendu pour le marché de l'embarqué :

- 1,3 milliard de processeurs 4 bits.
- 1,4 milliard de processeurs 8 bits.
- 375 millions de processeurs 16 bits.



- 127 millions de processeurs 32 bits.
- 3,2 millions de processeurs 64 bits.

A côté de cela, à cette époque, il a été vendu seulement 108 millions de processeurs (famille x86) pour le marché du PC grand public ! Pour 2004, il a été vendu environ 260 millions de processeurs pour le marché du PC grand public à comparer aux 14 milliards de processeurs tout type confondu (microprocesseur, microcontrôleur, DSP) pour le marché de l'embarqué. Le marché des processeurs pour l'embarqué selon *électroniques*. ce Réseau de recherche devrait croître de 6 % en 2005 pour un chiffre d'affaire mondial de 18 milliards USD, Les chiffres parlent d'eux-mêmes. On peut aussi tirer le constat actuel suivant :

- Moins de 2 % des processeurs vendus sont pour le marché du PC contre 98 % pour l'embarqué.

On utilise massivement pour le PC grand public le système d'exploitation commercial bien connu.

- Pour les 98 % autres processeurs vendus, on utilisera généralement un autre système d'exploitation. On trouvera ici dans 60 % des cas un système d'exploitation commercial spécialisé pour l'embarqué [Vx Works (système d'exploitation temps réel multitâche), QNX (système d'exploitation UNIX)]. Pour le reste, il s'agit d'un système d'exploitation home made mais de plus en plus optent pour des systèmes d'exploitation libres comme Linux pour l'embarqué.

- Moins de 10 % des processeurs vendus sont des processeurs 32 bits pour près de 31 % du chiffre d'affaire sur les processeurs. Cette part du chiffre d'affaire est estimée à près de 48 % pour 2008 : cela montre la migration rapide vers ces processeurs 32 bits dans l'embarqué, condition nécessaire pour pouvoir mettre en œuvre Linux.

- Si l'on regarde le prix moyen d'un processeur tout type confondu, on arrive à 6 USD par unité à comparer au prix moyen de 300 USD par unité pour un processeur pour PC. Le marché du processeur pour PC est très faible en volume mais extrêmement lucratif, Ces quelques chiffres permettent bien de prendre conscience de l'importance du marché de l'embarqué et de mettre fin à l'idée qu'en dehors du marché du PC, point de salut. Les grands secteurs de l'embarqué concernent les domaines suivants :

- Jeux et calcul général : application similaire à une application de bureau mais empaquetée dans un système embarqué : jeux vidéo, set top box...
- Contrôle de systèmes : automobile, processus chimique, processus nucléaire, système de navigation...
- Traitement du signal : radar, sonar, compression vidéo...
- Communication et réseaux : transmission d'information et commutation, Internet... [1]

### **I.5 Le langage C pour l'embarqué :**

Le langage de programmation C est la référence pour programmer les microcontrôleurs avec des exigences temps réel. En effet, ce dernier présente l'avantage d'être un langage de programmation orienté matériel. Le code machine généré par le compilateur est presque optimal en ce qui concerne d'une part la taille de la mémoire requise et d'autre part la vitesse d'exécution du programme. Par conséquent, environ 65% des applications embarquées sont programmées en C.

Le C est un langage :

- 1- universel : C n'est pas orienté vers un domaine d'applications spéciales comme SQL par exemple.
- 2- compact : C'est basé sur un noyau de fonctions et d'opérateurs limités qui permet la formulation d'expressions simples mais efficaces.
- 3- moderne : C'est un langage structuré, déclaratif et récursif. Il offre des structures de contrôle et de déclaration comme le langage Pascal.
- 4- près de la machine : C offre des opérateurs qui sont très proches de ceux du langage machine (Manipulations de bits, pointeurs...). C'est un atout essentiel pour la programmation des systèmes embarqués.
- 5- rapide : C permet de développer des programmes consistants et rapides.

6- indépendant de la machine : C'est un langage près de la machine (microprocesseur) mais il peut être utilisé sur n'importe quel système ayant un compilateur C.

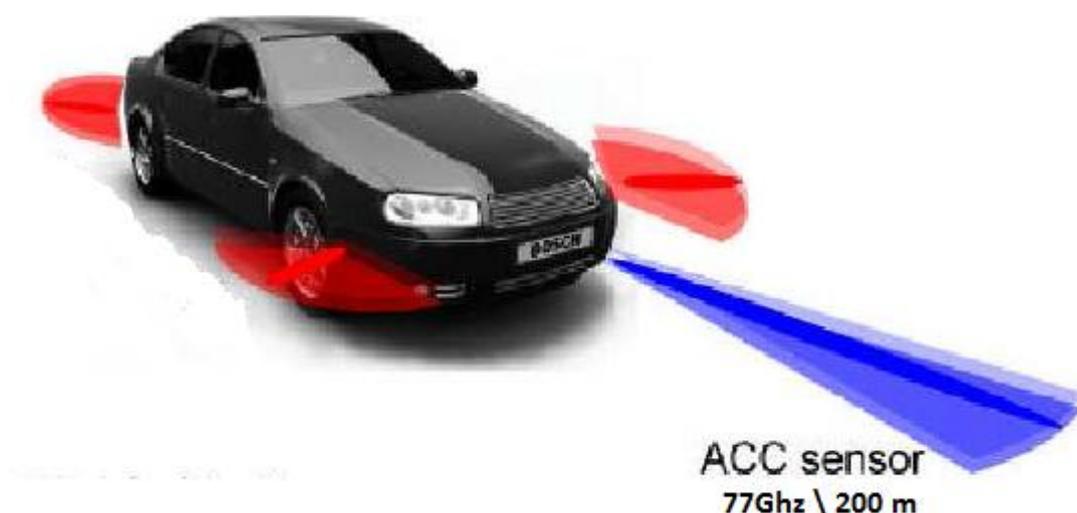
Au début C'était surtout le langage des systèmes UNIX mais on le retrouve comme langage de développement aujourd'hui de tous les systèmes d'exploitation.

7- portable: En respectant le standard ANSI-C, il est possible d'utiliser (théoriquement) le même programme sur tout autre système (autre hardware, autre système d'exploitation), simplement en

Le recompilant. Il convient néanmoins de faire attention dans le cas d'un système embarqué qui n'inclut pas toujours un système d'exploitation.[2]

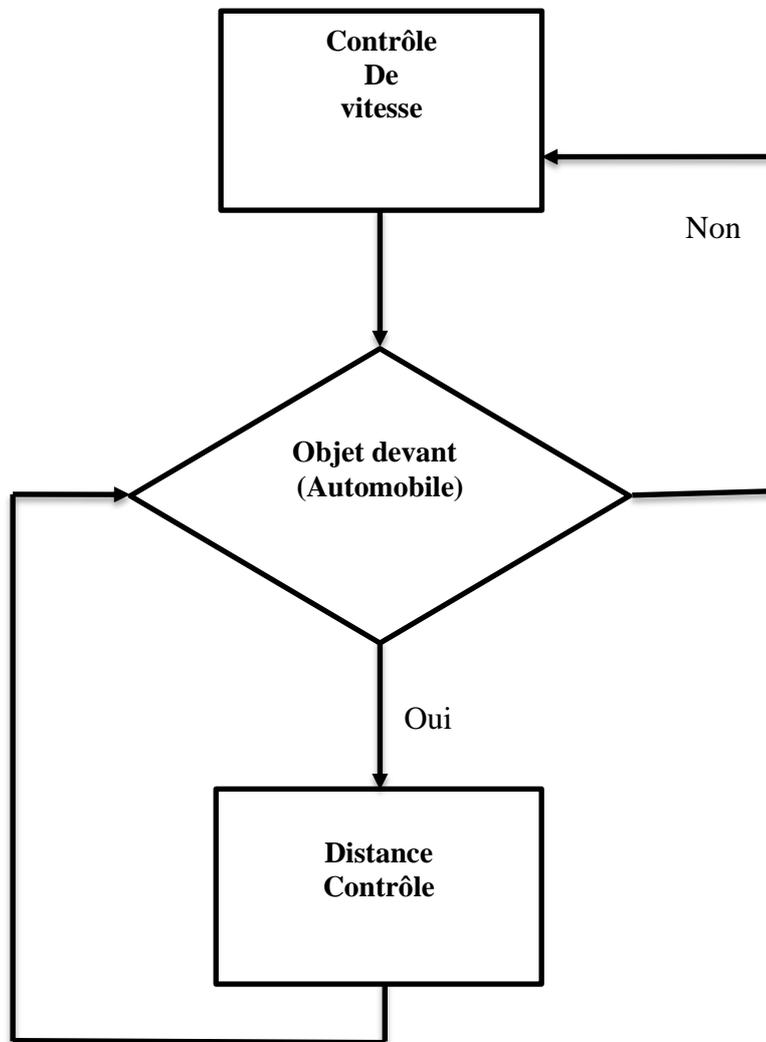
### **I.6 Exemple un système embarqué (radar ACC 77Ghz) :**

ACC : Initiales pour Adaptive Cruise Control, ce terme désigne le régulateur de vitesse gérant l'inter-distance avec le véhicule qui précède. L'usage d'un radar à ondes Millimétriques permet d'évaluer la position et la vitesse relatives du véhicule précédent. Une régulation intervient par action sur l'accélérateur et les freins pour maintenir entre. Les deux véhicules une distance qui respecte les contraintes de sécurité définies dans la, Stratégie implémentée dans le dispositif. Il y a plusieurs applications pour les systèmes radars automobile comme indiqué sur la Figure (I.3). Pour le système 77 GHz qui est adapté jusqu'à 200 m, le système radar est utilisé comme: radar de prospection, régulateur de vitesse et d'espacement (ACC), [3]



**Figure.I.3 : radar ACC 77Ghz.[3]**

### I.6.1 Organigramme de radar ACC :



### I.6.2 Fonctionnalité de radar ACC :

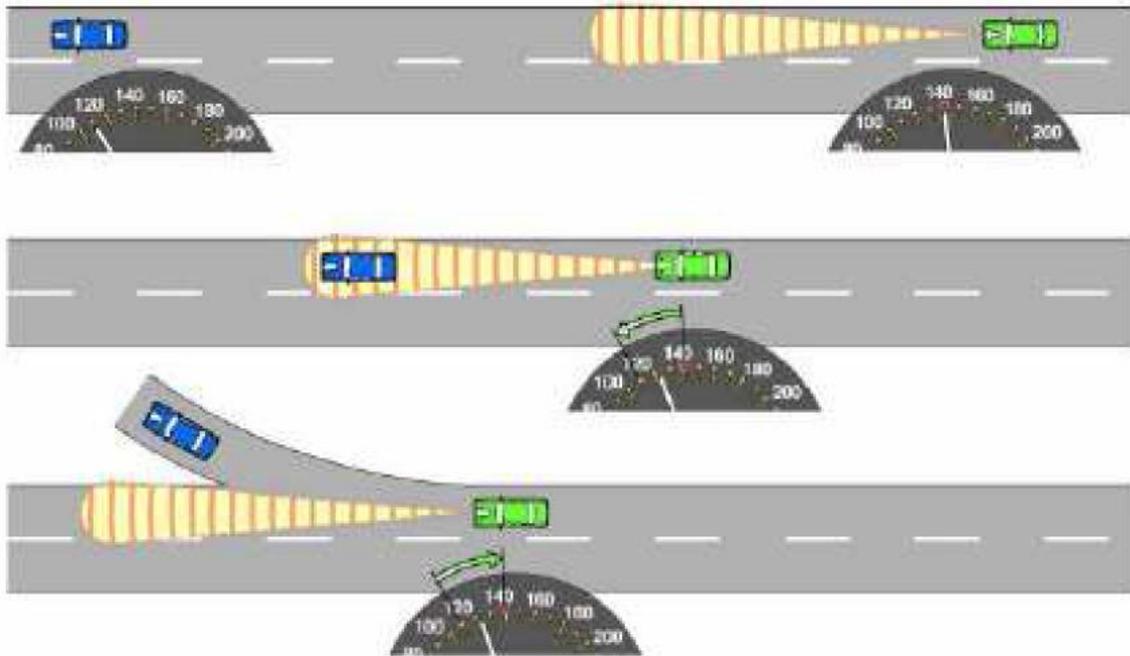


Figure.I.4 : Fonctionnalité de radar ACC [3]

### I.7 Conclusion :

Dans ce chapitre, nous avons présenté d'une manière les systèmes embarqué, leur principes de fonctionnement leurs architecteur ainsi que l'outil principale dévalement software, le langage c dédié à ce système.

---

# Chapitre II

---



## II.1 Introduction :

Le processeur constitue le noyau principale d'un système embarqué. Il doit avoir un certain nombre de caractéristiques lui permettant d'accomplir sa tâche convenablement : Rapidité, Consommation, puissance....

La notion de puissance est la capacité de traiter un grand nombre d'opérations par seconde sur de grands nombres et en grande quantité. Intrinsèquement la puissance se joue donc sur les trois critères suivants:

- La longueur des mots : données et instructions (on parle de largeur du bus des données).
- Le nombre d'octets que le microprocesseur peut adressé (on parle de largeur du bus des adresses).
- La vitesse d'exécution des instructions liée à la fréquence de fonctionnement de l'horloge de synchronisation exprimée en MHZ. Sur ce il ya différents architectures et technologies d'un système de processeur dédié.

Les classifications des microcontrôleurs à deux niveaux :

- Au niveau du processeur:
  - RISC : Reduced Instruction Set Computer
  - CISC : Complex Instruction Set Computer
- Au niveau de l'organisation de la mémoire
  - Architecture Von Neumann : une mémoire unique et pour le programme et pour les données
  - Architecture Harvard : le programme et les données sont stockées dans des mémoires physiquement séparées.



## II.2 Organisation du plan mémoire :

Il existe 2 types de plan mémoire :

- architecture de type Von Neumann
- architecture de type Harvard

### II.2.1 Architecture de Von Neumann et architecture de Harvard :

À chaque cycle d'horloge, le processeur sait par le compteur de programme l'instruction qu'il doit faire exécuter. Nous avons vu qu'il va chercher chaque instruction en mémoire, l'exécute avec les données correspondantes et retourne les données résultantes en mémoire.

Dans l'architecture de la machine de Von Neumann, le programme et les données sont enregistrés sur la même mémoire. Chaque instruction contient la commande de l'opération à effectuer et l'adresse de la donnée à utiliser, il faut donc souvent plusieurs cycles d'horloge pour exécuter une instruction. La figure II.1 indique une architecture simple de Von Neumann, constituée d'un bus de données et de programme et d'un bus d'adresses.

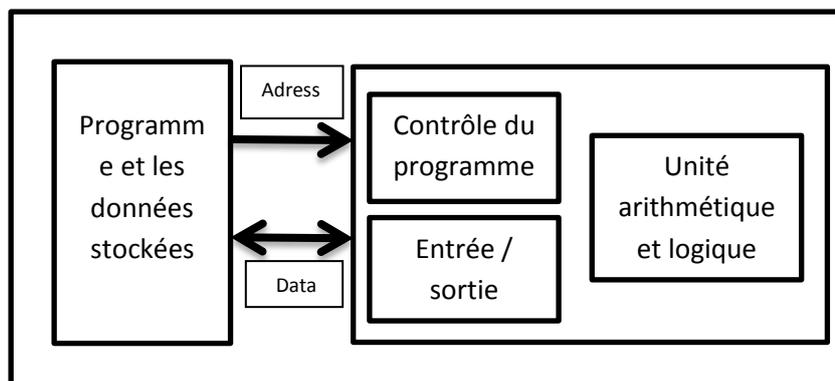
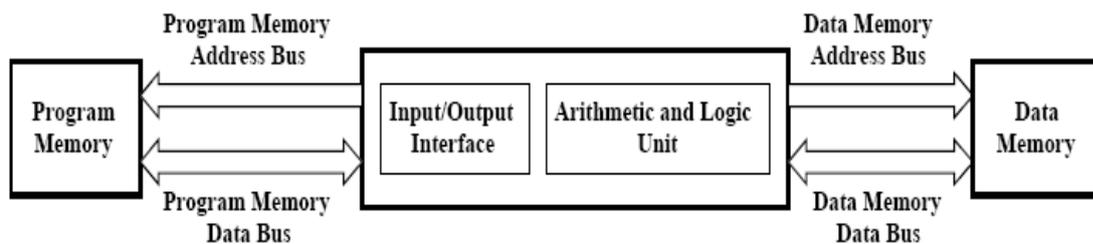


Figure.II.1 : Architecture de Von Neumann.

On voit que les échanges s'effectuent de manière simple entre l'unité arithmétique et logique (ALU), c'est-à-dire l'unité centrale et la mémoire unique, par un bus transitant les codes de programme et les données. On a ainsi des données « collées » aux instructions. Les microprocesseurs et beaucoup de microcontrôleurs utilisent cette architecture car elle est très souple pour la programmation.[4]

Dans l'architecture dite de Harvard (car mise au point dans cette université américaine en 1930), on sépare systématiquement la mémoire de programme de la mémoire des données : l'adressage de ces mémoires est indépendant. La figure (II.2) indique une architecture simple de Harvard constituée d'un bus de données, d'un bus de programme et de deux bus d'adresse.



**Figure.II.2 : Architecture de Harvard**

On voit que les échanges s'effectuent de manière double entre l'unité centrale et les deux mémoires, ce qui permet une grande souplesse pour l'enregistrement et l'utilisation des données. D'ailleurs, la mémoire de programme est également utilisée en partie comme mémoire de données pour obtenir encore plus de possibilités de traitement avec des algorithmes complexes.[4]

L'architecture Harvard est souvent utilisée dans :

- les processeurs numériques de signal (DSP) ;
- les microcontrôleurs, notamment les PIC de Micro chip et les AVR d'Atmel.

### II.2.2 Architecture de Harvard modifiée DSP (Digital Signal Processor) :

La structure de HARVARD est cependant pénalisante par le nombre de broches utilisées et le nombre de boitiers mémoires utilisés comme de nombreux autres constructeurs (INTEL avec le MCS 51, MICROCHIP avec les microcontrôleurs PIC) utilise une structure de HARVARD modifiée, qui consiste à n'utiliser qu'un bus commun pour les accès en mémoire externe (architecture de VON NEUMANN) et à utiliser une structure de HARVARD pour les accès en mémoire interne. Charge ensuite au programmeur, si la taille de la mémoire programme ou de la mémoire données est trop importante, de placer en mémoire interne le bout de programme et de données qui demande une rapidité optimale d'exécution.[5]

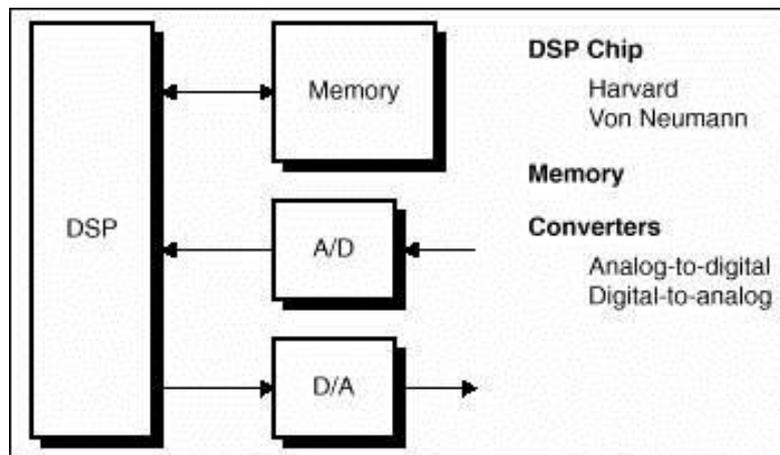


Figure.II.3 : Structure de Harvard modifiée

### II.3 Les type de processeur :

Il existe sur le marché deux types de processeurs:

#### II.3.1 Architecture RISC (Reduced Instruction Set Computer) :

Architecture dans laquelle les instructions sont en nombre réduit (chargement, branchement, appel sous-programme) et elles sont fréquemment utilisées. Le but est

d'éliminer les instructions rarement employées et de consacrer les ressources matérielles à exécuter les instructions relativement simples en un cycle d'horloge et à émuler les autres instructions à l'aide des séquences basées sur les instructions élémentaires. On trouve donc une meilleure performance à une vitesse donnée (le gain en performance envisageable est important mais dépend de la qualité du compilateur).[6]

### **II.3.2 Caractéristiques des machines RISC :**

1. un cycle d'horloge au maximum pour l'exécution d'une instruction ;
2. une dépendance plus forte par rapport aux compilateurs ;
3. toutes les instructions ont la même longueur ;
4. une utilisation systématique des registres (généralement plus nombreux que sur les processeurs CISC) ;
5. séquenceur câblé plutôt que programmé ;
6. une limitation des accès mémoire aux deux opérations LOAD et STORE. Stations de travail équipées de processeurs à architecture RISC : SUN, HP, IBM.

Processeurs RISC : Alpha (DEC), Power RISC (IBM), SPARC (SUN), PA-RISC (HP).[6]

### **II.3.3 Architecture CISC (Complex Instruction Set Computer) :**

C'est une architecture avec un grand nombre d'instructions. Le processeur doit exécuter des tâches complexes par instruction unique. Donc, pour une tâche donnée, une machine CISC exécute un petit nombre d'instructions mais chacun nécessite un plus grand nombre de cycles d'horloge (Pentium et PowerPC). Actuellement les deux technologies convergent : les processeurs CISC (Pentium par exemple) utilisent des instructions de plus en plus simples et exécutent parfois plusieurs instructions en un cycle d'horloge. [6]

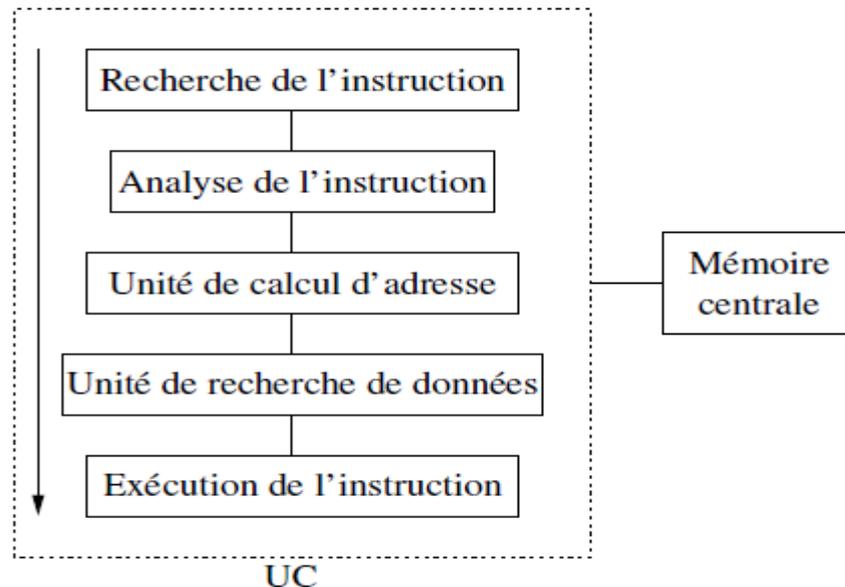
**II.3.4 Comparaison des technologies RISC et CISC :**

	<b>Cisc</b>	<b>Risc</b>
Signification	Complexe Instructions Set Computer.	Reduced Instructions Set Computer.
Traduction	Processeur à jeu d'instructions complexe.	Processeur à jeu d'instructions réduit.
Apparition	Technologie plus ancienne.	Technologie plus récente.
Principes fondamentaux	Complexité des instructions disponibles. Le jeu d'instructions s'est enrichi au fil du temps en gardant une compatibilité ascendante. Le décodage assez lent de chaque instruction.	- Report de la complexité matérielle sur le logiciel (compilateur). - Effectuer des accès à la mémoire. Uniquement au travers de deux instructions : Load et Store. Être adapté aux langages de haut niveau.
Jeu d'instruions	Environ 350 instructions.	- Environ 150 instructions. Utilisation d'un jeu d'instructions de base minimal, simulation des autres.
Format des instructions	Variable selon J'instruction.	Unique ou presque.
Durée d'exécution d'une instruction	instructions simples ne prenant qu'un seul cycle.	Instructions complexes prenant plusieurs cycles.
Hierarchisation de la mémoire	Moins de 10 registres internes spécialisés les nombreux accès à la mémoire.	- 64 registres internes et polyvalents.
Fonctionnement	- séquentiel - Logique de traitement et décodage qui occupe 50% de la surface du circuit Séquenceur microprogramme.	- pipeline - Logique de traitement et décodage qui occupe 5 à 10% de la surface du circuit. Séquenceur câblé. il reste de la place pour de nombreuses fonctions supplémentaires qui augmentent les performances.
Comparatif	Technologie moins puissante. Plus complexe, moins performante et plus chère.	- Inverse. - L'architecture plus adaptée à l'évolution l'informatique.

**Tableau II.1 : Comparaison des technologies RISC et CISC [7]**

## II.4 Processeur pipeline :

L'idée est inspirée de l'organisation du travail à la chaîne : cette technique permet d'effectuer davantage de travail par unité de temps. Chaque étape de l'exécution de l'instruction (chargement, décodage, recherche des données) se fait dans une unité séparée :



**Figure.II.4: organisation Processeur pipeline**

- la première est chargée dans la première unité ;
- la deuxième unité commence à décoder l'instruction pendant que la première travaille à charger la deuxième instruction ;
- un peu plus tard, la troisième unité travaille sur le calcul d'adresse pendant que la deuxième unité décode la deuxième instruction, et que la première unité charge l'instruction suivante (exemple : IBM 360/195, CDC STAR).[8]

Un microcontrôleur, est un composant électronique qui rassemble tous les éléments d'un "mini-ordinateur" et qui se présente sous la forme d'un circuit intégré. Un microcontrôleur permet de réaliser des systèmes et montages électroniques programmés. Cela veut dire que l'on pourra, avec le même montage, réaliser des fonctions très différentes qui dépendront du programme qui aura été programmé dans le microprocesseur.[9]

## **II.5 Le microcontrôleur ATMEL AVR (ATMEGA) :**

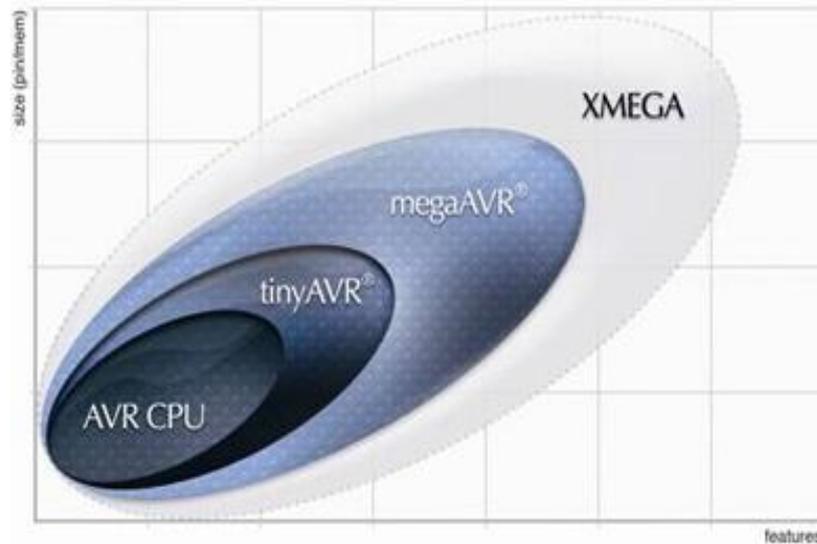
### **II.5.1 Définition:**

Les microcontrôleurs de la famille ATMEGA en technologie CMOS sont des modèles à 8 bits AVR (Architecteur Harvard Modifié) basés sur l'architecture RISC. En exécutant des instructions dans un seule cycle d'horloge simple, l'ATMEGA réalise des opérations s'approchant de 1 MIPS par MHZ permettant de réaliser des systèmes à faible consommation électrique et simple au niveau électronique.[10]

### **II.5.2 La Famille AVR :**

La famille AVR 8bits (ATMEL) regroupe une centaine de composants, chacun disponible dans plusieurs types de boîtiers physique (package). La famille AVR peut être découpée en 3 sous familles principales :

- Attiny AVR (8 à 20 broches)
- Méga AVR (32 à 100 broches)
- xMega AVR (44 à 100 broches)



**Figure II.5: La Famille AVR**

## II.6 Les caractéristique du $\mu$ ATMEGA32

Vous trouverez dans le tableau qui suit tous les modèles d'ATMEGA existant. Le modèle pris comme référence pour ce document est l'ATMEGA 32-16 PI pour être précis.

La série ATMEGA fournit les particularités suivantes :

- Une FLASH RAM de 32 Ko (Flash Programmable).
- Une EEPROM de 1 Ko (Electrique Ecriture Programmable ROM),
- Une RAM de 2 Ko octets (Random Access Memory).
- 40 lignes d'entrée-sortie universelles (Port A, B, C, D,).
- 32 registres de travaux universels qui dialoguent directement avec l'unité centrale ( ALU).
- Une horloge en temps réel (RTC).
- Trois Timer/Compteurs flexibles avec comparaison des modes, interruption internes et externes,
- Un comparateur analogique (entrée sur PB2, PB3).
- Un convertisseur Analogique/Numérique ADCde8 canaux à 10 bits,
- Un Chien de Garde programmable avec oscillateur interne.[10]



- Une ou deux interfaces série USART programmable & périodique (série asynchrone et synchrone).
- Une interface la série SP trois modes sélectionnables (série synchrone).
- Une interface I2C pour la gestion d'un bus à 2 fils, Un interface LCD pour piloter directement un écran LCD (modèles spéciaux).
- Tension d'alimentation de 2,7 V à 5,5 V.

## **II.7 L 'Architecture interne du ATMEGA32**

Le cœur AVR (Architecteur Harvard Modifié ) combine un jeu de 131 instructions riches avec 32 registres spéciaux travaillant directement avec l'Unité Arithmétique de Logique ALU, qui représente le registre d'accumulateur A(B ou D) dans les microcontrôleurs classiques.[10]

Nous abordons le vif du sujet avec le synoptique qui présente le fonctionnement général du microcontrôleur ATMEGA :

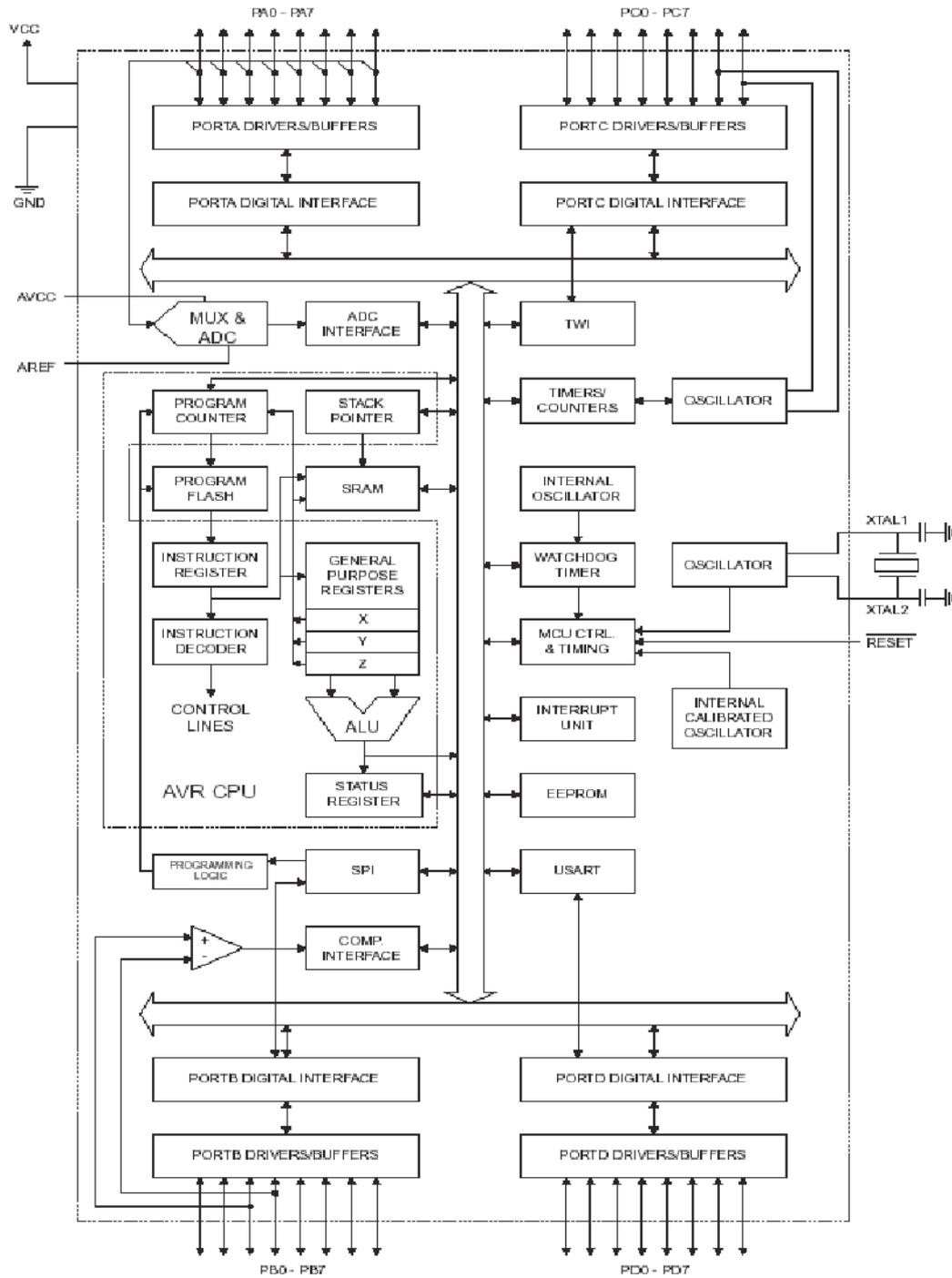


Figure II.6 : Synoptique générale d'un ATMEGA 32.

## II.8 Présentation Physique

L'ATMEGA se présente sous la forme d'un circuit intégré à 40 broches pour le modèle ATMEGA32 en boîtier PDIP ou le boîtier TQFP/MLF à 44 broches.

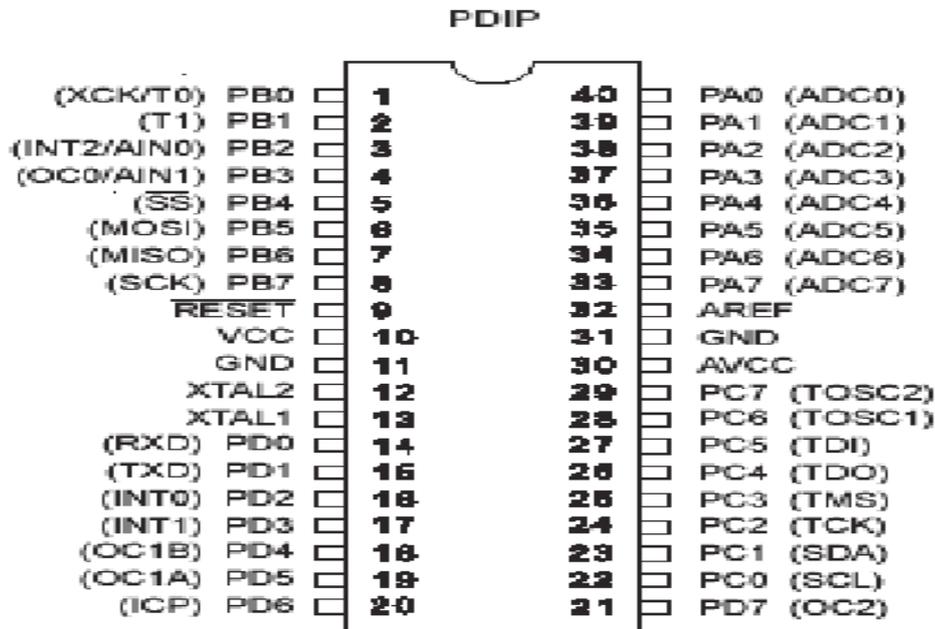


Figure II.7: brochage typique d'un ATMEGA32 en boîtier PDIP et PLCC.[10]

## II.11 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.[10]

## **II.12 : Le bus I2C et l'atmega 32:**

Le AVR atmega 32 possède des entrées/sorties spécialement prévues pour la communication I2C. Ce sont les broches nommées **RC3 (SCL)** et **RC4 (SDA)**. Elles sont directement reliées à un contrôleur I2C. Ce contrôleur prévoit la gestion de tout le protocole I2C (Prise de parole, collision, ligne occupée...). Il permet une gestion facile de la liaison, étudier en détail au chapitre 3.

## **II.13 Conclusion :**

Nous avons vu une étude détaillée sur microcontrôleurs et leur architecture et organisation de leur mémoire, après comparaison des technologies CISC et RISC on s'est intéressé au microcontrôleur ATMEL AVR (ATMEGA 32) conçu en technologie la plus récente, avec des composants RISC et une Architecture Harvard Modifiée.

---

# Chapitre III

---

---

---

### **III.1 Introduction :**

Le bus I2C (Inter Integrated Circuit) a été développé au début des années 80 par Philips semi-conducteurs pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur moderne est contrôlé le bus I2C avec microcontrôleur AVR En utilisant la programmation et choisissez le programme MikroC.

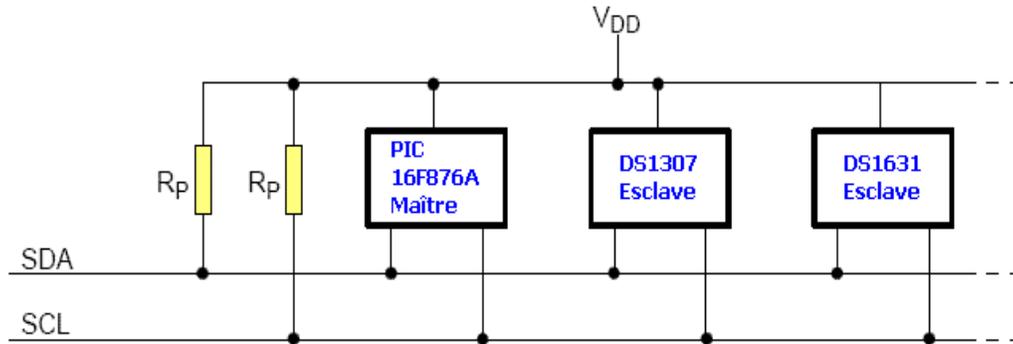
### **III.2 Présentation et caractéristiques du bus I2C :**

Le bus" I2C a été inventé pour simplifier la communication entre des composants divers dont le sigle signifie Inter Integrated Circuit ce qui donne IIC et par contraction I2C. Le protocole est initialement proposé par Philips mais adopté de nos jours par de très nombreux fabricants. C'est un bus de communication de type série synchrone. Ce bus est parfois nommé sous le nom de TWI (Two Wire Interface). Il permet de communiquer sur 3 fils :

- Signal de donnée: SDA (Serial Data)
- Signal d'horloge: SCL (Serial Clock)
- Signal de référence électrique : masse de périphériques.

Ceci permet de réaliser des équipements ayant des fonctionnalités très puissantes (En apportant toute la puissance des systèmes microprogrammes) et conservant un circuit imprimé très simple, par rapport un schéma classique (8bits de données, 16 bits d'adresse+ Les bits de contrôle). Les données sont transmises en série à 100Kbits/s en mode standard et jusqu'à 400Kbits/s en mode rapide. Ce qui ouvre la porte de cette technologie à toutes les applications où la vitesse n'est pas primordiale. Le nombre de composants qu'il est ainsi possible de relier est essentiellement limité par la charge capacitive des lignes SDA et SCL : 400 pF. [11]





Afin d'éviter les conflits électriques les Entrées/Sorties SDA et SCL sont de type "Collecteur Ouvert" Voici un schéma de principe :

Deux résistances de pull-up sont bien sûr placées entre les lignes SDA et SCL et l'alimentation (VDD).

Quand le bus n'est pas utilisé, SDA et SCL sont au niveau haut (niveau de repos).

Quand une ligne (SDA ou SCL) est au repos (niveau 1), on peut la forcer à 0.

Quand une ligne (SDA ou SCL) est au niveau 0, on ne peut pas la forcer à 1.

Dans l'exemple ci-dessous, l'horloge SCL est au niveau 0 car c'est le PIC 16F877A (maître) qui force cette ligne à 0 :

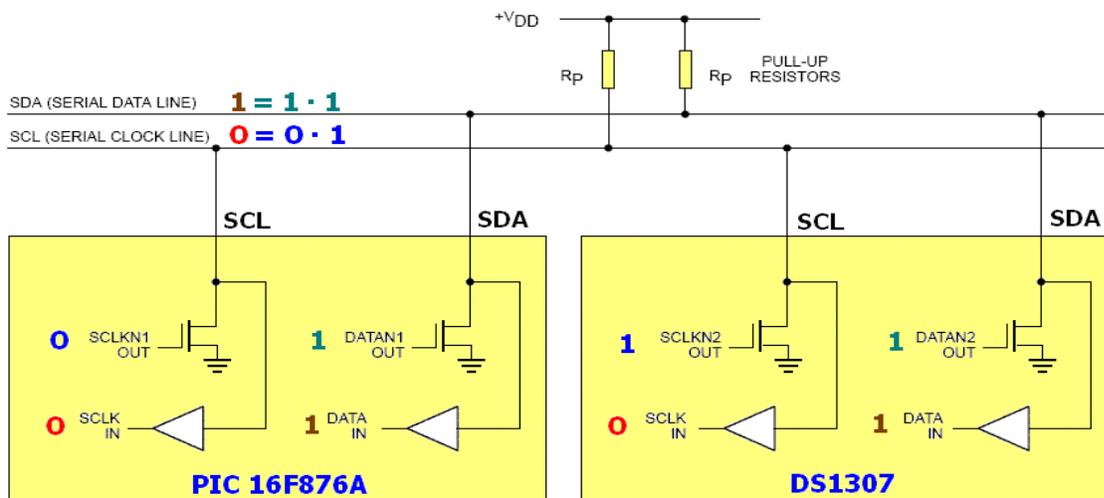


Figure III.1: Structure d'E/S d'un module I2C [12]

## III.3 Le protocole I2C :

Le protocole I2C définit des états logiques possibles sur SDA et SCL, et la façon dont doivent réagir les circuits en cas de conflits.

### III.3.1 La prise contrôle du bus :

Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (SDA et SCL à '1'). Pour transmettre des données sur le bus, il faut donc surveiller deux conditions Particulières :

- La condition de départ. (SDA passe à '0' alors que SCL reste à '1')
- La condition d'arrêt. (SDA passe à '1' alors que SCL reste à '1')

Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le **maître**. C'est lui qui génère le signal d'horloge.

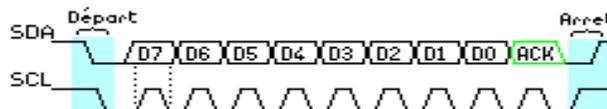


Figure III.2: Exemple de condition de départ et d'arrêt

### III.3.2 La transmission d'un octet :

Après avoir imposé la condition de départ, le maître applique sur SDA le bit de Poids fort D7. Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne SCL. Lorsque SCL revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis. Il envoie alors un bit ACK à '1' tout en scrutant l'état réel de SDA. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement. Les sorties de chacun étant à collecteurs ouverts, le maître voit le '0' et peut alors passer à la suite.

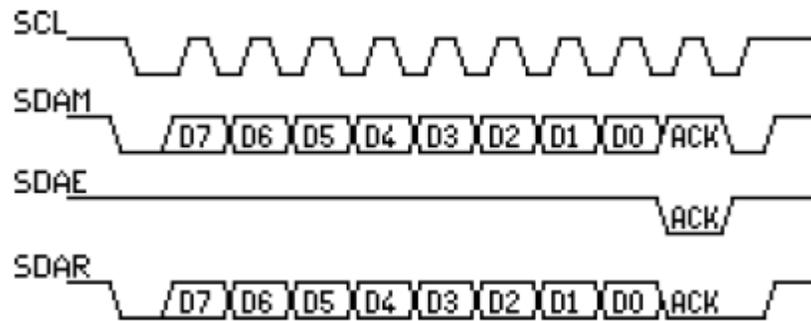


Figure III.3: Exemple de transmission réussie

Dans cet exemple :

- SCL : Horloge imposée par le maître.
- SDAM : Niveaux de SDA imposés par le maître.
- SDAE : Niveaux de SDA imposés par l'esclave.
- SDAR : Niveaux de SDA réels résultants.

### III.3.3 La transmission d'une adresse :

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il est nécessaire de définir pour chacun une adresse unique. L'adresse d'un circuit, codée sur sept bits, est définie d'une part par son type et d'autre part par l'état appliqué à un certain nombre de ces broches. Cette adresse est transmise sous la forme d'un octet au format particulier.



Figure III.4: Exemple d'octet d'adresse

On remarque ici que les bits D7 à D1 représentent les adresse A6 à A0, et que le bit D0 et remplacé par le bit de R/W qui permet au maître de signaler s'il veut lire ou écrire une donnée. [13]

### III.3.4 Écriture d'une donnée :

- Exemple d'écriture du DS1307 L'émetteur est le maître et le récepteur est l'esclave. Le registre d'adresse 0x04 du DS1307 contient la date Pour régler le calendrier au 27 du mois, il faut écrire la donnée 0x27 (codage BCD) dans le registre d'adresse 0x04 du DS1307.

Le bus I2C utilise le protocole suivant :



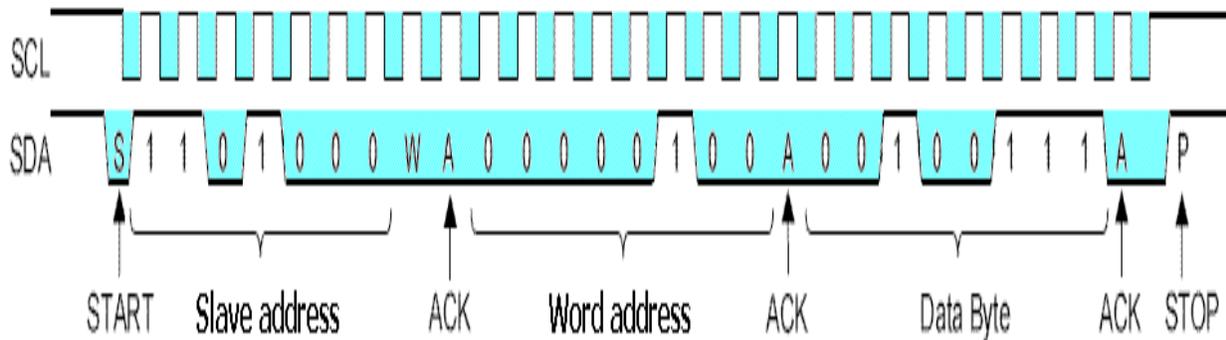
**Figure III.5: Ecriture d'une donnée (ds1307)**

- 1) Pour initier le dialogue, le maître crée une condition **Start**
- 2) Le maître envoie l'adresse de l'esclave (11001000) suivi du bit 0 (bit Write)
- 3) L'esclave répond (accusé de réception : bit Acknowledge)
- 4) Le maître envoie l'adresse du registre (0x04) à écrire
- 5) L'esclave répond (accusé de réception : bit Acknowledge)
- 6) Le maître envoie la donnée (0x27) à écrire
- 7) L'esclave écrit la donnée puis envoie un accusé de réception (bit Acknowledge)

8) Le maître termine le dialogue avec une condition **Stop**

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).

Un peu moins évident, le chronogramme correspondant des lignes SDA et SCL :



**Figure III.6: le chronogramme correspondant des lignes SDA et SCL [14]**

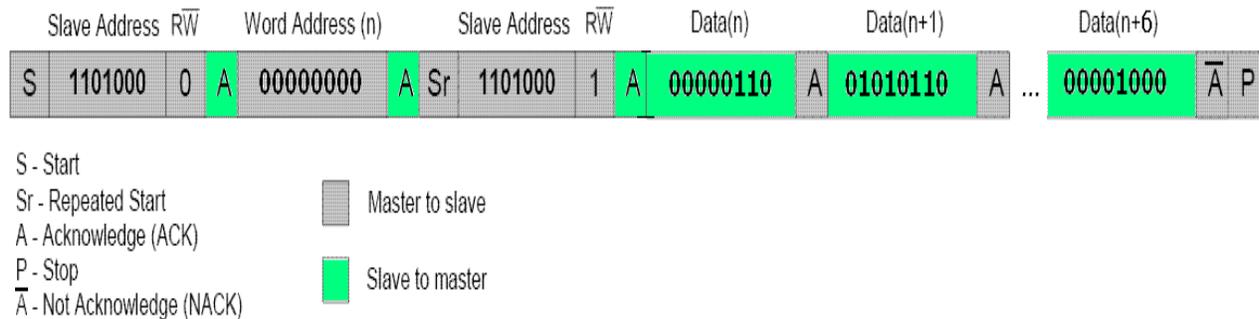
### III.3.5 Lecture d'une donnée :

- Exemple de lecture du DS1307

L'émetteur est l'esclave et le récepteur est le maître.

Les registres d'adresses 0x00 à 0x06 du DS1307 contiennent respectivement les secondes, minutes, heures, jours, dates, mois et années (cf. datasheet du DS1307).

Voici comment lire, le contenu des registres d'adresses 0x00 à 0x06 du DS1307 :



**Figure III.7: Lecture d'une donnée (ds1307)**

- 1) Pour initier le dialogue, le maître crée une condition **Start**
- 2) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 0 (bit Write)
- 3) L'esclave répond (accusé de réception : bit Acknowledge)
- 4) Le maître envoie l'adresse du registre (0x00) à lire
- 5) L'esclave répond (accusé de réception : bit Acknowledge)
- 6) Le maître émet une condition **Repeated Start**
- 7) Le maître envoie l'adresse de l'esclave (1010100) suivi du bit 1 (**bit Read**)
- 8) L'esclave répond (accusé de réception : bit Acknowledge)
- 9) L'esclave envoie le contenu du registre d'adresse 0x00 au maître
- 10) **Le maître répond** (accusé de réception : bit Acknowledge)
- 11) L'esclave envoie le contenu du registre d'adresse 0x01 (automatiquement incrémenté) au maître
- 12) Le maître répond (accusé de réception : bit Acknowledge)

13) L'esclave envoie le contenu du registre d'adresse 0x02 (automatiquement incrémenté) au maître

14) Le maître répond (accusé de réception : bit Acknowledge)

...

21) L'esclave envoie le contenu du registre d'adresse 0x06 (automatiquement incrémenté) au maître

22) Le maître répond (accusé de réception : bit **Not Acknowledge**)

23) Le maître termine le dialogue avec une condition Stop

Le bus I2C est maintenant libre (SCL = 1, SDA = 1 : niveaux de repos).

Le contenu du registre d'adresse 0x00 du DS1307 est 0x06 (codage BCD : 06 secondes).

Le contenu du registre d'adresse 0x01 est 0x56 (c'est-à-dire 56 minutes).

Le contenu du registre d'adresse 0x02 est 0x09 (c'est-à-dire 09 heures).

Le contenu du registre d'adresse 0x03 est 0x03 (c'est-à-dire Mardi).

Le contenu du registre d'adresse 0x04 est 0x07 (c'est-à-dire 7 ème jour du mois).

Le contenu du registre d'adresse 0x05 est 0x05 (c'est-à-dire mois de mai).

Le contenu du registre d'adresse 0x06 est 0x0D (c'est-à-dire année 2013).

(Mardi 07 mai 2013, 9 heures 56 minutes et 6 secondes). [12]

La programmation des microcontrôleurs AVR est supportée par plusieurs langages de Programmation tel que : MikroC for avr, MPLAB, Mikrobasic, HI-TECH C for AVR, flowcode . . etc. . Dans le chapitre présent nous présentons la procédure de programmation des microcontrôleurs..

Dans notre projet nous avons opté pour le compilateur de MIKRO C qui est un compilateur en langage C (langage évolué) et qui permet d'intégrer certaines routines en assembleur.

### **III.4 Présentation du MikroC :**

Le « MikroC » est un compilateur pour AVR Conçu par la société « Mikroelektronika », le compilateur C nouvelle génération "MikroC" pour microcontrôleurs AVR bénéficie d'une prise en main très facile. Il comporte plusieurs outils intégrés (mode simulateur, terminal de communication, gestionnaire 7 segments, analyseur statistique, correcteur d'erreur, explorateur de code...) ; Il a une capacité à pouvoir gérer la plupart des périphériques rencontrés dans l'industrie (Bus I2C, 1Wire, SPI, RS485, Bus CAN, cartes compact Flash, signaux PWM, afficheurs LCD et 7 segments...); de ce fait il est un des outils de développement incontournable et puissant.



Il est conçu pour fournir les solutions les plus faciles que possibles pour des applications se développant pour les systèmes à microcontrôleur. Il contient un large ensemble de bibliothèques de matériel, de composant et la documentation complète.

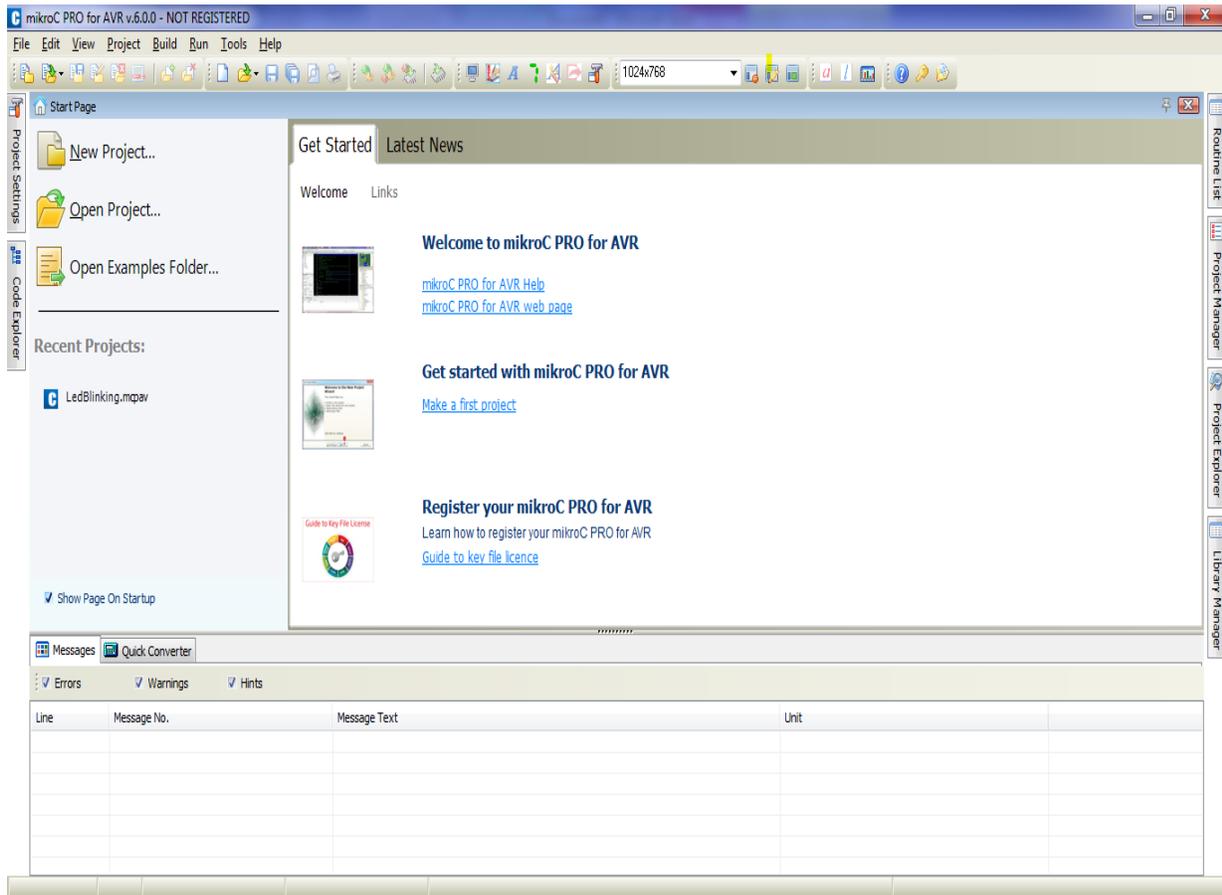


Figure III.8: Interface du logiciel MikroC

## III.4.1 Création d'un projet :

Le processus de création d'un nouveau projet est vraiment très simple. Sélectionnez New Project (Nouveau Projet) depuis le menu Project (Projet), comme indiqué sur (la figure III.9).

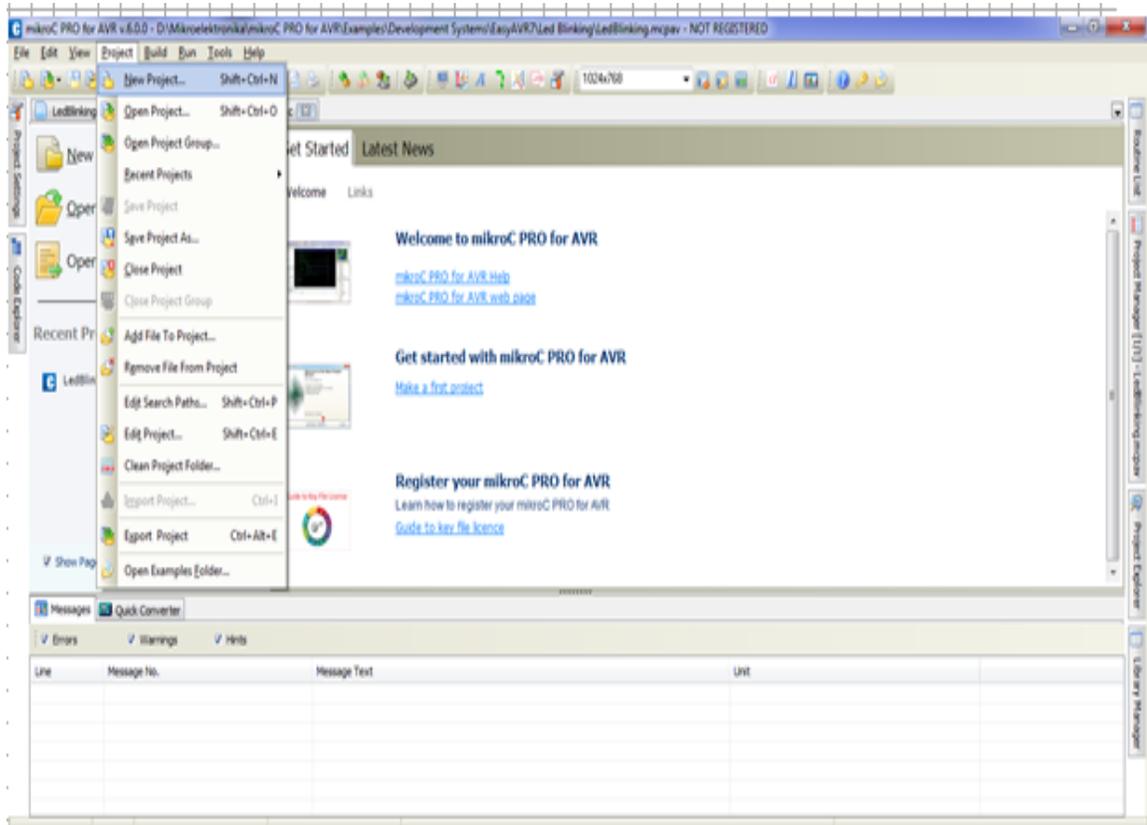
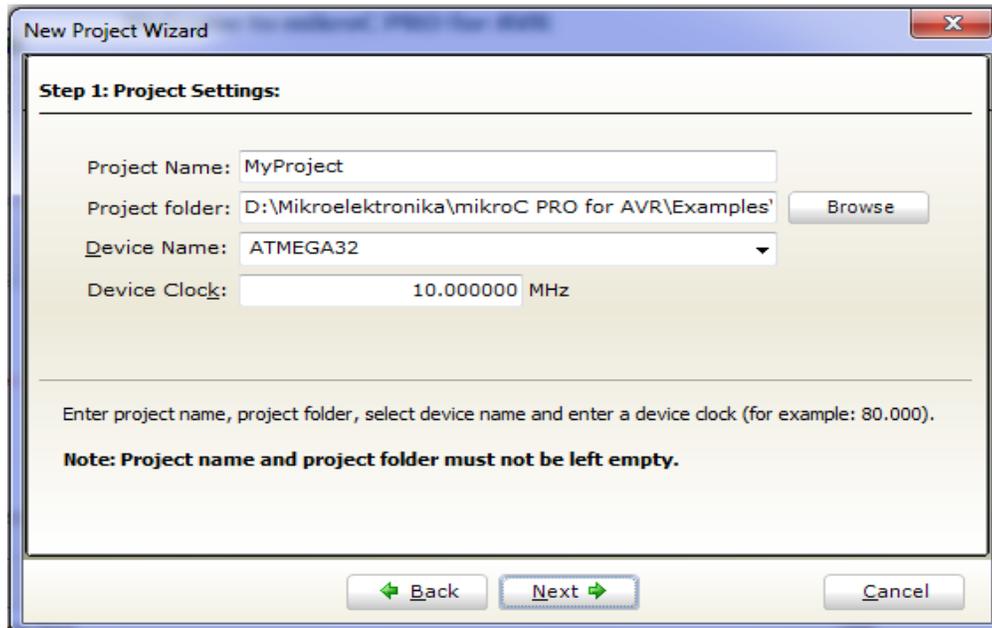


Figure III.9 : Création d'un projet en Mikro C

Une nouvelle fenêtre apparaîtra. Comme indiqué sur (figure 3.4), il y a plusieurs champs à renseigner comme le nom du projet, l'emplacement du projet, sa description, l'horloge et les options du composant. Le tableau device flags (options composant) est utilisé pour la configuration des paramètres du microcontrôleur.



**Figure III.10 : Les configurations de projet**

- 1- Nom de projet
- 2- Nom de l'emplacement du projet
- 3- Nom de composant
- 4- Le choix de la fréquence d'horloge interne de l'AVR

Après ça, une nouvelle fenêtre (figure III.11) vide s'affiche pour écrire notre programme

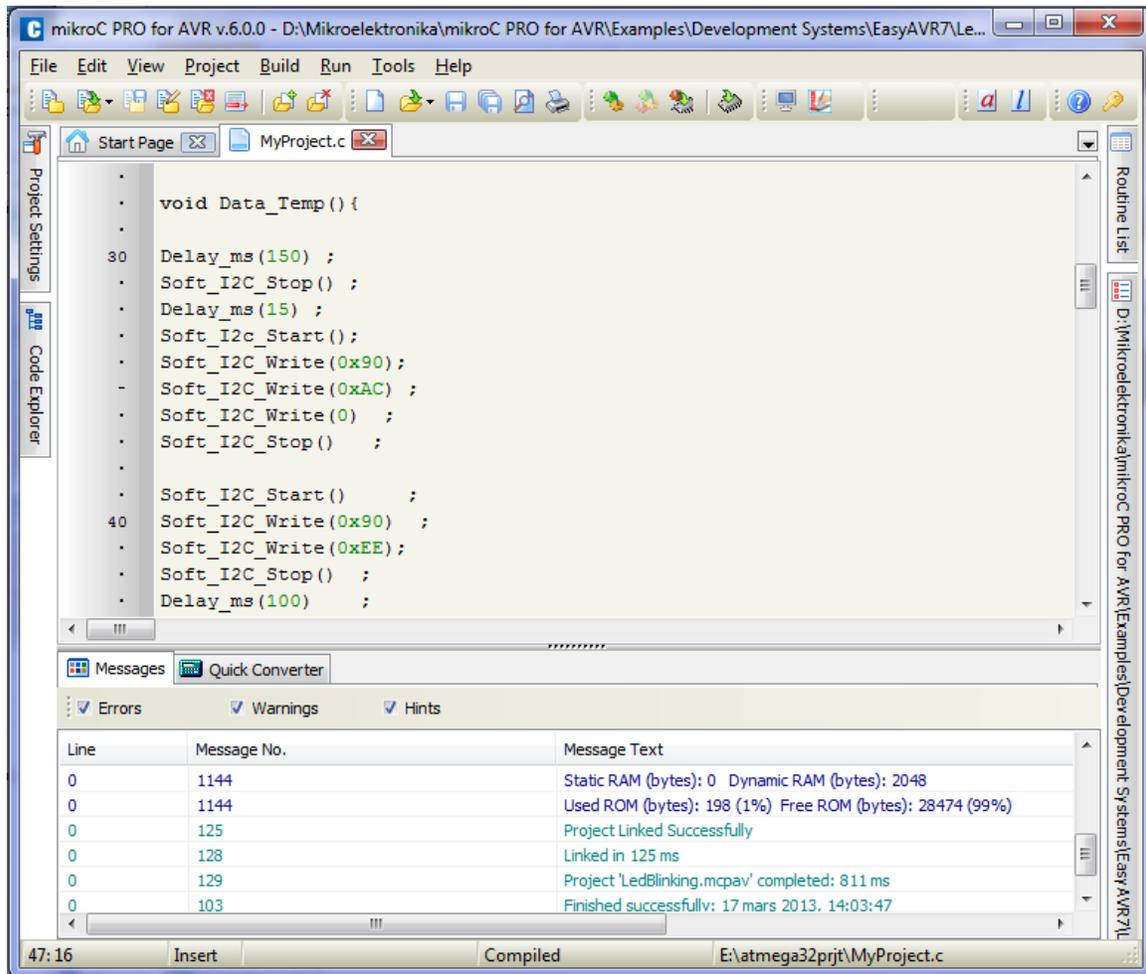


Figure III.11 : Fenêtre de saisie de programme

## III.4.2 Les bibliothèques du MikroC :

MikroC fournit quelques bibliothèques et nous avons utilisé uniquement les bibliothèques suivantes :

Bibliothèque	Les fonctions	La description
<b>I2C</b>	<b>Soft_I2C_Init</b>	Initialise I <sup>2</sup> C avec horloge souhaitée Exemple : Soft_I2C_Init();
	<b>Soft_I2C_Start</b>	Détermine si I <sup>2</sup> C bus est libre et lancer un signal START. Exemple : Soft_I2C_Start ();
	<b>Soft_I2C_Repeated_Start</b>	Répéter un signal START. Exemple : Soft_I2C_Repeated_Start();
	<b>Soft_I2C_Rd</b>	Lit un octet de l'esclave, et envoie pas signal d'acquiescement ACK si le paramètre est 0, sinon il envoie reconnaître. Exemple : take = Soft_I2C_Rd(0);
	<b>Soft_I2C_Wr</b>	Envoie octet de données (données de paramétrage) par l'intermédiaire du bus I <sup>2</sup> C. Exemple : Soft_I2C_Write(0xA3);
	<b>Soft_I2C_Stop</b>	Lancer un signal STOP. Exemple : Soft_I2C_Stop();

Tableaux III.1 : Les bibliothèques du MikroC

### III.5 Bus I2C et MikroC :

Mikro C fournit une bibliothèque prête du bus I2C (Tableaux 3.1), nous utilisons cette bibliothèque pour communiquer entre le maitre (Atmega 32) et l'esclave (DS 1621). L'exemple suivant montre comment écrire de fonction d'écrire en l'esclave ds 1621 avec les commentaires.

#### III.5.1 Fonction d'écriture :

```
void write_DS 1621 (char Temp_H1, Temp_H4, Temp_H2, Temp_H3)
{
Soft_I2c_Start (); // lancer un signal de départ (le bit de start)

Soft_I2C_Write (0x90); // envoyer un octet via I2C (adresse de composant + un bit de mode action
(R/W))

Soft_I2C_Write(0xAC) ; // command writes to the configuration register

Soft_I2C_Write(0) ;

Soft_I2C_Stop() ; // envoyer un signal de stop ( bit de stop)
}
```

Un exemple de l'utilisation de cette fonction :

```
write_ds1621 (T1); // afficher température
```

### III.5.2 Fonction de lecture :

```
void Read_DS 1621 (char Temp_H1, Temp_H4, Temp_H2, Temp_H3)
{
Soft_I2C_Start() ; // lancer un signal de départ (le bit de Start)

Soft_i2c_Write(0x9E); // envoyer un octet via I2C (adresse de composant + un bit de mode action
(R/W))

Soft_I2C_Write(0xAA) ; //command reads the last temperature conversion result

Soft_I2C_Start() ;

Soft_I2C_Write(0x9F) ; // envoyer un octet via I2C (adresse de composant + un bit de mode
action(R/W))

Temp_H3 = Soft_I2C_Read(1) ;
}
```

### III.6 Conclusion:

Dans ce chapitre nous avons étudié le protocole de bus I2C, c'est un protocole de communication rapide et facile à mettre en œuvre. Nous avons présenté le principe de lecture et d'écriture d'une donnée, il permet de créer des bus contenant plusieurs maitres et esclaves. Le logiciel de programmation mikro C a été utilisé dans l'implantation de ce protocole.

---

# Chapitre IV

---



## IV.1 Introduction :

L'objectif de notre projet est de mettre en œuvre une petite chaîne d'acquisition et d'affichage graphique de température de plusieurs points sous forme de courbe ou sous forme alphanumérique.

La chaîne utilisée plus d'un  $\mu$ C ATMAEL des capteurs de température a bus I2C et une matrice graphique LCD ou un affichent alphanumérique à cristaux liquide.

## IV.2 Schéma synoptique du projet :

La figure ci-dessous présente le schéma synoptique dès système d'affichage a développer on trouve :

- Le microcontrôleur AVR ATMEGA 32
- Les capteurs de température ds1621
- L'afficheur LCD

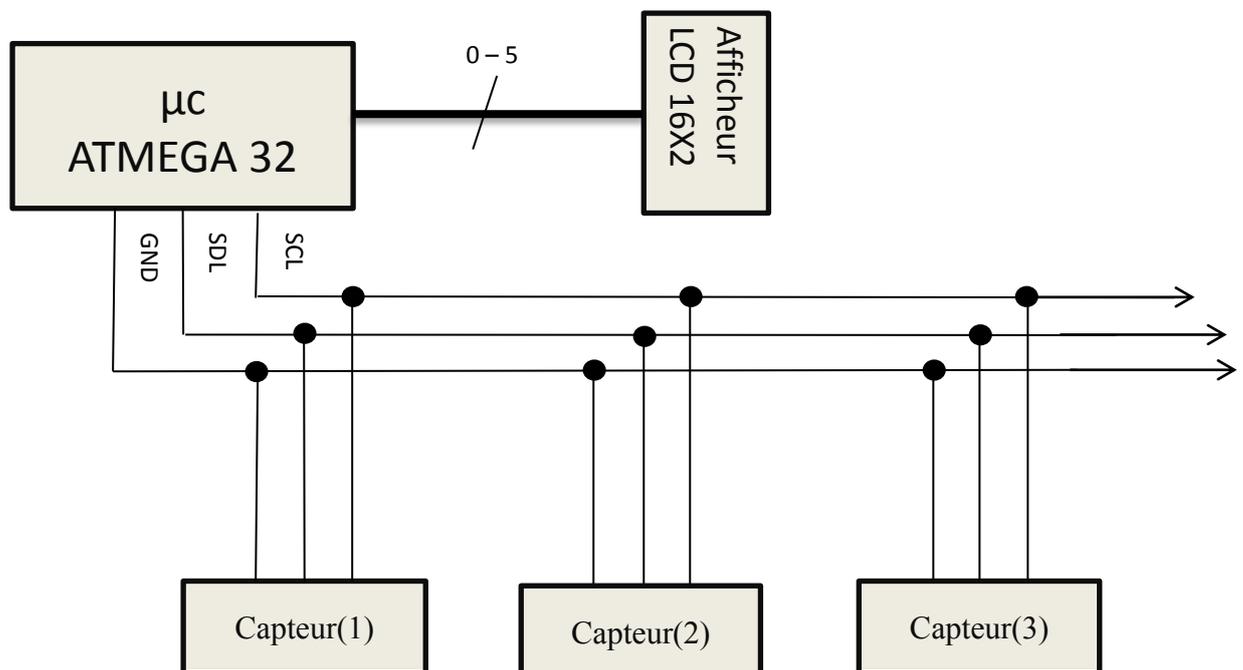


Figure.IV.1 : Schéma synoptique du projet

### IV.3 Présentation du logiciel ISIS :

Isis est un éditeur de schémas qui intègre un simulateur analogique / logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes.

Dans cette section nous allons commencer par la présentation de la fenêtre du logiciel ISIS Figure IV.2. [13]

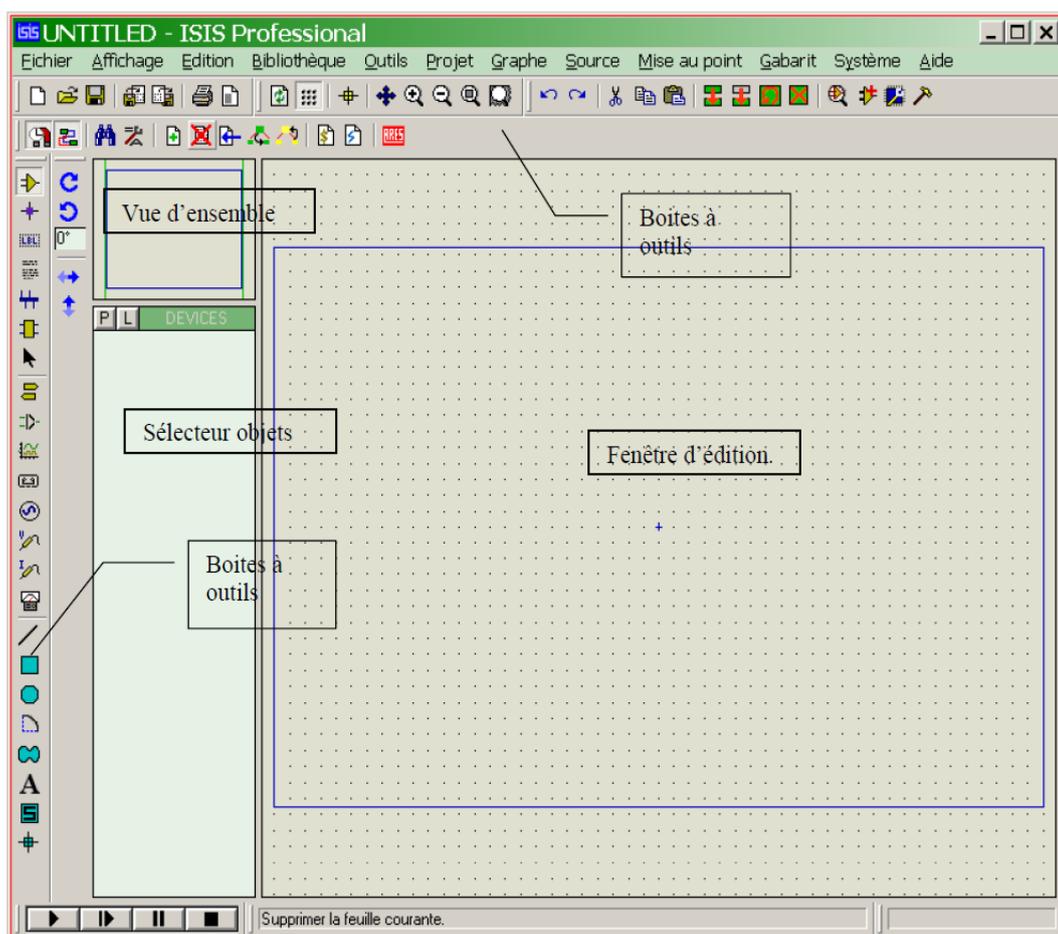


Figure.IV.2: Interface du logiciel ISIS

### **IV.3.1 Fenêtre d'ensemble (Vue d'ensemble) :**

Le cadre en bleu délimite l'espace de travail tel qu'il a été défini par la commande 'Définir taille des feuilles' du menu 'système'.

Le cadre en vert délimite La zone de travail, c'est à dire la partie du schéma visible dans la fenêtre principale.

- Vous pouvez déplacer cette zone de travail en pointant la souris sur la zone désirée de la fenêtre d'ensemble et en effectuant un clic gauche.
- Vous pouvez redéfinir la zone de travail dans la fenêtre d'ensemble en appuyant sur la touche majuscule 'shift' du clavier, associée au déplacement de la souris en maintenant appuyé le bouton gauche. [14]

### **IV.3.2 Fenêtre d'édition :**

La surface la plus grande de l'écran s'appelle "Fenêtre d'édition" et se comporte comme une fenêtre de dessin. C'est là que vous placez et câblez les composants.

### **IV.3.3 La boîte à outils :**

Elle est composée d'un ensemble d'icônes dont les fonctions seront détaillées ultérieurement et d'un sélecteur d'objet utilisé pour choisir les boîtiers, le style des pastilles, des traces, des traversées, etc. [14]

### IV.3.4 Organisation de la boîte à outils :

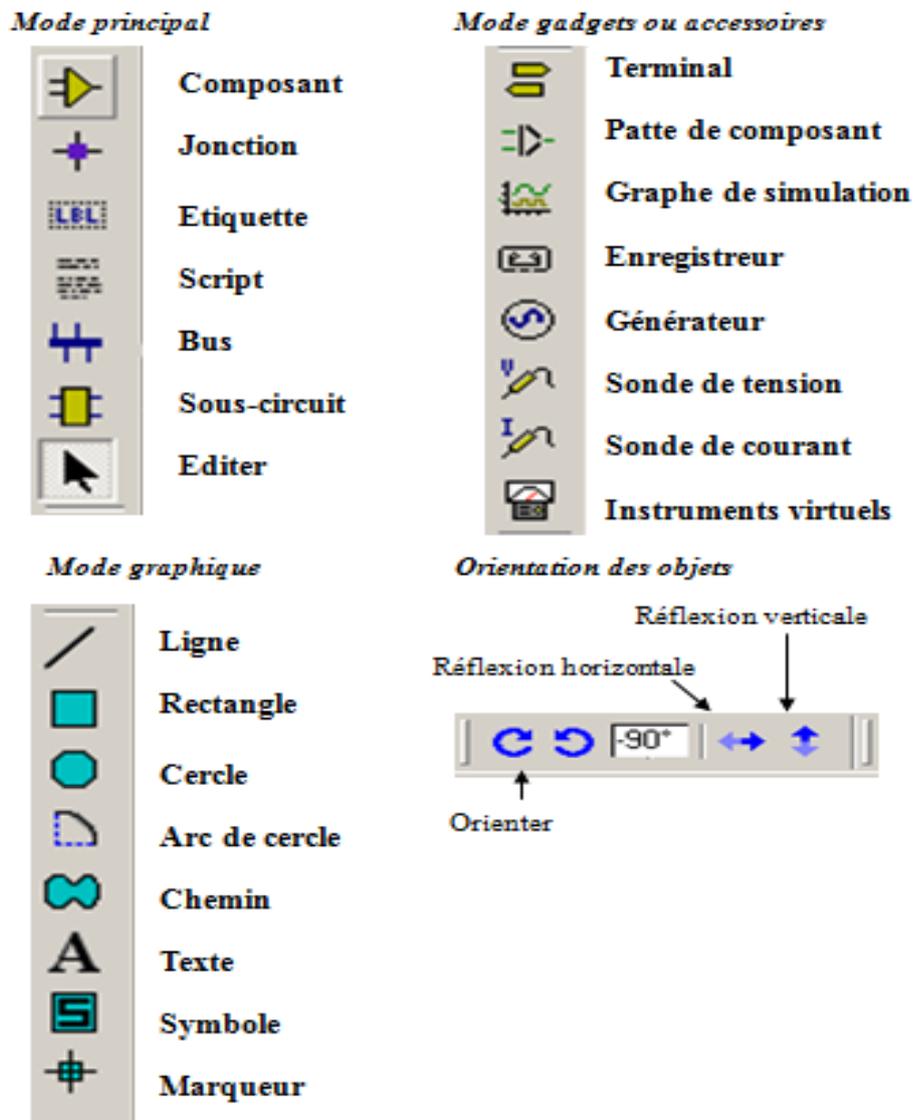


Figure IV.3: Organisation de la boîte à outils ISIS

### IV.4 Simulation par ISIS :

Dans notre projet nous avons utilisé le logiciel ISIS pour valider le schéma proposé : comme l'indique la figure (IV.4).

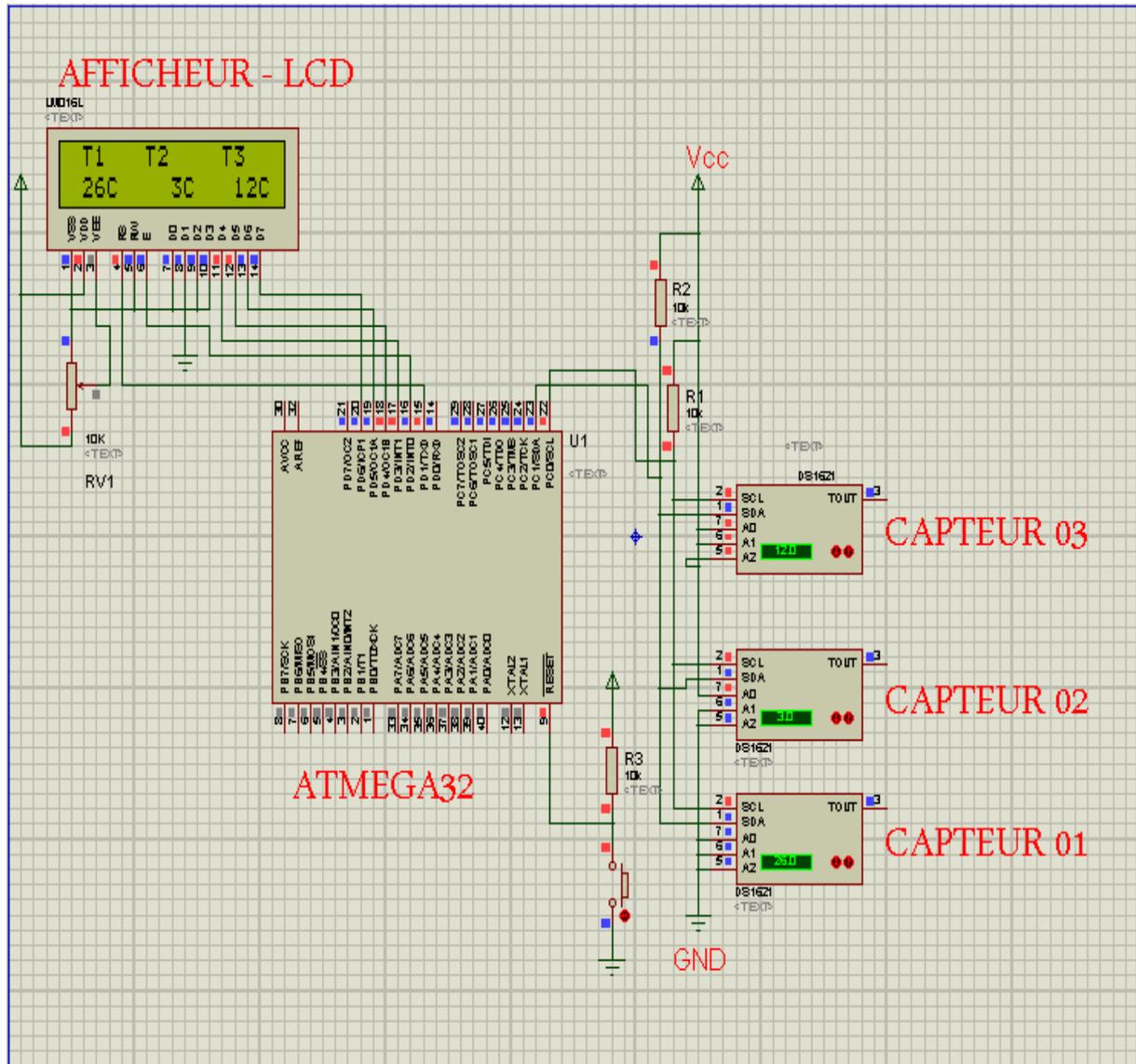


Figure IV.4: Le logiciel ISIS 7 en mode simulation du projet

#### IV.4.1 descriptif des composants utilisés :

##### IV.4.1.1 Unité de commande :

L'unité de contrôle est basé sur le ATMIGA 32 le schéma de cette unité est représenté en figure (II.6) du chapitre 2. Les ports de l'ATMIGA 32 seront configurés au besoin de notre application:

Le port C : est configuré comme sortie, les deux bits PC0/SCL, PC1/SDA utilisent pour communiquer avec capteur température DS1621 par bus I2C

Le port D : est configuré comme sortie et les six bits (PD1...PD6) restant commandent l'afficheur LCD.

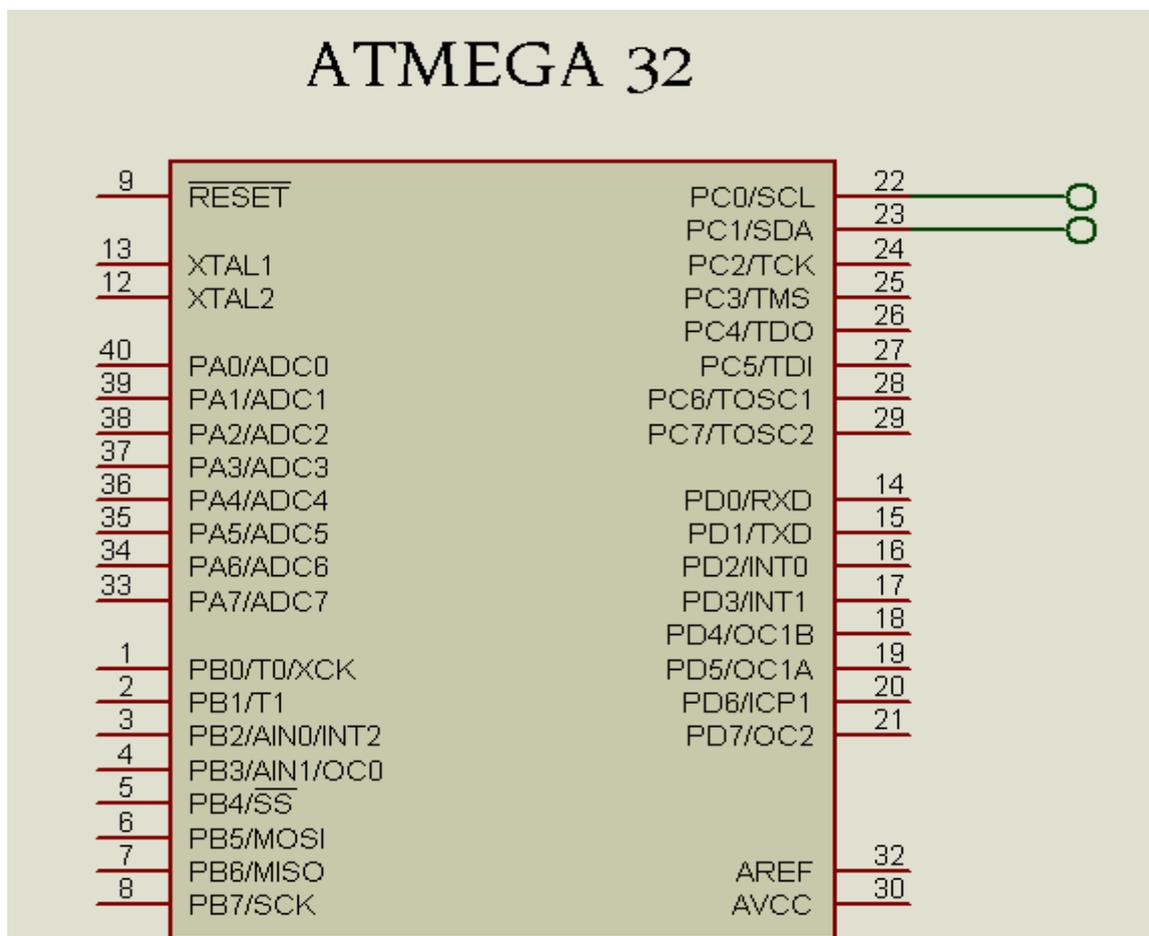


Figure IV.5: configuration le port c (SDA et SCL)

## IV.5 L'afficheur LCD:

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité.

### IV.5.1 Brochage LCD 2x16:

N° Broche	NOM	Description
1	VSS	Masse
2	VDD	Alimentation positive (V+)
3	VEE	Contraste réglable par potentiomètre *
4	RS	Sélection de registre (0= instruction; 1 = données)
5	R/W ou RD	Lecture ou écriture (1= lecture; 1=écriture)
6	E	Enable (validation, actif au niveau haut)
7	D0	Bit 0 du bus de données
8	D1	Bit 1 du bus de données
9	D2	Bit 2 du bus de données
10	D3	Bit 3 du bus de données
11	D4	Bit 4 du bus de données
12	D5	Bit 5 du bus de données
13	D6	Bit 6 du bus de données
14	D7	Bit 7 du bus de données

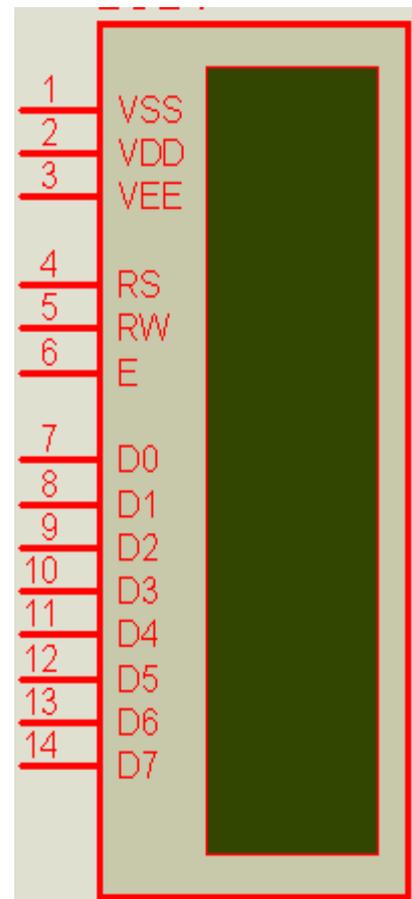


Figure IV.6: Brochage d'afficheur LCD

### IV.5.2 Mode 4 bits :

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur. Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (**D4 à D7**) de l'afficheur sont utilisées pour transmettre les données et les lire. Les 4 bits de poids faible (**D0 à D3**) sont alors connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur. Les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible. Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet.[15]

## IV.6 Le Capture de Temperature DS 1621:

### IV.6.1 Definition:

Le capteur rend une lecture numérique directe de la température mesurée possible, sans qu'un convertisseur analogique/numérique soit requis et il a déjà été calibré par le fabricant. Le capteur contient un thermomètre numérique et dispose en outre de fonctions comme une mémoire ou un thermostat numérique programmable.

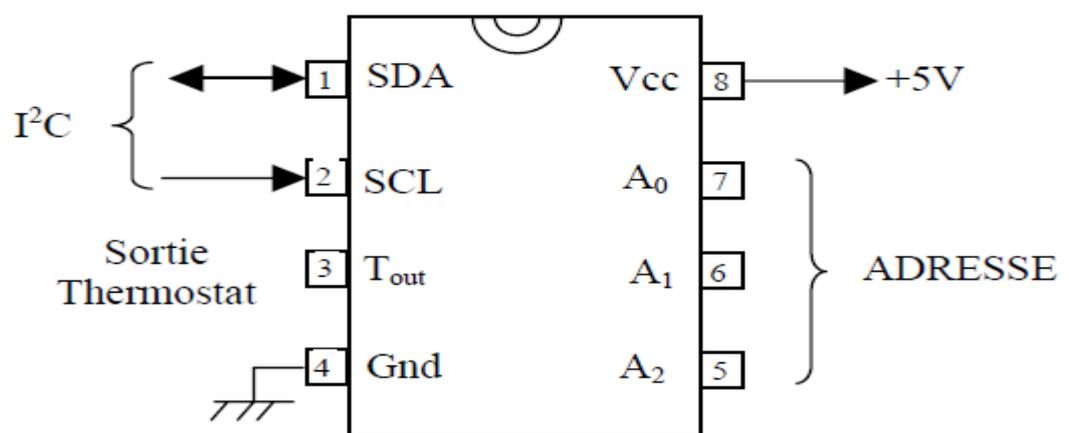
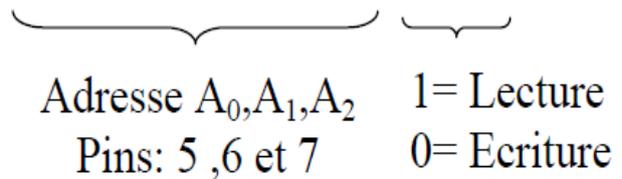
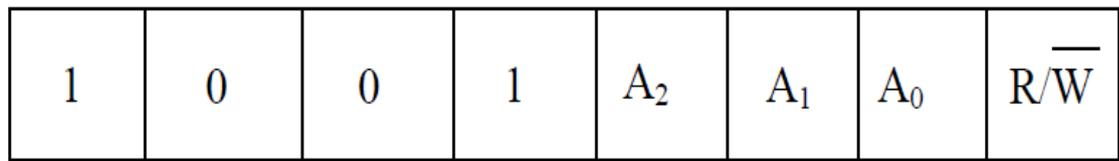


Figure IV.7:Brochage du capteur DS 1621

Tout dialogue du Maître avec un DS1621 esclave doit débiter par l'envoi d'un mot de contrôle.



**VI.6.2 Mot de contrôle :**



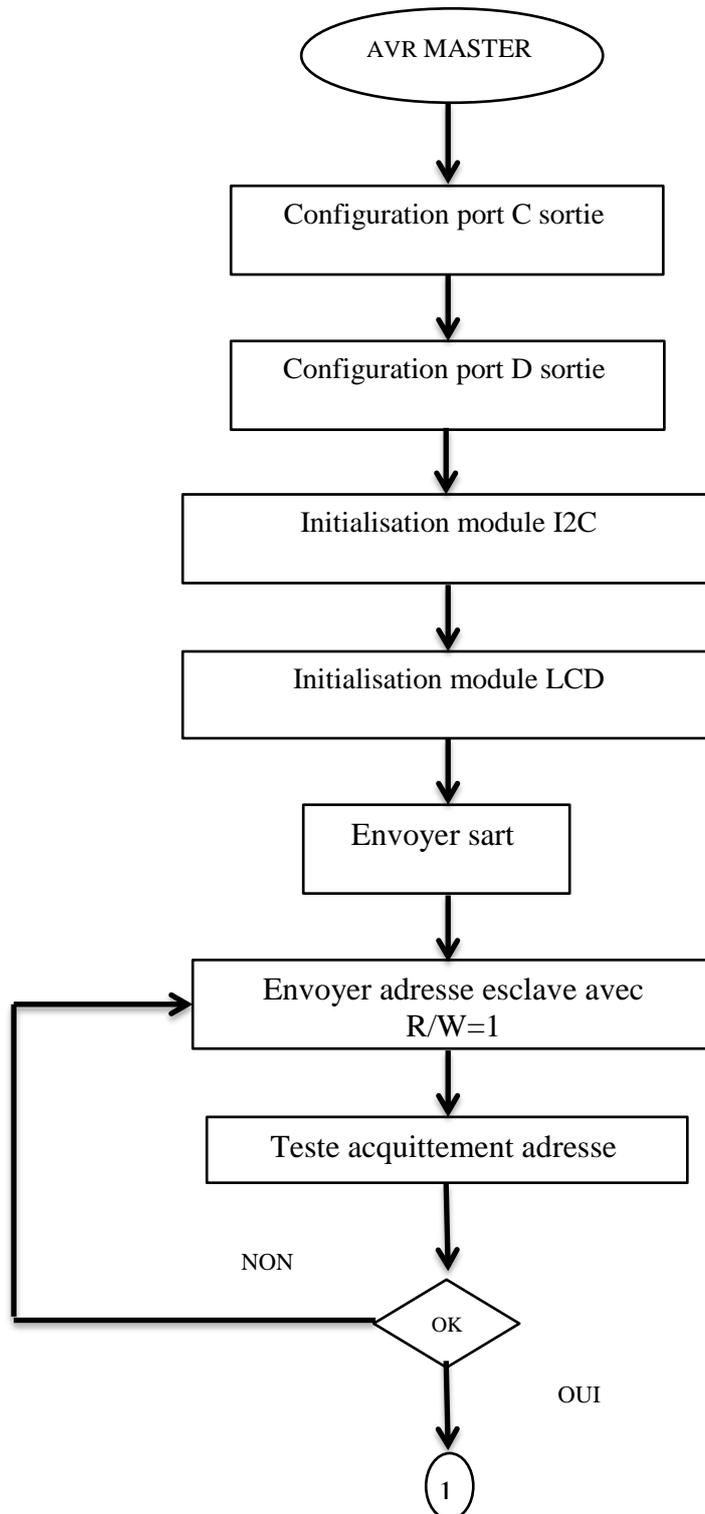
**VI.6.3 Commande :**

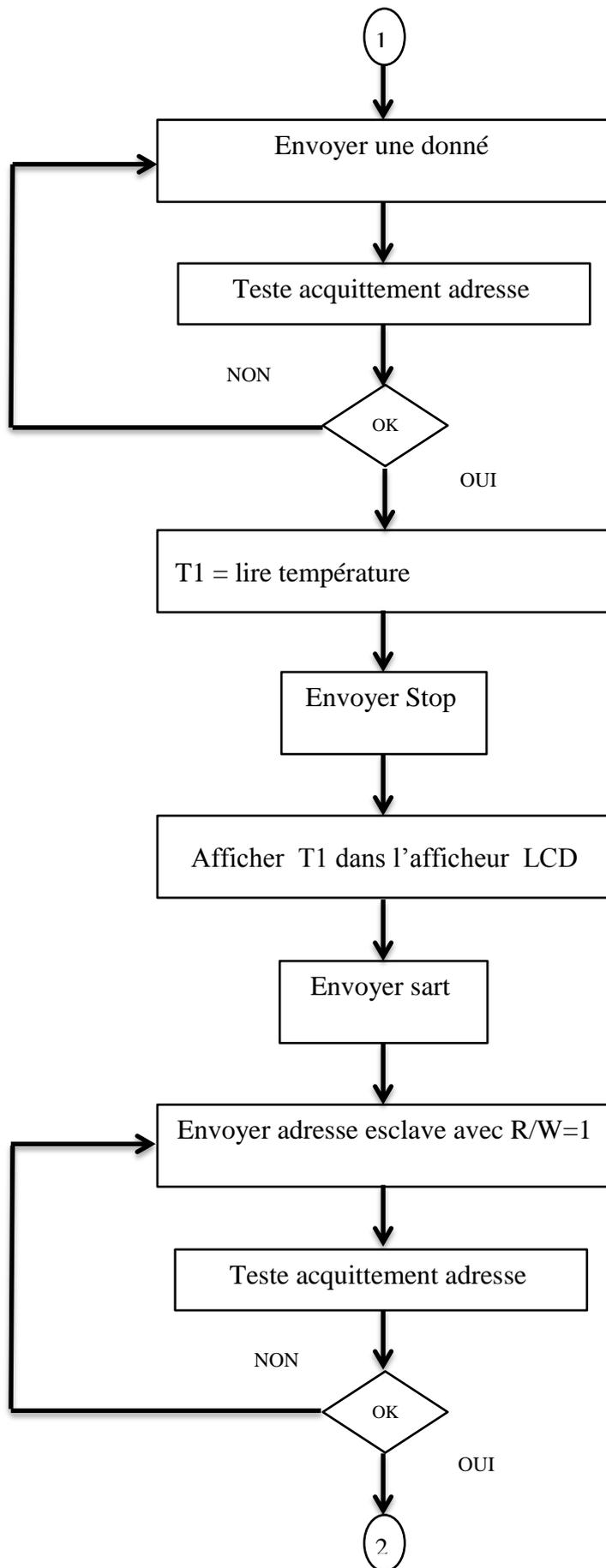
Le maître envoie ensuite une commande, suivie ou non de données.

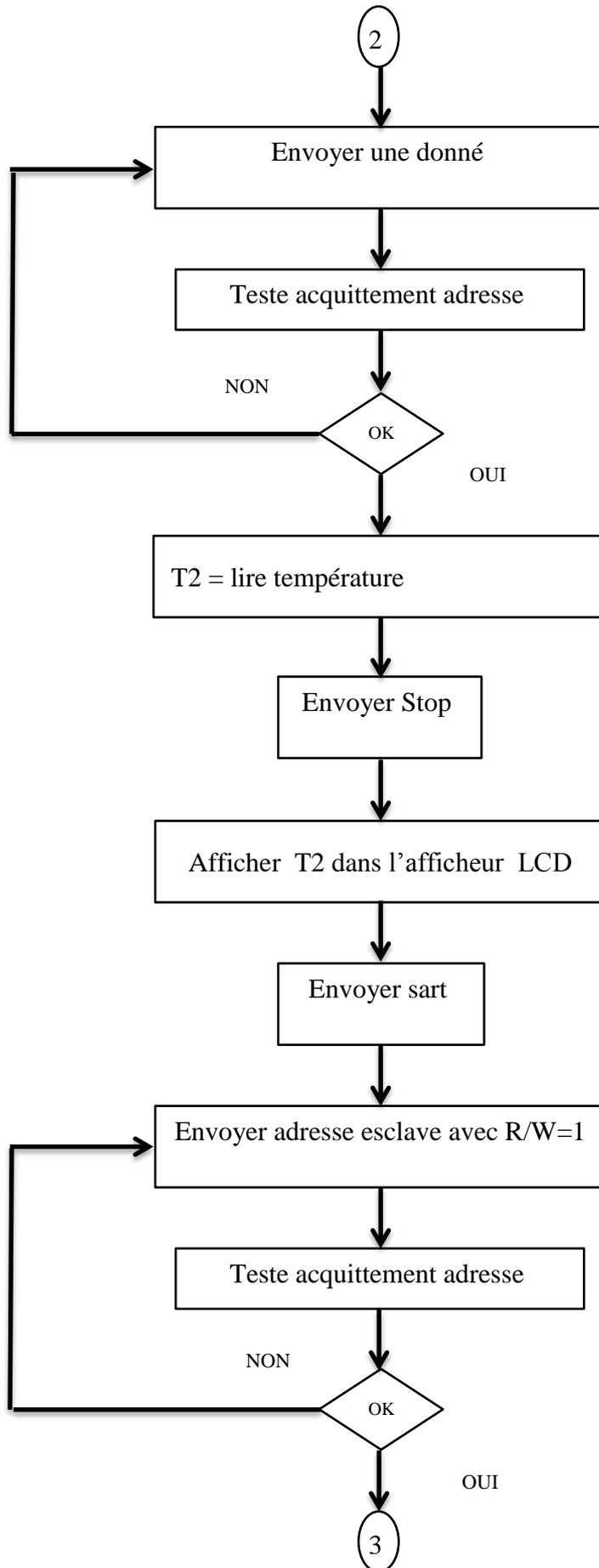
COMMANDE(en hexa)	DESCRIPTION
AC	Accès au registre de configuration (lecture ou écriture).
EE	Début de conversion.
22	Arrêt conversion.
AA	Lecture de la température. DS1621 renvoi 2 octets.
A1	Lecture ou écriture du seuil haut du thermostat : TH.
A2	Lecture ou écriture du seuil bas du thermostat : TL.

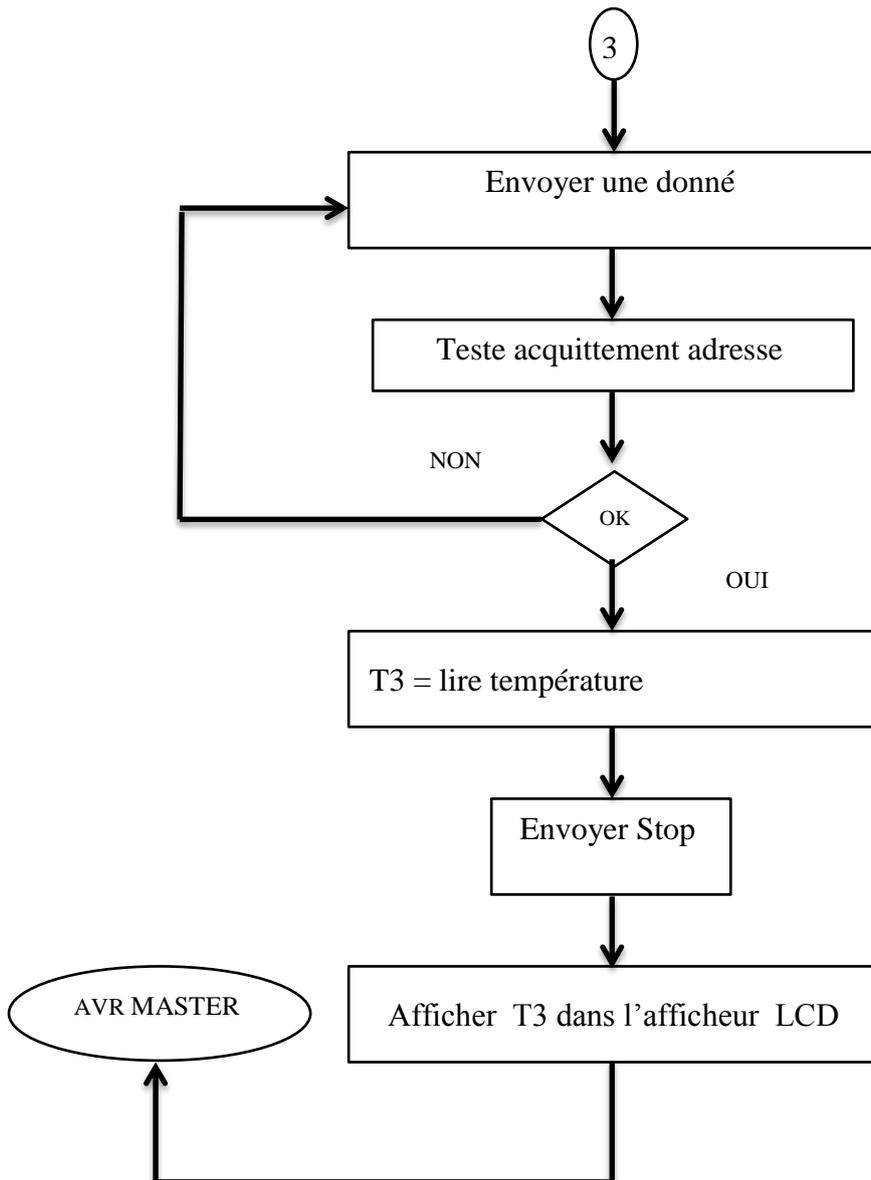
Les commandes AC, EE et 22 ne sont pas suivies de l'envoi de données. Par contre la commande AA est suivie de l'envoi par le DS1621 de 2 octets représentant la température. Les commandes A1 et A2 sont suivies elles aussi de 2 octets envoyés par le maître dans le cas d'une écriture des seuils du thermostat ou bien retournés par le DS1621 dans le cas d'une lecture.[16]

### IV.6 Organigramme de système :









### IV.7 Conclusion:

Dans le dernier chapitre nous pr sentons les mod les et r sultats de simulation en environnement   microcontr leur AVR ATMEGA 32 pour la mise en  uvre de notre application.

---

# Conclusion générale

---

### Conclusion générale

Notre projet de fin d'étude a pour but , l'étude et la réalisation d'un chaîne d'acquisition et d'affichage graphique et ou Alphanumérique de la température de plusieurs capteurs numériques de température . Ces capteurs sont reliés a des points dents d'un four de cuisson de briques par exemple pour optimiser la connectique des capteurs numériques ou CPU pilote le bus I2C est utilisé.

Le système à réaliser est considéré comme un system "enfouie " ou embarqué dans la chaîne de me Suse.

La chaîne a été simulée sans le logiciels "porteuse ISIS " le schéma a été valide conformément au résultat optimale.

Le programme est développé en langage C, compile pour l'atmega32 et également valider.

En conclusion ce projet a été bénéfique pour notre formation il nous a permis de :

- Comprendre la philosophie des systèmes embarquées, leur architecture et leur position Risc vis à vis des systèmes électroniques standards en termes d'utilisation de processeur.
- comprendre également les différentes architectures de processeur, les performances, les limites d'utilisation pour chaque type d'applications souhaitée.
- Programmer modestement en langage C la famille des microcontrôleurs AVR d'Atmel
- Connaitre la famille des microcontrôleurs AVR , connue par leur richesse en terme d'éléments d' E / S et possibilités de programmabilité et souplesse de développement et de mise en over pratique.

## *Bibliographie*

- [1] : Article: " Les systèmes embarqués " –  
([http://www.enseirb-matmeca.fr/cosynux/grand\\_public/embedded\\_intro\\_2006](http://www.enseirb-matmeca.fr/cosynux/grand_public/embedded_intro_2006)) - 04/03/2013
- [2] : " le langage C pour l'embarqué" –  
(<http://87.98.130.175/kadionik/systeme/langage-c-embarque>) - 05/03/2013
- [3] : thèse : Najat Demirel; Co-design d'un bloc PA-Antenne en technologie silicium pour application radar l'université bordeaux 1 École doctorale des Sciences Physiques et de l'Ingénieur; - ([http://tel.archives-ouvertes.fr/docs/00/58/60/71/PDF/Demirel\\_These](http://tel.archives-ouvertes.fr/docs/00/58/60/71/PDF/Demirel_These)) 2009/2010
- [4] : Article: " Architecture des DSP " - (<http://www.technologuepro.com/cours-dsp/chapitre-3-Architecture-des-DSP>) - 12/03/2013
- [5]: Architecture de Von Neumann et architecture de Harvard technique -  
(<http://www.scrigroup.com/limba/franceza/159/Architecture-de-Von-Neuman-et-41539.php>) - 15/03/2013
- [6]: "Architecture des ordinateurs" –  
(<http://www.elektronique.fr/documents-pour-le-site/Cours-II.pdf>) - 16/03/2013
- [7] : " Le micro-ordinateur" –  
(<http://www.mathinfo.univparis5.fr/~let/CoursHTML1/COURS/L'ORDI/L'ORDI%20annexes.pdf>) - 18/03/2013
- [8] : "Cours de processeur"  
(<http://perso.ibcp.fr/gilbert.deleage//Cours/processeur.pdf>) 30 - 32 - 20/03/2013
- [9] : "Qu'est-ce qu'un microcontrôleur ?" ([http://www.mon-club-elec.fr/pmwiki\\_mon\\_club\\_elec/pmwiki.php?n=MAIN.COMPRENDREQuezakoMicrocontrolleur](http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.COMPRENDREQuezakoMicrocontrolleur)) - 21/03/2013
- [10] : Microcontrôleur ATMELATMEGA -  
(<http://www.isetti.be/jlepot/labos2/files/AtMega32DocFr.pdf>) - 24/04/2013 - p43
- [11] : "Le bus I2C" -  
([http://artic.acbesancon.fr/genie\\_electronique/telech/sonde/doc\\_bus\\_i2c.pdf](http://artic.acbesancon.fr/genie_electronique/telech/sonde/doc_bus_i2c.pdf)) - 26/04/2013
- [12] : "Le bus I2C" -  
(<file:///C:/Users/ha/Desktop/i2c%20&/Mise%20en%20oeuvre%20du%20bus%20I2C%20avec%20le%20PIC%2016F876A.htm>) - 28/04/2013



[13] : "L'EDITEUR ISIS Version 6.5 SP5" - ([www.gesi.asso.fr/web/ressources-ressources-dl.php?fichier=46](http://www.gesi.asso.fr/web/ressources-ressources-dl.php?fichier=46)) - 04/05/2013

[14] : " SAISIE DE SCHEMA ISIS 6 ?" - ([http://www.louis-armand-paris.fr/elec/claroline/document/goto/?url=%2FTP\\_N%B01%2F1\\_Cours\\_ISIS.doc](http://www.louis-armand-paris.fr/elec/claroline/document/goto/?url=%2FTP_N%B01%2F1_Cours_ISIS.doc)) - 06/04/2013

[15] : Les Afficheurs LCD alphanumérique  
file:///H:/m%C3%A9moire/Les%20Afficheurs%20LCD%20alphanum%C3%A9rique.htm-  
08/05/2013

[16] : DS1621 <http://pdfserv.maximintegrated.com/en/ds/DS1621.pdf>- 10/05/2013