

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITE MOULOU D MAMMERI TIZI-OUZOU

Faculté de Génie Electrique et Informatique  
Département d'Electronique



Mémoire de fin d'étude en vue de l'obtention du diplôme  
Master en Electronique

Option : Electronique Industrielle

Thème :

**Etude et réalisation d'une alarme de  
température à base d'une carte arduino**

Présenté par :  
AMMARKHODJA Nassim

Encadré par :  
Mr Y.ATTAF

Promotion 2017/2018



# *Remerciements*

Au terme de ce projet de fin de cycle, je tiens d'abord à remercier très chaleureusement **Mr.ATTAF Youcef** qui m'a permis de bénéficier de son encadrement. Les conseils qu'il m'a prodigué, la patience, la confiance qu'il m'a témoigné ont été déterminants dans la réalisation de ce travail de recherche.

Et je tiens à remercier les membres du jury de leur présence et d'avoir accepté d'évaluer et examiner mon travail.

Un grand merci à mes amis et à ma famille, en particulier à mes parents pour m'avoir soutenu et aidé tout au long de mes études.

Enfin, j'adresse mes sincères remerciements à tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

<b>Figure</b>	<b>Nom de figure</b>	<b>Numéro de page</b>
Figure(I.1)	Courbe caractérisant la CTP et CTN	04
Figure(I.2)	Représentation schématique	04
Figure (I.3)	Thermocouple (principe)	05
Figure (I.4)	La relation force électromotrice /température de n'importe quel couple peut être déterminée.	05
Figure (I.5)	A courant constant, la tension V est linéaire en fonction de la température	06
Figure (I.6)	Exemple de structure de chaîne (cas général)	07
Figure (I.7)	Une chaîne de mesure informatisée	08
Figure (I.8)	Structure générale d'un capteur intelligent	09
Figure (I.9)	Compensation de température + unité de calcul	09
Figure (I.10)	Unité de calcul + fonctionnalités	10
Figure (I.11)	Brochage du lm35	13
Figure(I.12)	Calibration du lm35	13
Figure (II.1)	La carte Arduino UNO	17
Figure (II.2)	Microcontrôleur ATmega328	19
Figure (II.3)	Brochage de l atmega328	20
Figure (II.4)	Constitution de la carte Arduino UNO	23
Figure (II.5)	Interface IDE Arduino	24
Figure( II.6)	Paramétrage de la carte	25
Figure (II.7)	Les étapes de téléchargement du code	31
Figure (II.8)	Type de modules Bluetooth	32
Figure (II.9)	Module shield wifi	33
Figure (II.10)	Module XBee	33
Figure (II.11)	Capteur Arduino	34
Figure (II.12)	Capteur de température lm35	34
Figure (II.13)	Moteurs électriques	35

Figure (II.14)	Brochage d'un Afficheurs LCD	36
Figure (II.15)	Relais	36
Figure (II.16)	Buzzer tone	37
Figure (III.1)	Schéma synoptique du dispositif	40
Figure (III.2)	La carte réalisée sous ISIS-PROTEUS	41
Figure (III.3)	Le chemin du fichier de code HEX de notre programme	42

**Listes des figures**

**I**

**Sommaire**

## **Introduction générale**

### **Chapitre I**

#### **Généralités sur la mesure de la température**

I-introduction	01
II-les échelles de mesures de la température	02
II-1-le fahrenheit	02
II-2-le Celsius	02
II-3-le kelvin	02
II-4-Les méthodes utilisées dans la mesure	02
II-4-1-méthodes mécaniques	02
II-4-2-méthodes optiques	02
II-4-3-méthodes électriques	02
III-les capteurs de température	03
III-1-les capteurs actifs	03
III-1-1-la thermoélectricité	03
III-2-les capteurs passifs	03
III-3-les capteurs de température les plus connus	04
III-3-1-les thermistances	04
III-3-1-1-les types de thermistances	04
III-3-1-2-les caractéristiques d'une thermistance de type CTN et CTP	04

III-3-1-3-representation schématique	04
III-3-2-les thermocouples	05
III-3-3-les capteurs de température à semi-conducteur	06
III-3-4-les pyromètres optique	07
IV-chaine d'acquisition des données	07
IV-1-les chaines de mesures informatisées	08
IV-1-1 caractéristique d'une chaine de mesure informatisée	08
V-les capteurs intelligents	09
V-1-structure d'un capteur intelligent	09
V-1-1-structure minimale	09
V-1-2-structure complexe	10
V-2-fonctionnalités des capteurs intelligents	10
V-2-1-1'auto-adaptabilité	10
V-2-2-remplacement des données manquantes	10
V-2-3-précision et validation des mesures	10
V-2-4-traitement de signal	11
V-2-5-auto diagnostic	11
VI-définition du lm35	11
VI-1-brochage du lm35	12
VI-2-calibrage du lm35	12
VII-conclusion	13

## Chapitre II

# Le dispositif programmable Arduino

I Introduction	14
II Définition du module Arduino	14
II.1 Les gammes de la carte Arduino	15
II.2 Pourquoi Arduino UNO	17
II.3 La constitution de la carte Arduino UNO	18
II.3.1 Partie matérielle	18
II.3.1.1 Le MicrocontrôleurATMega328	18
II.3.1.2 Les sources de l'Alimentation de la carte	20
II.3.1.3 Les entrées & sorties	20
II.5.3.4 Les ports de communications	22
II.3.2 Partie programme	23
II.3.2.1 l'environnement de la programmation	23
II.3.2.2 Structure générale du programme (IDE Arduino)	24
II.3.2.3 Injection du programme	25
II.3.2.4 Description du programme	25
II.3.2.5 Les étapes de téléchargement du programme	31
II.4 Les Accessoires de la carte Arduino	32
II.4.1 Communication	32
II.4.1.1 Le module Arduino Bluetooth	32
II.4.1.2 Le module shield Arduino Wifi	33
II.4.1.3 Le Module XBee	33
II.4.2 Les capteurs	34

II.4.3 Les drivers	35
II.4.3.1 Les moteurs électriques	35
II.4.3.2 Les afficheurs LCD	35
II.4.3.3 Le relais	36
II.4.3.4 Le buzzer	37
II.5 Conclusion	38

## **Chapitre III**

### **Réalisation du dispositif expérimental**

I-introduction	39
II-les différentes étapes de réalisation	39
III-schéma synoptique général	39
IV-la réalisation virtuelle PROTEUS	40
V-explication et démarche	42
VI-composants utilisés	43
VII-conclusion	44

## **Conclusion générale**

## **Annexes**

Annexe A

Annexe B

Annexe C

## **Bibliographie**

# **Introduction générale**

Le développement industriel nécessite une instrumentation ,cette instrumentation est souvent coûteuse puisqu' on est dans l'incertitude des résultats ,mais grâce à la maîtrise des procédés de fabrication, on obtient quand même des produits de haute fiabilités, qui nous facilitent les tâches dans tous les domaines qui ont connu l'avantage du développement industriel, surtout en ce qui concerne la fabrication des semi-conducteurs, le champ d'application de ces derniers et les dispositifs électroniques en général est très vaste .

Dans la vie de tous les jours et surtout dans le monde industriels, le suivi et le contrôle des variations de la température est vitale, et cela au risque de surchauffe qui engendrerait un endommagement du matériel et des machines, ou tout simplement des dégâts domestiques.

Et c'est pour cela que notre travail se concentre sur l'utilisation d'un capteur, qui va capter la température et ses variations, avec bien sûr sa liaison avec un système ou une carte de commande (carte d'interface) tel que l'arduino, ainsi qu'un buzzer qui va jouer le rôle d'alarme.

Dans ce projet trois objectifs ont été visés :

- Le premier consiste à définir la température et les différentes unités de mesures ainsi que les différents types de capteurs.
- Le deuxième est de regrouper suffisamment d'informations sur une grande catégorie de cartes d'interfaçage (Arduino) : son langage de programmation, sa construction, son principe de fonctionnement.
- Le troisième consiste à réaliser un câblage électrique capable d'exécuter une action entre un capteur et une carte d'interfaçage (Arduino) en expliquant les différents blocs de sa construction.

Le premier chapitre sera une généralité sur les températures, les différentes unités de mesures, ainsi que les types de capteurs.

Le deuxième chapitre sera consacré à une étude approfondie sur les cartes d'interface tel que l'Arduino, puis, on mettra la lumière sur un modèle de base qui est (Arduino UNO) sa construction son environnement de programmation et son principe de fonctionnement afin de simplifier son utilisation.

Le troisième chapitre sera consacré à l'étude et la réalisation du dispositif expérimental ainsi que la description de chaque bloc du circuit.

Enfin, on terminera avec une conclusion générale qui résumera l'intérêt de notre étude : les différents résultats obtenus expérimentalement seront en annexe, donnant ainsi un aperçu sur les performances des cartes d'interfaçage, et une idée sur les problèmes à résoudre ultérieurement.

# **Chapitre I**

## **Généralités sur la mesure**

### **de la température**

## **I-Introduction :**

La notion de température à son origine réside dans la sensation qui nous fait dire qu'un corps est chaud ou froid.

Physiquement parlant elle est définie comme étant le facteur d'intensité de l'énergie thermique (qui est une forme dégradée de l'énergie la plus répandue dans le monde naturel et industriel).

La température est une grandeur intensive c.à.d. dépourvue des caractères d'additivité et d'associativité qui rendent proprement mesurable tous les facteurs d'extension (tels que longueurs, volume, masse, quantité d'électricité).

Rappelons que l'énergie thermique d'un corps quelconque est l'énergie cinétique de ses constituants dotés de liberté de mouvements, tels que molécules, atomes, électrons libres, due à l'agitation thermique de ces mêmes particules (par exemple, quand l'agitation est faible, l'objet est froid au toucher).

La température est un paramètre mesuré depuis l'Antiquité, et pour cette mesure des différentes méthodes ont été utilisées. Les systèmes de mesure de la température ont beaucoup évolué au cours des siècles pour s'adapter à la demande du marché.

## **II-Les échelles de mesure de la température :**

### **II.1-Le fahrenheit :**

La première échelle pour mesure de température a été proposé par le physicien Allemand **DANIEL GABRIEL FAHRENHEIT**, depuis, diverses échelles ont été proposées, dont l'échelle centésimale.

### **II.2-Le Celsius :**

Conçue par l'astronome suédois **Anders Celsius** en 1743 : 0 °C correspond au point de fusion de la glace et 100 ° C au point d'ébullition de l'eau sous une pression atmosphérique normale.

$$0^{\circ} \text{C} = 32 \text{ F} ; 100^{\circ} \text{C} = 212 \text{ F}.$$

### **II.3-Le Kelvin :**

L'unité de température la plus utilisées est le Kelvin (K).Elle mesure la température absolue, c'est-à-dire la température qui ne dépend pas de certaines propriétés du corps. Elle fut inventée par le mathématicien et physicien britannique sir **William Thomson Kelvin** au XIX siècle.La plus petite température que l'on puisse théoriquement atteindre est le zéro absolu 0K soit  $-273.15^{\circ} \text{C}$  ( $T \text{ Celsius} = T \text{ Kelvin} - 273.15$ ).Les écarts de température sont identiques sur les échelles Kelvin et Celsius.

## **II.4-Méthodes utilisées dans la mesure :**

### **II.4.1-Méthodes mécanique :**

Fondées sur la dilatation d'un solide, d'un liquide, d'un gaz à pression constante.

### **II.4.2-Méthodes optiques :**

Fondées sur la répartition spectrale du rayonnement.

### **II.4.3-Méthodes électriques :**

Reposant sur la variation thermique de la résistance.[1]

La suite de ce chapitre sera consacrée pour les capteurs électriques afin de s'approcher un peu de ce genre qui nous intéresse dans le domaine de l'électronique.

### **III-Les capteurs de températures :**

Les capteurs électriques ont l'avantage d'une plus grande souplesse d'emploi (information, transmission, enregistrement) tout en gardant une précision suffisante pour les emplois industriels et beaucoup d'emplois de laboratoire.

On peut décomposer les capteurs électriques de température en deux catégories :

- Les capteurs actifs, à couple thermoélectrique
- Les capteurs passifs, à résistance ou thermistance.

#### **III.1-Les capteurs actifs :**

Fonctionnent en générateur, un capteur actif est généralement fondé dans son principe sur un effet physique qui assure la conversion en énergie électrique de la forme d'énergie propre à la température à prélever.

Le signal de sorti d'un capteur actif peut être un courant, une tension ou une quantité de charge en fonction de l'intensité et de la température mesurée.

L'effet physique exploité pour la mesure de la température par les capteurs actifs et la thermoélectricité.

##### **III.1.1-La thermoélectricité :**

C'est la force électromotrice d'origine thermique obtenu à des jonctions de deux conducteurs de nature différente à des températures différentes.

#### **III.2. Les capteurs passifs :**

Il s'agit généralement d'impédances dont l'un des paramètres déterminant est sensible à la température, et dans l'expression littérale de cette impédance on trouve des termes lié à :

1. La géométrie de ce capteur ou de ses dimensions
2. Les propriétés électriques des matériaux utilisés comme la résistivité, la perméabilité
3. Magnétique, ou des différentes constantes.

L'effet physique exploité pour la mesure de la température par les capteurs passifs est la résistivité, et aussi la constante diélectrique pour les très basses températures.[2]

#### **III.3-Les capteurs de températures les plus connus :**

##### **III.3.1-Les thermistances :**

Pour une thermistance de composition donnée, à une température donnée, elle traduit cette température à une résistance spécifique. Leurs constitution est un alliage de semi-conducteurs

et d'oxydes métalliques (fer, titane) que l'on appelle oxyde céramiques, elles sont utilisées généralement pour mesurer des températures entre  $-46\text{ C}$  et  $150^{\circ}\text{ C}$ . Les propriétés de ces thermistances dépendent du choix du semi-conducteur utilisé ainsi que du dosage des oxydes. Les thermistances font parties des résistances non linéaire. On en rencontre de deux types :

### III.3.1.1-Les types de thermistances :

- **La CTN, coefficient de température négatif (NTC thermistor) :**

Les CTN sont des thermistances dont la résistance diminue de façon uniforme avec la température .

- **La CTP, coefficient de température positif (PTC thermistor) :**

Les CTP sont des thermistances dont la résistance augmente fortement avec la température dans la plage( $0^{\circ}\text{C}$  et  $100^{\circ}\text{C}$ ) mais diminue en dehors de cette zone .

### III.3.1.2-Les caractéristiques d'une thermistance de type CTP et CTN :

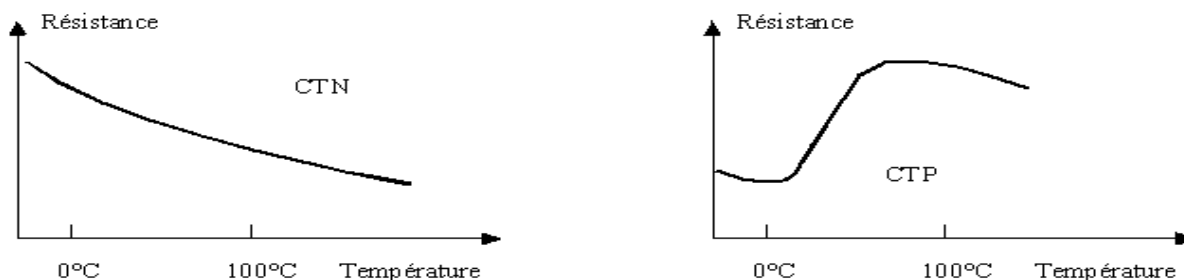


Figure I.1- courbes caractérisant la CTP et CTN.

### III.3.1.3-Représentation schématique :

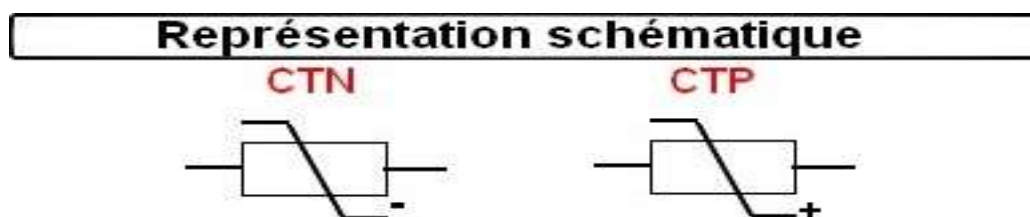


Figure I.2- représentation schématique.

Les paramètres essentiels d'une thermistance sont :

- La valeur de sa résistance.
- Sa sensibilité thermique ou coefficient de température .
- Sa stabilité (fourni par le constructeur).

Avantages : temps de réponse rapide, moins chers.

Inconvénients : loi non linéaire des caractéristiques dans les séries, sensible à l'auto échauffement et à la variation des résistances de connexion.

### III.3.2-Les thermocouples :

Principe : effet thermoélectricité (capteur actif)

Un circuit fermé, constitué de deux conducteurs A et B de nature différentes dont les jonctions sont à des températures  $T_1$  et  $T_2$  différentes, le thermocouple peut être considéré comme le siège d'une force électromotrice dite de Seebeck  $V_{AB}$ ,  $V_{AB}$  dépend de la nature des deux conducteurs et des températures  $T_1$  et  $T_2$ .

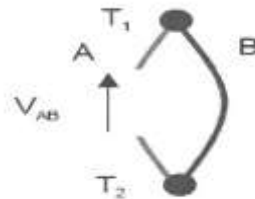


Figure I.3 : thermocouple (principe)

Les pouvoirs thermoélectriques des métaux et alliages (relation  $V=f(t)$ ) sont définis dans des tables par rapport à un métal de référence (Pb ou Pt) et par rapport à  $0^\circ\text{C}$

- Si deux jonctions à la température  $T_1$  et  $T_2$  produisent une tension  $V_1$ , et la température  $T_2$  et  $T_3$  produisent une tension  $V_2$  alors avec  $T_1$  et  $T_3$  on a une tension  $V_3=V_1+V_2$

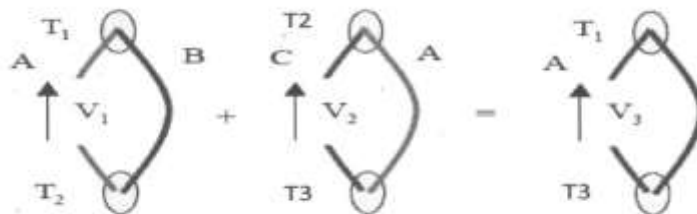


Figure I.2 : la relation force électromotrice /température de n'importe quel couple peut être déterminée.

### III.3.3-Les capteurs de température à semi-conducteur :

Une jonction silicium diode ou transistor alimenté par un courant constant donne une tension proportionnelle à la température .on utilise cette propriété pour fabriquer des capteurs. En générale, on ajoute un circuit intégré pour conditionné le capteur .il existe aussi des circuits intégrés permettant d'enregistrer la température sur une période longue.

La tension aux bornes du semi-conducteur (formant une diode ou un transistor) et le courant qui le traverse dépendent de la température :



Figure I.4 : A courant constant, la tension  $V$  est linéaire en fonction de la température

Il existe aussi des mesures de températures basées sur la mesure du bruit de fond, qui représente l'agitation thermique des électrons dans un matériau.

Avantage des capteurs à semi-conducteur :

- Les semi-conducteurs ont un cout faible.
- La simplicité de mise en œuvre.
- Bonne linéarité.[3]

### III.3.4-les pyromètres optiques :

La pyrométrie optique est une méthode de mesure de la température, basée sur la relation entre la température d'un corps et le rayonnement optique (infrarouge ou visible) que ce corps émet. L'intérêt de la pyrométrie optique est de permettre la détermination d'une température sans contact avec l'objet.

Les pyromètres optiques sont utilisés principalement dans les conditions ou il est impossible de traiter avec les capteurs classiques comme :

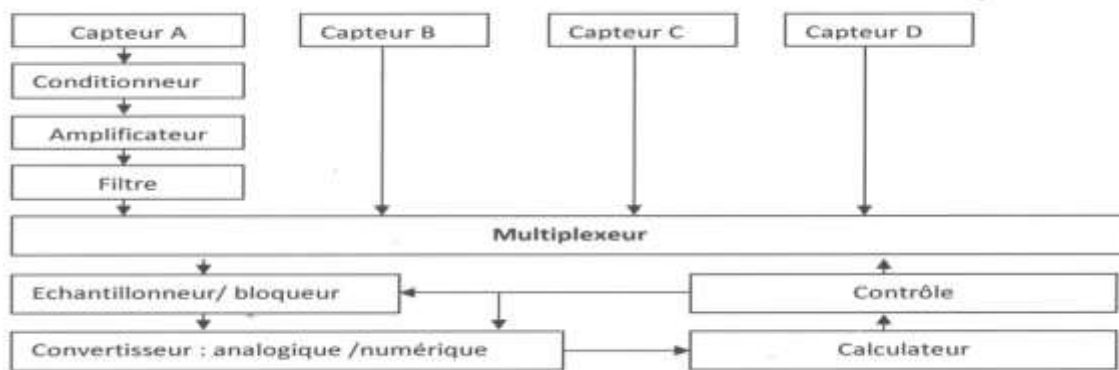
- Température très élevée (supérieur à 2000 °C)
- Mesure à grande distance.
- Environnement très agressif.
- Pièce en mouvement.

### IV-Chaines d'acquisition des données :

Un capteur généralement n'est pas mis en œuvre tout seul, mais il doit être intégré dans une chaîne qui est souvent constitué de trois parties :

- Acquisition des données (analogique par) :
  - Les capteurs
  - Les conditionneurs
  - Les amplificateurs
  - Multiplexage

- Transformation des données (généralement c'est la conversion analogique en numérique)
- Traitement des données (calculateur) [4]



FigureI.5 : exemple de structure de chaîne (cas général)

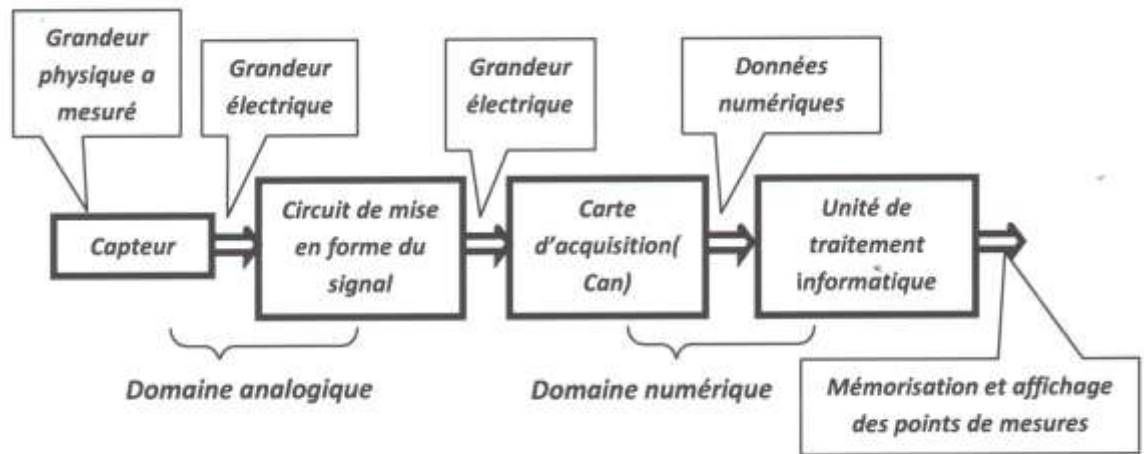
#### IV.1-Les chaînes de mesures informatisées :

De nos jours, avec les possibilités offertes par l'électronique et l'informatique, les capteurs délivrent des signaux électriques, et la quasi-totalité des chaînes de mesures sont des chaînes électroniques et informatiques.

##### IV.1.1-caractéristiques d'une chaîne de mesure informatisée :

La structure de base d'une chaîne de mesure informatisée comprend au minimum quatre étages :

- Un capteur sensible aux variations d'une grandeur physique et qui à partir de ces variations, délivre un signal électrique.
- Un conditionneur de signal dont le rôle principal est l'amplification du signal délivré par le capteur pour lui donner un niveau compatible avec l'unité de numérisation ; cet étage peut parfois intégrer un filtre qui réduit les perturbations présentes sur le signal.
- Unité de numérisation qui va échantillonner le signal à intervalles réguliers et affecter un nombre à chaque point d'échantillonnage.
- Une unité de traitement informatique peut exploiter maintenant les mesures qui sont une suite de nombres binaires (enregistrement, affichage de courbe, traitement mathématique, transmission des données...)

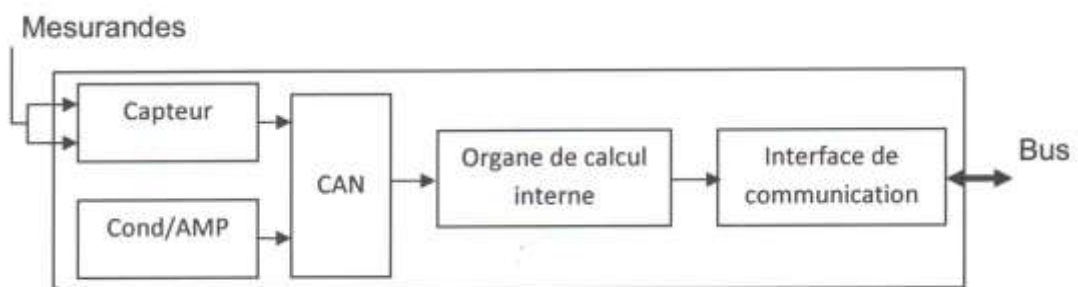


FigureI.6 : une chaîne de mesure informatisée

### V-Les capteurs intelligents :

On peut considérer un capteur intelligent comme un système qui se compose de plusieurs sous-systèmes, dont le rôle de chacun est bien distinct, avec l'intégration de tous ces systèmes dans un seul et unique boîtier. Un capteur intelligent se compose principalement de :

- Un ou plusieurs capteurs (comme élément intégré)
- Conditionneurs associé.
- Organe de calcul interne.
- Interface de communication.



FigureI.7 : structure générale d'un capteur intelligent

#### V.1-Structure d'un capteur intelligent :

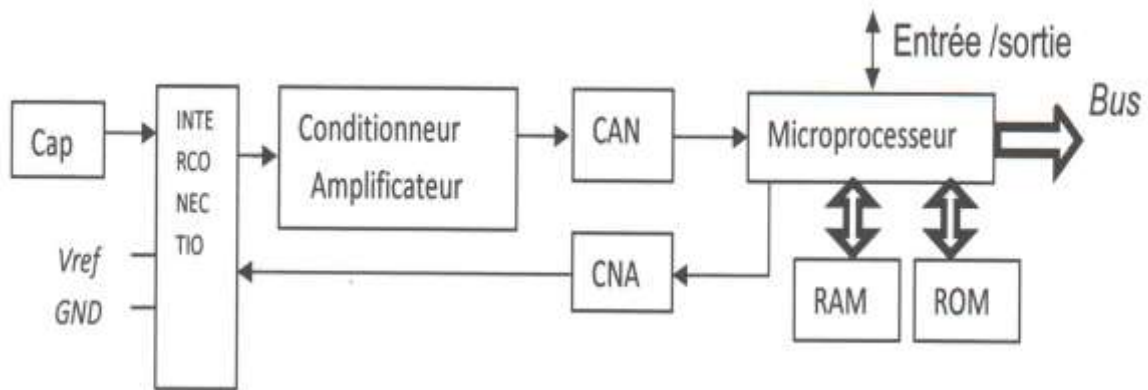
Un capteur intelligent a une structure très variable et dépend fortement des technologies utilisées.

### V.1.1-Structure minimale :



FigureI.8 : compensation de température + unité de calcul

### V.1.2-Structure complexe :



FigureI.9 : unité de calcul + fonctionnalités

## V.2-fonctionnalités des capteurs intelligents :

### V.2.1-L'auto adaptabilité :

Capacité du capteur intelligent à s'adapter au signal mesuré.

Exemples :

- Amplificateur gain variable.
- Filtre à fréquence de coupure variable.

### V.2.2-Remplacement des données manquantes :

Correction des défaillances ponctuelles du capteur =>technique permettant d'estimer les données manquantes à partir des données disponibles.

### V.2.3-Précision et validation des mesures :

Précision : prise en compte et compensation des grandeurs d'influences (température, pression ...)

Validation : évaluation de la qualité de mesure.

#### **V.2.4-Traitement de signal :**

Mise en place à proximité de la source de données d'un système de traitement de l'information et filtrage des résultats pour n'obtenir que l'information utile.

#### **V.2.5-Auto-diagnostic :**

La capacité d'un capteur d'effectuer l'évaluation de son état de fonctionnement et de diagnostiquer l'élément éventuellement en dysfonctionnement. [5]

Dans le chapitre qui suit, on a utilisé un capteur de température qui est le **lm35** .

#### **VI-Définition du lm 35 :**

Le capteur de température LM35 est un capteur analogique de température fabriqué par Texas Instruments. Il est extrêmement populaire en électronique, car précis, peu coûteux, très simple d'utilisation et d'une fiabilité à toute épreuve.

Le capteur de température LM35 est capable de mesurer des températures allant de  $-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$  dans sa version la plus précise et avec le montage adéquat, de quoi mesurer n'importe quelle température.

*N.B. Les versions grand public sont capables de mesurer des températures comprises entre  $-40^{\circ}\text{C}$  et  $+110^{\circ}\text{C}$ .*

La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de  $+10\text{mV}$ .

Le capteur LM35 supporte des températures assez extrêmes (jusqu'à  $-55^{\circ}\text{C}$  /  $+150^{\circ}\text{C}$ ), mais il n'en est pas de même pour les cartes Arduino.

Si vous soumettez une carte Arduino "classique" à ces températures extrêmes, elle va purement et simplement cesser de fonctionner ou se dégrader très rapidement. Cela est valable pour les cartes Arduino, mais aussi pour tous autres circuits électroniques fabriqués suivant les standards "grand public" (et non "industriels").

Pour information, les gammes de température classique en électronique sont les suivantes :

Grand public :  $0^{\circ}\text{C} \sim 70^{\circ}\text{C}$

Industrie :  $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$

Militaire :  $-55^{\circ}\text{C} \sim 125^{\circ}\text{C}$

Une des grandes forces du capteur LM35, qui fait sa popularité, c'est sa pré calibration en sortie d'usine. Tous les capteurs LM35 sont calibrés en degré Celsius lors de la fabrication. Cela signifie que vous n'avez absolument rien à faire pour calibrer le capteur, il l'est déjà au moment où vous le sortez de son sachet !

La précision garantie par le fabricant est de  $\pm 1^{\circ}\text{C}$  à  $25^{\circ}\text{C}$  et  $\pm 1.5^{\circ}\text{C}$  à  $-55^{\circ}\text{C}$  ou  $+150^{\circ}\text{C}$  pour la version la moins précise, ce qui est largement suffisant pour la plupart des applications. La version plus précise du LM35 (nommée "LM35A") a une précision garantie de  $\pm 0.5^{\circ}\text{C}$  à  $25^{\circ}\text{C}$  et  $\pm 1^{\circ}\text{C}$  à  $-55^{\circ}\text{C}$  ou  $+150^{\circ}\text{C}$ .

Une autre des grandes forces du capteur LM35, c'est sa linéarité exemplaire : moins de  $1^{\circ}\text{C}$  d'erreur sur la plage complète de  $-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$ . Comme chaque degré Celsius correspond à 10mV (soit 0.01 volt) et que la sortie du capteur est (quasi) parfaitement linéaire, convertir une mesure en température se résume à faire un bête produit en croix.

Pour en finir avec les compliments, le capteur LM35 fonctionne avec n'importe quelle tension d'alimentation comprise entre 4 volts et 30 volts, ce qui permet de l'utiliser dans virtuellement n'importe quel montage numérique ou analogique.

Passons maintenant aux points négatifs, ceux qui fâchent. Le capteur LM35 ne fonctionne pas en dessous de 4 volts, donc oubliez l'utilisation d'un LM35 avec des cartes Arduino 3.3 volts (Due, Zéro, etc.), sans alimentation 5 volts externe, ça ne marche pas.

*N.B. Le capteur LM35 peut être utilisé avec une entrée analogique 3v3, car  $150^{\circ}\text{C} = 1,5\text{V}$ , cependant, une alimentation 5v reste obligatoire pour que le capteur fonctionne.*

Autre souci, le capteur LM35 nécessite une alimentation négative pour mesurer des températures en dessous de  $0^{\circ}\text{C}$ . La sortie est proportionnelle à la température, par conséquent  $-10^{\circ}\text{C}$  équivaut à -0,1 volt. De fait, avec une simple alimentation 5v, comme celle des cartes Arduino, il n'est possible de mesurer que des températures positives.[3]

## VI.1-Brochage du lm 35 :

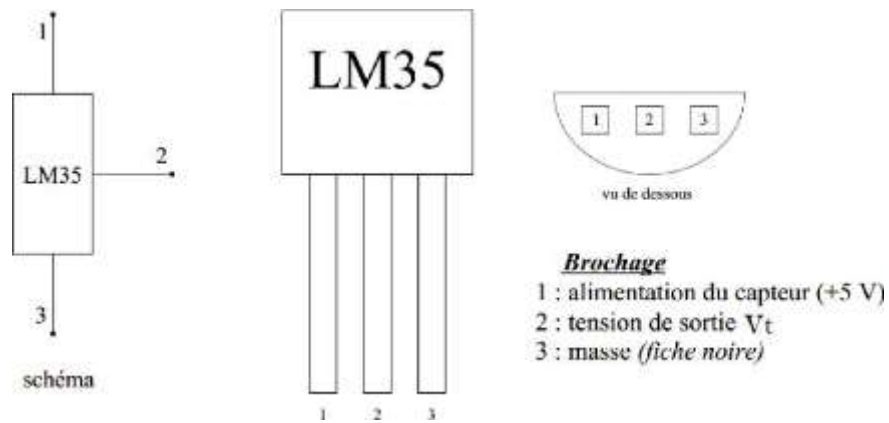


Figure I.10 : brochage du lm35.

## VI.2-Calibration du lm35 :

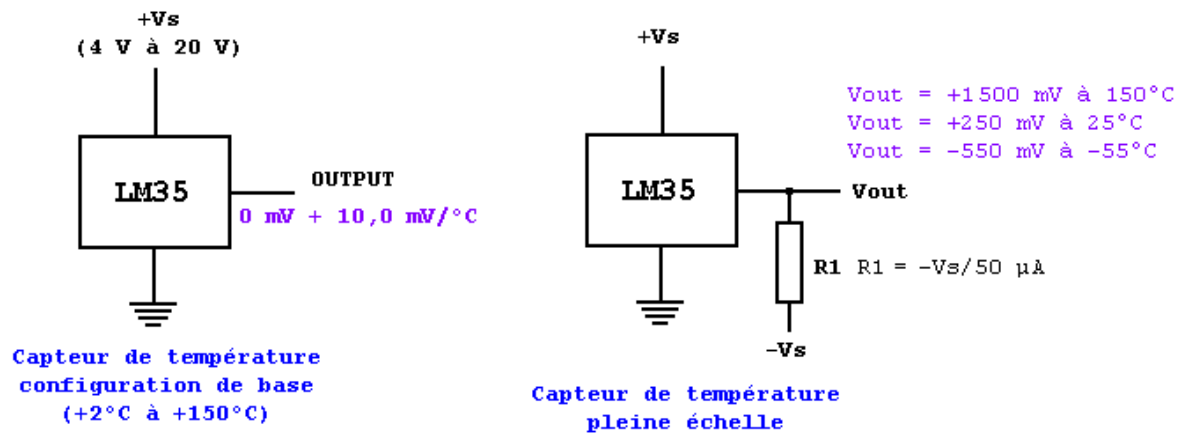


Figure I.11 : calibration du lm335.

## **VII-Conclusion :**

Dans ce chapitre nous avons vu les différentes échelles de mesure de la température, ainsi que les méthodes utilisées pour mesurer cette dernière.

Ensuite nous avons projeté la lumière sur les types de capteurs et leurs compositions, ainsi que leurs champs d'application.

Enfin nous avons défini plus profondément le capteur que nous allons utiliser, qui est le très connu lm35.

## **Chapitre II**

### **Le dispositif programmable**

#### **Arduino**

## **I - Introduction**

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation.

On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

## **II- Définition du module Arduino**

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle), dont les plans de la carte elle-même sont publiés en licence libre, mais certains composants de la carte : comme le microcontrôleur et les composants complémentaires ne sont pas en licence libre. Un microcontrôleur programmé, peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDEArduino. [7]

## II.1 Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques-unes afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programmable à l'aide du port USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- L'Arduino Duemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'Arduino Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega32U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération. [8]

Parmi ces types, nous avons choisi une carte Arduino UNO (carte Basique). L'intérêt de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure ci-dessous. [9]



*Figure II.1 La carte Arduino UNO*

## II.2 Pourquoi Arduino UNO

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation, et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs, tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré-assemblées coûtent moins de 2500 Dinars).
- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple** : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel Open source et extensible** : les cartes Arduino sont basées sur les Microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût. [20]

## II. 3 La constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader, de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

### II.3.1 Partie matérielle

Généralement, tout module électronique possédant une interface de programmation, est basé toujours dans sa construction sur un circuit programmable ou plus.

#### II.3.1.1 Le Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit.

Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure II.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.[9]



Figure II.2 Microcontrôleur ATmega328

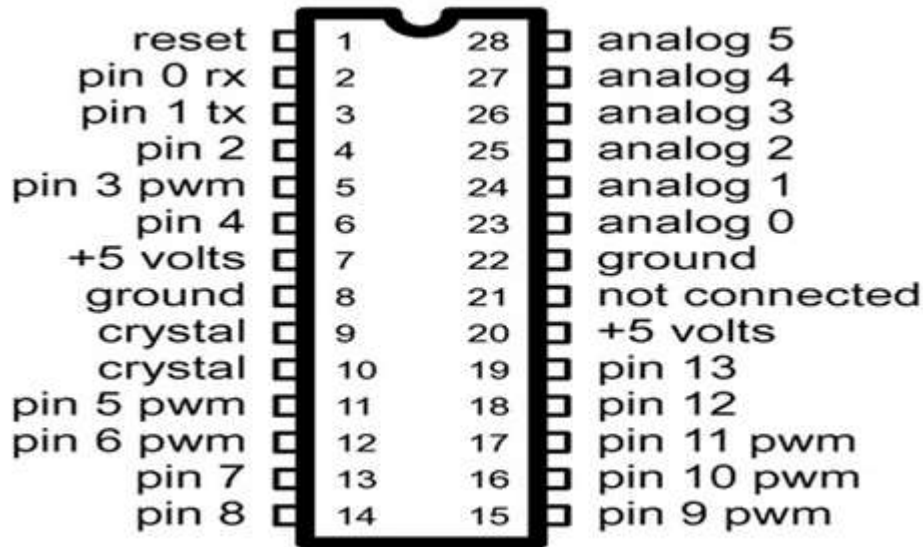


Figure II.3 Brochage de l atmega328

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La mémoire Flash:** C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- **RAM :** c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- **EEPROM :** C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [11]

### II.3.1.2 Les sources d'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :

- **VIN.** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche, ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V.** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- **3V3.** Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V. L'intensité maximale disponible sur cette broche est de 50mA. [8]

### II.3.1.3 Les entrées & sorties

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode()`, `digitalWrite()` et `digitalRead()` du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digital Write (broche, HIGH)`.

En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant,

ou sur un changement de valeur. -Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 1-bits à l'aide de l'instruction analog Write ( ).

- **SPI** (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- **I2C**: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").
- **LED**: Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead( ) du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference( ) du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé. [7]

#### **II.3.1.4 Les ports de communications**

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX).

On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil:

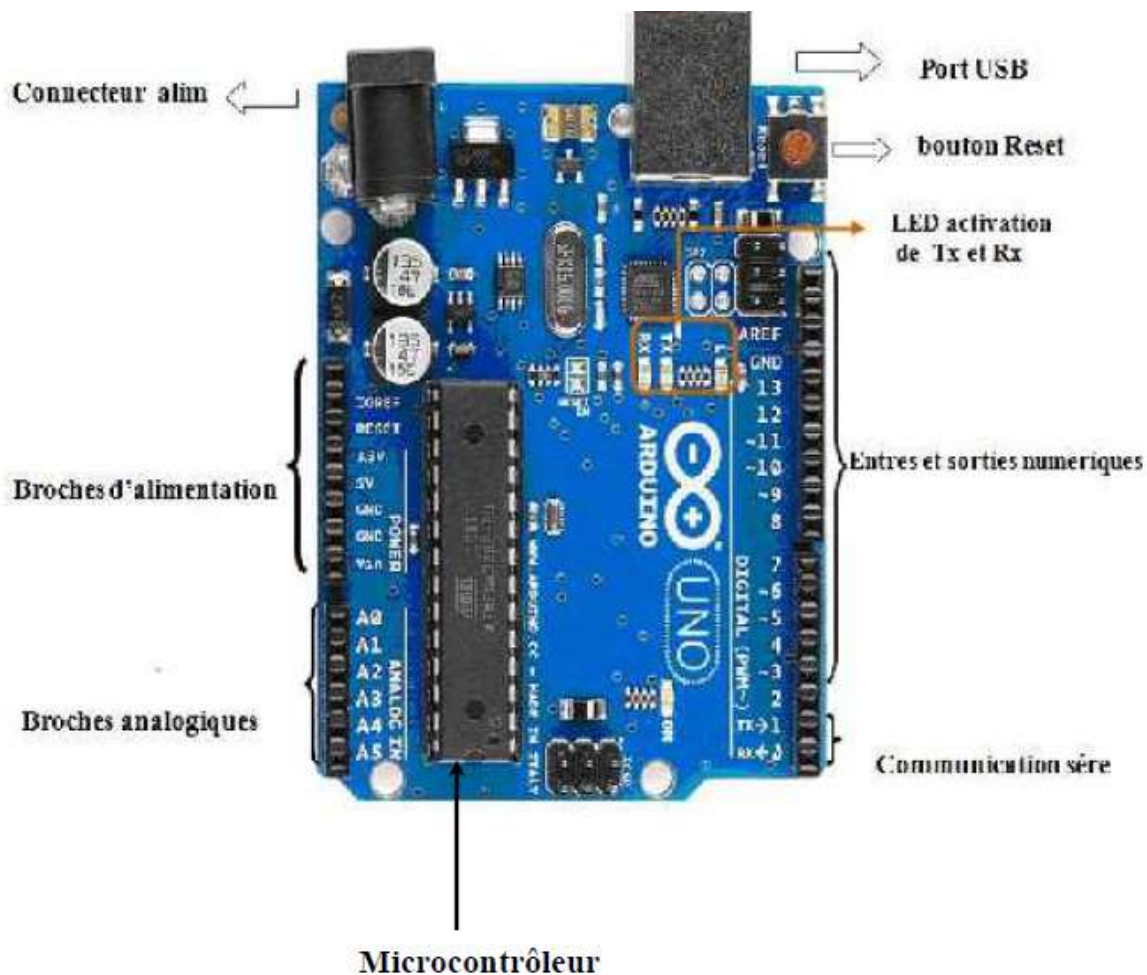


Figure II.4 Constitution de la carte Arduino UNO

### II.3.2 Partie programme

Une telle carte d'acquisition dont sa construction est basée sur un microcontrôleur, doit être dotée d'une interface de programmation comme est le cas de notre carte.

L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

### II.3.2.1 l'environnement de la programmation

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte, et transporte aussi l'information ce programme appelé IDE Arduino. [12]

### II.3.2.2 Structure générale du programme (IDE Arduino)

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

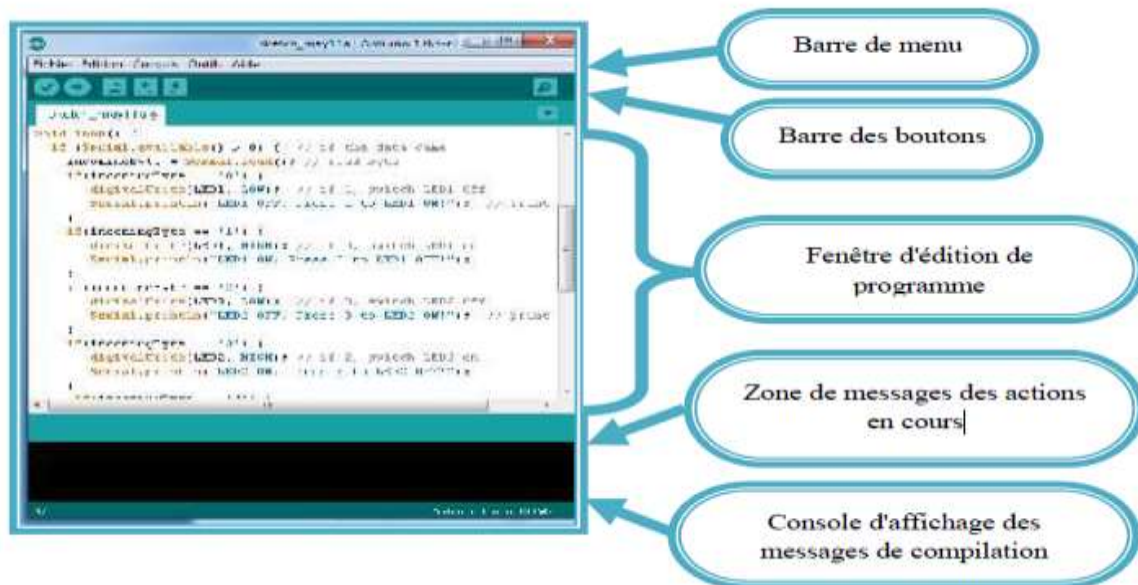


Figure II.5 Interface IDE Arduino

### II.3.2.3 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante.

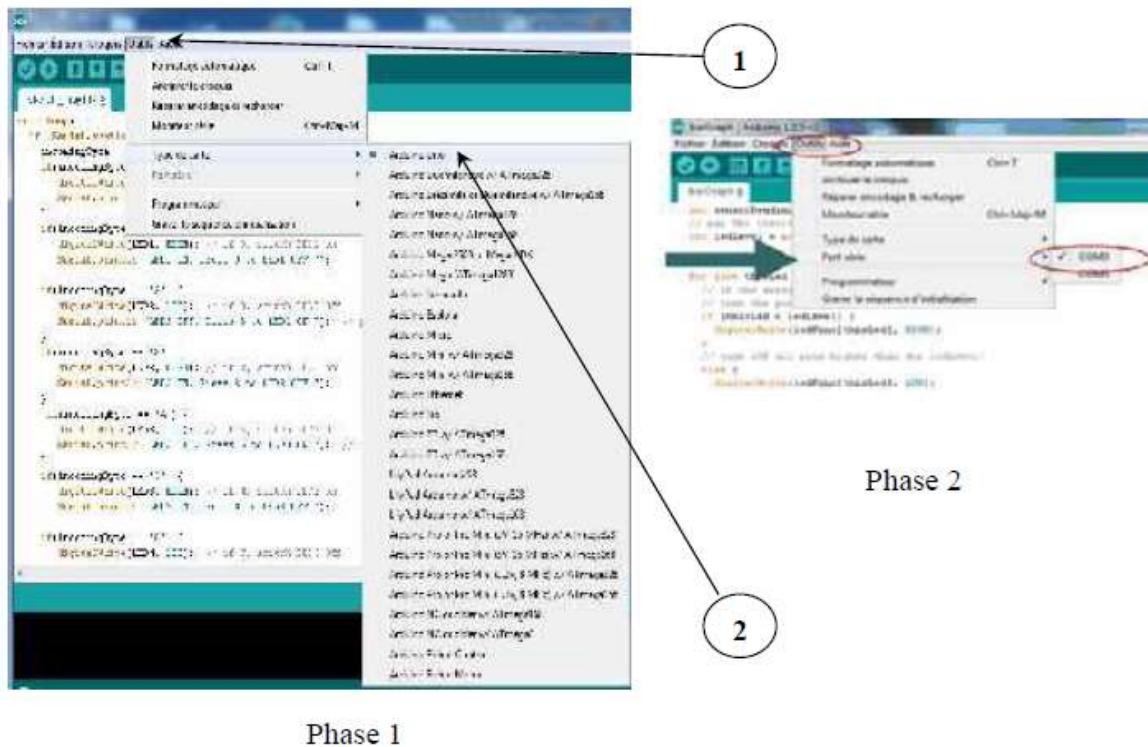


Figure II.6 Paramétrage de la carte

### II.3.2.4 Description du programme

Un programme arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

#### Commentaires

Les **commentaires** sont, en programmation informatique, des portions du code Source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

- 1 /\* programme de commande de température avec lm35-----
- 2 \*et fait également clignoter la diode de test de la carte-----
- 3 \*/-----

#### Définition des variables

Pour notre montage, on va utiliser une entrée analogique de la carte qui est par exemple A0 ; cette variable doit être définie et nommée capteur A0.

#### Configuration des entrées et des sorties void setup ()

les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques; ici on va configurer le capteur A0 en entrée ; pin mode (nom, état) est une des quatre fonctions relatives aux entrées – sorties numériques.

- 4/ void setup() {-----

```
5// mettre le capteur A0 comme entrée:-----  
6/ pinMode(capteur A0, INPUT) -----  
7 }-----
```

### ***Programmation des interactions void loop :***

Dans cette boucle, on définit les opérations à effectuer dans l'ordre **digital write** (nom, état) est une autre des quatre fonctions relatives aux entrées – sorties numériques.

- Delay** (temps en milliseconde) est la commande d'attente entre deux instructions.
- Chaque ligne d'instruction est terminée par un point-virgule.
- Ne pas oublier les accolades qu'encadrent la boucle.

### **Insertion des bibliothèques :**

Dans notre cas on doit insérer la bibliothèque de l'afficheur LCD.

Configuration des entrées et des sorties void setup () :

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques : dans notre cas en utilise

→ La pin A0 en entrée.

→ La pin 10 et 13 en sortie.

## Programmation des interactions void loop :

Dans cette boucle, on définit les opérations à effectuer dans l'ordre, en réalité c'est le programme à exécuter par le microcontrôleur.

### Programme général :

```
//insertion de la bibliothèque de l'afficheur

#include<LiquidCrystal.h>

LiquidCrystal lcd (12,11,5,4, 3,2);

#define capteur A0

void setup()

//choisir les entrées sorties

{ pinMode(13, OUTPUT); pinMode(10, OUTPUT);pinMode(A0, INPUT);

  int i;

//définir ou afficher le message qui suit cette instruction

for (i=0;i<16;i++){

  lcd.setCursor(i,0);

  lcd.print(" NASSIM ");

  delay(150); }

  lcd.noDisplay();

  delay(30);

  lcd.display();

  delay(100);

//définir ou afficher le message 2 qui suit cette instruction

for (i=0;i<16;i++){

  lcd.setCursor(i,0);

  lcd.print(" VOUS ");

  delay(150); }

  lcd.noDisplay();
```

```
delay(30);

lcd.display();

delay(100);

for (i=0;i<16;i++){

lcd.setCursor(i,0);

//définir ou afficher le message qui suit cette instruction

lcd.print(" PRESENTE");

delay(150); }

lcd.noDisplay();

delay(30);

lcd.display();

delay(100);

for (i=0;i<16;i++){

lcd.setCursor(i,0);

//définir ou afficher le message qui suit cette instruction

lcd.print(" ALARME DE ");

delay(150); }

lcd.noDisplay();

delay(30);

lcd.display();

delay(100);

for (i=0;i<16;i++){

//définir ou afficher le message 3 qui suit cette instruction

lcd.setCursor(i,0);

lcd.print(" TEMPERATURE ");

delay(150); }
```

```
lcd.noDisplay();

delay(30);

lcd.display();

delay(100);

lcd.begin(16,2);

lcd.setCursor(0,0);

lcd.print(" Thermometer");

}

//programme principal

void loop()

{

int valcapt = analogRead(A0);

float voltage = (valcapt /1024.0)*5000;

float degresCelsius = (voltage-0.5) /10;

lcd.setCursor(0,1);

lcd.print(degresCelsius);

lcd.write(0xdf);

lcd.print("C ");

lcd.print(((degresCelsius * 1.8) + 32));

lcd.write(0xdf);

lcd.print("F");

// condition pour que l'alarme sonne à 29°

if(degresCelsius>=29){

tone(13,600);

digitalWrite(10, HIGH);

delay(200);
```

```
noTone(13);  
  
digitalWrite(10, LOW);  
  
delay(200);  
  
} else noTone(13);  
  
digitalWrite(10, LOW);  
  
delay(200);  
  
}
```

### II.3.2.5 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.



Figure II.7 Les étapes de téléchargement du code

## II.4 Les Accessoires de la carte Arduino

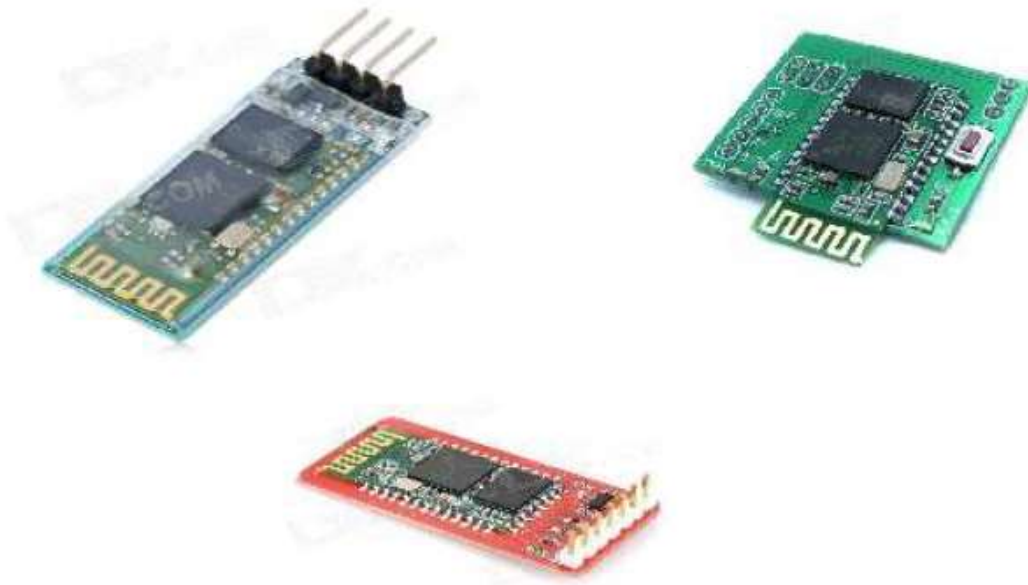
La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations.

### II.4.1 Communication

Le constructeur a suggéré qu'une telle carte doit être dotée de plusieurs ports de communications ; on peut éclaircir actuellement quelques types.

#### II.4.1.1 Le module Arduino Bluetooth

Le Module Microcontrôleur Arduino Bluetooth est la plateforme populaire Arduino avec une connexion série Bluetooth à la place d'une connexion USB, très faible consommation d'énergie, très faible portée (sur un rayon de l'ordre d'une dizaine de mètres), faible débit, très bon marché et peu encombrant.



*Figure II.8 Type de modules Bluetooth*

### II.4.1.2 Le module shield Arduino Wifi

Le module Shield Arduino Wifi permet de connecter une carte Arduino à un réseau internet sans fil Wifi.



*Figure II.9 Module shield wifi*

### II.4.1.3 Le Module XBee

Ce module permet de faire de la transmission sans fil, faible distance /consommation /débit/ prix. [13]



*Figure II.10 Module XBee*

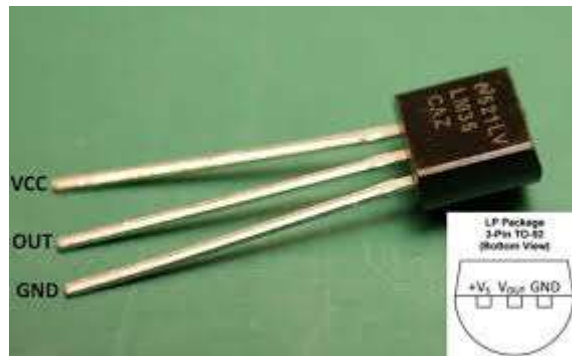
## II.4.2 Les capteurs

Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé.



*Figure II.11 Capteur Arduino*

Dans notre cas en utilise un LM35 qui est un capteur de température qui produit une tension qui varie en fonction de la température [13].



*Figure II.12 capteur de température lm35*

## II.4.3 Les drivers

Il existe plusieurs drivers comme des cartes auxiliaires qui peuvent être attachées avec l'Arduino afin de faciliter la commande ; on peut citer quelques types.

## 1- Des moteurs électriques



Figure II.13 Moteurs électriques

## 2- Les afficheurs LCD

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement.

Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux.

Les caractères sont prédéfinis. [13]

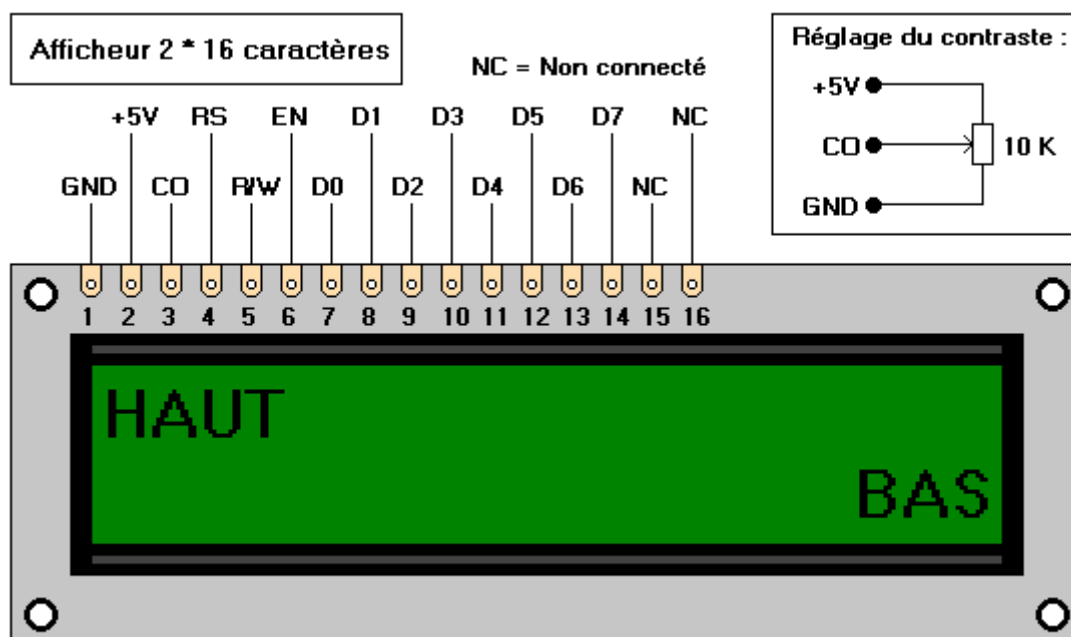


Figure II.14 brochage d'un Afficheurs LCD

### 3- Le relais

C'est un composant qui possède une bobine (électro-aimant) qui est parcourue par un courant électrique agissant sur un ou plusieurs contacts. Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en œuvre un mouvement mécanique. [14]



Figure II.15 Relais

### 4-Le buzzer

Un bipeur (en anglais *buzzer*) est un élément électromécanique ou piézoélectrique qui produit un son caractéristique quand on lui applique une tension : le bip. Certains nécessitent une tension continue, d'autres nécessitent une tension alternative.



Figure II .16 buzzer tone

## **II.5 Conclusion**

Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino uno, donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité les différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

## **Chapitre III**

### **Réalisation du dispositif**

### **Expérimental**

## **I-Introduction**

Dans ce chapitre, on présentera de manière sommaire une vue d'ensemble du dispositif << Alarme de température à base d'arduino uno >>

Ce travail d'initiation à base d'une carte Arduino UNO permet d'alerter en cas de dépassement du seuil de température programmé d'un système.

Après avoir donné dans le chapitre précédent une description théorique sur le module Arduino et son environnement de développement, on va procéder à l'application expérimentale ; pour cette raison, plusieurs blocs ont été nécessaires afin de réaliser une telle combinaison.

## **II- Les différentes étapes de la réalisation**

Notre réalisation pratique a été faite en trois parties:

- ➔ La première partie est la conception de tout le système électronique.
- ➔ La deuxième partie est la réalisation pratique du dispositif sur lab d'essai.
- ➔ La troisième partie est la mise en marche du dispositif.

La première partie de notre projet est très importante, on est passé par plusieurs étapes:

1. Chercher les différentes structures des blocs constituant notre maquette et qui vont avec les objectifs fixés et les moyens disponibles.
2. Présenter les différents éléments ou composants constituant chacun des blocs en choisissant des composants aux caractéristiques voulus, à défaut, on choisira ceux disponibles sur le marché.

Dans la deuxième partie « réalisation pratique », on passe par les deux étapes suivantes :

1. Présenter les différentes étapes de la réalisation pratique.
2. On assemble ensuite les composants suivant notre montage sur lab d'essai, commençant par l'alimentation générale de notre dispositif.

La troisième partie ; c'est la mise en marche.

### III- Schéma synoptique général

Le schéma synoptique général de notre dispositif est indiqué par la figure (III.1).

En ce qui concerne l'élément principal de ce dispositif, notre choix était le dispositif Arduino UNO.

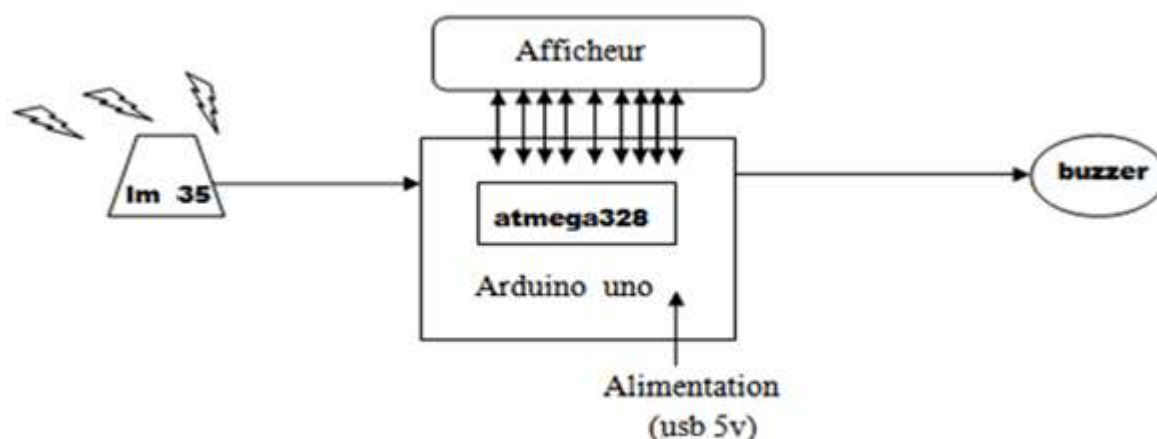


Figure III.1 Schéma synoptique du dispositif

La réalisation de notre système est représentée à la fin de ce chapitre.

### IV- La réalisation virtuelle « PROTEUS » :

Avant de passer à la réalisation pratique, nous avons utilisé un CAO: il s'agit de ISISPORTEUS, c'est un CAO électronique perfectionné conçu par Labcenter Electroniques qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le CAO électronique « PROTEUS » est disponible et téléchargeable sur ce lien [10], et se compose de nombreux outils regroupés en modules au sein d'une interface unique. Ce dernier nous permet de schématiser notre carte électrique et la simuler virtuellement comme le montre la figure suivante :

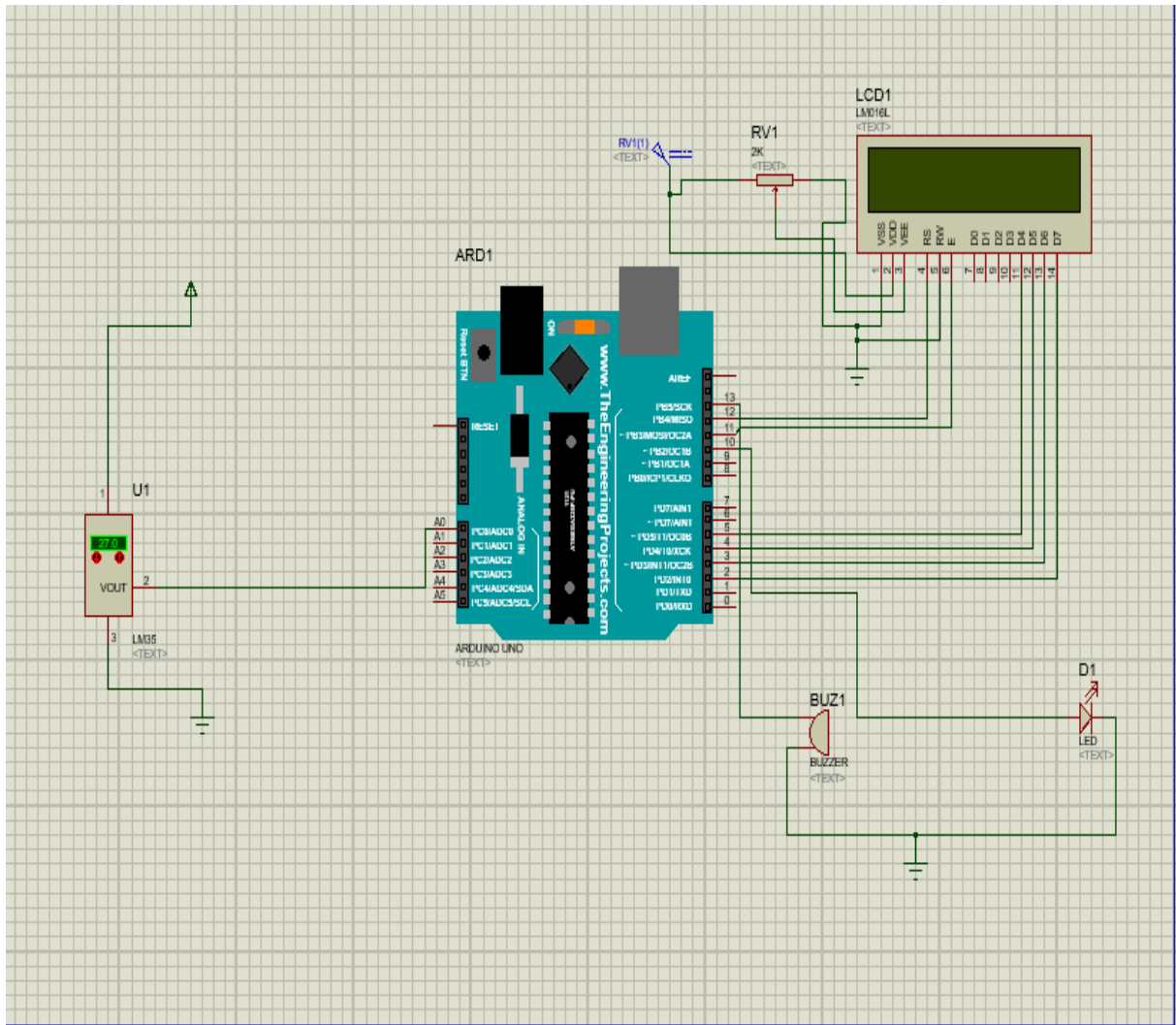
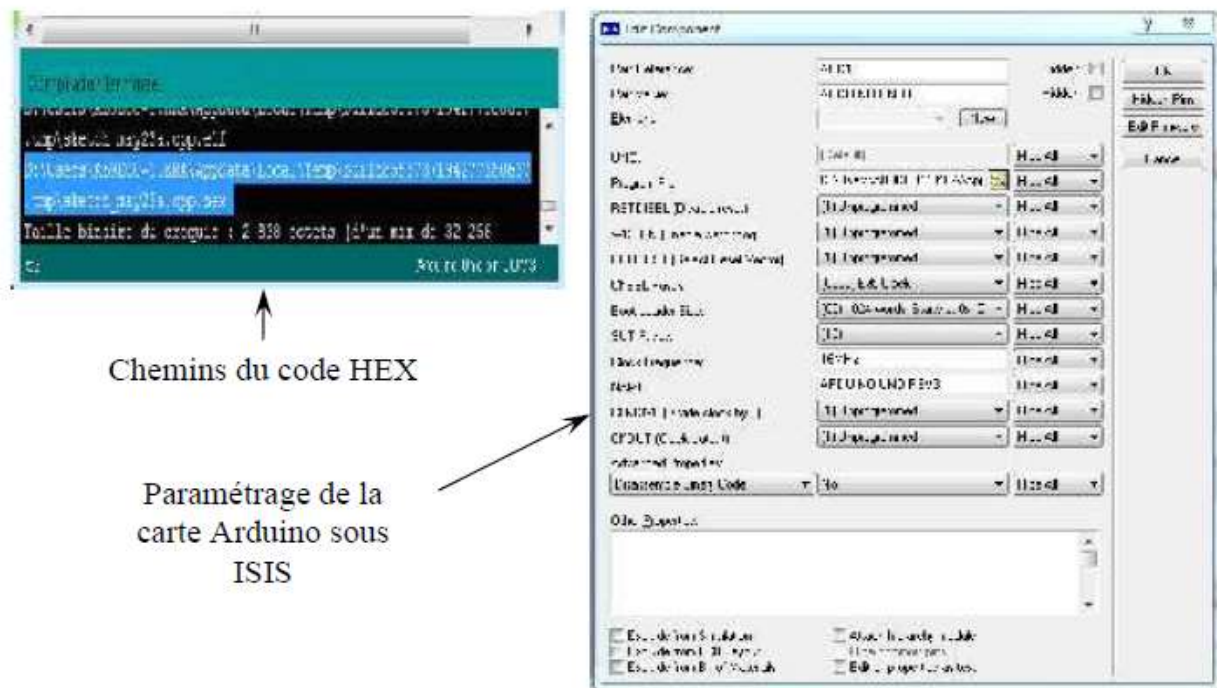


Figure III.2 La carte réalisée sous ISIS-PROTEUS

Ce CAO a la possibilité d'emporter même des codes hexadécimaux pour les réalisations qui contiennent des composants programmables, ou des cartes programmables « Arduino » comme dans notre réalisation.



Chemins du code HEX

Paramétrage de la  
carte Arduino sous  
ISIS

Figure III.3 Le chemin du fichier de code HEX de notre programme

### V-Explication et démarche :

Notre montage consiste à capter la température ambiante et cela grâce à notre capteur de température lm35, qui transforme la chaleur capté en tension, et grâce à sa linéarité, avec l'augmentation de la chaleur la tension augmente.

La tension récoltée par le capteur est transmise à l'aide de fils vers notre carte Arduino uno ,qui se charge d'analyser et de produire des signaux électriques avec le microcontrôleur atmel atmega 328, qui est un circuit intégré programmé au préalable, ainsi avec les composants complémentaires comme le régulateur de tension, ou l'oscillateur de quartz.

Le résultat est envoyé vers un afficheur LCD 16\*2, qui est relié à l'arduino avec une communication semi-parallèle et dont le contraste est réglable avec un potentiomètre.

Avec l'augmentation de la température, le capteur va transmettre les données récoltées à l'arduino, puis seront affichées sur l'afficheur. En cas de dépassement du seuil de température programmé au par avant qui est de 37° c, le buzzer se mettra à sonner et la led à clignoter, et cela jusqu'à ce que la température redescende en dessous du seuil.

## **V-Composants utilisés :**

-arduino uno.

-capteur de température LM-35.

-buzzer .

-afficheur LCD (16x2).

-une led.

-résistances.

-potentiomètre.

La photo globale de notre réalisation est mentionnée dans annexes

## ***Conclusion :***

Cette partie a donné lieu à la réalisation pratique d'un system électronique pour simplement alerter les fonctionnaires d'une usine, d'une maison..., en permettant d'afficher la température atteinte sur un afficheur, et en cas de dépassement du seuil de température programmé , une alarme va se déclenchée via le buzzer.

Sur le plan pratique, une manipulation adéquate du logiciel « Arduino » nous permet alors d'utiliser un compilateur ; il s'agit de « IDE » ce dernier possède une capacité de créer un code HEX, qui peut être injecte sur un microcontrôleur.

L'ensemble des travaux décrits dans ce chapitre sont :

- La conception des composants électroniques qui compose la carte réalisée à l'aide logiciel (ISIS- PROTUSE)
- Le choix des composants
- Les différentes étapes de réalisations.
- Le principe de fonctionnement

On peut conclure que les résultats obtenus sont satisfaisants compte tenu des limitations du matériel et des moyens dont nous disposons.

# **Conclusion générale**

Le système Arduino, est un outil pour fabriquer des dispositifs qui peuvent capter et contrôler d'avantage de choses du monde matériel que votre ordinateur. C'est une plateforme open-source d'électronique programmée, qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

La programmation du microcontrôleur AVR exige une connaissance en langage « C », comme soft et la maîtrise des outils « IDE», afin d'arriver à l'objectif souhaité.

Dans ce travail, un assemblage de plusieurs logiciels est fait afin de réaliser un tel résultat

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que je me suis confronté à plusieurs problèmes dans la partie réalisation et programme de la carte.

Cependant, on peut dire que malgré ces difficultés, les résultats obtenus à travers cette étude qu'ils soient pratiques ou théoriques, permettent d'ouvrir la porte à d'autres études.

Je souhaite vivement que ce projet puisse servir comme élément de base pour d'autres études plus approfondies pour le faire intégrer sous des systèmes plus complexes.

# **Annexes**

## Annexes

### Les Logiciels Utilisés

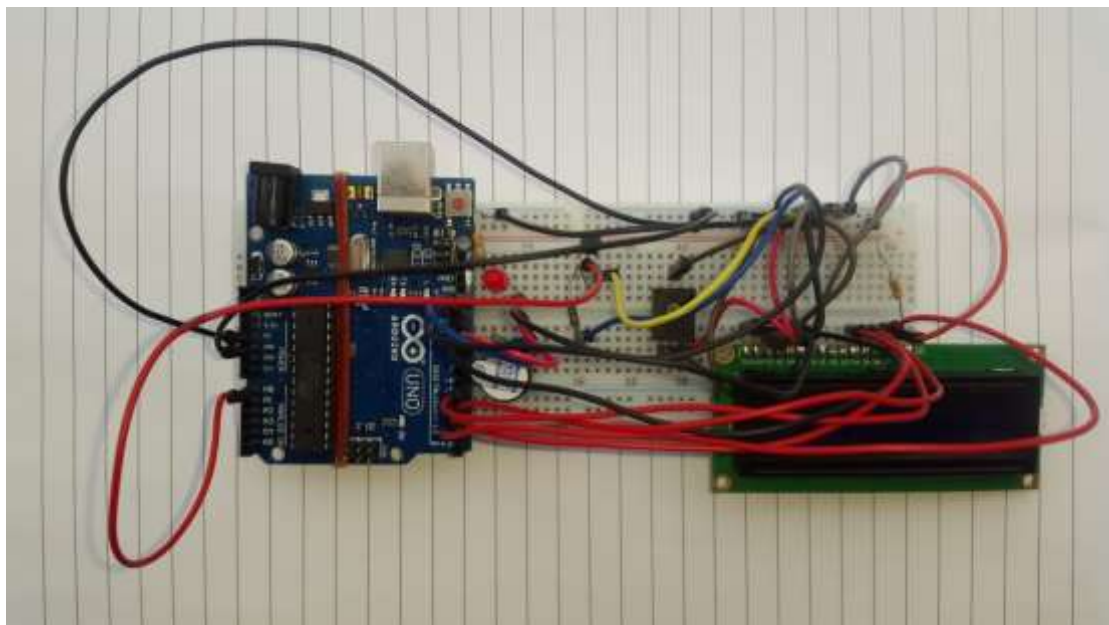
Pour programmer Arduino



Pour la simulation virtuelle de notre carte



Image de la réalisation finale



# **Bibliographie**

- [1] [http:// www.aviatechno.net/](http://www.aviatechno.net/)
- [2] <http://www.get.univ-littoral.fr/>
- [3] <http://www.cornetdumarker.net/>
- [4] <http://www.abcelectronique.com/>
- [5] <http://www.ibni.over-blog.com/>
- [6] <http://www.digikey.fr/>
- [7] <http://www.generationrobots.com/fr/152-arduino>
- [8] S.V.D.Reyvanth, G.Shirish, « PID controller using Arduino ».
- [9] C. Tavernier, « Arduino applications avancées ». Version Dunod.
- [10] <http://www.labcenter.com/>
- [11] [http://www.acm.uiuc.edu/sigbot/tutorials/ 2009-11-17-arduino-basics](http://www.acm.uiuc.edu/sigbot/tutorials/2009-11-17-arduino-basics)
- [12] Jean- Noël, « livret Arduino en français », centre de ressources art sensitif.

