

الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي والبحث العلمي

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

جامعة أبي بكر بلقايد - تلمسان -

**Université Aboubakr Belkaïd – Tlemcen –  
Faculté de TECHNOLOGIE**



## **MEMOIRE**

Présenté pour l'obtention du **diplôme** de **MASTER**

**En** : Electronique

**Spécialité** : Instrumentation

**Par** : ZAR HAMZA & MELLOUK ISLAM

**Sujet**

**Conception et réalisation d'un contrôleur de  
température automatique par carte Arduino**

Soutenu publiquement, le : 27/06/ 2022, devant le jury composé de :

M <sup>r</sup> BOUANATI Sidi Mohammed	MAA	Univ. Tlemcen	Président
M <sup>me</sup> HEDARA Rachida	MCB	Univ. Tlemcen	Encadreur
M <sup>me</sup> GHEFIR Fatiha Djazia	MAA	Univ. Tlemcen	Examineur

Année Universitaire : 2021-2022

## **Remerciement**

*Tout d'abord je tiens à remercier **ALLAH** le tout puissant de m'avoir donné la santé, la volonté, le courage et la patience pour mener à terme ma formation et pourvoir réaliser ce travail de recherche.*

*Je tiens à remercier, très sincèrement, le DR : **BRIXI.A**, responsable de la formation Master ELN, j'ai eu l'honneur et la chance de bénéficier de ses connaissances et compétences, de ses précieux conseils et de son suivi tout au long de notre parcours académique. Son sens élevé du devoir, je vous remercie infiniment.*

*Mes remerciements s'adressent particulièrement au DR : **HEDARA.R**, pour son encadrement de qualité, sa motivation professionnelle, ses conseils et critiques constructives, ses corrections, sa gentillesse et sa patience ainsi pour le temps qu'il a consacré à la réalisation de ce travail.*

*Mes remerciements vont également aux les membres du jury qui nous ont fait l'honneur d'accepter de participer au jury de soutenance et d'évaluer la présente mémoire : le président **Mr: BOUANATI.M**, Enseignant à l'université de Tlemcen, et l'examinatrice **Mme: GHEFIR.F**, Enseignante à l'université Tlemcen.*

*Ainsi mes enseignants, espérant que vous allez voir, dans ce manuscrit, les fruits du dévouement avec lequel vous avez fait preuve durant les enseignements que vous nous avez prodigué.*

*De peur d'en avoir oublié, je souhaite remercier tous ceux qui ont contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de ce parcours universitaire.*

## **Dédicace**

*Je demande à Dieu Tout-Puissant d'être au niveau requis après les soins, la recherche, l'arrangement, la coordination et le traitement des sujets jusqu'à ce que j'écrive cette recherche, que je demande à Dieu de recevoir votre approbation et d'ajouter une valeur scientifique aux bibliothèques, si Dieu le veut.*

*Je dédie cet humble travail à ceux qui sont restés debout et ont passé leur vie pour que j'atteigne ce degré, mon cher père, comme je n'oublie pas celui qui m'a élevé et nourri, ma chère mère.*

*Je dédie aussi ce travail à ma famille, mes amis, mes collègues de  
M2 instrumentation 2022*

## ***Dédicace***

*D'abord je remercie le mon Dieu qui m'a donné le courage pour arriver à la fin de mes études.*

*Je dédie ce modeste travail à :*

*Mon père et ma mère que dieu les garde et les protège pour leurs soutien moral et financier, pour leurs encouragements et les sacrifices qu'ils ont endurés.*

*Mes chers frères.*

*Mes chères sœurs.*

*Toute ma famille et mes amis*

*En fin à moi-même et mes collègues de la promotion 2022 Instrumentation*

## **Résumé**

Notre projet consiste à étudier et à réaliser un dispositif numérique d'un contrôleur de température à base d'une carte Arduino programmée de plage [-55°C à 155°C], en utilisant le programme IDE. Cet appareil consiste à contrôler et mesurer la température dans la situation locale. Nous avons simulé le fonctionnement d'un circuit de contrôle de température numérique en utilisant un capteur de température LM35, afficheur LCD et des composants d'alertes ce qui permet la surveillance du système. Notre circuit fonctionne de telle façon quand le capteur capte une variation de la température d'un lieu quelconque, le système commence à afficher la valeur de la température sur LCD. Une fois la valeur de température actuelle augmente au de la valeur de consigne qu'elle est fixée, les éléments de sécurités activent : le relai automatiquement vas couper le passage de courant pour éteindre donc l'élément chauffant tel que le chauffage qui est connecté à la sortie de relais et la LED rouge va allumer.

**Mots-clés :** Arduino UNO, Capteur de température, Afficheur LCD, CAN, Relais, I2C, PROTEUS.

## **Abstract**

Our project consists in studying and realizing a digital device of a temperature controller based on an Arduino board programmed with range [-55°C to 155°C], using the IDE program. This device is to control and measure the temperature in the local situation. We simulated the operation of a digital temperature control circuit using an LM35 temperature sensor, LCD display and alert components which allows monitoring of the system. Our circuit works in such a way when the sensor detects a variation of the temperature of any place, the system begins to display the value of the temperature on LCD. Once the current temperature value increases above the setpoint value that it is set, the safety elements activate: the relay will automatically cut off the current flow to therefore turn off the heating element such as the heating which is connected to the relay output and the red LED will light up.

**Key-words:** Arduino UNO, Temperature sensor, LCD display, CAN, Relay, I2C, PROTEUS.

## المخلص

يتكون مشروعنا من دراسة وتحقيق جهاز رقمي لجهاز التحكم في درجة الحرارة على أساس لوحة Arduino مبرمجة بنطاق من [-55 درجة مئوية إلى 155 درجة مئوية] ، باستخدام برنامج IDE. يقوم هذا الجهاز بالتحكم في درجة الحرارة وقياسها في الوضع المحلي. قمنا بمحاكاة تشغيل دائرة التحكم في درجة الحرارة الرقمية باستخدام مستشعر درجة الحرارة LM35 وشاشة LCD ومكونات التنبيه التي تسمح بمراقبة النظام. تعمل دائرتنا بهذه الطريقة عندما يكتشف المستشعر تباينًا في درجة حرارة أي مكان ، يبدأ النظام في عرض قيمة درجة الحرارة على شاشة LCD. بمجرد زيادة قيمة درجة الحرارة الحالية فوق قيمة نقطة الضبط التي تم ضبطها ، يتم تنشيط عناصر الأمان: سيقوم المرحل تلقائيًا بقطع تدفق التيار لإيقاف عنصر التسخين مثل التسخين المتصل بإخراج المرحل والأحمر سوف تضيء LED.

الكلمات الرئيسية: Arduino UNO ، مستشعر درجة الحرارة ، شاشة LCD ، CAN

، Relay ، I2C ، PROTEUS .

# Table des matières

<b>Introduction générale</b> .....	1
<b>CHAPITRE- I</b>	<b>Généralités sur les capteurs de température</b>
I.1. Introduction.....	3
I.2. Unités et conversion de température.....	3
I.3. Définition d'un capteur .....	4
I.4. Structure d'un capteur.....	4
I.5. Rôle du capteur .....	5
I.6. Capteur de température.....	5
I.6.1. Capteurs à contact.....	5
I.6.2. Capteurs de proximité.....	5
I.6.3. Les capteurs actifs .....	6
I.6.4. Les capteurs passifs .....	7
I.6.5. Les capteurs de températures les plus connus .....	7
I.6.5.1. Thermocouple .....	7
I.6.5.2. Capteur de température à résistance (RTD) .....	9
I.6.5.3. Thermistance .....	9
I.6.5.4. Les capteurs de température à semi-conducteur .....	11
I.6.5.5. Les pyromètres optiques .....	11
I.7. Classification des capteurs .....	12
I.7.1. Signal logique .....	12
I.7.2. Signal analogique .....	13
I.7.3. Signal numérique .....	14
I.8. Chaîne de mesure .....	15
I.9. Caractéristiques d'un capteur.....	16
I.10. Capteur de température LM35.....	17
I.10.1. Définition .....	17
I.10.2. Les caractéristiques de LM35 .....	17
I.10.3. Différentes versions de LM35 .....	18
I.10.4. Fonctionnement du capteur de température LM35 .....	20
I.10.5. Étalonnage du capteur .....	22

I.11. Conclusion .....	22
------------------------	----

## **CHAPITRE-II**

## **Description générale d'un Arduino UNO**

II.1. Introduction.....	24
II.2. Définition du module Arduino .....	24
II.3. L'intérêt et l'utilité de la carte Arduino .....	26
II.4. Applications du module Arduino.....	26
II.5. Les types de la carte Arduino .....	27
II.5.1. Arduino UNO.....	27
II.5.2. Arduino NANO.....	27
II.5.3. Arduino Due.....	28
II.5.4. Arduino Mega 2560.....	29
II.5.5. Arduino Leonardo .....	29
II.6. Choix de l'Arduino UNO .....	30
II.7. La propriété de la carte Arduino UNO.....	31
II.7. 1. Partie matérielle .....	31
II.7.2. Partie programme.....	35
II.7.2.1. Logiciel Arduino.....	36
II.7.2.2. Présentation de «IDE ».....	36
II.7.2.3. Programmer avec Arduino .....	37
II.7.2.4 Programme d'injection .....	37
II.7.2.5. Structure des commandes Arduino .....	37
II.7.2.6. Les étapes de téléversement du programme .....	39
II.8. Périphériques de la carte Arduino .....	39
II.8.1. Les capteurs .....	39
II.8.2. Les ports de communications.....	40
II.8.3. L'afficheur LCD.....	42
II.8.4. Interface I2C Arduino pour la connexion du LCD .....	44
II.8.5. Diode électroluminescente (LED).....	45
II.8.6. Transistor BC548.....	47
II.8.7. Le relais .....	48
II.9. Schéma synoptique d'un contrôleur de la température.....	49
II.10. Conclusion.....	49



## **CHAPITRE-III      Réalisation de contrôleur de température numérique par carte Arduino**

III.1. Introduction.....	51
III.2. Conception générale de l'application .....	51
III.3. Schéma synoptique des différents blocs .....	51
III.4. Réalisation du contrôleur de température .....	52
III.4.1 Partie électronique .....	52
III.4.2 Partie informatique .....	52
III.4.3 Partie Logiciel-Simulation .....	53
III.4.3.1 Présentation de PROTEUS .....	53
III.4.3.2 Logiciel ISIS .....	54
III.4.3.3 Conception logicielle en environnement Arduino .....	55
III.4.4. Fonctionnalités du programme .....	58
III.4.4.1 Organigramme de programme initial.....	60
III.4.4.2 Organigramme de programme principal.....	61
III.4.5. Aspects de la simulation .....	62
III.4.5.1 Bibliothèque Arduino pour Proteus .....	62
III.5 Résultats de la simulation .....	63
III.5.1. L'état initial du système .....	63
III.6 Logiciel ARES de conception de circuit imprimé .....	66
III.7 Liste des composants utilisés .....	68
III.8 Réalisation le thermomètre numérique sur plaque d'essais .....	68
III.9 Quelques mesures de températures effectuées avec le dispositif numérique réalisé .....	72
III.10 Conclusion .....	73
Conclusion générale .....	75
Références .....	76
Anexes.....	79

## Liste des figures

- Figure I.1** Structure d'un capteur
- Figure I.2** Schéma d'un capteur actif
- Figure I.3** Types de thermocouples
- Figure I.4** Capteurs à résistance RTD
- Figure I.5** Courbes caractérisant la CTP et CTN
- Figure I.6** Représentation schématique
- Figure I.7** Capteurs de thermistance
- Figure I.8** A courant constant, la tension  $V$  est linéaire en fonction de la température
- Figure I.9** Structure de capteur logique
- Figure I.10** Signal logique en fonction du temps
- Figure I.11** Structure d'un capteur logique
- Figure I.12** Signal analogique
- Figure I.13** Signal numérique
- Figure I.14** Schéma d'un capteur numérique
- Figure I.15** Chaîne de mesure informatisée
- Figure I.16** L'étendue de mesure : zone en dehors de laquelle le capteur ne transmet plus un signal en relation avec la grandeur physique capté
- Figure I.17** Capteur de température LM35
- Figure I.18** Tension de sortie LM35 en mV en fonction de la température
- Figure I.19** Calibration du LM335
- Figure II.1** La carte Arduino UNO
- Figure II.2** Le microcontrôleur de l'Arduino
- Figure II.3** Architecture d'un système à base d'une carte Arduino
- Figure II.4** Carte Arduino UNO
- Figure II.5** Carte Arduino Nano
- Figure II.6** Carte Arduino Due
- Figure II.7** Carte Arduino Mega 2560
- Figure II.8** Carte Arduino Leonardo
- Figure II.9** La carte Arduino UNO
- Figure II.10** Interface IDE Arduino
- Figure II.11** Structure des commandes Arduino

- Figure II.12** Étapes de téléchargement du code
- Figure II.13** Le module Arduino Bluetooth
- Figure II.14** Le Module XBee
- Figure II.15** Le module shields Arduino Wifi
- Figure II.16** L'afficheur LCD16\*2
- Figure II.17** Interface I2C Arduino pour la connexion de l'appareil
- Figure II.18** Interface I2C Arduino avec LCD16\*2
- Figure II.19** Diodes électroluminescentes (Rouge, Blue, Vert)
- Figure II.20** Transistor BC548
- Figure II.21** Module Relais
- Figure II.22** Schéma synoptique d'un contrôleur de la température
- Figure III.1** Schéma synoptique de dispositif de contrôleur de température numérique
- Figure III.2** Les deux phases de la réalisation
- Figure III.3** Logiciel ISIS
- Figure III.4** Fenêtre du démarrage de l'ISIS
- Figure III.5** Partie déclaration des variables dans le programme.
- Figure III.6** Partie déclaration dans le programme
- Figure III.7** Fonction « setup ( ) » du programme
- Figure III.8** Fonction « loop ( ) » du programme
- Figure III.9** Organigramme du programme initial
- Figure III.10** Organigramme du programme principal
- Figure III. 11** Intégrer carte Arduino sous Proteus.
- Figure III.12** Charge carte Arduino sous Proteus
- Figure III. 13** L'état initial du système
- Figure III. 14** Montage électrique de contrôleur de température numérique :  
Températures actuelle inférieure à le point de consigne
- Figure III. 15** Montage électrique de contrôleur de température numérique :  
Températures actuelle supérieure à le point de consigne
- Figure III.16** Montage de contrôleur de température numérique à base de la carte Arduino sur l'ARES
- Figure III.17** Dessin de montage de contrôleur de température numérique à base de la carte Arduino

- Figure III.18** Photos de contrôleur de température numérique réalisé en état marche  
(Température actuelle < Point de consigne de température (30 °C))
- Figure III.19** Contrôleur de température numérique réalisé en état marche  
(Température actuelle > Point de consigne de température (30 °C))
- Figure III.20** Contrôleur de température numérique réalisé en état marche  
(Température actuelle < Point de consigne de température (45 °C))
- Figure III.21** Photo de contrôleur de température numérique réalisé en état marche  
(Température actuelle > Point de consigne de température (45 °C))

## Liste des tableaux

<b>Tableau I.1</b>	Formule de calcul pour la conversion entre unités de température
<b>Tableau I.2</b>	Lettres de référence des thermocouples les plus couramment utilisés.
<b>Tableau I.3</b>	Plage de température et sensibilité moyenne de certains capteurs
<b>Tableau I.4</b>	Propriétés de LM35
<b>Tableau I.5</b>	Différentes versions du LM335
<b>Tableau II.1</b>	Types cartes Arduino et leur caractéristique
<b>Tableau II.2</b>	Rôle des <b>broches</b> d'afficheur LCD 16*2
<b>Tableau II.3</b>	Branchement de types Arduino avec interface I2C
<b>Tableau II.4</b>	Caractéristique de transistor
<b>Tableau III.1</b>	Résultats effectués de contrôleur de température réalisé

# **Introduction générale**

### Introduction générale

Au fil des années, le développement de l'industrie électronique a connu une course toujours plus rapide, notamment vers la miniaturisation, mais elle n'a pas quitté ses origines. En fait, le besoin d'appareils plus petits a incité les conducteurs à en faire plus pour répondre aux demandes du marché. C'est dans ce contexte que se situe ici le travail présenté dans cet article, dédié à la réalisation de régulateurs numériques de température.

Cet appareil de contrôle de température numérique est un appareil de mesure, d'affichage et de surveillance des valeurs de température. Il existe plusieurs phénomènes qui font face aux changements de température. Alors de nos jours, on distingue plusieurs types de thermomètres en fonction des phénomènes qui existent, notamment les thermomètres électroniques que l'on rencontre dans de nombreux domaines aujourd'hui, car tout est électronique. Dans le cadre de notre recherche, on nous a demandé d'entreprendre un projet afin d'améliorer nos connaissances et notre capacité à travailler dans le domaine de l'électronique. Le but de ce projet est d'implémenter un thermomètre numérique basé sur une carte Arduino-Uno et une électronique capable d'effectuer la mesure et la surveillance de la température et de l'afficher sur un écran LCD.

Le manuscrit est divisé en trois chapitres :

- Le chapitre 1, donne un aperçu sur les capteurs de température et leur mesure
- Le chapitre 2, se concentrera sur la structure, l'environnement de programmation et le principe de fonctionnement du modèle de base (Arduino-Uno) de la carte d'interface (Arduino), et effectuera des recherches approfondies sur la carte d'interface (Arduino).
- Le chapitre 3, présente la simulation et la réalisation de notre circuit électronique pour mesurer et surveiller la température
- Enfin, nous concluons par une conclusion générale résumant les intérêts de notre recherche.

# **Chapitre I**

## Généralités sur les capteurs de la température



## I.1 Introduction

Notre monde d'aujourd'hui est un monde de capteurs. Nous les rencontrons partout autour de nous où nous allons. Nous les retrouvons sous toutes leurs formes, dans notre voiture, dans l'avion que nous avons pris ou que nous prendrons un jour. Les capteurs améliorent notre vie et nous facilitent nos tâches, ils allument la lampe de la chambre dès qu'ils détectent notre présence, règlent automatiquement la température de la maison, nous alertent en cas d'incendie ou de fumée, nous ouvrent la porte dès qu'on s'en approche, et apportez-nous le café que nous adorons et bien d'autres tâches diverses et variées. Toutes les tâches que nous avons mentionnées précédemment et de nombreuses autres tâches d'automatisation sont possibles et faciles à mettre en œuvre sur le terrain grâce aux capteurs. Et ici nous allons en apprendre davantage sur les capteurs en générale, et sur les capteurs de température en particulier.

## I.2 Unités et conversion de température

Une méthode fiable et polyvalente pour identifier les différences de température corporelle ou de facteurs atmosphériques. L'échelle se compose d'unités. Cinq échelles ont été identifiées dans le monde, dont Fahrenheit, Celsius, Kelvin, Rankine et Réaumur. [1]

**1.Fahrenheit (°F):** Daniel Gabriel Fahrenheit, physicien allemand qui a construit le premier thermomètre à mercure pratique. Il a utilisé un mélange de glace, d'eau et de sel marin qu'il estime être à 0°F et utilise la formule pour la température corporelle de 96°F. Après avoir divisé l'intervalle en 96 parties, il a trouvé que l'eau gelée est à 32°F et que l'eau bouillante est à 212°F.

**2.Celsius (°C) :** L'astronome et physicien suédois Andres Celsius a construit son thermomètre à mercure en 1742. Il choisit de faire fondre de la glace à 0°C et de faire bouillir de l'eau à 100°C.

**3.Kelvin (°K) :** Sir William Thomson, Lord Kelvin a proposé une échelle absolue utilisant le zéro absolu (= -273K) comme température d'origine. Il reprend le même intervalle de tick au-dessus de Celsius (0°C = 273°K). Donc si l'eau gèle à 0°C, cela équivaut à geler à 273°K.

**4.Rankine (°R ou °RA) :** En 1859 L'Écossais William Rankine a introduit pour la première fois l'échelle de Rankine. Comme l'échelle Kelvin, le point de référence de l'échelle Rankin est le zéro absolu à 0°R. Rankin a la même taille que Fahrenheit, mais zéro est très différent. Le point de congélation de l'eau est égal à 491,67° Rankine.

**5.Réaumur (°Ré) :** En 1730, René de Réaumur introduit la gamme Réaumur. Ses points de référence sont le point de congélation de l'eau, 0°Re, et le point d'ébullition de l'eau, 80°Re.

Le tableau (I.1) ci-dessous illustre la formule pour convertir les lectures de température d'une unité à une autre.

De/En	En °C	En °F	En °K	En °Ra	En °Ré
De °C	1	$T(^{\circ}\text{C}) \times 1.8 + 32$	$T(^{\circ}\text{C}) + 273.15$	$(T(^{\circ}\text{C})+273.15)\times 1.8$	$T(^{\circ}\text{C}) \times 0.8$
De °F	$(T(^{\circ}\text{F}) - 32)/1.8$	1	$(T(^{\circ}\text{F}) - 32)/1.8 + 273.15$	$T(^{\circ}\text{F}) + 459.67$	$(T(^{\circ}\text{F})-32)\times 4/9$
De °K	$T(^{\circ}\text{K}) - 273.15$	$T(^{\circ}\text{K}) \times 1.8 - 459.67$	1	$T(^{\circ}\text{K}) \times 1.8$	$(T(^{\circ}\text{K})-273.15)\times 0.8$
De °Ra	$(T(^{\circ}\text{Ra}) - 491.67)/1.8$	$T(^{\circ}\text{Ra}) - 459.67$	$T(^{\circ}\text{Ra}) / 1.8$	1	$(T(^{\circ}\text{Ra})-491.67)\times 4/9$
De °Ré	$T(^{\circ}\text{Ré}) / 0.8$	$T(^{\circ}\text{Ré}) \times 9/4 + 32$	$T(^{\circ}\text{Ré})\times 1.25+273.15$	$T(^{\circ}\text{Ré})\times 9/4 + 273.15$	1

**Tableau I.1 :** Formule de calcul pour la conversion entre unités de température

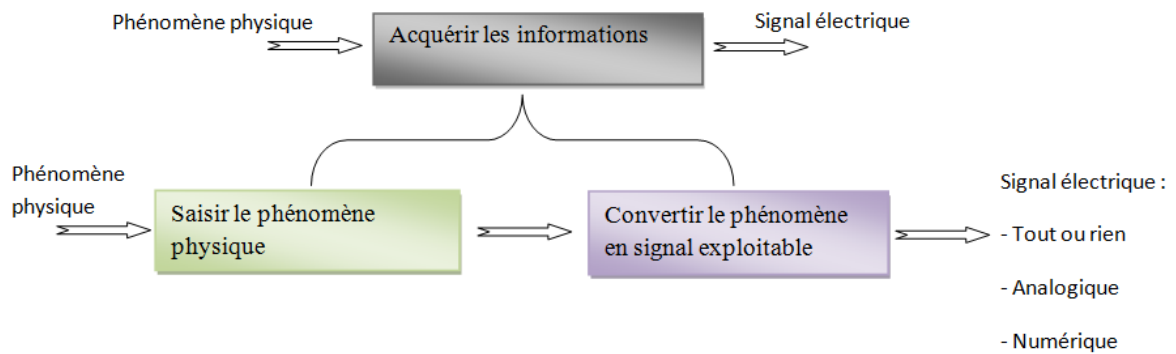
**I.3. Définition d'un capteur :**

Les capteurs sont des composants de la chaîne d'acquisition dans un système fonctionnel. Ce sont les premiers éléments rencontrés dans une chaîne de mesure, ils prélèvent une information sur le comportement de la partie opérative qui sera exploitée par la partie commande c'est-à-dire qu'ils transforment les grandeurs que l'on veut mesurer appelées mesurandes. Cette mesure peut être physique (lumière, pression, chaleur...). Ou chimique (gaz, liquide, acide...) d'un processus ou d'une installation en signaux électriques.[2]

**I.4. Structure d'un capteur :**

Chaque capteur est composé de deux parties :

- Une partie directe, qu'est affectée par la quantité à détecter ou à interpréter : l'objet à tester.
- Une autre fonction de traitement liée à la mise en forme et à la transmission des informations. [3]



**Figure I.1** : Structure d'un capteur

### I.5. Rôle du capteur :

Le rôle d'un capteur est de convertir une grandeur physique en une autre grandeur disponible, et convertir l'information lumineuse en un courant électrique mesurable. Un capteur va donc absorber l'énergie portée par les photons pour créer un flux d'électrons.

### I.6. Capteur de température :

Les capteurs de température sont des composants électriques et électroniques qui permettant de mesurer la température au moyen d'un signal électrique déterminé. Ils peuvent envoyer ce signal directement ou indirectement en modifiant la résistance. Ils sont également appelés capteurs thermiques. Entre autres choses, les capteurs de température sont utilisés pour contrôler les circuits de commutation. Les capteurs de température sont également appelés détecteurs de chaleur, sondes de température ou capteurs thermiques.[4]

L'objectif du capteur de température est d'obtenir une tension  $U$  (en V) proportionnelle à la température  $\theta$  (en °C) telle que :  $U = k \cdot \theta$

#### I.6.1. Capteurs à contact :

Ils nécessitent de contact direct avec la grandeur qu'on veut capter.

#### I.6.2. Capteurs de proximité :

Qui ne nécessitent pas de contact direct avec la grandeur mais il suffit de l'approcher de cette dernière pour obtenir l'information ; chacun de ces deux types se divise en trois catégories qui sont le capteur mécaniques, électrique et pneumatiques.

### I.6.3. Les capteurs actifs :

En tant que générateur, un capteur actif convertit généralement de l'énergie en énergie électrique en fonction d'effets physiques qui assurent la conversion en grandeur physique à prélever : thermique, mécanique ou radiante. [5]

Les plus classiques sont :

- ✚ **Effet thermoélectrique** : Un circuit formé de deux conducteurs chimiquement distincts avec des températures de jonction  $T_1$  et  $T_2$  est le lieu où se situe la force électromotrice  $e(T_1, T_2)$ .
- ✚ **Effet piézo-électrique** : L'application de contraintes mécaniques à certains matériaux dits piézoélectriques, comme le quartz, entraîne une déformation et l'apparition de charges identiques de signes différents sur des faces opposées.
- ✚ **Effet d'induction électromagnétique** : Un changement de flux magnétique dans un circuit induit une tension électrique.
- ✚ **Effet photo-électrique** : Libération de charges électriques dans un matériau sous l'influence d'un rayonnement optique ou, plus généralement, d'ondes électromagnétiques de longueurs d'onde inférieures au seuil caractéristique du matériau.
- ✚ **Effet Hall** : Le champ électrique  $B$  produit un champ électrique vertical  $E$  dans le matériau.
- ✚ **Effet photovoltaïque** : l'électrons et le trou sont libérés près de la jonction PN éclairée et leur mouvement modifie la tension à leurs bornes.[6]

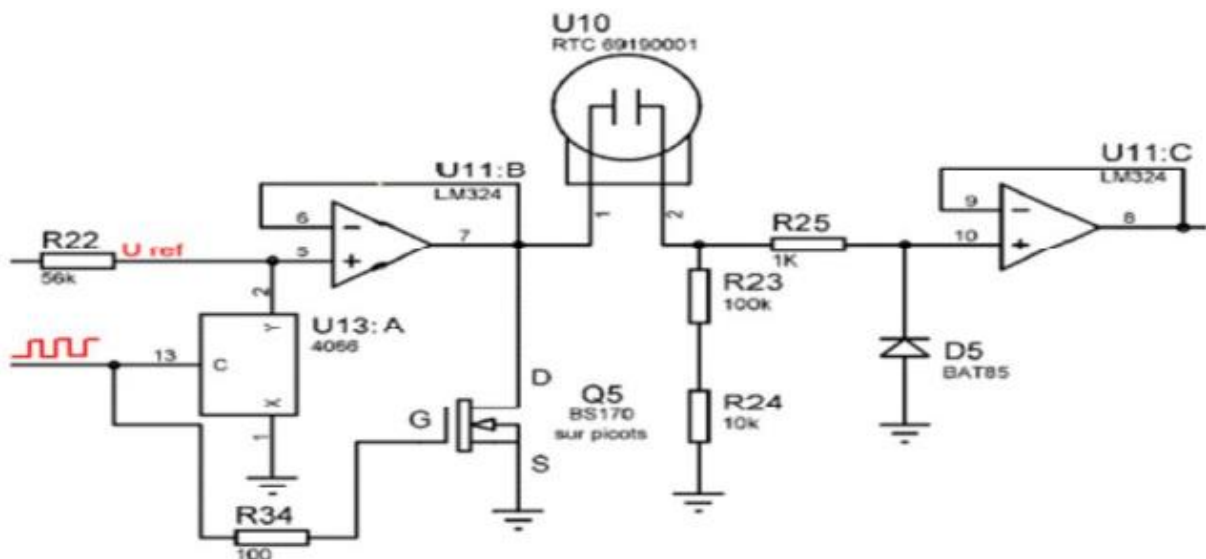


Figure I.2 : Schéma d'un capteur actif

### I.6.4. Les capteurs passifs

Ils ont besoin dans la plupart des cas d'apport d'énergie extérieure pour fonctionner. Ce sont des capteurs modélisables par une impédance. Des changements dans le phénomène physique étudié produisent des changements d'impédance. Une tension doit leur être appliquée pour obtenir un signal de sortie. [5]

Généralement l'impédance, l'un des paramètres déterminants est sensible à la grandeur mesurée. Le changement d'impédance se traduit par :

- Qu'il s'agisse d'un changement de taille de capteur, c'est ainsi que fonctionnent un grand nombre de capteurs de position, potentiomètres, inducteurs à fer mobile, condensateurs à armature mobile.
- Déformation due à la réduction de la force ou de l'amplitude, accélération de la pression (Armature de condensateur soumise à un différentiel de pression, extensomètre relié à une structure déformable).

L'impédance des capteurs passifs et leurs modifications ne peuvent être obtenues qu'en intégrant le capteur dans un circuit, qui est également alimenté et qui est son régulateur.[6]

### I.6.5. Les capteurs de températures les plus connus :

Comme toutes les technologies, les capteurs de température ont parcouru un long chemin au fil des ans. Aujourd'hui, il existe trois types principaux utilisés dans l'industrie :

#### 1. Thermocouple :

Il se compose de deux propriétés différentes reliées à une extrémité. Lorsqu'une jonction métallique est chauffée ou refroidie, une tension changeante est produite, qui peut ensuite être convertie en température. Les alliages de thermocouple sont généralement fournis sous forme de fil. [7]



**Sonde thermocouple avec  
Connecteur M12 surmoulé**



**Thermocouples hauts  
température et exotiques**



**Thermocouple,  
hermétiquement Scellé**

**Figure I.3:** Types de thermocouples

Le tableau (I.2) indique les thermocouples les plus utilisés :

Lettre	Conducteur positif	Conducteur négatif
<b>T</b>	Cuivre	Nickel-Cuivre (Constantan)
<b>J</b>	Fer	Constantan
<b>E</b>	Nickel-chrome (chromel)	Constantan
<b>K</b>	Chromel	Nickel-Aluminium (Alumel)
<b>S</b>	Platine-10% Rhodium	Platine-6 % Rhodium
<b>R</b>	Platine-13% Rhodium	Platine
<b>B</b>	Platine-30% Rhodium	Platine-6 % Rhodium
<b>N</b>	Nickel-Chrome-Silicium (Nicrosil)	Nickel-Chrome-Magnésium (Nisil)
<b>U</b>	Cuivre	Constantan
<b>G</b>	Tungstène	Tungstène-26 % Rhénium
<b>C</b>	Tungstène-5 % Rhénium	Tungstène-26 % Rhénium
<b>D</b>	Tungstène-3 % Rhénium	Tungstène-25 % Rhénium

**Tableau I.2:** Lettres de référence des thermocouples les plus couramment utilisés. [8]

Le domaine d'utilisation est souvent l'un des premiers facteurs de sélection d'un capteur, ainsi que sa sensibilité.

Le tableau (I.3) donnera les valeurs de ces deux paramètres pour certains capteurs.

Lettre	Domaine d'utilisation (°C)	Sensibilité moyenne ( $\mu\text{V}/^\circ\text{C}$ )
<b>T</b>	-200 à 370	<b>51</b>
<b>J</b>	-40 à 800	<b>55</b>
<b>E</b>	-270 à 870	<b>78.5</b>
<b>K</b>	-270 à 1270	<b>41</b>

<b>S</b>	-50 à 1600	<b>11.4</b>
<b>R</b>	-50 à 1600	<b>12.9</b>
<b>B</b>	0 à 1700	<b>10.6</b>
<b>N</b>	-270 à 1300	<b>38</b>
<b>G C D</b>	0 à 2600	<b>20</b>
<b>Pt/Au</b>	0 à 1000	<b>25</b>

**Tableau I.3 :** plage de température et sensibilité moyenne de certains capteurs [8]

**2. Capteur de température à résistance (RTD) :**

Les capteurs RTD mesurent la température en fonction des changements de résistance à l'intérieur d'une résistance métallique. Le RTD le plus populaire, appelé capteur PT100, utilise du platine et a une résistance de 100 ohms à 0 °C. [7]



**Capteurs RTD  
Hermétiquement Scellés**



**Capteurs de surface RTD avec  
caoutchouc de silicone autocollant**

**Figure I.4:** Capteurs à résistance RTD

**3. Thermistance :**

Un capteur à thermistance est un corps de détection de température en matériau semi-conducteur, qui se caractérise par une grande variation de résistance proportionnel à un petit variation de température. Les thermistances ont un coefficient de température négatif (également appelé thermistance CTN), ce qui signifie que la résistance de la thermistance réduit à mesure que la température augmente. En plus, les thermistances ont un coefficient de température positif (également appelé thermistance CTP).

Les caractéristiques des thermistances de type CTP et CTN sont reportées sur la figure I.5 et le symbole de capteur est reporté sur la figure I.6:

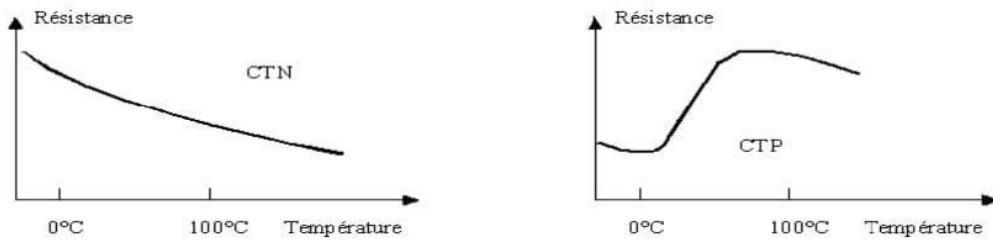


Figure I.5 : Courbes caractérisant la CTP et CTN.[9]

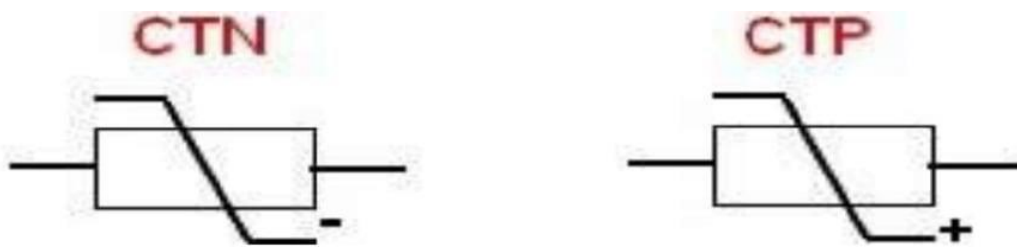


Figure I.6 : Représentation schématique [9]

Les paramètres de base de la thermistance sont :

- ✓ Sa valeur de résistance.
- ✓ Sa sensibilité thermique ou coefficient de température.
- ✓ Sa stabilité (fournie par le fabricant).
- ✚ **Les avantages** : temps de réponse rapide, et prix bon marché.
- ✚ **Les inconvénients** : Loi des propriétés non linéaires en série, sensible aux variations d'auto-échauffement et de résistance de connexion.



Sondes de thermistance  
Avec connecteurs M12



Capteur de température  
De l'air de thermistance

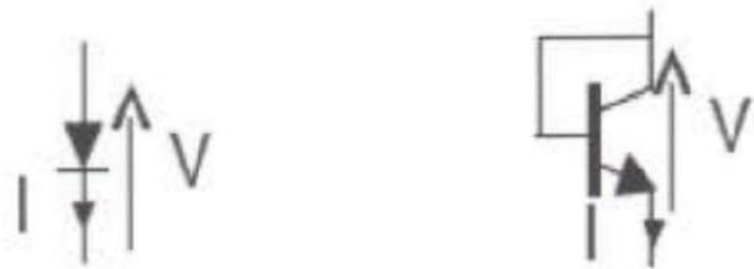
Figure I.7 : Capteurs de thermistance



#### 4. Les capteurs de température à semi-conducteur :

Une diode au silicium ou une jonction de transistor alimentée par un courant constant produit une tension proportionnelle à la température, cette caractéristique est utilisée pour fabriquer des capteurs. Typiquement, un circuit intégré est ajouté pour conditionner le capteur. Il existe également des circuits intégrés qui permettent un enregistrement de température à long terme.[9]

La tension à une borne semi-conductrice (formant une diode ou un transistor) et le courant qui la traverse dépendent de la température :



**Figure I.8** : A courant constant, la tension  $V$  est linéaire en fonction de la température [9]

Il existe aussi une mesure de température basée sur une mesure du bruit de fond, qui représente le matériau Agitation thermique des neutrons. L'un des avantages des capteurs semi-conducteurs : Les semi-conducteurs ont un cout faible, La simplicité de mise en œuvre et la bonne linéarité.

#### 5. Les pyromètres optiques :

La pyrométrie optique est une méthode de mesure de la température basée sur la relation entre la température d'un objet et le rayonnement optique (infrarouge ou visible) émis par cet objet. Ainsi, les capteurs utilisés sont des capteurs optiques, photoélectriques ou thermiques. Le but de la pyrométrie optique est de permettre de déterminer la température sans toucher l'objet.

Les pyromètres optiques sont principalement utilisés dans des conditions ne supportant pas les capteurs conventionnels, telles que : [10]

- Température très élevée (plus de 2000 °C) ;
- les mesures à longue distance ;
- les environnements très sévères ;
- Pièces mobiles.

### I.7. Classification des capteurs :

Les capteurs de température sont classés selon :

✚ **Apport énergétique** : soit être un capteur actif ou capteur passif

✚ **Type de détection** :

- Détection de contact (le capteur doit être en contact physique avec le phénomène pour le détecter).
- Détection sans contact (le capteur détecte un phénomène proche de lui. [11])

✚ **Type de sortie** : Le capteur et son régulateur peuvent également être classés par type de sortie :

- Capteur logique
- Capteur analogique
- Capteur numérique

#### I.7.1. Signal logique :

Capteurs logiques ou dits capteurs tout ou rien (TOR), la sortie est un état logique, transmettant une information binaire à la partie commande : l'information prend l'état 0 ou l'état 1. Chaque état a une signification dans le contexte du système.

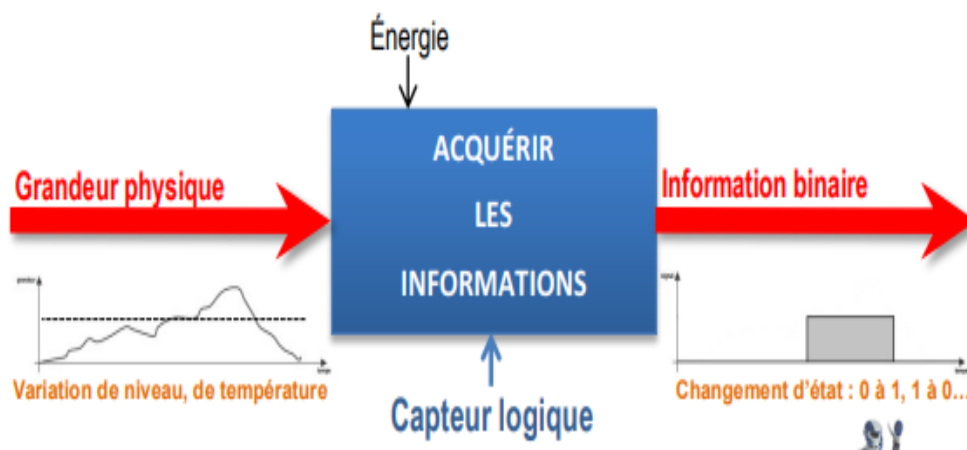


Figure I.9: Structure de capteur logique

Les signaux logiques ne prennent que deux états (valeurs) de niveau haut et de niveau bas. Les deux états de liaison sont généralement représentés par une seule charge. La tension est la plus courante, mais le courant est utilisé dans certains cas. Dans le cas de la tension, cette région est définie pour indiquer si l'on a action à une tension de niveau haut (1 logique) ou à un niveau bas (0 logique). [12]

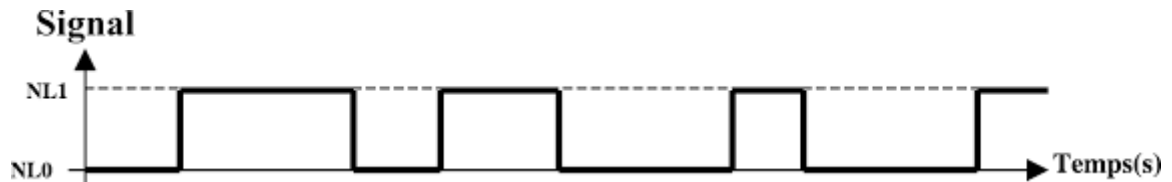


Figure I.10 : Signal logique en fonction du temps

I.7.2. Signal analogique :

Les capteurs analogiques transmettent un signal analogique, qui est un signal continu qui change dans le temps. Sa magnitude est également continue, ce qui signifie qu'elle peut prendre toutes les valeurs possibles (même décimales). [12]

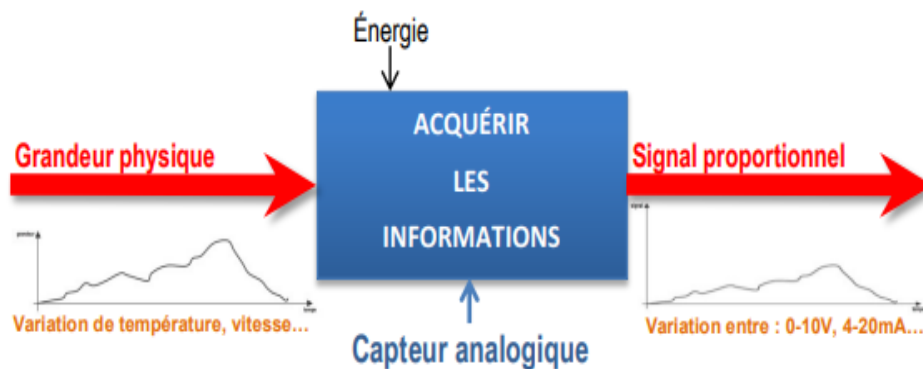
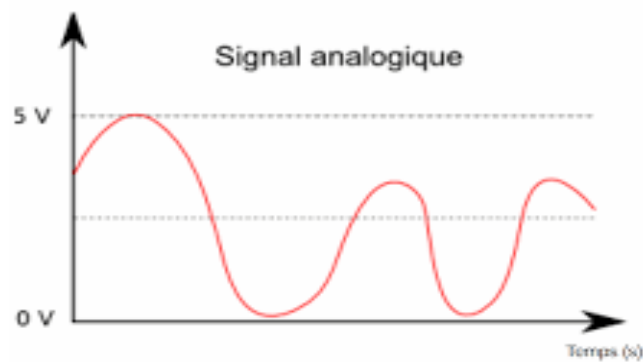


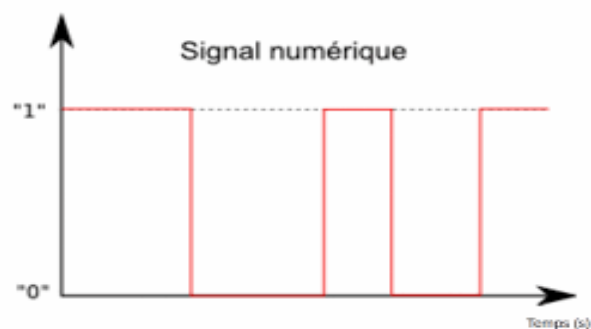
Figure I.11 : Structure d'un capteur logique



**Figure I.12 :** Signal analogique

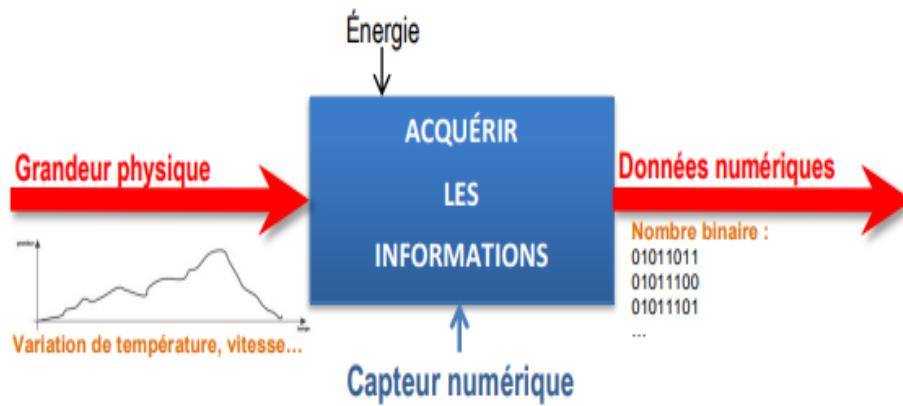
### I.7.3. Signal numérique :

Un signal électronique transmis sous forme codée binaire qui indique la présence ou l'absence de courant électrique, de niveaux de tension élevés et faibles ou de courtes impulsions d'une fréquence spécifique. Par rapport aux signaux analogiques, la transmission basée sur la technologie numérique offre des textes, des images et du son de meilleure qualité, ainsi que des fonctions supplémentaires.



**Figure I.13 :** Signal numérique

Les capteurs numériques produisent des nombres binaires (combinaisons de signaux logiques : 0000, 0001, 0010, etc.) à des intervalles fixes (fréquence), qui dépendent directement de la grandeur physique à capter. Les signaux sont encodés au sein des capteurs eux-mêmes par une électronique associée, ces capteurs sont également appelés encodeurs et compteurs. [12]



**Figure I.14 :** Schéma d'un capteur numérique

### I.8. Chaîne de mesure :

C'est un ensemble d'éléments nécessaires pour comprendre les valeurs des paramètres physiques d'un système.

La structure de base de la chaîne de mesure comprend au moins trois étapes :

- Des capteurs sensibles aux variations de grandeurs physiques et transmettant des signaux en fonction de ces variations.
- Les conditionneurs de signal dont la fonction principale est d'amplifier le signal du capteur à un niveau compatible avec l'affichage ou l'unité d'abonné ; parfois, des filtres peuvent être intégrés à ce stade pour réduire les interférences avec 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...).

Une unité d'affichage et/ou d'utilisation qui peut lire la valeur d'une grandeur et/ou l'utiliser dans le cas d'un asservissement par exemple.

Cette structure de base est présente dans toutes les chaînes de mesure, indépendamment de la complexité et de leur nature. Aujourd'hui, avec les capacités fournies par l'électronique et l'informatique, les capteurs fournissent des signaux électriques et la quasi-totalité des chaînes de mesure sont électroniques [13].

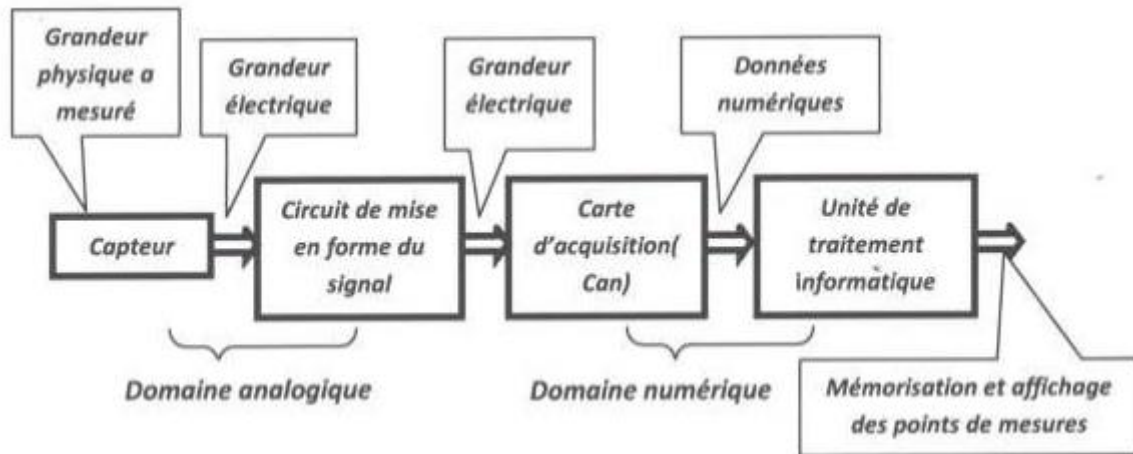


Figure I.15 : Chaîne de mesure informatisée

### I.9. Caractéristiques d'un capteur :

Les principales caractéristiques du capteur sont :

- ✚ **La résolution** : Le plus petit changement dimensionnel que le capteur peut mesurer.
- ✚ **La sensibilité** : Les modifications du signal de sortie sont liées aux modifications du signal d'entrée.
- ✚ **La précision** : La capacité du capteur à donner des mesures proches de la vraie valeur
- ✚ **Vitesse** : temps de réponse du capteur.
- ✚ **L'étendue de mesure** : La valeur extrême que le capteur peut mesurer.[11]

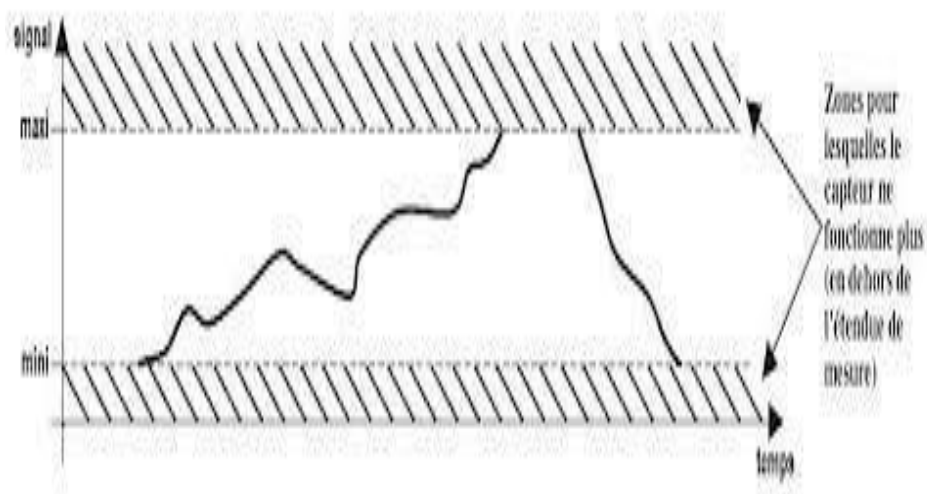


Figure I.16 : L'étendue de mesure : zone en dehors de laquelle le capteur ne transmet plus un signal en relation avec la grandeur physique capté



Consommation	60μA
Gain de la sonde (sortie)	10mV /°C
Précision	±0.75°C

**Tableau I.4 :** Propriétés de LM35

**I.10.3. Différentes versions de LM35 :**

Il existe trois types de boîtiers de capteurs LM35 : TO-46, TO-92 et TO-220, chacun l'une de ses caractéristiques est son champ d'action, qui se compose de plusieurs versions. Quelques exemples de boîtiers et leurs versions :

**TO-46 :**

- LM35AH : boîtier métal, plage -55°C à +150°C
- LM35CH : boîtier métal, plage -40°C à +100°C
- LM35DH : boîtier métal, plage 0°C à +100°C

**TO-92 :**

- LM35DZ : boîtier plastique, plage 0°C à +100°C
- LM35CZ : boîtier plastique, plage -55°C à +150°C

**TO-220 :**

- LM35DT : boîtier plastique, plage 2°C à 150°C. [14]

Le tableau (I.5) explique le brochage des différentes versions du LM335 :

Référence fabricant	Description	Type de capteur	Température de détection - Locale	Précision - La plus élevée
LM335Z/NOPB	SENSOR ANALOG - 40C-100C TO92-3	Analogique, local	40°C ~100°C	±6°C (±9°C)



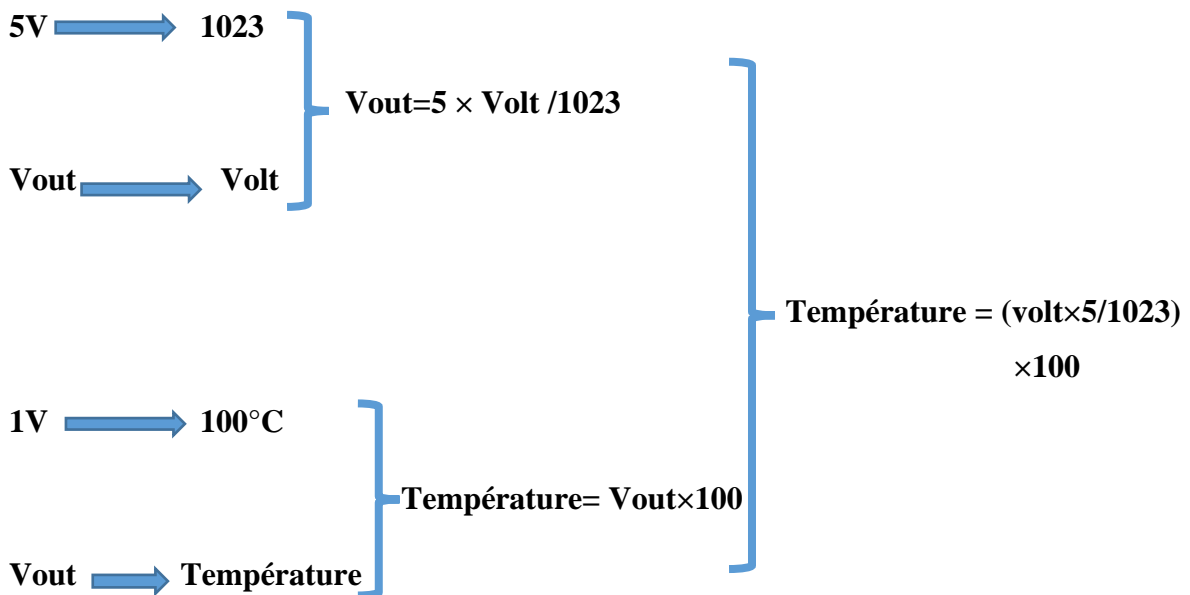
<b>LM335AZ/NOPB</b>	SENSOR ANALOG - 40C-100C TO92-3	Analogique, local	-40°C~100°C	±3°C (±5°C)
<b>LM335MX/NOPB</b>	SENSOR ANALOG - 40C-100C 8SOIC	Analogique, local	-40°C~100°C	±3°C (±5°C)
<b>LM335AH/NOPB</b>	SENSOR ANALOG - 40C-100C TO46-3	Analogique, local	-40°C~100°C	±3°C(±5°C)
<b>LM335AMX/NOPB</b>	SENSOR ANALOG- 40C-100C 8SOIC	Analogique, local	-40°C~100°C	±3°C(±5°C)
<b>M335M/NOPB</b>	SENSOR ANALOG- 40C-100C 8SOIC	Analogique, local	-40°C~100°C	±6°C(±9°C)
<b>LM335AM/NOPB</b>	SENSOR ANALOG- 40C-100C 8SOIC	Analogique, local	-40°C~100°C	±3°C(±5°C)
<b>LM335H/NOPB</b>	SENSOR ANALOG- 40C-100C TO46-3	Analogique, local	-40°C~100°C	±6°C(±9°C)
<b>LM335M</b>	SENSOR ANALOG- 40C-100C 8SOIC	Analogique, local	-40°C~100°C	±6°C(±9°C)
<b>LM335AM</b>	SENSOR ANALOG- 40C-100C 8SOIC	Analogique,local	-40°C~100°C	±3°C(±5°C)

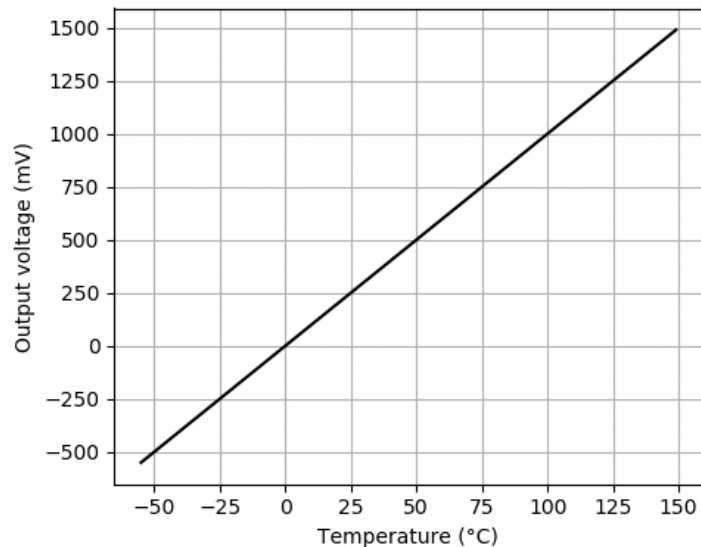
Tableau I.5: Différentes versions du LM335 [15]

**I.10.4. Fonctionnement du capteur de température LM35 :**

Le LM35 est un capteur qui convertit les changements de température en une différence de potentiel dans un rapport potentiométrique. Ce capteur est très simple à utiliser, il suffit de l'alimenter par les deux pattes VCC et GND, la sortie Vout sera récupérée par l'entrée analogique de l'Arduino UNO. La conversion est également facile car un volt correspond à 100 degrés Celsius, et la lecture analogique d'un signal de 0 à 5 V est codée de 0 à 1023 à l'aide de la formule suivante :  $Température = (volt \times 5 / 1023) \times 100$

Volt : est la valeur analogique (Valeur mesuré) convertie en numérique par le CAN.





**Figure I.18 :** Tension de sortie LM35 en mV en fonction de la température

Étapes nécessaires pour assembler le capteur de température :

1. Coupez le fil du capteur de température à environ 5 mm de long.
2. Coupez le fil de la résistance 1K à environ 2,5 mm de long.
3. À l'aide d'une troisième main et d'un fer à repasser propre, étamez les fils du capteur et de la résistance.
4. Connectez une extrémité de la résistance 1K à la broche "+" du capteur de température.
5. Coupez une longueur de câble à trois conducteurs (tel qu'un câble ruban). Je recommande au moins deux à trois pieds de longueur puisque vous voulez placer le capteur n'importe où dans la pièce. Je recommande une longueur maximale de 15 pieds.
6. Divisez et dénudez une extrémité du câble ruban et étamez le fil.
7. Soudez les deux conducteurs du câble ruban à l'extrémité libre de la résistance et la broche "-" de la résistance au capteur de température.
8. Vient maintenant, la partie la plus difficile. Soudez les conducteurs restants du câble ruban aux connexions de la résistance 1K et le « + » plomb du capteur de température. Vous voudrez vous assurer que le conducteur est juste la bonne longueur pour « tomber » en place, parce que si vous devez le plier environ, la tension dans le câble rendra la soudure très difficile.
9. Envelopper le câblage exposé dans de la colle chaude. Assurez-vous de ne pas goblacolle sur le capteur ; vous voulez laisser autant de capteur découvert que possible. Trop de colle peut interférer avec le temps de réponse du capteur.
10. Connectez  $V_{out}$  à l'une des entrées analogiques du HC11.



# **Chapitre II**

## Description générale d'un Arduino UNO

## II.1. Introduction

Maintenant, nous sommes dans un monde développé avec de nouvelles inventions, découvertes et découvertes chaque jour, et l'un des domaines les plus importants et les plus vastes est tout ce qui les entoure, des téléphones aux appareils en passant par les robots, les appareils et les véhicules autonomes.

Mais quelle est la raison de ce grand développement et quelle est la raison de la grande rapidité de ce développement.

La réponse réside dans la facilité d'innovation et d'invention aujourd'hui, où un enfant de 12 ans peut inventer un nouvel appareil, car tout le monde a de l'imagination, et c'est la raison principale de l'invention d'Arduino, et du développement que nous vivons.

Il n'y a pas si longtemps, les inventeurs avaient beaucoup de compétences et d'expérience pour inventer des choses, mais de nos jours, tout ce dont vous avez besoin est une idée et une certaine compétence pour créer quelque chose de nouveau, et c'est tout ce dont nous allons parler.

## II.2. Définition du module Arduino

Arduino est une plateforme similaire pour créer des modèles d'objets publicitaires, dont une carte électronique et d'un environnement de programmation.

Sans tout connaître ou ne pas tout savoir sur l'électronique, cet environnement et ce logiciel permettent aux utilisateurs de créer des tests en direct de leurs projets et d'utiliser de nombreuses ressources disponibles en ligne. [17]

Un Arduino est une carte électronique avec un microcontrôleur qui peut être programmé pour analyser et générer des signaux électriques.

✚ Circuit imprimé : C'est une carte sur laquelle sont soudés plusieurs composants électroniques, reliés entre eux par des circuits plus compliqués.

Donc, Arduino est une carte de circuit imprimé. L'image donne une idée de la taille par rapport à une USB carrée (à gauche de la photo, c'est à dire la même que votre imprimante par exemple). [18]



Figure II.1 : La carte Arduino UNO

- ✚ **Matériel libre** : En fait, le plan de la carte elle-même est gratuit pour tout le monde. Le concept de logiciel libre est important pour les questions de droits de propriété.
- ✚ **Microcontrôleur** : C'est le cœur de la carte Arduino. C'est un micro-ordinateur (ROM, RAM, processeur et entrée/sortie) que nous allons programmer. C'est un grand rectangle noir avec de nombreuses pattes. Une fois lancé et alimenté. L'avantage d'Arduino est qu'il nous offre un microcontrôleur, des entrées/sorties, des connecteurs et de l'alimentation sur une seule carte. La carte Arduino est construite autour d'un microcontrôleur Atmel AVR (pas toujours le même selon la date de sortie de la carte), et l'Arduino UNO a une capacité mémoire de 32000 octets. Soit 32 Ko, ce qui n'est vraiment pas beaucoup, mais permet d'exécuter le plus de projets ! [18]

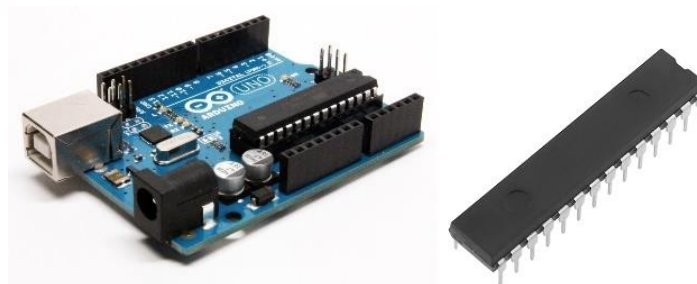


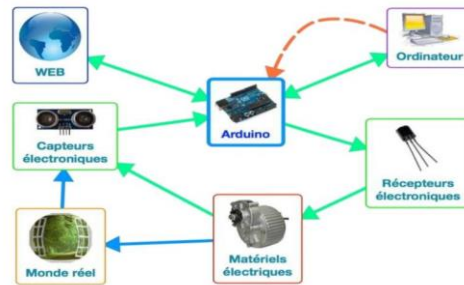
Figure II.2 : Le microcontrôleur de l'Arduino

Donc, Arduino est une carte qui s'y connecte sur l'ordinateur pour être programmée et peut fonctionner toute seule si l'alimentation est disponible.

Il peut recevoir et transmettre des informations provenant ou vers des appareils électroniques.

Vous trouverez ci-dessous un résumé schématique des principales interactions impliquées lors de la programmation de la carte Arduino pour la contrôler.

Les flèches vertes représentent la boucle des signaux électriques, les flèches orange pointillées représentent les programmes entrants, et les flèches bleues représentent les interactions avec le programme réel. [18]



**Figure II. 3 :** Architecture d'un système à base d'une carte Arduino [18]

### II.3. L'intérêt et l'utilité de la carte Arduino

Arduino est un système qui permet de relier les performances de programmation au niveau des produits électroniques, tout en simplifiant les schémas électroniques et les coûts de production. [19].

Le système Arduino nous offre la capacité de combiner les performances de programmation avec les performances des appareils électroniques. Le plus gros avantage de l'électronique pilotée par programme est qu'elle simplifie grandement la production de schémas électroniques, les coûts de production, et la charge de travail lors de la conception de cartes électroniques.

L'utilité est sans doute difficile à percevoir au début, mais une fois entrée dans le domaine d'Arduino, vous serez captivé par les fonctionnalités incroyablement puissantes et les applications possibles discutées ! [20]

### II.4. Applications du module Arduino

On peut réaliser des multiples projets par le module Arduino tels que :

- ✚ Contrôle des appareils électroménagers ;
- ✚ Fabriquer votre propre robot ;
- ✚ Jeux de Lumière ;
- ✚ Communiquer avec le PC ;
- ✚ Contrôle à distance des appareils mobiles ;
- ✚ Télécommander un appareil mobile (modélisme).



À l'aide d'Arduino, nous réaliserons des systèmes électroniques tels que des bougies électroniques, des calculatrices simplifiées, des synthétiseurs, etc. Tous ces systèmes seront conçus à base de cartes Arduino et d'une gamme assez large de éléments électroniques. [20]

### II.5. Les types de la carte Arduino

Arduino lui-même n'est pas une carte ou un microcontrôleur spécifique. Arduino préfère spécifier une famille entière. Choisir le bon pour votre projet n'est pas une tâche facile, et cela vaut la peine de réfléchir à celui que vous utiliserez. Parmi les plus utilisées dans le monde des systèmes embarquées on a : la UNO, la LEONARDO, la DUE, la MEGA et sa petite amélioration la MEGA2560, et la Yun ..... [21]

#### Arduino UNO

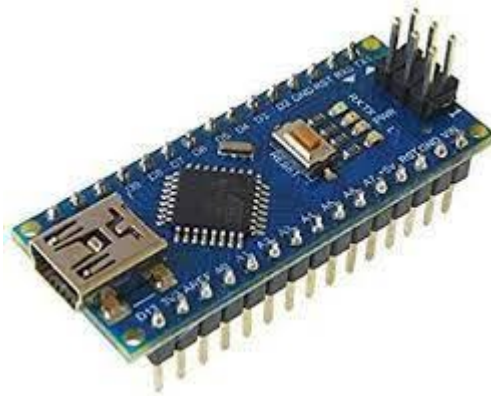
L'Arduino UNO le plus populaire employé. Il est alimenté par un processeur Atmega328 fonctionnant à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'EEPROM, 2 Ko de RAM, 14 E / S numériques, 6 entrées analogiques et un rail d'alimentation de 5V et 3,3V.



Figure II.4 : Carte Arduino UNO

#### Arduino Nano

L'Arduino Nano est particulièrement un Arduino UNO réduit, ce qui le rend idéal pour les espaces restreints et les projets nécessitant le moins de poids possible, tels que la construction de modèles ou les projets DIY de bricolage portable.



**Figure II.5 :** Carte Arduino Nano

Le Nano est exécuté à 16 MHz par un processeur Atmega328 et comprend 32 Ko de mémoire programme, 1 Ko EEPROM, 2 Ko de RAM, 14 E/S numériques, 6 entrées analogiques et des rails d'alimentation 5V et 3,3V.

### Arduino Due

Arduino Due est l'une des plus grandes cartes et la première carte Arduino à être alimentée par un processeur ARM.

L'UNO et le Nano fonctionnent à 5V, tandis que le DUO fonctionne à 3,3V - c'est important car une surtension peut causer des dommages irréparables à la carte. Le Due est alimenté par un Cortex-M3 ATSAM3X8E cadencé à 84 MHz et comprend 512 Ko de ROM et 96 Ko de RAM, 54 broches d'E/S numériques, 12 canaux PWM, 12 entrées analogiques et 2 sorties analogiques.



**Figure II.6 :** Carte Arduino Due

### Arduino Mega 2560

L'Arduino Mega est un peu semblable au Due, il dispose également de 54 E/S. Toutefois, au lieu d'être alimenté par un cœur ARM, il utilise un ATmega2560.



Figure II.7 : Carte Arduino Mega 2560

Le CPU est cadencé à 16 MHz, comprend 256 Ko de ROM, 8 Ko de RAM, 4 Ko d'EEPROM et marche à 5 V, ce qui le rend facile à utiliser avec la majorité des appareils électroniques conviviaux.

### Arduino Leonardo

La carte Arduino LEONARDO est basée sur un ATmega32u4 cadencé à 16 MHz, acceptant de gérer le port USB par un seul processeur. Des connecteurs situés sur le bord extérieur du circuit imprimé permettent de brancher une suite de modules supplémentaires.

Il peut être programmé à l'aide du logiciel Arduino. Le contrôleur ATmega32u4 permet de gérer les ports, ce qui accroît la flexibilité dans la communication avec l'ordinateur.[22]



Figure II.8 : Carte Arduino Leonardo

Le tableau II-1 décrit brièvement quelques importantes caractéristiques des cartes précitées :

Arduino	Microcontrôleur	Flash KO	EEPROM KO	SPAM KO	Broches d'E/S analogique	Broches d'entrée analogique	Vitesse du processeur
UNO	ATmega238P	32	1	2	14	6	16MHZ
LOENARDO	ATmega32U4	32	1	2.5	20	12	16MHZ
MEGA	ATmega1280	128	4	8	54	16	16MHZ
MEGA2560	ATmega2560	256	4	8	54	16	16MHZ
DUE	Atmel SAM3x8E	512	0	96	54	12	84MHZ
Yun	ATmega32u4	23	1	2.5	20	12	16MHZ

**Tableau II.1:** Types cartes Arduino et leur caractéristique [23]

Parmi ces types, nous avons choisi la carte Arduino UNO (carte de base). Le but de cette carte est de faciliter l'exécution d'une telle commande, comme détaillé ci-dessous.

## II.6. Choix de l'Arduino UNO

L'un des modèles de cartes Arduino les plus courants est l'Arduino UNO. Il s'agit de la première version stable de la carte Arduino. En plus d'être simple d'emploi, il possède toutes les fonctionnalités d'un microcontrôleur classique.

Offre simultanément aux parties intéressées les avantages suivants : [24]

- **Le prix (réduits):** 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 préassemblées coûtent moins de 2500 Dinars).
- **Multi plateforme :** le logiciel Arduino, écrit en JAVA, fonctionne sous Windows OS, Macintosh et Linux. La majorité des systèmes de microcontrôleurs fonctionnent seulement sous Windows.

- **Un environnement de programmation clair et simple** : L'environnement de programmation Arduino (logiciel Arduino IDE) est facile à utiliser pour les débutants mais suffisamment flexible pour les utilisateurs avancés.
- **Logiciel Open Source et extensible** : Le logiciel Arduino et le langage Arduino sont publiés sous une licence open source et peuvent être réalisés par des programmeurs expérimentés. Le logiciel de programmation du module Arduino est une application JAVA multiplateforme (fonctionnant sur n'importe quel système d'exploitation) qui agit comme un éditeur de code et un compilateur et peut transférer des programmes via une liaison série (RS232, Bluetooth ou USB, selon le module).
- **Matériel Open source et extensible** : La carte Arduino est basée sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, et les schémas des modules sont diffusés sous la licence Creative Commons. Les concepteurs de circuits expérimentés peuvent créer leur propre version de la carte Arduino pour la compléter et l'améliorer. Même les utilisateurs relativement inexpérimentés peuvent créer une version de planche d'essai d'une carte Arduino dans le but de concevoir comment cela fonctionne pour économiser de l'argent. [24]

## II.7. La propriété de la carte Arduino UNO

La carte Arduino Uno est reposée sur un ATMega328 cadencé à 16 MHz. C'est la carte microcontrôleur la plus simple et la plus économique pour Arduino. Des connecteurs situés sur le bord extérieur du circuit imprimé permettent de brancher une gamme de modules supplémentaires. [25]

### II.7.1. Partie matérielle

Généralement, tout module électronique avec une interface de programmation est continuellement reposé sur la structure d'un ou nombreux circuits programmables. [26]

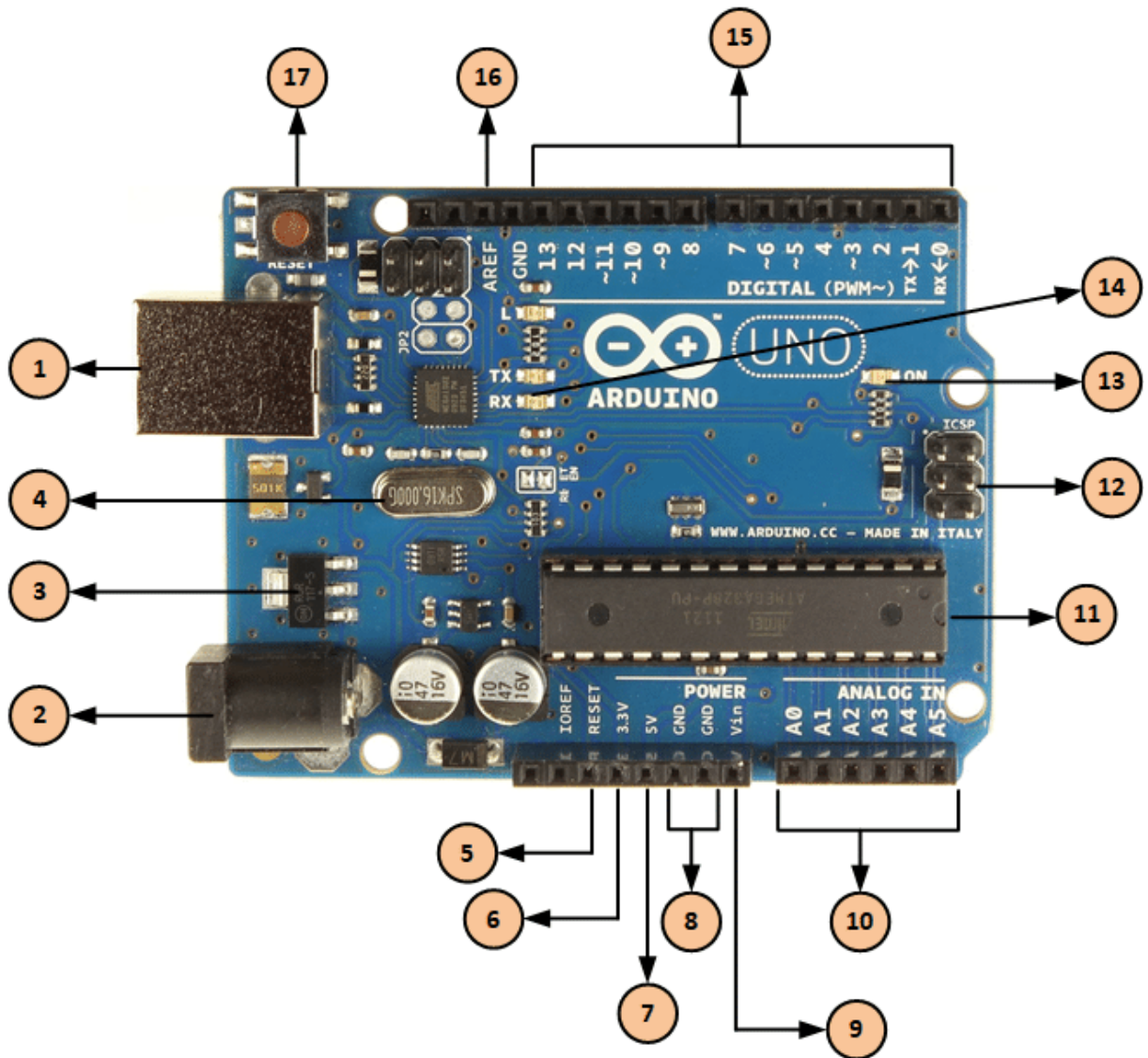





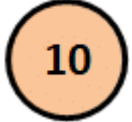


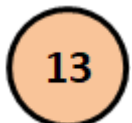


Figure II.9: La carte Arduino UNO [26]

<p style="text-align: center;"><b>1</b></p>	<p><b>Alimentation / Programmation par USB</b></p> <p>La carte Arduino peut être alimentée avec un câble USB relié à votre ordinateur. Tout ce dont vous avez besoin, c'est de connecter votre carte Arduino à votre ordinateur avec le câble USB type A/B.</p>
<p style="text-align: center;"><b>2</b></p>	<p><b>Alimentation via connecteur Jack DC</b></p> <p><i>Diamètre interne 2.1mm, externe 5.5mm</i></p> <div style="text-align: right;">  </div> <p>La carte Arduino peut être directement alimenté par ce connecteur Jack DC. Ce connecteur (2) est relié au régulateur de tension intégré à la carte. L'alimentation via ce connecteur (2) doit être comprise entre 5 et 12 V.</p>

	<p><b>Régulateur de tension</b></p> <p>La fonction du régulateur de tension (3) est de contrôler la tension d'alimentation de l'Arduino pour la stabiliser à la bonne tension du microcontrôleur et de chaque élément de la carte. La tension de stabilisation est de 5 Volts sur les cartes UNO.</p>
	<p><b>Oscillateur à quartz</b></p> <p>Un oscillateur à quartz est un composant électronique qui a la particularité d'avoir du quartz vibrant sous l'effet piézoélectrique à l'intérieur. Les propriétés électromécaniques du quartz lui permettent de vibrer à des fréquences très précises. Ce projet aide Arduino UNO à calculer les données de temps. En haut du composant, nous pouvons voir 16.000H9H. Cela signifie une fréquence de 16 000 000 Hz ou 16 MHz.</p>
	<p><b>Arduino Reset</b></p> <p>Vous pouvez redémarrer l'Arduino par "reset". Cela redémarrera votre programme depuis le début. Vous pouvez redémarrer l'Arduino UNO de deux manières : soit en utilisant le <b>bouton "Reset" (17)</b>, soit en connectant <b>un bouton externe sur la broche de la carte Arduino mentionnée "RESET" (5)</b>.</p>
	<p><b>Broches (3.3, 5, GND, Vin)</b></p> <ul style="list-style-type: none"> <li>- 3.3V (6) – Broche d'alimentation de tension 3.3 Volts</li> <li>- 5V (7) – Broche d'alimentation de tension 5 Volts</li> <li>- La plupart des composants conçus pour fonctionner avec Arduino fonctionneront bien avec 3,3 volts ou 5 volts.</li> <li>- GND (8) (Terre/masse) – Il y a nombreux pins de ce type sur la carte Arduino, elles sont toutes communes et peuvent être utilisées comme masse (potentiel 0 volt) pour le circuit.</li> </ul> <p>Vin (9) - Cette pin est utilisée pour alimenter l'Arduino à partir d'une source de tension externe. Il est connecté au circuit d'alimentation principal de la carte Arduino,</p>

 <p>10</p>	<p><b>Broches analogiques</b></p> <p>L'Arduino UNO dispose de 5 broches d'entrée analogiques, numérotées de A0 à A5. Ces broches sont utilisées pour lire des signaux analogiques provenant de capteurs tels que des capteurs d'humidité ou de température. La carte Arduino utilise un convertisseur analogique/numérique (convertisseur CAN) pour permettre au microcontrôleur de lire les signaux. Un signal sera converti en 10 bits. La valeur peut être lue sur une échelle de 1024 points.</p>
 <p>11</p>	<p><b>Microcontrôleur principal</b></p> <p>Chaque carte Arduino possède <b>son propre microcontrôleur</b> (11). Vous pouvez le considérer comme le cerveau de la carte Arduino. Les microcontrôleurs sur Arduino varient d'une carte à l'autre. Les microcontrôleurs proviennent généralement d'ATMEL. Avant de charger un nouveau programme à partir de l'IDE Arduino, vous devez savoir quel microcontrôleur de votre carte. Ces informations sont disponibles directement sur le composant. Pour plus de structure et de fonctions MCU, veuillez-vous référer à la fiche technique. (<i>Data sheet</i>).</p>
 <p>12</p>	<p><b>Connecteur ICSP</b></p> <p>Tout d'abord, le connecteur <b>ICSP</b> (<i>In-Circuit Serial Programming</i>) est un connecteur AVR, y compris MOSI, MISO, SCK, RESET, VCC et GND. Il s'agit d'un connecteur de programmation. Ce connecteur permet de programmer le microcontrôleur directement au niveau des couches les plus basses (bootloader, code ASM, etc.). C'est aussi un port appelé port SPI (Serial Peripheral Interface) qui permet de dialoguer avec d'autres composants SPI (écrans, capteurs, etc.). On ne va pas se préoccuper de ce connecteur au début des tutoriels.</p>
 <p>13</p>	<p><b>Indicateur LED d'alimentation</b></p> <p>Lorsque vous branchez l'Arduino sur une source d'alimentation, cette LED doit s'allumer pour indiquer que votre carte est correctement alimentée. Si ce voyant n'est pas allumé, cela signifie qu'il y a un problème avec votre alimentation, sans parler de l'alimentation ici.</p>



<p style="text-align: center;"><b>14</b></p>	<p><b>LEDs TX et RX</b></p> <p>Vous trouverez deux voyants sur votre carte : <b>TX (émission)</b> et <b>RX (réception)</b>. Ils apparaissent à deux endroits sur la carte Arduino UNO. Tout d'abord, sur les broches numériques 0 et 1, les broches responsables de la communication série sont indiquées. Deuxièmement, les LED TX et RX (13). Pendant l'envoi de données série, la LED TX clignote à un rythme variable. Le taux de clignotement dépend du débit en bauds utilisé par la carte. RX clignote pendant la réception. La vitesse de transmission est exprimée en bauds, ou en bits par seconde si le signal est binaire.</p>
<p style="text-align: center;"><b>15</b></p>	<p><b>Entrées/Sorties numériques</b></p> <p>La carte Arduino UNO possède <b>14 broches d'Entrées / Sorties numériques</b> (15), dont 6 peuvent fournir une sortie PWM (Pulse Width Modulation). Ces pins peuvent être configurées pour être utilisées comme pins d'entrée numérique pour lire des valeurs logiques (0 ou 1) ou numériques. Elles peuvent également être utilisées comme des pins de sortie pour conduire distincts modules comme des LEDs, des relais, etc. Les pins ordonnées “~” peuvent être utilisées pour produire des PWM.</p>
<p style="text-align: center;"><b>16</b></p>	<p><b>Broche AREF</b></p> <p>AREF est l'acronyme de "Analog Reference". Cette broche est de temps en temps utilisée pour définir <b>une tension de référence externe</b> (entre 0 et 5 volts) comme limite supérieure pour une broche d'entrée analogique.</p>

### II.7.2. Partie programme

Notre implémentation logicielle nécessite deux étapes : la première contient un programme pour injecter le microcontrôleur dans la carte Arduino du code IDE au code HEX, et la seconde contient un programme qui sera traité sous App Inventor Processed et installé sur un smart phone.[27]

### II.7.2.1. Logiciel Arduino

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (plus proche du langage VS) Après avoir tapé ou modifié le programme au clavier, il sera transféré sur la liaison USB et stocké dans la carte. Le câble USB alimente à la fois la carte et transfère les informations à ce programme appelé Arduino IDE. [28]

### II.7.2.2. Présentation de « IDE »

Comme avec n'importe quel langage de programmation, une interface flexible et simple peut fonctionner sur n'importe quel système d'exploitation Arduino basé sur C.

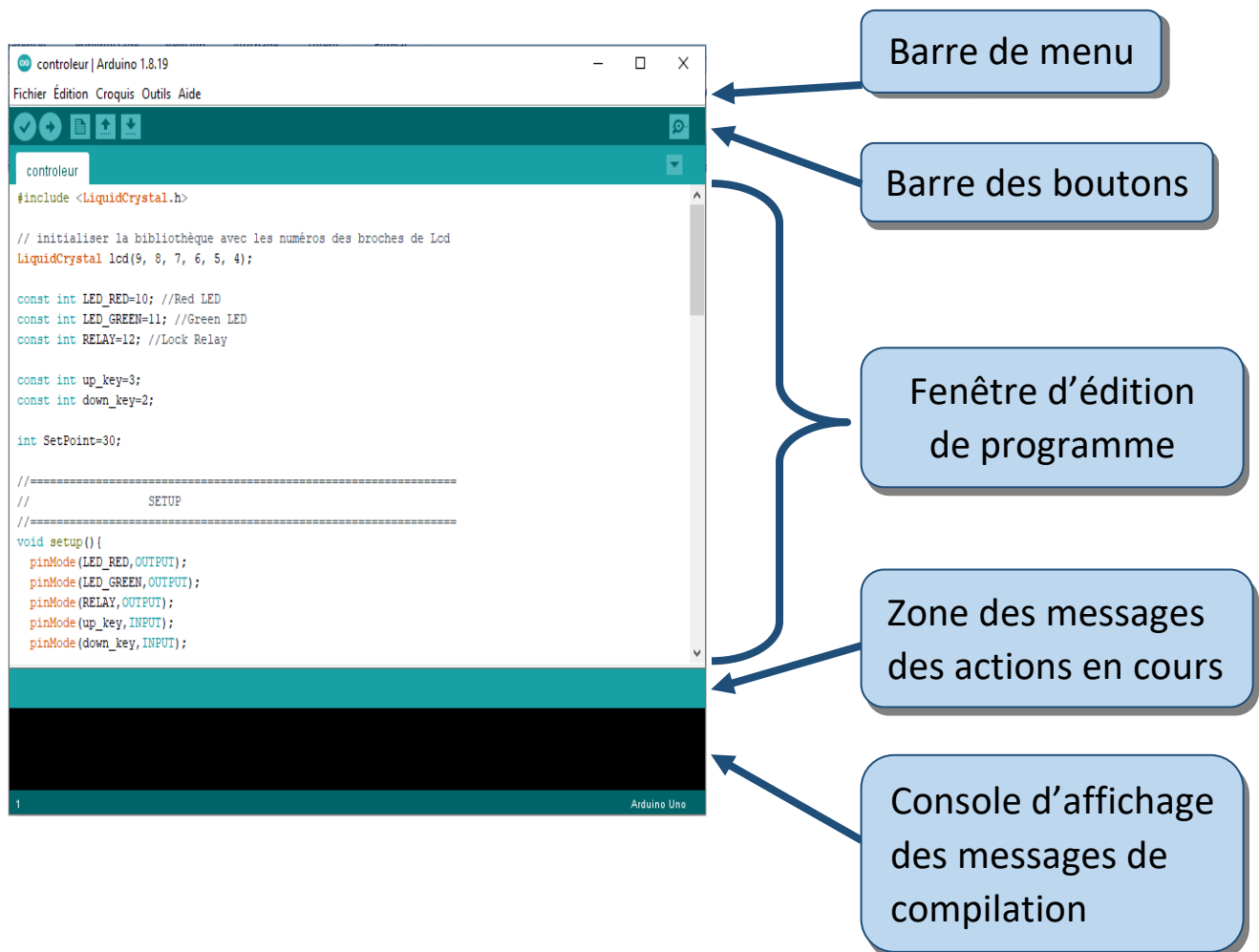


Figure II.10 : Interface IDE Arduino [29]

**II.7.2.3. Programmer avec Arduino**

Un programme utilisateur Arduino est une série d'instructions de base sous forme de texte ligne par ligne. La carte lit ensuite exécute les instructions les unes après les autres dans l'ordonnance défini par les lignes de code. [30]

**II.7.2.4. Programme d'injection**

Avant de téléverser le programme à la carte, il faut choisir le type de carte (Arduino UNO) et le numéro de port USB (COM 3).

**II.7.2.5. Structure des commandes Arduino**

```

// initialiser la bibliothèque avec les numéros des
broches de Lcd
//Tirez vers le haut les key pour consigne(setpoint)
//  Entrée de les points de consigne par l'utilisateur

int SetPoint=30;
const int up_key=3;

void setup(){
  pinMode(LED_RED,OUTPUT);
  pinMode(RELAY,OUTPUT);
  pinMode(up_key,INPUT);
}
void loop(){
  double  Temperature  =  ((5.0/1024.0)  *
analogRead(A0)) * 100;
  lcd.setCursor(0,0);
  lcd.print("Temperature:");
  lcd.print(Temperature);
  if(digitalRead(down_key)==LOW)
  {
    if(SetPoint>0)
    }
  if(digitalRead(up_key)==LOW)
    SetPoint++;
  }

```

**Commentaires :**

Toujours écrire des commentaires sur le programme : soit en multiligne, en écrivant entre des `/***/`, soit sur une ligne de code en se séparant du code avec `//`

**Définition des variables :**

Une variable est un symbole dans lequel il est possible de stocker une valeur. L'affectation de variable est l'action de stocker une valeur dans une variable.

**La configuration des entrées et sorties :**

Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques

**La programmation des interactions :**

Exécution de la boucle sans fin (fonction `loop()`): le code compris dans la boucle sans fin est exécuté indéfiniment.

**Figure II.11 :** Structure des commandes Arduino

### II.7.2.6. Les étapes de téléversement du programme :

Un simple chaînage doit être suivi pour injecter du code dans le menu Arduino via le port USB.

1. Nous utilisons le logiciel Arduino IDE pour concevoir ou ouvrir un programme existant.
2. Nous vérifions ce programme (compilation) avec le logiciel Arduino.
3. Si une erreur est notable, modifiez le programme.
4. Le programme est chargé sur la carte.
5. Câblage des composants électroniques.
6. Le programme s'exécute automatiquement après quelques secondes.
7. La carte est alimentée par le port USB ou par une source d'alimentation séparée telle qu'une pile de 9 volts.
8. nous vérifions que notre montage fonctionne.

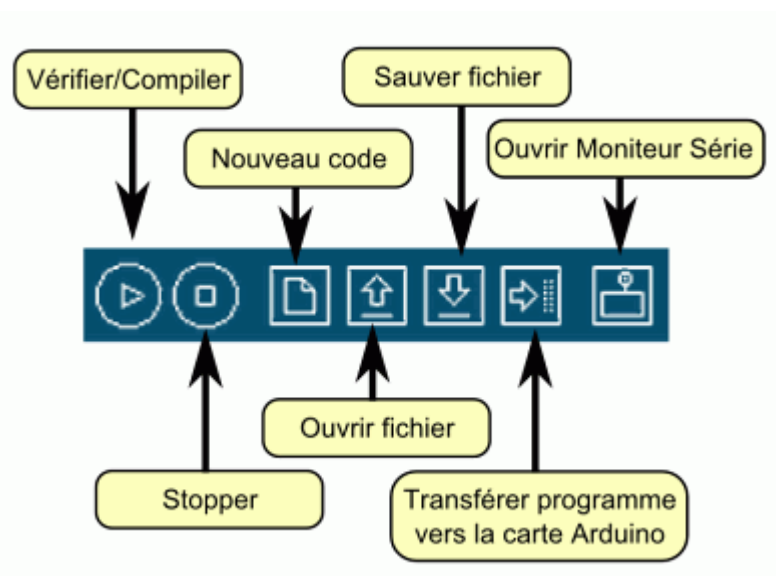


Figure II.12 : Étapes de téléchargement du code

## II.8. Périphériques de la carte Arduino

### II.8.1. Les capteurs

Les capteurs sont l'interface entre les processus physiques et les informations exploitables. Il ne mesure rien, mais fournit des informations en fonction des demandes qu'il accepte. Il fournit ces informations par l'intermédiaire du dispositif électronique qui lui est associé. [31]

### a. Branchement le capteur avec la carte Arduino

Les capteurs (analogiques ou numériques) doivent être connectés au boîtier Arduino, lui-même connecté. Si l'on envisage de brancher un grand nombre de capteurs, il est préférable de brancher la carte Arduino sur un adaptateur secteur (une entrée de la carte est prévue à cet effet). Mais pour 2 ou 3 capteurs de distance IR, normalement, l'alimentation USB de l'ordinateur suffit. [32]

## II.8.2. Les ports de communications

### a. Le module Arduino Bluetooth

Le module microcontrôleur Bluetooth Arduino est une plate-forme Arduino populaire, utilise une connexion série Bluetooth au lieu d'une connexion USB, très faible consommation d'énergie, très courte portée (une dizaine de mètres de rayon), faible débit, très bon marché et compact. [29]



Figure II.13 : Le module Arduino Bluetooth

**b. Le Module XBee**

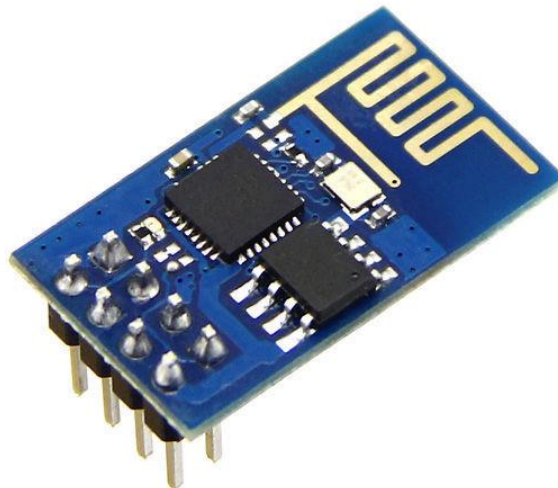
Ce module permet une transmission sans fil, faible distance/consommation/vitesse/prix.



**Figure II.14** : Le Module XBee

**c. Le module shields Arduino Wifi**

Le module Arduino Wifi Shield vous admet de connecter la carte Arduino au réseau Internet sans fil Wifi.



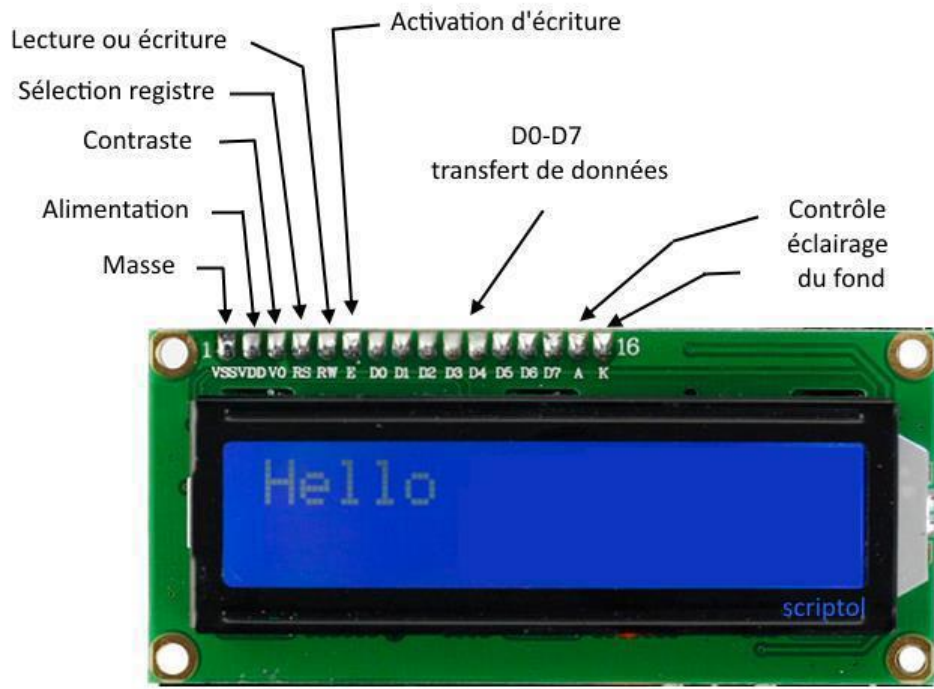
**Figure II.15** : Le module shields Arduino Wifi

### II.8.3. L'afficheur LCD

Aujourd'hui, ces écrans sont utilisés dans les affichages électroniques et ils sont devenus indispensables dans les systèmes technologiques.

Il existe nombreux types de moniteurs LCD sur le marché. Ils diffèrent par leurs caractéristiques techniques, leur tension d'exploitation et leurs dimensions [33].

La figure (II.16) montre un afficheur LCD16x2 :



**Figure II.16 :** L'afficheur LCD16\*2 [34].

#### a. Branchement de LCD (LMO 16L) avec la carte Arduino

L'écran LCD utilise 6 à 10 broches de données ((D0 à D7) + RS + E) et deux alimentations (+5V et masse). Nous allons y insérer une résistance de 10 k $\Omega$ . Ces 10 broches de données peuvent être insérées sur n'importe quelle entrée/sortie numérique de l'Arduino.

Le tableau (II.2) montre le rôle de chaque broche [33].



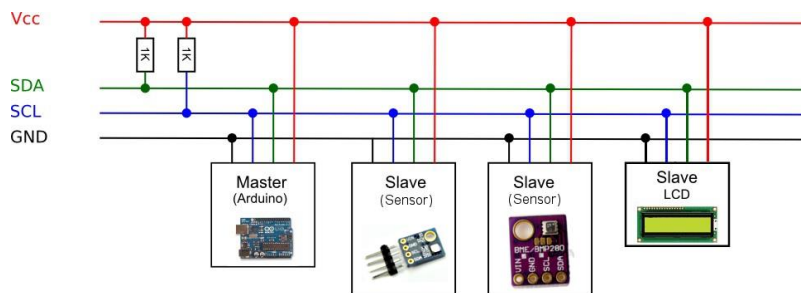
N°	Nom	Rôle de la broche
1	Masse	Alimentation : masse 0 V de l'afficheur
2	Vcc	Alimentation : +5 v de l'afficheur
3	V0	Réglage de contraste entre 0 et +5 V (plus proche de la masse)
4	Rs	Commutation de registre entre les instructions << 0 >> Et les données << 1 >> Si RS=1 on envoie une donnée. Si RS=0 on envoie une commande.
5	R/W	Commutation entre lecture << 1 >> (Read) et écriture << 0 >> (Write)
6	E	Entrée de validation (activation sur le front descendant de l'impulsion positive)
7	D0	Bus de données à trois états : << 0 >>. << 1 >>. Ou haute impédance
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	BI+	Anode de rétro éclairage (+5 V)
16	BI-	Cathode du rétro éclairage (masse)

Tableau II.2 : Rôle des broches d'afficheur LCD 16\*2 [33].

### II.8.4. Interface I2C Arduino pour la connexion du LCD

L'interface I2C est un bus série à deux fils destinés à interconnecter des circuits intégrés à l'intérieur de dispositifs électroniques, connu sous le nom de I2C ou IIC (Inter-Integrated Circuit). I2C a été développé par Philips au début des années 1980 comme un simple bus 8 bits pour la communication interne entre les circuits de l'électronique de commande (par exemple, les ordinateurs sur les cartes mères, les téléphones mobiles, etc.)

Dans un système I2C simple, il peut y avoir plusieurs dispositifs esclaves et un dispositif maître qui initie le transfert de données et synchronise le signal. Plusieurs esclaves (128 appareils maximum) peuvent être connectés aux lignes SDA (ligne de données) et SCL (ligne d'horloge). Le dispositif maître est souvent un contrôleur Arduino et les dispositifs esclaves peuvent être une horloge en temps réel, un écran ou des capteurs.



**Figure II.17 :** Interface I2C Arduino pour la connexion de l'appareil [35]

#### a. Branchement afficheur LCD Arduino avec module I2C

L'écran à cristaux liquides 1602 avec module I2C est connecté à la carte Arduino avec seulement 4 fils – 2 fils de données et 2 fils d'alimentation. La connexion du QAPASS 1602a à l'Arduino se fait de manière standard pour le bus I2C : la broche SDA est connectée au port A4, la broche SCL au port A5. L'écran LCD est alimenté par le port +5V. Consultez l'image ci-dessous pour obtenir un schéma de câblage détaillé de l'écran LCD 1602.

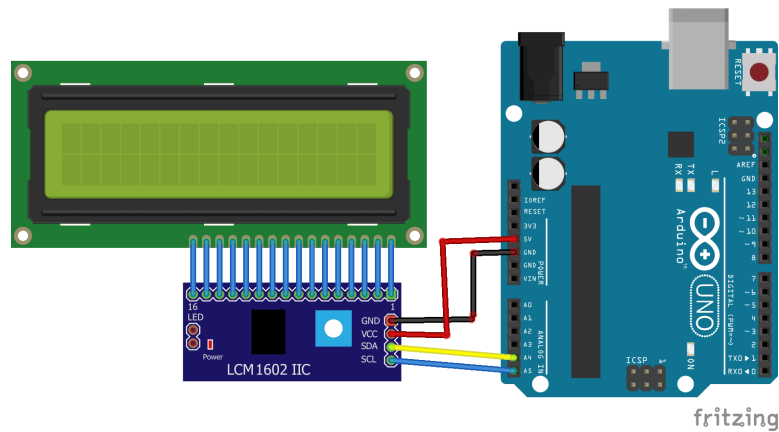


Figure II.18 : Interface I2C Arduino avec LCD16\*2

LCD 1602 I2C	Arduino Uno	Arduino Nano	Arduino Mega
GND	GND	GND	GND
VCC	5V	5V	5V
SDA	A4	A4	20
SCL	A5	A5	21

Tableau II.3 : Branchement de types Arduino avec interface I2C

Par défaut, tous les écrans 1602 dotés d'un module I2C ont l'adresse « 0x27 ». Si vous devez connecter plusieurs écrans 1602 à un microcontrôleur, vous devez modifier les adresses des appareils pour éviter toute coïncidence. Si vous retournez l'écran et regardez le module I2C, vous verrez les broches marquées « A0 », « A1 » et « A2 ». Si par défaut le LCD à l'adresse « 0x27 » sur le bus IIC, en fermant le cavalier « A0 », l'adresse de l'écran passera à « 0x26 ».

### II.8.5. Diode électroluminescente (LED)

LED est une diode électroluminescente, abrégée en LED ou LED (Light-Emitting Diode), qui est un élément électronique qui émet de la lumière lorsque le courant passe. Comme les diodes traditionnelles, les LED permettent au courant de circuler dans une direction (polarisation directe) et bloquent la circulation du courant dans la direction opposée (polarisation inverse) [36].

Par conséquent, chaque LED a sa propre tension de seuil.

**a. Avantages et inconvénients de la LED**

Les diodes électroluminescentes sont un composant de plus en plus incontournable. On trouve sa facilité de montage sur un circuit imprimé, sa faible consommation, sa résistance mécanique, sa petite taille, sa longue durée de vie et d'autres caractéristiques font de la diode électroluminescente un composant de plus en plus inévitable. Certaines critiques, comme le prix de cette technologie dans les écrans de téléviseurs ou la présence de matériaux toxiques dans sa fabrication, tentent cependant de freiner son développement.

Les LED font partie des dispositifs d'éclairage les plus performants du point de vue énergétique, car elles convertissent très peu d'électricité en chaleur. Pour l'instant, les LED bleues demeurent les moins efficaces de la gamme et beaucoup de recherches sont en cours pour la production et l'amélioration des dispositifs de production de lumière blanche à partir de LED.

**b. Domaines d'application des LED**

Actuellement, les diodes électroluminescentes sont des composants de plus en plus utilisés dans différents domaines tels que l'éclairage, les écrans de téléviseurs, les écrans d'ordinateurs ou la décoration. Certains évaluent que cette technologie représentera plus de 75 % de la lumière en 2020.



**Figure II.19 :** Diodes électroluminescentes (Rouge, Blue, Vert).

### II.8.6. Transistor BC548

Le BC548 est un transistor à jonction bipolaire NPN à usage général couramment utilisé dans les équipements électroniques. [37]

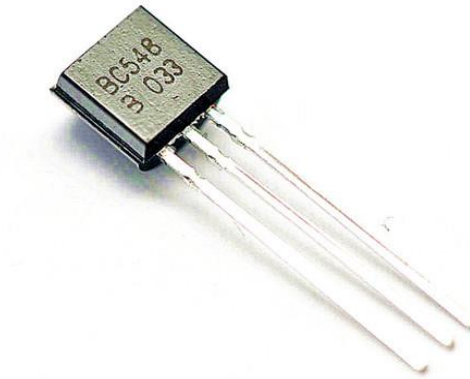


Figure II.20 : Transistor BC548

#### a. Caractéristique de transistor

Type de transistor	<b>NPN</b>
Polarisation	<b>Bipolaire</b>
Tensions collecteur-émetteur	<b>30 V</b>
Courant du collecteur	<b>0.1 A</b>
Puissance de dissipation	<b>0.5 W</b>
Boîtier	<b>TO92</b>
Renforcement du transistor	<b>75...800</b>
Montage	<b>THT</b>
Genre de l'emballage	<b>En vrac</b>
Fréquence	<b>100 MHz</b>

Tableau II.4 : Caractéristique de transistor

### b. Branchement du transistor avec la carte Arduino

Le transistor permet d'utiliser un signal de très faible puissance pour alimenter des modules nécessitant plus de puissance.

En particulier, il peut convertir les signaux MLI des microcontrôleurs en signaux PWM pour les appareils de plus grande puissance.

Les transistors se présentent généralement sous la forme de composants à 3 broches, commun, collecteur et base. Le commun est mis à la terre, la base est connectée à la commande MLI et le collecteur est connecté à la source de courant.[38]

### II.8.7. Le principe de fonctionnement d'un relais

Les relais électroniques sont des interrupteurs commandés par des tensions DC ou AC de faible puissance. La partie de commutation est utilisée pour piloter des charges de secteur à haute puissance, généralement jusqu'à 10 ou 16 Ampères. Un relais est un élément électromécanique qui sépare la partie puissance de la partie commande car ces deux parties sont électriquement isolées. Il se compose de deux parties :

- ✚ Une bobine (partie commande) qui engendre un champ magnétique lors du passage du courant.
- ✚ Contacts ou interrupteurs qui fonctionnent en existence d'un champ magnétique (partie puissance). Le courant passant ce contact varie entre 10 A et 16 A [39].

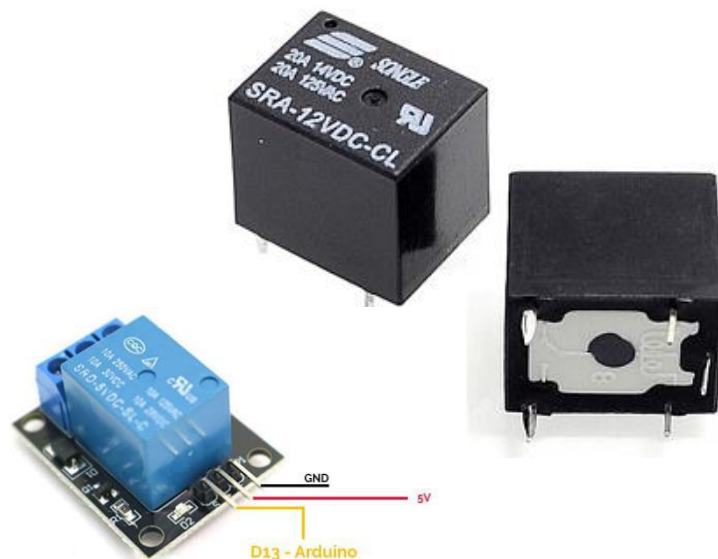


Figure II.21 : Module Relais

### II.9. Schéma synoptique d'un contrôleur de la température

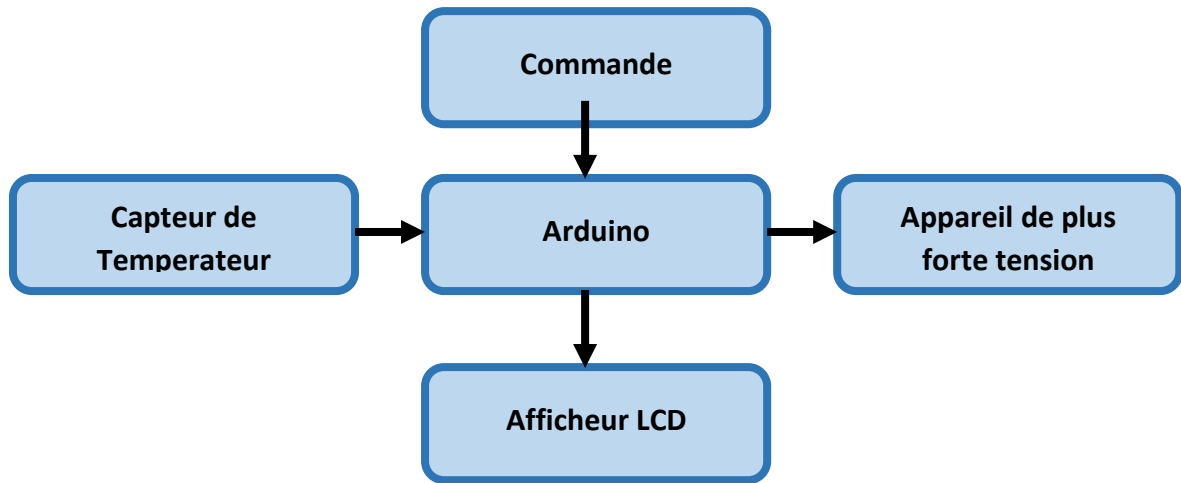


Figure II.22 : Schéma synoptique d'un contrôleur de la température

### II.10. Conclusion

Dans ce chapitre, nous avons présenté les caractéristiques de la carte Arduino, donné les raisons de son choix, puis nous avons cité les différents types de cette dernière. Ensuite, nous avons expliqué les deux parties de base d'une carte Arduino ; la partie matérielle et la partie programmation. Finalement, nous expliquons principe de fonctionnement de la carte Arduino et ses caractéristiques.

# **Chapitre III**

## Réalisation de contrôleur de température numérique par carte Arduino UNO



### III.1 Introduction

Dans ce chapitre nous aborderons brièvement la fabrication d'un contrôleur de température numérique, nous verrons le programme (logiciel) utilisé dans cette conception et comment programmer le microcontrôleur d'Arduino, il nous permet de faire un projet électronique, en plus de le faire tester expérimentalement.

### III.2 Conception générale de l'application

Le principe général de l'appareil est la détection de température et la solution de sécurité mise en place. Pour ce faire, nous utilisons une carte Arduino UNO pour contrôler les composants électroniques (relais, LEDs, interrupteurs tactiles, capteurs de température LM35) qui composent les différentes étapes de notre projet. Notre application vise à implémenter un thermomètre numérique basé sur Arduino UNO, qui nous permet de détecter des valeurs de température dans la plage [-55 °C à 155 °C]. La conception de l'application se compose de deux étapes de base. Avoir :

- a. Assemblage de composants électroniques.
- b. Simulation la carte Arduino UNO ensuite par logiciel ISIS Professional,

### III.3 Schéma synoptique des différents blocs

Notre projet consiste à mettre en œuvre un thermomètre numérique basé sur un capteur LM35 contrôlé par Arduino UNO, qui peut afficher la valeur de température mesurée sur un écran numérique LCD16x2 (Liquid Crystal Display), et la valeur affichée peut être contrôlée par un système de contrôle spécifique.

Le schéma de principe général de notre dispositif est représenté sur la figure (III.1). Pour faciliter cette étude, nous avons décomposé le schéma bloc en quatre blocs :

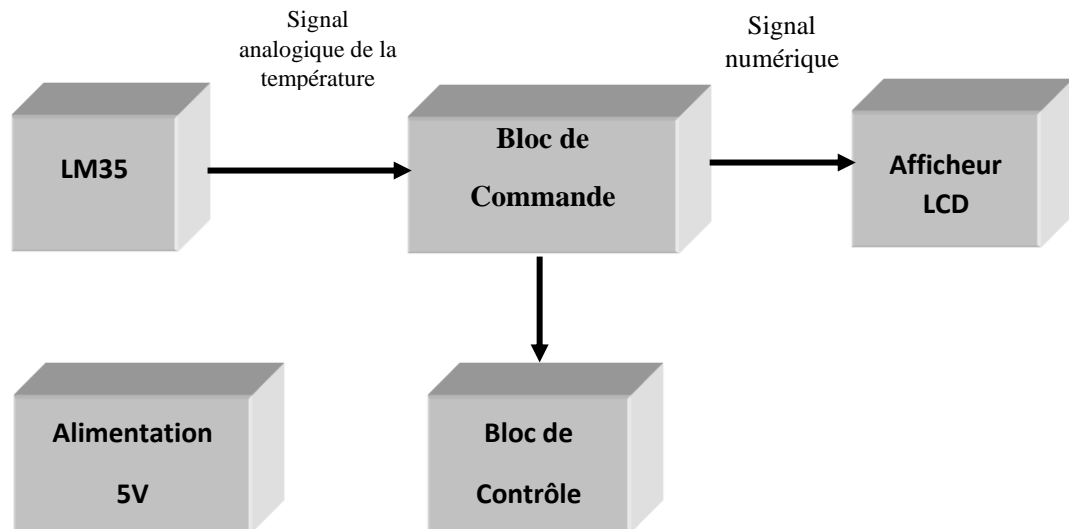
Bloc d'alimentation ;

Bloc de traitement et commande ;

Bloc de contrôle ;

Bloc de d'affichage ;

Quant aux éléments principaux de l'appareil, nous avons choisi l'appareil Arduino UNO.



**Figure III.1** : Schéma synoptique de dispositif de contrôleur de température numérique

## III.4 Réalisation du contrôleur de température

### III.4.1 Partie électronique

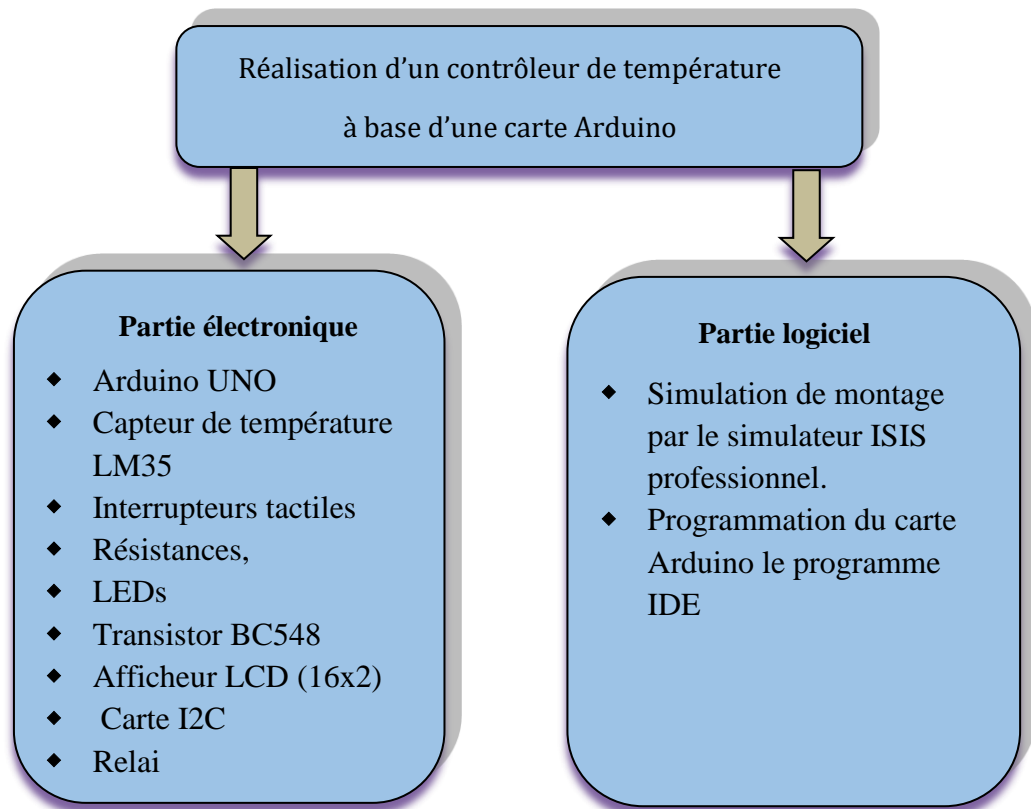
Identification des composants électroniques utilisés dans la conception et la connexion des ensembles électroniques.

### III.4.2 Partie informatique

Pour s'assurer le bon fonctionnement du montage électronique de cet appareil, une simulation de montage doit être réalisée au préalable avec le logiciel ISIS.

L'Arduino UNO dirige l'échange de toutes les opérations, divers traitements, et assure l'accord entre les différentes parties électroniques de l'appareil de mesure.

Par conséquent, l'UNO, afficheur, capteur de température LM35 est alimenté par une tension régulée de 5V



**Figure III.2:** Les deux phases de la réalisation

### III.4.3 Partie Logiciel-Simulation

Pour effectuer ce travail, il faut aller à plusieurs programmes.

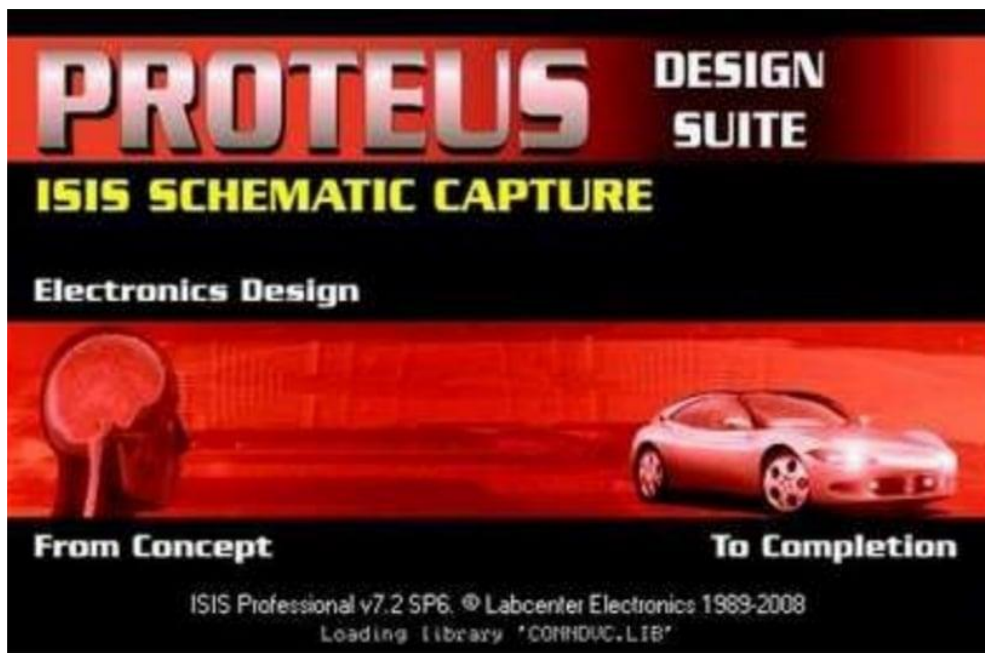
#### III.4.3.1 Présentation de PROTEUS [40]

Le Proteus est une suite logicielle pour électroniques. Il est développé par Labcenter Electronics, le logiciel inclus dans Proteus rend possible la CAO en l'électronique. La suite logicielle se compose des logiciels principaux : ISIS, ARES, PROSPICE et VSM. Il est couramment utilisé par les étudiants en électronique et en programmation. Le logiciel surpasse les autres programmes de simulation dans deux principaux avantages :

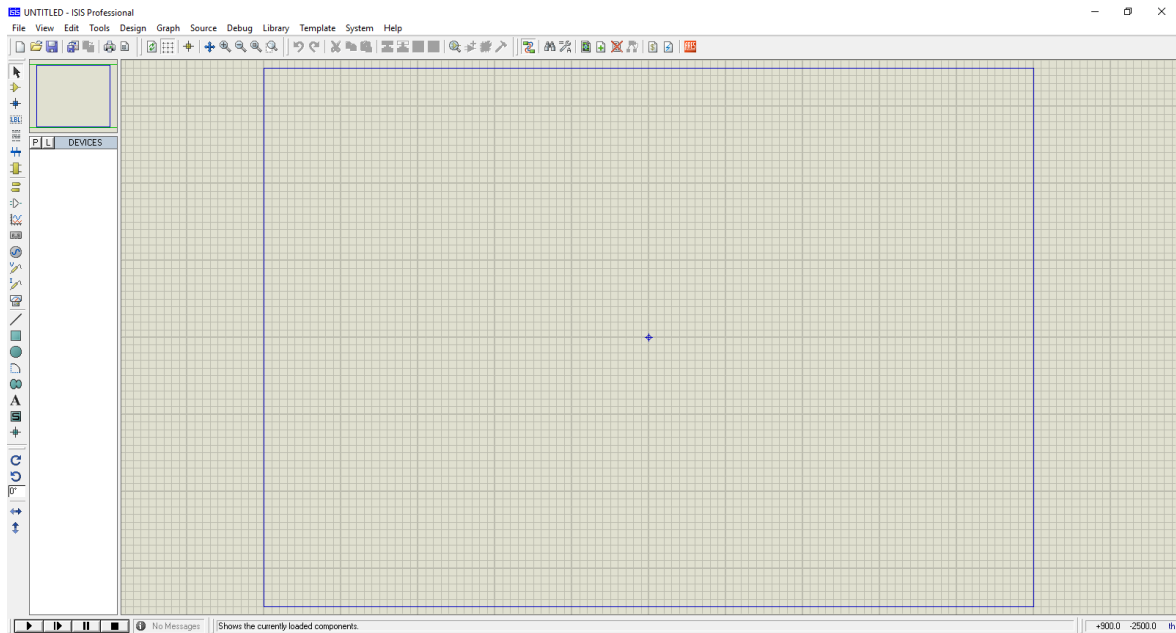
- ✓ Dessiner des circuits électroniques et les préparer pour l'impression sur des cartes de circuits imprimés (PCB).
- ✓ Des prototypes de simulation pour les microcontrôleurs les plus couramment utilisés existent et sont constamment mis à jour.

### **III.4.3.2. Logiciel ISIS [41]**

Ce programme est conçu pour fournir une gestion optimale de la conception et du développement des circuits électroniques introduits car il facilite l'utilisation d'outils puissants dans la construction de circuits. Il peut prendre en charge l'importation de schémas pour effectuer des processus de simulation ou de conception. Il s'agit d'un PCB et il convient de noter que les outils disponibles dans le programme offrent de nombreuses possibilités pour les opérations de dessin et comment l'afficher en termes de lignes de largeur, de motifs de hachures, de lignes d'écriture.



**Figure III.3:** Logiciel ISIS

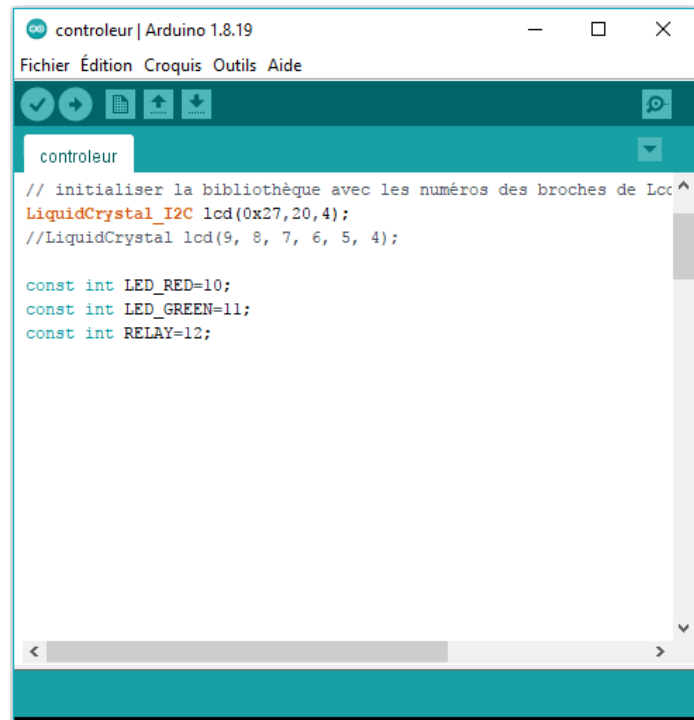


**Figure III.4 :** Fenêtre du démarrage de l'ISIS

### III.4.3.3. Conception logicielle en environnement Arduino

La partie logicielle est la partie la plus complexe de ce projet. Le programme a apparemment été écrit en langage Arduino dans la version 1.8.9 de l'environnement de développement. Voir "l'annexe" pour plus de détails sur l'environnement de développement et la structure du programme Arduino. Comme tout programme Arduino, notre programme se décompose en trois (03) parties :

✓ Déclaration des variables, des constantes, l'inclusion des bibliothèques

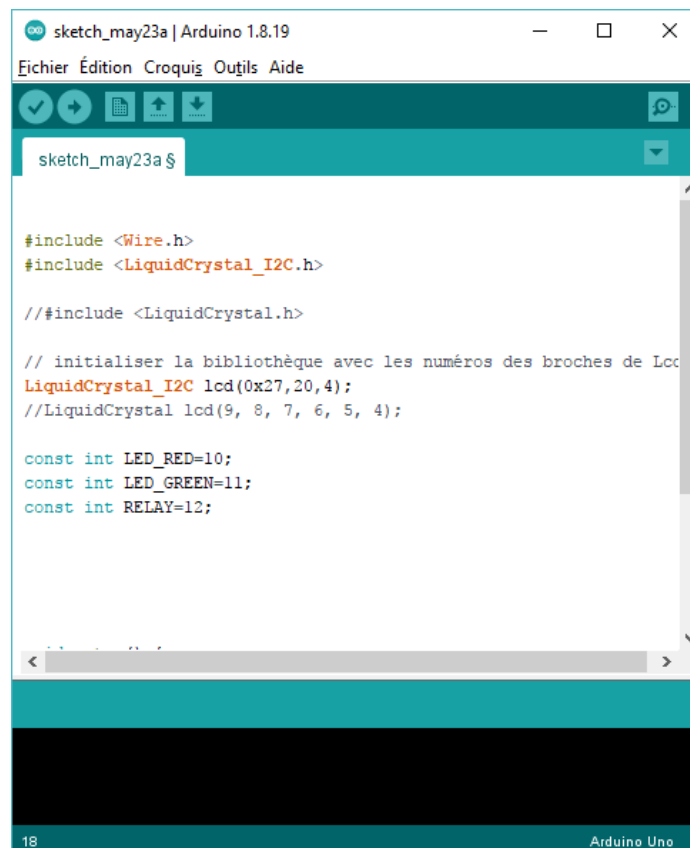


```
controlleur | Arduino 1.8.19
Fichier Édition Croquis Outils Aide

controlleur
// initialiser la bibliothèque avec les numéros des broches de Lcd
LiquidCrystal_I2C lcd(0x27,20,4);
//LiquidCrystal lcd(9, 8, 7, 6, 5, 4);

const int LED_RED=10;
const int LED_GREEN=11;
const int RELAY=12;
```

Figure III.5 : Partie déclaration des variables dans le programme.



```
sketch_may23a | Arduino 1.8.19
Fichier Édition Croquis Outils Aide

sketch_may23a $

#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//#include <LiquidCrystal.h>

// initialiser la bibliothèque avec les numéros des broches de Lcd
LiquidCrystal_I2C lcd(0x27,20,4);
//LiquidCrystal lcd(9, 8, 7, 6, 5, 4);

const int LED_RED=10;
const int LED_GREEN=11;
const int RELAY=12;
```

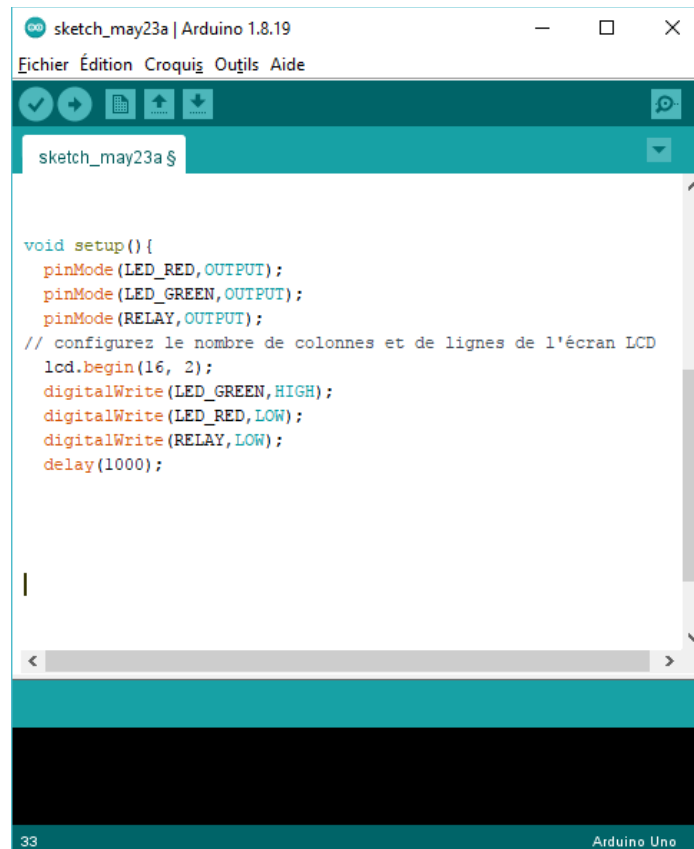
Figure III.6 : Partie déclaration dans le programme.

Comme vous pouvez le voir sur figure (III.6), le programme utilise beaucoup de variables et 2 bibliothèques, à savoir :

```
#include <Wire.h>
```

```
#include <LiquidCrystal_I2C.h>
```

✓ La fonction d'initialisation « **setup ( )** » :

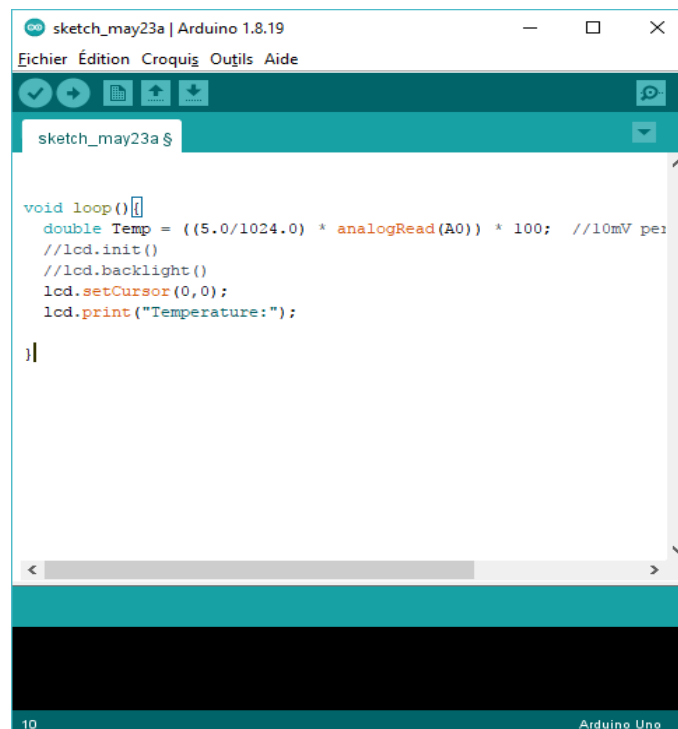
The image shows a screenshot of the Arduino IDE interface. The window title is "sketch\_may23a | Arduino 1.8.19". The menu bar includes "Fichier", "Édition", "Croquis", "Outils", and "Aide". The toolbar contains icons for saving, undo, redo, and other functions. The main editor area shows the following code:

```
void setup() {  
  pinMode(LED_RED, OUTPUT);  
  pinMode(LED_GREEN, OUTPUT);  
  pinMode(RELAY, OUTPUT);  
  // configurez le nombre de colonnes et de lignes de l'écran LCD  
  lcd.begin(16, 2);  
  digitalWrite(LED_GREEN, HIGH);  
  digitalWrite(LED_RED, LOW);  
  digitalWrite(RELAY, LOW);  
  delay(1000);  
}
```

The status bar at the bottom indicates "33" and "Arduino Uno".

**Figure III.7** : Fonction « **setup ( )** » du programme

✓ La fonction principale « `loop ( )` » :



```
sketch_may23a | Arduino 1.8.19
Fichier Édition Croquis Outils Aide

sketch_may23a $

void loop() {
  double Temp = ((5.0/1024.0) * analogRead(A0)) * 100; //10mV per
  //lcd.init()
  //lcd.backlight()
  lcd.setCursor(0,0);
  lcd.print("Temperature:");
}

10 Arduino Uno
```

**Figure III.8 :** Fonction « `loop ( )` » du programme

Après avoir édité le code, il ne reste plus qu'à compiler et le téléverser sur la carte.

### III.4.4 : Fonctionnalités du programme :

Lorsque nous développons des systèmes critiques, par exemple : la chaleur ou le feu, le seul aspect important que nous devons garder à l'esprit est le scénario du monde réel. "Incendie ou Surchauffe" pouvant survenir à tout moment. Cela signifie que notre système doit surveiller en permanence la chaleur 24h/7jours.

Nous verrons dans notre programme qu'il ne contient que deux appels de fonction dans une boucle void, l'une de ces fonctions utilisée pour calculer et afficher la température détectée par le capteur et l'autre pour comparer la température actuelle avec le point de consigne. C'est la fonction qui surveille la montée de température 24h/7jours. Cette fonction renvoie la température mesurée par le LM35 et les stocke dans la variable pour comparaison. Cette valeur de température est comparée à des valeurs spécifiques (point de consigne). La température est généralement comprise entre -55 °C et 155 °C. Cela variera selon les continents et les lieux.



Donc pour modifier la valeur de point consigne de température on a deux interrupteurs tactiles une pour augmenter le point de consigne de température fixe jusqu'à 155 °C, et la deuxième pour réduire le point de consigne de température fixe jusqu'à -55 °C.

Une fois la valeur de température actuelle augmente au de la valeur de consigne qu'elle est fixée, les éléments de sécurités activent : le relai automatiquement vas couper le passage de courant pour éteindre donc l'élément chauffant tel que le chauffage qui est connecté à la sortie relais. Si la température actuelle est inférieure à le point de consigne de température, le relais reste activé (chauffage). Nous affichons également l'état du chauffage sur les LEDs et sur l'écran LCD.

a. Organigramme du programme initial :

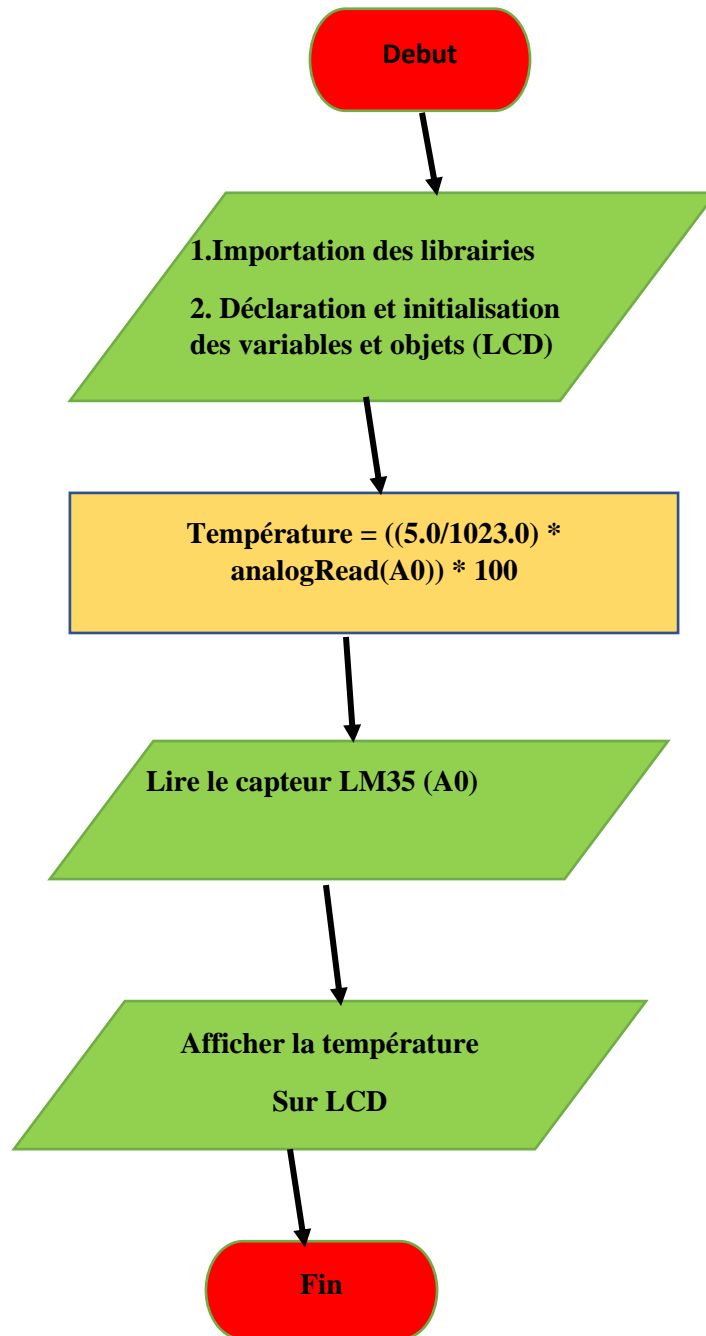


Figure III.9 : Organigramme du programme initial

b. Organigramme du programme principal

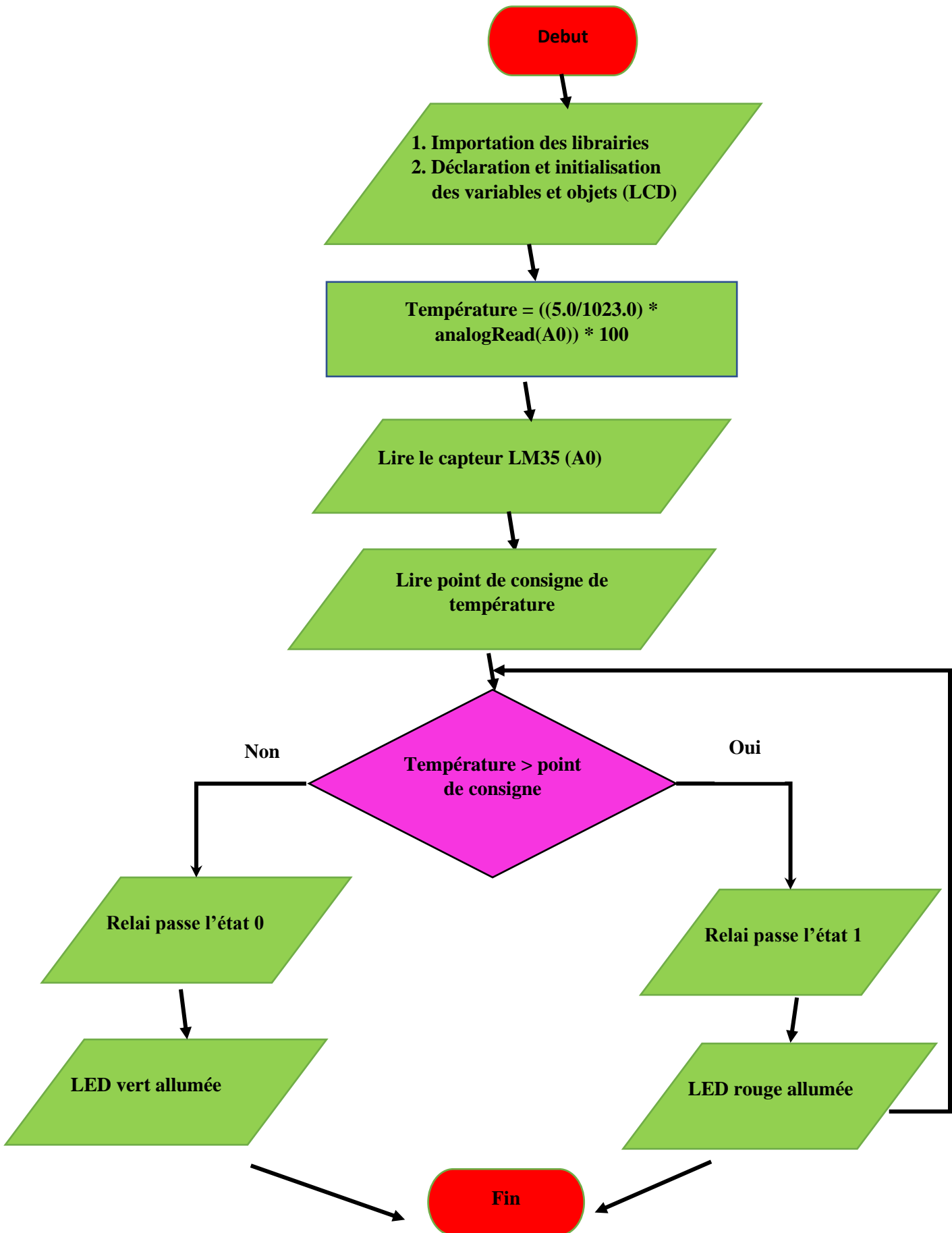


Figure III.10 : Organigramme du programme principal

### III.4.5 Aspects de la simulation

#### III.4.5.1 Bibliothèque Arduino pour Proteus

Il existe de nombreuses bibliothèques qui vous permettent d'étendre les fonctionnalités du langage Arduino. Ils facilitent la programmation des capteurs, des pré-actionneurs, des modules de communication et d'affichage...

Pour simuler une carte Arduino sur Proteus, il faut d'abord télécharger la bibliothèque Arduino disponible sur internet, puis il faut décompresser et copier les fichiers nommés "ArduinoTEP.LIB et Arduino.IDX" et les placer dans le dossier bibliothèque de notre logiciel Proteus. Enfin, nous devons redémarrer le logiciel Proteus et sélectionner l'Arduino Uno dans la section des composants qui recherche Arduino TEP comme suit :

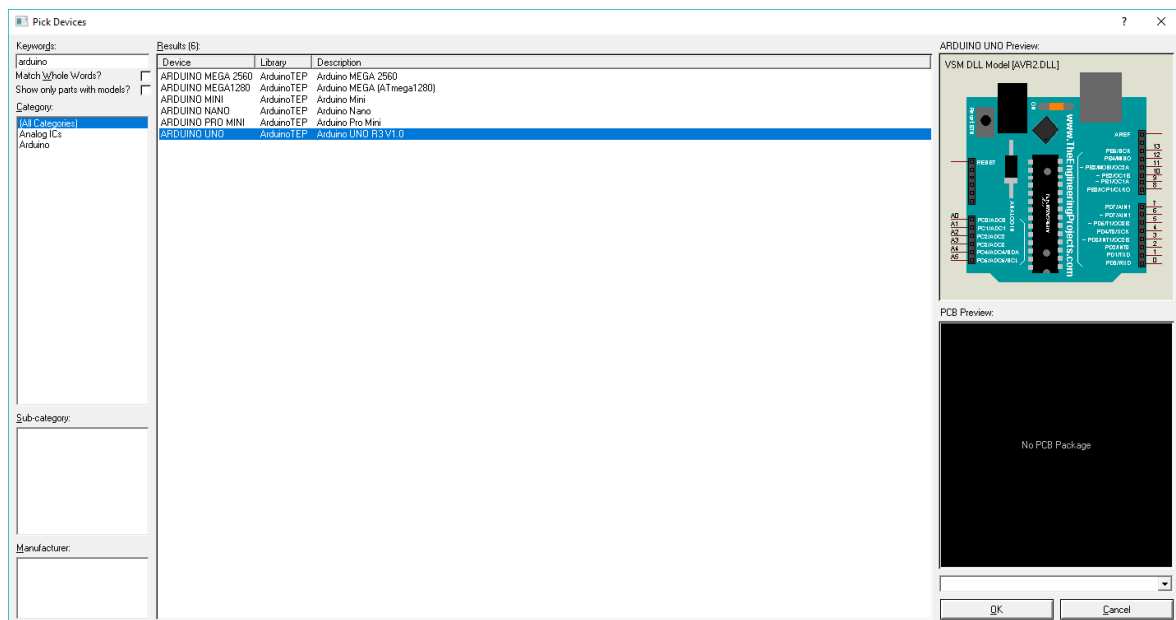


Figure III. 11 : Intégrer carte Arduino sous Proteus.

Avant de lancer la simulation, nous devons charger le fichier .Hex. Pour continuer, double-clique sur PIC, pour nous donner une fenêtre qui nous permet de rechercher des fichiers .Hex, dans cette fenêtre On peut aussi choisir le type d'Arduino UNO. Cette fenêtre correspond à ceci.

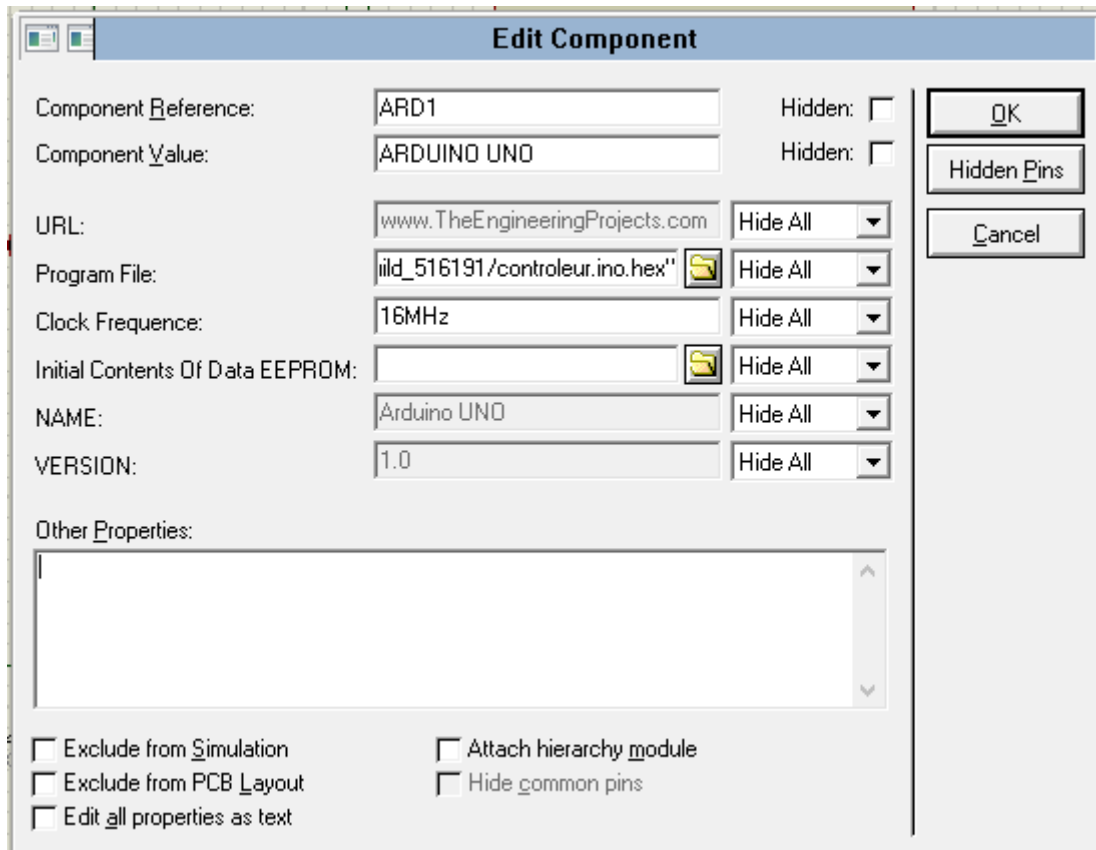
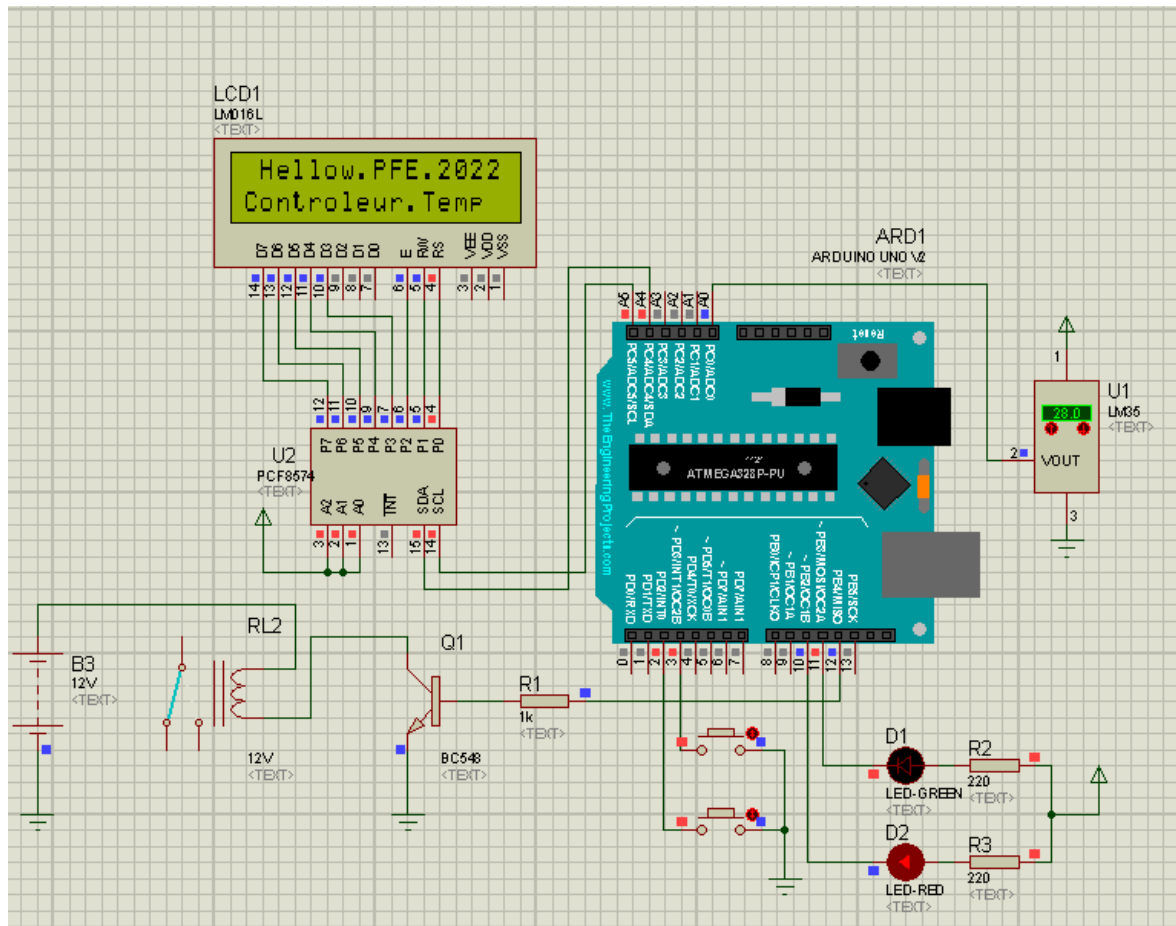


Figure III.12 : Charge carte Arduino sous Proteus.

### III.5. Résultats de la simulation :

#### III.5.1. L'état initial du système :

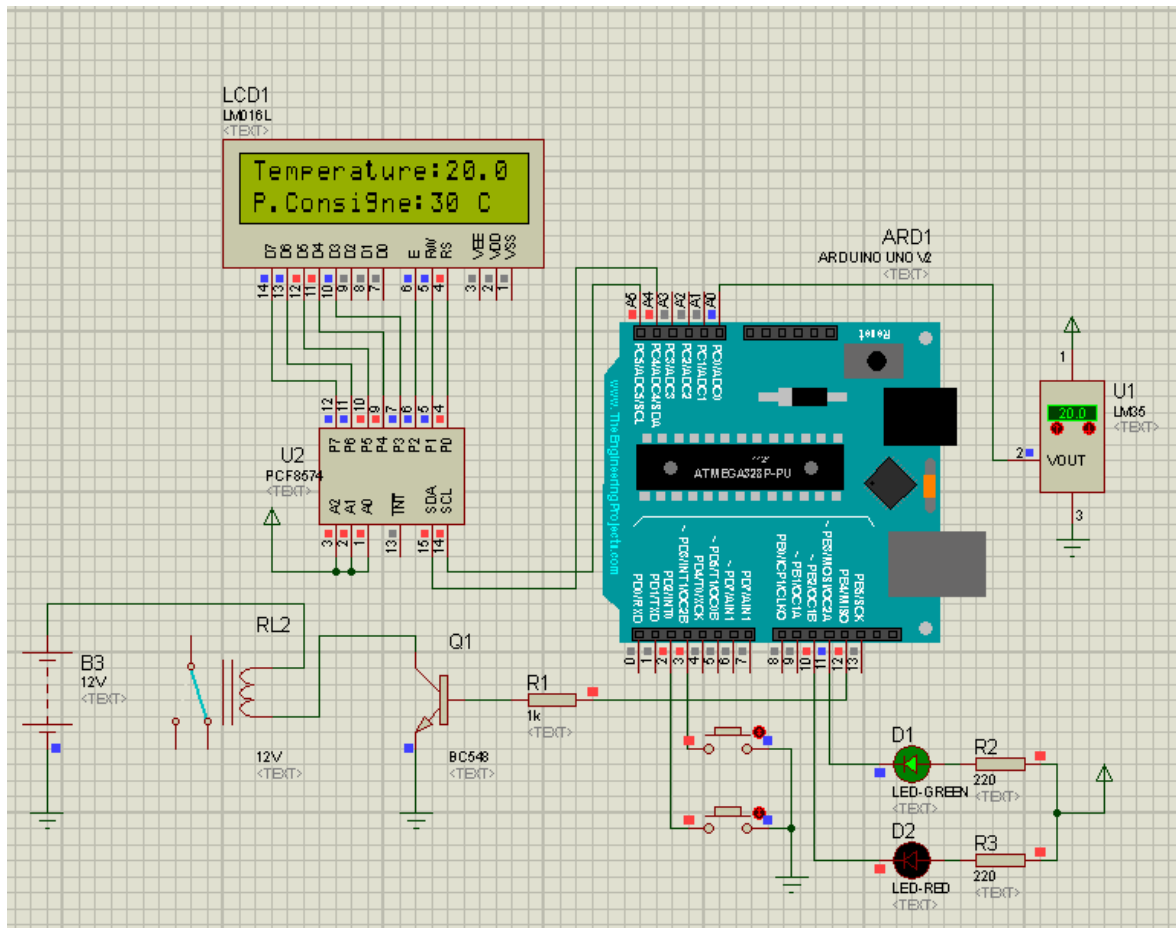
Lors du lancement de notre prototype, l'écran LCD affichera un message de bienvenue. (HELLOW PFE... !...etc.).



**Figure III. 13 :** L'état initial du système

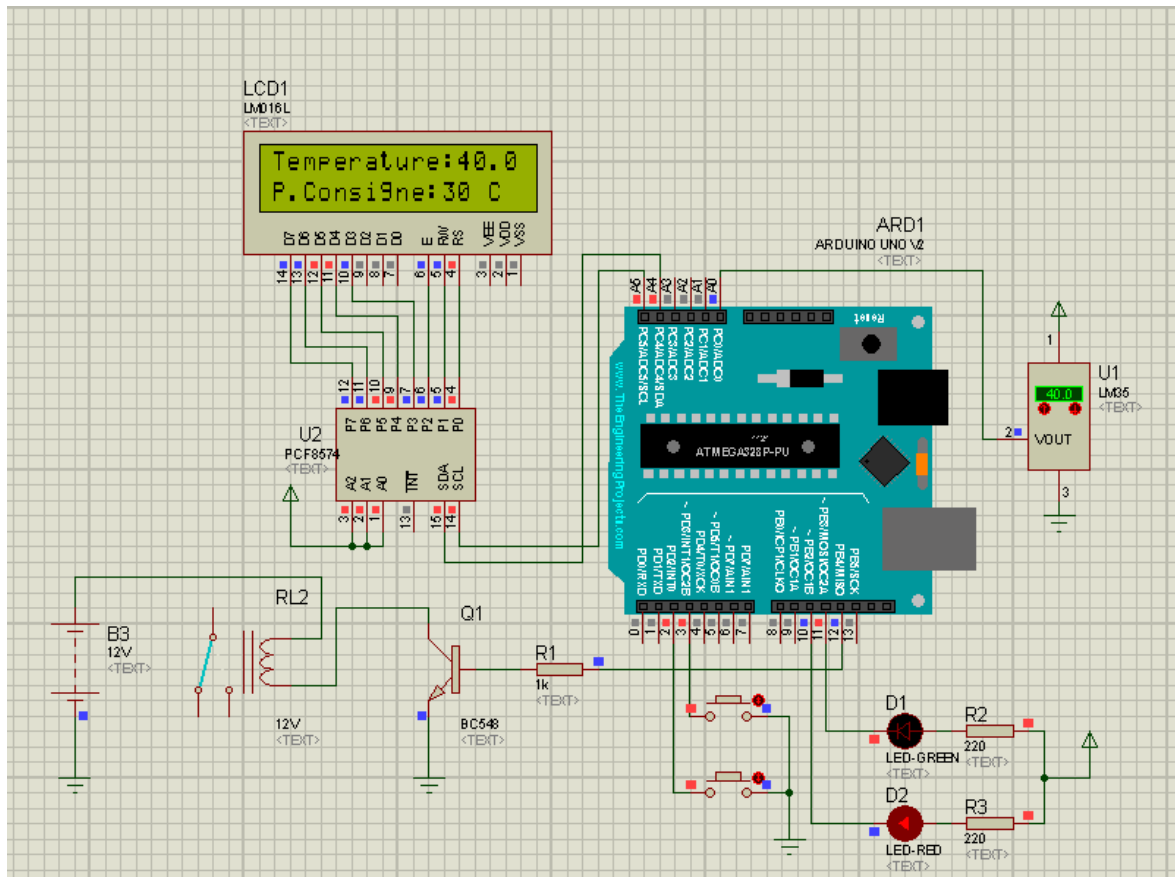
Après quelques secondes, l'écran affiche l'état du système et les informations sur la température.

Dans le cas où la température inférieure à la valeur de point de consigne de température, la LED verte s'allume (passante). Ainsi, la sortie numérique de la carte Arduino (broche n 12) se trouve à l'état bas (niveau logique 0) et le relais est à l'état ON (normal open).



**Figure III. 14 :** Montage électrique de contrôleur de température numérique :  
Températures actuelle inférieure à le point de consigne

Dans le cas où la température est supérieure à la valeur de point de consigne, le circuit d'alarme est activé : la LED verte est éteinte, la LED rouge est allumée, Ainsi, la sortie numérique de la carte Arduino (broche n 12) se trouve à l'état haut (niveau logique 1) le relais est déclenché et passe à l'état off (normal close).



**Figure III. 15 :** Montage électrique de contrôleur de température numérique :  
Températures actuelle supérieure à le point de consigne

### III.6 Logiciel ARES de conception de circuit imprimé [42]

Cette partie du logiciel ARES nous donne un accès facile à la conception des circuits imprimés, y compris le placement des composants et le routage 16 couches, que ce soit en mode manuel, semi-automatique, ou automatique.

Les algorithmes implémentés dans le logiciel permettent l'utilisation d'autres outils dans la dernière version, comme la réalisation de visualisations 3D de cartes.

La figure III.16 montre l'assemblage général d'une dimension du contrôleur de température, ainsi qu'une représentation du circuit.

La figure III.17 représente le dessin du montage de notre appareil de mesure sur ce logiciel, ainsi que la représentation du montage du circuit en une dimension.



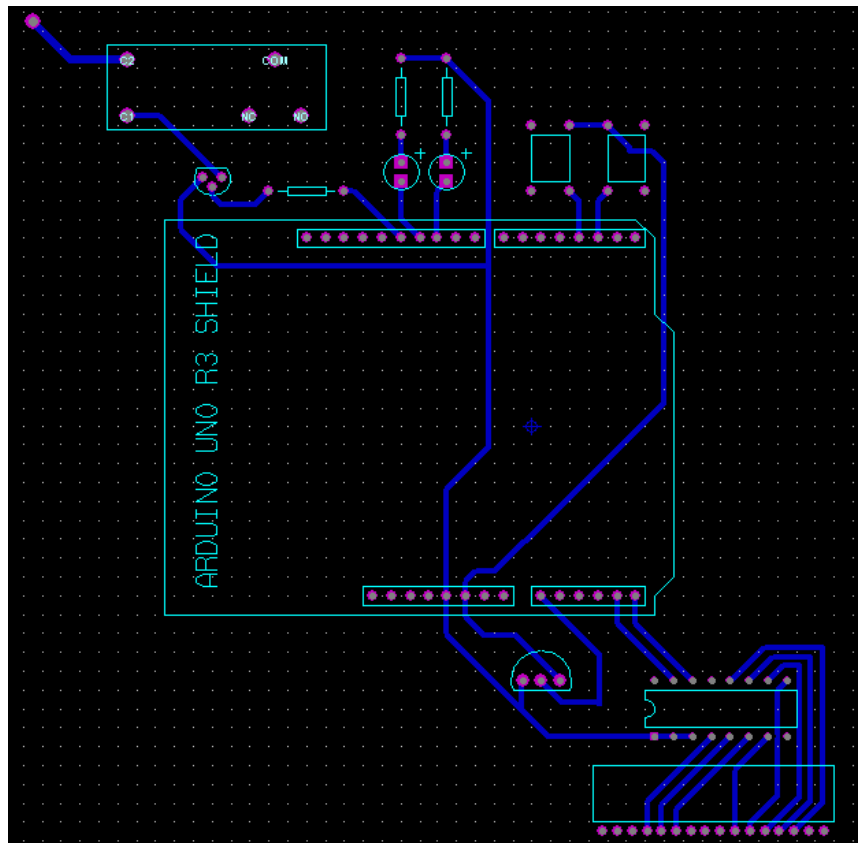


Figure III.16 : Montage de contrôleur de température numérique à base de la carte Arduino sur l'ARES

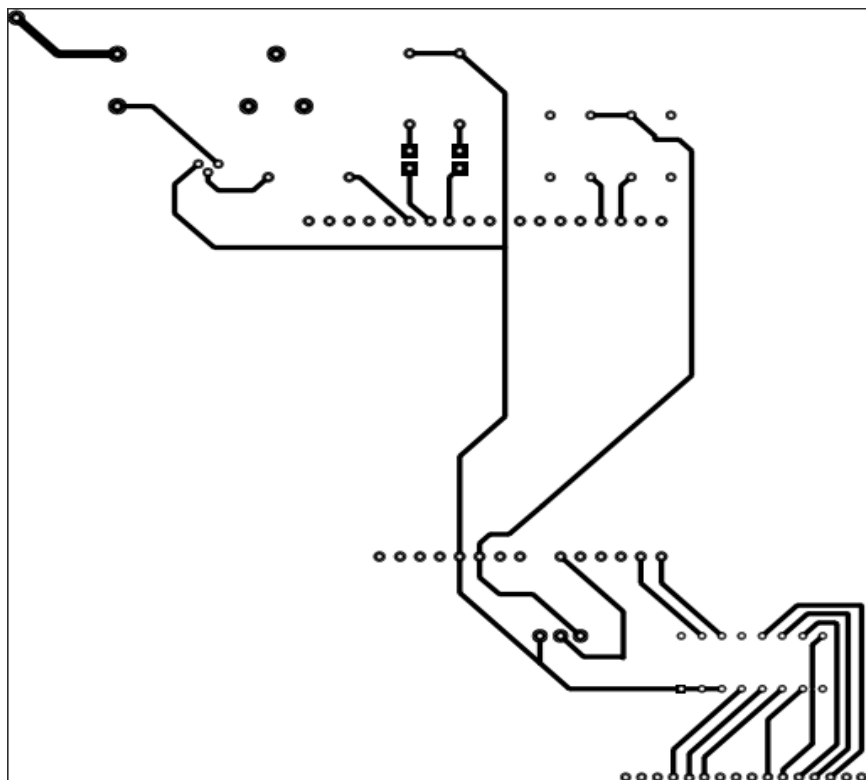


Figure III.17 : Dessin de montage de contrôleur de température numérique à base de la carte Arduino

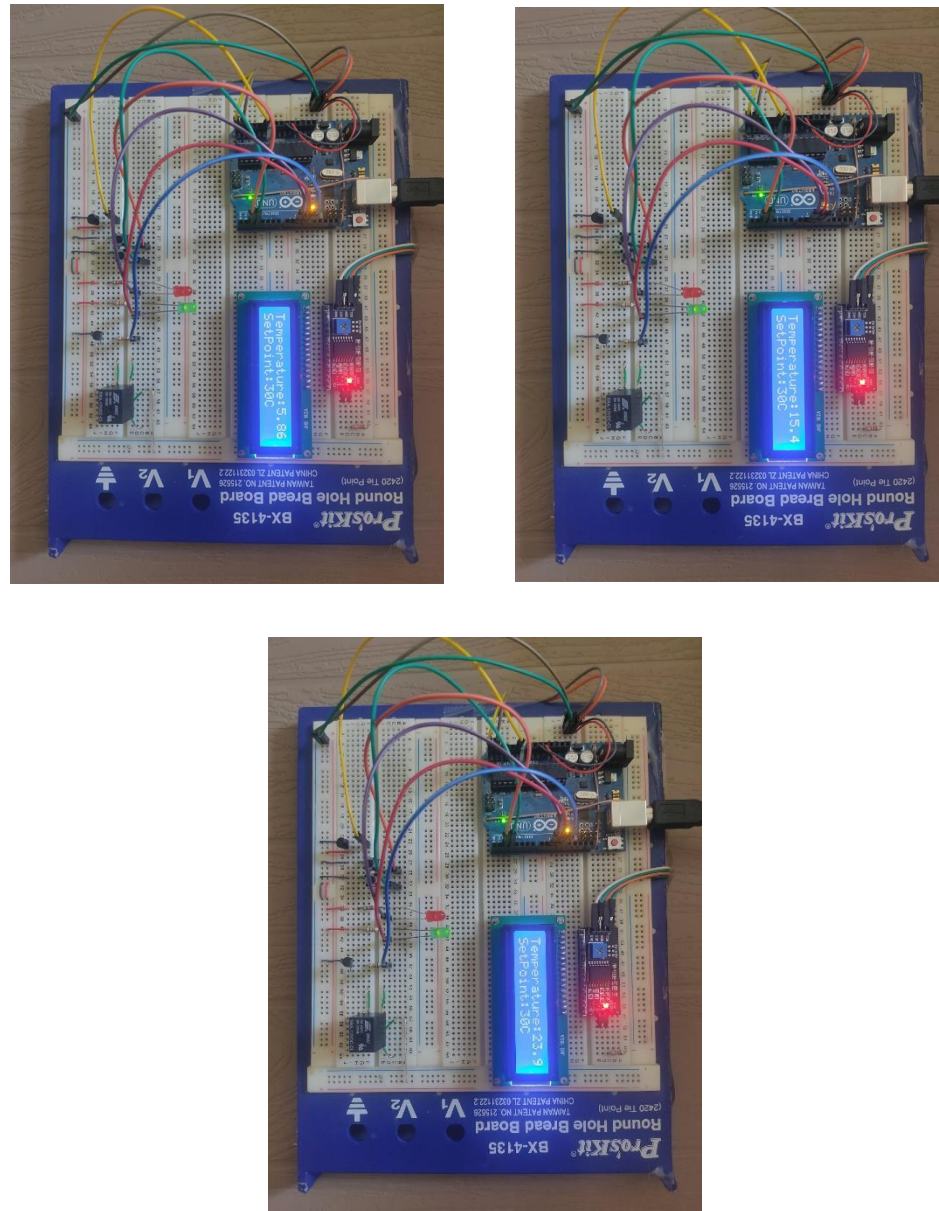
### III.7 Liste des composants utilisés

- ◆ Arduino UNO
- ◆ Capteur de température LM35
- ◆ Deux Interrupteurs tactiles
- ◆ Résistances,
- ◆ LEDs
- ◆ Transistor BC548
- ◆ Afficheur LCD (16x2)
- ◆ Carte I2C
- ◆ Relai

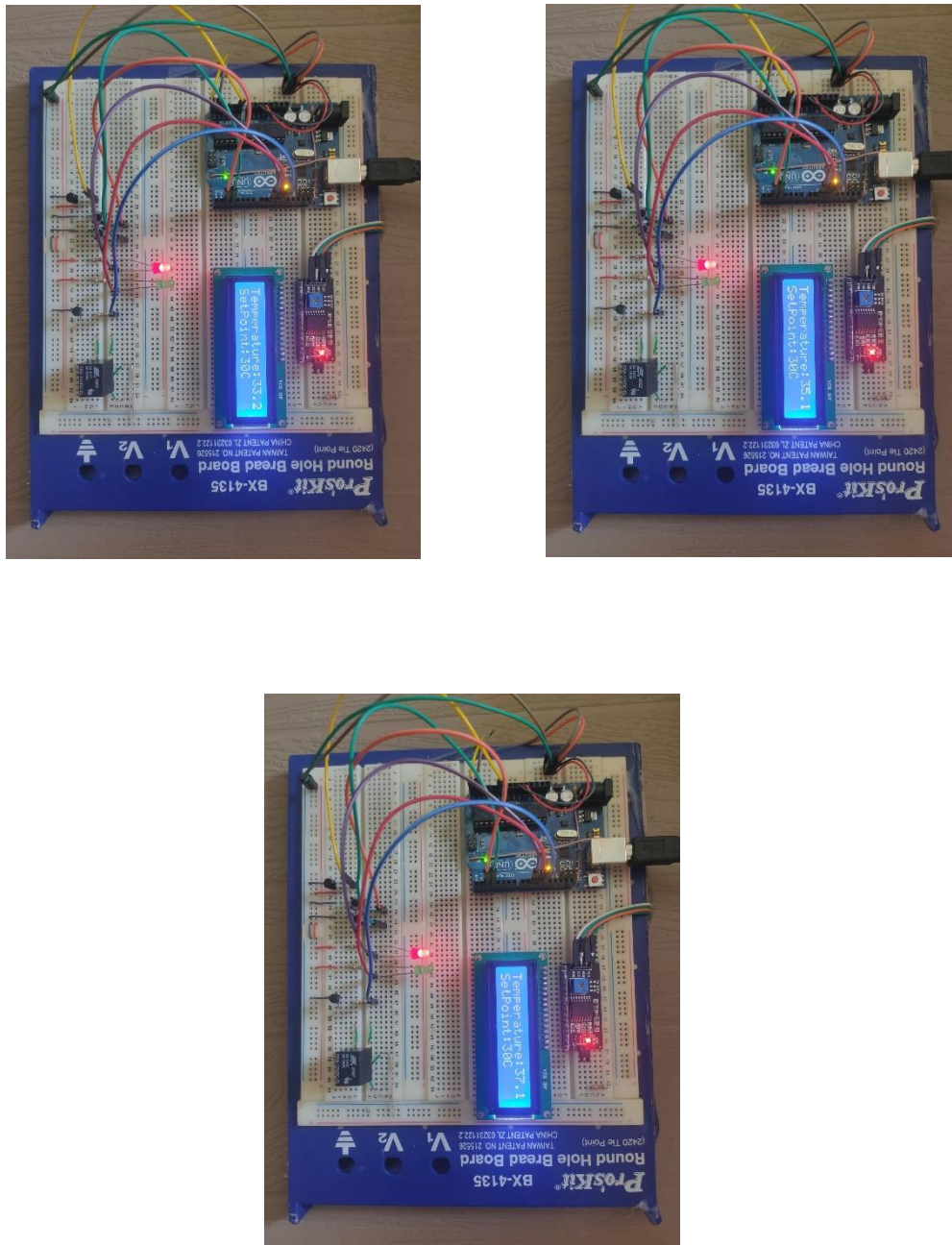
### III.8 Réalisation le thermomètre numérique sur plaque d'essais

La stratégie dans la partie réalisation est basée sur la simplicité des montages et le moindre coût, afin d'assurer le bon fonctionnement de notre carte dont réduire le nombre de connexions.

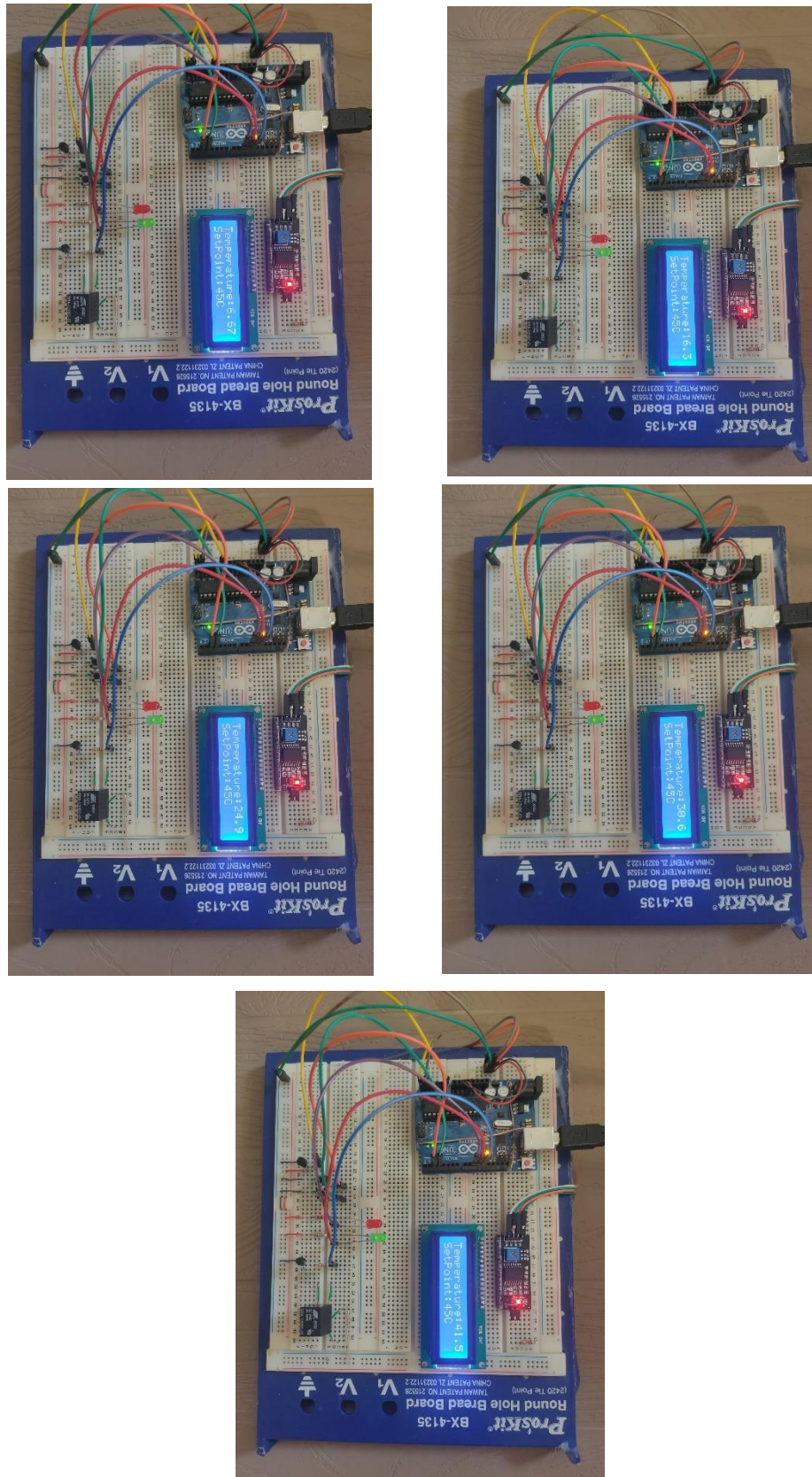
La figure III.15 illustre la réalisation de montage électrique de contrôleur de température numérique sur la plaque d'essai.



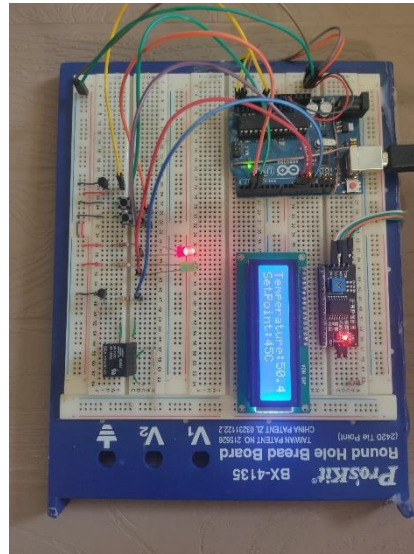
**Figure III.18:** Photos de contrôleur de température numérique réalisé en état marche  
(Température actuelle < Point de consigne de température (30 °C))



**Figure III.19:** Contrôleur de température numérique réalisé en état marche  
(Température actuelle > Point de consigne de température (30 °C))



**Figure III.20:** Contrôleur de température numérique réalisé en état marche (Température actuelle < Point de consigne de température (45 °C))



**Figure III.21:** Photo de contrôleur de température numérique réalisé en état marche  
(Température actuelle > Point de consigne de température (45 °C))

### III.9. Quelques mesures de températures effectuées avec le dispositif numérique réalisé :

Point de consigne	Température affichée avec différents états							
	Température actuelle < Point de consigne				Température actuelle > Point de consigne			
30 °C	5.86	15.4	23.9	29.3	35.1	37.1	40.5	42.4
	Température actuelle < Point de consigne				Température actuelle > Point de consigne			
45 °C	6.67	16.3	23	24.9	30.6	41.5	50.1	
	Température actuelle < Point de consigne				Température actuelle > Point de consigne			

**Tableau III.1 :** Résultats effectués de contrôleur de température réalisé

### III.10 Conclusion

Cette partie donne sur la mise en œuvre pratique de systèmes électroniques simples rappeler aux responsables d'une usine, d'une maison..., en autorisant les affichages La température atteinte sur l'afficheur, et en cas de dépassement du seuil de température programmé, une alarme va se déclenchée via les LEDs et le relais.

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 dans des conditions limitées du matériel et des moyens dont nous disposons.

# Conclusion générale



### Conclusion générale

Dans ce travail, nous avons détaillé l'élaboration d'un dispositif numérique de contrôleur de température basé sur Arduino UNO. Notre thermomètre utilise le capteur intégré LM35 pour mesurer la température. Les calculs et conversions nécessaires sont délégués au microcontrôleur et affichés sur un écran LCD, et les données sont traitées par le logiciel ISIS. En effet notre application permet de mesurer et de surveiller les variations de températures dans un lieu.

Nous utilisons le logiciel ISIS pour démarrer nos simulations. Cela nous amène à introduire des éléments dans le schéma qui permettent de considérer :

- La carte Arduino UNO.
- La carte I2C
- Le capteur de température LM35 qui mesure la température avec un signal électrique (tension) proportionnel à la température en °C
- Un afficheur LCD qui permet l'affichage des variations de température en °C.
- Relai

La conception de ce travail nous donne accès à plusieurs domaines, comme l'utilisation d'Arduino nous introduit au domaine  $\mu C$ , leur implémentation et leurs systèmes de développement et les différentes fonctionnalités qu'ils offrent.

En effet, les résultats obtenus montrent une bonne efficacité du système dans la détection et l'alerte en temps réel de la température.

---

## References

- [1] <https://blog.beamex.com/thermocouple-cold-junction-compensation>
- [2] L. Guenfoud, C. Mouzarine, « conception et réalisation d'un système du contrôle à distance de la qualité de l'air ». Mémoire de fin d'étude, Université Mouloud Mammeri de Tizi-Ouzou, 2018
- [3] <http://silanus.fr/bts/activites/Maugenet/co/structure d'un capteur.html>
- [4] C. Marlot. « Elaboration de céramiques transparentes Er YAG : synthèse de poudre par Co-précipitation et frittage SPS », Thèse soutenue le 12 mars (2013). Université de Bourgogne
- [5] Christophe Pijolat, « étude des capteurs microélectronique : actif et passif » ; Ecole nationale supérieure des mines de Saint-Etienne, 2004
- [6] [http://gatt.fr/CIRA/Cours/Instrum/CIRA1%20-%20202\)%20Capteurs.pdf](http://gatt.fr/CIRA/Cours/Instrum/CIRA1%20-%20202)%20Capteurs.pdf)
- [7] <https://www.omega.fr/prodinfo/sondes-temperature.html>
- [8] Djebli Assia, Bezza Fatna « Conception, simulation et réalisation d'un détecteur de pulsation cardiaque couplé à un thermomètre électronique », Mémoire de fin d'études, Université de Saida, Moulay Tahar, Soutenu en Juin 2018.
- [9] AMMARKHODJA Nassim, « Etude et réalisation d'une alarme de température à base d'une carte arduino » Mémoire de fin d'étude, Université Mouloud Mammeri de Tizi-Ouzou, 2018.
- [10] Ikhlas. BRADAI, Ghezal. Roumaïssa, « Etude, Simulation d'un Thermomètre Numérique à base d'Arduino ». Mémoire PFE, Université Ain Temouchent, 2021.
- [11] Georges Asch et coll, « Les capteurs en instrumentation industrielle » Livre à l'édition Dunod.
- [12] Alain Charbonneau « Les signaux électriques » Publication le 08/11/2016
- [13] <http://www.bedwani.ch/electro/ch25/index.htm>
- [14] <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- [15] <https://www.digikey.fr/fr/product-highlight/t/texas-instruments/lm335-temperature-sensor>.

- [16] Yacine Amkassou, Omar Elbarkani, Sliman Elnairy, « mini projet : Thermomètre a base du Arduino UNO », Ecole nationale des sciences appliquées de khouribga, 2012/2013.
- [17] <https://arduino.developpez.com/tutoriels/cours-complet-arduino/?page=vous-avez-dit-arduino>.
- [18] M. Zerrouki, R. Nesnas, « Conception et réalisation d'un système de commande d'une habitation ». Mémoire de fin d'études, Université Mouloud Mammeri « TIZI-OUZOU », 2017/2018.
- [19] [https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:presentation\\_arduino.pdf](https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:presentation_arduino.pdf)
- [20] <https://wiki.mdl29.net/lib/exe/fetch.php?media=elec:arduino-pour-bien-commencer-en-electronique-et-en-programmation.pdf>
- [21] A. Aghiles, H. Kaci- Moussa, « Conception et réalisation d'un système de régulation à base d'un microcontrôleur », Mémoire de fin d'études, Université Mouloud Mammeri « Tizi-Ouzou », 12 juillet 2018.
- [22] <https://maker.pro/arduino/tutorial/a-comparison-of-popular-arduino-boards>
- [23] H. KACI MOUSSA, « Conception et réalisation d'un système de régulation à base d'un microcontrôleur », Mémoire de fin d'études, Université Mouloud Mammeri De « Tizi-Ouzou », 12 juillet 2018.
- [24] I. Bradai, G. Roumaïssa, « Etude, Simulation d'un Thermomètre Numérique à base d'Arduino ». Mémoire PFE, Université Ain Temouchent, 202.1
- [25] <https://www.gotronic.fr/art-carte-arduino-uno-12420.htm>
- [26] <https://bentek.fr/2-arduino-uno>
- [27] M. Inguel, « Conception et réalisation d'un système Domotique par GSM ». Mémoire PFE, Université Mouloud Mammeri De Tizi-Ouzou, 2017.
- [28] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif
- [29] A. Krama, A. Gougui, « Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde ». Mémoire PFE, Université KASDI MERBAH OUARGLA, 2015.
- [30] S. Ben-Lounes, D. Ahmoune, « Conception et réalisation d'un système de vidéo-surveillance à base de la carte Arduino UNO>> ». Mémoire fin d'étude à L'UMMTO, BENNAMENE.K, 2014.

- [31] N. Ammarkhoudja, « Etude et réalisation d'une alarme de température à base d'une carte Arduino », Mémoire de fin d'études, Université Mouloud Mammeri « TIZI-OUZOU », 2017/2018.
- [32] <http://codelab.fr/3842#:~:text=A%20ce%20stade%2C%20le%20capteur,le%20patc h%20de%20Pure%20Data>).
- [33] <https://www.aurel32.net/elec/lcd.php>
- [34] Boudjedir Imen. « Un système embarqué pour la détection des gaz dangereux ». Université LARBI BEN M'HIDI D'OUOM EL BOUAGUI, Faculté des sciences et sciences appliquées, 06/2017
- [35] <https://arduino-france.site/i2c-scanner/>  
<https://arduino-france.site/lcd-1602/>
- [36] Ayad Benyounes, Tari Oussama « Simulation et réalisation d'un circuit détecteur de gaz interfacé à la carte Arduino ». Mémoire PFE, Université Tlemcen, 2020
- [37] BC548 - Wikipedia
- [38] <https://www.aranacorp.com/fr/utilisation-dun-module-transistor-avec-arduino>
- [39] <http://www.zpag.net/Electroniques/relais.htm>.
- [40] Proteus (ISIS et ARES) - Logiciel Électronique
- [41] <http://www.elektronique.fr/logiciels/proteus.php>
- [42] M. Lakhdari. « Introduction à la simulation et routage des circuits avec le logiciel PROTEUS V7 et V8 » Polycopie des travaux, USTO-MB 2016/2017

## LM35 Precision Centigrade Temperature Sensors

### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 1/4^\circ\text{C}$  at room temperature and  $\pm 1/2^\circ\text{C}$  over a full  $-55$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only  $60\ \mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40$  to  $+150^\circ\text{C}$  range ( $-10$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear  $+10.0\ \text{mV}/^\circ\text{C}$  scale factor
- $0.5^\circ\text{C}$  accuracy guaranteeable (at  $+25^\circ\text{C}$ )
- Rated for full  $-55$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than  $60\ \mu\text{A}$  current drain
- Low self-heating,  $0.06^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 1/4^\circ\text{C}$  typical
- Low impedance output,  $0.1\ \Omega$  for  $1\ \text{mA}$  load

### Typical Applications

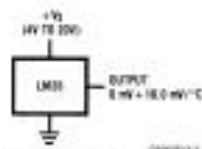
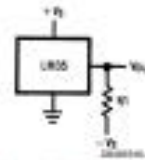


FIGURE 1. Basic Centigrade Temperature Sensor ( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )



Choose  $R_1 = -10/90\ \mu\text{A}$   
 $V_{\text{OUT}} = +1.000\ \text{mV}$  at  $+100^\circ\text{C}$   
 $= +200\ \text{mV}$  at  $+25^\circ\text{C}$   
 $= -800\ \text{mV}$  at  $-55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

## La fiche technique de LM35 (1)

LM35

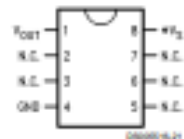
Connection Diagrams

**TO-46  
Metal Can Package\***



\*Case is connected to negative pin ( $V_{CC}$ )  
**Order Number LM35H, LM35AH, LM35CH, LM35CAH or LM35DH**  
 See NS Package Number H63H

**SO-8  
Small Outline Molded Package**



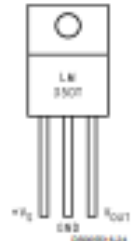
N.C. = No Connection  
**Top View**  
**Order Number LM35DM**  
 See NS Package Number M66A

**TO-92  
Plastic Package**



**Order Number LM35CZ, LM35CAZ or LM35DZ**  
 See NS Package Number Z33A

**TO-220  
Plastic Package\***



\*Tab is connected to the negative pin ( $V_{CC}$ )  
 Note: The LM35DT pinout is different than the discontinued LM35DP.  
**Order Number LM35DT**  
 See NS Package Number TA03F

La fiche technique de LM35 (2)



**Typical Applications**

**FIGURE 3. LM35 with Decoupling from Capacitive Load**

**FIGURE 4. LM35 with R-C Damper**

**CAPACITIVE LOADS**  
 Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easy to isolate or decouple the load with a resistor; see Figure 3. Or you can improve the tolerance of capacitance with a series R-C damper from output to ground; see Figure 4.

When the LM35 is applied with a 200Ω load resistor as shown in Figure 5, Figure 6 or Figure 8 it is relatively immune to wiring capacitance because the capacitance forms a bypass from ground to input, not on the output. However, as with any linear circuit connected to wires in a hostile environment, its performance can be affected adversely by intense electromagnetic sources such as relays, radio transmitters, motors with arcing brushes, SCR transients, etc. as its wiring can act as a receiving antenna and its internal junctions can act as rectifiers. For best results in such cases, a bypass capacitor from  $V_{CC}$  to ground and a series R-C damper such as 75Ω in series with 0.2 or 1 µF from output to ground are often useful. These are shown in Figure 7, Figure 14, and Figure 16.

**FIGURE 5. Two-Wire Remote Temperature Sensor (Grounded Sensor)**

**FIGURE 6. Two-Wire Remote Temperature Sensor (Output Referred to Ground)**

**FIGURE 7. Temperature Sensor, Single Supply, -55°C to +150°C**

**FIGURE 8. Two-Wire Remote Temperature Sensor (Output Referred to Ground)**

**FIGURE 9. 4-To-20 mA Current Source (0°C to +100°C)**

La fiche technique de LM35 (4)



