



وزارة البحث العلمي والتعليم العالي  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE  
LA RECHERCHE SCIENTIFIQUE  
جامعة عبد الحميد بن باديس مستغانم  
Université Abdelhamid Ibn Badis Mostaganem  
كلية العلوم و التكنولوجيا  
Faculté des Sciences et de la Technologie  
DEPARTEMENT DE GENIE DES PROCÉDES



N° d'ordre : M...../GC/2019

## MEMOIRE

Présenté pour obtenir le diplôme de

### MASTER EN GENIE ELECTRIQUE

Option: **Electronique Des Systèmes Embarqués**

Par

**Bouich Houria**

**Chibane Fadhila**

**Etude et Simulation D'une Carte d'Acquisition de température et  
d'Humidité à base d'Arduino UNO**

Soutenu le 2020 devant le jury composé de :

|                     |               |     |                          |
|---------------------|---------------|-----|--------------------------|
| <b>Président :</b>  | Mme F.Bechiri | MCB | Université de Mostaganem |
| <b>Examineur :</b>  | M W.Benstaali | MCA | Université de Mostaganem |
| <b>Rapporteur :</b> | Mme A.Abbad   | MCA | Université de Mostaganem |

Année Universitaire 2019/2020

## REMERCIEMENTS

NOUS REMERCIONS DIEU LE TOUT PUISSANT DE NOUS  
AVOIR DONNÉ LES APTITUDES POUR MENER À BIEN CE  
MODESTE TRAVAIL.

UN GRAND MERCI POUR NOTRE PROMOTEUR MME  
A.ABBAD POUR SA DISPONIBILITÉ, SON SUIVI ET SES  
CONSEILS TOUT AU LONG DE NOTRE TRAVAIL.

NOUS TENONS ÉGALEMENT À REMERCIER LES MEMBRES  
DU JURY QUI NOUS ONT FAIT L'HONNEUR DE JUGER  
NOTRE TRAVAIL.

NOUS REMERCIONS AUSSI TOUS LES ENSEIGNANTS DU  
DÉPARTEMENT DU GÉNIE ÉLECTRIQUE.

POUR FINIR NOUS REMERCIONS NOS PARENTS ET NOS  
AMIS QUI NOUS ONT SOUTENUS DURANT TOUS NOTRE  
CURSUS UNIVERSITAIRE.

## *Dédicaces*

*Nous dédions ce travail à :*

☐☐ *Nos chers mères et nos chers pères*

☐☐ *Nos Chers frères et sœurs*

☐☐ *Toute la famille*

☐☐ *Tous les amis*

☐☐ *Tous les collèges de la promotion 2020*

☐☐ *Tous ceux que nous aimons et tous*

*ceux qui nous aiment*

☐☐ *Tous ceux qui connaissent :*

**BOUICH HOURIA**

**CHIBANE FADHILA**

## Résumé du projet

Notre projet consiste à contrôler et mesurer la température et l'humidité dans la situation locale et à distance, en utilisant un microcontrôleur ArduinoUno, un module GSM et un capteur de température et d'humidité. En utilisant le logiciel Arduino, nous avons simulé le fonctionnement d'une carte d'acquisition et de contrôle en utilisant un module GSM et un téléphone portable basé sur Arduino, ce qui permet la surveillance du système à distance. Notre carte fonctionne de telle façon quand le capteur capte une variation de la température ou de l'humidité d'un local quelconque, le système commence à envoyer des SMS ou à appeler l'utilisateur pour l'avertir de ce problème. En outre, on peut éventuellement bloquer le système de fonctionner à distance, ou même il est possible de modifier les utilisateurs en changeant les numéros de téléphone.

**Mots clés:** Module GSM, ARDUINO UNO. Afficheur LCD, Téléphone mobile, Capteur de température et d'humidité (DHT).

## ABSTRACT

Our project consists of controlling and measuring the temperature and humidity in A local and remote situation, using an ArduinoUno microcontroller, a GSM module and a temperature and humidity sensor. Using Arduino software, we simulated the operation of an acquisition and control board using a GSM module and an Arduino-based cell phone, which allows for remote system monitoring. Our card works in such a way when the sensor detects a variation in temperature or humidity in any room, the system starts to send SMS or call the user to warn him of this problem. Moreover, one can optionally block the system from operating remotely, or even it is possible to modify the users by changing the phone numbers.

**Keywords:** GSM module, ARDUINO UNO. LCD display, Mobile phone, Temperature and humidity sensor (DHT).

## ملخص

يمكن مشروعنا في التحكم وقياس درجة الحرارة والرطوبة في الوضع المحلي والبعيد ، باستخدام متحكم ArduinoUno ووحدة GSM وجهاز استشعار درجة الحرارة والرطوبة. باستخدام برنامج Arduino ، قمنا بمحاكاة تشغيل لوحة اقتناء وتحكم باستخدام وحدة GSM وهاتف خلوي قائم على Arduino ، مما يتيح مراقبة النظام عن بُعد. تعمل بطاقتنا بهذه الطريقة عندما يكتشف المستشعر تبايناً في درجة الحرارة أو الرطوبة في أي غرفة ، يبدأ النظام في إرسال الرسائل القصيرة أو الاتصال بالمستخدم لتحذيره من هذه المشكلة. علاوة على ذلك ، يمكن للمرء بشكل اختياري منع النظام من العمل عن بُعد ، أو حتى من الممكن تعديل المستخدمين عن طريق تغيير أرقام الهواتف.

**الكلمات المفتاحية :** وحدة GSM ، ARDUINO UNO. شاشة عرض LCD ، هاتف محمول ، مستشعر درجة الحرارة والرطوبة (DHT).

## Liste des abréviations

|               |   |
|---------------|---|
| <b>GSM</b>    | Global System for Mobile communications             |
| <b>MMS</b>    | Multimedia Message Service                          |
| <b>SMS</b>    | Short Message Service                               |
| <b>BTS</b>    | Base Transceiver Station                            |
| <b>SIM</b>    | Subscriber Identity Module                          |
| <b>IMEI</b>   | International Mobile Equipment Identity             |
| <b>IMES</b>   | International Mobile Equipment Identity             |
| <b>PIN</b>    | Personal Identification Number                      |
| <b>USB</b>    | universal serial bus                                |
| <b>LCD</b>    | Liquid Crystal Display                              |
| <b>EEPROM</b> | Electrically Erasable Programmable Read-Only Memory |
| <b>TTL</b>    | Transistor-Transistor logic                         |
| <b>DHT22</b>  | Digital Humidity-temperature sensor                 |
| <b>CAO</b>    | conception assistée par ordinateur                  |
| <b>R/W</b>    | read/write  |
| <b>RS</b>     | register select                                     |
| <b>UMTS</b>   | Universal Mobile Telecommunications System          |
| <b>RAM</b>    | Random Access Memory                                |
| <b>UART</b>   | Universal Asynchronous Receiver Transmitter         |
| <b>GPRS</b>   | General Packet Radio Service                        |
| <b>BSC</b>    | Base Station Controller                             |
| <b>MSC</b>    | Mobile Service switching Center                     |
| <b>HRL</b>    | Home Location Register                              |
| <b>I2C</b>    | Circuit inter-intégré                               |
| <b>RX</b>     | Recevoir  |
| <b>TX</b>     | Transmettre   |

## Liste des figures et des tableaux

### Liste des figures :

#### CHAPITRE I :

|   |    |
|---|----|
| <b><u>Figure I.1</u></b> : Le schéma bloc du système.....                               | 07 |
| <b><u>Figure I.2</u></b> : Schéma récapitulatif.....                                    | 07 |
| <b><u>Figure I.3</u></b> : Un schéma représentant l'architecture du réseau GSM [8]..... | 10 |
| <b><u>Figure I.4</u></b> : Application d'un module GSM. [9].....                        | 11 |
| <b><u>Figure I.5</u></b> : Module GSM SIM800L.....                                      | 12 |
| <b><u>Figure I.6</u></b> : Brochage du module GSM SIM800L.....                          | 14 |
| <b><u>Figure I.7</u></b> : Schéma de fonctionnement. [10].....                          | 15 |
| <b><u>Figure I.8</u></b> : Schéma descriptif du fonctionnement général du système.....  | 17 |
| <b><u>Figure I.9</u></b> : Structure de la carte d'interface.....                       | 18 |
| <b><u>Figure I.10</u></b> : structure de la carte d'acquisition et de commande.....     | 19 |

#### CHAPITRE II:

|  |    |
|--|----|
| <b><u>Figure II.1</u></b> : Schéma synoptique du système.....                            | 21 |
| <b><u>Figure II.2</u></b> : La carte Arduino UNO.....                                    | 22 |
| <b><u>Figure II.3</u></b> : Module GSM SIM800L.....                                      | 23 |
| <b><u>Figure II.4</u></b> : Capteur DH22.....  | 23 |
| <b><u>Figure II.5</u></b> : Pins capteur DH22.....                                       | 23 |
| <b><u>Figure II.6</u></b> : Afficheur LCD 16x2.....                                      | 24 |
| <b><u>Figure II.7</u></b> : Branchement afficheur LCD 16x2.....                          | 25 |
| <b><u>Figure II.8</u></b> : L'interface I2C.....   | 26 |
| <b><u>Figure II.9</u></b> : Montage de module I2C avec l'afficheur LCD et l'Arduino..... | 27 |

#### CHAPITRE III:

|   |    |
|---|----|
| <b><u>Figure III.1</u></b> : Circuit électronique globale du projet.....                            | 30 |
| <b><u>Figure III.2</u></b> : Partie déclaration dans le programme.....                              | 31 |
| <b><u>Figure III.3</u></b> : Fonction « setup ( ) » du programme.....                               | 32 |
| <b><u>Figure III.4</u></b> : Fonction « loop ( ) » du programme.....                                | 33 |
| <b><u>Figure III. 5</u></b> : Organigramme d'envoi d'un SMS de l'acquisition de la température..... | 36 |
| <b><u>Figure III.6</u></b> : Organigramme de haut niveau du logiciel.....                           | 37 |
| <b><u>Figure III. 7</u></b> : Intégrer carte Arduino sous Proteus.....                              | 38 |
| <b><u>Figure III. 8</u></b> : Fichier GSM library sous Proteus.....                                 | 39 |
| <b><u>Figure III. 9</u></b> : L'état initial du système.....  | 40 |

**Figure III. 10** : Information de températures et humidité.....41

**ANNEXE**

**ANNEXE I : L'Arduino**

**Figure 1** : Carte arduino.....45

**Figure 2** : Composant **arduino uno**.....46

**Figure 3** : brochage de la carte arduino uno.....48

**Figure 4** : L'interface utilisateur du logiciel Arduino.....53

**ANNEXE II : Simulation Proteus.**

**Figure 1** : Programme de Proteus 8 Professional.....54

**Liste des tableaux :**

**Chapitre I :**

**Tableau I.1** : Les types des AT COMMANDS et les réponses.....15

**Tableau I.2** : Commande AT dédiées service SMS.....16

**Chapitre II :**

**Tableaux II.1** : Brochage des afficheurs LCD.....25

## Table des matières :

|  |     |
|--|-----|
| Remerciements.....                                     | i   |
| Dédicaces.....   | ii  |
| Résumé de projet.....                                  | iii |
| Liste des abréviations .....                           | iv  |
| Liste des tableaux et des figures.....                 | v   |
| Table des matières.....                                | vii |
| Introduction générale.....                             | 02  |
| <b><u>Chapitre I:Présentation du projet</u></b>        |     |
| I.1 : Introduction .....                               | 06  |
| I.2 : Systèmes d'acquisition de données.....           | 06  |
| I.3 : Présentation de notre système.....               | 07  |
| I.4 : Le Réseau GSM.....                               | 08  |
| I.4.1 : Définition.....                                | 09  |
| I.4.2 : Réseau cellulaire.....                         | 09  |
| I.4.3 : Architecture du réseau GSM.....                | 10  |
| I.5 : Le Module GSM.....                               | 10  |
| I.5.1 : Définition.....                                | 10  |
| I.5.2 : Entrées digitales ou numériques NO ou NF.....  | 11  |
| I.5.3 : Sorties du module GSM.....                     | 11  |
| I.6: Module GSM SIM800L.....                           | 12  |
| I.6.1 : Définition.....                                | 12  |
| I.6.2 : Fonctionnalités.....                           | 12  |
| I.6.3 : LED clignotant.....                            | 12  |
| I.6.4 : Détails de la broche.....                      | 12  |
| I.7 : Commandes AT.....                                | 14  |
| I.7.1 : Fonctionnement du protocole AT.....            | 15  |
| I.7.2 : Les types des AT COMMANDS et les réponses..... | 15  |
| I.7.3 : Commande AT dédiées service SMS.....           | 16  |
| I.8 : Cahier de charge.....                            | 16  |
| I.8.1 : Structure générale du système.....             | 18  |
| I.8.2 : La carte d'acquisition.....                    | 18  |
| I.9 : Conclusion.....                                  | 19  |



## **Chapitre II : Etude théorique du projet**

|  |    |
|--|----|
| II.1 : Introduction.....                       | 21 |
| II.2 : Schéma synoptique du projet.....        | 21 |
| II.3 : Le matériel utilisé dans le projet..... | 22 |
| II.3.1 : Une carte Arduino uno.....            | 22 |
| II.3.2 : Le Module GSM SIM800L.....            | 23 |
| II.3.3 : Capteur DHT22.....                    | 23 |
| II.3.4 : Afficheur LCD 16X2.....               | 24 |
| II.3.5 : L'interface I2C.....                  | 26 |
| II.4 : CONCLUSION.....                         | 28 |

## **Chapitre III : Simulation du système avec Proteus**

|  |    |
|--|----|
| III.1 : Introduction.....  | 30 |
| III.2 : Schéma électronique du projet.....   | 30 |
| III.3 : Le logiciel.....   | 31 |
| III.3.1 : La conception du logiciel sous l'environnement Arduino.....                              | 31 |
| III.3.2 : Fonctionnalités du logiciel.....   | 33 |
| III.3.2.1 : Organigramme d'envoi d'un SMS de l'acquisition de<br>la température et d'humidité..... | 36 |
| III.3.2.2 : Organigramme de haut niveau du logiciel.....   | 37 |
| III.4 : Démarches de la simulation.....  | 38 |
| III.4 .1 : Bibliothèque Arduino pour Proteus.....  | 38 |
| III.4.2 : Bibliothèque de module GSM pour Proteus.....   | 38 |
| III.5. Résultats de la simulation.....   | 39 |
| III.5.1.L'état initial du système.....   | 39 |
| III.6. Conclusion.....   | 41 |
| Conclusion générale.....   | 43 |

## **ANNEXES**

|  |    |
|--|----|
| ANNEXE I : L'Arduino.....                  | 45 |
| Annexe I.1 :Qu'est-ce-que l'Arduino ?..... | 45 |
| Annexe I.2 : La carte Arduino.....         | 46 |
| Annexe I.3 : Le logiciel Arduino.....      | 52 |
| ANNEXE II : Simulation Proteus.....        | 54 |
| Bibliographie et Sites internet.....       | 56 |

# Introduction générale

## Introduction générale

De nos jours le réseau GSM *Global System For Mobile Communications* dans le monde entier, compte des millions d'utilisateurs. Le téléphone portable à aujourd'hui prit une place très importante dans notre société. Ce moyen de communication de haute technologie a ouvert la porte à de diverses applications électroniques sans fil en l'interfaçant avec un PC ou un microcontrôleur. Quelque soit le processus à gérer ; Il est possible de le faire via l'envoi et la réception d'un simple SMS. Le réseau GSM couvre actuellement la plus part des territoires en conséquence la distance n'est désormais plus un problème [1].

Avec l'avancement technologique et les techniques de plus en plus pointues, les moyens traditionnels de transfert d'information malheureusement, ne répondent plus aux critères d'efficacité et aux contraintes de temps qui deviennent de plus en plus rigoureux. Par conséquent, pour maîtriser les aléas qui peuvent être rencontrés et accommoder nos besoins, le recours aux moyens de communication perfectionnés devient inévitable.

Le réseau GSM fournit des services de transmission de la voix et éventuellement de données à bas débit dans un environnement mobile. L'architecture du réseau GSM repose sur un ensemble d'équipements spécifiques aux réseaux mobiles. Il permet aussi l'envoi et la réception des SMS, MMS, messages vocaux, la messagerie (email)... et tous ce qui fonctionne en utilisant cette technologie actuelle introduite dans notre vie quotidienne.

Dans ce concept, notre projet, se présente comme une solution simple et facile qui peut assurer une bonne manipulation des informations existantes pour concrétiser notre objectif primordial qui est l'apport d'une solution à la problématique de gain de temps et d'argent en utilisant les technologies de télécommunication par réseau GSM.

Dans notre thèse, nous proposons d'étudier et de simuler une carte d'acquisition et de commande qui permet le contrôle et la surveillance à distance d'un local quelconque, via le réseau GSM. Le système permet à l'utilisateur de connaître l'état de son siège, plus clairement nous utilisons un capteur DHT11 qui capte un changement dans la température et de l'humidité dans un système, et qui envoie un SMS ou un bip à l'utilisateur pour l'informer de n'importe quel problème survenant à l'intérieur de l'endroit où le système est placé. Cette commande peut être instantanée ou planifiée.

Notre travail est divisé en trois chapitres. Dans le premier chapitre, nous présentons notre projet en introduisant quelques concepts tels que : les systèmes d'acquisition de donnée, le cahier de charge du système , et nous définissons la marche à suivre en parlant en général du réseau GSM et des caractéristiques techniques de l'unité choisie, et ainsi nous parlons en détail du principe de fonctionnement de notre carte. Dans le deuxième chapitre nous donnons un bref aperçu sur les composants électroniques utilisés dans la conception de notre carte. Enfin le troisième chapitre sera consacré aux résultats de la simulation de notre système par logiciel Proteus. Nous terminerons ce mémoire par une conclusion générale.

# **Chapitre I :**

## **Présentation du projet**

## Chapitre I : Présentation du projet

---

### **I.1 : Introduction :**

Le téléphone mobile a largement de puissance pour accéder au réseau téléphonique via les antennes relais qui traversent notre territoire. Cela devient possible par la suite en envoyant ou recevoir des informations sous forme de messages courts pour suivre tout processus sur un site distant ou un site mobile. La réalisation de cette opération nécessite le développement d'une électronique indépendante capable d'interagir avec la télécommunication. C'est pourquoi nous proposons d'étudier une carte d'acquisition et de contrôle autour d'un module GSM ou d'un téléphone portable qui permet le contrôle à distance des systèmes.

### **I.2 : Systèmes d'acquisition de données :**

Un système d'acquisition de données (DAS ou DAQ) est un ensemble de capteurs, de composants matériels et de logiciels qui contribuent à la mesure d'un phénomène physique ou électrique tel que le courant, la tension, la pression, le son ou la température avec un ordinateur. Un système DAS peut être aussi simple qu'un ordinateur personnel équipé de périphériques matériels et de logiciels d'acquisition de données. Contrairement à un système de mesure traditionnel, le système DAS basé sur PC sollicite les capacités de productivité, de traitement, de connectivité et d'affichage des ordinateurs standards afin d'offrir une solution de mesure plus économique, flexible et puissante. Ces périphériques convertissent certains types de signaux environnementaux (comme le son, la pression, la température ambiante, le taux de circulation d'un fluide, etc.) en un signal électrique. Très souvent, le système DAS intègre un circuit de prétraitement de signal, comme un filtre analogique dans la chaîne du signal. Ce prétraitement prépare le signal à la conversion de données du format analogique au format numérique via le convertisseur analogique/numérique (CAN). Ensuite ces données converties sont envoyées à l'unité de traitement principale, qui peut être un ordinateur. L'unité de traitement principale manipule les données à l'aide de matériel et de logiciels dédiés. Même si le système DAS peut exécuter un logiciel personnalisé, il s'agit la plupart du temps de packages logiciels pour PC. Il est possible d'utiliser ceux-ci pour le contrôle du matériel qui équipe le système DAS. [2]

## Chapitre I : Présentation du projet

### I.3 : Présentation de notre système :

Le système de contrôle par GSM doit permettre le contrôle des fonctions et la réception des informations à base d'un réseau mobile GSM dans toutes les zones du monde qui sont sous couverture d'un réseau GSM entre utilisateur et le système. Dans la figure I.1, nous présentons le schéma bloc représentant les différentes fonctions du système.

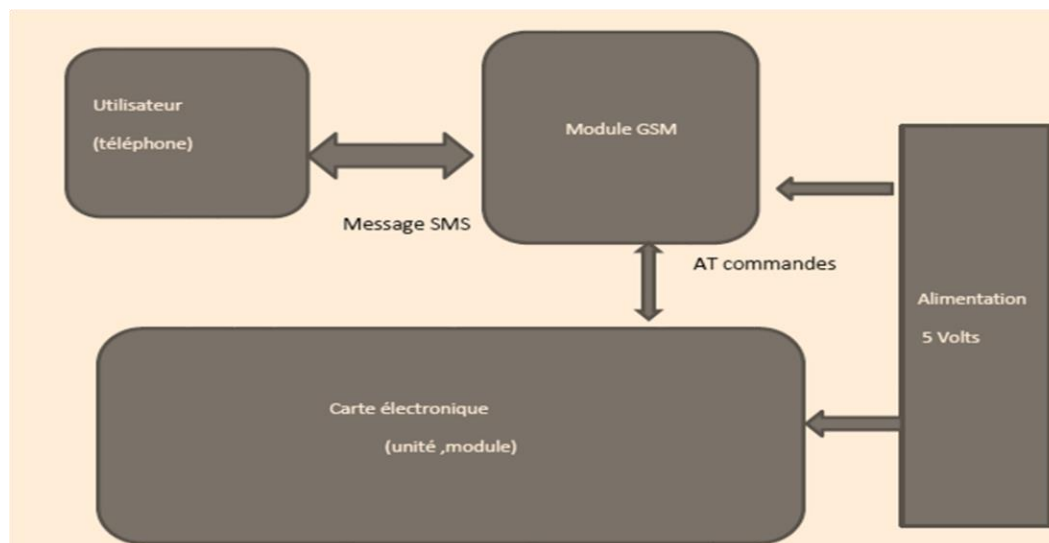


**Figure I.1** : Le schéma bloc du système.

**L'utilisateur** : permet l'envoi d'un message sms, qui contient des informations codées pour les commandes, à un module GSM par le biais d'un téléphone portable.

**Le Module GSM** : Il permet à son tour la réception ou l'envoi de sms et ainsi la communication avec une carte électronique par une communication série (UART).

**La Carte électronique**: elle permet la communication avec le module GSM, la lecture des messages, l'envoi de commandes et l'exécution des différents processus comme les traitements, le décodage et le stockage des informations, etc. [3].



**Figure I.2** : Schéma récapitulatif.

## Chapitre I : Présentation du projet

---

### **I.4 : Le Réseau GSM :**

#### **I.4.1 : Définition :**

Le réseau GSM (*Global System for Mobile communications*) est créé par la Conférence Européenne des administrations des Postes et télécommunications (CEPT) afin d'élaborer les normes de communications mobiles pour l'Europe dans la bande de fréquences de 890 à 915 [MHz] pour l'émission à partir des stations mobiles et 935 à 960 [MHz] pour l'émission à partir de stations fixes. Il constitue au début du 21ème siècle le standard de téléphonie mobile le plus utilisé en Europe. Il s'agit d'un standard de téléphonie dit « de seconde génération » (2G) puisque contrairement à la première génération de téléphones portables, les communications fonctionnent selon un mode entièrement numérique.

En Europe, le standard GSM utilise les bandes de fréquences 900 MHz et 1800 MHz. Aux Etats-Unis par contre, les bandes de fréquences utilisées sont les bandes 850 MHz et 1900 MHz. Ainsi, on qualifie de tri-bande (parfois noté tribande), les téléphones portables pouvant fonctionner en Europe et aux Etats-Unis et de bi-bande ceux fonctionnant uniquement en Europe. Le réseau GSM est adéquat pour les communications téléphoniques de parole. En effet, il s'agit principalement d'un réseau commuté, à l'instar des lignes ``fixes" et constitués de circuits, c'est-à-dire de ressources allouées pour la totalité de la durée de la conversation. Rien ne fut mis en place pour les services de transmission de données la norme GSM autorise un débit maximal de 9,6 kbps, ce qui permet de transmettre la voix ainsi que des données numériques de faible volume, par exemple des messages textes (SMS, pour Short Message Service) ou des messages multimédias (MMS, pour Multimedia Message Service) [4].

#### **I.4.2 : Réseau cellulaire :**

Le principe de ce système est de diviser le territoire en de petites zones, appelées cellules, et de partager les fréquences radio entre celles-ci. Ainsi, chaque cellule est constituée d'une station de base (reliée au Réseau Téléphonique Commuté, RTC) à laquelle on associe un certain nombre de canaux de fréquences à bande étroite, sommairement nommés fréquences.



## Chapitre I: Présentation du projet

---

L'ensemble des stations de base d'un réseau cellulaire est relié à un contrôleur de stations (en anglais Base Station Controller, noté BSC), chargé de gérer la répartition des ressources.

L'ensemble est constitué par le contrôleur de station et les stations de base connectées constituent le sous-système radio (en anglais BSS pour Base Station Subsystem). Enfin, les contrôleurs de stations sont eux-mêmes reliés physiquement au centre de commutation du service mobile (en anglais MSC pour Mobile Switching Center), géré par l'opérateur téléphonique, qui les relie au réseau téléphonique public et à internet. Le MSC appartient à un ensemble appelé sous-système réseau (en anglais NSS pour Network Station Subsystem), chargé de gérer les identités des utilisateurs, leur localisation et l'établissement de la communication avec les autres abonnés.

Le MSC est généralement relié à des bases de données assurant des fonctions complémentaires :

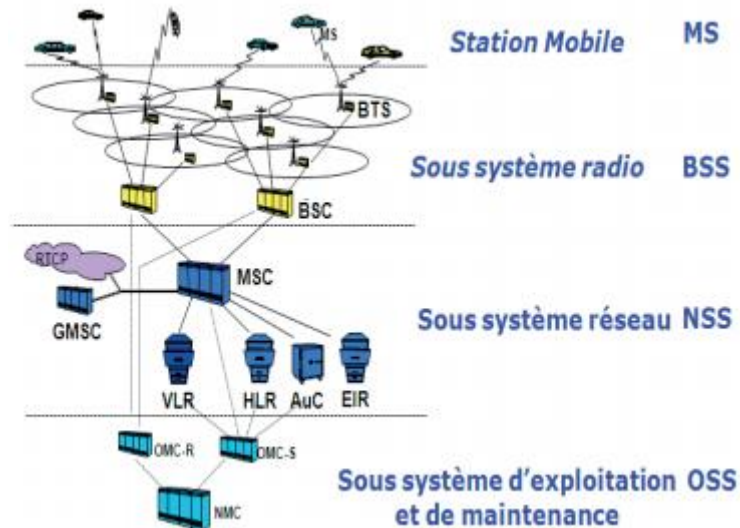
- ❖ Le registre des abonnés locaux (noté HLR pour Home Location Register): il s'agit d'une base de données contenant des informations (position géographique, informations administratives, etc.) sur les abonnés inscrits dans la zone du commutateur (MSC).
- ❖ Le Registre des abonnés visiteurs (noté VLR pour Visitor Location Register): il s'agit d'une base de données contenant des informations sur les autres utilisateurs que les abonnés locaux. Le VLR rapatrie les données sur un nouvel utilisateur à partir du HLR correspondant à sa zone d'abonnement. Les données sont conservées pendant tout le temps de sa présence dans la zone et sont supprimées lorsqu'il la quitte ou après une longue période d'inactivité (terminal éteint).
- ❖ Le registre des terminaux (noté EIR pour Equipment Identity Register) : il s'agit d'une base de données répertoriant les terminaux mobiles.
- ❖ Le Centre d'authentification (noté AUC pour Authentication Center) : il s'agit d'un élément chargé de vérifier l'identité des utilisateurs.

Le réseau cellulaire ainsi formé est prévu pour supporter la mobilité grâce à la gestion du hand over, c'est-à-dire le passage d'une cellule à une autre. Enfin, les réseaux GSM supportent également la notion d'itinérance (en anglais roaming), c'est-à-dire le passage du réseau d'un opérateur à un autre. [5].

## Chapitre I : Présentation du projet

### I.4.3 : Architecture du réseau GSM :

Le réseau GSM a pour rôle d'assurer la mobilité tout en conservant l'accès au réseau fixe. La figure 1.3 montre les entités principales du réseau GSM.



**Figure I.3 :** Un schéma représentant l'architecture du réseau GSM [6].

Le réseau GSM est composé de trois sous systèmes :

- ❖ le sous système radio appelé BSS (Base station Sub-System) : c'est le réseau d'accès radio qui assure la transmission radioélectrique et gère la ressource radio.
- ❖ le sous système d'acheminement appelé NSS (Network Sub-System) ou SMSS (Switching and Management Sub-System) qui comprend l'ensemble des fonctions nécessaires à l'établissement des appels et à la gestion de la mobilité. On peut dire que le NSS est le réseau cœur GSM.
- ❖ le sous système d'exploitation et de maintenance appelé OSS (Operation Sub-System) qui permet à l'opérateur d'administrer le réseau (coûts, performances, erreurs, sécurité,...).

### I.5 : Le Module GSM :

#### I.5.1 : Définition :

Les modules GSM sont des boîtiers électroniques autonomes (batterie interne), munis d'une carte SIM, comportant des entrées et des sorties. Ainsi comme un téléphone portable ils possèdent leur propre numéro de téléphone.

## Chapitre I : Présentation du projet

Les sorties sont actionnables en appelant le numéro de téléphone du boîtier ou par SMS. Lorsque vous appelez il rejette l'appel sans y répondre, donc il n'y a pas de frais de communication engagé, il a une mémoire non-volatile et sauvegarde les paramètres dans le cas d'une interruption de son alimentation.

Le module GSM est autonome grâce à sa batterie interne, ainsi en cas de coupure secteur il vous envoie un SMS pour vous le signaler. Un module GSM, envoie des SMS à plusieurs utilisateurs autorisés à les recevoir.

### I.5.2 : Entrées digitales ou numériques NO ou NF :

Pour brancher des détecteurs du type : Intrusion, présence, contact de porte, détecteur de fumée, Gaz, inondation, thermostat, Hygrométrie, Etc....

Ainsi vous recevrez un SMS lorsque ces seuils seront franchis.

### I.5.3 : Sorties du module GSM :

Les sorties sont activées en envoyant au modem GSM un SMS codé, un relais de puissance est alors activé pour commander tous matériels et équipements électriques ; Portes, portails, chauffages, climatisation, volets, moteurs, pompes,... etc.

Vous recevrez un SMS vous signalant la confirmation de l'ordre passée.



**Figure I.4** : Application d'un module GSM. [7].

## Chapitre I : Présentation du projet

### I.6: Module GSM SIM800L [8] :

#### I.6.1 : Définition :

Le module SIM800L est un petit module GSM / GPRS du monde avec une taille de 2.2 cm x 1.8 cm. C'est un module puissant qui démarre automatiquement et recherche automatiquement le réseau. Le module prend en charge le réseau GSM / GPRS quadri-bande, disponible pour la transmission à distance de données de messages SMS et GPRS. Le SIM800L communique avec le microcontrôleur via le port UART, et prend en charge les commandes telles que 3GPP TS 27.007, 27.005 et les commandes AT améliorées SIMCOM. En outre, la carte prend également en charge la technique A-GPS qui est appelée positionnement mobile et obtient la position par réseau mobile. Cette caractéristique fait qu'il peut également être un module de suivi.

Ce module nécessite une alimentation entre 3,4V et 4,4V. L'alimentation 5V de l'Arduino ne lui convient donc pas. Pour contrer ce problème d'alimentation, on ajoute une diode 1N4007 entre le 5V de l'Arduino et le pin VCC du SIM800L. Le SIM800L nécessite un pic de courant d'environ 2A.



**Figure I.5** : Module GSM SIM800L.

#### I.6.2 : Fonctionnalités :

- 2G quadri-bande 850/900/1800 / 1900MHz
- Recevoir et passer des appels à l'aide des sorties haut-parleur et microphone.
- Recevoir et envoyer des SMS.

## Chapitre I : Présentation du projet

---

- Écouter les émissions de radio FM
- Station mobile GPRS classe B.
- Contrôlé par AT Command (3GPP TS 27.007, 27.005 et SIMCOM Enhanced AT Commands)
- Prend en charge l'horloge en temps réel
- Plage de tension de fonctionnement 3,4 V ~ 4,4 V
- Prend en charge A-GPS
- Faible consommation d'énergie, 1mA en mode veille
- Carte micro SIM

### I.6.3 : LED clignotant :

Si l'alimentation du SIM800L est suffisante, la LED intégrée commence à clignoter en fonction de son fonctionnement. Si la puissance n'est pas suffisante, il recevra un signal médiocre et recherchera le réseau tout le temps (si vous recherchez toujours, ajustez légèrement le potentiomètre mais ne dépassez pas plus de 4,7 V). La fréquence du clignotement indique ce qui suit:

- **Chaque seconde:** recherche d'un réseau.
- **Toutes les trois secondes:** connectée à un réseau.
- **Deux fois par seconde:** connecté via GPRS.

### I.6.4 : Détails de la broche :

À l'arrière du module (figure 1.6), on peut trouver les détails de la broche imprimés dessus. Nous utiliserons 5 broches pour nous connecter à Arduino pour un fonctionnement de base. Ces broches sont :

**NET** - Broche d'antenne pour le module (type à ressort inclus). Vous pouvez utiliser une antenne externe via un connecteur uFL disponible sur le module pour une meilleure réception du signal.

**VCC** - Alimentation 3,4V à 4,4V avec min 2 Amp. (LM2596 avec une entrée adaptateur 9 / 12V 1Amp et il doit être ajusté à 4.2V recommandé).

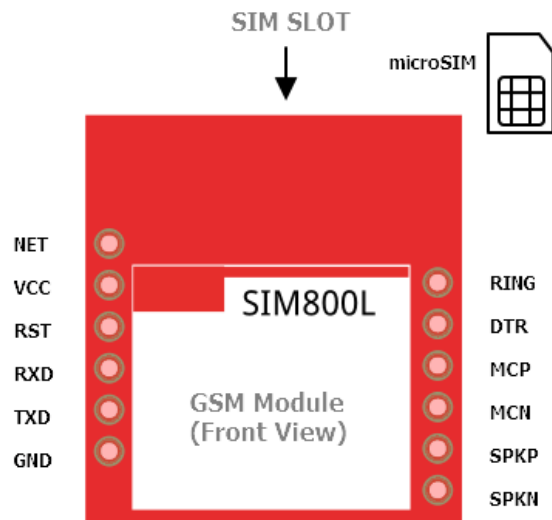
**RST** - Réinitialiser

**RXD** - Récepteur vers le module - Logique 3.3V (À connecter à la broche Arduino TX. Diviseur de tension recommandé).

**TXD** - Émetteur du module (à connecter à la broche Arduino RX).

**GND** - Terre.

## Chapitre I : Présentation du projet



**Figure I.6 :** Brochage du module GSM SIM800L.

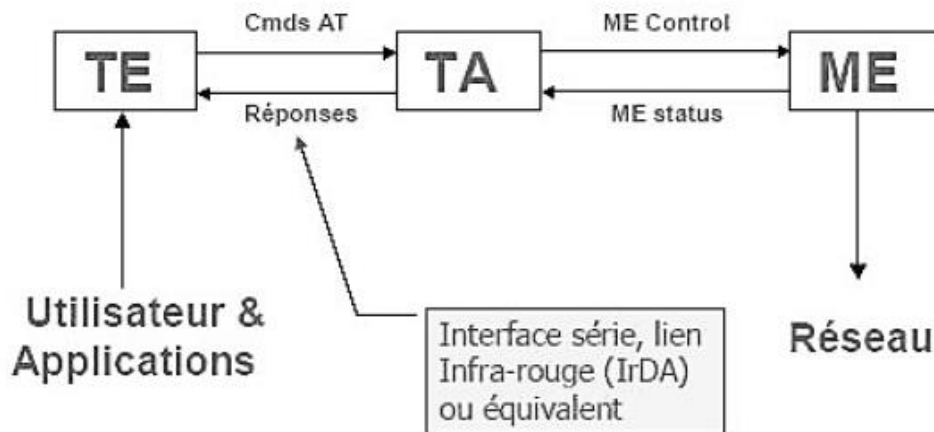
### I.7 : Commandes AT :

Les commandes Hayes parfois appelées commandes AT sont définies dans la norme GSM 07.07 (pour les SMS cf. GSM 07.05). AT est l'abréviation d'ATtention. Ces 2 caractères sont toujours présents pour commencer une ligne de commande sous forme de texte (codes ASCII). Les commandes permettent la gestion complète du mobile.

- Trois entités sont définies :
  - TE : Terminal Equipment (envoi et affiche les commandes).
  - TA : Terminal Adaptator (interface entre l'utilisateur et le mobile).
  - ME : Mobile Equipment.

## Chapitre I : Présentation du projet

### I.7.1 : Fonctionnement du protocole AT :



**Figure I.7 :** Schéma de fonctionnement. [9]

- **ME** (Mobile Equipement) : téléphone portable.
- **TE** (Terminal Equipement) : peut-être un ordinateur ou un microcontrôleur.
- **TA** (Terminal Adaptateur) : assure la liaison entre le ME et le TE.

**Remarque :** TA et ME forment une seule entité, par exemple un téléphone portable standard ou un terminal GSM contient dans son boîtier à la fois le TA et le ME.

### I.7.2 : Les types des AT COMMANDS et les réponses :

|                     |              |   |
|---------------------|--------------|---|
| Command de test     | AT+<X>=?     | Le ME renvoie la liste des paramètres et des plages de valeurs définies avec la commande d'écriture correspondante par processus internes |
| Command pour lire   | AT+<X>?      | Cette commande renvoie la valeur actuellement définie du paramètre ou des paramètres  |
| Command pour écrire | AT+<X>=<...> | Cette commande définit les valeurs de paramètre défini par l'utilisateur.   |
| Command d'exécution | AT+<X>       | La commande d'exécution lit les paramètres non variables affectés par les processus internes dans le GSM                                  |

**Tableau I.1 :** Les types des AT COMMANDS et les réponses.

## Chapitre I : Présentation du projet

### I.7.3 : Commande AT dédiées service SMS :

|         |   |
|---------|---|
| AT+CSMS | Sélection du service de messagerie                    |
| AT+CPMS | Sélection de ta zone mémoire pour le stockage des SMS |
| AT+CMGF | Sélection du format du SMS (PDU ou TEXT)              |
| AT+CSCA | Définition de l'adresse du centre de messagerie       |
| AT+CSDH | Affiche en mode TEXT le paramétrage des SMS           |
| AT+CSAS | Sauvegarde du paramétrage                             |
| AT+CRES | Restauration du paramétrage par défaut                |
| AT+CNMI | Indication concernant un nouveau SMS                  |
| AT+CMGL | Liste les SMS stockés en mémoire                      |
| AT+CMGR | Lecture d'un SMS                                      |
| AT+CMGS | Envoie un SMS   |
| AT+CMSS | Envoie d'un SMS stocké en mémoire                     |
| AT+CMGW | Écriture d'un SMS.                                    |
| AT+CMGD | Efface un SMS   |

**Tableau I.2** : Commande AT dédiées service SMS.

### I.8 : Cahier de charge

La réalisation du système va consister à mettre en œuvre un arduino uno, interfacé avec le module GSM, et pilotant une sortie de commande de puissance tout-ou-rien (type relais électromécanique). L'utilisateur pourra à l'aide de son téléphone portable, de commander, contrôler ou surveiller un système (ou une machine, voiture ...), en récupérant à tout instant l'état de son système via SMS. Cette commande peut être immédiate ou programmée.

Afin d'étudier notre carte, les étapes suivantes ont été suivies :

- ❖ Etude et choix d'un module GSM.
- ❖ Programmation.
- ❖ Conception et simulation du système.



## Chapitre I : Présentation du projet

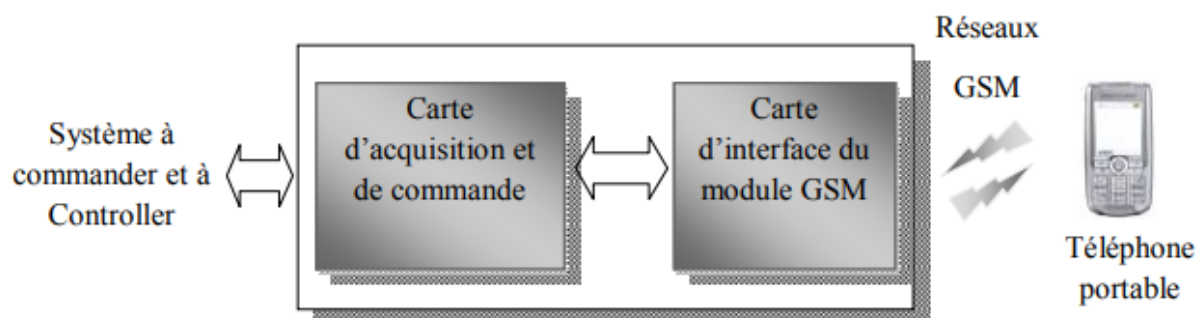
### I.8.1 : Structure générale du système :

Le fonctionnement du système (figure 1.10) est simple, il suffit d'interroger à l'aide d'un téléphone portable le module GSM (avec 1 bip par exemple), ce bip est scruté en permanence grâce au  $\mu$ C sur le port série (RX) de la carte d'acquisition, puis celle-ci fait l'acquisition des données des phénomènes physiques désirées par le biais du capteur, ensuite le  $\mu$ C transmet ces données sur le port série (TX) sous forme d'un SMS en suivant le chemin inverse.

Le choix de transmission des données sous forme d'un SMS, est justifié par différents critères :

- En situation normale, l'envoi d'un SMS est instantané
- jusqu'à ce que le mobile de destination soit opérationnel, le SMS lui sera transmis et les données ne seront pas perdues.
- la facilité déconcertante de lire les données en SMS
- la comptabilité assurée des caractères ASCII usuels (A...Z, a...z ,0...9) qui permet au  $\mu$ C de communiquer facilement avec le module GSM,
- le  $\mu$ C peut écrire les données utiles acquises en caractères ASCII sous forme de SMS en mode texte puis les envoyer via le Module GSM.

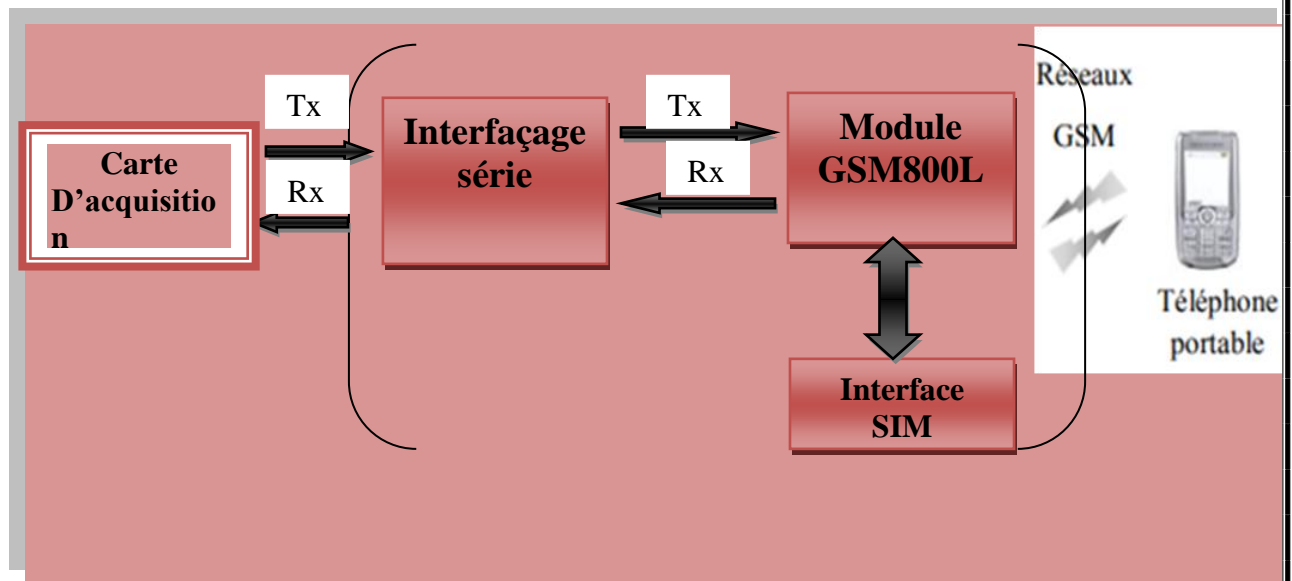
Il existe deux méthodes d'envoi et de réception des SMS (Short Message Service). Soit en utilisant le mode texte, soit en utilisant le mode PDU (Protocol Description Unit). On a choisi pour notre application d'utiliser le premier mode du fait qu'il est plus simple à utiliser. Le mode texte se base sur les commandes AT. Donc avec ces commandes, on peut lire, envoyer, effacer et recevoir des SMS en mode texte.



**Figure I.8:** Schéma descriptif du fonctionnement général du système.

## Chapitre I : Présentation du projet

La carte d'interface sert à adapter les signaux acquis ou émis par le port série avec la carte d'acquisition ou avec un ordinateur du 5V/3V. Elle est équipée par un module GSM 800L qui joue le rôle d'un récepteur des SMS envoyés par l'utilisateur qui seront traités par suite.

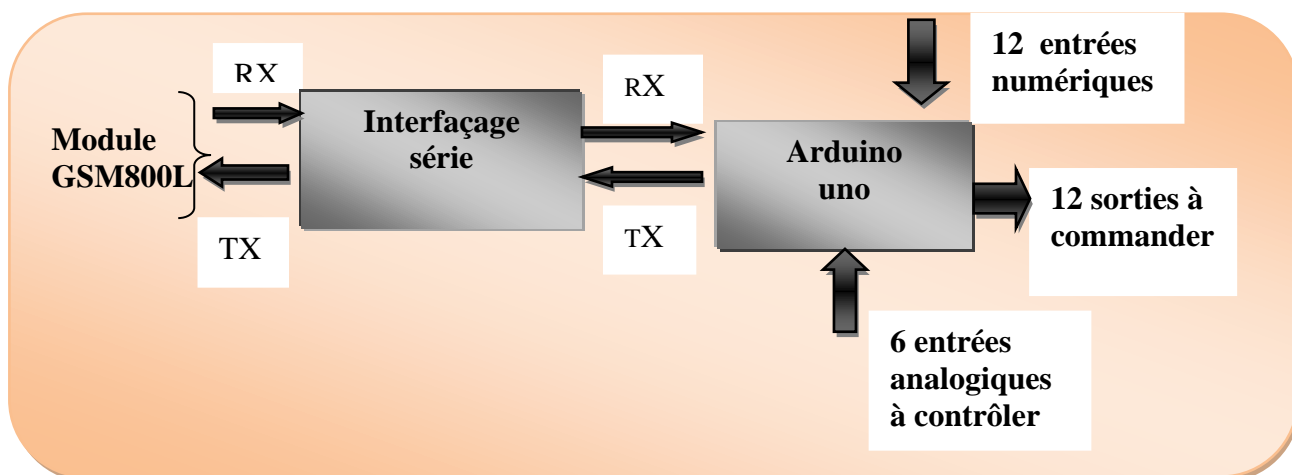


**Figure I.9:** Structure de la carte d'interface.

### I.8.2 : La carte d'acquisition :

Il s'agit d'une carte d'acquisition des données numérique et analogiques et de commande des relais à base d'un ARDUNO ATMEGA328P. En effet, cette carte est capable d'acquérir six entrées analogiques et douze entrées numériques. Elle pourra aussi faire la régulation et la génération des signaux de commande. Par conséquent, elle est polyvalente, capable de gérer diverses application. [10].

## Chapitre I : Présentation du projet



**Figure I.10:** structure de la carte d'acquisition et de commande.

### I.9 : Conclusion :

Dans ce chapitre, nous avons présenté notre système d'une manière générale. Nous avons expliqué quelques concepts qui rentrent dans l'étude et la simulation de notre système, tels que le module GSM, la carte d'acquisition et les commandes AT. Dans le chapitre suivant nous entamerons les différents composants électroniques qui rentrent dans la conception du système.

# **Chapitre II :**

## **Etude théorique du projet**

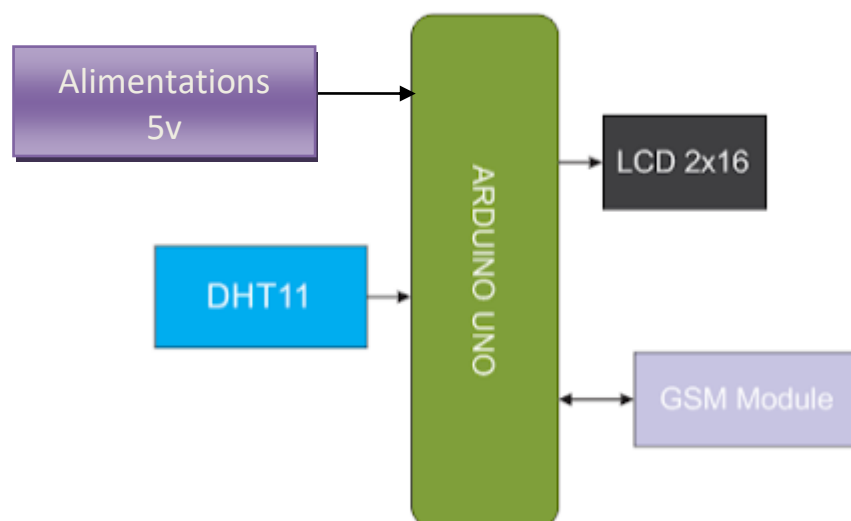
## Chapitre II : Etude théorique du projet

### II.1 : Introduction :

Ce chapitre présente une étude théorique des différents éléments qui composent le projet final de notre étude, qui est un système de carte d'acquisition et de contrôle autour d'une unité GSM via un téléphone mobile basé sur Arduino, et qui permet de surveiller le système à distance. Nous devons préciser que notre travail est une combinaison d'électronique et de programmation. Par conséquent nous allons aborder l'électronique embarquée qui est un sous-domaine de l'électronique et qui a le potentiel d'assembler la puissance de la programmation avec la puissance de l'électronique.

### II.2 : Schéma synoptique du projet :

Dans la figure 2.1 nous représentons le schéma synoptique de notre système. Il s'agit d'un module GSM en interface avec un arduino uno, comme la figure, le montre. L'objectif comme il a été déjà mentionné dans l'introduction générale est de surveiller un système en envoyant un message texte ou un appel vocal à l'utilisateur dans le but de l'informer de tout changement dans la température et de l'humidité.



**Figure II.1** : Schéma synoptique du système

## Chapitre II : Etude théorique du projet

---

Pour détailler un peu plus le fonctionnement de ce système, on peut dire quand une variation de température ou d'humidité est détectée par le capteur DHT11, ce dernier envoie un signal à l'arduino uno qui, envoie lui même une commande au registre à décalage. Le registre saisira le numéro de téléphone ce qui permettra l'envoi et la réception des messages. En outre, il est possible de savoir l'état du système à n'importe quel moment.

### II.3 : Le matériel utilisé dans le projet :

#### II.3.1 : Une carte Arduino uno : unité de traitement

Le microcontrôleur est un système électronique et informatique constitué de l'assemblage de plusieurs systèmes combinatoires et séquentiels tels que les registres, les bascules, les compteurs, les multiplexeurs, les démultiplexeurs, les décodeurs..., tous conçus à partir des portes logiques et donc des transistors et reprogrammables à l'aide de logiciel dédiés. Il existe un très grand nombre de modèles de microcontrôleurs. Il existe plusieurs fabricants qui proposent chacun plusieurs familles de microcontrôleurs, comptant chacune parfois des centaines de modèles.

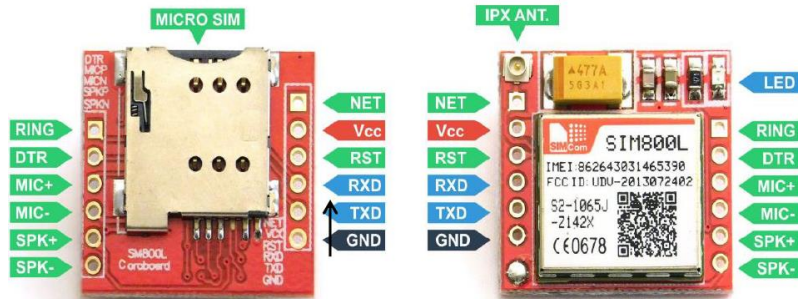
Dans ce projet, nous avons utilisé L'arduino qui est une carte à microcontrôleur, c'est-à-dire un ensemble constitué d'un microcontrôleur et toute la circuiterie nécessaire à son bon fonctionnement montés sur une seule carte de taille très réduite. La carte est accompagnée d'un logiciel multiplateforme servant à programmer la carte avec un langage de programmation dédié que l'on peut appeler le langage arduino, (plus de détails sur la carte dans « Annexe 1 »). Ce choix est motivé par plusieurs raisons surtout le nombre d'entrées sorties de la carte, l'espace mémoire et la facilité d'utilisation d'Arduino que ce soit dans le câblage ou la programmation.



**Figure II.2** : La carte Arduino UNO.

## Chapitre II : Etude théorique du projet

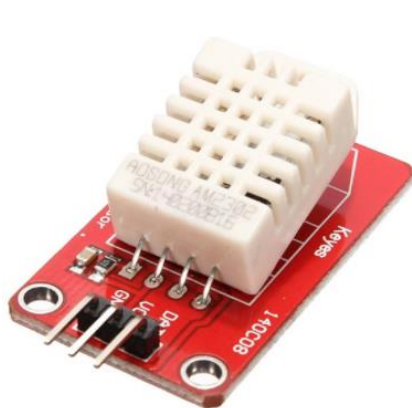
### II.3.2 : Le Module GSM SIM800L (voir chapitre I pages 12) :



**Figure II.3** : Module GSM SIM800L

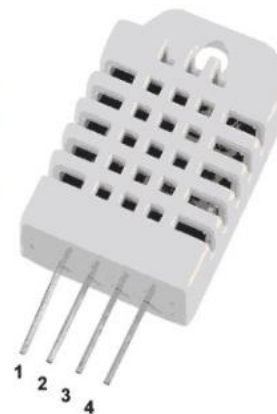
### II.3.3 : Capteur DHT22 [11]

Le DHT22 (figure 2.4 et 2.5), est un capteur numérique basique de température et d'humidité à faible coût. Il utilise un capteur d'humidité capacitif et une thermistance pour mesurer l'air environnant et crache un signal numérique sur la broche de données (aucune broche d'entrée analogique n'est nécessaire). Les connexions sont simples, la première broche à gauche à l'alimentation 3-5V, la deuxième broche à votre broche d'entrée de données et la broche la plus à droite à la terre.



**Figure II.4** : Capteur DH22

| DHT22 pins |      |
|------------|------|
| 1          | VCC  |
| 2          | DATA |
| 3          | NC   |
| 4          | GND  |



**Figure II.5** : Pins capteur DH22

## Chapitre II : Etude théorique du projet

---

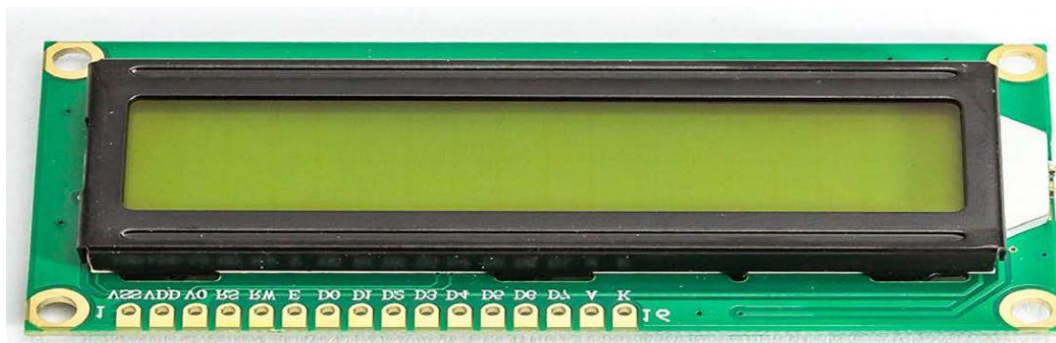
### ✚ Caractéristique du capteur DHT22 :

- ❖ Puissance: 3-5V
- ❖ Courant maximum: 2,5 mA
- ❖ Humidité: 0-100%, précision 2-5%
- ❖ Température: -40 à 80 ° C, précision  $\pm 0,5$  ° C

### II.3.4 : Afficheur LCD 16x2 [12] :

Les afficheurs à cristaux liquides sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils sont relativement bons marchés et s'utilisent avec beaucoup de facilité. L'afficheur LCD 2 lignes de 16 caractères est directement connecté aux broches du pic est alimenté par 5v, et comme le pic ne fait aucune interprétation des codes de commandes des afficheurs, il est compatible avec tous les modèles existants (de 1 à 4 lignes de 6 à 80 caractères).

Il existe deux interfaces normalisées : une version « parallèle » et une autre « série ». Dans notre étude l'afficheur communique avec le microcontrôleur via le PORTB en version parallèle où le transfert de données se fait en deux fois.



**Figure II.6** : Afficheur LCD 16x2.

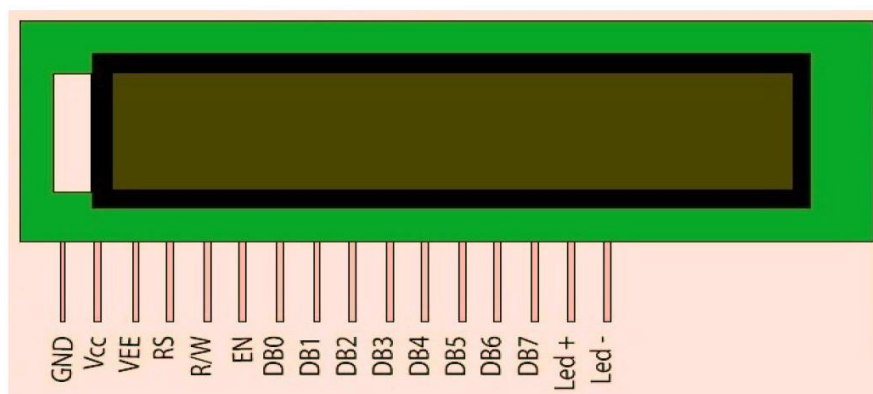


## Chapitre II : Etude théorique du projet

Le tableau 2.1 représente le brochage des afficheurs LCD et la figure 2.7 montre le branchement afficheur LCD 16x2.

| N° de pastille | Appellation | Fonction                                       |
|----------------|-------------|--|
| 1              | Vss         | Masse  |
| 2              | Vdd         | Alimentation de 5V                             |
| 3              | Vee         | Tension a appliqué pour gérer le contraste     |
| 4              | RS          | Registre Selecte (mode instruction ou données) |
| 5              | R/W         | Ecriture ou Lecture (Read/Write)               |
| 6              | E           | Enable (sélection de l'afficheur)              |
| 7-14           | DB0-DB7     | Data 0 à 7 (Données ou instructions)           |

**Tableau II.1** : Brochage des afficheurs LCD.



**Figure II.7** : Branchement afficheur LCD 16x2.

Dans notre application nous avons utilisé l'afficheur en écriture (RW=0), L'afficheur reçoit deux types de commande : soit des instructions permettant de gérer l'afficheur proprement dit (effacement de l'écran, affichage du curseur, clignotement du curseur... etc.), soit des données qui seront affichées à la position courante du curseur. La sélection du mode instruction est réalisée en mettant la broche RS à 0 (Registre Select) et en envoyant ensuite l'octet de commande sur le port de données (DB4 à DB7). Lorsque la broche RS est à 1 les valeurs envoyées sur les lignes DB4 à DB7 seront affichées sous forme de caractères ASCII à la position courante du curseur.

## Chapitre II : Etude théorique du projet

### II.3.5 : L'interface I2C [13] :

Ce LCD2004 est une excellente interface I2C (figure 2.8) pour les écrans LCD 2x16 et 4x20. Avec les ressources limitées en broches. Avec ce module LCD d'interface I2C, nous n'avons besoin que de 2 lignes (I2C) pour afficher la description: information. Il y a des milliers de composants qui utilisent une interface I2C, et les cartes de la famille Arduino peuvent toutes les contrôler. Les applications sont multiples : horloges temps réel, potentiomètres numériques, capteurs de température, boussoles numériques, mémoires, circuits radio FM, cartes d'extension d'entrées-sorties, contrôleurs d'afficheur LCD, accéléromètres, amplificateurs et bien d'autres encore. Nous pouvons connecter plusieurs composants adressables sur un même bus à tout moment (jusqu'à 112 périphériques I2C adressables sur un même bus en théorie). La figure 2.9 représente le montage de module I2C avec l'afficheur LCD et l'Arduino.

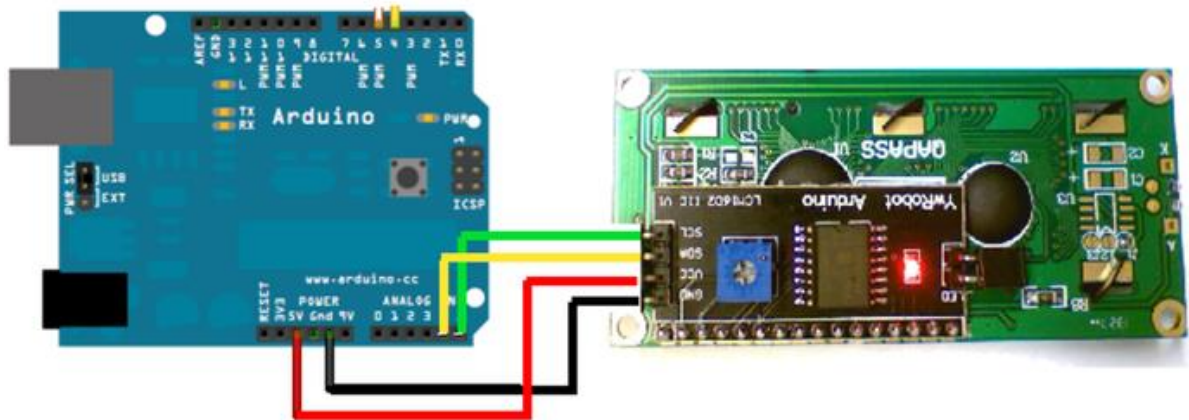


**Figure II.8** : L'interface I2C.

#### Spécifications :

- ❖ Compatible avec les écrans LCD 16x2 et 20x4
- ❖ Adresse I2C par défaut = 0X27
- ❖ Adresse sélectionnable - Plage de 0x20 à 0x27

## Chapitre II : Etude théorique du projet



**Figure II.9** : Montage de module I2C avec l'afficheur LCD et l'Arduino.

### II.4 : CONCLUSION

Ce chapitre de notre rédaction était consacré à la description du projet, à détailler son fonctionnement ainsi que sa tâche. Nous avons mentionné les composants électroniques qui composent l'ensemble du système. Nous avons présenté brièvement ARDUINO UNO, et nous remarquons qu'avec sa simplicité nous pouvons l'utiliser comme une unité de contrôle pour notre système. Dans le chapitre suivant, nous allons attaquer la partie simulation de la carte d'acquisition.

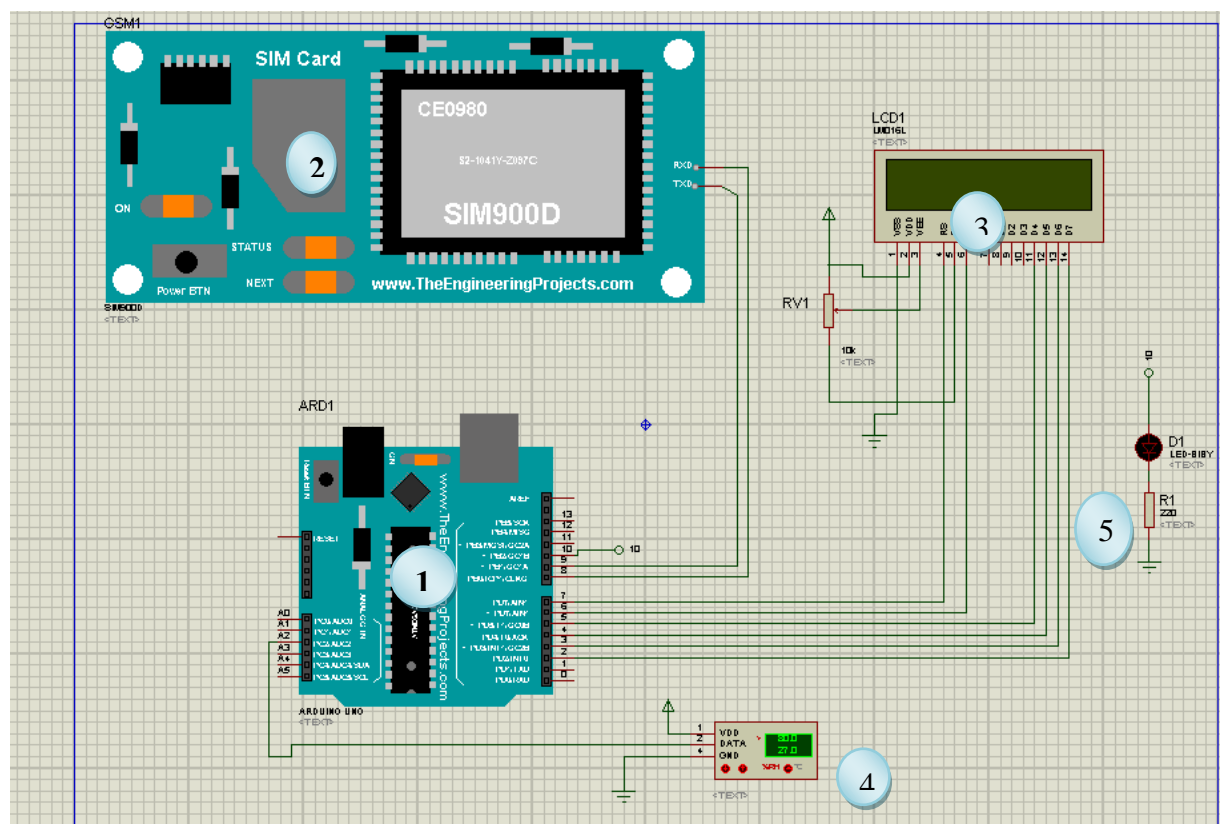
**Chapitre III :**  
**Simulation du système avec**  
**Logiciel Proteus**

## Chapitre III : Simulation du système avec Proteus

### III.1 : Introduction :

Dans ce chapitre, nous allons passer à la simulation du système. Nous allons d'abord commencer par l'élaboration d'un schéma électronique de câblage du système, ensuite nous réaliserons le logiciel afin de pouvoir simuler le projet avec le Proteus 7 Professional. Malheureusement et vu les circonstances, nous aurions souhaité continuer le travail en réalisant la maquette mais cela n'était pas possible à cause du confinement imposé par la pandémie du covid19.

### III.2 : Schéma électronique du projet



**Figure III.1 :** Circuit électronique globale du projet.

Ce schéma a été réalisé avec le logiciel Proteus 8 Professional, qui est en même temps le logiciel que nous avons choisi pour simuler notre projet. Les labels de même nom sont connectés. Nous avons numéroté les différents composants du schéma par des chiffres :

Le composant numéro 1 : la carte arduino uno.

Le composant numéro 2 : le module GSM (SIM900D).

## Chapitre III : Simulation du système avec Proteus

Le composant numéro 3 : c'est l'afficheur LCD 16x02. Le potentiomètre RV1 est celui qui permet de jouer sur le contraste de l'affichage

Le composant numéro 4 : c'est le capteur température et humidité (DHT22).

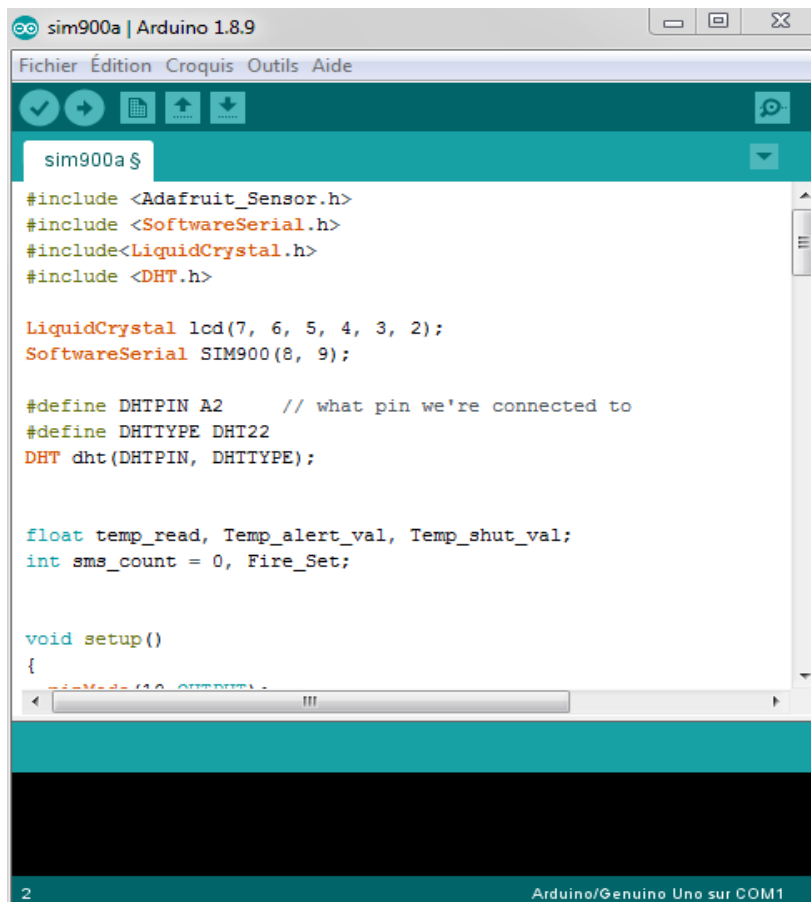
Le composant numéro 5 :c'est la LED.

### III.3 : Le logiciel :

#### III.3.1 : La conception du logiciel sous l'environnement Arduino :

La partie logicielle est la partie la plus complexe de ce projet. Le programme est évidemment écrit en langage Arduino dans la version 1.8.9 de l'environnement de développement. Des détails sur l'environnement de développement et la structure d'un programme Arduino sont donnés dans « Annexe 2.3 ». Comme tout programme Arduino, notre programme se décompose en trois (03) parties :

- ✓ Déclaration des variables, des constantes, l'inclusion des librairies...



```
sim900a | Arduino 1.8.9
Fichier Édition Croquis Outils Aide
sim900a $
#include <Adafruit_Sensor.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <DHT.h>

LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
SoftwareSerial SIM900(8, 9);

#define DHTPIN A2 // what pin we're connected to
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

float temp_read, Temp_alert_val, Temp_shut_val;
int sms_count = 0, Fire_Set;

void setup()
{
  pinMode(10, OUTPUT);
}
```

**Figure III.2** : Partie déclaration dans le programme.

Comme on peut le voir sur l'image, le programme utilise énormément de variables et 2 librairies à savoir :

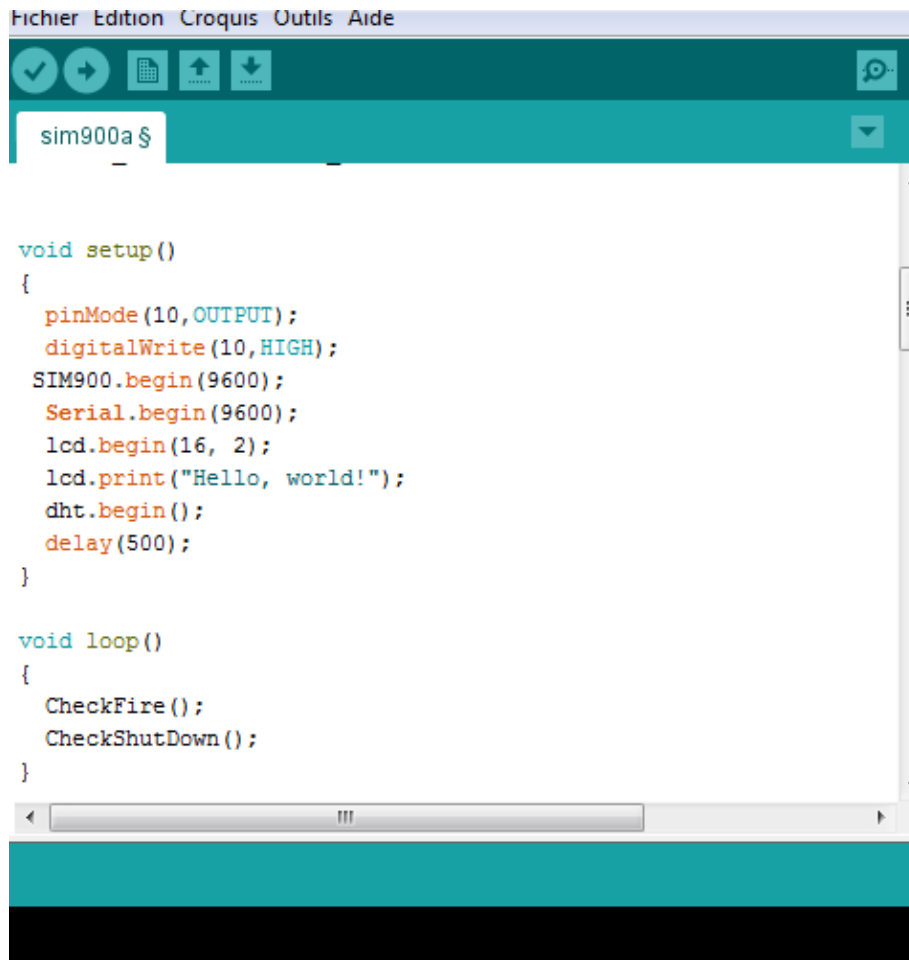
## Chapitre III : Simulation du système avec Proteus

---

La librairie : <Adafruit\_Sensor.h>

La librairie : <DHT.h> gère DHT

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

The image shows a screenshot of the Proteus IDE's code editor. The window title is 'sim900a \$'. The code is as follows:

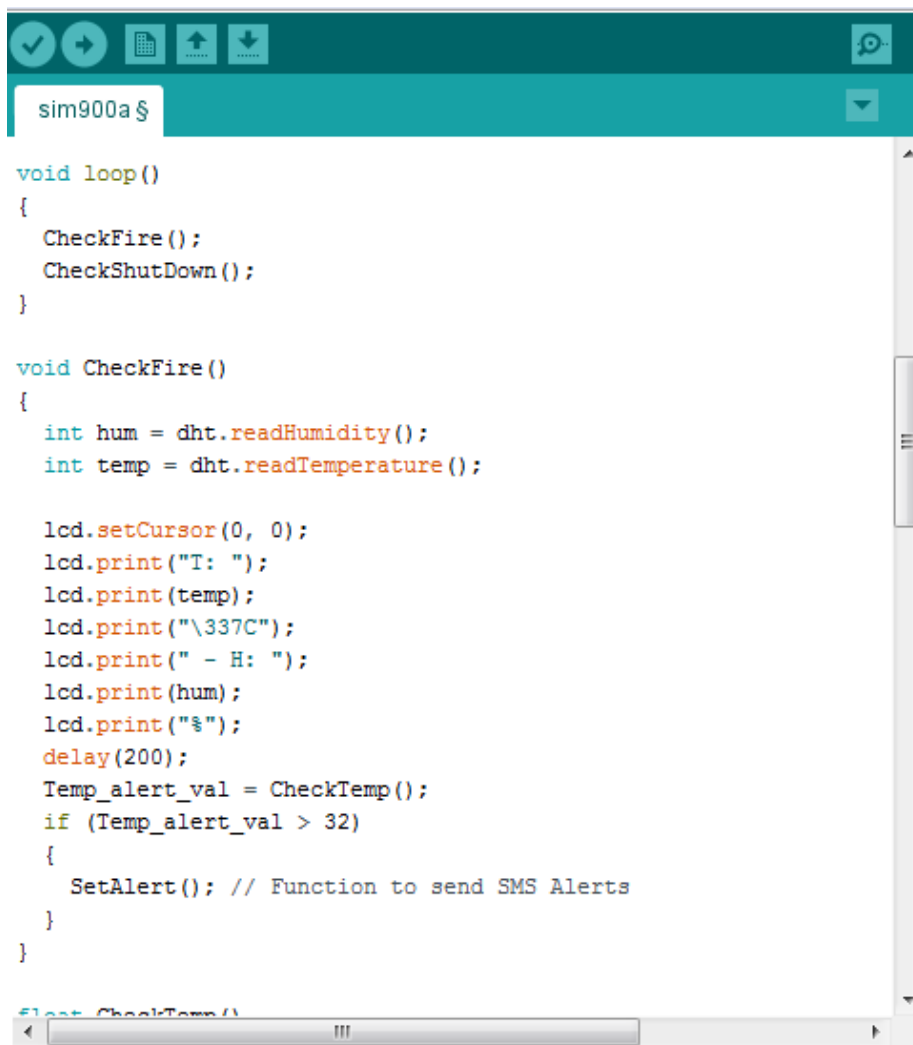
```
void setup()
{
  pinMode(10, OUTPUT);
  digitalWrite(10, HIGH);
  SIM900.begin(9600);
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("Hello, world!");
  dht.begin();
  delay(500);
}

void loop()
{
  CheckFire();
  CheckShutDown();
}
```

**Figure III.3** : Fonction « setup ( ) » du programme

## Chapitre III : Simulation du système avec Proteus

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

The image shows a screenshot of the Proteus IDE's code editor. The window title is 'sim900a \$'. The code is written in C and defines a 'loop()' function. Inside 'loop()', it calls 'CheckFire()' and 'CheckShutDown()'. The 'CheckFire()' function reads humidity and temperature from a DHT sensor, prints the temperature to an LCD, and checks if the temperature is above 32 degrees Celsius. If so, it calls 'SetAlert()' to send SMS alerts. A 'delay(200)' is used between sensor readings. The 'CheckTemp()' function is partially visible at the bottom of the screen.

```
void loop()
{
    CheckFire();
    CheckShutDown();
}

void CheckFire()
{
    int hum = dht.readHumidity();
    int temp = dht.readTemperature();

    lcd.setCursor(0, 0);
    lcd.print("T: ");
    lcd.print(temp);
    lcd.print("\337C");
    lcd.print(" - H: ");
    lcd.print(hum);
    lcd.print("%");
    delay(200);
    Temp_alert_val = CheckTemp();
    if (Temp_alert_val > 32)
    {
        SetAlert(); // Function to send SMS Alerts
    }
}

float CheckTemp()
```

**Figure III.4** : Fonction « `loop ()` » du programme.

On remarque que la fonction principale contient uniquement deux fonctions : `CheckFire ()`, et `CheckShutDown()`. Mais ces fonctions contiennent chacune environ une dizaine de fonctions et le programme compte plus de 1000 lignes dont nous citons tous les détails dans la section suivante « Fonctionnalités ». Après l'édition du code, il ne reste plus qu'à le compiler et le téléviser sur la carte.

### III.3.2 : Fonctionnalités du logiciel :

Lorsque nous développons des systèmes critiques (par exemple, la chaleur, l'humidité ou un incendie), le seul aspect important que nous devons garder à l'esprit



### Chapitre III : Simulation du système avec Proteus

est le scénario du monde réel. «Un incendie ou une surchauffe et une humidité» qui peuvent survenir à tout moment. Ce qui veut dire que notre système doit perpétuellement surveiller la chaleur et l'humidité 24h/7jours. Si nous regardons le programme, nous verrons qu'il ne contient que deux appels de fonction dans une boucle void () - c'est-à-dire Checkdht () et CheckShutDown ().

Checkdht () - C'est la fonction qui surveille les augmentations de température et d'humidité 24h/7jours. Cette fonction renvoie la température et l'humidité mesurées par DHT22 et les stocke dans la variable Temp\_alert\_val pour comparaison. Ces valeurs de température et d'humidité sont comparées à des valeurs spécifiques de 32 ° C et 60% respectivement. La température ambiante est généralement comprise entre 25 ° C et 30 ° C et entre 50% et 80% d'humidité sous les tropiques. Cela variera selon les continents et les lieux. Donc nous devons modifier cette valeur de comparaison en mesurant la température ambiante moyenne sur le site d'installation.

S'il fait trop chaud et humide, la température de la pièce augmentera au-dessus de 32 degrés (en quelques secondes) et le taux d'humidité de plus de 60%, la routine intérieure SetAlert () sera appelée. SetAlert () est la fonction qui contrôle le nombre d'alertes SMS envoyées à chaque numéro de mobile chargé dans le programme. Le nombre d'alertes SMS envoyés peut être modifié en modifiant l'état de la boucle de verrouillage. Sms\_count <3 - signifie que 3 alertes SMS seront envoyées à chaque numéro de téléphone mobile. Si nous voulons envoyer 5 alertes, nous devons simplement changer l'état d'arrêt en sms\_count <5. La fonction d'envoi de SMS (en utilisant les commandes AT) - SendTextMessage () sera appelée 3 fois si le nombre d'alertes SMS est 3. La fonction SendTextMessage () sera appelée autant de fois que le nombre d'alertes SMS spécifié dans le programme.

**Remarque** : Nous avons limité le nombre d'alertes SMS en utilisant une condition d'arrêt. Une fois que la température et l'humidité augmentent et que le nombre d'alertes SMS spécifié est envoyé, le système n'enverra plus de SMS! Le système suppose que son travail est terminé en envoyant des SMS. Les humains doivent intervenir pour ajuster la température ou l'humidité, par exemple pour ventiler ou refroidir une pièce. Après l'envoi des alertes, le système commencera à surveiller le processus d'arrêt. Une fois modifié, le système réactivera ses paramètres d'alerte SMS en remettant à zéro la variable sms\_count. CheckShutDown (), c'est la fonction qui

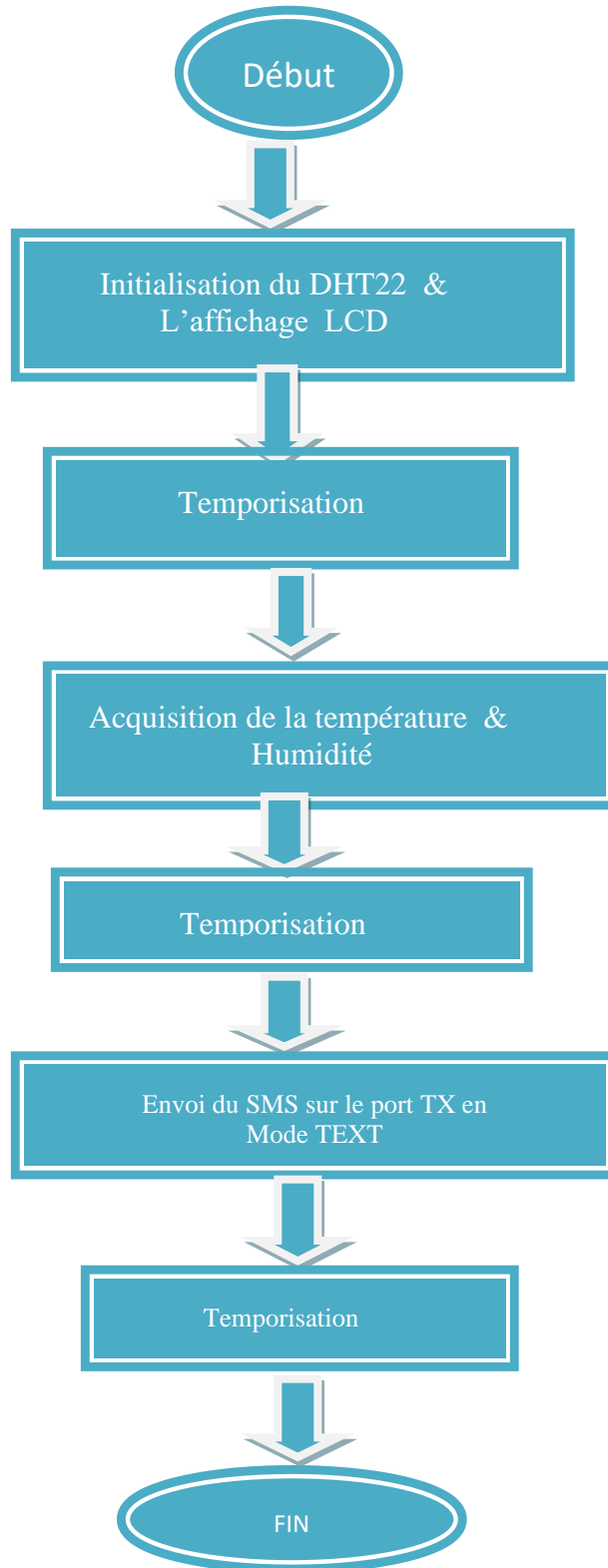
### **Chapitre III : Simulation du système avec Proteus**

---

surveille si elle est modifiée. Nous n'avons besoin de profiter de cette fonctionnalité que lorsque la température et l'humidité sont élevées. Pour limiter l'entrée aux expressions dans cette routine, nous avons introduit la variable Fire\_Set. Cet état changeant sera mis à 1 en cas de température et d'humidité élevées (vérifiez la publicité à l'intérieur de SetAlert ()). Les instructions à l'intérieur de CheckShutDown () ne seront exécutées que si la valeur Fire\_Set == 1. (S'il n'y a pas d'augmentation de température et d'humidité et que nous n'avons pas besoin de perdre du temps à mettre en œuvre les listes de contrôle d'arrêt). Nous considérons que la température et l'humidité peuvent être ajustées une fois que la température ambiante redevient normale.

## Chapitre III : Simulation du système avec Proteus

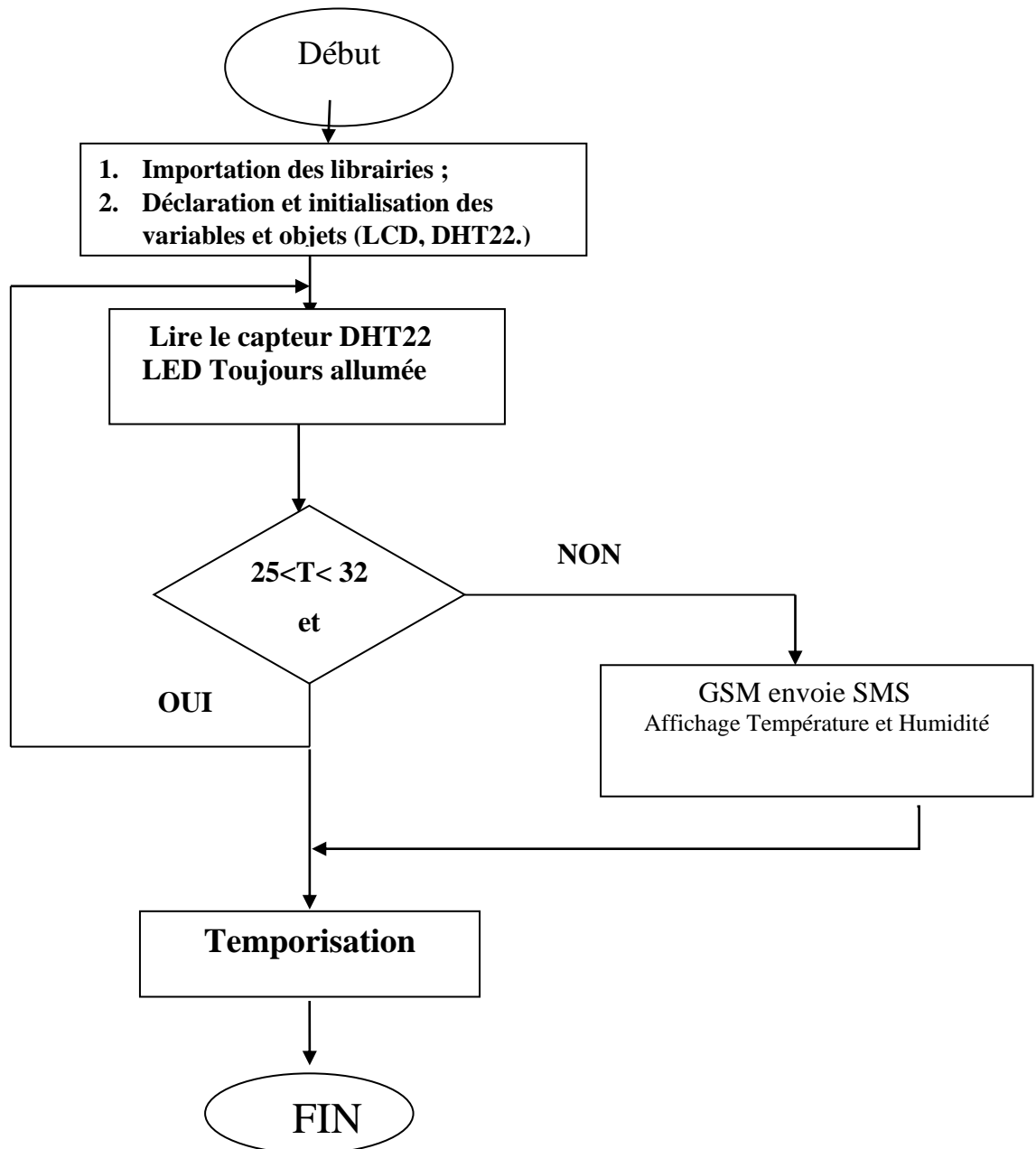
**III.3.2.1 :** Organigramme d'envoi d'un SMS de l'acquisition de la température et d'humidité



**Figure III. 5:** Organigramme d'envoi d'un SMS de l'acquisition de la température.

## Chapitre III : Simulation du système avec Proteus

### III.3.2.2 : Organigramme de haut niveau du logiciel.



**Figure III.6** : Organigramme de haut niveau du logiciel.

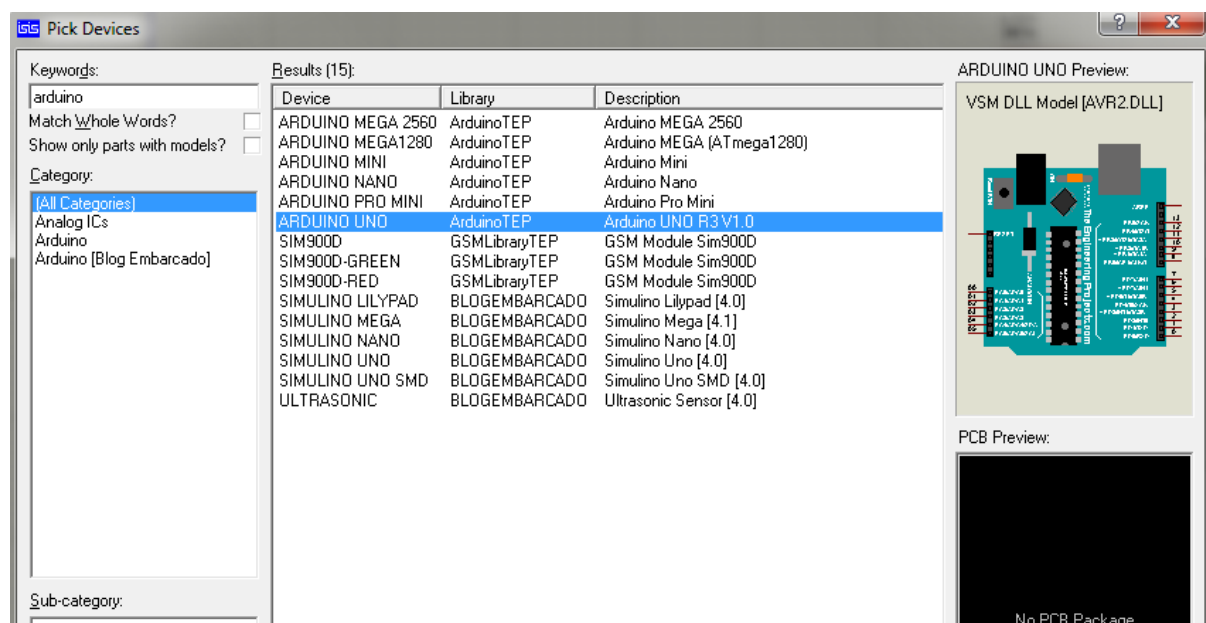
## Chapitre III : Simulation du système avec Proteus

### III.4 : Démarches de la simulation :

#### III.4.1 : Bibliothèque Arduino pour Proteus :

Il existe un très grand nombre de bibliothèques permettant d'étendre des fonctionnalités du langage Arduino. Elles permettent de faciliter la programmation de capteurs, préactionneurs, modules des communications et d'affichage, ...

Pour simuler la carte Arduino sur Proteus, on commence par télécharger la librairie Arduino disponible sur internet, ensuite on doit décompresser et copier les deux fichiers nommés « ArduinoTEP.LIB et Arduino.IDX » et les placer dans le dossier des bibliothèques de notre logiciel Proteus. Enfin il faut redémarrer le logiciel Proteus et dans la recherche de sections de composants pour ArduinoTEP on choisit Arduino Uno comme indiqué ci-dessous:



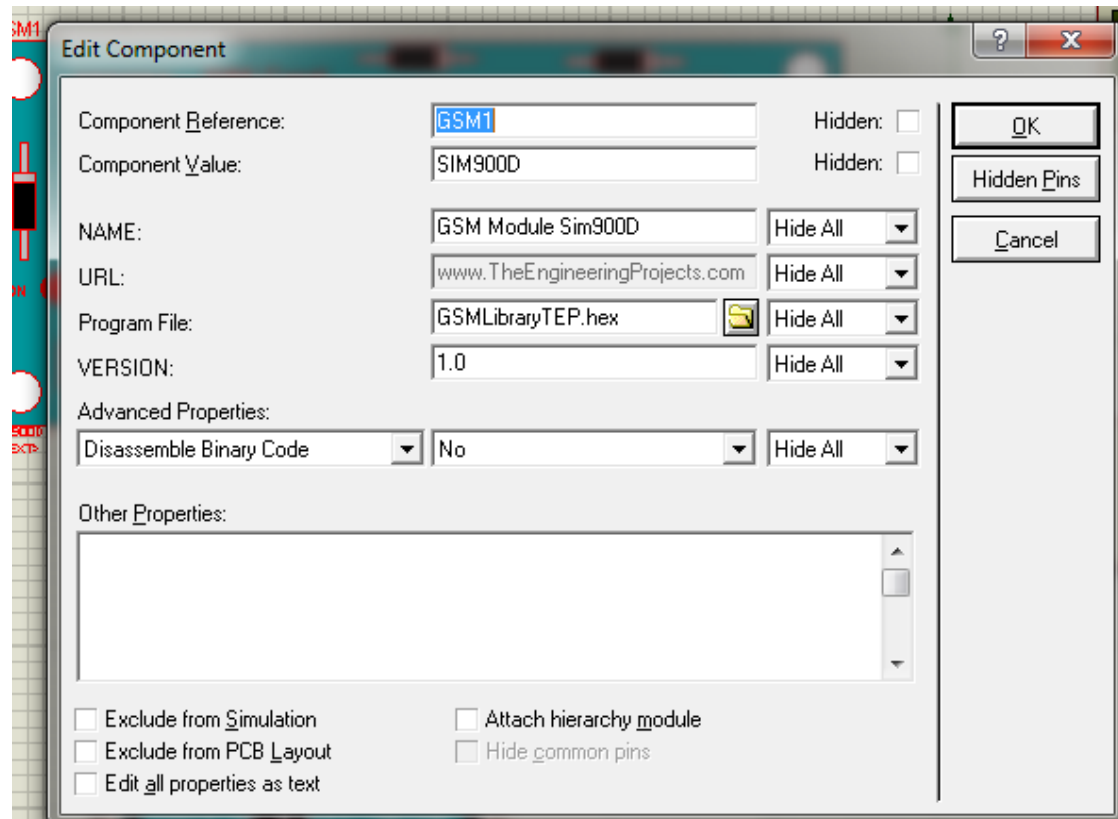
**Figure III. 7 :** Intégrer carte Arduino sous Proteus.

#### III.4.2 : Bibliothèque de module GSM pour proteus:

Pour cette bibliothèque du module GSM pour Proteus, elle est disponible sur internet et présente trois fichiers nommés « GSM library TEP.IDX, GSM library TEP.LIB, GSM library TEP.HEX ». Après les avoir téléchargés, nous les placeront dans la bibliothèque du logiciel Proteus. Au début, le module GSM est hors service.

## Chapitre III : Simulation du système avec Proteus

Pour ajouter ses fonctionnalités, nous devons double-cliquer sur ce le module GSM, ouvrir ses propriétés et sélectionner la section fichier du programme et choisir le fichier GSM library TEP.HEX comme indique la figure 3.8.



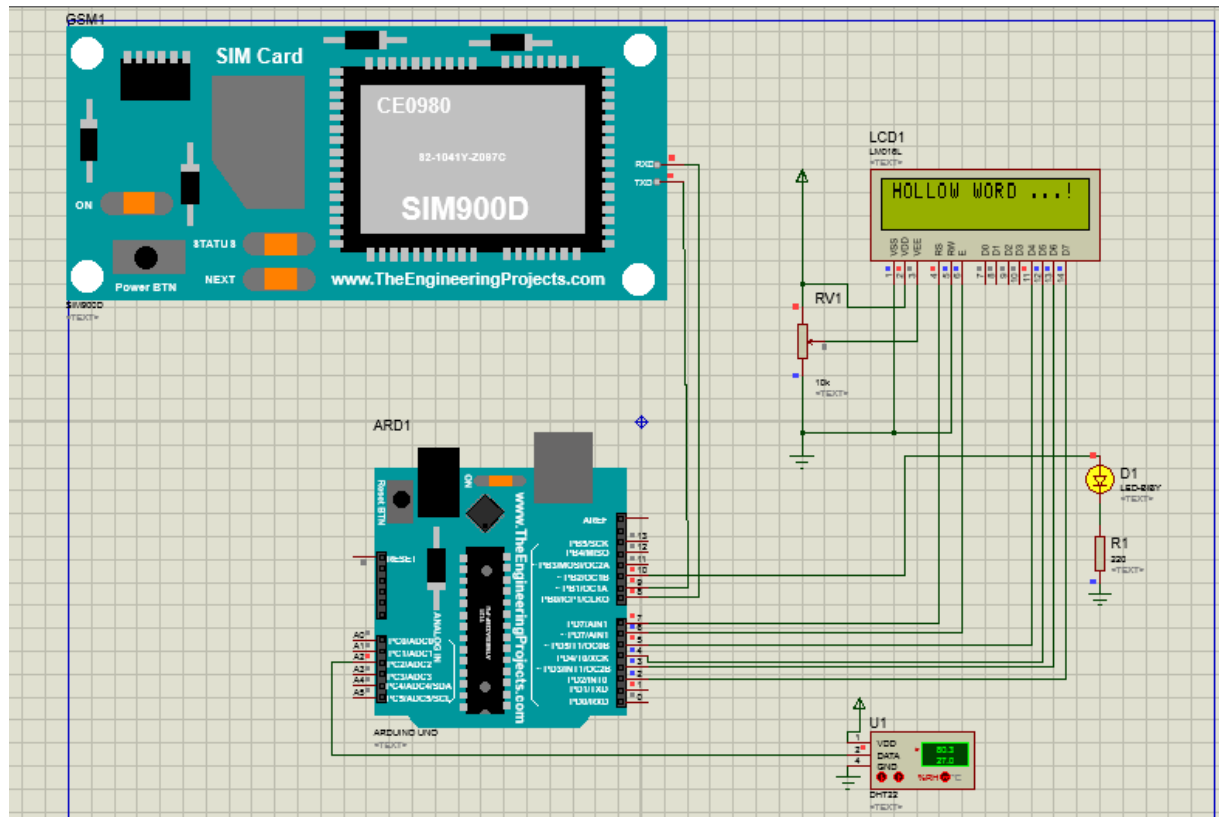
**Figure III. 8:** Fichier GSM library sous Proteus.

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

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

L'afficheur affiche des informations concernant notre projet (HELLO WORD... !...etc.).

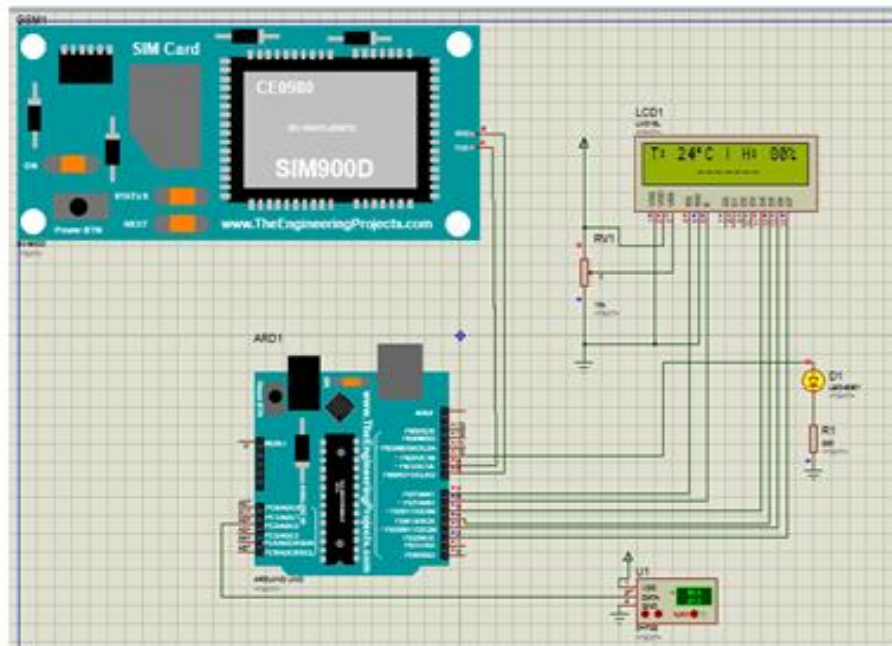
## Chapitre III : Simulation du système avec Proteus



**Figure III. 9 :** L'état initial du système.

Ultérieurement, l'afficheur affiche l'état du système et les informations concernant la température et le taux d'humidité et parallèlement il envoie des SMS à son utilisateur (figure 3.10).

## Chapitre III : Simulation du système avec Proteus



**Figure III. 10** : Information de températures et humidité

### III.6. Conclusion :

Dans ce chapitre la simulation de notre système a été présentée, incluant toutes les démarches de simulation ainsi que l'état initial et final du système. Il nous semble que le résultat obtenu est très satisfaisant, et les tests ont été réussis. Malheureusement, il manque la partie réalisation pour s'assurer du bon fonctionnement de notre projet pratiquement.



# Conclusion générale

## Conclusion générale

La conception et la réalisation de cartes d'acquisition de capteurs basées sur le microcontrôleur ARDUINO et autour du module GSM était l'objectif principal de notre projet, en effet notre application permet la mesure et la surveillance à distance d'un ou plusieurs systèmes mobiles ou distants. Malheureusement nous nous sommes contentés seulement de la partie simulation pour les raisons citées ultérieurement.

Le développement de ce projet comprend plusieurs parties, notamment, l'étude et la sélection de tous les composants du système, la programmation et enfin la simulation. Au cours de ce projet, nous avons eu l'occasion d'étudier et d'utiliser la diversité matérielle et logicielle (programmation, simulation et perception Arduino avec Proteus 8 Commandes AT et cryptage SMS ...) qui ont été très utiles pour notre projet et aussi pour approfondir nos connaissances.

Enfin, nous pouvons citer quelques avantages et inconvénients de notre système utilisant le GSM:

- ❖ L'avantage de notre système:
  - ✓ Portée illimitée dans une zone couverte par l'opérateur téléphonique.
  - ✓ débit élevé.
  - ✓ Gain en temps.
  - ✓ La taille de l'unité est petite et facile à entrer dans la réalité.
  - ✓ Surveillance à distance et réduction du nombre de déplacements.
- ❖ Inconvénients de notre système:
  - ✓ Le coût de la puce (module) nécessite un abonnement.
  - ✓ Un téléphone de l'un des trois opérateurs.
  - ✓ Consommation électrique très élevée. (300 mA à 3,8 V).

# ANNEXES

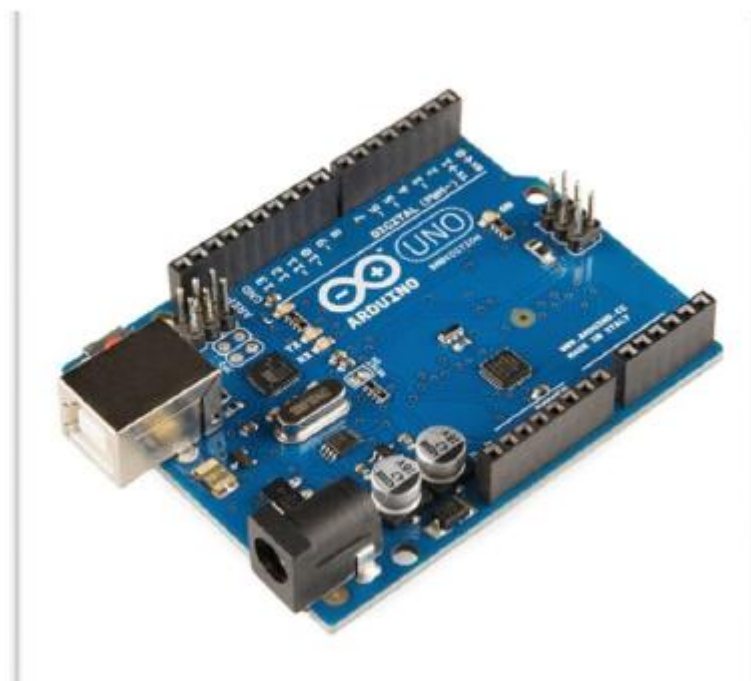
## ANNEXE I : L'Arduino

### **Annexe I.1 : Qu'est-ce que l'Arduino ?**

Arduino est une plateforme de prototypage à bas coût sous licence Creative Commons. Basée sur les microcontrôleurs ATMEL, elle permet de réaliser rapidement des projets électroniques sur les trois plateformes Linux, Mac et Windows.

Il est peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer avec ses logiciels (ex. : Macromedia Flash, Processing, Max/MSP, Usine Hollyhock, Pure Data, SuperCollider ...). En 2011, les versions vendues sont préassemblées. Des informations sont fournies pour ceux qui souhaitent assembler l'Arduino eux-mêmes.

«Le projet Arduino a reçu un titre honorifique à l'Ars Electronica 2006, dans la catégorie Digital Communities ». « 14 »



**Figure 1** : Carte arduino

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328 ou ATmega2560 pour les versions récentes, ATmega168 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).

Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

Les modules sont programmés au travers d'une connexion série RS-232, mais les connexions permettant cette programmation diffèrent selon les modèles. Les premiers Arduino possédaient un port série, puis l'USB est apparu sur les modèles UNO, tandis que certains modules destinés à une utilisation portable se sont affranchis de l'interface de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble).

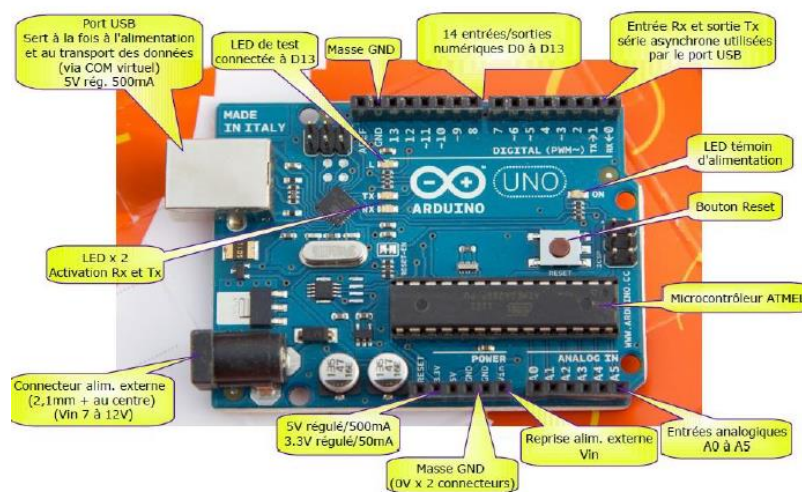
L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits.

### Annexe I.2 : La carte Arduino

La carte Arduino Uno est une carte à microcontrôleur basée sur ATmega328.

Elle dispose :

- ❖ de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- ❖ de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- ❖ d'un quartz 16Mhz,
- ❖ d'une connexion USB,
- ❖ d'un connecteur d'alimentation jack,
- ❖ d'un connecteur ICSP (programmation "in-circuit"),
- ❖ et d'un bouton de réinitialisation (reset). [14]



**Figure 2** : Composant arduino uno

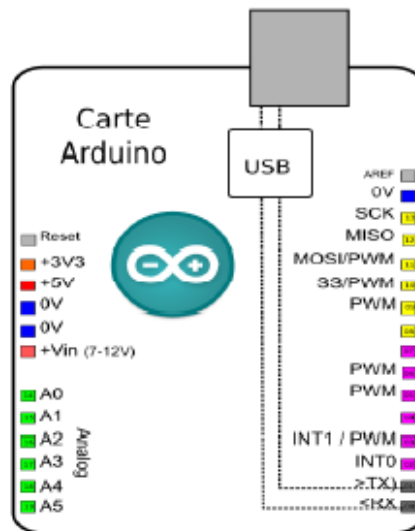
Il contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB (ou de l'alimenter avec un adaptateur secteur ou une pile, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

La carte Arduino Uno diffère de toutes les cartes précédentes car elle n'utilise par le circuit intégré FTDI usb-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

La carte UNO et la version 1.0 du logiciel seront la référence des versions Arduino à venir. La carte Uno est la dernière d'une série de carte USB Arduino, et le modèle de référence des plateformes Arduino; pour une comparaison avec les versions précédentes. [12]

|  |   |
|--|---|
| <b>Microcontrôleur</b>                               | ATmega328   |
| <b>Tension de fonctionnement</b>                     | 5V  |
| <b>Tension d'alimentation (recommandée)</b>          | 7-12V   |
| <b>Tension d'alimentation (limites)</b>              | 6-20V   |
| <b>Broches E/S numériques</b>                        | 14 (dont 6 disposent d'une sortie PWM)                                    |
| <b>Broches d'entrées analogiques</b>                 | 6 (utilisables en broches E/S numériques)                                 |
| <b>Intensité maxi disponible par broche E/S (5V)</b> | 40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)          |
| <b>Intensité maxi disponible pour la sortie 3.3V</b> | 50 MA   |
| <b>Intensité maxi disponible pour la sortie 5V</b>   | Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul |
| <b>Mémoire Programme Flash</b>                       | 32 KB (ATmega328) dont 0.5 KB sont utilisés par le bootloader             |
| <b>Mémoire SRAM (mémoire volatile)</b>               | 2 KB (ATmega328)  |
| <b>Mémoire EEPROM (mémoire non volatile)</b>         | 1 KB (ATmega328)  |
| <b>Vitesse d'horloge</b>                             | 16 MHz  |

### ❖ Brochage de la carte Uno



**Figure 3** : brochage de la carte arduino uno.

### ❖ Alimentation :

La carte Arduino Uno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

Les broches d'alimentation sont les suivantes :

- ❖ VIN. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). Vous pouvez alimenter la carte à l'aide de cette broche,

ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.

- ❖ 5V. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de tout autre source d'alimentation régulée.
- ❖ 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- ❖ GND. Broche de masse (ou 0V). « 14»
- ❖ **Entrées et sorties numériques**

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

- ❖ **Communication Serie:** Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
- ❖ **Interruptions Externes:** Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt()` pour plus de détails.



- ❖ **Impulsion PWM (largeur d'impulsion modulée):** Broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- ❖ **SPI (Interface Série Périphérique):** Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- ❖ **I2C:** Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie `Wire/I2C` (ou TWI - Two-Wire interface - interface "2 fils") .
- ❖ **LED:** Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

- **Communication**

La carte Arduino Uno dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte Arduino, ou avec d'autres microcontrôleurs. L'ATmega 328 dispose d'une UART (Universal Asynchronous Receiver Transmitter ou émetteur-récepteur asynchrone universel en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et 1 (TX). Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire. Cependant, sous Windows, un fichier .inf est requis.

Le logiciel Arduino inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte Arduino. Les LEDs RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré USB-vers-série et la connexion USB vers l'ordinateur (mais pas pour les communications série sur les broches 0 et 1). Une librairie Série Logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega 328 supporte également la communication par protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils") et SPI :

- ❖ Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C. Voir la documentation pour les détails
- ❖ Pour utiliser la communication SPI (Interface Série Périphérique), la librairie pour communication SPI est disponible. [14]

Dix-sept versions des cartes de type Arduino ont été produites et vendues dans le commerce à ce jour dans l'ordre chronologique ci-dessous :

1. *Serial Arduino* programmé avec une connexion série par connecteur DB9 et utilisant un ATmega8.
2. *Arduino Extreme*, programmable via une connexion USB et utilisant un ATmega8.
3. *Arduino Mini*, une version miniature de l'Arduino utilisant un ATmega168 de type CMS.
4. *Arduino Nano*, une version encore plus petite de l'Arduino alimenté par USB et utilisant un ATmega168 (avant la version 3) ou ATmega328 (à partir de la version 3.0) de type CMS.
5. *LilyPad Arduino*, une conception de type minimaliste pour permettre une application portable utilisant un ATmega168 de type CMS.
6. *Arduino NG*, programmable via une connexion USB et utilisant un ATmega8.
7. *Arduino NG plus*, programmable via une connexion USB et utilisant un ATmega168.
8. *Arduino Bluetooth(BT)*, programmable via une connexion Bluetooth et utilisant un ATmega 3284.
9. *Arduino Diecimila*, possède une interface USB et utilise un ATmega168 dans un boîtier format DIL28. (16 ko flash, 1 ko SRAM, 0,5 ko EEPROM)
10. *Arduino Duemilanove* (2009) utilise un ATmega168 et est alimenté en électricité par le connecteur USB ou une alimentation externe avec commutation automatique<sup>5</sup>. La nouvelle version<sup>6</sup> est équipé d'un ATmega328 (32 ko de flash, 2 ko de SRAM, et 1 ko d'EEPROM).

11. *Arduino Mega* est équipé d'un ATmega1280 de type CMS pour avoir des Entrées/Sorties supplémentaires et de la mémoire (128 ko flash, 8 ko SRAM, 4 ko EEPROM)<sup>7</sup>.

12. *Arduino Uno* utilise un ATmega328 comme les derniers modèles de Duemilanove, mais alors que le Duemilanove utilisait une puce FTDI pour la programmation via un connecteur USB, le Uno utilise une puce ATmega8U2 programmé comme un convertisseur série<sup>8</sup>.

13. *Arduino Mega2560* est équipé d'un ATmega2560 de type CMS, augmentant la mémoire totale disponible à 256 ko. Il est équipé aussi de la nouvelle puce USB ATmega8U2<sup>9</sup>.

14. *Arduino Ethernet* est une carte Arduino UNO intégrant un chip Wiznet W5100 pour rajouter la connectivité Ethernet intégré<sup>10</sup>.

15. *Arduino Leonardo* est une version bas coût de l'Arduino UNO à base d'un ATmega32U4.

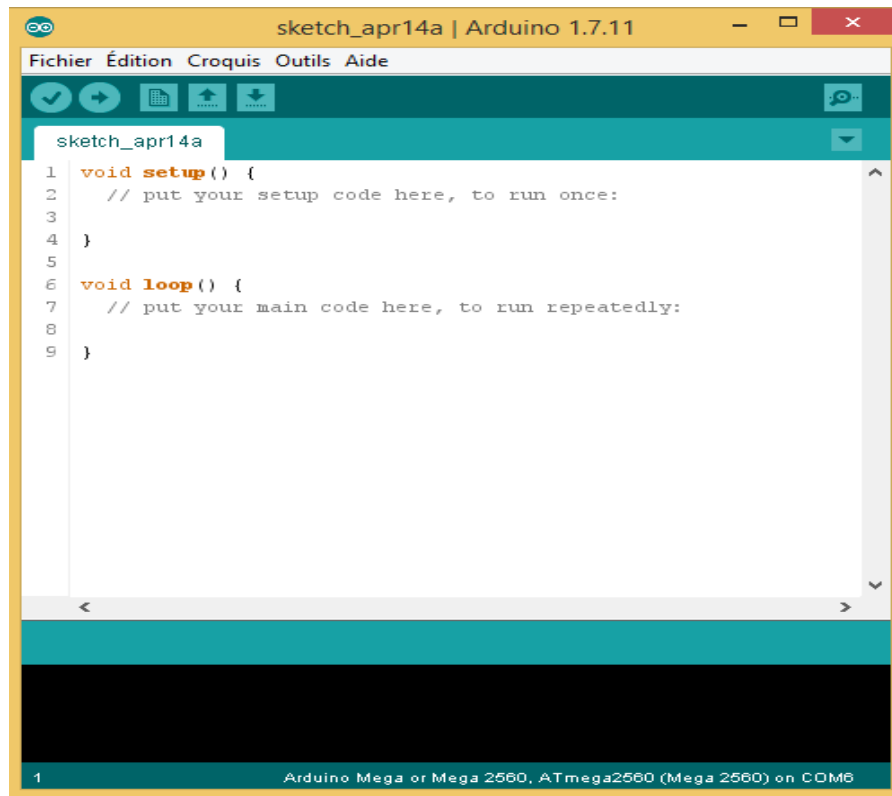
16. *Arduino DUE* est une évolution de l'Arduino Mega2560 avec un micro-contrôleur 32 bits Atmel SAM3X (ARM 32 bits Cortex-M3).

17. *Arduino Esplora* est une carte dérivée de l'Arduino Leonardo. Elle a la forme d'une manette de jeu. Contrairement aux autres Arduino, c'est une carte « tout-en-un » qui comporte de nombreux capteurs (température, accélération, lumière, microphone, potentiomètre...) ainsi que 4 boutons poussoirs, un potentiomètre et un joystick analogue.

### **Annexe I.3 : Le logiciel Arduino**

Le logiciel est un environnement de développement intégré (IDE) écrit en langage Java, totalement gratuit et facilement téléchargeable sur le site officiel de l'Arduino à savoir [www.arduino.cc](http://www.arduino.cc). Le logiciel étant multiplateforme, télécharger la version correspondant à votre système d'exploitation.

Une fois, le logiciel et ses pilotes, correctement installés, on peut donc programmer nos cartes Arduino. L'interface utilisateur du logiciel est présentée ci-dessous :



**Figure 4** : L'interface utilisateur du logiciel Arduino

On peut voir deux fonctions qui se présentent automatiquement dès qu'on ouvre le logiciel : il s'agit de la fonction *setup* ( ) et de la fonction *loop* ( ). En effet, tout programme Arduino, appelé sketch, doit contenir obligatoirement ces deux fonctions spécifiques. Même si, elles sont vides, le programme ne marche pas si au moins l'une d'entre elles est absente.

La fonction *setup* ( ) n'est exécutée qu'une seule fois juste après le lancement du programme. Elle contient généralement les instructions d'initialisation de certaines ressources de la carte telles que, par exemple, définition des lignes des ports parallèles soit en entrées ou en sorties, définition de la vitesse de fonctionnement du port série, etc...

Quant à la fonction *loop* ( ), elle est la boucle principale et se répète indéfiniment tant que l'Arduino restera sous tension. En d'autres termes, suite à un reset au moyen de son poussoir ou suite à une mise sous tension qui a pour effet de provoquer un reset automatique, l'Arduino exécute une seule fois les instructions contenues dans la fonction *setup* ( ) puis exécute ensuite indéfiniment les instructions contenues dans la fonction *loop* ( ) de ce même programme. Ces fonctions ne retournent pas de résultat, elles sont donc déclarées avec le mot clé « *void* ».

Selon le cas, une troisième partie peut ou non être présente dans un programme Arduino mais ne contient pas d'instructions exécutables. Il s'agit de la zone de définition de constantes au moyen du mot clé « *define* » ou de « *const* », d'inclusion d'éventuelles bibliothèques utilisés par le programme au moyen du mot clé « *include* », définition de variables, etc... Cette partie, si elle est présente, se place avant la fonction *setup* ( ), qui elle-même se place avant la fonction *loop* ( ).

Tout comme le langage C, le langage arduino est composé de fonctions arithmétiques et mathématiques, des opérateurs de comparaison et opérateurs logiques, les structures de contrôle (prise de décision, boucles et sauts), des fonctions de gestion du temps et des entrées/sorties et beaucoup d'autres fonctions spécifiques à l'Arduino (gestion du port série, gestion d'interruption, générateur de nombre aléatoire, jouer des notes de musique, etc...).

## **ANNEXE II : Simulation Proteus.**

### **Annexe II : Présentation de proteus**

Proteus 8 est un logiciel professionnel, utilisé dans l'électronique pour simuler des circuits et créer des typons. Il est également capable de simuler le fonctionnement du PIC avec tous les périphériques de la carte de commande. L'utilisation du logiciel «Proteus 8» permet de mieux visualiser le bon déroulement du système ainsi que d'avoir une idée claire sur la partie matérielle et la conception des circuits imprimés. Il nous permet de limiter les essais réels.



**Figure 1** : Programme de Proteus 8 Professional.

# Bibliographie

## Bibliographie et Sites internet :

- [1] **Projet de fin d'étude master 2 En : instrumentation électroniques** Présenté par: bendjafer hicham & midoun anesse Année Universitaire: 2016-2017.
- [2] [www.arrow.com](http://www.arrow.com)
- [3] **Projet de fin d'étude master 2 En : instrumentation électroniques** Présenté par: bendjafer hicham & midoun anesse Année Universitaire: 2016-2017.
- [4] [www.comment ça Marche.net//le standard GSM](http://www.comment ça Marche.net//le standard GSM)
- [5] <https://lfi-docs.blogspot.com/2013/03/lfi2-introduction-au-standard-gsm.html>
- [6] <https://www.researchgate.net/publication/optionisation radio des le reseau GSM>  
Par :Hicham Megnafi,N.Boukli-hacene,A.Barrichi.
- [7][www.gsm.domotique .com](http://www.gsm.domotique .com).
- [8] [www.factory forword.com](http://www.factory forword.com).
- [9]<https://www.studentcompanion.net/fr/interfacer-un-modem-gsmgprs-avec-un-microcontrôleur-pic/>
- [10] **Projet de fin d'étude pour l'obtention du Diplôme de MASTER2 En INSTRUMENTATION ELECTRONIQUE- Système de contrôle par GSM- Année Universitaire: 2016-2017**
- [11] [www.instructables.com//HOW TO USE DHT-SENSOR](http://www.instructables.com//HOW TO USE DHT-SENSOR).
- [12] **Projet de fin d'étude pour l'obtention du Diplôme de MASTER2 En INSTRUMENTATION ELECTRONIQUE- Système de contrôle par GSM- Année Universitaire: 2016-2017**
- [13] [www.arduino-info-ikispaces.com//lcd-blue-i2c.pdf](http://www.arduino-info-ikispaces.com//lcd-blue-i2c.pdf)
- [14] le grand livre d'arduino-2eme edition-2013  
<https://github.com> pour télécharger les bibliothèques ARDUINO:  
Arduino\_dht : "DHT.h"  
Arduino\_adafruit: <Adafruit\_Sensor.h>.  
Arduino \_ LiquidCrystal\_I2C :<LiquidCrystal\_I2C.h>  
<https://projectiot123.com> pour télécharger les bibliothèques porteuse :
- Bibliothèque arduino.**  
**Bibliothèque GSM.**  
**Bibliothèque DHT.**