

**MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU**



**FACULTE DE GENIE ELECTRIQUE ET DE L'INFORMATIQUE  
DEPARTEMENT D'ELECTRONIQUE**

**Mémoire**

**MASTER PROFESSIONNEL**

**Domaine : Sciences et technologies**

**Filière : Génie électrique**

**Spécialité : Electronique Industrielle**

**Thème**

**Réalisation d'un afficheur LCD  
lumineux défilant commandé par  
SMS à base de la carte ARDUINO  
UNO**

**Proposé et dirigé par :**

**Mr. LAZRI**

**Présentée par :**

**Mr. KEDIR RAFIK  
Mr. AKLI MOHAND**

**Année universitaire: 2015/2016**

## **Remerciement**

*Nous remercions tout d'abord par excellence sa grandeur « le bon dieu » qui nous a donné le courage et la patience tout au long de nous vie.*

*Nous remercions vont à mes chères parents, que Dieu les protège et leurs procure une longue vie.*

*Nous remercions notre promoteur Mr. LAZRI pour son aide, son orientation et ses conseils durant l'accomplissement du projet.*

*Nous tenons à remercier également Mr. Achour Pour son aide très précieuse et son soutien tout au long de notre travail.*

*Nous remercions également les membres du jury d'avoir accepté de juger ce modeste travail.*

# *Dédicaces*

*On dédie ce travail à nos parents qui ont eu foi en nous et qui ont su être là pour nous soutenir et nous encourager durant le long de nos études.*

*A tout les membres de nos familles ainsi qu'à tout nos amis(e).*

# Sommaire

## Introduction1

### Chapitre I : les afficheurs

Préambule .....	3
I.1. Les diodes électroluminescentes .....	3
I.1.1. Description le terme LED.....	3
I.1.2. Les Différents types de LED .....	4
I.1.3. Calcul de la résistance chutrice à associer avec une diode LED .....	7
I.1.4.L'utilisation des LED .....	8
I.1.5. Journal lumineux à LED .....	9
I.1.6.Affichage par matrice à LEDs .....	9
a) Structure d'un afficheur à matrice .....	11
I.2. La technologie des afficheurs.....	12
I.2.1. Définition des afficheurs a sept segments.....	12
I.2.2.les Caractères représentés.....	14
I.2.3.Désignation et commande des segments.....	14
I.3. Définition d'un afficheur LCD (Liquid Crystal Display).....	16
I.3.1.Présentation de l'afficheur LCD.....	18
I.3.2. Principe des cristaux liquides .....	19
I.3.3. Brochage de l'afficheur LCD .....	19
I.3.4. Fonctionnement.....	20

a) mode 8 bits.....	21
b) mode 4 bits.....	21
I.3.5.La mémoire de LCD .....	21
a) La mémoire d’affichage (DD RAM) .....	21
b) La mémoire du générateur de caractères (CG RAM).....	22
I.3.6. Command d’un afficheur LCD.....	22
a) Mode 8 bits .....	22
b) Mode 4 bits .....	23
I.3.7. Initialisation .....	24
a) en mode 8 bits .....	24
b) en mode 4 bits .....	24
I.3.8.L’éclairage .....	25
a) Eclairage transmissif .....	25
b) Eclairage réfléchif.....	25
c) Eclairage transflectif.....	25

## **Chapitre II : Présentation de la carte Arduino.**

Préambule .....	27
II.1. Les origines d’Arduino .....	27
II.2. Définition du module Arduino .....	28
II.3. Les gammes de la carte Arduino .....	29
II.4.1. Pourquoi choisi Arduino UNO.....	30

II.4.2. La constitution de la carte Arduino UNO.....	32
II.4.3. Alimentation de la carte Arduino UNO.....	33
II. 4.4. Protection du port USB contre la surcharge en intensité.....	35
II. 4. 5. Gestion des mémoires dans la carte Arduino.....	35
II.4. 6. Les entrées / sorties numériques.....	35
II.4. 6. Les entrées analogiques .....	36
II.4. 7. Arduino et la communication avec l'extérieur.....	37
II.4. 8. Caractéristique mécanique .....	38
II.4 .9. Programmation .....	39
II .4. 10. Réinitialisation « logicielle » automatique .....	39
II.5. Définition d'un microcontrôleur ATmega328 .....	41
II.6. Les principales caractéristiques de l'ATmega328.....	43
II.7. Comparateur Analogique (AC).....	45
II.8. Convertisseur Analogique / Numérique (ADC) .....	45
II.9. Conception et logiciel .....	45
II.9.1. Choix de langage .....	45
II.9.2. Le choix de l'IDE .....	46

## **Chapitre III : la réalisation**

Préambule .....	48
III.1.Principe de fonctionnement .....	48
III-Description des différents blocs de système .....	48
III .1.bloc émetteur .....	49
III.1.1. téléphone mobile .....	49
III.2. bloc de transmission .....	49
III.2.1. le module GSM .....	49
a) Définition en générale .....	49
III.2.2.Spécifications .....	49
III.2.3. le service mini messagerie SMS .....	50
III.3.La plaquette d'essai .....	51
II.4.Les différents branchements des différents blocs du système .....	52
III.4.1. branchement de la carte Arduino avec l'afficheur LCD .....	52
III.4.2. branchement des différents blocs du système.....	53
Conclusion .....	54
Programme.....	55
Bibliographie	

## Listes des figures

<b>Figure 1</b> : différents types de LED .....	4
<b>Figure 2</b> : LED bicolores .....	4
<b>Figure 3</b> : LED à plus de deux couleurs.....	5
<b>Figure 4</b> : LED haute luminosité .....	6
<b>Figure 5</b> : symbole de la LED.....	6
<b>Figure 6</b> : le sens de la LED .....	7
<b>Figure 7</b> : montage LED dans un circuit électrique.....	7
<b>Figure 8</b> : LED en série .....	8
<b>Figure 9</b> : luminaire à LED.....	9
<b>Figure 10</b> : afficheur selon une forme matrice avec des LED .....	10
<b>Figure 11</b> : Structure d'un afficheur à matrice .....	11
<b>Figure 12</b> : afficheur de sept segments .....	12
<b>Figure 13</b> : afficheur de sept segments dans son boîtier .....	13
<b>Figure 14</b> : schéma montrant la désignation de chacun des segments .....	15
<b>Figure 15</b> : Le circuit interne d'un afficheur de sept segments.....	15
<b>Figure 16</b> : commande d'un afficheur de sept segments par un décodeur....	16
<b>Figure 17</b> : afficheur LCD .....	17
<b>Figure 18</b> : brochage de l'afficheur LCD .....	19
<b>Figure 19</b> : la carte ARDUINO UNO .....	33

<b>Figure 20</b> : brochage de carte ARDUINO .....	37
<b>Figure 21</b> : Schéma électrique de carte ARDUINO UNO.....	41
<b>Figure 22</b> : microcontrôleur AT mega 328 .....	43
<b>Figure 23</b> : Les différents blocs du système.....	48
<b>Figure 24</b> : brochage Shield GSM M10.....	50
<b>Figure 25</b> : la plaquette d'essai sans soudeur .....	51
<b>Figure 26</b> : branchement de la carte ARDUINO avec l'afficheur LCD.....	52
<b>Figure 27</b> : schéma global de fonctionnement.....	53

---



**Listes de tableaux**

**Tableau 1:** description des broches de l’afficheur LCD ..... 20



## Introduction

Les afficheurs à cristaux liquides, autrement appelés afficheurs LCD (Liquid Crystal Display), sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), sont relativement bons marchés et s'utilisent avec beaucoup de facilité. Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leurs dimensions, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service. Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).

Dans notre réalisation, l'afficheur LCD lumineux, permet l'affichage d'un message envoyé par un téléphone portable.

Notre application est réalisée à base de la carte Arduino avec un module GSM, un afficheur LCD, l'afficheur donne aux critères affichés un effet (défilement, à droite, et à gauche).

Pour mener à bien notre travail, nous avons structuré notre mémoire en trois chapitres et une conclusion

Dans le premier chapitre, nous allons donner des généralités sur les afficheurs.

Dans le deuxième chapitre, nous allons décrire d'une façon détaillée la carte Arduino UNO.

Dans le troisième chapitre, nous procéderons à la réalisation pratique de l'afficheur LCD où nous avons détaillé chaque bloc du système, Nous terminerons par une conclusion en donnant des perspectives.

# **Chapitre I : généralité sur les afficheurs**

---

## **Chapitre I : généralité sur les afficheurs**

# Chapitre I : généralité sur les afficheurs

---

## **Préambule :**

Les afficheurs LCD sont devenus incontournables dans toutes applications qui demandent la visualisation.

Auparavant onéreux et difficiles à mettre en œuvre, ils sont maintenant bon marché, Les afficheurs LCD se ressemblent tous, à part le nombre de lignes et le nombre de colonnes, le fonctionnement et le brochage est standard et identique, Un des points intéressants est de pouvoir contrôler l'afficheur en mode 8bits ou en mode 4bits.

Dans ce chapitre, nous donnons des généralités sur les afficheurs. Pour ce faire, dans un premier temps, nous présentons les différents types d'afficheurs, ensuite nous expliquons la technologie des afficheurs. Dans un second temps, nous expliquons le principe de fonctionnement de l'afficheur LCD.

## **I.1. Les diodes électroluminescentes :**

### **I.1.1. Description le terme LED :**

Le terme LED est :

LED= Light Emitting Diode, diode électroluminescente

DEL= Diode Electroluminescent

HB-LED ou HBLED = High Brightness LED, LEDs haute luminosité

HP-LED ou HPLED= High power LED, LEDs haute puissance

La LED est un composant dit passif, de la famille des semi-conducteurs (comme la diode et le transistor) il s'agit d'une diode un peu particulière, qui a la propriété d'émettre de la lumière quand un courant la parcourt (de l'anode vers la cathode), et les couleurs que l'on trouve généralement sont : rouge, vert, jaune et bleu.

# Chapitre I : généralité sur les afficheurs

---

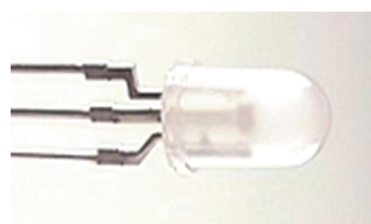


**Figure 1** : différents types de LED

## I.1.2. Les Différents types de LED :

- LED bicolores :

Ils sont associées dans un même boîtier transparent, 2 puces émettant chacune une couleur précise : rouge pour l'une, et vert pour l'autre. Les LED bicolores peuvent posséder deux pattes (montage des deux LED en parallèle retourné) ou trois pattes (montage des deux LED en série inversé avec point milieu).



**Figure 2** : LED bicolores

L'allumage simultané des deux puces vert et rouge (résultat jaune-orange) est plus facile avec la LED à trois pattes, sous une simple tension continue.

Pour la LED à deux pattes, il y a obligation d'inverser la polarité appliquée, selon la puce que l'on veut utiliser. Une tension alternative est donc requise pour allumer les deux puces en même temps (il est faux de dire en même temps, mais grâce à la persistance rétinienne, cet

# Chapitre I : généralité sur les afficheurs

---

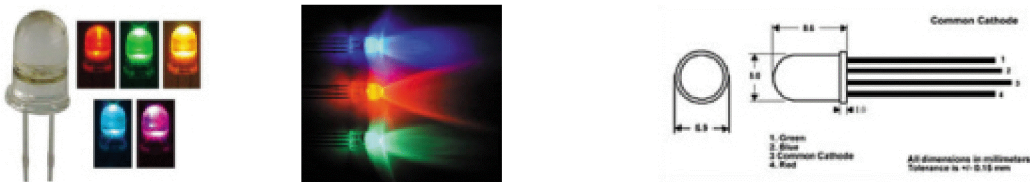
Effet « continu » peut être obtenu avec un signal alternatif de fréquence suffisant, sans scintillement visible).

- LED à plus de deux couleurs

Ils sont associées dans un même boîtier transparent, 3 ou 4 puces émettant chacune une couleur précise : rouge, vert, et bleu.

Du fait d'un éclairage généralement moindre de la puce bleue, cette dernière se trouve souvent doublée : 2 puces bleues pour une rouge et une verte.

Les deux puces bleues étant électriquement raccordées entre elles, la LED possède en tout quatre pattes : une pour la puce rouge, une pour la puce verte, une pour la ou les puces bleues, et la quatrième enfin pour le point commun.



**Figure 3** : LED à plus de deux couleurs

Si l'on se contente d'activer en tout ou rien chacune des trois couleurs, nous pouvons faire s'illuminer la LED avec 7 couleurs différentes (8 combinaisons possibles, mais on ne compte pas celles où tout est éteint).

Pour obtenir plus de variations possibles, plus de couleurs si on préfère, il faut rendre variable l'intensité lumineuse de chaque couleur, de façon indépendante.

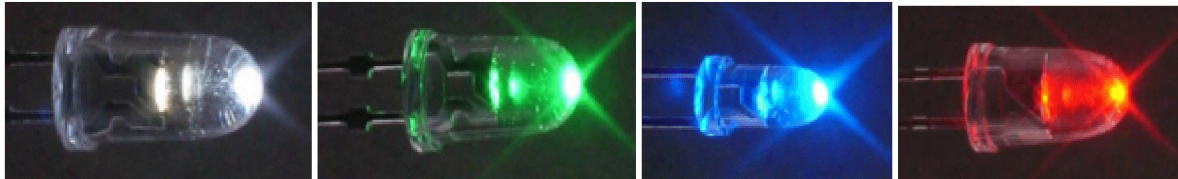
- LED haute luminosité

Ce type de LED, qui peut émettre dans l'ensemble des couleurs vues précédemment (les LED haute luminosité ne sont pas forcément blanches) constituera sans doute dans un très proche

## Chapitre I : généralité sur les afficheurs

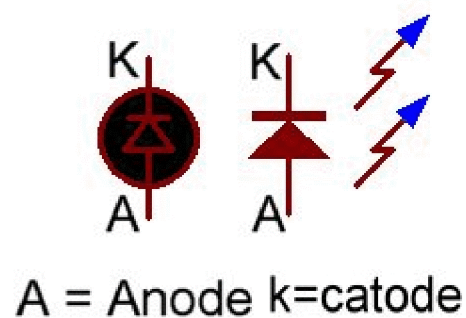
---

Avenir, une solution alternative intéressante à l'éclairage grand public, voir à l'éclairage professionnel de forte puissance (ca commence à venir, il suffit de regarder un peu autour de soi quand on va dans certains petits spectacles ou dans des salons).



**Figure 4 :** LED haute luminosité

Il est à noter que vu la puissance lumineuse développée par ces LED (certaines atteignent ou dépassent 50000 mcd), les mêmes précautions que celle qui s'appliquent pour les LASER sont de rigueur (ne pas diriger directement ces LED vers les yeux).

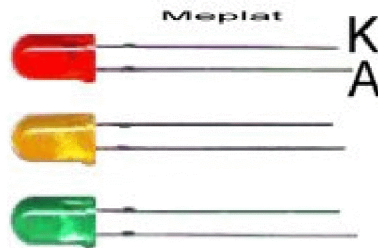


**Figure 5 :** symbole de la LED

# Chapitre I : généralité sur les afficheurs

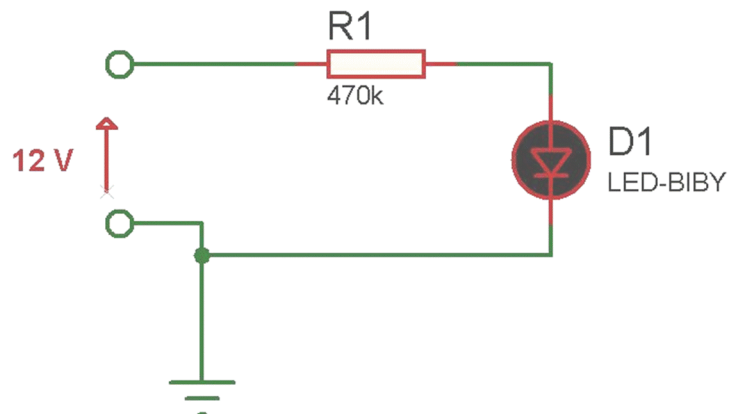
---

Le sens des LED est le même que celui des diodes ordinaires. Généralement, une rainure au bas du boîtier plastique sur la périphérie permet de connaître le sens de polarisation de la diode (la cathode est située du côté de la broche la plus courte).



**Figure 6 :** le sens de la LED

## I.1.3. Calcul de la résistance chutrice à associer avec une diode LED



**Figure 7 :** montage LED dans un circuit électrique

Supposons une tension d'alimentation de 12 volts (figure7).

Soit que pour qu'une diode LED s'éclaire normalement elle doit être traversée par un courant de 20 mA sous une tension de 1,8 volt (dépend de type de diode utilisée).

# Chapitre I : généralité sur les afficheurs

La tension ou borne de R1 doit être de :

$$12V - 1,8V = 10,2V$$

Selon la loi d'ohm  $R = u/I$

$$R = 10,2V / 0,02A = 510 \text{ ohms}$$

Bien qu'on doit utiliser la valeur normalisées, on doit donc placer une résistance de 470 ohms en série avec la diode LED pour que celle – ci puisse fonctionner correctement sous une tension d'alimentation de 12 volts.

Un autre paramètre important auquel il faut tenir compte est la puissance.

$$P = U * I$$

$$10,2 \text{ V} * 0,02 \text{ A} = 0,204 \text{ watts}$$

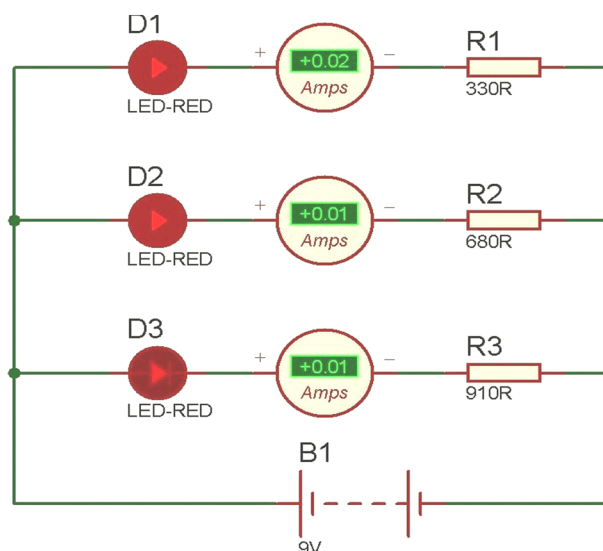
On devra donc utiliser une résistance capable de dissiper cette puissance.

## I.1.4.L'utilisation des LED

L'avantage d'utilisation des LED est qu'elles ne s'usent pas, elles sont moins chères que des voyants, elles consomment moins d'énergie.

Mes l'inconvénient et qu'elles ne peuvent fonctionner qu'avec une faible tension, et qu'elles n'éclairent pas beaucoup par rapport aux ampoules classiques.

Il faut donc ajouter une résistance en série pour utiliser une LED.



**Figure 8 : LED en série**

## Chapitre I : généralité sur les afficheurs

---

En fonction de la résistance l'intensité va varier et la luminosité aussi.

### I.1.5. Journal lumineux à LED :

Le journal lumineux dépend naturellement de la densité des points par unité de surface et la surface totale de la matrice est elle-même liée à la distance d'où elle doit être lue. Ainsi, pour un affichage sur un panneau géant, ces matrices peuvent avoir des dimensions assez grandes pour afficher un ou quelques caractères.

On peut aussi réaliser de nombreux effets lumineux qui se succèdent automatiquement comme par exemples :

- Défilement de gauche à droite ou de haut vers le bas.
- Affichage des messages ou des signes.

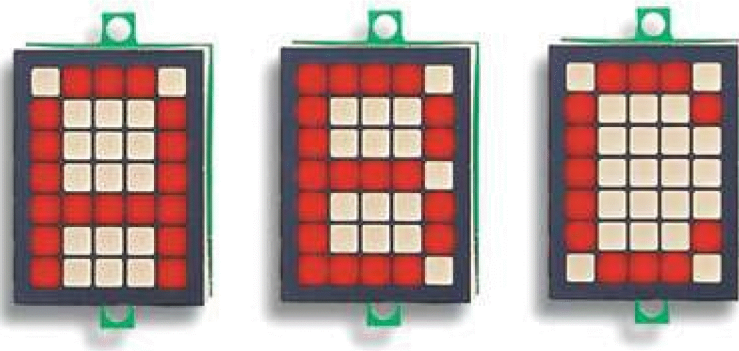


Figure 9 : affichage lumineux à LED

### I.1.6. Affichage par matrice à LED :

L'affichage est réalisé en allumant en même temps ou à des instants différents et périodiques les points lumineux nécessaires du tableau pour communiquer l'information désirée, en ce sens le tableau d'affichage est constitué de points lumineux couramment appelés pixels (c'est une abréviation anglaise de « Picture élément »).

La taille du pixel dépend de la distance de laquelle l'affichage est prévu d'être lu. La distance entre les pixels est imposée par la résolution désirée des caractères à afficher et par les dimensions du tableau.

# Chapitre I : généralité sur les afficheurs

En consultant le tableau d'affichage lumineux, on constate que les pixels sont définis par :

La longueur géométrique qui est le choix des diodes LED et l'espacement entre elles.

La grandeur photométrique qui est le choix des composants électriques qui commandent et alimentent les diodes.

Ce qui rend l'affichage une séquence de données logiques définissant l'extinction de certains pixels et l'allumage d'autres, de ce fait l'utilisation d'un ordinateur est indispensable et suivant les tâches à accomplir et la diversité des types d'affichage, on utilise un microcontrôleur qui gère les commandes nécessaires

Le tableau d'affichage est considéré comme étant une grille de diodes hautes luminosité organisées sur X lignes et Y colonnes et qui a une capacité d'affichage définie par le nombre de caractères qu'on peut afficher sur ce tableau.

## Exemple :

Un type d'afficheur très utilisé emploie 37 LED en forme de points disposés selon une matrice à 7 lignes et 5 colonnes plus 2 points décimaux (DP1 et DP2), l'un situé à droite et l'autre à gauche.

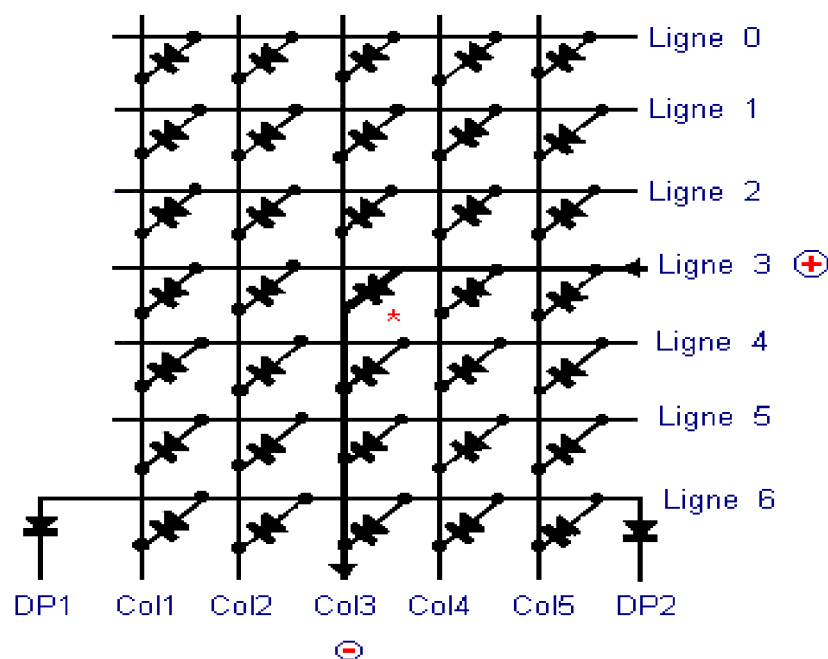


Figure 10 : afficheur selon une forme matrice avec des LED

## Chapitre I : généralité sur les afficheurs

---

### a) Structure d'un afficheur à matrice :

Comme pour l'afficheur à 7 segments en allumant les LED appropriées, il est possible d'obtenir les 10 chiffres décimaux ; de plus, grâce au nombre de LED plus important, il est possible de représenter d'autres caractères parmi lesquels toutes les lettres de l'alphabet et différents signes comme +, -, /, (,) et d'autres encore.

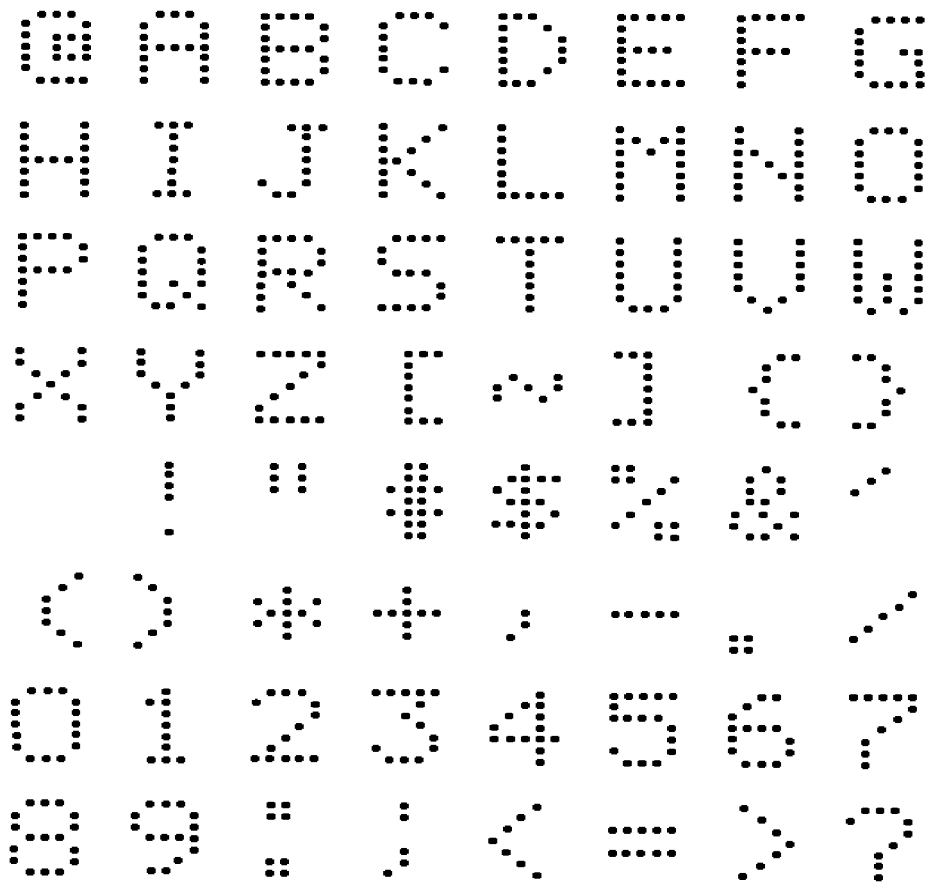


Figure 11 : Structure d'un afficheur à matrice

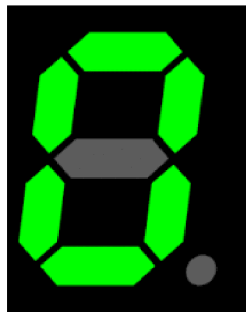
## I.2. La technologie des afficheurs :

### I.2.1. Définition des afficheurs a sept segments :

Les afficheurs 7 segments sont des afficheurs à LED constitués de 7 segments (une LED par segment), qui permettent d'afficher des chiffres et même parfois des lettres en fonction de l'application.

Il y a encore quelques années, les afficheurs 7 segments étaient partout : radio réveil, montre, calculatrice, micro-onde, four, minuteur, systèmes industriels, outil de mesure, etc. Aujourd'hui, les afficheurs 7 segments ont pour la plupart été remplacés par des afficheurs LCD ou des écrans graphiques plus haut de gamme. On ne trouve des afficheurs 7 segments que dans des cas d'usage très particuliers : voltmètre pour modélisme, carte électronique DIY ou pour un marché de niche, outils de mesure low cost, etc.

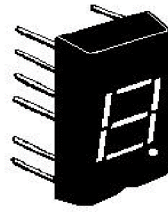
Les afficheurs 7 segments ont été oubliés, car trop sobres, les temps changent, les goûts aussi. Mais comme il s'agit simplement d'une série de LED dans un même boîtier, au final, c'est technologie très simple à mettre en œuvre. Et comme on dit, souvent, la solution la plus simple est la meilleure.



**Figure 12:** afficheur de sept segments

## Chapitre I : généralité sur les afficheurs

---



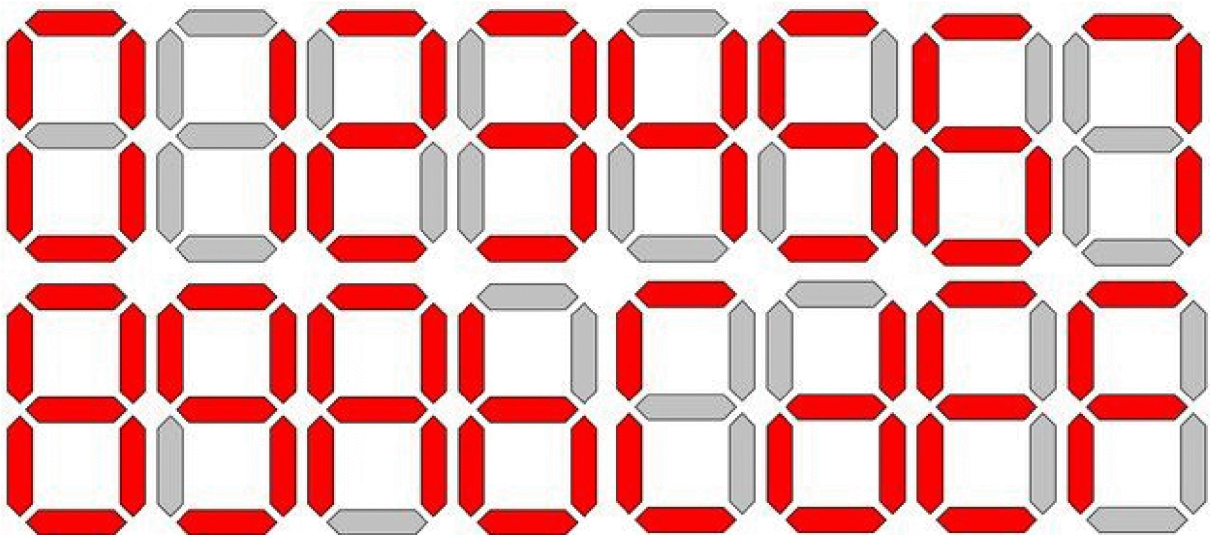
**Figure 13** : afficheur de sept segments dans son boîtier

Les afficheurs 7 segments sont constitués de 7 segments, d'où leur nom. Ces segments sont nommés A, B, C, D, E et F par convention, et ils se présentent dans l'ordre illustré ci-dessus.

Chaque segment correspond à une LED qu'il est possible d'allumer ou d'éteindre pour former des chiffres, des lettres et même des caractères spéciaux rudimentaires. En général, les afficheurs disposent de 7 segments et d'un "point décimal" qui peut être utilisé pour afficher des nombres à virgule ou des sous-unités (dixième de seconde par exemple).

Il existe une multitude de couleurs d'afficheurs 7 segments : rouge, vert, jaune, orange, bleu, blanc, etc. Il existe une multitude de tailles, du petit afficheur de quelques millimètres de côté à plusieurs dizaines de centimètres. À choisir.

Au final, qu'importent la couleur et la taille de l'afficheur, le principe de fonctionnement est le même.



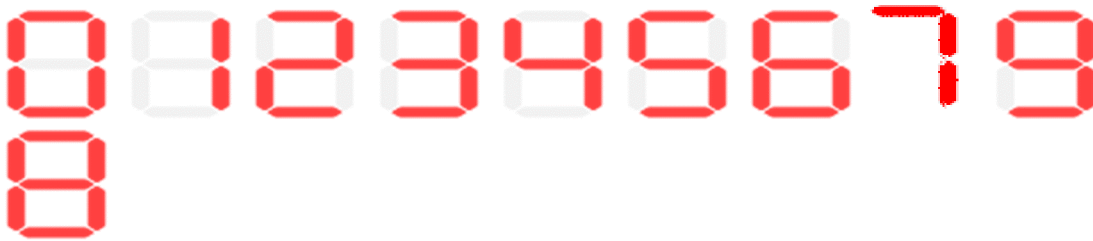
Chiffres de 0 à 9 et lettres de A à F

# Chapitre I : généralité sur les afficheurs

---

## I.2.2.les Caractères représentés :

- Les 10 chiffres représentés avec l’affichage à sept segments



- Le complément hexadécimal



En commandant convenablement l’allumage de certains segments, on visualise les nombre désirés.

Pour visualiser un 1, on allumera les segments b, c et pour un 2, les segments a, b, g, e, d par exemple.

## I.2.3.Désignation et commande des segments :

Dans un afficheur 7 segments, les segments sont généralement désignés par les lettres allant de A à G. Dans le cas où l’afficheur comporte un point, servant de séparateur décimal, celui-ci est désigné DP (de l’anglais décimal point) ; certains parlent dans ce cas d’un afficheur « 8 segments ».

Dans le cas d’afficheurs à DEL, deux cas de figures sont présents :

- Afficheur à anode commune : toutes les anodes sont reliées et connectées au potentiel haut.

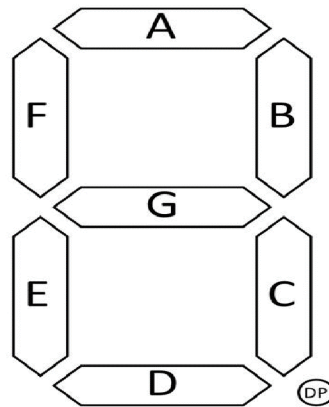
# Chapitre I : généralité sur les afficheurs

---

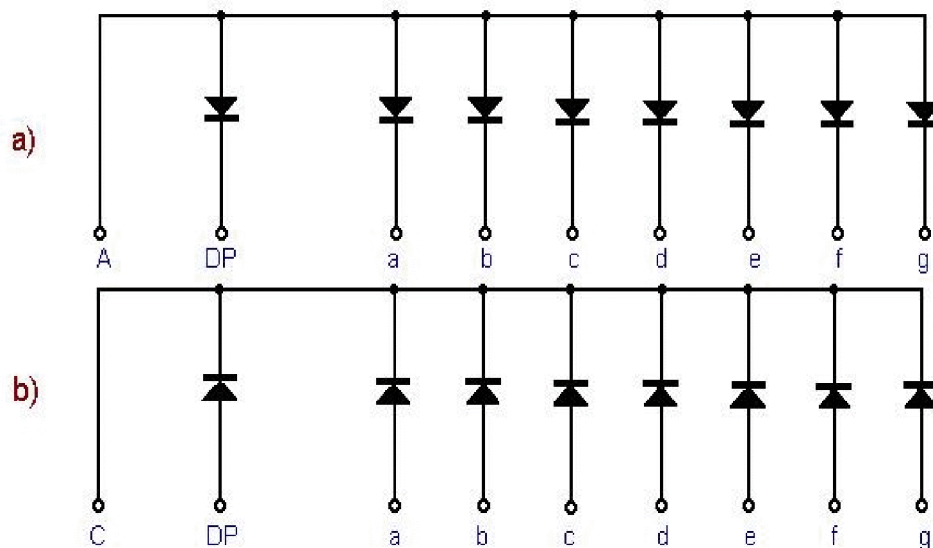
La commande du segment se fait par sa cathode mise au potentiel bas.

- Afficheur à cathode commune : toutes les cathodes sont reliées et connectées au potentiel bas.

La commande du segment se fait par son anode mise au potentiel haut.



**Figure 14 :** schéma montrant la désignation de chacun des segments

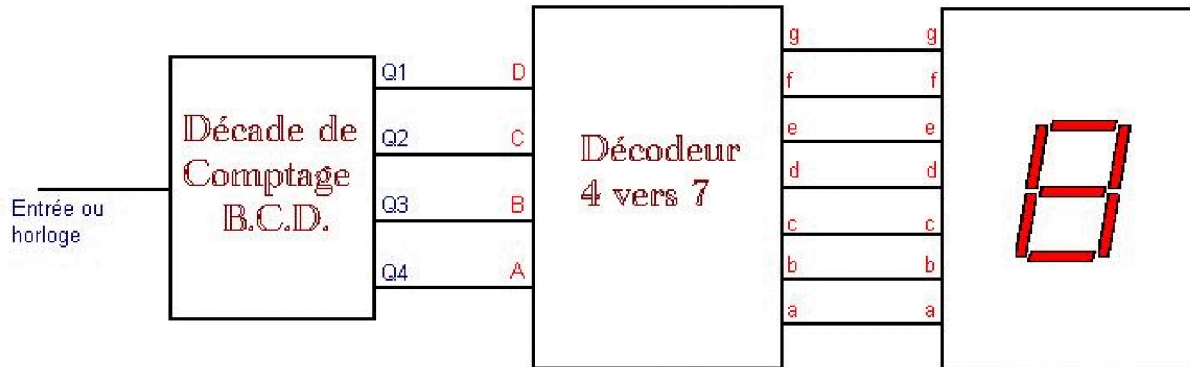


**Figure 15 :** Le circuit interne d'un afficheur de sept segments

# Chapitre I : généralité sur les afficheurs

---

Pour commander un afficheur 7 segment on utilise un décodeur (driver).



**Figure 16** : un afficheur sept segments commandé par un décodeur

On appelle décodeur 7 segments le système permettant de passer des mots d'entrée codé binaire 4bits aux mots de sortie codé 7 segments (dans notre cas l'affichage se fera de 0 à 9). L'interprétation visuelle de l'affichage du chiffre est formée par l'allumage des segments d'un afficheur.

Il existe une autre technologie qui s'appelle LCD de l'anglais « Liquid Crystal Display » le sigle signifie Afficheur ou Ecran à Cristaux Liquides.

Mais ces afficheurs contrairement aux LED n'émettent pas de lumière : ils ont besoin d'une source de lumière.

### **I.3. Définition d'un afficheur LCD (Liquid Crystal Display) :**

Les afficheurs à cristaux liquide ou L, C, D comprennent des segments comme les afficheurs à L, E, D.

Ils sont des modules compacts intelligents qui n nécessitent peu de composants externes pour un bon fonctionnement. Ils sont utilisés avec beaucoup de facilité.

Les afficheurs LCD sont constitués d'une couche de cristal liquide comprise entre deux plaques de verre revêtues de matériaux conducteurs.

L'alignement des molécules du cristal peut être changé par application d'une tension aux bornes de ses deux plaques.

## Chapitre I : généralité sur les afficheurs

---

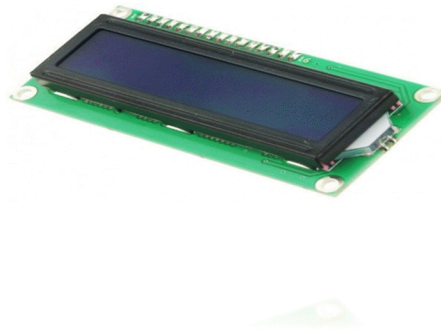
Ce changement d'alignement permet de visualiser des segments comparables à ceux d'un afficheur à LED à ceci après qu'ils n'émettent aucune lumière.

Un exceptionnel microprocesseur « pilot » de la famille CMOS diminue considérablement leur consommation. Ils sont pratiquement les seuls à être utilisés sur les appareils alimentés par pile.

Plusieurs afficheurs sont disponibles sur le marché et ne diffèrent les uns des autres, non seulement par leurs dimensions (de 1 à 4 lignes, de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leurs tensions de service.

Le composant PCF8574 dispose d'un port 8 bits. L'afficheur LCD peut donc être piloté à l'aide de ce composant. Cependant, le mode 4 bits sera requis pour le pilotage de l'afficheur et du back light.

L'utilisation des signaux de commandes RS, R/W et E, oblige à l'utilisation du mode 4 bits. Le back light est également commandé par I2C.



**Figure 17** : un afficheur LCD

Plusieurs afficheurs sont disponibles sur le marché et diffèrent les uns des autres, non seulement par leur dimension, (de 1 à 4 lignes de 6 à 80 caractères), mais aussi par leurs caractéristiques techniques et leur tension de service.

Certains sont dotés d'un rétro éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA).

Ils sont très utilisés dans le montage à microcontrôleur, et permettent une grande convivialité.

# Chapitre I : généralité sur les afficheurs

---

Ils peuvent aussi être utilisés lors de la phase de développement d'un programme, car on peut facilement afficher les valeurs de différentes variables.

## I.3.1. Présentation de l'afficheur LCD :

Le type d'afficheur étudié permet d'afficher :

- tous les caractères ASCII internationaux (codes 20h à 7Dh ou 7Eh) : chiffres et lettres, caractères de ponctuations, opérateurs arithmétiques, etc.
- les caractères nationaux (à partir des codes 80h ou A0h). Le jeu de caractère dépend de l'aversion du contrôleur
- des caractères définis par l'utilisateur (jusqu'à 8 caractères définis par l'utilisateur)

L'afficheur permet aussi de placer un curseur ou de faire clignoter le caractère à l'endroit où le prochain caractère doit être affiché. On peut réaliser des animations en décalant la fenêtre d'affichage.

L'afficheur est constitué de :

- un CI contrôleur Hitachi HD44780 ou équivalent
- un ou plusieurs CI unité d'extension (le contrôleur seul ne permet que de piloter 2 lignes de 8 caractères) :
  - un CI pour 2 lignes de 16 caractères
  - Plusieurs CI pour 4 lignes de 20 caractères
  - une matrice LCD de 2 lignes de 16 caractères ou de 4 lignes de 40 caractères

Le module dispose de :

- signaux de liaison avec le  $\mu\text{C}$  ou le système à  $\mu\text{P}$  de commande : DB7-DB0, EN, R/W, RS
- 2 broches pour l'alimentation VDD et VSS ( $VDD - VSS = 5\text{V}$ )
- Une broche pour régler de contraste de l'affichage (V0)
- Parfois deux broches pour l'alimentation de LED pour le rétro-éclairage

# Chapitre I : généralité sur les afficheurs

## I.3.2. Principe des cristaux liquides :

L'afficheur est constitué de deux lames de verre, distance de 20  $\mu\text{m}$  environ, lesquelles sont dessinées les mantisses formant les caractères.

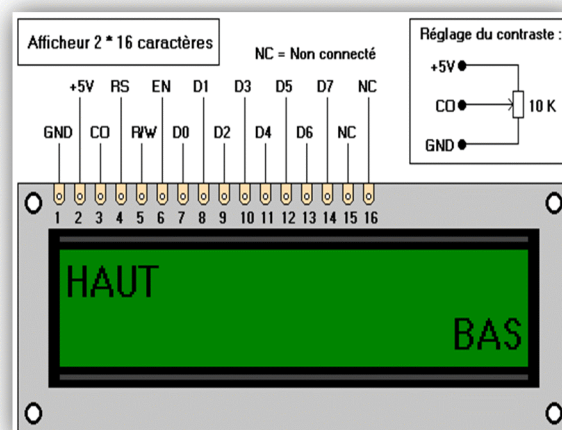
L'espace entre elles est rempli de cristal liquide normalement réfléchissant (pour les modèles réfléchissants). L'application entre les deux faces d'une tension alternative basse fréquence de quelques volts (3 à 5 V) le rend absorbant.

Les caractères apparaissent sombres sur fond clair, n'émettant pas de lumière, un afficheur à cristaux liquides réfléchissant ne peut être utilisé qu'avec un bon éclairage ambiant.

Sa lisibilité augmente avec l'éclairage. Les modèles transmissifs fonctionnent différemment : normalement opaque au repos, le cristal liquide devient transparent lorsqu'il est excité ; pour rendre un tel afficheur lisible, il est nécessaire de l'éclairer par l'arrière, comme c'est le cas pour les modèles rétro éclairé.

## I.3.3. Brochage de l'afficheur LCD :

Un circuit intégré spécialisé est chargé de la gestion du module, il remplit une double fonction, d'une part il commande l'affichage et de l'autre se charge de la communication avec l'extérieur. L'afficheur LCD a 14 broches en standard et souvent 16, les broches 15 et 16 servent au rétro-éclairage (une option).



**Figure18** : brochage de L'afficheur LCD

# Chapitre I : généralité sur les afficheurs

---

**Tableau 1: description des broches de l’afficheur LCD :**

Broche	nom	Description
1	Vss	Masse
2	Vdd	Alimentation 5v
3	CO	Variation de 0 à 5v permet de modifier le contraste de l’afficheur.
4	RS	Indique une commande ou une donnée à afficher (0 : commande/1 : Donnée)
5	R/W	Indique une écriture ou une lecture (0 : Ecriture/1 : lecture)
6	E	Indique une validation(le niveau Haut doit être maintenu 500s)
7	D0	Bus de donnée bidirectionnel
8	D1	
9	D2	
10	D3	
11	D4	
12	D5	
13	D6	
14	D7	
15	A	Anode retro éclairage (+5v)
16	k	Cathode retro éclairage (masse)

Les broches 15 et 16 ne sont présentées que sur les afficheurs LCD avec retro éclairage.

Les connexions à réaliser sont simples puisque l’afficheur LCD dispose de peu de broches, il faut évidemment, l’alimenter, le connecter à un bus de donnée (4 ou 8 bits), et connecter les broches E, R/W et RS

## **I.3.4 Fonctionnement :**

On envoie deux types d'informations à l'afficheur :

- les commandes qui permettent de l'initialiser : positionnement du curseur, effacement écran, etc. ;
- Les données à afficher

# Chapitre I : généralité sur les afficheurs

---

L'entrée RS permet de spécifier si on envoie une commande ou une donnée :

- RS=0 : instruction (commande).
- RS=1 : caractère (donnée).

L'afficheur dispose d'une entrée R/W pour spécifier une lecture ou une écriture :

- R/W=0 : écriture vers l'afficheur.
- R/W=1 : lecture de l'afficheur.

Pour valider tous les échanges sur le bus de données (D7-D0) on utilise l'entrée E de l'afficheur. Un front descendant sur cette entrée valide la donnée. En programmation, il faudra placer un court instant E à l'état haut puis à l'état bas.

Il est possible d'utiliser l'afficheur LCD en mode 8 bits normal ou en mode 4 bits pour économiser les broches de son  $\mu$ Contrôleur par exemple, c'est assez pratique :

## a) Mode 8 bits:

En mode 8 bits on place la donnée ou la commande sur le bus D7 à D0 et on valide avec E

## b) Mode 4 bits:

En mode 4 bits on place déjà les poids forts de la donnée ou la commande sur les bits de D7 à D4 et on valide une première fois avec E. Puis on va mettre le poids faibles sur les bits de D3 à D0 et on valide une seconde fois avec E.

L'envoi ou la lecture d'un octet s'effectue donc en 2 temps dans ce mode.

## I.3.5. La mémoire de LCD :

L'afficheur possède deux types de mémoire, la DD RAM et la CG RAM. La DD RAM est la mémoire d'affichage et la CG RAM est la mémoire du générateur de caractères.

### a) La mémoire d'affichage (DD RAM) :

La DD RAM est la mémoire qui stocke les caractères actuellement affichés à l'écran. Pour un afficheur de 2 lignes de 16 caractères, les adresses sont définies de la façon suivante :

ligne	visible	invisible
haut	00H ..... 0FH	10H ..... 27H
bas	40H ..... 4FH	50H ..... 67H

## Chapitre I : généralité sur les afficheurs

---

L'adresse 00H correspond à la ligne du haut à gauche, 0FH à droite.

L'adresse 40H correspond à la ligne du bas à gauche, 4FH à droite. La zone invisible correspond à la mémoire de l'afficheur (48 caractères).

Lorsqu'un caractère est inscrit à l'adresse 27H, le caractère suivant apparaît à la ligne suivante.

### **b) La mémoire du générateur de caractères (CG RAM) :**

Le générateur de caractère est quelque chose de très utile. Il permet la création d'un maximum de 8 caractères ou symboles 5x7.

Une fois les nouveaux caractères chargés en mémoire, il est possible d'y accéder comme s'il s'agissait de caractères classiques stockés en ROM.

La CG RAM utilise des mots de 8 bits de large, mais seul les 5 bits de poids faible apparaissent sur le LCD.

Ainsi D4 représente le point le plus à gauche et D0 le point le plus à droite.

Par exemple, charger un octet de la CG RAM à 1FH fait apparaître tous les points de cette rangée, charger un octet à 00H éteint tous ce point. Les 8 lignes d'un caractère doivent être chargées dans la CG RAM.

La CG RAM peut être utilisée pour créer des caractères en vidéo inversée, des caractères peut être contournée en utilisant une bibliothèque de 8 symboles résidant le système hôte. Un maximum de 8 caractères peut être affiché à la fois.

La CG RAM peut être rechargée périodiquement en fonction des besoins.

Si un caractère de la CG RAM qui est actuellement sur l'afficheur est changé, alors le changement est immédiatement apparent sur l'afficheur.

### **I.3.6. Command d'un afficheur LCD :**

Deux modes de fonctionnement de l'afficheur sont disponibles, le mode 4 bits et le mode 8 bits, modes que l'on choisira à l'initialisation de l'afficheur (voir plus bas).

#### **a) Mode 8 bits :**

Dans ce mode 8 bits, les données sont envoyées à l'afficheur sur les broches D0 à D7. On place la ligne RS à 0 ou à 1 selon que l'on désire transmettre une commande ou une donnée.

Il faut aussi placer la ligne R/W à 0 pour indiquer à l'afficheur que l'on désire effectuer une écriture.

## Chapitre I : généralité sur les afficheurs

---

Il reste à envoyer une impulsion d'au moins 450 ns sur l'entrée E, pour indiquer que des données valides sont présentes sur les broches D0 à D7.

L'afficheur lira la donnée sur le front descendant de cette entrée si on désire au contraire effectuer une lecture, la procédure est identique, mais on place cette fois la ligne R/W à 1 pour demander une lecture.

Les données seront valides sur les lignes D0 à D7.

### **b) Mode 4 bits :**

Il peut, dans certains cas, être nécessaire de diminuer le nombre de fils utilisés pour commander l'afficheur, comme, par exemple lorsqu'on dispose de très peu de broches d'entrées sorties disponibles sur un microcontrôleur.

Dans ce cas, on peut utiliser le mode quatre bits de l'afficheur LCD. Dans ce mode, seuls les 4 bits de poids fort (D4 à D7) de l'afficheur sont utilisés pour transmettre les données et les lire, les 4 bits de poids faible (D0 à D3) sont connectés à la masse. On a donc besoin, hors alimentation de sept fils pour commander l'afficheur, les données sont alors écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible.

Une impulsion positive d'au moins 450 ns doit être envoyée sur la ligne E pour valider chaque demi-octet ou n bit.

Dans les deux modes, on peut, après chaque action sur l'afficheur, vérifier que celui-ci est en mesure de traiter l'information suivante.

Pour cela, il faut demander une lecture en mode commande, et tester le flag busy BF. Lorsque BF=0, l'afficheur est prêt à recevoir une nouvelle commande ou donnée.

Il se peut qu'on dispose encore de moins de broches disponibles dans l'application envisagée. Dans ce cas, on peut alors relier la ligne R/W à la masse de façon à forcer l'afficheur en écriture. On a alors besoin, hors alimentation de seulement six fils en mode 4 bits, et dix fils en mode 8 bits, pour commander l'afficheur, mais on ne peut alors plus relier l'afficheur.

Ceci n'est pas gênant dans la mesure où on sait ce qu'on a écrit sur l'afficheur, mais on ne peut alors plus relire le flag busy. Il faut alors utiliser des temporisations après chaque écriture sur l'afficheur.

On perd alors un peu en temps d'affichage, mais on gagne une broche d'entrée sortie.

### I.3.7. Initialisation :

#### a) En mode 8 bits :

À la mise sous tension de l'afficheur, la ligne supérieure devrait être totalement sombre, celle du bas complètement claire.

Si tel n'était pas le cas, il faudra régler le contraste de l'afficheur en jouant sur la tension de la broche V0.

Avant de pouvoir utiliser l'afficheur, il faut tout d'abord l'initialiser. Pour cela, la première commande à envoyer, est la commande permettant de définir le mode de dialogue avec l'afficheur (DL), et le nombre de ligne sélectionnées (N), et l'envoyer plusieurs fois de façon à ce que celle-ci soit comprise, que le mode de départ soit quatre ou huit bits. On peut ensuite paramétrer l'afficheur, puis l'effacer.

Voici donc les différentes commandes (RS=0) à envoyer à l'afficheur LCD. Entre chaque valeur, il faut envoyer une impulsion d'au moins 450 ns sur ligne E :

- 33h, 33h, 33h : on force le LCD en mode 8bits (3×h)
- 38h : mode 8bits, 2lignes, caractères 5×7
- 0Ch : affichage en fonction, pas de curseur
- 06h : le curseur se déplacer vers la gauche
- 01h on efface l'afficheur

#### b) En mode 4 bits :

Pour l'initialisation d'un afficheur en mode quatre bits, on commence par forcer celui-ci dans le mode huit bits, puis quand on est sûr que celui-ci est valide, on bascule en mode quatre bits. Comme on ne sait au début de l'initialisation si l'afficheur est positionné en quatre ou huit bits, il est nécessaire d'envoyer la commande de passage en mode huit bits plusieurs fois de façon à ce que celle-ci soit comprise, que le mode de départ soit quatre ou huit bits.

Les données sont écrites ou lues en envoyant séquentiellement les quatre bits de poids fort suivi des quatre bits de poids faible, séparés par une impulsion positive d'au moins 450 ns sur la ligne E.

En résumé, voici sur quatre bits, les commandes (RS=0) à envoyer à l'afficheur LCD. Entre chaque valeur, il faut envoyer une impulsion positive sur la ligne E.

- 0h, 1h, 0h, 0h, 1h : on commence par effacer l'afficheur (01h)
- 3h, 3h, 3h : on force le LCD en mode 8bits (3×h)

## Chapitre I : généralité sur les afficheurs

---

- 2h ; on passe en mode 4 bits (20h)
- 2h, 8h : mode 4bits 2 lignes, caractères 5×7 (28h)
- 0h, Ch. : affichage en fonction, pas de curseur (0Ch)
- 0h, 6h : le curseur se déplace vers la gauche (06h)
- 0h, 1h : on efface l’afficheur

### I.3.8.L’éclairage :

Le mode d’éclairage va permettre d’offrir des fonctionnalités de réglages avancées. Ainsi on aura accès à des champs plus ou moins larges dans certains domaines de réglages. Il existe 3 grands modes de réglages.

#### a) Eclairage transmissif :

L’écran fonctionne avec un rétro éclairage qui est le système le plus classique. Ce mode offre la possibilité de régler la luminosité, mais malheureusement la durée de vie du rétro éclairage est limitée, ainsi que la luminosité. De plus il faut tout le temps régler la luminosité en fonction de l’éclairage ambiant.

#### b) Eclairage réflectif :

L’écran fonctionne avec la réflexion de la lumière extérieure. Ce système est très utilisé pour les montres, les PAD et les calculatrices.

Il offre l’avantage de donner une lumière naturelle adaptée à l’éclairage ambiant, de ce fait on a plus besoin de régler la luminosité. Cependant ce procédé montre ses défaillances en cas d’ambiance d’éclairage faible.

#### c) Eclairage transflectif :

Combinaison des deux systèmes précédents, l’éclairage transflectif fonctionne de la même manière que l’éclairage transmissif lorsque la lumière ambiante est assez forte et utilise l’éclairage réflectif lorsque celle-ci est trop faible.

## **Chapitre II : Présentation de la carte Arduino.**

---

## **Chapitre II : Présentation de la carte Arduino**

## Chapitre II : Présentation de la carte Arduino.

---

### Préambule

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée, pour la réalisation de notre afficheur lumineux, nous avons utilisé une carte d'acquisition de type Arduino afin d'assurer la communication entre les deux parties à savoir, la partie commande (téléphone portable) et les deux sites (maîtres et esclave) afin de pouvoir commander et piloter de système. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Dans ce travail, nous présentons la description de la carte Arduino qui sera utilisée pour notre application. En effet, les différentes parties constituant cette carte seront détaillées, à savoir un microcontrôleur, des ports d'entrée/sorties, une mémoire de type EEPROM. Aussi, le logiciel IDE utilisé pour la programmation sera présenté dans ce chapitre.

### II.1. Les origines d'ARDUINO :

A l'origine, ARDUINO avait été conçu pour aider les étudiants dans leur expérimentation. En 2005, il a été proposé à la vente par Massimo benzi et David cuartielles. En quelques années, il est devenu très populaire auprès des concepteurs, des étudiants et des artistes, par sa facilité d'emploi et sa robustesse.

Un point clé dans le succès de ce circuit est que tous les schémas de conception Arduino ont été rendus librement accessibles dans le cadre d'une licence créative Commons.

C'est ce qui a permis de voir apparaître des circuits compatibles encore moins chers seul le nom Arduino est protégé. Les clones portent souvent des noms contenant une mention du type « duino » Boarduino, Seeduino ou freeduino.

Cela dit, les cartes officielles qui sont fabriquées en Italie se vendent toujours très bien. Les revendeurs les plus importants ne vendent que des cartes officielles dans un joli emballage et avec une garantie de qualité.

## Chapitre II : Présentation de la carte Arduino.

---

Une autre raison du succès d'Arduino est que ce circuit n'est pas limité au microcontrôleur.

Une grande quantité de cartes d'extension compatibles avec est apparue ce genre de carte, appelée bouclier « Shield », s'enfiche directement sur la carte Arduino dans une relation carte-mère/carte-fille.

Vous trouverez cartes d'extension pour quasiment tous les besoins, ce qui peut souvent vous éviter de devoir sortir le fer à souder, car il est bien plus pratique de connecter plusieurs cartes d'extension en les empilant les unes sur les autres.

Voici quelques-unes des cartes boucliers les plus répandues

- Ethernet, pour permettre à une carte Arduino de devenir un serveur web :
- Moteur, pour piloter un moteur électrique ;
- Hôte USB, pour contrôler des périphériques USB
- Relais, pour contrôler des relais depuis la carte Arduino

### II.2. Définition du module ARDUINO :

Le module ARDUINO est un circuit imprimé en matériel libre plateforme de contrôle dont les plans de la carte elle-même sont publiés en licence libre certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre.

Un microcontrôleur programmé peut analyser et produire des signaux électrique de manière à effectuer des tâches très diverses.

Arduino est utilisé dans beaucoup d'application comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commandé des moteurs et faire des jeux de lumière, communique avec l'ordinateur, commander des appareils mobiles « modélisme ».

## Chapitre II : Présentation de la carte Arduino.

---

Chaque module d'Arduino possède un régulateur de tension +5V et un oscillateur à quartz 16MHZ « ou un résonateur céramique dans certains modèles ».

Pour programmer cette carte, on utilise l'logiciel IDE Arduino.

### II.3. Les gammes de la carte ARDUINO :

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

- Le NG d'Arduino avec une interface d'USB pour programme et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 « ATmega328 pour une plus nouvelle version ».
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programme et usage d'un ATmega168.
- L'Arduino Bluetooth avec une interface de Bluetooth pour programme en utilisant un microcontrôleur ATmega168.
- L'Arduino Diecimila, avec une interface d'USB et utilisant un microcontrôleur ATmega168.
- L'Arduino Duemilanove, en utilisant un microcontrôleur l'ATmega 168 « ATmega328 pour une plus nouvelle version » et actionné par l'intermédiaire de la puissance d'USB /DC.
- L'Arduino Méga, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisant microcontrôleur ATmega328.

## Chapitre II : Présentation de la carte Arduino.

---

- L'Arduino Mega2560, utilisant un microcontrôleur ATmega2560, et possédant toute la mémoire à 256 KBS, elle incorpore également le nouvel ATmega8U2 « ATmega16U2 dans le jeu de puces d'USB de prévision ».
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- L'Arduino Esplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi ces types, nous avons choisi une carte Arduino UNO « carte basique », l'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation.

L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB.

En outre, des bibliothèques de fonctions clé en main sont également fournies pour l'exploitation d'entrées-sorties, cette carte est basée sur un microcontrôleur ATmega328 et des composants complémentaires.

La carte Arduino contient une mémoire morte de 1 kilo, elle est dotée de 14 entrées/sorties digitales « dont 6 peuvent être utilisées en tant que sortie PWM », 6 entrées analogiques et un cristal à 16 MHz.

Une connexion USB et possédant un bouton de remise à zéro et une prise jack d'alimentation.

### II.4.1. Pourquoi choisir Arduino UNO :

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée.

## Chapitre II : Présentation de la carte Arduino.

---

Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser.

De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en n'offrant à personne intéressées plusieurs avantages cités comme suit :

Le prix « réduits » : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, « les cartes Arduino pré-assemblées coutent moins de 2500 Dinars ».

Multi plateforme : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

Un environnement de programmation clair et simple : l'environnement de programmation Arduino « le logiciel Arduino IDE » est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

Logiciel Open Source et extensible : le logiciel Arduino et le langage Arduino sont publiés sous licence Open source, disponible pour être complète par des programmeurs expérimentés.

Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes « fonctionnant sur tout système d'exploitation », servant d'éditeur de code et de compilateur, et qui peut transférer le programma au travers de liaison série « RS232, Bluetooth ou USB selon le module ».

Matériel Open source et extensible : les cartes Arduino sont basées sue les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant.

## Chapitre II : Présentation de la carte Arduino.

---

Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sue plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le cout.

### II.4.2. La constitution de la carte Arduino UNO :

Un module Arduino est généralement construit d'un microcontrôleur ATMEL.

AVR, et de composant complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possédé au moins un régulateur linéaire 5V et un oscillateur à quartz 16MHz « 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.

Modèle UNO de la société ARDUINO est une carte électronique dont le noyau est un microcontrôleur ATMEL de référence ATmega328. Le microcontrôleur ATmega328 est un microcontrôleur 8 bits de la famille AVR dont la programmation peut être réalisée en langage C. Cette carte possède 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme étant des sorties PWM (pulse width modulation)), 6 entrées analogiques avec un convertisseur Analogique/Numérique de 10 bits résolution, 1 résonateur céramique (quartz) de 16 MHz, 1 connecteur ICSP (In circuit serial programmation) qui permet d'injecter le boot loader à l'intérieur du microcontrôleur, 1 connecteur jack pour une alimentation extérieur, un bouton de reset pour mettre processus à zéro.

L'avantage de cette carte c'est qu'elle n'a pas besoin de pilote pour faire la conversion FTDI USB/série, elle a juste un petit microcontrôleur ATmega328 (pour la version 2) programmé comme convertisseur USB/série.

Elle contient tout ce que le microcontrôleur a besoin pour fonctionner, il faut seulement la connecter avec un câble USB à un ordinateur ou avec une alimentation externe.

## Chapitre II : Présentation de la carte Arduino.

L'intérêt principal des cartes ARDUINO est leur facilité de mise en œuvre. AARDUINO fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions (clé en main) sont également fournies pour l'exploitation d'entrées sorties courantes : gestion des E/S, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus I2C, exploitation de servomoteurs, émission/réception série...etc.

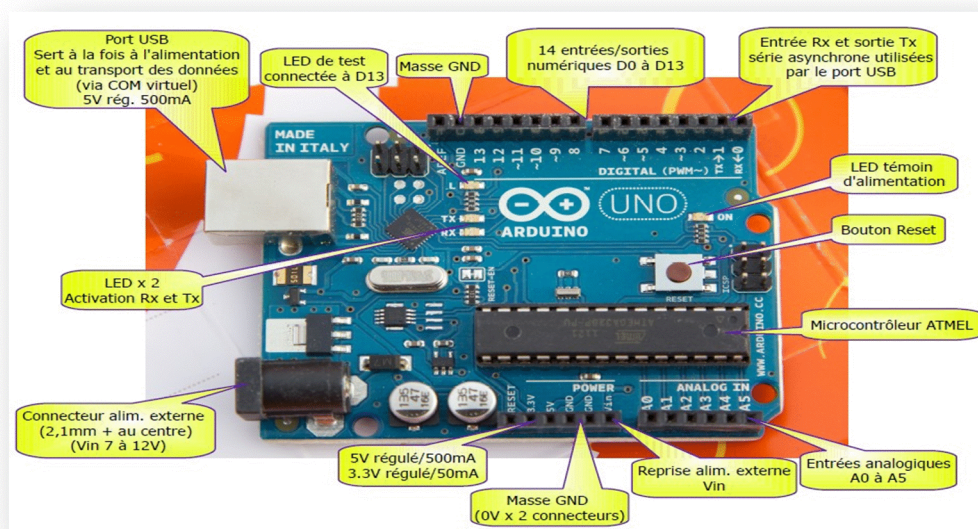


Figure 19 : la carte Arduino UNO

### II.4.3. Alimentation de la carte Arduino UNO :

La carte Arduino UNO peut être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500 mA) ou à l'aide d'une alimentation externe (non USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500 mA) ou des piles.

## Chapitre II : Présentation de la carte Arduino.

---

L'adaptateur secteur peut être connecté en branchant une prise de 2.1 mm dédiée du pôle 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 briches de la carte appelée 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. Si la carte est alimentée avec moins de 7 V, la broche 5 V 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.

Il est à noter qu'il est strictement dangereux d'utiliser une alimentation externe via la prise jack et avoir le câble USB connecté (risque d'endommager la carte).

Les broches d'alimentation sont les suivantes :

- VIN : à distinguer du 5V de la connexion USB ou autre source 5V régulée.

On peut alimenter la carte à l'aide de cette broche, ou si l'alimentation est fournie par le Jack d'alimentation, on peut 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, 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 toute autre source d'alimentation régulée.
- 3V3 : une alimentation de 3.3V fournie par le régulateur de 3.3V de la carte est disponible : ceci est intéressant maximale disponible sur cette broche est de 50mA.

## Chapitre II : Présentation de la carte Arduino.

---

### II. 4. 4. Protection du port USB contre la surcharge en intensité :

La carte Arduino UNO intègre un poly-fusible réinitialisable qui protège le port USB de votre ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité).

Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliquées au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

### II. 4. 5. Gestion des mémoires dans la carte Arduino

L'ATmega328 a 32 Ko mémoire FLASH pour stocker le programme (dont 0,5ko également utilisés par le bootloader). L'ATmega328 a également 2 ko de mémoire SRAM (volatile) et 1 ko d'EEPROM (non volatile –mémoire qui peut être lue à l'aide de la librairie EEPROM).

### II.4. 6. Les entrées / sorties numériques :

Chacune des 14 broches numériques de la carte UNO (numérotées de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant l'instruction `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 « résistance de rappel » (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)`.

Il ya entre ces broches celles qui des fonctionnalités en plus :

Communication série : broches 0 (RX) et 1 (TX). Utiliser pour recevoir (RX) transmettre (TX) les données séries de niveau TTL.

## Chapitre II : Présentation de la carte Arduino.

---

Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB vers série de la carte (composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur), on fait appelé à la transmission série a travers ces broches avec l'instruction `serial.print ()`, à condition que le câble USB soit déconnecté.

Interruption externes : broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.

Impulsion PWM (largeur d'impulsion modulée) : broches 3, 5, 6,. Fournissent une impulsion PWM 8-bits à 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.

12C : broches 4 (SDA) et 5 (SCL), supportent la communication de protocole 12C, disponible en utilisant la librairie `Wire/12C`.

LED : broches 13, il y a une LED incluse dans la care 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.

### II.4. 6. Les entrées analogiques :

La carte UNO dispose de 6 entrée de 0 à 5 analogiques (numérotée de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (sur 1024 niveaux soit de 0 à 1023) à l'aide de la fonction `analogRead ()` du langage Arduino.

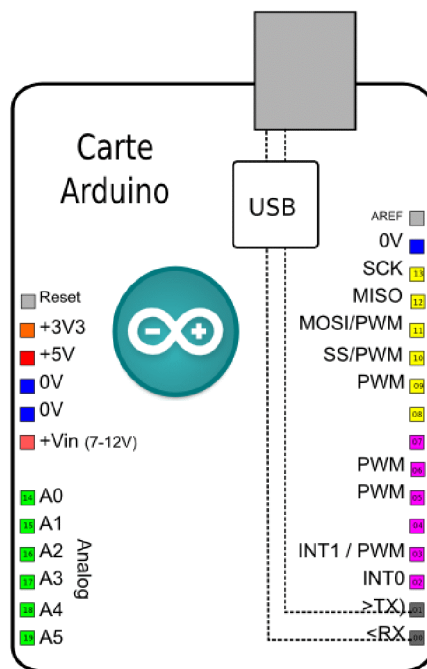
Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction `analogReference ()` du langage Arduino.

## Chapitre II : Présentation de la carte Arduino.

Autre broches :

Il y a deux autres broches disponibles sur la carte :

- **AREF** : tension de référence pour les entrées analogiques (si différent du 5V), utilisée avec l'instruction `analogReference()`.
- **RESET** : mettre cette broche au niveau BAS entraîne la réinitialisation (le redémarrage) du microcontrôleur, typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation du circuit.



**Figure 20** : brochage de carte arduino

### II.4. 7. Arduino et la communication avec l'extérieur :

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.

## Chapitre II : Présentation de la carte Arduino.

---

L'ATmega 328 dispose d'une UART (universal Asynchronous Receiver Transmitter ou émetteur -récepteur universel asynchrone en français) pour communication série de niveau TTL (5V) et qui est disponible sur les broches 0 (RX) et un (TX).

Un circuit intégré ATmega8U2 sur la carte assure la connexion entre cette communication série vers le port USB de l'ordinateur et apparait 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.

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 LED 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 la communication série sur les broches 0 et une librairie série logicielle permet également la communication série (limitée cependant) sur n'importe quelle broche numérique de la carte UNO.

L'ATmega328 supporte également ces protocoles I2C et SPI :

Le logiciel Arduino inclut la librairie Wire qui simplifie l'utilisation du bus I2C.

Pour utiliser la communication SPI (Interface série périphérique), la librairie pour communication SPI est disponible , il suffit de la faire inclure dans la programme au niveau de logiciel Arduino lors de la programmation.

### II.4. 8. Caractéristique mécanique :

Les longueurs et largeurs maximales de la UNO sont respectivement 6.86 cm et 5.33 cm, avec le connecteur USB et le connecteur d'alimentation jack s'étendant au-delà des dimensions de la carte.

Quatre trous de vis permettent à la carte d'être fixée sur une surface ou dans un boîtier.

## Chapitre II : Présentation de la carte Arduino.

---

Notre que la distance entre les broches 7 et 8 est de 0.16 pouces, et non un multiple des 0.1 pouces séparant les autres broches.

### **II.4 .9. Programmation :**

La carte Arduino UNO peut être programmée avec le logiciel Arduino, il suffit de sélectionner « Arduino UNO dans le menu Tools » board en fonction du microcontrôleur présent sur votre carte.

Le microcontrôleur ATmega328 présent sur la carte Arduino UNO est livré avec un bootloader « petit programme de démarrage vous » préprogrammé qui vous permet de transférer le nouveau programma dans le microcontrôleur externe.

Ce bootloader communique avec le microcontrôleur en utilisant le Protocol original STK500.

Vous pouvez bien sur passer outre le bootloader et programmer le microcontrôleur via le connecteur ICSP « programme série dans le circuit ».

La source du code le circuit intégré ATmega8U2 est disponible.

L'ATmega8U2 est chargé avec un bootloader DFU qui peut être active en connectant le cavalier au dos de la carte « près de la carte de l'Italie » et en réinitialisant le 8U2.

Vous pouvez alors utiliser le logiciel FLIP de chez Atmel « windows » ou le programme DFU « Mac OS X et linux ».

Pour charger le nouveau code, ou bien vous pouvez utiliser le connecteur ICSP avec un programme externe « pour récrire le bootloader DFU ».

### **II .4. 10. Réinitialisation « logicielle » automatique :**

Plutôt que de nécessiter un appui sur le bouton poussoir de réinitialisation avant un transfert de programme, la carte Arduino UNO a été conçue de telle façon qu'elle puisse être réinitialisée par un logiciel pourtant sur l'ordinateur.

## Chapitre II : Présentation de la carte Arduino.

---

Une des broches matérielles de control du flux « DTR » du circuit intégré ATmega8U2 est connecté à la ligne de réinitialisation de L'ATmega8U2 via un condensateur de 1001 nanofarads.

Lorsque cette broche est mise au niveau BAS, la broche de réinitialisation s'abaisse suffisamment longtemps pour réinitialiser le microcontrôleur. Le logiciel Arduino utiliser cette possibilité pour vous permettre de transférer votre programme dans la carte par un simple clic sur le bouton de transfert de la barre de boutons de l'environnement Arduino. Cela signifie que le bootloader peut avoir un temps mort plus court, la mise au niveau bas de la broche DTR étant bien coordonnée avec le début du transfert du programme.

Ce fonctionnement a d'autres implications, quand la carte UNO est connectée a un ordinateur fonctionnant soit sous MAC OS X ou linux, la carte se réinitialise à chaque fois qu'une connexion se fait entre elle et le logiciel « via le port USB ».

Pendant la demi-seconde suivant, le bootloader s'exécute sur la carte UNO. Puisqu'il est programmé pour ignorer les données mal formatées « c.à.d. toute donnée en plus du transfert d'un nouveau programme » il interceptera les quelques premiers octets de donnée envoyé à la carte après qu'une connexion soit établie. Si un programme données quand il démarre, assurez-vous que le logiciel avec lequel il communique attend une seconde après l'ouverture de la connexion avant d'envoyer ces données.

La carte UNO comporte une piste « du circuit imprimé » que vous pouvez couper pour désactiver la réinitialisation automatique les bords de chaque coté de la piste peuvent solidariser ensemble pour la réactiver.

Vous pouvez également désactiver la réinitialisation automatique en connectant une résistance de 110 ohm entre le +5V et la ligne de réinitialisation « reset »

## Chapitre II : Présentation de la carte Arduino.

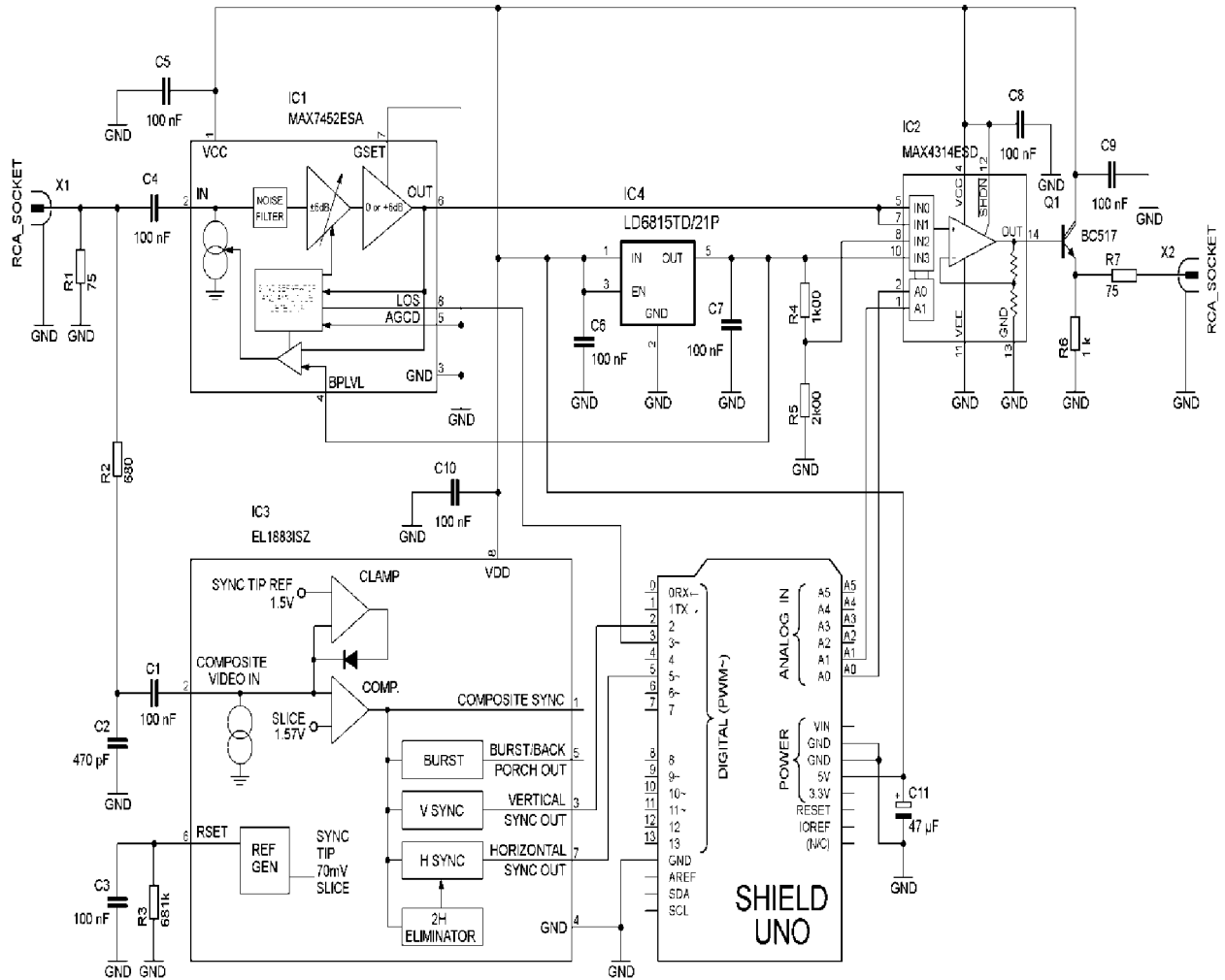


Figure 21 : Schéma électrique de carte Arduino UNO

### II.5. Définition d'un microcontrôleur ATmega328 :

Le microcontrôleur utilisé sur la carte ARDUINO UNO est un microcontrôleur ATmega328. C'est un microcontrôleur ATMEL de la famille AVR 8 bits.

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique.

Aujourd'hui, en soudant un grand nombre de composants encombrants; tels que les transistors : les résistances et les condensateurs tout peut être logé dans un petit boîtier

## Chapitre II : Présentation de la carte Arduino.

---

en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C.

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée.

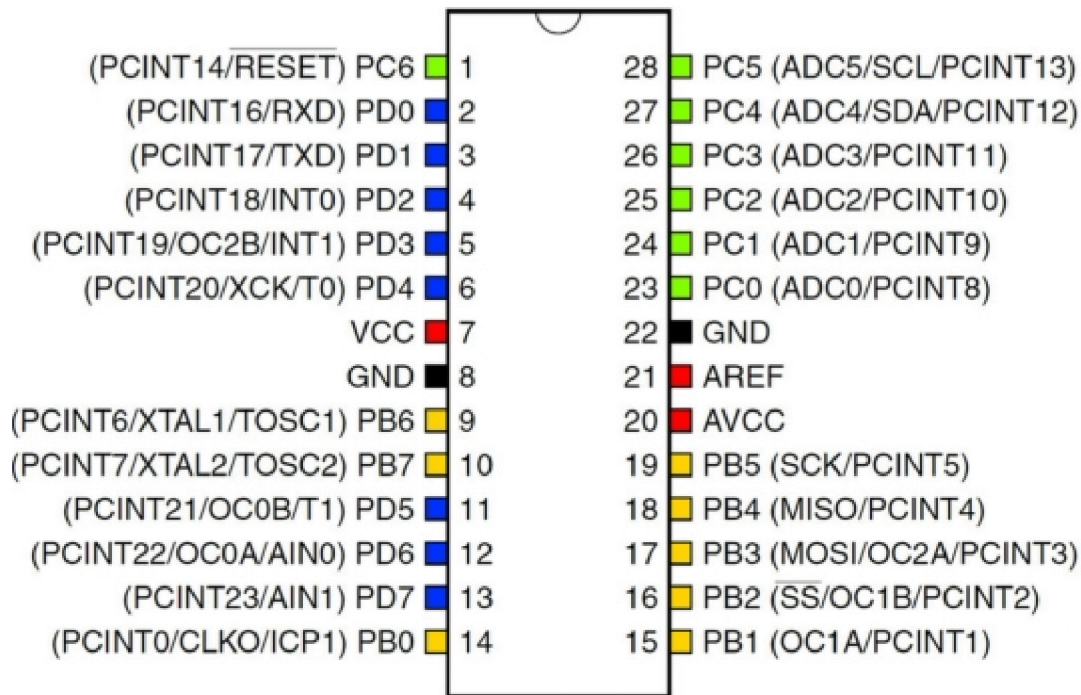
Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur, Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- **La Mémoire Flash** : c'est celle qui contiendra le programme à exécuter.

Cette mémoire est effaçable et réinscriptible mémoire programme de 32 KO « dont bootloader de 0.5 KO ».

- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. sa capacité est 2 Ko.
- **EEPROM** : c'est le disque dur du microcontrôleur. On y enregistre des informations qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

## Chapitre II : Présentation de la carte Arduino.



**Figure 22** : microcontrôleur ATmega 328

### II.6. Les principales caractéristiques de l'ATMega328 :

Le microcontrôleur ATMega328 de Atmel dispose

De 14 broches numériques d'entrées/sorties, dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée), réparties selon l'ordre suivant : OC0A (PD6), OC0 (PD5), OC1A (PB1), OC1B (PB3), OC2A (PB3), OC2B (PD3) et 2 (0 et 1) pour réception / émission série.

De 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques).

Ces entrées/ sorties sont réparties sur trois ports : port B, port D, port C (soit 23 broches E/s en tout).

- D'un courant max par broches E/S = 40 mA
- D'un courant max sur sortie 3.3V = 50mA

## Chapitre II : Présentation de la carte Arduino.

---

- D'une mémoire Flash de 32KB dont 512B utilisées par le boot loader.
- D'une mémoire SRAM de 2 KB.
- D'une mémoire EEPROM de 1KB.

Il contient aussi trois compteurs (Timer 0, Timer 1, Timer 2), le Timer 0 et le Timer 2 sont à comptage 8bits , le Timer 1 il est à comptage 16 bits, les Timers peuvent être utilisés pour génère les signaux PWM.

Certaines broches peuvent avoir plusieurs fonctions en fonction de ces programmations

PWM : le ATmega a 6 broches qui peuvent servir à cette fonction qui sont les broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3).

Convertisseur Analogique/Numérique : le ATmega328 possédé un convertisseur Analogique/Numérique d'une résolution de 10 bits, ce convertisseur peut être utilisé à travers 6 entrées multiplexées de ADC0(PC0) jusqu' à ADC5(PC5).

Gestion du bus 12C : ce bus est exploité via les deux broches SDA(PC5)/SCL(PC4).

Port série (USART) : émission/réception série via les broches TXD(PD1)/RXD(PD0).

Comparateur Analogique : le comparateur analogique intégré dans L'ATmega peut être utilisé à travers les deux broches AIN0(PD6) et AIN1(PD7), ce comparateur peut déclencher une interruption.

Watchdog Timer programmable : l'ATmega possédé un compteur dit de chien de garde programmable pour générer des interruptions à la fin de son comptage et il peut être utilisé comme étant un simple compteur.

Gestion d'interruption (24 sources possibles) : en résumé

- Interruption liées aux entrées INT0(PD2) et INT1(PD3).
- Interruption sur changement d'état des broches PCINT0 à PCINT23.
- Interruption liées aux Timers 0.1 et 2.
- Interruption liées au comparateur analogique.

## Chapitre II : Présentation de la carte Arduino.

---

- Interruption de fin de conversion ADC.
- Interruption du port série USART.

### II.7. Comparateur Analogique (AC) :

Le comparateur analogique compare les valeurs d'entrée sur la broche positive AIN0 (PB2) et la broche négative AIN1 (PB3). Quand la tension sur la broche AIN0 est plus haute que la tension sur la broche AIN1, le comparateur analogique ACO est mis à 1. Le comparateur analogique peut être utilisé pour déclencher la fonction de capture d'entrée du Timer/Compteur1. De plus, le comparateur analogique peut déclencher une interruption séparée exclusive. L'utilisateur peut choisir le front montant ou descendant pour déclencher la sortie ACO.

### II.8. Convertisseur Analogique / Numérique (ADC) :

Le convertisseur analogique/numérique ADC intégré dans l'ATMEGA est doté de caractéristique très intéressante avec une résolution sur 10 bits, 8 entrées simultanées avec une non linéarité inférieure à 1/2 LSB, une erreur à 0 V inférieure à 1 LSB, le temps de conversion est réglable de 65 à 260  $\mu$ S plus le temps est long, plus le résultat est précis. Près de 15000 échantillons/secondes avec le maximum de résolution sont possibles. Le convertisseur possède 7 entrées différentielles normales et 2 entrées différentielles avec gain optionnel de 10x (et 200x sur les boîtiers carré TQFP et MLF uniquement). La tension de référence peut être externe (conversion de 0 à AREF analogique, le maximum étant VCC) ou peut être interne avec la tension de référence de 2,56 V. L'ADC a une interruption sur conversion complète. La limitation du bruit en mode de sommeil est possible.

### II.9. Conception et logiciel :

#### II.9. 1. Choix de langage :

Pour réaliser ce projet, nous avons utilisé le langage de programmation C/C++, le C++ (qui est composé du c en grande partie) est un langage assez évolué, qui permet généralement de développer de façon plus souple et rapide et c'est l'un des langages de programmation les plus utilisés. Il est à la fois facile à utiliser et très efficace, c'est

## Chapitre II : Présentation de la carte Arduino.

---

un langage dit - de haut niveau-. Ce logiciel est l'un des logiciels les plus utilisés dans la programmation des microcontrôleurs et en particulier, pour la carte Arduino.

Il existe d'autres langages de haut niveau tels que JAVA, python, etc. mais pour la réalisation de notre projet, le langage C/C++ a été choisi.

### II.9.2. Le choix de l'IDE :

Un IDE (environnement de développement) libre et gratuit est distribué sur le site d'Arduino (compatible Windows, linux et Mac), d'autres alternatives existent pour développer pour Arduino (extensions pour Code Blocks, Visual Studio, Eclipse, Code, etc.).

L'interface de l'IDE Arduino est plutôt simple, il offre une interface minimale qui est épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique<sup>1</sup> et d'une barre d'outils rapide<sup>2</sup>. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus<sup>3</sup> plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console<sup>4</sup> affichant les résultats de la compilation du code source des opérations sur la carte, etc.

**Chapitre III : la réalisation**

## Chapitre III : la réalisation

---

### Préambule :

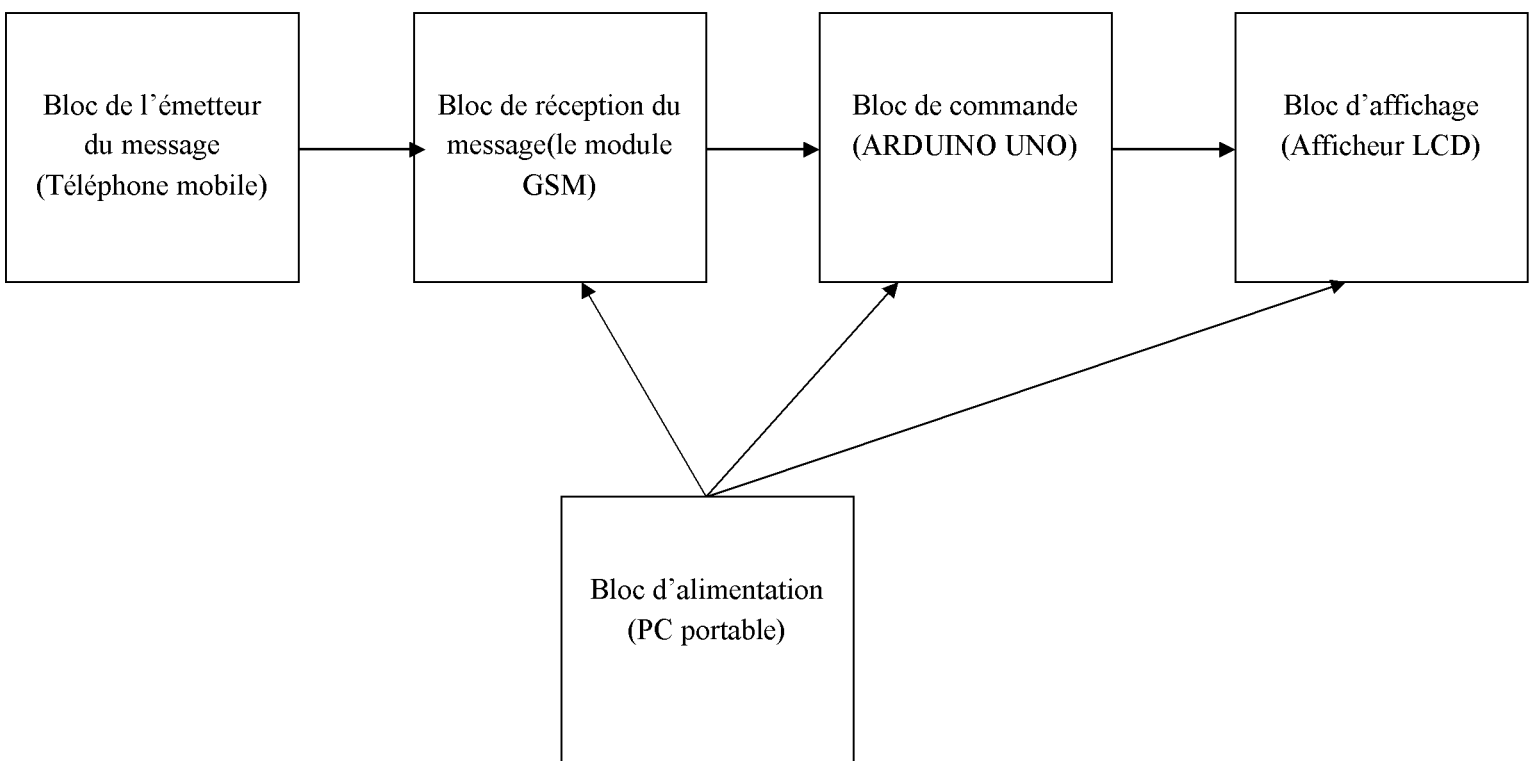
Ce chapitre illustre le principe de fonctionnement ainsi que les différents brochages des blocs et éléments du système.

### III.1.Principe de fonctionnement :

Après avoir reçu un message envoyé via un téléphone portable, ce dernier est réceptionné par le module GSM mais il ne sera pas lu avant que la carte ARDUINO UNO intervient en envoyant une commande pour la lecture d'un message présent au niveau du module GSM, après c'est la carte ARDUINO qui s'occupe de l'affichage de ce message sur l'afficheur LCD.

Les AT de commande utilisées sont des instructions pour commander le module GSM (bibliothèque sim900.h).

### III.2.Description des différents blocs de système :



**Figure 24.** Les différents blocs du système

### **III.3.bloc émetteur :**

#### **III.3.1. téléphone portable :**

Est un moyen de télécommunication par téléphone sans fil (téléphone portable), ce moyen de communication s'est largement répandu à la fin des années 1990. La technologie associée bénéficie des améliorations des composants électronique, notamment leur miniaturisation, ce qui permet aux téléphones d'acquérir des fonctions jusqu' alors réservées aux ordinateurs.

L'appareil téléphonique en lui-même peut être nommé (mobile, téléphone portable, téléphone cellulaire).

### **III.4. bloc de transmission :**

#### **III.4.1. le module GSM :**

##### **a) Définition en générale :**

Global System for Mobile communications norme numérique européenne utilisant plusieurs bandes de fréquences notamment de 900 à 1800 MHz, et sa portée peut atteindre jusqu'à 120Km. Le système GSM est aujourd'hui le principal système mobile.

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).

Le GSM utilise deux bandes de fréquences, l'une pour la voie montante (TX), l'autre pour la voie descendante (RX) auxquelles s'intègrent des canaux de signalisation, la puissance du signal est modulée selon la distance entre l'antenne et le GSM considéré, ce qui permet d'avoir une estimation de la distance entre un utilisateur.

---

#### **III.4.3.Spécifications :**

- Alimentation : 5V
- Connectivité : réseaux GSM et GPRS
- Type de communication : SMS, connexion de données GPRS, appels téléphoniques
- Antenne : non fournit (connecteur externe)

### Brochage de module GSM M10 :

---

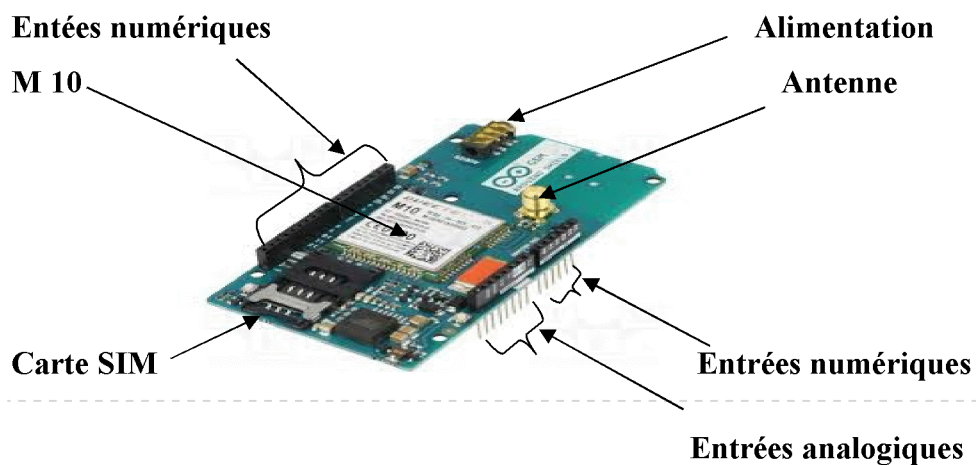


Figure 24 : brochage Shield GSM M10

### III.4.4. le service mini messagerie SMS :

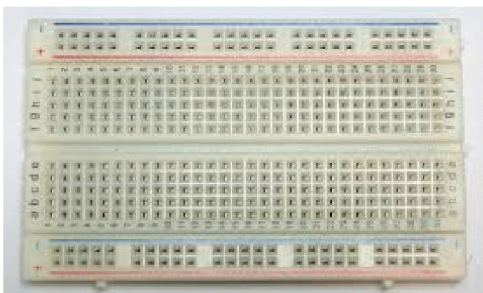
---

Le service de messagerie SMS, plus connu sous le sigle de SMS (pour « short message service » ou les noms de « texto » ou de « mini message » permet de transmettre de courts messages textuels. C'est l'un des services de la téléphonie mobile (il a été introduit par la norme GSM).

Un message est un ensemble de signes, il implique donc un codage par l'émetteur, et un décodage par le récepteur (d'où la nécessité d'un code commun). La théorie de l'information fut mise au point pour déterminer mathématiquement le taux d'information transmis dans la communication d'un message par un canal de communication, notamment en présence de parasites appelés bruits.

### III.5. La plaquette d'essai :

Une plaquette d'essai ou plaque d'essai sans soudure est un outil pédagogique indispensable pour découvrir l'électronique. Son principal avantage est de permettre de réaliser des montages rapidement sans souder aucuns composants. Il est donc possible de réutiliser le composant



**Figure 25 :** la plaquette d'essai sans soudeur

➤ Pour les autres blocs :

Bloc de commande (Arduino) = regarde le chapitre II

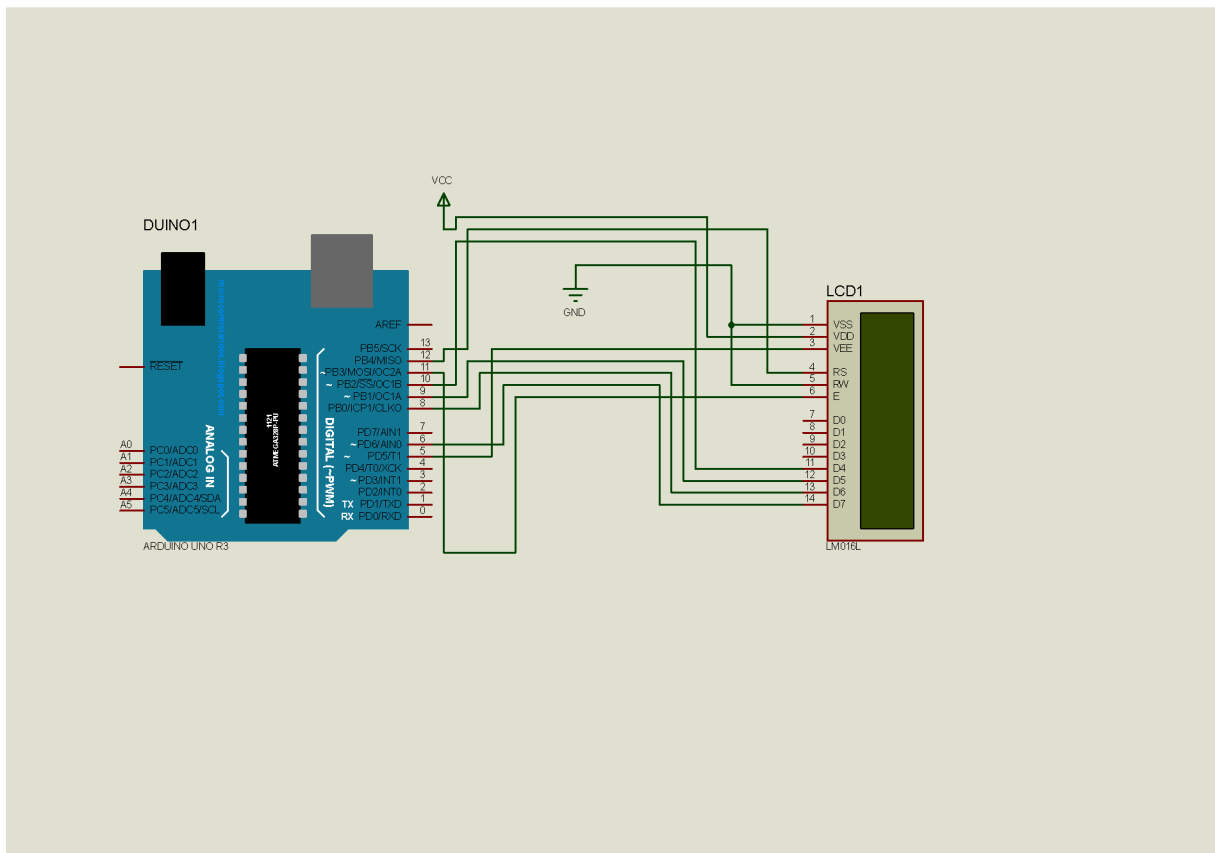
Bloc de d'affichage (LCD) = regarde le chapitre I

Pour réaliser le montage on utiliser les composant suivants :

- Une carte ARDUINO de type UNO
- Un modem GSM M10
- Un afficheur LCD
- Un téléphone mobile
- Une plaquette d'essai
- Des fils électrique
- Un câble USB

## II.6. Les différents branchements des différents blocs du système :

### III.6.1. branchement de la carte ARDUINO avec l'afficheur LCD :



**Figure 26 :** branchement de la carte ARDUINO avec l'afficheur LCD

## III.6.2. branchement des différents blocs du système :

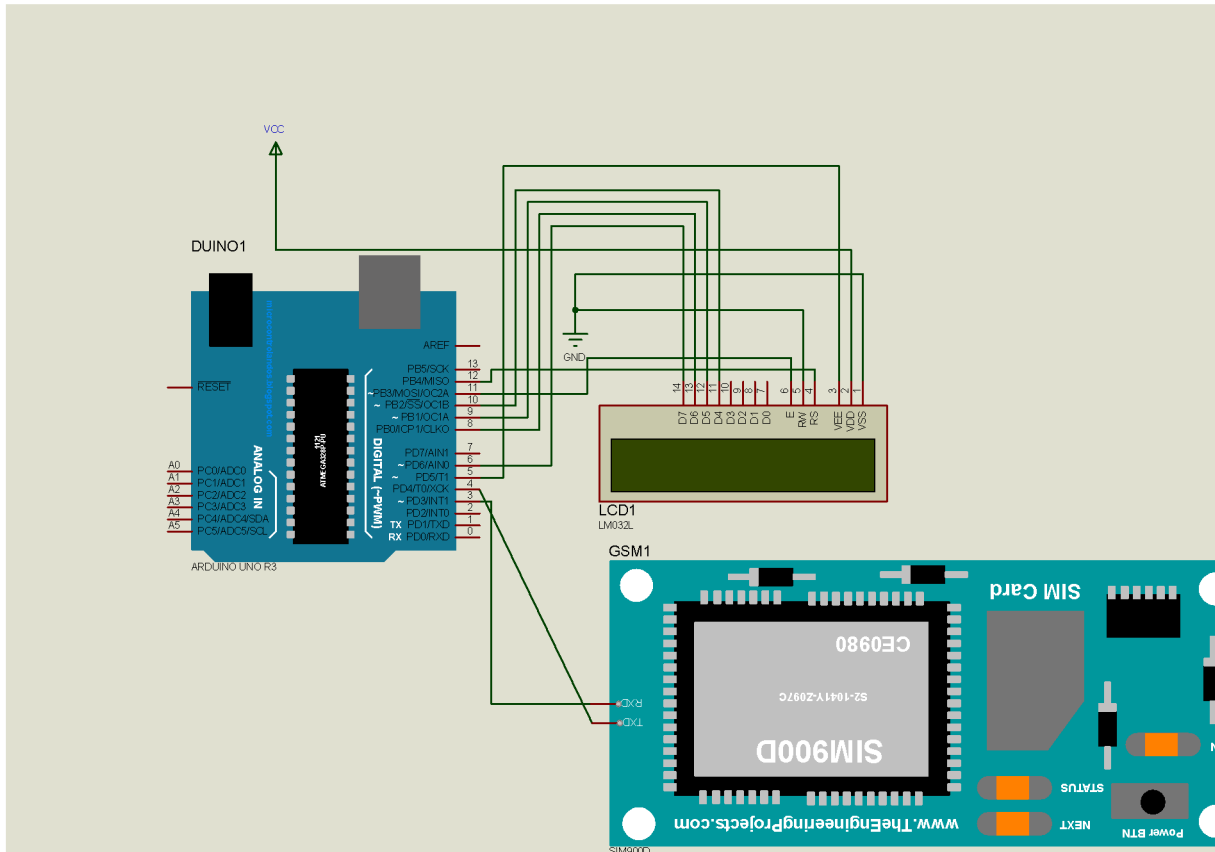


Figure 27 : schéma global de fonctionnement

### **Conclusion :**

En plus des explications de tous les blocs utilisés et leurs principes de fonctionnement pour réaliser notre application.

Cette dernière a nécessité également l'élaboration d'un programme. Donc nous avons réalisé un programme en langage C qui a permis de synchroniser le fonctionnement des différentes parties du système. En effet, après avoir envoyé un message sur l'afficheur, en quelques secondes pendant que le module GSM sera initialisé avant d'afficher ce message.

En perspective on propose l'exploitation de cette application avec d'autres afficheurs plus sophistiqués qui permettront l'affichage des images et vidéos en couleurs.

## **Le Programme**

```

#include <GSM.h>

#include <LiquidCrystal .h>

#define PINNUMBER ""

LiquidCrystal lcd(12, 11, 10, 9, 8, 6);

GSM gsmAccess;

GSMScanner scannerNetworks;

GSM_SMS sms;

char senderNumber[20];

char txtMsg[200];

int numChars = 0;

int validMessage = 0;

char myCellNumber[] = "+213772....." ;

void setup()
{
  analogWrite (5, 100);

  lcd.begin(16, 2);

  boolean notConnected = true;

  lcd.print("Init GSM ...");

  while(notConnected)
  {
    if(gsmAccess. begin(PINNUMBER)== GSM_READY
    notConnected = false;

  else
  {
    lcd.setCursor(0,1);

    lcd.print("Initialization.." );

  }
}

```

```
}  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Nom Operateur:");  
lcd.setCursor(0,1);  
lcd.print(scannerNetworks.getCurrentCarrier  
delay(2000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("Puiss Signale:");  
delay(2000);  
lcd.setCursor(0,1);  
lcd.print(scannerNetworks.getSignalStrength  
delay(2000);  
lcd.clear();  
lcd.setCursor(0,0);  
lcd.print("***** MSG *****");  
lcd.setCursor(0,1);  
lcd.print("Attente SMS" );  
}  
void loop()  
{  
char c;  
int i;  
int validMessage;  
if (sms.available())  
{
```

```
lcd.setCursor(0,1);  
lcd.print("SMS entrant");  
delay(2000);  
sms.remoteNumber (senderNumber, 20);  
if( strcmp( senderNumber, myCellNumbe  
{  
sms.flush();  
}  
else {  
i = 0;  
while( c = sms.read() ) {  
txtMsg[i] = c;  
i++;  
validMessage = 1;  
}  
numChars = i;  
}  
sms.flush();  
}  
if ( validMessage ) {  
if ( numChars <= 16 ) {  
for ( i = 0; i < numChars; i++) {  
lcd.setCursor(i, 1);  
lcd.print(txtMsg[i]);  
}  
}  
else {
```

```
for ( int q = 0; q <= numChars - 16
for (i = 0; i < 16; i++) {
lcd.setCursor(i, 1);
lcd.print(txtMsg[i+q]);
if (q == 0 && i == 15) {
delay(1800);
}
}
delay(650); // time for the scr
}
}
delay(1000); // time to hold the las
for (i = 0; i < 16; i++) {
lcd.setCursor(i, 1);
lcd.print(' ');
}
// delay(2000);
}
}
```

# **Bibliographie**

## **Ouvrages**

[01] C. Tavernier, « Arduino applications avancées ». Version Dunod.

[06] Eskimon, Olyte « Arduino pour bien commencer en électronique et en programmation ».

[07] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .

[8] Banzi, Massimo. Arduino : principe de base premiers montage par, Edition Dunod 2013.

[09] <http://arduino.cc/fr/Main/Librairies>. consulter le: mars 2015.

[8] Banzi, Massimo. Arduino : principe de base premiers montage par, Edition Dunod 2013.

[11] Bartmann, Erik. Le grand livre d'Arduino (texte imprimé), Edition Eyrolles, Del 2013.

## **Web sites**

Le site d'Arduino en français [www.arduino.cc/fr/](http://www.arduino.cc/fr/).

Le forum d'Arduino en français [www.arduino.cc/cgi-bin/yabb2/YaBB.pl?board=français](http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?board=français).

[Www. Technologuepro.com](http://www.Technologuepro.com)

[Www. sjeffroy.free.fr](http://www.sjeffroy.free.fr)

[Www. Electroniquepratique.com](http://www.Electroniquepratique.com)

[www.playarduino.fr](http://www.playarduino.fr).

[WWW.electronique-magazine.com](http://WWW.electronique-magazine.com)

## Résumé

Notre afficheur LCD, permet d'affichée à tous les messages envoyé via un téléphone portable. Après avoir reçu un message envoyé via un téléphone portable, ce dernier est réceptionné par le module GSM mais il ne sera pas lu avant que la carte ARDUINO UNO intervient en envoyant une commande pour la lecture d'un message présent au niveau du module GSM, après c'est la carte ARDUINO qui s'occupe de l'affichage de ce message sur l'afficheur LCD.