

**République Algérienne Démocratique Et Populaire**  
**Ministère de L'enseignement Supérieur Et De La Recherche Scientifique**



**Université Mohamed Seddik Benyahia, Jijel**

Faculté des sciences et de la technologie

Département d'électronique

**MEMOIRE DE FIN D'ETUDES**

Présenté en vue de l'obtention du diplôme de **MASTER** en  
**Systemes des télécommunications**

Thème :

**TRANSMISSION ET RECEPTION DES DONNEES  
NUMERIQUES SANS FILS EN UTILISANT DES  
MICROCONTROLEURS**

**Réalisé par M<sup>lles</sup> :**

- Ikram CHETTAB
- Sakina SOUILAH

**Encadré par :**

- Mr. Zoubir SANTOUH

**Promotion : Juillet 2019**

## Remerciements :

*Tout d'abord et avant tout, nous remercions le Dieu de tout puissant qui nous a donné la force, la volonté, la patience et le courage pour accomplir ce modeste travail.*

*Nous tenons à exprimer nos gratitude et notre profonde reconnaissance à nos chers parents, à toutes nos familles nos amis et à tous ceux qui nous ont aidé de près en de loin tout au long de notre travail*

*D'autres part nous tenons à remercier **Mr.Zoubir Santouh** notre encadreur pour son aide et son orientation aussi **Mr.Omar Tekkouk** pour ses conseils.*

*Nous tenons à exprimer notre gratitude aux membres de jury qui ont accepté de juger notre travail*

*Enfin, nous exprimons notre profonde reconnaissance à tous les enseignants qui ont contribué à notre formation.*

*A, tous, un grand merci.*

## *Dédicaces*

*Je tiens à dédier ce modeste travail à ma famille qui m'a beaucoup encouragé :*

*A mon cher père et à ma chère mère en témoignage et en gratitude de leur soutien permanent durant toutes mes années d'études.*

*A mes chères sœurs et chères copines qui sont la source de mon courage.*

*Sans oublier mon binôme Sakina, pour les moments de joie et de peine qu'on a partagés ensemble*

*Ainsi qu'à tous mes collègues de la promotion 2019.*

***Ikram.***

## *Dédicaces*

*Je dédie ce modeste travail :*

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse et leur soutien tout au long de mes études.*

*A mes chers frères Wassim et nawfel .*

*A mes chères copines pour leurs encouragements permanents, et leur soutien moral.*

*Ainsi à ma chère binôme Ikram.*

*Sans oublier mes collègues de la promotion de télécommunication 2019.*

*Sakina .*

# Table des matières

---

Remerciements.	i
Dédicaces.	ii
Liste des abréviations.	viii
Liste des figures/tableaux.	x

<b>Introduction générale</b> .....	1
------------------------------------	---

## **Chapitre I : Etude des techniques de modulation et du codage et quelque protocole de communication.**

I.1	Techniques de modulation.....	2
I.1.1	Introduction .....	2
I.1.2	Définition .....	2
I.1.3	Types de modulation.....	3
I.1.3.1	Modulation analogique .....	3
I.1.3.1.1	Modulation d’amplitude (AM).....	3
I.1.3.1.2	Modulation de fréquence (FM).....	4
I.1.3.1.3	Modulation de phase (PM).....	4
I.1.3.2	Modulation numérique.....	4
I.1.3.2.1	Modulation par décalage d’amplitude (ASK).....	4
I.1.3.2.2	Modulation par décalage de fréquence (FSK).....	5
I.1.3.2.3	Modulation par décalage de phase (PSK) .....	6
I.1.4	Modulation par impulsion.....	6
I.2	Techniques de codage.....	7
I.2.1	Codage NRZ.....	7
I.2.2	Codage NRZ-I (NRZ-INVERTED).....	8

## Table des matières

---

I.2.3	Codage RZ.....	8
I.2.4	Codage biphasé (Manchester).....	8
I.3	Protocoles de communications.....	10
I.3.1	Modes de communications .....	10
I.3.1.1	La liaison série synchrone.....	11
I.3.1.2	La liaison série asynchrone.....	11
I.3.1.3	Les différents modes d'exploitation d'une liaison série.....	11
I.3.2	La liaison point à point EIA-232 .....	12
I.3.2.1	Définition.....	12
I.3.2.2	Principe de fonctionnement du protocole RS-232 .....	12
I.3.3	Protocole de communication I2C .....	13
I.3.3.1	Définition .....	13
I.3.3.2	Principe de fonctionnement .....	14
I.3.4	L'interface de périphérique (SPI) .....	15
I.3.4.1	Généralités sur le SPI .....	15
I.3.4.2	Principe du fonctionnement du protocole SPI .....	16

## **Chapitre II : Etude d'un microcontrôleur avec exploration des modules USART, I2C, PSI, CAN.**

II.1	Les microcontrôleurs.....	19
II.1.1	Historique.....	19
II.1.2	Qu'est-ce qu'un microcontrôleur ? .....	19
II.1.3	Types des microcontrôleurs .....	20
II.1.4	Domaine d'application des microcontrôleurs.....	23
II.2	Le microcontrôleur ATmega328.....	24
II.2.1	Définition d'un Atmega328.....	24

## Table des matières

---

II.2.2	Description des broches d'Atmega328 .....	25
II.2.3	Architecture et fonctionnement d'Atmega328.....	27
II.2.3.1	Mémoire ATmega328.....	28
II.2.3.2	Base de temps Atmega328.....	29
II.2.3.3	Sous-système de chronométrage ATmega328 .....	29
II.2.3.4	Canaux de modulation de largeur d'impulsions .....	29
II.2.3.5	Interruptions ATmega328 .....	29
II.2.3.6	Communication série ATmega328 .....	30
II.2.3.6.1	ATmega série USART .....	30
II.2.3.6.2	Interface de périphérique série ATmega328 -SPI .....	31
II.2.3.6.3	Interface série à deux fils ATmega328 -TWI .....	31
II.2.3.6.4	Convertisseur analogique-numérique ATmega328 -CAN .....	32
II.3	Arduino.....	33
II.3.1	Constitution d'un Arduino UNO.....	34
II.3.1.1	Partie matérielle .....	34
II.3.1.2	Partie logicielle.....	37
II.3.1.2.1	Présentation du logicielle Arduino .....	37
II.3.1.2.2	Structure générale du programme (IDE Arduino) .....	38
 <b>Chapitre III : Réalisation pratique des platines d'émission et de réception des données numériques issues des capteurs et visualiser sur LCD.</b>		
III.1	Communication sans fil.....	39
III.1.1	Définition.....	39
III.1.2	Protocoles de communications sans fil.....	39
III.2	Outils de conception et réalisation .....	41
III.2.1	Modules Bluetooth .....	41
III.2.1.1	Module HC05.....	41

## Table des matières

---

III.2.1.2 Module HC-06.....	42
III.2.2 Les capteurs.....	43
III.2.2.1 Capteur de température LM35.....	43
III.2.2.2 Capteur d'humidité DHT11.....	45
III.2.2.3 Capteur ultrason HC-SR04.....	46
III.2.2.4 Capteur de lumière LDR.....	48
III.2.3 Servomoteur .....	48
III.3 Conception et réalisation.....	49
III.3.1 Système de communication Android Arduino.....	49
III.3.2 Système de communication entre deux Arduino .....	52
III.3.2.1 Transmission et réception d'une seule donnée.....	52
III.3.2.2 Transmission et réception d'un paquet de données numériques .....	53
III.3.2.2.1 Simulation avec ISIS PROTEUS .....	53
III.3.2.2.2 Réalisation pratique .....	55
Conclusion générale :.....	60
Annexes.....	61
Références biblio-Webographiques.....	65



# *Liste des abréviations*



<b>ADC</b>	Analog to Digital Converter.
<b>AM</b>	Amplitude modulation.
<b>ASK</b>	Amplitude Shift Keying.
<b>CAN</b>	Convertisseur analogique numérique.
<b>CI</b>	Circuit Intégré.
<b>CISC</b>	Complex Instruction Set Computer.
<b>DAC</b>	Digital to Analog Converter.
<b>EIA-232</b>	Electronic Industries Alliance 232.
<b>FM</b>	Frequency modulation.
<b>FSK</b>	Frequency Shift Keying.
<b>IDE</b>	Integrated Developpent Environnement.
<b>IEEE</b>	Institue of Electrical and Electronics Engineers.
<b>I2C</b>	Inter-integrated circuit.
<b>LAN</b>	Local Area Network.
<b>LCD</b>	Liquid Crystal Display.
<b>MOSI</b>	Master Out Slave In.
<b>MISO</b>	Master In Slave Out.
<b>NRZ</b>	No Return to Zero.
<b>PM</b>	Phase Modulation.

<b>PSK</b>	Phase Shift Keying.
<b>PWM</b>	Pulse Width Modulation.
<b>RISC</b>	Reduced Instruction Set Computer
<b>RS232</b>	Recommended Standard 232.
<b>RTC</b>	Real Time Clock.
<b>RZ</b>	Return to Zero.
<b>SCLK</b>	Serial Clock.
<b>SDA</b>	Serial Data.
<b>SPI</b>	Serial Peripheral Interface.
<b>SSn</b>	Slave Select n.
<b>UHF</b>	Ultra High Frequency.
<b>USART</b>	Universal Synchronous Asynchronous Receiver/Transmitter.
<b>WIFI</b>	Wireless Fidelity.
<b>WPAN</b>	Wireless Personal Area Network.

*Listes des figures et des  
tableaux*

## Liste des figures

---



<b>Figure I.1</b>	Schéma bloc de la modulation.	2
<b>Figure I.2</b>	Modulation ASK.	5
<b>Figure I.3</b>	Modulation FSK.	5
<b>Figure I.4</b>	Modulation QPSK.	6
<b>Figure I.5</b>	Méthodes de codage : (a) NRZ unipolaire, (b) NRZ bipolaire, (c) RZ unipolaire, (d) RZ bipolaire, (e) Manchester, (f) NRZI.	9
<b>Figure I.6</b>	Modes de communication série/parallèle.	11
<b>Figure I.7</b>	Protocole RS232.	12
<b>Figure I.8</b>	Le format de trame du protocole RS232.	13
<b>Figure I.9</b>	Schéma synoptique d'une liaison I2C série.	13
<b>Figure I.10</b>	Bus I2C (Condition de départ).	14
<b>Figure I.11</b>	Transmission des données sur liaison I2C série.	15
<b>Figure I.12</b>	Bus I2C (Condition d'arrêt).	15
<b>Figure I.13</b>	Synoptique d'une liaison SPI Maître-Esclave.	17
<b>Figure I.14</b>	Synoptique d'une liaison SPI Maître-Multi-Esclaves en mode normal.	17
<b>Figure I.15</b>	Synoptique d'une liaison SPI Maître-Multi-Esclaves en mode guirlande.	18

<b>Figure II.1</b>	Schéma interne d'un microcontrôleur.	21
<b>Figure II.2</b>	Les appareils de communication.	23
<b>Figure II.3</b>	Les appareils médicale.	23
<b>Figure II.4</b>	Les appareil de sécurité.	23
<b>Figure II.5</b>	Les appareils industriels.	24
<b>Figure II.6</b>	Hifi, TV, vidéo.	24
<b>Figure II.7</b>	Schéma de brochage d'Atmega328.	25
<b>Figure II.8</b>	Ports E/S de l'Atmega328.	26
<b>Figure II.9</b>	Schéma bloc d'Atmega328.	27
<b>Figure II.10</b>	Schéma fonctionnel d'un module USART.	30
<b>Figure II.11</b>	Schéma fonctionnel du module SPI.	31
<b>Figure II.12</b>	Schéma fonctionnel du module TWI.	32
<b>Figure II.13</b>	Module ADC.	33
<b>Figure II.14</b>	Logo Arduino.	33
<b>Figure II.15</b>	Constitution de la carte Arduino UNO.	34
<b>Figure II.16</b>	Brochage d'Arduino UNO.	35
<b>Figure II.17</b>	Ports de communications série.	37
<b>Figure II.18</b>	La fenêtre du logiciel Arduino.	37
<b>Figure II.19</b>	L'interface de l'Arduino IDE.	38
<b>Figure III.1</b>	Module WIFI ESP8266 .	40
<b>Figure III.2</b>	Logo WIFI.	40
<b>Figure III.3</b>	Module XBee S2TH.	40
<b>Figure III.4</b>	Logo ZigBee.	40
<b>Figure III.5</b>	Logo Bluetooth.	41
<b>Figure III.6</b>	Module Bluetooth HC-05.	41

<b>Figure III.7</b>	Module Bluetooth HC-06.	42
<b>Figure III.8</b>	Interfaçage d'un module Bluetooth avec la carte Arduino.	43
<b>Figure III.9</b>	Capteur de température LM35.	44
<b>Figure III.10</b>	Brochage de capteur de température.	44
<b>Figure III.11</b>	Interfaçage d'un capteur de température LM35 avec la carte Arduino.	46
<b>Figure II.12</b>	Brochage d'un capteur d'humidité DHT11.	46
<b>Figure III.13</b>	Exemple de liaison d'un capteur DHT11 avec Arduino.	46
<b>Figure III.14</b>	Capteur ultrasonique.	46
<b>Figure III.15</b>	Interfaçage du capteur ultrasons HC-SR04 avec la carte Arduino.	47
<b>Figure III.16</b>	Signale d'entrée et sortie du capteur HC-SR04.	48
<b>Figure III.17</b>	Photorésistance	48
<b>Figure III.18</b>	Laissions d'un servomoteur avec Arduino	49
<b>Figure III.19</b>	Schéma bloc d'un système communication Arduino-Androïde	50
<b>Figure III.20</b>	Schéma électrique du système de commande androïde Arduino	51
<b>Figure III.21</b>	(a) réalisation pratique du système de commande d'un servomoteur via Androïde, (b) degré de rotation 45°, (c) rotation à 90, (d) rotation à 180°.	52
<b>Figure III.22</b>	Schéma électrique d'une communication sans fils entre deux Arduinos. ( <i>Envoi d'une seule donnée</i> )	53
<b>Figure III.23</b>	Simulation d'une communication sans fils entre deux Arduinos. ( <i>Envoi d'une seule donnée</i> ).	54
<b>Figure III.24</b>	Schéma bloc d'une communication sans fils entre deux Arduinos. ( <i>Envoi d'un paquet de donnée</i> ).	55
<b>Figure III.25</b>	Schéma électronique d'une communication sans fils entre deux Arduinos. ( <i>Envoi d'un paquet de donnée</i> ).	56
<b>Figure III.26</b>	Simulation d'une communication sans fils entre deux Arduinos.	57

---

<b>Figure III.27</b>	Réalisation pratique d'une communication sans fils entre Arduino. (Envoi d'un paquet de donnée).	58
<b>Figure III.28</b>	Réalisation pratique sur circuits imprimés d'une communication sans fils entre Arduino. (Envoi d'un paquet de donnée).	58

### Liste des tableaux

---



<b>Tableau III.1</b>	Description des broches Bluetooth HC-05.	42
<b>Tableau III.2</b>	Description des broches Bluetooth HC-06.	43
<b>Tableau III.3</b>	Description des caractéristiques du capteur DHT11.	45
<b>Tableau III.4</b>	Description des caractéristiques du capteur HC-SR04.	47



# *Introduction générale*

### Introduction générale :

La nécessité de communiquer et de transmettre des informations à des endroits éloignés est antérieure à la civilisation et existe depuis l'aube de l'humanité. Ces informations peuvent être communiquées en personne ou via des moyens et des dispositifs permettant d'acheminer des informations vers des sites distants. Nous définissons ce transfert d'informations à travers le temps et le lieu comme des "télécommunications". Le terme «Télécommunications» désigne généralement la science et la technologie de la communication à distance.

L'idée d'échanger des informations et communiquer à distance a fait naître les réseaux informatiques. Les ordinateurs ont été d'abord connectés en local, créant des réseaux locaux propriétaires. Puis, le besoin d'envoyer des données à distance a élargi les communications entre sites distants, créant des réseaux à large étendue.

L'objectif principal de notre projet est de concevoir un système de communication sans fil qui consiste à transmettre et recevoir des données numériques en utilisant des microcontrôleurs. Nous avons fait le choix de travailler avec la carte Arduino à cause de sa simplicité et sa facilité de fonctionnement sur un réseau local sans fil.

Dans le premier chapitre, nous aborderons les principaux processus qui entrent dans les systèmes de transmission. Nous commencerons par introduire les techniques de modulations et du codage d'informations, notamment nous définirons les différents types de protocoles de communications.

Dans le second chapitre, nous introduirons l'architecture des microcontrôleurs avec exploration ses modules de communication, notamment l'architecture de la carte Arduino et son fonctionnement.

Le dernier chapitre sera consacré à la réalisation et conception matérielle et logicielle, nous commencerons par introduire quelque protocoles de communication sans fils ainsi que l'étude des différents composants utilisés dans notre projet.

# **CHAPITRE I**

---

**Etude des techniques de modulation,  
du codage et quelques protocoles  
de communication.**

Dans ce premier chapitre nous allons introduire les principaux processus qu'un système de communication peut parcourir lors de la transmission des données. Dans un premier temps nous allons définir les différentes techniques de modulation et de codage qui sont les mécanismes centraux de ces systèmes, on s'attarde ensuite à définir les types de communication en introduisant ses protocoles de communication.

## I.1 Technique de modulation :

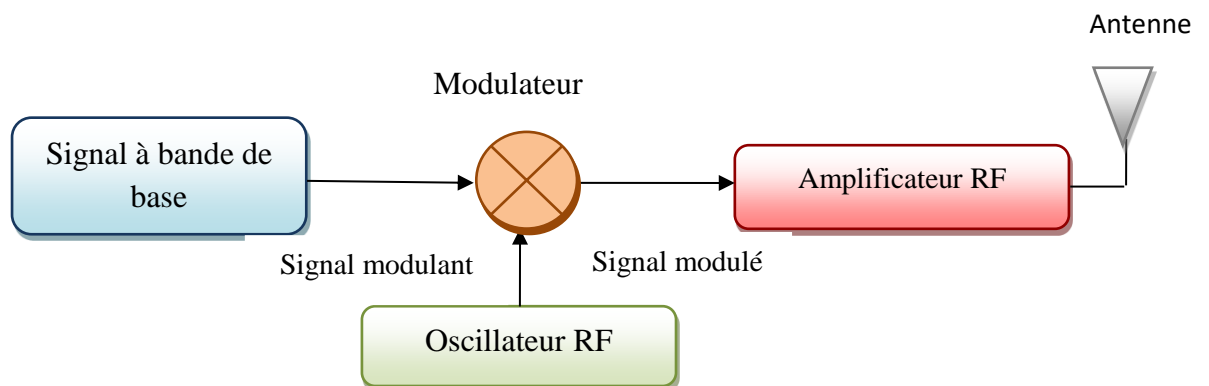
### I.1.1 Introduction :

Dans un système de transmission les signaux de basse fréquence, ne peuvent pas se propager efficacement sur de longue distance suite aux différents bruits et perturbations qu'ils peuvent rencontrer tout au long du trajet. Ils ont besoin de transporteurs pour les transporter sur plusieurs kilomètres.

Ces signaux sont donc acheminés par une onde radio porteuse de haute fréquence capable de parcourir de longues distances sans atténuations. C'est le processus de la modulation [1].

### I.1.2 Définition :

La modulation consiste à modifier un des paramètres de la porteuse afin d'intégrer un signal d'information dans le signal de porteuse et de l'envoyer sur un canal de transmission [2]. La figure (I.1) illustre les différents blocs qui constituent la modulation.



**Figure I.1:** Schéma bloc de la modulation.

Le but de la modulation est de transférer une donnée source sur un canal de la manière la mieux adaptée à ce canal et d'exploiter le maximum de données dans le moins de spectre possible. Cet objectif, appelé efficacité spectrale, mesure la rapidité avec laquelle les données peuvent être transmises dans une largeur de bande attribuée [2]. On distingue :

- ✓ Le signal d'information est un signal qui porte le message à envoyer que l'on appelle signal modulant  $m(t)$ .
- ✓ Le signal de la porteuse est un signal sinusoïdal de haute fréquence. Il s'écrit en général sous la forme :

$$s_p(t) = A(t)\cos[\varphi(t)] \quad (\text{I.1})$$

L'un des avantages importants de la modulation est la réduction de la taille de l'antenne car la longueur de l'antenne est proportionnelle à la moitié de la longueur d'onde, aussi la sélection d'une fréquence porteuse différente pour différentes stations d'émission peut éliminer le chevauchement des bandes de fréquences [2].

### I.1.3 Types de modulation :

La modulation peut être divisée en deux principaux types : modulation *analogique* et modulation *numériques*.

#### I.1.3.1 Modulation analogique :

En modulation analogique, un signal d'information analogique varie de façon continue et modifie un des paramètres indépendants de la porteuse décrivant son comportement qui sont son amplitude  $A(t)$ , et sa fréquence  $\varphi(t)$ . Dans ce cas, le changement du paramètre de porteuse est continu conformément au changement du signal analogique d'origine [2].

Chacun de ces paramètres peut être modifié directement avec le signal modulant, ce qui donne une sortie modulée [1]. La modulation analogique se divise elle-même en modulation d'amplitude, et modulation de fréquence.

##### I.1.3.1.1 Modulation d'amplitude (AM) :

En modulation d'amplitude, l'amplitude de l'onde porteuse varie en fonction du signal

---

modulant [3].

Un modulateur d'amplitude multiplie directement le signal de porteuse  $S_p(t)$  par le signal de message  $m(t)$  [1]. Le signal modulé s'écrit sous la forme :

$$S_m(t) = [A + m(t)]\cos\omega_p t \quad (\text{I.2})$$

### I.1.3.1.2 Modulation de fréquence (FM) :

En modulation de fréquence, la fréquence instantanée  $\omega(t)$  est une fonction du temps qui varie autour de la fréquence non modulée  $\omega_0$  d'une quantité  $\Delta\omega$  proportionnelle à la valeur instantanée du signal de modulation  $m(t)$  [1].

### I.1.3.1.3 Modulation de phase (PM) :

En modulation de phase, la phase devient fonction du temps [ $\theta(t) = \theta_0 \pm \Delta\theta(t)$ ] et change directement avec le signal de message modulant autour de la phase non modulée  $\theta_0$  de sorte que  $\Delta\theta(t) = \mu_p m(t)$  et  $\mu_p$  est l'indice de modulation de phase [1]. Ainsi, la phase instantanée de la porteuse modulée varie autour d'une valeur constante et l'écart de phase suit le signal modulant [2].

### I.1.3.2 Modulation numérique :

La modulation numérique élimine le bruit de communication et renforce l'intensité de l'intrusion du signal. Elle offre davantage de capacité de données, une sécurité de l'information élevée et l'accessibilité d'un système plus rapide grâce à une communication d'une qualité exceptionnelle. Par conséquent, les techniques de modulation numériques ont une demande énorme en raison de leur capacité à communiquer des quantités d'informations supérieures aux techniques AM (modulation analogique) [5]. On distingue :

#### I.1.3.2.1 Modulation par décalage d'amplitude (ASK) :

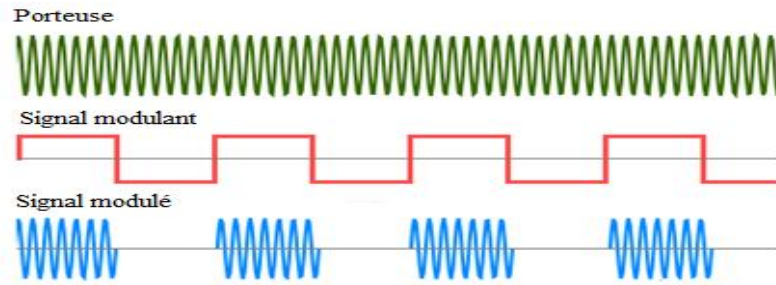
Consiste à faire varier l'amplitude du signal selon la loi de transcodage associée. Celui-ci s'exprime alors :

$$S(t) = A(t) \cos(\omega_0 t + \varphi) \quad (\text{I.3})$$

Avec :  $A(t) = \sum_k a_k h(t - kT_s)$

Où  $h(t)$  est un filtre de mise en forme des impulsions. Ce type de modulation est simple à réaliser mais assez peu employé pour  $M > 2$  car ses performances sont moins bonne que celle des autres modulations, notamment en ce qui concerne son immunité au bruit [4].

La figure suivante montre la forme d'onde de ce type de modulation :



*Figure I.2* : Modulation ASK.

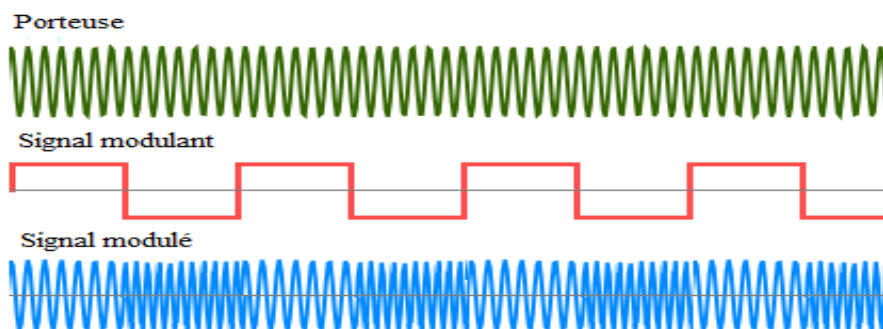
#### I.1.3.2.2 Modulation par décalage de fréquence (FSK) :

La fréquence instantanée, dérivée de la phase instantanée, qui peut prendre plusieurs valeurs associées aux états possibles. Après modulation, le signal a pour expression:

$$S(t) = A \cos[(\omega_0 t + 2\pi a_k \Delta F)t] \quad (\text{I.4})$$

Une modulation fréquemment utilisée considère le cas particulier  $m = 0.5$ . Le spectre est alors concentré autour de la fréquence porteuse et cette modulation prend naturellement le nom de modulation à déplacement minimum de fréquence [5].

La figure suivante illustre la forme d'onde de ce type de modulation :



*Figure I.3* : Modulation FSK.

### I.1.3.2.3 Modulation par décalage de phase (PSK) :

Le seul paramètre susceptible de varier est la phase de l'onde porteuse. A la sortie du modulateur, le signal s'exprime :

$$s(t) = A \sum_k (h(t - kT_s) \cos(\omega_0 t + \varphi_k)) \quad (\text{I.4})$$

Où A représente l'amplitude constante de l'onde porteuse et la valeur de la phase pendant un intervalle de temps  $[kT_s; (k + 1)T_s]$  [5].

Un exemple de la forme d'onde de ce type de modulation est présenté dans la figure (I.4).

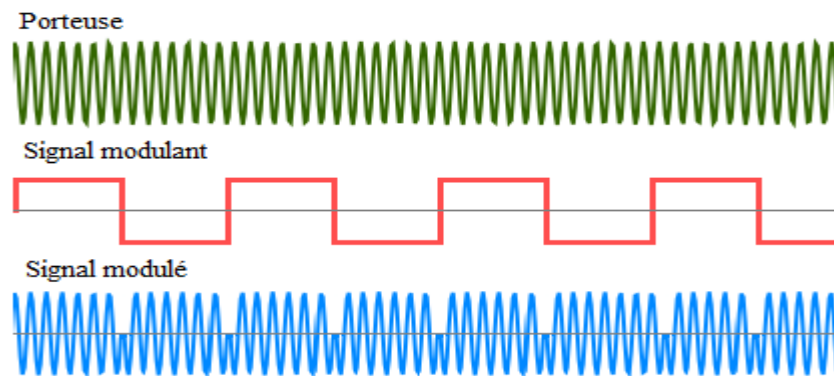


Figure I.4 : Modulation QPSK.

### I.1.4 Modulation par impulsion :

La modulation par impulsion consiste à moduler le signal à l'aide d'une porteuse d'un train d'impulsions à très haut taux de répétition (fréquence fondamentale élevée  $f_0$  et très courte période T). Cette modulation est divisée en modulation analogique d'impulsions et modulation numérique d'impulsion [7].

La modulation d'amplitude d'impulsion est une technique dans laquelle l'amplitude de chaque impulsion est contrôlée par l'amplitude instantanée du signal de modulation. C'est un système de modulation dans lequel le signal est échantillonné à intervalles réguliers et chaque échantillon est rendu proportionnel à l'amplitude du signal au moment de l'échantillonnage. Cette technique transmet les données par codage en amplitude d'une série d'impulsions de signal [6].



## I.2 Technique de codage :

Le codage est le processus de conversion des données ou d'une séquence donnée de caractères, symboles, alphabets, etc, dans un format spécifié, pour la transmission sécurisée de données. Il consiste à utiliser divers modèles de niveaux de tension ou de courant pour représenter les 1 et les 0 des signaux numériques sur la liaison de transmission [7].

De nombreuses normes de codage différentes sont utilisées dans les transmissions des données. Le point important est que l'émetteur et le récepteur doivent utiliser une norme de codage commune pour que les données puissent être interprétées correctement à la réception [8].

La technique de codage des données est divisée en types suivants, en fonction du type de conversion de données [7] :

- **Données analogiques en signaux analogiques** : Les techniques de modulation telles que la modulation d'amplitude, la modulation de fréquence et la modulation de phase de signaux analogiques entrent dans cette catégorie.
- **Données analogiques en signaux numériques** : Ce processus peut être qualifié de numérisation, qui est effectuée par modulation par codage d'impulsions.
- **Données numériques en signaux analogiques** : Les techniques de modulation telles que la modulation par décalage d'amplitude, la modulation par décalage de fréquence, la modulation par décalage de phase, etc., entrent dans cette catégorie.
- **Données numériques en signaux numériques** : Il existe plusieurs façons de convertir des données numériques en d'autres signaux numériques. Certains d'entre eux sont : NRZ, Codage biphase (Manchester), etc.

### I.2.1 Codage NRZ :

Ce code est l'un des plus anciens et des mieux connus de tous les codes de canaux. Fondamentalement, un 1 logique est représenté par une alimentation positive (niveau logique haut) pendant tout le créneau temporel attribué pour un seul bit, alors qu'un zéro logique est signalé par une alimentation négative (niveau logique bas) pendant toute la tranche de temps allouée pour un seul bit [8] [9].

La fin ou le début d'un bit ne sera pas indiqué et il conservera le même état de tension si la valeur du bit précédent et la valeur du bit actuel sont identiques [8].

La plupart des signaux NRZ sont unipolaires, ce qui signifie que les deux signaux ont la même polarité, généralement positifs. Certaines interfaces série utilisent un format NRZ bipolaire dans lequel les niveaux de tension positif et négatif sont utilisés, voir *la figure (I.5)* [10].

### I.2.2 NRZ - I (NRZ - INVERTED) :

Une variation de NRZ est NRZI, comme le montre *la figure (I.5f)*. Il semble similaire à NRZ, mais fonctionne différemment. Il nécessite un mécanisme de synchronisation. Chaque bit est codé à la demi-onde du signal d'horloge. En codage NRZI, si le bit est égal à 1, le signal change d'état, de négatif à positif ou de positif à négatif. Si le bit est égal à 0, le signal reste dans le même état. NRZI peut représenter les mêmes données que NRZ, avec moins de transitions. Cela signifie que cela peut être transmis plus rapidement, voire la figure [11].

Les codes NRZ ont pour **inconvénient** que la synchronisation de l'horloge de l'émetteur avec l'horloge du récepteur est complètement perturbée lorsqu'il existe une chaîne de 1 et de 0. Par conséquent, une ligne d'horloge séparée doit être fournie [8].

### I.2.3 Codage RZ :

Un autre format populaire est RZ ou revenir à zéro, comme illustré à *la figure (I.5c)*. Une impulsion positive représente un 1 binaire qui revient à zéro pendant un intervalle de temps d'un bit. Une largeur d'impulsion typique correspond à 50% du temps de bit. Aucune impulsion n'est transmise pour un 0 binaire [10].

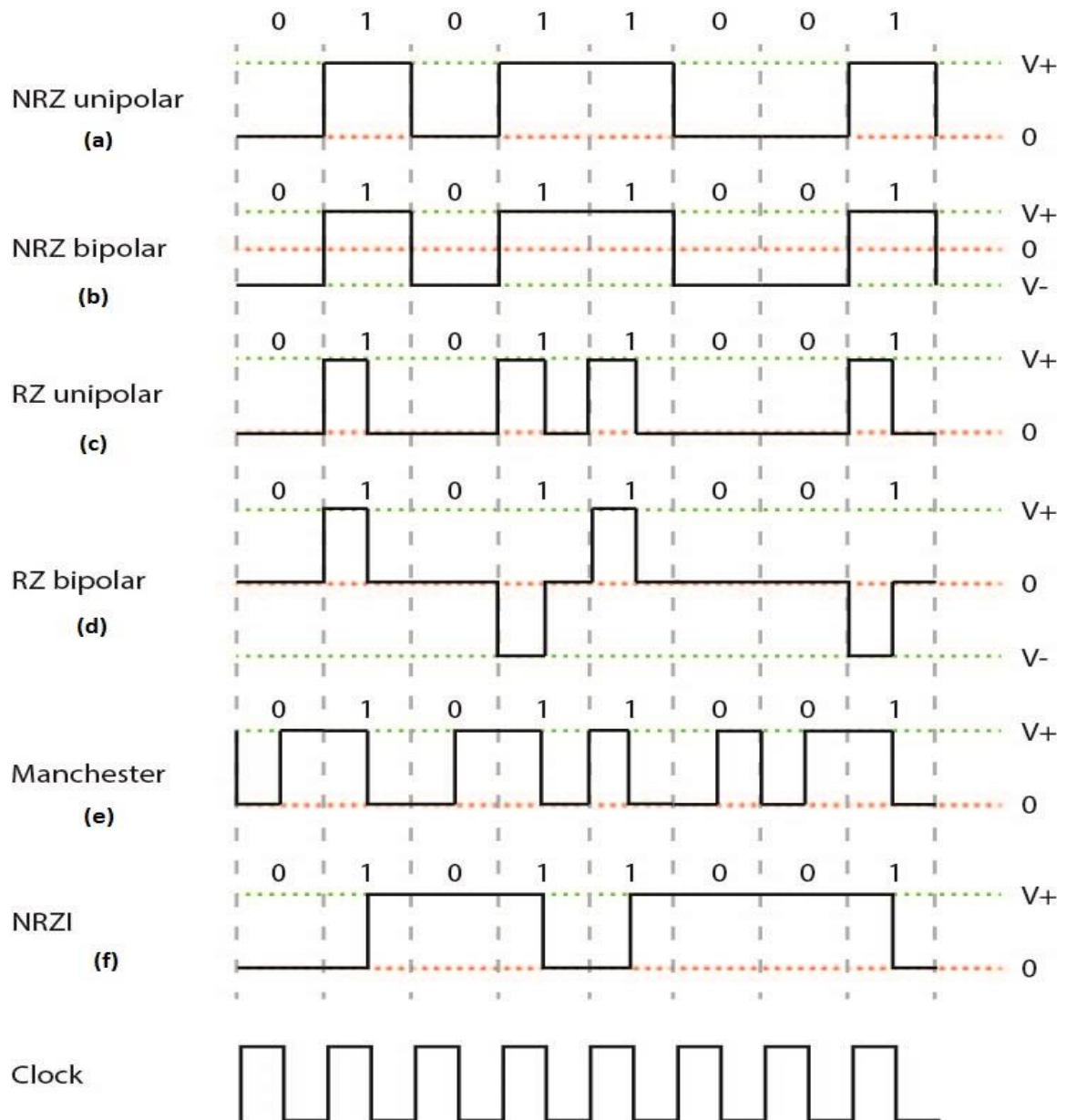
RZ bipolaire est une autre option, comme le montre *la figure (I.5d)*. Une impulsion positive représente un 1 binaire et une impulsion négative représente un 0 binaire. Aucune accumulation de courant continu ne se produit [10].

### I.2.4 Codage biphasé (Manchester) :

Le codage Manchester ou biphasé est une méthode de codage de ligne supplémentaire. Il est généralement bipolaire, mais unipolaire peut également être utilisé. Pour représenter

---

un 1 binaire, une impulsion positive est transmise pendant la moitié du temps de bit et une impulsion négative pour le reste du temps de bit. Un 0 binaire est transmis sous la forme d'une impulsion négative pendant la moitié du temps de bit et d'une impulsion positive pendant l'autre moitié du temps de bit. Notez que cela fournit une transition de tension au centre de chaque bit, ce qui rend la récupération d'horloge très facile [11].



**Figure 1.5 :** Méthodes de codage : (a) NRZ unipolaire, (b) NRZ bipolaire, (c) RZ unipolaire, (d) RZ bipolaire, (e) Manchester, (f) NRZI .

### I.3 Protocoles de communication :

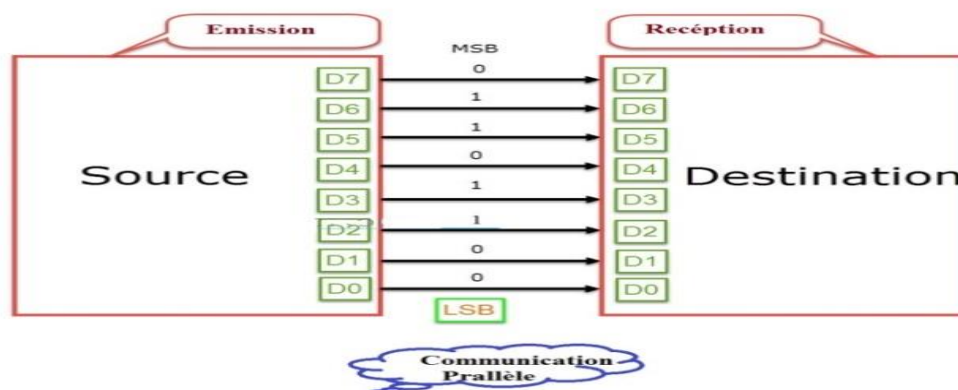
La communication est une terminologie très connue qui implique l'échange d'informations entre deux ou plusieurs supports. Dans les systèmes embarqués, la communication signifie l'échange de données entre deux microcontrôleurs sous forme de bits. Cet échange de bits de données dans le microcontrôleur se fait par un ensemble de règles définies appelées protocoles de communication [12].

D'abord nous commençons par quelques notions de base sur la communication électronique, puis nous expliquerons en détail le fonctionnement de chaque protocole.

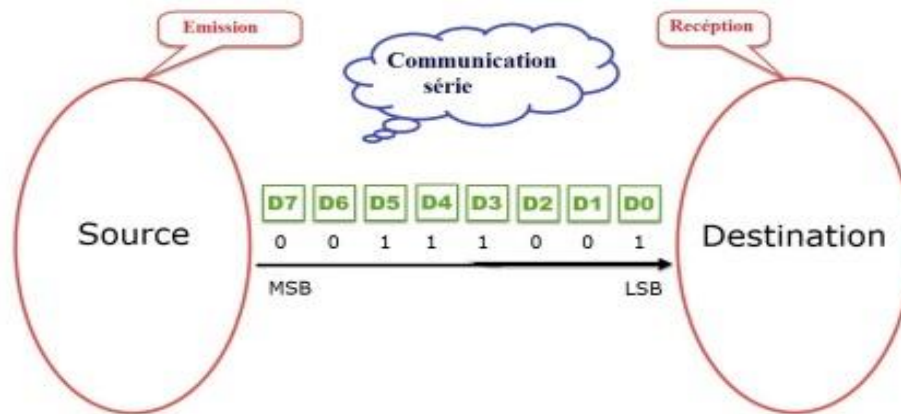
#### I.3.1 Modes de communication :

Jusqu'au début des années 1990, la communication entre les différentes composantes d'un système électronique industriel se limitait à la transmission de quelques octets par un microcontrôleur aux différents actionneurs. Le mode de plus répandu consistait à utiliser un port parallèle de microcontrôleur pour transmettre simultanément les bits d'un octet à l'aide de huit liaisons électriques. Cela nécessite un nombre important de liaisons filaires. Il paraît donc inconcevable de nos jours, vu la complexité des systèmes électroniques modernes [13].

L'utilisation d'une liaison série s'impose. Les données sont transmises les unes après les autres sur une seule et même liaison électrique. Pour transmettre les données sous forme sérielle, il existe deux grandes familles de liaison séries: les liaisons séries synchrones et les liaisons séries asynchrones (*figure I.6*) [13].



(a)



(b)

**Figure I.6 :** (a) Modes de communication parallèle, (b) Modes de communication série.

### I.3.1.1 La liaison séries synchrones :

La communication synchrone nécessite une ligne d'horloge partagée entre l'émetteur et le récepteur. La ligne d'horloge signale la validité des données de la ligne série [14].

### I.3.1.2 La liaison séries Asynchrones :

La communication asynchrone n'a pas de ligne d'horloge. Les données série asynchrones sont généralement envoyées mot par mot. Chacun est accompagné de bits de début et de fin qui définissent le début et la fin d'un mot [10].

### I.3.1.3 Les différents modes d'exploitation d'une liaison Série :

#### a. La liaison simplex :

Dans la méthode simplex, l'émetteur ou le destinataire peut être actif à la fois. Donc, si l'expéditeur transmet les données, le destinataire ne peut accepter que l'inverse. C'est une technique de communication à sens unique [12].

#### b. La liaison bidirectionnelle :

##### ✓ Liaison semi-duplex :

En mode semi-duplex, l'expéditeur et le destinataire peuvent être actifs mais pas simultanément. Donc, si l'expéditeur transmet, le destinataire peut accepter mais ne peut pas

envoyer et vice versa [12].

✓ **Liaison duplex intégral :**

En mode duplex intégral, le récepteur et l'émetteur peuvent s'envoyer des données simultanément [12].

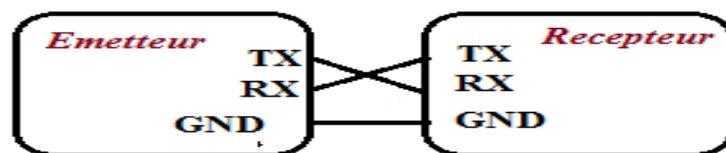
### I.3.2 La liaison point à point EIA-232 :

#### I.3.2.1 Définition :

RS232 est un protocole standard (norme recommandé 232) utilisé pour les liaisons série asynchrone permettant la communication bidirectionnelle entre deux équipement [12].

#### I.3.2.2 Principe de fonctionnement du protocole RS232 :

En RS232, les données sont transmises en série dans une direction sur une seule ligne. Pour établir une communication bidirectionnelle, il faut au moins trois fils (RX, TX et GND) en plus des signaux de commande. Un octet de données peut être transmis à tout moment, à condition que l'octet précédent ait déjà été transmis (*figure I.7*).



*Figure I.7* Protocole RS232.

RS232 suit le protocole de communication asynchrone, donc il utilise des bits de démarrage et d'arrêt pour informer le récepteur du moment où il doit vérifier les données.

Tout d'abord, l'émetteur, envoie un bit de départ au récepteur pour l'informer que la transmission de données commence à partir du bit suivant. Le bit de départ est toujours "0", ce bit permet de synchroniser l'horloge du récepteur.

Les caractères suivants sont des bits de données, le mot transmis peut être suivi ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission.

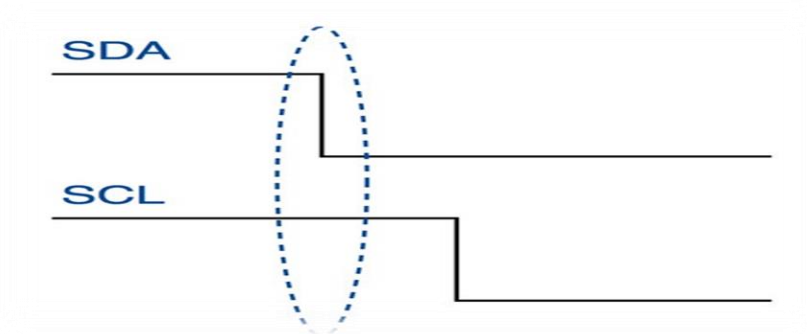


### I.3.3.2 Principe de fonctionnement :

La communication sur le bus I<sup>2</sup>C ne peut se faire qu'entre 2 abonnés. Lorsqu'un abonné prend le contrôle du bus, il devient le maître de la communication. Il génère le signal d'horloge SCL et communique avec un esclave. Selon le sens de la communication, il sera l'émetteur ou le récepteur [17].

#### a) La condition de départ :

- ✓ Un abonné prend le contrôle du bus I<sup>2</sup>C en émettant une condition de départ (*figure I.10*).
  - Niveau haut sur SCL.
  - Front descendant sur SDA.
- ✓ Cet abonné devient le maître.



*Figure I.10* : Bus I<sup>2</sup>C (Condition de départ).

#### b) Transmission de l'adresse :

- ✓ Après avoir pris le contrôle, le maître transmet un octet contenant l'adresse de l'esclave (sur 7 bits) ainsi que l'opération effectuée (écriture NL0 ou lecture NL1).
- ✓ La transmission d'un bit se fait :
  - Lorsque le signal SCL est au niveau haut
  - Lorsque l'esclave a détecté son adresse, il émet un bit d'acquiescement (ACK) au niveau logique bas.

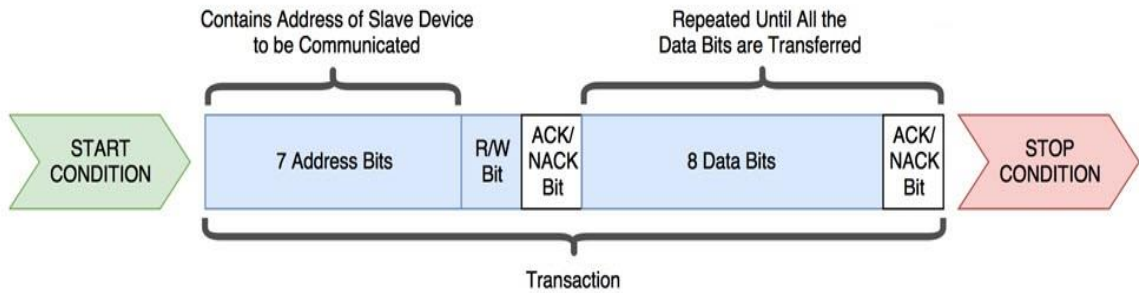
#### c) Transmission des données :

- ✓ Deux cas se présentent :
  - Le maître envoie des données à l'esclave : A la fin de la transmission de chaque



octet, l'esclave émet un acquittement.

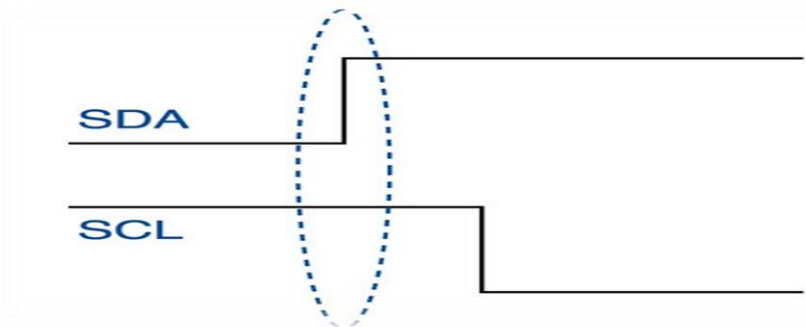
- L'esclave envoie des données au maître : A la fin de la transmission d'un octet, le maître émet un acquittement s'il veut recevoir encore un octet ou bien un non acquittement (NL 1) s'il a terminé de recevoir (*figure I.11*).



*Figure I.11* : Transmission des données sur liaison I2C série.

**d) Fin de la communication :**

- ✓ Pour terminer la communication, le maître émet une condition d'arrêt (*figure I.12*).
  - Niveau haut sur SCL.
  - Front montant sur SDA.



*Figure I.12* : Bus I2C (Condition d'arrêt).

- ✓ Tous les abonnés sont alors déconnectés du bus. SDA et SCL sont au niveau haut.

**I.3.4 L'interface de périphérique série (SPI) :**

**I.3.4.1 Généralités su SPI :**

Les interfaces de périphérique série ou SPI ont été développées par Motorola dans les

années 1980 en tant qu'interface standard, fiable et peu coûteuse entre le microcontrôleur (microcontrôleur de Motorola au début) et ses CI périphériques.

En raison de son interface simple, de sa flexibilité et de sa facilité d'utilisation, SPI est devenu un standard et bientôt d'autres fabricants de semi-conducteurs ont commencé à l'intégrer dans leurs puces.

L'interface de périphérique série (SPI) est un protocole de communication série synchrone qui permet une communication en duplex intégral à très grande vitesse.

Ce protocole est de type maître-esclave couramment utilisé pour l'interfaçage de microprocesseurs ou de microcontrôleurs avec des mémoires telles que EEPROM, RTC (horloge temps réel), ADC (convertisseurs analogique-numérique), DAC (convertisseurs numérique-analogique), des écrans tels que des LCD, des circuits audio, des capteurs tels que la température et la pression, des cartes mémoire telles que des cartes MMC ou SD.

Le bus SPI est composé de 4 signaux ou broches. Elles sont [17] :

- **SCLK (Serial Clock)** : Horloge du bus (produite par le maître).
- **MOSI (Master Out Slave In)** : Données du maître vers l'esclave actif.
- **MISO (Master In Slave Out)** : Données de l'esclave actif vers le maître.
- **SSn (Slave Select n)** : Sélection de l'esclave n à destination de la transmission.

#### **I.3.4.2 Principe de fonctionnement du protocole SPI :**

Pour commencer la communication SPI, le maître doit envoyer le signal d'horloge et sélectionner l'esclave en activant le signal CS. Généralement, la sélection de puce est un signal bas actif; par conséquent, le maître doit envoyer un 0 logique sur ce signal pour sélectionner l'esclave. SPI est une interface en duplex intégral, maître et esclave peuvent envoyer des données en même temps via les lignes MOSI et MISO respectivement. Pendant la communication SPI, les données sont simultanément transmises (sur le bus MOSI) et reçues (sur le bus MISO). Le front d'horloge série synchronise le décalage et l'échantillonnage des données. L'interface SPI offre à l'utilisateur la possibilité de sélectionner le front montant ou descendant de l'horloge pour échantillonner et / ou décaler les données [19].

L'image suivante représente un maître (processeur) connecté à un esclave (périphérique) à l'aide d'un bus SPI.

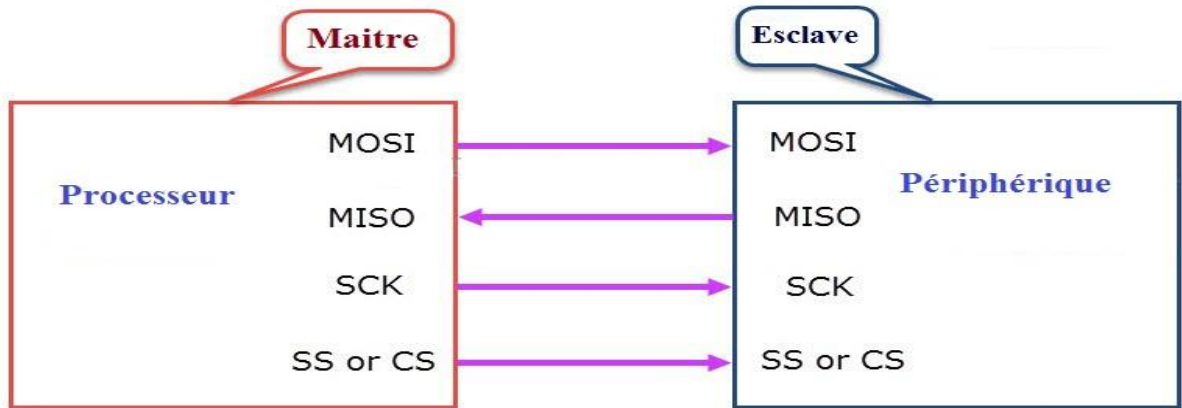


Figure I.13 : Synoptique d'une liaison SPI Maître-Esclave.

Dans le protocole SPI, il ne peut y avoir qu'un seul maître mais plusieurs périphériques esclaves. Les esclaves peuvent être connectés en deux modes.

✓ **En mode normal :**

Si le maître a plusieurs broches de sélection d'esclaves, les esclaves peuvent être câblés en parallèle de la manière suivante

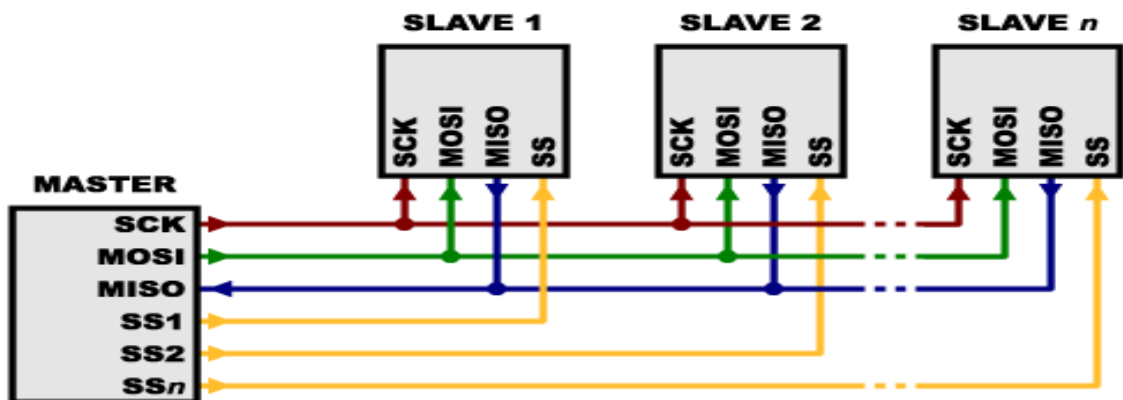


Figure I.14 : Synoptique d'une liaison SPI Maître-Multi-Esclaves en mode normal.

✓ **En mode guirlande :**

Si une seule broche de sélection d'esclave est disponible, les esclaves peuvent être chaînés comme suit:

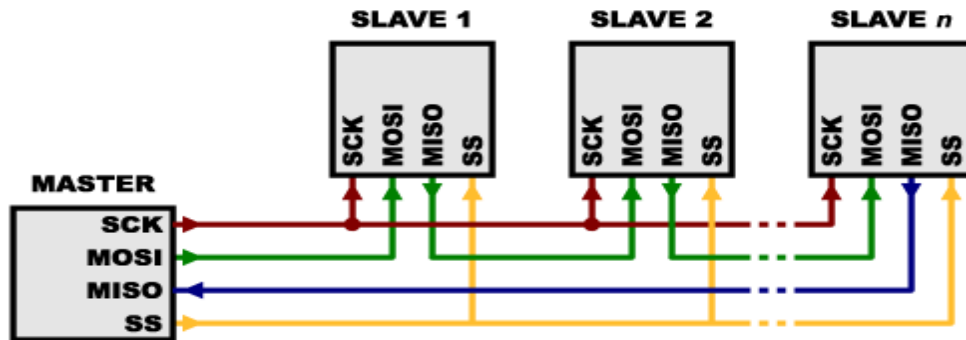


Figure I.15 : Synoptique d'une liaison SPI Maître-Multi-Esclaves en mode guirlande

## **CHAPITRE II**

---

**Etude d'un microcontrôleur avec**

**exploration des modules**

**USART, I2C, PSI, CAN.**

---

Au cours des 40 dernières années environ, nous avons assisté à une révolution silencieuse mais rapide en ce qui concerne l'informatique, l'électronique, la communication et le contrôle.

La preuve en serait donnée dans les combinés mobiles, les lecteurs MP3, les machines à laver automatiques, les fours à micro-ondes, les instruments médicaux de mesure de la pression artérielle ou numériques, le système de facturation dans les cabines téléphoniques STD, etc, pour ne citer que quelques exemples. Cette liste est énorme et augmente de façon exponentielle. Tout cela est possible grâce à l'utilisation de microcontrôleurs, qui ont évolué à partir de microprocesseurs.

Dans ce chapitre nous abordons les microcontrôleurs en général, l'ATmega et la carte Arduino en particulier.

## **II.1 Les microcontrôleurs :**

### **II.1.1 Historique :**

Jusqu'au milieu des années 1980, la plupart des produits électroniques étaient encore construits à l'aide de circuits à logique combinatoire.

À partir du début des années 1980, une minorité de fabricants ont commencé à intégrer des microprocesseurs à leurs produits afin de réduire le nombre de puces, ce qui a permis de réduire les coûts de fabrication et donc les prix à l'utilisateur final.

Les premiers microprocesseurs à 8 bits, tels que l'Intel 8080 ou le Zilog Z80, sont apparus pour la première fois à la fin des années 1970 et constituaient une avancée significative par rapport à ce qui s'était passé auparavant.

Au cours des années 1980, de plus en plus de produits étaient dotés d'un microprocesseur. Même si les microprocesseurs constituaient une énorme amélioration par rapport à ce qu'ils remplaçaient, ils ne constituaient pas une solution miracle pour réduire les coûts et la complexité de la conception des produits. Le problème était que, pour que le microprocesseur fasse quelque chose d'utile, il devait être entouré d'un grand nombre de puces supplémentaires pour les entrées/sorties (E / S) et nécessitait en général d'autres puces

---

de support, telles que des puces d'horloge décodeurs d'adresses.

Dans les années 90, l'amélioration des techniques de traitement du silicium et de fabrication des puces permettait de mettre de plus en plus de circuits sur une puce. Cela a notamment été utilisé pour compléter la puce du microprocesseur avec des fonctions et des fonctionnalités supplémentaires qui étaient auparavant implémentées par des puces externes séparées. Pour différencier ces nouvelles puces super-micros de leurs ancêtres plus simples, on les a appelées microcontrôleurs.

Une fois que les microcontrôleurs ont commencé à être intégrés dans des biens de consommation et des systèmes de contrôle dans les années 90, les miniaturisations électroniques déjà impressionnantes des deux décennies précédentes ont fait un bond en avant, en termes de réduction de taille et de capacité à offrir plus d'options et de fonctionnalités que jamais auparavant.

Au début des années 2000, personne ne songerait sérieusement à concevoir autre chose que le plus simple appareil électrique grand public sans utiliser de microcontrôleur. Ils sont partout; ils deviennent plus capables et plus complexes avec le temps [20].

### II.1.2 Qu'est-ce qu'un microcontrôleur :

Un microcontrôleur est une petite unité de calcul accompagné de mémoire, de ports d'entrée/sortie et de périphériques (des Timers, des convertisseurs analogique-numérique, des liaisons Séries, etc.) permettant d'interagir avec son environnement.

Les microcontrôleurs sont inévitables dans les domaines de l'informatique embarquée, de l'automatique et de l'informatique industrielle [21].

Ils permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés [22].

Le microcontrôleur est composé de quatre parties [23] :

- ✚ **Un microprocesseur** : Qui va prendre en charge la partie traitement des informations et envoyer des ordres. Il est lui-même composé d'une unité arithmétique et logique (UAL) et d'un bus de données. C'est donc lui qui va exécuter le programme embarqué dans le microcontrôleur.

- ✚ **Une mémoire de données (RAM)** : Dans laquelle seront entreposées les données

temporaires nécessaires aux calculs. C'est en fait la mémoire de travail qui est donc volatile.

- ✚ **Une mémoire programmable (ROM) :** Qui va contenir les instructions du programme pilotant l'application à laquelle le microcontrôleur est dédié. Il s'agit ici d'une mémoire non volatile puisque le programme à exécuter est à priori toujours le même.
- ✚ **Les ressources auxiliaires:** Correspond aux ressources auxiliaires. Celles-ci sont généralement :
  - Ports d'entrées / sorties parallèle et série.
  - Timers pour générer ou mesurer des signaux avec une grande précision temporelle.
  - Convertisseurs A/N pour traiter les signaux analogiques.

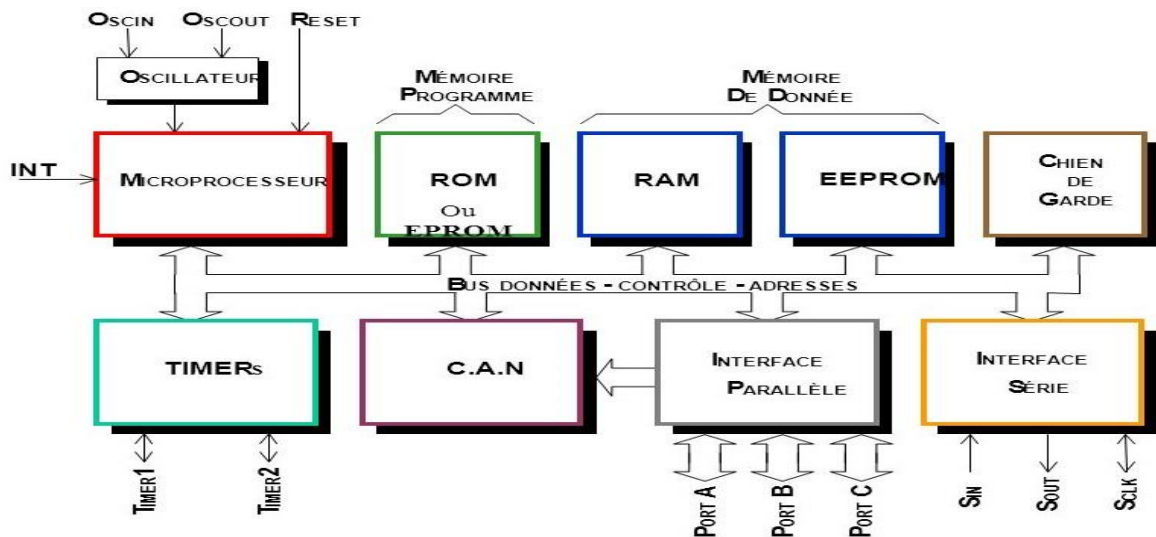


Figure II.1 : Schéma interne d'un microcontrôleur.

### II.1.3 Type des microcontrôleurs :

Il existe un grand nombre de types de microcontrôleurs différents disponibles sur le marché, développés et fabriqués par de nombreuses sociétés.

Les microcontrôleurs sont divisés en différentes catégories en fonction de la mémoire, de l'architecture, des bits et des jeux d'instructions [8] :

- ✚ **Bits :** Sur la base de la configuration des bits, le microcontrôleur est divisé en trois catégories.



- 
- ✓ **Microcontrôleur 8 bits** : Ce type de microcontrôleur est utilisé pour exécuter des opérations arithmétiques et logiques telles que l'addition, la soustraction, la division par multiplication, etc...
  - ✓ **Microcontrôleur 16 bits** : Ce type de microcontrôleur est utilisé pour effectuer des opérations arithmétiques et logiques lorsqu'une précision et des performances supérieures sont requises.
  - ✓ **Microcontrôleur 32 bits** : Ce type de microcontrôleur est généralement utilisé dans des appareils à commande automatique, tels que des machines automatiques, des appareils médicaux, etc....
- ✚ **Mémoire** : Sur la base de la configuration de la mémoire, le microcontrôleur est divisé en deux catégories.
- ✓ **Microcontrôleur de mémoire externe** : Ce type de microcontrôleur est conçu de manière à ne pas avoir de mémoire programme sur la puce. Par conséquent, il est nommé microcontrôleur de mémoire externe.
  - ✓ **Microcontrôleur à mémoire intégrée** : Ce type de microcontrôleur est conçu de manière à ce que le microcontrôleur dispose de tous les programmes et de la mémoire de données, compteurs et temporisateurs, des interruptions et des ports d'entrée / sortie sont intégrés à la puce. Par exemple: microcontrôleur Intel 8051.
- ✚ **Jeu d'instructions** : Sur la base de la configuration du jeu d'instructions, le microcontrôleur est divisé en deux catégories.
- ✓ **CISC** : CISC signifie ordinateur complexe de jeu d'instructions. Il permet à l'utilisateur d'insérer une seule instruction au lieu de nombreuses instructions simples.
  - ✓ **RISC** : RISC signifie ordinateurs à jeu d'instructions réduit. Il réduit le temps de fonctionnement en raccourcissant le cycle d'horloge par instruction.

Ainsi de la famille. Pour une même famille, il peut y avoir différentes versions avec différentes sources [24].

- La famille Atmel AT91 .
- La famille Atmel AVR .
- Le C167 de Siemens .
- La famille Hitachi H8 .

- La famille des PIC de Microchip.
- La famille des DSPIC de Microchip.
- La famille PSOC de Cyprées.
- La famille LPC21xxARM7-TDMI de Philips.

### II.1.4 Domaines d'application des microcontrôleurs :

- ✚ **Système de communication** : Les microcontrôleurs 8 bits sont souvent utilisés pour les téléphones portables simples et la téléphonie fixe alors que les microcontrôleurs 32 bits se retrouvent plutôt dans les Smartphones [25].



*Figure II.2* : Les appareils de communication.

- ✚ **Technique médicale** : Les instruments de mesure (par exemple mesure de la glycémie), les organes artificiels, etc [25].



*Figure II.3* : Appareil médical.

- ✚ **Les technologies de la sécurité** : Les systèmes pour gérer la sécurité dans les moyens de transport (par exemple : les passages à niveau), dans les bâtiments (par exemple: alarme incendie, effractions), etc [25].



*Figure II.4* Appareil de sécurité.

;

- + **Mécatronique et automation industrielle** : Installation pour la production de biens, pour la logistique, etc [25].



*Figure II.5* : Les appareils industriels.

- + **Electronique de consommation**: Appareil hifi, TV, vidéo, télécommande, etc. [25].



*Figure II.6* : Hifi, TV, vidéo.

## II.2 Le microcontrôleur ATmega328 :

### II.2.1 Définition d'un Atmega328 :

L'Atmel Atmega328 est un microcontrôleur à 28 broches et à 8 bits. L'architecture est basée sur le concept RISC (Reduced Instruction Set Computer) qui permet au processeur d'exécuter 20 millions d'instructions par seconde lorsqu'il fonctionne à 20 MHz.

L'Atmega328 est doté d'un large éventail de fonctionnalités. Les fonctionnalités peuvent être classées de manière pratique dans les systèmes suivants: système de mémoire, système portuaire, système de minuterie, convertisseur analogique-numérique (ADC), système d'interruption, et communications série [26].

### II.2.2 Descriptions des broches ATmega328 :

La figure (II.7) montre le schéma de brochage d'un microcontrôleur Atmega328 :

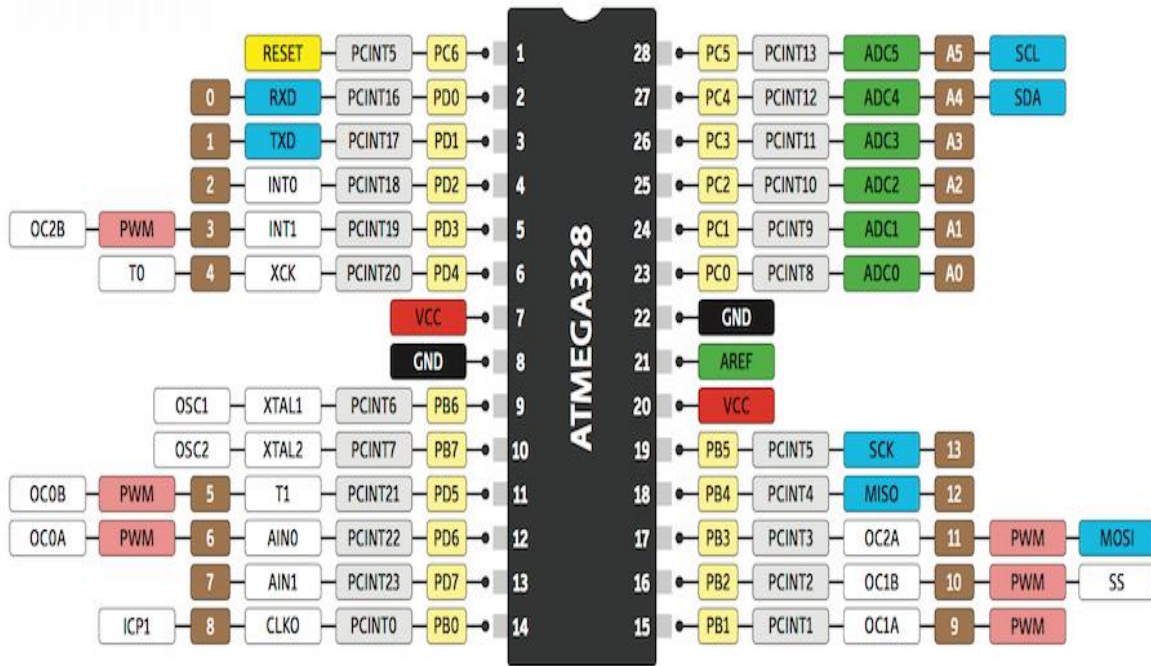


Figure II.7 : Schéma de brochage d'Atmega328.

**A. Digital I/O (entrées-sorties Tout Ou Rien) :**

L'Atmel ATmega328 est équipé de trois ports d'entrée / sortie (E / S) numériques à 8 bits à usage général [26]. La figure (II.8) montre les différents ports de l'Atmega328.

- ✓ **Port B:** est constitué des broches de **PB0** à **PB7**. Ce port est un port bidirectionnel de 8 bits ayant une résistance de rappel interne.
- ✓ **Port C :** est constitué des broches de **PC0** à **PC6**. Les tampons de sortie du port C ont des caractéristiques d'entraînement symétriques avec une capacité source ainsi qu'un taux de chute élevé.
- ✓ **Port D :** est constitué des broches **PD0** à **PD7**. C'est également un port d'entrée / sortie de 8 bits ayant une résistance de rappel interne.

**B. Broche d'alimentation :**

- **VCC :** est une alimentation.
- **AVCC :** est une broche de tension d'alimentation pour convertisseur analogique-numérique.
- **GND :** indique la masse et il a un 0V.
- **AREF :** broche de référence analogique pour convertisseur analogique numérique

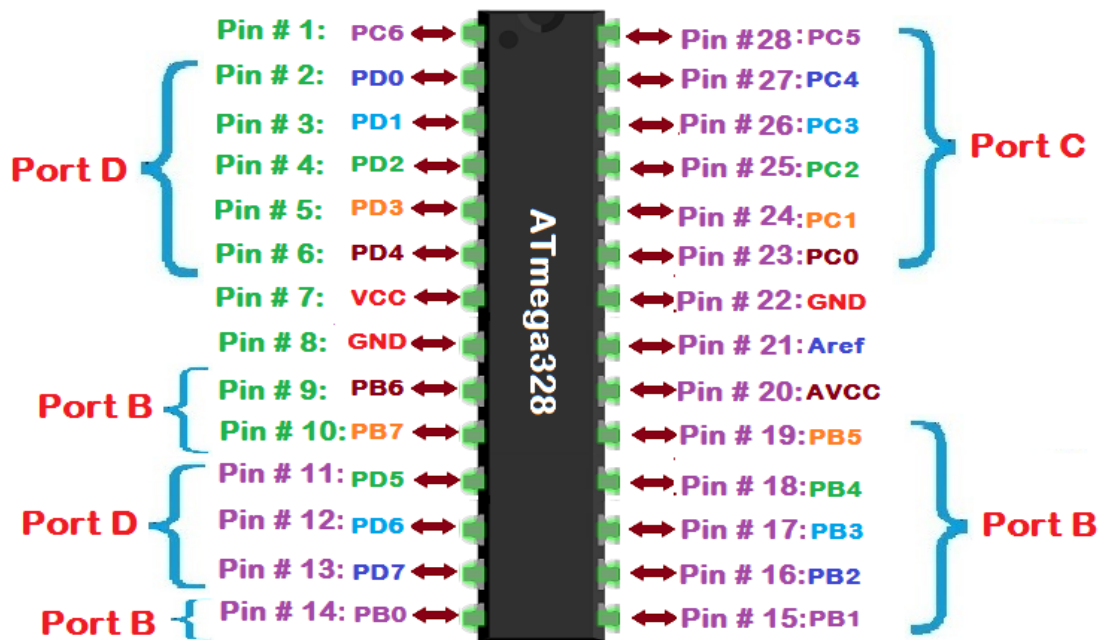


Figure II.8: Ports E/S de l'Atmega328.

**C. Plusieurs broches multi-fonction :** Certaines broches peuvent avoir plusieurs fonctions différentes [27] :

- ✓ **Timers/Counters :** Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16 bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB)
- ✓ **PWM :** 6 broches OC0A(PD6), OC0B(PD5), OC1A(PB1), OC1B(PB3), OC2A(PB3), OC2B(PD3)
- ✓ **Convertisseur analogique-numérique (résolution 10 bits) :** 6 entrées multiplexées ADC0(PC0) à ADC5(PC5)
- ✓ **Gestion bus I2C (TWI Two Wire Interface) :** le bus est exploité via les broches SDA(PC5)/SCL(PC4).
- ✓ **Port série (USART) :** émission/réception série via les broches TXD(PD1)/RXD(PD0)
- ✓ **Comparateur Analogique :** Broches AIN0(PD6) et AIN1 (PD7) .
- ✓ **Interruption Watch dog Timer programmable.**
- ✓ **Gestion d'interruptions :**
  - Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).





---

essentiellement de :

### II.2.3.1 Mémoire ATmega 328 :

L'ATmega328 est équipé de trois sections de mémoire principales [28] :

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

### II.2.3.2 Base de temps ATmega328 :

La vitesse à laquelle un microcontrôleur séquence par ces actions est contrôlé par une base de temps précise appelée **horloge**. La source d'horloge est acheminée dans le microcontrôleur afin de fournir une base de temps pour tous les sous-systèmes périphériques.

L'ATmega328 peut être cadencé en interne à l'aide d'une base de temps d'un condensateur à résistance (RC) pouvant être sélectionné par l'utilisateur, ou il peut être cadencé en externe. La base de temps interne de la télécommande est sélectionnée à l'aide de bits de fusible programmables. Vous pouvez choisir une fréquence de fonctionnement d'horloge interne fixe de 1, 2, 4 ou 8 MHz.

Pour fournir une gamme plus large de sélections de fréquence, une source de temps externe peut être utilisée. Les sources de temps externes, par ordre de précision et de stabilité croissantes, sont un réseau RC externe, un résonateur en céramique ou un oscillateur à cristal. Le concepteur de système choisit la fréquence de base de temps et le dispositif source d'horloge appropriés à l'application considérée [9].

### II.2.3.3 Sous-système de chronométrage ATmega328 :

L'ATmega328 est équipé d'un ensemble de **temporisateurs** qui permettent à l'utilisateur de générer un signal de sortie de précision, de mesurer les caractéristiques (période, rapport

---

cyclique, fréquence) d'un signal numérique entrant ou de compter des événements externes. Plus précisément, l'ATmega328 est équipé de deux compteurs / compteurs 8 bits et d'un compteur 16 bits [9].

#### II.2.3.4 Canaux de modulation de largeur d'impulsion :

Un signal PWM modulé en largeur ou PWM est caractérisé par une fréquence fixe et un facteur de marche variable. Le cycle de service est le pourcentage de temps pendant lequel un signal répétitif atteint un niveau logique haut pendant la période du signal.

L'ATmega328 est équipé de quatre canaux de modulation de largeur d'impulsion (PWM). Les canaux PWM, combinés à la possibilité de diviser la base de temps en différentes fréquences source d'horloge de sous-système PWM, permettent à l'utilisateur de générer une grande variété de signaux PWM: des signaux à faible facteur de fonctionnement à fréquence relativement élevée aux signaux à facteur d'utilisation élevé et à fréquence relativement basse.

Les signaux PWM sont utilisés dans une grande variété d'applications, notamment le contrôle de la position d'un servomoteur et le contrôle de la vitesse d'un moteur à courant continu [9].

#### II.2.3.5 Interruptions ATmega328

L'exécution normale d'un programme suit une séquence d'instructions désignée. Cependant, cette séquence d'événements normale doit parfois être interrompue pour répondre aux défauts et aux états hautement prioritaires à l'intérieur et à l'extérieur du microcontrôleur. Lorsque ces événements de priorité plus élevée se produisent, le microcontrôleur doit suspendre temporairement le fonctionnement normal et exécuter des actions spécifiques à l'événement, appelées une routine de service d'interruption. Une fois que l'événement de priorité supérieure a été traité, le microcontrôleur revient et continue le traitement du programme normal.

L'ATmega328 est équipé d'un complément de 26 sources d'interruption. Deux des interruptions sont fournies pour des sources d'interruption externes, tandis que les interruptions restantes prennent en charge le fonctionnement efficace des sous-systèmes périphériques du microcontrôleur [9].



### II.2.3.6 Communications série ATmega328

L'ATmega328 est équipé d'une multitude de sous-systèmes de communication série, notamment l'USART série, l'interface de périphérique série ou SPI et l'interface série à deux fils (TWI).

#### II.2.3.6.1 ATmega328 série USART :

L'ATmega328 est doté d'un matériel indépendant pour l'émetteur et le récepteur appelé USART acronyme pour *Universal synchronous Asynchronous Receiver Transmitter* est généralement utilisé pour les communications asynchrones. Cette fonction peut être utilisée pour implémenter une interface RS-232[29].

Pour maintenir la synchronisation entre l'émetteur et le récepteur, les bits de début et de fin de tram sont utilisés au début et à la fin de chaque octet de données dans une séquence de transmission. La figure suivante donne le schéma fonctionnel d'un module USART.

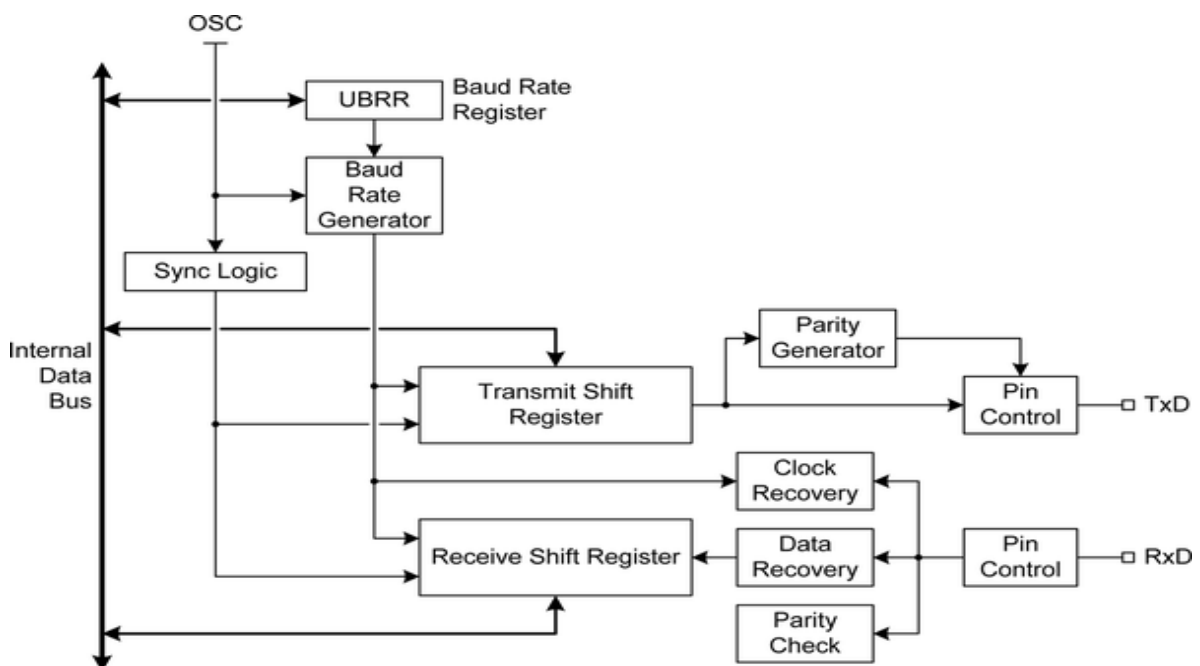


Figure II.10 : Schéma fonctionnel d'un module USART.

L'ATmega328 USART est assez flexible. Il peut être réglé sur divers débits de transmission de données, appelés débits en bauds (bits par seconde). L'USART peut également être défini pour des largeurs de bits de données de 5 à 9 bits avec un ou deux bits d'arrêt. En outre, l'ATmega328 est équipé d'un bit de parité généré par le matériel (pair ou

impair) et d'un matériel de contrôle de parité au niveau du récepteur. Un seul bit de parité permet de détecter une erreur d'un seul bit dans un octet de données. L'USART peut également être configuré pour fonctionner en mode synchrone [9].

### II.2.3.6.2 Interface de périphérique série ATmega328 – SPI :

La logique périphérique SPI de l'ATmega prend en charge les quatre modes de fonctionnement SPI standard. Les broches d'E / S sur le périphérique ATmega328 peuvent être configurées pour agir en tant que signaux MOSI, MISO et SCK utilisés par SPI [29].

Le SPI peut être considéré comme un registre à décalage synchrone de 16 bits avec une moitié de 8 bits résidant dans l'émetteur et l'autre moitié de 8 bits résidant dans le récepteur. L'émetteur est désigné comme maître puisqu'il fournit la source d'horloge de synchronisation entre l'émetteur et le récepteur. Le récepteur est désigné comme esclave [9]. La figure suivante montre le schéma fonctionnel du module SPI.

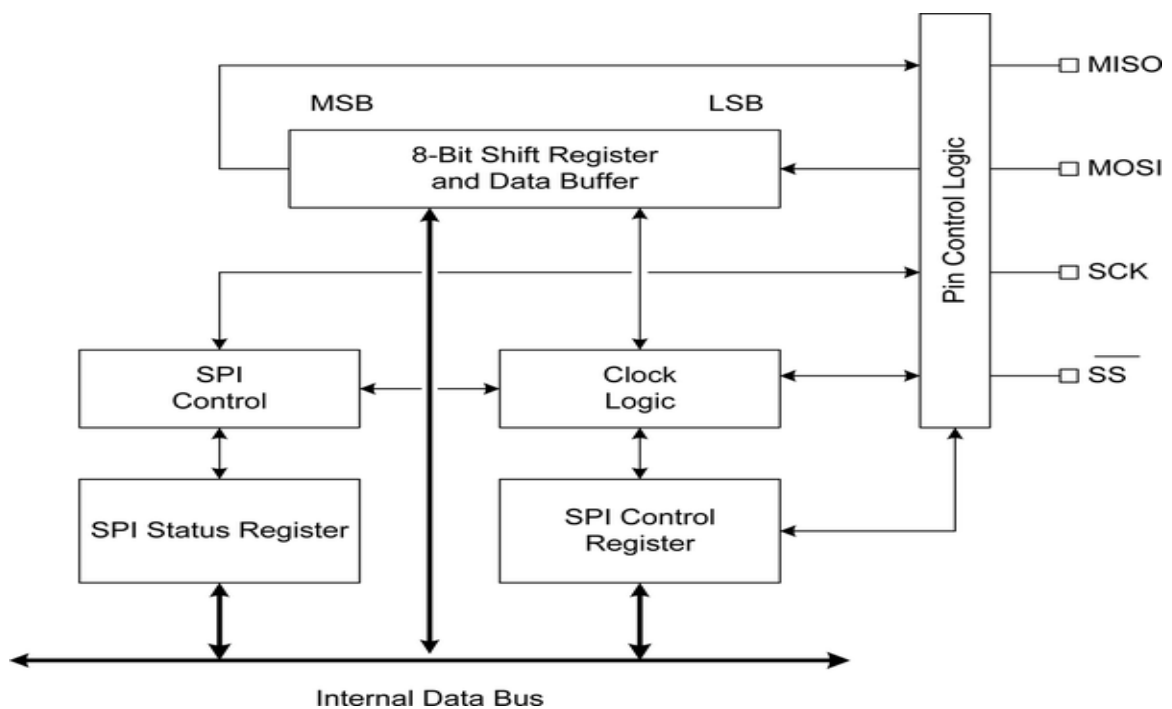


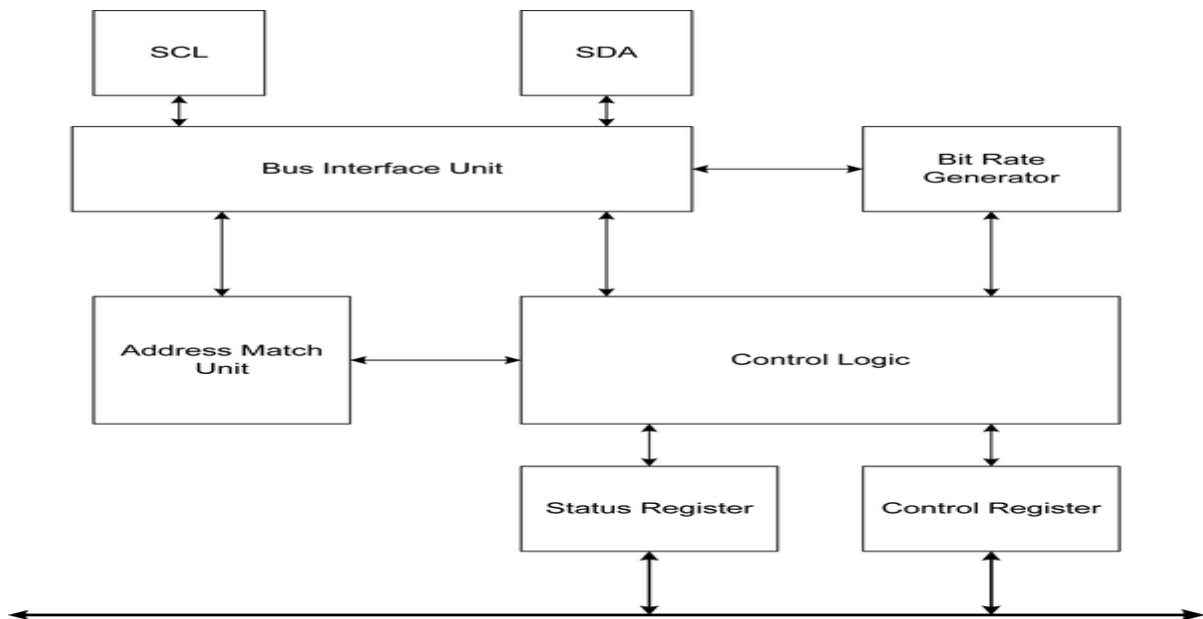
Figure II.11 : Schéma fonctionnel du module SPI.

### II.2.3.6.3 Interface série à deux fils ATmega328 – TWI :

La troisième forme d'E / S série prise en charge par de nombreux périphériques AVR telle que l'ATmega328 est l'interface à deux fils (TWI). Cette interface est compatible avec le protocole I2C. Il prend en charge les modes de fonctionnement maître et esclave, ainsi qu'une

adresse de périphérique 7 bits.

Avec un arbitrage de bus multi-maître l'interface TWI peut générer un état de réveil lorsque l'ATmega328 est en mode veille. En interne, le périphérique TWI est assez complexe, bien plus que les périphériques USART ou SPI [29]. La figure (II-12) présente un aperçu de l'interface TWI :



*Figure II.12* : Schéma fonctionnel du module TWI.

#### II.2.3.6.4 Convertisseur analogique-numérique ATmega328 – CAN :

Une grande partie du monde réel est analogique telle que les tensions, les courants, les niveaux de lumière, les forces, etc...., et les microcontrôleurs permettent la connexion du monde réel au monde du code qui est numérique .

Ce passage du monde analogique au numérique est le travail du matériel du *convertisseur analogique-numérique* (ADC). À l'aide du CAN intégré, l'Atmega328 peut prendre des mesures (de tension) à partir de capteurs analogiques et les transformer en nombres pour les utiliser dans les codes [30].

La figure (II.13) présente tous les composants principaux du périphérique ADC.

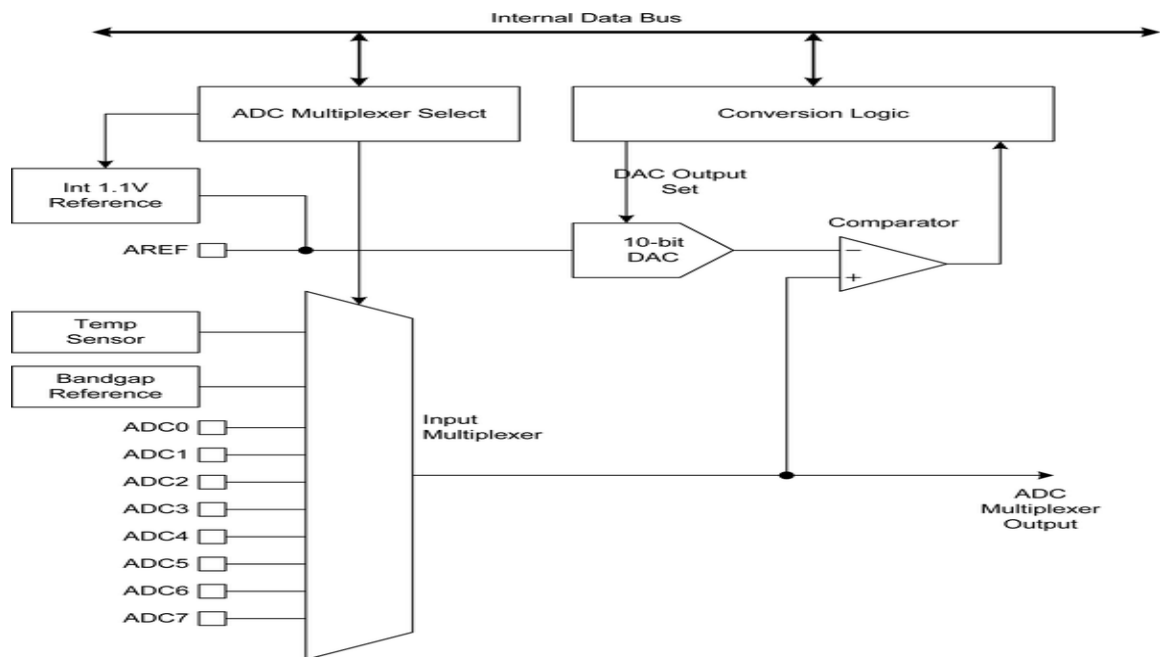


Figure III.13 Module ADC.

### II.3 Arduino :

Le système Arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, pour programmer des systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique [27].

L'univers Arduino repose sur deux piliers, le premier s'agit de la carte électronique programmable (Hardware), composée de plusieurs composants semi-conducteurs, de circuits intégrés et des périphériques, le deuxième s'agit de l'interface de programmation (Software), qui possède un langage de programmation très spécifique, basé sur les langages C et C++, adapté aux possibilités de la carte [31].



Figure II.14 : Logo Arduino.

En fait, il existe une variété de cartes Arduino, mais la plus commune est l'Arduino UNO, qui est décrit ici :

### II.3.1 Constitution d'un Arduino UNO :

Arduino UNO est un circuit imprimé en matériel libre sur lequel se trouve un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits, de manière à effectuer des tâches très diverses comme la domotique (le contrôle des appareils domestiques - éclairage, chauffage...), le pilotage d'un robot, etc [28] [31].

#### II.3.1.1 Partie matérielle :

La figure suivante montre la constitution d'une carte Arduino UNO :

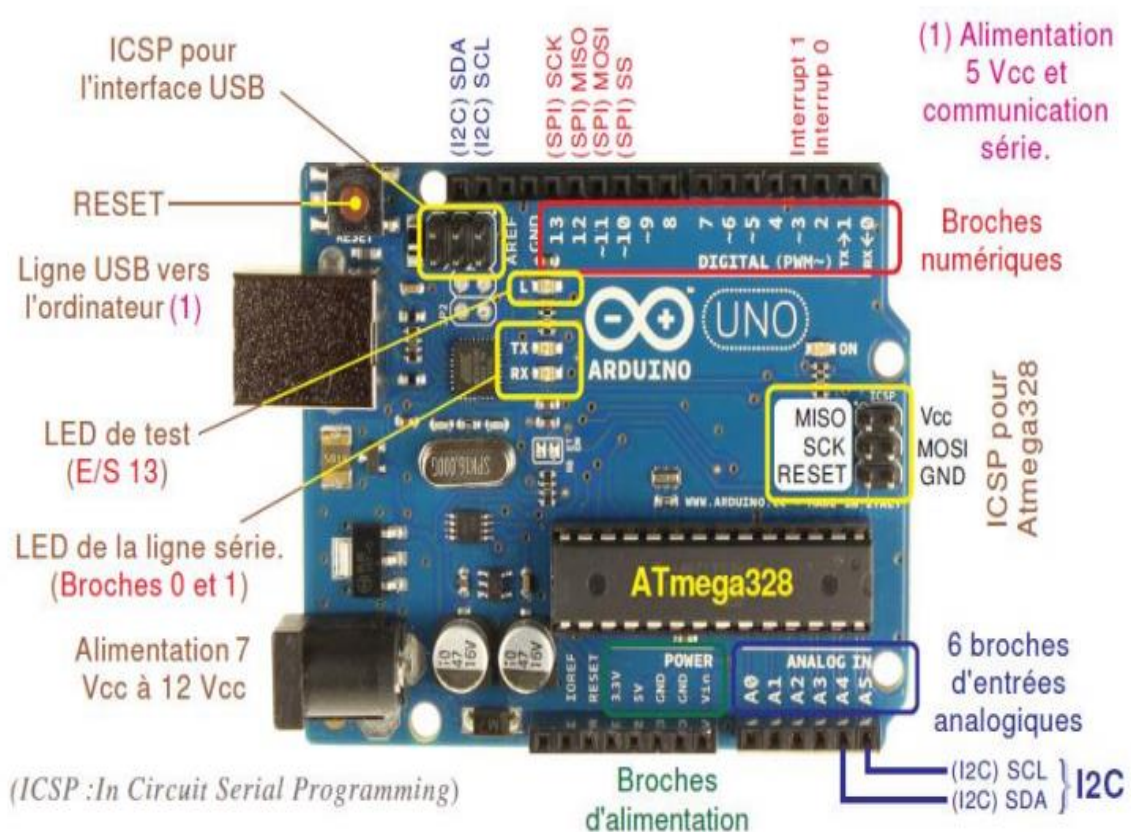


Figure II.15: Constitution de la carte Arduino UNO.

### A. Le Microcontrôleur ATmega328 :

Dans une carte Arduino UNO il y a un boîtier noir rectangulaire avec 28 "pattes" cette puce est le ATmega328, le cœur de ce board [32]. La figure suivante montre le brochage d'Arduino UNO .

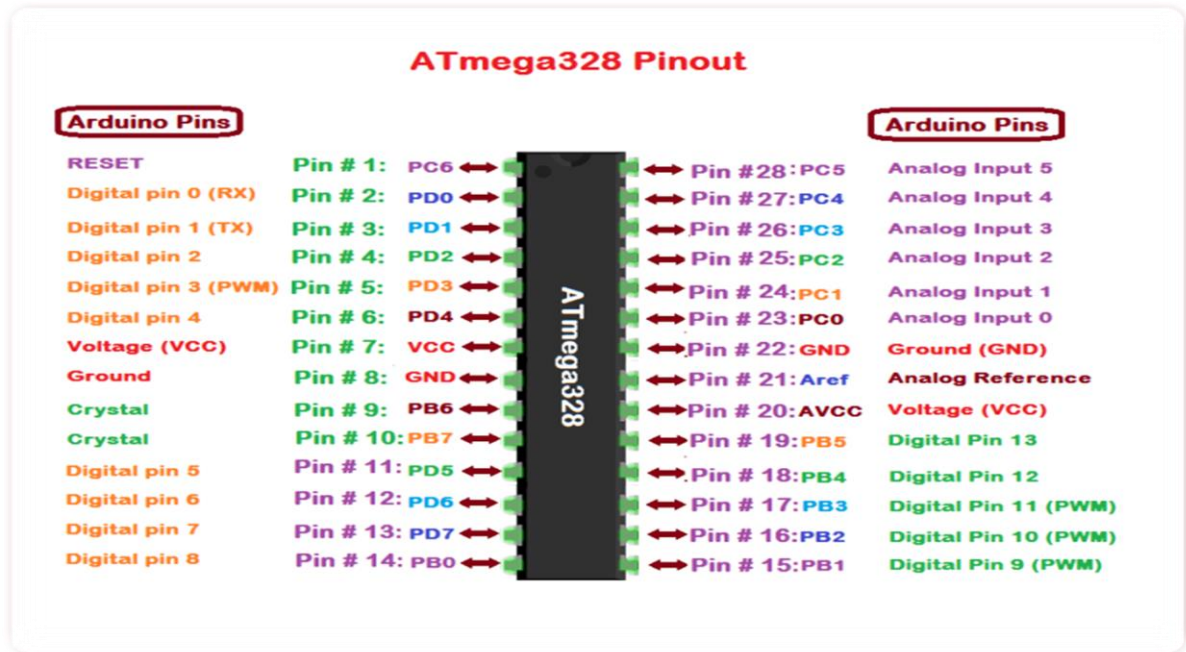


Figure II.16 : Brochage d'Arduino UNO.

### B. Les sources d'alimentation de la carte :

Elle comporte 6 broches d'alimentation [31] :

- **Une connexion Reset** : pour la réinitialisation.
- **Une connexion 3V3** : qui permet à des circuits de puissance compatible de se connecter à la carte Arduino.
- **Une connexion 5V** : fournit par le régulateur de la carte, cette tension est utile pour d'autres circuits électriques compatibles avec 5 volts.
- **Une connexion GND** : pour la masse.
- **Une connexion Vin** : renvoie la tension appliquée à partir de la prise d'alimentation et peut être utilisée pour alimenter d'autres circuits qui ont déjà un régulateur de tension.

### C. Les entrées & sorties :

L'Arduino a une rangée de broches en haut et en bas avec beaucoup d'étiquettes. Ces broches sont les connecteurs, qui sont utilisés pour attacher à capteurs et actionneurs [32].

---

### ✚ 14 broches d'entrée/sorties numériques (broches 0 à 13) :

Ces broches peuvent être des entrées ou des sorties. Les entrées servent à lire les informations des capteurs, tandis que les sorties servent à contrôler les actionneurs. Nous allons spécifier la direction (entrée ou sortie) dans l'esquisse que nous créons dans l'EDI.

Les entrées numériques ne peuvent lire que l'une des deux valeurs et les sorties numériques ne peuvent émettre que l'une des deux valeurs (HIGH et LOW) [32].

En plus, certaines broches ont des fonctions spécialisées [33] :

- **Interruptions Externes** (Broches 2 et 3) : Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **Impulsion PWM** (Broches 3, 5, 6, 9, 10, et 11) : Fournissent une impulsion PWM 8-bits à l'aide de l'instruction « **Analog Write ( )** ».
- **Interface Série Périphérique SPI** (Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK)) : Ces broches supportent la communication SPI disponible avec la librairie pour communication SPI.

Le bus SPI sur l'Arduino est une exception par rapport à la plupart des autres ports. Sur certains Arduinos, le bus SPI est présent comme en-tête ICSP.

- **Interface TWI** : (Broches 4 (SDA) et 5 (SCL)) : Supportent les communications de protocole I2C ou interface TWI, disponible en utilisant la librairie Wire/I2C
- **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.

### ✚ 6 broches d'entrée analogique (broches 0–5) :

Les broches d'entrée analogiques sont utilisées pour lire les mesures de tension des capteurs analogiques. Contrairement aux entrées numériques, qui ne peuvent distinguer que deux niveaux différents (HIGH et LOW), les entrées analogiques peuvent mesurer 1 024



niveaux de tension différents [32].

#### D. Les ports de communications :

Les ports série sont extrêmement simples. Cette simplicité est l'une des raisons pour lesquelles ils sont utilisés si souvent. Les données sont envoyées sur un fil, le fil de transmission (TX), et reçues sur un autre, le fil de réception (RX).

De l'autre côté du câble, il est connecté à un autre ordinateur avec une broche TX et une broche RX. À l'intérieur du câble lui-même, les fils TX et RX sont inversés. La broche TX d'un côté est connectée à la broche RX de l'autre côté (*figure II.17*) [33].



*Figure II.17:* Ports de communication.

#### II.3.1.2 Partie logicielle:

##### II.3.1.2.1 Présentation du logiciel Arduino

Le logiciel de programmation des modules Arduino est une application multiplateformes (compatible Windows, Linux et Mac), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module). La figure (II.18) montre la fenêtre du logiciel Arduino.



*Figure II.18 :* La fenêtre du logiciel Arduino.





# Chapitre III :

---

**Réalisation pratique des platines d'émission  
et de réception des données numériques issues  
des capteurs et visualiser sur LCD.**

Dans ce dernier chapitre, nous allons commencer par définir la communication sans fils avec introduction de quelque protocoles, puis nous abordons tous les outils de réalisation qu'on a utilisée dans notre projet, à la fin nous présentons deux système de communication sans fils, le premier entre un Arduino et androïde, le second entre deux Arduino.

### **III.1 Communication sans fil :**

#### **III.1.1 Définition :**

La communication sans fil est la méthode de transmission d'informations d'un point à un autre, sans utiliser aucune connexion telle que des fils, des câbles ou un support physique quelconque. L'émetteur et le récepteur peuvent être placés entre quelques mètres (comme une télécommande TV) ou bien entre quelques milliers de kilomètres (communication par satellite).

Le principal et important avantage de la communication sans fil est la mobilité. Outre la mobilité, la communication sans fil offre également souplesse et facilité d'utilisation, ce qui la rend de plus en plus populaire au jour le jour [35].

Différents types de signaux sont utilisés dans la communication entre les dispositifs pour la transmission sans fil de données :

- Transmission par radiofréquence.
- Transmission infrarouge.
- Transmission par micro-ondes.
- Transmission par ondes lumineuses [36].

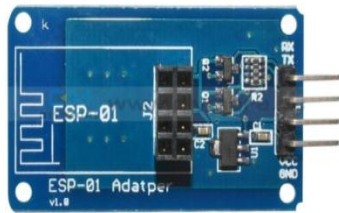
#### **III.1.2 Protocoles de communication sans fil :**

Les ondes radio sont employées par de multiples protocoles on cite quelques-uns :

### III.1.2.1 WIFI :

Le **WIFI** est une technologie basée sur la série de normes IEEE 802,11 qui utilise des fréquences radio de 2,4GHZ et 5GHZ pour déployer des réseaux locaux (LAN),il a une portée d'environ 50m. Il existe des modules wifi compatibles avec Arduino :

- Le module *Shield Arduino Wifi* permet de connecter une carte Arduino à un réseau internet sans fil Wifi [37].



*Figure III.1* : Module WIFI ESP8266



*Figure III.2* : Logo WIFI

### III.1.2.2 ZigBee :

**ZigBee** est un protocole de réseau local (LAN) de 2,4 GHz. Il a été développé en tant que spécification basée sur l'IEEE 802.15.4 pour une suite de protocoles de communication de haut niveau utilisés pour créer des réseaux personnels avec de petites radios numériques à faible puissance, il offre un débit de 250kbps maximum sur une fréquence de 2.4 GHZ[37].



*Figure III.3* : Module XBee S2TH



*Figure III.4* : Logo ZigBee

### III.1.2.3 Bluetooth :

Bluetooth est une technologie de réseau personnel sans fils (WPAN) à faible portée. Elle a été originellement mise au point par Ericsson en 1994 [38].

C'est une norme de communications permettant l'échange bidirectionnel de données entre des appareils sans liaison filaire en utilisant des ondes radio UHF sur une bande de

fréquence de 2,4 GHz [39].



*Figure III.5* : Logo Bluetooth

## III.2 Outils de conception et réalisation :

### III.2.1 Modules Bluetooth :

Il existe deux sortes de module Bluetooth, tous deux compatibles avec Arduino et utilisables sur une Breadboard. Ces deux modules utilisent le RS232 comme protocole de liaison série avec Arduino. On les distingue par le nombre de pattes d'entrées / sorties :

#### III.2.1.1 Module HC05 :

Le module **Bluetooth HC-05** est un module Bluetooth facile à utiliser, conçu pour la configuration transparente d'une connexion série sans fil. Le module Bluetooth HC-05 est un module Maître/ Esclave.

Les modules esclaves ne peuvent pas établir de connexion à un autre périphérique Bluetooth, mais peut accepter les connexions. Le module maître peut établir une connexion avec d'autres périphériques [40].



*Figure III.6* : Module Bluetooth HC-05.

Le module Bluetooth HC-05 a 6 broches[41] :

Broche	Description
<b>Key/EN</b>	Il est utilisé pour amener le module Bluetooth en mode de commandes AT. Si la touche Key / EN est définie sur High, ce module fonctionnera en mode commande. Sinon, par défaut, il est en mode données.
<b>VCC</b>	Connectez 5 V ou 3,3 V à cette broche.
<b>GND</b>	Broche de masse du module.
<b>TXD</b>	Transmission de données série (les données reçues sans fil par le module Bluetooth sont transmises en série sur la broche TXD)
<b>RXD</b>	Recevez les données en série (les données reçues seront transmises sans fil par le module Bluetooth).
<b>Etat</b>	Indique si le module est connecté ou non

*Tableau III.1* : Description des broches Bluetooth HC-05.

### III.2.1.2 Module HC06 :

Le module **Bluetooth HC-06** est un module "esclave" contrairement au module HC-05 qui est "maître".

Le module Bluetooth HC-06 permet d'établir une liaison Bluetooth (liaison série) entre une carte Arduino et un autre équipement possédant une connexion Bluetooth (Smartphone, tablette, seconde carte Arduino, etc [42]).

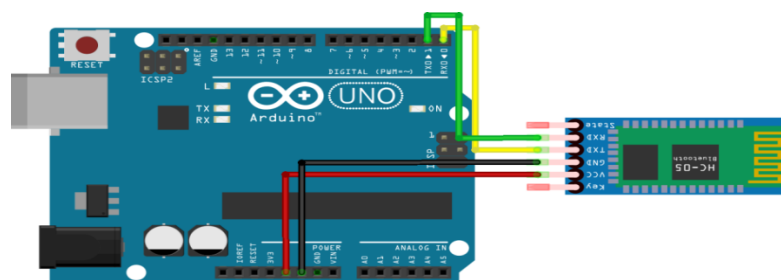


*Figure III.7* : Module Bluetooth HC-06.

Le module Bluetooth HC-06 n'a que 4 broches [41]:

Broche	Description
VCC	Connectez 5 V ou 3,3 V à cette broche.
GND	Broche de masse du module.
TXD	Transmission de données série (les données reçues sans fil par le module Bluetooth sont transmises en série sur la broche TXD)
RXD	Recevez les données en série (les données reçues seront transmises sans fil par le module Bluetooth).

*Tableau III.2* : Description des broches Bluetooth HC-06.



*Figure III.8* : Exemple d'interfaçage d'un module Bluetooth avec la carte Arduino.

### III.2.2 Les capteurs :

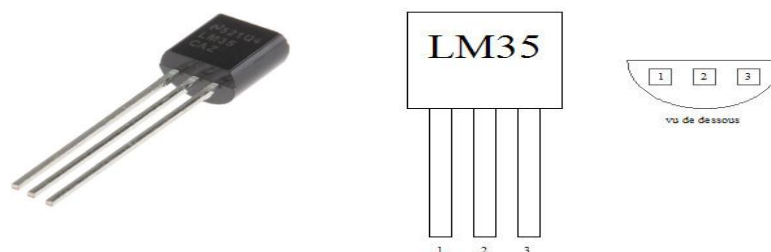
Un capteur est une interface entre un processus physique et une information manipulable. Il ne mesure rien, mais fournit une information en fonction de la sollicitation à laquelle il est soumis. Il fournit cette information grâce à une électronique à laquelle il est associé [33].

Dans notre manipulation, nous allons apprendre à interfacier notre carte Arduino avec différents capteurs. Nous allons présenter les capteurs suivants :

#### III.2.2.1 Capteur de température LM35 :

Les capteurs de température série LM35 sont des dispositifs de mesure de température à circuit intégré de précision avec une tension de sortie linéairement proportionnelle à la température en centigrades.

Le capteur analogique de température, est capable de mesurer des températures allant de  $[-55^{\circ}\text{C}$  à  $+150^{\circ}\text{C}$ ]. La sortie analogique du capteur est proportionnelle à la température. Il suffit de mesurer la tension en sortie du capteur pour en déduire la température. Chaque degré Celsius correspond à une tension de  $+10\text{mV}$  [7].



*Figure III.9:* Capteur de température LM35

#### a. Caractéristiques et spécifications du capteur :

- Calibré directement en Celsius (Celsius).
- Facteur d'échelle linéaire  $+ 10 \text{ mV} / ^{\circ} \text{C}$ .
- Précision assurée à  $0,5 ^{\circ} \text{C}$  (à  $25 ^{\circ} \text{C}$ ).



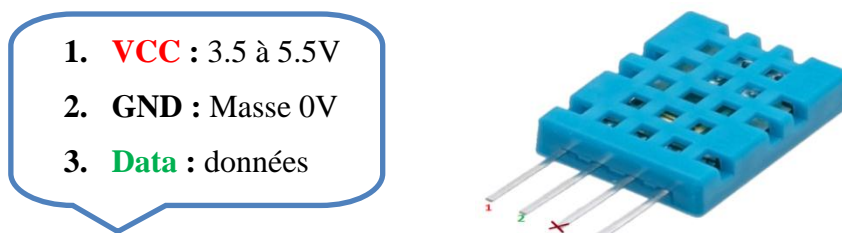


Température précision	de 0 à 50°C +/- 2°C
Humidité précision	de 20 à 96%RH précision +/- 5% RH

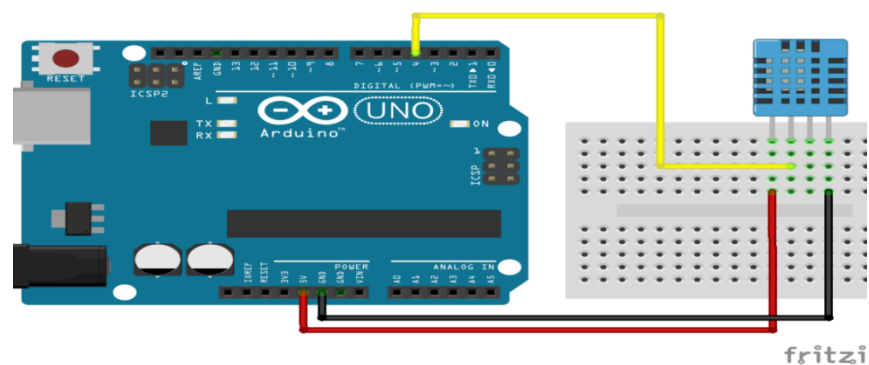
*Tableau III.3* : Caractéristique du capteur DHT11.

### b. Brochage

DHT11 possède 4 broches espacées de **2,45mm** ce qui permet de le brancher facilement sur une breadboard .Seul trois broches sont utiles (figure III.12):



*Figure III.12* : Brochage d'un capteur d'humidité DHT11.



*Figure III.13* : Exemple de liaison d'un capteur DHT11 avec Arduino.

## III.2.2.3 Capteur de distance à ultrason HC-SR04 :

Le capteur HC-SR04 utilise les ultrasons pour déterminer la distance d'un objet. Il offre une excellente plage de détection sans contact, avec des mesures de haute précision et stables [7].



*Figure III.14* : capteur ultrasonique

Le soleil et les matériaux noirs n'affectent pas l'opération, bien que les matériaux mous comme le tissu puissent être difficiles à détecter acoustiquement. Il vient complet avec module émetteur et récepteur à ultrasons [7].

a. **Caractéristiques et spécifications techniques du capteur :**

Caractéristique	Valeur
Alimentation	+5V DC
Courant de repos	<2mA
Courant de travail	15mA
Angle effectif	<15 °
Distance	2cm - 400 cm
Angle de mesure	30 degrés

*Figure III.4* : Caractéristique du capteur 'ultrason.

**b. Brochage :**

Le module ultrasonique HC-SR04 dispose de 4 broches, Terre, VCC, Trig et Echo. Les broches de masse et VCC du module doivent être connectées à la masse et aux broches de 5 volts de la carte Arduino, ainsi qu'aux broches Trig et Echo de toute broche d'E / S numérique de la carte Arduino [44].

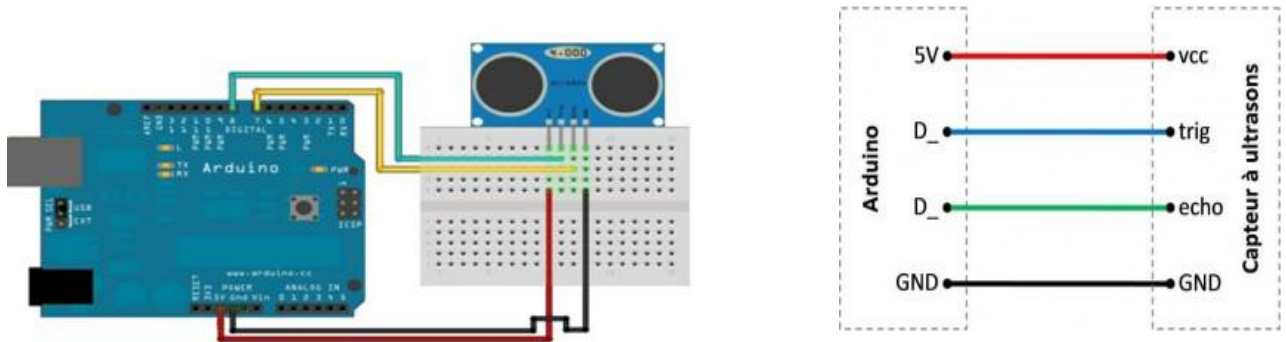


Figure III.15 : exemple d'interfaçage du capteur ultrasons HC-SR04 avec la carte Arduino.

**c. Principe de fonctionnement :**

Pour déclencher une mesure, il faut présenter une impulsion « high » (5v) d'au moins 10  $\mu$ s sur l'entrée « Trig ». Le capteur émet alors une série de 8 impulsions ultrasoniques à 40KHZ, puis il attend le signal réfléchi. Lorsque celui-ci est détecté, il envoie un signal « High » sur la sortie « Echo », dont la durée est proportionnelle à la distance [31].

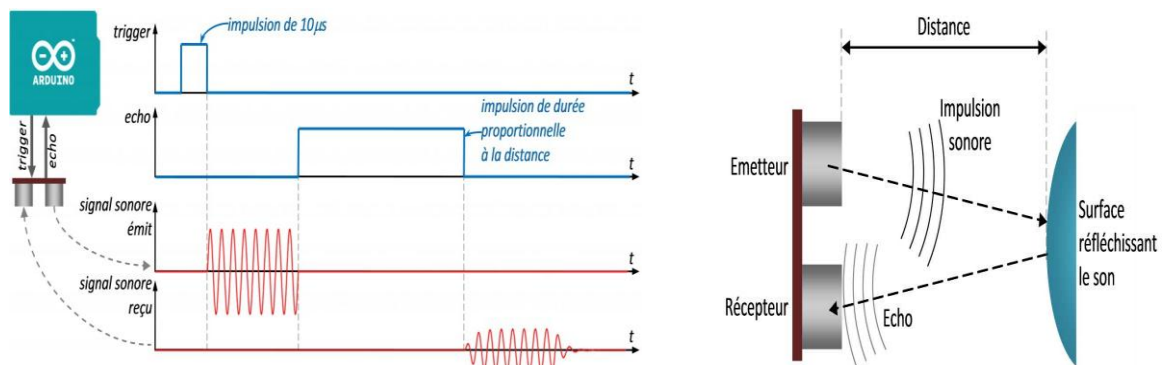


Figure III.16 : Signal d'entrée et sortie du capteur HC-SR04.

### III.2.2.4 Capteur de lumière (photorésistance) :

C'est un composant qui à l'image d'une résistance, nous renvoi une valeur en ohms. Si la lumière incidente est de fréquence suffisamment élevée, elle transportera une énergie importante.

Au contraire par manque de lumière elle transportera peu d'énergie. La résistance d'une photorésistance dépend donc de la luminosité ambiante. On va donc pouvoir détecter les variations de luminosité dans une pièce par exemple ou même à l'extérieur [45].



Figure III.17 : Photorésistance.

### III.2.3 Servomoteur :

Les servomoteurs servent en principe à actionner les parties mobiles d'un système. Ils sont prévus pour être commandés facilement en position ou en vitesse.

Un servomoteur se pilote par l'intermédiaire d'un câble à 3 fils. Ce câble permet à la fois de l'alimenter et de lui transmettre des consignes de position par le fil de signal :

- Le noir ou marron : La masse.
- Le rouge : La tension d'alimentation continue (+).
- Le jaune, orange ou blanc : Le signal de commande PWM [44].

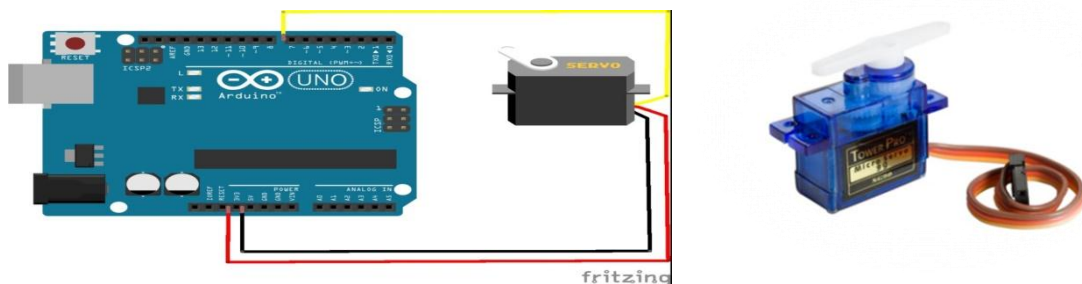


Figure III.18 : Brochage d'un servomoteur avec Arduino.

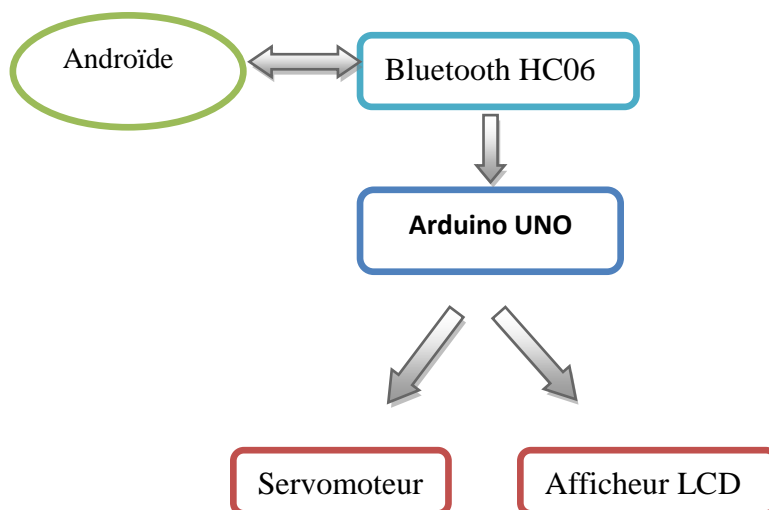
### III.3 Conception et réalisation :

Dans cette partie nous allons réaliser deux systèmes de communication sans fils, le premier entre un Android et Arduino et le second entre deux Arduino.

#### III.3.1 Système de communication Android Arduino :

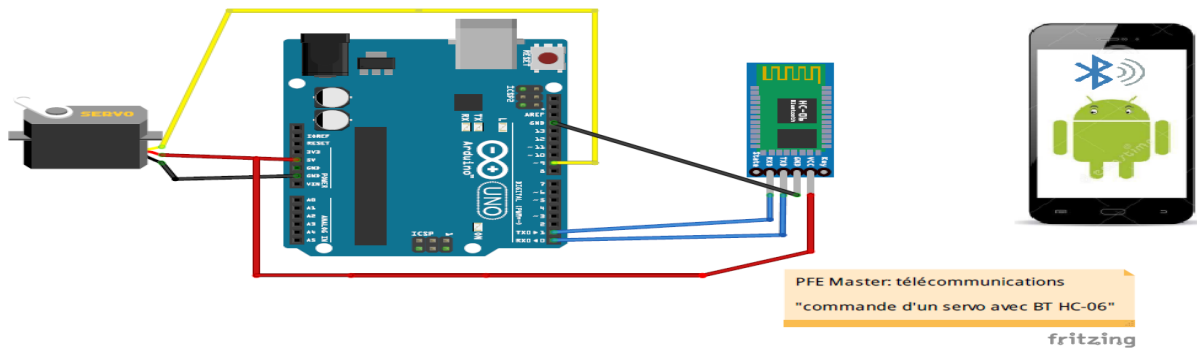
L'objectif est de réaliser un système de communication entre un Arduino et une application Android en utilisant le module Bluetooth esclave HC-06. Pour cela nous avons choisi de commander l'angle de rotation d'un servomoteur et afficher la valeur de l'angle sur l'afficheur LCD. Nous avons choisi l'application "terminal Bluetooth" téléchargé sur Google Play.

La figure ci-dessous montre le schéma bloc de ce système :



**Figure III.19** : Schéma bloc d'un système communication Arduino-Android.

Le schéma suivant illustre le circuit électronique ce système et la communication dessiné avec le logiciel FRITZING.



**Figure III.20:** Schéma électrique d'un système de communication Arduino-Android.

On voit que la broche de sortie le servomoteur est connecté à la pin 9 de l'Arduino ( fil jaune).

Les deux fils bleus représentent la liaison série entre le Bluetooth et Arduino selon le protocole RS232 (Rx de l'Arduino avec Tx du Bluetooth et Tx de l'Arduino avec le Rx du Bluetooth).

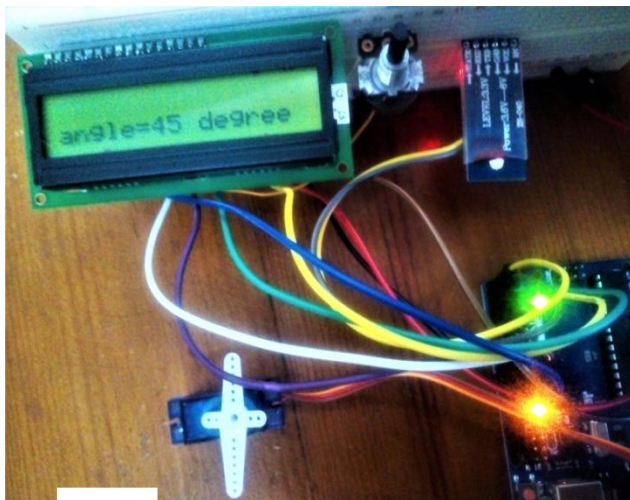
Sur le plan pratique, une manipulation adéquate du logiciel « Arduino » nous permet d'utiliser un compilateur ; il s'agit de « IDE ». Ce dernier possède une capacité de créer un code HEX transmis au microcontrôleur, d'une part une commande avec une application Androïde « Bluetooth terminal » fait tourner le servomoteur selon la valeur saisi, l'angle de rotation est donc affiché sur LCD.

La réalisation de ce système au labo de maquette a nécessité le matériel suivant :

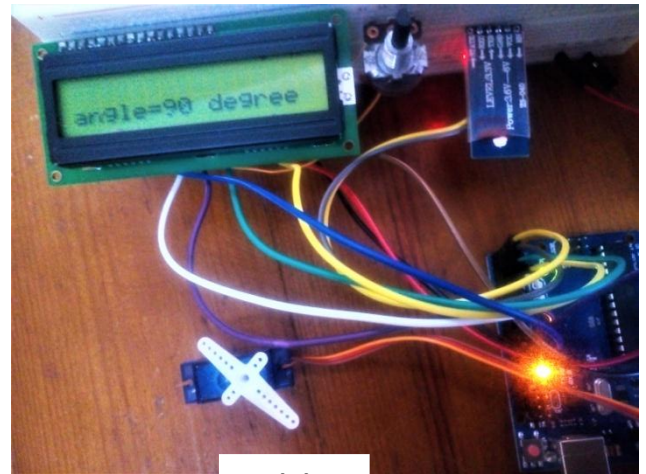
- Une plaque d'essai.
- Afficheur LCD.
- Un potentiomètre.
- Une carte Arduino.
- Module Bluetooth HC-06 (esclave).



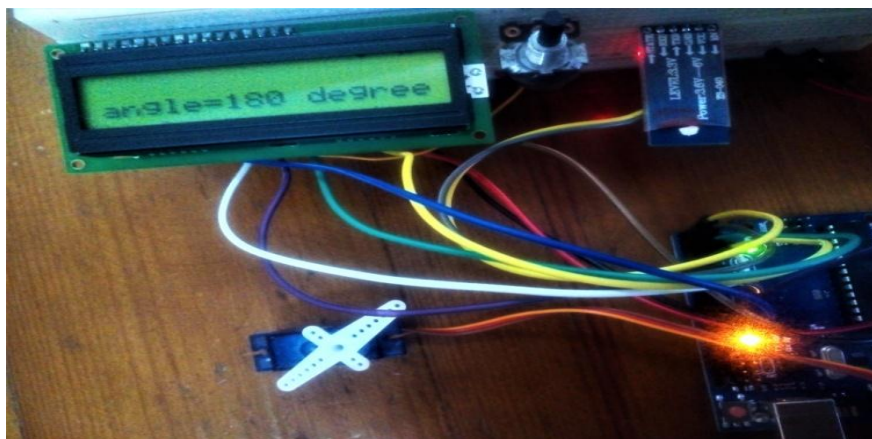
Nous avons obtenus les résultats pratiques suivants :



(a)



(b)



(c)

**Figure III.21** :Réalisation pratique du système de commande d'un servomoteur via Android,  
(a) Degré de rotation 45°, (b) Rotation à 90°, (c) Rotation à 180°.

### III.3.2 Système de communication entre deux Arduino :

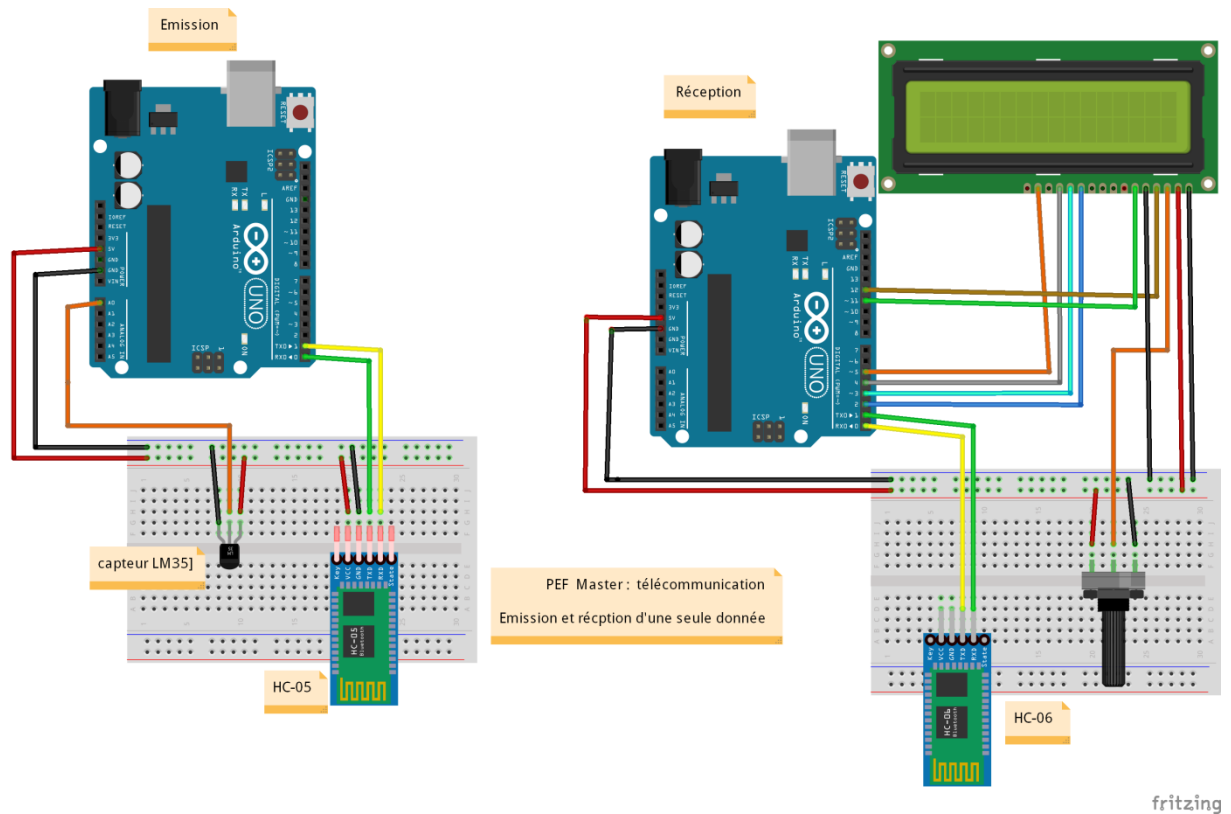
#### III.3.2.1 Transmission et réception d'une seule donnée :

L'objectif est de transmettre une donnée numérique en utilisant un capteur de température Lm35.

L'Arduino va permettre l'acquisition et l'envoi de la donnée via le Bluetooth maître HC-05. Ce dernier va ensuite connecter avec le HC-06 esclave qui va transmettre la donnée au deuxième Arduino qui va se charger de la réception et du décodage de l'information et permet d'afficher la grandeur mesurée sur l'afficheur LCD (figure III.22).



La figure suivante illustre le schéma électronique de ce système et la communication entre les composants. Dans la partie émission on voit bien la broche de sortie du capteur LM35 branchée à la broche analogique A0 de l'Arduino(en fils orange).



**Figure III.22** : Schéma électronique d'une communication sans fils entre deux Arduino.

*(Envoi d'une seule donnée)*

### III.3.2.1.1 Simulation avec ISIS PROTEUS :

Pour la réalisation pratique nous utilisons ISIS-PROTEUS, qui est un outil CAO électronique très perfectionné conçu par Labcenter Electronic qui permet de dessiner des schémas électroniques, de les simuler et de réaliser le circuit imprimé correspondant. Le CAO électronique « PROTEUS » se compose de nombreux outils regroupés en modules au sein d'une interface unique.

Isis PROTEUS comporte des bibliothèques de composants nécessaires pour dessiner un tel système. La figure ci-dessous montre la simulation sur Isis-Proteus avec les résultats affichés

sur LCD :

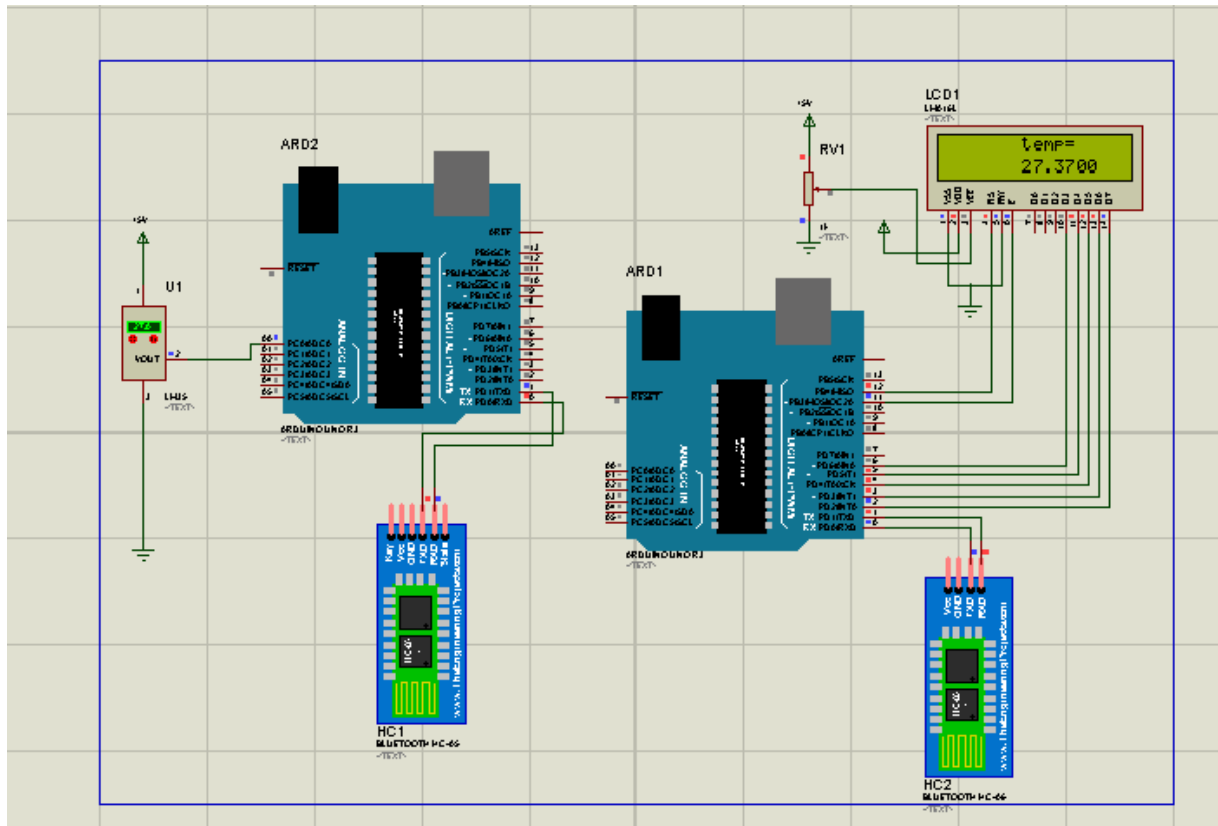


Figure III.23 : Simulation d'une communication sans fils entre deux Arduino.

(Envoi d'une seule donnée).

### III.3.2.2 Transmission et réception d'un paquet de données numériques :

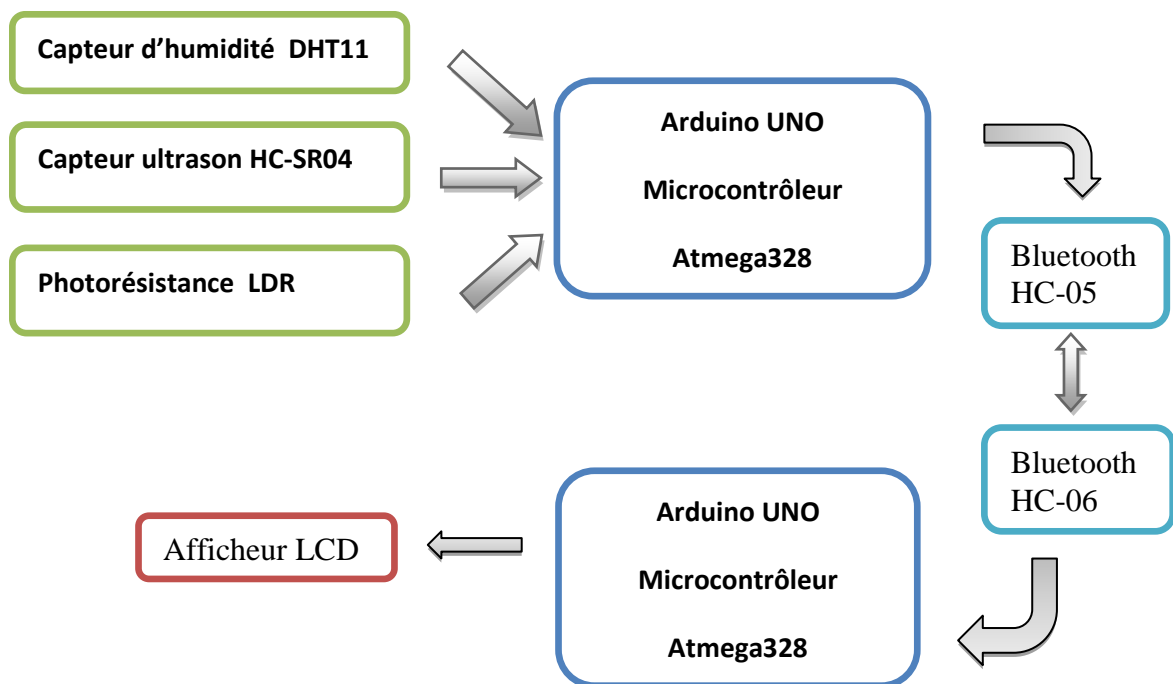
Dans ce système nous allons transmettre plusieurs données numériques. On a choisi donc trois capteurs :

- capteur d'humidité et de température DHT11.
- capteur de luminosité(LDR)
- capteur ultrasonique HC-SR04.

Ces trois capteurs vont donc afficher quatre valeurs sur LCD.

Le principe de transmission est le même que celui d'une seule donnée sauf que les données sont transmises en série l'une après l'autre entre l'Arduino et Bluetooth.

Les trois capteurs mesurent des grandeurs, l'Arduino va traiter ces grandeurs et les envoie au Bluetooth BT HC-06 avec la liaison série RS232, ce dernier va connecter au deuxième module BTHC-05 qui va se charger de les transmettre au second Arduino qui va à son tour traiter ces données reçues et les afficher sur l'écran LCD. La figure ci-dessous illustre le schéma bloc de ce système.



**Figure III.24** : Schéma bloc d'une communication sans fils entre deux Arduino.

*(Envoi d'un paquet de donnée).*

La figure ci-dessous illustre le schéma électronique de ce système.

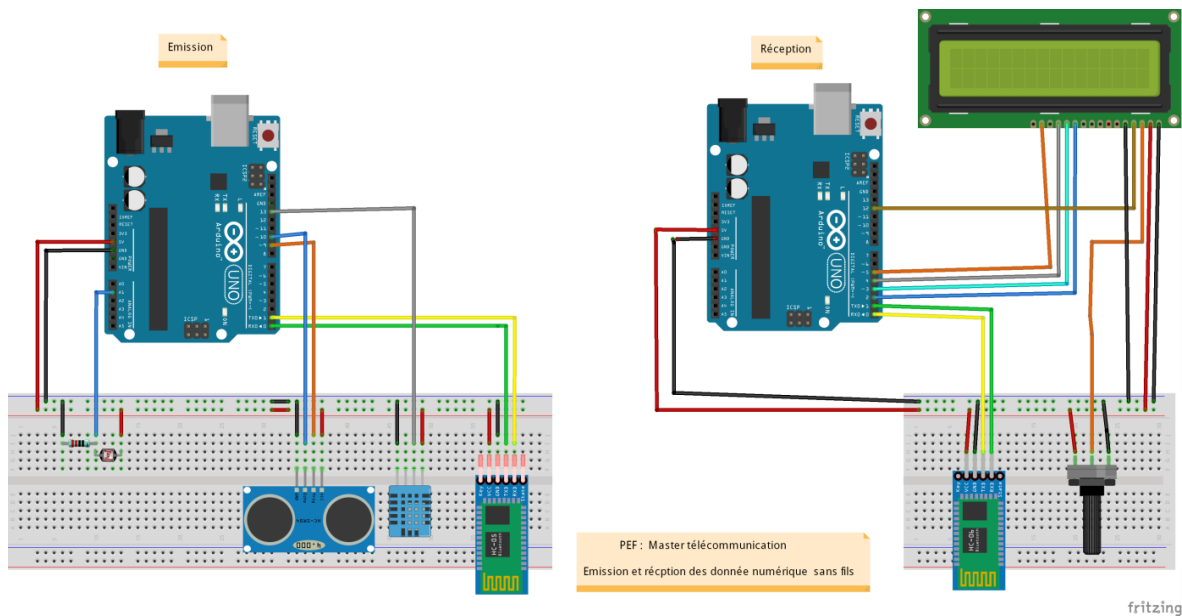


Figure III.25 : Schéma électronique pratique de la communication sans fil entre deux Arduino.

III.3.2.2.1 Simulation avec ISIS PROTEUS:

La figure suivante montre le schéma de simulation du système sur Isis -Proteus , on distingue bien les trois capteurs utilisés avec Arduino : Le LM35 , la photorésistance et le capteur ultrason .

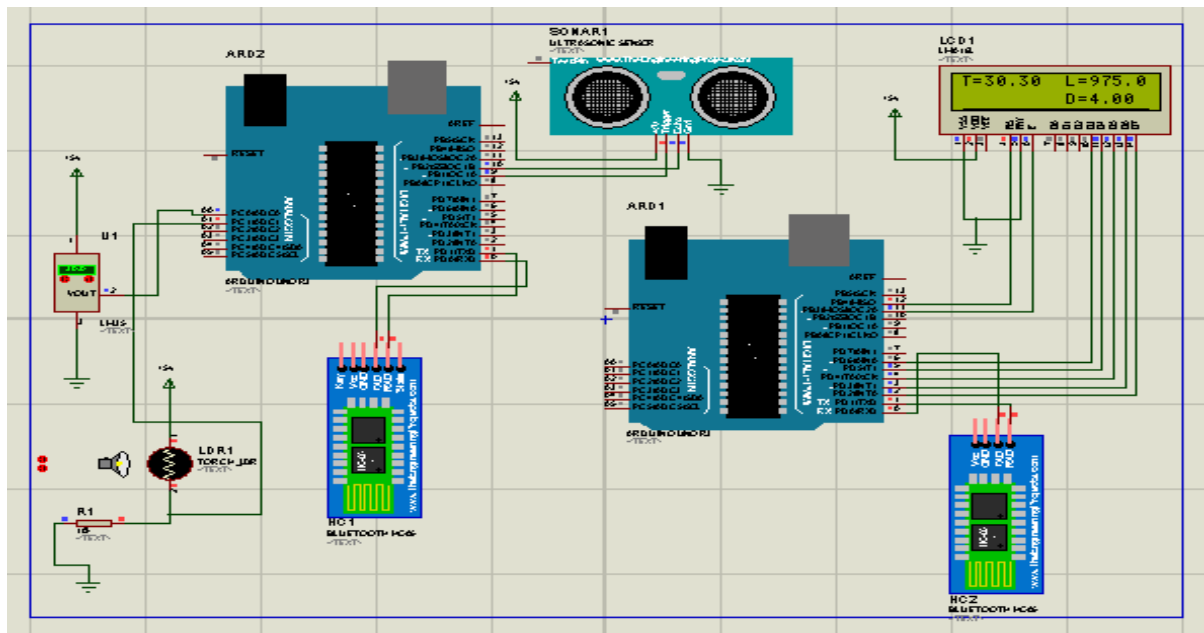


Figure III.26 : Simulation d'une communication sans fil entre deux Arduino.

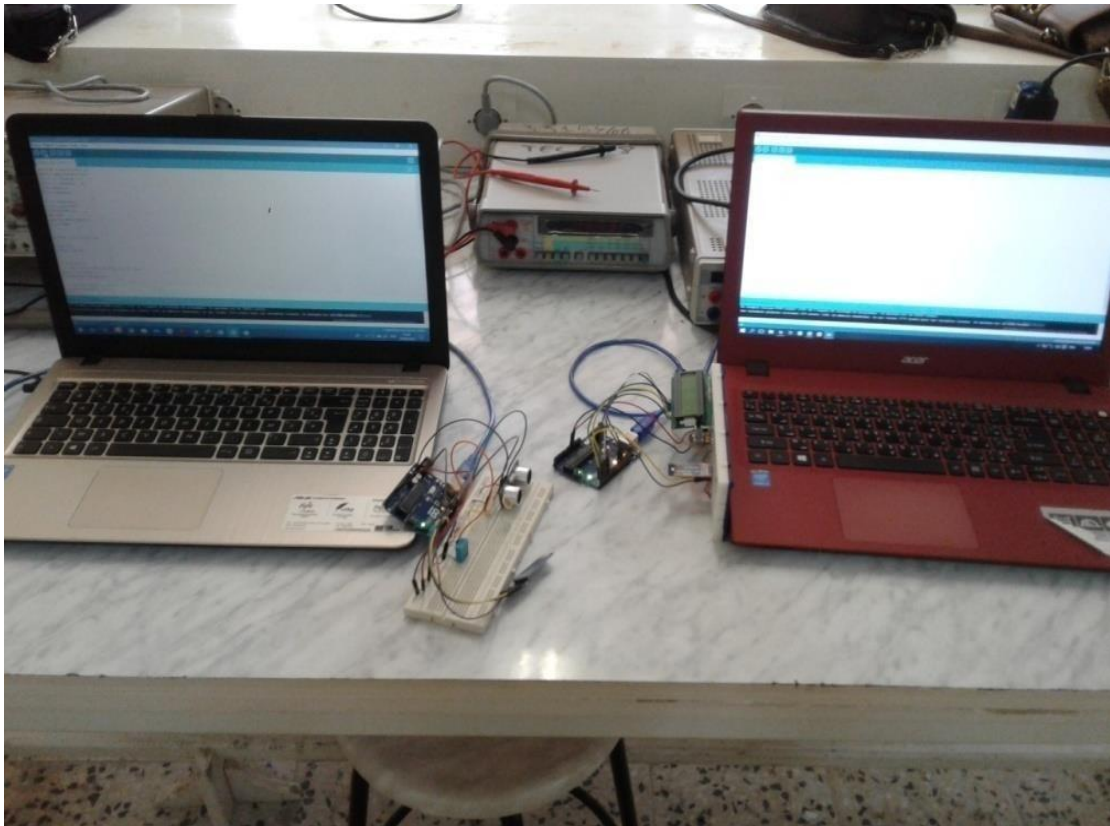
(Envoi d'un paquet de donnée).

**III.3.2.2 Réalisation pratique :**

La figure ci-dessous montre la réalisation finale au labo de maquette. Cette réalisation a nécessité les composants et modules suivants :

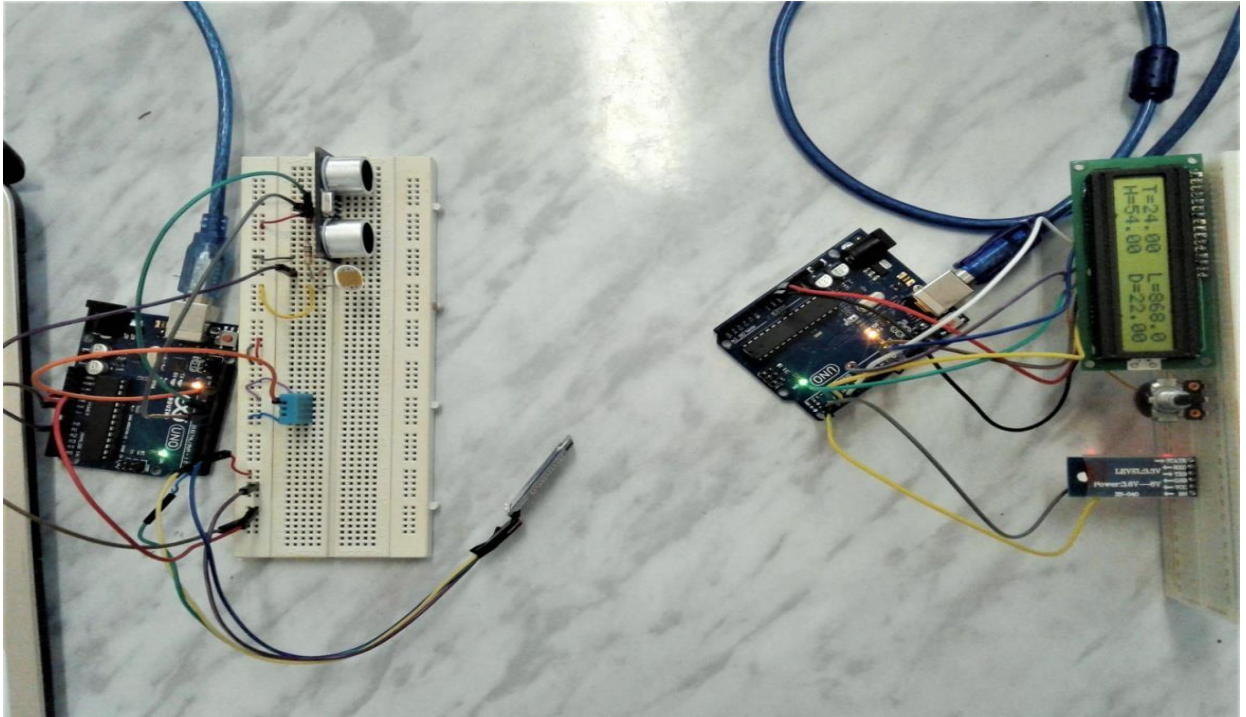
- Deux cartes Arduino UNO.
- Deux plaques d'essai.
- Trois capteurs : dht11, photorésistance, capteur ultrasonique.
- Des fils.
- Deux modules Bluetooth : BT HC-06, BT HC-05.
- Deux PC : Qui servent comme alimentation pour les cartes Arduino via le module USB.

La figure suivante montre le montage final réalisé au labo de maquette du système de communication entre deux Arduino avec trois capteurs en utilisant deux modules Bluetooth:

**(a)**



La figure suivante montre une image plus claire et précise de nos résultats affichées sur l'écran LCD.

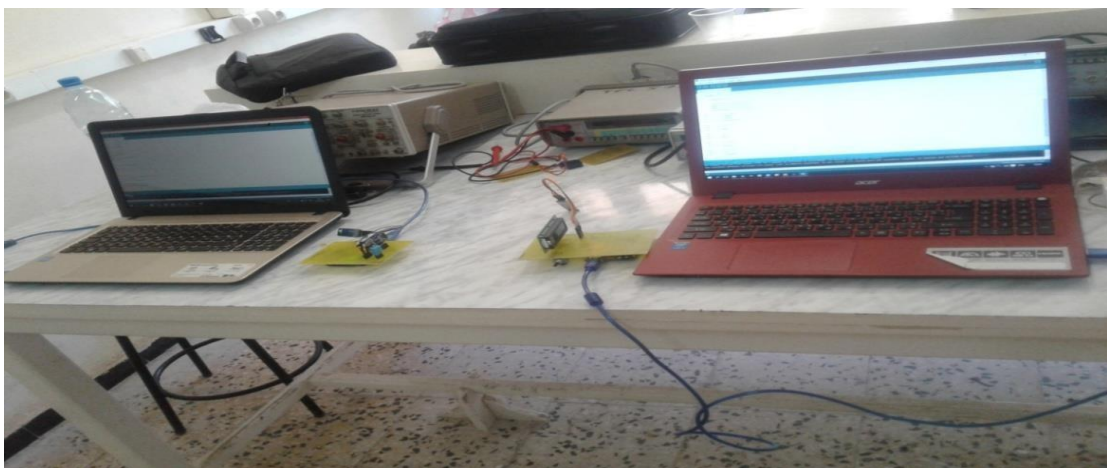


(b)

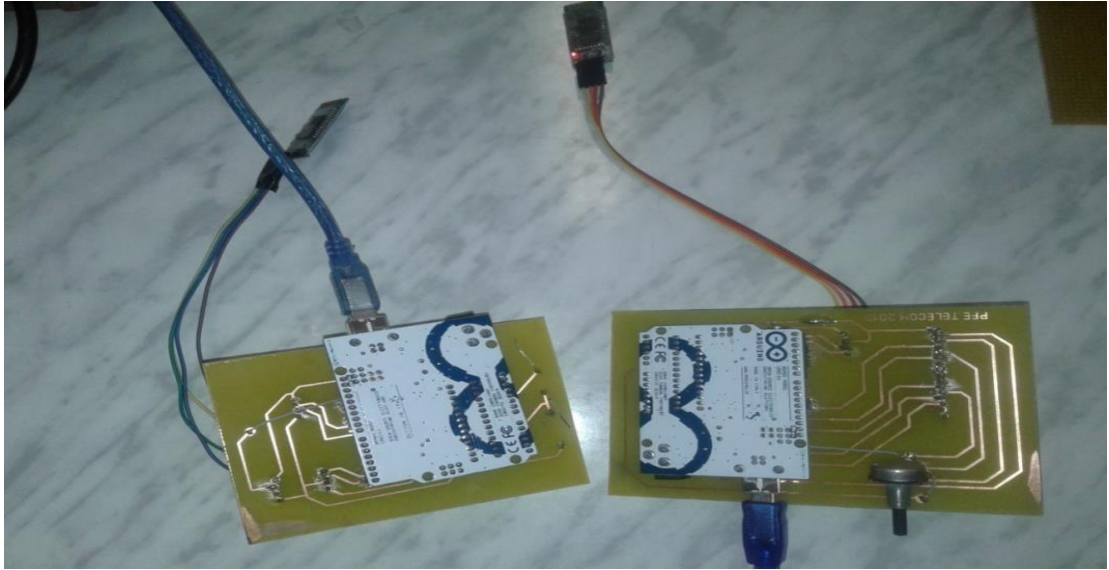
*Figure III.27 (a,b) : Réalisation pratique d'une communication sans fils entre deux Arduino.*

*(Envoi d'un paquet de donnée).*

Pour compléter notre travail, nous avons choisi d'implémenter notre système final sur des platines de circuits imprimés qu'on a réalisées au labo de maquette afin d'obtenir un système plus compact et plus fiable. Les images ci-dessous montrent le montage complet du système :

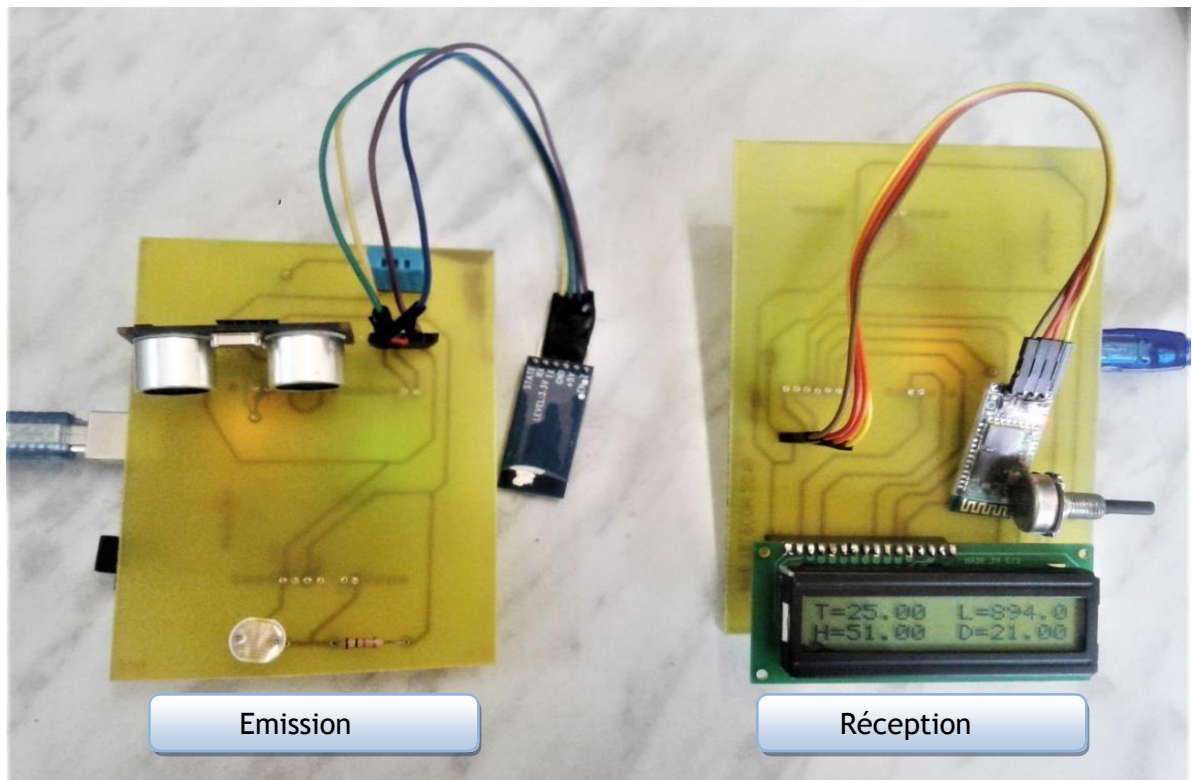


(c)



(d)

La figure ci-dessous montre le système sur des platines de circuit imprimés réalisés avec les résultats des données affichées sur l'afficheur LCD.



(e)

**Figure III.28 (c,d,e):**Réalisation pratique sur circuits imprimés d'une communication sans fils entre deux Arduino (Envoi d'un paquet de donnée).

# **Conclusion générale**



### Conclusion générale :

La communication sans fils facilite la transmission des données à distances, et offre une mobilité ce qui rend le transfert des données plus simple, et permet la création et la réalisation des systèmes plus performants. L'avènement de l'Arduino et le développement des systèmes numériques du simple circuit intégré au microprocesseur ainsi que le développement de multiples modules de communications sans fils compatibles avec Arduino (Bluetooth, RF, wifi ..... ) permet de développer des systèmes et des projets plus simples grâce à sa facilité de fonctionnement ,sa fiabilité et grâce à la documentation disponible et variation des exemples sur internet.

Dans le cadre de notre projet, nous avons réalisé deux systèmes de communications sans fil entre deux Arduino en utilisant deux modules Bluetooth.

Comme on a vu, on peut établir une communication entre les systèmes Android et un kit Arduino, pour la transmission et la réception des données numériques notamment dans le domaine de la domotique.

On peut aussi utiliser deux kits Arduino pour établir une communication unidirectionnelle ou bidirectionnelle pour l'envoi et la réception des données numériques issues de plusieurs capteurs pour commander des actionneurs ou pour afficher ces données sur un moniteur ou un LCD, comme dans le cas des réseaux de capteurs utilisés en industrie.

Les résultats de la réalisation sont très satisfaisants et concordent très bien avec ceux des simulations faites sous Proteus.

#### Perspectives :

Pour ce qui est de perspectives notre travail peut servir aux étudiants dans leurs projets de fin d'études comme élément de base pour réaliser et développer des systèmes plus performants et plus spécifiques tel que l'ouverture d'une porte à distance , réaliser des système de commande à distance, commande d'un robot, la domotique...

# *Annexes*

## Annexe 1 : Code Arduino fonctionne avec Smartphone via Bluetooth (Commande d'un servomoteur).

```

servo_mobile $
//importation de la bibliotheque .
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11,5,4,3,2);
#include <Servo.h>
Servo monservo;
// déclaration de variable .
int data;
int led=8;
int servoangle=0;
int servoPin=9;

void setup() {

Serial.begin(9600); //initialisation de la communication série par la commande Serial.begin,à 9600 bits par seconde.
monservo.attach(9); // Déclaration de pin
lcd.begin(16,2);
}
void loop() {

if(Serial.available()>0){
  data=Serial.read();
}

if(data=='0'){ // si la donnée reçu est égale à 0
  monservo.write(0); // le servo moteur a tourner à 0 degré
  lcd.setCursor(0,4);
  lcd.print("angle=0 degree"); // afficher l'angle 0 degré

  delay(5000); //Temps de traitement
  lcd.clear();
}
if(data=='1'){ // si la donnée reçu est égale à 1
  monservo.write(45); // le servo moteur a tourner à 45 degré
  lcd.setCursor(0,4);
  lcd.print("angle=45 degree");
  delay(5000);
  lcd.clear();
}
else if(data=='2'){ // si la donnée reçu est égale à 2
  monservo.write(90); // le servo moteur a tourner à 90 degré
  lcd.setCursor(0,4);
  lcd.print("angle=90 degree");
  delay(5000); //Temps de traitement
  lcd.clear();
}
else if(data=='3'){ // si la donnée reçu est égale à 3
  monservo.write(180); // le servo moteur a tourner à 180 degré
  digitalWrite(led,HIGH);
  lcd.setCursor(0,4);
  lcd.print("angle=180 degree");

  delay(5000); //Temps de traitement
  lcd.clear();
}
}
}

```

## Annexe 2 : Code Arduino fonctionne avec Bluetooth pour l'émission (Envoi d'un paquet de données).

```

system_emission $
#include<dht.h>//importation de la bibliotheque du capteur dht
dht DHT;
#define DHT11_PIN 12; //Déclaration de la PIN
#define max_distance 200; // definir la distance maximun pour l'ultrason

// decalation des variables
const int trigPin = 9; // pin "trig" du HC-SR04 connectée à la pin 9 de l'Arduino
const int echoPin = 10; // pin "trig" du HC-SR04 connectée à la pin 9 de l'Arduino
long duration;
float distance;
float valeur=A1; // pin LDR connectée à la pin A1 de l'Arduino
float hum; // déclaration de la variable float

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600); //initialisation de l'ecran LCD: 16colonnes et 2 lignes
  pinMode(trigPin, OUTPUT); //initialisation de la pin trig de HCSR05 en sortie
  pinMode(echoPin, INPUT); //initialisation de la pin echo de HCSR05 en entree

void loop() {

  valeur=analogRead(A1); //lire la valeur analogique du LDR
  int val = DHT.read11(DHT11_PIN); //lire la donnee depuis le capteur DHT11
  hum=DHT.humidity; // variable pour stocker l'humidité
  float temp=DHT.temperature; // variable pour stocker la temperature

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH); //lire l'echoPin , retourner le son onde Reads the echoPin, returns the sound wave travel time in microseconds
  distance= duration*0.034/2; // calculer la distance

  Serial.print('#'); // envoie d'un signe au buffer

  //imprimer les donnees dur le moniteur serie
  Serial.println(valeur);
  Serial.println(temp);
  Serial.println(hum);
  Serial.println(distance);

  delay(500); // délai d'attente
}

```

### Annexe 3 : Code Arduino fonctionne avec Bluetooth pour la réception (Envoi d'un paquet de données).

```

system_reception
#include <LiquidCrystal.h> // importation de la bibliotheque de LCD
LiquidCrystal Lcd(12,11,5,4,3,2); // importation de la bibliotheque de LCD
#define max_distance 200 // définir la distance maximum de HC-SR05
// declaration des variables
int data;
float valeur=0;
float temp;
float hum;
float distance;

void setup() {

  Serial.begin(9600); // initialisation de la communication série par la commande Serial.begin , à 9600 bit par seconde.

  Lcd.begin(16,2); //initialisation de l'ecran LCD: 16colonnes et 2 lignes
  Lcd.setCursor(0,0); //définir l'emplacement du prochain texte écrit sur l'ecran LCD
  Lcd.print("PFE Master 2 :"); // imprimer "PFE Master sur l'ecran LCD
  delay(4000); // délai d'attente 4000 secondes
  Lcd.clear(); // effacer l'ecran LCD

}
void loop() {

  if (Serial.available()>0){ // si le buffer disponible
    data=Serial.read(); // lire la valeur de la donnée sur le buffer
  }

  if(data=='#'){ // si la donnée sur le buffer egale à #

    valeur=Serial.parseFloat(); //renvoie le premier nombre à virgule flottante valide à partir du buffer série
    temp=Serial.parseFloat(); //renvoie le premier nombre à virgule flottante valide à partir du buffer série
    hum=Serial.parseFloat(); //renvoie le premier nombre à virgule flottante valide à partir du buffer série
    distance=Serial.parseInt();//renvoie le premier nombre à virgule flottante valide à partir du buffer série

    // imprimer sur le moniteur série
    Serial.print(valeur);
    Serial.print(temp);
    Serial.print(hum);
    Serial.print(distance);

    Lcd.setCursor(0,0);//définir l'emplacement du prochain texte écrit sur lcd
    Lcd.print("T="); // imprimer "T="( T pour temperature ) sur l'ecran LCD
    Lcd.print(temp);

    Lcd.setCursor(0,1);
    Lcd.print("H=");//imprimer "H="( H pour humidité ) sur l'ecran LCD
    Lcd.print(hum);

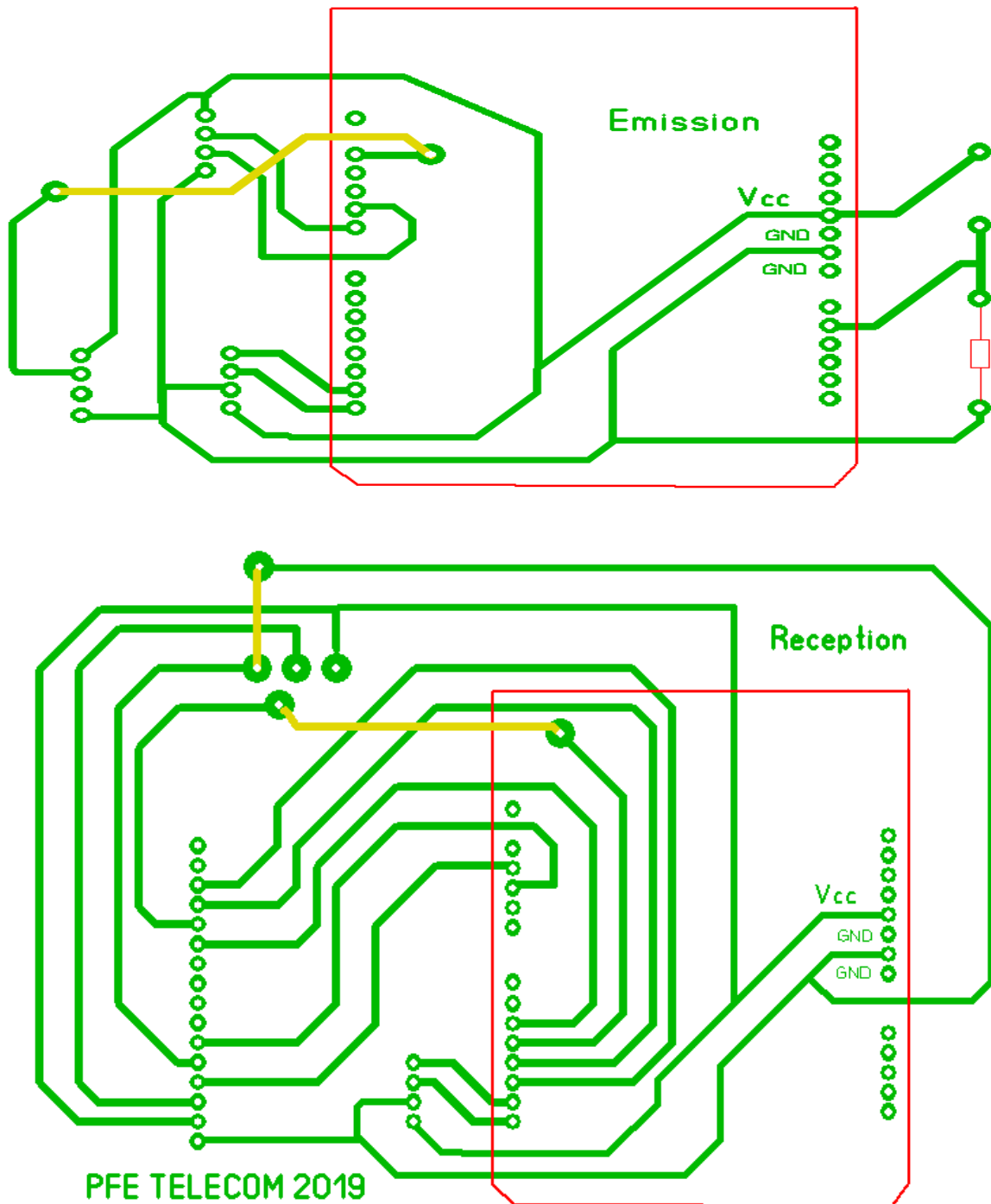
    Lcd.setCursor(9,0);
    Lcd.print("L=");//imprimer "L="( T pour luminosité ) sur l'ecran LCD
    Lcd.print(valeur);

    Lcd.setCursor(9,1);
    Lcd.print("D=");//imprimer "D="( D pour distance ) sur l'ecran LCD
    Lcd.print(distance);
  }
  delay(500);// délai d'attente
}
}

```

*Annexe 4* : Circuit imprimé dessiné avec logiciel Layout4.0

(Emission &amp; réception).



# *Références*

## Références Biblio-webographiques :

- [1] **S.Kundo**, Analog and Digital communication, Pearson, India,2010.
- [2] **E.Krouk, S.Semenov**, modulation and coding techniques in wireless communication, Wiley, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2011.
- [3] **J.Higgins**, introduction to SNG end ENG microwave, Focal Press,70 Blanchard Road, Suite 402, Burlington, MA 01803,2013.
- [4] **G. J. Foschini, M. J. Gans**, On limits of wireless communications in a fading environment when using multiple antennas. Wireless Personal Communications, 6(3):311–335, march 1998.
- [5] **A. Wittneben. Basestation** modulation diversity for digital simulcast. In Proceedings of the IEEE Vehicular Technology Conference (VTC'91), pages 848–853, May 1991.
- [6] <https://www.elprocus.com> [consulté Avril 2019].
- [7] <https://www.tutorialspoint.com> [consulté Avril 2019].
- [8] **S. Barrett**, “Arduino Microcontroller Processing for Everyone!, 3rd Edition,” Morgan & Claypool , 2013.
- [9] **M. Vercammen, J. Maes**, “Digital Audio Technology, 4th Edition, ” Focal Press,2013
- [10] **L. Frenzel**, “Habdbbook of serial communication Interfaces,” Newnes ,2015
- [11] **B. Grimes**, “Networked AV Systems,” McGraw-Hill Osborne Media, 2014.
- [12] <https://circuitdigest.com> [consulté mai 2019]
- [13] **C. Bergmann, F.Alin, C.Darrault, E.Garnier**, “Electronique et communication ,” Dunod, Paris,2010.
- [14] **J. Lovine**, Projets PIC pour les non-programmeurs, Newnes, 2011.
- [15] <https://www.electronicshub.org/rs232-protocol-basics/> [consulté juin 2019].
- [16] **D.Paret**, “Multiplexed Networks for Embedded Systems: CAN, LIN, FlexRay, Safe-by-Wire...,” John Wiley & Sons, 2007



- [17] <https://eduscol.education.fr> [consulté mai 2019]
- [18] <https://www.electronicshub.org> [consulté juin 2019]
- [19] <https://www.analog.com> [consulté juin 2019]
- [20] **A.Trevennor** ‘ *Practical AVR Microcontrollers: Games, Gadgets, and Home Automation with the Microcontroller Used in Arduino* ’’ Apress, 2012
- [21] **F.C.Astups, J.Lechalupé**, ‘ *Cours d’initialisation à Arduino,* ’’ Université Paul Sabatier, Mai 2014
- [22] **M. DJAFRI, D.CHELOUCHE**,’’ *ETUDE ET REALISATION D’UNE SERRURE ELECTRONIQUE CODEE (A BASE DU PIC 16F877/ARDUINO UNO)*,’’Mémoire en Science et Technologie, Filière Génie Electrique, Spécialité Informatique Industrielle, Université Larbi ben M’Hidi, Oum el Bouaghi, 2017
- [23] **Société SANDVIK-COROMANT**, « Fraisage, principes », Techniques de l’ingénieur, document BM7082. [En ligne].
- [24] [https://www.academia.edu/5163800/Chapitre3.\\_microcontrolleur](https://www.academia.edu/5163800/Chapitre3._microcontrolleur)[consulté mai2019].
- [25] **H.Takhi** ‘ *Conception et réalisation d’un robot mobile à base d’arduino,* ’’Université Amar Telidji - Instrumentation 2014 [en ligne ] . [consulté juin2019].
- [26] <https://www.theengineeringprojects.com/>[consulté juin 2019].
- [27] **B. Cottenceau**, ‘ *Cours Carte ARDUINO UNO Microcontrôleur ATmega328,* ’’ Microcontrôleurs EI3 Option AGI, Université Angers , France,2015 [en ligne].
- [28] <https://www.locoduino.org> [consulté juin 2019].
- [29] **J. M. Hughes**, ‘ *Arduino A Technical Reference,* ’’ O’Reilly Media, Inc , 2016
- [30] **E.Williams**,’’ *Marque: Programmation AVR,* ’’ Maker Media, Inc , 2014
- [31] **M.Djafri, D.Chelouche**, ‘ *Etude et réaliser la carte Arduino* ’’ Mémoire de Master en Science et Technologie, Filière Electronique, Spécialité Automatique, Université A.MIRA, Béjaia,2017.
- [32] **J.Culkin, E.Hagan**, ‘ *Learn Electronics with Arduino,* ’’ Maker Media, Inc, 2017.
- [33] **A.GOURARI** , ‘ *Etude, simulation et réalisation d’une station de détection permettant la mesure de la vitesse d’un mobile par effet Doppler en mode continu,* ’’Mémoire de Master en Electronique , spécialité Instrumentation électronique, Université Aboubakr Belkaïd,Tlemcen,2017

- [34] **L. GUENAOUA**, ''*Commande en position du a MCC par Arduino*, '' Mémoire de Master en science et technologie, Filière Génie électrique, Spécialité Instrumentation et contrôle industriel, 2017.
- [35] [https://www.electronicshub.org/wireless-communication-introduction-types-applications/#Why\\_Wireless\\_Communication](https://www.electronicshub.org/wireless-communication-introduction-types-applications/#Why_Wireless_Communication) [consulté Mai 2019]
- [36] <https://www.edgefx.in/different-types-wireless-communication-technologies/>  
[Consulté Mai 2019]
- [37] [https://www.anixter.com/fr\\_ca/ressources/documentation/fiches-techniques/2fpjk/comparaison-des-protocoles-de-communication-sans-fil.html](https://www.anixter.com/fr_ca/ressources/documentation/fiches-techniques/2fpjk/comparaison-des-protocoles-de-communication-sans-fil.html)  
[consulté Mai 2019]
- [38] <https://www.commentcamarche.net/contents/108-bluetooth-comment-ca-marche>[consulté Mai 2019]
- [39] <https://www.tutoriel-arduino.com/tutoriels/utiliser-hc-05-arduino>[consulté juin 2019]
- [40] [https://wiki.eprolabs.com/index.php?title=Bluetooth\\_Module\\_HC-05](https://wiki.eprolabs.com/index.php?title=Bluetooth_Module_HC-05) [consulté juin 2019]
- [41] <https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05->  
[consulté juin 2019]
- [42] <https://pecquery.wixsite.com/arduino-passion/copie-de-le-detecteur-a-ultrasons-h-1>  
[consulté juin 2019]
- [43] <https://www.carnetdumaker.net/articles/mesurer-une-temperature-avec-un-capteur-lm35-et-une-carte-arduino-genuino/>[consulté juin 2019]
- [44] **H.HAMOUCI**, '' Conception et réalisation d'une centrale embarquée de la domotique « Smart Home »'' Mémoire de Master en génie électrique , Université Mohammed V École Normale Supérieure d'Enseignement Technique - Rabat , Maroc ,2015
- [45] <http://idehack.com/blog/les-photoresistances-et-larduino/>[consulté juillet 2019].