

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE
UNIVERSITE AKLI MOAND OULHADJ-BOUIRA



Faculté des Sciences et Sciences Appliquées
Département Génie Electrique

Mémoire de fin d'étude
Présenté par
Boulahbal Ahlam
Elamouri Soumia Firyal

En vue de l'obtention du diplôme de **Master en**

Filière : ELECTRONIQUE

Option : Electronique des Systèmes Embarqués

Thème

**Conception et construction d'une interface graphique intelligente pour
le test et l'étalonnage des composants électroniques avec Arduino**

Soutenu le 28/12/2020

Devant le jury composé de

Mr.Bouhedda	MCA	UAMOB	Président
Mr.Fekik	MCA	UAMOB	Examinateur
Mr.Yahiou Abdelghani	MCA	UAMOB	Promoteur
Mr.Zarifi Adel		UAMOB	Co-Promoteur

Année Universitaire 2019/2020

Remerciements

*Nous tenons tout d'abord à remercier **DIEU** le tout puissant, qui nous a donné la force et la patience d'accomplir ce Modest travail.*

*Nous tenons à exprimer nos sincères remerciements à notre encadreur de mémoire, **Mr. YA-HIOU Abd el ghani**.*

*Nos Remerciements vont aussi à notre Co promoteur **Mr. ZARIFI Adel** de nous avoir proposé ce sujet de mémoire. Nous avons découvert grâce à lui le monde de la recherche dans les meilleures conditions.*

*Nous remercions chaleureusement **Mr. DAIFI Abdelhek** et **Mr. DAIFI Salah Eddine** qui nous ont conseillé et encouragé tout le long de notre travail. Nous leurs exprimons notre profonde gratitude pour leurs attention, et surtout pour la grande patience qu'ils ont manifesté. Sans leurs soutien et leurs disponibilité, ce travail n'aurait pas pu être couronné de succès.*

Nous remercions également l'ensemble des membres du jury pour l'honneur qu'ils nous ont fait en acceptant d'évaluer notre travail.

*Nous remercions tout les professeurs de l'université **AKLI MOHAND OULHADJ** notamment ceux du département de **GENIE ELECTRIQUE**.*

Sans oublier de remercier nos amis, nos proches et toutes les personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Et pour finir, nous présentons nos remerciements les plus sincères à nos chères familles, particulièrement à nos parents pour leur soutien inconditionnel et pour leur encouragement durant toutes nos années d'étude.

Dédicaces

*Je tiens tout d'abord à remercier **ALLAH** le tout puissant et miséricordieux, qui m'a donné La force et la patience d'accomplir ce Modeste travail.*

Tous les mots ne sauraient exprimer la gratitude, l'amour, le Respect, la reconnaissance, c'est tout simplement que : Je dédie cette thèse de Master à :

*À **MES CHERS PARENTS**, Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.*

*À mes chères Sœurs, **CHAHRAZED, AFAF** et **AYA**, Vous aviez toujours cru en moi, et c'est dans votre présence que j'ai puisé la volonté de continuer. je vous souhaite une vie pleine de bonheur et de succès.*

*À mes chers frères, **SYFEDDINE, ALAEDDINE**, pour leur appui et leur encouragement. Vous aviez rempli mes moments de joie et de bonheur. Je vous souhaite tout ce qu'il y a de meilleur.*

À mes grandes mères et père : je vous aime trop vous êtes notre baraka, que dieu vous garde.

*À tous les membres de ma famille petits et grands, spécialement mon oncle **OMAR**.*

*À mon binôme et ma chère amie **FERIEL**, je te remercie pour ton amitié chère à mon cœur, et je te souhaite tout le bonheur du monde.*

À toutes mes chères amies et à tous mes chers amis.

À tous ceux qui sont chères, avec qui j'ai partagé des moments des plus agréables et qui ont toujours été présent pour moi et à tous ceux qui m'aiment et que j'aime.

À tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

AHLAM.

Dédicaces

*Avec l'aide de bon **DIEU** j'ai pu réaliser ce modeste travail que je dédie :*

*À ma nièce **INES** qui j'aurais aimé qu'elle soit avec nous aujourd'hui.*

*À la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur, **MAMAN** que j'adore.*

*À l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi **PAPA**.*

*À mon cher frère **SALIM**, Le guide de mes désirs, le donneur avec plaisir, ma fierté et mon pouvoir, merci.*

*À mes chères sœurs **SALIMA** et **MERIEM** : En témoignage de mon affection fraternelle, de ma profonde tendresse et reconnaissance, je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protégé et vous garde.*

*À mon cher neveu **ANES** et ma chère nièce **LYDIA**, aucune dédicace ne saurait exprimer tout l'amour que j'ai pour vous, Votre joie et votre gaieté me comblent de bonheur. Puisse Dieu vous garder, éclairer votre route et vous aider à réaliser à votre tour vos vœux les plus chers.*

*À mon beau-frère **MOUNIR** pour son appui et son encouragement.*

*À mon binôme et ma chère amie **AHLAM**, en souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble.*

*À **Abdelhek**, tu étais toujours présent pendant ce travail, tu m'a conseillé et encouragé. Je suis chanceuse de t'avoir à mes cotés.*

À tous les membres de ma famille, petits et grands.

À toutes mes chères amies.

À tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

FERIEL.

Résumé

L'étalonnage dans le domaine technique est le processus de configuration d'un composant électronique pour fournir un résultat pour un échantillon dans une plage acceptable.

Dans ce projet, on propose de faire une conception d'une interface graphique intelligente pour le but d'étalonnage des composants électroniques à base d'Arduino et le logiciel Visual Studio (C#).

Les résultats intègrent la construction de l'application intelligente, les images de branchement des composants électroniques et les capteurs qui affichent le succès de l'exécution.

Afin de terminer ce mémoire, on propose aussi une ouverture sur des études complémentaires à réaliser pour prolonger le travail.

Mots clés : Arduino, Visual Studio(C#), étalonnage. . .

Abstract

Technical calibration is the process of configuring an electronic component to provide a result for a sample within an acceptable range.

In this project, we propose to design an intelligent graphical interface for the purpose of calibration of electronic components based on Arduino and the software Visual Studio (C#).

The results incorporate the construction of the intelligent application, the connection images of electronic components and the sensors that display execution success.

In order to complete this thesis, we also propose an opening on additional studies to be carried out to extend the work.

Key words : Arduino, Visual Studio (C#), calibration. . .

Table des matières

Table des matières	i
Table des figures	iii
Liste des tableaux	v
Liste des abréviations	vi
Introduction générale	1
1 Généralité sur l'électronique et les systèmes embarqués	3
1.1 Introduction	3
1.2 Les composants électroniques	3
1.3 Types des circuits électroniques	4
1.3.1 Circuits analogiques	4
1.3.2 Circuits numériques	4
1.3.3 Circuits mixtes	5
1.4 Les systèmes embarqués	6
1.5 Historique	6
1.6 Caractéristiques d'un système embarqué	6
1.6.1 La fiabilité	6
1.6.2 L'efficacité	6
1.6.3 Systèmes hybrides	7
1.6.4 Les périphériques d'entrée/sortie (IO)	7
1.7 Architecture d'un système embarqué	7
1.8 Conclusion	9
2 Les techniques de la communication	10
2.1 Introduction	10
2.2 Historique	10
2.3 Les classifications de la communication	11
2.3.1 La transmission parallèle	11
2.3.2 La transmission série	12
2.4 Différence entre la communication série et parallèle	12
2.5 Les modes de transmission	13
2.6 Les transmissions séries asynchrone et synchrone	13
2.6.1 La liaison asynchrone	13

2.6.2	La liaison synchrone	14
2.7	Les protocoles série asynchrones	14
2.7.1	Protocole RS-232	14
2.7.2	Protocole USB	15
2.8	Protocoles série synchrones	16
2.8.1	Protocole CAN	16
2.8.2	Protocole WIFI	16
2.9	Conclusion	17
3	Les processus d'étalonnage et tests	18
3.1	Introduction	18
3.2	Définition	18
3.3	Le Système international d'unités (SI)	19
3.4	Traçabilité et Normes	20
3.4.1	Traçabilité et hiérarchie d'étalonnage	20
3.4.2	Les diverses catégories d'étalons (normes)	21
3.5	Certificat d'étalonnage	21
3.6	Les rapports des tests	22
3.6.1	TAR et TUR	22
3.7	Incertitude d'étalonnage (incertitude de mesure)	22
3.8	Intervalle d'étalonnage	22
3.9	Documentation des résultats d'étalonnage	23
3.10	L'importance d'étalonnage	23
3.11	Conclusions	24
4	Matériels et logiciels	25
4.1	Introduction	25
4.2	Soft	25
4.2.1	GUI avec des objets physiques	25
4.2.2	Visual Basic	26
4.3	Hard	32
4.3.1	Présentation de la carte Arduino	32
4.3.2	Les gammes de la carte Arduino	33
4.3.3	Caractéristiques de la carte Arduino UNO	34
4.3.4	Constitution de la carte Arduino UNO	35
4.3.5	Le logiciel Arduino	38
4.4	Conclusion	39
5	Réalisation et implémentation du projet	40
5.1	Introduction	40
5.2	Les étapes de la réalisation	40
5.3	La réalisation pratique	40
5.3.1	Structure du système	40
5.3.2	Résultats de la réalisation expérimentale	42
5.4	Conclusion	50

Table des figures

1.1	Un schéma de circuit analogique, dans ce cas un simple amplificateur	4
1.2	Un FPGA Stratix IV d'Altera	5
1.3	Architecture d'un système embarqué	8
2.1	Transmission parallèle	11
2.2	Transmission série	12
2.3	La liaison asynchrone	14
2.4	La liaison synchrone	14
2.5	Première utilisation typique de RS-232	15
2.6	Applications et fonctions en USB	16
2.7	Communication WIFI client-serveur ESP8266 entre deux cartes	17
3.1	L'étalonnage des instruments électroniques	19
3.2	Le Système international d'unités (SI)	19
3.3	Hierarchie d'étalonnage décrite par l'exemple de l'Allemagne.	20
3.4	Exemple de Certificat d'étalonnage	21
4.1	Page de téléchargement Visual Studio 2019	27
4.2	Page de démarrage de Visual Studio 2019	27
4.3	Créer un nouveau modèle de projet	28
4.4	Configuration du projet	29
4.5	L'IDE de Visual Basic 2019 Express	29
4.6	Boîte à outils Visual Basic 2019	30
4.7	L'interface de conception	30
4.8	Fenêtre de code Visual Basic 2019	31
4.9	L'interface d'exécution	31
4.10	La boîte de message	32
4.11	Fonctionnement de l'Arduino	33
4.12	La carte Arduino UNO	34
4.13	Les composants de la carte Arduino UNO	35
4.14	Bronchage de la carte Arduino UNO	36
4.15	Interface de logiciel Arduino.	38
5.1	Schéma bloc du système.	41
5.2	L'application graphique intelligente	42
5.3	Résultats de réalisation de la LED	43
5.4	Résultats de réalisation de Bouton poussoir	44

5.5	Résultats de réalisation de Capteur de température et d'humidité	45
5.6	Résultats de réalisation des Potentiomètres et LDRs	46
5.7	Résultats de réalisation de Servo moteur	47
5.8	Résultats de réalisation de Feu rouge	47
5.9	Résultats de réalisation de LCD	48
5.10	Résultats de réalisation de Capteur de distance	49
5.11	Résultats de réalisation de la luminosité d'une led	50

Liste des tableaux

- 2.1 La comparaison entre la communication série et parallèle 12
- 2.2 Les modes de transmission 13

- 4.1 Les caractéristiques d'Arduino UNO 34

Introduction générale

L'objectif principal de l'évolution de l'électronique est de créer des systèmes de plus en plus puissants, miniaturisés et autonomes, tout en minimisant le coût d'étude et de production. Actuellement, les systèmes ne sont plus exclusivement électroniques mais impliquent la mise en œuvre d'autres technologies dans un même boîtier que l'électronique, comme la mécanique, l'optique, la chimie ...etc. La communauté des concepteurs adhère maintenant à l'approche **MBD** (Model Based Design) mais les difficultés d'optimisation du processus de conception sont augmentées par les considérations de la minimisation du temps de mise sur le marché, il faut donc développer de nouvelles approches efficaces de modélisation et de vérification haut niveau [1].

Le but de ce travail est la Conception et construction d'une interface graphique intelligente pour le test et l'étalonnage des composants électronique avec Arduino.

Pour cela, nous essayons de réaliser une interface graphique à base de visuel studio (C#) et Arduino UNO pour créer une application intelligente (sur pc ou bien sur Smartphone) dans le but de tester et étalonner des composants électroniques, obtenir et afficher les résultats a distances des capteurs électroniques (capteur de Température et Humidité, Capteur de lumière , Servomoteurs, ultra-son, potentiomètre, ... etc.) afin de faciliter notre étude.

Afin d'acquérir des connaissances pratiques, nous avons visé une société.

Cette société spécialisée dans le domaine technique. Elle se définit comme fabriquant des composants électroniques et commerçant local et en ligne. Afin de pouvoir régler le problème de retour des composants électroniques pour satisfaire les clients. Le directeur de la société nous a appelé pour présenter personnellement le problème suggéré ci-dessus afin de trouver les meilleures solutions.

Tant que nous sommes des futures ingénieurs en électronique, on a déplacé directement vers l'entreprise, nous avons fait une visite générale au terrain et on a constaté plusieurs problèmes. Parmi ces derniers ,nous trouvons que la société travaille avec des méthodes classiques pour faire des tests qui prennent beaucoup de temps et un facteur humain très élevé. Elle compte aussi sur des machines qui ne sont pas développées technologiquement, très chères ,avec un manque de fiabilité(ne sont pas numériques) ainsi qu'elle a un grand espace occupé.

La société n'a pas une méthode de traçabilité et elle propose une garanties aux clients dans chaque vente ,cela veut dire que les composants électroniques défectueux ne sont identifiés.

Pour finir, on a posé une question au directeur concernant les composants qui ont une grande

quantité de retour. Il nous a donné une liste détaillé pour faciliter le travail, Parmi ces composant on a : LCD, Capteur de température et de humidité, potentiomètre, servomoteur, LED, LDR,capteur de distance...etc.

Nous avons proposé une application qui a la traçabilité avec un numéro de série pour confirmer que les composants sont fabriqués par la société.

Organisation du mémoire

Le reste de notre mémoire est organisé comme suit :

- Dans le premier chapitre, nous allons voir une présentation générale sur l'électronique et les systèmes embarqués ainsi ses caractéristiques et l'architecture utilisée.
- Le deuxième chapitre est consacré à la présentation de la communication série, ses classifications, ses modes de transmission et la présentation de quelques protocoles de communication.
- Dans le troisième chapitre, nous présentons une brève description sur l'étalonnage ainsi ses différentes catégories et on finit par la traçabilité pour comparer les résultats de mesure.
- Le quatrième chapitre est réservé pour l'étude de la partie soft et hard.
Nous allons donner une description détaillée sur le fonctionnement de l'logicielde programmation visuel studio (C#), la présentation du système Arduino UNO, ses caractéristiques et on mettra la lumière sur le logiciel IDE Arduino.
- Le cinquième chapitre explique l'application pratique, l'étude et la réalisation de notre système.
- Enfin, nous tirons une conclusion générale qui résume l'ensemble de notre travail effectué ainsi que des perspectives.

Chapitre 1

Généralité sur l'électronique et les systèmes embarqués

1.1 Introduction

L'**électronique** comprend la physique, l'ingénierie, l'informatique et les applications qui traitent de l'émission, du flux et du contrôle des électrons dans le vide et dans la matière [1].

L'électronique a eu un effet majeur sur le développement de la société moderne. L'identification de l'électron en 1897, ainsi que l'invention subséquente du tube à vide (Vacuum Tubes) qui pouvait amplifier et rectifier les petits signaux électriques, cette invention est un point de départ du domaine de l'électronique. Cette distinction a commencé vers 1906 avec l'invention de la triode par **Lee De Forest**, qui a rendu possible l'amplification électrique des signaux radio et audio faibles avec un appareil non mécanique. Jusqu'en 1950, ce domaine était appelé «technologie radio» car son application principale était la conception et la théorie des émetteurs radio, des récepteurs et des tubes à vide [2].

Dans ce chapitre nous allons faire un détour non exhaustif sur les principaux axes selon lesquels s'articulent la recherche et l'industrie dans ce domaine. On va d'abord aborder quelques définitions, composant, types de l'électronique, quelques illustrations des systèmes embarqués, pour ensuite considérer avec un aperçu des principales caractéristiques qui régissent les systèmes embarqués, pour terminer avec une architecture générale d'un système embarqué.

1.2 Les composants électroniques

Un composant électronique est toute entité physique dans un système électronique utilisée pour influencer sur les électrons ou leurs champs associés d'une manière cohérente avec la fonction prévue du système électronique. Les composants sont généralement destinés à être connectés ensemble, généralement en étant soudés à une carte de circuit imprimé [**Printed Circuit Board (PCB)**], pour créer un circuit électronique avec une fonction spécifique (par exemple un amplificateur, récepteur radio ou oscillateur). Les composants peuvent être emballés individuellement ou en groupes plus complexes sous forme de circuits intégrés. Les composants électroniques de base sont les condensateurs, inductances, résistances, diodes, transistors, etc. Les composants sont

souvent classés comme actifs (par exemple, transistors et thyristors) ou passifs (par exemple, résistances, diodes, inductances et condensateurs).

1.3 Types des circuits électroniques

Les circuits électronique et les composants peuvent être divisés en deux types : analogiques et numériques. Un dispositif électronique peut être constitué des circuits qui ont l'un ou l'autre ou mixte des deux types :

1.3.1 Circuits analogiques

Les circuits électroniques analogiques sont ceux dans lesquels le courant ou la tension peuvent varier continue avec le temps pour correspondre aux informations représentées. Les circuits analogiques sont construits à partir des deux éléments fondamentaux : les circuits série et parallèle. Dans un circuit série, le même courant traverse une série des composants.

Dans un circuit parallèle, tous les composants sont connectés à la même tension et le courant se répartit entre les différents composants en fonction de leur résistance.

Les composants de base des circuits analogiques sont les fils, les résistances, les condensateurs, les inductances, les diodes et les transistors. Les circuits analogiques sont très souvent représentés sur des schémas, dans lesquels les fils sont représentés sous forme des lignes, et chaque composant a un symbole unique.

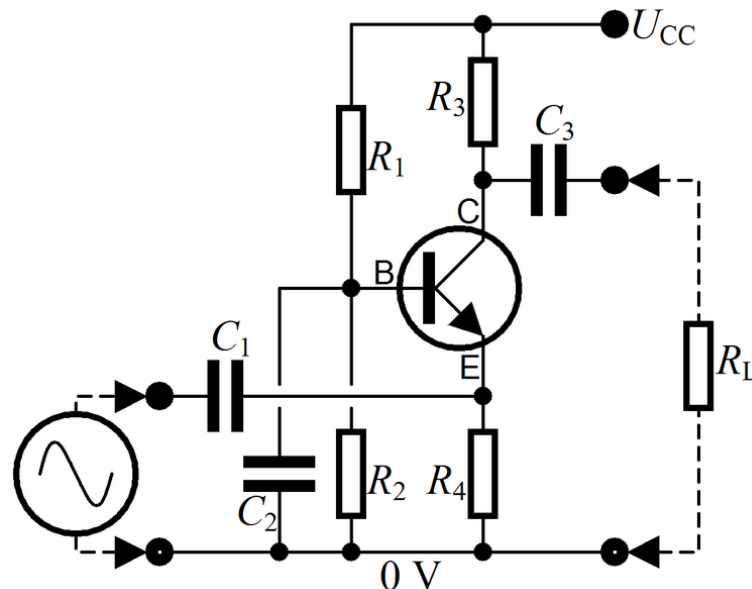


Fig. 1.1 – Un schéma de circuit analogique, dans ce cas un simple amplificateur

1.3.2 Circuits numériques

Dans les circuits électroniques numériques, les signaux électriques prennent des valeurs discrètes, pour représenter des valeurs logiques et numériques[3]. Ces valeurs représentent les infor-

mations en cours de traitement. Dans la grande majorité des cas, le codage binaire est utilisé : une tension (généralement la valeur la plus positive) représente un 1 binaire et une autre tension (généralement une valeur proche du potentiel de terre, 0 V) représente un «0» binaire. Les circuits numériques utilisent largement les transistors, interconnectés pour créer des portes logiques qui fournissent les fonctions de la logique booléenne : AND, NAND, OR, NOR, XOR et leurs combinaisons.

Les circuits numériques sont utilisés pour créer des cartes électronique à usage général, telles que des microprocesseurs et des circuits logiques personnalisés, **ASICs (Application-Specific Integrated Circuit)** . **FPGA (Field-Programmable Gate Arrays)**, des circuits logiques dont la configuration peut être modifiée après la fabrication, sont également largement utilisées dans le prototypage et le développement.



Fig. 1.2 – Un FPGA Stratix IV d'Altera

1.3.3 Circuits mixtes

Un circuit intégré à signaux mixtes ou bien un circuit hybride est un circuit intégré qui possède à la fois des circuits analogiques et des circuits numériques sur une seule puce semi-conductrice [4]. Dans les applications réelles, les conceptions à signaux mixtes sont partout, par exemple, les téléphones mobiles intelligents. Les circuits intégrés à signaux mixtes traitent également les signaux analogiques et numériques ensemble. Par exemple, un convertisseur analogique-numérique est un circuit à signaux mixtes. Les circuits ou systèmes à signaux mixtes sont généralement des solutions rentables pour la construction de toutes les applications électroniques grand public modernes.

1.4 Les systèmes embarqués

Un système embarqué est une combinaison d'un processeur informatique, d'une mémoire d'ordinateur et des périphériques d'entrée / sortie qui a une fonction dédiée au sein d'un système mécanique ou électrique à grand puissance [5].

Les systèmes embarqués modernes sont souvent basés sur des microcontrôleurs (c'est-à-dire des microprocesseurs avec mémoire intégrée et interfaces périphériques).

1.5 Historique

Les origines du microprocesseur et du microcontrôleur remontent au circuit intégré MOS, qui est une puce de circuit intégré fabriquée à partir de MOSFET (transistors à effet de champ à oxyde métallique et semi-conducteur) et a été développée au début des années 1960. En 1964, les circuits MOS avaient atteint une densité des transistors plus élevée et des coûts de fabrication inférieurs à ceux des circuits bipolaires. Les circuits MOS ont encore augmenté en complexité à un rythme prévu par la loi de Moore, conduisant à une intégration à grande échelle [**Large Scale Integration (LSI)**] avec des centaines des transistors sur un seul circuit MOS à la fin des années 1960. L'application des MOS LSI à l'informatique a été la base des premiers microprocesseurs, car les ingénieurs ont commencé à reconnaître qu'un système de processeur informatique complet pouvait être contenu sur plusieurs circuits MOS LSI [6].

Les premiers microprocesseurs à circuits multiples, les systèmes à quatre phases AL1 en 1969 et les Garrett AiResearch MP944 en 1970, ont été développés avec plusieurs MOS LSI. Le premier microprocesseur mono-circuit a été l'Intel 4004, sorti en 1971. Il a été développé par Federico Faggin, en utilisant sa technologie MOS à grille de silicium, avec les ingénieurs d'Intel Marcian Hoff et Stan Mazor, et l'ingénieur de Busicom Masatoshi Shima [7].

1.6 Caractéristiques d'un système embarqué

1.6.1 La fiabilité

De nombreux systèmes embarqués sont essentiels à la sécurité et doivent donc être fiables, la fiabilité est la probabilité qu'un système ne tombe pas en panne, la fiabilité englobe les aspects suivants d'un système :

- **Maintenabilité** : La maintenabilité est la probabilité qu'un système défaillant puisse être réparé dans un certain délai.
- **Disponibilité** : la disponibilité est la probabilité que le système soit disponible. La fiabilité et la maintenabilité doivent être élevées pour atteindre une haute disponibilité.
- **sûreté** : ce terme décrit la propriété qu'un système défaillant ne cause des catastrophes.
- **Sécurité** : Ce terme décrit la propriété que les données confidentielles restent confidentielles et que la communication authentique est garantie.

1.6.2 L'efficacité

Les systèmes embarqués doivent être efficaces. Les métriques suivantes peuvent être utilisées pour évaluer l'efficacité des systèmes embarqués :

- **Énergie** : de nombreux systèmes embarqués sont des systèmes mobiles qui obtiennent leur énergie à travers des batteries. Par conséquent, l'énergie électrique disponible doit être utilisée très efficacement.
- **Taille du code** : tout code à exécuter sur un système embarqué doit être stocké dans la mémoire de système, cela signifie que la taille du code devrait être petit que possible pour l'application prévue.
- **Poids** : Tous les systèmes portables doivent être de faible poids. Le faible poids est souvent un argument important pour acheter un certain système.
- **Coût** : pour les systèmes embarqués à volume élevé, en particulier chez les consommateurs électronique, la compétitivité sur le marché est une question extrêmement cruciale, et une utilisation efficace des composants matériels et le budget de développement logiciel sont nécessaires.

1.6.3 Systèmes hybrides

De nombreux systèmes embarqués sont des systèmes hybrides dans le sens où ils incluent pièces analogiques et numériques. Les parties analogiques utilisent des valeurs de signal continues en temps continu, tandis que les parties numériques utilisent des valeurs de signal discrètes en mode discret [8].

1.6.4 Les périphériques d'entrée/sortie (IO)

les systèmes embarqués sont connectés à l'environnement physique grâce à des capteurs collectant des informations sur cet environnement et des actionneurs contrôlant cet environnement.

1.7 Architecture d'un système embarqué

L'architecture d'un système embarqué se définit par le schéma suivant :

1. **FPGA** : (Field Programmable Gate Arrays) désigne un circuit intégré composé d'un réseau des cellules programmables. Chaque cellule est capable de réaliser une fonction, choisie parmi plusieurs possibles. Les interconnexions sont également programmables.
2. **Capteur** : un capteur est un dispositif permettant de détecter, en vue de le quantifier et de le représenter, un phénomène physique sous la forme d'un signal, généralement électrique.
3. **Actionneur** : un dispositif qui transforme l'énergie fournie en un phénomène physique qui fournit un travail.
4. **ASIC** : (Application Specific Integrated Circuit) en anglais, ou (Circuit intégré propre a une application) en français est un type de circuit électronique particulier. Il s'agit d'un circuit qui intègre, sur une même puce, l'ensemble des éléments actifs indispensables pour qu'une fonction ou qu'un ensemble électronique puisse se réaliser. Autrement dit, l'ASIC est un circuit intégré exclusivement dédié à une application et à un utilisateur. Il s'oppose en cela aux circuits intégrés standards qu'il fallait auparavant assembler sur plusieurs cartes afin d'accéder à une fonction précise.
5. **Convertisseur A/N** : un convertisseur analogique - numérique transforme une grandeur physique (tension, courant) en une valeur numérique. Généralement, il possède :

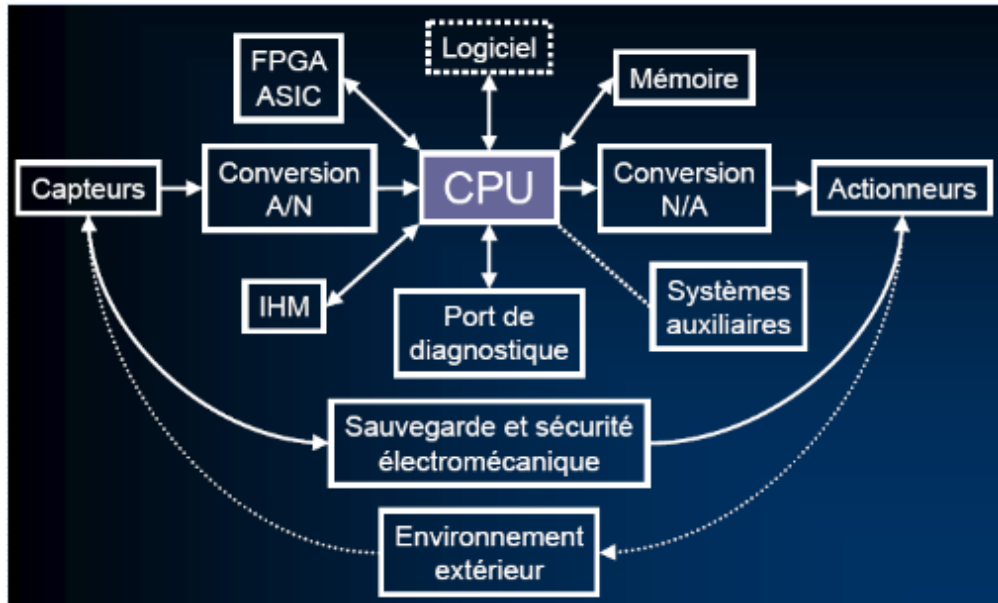


Fig. 1.3 – Architecture d'un système embarqué

- une entrée " début de conversion " qui permet de démarrer la conversion (Start).
- une sortie " fin de conversion " qui indique que la conversion est terminée (End).
- une entrée analogique (courant ou tension).
- plusieurs sorties numériques.

6. **Convertisseur N/A** : un convertisseur numérique-analogique permet de traduire une information numérique (binaire) en une information analogique.
7. **IHM** : (Interface Homme-Machine) est l'interface utilisateur qui relie l'opérateur au dispositif de commande d'un système. Certaines de ces interfaces traduisent les données des systèmes en représentations graphiques, textuelles, tactiles ou vocale lisibles par l'humain. Dans ce type d'interfaces l'opérateur voit des schémas des systèmes.
8. **CPU** : (Central Processing Unit) se dit (Unité Centrale de Traitement) en français, désigne la plus parts du temps le processeur d'un ordinateur, son rôle est de réaliser les différents calculs nécessaires au bon fonctionnement de l'ordinateur et d'exécuter les instructions, elle représente l'élément indispensable de la machine.

Cette architecture peut varier selon les systèmes : on peut par exemple, ne pas trouver des systèmes auxiliaires dans des nombreux systèmes embarqués autonomes et indépendants. En revanche, l'architecture de base est la plupart du temps composée d'une unité centrale de traitement (CPU), d'un système d'exploitation qui réside parfois uniquement en un logiciel spécifique, ou une boucle d'exécution. De même l'interface IHM n'est pas souvent existante, mais est souvent utile pour reconfigurer le système ou vérifier son comportement. Le fonctionnement du système se résume ainsi :

- Il reçoit des informations de l'environnement extérieur qu'il converti en signal numérique.
- L'unité de traitement composée du CPU, de la mémoire, du logiciel, de l'ASIC et éventuellement de système externe traite l'information.
- Le traitement génère éventuellement une sortie qui est envoyée vers la sortie.

1.8 Conclusion

Ce premier chapitre a présenté une brève description sur l'électronique et les systèmes embarqués a pour but de faire un tour absolument non exhaustif sur les différentes facettes que peut avoir ce domaine.

Ce qu'on peut retenir de ce chapitre, que le domaine des systèmes embarqués est un domaine vaste et pluridisciplinaire, qui a réussi à faire converger plusieurs disciplines a première vue totalement disjointes pour former une technologie des plus utilisées de nos jours. Les principaux domaines qui ont une relation directe ou indirecte avec les systèmes embarqués sont : les systèmes d'exploitation et la programmation système, le génie logiciel, la conception et/ou exploitation d'architecture matériel, l'électronique, bus de communication, réseaux de télécommunication, sécurité informatique. . .

Dans le chapitre suivant nous concentrerons à l'étude de la communication série.

Chapitre 2

Les techniques de la communication

2.1 Introduction

En général, la communication est peut être l'approche la plus utilisée pour transférer des informations entre les équipements de traitement des données et les périphériques. Chaque appareil peut être notre ordinateur personnel ou mobile fonctionne sur le protocole série. Le protocole est la forme de communication sécurisée et fiable ayant un ensemble des règles adressées par l'hôte source (expéditeur) et l'hôte de destination (récepteur).

Dans un système embarqué, la communication est le moyen d'échange des données en utilisant différentes méthodes sous forme de binaire numérique.

Dans ce chapitre on s'intéresse à l'étude de la communication série et parallèle avec ses différents modes de transmission, la différence entre eux et on se termine par les protocoles de communication.

2.2 Historique

Au début des années 1960, un comité de normalisation, aujourd'hui connu sous le nom d'Électronique Industries Association (EIA), a développé une norme d'interface commune pour les équipements de communication des données. À ce moment-là, la communication des données était censée signifier un échange des données numériques entre un ordinateur central et un terminal informatique distant, ou éventuellement entre deux terminaux sans un ordinateur impliqué. Ces appareils étaient connectés par des lignes téléphoniques et par conséquent requis un modem à chaque extrémité pour la traduction du signal. Bien que simples dans leur concept, les nombreuses possibilités d'erreurs des données survenues lors de la transmission des données via un canal analogique exigeait une conception relativement complexe. On pensait qu'une norme était d'abord nécessaire pour garantir communication fiable, et deuxièmement pour permettre l'interconnexion des équipements produits par différents fabricants, favorisant ainsi les avantages de la production de masse et de la concurrence [9]. De ces idées, Numéro standard recommandé 232, Révision C (RS232C) est né. Il a précisé tensions de signal, synchronisation du signal, fonction du signal, protocoles d'échange d'informations et mécanique connecteurs.

Depuis plus de 40 ans que cette norme a été élaborée, l'EIA a publié trois modifications, la plus récente étant la norme EIA232E introduite en 1991. Au-delà du changement le nom de la norme de RS232 à EIA232, certaines lignes de signaux ont été renommées et diverses nouvelles ont été définies, y compris un conducteur de blindage.

2.3 Les classifications de la communication

L'information élémentaire à transmettre est le mot binaire (constitué de 4, 8, 16, n bits).

Deux techniques sont mise en œuvre selon que tous les bits d'un même mot sont transmis au même temps (**transmission parallèle**) ou successivement (**transmission série**).

2.3.1 La transmission parallèle

La transmission parallèle est caractérisée par un transfert simultané de tous les bits d'un même mot. Elle nécessite autant de conducteurs qu'il y a des bits à transmettre et un conducteur commun (liaison asymétrique) ou autant de paires que de fils si la masse n'est pas commune (liaison symétrique). Un conducteur supplémentaire peut être utilisé pour transmettre un signal qui assurera la synchronisation entre les intervalles d'émission et ceux de réception. La synchronisation peut aussi être obtenue par lecture des transitions du signal reçu [10].

La transmission parallèle autorise une grande vitesse de transmission (débit). Un coût élevé (nombre de conducteurs) et une distance franchissable limitée réservent la transmission parallèle aux liaisons de processeur à processeur ou d'hôte à hôte. L'exemple le plus courant est la liaison imprimante-ordinateur.

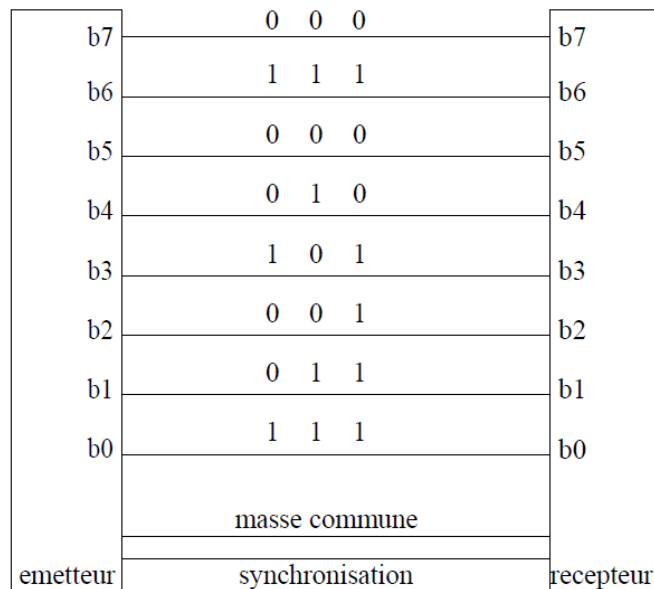


Fig. 2.1 – Transmission parallèle

2.3.2 La transmission série

En transmission série, tous les bits d'un mot ou d'un message sont transmis successivement sur une même ligne. Dans les calculateurs, les données (bits) sont traitées en parallèle (bus). La transmission série nécessite une interface de conversion pour sérialiser les bits à l'émission (conversion parallèle/série) et les désérialiser à la réception (conversion série/parallèle). La transmission série ne nécessite pour la transmission des données que deux conducteurs d'un coût moins important, elle est utilisée pour les transmissions sur des distances importantes [10].

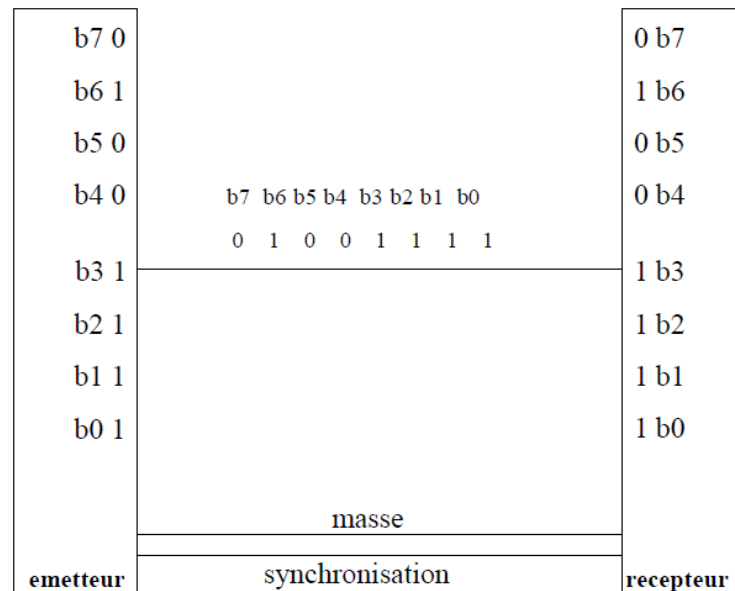


Fig. 2.2 – Transmission série

2.4 Différence entre la communication série et parallèle

Communication série	Communication parallèle
Envoie les données bit par bit à une impulsion.	Transfert un bloc de données à la fois.
La vitesse de communication est lente.	La vitesse de communication est rapide.
Le cout d'installation est faible.	Le cout d'installation est élevé.
Préférè pour la communication à longue distance.	Utilisé pour la communication à courte distance.
Nécessite un fil pour transmettre les données.	Nécessite 'n' nombre de lignes pour transmettre 'n'bits.
Exemple : ordinateur à ordinateur.	Exemple : Ordinateur à imprimante multifonction.

TABLE 2.1 – La comparaison entre la communication série et parallèle

2.5 Les modes de transmission

Les modes de transmission [11] sont classés comme **Simplex**, **Half Duplex** et **Full Duplex**. Il y aura une liaison point à point reliant deux stations émettrices / réceptrices pour chaque mode de transmission.

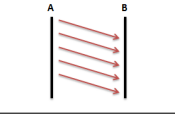
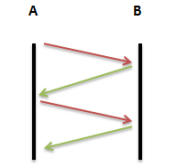
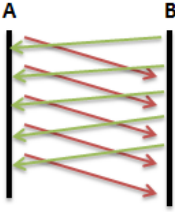
	<p>Mode simplex : l'une des stations émet et l'autre reçoit. La communication est donc unidirectionnelle pure.</p>
	<p>Mode semi-duplex (half duplex ou alternatif) : la communication est unidirectionnelle, mais le sens de transmission change alternativement : une station émet, l'autre reçoit, puis c'est la station réceptrice qui devient émettrice et réciproquement,... etc.</p>
	<p>Mode duplex (full duplex) : caractérise une liaison dans laquelle les données circulent de façon bidirectionnelle et simultanément. Ainsi, chaque extrémité de la ligne peut émettre et recevoir en même temps, ce qui signifie que la bande passante est divisée par deux pour chaque sens d'émission des données si un même support de transmission est utilisé pour les deux transmissions..</p>

TABLE 2.2 – Les modes de transmission

2.6 Les transmissions séries asynchrone et synchrone

Etant donné les problèmes qui pose la liaison de type parallèle, c'est la liaison série qui est la plus utilisée. Toutefois, puisqu'un seul fil transporte l'information, il existe un problème de synchronisation entre l'émetteur et le récepteur, c'est-à-dire que le récepteur ne peut pas distinguer les caractères (ou même de manière plus générale les séquences de bits) car les bits sont envoyés successivement. Il existe donc deux types de transmission permettant de remédier à ce problème :

2.6.1 La liaison asynchrone

Dans laquelle chaque caractère est émis de façon irrégulière dans le temps (par exemple un utilisateur envoyant en temps réel des caractères saisis au clavier). Ainsi, imaginons qu'un seul bit soit transmis pendant une longue période de silence... le récepteur ne pourrait savoir s'il s'agit de 00010000, ou 10000000 ou encore 00000100.

Afin de remédier à ce problème, chaque caractère est précédé d'une information indiquant le début de la transmission du caractère (l'information de début d'émission est appelée bit START) et terminé par l'envoi d'une information de fin de transmission (appelée bit STOP, il peut éventuellement y avoir plusieurs bits STOP) [12].

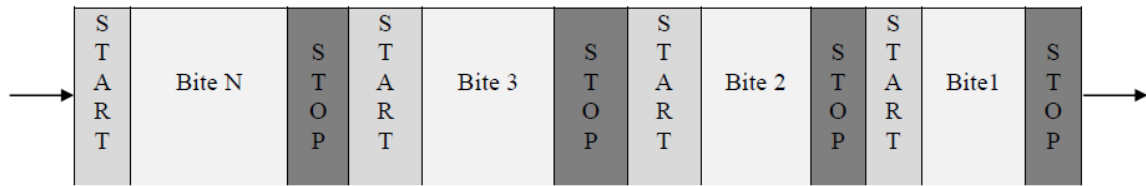


Fig. 2.3 – La liaison asynchrone

2.6.2 La liaison synchrone

Dans laquelle émetteur et récepteur sont cadencés à la même horloge. Le récepteur reçoit de façon continue (même lorsque aucun bit n'est transmis) les informations au rythme où l'émetteur les envoie. C'est pourquoi il est nécessaire qu'émetteur et récepteur soient cadencés à la même vitesse. De plus, les informations supplémentaires sont insérées afin de garantir l'absence d'erreurs lors de la transmission.

Lors d'une transmission synchrone, les bits sont envoyés de façon successive sans séparation entre chaque caractère, il est donc nécessaire d'insérer des éléments de synchronisation, on parle alors de **synchronisation au niveau caractère** [12].

Le principal inconvénient de la transmission synchrone est la reconnaissance des informations au niveau du récepteur, car il peut exister des différences entre les horloges de l'émetteur et du récepteur. C'est pourquoi chaque envoi des données doit se faire sur une période assez longue pour que le récepteur la distingue. Ainsi, la vitesse de transmission ne peut pas être très élevée dans une liaison synchrone.

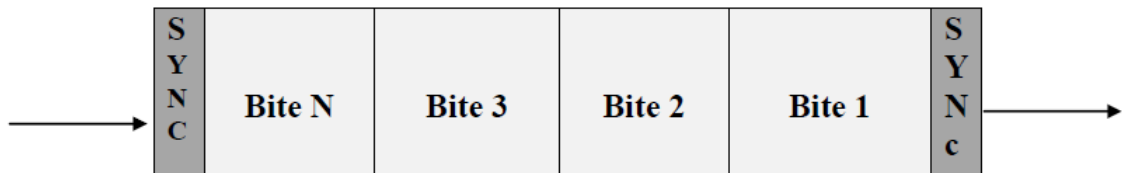


Fig. 2.4 – La liaison synchrone

2.7 Les protocoles série asynchrones

2.7.1 Protocole RS-232

Les interfaces série sont parmi les plus anciennes interfaces informatiques, antérieures à des nombreuses fonctionnalités des systèmes informatiques modernes, y compris le modèle OSI. Un certain nombre d'interfaces série différentes et des protocoles ont été définis au fil. La norme RS-232 a été créée en 1960 et a été fournie sur la plupart des ordinateurs personnels pendant des nombreuses années. Aujourd'hui, peu de PC fournissent un port série mais RS-232 est toujours

utilisé dans certains types d'équipements industriels. Les ports série ne fonctionnent pas à des vitesses élevées selon les normes modernes, mais leurs exigences matérielles et logicielles sont à la fois minimal [13].

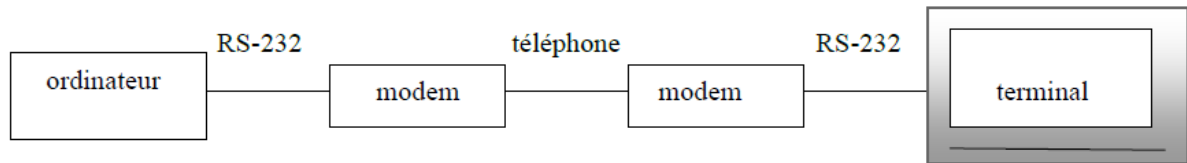


Fig. 2.5 – Première utilisation typique de RS-232

La **Fig. 2.5** montre un cas d'utilisation précoce typique des connexions RS-232, cette configuration aide à expliquer une partie de la terminologie utilisée dans la norme. Lorsqu'un ordinateur avait la taille d'une pièce, les utilisateurs souvent assis ailleurs et connectés à l'ordinateur via des lignes série. Les modems étaient utilisés pour transmettre des données sur la ligne téléphonique. Des liaisons RS-232 ont été utilisées pour connecter les modems au ordinateur et terminal, respectivement. Dans ce scénario, le terminal est un équipement terminal des données (DTE) et le modem est un équipement de terminaison de circuit des données (DCE).

2.7.2 Protocole USB

Le bus série universel, plus communément appelé USB, est utilisé dans des milliards d'informatique dispositifs. Étant donné son omniprésence dans l'informatique, il n'est pas surprenant qu'il soit utilisé pour interface système, soit pour se connecter à un PC hôte, soit pour connecter l'ordinateur à un autre appareil. L'USB a évolué sur plusieurs versions au cours des deux dernières décennies, offrant performances et autres fonctionnalités au fil du temps. USB 1.1 fonctionnait à 12 Mbit/s, USB 2.0 à 480 Mbit/s, USB 3.0 à 5 Gbit/s, USB 3.1 à 10 Gbit/s et USB 3.2 à 20 Gbit/s.

Du point de vue des applications s'exécutant sur l'hôte USB, les périphériques sur un bus USB fournissent des fonctions aux applications sur l'hôte. Comme le montre la **Fig. 2.6**, chaque application traite une fonction, il ne voit pas le bus dans son ensemble. Les applications utilisent des interfaces de programmation d'application (API) USB pour gérer les fonctions, et non les opérations de périphérique de bas niveau [14].

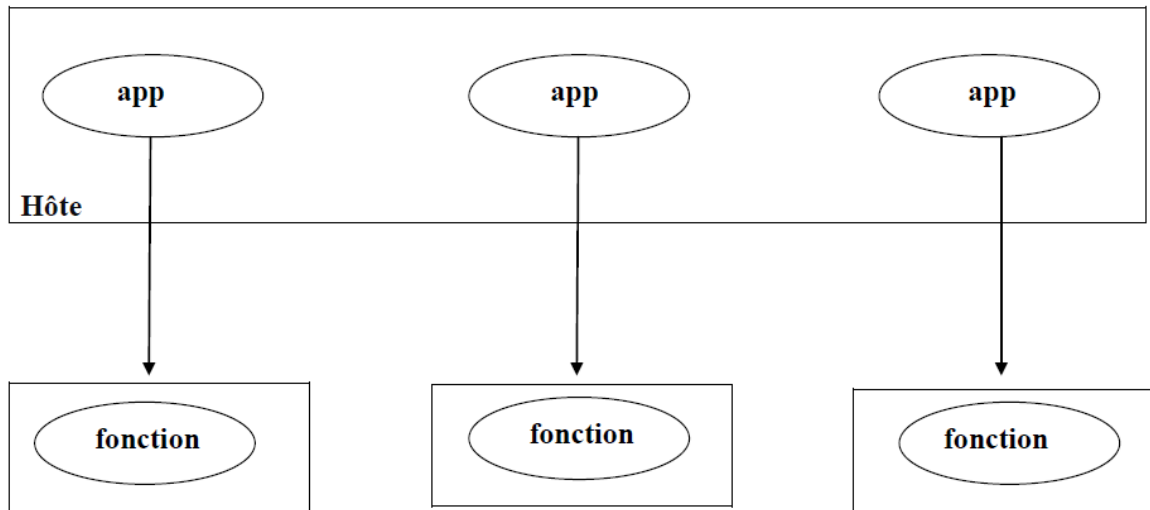


Fig. 2.6 – Applications et fonctions en USB

2.8 Protocoles série synchrones

Les protocoles de communication synchrones sont les meilleures ressources pour les périphériques embarqués. L'avantage est que vous pouvez interfacer plus d'appareils sur le même bus.

Certains des protocoles synchrones sont :

2.8.1 Protocole CAN

Ce protocole est dédié aux systèmes de véhicules ou automobiles. Il s'agit d'un protocole orienté message utilisé pour le câblage électrique multiplex pour économiser le cuivre. Il s'agit d'un bus multi-série multi-maître utilisé dans des applications telles que le démarrage / arrêt automatique des véhicules, les systèmes anticollision...etc.[9].

2.8.2 Protocole WIFI

Le **Wi-Fi** (Wireless Fidelity) [13] est une bonne solution pour profiter librement de beaucoup de produit high-tech : netbook, PC portable, lecteur multimédia de salon, tablette Internet, smartphone, imprimante, Le protocole de communication est basé sur le principe d'une connexion Ethernet.

Présent dans la majorité des appareils actuels, il assure une connexion sans-fil qui est parfois sujette à des déconnexions intempestives. Ce problème survient lorsque la distance est importante ou que l'habitation offre de nombreux obstacles et que le réseau Wi-Fi fonctionne avec la norme **Wi-Fi g** (seulement 54 Mbits/s). La norme **Wi-Fi n** corrige généralement ces difficultés grâce à un débit bien supérieur (300 Mbits/s).

Il est nécessaire d'un point de vue pour s'assurer que les équipements (pc portable, netbook,) intègrent cette norme, tout comme la box Internet.

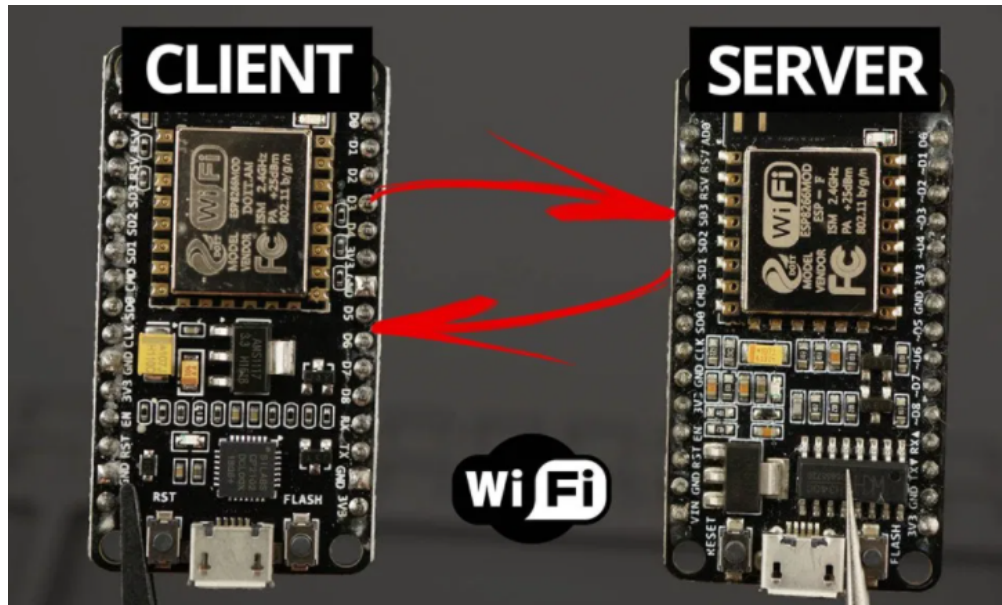


Fig. 2.7 – Communication WIFI client-serveur ESP8266 entre deux cartes

2.9 Conclusion

Au cours de ce chapitre, on a découvert que la communication est la partie vitale dans le domaine d'électronique et systèmes embarqués. Le taux de transfert des données est critique si deux appareils veulent échanger des informations sur le même bus. Par conséquent, il est nécessaire de choisir un protocole valide pour toute application, et que **la transmission parallèle** est utilisée pour une distance plus courte, fournit une plus grande vitesse. Tandis que, **la transmission série** est fiable pour le transfert des données à plus longue distance. Pour conclure, les deux transmissions **série et parallèle** sont importantes pour le transfert des données.

Après cette étude de la communication, nous établirons dans le chapitre suivant les processus d'étalonnage.

Chapitre 3

Les processus d'étalonnage et tests

3.1 Introduction

L'étalonnage des instruments est l'un des principaux processus utilisés pour maintenir la précision des instruments.

La première utilisation des mots «étalonnage» et «étalonner» remonte à la guerre civile américaine. Pendant cette période, les mots entraient officiellement dans la langue anglaise dans les descriptions d'artillerie. À l'époque, les premiers exemples d'appareils de mesure étaient perceptifs et relativement faciles à vérifier.

Par la suite, nous présenterons une idée générale sur les processus d'étalonnage, ses catégories, la traçabilité, les rapport d'exécution des tests.

3.2 Définition

Selon la publication de l'organisation internationale de normalisation intitulée vocabulaire international des termes fondamentaux et généraux de métrologie (publiée en 1993 et connue sous le nom de VIM), l'étalonnage est l'ensemble des opérations qui établissent, dans des conditions spécifiées, la relation entre les valeurs indiquées par un instrument de mesure, un système de mesure ou les valeurs représentées par une mesure matérielle et les valeurs connues correspondantes d'un mesurande la compréhension de l'étalonnage n'est pas complète sans comprendre la traçabilité [17].

- **Traçabilité** : le concept d'établir un étalonnage valide d'un étalon ou d'un instrument de mesure par comparaison étape par étape avec de meilleurs étalons jusqu'à un étalon national ou international accepté.
- **Norme (étalon)** : une norme est une mesure matérielle ou une propriété physique qui définit ou reproduit l'unité de mesure d'une quantité de base ou dérivée [18].

Cependant, une norme doit également être vérifiée par rapport à une norme plus élevée pour établir sa précision et sa traçabilité. Étant donné que le même argument serait valable même pour la norme supérieure, cette hiérarchie des normes doit conduire à un niveau au-dessus duquel la comparaison n'est pas possible.



Fig. 3.1 – L'étalonnage des instruments électroniques

3.3 Le Système international d'unités (SI)

Le système SI est le système international d'unités qui définit les unités de base utilisées en science de la mesure. Le système SI définit 7 unités de base (le mètre, le kilogramme, la seconde, l'ampère, le kelvin, la mole et la candela) ainsi que 22 unités dérivées [19].

Les unités de base proviennent des constantes naturelles physiques. C'est le BIPM (Bureau International des Poids et Mesures) qui supervise le système SI.



Fig. 3.2 – Le Système international d'unités (SI)

3.4 Traçabilité et Normes

3.4.1 Traçabilité et hiérarchie d'étalonnage

Pour pouvoir comparer les résultats de mesure, ils doivent être "traçables" à une norme nationale ou internationale via une chaîne de comparatifs des mesures. À cette fin, les valeurs affichées de l'instrument de mesure utilisé ou d'un étalon de mesure sont comparées sur une ou plusieurs étapes à cette norme. À chacune de ces étapes, l'étalonnage avec un étalon préalablement étalonné avec un classement supérieur est effectué.

Conformément à l'ordre de classement des normes :

- À partir de la norme de travail ou de la norme d'usine et de l'étalon de référence à l'étalon national.
- Les organismes d'étalonnage ont une hiérarchie d'étalonnage. Cela va du laboratoire d'étalonnage interne au laboratoire d'étalonnage accrédité et au laboratoire national institut métrologique (**Fig. 3.3**). Le service d'étalonnage allemand DKD (Deutscher Kalibrierdienst) désigne les éléments

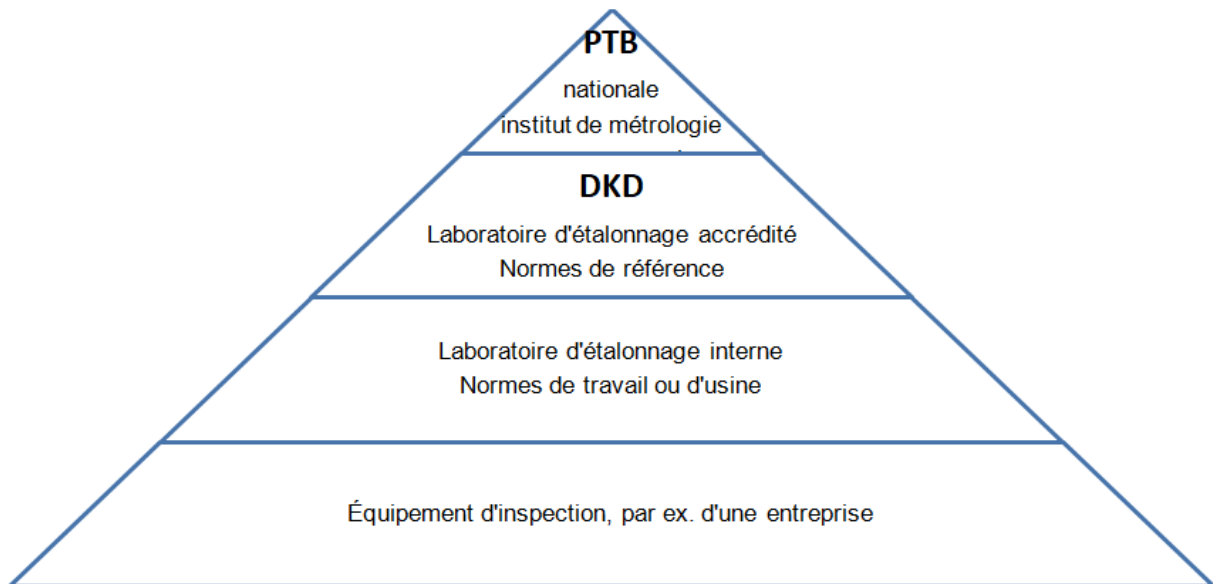


Fig. 3.3 – Hiérarchie d'étalonnage décrite par l'exemple de l'Allemagne.

suivants comme éléments essentiels de la traçabilité [20] :

- La chaîne de comparaison ne doit pas être interrompue.
- À chaque étape de la chaîne d'étalonnage, l'incertitude de mesure doit être connue afin que l'incertitude de mesure totale puisse être calculée. En règle générale, un instrument de mesure de rang supérieur doit avoir une précision de mesure trois à quatre fois supérieure à l'instrument calibré par lui.
- Chaque étape de la chaîne d'étalonnage doit être documentée, tout comme le résultat.
- Tous les organismes réalisant une étape de cette chaîne doivent prouver leur compétence par accréditation.
- Selon la précision de mesure requise et les exigences techniques, les étalonnages doivent être répétés à intervalles appropriés.

3.4.2 Les diverses catégories d'étalons (normes)

- **Étalon national** : étalon primaire reconnu par une décision nationale, dans un pays, pour servir de base à l'attribution de valeurs aux autres étalons de la grandeur concernée.
- **Étalon secondaire** : étalon dont la valeur est établie par comparaison à un étalon primaire de la même grandeur.
- **Étalon de référence** : étalon, en général de la plus haute qualité métrologique disponible en un lieu donné ou dans une organisation donnée, dont dérivent les mesurages qui y sont faits.
- **Étalon de travail** : étalon qui est utilisé couramment pour étalonner ou contrôler des mesures matérialisées, des appareils de mesure ou des matériaux de référence.
- **Étalon de transfert** : étalon utilisé comme intermédiaire pour comparer entre eux des étalons de référence : étalon, en général de la plus haute qualité.

3.5 Certificat d'étalonnage

La définition de l'étalonnage inclut le mot **documenté**. Cela signifie que toute comparaison d'étalonnage doit être enregistrée. Ce document s'appelle la plupart du temps **certificat d'étalonnage**.

Un certificat d'étalonnage inclut le résultat de la comparaison ainsi que toute autre information pertinente à l'étalonnage comme le type d'équipement utilisé, les conditions environnementales, les signataires, la date de l'étalonnage, le numéro de certificat, les incertitudes de mesure...etc.

pehamed
FEHA MED. GERÄTE GMBH

CALIBRATION CERTIFICATE

No. of Calibration Certificate : 005-03/2006
 Calibrated Device : testfilm
 S/N : 005
 Client : G.I.P.S.
 Calibration Date : 31.03.2006

Procedure: The calibration of the Testfilm was made by comparison of densitometric measurements with the densitometer "Gretag D200 s/n 3246 11885" which was used as reference unit. The referencdensitometer "Gretag D200 s/n 3246 1185" was calibrated by a national standard acc. Calibration standards.

Step values	2	3	4	5	6	7
1	0,21	0,21	0,22	0,24	0,27	0,32
8	0,52	0,71	0,97	1,32	1,75	2,21
15	2,97	3,20	3,32	3,39	3,44	3,47

This Calibration Certificate is only valid for the testfilm with the above mentioned serial number.
 The calibration of this testfilm is valid for one year!

Sutzbach, 31.03.2006

Lutz Kippner

PEHAMED Geräte GmbH
Münsterstr. 28
D-65840 Sutzbach | H. (Germany)

Tel: 06196 / 60040-20
Fax: 06196 / 60040-10
info@pehamed.de

Geschäftsführer: Klaus Schwenhagen
HRB Frankfurt/B. 24206
IC Nr. DE 113810211

Fig. 3.4 – Exemple de Certificat d'étalonnage

3.6 Les rapports des tests

3.6.1 TAR et TUR

On utilise parfois le rapport d'exactitude de test [**Test Accuracy Ratio (TAR)**] et le rapport d'incertitude de test [**Test uncertainty Ratio (TUR)**] pour exprimer la différence entre l'instrument à étalonner et l'étalon de référence utilisé. Il s'agit du rapport entre l'exactitude (ou l'incertitude) entre l'instrument étalonné et l'étalon de référence.

On parle souvent d'un rapport TAR de 4 pour 1, ce qui signifie que l'étalon de référence est quatre fois plus précis que l'unité étalonnée (DUT ou device under test en anglais). Cela signifie que les spécifications d'exactitude de l'étalon de référence doivent être quatre fois meilleures (inférieures) que celle de l'unité étalonnée. L'utilisation des rapports TAR/TUR (par exemple 4 pour 1) permet de s'assurer que l'étalon de référence est suffisamment bon pour l'application donnée [21].

Cependant, souvenez-vous que le TAR, par exemple, ne prend en compte que les spécifications d'exactitude de l'instrument à étalonner et n'inclut pas les incertitudes des différents composants du processus d'étalonnage. En fonction du type d'étalonnage, les incertitudes de certains composants peuvent parfois être plus importantes que l'exactitude spécifiée. On recommande de toujours calculer l'incertitude totale d'un étalonnage.

3.7 Incertitude d'étalonnage (incertitude de mesure)

Lors de l'étalonnage d'un instrument avec un appareil de plus haut niveau, le procédé inclut toujours une certaine incertitude. L'incertitude représente la quantité de **doute** dans un procédé d'étalonnage, elle exprime donc à quel point l'étalonnage est **bon**. Les sources d'incertitude sont variées et incluent l'instrument testé, l'étalon de référence, la méthode d'étalonnage ou encore les conditions environnementales.

Dans le pire des cas, si l'incertitude d'étalonnage est supérieure à la précision ou à la tolérance autorisée de l'instrument à étalonner, alors l'étalonnage n'a aucun sens. Le but final est que l'incertitude totale d'étalonnage soit suffisamment faible par rapport à la limite de tolérance de l'unité étalonnée. L'intolérance totale d'étalonnage doit toujours apparaître dans le certificat d'étalonnage.

3.8 Intervalle d'étalonnage

Les étalonnages doivent être répétés à intervalles appropriés [22]. Un instrument de mesure, par exemple, devrait être périodiquement recalibré car des changements dans ses caractéristiques peuvent se produire pendant son utilisation et après un certain temps. Le recalibrage à intervalles appropriés assure la détection de ces changements possibles. La longueur de ces intervalles dépendra d'un certain nombre de variables, telles que :

- Incertitude requise.
- Fréquence d'utilisation.
- Mode d'utilisation.

- Conditions environnementales d'utilisation.
- Stabilité de l'équipement.

3.9 Documentation des résultats d'étalonnage

Les résultats obtenus par un étalonnage sont documentés dans un certificat d'étalonnage ou un rapport d'étalonnage. L'incertitude de mesure associée doit être indiquée.

Les informations obtenues avec l'étalonnage peuvent être exprimées par un énoncé, une fonction, une correction additive ou multiplicative de l'indication, un diagramme d'étalonnage ou une courbe d'étalonnage. Pour ces derniers, le Vocabulaire International de Métrologie donne les définitions suivantes :

- **Diagramme d'étalonnage** : expression graphique de la relation entre l'indication et le résultat de mesure correspondant.

Remarque 1 : un diagramme d'étalonnage est la bande du plan définie par l'axe de l'indication et l'axe du résultat de mesure qui représente la relation entre une indication et un ensemble des valeurs de quantité mesurées. Une relation un à plusieurs est donnée, et la largeur de la bande pour une indication donnée fournit l'incertitude de mesure instrumentale.

Remarque 2 : d'autres expressions de la relation incluent une courbe d'étalonnage et l'incertitude de mesure associée, une table d'étalonnage ou un ensemble de fonctions.

Remarque 3 : ce concept concerne un étalonnage lorsque l'incertitude de mesure instrumentale est grande par rapport aux incertitudes de mesure associées aux valeurs de quantité des étalons de mesure [23].

- **Courbe d'étalonnage** : expression de la relation entre l'indication et la valeur mesurée correspondante.

Remarque : une courbe d'étalonnage exprime une relation biunivoque qui ne fournit pas de résultat de mesure car elle ne contient aucune information sur l'incertitude de mesure [23].

Alors qu'une courbe d'étalonnage nécessite une déclaration distincte pour l'incertitude de mesure, un diagramme d'étalonnage fournit une déclaration complète des informations obtenues avec l'étalonnage, y compris la valeur mesurée et l'incertitude de mesure associée.

3.10 L'importance d'étalonnage

Dans le cadre d'un procédé industriel, les raisons d'étalonner sont nombreuses. Voici un exemple des plus communs [24] :

- L'exactitude de toutes les mesures diminue avec le temps.
- Les réglementations exigent des étalonnages réguliers.
- Le système de qualité requiert des étalonnages.
- L'argent : certains paiements dépendent des résultats de mesure.
- La qualité des produits fabriqués.
- La sécurité des clients comme celle des employés.
- Des considérations environnementales.

3.11 Conclusions

Dans cette partie, on a découvert que l'étalonnage des instruments de procédé consiste à comparer et documenter la mesure d'un instrument avec celle d'un étalon de référence traçable. L'étalonnage est essentiel pour nous assurer que nos mesures sont valides.

le chapitre qui suit fera l'objet de définir la partie soft et hard de notre système.

Chapitre 4

Matériels et logiciels

4.1 Introduction

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

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toute personne en ayant l'envie. Comme tout projet dans le domaine, l'utilisation d'un micro contrôleur est inévitable, dans notre cas nous utiliserons Visual studio Basic (C#), qui sera détaillé par la suite, suivi par une présentation de l'ArduinoUNO.

Donc l'objectif de ce chapitre est de mettre l'accent sur la description des différents matériels et logiciels qu'on a utilisés pour la réalisation, la programmation et la création d'une application (interface graphique).

4.2 Soft

4.2.1 GUI avec des objets physiques

Les systèmes embarqués vont de l'absence d'interface utilisateur (dans les systèmes dédiés à une seule tâche), aux interfaces utilisateur graphiques complexes qui ressemblent aux systèmes d'exploitation développés. Les appareils embarqués simples utilisent des boutons, des LED, des écrans LCD avec un système de menu simple.

Des appareils plus sophistiqués qui utilisent un écran tactile dans lesquels une application est installée pour contrôler plusieurs périphériques offrent une flexibilité tout en minimisant l'espace utilisé :par exemple nous pouvons changer les boutons physiques avec d'autres virtuelles.

4.2.2 Visual Basic

Une brève histoire de Visual Basic

Visual Basic est un langage de programmation événementiel de troisième génération publié pour la première fois par Microsoft en 1991. La version finale de Visual Basic classique était Visual Basic 6. Ce dernier est un langage de programmation convivial conçu pour les débutants. Il permet à quiconque de développer facilement des applications Windows GUI. De nombreux développeurs préfèrent toujours VB6 à son successeur VB.NET.

En 2002, Microsoft a publié Visual Basic.NET (VB.NET) pour remplacer Visual Basic 6. Par la suite, Microsoft a déclaré VB6 un langage de programmation hérité en 2008. Cependant, Microsoft fournit toujours une certaine forme de prise en charge pour VB6.

VB.NET est un langage de programmation entièrement orienté objet implémenté dans le .NET Framework. Il a été créé pour répondre au développement du Web ainsi que des applications mobiles. Par la suite, Microsoft a publié de nombreuses versions de VB.NET. Il s'agit de Visual Basic 2005, Visual Basic 2008, Visual Basic 2010, Visual Basic 2012, Visual Basic 2013, Visual Basic 2015, Visual Basic 2017 et Visual Basic 2019. Bien que la partie .NET a été supprimé en 2005, toutes les versions du langage de programmation Visual Basic publiées depuis 2002 sont considérées comme le langage de programmation VB.NET [25].

Microsoft a publié Visual Studio 2019 au début de 2019. VS 2019 nous permet de coder dans différents langages de programmation et différents plates-formes, Visual Basic 2019 est l'un d'eux. Les autres langages de programmation sont C#, C ++, F#, JavaScript, Java et Python.

Visual Basic 2019 est la dernière version du langage de programmation VB.NET publié par Microsoft.

Installation de Visual Studio 2019

Nous pouvons télécharger la version gratuite de Visual Studio 2019 à partir du site officiel suivant : (<https://visualstudio.microsoft.com/vs/>) Cliquons sur le lien pour afficher la page de téléchargement de Visual Studio 2019, comme indiqué dans **la Fig.4.1** éditions de Visual Studio 2019.

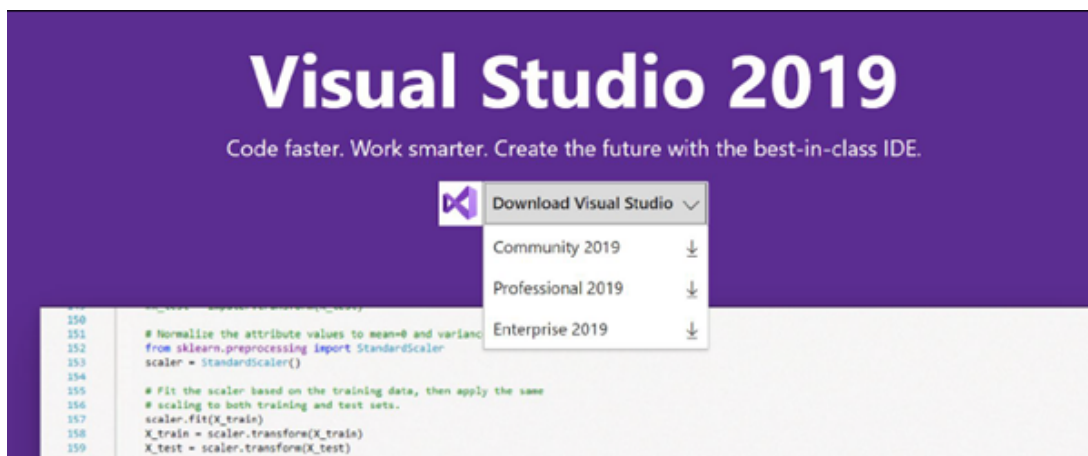


Fig. 4.1 – Page de téléchargement Visual Studio 2019

Nous pouvons choisir la communauté Visual Studio gratuite 2019 ou la solution professionnelle complète 2019 et de bout en bout entreprise 2019 à télécharger. La version gratuite fournit un IDE complet pour les étudiants, la communauté open source et les particuliers.

Création d'un projet Visual Basic 2019

Pour créer un projet Visual Basic 2019, lançons Microsoft Visual Studio 2019. Le lancement de VS 2019 fera apparaître la page de démarrage de Visual Studio 2019, comme indiqué dans la Fig.4.2.

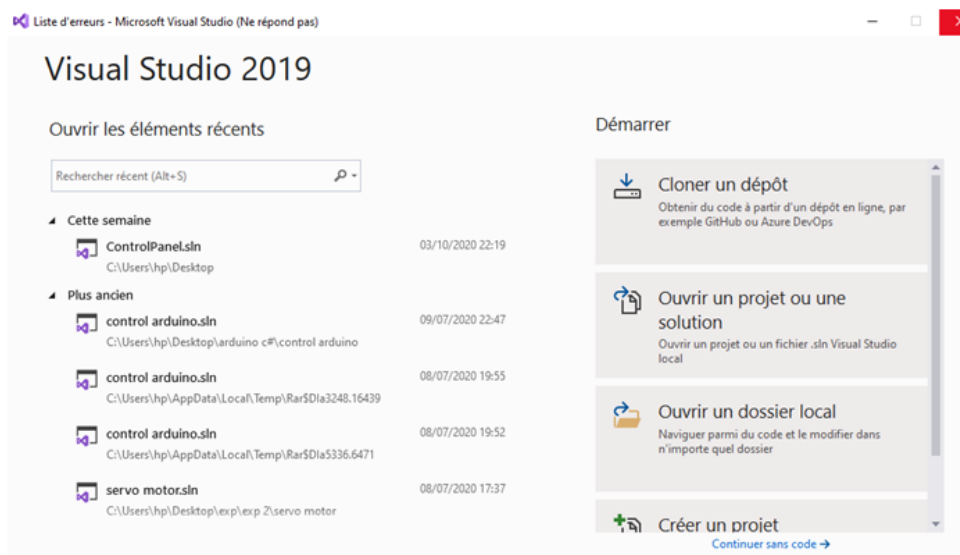


Fig. 4.2 – Page de démarrage de Visual Studio 2019

La page de démarrage de Visual Studio 2019 comprend deux sections, la section Ouvrir récent et la section Prise en main. Dans la page de démarrage, nous pouvons sélectionner un fichier de projet récent ou choisir n'importe quelle option dans la section Prise en main. Nous pouvons choisir de cloner un projet depuis GitHub ou Azure DevOps, ouvrir un projet ou une solution, ouvrir un dossier local, créer un nouveau projet, ou continuer sans code.

Créons un nouveau projet en cliquant sur l'option Créer un nouveau projet. Nous verrons la page Créer un nouveau modèle de projet, comme illustré dans la **Fig. 4.3**. Dans la page Créer un nouveau projet, sélectionnons le langage Visual Basic dans notre cas nous utiliserons C#.

Ensuite, sélectionnons le modèle d'application Windows Forms (.NET Framework) comme nous voulons développer un projet de bureau Windows, comme illustré à la **Fig. 4.3**.

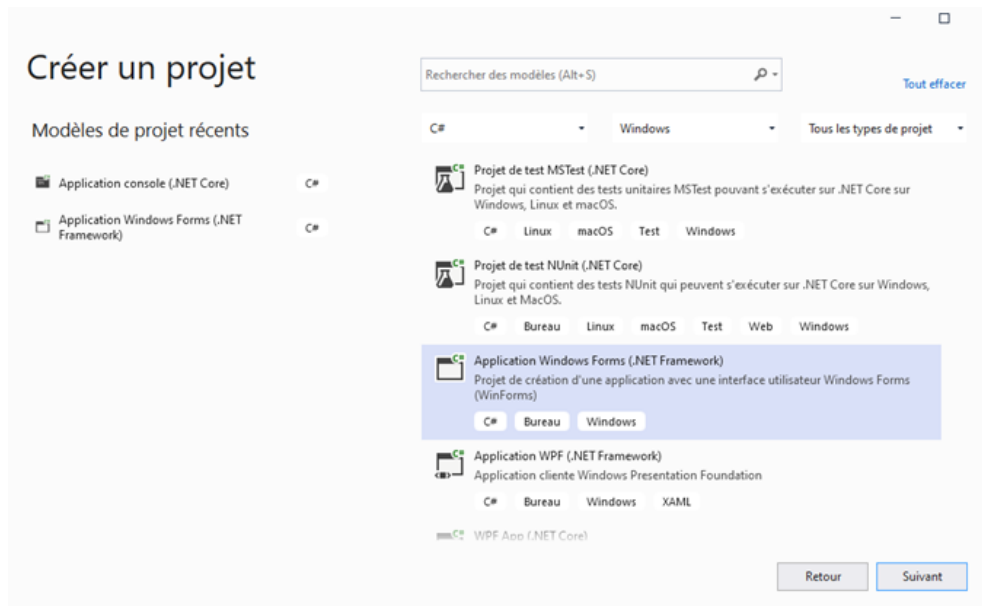


Fig. 4.3 – Créer un nouveau modèle de projet

En cliquant sur le modèle de projet sélectionné, la page de configuration du projet apparaît, comme illustré à la **Fig. 4.4**. Nous pouvons configurer notre projet en tapant le nom du projet et en sélectionnant quelques autres options.

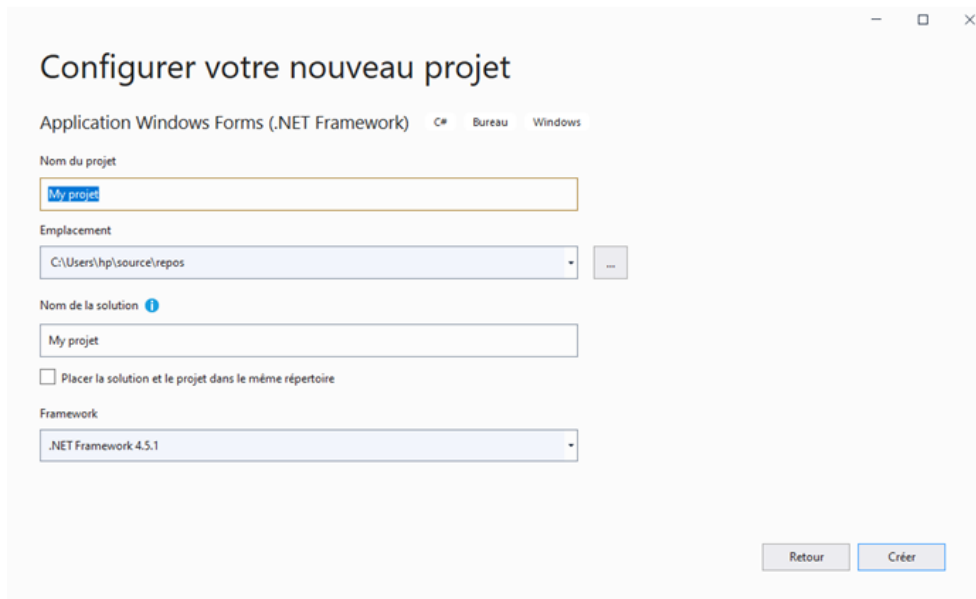


Fig. 4.4 – Configuration du projet

Au bas de cette boîte de dialogue, nous pouvons modifier le nom de projet par défaut `WindowsApplication1` en un autre nom que nous aimons, par exemple, `mon projet`. Après avoir renommé le projet, cliquons sur OK pour continuer. L'EDI Windows de Visual Basic 2019 apparaîtra, comme illustré à la Fig. 4.5. Visual Basic 2019 IDE comprend quelques fenêtres, la fenêtre Formulaire, la fenêtre Explorateur des solutions et la Fenêtre Propriétés. Il se compose également d'une boîte à outils contenant des nombreux contrôles utiles qui permettent au programmeur de développer ses programmes Visual Basic 2019.

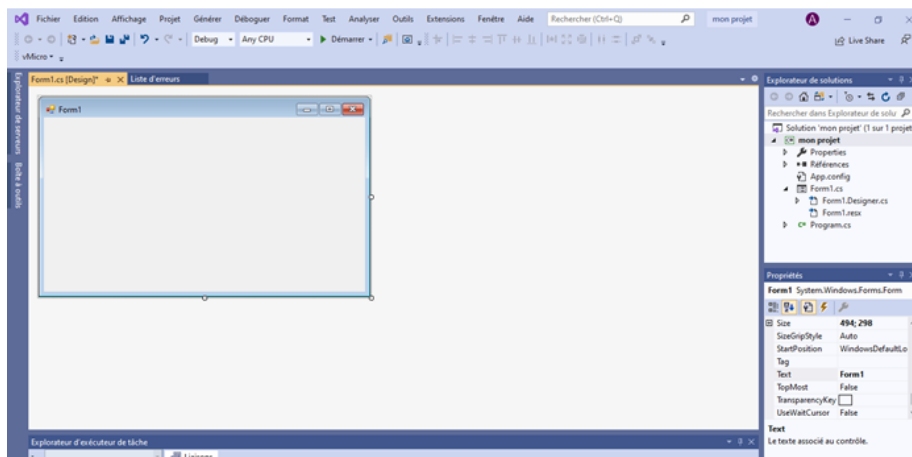


Fig. 4.5 – L'IDE de Visual Basic 2019 Express

La boîte à outils ne s'affiche pas tant que nous n'avons pas cliqué sur l'onglet Boîte à outils. Lorsque nous cliquons sur l'onglet Boîte à outils ou utilisons les touches de raccourci `Ctrl + Alt + x`, la boîte à outils des commandes communes apparaîtra, comme la montre la Fig. 4.6. Nous pouvons faire glisser et déplacer notre boîte à outils et l'ancrer à droite, à gauche, en haut ou en

bas de l'EDI.

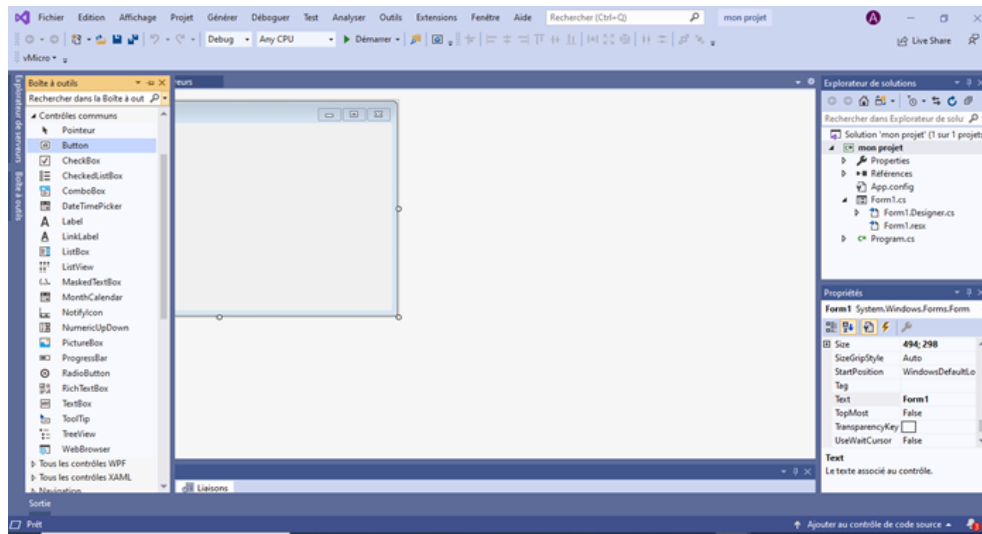


Fig. 4.6 – Boîte à outils Visual Basic 2019

Ensuite, nous allons vous montrer comment créer notre première application VB2019. Tout d'abord, changeons le texte du formulaire en «Ma première application VB 2019» dans la fenêtre des propriétés; il apparaîtra comme le titre de l'application. Ensuite, insérons un bouton et changeons son texte en OK. L'interface de conception est illustrée à la **figure 4.7**.

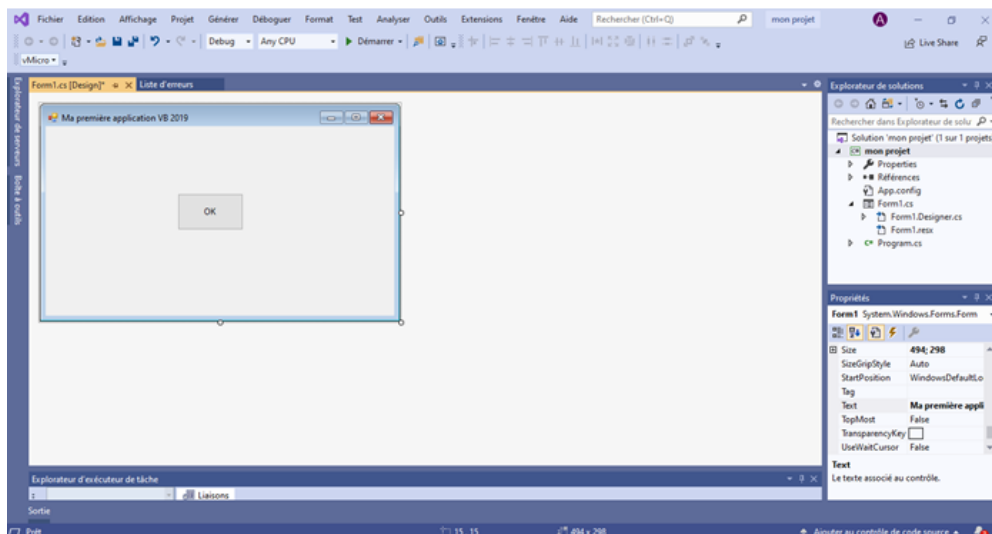


Fig. 4.7 – L'interface de conception

Cliquons maintenant sur le bouton OK pour afficher la fenêtre de code et entrons la déclaration suivante, comme illustré dans la **Fig. 4.8** :
(`MessageBox.Show ("Ma première application Visual Basic 2019")`).

Cliquer sur le bouton Démarrer de la barre d'outils ou appuyer sur F5 pour exécuter l'application lancera l'interface d'exécution, comme illustré à la **Fig. 4.8**. L'exécution de l'application en

cliquant sur le bouton OK fera apparaître une boîte de dialogue qui affiche le message «Ma première application Visual Basic 2019», comme illustré à la **Fig. 4.9**. La fonction `MessageBox.Show` est une fonction intégrée fonction de Visual Basic 2019 qui peut afficher le texte inclus dans les supports.

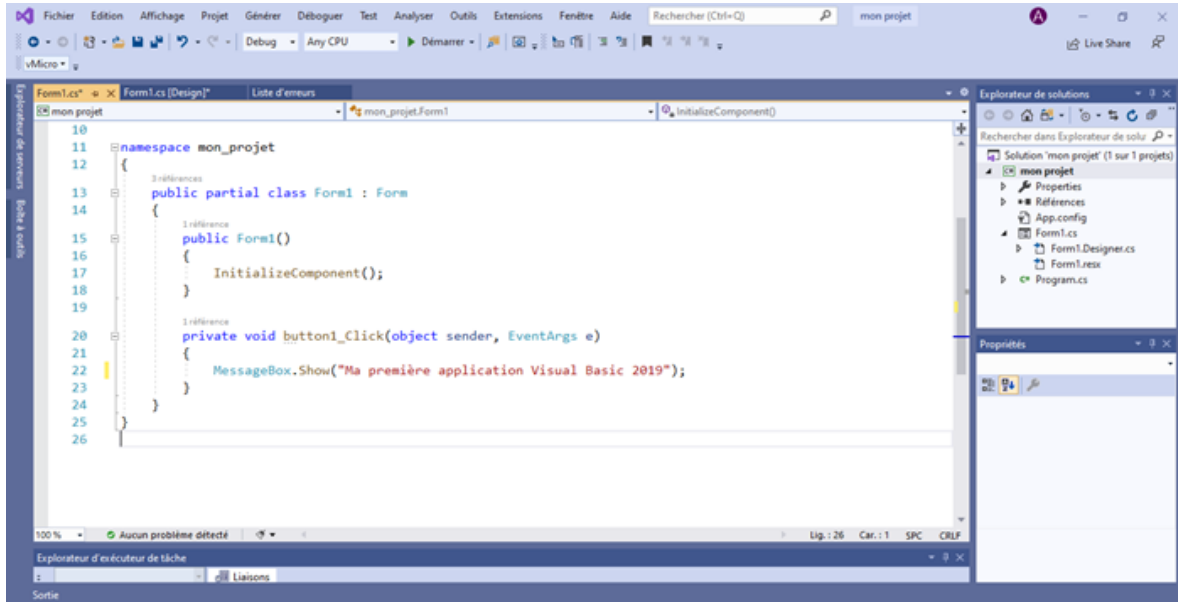


Fig. 4.8 – Fenêtre de code Visual Basic 2019

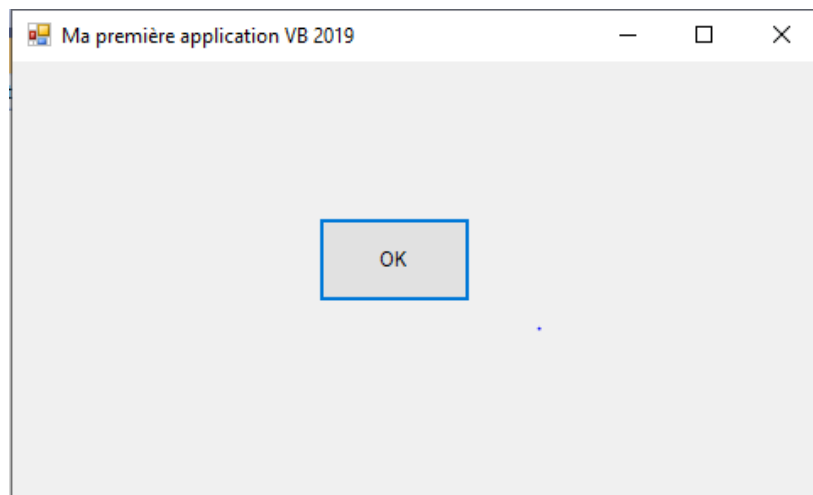


Fig. 4.9 – L'interface d'exécution

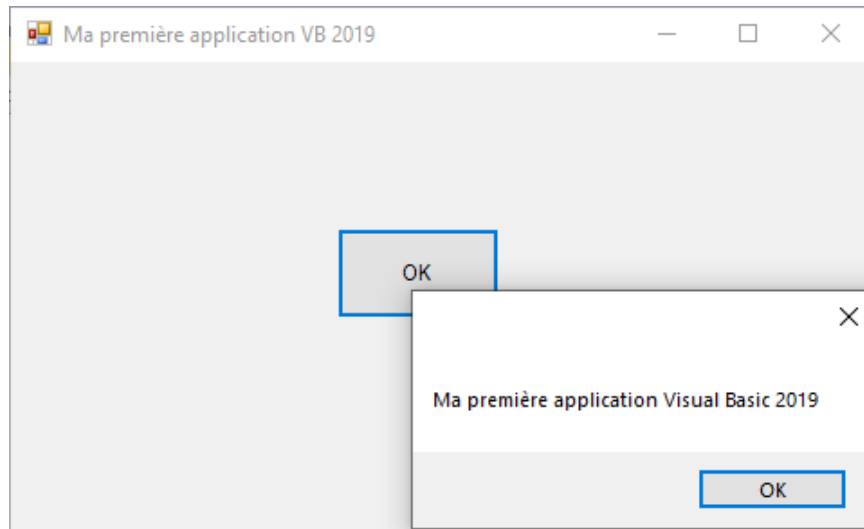


Fig. 4.10 – La boîte de message

4.3 Hard

4.3.1 Présentation de la carte Arduino

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Sans tout connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités des relations humain/machine ou environnement/machine. Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions [26].

L'Arduino est une plateforme de contrôle, elle est constituée de deux choses :

- **Le logiciel (Software)** : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.
- **Partie Hardware (Le matériel)** : cartes électroniques dont les schémas sont en libre circulation sur internet.

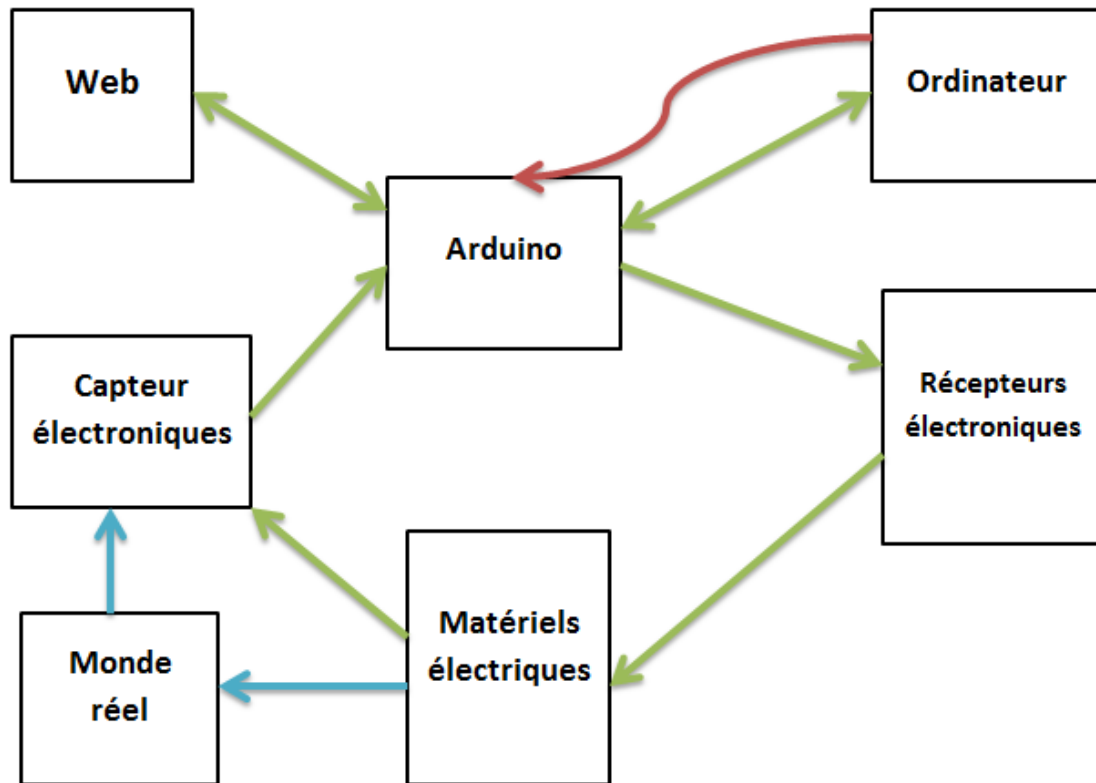


Fig. 4.11 – Fonctionnement de l'Arduino

4.3.2 Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques uns afin d'éclaircir l'évaluation de ce produit scientifique et académique [27] :

- **L'Arduino Mega**, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- **L'Arduino UNO**, utilisations microcontrôleur ATmega328.
- **L'Arduino Mega2560**, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).

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



Fig. 4.12 – La carte Arduino UNO

4.3.3 Caractéristiques de la carte Arduino UNO

Microcontrôleur	ATmega328
Tension de fonctionnement	5 V
Tension d'entrée (recommandée)	7 à 12 V
Tension d'entrée (limite)	6 à 20 V
Broches E/S numériques	14 dont 6 sorties PWM
Broches d'entrées analogiques	6
Courant max par broches E/S	40 mA
Courant max sur sortie	3,3V 50mA
Mémoire Flash	32 KB dont 0.5 KB utilisée par le bootloader
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB
Vitesse de l'horloge	16 MHz

TABLE 4.1 – Les caractéristiques d'Arduino UNO

4.3.4 Constitution de la carte Arduino UNO

Un module ArduinoUno est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328 ou ATmega2560 pour les versions récentes, ATmega168 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est pré-programmé avec un boot loader de façon à ce qu'un programmeur dédié ne soit pas nécessaire [27].

La carte ArduinoUno contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB et possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure ci-dessous :

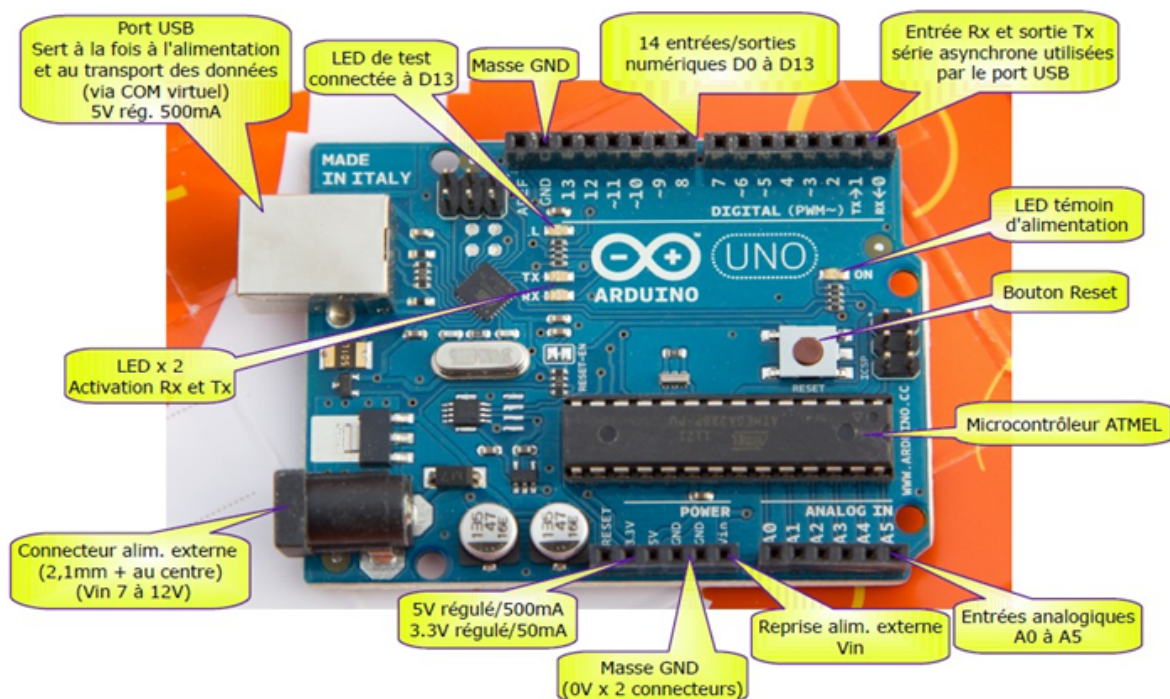


Fig. 4.13 – Les composants de la carte Arduino UNO

Branchage de la carte ArduinoUno

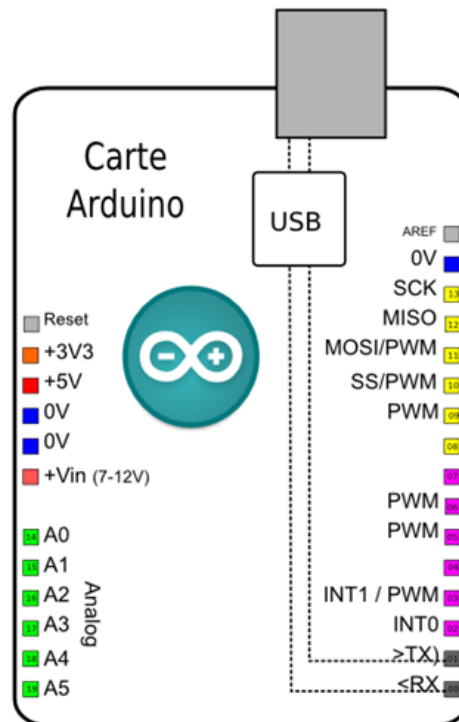


Fig. 4.14 – Branchage de la carte Arduino UNO

- **Alimentation** : la carte ArduinoUno peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positive au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelés GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte UNO est entre 7V et 12V.
- **Les broches d'alimentation sont les suivantes** :
 - VIN : la tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe. Nous pouvons alimenter la carte à l'aide de cette broche ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
 - 5V : la tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info :les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette

broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.

-3V3 : une alimentation de 3.3V fournie par le circuit intégré **FTDI** (circuit intégré faisant l'adaptation du signal entre le port USB de notre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA GND .Broche de masse (ou 0v).

- **Entrées et sorties numériques** : chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions `pinMode`, `digitalWrite` et `digitalRead` du langage Arduino. Ces broches fonctionnent en 5V.

Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction `digitalWrite` (broche, HIGH). De plus, certaines broches ont des fonctions spécialisées :

1. **Communication Série** : broches 0 (RX) et 1 (TX), utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U programmé en convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.
2. **Interruptions Externes** : broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur. Voir l'instruction `attachInterrupt` () pour plus de détails.

-**Impulsion PWM (largeur d'impulsion modulée)** : broches 3, 5, 6, 9, 10, et 11. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite` ().

-**SPI (Interface Série Périphérique)** : broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la bibliothèque pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.

-**I2C** : broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (TwoWire Interface - Interface "2 fils"), disponible en utilisant la bibliothèque `Wire/I2C` (ou `TWI - Two-Wire interface - interface "2 fils"`) .

-**LED** : broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

- **Les ports de communications** : la carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL). Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328. Programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur. Comme un port de communication virtuel pour le logiciel sur

l'ordinateur. La connexion série de l'Arduino est très pratique pour communiquer avec un PC.

4.3.5 Le logiciel Arduino

Le logiciel de programmation de la carte Arduino de code (langage proche de C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimenté à la fois en énergie la carte et transporte aussi l'information, ce programme appelé IDE Arduino (l'Arduino IDE est le plus utilisé, ainsi, il est disponible sur le site officiel pour télécharger gratuitement (www.arduino.cc)).

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

Présentation de «IDE »

L'interface du logiciel Arduino se présente de la façon suivante :

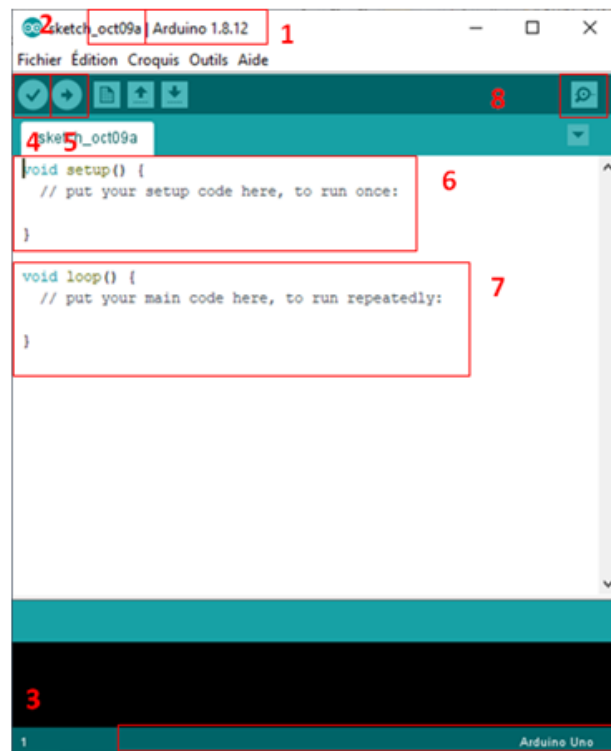




Fig. 4.15 – Interface de logiciel Arduino.

On a numéroté les informations les plus importantes de 1 à 8, on peut distinguer trois parties, 1-3 le logiciel, la carte, le nom. Ensuite le 4-5 sont des boutons permettant de vérifier et téléverser le code. Pour finir de 6-7 c'est le code :

1. La version du logiciel de l'Arduino, il faut faire attention si nous essayons un code sur internet il n'est pas forcément compatible si il n'est pas dans la même version.
2. Le nom du sketch qui correspond à la date du jour ou nous avons ouvert notre sketch, il suffit de l'enregistrer pour qu'il change de nom.

3. Le modèle de l'arduino sur le port qui correspond.
4. Le bouton valider  permet de vérifier le code pour voir s'il y a des erreurs dedans.
5. Le bouton téléverser  permet de transmettre le code sur l'arduino une fois fini. Il est conseillé d'enregistrer et de vérifier le code avant de cliquer sur ce bouton.
6. void setup ()c'est la partie du code qui va s'effectuer lors du branchement de la carte arduino ou lorsqu'on clique sur le bouton rouge reset. Elle permet d'initialiser les variables, le sens des broches et les librairies utilisées. Même si nous mettons rien dedans il le laisser dans le code.
7. void loop ()est une fonction qui va s'effectuer sous forme de boucle (=loop en anglais), qui permet au programme de s'effectuer et de répondre.
Cette fonction est aussi obligatoire dans le code.
8. Le moniteur série affiche des informations de base en temps réel. Il peut par exemple remplacer un écran LCD, parce qu'il est directement sur l'ordinateur.

4.4 Conclusion

Dans ce chapitre nous avons abordé le développement de l'interface de test et de commande en visual studio, et expliqué toutes ses fonctions de réalisation . Nous avons également exposé une présentation de l'Arduino Uno, ses caractéristiques et le fonctionnement du programme établi.

Chapitre 5

Réalisation et implémentation du projet

5.1 Introduction

Dans ce chapitre, nous allons développer les étapes nécessaires pour aboutir à notre objectif et faciliter la réalisation de notre projet **”Conception et construction d’une interface graphique intelligente pour le teste et l’étalonnage des composants électroniques avec Arduino”** ainsi que les Outils de programmation (visuel studio C#, Arduino Uno) et les branchements des différents composants électroniques .

5.2 Les étapes de la réalisation

La réalisation de notre dispositif a connu cinq étapes :

- **Première étape** : c’est l’étape la plus difficile car elle consiste de la recherche des informations bibliographiques et techniques à utiliser pour aboutir l’objectif.
- **Deuxième étape** : création de l’interface graphique (application intelligente) dans visuel studio.
- **Troisième étape** : test de chaque composant électronique et programmation sur Arduino.
- **Quatrième étape** : réalisation du montage complet sur une plaque d’essai avec l’exécution du programme implanté dans visuel studio.
- **Cinquième étape** : interprétation des résultats .

5.3 La réalisation pratique

5.3.1 Structure du système

Le système conçu comporte un bloc principal constitué d’une application réalisée par visuel studio (C#) et une carte à microcontrôleur du type Arduino UNO . Cette carte est connectée à un ensemble des périphériques.

Notre système est composé de :

- **Software** : cette partie compte sur une application intelligente développée technologiquement par Visuel Studio (C#).
- **Hardware** : la partie est basée sur la carte Arduino UNO, qui prend en charge la réception et l’envoi des différentes informations.

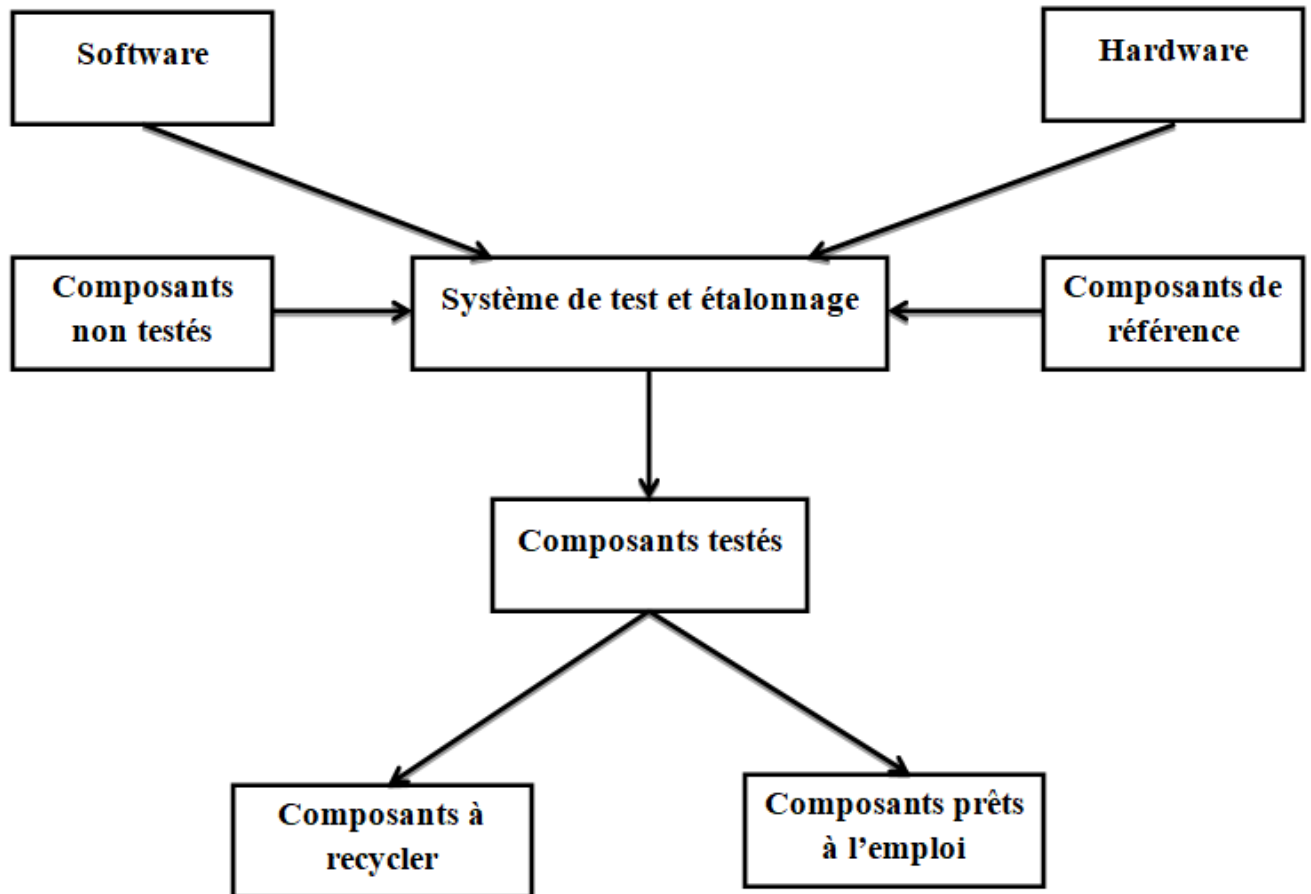


Fig. 5.1 – Schéma bloc du système.

— Les composants électroniques :

- Capteur de température et humidité.
- Capture de lumière (LDR).
- Bouton poussoir.
- Le capteur à ultra-son.
- LED.
- LCD.
- Servomoteur.
- Resistance.

5.3.2 Résultats de la réalisation expérimentale

La **fig. 5.2** représente l'application graphique intelligente de notre système avant l'exécution des programmes.

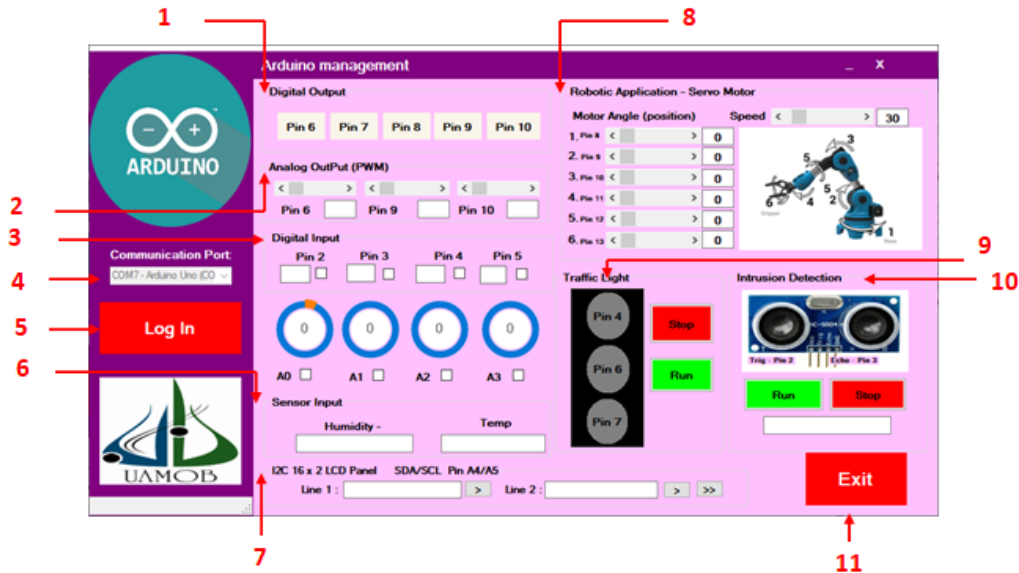


Fig. 5.2 – L'application graphique intelligente

On a numéroté les informations importantes de 1 à 11 :

1. La barre de communication port affiche le COM série de chaque ordinateur, dans notre cas on travaille avec COM7.
2. L'appui sur LOG IN, nous permet de connecter l'application.
3. Digital Output : présente les commandes des composants numériques.
4. Analog Output(PWM) : présente les commandes des composants analogiques.
5. Digital Input : **La 1ère ligne** c'est la réception des données des composants numériques. **La 2ème ligne** c'est la réception des données des composants analogiques.
6. Sensor Input : affiche l'humidité et la température.
7. LCD : **Line 1**, affiche le texte voulu dans la 1ère ligne. **Line 2**, affiche le texte voulu dans la 2ème ligne.
8. Robotic Application- Servo Motor : le curseur ajuste l'angle de rotation de servo motor.
9. Traffic Light : le panneau désigne le cas du feu de signalisation.
10. Intrusion Detection :repréente le capteur de distance .
11. EXIT : c'est la sortie de l'application.

Les tests et l'étalonnage des composants électroniques

Le démarrage de l'application se fait en appuyant sur le bouton LOG IN.

— LED

Après cela, nous cliquons sur le bouton PIN6 des sorties numériques qui signifie le PIN6 sur l'arduino qui est déjà lié à une LED, nous voyons la LED allumée.

Résultat : Le bon fonctionnement de la LED.

L'appui sur le Pin6 = allumer la LED

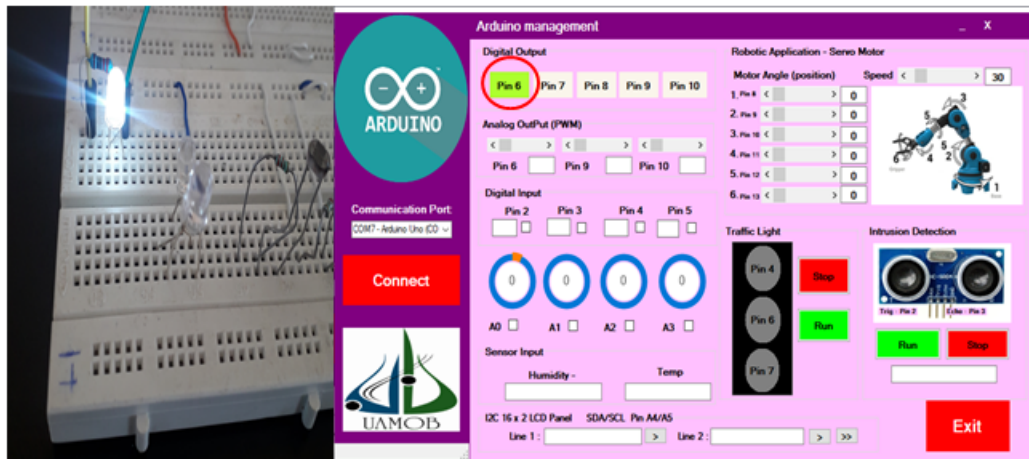


Fig. 5.3 – Résultats de réalisation de la LED

— Bouton poussoir

Par la suite, on va cocher la sortie digitale PIN2 et on cliquant sur le bouton poussoir, on revient sur l'application, on voit la petite surface devient verte, celà signifie la lumage de la LED.

Résultat : Le bon fonctionnement du bouton poussoir.

L'appui sur le bouton poussoir = allumer la LED

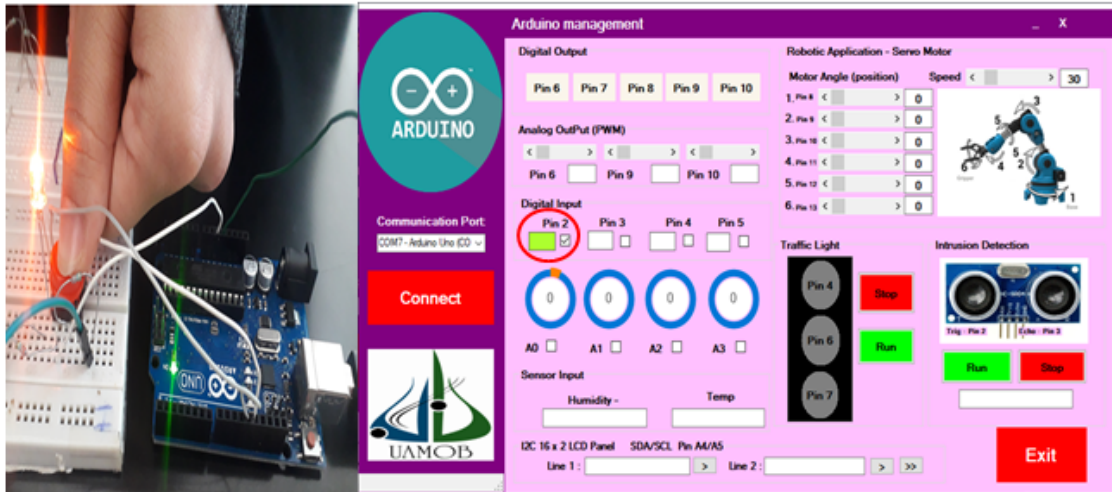


Fig. 5.4 – Résultats de réalisation de Bouton poussoir

— Capteur de température et d'humidité

Pour l'affichage de la température et l'humidité, on doit appuyer sur la barre a l'espace blanc.

Résultat : Le bon fonctionnement du capteur.

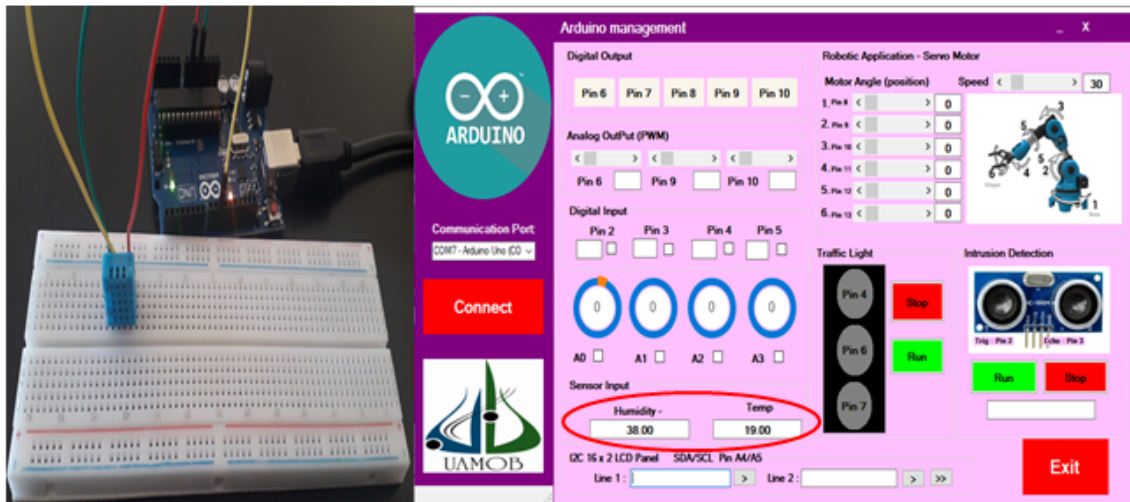
Affichage de la température et l'humidité dans les conditions normales

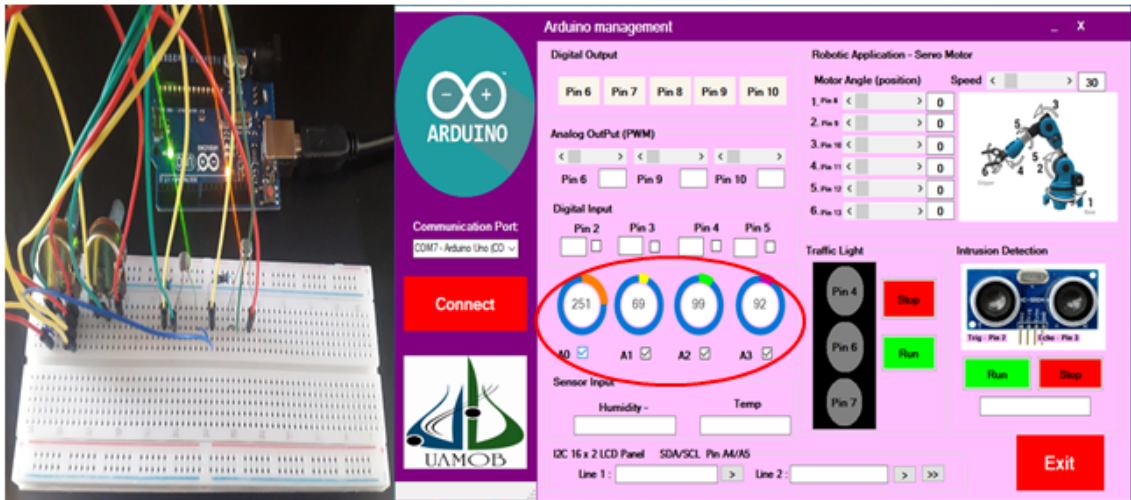
Fig. 5.5 – Résultats de réalisation de Capteur de température et d'humidité

— Potentiomètre et LDR

Nous ajustons les potentiomètres qui sont reliés au PIN A1 et A2, et aussi par la projection de la lumière envers les LDRs qui sont aussi reliés au A0 et A3 respectivement, on voit que les valeurs des 2 potentiomètres et la luminosité des LDRs s’augmentent.

Résultat : Le bon fonctionnement des potentiomètres et LDRs.

Absence de la lumière + valeur minimale des potentiomètres



Présence de la lumière + valeur maximale des potentiomètres

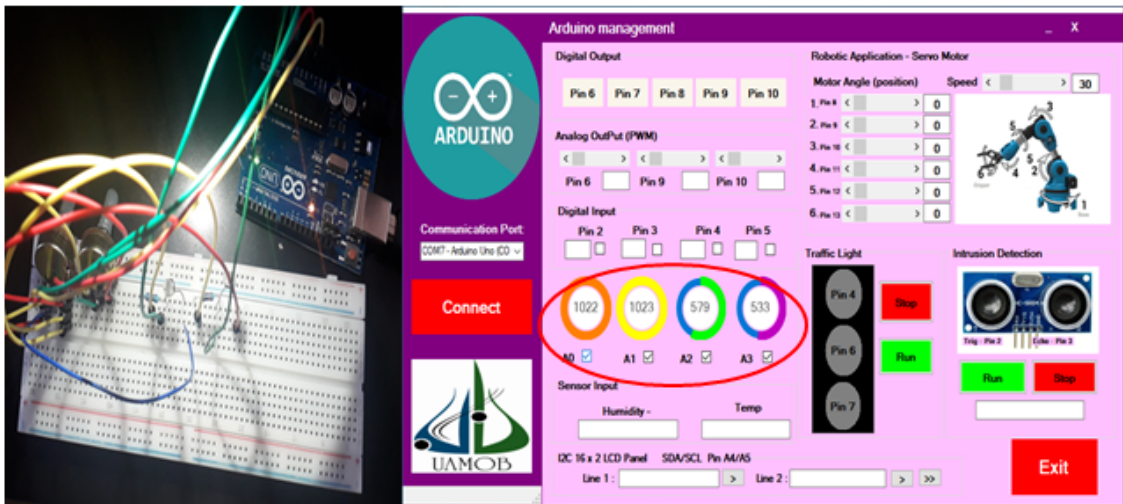


Fig. 5.6 – Résultats de réalisation des Potentiomètres et LDRs

— Servo moteur

L'ajustement du curseur change l'angle de rotation de servo motor.

Résultat : Le bon fonctionnement du servo motor.

L'angle de rotation du servomoteur est 40°

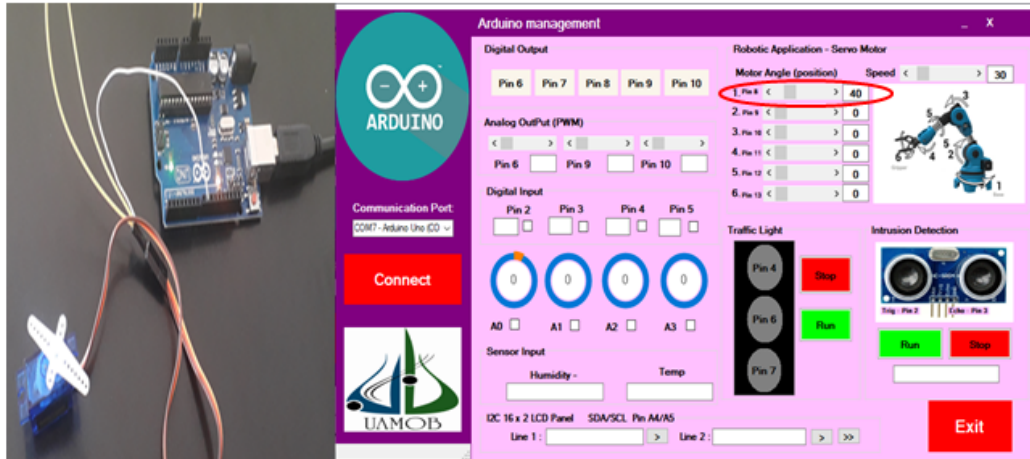


Fig. 5.7 – Résultats de réalisation de Servo moteur

— Feu rouge

L'exécution du bloc de feu de signalisation ce faite par l'appui sur le bouton RUN, nous voyons l'allumage des feux (rouge, jaune, vert), 2 secondes pour chacune.

Résultat : Le bon fonctionnement des LEDs.

Signalisation rouge du panneau tri-colo

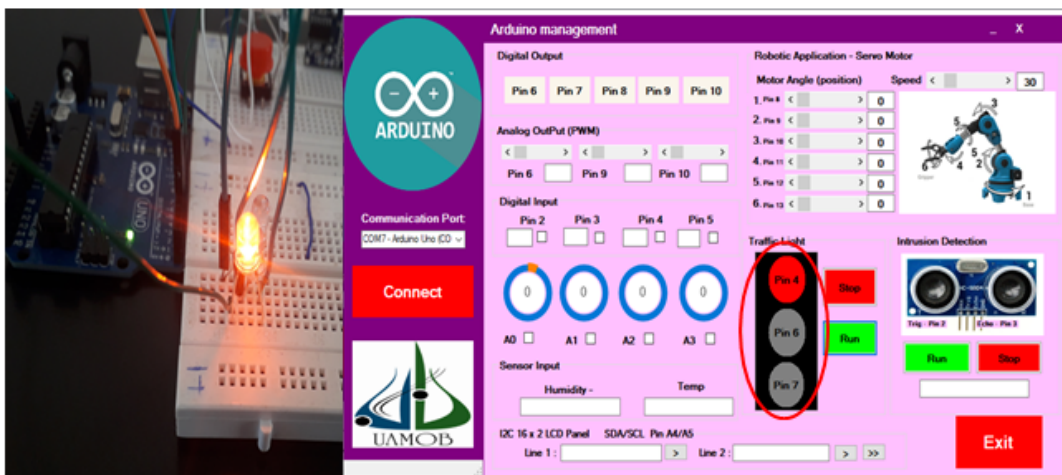


Fig. 5.8 – Résultats de réalisation de Feu rouge

— LCD

On passe a l'afficheur LCD 16*2, nous écrivons un mot dans la 1ère barre et on valide,

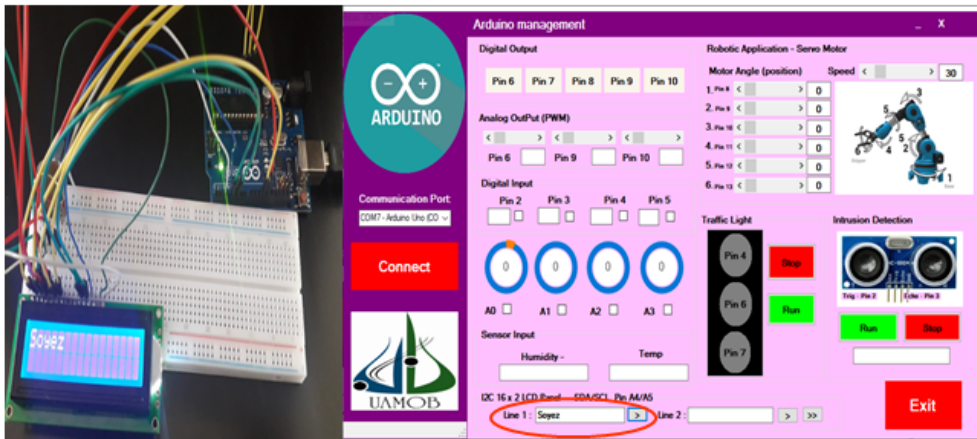
nous voyons l’affichage de la 1ère ligne sur LCD.

Même chose pour la 2ème ligne.

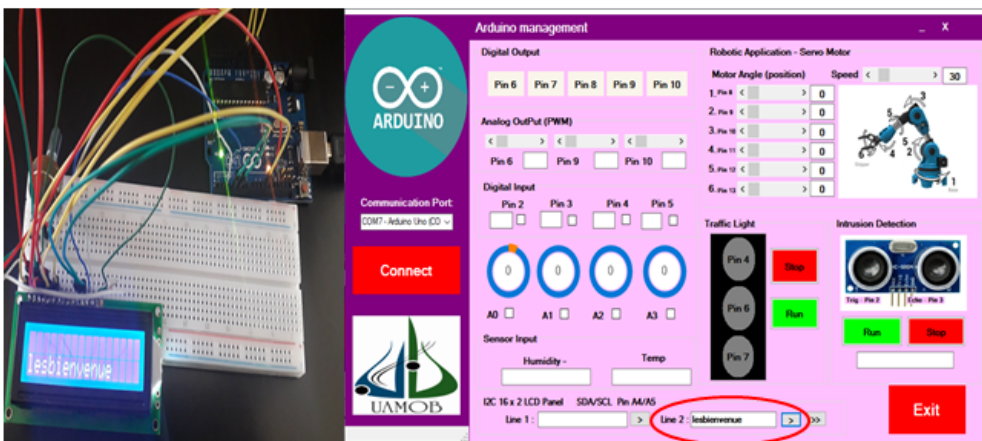
Pour l’affichage des 2 lignes au même temps suffit d’appuyer sur le bouton qui a les 2 flèches.

Résultat : Le bon fonctionnement de LCD.

Affichage de la 1ère ligne



Affichage de la 2ème ligne



Affichage des deux lignes au même temps

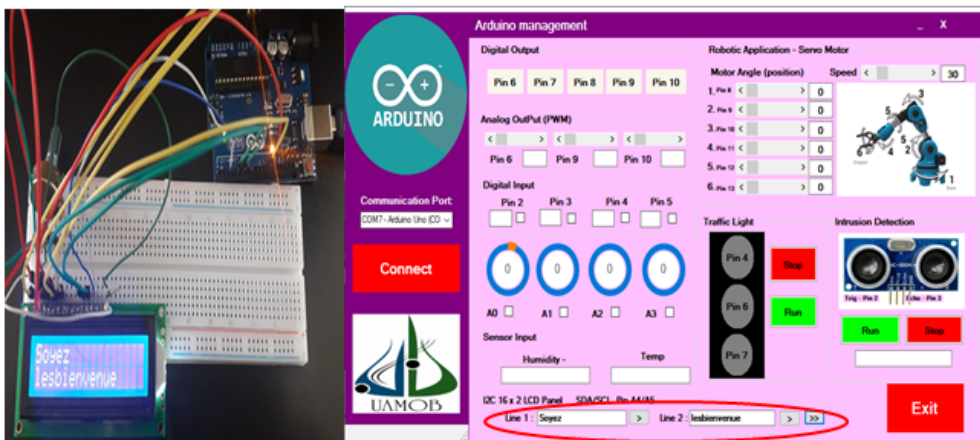


Fig. 5.9 – Résultats de réalisation de LCD

— Capteur de distance

Pour exécuter le capteur ultra-son, on appui sur le bouton RUN, la distance s'affichera 0, après on remarque l'augmentation de la valeur, cela signifie une mise d'obstacle puis on l'enlève.

Résultat : Le bon fonctionnement du capteur.

La distance entre le capteur ultra-son et l'obstacle = 0 cm



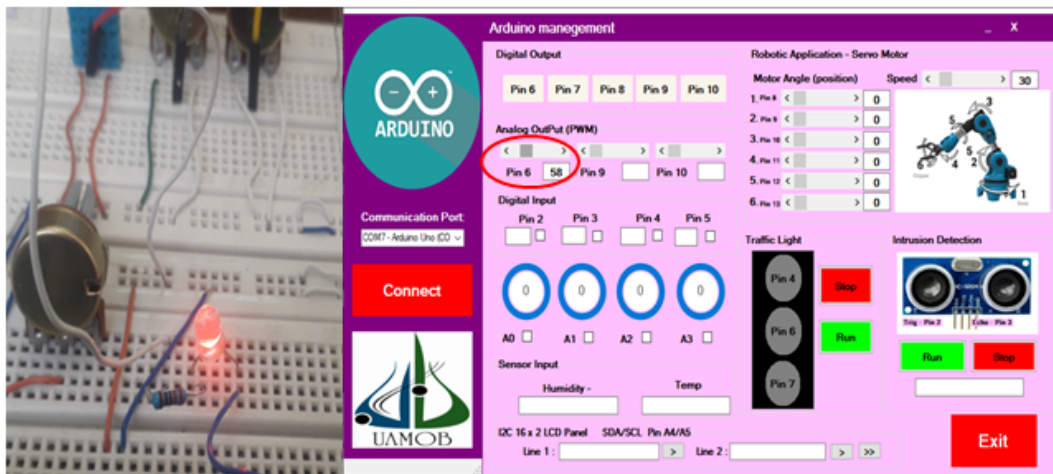
La distance entre le capteur ultra-son et l'obstacle = 17 cm



Fig. 5.10 – Résultats de réalisation de Capteur de distance

— La luminosité d'une LED

La LED est à la luminosité minimale



La LED est à la luminosité maximale

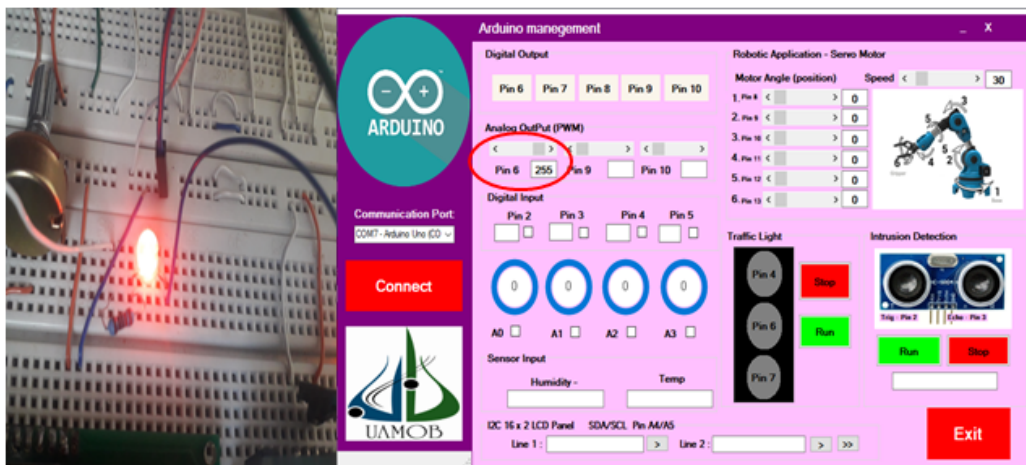


Fig. 5.11 – Résultats de réalisation de la luminosité d'une led

5.4 Conclusion

Ce chapitre est réservé à la conception et la réalisation du système, puis nous présentons les résultats obtenus.

Les résultats obtenus par réalisation montrent les avantages qu'apporte la technique du test et l'étalonnage ainsi le succès et le bon fonctionnement de notre application.

Conclusion générale

Le travail effectué de ce mémoire rentre dans le cadre de la conception et la construction d'une interface graphique intelligente pour le test et l'étalonnage des composants électroniques avec Arduino.

Pour ce faire, nous avons apporté une attention particulière à l'élaboration des différents types des composants électroniques associés à Visual Studio (C#) et Arduino, en apprenant de façon complète la réalisation de l'application technologique.

Malgré le développement des interfaces graphiques, les applications intelligentes sont très utilisées de nos jours, notamment dans le domaine électronique, technique et encore dans les domaines de recherche.

Le choix de ce genre des applications parmi les autres types est basé sur l'avantage de test et l'étalonnage des composants électroniques pour éviter tous les problèmes existants.

Les composants électroniques sont branchés sur la plaque d'essai, liés principalement à l'Arduino et Visual Studio (C#) qui a programmé cette application. Dans ce mémoire, nous avons rappelé des généralités sur les systèmes embarqués. Puis nous avons présenté la communication série et les processus de l'étalonnage et tests ainsi que la partie soft et hard du système applicable.

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous sommes confrontés à plusieurs problèmes surtout dans la partie pratique.

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

Dans ce travail, nous avons contribué à l'interface graphique (l'application intelligente) tout en travaillant avec un microcontrôleur Arduino UNO. De bons résultats ont été obtenus ce qui justifie l'efficacité de notre réalisation. Comme perspective, nous proposons de travailler avec des applications intelligentes de test plus développées dans tous les domaines de recherche pour assurer la bonne qualité des composants électroniques.

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

Bibliographie

- [1] Encyclopedia Britannica, "*electronics | Devices, Facts,History*", Retrieved (19 September 2018),
<https://www.britannica.com/technology/electronics>.
- [2] Alan Chodos, Jennifer Ouellette, "*October 1897 : The Discovery of the Electron*", (October 2000)(Volume 9, Number 9),
<https://www.aps.org/publications/apsnews/200010/history.cfm>.
- [3] John Hayes, " "*Introduction to Digital Logic Design*", (1993), ISBN : 978-0201154610.
- [4] Saraju Mohanty, "*Nanoelectronic Mixed-Signal System Design*", McGraw-Hill, (2015), ISBN 978-0071825719.
- [5] Jack Ganssle, Michael Barr "*Embedded systems dictionary*", CMP Books, San Francisco 2003, ISBN : 1-57820-120-9.
- [6] Ken Shirriff, "*The Surprising Story of the First Microprocessors*", [IEEE Spectrum](https://www.elsevier.com/locate/S0018-9219(16)30080-9). Institute of Electrical and Electronics Engineers, (30 Aug 2016).
- [7] Computer History Museum, "*1971 : MICROPROCESSOR INTEGRATES CPU FUNCTION ONTO A SINGLE CHIP*", <https://www.computerhistory.org>.
- [8] Peter Marwedel, "*Embedded System Design*", Springer, Netherlands (2006), ISBN-13 978-0-387-29237-3 (PB).
- [9] Bai Ying, 1956, "*The Windows Serial Port Programming Handbook*", (2005) by CRC Press LLC.
- [10] Transmission-de-donnees-les-modes-de-transmission, (juin 2017),
<https://www.commentcamarche.net/contents/>.
- [11] Guy Pujolle "*Protocoles de transmission de données*", (février 1997).
- [12] Jan Axelson, "*Serial Port Complete COM Ports, USB Virtual COM Ports, and Ports for Embedded Systems Second Edition*", Copyright 1998, 1999, 2000, and 2007 by Janet L. Axelson.
- [13] Marilyn Wolf, "*Embedded System Interfacing Design for the Internet-of-Things (IoT) and Cyber-Physical Systems (CPS)*", (2019) .
- [14] Hewlett Packard Company, Intel Corporation, Microsoft Corporation, Renesas Corporation, ST-Ericsson, Texas Instruments, Universal Serial Bus 3.1 Specification, Revision 1.0, (July 26 2013).

- [15] NXP Semiconductors, UM10204, I2C-Bus Specification and User Manual, Rev. 03, 19 June, (2007).
- [16] [Pcisig.com](https://www.pcisig.com/specifications/pciexpress/base3/), PCI-SIG - PCI Express Base 3.0 Specification. Retrieved (25 May 2015), from <https://www.pcisig.com/specifications/pciexpress/base3/>.
- [17] International Organization for Standardization, International Vocabulary of Basic and General Terms in Metrology, 2nd ed, (1993).
- [18] International organization for Standardization, "*Measurement management systems : requirements for measurement processes and measuring equipment*",(ISO 10012 :2003).
- [19] David B. Newell and Eite Tiesinga, Editors "*The international system of unitis (SI)*", (2019).
- [20] Calibration TechnologyBasics, reference instruments for pressure andtemperature, professional calibration Christian Elbert (2012) All rights reserved with Süddeutscher Verlag onpact GmbH, 81677 Munich www.sv-onpact.de .
- [21] American National Standards Institute/National Conference of Standards Laboratories, "*Calibration laboratories and measuring and test equipment : general requirements*",(ANSI/NCSL Z540-1-1994).
- [22] Guidelines for the determination of calibration intervals of measuring instruments. International Laboratory Accreditation Cooperation.ILAC G24 :2007, (2007).
- [23] International vocabulary of metrology – basic and general concepts and associated terms (VIM), 3rd edn. Joint Committee for Guides in Metrology. JCGM 200 :2012, (2012).
- [24] Quality management systems – requirements. International Organization for Standardization, Geneva ISO 9001 :2008 (2008).
- [25] Dr.Liew "*Visual Basic 2019 Made Easy*", (2019).
- [26] Patrik Renard Acquisition de données du capteur, Georges, Asch, E, chambard, , Günther 528p, (2003), Dunod.
- [27] Jean- Noël Mantagné, "*Livret Arduino en français* " , centre de ressources art sensitif, (11/2006).

Annexe

Matériels et leurs caractéristiques

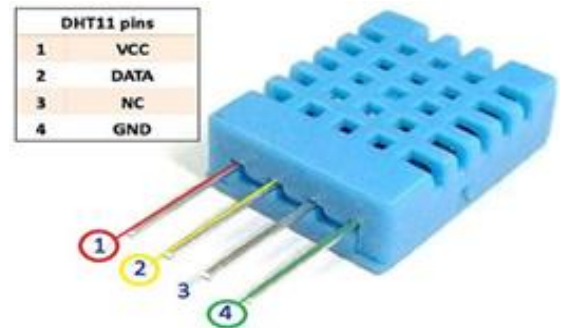
— La carte Arduino

Dans notre projet nous avons utilisé une carte ARDUINO UNO basée sur un microcontrôleur ATmega328 cadencé à 16 MHz.



— Capteur d'humidité et température

Le capteur DHT11 est capable de mesurer des températures de 0 à +50°C avec une précision de +/- 2°C et des taux d'humidité relative de 20 à 80%, avec une précision de +/- 5%. Une mesure peut être réalisée toutes les secondes.



— Capteur de distance

Ce capteur est bien connu des amateurs de robotique et d'Arduino : le **HC-SR04**, ce dernier est un capteur à ultrason lowcost et fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3mm (en théorie, dans la pratique ce n'est pas tout à fait exact).



— Bouton poussoir

Est un interrupteur simple qui permet de contrôler les capacités d'un processus.



— LCD (Liquid Cristal Display)

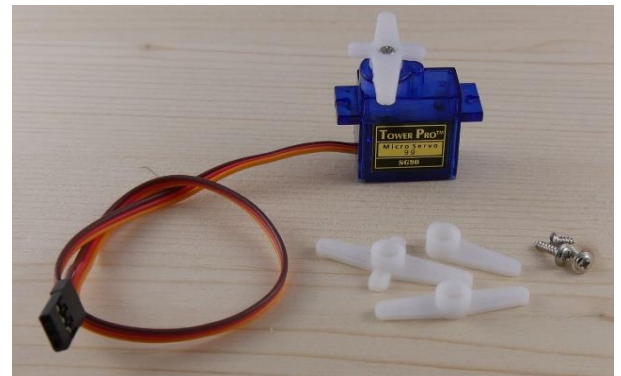
Les afficheurs LCD 16*2 (16 caractères et 2 lignes) sont des modules compacts intelligents et nécessitent peu de composants externes pour un bon fonctionnement. Ils consomment relativement peu (de 1 à 5 mA), dotés d'un rétro-éclairage de l'affichage. Cette fonction fait appel à des LED montées derrière l'écran du module, cependant, cet éclairage est gourmand en intensité (de 80 à 250 mA). Aujourd'hui ces écrans sont utilisés dans presque tous les affichages électroniques.



N	Nom	Rôle
1	VSS	Masse
2	Vdd	+5V
3	Vo	Réglage du contraste
4	RS	Sélection du registre (commande ou donnée)
5	R/W	Lecture ou écriture
6	E	Entrée de validation
7 à 14	D0 à D7	Bits de données
15	A	Anode du rétro-éclairage (+5V)
16	K	Cathode du rétro-éclairage (masse)

— Servomoteur

On a utilisé un servomoteur, communément appelé "servomoteur 9 grammes", par souci de consommation électrique et laplupart des servomoteurs fonctionnent en 5 volts.



—LDR (Light Dependent Resistor)

Une photo-résistance, on la désigne aussi par LDR, est un composant dont la valeur en ohms dépend de la lumière à laquelle il est exposé.



— Potentiomètre

Potentiomètre de type linéaire , dont la valeur de résistance varie proportionnellement à la distance entre ses bornes et le curseur .

Valeur	100k Ω
Durée de vie	100000 cycles
Dimension du corps	16mm
Course	300°
Force d'arrêt	6,5 kg/cm
Force de rotation	0.036 to 0,09 kg/cm



— Résistance

Résistance à film métallique est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance (mesurée en ohms) à la circulation du courant électrique.

Valeur	220 Ω
Tolérance	1 / 6W
plomb axial	1%,
bandes pour projets et expériences électroniques de bricolage	5 bandes



— LED (Light Emitting Diode)

Diode électroluminescente, on la désigne aussi par LED est un dispositif opto-électronique capable d'émettre de la lumière lorsqu'il est parcouru par un courant électrique.

