

Département génie électrique

Filière : Génie Mécatronique

Mémoire de Projet de fin d'année

Conception et réalisation d'un système de vote avec affichage des résultats

Présenté par

Zhani Badr Eddin

Mejri Louay

Encadré par

Harabi Ferid

Remerciements

Au terme de ce travail, nous remercions **Dieu** le tout puissant de nous avoir donné la force et la patience pour réaliser ce travail.

Nous tenons à exprimer notre profonde gratitude et nos vifs remerciements à notre encadrant **Mr. Harabi Ferid** pour tous les efforts qu'il a fait ainsi que pour la confiance qu'il nous a accordé durant tout le déroulement de ce projet.

Nous tenons à remercier tous les enseignants de la filière « **Génie Mécatronique** » de l'**École National d'ingénieurs de Carthage** pour leurs contributions à notre formation, et également nos camarades, amis pour leurs aides précieuses, leurs encouragements permanant et leur soutien moral.

Finalement, Nous ne remercierons jamais assez nos parents qui nous ont le plus soutenu et encouragé durant toute notre vie, c'est grâce à eux que nous en sommes là aujourd'hui.

Table des matières

Introduction générale.....	1
Chapitre I : Etude bibliographique	2
I.1. Introduction :.....	2
I.2. Etude de l'existant :	2
I.2.1. Problématique :	2
I.3. Analyse des besoins :.....	3
I.3.1. Le diagramme Bête à cornes	3
I.3.2. Diagramme SADT :.....	3
I.3.3. Diagramme Pieuvre :	4
I.4. Proposition d'une solution :	5
I.4.1. Description et analyse de solution :.....	5
I.4.2. Présentation de logigramme :.....	6
I.5. Conclusion :	7
CHAPITRE II : Etude et conception de la solution proposée.....	8
II.1. Introduction :	8
II.2 Définition du module Arduino :	8
II.2.1. Pourquoi Arduino UNO :.....	9
II.2.2. La constitution de la carte Arduino UNO :	10
II.3. Les Accessoires de la carte Arduino :	14
II.3.1. Les afficheurs LCD :.....	14
II.3.2. Le bouton poussoir :	16
II.3.3. Alimentation des cartes Arduino :	17
II.4. Conclusion :.....	17
CHAPITRE III : Réalisation pratique	18
III.1. Introduction.....	18

III.2. Description de l'environnement expérimental	18
III.3. Procédure d'affichage des résultats de vote	19
III.3.1. Étape 1 : simulation sur logicielle Proteus 7 :.....	19
III.3.2. Étape 2 : Programmation Arduino :	20
III.4. Résultats pratique obtenir.....	24
III.5. Etude des prix :	25
III.6. Conclusion :	26
Conclusion générale	27
Références Bibliographiques	28

Liste des figures

Figure I.1 : schéma de principe de l'état actuel	2
Figure I.2 : Bête à cornes	3
Figure I.3 : Diagramme SADT.....	3
Figure I.4 : Diagramme de Pieuvre	4
Figure I.5 : Logigramme du système.....	7
Figure II.1 : Carte Arduino UNO.....	9
Figure II.2 : Le composant CMS	10
Figure II.3 : Les ports de communications de l'Arduino UNO [1]	13
Figure II.4 : Structure générale du programme	14
Figure II.5 : Paramétrage de la carte [2]	14
Figure II.6 : Afficheur LCD	15
Figure II.7 : Câblage de LCD avec la LCM [3]	16
Figure II.8 : Bouton poussoir monostable.....	16
Figure II.9 : Ports d'alimentation de la carte	17
Figure III.1 : Environnement et matériel utilisés pour la réalisation pratique de notre projet de fin d'année	19
Figure III.2 : Importation des bibliothèques de Proteus 7.....	19
Figure III.3 : Circuit complète sur le Proteus 7	20
Figure III.4 : IDE-Arduino.....	20
Figure III.5 : Importation des bibliothèques Arduino.....	21
Figure III.6 : Implémentation de LCD sur la carte	21
Figure III.7 : Configuration des E/S	22
Figure III.8 : Définition des messages	22
Figure III.9 : Extrait du programme.....	23
Figure III.10 : Compilation et chargement du code.....	23
Figure III.11 : Message d'OUTPUT.....	23
Figure III.12 : Circuit des boutons et leurs rôles	24
Figure III.13 : Exécution de programme.....	24
Figure III.14 : Résultat obtenu après l'appui sur S3.....	24
Figure III.15 : Lancement d'un nouvel essai de vote	25

Liste des tableaux

Tableau I : Les fonctions des services du diagramme de pieuvre et leurs services	5
Tableau II : symboles de logigramme.....	6
Tableau III : devis des composantes utilisées [4]	25

Introduction générale

L'élection peut être une question clé lorsqu'il s'agit de décider qui sera le ou les dirigeants ou représentants suivants visiteront par des moyens démocratiques. Les processus de vote qui prévalent lors de ces élections sont lents et pénibles et le résultat est généralement inexact. Ainsi, les gens perdent confiance dans le processus électoral et, par conséquent, dans leurs dirigeants. Ce travail a permis de mettre au point un système électoral électronique qui facilite la méthode de sélection de ces dirigeants d'une manière extrêmement rapide, libre et équitable. L'appareil a été développé en interfaçant un une carte Arduino Uno, des boutons poussoirs et un afficheur LCD. Ce projet est codé à l'aide de l'IDE Arduino.

Le système de vote intelligent automatisé et moderne est une option. Il permet la collecte, le traitement et l'enregistrement électroniques des votes à l'aide d'appareils Arduino. Les résultats sont automatiquement enregistrés et comptabilisés de manière équitable, et les électeurs peuvent utiliser des interfaces conviviales pour voter.

Ce rapport est divisé en trois sections :

Le premier chapitre sera consacré à une analyse bibliographique approfondie expliquant l'analyse fonctionnelle de système de vote intelligent avec la proposition de solution possible et leur fonctionnement. Le deuxième chapitre sera consacré à la solution retenue, qui inclura à la fois les composants logiciels et matériels de notre projet.

Enfin, nous détaillerons dans le troisième et dernier chapitre la mise en œuvre électronique de notre projet, ainsi que la programmation en langage C++ liée à notre carte Arduino et l'affichage des résultats des votes.

Chapitre I : Etude bibliographique

I.1. Introduction :

Les systèmes de vote intelligents sont des programmes informatiques qui rendent le vote plus facile et plus sûr. Ils peuvent être utilisés dans divers contextes, notamment les élections présidentielles et législatives, les assemblées annuelles d'entreprise et les sondages en ligne pour des enquêtes ou des concours.

I.2. Etude de l'existant :

Le vote à main levée est une méthode de vote utilisée dans les réunions et les assemblées, où les participants votent en levant simplement la main pour exprimer leur choix. Bien que cette méthode soit simple et rapide, elle n'est pas toujours considérée comme la plus fiable, car elle ne permet pas de garantir l'anonymat et peut être influencée par des facteurs externes tels que la pression sociale.



Figure I.1 : schéma de principe de l'état actuel

I.2.1. Problématique :

Cependant, dans certaines situations, le vote à main levée peut être utilisé avec succès, notamment dans les petites assemblées ou les groupes où la confiance et la transparence sont élevées. Pour maximiser la fiabilité du vote à main levée, il est important de garantir que chaque votant a une chance égale de voter et que le comptage des votes est effectué de manière transparente et précise. Dans l'ensemble, le vote à main levée est une procédure de vote simple mais limitée qui peut être utilisée efficacement dans certaines circonstances. Cependant, la

technologie peut être utilisée pour améliorer la fiabilité et la précision du vote à main levée. Dans ce contexte, nous proposons un système de vote intelligent.

I.3. Analyse des besoins :

Il s'agit d'exprimer avec exactitude le but et les limites de l'étude. Pour cela on peut utiliser la représentation « bête à corne » Et en posant les trois questions suivantes :

I.3.1. Le diagramme Bête à cornes

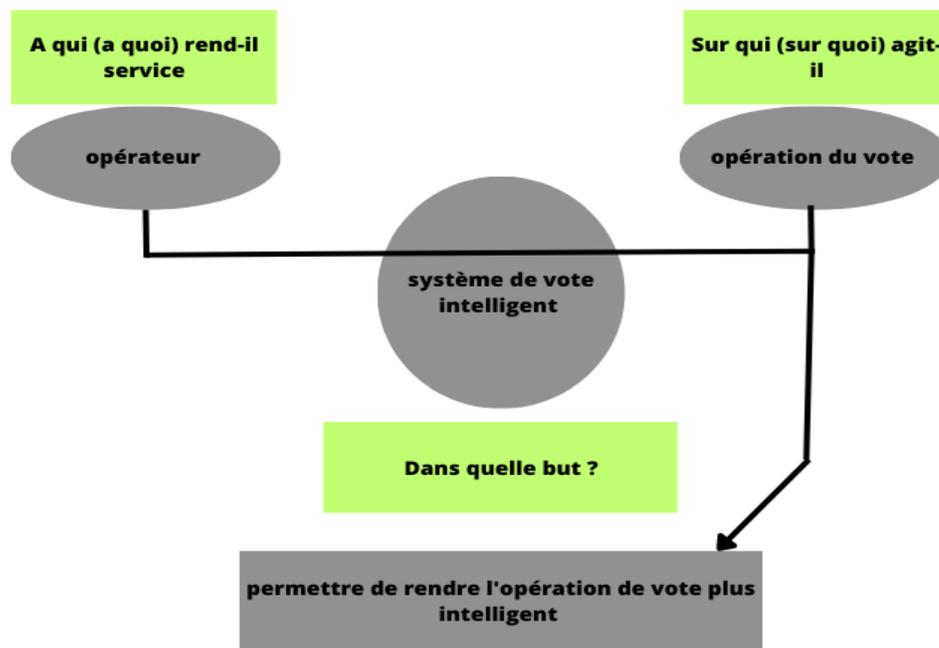


Figure I.2 : Bête à cornes

I.3.2. Diagramme SADT :

Le diagramme SADT est un outil graphique associé à une méthode d'analyse descendante modulaire et hiérarchisée.

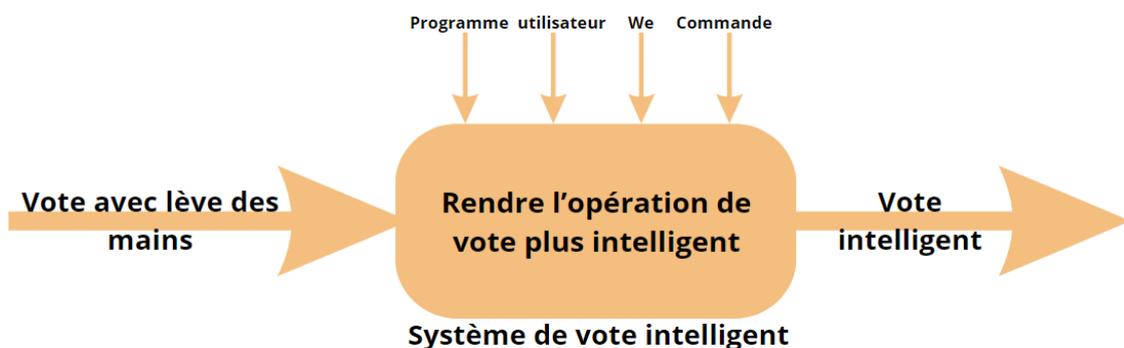


Figure I.3 : Diagramme SADT

I.3.3. Diagramme Pieuvre :

Le diagramme de « Pieuvre » met en évidence les relations entre les divers éléments du milieu environnant et le produit. Ces différentes relations sont appelées les fonctions de service qui conduisent à la satisfaction du besoin.

Les fonctions principales (FP) expriment obligatoirement des relations d'interaction (relations, par l'intermédiaire du produit, entre au moins deux composants au milieu envirant).

Les fonctions de contraintes (FC) expriment obligatoirement des relations d'adaptations, de réaction ou de résistance (relation entre le produit et un élément du milieu environnant).

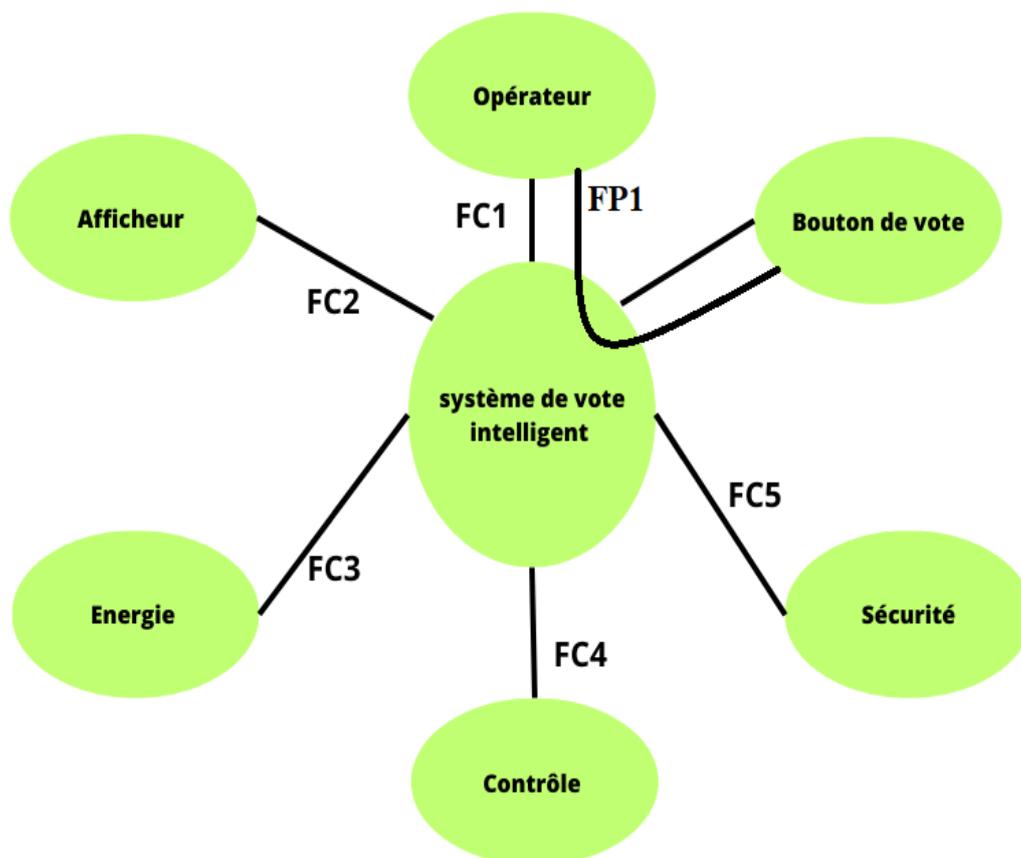


Figure I.4 : Diagramme de Pieuvre

Le tableau représente les fonctions des services du diagramme de pieuvre et leurs services.

Tableau I : Les fonctions des services du diagramme de pieuvre et leurs services

Les fonctions des services (FS)	Expression des fonctions des services
FP1	Permettre à l'utilisateur de prise sa décision.
FC 1	Faciliter à l'utilisateur l'opération de vote.
FC 2	Permettre d'afficher les résultats.
FC 3	S'adapter à l'énergie disponible.
FC 4	Assurer le contrôle de fonctionnement de système.
FC 5	Être sécurisé

I.4. Proposition d'une solution :

Notre projet « système de vote intelligent avec affichage » se compose d'une carte Arduino Uno avec des boutons poussoirs permettre aux utilisateurs de prise ces décisions et un afficheur LCD pour afficher les résultats de vote.

L'étude de programmation Arduino doit tenir compte le nombre d'appui sur les boutons de vote qui sont ciblées pour choisir une de deux choix qui sont oui ou non.

Notre solution technique est basée sur :

- La conception de l'existant.
- Programmation .
- Mise en place du système de vote intelligent.

I.4.1. Description et analyse de solution :

Pour expliquer le principe de fonctionnement de notre système, nous supposons que chaque électeur a le choix de choisir une de deux choix : soit oui soit non donc l'appui sur les deux boutons oui/non simultanément sera interdit.

Chaque électeur a le droit de voter uniquement une seule fois (pour bien comprendre ce point on mettre l'exemple suivant : lors l'opération du vote si l'électeur a choisi oui, il interdit de rechoisir ni oui ni non =action bloquée).

L'affichage du résultat sera à la fin de procédure pour garantir le secret du vote. Ces derniers règlements seront tenus en compte lors de la phase de programmation.

Remarque : la perturbation de l'un de ces derniers points n'arrête pas le processus de vote.

I.4.2. Présentation de logigramme :

Nous devons réaliser un logigramme expliquant le déroulement des différentes séquences.

Le logigramme permet de visualiser de façon séquentielle et logique les actions à mener et les décisions à prendre pour atteindre un objectif. Ils sont créés généralement en ajoutant et liant des symboles, tels que des losanges, rectangle, un cercle, etc., qui sont habituellement utilisés pour représenter la manière dont un processus fonctionne, montrant au minimum la séquence des étapes.

Le tableau suivant Présente les différents symboles de logigramme et leurs significations.

Tableau II : symboles de logigramme

	Première étape et dernière étape
	Autres étapes ou actions
	Condition de distance et choix de décision
	Mise à disposition des données de vote
	Lien entre deux activités

Logigramme de système :

La figure ci-dessous montre comment fonctionne les différentes taches dont la description est expliquée par le schéma suivant :

- S1 : bouton poussoir pour choisir oui
- S2 : bouton poussoir pour choisir non
- Résultat : bouton poussoir pour afficher les résultats du vote
- Reset : bouton poussoir pour réinitialiser le système

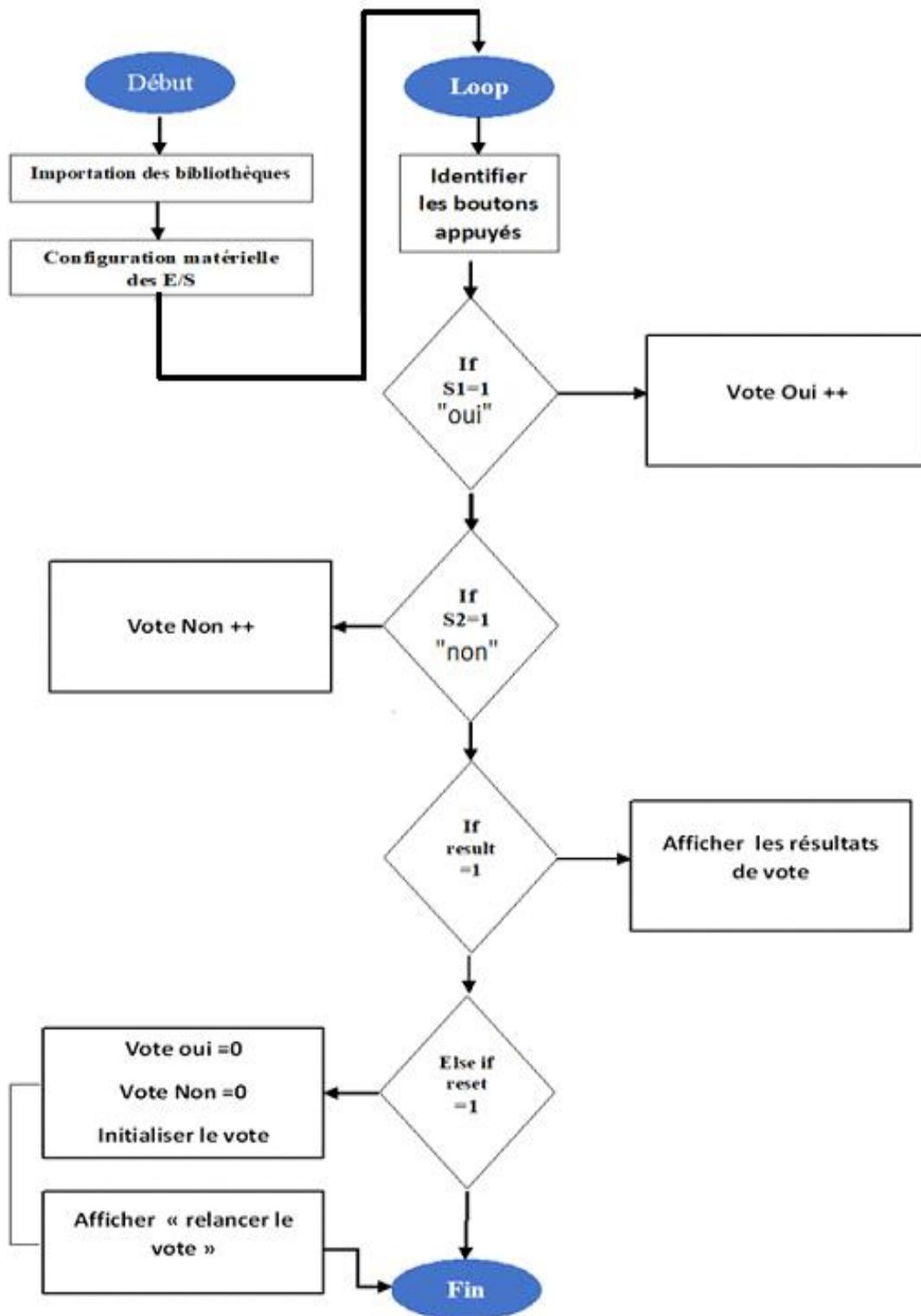


Figure I.5 : Logigramme du système

I.5. Conclusion :

Au niveau de ce chapitre nous avons étudié l'analyse fonctionnelle de système de vote intelligent avec la proposition de solution possible et leur fonctionnement.

CHAPITER II : Etude et conception de la solution proposée

II.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 de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie : ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

II.2 Définition du module Arduino :

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses.

Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme).

Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino.

II.2.1. Pourquoi Arduino UNO :

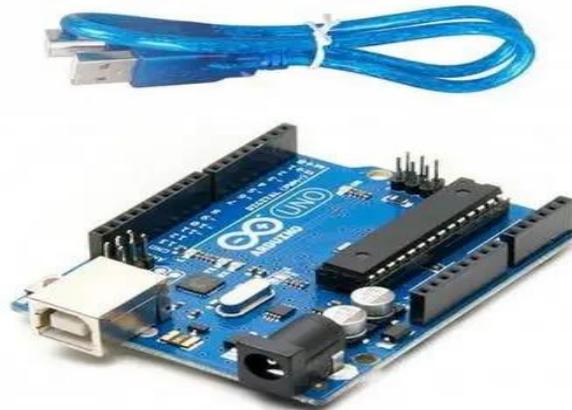


Figure II.1 : Carte Arduino UNO

Il existe des nombreuses cartes électroniques avec des plates-formes de programmation basées sur des microcontrôleurs disponibles. Tous ces outils prennent les détails complexes de la programmation et les regroupent dans une conception intuitive. Le système Arduino offre aux parties intéressées plusieurs avantages, qui sont énumérés comme suit :

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino préassemblées coûtent moins de 100 Dinars)
- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple** : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

• **Matériel Open source et extensible** : les cartes Arduino sont basées sur les Microcontrôleurs ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût

II.2.2. La constitution de la carte Arduino UNO :

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, 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 préprogrammé avec un bootloader de façon qu'un programmeur dédié ne soit pas nécessaire.

II.2.2.1. Partie matérielle :

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.

Le Microcontrôleur ATmega328 :

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors ; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.

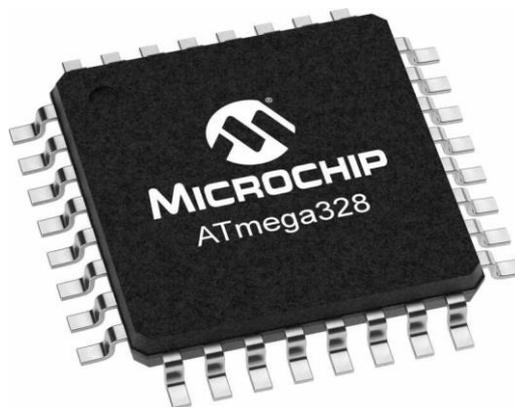


Figure II.2 : Le composant CMS

Le microcontrôleur ATmega328 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement sur :

- La mémoire Flash : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 32Ko (dont bootloader de 0.5 ko).
- RAM : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 2 ko.
- EEPROM : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.

Les sources de l'alimentation de la carte :

On peut distinguer deux genres de sources d'alimentation (Entrée&Sortie) et cela comme suit :

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

Les entrées & sorties :

Cette carte possède 14 broches numériques (numérotée de 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions « `pin_Mode ()` » ou « `digital_Write ()` » ou « `digital_Read ()` » 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 `digital Write (broche, HIGH)`.

La carte UNO dispose 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction « `analog_Read ()` » du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction « `analog_Reference ()` » du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé.

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, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :

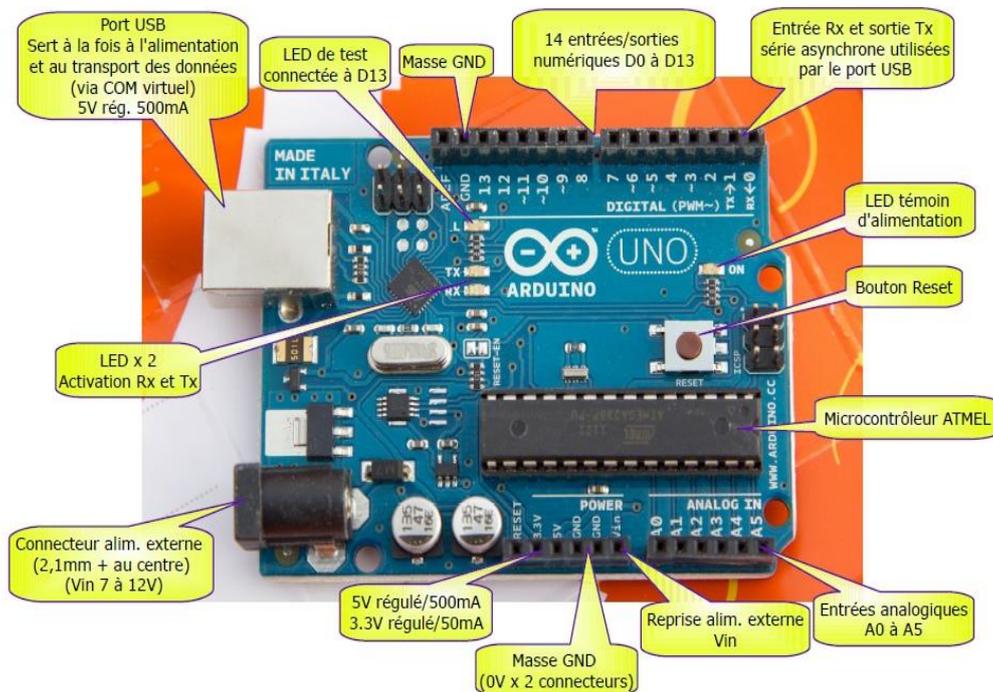


Figure II.3 : Les ports de communications de l'Arduino UNO [1]

II.2.2.2. Partie programme :

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur doit être dotée d'une interface de programmation comme est le cas de notre carte. L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

L'environnement de la programmation Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du 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 alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino.

Structure générale du programme (IDE Arduino) :

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.

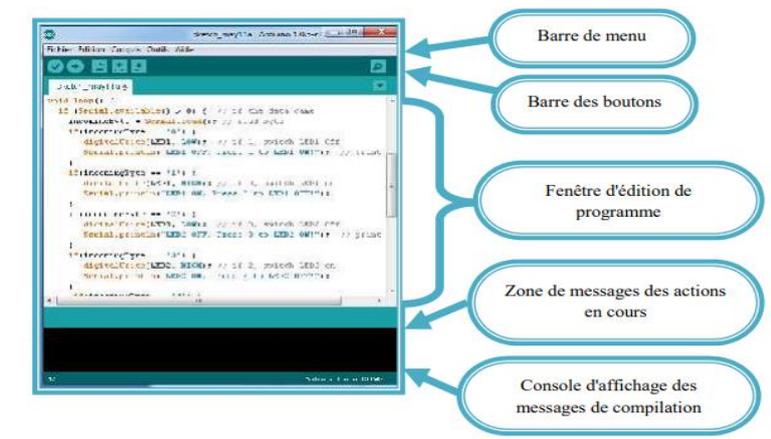


Figure II.4 : Structure générale du programme

Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 4) comme à titre d'exemple cette figure suivante.

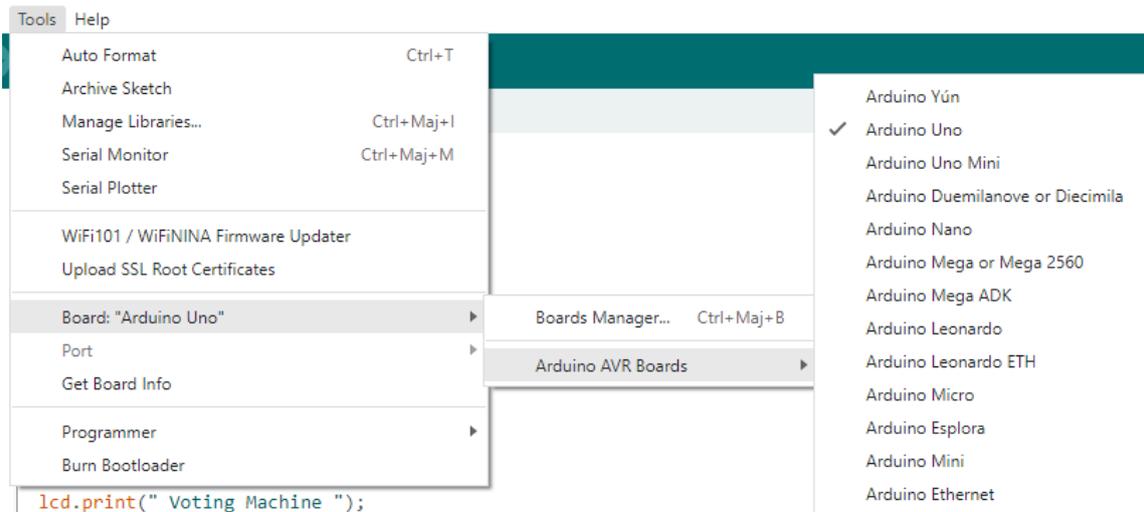


Figure II.5 : Paramétrage de la carte [2]

II.3. Les Accessoires de la carte Arduino :

La carte Arduino généralement est associée aux accessoires qui simplifient les réalisations

II.3.1. Les afficheurs LCD :

Les afficheurs LCD sont devenus indispensables dans les systèmes techniques qui nécessitent l'affichage des paramètres de fonctionnement.

Ces Afficheurs permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux « les caractères sont prédéfinis ».

Les afficheurs « LCD I2C » : sont couramment utilisés dans les projets électroniques et de robotique pour afficher des données ou des messages. Ils sont généralement compacts et peu coûteux, et peuvent être facilement intégrés dans de nombreux projets.



Figure II.6 : Afficheur LCD

Pour utiliser un **afficheur « LCD I2C »**, vous devez d'abord le connecter à un microcontrôleur ou à une carte de développement compatible en utilisant les broches SDA et SCL pour la communication en bus I2C. Vous devez ensuite écrire du code en utilisant un langage de programmation compatible avec votre microcontrôleur ou votre carte de développement pour envoyer des données à l'afficheur et lui indiquer quoi afficher. De nombreux exemples et tutoriels en ligne montrent comment utiliser **un afficheur LCD I2C** dans des projets électroniques et de robotique.

L'**écran « I2C LCD »** permet d'afficher simplement quelques informations au format texte. Cet afficheur est en particulier une **interface visuelle entre un système** (projet) et l'homme(utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploiter par l'utilisateur d'un système.

Il est fréquemment utilisé dans les applications à base microcontrôleur comme Arduino et micro-bit.

Le module **I2C LCD** contient 4 broches :

- Signal de donnée : SDA
- Signal d'horloge : SCL
- Signal d'alimentation : +VCC (5V)
- Signal de masse : GND

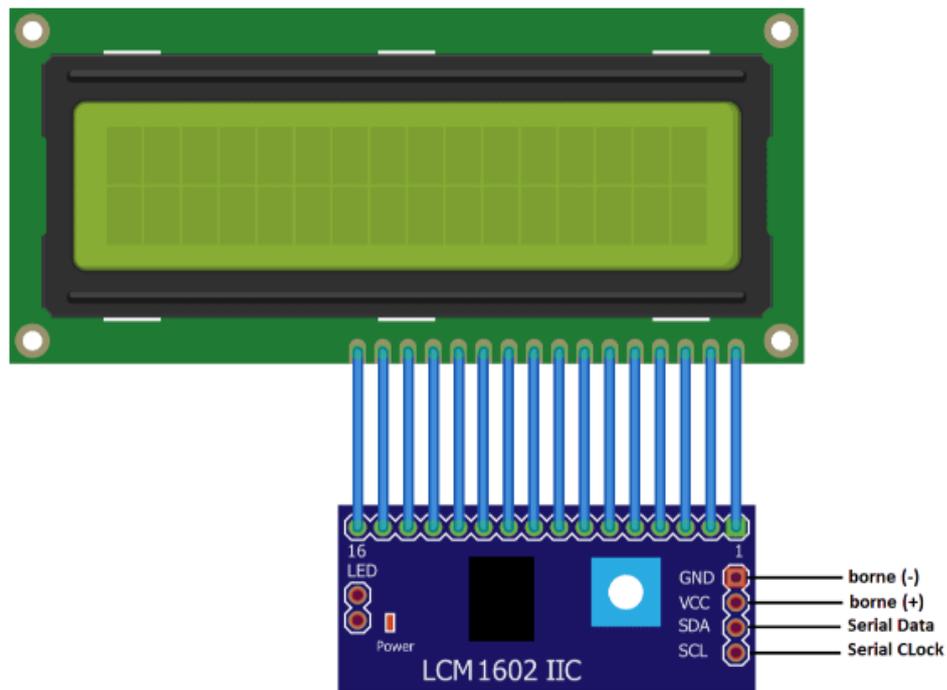


Figure II.7 : Câblage de LCD avec la LCM [3]

II.3.2. Le bouton poussoir :

Un **bouton poussoir** est un interrupteur (ou contacteur) **monostable** : il retourne seul dans la position repos (« relâché »).



Figure II.8 : Bouton poussoir monostable

II.3.3. Alimentation des cartes Arduino :

En général, on considère que tout est à peu près bon pour alimenter une carte Arduino : c'est une grosse erreur, et il faut au contraire soigneusement étudier l'alimentation du module Arduino et des périphériques et autres dispositifs associés.

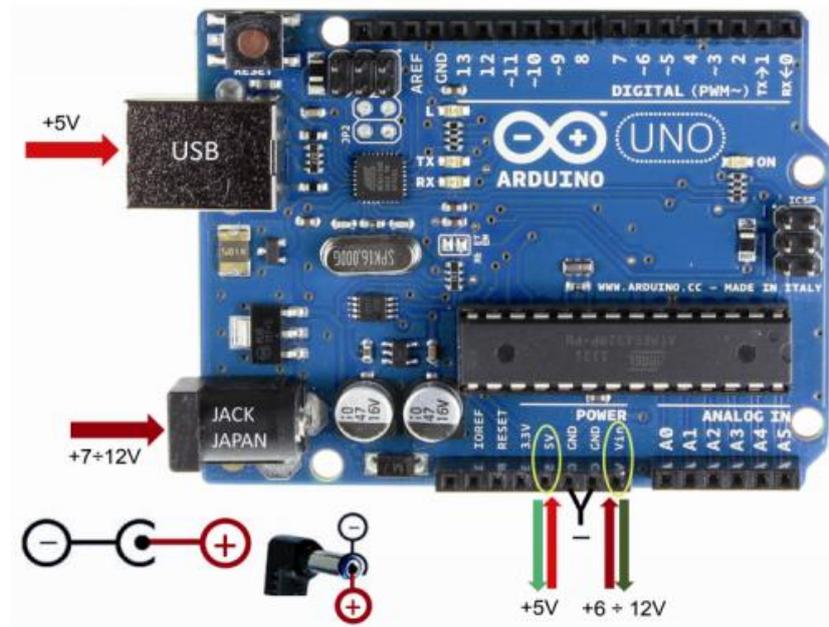


Figure II.9 : Ports d'alimentation de la carte

II.4. Conclusion :

Dans ce chapitre, nous avons projeté la lumière sur une carte d'acquisition qui est l'Arduino donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'Arduino ; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte Arduino sans oublier ses caractéristiques.

Le chapitre suivant sera consacré à l'étude et la réalisation d'un système de vote intelligent et mettre en œuvre tous les données pratiques qui on a cité dans ce dernier chapitre.

CHAPITER III : Réalisation pratique

III.1. Introduction

Après avoir donné dans le chapitre précédent le contexte et le principe de base de notre solution proposée pour réaliser un système de vote intelligent, nous présenterons dans ce présent chapitre la méthodologie de la réalisation pratique en détail de notre solution. De ce fait, nous décrirons tout d'abord l'environnement et l'implantation du matériel utilisé pour la réalisation pratique. Ensuite, nous expliquerons étape par étape la procédure de notre travail pratique qui nous a permis d'afficher les résultats de vote avec succès.

III.2. Description de l'environnement expérimental

Sur la figures « III.1 », nous décrivons notre installation expérimentale pour réaliser pratiquement l'affichage des résultats avec des simples messages qui sert facilitent à bien guider l'électeur pour rien avoir une faute ou un échec lors de l'opération comme le montre la figure, nous avons utilisé une carte Arduino UNO. Nous avons branché cette carte avec un afficheur LCD i2c (2*16) et des boutons poussoirs insérées dans un plaque à essais pour choisir l'un des deux choix oui ou non, Les sorties des boutons sont reliées directement à l'entrée des pins de la carte. Nous avons aussi utilisé une source d'alimentation externe qui permet d'alimenter notre carte avec une tension de 5v.

Dans la carte Arduino, un programme en C++ est installé qui décrit le mode de fonctionnement de notre système de vote tell que l'incréméntation des votes, l'initialisation des votes, les commentaires et bien sur ce programme décrit la fonction de chaque bouton. Pour réaliser l'affichage, nous avons installé sur la carte Arduino la bibliothèque <LCD_I2C.h> qui nous a permis d'utiliser le module I2C.

Dans la suite de ce paragraphe, nous détaillerons étape par étape la procédure pratique réalisée au cours de notre projet de fin d'année.

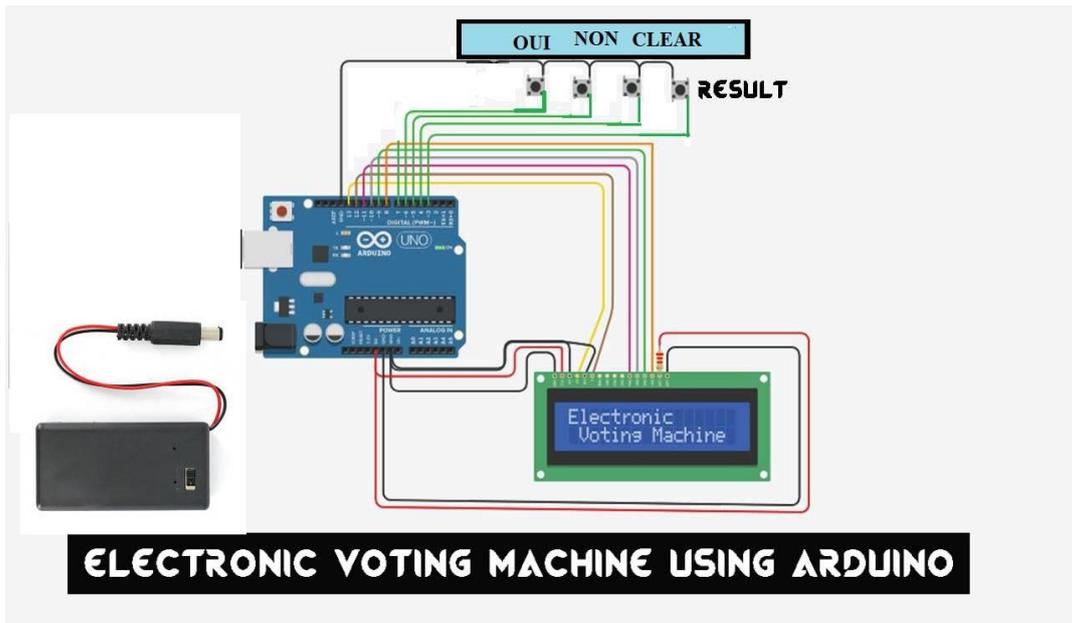


Figure III.1 : Environnement et matériel utilisés pour la réalisation pratique de notre projet de fin d'année

III.3. Procédure d'affichage des résultats de vote

III.3.1. Étape 1 : simulation sur logicielle Proteus 7 :

- **Proteus 7 :**

Proteus est une suite logicielle permettant la CAO électronique éditée par la société LAB-center Electronics. Proteus est composé de deux logiciels principaux : ISIS, permettant entre autres la création de schémas et la simulation électrique, et ARES, destiné à la création de circuits imprimés.

Grâce à des modules additionnels, ISIS est également capable de simuler le comportement d'un microcontrôleur (PIC, Atmel, 8051, ARM, HC11...) et son interaction avec les composants qui l'entourent.

Pour avoir simuler notre projet a l'aide de Proteus 7 il faut d'importer la bibliothèque Arduino ;
Ce figure ce dessous représente cette étape :

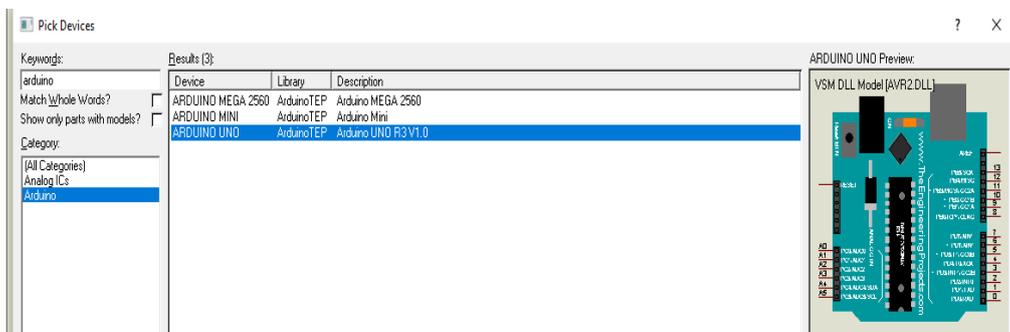


Figure III.2 : Importation des bibliothèques de Proteus 7

- Après avoir installé le bibliothèque Arduino, nous pouvons l'insérer dans l'emplacement dédié dans Proteus les différents composants et démarrer la simulation.

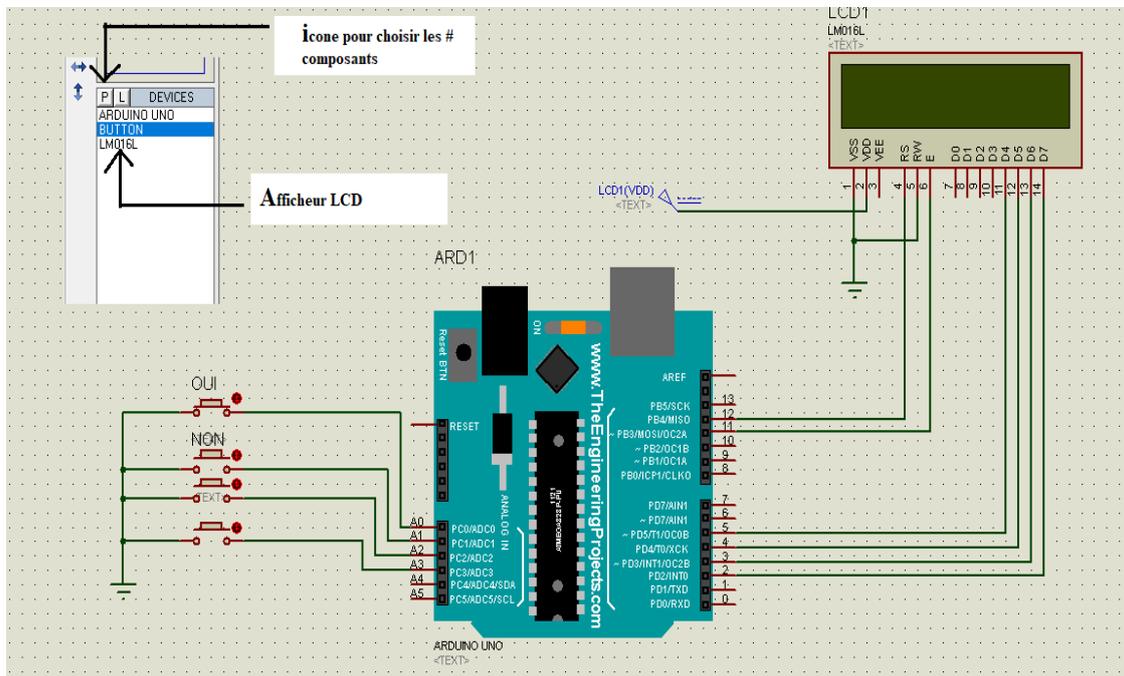


Figure III.3 : Circuit complète sur le Proteus 7

III.3.2. Étape 2 : Programmation Arduino :

Pour utiliser la carte Arduino, nous avons commencé tout d'abord par installation du logiciel Arduino IDE comme illustré sur la figure suivante :



Figure III.4 : IDE-Arduino

III.3.2.1. Importation des bibliothèques :

Après avoir installé le logiciel Arduino IDE, il faut faire des étapes de préparations pour réussir la phase de programmation, en premier lieu nous devons importées le bibliothèque LCD.I2C.

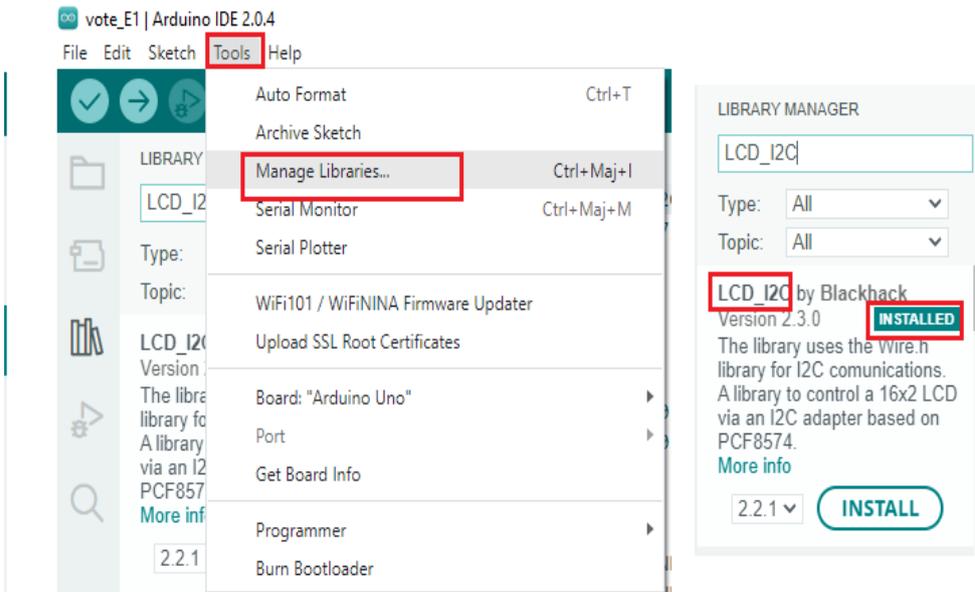


Figure III.5 : Importation des bibliothèques Arduino

III.3.2.2. Identifier l'adresse de l'afficheur LCD :

L'identification de l'adresse de l'afficheur LCD permet d'établir une communication précise avec le périphérique et de contrôler son fonctionnement de manière appropriée.

Remarque : pour réaliser cette étape il faut que l'afficheur soit implémenté sur la carte.

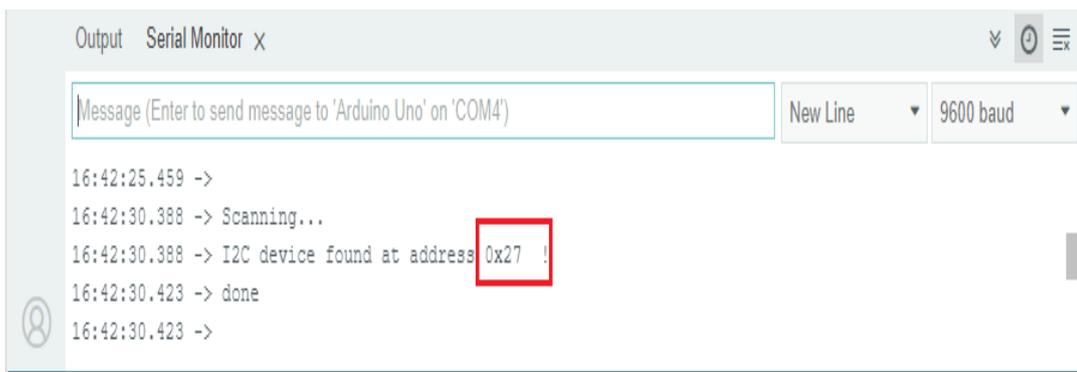


Figure III.6 : Implémentation de LCD sur la carte

III.3.2.3. Configurations des entrées /sorties :

Les boutons poussoirs (S1, S2, S3, S4) dans notre projet représentent les entrées, et l'afficheur représente la sortie.

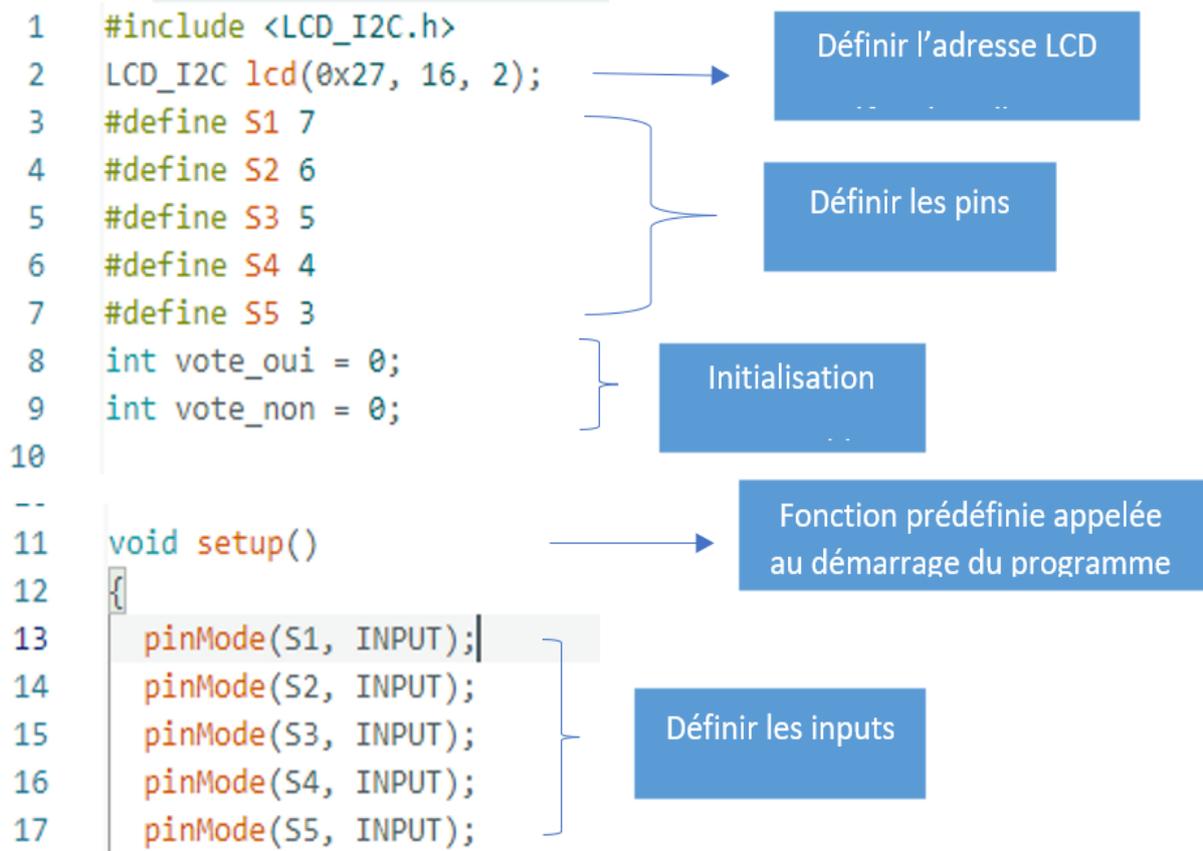


Figure III.7 : Configuration des E/S

III.3.2.4. Saisir des messages :

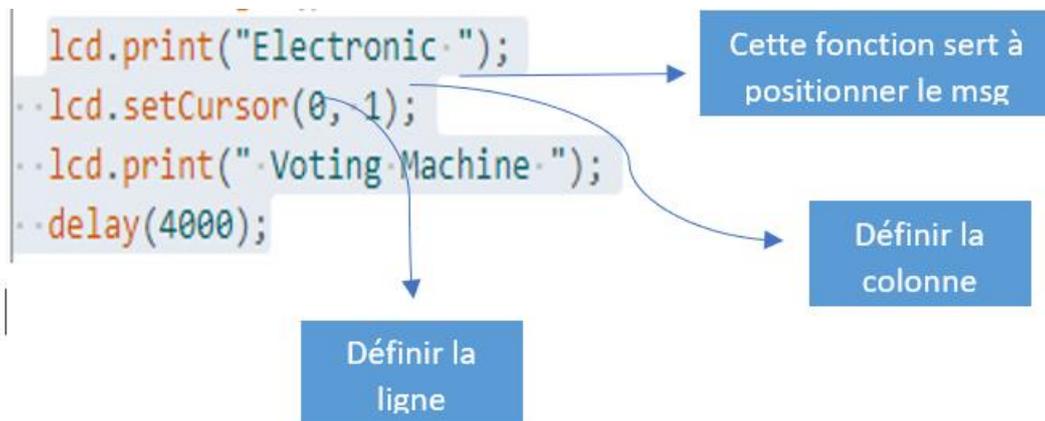


Figure III.8 : Définition des messages

III.3.2.5. Programme principale (Void Loop) :

Pour bien décrire le fonctionnement de notre système de vote, nous avons utilisés une boucle conditionnelle :

```

36 void loop()
37 {lcd.backlight();
38   lcd.setCursor(1, 0);
39   lcd.print("oui");
40   lcd.setCursor(1, 1);
41   lcd.print(vote_oui);
42
43   lcd.setCursor(13, 0);
44   lcd.print("non");
45   lcd.setCursor(13, 1);
46   lcd.print(vote_non);
47   if (digitalRead(S1) == 0)
48     vote_oui++;
49   while (digitalRead(S1) == 0);
50
51   if (digitalRead(S2) == 0 )
52     vote_non++;
53   while (digitalRead(S2) == 0 );
54

```

Contrôler la lumière d'affichage

Boucle conditionnelle

Figure III.9 : Extrait du programme

III.3.2.6. Compilation et chargement du programme :

Cette phase nous aide à vérifier syntaxiquement si notre code est vrai pour donner suite à des messages qui annoncent des erreurs ou des alertes.

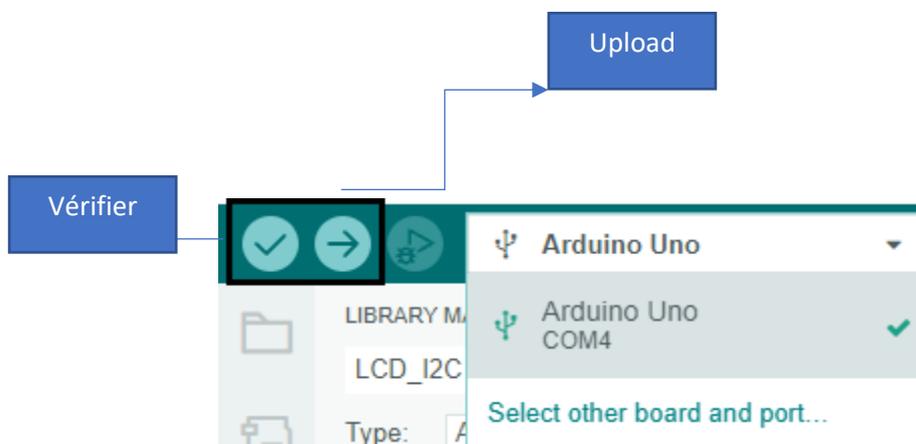


Figure III.10 : Compilation et chargement du code

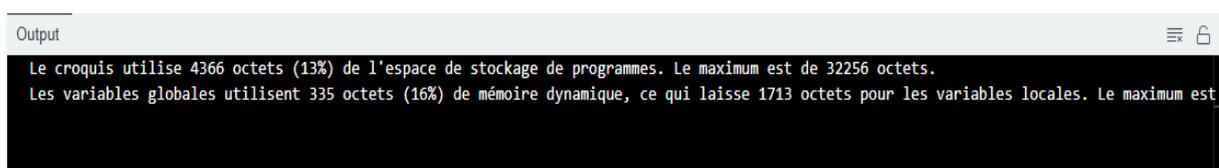


Figure III.11 : Message d'OUTPUT

III.4. Résultats pratique obtenir

Une fois tout est prêt, nous exécutons le programme sur la carte Arduino UNO par la commande Upload, on vérifie que le port de transmission de programme est bien choisi « COM4 ». Dans notre cas, nous obtenons les résultats suivants comme le montre la figure ci-dessous.

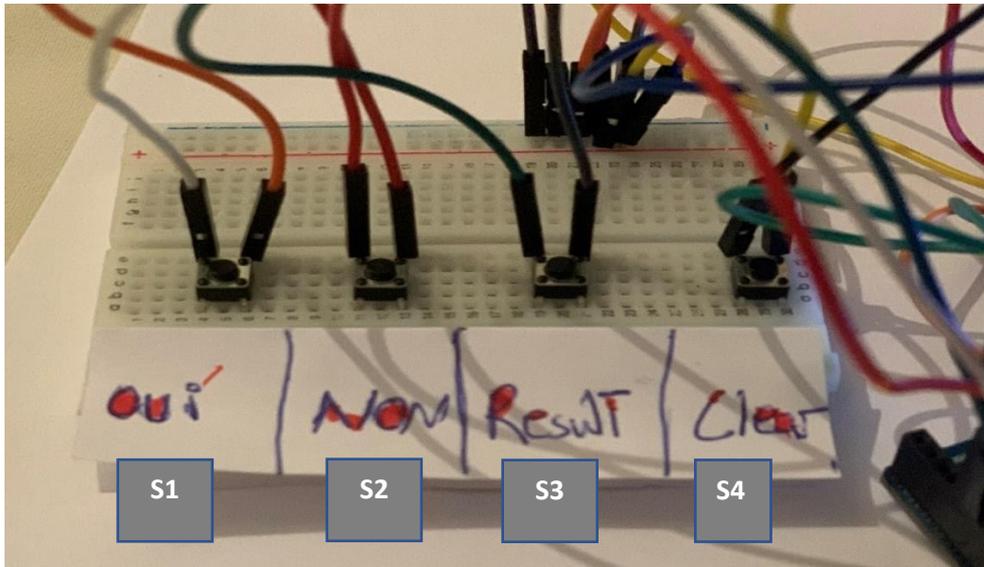


Figure III.12 : Circuit des boutons et leurs rôles

Lancement de vote au hasard pour 9 personnes, on obtient les résultats suivants :



Figure III.13 : Exécution de programme

On appuie sur le bouton S3 (Affichage des résultats) :

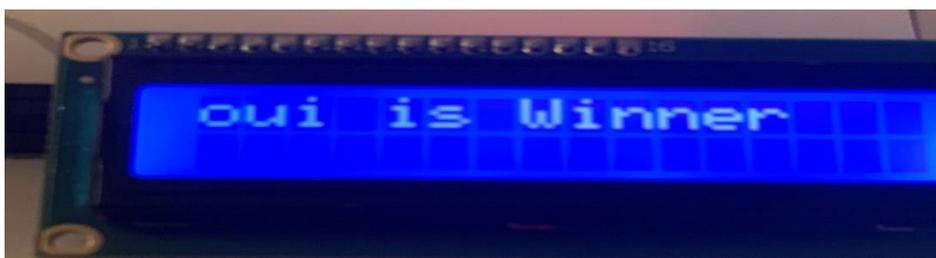


Figure III.14 : Résultat obtenu après l'appui sur S3

Pour lancer un nouvel essai de vote, on appuie sur le bouton S4 (Clear) qui permet d’initialiser les votes à 0 et affiche le message « Next round » comme montre les deux figures suivantes :

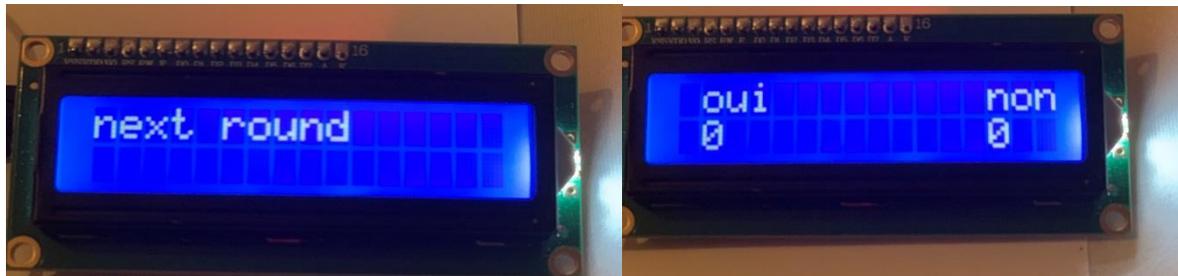


Figure III.15 : Lancement d'un nouvel essai de vote

Avec ce résultat, nous pouvons confirmer que nous avons abouti à notre but qui est l’affichage des résultats de vote.

III.5. Etude des prix :

L'étude de prix est une analyse approfondie visant à déterminer le coût total d'un produit, d'un projet ou d'un service. Elle consiste à évaluer les différents éléments de coûts associés, tels que les matières premières, la main-d'œuvre, les frais généraux, les frais de distribution, les coûts d'exploitation, etc.

L'étude de prix est essentielle pour une gestion efficace des coûts et une fixation appropriée des prix. Cette étude se base principalement à :

- L’identification des différentes composantes
- Estimation des quantités nécessaire
- Evaluation des couts unitaires
- Calcul du cout total

Tableau III : devis des composantes utilisées [4]

Composant	Quantité	Unité	Prix unitaire	Montant
Carte Arduino UNO	1	CA	42,000	42,000
Afficheur LCD (16*2)	1	A16	18,000	18,000
Boutons monostable	4	Bp	1,200	4,800
Plaque d'essai	1	PI	7,000	7,000
Nappe de fils (M/F)	1	Mf	1,800	1,800
			Prix totale (TND)	73,600

Il est important de noter que les prix des composantes peuvent varier en fonction de nombreux facteurs, tels que la demande du marché, la disponibilité des matières premières, les fluctuations des taux de change, etc. Il est donc recommandé de réaliser une étude des prix régulièrement pour obtenir les informations les plus à jour.

III.6. Conclusion :

Nous avons essayé au cours de ce chapitre d'implémenter l'environnement de travail de notre projet, nous avons aussi mis l'accent sur la phase de programmation pour faire l'affichage des résultats de vote. Les résultats trouvés montrent que nous avons satisfait les exigences du cahier des charges de notre projet de fin d'année.

Conclusion générale

En conclusion, la conception et la réalisation d'un système de vote avec affichage des résultats sont des éléments clés pour garantir la transparence, l'efficacité et la confiance dans les processus démocratiques. Un tel système permet de recueillir les votes de manière fiable, d'assurer un dépouillement précis et rapide, et de rendre les résultats accessibles au public de manière transparente. Plusieurs facteurs doivent être pris en compte lors de la conception de ce système. La sécurité est notre priorité absolue pour prévenir les tentatives de fraude et de falsification. Des mesures de sécurité strictes, y compris l'utilisation de méthodes de cryptage, de pare-feu et de contrôles d'accès, doivent être prises pour maintenir l'efficacité du processus de vote.

Ensuite, il est important de penser à la convivialité et à l'ergonomie du système. Pour assurer un vote fluide et efficace, le système doit être facile à utiliser pour les électeurs sans confusion ni difficulté. L'interface doit être facile à utiliser pour tous les utilisateurs et doit être conçue en tenant compte des besoins des utilisateurs et des limitations physiques. Dans le même temps, la fiabilité et la disponibilité du système doivent être assurées. Pour s'assurer que le système fonctionne correctement et est capable de gérer un grand nombre de votes sans interruption, des tests rigoureux et des procédures de sauvegarde doivent être en place.

Enfin, la transparence et la présentation des résultats sont des facteurs importants pour renforcer la confiance du public. Les résultats doivent être présentés d'une manière facile à comprendre et fournir des informations spécifiques sur les chiffres, les pourcentages et les tendances. Les résultats sont accessibles à tous afin que vous puissiez garantir l'équité du processus de vote et vous assurer que les résultats sont exacts.

Références Bibliographiques

[1] : <https://bentek.fr/2-arduino-uno/>

[2] : <https://www.javatpoint.com/arduino-ide>

[3] : <https://www.robotique.tech/tutoriel/afficher-un-texte-sur-lafficheur-i2c-lcd-1602a-avec-arduino/>

[4] : <https://bestbuytunisie.tn/>