

République Algérienne Démocratique et Populaire
Ministre de l'Enseignement Supérieure et de la Recherche Scientifique

Université d'Oran 2 Mohamed Ben Ahmed
Institut de Maintenance et de Sécurité Industrielle
Département De Maintenance En Instrumentation



Mémoire

MASTER ACADEMIQUE

Filière : Génie Industriel

Spécialité : Génie Industriel

**Réalisation d'une ligne de remplissage et de bouchage d'une
bouteille à base d'Arduino**

Réalisé par :

ATALLAH Fatima Zohra

Encadré par :

Mme BENAICHA Halima

Année Universitaire : 2021-2022

Remerciement

Je remercie **DIEUX** le tout puissant de m'avoir donné la santé et la volonté d'entamer et de terminer ce travail.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'encadrement de **Mme H. Benaïcha**, je vous remercie pour la qualité de votre encadrement exceptionnel, d'accorder votre confiance et votre grande indépendance à l'exécution de ce travail.

Mes remerciements s'adressent à Madame la présidente **Boulekehrass Amína** et Monsieur l'examineur **Meleki Ibrahim** qui ont accepté de jurer mon sujet.

Mes remerciements vont également à tous nos enseignants de l'Institut de Maintenance et de Sécurité Industrielle **IMSI, Université d'Oran 2**, pour leurs générosités et beaucoup de patience.

Dédicace

Ce projet de fin d'étude est dédié :

À mes chers parents et, qui m'ont toujours poussé et motivé dans mes études. Ce travail représente donc l'aboutissement du soutien et des encouragements qu'ils m'ont prodigué tout au long de mes études. Que **DIeux** leurs prête bonheur et longue vie.

À mon frère **Amine** et mes sœurs **Nacira** et **Radjaa**, pour leur amour et leur soutien inconditionnel.

À mon cher **Kadrou**, pour la patience et le soutien dont il a fait preuve pendant toute la durée de ce travail et à qui je voudrais exprimer mes affections et mes gratitude, ce travail n'aurait jamais été ce qu'il est aujourd'hui sans lui.

À mes chères amies **Karima**, **Fatima Zahra**, **Amina**, **AYA**, **Iness** et **Oumaima**, En souvenir des moments heureux passés ensemble, avec mes vœux sincères de réussite, bonheur, santé et de prospérité.

A tous ceux qui j'ai omis involontairement de citer...

Résumé

Grace à l'avancement technologique actuel, l'industrialisation moderne a fait un progrès énorme en ce qui concerne le domaine de l'automatisation des systèmes, en les rendant basés sur les API pour ce qu'elles représentent comme avantage en matière de cout, minimisation des efforts et amélioration de la productivité.

L'objectif de ce projet est de réaliser une chaîne de production d'eau embouteillée qui remplira les critères de base et assurera un bon fonctionnement. Cette machine sera automatisée via une carte Arduino Uno Rev3 et réalisée en matériel électrique et/ou électronique.

Abstract

In the last century, the industrial world has experienced a qualitative leap from the dependence of the human being to be completely dependent on the machine. Thanks to SAP, the production of any product became easier, all we need now is an idea, primary material and a machine, intended for thousands of workers and complex task. System automation increases productivity and safety and, in addition, decreases cost.

The goal of this project is to build a bottled water production line that will fulfill the basic criteria and ensure adequate operation. The machine will be automated using an Arduino UNO Rev 3 board and made of electrical and/or electronic equipment.

Tables des matières

Résumé I

Liste des figures VII

Liste des tableaux XII

Abréviation XIII

Introduction générale 1

Chapitre I : Système automatisé et Généralités sur la ligne d'embouteillage

I.1. Introduction	03
I.2. Motivation et contexte de travail	03
I.3. Présentation du cahier des charges.....	03
I.4. Introduction philosophique / historique sur l'automatisation.....	04
I.5. Généralités et définitions.....	05
I.5.1. Système technique et automatisation	05
I.5.2. Système automatisé.....	05
I.5.3. Système automatisé de production (SAP).....	06
I.5.4. Description différentes parties d'un système automatisé.....	07
I.5.4.1. Partie Opérative.....	08
I.5.4.1.1. Pré-actionneurs	08
I.5.4.1.2. Actionneurs	08
I.5.4.1.3. Capteurs	08
I.5.4.2. Partie commande.....	08
I.5.4.3. Partie relation	09
I.5.5. Fonction globale d'un système automatisé	09
I.5.5.1. Matière d'œuvre	10
I.5.5.2. Valeur ajoutée	10
I.6. Les avantages et les inconvénients d'un système automatisé	10
I.6.1. Les avantages d'un système automatisé	11
I.6.2. Les inconvénients d'un système automatisé	11
I.7. Ligne ou chaîne d'embouteillage	11

I.8. Système automatisé de ligne d’embouteillage	12
I.8.1. Principe de fonctionnement	12
I.8.2. Les étapes clés de l’embouteillage	13
I.8.2.1. Remplissage	13
I.8.2.1.1. Les modes de machine de remplissage	13
I.8.2.1.1.1. Machine de remplissage automatique	13
I.8.2.1.1.2. Machine de remplissage semi-automatique.....	13
I.8.2.1.1.3. Machine de remplissage manuelle.....	14
I.8.2.1.2. Choix des machines de remplissage.....	14
I.8.2.1.3. Les types des machines de remplissage	15
I.8.2.1.4. Description de la remplisseuse.....	15
I.8.2.2. Bouchage	15
I.8.2.2.1. Les modes de la machine de bouchage.....	15
I.8.2.2.1.1. Bouchage par enfoncement.....	16
I.8.2.2.1.2. Bouchage à vis.....	16
I.9. Conclusion	17

Chapitre II : Matériels et logiciels utilisés

II.1. Introduction.....	18
II.2. Partie Matériels.....	19
II.2.1. Choix de la carte programmable.....	19
II.2.2 Pourquoi a-t-on choisi l’Arduino ?	19
II.2.3 Qu’est-ce qu’un Arduino ?	19
II.2.4 Qu’est-ce qu’une carte Arduino ?	19
II.2.5 Les gammes de la carte Arduino	19
II.2.6 Arduino UNO Rev3	21
II.2.6.1. Pourquoi Arduino UNO Rev3 ?	21
II.2.6.2. Spécifications techniques	22
II.2.6.3. Composants	23
II.3 Capteurs	25
II.3.1. Qu’est-ce qu’un capteur ?	25
II.3.2 Capteurs utilisés	25
II.3.2.1 Capteur IR de proximité FC-51	26

II.4	Actionneur	28
II.4.1	Qu'est-ce qu'un actionneur ?	28
II.4.2	Actionneurs utilisés	28
II.4.2.1	Module d'alimentation abaisseur DC-DC LM2596HVS et Power supply.....	28
II.4.2.2	Moteur pas à pas NEMA17 et le module L298N	30
II.4.2.3	Moteur 3 – 12V gear DC et le module L293D	34
II.4.2.4	Afficheur LCD 16*2	37
II.4.2.5	Bouton poussoir Schneider	40
II.4.2.6	Servomoteur MG995 360 degrés (à rotation continue).....	41
II.4.2.7	Les relais	44
II.4.2.8	Pompe à eau 12v	45
II.5	Partie logiciel	47
II.5.1	Logiciel Arduino	47
II.5.1.1	Qu'est-ce que le logiciel Arduino ?	47
II.5.1.2	Langage de programmation Arduino	47
II.5.1.3	Structure d'un programme Arduino	47
II.5.1.3.1	L'interface du logiciel	47
II.5.1.3.2	Les étapes pour exécuter un programme dans IDE de	48
II.5.1.3.3	Rôles des instructions de la barre de commande.....	50
II.5.1.3.4	Les différentes règles à respecter pour une exécution.....	50
Correcte		
II.5.2.	Logiciel Proteus	51
II.5.2.1.	Qu'est-ce que le logiciel Proteus ?	51
II.5.2.2.	Logiciel ISIS Proteus	51
II.6	Conclusion	52

Chapitre III : Conception et réalisation pratique

III. 1. Introduction	53
III.2. Etude des besoins du système	54
III.2.1. Besoins fonctionnels	54
III.2.2. Besoins non fonctionnels	54
III.3. Principe de fonctionnement de notre machine de remplissage et de bouchage	55
III.3.1. Principales étapes du fonctionnement du système	55
III.3.2. Diagramme d'activité.....	56
III.4. Software	57
III.4.1. Programmation	57
III.4.1.1. Initialisation	58
III.4.1.2. Démarrage du système	61
III.4.1.3. Convoyeur.....	61
III.4.1.4. Remplissage.....	62
III.4.1.5. Bouchage	65
III.4.1.6. Ejection	67
III.4.2. Simulation	70
III.4.2.1. Lancement du logiciel	70
III.4.2.2. Sélection et branchement des composants.....	72
III.4.2.3. Essai.....	75
III.5. Réalisation	79
III.5.1. Convoyeur	80
III.5.1.1. Découpage	80
III.5.1.2. Assemblage.....	85
III.5.2. Mécanisme de rotation	90

III.5.2.1. Découpage	90
III.5.2.2. Assemblage	94
III.5.3. Finition	100
III.5.3.1. Mécanisme des servomoteurs	100
III.5.3.2. Support du convoyeur	103
III.5.3.3. Modification du mécanisme de rotation.....	104
III.5.3.4. Panneau de commande.....	105
III.5.3.5. Branchement	107
II.6. Conclusion.....	111
Conclusion générale.....	112
Annexe.....	113
Bibliographie.....	122

Liste des figures

Figure I.1 : Une série de systèmes automatisés destinés à souder la carrosserie des véhicules.

Figure I.2 : Exemple des systèmes automatisés.

Figure I.3 : Schéma fonctionnelle d'un système de production

Figure I.4 : Exemple d'un système automatisé d'emballage.

Figure I.5 : Architecture générale d'un système automatisé.

Figure I.6 : Exemples de capteurs

Figure I.7 : Comparaison d'un système automatisé avec le corps humain.

Figure I.8 : Fonction globale d'un système automatisé

Figure I.9 : machine de remplissage automatique.

Figure I.10 : Machine de remplissage semi-automatique.

Figure I.11 : Machine de remplissage manuel.

Figure I.12 : Un convoyeur.

Figure I.13 : Bouchage par enfoncement

Figure I.14 : Bouchage à vis avec réalisation du filetage (sans filetage préexistant).

Figure II.1 : Logo d'Arduino

Figure II.2 : Les gammes de la carte d'Arduino

Figure II.3 : Photo réelle de la carte d'Arduino UNO Rev3

Figure II.4 : Composants de la carte Arduino.

Figure II.5 : Photo réelle du capteur IR de proximité Fc-51.

Figure II.6 : Fonctionnement du capteur IR de proximité Fc-51

Figure II.7 : Branchement réel du capteur IR de proximité Fc-51

Figure II.8 : Branchement du capteur IR de proximité Fc-51 sous Proteus.

Figure II.9 : Photo réelle du Module d'alimentation abaisseur DC-DC LM2596HVS

Figure II.10 : Photo réelles du transformateur à découpage (Power supply)

Figure II.11 : Branchement réelles de l'abaisseur et du transformateur

Figure II.12 : Photo réelles du moteur pas à pas NEMA17

Figure II.13 : Photo réelles du module L298N

Figure II.14 : Branchement réel du module L298N

Figure II.15 : Branchement du moteur NEMA17 et le module L298N sous Proteus

Figure II.16 : Photo réel du moteur DC 3-12v

Figure II.17 : Photo réelle du module L293D

Figure II.18 : Branchement réel du moteur DC 3-12v et du module L293D

Figure II.19 : Branchement du moteur DC 3-12v et du module L293D sous Proteus

Figure II.20 : Photo réelle de l'afficheur LCD 16*2

Figure II.21 : Branchement réel de l'afficheur LCD 16*2

Figure II.22 : Branchement de l'afficheur LCD 16*2 sous Proteus

Figure II.23 : Photo réelle du bouton poussoir Schneider

Figure II.24 : Branchement réel du bouton poussoir Schneider

Figure II.25 : Branchement du bouton poussoir Schneider sous Proteus

Figure II.26 : Photo réelle du servomoteur MG995

Figure II.27 : Branchement réel du servomoteur MG995

Figure II.28 : Branchement du servomoteur MG995 sous Proteus

Figure II.29 : Photo réelle du module 'relais'

Figure II.30 : Photo réelle de la pompe à eau 12v

Figure II.31 : Branchement réel de la pompe à eau 12v et le module relais

Figure II.32 : Branchement de la pompe à eau 12v et le module relais sous Proteus

Figure II.33 : Logo du l'IDE de l'Arduino

Figure II.34 : Les étapes d'un programme Arduino

Figure II.35 : Fenêtre d'édition de programme

Figure II.36 : Type de la carte Arduino

Figure II.37 : Choix du port

Figure II.38 : Barre d'outils Arduino

Figure II.39 : Photo réel du téléversement du programme vers la carte Arduino

Figure II.40 : Fenêtre d'édition de programme

Figure II.41 : Logo de Proteus

Figure III.1 : Photo réelle de la découpe du 1^{er} morceau de bois

Figure III.2 : Photo réelle de la découpe du 2eme morceau de bois

Figure III.3 : Photo réelle de l'appui

Figure III.4 : Photo réelle des supports des moteurs DC

Figure III.5 : Photo réelle de la roue avant modification

Figure III.6 : Photo réelle de la roue après la modification

Figure III.7 : Photo réelle du tube PVC

Figure III.8 : Photo réelle du tube PVC après la modification

Figure III.9 : Résultat final de l'insertion de la roue à l'intérieur du tube PVC

Figure III.10 : Photo réelle du moteur DC après l'enlèvement des supports

Figure III.11 : Vue A de la fixation des moteurs sur les extrémités du support

Figure III.12 : Vue B de la fixation des moteurs sur les extrémités du support

Figure III.13 : Vue C de la fixation des moteurs sur les extrémités du support

Figure III.14 : Assemblage de l'ensemble (support+ 2 moteurs) et de l'appui

Figure III.15 : Vue A de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs

Figure III.16 : Vue B de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs

Figure III.17 : Vue C de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs

Figure III.18 : Photo réelle du tapis utilisé

Figure III.19 : Photo réelle de la découpe du tapis

Figure III.20 : Vue A du résultat finale du convoyeur

Figure III.21 : Vue B du résultat finale du convoyeur

Figure III.22 : la découpe du support

Figure III.23 : Prendre des mesures exactes pour couper le récipient

Figure III.24 : la découpe du bois utilisant la machine de bois combiné

Figure III.25 : la découpe du bois utilisant la machine à scie verticale

Figure III.26 : Préparation du récipient

Figure III.27 : Résultat final du récipient

Figure III.28 : Photo réelle du support des bouteilles

Figure III.29 : Découpe de la tige

Figure III.30 : L'enlèvement du pignon du moteur

Figure III.31 : Assemblage du moteur et de la tige via un accouplement

Figure III.32 : photo du support après l'avoir percé.

Figure III.33 : Photo réelle du nouveau corps (moteur + accouplement +tige +support
+récipient)

Figure III.34 : Fixation du moteur NEMA17 sur le support en bois

Figure III.35 : Photo réelle du support vertical fixé sur la base du bois rouge

Figure III.36 : Photo réelle du support horizontal fixé sur le support vertical

Figure III.37 : Insertion de la tige dans l'extrémité du support

Figure III.38 : Vue A du résultat final du mécanisme de rotation

Figure III.39 : Vue B du résultat final du mécanisme de rotation

Figure III.40 : Vue C du résultat final du mécanisme de rotation

Figure III.41 : Collage du support du servomoteur sur le bras

Figure III.42 : Fixation du 1^{er} bras pour caler la bouteille

Figure III.43 : Fixation du bras pour boucher la bouteille

Figure III.44 : Fixation du bras pour éjecter la bouteille remplie et boucher

Figure III.45 : Vue A du support ajouté

Figure III.46 : Vue B du support ajouté

Figure III.47 : Vue A de la modification du support

Figure III.48 : Vue B de la modification du support

Figure III.49 : Photo réelle du panneau de commande

Figure III.50 : Résultat finale du panneau

Figure III.51 : Fixation du Power Supply et de l'abaisseur

Figure III.52 : Photo réelle du branchement des composants sur la plaque d'essai

Figure III.53 : Vue A des composants fixé à côté de la plaque d'essai

Figure III.54 : Vue B des composants fixé à côté de la plaque d'essai

Figure III.55 : Vue A du branchement de la carte Arduino avec les composants

Figure III.56 : Vue B du branchement de la carte Arduino avec les composants

Figure III.57 : Vue C du branchement de la carte Arduino avec les composants

Liste des tableaux

Tableau II.1 : Caractéristiques techniques de la carte Arduino UNO Rev3

Tableau II.2: Définition des composants de la carte Arduino UNO

Tableau II.3 : Caractéristiques du Capteur IR de proximité FC-51

Tableau II.4 : Caractéristiques du Module d'alimentation abaisseur DC-DC LM2596HVS

Tableau II.5 : caractéristiques du transformateur à découpage Power supply

Tableau II.6 : Caractéristiques du Moteur pas à pas NEMA17

Tableau II.7 : Caractéristiques du module L298N

Tableau II.8 : Caractéristique du moteur DC 3-12v

Tableau II.9 : caractéristiques du module L298N

Tableau II.10 : Caractéristique de l'afficheur LCD 16*2

Tableau II.11 : Caractéristiques du bouton poussoir Schneider

Tableau II.12 : Caractéristiques du servomoteur MG995

Tableau II.13 : Caractéristique du module 'relais'

Tableau II.14 : Caractéristique du module 'relais'

Abréviation

USB : Universal Serial Bus.

SRAM : Static Random Access Memory.

PWM : Pulse Width Modulation.

E/S : Entrées /Sorties.

EEPROM : Electrically Erasable Programmable Read-Only Memory.

LED : Light-Emitting Diode.

AREF : Analog REference

SDA : Serial Data Line.

SCL : Serial Clock Line.

ICSP : In Circuit Serial Programming.

RX : Stands for Receiver.

TX : Stands for Transmitter.

GND : Ground.

LCD : Liquid Cristal Display RS : Registre Select.

RW : Read Write.

A : Anode.

K : Cathode.

IR : Infrarouge.

NO : Normaly Open.

NC : Normaly Closed.

SD : Secure Digital.

SPI : Serial Periphical Interface.

CS : ChipSelect.

MISO : Master Input, Slave Output.

MOSI : Master Output, Slave Input.

SCK : Clock.

IDE : Integrated Development Environment.

INTRODUCTION GENERALE

Introduction générale

"Depuis toujours l'homme est en quête de bien être". Cette réflexion (qui rejoint la notion de besoin) peut paraître bien éloignée d'un contexte de sciences industrielles, pourtant c'est la base de l'évolution des sciences en général, et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grand nombre, l'automatisation des tâches est alors apparue : remplacer l'homme dans des actions pénibles, délicates ou répétitives.

Parmi de milliers systèmes automatisés, nous nous sommes intéressés aux lignes d'embouteillages de production, ces dernières sont de plus en plus utilisées dans divers domaines industrielles, alimentaires, non alimentaires et autres. Plusieurs usines et grandes entreprises s'arrachent sur le marché pour produire ces machines devenues à la mode lors de ces dernières années.

Si nous nous sommes captivés par ces lignes de production, c'est qu'elles regroupent pratiquement tous les avantages et les critères d'un système automatisé de production. En effet, elles permettent de gagner un temps en or en enchainant des tâches répétitives et fatigantes pour un être humain. De plus, grâce à la redondance et aux automates développés, les machines travaillent sans arrêt 24h/24 et 7j/7, ce qui permet d'augmenter le ratio de productivité d'une façon hyper croissante. En supplément de ces avantages techniques, une ligne d'embouteillage muni d'un ERP assure une fiabilité maximale et une gestion d'ordonnancement facile, ce qui engendra une optimisation du bénéfice et un approvisionnement idéal aux probables défaillances, qui seront minimisées grâce aux maintenances préventives et curatives.

Pour tous ces bénéfices, nous avons décidé de choisir ce thème comme le projet de notre fin d'étude. Nous allons donc planifier, concevoir, simuler, réaliser et fabriquer une machine d'embouteillage afin de mieux comprendre le projet et de donner une contribution à ce sujet en mettant en disposition toutes les connaissances acquises au cours de notre cursus pluridisciplinaire.

Notre système sera piloté par une carte programmable Arduino Uno, ainsi que plusieurs capteurs et actionneurs qui assureront le bon fonctionnement de la machine. Cette dernière sera étudiée à travers une riche documentation et simulée grâce au logiciel Proteus.

Notre travail sera composé de trois chapitres :

Le premier chapitre présente les concepts de base nécessaires à la bonne compréhension du mémoire. Pour cela nous avons, en premier lieu, situé notre projet dans son contexte de travail en présentant notre cahier des charges ainsi que nos objectifs. Nous avons ensuite exposé une vue sommaire sur les différents systèmes d'embouteillages, leurs avantages et leurs différentes technologies.

Le deuxième chapitre sera réservé aux différents matériels et logiciels nécessaires pour la réalisation de notre système. Pour cela nous avons projeté en premier lieu la lumière sur les composants électroniques que nous avons utilisés en expliquant leur principe de fonctionnement tout en se basant sur la carte Arduino puisqu'elle représente le cœur de notre projet. Nous avons présenté ensuite les logiciels Arduino IDE et Proteus adoptés dans ce travail.

Le troisième chapitre sera consacré à la conception ainsi que la réalisation pratique de notre projet. Pour cela, nous décrivons d'abord les fonctionnalités ainsi que le principe de fonctionnement de notre système en déterminant son architecture globale ainsi que les diagrammes de séquences correspondantes. Nous présentons ensuite les différents schémas de simulation. Nous expliquons enfin les différentes étapes

de la mise en œuvre de notre système d'embouteillage en expliquant la partie Software constituée de la partie programmation et celle de la simulation.

Ce rapport sera terminé par une conclusion générale qui récapitule des différentes phases de notre travail en signalant ses côtés bénéfiques et énonçant quelques perspectives.

CHAPITRE I :

SYSTÈME AUTOMATISÉ ET GÉNÉRALITÉS SUR LA LIGNE D'EMBOUTEILLAGE

I.1. Introduction :

L'être humain, par sa nature, a toujours recherché le moyen d'économiser ses efforts. Il n'a jamais cessé de mettre son intelligence et son imagination au service de ce but et ceci afin de créer un partenaire qui "fera" le travail à sa place. L'arrivée récente des systèmes automatisés et appareils électroménagers tel que les robots, les aspirateurs, ... (devenant de plus en plus familiers) permettent d'éliminer bon nombre de travaux pénibles et de réaliser des tâches répétitives et fastidieuses.

L'automatisation industrielle joue un rôle de plus en plus important tant dans l'économie mondiale ainsi que dans l'expérience quotidienne. Actuellement, pour les entreprises, le but de l'automatisation est passé de la croissance ; de la productivité et de la réduction des coûts à des problèmes plus vastes.

Ce travail fait suite à l'idée du triage et du remplissage automatique des bouteilles. Le système fait appel à un microcontrôleur. Le tri des bouteilles se fait sur le tapis roulant via des capteurs IR puis ces bouteilles sont remplies en actionnant une pompe d'eau. Les convoyeurs sont utilisés dans de nombreuses industries automatisées pour déplacer des pièces d'un endroit à un autre. Le deuxième souci est le remplissage des bouteilles. Une fois la bouteille triée, le tapis roulant la transfère sous le réservoir d'eau pour le remplissage.

Les systèmes automatisés font tellement partie de notre quotidien qu'on n'a pas toujours conscience de la technicité de ces dispositifs. Les usagers interagissent très naturellement avec des machines à laver ; des portes de garage ; des ascenseurs ; des distributeurs de boissons ou des bornes automatiques d'accueil. Les lignes de production des usines modernes sont également équipées de systèmes automatisés capables d'accomplir de nombreuses tâches, parfois très complexes. Les trains et les avions embraquent de nombreux systèmes automatisés qui facilitent leur pilotage.

I.2. Motivation :

De nos jours, les systèmes automatisés sont de plus en plus répandus au sein des ateliers, usines et grandes entreprises industrielles. Ce nouveau monde datant de seulement quelques décennies n'est pas dû au hasard mais à plusieurs paramètres. En effet, l'automatisation permet d'effectuer un ensemble de tâches sans l'intervention humaine, elle permet de sécuriser, simplifier et effectuer des opérations dont la majorité sont pénibles, répétitives et exécrables dangereux. En partant de ce principe, nous avons été motivés à réaliser une machine de remplissage et de bouchage, un process qui regorge d'un nombre infini d'avantages dont ceux que nous avons cités précédemment. Nous citerons comme principaux atouts de cette ligne de production qui nous ont poussé à choisir ce thème au dépend d'autres réalisations, simulations ou études :

- Une réduction des coûts en minimisant le personnel et les équipes de nuit.
- Un gain énorme de temps, puisque la ligne d'embouteillage peut être triplement plus rapide que l'humain.
- Une fiabilité avoisinante les 100% grâce aux nombreux capteurs et fin de courses.
- Une excellente gestion de stock grâce à un ERP et un HMI de visualisation permettant d'observer et d'analyser le nombre exact de bouteilles remplies et bouchées.

I.3. Présentation du cahier des charges

Cahier de charge :

Le cahier des charges d'un projet est une description complète du projet dans le langage naturel (non technique) compréhensible par toute personne ingénieur ou non ingénieur.

Dans ce cahier, Il s'agit d'un système automatisé de production des bouteilles d'eau. Il décrit une partie du processus de remplissage, de bouchage et d'éjection des bouteilles dans un réceptacle. Le système est réalisé autour de :

- Un tapis roulant permettant le déplacement des bouteilles.
- Un poste de remplissage commandé par une pompe à eau 12v.
- Un poste de bouchage commandé par un servomoteur.

Le déclenchement de la chaîne d'embouteillage se fait par action sur un bouton START. Cette étape permettra l'acheminement des bouteilles à travers une bande transporteuse entraînée par des moteurs jusqu'à la détection de l'une d'entre elles par le premier capteur. Une bouteille est alors présente au poste de remplissage et une pompe se déclenche afin de la remplir. Ensuite, cette dernière tourne grâce au mécanisme actionné par le moteur pas à pas jusqu'à la seconde détection par un deuxième capteur. le bouchage est réalisé grâce à un jeu de servomoteurs. Le mécanisme de rotation se déclenche alors une dernière fois afin d'accomplir la dernière étape du cycle dont la tâche est d'éjecter la bouteille dans un réceptacle à l'aide d'un bras actionné par un servomoteur.

Il est à noter que le cycle ne recommencera que si les opérations de remplissage, de bouchage et d'éjection sont achevées.

I.4. Introduction "Philosophique" / "Historique" :

"Depuis toujours l'homme est en quête de bien être". Cette réflexion (qui rejoint la notion de besoin) peut paraître bien éloignée d'un cours de Sciences Industrielles, pourtant c'est la base de l'évolution des sciences en général, et de l'automatisation en particulier. L'homme a commencé par penser, concevoir et réaliser. Lorsqu'il a fallu multiplier le nombre d'objets fabriqués, produire en plus grand nombre, l'automatisation des tâches est alors apparue : remplacer l'homme dans des actions pénibles, délicates ou répétitives.

Citons par exemple quelques grands hommes, avec les premiers développements de l'ère industrielle au XVIIIème siècle, Watt, avec ses systèmes de régulation à vapeur, Jacquard et ses métiers à tisser automatiques... Une liste exhaustive serait bien difficile à établir !

Enfin, le développement des connaissances, et des outils mathématiques, ont conduit à un formidable essor des systèmes automatisés, et des systèmes asservis, dans la deuxième moitié du 20ème siècle. Certains se hasardent à rapprocher l'Automatique et la philosophie, observant d'étranges similitudes entre les processus propres à l'homme et l'approche technologique.

Mais au fait qu'est-ce qu'un système ? Bien difficile de répondre à une telle question ! Notre point de vue porte sur les systèmes de production et les systèmes pluri-techniques en général, nous pouvons néanmoins en donner une définition plus large [1].

I.5. Généralité et définitions :

Un système : est un ensemble organisé d'éléments interagissant entre eux et avec l'extérieur, dans le but de réaliser une fonction définie.

I.5.1. Système technique et automatisation :

Système technique est généralement un processus ou un dispositif industriel destiné à la production ou la transformation. Ce processus se compose d'un ensemble de machines.

L'automatisation d'un système technique consiste à assurer une commande (valeur ajoutée) en utilisant un outil technologique.

I.5.2. Système automatisé :

C'est un système qui exécute toujours le même cycle de travail et qui est programmé à l'avance, sans l'intervention de l'utilisateur. L'opérateur assure la programmation, le démarrage et l'arrêt du système. Ce dernier simplifie, sécurise et rend moins pénibles les tâches répétitives et opérationnelles. Par contre, dans un système mécanique, l'utilisateur commande et contrôle l'ensemble des opérations.

De nos jours, on trouve des systèmes automatisés un peu partout. Dans l'industrie, par exemple, on les utilise pour réaliser des tâches dangereuses ou particulièrement pénibles. (Figure 1).



Figure I.1 : Une série de systèmes automatisés destinés à souder la carrosserie des véhicules.

Exemple de systèmes automatisés. (Figure2) :



Figure I.2 : Exemple des systèmes automatisés.

I.5.3. Système automatisé de production :

Un système de production est dit automatisé lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences et/ou en étapes.

Un système de production répond au besoin d'élaborer des produits, de l'énergie ou de l'information à un coût rentable pour l'utilisateur du système.

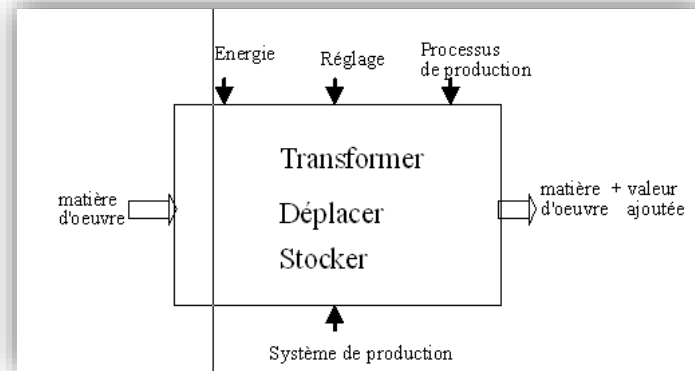


Figure I.3 : Schéma fonctionnelle d'un système de production.

Exemples de système automatisé de production



Figure I.4 : Exemple d'un système automatisé d'emballage.

Autre exemple de systèmes automatisés de production :

- Chaîne de palettisation.
- Transformer l'énergie Acquérir les états du système Moteur.
- Vérin Capteur de Position Agir sur la matière d'œuvre Ventouse.

Les systèmes automatisés, utilisés dans le secteur industriel, possèdent une structure de base identique. Ils sont constitués de plusieurs parties plus ou moins complexes reliées entre elles :

- la partie opérative (PO) ;
- la partie commande (PC) ou système de contrôle/commande (SCC) ;
- la partie relation (PR) de plus en plus intégrée dans la partie commande

Autres exemples de systèmes automatiques de production (SAP) :

- Transformer l'énergie Acquérir les états du système Moteur.
- Chaîne de palettisation.
- Vérin Capteur de Position Agir sur la matière d'œuvre Ventouse.

I.5.4. Description des différentes parties :

Un système automatisé est composé de deux parties principales : la partie opérative et la partie commande ; à lesquelles s'ajoute une troisième c'est la partie de dialogue [2] :

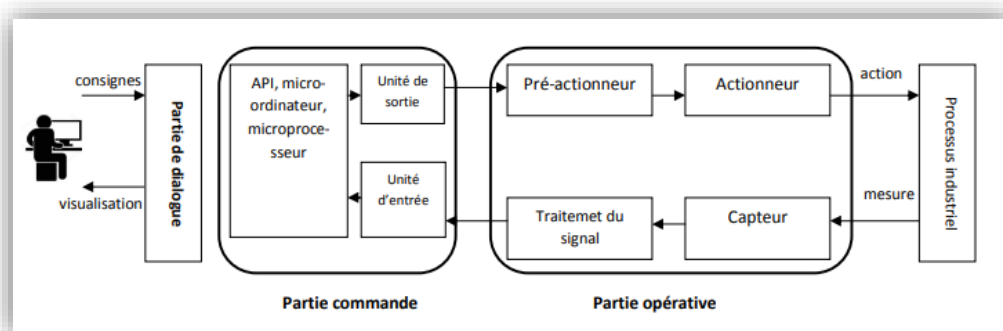


Figure I.5 : Architecture générale d'un système automatisé.

I.5.4.1. La partie opérative :

C'est la partie visible du système. Elle comporte les éléments du procédé, Il se compose des ensembles suivants :

I.5.4.1.1. Pré-actionneurs :

Distributeurs, contacteurs ; qui reçoivent des ordres de la partie commande.

I.5.4.1.2. Actionneurs :

Vérins, moteurs, vannes ; qui ont pour rôle d'exécuter ces ordres. Ils transforment l'énergie pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique.

I.5.4.1.3. Capteurs :

Les capteurs informent la partie commande de l'exécution du travail. Par exemple, on va trouver des capteurs mécaniques, pneumatiques, électriques ou magnétiques montés sur les vérins. Le rôle des capteurs (ou détecteurs) est donc de contrôler, mesurer, surveiller et informer la PC sur l'évolution du système.






				
LDR	Capteur fin course	Capteur température	Capteurs ILS	Bouton poussoir
Détecte la lumière	Détecte un appui (porte par exemple)	Mesure la température	Détecte un champ magnétique	Détecte un appui

Figure 6 : Exemples de capteurs

I.5.4.2. Partie commande :

Ce secteur de l'automatisme gère selon une suite logique le déroulement ordonne des opérations à réaliser. Il reçoit des informations en provenance des capteurs de la Partie Opérative, et les restitue vers cette même Partie Opérative en direction des pré-actionneurs et actionneurs. L'outil de description de la partie commande s'appelle le Graphe Fonctionnel de Commande Étape / Transition (GRAFCET).

I.5.4.3. Partie Relation :

Sa complexité dépend de l'importance du système. Elle regroupe les différentes commandes nécessaires au bon fonctionnement du procédé, c'est à dire marche/arrêt, arrêt d'urgence, marche automatique, etc..... L'outil de description s'appelle le Guide d'Études des Modes de Marches et d'Arrêts (GEMMA).

Les outils graphiques, que sont le GRAFCET et le GEMMA, sont utilisés par les automaticiens et les techniciens de maintenance.

Un système automatisé peut être assimilé à un Homme : (Figure 7)

- Le Cerveau est la partie commande.
- Les 5 sens sont les capteurs.
- Les Muscles sont les actionneurs.
- Les Nerfs sont les liaisons entre ses diverses parties.

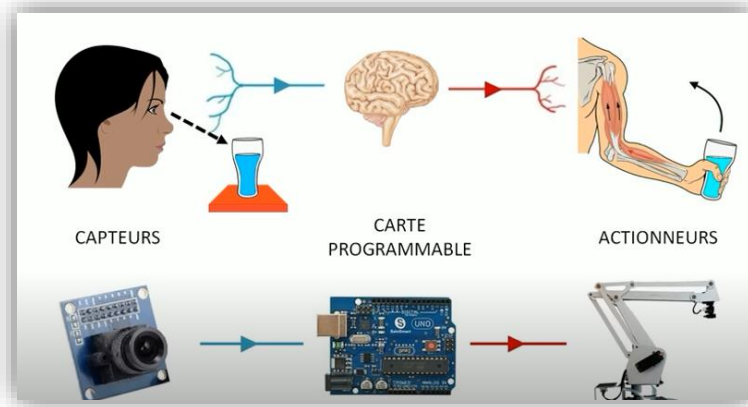


Figure I.7 : Comparaison d'un système automatisé avec le corps humain.

I.5.5. Fonction globale d'un système automatisé :

La fonction globale d'un système automatisé, donnée par la Figure 7, est de conférer une valeur ajoutée à un ensemble de matière d'œuvre (MO) dans un environnement ou un contexte donné.

Toute l'énergie nécessaire à la transformation du produit est fournie par une source extérieure ; le constituant automate dirige la succession des opérations. L'homme surveille le système et peut dialoguer avec lui par l'intermédiaire du pupitre [3].

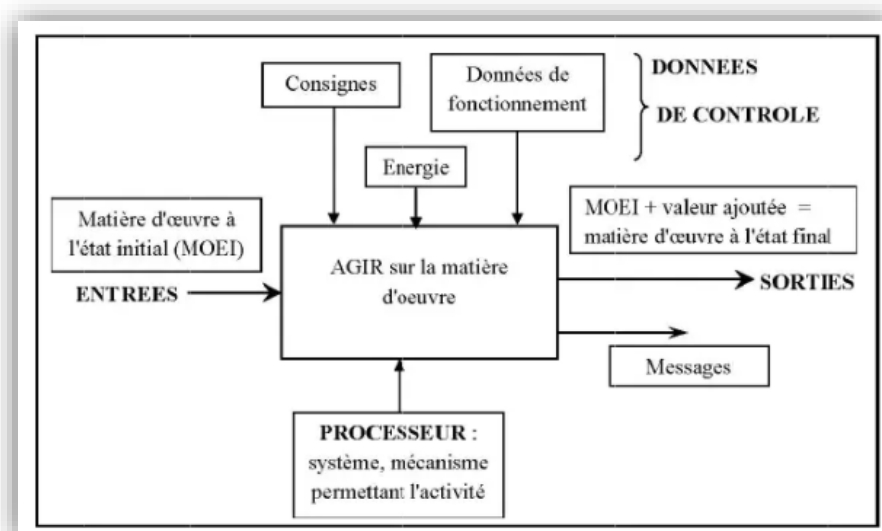


Figure I.8 : Fonction globale d'un système automatisé

I.5.5.1. Matière d'œuvre

Une matière d'œuvre peut être présentée sous plusieurs formes ; par exemple :

- Un produit : c'est à dire une matière d'état solide, liquide ou gazeux ;
- De l'énergie : sous forme électrique, thermique, hydraulique ;
- De l'information : sous forme écrite, physique, audiovisuelle ;
- L'être humain : pour les former, transporter ;

I.5.5.2. Valeur ajoutée

La valeur ajoutée à ces matières d'œuvre est l'objectif global pour lequel a été défini, conçu, réalisé et modifié le système. Elle peut résulter d'un(e) :

- Modification physique :
 - Traitement mécanique (usinage) ;
 - Traitement thermique (cuisson) ;
- Arrangement particulier (montage, emballage, assemblage...)
- Transfert : manutention, transport, stockage ;
- Prélèvement d'information (mesure, contrôle, ...) ;

I.6. Avantage et inconvénients de l'automatisation :

Certaines personnes pensent que l'automatisation finira par ronger tous les emplois humains. Il s'agit d'un blog pour évaluer si l'automatisation est une bonne ou une mauvaise chose pour la génération d'aujourd'hui et les générations futures.

La préoccupation concernant l'automatisation des tâches remonte à la revue industrielle. L'idée que le travail serait automatisé a fait craindre une baisse du rythme de travail, avec des conséquences sur l'égalité et la progression humaines. Il y a cent ans, une personne moyenne travaillait environ 15 heures par semaine ; aujourd'hui, nous travaillons en moyenne 40 heures par semaine. L'automatisation s'est produite à la fois dans le secteur manufacturier et dans le secteur des services. Dans la fabrication, les robots ont été utilisés pour fabriquer plus rapidement et plus efficacement des pièces pour les marchandises. Dans les industries de services, des programmes de service client automatisés ont émergé qui ressemblent souvent à des appels automatisés.

La question de la sécurité de l'emploi n'est pas nouvelle. En fait, c'est un sujet de débat dans le monde du travail depuis des décennies. Ce qui est nouveau, c'est la capacité des ordinateurs à effectuer certaines tâches mieux et plus efficacement que les humains. Que ce soit par le développement de systèmes automatisés ou les progrès de l'intelligence artificielle et de l'apprentissage automatique ; nous continuerons à voir plus d'automatisation dans divers domaines car il deviendra possible et rentable pour les entreprises d'automatiser certaines tâches.

L'automatisation peut vous faire gagner du temps et de l'argent gratuitement. Si vous avez affaire à des gens. L'automatisation peut augmenter la productivité L'automatisation peut vous rapporter une belle augmentation si vous devez traiter avec des personnes. Inconvénients de l'automatisation Les gens ne vous aimeront pas si vous utilisez l'automatisation. Les gens ont besoin d'interagir avec les gens à un moment donné de leur vie. Si vous refaites la même chose à plusieurs reprises, l'automatisation peut en fait vous rendre moins efficace. Si vous faites la même chose encore et encore, l'automatisation peut vous rendre moins efficace. Vous êtes finalement contrôlé par la machine. Vous n'êtes plus libre de choisir vos tâches et vos projets. Êtes-vous prêt à laisser votre salaire diminuer ? Êtes-vous prêt à moins traiter avec les gens ? Il y a plein de gens qui ne le seraient pas.

I.6.1. Les avantages d'un système automatisé

- Améliorer les conditions de travail (effectuer des tâches pénibles, dangereuses et répétitives),
- Sécurité,
- Précision,
- Réduire les coûts de fabrications (produit plus compétitif),
- Augmenter la productivité (réduire le temps de travail nécessaire à la production, donc augmenter les cadences de travail),
- Flexibilité (une machine peut s'adapter à plusieurs productions),
- Confidentialité (une machine ne peut pas parler),
- Un système automatisé peut travailler 24h sur 24h,
- Maintenir la qualité.

I.6.2. Les inconvénients d'un système automatisé

- Incidence sur l'emploi (licenciement – chômage),
- Investissement pour l'achat de machines,
- Coût de maintenance et pannes,
- Consommation d'énergie,
- Formation d'un personnel plus qualifié (technicien de maintenance, de control, etc.) [4].

I.7. Ligne ou chaîne de production (définition) :

Qu'est-ce qu'une ligne ou une chaîne de production ?

Une ligne de production représente une série de postes de travail organisés ayant pour fonction finale de produire un bien ou un service. On parle aussi de chaîne de production qu'on retrouve notamment au sein des usines, des fabriques, des manufactures et autres bureaux d'activité.

La ligne de production est aussi l'ensemble des processus nécessaires et structurés pour la fabrication d'un produit manufacturé ou service allant des actions ou opérations de transformation des matières premières ou de produits semi-finis jusqu' à la distribution sur le marché de l'offre et de la demande.

À toute chaîne de production, on associe une capacité de production à même de pouvoir répondre aux exigences du marché – consommateurs ; fournisseurs ; sous-traitants ; etc.

Au fond, la ligne de production, c'est le lieu de fabrication du bien ou service incluant tous les mécanismes utiles, organisés et coordonnés – bâtiment industriel ; machines ; équipements ; maintenance ; contrôle qualité ; système intelligent d' information ; capital humain performant ; etc.

En filigrane, on comprend que la ligne de production nécessite des actions en amont – ou actions à l'origine – telles que le transport de marchandises des matières premières ou produits semi-finis. Cette partie en amont est appelée la chaîne d'approvisionnement ou la *supply chain*.

Puis viennent des actions en aval – ou actions par la suite – telles que la livraison des produits finis qu'on appelle la chaîne de distribution.

L'ensemble de ces techniques, de l'approvisionnement à la distribution en passant par la ligne de production ou le stockage, est appelé la logistique dont la fonction complexe est centrale dans tout business. Pour une entreprise donnée, la chaîne de production est celle qui réalise – en général – la part la plus élevée en valeur ajoutée si l'on considère toute la chaîne de valeur.

Voici un exemple d'une subdivision ou d'une décomposition d'une ligne de production classique :

- Zone de réception des ressources naturelles, matières premières ou produits semi-finis – arrivée par trains, camions, avions, drones, etc. ;
- Zone de stockage des marchandises à l'arrivée – déplacements de palettes par robots, drones, par matériel ou chariot de manutention, etc. ;
- Zone d'usinage et d'assemblage – équipements robotiques pour transformer les matières premières ou produits semi-finis en produits finis ;
- Zone d'expérimentations et de tests – laboratoires de recherche et de développement, d'analyses des produits pour détecter des défauts, défaillances ou pour amplifier la qualité ;
- Zone de contrôle qualité – contre-vérification qualitative et intensive des produits finis ;

I.8. Système automatisé de ligne d'embouteillage :

I.8.1. Principe de fonctionnement

Pour obtenir une bouteille d'eau consommable, on passe par plusieurs étapes. Tout d'abord, les bouteilles sont à l'origine des préformes. Ces dernières sont basculées à une machine où elles sont chauffées puis soufflées pour obtenir une bouteille vide prête au remplissage avec de l'eau minérale. Ces bouteilles sont transportées grâce à un convoyeur à air jusqu'à une autre machine pour être remplies et bouchées. Comme tout produit consommable, les bouteilles comportent une DLUO (Date Limite d'Utilisation Optimale), et le numéro du lot, qui sont marqués au laser afin d'assurer la traçabilité des produits. Ensuite Les bouteilles seront mises sur un convoyeur mécaniques jusqu'à une machine qui se charge de coller l'étiquette sur les bouteilles qui sont pleines, bouchées, contenant une date d'expiration. Ensuite les bouteilles sont contrôlées par un ouvrier ou par une machine de contrôle. Les bouteilles qui ne respectent pas les normes à la fin de la chaîne seront éjectées. Le reste de bouteille satisfaisant les normes seront transportés dans des rails pour être regroupées dans des cartons par des ouvrier. À la fin, les cartons contenant les bouteilles seront scotchés par une machine.

I.8.2. Les étapes clés de l'embouteillage

La mise en bouteille nécessite deux étapes « remplissage et bouchage »

I.8.2.1. Remplissage

Les machines de remplissage sont des machines destinées pour remplir et conditionner différents produits alimentaires afin d'atteindre un produit fini. Comme beaucoup de secteurs qui existent on trouve de différents types et gammes des machines dans chaque domaine.

Les bouteilles vides sont transportées jusqu'à l'étoile d'entrée de la remplisseuse, qui les positionne sur les fourches des vannes. Les opérations de remplissages commencent.

I.8.2.1.1. Les modes des machines de remplissage :

I.8.2.1.1.1. Machine de remplissage automatique :

Les machines de remplissage automatique sont des machines qui exécutent les opérations de remplissage de façon automatique sans l'intervention humaine. Ces machines ont pour principale avantage la facilité de travail en cas de cadence élevé donc une productivité élevée.



Figure I.9 : machine de remplissage automatique.

I.8.2.1.1.2. Machine de remplissage semi-automatique :

Les machines de remplissage semi-automatique exécutent une partie des opérations, le reste nécessite une intervention humaine pour accomplir l'état final du produit. Ces machines sont caractérisées par un rendement faible.



Figure I.10 : Machine de remplissage semi-automatique.

I.8.2.1.1.3 Machine de remplissage manuel :

Les machines de remplissage manuel présentent une simple installation où toutes les opérations sont accomplies manuellement.



Figure I.11 : Machine de remplissage manuel.

I.8.2.1.2. Choix des machines de remplissage :

Il existe différents critères en site parmi eux :

- 1) Le contenant (bouteilles, sac, bidons).
- 2) Le contenu (liquide, solide).
- 3) Le mode de fonctionnement.
- 4) La cadence de production à atteindre selon le volume du contenant.
- 5) La charge.
- 6) La technologie de remplissage ou le type de bec.
- 7) Le pas de la machine, c'est-à-dire la distance entre deux bacs de remplissage qui détermine le plus petit et le plus grand diamètre de bouteille possible pour la circulation dans la machine.
- 8) L'environnement de travail

I.8.2.1.3. Les types des machines de remplissage :

Les remplisseuses peuvent être divisées en :

- Machine de remplissage liquide.
- Machine de remplissage de pâte.
- Machine de remplissage de poudre.
- Machine de remplissage de granulés.

Nous allons nous intéresser par la machine de remplissage liquide.

I.8.2.1.4. Description de la remplisseuse :

La machine que nous allons l'élaborer sert à remplir les produits liquides automatiquement. Elle est équipée d'un :

- 1) Vérin

- 2) Convoyeur qui permet la circulation des bouteilles par un moteur pas à pas.



Figure I.12 : Un convoyeur.

- 3) Capteur qui se situe dans le bord du convoyeur permet la détection de la bouteille.
- 4) Pupitre qui contient les différents boutons de marche d'arrêt et les voyants.

I.8.2.2. Bouchage

Le bouchage est l'étape finale de l'usine embouteillage eau. Les bouteilles sont déplacées vers le haut vers une machine de bouchage qui dépose un bouchon sur chacune d'entre elles et scelle l'eau dans une atmosphère protégée. Après le bouchage la bouteille est portée, à l'aide d'un guide, à l'étoile et de là sur la bande de sortie. Les bouteilles sortent de la machine, continuant leur chemin en passant par une machine de marquage qui indique sur les bouteilles la date d'expiration grâce à un laser, et entrent dans le cycle de production des machines installées en aval.

I.8.2.2.1. Les modes de la machine de bouchage :

Plusieurs types de boucheuses existent, adaptés aux divers types de bouchons. Le bouchage peut être effectué en ligne, les cadences sont alors limitées à 100 bouchages par minute, ou sur un carrousel, ce qui permet, avec un nombre élevé de têtes, d'atteindre 5 000 bouchages par minute.

I.8.2.2.1.1. Bouchage par enfoncement

Le bouchon, en liège ou en plastique, est comprimé par une série de cylindres puis enfoncé par une presse (figure 13). Il est légèrement lubrifié par une cire afin d'en faciliter le glissement.

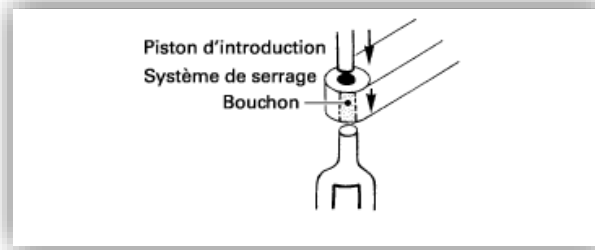


Figure I.13 : Bouchage par enfoncement

I.8.2.2.1.2. Bouchage à vis

Sans filetage préexistant :

Une capsule en métal ductile est enfoncée sur le filetage de la bouteille par des molettes en acier (figure 14). Ce système convient aux bouteilles en plastique (PET en particulier).

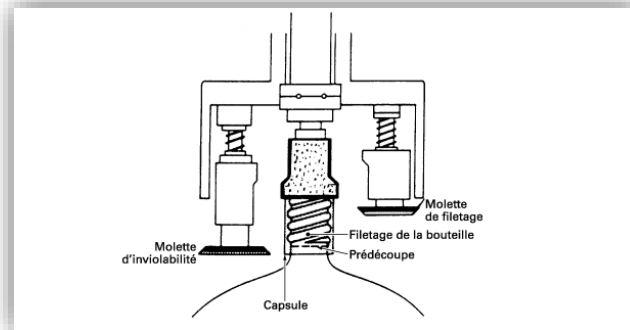


Figure I.14 : Bouchage à vis avec réalisation du filetage (sans filetage préexistant).

Avec filetage préexistant :

Des mandrins pneumatiques ou mécaniques sont utilisés. Les mandrins pneumatiques comportent un anneau flexible avec un trou central et une arrivée d'air comprimé. Lorsque la pression d'air est appliquée, l'anneau flexible gonfle ou se rigidifie, il enserre alors le bouchon fileté, le mandrin descend sur la bouteille et le visse automatiquement. Le couple de vissage est contrôlé par un embrayage pneumatique : il comporte plusieurs disques qui sont comprimés lorsque la pression est appliquée, ce qui permet à l'axe motorisé d'entraîner le mandrin. Quand le bouchon est suffisamment vissé pour que le couple devienne égal à la force

appliquée sur les disques, les disques commencent à glisser latéralement et cela déconnecte l'axe motorisé du mandrin. Il existe aussi des mandrins mécaniques et des embrayages mécaniques à ressorts.

Ces systèmes conviennent pour toutes les bouteilles et tous les flacons, en verre en ou plastique.

I.9. Conclusion :

Cette partie permet de comprendre la structure d'un Système Automatisé de Production et de définir les différentes parties de ce système.

Dans ce chapitre nous avons donné un aperçu général sur la ligne d'embouteillage, et le principe de fonctionnement de ce système, tout en justifiant les raisons qui nous ont motivés à choisir ce thème

CHAPITRE II :

MATÉRIELS ET LOGICIELS UTILISÉS

II.1. Introduction :

Comment faire des montages électroniques, simplement, en utilisant un langage de programmation ? La réponse, c'est l'Arduino qui l'apporte, celui-ci a été conçu pour être accessible à tous par sa simplicité. Mais il peut également être d'usage professionnel, tant les possibilités d'application sont nombreuses. Ces cartes polyvalentes sont donc parfaites pour nous, débutants, qui ne demandent qu'apprendre et progresser. Grâce à cette carte, ces fonctions sont réalisées par des capteurs, des actionneurs, des automates et plus généralement par des équipements électriques et/ou électroniques.

Dans ce chapitre nous allons présenter les différents matériels et logiciels nécessaires pour la réalisation de notre projet de ligne d'embouteillage.

II.2. Partie Matériels

Dans cette partie, nous allons évoquer la carte programmable choisie, ainsi que les différents composants électroniques utilisés pour réaliser notre machine, ainsi que l'unité d'alimentation qui est indispensable pour alimenter notre système.

II.2.1. Choix de la carte programmable :

Pour nos projets électroniques, nous avons souvent besoin d'une carte programmable. Cependant, parmi de centaines de cartes qui existent sur le marché, il est difficile de faire un choix adéquat. Ce choix doit être effectué selon notre projet et le cahier des charges proposé.

Nous pouvons donc faire une étude comparative entre les principales plateformes de développement de ces cartes en tenant compte de leurs avantages, inconvénients, fonctionnalités, complexité ainsi que leurs coûts.

Grâce à leur grande variété et leur adaptation à certaines applications spécifiques, nous avons constaté que les cartes Arduino sont les microcontrôleurs les mieux adaptés à notre système de machine de remplissage et de bouchage, puisqu'elles sont suffisantes et leur rapport qualité-prix est abordable.

II.2.2. Pourquoi a-t-on choisi l'Arduino ?

Le choix de l'Arduino est justifié par plusieurs paramètres tels que :

- **Open Source et Open Hardware** : Possibilité d'utilisation du schéma d'origine et le modifier ensuite pour produire la carte et la vendre sans avoir à payer des droits d'auteur.
- **Tarifs extrêmement abordables** : Le schéma étant en libre accès, différentes entreprises produisent les cartes et les exposent au marché par un coût réduit.
- **Echanges nombreux au sein de la communauté Arduino** : Il existe maints forums et documentations consultables en ligne afin de pouvoir surmonter les difficultés rencontrées aux branchements ou à la programmation.
- **Multiplateforme** : Afin de programmer une carte Arduino, il faut la connecter à un ordinateur puis utiliser le logiciel Arduino (IDE) qui permet de programmer la carte. L'IDE est disponible sur Windows, Mac OSX et Linux, contrairement à d'autres microcontrôleurs.
- **Clarté de l'environnement de programmation** : Conçu pour des débutants, Arduino est très instinctif et donnera le goût à la programmation, il est d'ailleurs très facile de débiter sur Arduino.

I.2.3. Qu'est-ce qu'un Arduino ?

Arduino est une plate-forme de prototypage open-source d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation [5].



Figure II.1 : Logo d'Arduino

II.2.4. Qu'est-ce qu'une carte 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. Cette carte est équipée essentiellement d'un microcontrôleur, d'un port USB ainsi que des connecteurs d'entrées/sortie (plus ou moins nombreux selon les modèles) [6].

Le microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer et piloter des tâches très diverses.

La carte Arduino repose sur deux parties, le **Hardware** et le **Software** dont le **Hardware** est la carte électronique programmable, quant au **Software** c'est l'interface de programmation avec un langage spécifique basé sur les langages C et C++.

La carte Arduino est utilisée dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée (la domotique, le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles).

II.2.5. Les gammes de la carte Arduino

La carte Arduino désigne toute une gamme des cartes à des microprocesseurs. Afin de choisir celle qui convient le plus à notre projet, il est utile de donner les spécifications techniques aux différentes cartes. Nous citons quelques-uns par la suite, données par Figure.II.3.

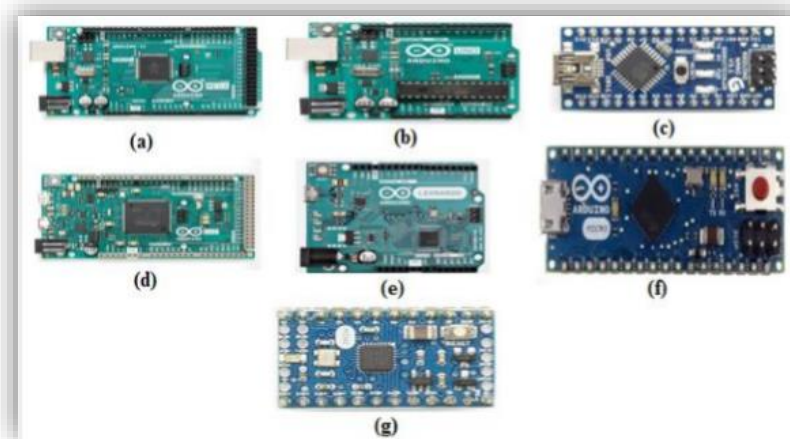


Figure II.2 : Les gammes de la carte d'Arduino.

- **Arduino Mega2560 Rev3** : Cette carte, donnée par Figure.II.2. (a), dispose de 16 entrées analogiques, de 15 canaux PWM et d'un matériel compatible avec les Shields Arduino. La compatibilité logicielle avec Mega ne peut pas toujours être garantie. Le processeur est cadencé à 16 MHz et comprend 256 Ko de ROM, 8 Ko de RAM, 4 Ko d'EEPROM et fonctionne à 5 V, ce qui facilite son utilisation avec la plupart des appareils électroniques conviviaux.
- **Arduino Uno** : Cette carte, illustrée par Figure.II.2. (b), est sans doute l'Arduino le plus populaire. Il est alimenté par un processeur Atmega328 qui fonctionne à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'eprom, 2 Ko de RAM, 14 Entrée/ Sorties numériques, 6 entrées analogiques et un rail d'alimentation de 5V et 3,3V.
- **Arduino Nano** : Cette carte, donnée par Figure.II.2.(c), est essentiellement un Arduino UNO réduit. Le Nano est alimenté par un processeur Atmega328 fonctionnant à 16 MHz, comprend 32 Ko de mémoire programme, 1 Ko d'eprom, 2 Ko de RAM, 14 entrées-sorties numériques, 6 entrées analogiques et des rails d'alimentation 5V et 3,3V. Contrairement au système UNO, le Nano ne peut pas se connecter aux platines de prototypages. Les cartes Arduino Nano sont souvent l'option de carte Arduino la moins chère, ce qui les rend rentables pour les grands projets.
- **Arduino DUE** : Cette carte, donnée par Figure.II.2.(d), est l'une des cartes les plus grandes et la première carte Arduino à être alimentée par un processeur ARM. Alors que l'UNO et Nano fonctionnent à 5V, la DUO fonctionne en 3,3V – il est important de le noter, car une surtension endommagerait irrémédiablement la carte. Alimenté par un Cortex-M3 ATSAM3X8E cadencé à 84 MHz, le DUE dispose de 512 Ko de ROM et de 96 Ko de RAM, de 54 broches d'entrée/sortie numériques, de 12 canaux PWM, de 12 entrées analogiques et de 2 sorties analogiques.
- **Arduino Leonardo** : L'Arduino Leonardo, donné par Figure.II.2.(e), est une carte à microcontrôleur basée sur le composant ATmega32u4. Elle dispose de 20 entrées/sorties digitales (dont 7 peuvent être utilisées comme sorties PWM et 12 en tant qu'entrées analogiques), un oscillateur cristal 16 MHz, une connexion micro USB, un jack d'alimentation, un connecteur ICSP, et un bouton reset. Cette carte dispose aussi de tout le

nécessaire pour supporter le microcontrôleur. Elle peut être connectée à un ordinateur à l'aide d'un câble USB et alimentée avec un adaptateur AC/DC ou une batterie pour démarrer.

- **Arduino Micro** : Cette carte, donnée par Figure.II.2.(f), est basée sur le microcontrôleur ATmega324, c'est une version compacte et réduite de l'Arduino Leonardo. Il dispose de 20 entrée/sorties numériques (dont 7 peuvent être utilisées comme sorties PWM), 14 entrées analogiques, d'un oscillateur à quartz de 16 MHz, une connexion micro USB.
- **Arduino Mini** : Cette carte, donnée par Figure.II.2.(g), est basée sur le microcontrôleur ATmega328 en version 5.0, c'est une version compacte et réduite de l'Arduino. Il dispose de 14 entrées/sorties numériques (dont 6 peuvent être utilisées comme sorties PWM), 8 entrées analogiques dont 4 sont réparties sur les pins, d'un oscillateur à quartz de 16 MHz, d'une connexion RS TTL, un connecteur ICSP, et un bouton de reset.

II.2.6. Arduino UNO Rev3 :

Parmi les types d'Arduino cités ci-dessus, nous avons opté pour une carte Arduino UNO Rev3. Cette carte présentera le cœur de notre projet.



Figure II.3 : Photo réelle de la carte d'Arduino Uno Rev3

II.2.6.1. Pourquoi Arduino UNO ?

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit :

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino préassemblées coûtent moins de 2500 Dinars).
- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows

- Un environnement de programmation clair et simple : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également
- Logiciel Open Source et extensible : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le 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 Atmel 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.2.6.2. Spécifications techniques

Tableau. II.1 présentant les principales caractéristiques de la carte Arduino UNO Rev3.

Tableau II.1 : Caractéristiques techniques de la carte Arduino UNO Rev3

Microcontrôleur	ATMEGA328
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12 V
Broches d'E / S numériques	14 (dont 6 disposent d'une sortie PWM pour commander les moteurs)
Broches d'E / S numériques PWM	15 (D2-D13 ; D44-D46)
Broches d'entrée analogique	6 (utilisables aussi en broches E/S numériques)
Courant CC par broche d'E / S	20 mA
Courant CC pour broche 3,3 V	50 mA
Mémoire flash	32 Ko
Mémoire RAM (mémoire volatile)	2 Ko
EPROM (mémoire non volatile)	1 Ko
Vitesse de l'horloge	16 MHz
LED_BUILTIN	D13
Dimensions	74 x 53 x 15 mm
Consommation maxi admise sur port USB (5V)	500 mA avant déclenchement d'un fusible
Intensité maxi disponible par broche E/S (5V)	40 mA par sortie, mais ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)

II.2.6.3. Composants

Les composants de la carte Arduino représentées par Figure II.4 sont détaillés dans Tableau II.2.

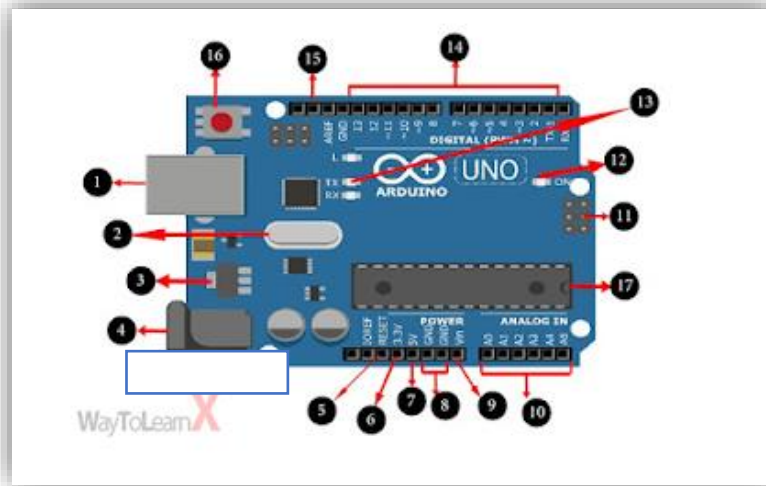















Figure II.4 : Composants de la carte Arduino.

Tableau II.2 : Définition des composants de la carte Arduino UNO

Numéro du Composant(s)	Définition
1	Power USB (1) La carte Arduino peut être alimentée à l’aide du câble USB de votre ordinateur. Tout ce que vous devez faire est de connecter le câble USB à la connexion USB.
2	Crystal Oscillator (2) L’oscillateur aide Arduino à traiter les problèmes de temps. Comment Arduino calcule-t-il le temps ? La réponse est, en utilisant l’oscillateur. Le nombre imprimé sur l’oscillateur Arduino est de 16.000H9H. Il nous dit que la fréquence est de 16 000 000 Hertz ou 16 MHz.
3	Régulateur de tension (3) La fonction du régulateur de tension est de contrôler la tension donnée à la carte Arduino et de stabiliser les tensions continues utilisées par le processeur
4	Power (Barrel Jack) (4) Cartes Arduino peuvent être alimentés directement à partir de l’alimentation AC en le connectant à la prise.et d’autres éléments.

	<p>Arduino Reset (5) Vous pouvez réinitialiser votre carte Arduino, c'est-à-dire démarrer votre programme dès le début. Vous pouvez réinitialiser la carte UNO de deux façons. Tout d'abord, en utilisant le bouton de réinitialisation (16) sur la carte. Deuxièmement, vous pouvez connecter un bouton de réinitialisation externe à la broche Arduino identifiée RESET (5)</p>
	<p>Pin (3.3) (6) 3.3V – Sortie alimentation 3.3 volt</p>
	<p>Pin (5)</p> <ul style="list-style-type: none"> ▪ (7) 5V – Sortie alimentation 5 volt ▪ La plupart des composants utilisés avec la carte Arduino fonctionnent bien avec 3,3 volts et 5 volts.
	<p>Pin (GND) (8) GND (Ground) – Il existe plusieurs broches GND sur l'Arduino, dont l'une peut être utilisée pour relier votre circuit.</p>
	<p>Pin (Vin) (9) Vin – Cette broche peut également être utilisée pour alimenter la carte Arduino à partir d'une source d'alimentation externe, comme l'alimentation secteur</p>
	<p>Analog pins (10) La carte Arduino UNO comporte cinq broches d'entrée analogiques A0 à A5. Ces broches peuvent lire le signal d'un capteur analogique comme le capteur d'humidité ou le capteur de température et le convertir en une valeur numérique pouvant être lue par le microprocesseur.</p>
	<p>Pin ICSP (11) Principalement, ICSP (12) est un AVR, un petit en-tête de programmation pour Arduino composé de MOSI, MISO, SCK, RESET, VCC et GND. Il est souvent appelé un SPI (Serial Peripheral Interface), qui pourrait être considéré comme une « expansion » de la sortie. En fait, vous débrouillez le périphérique de sortie sur le maître du bus SPI.</p>
	<p>Indicateur d'alimentation LED (12) Cette LED s'allume lorsque vous branchez votre Arduino dans une source d'alimentation pour indiquer que votre carte est sous tension correctement. Si cette lumière ne s'allume pas, il y a quelque chose qui ne va pas avec la connexion.</p>
	<p>LEDs TX et RX (13) Sur votre carte, vous trouverez deux étiquettes : TX (transmission) et RX (réception). Ils apparaissent en deux endroits sur la carte Arduino. Tout d'abord, au niveau des broches numériques 0 et 1, pour indiquer les broches responsables de la communication série. Deuxièmement, la led TX et RX</p>

	(14). La led TX clignote avec une vitesse différente lors de l'envoi des données en série. La vitesse de clignotement dépend du débit en bauds utilisé par le tableau. RX clignote pendant la réception.
	E/S numérique (14) La carte Arduino UNO comporte 14 broches d'E/S numériques (14) (dont 6 fournissent une sortie PWM (Modulation de largeur d'impulsion). Ces broches peuvent être configurées pour fonctionner comme broches numériques d'entrée pour lire les valeurs logiques (0 ou 1) ou comme numérique Des broches de sortie pour piloter différents modules comme les LED, les relais, etc. Les broches étiquetées «~» peuvent être utilisées pour générer PWM.
	AREF (15) AREF signifie Référence analogique. Il est parfois utilisé pour régler une tension de référence externe (entre 0 et 5 Volts) comme limite supérieure pour les broches d'entrée analogiques.
	Arduino Reset (16) Vous pouvez réinitialiser votre carte Arduino, c'est-à-dire démarrer votre programme dès le début. Vous pouvez réinitialiser la carte UNO de deux façons. Tout d'abord, en utilisant le bouton de réinitialisation (16) sur la carte. Deuxièmement, vous pouvez connecter un bouton de réinitialisation externe à la broche Arduino identifiée RESET (5).
	Microcontrôleur (17) Chaque carte Arduino possède son propre microcontrôleur (17). Vous pouvez l'assumer comme cerveau de votre carte Arduino. Le CI(circuit intégré) principal sur l'Arduino est légèrement différent de la carte. Les microcontrôleurs sont habituellement de la société ATMEL. Vous devez savoir quel type du CI sur votre carte avant de charger un nouveau programme de l'IDE Arduino. Cette information est disponible en haut du CI. Pour plus de détails sur la construction et les fonctions CI, vous pouvez consulter la fiche technique.

II.3. Capteurs

II.3.1. Qu'est-ce qu'un capteur ?

Un capteur réalise l'acquisition d'une grandeur physique (température, luminosité, présence, distance...) qu'il transforme en un signal de sortie logique (grandeur électrique), analogique ou numérique afin qu'il puisse être traité par la partie commande du système (actionneurs) [7].

II.3.2. Capteurs utilisés

Dans notre réalisation de chaine d'embouteillage nous avons eu besoin que d'un seul capteur. Sa présentation, ses caractéristiques ainsi que son branchement (réel et en utilisant logiciel Proteus) avec l'Arduino UNO Rev3 ont détaillés ci-dessous.

II.3.2.1. Capteur IR de proximité FC-51

- **Présentation**

Le capteur de proximité FC-51, donné par Figure 5 est un capteur muni d'une paire de tubes d'émission et de réception IR [8]. Le tube de transmission envoi une centaine de fréquences Infrarouge qui sera réfléchi lors de la détection des obstacles (surface réfléchissante) sur une plage allant de 2cm à 30cm et sur un angle de 35° et reçue par le tube récepteur, comme le montre **Figure.II.5**. Après le traitement avec le circuit du comparateur, la lumière verte est allumée et l'interface de sortie du signal émet un signal numérique (signal à faible tension). La distance de détection peut être ajustée avec le potentiomètre.

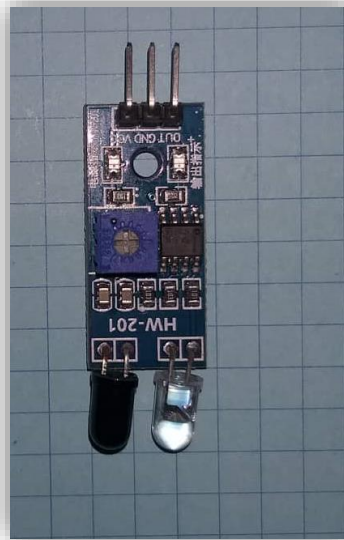


Figure II.5 : Photo réelle du capteur IR de proximité Fc-51.

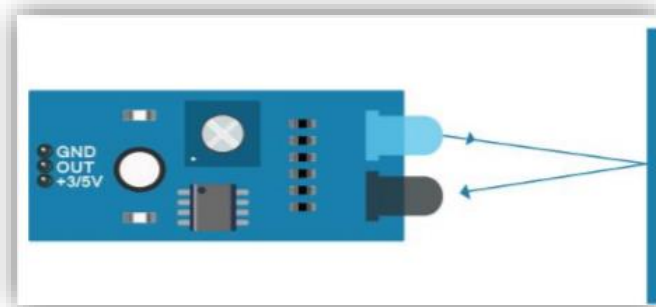


Figure II.6 : Fonctionnement du capteur IR de proximité Fc-51

- **Caractéristiques**

Tableau II.3 donne les principales caractéristiques du capteur IR de proximité FC-51

Tableau II.3 : Caractéristiques du Capteur IR de proximité FC-51

Tension de fonctionnement	3,3 ~ 5 V.
Plage de détection	2 ~ 30 cm.
Angle de détection	35°.
Câblage	VCC-VCC ; GND-GND ; OUT-IO
Dimensions	Dimensions
Dimensions Poids	Environ 3 g.

• Branchement

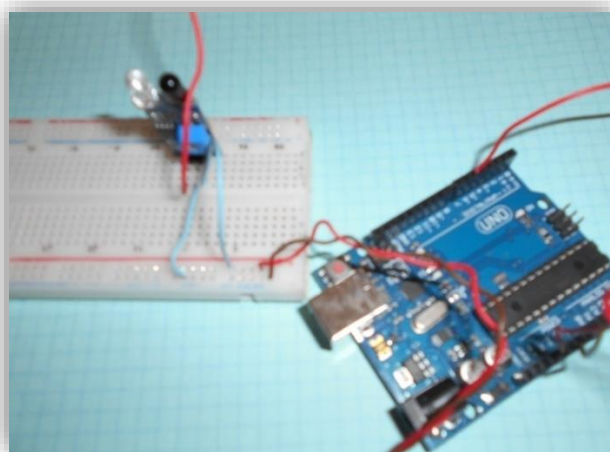


Figure II.7 : Branchement réel du capteur IR de proximité Fc-51.

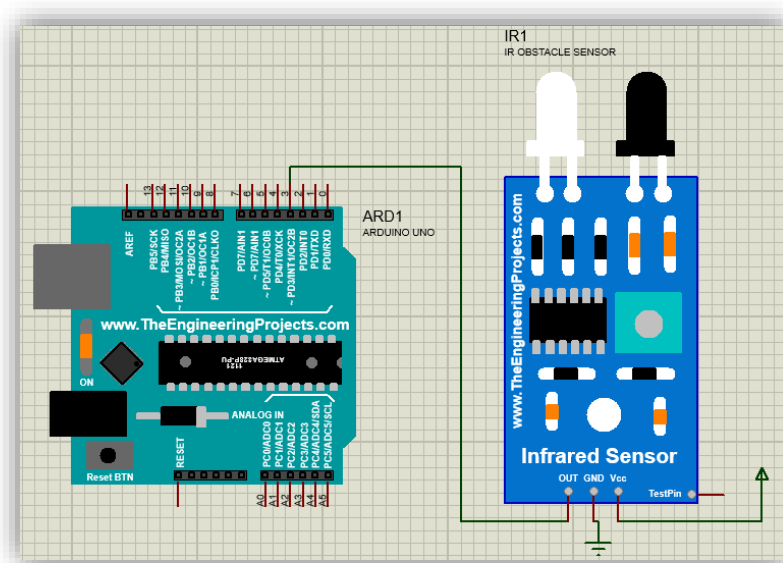


Figure II.8 : Branchement du capteur IR de proximité Fc-51 sous Proteus.

II.4. Actionneurs

II.4.2. Qu'est-ce qu'un actionneur ?

Dans une machine, un actionneur est un objet qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Les actionneurs servent à actionner et sont à l'origine d'une action du système domotique, automatique (un moteur pour fermer une porte, une lampe pour avertir, ...). Dans les définitions de l'automatisme, l'actionneur appartient à la partie opérative d'un système automatisé.

II.4.3. Actionneurs utilisés

Dans notre projet de ligne d'embouteillage nous avons utilisés plusieurs actionneurs. Leurs présentations, leurs caractéristiques ainsi que leurs branchements avec l'Arduino UNO Rev3 sont détaillées ci-dessous.

II.4.3.1. Module d'alimentation abaisseur DC-DC LM2596HVS et Power supply 12v 5A

- **Présentation**

Un convertisseur DC-DC est un dispositif électronique de puissance qui accepte une tension d'entrée continue et fournit également une tension de sortie continue. Le module abaisseur DC-DC LM2596 est un convertisseur qui donne en sortie une tension plus petite que celle de l'entrée. Ici, le régulateur de tension à découpage LM2596, illustré par Figure II.19, est utilisé pour réguler la tension. Ce régulateur est un circuit intégré monolithique (un seul bloc) parfaitement adapté à la conception simple et pratique d'un régulateur à découpage abaisseur (convertisseur abaisseur), son efficacité est nettement supérieure à celle des régulateurs linéaires à trois bornes populaires.



Figure II.9 : Photo réelle du Module d'alimentation abaisseur DC-DC LM2596HVS

- **Caractéristiques du Module d'alimentation abaisseur DC-DC LM2596HVS (tableau II.4)**

Tableau II.4 : Caractéristiques du Module d'alimentation abaisseur DC-DC LM2596HVS

Tension d'entrée	4,5V-48V
Tension de sortie	3V-40V
Courant d'entrée	3A(Maximum)
Efficacité de conversion	92% (la plus élevé)
Ondulation de sortie	-30mV
-30mV Fréquence de commutation	150 KHz
Dimensions	43*20.5*12.5mm
Poids	13g

❖ Remarque

Comme nous l'avons souligné lors de la partie présentation, le convertisseur abaisseur de tension réduit la tension d'entrée d'une alimentation afin que la tension de sortie soit dans les normes et n'endommage pas les modules utilisés. Nous avons donc utilisé une alimentation 12V/5A donné par la figure II.10

- **Caractéristiques du transformateur à découpage : Power supply 12v 5A**

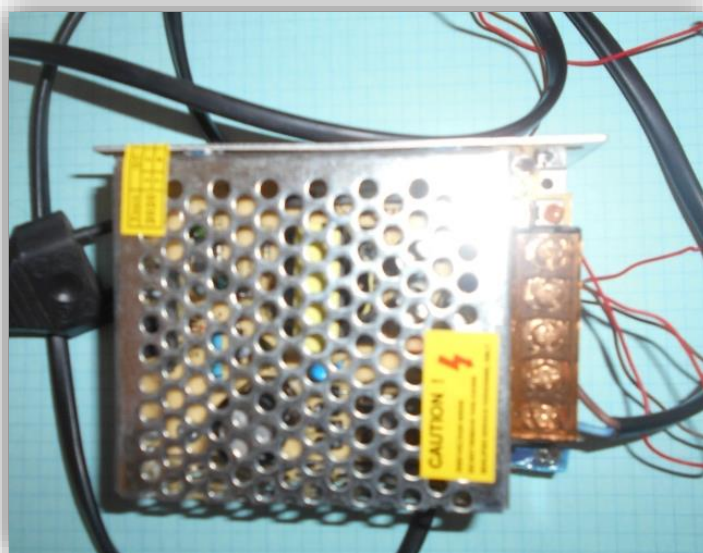


Figure II.10 : Photo réelles du transformateur à découpage (Power supply)

- **Caractéristique du transformateur à découpage Power supply (tableau II.5)**

Tableau II.5 : caractéristiques du transformateur à découpage Power supply

Tension d'entrée	AC 110 ~ 220V +/- 15%
Tension de sortie	DC 12V
Courant de sortie	5A
Puissance	60 W
Poids net	205g.
Dimension	11 x 7.8 x 3.5 cm
Matériel de Shell	métal cas / aluminium cas
Température de fonctionnement	0 – 40°C
Température se stockage	20 – 60°C
Humidité ambiante	0 – 95% non-condensation
Conformité de sécurité	FCC/CE/ROHS

- **Branchement**

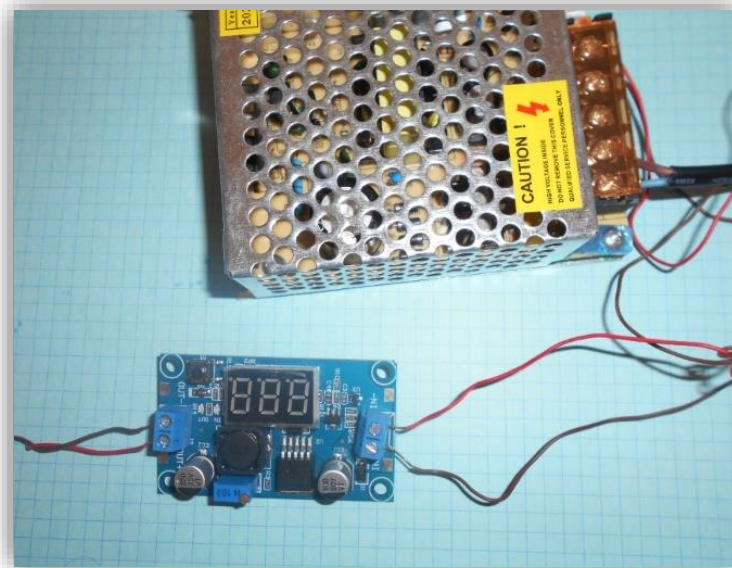


Figure II.11 : Branchement réelles de l'abaisseur et du transformateur

II.4.3.2. Moteur pas à pas NEMA17 et le module L298N :

Nous avons regroupé ces deux modules ensemble car le branchement du NEMA17 dépend de celui de l'interface driver L298N.

- **Présentation du moteur NEMA17 :**

Un moteur pas à pas est un moteur électrique à courant continu sans balai qui divise une rotation complète en un certain nombre de pas égaux. La position du moteur peut alors être

commandée pour se déplacer et se maintenir à l'une de ces étapes sans aucun capteur de rétroaction (un contrôleur en boucle ouverte), tant que le moteur est soigneusement dimensionné pour l'application. Il s'agit d'un moteur pas à pas biphasé ; qui se compose de deux bobines. Les pôles de ces bobines sont dans l'ordre (1A et 2A, 1B et 2B). Ces derniers fonctionnent avec un système binaire (0---0V et 1---12V) qui assure la rotation d'un pas à chaque décalage binaire d'une manière successive, à gauche ou à droite selon le sens horaire ou le sens contraire.



Figure II.12 : Photo réelles du moteur pas à pas NEMA17

- **Caractéristiques du moteur NEMA17 (tableau II.6)**

Tableau II.6 : Caractéristiques du Moteur pas à pas NEMA17

Phases	2
Angle de pas	1,8 ± 5% degré / phase
Tension	4,2 V CC
Courant	1,2 A / phase
Résistance	8,8 ± 10% Ω / phase
Inductance	6,0 mH ± 20% (1 kHz 1 V eff.)
Style de plomb	AWG26 UL1007
Couple de maintien	40 N.cm min
Couple de détente	2,6 N.cm max
Résistance d'isolement	100MΩ Min (DC 500V)
Classe d'isolation	B (130 ° C)
Couple de rotor	57 g.cm ²
Longueur du câble	70cm
Dimensions	39.00mm x 39.00mm x 36.00mm
Poids	160g

- **Le module L298N :**

Le module L298N est un Double Pont-H destiné au contrôle de moteur (H-Bridge Motor Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation. C'est un module extrêmement utile pour le contrôler de robots et ensembles mécanisés. Il peut contrôler deux moteurs courant continu ou un moteur pas-à-pas 4 fils 2 phases. Il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faible, idéal donc pour un microcontrôleur). Il peut piloter des charges inductives comme des relais, moteurs continus et moteurs pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retours. Il s'agit d'un module prêt à l'emploi.



Figure II.13 : Photo réelles du module L298N

- **Caractéristiques du module L298N (tableau II.7)**

Tableau II.7 : Caractéristiques du module L298N

Tension d'alimentation (partie puissance VMS)	5 à 35 V
Tension d'alimentation (partie logique VSS)	4.5 à 5.5 V
Courant crête en sortie	2A par pont
Intensité de courant dans la partie logique	0 à 36 mA
Signal de commande	Niveau haut : 4.5 à 5.5 V Niveau bas : 0 V
Puissance consommée maximale	20 W
Température de stockage	-25°C à + 130°C
Taille de la plaque	55mm x 60mm x 30mm
Poids	33 g

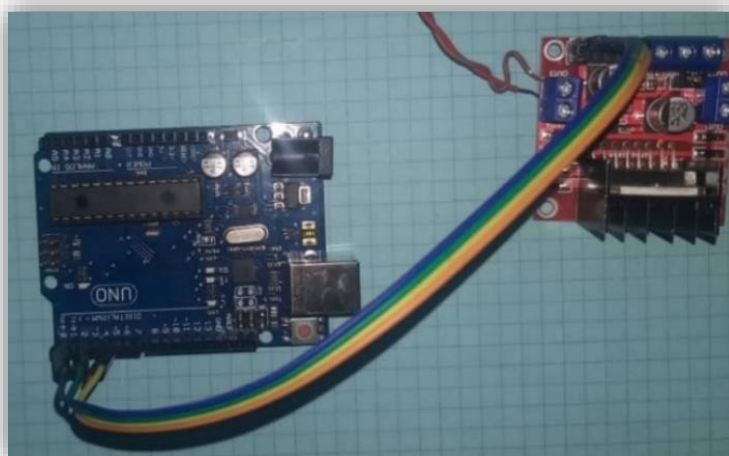


Figure II.14 : Branchement réel du module L298N

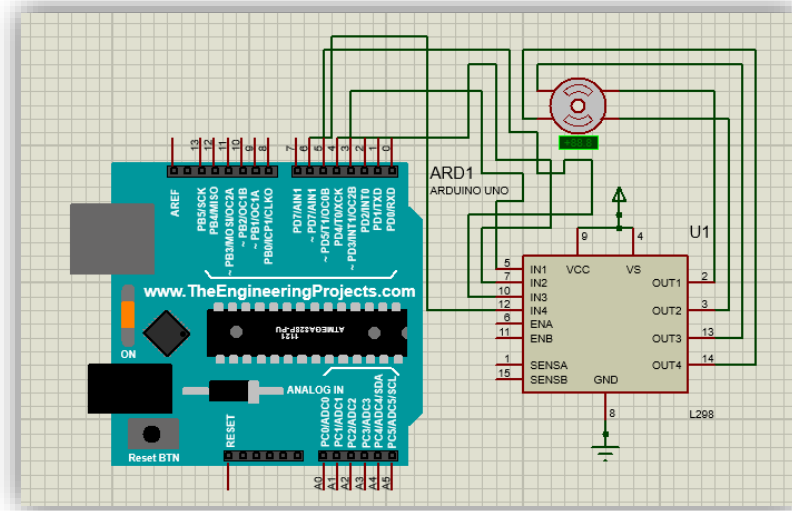


Figure II.15 : Branchement du moteur NEMA17 et le module L298N sous Proteus

II.4.2.3. Moteur 3 – 12V gear DC :

Ce moteur à engrenage en plastique durable (aussi connus sous le nom TT) est un moyen facile et peu coûteux de faire avancer notre projet. Il s'agit d'un moteur à boîte de vitesse à courant continu TT avec un rapport d'engrenage de 1.48, et il est livré avec 2 fils de 200 mm avec des connecteurs mâles de 0.1 adaptés aux planches à pain. Parfait pour le branchement sur une breadboard ou des borniers.

On peut alimenter ce moteur avec 3v DC jusqu'à 6v DC, il ira bien sur un peu plus vite aux tensions plus élevées.

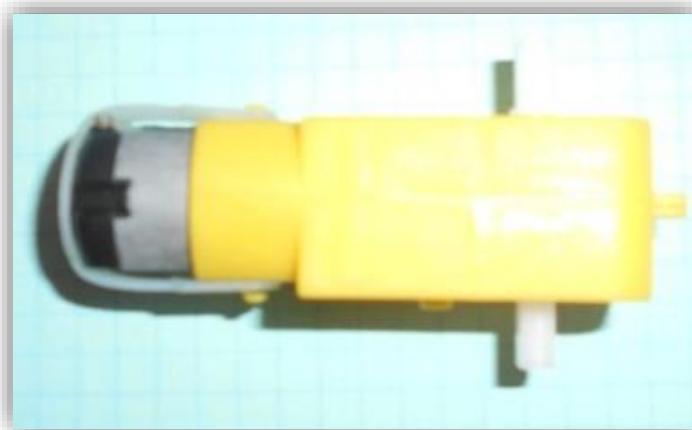


Figure II.16 : Photo réel du moteur DC 3-12v

- **Caractéristique du moteur DC 3-12v**

Tableau II.8 donne les principales caractéristiques du module DC 3-12v

Tableau II.8 : Caractéristique du moteur DC 3-12v

Tension	DC 3 – 6 v
Courant	100mA – 120mA
Taux de réduction	1.48
Poids du moteur	29 g / pièce
Taille de moteur	70 x 22 x 18 mm
RPM	100 – 240
Bruit	< 65 dB

- **Le L293D :**

Le composant L293D est un pont de puissance composé de plusieurs transistors et relais qui permet d'activer la rotation d'un moteur. Ce composant se trouve facilement sur EBay et ne coûte pas très cher.

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. Il est également possible de constituer deux pont-h afin de piloter deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre.



Figure II.17 : Photo réelle du module L293D

- **Caractéristique**

Tableau II.9 donne les principales caractéristiques du module L298N

Tableau II.9 : caractéristiques du module L298N

Caractéristique	Valeur
Nombre de pont-H	2
Courant Max (Régime continue)	600mA (x2)
Courant de pointe Max <2ms	1200mA
Vs Max Alim moteur	36 V
VSS Max Alim logique	7 V
Nombre de broche	16 DIP
Perte de tension	1.3 à 1.4 v (typical)

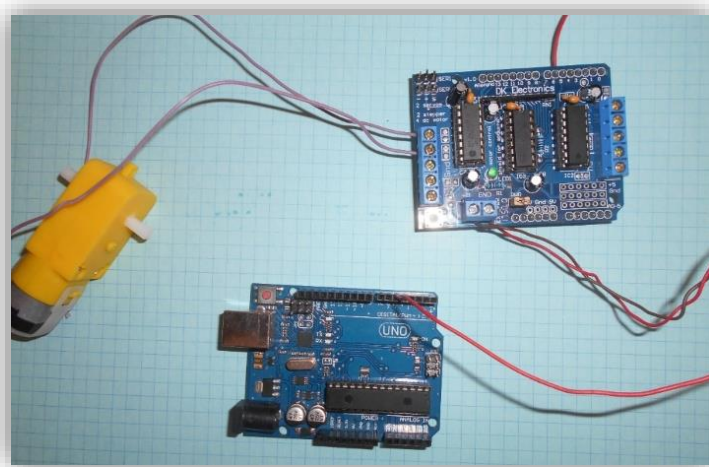


Figure II.18 : Branchement réel du moteur DC 3-12v et du module L293D

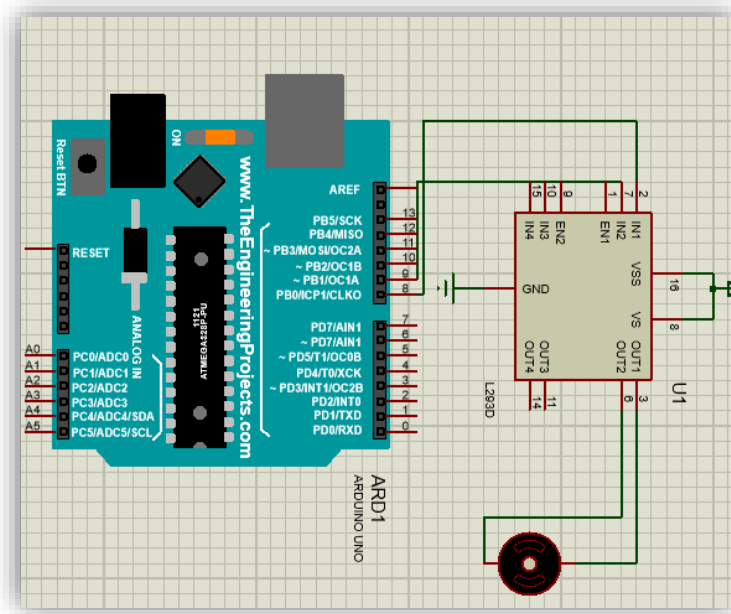


Figure II.19 : Branchement du moteur DC 3-12v et du module L293D sous Proteus

II.4.2.4. Afficheur LCD 16*2

- **Présentation**

LCD est l'abréviation du terme anglais "Liquide Crystal Display" qui signifie en français « Écran à cristaux liquides » d'où afficheur LCD. L'afficheur LCD 16*2 est un dispositif d'affichage de données d'un usage général dans le monde électronique de type alphanumérique. C'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 exploitées par l'utilisateur d'un système [9].

Les afficheurs à cristaux liquides utilisent la propriété de modulation de lumière des cristaux liquides. Les écrans à cristaux liquides sont composés de deux couches de polariseurs, avec des directions de polarisation perpendiculaire, prenant en sandwich deux plaques de verres entre lesquelles sont placés les cristaux liquides. Sur les plaques de verre se trouve une matrice d'électrodes pour chaque pixel. Une tension appliquée entre les électrodes d'un pixel entraîne un changement d'orientation des molécules et donc la transparence du pixel qui peut alors laisser, ou non, passer la lumière du rétro éclairage.

- Les pins de l'afficheur LCD 16*2 (voir Figure.II.20)



Figure II.20 : Photo réelle de l'afficheur LCD 16*2

- ✓ **Eclairage de l'écran** : l'Anode (A) est utilisée pour le rétro-éclairage '5V' et la Cathode (K) pour la masse 'GND'.
 - ✓ **L'alimentation** : VDD.
 - ✓ **La masse** : VSS.
 - ✓ **Lecture et écriture** : RW.
 - ✓ **Réglage du contraste** : le pin V0 est connecté à un potentiomètre qui assure le réglage de la différence d'intensité lumineuse (contraste).
 - ✓ **Registre commande ou donnée** : le pin RS.
 - ✓ **Entrée de validation** : le pin E.
 - ✓ **Bits de donnée** : D0, D1, D2, D3, D4, D5, D6 et D7.
 - ✓
- **Caractéristiques**

Tableau II.10 donne les principales caractéristiques de l'afficheur LCD 16*2

Tableau II.10 : Caractéristique de l'afficheur LCD 16*2

Tension de fonctionnement	5V
Possibilité d'affichage	2 lignes X 16 caractères
Type LCM	Caractères
Utilisation	Les photocopieurs, les télécopieurs, les imprimantes laser, les équipements de test industriel, les équipements de réseau tels que les routeurs et les dispositifs de stockage
Taille de la zone de visualisation	64.5mm x 16mm.
Dimensions	80mm x 36mm x 12mm

Poids	30g
-------	-----

- **Branchement**

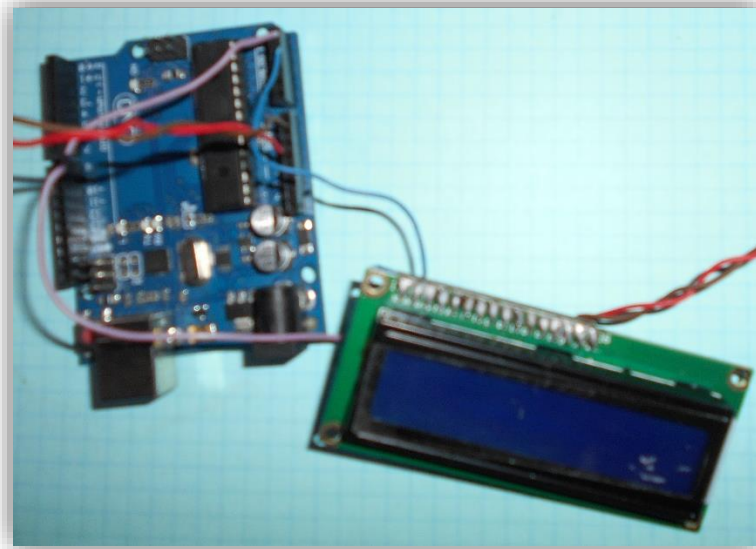


Figure II.21 : Branchement réel de l'afficheur LCD 16*2

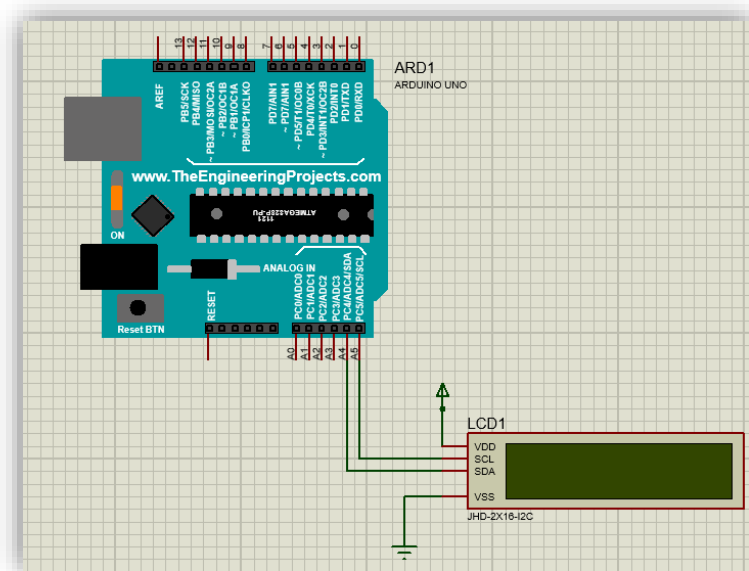


Figure II.22 : Branchement de l'afficheur LCD 16*2 sous Proteus

II.4.2.5. Bouton poussoir Schneider

- **Présentation**

Ce bouton poussoir vert Harmony XB4, de la marque Schneider Electric, est un dispositif servant à actionner un appareil électrique comme une sonnerie ou une sirène d'alarme. Il a pour rôle d'enclencher un mécanisme auquel il est associé. Grâce à son mécanisme à impulsion, il a la capacité de revenir automatiquement à sa position initiale après avoir été déclenché. Ce bouton poussoir rond, de couleur verte, s'installe facilement dans les boîtes d'encastrement de diamètre 22 mm.



Figure II.23 : Photo réelle du bouton poussoir Schneider

- **Caractéristiques :**

Tableau II.11 donne les principales caractéristiques du bouton poussoir Schneider

Tableau II.11 : Caractéristiques du bouton poussoir Schneider

Gamme de produits	Harmony XB4
Matériau de la collerette	Métal plaqué chrome
Matière de l'embase de fixation	Zamak
Diamètre de fixation	22 mm
Forme de la tête de l'unité de signalisation	Rond
Type d'unité de commande	Rappel à ressort
Description des contacts	1 "F"
Fonctionnement des contacts	À action dépendant
Mode de raccordement	Borniers à vis-étrier : $\leq 2 \times 1,5\text{mm}^2$ avec embout conformément à EN/IEC 60947-1 / Borniers à vis-étrier : $1 \times 0,22$ à $2 \times 2,5 \text{mm}^2$ sans embout conformément à EN/IEC 60947-1
Hauteur	47 mm
Largeur	30 mm
Profondeur	52 mm
Masse du produit	0,08 kg

- **Branchement**

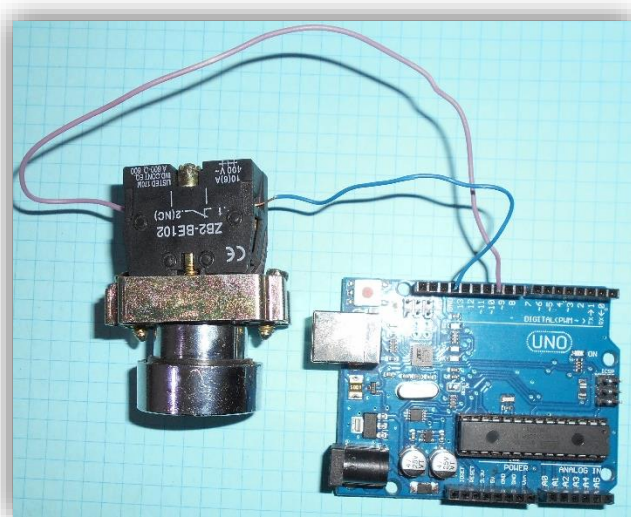


Figure II.24 : Branchement réel du bouton poussoir Schneider

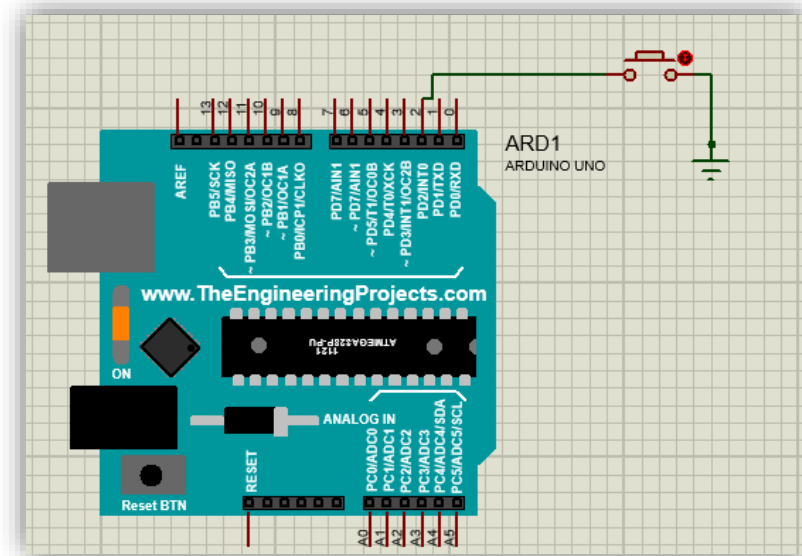


Figure II.25 : Branchement du bouton poussoir Schneider sous Proteus

II.4.2.6. Servomoteurs

- **Présentation**

Le servomoteur MG995, illustré par Figure II.15, est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure [10]. C'est donc un système asservi. Le servomoteur intègre dans un même boîtier, la mécanique et l'électronique, pour la commande et l'asservissement du moteur. La position du MG995 est définie en rotation continue de 360° et cela dans le sens horaire et le sens contraire.



Figure II.26 : Photo réelle du servomoteur MG995

- **Caractéristique**

Tableau II.12 donne les principales caractéristiques du servomoteur MG995

Tableau II.12 : Caractéristiques du servomoteur MG995

Type	Tower ProMG995
Dimensions	40 x 20 x 36,5mm
Poids	60g
Vitesse	0.16 s/60° sous 4.8V - 0.13 s/60° sous 6.0V
Couple de décrochage	11Kg/cm sous 4.8V - 13Kg/cm sous 6.0V
Tension d'alimentation	4.8V à 6V
Amplitude	de 0 à 180°

- **Branchement**

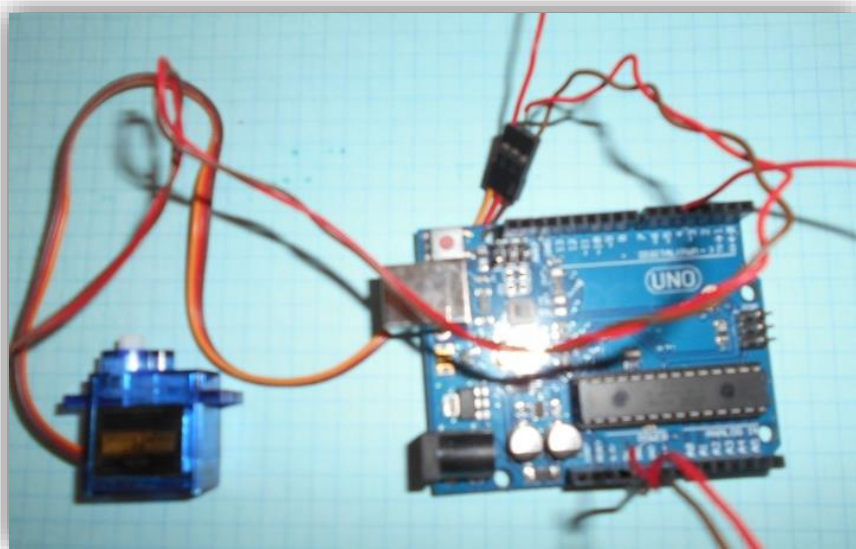


Figure II.27 : Branchement réel du servomoteur MG995

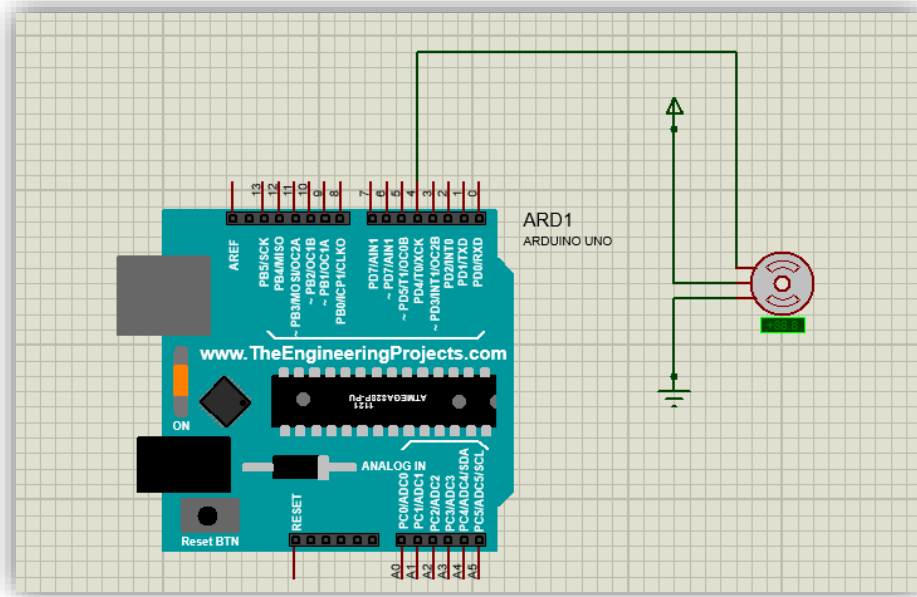


Figure II.28 : Branchement du servomoteur MG995 sous Proteus

II.4.2.7. Les relais

Comme son nom l'indique, il sert en tout premier lieu à « relayer », c'est-à-dire à faire une transition entre un courant faible et un courant fort. Mais il sert également à commander plusieurs organes simultanément grâce à ses multiples contacts synchronisés. Il permet également la transition entre deux sources différentes en isolant ces dernières. Il autorise des temporisations, des verrouillages, des impulsions.



Figure II.29 : Photo réelle du module 'relais'

- **Caractéristiques**

Tableau II.13 donne les principales caractéristiques du module relais

Tableau II.13 Caractéristique du module 'relais'

Courant de consommation	58,3 mA (12 V), 29,2 mA (24 V) : Courant consommé par la bobine donc courant de commande
Tension min. de commutation	3,75 V (5 Vcc.), 9 V (12 Vcc.) Tension à partir de laquelle le relais ferme ses contacts
Tension nominale	12 V ou 24 V (suivant les modèles)
Courant de commutation des contacts	6 A Courant pouvant être interrompu
Tension de commutation/tension max	250 V c.a./440 V cc
Pouvoir nominal de coupure	1500 VA
Rigidité diélectrique bobine-contacts	4000 V eff. Tension que l'on peut appliquer entre les contacts et la bobine sans risque d'arc électrique
Matériau de contacts	AgNi 0,15 plaqué or
Durée de vie mécanique	10 x 10 ⁶ man.

II.4.2.8. Pompe à eau 12v :

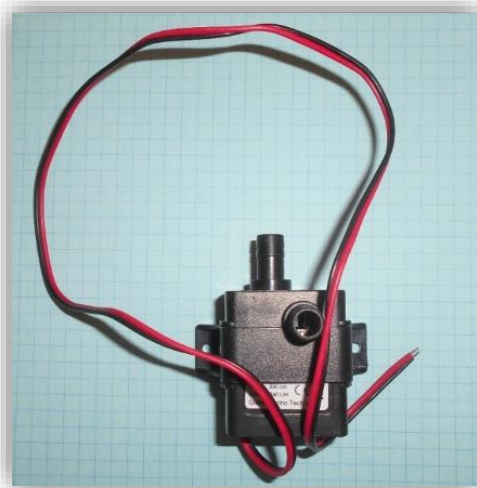


Figure II.30 : Photo réelle de la pompe à eau 12v

- **Caractéristique :**

Tableau II.14 donne les principales caractéristiques de la pompe d'eau 12V

Tableau II.14 : Caractéristique de la pompe d'eau 12v

Tension	CC 12V
Puissance	4,8W
Débit	240L / H
Ascenseur	Max 3 m
Poids	0,10 kg
Sortie d' eau	8,5 mm
Dimensions	37 x 55 x 27 mm
Longueur de câble	51,5 cm /

- **Branchement**

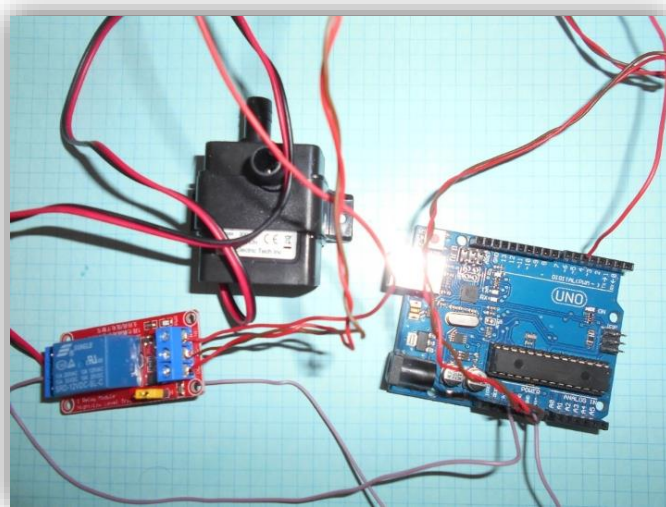


Figure II.31 : Branchement réel de la pompe à eau 12v et le module relais

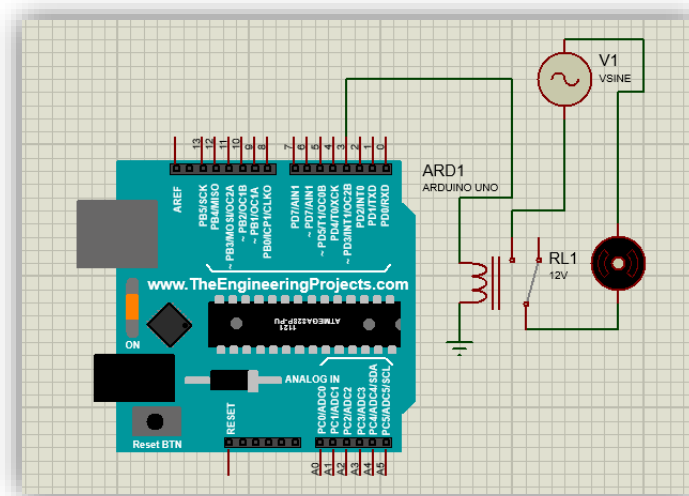


Figure II.32 : Branchement de la pompe à eau 12v et le module relais sous Proteus

II.5. Partie logicielle

II.5.1. Logiciel Arduino

II.5.1.1. Qu'est-ce que le logiciel Arduino ?

Les créateurs de l'Arduino ont développé un logiciel pour que la programmation des cartes Arduino soit visuelle, simple et complète à la fois. C'est ce que l'on appelle une IDE, qui signifie Integrated Développement Environnement ou Environnement de Développement Intégré en français (donc EDI) écrit en langage JAVA. L'IDE Arduino est le logiciel qui permet de programmer les cartes Arduino. L'IDE affiche une fenêtre graphique qui contient un éditeur de texte et tous les outils nécessaires à l'activité de programmation. Nous pouvons donc saisir notre programme, l'enregistrer, le compiler, le vérifier, le transférer sur une carte Arduino.



Figure II.33 : Logo du l'IDE de l'Arduino

II.5.1.2. Langage de programmation Arduino

Le langage Arduino est basé sur le C/C++. Le langage de programmation d'Arduino est en effet une implémentation de Wiring (une plate- forme open source similaire de physical computing qui proposait elle-même une bibliothèque appelée Wiring qui simplifie les opérations d'entrée/sortie). Un programme Arduino est aussi appelé un sketch.

II.5.1.3. Structure d'un programme Arduino

- Prise en compte des instructions de la partie déclarative
- Exécution de la partie configuration (**fonction setup()**),
- Exécution de la boucle sans fin (**fonction loop()**) : le code compris dans la boucle sans fin est exécuté indéfiniment.

II.5.1.3.1. Interface du logiciel

Les programmes Arduino sont structurés sous un schéma sketch qui contient deux blocs de fonctions obligatoires setup/loop.

La fonction « setup » est une fonction d'initialisation de la carte, exécuté une fois ; elle est utilisée au début du programme après les déclarations pour initialisation, démarrage des bibliothèques, modification du paramétrage des broches.

La fonction « loop » est une fonction principale exécutée indéfiniment en boucle.

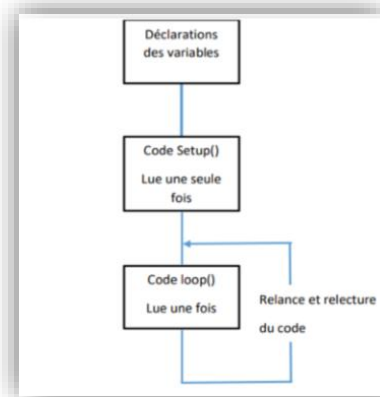


Figure II.34 : Les étapes d'un programme Arduino

II.5.1.3.2. Les étapes pour exécuter un programme dans IDE de l'Arduino

- Etape 1: Ouvrir le logiciel Arduino

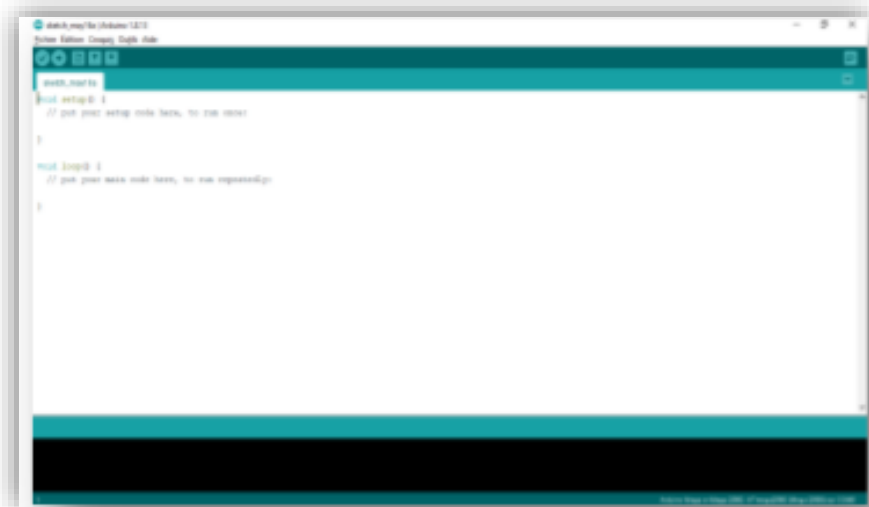


Figure II.35 : Fenêtre d'édition de programme

- Etape 2: Choisir le type de carte en sélectionnant Outils > Type de carte > Arduino UNO ou Arduino UNO Rev3.

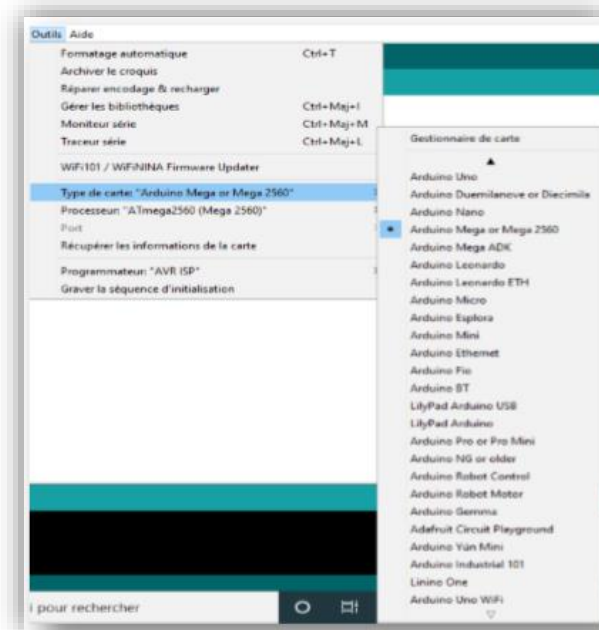


Figure II.36 : Type de la carte Arduino

- Etape 3: connecter la carte en faisant Outils > Port > COM

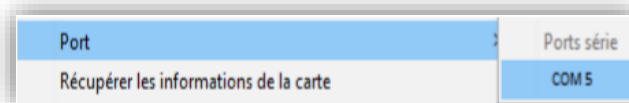


Figure II.37 : Choix du port

- Etape 4: Ecrire le programme, le sauvegarder et l'exécuter.



Figure II.38 : Barre d'outils Arduino

- Etape 5 : Téléverser le programme dans la carte Arduino via la connexion USB.



Figure II.39 : Photo réel du téléversement du programme vers la carte Arduino

II.5.1.3.3. Rôles des instructions de la barre de commande

Les différentes instructions de la barre de commande du logiciel Arduino sont expliquées par Figure.II.52.

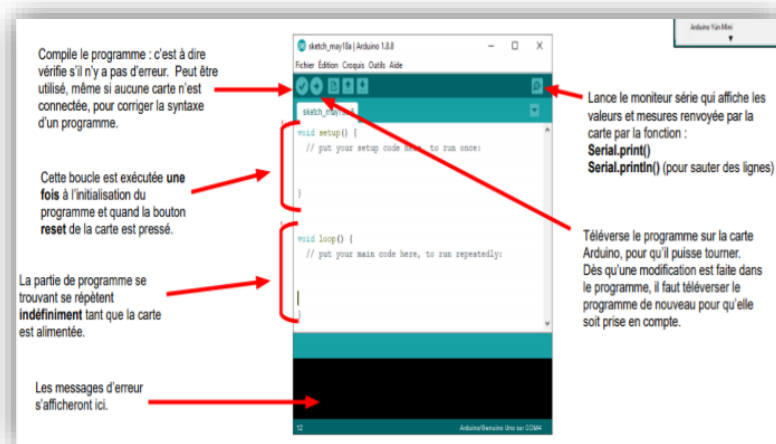


Figure II.40 : Fenêtre d'édition de programme

II.5.1.3.4. Les différentes règles à respecter pour une exécution correcte

- Le logiciel demande de sauvegarder le programme dans un dossier qui porte le même nom à la première compilation.
- Les majuscules et minuscules sont prises en compte dans les fonctions et les variables.
- La plupart des lignes se termine par un point-virgule ;
- Les blocs, boucles et fonctions sont encadrés par des accolades { }.
- Toutes les variables doivent être déclarées (nom et type).
- Les commentaires sont définis par // en début de ligne. Les blocs de commentaires par /* au début et */ à la fin.

II.5.2. Logiciel Proteus

II.5.2.1. Qu'est-ce que le logiciel Proteus ?

Le logiciel Proteus, est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics dont les logiciels incluent dans Proteus permettent la CAO dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : ISIS, ARES, que nous avons utilisés dans ce projet [11].

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation l'utilisent. Proteus possède plusieurs avantages parmi lesquelles, nous citons :

- Pack contenant des logiciels facile et rapide à comprendre et à utiliser.
- Le support technique est performant.
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet.

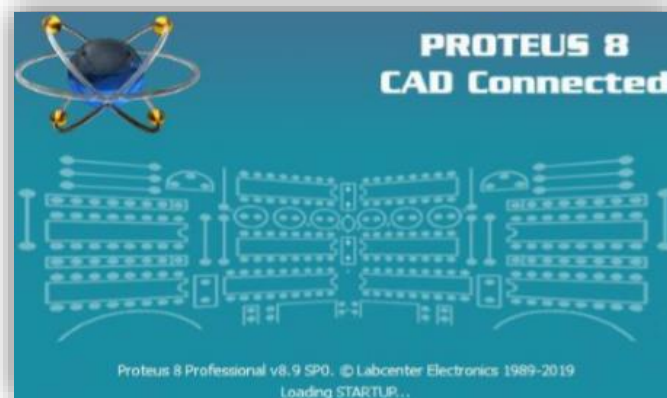


Figure II.41 : Logo de Proteus

II.5.2.2. Logiciel ISIS Proteus

Le logiciel ISIS de Proteus est principalement connue pour éditer des schémas électriques. Par ailleurs, ce logiciel permet également de simuler ces schémas. Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations, car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits.

II.6. Conclusion :

Dans ce chapitre nous avons fait une étude théorique de notre projet. Tout d'abord nous avons présenté la carte Arduino qui constitue le cœur de notre réalisation ainsi que les différents composants électroniques que nous avons utilisés. Ces composants sont ensuite classés en deux parties, capteurs et actionneurs, en expliquant leurs principes de fonctionnement et en donnant leur caractéristiques et le branchement avec la carte Arduino Uno.

Nous avons enfin, présenté les logiciels adoptés dans ce travail à savoir IDE Arduino, Proteus. Les informations acquises dans ce chapitre seront prises en compte dans la réalisation pratique de la ligne d'embouteillage qui sera expliquée dans le prochain chapitre.

CHAPITRE III :

CONCEPTION ET RÉALISATION PRATIQUE

III.1. Introduction

Après avoir évoqué les principales généralités sur les systèmes de remplissages et de bouchage de bouteilles dans le premier chapitre ainsi que les différents composants électroniques, les caractéristiques de base du microcontrôleur (Hardware) et les logiciels de programmation, simulation et schématisation (Software) dans le second chapitre, nous allons à présent détailler la partie la plus importante de notre travail qui est la fabrication concrète de notre système et qui représente le cœur de notre projet de fin d'étude. Dans ce chapitre nous allons nous focaliser essentiellement sur la partie théorique décrite précédemment pour tracer une feuille de route cohérente et claire du travail pratique qu'on va effectuer dans notre réalisation, pour cela nous devons procéder étape par étape comme suit :

- Etudier et analyser les besoins fonctionnels et non fonctionnels de notre système.
- Définir le principe de fonctionnement de notre machine et schématiser les différentes étapes en un diagramme d'activité.
- Expliquer la partie Software en décrivant notre programme Arduino Ide en langage humain.
- Simuler notre système à travers le logiciel Proteus en téléversant le programme sur la carte Arduino virtuelle.
- Brancher le système électronique et créer une carte pour cela.
- Réaliser la partie Hardware en fabriquant les différents mécanismes de notre système.
- Achever la réalisation en connectant la partie Hardware et la partie Software.

III.2. Etude des besoins du système

Dans cette partie nous allons éclaircir et clarifier l'ordre de priorité de notre système, et cela en divisant les besoins de ce dernier en deux parties.

Les besoins fonctionnels sont des étapes qui répondent aux exigences de notre cahier de charge et auxquelles ils assureront le bon fonctionnement de notre système. Quant aux besoins fonctionnels, ces derniers sont considérés comme des options supplémentaires qui n'influencent en aucun cas la sûreté de fonctionnement du système.

III.2.1. Besoins fonctionnels

Notre projet de remplissage et de bouchage de bouteilles doit répondre à 5 besoins fonctionnels majeurs, ces derniers doivent tous être exécutés dans un ordre de priorité bien défini et avec un processus similaire à chaque cycle afin d'assurer le bon fonctionnement de notre machine. Les axes importants pour cela sont résumés en :

Détection de la bouteille à l'aide d'un capteur infrarouge.

Remplissage La pompe à eau est l'élément principal à cette étape.

Bouchage Grâce à un mécanisme qu'on détaillera plus tard dans ce chapitre.

Ejection un autre mécanisme a été créé pour que cette étape se passe sans que le système rencontre des problèmes.

Initialisation du système Il est indispensable que le procédé redémarre à chaque fois du même endroit et de la même façon.

III.2.2. Besoins non fonctionnels

Après avoir définis les besoins fonctionnels de notre système en quelques actions majeures et indispensables à la fois, nous allons désormais présenter les besoins non fonctionnels qui n'influencent pas le bon déroulement des cycles de notre processus, mais qui agissent à la fois en matière de robustesse, design, rapidité ainsi que d'autres fonctionnalités qui restent des perspectives futures au développement de notre projet de fin d'étude :

Rapidité du processus en installant des moteurs plus performants ce qui augmentera la vitesse du convoyeur.

Capacité supérieure en augmentant la distance entre les deux cylindres ce qui conduira à un tapis plus long et donc une possibilité de contenir encore plus de bouteilles.

Diversification du volume des bouteilles en modifiant le système de telle sorte que le remplissage se fera en plusieurs volumes selon la taille de la bouteille.

Robustesse du système en utilisant des composants dont les caractéristiques sont les meilleurs sur le marché.

III.3. Principe de fonctionnement de notre machine de remplissage et de bouchage

Avant de commencer la fabrication de notre machine ainsi que la programmation et le branchement, nous devons établir toutes les étapes du processus pour bien comprendre le principe de fonctionnement de la machine afin de tracer une feuille de route qui nous facilitera la partie Software et la partie Hardware.

Pour cela, nous allons commencer par définir toutes les étapes nécessaires à la sûreté de fonctionnement de notre système, en détaillant méticuleusement les besoins fonctionnels qu'on a cités lors de la partie précédente, par la suite, on enchaînera par un diagramme d'activité, qui sera la portière à la partie qui suivra et qui sera dédié à la programmation du microcontrôleur et des autres composants.

III.3.1. Principales étapes du fonctionnement du système

Dans cette partie nous allons détailler toutes les étapes qui assureront à chaque cycle de s'accomplir dans les conditions optimales.

Les trois actions principales de notre système sont le remplissage de la bouteille, son bouchage et son éjection vers un réceptacle.

Chaque action comportera des étapes et des détails importants, certains d'eux effectués grâce à des mécanismes et des ajustements techniques, alors que d'autres seront complètement automatisés grâce aux différents capteurs et moteurs que nous avons établis dans notre cahier de charge.

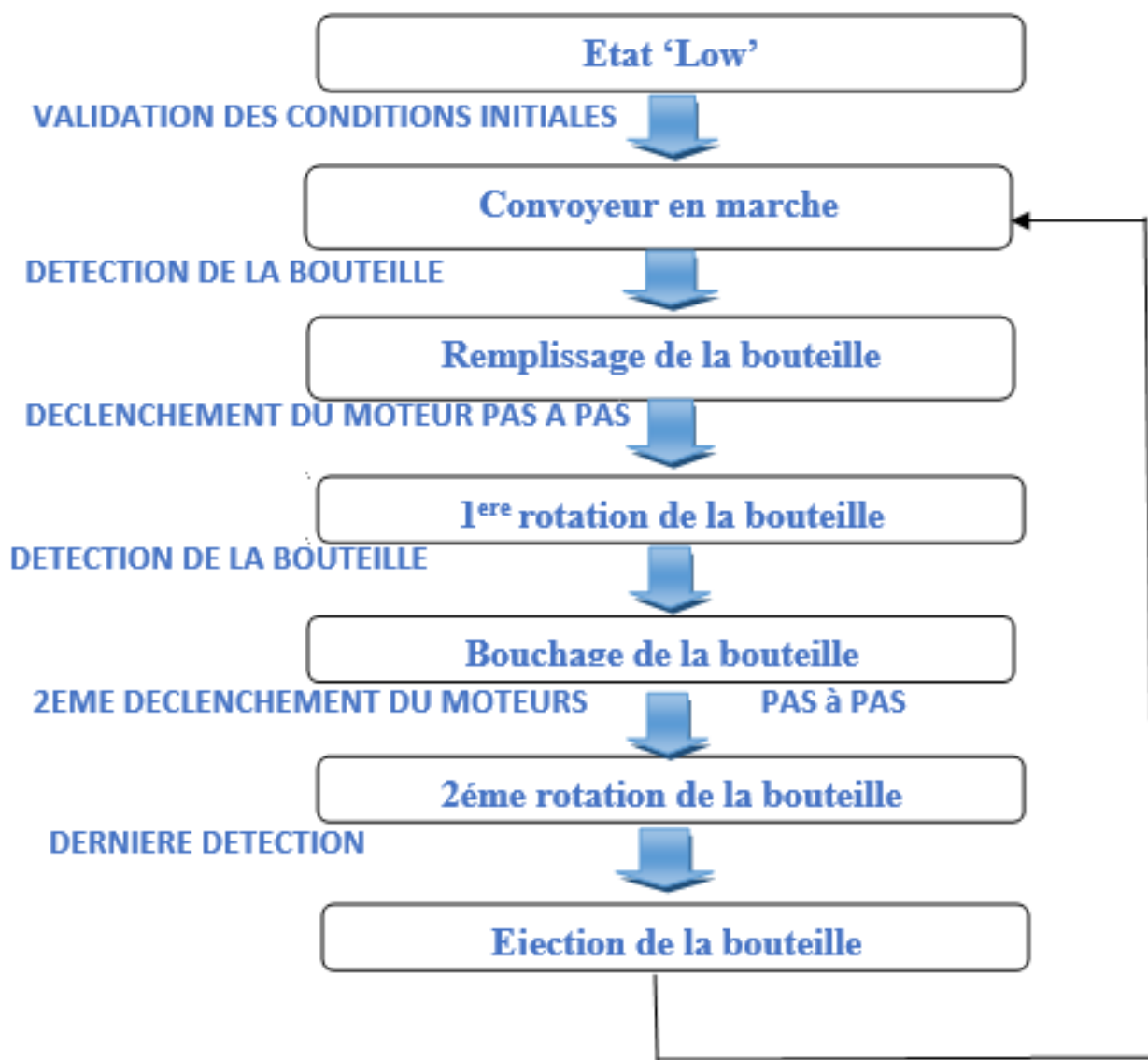
Nous résumerons ces étapes en plusieurs points :

- Dépôt des bouteilles sur le convoyeur.
- Allumage du système.
- Vérification des conditions initiales grâce aux capteurs.
- Déplacement des bouteilles linéairement après l'état en marche du convoyeur actionné par les moteurs.
- Détection de la bouteille au point A.
- Remplissage de la bouteille.
- Rotation de la bouteille grâce au mécanisme actionné par le moteur pas à pas.
- Détection de la bouteille au point B.
- Bouchage de la bouteille grâce au mécanisme des servo moteurs.
- Rotation de la bouteille.
- Détection de la bouteille au point C.
- Ejection de la bouteille vers le réceptacle.
- Initialisation du système.

III.3.2. Diagramme d'activité

Afin d'avoir une meilleure compréhension du principe de fonctionnement du système de remplissage et de bouchage, nous avons procédé à une schématisation de toutes les étapes que nous avons citées ci-dessus, le diagramme d'activité que nous allons présenter est une étape indispensable à la programmation du microcontrôleur et du système

Voici le diagramme d'activité de notre procédé :



III.4. Software

Après avoir détaillé le principe de fonctionnement de notre système en expliquant le procédé étape par étape tout en schématisant ceci en un diagramme d'activité. Nous allons à présent pouvoir entamer la partie Software de ce projet qui sera composée de deux phases, la première sera dédiée à la programmation, quant à la deuxième, elle concernera la simulation sur le logiciel 'Proteus'.

III.4.1. Programmation

Dans cette partie, nous allons exposer le programme qu'on a établi pour piloter notre carte Arduino et son microcontrôleur Atmega. Nous allons expliquer toutes les étapes du code en traduisant le langage C++ en langage humain afin de clarifier le fonctionnement de notre système.

La première étape à faire a été d'installer le logiciel Arduino, puis de l'exécuter et enfin procéder au démarrage de l'IDE, il est utile de noter qu'on a installé la version 1.8.18 de l'Arduino compatible avec les systèmes Windows, Linux et IOS.



Après le démarrage de l'Arduino IDE, il est indispensable de donner un nom au sketch utilisé et de l'enregistrer par la suite. Dans notre cas, nous avons sauvegardé le fichier sous le nom de 'Remplissage et bouchage de bouteilles'.



```

Remplissage_et_bouchage_de_bouteilles | Arduino 1.8.18
Fichier Édition Croquis Outils Aide

Remplissage_et_bouchage_de_bouteilles

void setup() {
  // put your setup code here, to run once:

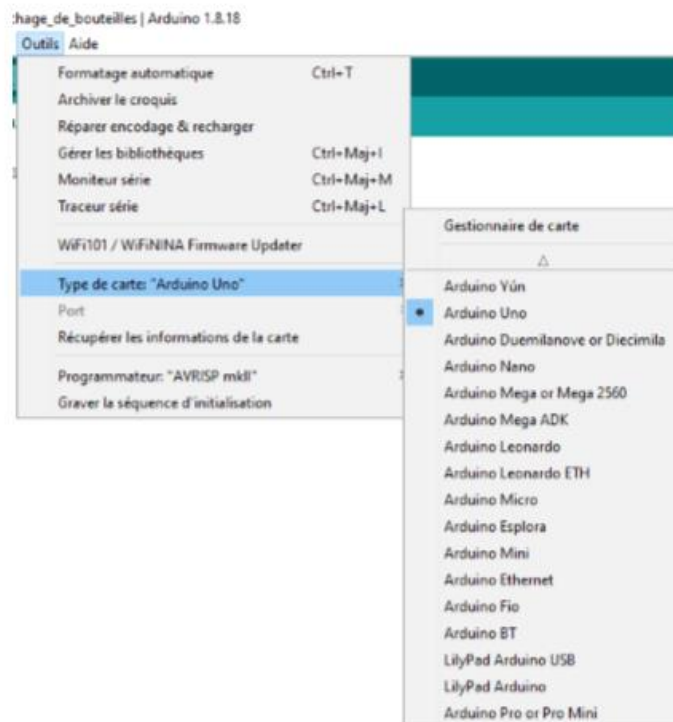
}

void loop() {
  // put your main code here, to run repeatedly:

}

```

La dernière phase à faire avant de commencer à créer le code, est de sélectionner le type de carte correspondant au microcontrôleur, nous avons donc choisis la carte Arduino Uno sur Outils -> Type de carte.

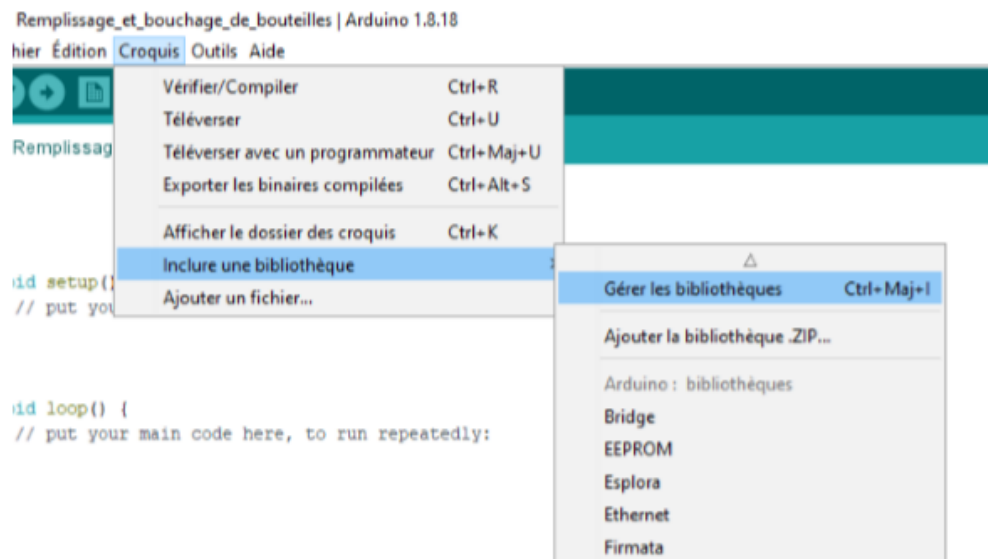


III.4.1.1. Initialisation

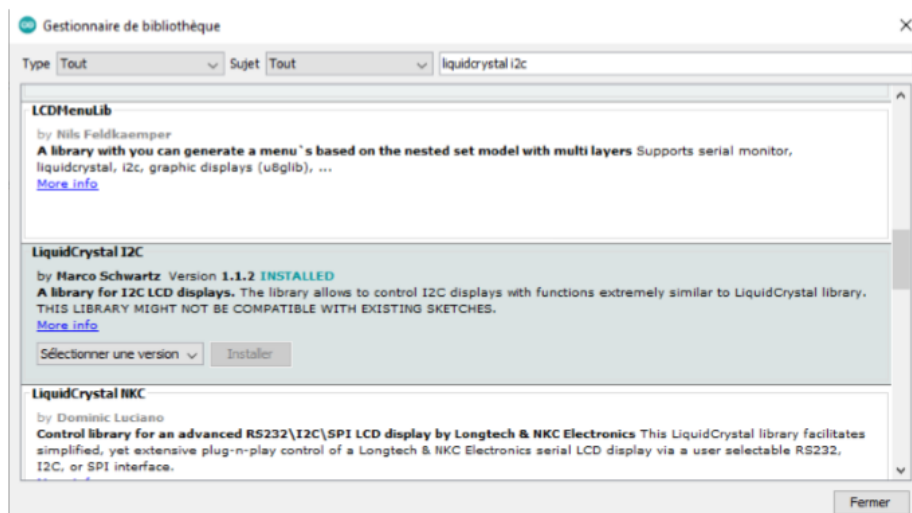
L'allumage du système via une alimentation de 12V 5A, provoque immédiatement l'initialisation de l'afficheur LCD 16*2. Ce dernier affiche un message qui assure le bon démarrage de la machine.

Nous avons donc pris le soin de programmer notre 'Display' afin d'afficher le message suivant 'Machine prête à être utilisée'.

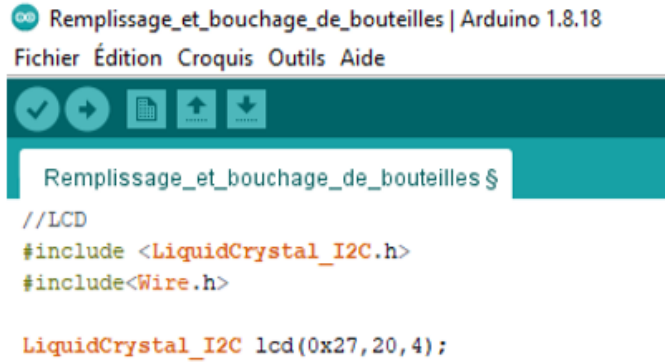
Pour cela nous avons tout d'abord commencé par installer la bibliothèque correspondante à notre afficheur LCD et son protocole I2C. Nous avons donc sélectionné Croquis->Inclure une bibliothèque->Gérer une bibliothèque.



Ensuite nous avons recherché la bibliothèque 'LiquidCrystal_I2C', que nous avons installé par la suite.



Après avoir installé la bibliothèque correspondante à notre afficheur, nous allons à présent déclarer l'afficheur et les différentes bibliothèques liées à son fonctionnement.



```
Remplissage_et_bouchage_de_bouteilles | Arduino 1.8.18
Fichier Édition Croquis Outils Aide
Remplissage_et_bouchage_de_bouteilles $
//LCD
#include <LiquidCrystal_I2C.h>
#include<Wire.h>

LiquidCrystal_I2C lcd(0x27,20,4);
```

Dans la partie ‘Setup’, il faut initialiser notre LCD.

```
void setup() {
  lcd.init()
}
```

Nous avons créé un sous-programme dans la partie ‘Loop’ pour afficher le message de démarrage ‘Machine prête à être utilisée’.

```
void loop() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("MACHINE PRETE A");
  lcd.setCursor(0, 1);
  lcd.print("ETRE UTILISEE");
}
```

III.4.1.2. Démarrage du système

Après avoir assuré que la machine est prête, nous allons à présent démarrer le système en appuyant sur un bouton qu'on nommera 'Start'.

Nous avons tout d'abord déclaré le bouton sur le Pin 2 de notre carte Arduino Uno.

```
int Start=2;
```

Le bouton est un composant dont les caractéristiques pilotent d'autres fonctionnalités, il est donc configuré comme étant une entrée 'INPUT' dans la partie 'Setup'.

```
pinMode (2, INPUT_PULLUP);
```

Dans la partie 'Loop' nous allons créer un sous-programme qui permettra au système de se déclencher chaque fois que le cycle du système est arrivé à sa fin. Il est important de préciser que si le bouton n'est pas actionné la machine reste hors-service. Dans le cas contraire, la boucle sera interrompue et le système passe aux sous-programmes qui suivent.

```
while(true)
{
  if(Start==LOW) {break;}
}
}
```

III.4.1.3. Convoyeur

Une fois que le bouton est activé, le convoyeur commence à marcher et ceci grâce aux quatre moteurs fixés aux extrémités du tapis roulant.

Les quatre moteurs dont l'alimentation est de 5V, sont déclarés respectivement sur les Pin 3, 4, 8 et 9 de notre carte et sont nommés 'Moteur 1', 'Moteur 2', 'Moteur 3' et 'Moteur 4' en référence à leur emplacement sur le côté gauche ou le côté droit.

```
int Moteur1=3;
int Moteur2=4;
int Moteur3=8;
int Moteur4=9;
```

Les moteurs sont des actionneurs souvent pilotés par des drivers, des transistors ou des relais (on a utilisé un circuit intégré L293D dans notre cas), ils sont donc configurés comme étant une sortie 'OUTPUT' dans la partie d'initialisation 'Setup'.

```
pinMode(3, OUTPUT);  
pinMode(4, OUTPUT);  
pinMode(8, OUTPUT);  
pinMode(9, OUTPUT);
```

Pour faire tourner le convoyeur et donc les bouteilles qui sont en dessus du tapis roulant, il faut activer nos quatre moteurs, pour cela on a créé ce mini sous-programme, dont le principe est simple mais efficace, puisqu'il permet aux moteurs de changer leurs états de 'Low' vers 'High' et dont la vitesse est constante, ce qui est favorable pour l'acheminement des bouteilles de manière sûre.

```
digitalWrite(Moteur1, HIGH);  
digitalWrite(Moteur2, HIGH);  
digitalWrite(Moteur3, HIGH);  
digitalWrite(Moteur4, HIGH);
```

Une fois que le bouton 'Start' est appuyé et que le convoyeur est déclenché, le message 'Machine en marche' est affiché sur notre LCD.

```
lcd.init();  
lcd.backlight();  
lcd.setCursor(0, 0);  
lcd.print("  MACHINE  ");  
lcd.setCursor(0, 1);  
lcd.print("  EN MARCHE  ");
```

III.4.1.4. Remplissage

Cette étape est la plus complexe de notre système, car le processus de remplissage exige une synchronisation parfaite entre les différents composants électroniques, les mécanismes du système ainsi que les blocs de programmation, qui assureront que chaque phase soit effectuée dans un laps de temps bien précis et dans un ordre chronologique défini.

En premier lieu, la bouteille vide arrive jusqu'au mécanisme de rotation, cette action est détectée par un capteur qu'on nommera 'Capteur A', ce dernier est déclaré sur le pin 5 de l'Arduino.

```
int CapteurA=5;
```

Les capteurs fournissent une tension proportionnelle à leurs mesures, ils sont donc configurés comme étant des entrées dans la partie 'Setup'.


```
pinMode(5, INPUT);
```

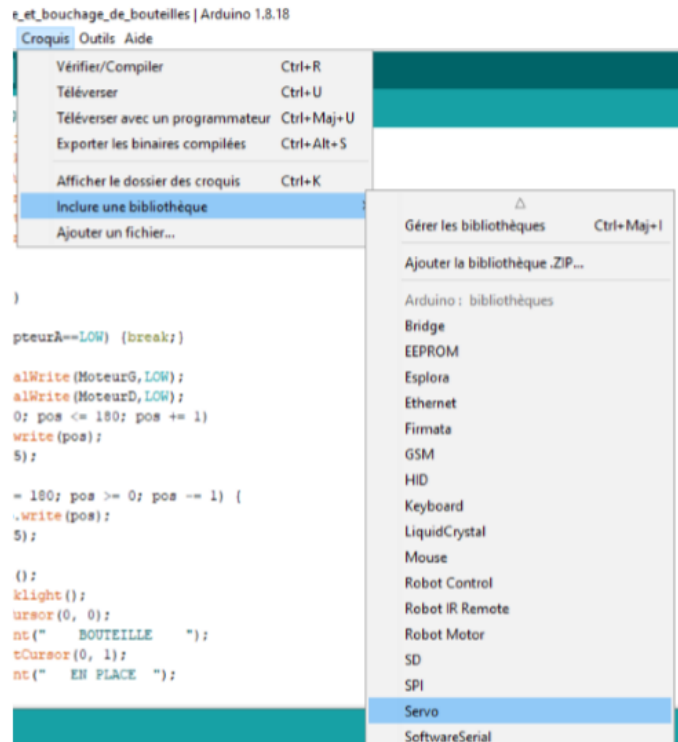

La détection de la bouteille par le capteur A dans le mécanisme de rotation, provoque immédiatement deux actions simultanées, la première consiste à l'arrêt du convoyeur et donc au changement d'états des quatre moteurs. Quant à la deuxième action, elle assure la fixation de la bouteille grâce à une barrière actionnée par un servomoteur. Le message 'Bouteille en place' est affiché sur le LCD une fois que la bouteille est détectée. le sous-programme suivant (dans la partie 'Loop') a été modélisé afin d'assurer le que toutes les étapes citées soient réalisées méticuleusement.

```

while(true)
{
  if(capteurA==LOW) {break;}
}
digitalWrite (Moteur1,LOW);
digitalWrite (Moteur2,LOW);
digitalWrite (Moteur3,LOW);
digitalWrite (Moteur4,LOW);
for (pos = 0; pos <= 180; pos += 1)
{ servoA.write(pos);
  delay(15);
}
for (pos = 180; pos >= 0; pos -= 1) {
  servoA.write(pos);
  delay(15);
}
lcd.init();
  lcd.backlight();
lcd.setCursor(0, 0);
  lcd.print("  BOUTEILLE  ");
  lcd.setCursor(0, 1);
  lcd.print("  EN PLACE  ");

```

Il est utile de signaler que les servomoteurs contiennent des bibliothèques prédéfinies sur l'IDE de l'Arduino, tout comme le LCD, on peut facilement les déclarer en les recherchant sur  Groquis. Inclure une bibliothèque Servo.



Sur la partie ‘Déclaration’ la bibliothèque du servomoteur est inclus automatiquement, ce dernier sera nommé ‘servoA’ sur la partie ‘Setup’ et sera configuré sur le pin numéro 6 de notre carte Arduino Uno.

```

#include <Servo.h>

Servo servoA=6;

```

Après que la bouteille soit bien fixée sur le système de rotation, on déclenche la pompe à eau via un relais. Le relais sera configuré comme sortie et sera déclaré sur le pin 7 de l’Arduino sous le nom de ‘Pompe’

```

int Pompe=7;

pinMode(7,OUTPUT);

```

L’action d’ouvrir le relais permet d’actionner la pompe à eau pendant un laps de temps bien calculé, et donc le remplissage de la bouteille au niveau souhaité, le message ‘Remplissage en cours’ sera étalé sur l’afficheur 16*2. Après cette action, l’étape de remplissage sera bouclée.

```
digitalWrite (Pompe, HIGH);
delay (753);

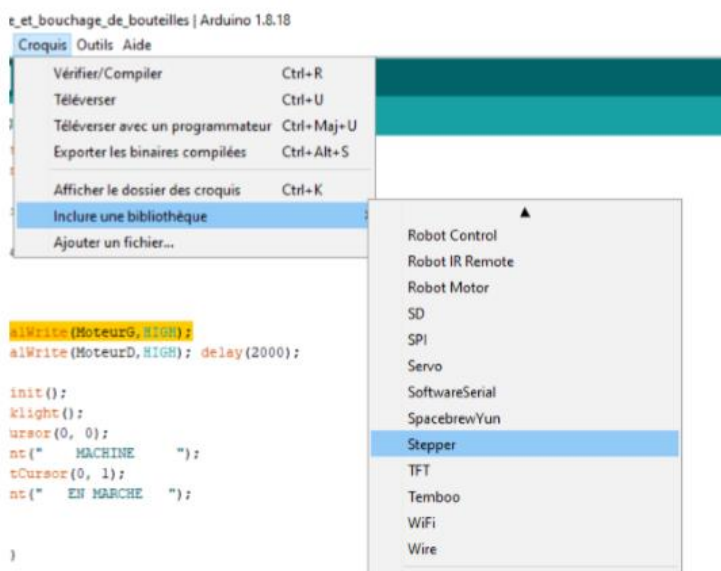
lcd.init ();
lcd.backlight ();
lcd.setCursor (0, 0);
lcd.print ("    REPLISSAGE    ");
    lcd.setCursor (0, 1);
    lcd.print ("    EN COURS    ");
```

La dernière action de cette étape de remplissage est d'afficher un message sur le LCD indiquant le passage à l'étape qui suit, le message correspondant est 'Bouteille remplie avec succès'.

```
lcd.init ();
lcd.backlight ();
lcd.setCursor (0, 0);
lcd.print ("BOUTEILLE REMLIE");
    lcd.setCursor (0, 1);
    lcd.print ("    AVEC SUCCES    ");
```

III.4.1.5. Bouchage

Une fois la bouteille remplie, on passe à l'étape suivante qui consiste à la boucher. Pour cela, il faut en premier lieu faire tourner le mécanisme de rotation grâce au moteur Nema 17 via le driver L298N. Ce dernier est déclaré par une bibliothèque nommée 'Stepper' qu'on peut trouver dans le volet 'Inclure une bibliothèque' de la même manière que l'afficheur LCD et le servomoteur.



Le moteur pas à pas va occuper deux pins de l'Arduino, le pin numéro 10 sera déclaré pour commander la direction 'DIR', quant au pin 11, il sera configuré pour gérer le nombre de pas 'STEP'.

```
Stepper stepper(STEPS, 10, 11);
```

```
int STEP=10;
int DIR=11;
```

```
pinMode(10, OUTPUT);
pinMode(11, OUTPUT);
```

Le sous-programme suivant permet de piloter le Nema 17 et de faire tourner le mécanisme de rotation avec grande précision, et ceci en changeant le nombre de pas de chaque intervalle. Le moteur Nema 17 tournera et engendrera la rotation de la bouteille. Lorsque le capteur détectera la présence de cette dernière, le moteur pas à pas s'arrêtera de tourner et la bouteille sera donc prête à être boucher. On nommera ce capteur 'CapteurB'.

```
while(digitalRead(CapteurB)==HIGH);
digitalWrite(STEP, HIGH);
digitalWrite(DIR, HIGH);

if(digitalRead(CapteurB)==LOW);
break;
}
```

Le capteur 'CapteurB' est déclaré dans la partie 'Setup' sur le Pin numéro 12 de la carte Arduino. Pour la même raison que celle du 'CapteurA', il est configuré comme étant une entrée.

```
int CapteurB=12;
```

```
pinMode(12, INPUT);
```

Un message est affiché pour nous rappeler de poser un bouchon manuellement sur le dessus de la bouteille, avec un laps de temps de 5 secondes pour nous permettre de placer le bouchon.

```
lcd.init();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print(" PLACEZ ");
lcd.setCursor(0, 1);
lcd.print(" LE BOUCHON");

delay(5000);
```

À ce stade de cette étape, le message 'Bouchage en cours' est affiché sur le LCD.

```

    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("    BOUCHAGE    ");
    lcd.setCursor(0, 1);
    lcd.print("    EN COURS    ");

```

Une fois la bouteille détectée par le 'CapteurB' le processus de bouchage est déclenché grâce à un jeu de servomoteur qui actionne un bras verticalement afin de forcer la fermeture du bouchon sur la bouteille, on nommera le servomoteur 'ServoB'.

```

        for (pos = 0; pos <= 180; pos += 1)
    { ServoB.write(pos);
      delay(15);
    }
    for (pos = 180; pos >= 0; pos -= 1) {
      ServoB.write(pos);
      delay(15);
    }

```

Il est utile de souligner que le servomoteur 'ServoB' est déclaré sur le pin 13 de notre microcontrôleur

```
Servo ServoB=13;
```

Comme lors de la précédente étape, il est utile d'écrire un message qui indique à l'utilisateur le passage à l'étape suivante. Le message qu'on a programmé est 'Bouteille bouchée avec succès'.

```

    lcd.init();
    lcd.backlight();
    lcd.setCursor(0, 0);
    lcd.print("BOUTEILLE REMPLIE");
    lcd.setCursor(0, 1);
    lcd.print("    AVEC SUCCES    ");

```

III.4.1.6. Ejection

Maintenant que la bouteille est remplie et bouchée, il nous reste plus qu'à l'éjecter dans un réceptacle hors du mécanisme de rotation, cela représentera la dernière étape de ce procès et marquera la fin du cycle.

Pour éjecter la bouteille, il faut tout d'abord que le mécanisme tourne via le moteur Nema 17 d'un certain nombre de pas, ce dernier s'arrête lorsque la bouteille est détectée par le dernier capteur du process qu'on nommera 'CapteurC'.

```
while (digitalRead(CapteurC)==HIGH);

digitalWrite( STEP,HIGH);
digitalWrite( DIR,HIGH);

if (digitalRead(CapteurC)==LOW);

break;

}
```

Le capteur 'CapteurC' est déclaré dans la partie 'Setup' sur le Pin numéro A0 de la carte Arduino. Comme nous l'avons déjà expliqué pour les capteurs précédents, il est configuré comme étant une entrée.

```
int CapteurC=A0;

pinMode (A0, INPUT);
```

Une fois la bouteille détectée par le 'CapteurC', le servomoteur 'ServoA' revient à sa position initiale, la bouteille n'est donc plus fixée.

```
for (pos = 180; pos >= 0; pos -= 1) {
  servoA.write(pos);
  delay(15);
}
```

Un autre servomoteur qu'on nommera 'ServoC' actionnera un bras qui fera tomber la bouteille dans le réceptacle. On fera afficher le message 'Bouteille éjectée avec succès' dès que le 'ServoC' est déclenché.

```

for (pos = 0; pos <= 180; pos += 1) {
  // in steps of 1 degree
  ServoC.write(pos);
  delay(1000);
}
for (pos = 180; pos >= 0; pos -= 1) {
  ServoC.write(pos);
  delay(1000);
}
  for (pos = 0; pos <= 180; pos += 1) {
    // in steps of 1 degree
    ServoC.write(pos);
    delay(1000);
  }
for (pos = 180; pos >= 0; pos -= 1) {
  ServoC.write(pos);
  delay(1000);
}

lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("BOUTEILLE EJECTE");
  lcd.setCursor(0, 1);
  lcd.print("AVEC SUCCES");

```

Le servomoteur ‘ServoB’ est déclaré sur le pin A1 de notre carte Arduino Uno via la bibliothèque « Servo ».

```
Servo ServoC=A1;
```

L’éjection de la bouteille marque la fin du procès de remplissage, bouchage et éjection d’une bouteille via l’Atmega.

On affichera un dernier message qui indiquera que toutes les étapes ont été réalisées. Le message est ‘FIN DU CYCLE’.

```

lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("FIN DU CYCLE");
  lcd.setCursor(0, 1);
  lcd.print("MERCI!!!");

}

```

III.4.2. Simulation

Après avoir achevé la partie programmation de la machine dans la partie précédente. Nous allons à présent évoquer la toute dernière phase de l'étape Software.

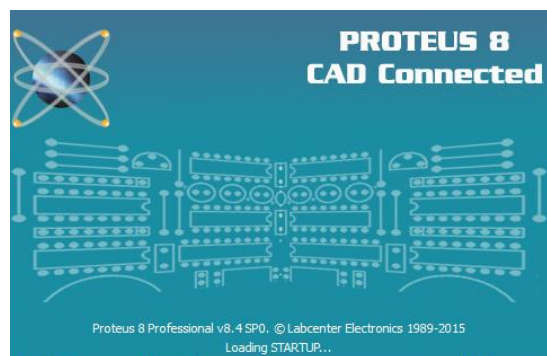
Dans cette partie, nous allons visualiser le fonctionnement de notre microcontrôleur, ainsi que les autres composants, et ceci grâce à une simulation qui nous permettra de tester notre Arduino Uno et les différents cycles de notre système.

Il est utile de préciser qu'il est indispensable de simuler et de visualiser un système dans n'importe quel projet. Si ce dernier assure un bon fonctionnement et que le programme ne représente aucune défaillance, nous pouvons dans ce cas attaquer la partie Hardware qui à brancher et à réaliser la machine.

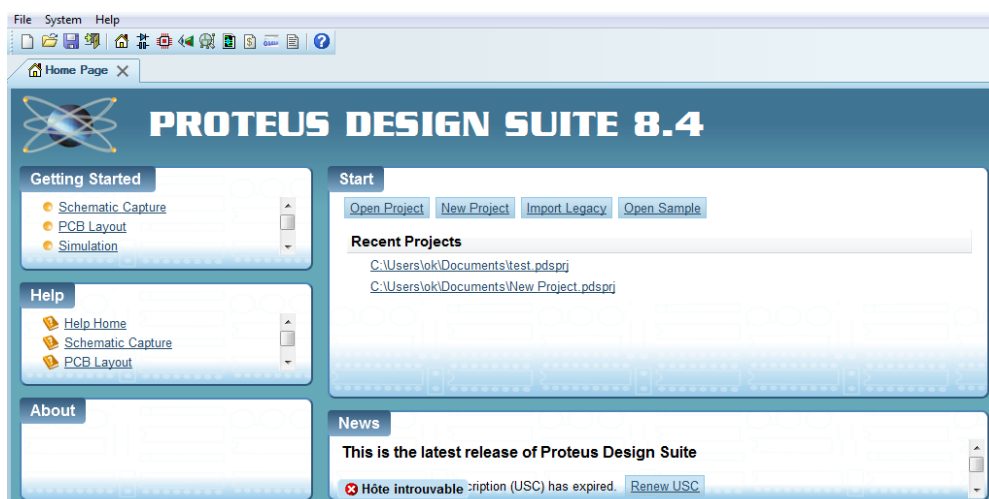
III.4.2.1. Lancement du logiciel

Afin de visualiser notre système, nous allons utiliser le logiciel Proteus, ce dernier est connu pour sa large bibliothèque et sa compatibilité avec les cartes Arduino dont la carte Arduino Uno.

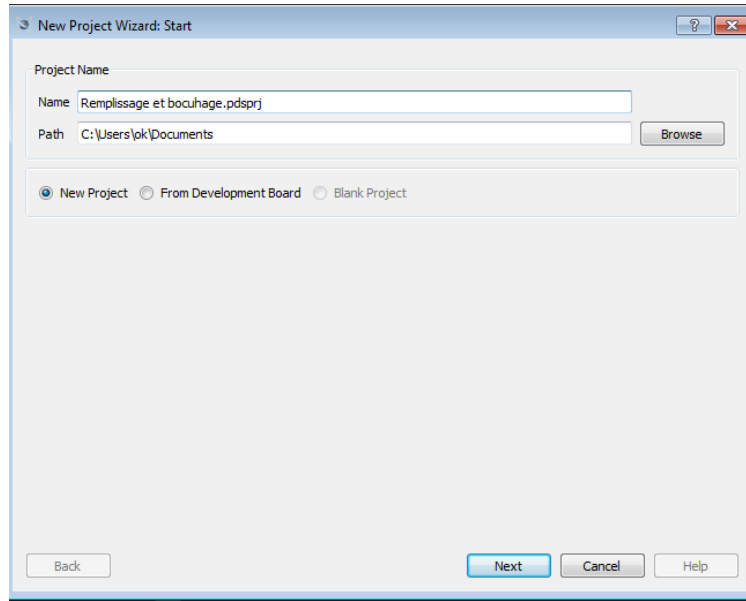
Pour cela, nous avons installé la version 8.4 du logiciel Proteus. Ensuite, nous avons exécuté le logiciel et lancé le programme comme le montre la figure suivante.



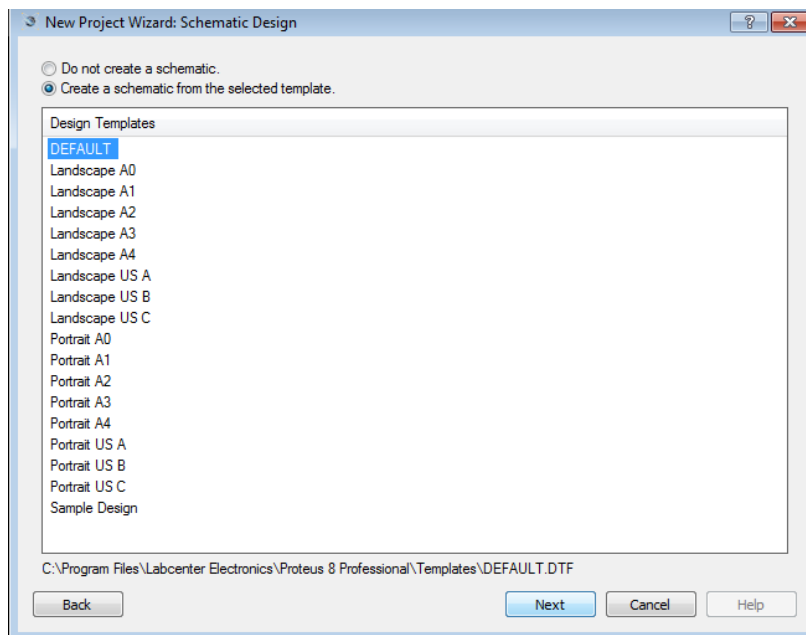
Après avoir lancé le logiciel nous devons ouvrir un nouveau projet en appuyant sur 'New Project'

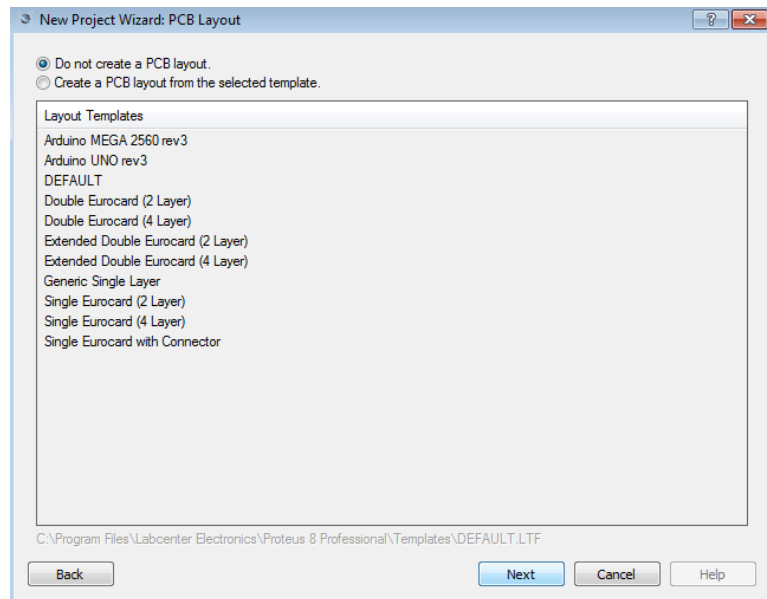


Nous devons ensuite sauvegarder un nom au projet, nous l'enregistrerons donc sous 'Remplissage et bouchage'.

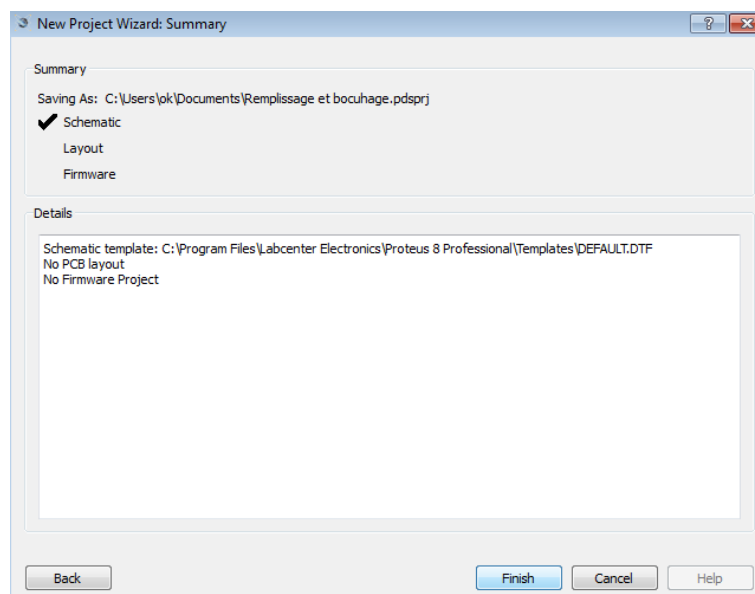


Nous enchaînerons par choisir le mode de schématisation standardisé par 'Default', et nous ajoutons par la suite les cartes Arduino en mentionnant la possibilité de créer des pistes en sélectionnant 'Do not create a pcb layout'. Les deux figures ci-dessous montrent cela.





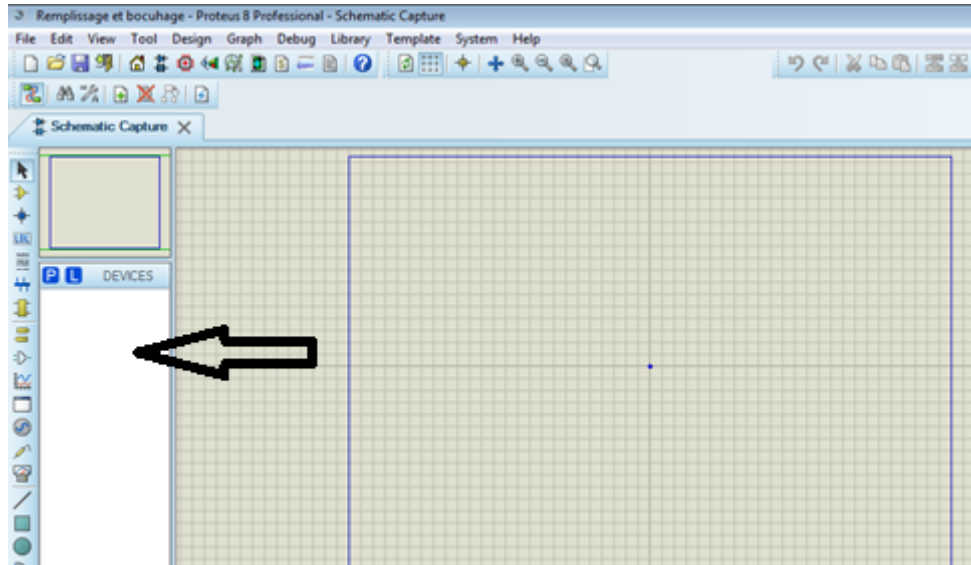
Enfin, nous devons conclure l'ouverture du nouveau projet 'Remplissage et bouchage' en appuyant sur le bouton 'Finish'.



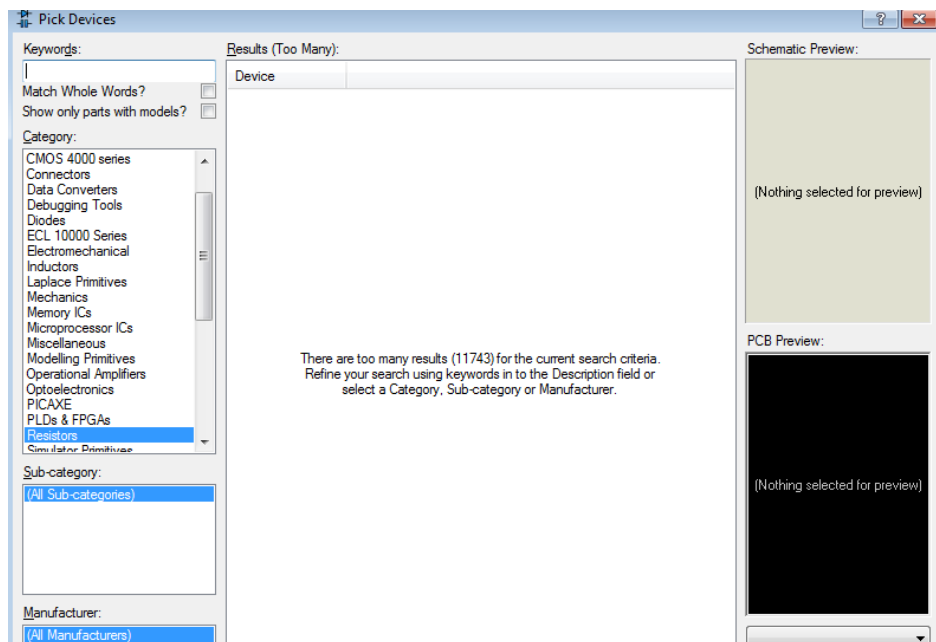
III.4.2.2. Sélection et branchement des composants

Après avoir ouvert la fenêtre 'Schematic Capture' qui correspond à 'ISIS Proteus', nous devons à présent sélectionner tous les composants qui forment le système et que nous avons cités en détails dans le deuxième chapitre.

Pour cela, il suffit d'effectuer un double clic sur le volet 'Device'. Comme le montre la figure suivante.

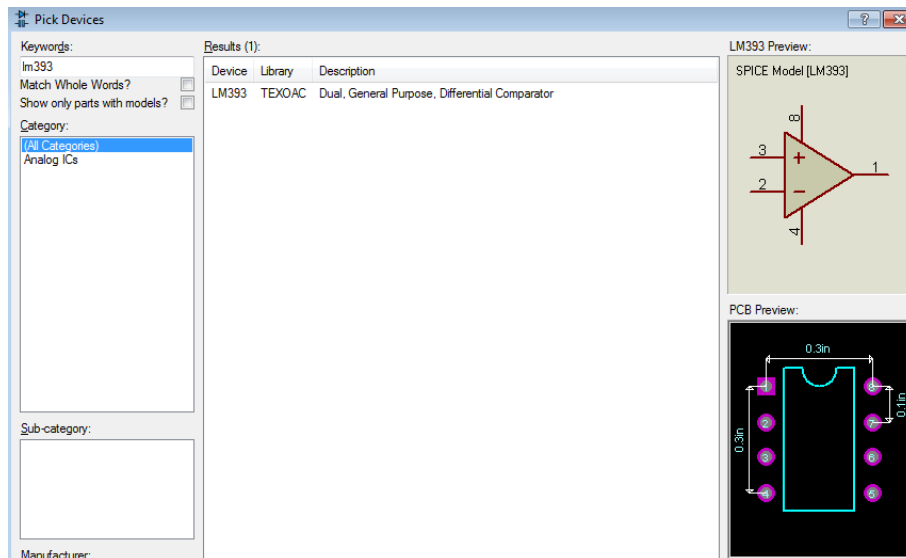


En cliquant deux fois sur 'Device', le tableau 'Pick Device' apparaît, ce dernier contient une bibliothèque qui contient pratiquement tous les composants électroniques réparties dans plusieurs catégories.



Dans la fenêtre 'Keywords', nous indiquons les mots clés de chaque composant. Puis on clique sur les éléments correspondants, ces derniers s'ajoutent automatiquement sur le volet 'Device'.

Certains composants comme le capteur infrarouge, ne figurent pas dans la bibliothèque, nous pouvons donc chercher le microcontrôleur qui commande le capteur et l'ajouter à sa place. Pour le capteur IR, nous ajoutons le microcontrôleur 'LM393'.



Les mots clés correspondants à nos composants sont comme suit :

Arduino Uno ---→ ARDUINO UNO

Moteurs --→ MOTOR-DC

Pompe à eau --→ MOTOR

Moteur pas à pas Nema 17 --→ MOTOR-BISTEPER

Capteur infrarouge--→ LM393

Servomoteur---→ MOTOR-PWMSERVO

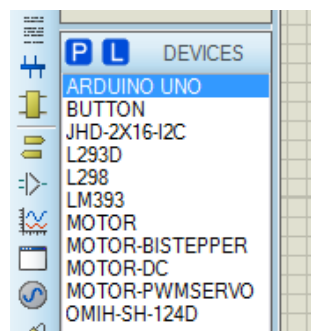
Relais --→ OMH-SH-1240

Driver Moteur --→ L293D

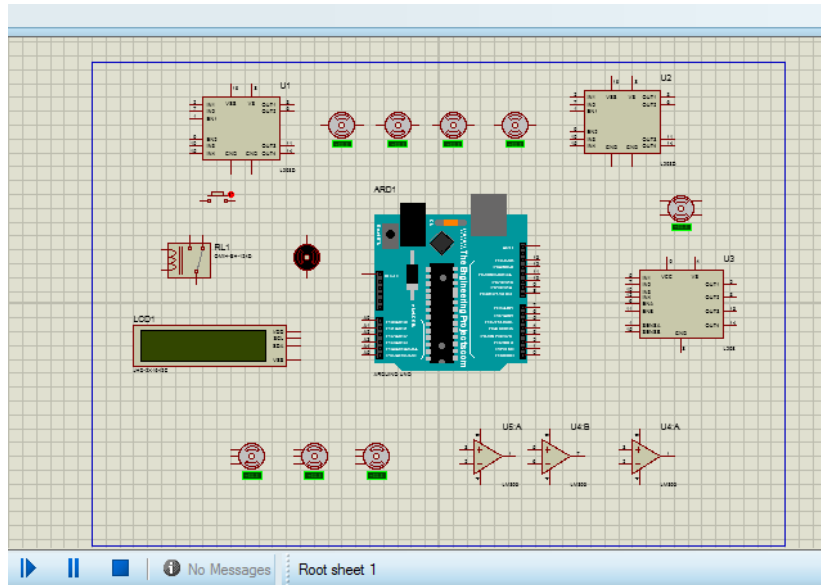
Driver Nema ---→ L298

Bouton ---→ BUTTON

Afficheur --→JHD-2X16-I2C

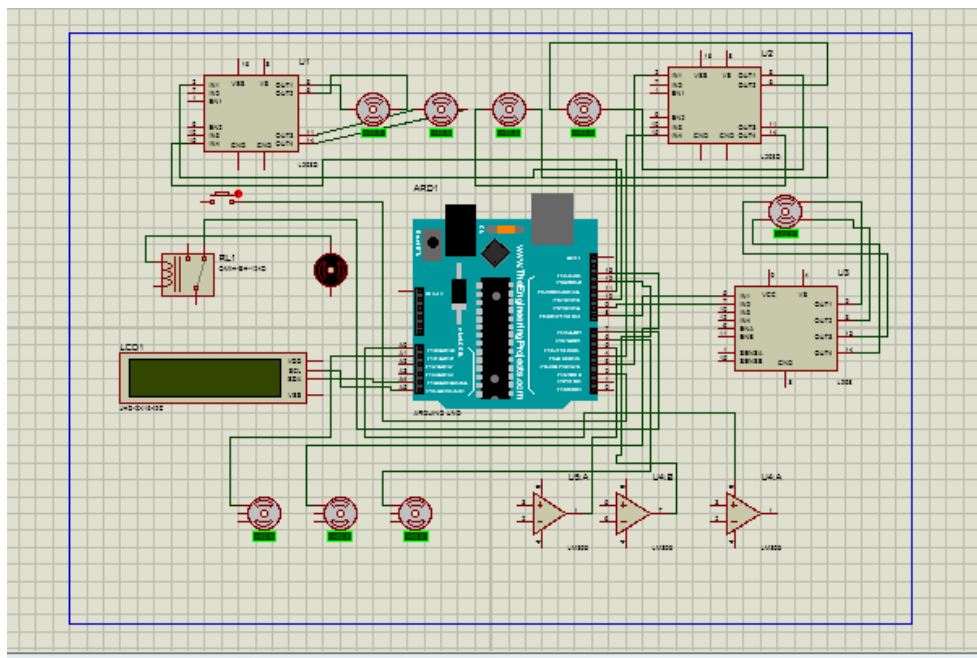


Après avoir sélectionné tous les composants de notre système et les avoir classifié sur le volet 'Device', on doit à présent situer ces éléments sur la feuille de travail de 'ISIS Proteus'. Les éléments comme le capteur et le moteur, sont placés plusieurs fois selon le cahier de charge de notre système.



Une fois que tous les composants sont sur la feuille de travail du logiciel Proteus, on passe au branchement des éléments électroniques avec la carte mère Arduino Uno. Il est nécessaire de respecter la configuration de chaque composant avec les pins de l'Arduino réalisée lors de la partie précédente de programmation.

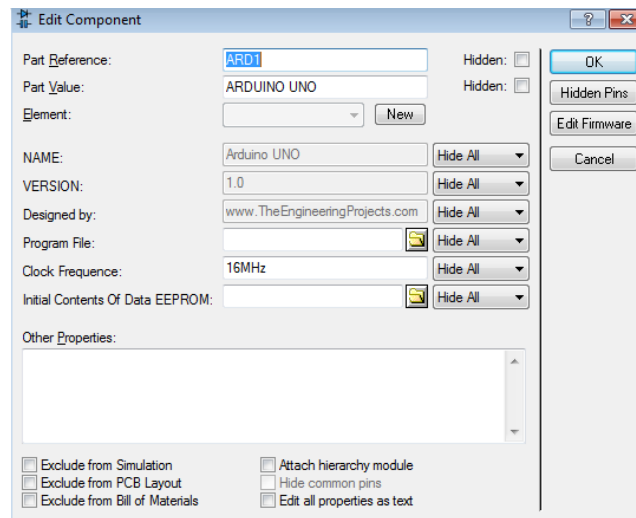
Il faut aussi ajouter une alimentation pour lier chaque élément avec la masse GND ainsi que le 12V et 5V.



III.4.2.3. Essai

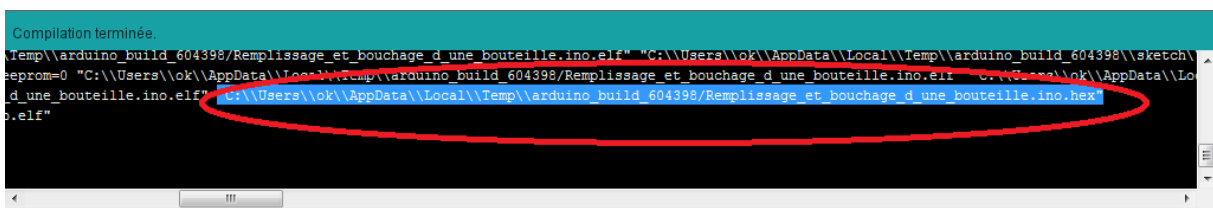
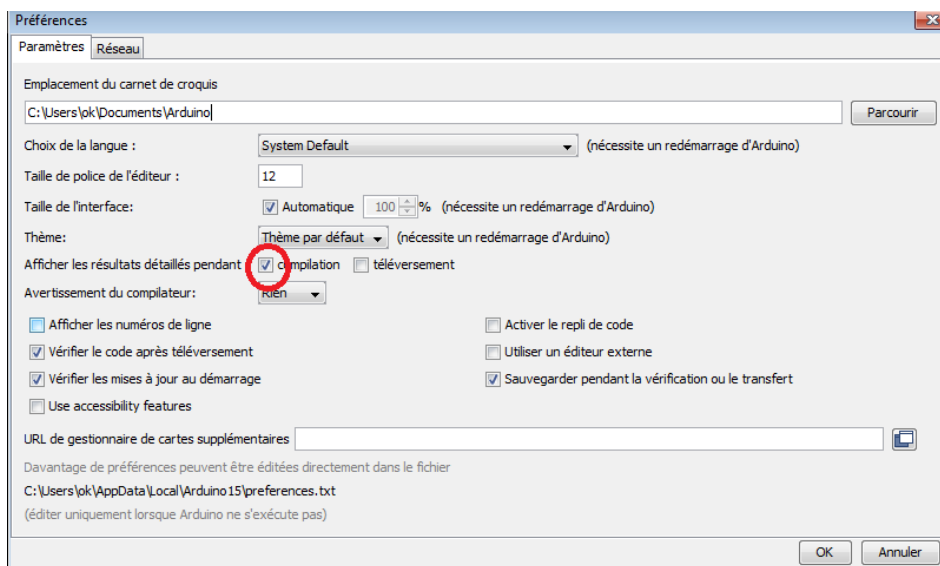
Après avoir branché l'intégralité des composants avec notre carte Arduino Uno dans les pins correspondants au code de l'IDE, on peut à présent effectuer la simulation et visualiser le fonctionnement de notre système.

En premier lieu. Il faut téléverser le code sur l'Arduino Uno. Pour cela il suffit d'effectuer un double clic sur la carte.

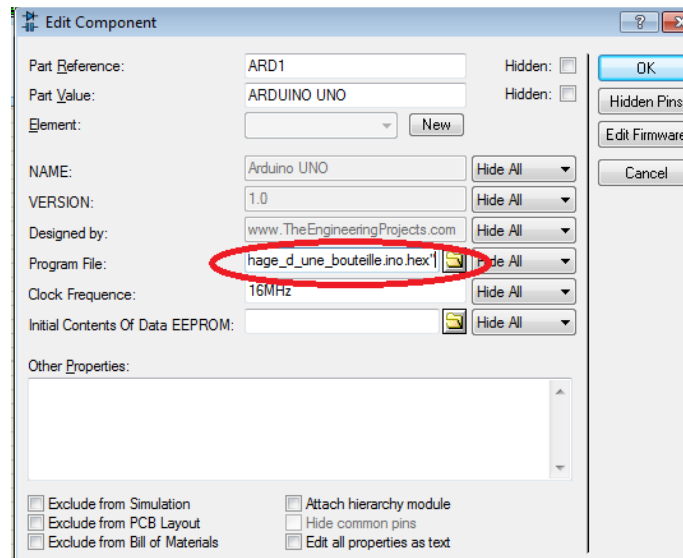


Ensuite, il faut lancer le logiciel Arduino IDE et ouvrir le code que nous avons enregistré sous le nom 'Remplissage et bouchage d'une bouteille'. Après cela, on copie le fichier '.hex' se situant sur la console de l'IDE qui assure la compilation du programme.

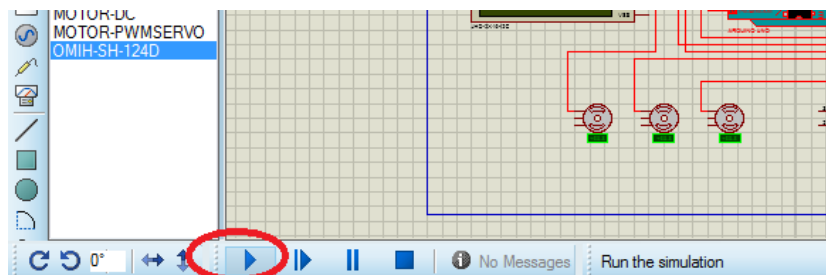
Il est utile de mentionner qu'il faut afficher les résultats pendant la compilation sur le volet 'Préférences'.



Une fois le fichier '.hex' copié, il suffit de le coller sur 'Program File' pour qu'il soit téléversé sur la carte Arduino Uno.



Après avoir téléversé le code sur la carte Arduino Uno, on lance la simulation grâce au bouton 'Run the simulation'



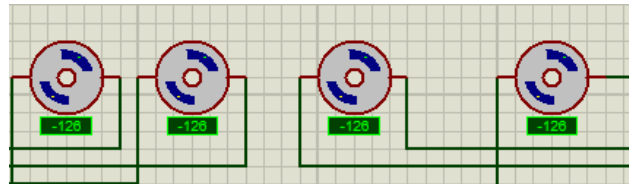
Une fois la simulation lancée, nous observons si le système fonctionne selon le diagramme d'activité et le principe de fonctionnement qu'on a établi lors des précédentes étapes.

Pour cela, il faut que ces étapes soient validées :

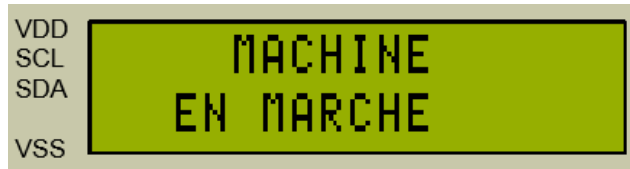
- Les messages affichés sur le LCD.
- L'état des moteurs, des servomoteurs, de la pompe à eau et du moteur pas à pas à chaque détection.



Message1 → Validé



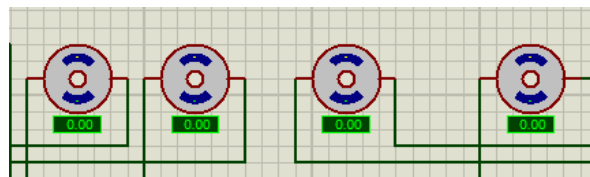
Déclenchement des moteurs → Validé



Message2 → Validé



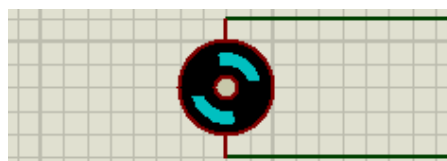
Message3 → Validé



Arrêt des moteurs -> Validé



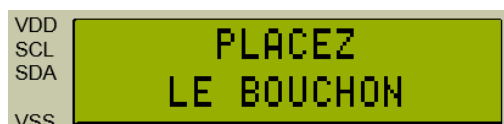
Message4 → Validé

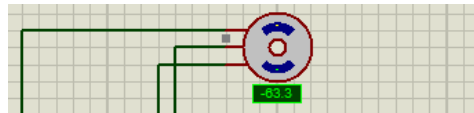
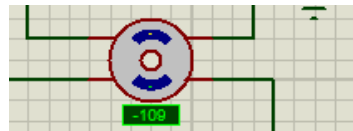


Déclenchement de la pompe à eau → Validé



Message5 → Validé



Message6 → Validé**Message7 → Validé****Bouchage via servo → Validé****Message8 → Validé****Rotation du mécanisme via Nema → Validé****Message9 → Validé****Message10 → Validé**

Les résultats de la simulation ont été satisfaisants, nous pouvons donc attaquer la prochaine partie qui concernera la réalisation.

III.5. Réalisation

Après avoir établi le principe de fonctionnement du procédé, programmé, simulé le système et effectuer les tests de branchements, nous pouvons à présent enchaîner par la dernière partie de ce travail qui consiste à la réalisation réelle de notre machine.

Dans cette partie, nous allons fabriquer notre propre système, en se basant sur les précédentes parties du chapitre et en reliant la phase Hardware et Software. Pour cela, nous allons passer par un ordre chronologique qui consistera à calculer, découper et enfin assembler les différents supports et bases qui formeront le convoyeur et le mécanisme de rotation ainsi que les différents composants électroniques et la carte Arduino qui piloteront la machine fabriquée.

III.5.1. Convoyeur

Dans cette partie, nous allons détailler les différentes étapes qui nous ont menés à concevoir le tapis roulant qui constitue la première partie de notre système et qui permet d'acheminer les bouteilles. Pour cela nous allons procéder par diviser cette partie en deux phases, la première consiste aux calculs et à la découpe des différents morceaux de bois ainsi qu'aux autres parties interagissant à la création du convoyeur. Quant à la deuxième, elle exposera la lumière sur l'assemblage de l'ensemble des matériels découpés pour enfin réaliser cette première partie de la machine.

III.5.1.1. Découpage

Dans un premier lieu, nous avons découpé un morceau de bois épais de 8*30*5 cm ainsi que deux autres morceaux fins de 15*45*0.5 cm.

Nous avons réalisé cette découpe chez un menuisier. Ce dernier a effectué cette tâche en utilisant une machine à bois combinée.

Les deux photos ci-dessous montrent l'action réelle du découpage de nos morceaux de bois grâce à la machine correspondante.



Figure III.1 : Photo réelle de la découpe du 1^{er} morceau de bois



Figure III.2 : Photo réelle de la découpe du 2eme morceau de bois

Nous avons procédé à cette découpe pour constituer les supports du convoyeur. En effet, le premier morceau de bois sera considéré comme appui pour les deux autres morceaux fins. Quant aux deux derniers ils permettront aux moteurs d'être visés aux extrémités.

Les deux photos suivantes sont des illustrations réelles du résultat de la découpe du morceau de bois épais et les deux autres morceaux fins.



Figure III.3 : Photo réelle de l'appui



Figure III.4 : Photo réelle des supports des moteurs DC

Après avoir terminé la découpe des supports constitués en bois, nous allons à présent détailler le mécanisme qui permettra au tapis de rouler.

Premièrement, nous avons choisis d'utiliser des roues dont les axes sont adaptés aux moteurs à courant continu.

Ces roues sont appropriées, puisque leurs rotations sont parfaitement synchronisées aux alternances des moteurs.

Nous avons par la suite, procédé à une petite, mais importante modification, qui consiste à enlever le pneu qui enveloppe la roue. Pour cela, nous avons utilisé un cutter afin de séparer la poulie du matériel qui l'enrobait.

Il est utile de souligner que la taille de la roue est de 65*26mm, alors que celle du trou de la roue est de 5.3*3.66mm.

Les photos ci-dessous montrent l'état des roues avant et après les modifications réalisées pour ôter le pneu



Figure III.5 : Photo réelle de la roue avant modification



Figure III.6 : Photo réelle de la roue après la modification

Afin d'achever le mécanisme qui permettra au convoyeur de rouler, nous avons choisis d'utiliser des tubes en PVC.

Pour cela, nous avons acheté un tuyeau PVC de 50mm, soit un tout petit peu plus petit que le diamètre de notre roue sans pneu.

La photo suivante montre le tube que nous avons utilisé.



Figure III.7 : Photo réelle du tube PVC

Ensuite, nous avons découpé le tube en deux morceaux de 8 cm, cette mesure n'est pas du au hasard mais à la taille du morceau de bois épais.

Le petit bout de tube ci-dessous, est la photo réelle du morceau de 8 cm découpé à l'aide d'une scie.



Figure III.8 : Photo réelle du tube PVC après la modification

En dernière partie, nous avons chauffé le tube PVC pendant quelques minutes pour qu'on puisse le déformer facilement.

Ensuite, nous avons inséré la roue à l'intérieur du tube en forçant et en déformant l'entrée jusqu'au bout.

Une fois que la roue est bien en place et en bonne position (l'axe doit être orienté vers l'extérieur), on provoque une force pour que la roue reste bien fixée à l'intérieur du tuyau.

Nous finissons par faire plonger le corps (tube + roue) dans de l'eau glacé pour le rendre rigide et fixe.

La photo suivante montre le résultat des étapes précédentes.



Figure III.9 : Résultat final de l'insertion de la roue à l'intérieur du tube PVC

III.5.1.2. Assemblage

Après avoir découpé tous les morceaux de bois et autres ustensiles nécessaires à la fabrication du convoyeur, nous allons à présent détailler leur assemblage et les astuces que nous avons effectuées pour minimiser le maximum la probabilité de défaillance.

Avant de commencer l'assemblage nous avons effectué quelques modifications sur les moteurs dc, et ceci en ôtant des supports qu'on avait jugé inutiles. La photo suivante illustre l'action.

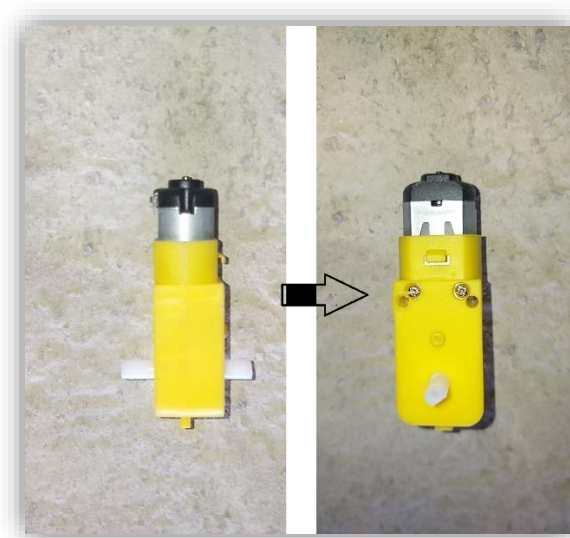


Figure III.10 : Photo réelle du moteur DC après l'enlèvement des supports

Nous avons aussi fait un traçage et des mesures bien précises sur la partie arrière des deux morceaux de bois pour faciliter la fixation et l'assemblage des différentes parties qui interagissent sur le tapis roulant.

Une fois les moteurs taillés et adaptés à la fixation sur l'appui correspondant, nous avons vissé deux des quatre actionneurs sur les extrémités d'une des deux morceaux de bois fins dans le but d'obtenir le premier gabarit.

Les photos ci-dessous montrent la fixation des moteurs ainsi que le traçage réalisé.

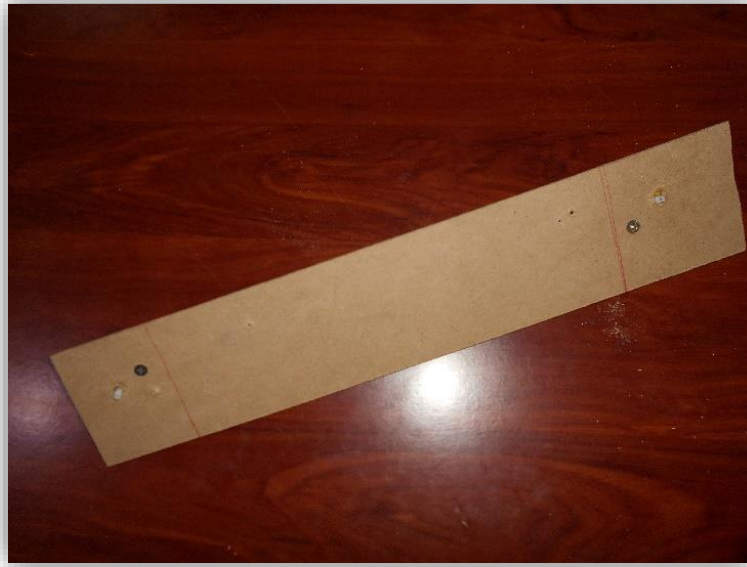


Figure III.11 : Vue A de la fixation des moteurs sur les extrémités du support

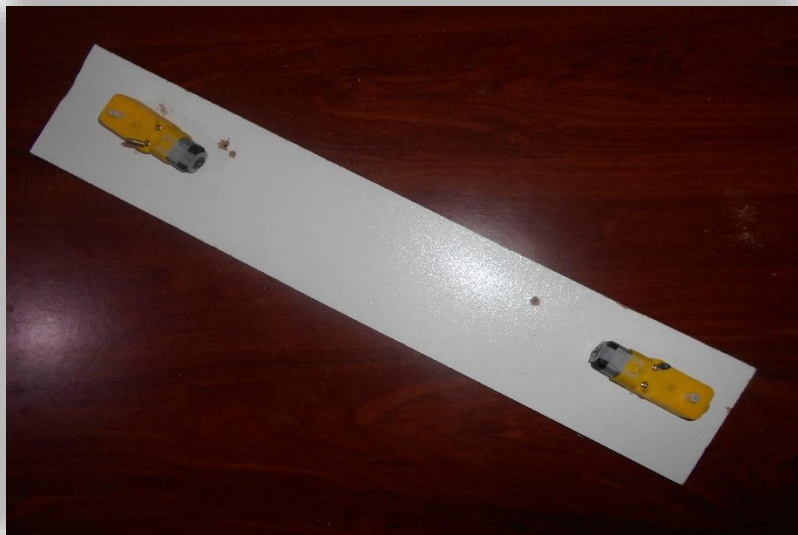


Figure III.12 : Vue B de la fixation des moteurs sur les extrémités du support



Figure III.13 : Vue C de la fixation des moteurs sur les extrémités du support

Nous avons ensuite fixé l'ensemble 'Support + 2 moteurs' sur le morceau de bois épais avec des vis comme le montre la figure suivante.



Figure III.14 : Assemblage de l'ensemble (support+ 2 moteurs) et de l'appui

Ensuite, nous avons refait le même travail avec l'autre morceau de bois fin ainsi que les deux moteurs restants, on a fixé l'ensemble sur le côté opposé du morceau de bois épais.

Puis, Nous avons insérés les quatre roues enveloppées dans les deux tubes en PVC sur les axes de chacun des quatre moteurs.

Les trois figures suivantes, illustrent les différentes vues de cette étape.



Figure III.15 : Vue A de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs



Figure III.16 : Vue B de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs



Figure III.17 : Vue C de l'insertion de l'ensemble (roue + PVC) sur les axes des moteurs

Une fois les tubes PVC en place, nous procédons à la dernière étape de cette phase qui consiste à ajouter un tapis (comme le montre la figure ci-dessous), que nous avons acheté chez un quincailler.



Figure III.18 : Photo réelle du tapis utilisé

Ce tapis a été ensuite taillé à l'aide d'un cutter, selon les mesures prises entre les deux extrémités des tubes PVC. Le tapis est très long mais étroit, comme le montre la photo ci-après.



Figure III.19 : Photo réelle de la découpe du tapis

Enfin, nous avons tendu le tapis au maximum afin qu'il puisse acheminer correctement les bouteilles, et nous l'avons attaché grâce à une colle forte de silicone. Le résultat de la réalisation du convoyeur est illustré sur les deux figures ci-dessous, et il s'est avéré très satisfaisant.



Figure III.20 : Vue A du résultat final du convoyeur



Figure III.21 : Vue B du résultat final du convoyeur

III.5.2. Mécanisme de rotation

Après avoir fini la réalisation de la première partie du projet qui est le convoyeur, nous passons dorénavant à la seconde moitié de ce procédé.

Dans cette partie, nous allons détailler toutes les étapes qui nous ont conduits à la fabrication du mécanisme de rotation, qui permettra d'effectuer les trois tâches primordiales de ce projet qui sont : remplissage, bouchage et éjection.

Pour cela nous allons procéder de la même manière que pour le convoyeur, c'est-à-dire, que nous allons diviser notre travail en partie découpage et en partie assemblage.

III.5.2.1. Découpage

Tout d'abord, nous avons chez un menuisier, découpé deux morceaux de bois de 20*25 cm, avec une machine à bois combinée.

Le premier morceau de bois sera considéré comme support pour les bouteilles d'eau et sera fixé en bas de la tige.

Quant au deuxième morceau de bois, il a été conçu de telle sorte que la bouteille soit calée sur l'un de ses quatre espaces qu'on a créé. Ces espaces ont été calculés en se basant sur le diamètre d'une petite bouteille d'eau standard.

Les espaces de formes carrées, donneront une allure d'un plus (+) inversé au morceau de bois. Pour les concevoir, nous avons d'abord utilisé la machine de bois combinée. Puis, nous avons terminé par les taillés via une machine à scie verticale.

Il est utile de préciser que nous avons découpé nous même le support de bouteilles dans le but d'obtenir les mesures exactes. L'action a été effectuée à l'aide d'une machine meuleuse à disque en bois. Les photos suivantes illustrent l'intégralité de cette étape.



Figure III.22 : la découpe du support



Figure III.23 : Prendre des mesures exactes pour couper le récipient



Figure III.24 : la découpe du bois utilisant la machine de bois combiné

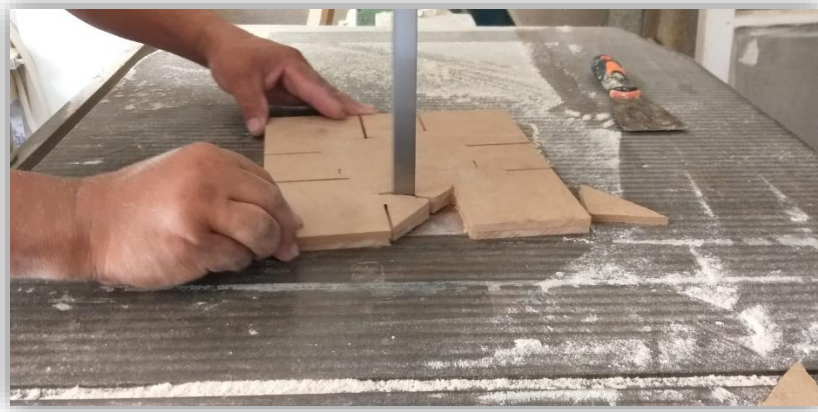


Figure III.25 : la découpe du bois utilisant la machine à scie verticale



Figure III.26 : Préparation du récipient

En dernière étape de cette phase nous avons poli les morceaux de bois à l'aide d'un papier abrasif, le résultat final du découpage est comme suit.



Figure III. 27 : Résultat final du récipient



Figure III.28 : Photo réelle du support des bouteilles

Enfin, nous avons découpé une tige trapézoïdale de 10 mm de diamètre à la hauteur dont nous aurons besoin.



Figure III.29 : Découpe de la tige

III.5.2.2. Assemblage

Après avoir découpé les deux morceaux de bois, nous allons voir comment le mécanisme de rotation sera assemblé.

Avant d'avoir commencé à assembler la tige, le moteur, le support et le récipient. Il est nécessaire d'indiquer que le Nema 17 était muni d'un pignon correspondant à une imprimante, ce qui nous a obligé à le modifier en ôtant ce dernier.

Pour cela, nous nous sommes rendus chez un mécanicien qui a réussi à l'enlever à coup de pression grâce à une machine de poinçonnage comme le montre la photo ci-dessous.



Figure III.30 : L'enlèvement du pignon du moteur

Une fois l'axe du moteur pas à pas libéré, nous avons commencé l'assemblage du mécanisme.

Nous avons donc créé un corps 'moteur +tige' via un accouplement adaptées à tous les types de Nema. Nous avons utilisé des vis, puis nous avons renforcé le tout par de la colle forte pour que la rotation se fait correctement.

La photo ci-après montre le premier assemblage que nous avons effectué.



Figure III.31 : Assemblage du moteur et de la tige via un accouplement

Ensuite, nous avons insérer les deux morceaux de bois (récipient et support). Pour cela, nous avons fait un petit trou à leurs axes à l'aide d'une baguette chauffante. Cette figure du support illustre l'opération.



Figure III.32 : photo du support après l'avoir percé.

Une fois les morceaux prêts, nous les insérons sur la tige en les serrant avec des écrous et des contre écrous, puis en les collant avec de la silicone pour être sûr que la synchronisation avec la rotation pas à pas soit parfaite et juste. Ces deux photos montrent désormais le nouveau corps (moteur +accouplement +tige +support +récipient)



Figure III.33 : Photo réelle du nouveau corps (moteur + accouplement +tige +support +récipient)

Une fois le corps assemblé, nous l'avons fixé sur un bout de bois bien épais qui sera considéré comme support. Nous avons donc fait un vide similaire à un trou ovale sur ce morceau de bois, puis on l'a remplie avec une matière fixatrice et nous avons enfoncé le moteur pas à pas dedans. Comme le montre la photo suivante.



Figure III.34 : Fixation du moteur NEMA17 sur le support en bois

Après avoir renforcé le corps sur un support, nous avons commencé des essais manuels de rotation, et nous avons remarqué que la tige tournée mais n'était pas stable, ce qui engendre un mouvement agité de l'ensemble du corps, ce qui serait un véritable souci pour la rotation des bouteilles.

Pour régler ce problème, nous avons décidé de fabriquer un mécanisme qui limitera le champ de mouvement de la tige.

Nous avons donc d'abord ajouté une base constituée de deux grands supports en bois rouge qui permettra de renforcer l'ensemble du système.

Ensuite, nous avons vissé un support vertical sur cette base, à la hauteur de la tige grâce à un 'L', puis nous avons fixé un autre support horizontal de la même manière sur le premier support.

Un trou au même diamètre de la tige a été conçu à l'extrémité du support horizontal pour que la tige soit insérée dedans et n'aura donc qu'un petit champ de liberté, ce qui permettra au système de tourner à la perfection.

Les photos de cette étape sont illustrées ci-dessous.



Figure III.35 : Photo réelle du support vertical fixé sur la base du bois rouge



Figure III.36 : Photo réelle du support horizontal fixé sur le support vertical



Figure III.37 : Insertion de la tige dans l'extrémité du support

Les photos qui suivent montrent le résultat final du mécanisme de rotation, qui comme le convoyeur s'est avéré satisfaisant



Figure III.38 : Vue A du résultat final du mécanisme de rotation



Figure III.39 : Vue B du résultat final du mécanisme de rotation



Figure III.40 : Vue C du résultat final du mécanisme de rotation

III.5.3. Finition

Après avoir découpé et assemblé les deux compartiments essentiels de notre système, qui sont le convoyeur et le mécanisme de rotation. Nous allons à présent voir les dernières modifications appliquées sur le procédé pour minimiser toutes sortes de défaillances.

III.5.3.1. Mécanisme des servomoteurs

Comme nous l'avons déjà évoqué lors du principe de fonctionnement, chaque étape du cycle d'embouteillage de notre système est pilotée à travers des bras actionnés par des servomoteurs.

Pour cela nous avons utilisé des abaisses langues comme bras et nous les avons collés sur les supports correspondants aux axes des servomoteurs, comme le montre la figure ci-dessous.



Figure III.41 : Collage du support du servomoteur sur l'abaisse langue

Une fois les abaisses langues collés sur les supports, nous les insérons sur les axes des trois servomoteurs.

Le premier servomoteur est fixé sur le support du mécanisme de rotation afin d'actionner un bras permettant de caler la bouteille et de sécuriser l'étape du remplissage.

La photo suivante illustre cette action.



Figure III.42 : Fixation du 1^{er} bras pour caler la bouteille

Le second servomoteur est lui fixé sur le dessus du mécanisme de rotation, soit dans la partie où la tige est limitée.

Dans ce cas-là, le servomoteur actionne un bras permettant de boucher la bouteille remplie auparavant, comme le montre la photo ci-après.



Figure III.43 : Fixation du bras pour boucher la bouteille

Quant au troisième et dernier servomoteur de notre jeu de bras, il est fixé sur l'autre partie du support interagissant sur le mécanisme de rotation.

Ce dernier servomoteur actionne l'abaisse langue afin d'éjecter la bouteille remplie et bouchée end dehors de la machine.

La figure ci-dessous illustre cette dernière étape.

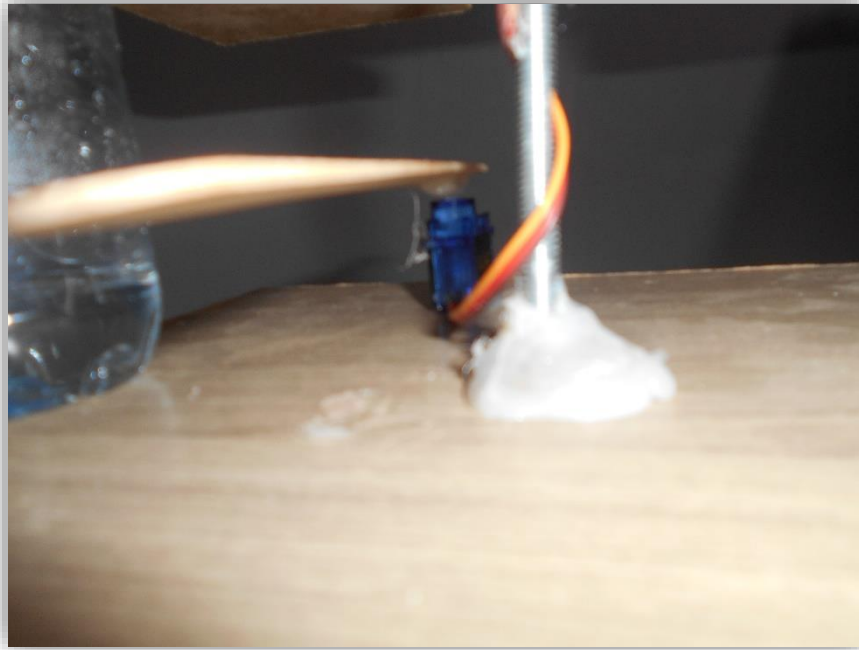


Figure III.44 : Fixation du bras pour éjecter la bouteille remplie et boucher

III.5.3.2. Support du convoyeur

Lorsqu'on a terminé de mettre en place le convoyeur et le mécanisme de rotation, nous avons remarqué que ce dernier était plus élevé que le tapis roulant, ce qui engendrerait un faux acheminement des bouteilles d'eau de la première partie à la deuxième.

Pour régler ce problème nous avons donc conçu un support sous forme de table, afin que le convoyeur soit à la hauteur du mécanisme de rotation.

Ce support a été créé, en assemblant bout de bois avec un grand morceau de même matière, en utilisant des vis et de la colle forte sur les pieds de la table pour le fixer sur le grand support du système.

Ces deux photos illustrent parfaitement ce que nous avons décrit.



Figure III.45 : Vue A du support ajouté



Figure III.46 : Vue B du support ajouté

III.5.3.3. Modification du mécanisme de rotation.

Nous nous sommes aussi confrontés à une autre complication comme tant d'autres au cours de ce projet.

Nous avons remarqué que le mécanisme de rotation touchait les bordures du tapis lorsqu'il tournait, ce qui bloquait la rotation et chamboulait tous nos calculs.

Afin de régler ce problème nous avons modifié le support du mécanisme de rotation, en effectuant une découpe diagonale sur les extrémités du support pour qu'il puisse éviter de se frotter aux tapis.

Après plusieurs essais, le résultat s'est avéré satisfaisant. Les deux figures suivantes expliquent d'avantage cette étape.



Figure III.47 : Vue A de la modification du support



Figure III.48 : Vue B de la modification du support

III.5.3.4. Panneau de commande

Afin de permettre à l'utilisateur de commander la machine et de visualiser les étapes sur un écran LCD, nous avons conçu un panneau de commande.

Pour cela, nous avons découpé un morceau de bois de 60*20 cm. Pour ensuite, faire des ouvertures correspondantes à la taille d'un bouton poussoir 'Start' et d'un afficheur I2C 16*2 LCD.

La figure ci-dessous montre le panneau de commande et les ouvertures que nous avons évoquées.



Figure III.49 : Photo réelle du panneau de commande

De plus de commander et d'afficher, le panneau de commande à un coté extrêmement important dans le volet 'design', car il permet de cacher toute l'installation des câbles électroniques ainsi que de donner une vue plus moderne à la machine, comme le montre la photo suivante.

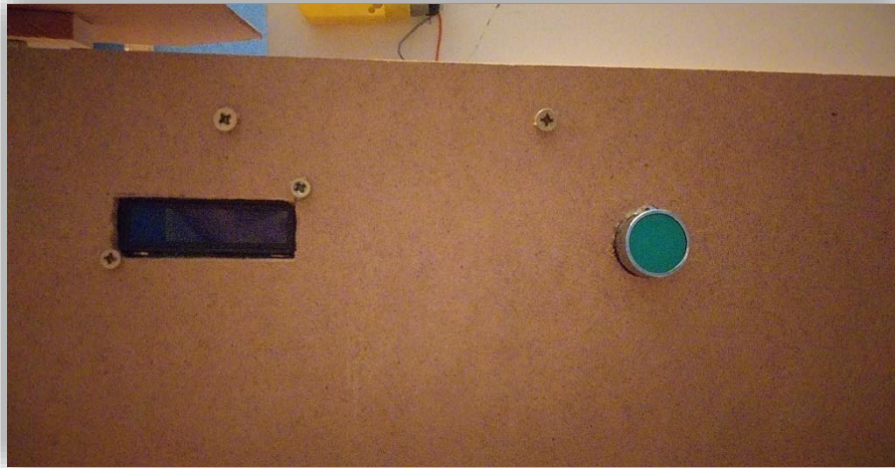


Figure III.50 : Résultat final du panneau

III.5.3.5. Branchement

La dernière étape de notre travail était d'effectuer le branchement de tous les composants électroniques avec la carte Arduino et l'alimentation.

En premier lieu, nous avons choisis l'extrémité de la machine pour fixer notre Power Supply 12V5A, ainsi que l'abaisseur de tension. Pour cela nous avons utilisé de la colle forte. Il est utile de préciser que nous avons coupé une prise d'un ancien téléviseur pour l'utiliser dans notre système.

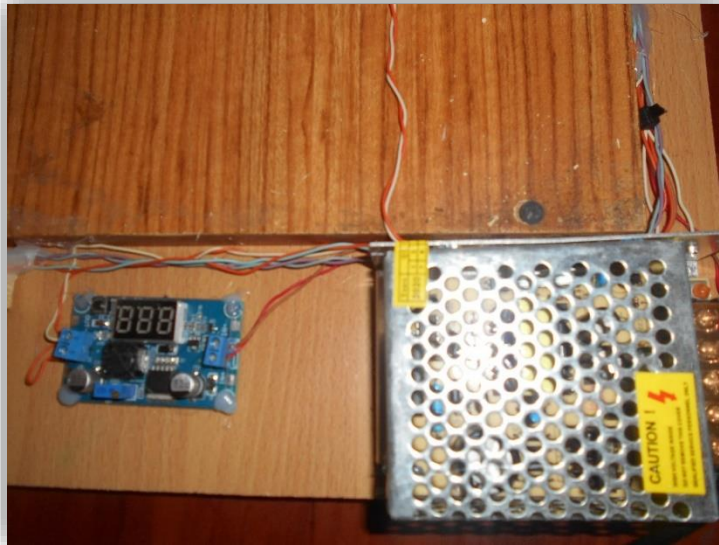


Figure III.51 : Fixation du Power Supply et de l'abaisseur

Nous avons utilisé une plaque d'essai pour distribuer l'alimentation en 12V et 5V aux composants.

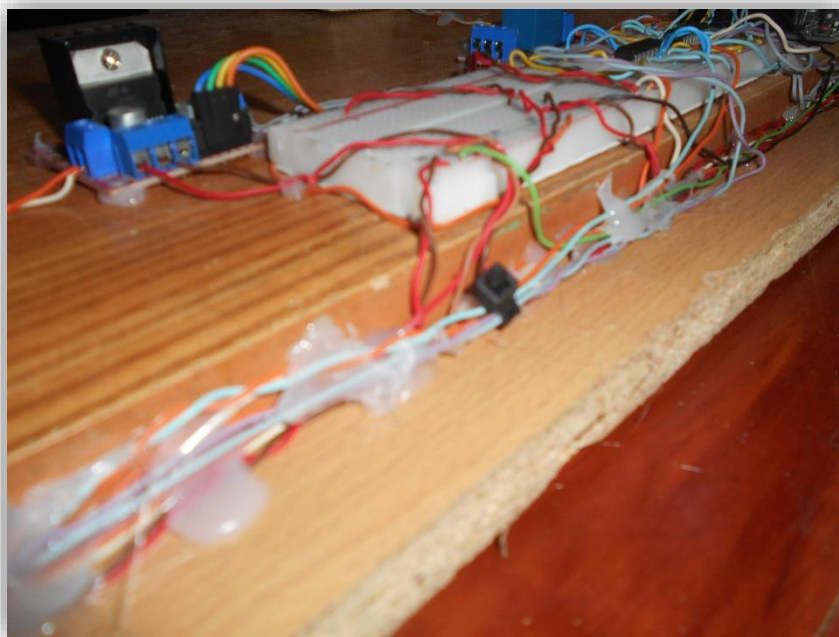


Figure III.52 : Photo réelle du branchement des composants sur la plaque d'essai

Nous avons fixé l'Arduino Uno, le relais, le L298N ainsi que les L293D, à côté de la plaque d'essai pour faciliter le branchement, comme le montre ces deux figures.

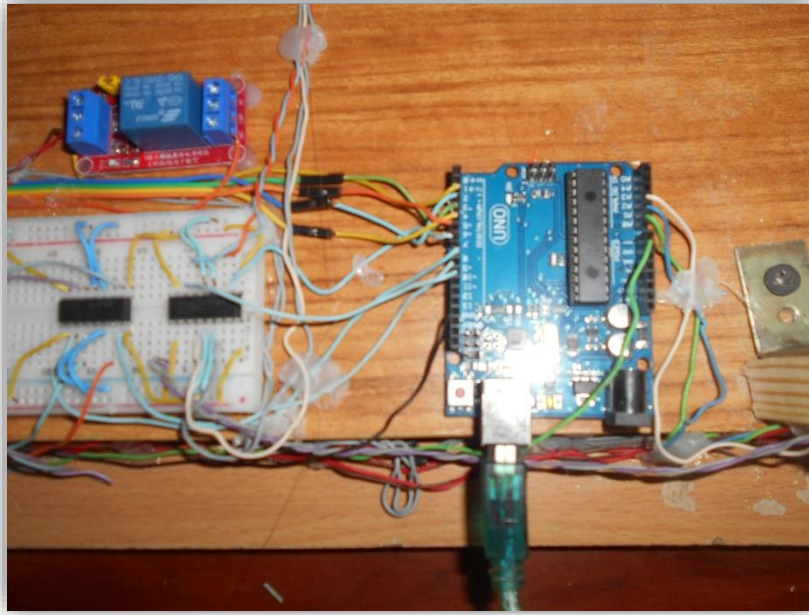


Figure III.53 : Vue A des composants fixé à côté de la plaque d'essai

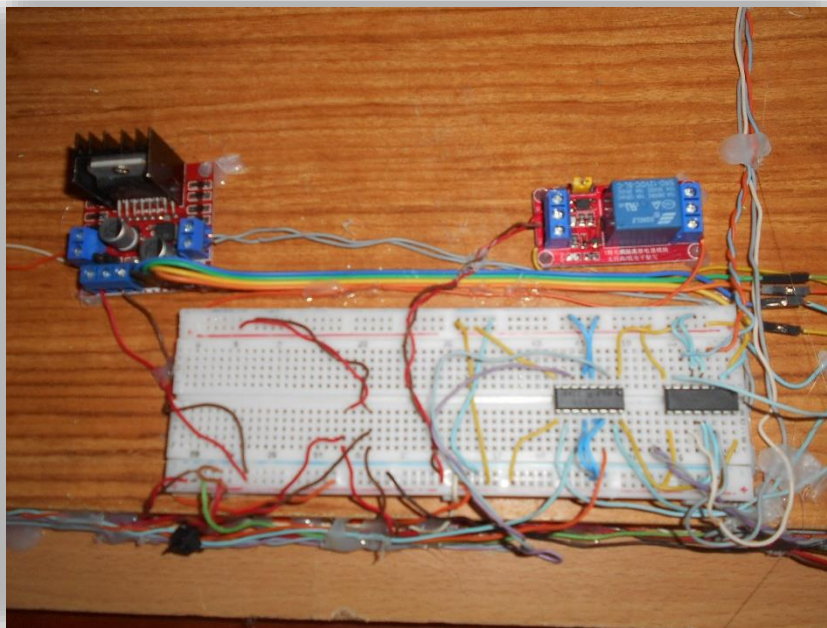


Figure III.54 : Vue B des composants fixé à côté de la plaque d'essai

Quelques astuces de branchement. Nous avons utilisé de la colle forte pour fixer les câbles sur la petite bague qu'on a créée en superposant deux supports. On a aussi renforcé le tout par des attaches zippées.

Pour les servomoteurs nous avons branché via les deux supports verticaux et horizontaux du mécanisme de rotation.

Enfin, les moteurs ont été raccordés grâce à des dominos fixés sur les pieds du support du convoyeur.

Les trois photos ci-dessous illustrent le câblage ainsi que les astuces de branchement qu'on a expliqué.



Figure III.55 : Vue A du branchement de la carte Arduino avec les composants

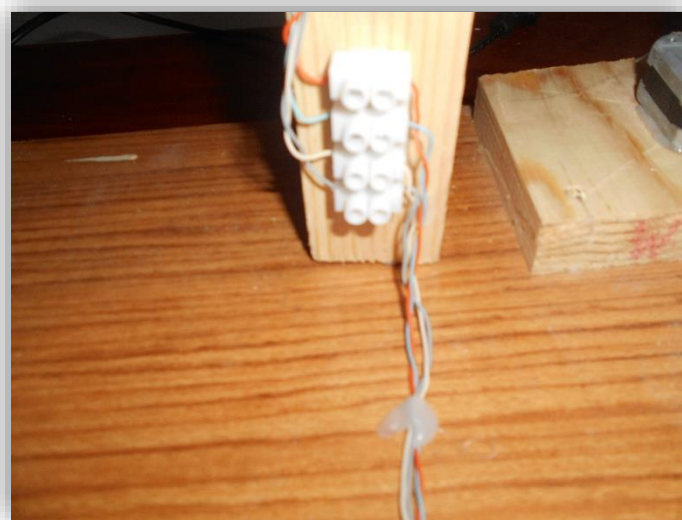


Figure III.56 : Vue B du branchement de la carte Arduino avec les composants

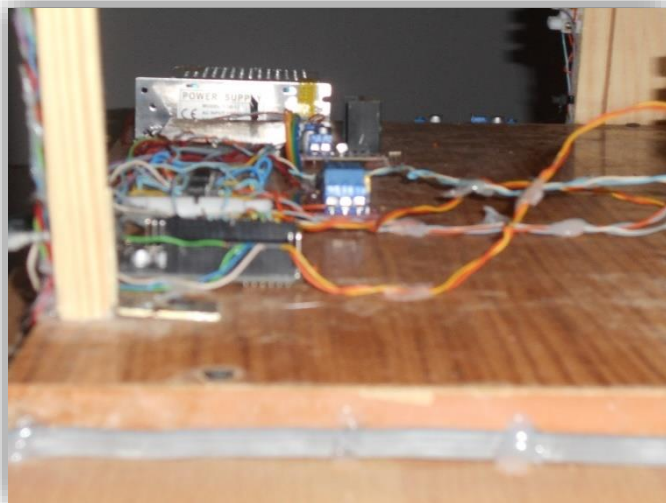


Figure III.57 : Vue C du branchement de la carte Arduino avec les composants

III.6. Conclusion

Dans ce chapitre, nous avons vu toutes les étapes qui nous menés à concevoir notre machine de remplissage et de bouchage à base d'Arduino Uno.

Nous avons commencé par analyser les besoin fonctionnels et non fonctionnels de notre système en les schématisant dans un diagramme d'activité. Puis, nous avons établi un principe de fonctionnement détaillé qui englobe toutes les étapes du procédé.

Une fois la feuille de route tracé, nous avons ensuite attaqué la partie 'Software'. Cette dernière a été divisée en deux phases. La première est consacrée à la programmation de notre carte Arduino avec le logiciel correspondant. Quant à la deuxième, nous avons fait une simulation complète du code avec le progiciel Proteus.

Après avoir réussi la simulation, nous avons terminé par expliquer toutes les étapes de la réalisation, en exposant toutes les photos réelles des différentes parties qui constituent notre machine, ainsi que le branchement et les mécanismes interagissant sur le fonctionnement de notre système.

CONCLUSION GENERALE

Conclusion générale et perspectives

Les lignes d'embouteillages sont de plus en plus présentes dans les grandes usines de production de boissons, chips, compléments alimentaires et une multitude d'autres produits alimentaires et non alimentaires. Elles permettent à une entreprise de gagner en croissance, en productivité et d'éliminer un maximum de défaillances.

L'élaboration de ce travail dans le cadre du projet de fin d'étude, nous a permis d'approfondir nos connaissances théoriques en électronique et en langage de programmation, la machine fabriquée nous a aussi permis d'acquérir une bonne expérience au niveau de la conception et du branchement.

La machine fabriquée a été conçue en bois, et piloté par une carte Arduino Uno. Cette dernière représentait le cerveau de notre projet en contrôlant des moteurs à courant continu, des servomoteurs et des moteurs pas à pas.

La réalisation matérielle et logicielle de ce projet suivie d'une phase de validation et de simulation a donné des résultats attendus et satisfait notre cahier des charges.

Ce projet nous a fait découvrir aussi un secteur que nous ne connaissions pas vraiment et qui nous a intéressés de plus en plus au fur et à mesure que nous approfondissions nos recherches. Cela nous a permis de recueillir des informations techniques et des explications nécessaires à la compréhension du principe de fonctionnement de certaines technologies.

Il est certain que notre projet est loin d'être terminé. Comme perspectives, nous souhaitons l'améliorer et l'enrichir avec des technologies encore plus modernes et lui apporter plus de fonctionnalités pour qu'on puisse l'exposer au grand public ; parmi les attentes pour notre projet nous citons :

- Faire une extension du convoyeur et utilisé des moteurs 24V, afin de pouvoir acheminer un maximum de bouteilles.
- Ajouter un système de capsulage pour avoir la polyvalence et pouvoir effectuer le bouchage de n'importe quel type de bouteille.
- Créer une application pour recevoir en temps réel toutes les informations du remplissage, bouchage et éjection des bouteilles.
- Insérer un système de contrôle à distance de la machine via un port Bluetooth ou un réseau wifi.
- Fabriquer la même machine, mais en utilisant de la tôle noire au lieu du bois pour donner robustesse et modernité au système.

ANNEXE

Annexe A

Microcontrôleurs picoPower 8 bits ATmega328P megaAVR

Le microcontrôleur Atmel ATmega328P est un circuit megaAVR 8 bits basé sur l'architecture RISC améliorée AVR. Il intègre la technologie picoPower qui offre une consommation ultra-faible et des modes de veille à faible consommation, idéal pour les applications alimentées par batterie.

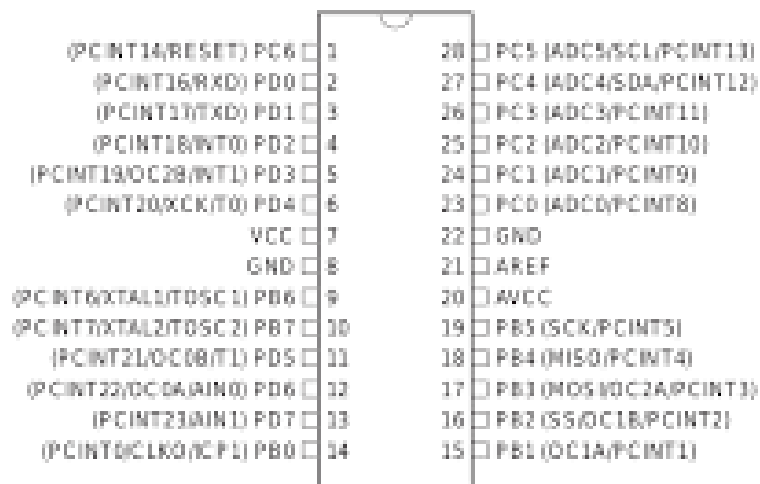
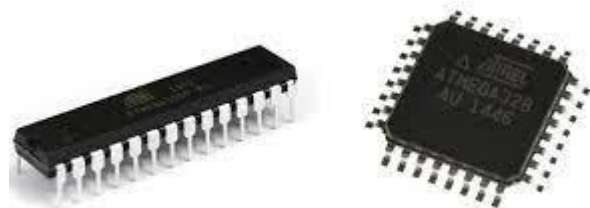


Schéma des broches de l'Arduino UNO Rev3

Caractéristiques du microcontrôleur

- Fréquence d'utilisation : 20 MHz max.
- Jusqu'à 20 MIPS à 20 MHz
- Flash 32 ko
- EEPROM 1 024 o
- SRAM 2 ko
- 131 instructions : la plupart des cycles d'horloge simples
- 32 registres d'usage général
- 23 GPIO

- Reset à la mise sous tension et détection de baisse de tension programmable
- Oscillateur étalonné en interne
- Sources d'interruption externe et interne
- Six modes veille d'économie d'énergie
- Interruptions interne et externe
- Fonctionnement entièrement statique

Caractéristiques techniques

Attribut	Valeur
Série	ATmega
Type de boîtier	PDIP
Type de montage	Traversant
Nombre de broches	28
Cœur du circuit	AVR
Largeur de bus de données	8bit
Taille de la mémoire programme	1 kB, 32 kB
Fréquence maximum	20MHz
Taille de la RAM	2 Ko
Canaux USB	0
Nombre d'unités PWM	3
Nombre de canaux SPI	1
Nombre de canaux CAN	0
Nombre de canaux I2C	1
Tension d'alimentation de fonctionnement typique	1,8 → 5,5 V
Nombre de canaux UART	0
Nombre de canaux USART	1
Nombre de canaux Ethernet	0
Type de mémoire programme	Flash
Dimensions	37.4 x 6.76 x 3.28mm
Nombre de canaux LIN	0

Attribut	Valeur
Hauteur	3.28mm
Température de fonctionnement minimum	-40 °C
Architecture de jeu d'instruction	RISC
Nombre de canaux PCI	0
Modulation de largeur d'impulsion	2 (6 x 8 bits), 6 x 16 bits
Largeur	6.76mm
CAN	6 x 10 bits
Longueur	37.4mm
Température d'utilisation maximum	+85 °C
Nombre d'entrée CNA	1

Annexe B

Schéma électronique de la carte Arduino UNO Rev3

Schéma simplifié de la carte Arduino UNO Rev3 dont les broches du microcontrôleur sont reliées à des connecteurs selon le schéma ci-

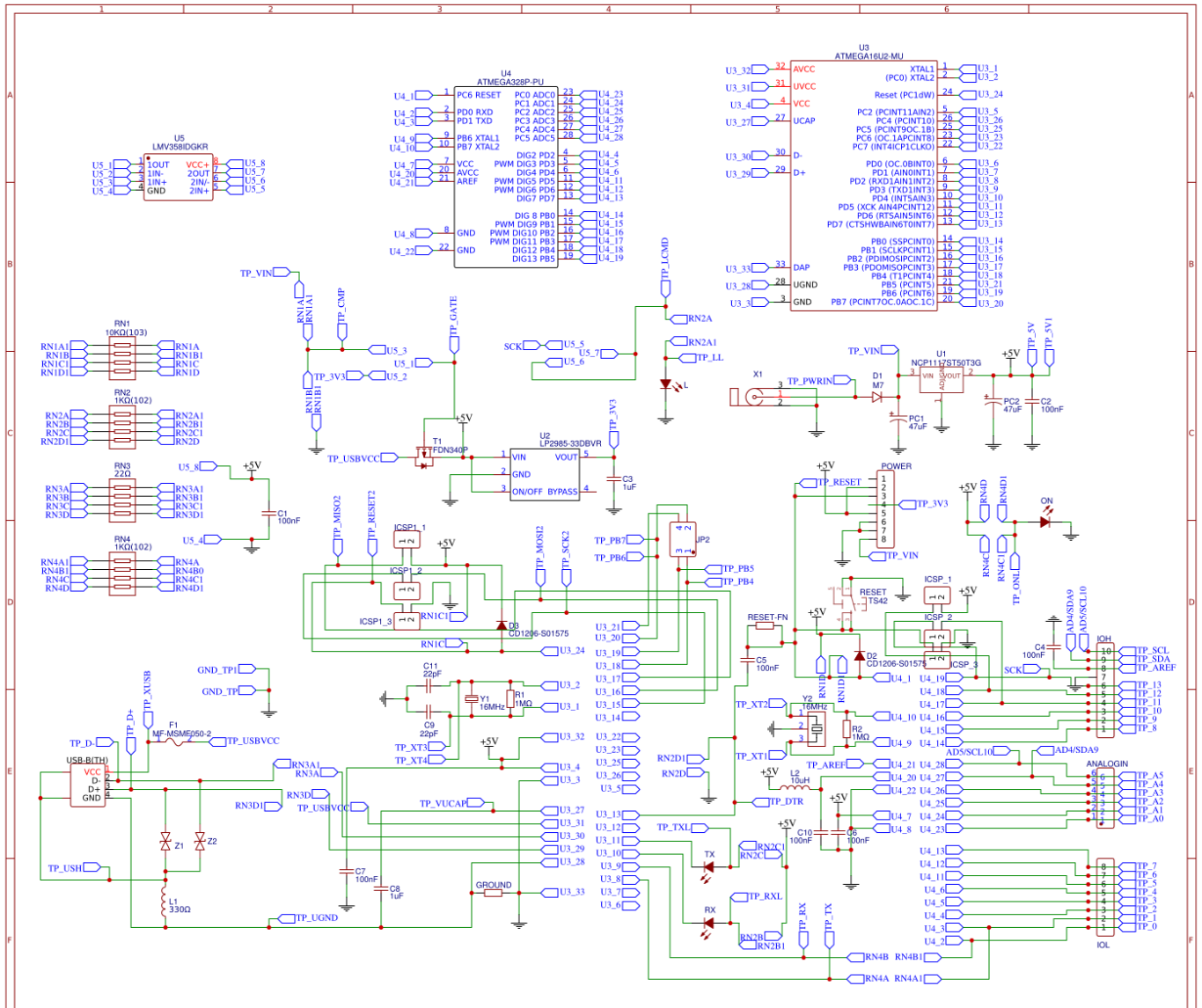


Schéma simplifié de l'Arduino UNO Rev3

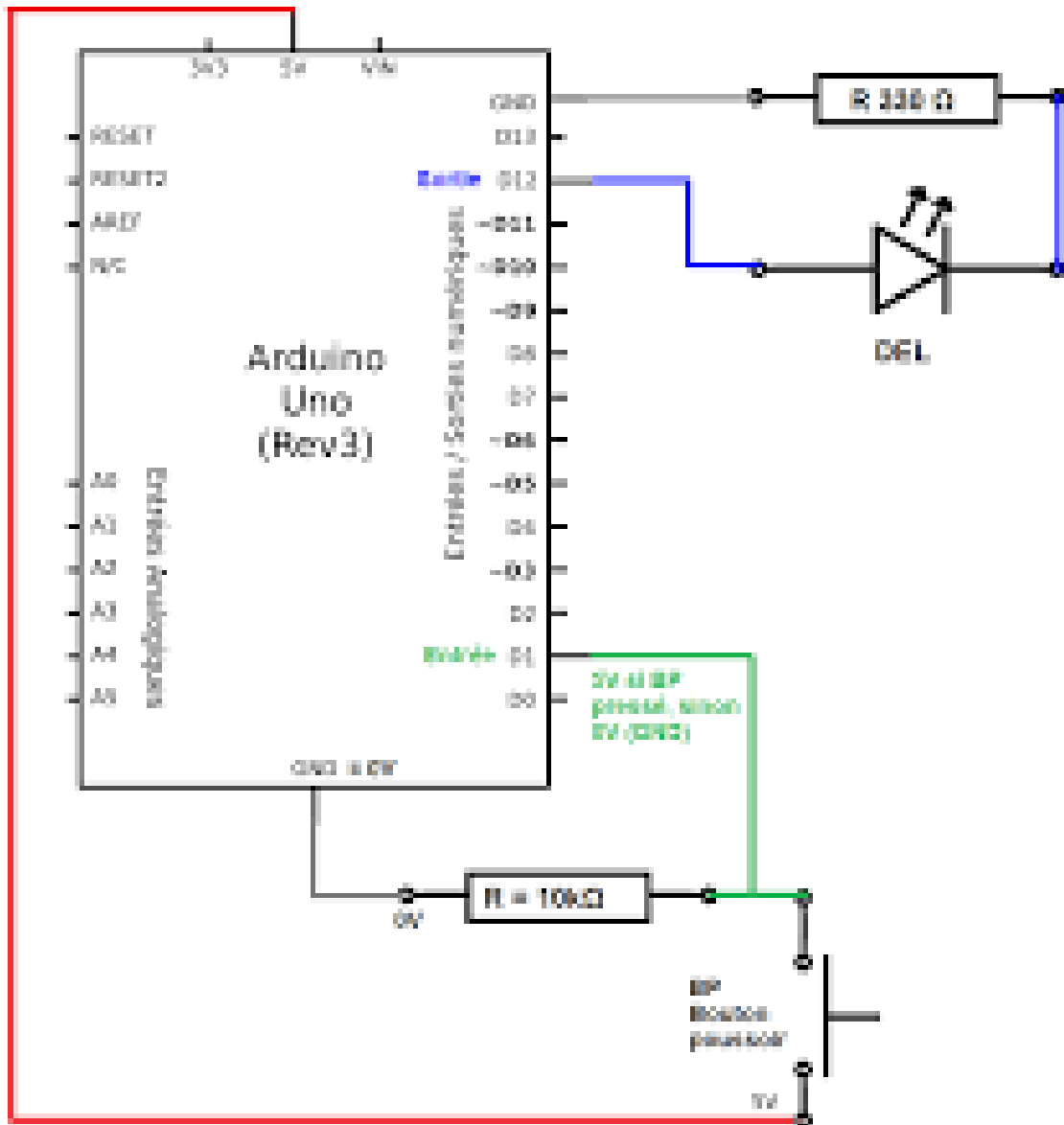


Schéma simplifié de l'Arduino UNO Rev3

Annexe C

La fiche technique du module L298N

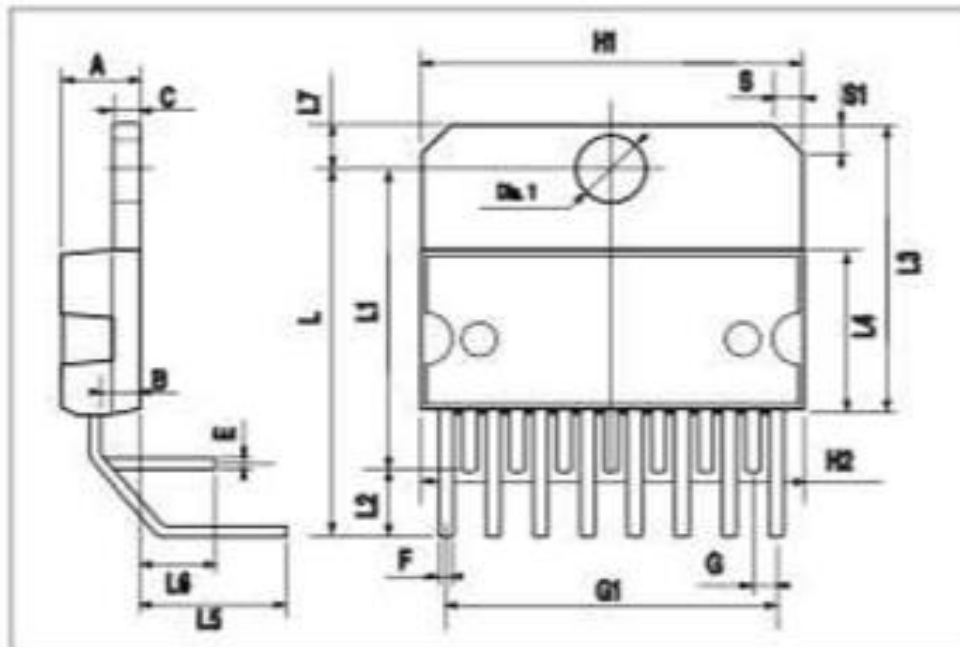
L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.8			0.071
E	0.49		0.55	0.019		0.022
F	0.86		0.75	0.034		0.030
D	1.14	1.27	1.4	0.045	0.051	0.055
D1	17.57	17.78	17.81	0.692	0.700	0.705
H1	19.8			0.779		
H2			20.2			0.795
L		20.37				0.802
L1		18.02				0.710
L2		2.54				0.100
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.8	0.406	0.421	0.429
L5		5.25				0.208
L6		2.55				0.101
L7	2.85		2.9	0.114		0.114
S	1.8		2.8	0.071		0.110
S1	1.8		2.8	0.071		0.110
Dw1	3.85		3.85	0.152		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



11/13

La fiche technique de L 298 (01)

L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.0			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.012
D (1)	15.8		16	0.622		0.630
D1	3.4		3.5	0.134		0.138
E	13.0		14.5	0.512		0.570
e		1.27			0.050	
e2		11.43			0.450	
E1 (1)	10.0		11.1	0.433		0.437
E2			2.0			0.114
E3	3.9		4.2	0.229		0.244
G	0		0.1	0.000		0.004
H	15.5		15.8	0.610		0.620
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	12 (max.)					
S	8 (max.)					
T		1.0			0.039	

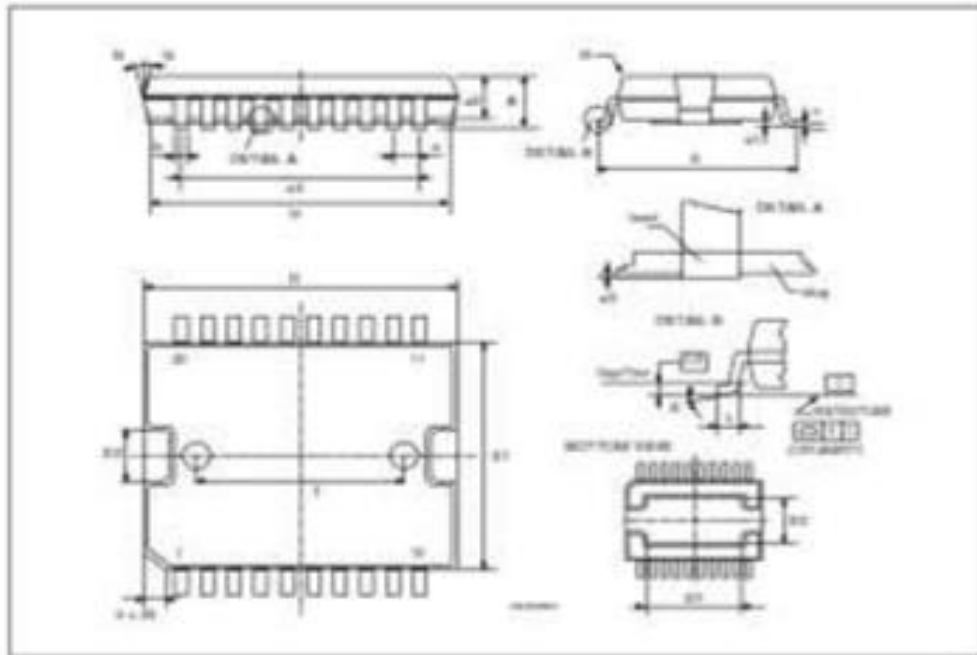
(1) D and T do not include mold flash or protrusions.
 Mold flash or protrusions shall not exceed 0.15 mm (0.006").
 Critical dimensions: "E", "E1" and "a2"

OUTLINE AND MECHANICAL DATA



JEDEC MO-158

PowerSO20



12112



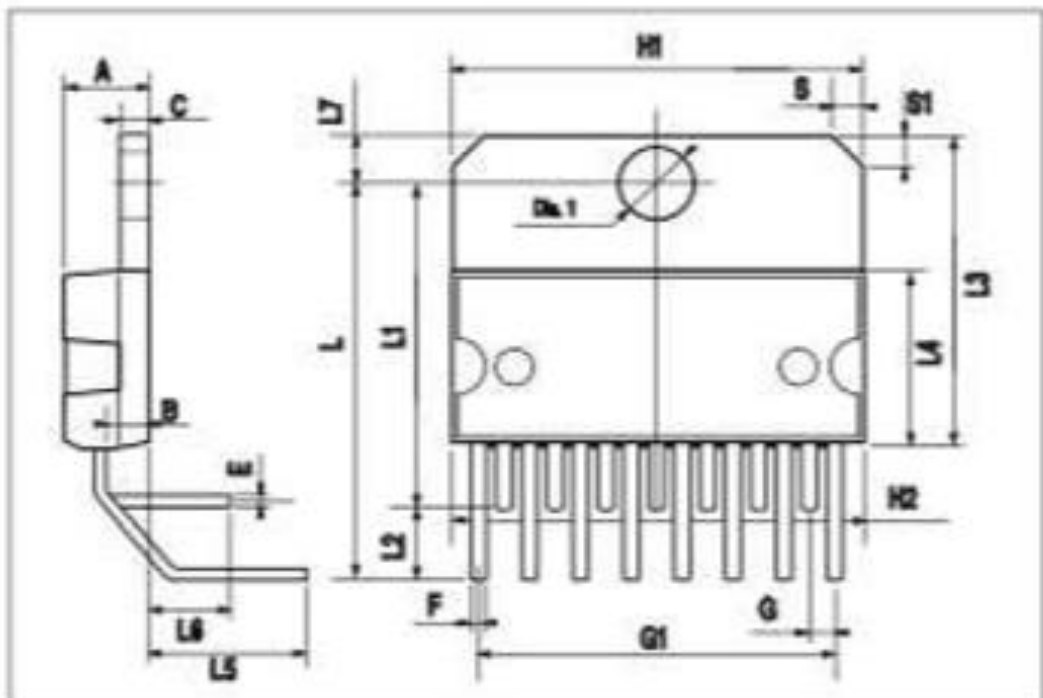
La fiche technique de L 298 (02)

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.05			0.104
C			1.8			0.083
E	2.49		0.35	0.019		0.022
F	0.85		0.75	0.033		0.030
G	1.34	1.27	1.4	0.045	0.050	0.055
G1	17.87	17.75	17.81	0.699	0.700	0.705
H1	15.8			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.32			0.713	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.2	10.7	10.8	0.402	0.421	0.429
L5		5.28			0.208	
L6		2.35			0.094	
L7	2.85		2.8	0.104		0.114
S	1.8		2.6	0.071		0.102
S1	1.8		2.6	0.071		0.102
Dw1	3.05		2.05	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



L298

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.0			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.6	0.370		0.386
E	13.9		14.2	0.547		0.575
e		1.27			0.050	
a3		11.43			0.450	
E2 (1)	10.0		11.1	0.433		0.437
E2			2.0			0.114
E3	5.8		6.2	0.228		0.244
D	0		0.1	0.000		0.004
H	15.5		15.8	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
H	12 (max.)					
D	8 (max.)					
T		1.0			0.039	

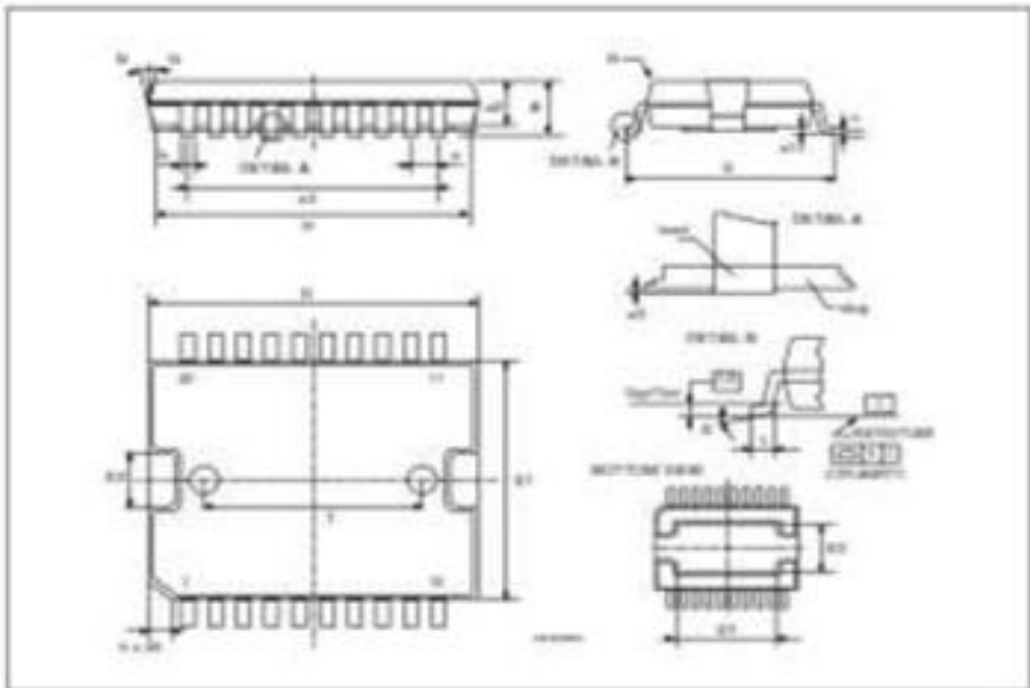
(1) "D" and "E" do not include mold flash or protrusions.
 Mold flash or protrusions shall not exceed 0.13 mm (0.005").
 Critical dimensions: "E", "D" and "a3"

OUTLINE AND MECHANICAL DATA



JEDEC MO-188

PowerSO20



12/13



BIBLIOGRAPHIE

Bibliographie

- [1] : CH 1- système automatisé- système bouclé (PRESENTATION DES SYSTEMES AUTOMATISES)
- [2] : Ch1 (Les systèmes automatisés et leur structure) cour de l'Architecture des Systèmes Automatisés. 2ème année Automatique (2018/2019) Univ Batna2.
- [3] : A. MTIBAA « Les Systèmes automatisés de Production », Université de Monastir Ecole Nationale d'Ingénieurs de Monastir, Tunisie
- [4] : S. BENDIDANI et A. MILOUD 'Etude d'un système automatisé (cas d'une serre), mémoire de Master en Electromécanique, Université de BELHADJ Bouchaïb d'Ain-Temouchent ,2018
- [5] : J.N Montagné, Centre de Ressources Art Sensitif, novembre 2006.
- [6] : W. Benlahcen, "Un éclairage redondant", mémoire de fin d'études pour d'ingénieur d'état en Instrumentation, Université Batna, 2007.
- [7] : N. Mechalaïne, "Etude et réalisation d'un système intelligent pour la commande d'éclairage publique", mémoire de master 2, Université Oum El Bouaghi, Juin 2018.
- [8] : F. Bouabdelli et H. Senouci, "Détecteur de distance à infrarouge", mémoire de master, Université Abou Bakr Belkaïd de Tlemcen, Juin 2018.
- [9] : I. Boudjedir, "Un système embarqué pour la détection des gaz dangereux à base d'une carte Arduino", mémoire de fin d'étude pour Master", Université Oum El Bouaghi, 2017.
- [10] : M. Iamrache et N.Bentoumi, "Etude et réalisation d'un bras robotique", mémoire de fin d'étude de master professionnel en électronique industriel, Université Mouloud Mammeri De Tizi-Ouzou, 2017 .
- [11] : M. Coulibaly, "Détection du complexe QRS par la Méthode de Laguna", Mémoire de l'obtention du diplôme de master, Université Badji Mokhtar Annaba, 2019

Sites Internet

<https://www.arduino.cc/>

www.alldatasheet.fr

