

الجمهورية الجزائرية الديمقراطية الشعبية

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان

**Université Aboubakr Belkaïd-Tlemcen**

كلية التكنولوجيا

Faculté de Technologie

**Département de Génie Electrique et Electronique (GEE) Filière : Electronique**



**MASTER INSTRUMENTATION**

**PROJET DE FIN D'ETUDES**

Présenté par Flitti Fatima & Hadbi Rahma

Intitulé du Sujet

**Réalisation d'un système de remplissage automatique de  
bouteilles en utilisant une carte Arduino**

Soutenu en 07/07/2021, devant le jury composé de :

**M<sup>r</sup>**MASSOUM Nourreddine

MCB

Univ. Tlemcen

Président

**M<sup>r</sup>**ZOUGAGH Nabil

MCB

Univ. Tlemcen

Examineur

**M<sup>r</sup>**BOUNATI Sidi Mohammed

MAA

Univ.Tlemcen

Encadrant

**Année Universitaire 2020-2021**

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

## **Remerciements**

*Nous remercions glorifions Allah le tout puissant et miséricordieux d'avoir guidé nos pas pour accomplir ce modeste travail et de le mené à terme. Ce dernier, a été Effectué au sein de l'Université Abou Bakr Belkaid de Tlemcen et plus précisément au département de Génie Electrique et Electronique de la Faculté de Technologie.*

*Nous tenons à remercier vivement nos encadreurs, Mr BOUANATI Sidi Mohammed, d'avoir accepté de nous guider tous le long de ce travail. Et cela, avec leur grande disponibilité, leur rigueur scientifique et leurs précieux conseils qui nous ont permis de travailler dans les meilleures conditions, et ce, depuis le début de la préparation de notre projet.*

*Nous tenons à remercier, aussi, Mr MASSOUM Nourredine pour l'honneur, de bien vouloir accepter de présider le jury de notre soutenance.*

*Les remerciements sont, également, adressés à Mr ZOUGAGH Nabil. Monsieur, nous sommes très sensibles à l'honneur que vous nous faites en acceptant d'examiner le présent travail que nous souhaitons est d'un bon niveau,*

*Par ailleurs, trouverons ici, nos sincères reconnaissances à toute personne ayant contribué de près Ou de loin à la réalisation de ce projet de fin d'étude.*

# DEDICACE

**Au nom du tout puissant**

*Je dédie ce modeste travail :*

*A mes très chers parents. Ce travail est le fruit de leurs soutiens depuis toujours, leurs présences à mes cotes et leurs affections. A mes frères : Samir et Mohamed.*

*A mes sœurs : Nawel et Amina. A ma belle-sœur : Nawel.*

*A mes neveux : Houdeyfa et Maissa. A mes amies : Samia et Leila.*

*A mon binôme Rahma et sa famille.*

*A mes professeurs qui m'ont enseigné et qui par leurs compétences M'ont aidé dans la poursuite de mes études.*

*A tous mes amis de la promotion Instrumentation  
2020/2021*

*Fatima*

## Dédicace

Je dédie ce travail à mes chers parents," mon père" et "Ma Mère "qui M'a soutenu avec leurs Doua ASDSQ. Et aider Tout au long de mon parcours, sans eux je ne serai jamais Arrivé là où je suis.

A mon frère Mohammed qui a toujours été là pour moi avec toutes les Moyennes possibles

À Mes très chères sœurs (Asma, Loubna et Ratiba)

À Mes nièces (ines, Yousra, Kawther, Khalil et iYeD) que j'adore

A tout la famille

A mon binôme FLITTI Fatima et sa famille.

A mes amis qui ont étudié à mes côtés et à mes amis

A tous mes amis de la promotion Instrumentation  
2020/2021

**Rahma**

## Liste des abréviations

API : Automate Programmable Industriel.  
PLC : Programmable Logic Controller.  
CPU : Central Processing Unit.  
PS : Power Supplay.  
E/S : Entrée/sortie.  
PC : Personal Computer.  
  
LD : Ladder.  
FBD : Function Block Diagram.  
IL : Instruction List.  
ST :Structured Text.  
SFC : Sequential Function Charts.  
Grafcet : GRA : graphe.  
F : fonctionnel.  
C : commande.  
E : Etape.  
T : Transition.  
TIA PORTAL : Totally Integrated Automation.  
MHz : Megahertz.  
USB : Universal Serial Bus.  
Ko :kilooctet.  
RAM : Random Access Memory.  
EEPROM : Electrically Erasable Programmable Read-Only Memory.  
  
A/N : Analogique/numérique.  
PWM : Pulse Width Modulation.  
RX : Ordonnance ou Traitement.  
TX : Emission ou Transmission. Wifi : Wireless Fidelity.  
IDE : Integrated Development Environment.  
IR : Infrarouge.  
LED : Light Emitting Diode.  
K : Kilo.

VCC : Vin de Consommation Courante.

GND : *Ground*.

LCD : Liquid Cristal Display.

CC : Courant Continu.

DC :

MOSFET : Métal Oxide Semi-conducteur Field Affect Transistor

ISIS : Intelligent Schematic Input System.

NO : Normal Open.

NC : Normal Close.

ADC : convertisseurs analogiques numériques.

## Table de matières

Titre	N° page
Liste des abréviations.....	6
Liste des figures.....	11
Liste des tableaux.....	13
Introduction générale.....	15
<i>Chapitre I : Généralités sur les API</i>	
I.1 Introduction.....	18
I.2 Définition des automates programmables industriels.....	18
I.3 Les avantages d'API.....	19
I.4. Composants des API.....	19
I.4.1 Structure interne d'un API.....	19
I.4.1.1 Module d'alimentation.....	20
I.4.1.2 Module de CPU (Central Processing Unit).....	20
I.4.1.3 Module d'entrées/sorties.....	20
I.4.1.4 Outil de programmation d'un automate programmable industriel.....	21
I.4.2 Aspect extérieur d'un API.....	22
I.4.2.1 Différence entre un API compact et modulaire.....	23
I.5 Principe de fonctionnement d'un automate programmable industriel.....	24
I.6 Programmation d'un API.....	25
I.6.1 Ladder (langage contact).....	26
I.6.2 Logigramme ou FBD (Function Block Diagram).....	27
I.6.3 Instruction List (IL).....	27
I.6.4 Structured ST (structured text).....	28
I.6.5 Grafsetou SFC.....	29
I.7 TIA PORTAL (Totally Integrated Automation).....	30
I.8 Conclusion .....	30
<i>Chapitre II : Description des différents composants utilisés pour la réalisation du projet</i>	
II.1 Introduction.....	32
II.2 La carte Arduino.....	32
II.2.1 Description de la carte Arduino UNO.....	32
II.2.2 Bloc d'alimentation.....	32

II.2.3 Microcontrôleur.....	33
II.2.4 Entrées numériques.....	34
II.2.5 Entrées analogiques.....	35
II.2.6L'environnement de la programmation (IDEARDUINO).....	35
II.3 Le capteur infrarouge (IR).....	37
II.3.1Principe de fonctionnement du capteur infrarouge.....	37
II.3.2 Câblage du module IR avec Arduino UNO.....	39
II.4 Les Afficheurs LCD.....	40
II.5 Relais électrique.....	40
II.6 Electrovanne.....	41
II.7 Convoyeur.....	43
II.8 Moteur à courant continu .....	44
II.9Pompe d'eau.....	45
II.10Transistor à effet de champ(MOSFET).....	45
II.11 La diode.....	46
II.12 Résistance.....	47
II.13 Conclusion.....	48
<b><i>Chapitre III: Simulation et réalisation de circuit</i></b>	
III.1Introduction.....	50
III.2LapartieSimulation.....	51
III.2.1 Présentation de PROTEUS.....	51
III.2.1.1 Présentation du module ISIS (Intelligent Schematic Input System).....	51
III.2.1.2 Présentation du module ARES.....	51
III.3 Schémas et méthodologie suivie.....	52
III.3.1 Description du proto type.....	52
III.3.1.1 Bloc de détection.....	52
III.3.1.2 Bloc de convoyeur.....	53
III.3.1.3 Bloc d'action.....	54
III.4 Circuit global.....	55
III.5 Partie réalisation.....	56
III.6 Conclusion.....	62
Conclusion générale.....	63
Références.....	65

Annexe .....	<b>69</b>
Résumé.....	<b>71</b>
Abstract.....	<b>71</b>
الملخص .....	<b>71</b>

## Listes des figures

Titre	N° page
<i>Chapitre I</i>	
<b>Figure I.1.</b> Automate programmable industriel.....	18
<b>Figure I. 2.</b> Un automate programmable industriel.....	19
<b>Figure I.3.</b> Module d'alimentation d'un API.....	20
<b>Figure I.4.</b> Module de CPU d'un API.....	21
<b>Figure I.5.</b> Les interfaces d'entrées/sorties.....	21
<b>Figure I.6.</b> Outil de programmation.....	22
<b>Figure I.7.</b> Automate modulaire.....	23
<b>Figure I.8.</b> Automate fixe.....	23
<b>Figure I.9</b> fonctionnement d'un automate programmable industriel.....	25
<b>Figure I.10.</b> Cycle d'exécution d'un API.....	25
<b>Figure I.11.</b> Principaux langages de programmation des API.....	26
<b>Figure I.12.</b> Langage Ladder.....	27
<b>Figure I.13.</b> Langage Logigramme.....	27
<b>Figure I.14.</b> Langage IL.....	28
<b>Figure I.15.</b> Langage ST.....	29
<b>Figure I.16.</b> Un Graf cet SFC.....	30
<i>Chapitres II</i>	
<b>Figure II.1 :</b> Descriptions de la carte Arduino UNO.....	32
<b>Figure II.2 :</b> Bloc d'alimentation de la carte Arduino.....	33
<b>Figure II.3:</b> MicrocontrôleurATMega328 .....	34
<b>Figure II.4 :</b> Broches numériques de la carte Arduino UNO.....	35
<b>Figure II.5 :</b> Broches d'entrées Analogiques.....	35
<b>Figure II.6 :</b> Structure générale d'IDE Arduino.....	36
<b>Figure II.7 :</b> Capteur Infrarouge.....	37
<b>Figure II.8 :</b> Principe de fonctionnement d'IR.....	38
<b>Figure II.9 :</b> schéma du circuit de capteur IR.....	39
<b>Figure II.10 :</b> branchement d'Arduino avec capteur IR.....	39
<b>Figure II.11 :</b> Afficheur LCD 16*2.....	40
<b>Figure II.12 :</b> relais électrique et ses symboles.....	41

<b>Figure II.13</b> : les constitutions de relais.....	<b>41</b>
<b>Figure II.14</b> :Electrovanne.....	<b>42</b>
<b>Figure II.15</b> : principe de fonctionnement de l'électrovanne.....	<b>43</b>
<b>Figure II.16</b> :Convoyeur.....	<b>43</b>
<b>Figure II.17</b> : Moteur à courant continu.....	<b>44</b>
<b>Figure II.18</b> : Une pompe d'essuie-glace.....	<b>45</b>
<b>Figure II.19</b> :Transistor MOSFET.....	<b>45</b>
<b>Figure II.20</b> :diode1N400.....	<b>46</b>
<b>Figure II.21</b> :Résistance électrique.....	<b>47</b>

### *Chapitre III*

<b>Figure III.1</b> : Représentation schématique du prototype.....	<b>49</b>
<b>Figure III.2</b> : Le branchement du module IR avec la carte Arduino UNO sous PROTEUS.....	<b>51</b>
<b>Figure III.3</b> : Branchement du bloc convoyeur sous ISIS.....	<b>52</b>
<b>Figure III.4</b> : Branchement du bloc d'action sous ISIS.....	<b>53</b>
<b>Figure III.5</b> : Aperçu globale du notre prototype sous ISIS.....	<b>54</b>
<b>Figure III.6</b> : La mise en marche du système.....	<b>54</b>
<b>Figure III.7</b> : Affichage du message «welcome».....	<b>55</b>
<b>Figure III.8</b> : Circuit de contrôle de vitesse du moteur.....	<b>55</b>
<b>Figure III.9</b> : Circuit de contrôle de vitesse du moteur sous ISIS.....	<b>56</b>
<b>Figure III.10</b> : Affichage du message « bottle notd et ected».....	<b>56</b>
<b>Figure III.11</b> :Bouteille détecté.....	<b>57</b>
<b>Figure III.12</b> : l'activation de 57relais.....	<b>57</b>
<b>Figure III.13</b> : Message « bottle filled».....	<b>58</b>
<b>Figure III.14</b> : Prototype de remplissage des bouteilles « la face avant».....	<b>57</b>
<b>Figure III.15</b> : Prototype de remplissage des bouteilles « la face arrière».....	<b>59</b>
<b>Figure III.16</b> : Représentation schématique du prototype.....	<b>59</b>
<b>Figure III.17</b> : grafc et de notreproto type.....	<b>60</b>

## Liste des tableaux

Titre	N° page
<b>Tableau I.1</b> :la différence entre l'API modulaire et API compact .....	<b>23</b>
<b>Tableau II.1</b> : les caractéristiques techniques du capteur infrarouge.....	<b>37</b>
<b>Tableau III.1</b> : branchement d'électrovanne et pompe avec Arduino.....	<b>53</b>

# *Introduction générale*

Ces dernières années, l'utilisation des automates programmables industriels est omniprésente dans les installations automatisées des machines pour optimiser la productivité [1].

Dans le domaine de l'industrie, un système automatisé est composé d'une partie commande (PC) et d'une partie opérative (PO), pour assurer le bon fonctionnement de la chaîne de production [2].

Ces systèmes permettent de réduire les coûts de la production. Ils permettent ainsi d'effectuer des tâches difficiles et répétitives (ligne de montage) à la place de l'opérateur telles que le remplissage des bouteilles et l'emballage des produits [3].

Cependant, il est possible d'accomplir de telles tâches en utilisant des modules électroniques (Arduino UNO, capteur IR...etc.). C'est dans ce contexte que notre projet s'intervient.

En effet, le but de ce travail est de réaliser un système de remplissage des bouteilles automatique en utilisant une carte Arduino UNO.

Notre projet de fin d'étude est réparti en trois chapitres :

➤ En premier lieu, nous présentons les automates programmables industriels (API) et ses composants. Dans la deuxième partie de ce chapitre nous citons le principe de fonctionnement et les différents langages de programmation d'un API.

➤ Le deuxième chapitre est consacré à la description des composants utilisés pour la réalisation des différents étages qui constituent notre prototype.

➤ Enfin, le dernier chapitre représente la partie la plus intéressante de notre mémoire. Elle englobera la partie simulation et la partie réalisation de notre système.

# *Chapitre I*

## *Généralités sur les automates programmables industriels*

## I.1 Introduction

L'automatisation des systèmes de production est développée pour réduire le coût et la complexité de l'installation des machines, pour minimiser l'intervention de l'homme dans le processus de fabrication et d'optimiser la productivité [1].

L'objectif de ce chapitre est de donner un aperçu global sur les systèmes d'API et leur principe de fonctionnement.

## I.2 Définition des automates programmables industriels

Un automate programmable industriel-API (en Anglais Programmable Logic Controller- PLC) est une forme spécifique de contrôleur à microprocesseur qui utilise une mémoire programmable pour sauvegarder les instructions et qui exécute diverses fonctions, qu'elles soient logiques de séquençement, de temporisation, de comptage ou arithmétique pour commander les machines et automatiser les processus [2].

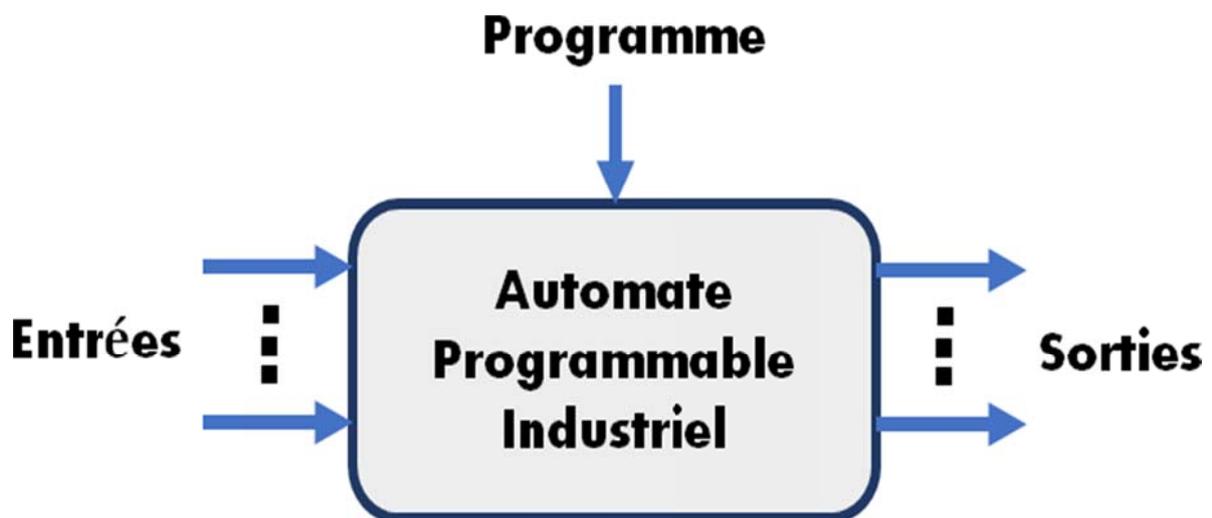


Figure I.1. Automate programmable industriel

Les automates programmables industriels sont similaires aux ordinateurs. Ils permettent de piloter les procédés industriels en temps réel [4].

Outre la résolution de problèmes de câblage, les systèmes d'API nous offrent la

---

possibilité d'effectuer plusieurs opérations telles que : le comptage, la comparaison et le traitement de signaux analogiques [4].

### **I.3 Les avantages d'API**

L'utilisation de l'automate programmable dans le secteur de l'industrie présente plusieurs avantages.

- Le dépannage et la maintenance des éventuelles pannes par des techniciens de formation électromécanique [1].
- Une meilleure fiabilité [1].
- Les langages de programmation des API sont simples et faciles à comprendre et surtout orienté sur les opérations de communications et les opérations logiques [2].
- Simplification du câblage
- Le prix de revient est très intéressant, surtout comparé à une solution à base du microprocesseur [5].

### **I.4 Composants des API**

Les automates programmables industriels comportent deux structures, une structure interne et une structure externe.

#### **I.4.1 Structure interne d'un API**

Un automate programmable se compose de quatre parties :

- Module d'alimentation.
- Module de CPU.
- Module d'entrées/sorties.
- Outil de programmation.

Ces modules sont disposés et arrangés l'un à côté des autres [6].

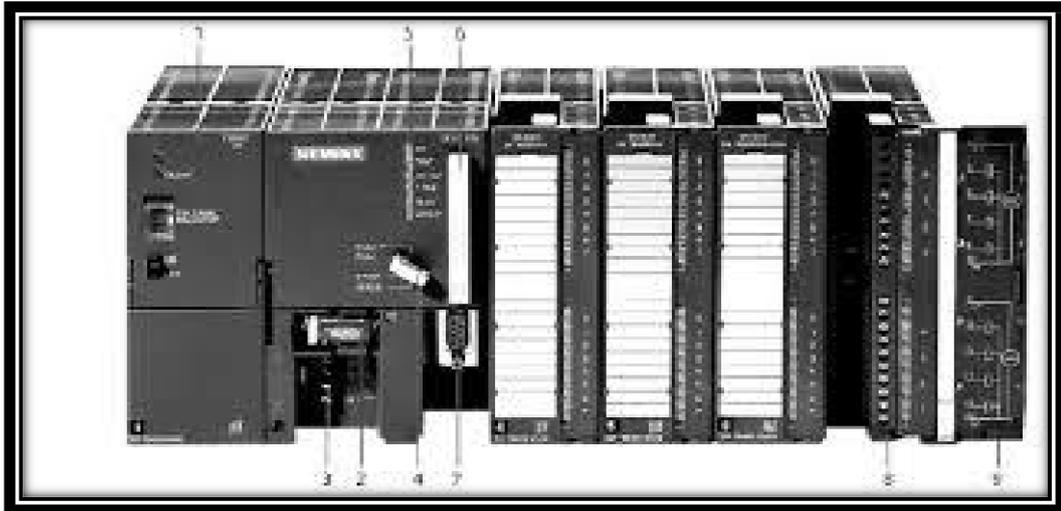


Figure I. 2. Un automate programmable industriel [7].

#### I.4.1.1 Module d'alimentation

L'automate est généralement alimenté par une tension alternative de 220V-50Hz. Cette unité permet de convertir cette tension d'entrée en une tension continue de 24V afin d'alimenter le module de CPU et les modules d'entrées et de sorties [2].

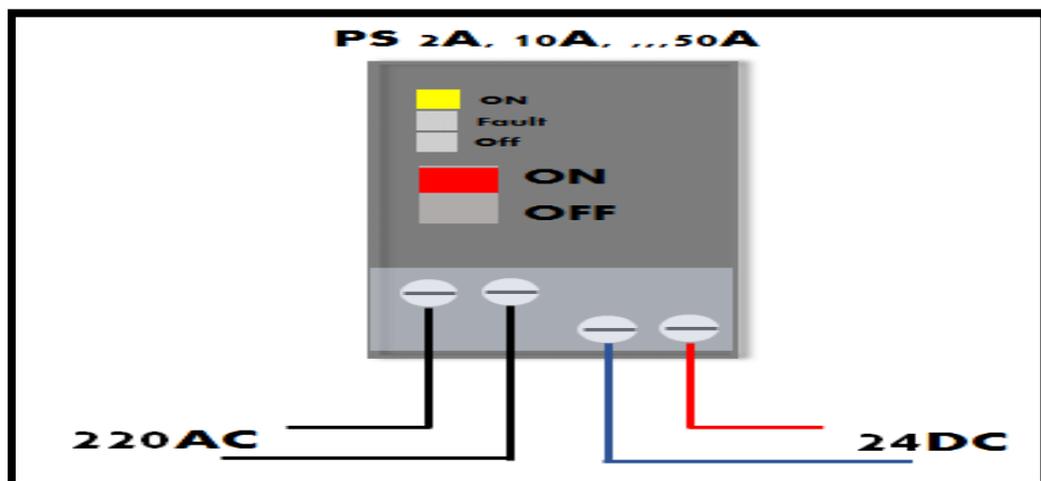


Figure I.3. Module d'alimentation d'un API [6].

#### I.4.1.2 Module de CPU (Central Processing Unit)

L'élément principal de l'unité CPU est le microcontrôleur. Ce module permet d'interpréter les signaux d'entrées et d'exécuter les actions de commande suivant le

programme enregistré en mémoire, en communiquant les décisions sous forme des signaux d'actions [6].

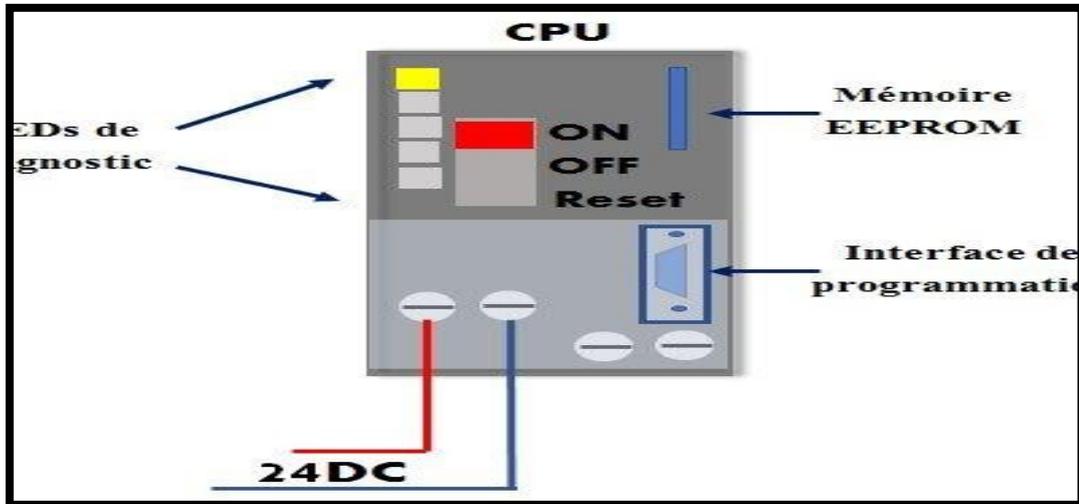


Figure I.4. Module de CPU d'un API [6]

### I.4.1.3 Module d'entrées/sorties

Le module d'entrées-sorties joue le rôle d'intermédiaire entre le système et le monde extérieur. Il permet d'établir des connexions avec des éléments d'entrées (les capteurs) et les éléments de sorties tels que les moteurs, les électrovannes... etc[2].

Ainsi, à travers ce module que se fait la saisie des instructions de programme depuis un périphérique d'entrée ou un terminal [2].

L'interface d'entrée contient des adresses d'entrées, dont chaque élément d'entrée est connecté à une de ces adresses [2].

De la même façon, l'interface de sortie contient des adresses de sorties, dont chaque sortie est connectée à une de ces adresses [2].

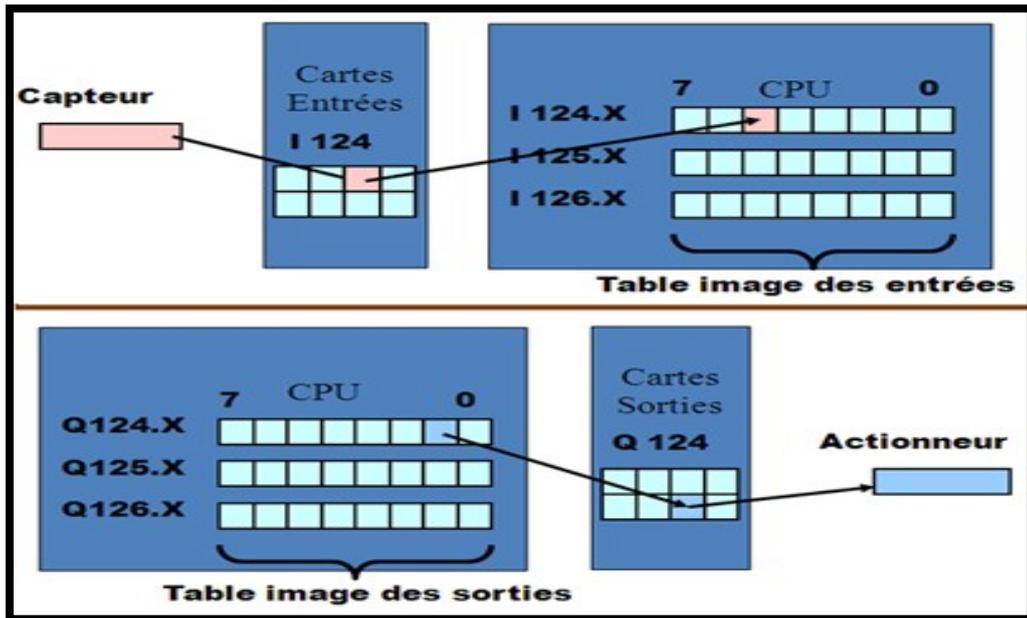


Figure I.5. Les interfaces d'entrées/ sorties [6].

#### I.4.1.4 Outil de programmation d'un automate programmable industriel

L'ordinateur représente l'outil le plus utilisable pour la programmation des automates.

Tous les constructeurs d'API possèdent leur propre logiciel fourni avec l'automate, ce qui permet de modifier, d'éditer, et télécharger le programme de l'automate programmable industriel [4].

Aussi, Sur un PC, on va avoir une plus grande visibilité sur l'exécution du programme [4].

La communication entre un API et un PC peut être assurée (garantie) via un adaptateur RS232/USB ou via une liaison réseau Ethernet [4].

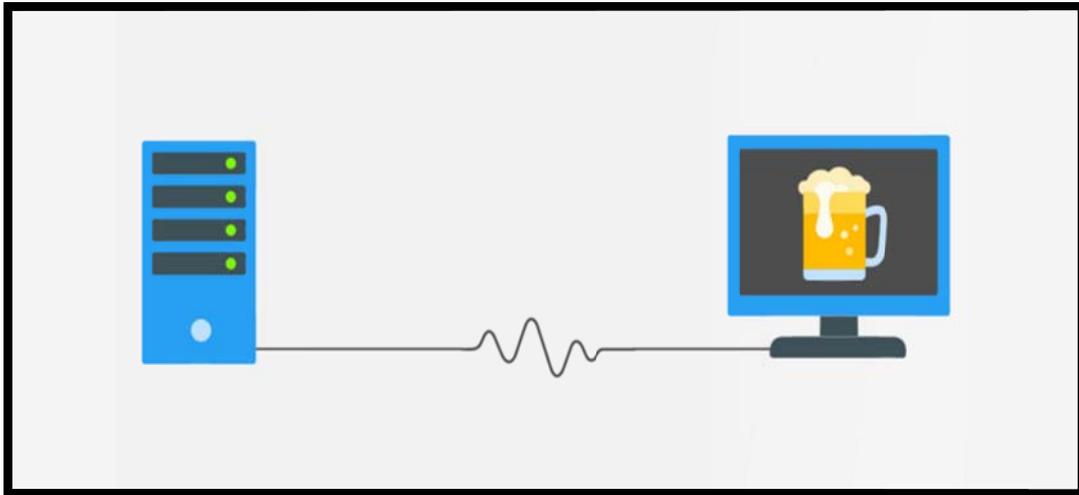


Figure I.6. Outil de programmation [8].

#### I.4.2 Aspect extérieur d'un API

Il existe deux types de configuration des API :

La configuration fixe est la configuration dont les entrées-sorties sont fixes et packagés avec la CPU. Donc le nombre des automates est fixe selon le type d'automate [4].

Dans la configuration modulaire, l'automate est divisé en plusieurs modules qui viennent se clipser sur ce qu'on appelle un rack [4].



Figure I.7. Automate modulaire [9].



Figure I.8. Automate fixe [10].

##### I.4.2.1 Différences entre un API compact et modulaire [11]

Pour mieux comprendre ces configurations nous avons présenté la différence entre ces

deux types d'automates dans le tableau (I.1)

Contenu	API compact	API modulaire
Nom	L'API compact est nommé aussi <b>API fixe</b>	L'API modulaire est nommée aussi <b>API monté en rack</b>
E/S	Le nombre des entrées et sorties dans l'API compact est fixe. Car les capacités d'E/S sont décidées par le constructeur mais pas par l'utilisateur.	Le nombre des entrées et sorties n'est pas fixe. Des entrées et des sorties peuvent être supprimées ou ajoutées aux systèmes API modulaires par l'utilisateur.
Réparation	La réparation d'API compact n'est pas facile.	L'entretien et la réparation d'API modulaire est facile par rapport à l'API compact
Mémoire	Il a moins de capacités de mémoire pour le stockage des données.	Il a plus de mémoire et de capacités pour le stockage des informations
utilisation	Il est utile pour les petites applications.	Il est utilisable pour des fins industrielles et pour la croissance industrielle future.
taille	Il est petit de taille	Il se produit avec une grande taille
Coût	Il a un faible coût. Un modèle économique	Il est plus coûteux que Compact PLC.

**Tableau I.1** La différence entre l'API modulaire et API compact

## **I.5 Principe de fonctionnement d'un automate programmable industriel**

Dans le secteur d'industrie, il existe plusieurs marques de l'API. Nous citons quelques-unes : **SIMENS, Schneider Electrique, Allen-Bradley, Omran, Crouzet, MITSUBISHI, VIGOR, WAGO....** [12]

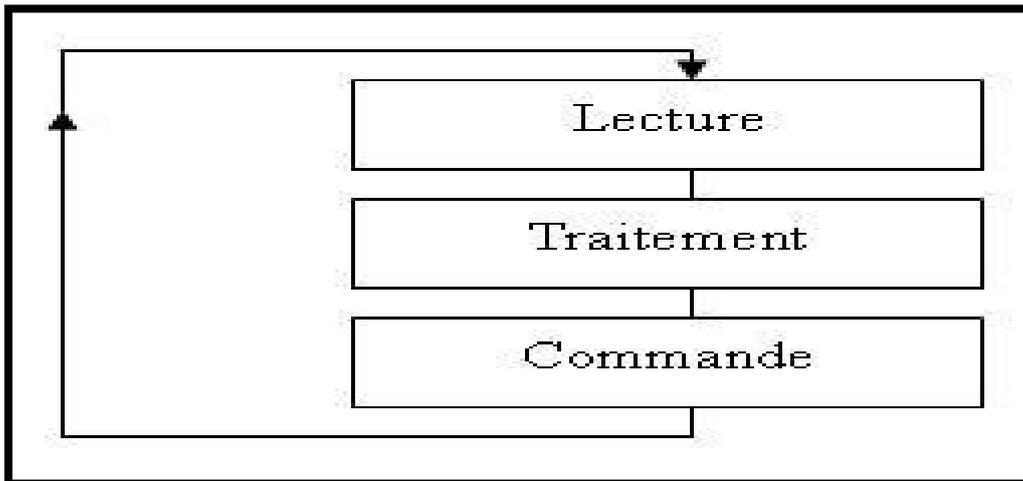
La marque d'automate reçoit des informations et des données via ses entrées [13].

Ensuite, ces données sont enregistrées et stockées dans la mémoire du CPU. Par la suite, le traitement des données est effectué selon le programme élaborer par l'opérateur [13].

Finalement, les résultats obtenus sont libérés par les sorties de l'automate [13].

La commande de sortie ne se fait pas directement via la sortie mais plutôt par un contacteur.

Ce contacteur permet la protection de la sortie d'automate [13].

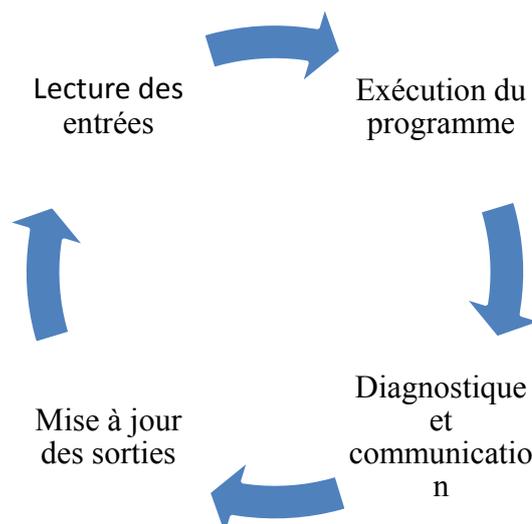


**Figure I.9** Fonctionnement d'un automate programmable industriel [14].

La durée d'un cycle d'API dépend de la taille du programme et la puissance du programme.

Ce cycle ne change pas, il reste toujours le même [13].

L'élément essentiel dans un automate programmable est l'unité centrale car c'est elle qui reçoit, stocke, mémorise et traite les informations et les données entrantes, et détermine l'état des informations sortantes en fonctions du programme établi [13].



**Figure I.10.** Cycle d'exécution d'un API.

## I.6 Programmation d'un API

Dans les systèmes à microprocesseur, les programmes sont chargés sous forme de code machine, c'est-à-dire sous forme d'une série de nombres binaires qui représentent des instructions [15].

Les 5 types de langages de programmation API les plus connues sont illustrés dans la figure (II.11) :

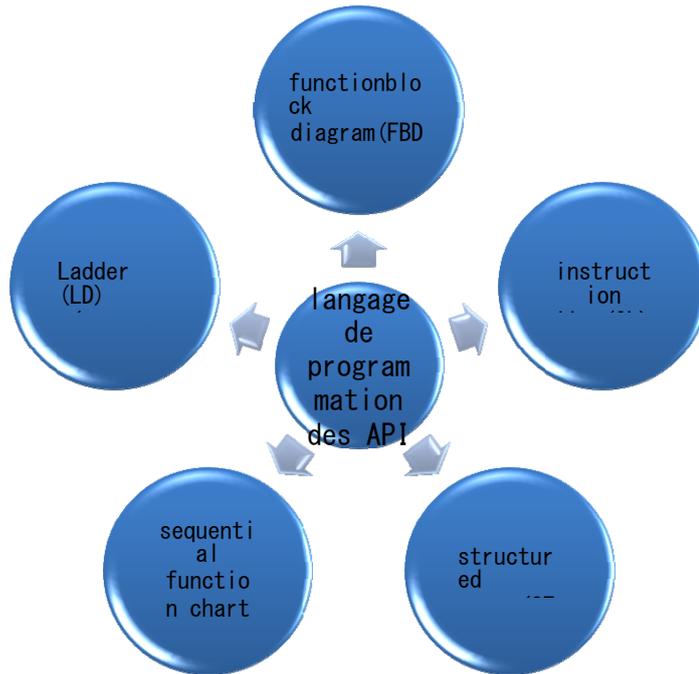


Figure I.11. Principaux langages de programmation des API.

### I.6.1 Ladder (langage contact)

Langage Ladder, aussi appelé langage à relais ou à réseau en échelle ou bien langage à contact est utilisé pour programmer des fonctions logiques dans un API [16-17].

Pour programmer ses fonctions, il faut juste copier et transcrire les équations logiques sous formes des schémas électriques à l'aide de symboles placés [17].

Les symboles graphiques utilisés dans Ladder sont : relais, bobine, contacts, des blocs fonctionnels organisé en réseaux [17].

Ce langage est composé deux barres verticales qui représentent les lignes d'alimentations et de réseaux [18].

Chaque réseau correspond à une équation logique. Cette équation permet le calcul

d'un résultat en fonction de signaux d'entrées. Ses signaux sont représentés par un contact(interrupteur) normalement fermé/ normalement ouvert. Le résultat obtenu est représenté par une bobine [18].

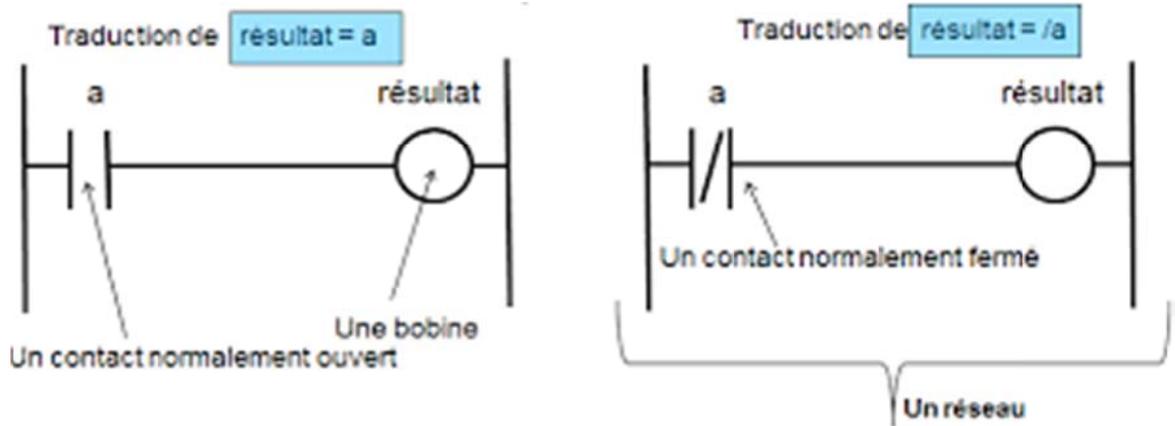


Figure I.12. Langage Ladder [18].

### I.6.2 Logigramme ou FBD (Function BlockDiagram)

FBD est un langage de programmation orienté graphiquement. Il fonctionne avec une liste de réseaux. Chaque réseau se compose d'une structure graphique constituée de boîtes et de lignes de connexion représentant une expression arithmétique ou logique, un appel de bloc fonctionnel, un saut ou une instruction de retour [19].

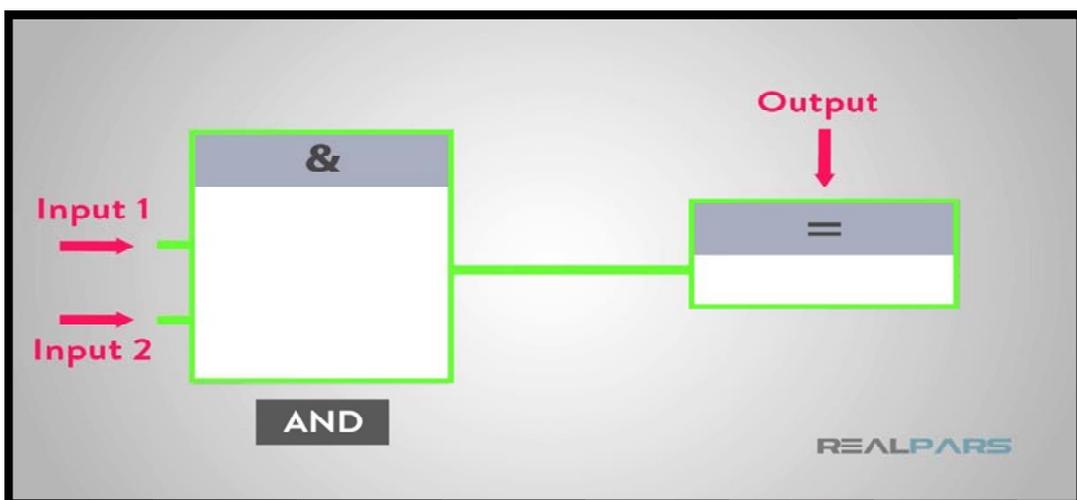


Figure I.13. Langage Logigramme. [20]

### I.6.3 Instruction List(IL)

L'IL est un langage textuel, non graphique, il ressemble au programme du langage assembleur [22].

La liste d'instructions se constitue d'une suite d'instructions. Chaque instruction commence par une nouvelle ligne et contient un opérateur et selon le type d'opération, un ou plusieurs opérandes séparés par des virgules. Devant une instruction, il peut y avoir une marque d'identification (étiquette) suivie de deux points (:). [23].

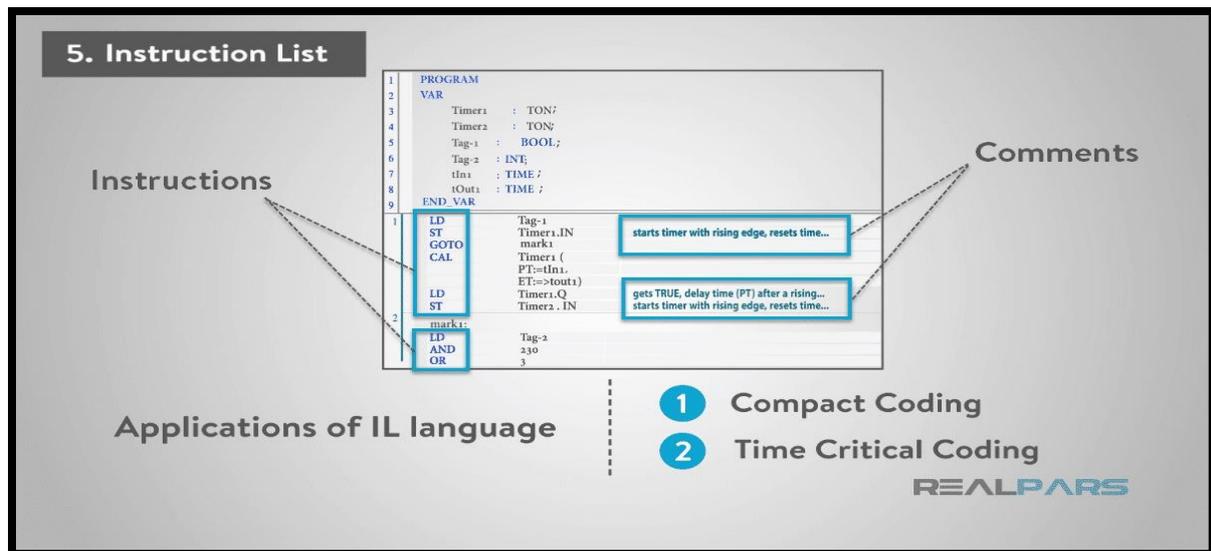


Figure I.14. Langage IL [24]

### I.6.4 Structured ST (structured text)

Le quatrième langage de programmation d'automate programmable industriel est le ST [24]. Ce langage est plus haut niveau qui ressemble à pascal, C. Il exécute des tâches complexes, et des calculs mathématiques. Il traite les chaînes de caractères[25].

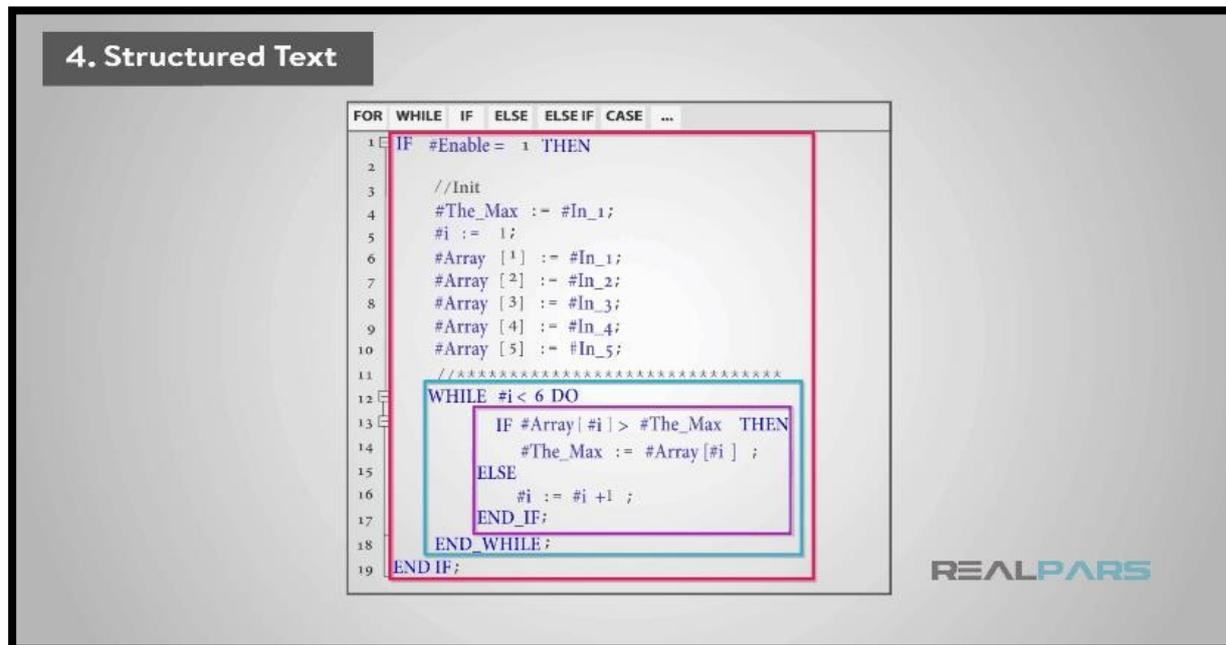


Figure I.15. Langage ST [24]

Généralement, il tend à remplacer le grafcet et le Ladder [25].

### I.6.5 Grafcet ou SFC[26]

Le grafcet est un langage fonctionnel graphique utilisé pour la description des opérations et des comportements des automatismes séquentiels.

Le mot grafcet signifie :

GRA : graphe. F : fonctionnel. C : commande. E : Etape.

T : Transition.

Le procédé est représenté comme une suite connue d'étapes, connectées entre elles via des transitions, une condition booléenne est liée à chaque transition. Les actions dans les étapes sont décrites avec ST, IL, LD, ou FBD :

- Un programme SFC doit contenir au moins une étape initiale.
- Une étape ne peut pas être suivie d'une autre étape.
- Une transition ne peut pas être suivie d'une autre transition.

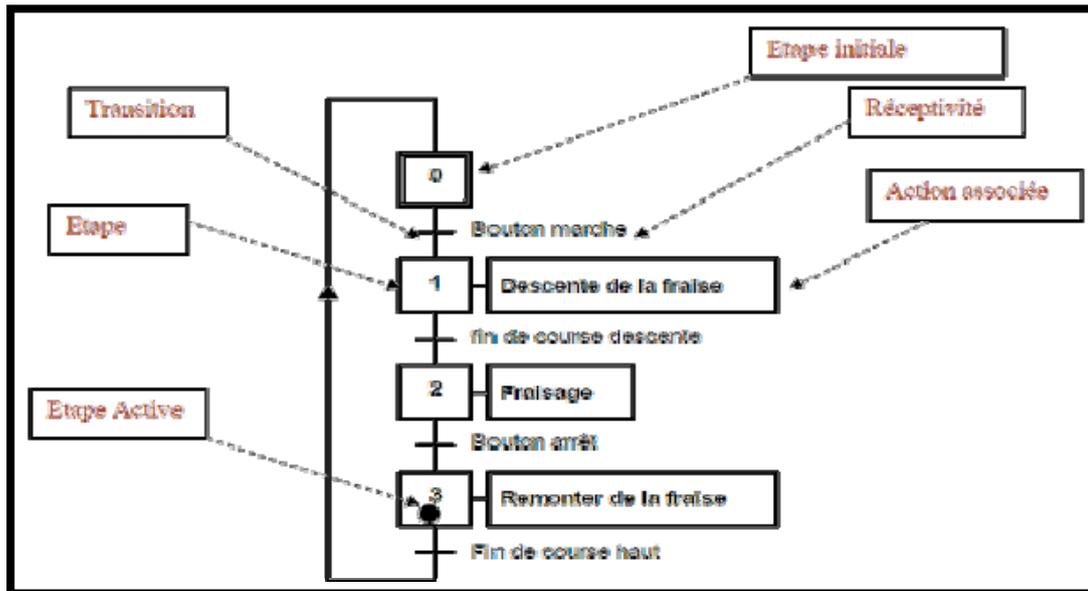


Figure I.16. Un Grafcet SFC [27]

## I.7 TIA PORTAL (Totally Integrated Automation)

TIA Portal (Totally Integrated Automation) est un logiciel tout en un qui permet de programmer des automates. Cette plate-forme est le nouvel environnement de travail siemens qui permet la mise en œuvre des solutions d'automatisation avec un système d'ingénierie intégré comprenant les logiciels SIMATIC STEP7 et SIMATIC WinnCC [26].

## I.8 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur les automates programmables industriels, leur principe de fonctionnement, ainsi que leurs langages.

L'automate programmable industriel est aujourd'hui le dispositif le plus répandu pour réaliser des automatismes. On le trouve pratiquement dans tous les secteurs de l'industrie car il répond aux besoins d'adaptation pour un grand nombre d'opérations[28].

Ce système d'automate programmable industriel est omni présent dans la réalisation des machines industrielles.

Cependant, il est possible de faire la conception de ces systèmes en utilisant les circuits électroniques.

*in*

## *Chapitre II*

*Description des différents  
composants utilisés pour la  
réalisation du projet*

## II.1 Introduction

L'objectif de ce chapitre est de présenter les dispositifs que nous avons utilisés dans notre réalisation du projet, ainsi que leur principe de fonctionnement.

## II.2 La carte Arduino

Il existe plusieurs types de carte Arduino à savoir carte Arduino MEGA, carte Arduino Bluetooth et carte Arduino UNO. Cette dernière est utilisée pour la réalisation de notre projet [29].

### II.2.1 Description de la carte Arduino UNO

La carte Arduino UNO est constituée de plusieurs éléments tels que, le régulateur linéaire 5V, un oscillateur Quartz 16 MHz, et un convertisseur analogique numérique. Cependant, l'élément principal de ce module est le Microcontrôleur AtmelAVR[29].

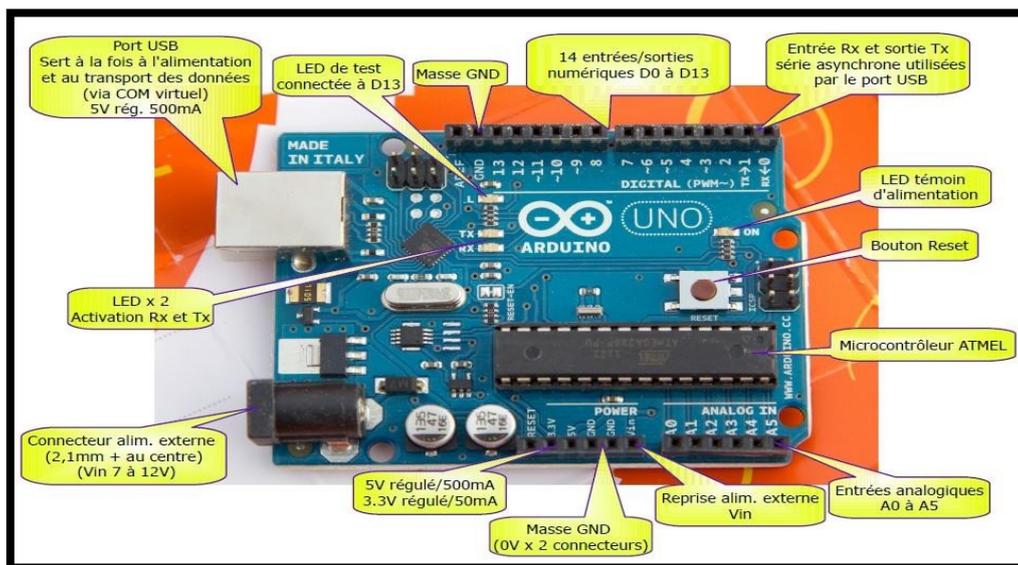


Figure II.1 Description de la carte Arduino UNO [30]

### II.2.2 Bloc d'alimentation

Afin de fonctionner, le module Arduino UNO doit être alimenté par une tension continue externe [29].

En effet, il est possible d'alimenter la carte Arduino soit par un ordinateur via le port USB en utilisant un câble. Ou bien via une source d'alimentation externe qui génère une tension continue varie entre 7V et 12V (exemple : batterie 9V). Par la suite, le régulateur réduit cette tension à 5V pour assurer le bon fonctionnement de l'Arduino UNO [29].

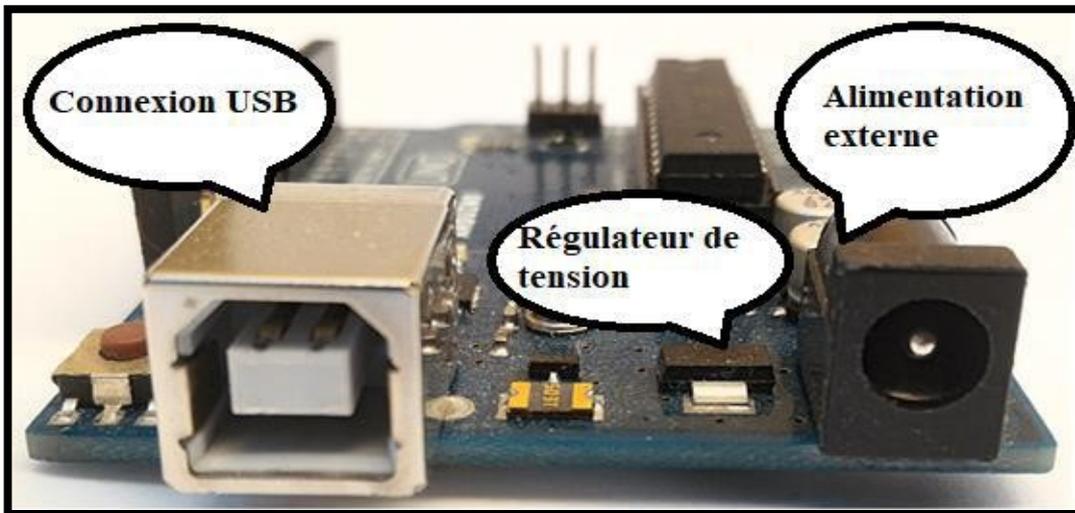


Figure II.2 Bloc d'alimentation de la carte Arduino [29]

### II.2.3 Microcontrôleur

Le microcontrôleur (ATMEGA328) représente l'élément principal de la carte Arduino UNO, c'est un circuit intégré ATMEL de la famille AVR 8 bits [29].

Le microcontrôleur ATMEGA328 embarque plusieurs éléments complexes dont chacun possède une fonction bien déterminé [29].

En effet, ces éléments sont : le processeur central qui permet l'exécution du programme dans une mémoire Flash, effaçable et réinscriptible de 32Ko et une mémoire (EEPROM) de 1Ko qui permet l'enregistrement des données et le stockage des informations, et une mémoire volatile(RAM) de 2Ko pour sauvegarder les variables et les données créer par le programme réalisé en langage C.

Aussi, la puce (ATMEGA328) est constituée de plusieurs ressources auxiliaires telles que les convertisseurs analogiques numériques(ADC) pour l'analyse et le traitement des signaux analogiques, les broches d'entrées/sorties séries et parallèles et les Timers [29].

Ces 23 I/O broches permettent d'établir une liaison entre le microcontrôleur et les différents dispositifs électroniques [29].

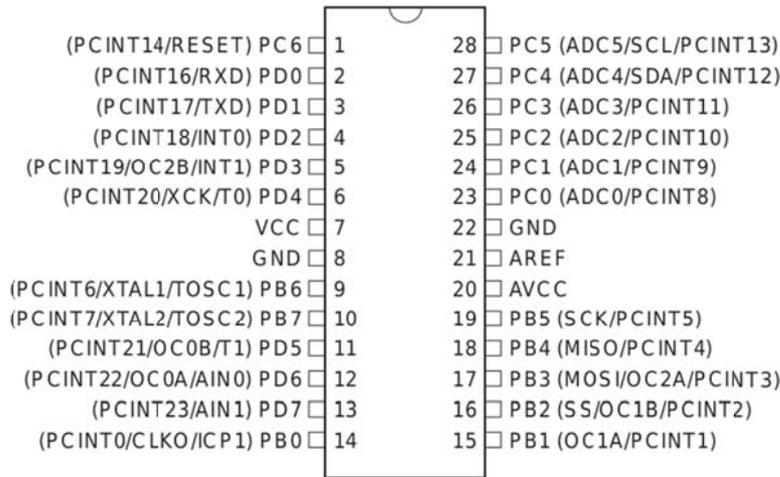


Figure II.3 Microcontrôleur ATM EGA 328 [29]

## II.2.4 Entrées numériques

La carte Arduino UNO dispose de 14 broches numériques (D0 à D13). Ces derniers peuvent être configurées via la programmation sous l'environnement Arduino IDE soit comme des entrées ou bien comme des sorties [29].

Ces broches numériques peuvent prendre deux niveaux logiques : niveau HAUT, et niveau BAS.

Autrement dit, lors de l'activation de la broche, la tension générée est de 5V. cependant, dans le cas inverse la tension présente est de 0V [29].

Ainsi, les broches (3,5,6,9,10,11) permettent de générer des signaux de type PWM.

Chaque broche peut fournir ou recevoir un maximum de 40 mA d'intensité et possède une résistance interne qui varie entre (20-50 K $\Omega$ ) [29].

Finalement, les deux broches RX et TX représentent respectivement la ligne de réception série et de transmission série des données. Ces lignes sont utilisées pour la communication avec d'autres modules électroniques tels que le module Bluetooth et le module wifi [29].



Figure II.4 Broches numériques de la carte Arduino UNO [29]

### II.2.5 Entrées analogiques

L'Arduino UNO dispose de 6 broches analogiques (A0 à A5). Ces entrées peuvent fournir une mesure de tension, que varie entre 0V et 5V avec une résolution de 10 bits (c.à.d. sur 1024 niveaux soit de 0 à 1023) [29].

Cependant, le courant qui passe à travers ces entrées est très faible. Ceci est dû à la présence d'une résistance interne importante [29].

Ainsi, il est possible de configurer ces entrées analogiques comme des broches numériques [29].



Figure II.5 : Broches d'entrées Analogiques [29]

### II.2.6 L'environnement de la programmation (ArduinoIDE)

L'environnement de développement Arduino IDE est un logiciel utilisé pour interpréter et compiler des codes pour développer des programmes utilisés sur un plateau Arduino. Il est compatible avec Windows et avec Linux. Donc, sa polyvalence est un point de

plus en faveur de ces appareils pour les projets électroniques [31].

Cet environnement contient un éditeur de texte pour écrire du code, une console de texte, une zone de message, une barre d'outils avec des boutons pour les fonctions communes et une série de menus [32].

Il se relie au matériel Arduino pour télécharger des programmes et communiquer avec eux [32].

Les programmes Arduino comportent trois parties [33] :

- Déclaration des variables et constantes globales.
- Fonction Setup () : exécutée au démarrage du programme ou lors de l'appui sur le boutonReset.
- Fonction Loop () : boucle exécutée sans fin (enfin tout pendant que l'Arduino est sous tension).

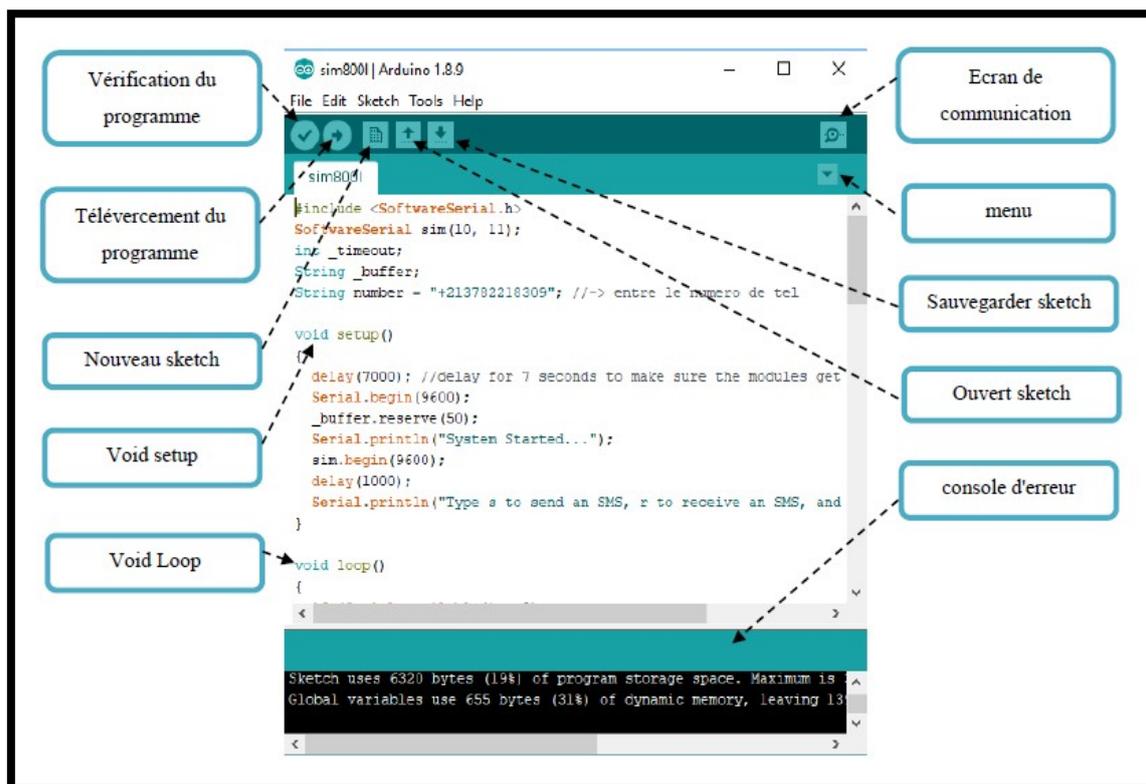


Figure II. 6 Structure générale d'IDE Arduino [29]

### II.3 Le capteur infrarouge

Un capteur infrarouge(IR) est un dispositif électronique qui mesure et détecte certains aspects physique tels que la chaleur et le rayonnement [34].

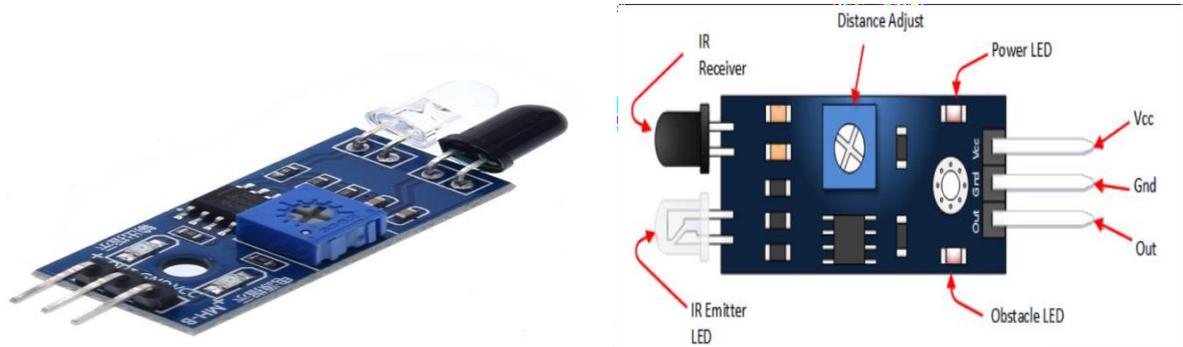


Figure II.7 Capteur Infrarouge [35] Ce circuit est réalisé en utilisant les éléments suivants:

[36]

- LM358IC2.
- Paire émetteur et récepteur IR.
- Résistances de la gamme des kilo-ohms.
- Résistances variables. (Pour régler la sensibilité).
- LED (diode électroluminescente).

Le tableau suivant présente les caractéristiques techniques d'un module infrarouge :

[37]

<b>Voltage de fonctionnement</b>	<b>DC 3.3V-5V</b>
<b>Intensité du courant</b>	<b>&gt;ou égale à 20mA</b>
<b>Température de fonctionnement</b>	<b>-10° C jusqu'à +50° C.</b>
<b>Plage de distance</b>	<b>2-40cm</b>
<b>Interface IO</b>	<b>Interface 4fils (GND/VCC/OUT/EN)</b>
<b>Signal de sortie</b>	<b>Tension TTL</b>
<b>Taille</b>	<b>41.717mm</b>
<b>Angle effectif</b>	<b>35°</b>
<b>Poids</b>	<b>5g</b>
<b>Une bonne précision</b>	

Tableau II.1 Les caractéristiques techniques du capteur infrarouge [37]

### II.3.1 Principe de fonctionnement du capteur infrarouge

Un circuit de capteur IR est l'un des modules de base dans un appareil électronique. Les principaux dispositifs qui composent ce capteur IR sont une LED IR et une photodiode IR, un amplificateur opérationnel LM358 et une résistance variable VR1 pour régler la sensibilité du capteur. Le schéma de ce circuit est illustré dans la figure II.8

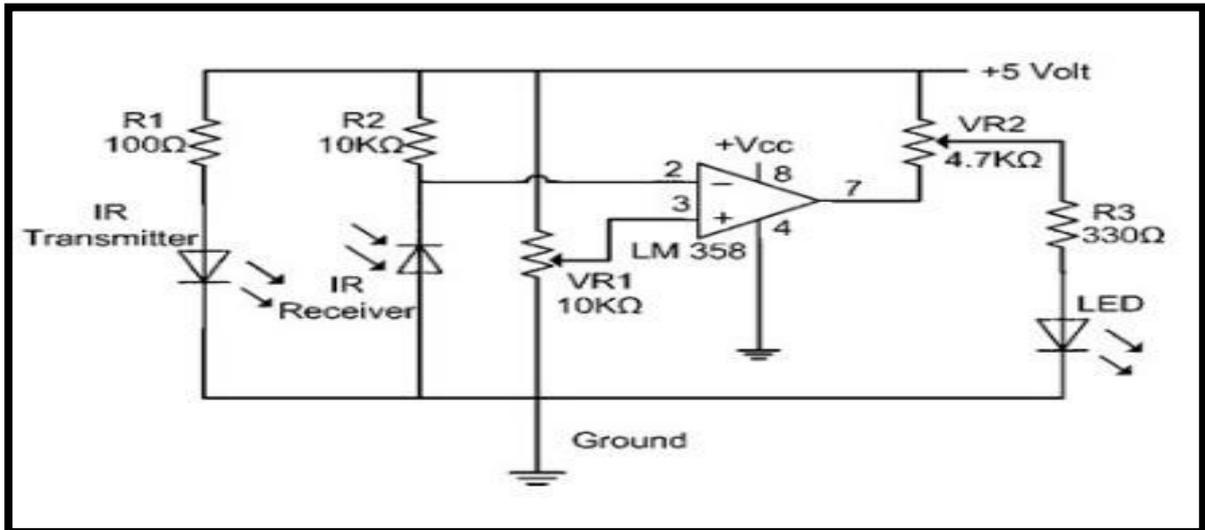


Figure II .8 : Schéma du circuit de capteur IR [36]

En effet, L'émetteur IR (LED IR) permet de générer et d'émettre des radiations infrarouges [36].

En absence d'un objet, le potentiel du récepteur IR est plus important par rapport au potentiel de résistance variable. De ce fait la sortie d'amplificateur opérationnel LM358 se trouve à l'état bas (0V) [36].

Dans le cas inverse, la présence d'un obstacle, les rayonnements émis par la LED IR sont réfléchis vers le récepteur IR. Dans ce cas la tension aux bornes de la photodiode devient faible par rapport à la tension de référence [36].

Par conséquent, la sortie du comparateur (AOP LM358) change d'état et passe de 0V à 5V [36].

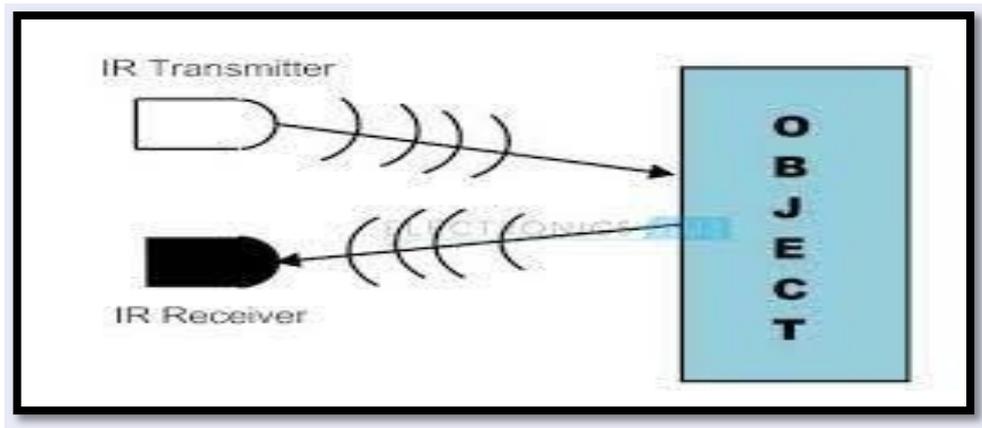


Figure II.9 Principe de fonctionnement d'IR [37]

### II.3.3 Câblage du module IR avec Arduino UNO

La première étape est de relier les broches VCC et GND du IR aux broches 5V et GND de l'Arduino.

Puis, connecter la sortie analogique ou numérique du capteur IR avec l'entrée analogique ou numérique d'Arduino (voir figure II.10).

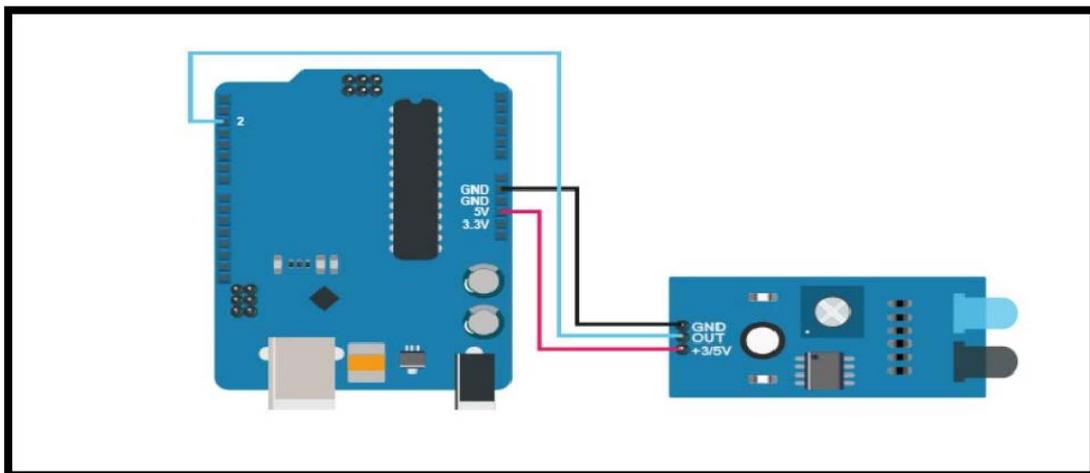


Figure II.10 : Branchement d'Arduino avec capteur IR[38]

## II.4 Les Afficheurs LCD

L'afficheur LCD (en Anglais « Liquid Crystal Display » et en français « Écran à cristaux liquides ») est une interface visuelle entre un système (projet) et l'être humain (utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Donc il affichera des données susceptibles d'être exploitées par l'utilisateur d'un système [39].

L'écran à cristaux liquides ou LCD utilise un mode d'affichage numérique sur un écran plat à faible consommation d'électricité. Aujourd'hui ces écrans sont utilisés dans presque tous les affichages électroniques [29].

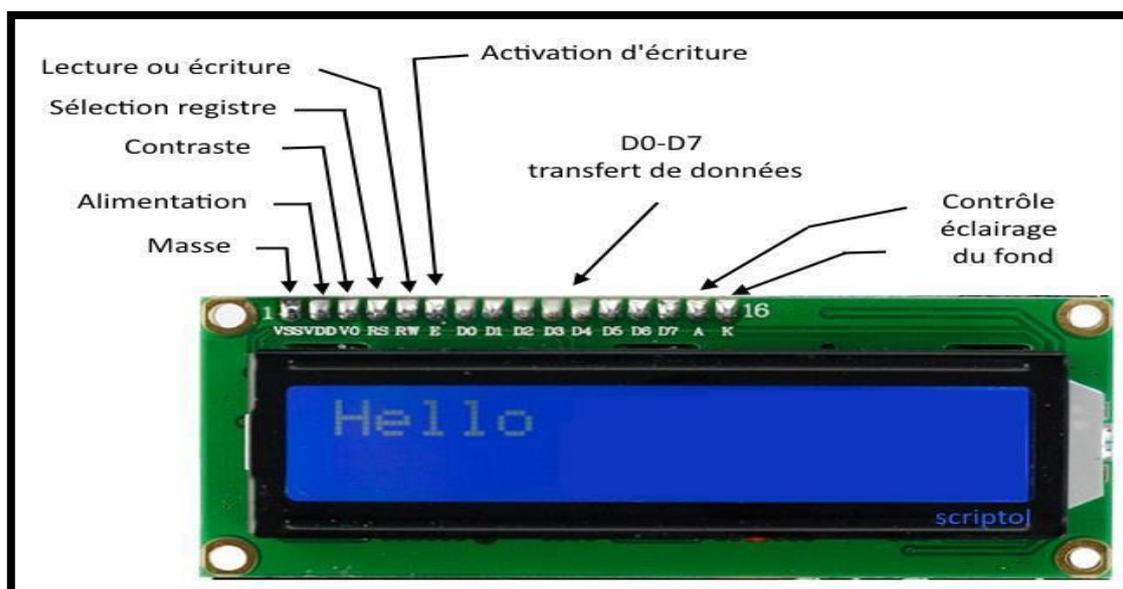
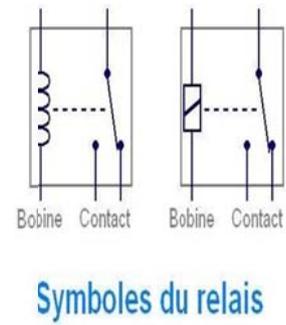
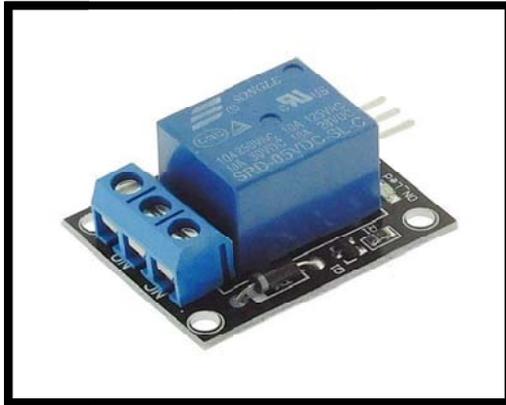


Figure II.11 Afficheur LCD 16\*2 [29]

## II.5 Relais électrique

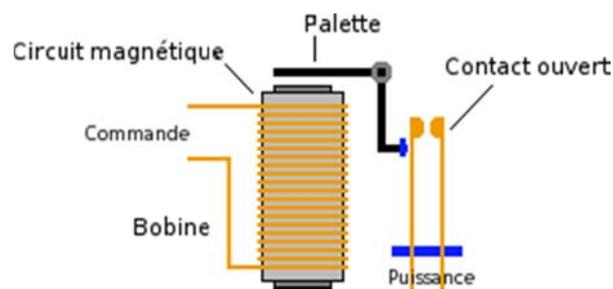
Le relais est un interrupteur qui se contrôle avec une tension continue ou alternative de faible puissance [40].

La partie commutateur (interrupteur) est utilisée pour piloter des charges secteur de forte puissance [40].



**Figure II.12** Relais électrique et ses symboles [41]

Un relais est principalement composé d'un électroaimant. Une fois, mis sous une tension, il transmet une force à un système de commutation électrique : les contacts [40].



**Figure II.13** Les constitutions de relais [40].

Le système de commutation peut être constitué d'un un ou plusieurs interrupteurs simples effets appelés contacts normalement ouvert(NO) ou normalement fermé (NF ou NC). Ces interrupteurs sont adaptés aux courants et à la palette (gamme) de tension à transmettre à la partie puissance [40].

Un relais a un enroulement comme organe de commande. La tension appliquée à ce bobinage va générer un courant. Ce dernier génère un champ électromagnétique à l'extrémité de la bobine. Ce champ magnétique attire une pièce mécanique métallique monté sur un axe mobile qui déplacera donc des contacts mécaniques. Lors de l'absence de courant dans la bobine les contacts reviennent à leur position de repos grâce à un ressort de rappel[42].

## II.6 Electrovanne

Une électrovanne représente un moyen simple et économique afin de contrôler les réseaux de fluide à distance grâce à un signal électrique. Les fonctions courantes pour lesquelles une électrovanne peut être utilisée sont : l'alimentation à distance, le sectionnement de tuyauterie, la fonction de by-pass, la mise à l'air, la purge des réservoirs et des tuyauteries, la distribution par dérivation [43].

Les secteurs principaux des électrovannes sont le traitement des eaux, réseaux de procédés, réseau de vapeur, électrovanne pour l'air comprimé, pour les réseaux de fluides caloporteur, eau chaude, eau froide, eau glycosylée (réchauffage, refroidissement, échangeur). Électrovanne pour réseaux d'incendie, réseaux de gaz, réseaux de combustibles (fuel). Électrovanne pour les constructeurs de machines [43].



Figure II.14 Electrovanne [29]

Le principe de fonctionnement de l'électrovanne est similaire à celui du relais, elle comporte une bobine à l'intérieur, qui une fois sous tension, tire le matériau conducteur (piston) à l'intérieur, permettant ainsi l'écoulement du liquide. Lorsque celle-ci est hors tension, il repousse le piston dans la position initiale à l'aide du ressort et bloque à nouveau l'écoulement du liquide. Au cours de ce processus, la bobine tire une grande quantité de courant et peut ainsi engendrer un problème d'hystérésis. Il est donc impossible de piloter une bobine de solénoïde directement à travers un circuit logique. Dans notre prototype, nous utilisons une électrovanne pour contrôler le débit de liquides [44].

Donc, il est important de tenir compte de ces éléments lors de la conception du circuit de commande de l'électrovanne [44].

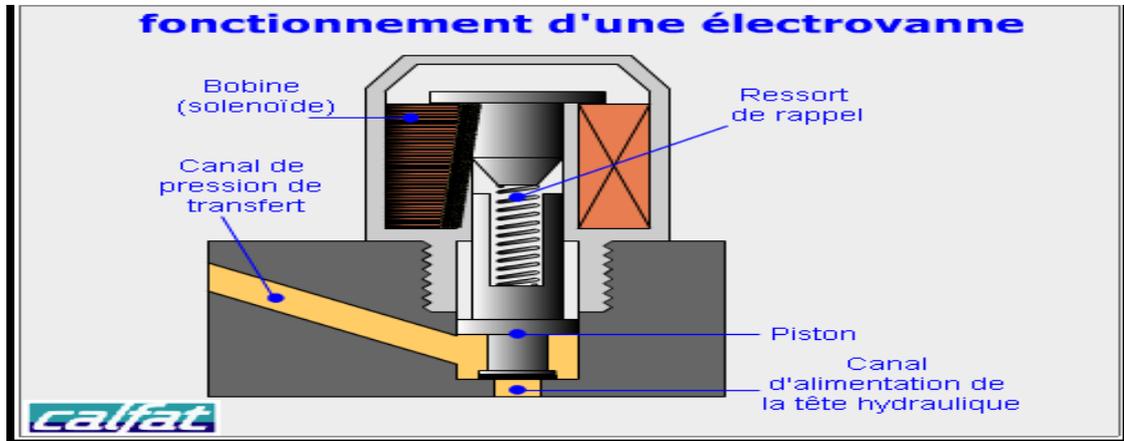


Figure II.15 Principe de fonctionnement de l'électrovanne [45]

## II.7 Convoyeur

Un convoyeur est un système de convoyage ou de transport qui fonctionne à l'aide d'un moteur à courant continu. Il est conçu pour le transport et le déchargement des produits solides de façon continue sur un trajet prédéterminé [46].

Il existe plusieurs types de convoyeurs, parmi ces types on cite quelques-uns : [46]

- Convoyeur à bande.
- Convoyeur à chaînes.
- Convoyeur à rouleaux.

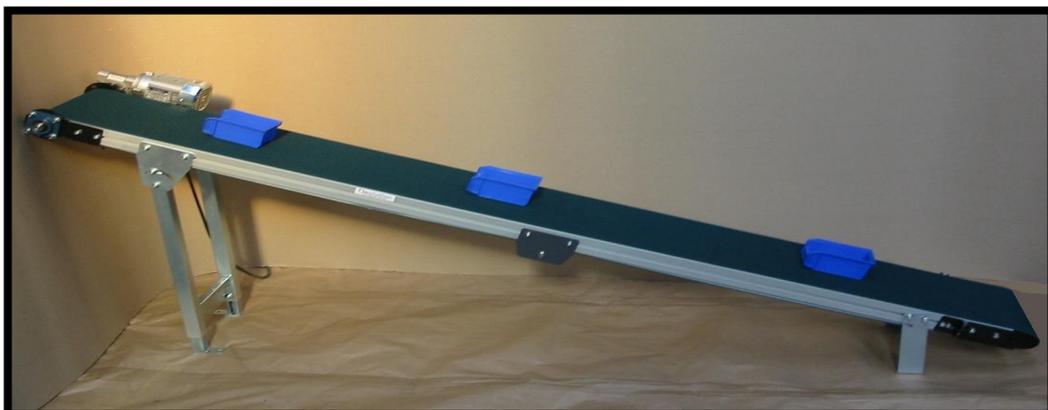


Figure II.16 Convoyeur [46]

## II.8 Moteur à courant continu

Un moteur à courant continu appelé aussi machine à courant continu (MCC, MDC) est un convertisseur qui transforme l'énergie électrique en énergie mécanique, capable de convertir une énergie bidirectionnelle entre un appareil électrique porté en courant continu et un appareil mécanique [47].



Figure II.17 Moteur à courant continu [47]

Le principe de fonctionnement d'un moteur à courant continu produit un couple électromagnétique sur le rotor du conducteur, qui est transmis par un courant continu et se déplaçant dans un champ magnétique [48].

Un inducteur (stator) génère un champ magnétique avec une direction fixe. Ce champ peut être obtenu par un aimant permanent ou par un électro-aimant [48].

L'induit (rotor) ou l'armature porte des conducteurs en raison du passage du courant continu. Les rotations sont le siège et le support des forces qui produisent le couple du rotor. De cette rotation, une variation génère une force électromotrice aux extrémités de la bobine [48].

## II.9 Pompe d'eau

Une pompe est un dispositif qui permet de refouler et d'aspirer un liquide [59].

La pompe est composée de trois parties distinctes : [50]

- La partie moteur, qui fournit la puissance nécessaire au pompage;
- La transmission, qui transmet cette puissance à la partie hydraulique;
- La partie hydraulique, qui transmet cette puissance à l'eau pour la déplacer (l'aspirer et/ou la refouler).



Figure II.18. Une pompe d'essuie-glace [51]

## II.10 Transistor à effet de champ (MOSFET)

Le transistor à effet de champ (MOSFET) est un dispositif électronique constitué de trois pattes : la grille, le drain et la source. Il représente l'élément de base dans l'industrie pour la construction de portes logiques [52-53].

Ce dispositif commandé en tension, car son courant de sortie varie en fonction d'une faible tension appliquée à sa grille [52-53].

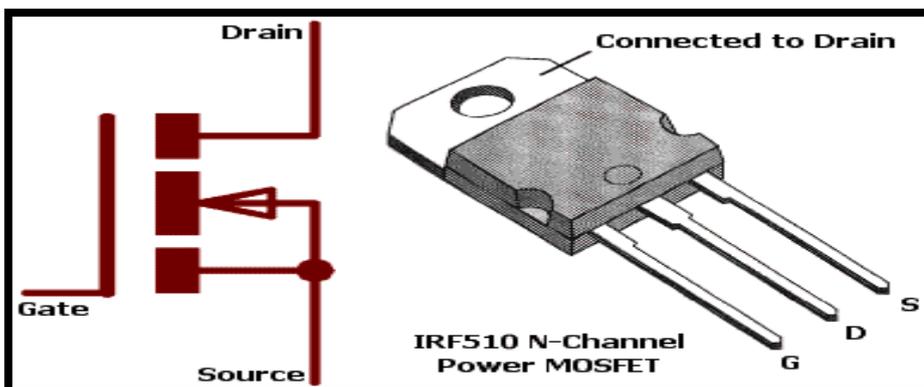


Figure II.19. Transistor MOSFET [54]

Le transistor MOSFET est contrôlé en appliquant une tension positive à sa grille.

En effet, au moment où cette tension dépasse une certaine valeur, il permet au courant de circuler entre la source et le drain [55].

Lorsque la tension de la grille est nulle, le transistor est un commutateur (interrupteur) ouvert [55].

La vitesse de commutation du transistor MOSFET est ralentie par les capacités parasites. [55] Dans notre prototype nous avons utilisé l'**IRFZ44N**.

L'**IRFZ44N** est un transistor MOSFET à canal N avec un courant de drain élevé de 49A et une faible valeur de résistance de 17,5  $\Omega$ . Il possède une tension de seuil de 4V [56].

Par conséquent, il est souvent utilisé avec des microcontrôleurs pour alimenter avec 5V [56].

## II.11 La diode

La diode est un composant électronique constitué de deux broches anode (A) et cathode(K). Ce dispositif permet le passage du courant dans un seul sens .il agit comme un commutateur [57- 58].

La surface de séparation des régions de type P et N est une jonction PN [59].

Une diode possède deux états :

Un état passant, bloqué.

La diode passera d'un état à l'autre suivant le sens du courant et aussi suivant la tension à ses bornes. Celle-ci doit être dépassé un certain seuil afin le courant puisse circuler.

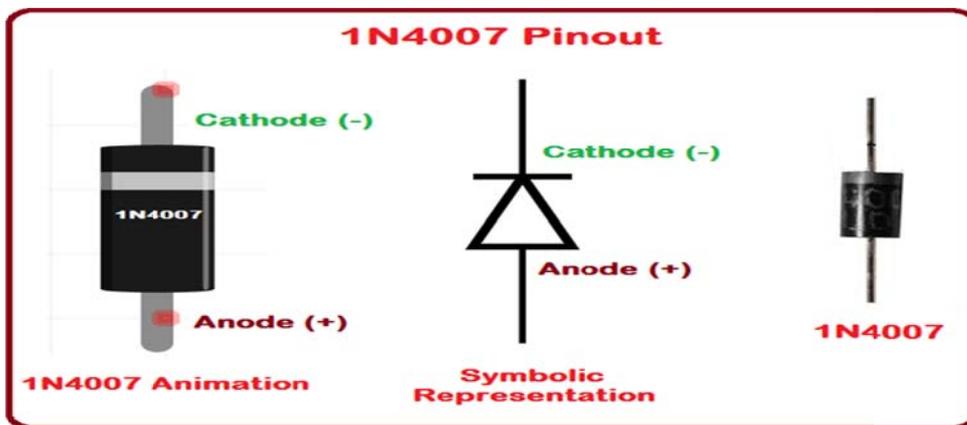


Figure II.20 Diode 1N400 [60]

## II.12 Résistance

Une résistance est un dispositif électronique passif dont la fonction essentielle est de résister et limiter la circulation de courant électrique. L'unité de résistance est l'ohm ( $\Omega$ ) mesurable par un ohmmètre [61-62].

Lorsque la résistance est traversée par un courant électrique, provoque une différence de potentiel entre ses bornes [62].

Il existe plusieurs méthodes pour déterminer la valeur de la résistance R. On cite deux méthodes de ces méthodes :

- La loi d'Ohm :  $R=U/I$

Avec :

U : la tension (V) I : le courant (A)

- Code de couleur:

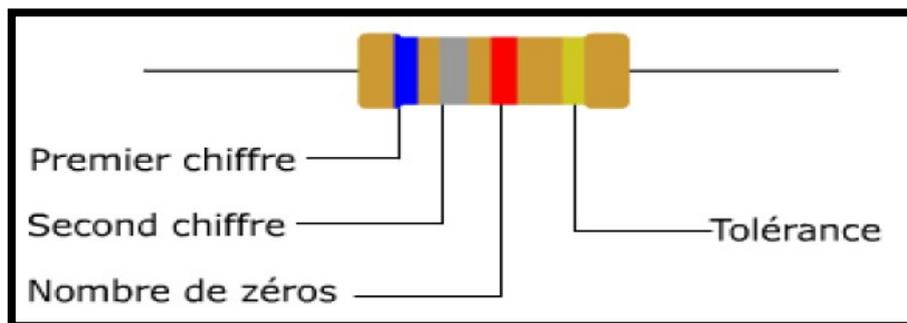


Figure II.21 Résistance électrique [62]

## II.13 Conclusion

Dans ce chapitre nous avons donné une description sur les différents dispositifs qui constituent notre prototype.

Les composants essentiels sont le module infrarouge et la carte Arduino UNO.

Le capteur IR est utilisé pour la détection des bouteilles d'eaux, tandis que la carte Arduino UNO est utilisée pour le contrôle des différents éléments tels que : le relais, l'électrovanne...etc.

*Chapitre III*  
*Simulation et réalisation*  
*de circuit*

### III.1 Introduction

Notre projet consiste à réaliser un système de remplissage automatique des bouteilles d'eaux.

Dans ce chapitre nous allons présenter le principe de fonctionnement ainsi que les différents blocs qui constituent notre projet.

Ces différents étages sont représentés dans la figure (Figure III.1) :

L'élément principal de notre circuit est la carte Arduino UNO. Cette dernière permet le contrôle des différents éléments.

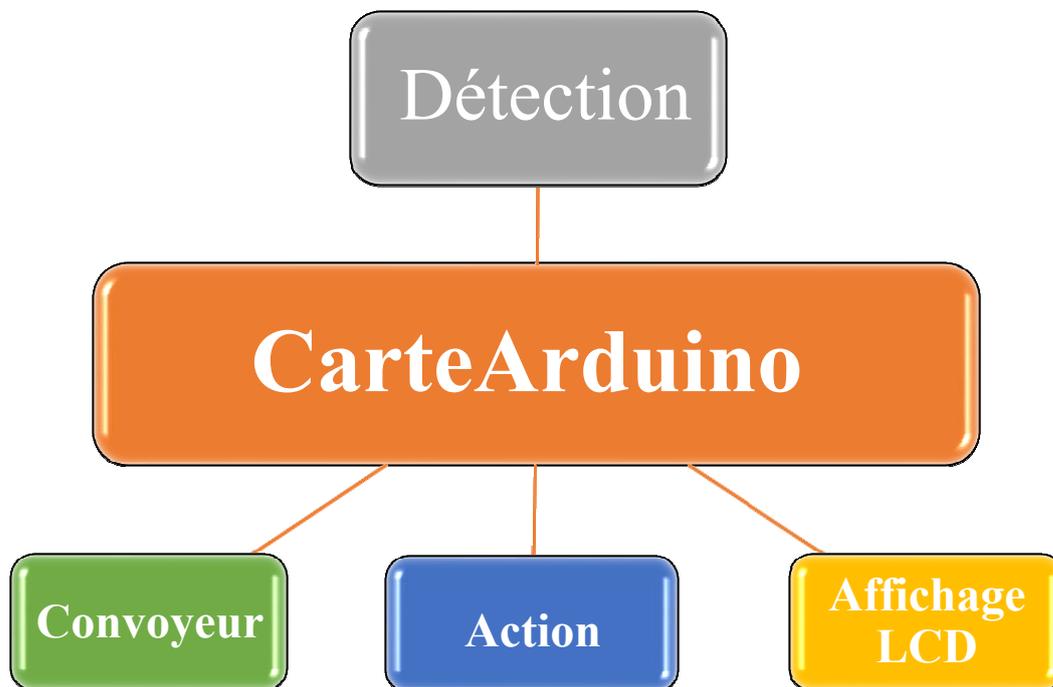


Figure III.1. Représentation schématique du prototype

Cependant, avant de passer à la partie réalisation, il est très intéressant de simuler notre circuit.

Cette simulation est réalisée en utilisant le logiciel « PROTEUS » :

### III.2 Partiesimulation

La simulation des circuits électroniques représente un élément essentiel pour le développement de la recherche scientifique et technologique. Cette tâche nous offre la possibilité de minimiser les coûts en utilisant une suite d'essais expérimentaux[63].

En effet il existe plusieurs logiciels de simulation tels que : multisim, PDN, analyzer...etc. Cependant, le logiciel le plus répandu est le logiciel « PROTEUS ». Dans la partie suivante, nous allons donner un aperçu sur ce simulateur [63].

### **III.2.1 Présentation du logiciel PROTEUS**

PROTEUS représente une suite logicielle de conception assisté par ordinateur (CAO) développé par Labcenter Electronics [63].

Cet environnement, réservé au domaine d'électronique, nous permet la conception et la simulation de différents systèmes [63].

Les deux principaux modules de simulation PROTEUS sont « ISIS » et « ARES » [64].

#### **III.2.1.1 Présentation du module ISIS (Intelligent Schematic Input System)**

Le module ISIS du logiciel PROTEUS représente un moyen de développement et de simulation et des montages électroniques [63-64].

En effet, cet outil graphique, simple et interactif nous permet d'éditer les différents schémas électroniques. Ainsi, l'utilisation de cet environnement, nous offre la possibilité de révéler et de corriger les erreurs commises lors de l'étape de conception[63-64].

#### **III.2.1.2 Présentation du module ARES**

ARES est un outil destiné à la réalisation des circuits électroniques imprimés.

Cette interface nous permette dans un premier temps de placer les différents dispositifs [64].

Par la suite, il suffit de lier les différents composants en utilisant les différents modes de routages (routages manuelles et automatiques) [63-64].

Outre l'optimisation du circuit, ce module nous aide à établir une visualisation 3D de la carte électronique réalisée [63].

### III.3 Schémas et méthodologie suivie

#### III.3.1 Description du prototype

Notre prototype est divisé en trois blocs. Ces 3 principaux étages sont :

- Bloc de détection de la présence debouteilles.
- Bloc d'action
- Bloc deconvoyeur.

##### III.3.1.1 Bloc de détection

L'élément principal de cet étage est le module IR. Ce module est connecté avec la carte Arduino UNO via une sortie numérique.

En effet, lors de la présence d'un obstacle, une partie de la lumière émise est réfléchiée vers la photodiode. Ceci permet de générer une tension à la sortie de l'amplificateur opérationnel LM358. Par conséquent, la sortie du module IR passe à l'état haut (5V).

La figure ci-dessous représente le branchement de ce module avec la carte Arduino UNO sous ISIS:

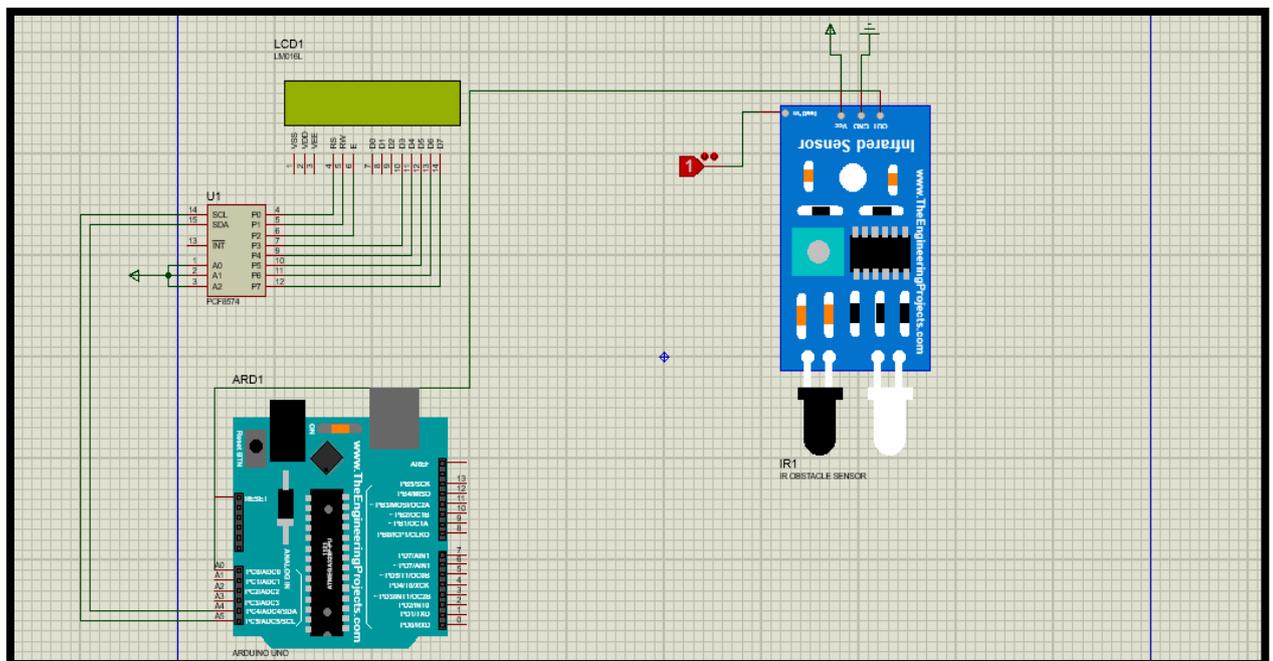


Figure III. 2. Le branchement du module IR avec la carte Arduino UNO sous PROTEUS.

### III.3.1.2 Bloc de convoyeur

Ce bloc est constitué de plusieurs éléments à savoir le tapis roulant, engrenages et un moteur DC. Ce dernier représente l'élément principal de ce bloc. Il est relié avec la carte Arduino UNO via une sortie numérique PWM.

Ainsi, le contrôle de la vitesse de ce moteur DC (démarrage/arrêt) est géré par un programme Arduino.

Le fonctionnement de notre convoyeur nécessite l'application d'une tension externe. Pour cela nous avons utilisé un transformateur 220/12 V.

En effet, lors de la mise en marche, le moteur DC fait tourner les engrenages. Ces derniers font tourner le tapis roulant.

Le circuit du bloc de convoyeur réalisé sous ISIS est donné via la figure III.3:

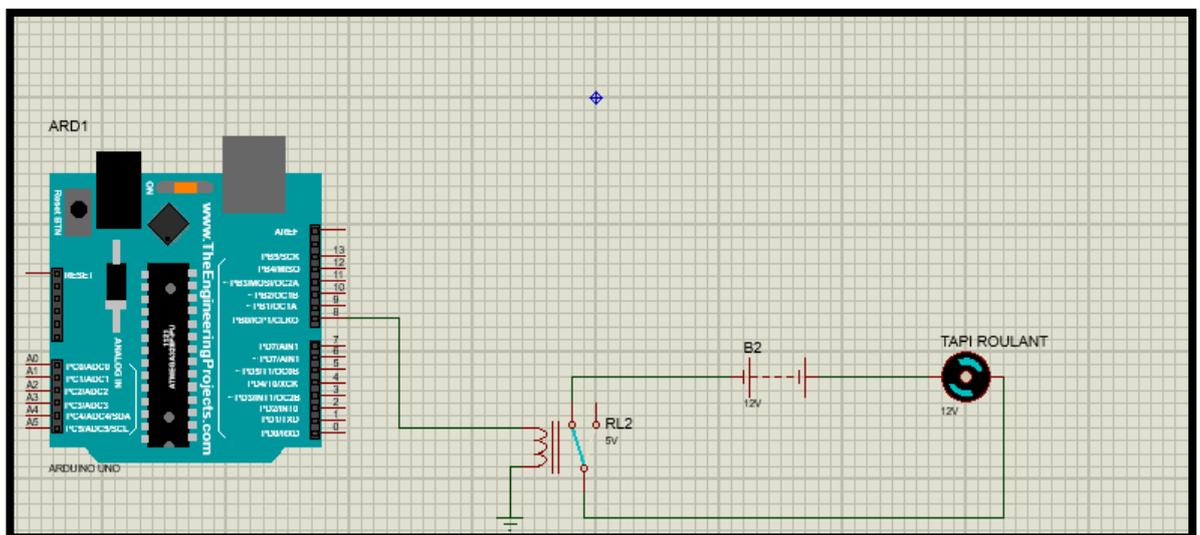


Figure III.3. Branchement du bloc convoyeur sous ISIS

### III.3.1.3 Bloc d'action

Le bloc d'action est constitué d'une électrovanne et une pompe (moteur DC), alimentés par une tension externe (9V). Ces deux éléments sont connectés à une carte Arduino UNO via un module de relais (voir **Figure III.4**).

Le mode de fonctionnement de ces dispositifs est géré par un programme téléversé vers la carte Arduino UNO.

Le tableau ci-dessous indique le branchement de ces composants électroniques :

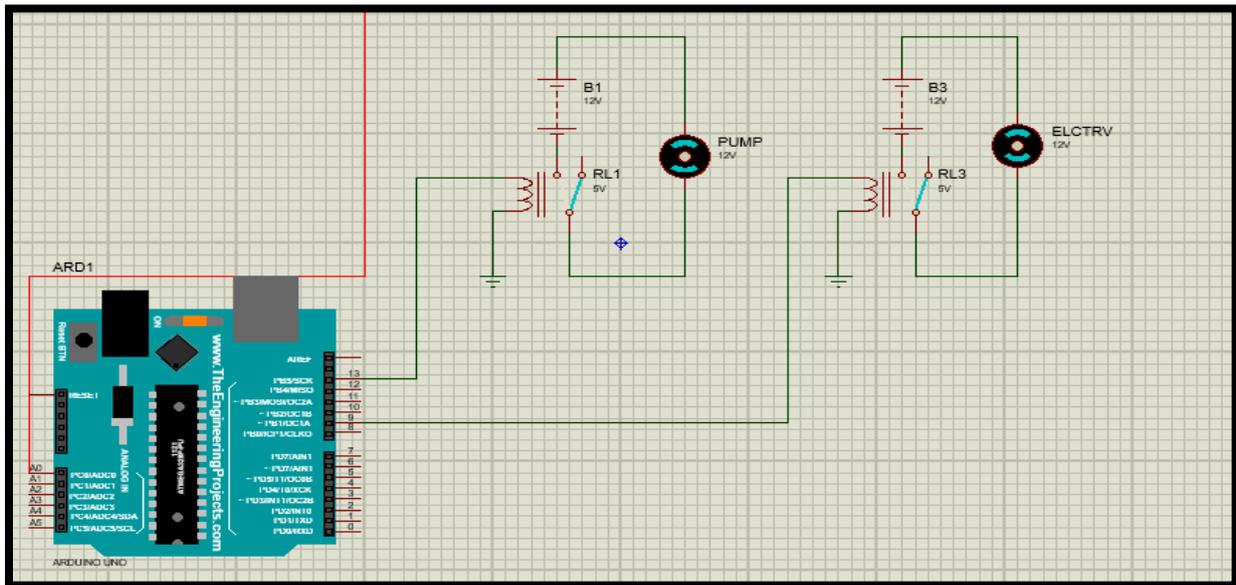
le composant	Electrode+	Electrode-
Electrovanne	Le pin numérique 5 de la carte Arduino UNO	Vers le pin GND de la carte Arduino
Pompe	Le pin numérique 4 de la carte Arduino UNO	Vers le pin GND de la carte Arduino

**Tableau III.1** : Branchement d'électrovanne et pompe avec Arduino

Lors de l'absence de la détection, les relais ne sont pas activés. Par conséquent. L'électrovanne se trouve à l'état fermé et la pompe s'est mise à l'arrêt.

Dans le cas inverse, les relais sont activés. De ce fait, l'électrovanne passe à l'état ouvert et la pompe s'est mise en marche.

Le bloc d'action est représenté dans la figure ci-dessous :



**Figure III.4.** Branchement du bloc d'action sous ISIS

### III.4 Circuit global

La figure ci-dessous représente une vue globale du prototype simulé sous PROTEUS-ISIS :

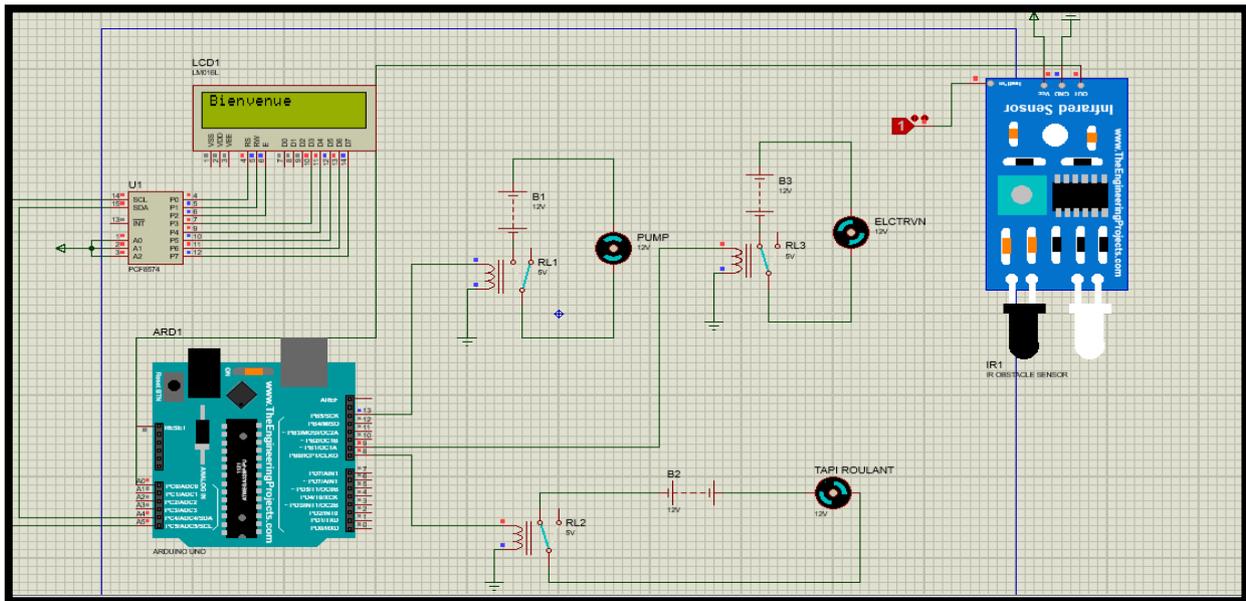


Figure III.5. Aperçu globale du notre prototype sous ISIS

### III.5 Partie réalisation

Lorsque notre prototype est mis en marche, l'écran LCD affiche un message d'accueil « Welcome ».

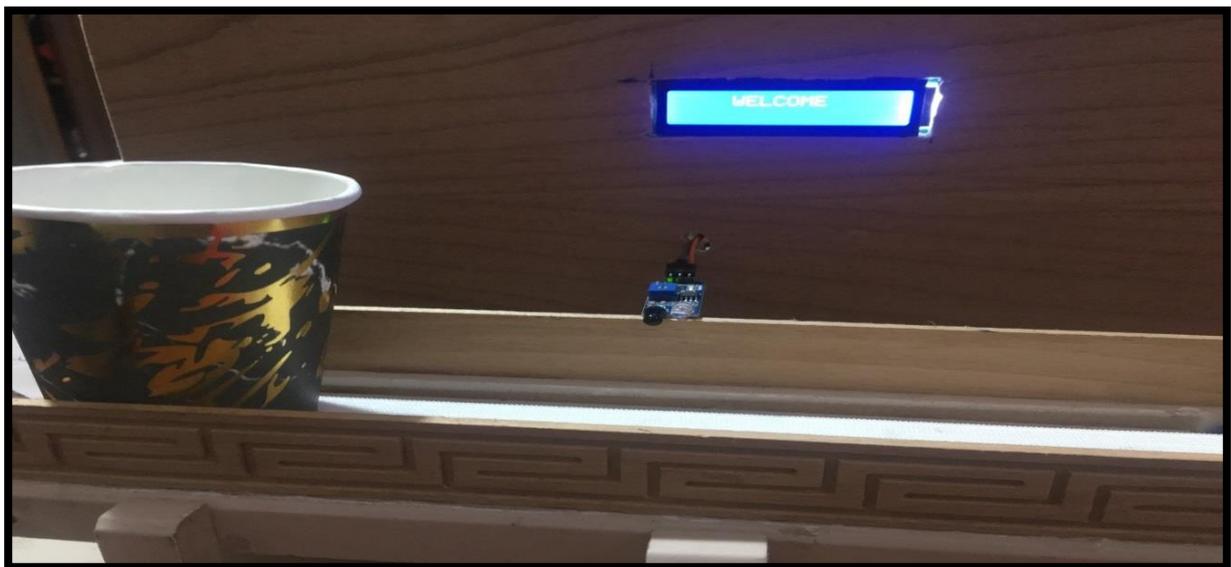
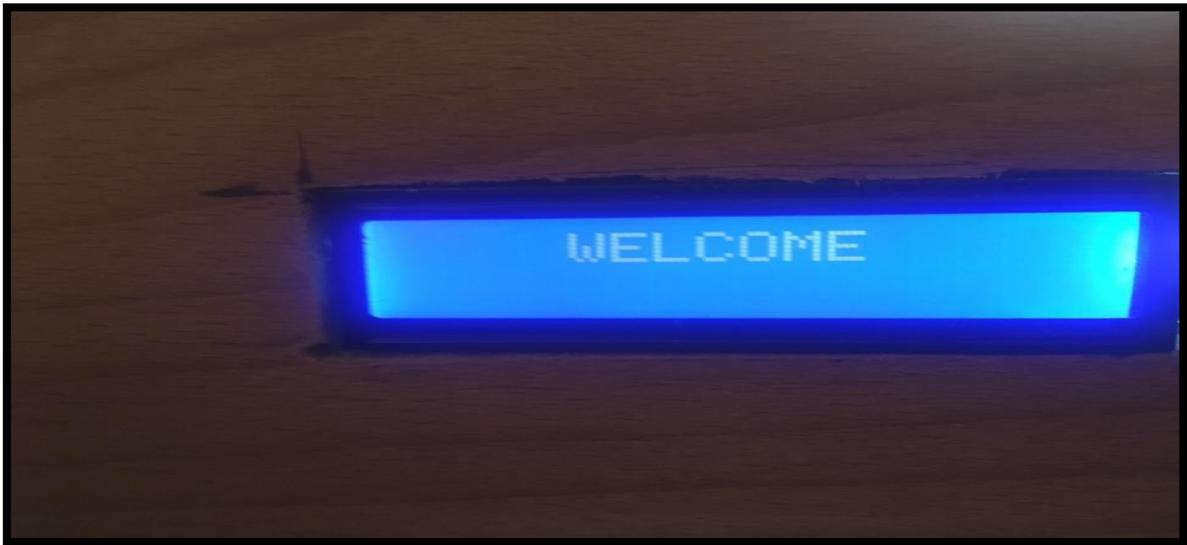


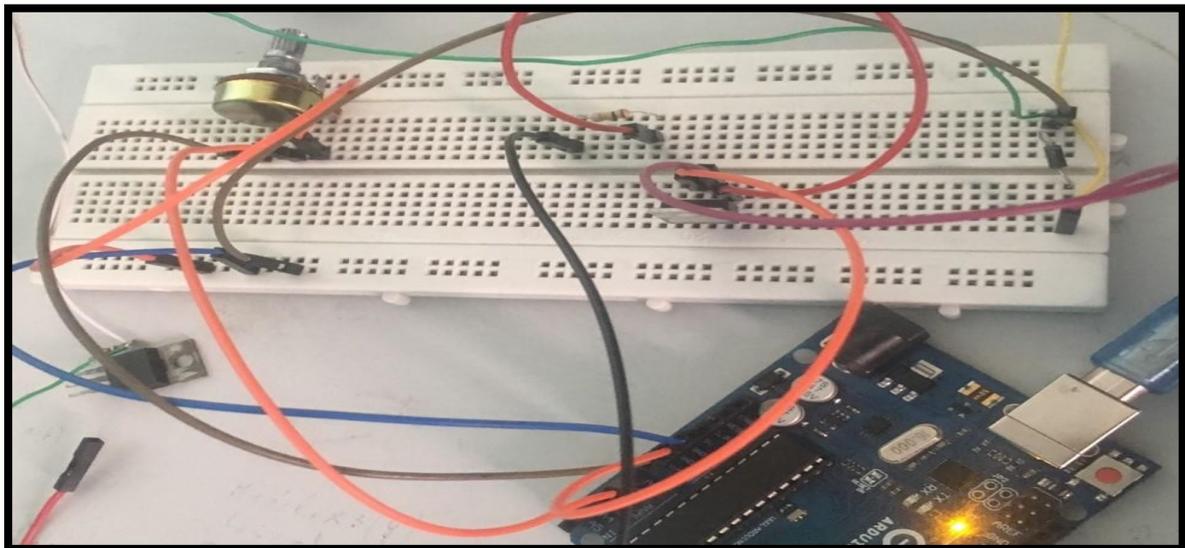
Figure III.6. La mise en marche du système



**Figure III.7.** Affichage du message welcome

Après quelques instants, le moteur DC du convoyeur est opérationnel et tourne avec une vitesse adéquate.

Le contrôle de cette vitesse nécessite l'intervention d'une partie de programmation en utilisant la plateforme Arduino IDE et d'un circuit en utilisant le transistor IRFZ44N. Le montage de ce circuit est représenté dans la figure III.8 :



**Figure III.8.** Circuit de contrôle de vitesse du moteur

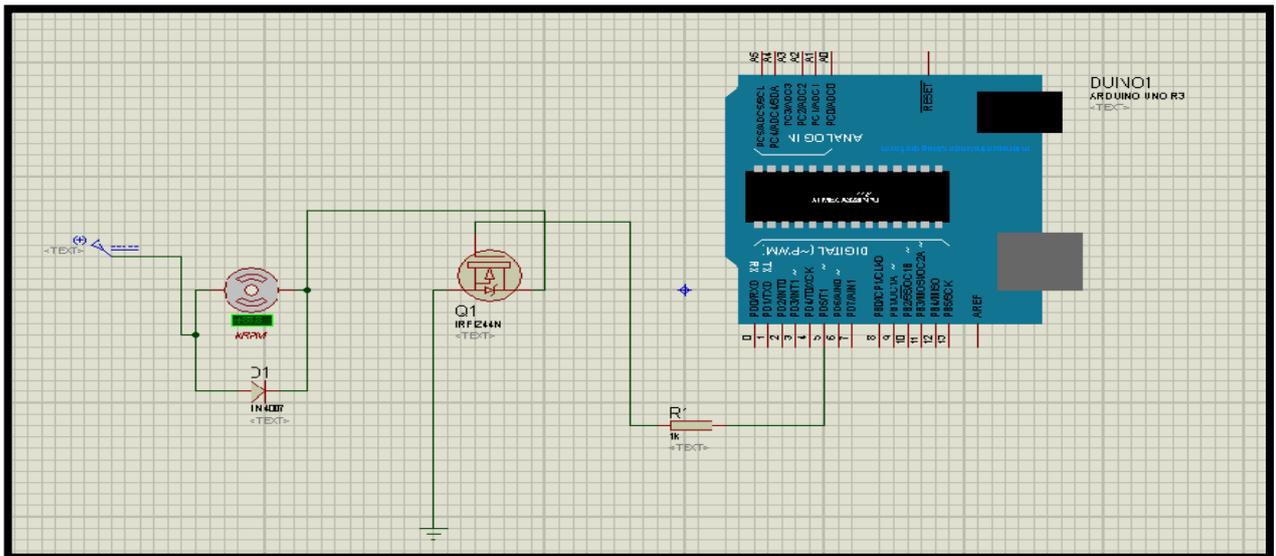


Figure III.9 Circuit de contrôle de vitesse du moteur sous ISIS

En absence des bouteilles, le système est inactif. Dans ces conditions, l'électrovanne et la pompe sont inactives. Par conséquent, l'afficheur LCD envoi le message « bottle not



Figure III.10. Affichage du message bottle not detected

L'étape suivante consiste à charger les bouteilles vides sur le convoyeur. Ces bouteilles vides sont acheminées vers l'unité de remplissage.

En effet, lors de la détection de la présence d'une bouteille vide le module IR génère

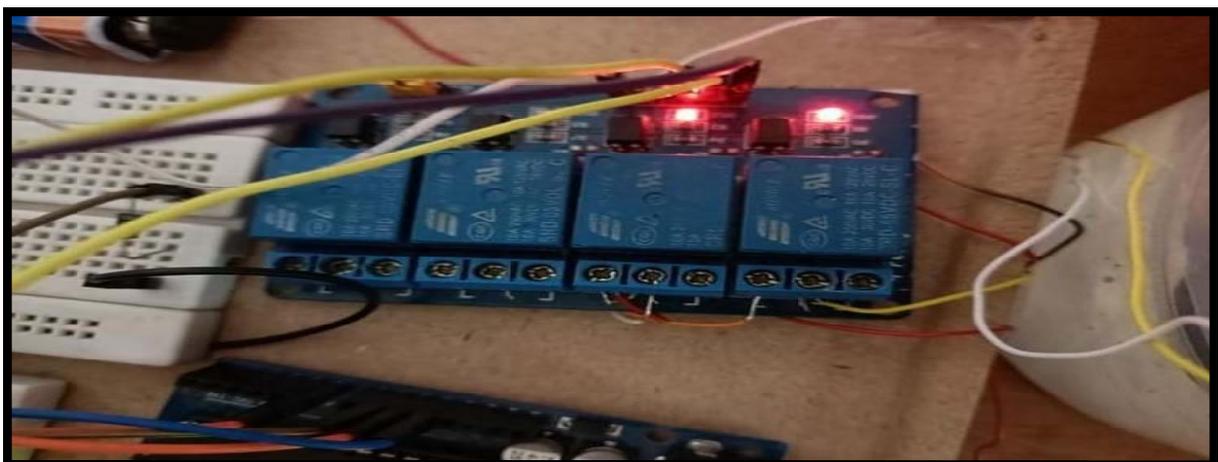
une tension à la sortie. Cette tension est délivrée à la carte Arduino UNO. Dans ce cas le relais change de position et passe de l'état NC vers l'étatNO.

Ceci entraîne l'arrêt du moteur DC du convoyeur. Par conséquent, l'écran LCD affiche le message suivant « bottle detected » (voir figure III.11).



**Figure III.11.** Bouteille détecté

Une fois détecter, la sortie numérique de la carte Arduino (pin n05) se trouve à l'état haut (niveau logique 1) et le relais passe de l'état NO (normal open) vers l'état NC (normal close). Par conséquent, l'électrovanne est alimentée et passe de l'état fermé vers l'étatouvert.



**Figure III.12.** Représente l'activation de relais

Ensuite, la sortie numérique de la carte Arduino (pin n 04) se trouve à l'état haut, ceci

entraîne la mise en marche de la pompe et permet le remplissage des bouteilles (voir figure III.13). Le temps nécessaire pour le remplissage est fixé par le programme Arduino.



**Figure III.13** Remplissage des bouteilles

Après le remplissage, l'écran LCD affiche le message « bottle filled».

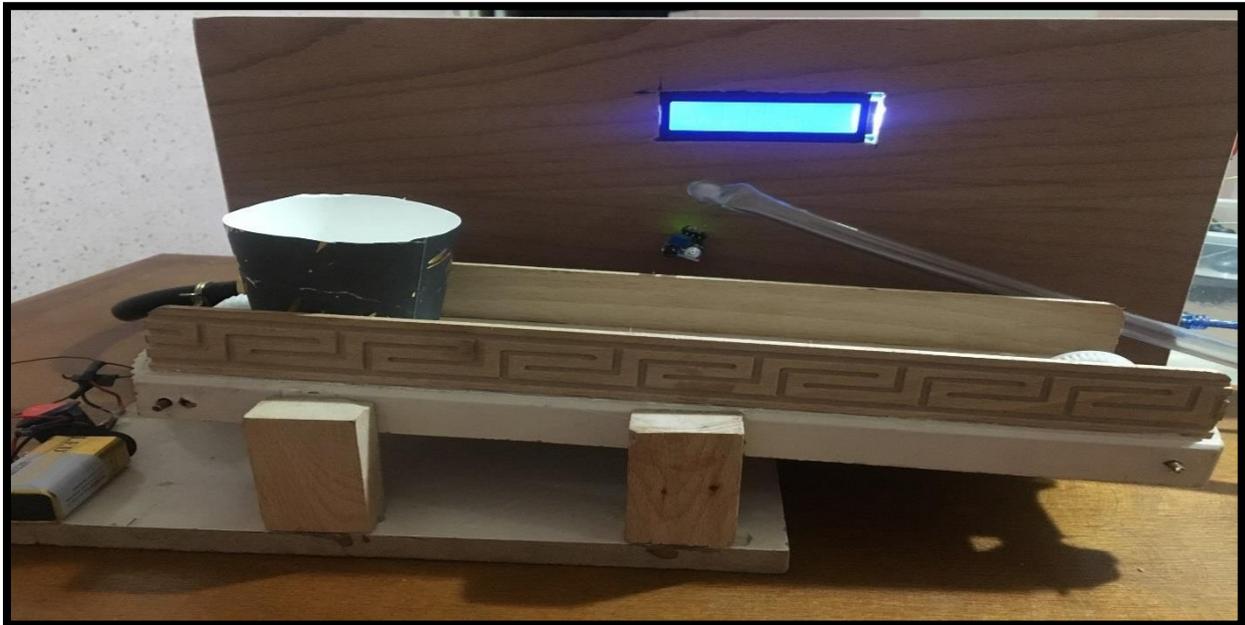


**Figure III.14.** Message « bottle filled»

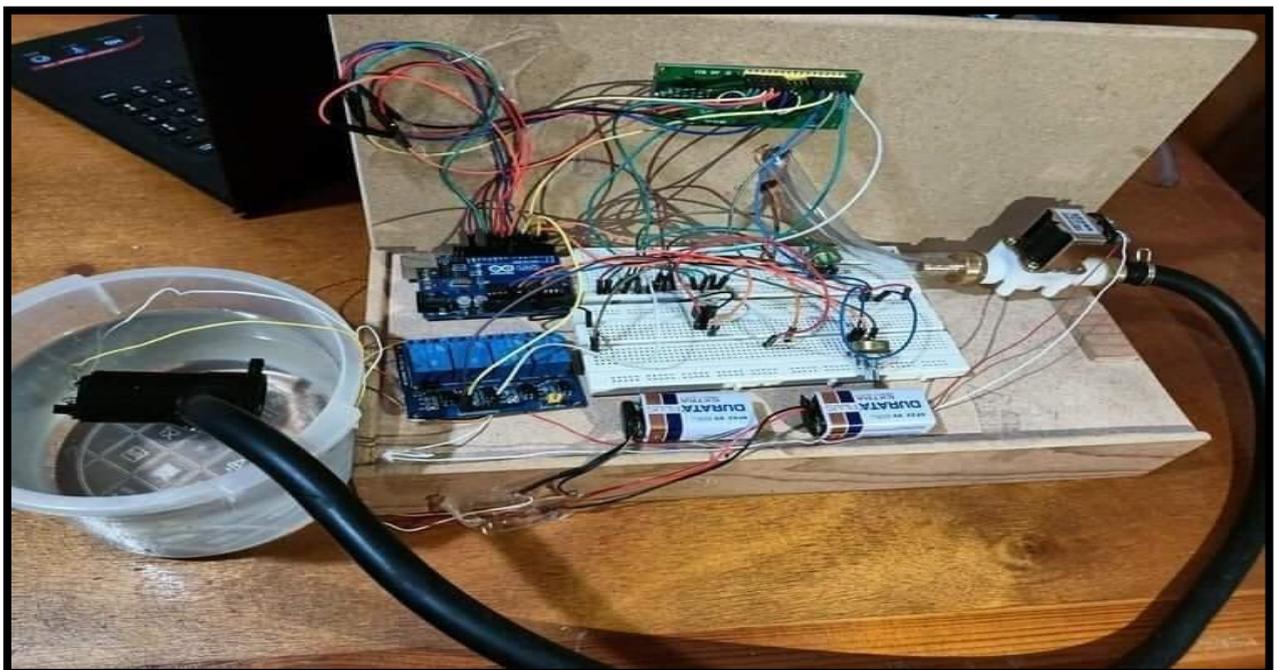
Une fois que ces opérations sont achevées, le système retourne à l'état initial (absence

du bouteille).

Un aperçu global de notre prototype réalisé est représenté dans les figures III.15 et III.16



**Figure III.15:** Prototype de détection des bouteilles « la face avant ».



**Figure III.16 :** Prototype de remplissage des bouteilles « la face arrière ».

Il est possible de réaliser notre prototype en utilisant les API. Pour cela, il suffit de

créer le grafcet en utilisant le logiciel « TIA PORTAL ». Ce grafcet est illustré dans la figure(III.17)

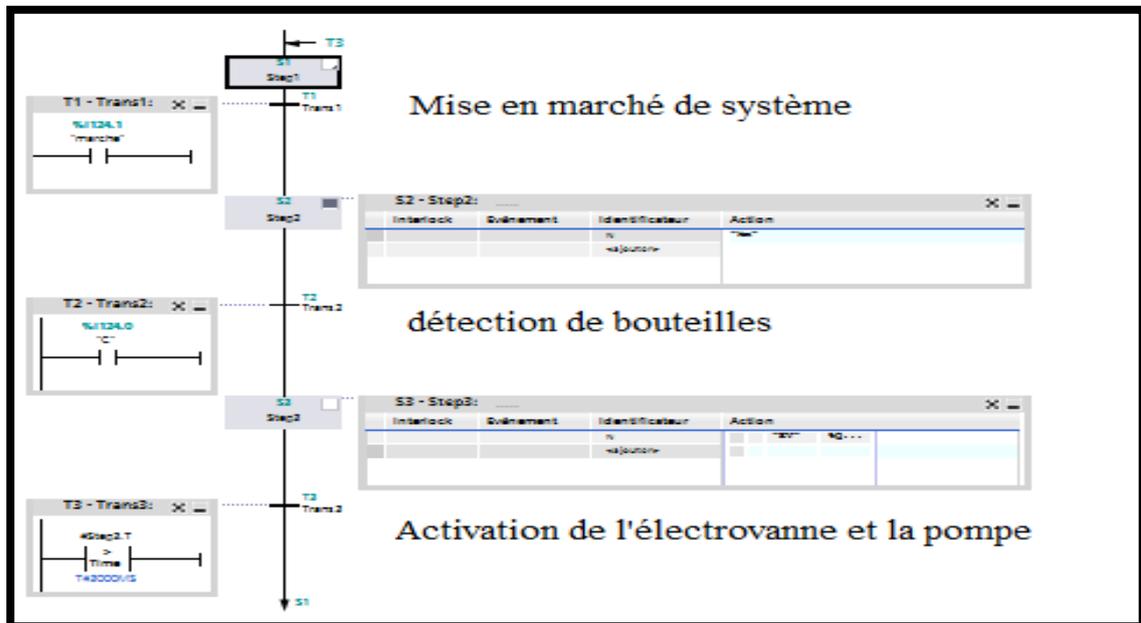


Figure III.17. Grafcet de notre prototype.

Le moteur tourne et le tapis roulant transporte les bouteilles vides jusqu'à la position C1 qui sera détectée par le capteur C.

Une fois arrivé à cette position, le moteur s'arrête, le remplissage commence, l'électrovanne s'ouvre pour une durée de 20s et se referme.

Le moteur démarre de nouveau et le tapis avance et emmène la bouteille à sa position finale.

### **III.6 Conclusion**

Dans ce chapitre nous avons présenté les différents étages nécessaires à la réalisation de notre projet.

Cependant, avant de passer à la réalisation, il était important de simuler les différents blocs en utilisant le logiciel PROTEUS.

Par la suite, nous avons présenté les différentes démarches qui nous a permis de réaliser notre prototype.

Enfin, nous avons lancé plusieurs tests afin de valider le bon fonctionnement du notre système.

## *Conclusion générale*

## *Conclusion Générale*

---

Les systèmes de remplissages automatiques de bouteilles sont omniprésents dans l'industrie.

Ces machines de remplissages exploitent le domaine de PLC.

Cependant, il est possible de réaliser de tels systèmes en utilisant une carte Arduino UNO. C'est dans ce contexte que notre travail intervient.

En effet, L'objectif principal de notre mémoire est de concevoir un système de remplissage automatique de bouteilles en utilisant la carte Arduino UNO.

Notre prototype est constitué de plusieurs étage à savoir le bloc de détection, bloc d'action, bloc de convoyeur, et bloc d'affichage.

En premier lieu nous avons procédé à la simulation de ces différents étages sous l'environnement PROTEUS ISIS.

Ceci nous a permis d'une part d'optimiser la conception de notre circuit global et d'autre part de développer le programme sous l'Arduino IDE.

Enfin, et afin de vérifier le fonctionnement de notre projet plusieurs tests ont été effectués. En effet, les résultats obtenus valident le bon fonctionnement de notre système.

Les perspectives qui découlent de ce mémoire portent sur l'amélioration de notre prototype, en ajoutant un capteur ultrason afin de détecter le niveau de liquide dans les bouteilles.

De plus, il est intéressant d'installé un système qui permet l'insertion automatique des bouchons.

## *Références*

- [1]: Bouacem Mourad, Bouhamadouche Hamza, Etude et Automatisation du Système d'Emplissage de Bouteilles de Gaz butane Par l'API S7-300 : université Mouloud Mammeride Tizi-Ouzou, Faculté de Génieélectrique et d'informatique, département d'électronique
- [2]: Namaoui Nazha, Automatisation d'un système de remplissage de quatre trémies de sucre, avec supervision HMI, CEVITAL. Université Abderrahmane MIRA 2014
- [3]: BELAIDI Ouerdia BELHACENE Lydia, Conception d'une Automatisation et d'une Supervision d'un Four de Cuisson d'Emallage (E11) au sein de l'ENIEM : Université Mouloud Mammeri De Tizi-Ouzou, Faculté De Génie Electrique Et D'informatique, DEPARTEMENT D'AUTOMATIQUE2017/2018.
- [4]: William BOLTON, « programmable LogicControllers »,2009
- [5]:<https://www.techno-science.net/definition/6694.html>
- [6]:<http://www.estusmba.ac.ma/coursenligne/GES2Automatismes%20logiques%20Industriels-CRS-EI%20Hammoumi.pdf>
- [7]:[https://www.researchgate.net/publication/349039684\\_Automates\\_Programmables\\_Industriels\\_Description\\_et\\_programmation](https://www.researchgate.net/publication/349039684_Automates_Programmables_Industriels_Description_et_programmation)
- [8]:[https://www.geea.org/IMG/pdf/LES\\_AUTOMATES\\_PROGRAMMABLES\\_INDUSTRIELS\\_pour\\_GEEA.pdf](https://www.geea.org/IMG/pdf/LES_AUTOMATES_PROGRAMMABLES_INDUSTRIELS_pour_GEEA.pdf)
- [9] :<https://letket.com/everything-about-api-application-programming-interface/>
- [10] :[https://www.technologuepro.com/cours-automate-programmable\\_industriel/Les-automates-programmables-industriels-API.htm](https://www.technologuepro.com/cours-automate-programmable_industriel/Les-automates-programmables-industriels-API.htm)
- [11] :[http://www.eaton.fr/EatonFR/ProductsSolutions/Electrique/ProduitsServices/AutomatismeControleMoteur/AutomatismeControleVisualisation/APIModulairesetCompacts/EC4P/PCT\\_3037012](http://www.eaton.fr/EatonFR/ProductsSolutions/Electrique/ProduitsServices/AutomatismeControleMoteur/AutomatismeControleVisualisation/APIModulairesetCompacts/EC4P/PCT_3037012)
- [12] :<https://dipslab.com/difference-between-compact-modular-plc/>
- [13] :<https://azaiez-houssam.developpez.com/articles/automatisme/api/introduction/>
- [14] :<https://www.futura-sciences.com/tech/definitions/informatique-automate-programmable-10525/>
- [15] :[https://fac.umc.edu.dz/ista/pdf/cours/L3-GIM\\_R%C3%A9seau%20automatescours1\\_Programmation%20des%20API.pdf](https://fac.umc.edu.dz/ista/pdf/cours/L3-GIM_R%C3%A9seau%20automatescours1_Programmation%20des%20API.pdf)
- [16] :<https://www.automation-sense.com/blog/automatisme/le-langage-ladder-pour-les-nuls.html>

- [17] :<http://sii-technologie.spip.ac-rouen.fr/IMG/pdf/ladder.pdf>
- [18] :[http://public.iutenligne.net/informatique/informatique-industrielle/deprez\\_maillfert/Autom/2Pupitre/premieres\\_programmations\\_et\\_dcouverte\\_du\\_langage\\_ladder.html](http://public.iutenligne.net/informatique/informatique-industrielle/deprez_maillfert/Autom/2Pupitre/premieres_programmations_et_dcouverte_du_langage_ladder.html)
- [19] :[https://product-help.schneider-electric.com/Machine%20Expert/V1.1/fr/SoMProg/SoMProg/FBD\\_LD\\_IL\\_Editor/FBD\\_LD\\_IL\\_Editor-4.htm](https://product-help.schneider-electric.com/Machine%20Expert/V1.1/fr/SoMProg/SoMProg/FBD_LD_IL_Editor/FBD_LD_IL_Editor-4.htm)
- [20] :<https://realpars.com/ladder-logic-vs-function-block-diagrams/>
- [21] :[https://en.wikipedia.org/wiki/Instruction\\_list](https://en.wikipedia.org/wiki/Instruction_list)
- [22] :<https://www.motioncontroltips.com/instruction-lists-ils-plc-programming/>
- [23] :[https://infosys.beckhoff.com/english.php?content=../content/1033/tcpcontrol/html/TcPlcCtrl\\_Languages%20IL.htm](https://infosys.beckhoff.com/english.php?content=../content/1033/tcpcontrol/html/TcPlcCtrl_Languages%20IL.htm)
- [24] :<https://realpars.com/plc-programming-languages/>
- [25] [http://public.iutenligne.net/informatique/informatique-industrielle/deprez\\_maillfert/Autom/2Pupitre/programmation\\_en\\_texte\\_structur.html](http://public.iutenligne.net/informatique/informatique-industrielle/deprez_maillfert/Autom/2Pupitre/programmation_en_texte_structur.html)
- [26] :MellaliSofyane, Yousfi Lounis, étude de l'automatisation et de la supervision d'un procédé de lavage de filtres Niagara à CEVITAL –TIA PORTAL V12- université Abderrahmane MIRA de Bejaia.
- [27] :<http://jackadit.com/index.php?p=sysprod3>
- [28] : KharatiBoualem, HidoucheRaouf, Automatisation et supervision d'un système d'entraînement de la centreuse M3T par l'automate programmable industriel(API) S7/300. Université M'HAMED BOUGARABOUMERDES.
- [29] : Ayad Benyounes & Tari Oussama, Simulation et réalisation d'un circuit détecteur de gaz interfacé à la carte Arduino. Université Abou bakrBelkaïd-Tlemcen, Faculté de Technologie. 2019/2020
- [30] :[http://www.lycee-ferry-versailles.fr/ee-tsi/12\\_arduino/html/p2.html](http://www.lycee-ferry-versailles.fr/ee-tsi/12_arduino/html/p2.html)
- [31] :<https://arduinotutoriels.com/arduino-ide-quest-ce-que-lenvironnement-de-developpement-et-de-programmation-arduino-et-queelles-parties-a-t-il/#:~:text=%C3%80%20quoi%20s'agit%20,utilis%C3%A9s%20sur%20un%20plateau%20Arduino.>
- [32] :<https://www.cours-gratuit.com/cours-arduino/tutoriel-environnement-de-programmation-arduino-ide>
- [33] :<https://www.auditsi.eu/?p=7085>

- [34] :<https://french.alibaba.com/product-detail/smart-electronics-new-for-diy-smart-car-robot-reflective-photoelectric-3pin-ir-infrared-obstacle-avoidance-sensor-module-60573629048.html>
- [35] :<http://qqtrading.com.my/ir-infrared-obstacle-detection-sensor-module-fc-5>
- [36] :<https://www.elprocus.com/infrared-ir-sensor-circuit-and-working/>
- [37] :<https://i.pinimg.com/originals/ea/03/6d/ea036d6f3b25e8eb0f15b0ca1a81060f.jpg>
- [38] :<http://idehack.com/blog/tutoriel-arduino-module-ir-proximity-sensor/>
- [39] :<https://plaisirarduino.fr/afficheur-lcd-comment-lexploiter/>
- [40] :[https://sti2d.ecolelamache.org/le\\_relais\\_lectromcanique.html](https://sti2d.ecolelamache.org/le_relais_lectromcanique.html)
- [41] :<https://www.aranacorp.com/fr/utilisation-dun-module-relais-avec-arduino/>
- [42] : Mlle Touala Aicha. Mlle Djali Hanane : Etude et réalisation d'un relais piloté par un son. Encadré par Mr Ali BELAID. Université Abou bakrBelkaïd– Tlemcen – Faculté de Technologie.2016-2017.
- [43] :<http://www.accofits.com/electrovannesvannesaccofits>
- [44] :<https://circuitdigest.com/microcontroller-projects/how-to-control-solenoid-valve-using-arduino>
- [45] :<https://www.astuces-pratiques.fr/electronique/le-relais-principe-de-fonctionnement>
- [46] :<https://fr.wikipedia.org/wiki/Convoyeur>
- [47] :[https://www.wikimeca.org/index.php/Moteur\\_%C3%A0\\_courant\\_continu](https://www.wikimeca.org/index.php/Moteur_%C3%A0_courant_continu)
- [48] :[http://colbertserv.lyceecolbert-tg.org:3007/cours\\_motorisation\\_electrique/viewer/visu.php?f=82#:~:text=Le%20principe%20de](http://colbertserv.lyceecolbert-tg.org:3007/cours_motorisation_electrique/viewer/visu.php?f=82#:~:text=Le%20principe%20de)
- [49] :<https://www.astuces-pratiques.fr/electronique/le-relais-principe-de-fonctionnement>
- [50] :<https://fr.wikipedia.org/wiki/Pompe>
- [51] :<https://www.oieau.fr/ReFEA/fiches/pompes/PompesMotorisees.htm>
- [52] :<https://www.protruckshop.com/reference/65451020-herth-buss-elparts-pompe-d-eau-de-nettoyage-nettoyage-des-vitres-399432>
- [53] :[https://electronoobs.com/eng\\_circuitos\\_tut35.php](https://electronoobs.com/eng_circuitos_tut35.php)
- [54] :[https://fr.m.wikiversity.org/wiki/Transistor/Transistor\\_MOSFET](https://fr.m.wikiversity.org/wiki/Transistor/Transistor_MOSFET)
- [55] :<http://genelaix.free.fr/spip.php?article38>
- [56] :<https://www.astuces-pratiques.fr/electronique/les-transistors-mosfet-de-puissance>
- [57] :<https://www.google.dz/search?dcr=0&q=What+is+IRFZ44N%3F&sa=X&ved=2ahUKewjE0tGp0YjxAhXwyIUKHT-qAW8Qzmd6BAgkEAU>

## *Références*

---

[58] : <https://www.fluke.com/fr/apprendre/blog/installations-electriques/qu-est-ce-qu-une-diode>

[59] : <http://www.elektronique.fr/cours/composant-diode.php>

[60] : <http://projet.eu.org/pedago/sin/1ere/9-diode.pdf>

[61] : <https://www.theengineeringprojects.com/downloads?file=71151>

[62] : <https://eepower.com/resistor-guide/resistor-fundamentals/what-is-a-resistor/#>

[63] : Ziouche Djawed, Mebrek Salah Eddine : Réalisation d'un capteur de présence piloté par Arduino, Université Aboubakr Belkaïd – Tlemcen – Faculté de TECHNOLOGIE, 2016

[64] : Aftis Fatah, Akkouche Merzouk. Conception et simulation d'un multi-afficheur pour une ligne de production. Bejaï99a : Université A/MIRA Bejaïa, Faculté de Technologie, 2014-2015

[65] : <http://blewando.fr/elv/Promo2016/th4/pag1.html>.

## Résumé

Les machines de remplissages de bouteilles automatiques sont le plus souvent utilisées dans les industries des boissons.

Le travail présenté dans ce mémoire, la simulation et la réalisation d'un système de remplissage des bouteilles automatique en utilisant la carte Arduino.

Notre prototype est constitué de plusieurs périphériques tels qu'électrovanne, pompe, moteur.....etc.

Ces périphériques sont contrôlés par une carte Arduino UNO.

En effet, lors de la détection des bouteilles le microcontrôleur de la carte Arduino exploite ces informations pour activer le circuit d'action (électrovanne-pompe).

**Mots clé :** capteur IR-Arduino.

## Abstract

Automatic Bottle Filling Machines are most commonly used in beverages and soft drink industries.

The work presented in this memory, is simulation and realization of an automatic bottlefilling system using Arduino UNO.

Our prototype consists of several peripherals such as solenoid valve, pump, motor... these peripherals are controlled by an Arduino UNO board.

Indeed, when the bottles are detected, the microcontroller of Arduino board uses these informations to activate the action circuit.

**Keywords :** sensor- Arduino.

## الملخص

غالبًا ما تستخدم آلات تعبئة الزجاجات الأوتوماتيكية في صناعات المشروبات.

العمل المقدم في هذه المذكرة هو محاكاة وتحقيق نظام تعبئة الزجاجات الأوتوماتيكية باستخدام لوحة أرد وينو.

يتكون نموذجنا الأولي من العديد من الأجهزة الطرفية مثل صمام الملف اللولبي والمضخة والمحرك... يتم التحكم في هذه الأجهزة الطرفية

بواسطة لوحة أرد وينو.

في حالة استشعار القارورات. تستخدم وحدة التحكم الدقيقة هذه المعلومات لتنشيط دائرة العمل.

الكلمات المفتاحية: حساس- أرد وينو