

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLEB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الآلية و إلكتروتفني
Département d'Automatique et électrotechnique



Mémoire de Master

Filière : Automatique

Spécialité : Automatique & Informatique Industrielle

Présenté par

KOUTCHOUK Oussama

Conception et Réalisation d'un système de pesage automatique pour le contrôle de qualité des bouteilles de shampoing VENUS

Proposé et dirigé par : - Pr. Guessoum Abderrezak

- Mr. Dikhai Abdellatif

Année Universitaire 2021-2022

Remerciements

Je remercie Dieu le tout puissant de m'a donné le courage et la volonté de parvenir à la fin de notre parcours universitaire.

Je voudrais à remercier sincèrement Mr GUESSOUM Abderrezak, En tant qu'Encadreur, pour ses précieux conseils et son orientation durant tout le long de mon travail.

Mes remerciements vont aussi à tous les enseignants de l'université Saad Dahleb de Blida 1 qui ont contribué à notre formation.

Je remercie également le responsable et CO-encadreur de l'usine VENUS qui m'ont aidé durant toute la période de mon stage, et à toutes les personnes qui m'ont aidées de près ou de loin au sein de l'usine VENUS EMBALAGE.

Je voudrais aussi exprimer mes remerciements aux membres du jury d'avoir fait l'honneur d'accepter d'examiner ce travail.

Enfin, j'adresse mes plus sincères remerciements à mon père, mère, frère et sœurs, et à tous mes proches et amis, qui m'a toujours soutenu et encouragé au cours de la réalisation de ce mémoire.

Dédicaces

Je dédie ce mémoire à Ma Mère La plus belle créature que Dieu a créée sur terre à cette source de tendresse, de patience et de générosité ! Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

A Mon Père Qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi Aucune dédicace ne saurait exprimer l'amour, l'estime, le dévouement et le respect que j'ai toujours eu pour vous.

A mon cher frère et mes chères sœurs ceux qui ont partagé avec moi tous les moments d'émotions lors de la réalisation de ce travail. Ils m'ont chaleureusement supporté et encouragé tout au long de mon parcours.

Et à la fin, une dédicace spéciale à tous nos enseignants durant notre cursus scolaire, du primaire à l'université. A tous les musulmans dans le monde entier A tous ceux qui, par un mot, m'ont donné la force de continuer.

Résumé

Ce projet de fin d'étude a pour but en première lieu le développement d'un programme informatique pour la simulation et réalisation d'un pesage automatique en ligne, en convoyeur pour les bouteilles du shampoing fabriquées au sein de l'usine VENUS en respect des normes en vigueur.

Ce projet est basé sur l'Automate Programmable Industrielle Siemens S7-300, le logiciel TIA PORTAL et PLCSIM pour le programme et la simulation 3D en FACTORY IO.

Et pour la réalisation est basé sur l'Arduino

Mots Clés : TIA PORTAL / FACTORY IO / API / SIEMENS / ARDUINO / PESAGE AUTOMATIQUE

ملخص :

هذا المشروع يهدف إلى تطوير برنامج لإنجاز و محاكاة لوزن القارورات الشامبو في شركة فينوس على خط الحزام الناقل تلقائيا بدون تدخل الإنسان. و يجب ان تكون مطابقة لمعايير الوزن.

يحتاج هذا المشروع إلى

API S7-300 و برنامجي TIA PORTAL و PLCSIM للبرمجة و FACTORY IO لمحاكاة ثلاث أبعاد و أردوينو .

الكلمات المفتاحية : جهاز تحكم منطقي قابل للبرمجة / TIA PORTAL / FACTORY IO / الوزن التلقائي / أردوينو

ABSTRACT:

The Objective Of this project is to develop a program to simulate and realize it so he can calculate the shampoo's bottle weight in the belt conveyor that made in VENUS Industry. So, the bottles it could be in the standards weight.

This project is based on the Siemens S7-300 Programable logic controller, the Software TIA Portal for program and the 3D FACTORY IO simulation with ARDUINO.

Keywords: TIA PORTAL / FACTORY IO / PLC / SIEMENS / ARDUINO / AUTOMATIC WEIGHTING

Table des matières

Introduction générale	1
-----------------------------	---

Chapitre 1 : Présentation générale sur l'entreprise et les systèmes automatisés industriels

Partie 1 : Présentation générale sur l'entreprise

1.1 Introduction.....	4
1.2 Historique et Situation Géographique.....	4
1.3 Organigramme de l'entreprise	7
1.4 Différentes machines de fabrication.....	9
1.4.1 Injection.....	9
1.4.2 Injection-soufflage.....	11
1.4.3 Extrusion.....	12
1.4.4 Extrusion soufflage.....	13
1.5 Différentes étapes de fabrication	15
1.6 Problématique	19

Partie 2 : Généralités sur les systèmes automatisés industriels

1.7 Introduction	19
1.8 Les Systèmes Automatisés	19
1.9 Avantages et inconvénients de l'automatisation.....	20
1.9.1 Les Avantages	20
1.9.2 Les Inconvénients.....	20
1.10 Structure d'un système automatisé.....	21
1.10.1 Partie opérative.....	22
1.10.2 Partie commande.....	24
1.10.3 Partie pupitre	24
1.11 Le rôle de la partie commande	24
1.12 La solution proposée à la problématique.....	25

1.13 Conclusion	25
-----------------------	----

Chapitre 2 : Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

2.1 Introduction	27
2.2 Généralités sur l'automate programmable	27
2.3 Principe de fonctionnement d'un automate programmable	28
2.4 Aspects externes	29
2.4.1 Compacte	29
2.4.2 Modulaire	30
2.5 Aspect interne	30
2.5.1 Module d'alimentation	30
2.5.2 Unité de traitement ou processeur	30
2.5.3 Mémoire programme	31
2.5.4 Mémoire de données	31
2.5.5 Interface d'entrée	31
2.5.6 Interface de sortie	31
2.6 Critère de choix d'un API	32
2.7 Langage de programmation des API	32
2.8 Présentation de SIMATIC S7-300	34
2.8.1 Composants d'un S7-300	34
2.8.2 Configuration de sous-réseaux	38
2.9 Tia portal	38
2.9.1 Vue du projet et vue du portail	38
2.9.2 Instructions structurées étape par étape	40

2.9.3	Le matériel de SIMATIC S7-300	42
2.9.4	Enregistrer et compiler la configuration matérielle.....	45
2.9.5	Télécharger la configuration matérielle sur l'appareil.....	45
2.9.6	Simulateur S7-PLCSIM.....	47
2.10	Factory I/O.....	49
2.10.1	Introduction	49
2.10.2	L'interface du logiciel.....	50
2.10.3	fenêtre de la palette.....	50
2.10.4	les pilotes (Drivers)	51
2.10.5	Configuration.....	53
2.11	Conclusion	54

Chapitre 3 : Présentation du Programme en Tia Portal v13 et la simulation en Factory I/O

3.1	Introduction	56
3.2	Système à pesage (Description générale du système)	56
3.3	Panneau de commande	57
3.3.1	Définition	57
3.3.2	Structure du panneau de commande	57
3.4	Les actionneurs et capteurs du système	59
3.4.1	Les actionneurs (sorties)	59
3.4.2	Les capteurs (entries)	59
3.4.3	Sortie analogique	59
3.4.4	Entre analogique	60
3.5	Connexion entre FACTORY IO et PLCSIM	65

3.6	Programmation sur TIA Portal V13.....	70
3.6.1	Configuration matérielle.....	70
3.6.2	Table des mnémoniques (Les variables d'API)	71
3.6.3	Le programme d'API.....	72
3.7	Conclusion.....	80

Chapitre 4 : Présentation de la carte Arduino et Réalisation du Projet

4.1	Introduction.....	82
4.2	Définition de la carte Arduino.....	82
4.2.1	Le Microcontrôleur ATmega328.....	83
4.2.2	Caractéristiques de la carte Arduino UNO	83
4.3	Programmation avec l'IDE Arduino.....	84
4.3.1	Lancement du logiciel	84
4.3.2	L'enivrement de la programmation.....	85
4.3.3	Structure générale du programme (IDE Arduino).....	85
4.4	Injection du programme	87
4.5	Description du programme	88
4.5.1	Commentaires.....	88
4.5.2	Les Variables	89
4.5.3	Configuration des entrées et des sorties void setup ().....	89
4.5.4	Programmation des interactions void loop ()	89
4.6	Niveaux logique de connecteur numériques.....	89
4.7	Les étapes de téléchargement du programme.....	90
4.8	Les composants utilisé dans la carte Arduino.....	91

4.8.1	Les câbles.....	91
4.8.2	BreadBoard (plaque d'essai)	92
4.8.3	Le Servomoteur.....	92
4.8.4	Capteur infrarouge.....	93
4.8.5	Cellules de charge à jauge de contrainte	94
4.8.5.1	Pont de Wheatstone	96
4.8.6	MODULE HX711.....	97
4.8.7	Afficheur LCD 16x2 (Liquide Crystal Display)	98
4.8.8	Buzzer (beeper)	99
4.8.9	Potentiomètre (résistance variable)	99
4.8.10	bouton poussoir	99
4.8.11	LED	100
4.9	La réalisation du projet avec arduino.....	101
4.10	Montage et câblage	101
4.11	Programme de Arduino	104
4.12	Conclusion.....	107
	Conclusion générale.....	108

Liste des figures

Chapitre 1

Figure 1.1 : Situation géographique de UEP VENUS satellite (Google Maps.....	6
Figure 1.2 : Situation géographique de UEP VENUS (Google Maps.....	6
Figure 1.3 : machine soufflage-extrudeuse	7
Figure 1.4 : Fonctionnement de la machine à injection.....	10
Figure 1.5 : <i>Différentes</i> séquences d'injection-soufflage.....	12
Figure 1.6 : Principe de fabrication d'une bouteille en extrusion-soufflage.....	13
Figure1.7 : Granules de thermoplastique du type PEHD.....	14
Figure 1.8 : Colorant blanche utilisé pour les bouteilles de shampooing VENUS.....	15
Figure 1.9 : Les bouteilles recyclé	16
Figure 1.10 : Scale weight blender.....	16
Figure 1.11 : Lucy dans l'industrie de chocolat	20
Figure 1.12 : Emballage de chocolat automatique.....	21
Figure 1.13 : Schéma représente la structure d'un système automatisé.....	21
Figure 1.14 : Schéma de la partie opérative.....	22
Figure 1.15 : Types d'actionneurs en automatisation industrielle.....	23
Figure 1.16 : Fonction d'un capteur.....	23
Chapitre 2	
Figure 2.1 : Fonctionnement d'un API.....	28
Figure 2.2 : API compact.....	29
Figure 2.3 : API modulaire.....	30

Figure 2.4 : Les langages du norme CEI 61131 associés à la programmation des automates.....	32
Figure 2.5 : Exemple de montage : composants d'un S7-300.....	34
Figure 2.6 : Vue du Portail.....	39
Figure 2.7 : Vue du Projet.....	40
Figure 2.8 : Icon TIA Portal V13.....	40
Figure 2.9 : Création un nouveau projet au menu.....	41
Figure 2.10 : Modification le nom du projet.....	41
Figure 2.11 : Appareils et réseaux.....	42
Figure 2.12 : Choisir un processeur.....	43
Figure 2.13 : Les instructions et bibliothèques et pour ajoute matériel.....	43
Figure 2.14 : Les composants d'un projet.....	44
Figure 2.15 : Zones d'adressage.	44
Figure 2.16 : Barre de menu.....	45
Figure 2.17 : Navigateur du projet.....	45
Figure 2.18 : Téléchargement étendu.	46
Figure 2.19 : Charger l'aperçu.....	47
Figure 2.20 : Simulateur S7-PLCSIM.....	48
Figure 2.21 : Factory IO.....	49
Figure 2.22 : Barre de menu.....	50
Figure 2.23 : Fenêtre de la palette.....	51
Figure 2.24 : Barre des pilotes.....	52

Figure 2.25 : Menu du pilote.....	52
Figure 2.26 : Barre de configuration	53
Figure 2.27 : La fenêtre de configuration	54
Chapitre 3	
Figure 3.1 : Panneau de commande	58
Figure 3.2 : Système de pesage.....	58
Figure 3.3 : La partie de démarrage	60
Figure 3.4 : Convoyeur de peseur.....	60
Figure 3.5 : La partie de direction vers le recyclage.....	62
Figure 3.6 : Pusher pousse la boite de 3kg vers recyclage.....	62
Figure 3.7 : Capteur photoélectrique de proximité.....	63
Figure 3.8 : La partie de boites au poids moyennes à deux faces.....	64
Figure 3.9 : Panneau de commande complète.....	65
Figure 3.10 : Les actionneur et capteurs utilisé dans FACTORY I/O.....	65
Figure 3.11 : Simulation du programme au PLCSIM.....	66
Figure 3.12 : Chargement du programme dans dispositif.....	66
Figure 3.13 : Démarrage du Monitoring.....	67
Figure 3.14 : Run le PLCSIM.....	67
Figure 3.15 : Changement entre le mode édition et le mode exécution	68
Figure 3.16 : FACTORY IO été connecté avec PLCSIM.....	68
Figure 3.17 : Les affichages du compteur et du poids.....	69
Figure 3.18 : Pusher pousse une boite lourde	69
Figure 3.19 : Configuration matérielle.....	70

Figure 3.20 : Variables API 1/2.	71
Figure 3.21 : Variables API 2/2.....	71
Figure 3.22 : Démarrage du système	72
Figure 3.23 : Allumer et éteindre les voyant vert et rouge.....	73
Figure 3.24 : Démarrage des convoyeurs.....	73
Figure 3.25 : Convertir l'entier IN en un nombre à virgule flottante.....	74
Figure 3.26 : Modification du matériel dans le TIA PORTAL pour être compatible avec le peseur.....	75
Figure 3.27 : Arrondissement de la valeur obtenu du poids.....	75
Figure 3.28 : La comparaison d'objet pour être le poids dans les normes.....	76
Figure 3.29 : L'activation de voyant et Pusher.....	76
Figure 3.30 : Addition à la valeur de poids.....	77
Figure 3.31 : Affichage du poids.....	78
Figure 3.32 : Comptage des boites Lourde.....	79
Figure 3.33 : Comptage des boites légère.....	79
Figure 3.34 : Comptage des boites moyenne.....	80

Chapitre 4

Figure 4.1 : La Carte ARDUINO UNO.....	83
Figure 4.2 : Microcontrôleur ATmega328.....	83
Figure 4.3 : Icône exécutable de l'Arduino.....	84
Figure 4.4 : Le démarrage de l'Arduino software.....	84
Figure 4.5 : Interface IDE Arduino.....	86
Figure 4.6 : Sélection de carte Arduino.....	87

Figure 4.7 : Sélection du port.....	87
Figure 4.8 : Définir les boutons de programmation	90
Figure 4.9 : Compilation terminée.....	90
Figure 4.10 : Téléversement terminé.....	91
Figure 4.11 : Câble USB type A-B	91
Figure 4.12 : Jumper wires (Les fils de liaison)	92
Figure 4.13 : BreadBoard (plaque d'essai)	92
Figure 4.14 : Servomoteur sg90.....	93
Figure 4.15 : Détecteur d'objet capteur infrarouge	94
Figure 4.16 : Diagramme de cellule de charge à jauge de contrainte.....	95
Figure 4.17 : Diagramme plus détaillé des jauges de contrainte sur les cellules de charge à barres lorsqu'une force est appliquée.....	95
Figure 4.18 : Cellule de charge à barre entre une configuration à deux plaques.....	96
Figure 4.19 : Le circuit du pont de Wheatstone.....	96
Figure 4.20 : Jauge de contrainte.....	97
Figure 4.21 : Module HX711.....	97
Figure 4.22 : Interface de LCD 16x2.....	98
Figure 4.23 : Potentiomètre.....	99
Figure 4.24 Symboles conventionnels des boutons.....	100
Figure 4.25 Bouton poussoir.....	100
Figure 4.26 Les Leds.....	100
Figure 4.27 : branchement du jauge contrainte et HX711	101
Figure 4.28 : Câblage les composants avec Arduino.....	102
Figure 4.29 : le Programme de Arduino.....	107

Liste des tableaux

Tableau 2.1 Les composants de montage d'un S7-300.....	35
Tableau 2.2 Composants d'un S7-300.....	38
Tableau 3.1 Panneau de commande	57
Tableau 4.1 : Caractéristiques de la carte Arduino UNO.....	84
Tableau 4.2 : Définir le montage des composants.....	104

Listes des acronymes et abréviations

API : Automate Programmable industrielle

CEI : Communauté des États indépendants

ES : Entrée Sortie

IDE: integrated development environment

IO: Input Output

IR : infrarouge

ISO: International Organization for Standardization

PC : Partie Commande

PEBD : Polyéthylène Basse Densité

PEHD : Polyéthylène Haute Densité

P-DG : Président Directeur Générale

PLC : Programmable Logic Controller

PO : Partie Opérative

PP : Polypropylène

PWM: Pulse Width Modulation

RAM: Random Access Memory

SAPECO : Société Algérienne de Produits d'Entretien et Cosmétique

SFC : Sequential Function Chart

TIA PORTAL: Totally Integrated Automation Portal

TOR : tout ou rien

UEP : Unité Emballage Plastique

Introduction générale :

Actuellement le monde industriel doit offrir des produits de qualité dans des délais courts et à des prix compétitifs. Avec la progression continue de la technologie, les critères de demande ne s'arrêtent pas uniquement à l'augmentation de la productivité mais assurent aussi une qualité supérieure de ces produits, l'amélioration des conditions de travail et l'accroissement de la sécurité . Mais ces objectifs ne peuvent être atteints que s'ils respectent l'exigence de l'industrie moderne que l'on peut qualifier d'industrie de qualité et de quantité qui ne cesse elle aussi d'exiger un matériel de contrôle de plus en plus performant.

Le domaine de l'automatisation industrielle a facilité les processus de contrôle de diverses machines, composants et dispositifs, Les systèmes automatisés éliminent la plupart des problèmes et des obstacles au développement de la technologie industrielle

Dans le cadre d'un stage de deux mois au laboratoire VENUS à Blida, nous avons pu constater qu'au niveau de la ligne de production des bouteilles, il faut améliorer et automatiser les systèmes de pesage automatique qui ne respectent pas les normes, notamment au niveau de la zone de contrôle de qualité, Le travail se fait manuellement, d'où une faible productivité et une perte de temps et de l'argent. Cette observation est la principale motivation pour choisir le sujet de notre thèse **Le pesage automatique** .

L'objectif principal de notre travail est de faire une simulation et commander un tapis de peseur commandé par un Automate Programmable Industriel S7-300. Ce dernier a pour fonction de traiter les information entrantes pour émettre des ordres de sorties en fonction d'un programme dans TIA PORTAL 'LADDER' C'est un automate programmable à usage industriel destiné à commander les systèmes en temps réel.

Pour aboutir à cette fin, nous avons organisé notre travail en quatre chapitres :

- Introduction
- Chapitre 1 : Présentation générale sur l'entreprise et les systèmes automatisés industriels.
- Chapitre 2 : Les automates programmables industriels, Programmation TIA PORTAL V13 et Simulation FACTORY IO
- Chapitre 3 : Présentation du Programme en Tia Portal v13 et la simulation en Factory IO
- Chapitre 4 : Présentation de la carte Arduino et Réalisation du Projet
- Conclusion Générale

Chapitre 1 : Présentation générale sur l'entreprise et les systèmes automatisés industriels

Partie 1 : Présentation générale sur l'entreprise

1.1 Introduction

Ce chapitre présente la structure dans laquelle nous avons effectué notre stage de fin d'étude. Il donne un aperçu sur les différents types de plastiques et de machines, de son historique, sa situation géographique et son organisation interne.

Et d'introduire les systèmes automatisés et leurs avantages dans nos vies et surtout dans l'industrie pour simplifier les tâches de l'être humain ou bien sans intervention d'humaine comme l'emballage ou bien le pesage automatique dans notre cas.

1.2 Historique et Situation Géographique :

Le Laboratoires VENUS (SAPECO) / Société Algérienne de Produits d'Entretien et Cosmétique a été fondée par son actuel P-DG Mourad Moula en 1981. Présentes depuis 40 ans sur le marché Algérien, les laboratoires venus développent leur expertise autour d'une politique d'innovation et de mise à niveau permanente pour offrir aux consommateurs des produits de qualité au meilleur prix, Avec leur expertise, Les Laboratoires Vénus œuvrent pour la satisfaction des consommateurs en les mettant au centre de ses intérêts et ce, grâce à un développement d'activité visant la préservation de l'environnement naturel, la santé et la sécurité de ses consommateurs mais aussi de son équipe. [1]

La science de la mesure chez Laboratoires Venus, exige pour chaque entreprise, d'assurer une vérification permanente de ses équipements de mesure. C'est le cas des laboratoires Venus qui ont créé en 2014 un laboratoire métrologique offrant des services en interne ainsi qu'en externe. Ce dernier, fait appel à des références pour contrôler régulièrement les moyens et instruments de mesure.

En mars 2020, Laboratoires VENUS se lancent dans la fabrication de gel hydro-alcoolique pour lutter contre la pandémie de coronavirus en Algérie et elle fait un don de 10 000 unités aux hôpitaux de la wilaya de Blida.

Son activité initiale qui a débuté avec trois (03) shampoings, a vite évolué vers une plus grande variété de produits cosmétiques et d'hygiène corporelle, qui englobe présentement plus de quatre vingt (80) produits. Les produits de soin dermique sont fabriqués sous licence en collaboration avec l'Entreprise française SOFIA / COSMÉTIQUES.

L'entreprise a obtenu deux certifications ISO pour le système management :

- ISO 9001 qui définit les critères pour un système de qualité de management international.
- ISO 14 001 qui définit les critères pour un système de management respectueux de l'environnement. [3]

Les Laboratories Venus se sont vus discerner plusieurs récompenses :

- 2003 : Prix Algérien de la Performance
- 2011 : Prix d'Encouragement pour export
- 2012 : Prix de l'Innovation
- 2015 : Prix Algérien de la Qualité

Afin de faire face aux insuffisances inhérentes aux emballages produits notamment localement, les Laboratoires VENUS ont décidé de se prendre en charge par ses moyens propres. De ce fait, les Laboratoires VENUS ont engagé un projet d'investissement appréciable pour la création d'une unité d'emballage située au niveau de la Zone Industrielle ATLAS sur la Route de Beni Tamou-Blida. [1]



Figure 1.1 Situation géographique de UEP VENUS satellite (Google Maps)

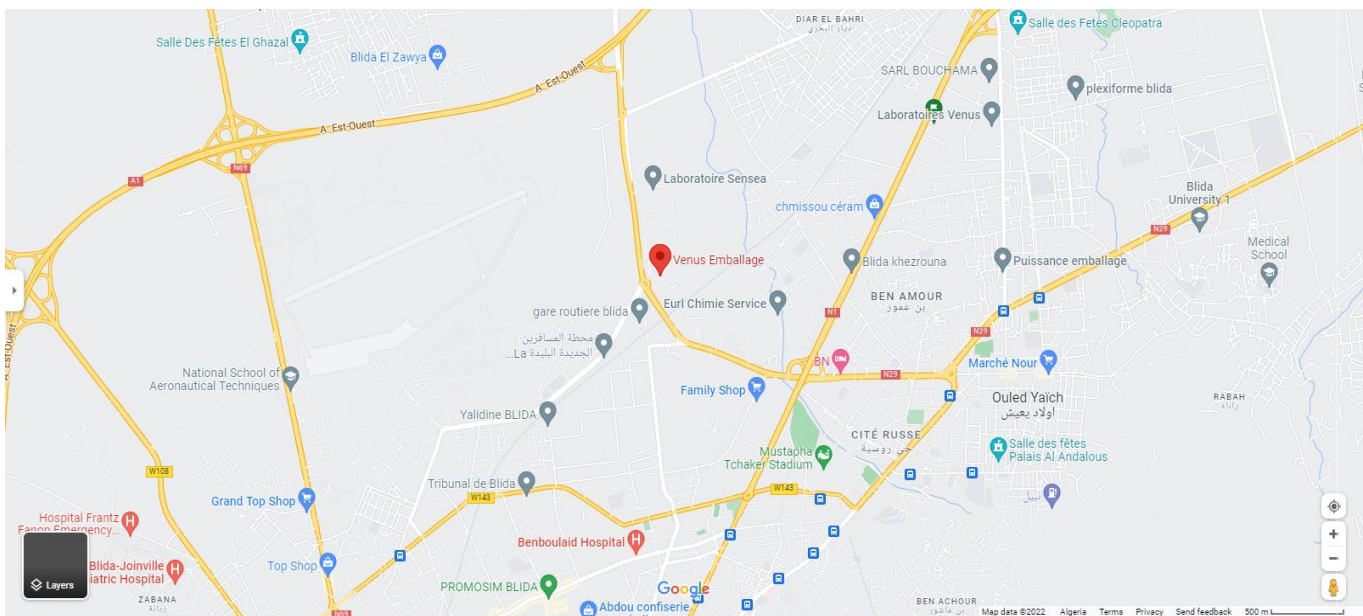


Figure 1.2 Situation géographique de UEP VENUS (Google Maps)

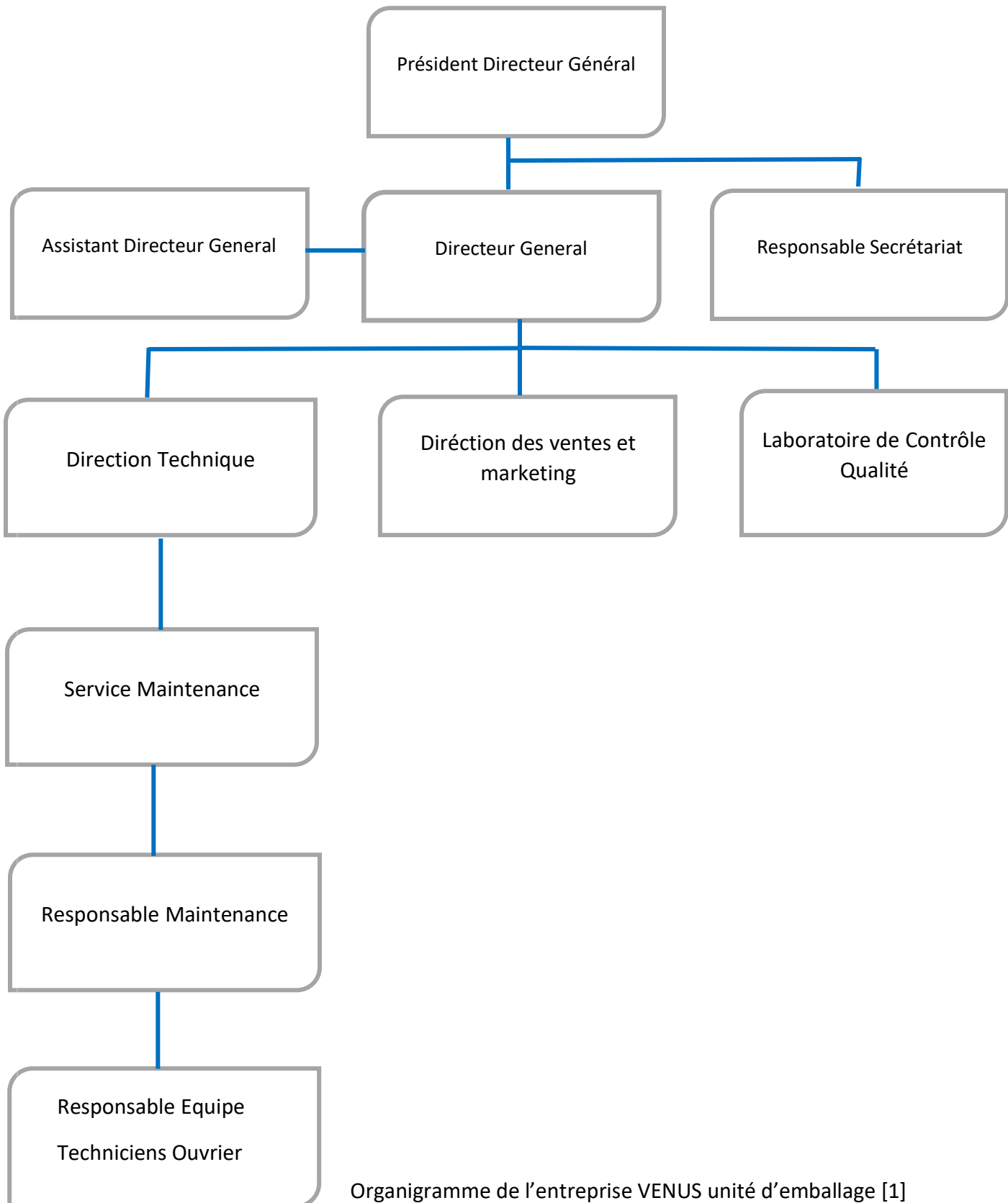
VENUS Unité Emballage Plastique (UEP), a débuté son activité en Mars 2010, comme Filiale des Laboratoires. VENUS SAPECO, Elle est spécialisée dans l'injection et le soufflage des emballages en plastique (flacons, bouchons) en PEHD et PP. Elle dispose de la dernière génération de machines électriques souffleuses (MAGIC - Italiennes) et des presses à injection (françaises). [1]



Figure 1.3 : machine soufflage-extrudeuse

1.3 Organigramme de l'entreprise

L'organigramme de l'usine se présente comme suit :



1.4 Différentes machines de fabrication :

1.4.1 Injection

Dans le moulage par injection, la matière est fluidisée dans un organe approprié puis injectée dans l'empreinte (ou empreinte) du moule sous haute pression. L'une des caractéristiques les plus populaires du moulage par injection est la capacité de travailler à des vitesses élevées. Pour des objets de taille moyenne (quelques centaines de grammes), le temps de cycle est de quelques dizaines de secondes. Pour les objets fins, une fraction de seconde et les objets très épais des dizaines de minutes. En raison de la mauvaise conductivité thermique des plastiques, le temps de refroidissement détermine généralement la durée du cycle. Pour des raisons économiques, la limite supérieure de l'épaisseur de paroi des produits moulés par injection est d'environ 6 mm.

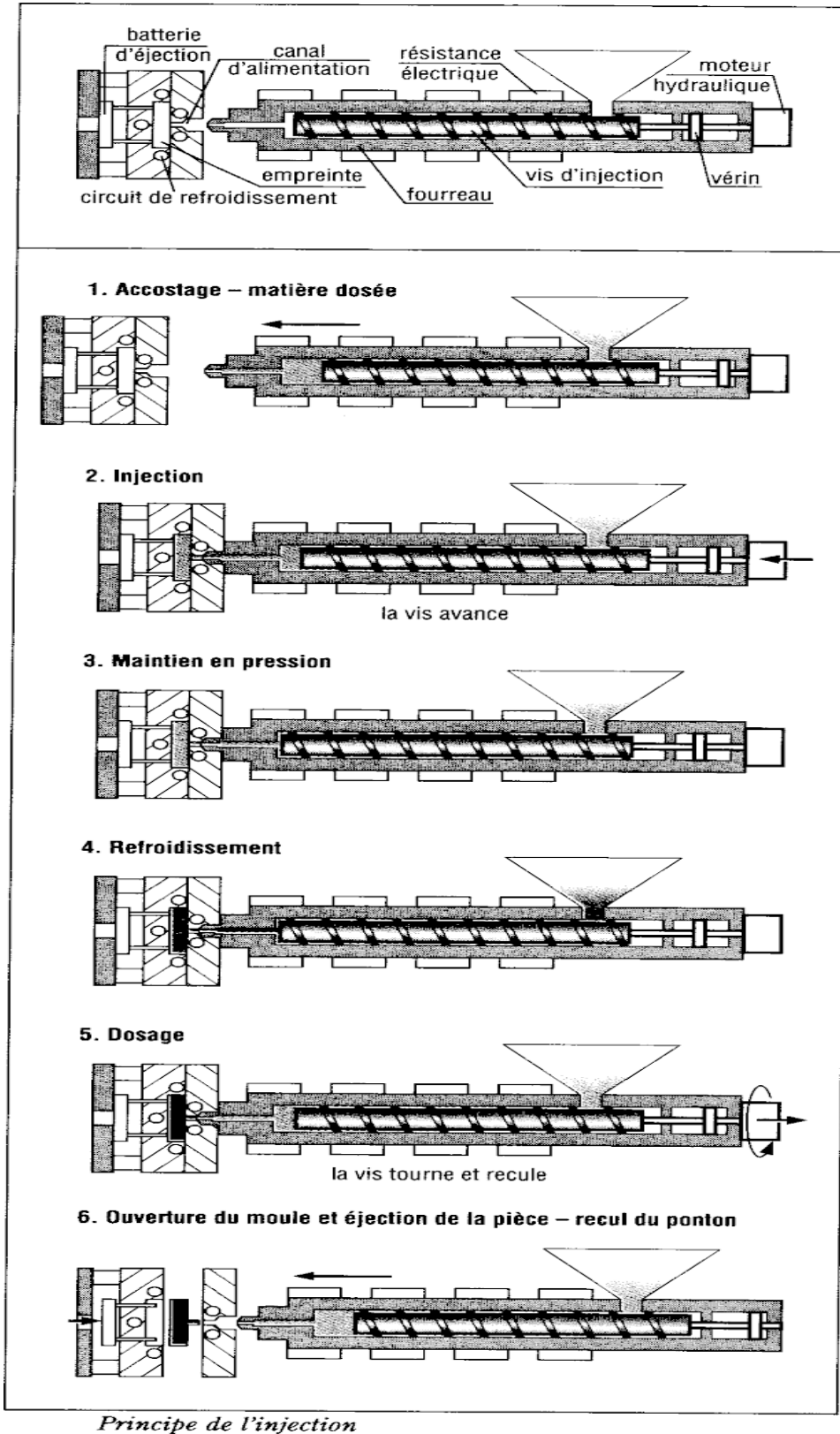


Figure 1.4 fonctionnement de la machine à injection [2]

1.4.2 Injection-soufflage

L'injection-soufflage permet de réaliser des corps creux qui présentent de bonnes propriétés mécaniques. Cette technique est essentiellement réservée aux thermoplastiques.

Le cycle de fabrication se compose de quatre étapes

- 1- Réalisation par injection d'une préforme.
- 2- Transfert de cette préforme dans une station de réchauffage pour que le matériau soit réchauffé dans un domaine d'état caoutchoutique.
- 3- Transfert de la paraison chaude dans un moule et soufflage pour que le polymère vienne en contact avec les parois du moule.
- 4- Refroidissement et éjection de la pièce.

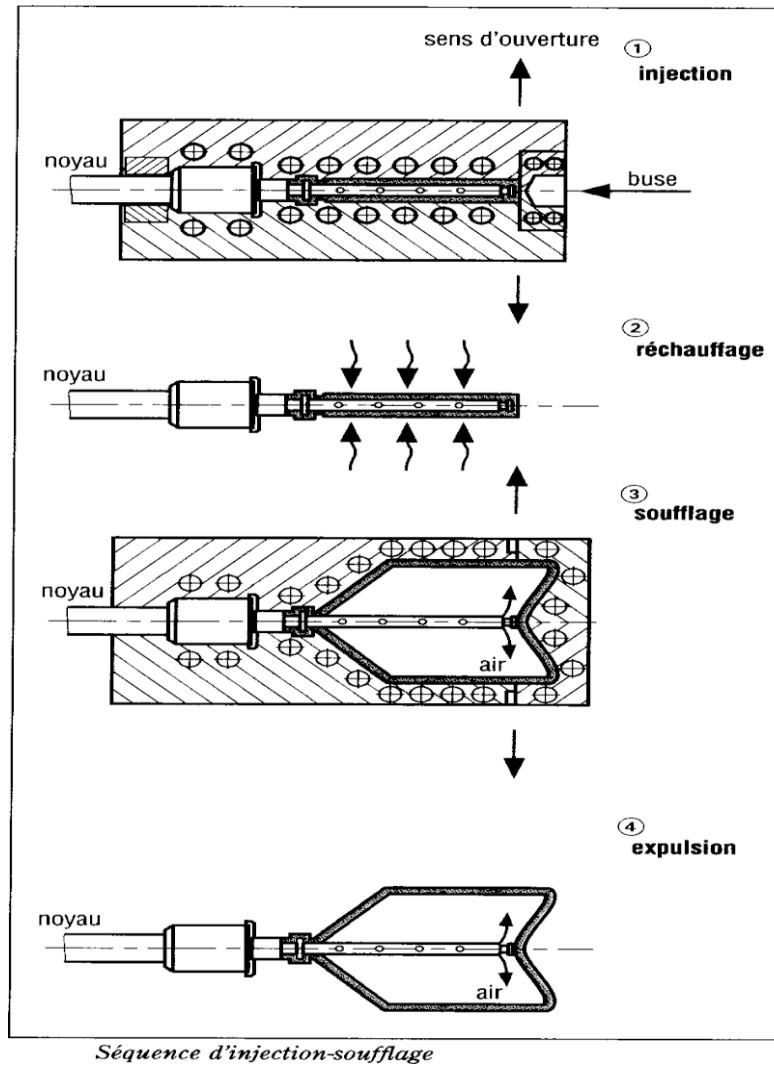


Figure 1.5 Différentes séquences d'injection-soufflage [2]

1.4.3 Extrusion

L'extrusion est une technologie de transformation de matériaux thermoplastiques, qui permet d'obtenir des produits finis ou semi-finis grâce à un processus technologique continu. La technologie permet d'élargir les types d'objets de fabrication en utilisant des matériaux adaptés, et d'élargir la gamme de produits des profilés rigides et flexibles à la fabrication de plaques, feuilles, films, objets creux, particules plastifiées, câbles multibrins et multicolores, laminés films et feuilles, profilés expansés et allégés, etc.

Presque tous les matériaux thermoplastiques sont extrudés au moins une fois dans leur processus de préparation, mais seule une partie d'entre elles sort sous forme de produit fini.

L'extrusion est largement utilisée dans la fabrication de particules et de composés, qui sont ensuite traités par d'autres méthodes (telles que l'injection) et de produits semi-finis, tels que des plaques, des feuilles, des films, des tuyaux, etc .

Quel que soit le principe de fonctionnement, chaque extrudeuse effectue trois opérations principales:

- Transport de la matière du point d'alimentation au moule le long du cylindre,
- Plastification grâce au mélange et à la chaleur apportée de l'extérieur,
- Augmentation progressive de la pression du matériau et la force à travers le moule pour le façonnage, puis la fixation par refroidissement. [2]

1.4.4 Extrusion soufflage

Le moulage par extrusion et soufflage est un procédé de mise en forme de pièces thermoplastiques creuses qui existe depuis les années 1950. Très utilisé pour le flaconnage, Ce procédé consiste à extruder à chaud un tube polymère appelé "paraison" et à le placer à l'intérieur d'un moule. On vient ensuite gonfler ce tube par injection d'air, ce qui permet de créer des pièces dont la forme correspond à l'empreinte du moule.

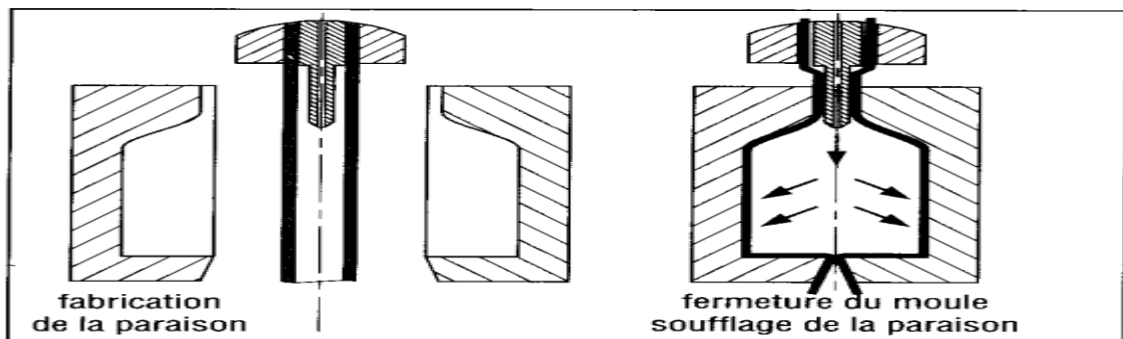


Figure 1.6 principe de fabrication d'une bouteille en extrusion-soufflage [2]

Voici les 3 principaux polymères utilisables :

- Polyéthylène Haute Densité (PEHD) le plus répandu, pour toutes applications
- Polyéthylène Basse Densité (PEBD) pour les pièces souples.
- polypropylène (PP) pièces d'aspect.



Figure1.7 Granules de thermoplastique du type PEHD

1.5 Différentes étapes de fabrication

La fabrication du plastique (les bouteilles du shampoing) par la machine extrudeuse-souffleuse passe par les étapes citées ci-dessous :

• ÉTAPE 0 : Équilibrage de trois matières

Pour fabriquer la paraison il y'a trois matières qui passera par le mesure (scale weight)

Ce sont :

- PEHD
- Colorant
- Les déchets des bouteilles (recycle)



Figure 1.8 colorant blanche utilisé pour les bouteilles de shampoing VENUS



Figure 1.9 Les bouteilles recyclé



Figure 1.10 scale weight blender

• ÉTAPE 1 : Extrusion de la paraison

La matière plastique est introduite sous forme de granulés dans une trémie. Ces granulés sont ensuite chauffés (jusqu'à une température de 200°C pour le polyéthylène) et ramollis dans une vis d'extrusion. Le vérin d'extrusion pousse alors la matière jusqu'à la tête contenant la filière d'extrusion. La matière extrudée prend alors la forme d'un tube appelé "paraison" d'une longueur définie, autour duquel est positionné le moule ouvert.

• ÉTAPE 2 : Fermeture du moule

On vient ensuite mettre en contact et refermer les deux parties du moule autour de cette paraison. Pour que la fermeture soit hermétique, la matière est collée à chacun des 2 bouts. Le corps creux hermétique ainsi formé est prêt à accueillir l'aiguille de soufflage.

• ÉTAPE 3 : Soufflage de la paraison

Il existe plusieurs techniques de soufflage de la paraison, par canne ou par aiguille de soufflage. Dans les 2 cas, le principe est le même : insuffler de l'air pour plaquer la matière contre les parois du moule. Ce moule est ensuite refroidi afin de figer la matière, tout en la maintenant sous pression.

• ÉTAPE 4 : Dégonflage et ouverture

Une fois la matière figée, on relâche la pression au niveau de l'aiguille : la pièce se dégonfle. Le moule s'ouvre alors et libère la pièce toujours fermée à ses 2 bouts.

• ÉTAPE 5 : Finition

Les deux parties qui servaient à maintenir la paraison hermétiquement fermée peuvent maintenant être retirées, à l'aide d'un outil coupant. Ces déchets de démoulage sont récupérés et réintroduits dans le cycle de production. La pièce peut également subir d'autres opérations de finition:

- Découpe.
- Ébavurage.
- Flammage.
- Perçage.
- Assemblages de plusieurs parties

•ÉTAPE 6 : Testage flacon

Ce dispositif, en soufflant l'air dans le goulot, fait le contrôle d'étanchéité des flacons en vérifiant l'absence de micro- trous. Par le pupitre de commande on peut régler tous les paramètres pour ce contrôle, en incluant temps de souffle et de contrôle. Il y a aussi des graphiques intuitifs qui montrent l'effective pression du contrôle et les tolérances indiquées en millibars. Le test peut bien être effectué individuellement en utilisant une sonde de précision pour chaque flacon ou collectivement en activant l'expulsion de tous les flacons d'un cycle.

•ÉTAPE 7 : Refroidissement

Le refroidissement des moules et des cannes pendant le cycle productif revêt une importance capitale dans la mesure où il influe directement sur la quantité et la qualité de la production. La durée de refroidissement est un paramètre variable ; Le refroidissement est une étape sensible, si la pièce est sortie trop rapidement, elle se déforme. Un refroidissement plus lent peut ainsi être nécessaire si la pièce doit comporter des parties parfaitement droites. La distribution de l'eau de refroidissement des moules est placée sur le chariot qui prévoit un nombre adéquat d'entrées et de sorties pour chacun des demi-coquilles du moule. Le raccordement s'effectuera au moyen de tuyaux flexibles et raccords.

1.6 Problématique :

Quand les bouteilles sont passées par toutes les étapes, quelques fois des bouteilles ont différent poids, en plus ou en moins, à cause des vis qui tournent avec une température plus ou moins élevée. Alors, un travailleur prend toutes les bouteilles. Il y'a environ 5 ou 6 machines qui produisent de 6 à 12 bouteilles chacune. Et il mesure les bouteilles manuellement chaque ½ heure. Cela prend du temps, et quelques bouteille qui passent avec un mauvais poids. Si le poids n'est pas dans les norme. les clients vont acheter plus de plastique que de shampoing. Il faut ainsi développer un système automatique de pesage.

Partie 2 : Généralités sur les systèmes automatisés industriels

1.7 Introduction :

Toutes les industries de production consistent en une série de machines simples ou compliquées qui fais la combinaison de matières premières, subir une séquence transformation et intégration afin de fabriquer un produit final.

1.8 Les Systèmes Automatisés :

Un système automatisé est capable de gérer des projets complexes. Il peut nous aider à augmenter la production. Une machine ou un système est dit automatisé lorsque le processus qui permet de passer d'une situation initiale à une situation finale se fait parfois sans intervention humaine et que ce comportement est répétitif chaque fois que les états qui caractérisent la situation initiale sont remplis. C'est un ensemble organisé de moyens techniques interconnectés des moyens de commande et de contrôle qui assurent un fonctionnement reproductible plus ou moins indépendant des interventions humaines. Brièvement, on peut dire que l'automatisation est un ensemble de procédés qui rend l'exécution d'une tâche automatique, après avoir reçu les consignes d'un opérateur. C'est un système qui réalise des opérations et pour lequel l'homme n'intervient que dans la programmation du système et dans son réglage par l'intermédiaire d'un pupitre. (Sans l'intervention humaine). [4]

1.9 Avantages et inconvénients de l'automatisation

1.9.1 Les avantages

- S'adapte facilement à tous les milieux de production.
- La souplesse d'utilisation.
- La création des postes pour les automaticiens.
- Accélération des processus de production, en gardant un produit de qualité.

1.9.2 Les inconvénients

- La suppression d'emplois.
- Le coût élevé du matériel.
- La maintenance doit être structurée.

Alors les avantages de l'automatisation c'est mieux et plus par rapport les inconvénients comme illustré dans le vidéo de 'Lucy and the chocolate factory'. C'est une usine d'emballage de chocolat manuellement, mais le problème est que le convoyeur ça marche vite et les boules de chocolat sont très nombreux alors y'a beaucoup de chocolat qui sont passe sans emballage.



Figure 1.11 Lucy dans l'industrie de chocolat

Mais l'emballage automatique avec des robots est très simple et plus rapide.

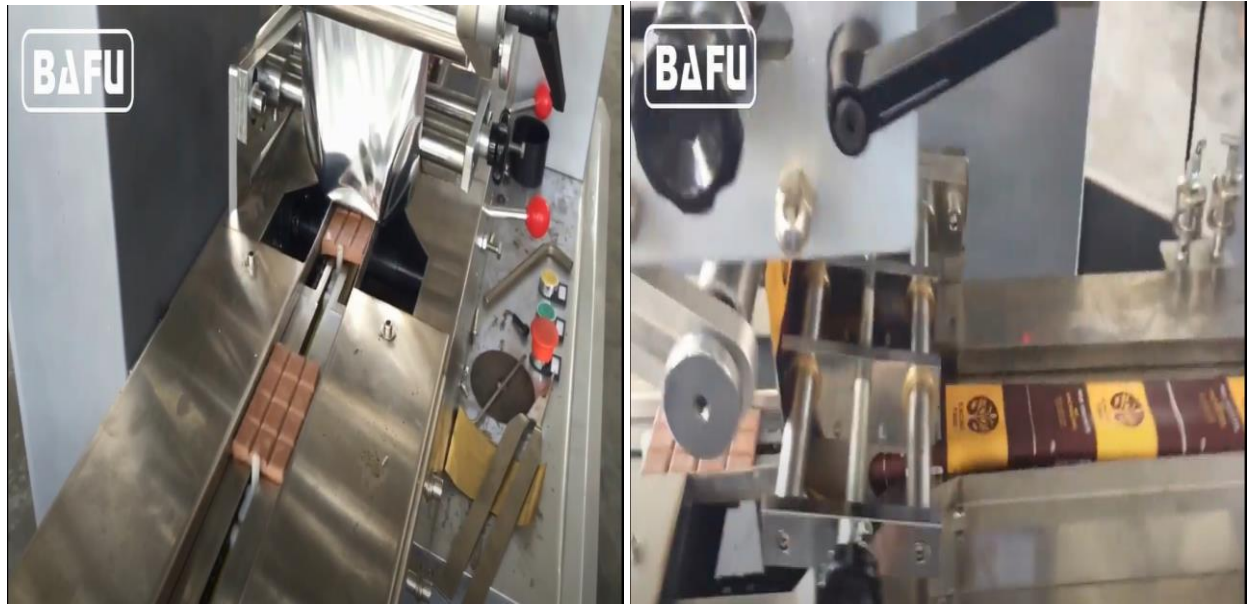


Figure 1.12 Emballage de chocolat automatique

1.10 Structure d'un système automatisé [4] :

Un système automatisé se représente sous la forme d'un schéma identifiant trois parties (PO, PC, PP) du système et exprimant leurs interrelations (information, ordre, compte rendu, consigne).

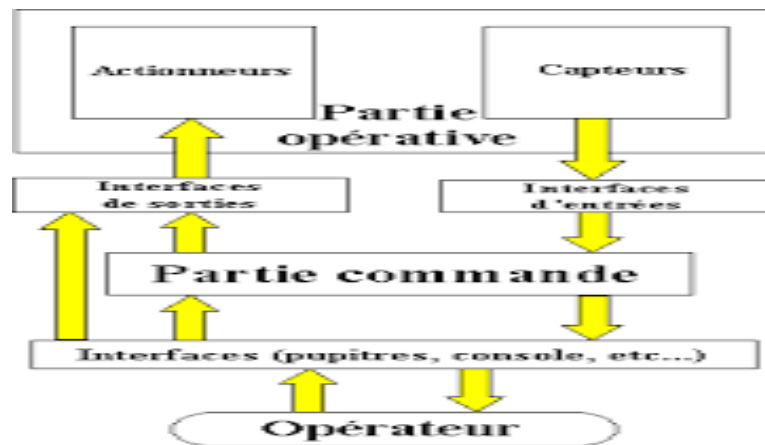


Figure 1.13 schéma représente la structure d'un système automatisé

1.10.1 Partie opérative (P.O) [4] :

Appelée aussi partie puissance, C'est la partie visible du système et le processus physique automatisé, elle est le siège des conversions énergétiques, reçoit les ordres de la partie commande et lui adresse des comptes rendus. Cette partie se compose d'un ensemble d'éléments qui sont : l'unité de production, qui a pour fonction de réaliser la fabrication ou la transformation dans le processus industriel, près actionneurs, actionneurs, effecteurs et capteurs, comme indique la figure ci-après.

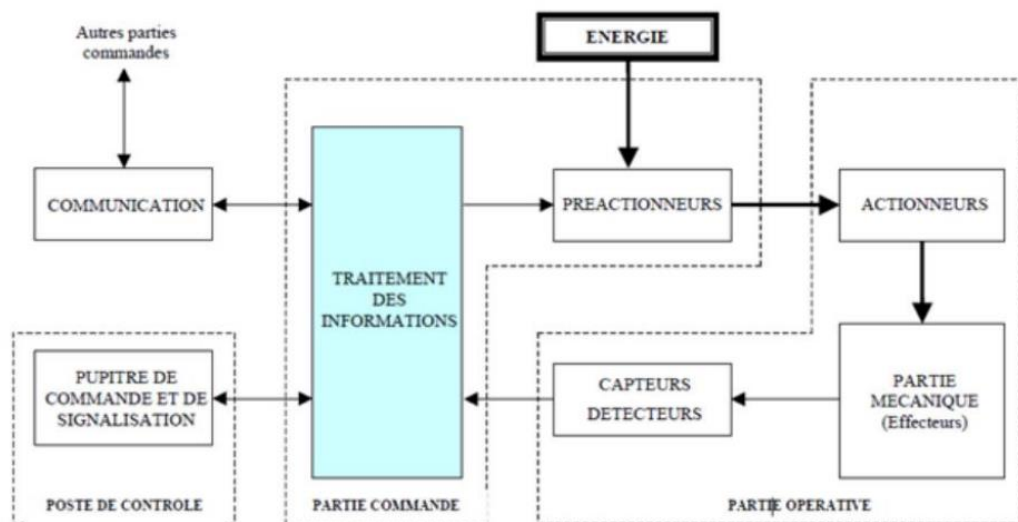


Figure 1.14 Schéma de la partie opérative

Elle comporte les éléments du procédé c'est-à-dire :

- **Pré-actionneurs :**

(distributeurs, contacteur) qui reçoivent des ordres de la partie commande.

- **Actionneurs :**

L'actionneur est l'organe de la partie opérative qui reçoit un ordre de la partie commande via un éventuel pré-actionneur, convertit l'énergie (pneumatique, hydraulique ou électrique) qui lui fournit en un travail utile l'exécution de tâche, on citera les moteurs, les vérins. Les types

d'actionneur en automatisation industrielle citée ci-dessous : (a) servomoteur, (b) moteur à courant continu, (c) moteur pas à pas, (d) vérin pneumatique

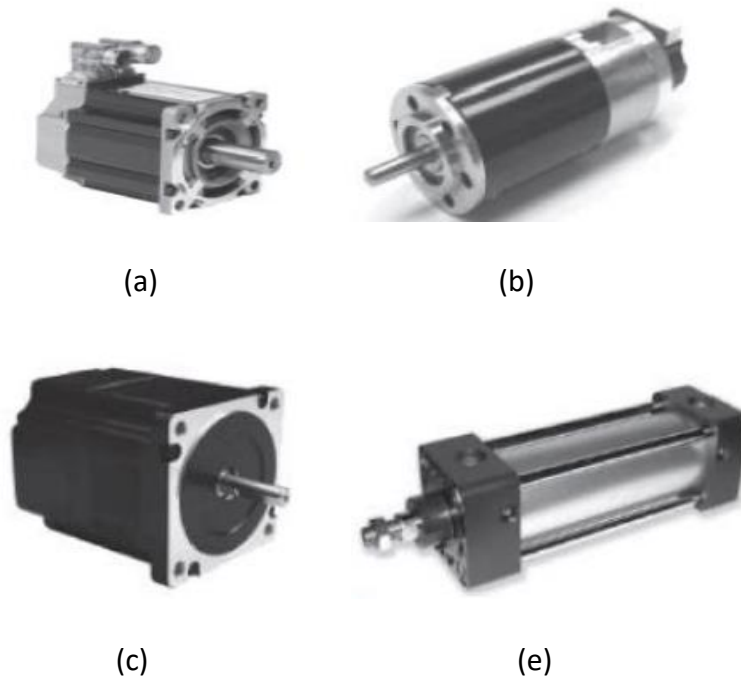


Figure 1.15 Types d'actionneurs en automatisation industrielle

- **Capteurs :**

Les capteurs sont des composants de la chaîne d'acquisition dans une chaîne fonctionnelle. Ils prélèvent une information (position, vitesse, pression...) et sont placés à la frontière entre la partie opérative et la partie commande et ainsi, ils détectent la position ou la présence des différents mobiles et transmettent ces informations à la partie commande.



Figure 1.16 Fonction d'un capteur

1.10.2 Partie commande (P.C) [4]

La partie commande d'un automatisme est le centre de décisions. Il reçoit des informations en provenance des capteurs de la partie opérative, et les restitue des ordres vers cette même partie opérative en direction des pré-actionneurs et actionneurs, elle peut se composer de trois parties : un ordinateur, un logiciel et une interface.

1.10.3 Partie pupitre (P.P) [4]

La partie opérative et la partie commande sont de nature différente, pour que les informations circulent correctement entre ces deux parties, on va utiliser un objet appelé interface. Ces interfaces sont en fait des sorties de traducteurs qui relient la partie commande et la partie opérative.

Le pupitre permet à l'opérateur de dialoguer et de commander la partie opérative.

Il comporte :

- Des capteurs de commande (marche, arrêt, arrêt d'urgence...).
- Des voyants de signalisation (mise sous tension, fonctionnement anormal, buzzer...).
- Des appareils de mesure de pression (manomètre), de tension (voltmètre), d'intensité (Ampèremètre).

1.11 Le rôle de la partie commande [4]

Le rôle de cette partie est :

- D'émettre les ordres de fonctionnement de la partie opérative, ces ordres sont transmis aux pré-actionneurs.
- De recevoir les informations transmises par les capteurs relatives la situation de la partie opérative.
- De recevoir les consignes de fonctionnement en provenance du pupitre.
- D'assurer le traitement des informations suivant une logique donnée (programme) afin d'élaborer des ordres.

1.12 La solution proposée à la problématique

Dans l'industrie il y'a un Equipment qui s'appelle (scale weight) : c'est un convoyeur et en même temps un peseur du poids ; alors si les bouteilles sont différent en poids à la consigne, plus lourd ou plus légers, le vérin (ou bien Pusher) comme on utilise dans la simulation FACTORY IO, va pousser la bouteille pour la rediriger vers la ligne de recyclage.

1.13 Conclusion

Dans ce chapitre nous avons présenté des généralités sur l'entreprise, Nous avons présenté l'historique, son organigramme, le plastique et leurs machines, et les systèmes automatisé et leur structure et ces objectifs industriels.

Le chapitre suivant présentera les automates programmables industriels, le logiciel TIA PORTAL V13 et le programme ' LADDER ' et la simulation de la solution proposée dans FACTORY IO.

Chapitre 2 :

**Les automates programmables industriel,
Programmation TIA PORTAL V13 et simulation
FACTORY IO**

2.1 Introduction :

Dans ce chapitre, nous parlerons de la description des automates de manière plus générale et de manière plus détaillée de l'automate S7-300, ainsi que du logiciel de programmation TIA Portal V13, qui est le logiciel de base pour la configuration. Ensuite, un aperçu de Factory IO sera présenté.

2.2 Généralités sur l'automate programmable

L'automate programmable industriel (API) est l'un des appareils de commande les plus utilisés dans le domaine industriel. Les API sont apparues pour la première fois à la fin des années soixante aux Etats-Unis où ils répondaient aux désirs des industries de l'automobile de développer des chaînes de fabrication automatisées qui pourraient suivre l'évolution des techniques et des modèles fabriqués. Un automate programmable est un appareil électronique programmable par un utilisateur automatique (non informaticien) et destiné à piloter en environnement logiques séquentiels ou combinatoires. On les appelle aussi "Programmable Logic Controller P.L.C." ou 'Programmable Controller PC'. Les systèmes automatisés sont destinés à être utilisés dans un environnement industriel, qui utilise une mémoire programmable pour le stockage interne des instructions orientés utilisateurs aux fins de mise en œuvre des fonctions spécifiques, telles que des fonctions de logique, de mise en séquence, de temporisation, de comptage et de calcul arithmétique, pour commander au moyen d'entrée et de sortie TOR ou analogique divers type de machines ou processus. L'automate programmable et ses périphériques associés sont conçus pour pouvoir facilement s'intégrer un système d'automatisme industriel, et être facilement utilisés dans toutes leurs fonctions prévues. [4]

Il exécute une suite d'instructions introduites dans sa mémoire sous forme de programme et s'apparente par conséquent aux machines de traitement de l'information. Trois caractéristiques fondamentales le distinguent totalement des outils d'informatiques tels que les ordinateurs utilisés dans les entreprises :

- Connexion directe aux différents capteurs et actionneurs grâce à ces entrées/sortie.
- Fonctionnement dans des conditions industrielles sévères (température, vibrations, humidité, microcoupure de l'alimentation en énergie électrique...).
- Son aspect pratique grâce à la possibilité de sa programmation en utilisant un langage spécialement développé pour les automates (Step7 TiaPortal)

2.3 Principe de fonctionnement d'un automate programmable

Tous les automates fonctionnent selon le même mode opératoire :

- **Traitement interne** : l'automate effectue des opérations de contrôles et met à jour certains paramètres systèmes (détection des passages en RUN/STOP).
- **Lecture des entrées** : l'automate lit les entrées (de façon synchrone) et les recopies dans la mémoire image des entrées
- **Exécution du programme** : l'automate exécute le programme instruction par instruction et écrit les sorties dans la mémoire image des sorties.
- **Ecriture des sorties** : l'automate bascule les différentes sorties (de façon synchrone) aux positions définies dans la mémoire image des sorties. Ces quatre opérations sont effectuées continuellement par l'automate (fonctionnement cyclique).[5]

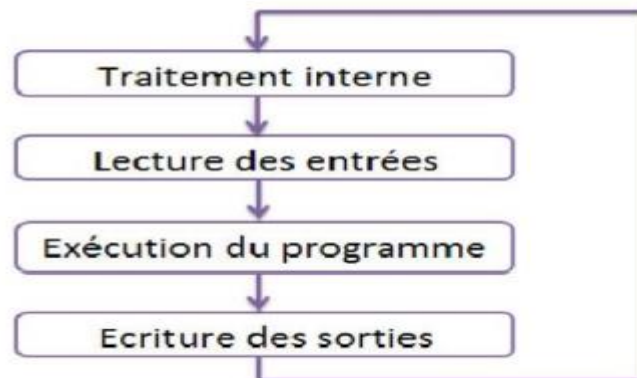


Figure 2.1 Fonctionnement d'un API [6]

2.4 Aspects externes

Il existe deux manières d'intégrer les E/S (Entrées/Sorties) dans l'automate : compacte et modulaire.

2.4.1 Compacte:

Il intègre le processeur, l'alimentation et les entrées/sortie. Il peut réaliser certaines fonctions supplémentaires et recevoir des extensions limitées, De plus, pour certains modèles, si une pièce de l'unité tombe en panne, l'unité entière doit être remplacée. Le principal avantage de ce type est le moindre coût, Il est généralement destiné à la commande de petits automatismes.



Figure 2.2 API compact

2.4.2 Modulaire :

Dans ce modèle, le processeur, l'alimentation et les interfaces entrées/sorties résident dans des unités séparées (modules), les modules se branchent sur un Châssis (rack). Ces automates sont intégrés dans les automatismes complexes de grande puissance et de capacité de traitement.



Figure 2.3 API modulaire

2.5 Aspect interne

Un automate programmable industriel est donc constitué de :

2.5.1 Module d'alimentation :

Le module d'alimentation transforme l'énergie externe provenant du réseau en la mettant en forme afin de fournir aux différents modules de l'API les niveaux de tension nécessaires à leur bon fonctionnement. Plusieurs niveaux de tension peuvent être utilisés par les circuits internes (3V, 5V, 12V, 24V...). Il sera dimensionné en fonction des consommations des différentes parties.

2.5.2 Unité de traitement ou processeur :

Le processeur gère l'ensemble des échanges informationnels en assurant :

- La lecture des informations d'entrée.
- L'exécution des instructions du programme mis en mémoire.
- La commande ou l'écriture des sorties.

2.5.3 Mémoire programme :

La mémoire programme de type RAM contient les instructions à exécuter par le processeur afin de déterminer les ordres à envoyer aux pré-actionneurs reliés à l'interface de sortie en fonction des informations recueillies par les capteurs reliés à l'interface d'entrée.

2.5.4 Mémoire de données :

La mémoire de donnée permet le stockage de :

- L'image des entrées reliées à l'interface d'entrée.
- L'état des sorties élaborées par le processeur.
- Les valeurs internes utilisées par le programme (résultats de calculs, états intermédiaires...).
- Les états forcés ou non des entrées/sorties.

2.5.5 Interface d'entrée :

L'interface d'entrée permet la connexion à l'API d'un multiple de capteurs pouvant être de type :

- TOR (logiques ou Tout Ou Rien).
- Numériques.
- Analogiques.

Ces différentes entrées sont mises en forme par l'interface d'entrée avant d'être stockées dans la mémoire de données.

2.5.6 Interface de sortie :

L'interface de sortie permet la connexion de l'API aux pré-actionneurs, qui peuvent être de type :

- TOR (logiques ou Tout Ou Rien).
- Numériques.

- Analogiques. [7]

2.6 Critère de choix d'un API :

Le choix d'un automate programmable et sa mise en œuvre dépendent d'un cahier des charges. Ils se distinguent par leurs puissances. Cette puissance exprime la capacité d'un automate à gérer des procédés plus ou moins complexes. Les principaux critères sont :

- La rapidité d'exécution.
- Le nombre d'entrée sorties qu'il est capable de gérer.
- La capacité mémoire.
- Le langage du dialogue (la console détermine le langage de programmation).
- Le type de communication avec d'autre système. [8]

2.7 Langage de programmation des API

Le terme langage de programmation API fait référence à la méthode par laquelle l'utilisateur communique des informations à l'API. La norme CEI 61131 a été établie pour normaliser les multiples langages associés à la programmation des automates en définissant les cinq langages standard suivants :[9]

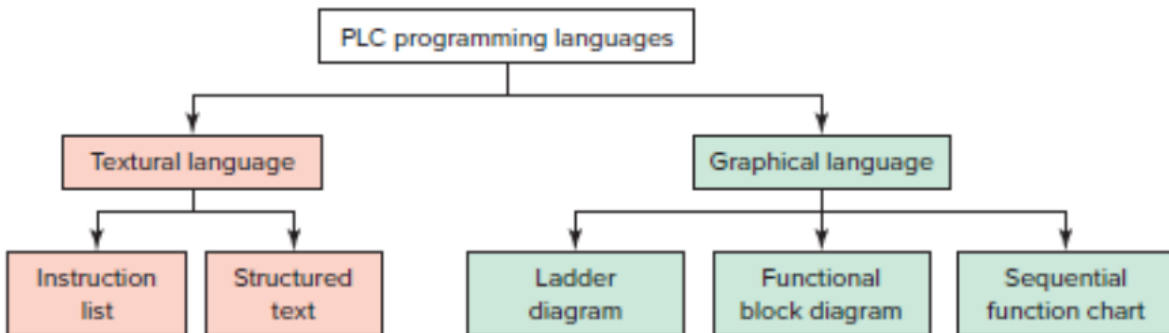


Figure 2.4 Les langages du norme CEI 61131 associés à la programmation des automates.

- **Langage CONT (contacte ou Ladder)**

Dans le langage de programmation graphique CONT, la représentation est fondée sur des schémas à relais. Les éléments d'un tel schéma, comme par exemple les contacts à ouverture ou les contacts à fermeture sont reliés pour former des réseaux. Un ou plusieurs de ces réseaux forment la section d'instructions complète d'un bloc de code. [9]

- **Langage GRAPH (Grafcet ou SFC)**

C'est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite connue d'étapes (états stables), reliées entre elles par des transitions, une condition booléenne est attachée à chaque transition. [9]

- **Langage FBD (Functional Block Diagram)**

Le langage de programmation FBD utilise les pavés logiques bien connus dans l'algèbre booléenne pour la représentation logique. Il permet en outre de représenter des fonctions complexes, telles que les fonctions mathématiques en les mettant directement en liaison avec ces pavés logiques. [9]

- **Langage LIST (Liste d'instruction)**

Le langage de programmation LIST (liste d'instructions) est un langage textuel proche du langage machine. Chaque instruction correspond à une étape de l'exécution du programme par le CPU. Vous pouvez regrouper plusieurs instructions en réseaux. [9]

- **Langage SCL (ST)**

Le langage de programmation SCL (Structured Control Language) optionnel est un langage évolué textuel, Grâce à ses instructions en langage évolué et contrairement au langage LIST, ce langage proche du PASCAL ou C, simplifie entre autres la programmation de boucles et de branches conditionnelles. SCL est de ce fait tout particulièrement adapté au calcul de formules, aux algorithmes d'optimisation complexes ou à la gestion de grandes quantités de données. [9]

C'est possible d'utiliser plusieurs langages dans un seul programme, Mais nous n'utiliserons qu'un seul langage qui est LADDER dans notre projet.

2.8 Présentation de SIMATIC S7-300

La famille S7-300 est constituée d'automates programmables de conception modulaire utilisés pour des automatismes de gamme moyenne des produits SIEMENS, On utilise une console de programmation (PG) pour programmer le S7-300. Reliez la PG à le CPU à l'aide d'un câble PG. et peuvent être connectés entre eux au moyen d'un câble-bus PROFIBUS. Un automate S7-300 consiste en un CPU, un module d'alimentation PS, un module de comptage FM, un module de signaux SM et un processeur de communication.

2.8.1 Composants d'un S7-300

Un S7-300 se compose de plusieurs constituants. La figure suivante vous présente un montage possible :

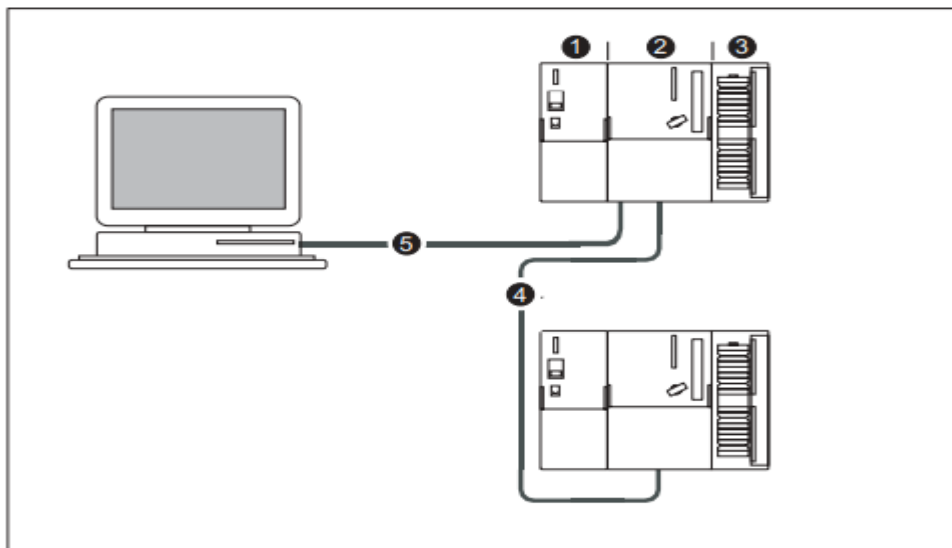


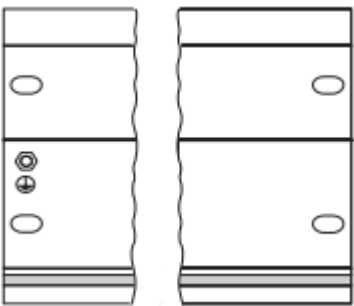

Figure 2.5 Exemple de montage : composants d'un S7-300 [10]

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

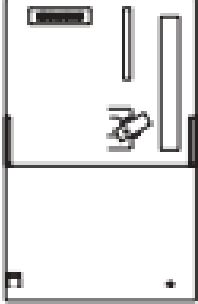

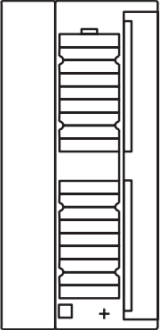
La figure vous montre sous le numéro	Les composants suivants d'un S7-300
(1)	Alimentation
(2)	Module unité centrale
(3)	Module de signaux
(4)	Câble-bus PROFIBUS
(5)	Câble de raccordement d'une console de programmation

Tableau 2.1 Les composants de montage d'un S7-300 [10]

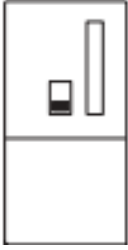
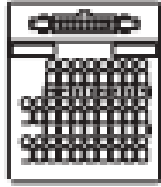
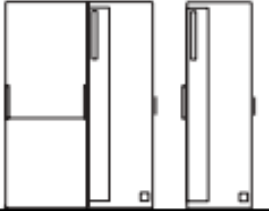
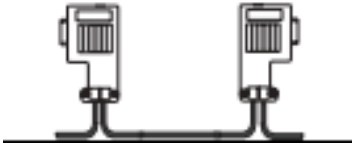

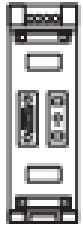
Le tableau suivant présente les principaux composants ainsi que leur fonction.

Composants	Fonction	Figure
Profilé-support (châssis) Accessoires : Etrier de connexion des blindages	C'est le châssis pour un S7-300	
Alimentation (PS) (Power Supply)	Elle convertit la tension réseau (AC 120/230 V) en tension de service DC 24 V et assure l'alimentation du S7-300 ainsi que l'alimentation externe pour les circuits de charge DC 24 V.	

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

<p>CPU Accessoires : connecteur frontal (pour CPU à périphérie intégrée)</p>	<p>Elle exécute le programme utilisateur alimente le bus de fond de panier du S7-300 en 5 V communique avec les autres partenaires d'un réseau MPI via l'interface MPI. En outre, un CPU peut être maître ou esclave DP sur un sous-réseau PROFIBUS.</p>	
<p>Modules de signaux (SM) (Signal Module) (Modules d'entrées TOR, modules de sorties TOR, modules d'entrées/sorties TOR, modules d'entrées analogiques, modules de sorties analogiques, modules d'entrées/sorties analogiques) Accessoires : Connecteur frontal</p>	<p>Il adapte les différents niveaux de signaux des signaux de processus au S7-300.</p>	
<p>Modules de fonction (FM) (Fonction Module) Accessoires : Connecteur frontal</p>	<p>Ils réalisent les tâches du traitement des signaux de processus critiques au niveau du temps et exigeant beaucoup de mémoire. Par exemple le positionnement ou le réglage</p>	

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

<p>Processeur de communication (CP) Accessoires : Cable de raccordement</p>	<p>Il soulage le CPU des taches de communication, par exemple CP 342-5 DP pour liaison au PROFIBUS-DP</p>	
<p>SIMATIC TOP connecte Accessoires : Module frontal enfichable avec raccordement à câble plat</p>	<p>Ils servent au câblage des modules TOR</p>	
<p>Coupleur (IM) (Interface Module) Accessoires : Cable de raccordement</p>	<p>Il relie les différentes rangées d'un S7-300 entre elles</p>	
<p>Câble-bus PROFIBUS avec connecteur de bus</p>	<p>Ils relient les partenaires d'un sous-réseau MPI ou PROFIBUS entre eux</p>	
<p>Câble PG</p>	<p>Il relie un PG/PC avec un CPU</p>	
<p>Répéteur RS 485</p>	<p>Ils servent à renforcer les signaux dans un sous-réseau MPI ou PROFIBUS ainsi qu'à coupler les segments d'un sous-réseau MPI ou PROFIBUS</p>	


Console de programmation (PG) ou PC avec logiciel STEP 7	Vous avez besoin d'une PG pour configurer, paramétrer, programmer et tester le S7-300	
----------------------------------------------------------	---------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------

Tableau 2.2 Composants d'un S7-300 [10]

2.8.2 Configuration de sous-réseaux [10]

Conformément aux différentes exigences des niveaux d'automatisation (niveaux de conduite, de cellules, de terrain et de capteurs/actionneurs), SIMATIC offre les sous-réseaux suivants :

- Interface multipoint (MPI)
- PROFIBUS
- Couplage point à point
- Industriel Ethernet
- Interface actionneur/capteur (ASI)

2.9 TIA Portal

2.9.1 Vue du projet et vue du Portail

Le TIA Portal a deux vues importantes. Au démarrage, TIA Portal affiche par défaut la vue du Portail. Cette vue facilite la mise en route, surtout pour les utilisateurs débutants.

La vue du Portail fournit une vue tâche-orientée des outils pour travailler sur le projet. Ici, vous pouvez rapidement décider ce que vous voulez faire et ouvrir l'outil pour la tâche à accomplir. Si nécessaire, une modification de la vue du projet s'effectue automatiquement pour la tâche sélectionnée.

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

La Figure 2.6 montre la vue du portail. En bas à gauche, il y a une option pour changer entre cette vue et la vue du projet.

La vue du projet, illustrée à la Figure 2.6, est utilisée pour la configuration matérielle, programmation, création de la visualisation et de plusieurs autres tâches. Par défaut, la vue projet affiche la barre de menus avec les barres d'outils en haut, tous les composants d'un projet sur la gauche et c'est appelé "cartes de tâche", les instructions et bibliothèques, à droite.[11]

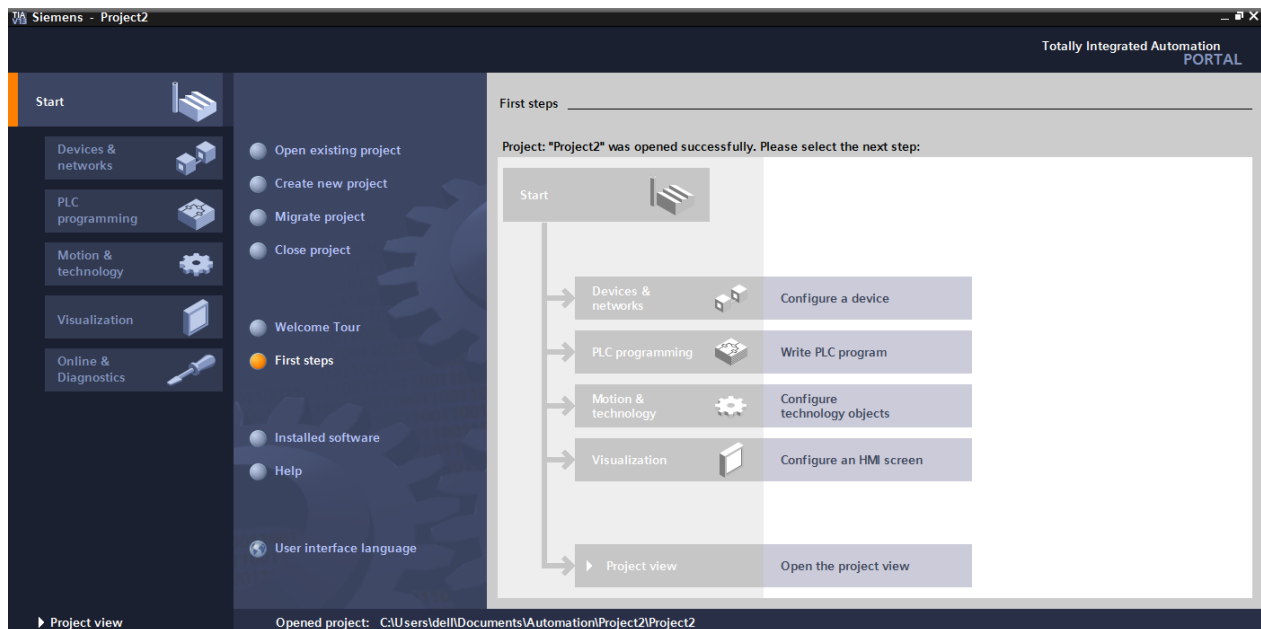


Figure 2.6 Vue du Portail

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

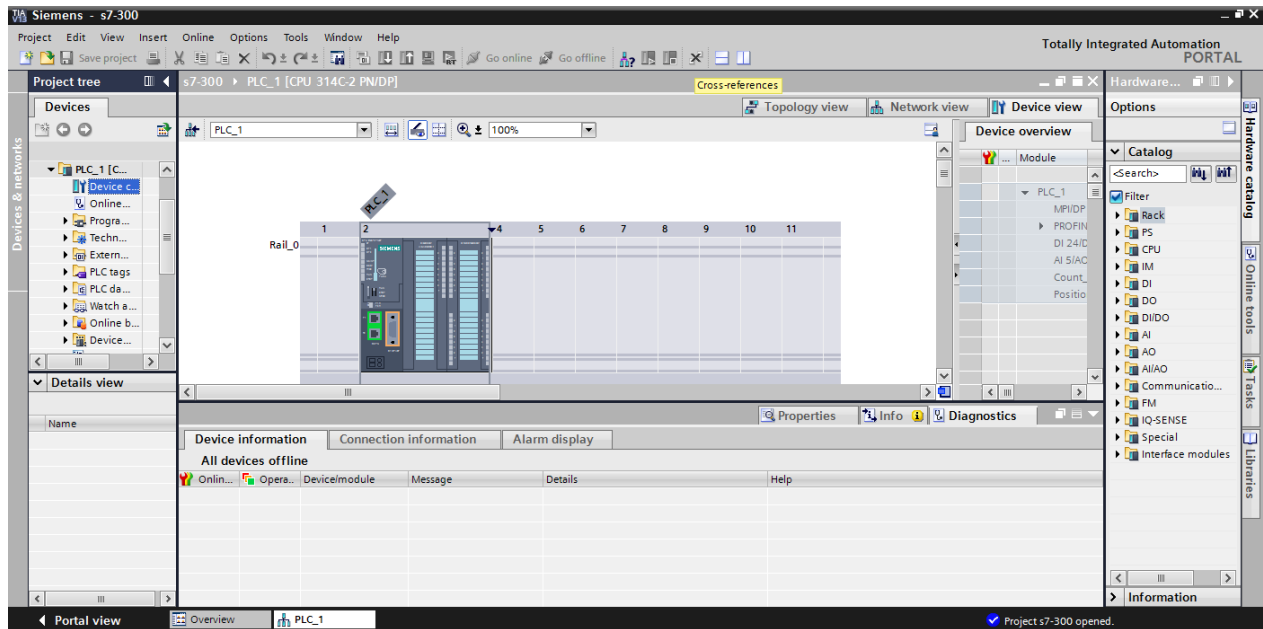


Figure 2.7 Vue du Projet

2.9.2 Instructions structurées étape par étape

Créer un nouveau projet

Sélectionnez le Totally Integrated Automation Portal (Figure 2.8) qui s'ouvre ici par un double-clic.



Figure 2.8 Icon TIA Portal V13

Dans la vue du portail sous le menu "Démarrer" (Figure 2.9), sélectionnez la commande "Créer un projet".

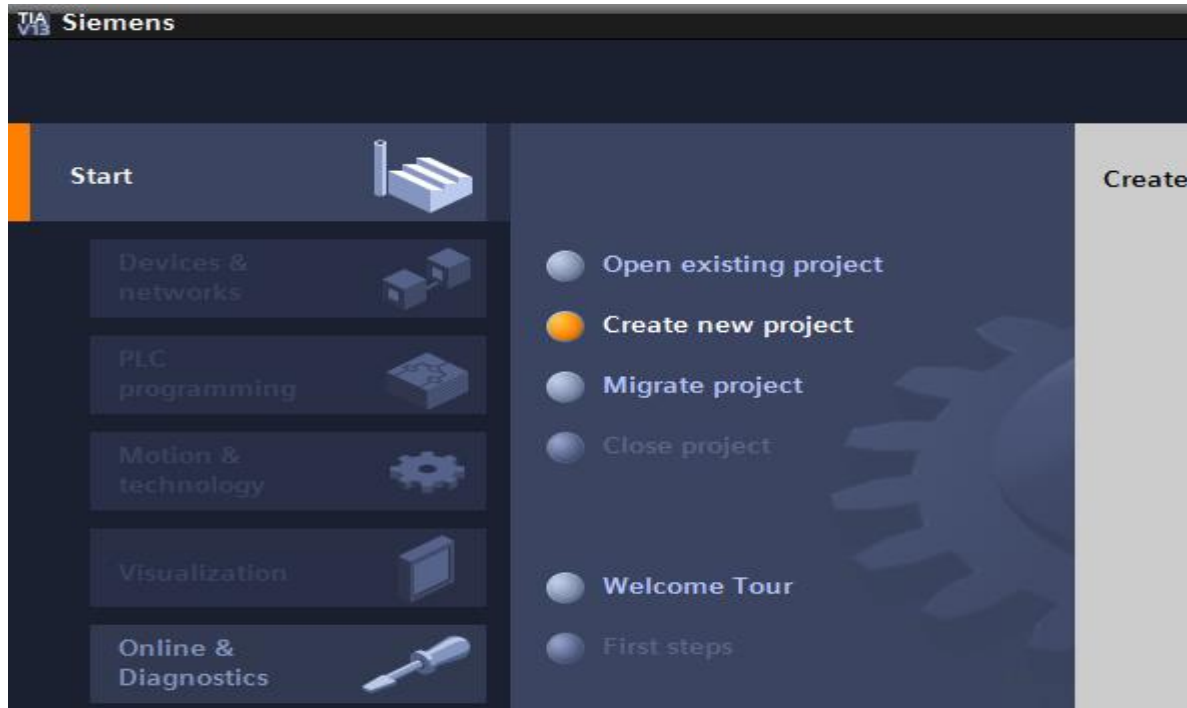


Figure 2.9 création un nouveau projet au menu

Modifiez le nom du projet (Figure 2.10), le chemin, l'auteur et le commentaire selon le cas et cliquez sur → "Créer".

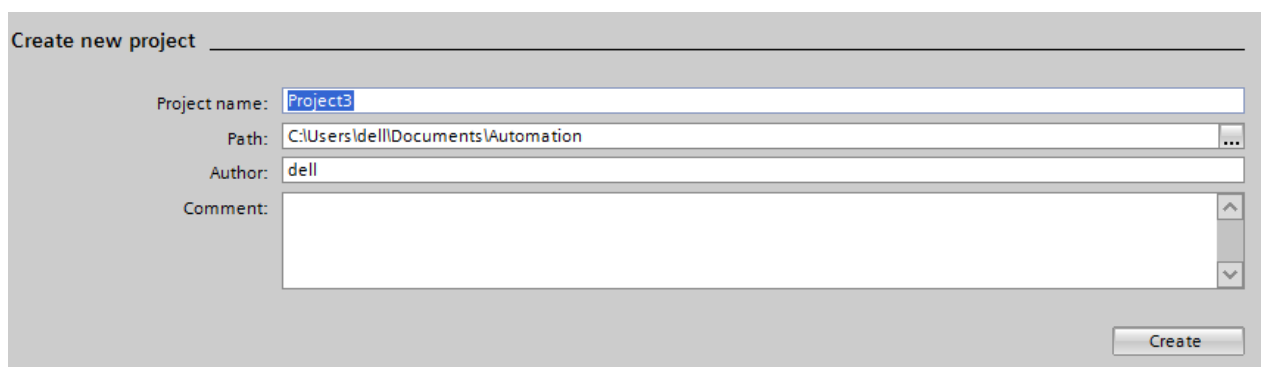


Figure 2.10 Modification le nom du projet

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

Le projet sera créé et ouvert et le menu "Démarrer", " mise en route " s'ouvrira automatiquement.

2.9.3 Le matériel de SIMATIC S7-300

Dans le "Démarrer" en vue Portal sélectionnez → " mise en route " → "Appareils et réseaux" → "Configurer un appareil" → Le menu "Afficher tous les appareils" s'ouvre dans le portail "Appareils & Réseaux" → Passer au menu "Ajouter un appareil" → Créez un nouveau CPU Utiliser un modèle de CPU S7-300 → cocher "Ouvrir la vue des appareils" et cliquer sur ajouter. Le TIA portal passe maintenant automatiquement à la vue du projet comme la figure 2.7

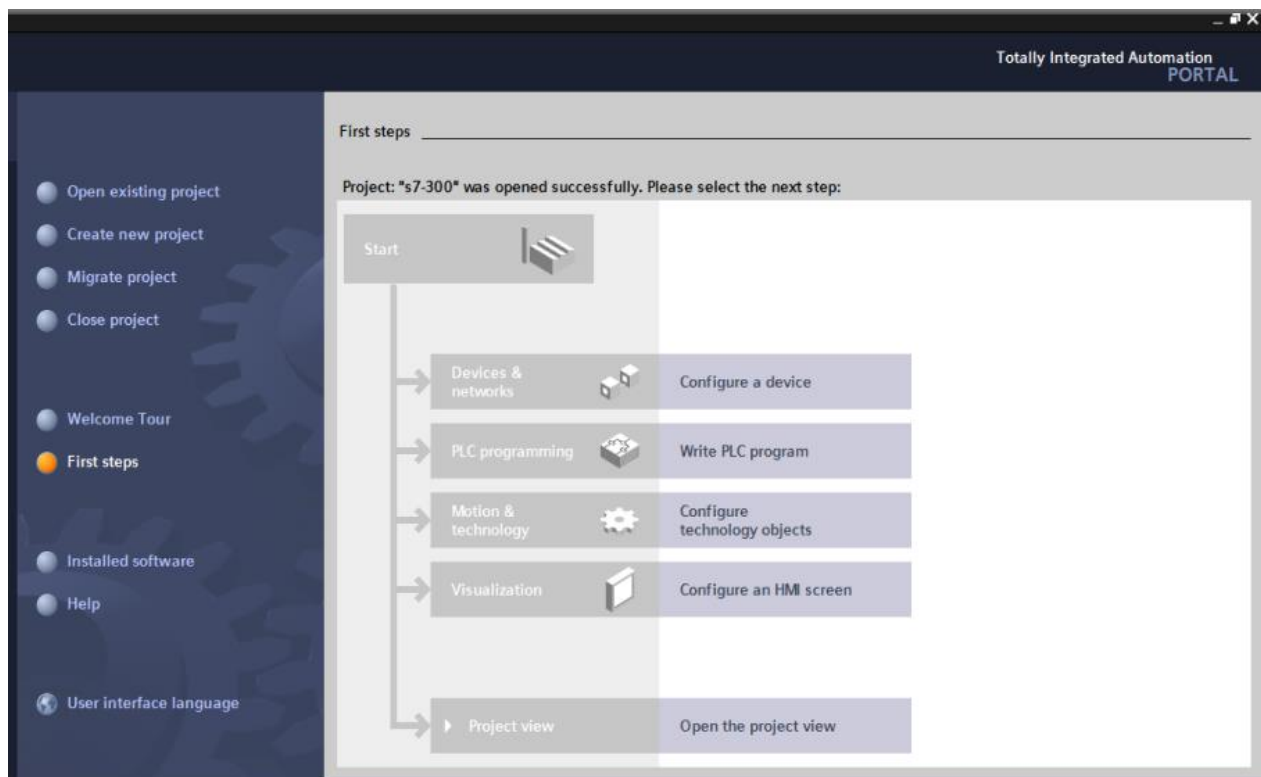


Figure 2.11 Appareils et réseaux

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

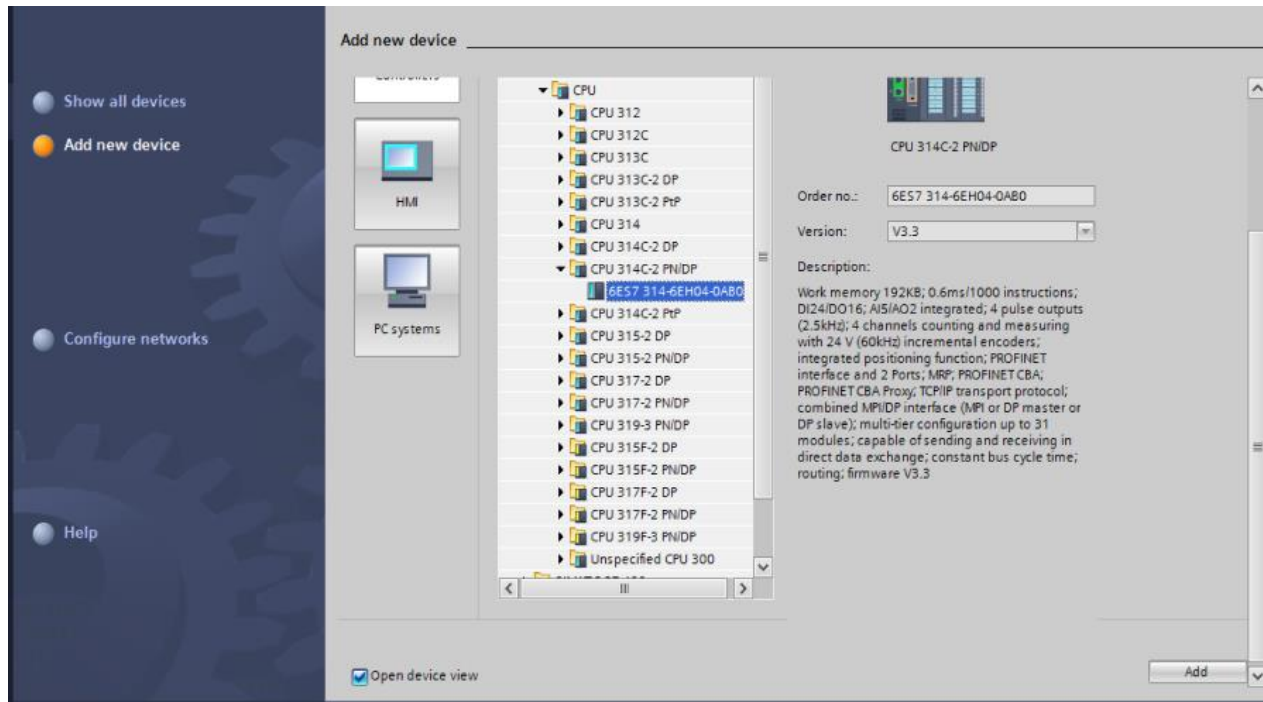


Figure 2.12 choisir un processeur

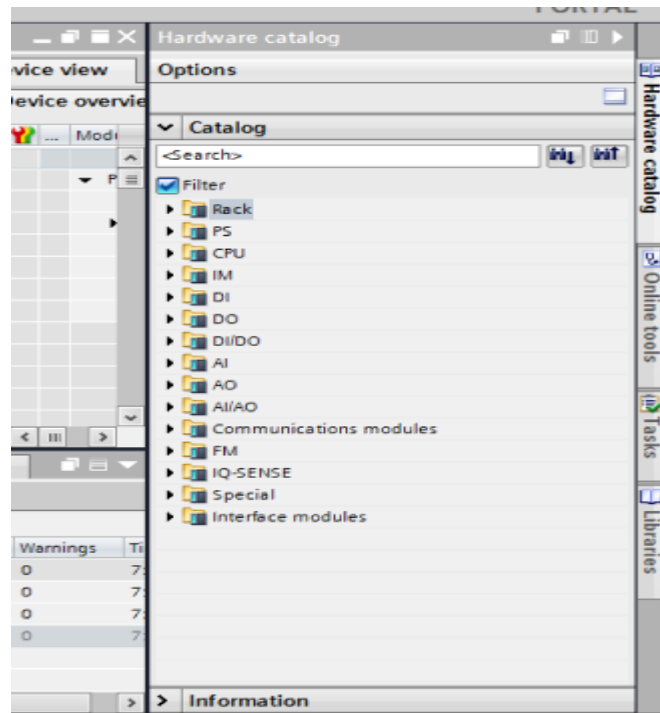


Figure 2.13 les instructions et bibliothèques et pour ajoute matériel

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

Le Programme sera fait dans bloc de programme et les variables actionneur capteur dans 'PLC TAG'

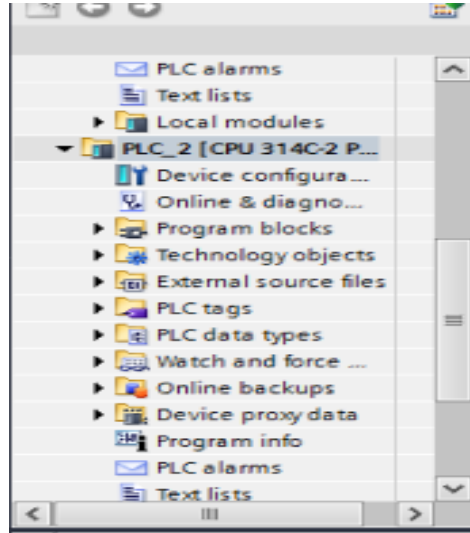
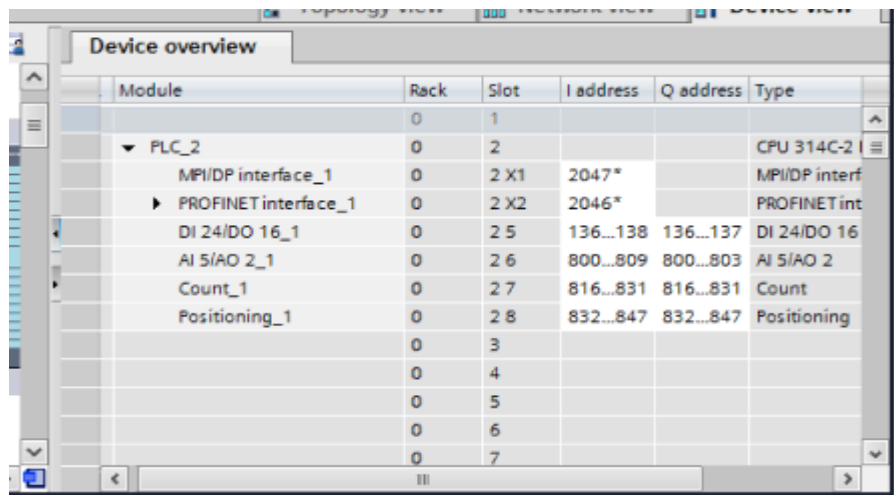


Figure 2.14 les composants d'un projet

L'étape suivante consiste à vérifier les plages d'adressage (Figure 2.15) des entrées et des sorties et à les adapter si nécessaire. DI/DO doit avoir une plage d'adressage de 136...138 et 136...137 et AI/AO doit avoir une zone d'adressage de 800...809 et 800...803, respectivement.



Module	Rack	Slot	I address	Q address	Type
PLC_2	0	2			CPU 314C-2
MPI/DP interface_1	0	2 X1	2047*		MPI/DP interf
PROFINET interface_1	0	2 X2	2046*		PROFINETint
DI 24/DO 16_1	0	2 5	136...138	136...137	DI 24/DO 16
AI 5/AO 2_1	0	2 6	800...809	800...803	AI 5/AO 2
Count_1	0	2 7	816...831	816...831	Count
Positioning_1	0	2 8	832...847	832...847	Positioning
	0	3			
	0	4			
	0	5			
	0	6			
	0	7			

Figure 2.15 Zones d'adressage.

2.9.4 Enregistrer et compiler la configuration matérielle

Avant de compiler la configuration, vous devez enregistrer votre projet en cliquant sur le bouton → « Enregistrer le projet » (Figure 2.16). Pour compiler votre CPU avec la configuration de l'appareil, sélectionnez d'abord le dossier → "CPU_300 [CPU314C-2 PN/DP]" et cliquez sur → "Compiler".



Figure 2.16 Barre de menu

2.9.5 Télécharger la configuration matérielle sur l'appareil

pour télécharger l'intégralité de votre CPU, sélectionnez le dossier → "CPU_300 [CPU314C-2 PN/DP]" et cliquez sur l'icône "Télécharger" → le gestionnaire de configuration des propriétés de connexion (téléchargement étendu) s'ouvre → Tout d'abord, l'interface doit être correctement sélectionné. Cela se passe en trois étapes.

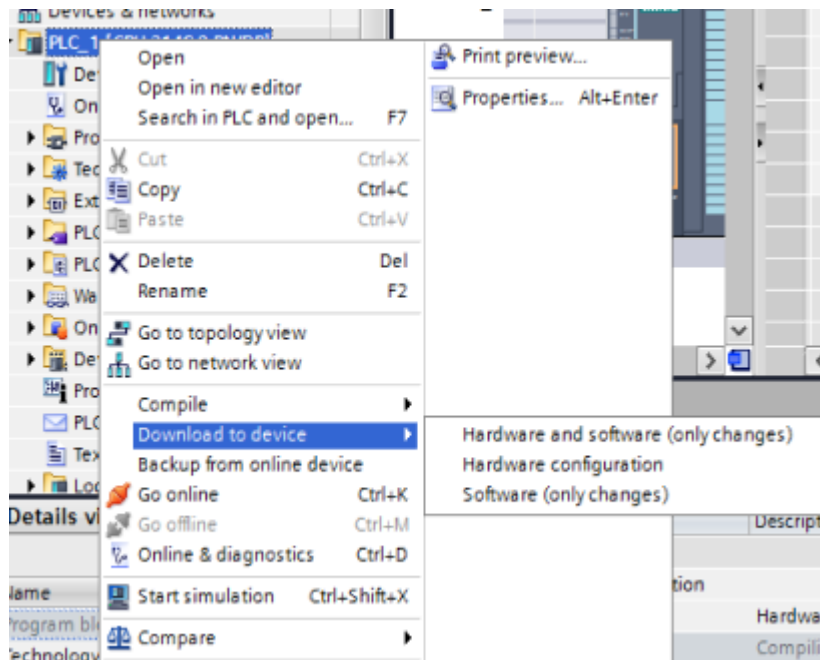


Figure 2.17 navigateur du projet

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

Type de l'interface PG/PC →PN/IE → Interface PG/PC ici : PLCSIM et cliquer sur commencer la recherche et charge

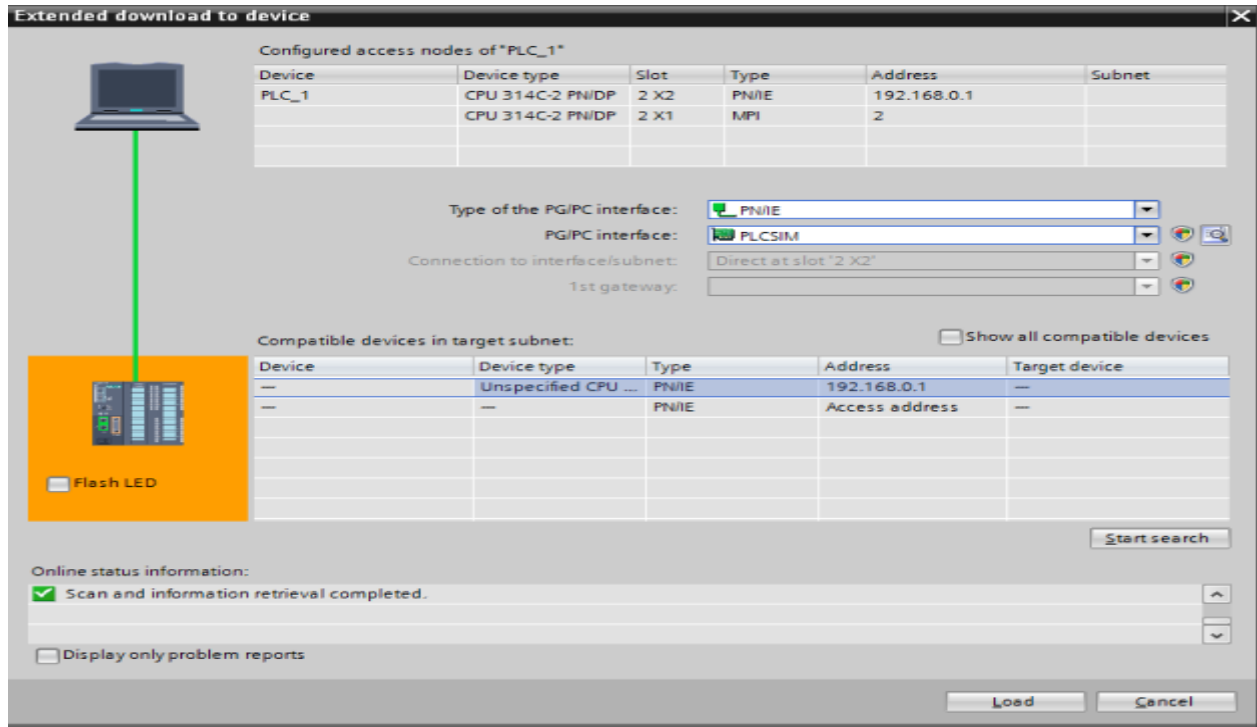


Figure 2.18 Téléchargement étendu.

Après un téléchargement réussi, la vue du projet s'ouvrira à nouveau automatiquement. Un rapport de chargement apparaît dans le champ d'information sous "Général". Cela peut être utile lors du dépannage d'un téléchargement infructueux. [11]

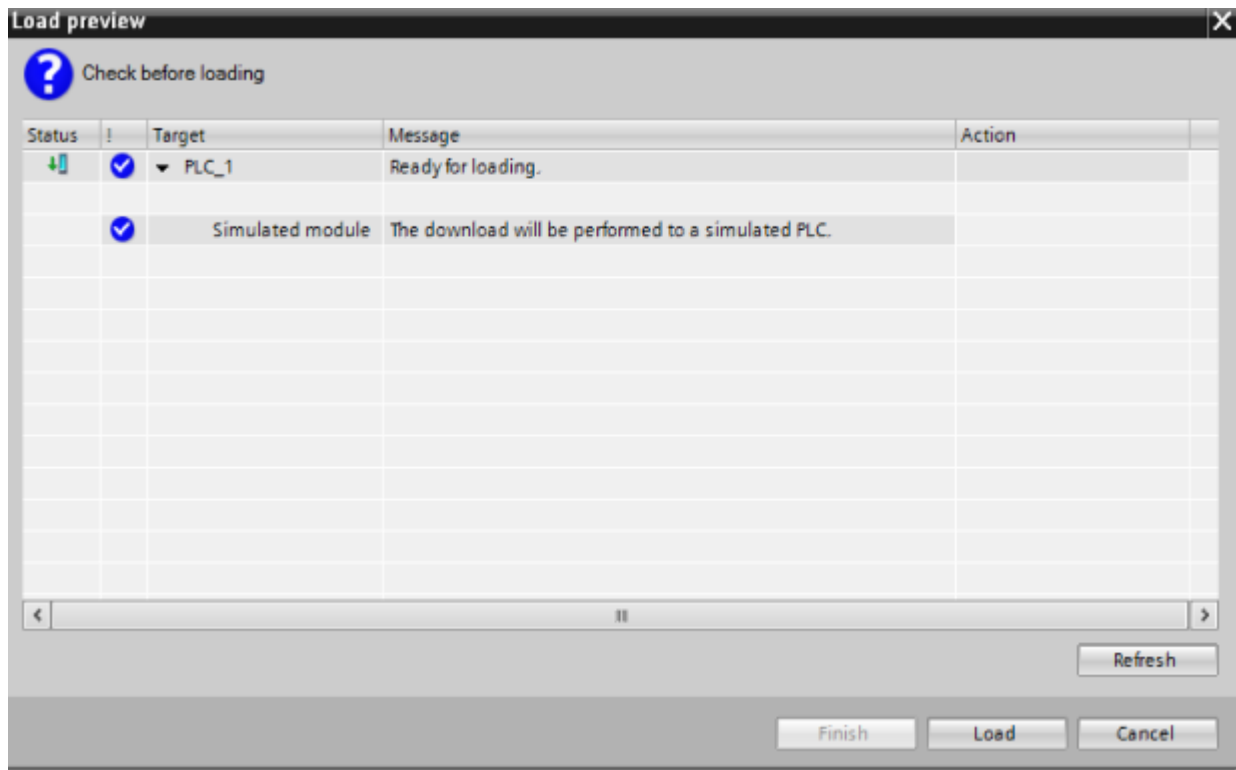


Figure 2.19 Charger l'aperçu.

2.9.6 Simulateur S7-PLCSIM

Dans S7-PLCSIM (Figure 2.20), vous pouvez exécuter votre programme utilisateur STEP 7 et l'essayer dans un automate programmable simulé. Cette simulation s'exécute sur votre PC ou console de programmation, une Field PG par exemple. La simulation étant réalisée entièrement dans le logiciel STEP 7, vous n'avez pas besoin de matériel S7 (CPU ou modules de signaux). Avec S7-PLCSIM, vous pouvez simuler des programmes utilisateur STEP 7 qui ont été développés pour les automates S7-300, S7-400.

S7-PLCSIM offre une interface simple au programme utilisateur STEP 7 servant à visualiser et à modifier différents objets tels que les variables d'entrée et de sortie. Pendant que votre programme est traité par le CPU simulé, vous pouvez recourir au logiciel STEP 7. Par exemple, vous pouvez visualiser et forcer des variables avec la table des variables (VAT). S7-PLCSIM offre

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

une interface utilisateur graphique permettant de visualiser et de modifier les variables des programmes d'automatisation, d'exécuter en mode cyclique ou automatique le programme du système cible simulé ou de modifier l'état de fonctionnement de l'automate simulé.

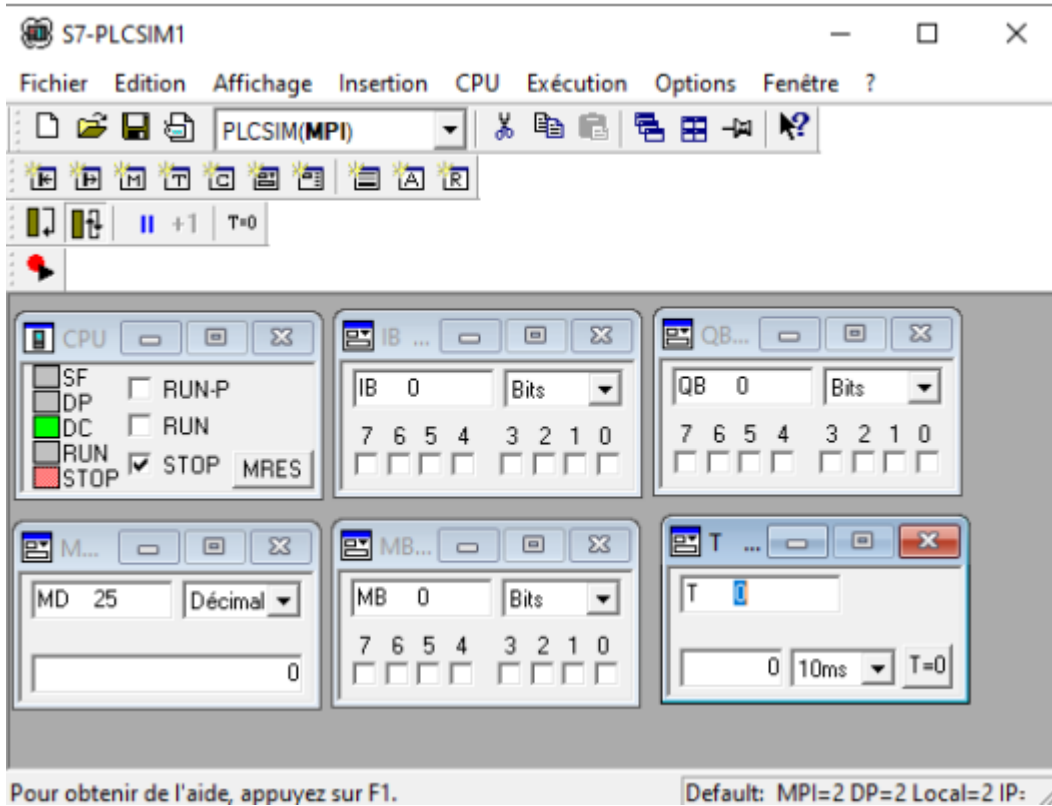


Figure 2.20 Simulateur S7-PLCSIM

2.10 Factory I/O

2.10.1 Introduction

Factory I/O est un logiciel de simulation 3D pour la conception et la simulation Système industriel complet. Développé dans le même esprit que les jeux vidéo actuels, Factory I/O offre un haut niveau de réalisme lors des phases de conception et de simulation. Le logiciel peut interagir avec un véritable automate, permettant le contrôle à l'extérieur.

Le logiciel de simulation Factory IO (Figure 2.21) est disponible en 7 versions ou éditions :

L'édition de démarrage, l'édition Allen Bradley pour les automates Allen Bradley, le Siemens Edition pour les automates Siemens, l'édition Modbus et OPC pour Modbus et OPC

Communications, l'édition MHJ pour le contrôleur virtuel WinSPS-S7 et le simulateur d'automate Siemens WinPLC-Engine, l'édition Automgen pour la connexion avec le logiciel Automgen, l'édition Ultimate qui est la version complète incluant un SDK pour les développements des drivers de communication.[12]

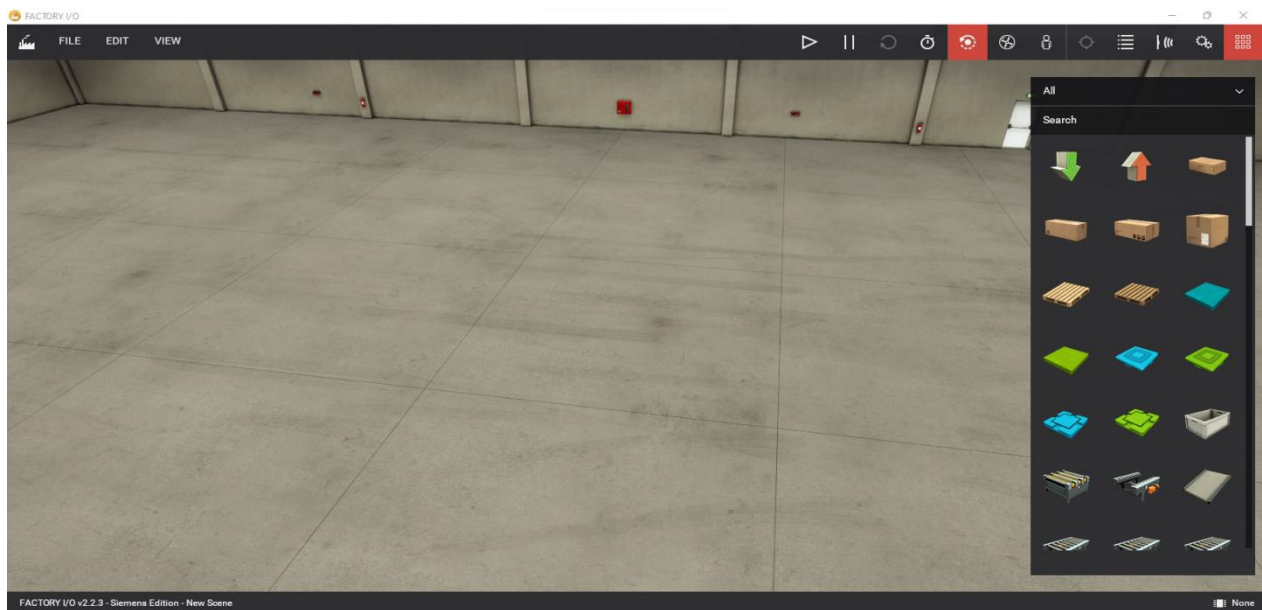


Figure 2.21 Factory IO

2.10.2 L'interface du logiciel

L'interface du logiciel est assez simple et comporte 3 menus (Figure 2.22) et plusieurs boutons pour gérer les caméras, afficher la palette, afficher les capteurs, afficher les actionneurs, effectuer une simulation etc...

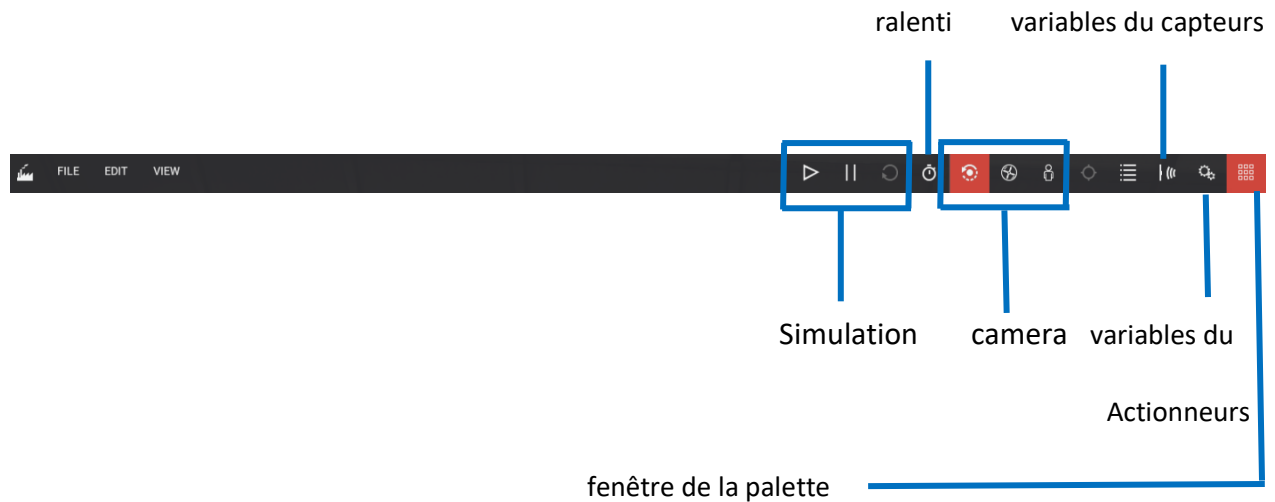


Figure 2.22 Barre de menu

2.10.3 fenêtre de la palette

Au niveau fenêtre de la palette se trouvent les différents composants permettant de concevoir votre système automatisé (Figure 2.23). À la palette, nous pouvons trouver des convoyeur, boutons poussoir, voyants, etc.

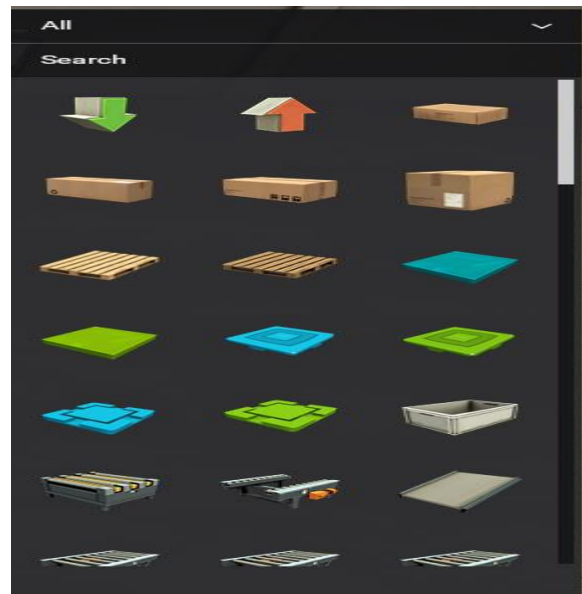


Figure 2.23 fenêtre de la palette

Votre imagination sera limitée, vous pouvez concevoir plusieurs types de systèmes par simple glisser-déposer du composant dans l'interface d'édition. [12]

2.10.4 les pilotes (Drivers)

Les pilotes sont utilisés pour contrôler le système automatisé conçu via l'interface d'édition. Il existe plusieurs types de pilotes : les pilotes pour les automates Siemens, Schneider, Rockwell etc, ainsi que le pilote Connect I/O qui est le logiciel du contrôleur intégré aux Factory IO. Pour activer un pilote donné, cliquez sur le coin inférieur droit du logiciel ou bien appuie sur F4 ou fichier -->pilot.

Vous pouvez choisir parmi la liste des pilotes disponibles

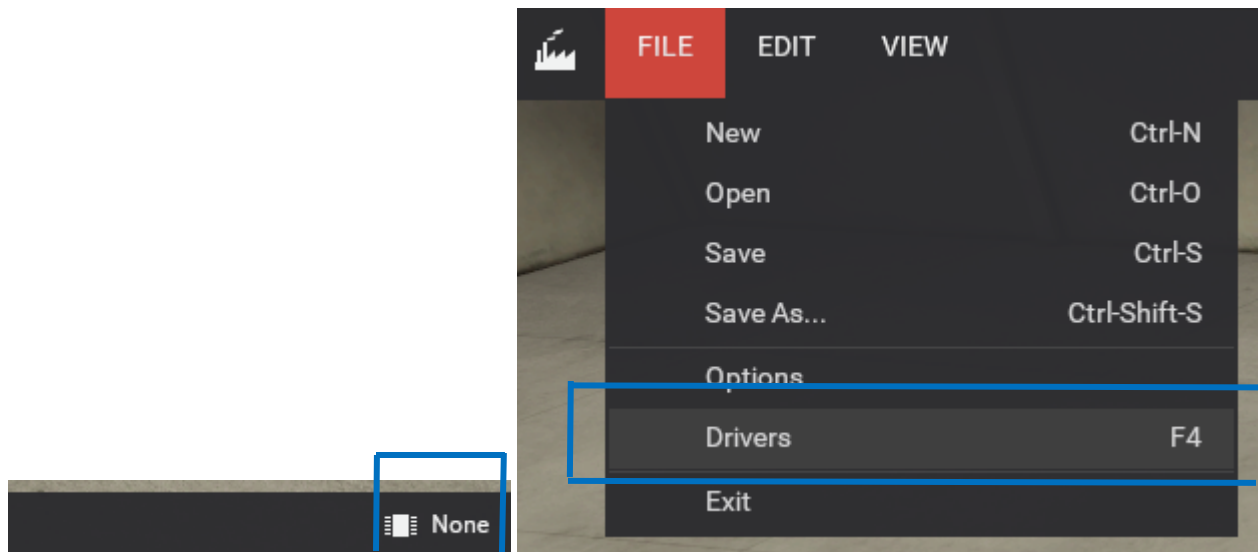


Figure 2.24 Barre des pilotes

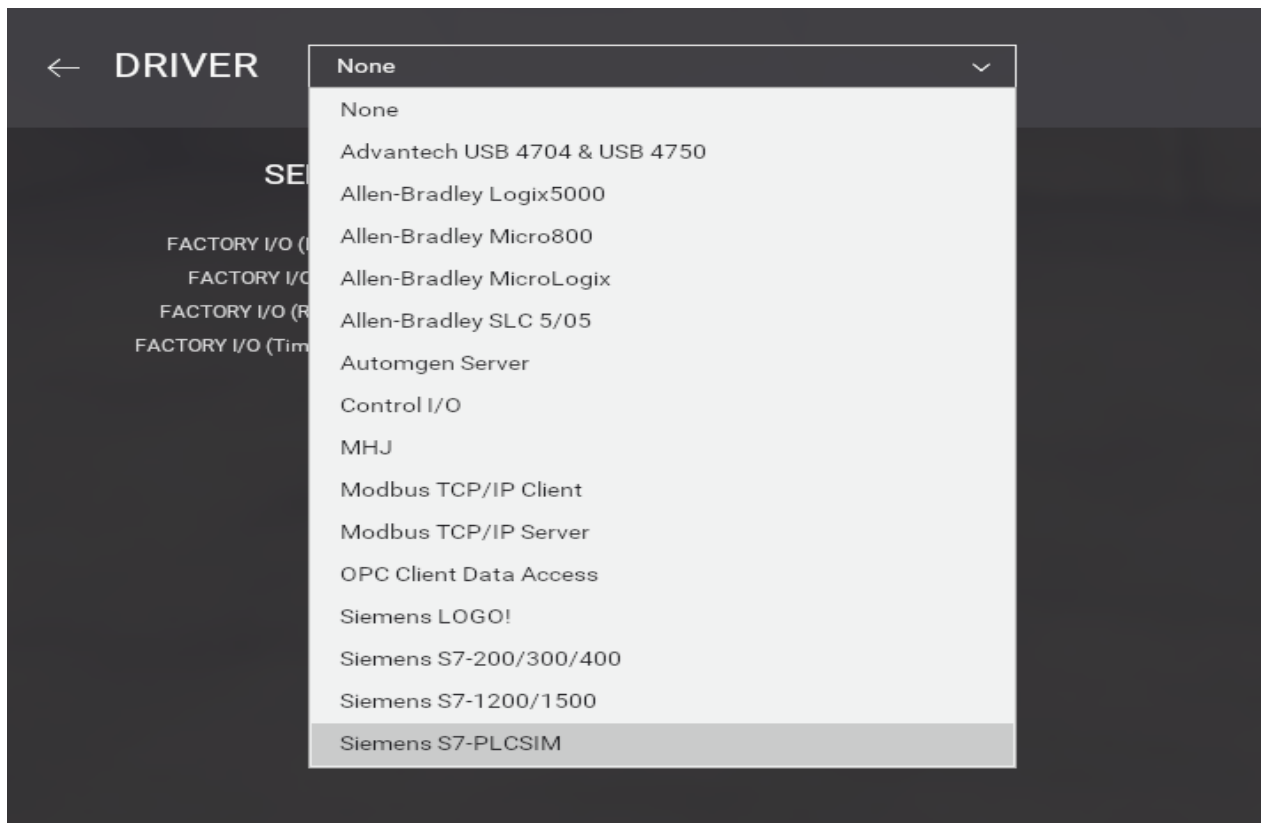


Figure 2.25 Menu du pilote

2.10.5 Configuration

Nous pouvons configurer Factory IO dépend de notre automate comme adressage d'entrées et sorties ou type de données mot (word) ou mot double (Dword) comme indiqué sur cette (Figure 2.26) et (Figure 2.27).

Et on cliquer sur Connect pour la connexion de Factory IO avec API.

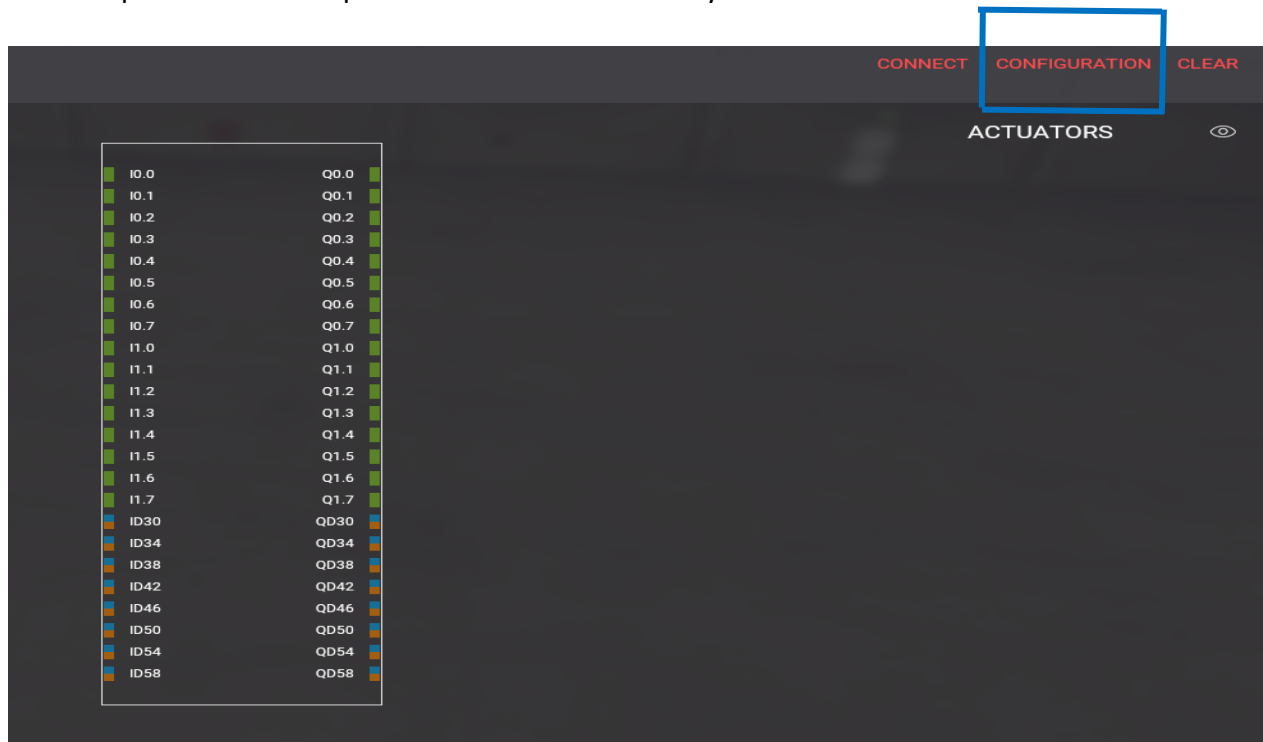


Figure 2.26 barre de configuration

Chapitre 2 :Les automates programmables industriel, Programmation TIA PORTAL V13 et simulation FACTORY IO

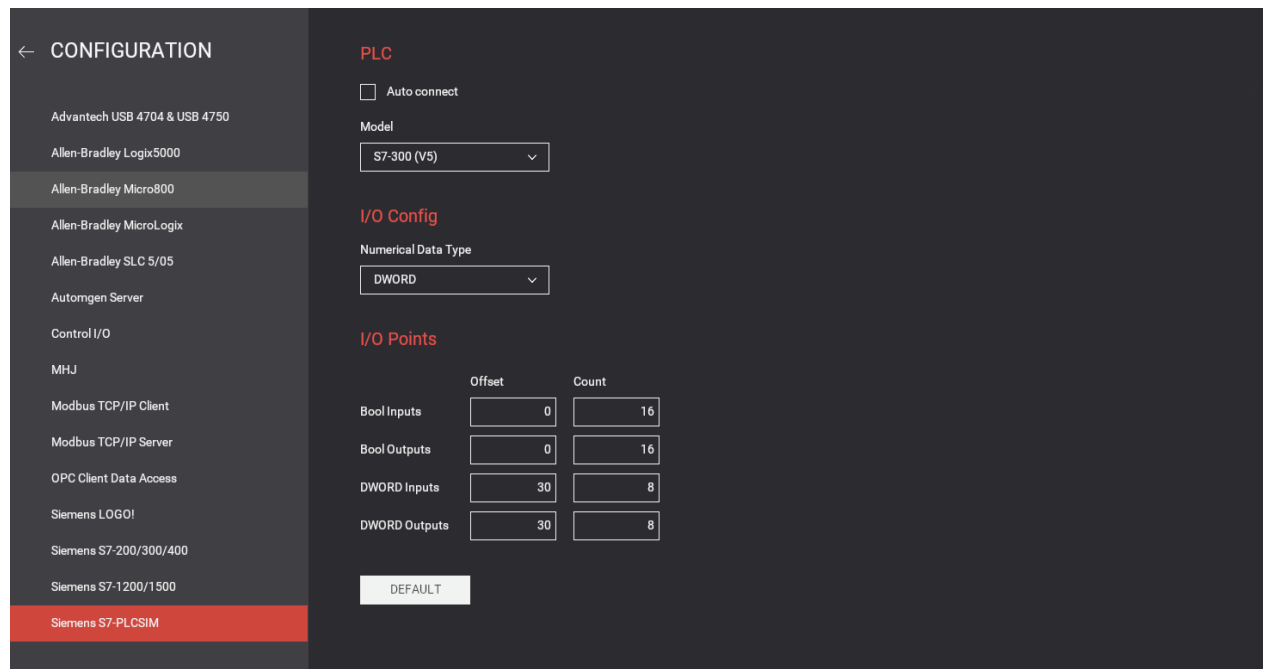


Figure 2.27 la fenêtre de configuration

2.11 Conclusion

Dans ce chapitre, nous avons vu une description de l'automate programmable S7-300, du logiciel de programmation TIA Portal V13 et du logiciel Factory I/O, après quoi nous avons détaillé les étapes de création et de configuration d'un projet. A la fin, une description de step7 et Factory I/O présentée. Dans le prochain chapitre, nous présenterons la description de notre Programme ainsi que les étapes de développement de notre système de contrôle automatisé.

Chapitre 3 : Présentation du Programme en Tia Portal v13 et la simulation en Factory I/O

3.1 Introduction

Dans ce chapitre nous présentons le système à pesage automatique de Factory I/O, Nous commençons par une description générale du système. Nous détaillerons ensuite le programme utilisé pour le pesage automatique avec TIA PORTAL. A la fin du chapitre, des résultats de simulations seront présentés.

3.2 Système à pesage (Description générale du système)

Le triage du poids est composé de quatre convoyeurs, le premier convoyeur est pour diriger les nouvelles boîtes qui sont produise (il n'y a pas des bouteilles dans le logiciel de FACTORY IO donc j'ai utilisé les boites) vers le deuxième convoyeur qui est le peseur (scale weight) c'est un convoyeur et en même temps un peseur. On a quatre catégories de boîtes de poids différents, de 3Kg,8Kg, 10 Kg et 15 Kg. On accepte seulement celles de 8Kg et 10KG.

Les bouteilles en Venus vont du 22 grammes à 29 grammes, les normes vont de 25 g à 27 g ; les autres poids ne sont pas acceptés.

Dans le programme, j'affiche le poids de la boite plus 17; ce qui signifie en gramme c'est une conversation de réalité vers la simulation, ça veut dire :

- 1) 3kg -> 20g
- 2) 8kg -> 25g
- 3) 10kg -> 27g
- 4) 15kg -> 32g

Donc nous acceptons seulement 25g et 27g.

Si les boîtes qui sont dans le peseur ne sont pas dans les normes, il y'a un Pusher qui va pousser les boites vers le troisième convoyeur pour le rediriger vers le recyclage.

Si les boites sont dans les normes ils continueront vers le quatrième convoyeur pour remplir les cases de bouteilles ou des boîtes.

Il y'a aussi quatre afficheurs, l'un pour afficher le poids actuel de la boîte, les trois autres pour compter les boîtes légères, moyennes et lourdes ; la boîte moyenne est détecté par un capteur.

Il ya cinq boutons : bouton d'arrêt bouton marche, et trois boutons pour réinitialiser les compteurs des boîtes.

Le bouton d'arrêt et marche ce sont des bouton lumineux, arrêt rouge et marche vert, et un voyant jaune allumé si les bouteilles ne sont pas dans les normes.

3.3 Panneau de commande

3.3.1 Définition

Les affichages et voyants nécessaires à la surveillance de la machine sont regroupés dans un tableau de commande, généralement associé aux différentes commandes de la machine. Un tableau de commande est également un appareil, généralement fixé au mur, dans une installation, industrielle ou autre, regroupant les différents affichages (compteurs, écrans, voyants...) permettant de contrôler son fonctionnement. Il est généralement situé dans une salle de contrôle, regroupé avec des appareils permettant de réguler ce fonctionnement et de contrôler les opérations. Cette pièce comprend également les moyens de communication avec l'extérieur.

3.3.2 Structure du panneau de commande

Le panneau de commande se compose de plusieurs boutons différents illustrés dans le Tableau 3.1. Figure 3.1 montrent les images de la structure du panneau de commande.

Description	Type
Écran d'affichage	7 segments
Bouton marche	Bouton poussoir
Bouton arrêt	Bouton poussoir
Bouton réinitialiser	Bouton poussoir
Bouton d'urgence	Bouton poussoir d'arrêt d'urgence
Voyant jaune	LED jaune

Tableau 3.1 Panneau de commande



Figure 3.1 Panneau de commande

Triage des boites

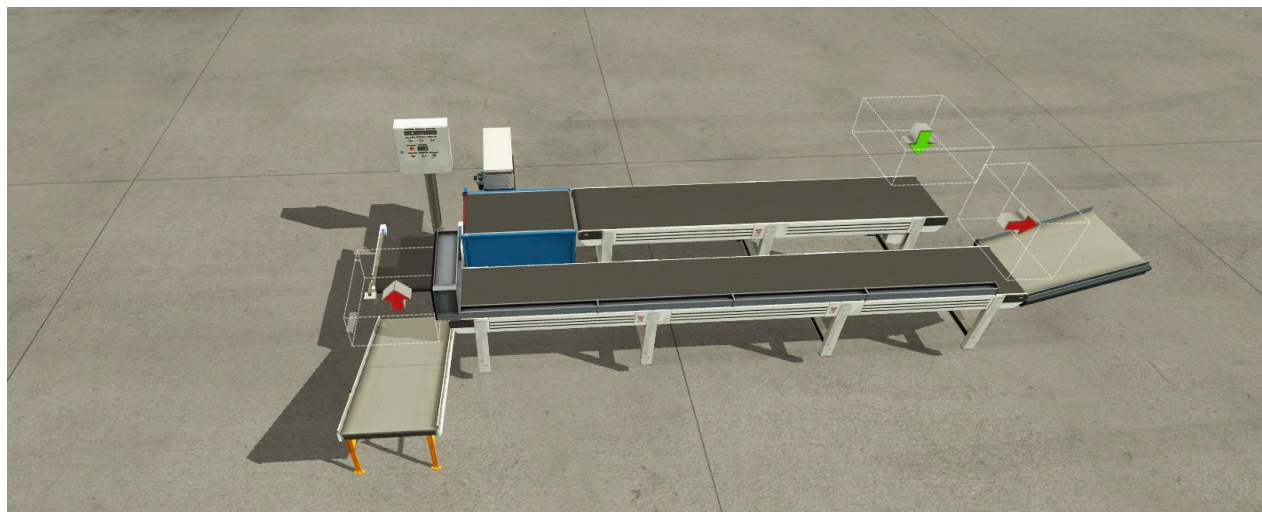


Figure 3.2 Système de pesage

3.4 Les actionneurs et capteurs du système

3.4.1 Les actionneurs (sorties)

- Convoyeur (quatre mètres)
- Convoyeur (six mètres)
- Mini convoyeur
- Convoyeur de pesage
- Voyant vert démarrage
- Voyant rouge arrêt
- Voyant jaune indicateur
- Poussoir

3.4.2 Les capteurs (entries)

- Bouton d'urgence
- Bouton d'arrêt
- Bouton marche
- Réinitialisation léger
- Réinitialisation Moyen
- Réinitialisation Lourd
- Capteur photoélectrique diffus

3.4.3 Sortie analogique

- Afficheur du poids
- Afficheur pour compter les boites lourdes
- Afficheur pour compter les boites moyennes
- Afficheur pour compter les boites légères

3.4.4 Entrée analogique

- Poids de la balance du convoyeur

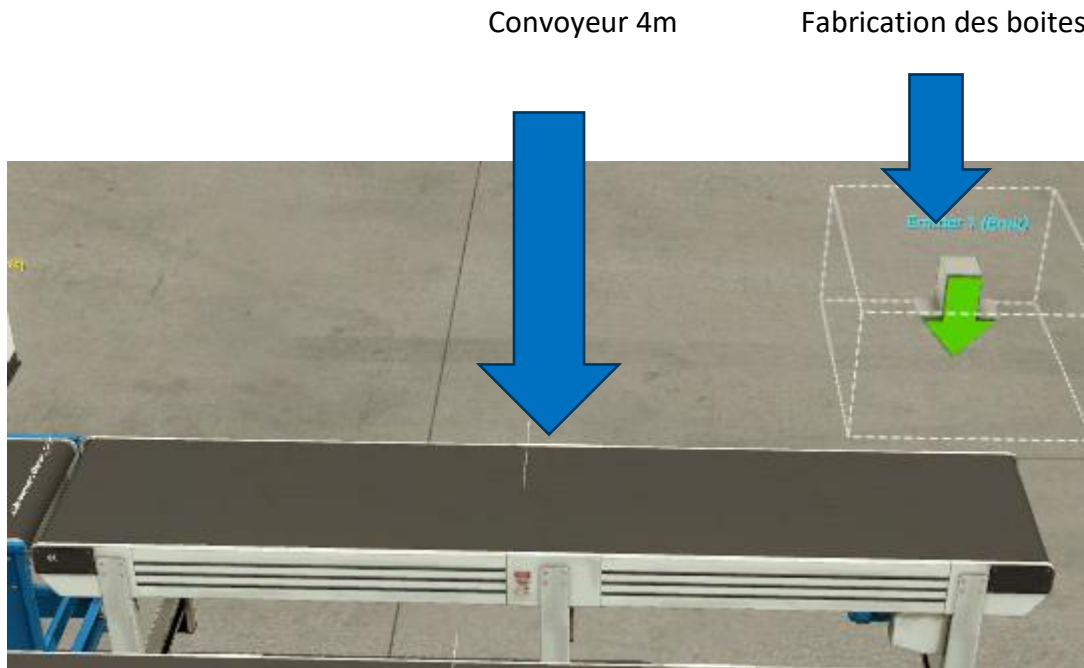


Figure 3.3 la partie de démarrage

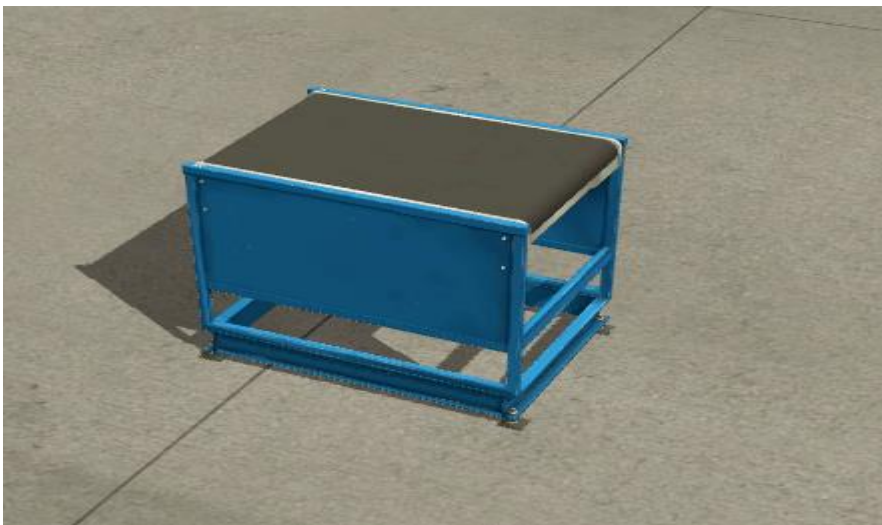


Figure 3.4 Convoyeur peseur

Un Convoyeur peseur peut généralement être facilement intégrée à une chaîne de montage existante pour peser des produits afin d'augmenter la productivité.

pendant que le convoyeur est en mouvement et sont destinés au pesage des bouteilles, cartons ou produits non emballés. Les Convoyeurs peseur sont capables de s'intégrer dans à peu près n'importe quel processus industriel qui doit peser des articles lors de leur passage. [24]

[<https://www.centralcarolinascale.com/conveyor-scales.htm>]

Les Avantages

- Réduisez les délais de traitement en triant
- Augmentez la précision des sorties et évitez les colis sous-remplis et sur-remplis
- La conception modulaire permet des modifications pour s'adapter à une variété de conteneurs et d'applications

Considérations

Lors de l'automatisation des processus de pesage et du choix d'une solution de peseur en ligne, il est important de prendre en compte :

- Type, taille et forme des articles en cours de traitement
- Conception globale du système et processus pour déterminer où placer au mieux les stations de pesage [25]

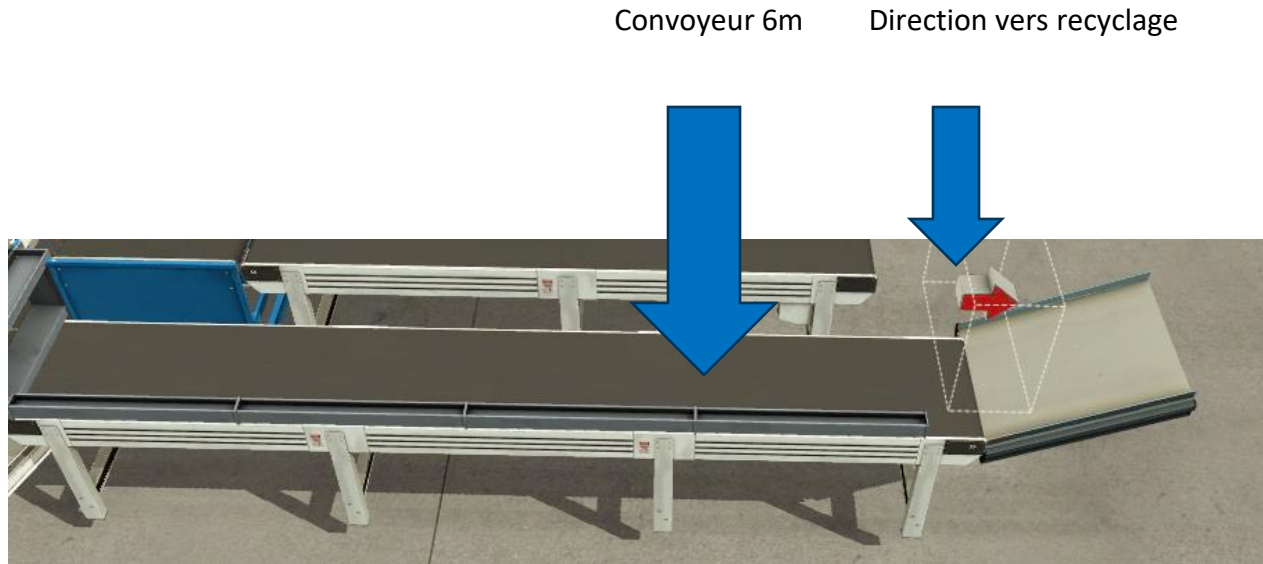


Figure 3.5 la partie de direction vers le recyclage

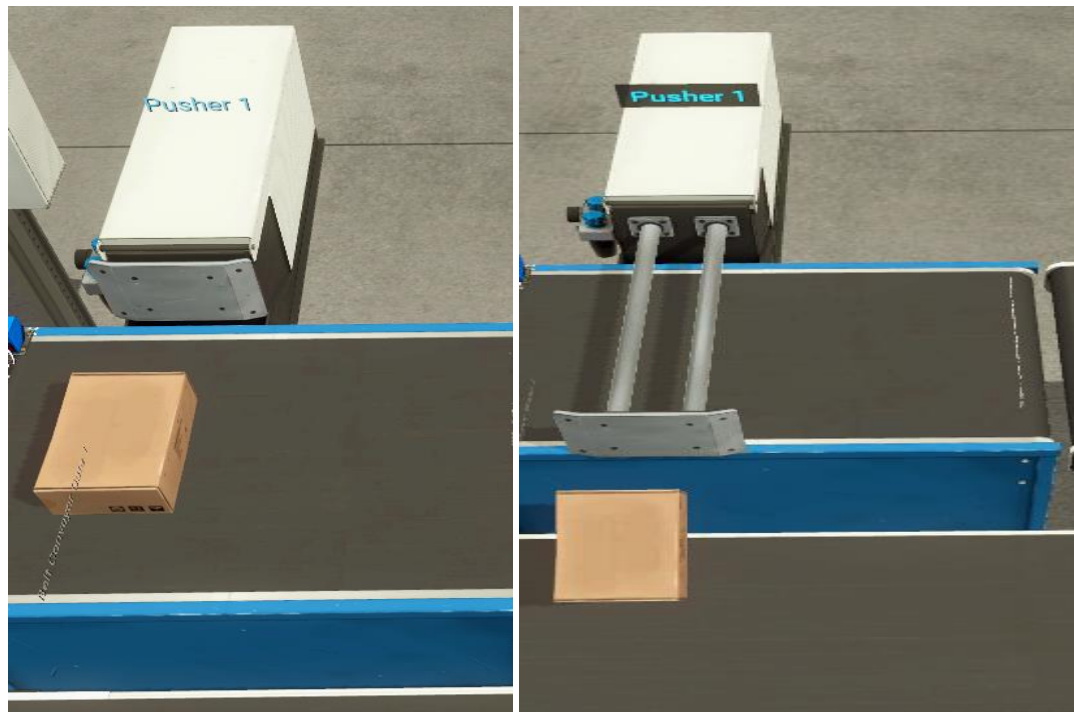


Figure 3.6 Pusher pousse la boîte de 3kg vers recyclage

Trieur à poussée pneumatique avec deux capteurs fin de course indiquant les bordures avant et arrière. Comprend également une servovalve qui peut être utilisée pour ajuster et mesurer la position de la tige. Le contrôle peut se faire avec des valeurs numériques ou analogiques selon la configuration choisie.[26]



Figure 3.7 : Capteur photoélectrique de proximité

Les capteurs photoélectriques diffus ont l'émetteur et le récepteur ensemble dans le même composant.

Pour que le capteur diffus fonctionne, l'émetteur du capteur doit être pointé vers un objet afin que la lumière se déplace de l'émetteur du capteur vers l'objet, puis rebondisse vers le récepteur du capteur.[27]

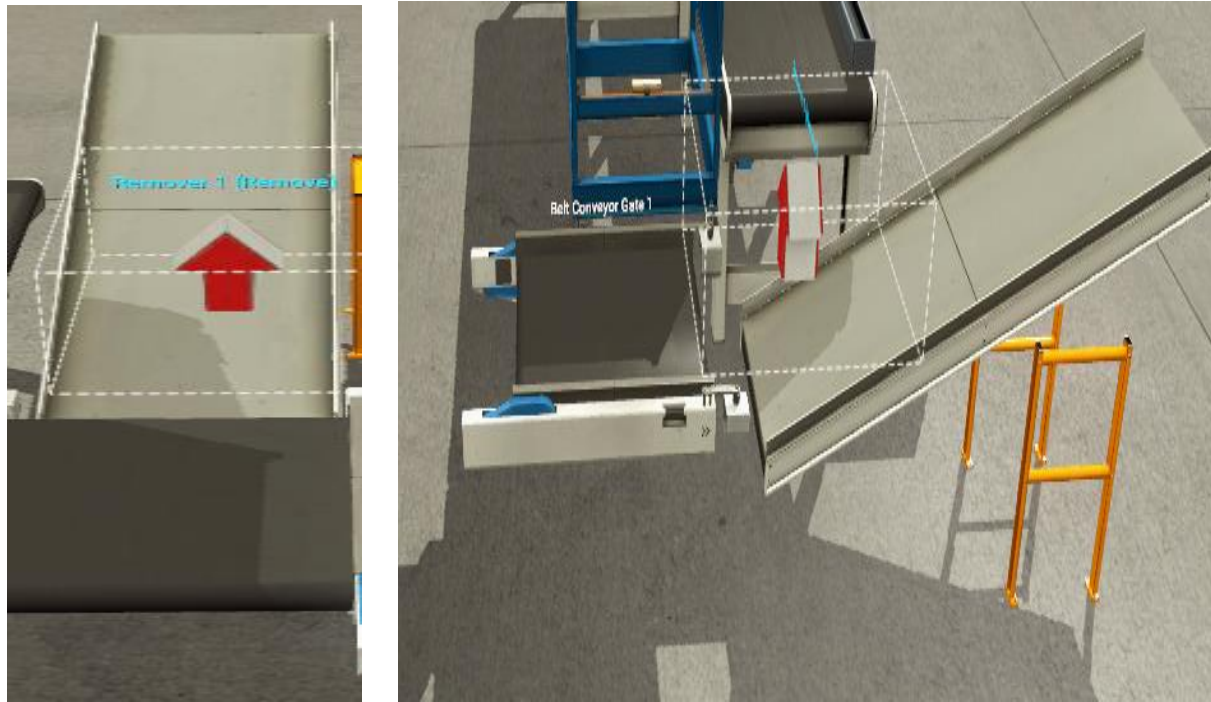


Figure 3.8 la partie de boites au poids moyennes à deux faces



Figure 3.9 panneau de commande complète

On glisse les actionneurs et capteur à l'adresse appropriée

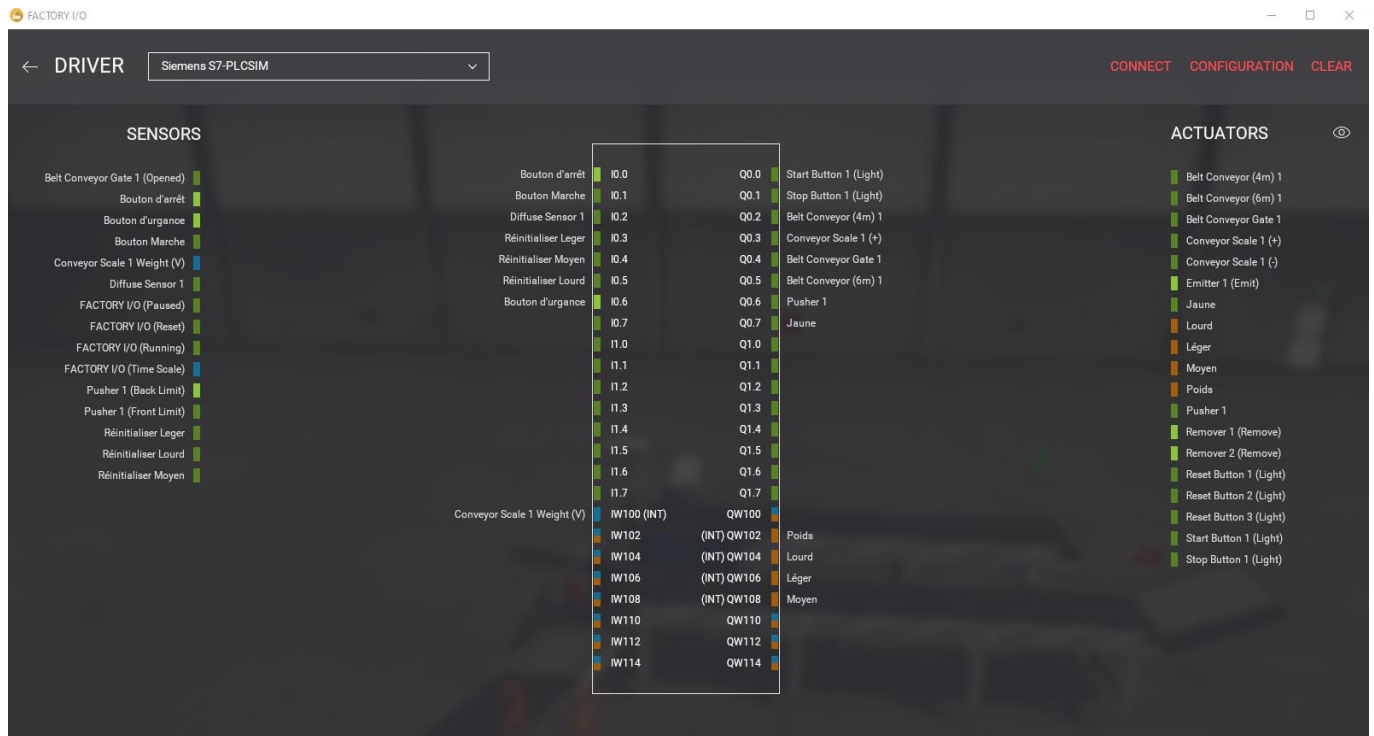


Figure 3.10 Les actionneur et capteurs utilisé dans FACTORY I/O

3.5 Connexion entre FACTORY IO et PLCSIM

Pour connecter le FACTORY IO avec le PLCSIM premièrement on est besoin d'exécute le PLCSIM et après appuie sur connect sur le FACTORY IO.

Et pour l'exécution du programme de TIA PORTAL au FACTORY IO, on compile le programme block et clique sur démarrer la simulation et charge le programme dans dispositif.

Après on clique sur Monitoring pour la visualisation.

Lorsque le monitoring est prêt on clique sur RUN dans PLCSIM.

Maintenant le FACTORY IO est bien connecté avec PLCSIM et le programme de TIA PORTAL été déjà exécuté dans FACTORY IO la dernière étape et clique sur 'changer entre le mode édition et le mode exécution' le triangle comme la Figure 3.15.

Compiler démarrer la simulation

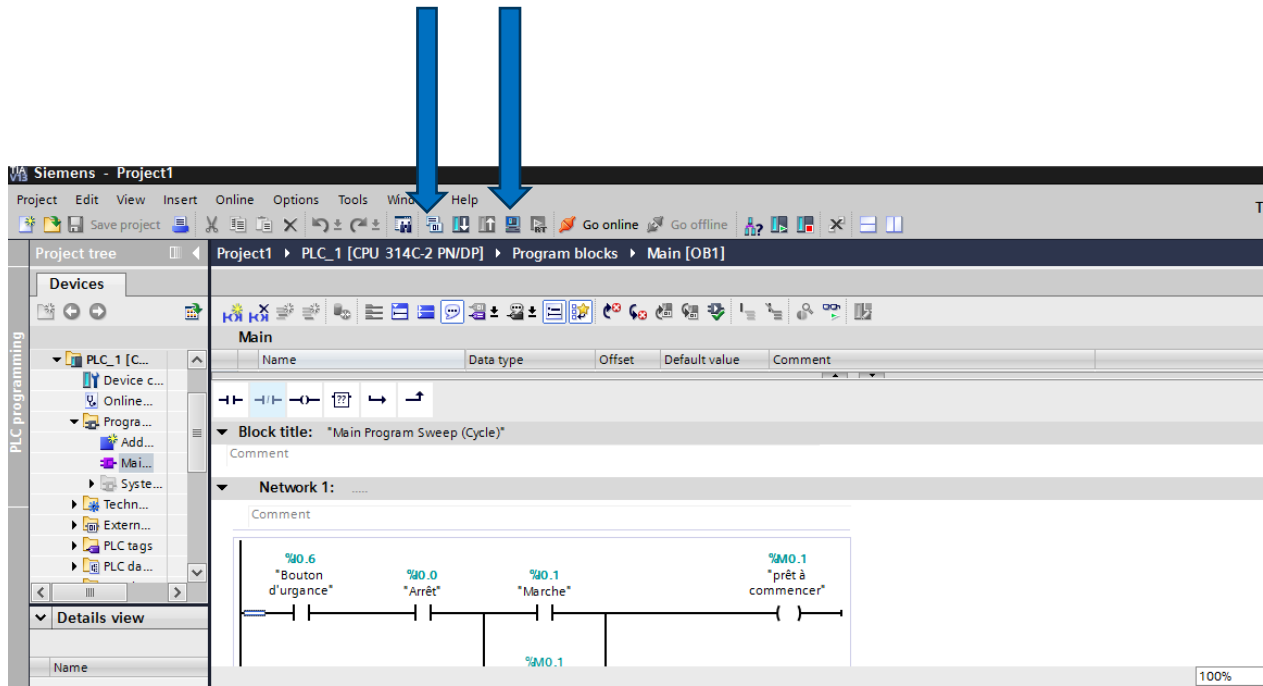


Figure 3.11 simulation du programme au PLCSIM

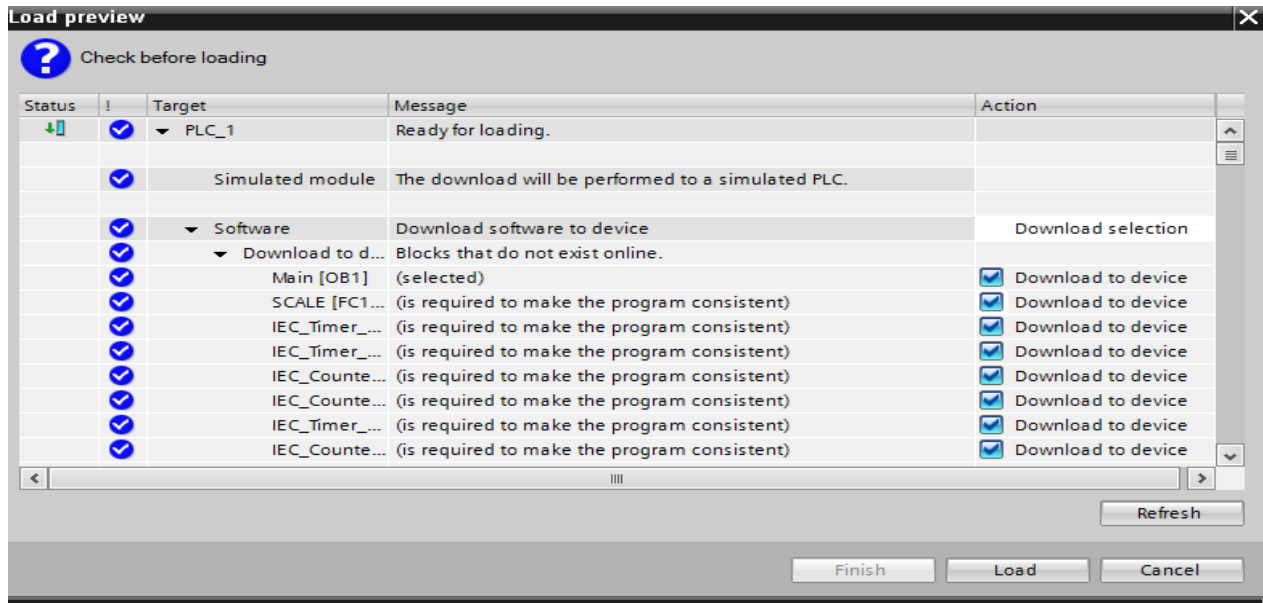


Figure 3.12 chargement du programme dans dispositif

Monitoring

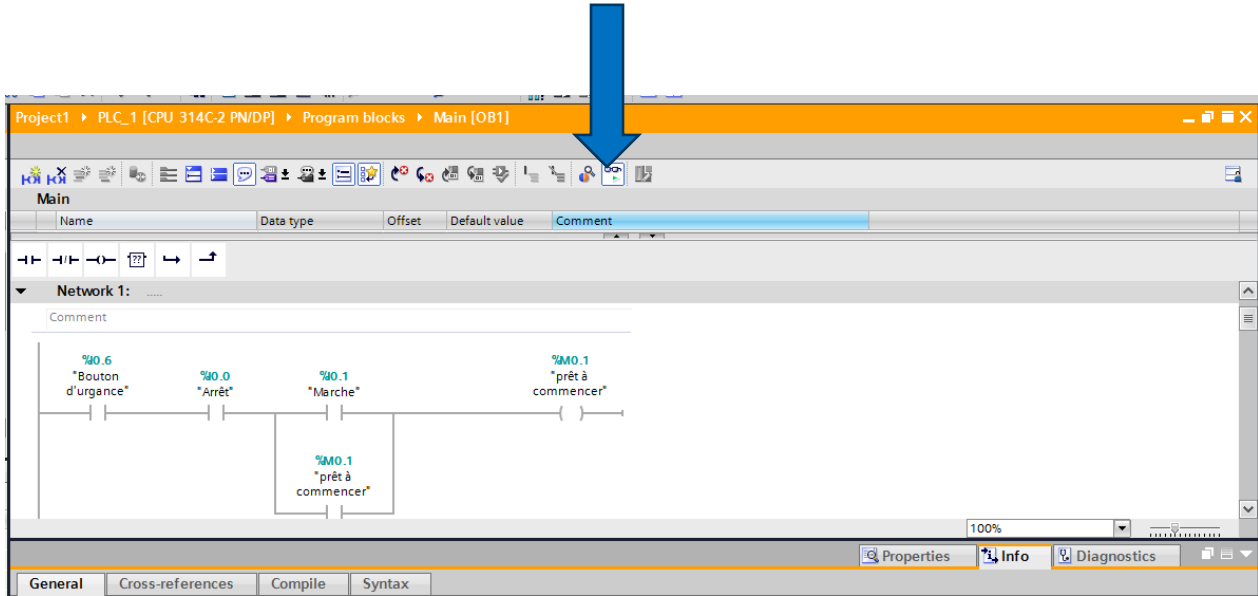


Figure 3.13 démarrage du Monitoring

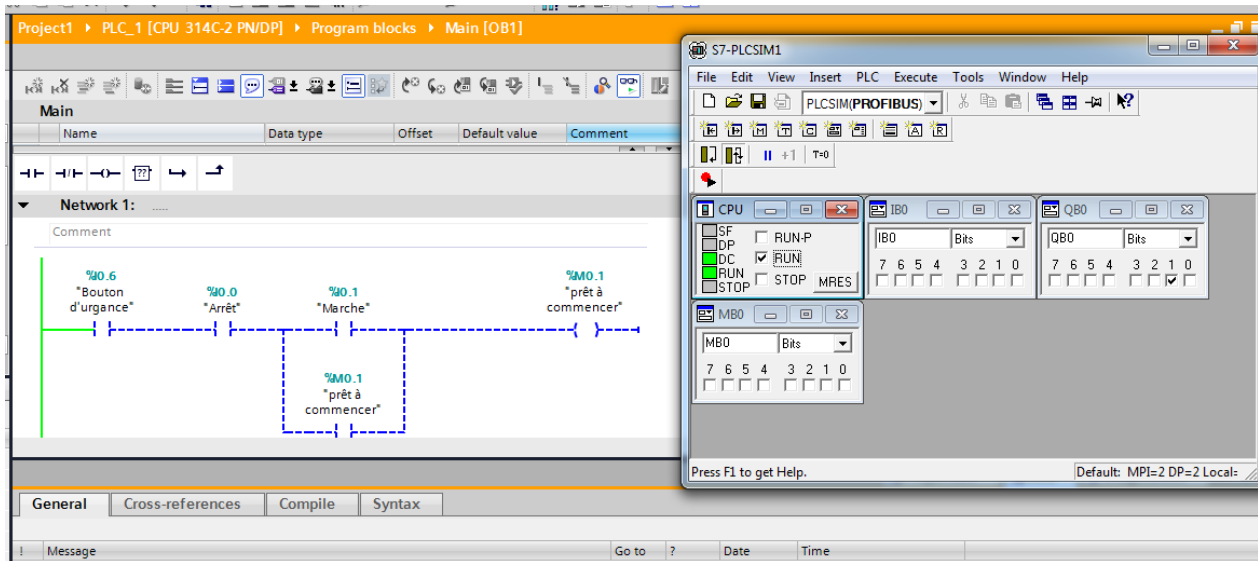


Figure 3.14 run le PLCSIM



Figure 3.15 changement entre le mode édition et le mode exécution

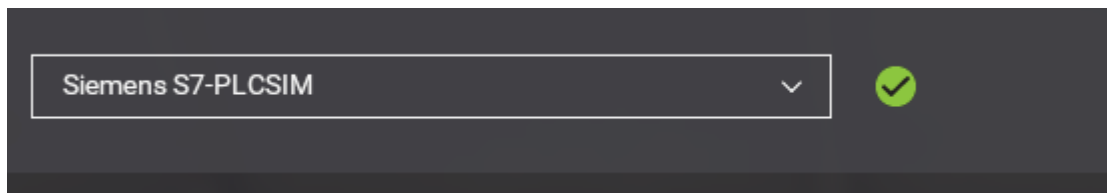


Figure 3.16 FACTORY IO été connecté avec PLCSIM



Figure 3.17 les affichages du compteur et du poids



Figure 3.18 Pusher pousse une boîte lourde

3.6 Programmation sur TIA Portal V13

3.6.1 Configuration matérielle

La création d'un projet sur TIA Portal V13 commence par la définition d'une configuration matérielle, dont l'intérêt est de spécifier les modules d'alimentation, d'entrée et de sortie pour le projet, afin de le connecter avec le CPU correspondante au niveau de l'automate.

Les exigences matérielles et logicielles sont les suivantes :

- SIMATIC S7-300
- CPU : CPU 314C-2 PN/DP
- DI/DO : DI 24/DO 16
- AI/AO : AI 5/ AO 2
- AO : AO 4

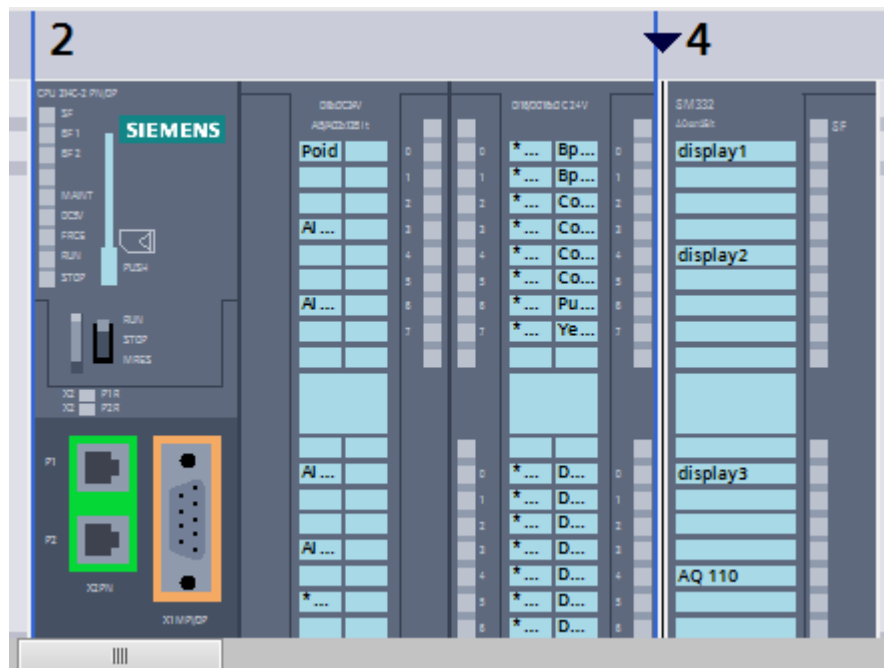


Figure 3.19 Configuration matérielle

3.6.2 Table des mnémoniques (Les variables d'API)

La table des mnémoniques nous permet de définir la liste des variables qui seront utilisées lors de la programmation. Il nous permet de désigner l'ensemble des entrées et sorties de notre système, La définition des variables au niveau de la table des variables automate.

	Name	Data type	Address	Retain	Visibl...	Acces...	Comment
1	Arrêt	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bouton arret
2	Marche	Bool	%I0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bouton marche
3	Vv	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	voyant vert
4	Vr	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	voyant rouge
5	prêt à commencer	Bool	%M0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	systeme est prêt à commencer
6	ConvoyeurM	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	premier convoyeur Marche
7	ConvoyeurPM	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	convoyeur peseur Marche
8	Convoyeur2M	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	deuxième convoyeur Marche
9	Convoyeur3M	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	troisième convoyeur Marche
10	Pusher	Bool	%Q0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	le poussoir
11	Poids	Int	%IW100		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	la valeur entre du poids
12	VJ	Bool	%Q0.7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	voyant jaune

Figure 3.20 Variables API 1/2

13	display	Int	%QW102		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	affichage du poids
14	display1	Int	%QW104		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	affichage le nombre du boite lourd
15	display2	Int	%QW106		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	affichage le nombre du boite leger
16	display3	Int	%QW108		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	affichage le nombre du boite moyenne
17	capteur	Bool	%I0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Capteur photoélectrique diffus
18	Rle	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	réinitialisation leger
19	RM	Bool	%I0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	réinitialisation moyen
20	RL	Bool	%I0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	réinitialisation lourd
21	Br	Bool	%I0.6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	bouton d'urgence
22	affiche poids	DInt	%MD9		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	affichage du poids
23	Dll	Bool	%M1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	detection leger
24	DL	Bool	%M1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	detection lourd
25	poid objet	DInt	%MD7		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	arrondissement du poids vers unité
26	résultat obtenu	Real	%MD6		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	le poids de boite
27	Ei	Word	%MW2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	erreur information
28	uniP	Bool	%M1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	unipolaire

Figure 3.21 Variables API 2/2

3.6.3 Le programme d'API

Dans les figures suivantes, nous présenterons le programme du système au TIA PORTAL avec le langage LADDER.

Dans le premier réseau on a le début du programme et départ du système.

Le bouton d'urgence et bouton d'arrêt ce sont des interrupteurs normalement fermés alors le système ça marche lorsque on appuie sur le bouton poussoir marche, et s'arrête quand on appuie sur le bouton arrêt ou bouton d'urgence après le cycle va continuer avec le memento %M0.2.

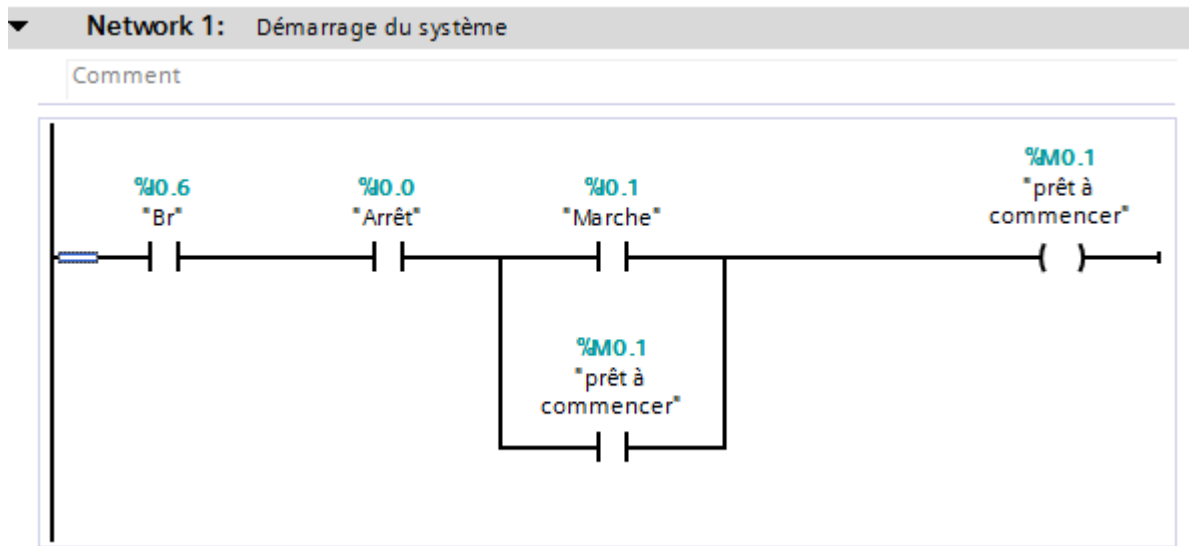


Figure 3.22 démarrage du système

Les réseaux de 2 à 4 nous montrent que le voyant vert s'allume et le voyant rouge s'éteint et tout le convoyeur marche à cause de memento %M0.1.

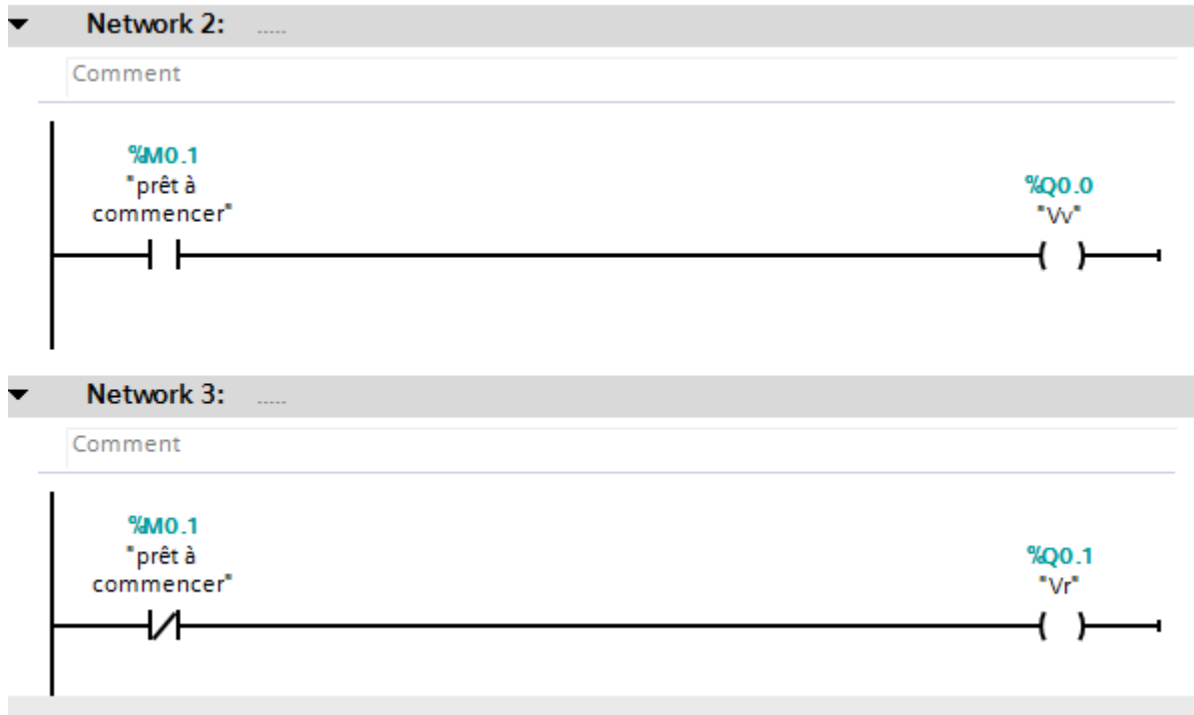


Figure 3.23 allumer et éteindre les voyant vert et rouge

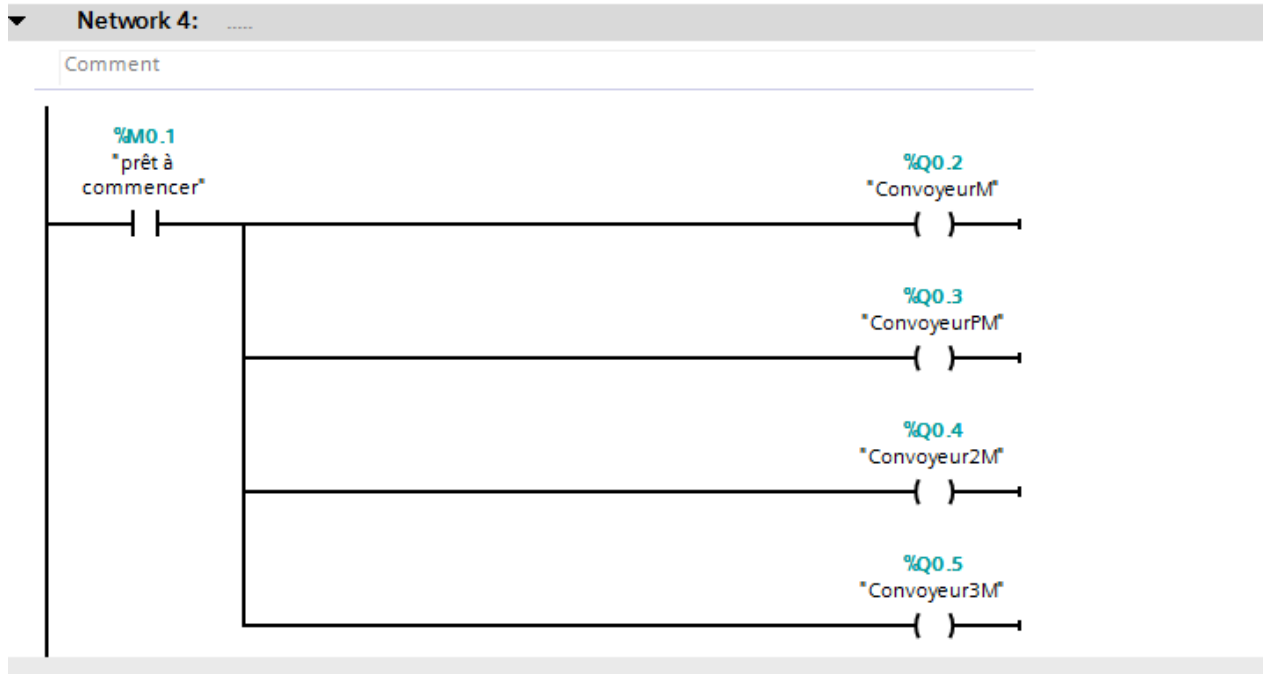


Figure 3.24 démarrage des convoyeurs

Le réseau 5

Utilisez l'instruction "Scale" pour convertir l'entier au paramètre IN en un nombre à virgule flottante, qui peut être mis à l'échelle en unités physiques entre une valeur limite inférieure et une valeur limite supérieure. Les paramètres LO_LIM et HI_LIM vous permettent de définir la limite inférieure et la limite supérieure de la plage de valeurs sur lesquelles la valeur d'entrée est mise à l'échelle. Le résultat de l'instruction est sorti au paramètre OUT.

IN : Valeur d'entrée à mettre à l'échelle et ou le peseur été connecté

HI_LIM : limite inférieure c'est 20

LO_LIM : limite supérieure c'est 0

BIPOLAR : Indique si la valeur du paramètre IN doit être interprétée comme bipolaire ou unipolaire. Le paramètre peut prendre les valeurs suivantes :

1 : Bipolaire

0 : Unipolaire

RET_Val : erreur information (si le IN est supérieur ou inférieur à HI_LIM ou LO_LIM)

OUT : résultat de l'instruction [13]

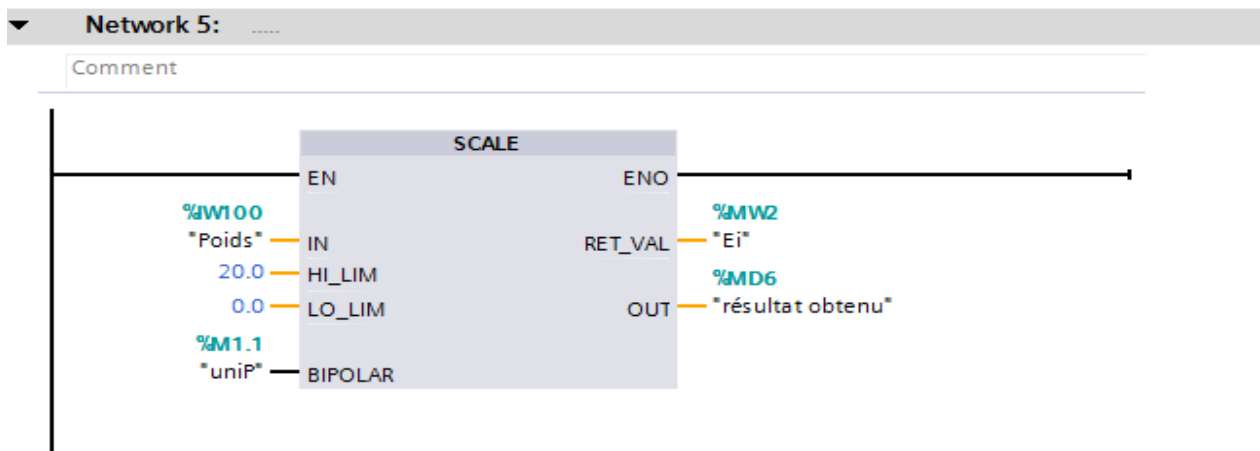


Figure 3.25 convertir l'entier IN en un nombre à virgule flottante

Le peseur ça fonction entre 0 et 10 volt et instruction " scale " convertir le voltage vers le poids

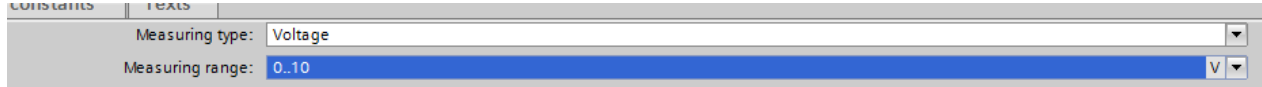


Figure 3.26 modification du matériel dans le TIA PORTAL pour être compatible avec le peseur

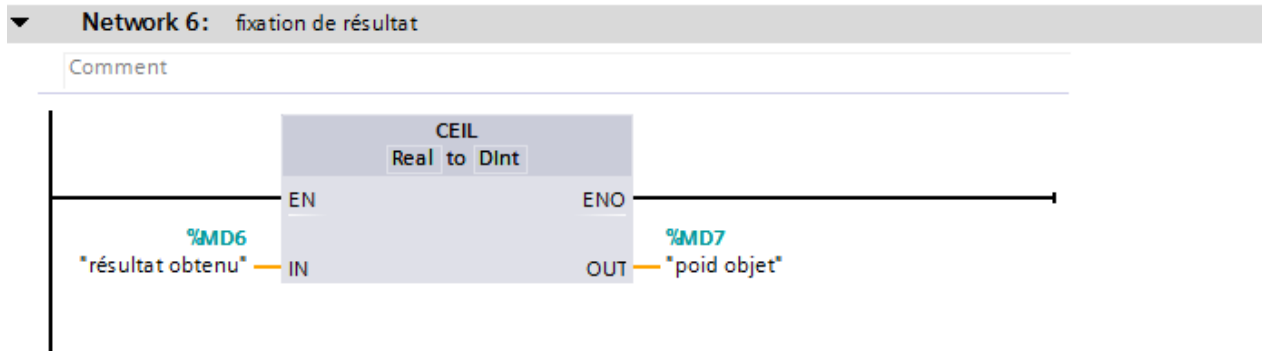


Figure 3.27 arrondissement de la valeur obtenu du poids

Réseaux 7 et 8 c'est l'activation des mémoires %M1.4 %M1.5 à partir de la comparaison de poids pour connaitre c'est le poids est lourd ou léger, ensuite après la comparaison du poids on va attendre un certain temps avec le temporisateur TON pour être l'objet complètement dans le peseur.

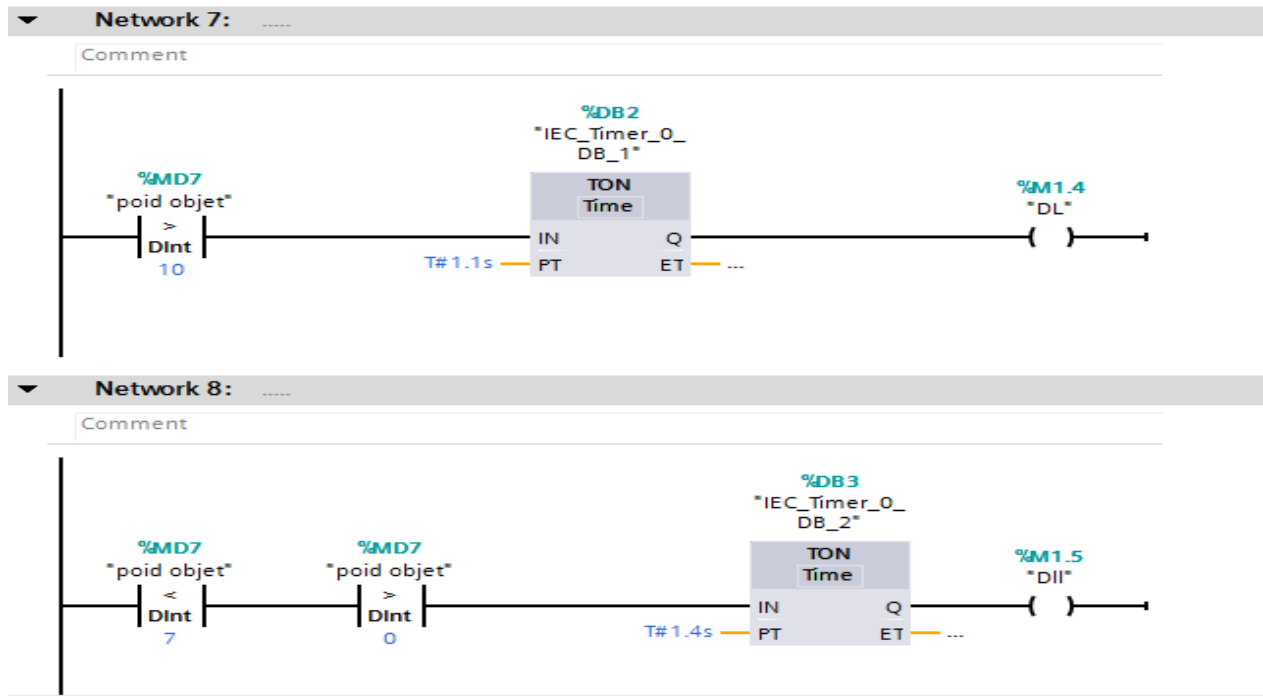


Figure 3.28 la comparaison d'objet pour être le poids dans les normes

Le réseau 9 montre que si les mémoire %M1.4 ou %M1.5 est activé le voyant jaune s'allume et le poussoir sortir pendant 1.8 seconds avec le temporisateur TP pour rediriger les boites vers la partie de recyclage.

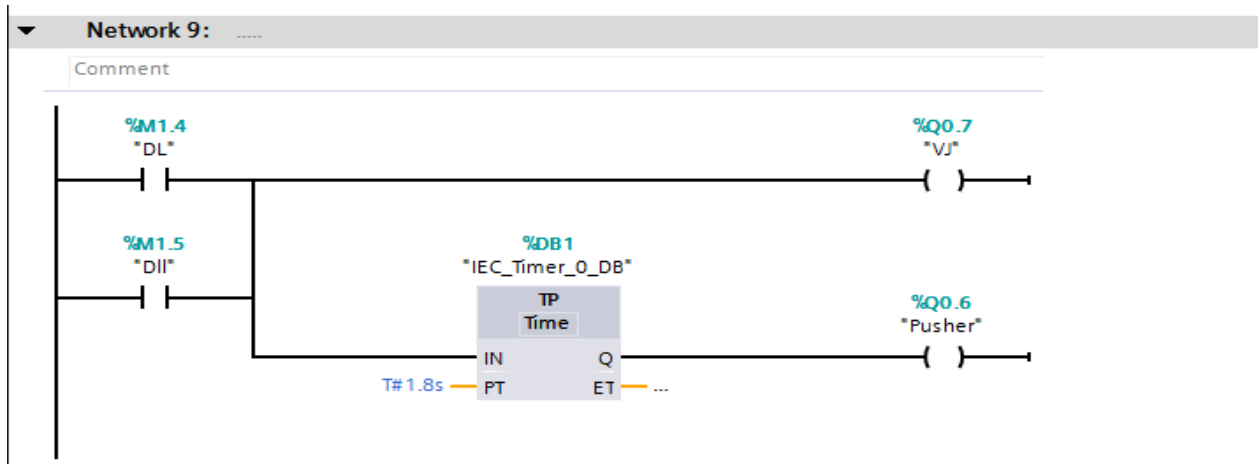


Figure 3.29 l'activation de voyant et Pusher

Réseau 10 on ajoute 17 à la valeur du poids pour les boites dans la simulation sois comme les bouteilles de venus en gramme.

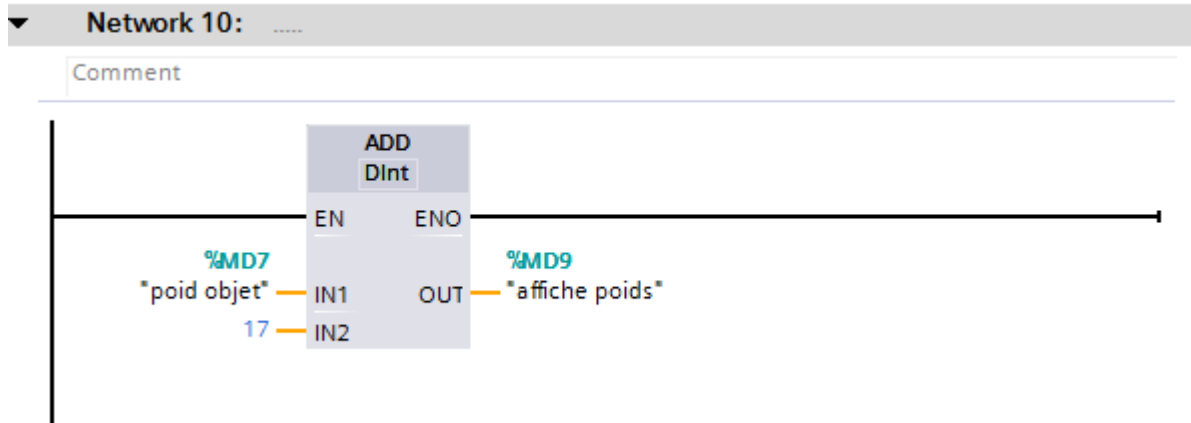


Figure 3.30 addition à la valeur de poids

Les réseaux 11 et 12 pour afficher la valeur du poids des boites dans l'affichage mais la valeur arrondis nous avons déjà ajouté 17 alors si la valeur du poids est supérieure à 0 (il y'a un objet dans le peseur) on l'affiche. Si le poids est égal à 0 on l'affiche la valeur précédente (seulement la valeur arrondir) sans addition. Paracerque si on affiche la même valeur, l'affichage montre nous la valeur 17 même si il n'y a pas un objet dans le peseur.

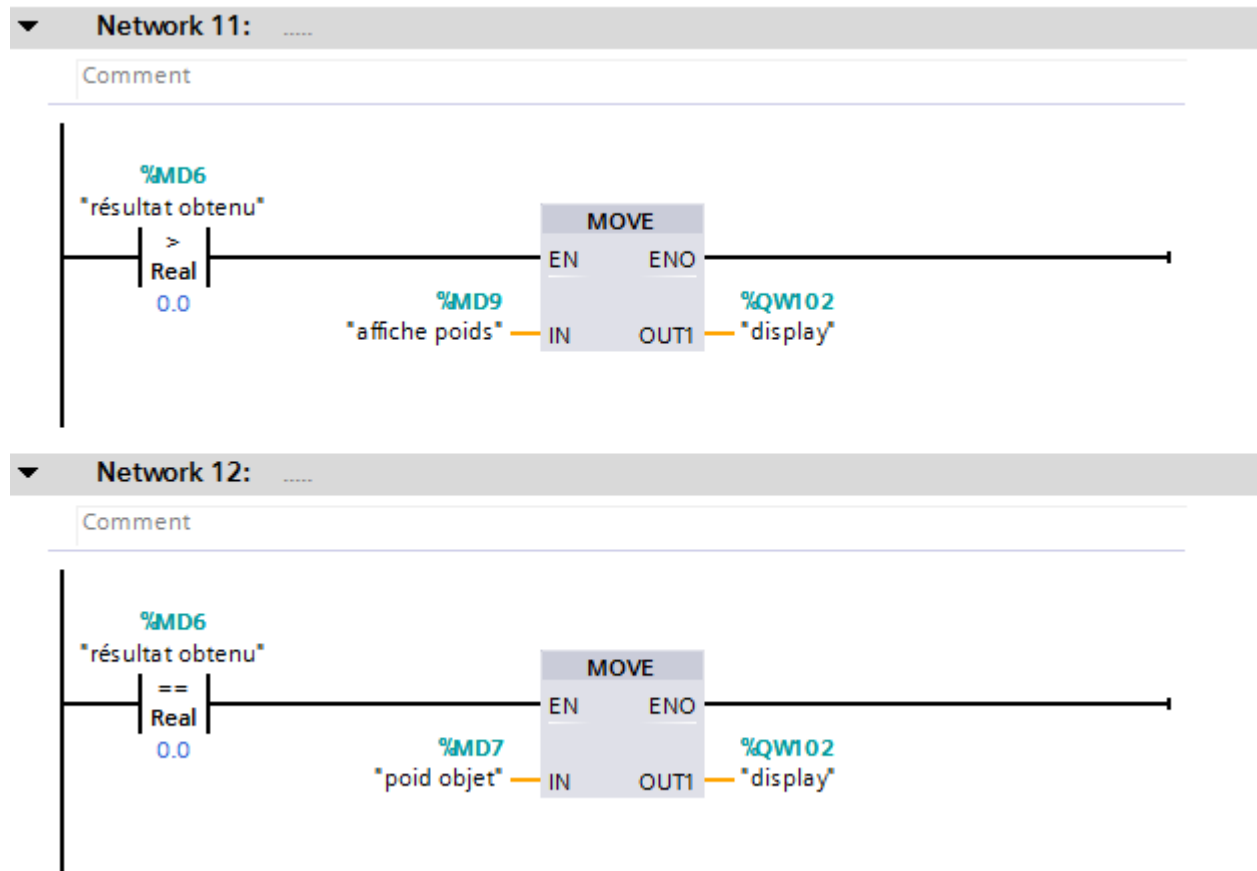


Figure 3.31 affichage du poids

Les 3 dernière réseaux 13,14 et 15 pour compter les boites lourde moyenne et légère

Et affiche le nombre de ces bouteilles dans différent affichage et il y'a 3 bouton pour réinitialiser chaque compteur.

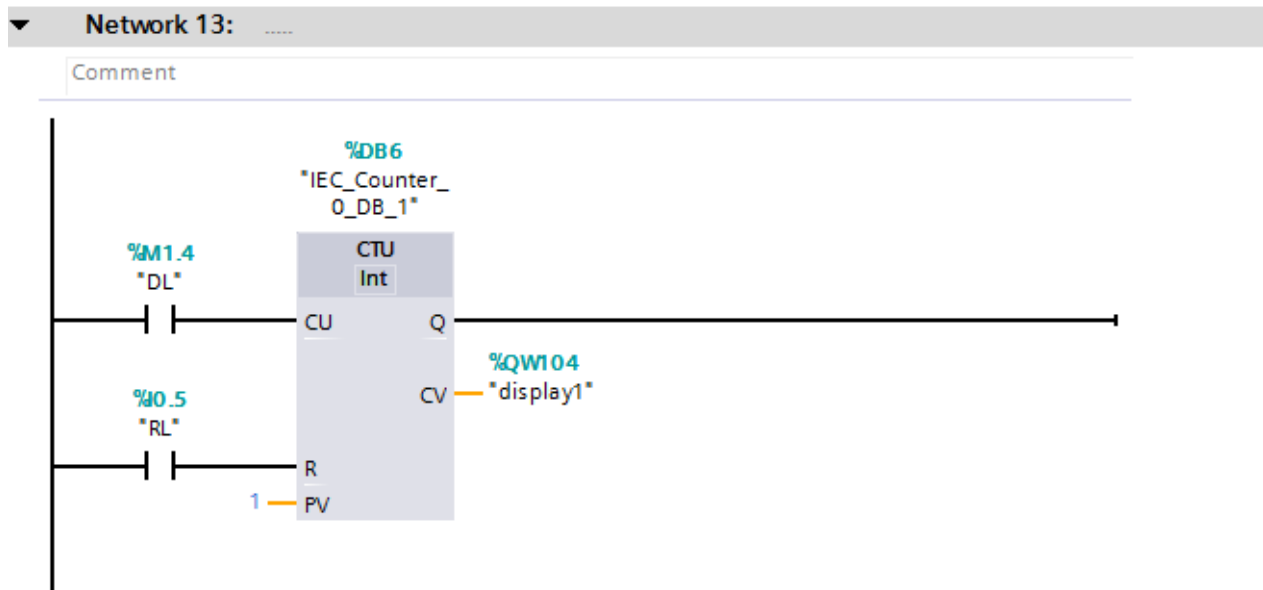


Figure 3.32 comptage des boites Lourde

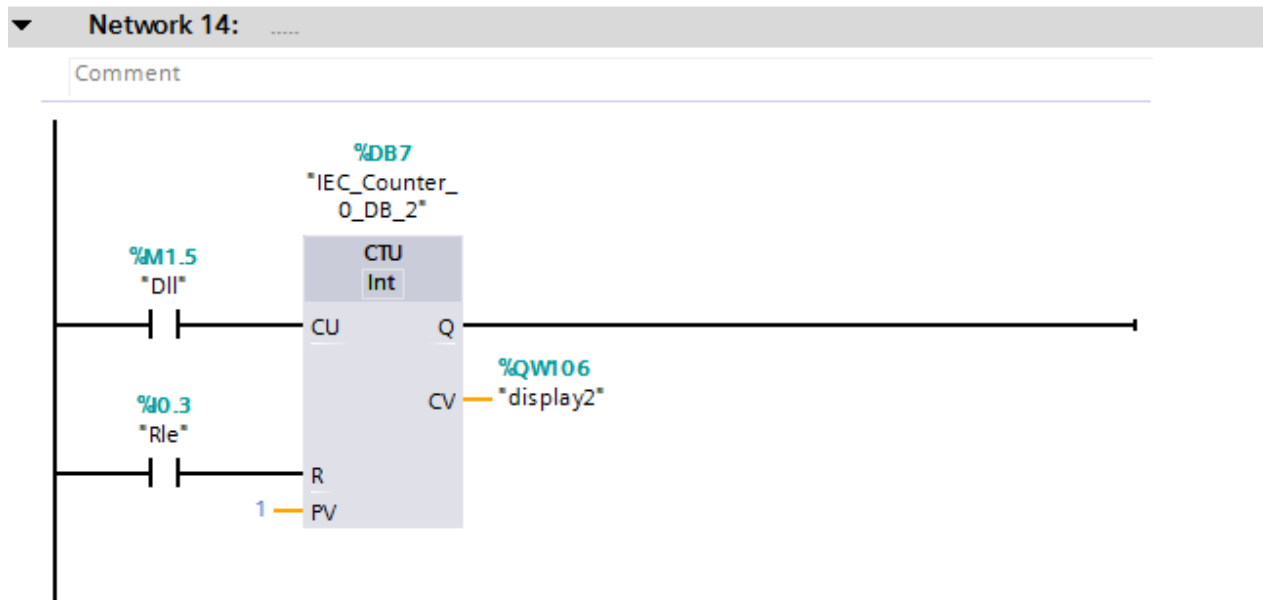


Figure 3.33 comptage des boites légère

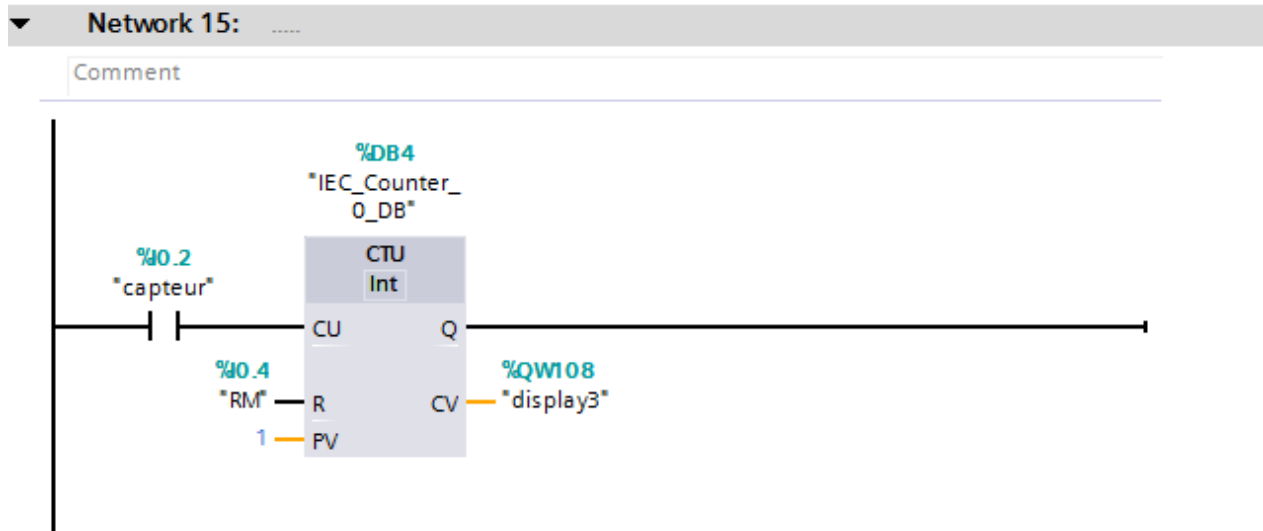


Figure 3.34 comptage des boites moyenne

3.7 Conclusion

Dans ce chapitre nous avons présenté le système à pesage automatique et la simulation du ce système de triage automatique pour les boites moyennes lourds et légères à partir du logiciel FACTORY I/O et TIA PORTAL V13

Le chapitre suivant présentera la carte Arduino et les composants utilisé dans la réalisation du ce projet.

Chapitre 4 : Présentation de la carte Arduino et Réalisation du Projet

4.1 Introduction :

Dans ce chapitre, nous allons présenter dans la première partie la description théorique et générale sur la carte Arduino et divers composants utilisés dans ce projet et logiciel de l'Arduino. Et dans la deuxième partie la réalisation du projet de pesage automatique.

4.2 Définition de la carte Arduino

Le module Arduino est un circuit imprimé en matériel libre : les plans sont consultables et tout le monde peut fabriquer une copie exacte de l'Arduino ce qui veut dire que vous pouvez en faire ce que voulez du moment que vous partagez ou distribuez votre travail sous les mêmes conditions. Contrairement à la plupart des objets manufacturés de notre environnement comme les machines à laver, les téléphones, les ordinateurs, seuls le nom et le logo sont réservés, dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. [14]

Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobile. [14]

En gros, on y branche des capteurs, le programme de l'Arduino traite les informations données par le capteur et déclenche des actions (comme allumer ou éteindre, augmenter, diminuer...).

La carte Arduino Uno est basée sur un microcontrôleur ATmega328 cadencé à 16 MHz et un régulateur de tension +5 V. Des connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires.

Pour pouvoir le programmer, il suffit de la connecter à un ordinateur à l'aide d'un câble USB, puis vérifiez le port, après cela, vous pouvez envoyer le code à l'Arduino.

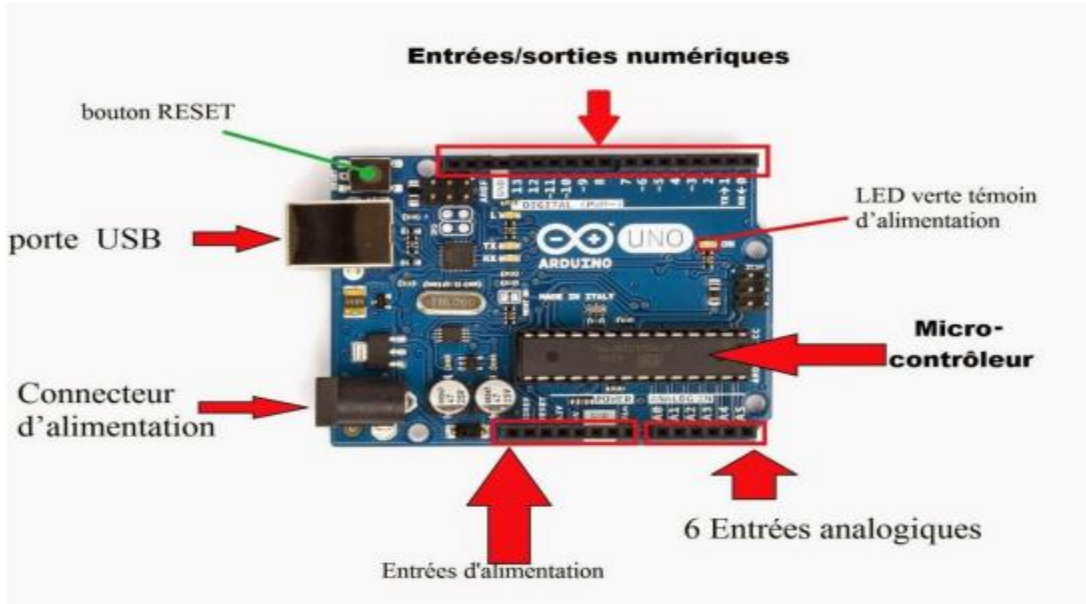


Figure 4.1 : La Carte ARDUINO UNO

4.2.1 Le Microcontrôleur ATmega328

Un microcontrôleur ATmega328 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors; les résistances et les condensateurs tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure 4.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino. [15]



Figure 4.2 : Microcontrôleur ATmega328

4.2.2 Caractéristiques de la carte Arduino UNO

Microcontrôleur	ATmega328
Fréquence d'horloge	16MHZ
Tension d'alimentation interne (usb)	5V
Tension d'alimentation externe	6-20V
Portes numériques	14 entrées et sortie
Portes analogiques	6 entrées analogique

Courant max par broches E/S	40 mA
Courant max sur sortie 3,3V	50mA
Mémoire Flash	32 KB
Mémoire SRAM	2 KB
Mémoire EEPROM	1 KB84
Dimensions	68.6mm x 53.3mm

Tableau 4.1 : Caractéristiques de la carte Arduino UNO

4.3 Programmation avec l'IDE Arduino

4.3.1 Lancement du logiciel :

Comme toutes les carte électronique qui, leurs constructions se basent sur un microcontrôleur doivent être dotées d'une interface de programmation comme est le cas de notre carte Arduino Uno.

Démarrez le programme en double-cliquant sur l'icône avec le symbole vert "infini". Il s'agit de l'exécutable du logiciel. Après un bref moment de réflexion, l'image suivante apparaît:

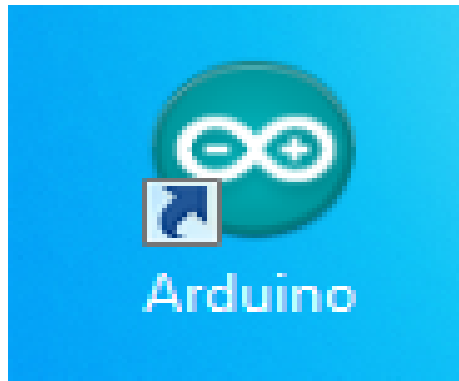


Figure 4.3 : Icône exécutable de l'Arduino

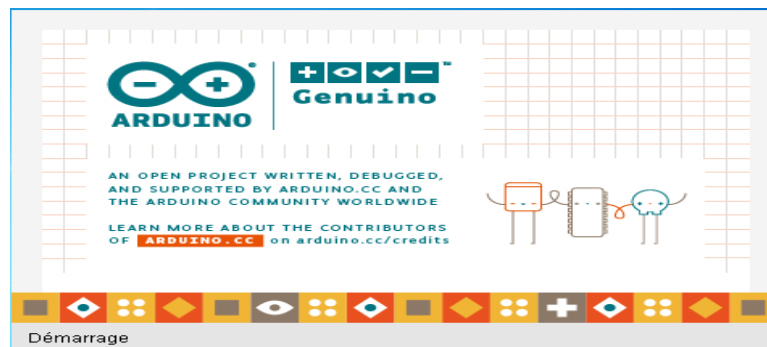


Figure 4.4 : Le démarrage de l'Arduino software

4.3.2 l'environnement de la programmation

Sur un ordinateur, le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C) , avec des fonctions et des bibliothèques spécifiques à ARDUINO (gestion des entrées /sorties). Une fois le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte au travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.

L'environnement de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux). [16]

4.3.3 Structure générale du programme (IDE Arduino)

Le logiciel de programmation d'Arduino est connu par son interface souple et simple qu'on peut exécuter sur n'importe quel système d'exploitation déjà cités. L'interface du logiciel Arduino se présente de la façon suivante :

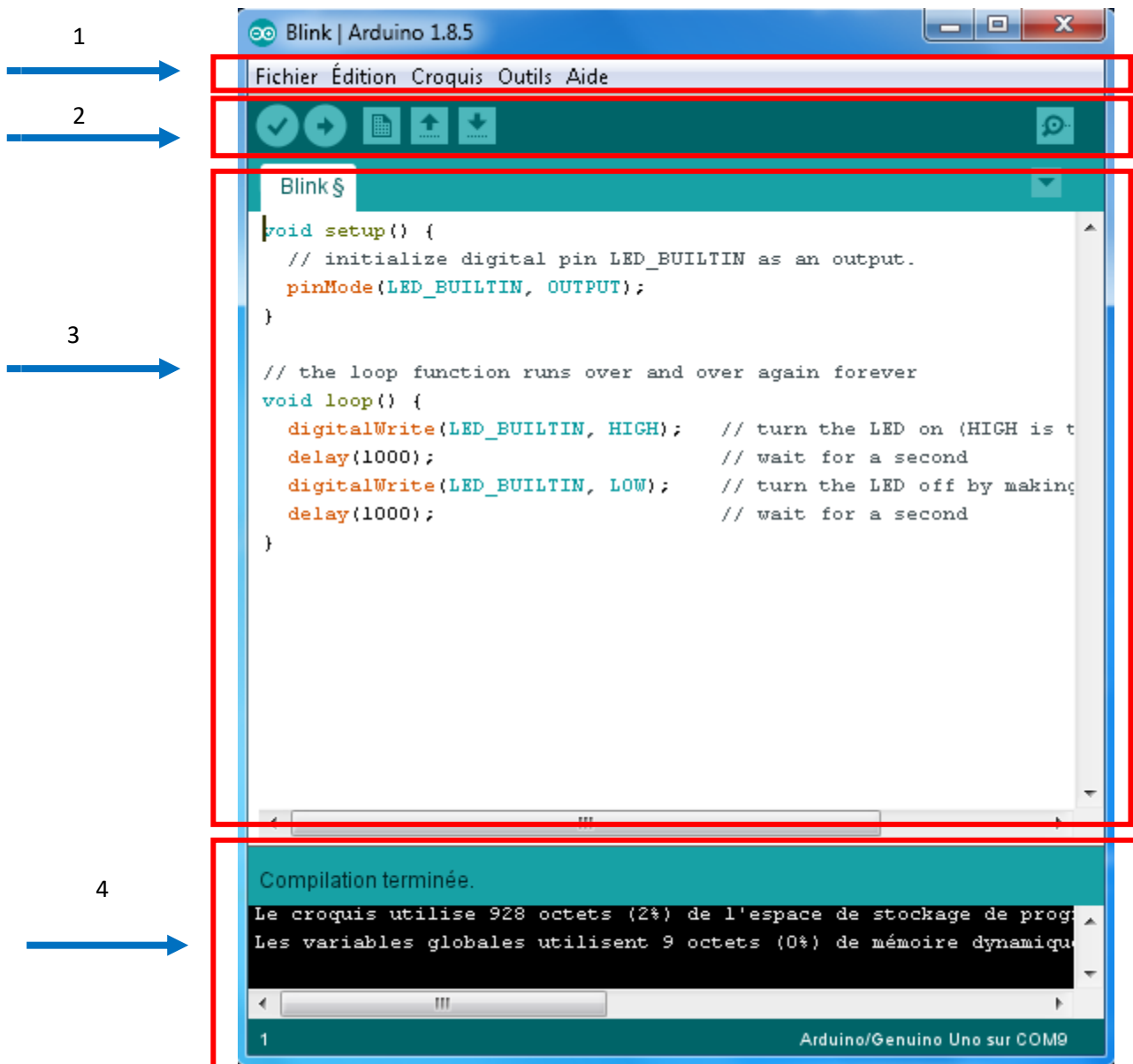


Figure 4.5 : Interface IDE Arduino

- 1 : Les options de configuration du logiciel
- 2 : Boutons pour la programmation des cartes
- 3 : Programme à créer
- 4 : Débogueur (affichage des erreurs de programmation)

4.4 Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (COM 3) comme à titre d'exemple cette figure suivante.

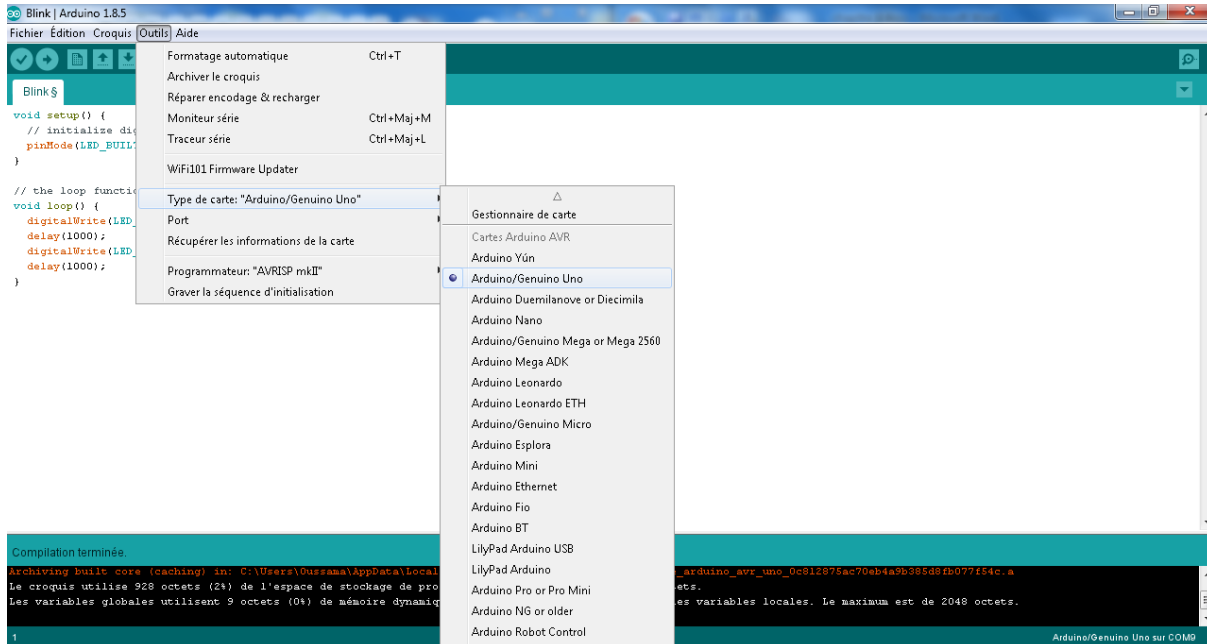


Figure 4.6 : Sélection de carte Arduino

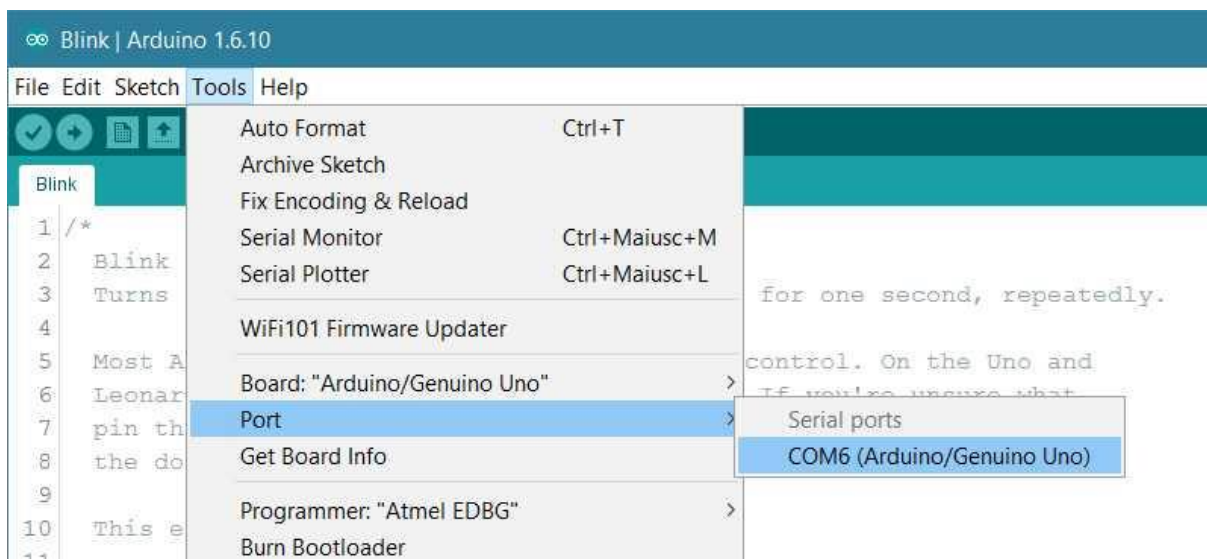


Figure 4.7 : Sélection du port

4.5 Description du programme

Après avoir téléchargé et installé l'IDE de l'arduino et avant de commencer son premier projet, il est primordial d'apprendre la syntaxe du langage de programmation sur l'IDE de l'arduino. Le langage Arduino est basé sur les langages C/C++.

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. Le microcontrôleur lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code.

la Structure générale du programme est comme suite :

- **Définition des variables et constantes et librairies** : Dans cette partie, on déclare des éléments utilisés tout au long du programme : les constantes (statiques) et les variables (dynamiques). Ce sont des emplacements mémoire utilisés pour stocker des données (des valeurs) utilisables dans la suite du programme. et déclaration d'une bibliothèque comme `#include <Servo.h>` Dites à l'éditeur Arduino que vous souhaitez accéder aux fonctions de la bibliothèque Servo avec cette déclaration au début de votre esquisse, avant la fonction de configuration.
- **Void setup()** (Initialisation) : C'est une partie initialisation et configuration des entrées/sorties. Elle est appelée et n'est exécutée qu'une seule fois au démarrage du programme, Les broches numériques de l'Arduino peuvent aussi bien être configurées en entrées qu'en sorties.
- **Void loop()** (exécution) : loop (boucle en anglais) c'est la partie principale contenant le programme, fait exactement ce que son nom suggère et s'exécute en boucle sans fin, La fonction loop() est obligatoire, même vide, dans tout programme.

4.5.1 Commentaires

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme : c'est un moyen qu'on utilise pour ne pas s'égarer entre les lignes de codes écrits, et ça permet de retrouver la ligne qu'on veut modifier en cas d'erreurs de syntaxe.

1. `//` (commentaire simple ligne)

2. `/*-----`

`-----`

`-----`

`*/` (commentaire multi-lignes)

4.5.2 Les variables

Une variable est un espace réservé dans la mémoire de l'ordinateur. Elle est caractérisée par un nom qui permet d'y accéder facilement. Il existe différents types de variables identifiés par un mot-clé dont les principaux sont :

- nombres entiers (int)
- nombres à virgule flottante (float)
- texte (String)
- valeurs vrai/faux (boolean)

4.5.3 Configuration des entrées et des sorties void setup ()

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques, en utilisant l'instruction :

pinMode(nom, état) par exemple **pinMode**(ledPin, OUTPUT);

4.5.4 Programmation des interactions void loop ()

Dans cette partie du programme, on définit les opérations à effectuer, en utilisant des fonctions. Il est important à savoir aussi qu'il est nécessaire de terminer chaque ligne d'instruction par un point virgule sinon elle ne sera pas prise en considération par le compilateur et elle sera ignorée. Et ne pas oublier les accolades qui encadrent la boucle.

Les fonctions qu'on utilise souvent sont :

- **digitalWrite**
- **digitalRead**
- **int analogRead**
- **analogWrite** (PWM: écrire une valeur analogique)
- **delay** (temps en milliseconde) est la commande d'attente entre deux instructions.

4.6 Niveaux logiques des connecteurs numériques:

- HIGH (état 1)
- LOW (état 0)
- INPUT (configuré en entrée)
- OUTPUT (configuré en sortie)

Exemple: **digitalRead**(ledPin, HIGH).

4.7 Les étapes de téléchargement du programme

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte Arduino via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE Arduino.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on modifie le programme.
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

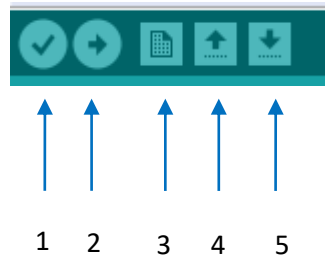


Figure 4.8 : définir les boutons de programmation

1 / Vérifier/Compiler : S'il n'y a pas d'erreur à la compilation, le message « compilation terminée » s'affiche suivi de la taille du programme.

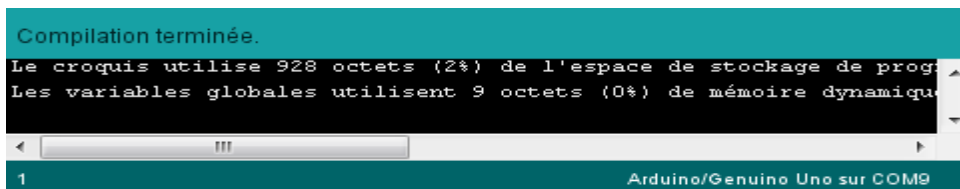


Figure 4.9 : compilation terminée

2 / Téléverser : Le programme est téléchargé sur l'arduino, le message « Téléversement terminé » s'affiche une fois le téléchargement terminé.

```
Téléversement terminé
Le croquis utilise 930 octets (3%) de l'espace de stockage de programmes. Le maximum est
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laisse 2039 o
```

Figure 4.10 : Téléversement terminé

3/ Nouveau : crée un nouveau fichier de programmation.

4/ Ouvrir : ouvrir un fichier déjà enregistré.

5/ Enregistre : pour enregistrer un fichier et le programme.

4.8 Les composants utilisés dans la carte Arduino

On va présenter dans cette partie le matériel que nous avons utilisé pour réaliser ce projet.

4.8.1 Les câbles

Il en existe deux types de câble essentiels pour la mise en œuvre des montages, qui sont présentés ci-dessous :

- **Câble USB** : qui sert à relier la carte Arduino à l'ordinateur, soit pour le chargement du programme, ou l'alimentation de la carte.



Figure 4.11 : Câble USB type A-B

- **Les fils de liaison** : qui servent à connecter les composants pour construire des montages des objets connectés. il ya 3 types male-male , femelle- femelle et femelle-male.

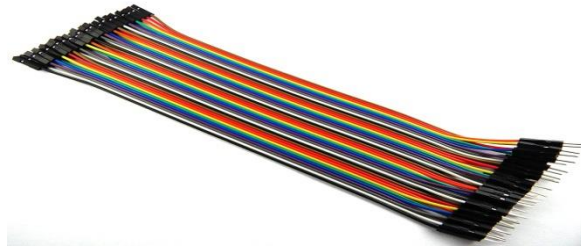


Figure 4.12 : Jumper wires (Les fils de liaison)

4.8.2 BreadBoard (plaque d'essai)

La plaque d'essai est une plaque en plastique isolant et pleine de trous. Elle sert à tester des montages avant de souder les composants.

La plaque est répartie en deux parties, partie extérieure où les trous sont reliés de façon horizontale, et la partie intérieure où les trous sont reliés de façon verticale.

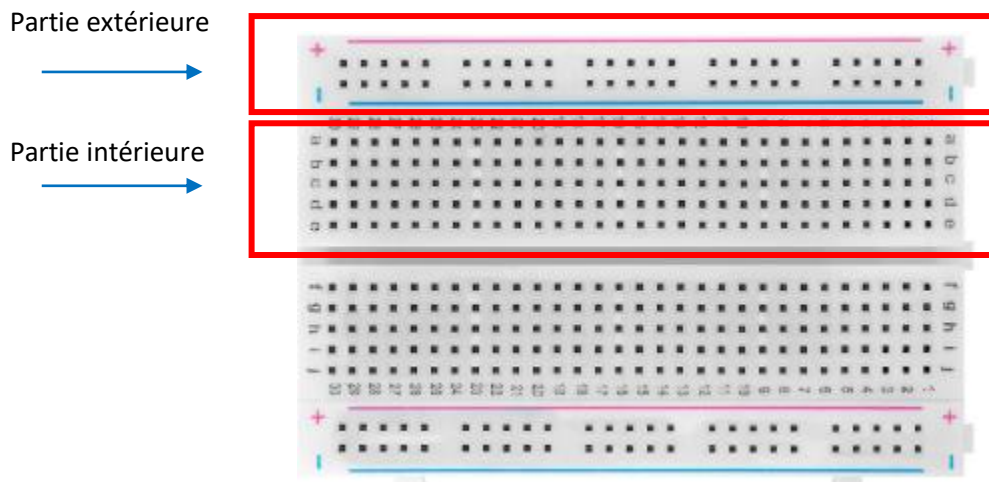


Figure 4.13 : BreadBoard (plaque d'essai)

4.8.3 Le Servomoteur

Le servomoteur (souvent abrégé en « servo », provenant du latin servus qui signifie « esclave ») est un système qui a pour but de produire un mouvement précis en réponse à une commande externe, C'est un actionneur qui mélange l'électronique, la mécanique et l'automatique. Il est composé :

- D'un moteur à courant continu
- D'un axe de rotation
- D'un capteur de position de l'angle d'orientation de l'axe

- D'une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu

Un servomoteur est capable d'attendre des positions prédéterminées dans les instructions qui lui ont été données, puis de les maintenir. Il a l'avantage d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera la consigne d'instruction que vous lui avez envoyée en son entrée. Même si un obstacle se trouve sur la route, qui viendrait à lui faire changer l'orientation de sa trajectoire, le servomoteur essaiera de conserver la position. Pour un ajustement précis de la position, le moteur et son réglage sont équipés d'un système de mesures qui détermine la position courante. [17]



Figure 4.14 : Servomoteur sg90

Câblage du Servomoteur :

Marron : pour GND

Rouge : pour VCC -> 5v

Orange : pour signal

4.8.4 Capteur infrarouge

Lorsque le module détecte un obstacle, le voyant vert s'allume au niveau de la carte, tandis que le port OUT maintient une sortie avec un signal low, le module détecte sur une distance de 2 ~ 30cm, l'angle de détection est de 35°, le potentiomètre de réglage s'ajuste pour la distance. Dans le sens horaire, détection pour l'augmentation de la distance; potentiomètre de réglage dans le sens antihoraire, réduisant la distance de détection. Le port OUT du module de capteur peut être directement connecté au microcontrôleur. IO peut également piloter directement un relais 5V. Le module peut être alimenté par une alimentation 3-5V DC. Le voyant rouge d'alimentation s'allume lorsque le module est sous tension. [18]

Utilisation du capteur IR est :

- Détection des obstacle dans un rebot
- Suiveur de ligne
- Détection et comptage des objets

Câblage du Capteur infrarouge

Le capteur a 3 pins VCC vers 5v GND vers GND et OUT vers digital pin de l'arduino

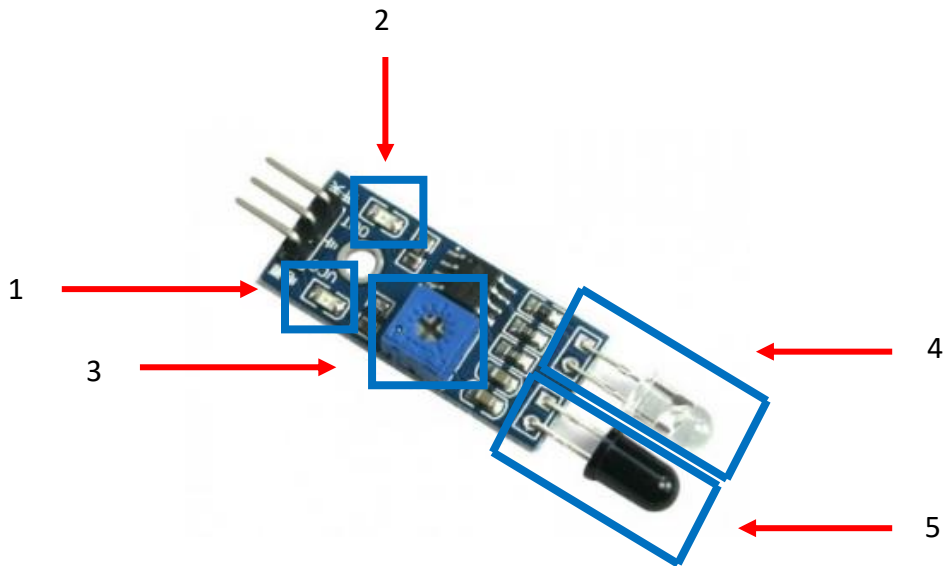


Figure 4.15 : détecteur d'objet capteur infrarouge

1. LED alimentation
2. LED détection d'objet
3. Potentiomètre (réglage de l'intervalle de détection)
4. IR (transparente) c'est émetteur
5. IR (noire) c'est récepteur

4.8.5 Cellules de charge à jauge de contrainte

Jauge de contrainte est un élément mécanique dont la force est détectée par la déformation d'une jauge de contrainte sur l'élément.

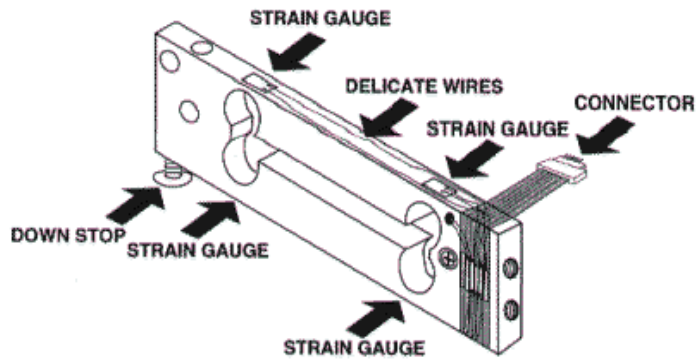


Figure 4.16 : Diagramme de cellule de charge à jauge de contrainte

Dans les cellules de charge à jauge de contrainte à barre, la cellule est configurée en formations en "Z" de sorte qu'un inertie de rotation est appliqué à la barre et les quatre jauges de contrainte sur la cellule mesureront la distorsion de flexion, deux mesurant la compression et deux la tension. Lorsque ces quatre jauges de contrainte sont installées dans une formation de pont de Wheatstone, il est facile de mesurer avec précision les petits changements de résistance des jauges de contrainte. [19]

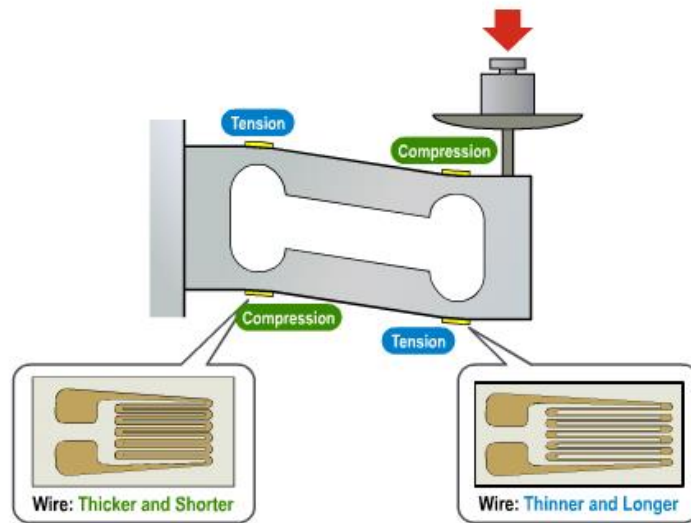


Figure 4.17 : Diagramme plus détaillé des jauges de contrainte sur les cellules de charge à barres lorsqu'une force est appliquée

La plupart des cellules de charge à jauge de contrainte fonctionnent de manière très similaire, mais peuvent varier en taille, en matériau et en configuration mécanique, ce qui peut conduire à ce que chaque cellule ait des charges maximales et des sensibilités différentes qu'elle peut gérer. Pour quelques configurations mécaniques possibles de la cellule de pesée. [19]

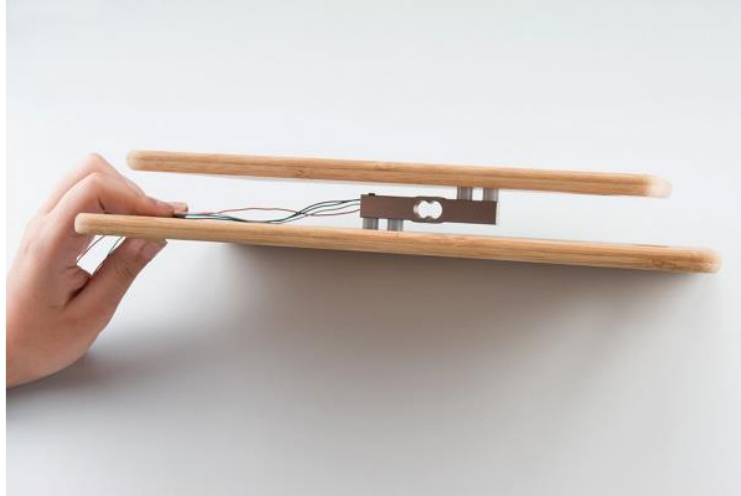


Figure 4.18 : Cellule de charge à barre entre une configuration à deux plaques

4.8.5.1 pont de Wheatstone :

Le circuit du pont de Wheatstone n'est rien de plus que deux simples arrangements série-parallèle de résistances connectées entre une borne d'alimentation en tension et la terre produisant une différence de tension nulle entre les deux branches parallèles lorsqu'elles sont équilibrées.

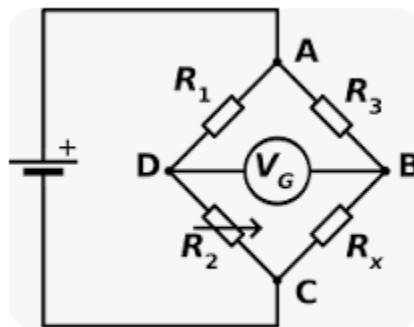


Figure 4.19 : Le circuit du pont de Wheatstone

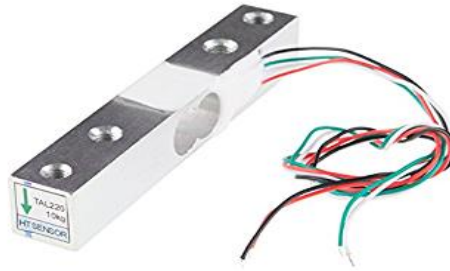


Figure 4.20 : jauge de contrainte

- Rouge et noir pour l'alimentation
- Vert et blanc pour mesure la tension
- Voltage : 5VDC
- Max Load : 5KG

4.8.6 MODULE HX711

Le HX711 est un convertisseur analogique-numérique (CAN) de précision à 24 bits conçu pour les balances de pesage et les applications de contrôle industriel pour s'interfacer directement avec un capteur de poids. Il est spécialement conçu pour amplifier les signaux des cellules et les rapporter à un autre microcontrôleur. Qui facilite la communication entre la cellule de charge et le microcontrôleur Il est assez facile à utiliser car il est capable de s'interfacer avec Arduino. [20]

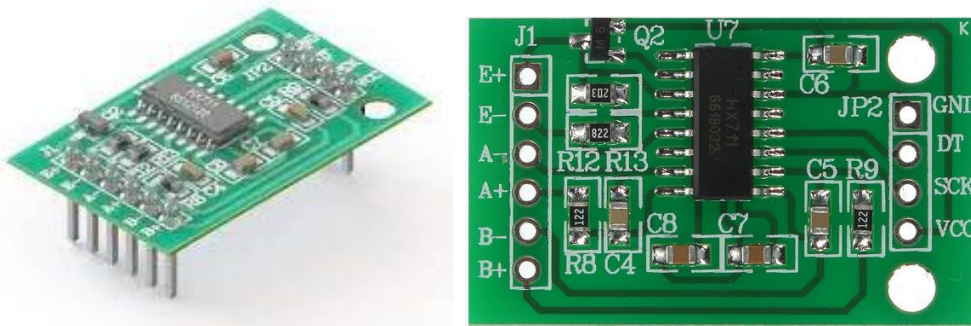


Figure 4.21 : module HX711

Spécifications du module ADC HX711 :

- Tension d'entrée différentielle : ± 40 mV
- Précision des données : 24 bits
- Fréquence de rafraîchissement : 10/80 Hz
- Tension de fonctionnement : 2,7 V à 5 VCC
- Courant de fonctionnement : < 10 mA

DT pour la transmission des bits de donnée

SCK pour le signal d'horloge

VCC vers 5 volt de l'Arduino

GND vers GND de l'Arduino

4.8.7 Afficheur LCD 16x2 (Liquide Crystal Display)

Les écrans LCD ou écrans à cristaux liquides sont de plus en plus courants dans notre environnement, que ce soit pour afficher des informations utiles ou pour servir de sélecteur de commande. Il existe des écrans de tout type allant du simple afficheur de caractères aux écrans géants couleur. Les écrans les plus couramment utilisés dans des applications Arduino sont les écrans à caractères alphanumérique.

Ils permettent d'afficher des lettres, des chiffres et quelques caractères spéciaux. Les caractères sont prédéfinis.

Grâce à la commande par un microcontrôleur il peut correspondre avec le monde extérieur sans passer par l'ordinateur ou le Serial Monitor. Afficheur LCD 16x2 a 2 lignes de 16 caractères chacune. [21]

L'afficheur a été utilisé pour afficher le poids et compter le nombre des bouteilles moyen lourd et léger.



Figure 4.22 : interface de LCD 16x2

4.8.8 Buzzer (beeper)

C'est une composante électronique émettant un signal sonore, l'intensité du son dépend de l'intensité du courant. Les utilisations typiques des buzzers comprennent les dispositifs d'alarme.

4.8.9 Potentiomètre (résistance variable)

Un potentiomètre est un type de résistance variable à trois bornes, dont une est reliée à un curseur se déplaçant sur une piste résistante terminée par les deux autres bornes. Les potentiomètres sont couramment employés dans les circuits électroniques. Ils servent par exemple à contrôler le volume d'une radio, la vitesse d'un moteur ou bien luminosité d'affichage de LCD.



Figure 4.23 : Potentiomètre

4.8.10 Bouton poussoir

Le bouton poussoir ou interrupteur poussoir, est la base de l'interactivité, entre l'homme et la machine. Il permet de commander un objet, une machine ou simplement une lumière. Il existe deux types de boutons poussoir, le normalement ouvert dit "NO" et le normalement fermé dit "NF".

Lorsque l'on appuie sur le bouton poussoir, celui-ci laisse passer le courant, et lorsqu'il revient à sa position initiale (simplement en lâchant la pression), il ne laissera plus passer le courant. C'est un bouton poussoir de type "NO".

Et inversement, lorsque l'on appuie sur le bouton, celui-ci ne laisse pas passer le courant, lorsque l'on relâche celui-ci laisse passer le courant. Il est de type "NC". [22]

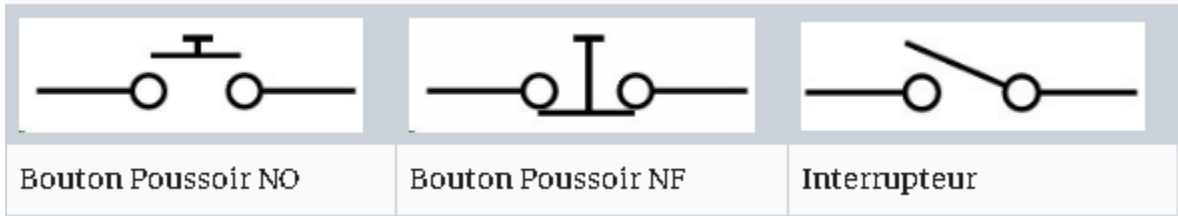


Figure 4.24 Symboles conventionnels des boutons [23]



Figure 4.25 Bouton poussoir

4.8.11 Led

La LED (Light Emitting Diode) est textuellement une “diode émettant de la lumière”. Elle est également appelée DEL (Diode électroluminescente), Les produits d'éclairage à DEL produisent de la lumière jusqu'à 90 % plus efficacement que les ampoules à incandescence. Un courant électrique passe à travers une micropuce, qui illumine les minuscules sources lumineuses que nous appelons LED et le résultat est la lumière visible.



Figure 4.26 Les Leds

4.9 La réalisation du projet avec Arduino

Ce système est basé sur un peseur qui mesure le poids d'un objet, si l'objet est dans les normes le premier servomoteur va tourner et pousser l'objet vers sa place, sinon l'objet est lourd ou léger, alors n'étant pas dans les normes, le deuxième servomoteur va tourner et pousser l'objet vers le convoyeur qui redirige les objets inutile vers le recyclage.

4.10 Montage et Câblage :

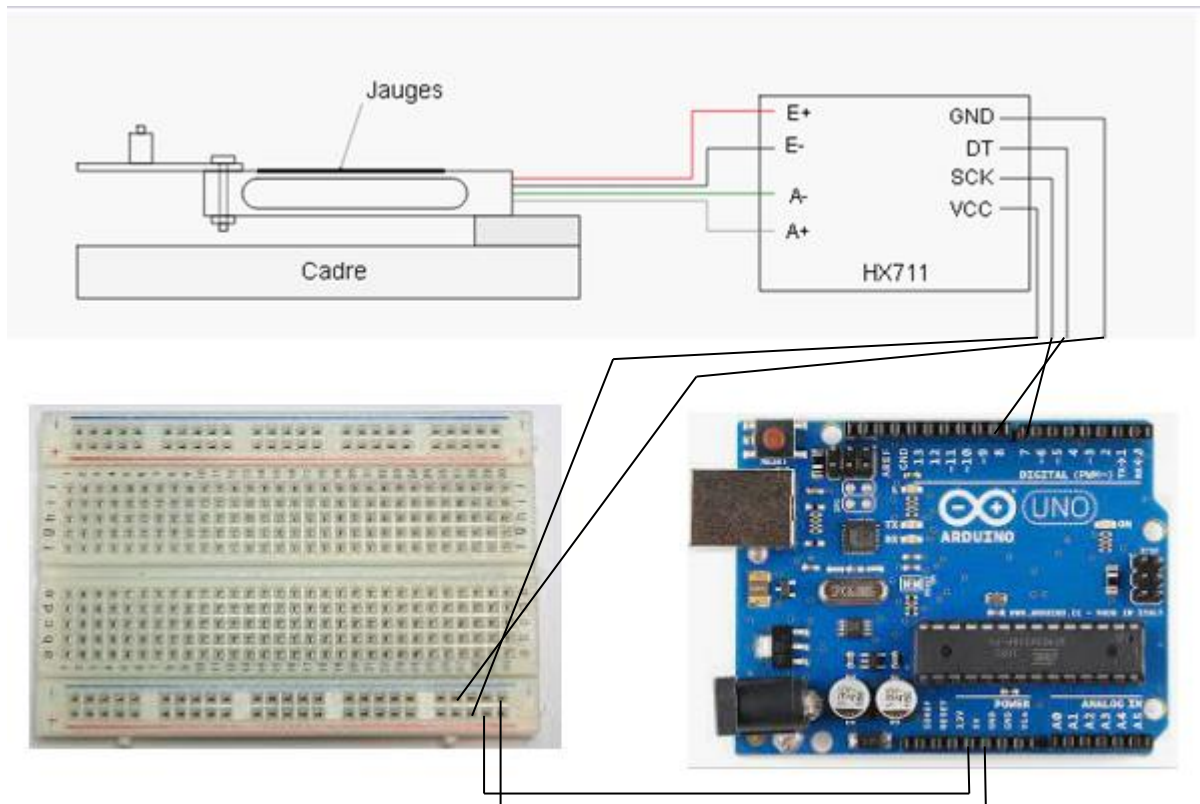


Figure 4.27 : branchement du jauge contrainte et HX711

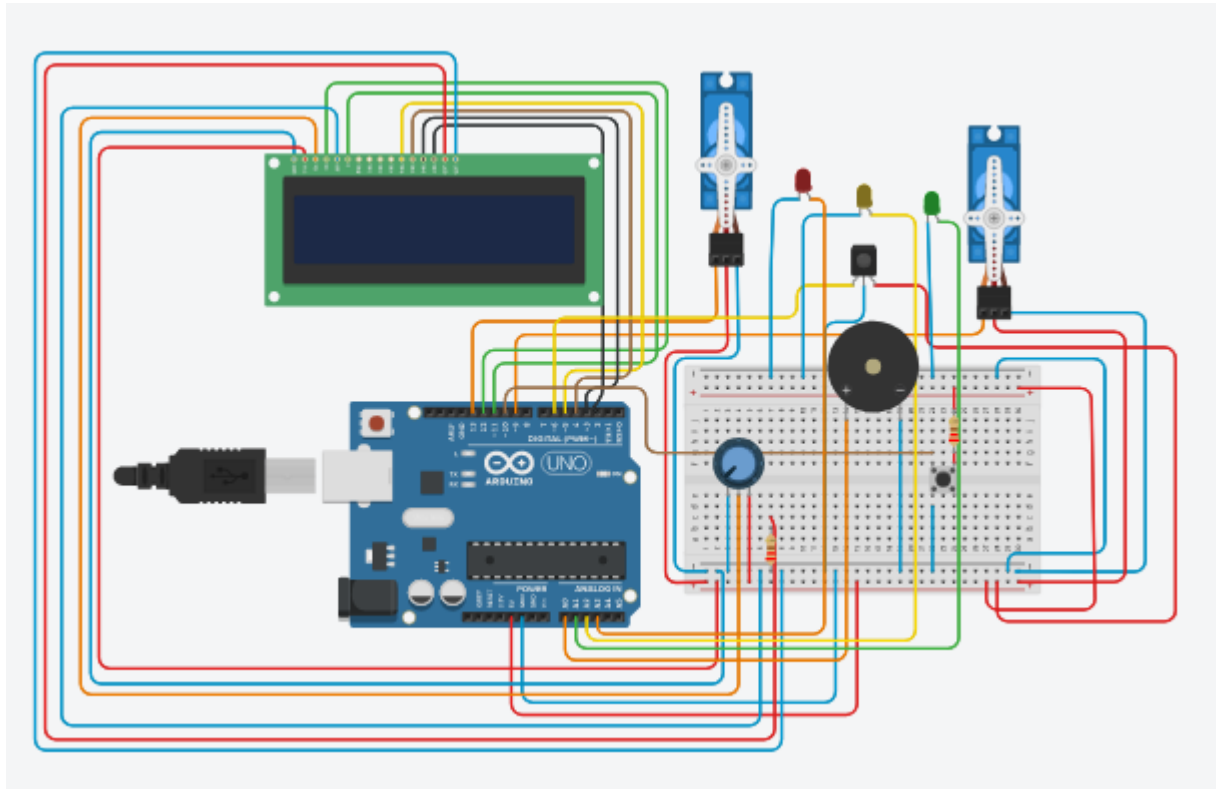


Figure 4.28 : Câblage les composants avec Arduino

composants		PIN
LCD	Vss	GND
	VDD	5V
	VO	Potentiomètre
	RS	12
	RW	GND
	E	11
	D4	5
	D5	4
	D6	3
	D7	2
	A	5V (avec résistance)
	K	GND

HX711	GND	GND
	SCK	7
	DT	8
	VCC	5V
Servomoteur Droite	GND	GND
	VCC	5V
	OUT	9
Servomoteur Gauche	GND	GND
	VCC	VCC
	OUT	13
Capteur Infrarouge	GND	GND
	VCC	5V
	OUT	6
Buzzer	Négative	GND
	Positive	A0
LED verte	Anode	A1
	Cathode	GND
LED jaune	Anode	A2
	Cathode	GND
LED rouge	Anode	A3
	Cathode	GND
Bouton Poussoir	Point a	5V (avec résistance)
		10
	Point b	GND

Potentiomètre	GND	GND
	VCC	5V
	OUT	VO (LCD)

Tableau 4.2 : Définir le montage des composants

4.11 Programme de Arduino :

```
////////// début de déclaration des variables et des composants //////////  
int ledV = A1;  
int ledJ = A2;  
int ledR = A3;  
int irPin = 6;  
int irState = HIGH;  
int buzzer=A0;  
int BP=10;  
int a;  
int b;  
#include "HX711.h"  
int le=0;  
int M=0;  
int LO=0;  
HX711 scale(8, 7);  
float calibration_factor = -761; // ce facteur de calibration est ajusté en fonction de ma cellule de charge  
float units;  
float P;  
int s;  
int k1=0;  
  
#include <LiquidCrystal.h>  
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;  
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);  
  
#include <Servo.h>  
Servo myservoG;  
Servo myservoD;  
  
////////// fin de déclaration des variables et des composants //////////  
  
void setup() {  
  //////////// début de définition des entres et sortie et initialisation //////////  
  pinMode(ledV,OUTPUT);  
  pinMode(ledJ,OUTPUT);  
  pinMode(ledR,OUTPUT);  
  pinMode(BP,INPUT);  
  lcd.begin(16, 2);  
  Serial.begin(9600);  
  pinMode(irPin,INPUT);  
  
  scale.set_scale();  
}
```


Chapitre 4 : Présentation de la carte Arduino et Réalisation du Projet

```
scale.tare();

myservoG.attach(13);
myservoD.attach(9);
myservoD.write(180);
myservoG.write(175);
}

////////////////// début de définition des entrées et sorties //////////////////

void loop() {

////////////////// début d'éteindre les led //////////////////

    digitalWrite(ledV,LOW);
    digitalWrite(ledJ,LOW);
    digitalWrite(ledR,LOW);
    //////////////////// FIN d'éteindre les led //////////////////

    irState = digitalRead(irPin);
    scale.set_scale(calibration_factor);
    units = scale.get_units(), 10;
    if (units < 0)
    {
        units = 0.00;
    }

    P=units-floor(units);
    if (P>=0.5) {
        P=ceil(units); }
    if (P<0.5) {
        P=floor(units);}
    lcd.setCursor(0,0);
    lcd.print("POIDS = ");
    lcd.setCursor(6,0);
    lcd.print(P);
    lcd.setCursor(0,1);
    lcd.print("L=");
    lcd.setCursor(2,1);
    lcd.print(L0);
    lcd.setCursor(5,1);
    //////////////////// affichage du poids et l'incrementation les differents type des bouteilles en LCD //////////////////
    lcd.print("M=");
    lcd.setCursor(7,1);
    lcd.print(M);
    lcd.setCursor(10,1);
    // ...
}
```

Chapitre 4 : Présentation de la carte Arduino et Réalisation du Projet

```
lcd.setCursor(10,1);
lcd.print(" l=");
lcd.setCursor(13,1);
lcd.print(le);
s=P;
if (s!=0) {
  delay(500);
  units = scale.get_units(), 10;
  P=units-floor(units);
if(P>=0.5) {
  P=ceil(units); }
if(P<0.5) {
  P=floor(units);}
  if(P==s){          //////////////// comparaison de valeur mesuré actuelle et valeur avant 500 millisecondes ////////////////

if (P < 25 && P > 1){
  digitalWrite(LedR,HIGH);
  tone(buzzer,5500,400);
  myservoG.write(90);
  delay(400);      //////////////// pousser les bouteilles légères par le premier servomoteur vers le chemin de recyclage et buzzer faire une speciale tone

delay(400);
  //////////////// pousser les bouteilles légères par le premier servomoteur vers le chemin de recyclage et buzzer faire une speciale tone

  //////////////// et incrementation des bouteilles legère
  myservoG.write(175);
  delay(400);
  le=le+1;
}

if (P > 28){
  digitalWrite(LedR,HIGH);
  tone(buzzer,250,400);
  myservoG.write(90);
  delay(400);
  //////////////// pousser les bouteilles lourds par le premier servomoteur vers le chemin de recyclage et buzzer faire une speciale tone

  //////////////// et incrementation des bouteilles lourds

  myservoG.write(175);
  delay(400);
  lO=lO+1;
}
```

```
if (P >= 25 && P <= 27){
  digitalWrite(ledV,HIGH);
myservoD.write(0); /////////////// pousser les bouteilles moyenne par le deuxieme servomoteur vers la place des bouteilles

  /////////////// et incrementation des bouteilles moyenne

delay(400);
kl=1;
myservoD.write(180);
}
}
}

if(irState == LOW && kl==1){ /////////////// l'incrementation des bouteilles moyennes fait avec deux conditions : servomoteur doit marche et le capteur IR detecte un objet
  M=M+1;
  kl=0;
}
a=digitalRead(BP);
b=1;
Serial.println(a);
if(a!=b){
  digitalWrite(ledJ,HIGH);
  delay(400);
  lcd.clear();
  LO=0; /////////////// Bouton poussoire est pour réinitialiser le comptage des toute les bouteilles
  M=0;
  le=0;}
else {
  LO=LO;
  M=M;
  le=le;}
}
```

Figure 4.29 : le Programme de Arduino

4.12 Conclusion

Dans ce chapitre, nous avons vu une description de Arduino uno et ses caractéristiques et le logiciel IDE utilisé de Arduino, on a cité les différentes composant utilisés dans la réalisation A la fin, on a décrit le montage et le câblage pour réalisé ce projet .

Conclusion Générale

L'accomplissement de ce projet de fin d'études nous a permis de mettre en œuvre mes connaissances pratiques et théoriques acquises durant tout le stage.

Ce projet a permis d'étudier, de modifier et aussi de programmer un système de contrôle de poids des bouteilles en ligne de production, à base de l'automate programmable Siemens S7-300 et simulé dans FACTORY IO. Le système est réalisé avec Arduino Uno.

D'une manière générale, ce travail m'a permis d'augmenter et enrichir mes expériences pratiques dans le domaine d'électrotechnique, électronique, sécurisation et Automatisation de système. Surtout dans :

- La programmation des API.
- L'utilisation du logiciel TIA PORTAL et FACTORY IO
- La programmation de Arduino et ses composants

Néanmoins j'ai rencontré quelques obstacles pendant la réalisation de ce projet et qui sont :

- Afficheur de mon pc a été endommagé
- La difficulté de l'installation du logiciel TIA Portal V13 en Windows 11 , alors j'ai utilisé la machine virtuelle avec Windows 7.
- Le fonctionnement de la jauge de contrainte.

Enfin, nous estimons que ce mémoire peut être développé comme perspective, pour introduire d'autres technique de mesure comme placer un petit convoyeur au dessus de la jauge de contrainte pour être mesure et traité les bouteilles au même temps les bouteilles va prendre leur chemin.

Bibliographie

[1] <https://laboratoiresvenus.com>

Consulté le 2022/07/10

[2] RECORD, Etat de l'art concernant les grands procédés de mise en œuvre des matières plastiques issues de processus de régénération de déchets. Efficacité économique des filières de recyclage des plastiques, 2003, 175 p, n°00-0905/1A.

[3] Vénus S.A.P.E.C.O (s.d.). Page facebook Vénus S.A.P.E.C.O. En ligne :

https://www.facebook.com/pg/laboratoiresvenus/about/?ref=page_internal

Consulté le 2022/07/10

[4] N.BELKADA et S.BENOUALI ,<< amélioration et développement d'une solution de commande et supervision de la station d'ensachage de MIS à base d'un API S7- 300>> mémoire de fin d'études de Master professionnel, Spécialité : Automatique et Informatique industrielle, 2017/06/20

[5] L. Bergougnoux, API automate programmable industriel, Poly Tech Marseille, Support cours, 2004/2005.

[6] E. Ulamine, A. Hammouali, Conception de régulation de niveau avec un automate programmable, Mémoire de fin d'étude, Master Académique OPTION : ELECTRONIQUE DES SYSTÈMES EMBARQUÉS 2018/2019.

[7] B. K. Kangni, Introduction des automates programmables industriels sur les locomotives diesels électriques a l'O.T.P : incidences économiques et technique, Projet de fin d'étude, Université cheikh Anta Diop, 1992.

[8] L. Andjouh, R. Touati, Automatisation et supervision de la fosse de relevage de la raffinerie d'huile au niveau du complexe agroalimentaire Cevital, Projet de fin d'étude, Université de Bejaia, 2012/2013.

[9] Frank D. Petruzella, Programmable Logic Controllers, Fifth Edition, McGraw-Hill Education, 2016.

[10] Système d'automatisation S7-300 Installation et configuration : CPU 312 IFM - 318-2 DP Manuel de mise en œuvre Edition 06/2003

[11] Learn-/Training Textbook Siemens Automation Cooperates with Education (SCE), Siemens AG 2017.

[12] <https://factoryio.com/docs/>

Consulté le 16/07/2022

Bibliographie

- [13] logiciel du tia portal
- [14] C. Tavernier, « Arduino applications avancées ». Version Dunod
- [15] <http://www.generationrobots.com/fr/152-arduino>. Consulté le: mars 2015.
- [16] Jean- Noël, « livret Arduino en français » , centre de ressources art sensitif .
- [17] <https://fr.wikipedia.org/wiki/Servomoteur> Consulté le 2022/08/24
- [18] <https://www.cdiscount.com/bricolage/electricite/vs-elec-ir-infrarouge-capteur-evitement-obstacles/f-1661416-auc3665662005397.html#desc> Consulté le 2022/08/24
- [19] <https://learn.sparkfun.com/tutorials/getting-started-with-load-cells/load-cell-basics>
Consulté le 2022/08/24
- [20] <https://www.f-legrand.fr/scidoc/docmml/sciphys/arduino/deformation/deformation.html>
Consulté le 2022/08/24
- [21] Le Grand livre d'Arduino. ERIK Bartman 2ème édition
- [22] <http://ihm3d.fr/httpwww-ihm3d-frle-bouton-poussoir.html> Consulté le 2022/09/03
- [23] https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/743_gestion-des-entrees-sorties/3423_un-simple-bouton/#1-10762_quest-ce-quun-bouton Consulté le 2022/09/03
- [24] <https://www.centralcarolinascale.com/conveyor-scales.htm> Consulté le 2022/09/03
- [25] <https://www.bastiansolutions.com/solutions/technology/conveyor-systems/in-line-scales/> Consulté le 2022/09/03
- [26] [https://docs.factoryio.com/manual/parts/light-load/#pusher\]=](https://docs.factoryio.com/manual/parts/light-load/#pusher]=)
Consulté le 2022/09/06
- [27] <https://realpars.com/photoelectric-sensor/> Consulté le 2022/09/06

Annexe

