

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 DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Filière Électronique
Spécialité **Électronique des Systèmes Embarqués**

présenté par

KERROUM Ilyes
ELOUARET Ahmed

Réalisation d'une trieuse automatisée à base d'un automate **SIEMENS S7-1200**

Proposé par : Mr. FAS Mohamed Lamine

REMERCIEMENT

Avant tout, il ne saurait question d'entreprendre les remerciements sans avoir remercié le BON DIEU de nous avoir permis de réaliser ce travail.

Nous tenons à remercier tout d'abord notre encadreur Monsieur M.L.FAS, pour sa patience, et surtout pour sa confiance, ses remarques et ses conseils, sa disponibilité et sa bienveillance. Qu'il trouve ici le témoignage de notre profonde gratitude.

Nos remerciements vont aussi à l'ensemble de nos professeurs tout le long de ces années d'études.

Sans oublier de remercier tous les membres de nos familles respectives pour leurs soutient et leurs encouragements, particulièrement nos chères parents.

Enfin, à toutes les personnes ayant participé de près ou de loin à la réalisation de ce projet, un grand merci.

K. Ilyes

E. Ahmed

ملخص

تقدم هذه المذكرة دراسة وتنفيذ آلة لفرز التفاح وفق اللون وتتم العملية بدايةً من التقاط الصور المتتالية للعينات بواسطة Pi «Pi Camera» ليقوم اللوح البرمجي «Raspberry pi» بمعالجة الصور وفق خوارزمية مبرمجة بلغة «Python»، لتُبعث الأوامر بعدها إلى المبرمج الصناعي من نوع Siemens S7-1200 ليتحكم بنفسه في الناقل والدافعات كما يشرف على الآلة شاشة اللمس من نوع SAMKOON. أما عن تغذية الجزء العملي والجزء التحكمي بالطاقة اللازمة فقد تم صنع خزانة كهربائية مجهزة بجميع المعدات اللازمة مثل أجهزة السلامة والإشارات والحماية.

Résumé

Ce mémoire présente la conception et la réalisation d'une trieuse des pommes selon leurs couleurs. Le processus commence par la capture d'images successives d'échantillons par "Pi Camera" pour que "Raspberry pi" traite ces images selon un algorithme programmé par le langage de "Python". Puis les informations sont envoyées au API de type « Siemens S7-1200 » pour commander la partie opérative. Pour superviser la machine en temps réel on a utilisé un écran tactile IHM de type « SAMKOON ». Quant à l'alimentation du partie opérative et du partie commande avec l'énergie nécessaire, une armoire électrique a été équipée par tous les matériels nécessaires comme les appareils de sécurité et protection.

Abstract

The paper presents the study and implementation of a machine to sort apples according to color. The process begins with the capture of successive images of samples by "Pi Camera" so that "Raspberry pi" processes the images according to an algorithm programmed by the "Python" language. Then the commands sent to the PLC "Siemens S7-1200" type to control the belt and the actuators, as well as to supervise the machine by the "SAMKOON" HMI.

As for supplying the operating part and the control part with the necessary energy, an electrical cabinet has been made equipped with all the necessary equipment such as safety devices and protection.

Listes des acronymes et abréviations

DC : Courant continue.

API : Automate programmable industriel.

PLC : Programmable Logic Controller.

CPU : Unité centrale de traitement.

f: fréquence du courant d'alimentation (HZ).

AC : Courant alternative.

PC : Partie Commande

PO : Partie Opérative.

IHM : Interface Homme-Machine.

TIA : Totally Integrated Automation.

ROM : Mémoire morte.

HSV : Hue Saturation Value.

RVB : Rouge Vert Blue.

GPIO : General Purpose Input/Output

E/S : Entrées Sorties.

BO : Block d'organisation.

DSP : Digital Signal Processor.

CPU : Central Processing Unit.

Liste des figures

CHAPITRE I

Figure I. 1 Structure d'un système automatisé.....	3
Figure I. 2 : L'automate programmable S7-1200 et ces modules.....	7
Figure I. 3: La CPU S7-1214C (DC/DC/RLY).....	9
Figure I. 4: L'état des modes fonctionnement de S7-1200.....	9
Figure I. 5 : Modules et mémoire d'API S7-1200.....	10
Figure I. 6 : Les connecteurs d'un Raspberry.....	11
Figure I. 7 : Connecteur RCA.....	14
Figure I. 8 : Connecteur DSI.....	15
Figure I. 9: Les fichiers de NOOBS.....	17
Figure I. 10: Interface graphique de Raspberry.....	20
Figure I. 11 : Interface Homme-Machine.....	24
Figure I. 12 : Face avant de SK-102HS.....	25

CHAPITRE II

Figure II 1: Principe de fonctionnement du capteur.....	28
Figure II 2 : Symbole du capteur photoélectrique de 3fil.....	29
Figure II 3 : Différents types de détection d'un capteur photoélectrique.....	30
Figure II 4 : Capteur photoélectrique, détection par barrage.....	30
Figure II 5 : Capteur photoélectrique, détection par cellules reflex.....	30
Figure II 6 : Capteur photoélectrique, détection par cellules reflex.....	31
Figure II 7 : Le distributeur (Schéma de principe).....	32

Figure II 8 : Relais électrique 5V DC.....	33
Figure II 9 : Symbole de relais 5V DC.....	33
Figure II 10 : Image et symbole d'un contacteur.....	34
Figure II 11 : Distributeur 5/2 bistable commande électrique.....	35
Figure II 12 : Le symbole d'un distributeur 2/2 commande poussoir.....	35
Figure II 13 : Les différents symboles d'un distributeur.....	36
Figure II 14 : Constituants de base d'un vérin.....	38
Figure II 15 : Vérins simple effet.....	38
Figure II 16 : Vérins double effet.....	38
Figure II 17 : Symboles des Vérins simple et double effet.....	39
Figure II 18 : Composants d'un moteur monophasé.....	40
Figure II 19 : Schéma de principe de fonctionnement d'un moteur monophasé.....	40
Figure II 20 : Schéma de principe d'un moteur à condensateur permanent.....	42

CHAPITRE III

Figure III. 1 : Caractéristiques des contacteurs Schneider LC1D25.....	44
Figure III. 2 : Symbole d'un contacteur.....	44
Figure III. 3 : Caractéristiques du contact auxiliaires Schneider LADN40.....	45
Figure III. 4 : Caractéristiques électriques du disjoncteur magnétothermique utilisé.....	45
Figure III. 5 : Caractéristiques électriques du sectionneur porte fusible.....	45
Figure III. 6 : Image et caractéristiques du fusible utilisé.....	46
Figure III. 7 : 13.....	46

Figure III. 8 : Schéma synoptique d'une alimentation 24V DC.....	46
Figure III. 9 : Image et caractéristiques électriques du bouton d'arrêt d'urgen..	47
Figure III. 10 : Image et caractéristiques électriques du bouton poussoir.....	47
Figure III. 11 : Image et caractéristiques électriques du commutateur.....	47
Figure III. 12 : Image et caractéristiques électriques des voyants.....	48
Figure III. 13 : Image et caractéristiques électriques du buzzer.....	48
Figure III. 14 : Image et caractéristiques de la borne électrique.....	48
Figure III. 15 : LM2596.....	48
Figure III. 16 : Principe de fonctionnement d'un convertisseur Buck.....	49
Figure III. 17 : Schéma de base d'un convertisseur Buck.....	49
Figure III. 18 : Lumière LED.....	50

CHAPITRE IV

Figure IV. 1 : Schéma structurel de la machine trieuse.....	54
Figure IV. 2 : cylindre de HSV.....	55
Figure IV. 3 : Gamme de teinte	56
Figure IV. 4 : Organigramme du traitement d'image.....	57
Figure IV. 5 : Composants de la chambre d'imagerie.....	58
Figure IV. 6 : Image HSV du pomme jaune.....	59
Figure IV. 7 : Image RVB du pomme jaune.....	59
Figure IV. 8 : Image RVB du pomme rouge.....	59
Figure IV. 9 : Image HSV du pomme rouge.....	59
Figure IV. 10 : Masque d'image du pomme rouge.....	60

Figure IV. 11 : Masque d'image du pomme jaune.....	60
Figure IV. 12 : Masque + HSV image du pomme rouge.....	60
Figure IV. 13 : Masque + HSV image du pomme jaune.....	60
Figure IV. 14 : Les résultats de la valeur moyenne de la couleur jaune du pomme ..	61
Figure IV. 15 : Vue de dessus du convoyeur et de la chambre d'imagerie.....	62
Figure IV. 16 : Convertisseur Buck et la liaison du carte Raspberry Pi aux relais	62
Figure IV. 17 : L'armoire électrique et IHM	63
Figure IV. 18 : Table des variables E/S.....	64
Figure IV. 19 : Réseau pour mise en marche avec auto-maintien.....	65
Figure IV. 20 : Réseau pour la commande des trois vitesses.....	65
Figure IV. 21 : Réseau d'arrêt d'urgence.....	65
Figure IV. 22 : Réseau pour sortir la tige du vérin 1.....	66
Figure IV. 23 : Réseau temporisation de la tige sortant du vérin 1	66
Figure IV. 24 : Réseau pour rentrer la tige du vérin1	66
Figure IV. 25 : Réseau pour sortir la tige du vérin 2.....	67
Figure IV. 26 : représente la temporisation de la tige sortant du vérin 2	67
Figure IV. 27 : Réseau pour rentrer la tige du vérin 2.....	67
Figure IV. 28 : Réseau pour gérer le manque de produit	68
Figure IV. 29 : Réseau pour comptage les pommes jaunes.....	68
Figure IV. 30 : Réseau pour comptage les pommes rouges.....	69
Figure IV. 31 : l'interface de logiciel SKTOOL.....	70
Figure IV. 32 : La première fenêtre de l'IHM.....	71
Figure IV. 33 : La deuxième fenêtre de l'IHM.....	72

Liste des tableaux

CHAPITRE I

Tableau I.1 : Tableau de comparaison des CPU S7-1200.....	8
Tableau I.2 : Comparatif des modèles de Raspberry PI.....	13
Tableau I.3 : Tableau de caractéristique de SK-102HS.....	26

Sommaire

Introduction générale	1
CHAPITRE I : L'automate programmable et l'interface Homme-Machine	2
I.1. Introduction.....	3
I.2. Définition d'un système automatisé.....	3
I.3. Les buts d'un système automatisé.....	3
I.4. Structure d'un système automatisé.....	4
I.4.1. Partie opérative.....	4
I.4.2. Partie commande.....	4
I.4.3. Présentation du Raspberry pi.....	10
I.5. Partie de supervision.....	23
I.5.1. Définition de la supervision.....	23
I.5.2. Avantages de la supervision.....	23
I.5.3. Interface Homme-Machine.....	24
I.5.4. IHM SAMKOON SK-102HS.....	24
I.6. Conclusion.....	26
CHAPITRE II : Les actionneurs pneumatiques et électriques	27
II.1. Introduction.....	28
II.2. Les capteurs.....	28
II.2.1. Définition	28
II.3. Les pré-actionneur.....	31
II.3.1. Définition	31
II.3.2. Fonctionnement.....	31
II.3.3. Classification des pré-actionneurs.....	32
II.4. Les actionneurs hydrauliques et pneumatiques.....	36
II.4.1. Les actionneurs hydrauliques.....	36
II.4.2. Les actionneurs pneumatiques.....	36
II.4.3. Les vérins	37
II.5. Les actionneurs électriques.....	39
II.5.1. Le moteur asynchrone monophasé.....	39

II.6. Conclusion.....	42
CHAPITRE III : Les appareillages électriques	43
III.1. Introduction.....	44
III.2. Description des différents appareillages électriques utilisés.....	44
III.2.1. Les contacteurs.....	44
III.2.2. Les accessoires complémentaires.....	45
III.2.3. Appareils de protection	46
III.2.4. Alimentation continu de 24V DC.....	46
III.2.5. Bouton d'arrêt d'urgence.....	47
III.2.6. Les boutons poussoirs.....	47
III.2.7. Le commutateur.....	47
III.2.8. Les appareils de signalisation.....	48
III.2.9. Borne électrique.....	48
III.2.10. Convertisseur Buck réglable (LM2596).....	48
III.2.11. Lumière LED	50
III.3. Conclusion.....	51
CHAPITRE IV : Réalisation de la machine de tri (une trieuse)	52
IV.1. Introduction.....	53
IV.2. Cahier de charge.....	53
IV.3. Schéma fonctionnel du système.....	54
IV.4. Traitement d'image.....	55
IV.4.1. Introduction à l'espace des couleurs (conversion de RVB en HSV).....	55
IV.4.2. Organigramme du traitement d'image	56
IV.4.3. Méthodes et matériaux de traitement.....	58
IV.4.4. Convertir l'image capturée.....	59
IV.5. Partie de programmation.....	64
IV.5.1. Programmation d'API par TIA Portal	64
IV.5.2. Programmation de l'IHM.....	69
Conclusion générale	73
Référence bibliographique.....	74
Annexe.....	75

Introduction générale

Au cours des dernières décennies, les procédés industriels ont connu un essor vertigineux, grâce à l'automatisme. Ainsi, la mondialisation des économies et l'accentuation de la concurrence poussent davantage les entreprises à automatiser leurs processus de production, afin d'assurer leur pérennité en s'offrant la compétitivité, tout en améliorant les conditions de travail de leur personnel ; en supprimant les tâches pénibles et répétitives.

Les automates programmables industriels apportent la solution sur mesure pour les besoins d'adaptation et de flexibilité de nombreuses activités économiques actuelles. Ils sont devenus aujourd'hui les constituants les plus répandus des installations automatisées.

La couleur est la caractéristique la plus importante pour une classification et un tri précis des fruits et légumes. En raison du besoin toujours croissant de fournir des fruits et légumes de haute qualité dans un court laps de temps, le classement automatisé des produits agricoles devient une priorité particulière parmi de nombreuses associations d'agriculteurs. L'impulsion de ces tendances peut être attribuée à une prise de conscience accrue par les consommateurs de leur meilleur bien-être sanitaire et à une réponse des producteurs à la nécessité de fournir des produits de qualité garantie avec cohérence. C'est dans ce contexte qu'intervient le domaine de l'inspection automatique et de la vision par ordinateur pour jouer le rôle important de contrôle qualité des produits agricoles.

L'objectif principal de notre travail est de réaliser une machine de tri (une trieuse) des pommes selon leurs couleurs commandée par un automate programmable industriel de type S7-1200 et supervisée par un écran tactile afin d'avoir un suivi en temps réel.

Pour atteindre notre objectif, nous avons organisé notre mémoire en quatre chapitres :

Le premier chapitre aborde les systèmes automatisés, comme « API siemens S7-1200 », « Raspberry Pi 2 », les modules communication et d'alimentation. Dans la suite de ce chapitre on présente l'écran tactile utilisé (IHM) de marque SAMKOON SK-102HS.

Le deuxième chapitre consacré sur la partie opérative notamment les capteurs les pré-actionneurs, les actionneurs pneumatiques et électriques.

Le troisième chapitre sera dédié la partie de puissance (l'armoire électrique) et leurs éléments essentiels tel que les éléments de protections, systèmes de réglettes et de fixations, Un arrêt d'urgence ...etc.

Le quatrième chapitre, qui est la partie réalisation de notre projet. Cette réalisation est basée sur la commande d'une machine de tri commandée par un API et supervisé avec un IHM.

Enfin, nous clôturons notre document avec une conclusion générale qui est dédiée à une synthèse du travail présenté.

CHAPITRE I

L'automate programmable et l'interface Homme-Machine

I.1. Introduction

En tant qu'automaticien on a tendance à avoir une vue systémique sur les machines et les processus. Notre objectif consiste à automatiser et à asservir ces systèmes afin d'accroître la productivité, améliorer la qualité et la sécurité. Le plus important c'est de remplacer l'homme dans ces actions pénibles, délicates et répétitives.

Dans ce chapitre, nous allons définir les systèmes automatisés et les parties qui constituent ces derniers tels que les API, l'Interface Homme-Machine (IHM) et le microcontrôleur Raspberry.

I.2. Définition d'un système automatisé

Un système automatisé ou automatique est un système réalisant des opérations pour lequel l'homme n'intervient que dans la programmation du système et dans son réglage.

Un **système automatisé** est toujours composé d'une **Partie Commande** et d'une **Partie Opérative**. Pour faire fonctionner ce système, l'**opérateur** (personne qui va faire fonctionner le système) va donner des consignes à la **Partie Commande**. Celle-ci va traduire ces consignes en ordres qui vont être exécutés par la **Partie Opérative**. Une fois les ordres accomplis, la **Partie Opérative** va le signaler à la **Partie Commande** (elle fait un compte-rendu) qui va à son tour le signaler à l'**Opérateur**. Ce dernier pourra donc dire que le travail a bien été réalisé.

I.3. Les buts d'un système automatisé

Il y a plusieurs avantages de système automatisé, on peut citer :

- ✓ Réaliser des tâches complexes ou dangereuses pour l'homme
- ✓ Effectuer des tâches pénibles ou répétitives
- ✓ Gagner en efficacité et en précision.

I.4. Structure d'un système automatisé

I.4.1. Partie opérative

Cette partie exécute les ordres reçus de la partie commande, elle transforme les signaux de commande en énergie électrique, pneumatique ou hydraulique pour réaliser le fonctionnement du système. En même temps, elle transmet l'état du système à la partie commande à travers les capteurs. Dans le chapitre suivant on parle plus en détail de la partie opérative surtout sur les pré-actionneurs, les actionneurs pneumatique et électrique et les capteurs de proximités.

I.4.2. Partie commande

C'est la partie qui gère le fonctionnement du système automatisé. Elle est en général composée d'un ordinateur qui contient dans ses mémoires un programme. Elle transmet les ordres aux actionneurs de la partie opérative à partir :

- Du programme qu'elle contient,
- Des informations reçues par les capteurs,
- Des consignes données par l'utilisateur ou par l'opérateur [8].

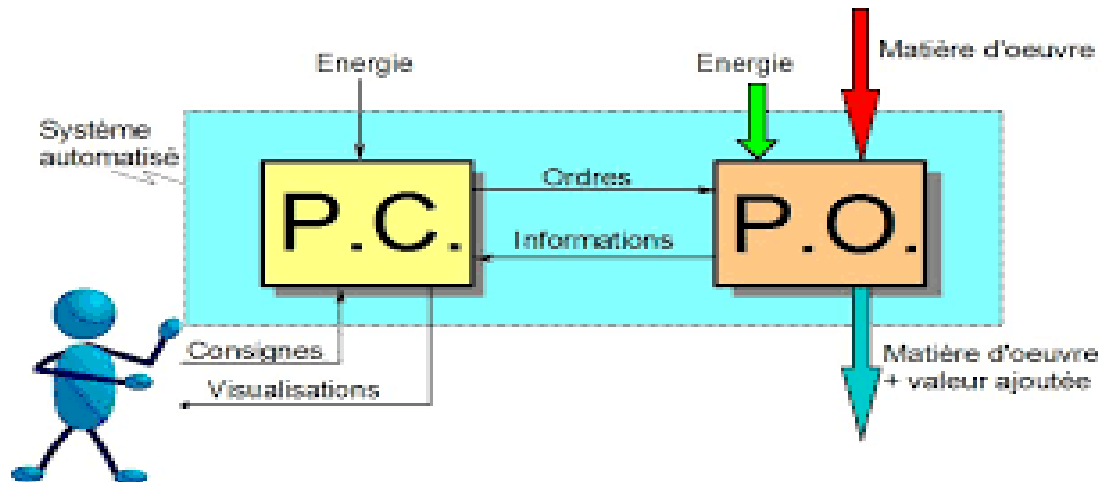


Figure I. 1 Structure d'un système automatisé

A. Unité de traitement d'informations

L'unité de traitement d'informations est un ensemble d'appareils (électriques, mécaniques ou biologiques) permettant de traiter automatiquement des informations. Cette opération peut être gérée par un **API**, une carte **Arduino** ou **DSP**... Dans notre système nous avons utilisé un **API** et une carte **RASPERRY** comme unité de traitement d'informations.

B. Automate Programmable Industriel

L'Automate Programmable Industriel (API) est un appareil électronique programmable, adapté à l'environnement industriel, qui réalise des fonctions d'automatisme pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique [3].

B.1. Description des éléments d'un API

➤ L'alimentation

L'unité d'alimentation est indispensable puisqu'elle converti une tension alternative (220V ou 110V) en une base tension continue (24V, 5V...) nécessaire au processeur et aux modules d'entrées sorties.

➤ **Unité centrale (Processeur)**

L'unité centrale est le regroupement du processeur et de la mémoire centrale. Elle commande l'interprétation et l'exécution des instructions programme. Les instructions sont effectuées les unes après les autres, séquencées par une horloge.

Exemple: Si deux actions doivent être simultanées, l'API les traite successivement.

➤ **Interfaces d'entrées / sorties**

L'interface d'entrée comporte des adresses d'entrée. Chaque capteur est relié à une de ces adresses. L'interface de sortie comporte de la même façon des adresses de sortie. Chaque pré-actionneur est relié à une de ces adresses. Le nombre de ces entrées et sorties varie en fonction du type d'automate. Les cartes d'entrées et de sorties sont modulaires, la modularité varie entre 8, 16 et 32 voies.

- ☑ **Les entrées TOR :** L'information ne peut prendre que deux états (Vrai/Faux, 0 ou 1). Elles permettent de raccorder à l'automate les différents capteurs logiques tels que : boutons poussoirs, thermostats, fins de course, capteur de proximité....
- ☑ **Les sorties TOR :** Elles permettent de raccorder à l'automate les différents pré-actionneurs tels que, électrovannes, contacteurs, voyants.
- ☑ **Les entrées numériques :** Utilisées pour les API haute ou moyenne gamme effectuant des traitements numériques. La longueur définit par la taille du mot mémoire de l'API (ex : 16 bits).
- ☑ **Les Entrées/Sorties analogiques :** Transforment une grandeur analogique en une valeur numérique et vice versa. La précision dépend du nombre de bits utilisés. Technologiquement, les EA/SA sont caractérisées par l'amplitude du signal analogique (typiquement 0/10V ou -10/+10V) et par le courant correspondant.

➤ **La mémoire**

Elle est conçue pour recevoir, gérer, stocker des informations issues des différents secteurs du système, qui sont le terminal de programmation (PC ou console) et le processeur. Elle reçoit également des informations en provenance des capteurs. Il existe dans les automates deux types de mémoires qui remplissent des fonctions différentes :

- **La mémoire Langage** où est stocké le langage de programmation. Elle est en général en lecture seulement (ROM : mémoire morte).

- **La mémoire Travail** utilisable en lecture-écriture pendant le fonctionnement, c'est la RAM (mémoire vive). Elle s'efface automatiquement à l'arrêt de l'automate (nécessite une batterie de sauvegarde).

B.2. Programmation d'API

La programmation d'un API consiste à traduire dans le langage spécialisé de l'automate, les équations de fonctionnement du système à automatiser. Parmi les langages normalisés, on cite quelques-uns des plus connus et plus utilisés :

- Langage à contacts (LADDER) ;
- Langage List d'instructions (Instruction List) ;
- Langage GRAFCET (SequentialFunctionChart : SFC) ;
- Langage littéral structuré.

Généralement, les constructeurs d'API proposent des environnements logiciels graphiques pour la programmation [17].

B.3. Critères de choix de l'automate

Il revient à nous d'établir le cahier des charges de notre système et de chercher sur le marché l'automate le mieux adapté à nos besoins. Cela est fait en considérant un certain nombre de critères importants :

- ✓ Avoir les compétences et l'expérience nécessaire pour programmer la gamme d'automate.
- ✓ Le nombre et le type d'entrées et de sorties nécessaires.
- ✓ La communication envisagée avec les autres systèmes.
- ✓ Les capacités de traitement de la CPU.
- ✓ Les moyens de sauvegarde.
- ✓ La fiabilité et la robustesse.
- ✓ Le coût d'investissement, de fonctionnement, de maintenance de l'équipement.
- ✓ La qualité du service après-vente.

En tenant compte des points mentionnés ci-dessus, et pour résoudre les problèmes rencontrés, on choisit l'automate **S7-1200** qui est la nouvelle gamme de **SIEMENS**.

C. Présentation de l'automate S7-1200

L'API **S7-1200** offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation. Sa forme *compacte*, sa

configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées.

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant.

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les réseaux PROFIBUS, GPRS, RS485 ou RS232.[6]



Figure I. 2 :L'automate programmable S7-1200 et ces modules

C.1. Principe de fonctionnement du S7-1200

Une fois le programme chargé, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans l'application. Il surveille les entrées et modifie les sorties conformément à la logique du programme utilisateur, qui peut contenir des instructions booléennes, de comptage, de temporisation, ou mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents [17],[18].

C.2. Technologie intégrée de S7-1200

- **Entrées rapides :** Le nouvel automate SIMATIC S7-1200 comporte jusqu'à six compteurs High-Speed. Trois entrées à 100 kHz et trois autres entrées à 30 kHz, sont intégrées en continu pour des fonctions de comptage et de mesure.
- **Sorties rapides**
Deux sorties rapides pour des trains d'impulsions de 100kHz sont également intégrées et permettent de piloter la vitesse et la position d'un moteur pas à pas ou d'un actionneur.

C.3.Choix de la CPU

Les CPU du système SIMATIC S7-1200 se déclinent en cinq classes de performances : CPU1211C, CPU1212C et CPU1214C,CPU1215C et CPU 1217C, chacune d'elles pouvant être étendue en fonction des besoins de la station. Les CPU's de l'automate S7-1200 sont données dans le tableau suivant [18]:

Tableau 1CPU	CPU 1211C	CPU 1212C	CPU1214C	CPU1215C	CPU1217C
Mémoire de travail	50 ko	75 ko	100 ko	125 ko	150 ko
E/S TOR	6 entrées 4 sorties	8 entrées 6 sorties	14 entrées 10 sorties	14 entrées 10 sorties	14 entrées 10 sorties
E/S analogiques	2 entrées	2 entrées	2 entrées	2 entrées 2 sorties	2 entrées 2 sorties
Modules E/S extensible	Aucune	2 modules	8 modules	8 modules	8 modules
Module de communication (CM)	3 modules	3 modules	3 modules	3 modules	3 modules
Port de communication Ethernet PROFINET	1 porte	1 porte	1 porte	2 portes	2 portes

Tableau I.1 : Tableau de comparaison des CPU S7-1200

Après avoir étudié notre système quand doit réaliser et après la comparaison entre les CPU disponibles, on a choisi la CPU 1214C DC/DC/Rely de référence **6ES7212-1HE40-0XB0** qui réponde à nous besoins (Figure 3.3).



Figure I. 3: La CPU S7-1214C (DC/DC/RLY)

C.4. Modes de fonctionnement de la CPU :

La CPU a les modes de fonctionnement suivants :

- En mode « **STOP** », la CPU n'exécute pas le programme, et on ne peut pas charger un projet.
- En mode « **STARTUP** », la CPU entame une procédure de démarrage.
- En mode « **RUN** », le programme est exécuté de façon cyclique, certaines parties d'un projet peuvent être chargées dans la CPU en mode RUN [17],[18].



Figure I. 4: L'état des modes fonctionnement de S7-1200

- ◆ Une lumière JAUNE indique le mode STOP.
- ◆ Une lumière VERTE indique le mode RUN.
- ◆ Une lumière CLIGNOTANTE indique le mode STARTUP.

C.5. Les différents modules d'API S7-1200

Le SIMATIC S7-1200 est un automate modulaire et tout un éventail de modules l'accompagnent. Les voici [18]:

1. Modules centraux CPU
2. Module de puissance PM (Power Module)
3. Signal Boards SB
4. Modules de signal SM (Signal Module)
5. Modules de communication CM (Communication Module)
6. Les cartes mémoire

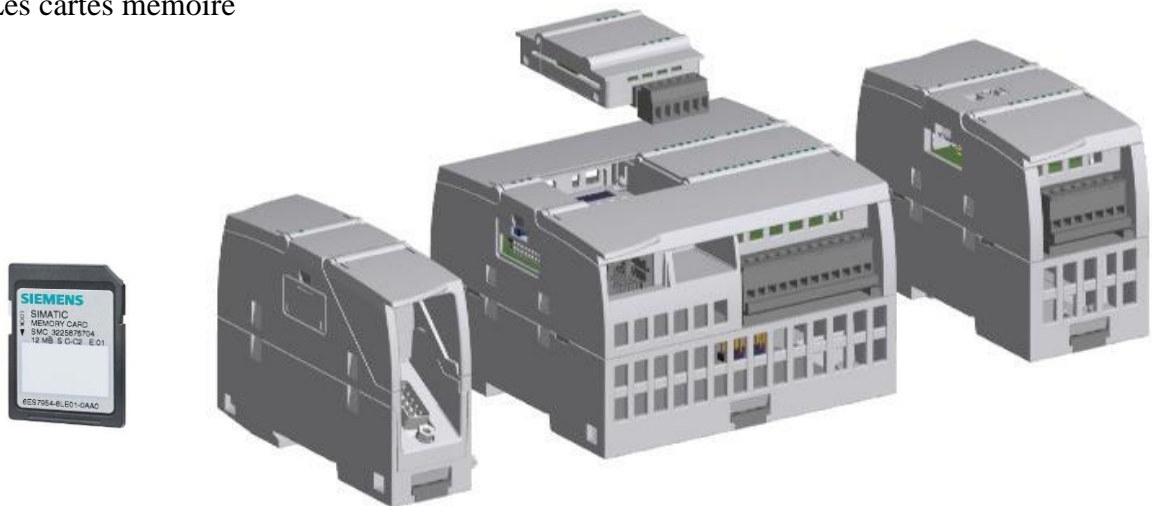


Figure I. 5 : Modules et mémoire d'API S7-1200

I.4.3. Présentation du Raspberry pi

Le Raspberry Pi est une unité centrale d'ordinateur mesurant 56 x 85 mm, environ les dimensions d'une carte de crédit. Ses fonctionnalités n'en restent pas moins impressionnantes, comme vous le découvrirez dans cette activité. Un petit retour sur l'histoire du Raspberry Pi (surnommé RasPi) et sur sa cible du marché initiale permet d'en comprendre les contraintes et les limitations. Le concept de RasPi a été dévoilé autour de 2006 par EbenUpton et ses collègues de laboratoire d'informatique de l'université de Cambridge, en Angleterre. Ils étaient préoccupés par la baisse de niveau de connaissances et de compétences des nouveaux étudiants en informatique en comparaison de leurs aînés. EbenUpton a alors décidé de créer un ordinateur à faible prix, car il supposait que les parents craignaient de laisser leurs enfants jouer avec des PC modernes relativement coûteux. C'est de cette idée qu'est parti le développement du RasPi très bon marché. Cet ordinateur devait donner aux jeunes l'opportunité d'apprendre et de pratiquer la programmation, sans que leurs parents ne s'inquiètent des dégâts éventuels sur la machine. EbenUpton a réuni plusieurs personnes pour constituer la fondation Raspberry Pi. Cette association caritative, enregistrée au Royaume-Uni, avait pour objectif de promouvoir l'esprit

informatique et l'intérêt pour cette discipline, en particulier chez les plus jeunes, en utilisant le RasPi comme plateforme de départ. Elle semble avoir atteint ce but louable, puisque les ventes du RasPi initialement estimées à 10 000 unités ont été largement dépassées ; elles approchent aujourd'hui le million d'exemplaires. Sur le site de la fondation www.raspberrypi.org, vous trouverez de nombreuses informations sur la carte, son actualité, des forums, des FAQs, etc. Pour arriver à un ordinateur bon marché, la décision capitale a été d'articuler sa conception autour d'une puce de type SoC (System on a Chip). Dans une telle puce, la mémoire, le microprocesseur et le processeur graphique sont physiquement placés sur la même « galette » de silicium, ce qui permet de réduire la taille du circuit imprimé (PCB : Printed Circuit Board) et le nombre de piste de connexion. La fondation a mis en place un partenariat avec Broadcom de façon à pouvoir exploiter ses schémas de conception, que ce soit pour le microprocesseur ou pour le processeur graphique du SoC. La puce et les autres composants à connaître, comme les connecteurs sont identifiés sur la figure ci-dessous.

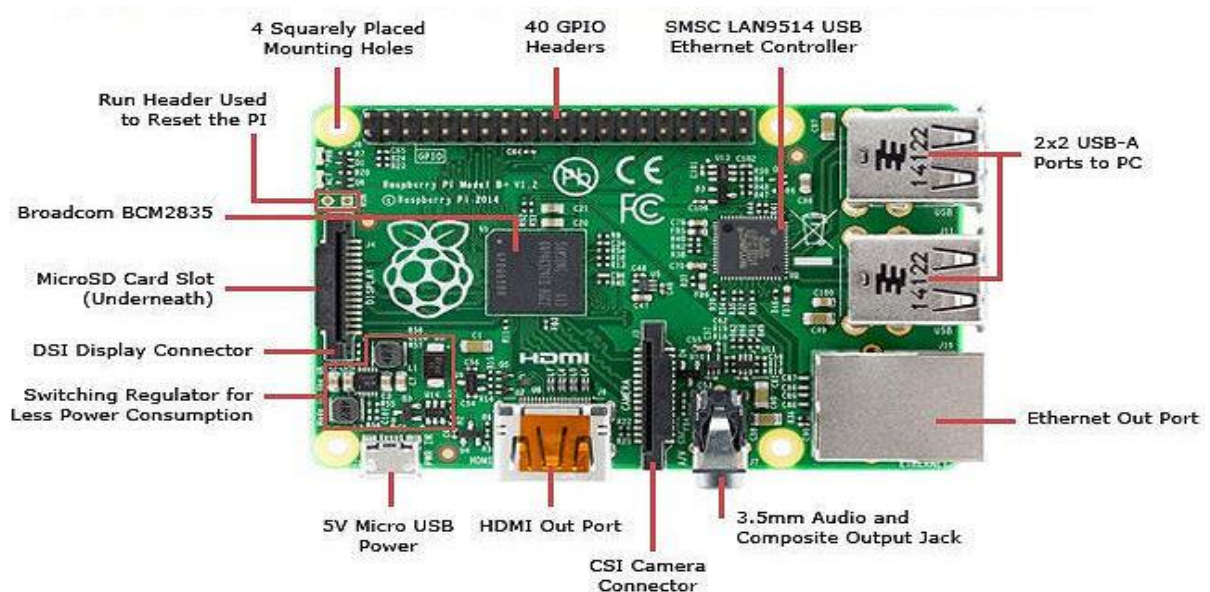


Figure I. 6: Les connecteurs d'un Raspberry

A. Que peut-on faire avec Le Raspberry Pi ?

Un certain nombre de choses que vous pouvez faire avec votre Raspberry Pi prendre le contrôle du matériel via Python, vous en servir comme media center, préparer des projets faisant appel à de la vidéo, ou concevoir des jeux en langage Scratch. La beauté de la chose, c'est que vous le faites avec un ordinateur, certes très petit, mais capable de faire les mêmes choses qu'un

ordinateur standard (il sera peut-être un peu plus lent pour certaines applications de bureau, mais bien meilleur pour d'autres). De plus, les bonnes capacités multimédia et 3D du Raspberry Pi permettent de le transformer en plateforme de jeux.

B. À La découverte du Raspberry Pi

La carte Raspberry Pi est une merveille de miniaturisation, qui intègre une puissance de calcul considérable sur une surface pas plus grande qu'une carte de crédit. Le Pi est capable de réaliser des prouesses, mais avant de vous aventurer sur cette terre inconnue, il est préférable de commencer par apprendre quelques petites choses

C. Tour du Propriétaire de la carte Raspberry Pi

Depuis son lancement où il n'y avait que deux modèles (les modèles A et B qui ne sont plus commercialisés aujourd'hui), la famille Raspberry Pi s'est considérablement agrandie. La gamme actuelle se compose de cinq modèles principaux le Raspberry Pi modèle A+, le Raspberry Pi modèle B+, le Raspberry Pi 2, le Rasp-berry Pi 3 et le Raspberry Pi Zéro. Mis à part le Raspberry Pi Zéro, qui est un modèle d'entrée de gamme, spécialement conçu pour être le moins cher possible et avoir une taille minimale, tous les modèles partagent une conception à peu près similaire qui ne se différencie que par des caractéristiques comme le nombre de ports USB, la présence ou l'absence de ports réseau et la puissance de leur processeur. La gamme compte également un sixième modèle qui est moins courant le Raspberry Pi Computer Module. Conçu pour un usage industriel sur des cartes personnalisées, le Computer Module s'exécute sous le même logiciel que les autres cartes de la famille, mais son étude dépasse le cadre de cet ouvrage.

	 Raspberry Pi 4 B 8GB	 Raspberry Pi 4 B	 Raspberry Pi 3 Model A+	 Raspberry Pi 3 B+	 Raspberry Pi Zero WH	 Raspberry Pi Zero W
Date de sortie	28 mai 2020	24 juin 2019	15 nov. 2018	14 mars 2018	12 janv. 2018	28 fév. 2017
Description	Version du Raspberry Pi 4 B avec 8Go de mémoire et une modification de l'agencement des composants pour l'alimentation électrique.				connecteur GPIO déjà soudé	
Caractéristiques						
Prix	76,00 \$US	37,95 €	26,00 \$US	39,09 €	14,00 €	10,44 €
SOC						
SOC Type	Broadcom BCM2711	Broadcom BCM2711	Broadcom BCM2837B0	Broadcom BCM2837B0	Broadcom BCM2835	Broadcom BCM2835
Core Type	Cortex-A72 (ARM v8) 64-bit	Cortex-A72 (ARM v8) 64-bit	Cortex-A53 64-bit	Cortex-A53 64-bit	ARM11762Z-S	ARM11762Z-S
Nombre de coeurs	4	4	4	4	1	1
GPU	VideoCore VI	VideoCore VI	VideoCore IV	VideoCore IV	VideoCore IV	VideoCore IV
Fréquence processeur	1,5 GHz	1,5 GHz	1,4 GHz	1,4 GHz	1 GHz	1 GHz
Mémoire vive	8 GB LPDDR4	1 GB, 2 GB, 4 GB LPDDR4	512 MB DDR2	1 GB DDR2	512 MB	512 MB
Wired Connectivity (On-Board)						
USB	2x USB3.0 + 2x USB2.0	2x USB3.0 + 2x USB2.0	1x USB 2.0	4x USB2.0	micro & micro OTG	1 x micro OTG
Ethernet	Gigabit	Gigabit		Gigabit - Over USB 2.0	module supp (14,99€)	module supp (14,99€)
SATA Ports						
Port HDMI	2x micro HDMI	2x micro HDMI		HDMI	mini	mini
Analog Video Out	shared with audio jack	shared with audio jack	shared with audio jack	shared with audio jack	via unpopulated pin	via unpopulated pin
Analog Audio Out	3.5mm jack	3.5mm jack	3.5mm jack	3.5mm jack	HDMI audio	HDMI audio
SPI						
I2C						
GPIO						
LCD Panel						
Camera						
SD/MMC	microSD	microSD	microSD	microSD	microSD	microSD
Serial				RV/TX UART		
Wireless Connectivity (On-Board)						
Wi-Fi	2,4GHz and 5GHz 802.11 b/g/n/ac	2,4GHz and 5GHz 802.11 b/g/n/ac	2,4GHz and 5GHz 802.11 b/g/n/ac	2,4GHz and 5GHz 802.11 b/g/n/ac	802.11n	802.11n
Bluetooth®	5.0	5.0	4.2, BLE	4.2, BLE	4.1	4.1
Dimensions						
Hauteur	85,6 mm	85,6 mm	65 mm	86,6 mm	30 mm	30 mm
Largeur	56,6 mm	56,6 mm	56 mm	56,6 mm	66 mm	66 mm
Profondeur	11 mm	11 mm	11 mm	17 mm	13 mm	5 mm
Poids	46 g	46 g	29 g	46 g	12 g	9 g
Power						
Power ratings	1,25 A @5V	1,25 A @5V		1,13 A @5V	180 mA	180 mA
Power sources	USB-C	USB-C	microUSB GPIO	microUSB, GPIO	microUSB or GPIO	microUSB, GPIO
Power Over Ethernet	with PoE Hat	with PoE Hat		with PoE Hat		

Tableau I.2 : Comparatif des modèles de Raspberry Pi

D. Mise en route du Raspberry pi

Pour utiliser votre Pi, vous aurez besoin de certains périphériques supplémentaires. Un écran vous permettra de voir ce que vous faites, un clavier et une souris constituant vos périphériques de saisie. Dans ce chapitre, vous découvrirez comment connecter ces périphériques au Pi, et

apprendrez à établir une connexion réseau si vous avez un modèle B, B+, Pi 2, Pi 3 ou Pi 4. Vous apprendrez également à télécharger et à installer un système d'exploitation pour le Pi.

A. Connexion de l'écran

Avant de pouvoir commencer à utiliser votre Raspberry Pi, vous devez connecter un écran. Le PI prend en charge trois sorties vidéo différentes la vidéo composite, la vidéo sur le port HDMI et la vidéo sur le port DSI. La vidéo composite et la vidéo sur le port HDMI sont faciles à mettre en œuvre et sont décrites dans cette section, mais la vidéo sur le port DSI requiert certains matériels spécialisés, comme un écran tactile

A.1. Vidéo Composite

La vidéo composite, auparavant disponible via le port jaune et argent, situé en haut des anciens modèles de Pi, et connu sous le nom de connecteur RCA, se trouve aujourd'hui en bas de la plupart des cartes sous la forme d'une prise jack 3,5 mm audiovisuelle. La sortie vidéo composite est conçue pour connecter le Raspberry Pi aux anciens périphériques d'affichage.

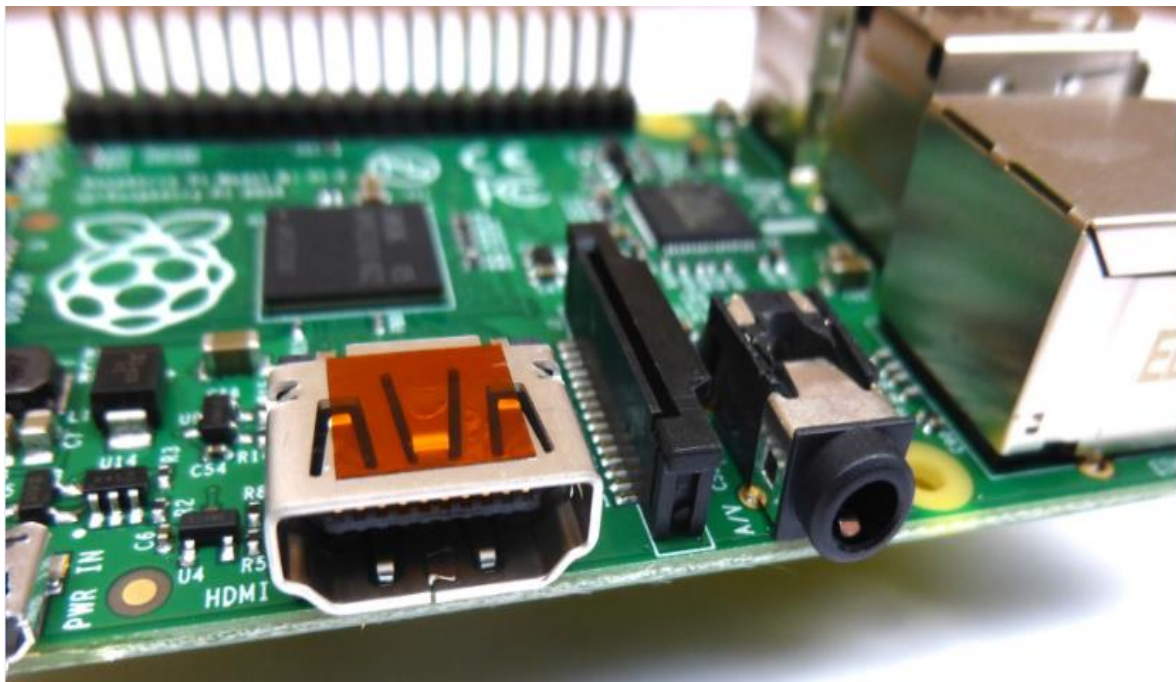


Figure I. 7: Connecteur RCA

Pour utiliser la sortie vidéo composite, vous avez besoin d'un câble adaptateur audiovisuel. Ces câbles, qui sont disponibles pour un coût très bas dans n'importe quel magasin d'électronique, divisent la sortie de la prise jack en trois prises RCA : une prise jaune fournit la connexion vidéo composite, tandis que les prises rouge et blanche offrent les deux canaux de sortie audio stéréo.

Il suffit de brancher le câble de l'adaptateur dans la prise audiovisuelle et de brancher les trois câbles RCA sur celles de votre téléviseur ou d'un autre dispositif d'affichage.

Le Raspberry Pi Zéro n'a pas de prise jack audiovisuelle 3,5 mm à la place, un câble vidéo composite peut être soudé dans les deux trous sur le dessus de la carte où la mention TV. Notez, cependant, que le signal du Pi Zéro n'inclut pas d'audio analogique, contrairement à la prise audiovisuelle des modèles plus grands de Pi.

A.2. Vidéo sur Le Port HDMI

On peut obtenir une meilleure qualité d'image en utilisant le connecteur HDMI (High-Définition Multimedia Interface), situé en bas de la carte. Contrairement à la connexion composite analogique, le port HDMI offre une connexion numérique à haute vitesse pour l'affichage d'images parfaites sur les écrans d'ordinateurs et les téléviseurs HD. En utilisant le port HDMI, un Pi peut afficher des images en résolution Full HD (1920×1080) sur la plupart des téléviseurs HD modernes. Dans cette résolution, on peut afficher des images très détaillées à l'écran.

A.3. Vidéo sur Le Port DSI

La troisième et dernière sortie vidéo du Raspberry Pi, qui se situe au-dessus de la fente pour carte micro SD, en haut de la carte, est un petit connecteur en forme de ruban, protégé par une couche de plastique. Il s'agit d'une norme vidéo connue sous le nom DSI (Display Serial Interface), qui est utilisée pour les écrans plats des tablettes et des smartphones. L'utilisation la plus courante du port DSI sur le Raspberry Pi consiste à connecter l'écran tactile du Pi. Le Raspberry Pi Zéro ne comportant pas de connecteur DSI, il ne peut donc pas être utilisé avec l'écran tactile conçu pour le Raspberry Pi.

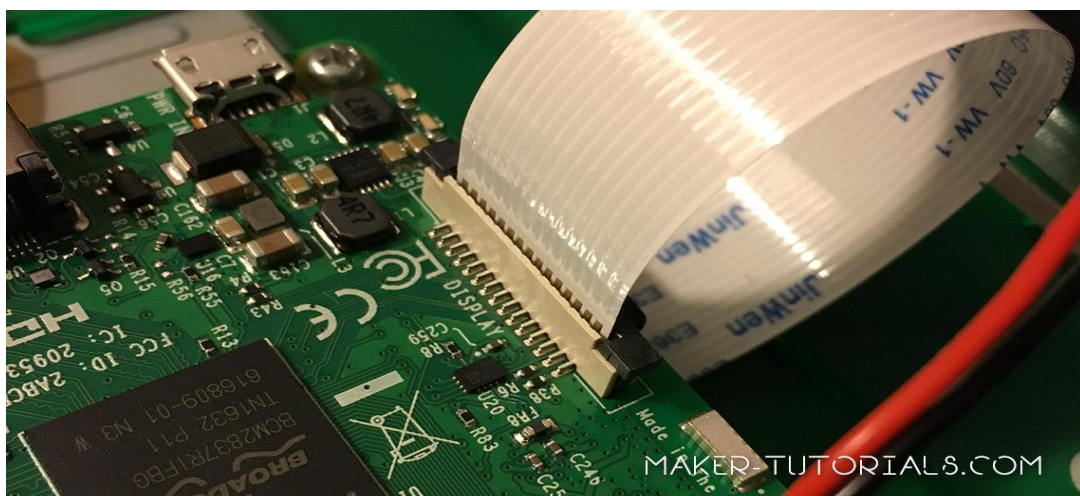


Figure I. 8 : Connecteur DSI

B. Connexion audio

La prise jack audiovisuelle 3,5 mm se situe au bas de la carte. Il s'agit du même connecteur que celui qui est utilisé pour les casques et les microphones sur les appareils audio, et il est câblé exactement dans la même prise que celle qui est employée pour la sortie vidéo composite. Si vous le souhaitez, vous pouvez simplement connecter des écouteurs sur ce port pour bénéficier rapidement de la fonctionnalité audio. Si vous cherchez quelque chose de plus permanent, vous pouvez vous servir de haut-parleurs standards pour PC qui ont un connecteur 3,5 mm ou bien acheter des câbles adaptateurs.

C. Connexion du clavier et la souris

Le branchement du clavier et de la souris consiste simplement à les connecter aux ports USB, soit directement, soit par le biais d'un concentrateur USB, soit en utilisant un adaptateur USB OTG dans le cas du Raspberry Pi Zéro. Vous pouvez connecter un périphérique à n'importe quel port USB, tous les ports étant reliés au processeur du Raspberry Pi de la même manière.

D. Installation de Noobs sur une carte SD

La Fondation Raspberry Pi fournit un utilitaire pour le Pi connu sous le nom de NOOBS (New Out-Of-Box Software¹). Cet outil a pour but de rendre aussi facile que possible la configuration du Pion. Vous pouvez télécharger la dernière version du logiciel NOOBS sur le site de la fondation Raspberry Pi. Cet utilitaire offre une sélection de différents systèmes d'exploitation qui peuvent être installés sur le Raspberry Pi, ainsi que des programmes permettant de modifier les configurations logicielles et matérielles. Pour installer NOOBS sur une carte micro-SD vierge, vous devez utiliser une carte micro-SD ayant au moins une capacité de 8 Go, afin de vous donner suffisamment d'espace pour installer d'autres logiciels que vous utiliserez sur le Pi. Vous devez également disposer d'un ordinateur avec un lecteur de carte micro-SD. Si vous avez déjà utilisé votre carte micro-SD avec un autre appareil, comme un appareil photo numérique ou une console de jeux, utilisez le programme de formatage de l'Association SD Carte pour formater la carte SD et la préparer pour l'installation. Si la carte est neuve, vous pouvez sans risque ignorer cette étape.

Le logiciel NOOBS est fourni sous la forme d'une archive Zip. Il s'agit d'un format de fichiers où les données sont compressées, ce qui permet qu'elles tiennent moins de place et sont donc téléchargées plus rapidement. Après avoir ouvert le fichier utilisez l'option d'éjection de votre système d'exploitation avant de retirer la carte micro-SD, puis insérez-la dans la fente prévue pour accueillir la carte micro-SD sur le Pi

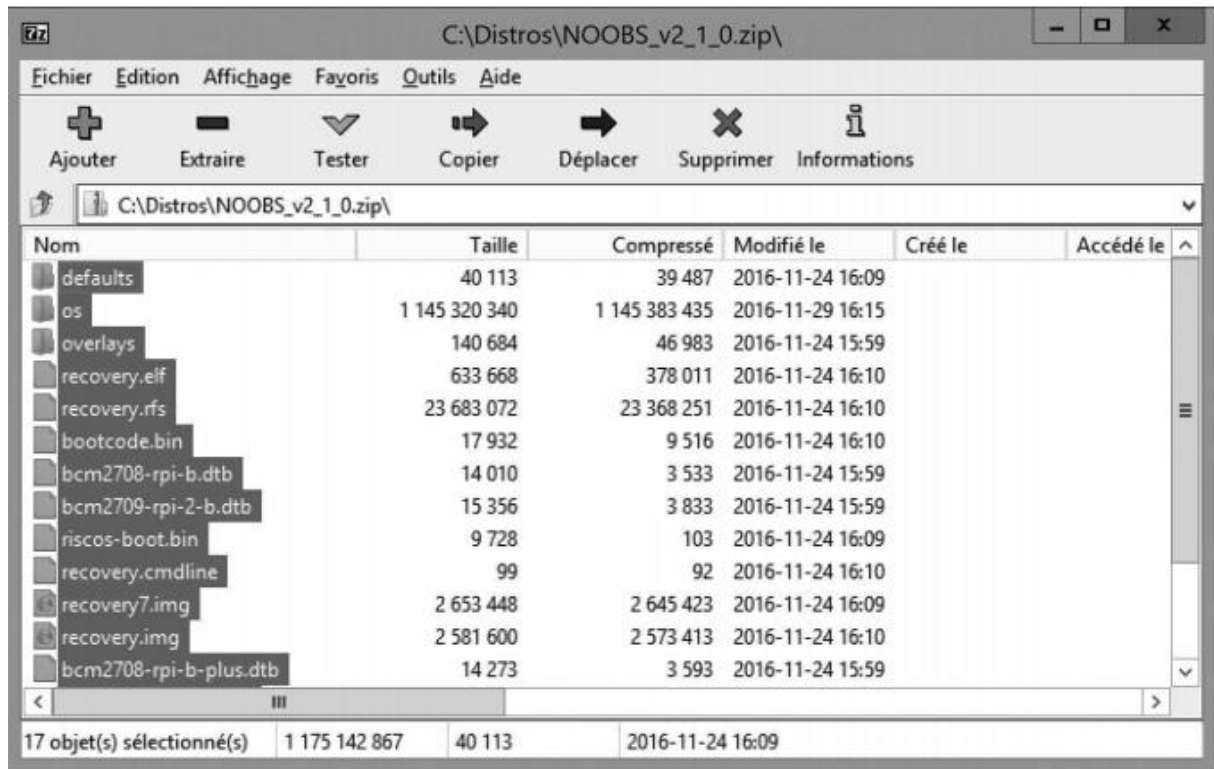


Figure I. 9: Les fichiers de NOOBS

E. Raccordement au Réseau

E.1. Raccordement à un Réseau filaire

Pour raccorder votre Raspberry Pi au réseau, vous devez connecter un câble Ethernet RJ-45 entre le Pi et un Switch ou un routeur. Si vous n'avez pas de routeur, vous pouvez faire communiquer votre ordinateur de bureau ou votre portable avec le Pi en connectant les deux appareils directement avec un câble réseau.

Habituellement, la connexion de deux clients du réseau de cette façon nécessite un câble spécial, appelé câble croisé. Dans un câble croisé,

Les paires de fils assurant la réception et l'émission sont permutées afin que les deux appareils ne puissent pas se parler en même temps, tâche qui est généralement gérée par un Switch.

Le Raspberry Pi est cependant suffisamment intelligent pour gérer cela. Le port RJ-45 situé sur le côté du Pi (figure 2.6) inclut une fonctionnalité connue sous le nom d'auto-MDI, qui lui

permet de se reconfigurer automatiquement. Ainsi, vous pouvez utiliser n'importe quel câble RJ-45, qu'il soit croisé ou non, pour connecter le PI au réseau et il ajustera sa configuration automatiquement.

Quand un câble réseau est connecté, le Pi reçoit automatiquement les informations dont il a besoin pour accéder à Internet, lors du chargement de son système d'exploitation, grâce au protocole DHCP (Dynamic Host Configuration Protocol). Ceci assigne au Pi une adresse IP (Internet Protocol) sur votre réseau et communique l'adresse de la passerelle qui doit être utilisée pour accéder à Internet (il s'agit en général de l'adresse IP de votre routeur ou de votre modem).

E.2. Raccordement à un réseau sans fil

Le Raspberry Pi 3 et 4, les seuls modèles de la gamme qui soit livré avec une prise en charge intégrée du réseau Wi-Fi, mais comme pour le réseau filaire, il est possible d'ajouter le support Wi-Fi à un Pi en utilisant un adaptateur USB sans fil.

En utilisant un tel dispositif, le Pi peut se connecter à un large éventail de réseaux sans fil, y compris ceux qui adoptent la dernière norme 802.11ac qui assure une grande vitesse de communication

F. Connexion de l'alimentation

Le Raspberry Pi est alimenté par le biais d'un petit connecteur micro-USB qui se trouve en bas à gauche de la carte (en bas à droite du Raspberry Pi Zéro). Ce connecteur est le même que celui qui se trouve sur la plupart des Smartphones et de nombreuses Tablettes. De nombreux chargeurs conçus pour les Smartphones fonctionnent avec le Raspberry Pi, mais pas tous. Le Pi demande plus d'énergie que la plupart des périphériques alimentés par une prise micro-USB par exemple, le Pi 3 peut nécessiter un courant de 2 ampères pour fonctionner. Certains chargeurs ne pouvant fournir qu'un courant maximal de 500 mA, cela provoque des problèmes intermittents dans l'exploitation du Pi. Il est possible de relier le Pi au port USB d'un ordinateur portable ou d'un ordinateur de bureau, mais ce n'est pas recommandé. Comme avec les chargeurs les moins puissants, les ports USB d'un ordinateur ne peuvent pas fournir le courant nécessaire au bon fonctionnement du Pi. Connectez le câble d'alimentation micro-USB uniquement lorsque vous êtes prêt à commencer à utiliser le Pi. Comme il n'y a aucun bouton d'alimentation sur le Pi, il se met en marche dès que le câble est connecté.

G. Installation du système d'exploitation

G.1. Installation à L'aide de Noobs

Si vous avez installé Noobs sur votre carte SD ou bien acheté une carte micro-SD avec NOOBS préinstallé, un menu s'affiche lorsque vous allumez le Raspberry Pi pour la première fois. Ce menu fournit une liste des systèmes d'exploitation conçus pour le Pi que vous pouvez installer (vous devez au minimum en installer un, mais vous pouvez aussi en installer plusieurs).

G.2. Introduction à Raspbian

Raspbian est le nom donné à une variante personnalisée de la distribution Linux Debian qui est très populaire. Debian est l'une des plus anciennes distributions Linux qui se caractérise par sa stabilité, une compatibilité élevée et d'excellentes performances, même sur des ordinateurs modestes, ce qui en fait un choix idéal pour le Raspberry Pi. Raspbian est donc basée sur la Debian (sa distribution parente) en y ajoutant des outils personnalisés et des logiciels pour rendre l'utilisation du Raspberry Pi aussi facile que possible.

Pour réduire au minimum la taille du téléchargement, l'image de la Raspbian inclut uniquement un sous-ensemble du logiciel qui est disponible pour la version classique destinée aux ordinateurs de bureau. Elle inclut des outils pour naviguer sur le Web, pour programmer en Python et pour utiliser le Pi avec une interface graphique. Vous pouvez installer rapidement des logiciels supplémentaires à l'aide du gestionnaire de paquets de la distribution, **apt**. Raspbian inclut un environnement de bureau nommé PIXEL (Pi Improved Xwindows Environment, Lightweight). Conçu pour offrir une interface utilisateur attrayante en se basant sur le logiciel X Window System, PIXEL fournit une interface familière et conviviale si vous avez déjà utilisé Windows, OSX ou d'autres systèmes d'exploitation dotés d'une interface graphique pilotable à la souris.

Pour utiliser Raspbian, vous devez saisir un nom d'utilisateur et un mot de passe. Le nom d'utilisateur par défaut est pi, et le mot de passe par défaut est Raspberryvous allez apprendre à modifier ces données plus loin dans ce chapitre.

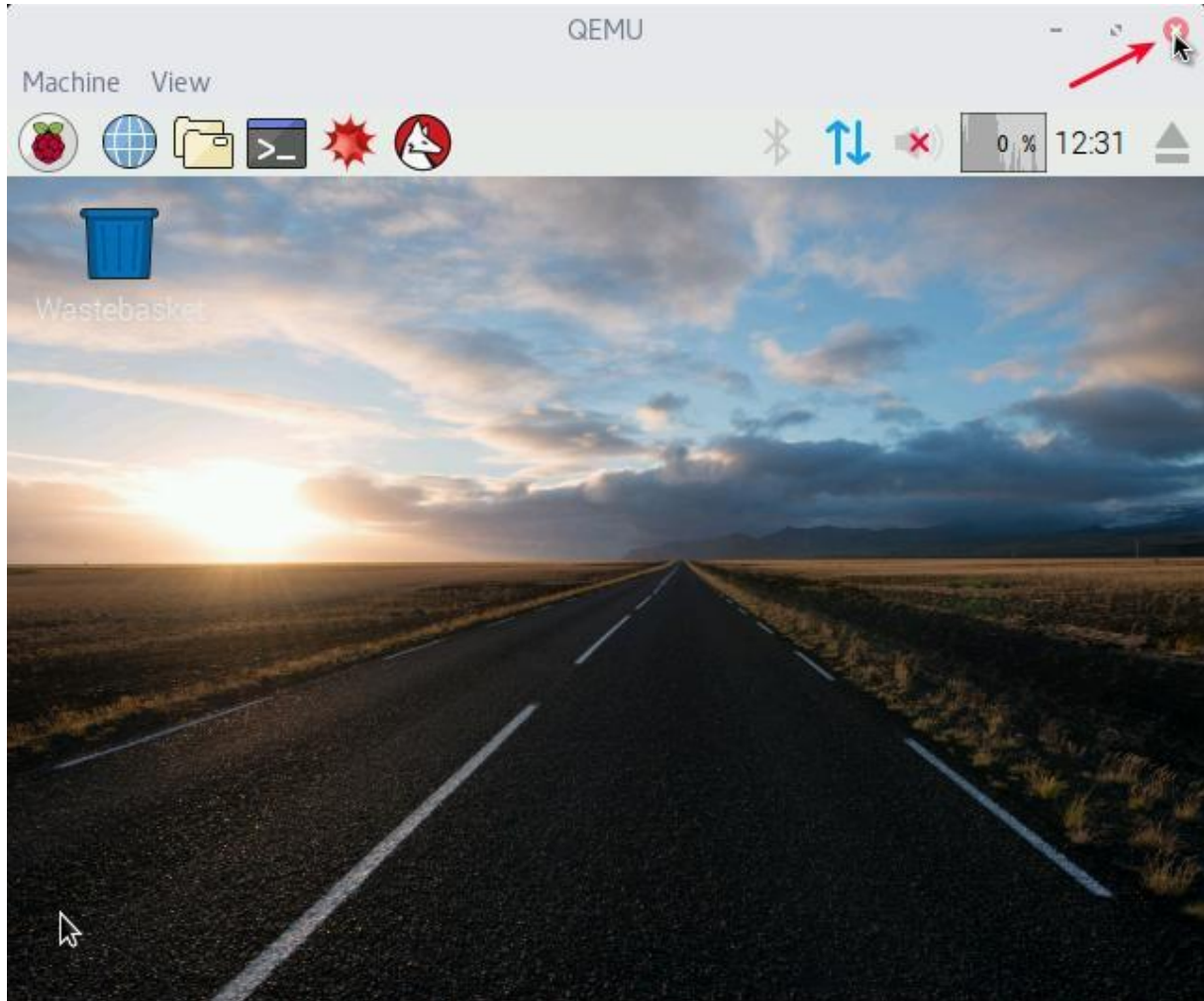


Figure I. 10: Interface graphique de Raspberry

H. Programmation

- **Blue J Java IDE** : puissant environnement de développement intégré (IDE) écrit spécialement pour le langage de programmation Java.
- **Geany** : à mi-chemin entre un éditeur de texte simple et un IDE parfaitement fonctionnel, Geany est un outil populaire pour écrire des programmes dans différents langages.
- **Greenfoot Java IDE** : environnement de développement intégré (IDE) visuel pour Java, conçu spécifiquement pour les jeunes utilisateurs et les débutants en programmation.
- **Mathematica** : puissant logiciel de calcul issu des mathématiques symboliques et développé par Stephen Wolfram. Toutes les installations de Raspbian bénéficient d'une licence gratuite pour utiliser ce logiciel qui est généralement cher.

- **Node-Red:** éditeur de flows (flux) basé sur un navigateur pour le framework Node.JS, conçu pour faciliter le développement de projets matériels et logiciels relativement complexes.
- **Python 2 (IDLE) :** IDE écrit spécialement pour Python. Vous en saurez plus sur l'utilisation d'IDLE quand vous apprendrez à écrire vos propres programmes Python au chapitre 11.
- **Python 3 (IDLE) :** en cliquant sur cette entrée de menu, on charge une version d'IDLE configurée pour utiliser le langage de programmation plus récent Python 3, au lieu de la version 2.7 de Python, qui est celle par défaut. Les deux versions sont en grande partie compatibles entre elles, mais certains programmes peuvent nécessiter des fonctionnalités de Python 3.
- **Scratch :** langage de programmation graphique destiné aux jeunes enfants.
- **Sense HAT Emulator :** permet d'émuler l'utilisation de capteurs avec la carte additionnelle Sense HAT.
- **Sonic Pi :** environnement de développement conçu pour enseigner les concepts de base de la programmation par le biais de la création musicale.
- **Wolfram:** langage développé par le créateur de Mathematica et conçu pour le traitement de la connaissance. Wolfram est extrêmement puissant, mais sa maîtrise demande un certain temps.

❖ *Internet*

- **Claws mail :** client de messagerie puissant, équivalent à Microsoft Outlook.
- **Navigateur Web Chromium :** navigateur web, équivalent à Microsoft Edge ou Internet Explorer.
- **Raspberry Pi resources :** raccourci vers des ressources en ligne pour vous aider à tirer le meilleur parti de votre Raspberry Pi et de Raspbian.
- **The MagPi :** raccourci vers la page d'accueil de The MagPi, le magazine officiel du Raspberry Pi. Chaque numéro de ce mensuel est gratuit et peut être téléchargé sous la forme d'un document PDF.

❖ *Jeux*

- **Minecraft Pi :** version éducative du célèbre jeu Minecraft. Éditée par la société Mojang, cette version est décrite en détail au chapitre 12.
- **Python Games :** sélection de jeux simples écrits en langage Python, qui permettent à la fois de jouer et d'apprendre à programmer en Python.

I. Port GPIO

Le Pi est contient des ports GPIO (général Purpose Input-Output, ou en français, entrées sorties à usage général), situé en haut à gauche du circuit imprimé du Pi.

Le GPIO permet au Pi de communiquer avec les autres composants et circuits et peut servir de contrôleur si on l'intègre dans un circuit électronique plus large. Grâce au port GPIO, il est possible de mesurer la température, de déplacer des servos et communiquer avec d'autres périphériques informatiques en utilisant différents protocoles, dont SPI (Serial Peripheral Interface) et I²C (Inter-Integrated Circuit).

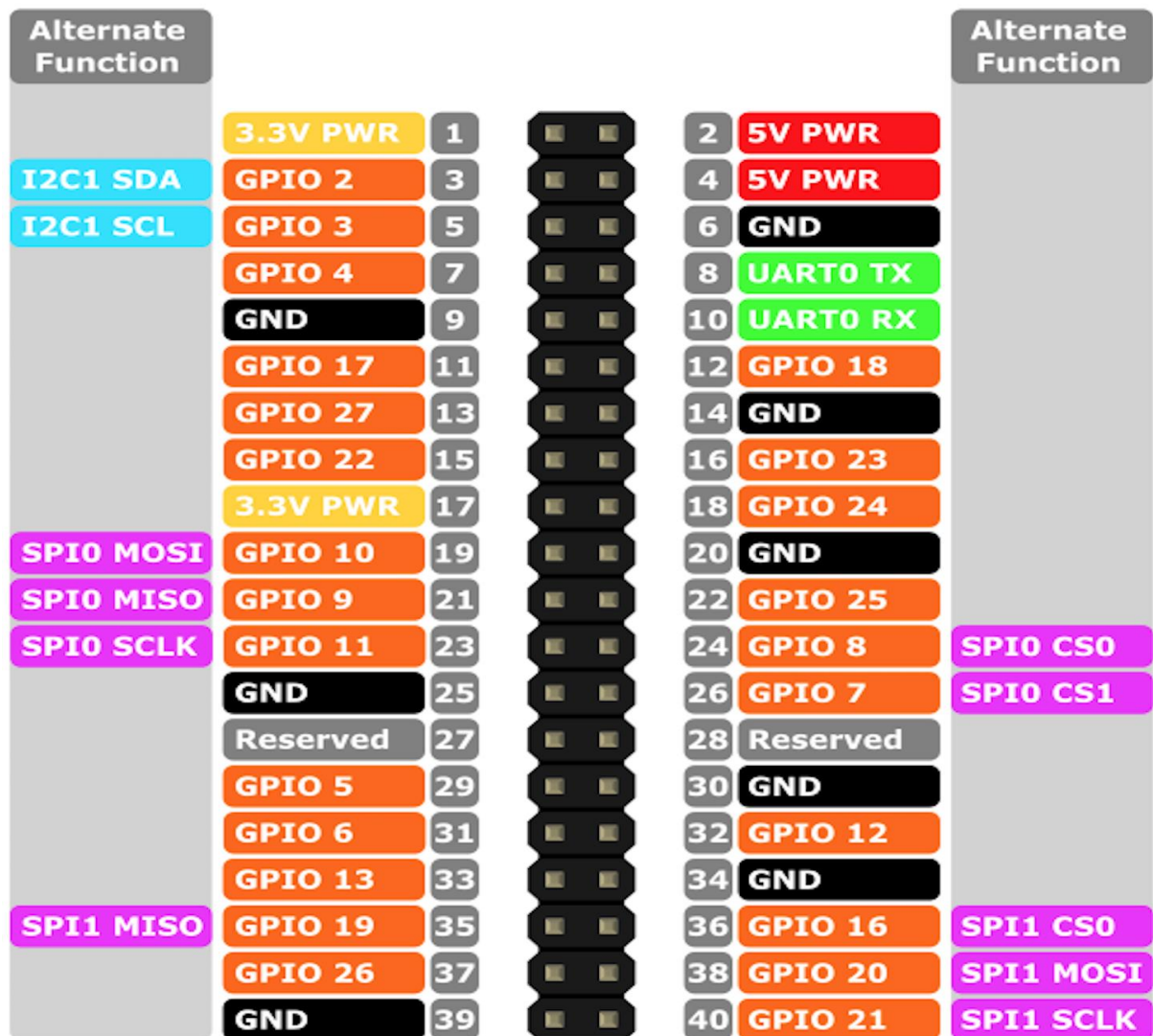


Figure 3.11 : Les ports GPIO de Raspberry pi

I.1. Caractéristiques du GPIO

Selon votre modèle de Raspberry Pi, le port GPIO fournit par défaut au moins huit broches à usage général : les broches 7, 11, 12, 13, 15, 16, 18 et 22. Ces broches peuvent prendre trois états high, où la broche fournit une tension positive de 3,3 V, low, où la tension est égale à 0 V ou à la masse, et input (entrée). Les deux sorties correspondent à 1 et à 0 en logique binaire et peuvent être utilisées pour activer ou désactiver d'autres composants. Les Raspberry Pi qui ont un port GPIO de 40 broches offrent un certain nombre de broches supplémentaires à usage général, qui sont décrites dans le schéma de brochage.

I.2. Mise en garde

Si vous connectez une alimentation en 5 V à n'importe quelle broche du port GPIO du Raspberry Pi ou si vous reliez directement une des broches d'alimentation (Broches 2 et 4) à n'importe quelle autre broche, vous allez endommager votre Pi. Comme le port est câblé directement aux broches du processeur Broadcom BCM283x, vous allez causer des dommages irréversibles. Soyez toujours très prudent lorsque vous travaillez avec le port GPIO.

I.5. Partie de supervision

C'est la partie qui effectue **la supervision** du système. Aussi appelé l'interface homme machine. Elle permet d'effectuer des réglages d'afficher des messages et de gérer les défauts.

I.5.1. Définition de la supervision

La supervision est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés. La supervision concerne l'acquisition de données (mesures, alarmes, retour d'état de fonctionnement) et des paramètres de commande des processus généralement confiés à des automates programmables. [11]

En informatique industrielle, la supervision des procédés peut être une application de surveillance, de contrôle-commande ou de diagnostic ou l'ensemble de ces dernières réunies. Elle se fait à travers un logiciel fonctionnant sur un ordinateur en communication, via un réseau local ou distant industriel, avec un ou plusieurs équipements.

I.5.2. Avantages de la supervision

Un système de supervision donne de l'aide à l'opérateur dans la conduite du processus, son but est de présenter à l'opérateur des résultats expliqués et interprétés et son avantage principal est:

- Surveiller le processus à distance.

- La détection des défauts.
- Le diagnostic et le traitement des alarmes.
- Traitement des données.

I.5.3. Interface Homme-Machine

La programmation et la mise en marche d'une installation industrielle automatisée ne sont pas suffisantes, il est donc nécessaire de visualiser l'état et le mode de fonctionnement de l'installation. Il existe plusieurs configurations d'interface de contrôle / commande. La configuration la plus simple est de rassembler toutes les informations sur une **Interface Homme-Machine**, pour faciliter la tâche de l'opérateur [11].

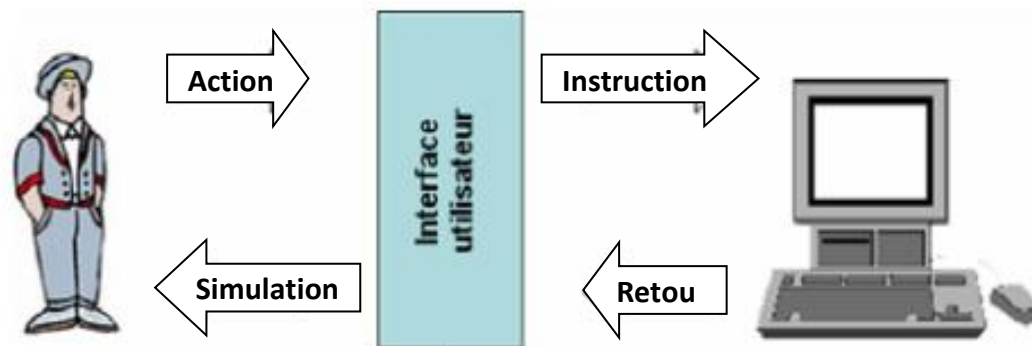


Figure I. 11 : Interface Homme-Machine

I.5.4. IHM SAMKOON SK-102HS

La série SK IHM de SAMKOON est un produit d'interface Homme-Machine de haute qualité, il intégrant entièrement une unité centrale, une unité d'entrée et de sortie, un écran, une mémoire et d'autres unités de module, il peut être largement appliqué dans les systèmes de contrôle industriels de tous les secteurs de l'industrie.

Avec une conception optimale via matériel et logiciel, il est conforme aux exigences de contrôle de la machine pour la précision et la précision du toucher, ainsi que pour les couleurs d'écran.

Sur la base de la version précédente, l'IHM de la série SK apporte de grandes améliorations avec un temps de démarrage plus rapide, vitesse de communication supérieure et une sensibilité plus élevée qu'auparavant. Il fournit une interface série standard pouvant être connectée à d'autres périphériques. Principalement compatible avec la communication série, il peut communiquer avec les principaux API du marché, tels que SAMKOON, MITSUBISHI, SIEMENS, OMRON, MODICON.



Figure I. 12: Face avant de SK-102HS

A. Différents ports de connexions et d'alimentation de SK-102HS

La figure ci-dessous représente les différents ports de connexions et d'alimentation de SK-102HS.

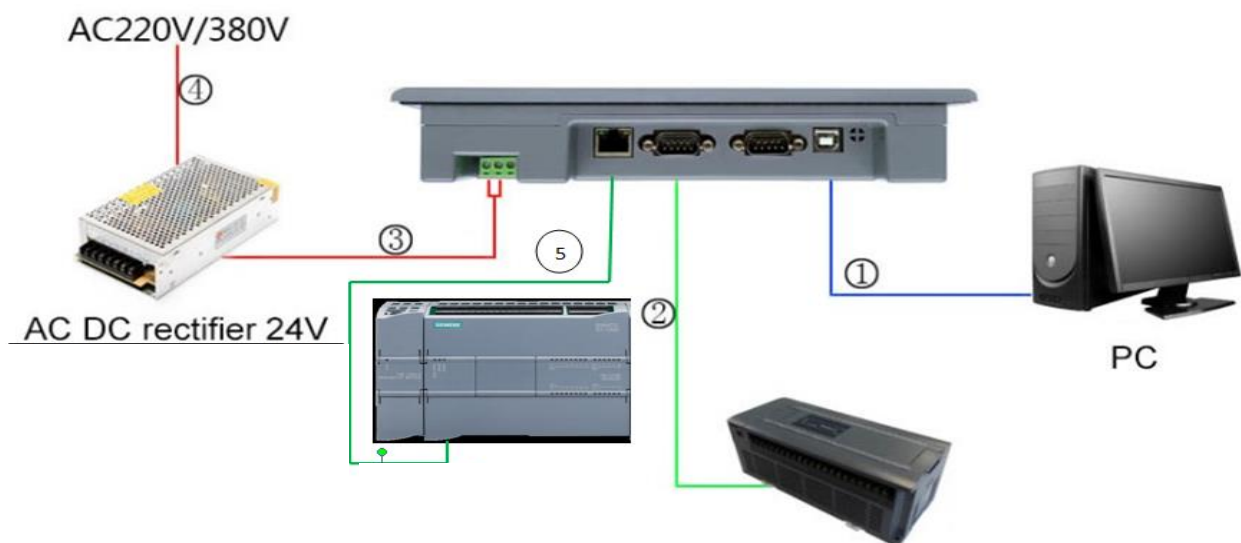


Figure 3.14 : Schéma de connexion de l'écran tactile de la série SK

- ① Un port de programmation (USB).
- ② Deux ports de programmation (COM).
- ③ Un connecteur d'alimentation (24V DC).
- ④ Alimentation de 220V ou 380V alternative
- ⑤ Un connecteur de connexion (Ethernet RG45).

B. Caractéristiques techniques de SK-102HS

Le tableau suivant représente les caractéristiques techniques de SK-102HS.

Tension d'alimentation	24V DC (+/-15%)
Consommation d'énergie	6W
Portes séries	COM1 et COM2 pour liaison série RS232/422/485
Port USB-A	Support pour une vitesse de transmission de 10M/S
Port USB-B	Support pour flash disc, imprimante, souris et autres périphériques USB
Porte Ethernet	Réaliser la communication entre les IHM ou IHM et le dispositif de commande via Ethernet (S7-1200).
Mémoire	128M FLASH + 128M DDR2
Processeur	processeur hautes performances de qualité industrielle: cortex A8, jusqu'à 600 MHz
Logiciel de programmation	SKTOOL 6.2

Tableau I.3 : Tableau de caractéristique de SK-102HS

I.6. Conclusion

Dans ce chapitre nous avons étudié les systèmes automatisés, on a basé beaucoup plus sur la partie commande et la partie supervision, ainsi la description de microcontrôleur le Raspberry. Dans la première partie de commande on a parlé sur les automates programmables et ces structures internes et externes.

Dans la seconde partie, nous atteignons a la partie de description de Raspberry qui contient une aperçu générale sure les Raspberry qui existe ainsi que la configuration nécessaire avec sa langages programmation.

La partie IHM nous permettra de contrôler et de commander notre unité à distance et nous présenter l'IHM SAMKOON SK-102HS et la possibilité de communiquer avec les principaux API de marché.

CHAPITRE II

Les actionneurs pneumatiques et électriques

II.1. Introduction

La Partie Opérative (P.O) d'un automatisme est formée de l'ensemble des divers organes physiques qui interagissent sur le produit pour lui conférer une valeur ajoutée : les pré-actionneurs, les actionneurs et les capteurs. Les pré-actionneurs servent de relais de puissance entre la commande et les actionneurs qui agissent et transforment le produit. Les capteurs recueillent les informations : état ou position du produit, alarmes, etc. traduisant un changement d'état du procédé. Ces changements induisent le calcul de la commande du procédé de production par son système de Contrôle/Commande ou Partie Commande.

Dans ce chapitre, nous présenterons quelque éléments de la partie opérative tel que les capteurs, les pré-actionneurs, les actionneurs pneumatiques (vérins pneumatiques) et électriques (Moteur asynchrone monophasé).

II.2. Les capteurs

II.2.1. Définition

Le capteur est un composant de la chaîne d'acquisition dans une chaîne fonctionnelle. Le capteur prélève une information sur le comportement de la partie opérative et la transforme en une information exploitable par la partie commande. Une information est une grandeur physique (présence d'objet, chaleur, lumière, etc...). Pour pouvoir être traitée, cette information sera portée par un support physique (énergie) on parlera alors de signal. Les signaux sont généralement de nature électrique.

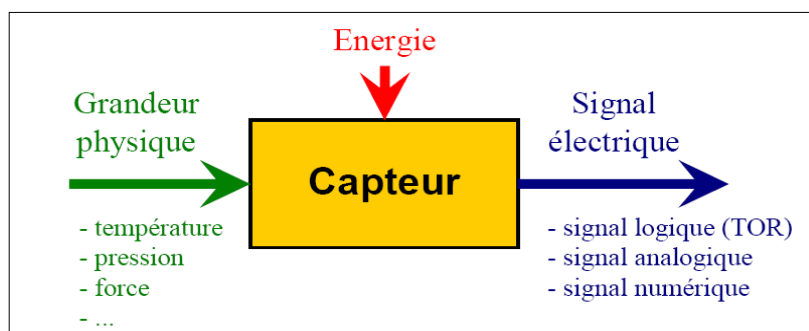


Figure II 1: Principe de fonctionnement du capteur

On peut caractériser les capteurs selon deux critères :

- En fonction de la grandeur mesurée ; on parle alors de capteur de position, de température, de vitesse, de force, de pression, etc.

- En fonction du caractère de l'information délivrée ; on parle alors de capteurs logiques appelés aussi capteurs tout ou rien (TOR), de capteurs analogiques ou numériques.

On peut alors classer les capteurs en deux catégories :

- ✓ Les capteurs à contact qui nécessitent un contact direct avec l'objet à détecter.
- ✓ les capteurs de proximité.

Chaque catégorie peut être subdivisée en trois catégories de capteurs : les capteurs mécaniques, électriques, pneumatiques.

Dans ce chapitre nous allons présenter le capteur optique car c'est le capteur utilisé dans notre projet.

A. Le capteur photoélectrique (Capteur optique)

A.1. Définition

Un capteur photoélectrique est un capteur de proximité. Il se compose d'un émetteur de lumière associé à un récepteur. La détection d'un objet se fait par coupure ou variation d'un faisceau lumineux. Le signal est amplifié pour être exploité par la partie commande.

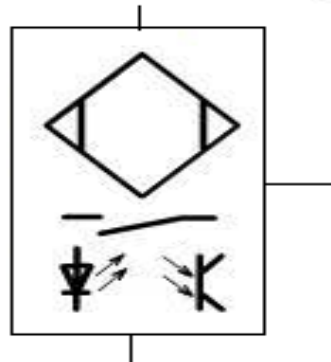


Figure II 2 : Symbole du capteur photoélectrique de 3fil

A.2. Avantages

- ✓ Pas de contact physique avec l'objet détecté.
- ✓ Détection d'objets de toutes formes et de matériaux de toutes natures.
- ✓ Détection à très grande distance (jusqu'au 30m).
- ✓ Sortie statique pour la rapidité de réponse ou sortie à relais pour la commutation de charges jusqu'à 2 A.
- ✓ Généralement en lumière infrarouge invisible, indépendante des conditions d'environnement.

A.3. Détections

- Tous les objets.
- Dépend de l'opacité et de la réflexion de l'objet.
- n capteur photoélectrique peu détecté jusqu'à plusieurs mètres.

A.4. Principe de fonctionnement

Il existe trois systèmes de détection :

1. Système barrage
2. Système reflex
3. Système proximité



Figure II 3 : Différents types de détection d'un capteur photoélectrique

❑ Détection par barrage

Où l'objet à détecter coupe un faisceau lumineux situé entre l'émetteur et le récepteur, La distance de détection peut atteindre 30 m.

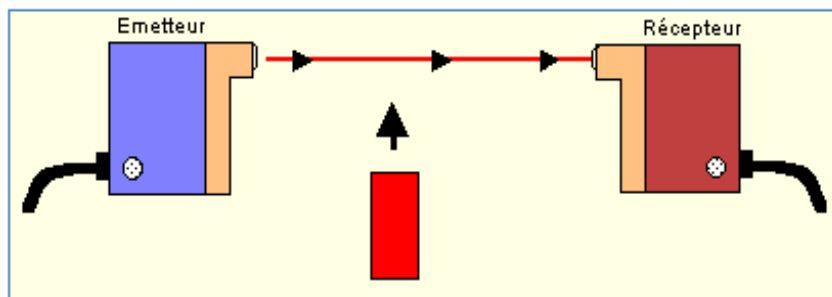


Figure II 4 : Capteur photoélectrique, détection par barrage

❑ Détection par réflex (Les cellules reflex)

Sont composées d'un émetteur/récepteur (dans le même boîtier) et d'un réflecteur (catadioptré). L'émetteur envoie le faisceau qui revient vers le récepteur après s'être réfléchi sur le catadioptré. L'objet à détecter coupe le faisceau.

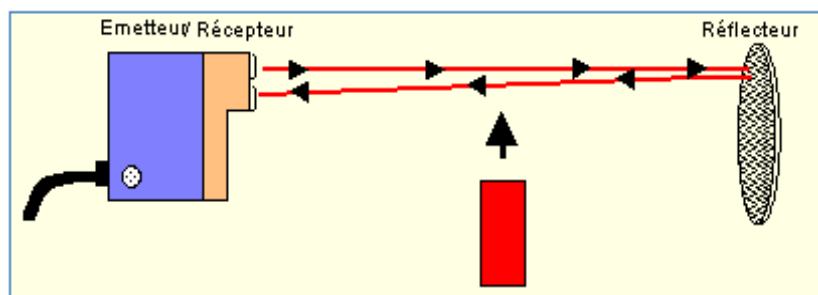


Figure II 5 : Capteur photoélectrique, détection par cellules reflex

La distance de détection est **2 à 3 fois inférieure** au système en barrage.

❑ Détection par proximité

Sont dotées d'un émetteur qui envoie le faisceau. Celui-ci se réfléchit directement sur l'objet à détecter lui-même avant de retourner au récepteur.

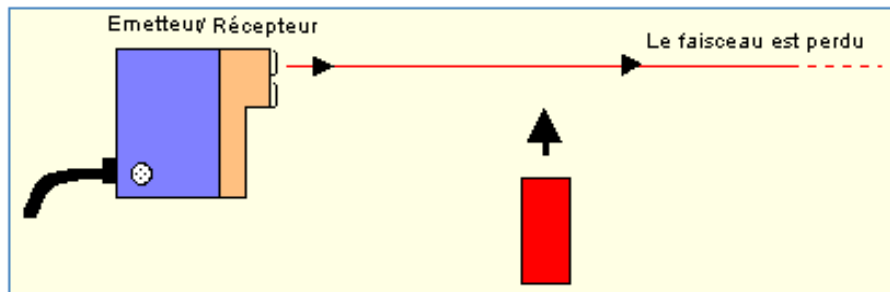


Figure II 6 : Capteur photoélectrique, détection par cellules reflex

La distance de détection (assez faible) et son efficacité dépendent de la couleur et de la taille de l'objet à détecter.

A.5. Branchement

Deux types de branchement sont retenus :

- ❑ La technique **3 fils** pour les détecteurs alimentés en courant continu, deux fils servent à l'alimentation (généralement le bleu la masse et le marron 24V), le troisième (Noire) à la transmission du signal de sortie.
- ❑ La technique **4 fils** pour les détecteurs alimentés en courant continu, généralement deux fils pour l'alimentation (Marron + et Bleu -), le troisième fil (Noir) pour le signal NO et le quatrième fil (Blanc) pour le signal NF.

II.3. Les pré-actionneur

II.3.1. Définition

Un pré-actionneur permet de distribuer de l'énergie à un actionneur, ce dernier convertissant l'énergie reçue en énergie utile.

II.3.2. Fonctionnement

La principale fonction d'un pré-actionneur est de transmettre un ordre de la partie commande à la partie opérative. Généralement utilisé pour commander des puissances en fonction d'un signal de

commande de faible puissance. Son rôle est donc de générer l'énergie de commande de l'actionneur.

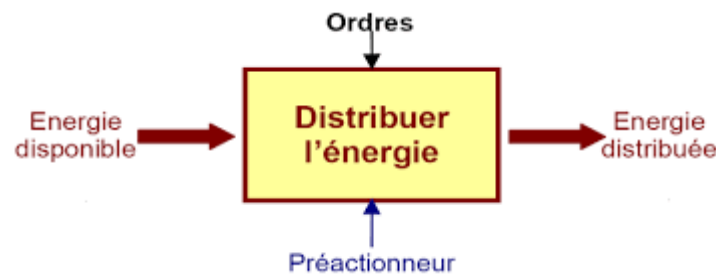


Figure II 7 : Le distributeur (Schéma de principe)

Remarque : la plupart des pré-actionneurs sont dits « tout ou rien », c'est-à-dire que :

- soit ils empêchent l'énergie d'aller vers l'actionneur.
- soit ils font passer tout le flux d'énergie disponible vers l'actionneur. Ils sont alors soit « ouvert » soit « fermé », tel un interrupteur : un préactionneur tout ou rien commande l'établissement ou l'interruption de la circulation de l'énergie entre une source et un actionneur.

II.3.3. Classification des pré-actionneurs

En fonction des grandeurs d'entrée et de sortie, on peut classer les pré-actionneurs par :

- Pré-actionneurs électriques (les contacteurs et les relais)
- Pré-actionneurs pneumatiques (les distributeurs).

A. Les pré-actionneurs électriques

A.1. Définition

Les pré-actionneurs sont des constituants qui, sur ordre de la partie de commande, assurent la distribution de l'énergie de puissance aux actionneurs. Dans les circuits électriques, les pré-actionneurs sont généralement soit un relais, soit un contacteur. Le contacteur est un relais avec la différence que ses contacts sont prévus pour supporter un courant beaucoup plus important. Ainsi, un contacteur est utilisé afin d'alimenter des consommateurs (des moteurs,...) de fortes puissances (plus de 50 kW).

Le contacteur assure en plus l'extinction de l'arc électrique qui accompagne souvent la commutation de l'énergie de forte puissance. En effet, quand on ouvre un circuit en cours de fonctionnement, le contact en cause provoque un arc électrique qui peut être dangereux pour les biens et les personnes.

A.2. Le relais

Le relais est un composant électrique réalisant la fonction d'interfaçage entre un circuit de commande, généralement bas niveau, et un circuit de puissance alternatif ou continu (Isolation galvanique). On distingue deux types de relais :

- Le relais électromagnétique.
- Le relais statique.



A.2.1. Relais électromagnétique Figure II 8 Relais électrique 5V DC

Principe de fonctionnement

Un relais électromagnétique est constitué d'une bobine alimentée par le circuit de commande, dont le noyau mobile provoque la commutation de contacts pouvant être placé dans un circuit de puissance. Le relais électromagnétique est réservé pour les faibles puissances.

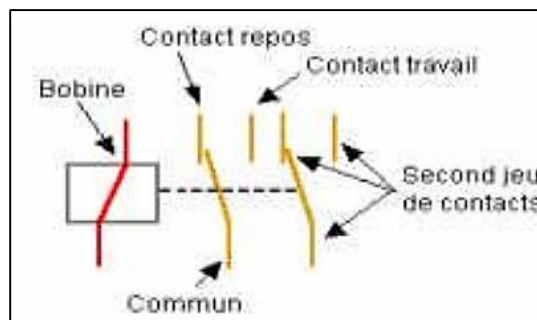


Figure II 9 : Symbole de relais 5V DC

Caractéristiques fondamentales

- ✓ **Tension d'alimentation** : C'est une tension continue qui permet d'exciter la bobine.
- ✓ **La résistance de la bobine** : paramètre permettant de déterminer le courant circulant dans le circuit de commande.
- ✓ **Le courant des contacts** : c'est le courant maximal que peut commuter les contacts de relais sans dommage.

A.3. Le contacteur

Le contacteur est un relais électromagnétique qui permet grâce à des contacts (pôles) de puissance d'assurer le fonctionnement de moteurs, de résistances ou d'autres récepteurs de fortes puissances.

Suivant le modèle, il possède aussi des contacts auxiliaires intégrés ouverts ou fermés, il est possible d'ajouter des additifs ou blocs auxiliaires servant uniquement pour la télécommande ou la signalisation

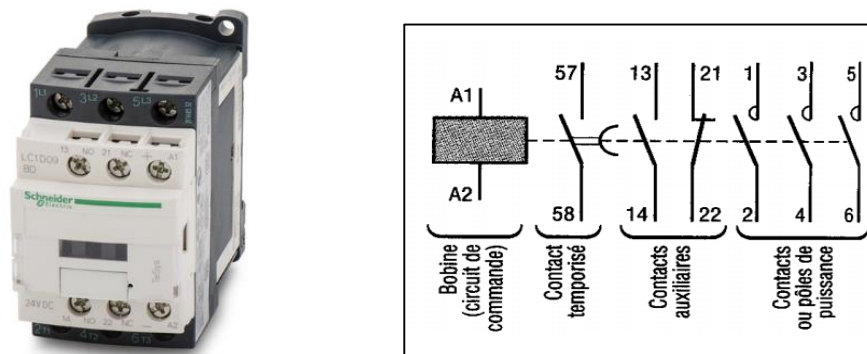


Figure II 10 : Image et symbole d'un contacteur

A.3.1. Principe de fonctionnement

La bobine du contacteur (bornes A1-A2), peut-être alimentée en courant alternatif ou en courant continu (24V, 48V, 110V, 230V, 400V). Lorsque la bobine est alimentée, un champ magnétique se forme, la partie mobile de l'armature est attirée contre la partie fixe et les contacts se ferment (ou s'ouvrent suivant le modèle). Lorsque la bobine n'est pas alimentée, le ressort de rappel sépare les deux parties de l'armature et maintient les contacts de puissance ouverts (ou fermés).

B. Les distributeurs

Les distributeurs sont des pré-actionneurs qui ont pour rôle de diriger le fluide (sous pression ou sans pression) dans certaines directions. C'est grâce à eux qu'on peut piloter la sortie ou la rentrée de tige d'un vérin.

Un distributeur est caractérisé par :

- ✓ **Le type de commande** (manuel, électrique, hydraulique, pneumatique,...ect) ;
- ✓ **Sa stabilité** (monostable ou bistable) ;
- ✓ **Le nombre de position** et le nombre d'orifices.

B.1. Stabilité d'un pré-actionneur pneumatique

Un pré-actionneur est dit monostable s'il a besoin d'un ordre pour le faire passer de sa position repos à sa position travail, et que le retour à sa position repos s'effectue automatiquement lorsque l'ordre disparaît : il n'est stable que dans sa position repos.

Exemples de pré-actionneurs monostables : un bouton poussoir, un relais électromagnétique Un pré-actionneur est dit bistable s'il a besoin d'un ordre pour passer de sa position repos à sa position travail et qu'il reste en position travail à la disparition de cet ordre. Il ne peut revenir à sa position repos que s'il reçoit un second ordre : il est stable dans les deux positions repos et travail.



Figure II 11 : Distributeur 5/2 bistable commande électrique

B.2. Les différents types de distributeurs pneumatique et hydraulique

On désigne un distributeur avec deux chiffres : Le premier chiffre désigne le nombre d'orifices et le deuxième désigne le nombre de positions de communication.

Exemple : un distributeur 2/2 possède 2 orifices et 2 positions de communication.

La schématisation d'un distributeur permet de connaître le nombre de positions, d'orifices, de voies, et les différents types de pilotages. Le symbole d'un distributeur contient :

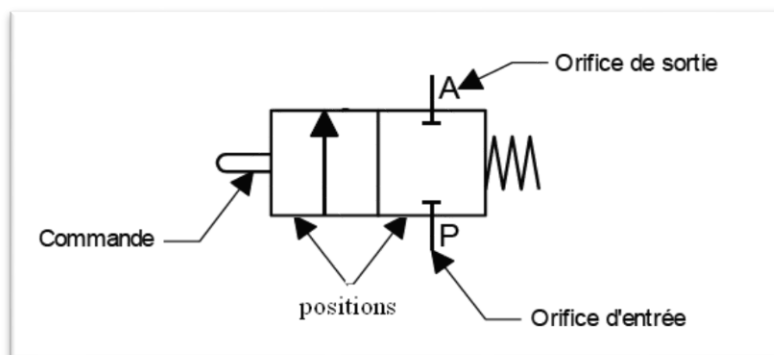
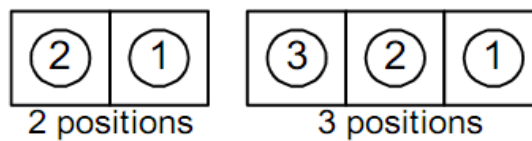


Figure II 12 : Le symbole d'un distributeur 2/2 commande poussoir

B.3. Principe de la symbolisation d'un distributeur pneumatique et hydraulique

Les positions se représentent à l'aide de 2 ou 3 cases :



Les orifices sont représentés comme ci-dessous :

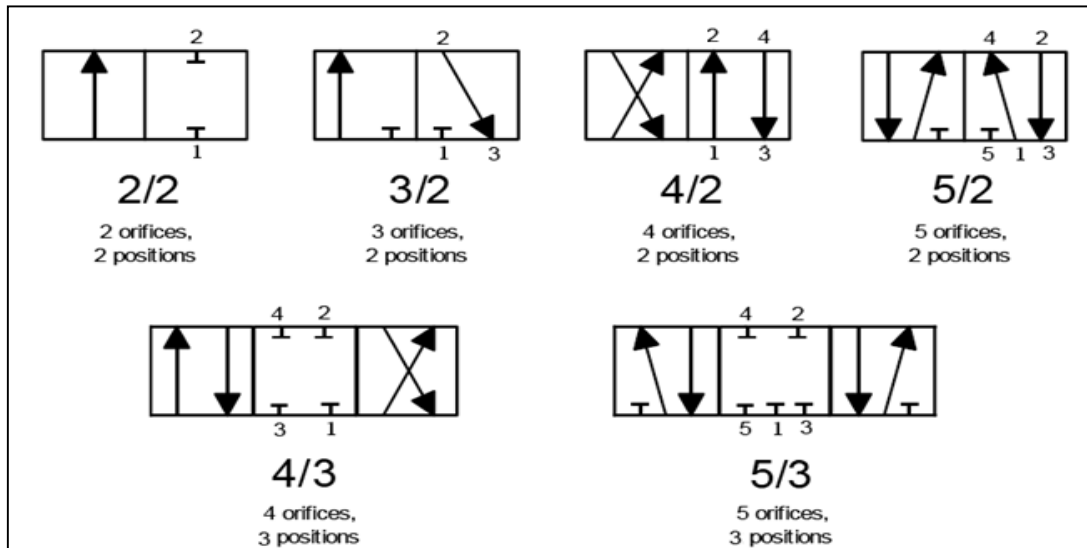


Figure II 13 : Les différents symboles d'un distributeur

II.4. Les actionneurs hydrauliques et pneumatiques

II.4.1. Les actionneurs hydrauliques

Les actionneurs hydrauliques présentent l'intérêt d'avoir un excellent rapport puissance/masse, ce qui justifie leur utilisation en particulier pour les robots de forte puissance. Un autre de leur avantage est que les frottements sont réduits et leur temps de réponse court. Ils constituent donc de bon organe d'asservissement en particulier dans le cas où l'on a à produire des efforts importants à vitesse lente. Toutefois leur commande est délicate en particulier pour un asservissement en boucle ouverte. En raison de la montée en puissance des actionneurs électriques ce type d'actionneur a tendance à être abandonné.

II.4.2. Les actionneurs pneumatiques

Les actionneurs pneumatiques utilisent de l'air comprimé à faible pression (de l'ordre de 10 bars). En raison de la compressibilité de l'air il est nécessaire d'utiliser ces actionneurs en boucle fermée mais ils présentent l'intérêt de supporter des régimes de surcharge et d'avoir un bon amortissement. Ce type d'actionneur est plutôt utilisé pour des manipulateurs très simples ainsi que pour la commande des pinces ou d'outils de manipulation.

II.4.3. Les vérins

A. Définition

Vérin est un actionneur qui convertit une énergie pneumatique ou hydraulique d'entrée en énergie mécanique de sortie utilisable pour obtenir une action définie. Leur classification tient compte de la nature du fluide, pneumatique ou hydraulique, et du mode d'action de la tige :

- **Simple effet** : air comprimé admis sur une seule face du piston *Tirer* ou *Pousser*.
- **Double effet** : air comprimé admis sur les deux faces du piston *Tirer* et *Pousser*.

B. Caractéristiques principales d'un vérin

Les vérins sont principalement utilisés lorsque l'on veut des mouvements linéaires rapides (transfert, serrage, indexage, bridage, éjection, assemblage, ...). Ils ont caractérisés par :

- **Le diamètre D du piston** lié à la tige (8 mm à 320 mm dans les applications courantes), D est calculé pour obtenir l'effort axial voulu en sortie.
- **La course L du piston** (donc de la tige, de 1 mm à 3000 mm, L est choisie en fonction du déplacement souhaité.
- **La pression p d'alimentation** du vérin (en bar ou en Pascal, 1 bar = 100 000 Pa), p est imposée par le réseau de branchement (8 bars en général).
- **L'effort axial F que la tige** peut exercer à la sortie (en Newton) donnée par la relation :

$$F = p \times S \quad 2.1$$

Avec : F la force que peut fournir le vérin en « N ».

p la pression de l'air comprimé en « **bar ou Pa** ».

S la surface sur laquelle l'air comprimé agit en « m^2 ».

- **la vitesse de déplacement V** de la tige exprimée en $m.S^{-1}$ que l'on adapte en réglant le débit d'air au niveau de l'échappement (en général $0,2 m/S < V < 0,3 m/S$).

C. Principe de fonctionnement

C'est l'air comprimé qui, en pénétrant dans l'une des chambres, pousse le piston. La tige se déplace. L'air présent dans l'autre chambre est donc échappé et évacué du corps du vérin.

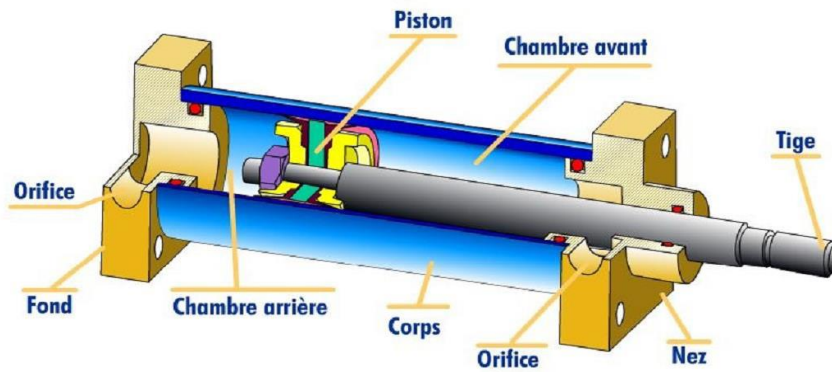


Figure II 14 : Constituants de base d'un vérin

D. Différents Types des Vérins

D.1. Vérin simple effet

L'air comprimé qui commande le déplacement de la tige est admis d'un seul côté du piston (commande en sortie). Le retour de la tige en position initiale est généralement obtenu à l'aide d'un ressort. Ces vérins ne permettent pas d'exercer un effort au retour en position initiale.

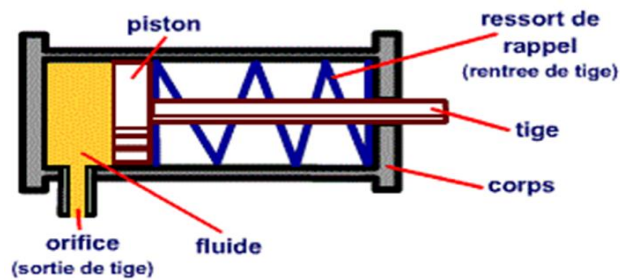


Figure II 15 : Vérins simple effet

D.2. Vérins double effet

Ces vérins comportent deux orifices d'admission de l'air comprimé ce qui permet de commander la sortie ou la rentrée de la tige en fonction du besoin. Ainsi ces vérins permettent d'exercer un effort dans les deux sens.

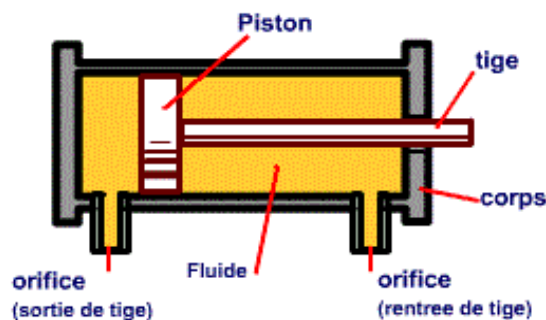


Figure II 16 : Vérins double effet

C. Symbole

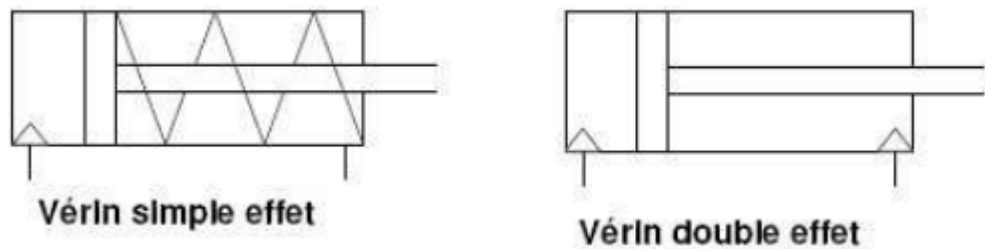


Figure II 17 : Symboles des Vérins simple et double effet

II.5. Les actionneurs électriques

II.5.1. Le moteur asynchrone monophasé

A. Définition

Le moteur asynchrone monophasé, bien que moins utilisé dans l'industrie que son homologue triphasé, représente néanmoins une part d'applications non négligeable dans les petites puissances et dans les applications du bâtiment qui utilisent le réseau monophasé 220 V.

A puissance égale, il est plus volumineux qu'un moteur triphasé. Par ailleurs, son rendement et son $\cos\phi$ sont beaucoup plus faibles que dans le cas du triphasé et ils varient considérablement en fonction d'une part de la puissance, d'autre part du constructeur.

B. Constitution

Le moteur monophasé, comme le moteur triphasé, est composé de deux parties : le stator et le rotor.

- ✓ **Le stator** : Il comporte un nombre pair de pôles et ses bobinages sont raccordés sur le réseau d'alimentation.
- ✓ **Le rotor** : Il est le plus souvent à cage d'écureuil

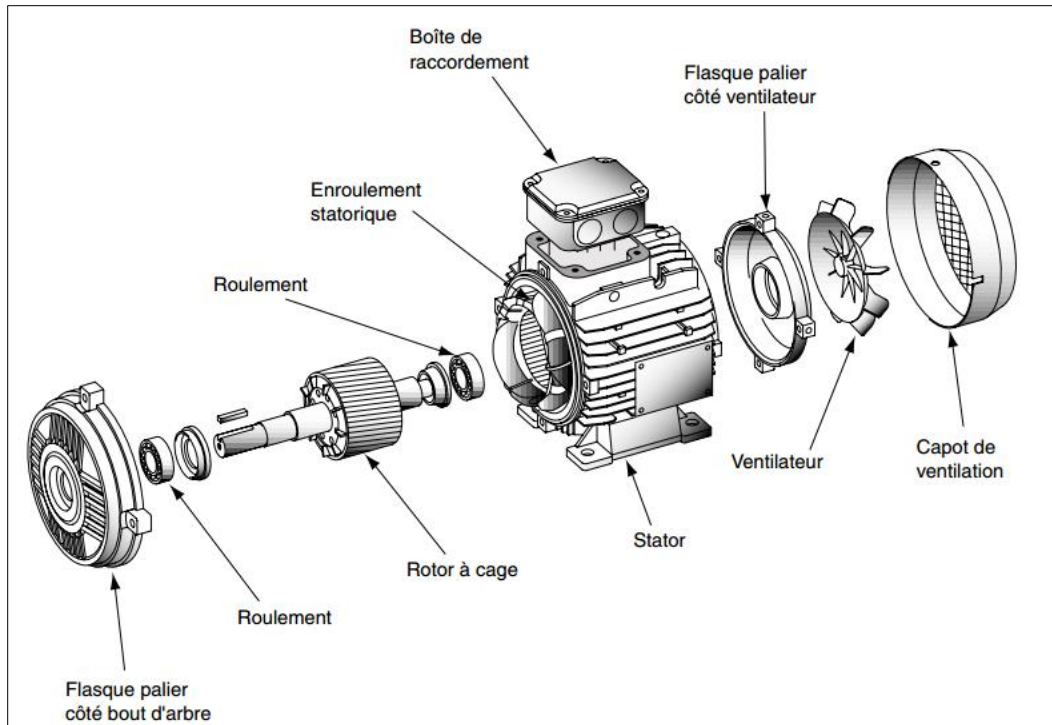


Figure II 18 : Composants d'un moteur monophasé

C. Principe de fonctionnement

Considérons un stator comprenant deux enroulements raccordés sur le réseau d'alimentation L1 et N (Figure 2.19). Le courant alternatif monophasé engendre dans le rotor un champ alternatif simple H qui est la superposition de deux champs tournants H1 et H2 de même valeur et de sens contraires.

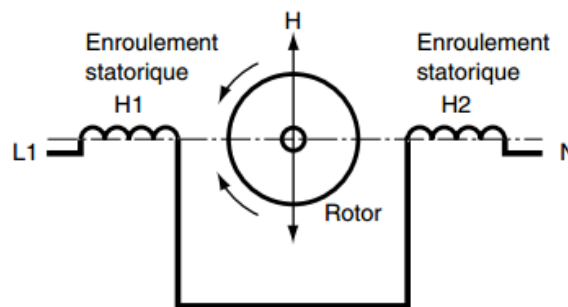


Figure II 19 : Schéma de principe de fonctionnement d'un moteur monophasé

A l'arrêt, le stator étant alimenté, ces champs présentent le même glissement par rapport au rotor et produisent par conséquent deux couples égaux et opposés. Le moteur ne peut démarrer.

Une impulsion mécanique sur le rotor provoque une inégalité des glissements. L'un des couples diminue pendant que l'autre augmente. Le couple résultant provoque le démarrage du moteur dans le sens où il a été lancé.

Afin de résoudre ce problème de couple lors de la phase de démarrage, un deuxième bobinage décalé de 90° est inséré dans le stator. Cette phase auxiliaire est alimentée par un artifice de déphasage (condensateur ou inductance) ; une fois le démarrage effectué la phase auxiliaire peut être supprimée.

D. Types des moteurs monophasés

Il y a plusieurs types des moteurs monophasés, les plus populaires sont :

- Moteur à phase auxiliaire
 - ◆ Moteurs à phase auxiliaire résistive
 - ◆ Moteurs à phase auxiliaire capacitive
- Moteurs à condensateur permanent
- Moteurs à bagues de court-circuit (Cage d'écureuil)
- Moteurs universels série

D.1. Moteur à condensateur permanent

A. Propriétés

Les moteurs à condensateur permanent ressemblent fortement aux moteurs à phase auxiliaire capacitive.

Les principales différences résident :

- ✓ dans l'enroulement auxiliaire.
- ✓ dans les types de condensateur utilisés.

Ainsi, le nombre de spires de l'enroulement auxiliaire est supérieur à celui de l'enroulement principal, alors que le condensateur est non pas électrolytique mais au papier imprégné d'huile (figure ci-dessous).

B. Choix d'un moteur monophasé à condensateurs permanents

- ✓ pour faible couples de démarrage
- ✓ faible puissance car mauvais rendement
- ✓ simples ; robustes, faibles coûts de maintenance
- ✓ silencieux
- ✓ pour les sècheurs à cheveux, humidificateur, four à micro-ondes, etc.

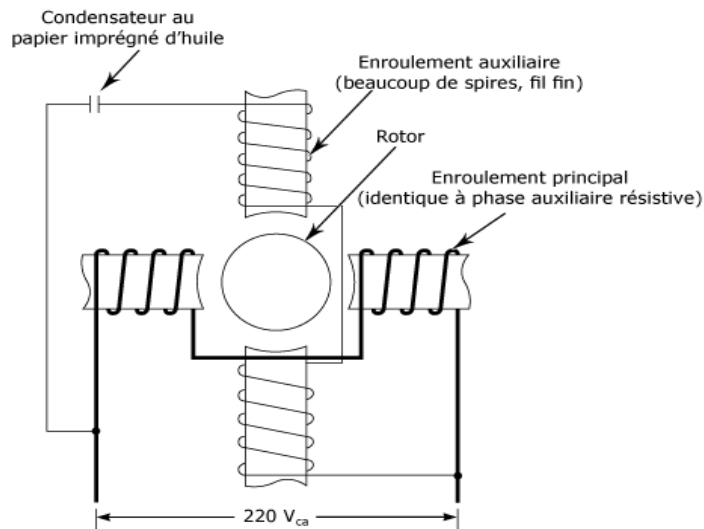


Figure II 20 : Schéma de principe d'un moteur à condensateur permanent

Ces différences font qu'il n'est pas nécessaire d'avoir un interrupteur centrifuge pour ouvrir le circuit auxiliaire quelque temps après le démarrage.

C. Avantages et inconvénients des moteurs à condensateur permanent

Le moteur à condensateur permanent a pour principaux avantages sa simplicité et son fonctionnement doux. En effet, ce type de moteur est très silencieux et, contrairement aux autres moteurs monophasés, son degré de vibration est réduit.

Il a cependant pour inconvénient un couple de démarrage faible et le coût élevé du condensateur au papier imprégné d'huile le rend trop cher pour les grandes puissances. Il est donc rare qu'on rencontre un moteur à condensateur permanent qui dépasse 500 W de puissance.

II.6. Conclusion

Dans ce chapitre nous avons présenté la partie opérative de notre projet tel que les capteurs, les pré-actionneurs, les actionneurs pneumatique/hydraulique et électriques. Dans le premier temps nous avons donné une idée générale sur capteurs notamment la définition, les types de capteurs, on a parlé plus détail sur le capteur de proximité photoélectrique qu'on a utilisé dans notre projet. Dans la seconde partie on a parlé sur les pré-actionneurs et les actionneurs pneumatiques et hydrauliques. Par la suite on a expliqué les actionneurs électriques comme le moteur électrique monophasé.

CHAPITRE III

Les appareillages électriques

III.1. Introduction

Une armoire électrique industrielle comporte un tableau électrique qui alimente et contrôle un automate industriel ou une chaîne de montage (alimentée la partie opérative et commande). En effet, divers éléments du tableau servent à commander l'installation électrique, comme les fusibles ou encore les interrupteurs. L'armoire industrielle permet également de protéger ces divers éléments de facteurs extérieurs, comme des conditions climatiques rudes ou encore de la poussière. Nous présentons dans ce chapitre tout le nécessaire afin de constituer notre armoire électrique industrielle, de l'enveloppe aux composants en passant par les divers accessoires.

III.2. Description des différents appareillages électriques utilisés

III.2.1. Les contacteurs

Dans notre armoire électrique on a utilisé deux contacteurs de marque Schneider, le premier utilisé pour alimenter le moteur asynchrone monophasé du convoyeur et le deuxième fonctionne avec le bouton d'arrêt d'urgence pour couper l'alimentation électriques sur les actionneurs possède les caractéristiques suivantes



Figure III. 1 : Caractéristiques des contacteurs Schneider LC1D25

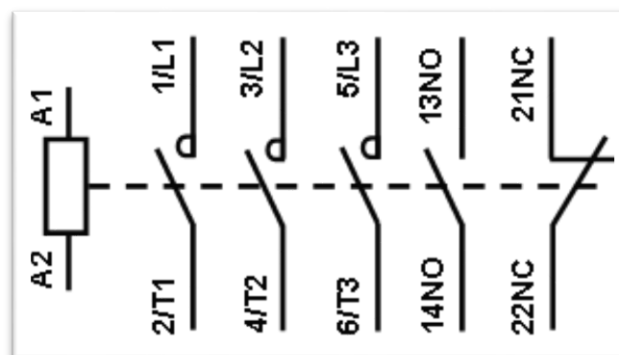


Figure III. 2 : Symbole d'un contacteur

III.2.2. Les accessoires complémentaires

Le contact auxiliaires utilisé de marque Schneider LADN40, la figure suivante présente les caractéristiques techniques de ce contact.

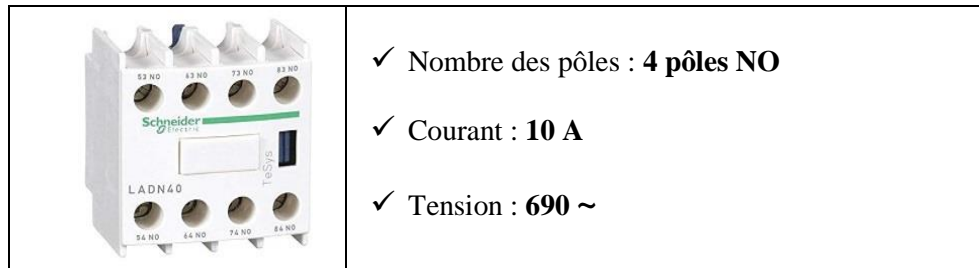


Figure III. 3 : Caractéristiques du contact auxiliaires Schneider LADN40

III.2.3. Appareils de protection

A. Disjoncteur Magnétothermique

Pour protéger notre système (installation) contre les court-circuit et les surcharges en utilisant un disjoncteur magnétothermique de marque Schneider possède les caractéristiques :

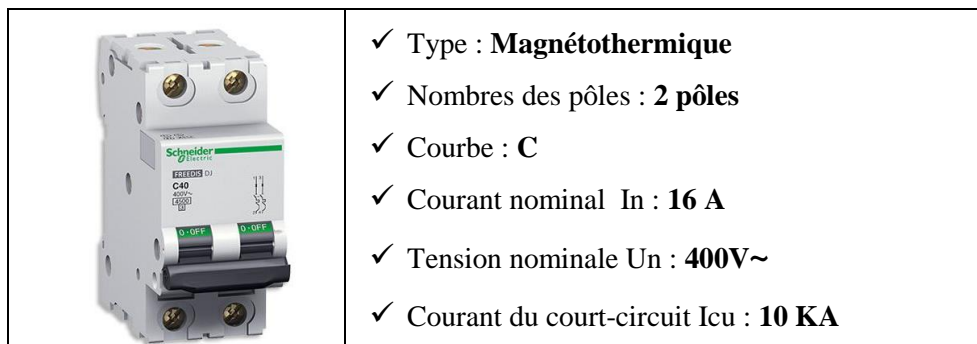


Figure III. 4 : Caractéristiques électriques du disjoncteur magnétothermique utilisé

B. Sectionneur porte fusible Schneider A9N15651

Dispositif de protection à fusibles pour les circuits de commande ou puissance jusqu'à 32A.

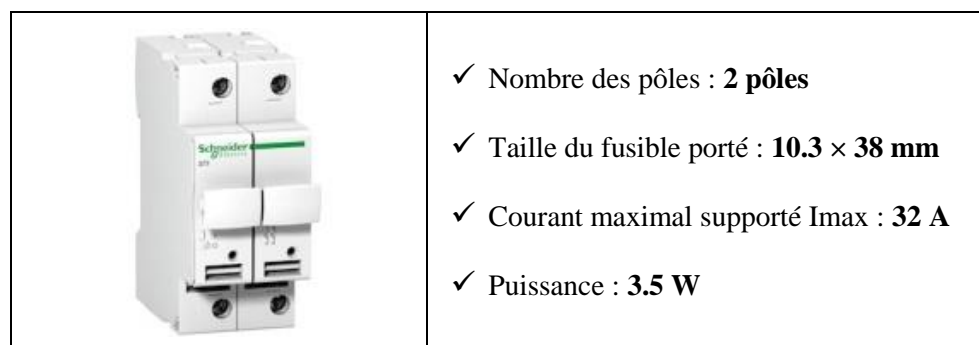


Figure III. 5 : Caractéristiques électriques du sectionneur porte fusible

C. Fusible céramique gG 2A

Le rôle du fusible est de protéger un circuit contre les surintensités ou les court-circuit. Le fusible utilisé de type céramique gG possède les caractéristiques suivantes :

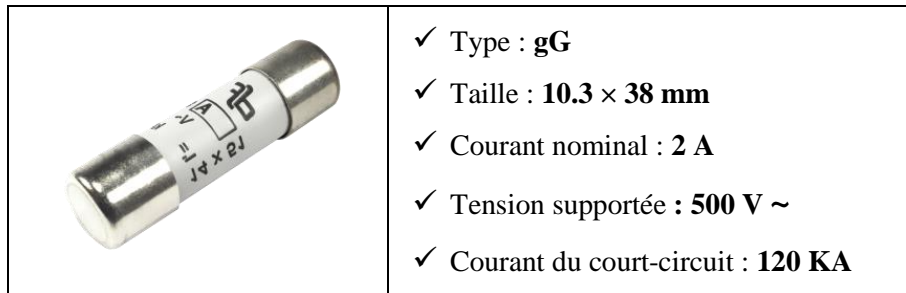


Figure III. 6 : Image et caractéristiques du fusible utilisé

III.2.4. Alimentation continu de 24V DC

Pour alimenter l'API et les autres appareillages électriques qui fonctionnent avec 24V DC on utilise une alimentation de découpage de 24 V DC.

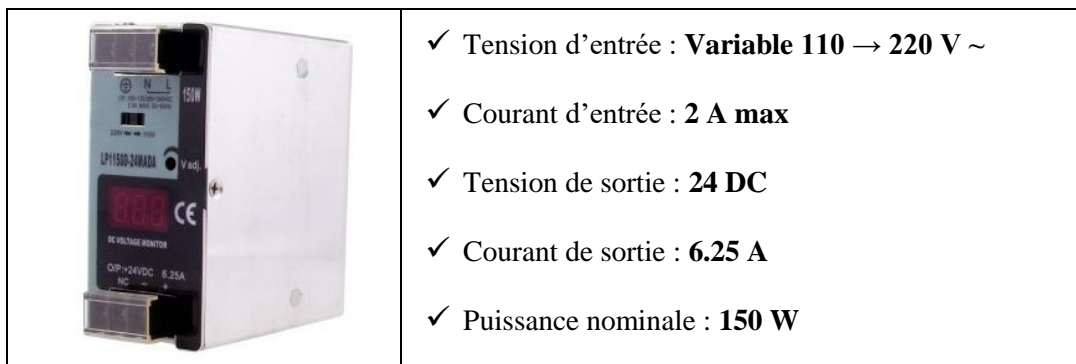


Figure III. 7 : Image et caractéristiques électriques d'alimentation continue utilisée

Les alimentations stabilisées sont utilisées pour fournir une tension de 24V courant continu.

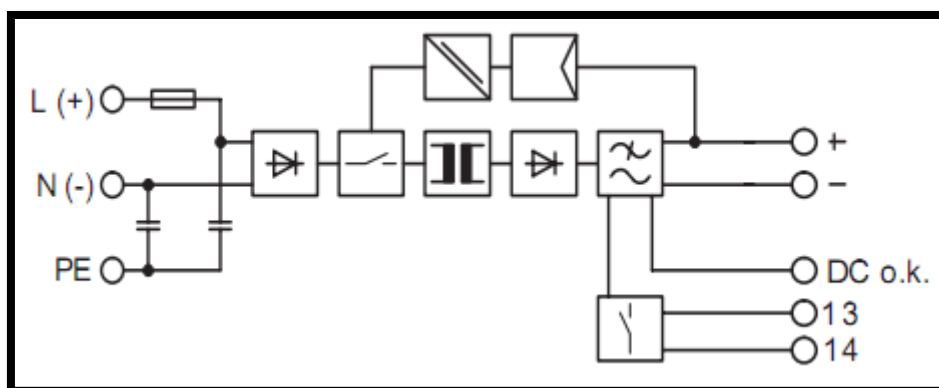


Figure III. 8 : Schéma synoptique d'une alimentation 24V DC

III.2.5. Bouton d'arrêt d'urgence

En cas un de blessures survient ou lorsque le flux de requiert on a utilisé un bouton d'arrêt d'urgence de type Relâchement par torsion pour couper l'alimentation électrique sur les actionneurs.

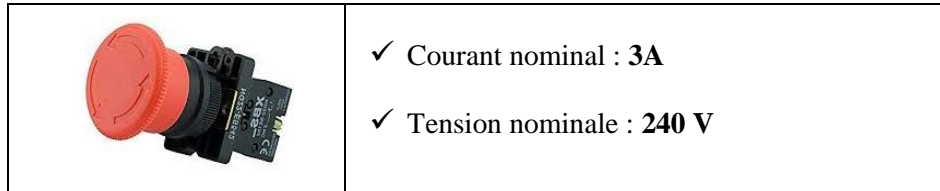


Figure III. 9 : Image et caractéristiques électriques du bouton d'arrêt d'urgence

III.2.6. Les boutons poussoirs

Pour mise en marche et mise en arrêt notre machine de tri on a utilisé deux boutons poussoirs, un bouton poussoir à fermeture NO et un boutons poussoir à ouverture NC.

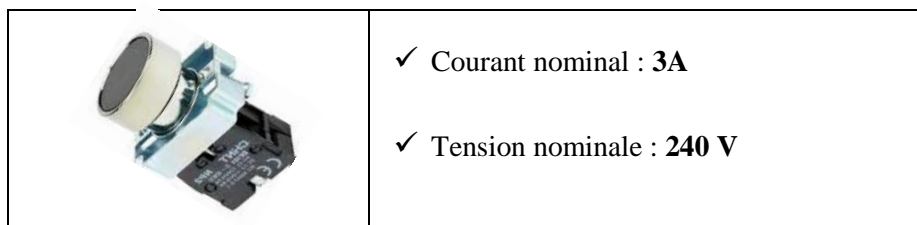


Figure III. 10 : Image et caractéristiques électriques du bouton poussoir

III.2.7. Le commutateur

Pour établir ou interrompre le circuit électrique d'une façon manuelle on a utilisé un commutateur de 2 positions

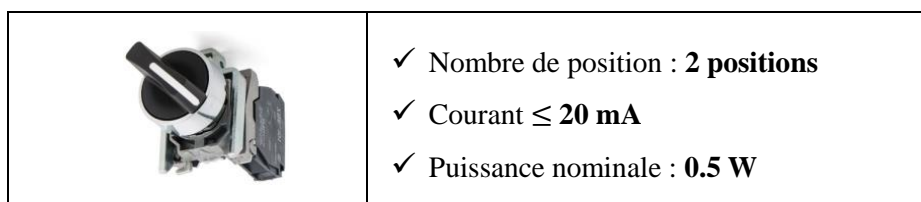


Figure III. 11 : Image et caractéristiques électriques du commutateur

III.2.8. Les appareils de signalisation

A. Les lampes (voyants)

Notre circuit et équipement électrique est signalés par des lampes (voyants) de signalisation de différents couleurs (rouge, vert et orange).

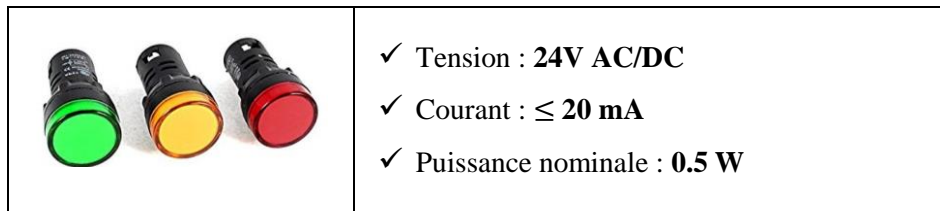


Figure III. 12 : Image et caractéristiques électriques des voyants

B. Appareil de sonore (Buzzer)

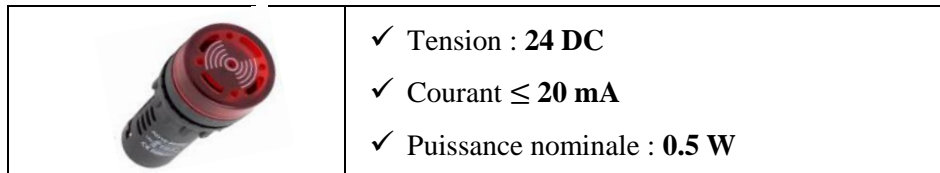


Figure III. 13 : Image et caractéristiques électriques du buzzer

III.2.8. Borne électrique

Pour distribuer la terre, la phase et le neutre (l'alimentation) au sein du notre tableau électrique ou pour raccorder un ou plusieurs câbles électrique on a utilisé un borne électrique possède les caractéristiques suivants :

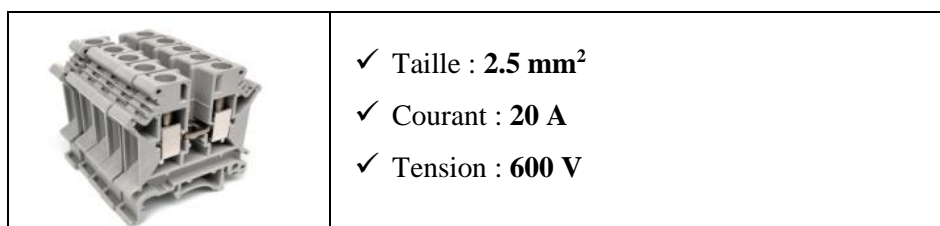


Figure III. 14 : Image et caractéristiques de la borne électrique

III.2.9. Convertisseur Buck réglable (LM2596)

Un convertisseur buck, ou hacheur série, est une alimentation à découpage qui convertit une tension continue en une autre tension continue de plus faible valeur. Un convertisseur Buck bien conçu possède un fort rendement (jusqu'à 95 %) et offre la possibilité de régler la tension de sortie.



Figure III. 15 : LM2596

A. Principe de fonctionnement

Le circuit électrique ci-dessous représente le Principe de fonctionnement d'un convertisseur Buck.

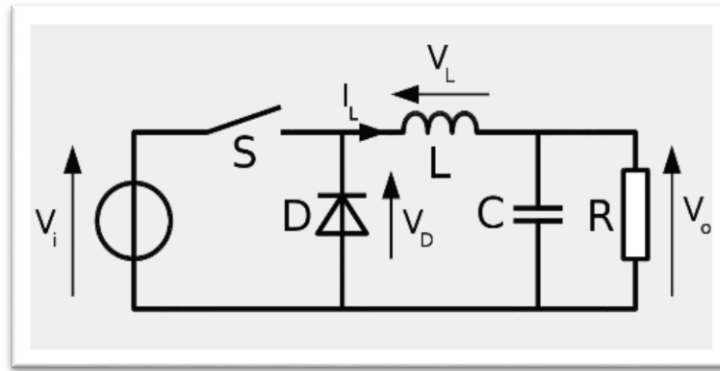


Figure III. 16 : Principe de fonctionnement d'un convertisseur Buck

La pièce essentielle est l'inductance, qui passe alternativement de charge (réduisant la tension, fournit par le générateur, disponible pour le circuit en aval) à source (fournissant une tension alors que le générateur est mis hors circuit). Le fonctionnement peut être divisé en deux phases suivant l'état de l'interrupteur S :

- On ferme l'interrupteur S. Le courant fourni par le générateur, initialement nul, augmente linéairement et traverse l'inductance. L'inductance s'oppose à cette augmentation du courant, produisant une tension opposée, et elle stocke l'énergie reçue sous forme magnétique. la tension aux bornes de l'inductance vaut $V_L = V_i - V_o$. La diode étant polarisée en inverse avec la tension V_i du générateur, aucun courant ne la traverse.
- L'interrupteur est ouvert. Le générateur est mis hors circuit, plus aucun courant ne le traverse. La diode devient passante afin d'assurer la continuité du courant dans l'inductance. Le courant traversant l'inductance décroît. L'inductance s'oppose à cette réduction du courant, produisant une tension qui la met en situation de source pour le circuit aval, en utilisant l'énergie magnétique stockée à la phase précédente.

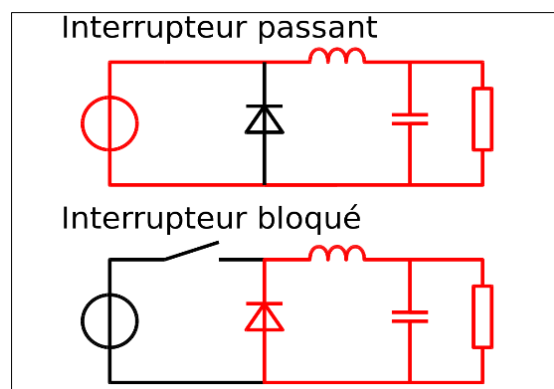


Figure III. 17 : Schéma de base d'un convertisseur Buck

La tension aux bornes de l'inductance vaut $V_L = -V_o$

Un circuit de commande de l'interrupteur le fait alterner à un rythme adéquat pour obtenir la tension désirée. La diode peut être remplacée par un autre interrupteur.

B. Applications

Ce type de convertisseur est utilisé pour des applications que l'on peut classer en deux catégories :

- ☑ Les applications visant à obtenir une tension continue fixe (et parfois régulée) à partir d'un générateur de tension continue supérieure.
 - Conversion des 12-24V fournis par une batterie d'ordinateur portable vers les quelques Volts nécessaires au processeur.
 - Conversion de la tension du secteur redressée en continu fixe.
- ☑ Les applications permettant d'obtenir une tension réglable mais toujours inférieure à celle présente à l'entrée.
 - Variateur de tension continue.

Pour ces deux catégories d'application, on souhaite également que l'impédance interne du système ainsi créée soit faible - du même ordre de grandeur que celle du générateur qui alimente le dispositif - ce qui interdit l'usage d'un diviseur de tension dissipant sous forme de chaleur l'excès de tension et possédant un faible rendement ce qui est rédhibitoire pour les applications d'électronique de puissance.

III.2.10. Lumière LED

Nous avons utilisé un type de lumière LED 12 V comme source de lumière dans la chambre de traitement d'image. L'éclairage nécessaire aide à prendre des photos claires, ce qui donne de bons résultats de tri.

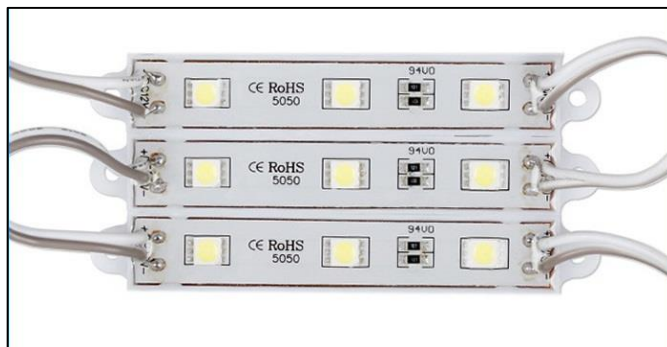


Figure III. 18 : Lumière LED

III.3. Conclusion

Après avoir terminé ce chapitre nous avons constaté que l'armoire électrique joue un rôle très important dans n'importe quelle installation électrique soit domestique ou industrielle. Nous avons également parlé sur les différents matériels utilisés d'une armoire électrique tel que : les contacteurs, les fusibles, sectionneur porte fusible, les disjoncteurs, l'arrêt d'urgence ...etc.

CHAPITRE IV

Réalisation de la machine de tri automatisée

IV.1. Introduction

Notre but est de réaliser une machine automatisée pour le tri des pommes en fonction de sa couleur à travers d'une caméra qui prise des images et un processeur « Raspberry » qui fait le traitement d'images commandée par un API et supervisée par un IHM. Dans ce chapitre on va présenter en détail les différentes parties réalisées notamment la partie de commande, la partie opérative et la partie de puissance.

IV.2. Cahier de charge

- ◆ Lors qu'en appuyant sur le bouton poussoir « Marche » le moteur du convoyeur doit démarrer et la lampe vert est allumée.
- ◆ Le chargement des pommes sur la bande du convoyeur est fait manuellement. Il y a deux types (qualités) des pommes, une pomme jaune et l'autre rouge. Pour détecter les deux couleurs on a installé deux relais notés Relais-jaune et Relais-rouge (Relais-jaune pour la pomme jaune et Relais-rouge pour la pomme rouge) Ceux qui attendent que la Raspberry effectué le traitement d'image et émette un signal vers l'API à travers le relais dépende de chaque couleur.
- ◆ Lorsque le Raspberry activer le Relais-jaune pour une pomme jaune la tige du vérin 1 doit sortir après certain temps et rentrer simultanément juste pour injecter la pomme jaune dans la caisse 1. Si le Relais-rouge est active donc le vérin 2 sortir pour injecter la pomme rouge dans la caisse 2.
- ◆ Si le capteur ne détectent aucune pomme ainsi que les relais son désactiver pendant 20s une lampe orange clignoter avec une fréquence de 1Hz indiquée qu'il y a un manque des produits.
- ◆ Un système de surveillance du comportement de la partie opérative ou de la partie commande est installé. S'il y a un défaut dans les deux parties, un arrêt d'urgence est installé pour couper l'énergie sur ces dernies et déclenché un buzzer.

IV.3. Schéma fonctionnel du système

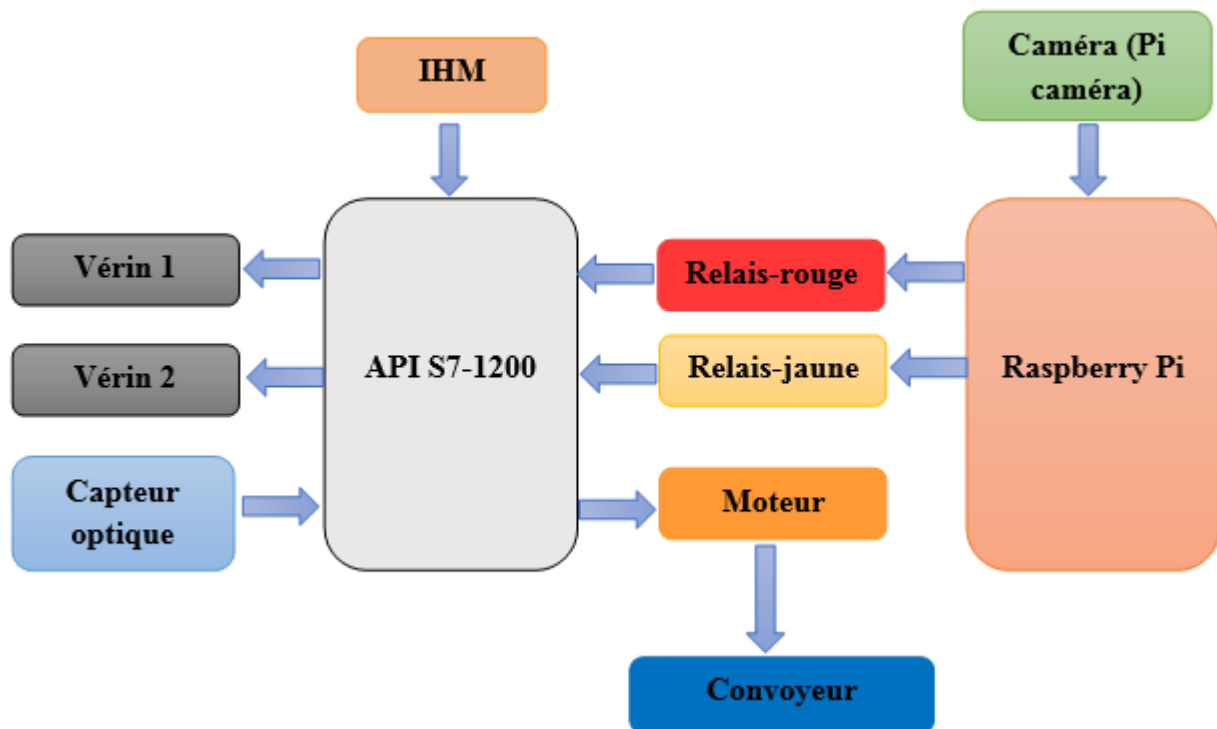


Figure IV. 1 : Schéma structurel de la machine trieuse

- Construire le schéma de principe pour une idée est une partie nécessaire afin d'examiner les opérations globales des solutions définies.
- La figure 1 ci-dessus montre le bloc du projet. Ce système automatisé est conçu pour surmonter les problèmes des techniques manuelles. Ici, le modèle matériel est conçu qui contient :
 - ✓ Un convoyeur auquel le moteur à courant alternatif est connecté.
 - ✓ Un ensemble de classement contenant deux vérins pneumatiques séparant les deux catégories traitées en deux boîtes.
 - ✓ Une caméra (Pi camera).
 - ✓ Un capteur optique.
 - ✓ Une carte Raspberry pi (model Pi 2 B)
 - ✓ Un automate Programmable Industriel (Siemens S7-1200)
 - ✓ Un écran IHM (SK-102HS)
 - ✓ Deux relais 5v sont connectés à l'API comme entrées
- La carte Raspberry Pi 2 est reliée aux deux relais (relais-jaune et relais-rouge) via les ports **GPIO** (General Purpose Input/Output, littéralement Entrée-sortie à usage général)

- L'automate est relié aux différents éléments de la partie opérative via les contacts E/S de l'API tel que les capteurs, les boutons poussoirs, le contacteur, le buzzer...etc.
- L'IHM est lié à l'automate, cette liaison permet à l'écran de recevoir des informations depuis l'automate et donner des ordres à ce dernier via le bus de communication Ethernet (câble de connexion RJ45).

L'objectif est de développer un système complet de détection de couleur avant analyse de qualité et classement des fruits par image numérique. L'ensemble du système sera soumis à une analyse en temps réel en tant que soumission possible.

IV.4. Traitement d'image

IV.4.1. Introduction à l'espace des couleurs (conversion de RVB en HSV)

La vision des couleurs peut être traitée à l'aide de l'espace colorimétrique RVB ou de l'espace colorimétrique HSV. L'espace colorimétrique RVB décrit les couleurs en termes de quantité de rouge, de vert et de bleu présent. L'espace colorimétrique HSV décrit les couleurs en termes de teinte(en anglais : Hue), de saturation et de valeur.

Dans les situations où la description des couleurs joue un rôle intégral, le modèle de couleur HSV est souvent préféré au modèle RVB. Le modèle HSV décrit les couleurs de la même manière que l'œil humain a tendance à percevoir la couleur. RVB définit la couleur en termes de combinaison de couleurs primaires, alors que HSV décrit la couleur en utilisant des comparaisons plus familières telles que la couleur, l'éclat et la luminosité.

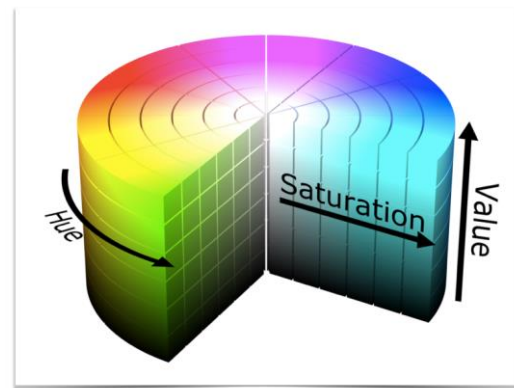


Figure IV. 2 : Cylindre de HSV

La figure ci-dessus illustre comment la teinte, la saturation et la valeur sont définies.

- ❖ La teinte représente le type de couleur. Il peut être décrit en termes d'angle sur le cercle ci-dessus. Bien qu'un cercle contienne 360 degrés de rotation, la valeur de la teinte est normalisée dans une plage de 0 à 255, 0 étant rouge.
- ❖ La saturation représente l'éclat de la couleur. Sa valeur est comprise entre 0 et 255. Plus la valeur de saturation est basse, plus la couleur est grise, ce qui la fait apparaître pâle.
- ❖ La valeur représente la luminosité de la couleur. Il va de 0 à 255, 0 étant complètement sombre et 255 étant totalement clair.

- ❖ Le blanc a une valeur HSV de 0-255, 0-255, 255. Le noir a une valeur HSV de 0-255, 0-255, 0. La description dominante du noir et blanc est le terme valeur. La teinte et le niveau de saturation ne font pas de différence lorsque la valeur est au niveau d'intensité max ou min.

La caméra (caméra pi) couleur utilise le modèle RVB pour déterminer la couleur. Une fois que la caméra a lu ces valeurs, elles sont converties en valeurs HSV. Les valeurs HSV sont ensuite utilisées dans le code pour déterminer l'emplacement d'une couleur spécifique que nous recherchons. Les pixels sont vérifiés individuellement pour déterminer s'ils correspondent à un seuil de couleur prédéterminé.

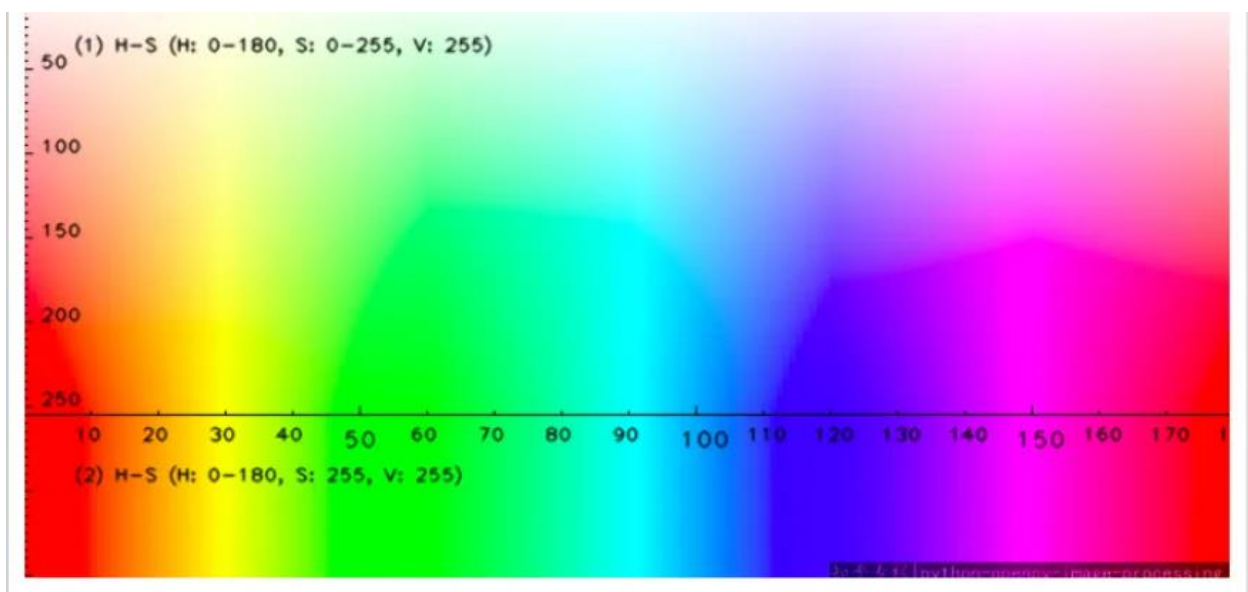


Figure IV.3 : Gamme de teinte

IV.4.2. Organigramme du traitement d'image

La caméra est configurée pour capturer 60 images par seconde, ce qui est suffisant pour que le système fonctionne en temps réel. Chaque image capturée est traitée selon les étapes d'organigramme ci-dessous :

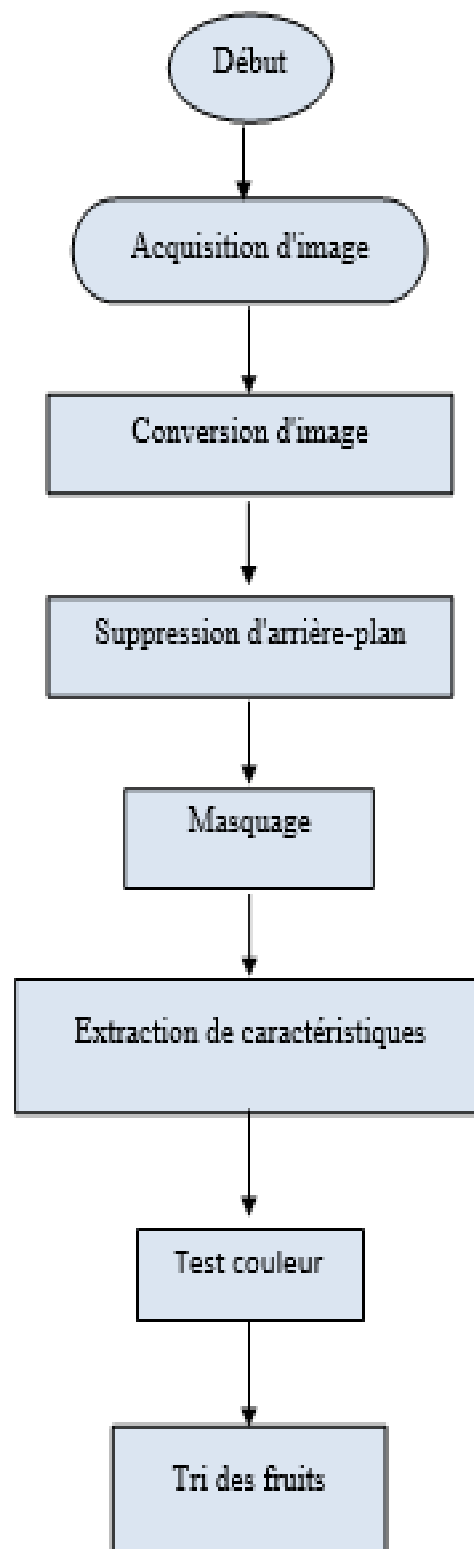


Figure IV.4 : Organigramme du traitement d'image

IV.4.3. Méthodes et matériaux de traitement

a. L'acquisition des données

L'image couleur du fruit pomme est l'entrée du système conçu. On constate que les images d'un même objet varient en fonction de la source de lumière, de l'intensité de la lumière, de l'arrière-plan, de la distance par rapport à l'appareil photo et des paramètres de l'appareil photo, etc. Pour maintenir la cohérence de tous les échantillons de fruits, une chambre d'imagerie est conçue comme indiqué sur la figure IV.5. Une boîte en plastique cubique de couleur noir est utilisée comme chambre d'imagerie. La base de la boîte est enduite de papier blanc, pour réduire les reflets de la base. La surface intérieure de la boîte est recouverte d'un matériau réfléchissant la lumière. Les LED montées sur le dessus sont utilisées comme source de lumière à l'intérieur du boîtier d'imagerie ; qui est ensuite recouvert de feuille de diffusion. L'appareil photo utilisé est l'appareil PI camera model OV5647, avec la distance entre un fruit et le camera est de 18 cm pour toutes les images prises.

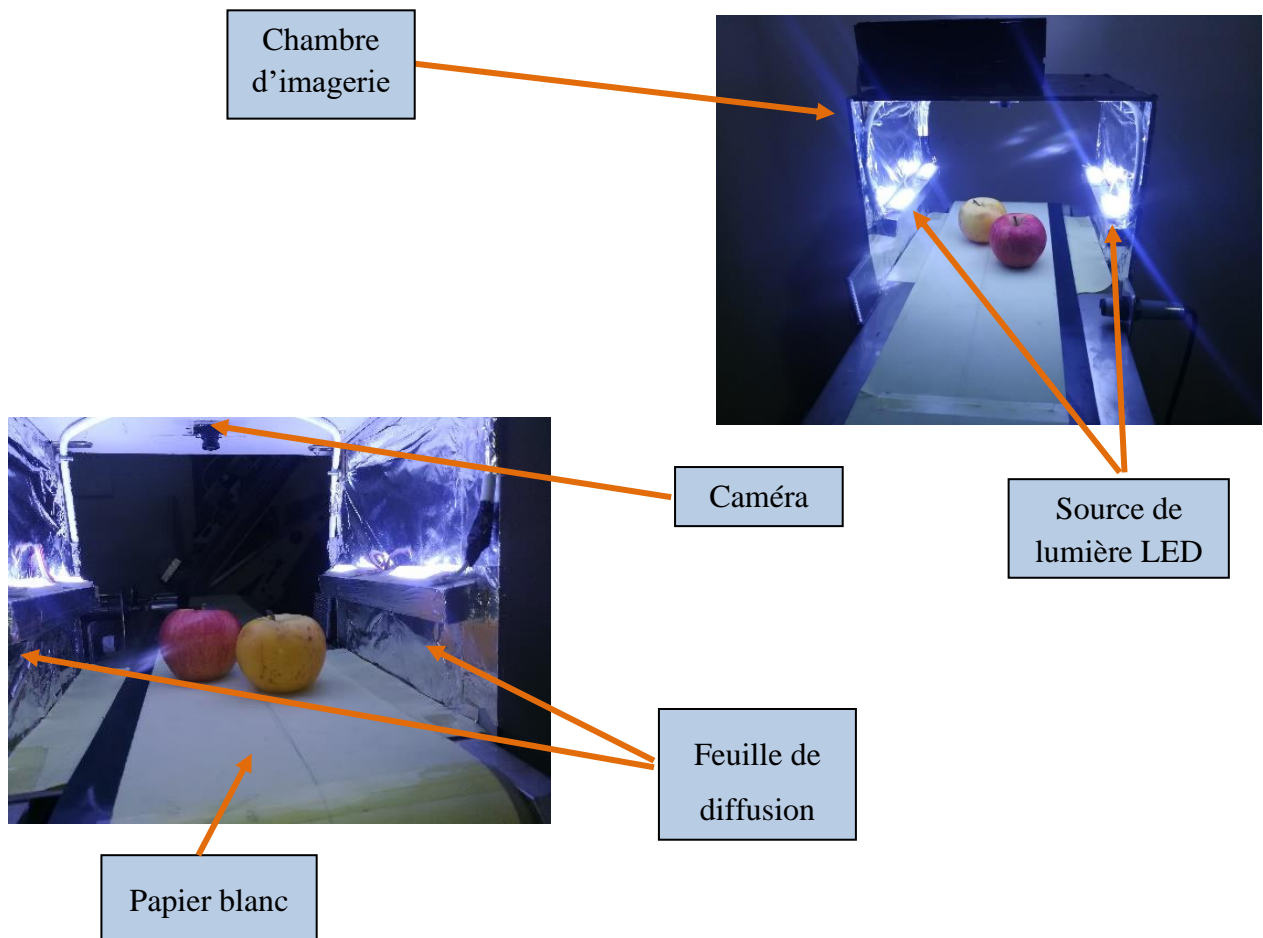


Figure IV.5 : Composants de la chambre d'imagerie

IV.4.4. Convertir l'image capturée

a. Langage de la programmation

Python est un langage de programmation à usage général lancé par *Guido van Rossum* qui est devenu très rapidement populaire, principalement en raison de sa simplicité et de sa lisibilité du code. Il permet au programmeur d'exprimer des idées en moins de lignes de code sans réduire la lisibilité.



En utilisant ce dernier, nous avons converti l'image de RVB en HSV, nous avons obtenu le résultat suivant :

❖ L'image de pomme jaune :



Figure IV.6 : Image RVB de la pomme jaune

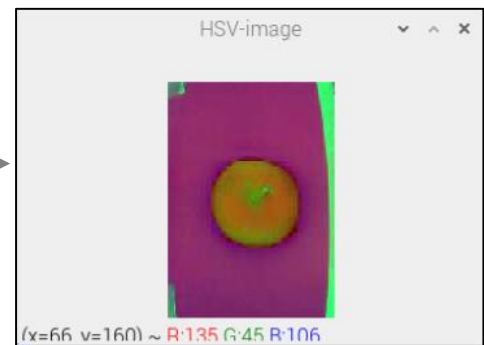


Figure IV.7 : Image HSV de la pomme jaune

❖ L'image de pomme rouge :

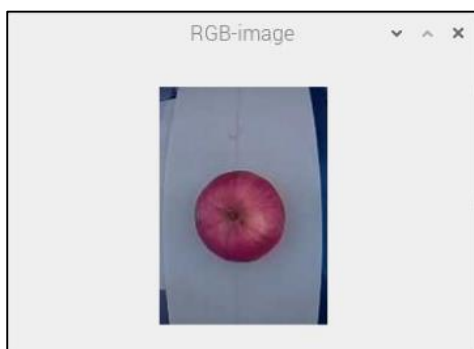


Figure IV.8 : Image HSV de la pomme rouge



Figure IV.9 : Image RVB de la pomme rouge

b. Le masquage

Le masquage est un filtrage. Il consiste à déplacer le masque de filtre d'un point à un autre dans une image. Pour une image floue, les intensités de pixels sont aussi lisses que possible. Le masquage consiste à séparer une image de son arrière-plan et à placer l'image sur un autre arrière-plan. Les pixels sont ajustés géométriquement pour correspondre à la projection cartographique souhaitée. Pour augmenter la netteté d'une image, le masquage est utilisé. Après le traitement nous avons obtenu le résultat suivant :

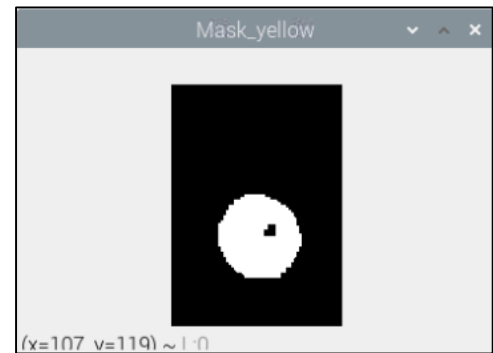
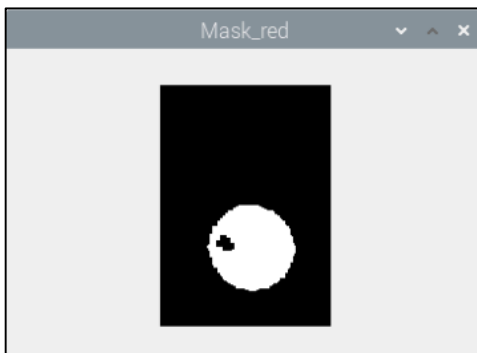


Figure IV. 10 : Masque d'image de la pomme rouge

Figure IV. 11 : Masque d'image de la pomme jaune

c. Bitwise Operations

Les opérations Bitwise sont utilisées dans la manipulation d'image et utilisées pour extraire des parties essentielles de l'image. Les opérations au niveau du bit utilisées sont : **AND**, **OR**, **XOR**, **NOT**. En outre, les opérations Bitwise aident au masquage d'image. La création d'image peut être activée à l'aide de ces opérations. Ces opérations peuvent être utiles pour améliorer les propriétés des images d'entrée.

Remarque : les opérations au niveau du bit doivent être appliquées aux images d'entrée de mêmes dimensions.

Nous avons utilisé l'opération **Bitwise AND**, et nous avons obtenu ce résultat suivant



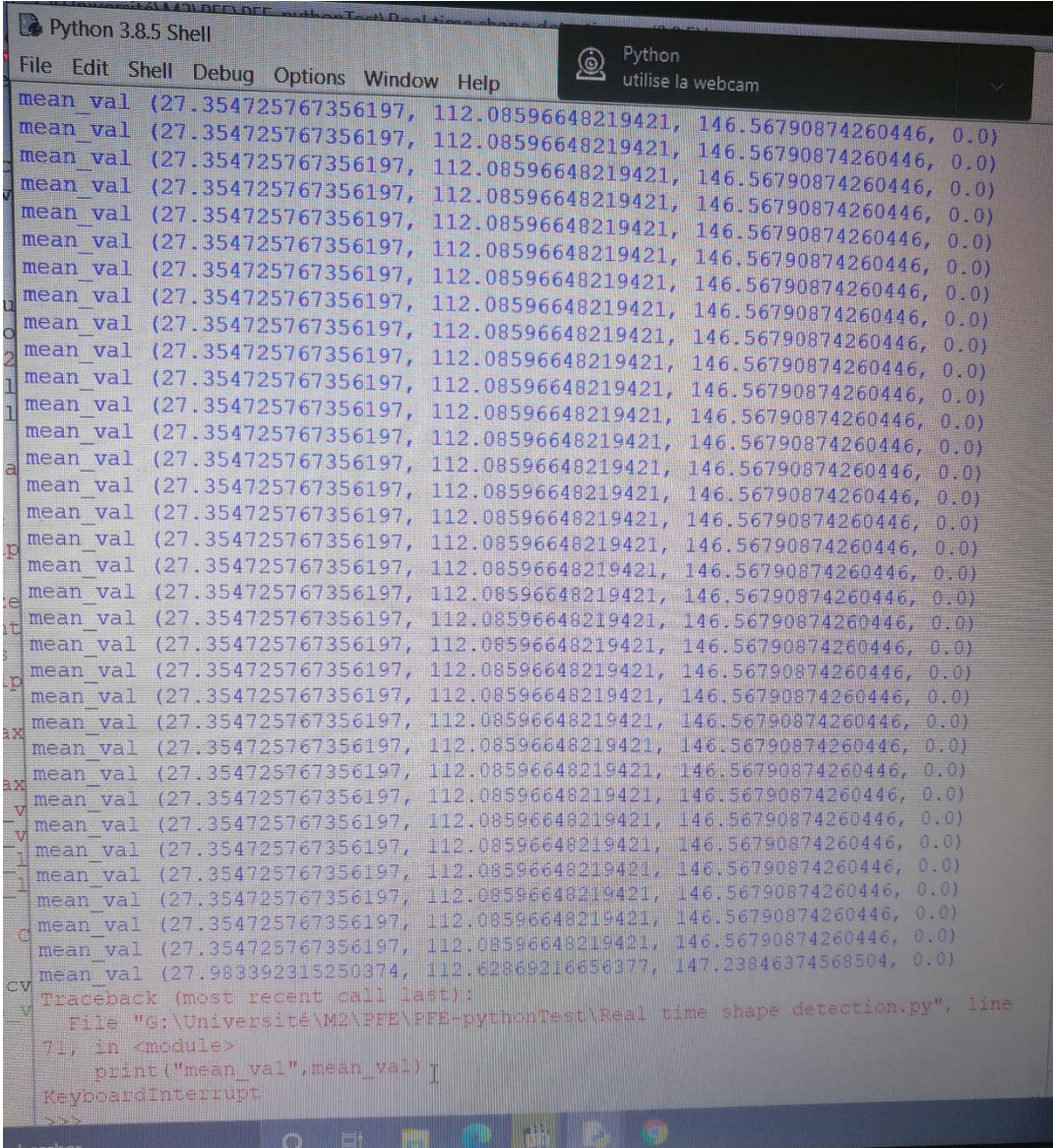
Figure IV. 12 : Masque + HSV image du pomme rouge



Figure IV. 13 : Masque + HSV image du pomme jaune

d. Utilisation de la couleur moyenne comme caractéristique

Le calcul de la couleur moyenne de chaque canal de couleur permet de distinguer chaque catégorie de fruits en fonction de l'intensité de couleur prédominante. Ici, nous avons utilisé le calcul de la valeur moyenne (mean_val) comme indice pour distinguer chaque catégorie. Les résultats apparaissent également dans les images ci-dessous :



```
Python 3.8.5 Shell
File Edit Shell Debug Options Window Help
Python
utilise la webcam

mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.354725767356197, 112.08596648219421, 146.56790874260446, 0.0)
mean_val (27.983392315250374, 112.62869216656377, 147.23846374568504, 0.0)
Traceback (most recent call last):
  File "G:\Université\M2\PFPE\PFPE-pythonTest\Real time shape detection.py", line
  71, in <module>
    print("mean_val",mean_val)
KeyboardInterrupt
>>>
```

Figure IV.14 : Les résultats de la valeur moyenne de la couleur jaune de la pomme

Les figures ci-dessous représentent des photos de différents équipements utilisés de notre machine



Figure IV.15 : Vue de dessus du convoyeur et de la chambre d'imagerie

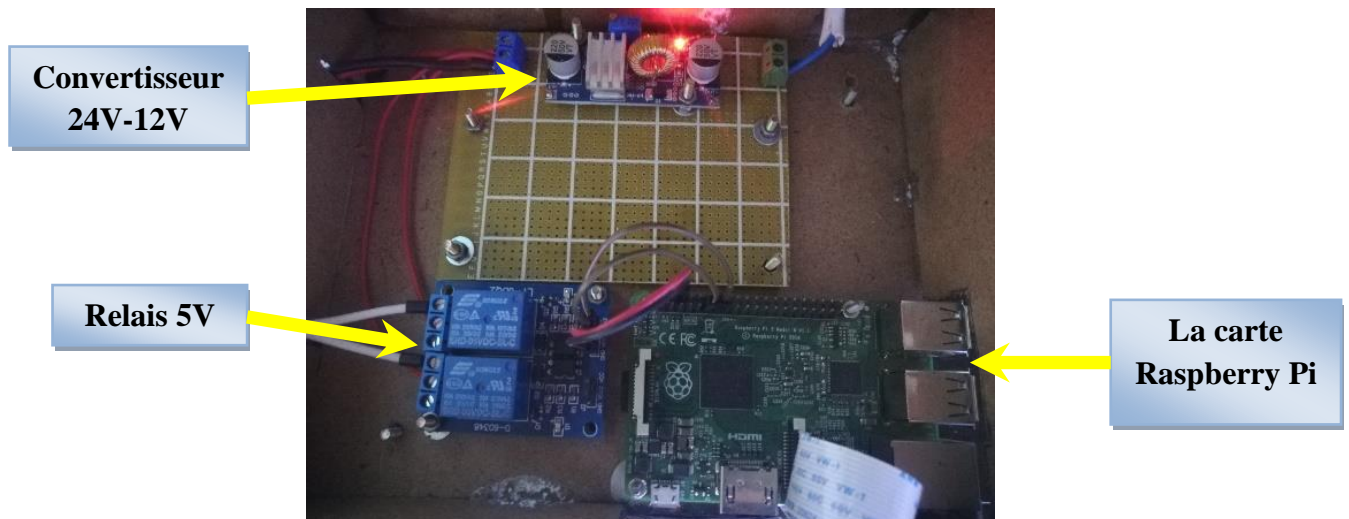


Figure IV.16 : Convertisseur Buck et la liaison du carte Raspberry Pi aux relais

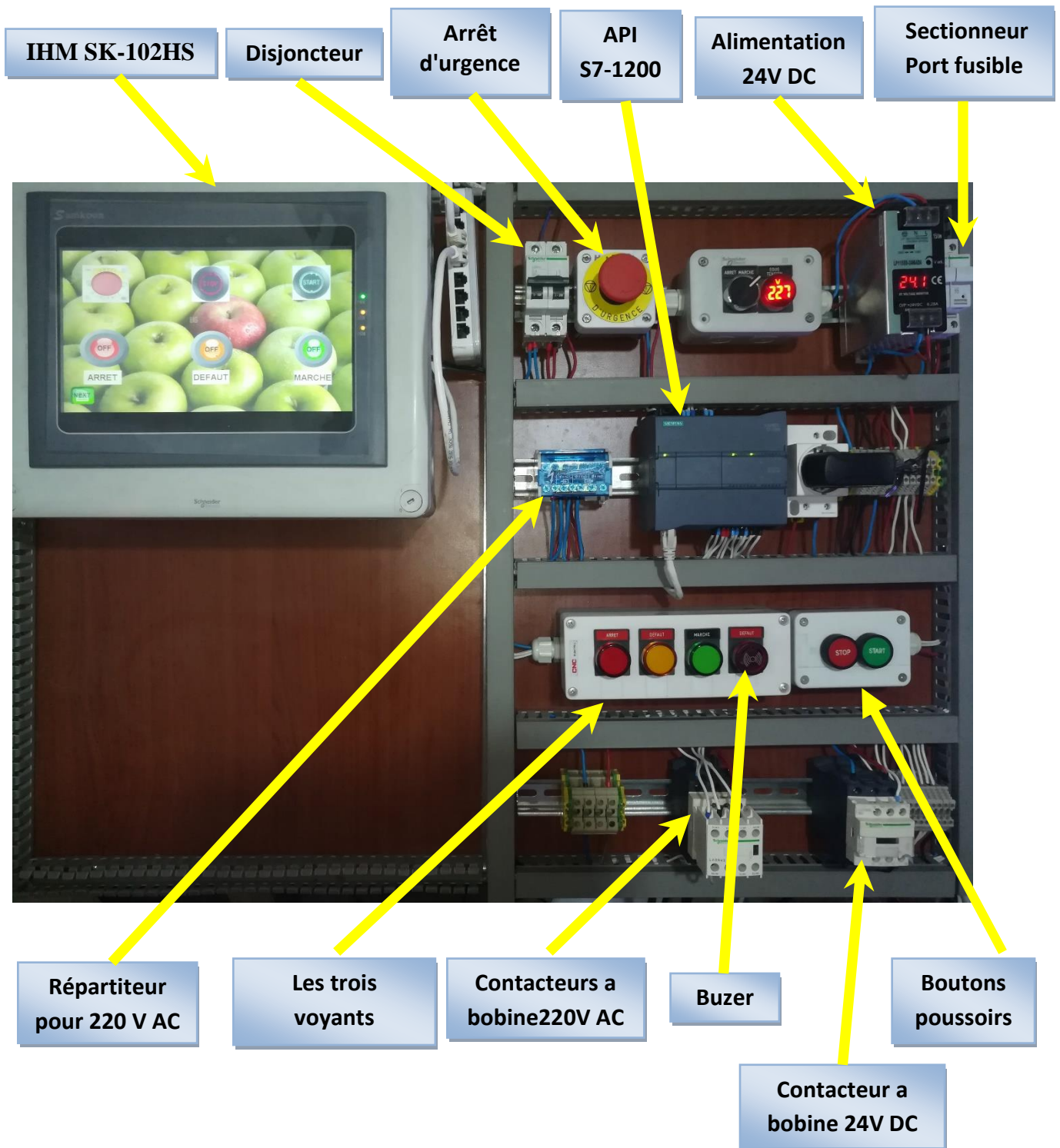


Figure IV.17 : L'armoire électrique avec IHM

IV.5. Partie de programmation

IV.5.1. Programmation d'API par TIA Portal

a. Table des variables E/S

Dans cette table on a représenté toutes les variables et leurs adresses utilisées de noter programme.

	Nom	Table des variables	Type de données	Adresse	Réma...	Visibl...	Acces...	Commentaire
1	BPM	Table de variables s...	Bool	%I0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	BPA	Table de variables s...	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	CO1	Table de variables s...	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	AU	Table de variables s...	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5	RELAY Jaune	Table de variables s...	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	RELAY Rouge	Table de variables s...	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	LV	Table de variables s...	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	LO	Table de variables s...	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	LR	Table de variables s...	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	BU	Table de variables s...	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	KM1	Table de variables s...	Bool	%Q0.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	verin 1 sortie	Table de variables s...	Bool	%Q0.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	verin 1 entré	Table de variables s...	Bool	%Q0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14	maintien	Table de variables s...	Bool	%M7.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15	Clock_Byte	Table de variables s...	Byte	%MB50	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16	Clock_10Hz	Table de variables s...	Bool	%M50.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
17	Clock_5Hz	Table de variables s...	Bool	%M50.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
18	Clock_2.5Hz	Table de variables s...	Bool	%M50.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
19	Clock_2Hz	Table de variables s...	Bool	%M50.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
20	Clock_1.25Hz	Table de variables s...	Bool	%M50.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
21	Clock_1Hz	Table de variables s...	Bool	%M50.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
22	Clock_0.625Hz	Table de variables s...	Bool	%M50.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
23	Clock_0.5Hz	Table de variables s...	Bool	%M50.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
24	fin de temp1	Table de variables s...	Bool	%M8.4	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
25	fin de temp2	Table de variables s...	Bool	%M8.5	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
26	verin 2 sortie	Table de variables s...	Bool	%Q0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
27	verin 2 entré	Table de variables s...	Bool	%Q1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
28	BPM(1)	Table de variables s...	Bool	%M8.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
29	BPA(1)	Table de variables s...	Bool	%M8.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
30	AU(1)	Table de variables s...	Bool	%M9.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
31	Jaune compteur	Table de variabl...	Int	%QW9	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
32	Rouge compteur	Table de variables s...	Int	%QW12	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
33	COMPTAGE REMI 0	Table de variables s...	Bool	%M9.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure IV.18 : Table des variables E/S

b. Programme du Block d'Organisation BO

La figure ci-dessous représente le premier réseau, ce réseau pour mise en marche avec auto-maintien.

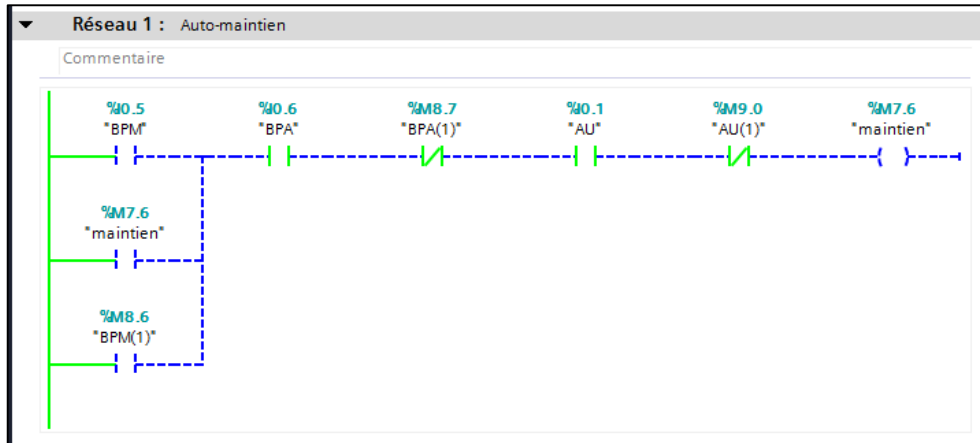


Figure IV.19 : Réseau pour mise en marche avec auto-maintien

Le deuxième réseau représente le démarrage du moteur.

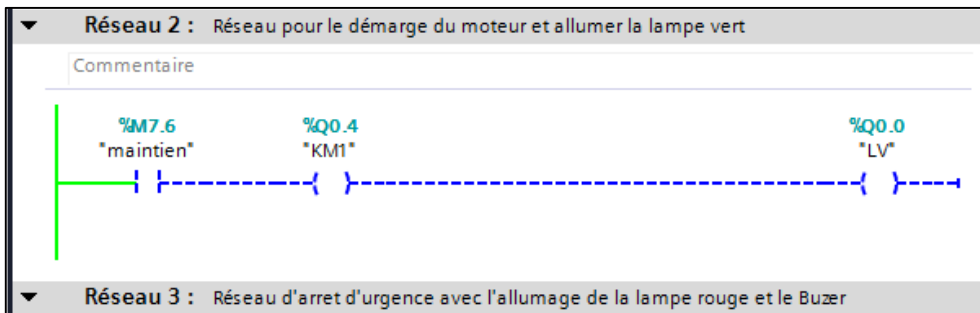


Figure IV.20 : Réseau pour la commande des trois vitesses

Ce réseau représente la gestion du bouton d'arrêt d'urgence.

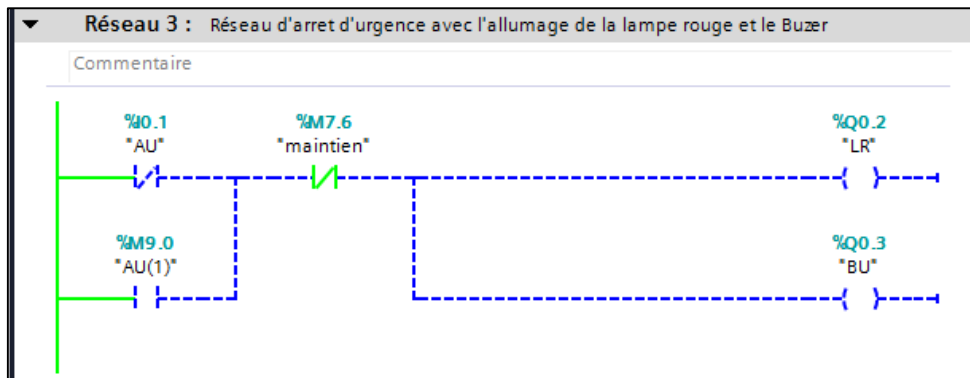


Figure IV.21 : Réseau d'arrêt d'urgence

La figure ci-dessous représente le réseau pour sortir la tige du vérin 1.

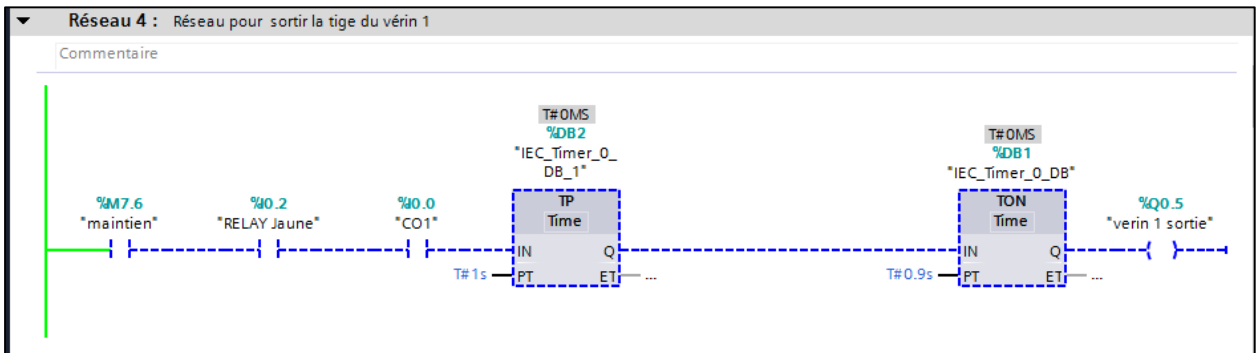


Figure IV.22 : Réseau pour sortir la tige du vérin 1

La figure ci-dessous représente la temporisation de la tige sortant du vérin 1 pendant 300 ms

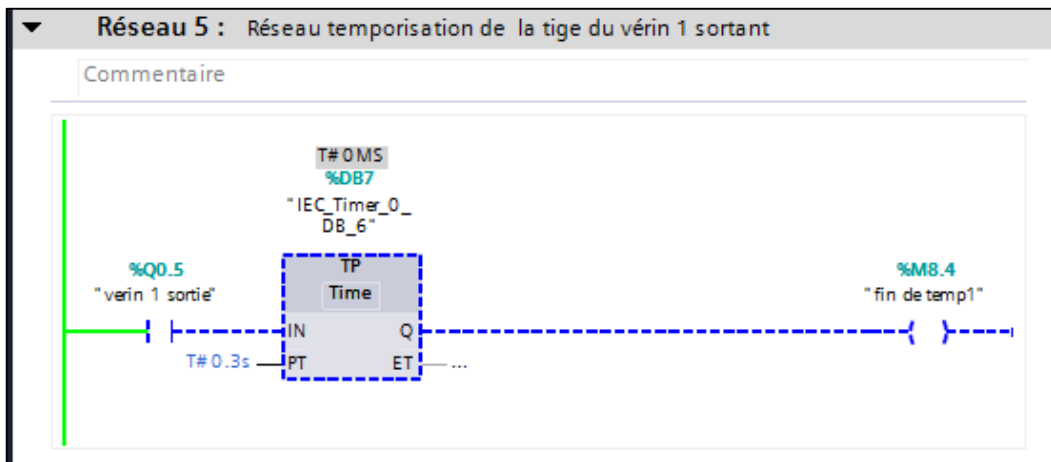


Figure IV.23 : Réseau temporisation de la tige sortant du vérin 1 pendant 300 ms

La figure ci-dessous représente le réseau pour la commande de rentrer la tige du vérin 1

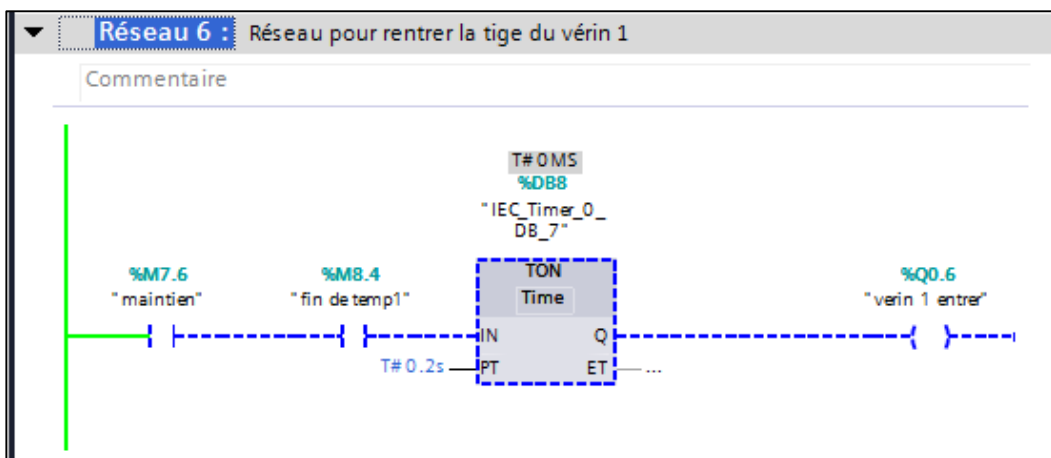


Figure IV.24 : Réseau pour rentrer la tige du vérin1

La figure ci-dessous représente le réseau pour sortir la tige du vérin 2.

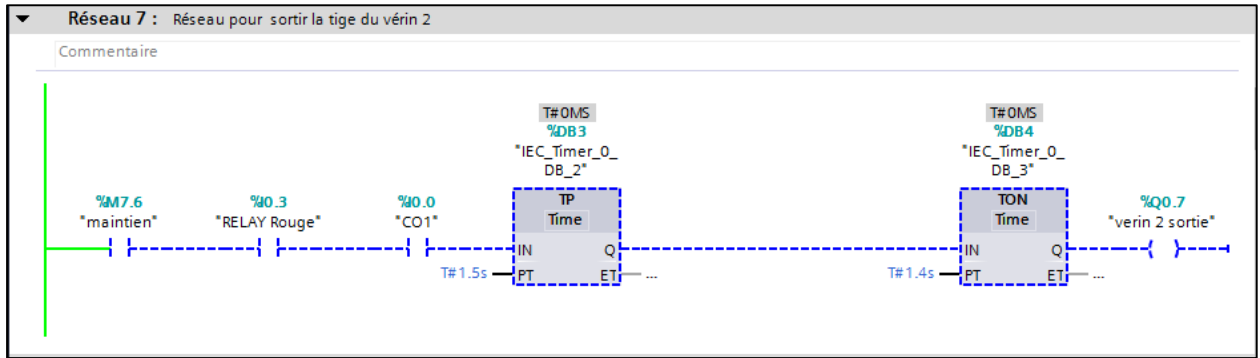


Figure IV.25 : Réseau pour sortir la tige du vérin 2

La figure ci-dessous représente la temporisation de la tige sortant du vérin 2 pendant 300 ms.

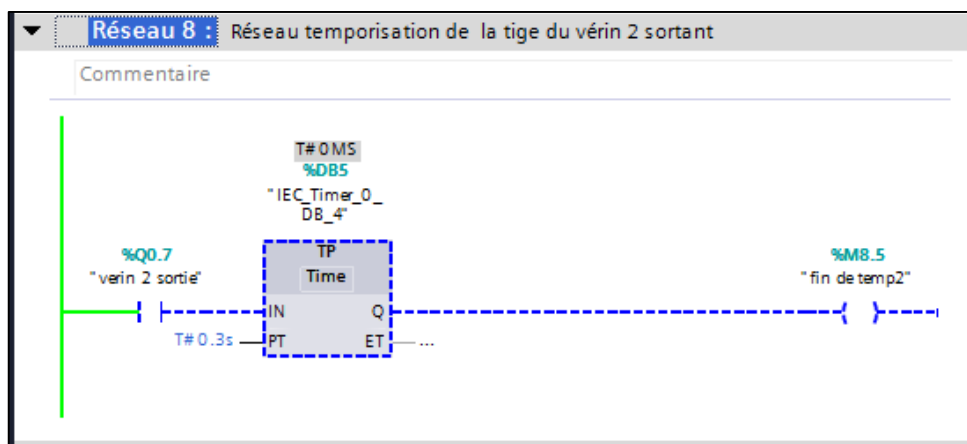


Figure IV.26 : La figure ci-dessous représente la temporisation de la tige sortant du vérin 2 pendant 300 ms

La figure ci-dessous représente le réseau pour la commande de rentrer la tige du vérin 2

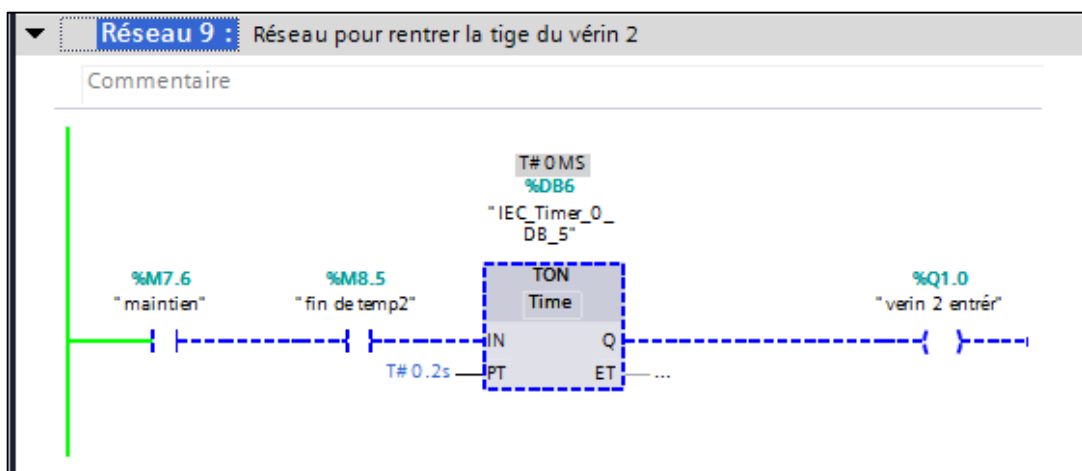


Figure IV.27 : Réseau pour rentrer la tige du vérin 2

La figure ci-dessous pour gérer la manque de produit

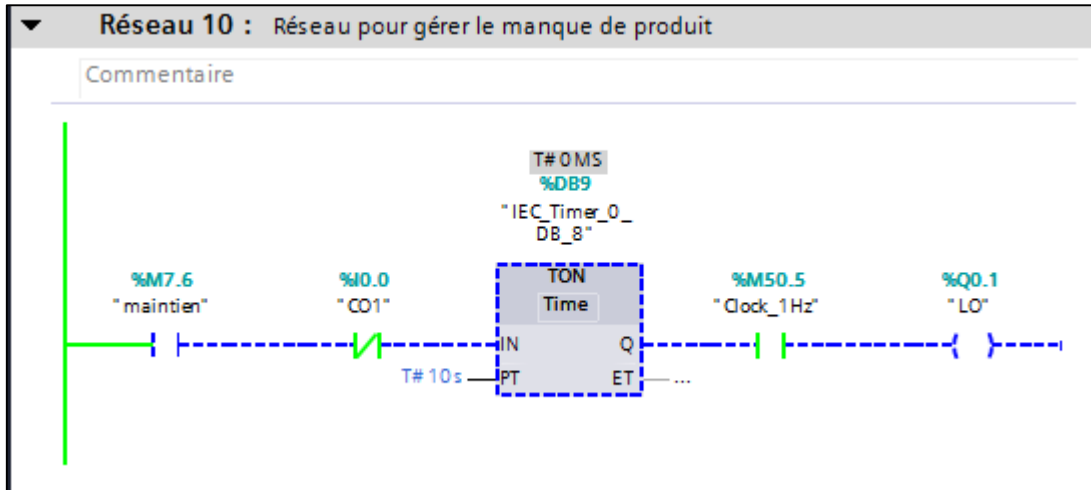


Figure IV.28 : Réseau pour gérer le manque de produit

La figure ci-dessous pour le comptage des pommes jaunes

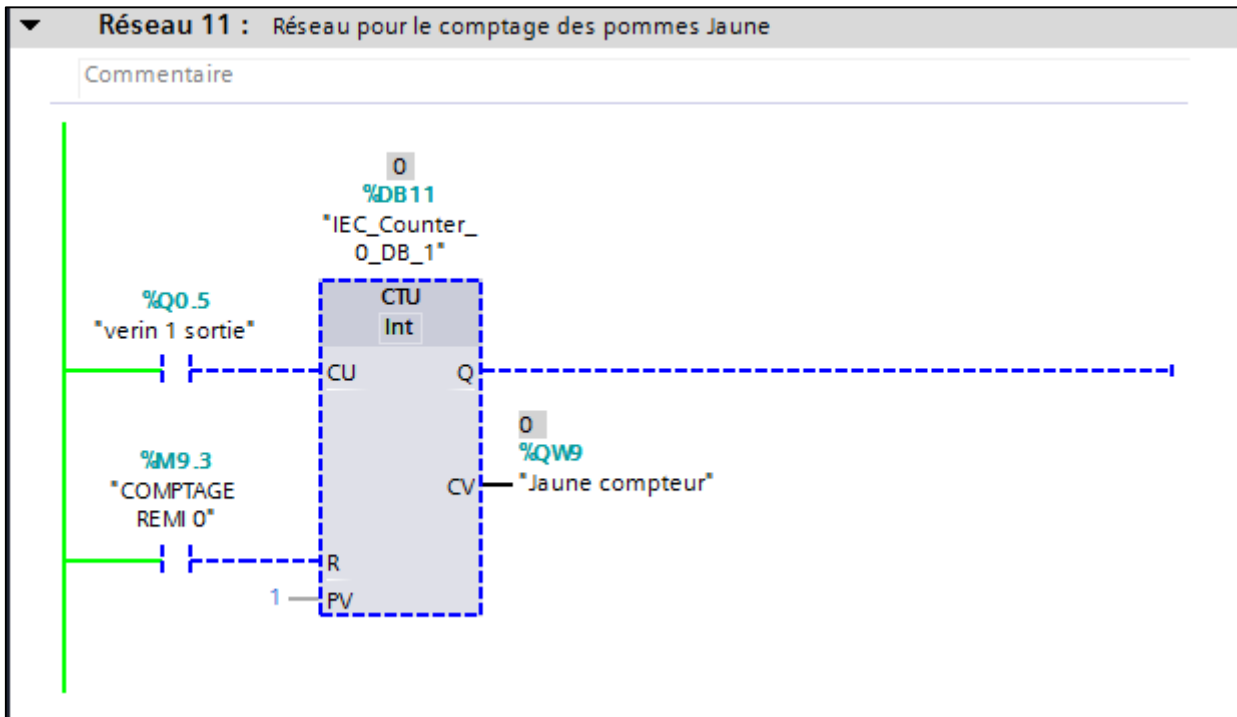


Figure IV.29 : Réseau pour comptage les pommes jaunes

La figure ci-dessous pour le comptage des pommes jaunes

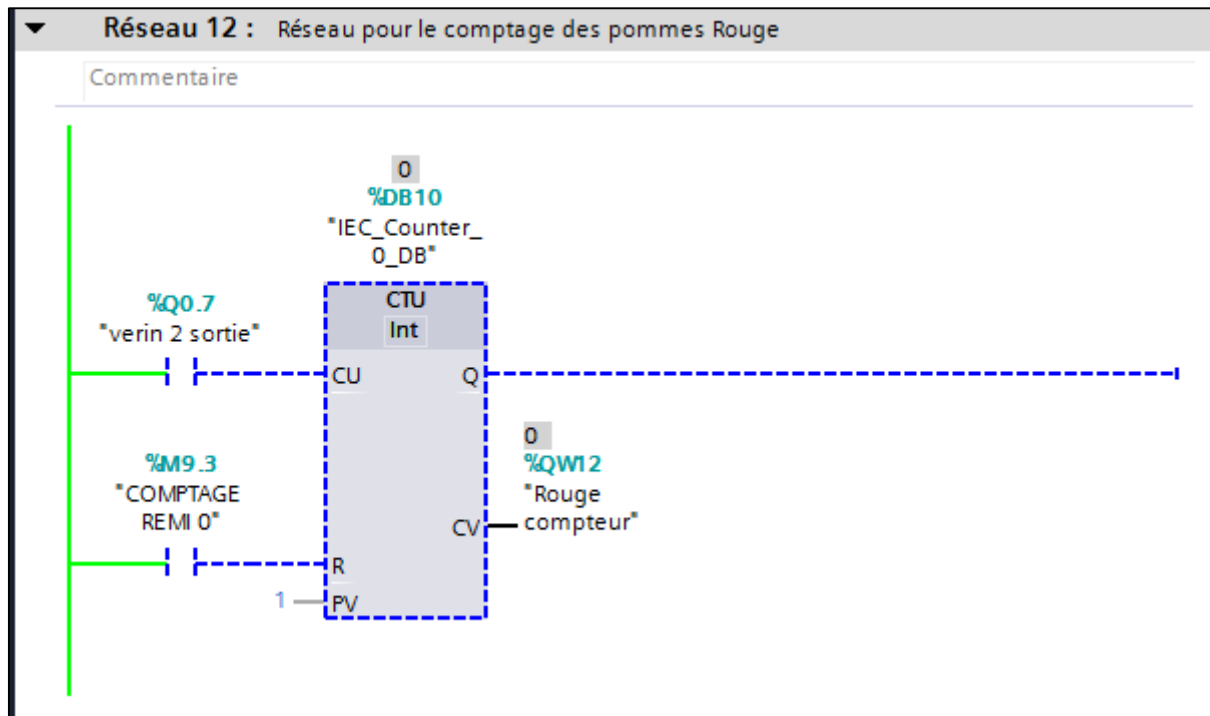


Figure IV.30 : Réseau pour comptage les pommes rouges

IV.5.2. Programmation de l'IHM

a. Logiciel « SKTOOL »

SKTOOL est un logiciel d'édition de configuration de Samkoon IHM, SKTOOL prend en charge les pilotes de communication PLC des principaux fabricants mondiaux, tels que Samkoon, Siemens, Mitsubishi, Fujitsu, Panasonic, Schneider...etc. Le logiciel de configuration SKTOOL est un système de développement avec un environnement de développement intégré et de nombreuses fonctions avancées. L'interface de développement et les domaines fonctionnels sont illustrés dans la figure suivant :



Figure IV.31 : L'interface de logiciel SKTOOL

b. Programme IHM dans SAKTOOL

Pour programmer IHM nous avons utilisé plusieurs éléments pour commander et visualiser l'état de la machine trieuse, Notre programme IHM contient deux fenêtres :

- La première fenêtre est représentée une page d'accueil, la page de garde de notre projet.
 - ☑ Deux touches pour démarrer et arrêter le moteur du convoyeur
 - ☑ Une touche pour l'arrêt d'urgence
 - ☑ Trois voyants indiquant l'état du convoyeur (Marche, Arrêt, Manque des pièces)
 - ☑ Une touche pour passer à la deuxième fenêtre.

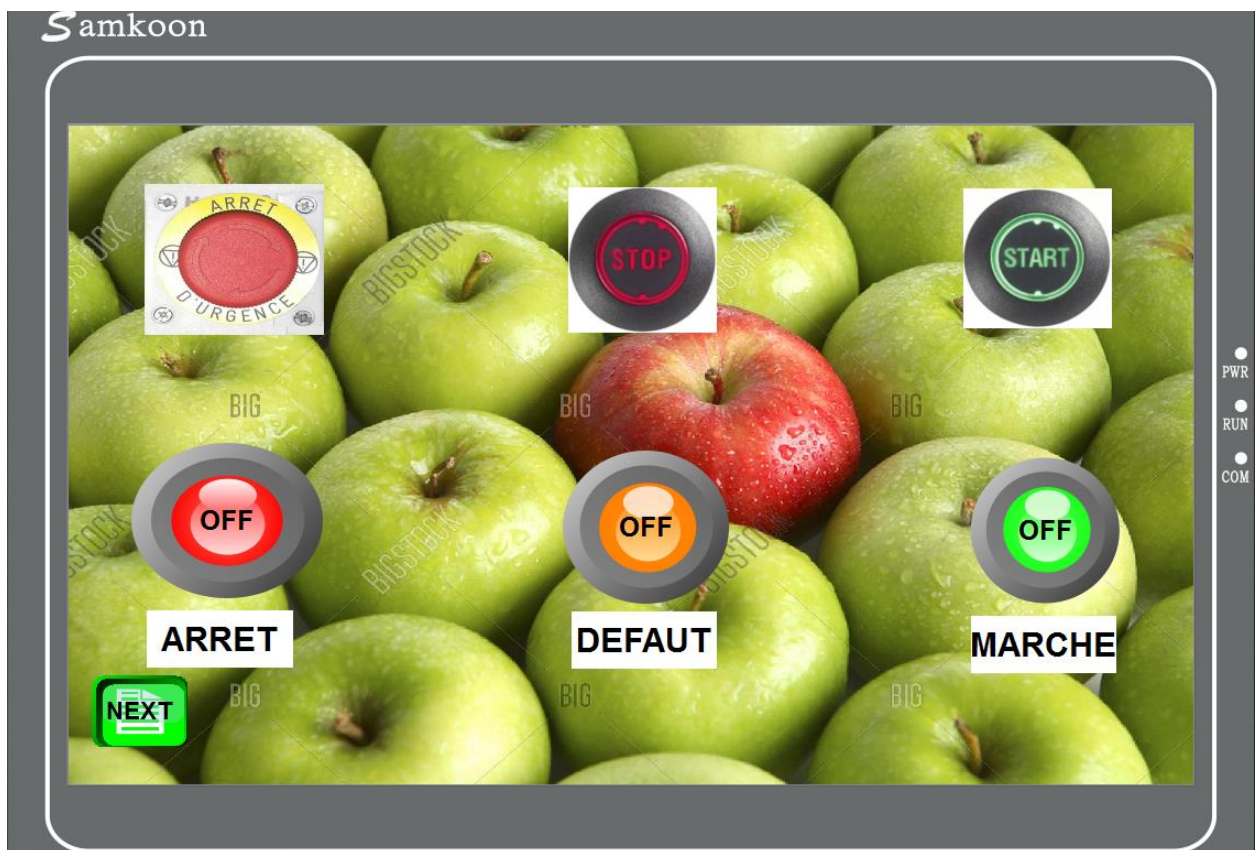


Figure IV.32 : La première fenêtre de l'IHM

- La deuxième fenêtre du l'IHM contient :
 - ☑ Un compteur pour les pommes jaunes.
 - ☑ Un compteur pour les pommes rouges.
 - ☑ Une touche pour la remise à zéro des compteurs.
 - ☑ Une touche pour revenir à la première page.



Figure IV.33 : La deuxième fenêtre de l'IHM.

IV.6. Conclusion

Dans ce chapitre nous avons présenté la conception et réalisation de notre machine trieuse, dans la première partie on a parlé sur le chier de charge et les étapes principales impliquées pour réaliser notre objectif, dans la seconde partie on a abordé la partie du traitement d'image et les résultats obtenus pour traiter les pommes selon leurs couleurs. Dans le coté de programmation on a divisé cette partie en deux partie : la programmation de l'API par le logiciel TIA Portal et la programmation de l'IHM.

Conclusion générale

Ce travail présente une machine de tri d'objets capable de trier les pommes en fonction de leur couleur grâce à une vision de caméra et un microcontrôleur Raspberry Pi qui fait le traitement d'image et le système commandé par un automate programmable industriel de type S7-1200 pour contrôler le convoyeur et les actionneurs. Nous avons aussi supervisé notre système avec un écran tactile afin d'avoir un suivi en temps réel.

Le système de tri des objets basé sur la couleur a été mis en œuvre dans ce travail de projet. Le travail utilise IDE Python avec bibliothèque Open CV. Les algorithmes ont été testés et fonctionnent bien pour les 2 couleurs de pomme rouge, et jaune.

Nous nous sommes basés dans ce travail sur la technique de segmentation dans le traitement d'image pour effectuer le processus de tri, et ce travail peut être fait avec autre technique comme « machine Learning, deep Learning, feature recognition ...ect » afin d'améliorer les résultats.

Pour réaliser notre machine on a divisé le travail en quatre chapitres.

Le premier chapitre aborde les systèmes automatisés, comme l'automate programmable Siemens S7-1200, les modules de communication et d'alimentation. Dans la suite de ce chapitre on présente le microcontrôleur Raspberry et sa description et aussi l'écran tactile utilisé (IHM) de marque SAMKOON SK-102HS.

Le deuxième chapitre est consacré à la partie opérative notamment les pré-actionneurs, les actionneurs pneumatiques et électriques et les capteurs.

Le troisième chapitre sera dédié à la partie puissance (l'armoire électrique) et leurs éléments essentiels tels que les éléments de protection, systèmes de réglettes et de fixations, Un arrêt d'urgence ...etc.

Le quatrième chapitre, qui est la partie réalisation de notre projet. Cette réalisation est basée sur la commande d'une machine de tri grâce à une vision de caméra contrôlée par un Raspberry, le système automatisé et commandé par un API et supervisé avec un écran tactile.

Enfin, nous clôturons notre document avec une conclusion générale qui est dédiée à une synthèse du travail présenté.

Référence Bibliographiques

- [1] M.L. Fas, " Cours actionneurs ", Support de cours, Université Saad Dahleb de Blida 1, 2017.
- [2] J. Perrain, F. Binet, J.J. Doumer, C. Merlaud, J.P. Trichard, "Automatique et informatique industrielle", Edition Nathan, 2006.
- [3] A. Simon, "Automates programmables industriels Niveau 1", Edition l'Elan-Liège, 1991.
- [4] E. Gaucheron, " Les moteurs électriques ... pour mieux les piloter et les protéger", Collection technique Cahier technique N° 207, Schneider Electric, 2004.
- [5] G. Chateigner, M. Boës, D. Bouix, J. Vaillant, D. Verkindère, "Manuel de génie électrique", Livre DUNOD, 2005.
- [6] M. Bertrand, "Technique de l'ingénieur, automates programmables industriels", 1^{ère} Edition, 2001.
- [7] Document Siemens, "Information et formation, automatisation et entraînements, programmation niveau A", Edition Siemens AG, 2003.
- [8] Ir.H. Lecocq, " Les automates programmables, Tome I, Caractéristiques et méthodologie de programmation ", Support de cours, université de Liège, 2005.
- [9] E. Gaucheron, " Les moteurs électriques ... pour mieux les piloter et les protéger", Collection technique Cahier technique N° 207, Schneider Electric, 2004.
- [10] L. Sari, "Prise en Main de TIA Portal (Siemens)", Support de cours, Département GEII, IUT de Toulon, 2016.
- [11] J.C. Boinot, Y. Maridor, N. Croset, S. Pittet, "Commande programmable, tome II ", Support de cours, Centre professionnel du noir vaudois, 2014.
- [12] M.L. Fas, " Cours Capteurs et Chaines de Mesures ", support de cours, université Saad Dahleb de Blida 1, 2017.
- [13] Centre de Technologie Avancé, "Programmation des automates S7-300 – Introduction au logiciel TIA Portal", Fédération Wallonie-Bruxelles, 2016.
- [14] W. Bolton, "Automates Programmables Industriels", 2^{ème} édition, DUNOD, 2015.
- [15] Support de cours Siemens, "Initiation à la programmation du SIMATIC S7-1200 avec TIA Portal VX", Industry Automation and Drive Technologies, 2016.
- [16] Industry Support Siemens, "SIMATIC S7 Automate programmable S7-1200", Manuel système, 2014.
- [17] L. Docquier, "Pneumatique", Cours formatpme, 2008.
- [18] <https://www.pyimagesearch.com/category/tutorials/>
- [19] https://docs.opencv.org/master/d9/df8/tutorial_root.html
- [20] <https://py2py.com/we-already-have-rgb-so-why-we-need-hsv/>

Annexe A

Schémas de commande et de puissance

1. Schéma de commande

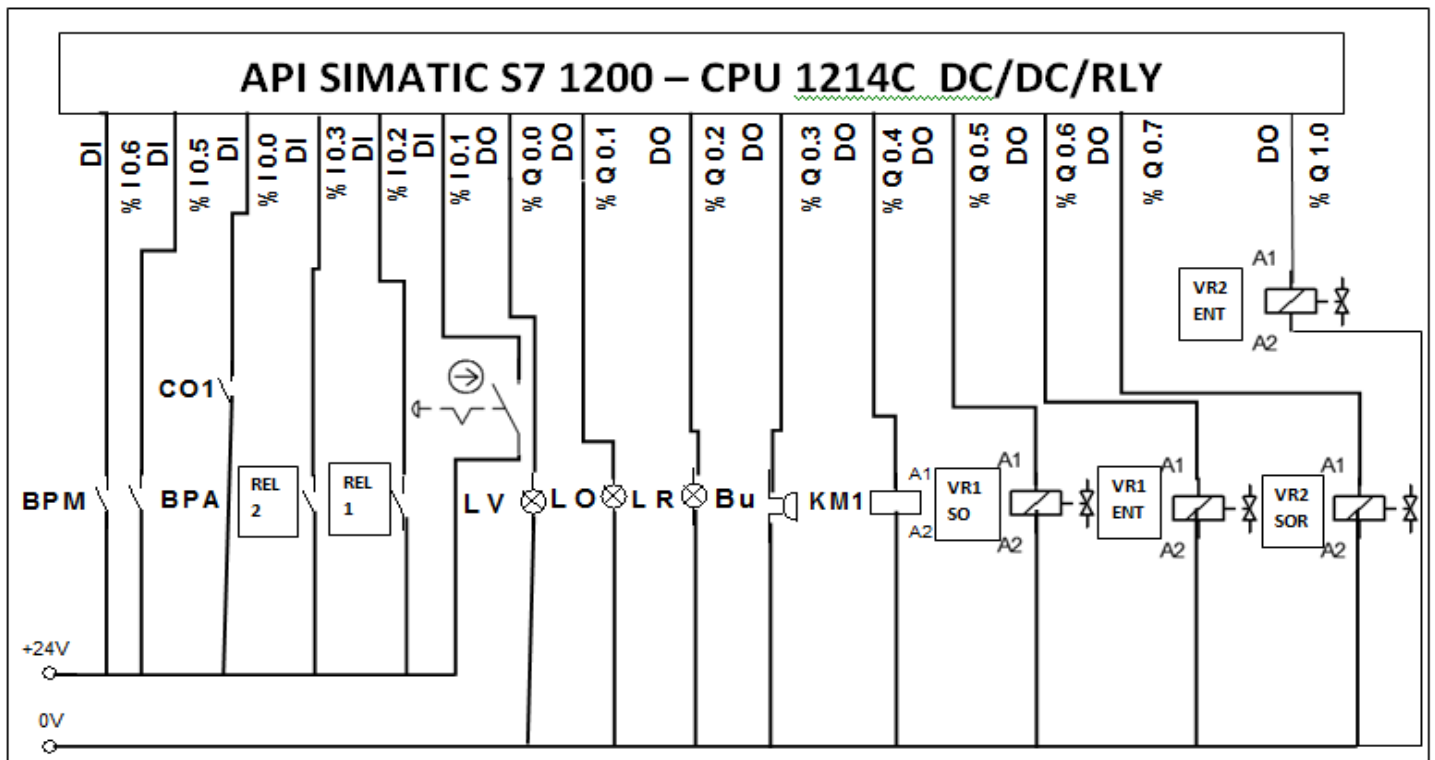


Figure A.1 : Schéma de commande

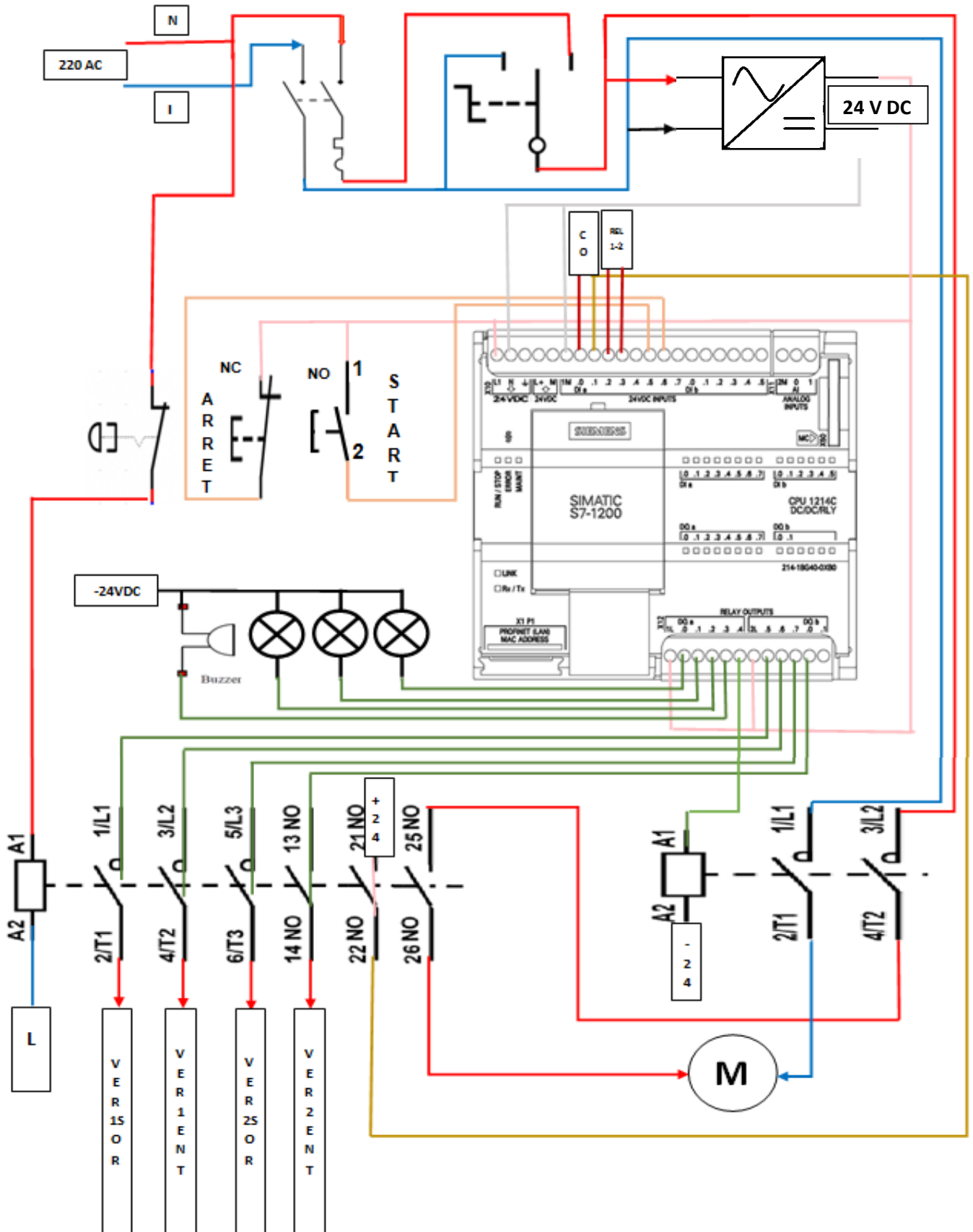


Figure A.2 : Schéma de puissance

Annexe B

TIA Portal

I. Logiciel de programmation TIA Portal

Pour programmer notre partie de commande (API S7-1200), on a utilisé le logiciel de Siemens TIA Portal V13 [10].

I.1. Présentation du logiciel TIA Portal V13

La plate-forme TIA Portal (TotallyIntegrated AutomationPortal) est la dernière évolution des logiciels de travail SIEMENS qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré, dans un seul logiciel cette plate-forme regroupe la programmation des différents dispositifs d'une installation. On peut donc programmer et configurer, en plus de l'automate, les dispositifs HMI les variateurs...etc.

Le logiciel STEP 7 Professional (TIA Portal V13) est l'outil de programmation des nouveaux automates comme [13] : SIMATIC S7-1500, S7-1200, S7-400, S7- 300

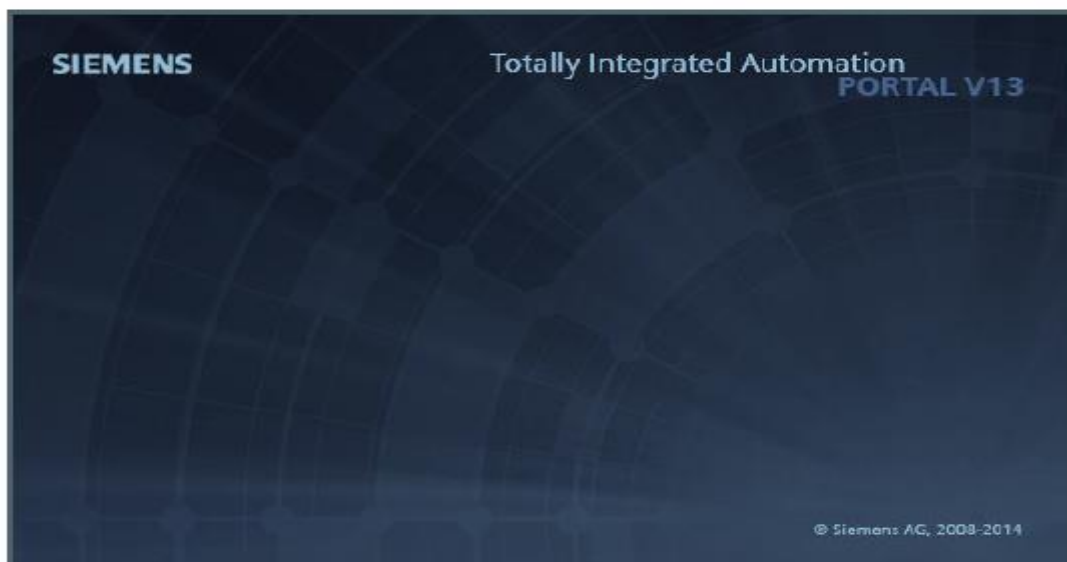


Figure 1 : Logo de TIA Portal V13

I.2. Création d'un nouveau projet

- Cliquez sur « **Nouveau projet** »
- Complétez les données nécessaires

- ☑ Nom du projet : **FAS**
 - ☑ Chemin : C:\Users\pc click\Desktop\projet tia
 - ☑ Auteur : Par défaut prenant le nom de Pc
 - ☑ Commentaire : un petit commentaire sur le programme
- Cliquez sur le bouton « **Créer** »

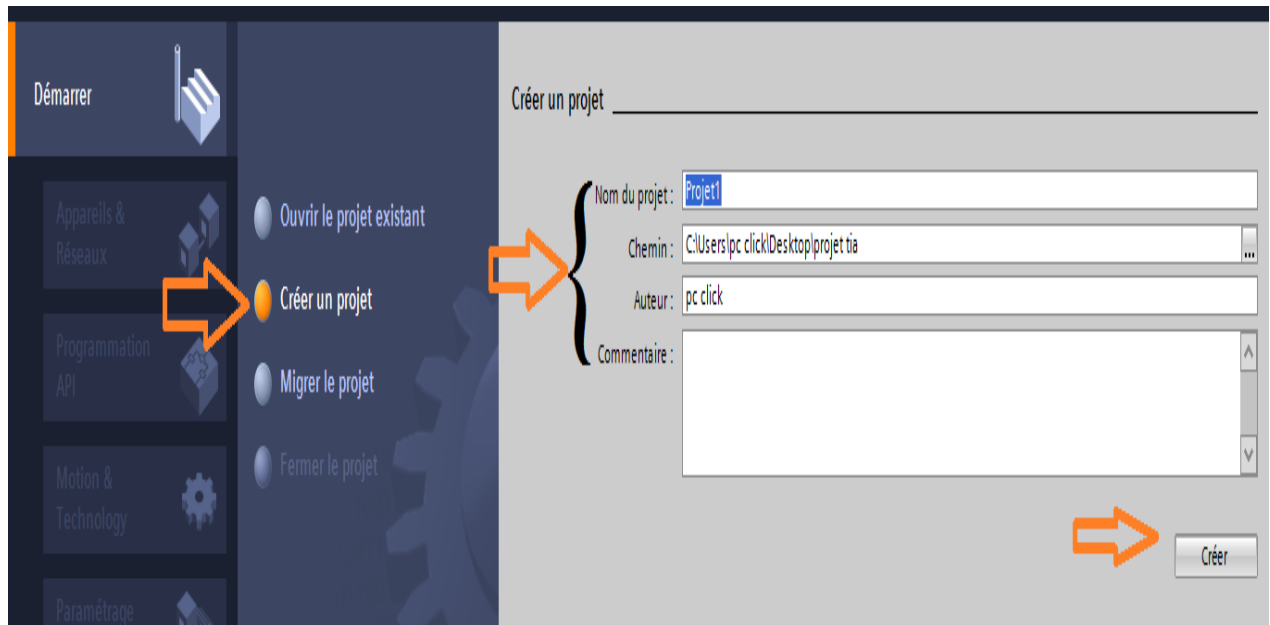


Figure 2 : Vue TIA Portal (Création d'un projet)

I.3. Configurer et ajouter un appareil

Après la création de notre projet on doit configurer l'appareil qu'on doit programmer.

- Cliquez sur « **Configurer un appareil** » [9].

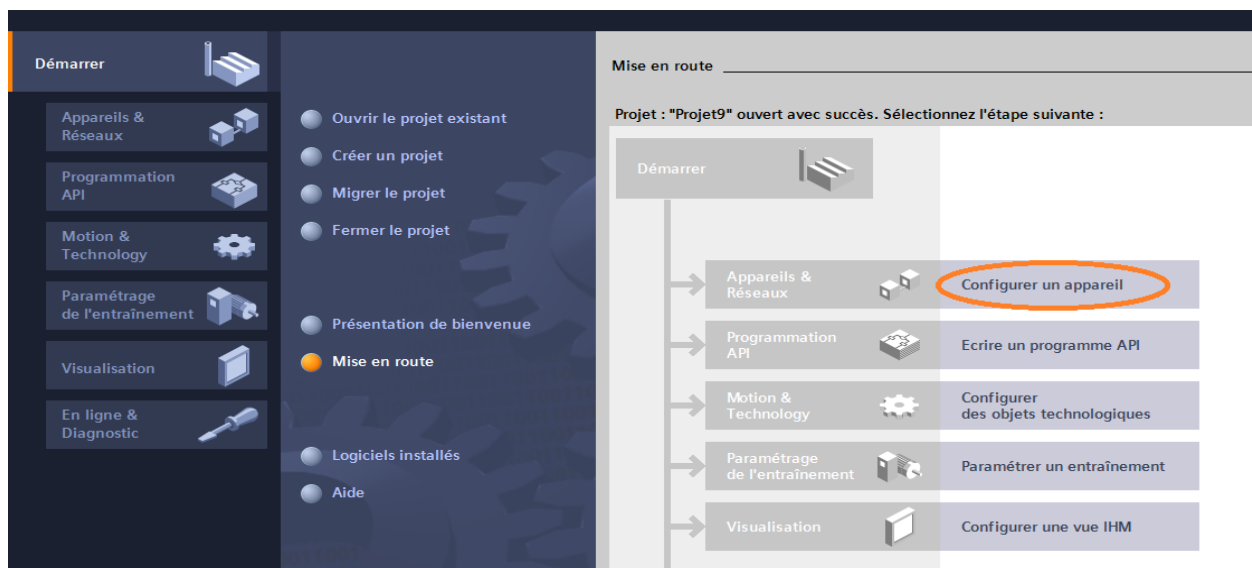


Figure 3 : Configurer un appareil sur TIA Portal

- Cliquez sur « **Ajouter un appareil** ».
- Cliquez sur « **SIMATIC S7-1200** ».
- Cliquez sur « **CPU** ».
- Sélectionner le CPU utilisé (dans notre projet on a utilisé le CPU 1214C DC/DC/RLY).
- Sélectionner le numéro de série de CPU concerné.
- Cliquez sur « **Ajouter** ».

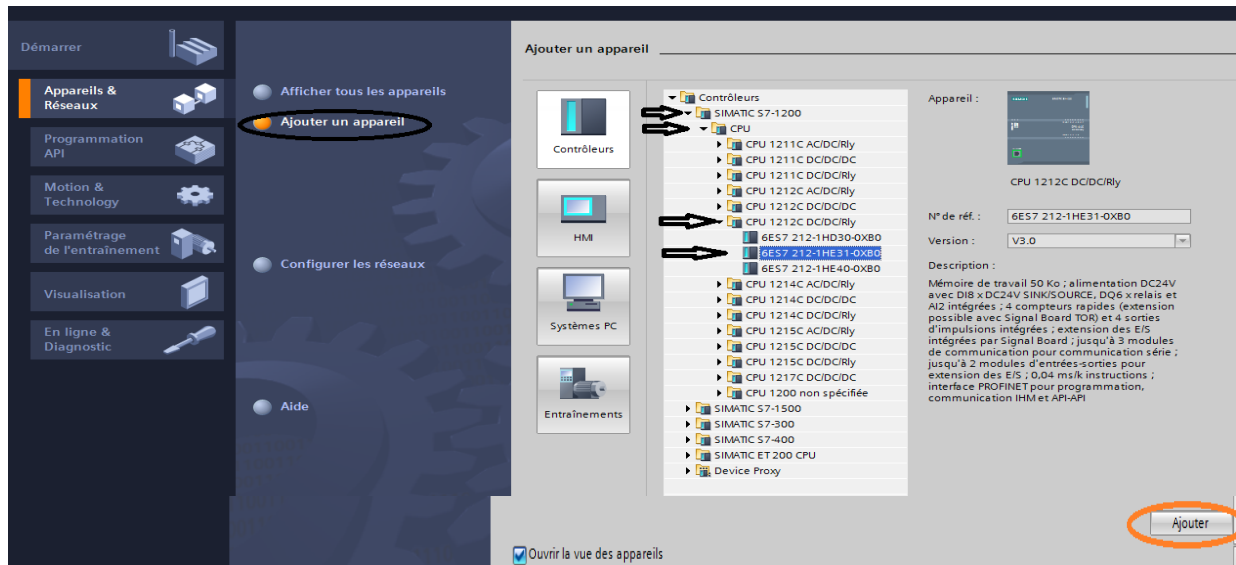


Figure 4 : Ajouter un appareil sur TIA Portal

I.4. L'interface et l'adresse Ethernet de CPU

Après la configuration et l'ajoute d'appareil on a l'interface suivant :

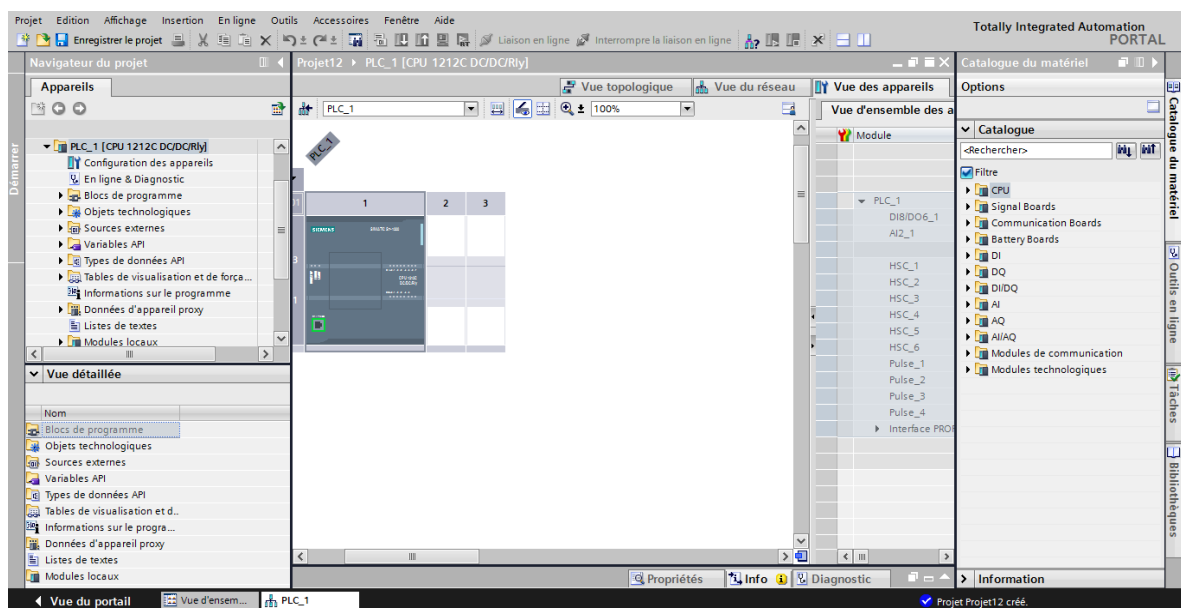


Figure 5 : Vue de l'interface de TIA Portal

➤ l'adresse Ethernet de CPU

Toujours dans les propriétés de la CPU, il est possible de définir son adresse Ethernet. Un double clic sur le connecteur Ethernet de la station fait apparaître la fenêtre d'inspection permettant de définir ses propriétés. Pour établir une liaison entre la CPU et la console de programmation, il faut affecter aux deux appareils des adresses appartenant au même réseau. On utilisera comme adresse pour l'automate « **192.168.0.1** » de l'automate.

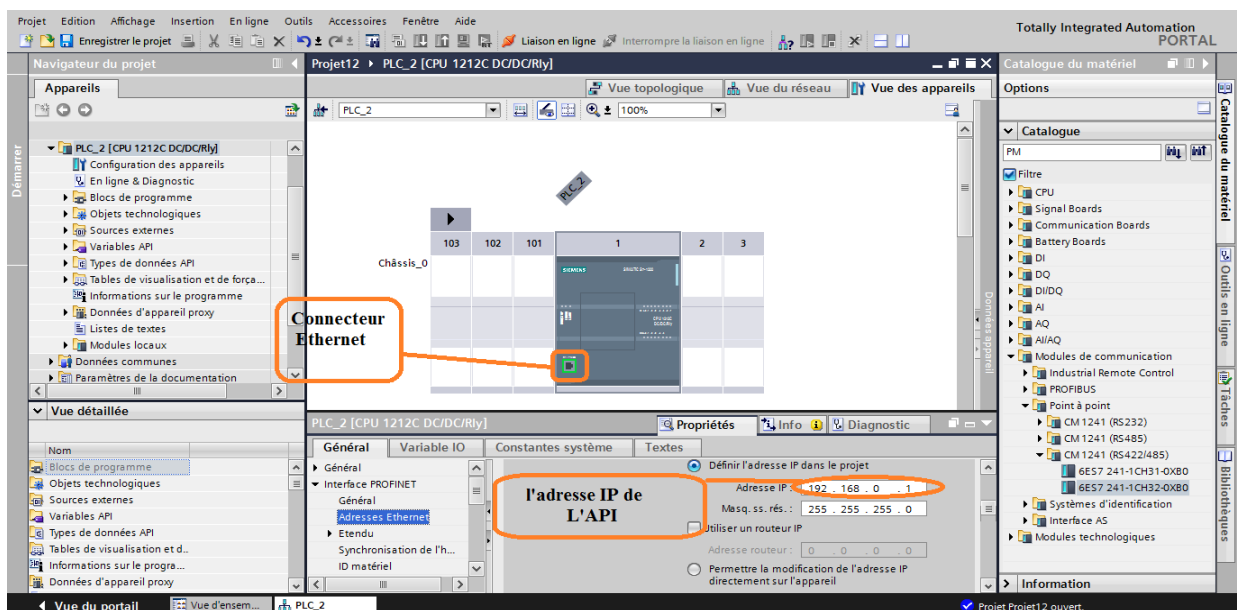
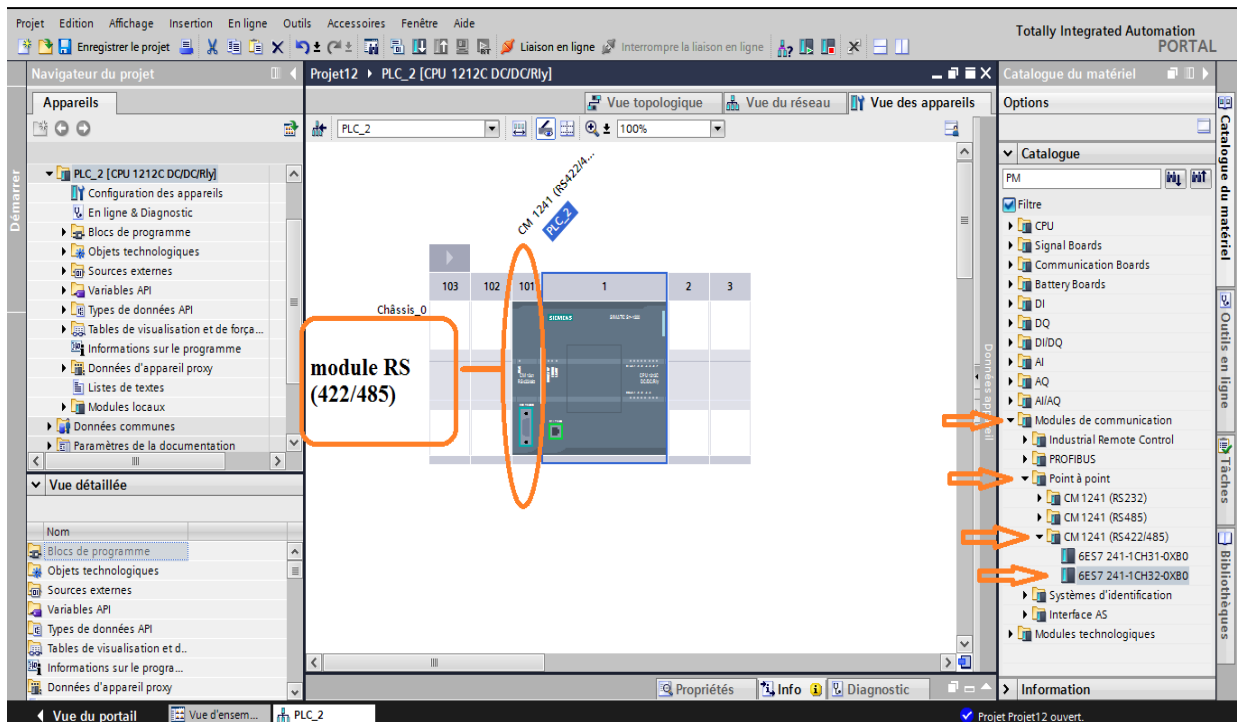


Figure 6 : Adresse Ethernet de la CPU

I.5. Langages de programmation

Le TIA PORTAL met à disposition de puissants éditeurs pour la programmation des automates SIMATIC S7 [12],[13].

- ✓ Texte structuré (SCL).
- ✓ Schéma à contacts (CONT).
- ✓ Logigramme (LOG).

❖ CONT et LOG – Langages de programmation graphiques

➤ LOG

Le logigramme (LOG) est un langage de programmation graphique qui utilise les boîtes de l'algèbre de Boole pour représenter les opérations logiques. Les fonctions complexes, comme par exemple les fonctions mathématiques, peuvent être représentées directement combinées avec les boîtes logiques.

➤ CONT (LADDER)

Le langage LADDER ou schéma à contacts est un langage de programmation graphique très populaire auprès des automaticiens. Il ressemble un peu aux schémas électriques, et est facilement compréhensible. Il existe 3 types d'éléments de langage :

- ❖ Les entrées (ou contact), qui permettent de lire la valeur d'une variable booléenne.
- ❖ Les sorties (ou bobines) qui permettent d'écrire la valeur d'une variable booléenne.
- ❖ Les blocs fonctionnels qui permettent de réaliser des fonctions avancées.

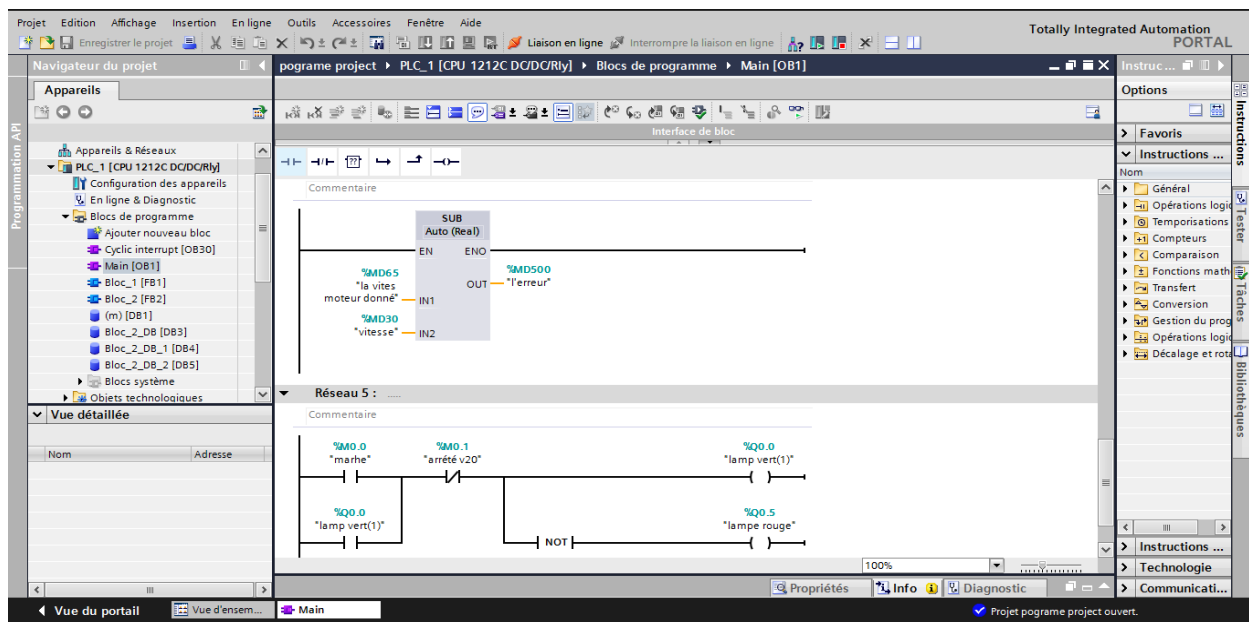


Figure 7 : Présentation d'un schéma à contacte (CONT)

❖ SCL (Structured Control Language)

Le SCL (Structured Control Language) correspond au langage textuel de haut niveau ST (Structured Text) défini dans la norme CEI 61131-3. SCL convient notamment à la programmation rapide d'algorithmes complexes et de fonctions mathématiques ou à des missions relevant du domaine du traitement des données.

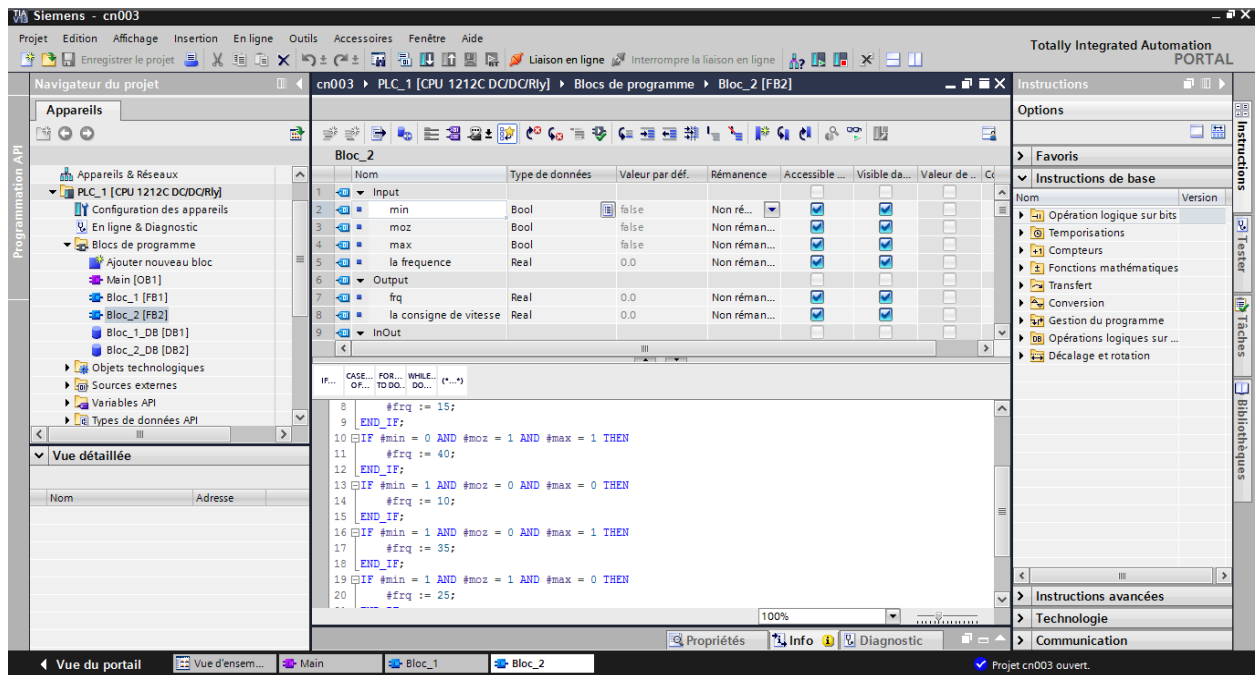


Figure 8 : Présentation d'un langage structuré (SCL).

a. Blocs de programme

L'automate met à disposition différents types de blocs qui contiennent le programme et les données correspondantes. Selon les exigences et la complexité du processus, il est possible de structurer le programme en différents blocs : OB, FB et FC.

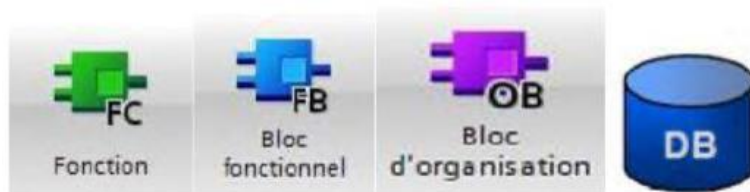


Figure 9 : Les différents blocs de programmation

❖ **Les blocs d'organisation OB** : sont appelés par le système d'exploitation en liaison avec les événements suivants :

- Comportement au démarrage.
- Exécution cyclique du programme.

- Exécution du programme déclenchée par des alarmes (cyclique, processus, diagnostic,...).
- Traitement des erreurs.

Pour que le traitement du programme démarre, le projet doit posséder au moins un OB cyclique (par exemple l'OB 1).

❖ **Les fonctions FC** : sont des blocs de code sans mémoire.

Les données des variables temporaires sont perdues après l'exécution de la fonction. Si on veut mémoriser ces données, il faut utiliser des opérands globaux. Elles sont utilisées pour la programmation de fonctions utilisées plusieurs fois. On simplifie de ce fait la programmation.

❖ **Les blocs fonctionnels FB** : sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'accéder même après le traitement de blocs.

❖ **Les blocs de donnée DB** : sont des zones données du programme utilisateur qui contiennent des données utilisateur.

b. Compilation et chargement de programme et la configuration matérielle

Une fois la configuration matérielle et la programmation réalisée, il faut la compiler et la charger dans l'automate. La compilation se fait à l'aide de l'icône « **Compiler** » de la barre de tâche. On sélectionne l'API dans le projet puis on clique sur l'icône « **Compiler** ». En utilisant cette manière, on effectue une compilation matérielle et logicielle.

Pour charger la configuration et le programme dans l'automate, on effectue un clic sur l'icône « **Charger dans l'appareil** » (figure 1.21).

La fenêtre de la figure 1.22 s'ouvre et on doit faire le choix du mode de connexion (PN/IE, Profibus, MPI). Si on choisit le mode PN/IE, l'API doit posséder une adresse IP. Pour faire la liaison entre l'automate et le PC et on charge le programme contenu dans le PC à l'automate.

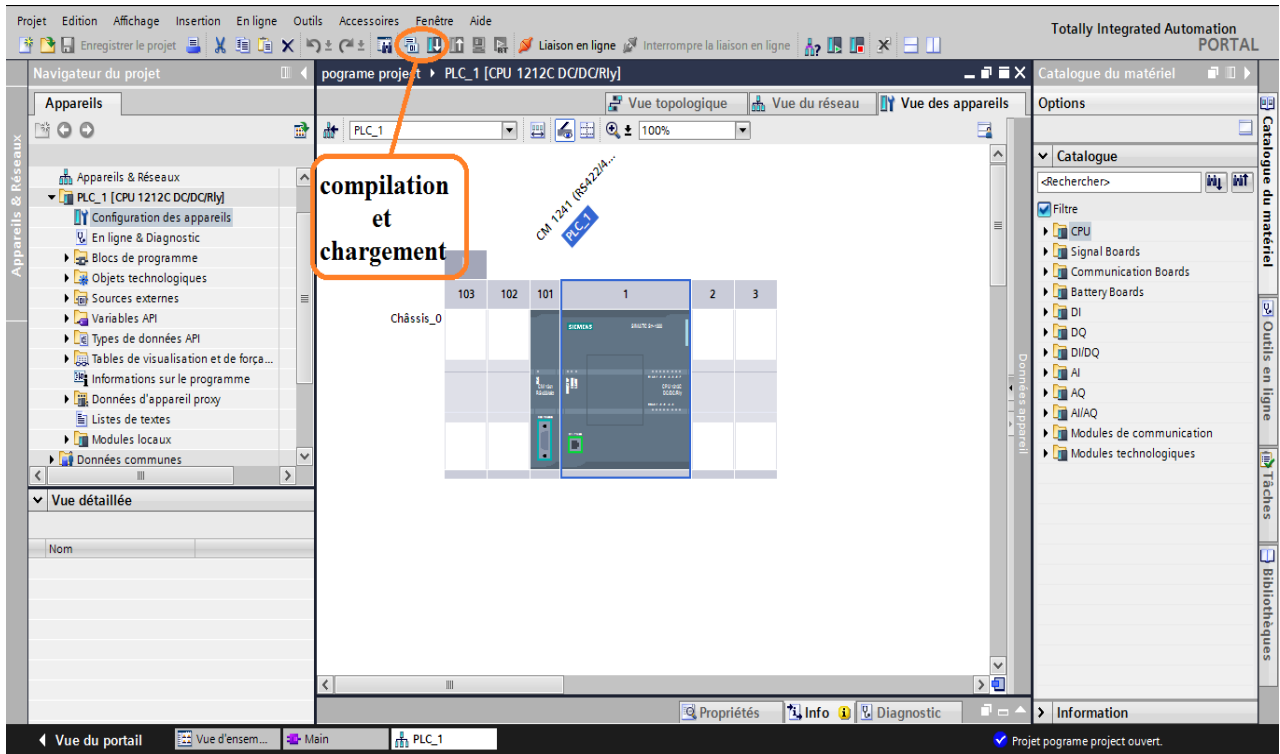


Figure 10 : Compilation et chargement dans l’API

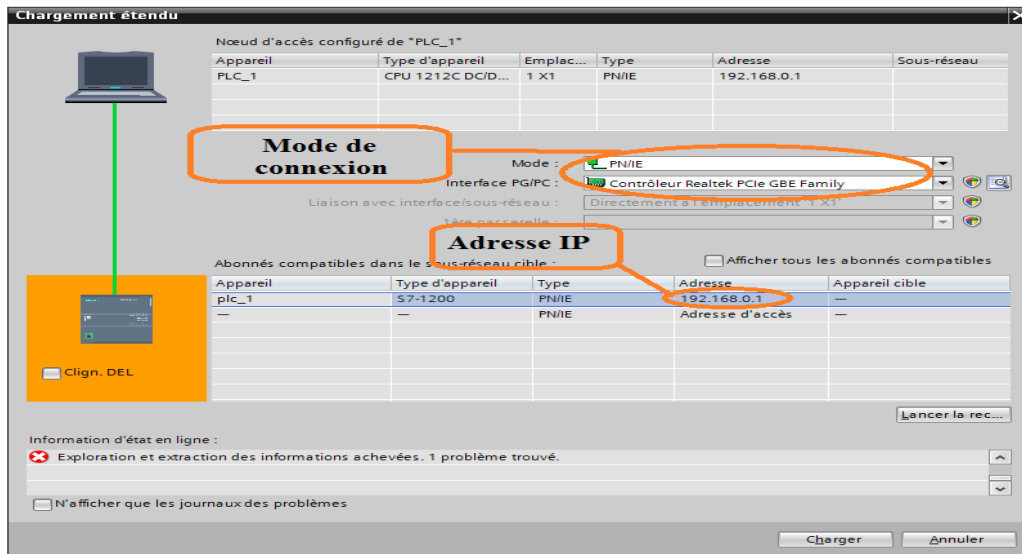


Figure 11 : Mode de connexion

c. Les variables API

c.1. Adresses symbolique et absolue

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,...) possède une adresse symbolique et une adresse absolue[13].

- L’adresse absolue représente l’identificateur d’opérande (I, Q, M,...) et son adresse et numéro de bit.

- **L'adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex : Bouton_Marche).

Le lien entre les adresses symbolique et absolue se fait dans la table des variables API.

Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

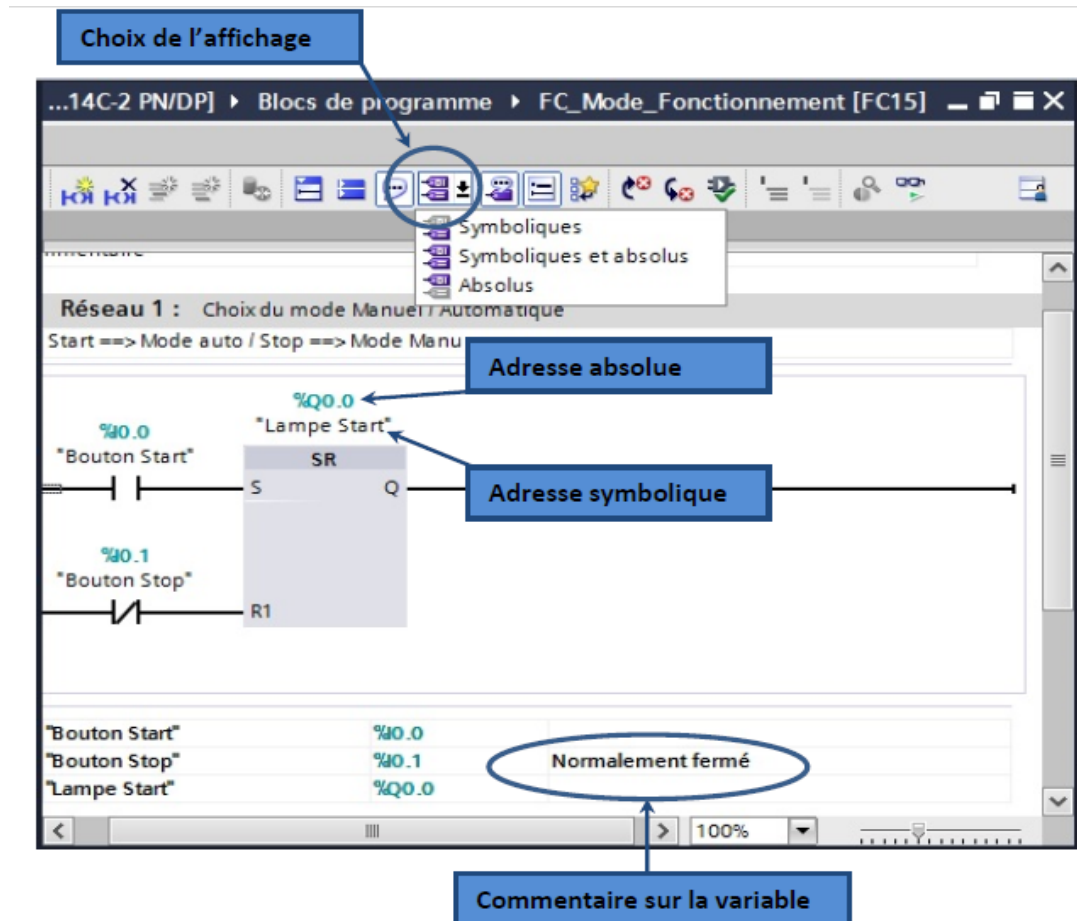


Figure 12 : Adresses symbolique et absolue

c.2. Table des variables API

C'est dans la table des variables API que l'on va pouvoir déclarer toutes les variables et les constantes utilisées le programme [13].

Lorsque l'on définit une variable API, il faut définir :

- ✓ **Un nom** : c'est l'adressage symbolique de la variable.
- ✓ **Le type de donnée** : BOOL, INT,...
- ✓ **L'adresse absolue** : par exemple Q1.5

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

The screenshot displays the 'Table de variables standard' (Standard Variables Table) within a software interface. The table lists 13 variables with their names, data types, addresses, and visibility/accessibility settings. The 'Lampe Q2' variable is highlighted, and its properties are shown in the 'Propriétés' window below.

	Nom	Type de données	Adresse	Réma...	Visible..	Acces...	Commentaire
1	Lampe Start	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2	Lampe Reset	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3	Lampe Q1	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4	Lampe Q2	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sur face avant de la station
5	Conveyeur vers la droite	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
6	Conveyeur vers la gauche	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7	Déplacement pince vers le haut	Bool	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8	Déplacement pince vers le bas	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9	Ouverture pince	Bool	%Q1.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10	Fermeture pince	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11	Rotation pince position 0°	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12	Rotation pince position 180°	Bool	%Q1.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13	Bouton Start	Bool	%I0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

The 'Propriétés' window for 'Lampe Q2' shows the following details:

- Nom : Lampe Q2
- Type de données : Bool
- Adresse : %Q0.3
- Commentaire : Sur face avant de la station

Figure 13 : Table des variables API

Remarque:

- La commande copier / coller permet de copier une table des variable sur Excel facilement.
- En sélectionnant le coin inférieur droit d'une cellule d'adresse ou d'un nom et en le faisant glisser vers le bas, on peut créer des variables automatiquement (comme dans Excel).
- Visible dans IHM : seules les variables cochées peuvent être visibles lors de la configuration HMI.
- Accessible dans HMI (uniquement S7 – 1200) : permet au pupitre HMI l'accès en ligne aux variables API sélectionnées.

D. Signalisation des erreurs dans la table des variables

- Lorsqu'il y a une erreur de syntaxe dans la table des variables API, celle-ci est signalée en rouge ou en orange. Lorsque l'on sélectionne la case colorée, un message signalant le type d'erreur apparaît.
- Une table des variables contenant des erreurs peut être enregistrée mais ne pourra pas être compilée et chargée dans l'automate [13].

Nom	Type de données	Adresse	Réma...	Visible dans IHM	Accessible depuis IHM	Commentaire
1 Lampe Start	Bool	%Q0.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
2 Lampe Reset	Bool	%Q0.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
3 Lampe Q1	Bool	%Q0.2		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
4 Lampe Q2	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
5 Convoyeur vers la droite	Bool	%Q0.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Sur face avant de la station
6 Convoyeur vers la gauche	Bool	%Q0.5		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
7 Déplacement pince vers le haut	Int	%Q1.0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
8 Déplacement pince vers le bas	Bool	%Q1.1		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
9 Ouverture pince	Bool	%Q0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
10 Fermeture pince	Bool	%Q1.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
11 Rotation pince position 0°	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
12 Rotation pince position 180°	Bool	%Q1.4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
13 Lampe Q2(1)	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
14 Bouton Start	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
15 Bouton Stop	Bool			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
16 Interrup...ur Auto / Menu	Bool	%I0.2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	Normalement fermé
17 Bouton Reset	Bool	%I0.3		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Figure 14 : Signalisation des erreurs dans la table des variables

E. Renommer / réassigner des variables

Il est possible de renommer et réassigner les variables API dans la fenêtre de programmation. Pour cela, il faut faire un clic droit sur la variable en question et choisir l'option renommer ou réassigner.

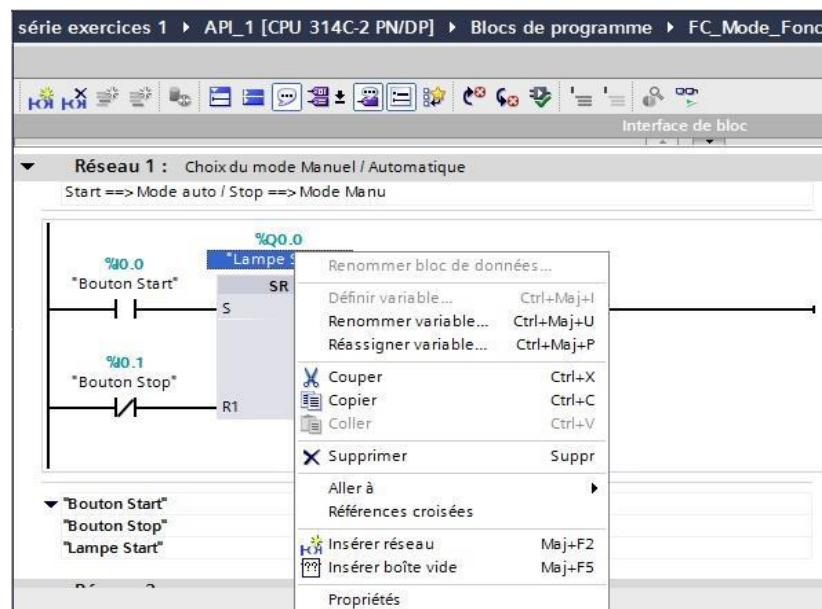
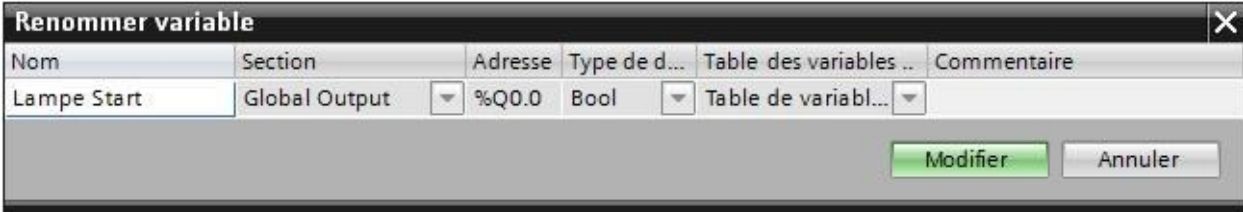


Figure 15 : Renommer / réassigner des variables (a)

La fenêtre si dessous s'ouvre.



Nom	Section	Adresse	Type de d...	Table des variables ..	Commentaire
Lampe Start	Global Output	%Q0.0	Bool	Table de variabl...	

Modifier Annuler

Figure 16 : Renommer / réassigner des variables (b)

- **Renommer la variable** donne l'accès aux paramètres Nom et Commentaire.
- **Réassigner la variable** donne l'accès aux paramètres Section, Adresse, Type de données, Table des variables API et Commentaire.

Annexe C

I.1. propriété du port de communication

Pour que l'automate (S7-1200) soit correctement lié à IHM, les propriétés doivent être configurées en TIA PORTAL Par des étapes comme indiqué ci-dessous :

1

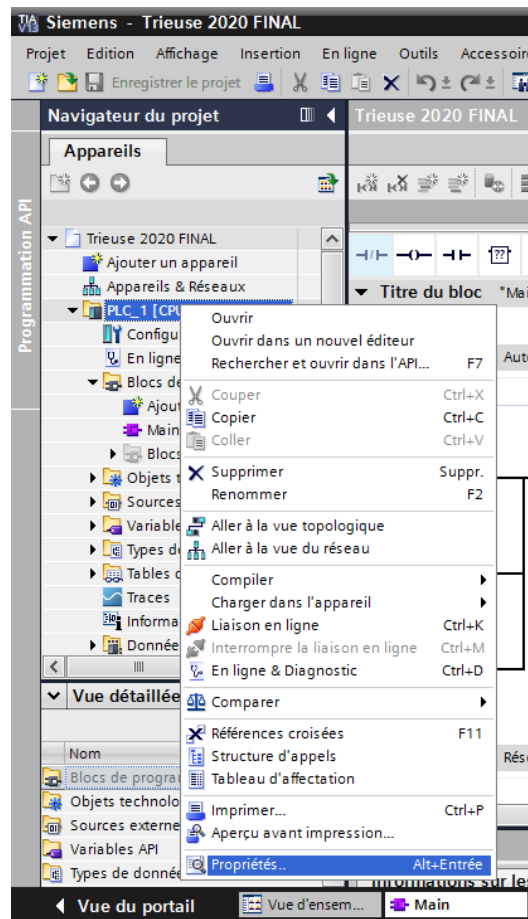


Figure 1 : Affichage des paramètres d'automate Siemens S7-1200

2

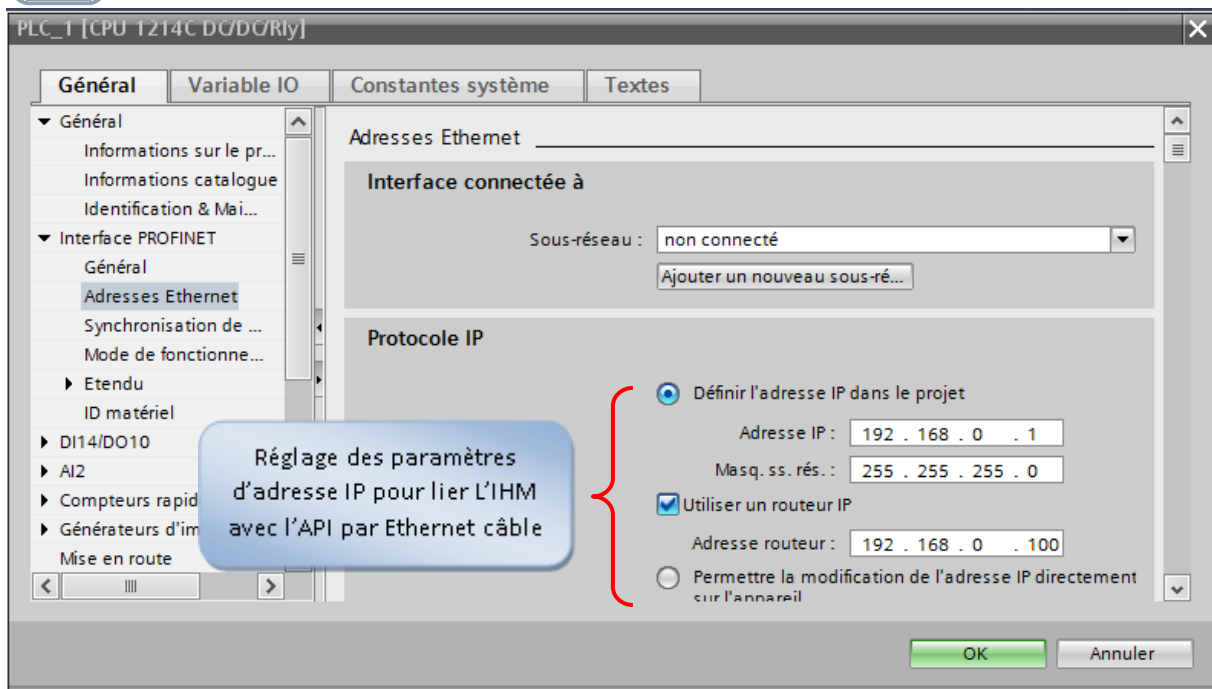


Figure 2 : fenêtre des Paramètres de communication

3

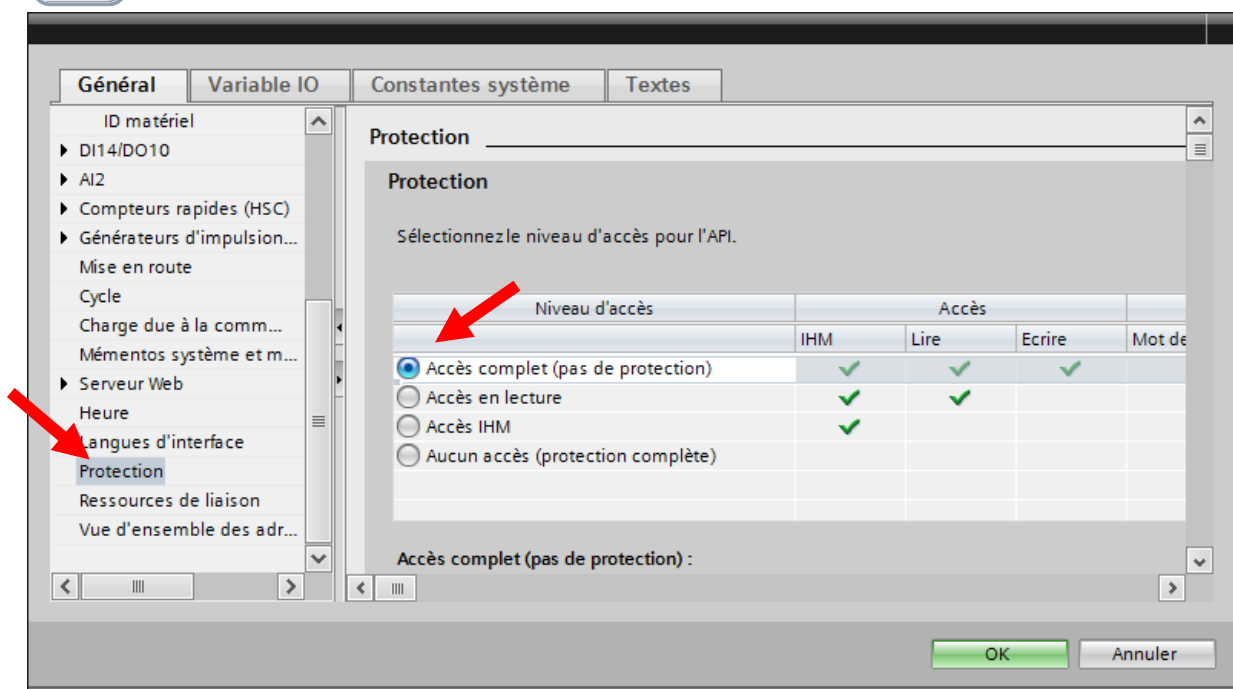


Figure 3 : fenêtre des Paramètres de protection

4

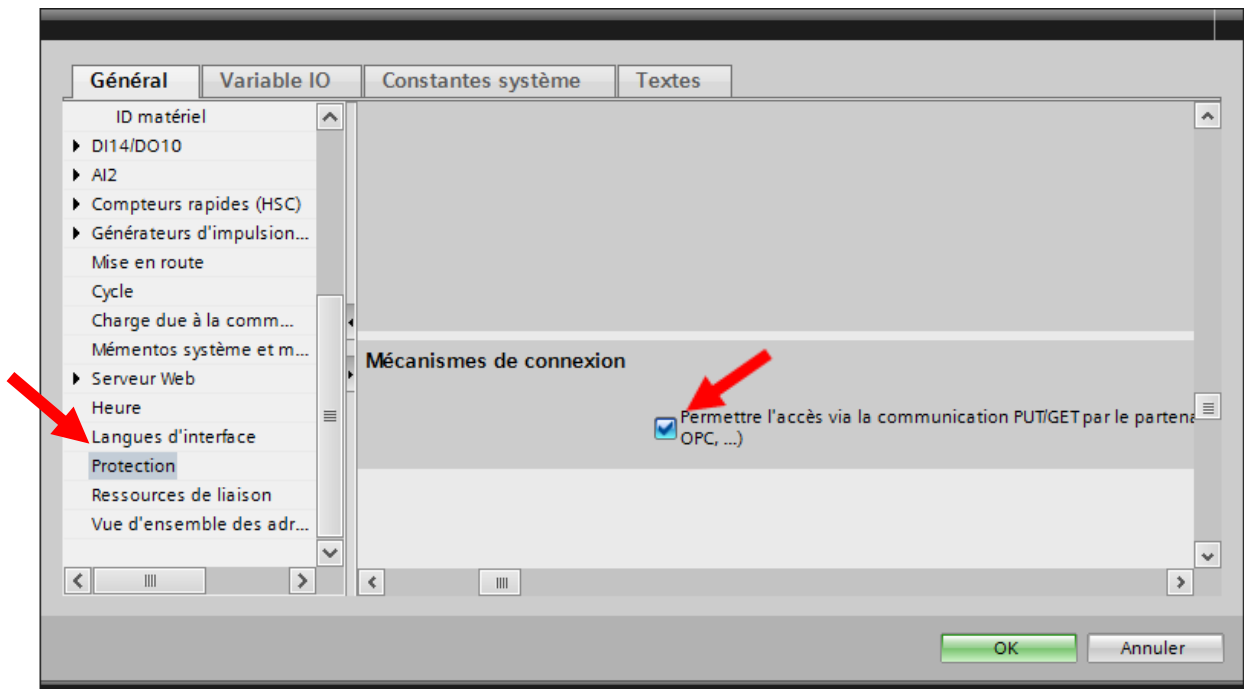


Figure 3 : fenêtre des Paramètres de protection

II. Le logiciel SKTOOL

Créé pour la série IHM de Samkoon SK à partir de Shenzhen Samkoon Technologie, SKTOOL est un logiciel d'édition de configuration de Samkoon IHM , SKTOOL prend en charge les pilotes de communication PLC des principaux fabricants mondiaux, tels que Samkoon, Siemens, Mitsubishi, Omron, Fujitsu, Panasonic, Schneider, Emerso , Le logiciel de configuration SKTOOL est un système de développement avec un environnement de développement intégré et de nombreuses fonctions avancées, il est facile à apprendre et à appliquer, ainsi les ingénieurs peuvent facilement développer leurs propres projets pour la configuration utilisée dans SKTOOL.

II.1. Création d'un projet et configuration

Cette opération est nécessaire créer un nouveau projet, Sélectionnez Nouveau projet dans le menu Fichier ou sur le bouton Nouveau projet, La figure ci-dessus représente la boîte de dialogue Nouveau projet dans laquelle l'utilisateur décide du nom, du chemin d'accès, du modèle et du mode d'affichage

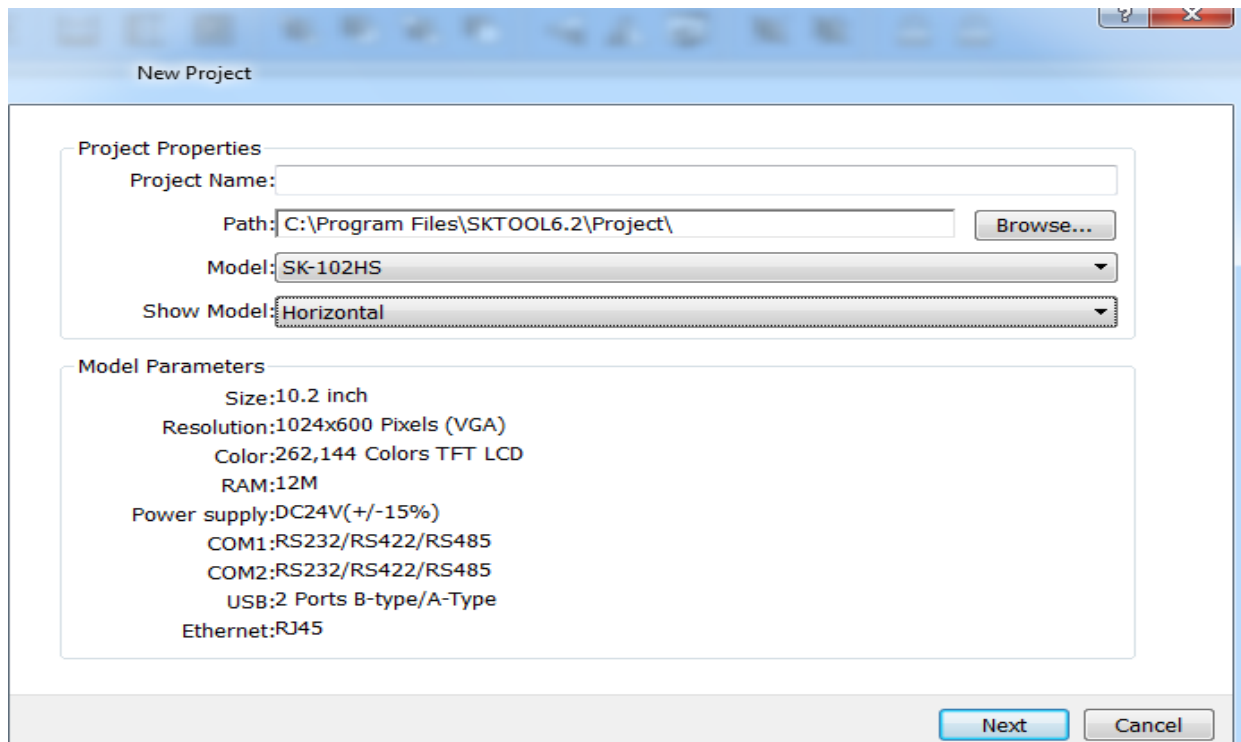


Figure 4 : Création d'un projet

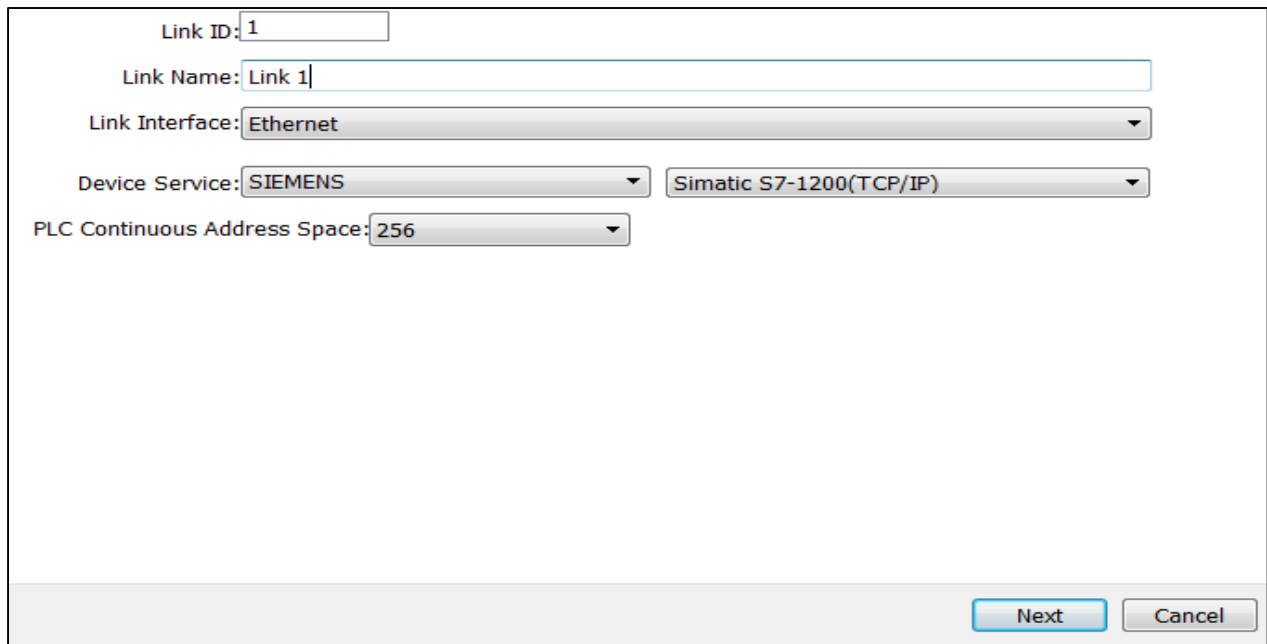
Le modèle spécifique est déterminé par le modèle IHM que utilisé, dans notre banc d'essai en utilise SK-102HS, puis cliquez sur Suivant pour définir le port de communication.

II.2. Boîte de dialogue Paramètres de communication

La boîte de dialogue Paramètres de communication sert à définir les paramètres de communication du port de communication de l'IHM. Seuls des paramètres de communication corrects peuvent assurer une communication normale entre l'automate et les écrans tactiles, Les paramètres de communication spécifiques dépendent du modèle d'automate à connecter.

Les paramètres le plus importants dans La boîte de dialogue Paramètres de communication :

1. Nom de la connexion: nom du port de communication.
2. Service de périphérique: sélection de la marque de l'automate et du type de CPU.
3. Connexion Interface: Select COM port or Ethernet port
4. Intervalle d'adresse continue de l'automate: définit l'espace adresse consécutive de l'automate; la valeur par défaut est recommandée.



The image shows a software dialog box titled 'Communication Parameter'. It contains several input fields and dropdown menus. The 'Link ID' field contains the number '1'. The 'Link Name' field contains 'Link 1'. The 'Link Interface' dropdown menu is set to 'Ethernet'. The 'Device Service' dropdown menu is set to 'SIEMENS', and the adjacent dropdown menu is set to 'Simatic S7-1200(TCP/IP)'. The 'PLC Continuous Address Space' dropdown menu is set to '256'. At the bottom right of the dialog box, there are two buttons: 'Next' and 'Cancel'.

Figure 5 : Boîte de dialogue Paramètre de communication

La figure ci-dessus représente la boîte de dialogue des paramètres du port de communication dans laquelle les utilisateurs choisissent les services de périphérique correspondants en fonction du type de périphérique de connexion. Pour utiliser une connexion Ethernet, choisissez l'option «Connexion directe (Ethernet)» dans le «Type de connexion». Cliquez sur «Suivant» dans la boîte de dialogue Paramètres de communication pour afficher une boîte de dialogue Nouvel écran.

II.3. Paramètres de l'écran

Boîte de dialogue Nouvel écran dans laquelle vous pouvez définir le nom et la couleur d'arrière-plan des écrans. Le nom par défaut est un numéro d'écran. la couleur d'arrière-plan par défaut est le bleu. Lorsque nous créons chaque fois une nouvelle image, le logiciel de configuration SKTOOL attribue un numéro d'écran au nouvel écran pour indiquer l'ordre d'affichage du projet en cours.

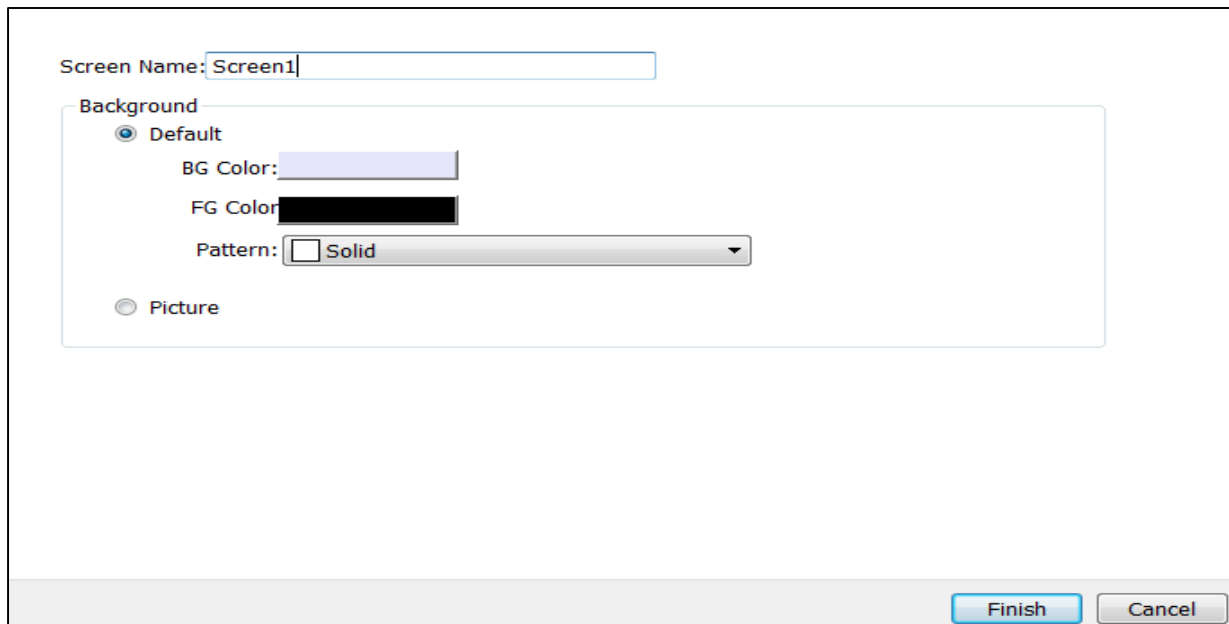


Figure 6 : Nouvelle boîte de dialogue

Cliquez sur Terminer pour terminer les étapes ci-dessus, le système ouvrira immédiatement l'écran que vous venez de créer.

II.4. La propriété du port de communication

La propriété du port de communication est appliquée pour définir les paramètres de communication entre l'IHM et les périphériques connectés. Double-cliquez sur "Link" et "Link 1" qui se trouvent dans Coordinateur de projet, comme le montre dans la figure III.10

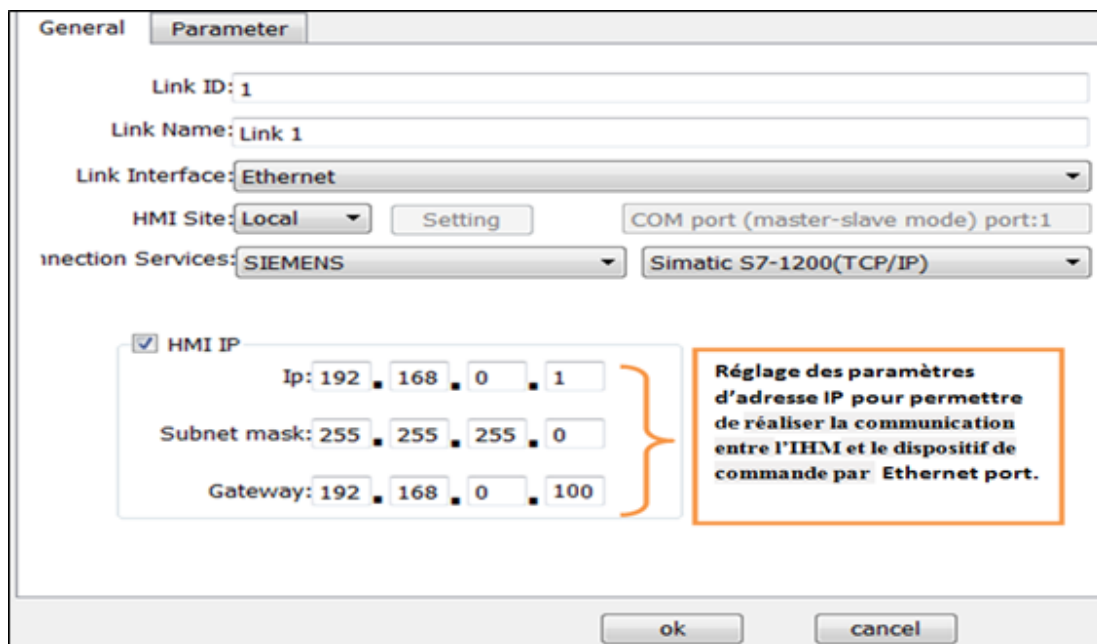





Figure 7 : Boîte de dialogue propriété du port

Pour régler les paramètres d'adresse IP il nous faut cocher la case IHM IP :

- IP : c'est l'adresse IP spécialisé pour IHM lui-même.
- Gateway : Adresse IP de l'automate programmable communiqué avec L'IHM par la porte Ethernet.

II.5. Compilation et chargement de programme

Une fois l'édition de la configuration terminée, vous devez cliquer sur le bouton  puis cliquez sur  Pendant la compilation, la fenêtre de sortie d'informations affichera la situation en temps réel, Il ne peut pas terminer la compilation si une erreur existe. Il est nécessaire de compiler avant de télécharger, alors que l'étape de conservation est nécessaire avant de compiler. Après pour charger le programme vous devez cliquer sur le bouton .

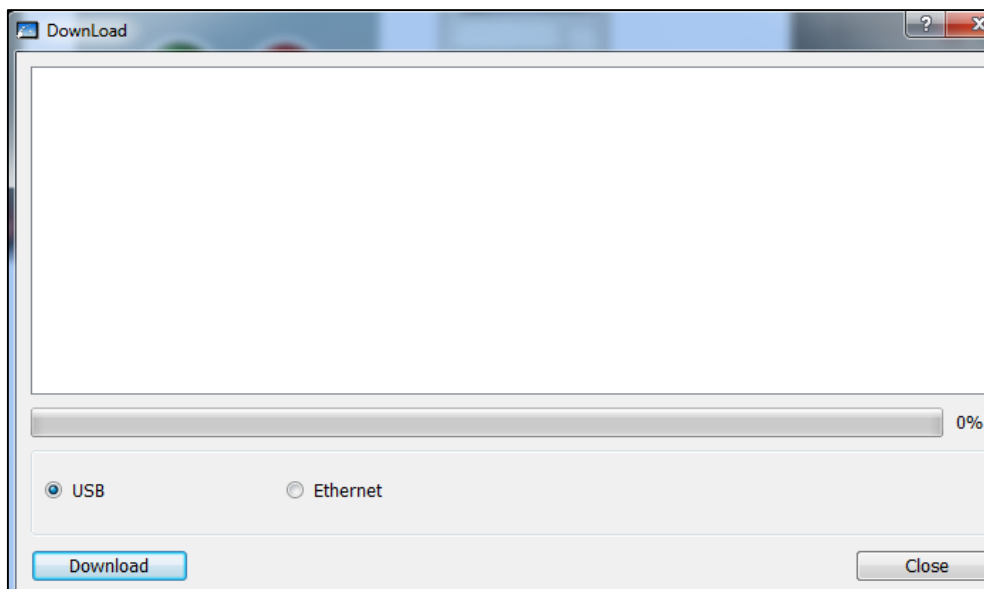


Figure 8 : Boîte de dialogue de téléchargement

Enfin, exécutez la commande USB dans le menu Télécharger pour télécharger via un câble USB. Cliquez sur télécharger; la barre de progression du milieu montre la progression du téléchargement. Après le téléchargement, une fenêtre contextuelle montre l'achèvement du téléchargement sur USB. Cliquez sur OK pour terminer la tâche de téléchargement. Comme le montre dans la figure 6.

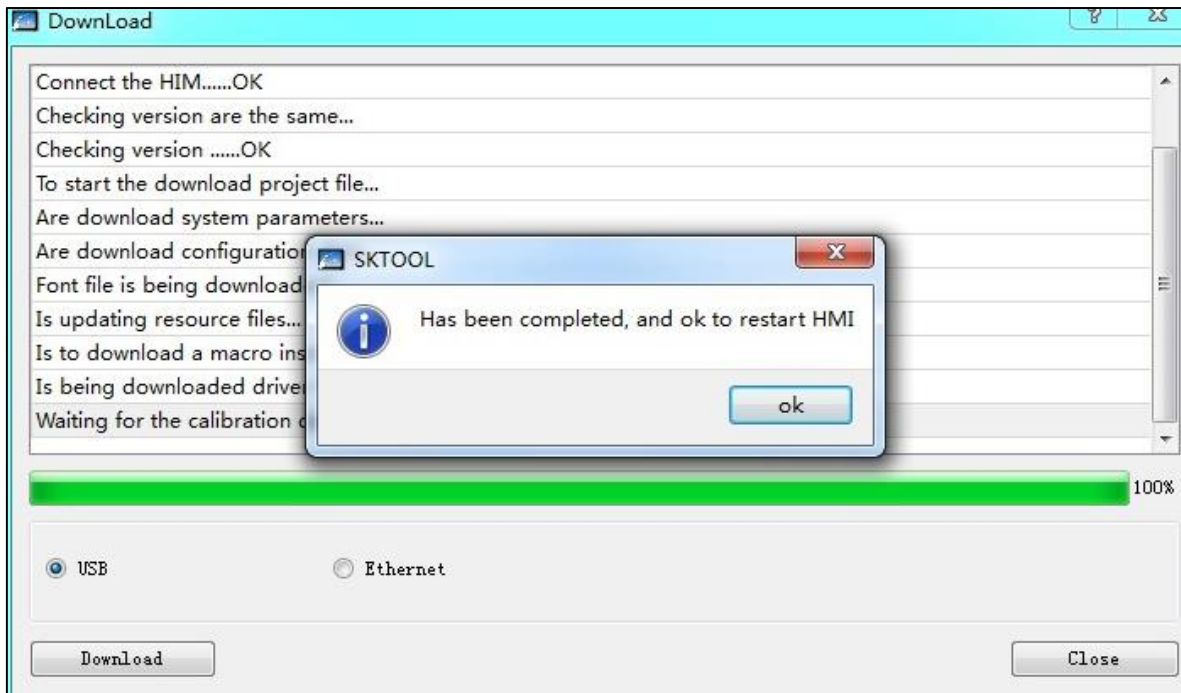


Figure 9 : Téléchargement terminé

L'étape suivante est la connexion à l'automate pour voir si le projet de configuration s'exécute correctement dans l'IHM.

III.Programme Python :

Les algorithmes du traitement d'image sont réalisés par le script Python indiqué ci-dessous :

```

*Programme_Final trieuse 2020 .py - G:\Université\M2\PFE\Rédaction\Format Word 2020\Mémoire final\Programme_Final trieuse ...
File Edit Format Run Options Window Help

## Appele des bibliothèques nécessaires

import cv2 # Appel à bibliothèque Open CV
import numpy as np # Appel à bibliothèque numpy qui fait les calculs matriciel
import RPi.GPIO as GPIO # Appel à bibliothèque GPIO qui gère les portes de Raspberry
from picamera import PiCamera # Appel à bibliothèque concernant le module de camera
from picamera.array import PiRGBArray # Le module picamera.array fournit un ensemble de classes
# qui aident à construire des tableaux numpy
# n dimensions à partir de la sortie de la caméra.

GPIO.setwarnings(False)

def nothing(x):
    pass

font = cv2.FONT_HERSHEY_COMPLEX

## configuration des paramètre de caméra

camera = PiCamera()
camera.resolution = (320, 240)
camera.framerate = 60 # Pour prendre 60 image par seconde
rawCapture = PiRGBArray(camera, size=(320, 240))

## Boucle pour prendre des photos en continu

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    belt = image[10:195,82:213] # Cibler la zone d'intérêt
    hsv = cv2.cvtColor(belt, cv2.COLOR_BGR2HSV) # convertir l'image de RGB --> HSV

    lower_yellow = np.array([0, 62, 27]) #définir l'intervalle de Jaune
    upper_yellow = np.array([42, 193, 255]) #entre les Valeurs min et max

    lower_red = np.array([140, 78, 22]) #définir l'intervalle de Jaune
    upper_red = np.array([180 ,255, 166]) #entre les Valeurs min et max

    mask_yellow = cv2.inRange(hsv, lower_yellow, upper_yellow)#faire le masque pour afficher
    mask_red = cv2.inRange(hsv, lower_red, upper_red) #la couleur cibler

    kernel = np.ones((5, 5), np.uint8) #technique utilisé pour éliminer
    mask_yellow = cv2.erode(mask_yellow, kernel) #le Bruit en maximum
    mask_red = cv2.erode(mask_red, kernel)

    result_yellow = cv2.bitwise_and(belt, belt, mask=mask_yellow)# Extraction seulement
    result_red = cv2.bitwise_and(belt, belt, mask=mask_red) # l'image d'objet

    mean_val_yellow = cv2.mean(result_yellow, mask=mask_yellow)#les valeurs moyenne d'objet
    mean_val_red = cv2.mean(result_red, mask=mask_red) #les valeurs moyenne d'objet

    H_y = mean_val_yellow[0]
    S_y = mean_val_yellow[1]
    V_y = mean_val_yellow[2]

    H_r = mean_val_red[0]
    S_r = mean_val_red[1]
    V_r = mean_val_red[2]

    ## Test pour Sélectionnee la couleur jaune uniquement

    if H_y > 50 and H_y < 78 and S_y > 84 and S_y < 96 and V_y > 103 and V_y < 122 :
        print("yellow")
        k=0
        while k<100000 :
            GPIO.setmode(GPIO.BCM)
            GPIO.setup(17, GPIO.OUT)
            k=k+1
        GPIO.cleanup(17)

```

```
    ## Test pour Sélectionner la couleur rouge uniquement
if H_r > 60 and H_r < 75 and S_r > 12 and S_r < 59 and V_r > 88 and V_r < 112 :
    print("red")
    k=1
    while k<100000 :
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(27, GPIO.OUT)
        k=k+1
    GPIO.cleanup(27)

cv2.imshow('Frame',image)

cv2.imshow('Mask_yellow',mask_yellow)      #
cv2.imshow('Mask_red',mask_red)           #
#
cv2.imshow("result_yellow", result_yellow) # Pour afficher toutes les images
cv2.imshow("result_red", result_red)       # qui ont été effectuées sur
#les différentes opérations
cv2.imshow("RGB-image", belt)              #
cv2.imshow("HSV-image", hsv)               #

key = cv2.waitKey(1) #
rawCapture.truncate(0) # Lorsque en appuyant sur le bouton ESC,
if key == 27:      # toutes les fenêtres de travail sont fermées
    break

camera.close()      # Éteignez la caméra
cv2.destroyAllWindows() # Fermez toutes les fenêtres de travail
```

Figure 10 : Script Python