

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU



FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'AUTOMATIQUE

Mémoire de Fin d'Etudes de MASTER PROFESSIONNEL

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Automatique et informatique
industrielles**

Présenté par
Ghiles OUKALI
Nabil MOKHTARI

Thème

Conception et réalisation d'une carte électronique pour le contrôle du convoyeur CU-4001

Mémoire soutenu publiquement le 13/07/ 2017 devant le jury composé de :

CHILALI Ouardia	MAA
YOUSFI Safia	MAA
TOUAT Mohand Achour	MCB
MOULOUDJ Kamal	Co-encadreur

Remerciements

Avant tout, nous remercions le bon **Dieu** le tout puissant de nous avoir donné le courage, la volonté et la patience durant toutes les années d'études et que grâce à lui ce travail a pu être réalisé.

Nos vifs remerciements vont en premier lieu, à nos chers parents et familles de nous avoir aidé, encouragés et soutenus tout au long de ces années et qui continuent de croire en nous en dépit de tout.

Je souhaite exprimer mes vifs remerciements à :

Notre promoteur Mr.M.A.TOUAT qui a cru en nous, qui nous a soutenus et orienter tout au long notre travail et pour tous les moyens mis à notre disposition pour l'élaboration de ce mémoire. Qu'il trouve ici l'expression de notre reconnaissance.

Nous souhaitons également exprimer notre profonde gratitude et nos vifs remerciements à Mr.K.MOULOUDJ formateur et enseignant à l'école technique de Tizi-Ouzou, qui n'a jamais cessé de nous suivre et orienter tout au long de notre travail, ainsi que pour ses conseils judicieux. On a pu bénéficier à la fois de ses compétences scientifiques, et de sa grande disponibilité, tant pour résoudre les difficultés rencontrées lors de notre réalisation, de répondre à nos questions.

Nous remercions d'avance, les membres de jury qui nous font l'honneur de juger notre travail. Nous espérons être à la hauteur de leurs attentes.

Nous sommes aussi très reconnaissantes à tous les enseignants qui ont veillé au bon déroulement de notre formation tout au long de notre cursus. Qu'ils trouvent ici l'expression de notre respect et remerciements les plus sincère.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que ce modeste travail voit le jour.

Table des matières

Introduction générale.....	5
----------------------------	---

CHAPITRE 1 : Description du convoyeur CU-4001 et son fonctionnement général.

I.1) Unité du convoyeur CU-4001	6
I.1.1) Introduction.....	6
I.2) Spécifications.....	7
I.3) Structure du convoyeur simulateur.....	8
I.3.1) Description de la structure.....	8
I.3.2) Avant utilisation	10
I.4) Fonctionnement et cahier des charges.....	11
I.4.1) Fonctionnement en général.....	11
I.4.2) Avant programmation.....	12
I.4.3) modes d'utilisation de convoyeur CU-4001 :.....	13
I.4.4) Cahier des charges.....	15
I.5) CONCLUSION	16

CHAPITRE 2: présentation du système de contrôle et le choix des composants.

II.1) Introduction.....	17
II.2) présentation du système de contrôle.....	17
II.2.1) conception du circuit imprimé de la carte électronique.....	19
II.2.2) Principes de réalisation.....	21
II.3) choix et rôles des composants.....	22
II.3.1) Le microcontrôleur ATMEGA 328.....	22
II.3.2) L'afficheur LCD 4*20.....	23

II.3.2.1) Description des broches et connexion de l’afficheur LCD avec un microcontrôleur.....	23
II.3.2.2) Caractéristiques de l’afficheur et fonctionnement.....	23
II.3.2.3) Mode d’utilisation de l’afficheur	25
II.3.3) Le potentiomètre, Le buzzer piézo-électrique , Le Quartz.....	25
II.3.4) Les résistances, Les LEDs, Les condensateurs	27
II.3.5) La carte relais.....	29
II.3.6) Le connecteur USB	30
Conclusion.....	31

CHAPITRE 3 : présentation du logiciel de programmation, du langage de programmation et du programme contrôlant le convoyeur.

III.1) présentation du logiciel de programmation	32
III.1.1) introduction à l’Arduino	32
III.1.2) Le logiciel Arduino (IDE Arduino)	32
III.2) présentation du langage de programmation	32
III.2.1) C Arduino.....	33
III.2.1.1) Présentation de (EDI) Arduino.....	33
III.2.1.2) Différents compositions de l'espace de développement intégré (EDI).....	34
III.3) Organigramme montrant les étapes de programmation de convoyeur.....	42
III.4) présentation du programme contrôlant le convoyeur CU-4001	43
Conclusion générale	45

Tables des figures

Figure 1 : convoyeur CU-4001.

Figure 2 : Image des différents composants du convoyeur simulateur CU-4001.

Figure 3 : schémas montrant le fonctionnement général du convoyeur.

Figure 4: fonctionnement du mode manuel.

Figure 5: fonctionnement du mode automatique.

Figure 6 : Image de la carte électronique.

Figure7 : Logiciel TCI tracé de circuit imprimé

Figure 8: circuit imprimé de notre carte électronique.

Figure9 : Le microcontrôleur de la carte électronique.

Figure 10 : Image de l'afficheur LCD 4*20.

Figure 11 : image du potentiomètre.

Figure12 : Buzzer piézo-électrique.

Figure13: Le quartz.

Figure14 : Code des couleurs des résistances.

Figure 15: La LED.

Figure16 : Image de la carte relais 4 canaux avec shield.

Figure 17 : Aperçue des liaisons entre la carte et le convoyeur.

Figure 18 : Les différentes parties de la fenêtre principale du logiciel Arduino.

Figure 19 : Les différentes barres de la fenêtre principale du logiciel Arduino.

Figure 20 : Barre de boutons pour le programme C-Arduino.

Figure 21 : Les différentes composantes du programme Arduino.

Figure 22 :L'écriture du programme.

Figure 23 : Compilation du programme.

Figure 24 : Console d'affichage.

Figure 25 : Sélection de la cible.

Figure 26 : Sélection du port.

Figure 27 : Bouton de transfert du programme.

Introduction générale:

L'unité de convoyage de type ceinture CU-4001 est un simulateur de tri qui fonctionne avec un entraîneur d'automate. Certaines des caractéristiques de CU-4001 sont la capacité d'évaluer la taille des pièces en déplacement sur la ceinture pour les trier, et aussi compter le nombre de pièces passant par un point de contrôle.

En outre, le système peut être préréglé pour générer un signal lorsqu'un certain nombre de pièces est atteint. Un contrôleur logique programmable est nécessaire pour améliorer l'efficacité des expériences.

Dans ce mémoire nous avons réalisé une carte électronique avec accessoires nécessaires, et conçu un programme avec comme langage le C-Arduino qui a pour fonction le contrôle des différentes tâches de l'unité de convoyage CU-4001.

Ce mémoire est organisé en trois chapitres à travers lesquels, nous décrivons le travail effectué pour la conception et la réalisation de notre système.

Dans le premier chapitre nous présentons la structure de l'unité de convoyage CU-4001, le fonctionnement général et les spécificités du cahier de charges sur lequel nous nous sommes basés pour la réalisation de notre projet.

Le second chapitre décrit la phase de conception de la carte de commande. On y décrit les principaux composants constituant notre carte électronique en justifiant leurs choix.

Le troisième et dernier chapitre présente le programme contrôlant le système. On y découvre les différentes tâches de notre carte, ses modes de fonctionnement ainsi que la mise en place de la partie logicielle et les résultats obtenus.

Nous concluons notre travail par une conclusion générale et les perspectives.

CHAPITRE 1 : Description du convoyeur CU-4001 et son fonctionnement général

1) Unité du convoyeur CU-4001 :

1.1) Introduction :



Figure 1 : convoyeur CU-4001

L'unité de convoyeur (transport) CU-4001 est un simulateur de convoyeur à courroie qui génère un environnement d'usine réaliste pour l'utilisateur. Il est désigné pour être utilisé avec le contrôleur programmable ED-4200

La fonction principale de cette unité est la capacité de compter les objets entre deux convoyeurs par la sortie du capteur. Détection de la déviation par la taille des objets en mouvement sur le convoyeur. Déclenchement des objets et modification de la direction des objets en mouvement.

La vitesse, le démarrage / arrêt et la modification de la direction des objets sont directement contrôlés par un programme extérieur. En outre, il est possible de contrôler le convoyeur en mode manuel.

2) Spécifications:

Taille du convoyeur60mm (longueur) x570mm (largeur).

vitesse mobile.....Niveau 1=12mm/s (approx.).

Niveau 2=18mm/s (approx.).

Détecteur porteur d'objets.....Capteur photo réfléchissant 2 voies.

moteur.....10 RPM (avec accélération).

Condition rejector.....2 positions.

Mode opératoiremanuel et automatique avec PLC.

Comparateur.....compteur à 2 chiffres (0~99).

Puissance d'entrée.....110/220V, 50~60Hz.

Poids.....11Kg.

3) Structure du convoyeur simulateur :

3.1) Description de la structure :

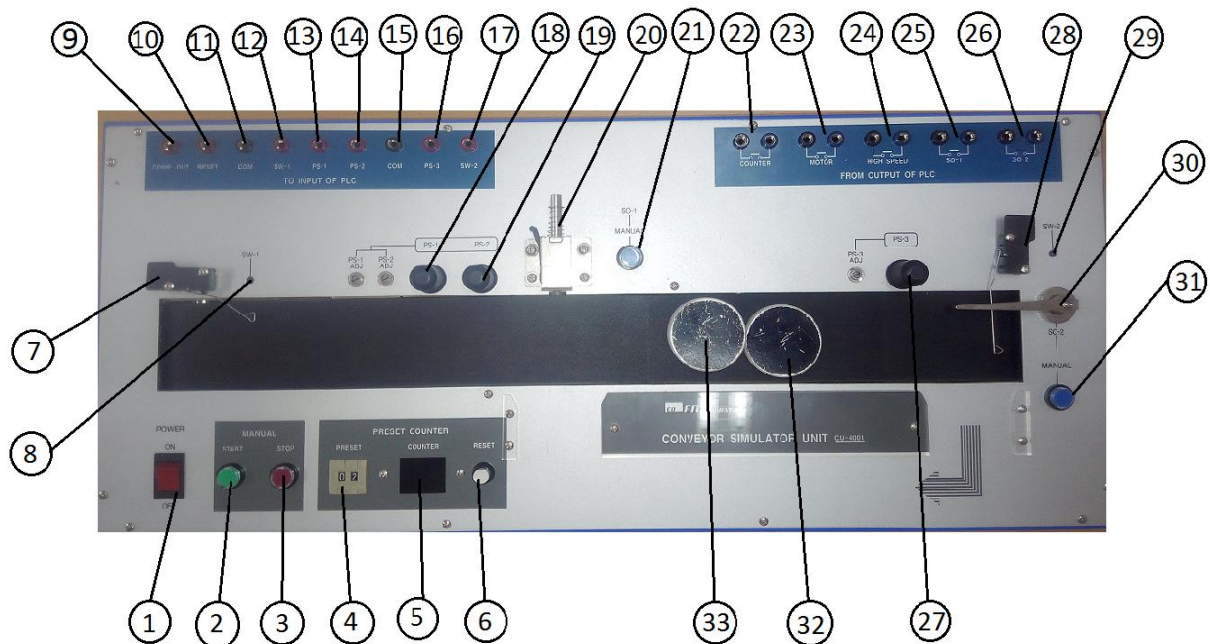


Figure 2 : Image des différents composants du convoyeur simulateur CU-4001.

- ① Power switch.....Allumage du CU-4001.
- ② START (manuel).....Bouton poussoir pour activer le convoyeur en mode manuel.
- ③ STOP (manuel).....Bouton poussoir pour désactiver le convoyeur en mode manuel.
- ④ PRESETCeci est pour le contrôle prééglé, et l'unité de transport peut générer un signal lorsque le nombre a atteint le nombre Prédéterminé de quantité.
- ⑤ COUNTER Display.....Il a un affichage à 2 chiffres.
- ⑥ RESET button.....Remise à zéro à l'indication du compteur.
- ⑦ SW-1 micro-switch.....Peut générer un signal de début pour le convoyeur.
- ⑧ Indicator.....LED indiquant si le SW-1 est ON.

- ⑨ Comparator signal output terminal.....Générer la sortie dans le cas où la
quantité comptée de l'affichage du compteur
est supérieure à la PRESET.
- ⑪ COM Terminal.....Borne de sortie des signaux 9 ~17.
- ⑫ Output terminal of SW-1.....Borne de sortie du commutateur 7.
- ⑬ Output terminal of PS-1.....Borne de sortie du capteur 18.
- ⑭ Output terminal of PS-2.....Borne de sortie du capteur 19.
- ⑮ COM terminalBorne de sortie identique à 11.
- ⑯ Output terminal of PS-3.....Borne de sortie du capteur 18.
- ⑰ Output terminal of SW-2.....Borne de sortie de la limite d'extrémité du micro-
interrupteur 28.
- ⑱ PS-1 sensorIl s'agit d'un photocapteur pour détecter les objets dont le
Niveau de hauteur est faible.
- ⑲ PS-2 sensor.....Il s'agit d'un photocapteur pour détecter les objets dont
le Niveau de hauteur est fort.
- ⑳ S0-1 solenoid.....Déplacer les objets vers un autre endroit en fonction
du signal de sélection de l'objet.
- ㉑ S0-1 manuel switch.....Il fait fonctionner le 20 manuellement.
- ㉒ Counter input terminal.....Borne d'entrée de l'affichage du compteur.
- ㉓ Motor ON/OFF terminal.....Borne d'entrée pour contrôler le moteur du convoyeur.
- ㉔ HIGH SPEED terminalBorne d'entrée pour accélérer la rotation du
moteur du convoyeur.
- ㉕ S0-1 input terminal.....Borne d'entrée pour contrôler le solénoïde 20.
- ㉖ S0-2 input terminal.....Borne d'entrée pour Contrôler le solénoïde 30.
- ㉗ PS-3 sensor.....Photocapteur pour détecter l'accumulation d'objet.
- ㉘ SW-2 micro-switch.....Le commutateur pour le capteur de limitation de fin.

- ②⑨ Indicator.....Il sera allumé lorsque le commutateur 31 est ON.
- ③① S0-2 Solenoid.....Il est possible de changer la direction de l'objet en mouvement par le capteur de fin de course.
- ③① S0-2 Manual switch.....Il fait fonctionner le 30 manuellement.
- ③② Petite pièces.
- ③③ Grande pièces.

3.2) Avant utilisation:

-Vérifier si la sélection du courant d'entrée est correcte ou non.

-La borne d'entrée du simulateur de transport CU-4001 n'est pas nécessaire pour appliquer le courant elle n'est nécessaire que pour établir un contact pour les contrôles d'entrée. Donc, si elle est contrôlée par un contrôleur logique programmable (PLC), la sortie de l'automate doit être effectuée en sortie de contact.

-De plus, la borne de sortie de CU-4001 doit être réalisée en sortie de contact sans appliquer de tension. Ainsi, dans le cas du contrôleur programmable ED-4200, il suffit de connecter la borne de sortie à l'entrée.

-Ne jamais appliquer de force sur la courroie de convoyage pendant le déplacement du convoyeur. Cela peut endommager le moteur de transport lorsque le convoyeur est en état de stationnement.

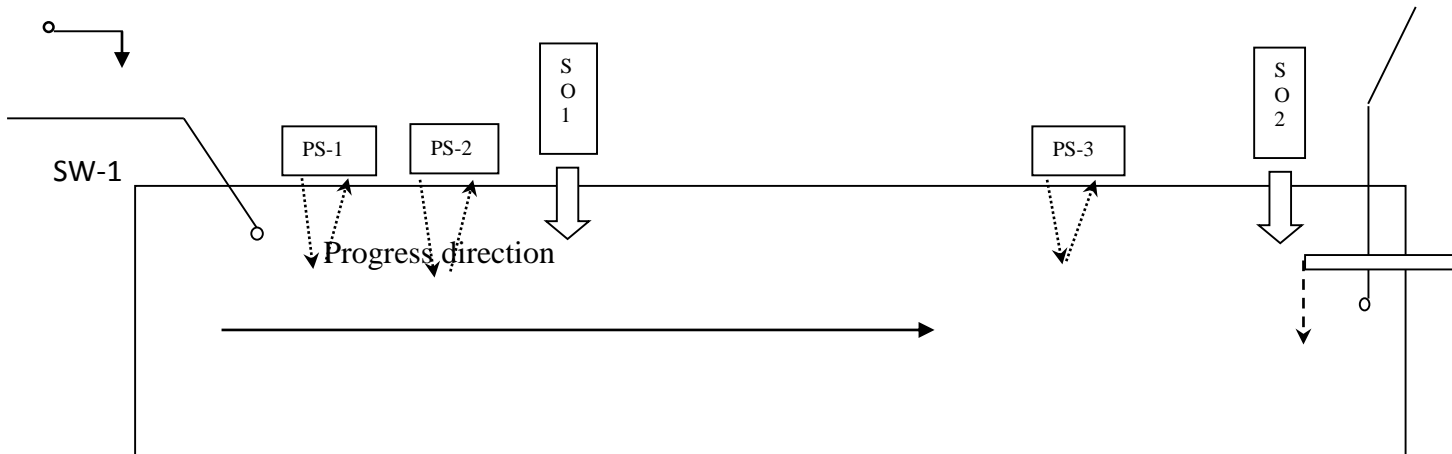
Les capteurs de détection d'objet de CU-4001 (PS-1 et PS-2) sont capables de détecter les pièces uniquement s'il est réglé pour répondre à la Hauteur de détection de l'objet. Il est impossible de détecter les pièces d'une taille différente.

-Le simulateur de convoyage CU-4001 est un équipement de précision, il peut facilement s'endommager pendant l'utilisation ou le transport.

4) Fonctionnement et cahier des charges:

4.1) Fonctionnement en général:

Dans les schémas ci-dessous on montre le fonctionnement du convoyeur :



*PS-1 PS-2 PS-3..... Photo capteur.

*SO-1 SO-2 Solénoïde.

Figure 3 : schémas montrant le fonctionnement général du convoyeur.

L'utilisation de l'unité de convoyage CU-4001 se fait comme suit :

-Le micro-interrupteur (SW-1) émet un signal lorsqu'on place un objet sur le convoyeur.

-Il est possible de tracer la hauteur de l'objet par photocapteur (PS-1 et PS-2) lorsque l'objet se déplace sur le convoyeur, la sortie du signal du capteur peut actionner le solénoïde (SO-1). Aussi le photo capteur PS-1 ou PS-2 est utilisé pour compter la quantité d'objets, et la sortie du capteur peut afficher la quantité de comptage avec la connexion de la sortie à l'entrée du compteur du CU-4001 ou par un autre afficheur.

-Et quand l'objet normalement passé est arrivé à la fin de la ligne de transport, il est détecté par un interrupteur de fin de course (SW-2). Cette sortie détectée est possible pour arrêter le convoyeur à travers un programme extérieur ou pour faire fonctionner le changeur de direction de mouvement des objets (solénoïde SO-2).

-Si nous voulions obtenir le signal de contrôle pour transporter la quantité sur le convoyeur (exemple 10 paquets) .en connectant la sortie de PS-2 ou PS-1 à l'entrée du compteur, il est possible d'obtenir la sortie de comparaison à chaque détection de 10 paquets avec réglage du comparateur du CU-4001 a 10.

-Il est également possible de changer la vitesse du moteur, et l'entrée-sortie de chaque fonction peut fonctionner avec la possibilité d'opérer selon la conception du programme.

4.2) Avant programmation:

-Avant la programmation, pour désigner le circuit de connexion entrée / sortie entre CU-4001 et Arduino tell configuré dans le programme.

-la connexion entre CU-4001 et Arduino se fait après programmation.

4.3) modes d'utilisation de convoyeur CU-4001 :

a) mode manuel :

Dans la figure ci-dessous nous expliquons comment fonctionne le mode manuel :

Bouton d'arrêt

Bouton de démarrage



Reset

éjecteurs SO-1 et SO-2

Figure 4: fonctionnement en mode manuel.

Le mode manuel fonctionne comme suit :

- Bouton de démarrage : pour actionner le moteur.
- Bouton d'arrêt : pour éteindre le moteur.
- Reset : pour la remise à zéro du comparateur à 2 bits en mode manuel.

-Ejecteurs SO-1 et SO-2 : pour éjecter les pièces manuellement.

b) mode automatique :

Dans l'image ci-dessous nous montrons comment contrôler le convoyeur automatiquement :

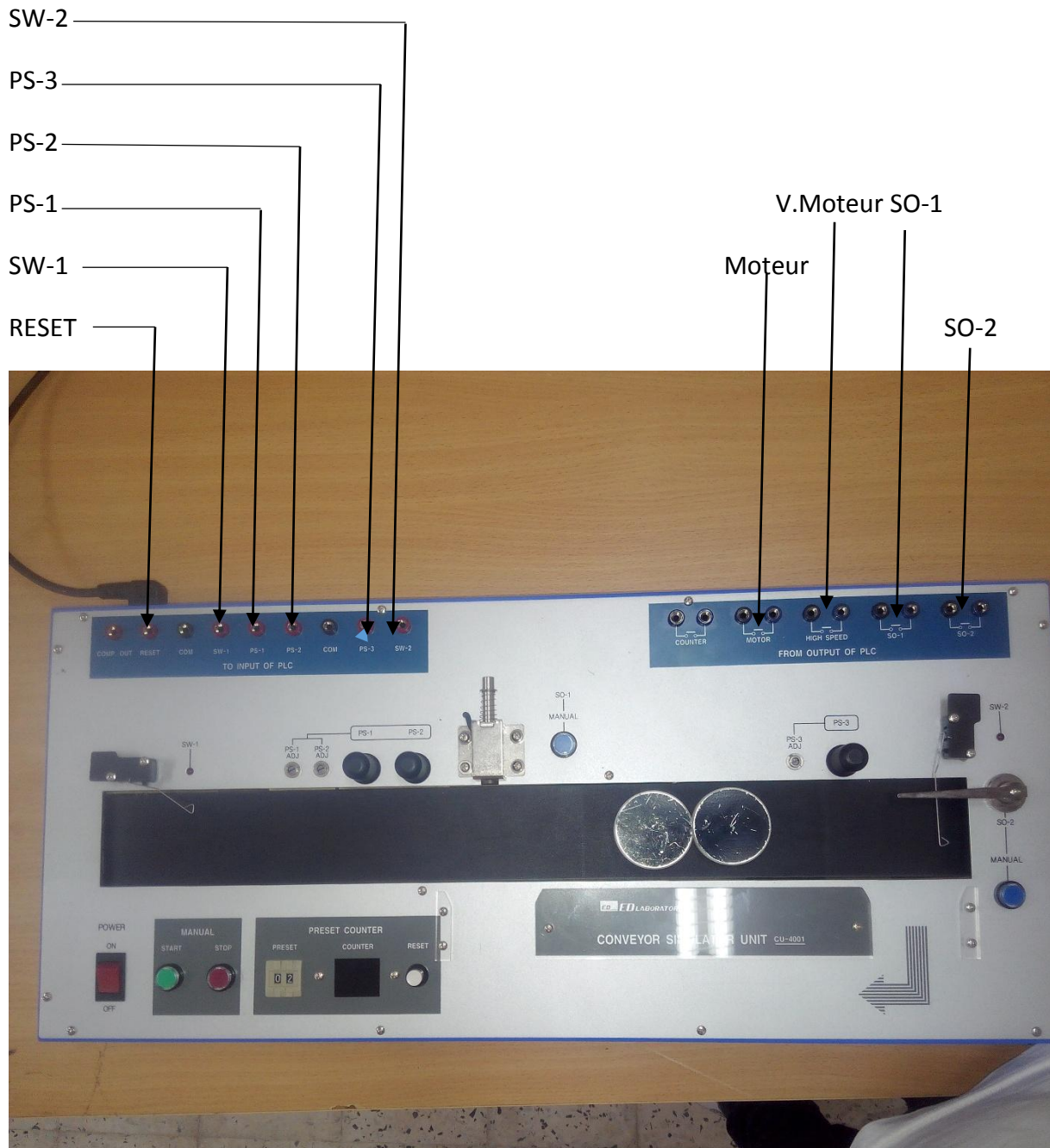


Figure 5: fonctionnement du mode automatique.

RESET: sortie Pour mettre le compteur à zéro en mode automatique.

SW-1 : sortie de Capteur début de cours

SW-2 : sortie de Capteur de fin de cours

PS-1 : sortie de Détecteur de petites et grandes pièces

PS-2 : sortie de Détecteur de grandes pièces

PS-3 : sortie de Détecteur de petites et grandes pièces

Moteur : entrée pour actionner le moteur en

V.Moteur : entrée pour passer la vitesse du moteur au niveau supérieur

SO-1 : entrée pour éjecter les grandes pièces en mode automatique.

SO-2 : entrée pour éjecter les petites pièces en mode automatique.

4.4) Cahier des charges:

-Le moteur démarrera avec le signal de SW-1 lorsqu'on met une pièce sur le convoyeur, et le moteur s'arrêtera lorsque la pièce est détectée par SW-2.

-Il est possible de démarrer le moteur par SW-1 et le moteur sera arrêté avec le signal de sortie du comparateur, qui fera la soustraction des pièces comptées par PS-1 et celles comptées par PS-2 et PS-3 (le nombre de toutes les pièces par le nombre de petites et grandes pièces), et si le nombre trouvé est nul le moteur s'arrêtera car il n'y a plus de pièces sur le convoyeur.

-Le compteur comptabilise la quantité de toutes les pièces avec le signal de PS-1 après que chacune d'elle le traverse.

-le nombre de grandes pièces sera comptabilisé par le photocapteur PS-2 après que chacune d'elle le traverse.

-le nombre de petites pièces sera comptabilisé par le photocapteur PS-1 ou en faisant la soustraction du nombre de pièces comptées par PS-1 et PS-2.

-si le photocapteur PS-1 détecte une pièce et que celle-ci n'est pas détecté par le photocapteur PS-3 ça veut dire que cette pièce est petite et ne sera éjecté par le solénoïde SO-1.

-par contre si la pièce est détectée par PS-1 et PS-2 en même temps ça veut dire que c'est une grande pièce elle sera directement éjecté par le solénoïde SO-1.

-après que la petite pièce arrive au SW-2 et que le moteur soit arrêté le solénoïde S0-2 l'éjectera après qu'une certaine période de temps soit écoulée.

-une fois les pièces arrivées au photocapteur PS-1 la vitesse du moteur du tapis augmentera vers la vitesse max, et celle-ci diminuera une fois que ces pièces arrivent à PS-3.

CONCLUSION :

Dans ce chapitre nous avons présenté le convoyeur simulateur CU-4001, son mode de fonctionnement, sa structure et ses différents composants et pour le système de contrôle nous allons utiliser un circuit électronique que nous allons présenter dans le deuxième chapitre.

CHAPITRE 2: présentation du système de contrôle et le choix des composants.

1) Introduction :

Après que nous avons donné la description de la machine et du système à contrôler et rédiger le cahier de charges, il y a lieu d'étudier la structure du système pour pouvoir traiter, visualiser et contrôler les différentes vitesses du moteurs, carte électronique et afficheurs. Dans ce chapitre nous allons présenter la carte électronique du système de contrôle et les critères de choix de chaque composant.

2) présentation du système de contrôle :

Notre système de contrôle est un circuit électronique qui est réalisé à base d'une carte électronique dont les composants sont les suivants :

- Un microcontrôleur ATMEGA 328.
- Un afficheur LCD 4*20.
- Un buzzer piézo-électrique
- Des capacités et des résistances.
- Des connecteurs d'entrée/sortie.
- Un port USB.
- Un bouton poussoir.
- Un quartz.
- Des LEDs.
- Des ports d'entrée\sortie.

-Et puisque le signal émis par la carte est de 5V et que le moteur a besoin de 220V nous avons utilisé une carte relais 4 canaux pour contrôler le moteur, le changement de vitesse et les deux éjecteurs SO-1 et SO-2.

- Pour la liaison entre la carte relais et le convoyeur nous avons utilisé des fiches bananes.

L'image ci-dessous montre notre carte électronique avec ses différents composants :

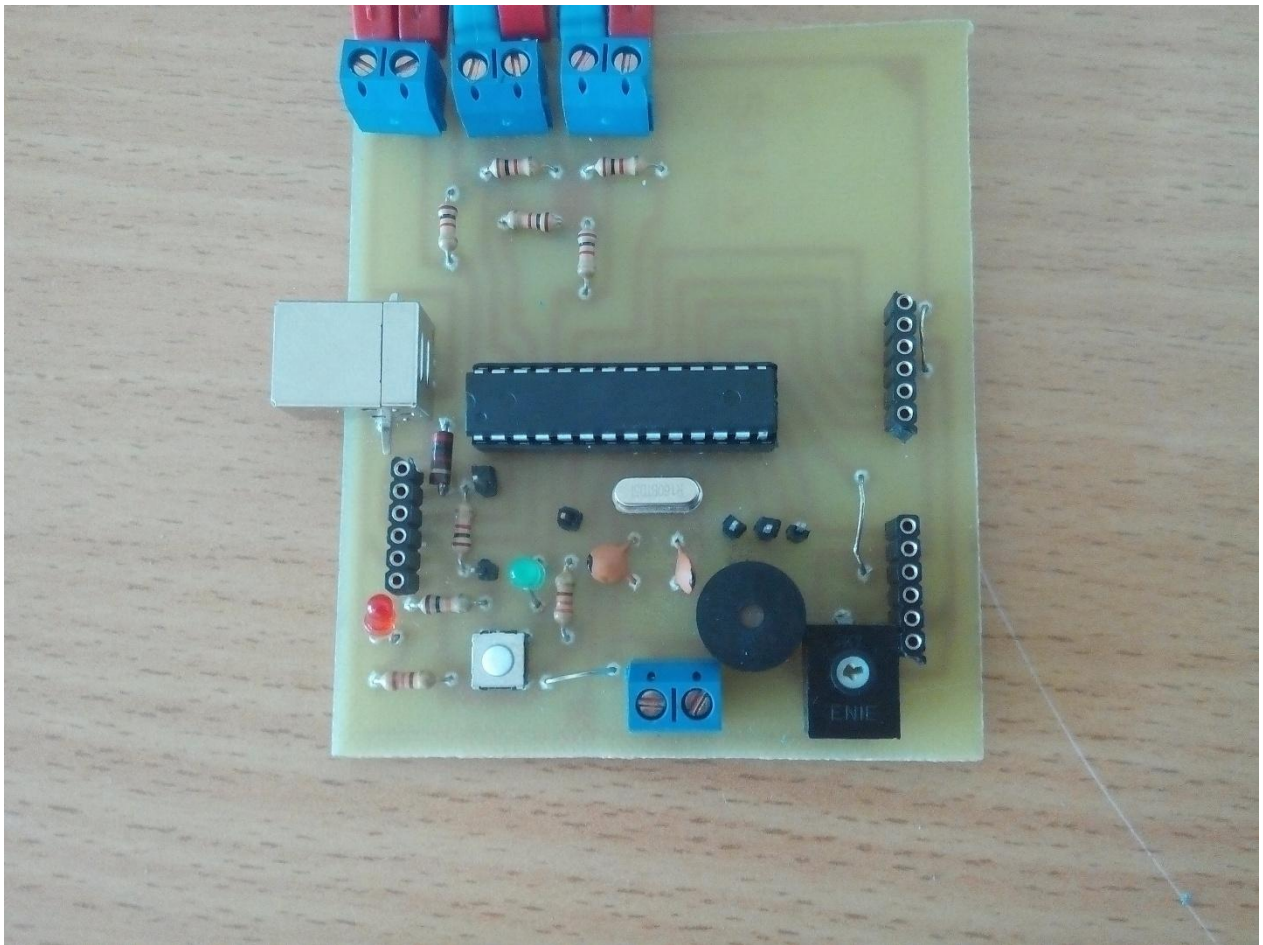


Figure 6 : Image de la carte électronique.

Pour avoir réalisé notre carte électronique ci-dessus nous nous sommes inspiré de la carte Arduino uno qui est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressources disponibles en ligne.

Arduino est un projet open source (les plans, les schémas, etc. sont accessibles et libres de droits) la grande communauté d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ces questions et apporte un travail énorme de documentation au projet.

Les projets Arduino peuvent être autonomes, comme ils peuvent communiquer avec d'autres logiciels installés sur l'ordinateur tel que Flash, Processing ou MaxMPS, Matlab).

2.1) conception du circuit imprimé de la carte électronique :

Nous avons commencé par faire le typon de notre carte à l'aide du logiciel TCI qui est un logiciel pour les électroniciens, destiné à la conception de circuits imprimés. Ce logiciel permet de tracer son circuit simple ou double face rapidement et d'imprimer chacune des couches sur papier ou sur transparent bureautique (pour obtenir un film directement exploitable).

Ce logiciel est particulièrement bien adapté aux prototypes, et aux petites séries, donc convient bien aux amateurs, labos et enseignants.

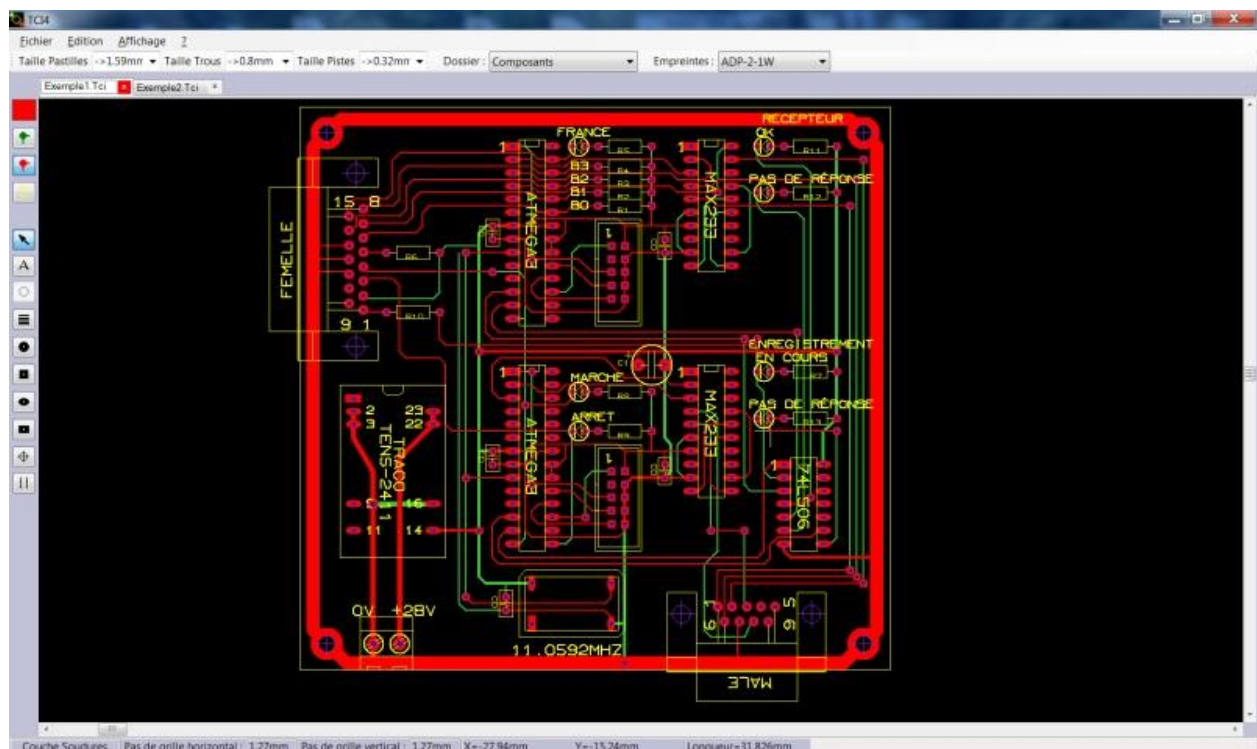


Figure7 : Logiciel TCI tracé de circuit imprimé version 4.4

Le circuit imprimé de notre carte est montré en image ci-dessous :

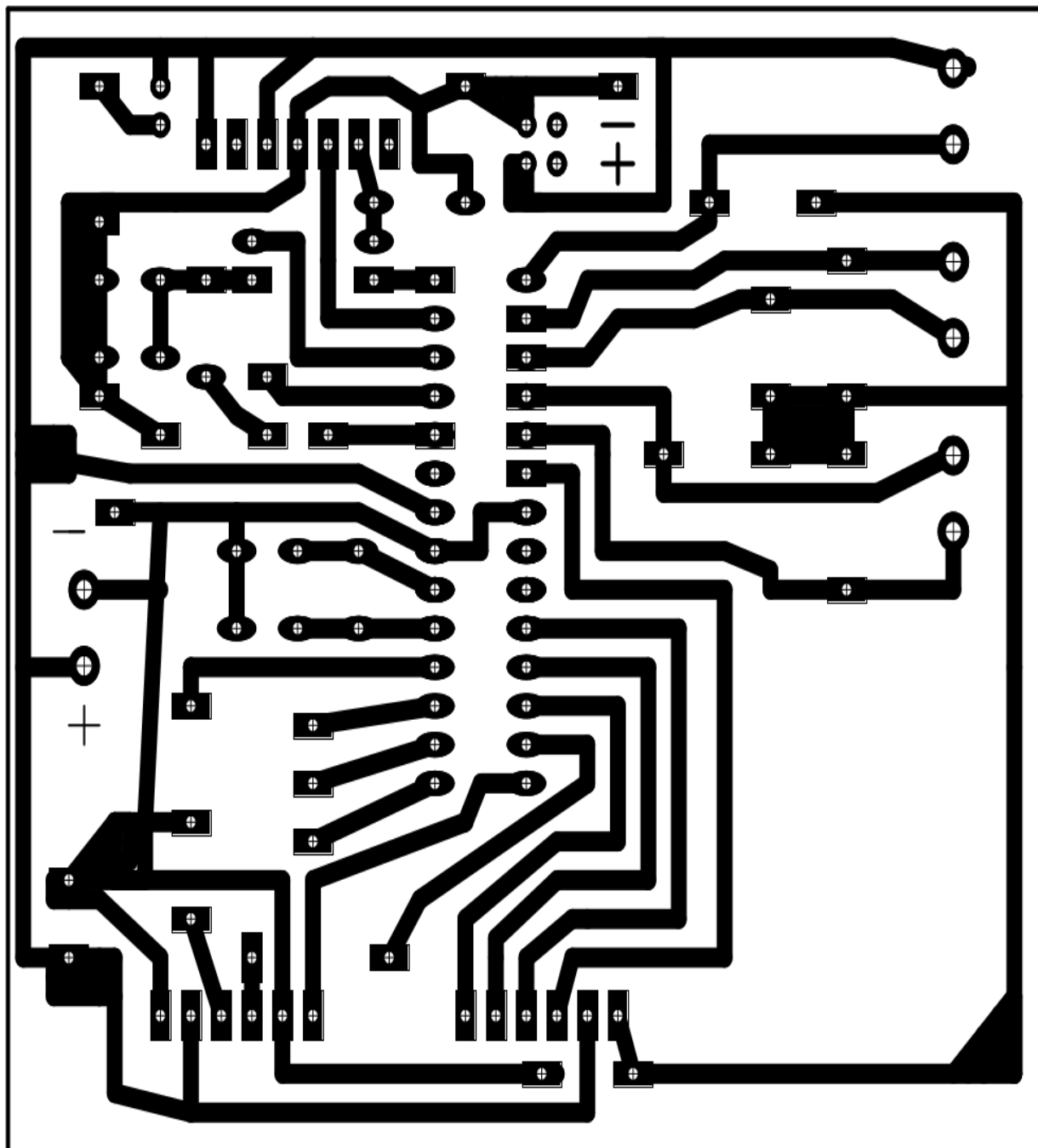


Figure 8: circuit imprimé de notre carte électronique.

2.2) Principes de réalisation :

Les montages électroniques se réalisent toujours sur un circuit imprimé, c'est-à-dire un support isolant sur lequel des pistes de cuivre réalisent la quasi-totalité des connexions entre les composants qu'il supporte. Le matériau utilisé est presque toujours du verre époxy c'est-à-dire un isolant à base de fibres de verre compressées et imprégnées de résine époxy. Une pellicule de cuivre très fine ($< 100 \mu\text{m}$) est fortement collée sur le support.

Le traitement consiste à supprimer le cuivre là où il n'y a pas de liaison à assurer et à le laisser au contraire là où on doit relier les composants. Le cuivre est supprimé par attaque chimique, généralement au perchlorure de fer. L'action de supprimer le cuivre par attaque chimique s'appelle la gravure. Toute la difficulté réside dans le fait qu'il faut protéger le cuivre là où il doit rester. On peut utiliser un feutre spécial avec une encre résistante mais il faut alors tracer le dessin des pistes à la main, ce qui ne peut pas convenir pour les circuits imprimés compliqués. De plus, cette méthode ne permet pas de réaliser les circuits imprimés en série : si vous voulez 20 circuits imprimés identiques, il faudra redessiner 20 fois le typon à la main, sur chaque plaque !

La méthode la plus utilisée aujourd'hui pour réaliser un circuit imprimé consiste à utiliser un Typon représentant le tracé des pistes que l'on voudra faire apparaître sur le circuit.

Ce typon pourra être dessiné à la main sur une feuille de papier calque, ou tracé sur ordinateur avec un logiciel spécialisé puis imprimé sur un transparent. L'avantage d'utiliser un typon est double : on pourra facilement fabriquer plusieurs circuits imprimés identiques « à la chaîne », et il est possible de réaliser des circuits imprimés complexes, possédant beaucoup de composants, et demandant une grande précision.

3) choix et rôles des composants:

3.1) Le microcontrôleur ATMEGA 328:

La figure ci-dessous nous montre le microcontrôleur ATMEGA 328 :



Figure9 : Le microcontrôleur de la carte électronique.

C'est le processeur de la carte, il s'occupe de tout ce qui est calculs, exécution des instructions du programme et gestion des ports d'entrées/sorties.

L'ATMEGA 328 a 32Ko de mémoire FLASH pour stocker le programme (dont 0.5Ko également utilisé par le boot loader), l'ATMega328 a également 2Ko de mémoire RAM (volatile) et 1Ko d'EEPROM (non volatile: mémoire qui peut être lue à l'aide de la librairie EEPROM).

Pour les besoins de notre carte le microcontrôleur devait être petit et bon, nous avons donc opté pour le circuit ATMega328 qui réunit ces deux conditions. Il est associé à des ports entrées/sorties qui permettent à l'utilisateur de communiquer avec l'environnement

3.2) L'afficheur LCD 4*20:

La figure ci-dessous est une photo prise de l'afficheur LCD 4*20 :



Figure 10 : Image de l'afficheur LCD 4*20

Il permet d'afficher des informations diverses. Il s'agit d'un afficheur 20 caractères sur 4 lignes avec affichage rétro éclairé bleu.

3.2.1) Description des broches et connexion de l'afficheur LCD avec un microcontrôleur :

Le dialogue avec un μC ou μP se fait par un bus de données de 8 bits ou de 4 bits. Les échanges d'informations sont synchronisés par des signaux de commandes:

- R/W (Lecture/écriture).
- RS (Registre Select : Registre de sélection).
- E (Enable : Mémorisation).

3.2.2) Caractéristiques de l'afficheur et fonctionnement :

L'afficheur utilisé dispose de :

- 4 lignes de 20 caractères.
- Une RAM (DDRAM : DATA RAM) de 80 caractères correspondant.
- Une RAM (CGRAM : CHARACTER GRAPHIC RAM) permettant de créer de nouveaux caractères.

Fonctionnement de l'afficheur :

- L'afficheur dispose de deux registres permettant de gérer celui-ci:
- Le registre d'instruction I R: (Instruction Register) : C'est le registre de contrôle, suivant la valeur que l'on met dedans l'afficheur exécute des opérations de configurations, exemple: "effacement de l'écran". Il permet aussi de positionner le curseur parmi les 80 adresses de la DDRAM, ou des 16 adresses de la CGRAM.
- Le registre de données D R: (Data Register); Suivant la valeur que l'on met dedans l'afficheur peut:
 - *Afficher un caractère (Code ASCII ou spécifiques).
 - *Créer une ligne d'une matrice d'un nouveau caractère.

L'accès de ces registres est fonction des valeurs des signaux R/W et RS.

RS	R/W	Registre sélectionné
0	0	Ecriture dans IR : Registre de contrôle ou d'instruction
0	1	Lecture de IR : Registre de contrôle ou d'instruction
1	0	Ecriture de DR : Registre de données
1	1	Lecture de DR : Registre de données

L'utilisation de l'afficheur LCD sous Arduino :

1- Importer la librairie qui permet d'utiliser la LCD :

```
#include <Liquide crystal.h >
```

2- Declarer les broche a utiliser ;

```
Liquide crystal lcd ( , , , , , ) ;
```


```
Void setup(){
```

3-Declarer le type d'affichage ;

```
Lcd .begin (20,4) ;
```

4-indique l'endroit on positionneur le curseur :

```
Lcd set cursor(0,0) ;
```



colonne ligne

5-afficher un message :

```
Lcd.print ('message') ;
```

```
Void loop(){
```

On peut l'utiliser pour faire clignoter Le message en utilisant les instructions

led .display() ;et led .no display() ;

Ou pour défiler le texte de gauche droite ou de droite a gauche en utilisant lcd scroll display left() ; ou scroll Display wright() ;et contrôler La vitesse de déplacement avec delay() ; .

3.2.3) Mode d'utilisation de l'afficheur :

On utilise l'afficheur pour afficher :

- Message d'accueil.
- Moteur allumé.
- Moteur éteint.
- Augmentation de vitesse.
- Nombre de pièces entrantes.
- Nombre de petites pièces.
- Nombre de grandes pièces.

3.3) a)Le potentiomètre :

Les potentiomètres se sont les petits modules qui servent dans la télécommande à mesurer la valeur des joysticks, plus généralement, ils mesurent la rotation d'un axe, auquel on peut ajouter des graduations pour une meilleure utilisation.

Mais bien heureusement l'utilisation des potentiomètres ne s'arrête pas là,
Dans notre cas nous l'utilisant pour régler le contraste de l'afficheur LCD.



Figure 11 : image du potentiomètre

b) Le buzzer piézo-électrique :

Délivre un bip sonore bien distinct pour exprime la validation de la commande de l'utilisateur.



Figure12 : Buzzer piézo-électrique

Le transducteur exprimé sur la figure ci-dessus est un composant réversible ; il peut aussi bien être utilisé en capteur de chocs ou de vibrations qu'en actionneur pouvant émettre des sons stridents parfois modulables.

Dans notre cas nous l'avons utilisé pour donner un bip lors :

- allumage du moteur.
- réinitialisation du moteur .

c) Le Quartz :

En électronique, un quartz est un composant qui possède comme propriété utile d'osciller à une fréquence stable lorsqu'il est stimulé électriquement. Les propriétés piézoélectriques remarquables du minéral_de_quartz permettent d'obtenir des fréquences d'oscillation très précises, ce qui en font un élément important en électronique_numérique ainsi qu'en électronique_analogique.



Figure13 : Le quartz.

Pour notre carte nous avons utilisé un Quartz de 16 MHz pour donner la vitesse d'exécution nécessaire pour le microcontrôleur.

3.4) a) Les résistances :

La résistance s'oppose au passage du courant, proportionnellement à sa résistance exprimé en OHM. Un code de couleurs ci-dessous permet de reconnaître cette valeur.

Symbole européen



Code des couleurs des résistances au delà de 1000 Ohms, on parle en KiloOhms, par exemple 10 K est 10 KiloOhms, puis en MegaOhms notés M

The diagram shows a resistor with five color bands. The first band is red (1), the second is red (1), the third is yellow (10), the fourth is black (0), and the fifth is gold (tolerance). The value is 11000 Ω (11 kΩ) with a 5% tolerance.

Chiffre	0	1	2	3	4	5	6	7	8	9	Or	Argent
Multiplieur	1	10	10 ² 100	10 ³ 1000	10 ⁴ 10000	10 ⁵	10 ⁶					
Précision	20%										5%	10%

Figure14: Code des couleurs des résistances.

Nous avons utilisé :

- une variable utilisé avec le potentiomètre pour le contraste.
- 5 résistances pour mettre a zéro la tension dans les fils.
- ainsi que d'autres utilisé comme deviseurs de tensions.

b) Les condensateurs :

Les condensateurs peuvent stocker un peu de courant si on les charge, mais comme un tonneau percé, ils renvoient ce courant instantanément s'ils sont branchés à un organe consommateur de courant.

Ils peuvent être polarisés ou non, dans des boitiers très divers, Leur valeur s'exprime en FARAD (F) son Symbole :



Utilisé pour lisser la tension.

c) Les LEDs :

La diode électroluminescente (LED) émet de la lumière .Elle est polarisée : la patte ''+' est la plus longue, l'autre patte est la patte ''-'

Les boches numérique de l'Arduino lorsque elles sont configurées en sortie et quelles sont à l'état 1 ou haut (HIGH), fournissent une tension de 5 volts, supérieur à ce que peut accepter une LED courante sauf certaines LEDs.

Les LEDs doivent donc être couplées en série avec une résistance

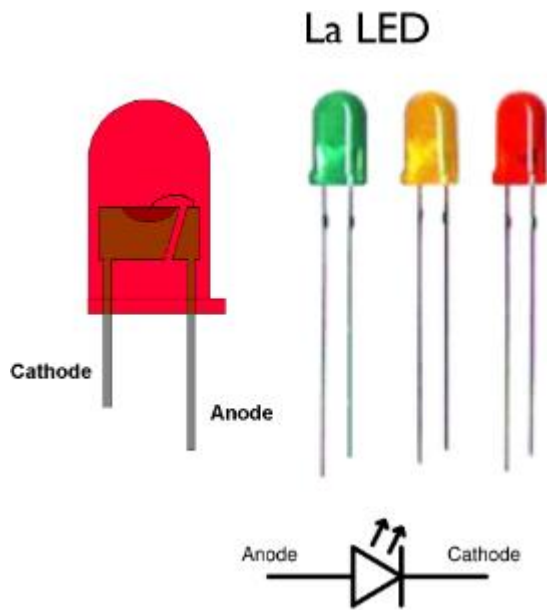


Figure15 : La LED

3.5) La carte relais:

Une Carte relais est un organe électrique permettant de dissocier la partie puissance de la partie commande. Elle permet l'ouverture et la fermeture d'un circuit électrique par un second circuit complètement isolé.

La tension sur les pins I/O de l'Arduino est de 5V, un peu limite quand on sait que la tension de l'appareillage électrique de la maison est de 240V, voir 400V si vous disposez du triphasé.

Les relais sont utilisés très régulièrement, par exemple pour démarrer un moteur, commander une prise, une lumière, réguler le chauffage.

Vu que notre carte électronique fait la fonction d'une carte de commande donc on n'aura pas besoin d'une carte de puissance. Pour cela nous avons opté pour une carte relais pour pouvoir donner suffisamment d'énergie pour fonctionner les moteurs qui ont besoin d'une tension de 220 Volts pour leur fonctionnement.

Le principe est simple, selon le relai qui est actionné par notre carte électronique, le moteur qui est relié à ce même relai s'actionne.

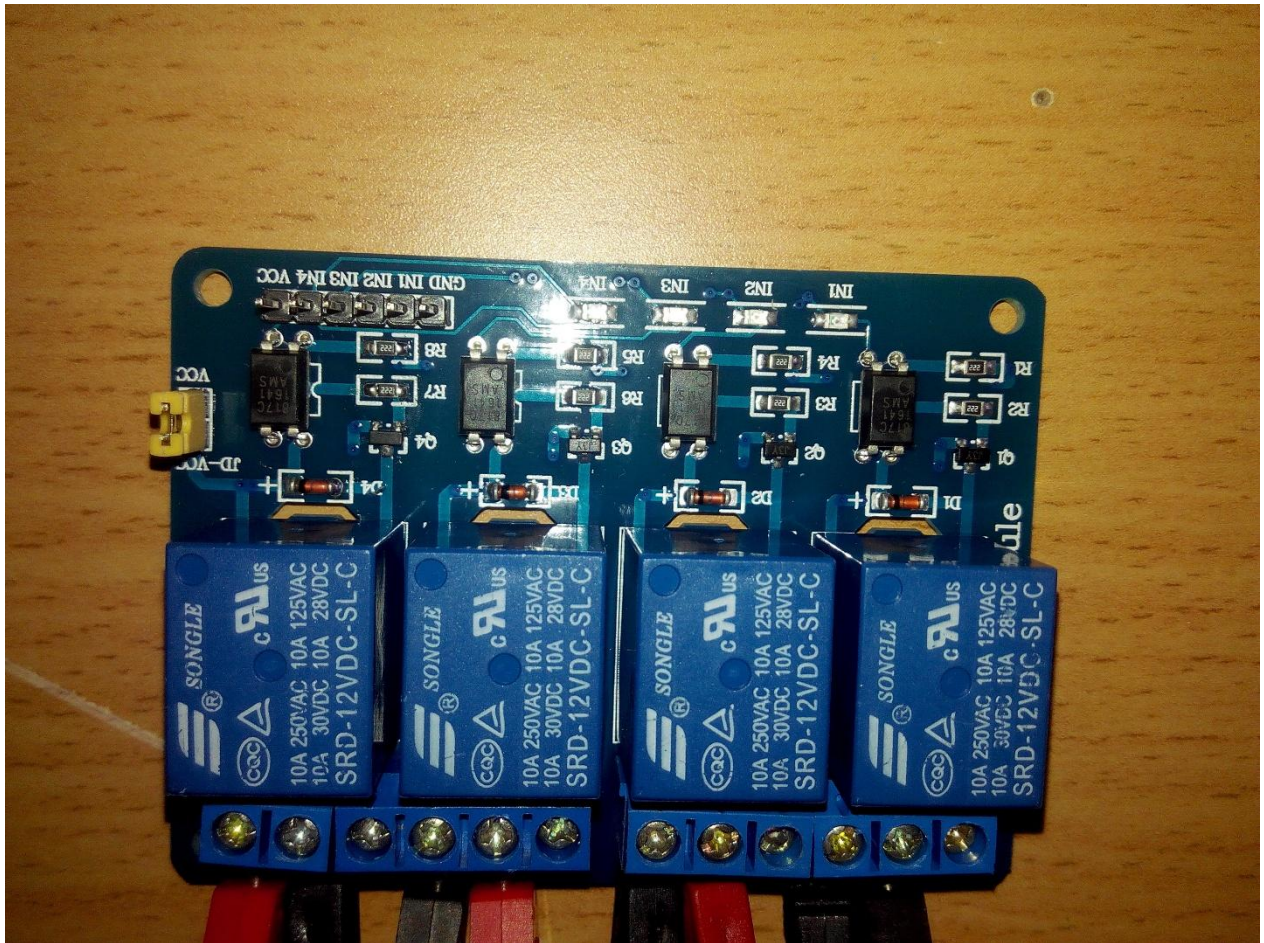


Figure 16: Image de la carte relais 4 canaux avec shield.

3.6) Le connecteur USB :

Il est utilisé pour le transfert des données entre la carte électronique et le PC, et aussi pour l'alimentation électrique.

L'image ci-dessous montre le brochage de notre système de contrôle au convoyeur CU-4001.

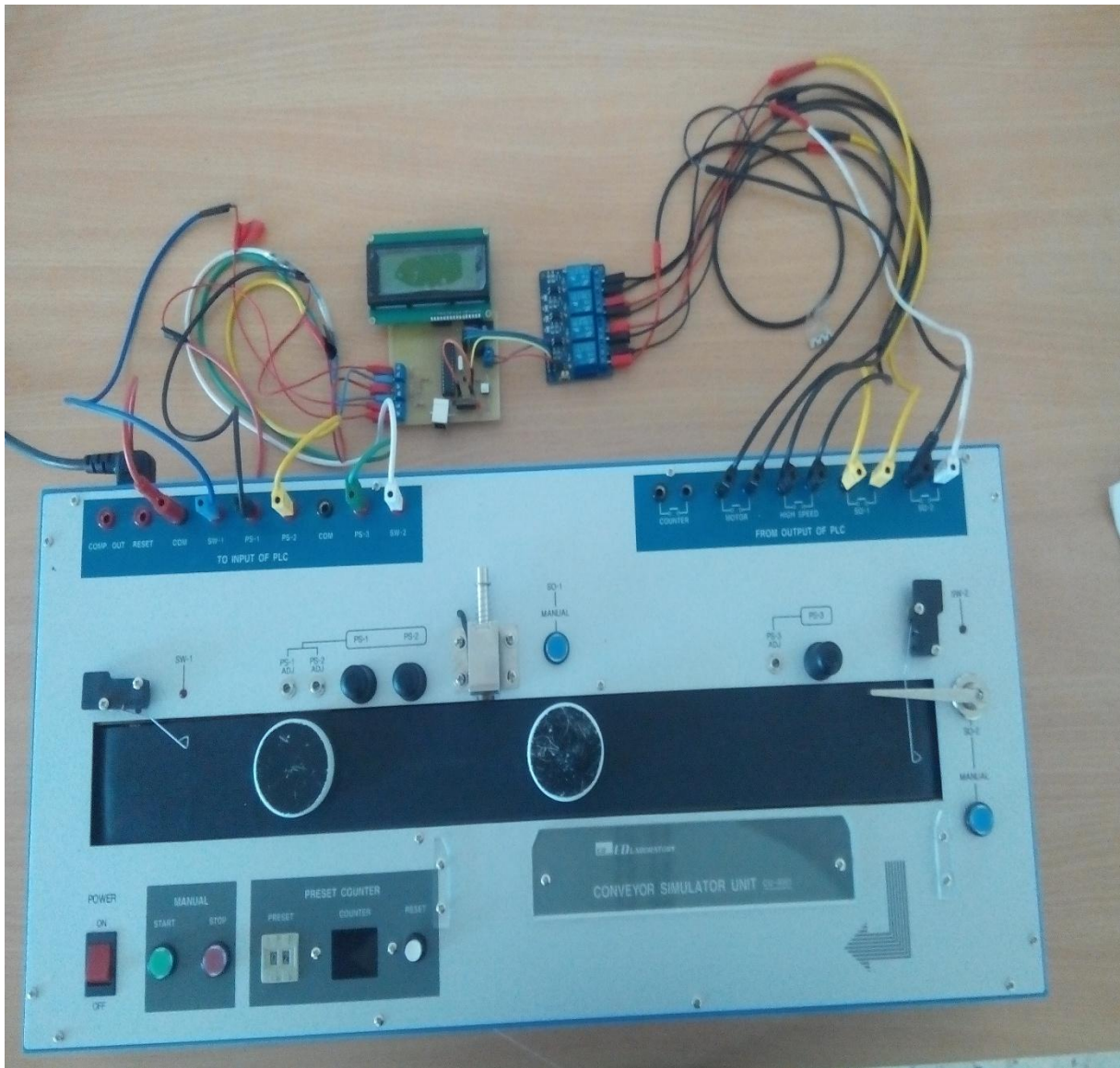


Figure 17 : Aperçue des liaisons entre la carte et le convoyeur.

Conclusion :

Dans ce chapitre, nous avons choisi et passé en revue l'architecture matérielle et logicielle de notre système de contrôle du convoyeur ainsi que les principales caractéristiques des composants électroniques et qui constitue notre système. De même, nous avons présenté leurs configurations externes et internes et la manière avec laquelle, chacun d'eux, va être connecté avec les autres.

CHAPITRE 3 : présentation du logiciel de programmation, du langage de programmation et du programme contrôlant le convoyeur.

1) présentation du logiciel de programmation :

1.1) introduction à l'Arduino :

Arduino est un projet créé par une équipe de développeurs, composés de six individus, cette équipe a créé le « système Arduino ».

C'est un outil qui va permettre au débutant, amateurs ou professionnel de créer des systèmes électroniques plus ou moins complexes.

Le système Arduino, nous donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, la programmation de systèmes électroniques. Le gros avantage de l'électronique programmée c'est qu'elle simplifie grandement les schémas électroniques et par conséquent, le coût de la réalisation, mais aussi la charge de travail à la conception d'une carte électronique.

L'utilité est sans doute quelque chose que l'on perçoit mal lorsque l'on débute, mais une fois que vous serez rentré dans le monde de l'Arduino, vous serez fasciné par l'incroyable puissance dont il est question et des applications possibles !

1.2) Le logiciel Arduino (IDE Arduino)

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java inspirée du langage Processing.

L'IDE permet d'écrire, de modifier un programme ou de le convertir en une série d'instructions compréhensibles pour la carte.

C'est un logiciel de programmation par code, qui contient une cinquantaine de commandes différentes. À l'ouverture, l'interface visuelle du logiciel ressemble à ceci :
Des icônes de commande en haut, une page blanche vierge, une bande noire en bas.

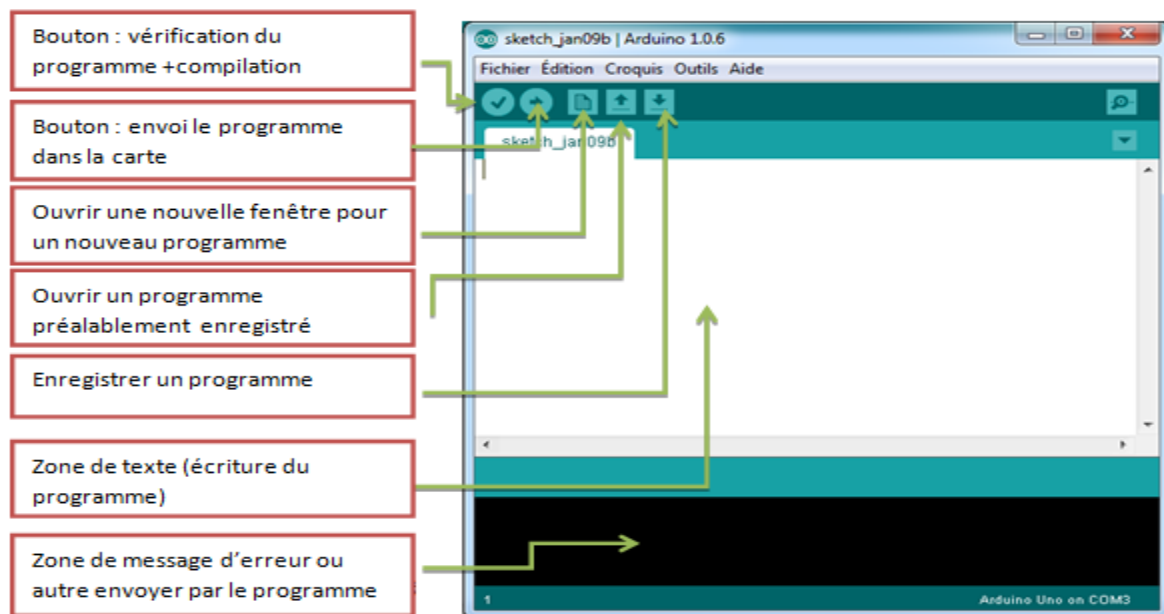


Figure 18 : Les différentes parties de la fenêtre principale du logiciel Arduino.

- ✓ On conçoit d'abord un programme avec le logiciel Arduino ;
- ✓ On vérifie ce programme avec le logiciel (compilation) ;
- ✓ Des messages d'erreur apparaissent éventuellement... On corrige puis on vérifie à nouveau ;
- ✓ On retire le programme précédent sur la carte Arduino (cette opération se fait automatiquement lors de la télétransmission (l'upload) du nouveau programme ;
- ✓ On télécharge (upload) ce programme sur la carte Arduino ;
- ✓ L'exécution du programme sur la carte est automatique quelques secondes plus tard ou à ses prochains allumages.

2) Présentation du langage de programmation

2.1) C Arduino

2.1.1) Présentation de l'espace développement Intégré (EDI) Arduino:

Comme cela a été dit, la carte Arduino présente le noyau de notre système. Elle est programmée par un logiciel compatible appelé **Arduino EDI** (Espace de développement intégrée) qui est une application écrite en Java inspiré du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

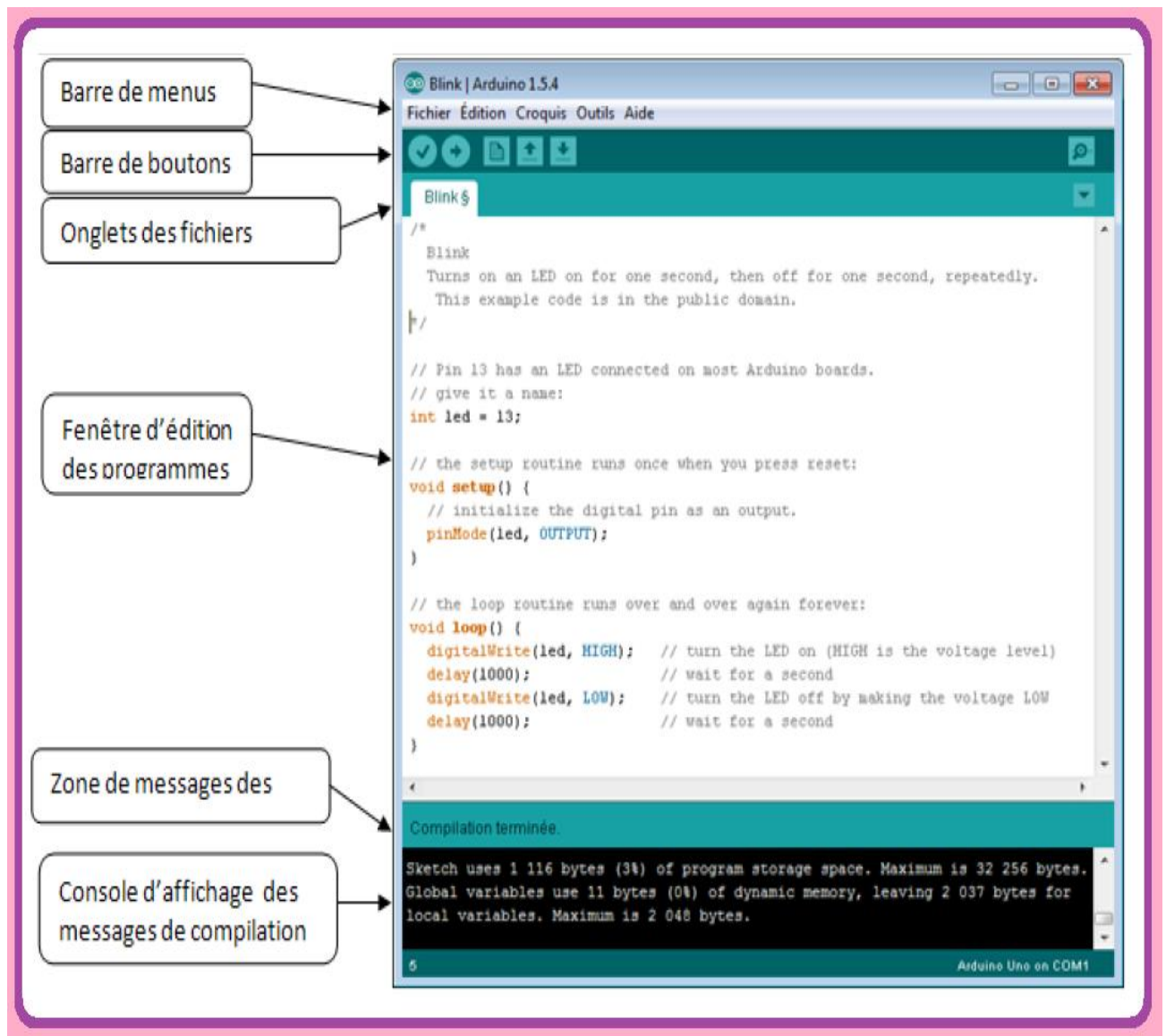
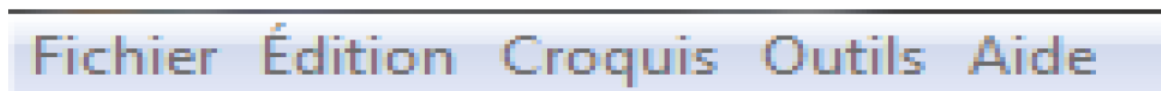


Figure 19 : Les différentes parties de la fenêtre principale de logiciel Arduino.

2.1.2) Différents compositions de l'espace de développement intégré (EDI):

Une barre de menus:






Cette barre contient les icones suivants:

- ✓ **File (Fichier):** ce menu contient les différentes options de créations d'ouverture de sauvegarde, d'impression du programme, ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.

- ✓ **Edite (Editer)** : ce menu contient les options de copier, coller et sélectionner les options de recherche.
- ✓ **Sketch (programme ou séquence)** : ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de librairies ou de fichiers.
- ✓ **Tools (outils)** : c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé.
- ✓ **Help (Aide)** : ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau de logiciel Arduino.



Figure 20 : Barre de boutons pour le programme C Arduino

- ✓ **Vérifier/compiler** : Vérifie le code à la recherche d'erreur,
- ➔ **Transférer vers la carte**: compile votre code et le transfère vers la carte Arduino.
-  **Nouveau**: Crée un nouveau code (ouvre une nouvelle fenêtre),
-  **Ouvrir**: ouvre la liste de tous les programmes dans le "livre de programmes",
Enregistrer: Enregistre un programme,
-  **Moniteur série**: Ouvre la fenêtre du moniteur (on terminal) série.

a) Structure d'un programme Arduino:

- ✓ **Un éditeur:**
C'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation.
- ✓ **Une zone de message:**
Comme son nom l'indique, cet espace est utilisé par l'EDI pour communiquer à l'utilisateur des informations sur l'état du programme et des actions en cours.
- ✓ **Une console texte:**

Affiche le message concernant le résultat de la compilation du programme (il indique si le programme comporte des erreurs).

b) Programme Arduino:

Description de la structure d'un programme Arduino:

Le langage Arduino est basé sur le langage C/C++ :

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne, la carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure se décompose en trois parties :

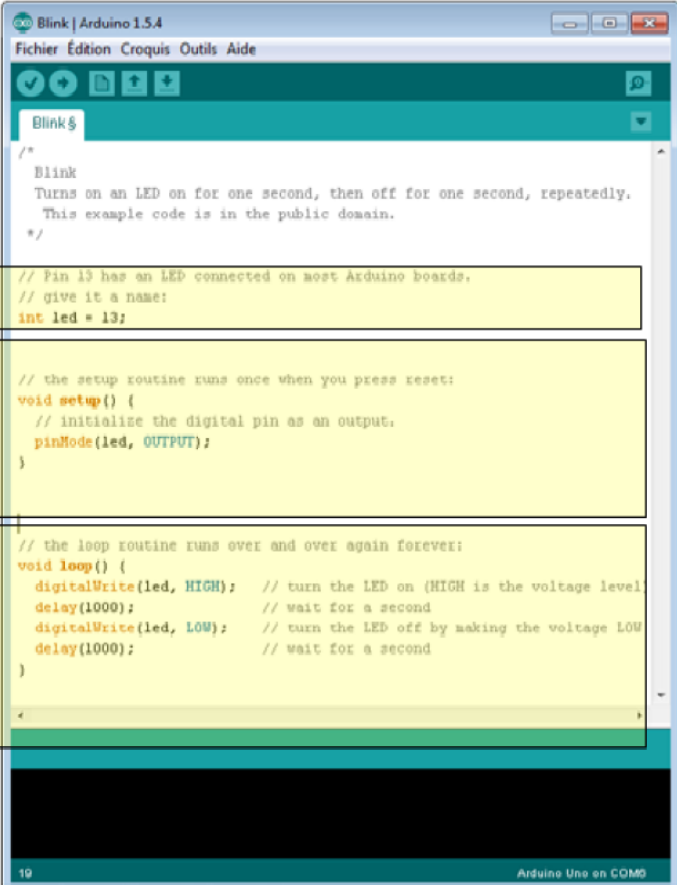
	
Définition des constantes et variables globales	<pre>// Pin 13 has an LED connected on most Arduino boards, // give it a name: int led = 13;</pre>
Fonction principale : <i>VOID SETUP()</i> Initialisation des ressources de la carte Configuration des entrées/sorties Définition de la vitesse de fonctionnement	<pre>// the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); }</pre>
Fonction boucle : <i>VOID LOOP()</i> Description du fonctionnement général du programme Gestion des interactions entre les entrées/sorties Cette partie sera exécutée en boucle	<pre>// the loop routine runs over and over again forever: void loop() { digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second digitalWrite(led, LOW); // turn the LED off by making the voltage LOW delay(1000); // wait for a second }</pre>

Figure 21 : Différentes composantes du programme Arduino

Le code minimal:

Avec Arduino, nous devons utiliser un code minimal lorsque l'on crée un programme. Ce code permet de diviser le programme que nous allons créer en deux parties, soit void setup () ou void loop () décrites en haut.

c) Jeux d'instructions du langage Arduino:

Syntaxes du Programme:

- Chaque instruction se termine par un ";",
- Les **accolades** "{" et "}" sont les "conteneurs" du code du programme, elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades,
- Les **commentaires** sont des lignes de texte incluses dans le programme et qui ont pour but de nous informer de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur. Les commentaires sont précédés des caractères "//" ou bien encadrés par " /* " et " */".
- Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires,
- Un nombre en binaire doit être précédé de "B",
- un nombre en hexadécimal doit être précédé par les caractères "0x".

Commandes du programme Arduino:

❖ Structure générale

- **void setup()** (configuration-préparation)
- **void loop()** (exécution)

❖ Contrôle et conditions

- **if** (si...)
- **if...else** (si...alors...)
- **for** (pour...)
- **switch case** (dans le cas ou...)
- **while** (pendant que...)

❖ Opérations de comparaison

- **=** (équivalent à)
- **!=** (différent de)
- **<=** (inférieur ou égal à)
- **>=** (supérieur ou égal à)

❖ **Operations booléennes**

- **&** (Et)
- **!=** (et pas)
- **||** (ou)

❖ **Autres commandes**

- **//** = (Commentaire simple ligne)
- **/**/** = (Commentaire multi-lignes),
- **#define** = (donner une valeur à un nom)

❖ **Variables**

- **char** = (variable 'caractère')
- **int** = (variable 'nombre entier')
- **long** = (variable 'nombre entier de très grande taille')
- **string** = (variable 'chaîne de caractères')
- **array** = (tableau de variables)

Variables:

❖ **Niveaux logiques des connecteurs numériques**

- **HIGH** = (état 1),
- **LOW** = (état 0),
- **INPUT** = (configuré en entrée),
- **OUTPUT** = (configuré en sortie).

Fonctions:

❖ **Entrées-sorties numériques**

- **pinMode(broche, état)** (configuration des broches en entrée ou en sortie),
- **digitalWrite(broche, état)** (écrire un état sur une broche numérique),
- **digitalRead(broche)** (lire un état sur une broche numérique),
- **unsigned long pulseIn (broche, état)** (lire une impulsion sur broche numérique).

❖ **Entrées analogiques**

- **int analogRead(broche)** (lire la valeur d'une broche analogique)

- `analogWrite(broche,valeur)` (PWM: écrire une valeur analogique sur les broches 5, 6, 9, 10 et 11)

❖ Gestion du temps

- `unsigned long millis()` (temps de fonctionnement du programme),
- `delay(ms)` (attente, en millisecondes),
- `delayMicroseconds(us)` (attente, en microsecondes).

d) Etapes à suivre pour programmer la carte:

Ecriture du programme:

Cette première étape consiste à saisir le programme.

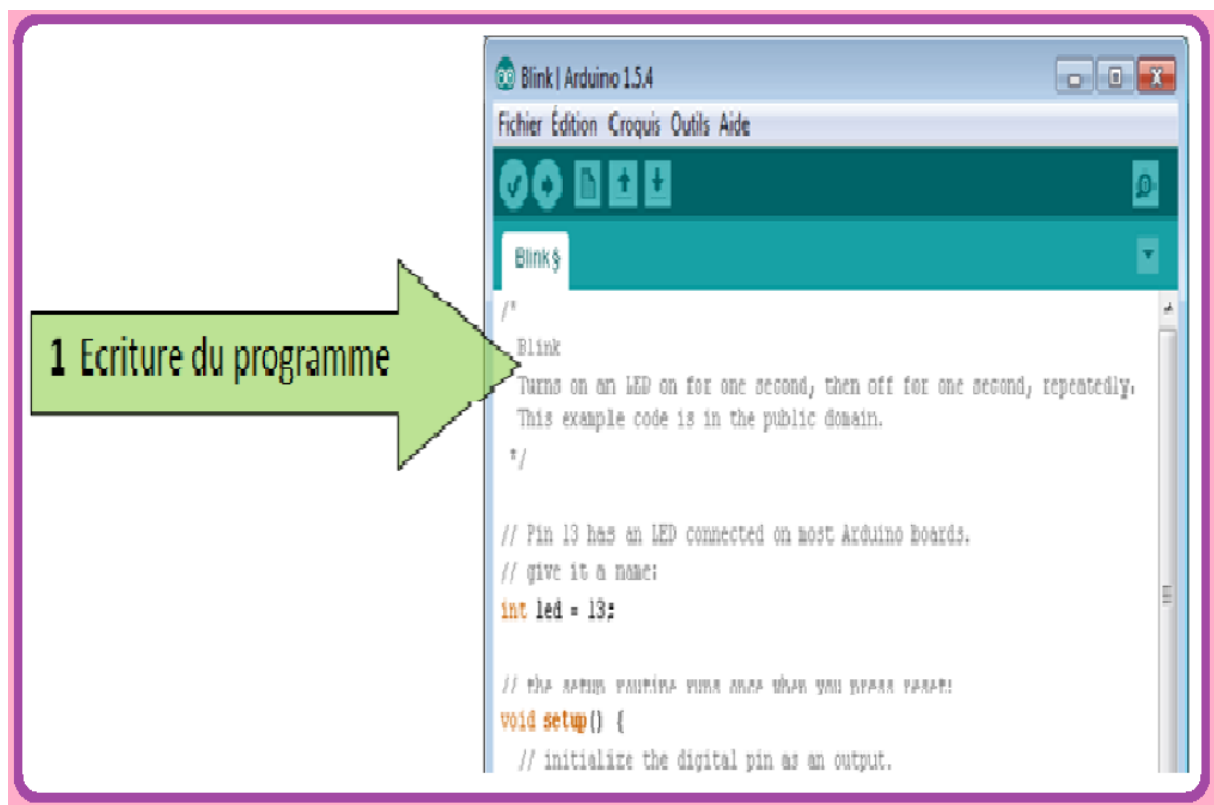


Figure 22 :L'écriture du programme

Compilation du programme:

Dans cette partie, on vérifie si le code contient des erreurs de syntaxes, en cas d'anomalie dans le programme, le compilateur nous renseigne sur le type d'erreur et la ligne ou elle se trouve, (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menus).

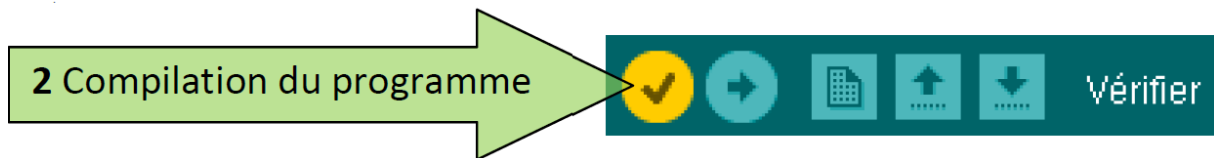


Figure 23 : Compilation du programme

Si aucune erreur n'est détectée, le programme est compilé et le message "Compilation terminée" apparaît, suivi de taille du programme.

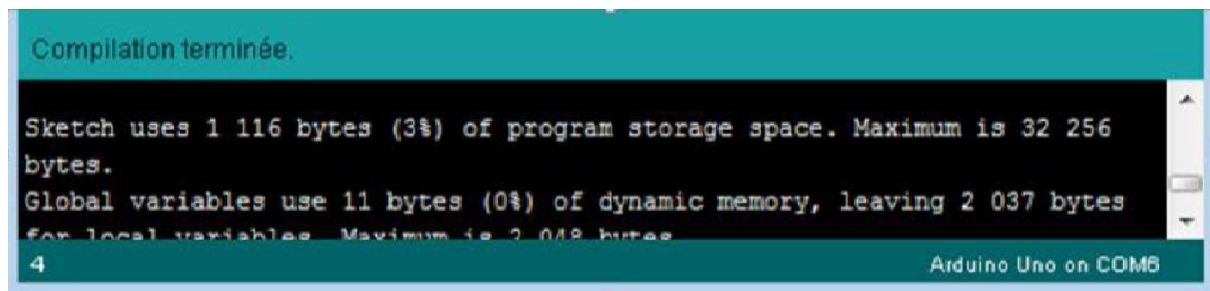


Figure 24 : Console d'affichage

Sélection de la cible (type de carte):

Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type Arduino (cible) depuis le menu **Outils>Board**.

La carte doit évidemment être connectée à l'ordinateur via un câble USB.

Vérifions que c'est bien le nom "Arduino nano" qui est coché, si ce n'est pas le cas, on le coche.

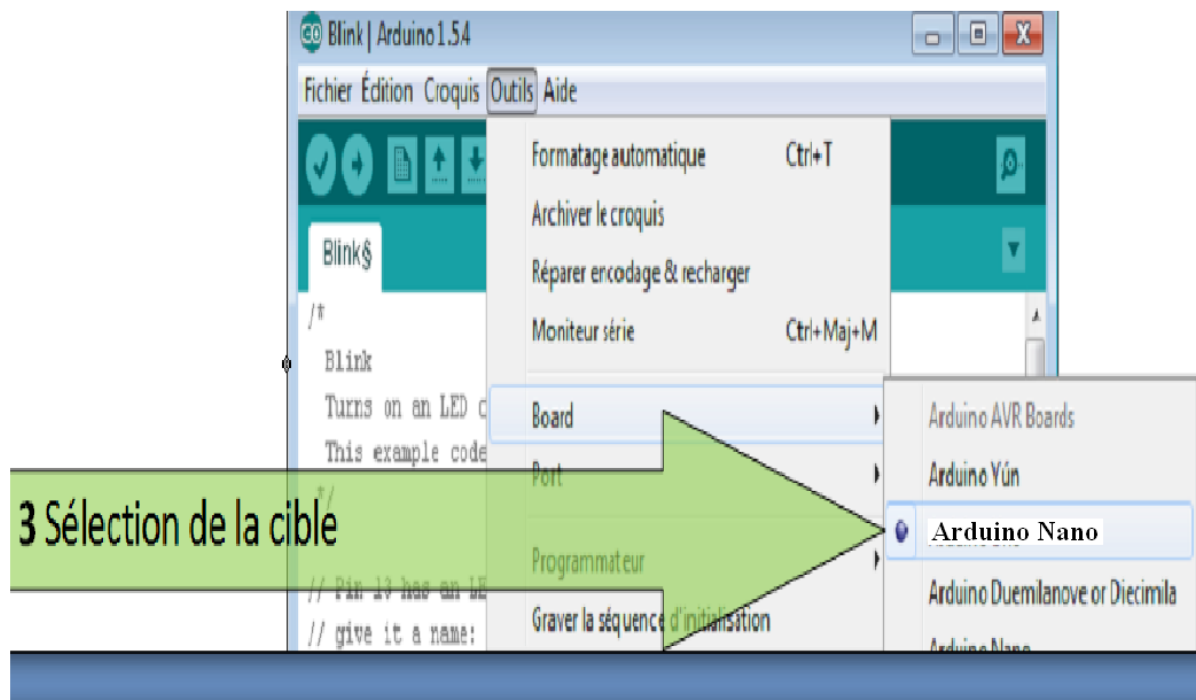


Figure 25 : Sélection de la cible

Sélection du port:

Sélectionner le port série sur lequel est branché cette carte (depuis le menu **Outils>Port**)

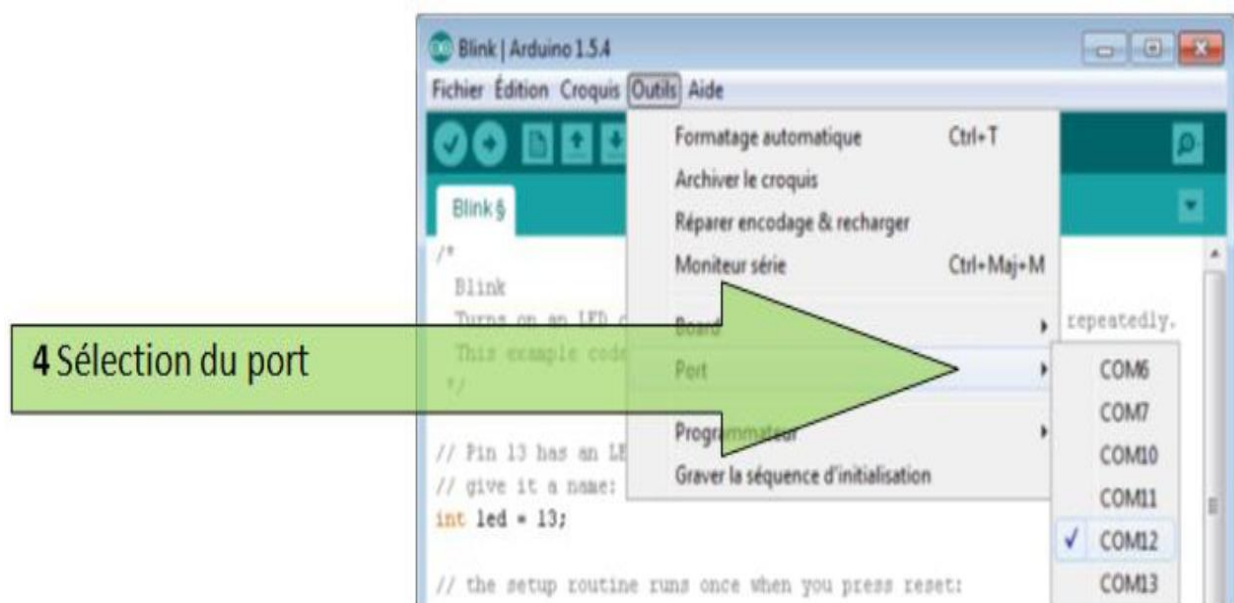


Figure 26 : Sélection du port

Transfert du programme:

Cela se fait en cliquant sur le bouton.

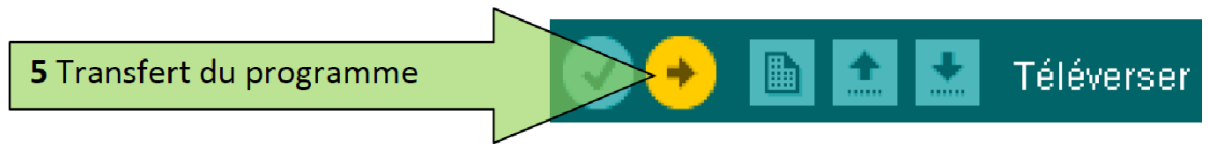
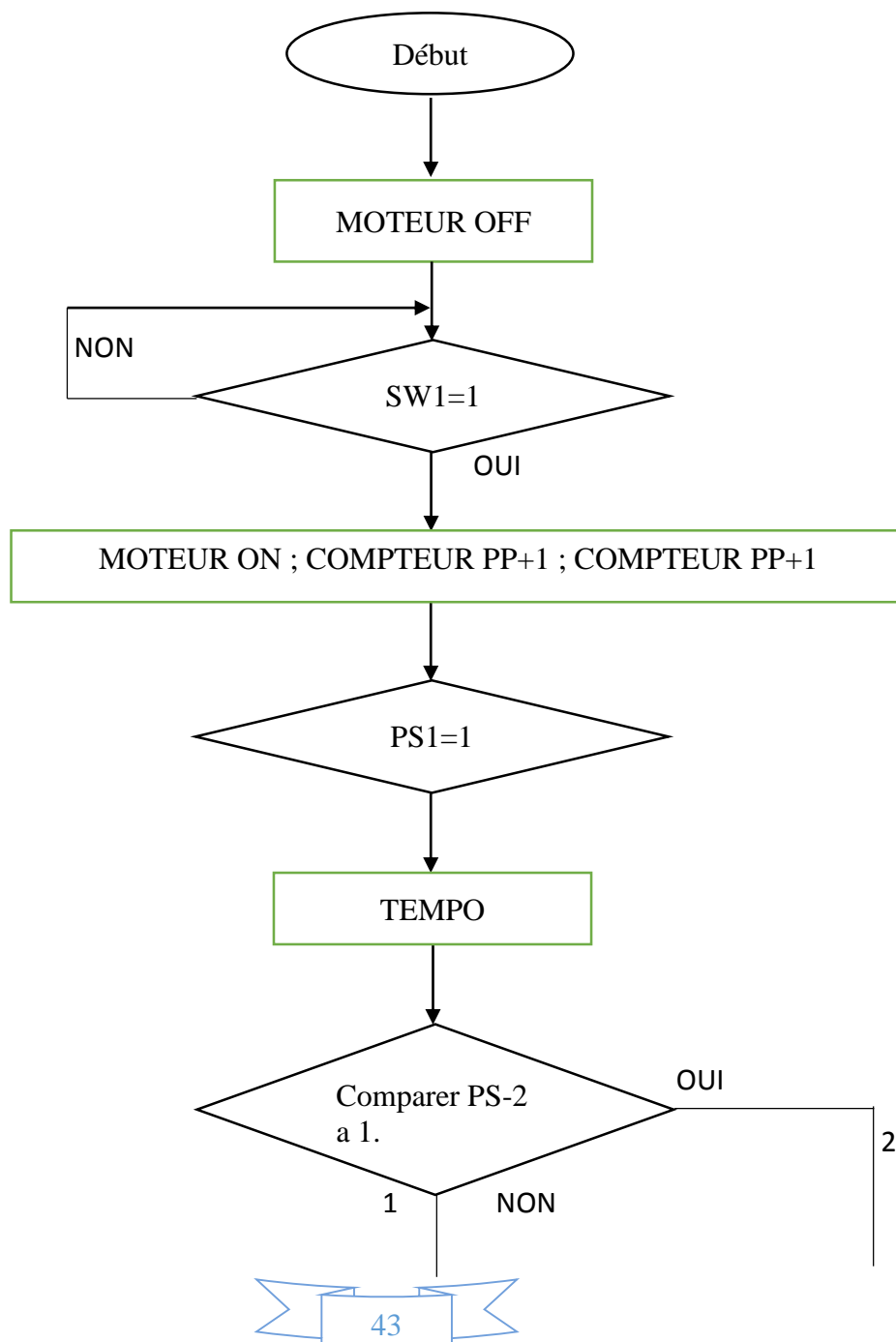
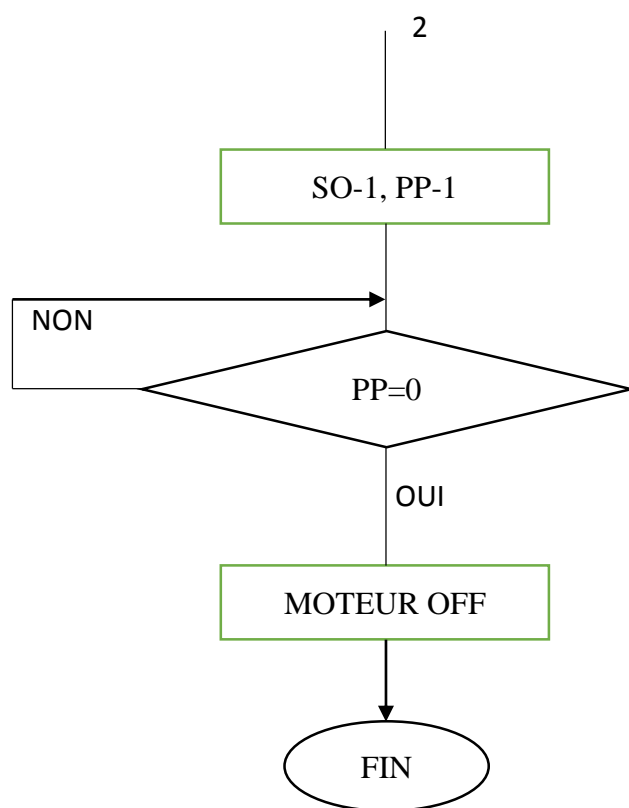
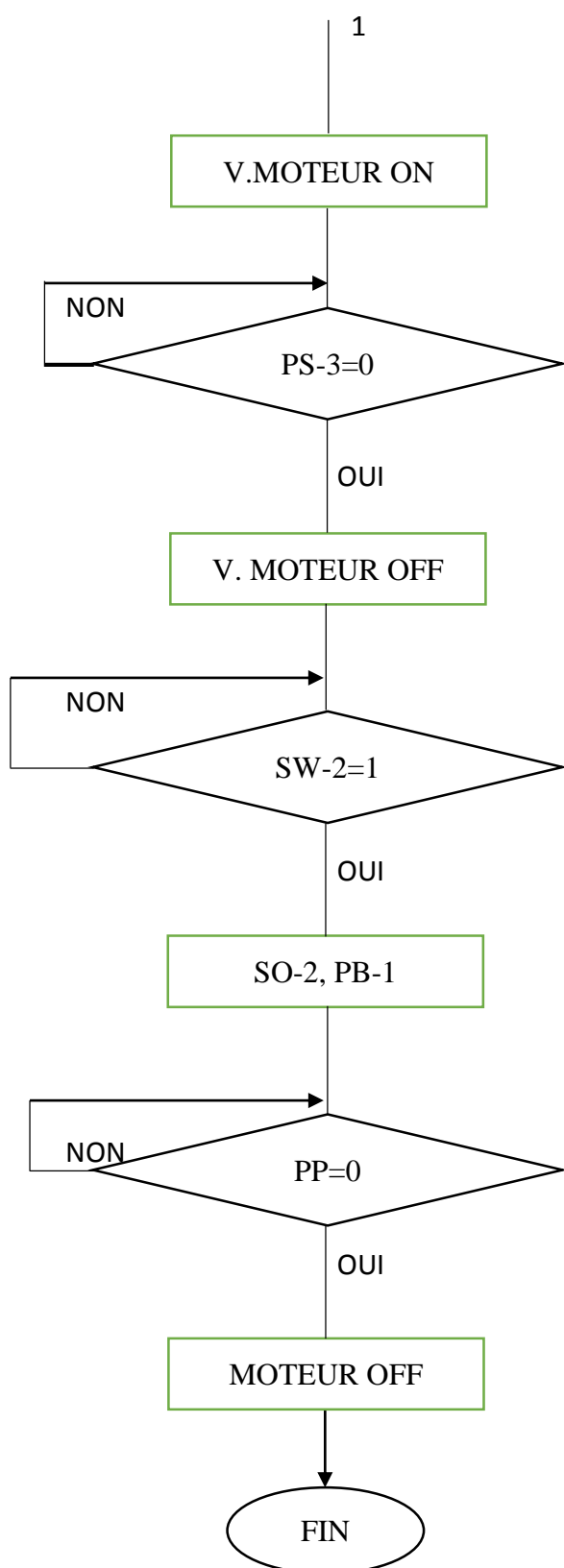


Figure 27 : Bouton de transfert du programme

Le programme est téléchargé vers la carte, le message "télé versement terminé" s'affiche une fois l'opération terminée.

3) Organigramme montrant les étapes de programmation de convoyeur :





4) Présentation de la structure du programme contrôlant le convoyeur

CU-4001 :

*Nous allons commencer par ramener la bibliothèque de l'afficheur lcd avec l'instruction :

```
#include <LiquidCrystal.h>
```

*Puis les broches le concernant avec l'instruction :

```
LiquidCrystal lcd( , , , , );
```

*Puis nous déclarons les différentes variables et constantes.

*puis dans le void setup() :

-déclarations du type de l'afficheur avec l'instruction :

```
lcd.begin(20, 4);
```

-ainsi que la définition des différentes entrées et sorties avec l'instruction :

```
pinMode(.....,INPUT); //pour les entrées
```

```
pinMode(.....,OUTPUT); //pour les sorties
```

*dans la fonction void loop() nous ferons appel au différentes fonctions de notre programme comme suit :

```
Void loop() {
```

```
    ReceptionSerie(); //c'est une fonction que nous avons utilisé pour le test des ports d'entrées et de sorties ,son rôle c'est de lire la chaine de caractère que nous lui insérer.
```

```
    Commandes(); // cette fonction est utilisé pour le test en mode manuel et son rôle consiste a appeler une autre fonction extérieure "compar" puis exécuter des instructions selon le résultats de la comparaison .
```

```
    ACTION(); // elle sert de lien entre la fonction commandes() et la machine .
```

```
    TRAVAIL(); //c'est la fonction principale pour le mode automatique car elle exécute les instructions selon le résultat des capteurs.
```

*en dehors de la fonction void loop() nous traiterons les différentes fonctions qui sont composées de boucle 'if' et 'while' ainsi que :

-digitalread() ; pour la lecture sur les capteurs .

-digitalwrite(.....,HIGH) ;pour allumer/activer.

-digitalwrite(.....,LOW) ;pour éteindre/desactiver.

Conclusion générale

L'objectif initial de ce travail était la réalisation d'une carte de commande du convoyeur simulateur CU-4001 à base d'une carte électronique et d'adopter une

méthodologie permettant d'étudier les différentes étapes de fonctionnement de ces dispositifs.

Nous avons commencé notre travail par la présentation du convoyeur comme un système à automatiser, les différents éléments et composants de ce système ont été présentés du point de vue des parties constituant un automate.

Nous avons ensuite élaboré le circuit imprimé de la carte qui sera utilisée pour la réalisation de celle-ci, puis nous avons fait les différentes liaisons entre la carte et la maquette.

La conception des schémas électroniques a été faite et simulée sur Proteus avant la mise en œuvre. Nous avons validé ce travail par des tests sur la maquette mise à notre disposition par le département d'automatique.

Malgré les difficultés que nous avons rencontrées et qui étaient un obstacle pour l'avancement de notre projet, le résultat final est un succès, car la carte de commande fonctionne correctement selon le cahier des charges proposé.

En perspective ce travail pourra s'élargir en termes d'application, tout en ajoutant quelques options pour la maquette. Nous proposons les améliorations suivantes :

- ✓ Ajouter des capteurs de sécurité en cas de dépassement de la course normale ;
- ✓ Modifier les positions des capteurs PS-1 et PS-2 pour faciliter le comptage et le tri ;
- ✓ Utiliser un capteur de début de course qui détecte directement lorsque on met une pièce sur le convoyeur, à la place du micro-switch SW-1 ;
- ✓ Ajouter des boutons de sélection de vitesse ;

Enfin nous souhaitons que ce travail soit d'une grande utilisation pour les TP ainsi que pour toute autre utilité pédagogique.

BIBLIOGRAPHIE

[1]-WWW.arduino.cc consulté en juin 2017.

[2]-WWW.smd.ee consulté en juin 2017.

[3]-<http://WWW.elektronique.fr> consulté en juin 2017.

[4]-www.alldatasheet.com. Consulté en Mai 2017.

[5]-www.fr.wikipedia.org consulté en juin 2017.

[6]-www.fr.openclassrooms.com consulté en Mai 2017.

[7]-www.developpez.net consulté en juin 2017.

[8]-www.pinterest.com consulté en Mai 2017.

[9]-www.bestinstrument.co.id consulté en Mai 2017.

[10]-[**Arduino pour bien commencer en électronique et en programmation ; Par Eskimon et olyte.**](#)

[11]-[**Livret Arduino en français par Jean-Noël Montagné,**](#)