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

Université Yahia Farés de Médéa
Faculté de Technologie

Filiére : génie mécanique
Spécialité : mécatronique

ETUDE D'UNE MACHINE TRIE DES COUILEURS

Pour l'obtention du diplôme de Master en Génie mécanique

Présenté par :
- BOUCHNEFA ABBES
- BEN SEBSI ALI

Proposé par :
PR.ABU DURA SALAM

PROMOTION 2021/2021

الهدف من مشروعنا هو دراسة و تحقيق دارة الكترونية تعتمد على البطاقة الالكترونية اردوينو للتحكم في محركات المؤازرة عن طريق حساس لألوان TCS3200

لتحقيق مشروعنا قمنا ببرمجة الميكرو كونترولر الذي تحتويه بطاقة الاردوينو.

Le but de notre projet est d'étudier et de réaliser un circuit électronique basé sur la carte électronique ARDUINO pour contrôler les servomoteurs à travers une couleur sensible TCS3200.

Pour mener à bien notre projet, nous avons programmé le microcontrôleur, qui contient la carte ARDUINO UNO.

The goal of our project is to study and realize an electronic circuit based on the Arduino electronic board to control the servomotors through a sensitive color TCS3200.

To carry out our project, we have programmed the microcontroller, which contains the ARDUINO card.

Sommaire

Chapitre I :

Principe de Fonctionnement

- Introduction ……..….03
- II.1. Partie Alimentation ……..05
- II.2.partie traitement ……05
- II.3. La partie commande ……..…06
- II.4. La partie opérative ……..…….08
- Conclusion ……..……...……13

Chapitre I I :

Arduino

- Introduction……...15
- I.1 La carte Arduino UNO ……..……..…15
- I.1 -1 Circuit d'alimentation de la carte Arduino……..16
- I.1 -2 Entrées/sorties numériques…….……17
- I.1 -3 Le Microcontrôleurs AVR…….……..…18
- I.1-3-1Familles des microcontrôleurs AVR……..…18
- I.1-3-2 Le code des microcontrôleurs AVR…….21
- I.1-3-3Les mémoires de microcontrôleur ……...…23
- I.1-3-4 Sources de temps dans le microcontrôleur AVR …..…23
- I.1-3-4 -1 Oscillateur de Cristal externe……..……24
- conclusion ……..25

Chapitre III :

Réalisation

- **Introduction……..……..…26**
- **III.1. Conception et simulation des schémas…….……..26**
- **III.1.1 Partie soft ……...…27**
- **III.1.1.1 Logiciels utilisés……...……28**
- **Le logiciel Fritzing ……..……...…..29**
- **A.1 Introduction au logiciel Fritzing……..…….30**
- **Programme d'arduino uno……...……31**
- **B.1. Application Arduino.....34**
- **B.1.1. Interface du logiciel Arduino……34**
- **B.2. Description du Menu fille……35**
- **B.3. Les boutons essentiels de l'interface du logiciel Arduino...36**
- **B.4.Les fonctions utilisées Dans la programmation de la carte Arduino……..……..37**
- **Programmer Utilisé ……..……..……..….38**
- **III.2. Partie hard ……..……45**
- **III.2.1. Préparation des pièces nécessaire…….……...45**
- **III.2.2. L'assemblage des pièces en mode essais…...…….…..46**
- **III.2.3. Exécuter le programme …….……..46**
- **III.2. 4.Création de la maquette du projet…….……47**
- **III.2.5.Réglage finale de la machine……49**
- **Conclusion ……..…51**

Chapitre IV :

maintenance

- **Introduction……...….53**
- **IV.1.Définition de la maintenance ……..…….53**
- **IV.2.Les méthodes de maintenance……..53**
- **IV.2.1. Maintenance préventive ……..…….…54**
- **IV.2.1.1. Hardware……...…….54**
- **IV.2.1.2Maintenance curative……..……..... 54**
- **IV.3. Organigrammes de maintenance de la machine qui trie les choses selon les couleurs …..……..….. 55**
- **Conclusion……..……...….56**

La liste des figures

Chapitre I :

Principe de Fonctionnement

- **Schéma II-1- Schéma synoptique……..……..04**
- **Fig. -II-1 partie alimentation……...……...…..05**
- **Fig. -II-3 partie traitement……..….. 06**
- **Fig. -II-2 partie commande……...……...……07**
- **Fig. -II-4 La partie du programme responsable du compte du nombre Des boules…….08**
- **Fig.-II-5 circuit d'affichage……..……09**
- **Fig. -II-6 la partie du programme responsable de La commande du premier Servo-moteur…….……09**
- **Fig. -II-7 la parti programme Le deuxième servo- moteur est le transport de la boule vers l'endroit choisie selon sa couleur……...10**
- **Fig. II.8 circuit Servo -moteur ……..…10**
- **Schéma II-2-plan d'action du Project……11**
- **Fig. -II-9 Schéma Globale du circuit électronique……12**

Chapitre II :

ARDUINO

- **Fig. I-1.La carte ARDUINO UNO…….……..……15**
- **Fig. I-2.pins d'alimentation de la carte ARDUINO…….....……16**
- **Fig. I.3. Entrées/sorties numériques D0 à D13…….……...……...17**
- **Fig. I.4 : Microcontrôleur ATMEL ATMega328P……...…..21**
- **Fig. I.5 : Modèles de microcontrôleurs AVR avec une enveloppe externe du type PDIP et TQFP……..……..……..22**
- **Fig I.6 . Oscillateur de cristal externe……..24**

Chapitre III :

Réalisation

- **Fig. -III-1 PAGE D'ACCUEIL DU LOGICIEL…….……27**
- **Fig. -III-2 PLATINE D'ESSAI……28**
- **Fig. -III-3 VUE SCHEMATIQUE……..29**
- **Fig. -III-4 CIRCUIT IMPRIMÉ……..30**
- **Fig. -III-5 L'image du composant…….……..31**
- **Fig. -III-6 Symbole du composant……31**
- **Fig. -III-7 Nombre et position des pistes……...……..…32**
- **Fig. -III-8 Bibliothèque de composants……33**
- **Fig.-III-9 interface du logiciel ARDUINO…….….34**
- **Fig.-III-10 Menu fille……..……..…...35**
- **Fig.-III-11 Les boutons de l'interface du logiciel……...36**
- **Fig.-III-12 composants utilisés…….....45**
- **Fig.-III-13 Test du circuit……....46**
- **Fig.-III-14 Outils utilisés……..…47**
- **Fig.-III-15 Structure finale……...48**
- **Fig.-III-16 les valeurs affichées par moniteur série……...……...…….49**

Chapitre IV :

Maintenance

- **Fig.- IV- 1 Organigramme de la maintenance…..……..53**
- **Fig. -IV- 2 Organigrammes de la maintenance de la machine qui trie les choses selon les couleurs……..…….……55**

La liste des Tableaux

- **Tableau I.1 : Familles du pic AVR…….....20**
- **Tableau I.2 : Le code de l'environnement de travail du processeur……21**
- **Tableau I.3 : Le code du MICROCONTRLEUR…….….22**
- **Tableau- III-1 Les fonctions spécifiques à l'utilisation de la carte**

  ARDUINO.....37

Introduction générale

Depuis plusieurs années, l'électronique ne cesse de gagner du terrain a l'égard des autres technologique et cela en mettant en place, chaque fois, des composants avec des performances et caractéristique incomparables (cout, rapidité de fonctionnement,…).

Ce qui les rend a la portée de tout le monde en pendant des tubes a vide les transistors les circuits intègres les processeurs et enfin les micros contrôleurs on réussi a mettre au point des appareils avec un minimum de composants et celui en adaptant la méthode dite matérielle qui consiste a condenser des dizaines de composants sur une même platine ce qui représentait des inconvénients énormes.

Profitant de ce développement, nous avons décides de Conception et contrôle, d'une machine industrielle pour trier les choses selon leurs couleurs

Pour mener à bien notre travail on a divisé notre travail en plusieurs parties :

- Chapitre I : principe de fonctionnent.
- Chapitre II: Ardouino..
- Chapitre III : Conception et contrôle.
- Chapitre IV : Maintenance.

Chapitre I

Principe de Fonctionnement.



Introduction :

Dans ce chapitre nous allons expliquer le principe de fonctionnement de notre carte de commande pour cela on utilise le schéma synoptique suivant :

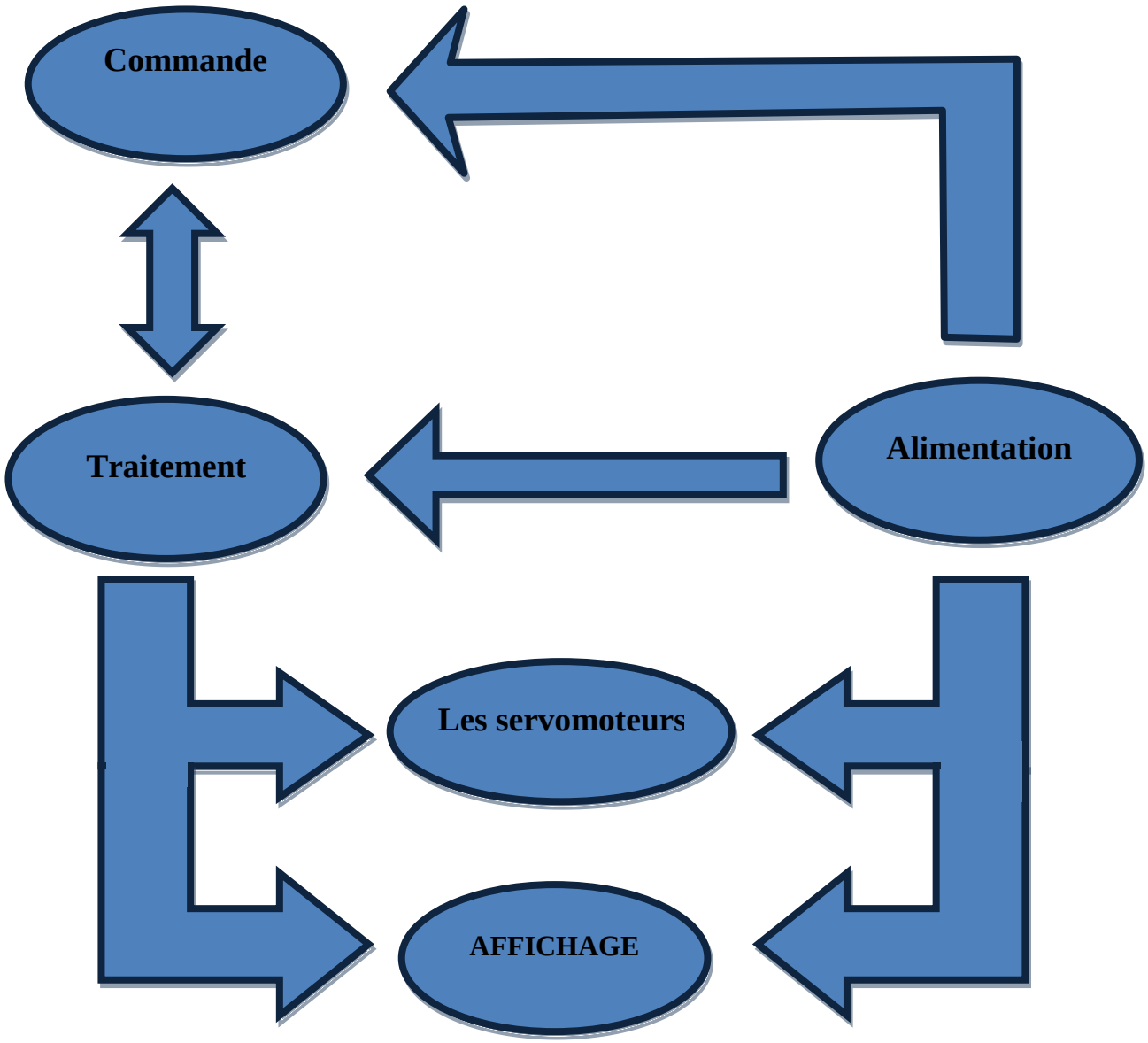


Schéma II-1- Schéma synoptique.

II.1. Partie Alimentation :

L'alimentation de notre système est fournie par un Chargeur **5V/ 1A.**

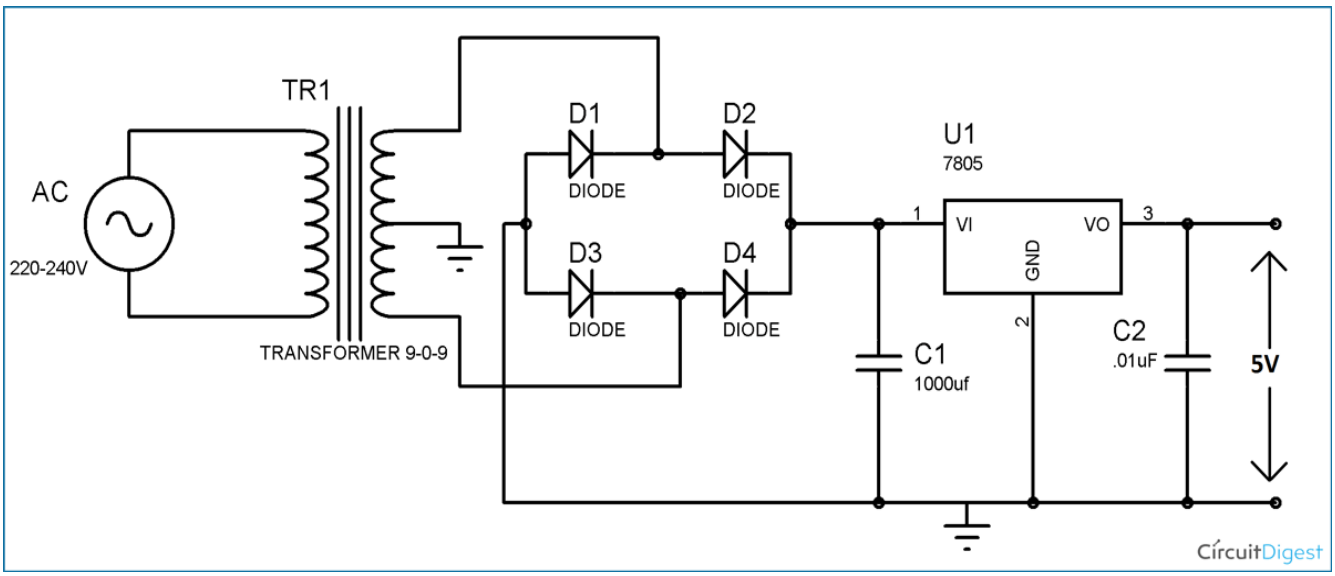


Fig. -II-1 partie alimentation.

II.2.partie traitement :

L'élément essentiel de notre système est la carte **ARDUINO UNO** qu'est une carte microcontrôleur basée sur la série atMēga.

Elle dispose de 14 broches d'entrée / sortie numériques, De 6 entrées analogiques, D'un quartz 16 mhz. , d'une connexion **USB**, D'une prise d'alimentation, D'un connecteur icsp et d'un bouton de réinitialisation.

Elle contient tout le nécessaire pour supporter le microcontrôleur, Connectes –la simplement à un ordinateur avec un câble **USB** ou alimentes-la avec un adaptateur secteur ou un chargeur.

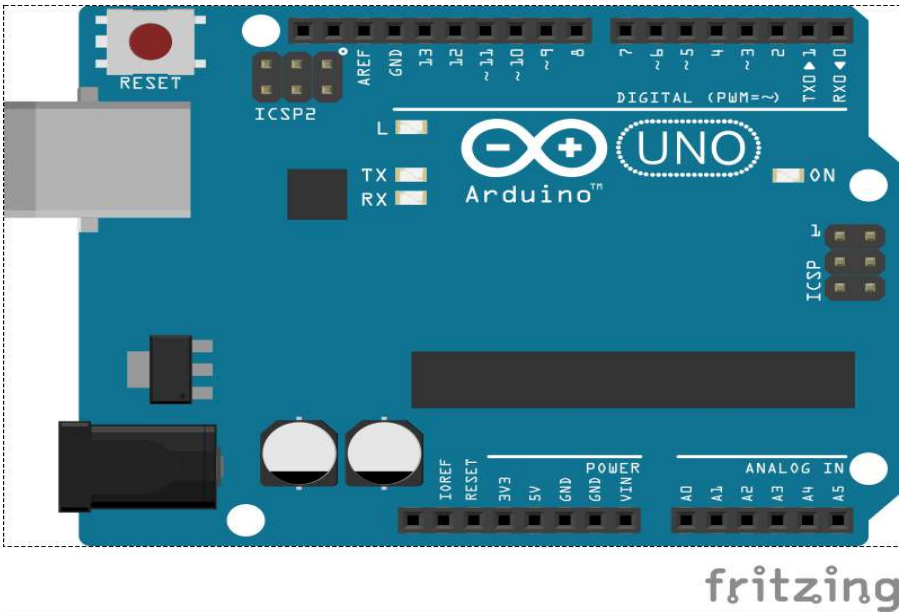


Fig. -II-2 partie traitement

II.3.La partie commande :

La commande de notre machine est effectuée par le capteur de couleur TCS 3200. Le rôle de l'ARDUINO est traiter l'information en voyer via le capture de couleur TCS 3200, ce dernier détecte la lumière couleur à l'aide d'un réseau de photodiodes **8 x 8**.

Ensuite, à l'aide d'un convertisseur courant-fréquence, les lectures des photodiodes sont converties en une onde carrée avec une fréquence directement proportionnelle à l'intensité lumineuse.

Enfin, en utilisant la carte **ARDUINO**, nous pouvons lire la sortie en onde carrée

Et obtenir les résultats pour la couleur.

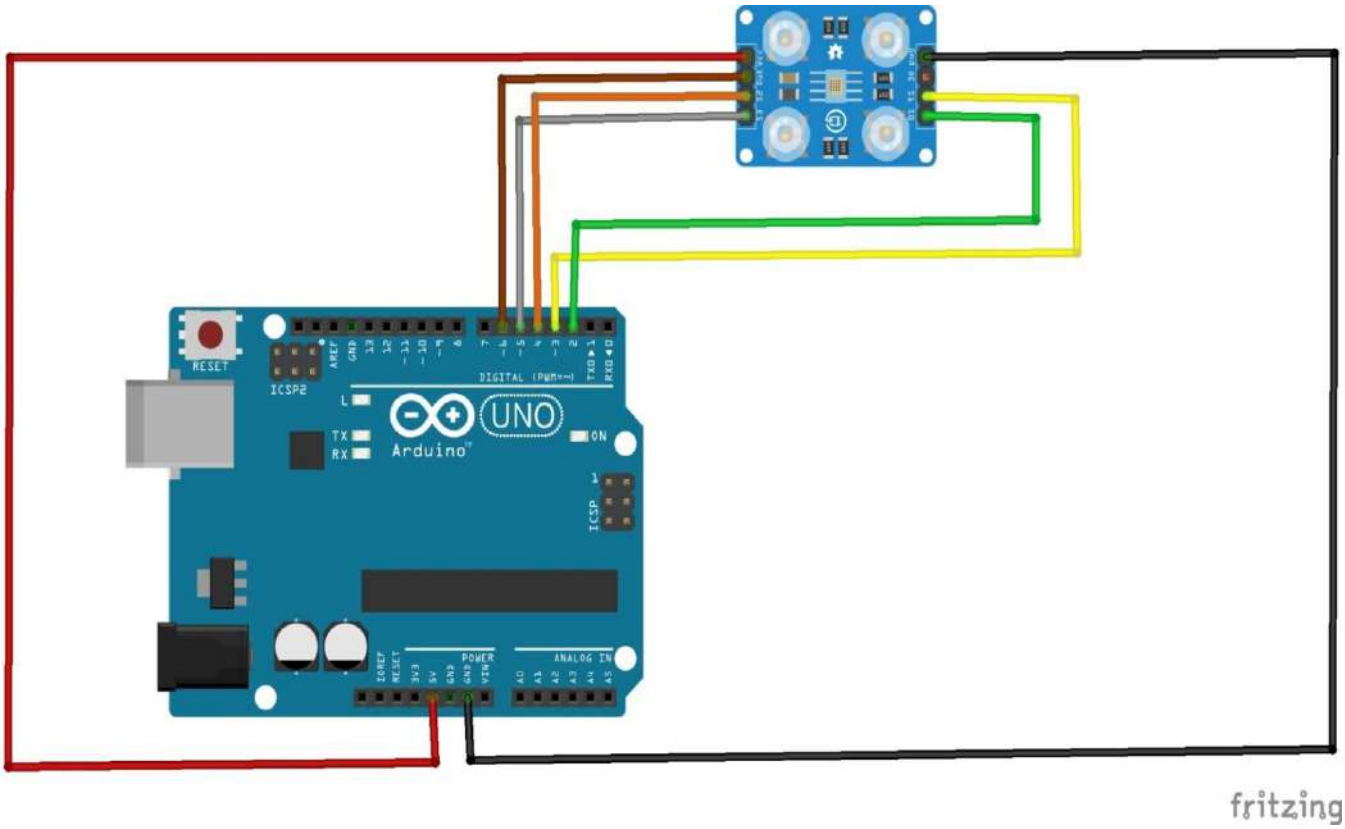


Fig. -I-3 partie commande

II.4. La partie opérative :

Dans cette partie on trouve deux actionneurs, l'affichage LCD pour afficher le nombre de boules triées pour chaque couleur et le nombre total des boules. Le deuxième actionneur est le servo moteur.

La partie du programme qui est responsable de cette opération est :

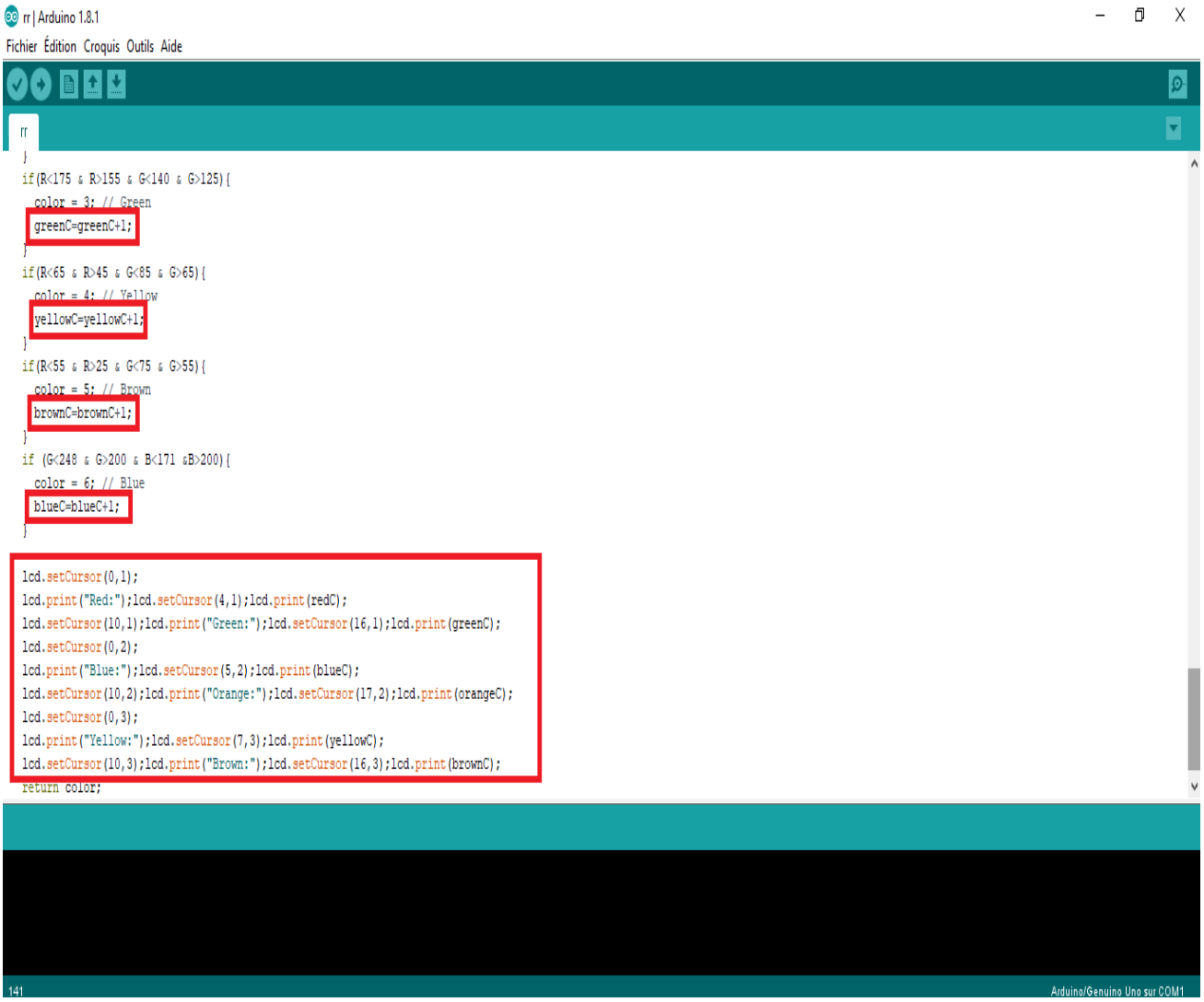


Fig. -I-4 La partie du programme responsable du compte du nombre des boules

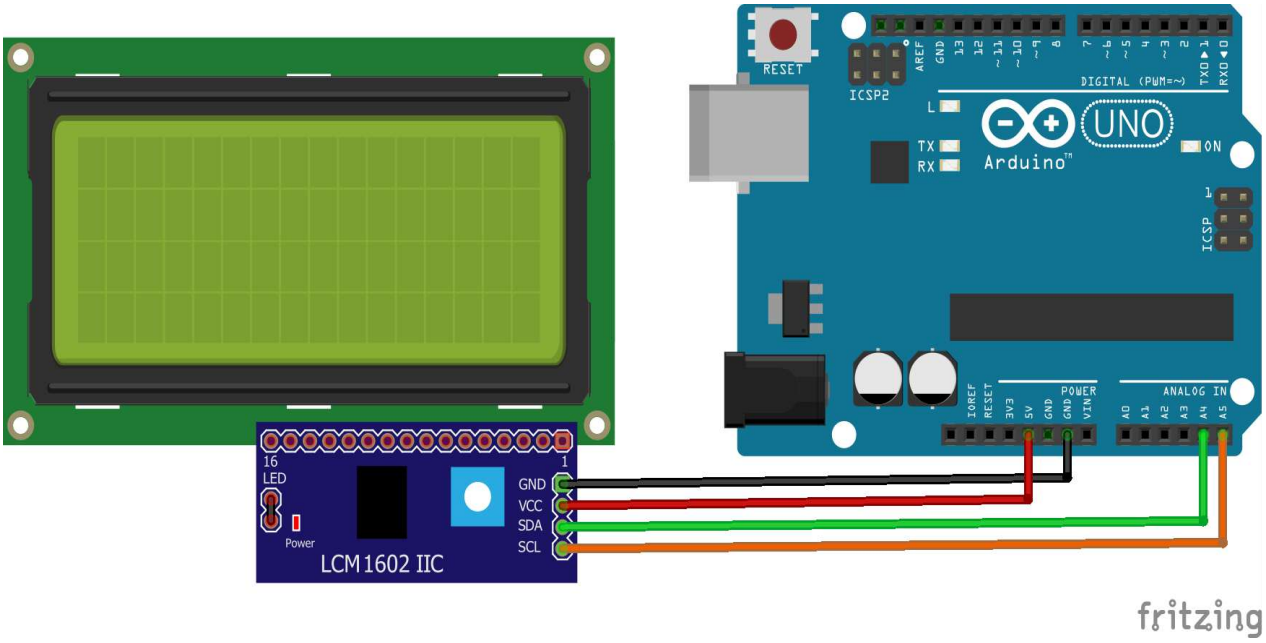


Fig. -II-5 circuit d'affichage

Dans notre projet on a utilisé deux servo moteur commandés directement par l'ARDUINO.

Le premier servo moteur : à le rôle du déplacement des boules vers le capteur TCS3200 pour détecter la couleur.

```
for(int i = 65; i > 29; i--) {
  topServo.write(i);
  delay(2);
}
delay(200);

for(int i = 29; i < 115; i++) {
  topServo.write(i);
  delay(2);
}
```

1

```
}
void loop() {
  topServo.write(115);
  delay(500);

  for(int i = 115; i > 65; i--) {
    topServo.write(i);
    delay(2);
  }
  delay(500);
```

2

Fig. -I-6 la partie du programme responsable de La commande du premier Servo-moteur.

Le rôle du deuxième servo moteur est le transport de la boule vers l'endroit choisie selon sa couleur.

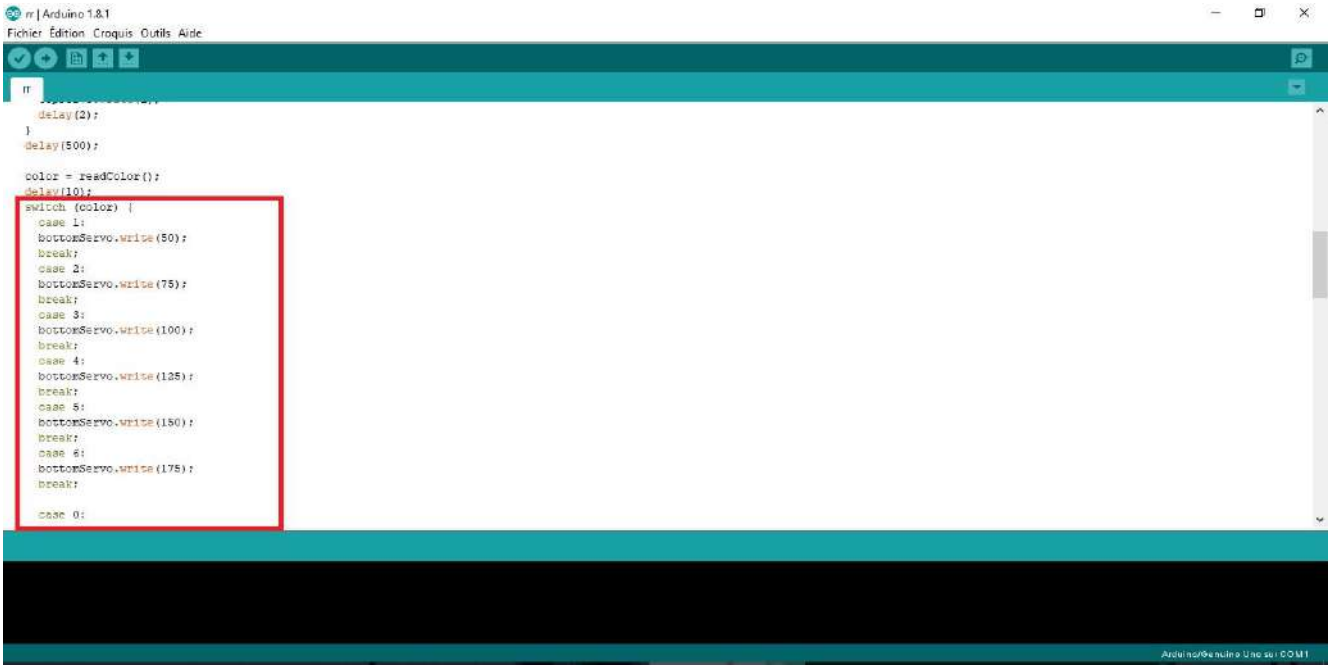


Fig. -II-7 la partie du programme responsable du guidage du deuxième servo - moteur est le transport de la boule vers l'endroit choisie selon sa couleur.

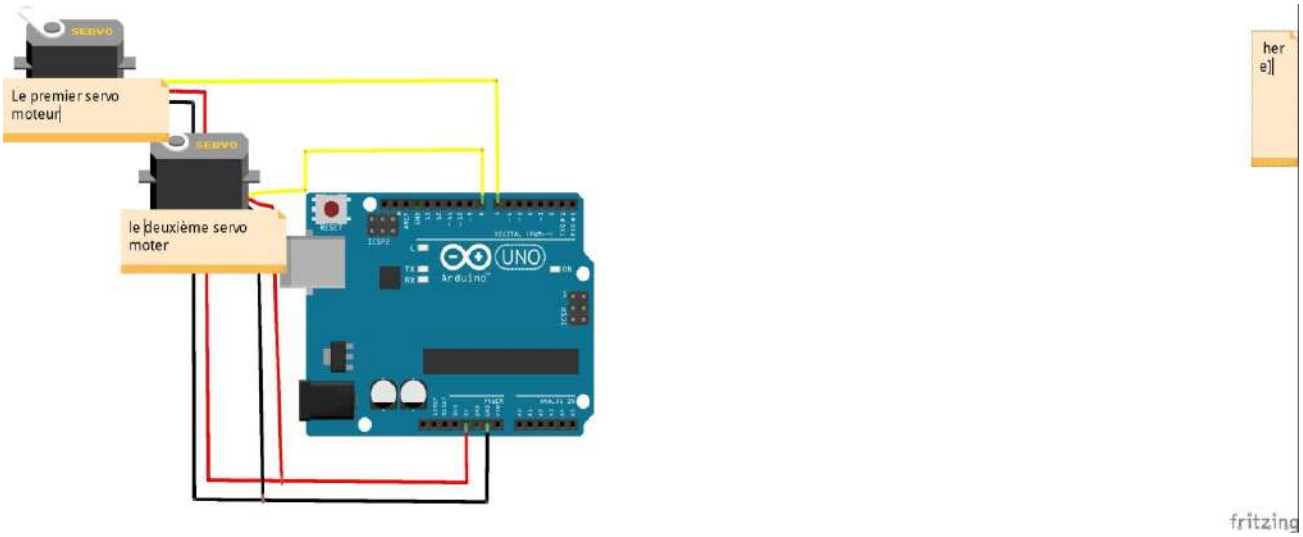


Fig. -I-8 circuit du Servo – moteur connecté à l'ARDUINO

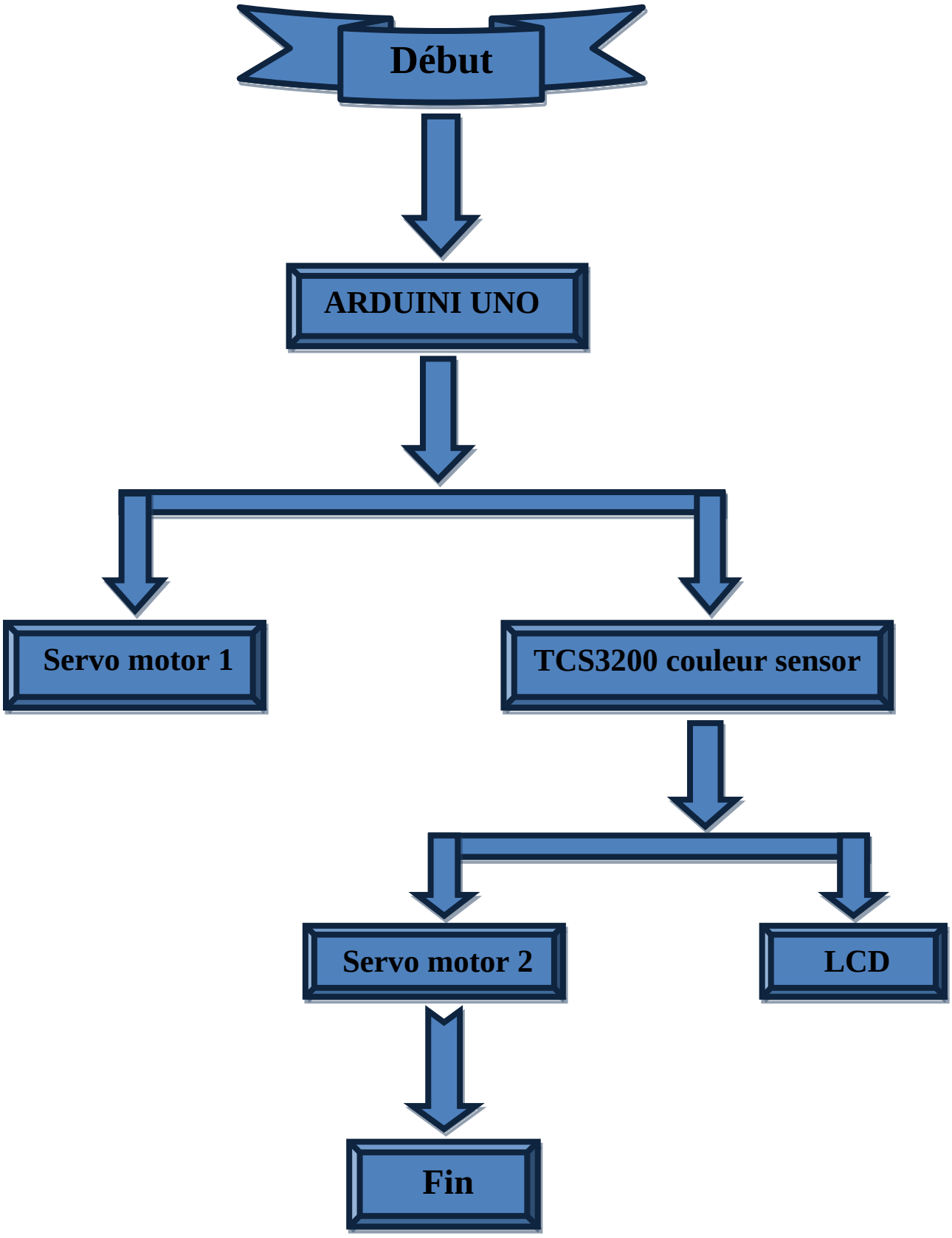


Schéma I-2-plan d'action du Project.

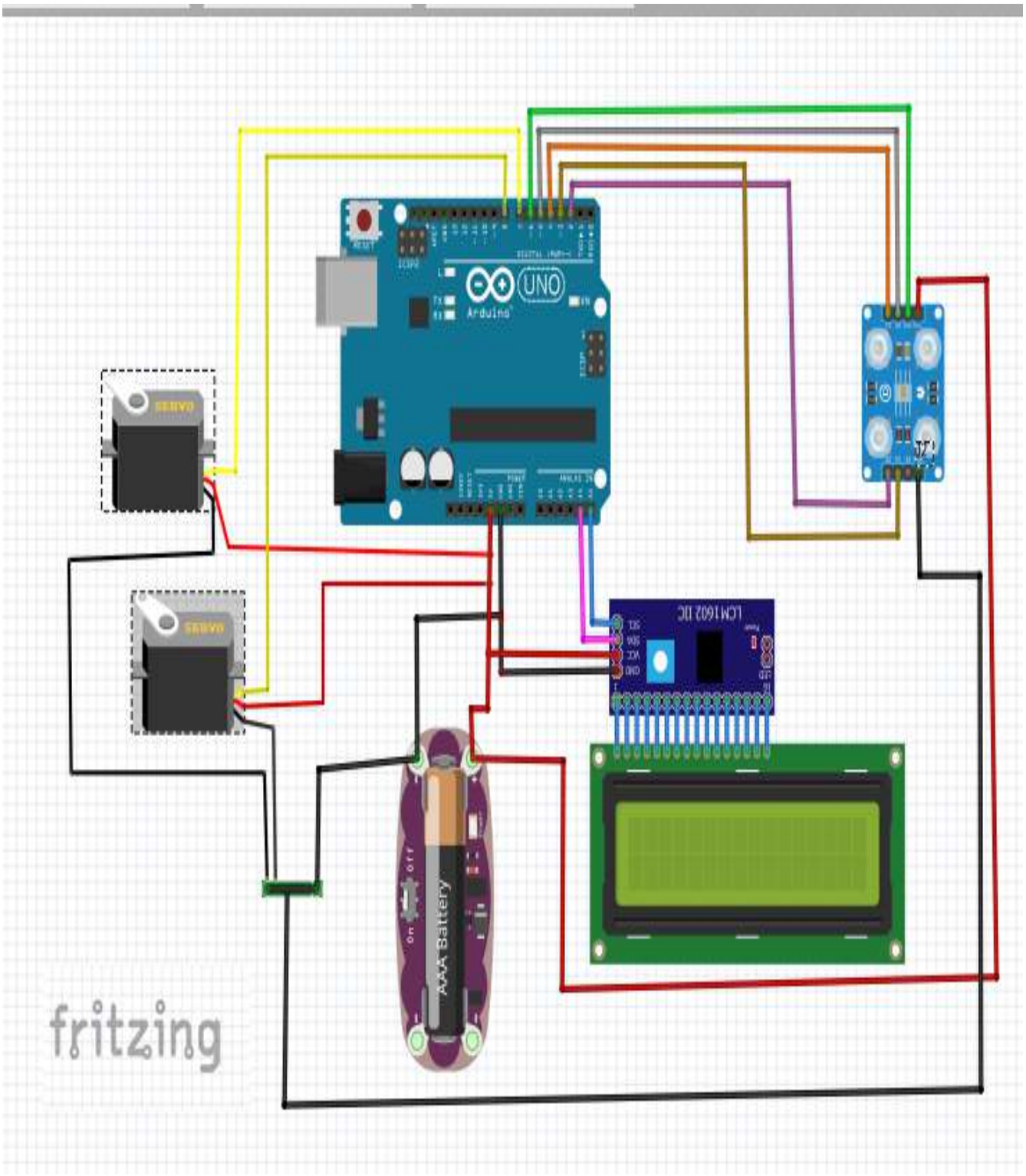


Fig. -I-9 Schéma Globale du circuit électronique.

Conclusion :

Ce dernier chapitre a été conclu par une réalisation pratique en exploitant une méthodologie d'analyse hiérarchique. Cette analyse a été suivie d'une explication détaillée des différentes étapes qu'il faut suivre pour la réalisation.

En fin, le tout a été couronné par la réalisation d'une machine industrielle pour trier les choses selon leur couleur à base d'ARDUINO.

Chapitre II

Ardouino

.



Introduction :

Dans cette partie nous allons expliquer les différents composants de notre projet, notamment la carte Arduino, servomoteur, moteur DC, afficheur LCD, capteur de couleur TCS3200…etc.

I.1 La carte ARDUINO UNO :

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8bits de la famille AVR dont la programmation peut être réalisée en langage C.

Le microcontrôleur ATmega 328, contient des ports d'entrée sortie pour lui permettre d'être connecter aux composants électroniques.

Cette carte comporte aussi 14 connecteurs entrée / sortie numériques, 6 parmi eux peuvent être utilisés comme sorties PWM (PULSE-Width-modulation).Le circuit comporte également un quartz de 16MHz, un port USB pour la connexion à l'ordinateur, une entrée pour l'alimentation.

On peut considérer la carte ARDUINO comme une plateforme de développement et un mini programmateur qui sont destinés à être utilisés directement.

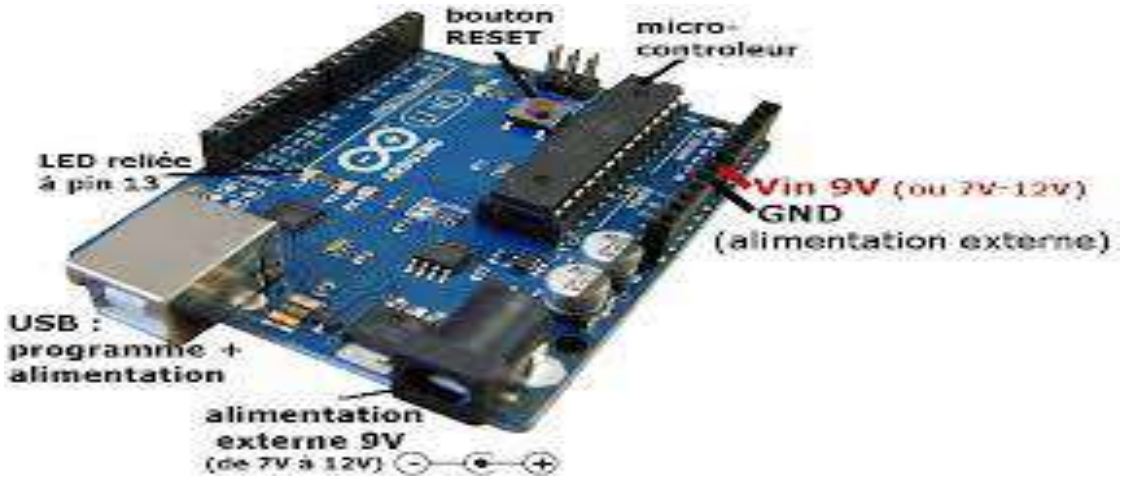


Fig. II.1.La carte ARDUINO UNO.

I.1 -1 Circuit d'alimentation de la carte ARDUINO :

L'alimentation du circuit est assurée soit par le port **USB** ou via une source de tension externe ou même par une batterie.

La tension d'entrée peut être de 7 V à 12 V pour cela on utilise l'entrée Vin, nous pouvons repérer juste sous le connecteur USB un composant qui est le régulateur de tension 5 V. Ce composant se charge de stabiliser et de réguler la tension à 5 V.

En comparaison des autres composants montés en surface (CMS), ce régulateur est assez volumineux, ce qui lui permet de mieux dissiper la chaleur qu'il dégage pour faire baisser la tension en délivrant un ampérage suffisant. Cela permettra au circuit de piloter correctement les composants externes.

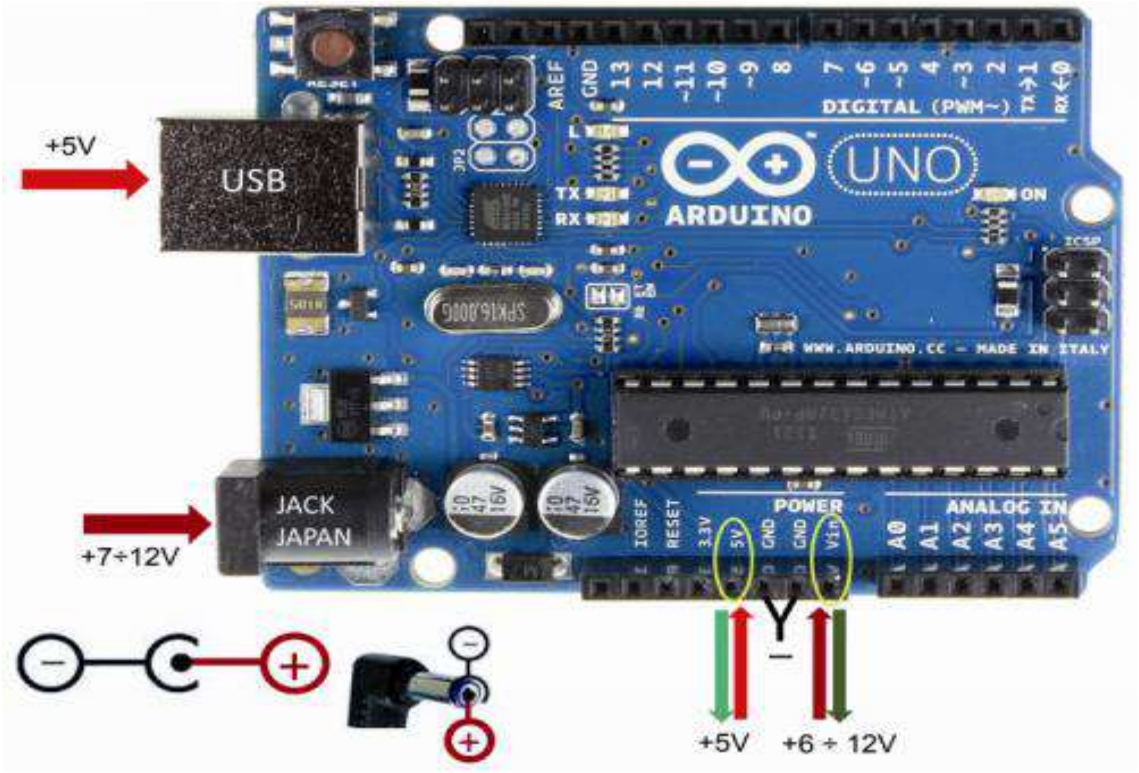


Fig. II.2.pins d'alimentation de la carte Arduino.

I.1 -2 Entrées/sorties numériques :

Passons maintenant aux deux connecteurs du haut de la carte, et commençons par celui de droite. Les légendes désignent des entrées numériques de 0 à 13 en partant de la droite (Digital). Ces broches peuvent servir d'entrées ou de sorties.

Lorsqu'on les utilise en tant que sorties, elles se comportent comme les broches de source de tension présentées plus haut, sauf qu'elles fonctionnent toutes en 5 V et qu'elles peuvent être activées et désactivées depuis un programme. Si on active une de ces broches, la tension présente sera à 5 V. Si on la désactive, la tension sera à 0V. Comme pour les broches de tension, il faut veiller à ne jamais dépasser le courant maximal qu'elles peuvent supporter. Les deux premières broches à droite (0 et 1) portent une légende supplémentaire (RX et TX) pour la transmission et la réception. Ces deux broches sont réservées à la communication ; elles sont, d'une façon indirecte, les connexions en émission et en réception de la connexion USB vers l'ordinateur.

Toutes les broches numériques peuvent supporter jusqu'à 40 mA (milliampères) à 5 V. C'est largement suffisant pour alimenter une LED standard, mais ne surement pas alimenter directement un moteur électrique.

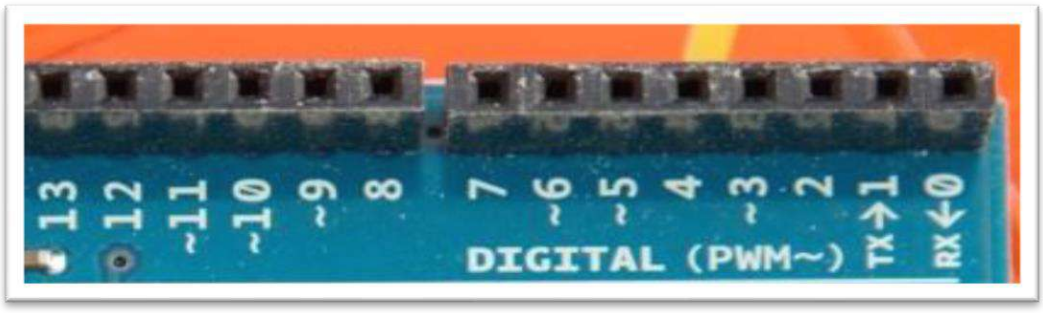


Fig. II.3. Entrées/sorties numériques D0 à D13.

I.1 -3 Le Microcontrôleurs AVR :

Le Microcontrôleurs AVR est l'un des produits de la société américaine ATMEL, Il a été développé dans les laboratoires d'une société en Norvège, à la fin des années nonante. Et l'un des plus microcontrôleurs répandus car il se caractérise par des nombreuses caractéristiques qui le rendent approprié pour de nombreuses applications.

La société américaine ATMEL a apporté une révolution dans le monde des microcontrôleurs avec le produit microcontrôleurs AVR qui a dépassé largement beaucoup les microcontrôleurs 8 bits, il se caractérise par des performances élevées et une faible puissance, Dans les microcontrôleurs AVR la plupart des instructions s'exécute en un seul cycle Ces microcontrôleurs fournis également une mémoire effaçable et un logiciel d'écriture pour plus de 1000000 fois, Et la société ATMEL assure que le programme fonctionne correctement jusqu'à 25 ans, le microcontrôleur AVR possède également des unités périphériques multifonction compact qui prévoit l'utilisation de circuits intégrés externe, aussi il fourni un adaptateur analogiques numérique multi canal intégré .

En plus la possibilité d'une programmation des microcontrôleurs sans être séparée de système.

I.1-3-1Familles des microcontrôleurs AVR :

Le microcontrôleur utilisé sur la carte ARDUINO UNO est un microcontrôleur ATMega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

Est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (mémoire morte pour le programme, mémoire vive pour les données), unités périphériques et interfaces d'entrées-sorties. Les microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique (quelques milliwatts en fonctionnement, quelques nano watts en veille), une vitesse de fonctionnement plus faible (quelques mégahertz à quelques centaines de mégahertz) et un coût réduit par rapport aux microprocesseurs utilisés dans les ordinateurs personnels.

Les familles des microcontrôleurs AVR 8-bit se divise à quatre groupes de base, plus d'autres groupes avec des caractéristiques spéciales.

- ***Famille AT90Sxxxx :***

  Famille classique qui a été les premiers microcontrôleurs AVR lancés en 1997 mais leur fabrication est arrêtée.

- ***Famille ATinyxx :***

  Une petite famille de microcontrôleurs AVR développée et qui est apparue au début des années 2000 avec le nombre de pôles que quelques-uns (6 ~ 32 pin) avec une mémoire de programme de petite taille (0.5 ~ 16KB).

- ***Famille ATmegaxxxx :***

  Une grande famille de microcontrôleurs développé AVR, qui est apparu au début des années 2000 et possède un grand nombre de pôles (28 ~ 100 pin) et une mémoire de programme de petite grande taille (8 ~ 256KB) et une variété de propriétés sur la diapositive.

- ***Famille ATxmegaxxxx :***

  Une famille sophistiqué et la plus récente des microcontrôleurs AVR a l'écart d'acquisition en 2008 et possède une variété de caractéristiques et de la capacité d'un nombre relativement important de traitement et de fréquences à ses devancières (32MHz) ... car il a un certain nombre de grands pôles (44 ~ 100 broches) et la taille de la mémoire est grande (16 ~ 384KB) en plus des nouvelles fonctionnalités ne sont pas disponibles sur la diapositive.

En plus des quatre familles de base il existe d'autres familles avec des fonctions Spéciales.

BatteryM AVR	Lighting AVR	USB AVR	megaAVR	tinyAVR
18 ~ 48 Pin	24 ~ 32 Pin	32 ~ 64 Pin	28 ~ 100 Pin	8 ~ 32 Pin
MAX I/O 4~18	MAX I/O 19~27	MAX I/O 22~48	MAX I/O 23~86	MAX I/O 6~28
4KB~40KB Flash	8KB~16KB Flash	8KB~128KB Flash	4KB~256KB Flash	1KB~8KB Flash
256B~1KB EPROM	512B EPROM	512B~4KB EPROM	512B~4KB EPROM	64B~512B EPROM
512B~2KB SRAM	512B~1KB SRAM	512B~8KB SRAM	512B~16KB SRAM	32B~512B SRAM
Up To 8MIPS	Up To 16MIPS	Up To 16MIPS	Up To 20MIPS	Up To 20MIPS
1.8V – 25V	2.7V – 5.5V	2.7V – 5.5V	1.8V – 5.5V	1.8V – 5.5V



Automotive AVR	CAN AVR	LCD AVR	AVR Z-Link	xmegaAVR
14 ~ 64 Pin	64 Pin	64 ~ 100 Pin	MCU Wireless	44 ~ 100 Pin
MAX I/O 6~54	32KB~128KB Flash	MAX I/O 54~69	chipset for:	MAX I/O 36~78
2KB~128KB Flash	1KB~4KB EPROM	16KB~64KB Flash	IEEE 802.15.4	16KB~384KB Flash
128B~4KB EPROM	1K~4KB SRAM	512B~2KB EPROM	and	1KB~4KB EPROM
128B~4KB SRAM	Up To 16MIPS	1KB~4KB SRAM	ZigBee	2KB~32KB SRAM
Up To 16MIPS	2.7V – 5.5V	Up To 20MIPS	applications.	Up To 32MIPS
2.7V – 5.5V		1.8V – 5.5V		1.8V – 3.6V

Tableau II.1 : Familles du pic AVR.

I.1-3-2 Le code des microcontrôleurs AVR :

Dans ce paragraphe, nous allons expliquer le codage de la société ATMEL processeurs de la famille AVR de manière à ce que nous pouvons apprendre beaucoup de choses sur ce dernier, Prenons par exemple le microcontrôleur ATmega8 décrit dans la Figure suivante, Notez que sur le couvercle on trouve le code du fabricant (Atmel) et le numéro de série (01619i) et l'environnement de travail de processeur qui est ATmega8-16PU.



Fig. II.4 : Microcontrôleur ATMEL ATMega328P.

Le code de l'environnement de travail du processeur désigne le type d'application que peut utiliser ce dernier .Ces applications se divisent en trois types qui sont citées dans le tableau ci-dessous, la différence de la pièce est en termes de la capacité du processeur pour résister à des températures élevées et du bruit :

C (Commercial)	I (Industriel)	M (Martial)
Les applications commerciales	Les applications industrielles	Les applications militaires

.

.

Tableau II.2 : Le code de l'environnement de travail du processeur

Le code du processeur est de la forme suivante :

le code	Importance
AT	Abréviation du nom du fabricant ATMEL
Méga	Une famille qui appartenait au processeur (90S, Méga, minuscule, XMEGA)
8	Ce chiffre reflète la taille de la mémoire de programme KB (256KB, 128, 64, 32, 16, 8)
L	Dans le cas où le code porte un caractère L, cela signifie qu'il est capable de travailler à un effort faible (V2.7 ~ 5.5V) et sans ce caractère, cela signifie que le processeur fonctionne à des efforts (4.5V ~ 5.5V).
8	La fréquence maximale du processeur (8, 16,20, 24,36).
PU	forme de la couverture de la diapositive, peut être soit du type qui est soudé sur la couche inférieure de circuit imprimé du symbole PU (SOIC), soit le type de surface qui est soudée à la couche supérieure de circuit imprimé du symbole AU (TQFP)

Tableau II.3 : Le code du MICROCONTRLEUR.

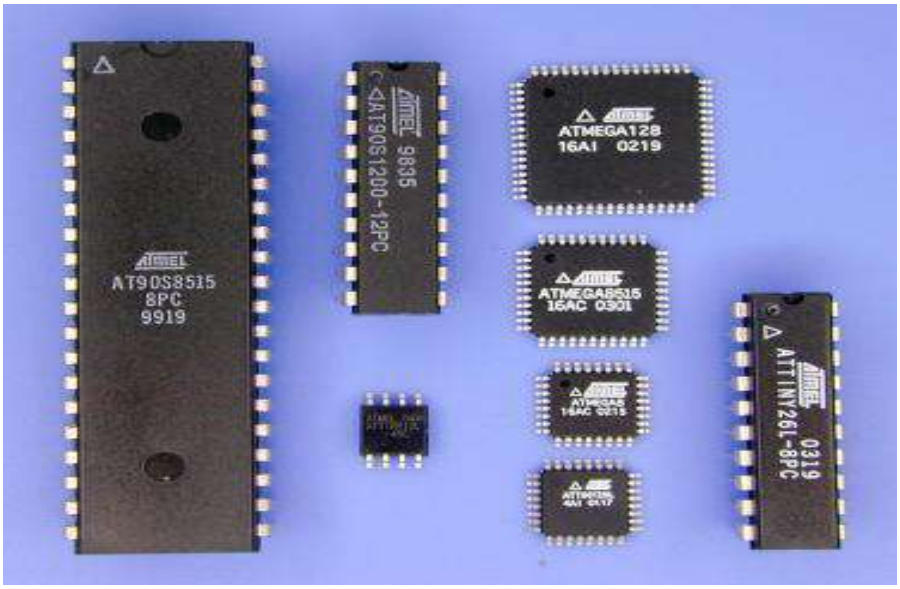


Fig. II.5 : Modèles de microcontrôleurs AVR avec une enveloppe externe Du type PDIP et TQFP.

I.1-3-3Les mémoires de microcontrôleur :

La plupart des microcontrôleurs AVR dispose de trois types de mémoires embarquées sur la diapositive: mémoire de données (SRAM) , mémoire de programme (Flash Memory) mémoire de données (EEPROM).

❖ Mémoire de programme (flash Memory) :

Elle est intégré dans la puce , et de taille 32KB, Dédié au stockage du programme elle est divisé en deux parties: section Boot programme et la section de logiciel d'application afin d'être protégé contre le vol. 10000 cycle de l'Enquête/écriture Au moins, elle est également évolutive.

Pour la reprogrammation, on peut la reprogrammer dans le système sans qu'elle soit transférée.

❖ La mémoire de données (SRAM) :

Mémoire statique d'accès aléatoire (statice Rand om Access Memory), est une mémoire à Accès temporaire lors d'une opération sur les variables.

❖ La mémoire de données (EEPROM) :

Son rôle est pour le stockage des données et de permettre la pérennité, dans cette mémoire on peut effectuer plus de 1000000 cycle d'effacement / écriture.

I.1-3-4 Sources de temps dans le microcontrôleur AVR :

- Le microcontrôleur AVR possède de nombreuses sources de signaux de synchronisation, qui sont divisés principalement en cinq sources:
- Résonateur cristal / céramique externe
- Cristal à basse de fréquence externe
- Oscillateur RC externe

- ❖ Oscillateur RC interne étalonné
- ❖ Horloge externe.

I.1-3-4 -1 Oscillateur de Cristal externe:

Pour la plus grande précision des signaux de synchronisation et si la fréquence de travail est supérieure à 8MHz, Vous pouvez utiliser un cristal externe à bascule, il est relié par les pôles XTAL1 et XTAL2, en tenant compte d'ajouter la valeur de charge externe des condensateurs comme il est indiqué dans la Figure suivante.

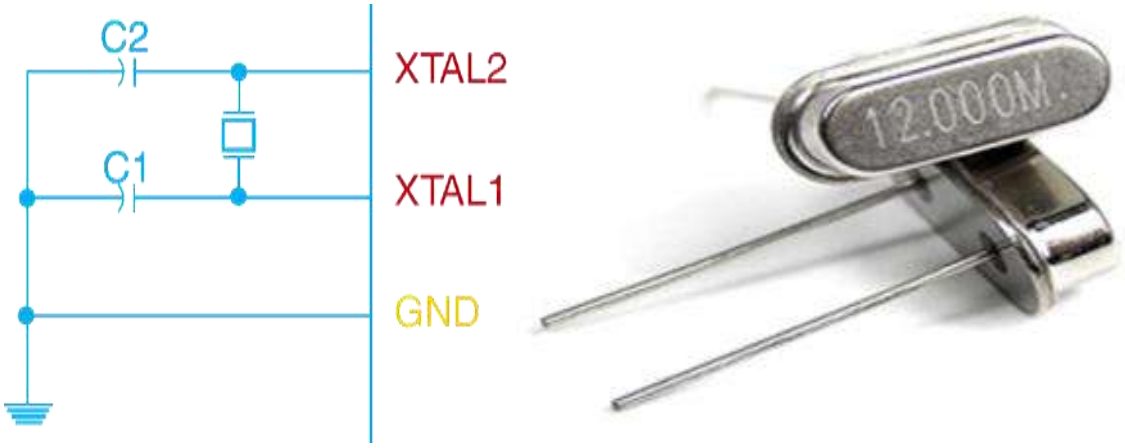


Fig II.6 : Oscillateur de cristal externe.

I.2 Les Moteurs :

Est une machine électrique, Il s'agit d'un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie entre une installation électrique parcourue par un courant et un dispositif mécanique ; selon la source d'énergie.

- En fonctionnement moteur, l'énergie électrique est transformée en énergie mécanique.
- En fonctionnement générateur, l'énergie mécanique est transformée en énergie électrique (elle peut se comporter comme un frein). Dans ce cas elle est aussi appelée dynamo.

Chapitre III

Conception et contrôle.



Introduction :

Après avoir terminé l'étude théorique du montage, nous allons procéder à la Conception et contrôle pratique en commençant par la conception du schéma et la simulation du montage par PC et en termine par le câblage du circuit et le teste du système.

III.1. Conception et simulation des schémas:

Notre Conception et contrôle à passer par deux étapes : étape soft et étape hard

III.1.1 Partie soft :

III.1.1.1 Logiciels utilisés:

A. Le logiciel Fritzing :

A.1. Introduction au logiciel Fritzing:

Se voulant dans la ligne d'ARDUINO et de processing, Fritzing est un projet de logiciel libre, destiné aux non-professionnels de l'électronique. Il a notamment pour vocation de favoriser l'échange de circuits électroniques libres et d'accompagner l'apprentissage de la conception de circuits.

Le logiciel conçu par la faculté de sciences appliquée de l'Université de Potsdam et dont le développement est assuré par la fondation, également nommée Fritzing, est un logiciel d'édition de circuit imprimé. Il est disponible dans seize langues dont le français.

Il est adapté au débutant ou confirmés en électronique pour faire rapidement des circuits simples, et est également un bon outil didactique pour apprendre à bidouiller en électronique par la pratique : il est utilisable en classe à partir de 12 ans et en université.



Fig. -III-1 PAGE D'ACCUEIL DU LOGICIEL FRITZING

Le logiciel comporte trois vues principales :

- La « Platine d'essai », où l'on voit les composants tels qu'ils sont dans la réalité et où l'on construit le montage.

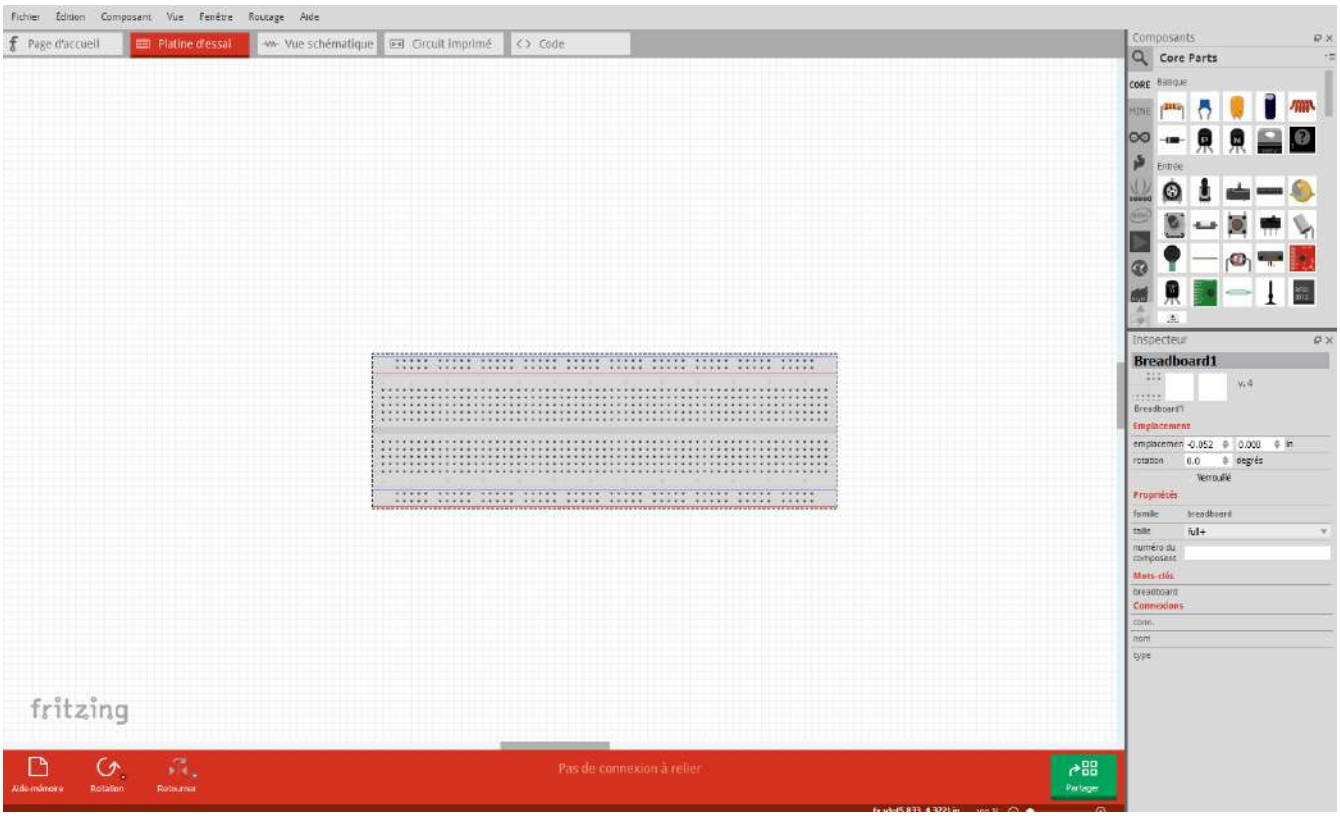


Fig. -III-2 PLATINE D'ESSAI.

- La « Vue schématique », représentant le schéma fonctionnel du circuit.

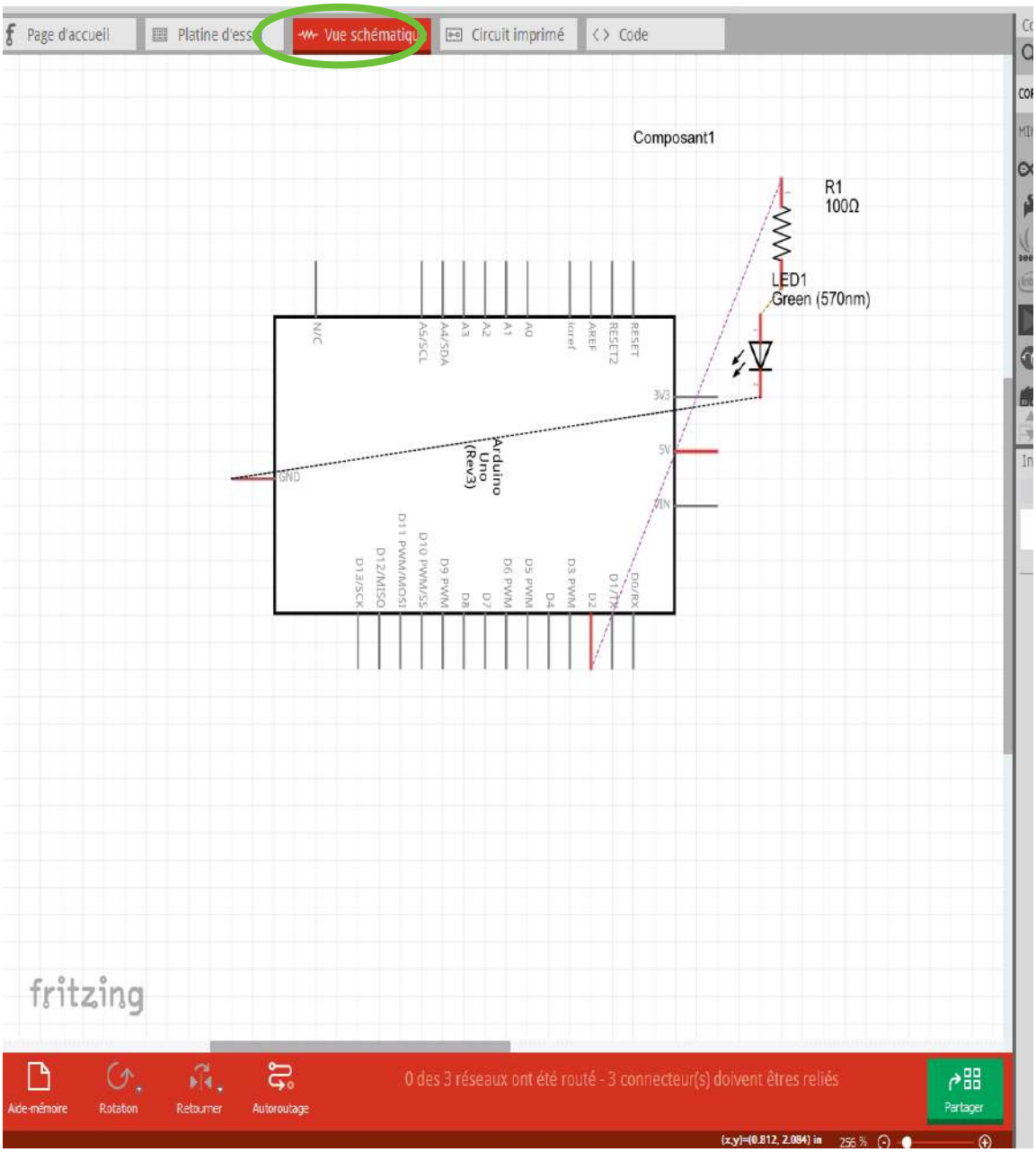


Fig. -III-3 VUE SCHEMATIQUE

- Le « Circuit imprimé », représentant la vue du circuit imprimé tel qu'il sera sorti en PDF pour être imprimé.

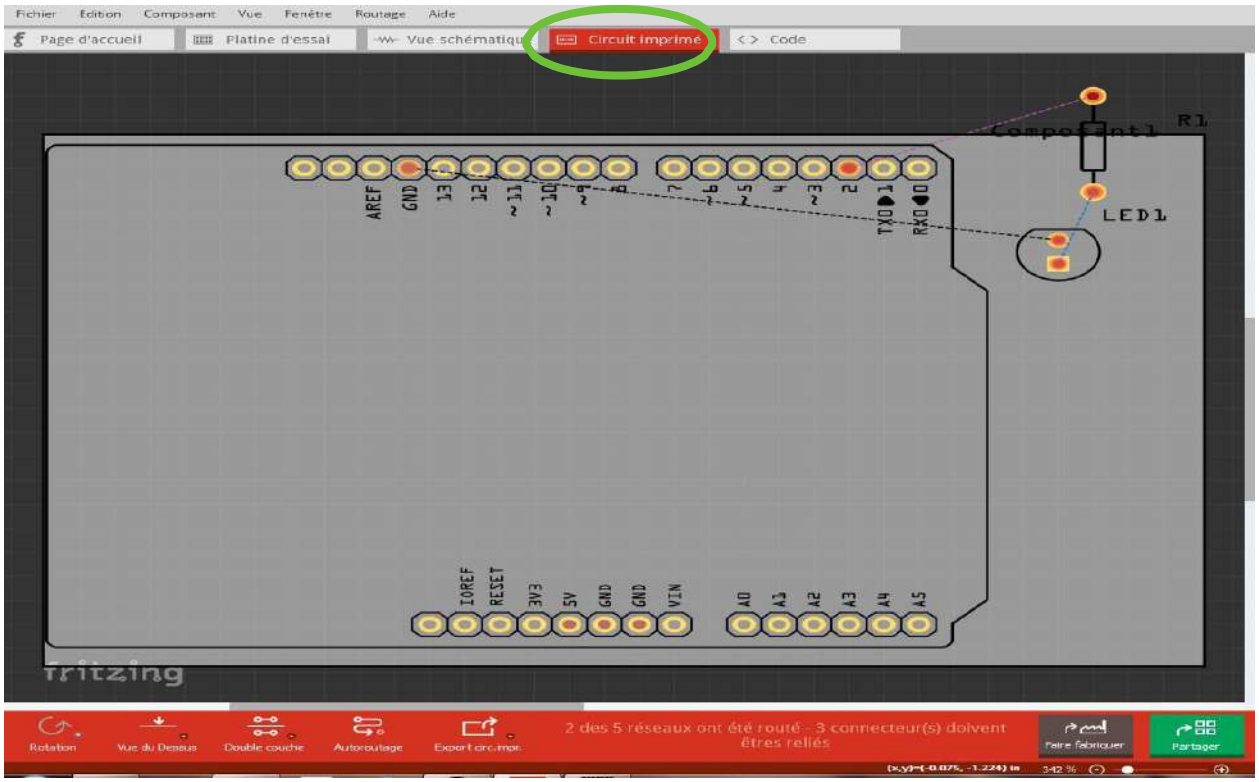


Fig. -III-4 CIRCUIT IMPRIMÉ

La bibliothèque de composants utilise des fichiers au format ouvert « Fritzing Part Format », d'extension .fzp, qu'il est possible d'augmenter. Chaque composant est défini à l'aide de 3 éléments qui doivent pouvoir s'adapter aux trois vues du logiciel:

- l'image du composant, qui peut être réalisée à partir d'une image vectorielle au format SVG (pouvant donc inclure des bitmaps au format PNG ou JPEG).

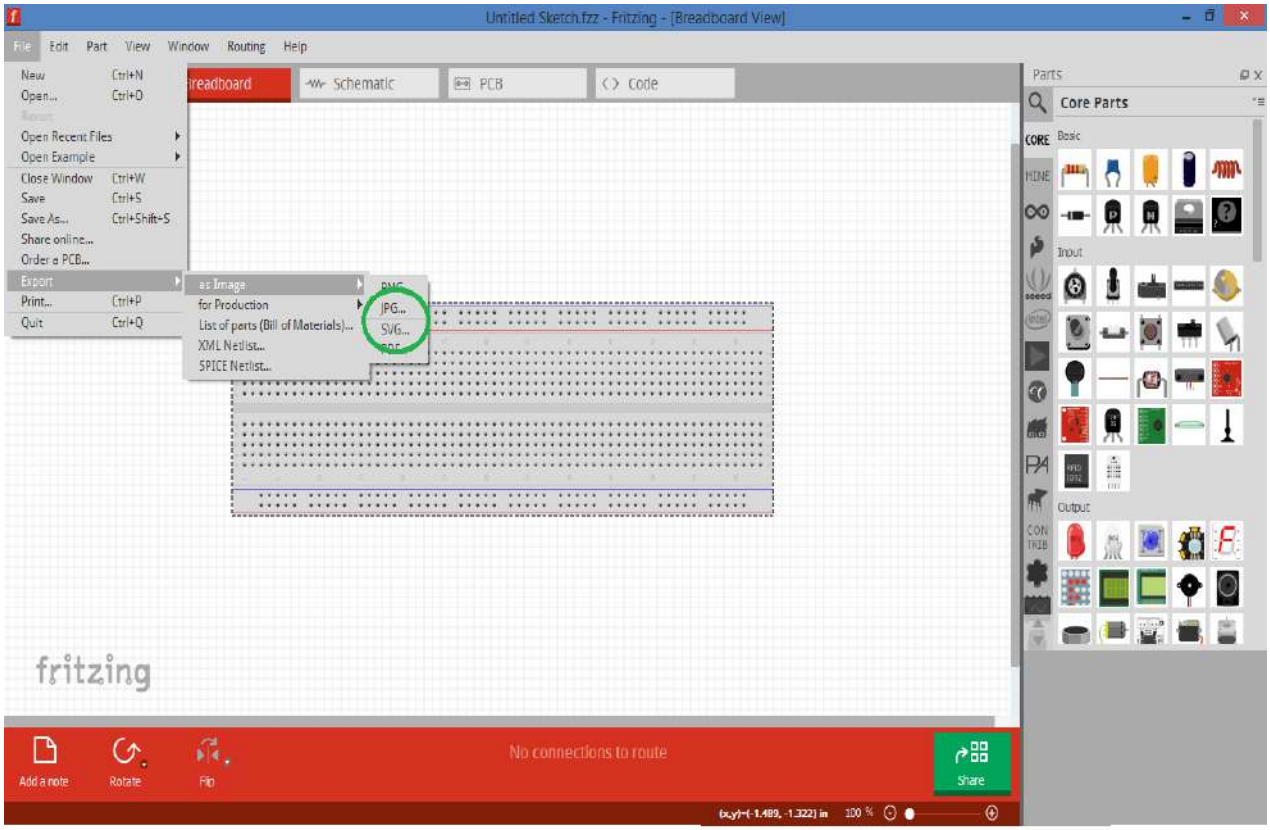


Fig. -III-5 L'image du composant

- Le symbole du composant.

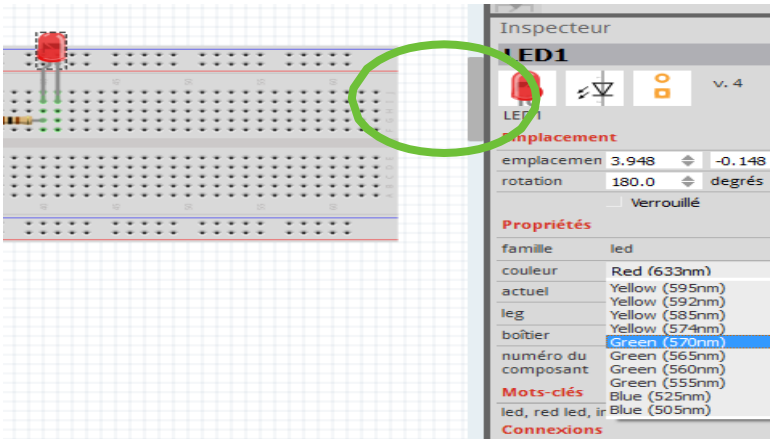


Fig. -III-6 Symbole du composant

- La représentation du composant sur le circuit imprimé (nombre et position des pistes).

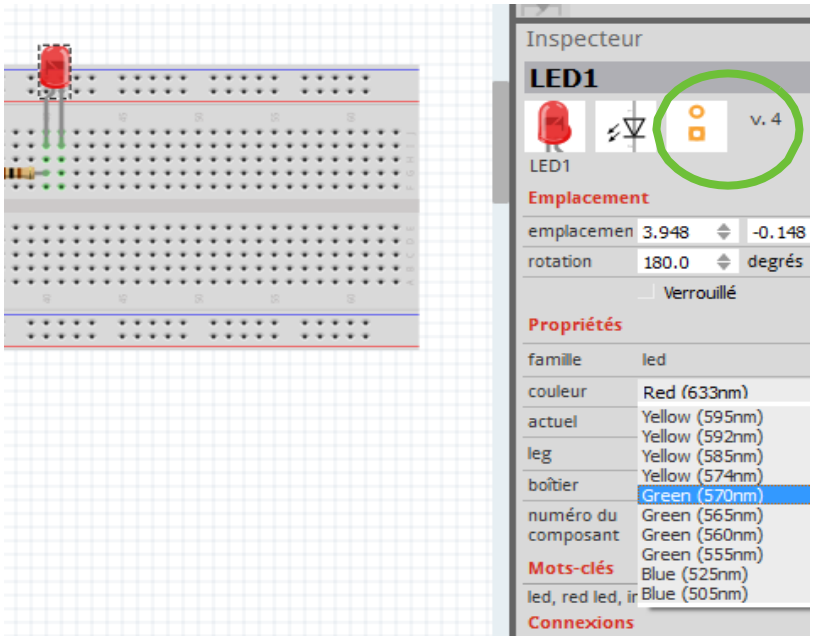


Fig. -III-7 Nombre et position des pistes

Parmi les composants proposés par défaut, on peut citer :

- Les composants électroniques standards (résistance, diodes, transistors, etc.)
- Les circuits intégrés logiques simples les plus répandus.
- Les capteurs les plus courants (commutateur, potentiomètre, accéléromètre, détecteur de lumière, etc.)
- Les composants de sortie les plus courants (LEDs, Super LEDs, quelques écrans LCD répandus, haut-parleurs, servo-moteurs, relais, etc.)
- Différents types d'alimentations.
- Les connecteurs les plus courants (USB, Jack, DB9, MicroSD, etc.)
- La majorité des cartes ARDUINO répandues, dont la série des ARDUINO textiles.
- Différents microcontrôleurs (ARDUINO, Ada fruit, etc.)

- Quelques platines d'essai (grille matérielle servant à la conception et au test de circuits électroniques).



Fig. -III-8 Bibliothèque des composants

Ce logiciel, tout en étant simple d'usage, très visuel et accompagné de didacticiels guidant pas à pas l'utilisateur, s'avère également être un logiciel complet.

- Il permet d'exporter dans les formats d'autres logiciels de conceptions de circuits imprimés répandus ; Eagle et Gerber.
- Il permet d'exporter les typons au format PDF ou SVG afin de pouvoir les imprimer et ainsi insoler une plaque de circuit imprimé.

Une aide minimum est fournie en français et d'autres langues avec le logiciel, mais une grande partie de l'aide est en ligne directement sur le site des auteurs et uniquement en anglais.

B. Application ARDUINO :

L'application ARDUINO est un logiciel qui permet de programmer la carte ARDUINO. Il offre une multitude de fonctionnalités.

B.1. Interface du logiciel ARDUINO :

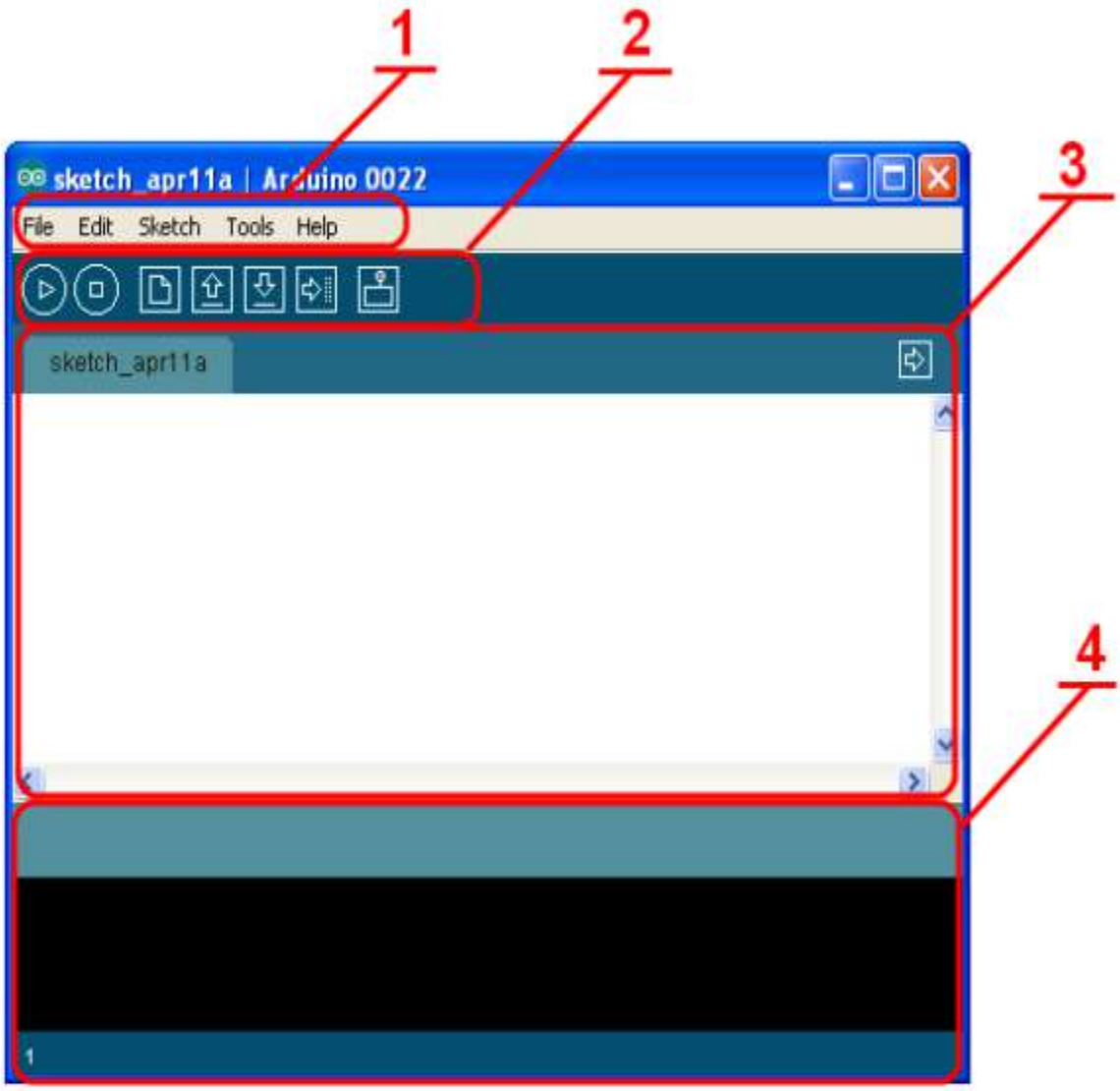


Fig.-III-9 interface du logiciel ARDUINO.

B.2. Description du Menu fille :

1. options de configuration du logiciel

2. boutons pour la programmation des cartes

3. programme à créer

4. débogueur (affichage des erreurs de programmation)

Le menu File dispose d'un certain nombre de choses qui vont être très utiles :

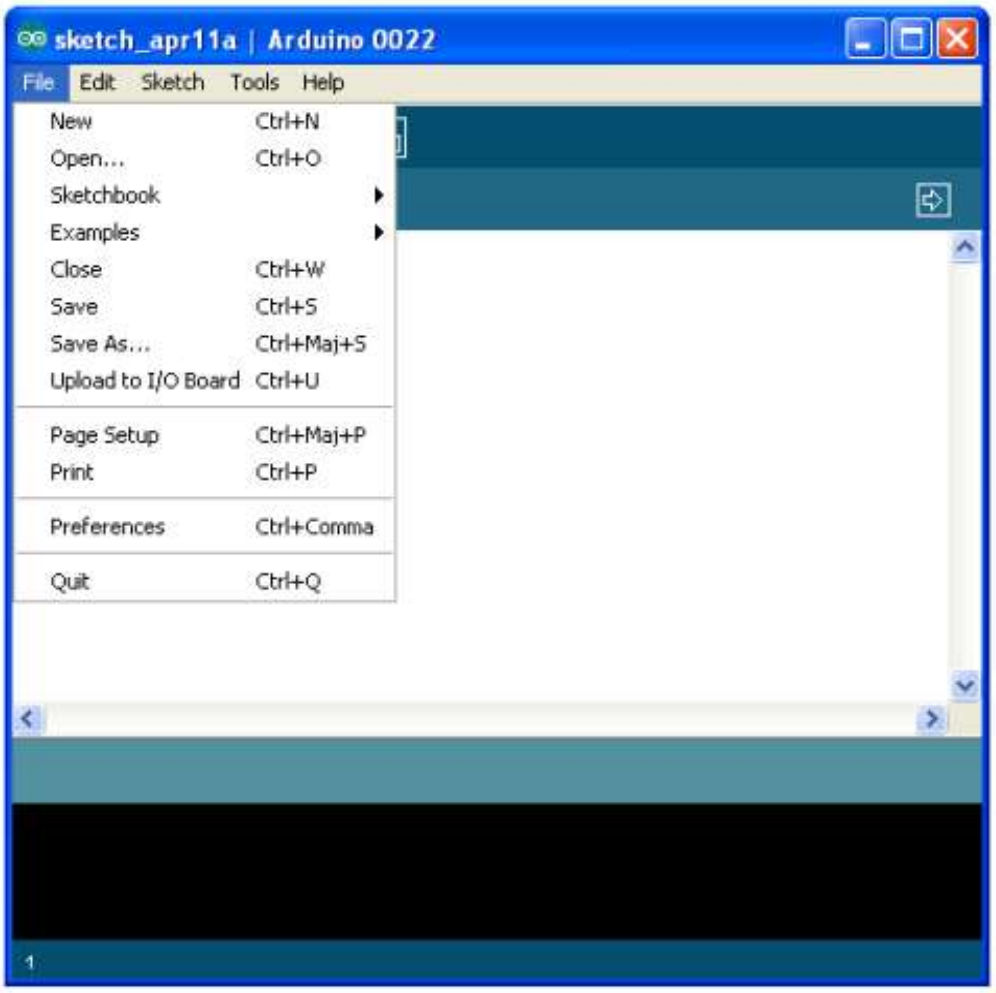


Fig.-III-10 Menu fille

• New (nouveau) : va permettre de créer un nouveau programme. Quand on appuie sur ce bouton, une nouvelle fenêtre, identique à celle-ci, s'affiche à l'écran.

• Open... (Ouvrir) : avec cette commande, on peut ouvrir un programme existant.

• Save / Save as... (Enregistrer / enregistrer sous...) : enregistre le document en cours / Demande où enregistrer le document en cours.

• Exemples (exemples) : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existant.

B.3. Les boutons essentiels de l'interface du logiciel ARDUINO :

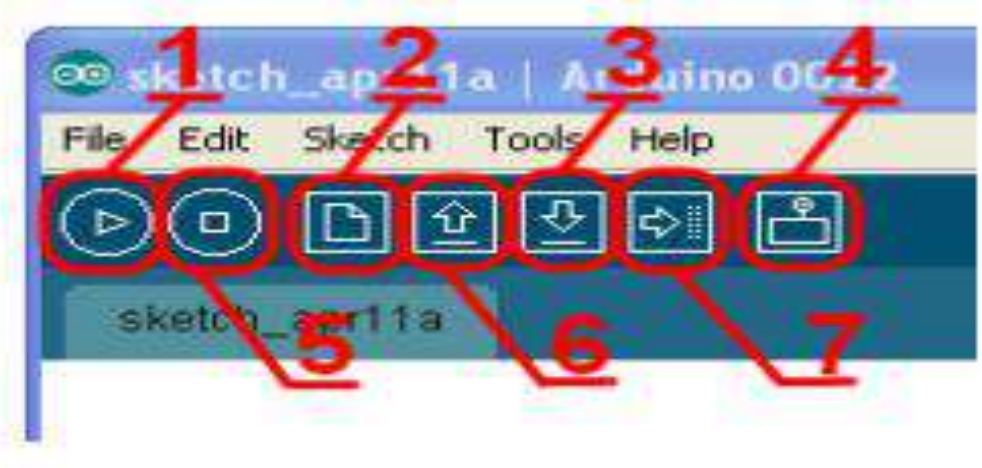


Fig.-III-11 Les boutons de l'interface du logiciel.

1. permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme

2. Créer un nouveau fichier

3. Sauvegarder le programme en cours

4. Liaison série

5. Stoppe la vérification

6. Charger un programme existant

7. Compiler et envoyer le programme vers la carte

B.4.Les fonctions utilisées Dans la programmation de la carte ARDUINO :

Le projet ARDUINO a développé des fonctions spécifiques à l'utilisation de la carte qui ont été listées ci-dessous.

Structure	Constants	Functions
• setup() • loop()	• HIGH, LOW • INPUT, OUTPUT, INPUT_PULLUP • LED_BUILTIN	E/S numérique • pinMode() • digitalWrite() • digitalRead() E/S analogique • analogReference() • analogRead() • analogWrite() - PWM E/S avancée • tone() • noTone() • shiftOut() • shiftIn() • pulseIn() Temps • millis() • micros()
		• delay() • delayMicroseconds() Bits et octets • lowByte() • highByte() • bitRead() • bitWrite() • bitSet() • bitClear() • bit() Interruptions externes • attachInterrupt() • detachInterrupt() Interruptions • interrupts() • noInterrupts() Communication • Serial • Stream

Tableau- III-1 les fonctions spécifiques à l'utilisation de la carte Arduino.

Programme Utilisé:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>


#include <Servo.h>
#define S0 2
#define S1 3
#define S2 4
#define S3 5
#define sensorOut 6
LiquidCrystal_I2C lcd(0x27,20,4);
Servo topServo;
Servo bottomServo;
int redC=0;
int greenC=0;
int blueC=0;
int yellowC=0;
int orangeC=0;
int brownC=0;
int frequency = 0;
int color=0;
void setup() {
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0,0);
```

```
  lcd.print(" ... Bienvenue ...");


  delay(3000);
  lcd.clear();
  lcd.print(" << Color sensor >>");
  pinMode(S0, OUTPUT);
  pinMode(S1, OUTPUT);
  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(sensorOut, INPUT);
  digitalWrite(S0, HIGH);
  digitalWrite(S1, LOW);
  topServo.attach(7);
  bottomServo.attach(8);
  Serial.begin(9600);
}
void loop() {
  topServo.write(115);
  delay(500);

  for(int i = 115; i > 65; i--) {
    topServo.write(i);
    delay(2);
  }
  delay(500);
```

```
color = readColor();
delay(10);
switch (color) {
  case 1:
  bottomServo.write(50);
  break;
  case 2:
  bottomServo.write(75);
  break;
  case 3:
  bottomServo.write(100);
  break;
  case 4:
  bottomServo.write(125);
  break;
  case 5:
  bottomServo.write(150);
  break;
  case 6:
  bottomServo.write(175);
  break;

  case 0:
  break;
```

```
  }
  delay(300);

  for(int i = 65; i > 29; i--) {
    topServo.write(i);
    delay(2);
  }
  delay(200);

  for(int i = 29; i < 115; i++) {
    topServo.write(i);
    delay(2);
  }
  color=0;
}

int readColor() {

  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  frequency = pulseIn(sensorOut, LOW);
  int R = frequency;

  Serial.print("R= ");
  Serial.print(frequency);
```

```
Serial.print("   ");
delay(50);
digitalWrite(S2, HIGH);
digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);
int G = frequency;

Serial.print("G= ");
Serial.print(frequency);
Serial.print("   ");
delay(50);

digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);

frequency = pulseIn(sensorOut, LOW);
int B = frequency;

Serial.print("B= ");
Serial.print(frequency);
Serial.println("   ");
delay(50);
if(R<80 & R>60 & G<250 & G>145){
  color = 1;// Red
```

```
  redC=redC+1;


}
if(G<55 & G>43 & B<47 &B>35){
  color = 2; // Orange
  orangeC=orangeC+1;
}
if(R<175 & R>155 & G<140 & G>125){
  color = 3; // Green
  greenC=greenC+1;
}
if(R<65 & R>45 & G<85 & G>65){
  color = 4; // Yellow
  yellowC=yellowC+1;
}
if(R<55 & R>25 & G<75 & G>55){
  color = 5; // Brown
  brownC=brownC+1;
}
if (G<248 & G>200 & B<171 &B>200){
  color = 6; // Blue
  blueC=blueC+1;
}


lcd.setCursor(0,1);
```

```
  lcd.print("Red:");lcd.setCursor(4,1);lcd.print(redC);
  lcd.setCursor(10,1);lcd.print("Green:");lcd.setCursor(16,1);lcd.print(greenC);
  lcd.setCursor(0,2);
  lcd.print("Blue:");lcd.setCursor(5,2);lcd.print(blueC);
  lcd.setCursor(10,2);lcd.print("Orange:");lcd.setCursor(17,2);lcd.print(orangeC);
  lcd.setCursor(0,3);
  lcd.print("Yellow:");lcd.setCursor(7,3);lcd.print(yellowC);
  lcd.setCursor(10,3);lcd.print("Brown:");lcd.setCursor(16,3);lcd.print(brownC);
  return color;
}
```

III.2. Partie hard :

Dans cette partie Nous allons trouver les étapes du Processus pour créer ce projet

III.2.1. Préparation des pièces nécessaire :

Au début, Nous apportons les pièces électroniques nécessaires.

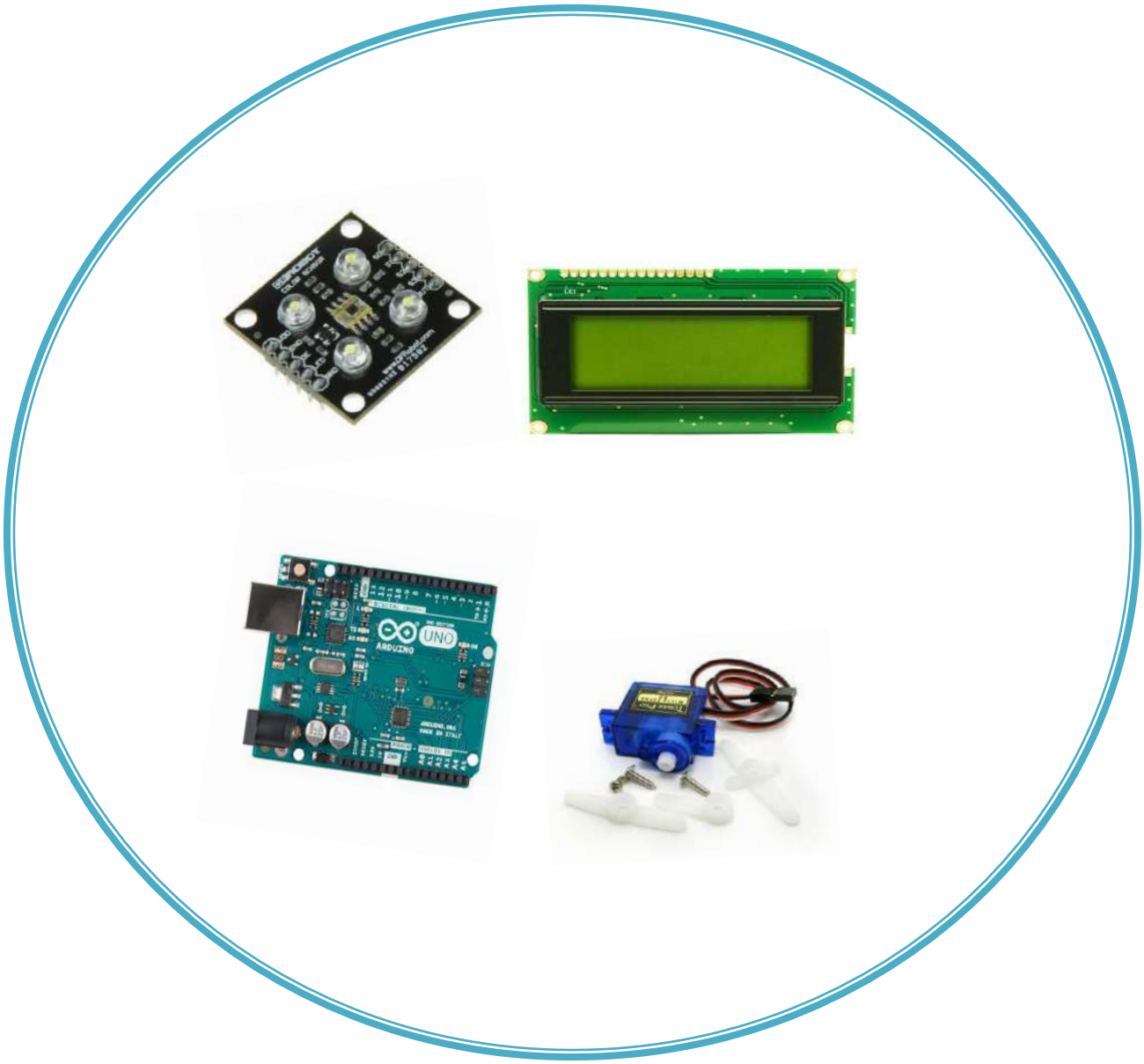


Fig.-III-12 composants utilisés

III.2.2. L'assemblage des pièces en mode essais :

Après l'assemblage des composants électroniques, nous connectons le circuit pour l'expérimentation

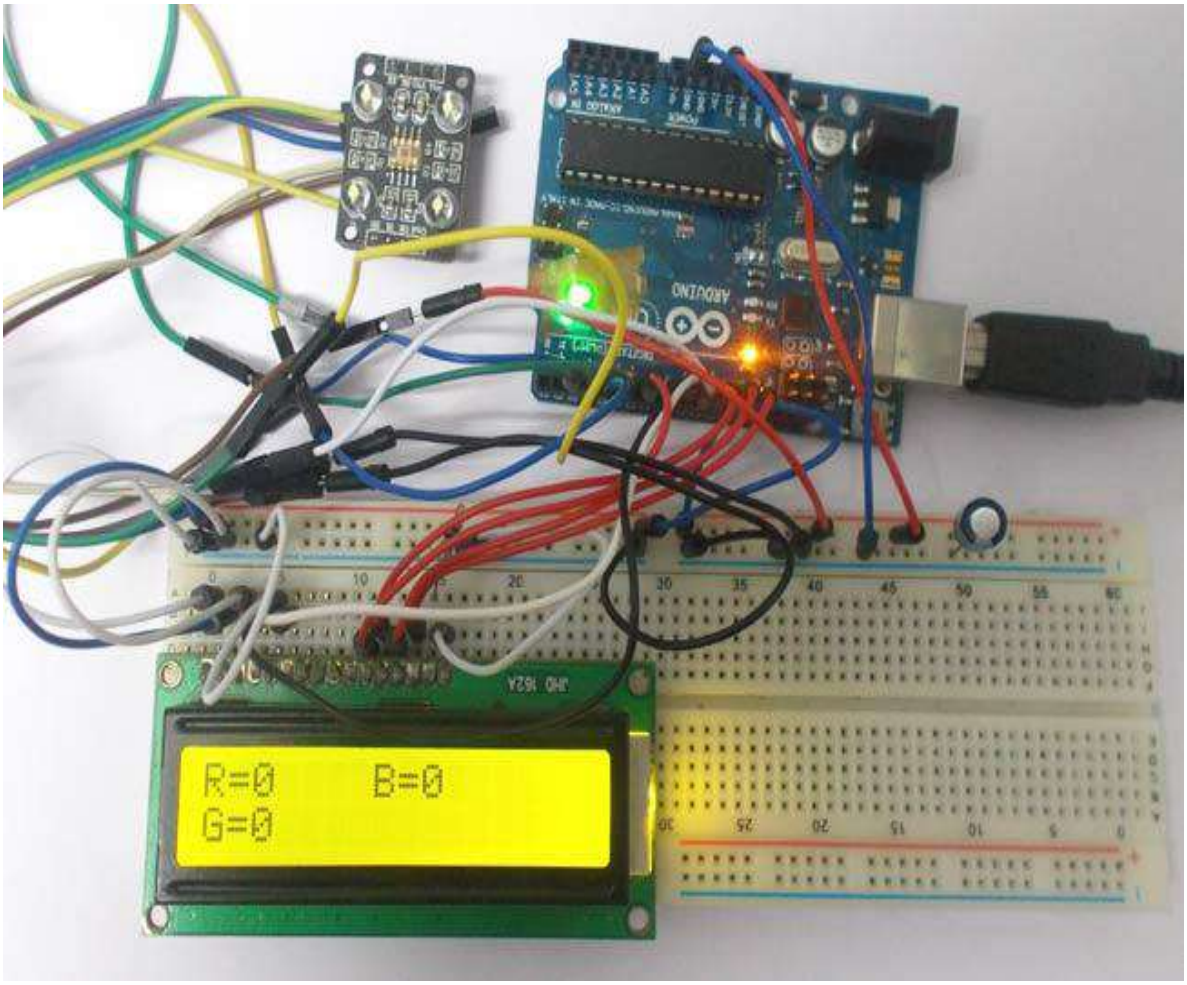


Fig.-III-13 Test du circuit

III.2.3. Exécuter le programme :

Après avoir connecté les différents composants électroniques

Nous connectons ARDUINO à l'ordinateur pour envoyer le programme et ajuster les unités de traitement

III.2. 4.Création de la maquette du projet:

Pour faire cette machine, nous devons apporter les outils appropriées

- planches en bois
- Tube en plastique
- Petite scie
- La colle
- Des vis
- Marteau





Fig.-III-14 Outils utilisés

Les étapes de Conception et contrôle de la maquette :

1-Dessinez les dimensions sur des planches en bois puis découpez-les selon les dessins.

Après avoir coupé les pièces de bois, nous les assemblons et les connectons avec des pièces électroniques pour obtenir la structure finale.

Enfin Nous effectuons le processus d'expérimentation pour assurer le bon fonctionnement de cette machine.



Fig.-III-15 Structure finale.

III.2.5.Réglage finale de la machine :

Avant de tester notre machine elle nécessité un réglage de la couleur pour cela nous passons par les étapes suivantes

- On ouvre l'application arduino
- On téléverse le programme à la carte arduino
- On clic sur l'icone moniteur série
- L'écran affiche les valeurs suivantes



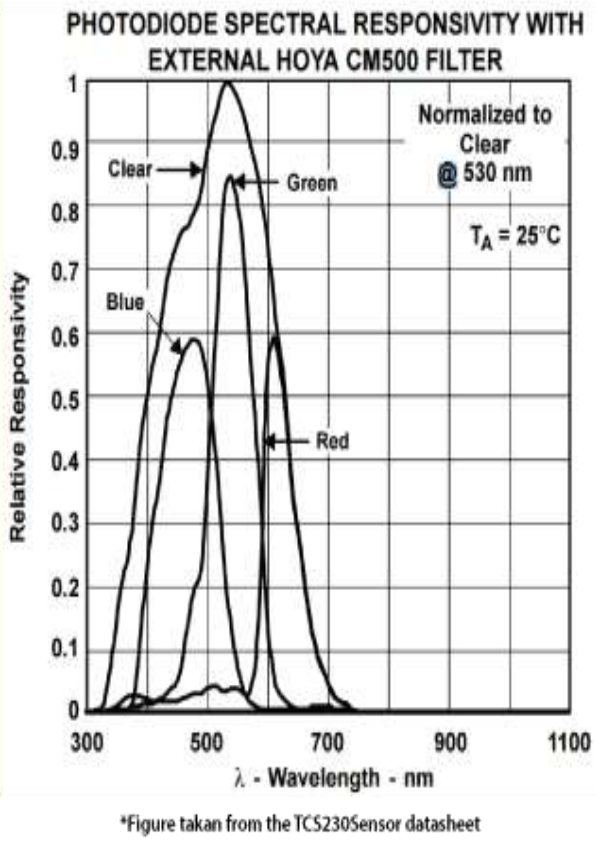


Fig.-III-16 Les valeurs affichées par moniteur série.

- Les valeurs des couleurs RGB varie en fonction de l'emplacement de la boule par rapport au capteur de la couleur

Exemple :

Lorsque j'apporte la boule jaune devant le capteur, je l'obtiens :

R = 22 G = 29 B = 32

Apres le test on n'a conclu que l'intervalle des couleurs RGB suivant :

(R> 17 & R <27 & G> 25 & G <34) aide le capteur a détecter la

Couleur « jaune sans aucune erreur ».

Pour obtenir cet intervalle on a passé par les mêmes étapes précédentes plus le réglage de l'intervalle de la couleur pour plus de précision et pour éviter les erreurs de détection.

De cette façon, nous prenons des lectures dans différentes couleurs que nous devons utiliser et ajoutons ces valeurs dans notre code

if (R <25 & R> 15 & G <33 & G> 23) {color = 1; // Yalow}

if (R <55 & R> 45 & G <49 & G> 39) {color = 2; // green}

if (R <70 & R> 60 & G <10 & G> 20) {color = 3; // blue}

if (R <7 & R> 11 & G <21 & G> 35) {color = 4; // red}

Enfin nous chargeons le coude final dans l'arduino pour le test final

Conclusion :

Ce chapitre nous a permis de connaitre la Conception et contrôle pratique de notre projet et la simulation en utilisant les logiciels Fritzing plus la programmation de la carte ARDUINO, plus le test final en résumé les différentes étapes pour passer d'un schéma électronique à une machine électronique.

Chapitre IV

Maintenance.



Introduction :

Le maintien des équipements de production est un enjeu clé pour la production des usines aussi bien pour la qualité des produits. Tout équipement électronique ou informatique du plus simple au plus complexe nécessite une maintenance pour assurer un fonctionnement optimal.

Dans ce chapitre nous allons écrire la démarche à suivre pour établir une maintenance efficace sur le plan HARD que celui SOFT.

IV.1.Définition de la maintenance :

La maintenance est définie comme étant l'ensemble des opérations permettant de maintenir ou de rétablir un bien dans un état spécifique pour un service déterminé, donc effectuer des opérations qui permettent de conserver le potentiel du matériel pour assurer la continuité et la qualité de production .

IV.2.Les méthodes de maintenance :

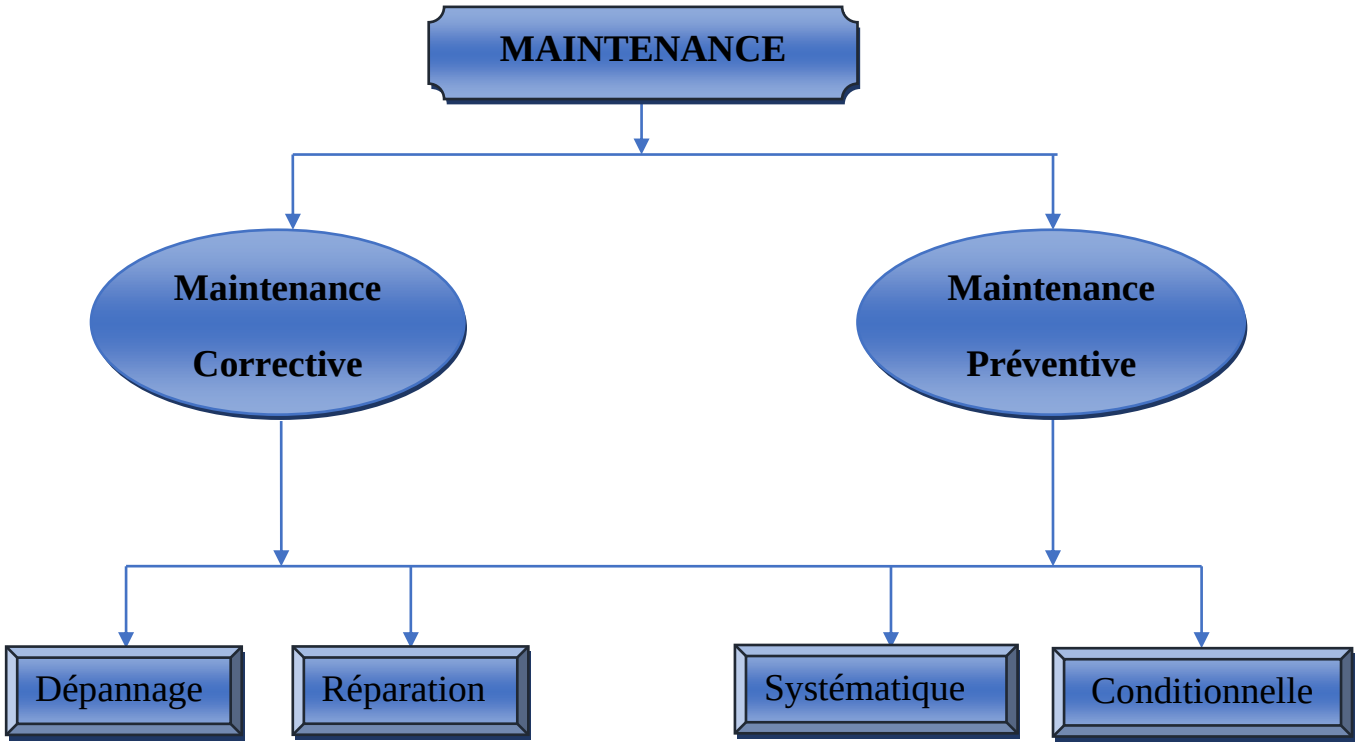


Fig.- IV- 1 Organigramme de la maintenance.

IV.2.1. Maintenance préventive :

Cette maintenance ayant pour objet de réduire la probabilité de défaillance ou de dégradation d'un bien service rendu. Ce type de maintenance est effectuée avant la panne pour :

- Eviter les causes d'accident grave.
- Diminuer la probabilité de défaillance pendant le service.
- Augmenter la durée de vie du matériel.

Il y deux type de maintenance préventive:

IV.2.1.1. Hardware:

Ce type de maintenance est résumé comme suit :

- Pour vouloir faire un essai, le branchement de l'interface avec l'ARDUINO sera hors tension.
- nettoyer les ports de temps en temps.

IV.2.1.2. Maintenance curative :

C'est l'ensemble des activités réalisées après la défaillance d'un bien ou la dégradation de sa fonction, pour lui permettre d'accomplir une fonction requise, au moins provisoirement, la maintenance curative comprend en parti le lancement et son diagnostic, la remise en état sans modification et le contrôle du bon fonctionnement.

IV.3. Organigrammes de maintenance de la machine qui trie les choses selon les couleurs :

L'organigramme suivant nous montre comment qu'on puise suivre la maintenance curative HARD tous en commençant du début à la fin de cet organigramme

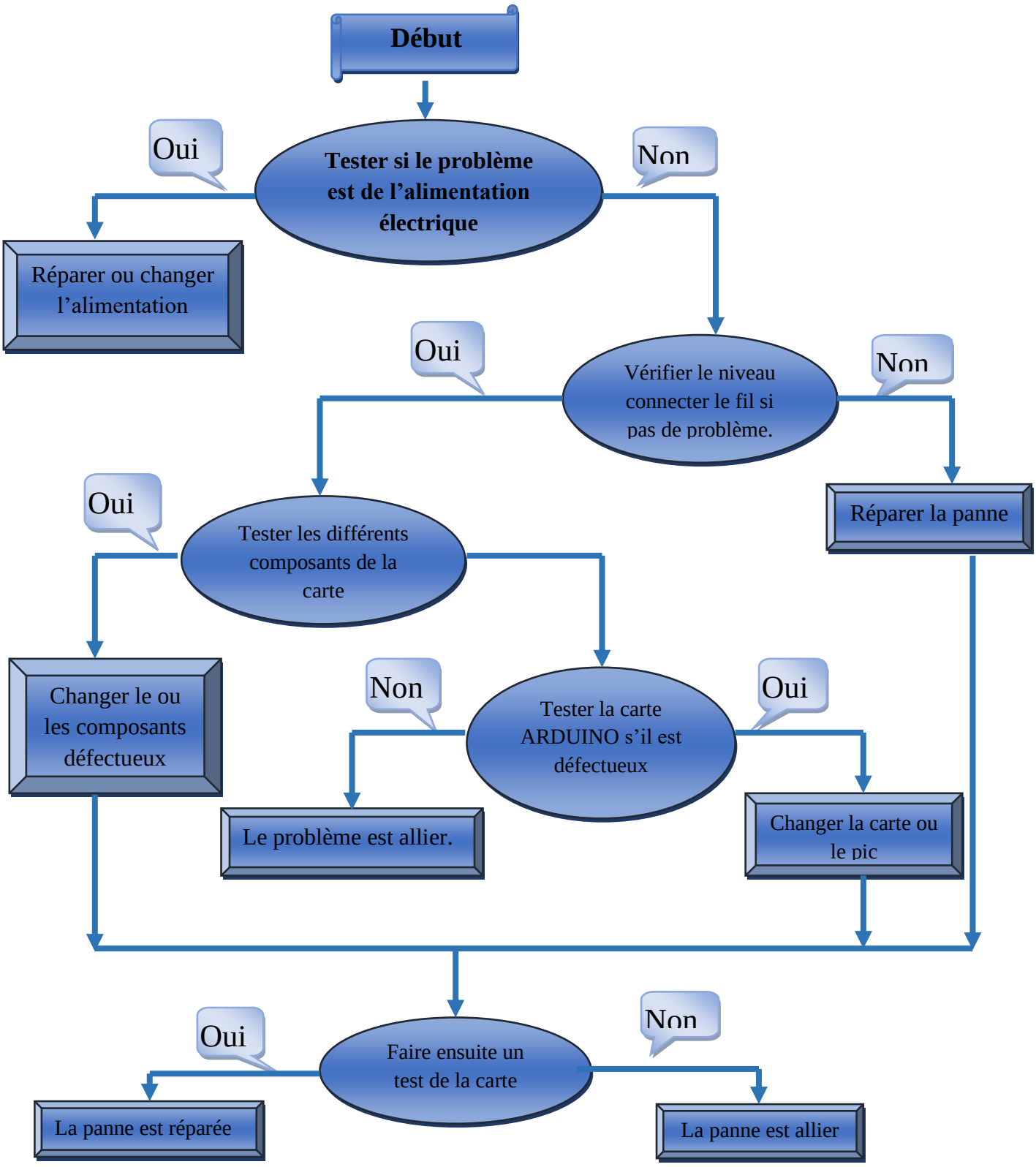


Fig.- IV- 2 Organigrammes de la maintenance de la machine qui trie les choses selon les couleurs

Conclusion :

La maintenance est une opération indispensable, elle doit se faire quotidiennement, dans la mesure où elle assure la sécurité du personnel ainsi que le matériel.

A la fin de ce chapitre on conclu que la maintenance devra être préventive et périodique, pour le bon fonctionnement de la réalisation électronique et pour éviter la défaillance du matériel.

CONCLUSION GÉNÉRALE :

La création de « **l'ARDUINO** » présente un coût qui aura effectué une révolution contre l'électronique traditionnelle pour les amateurs professionnels. Pour cela on a opté la réalisation d'une machine à la base de l'ARDUINO placé dans une carte de commande ruinant d'autres composants électronique.

La Conception et contrôle de cette projet , a fait l'objet d'une étude approfondie est minutieusement élaboré, mais lors de la réalisation nous avons rencontré quelques difficultés a trouvé certains composants tels que l'ARDUINO et la programmation en **langage c** , mais à la fin le but recherché est atteint.

Pour tout clore on espère que ce modeste travail servira d'outil de travail pour la génération à venir.