

République Algérienne Démocratique et Populaire

Ministère de L'enseignement Supérieur et de la Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique

DEPARTEMENT D'ELECTRONIQUE

Mémoire De Fin d'Etudes DE MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : INSTRUMENTATION

Présenté par :

OUCHAOUA NADIR

MEGHNINE DJAFAR

Thème

**Conception et réalisation d'un système de tri
robotisé.**

Mémoire soutenu publiquement le 25/09/2018 devant le jury composé de

PRESIDENT

U.M.M.T.O

EXAMINATEUR

U.M.M.T.O

ENCADREUR

Mr. TAHANOUT

U.M.M.T.O

Promotion: 2017/2018

Remerciements



Nous remercions d'abord le bon Dieu qui nous a donné le courage, la patience, la santé et la volonté d'arriver à la fin de ce travail.

*Nous tenons à remercier notre promoteur Monsieur **M. TAHANOUT** pour son aide, sa disponibilité et les conseils qu'il n'a cessé de nous prodiguer pour l'aboutissement de ce modeste travail, qu'il trouve ici notre profonde gratitude et toute notre reconnaissance et notre respect.*

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner ce travail et de l'enrichir par leurs propositions.

Enfin à toutes et tous ceux qui ont contribué au déroulement de ce travail. Que tous les enseignants ayant contribué et participé à notre formation trouvent ici notre profond respect.

Dédicaces



Ce travail, et bien au-delà, je le dois à mes très chers parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots. Que dieu vous protège et vous garde pour nous.

A mes sœurs

A mes frères

A mes beaux-frères

A mes nièces

Enfin a tout(e)s mes ami(e)s

Nadir

Dédicaces



Ce travail, et bien au-delà, je le dois à mes très chers parents qui m'ont fourni au quotidien un soutien et une confiance sans faille et de ce fait, je ne saurais exprimer ma gratitude seulement par des mots. Que dieu vous protège et vous garde pour nous.

A mes sœurs

A mes frères

A mes beaux-frères

A mes nièces

Enfin a tout(e)s mes ami(e)s

SOMMAIRE

Introduction	1
--------------------	---

Chapitre 1: Robotisation des chaines de productions

1.1 Préambule	2
1.2 Définition d'une chaîne de production	2
1.3 Historique de l'évolution de la chaîne de production	3
1.4 Robotisation des chaines de productions :	4
1.4.1 Définition de la robotique.....	4
1.4.2 la robotisation de certaine tache dans une chaîne de production.....	5
1.4.2.1 Soudage à l'arc	5
1.4.2.2 Manutention et service de machine.....	6
1.4.2.3 Packaging et conditionnement	7
1.4.2.4 Peinture (revêtement)	8
1.4.2.5 Le tri sélectif	9
1.5 Présentation de travail à affecter (la problématique)	10
1.6 Discussion	11

Chapitre 2 : Conception matériel du système de tri

2.1 Préambule	12
2.2 Définition de tri sélectif	12
2.3 Conception du système de tri	12
2.3.1 Capteur de couleur TCS230.....	13
2.3.1.1 Description	13
A) photodiode	14
B) fonctionnement de la photodiode	15
2.3.1.2 Comment le capteur de couleur identifie les couleurs	17
2.3.1.3 Caractéristique de capteur TCS230.....	19
2.3.2 La Carte Arduino	21

2.3.2.1 Historique	21
2.3.2.2 Présentation de système Arduino.....	21
A) Applications d'Arduino	22
B) Pourquoi choisir Arduino.....	23
2.3.2.3 la carte Arduino	24
A) La composition de la carte arduino mega	26
B) La fiche technique de la carte arduino mega	26
C) Guide d'utilisation de la carte arduino méga.....	27
2.3.3 Le bras robotique	28
2.3.3.1 Le fonctionnement de bras robotisé	28
2.3.3.2 Les composants du bras robotique	29
A) L'architecture mécanique du bras	29
B) Le Servomoteur	31
B.1) Composition d'un servomoteur.....	31
B.2) La Commande d'un servomoteur	33
B.3) L'asservissement du servomoteur.....	34
B.4) Le choix des servomoteurs	35
B.5) Caractéristique des servomoteurs RC SR431 et RC SR311.....	37
2.3.3.3 La fiche technique du bras robotique	38
2.3.3.4 le shield de pilotage du bras	38
2.4 Discussion	40

Chapitre 3 : Conception logiciel du système

3.1 Préambule	41
3.2 Système Arduino	41
3.2.1 Présentation du logiciel Arduino	42
3.2.1.1 IDE Arduino	42
A) Description	42
B) L'interface du logiciel.....	43

C) Le langage Arduino.....	45
C.1) Fonctions de langage	46
C.2) Structure d'un programme	47
3.3 Organisation et définition des taches de notre système de tri.....	48
3.4 Organigramme général du système	50
3.5 Discussion	52

Chapitre 4 : Réalisation et test

4.1 Préambule	53
4.2 La Réalisation	53
4.2.1 Le capteur de couleur TCS 230	54
4.2.2 Test de capteur TCS 230	54
4.2.3 Le bras robotique	60
4.2.3 .1 Les différents mouvements du bras	61
A) Rotation horizontal du bras	61
B) Rotations verticale du bras	62
C) L'ouverture et la fermeture de la pince	63
4.2.3.2 Pilotage du bras avec arduino	63
A) Commande des servomoteurs	64
B) Connexion de shield avec carte arduino	65
C) Câblage des servomoteurs avec le shield	66
D) La programmation du bras	67
D.1) Partie déclaration	67
D.2) Partie initialisation	67
D.3) la partie principale	68
4.2.3.3 Test du bras.	69
A) Rotation verticale.....	69
B) Rotation horizontale de la base	71
C) Ouverture et fermeture de la pince.....	72

D) Déplacement d'un objet	73
4.2.4 Réalisation et test du système de tri	74
4.3 Discussion	77
Conclusion	78
Bibliographie	
Annexes	

Liste des figures

Chapitre 1 : Robotisation des chaines de productions

Figure 1.1 : Fonctionnement d'une chaine de production.....	2
Figure 1.2 : Chaine de production agroalimentaire.....	4
Figure 1.3 : Robot industrielle de soudure.....	5
Figure 1.4 : Robot de soudure à l'arc.....	6
Figure 1.5 : Robots de manutentions et de déplacement.	7
Figure 1.6 : Robot de packaging.	8
Figure 1.7 : Robots de la peinture.	9
Figure 1.8 : Centre tri logistique.	10

Chapitre 2 : Conception matériel du système de tri

Figure 2.1 : Schéma synoptique du système.	12
Figure 2.2 : Le capteur TCS230.	13
Figure 2.3 : Les composants de module TCS230.	13
Figure 2.4 : La matrice de photodétecteurs du capteur TCS230.	14
Figure 2.5 : Symbole de la photodiode.	14
Figure 2.6 : Bande d'énergie dans un semi-conducteur.	15
Figure 2.7 : La jonction PN du une photo diode.	16
Figure 2.8 : La matrice photodiodes de capteur.	17
Figure 2.9 : Schéma synoptique du capteur TCS230	17
Figure 2.10 : Longueur d'onde des couleurs.	18
Figure 2.11 : Schéma interne de capteur.	18
Figure 2.12 : Couleurs mesuré selon les photodiodes activés.	19

Figure 2.13 : circuit de conditionnement du capteur TCS230.....	20
Figure 2.14 : système arduino.....	22
Figure 2.15 : Exemples de projets arduino.....	23
Figure 2.16 : Exemple de différente carte arduino	24
Figure 2.17 : Elément constituant la carte arduino	26
Figure 2.18 : Exemple de bras robot à 6 degrés de liberté.....	29
Figure 2.19 : Architecture de bras robotique.....	30
Figure 2.20 : Les servomoteurs du bras robotique.....	30
Figure 2.21 : Un servomoteur.....	31
Figure 2.22 : Les éléments visibles du servomoteur.....	32
Figure 2.23 : Elément non visible de servomoteur.....	33
Figure 2.24 : illustration du signal de contrôle PWM.....	33
Figure 2.25 : L'asservissement du servomoteur.....	34
Figure 2.26 : notion de couple d' un servomoteur	36
Figure 2.27 : Les servomoteur du bras.....	37
Figure 2.28 : Le shield de contrôle des servomoteurs.....	39
Figure 2.29 : Angle de fonctionnement autorisé pour chaque servomoteur.....	40
 Chapitre 3 : Conception logiciel du système	
Figure 3.1 : Composition du système Arduino	41
Figure 3.2 : Programmation de la carte arduino.	42
Figure 3.3 : L'interface du logiciel Arduino.	43
Figure 3.4: Le menu fichier de l'IDE Arduino.	44
Figure 3.5 : Boutons pour la programmation des cartes.	45
Figure 3.6 : Les fonctions Setup et loop de langage arduino.	46

Figure 3.7 : La structure d'un programme.	47
Figure 3.8: Définition des taches du système de tri.	49
Figure 3.9 : Organigramme de fonctionnement du système.	51
Chapitre 4 : Réalisation et test	
Figure 4.1 : schéma globale du système.....	53
Figure 4.2 : schéma de branchement de capteur TCS230 a la carte arduino.....	54
Figure 4.3 : téléverser le programme de détection des couleurs vers la carte arduino.....	55
Figure 4.4: visualisation des résultats du detection de couleur sur le moniteur série.....	56
Figure 4.5: programme de définition des couleurs	56
Figure 4.6 : résultat de la détection de la couleur rouge.....	57
Figure 4.7 : résultat de la détection de la couleur verte.....	58
Figure 4.8 : résultat de la détection de la couleur bleue.....	59
Figure 4.9 : les 6 degrés de liberté du bras.....	60
Figure 4.10 : rotations horizontales du bras.	61
Figure 4.11 : les rotations verticales du bras.	62
Figure 4.12 : ouverture et fermeture de la pince.	63
Figure 4.13 : les servomoteurs du bras robotique.	63
Figure 4.14 : schéma synoptique de pilotage du bras robotique.	64
Figure 4.15: shield de commande des servomoteurs.	65
Figure 4.16 : connexion du shield avec carte arduino	65
Figure 4.17 : Câblage des servomoteurs avec le shield.	66
Figure 4.18: la déclaration des librairies < Servo.h> et <Braccio.h>.....	67
Figure 4.19 : Fonctions d'initialisation et configuration de la position initiale pour le bras...	68
Figure 4.20 : fonction contrôle des mouvements du bras robotique.	68
Figure 4.21 : l'instruction braccio.servoMovement.	69

Figure 4.22 : le programme de la rotation vertical.	69
Figure 4.23 : les rotations verticales de corps.	70
Figure 4.24 : le programme de la rotation de la base.	71
Figure 4.25 : les rotations horizontales de la base.	71
Figure 4.26 : le programme d'ouverture et de fermeture de la pince.	72
Figure 4.27 : ouverture et fermeture de la pince.	72
Figure 4.28 : le programme de déplacement d'un objet.	73
Figure 4.29 : déplacement d'un objet.	74
Figure 4.30 : Vue globale du système.	75
Figure 4.31 : l'opération de tri	76

Liste des tableaux

Tableau 2.1 : sélection de couleur à mesurer.....	18
Tableau 2.2 : sélection de la fréquence de sortie.	19
Tableau 2.3 : description des broches du module TCS230.	20
Tableau 2.4 : fréquence de travail, nombre de bits des différentes cartes arduino.....	25
Tableau 2.5 : entrée et sortie, espace mémoire des différentes cartes arduino	25
Tableau 2.6 : fiche technique de la carte arduino mega.	27
Tableau 2.7 : caractéristique du servomoteur.....	37
Tableau 2.8 : fiche technique du bras robotique.	38

Glossaire

USB : Universal Serial Bus : Bus universel en série.

IDE : Integrated Development Environment : environnement de développement libre.

UART : Universal Asynchronous Receiver Transmitter : émetteur-récepteur asynchrone universel.

PWM : Pulse Width Modulation : modulation de largeur d'impulsions.

I²C : Inter-Integrated Circuit.

SPI : Serial Peripheral Interface.

E/S : Entrées/Sorties.

Tx-Rx : Transmission et Réception.

ICSP : In Circuit Serial Programming.

EEPROM : Electrically-Erasable Programmable Read-Only Memory : mémoire morte effaçable électriquement et programmable.

IN : INPUT (Entrées).

OUT: OUTPUT (Sorties).

GND: Ground.

VCC, VDD : Alimentation Tension Continue.

CMOS: Complementary Metal Oxide Semiconductor.

3D : Trois dimensions ou tridimensionnel.

EDF : Électricité de France.

CAN : Convertisseur analogique numérique.

WIFI: Wireless Fidelity.

CA: Courant alternative.

CC: Courant continue.

DOF: Degrees of freedom: degrés de liberté.

Introduction

Introduction

Depuis quelques années, les chaînes de production industrielles et celles de la distribution sont de plus en plus robotisées. La robotique est essentiellement cantonnée au domaine manufacturier pour réaliser des tâches complexes, pénibles et/ou répétitives : l'assemblage, le soudage, la peinture, le tri sélectif, la manutention, la manipulation des produits dangereux (nucléaire), et pour permettre opérer dans des milieux inaccessibles à l'homme.

Certaines tâches dans une chaîne de production sont encore manuelles ou semi-automatiques, ce qui présente des difficultés et problèmes de sécurité dans certains cas pour l'être humain. Cela engendre inévitablement la diminution de la productivité et de la qualité des produits. Par conséquent, ceci constitue un facteur négatif pour le développement économique de l'entreprise, notamment, accroître ses parts de marché ou sa rentabilité. Par ailleurs, le développement du commerce électronique « e-commerce » nécessite une distribution plus efficace qui répond à une plus grande clientèle dont les transactions sont effectuées par milliers sur des plateformes web. Ceci demande un traitement automatique optimal au-delà des capacités humaines.

Dans ce cadre, notre projet consiste à concevoir et réaliser un système de tri sélectif robotisé à la fin d'une chaîne de production. Le critère de tri est basé sur la couleur du produit. Pour réaliser le système, nous avons utilisé un capteur de couleur et un bras robotique à 6 degrés de liberté. L'ensemble est commandé par une carte arduino Mega.

Pour bien présenter notre travail, le mémoire est composé de quatre chapitres. Le premier chapitre présente l'intérêt de la robotisation des chaînes de productions. La problématique de tri dans une chaîne de distribution n'étant pas vraiment différente, nous nous contentons dans cette étude à travailler essentiellement par rapport à des chaînes de production. Le deuxième chapitre est consacré à la conception matérielle de notre système et les différentes étapes le permettant. Le troisième chapitre est dédié à la conception logicielle et différents programmes qui permettent le fonctionnement du système. Le quatrième chapitre est réservé exclusivement à l'illustration de la réalisation pratique.

Chapitre 1

Robotisation des chaines de productions

1.1 Préambule

Dans ce chapitre nous nous sommes intéressés à l'illustration de l'intérêt de robotiser les chaînes de productions et de distribution. La robotisation de certaines tâches permet d'améliorer la productivité et la qualité des services. Les robots sont généralement utilisés pour effectuer des tâches dangereuses, hautement répétitives désagréables, pénibles et impossibles pour l'être humain.

1.2 Définition d'une chaîne de production

La ligne de production est l'ensemble des processus nécessaires et structurés pour la fabrication d'un produit manufacturé ou service allant des actions ou opérations de transformation des matières premières ou de produits semi-finis jusqu'à la distribution sur le marché de l'offre et de la demande. Une chaîne de production comporte tous les mécanismes utiles, organisés et coordonnés : bâtiment industriel, machines, équipements, maintenance, contrôle qualité, système intelligent d'information, capital humain performant etc. On donne comme exemple : chaîne de production automobile, agroalimentaire, production pharmaceutique, etc. La Figure 1.1 représente le principe de fonctionnement d'une chaîne de production. [1]



Figure 1.1 : fonctionnement d'une chaîne de production.

1.3 Historique de l'évolution de la chaîne de production

Au début les procédés de production ont été limités par la disponibilité d'une source d'énergie. Les moulins à vent et les moulins à eau fournissaient la puissance pour les processus lourds, et la main d'œuvre était employée en minoterie masse, surtout pour des activités exigeant plus de précision. Déjà au 18^e siècle avec les matières premières, l'énergie et la population en différents endroits, la production était distribuée entre plusieurs sites. La concentration de beaucoup de personnes dans les usines, et plus tard l'usine type, comme les filatures de coton, ont orienté la tendance vers différents processus localisés au même endroit.

Avec le développement du moteur à vapeur dans la dernière moitié du 18^e siècle, les éléments de production sont devenus moins dépendants de la localisation de la source d'énergie, et donc le traitement des marchandises fut déplacé à la source des matériaux au près du lieu de vie des personnes pour exécuter le travail. Des processus séparés, pour différentes étapes de traitement, ont été introduites dans le même bâtiment, et les diverses étapes du raffinage ou de la fabrication ont été combinées. Oliver Evans aux États-Unis a réuni les étapes du processus de minoterie dans les années 1780 pour former ce qui est identifié comme la première chaîne de production, le débit de sortie d'un processus alimentant directement le suivant. Avec l'augmentation de l'utilisation de la puissance de la vapeur, et l'augmentation des machines pour remplacer les personnes, l'utilisation intégrée des techniques dans des lignes de production a stimulé les révolutions industrielles de l'Europe et des États-Unis. La prochaine étape était le concept de la chaîne de montage, comme celle introduite par Eli Whitney. Ceci fut mis en place à la Compagnie Ford Motors en 1913, où Henry Ford introduisit l'innovation de déplacer, sans interruption, les voitures à assembler d'un poste de travail à l'autre.

Au fil du temps la chaîne de production est développée et améliorée pour augmenter la productivité et la qualité des biens et de services. Avec les développements scientifiques, et technique spécifiquement de l'électronique et de l'informatique mais aussi mathématique, mécanique, matériaux, engendrent l'intégration de la robotique dans tous les domaines, notamment en construction automobile, chez la plupart des fabricants d'ordinateurs, l'industrie agroalimentaire, pharmaceutique, centre logistique...etc. Leurs capacités d'effectuer rapidement des travaux répétitifs ne cessent de croître. On les emploie également

dans des environnements difficilement supportables par l'homme caractérisés par des conditions extrêmes de température ou de pression, radioactivité élevée. La figure 1.2 montre une chaîne de production agroalimentaire. [1]



Figure 1.2 : chaîne de production agroalimentaire.

1.4 Robotisation des chaînes de productions

1.4.1 Définition de la robotique

La robotique est la science et la technique qui s'intéresse aux robots, permettant la conception et la construction de machines automatiques ou de robots. En fait, il s'agit d'un ensemble de disciplines techniques (mécanique, électronique, automatique, informatique) articulées autour d'un objet communs. Cet objectif est l'automatisation flexible de nombreux secteurs de l'activité humaine réputés jusqu'à très récemment comme ne pouvant se passer de la présence de l'homme, et l'objectif c'est le robot, sorte de machine universelle dont l'homme rêve depuis toujours pour le remplacer dans les tâches difficiles. La figure 1.3 représente un robot industriel de soudure. [2]

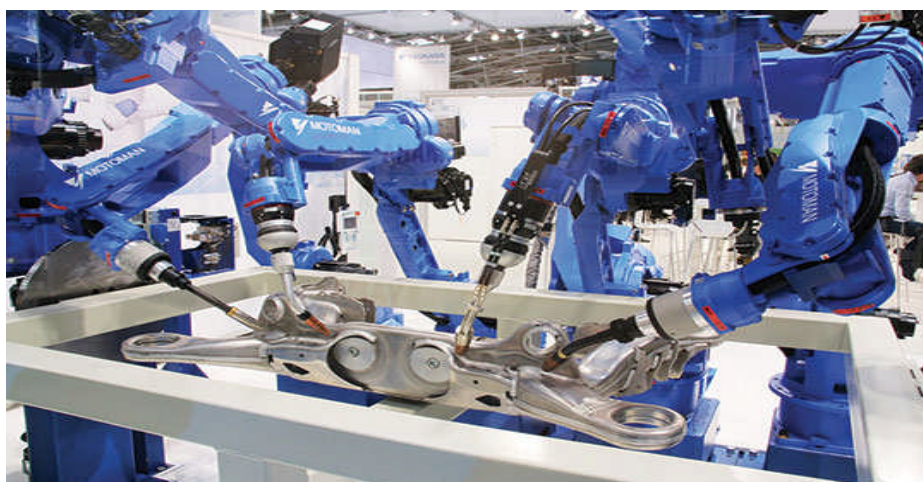


Figure 1.3 : robot industriel de soudure.

1.4.2 La robotisation de tâches dans une chaîne de production

Les robots sont généralement utilisés pour effectuer des tâches dangereuses, hautement répétitives désagréables, pénibles et impossibles pour l'être humain. Ils ont de nombreuses fonctions différentes telles que la manutention, l'assemblage, le soudage à l'arc, le soudage par résistance, la charge d'outil machine et les fonctions de déchargement, la peinture, le tri sélectif. Certaines tâches dans une chaîne de production sont encore manuelle ou semi-automatique, ce qui rend la tâche plus difficile et dangereuses dans certains cas pour l'être humain, cela engendre la diminution de la productivité de la qualité des produits, ce qui va être un facteur négatif pour le développement économique de l'entreprise, surtout avec la concurrence, chaque entreprise cherche à satisfaire ses clients et à les fidéliser afin d'accroître sa part de marché et sa rentabilité.

On va voir quelques applications et tâches manuelles à robotiser :

- Soudage à l'arc ;
- Manutention et service de machine ;
- Packaging et conditionnement ;
- Peinture (revêtement) ;
- Le tri sélectif.

1.4.2.1 Soudage à l'arc

Les autres avantages de la robotisation des opérations de soudage à l'arc résident dans la suppression de tâches pénibles et dangereuses pour la santé du fait de la nocivité des fumées ou de la difficulté à manipuler des pièces de dimensions importantes. La robotisation

permet également de faire face au manque de main-d'œuvre qualifiée dans ce domaine. Le gain de productivité est très important de deux à cinq fois plus que pour les opérations de soudage manuel. La figure 1.4 présente un robot de soudage. [2]

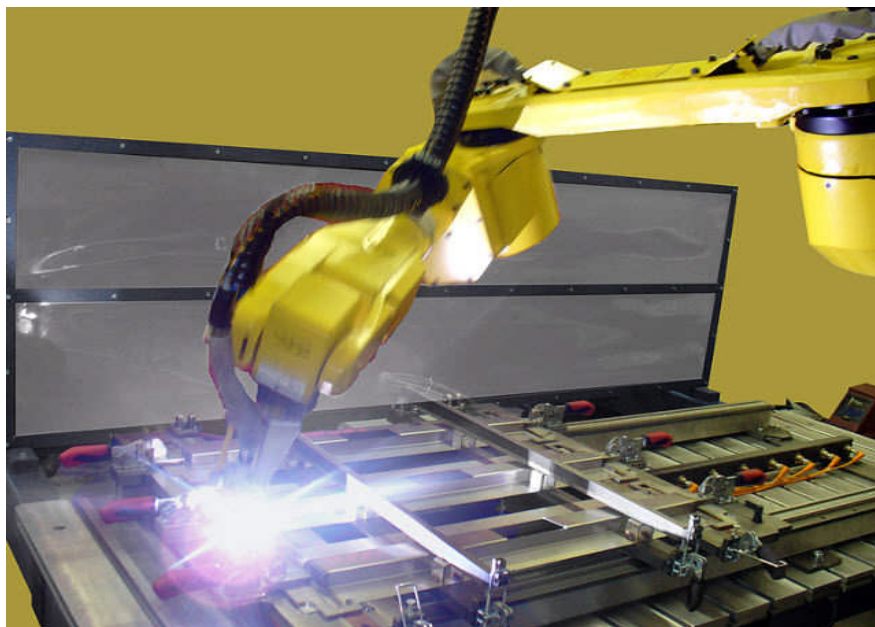


Figure 1.4 : robot de soudure à l'arc.

1.4.2.2 Manutention et service de machine

La manutention est le rôle primordial dans la chaîne de production, consiste à soulever, déplacer ou manutentionner des charges lourdes (pièce, produit, matière première...) à l'aide principalement de grues mobiles, mais aussi de bras de grues, de chariots (généralement sont des outilles manuelle) .Certaines filières de production agricole ou industrielle font appel à des manutentionnaires appelés aussi manœuvres. Ces employés réalisent des tâches manuelles peu qualifiées qui ne nécessitent aucune connaissance technique particulière. Les opérations de manutention sont les plus communément réalisées par des robots. Le robot manipule une ou plusieurs pièces (ou produits) pour les déplacer d'un endroit à un autre. La grande flexibilité du robot lui permet également de modifier les tâches à réaliser, de le déplacer en cas de besoin, et assuré la qualité et la sécurité de service à long terme. La figure 1.5 représente un robot de manutentions et de déplacement. [2]



Figure 1.5: robots de manutentions et de déplacement.

1.4.2.3 Packaging et conditionnement

Le conditionnement et l'emballage des matériaux et des produits final (produit :alimentaire, pharmaceutique, électroménagère ...) intègre les protections climatiques, physico-chimiques et mécaniques qui assure la sécurité du produit dans sa manutention, sa conservation, son stockage et son transport(agents de dégradation) , La robotique est la solution idéale pour répondre aux demandes du secteur car elle offre une très grande flexibilité grâce à une gamme très large de robots aux morphologies variées capables de s'adapter à toutes les utilisations.

Les robots de packaging sont typiquement utilisés autour d'une ligne équipée de convoyeurs et de systèmes de vision. Le packaging regroupe des opérations variées comme la dépalettisation l'emballage, l'encaissage, la palettisation ou l'emboilage et le picking. La figure 1.6 représente un robot de packaging. [2]



Figure 1.6 : robot de packaging.

1.4.2.4 Peinture (revêtement)

La peinture est un revêtement appliqué en une ou plusieurs couches sur des SUR différents pièces et produits industriels (châssis véhicule, , bois, plastiques, métaux, ...) .Dont les deux fonctions principales sont de protéger les supports (bois, plastiques, métaux, maçonneries...) et de les décorer (couleur, brillance, structure...). Selon les besoins, Certaines peuvent avoir des propriétés autonettoyantes, phytochromes, hydrophobes, dépolluantes, anticorrosives, anti-fissures, anti-moisissures, antibactériennes ou encore des propriétés permettant d'améliorer l'isolation thermique ou électromagnétique. Les applications de peinture et de pulvérisation avec robot se développent compte tenu des contraintes de plus en plus importantes imposées pour la protection de l'environnement et de santé. En effet, les produits utilisés sont souvent toxiques et doivent être appliqués de manière précise pour un rendu optimal. L'utilisation d'un robot spécifique peinture offre un contrôle précis du processus peinture (apprêts, base, vernis) et des différents paramètres (largeur de jet, air d'atomisation, pression...). [2]

La Figure 1.7 représente un robot de la peinture :



Figure 1.7 : robots de la peinture.

Les avantages pour l'utilisateur sont multiples. Tout d'abord, les quantités de peinture ou de solvants utilisées sont réduites, ce qui limite notablement l'impact sur l'environnement. En effet, la qualité de la trajectoire du robot associée au contrôle du processus d'application permet d'optimiser les épaisseurs spécifiées avec un minimum de variation, assurant un gain productivité, une qualité accrue des pièces, une diminution des quantités de peinture utilisées, et une baisse conséquente des coûts.

1.4.2.5 Le tri sélectif

Le tri sélectif consiste à trier les produits (pièce, matière première,) dans une chaîne de production ou dans les centres logistiques suivant plusieurs paramètres (nature, type, taille, couleur...) pour faciliter ensuite leur traitement : stockage, préparation, packaging et emballage, livraison etc.

Les systèmes de tri sont souvent utilisés pour les applications suivantes :

- Commerce de gros et de détail ;
- Chaîne de production agroalimentaire et industrielle ;
- Industrie pharmaceutique et cosmétique ;
- Centres de tri postal ;
- Commerce électronique et vente par correspondance ;
- Centres de tri de déchets.

Les systèmes de tri automatisés sont essentiels pour les entreprises à haute capacité de fabrication et distribution, pour augmenter constamment la qualité et la fiabilité des services. La robotisation de cette tâche (tri sélectif) permet de simplifier et de faciliter le tri, contrairement à la méthode manuelle, et amélioré la productivité et la qualité de tri, les robots trieurs sont adaptés à tous les types de chargements et peuvent être équipés de systèmes d'entrée et de sortie automatiques ou manuels. Figure 1.8 représente un centre tri logistique.



Figure 1.8 : centre tri logistique.

1.5 Présentation de travail à affecter (la problématique)

Dans le cadre de notre travail nous nous sommes intéressés au tri sélectif en fin d'une chaîne de production basé sur le critère de la couleur visible d'un produit. La tâche principale de notre système est de trier les produits selon 3 couleurs rouge, bleu et vert. Après la sélection, le produit ou la marchandise arrivent directement dans un bac de stockage, emballage ou sac spécifique à chaque couleur.

Afin d'aboutir à un système de tri sélectif (critère de sélection : la couleur) tel que c'est défini plus haut, il faudrait répondre à ces deux questions principales:

- Comment détecter la couleur du produit ?
- Comment sélectionner un produit une fois que sa couleur est détectée ?

1.6 Discussion

Dans ce chapitre on a présenté les avantages et les intérêts de robotiser les chaînes de production. Notre projet consiste à concevoir un système de tri sélectif robotisé et performant. Le chapitre suivant est consacré à la conception matérielle du projet, dont on va voir le principe de fonctionnement de tous les éléments constituant le système.

Chapitre 2

Conception matériel du
système de tri

2.1 Préambule

Dans ce chapitre, nous allons voir la conception matérielle de notre système de tri, et définir et expliquer les caractéristiques des éléments et composants électroniques constituant le système de tri.

2.2 Définition de tri sélectif

Le tri sélectif consiste à trier et séparer les produits (pièce, matière première...etc) dans une chaîne de production ou dans les centres logistiques suivant plusieurs paramètres (nature, type, taille, couleur des produit) pour faciliter ensuite leur traitement : stockage, préparation, packaging et emballage, livraison, transport etc.

2.3 Conception du système de tri

Le système que nous allons concevoir a pour objectif de détecter la couleur des produits sur la chaîne de production à l'aide d'un capteur de couleur, ensuite un bras robotique va placer le produit dans un bac de stockage spécifique à sa couleur, selon le cahier des charges suivant :

Notre système sera constitué d'une partie mesure (capteur de couleur), et une partie actionneurs (bras robotique), et une partie acquisition et commande (carte arduino) de bras et de capteur .Pour une bonne conception de notre projet, on doit comprendre le principe de fonctionnement des éléments constituant notre système (voir figure 2.1) :

- Capteur de couleur TCS230;
- Carte arduino(Mega).
- Bras robotique ;

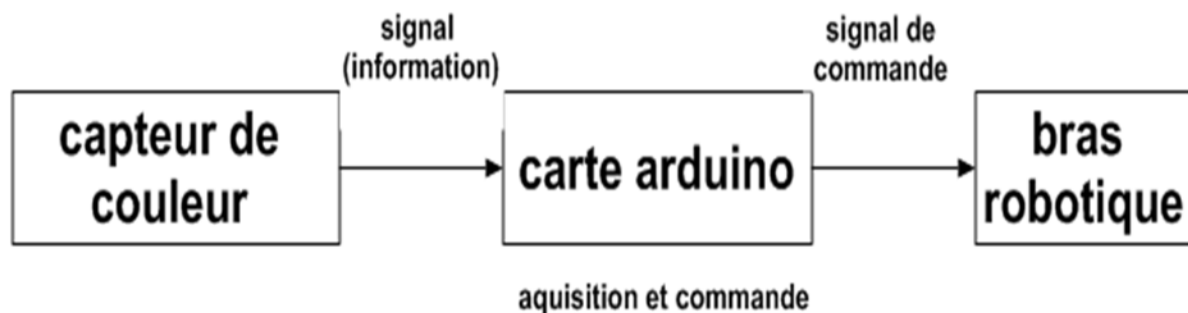


Figure 2.1 : schéma synoptique du système.

2.3.1 Capteur de couleur TCS230

2.3.1.1 Description

Nous allons voir en détails la perception des couleurs par le capteur TCS230 qui est représenté sur la figure ci-dessous :

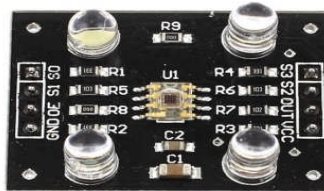


Figure 2.2 : le capteur TCS230.

Le capteur est un convertisseur couleur programmable lumière-fréquence, associe des photodiodes au silicium configurables et un convertisseur courant-fréquence sur un seul circuit intégré monolithique CMOS. La sortie est une onde carrée (Cycle d'utilisation de 50%) avec une fréquence directement proportionnelle à l'intensité lumineuse (éclairage énergétique). la sortie complète la fréquence peut être ajustée par l'une des trois valeurs prédéfinies via deux broches d'entrée de contrôle. Entrées numériques et sortie numérique permettre une interface directe avec un microcontrôleur ou un autre circuit logique, comme le montre la figure 2.3: [3]

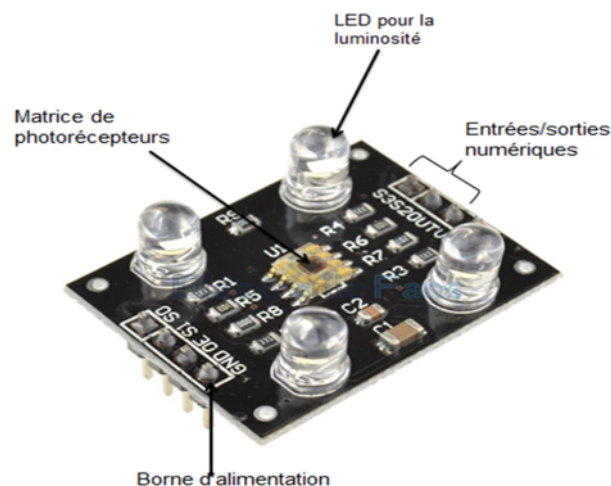


Figure 2.3 : les composants de module TCS230.

Le convertisseur lumière-fréquence est une matrice de photodétecteurs est composée d'un tableau de 8 x 8 photodiodes, comme le montre la figure suivante : [4]

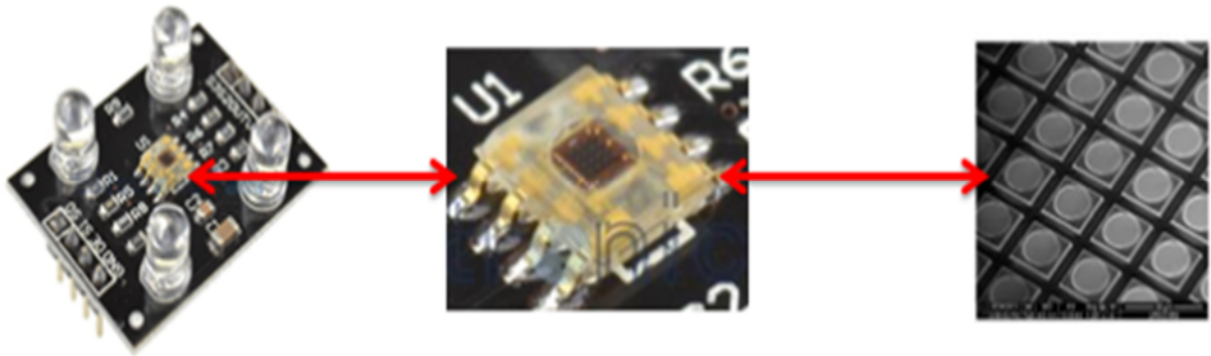


Figure 2.4 : la matrice de photodétecteurs du capteur TCS230.

Le capteur TCS230 est composé principalement de photodiode, nous allons voir le principe de fonctionnement de la photodiode :

A) photodiode

Une photodiode est un composant semi-conducteur ayant la capacité de détecter un rayonnement du domaine optique et de le transformer en signal électrique.

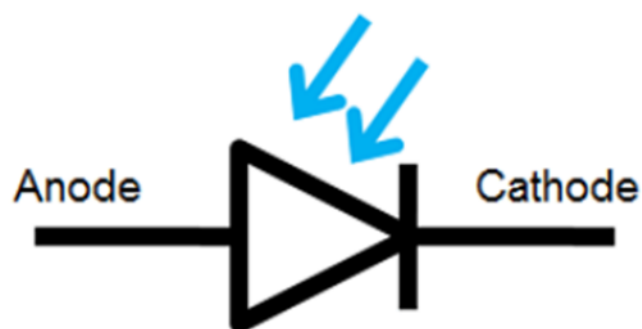


Figure 2.5 : symbole de la photodiode.

Comme beaucoup de diodes en électronique elle est constituée d'une jonction PN. Cette configuration de base fut améliorée par l'introduction d'une zone intrinsèque (I) pour constituer la photodiode PIN. En absence de polarisation (appelé mode photovoltaïque) elle

créée une tension. En polarisation inverse par une alimentation externe (mode photoampérique), elle crée un courant. On repère 3 régions distinctes : [5]

1. une zone de charge d'espace (ZCE) appelée couramment zone de déplétion et de diffusion.
2. une région neutre de type N.
3. une région neutre de type P.

B) fonctionnement de la photodiode

Lorsqu'un semi-conducteur est exposé à un flux lumineux, les photons sont absorbés à condition que l'énergie du photon ($E_{ph} = h\nu$) soit supérieure à la largeur de la bande interdite (E_g).



Figure 2.6 : bande d'énergie dans un semi-conducteur.

Ceci correspond à l'énergie indispensable que doit absorber l'électron afin qu'il puisse quitter la bande de valence (où il permet d'assurer la cohésion de la structure) vers la bande de conduction, le rendant ainsi mobile et capable de générer un courant électrique. L'existence de la bande interdite entraîne l'existence d'un seuil d'absorption tel que : $h\nu_0 = E_g$. Lors de l'absorption d'un photon, deux phénomènes peuvent se produire : [5]

La photoémission : c'est la sortie de l'électron hors du matériau photosensible. L'électron ne peut sortir que s'il est excité près de la surface.

La photoconductivité : l'électron est libéré à l'intérieur du matériau. Les électrons ainsi libérés contribuent à la conductivité électrique du matériau.

Quand les photons pénètrent dans le semi-conducteur pourvus d'une énergie suffisante, ils peuvent créer des photoporteurs en excès dans le matériau. On observe alors une augmentation du courant. Deux mécanismes interviennent simultanément :

Il y a création de porteurs minoritaires, c'est-à-dire des électrons dans la région P et des trous dans la région N. Ceux-ci sont susceptibles d'atteindre la ZCE par diffusion et d'être ensuite propulsés vers des zones où ils sont majoritaires. En effet, une fois dans la ZCE, la polarisation étant inverse, on facilite le passage des minoritaires vers leur zone de prédilection. Ces porteurs contribuent ainsi à créer le courant de diffusion comme le montre la figure 15 ci-dessous. [5]

Il y a génération de paire électron trou dans la ZCE, qui se dissocient sous l'action du champ électrique ; l'électron rejoignant la zone N, le trou la zone P. Ce courant se nomme le courant de transit ou photocourant de génération.

Ces deux contributions s'ajoutent pour créer le photocourant qui s'additionne au courant inverse de la jonction, comme le montre la figure suivante :

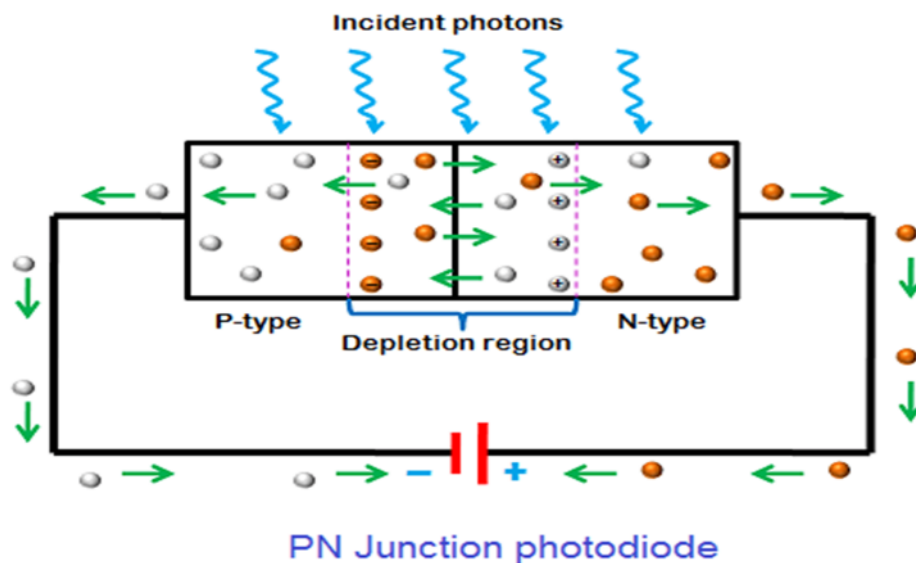


Figure 2.7 : la jonction PN d'une photodiode.

2.3.1.2 Comment le capteur de couleur identifie les couleurs

La matrice est composée de 16 photodiodes ont des filtres bleus, 16 photodiodes ont des filtres verts, 16 photodiodes ont des filtres rouges et 16 photodiodes sont claires sans filtres, qui sont représentés sur la figure suivante : [4]



Figure 2.8 : la matrice photodiodes de capteur.

Dans la matrice se trouve un oscillateur qui produit un signal dont la fréquence est proportionnelle à l'intensité de la couleur choisie. Cette matrice capte les différentes couleurs composées d'ondes et les transmet à un convertisseur, qui les converties en fréquence. [6]

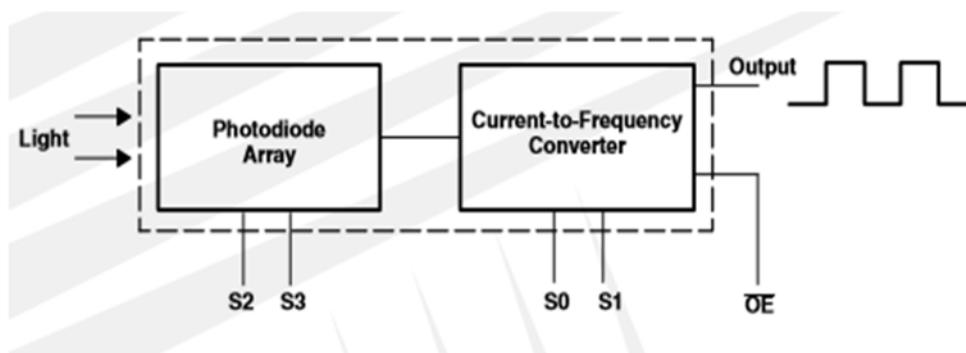


Figure 2.9 : schéma synoptique du capteur TCS230.

On peut voir sur la figure 2.10 qu'une longueur d'onde correspond à une fréquence qui définit une couleur. La fréquence correspond à une période qui résulte de la division $1/T$.

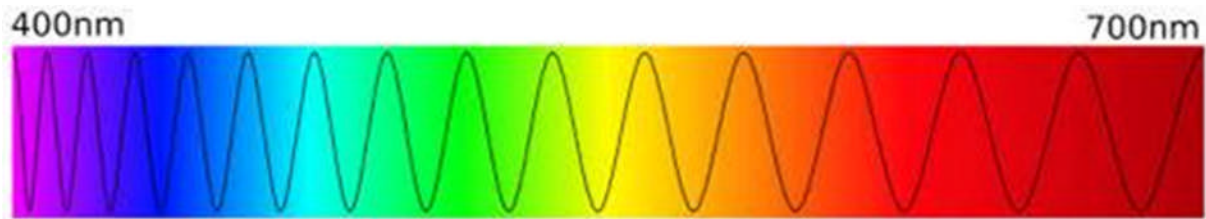


Figure 2.10 : spectre de couleur visible longueur d'onde.

Les 16 photodiodes de la même couleur sont connectées en parallèle, et comme le capteur ne contient qu'un seul convertisseur lumière-fréquence, il est nécessaire de sélectionner la couleur à mesurer, en utilisant les deux broches de commande S2 et S3, nous pouvons sélectionner celles qui seront lues. Ceci est accompli simplement en adressant une entrée de multiplexeur appropriée, qui représenté sur la figure 2.11 : [4]

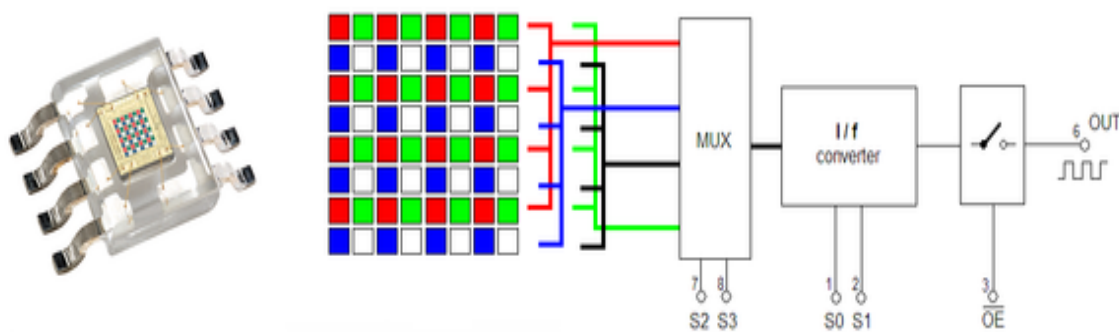


Figure 2.11 : schéma interne de capteur.

Par exemple, si nous voulons détecter la couleur rouge, nous pouvons simplement utiliser les 16 photodiodes filtrées en rouge en réglant les deux broches S2 et S3 sur le niveau logique bas en fonction du tableau suivant 2.1: [4]

S2	S1	Type de photodiode
L	L	Rouge
L	H	Bleu
H	L	Clair (sans filtre)
H	H	Vert

Tableau 2.1 : sélection de couleur à mesurer. [3]

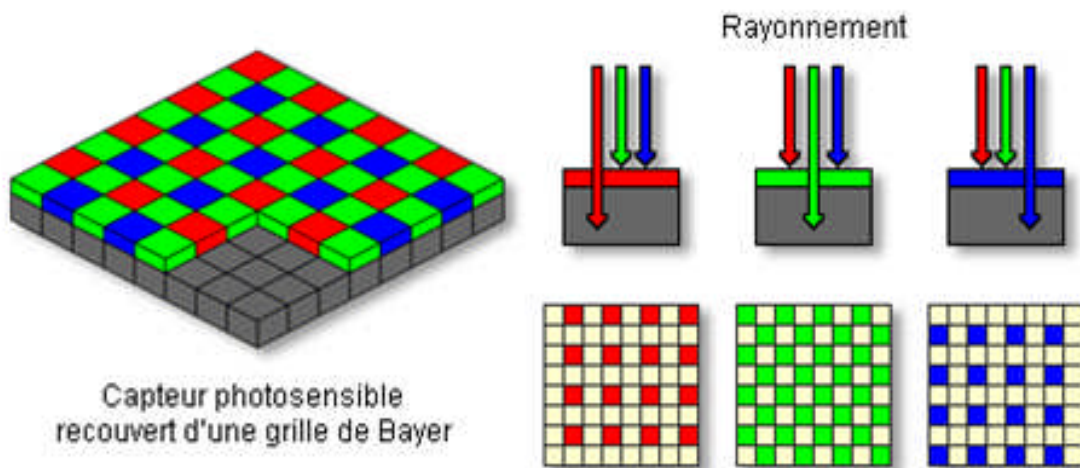


Figure 2.12 : couleurs mesuré selon les photodiodes activés.

Le capteur dispose de deux autres broches de commande, S0 et S1, qui servent à mettre à l'échelle la fréquence de sortie. La fréquence peut être ajustée à trois valeurs prédéfinies différentes 100%, 20% ou 2%. Cette fonction de mise à l'échelle de fréquence permet d'optimiser la sortie du capteur pour différents compteurs de fréquence ou microcontrôleurs en fonction de tableau 2.2 suivant : [4]

S0	S1	Échelle de fréquence de sortie
L	L	Éteindre
L	H	2%
H	L	20%
H	H	100%

Tableau 2.2 : sélection de la fréquence de sortie. [3]

2.3.1.3 Caractéristique de capteur TCS230

Nous allons voir les différentes caractéristiques du capteur, et la recommandation d'utilisation pour assurer le bon fonctionnement.

La figure ci-dessous représente circuit de conditionnement du capteur TCS230:

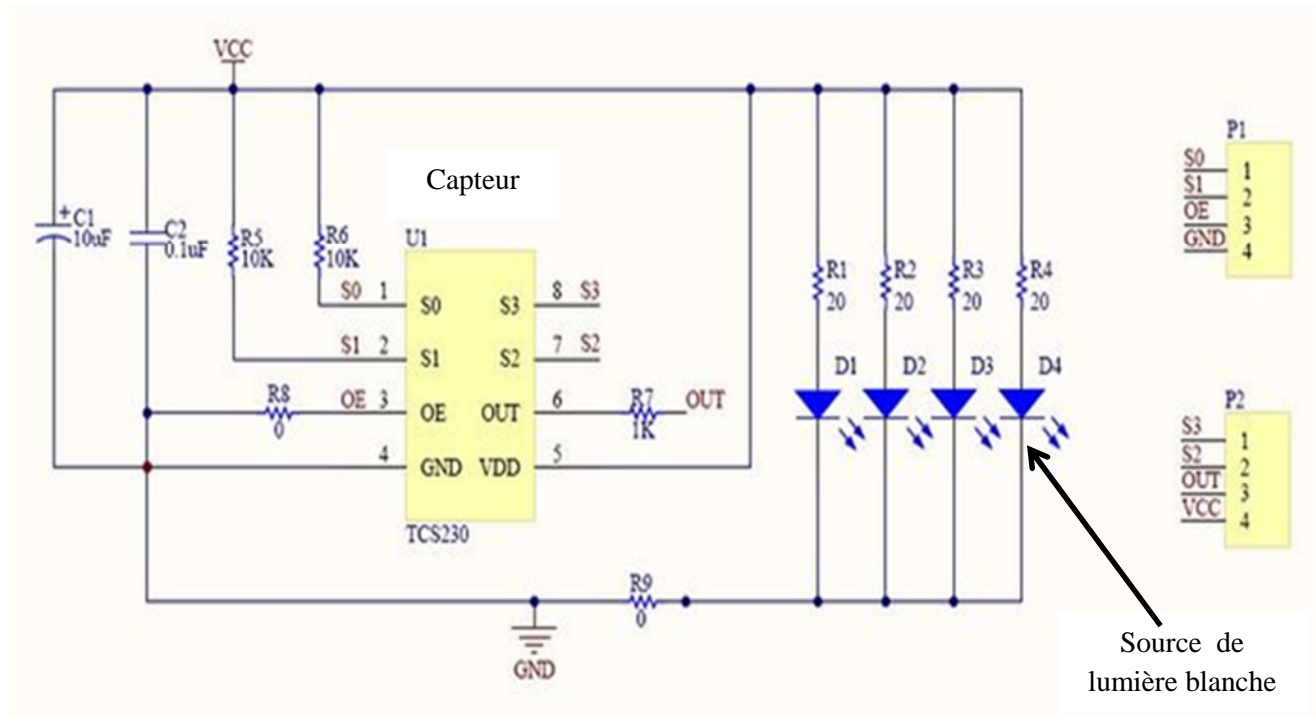


Figure 2.13 : circuit de conditionnement du capteur TCS230. [7]

Le module TCS230 contient 8 broches (voir annexe A), nous allons voir l'utilité de chaque broche dans le tableau suivant :

Broche	Entré/sortie	Description
Nom	N°	
GND	4	La masse 0 volt .
OE	3	Entré
OUT	6	Sortie
S0, S1	1, 2	Entré
S2, S3	7, 8	Entré
VDD	5	Alimentation 5 volts .

Tableau 2.3 : description des broches du module TCS230. [3]

❖ Fiche technique du TCS230 (voir annexe A) [3]

- ✓ Tension d'alimentation VDD : 2,7 à 5,5 V.
- ✓ Tension d'entrée de haut niveau : 2 à VDD.
- ✓ Tension d'entrée de bas niveau : 0 à 0,8 V.
- ✓ Plage de température à l'air libre : 0 à 70 °C.
- ✓ Meilleure plage de détection: 10mm.
- ✓ Erreur de non-linéarité: 0,2% à 50 kHz.
- ✓ Coefficient de température: 200 ppm/°.
- ✓ Cycle de charge: 50%.
- ✓ Dimensions: 30 x 28 x 7 mm.
- ✓ Poids: 18 gramme.

2.3.2 La Carte Arduino**2.3.2.1 Historique**

Le projet « Arduino » a été initié par un groupe d'enseignants et d'étudiants d'une école de design italienne en 2004 – 2005. Le projet répondant au doux nom de Arduino et mettant en œuvre une petite carte électronique programmable et un logiciel multiplateforme, qui puisse être accessible à tout un chacun dans le but de créer facilement des systèmes électroniques. [8]

2.3.2.2 Présentation de système Arduino

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

Le système Arduino est une plateforme open source basée sur une carte programmée à microcontrôleur AVR et un environnement de développement permettant d'écrire, compiler et tester un programme. Les cartes et modules Arduino sont pourvus d'entrées-sorties qui peuvent recevoir des signaux de capteurs ou interrupteurs et peuvent commander des moteurs, éclairages, etc. [9]

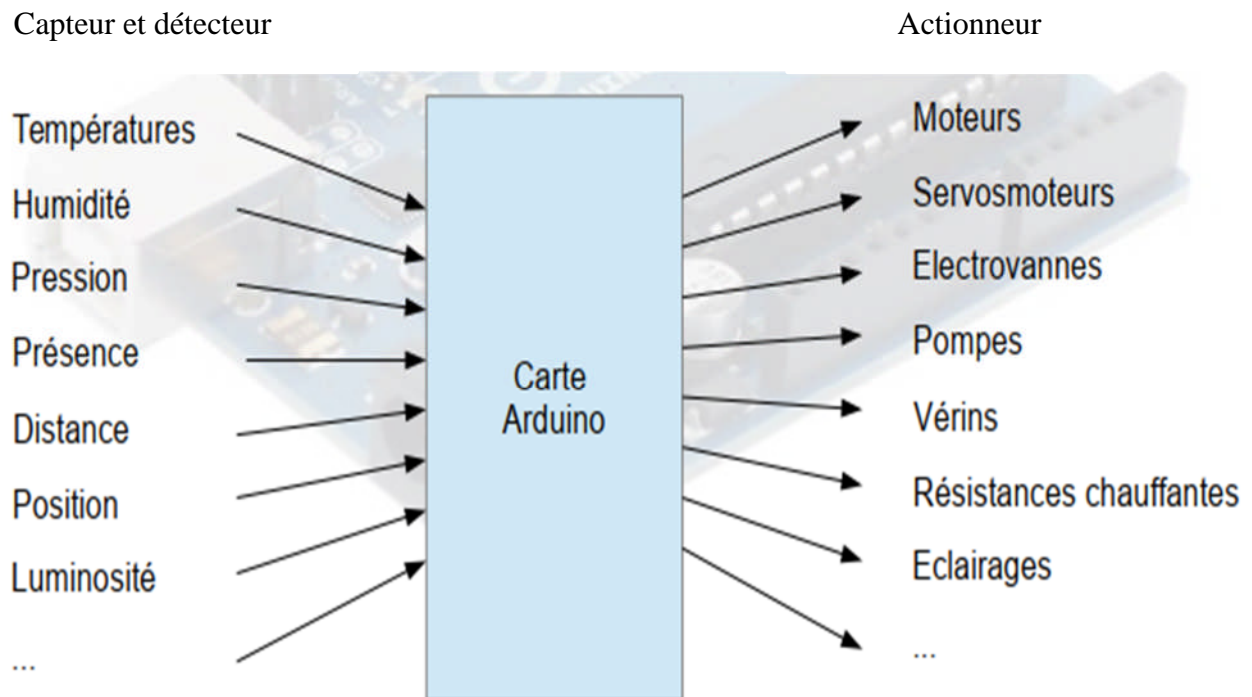


Figure 2.14 : système arduino. [10]

Cet environnement matériel et logiciel donne la possibilité d'allier les performances de la programmation à celles de l'électronique, 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, ce qui va permettre à l'utilisateur de formuler ses projets par l'expérimentation directe.

Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions. [11]

A) Applications d'Arduino

Le système Arduino nous permet de réaliser un grand nombre d'applications dans tous les domaines, l'étendue de l'utilisation de ce système est gigantesque. Pour quelques exemples, nous pouvons: [11]

- Contrôler les appareils domestiques;
- Fabriquer un robot;
- Faire un jeu de lumières;
- Communiquer avec l'ordinateur;

- Télécommander un appareil mobile (modélisme);
- Construire un appareil de mesure de tension/courant;
- Fabriquer une imprimante 3D;
- Récupérer les informations de consommation via le téléinformatique d'EDF ;
- Domotique et maison intelligente ;
- Etc.

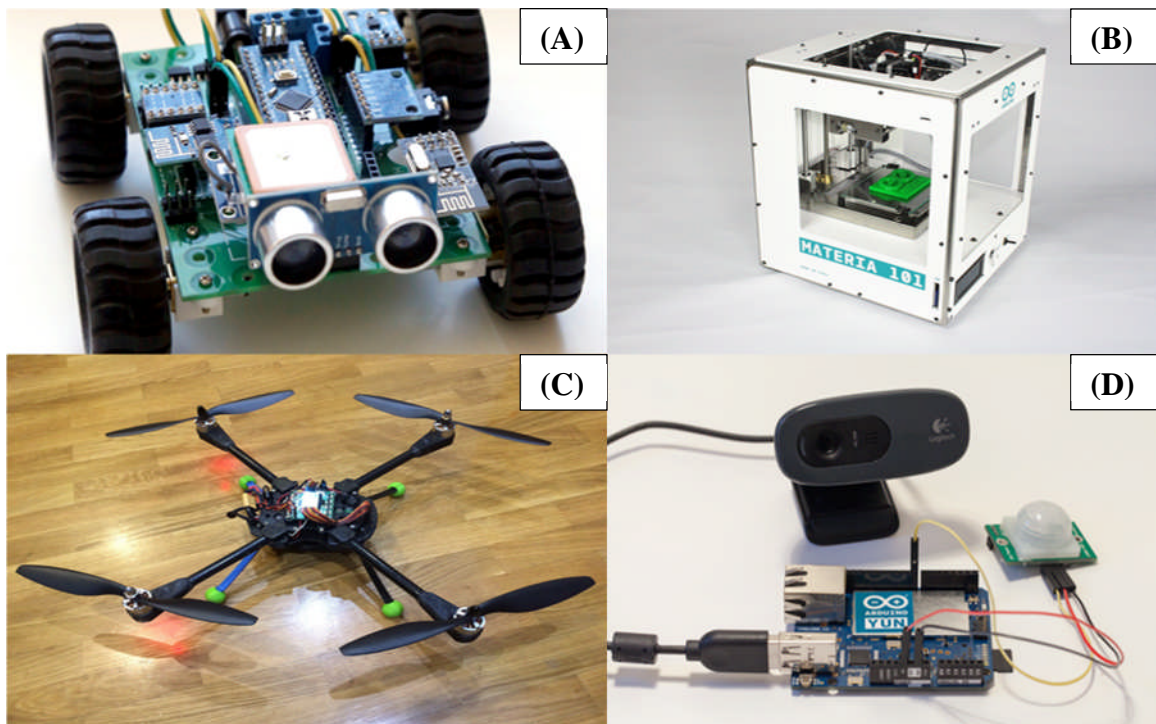


Figure 2.15 : exemples de projets arduino. (A) robot mobile,(B) imprimante3D,
(C) drone , (D) système de surveillance.

B) Pourquoi choisir Arduino [12]

1. Arduino est « Open Source ». Ce qui veut dire que vous pouvez récupérer le schéma d'origine, le modifier et l'utiliser pour produire la carte et la vendre sans payer des droits d'auteur.
2. Une compatibilité sous toutes les plateformes, à savoir : Windows, Linux et Mac OS.
3. Une liberté quasi absolue. Elle constitue en elle-même deux choses :
 - Le logiciel : gratuit et open source, développé en Java, dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

- Le matériel : cartes électroniques dont les schémas sont en libre circulation sur internet.
4. Une communauté ultra développée ! Des milliers de forums d'entraide, de présentations de projets, de propositions de programmes et de bibliothèques etc...

2.3.2.3 la carte Arduino

Les cartes Arduino possèdent un microcontrôleur facilement programmable ainsi que de nombreuses entrées-sorties. Plusieurs cartes Arduino existent et qui se différencient par la puissance du microcontrôleur ou par la taille et la consommation de la carte, par exemple : Arduino Uno, Genuino, Arduino Nano, Leonardo, Arduino mega, Seeduino...etc, qui sont présente sur la figure 2.16:

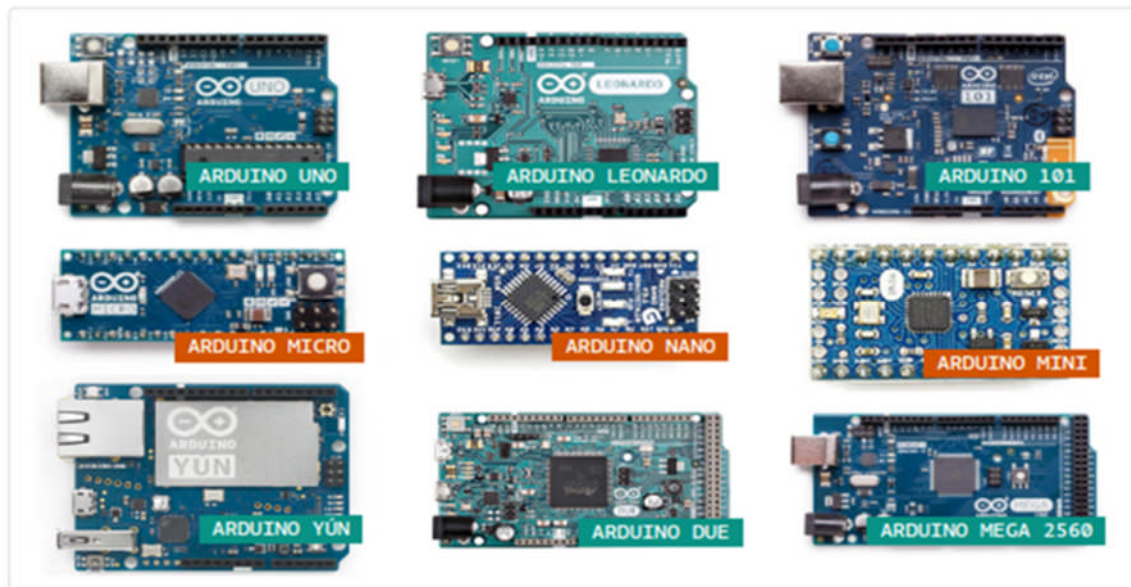


Figure 2.16 : exemple de différentes cartes Arduino .

Le choix du type de carte Arduino s'effectue en fonction des besoins de votre projet, il y a des critères à prendre en compte pour choisir sa carte :

- **Le nombre d'entrées-sorties numériques ou analogiques ;**
- **Le Microcontrôleur ;**
- **La vitesse de travail ;**
- **La Taille de la mémoire ;**
- **possibilités de communication (I2C, SPI, CAN, Bluetooth , Wifi).**

Les deux tableaux ci-dessous dressent un comparatif de différentes cartes utilisables, on ne prend pas en compte les cartes spécialisées pour l'internet ou l'équipement de vêtements. Ces tableaux vous donnent une vue synoptique des possibilités de chaque carte, ce qui peut déjà vous aider dans votre choix. [13]

Carte Arduino	Prix (euros)	Microcontrôleur (µC)	Type (bits)	CLK (MHz)	Alim (V)
Micro	18	ATmega32U4	8	16	5
Leonardo	18	ATmega32U4	8	16	5
Uno	20	ATmega328P	8	16	5
Nano	20	ATmega328P	8	16	5
MKR Zero	21	ATSAMD21G18	32	48	3,3
101 ⁽²⁾	29	Intel Curie	32	32	3,3
Due	34	AT91SAM3X8E	32	84	3,3
Mega	35	ATmega2560	8	16	5
Zero	39	ATSAMD21G18	32	48	3,3
M0 PRO	39	ATSAMD21G18	32	48	3,3

Table 2.4 : fréquence de travail, nombre de bits des différentes cartes arduino. [13]

Carte Arduino	Flash (KB)	Bootloader (KB)	SRAM (KB)	EEPROM (KB)	Digital I/O	Sorties PWM	Analog I/O
Micro	32	4	2,5	1	20	7	12
Leonardo	32	4	2,5	1	20	7	12
Uno	32	0,5	2	1	14	6	6
Nano	32	2	2	1	14	6	8 ⁽¹⁾
MKR Zero	256	8	32	-	22	12	7/1
101 ⁽²⁾	196	-	24	-	14	4	6
Due	512	-	96	-	54	12	12/2
Mega	256	8	8	4	54	15	16
Zero	256	-	32	-	20	10	6/1
M0 PRO	256	-	32	-	20	12	6/1

Table 2.5 entrée et sortie, espace mémoire des différentes cartes arduino. [13]

Pour une bonne conception de notre système de tri, on a choisi d'utiliser la carte arduino Mega, elle est la mieux adapté pour notre application, non seulement elle permet de contrôler tout le système (capteur TCS230 et bras robotique), mais aussi elle offre une possibilité de modifier, améliorer, et ajouter des extensions pour notre système.

A) La composition de la carte arduino Mega

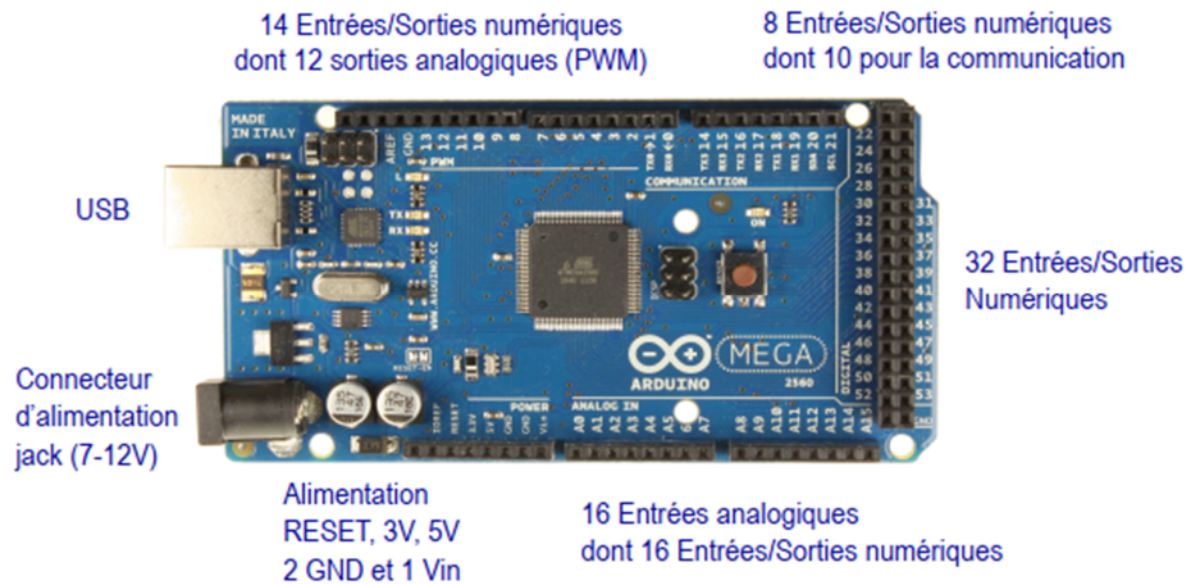


Figure 2.17 : élément constituant la carte arduino.

L'Arduino Mega 2560 est une carte à microcontrôleur basée sur l'ATmega2560. Il possède 54 broches d'entrée / sortie numériques (dont 15 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur à quartz de 16 MHz, une connexion USB, une prise d'alimentation, un en-tête ICSP, et un bouton de réinitialisation. Il contient tout le nécessaire pour prendre en charge le microcontrôleur, connectez-le simplement à un ordinateur avec un câble USB ou alimentez-le avec un adaptateur CA / CC ou une batterie pour démarrer. [14]

B) La fiche technique de la carte arduino Mega

Le tableau 2.6 présente toutes les informations et caractéristiques concernant la carte arduino mega :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limites)	6-20V
Broches numériques E / S	54 (dont 14 fournissent une sortie PWM)
Pins d'entrée analogiques	16

Courant continu par broche I / O	20 Ma
Courant DC pour 3.3V Pin	50 mA
Mémoire Flash	256 Ko, dont 8 KB utilisé par boot loader
SRAM	8 Ko (ATmega2560)
EEPROM	4 Ko (ATmega2560)
Vitesse d'horloge	16 MHz
Longueur	101,52 mm
Largeur	53.3 mm
Poids	37 g

Tableau 2.6 : fiche technique de la carte arduino Mega.

C) Guide d'utilisation de la carte arduino Mega [13]

- ✓ **Interface USB/série** : Permet d'établir une communication avec un ordinateur, directement avec un câble USB, afin de programmer le contrôleur ou d'échanger des informations avec un programme qu'il exécute.
- ✓ **Alimentation** : Ce circuit assure l'alimentation de l'ensemble des composants et des sorties suivant deux modes différents :
 - ❖ Lorsque la carte est connectée à un ordinateur via USB, c'est le port USB de l'ordinateur qui fournit l'énergie (5 V).
 - ❖ Lorsqu'on branche une source d'énergie au connecteur de la carte (batterie, transformateur ou pile), le système peut fonctionner de manière autonome.

On conseille en général de l'alimenter plutôt entre 7 et 12 V pour garder une marge en basse tension et éviter que le circuit ne chauffe trop.
- ✓ **Entrées/Sortie** : C'est à partir de ces connexions que le microcontrôleur est relié au différents capteur et actionneur. Une carte Arduino standard est dotée de :
 - ❖ 16 entrées **analogiques**.
 - ❖ 54 entrées/sorties **numériques** dont 14 peuvent assurer une sortie **PWM**

Certaines broches ont des fonctions spécialisées:

- ✓ **Communication Série** : Broches 0 (RX) et 1 (TX), Broches19 (RX) et 18 (TX) Broches 17 (RX) et 16 (TX), Broches 15 (RX) et 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega8U2 programmé en

convertisseur USB-vers-série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur.

- ✓ **Interruptions Externes** : Broches 2, 3, 18, 19,20 et 21. ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- ✓ **SPI (Interface Série Périphérique)** : Broches 50(SS), 51 (MOSI), 52 (MISO), 53 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI.
- ✓ **I2C**: Broches 20 (SDA) et 21 (SCL). Supportent les communications de protocole I2C. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.
- ✓ **Mémoire** L'ATmega328 a 256Ko (avec 8 KB occupées par le boot loader). Il a également 8 Ko de SRAM et 4 Ko de mémoire EEPROM.

2.3.3 Le bras robotique

Un bras robotique est un système mécanique poly-articulé, généralement programmable commandé en position, polyvalent, a plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces, des outils et des dispositifs spécialisés, au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches avec quelques fonctions similaires à un bras humain. Il est fabriqué à l'aide des composants : châssis, moteur, microcontrôleur. Le bras robotique ne sera plus un outil qui effectue une tâche répétitive dans un espace sécurisé, mais un partenaire des activités quotidiennes. Les contraintes du bras robotique sont liées à la sécurité et au confort de l'homme est destiné à faciliter les tâches pour l'homme et amplifier le rendu. Prenons l'exemple du bras manipulateur en milieu industriel, il réalise des tâches de façon répétitive, à hautes vitesses, pour satisfaire le temps d'exécution.

La conception de notre système de tri est basée sur le bras robotique, qui représente la partie actionneur du système, qui est commandé par la carte arduino.

Nous allons voir le principe de fonctionnement et les caractéristiques du bras robotisé :

2.3.3.1 Le fonctionnement de bras robotisé

Nous avons utilisé un bras robotique qui possède plusieurs caractéristiques, capable de lever un poids de 150g et d'effectuer des mouvements cycliques, et la plus importante caractéristique est le nombre de degrés de liberté (DOF = Degree Of Freedom), chaque degré de liberté représentant un mouvement que le bras peut réaliser. L'objectif étant

d'imiter le bras humain, il nécessite donc une articulation qui permet soit un mouvement de rotation, soit de translation.[2]

La figure suivante représente les degrés de liberté d'un bras robot :

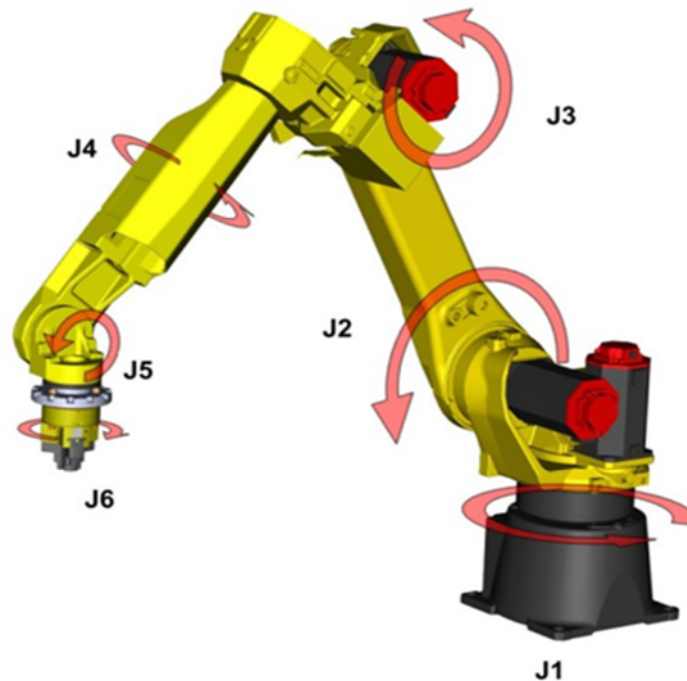


Figure 2.18 : exemple de bras robot à 6 degrés de liberté.

2.3.3.2 Les composants du bras robotique

A) L'architecture mécanique du bras

Généralement l'architecture mécanique d'un bras robotique est constituée :

- base : socle, disque rotatif ;
- corps (segment) : bras supérieur, bras inférieur ;
- Articulation (actionneur) : moteur, servomoteur ;
- organe terminal (outil) : effecteur, une pince ...etc.

La figure 2.19 représente l'architecture de mécanique de bras robotique qu'on va utiliser dans notre projet. Pour plus de détails voir annexe (B)

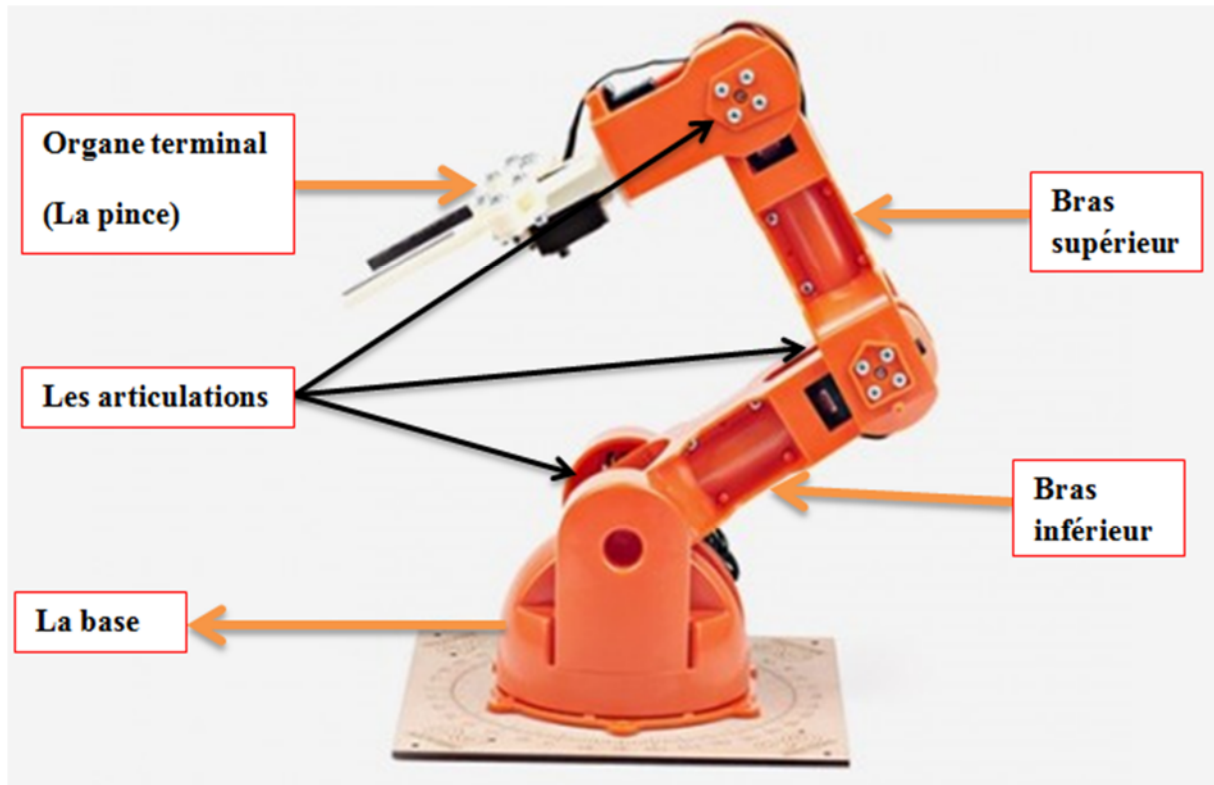


Figure 2.19 : architecture de bras robotique.

Les articulations du bras robotique sont articulées grâce à six servomoteurs qui permettent de reproduire les différents mouvements, qui sont représentés sur la figure ci-dessous :

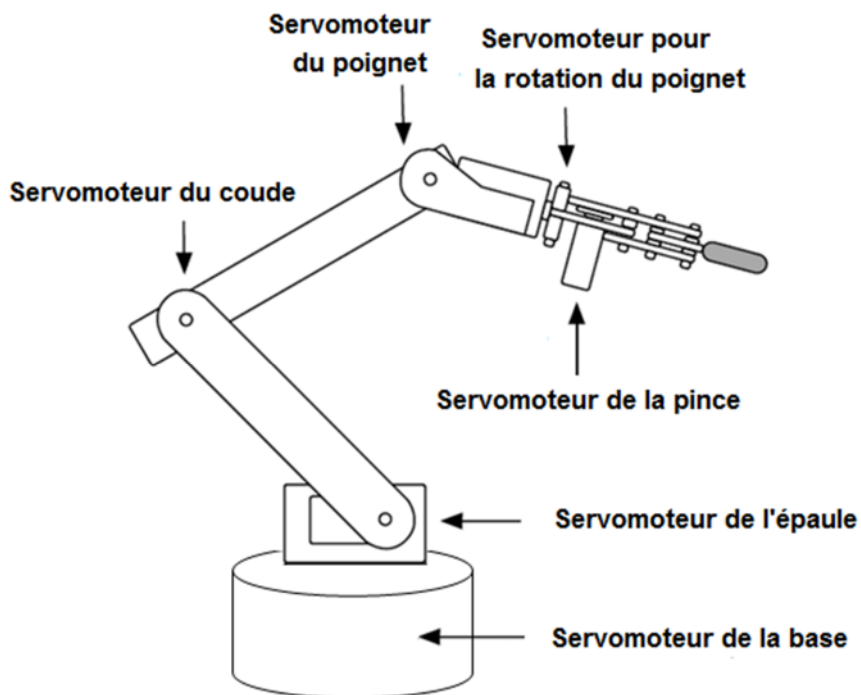


Figure 2.20 : les servomoteurs du bras robotique.

B) Le Servomoteur

Le servomoteur est un moteur électrique capable d'effectuer un déplacement angulaire et de maintenir une opposition à un effort statique, et dont la position est vérifiée en continu et corrigée en fonction de la mesure, donc c'est un système asservi. La plage de l'angle est généralement de 0 à 180 degrés, il en existe également dont l'angle peut se débattre sur seulement 90° et d'autres ayant un plus grand débattement sur 360°, et il y a ceux qui ont la possibilité de faire plusieurs tours sont souvent appelés servo-treuil. Enfin, les derniers, qui peuvent faire tourner leur axe sans jamais se buter, sont appelés servomoteurs à rotation continue. [15]

Donc le servomoteur est un actionneur (système produisant une action) qui mélange l'électronique, la mécanique et l'automatique.

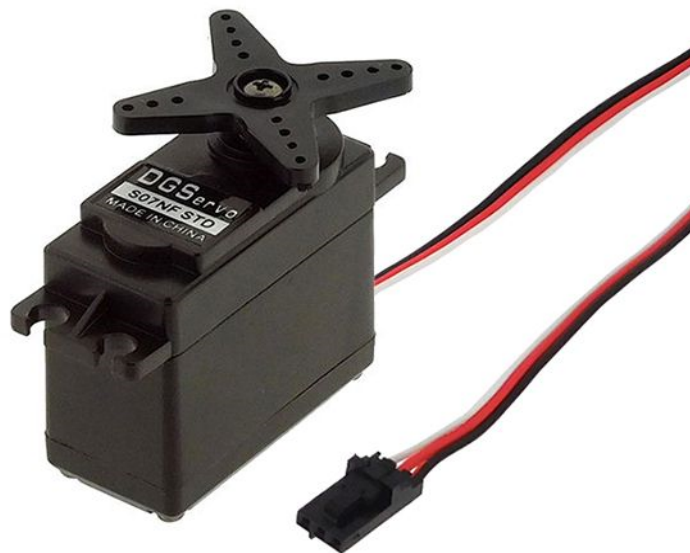


Figure 2.21 : un servomoteur.

B.1) Composition d'un servomoteur

Les servomoteurs ont un rôle primordial pour contrôler notre bras robotique, parce qu'ils ont l'avantage d'être asservis en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera une consigne d'orientation que vous lui envoyez en son entrée, l'électronique interne du servomoteur essaiera de conserver et de maintenir cette position, et

quelle que soit la force que l'on exerce sur le bras du servomoteur, ce qui permet au bras robotique d'effectuer les différents mouvements.

Le servomoteur est composé de plusieurs éléments visible et non visible :

Les éléments visibles (voir figure 2.22) : [16]

- Les fils, qui sont au nombre de trois :
 - rouge : pour l'alimentation positive (4.5V a 6V en général)
 - noir ou marron : pour la masse (0V)
 - orange, jaune, blanc : entrée du signal de commande.
- Palonnier ou l'axe de rotation sur lequel est monte un accessoire en plastique ou en métal.
- Le boitier (corps) qui le protège.

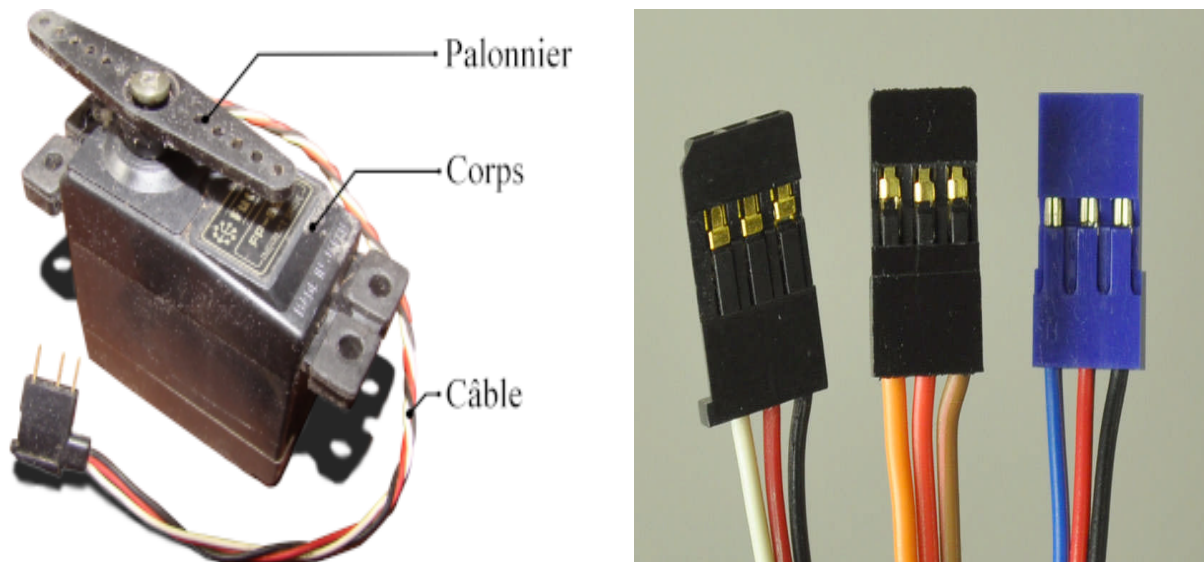


Figure 2.22 : les éléments visibles du servomoteur.

Les éléments non visibles (voir figure 2.23) [16]

- un moteur à courant continu.
- des engrenages pour former un réducteur, dont l'objectif est de réduire la vitesse et d'augmenter le couple (la puissance).
- un capteur de position de l'angle d'orientation de l'axe (un potentiomètre bien souvent), qui permet au servomoteur de garder l'angle d'inclinaison choisit.
- une carte électronique pour le contrôle de la position de l'axe et le pilotage du moteur à courant continu.

Le moteur est fixé par engrenages à la roue de commande. Au fur à mesure que le moteur tourne, la résistance du potentiomètre change, de sorte que le circuit de commande peut réguler précisément la quantité de mouvement et dans quelle direction. [16]

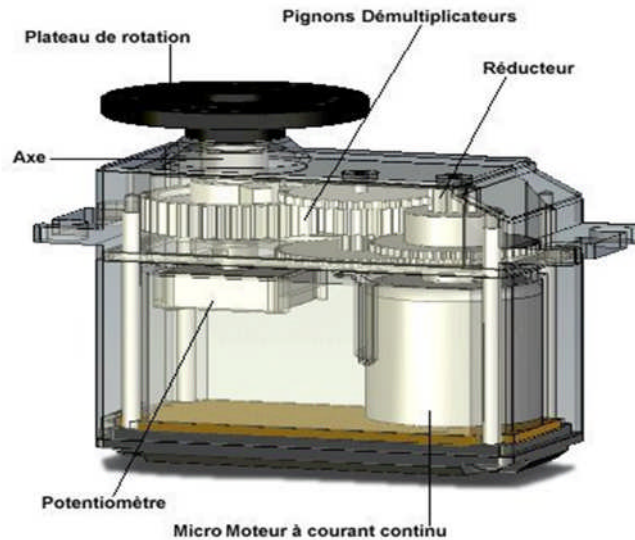


Figure 2.23 : élément non visible de servomoteur.

B.2) La Commande d'un servomoteur

Les servomoteurs sont contrôlés en envoyant une impulsion électrique de largeur variable, ou modulation de largeur d'impulsion (PWM), à travers le fil de commande. Il existe une impulsion minimale, une impulsion maximale et un taux de répétition. Cette impulsion de quelques millisecondes qui se répète à intervalle régulier (ce que l'on appelle la période), et dont la durée détermine l'angle, comme le montre la figure suivante : [16]

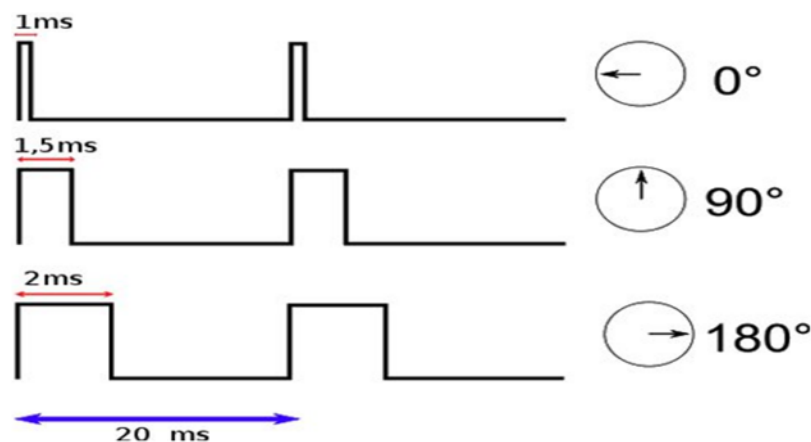


Figure 2.24 : illustration du signal de contrôle PWM pour 0°, 90° et 180°. [16]

Pour que le servomoteur reste à une position donnée, il faut transmettre toutes les 20 millisecondes (soit à une fréquence de 50Hz) une impulsion (état haut) d'une longueur comprise entre 1 et 2 millisecondes :

- Une impulsion de 1 milliseconde correspond à un angle de 0° ;
- Une impulsion de 2 millisecondes correspond à un angle de 180° ;
- En envoyant une impulsion d'une longueur intermédiaire, on obtient des angles différents, 90° avec une impulsion de 1.5 milliseconde par exemple.

A travers cette illustration, que la durée de l'état haut du signal PWM fixe la position du bras du servomoteur à un angle déterminé.

B.3) L'asservissement du servomoteur

Pour pouvoir réaliser le maintien de la position du bras (plateau de rotation) de manière correcte, le servo utilise une électronique de commande . On peut la nommer électronique d'asservissement , car c'est elle qui va gérer la position du bras du servomoteur. Cette électronique est constituée d'une zone de comparaison qui compare la position du bras du servo au signal de commande comme le montre la figure 2.25 ci-dessous.

Le deuxième élément qui constitue cette électronique, c'est le capteur de position du bras. Ce capteur n'est autre qu'un potentiomètre couplé à l'axe du moteur. La mesure de la tension au point milieu de ce potentiomètre permet d'obtenir une tension image de l'angle d'orientation du bras. Cette position est ensuite comparée, à la consigne (le signal de commande) qui est transmise au servomoteur. Après une rapide comparaison entre la consigne et la valeur réelle de position du bras, le servomoteur (du moins son électronique de commande) va appliquer une correction si le bras n'est pas orienté à l'angle imposé par la consigne. [16]

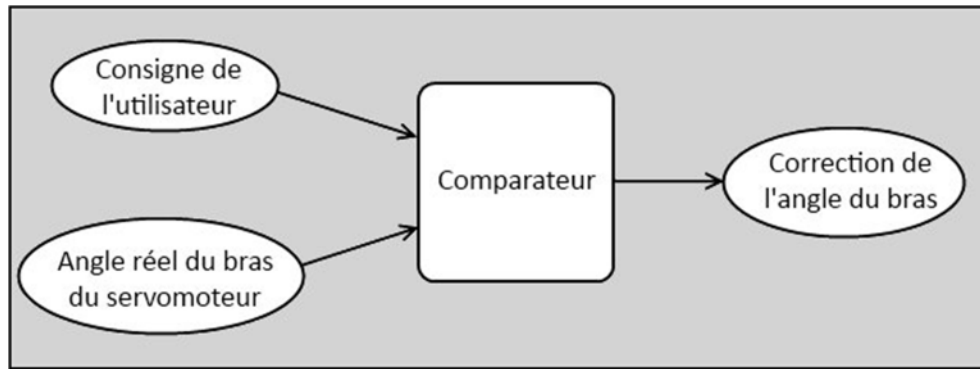


Figure 2.25 : L’asservissement du servomoteur. [16]

B .4) Le choix des servomoteurs

Pour bien choisir les différents types de servos se distingue par 3 caractéristiques:

- Vitesse de rotation par seconde ;
- Course en degré ;
- Couple exprimé en kg.cm.

Pour contrôler notre bras robotique , il est important de bien choisir angle de débattement (90° ,180°, 360°) , vitesse de rotation et surtout de bien choisir le couple des servomoteurs, ce dernier va permettre aux servomoteur de supporter le poids de la structure , afin d’effectuer des tâches et différents mouvements en toute souplesse, d’une façon répétitifs sans aucun risque.

Le couple est exprimé en kg.cm (kilogramme.centimètre) indique la relation entre la longueur du palonnier (bras du servomoteur) et la force exercée sur le bras du servomoteur (par le poids de l’objet que l’on souhaite mettre en mouvement). [16]

Le couple est calculer par la formule suivant :

$$C=F \times d \dots\dots\dots (1)$$

C : le couple du servomoteur exprimé en kg.cm.

F : la force exercée sur le bras du servomoteur (en kilos) .

d : la distance (en m) à laquelle s’exerce cette force par rapport à l’axe de rotation du servomoteur

Donc pour calculer la charge maximale que peut supporter le servomoteur on utilise la formule suivante :

$$P_{max} = \frac{C}{d} \dots\dots\dots (2)$$

P_{max} : poids maximal de charge en kilogramme (kg) ;

C : couple du servomoteur, en kilogramme centimètre (kg.cm) ;

d : distance à laquelle le poids est placé en centimètre (cm).

Par exemple dans la figure 2.26 qui montre un servomoteur avec un couple de 3,2 kg.cm permet de faire tourner sur son axe un objet de 3,2 kg sur un palonnier de 1 cm de long, ou alors faire tourner un objet d'un poids maximum de 320g sur un palonnier de 10 cm de long. [16]

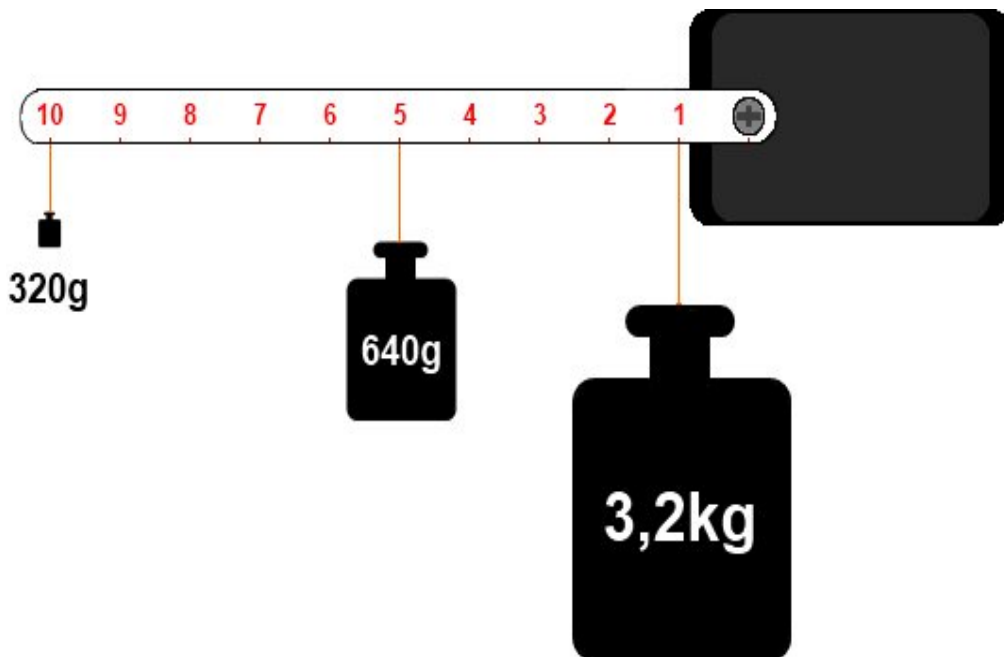


Figure 2.26 : notion de couple d’ un servomoteur . [16]

Notre bras robotique comporte 6 servomoteurs , comme le montre la figure 2.27 :

- 4 servomoteurs SpringRC SR431 : pour contrôler les mouvements de corps ;
- 2 servomoteurs SpringRC SR311 : contrôler la rotation et l’ouverture de la pince.



Figure 2.27 : Les servomoteur du bras.

B.5) Caractéristique des servomoteurs RC SR431 et RC SR311

Caractéristique	RC SR431	RC SR311
Signal de contrôle	Analogique PWM	Analogique PWM
Couple	4,8V: (12,2 kg-cm) 6,0 V: (14,5 kg-cm)	4,8V: (3,1 kg-cm) 6,0 V: (3,8 kg-cm)
Poids	62,0 g	27,0 g
Dimensions	42,0 × 20,5 × 39,5 mm	31,3 × 16,5 × 28,6 mm
La vitesse	4.8V: 0,20 sec / 60 ° 6.0V: 0.18 sec / 60 °	4.8V: 0.14 sec / 60 ° 6.0V: 0.12 sec / 60
Support de rotation	Doubles roulements	Doubles roulements
Matériel d'engrenage	Métal	Métal
Gamme de rotation	180 °	180 °
Type de connecteur	J	J

Table 2.7 : caractéristique des servomoteurs RC SR431et RC SR311. [17]

2.3.3.3 La fiche technique du bras robotique

Poids	792 g
Plage de distance de fonctionnement maximale	80 cm
Hauteur maximale	52 cm
Largeur de base	14 cm
Largeur de la pince	90 mm
Longueur de câble	40 cm
Capacité de chargement	Poids maximum à 32 cm distance de fonctionnement: 150 g

Table 2.8 : fiche technique du bras robotique. [17]

2.3.3.4 le shield de pilotage du bras

Le bras robotique est équipé d'un shield (carte de commande) pour assurer la commande de 6 servomoteurs.

Un shield (boucliers en français) est une carte supplémentaire que l'on connecte directement sur la carte arduino qui a pour but d'ajouter des composants sur la carte. Ces shield viennent généralement avec une librairie permettant de les contrôler. Dans notre cas on a utilisé un shield pour contrôler le servomoteur. Par exemple on trouve des shields Ethernet, de contrôle de moteur, lecteur de carte SD, afficheur graphique couleur, interface, GPS, etc...

Le principal avantage de ces shields est leurs simplicités d'utilisation. Il suffit de les emboîter sur la carte arduino pour les connecter, les circuits électronique et les logiciels sont déjà faits et on peut en empiler plusieurs. C'est un atout majeur pour ces cartes pour pouvoir tester facilement de nouvelles fonctionnalités. [18]

Le shield qu'on a utilisé pour contrôler le bras est représenté sur la figure 2.28 :

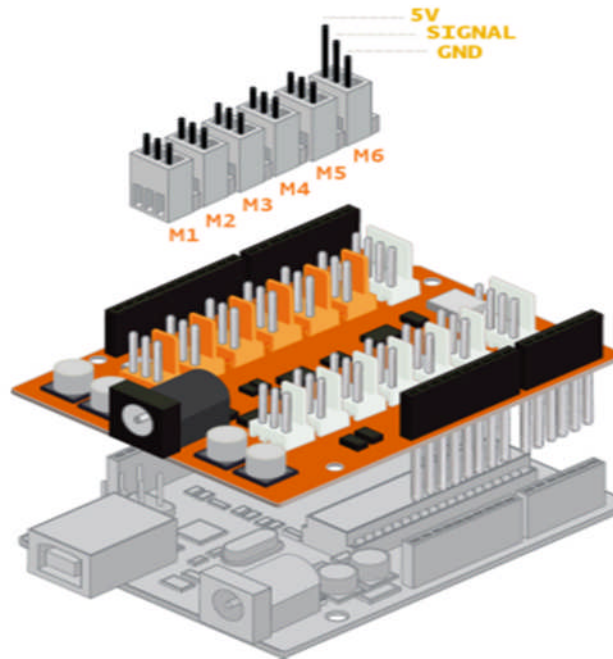


Figure 2.28 : le shield de contrôle des servomoteurs.

❖ La fiche technique du shield

- Tension de fonctionnement : 5 V ;
- Consommation électrique : 20 mW ;
- Courant maximal des connecteurs M1 à M4 : 1.1A ;
- Courant maximal des connecteurs M5 et M6 : 750mA.

Nous allons voir l'intervalle des angles autorisés pour chaque servomoteur, représenté sur la figure 2.29 : [19]

- Servomoteur M1 valeurs admises de 0 ° à 180 ° ;
- Servomoteur M2 valeurs admises de 15 ° à 165 ° ;
- Servomoteur M3 valeurs admises de 0 ° à 180 ° ;
- Servomoteur M4 valeurs admises de 0 ° à 180 ° ;
- Servomoteur M5 valeurs admises de 0 ° à 180 ° ;
- Servomoteur M6 valeurs admises de 10 ° à 73 °.

Un pas de retard de quelques millisecondes entre le mouvement de chaque servomoteur autorisé des valeurs de 10 à 30 ms

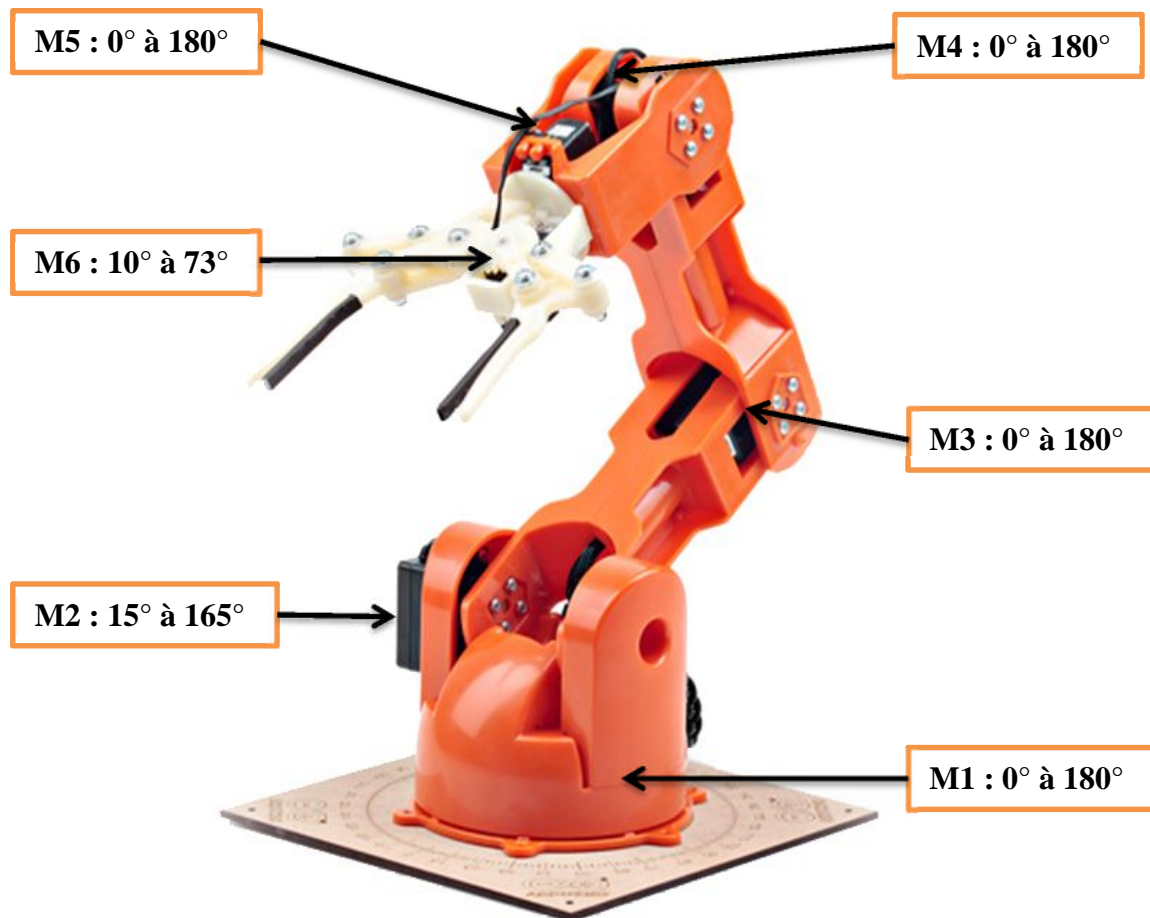


Figure 2.29 : Angle de fonctionnement autorisé pour chaque servomoteur.

2.4 Discussion

Dans ce chapitre on a présenté le principe de fonctionnement de notre système, ainsi les différents composants et matériaux utilisés pour la conception matériel.

La conception est une étape indispensable dans le but de réaliser un projet et lui assurer un bon fonctionnement. Dans le chapitre suivant nous allons voir la conception logicielle, ou on va présenter les logiciels et les programmes nécessaires au fonctionnement de notre système de tri.

Chapitre 3

Conception logiciel du
système de tri

3.1 Préambule

Dans ce chapitre nous allons voir la conception logicielle de notre système de tri, dont on va présenter en détail le système arduino, nécessaires au fonctionnement de notre système.

3.2 Système Arduino

Le système arduino donne la possibilité d'allier les performances de la programmation à celles de l'électronique. Plus précisément, pour programmer des 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. [11]

Le système arduino permet de :

- contrôler les appareils domestiques ;
- fabriquer votre propre robot ;
- faire un jeu de lumières ;
- communiquer avec l'ordinateur ;
- télécommander un appareil mobile (modélisme) ;
- etc.

Le système arduino est composé de deux choses principales : le matériel et le logiciel :

- Le matériel : Il s'agit d'une carte électronique programmable basée autour d'un microcontrôleur Atmega du fabricant Atmel. dans notre projet on a utilisé la carte arduino Mega.
- Le logiciel : permet de programmer la carte arduino. Il offre une multitude de fonctionnalités.

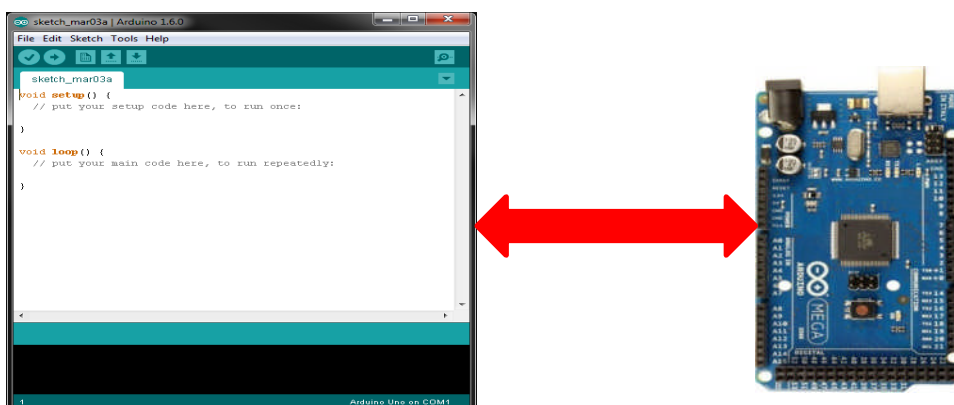


Figure 3.1 : composition du système arduino.

3.2.1 Présentation du logiciel Arduino

3.2.1.1 IDE Arduino

A) Description

L’environnement de programmation arduino appelé IDE (environnement de développement) est distribué sur le site d’arduino (compatible avec Windows, Linux et mac), A noté qu’on peut associe au compilateur arduino d’autre programme saisie avec arduino (extensions pour code blocks, visuel studio, Eclipse, etc.). Et la carte arduino utilise le langage de programmation C/C++ parce qu’il est facile à utiliser et très efficace.

Ce logiciel permet la communication direct avec la carte arduino, servant d’éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série vers la carte arduino . Il est également possible de se passer de l’interface arduino, et de compiler et uploader les programmes via l’interface en ligne de commande

La compilation consiste à analyser les instructions de code source que vous avez écrites pour les convertir en séquences de code machine, c'est-à-dire le langage binaire que peut comprendre le microcontrôleur de la carte arduino, comme le montre la figure ci-dessous :

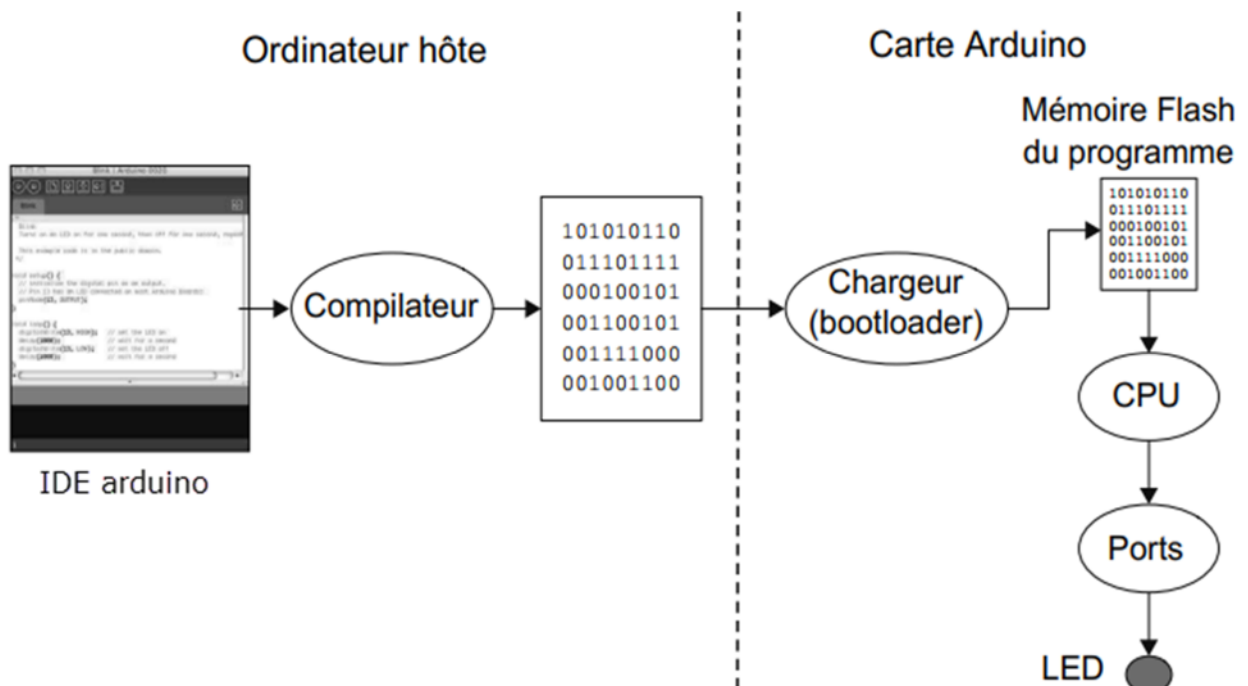


Figure 3.2 : programmation de la carte arduino. [20]

l'IDE Arduino permet :

- d'éditer un programme : des croquis (sketch en Anglais) ;
- de compiler ce programme dans le langage « machine » de l'Arduino ;
- de téléverser le programme dans la mémoire de l'Arduino ;
- de communiquer avec la carte Arduino grâce au terminal.

B) L'interface du logiciel

L'interface de l'IDE Arduino est plutôt simple (voir la figure40), il offre une interface minimale et épurée pour développer un programme pour les cartes Arduino.

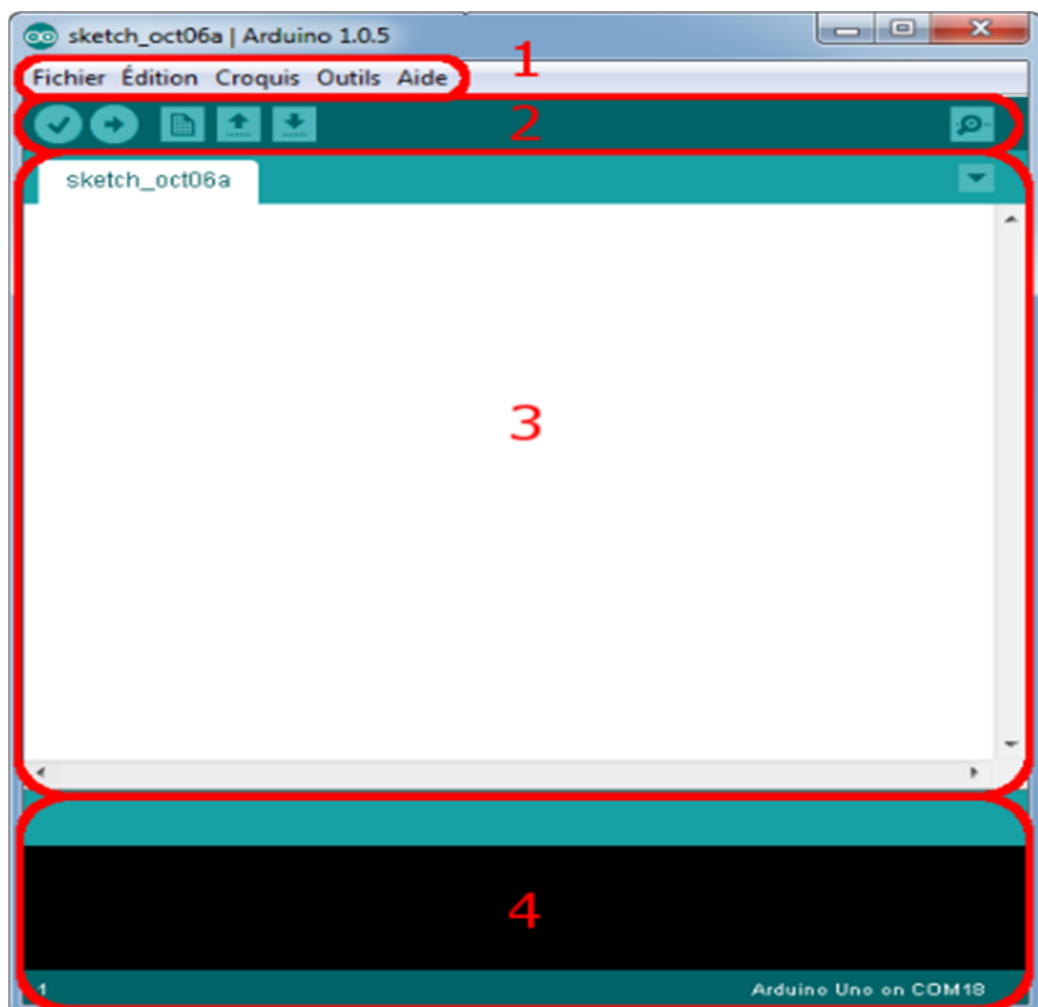


Figure 3.3 : L'interface du logiciel Arduino.

L'interface du logiciel Arduino se présente de la façon suivante : [21]

- Le cadre numéro 1 : ce sont les options de configuration du logiciel
- Le cadre numéro 2 : il contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes
- Le cadre numéro 3 : ce bloc va contenir le programme que nous allons créer
- Le cadre numéro 4 : celui-ci est important, car il va nous aider à corriger les fautes dans notre programme. C'est le débogueur.

Parmi les options de configuration du logiciel on trouve le menu File dispose d'un certain nombre de choses qui vont être très utiles, comme le montre la figure 41 suivante: [21]

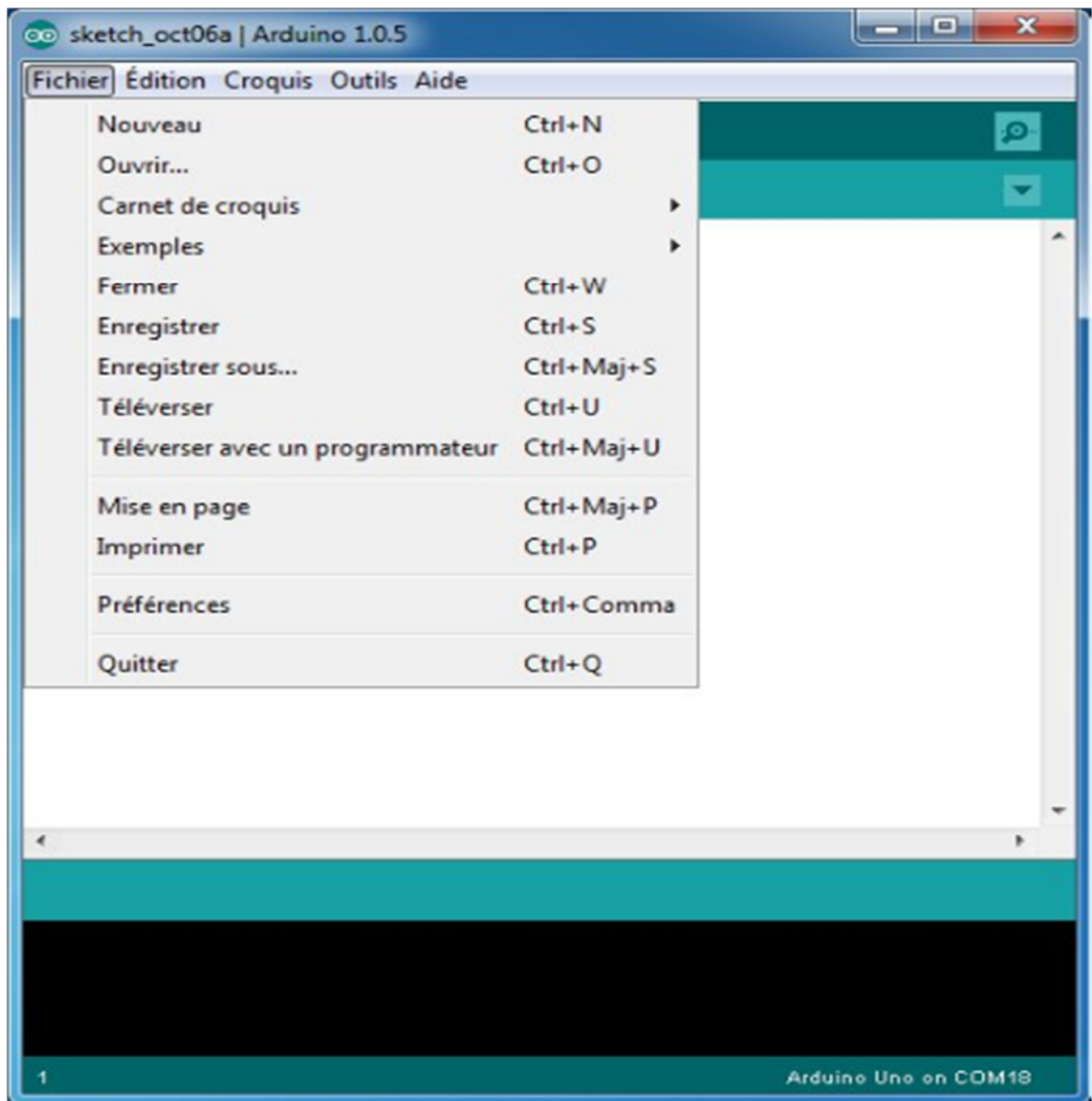


Figure 3.4 : le menu fichier de l'IDE Arduino. [21]

- 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.
- Ouvrir : avec cette commande, on peut ouvrir un programme existant.
- Enregistrer / Enregistrer sous : enregistre le document en cours / demande où enregistrer le document en cours.
- Exemples : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existant.
- les boutons pour la programmation des cartes qui facilitent l'utilisation de l'IDE qui sont représenté dans la figure (42) suivante :

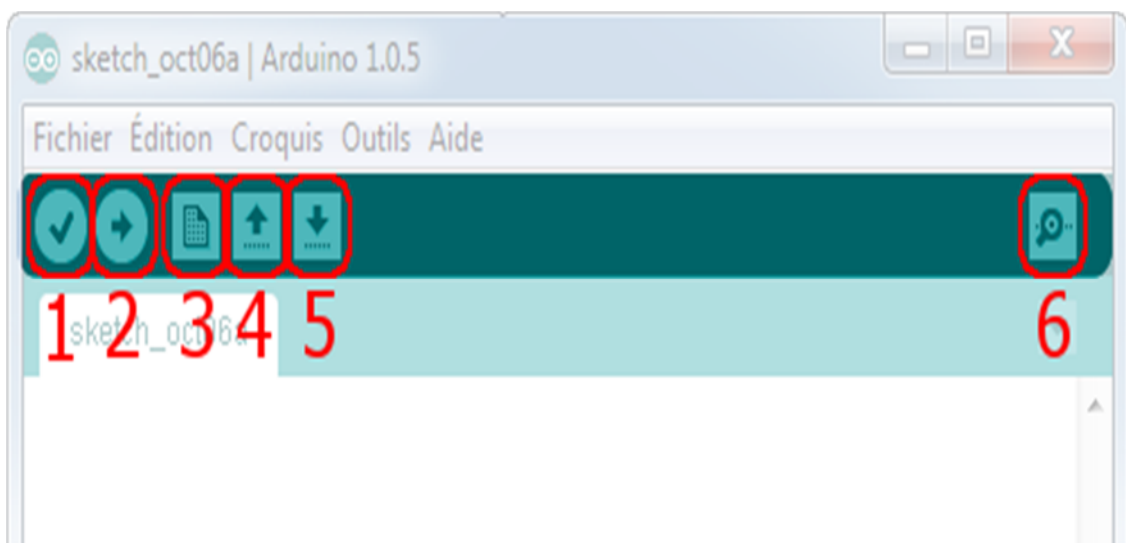


Figure 3.5 : boutons pour la programmation des cartes. [21]

- Bouton 1 : Ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 : Charger (téléverser) le programme dans la carte arduino.
- Bouton 3 : Crée un nouveau fichier.
- Bouton 4 : Ouvre un fichier.
- Bouton 5 : Enregistre le fichier.
- Bouton 6 : Ouvre le moniteur série.

C) Le langage Arduino

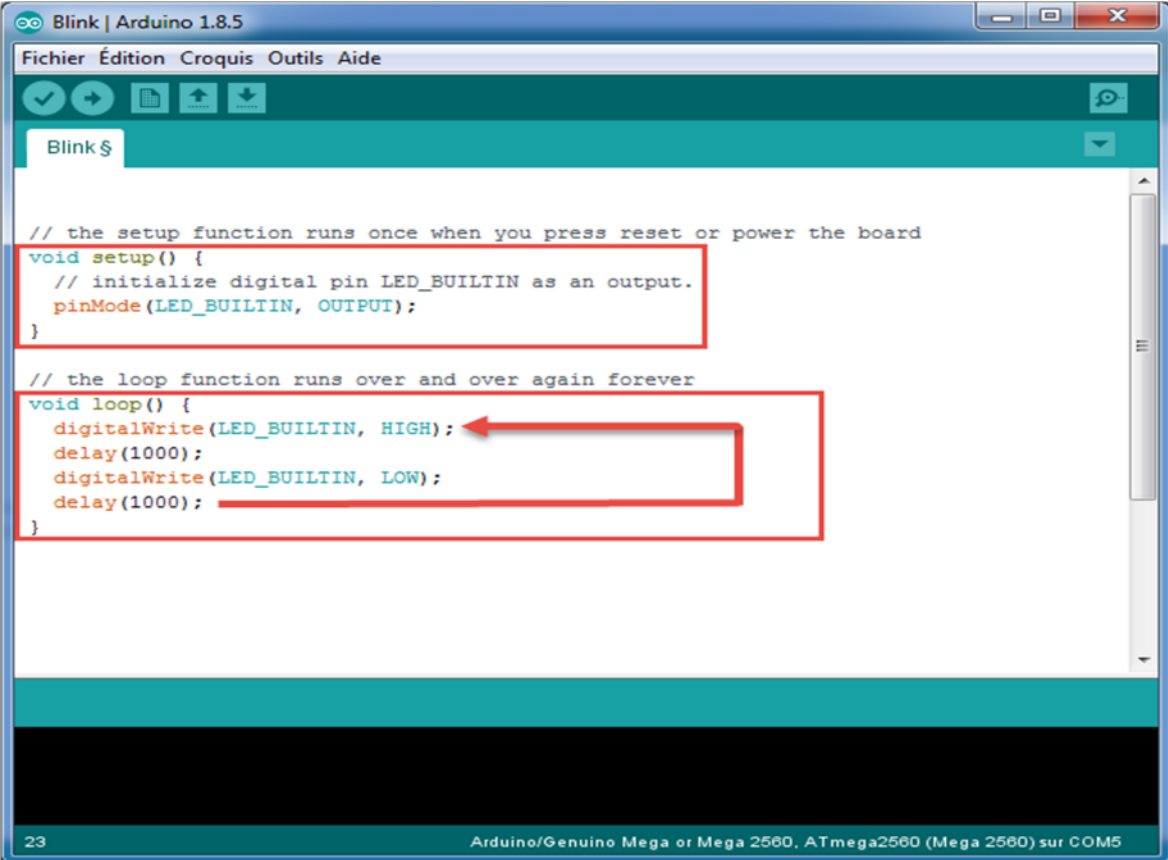
Le langage arduino est inspiré de plusieurs langages, on retrouve notamment des similarités avec le C, le C++, le Java et le Processing. le langage est basé sur le câblage impose une structure particulière typique de l'informatique embarquée, lorsque nous écrivons

un arduino croquis, nous faisons implicitement l'utilisation de la bibliothèque de câblage, qui est inclus avec l'IDE. [2]

C.1) Fonctions de langage

Pour de faire des programmes exécutables en utilisant seulement deux fonctions: `setup ()` et `loop ()`, qui sont représenté sur la figure 43 :

- **La fonction `setup`** : qui s'exécute une seule fois au démarrage, contiendra toutes les opérations nécessaires à la configuration de la carte (directions des entrées sorties, protocoles de communications, débits de communications série, etc.).
- **la fonction `loop`** : elle est exécutée en boucle après l'exécution de la fonction `setup`, elle s'exécute en boucle sans fin, permettant au programme de s'exécuter et de répondre, ce qui permet de contrôler activement la carte Arduino. Cette fonction est le noyau de la plupart des programmes.



```
Arduino IDE - Blink | Arduino 1.8.5
Fichier Édition Croquis Outils Aide
Blink $

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}

23
Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) sur COM5
```

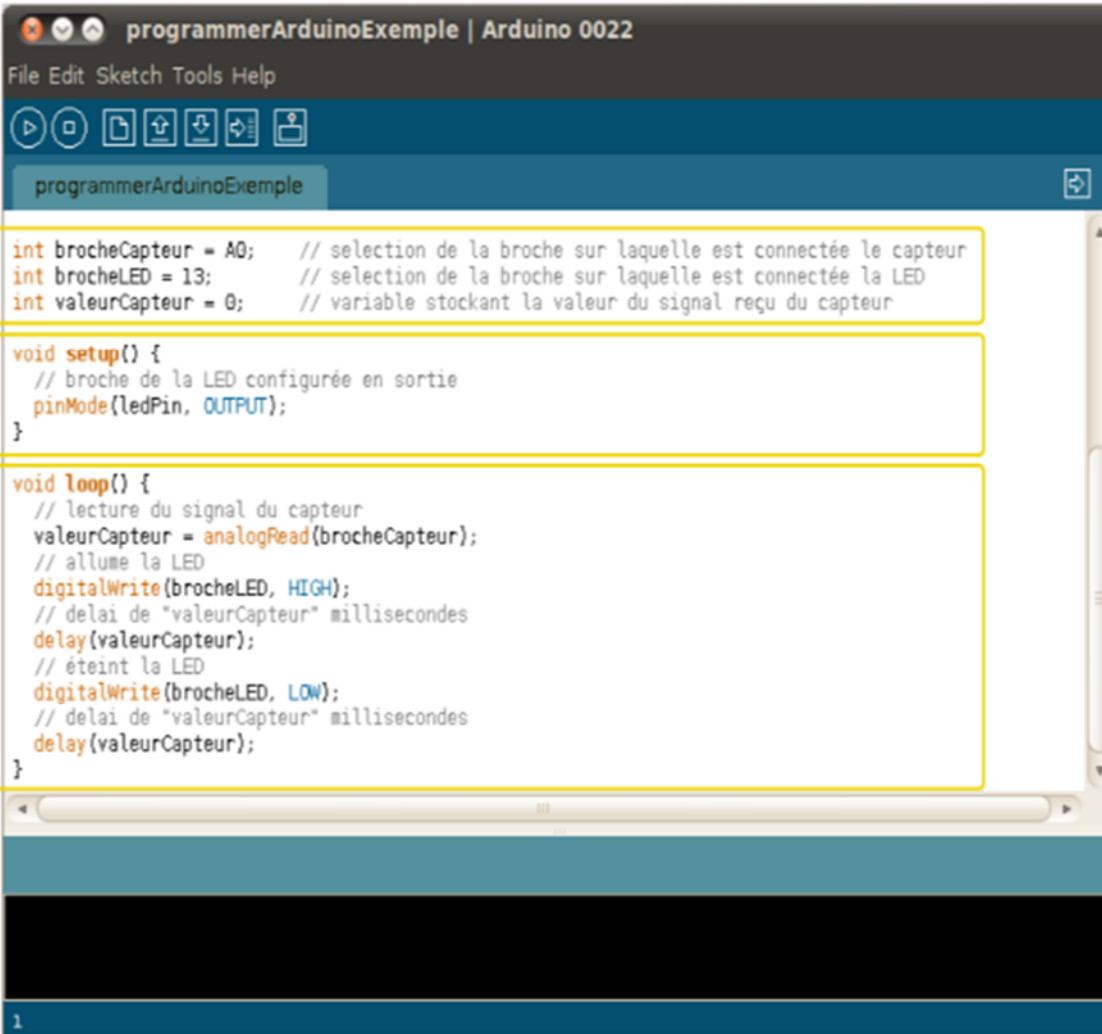
Figure 3.6 : Les fonctions Setup et loop de langage arduino.

C.2) Structure d'un programme [22]

Un programme arduino comporte trois parties qui sont représenté sur la figure ci-dessous :

1. la partie déclaration des variables (optionnelle) ;
2. la partie initialisation et configuration des entrées/sorties : la fonction `setup()` ;
3. la partie principale qui s'exécute en boucle : la fonction `loop()`.

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage arduino.



```
programmerArduinoExemple | Arduino 0022
File Edit Sketch Tools Help
programmerArduinoExemple

1 int brocheCapteur = A0; // selection de la broche sur laquelle est connectée le capteur
  int brocheLED = 13; // selection de la broche sur laquelle est connectée la LED
  int valeurCapteur = 0; // variable stockant la valeur du signal reçu du capteur

2 void setup() {
  // broche de la LED configurée en sortie
  pinMode(ledPin, OUTPUT);
}

3 void loop() {
  // lecture du signal du capteur
  valeurCapteur = analogRead(brocheCapteur);
  // allume la LED
  digitalWrite(brocheLED, HIGH);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
  // éteint la LED
  digitalWrite(brocheLED, LOW);
  // delai de "valeurCapteur" millisecondes
  delay(valeurCapteur);
}
```

Figure 3.7 : la structure d'un programme.

Lorsque le code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments : [22]

- En orange, apparaissent les mots-clés reconnus par le langage Arduino comme des fonctions existantes;
- En bleu, apparaissent les mots-clés reconnus par le langage Arduino comme des constantes ;
- En gris, apparaissent les commentaires qui ne seront pas exécutés dans le programme. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes:
 - Dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire;
 - L'on peut encadrer des commentaires sur plusieurs lignes entre « /* » et « */ ».

3.3 Organisation et définition des tâches de notre système de tri

Comme nous avons vu dans le chapitre précédent, not système a pour but de faire un tri sélectif des produits selon 3 couleur (rouge, vert, bleu). Pour réaliser le système on a utilisé :

- un capteur de couleur TCS 230 : détecter la couleur des produit ;
- une carte arduino méga : traitement des données et commandes des actionneurs ;
- bras robotisé : déplacer et stocké les produit.

La figure 3.8 montre les différentes tâches de notre système de tri :

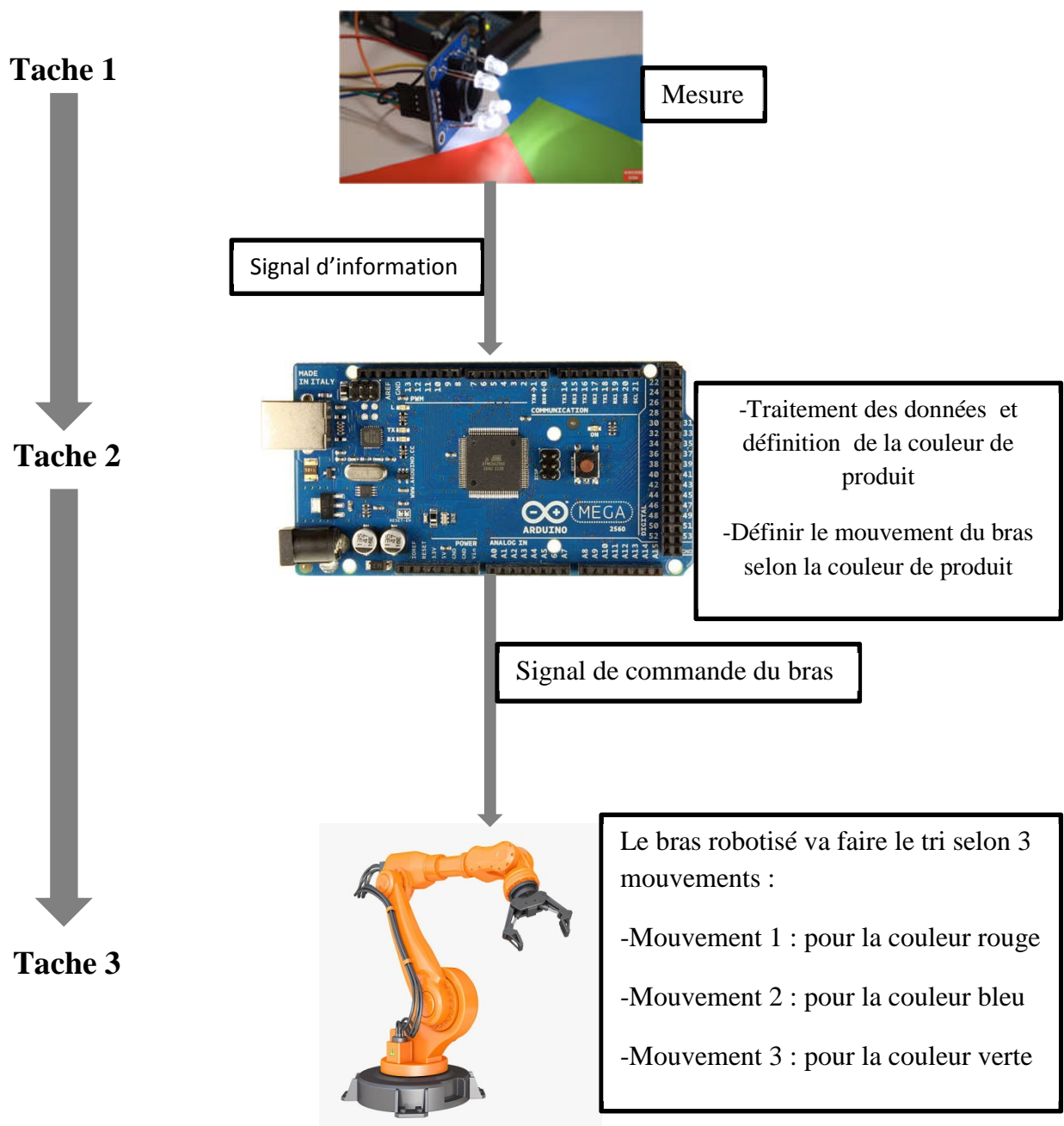


Figure 3.8 : définition des tâches du système de tri.

Les opérations de tri sont décomposées en trois tâches principales :

La première tâche:

- le capteur mesure la couleur de produit, et il transmet un signal a la carte arduino .

La deuxième tâche :

- traitement de la donnée reçue par le capteur TCS 230, et définition de la couleur de produit.
- définition du mouvement du bras robotique pour effectuer le déplacement du produit.

La troisième tâche :

- le bras robotisé déplace le produit à la zone de stockage.

3.4 Organigramme général du système

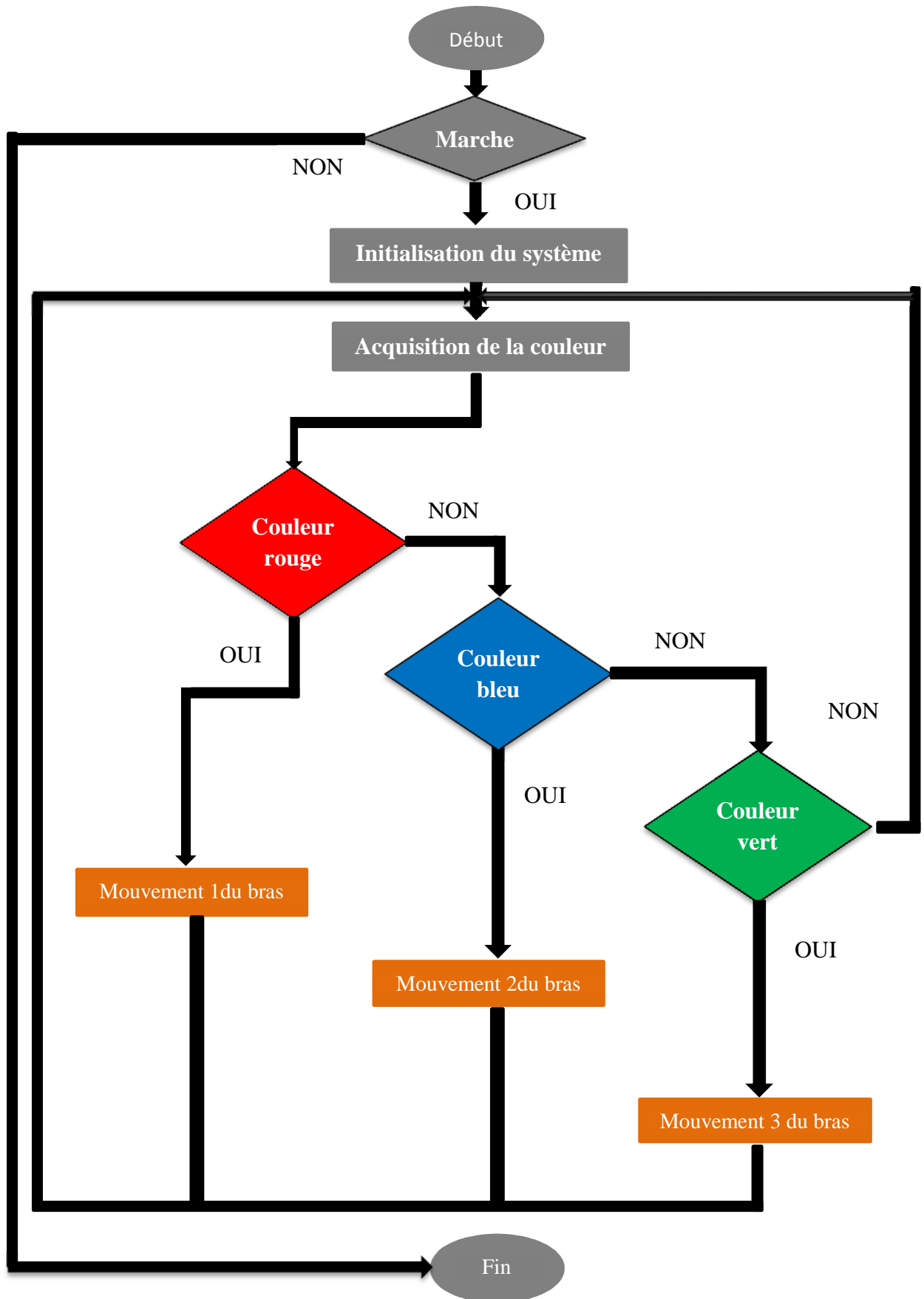


Figure 3.9 : Organigramme de fonctionnement du système.

3.5 Discussion

Dans ce chapitre, nous avons illustré les étapes suivies pendant la conception logiciel de notre système, dont on a présenté : l'environnement de développement arduino, l'organisation et définition des tâche qui compose le système de tri, organigramme de fonctionnements.

Ce chapitre nous a permet de comprendre le système arduino, et d'approfondir nos connaissance dans l'électronique embarqué.

Le chapitre suivant sera consacré à la réalisation du projet, dont on va voir les différents tests et les étapes de réalisation

Chapitre 4

Réalisation et test

4.1 Préambule

Dans ce chapitre nous allons présenter les étapes de réalisations de notre système, dont on va voir les schémas et le câblage des différents composants, ensuite on va procéder aux différents tests.

4.2 La Réalisation

La réalisation se déroule en deux parties :

- Chaque partie du système est réalisée et testée séparément ;
- Connecter les différentes parties et réaliser le système.

Le schéma global du système est représenté par la figure 4.1:

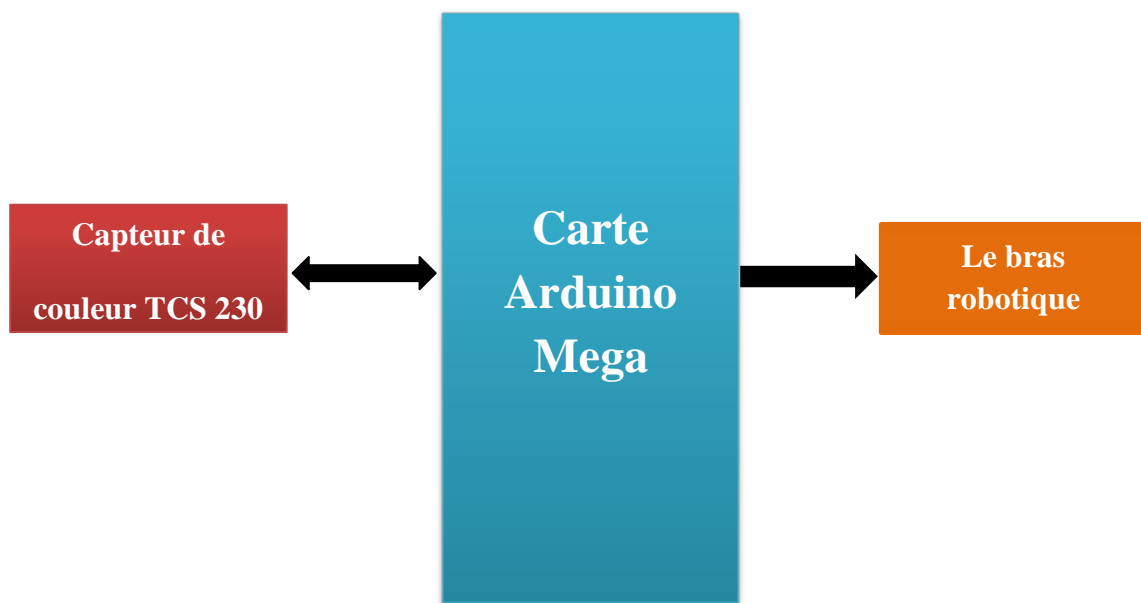


Figure 4.1 : schéma globale du système

Le système que nous allons réaliser est composé de trois partie :

- Partie mesure : grâce au capteur de couleur TCS 230 qui permet de faire la mesure de la couleur ;
- Partie acquisition et commande : la carte arduino reçoit les données du capteur, et elle commande le bras robotique.
- Partie actionneur : bras robotique fait le tri sélectif des produits selon leur couleur.

4.2.1 Le capteur de couleur TCS 230

Le rôle de ce capteur est de pouvoir détecter 3 couleurs (rouge, bleu, vert), ensuite envoyer les signaux (la mesure) à la carte arduino qu'on a programmé pour faire l'acquisition et le traitement des signaux reçu pour définir la couleur de produit.

Le schéma du branchement électrique de module TCS230 avec la carte arduino est présenté sur la figure 4.2 :

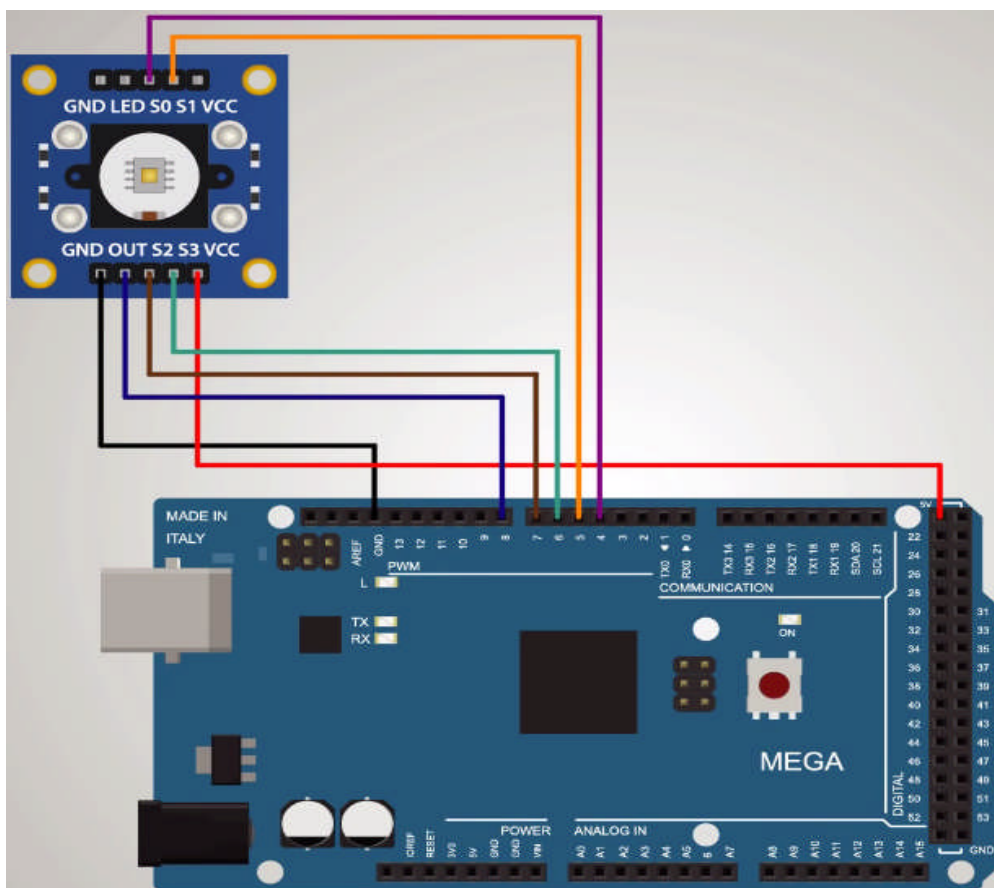


Figure 4.2 : Schéma de branchement de capteur TCS230 a la carte arduino.

4.2.2 Test de capteur TCS 230

Pour tester le module TCS230 on doit :

- Lancer le logiciel arduino et téléverser le programme de detection de couleur vers la carte arduino ;
- Ouvrir le moniteur série de logiciel arduino pour visualiser la perception des 3 couleurs (rouge, vert, bleu).

➤ **Téléverser le programme vers la carte arduino :**

La figure suivante nous montrent comment téléverser le programme de détection des couleurs vers la carte arduino, pour plus de détails voir annexe C :

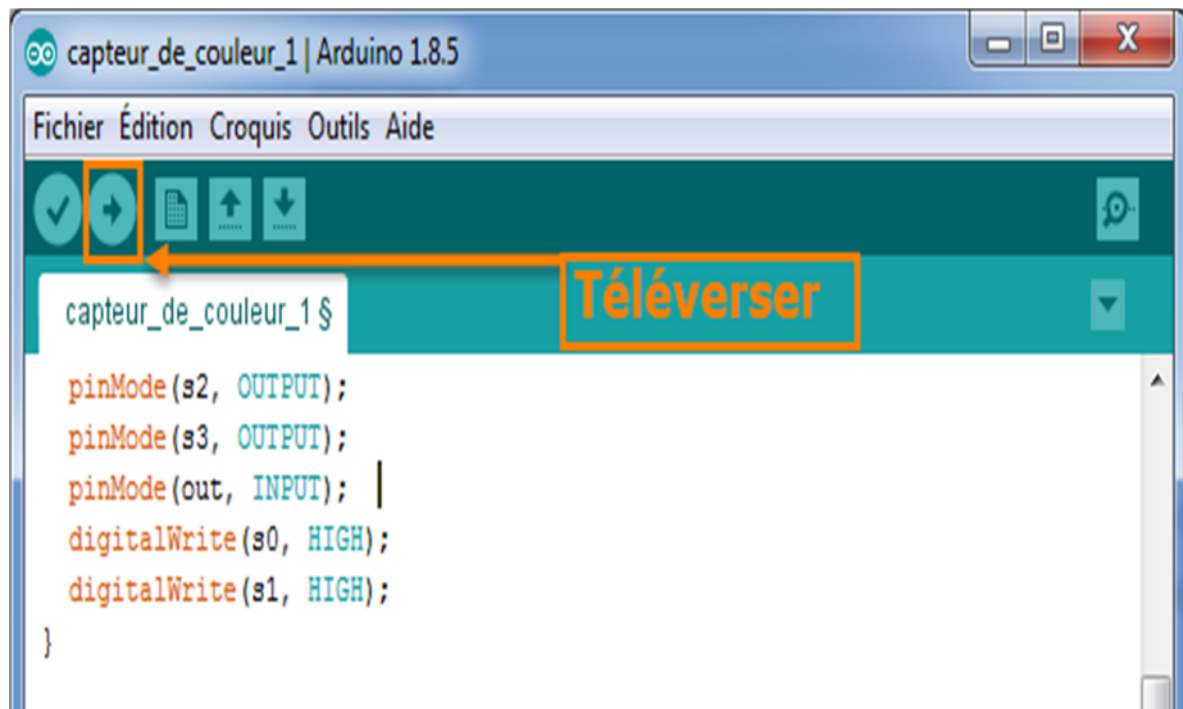


Figure 4.3 : téléverser le programme de détection des couleurs vers la carte arduino.

➤ **Test de capteur et visualisation des résultats sur moniteur série :**

Quand on place un objet d'une certaine couleur devant la matrice de photorécepteurs, les données reçues sont transmises vers la carte arduino .

Maintenant, si nous exécutons le moniteur serie , nous commencerons à obtenir trois valeurs. Les trois valeurs diffèrent en raison de la sensibilité différente de chaque type de photodiode.

La figure 4.4 représente les résultats de la détection sur le moniteur série,chaque couleur est représenté avec une lettre :

- **R : rouge ;**
- **V : vert ;**
- **B : bleu.**

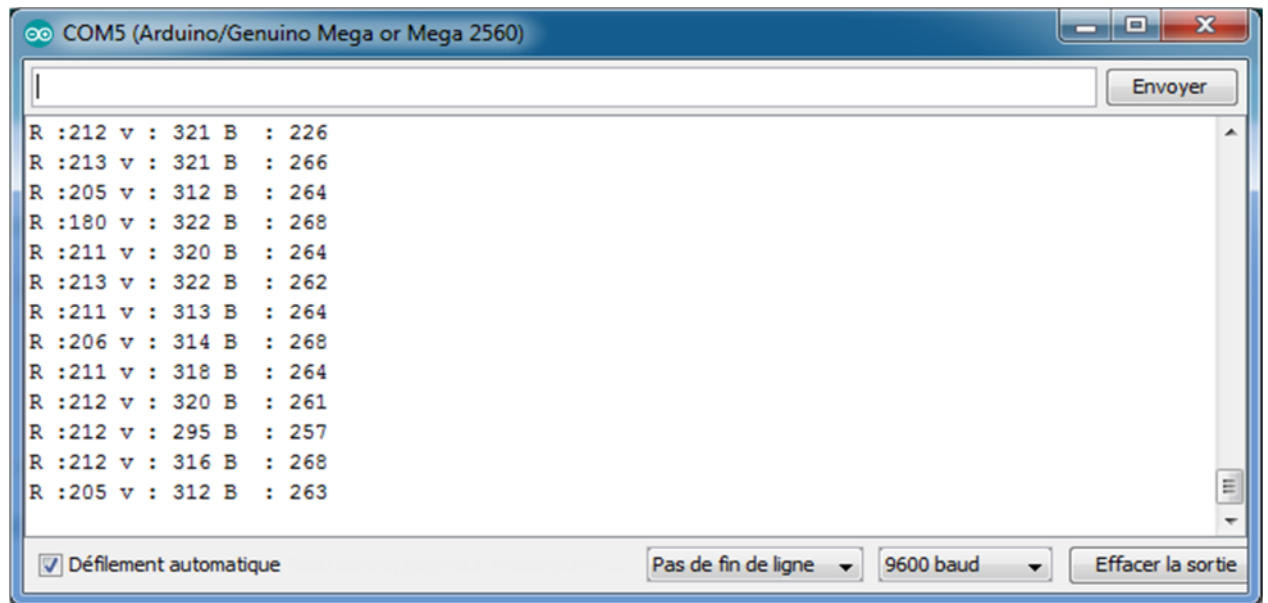


Figure 4.4 : visualisation des résultats du detection de couleur sur le moniteur série

Pour determiner la couleur de l'objet , on a fait une comparaison des valeurs ensuite on a ajouter un seuil pour la couleur rouge ,comme le montre la figure 4.5 :



Figure 4.5 : programme de définition des couleurs .

Nous allons voir comment les valeurs réagissent lorsque nous apportons différentes couleurs devant le capteur:

La couleur Rouge :

voyons maintenant sur la figure 4.6 comment les valeurs réagissent lorsque nous apportons la couleur rouge devant le capteur.

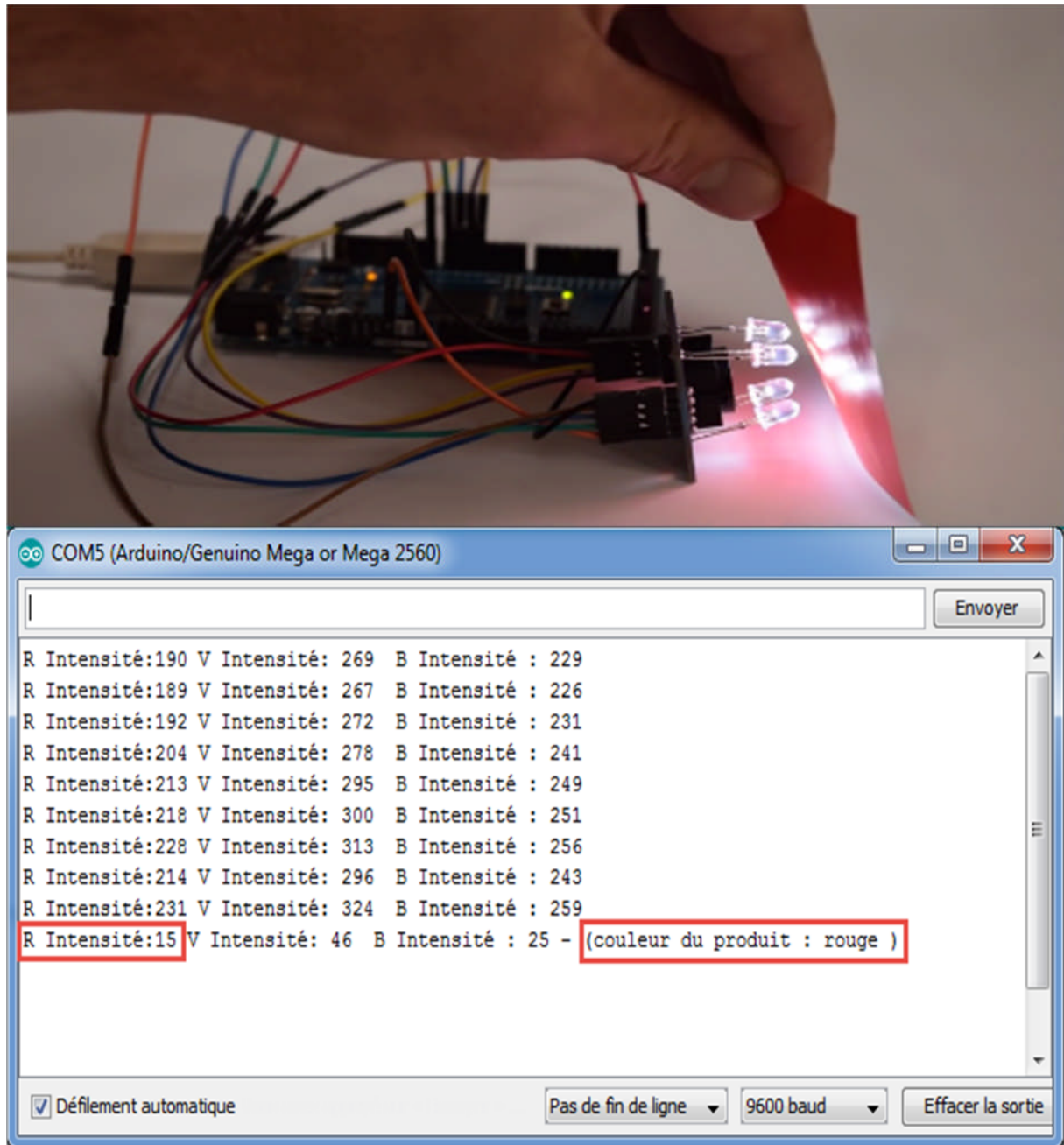


Figure 4.6 : résultat de la détection de la couleur rouge.

Remarque :

la valeur initiale de la couleur rouge diminuera, de 231 à 15. Cette valeur est comparée avec les autres valeurs, afin de définir que la couleur de l'objet est rouge .

La couleur verte :

voyons maintenant sur la figure 4.7 ,comment les valeurs réagissent lorsque nous apportons la couleur verte devant le capteur :

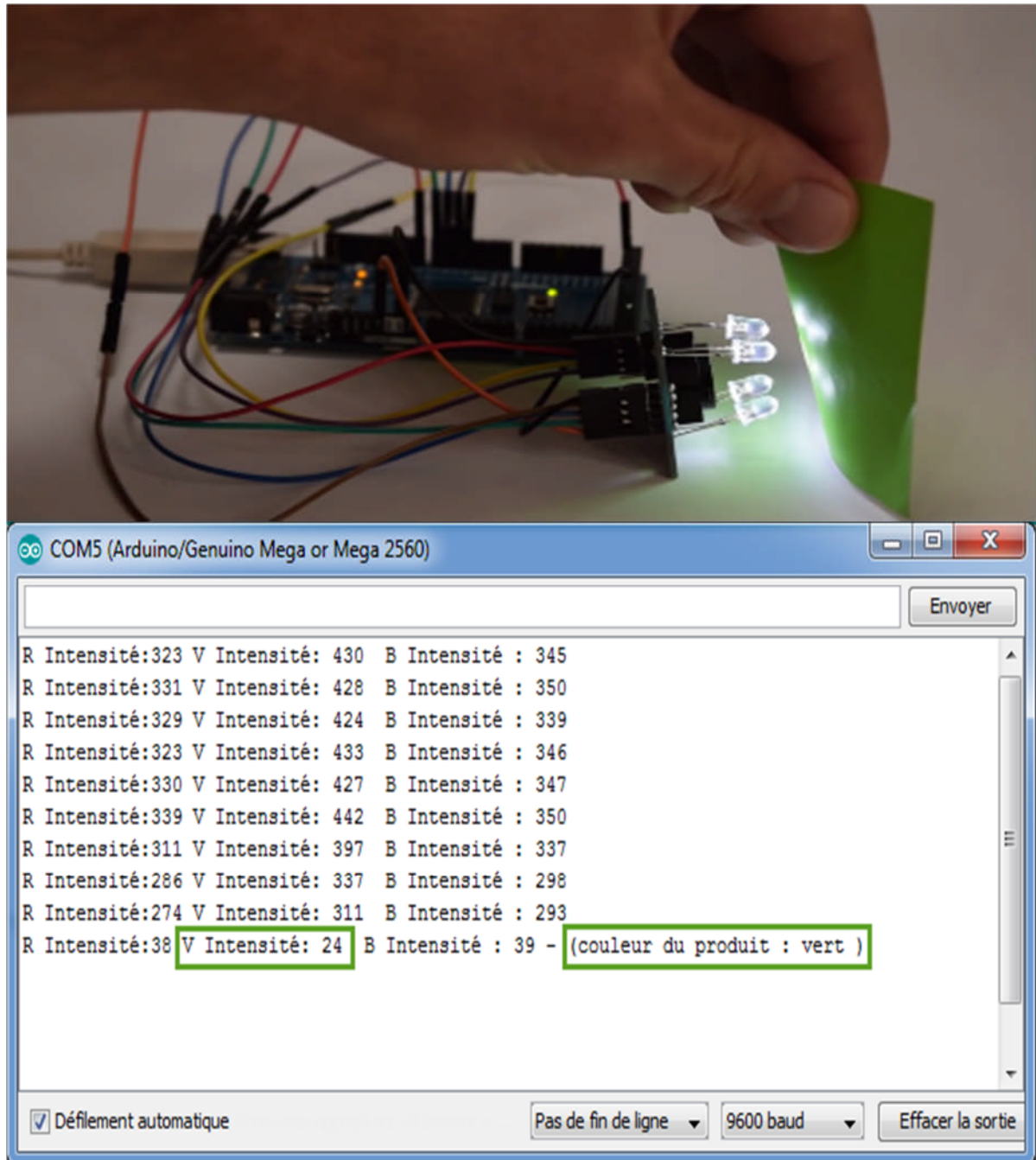


Figure 4.7 : résultat de la détection de la couleur verte.

Remarque :

la valeur initiale de la couleur verte diminuera, de 311 à 24. Cette valeurs est comparée avec les autres valeurs, afin de définir que la couleur de l'objet est verte.

La couleur bleue :

voyons maintenant sur la figure 4.8 ,comment les valeurs réagissent lorsque nous apportons la couleur bleue devant le capteur :

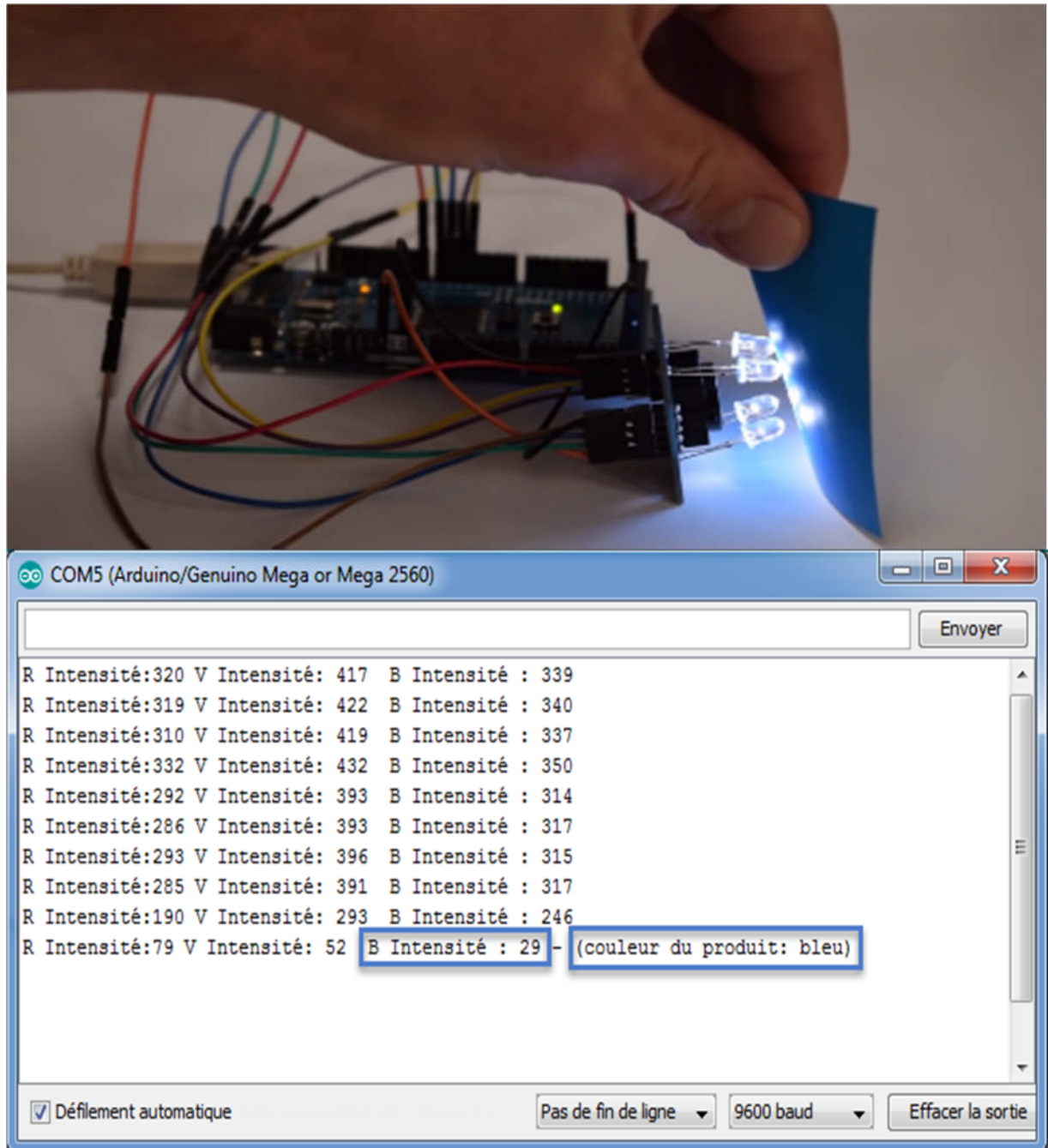


Figure 4.8 : résultat de la détection de la couleur bleue.

Remarque :

La valeur initiale de la couleur bleue diminuera de 311 à 24, cette valeurs est comparée avec les autres valeurs, afin de définir que la couleur de l'objet est bleue.

4.2.3 Le bras robotique

La réalisation d'un bras robotique nécessite une conception bien détaillée et précise pour un résultat fiable. Les pièces du bras ont été faites en plastique dur, et conçu avec l'imprimante 3D ce qui donne une structure rigide permettent de garder un bon équilibre de bras.

Le rôle du bras dans notre système est de pouvoir déplacer les produits et les stockés dans l'endroit spécifique à leurs couleur.

Les bras robotique que nous avons utilisé comporte 6 degrés de liberté (DOF = Degree Of Freedom), chaque degré de liberté représentant un mouvement que le bras peut réaliser, comme le montre la figure 4.9 suivante :

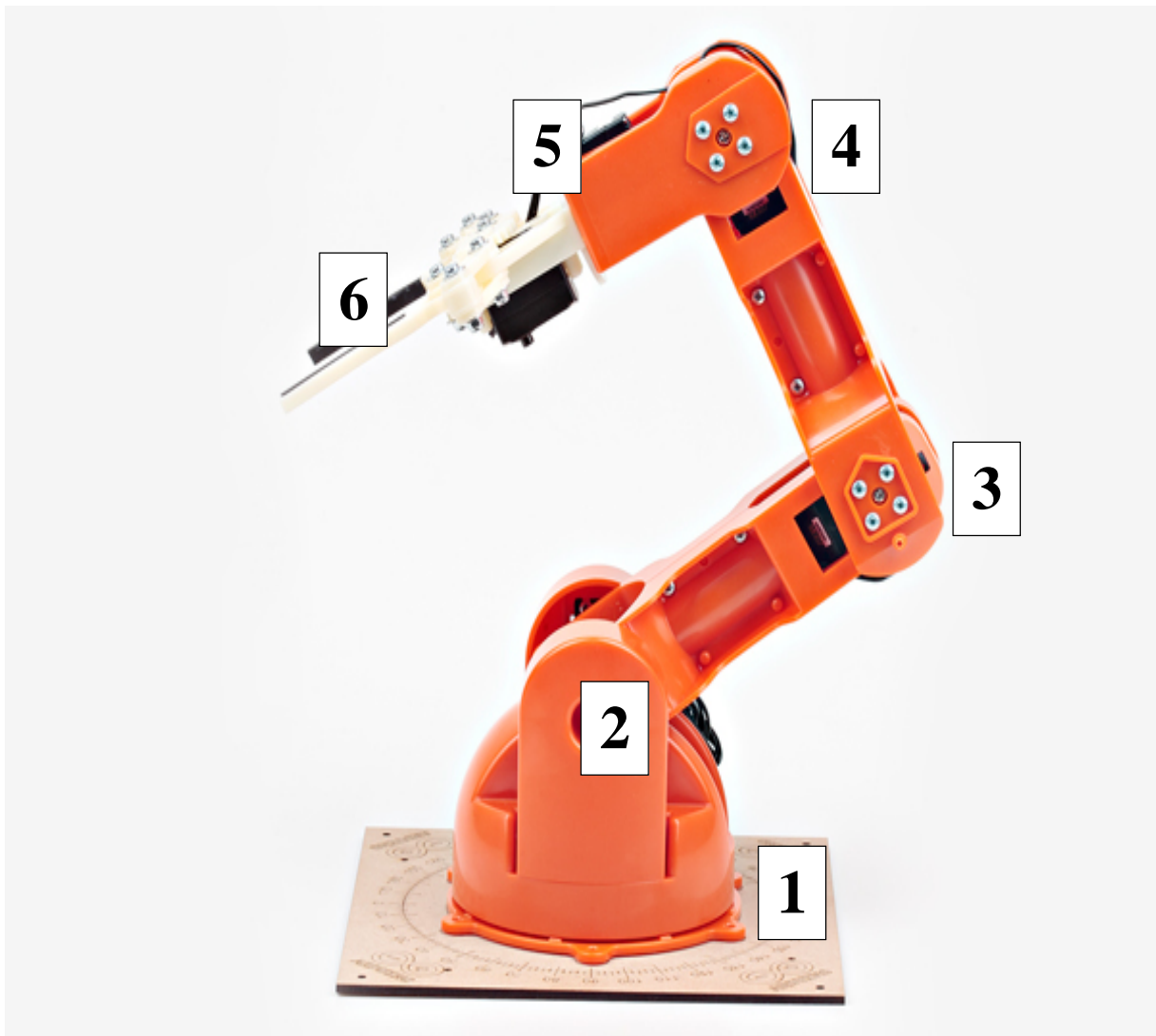


Figure 4.9 : les 6 degrés de liberté du bras.

4.2.3.1 Les différents mouvements du bras

Les mouvements du bras sont décomposés comme suite :

A) Rotation horizontal du bras

- La première rotation, celle qui effectue le pivotement du bras pour déplacer un objet n'importe où autour de lui, est faite par un axe horizontal. La rotation est engendrée par un servomoteur qui est fixé dans la base, il ne peut que l'entraîner en rotation sur 180 degrés.
- La deuxième rotation, celle qui effectue le pivotement de la pince pour déplacer un objet n'importe où autour de lui, est faite par un axe horizontal. La rotation est engendrée par un servomoteur, qui est fixé entre le bras et la pince, il ne peut que l'entraîner en rotation sur 180 degrés.

La figure 4.10 représente les différentes rotations horizontales :

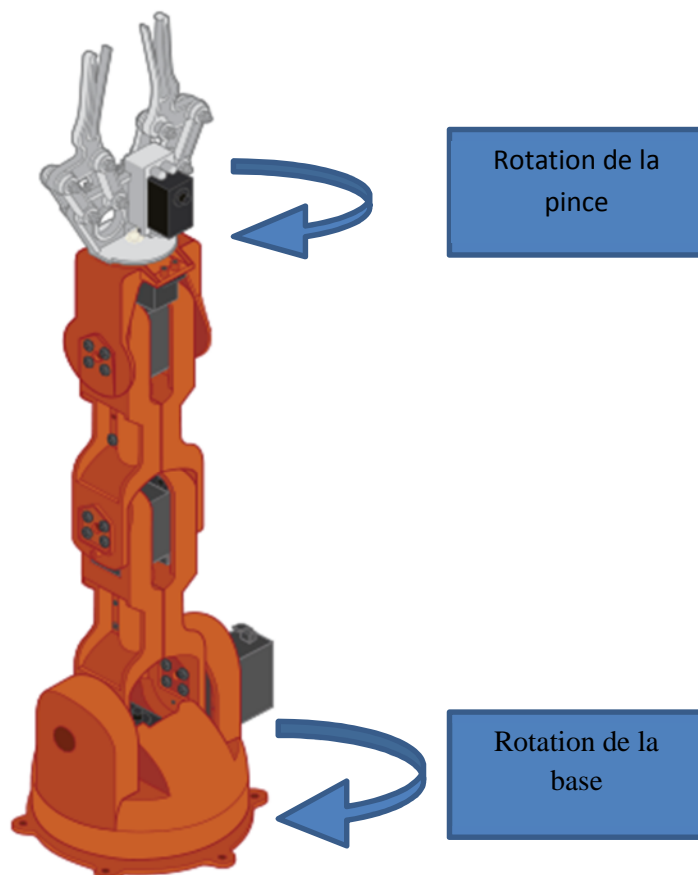


Figure 4.10 : rotations horizontales du bras.

B) Rotations verticale du bras

Nous avons trois rotations verticales du bras robotique ; deux rotations sont effectuées le pivotement du bras inférieur et supérieur et la troisième celle du poignet. La rotation du bras inférieur est engendrée par un servomoteurs qui est fixés sur le disque rotatif (la base), ainsi que celle du bras supérieur est engendrée par un seul servomoteur qui est fixé en bas du bras inférieur. Alors que le servomoteur qui pilote le poignet est fixé en haut du bras supérieur. Toutes ses rotations sont représentées sur la figure 4.11 :

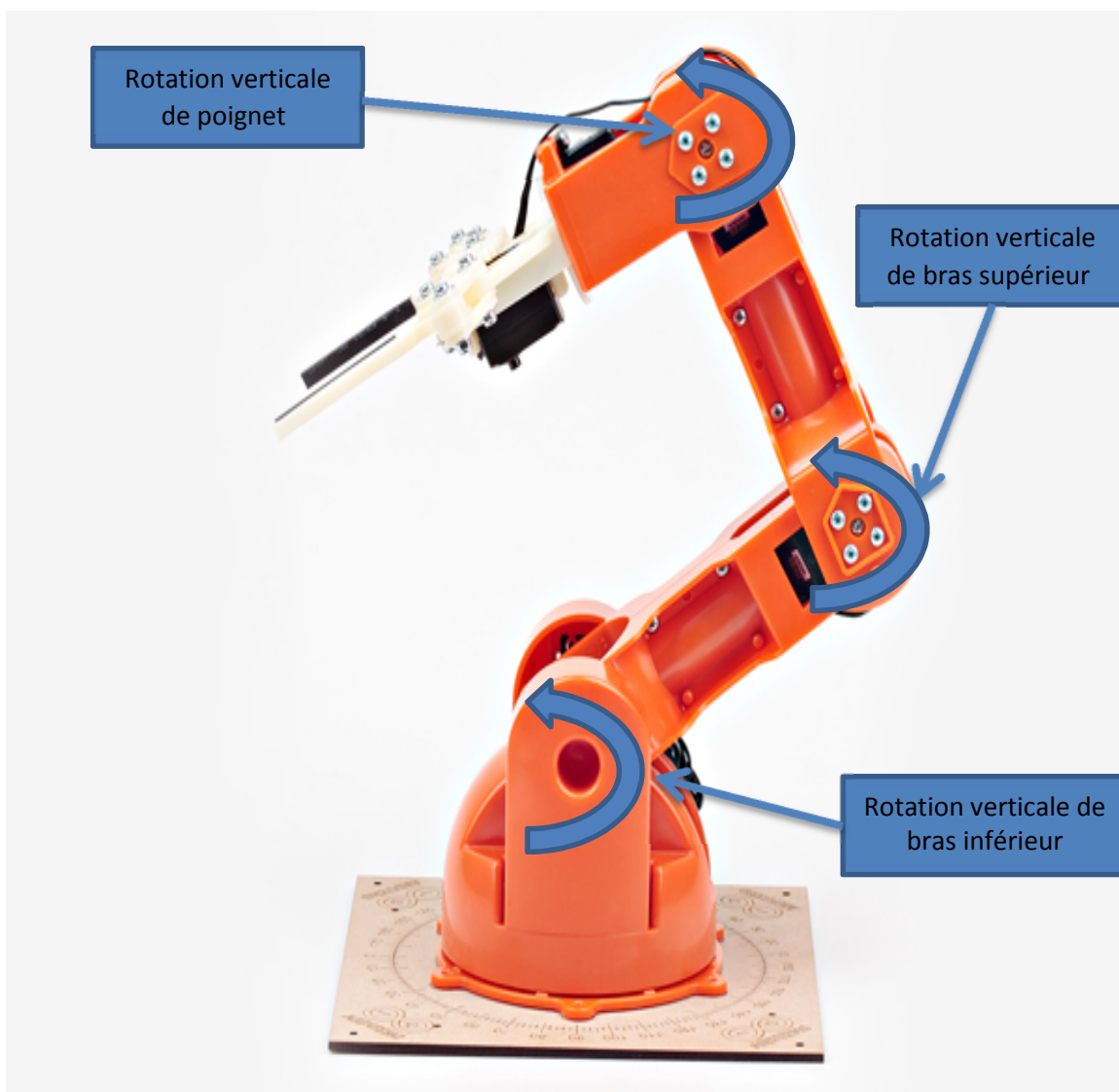


Figure 4.11 : les rotations verticales du bras.

C) L'ouverture et la fermeture de la pince

L'ouverture et la fermeture de la pince sont présentées sur la figure 4.12, elles sont effectuées par un axe droit. L'ouverture et la fermeture de la pince est engendrée par un mini servomoteur, qui est fixé au centre de la pince, il ne peut que l'entraîner sur 180 degrés.

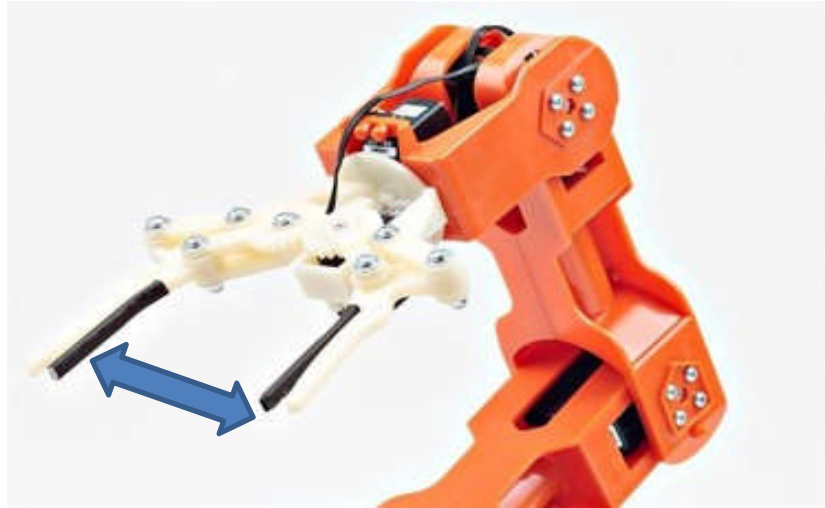


Figure 4.12 : ouverture et fermeture de la pince.

4.2.3.2 Pilotage du bras avec arduino

Les articulations du bras robotique sont articulées grâce à six servomoteurs qui permettent de reproduire les différents mouvements, comme le montre la figure suivante :

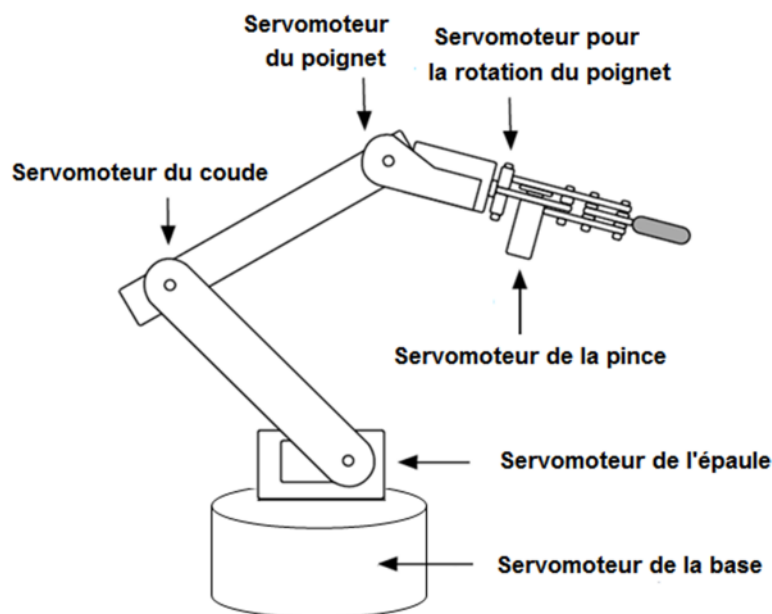


Figure 4.13 : les servomoteurs du bras robotique.

Pour piloter le bras nous allons utiliser, (voir figure 4.14) :

- Carte arduino Méga ;
- Shield pour contrôler les servomoteurs (carte de commande) ;
- Alimentation ;
- 6 servomoteurs ;
- Ordinateur ;
- Le bras robotique.

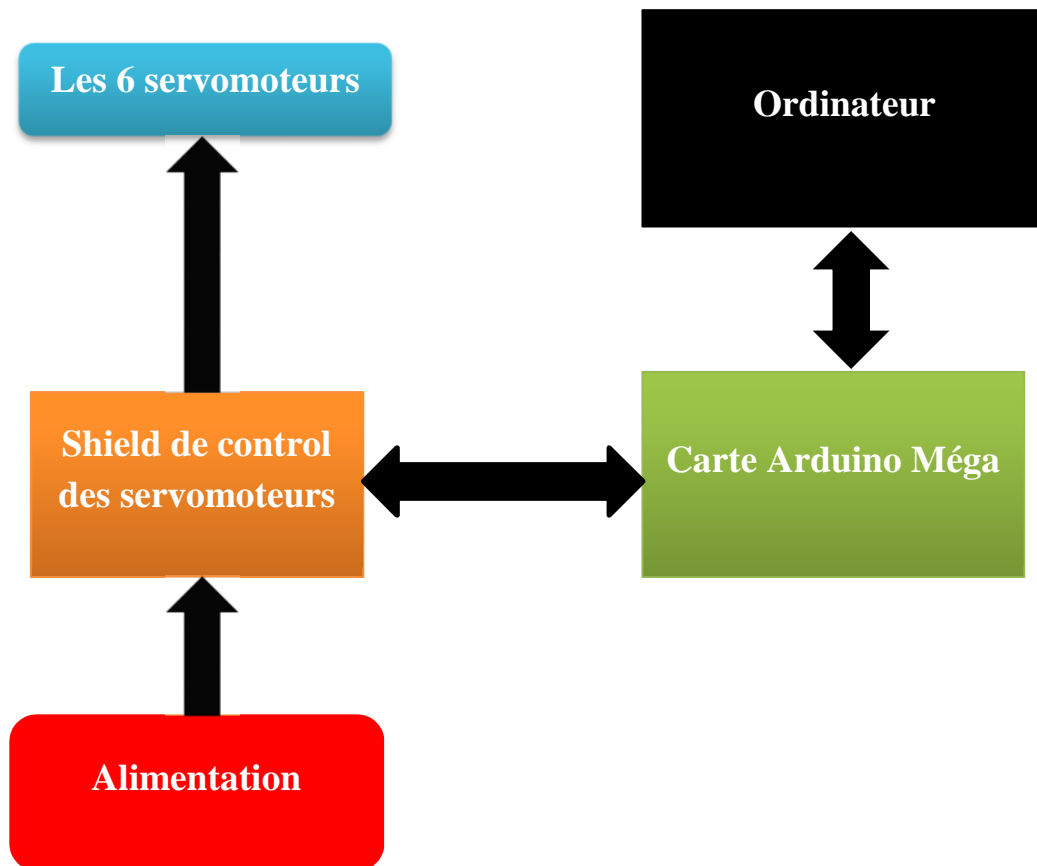


Figure 4.14 : schéma synoptique de pilotage du bras robotique.

A) Commande des servomoteurs

Pour piloter les 6 servomoteurs nous allons utiliser le shield (module), représenté sur la figure suivante :

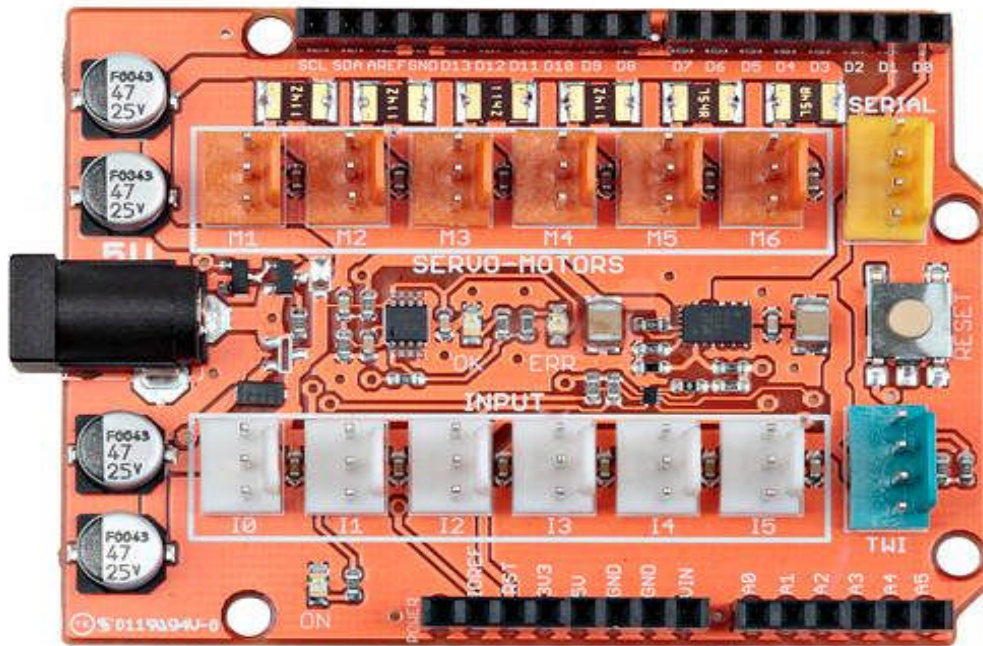


Figure 4.15: shield de commande des servomoteurs.

B) Connexion de shield avec carte arduino

Le shield est monté sur la carte arduino, comme le représente la figure 4.16 suivante :

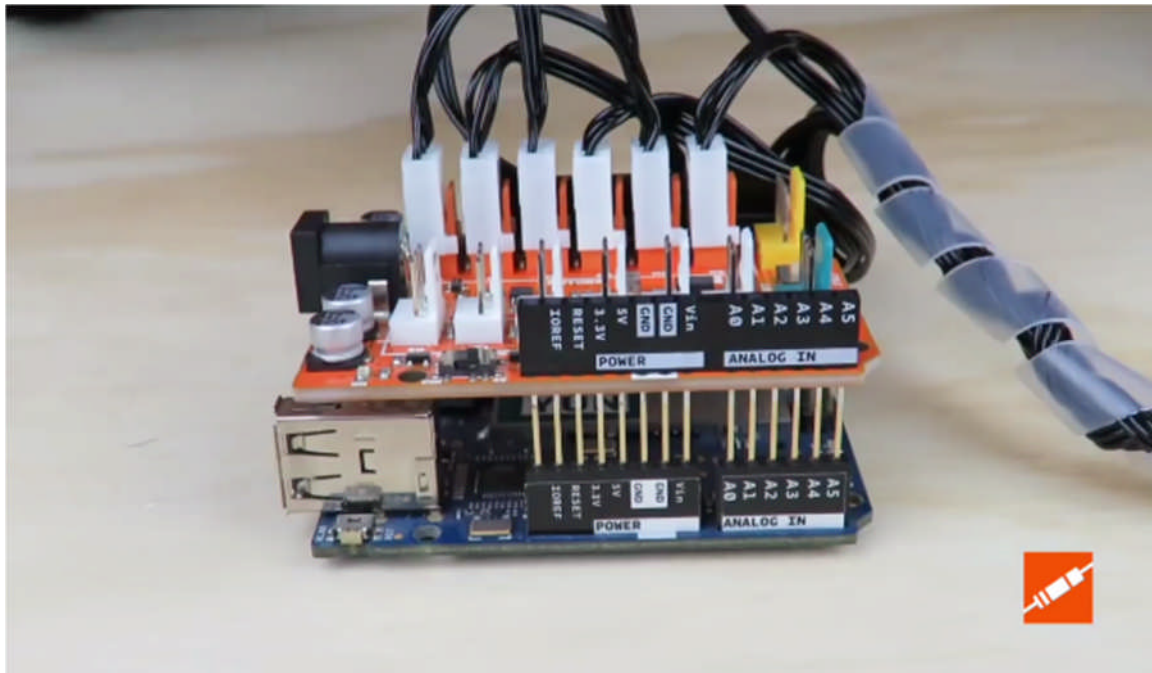


Figure 4.16 : connexion du shield avec carte arduino.

C) Câblage des servomoteurs avec le shield

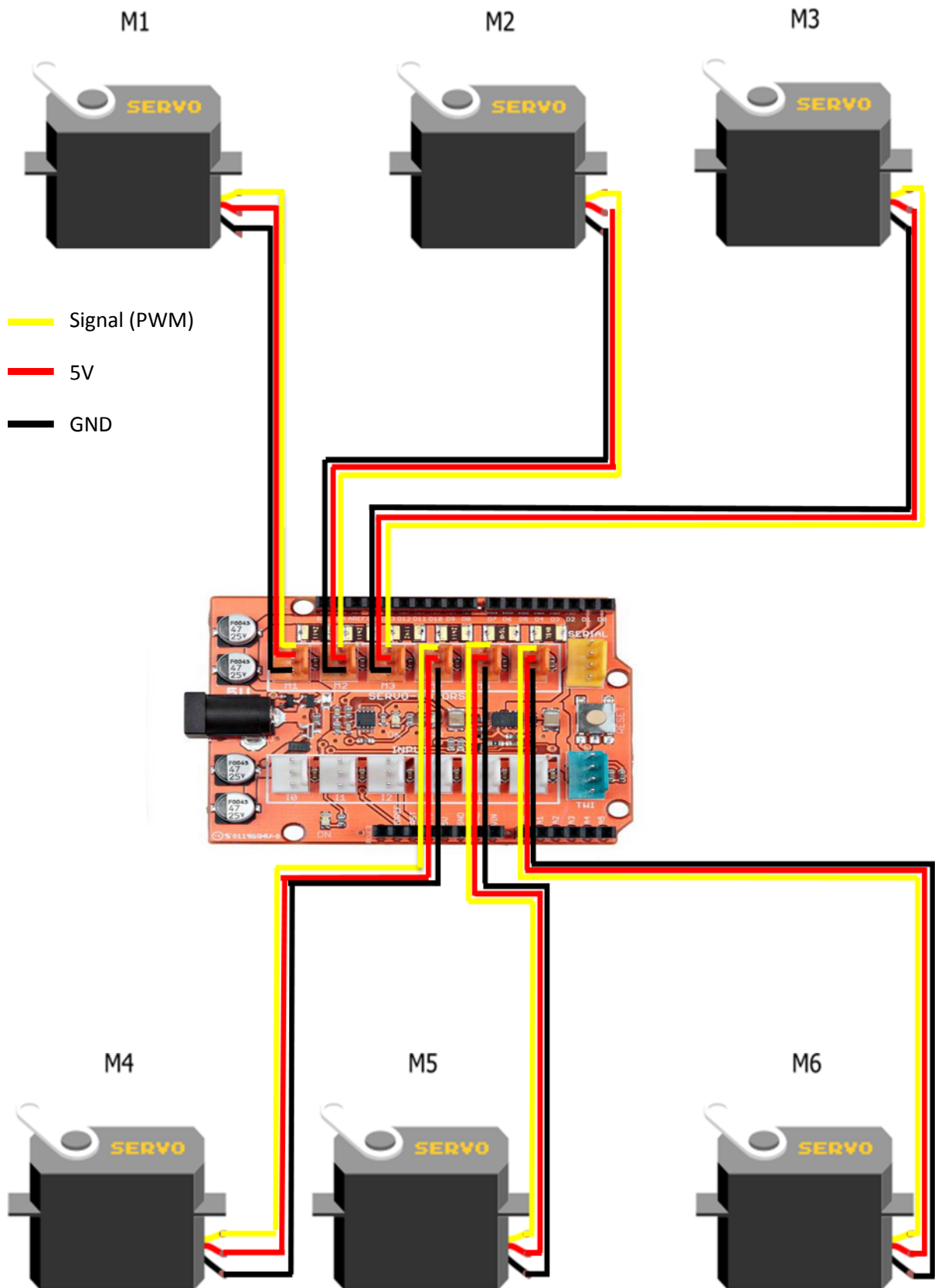


Figure 4.17 : câblage des servomoteurs avec le shield.

D) La programmation du bras

Le programme du bras est composé de trois parties :

- **Partie déclaration ;**
- **Partie initialisation ;**
- **Partie principale.**

D.1) Partie déclaration

Dans cette partie on va déclarer les variables nécessaires pour la programmation du bras. Donc on va inclure les bibliothèques `<Servo.h>` et `<Braccio.h>` des fonctionnalités utilisées pour contrôler les servomoteurs. [19]

La figure 4.18 représente la déclaration des bibliothèques `<Servo.h>` et `<Braccio.h>` :

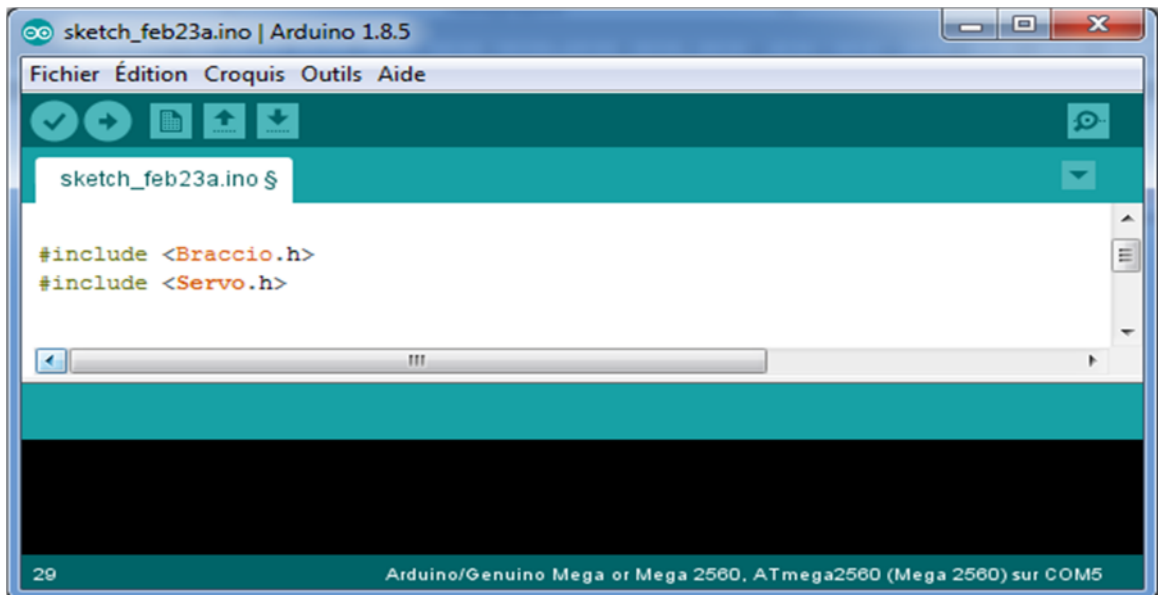


Figure 4.18 : la déclaration des bibliothèques `<Servo.h>` et `<Braccio.h>`.

D.2) Partie initialisation [19]

Les Fonctions **Braccio.begin()** d'initialisation et configuration de la position initiale pour le bras ,tous les servomoteurs seront positionnés dans la position de "sécurité" (voir figure 4.19):

- Base (M1): 90 degrés ;
- Épaule (M2): 45 degrés ;
- Coude (M3): 180 degrés ;
- Poignet vertical (M4): 180 degrés ;
- Rotation du poignet (M5): 90 degrés ;
- Pince (M6): 10 degrés.

```
sketch_feb23a.ino §  
  
void setup() {  
  //Initialization functions and set up the initial position for Braccio  
  //All the servo motors will be positioned in the "safety" position:  
  //Base (M1):90 degrees  
  //Shoulder (M2): 45 degrees  
  //Elbow (M3): 180 degrees  
  //Wrist vertical (M4): 180 degrees  
  //Wrist rotation (M5): 90 degrees  
  //gripper (M6): 10 degrees  
  Braccio.begin();  
}
```

Figure 4.19 : La Fonctions d'initialisation et configuration de la position initiale pour le bras.

D.3) la partie principale

La boucle principale (void loop) contient les instructions nécessaires pour piloter le bras, qui sont représenté sur la figure ci-dessous :

```
sketch_feb23a.ino §  
  
//Starting position  
  
//          (delai      M1 , M2 , M3 , M4 , M5 , M6);  
Braccio.ServoMovement (20,      0, 90, 180, 150, 90, 20);  
delay(1000);  
  
Braccio.ServoMovement (30,      0, 100, 175, 180, 90, 20);  
delay(2000);
```

Figure 4.20 : fonction contrôle des mouvements du bras robotique.

L'instruction <Braccio.servoMovement> est composée (voir figure 4.21) :

```
//          (delai      M1 , M2 , M3 , M4 , M5 , M6);
Braccio.ServoMovement (20,          0, 90, 180, 150, 90, 20);
```

Figure 4.21 : l'instruction braccio.servoMovement.

- un délai de quelques millisecondes entre le mouvement de chaque servo. Valeurs autorisées de 10 à 30 ms : [19]
- M1 = degrés de base. Valeurs autorisées de 0 à 180 degrés ;
- M2 = degrés d'épaule. Valeurs autorisées de 15 à 165 degrés ;
- M3 = degrés du coude. Valeurs autorisées de 0 à 180 degrés ;
- M4 = degrés verticaux du poignet. Valeurs autorisées de 0 à 180 degrés ;
- M5 = degrés de rotation du poignet. Valeurs autorisées de 0 à 180 degrés ;
- M6 = degrés de préhension. Valeurs autorisées comprises entre 10 et 73 degrés, 10: la pince est ouvert, 73: la pince est fermée.

4.2.3.3 Test du bras

Pour tester le bras on a utilisé plusieurs programmes qui lui permet d'effectuer des différents mouvements.

A) Rotation verticale

Le programme représenté sur la figure 4.22, permet au bras d'effectuer des mouvements verticaux de corps :

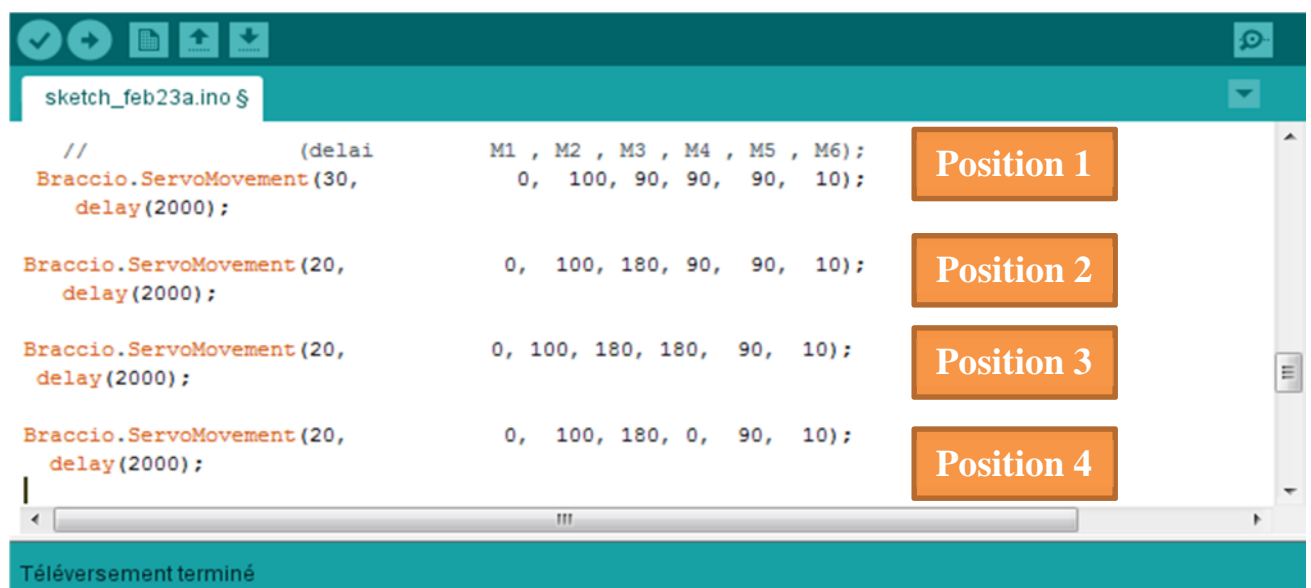


Figure 4.22 : le programme de la rotation vertical.

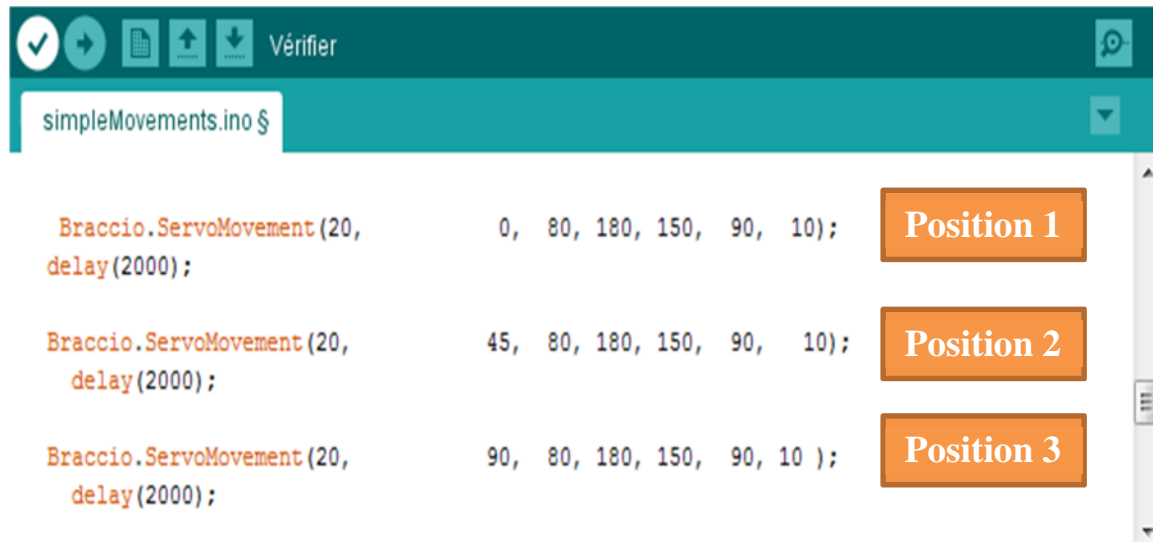
La figure 4.23 nous montre les 4 positions effectuées par le bras :



Figure 4.23 : les rotations verticales de corps.

B) Rotation horizontale de la base

Le programme représenté sur la figure 4.24 permet au bras d'effectuer des mouvements horizontaux de la base :



```
simpleMovements.ino $  
  
Braccio.ServoMovement(20, 0, 80, 180, 150, 90, 10); Position 1  
delay(2000);  
  
Braccio.ServoMovement(20, 45, 80, 180, 150, 90, 10); Position 2  
delay(2000);  
  
Braccio.ServoMovement(20, 90, 80, 180, 150, 90, 10); Position 3  
delay(2000);
```

Figure 4.24 : le programme de la rotation de la base.

La figure suivante montre les rotations horizontales de la base du bras :

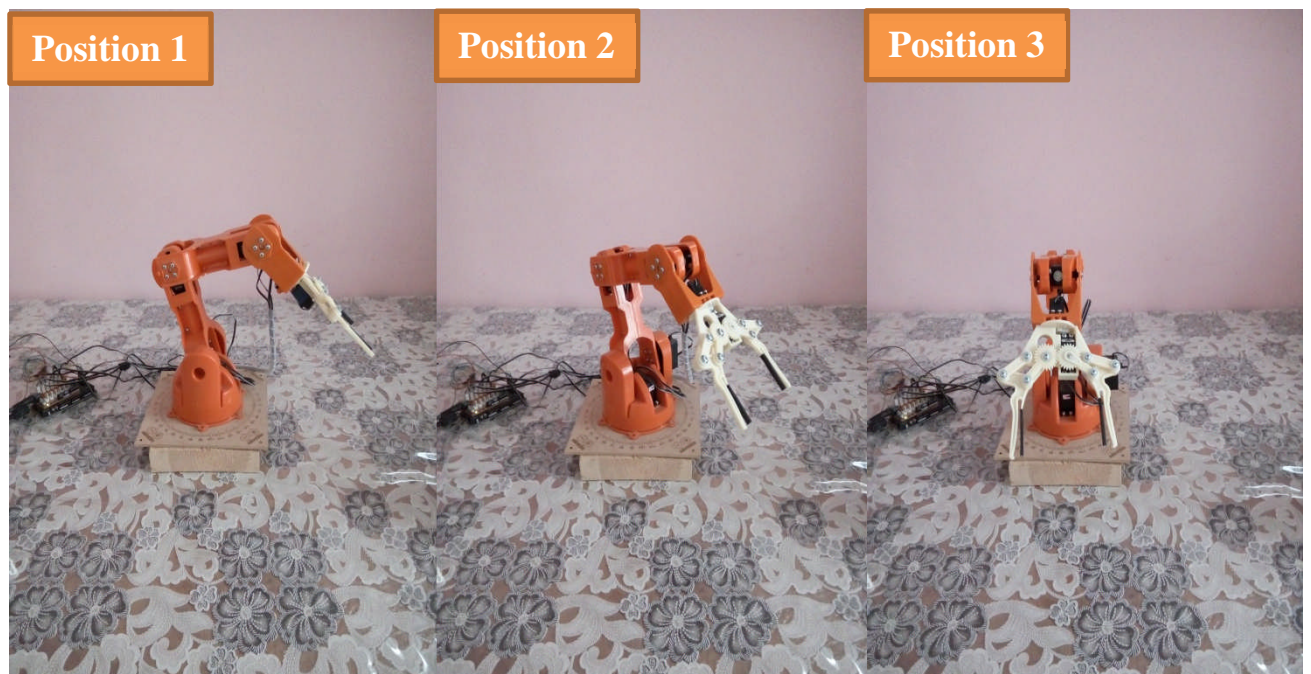


Figure 4.25 : les rotations horizontales de la base.

C) Ouverture et fermeture de la pince

Le programme représenté sur la figure 4.26 permet l'ouverture et la fermeture de la pince en utilisant le servomoteur M6, la valeur 10: la pince est ouverte, la valeur 70: la pince est fermée.



```
simpleMovements.ino $  
  
// (delai M1 , M2 , M3 , M4 , M5 , M6);  
Braccio.ServoMovement(20, 0, 80, 180, 150, 90, 10);  
delay(2000);  
  
Braccio.ServoMovement(20, 0, 80, 180, 150, 90, 70);  
delay(2000);
```

Téléversement terminé

Figure 4.26 : le programme d'ouverture et de fermeture de la pince.

La figure 4.27 montre le mouvement d'ouverture et de fermeture de la pince :

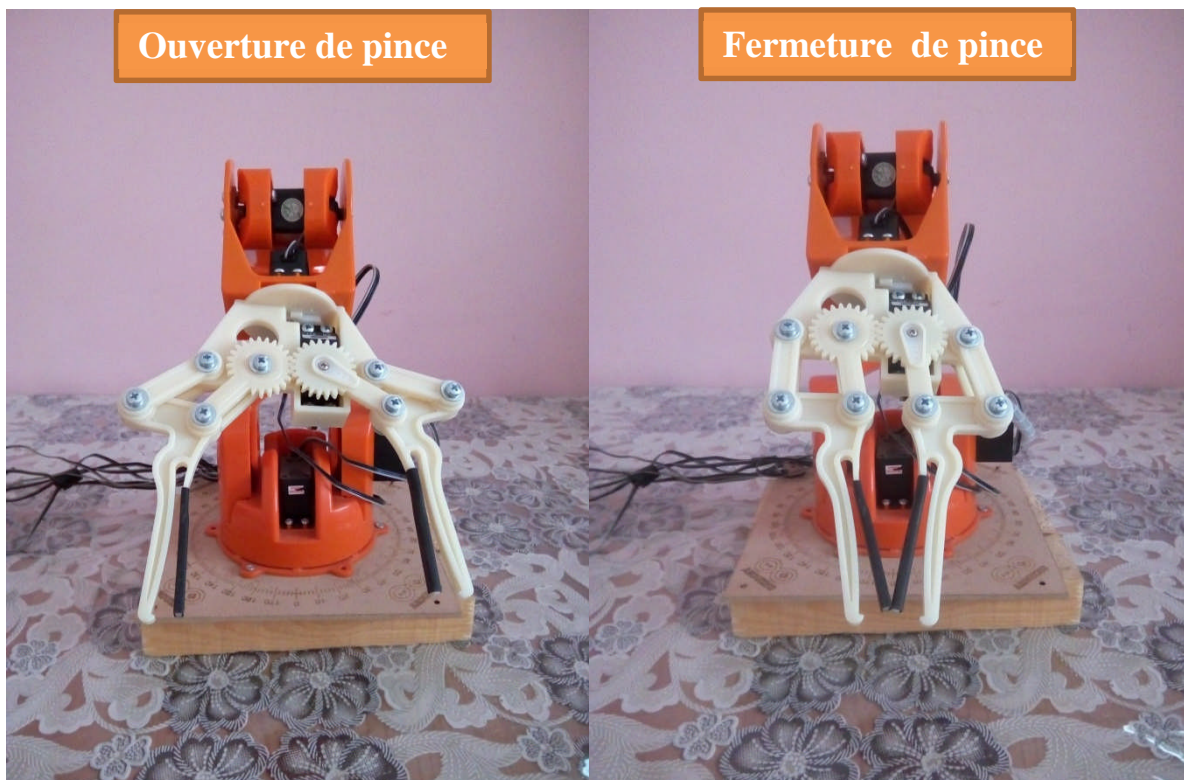


Figure 4.27 : ouverture et fermeture de la pince.

D) Déplacement d'un objet

Pour déplacer un objet, le bras doit effectuer différents mouvements pour ce positionné correctement afin d'attraper l'objet et de le déplacer d'un point a un autre. le programme représenté sur la figure 4.28, nous permet de réaliser cette tâche.

```

//deplacement d objet
      (delai      M1 , M2 , M3 , M4 , M5 , M6);
  Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 10);
  delay(2000);

  Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 10);
  delay(2000);

  Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 56 );
  delay(2000);

  Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 56);
  delay(2000);

  Braccio.ServoMovement(20,      90, 45, 100, 180, 90, 56);
  delay(2000);

  Braccio.ServoMovement(20,      90, 90, 30, 30, 90, 56);
  delay(2000);

  Braccio.ServoMovement(20,      90, 90, 30, 30, 90, 10);
  delay(2000);
    
```

Séquence 1

Séquence 2

Séquence 3

Séquence 4

Séquence

Séquence 6

Téléversement terminé

Figure 4.28 : le programme de déplacement d'un objet.

La figure 4.27 montre les mouvements effectués par le bras pour déplacer un objet :

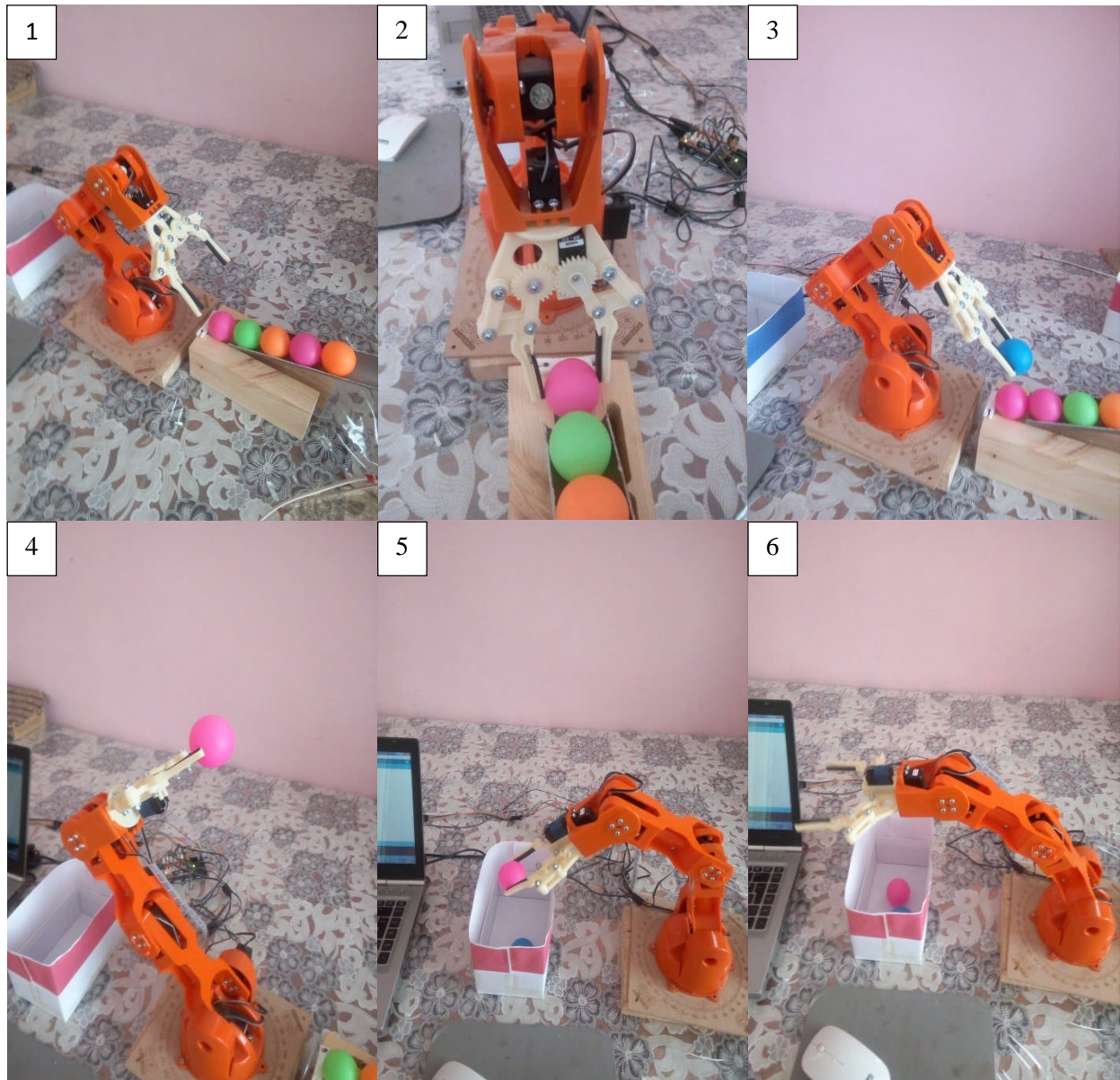


Figure 4.29 : séquence de déplacement d'un objet.

4.2.4 Réalisation et test du système de tri

La figure 4.30 présente notre système global que nous avons obtenu après l'assemblage de ses différents éléments :

- Un bras robotique ;
- Capteur de couleur TCS230 ;
- la carte arduino, et shield ;
- 3 bac de stockage pour chaque couleur.



Figure 4.30 : Vue globale du système.

Après mise en place de notre système nous avons effectué les différents tests. Dès que le système est en marche, il détecte la couleur de produit (la boule), ensuite le bras va le déplacer dans le bac spécifique à sa couleur. Pour plus de détails sur le programme de tri voir annexe C.

L'opération de tri est présentée sur la figure 4.31 :

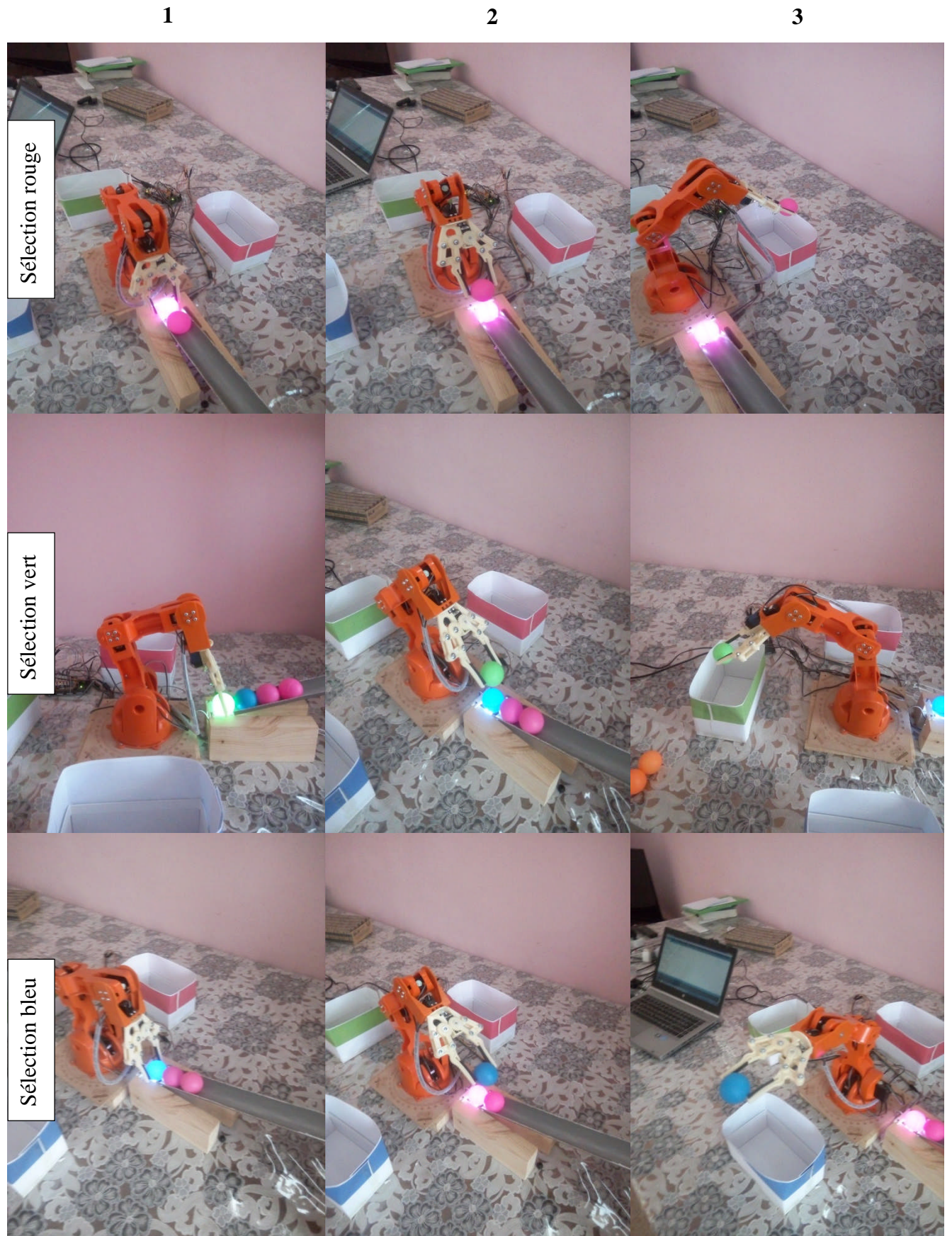


Figure 4.31 : l'opération de tri.

4.3 Discussion

Dans ce chapitre, nous avons illustré les étapes suivies pendant la réalisation de notre projet par des figures et des schémas de câblages électroniques.

Nous avons testé chaque élément constituant le système de tri :

- Test de capteur TCS230 : on a réalisé différents essais pour comprendre comment le capteur détecte les couleurs.
- Test de bras : on a présenté comment piloter le bras afin d'effectuer une tâche comme déplacement d'un objet.

Conclusion

Conclusion

Après étude, notre travail s'est soldé par une réalisation d'un système de tri basé sur une sélection par la couleur d'un produit en fin d'une chaîne de production. Cette réalisation est conçue autour d'un bras robotique piloté par une carte arduino Mega permettant la sélection selon les trois couleurs visible rouge, vert et bleu détectées grâce au capteur TCS 230.

Notre système de tri consiste à détecter la couleur d'un produit en fin de chaîne de production à l'aide du capteur TCS 230. Puis, le bras robotique exécute une séquence de mouvements qui permet de prendre l'objet et l'insérer dans un bac correspondant à l'une des couleurs rouge, vert ou bleu. La détection de la couleur et la commande du bras sont programmées à l'aide de l'interface arduino de version 1.8.5 dont le compilateur est basé sur le langage C. Le bras robotique réalise une opération de tri à une vitesse de 4 tris/min en prenant un objet de 10 g et diamètre de 4 cm.

Notre système ainsi réalisé présente un prototype avec des caractéristiques qui peuvent être adaptées à n'importe quel bras robotique industriel. En outre, sa conception logicielle est facilement modifiable de sorte à effectuer une détection d'une couleur plus complexe différente du rouge, vert ou bleu par exemple. Pour pouvoir utiliser d'autres critères de sélection il faudrait équiper le système par d'autres capteurs. C'est le cas par exemple de détection de formes où on doit utiliser une caméra avec une programmation logiciel plus complexe faisant intervenir des algorithmes de traitement d'image. Et pour améliorer la vitesse de tri, on peut envisager d'utiliser à la place des bacs des convoyeurs de sorte à minimiser la distance de déplacement du bras robotique.

Nous souhaitons que ce modeste travail serve davantage pour les promotions qui viennent et constitue une référence pour des projets futurs.

Bibliographie

Bibliographie

- [1] : https://fr.wikipedia.org/wiki/Cha%C3%A9ne_de_production
- [2] : IAMRACHE MASSINISSA, mémoire de fin d'étude : Etude et réalisation d'un bras robotique, Master ELN, département de l'électronique, UMMTO, 2016/2017.
- [3] : https://www.pobot.org/IMG/pdf/tcs230_datasheet.pdf
- [4] : <https://howtomechatronics.com/tutorials/arduino/arduino-color-sensing-tutorial-tcs230-tcs3200-color-sensor/>
- [5] : <http://www.composelec.com/photodiode.php>
- [6] : <https://lescouleurs.wordpress.com/2016/02/08/introduction/>
- [7] : http://www.elec freaks.com/wiki/index.php?title=Programmable_Color_Light-to-Frequency_Converter_Module
- [8] : E.BARTMANN, 2014.LE GRAND LIVRE D'ARDUINO, EYROLLES. 3ème édition
- [9] : SAIDI DYHIA, mémoire de fin d'étude : Conception et réalisation d'un bateau miniature piloté par androïde, Master ELN, département de l'électronique , UMMTO, 2017/2018.
- [10] : https://wiki.mdl29.net/lib/exe/fetch.php?media=robotsarduino:presentation_arduino.pdf
- [11] : <http://projet.eu.org/pedago/sin/tutos/arduino.pdf>
- [12] : <https://www.maison-et-domotique.com/72194-devriez-utiliser-cartes-arduino-vos-projets-domotique/>
- [13] : <http://www.locoduino.org/spip.php?article234>
- [14] : <https://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>
- [15] : <https://fr.wikipedia.org/wiki/Servomoteur>
- [16] : <https://eskimon.fr/tuto-arduino-602-un-moteur-qui-a-de-la-t%C3%A0te-le-servomoteur>
- [17] : <https://store.arduino.cc/tinkerkit-braccio>
- [18] : https://fablab.univ-tlse3.fr/wiki/images/9/92/Cours_arduino_v0.2.pdf

[19] : <https://docs-emea.rs-online.com/webdocs/14da/0900766b814da22f.pdf>

[20]:https://www.pearson.fr/resources/download.cfm?GCOI=27440100540280&thefile=2578_chap03.pdf

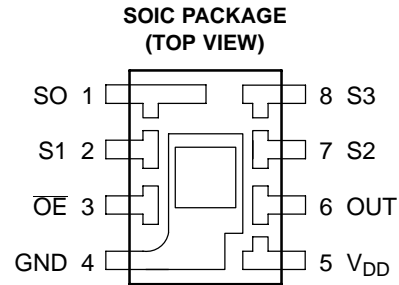
[21] : https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742_decouverte-de-larduino/3416_le-logiciel/

[22] : <https://fr.flossmanuals.net/arduino/programmer-arduino/>

[23] : E.BARTMANN, 2014.LE GRAND LIVRE D'ARDUINO, EYROLLES.

ANNEXE A

- High-Resolution Conversion of Light Intensity to Frequency
- Programmable Color and Full-Scale Output Frequency
- Communicates Directly With a Microcontroller
- Single-Supply Operation (2.7 V to 5.5 V)
- Power Down Feature
- Nonlinearity Error Typically 0.2% at 50 kHz
- Stable 200 ppm/°C Temperature Coefficient
- Low-Profile Surface-Mount Package

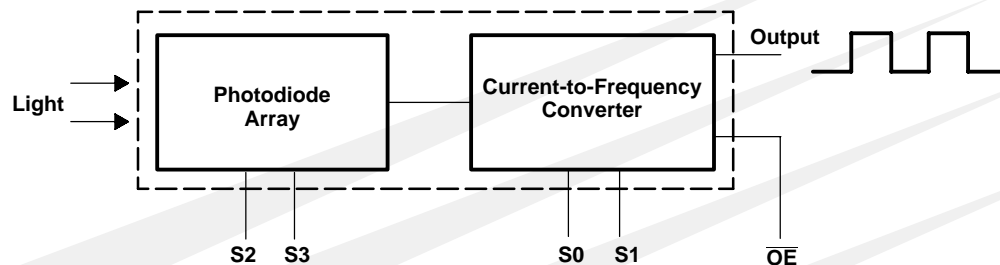


Description

The TCS230 programmable color light-to-frequency converter combines configurable silicon photodiodes and a current-to-frequency converter on single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three preset values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. Output enable (\overline{OE}) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

The light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters. The four types (colors) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All 16 photodiodes of the same color are connected in parallel and which type of photodiode the device uses during operation is pin-selectable. Photodiodes are 120 μm x 120 μm in size and are on 144- μm centers.

Functional Block Diagram



TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 - FEBRUARY 2003

Terminal Functions

TERMINAL NAME	NO.	I/O	DESCRIPTION
GND	4		Power supply ground. All voltages are referenced to GND.
\overline{OE}	3	I	Enable for f_o (active low).
OUT	6	O	Output frequency (f_o).
S0, S1	1, 2	I	Output frequency scaling selection inputs.
S2, S3	7, 8	I	Photodiode type selection inputs.
V_{DD}	5		Supply voltage

Table 1. Selectable Options

S0	S1	OUTPUT FREQUENCY SCALING (f_o)	S2	S3	PHOTODIODE TYPE
L	L	Power down	L	L	Red
L	H	2%	L	H	Blue
H	L	20%	H	L	Clear (no filter)
H	H	100%	H	H	Green

Available Options

DEVICE	T_A	PACKAGE - LEADS	PACKAGE DESIGNATOR	ORDERING NUMBER
TCS230	- 25°C to 85°C	SOIC-8	D	TCS230D

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)†

Supply voltage, V_{DD} (see Note 1)	6 V
Input voltage range, all inputs, V_I	- 0.3 V to $V_{DD} + 0.3$ V
Operating free-air temperature range, T_A	0°C to 70°C
Storage temperature range	- 25°C to 85°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to GND.

Recommended Operating Conditions

	MIN	NOM	MAX	UNIT
Supply voltage, V_{DD}	2.7	5	5.5	V
High-level input voltage, V_{IH}	$V_{DD} = 2.7$ V to 5.5 V		2	V_{DD}
Low-level input voltage, V_{IL}	$V_{DD} = 2.7$ V to 5.5 V		0	0.8
Operating free-air temperature range, T_A	0		70	°C

TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 - FEBRUARY 2003

Electrical Characteristics at $T_A = 25^\circ\text{C}$, $V_{DD} = 5\text{ V}$ (unless otherwise noted)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	$I_{OH} = -4\text{ mA}$	4	4.5		V
V_{OL}	Low-level output voltage	$I_{OL} = 4\text{ mA}$		0.25	0.40	V
I_{IH}	High-level input current				5	μA
I_{IL}	Low-level input current				5	μA
I_{DD}	Supply current	Power-on mode		2	3	mA
		Power-down mode		7	15	μA
	Full-scale frequency (See Note 2)	$S0 = H, S1 = H$	500	600		kHz
		$S0 = H, S1 = L$	100	120		kHz
		$S0 = L, S1 = H$	10	12		kHz
	Temperature coefficient of output frequency	$\lambda \leq 700\text{ nm}, -25^\circ\text{C} \leq T_A \leq 70^\circ\text{C}$		± 200		ppm/ $^\circ\text{C}$
k_{SVS}	Supply voltage sensitivity	$V_{DD} = 5\text{ V} \pm 10\%$		± 0.5		%/V

NOTE 2: Full-scale frequency is the maximum operating frequency of the device without saturation.

TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 - FEBRUARY 2003

Operating Characteristics at $V_{DD} = 5\text{ V}$, $T_A = 25^\circ\text{C}$, $S_0 = \text{H}$, $S_1 = \text{H}$ (unless otherwise noted)
(See Notes 3, 4, 5, 6, and 7).

PARAMETER	TEST CONDITIONS	CLEAR PHOTODIODE S2 = H, S3 = L			BLUE PHOTODIODE S2 = L, S3 = H			GREEN PHOTODIODE S2 = H, S3 = H			RED PHOTODIODE S2 = L, S3 = L			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
f_o Output frequency	$E_e = 45.6\ \mu\text{W}/\text{cm}^2$, $\lambda_p = 470\ \text{nm}$	16	20	24	11.2	16.4	21.6							kHz
	$E_e = 39.2\ \mu\text{W}/\text{cm}^2$, $\lambda_p = 524\ \text{nm}$	16	20	24				8	13.6	19.2				kHz
	$E_e = 32.8\ \mu\text{W}/\text{cm}^2$, $\lambda_p = 635\ \text{nm}$	16	20	24							14	19	24	kHz
	$E_e = 0$		2	12		2	12		2	12		2	12	Hz
R_e Irradiance responsivity (Note 8)	$\lambda_p = 470\ \text{nm}$		439			360			88			31		Hz/ ($\mu\text{W}/\text{cm}^2$)
	$\lambda_p = 524\ \text{nm}$		510			189			347			46		
	$\lambda_p = 565\ \text{nm}$		548			49			318			110		
	$\lambda_p = 635\ \text{nm}$		610			30			37			579		
Saturation Irradiance (Note 9)	$\lambda_p = 470\ \text{nm}$		1370			1670								$\mu\text{W}/\text{cm}^2$
	$\lambda_p = 524\ \text{nm}$		1180						1730					
	$\lambda_p = 565\ \text{nm}$		1090						1890					
	$\lambda_p = 635\ \text{nm}$		980								1040			
R_v Illuminance responsivity (Note 10)	$\lambda_p = 470\ \text{nm}$		585			480			117			41		Hz/ lx
	$\lambda_p = 524\ \text{nm}$		98			36			67			9		
	$\lambda_p = 565\ \text{nm}$		92			8			53			18		
	$\lambda_p = 635\ \text{nm}$		407			20			25			386		
Nonlinearity (Note 11)	$f_o = 0$ to 5 kHz		$\pm 0.1\%$			$\pm 0.1\%$			$\pm 0.1\%$			$\pm 0.1\%$		% F.S.
	$f_o = 0$ to 50 kHz		$\pm 0.2\%$			$\pm 0.2\%$			$\pm 0.2\%$			$\pm 0.2\%$		% F.S.
	$f_o = 0$ to 500 kHz		$\pm 0.5\%$			$\pm 0.5\%$			$\pm 0.5\%$			$\pm 0.5\%$		% F.S.
Recovery from power down			100			100			100			100		μs
Response time to output enable (OE)			100			100			100			100		ns

- NOTES: 3. Optical measurements are made using small-angle incident radiation from a light-emitting diode (LED) optical source.
4. The 470 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 470\ \text{nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 35\ \text{nm}$, and luminous efficacy = 75 lm/W.
5. The 524 nm input irradiance is supplied by an InGaN light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 524\ \text{nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 47\ \text{nm}$, and luminous efficacy = 520 lm/W.
6. The 565 nm input irradiance is supplied by a GaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 565\ \text{nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 28\ \text{nm}$, and luminous efficacy = 595 lm/W.
7. The 635 nm input irradiance is supplied by a AlInGaP light-emitting diode with the following characteristics: peak wavelength $\lambda_p = 635\ \text{nm}$, spectral halfwidth $\Delta\lambda_{1/2} = 17\ \text{nm}$, and luminous efficacy = 150 lm/W.
8. Irradiance responsivity R_e is characterized over the range from zero to 5 kHz.
9. Saturation irradiance = (full-scale frequency)/(irradiance responsivity).
10. Illuminance responsivity R_v is calculated from the irradiance responsivity by using the LED luminous efficacy values stated in notes 4, 5, and 6 and using $1\ \text{lx} = 1\ \text{lm}/\text{m}^2$.
11. Nonlinearity is defined as the deviation of f_o from a straight line between zero and full scale, expressed as a percent of full scale.



TYPICAL CHARACTERISTICS

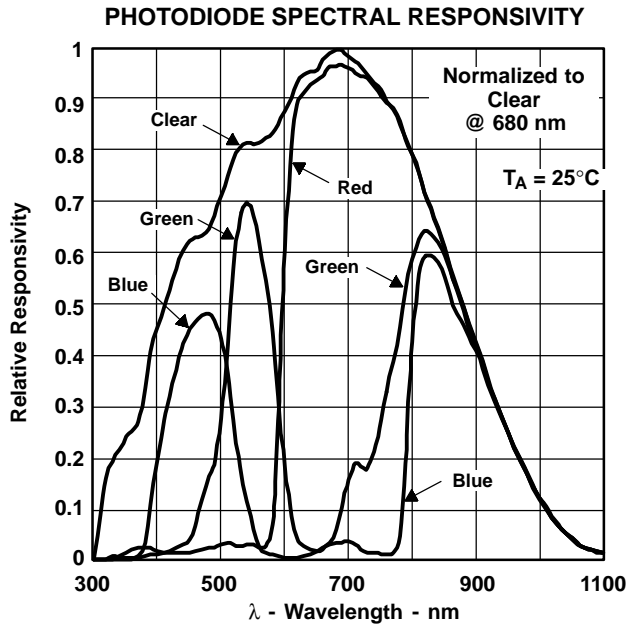


Figure 1

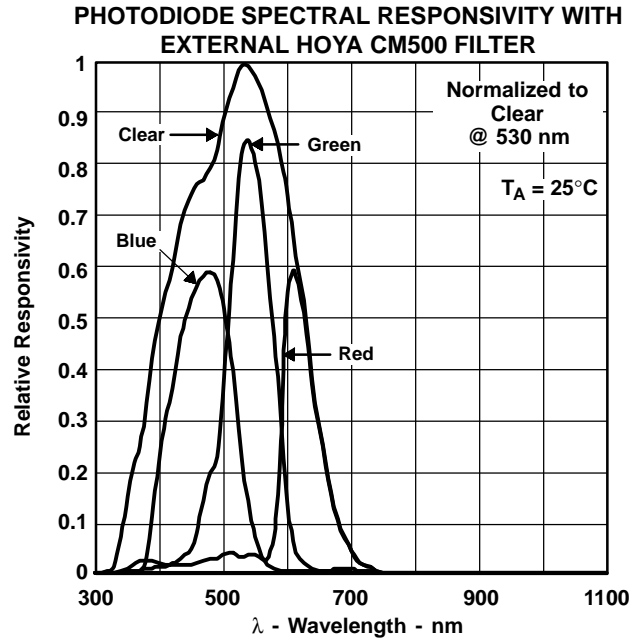


Figure 2

TCS230 PROGRAMMABLE COLOR LIGHT-TO-FREQUENCY CONVERTER

TAOS046 - FEBRUARY 2003

APPLICATION INFORMATION

Power supply considerations

Power-supply lines must be decoupled by a 0.01- μ F to 0.1- μ F capacitor with short leads mounted close to the device package.

Input interface

A low-impedance electrical connection between the device \overline{OE} pin and the device GND pin is required for improved noise immunity.

Output interface

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances. If lines greater than 12 inches are used on the output, a buffer or line driver is recommended.

Photodiode type (color) selection

The type of photodiode (blue, green, red, or clear) used by the device is controlled by two logic inputs, S2 and S3 (see Table 1).

Output frequency scaling

Output-frequency scaling is controlled by two logic inputs, S0 and S1. The internal light-to-frequency converter generates a fixed-pulsewidth pulse train. Scaling is accomplished by internally connecting the pulse-train output of the converter to a series of frequency dividers. Divided outputs are 50%-duty cycle square waves with relative frequency values of 100%, 20%, and 2%. Because division of the output frequency is accomplished by counting pulses of the principal internal frequency, the final-output period represents an average of the multiple periods of the principle frequency.

The output-scaling counter registers are cleared upon the next pulse of the principal frequency after any transition of the S0, S1, S2, S3, and \overline{OE} lines. The output goes high upon the next subsequent pulse of the principal frequency, beginning a new valid period. This minimizes the time delay between a change on the input lines and the resulting new output period. The response time to an input programming change or to an irradiance step change is one period of new frequency plus 1 μ S. The scaled output changes both the full-scale frequency and the dark frequency by the selected scale factor.

The frequency-scaling function allows the output range to be optimized for a variety of measurement techniques. The scaled-down outputs may be used where only a slower frequency counter is available, such as low-cost microcontroller, or where period measurement techniques are used.

Measuring the frequency

The choice of interface and measurement technique depends on the desired resolution and data acquisition rate. For maximum data-acquisition rate, period-measurement techniques are used.

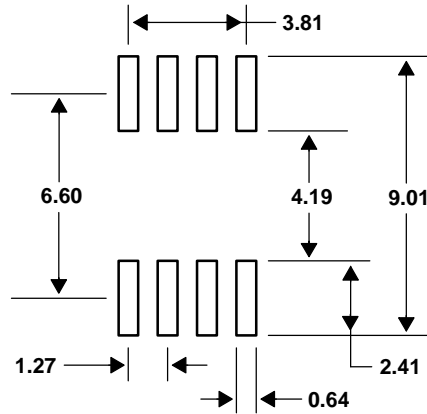
Output data can be collected at a rate of twice the output frequency or one data point every microsecond for full-scale output. Period measurement requires the use of a fast reference clock with available resolution directly related to reference clock rate. Output scaling can be used to increase the resolution for a given clock rate or to maximize resolution as the light input changes. Period measurement is used to measure rapidly varying light levels or to make a very fast measurement of a constant light source.

Maximum resolution and accuracy may be obtained using frequency-measurement, pulse-accumulation, or integration techniques. Frequency measurements provide the added benefit of averaging out random- or high-frequency variations (jitter) resulting from noise in the light signal. Resolution is limited mainly by available counter registers and allowable measurement time. Frequency measurement is well suited for slowly varying or constant light levels and for reading average light levels over short periods of time. Integration (the accumulation of pulses over a very long period of time) can be used to measure exposure, the amount of light present in an area over a given time period.

APPLICATION INFORMATION

PCB pad layout

Suggested PCB pad layout guidelines for the D package are shown in Figure 3.



- NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.

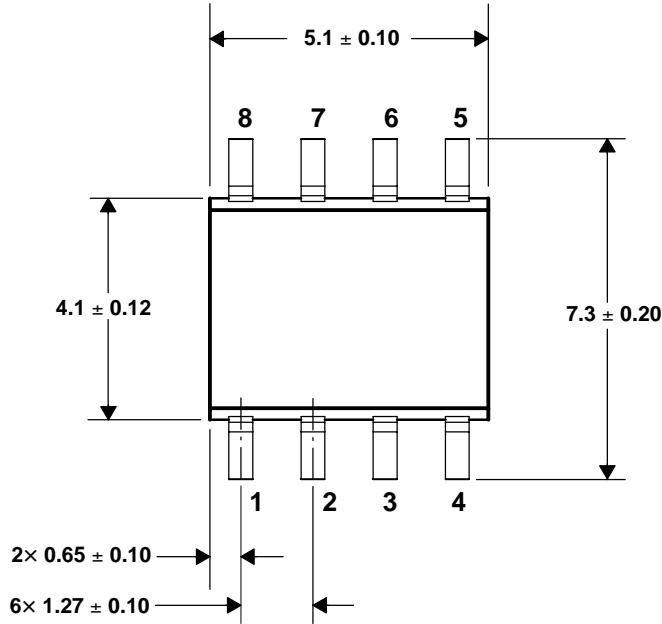
Figure 3. Suggested D Package PCB Layout

**TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER**

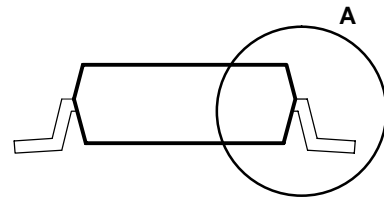
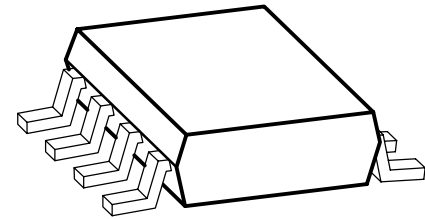
TAOS046 - FEBRUARY 2003

MECHANICAL INFORMATION

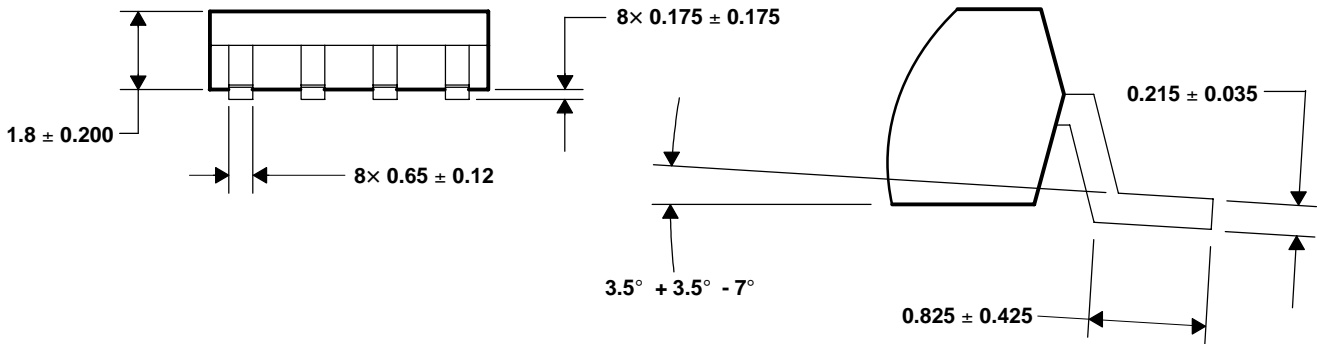
PACKAGE D



PLASTIC SMALL-OUTLINE PACKAGE



DETAIL A



- NOTES:
- A. All linear dimensions are in millimeters.
 - B. Package is molded with an electrically nonconductive clear plastic compound having an index of refraction of 1.55.
 - C. Actual product will vary within the mechanical tolerances shown on this specification. Designs for use of this product **MUST** allow for the data sheet tolerances.
 - D. Pin 4 (GND) is mechanically connected to the die mount pad.
 - E. The 8×8 photodiode array area is $1.15 \text{ mm} \times 1.15 \text{ mm}$ (1.33 sq. mm).
 - F. This drawing is subject to change without notice.

Figure 4. TCS230 Mechanical Specifications

PRODUCTION DATA — information in this document is current at publication date. Products conform to specifications in accordance with the terms of Texas Advanced Optoelectronic Solutions, Inc. standard warranty. Production processing does not necessarily include testing of all parameters.

NOTICE

Texas Advanced Optoelectronic Solutions, Inc. (TAOS) reserves the right to make changes to the products contained in this document to improve performance or for any other purpose, or to discontinue them without notice. Customers are advised to contact TAOS to obtain the latest product information before placing orders or designing TAOS products into systems.

TAOS assumes no responsibility for the use of any products or circuits described in this document or customer product design, conveys no license, either expressed or implied, under any patent or other right, and makes no representation that the circuits are free of patent infringement. TAOS further makes no claim as to the suitability of its products for any particular purpose, nor does TAOS assume any liability arising out of the use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages.

TEXAS ADVANCED OPTOELECTRONIC SOLUTIONS, INC. PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN CRITICAL APPLICATIONS IN WHICH THE FAILURE OR MALFUNCTION OF THE TAOS PRODUCT MAY RESULT IN PERSONAL INJURY OR DEATH. USE OF TAOS PRODUCTS IN LIFE SUPPORT SYSTEMS IS EXPRESSLY UNAUTHORIZED AND ANY SUCH USE BY A CUSTOMER IS COMPLETELY AT THE CUSTOMER'S RISK.

LUMENOLOGY is a registered trademark, and TAOS, the TAOS logo, and Texas Advanced Optoelectronic Solutions are trademarks of Texas Advanced Optoelectronic Solutions Incorporated.

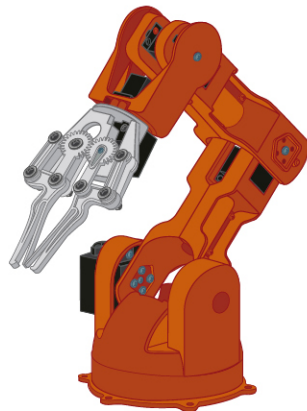
**TCS230
PROGRAMMABLE
COLOR LIGHT-TO-FREQUENCY CONVERTER**

TAOS046 - FEBRUARY 2003

ANNEXE B

BRACCIO

**QUICK
START
GUIDE**

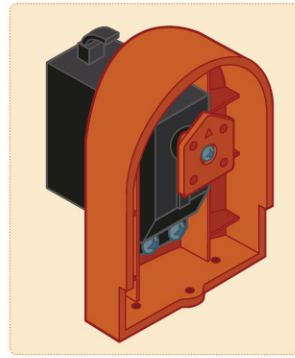
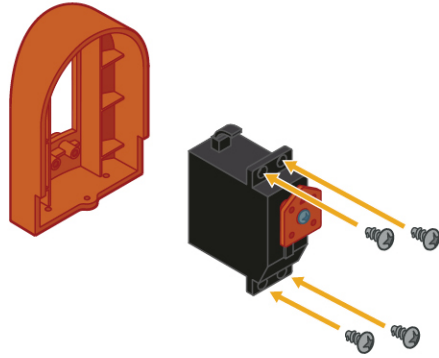


UNLOCK THE UNLIMITED POSSIBILITIES
OF ROBOTICS WITH THE BRACCIO

ASSEMBLY INSTRUCTIONS

CAUTION: DO NOT OVER-TIGHTEN SCREWS

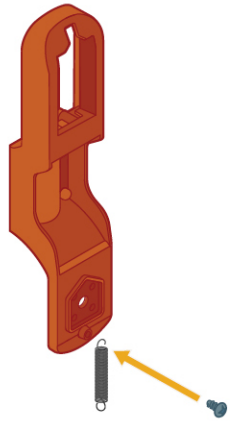
1



M2

4 x  Ø 3mm

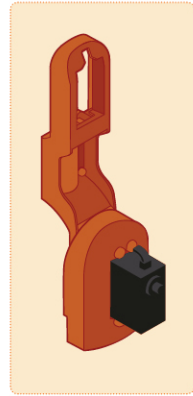
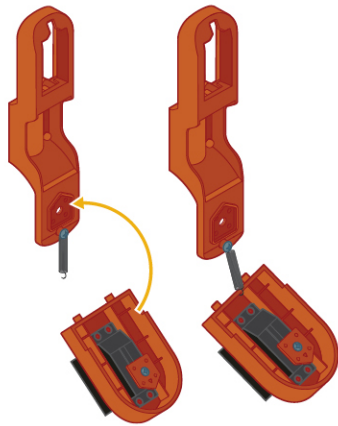
2



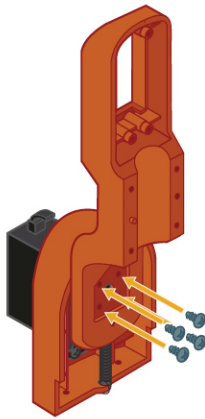
1 x  Ø 2mm

1 x 

3

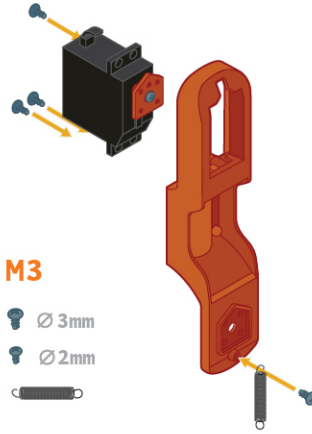


4



4 x  Ø 3mm

5

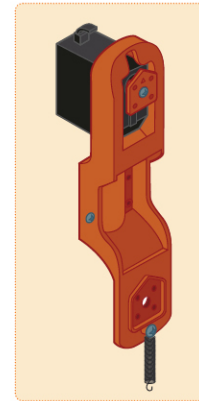


M3

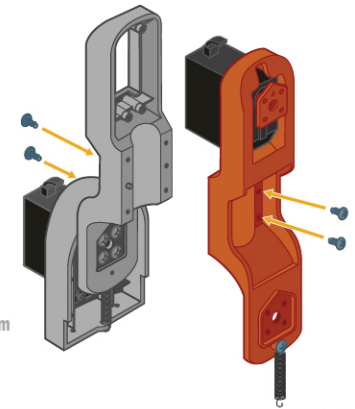
3 x  Ø 3mm

1 x  Ø 2mm

1 x 

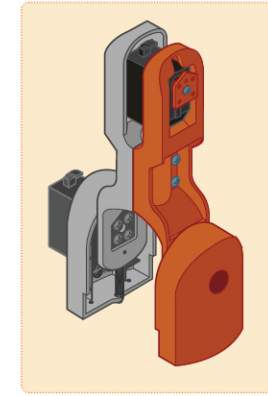
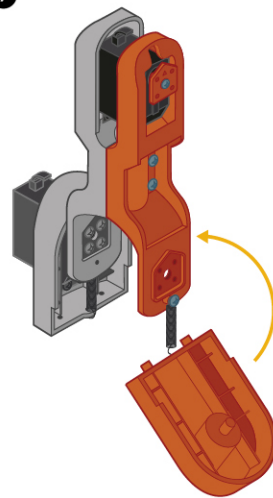


6

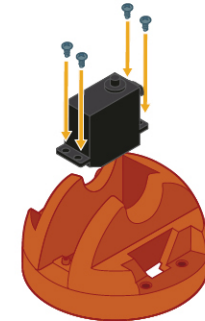


4 x  Ø 3mm

7

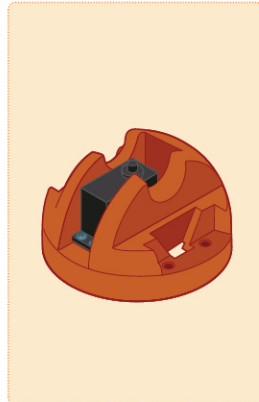


8

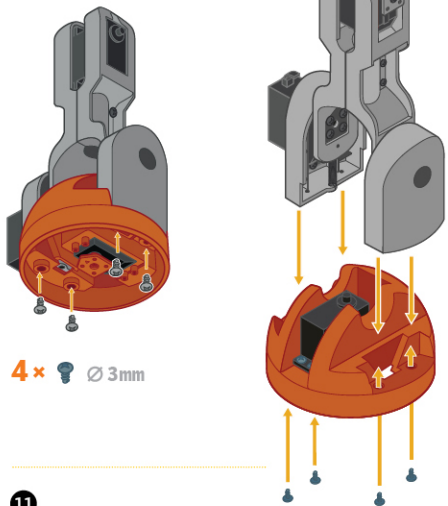


M1

4 x  Ø 3mm

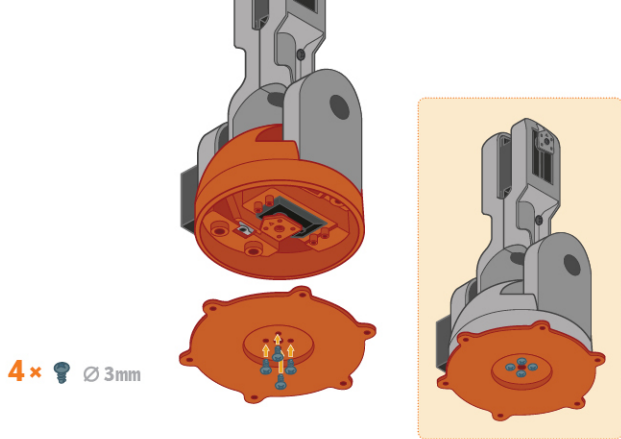


9



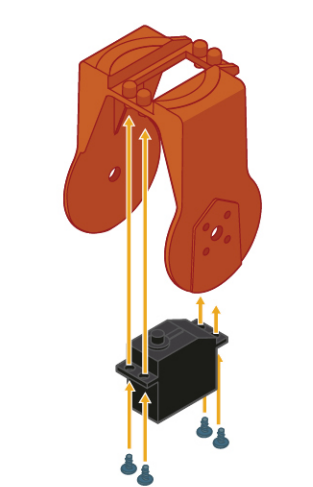
4 ×  Ø 3mm

10



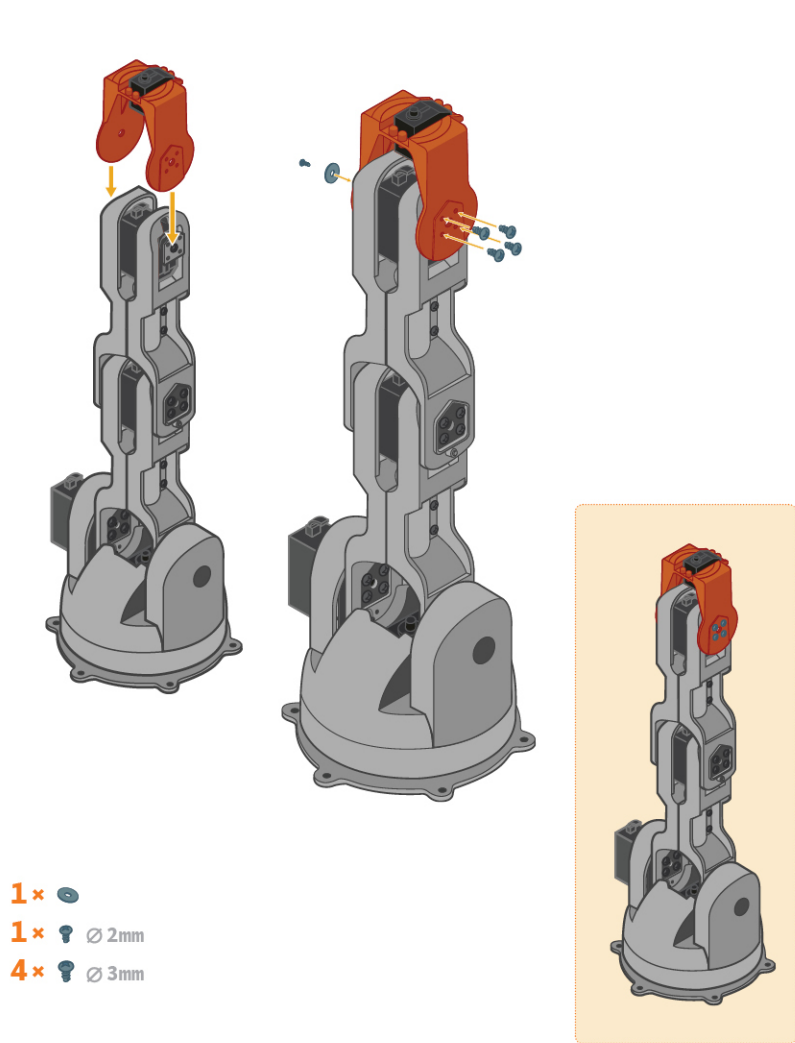
4 ×  Ø 3mm




14



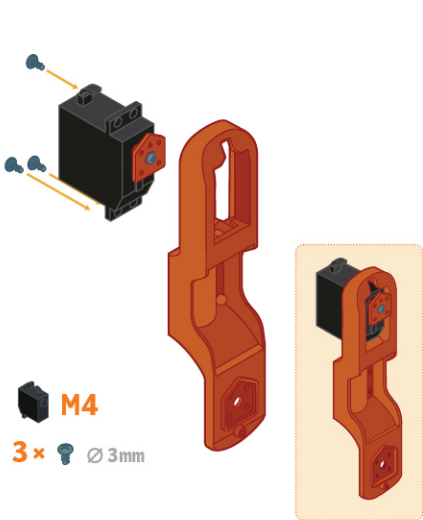
4 ×  Ø 3mm

15



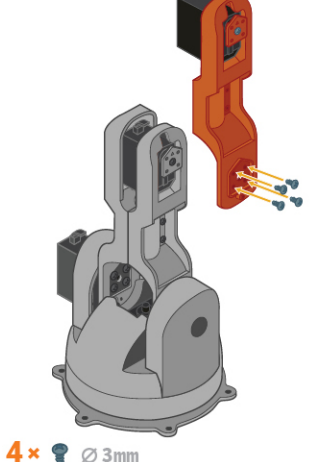
1 ×  Ø 2mm
1 ×  Ø 2mm
4 ×  Ø 3mm

11



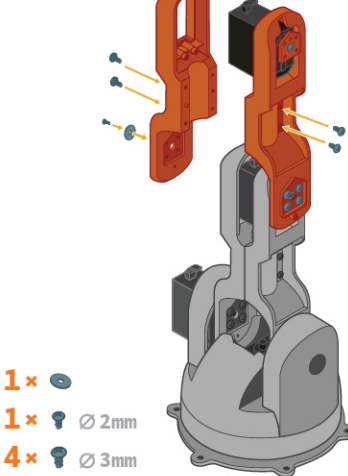
M4
3 ×  Ø 3mm




12

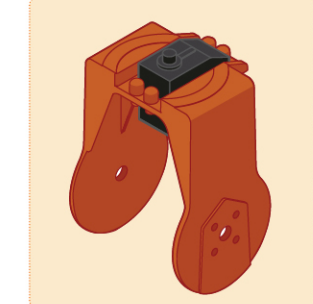


4 ×  Ø 3mm

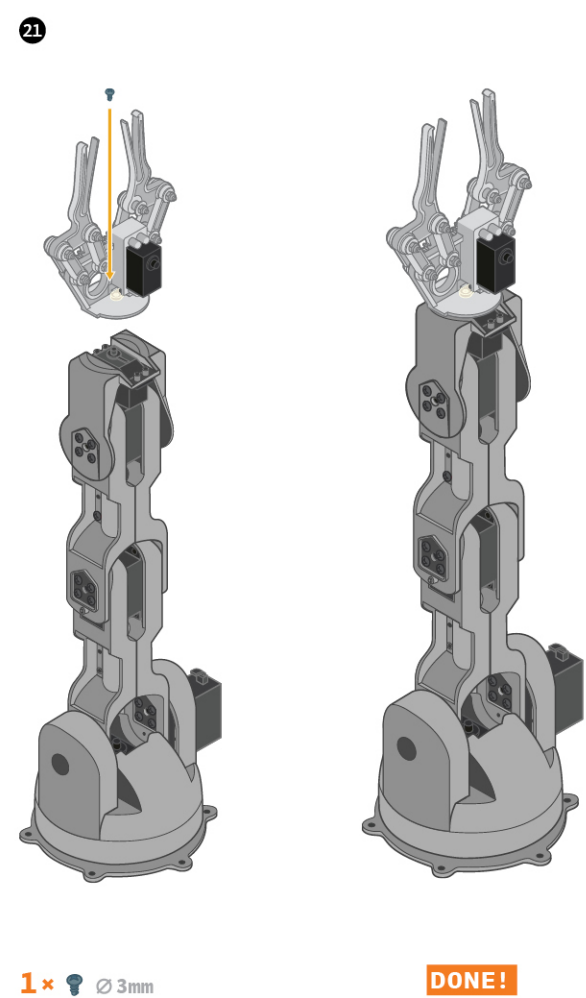
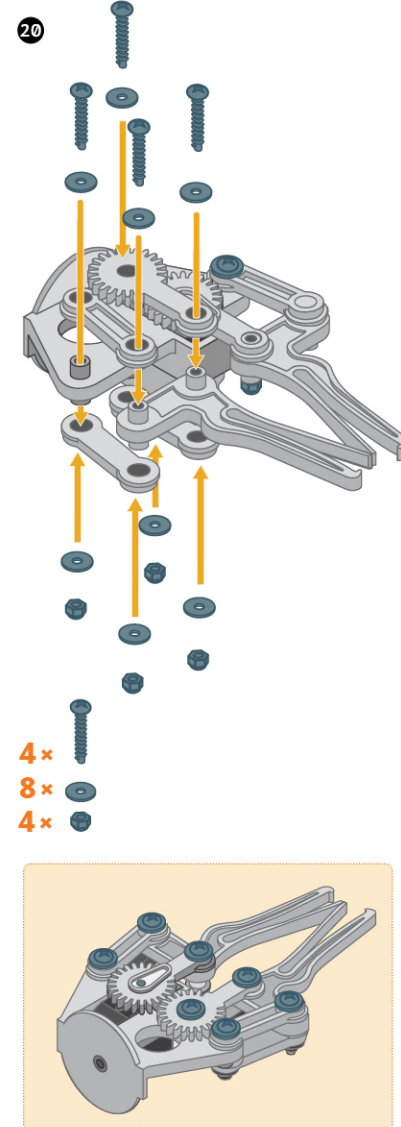
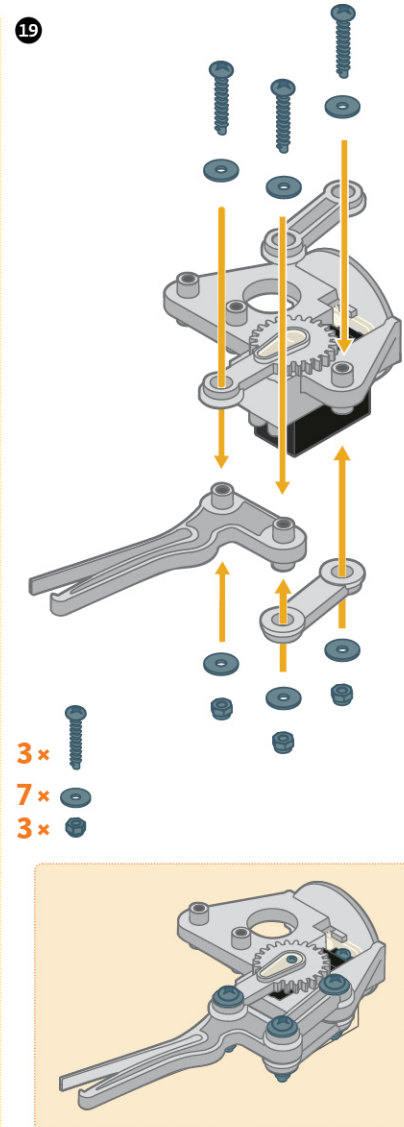
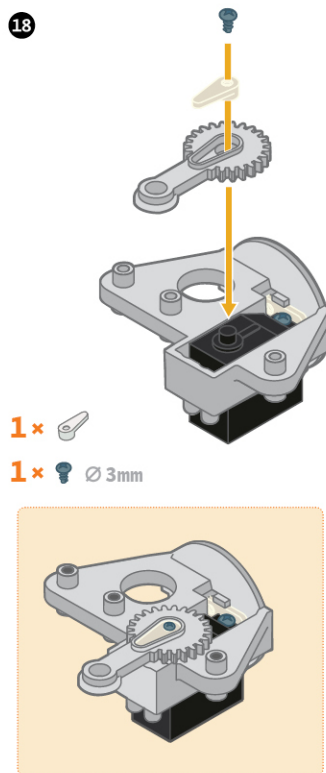
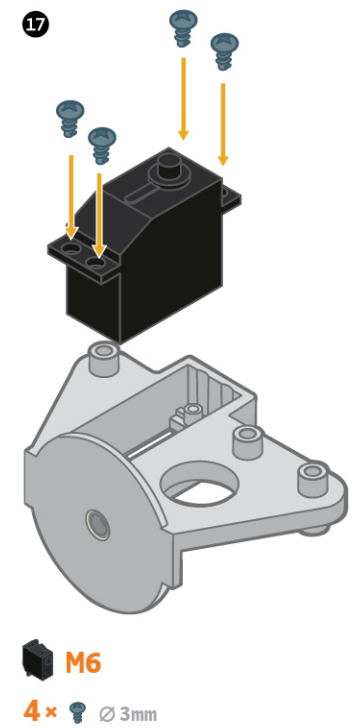
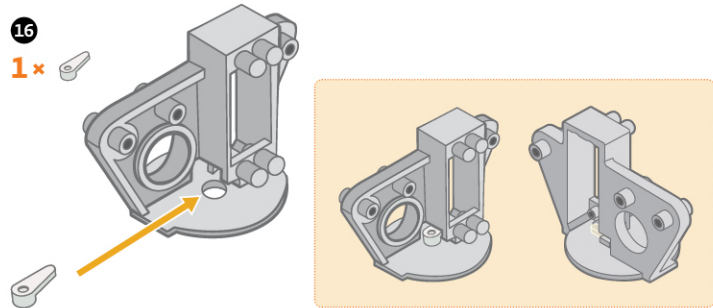
13



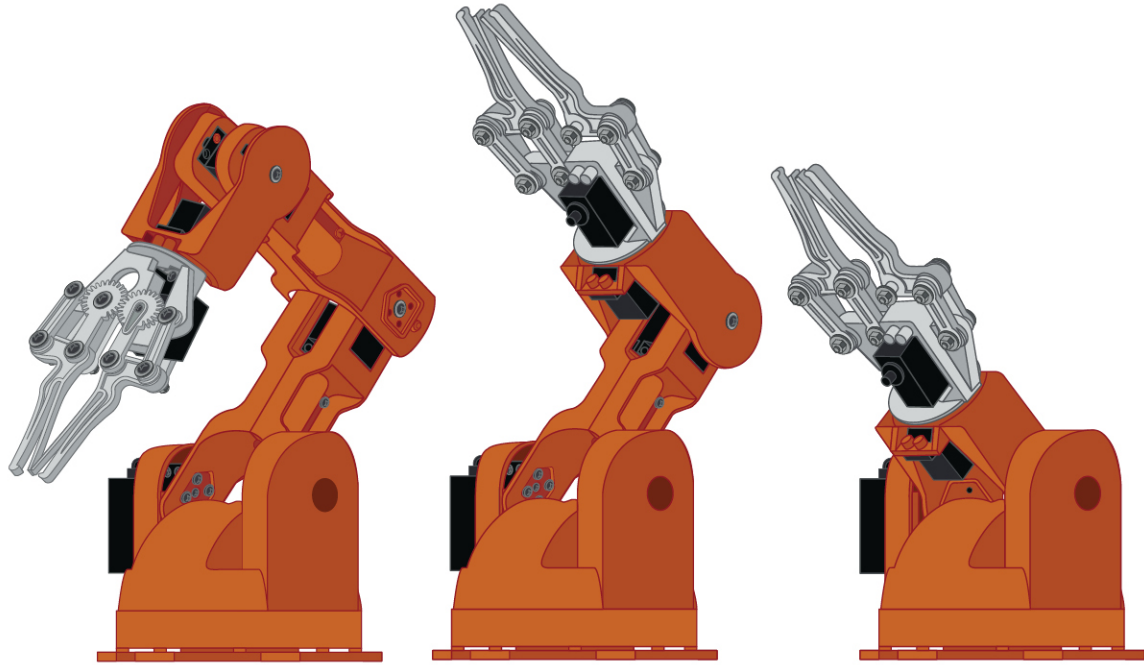
1 ×  Ø 2mm
1 ×  Ø 2mm
4 ×  Ø 3mm



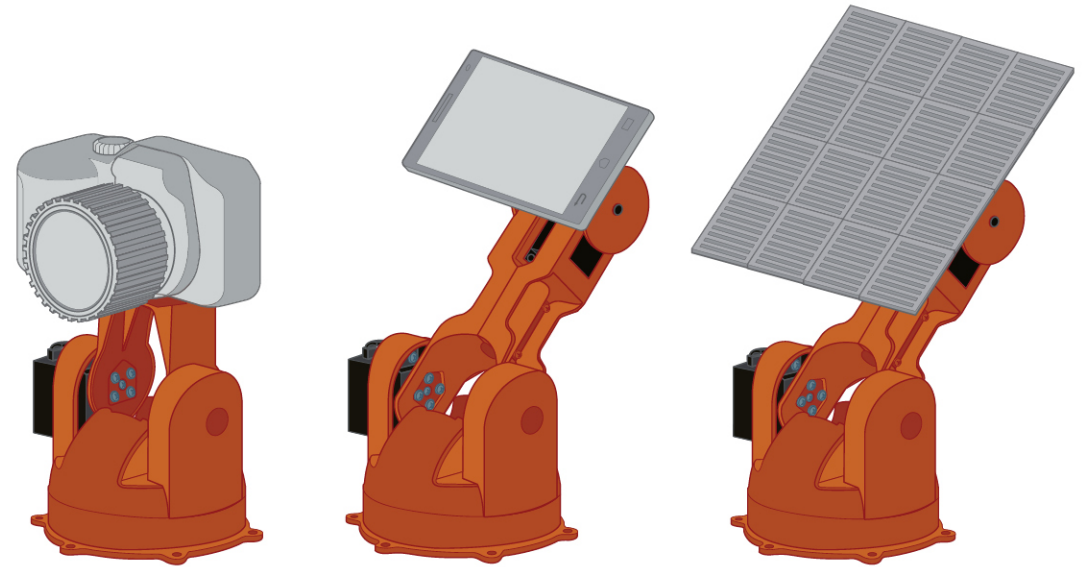
M5
4 ×  Ø 3mm



UNLOCK THE UNLIMITED
POSSIBILITIES OF ROBOTICS
WITH THE BRACCIO



DESIGNED FOR VERSATILITY,
THE BRACCIO CAN SUPPORT VARIOUS
OBJECTS ON THE END OF THE ARM



MOTORS ASSEMBLY

MOTOR "1" BASE

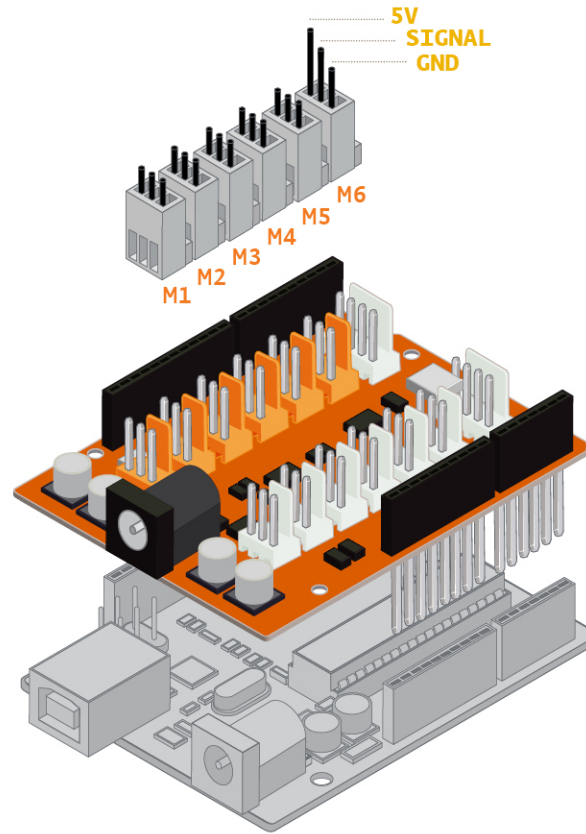
MOTOR "2" SHOULDER

MOTOR "3" ELBOW

MOTOR "4" VERTICAL WRIST

MOTOR "5" ROTATORY WRIST

MOTOR "6" GRIPPER



ARDUINO COMPATIBLE BOARDS

UNO	LEONARDO
UNO SMD	LEONARDO ETH
DUE	YUN
MEGA 2560	TIAN
MEGA ADK	UNO WIFI
ETHERNET	

CONNECTING TO YOUR COMPUTER

1 DOWNLOAD THE SOFTWARE

Get the latest version of the Arduino Software from [ARDUINO.ORG/DOWNLOADS](https://www.arduino.org/downloads)

2 CONNECT THE BRACCIO

Plug the USB cable to the control board port, and wait for hardware installation to finish

3 CONNECT YOUR BOARD


Start the Arduino IDE

Select **Tools** → **Board** → Select the board you are using

Select the correct serial port from **Tools** → **Port**

4 LOAD A SKETCH

Select a sketch from **File** → **Examples** → **Braccio**

Press the Upload button  and wait for the program to finish uploading

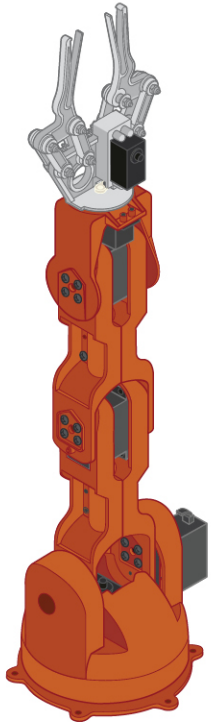
CONGRATULATIONS!
You are ready to experiment and play

ONLINE TUTORIALS AND INFORMATION:
[ARDUINO.ORG/BRACCIO](https://www.arduino.org/braccio)

RUN YOUR SKETCH

1 TESTBRACCIO90

"testBraccio90" is a setup sketch allowing you to check the alignment of all the servo motors. It is also the first sketch you need to run on the Braccio. The sketch will position the Braccio in the upright position as seen in the picture below. If it doesn't put the Braccio in the exact setting, you need to realign the position of the servo motors.



M1 = base degrees
M2 = shoulder degrees
M3 = elbow degrees
M4 = vertical wrist degrees
M5 = rotatory wrist degrees
M6 = gripper degrees

`Braccio.begin();`

Initialization functions and set up the initial position for Braccio.

All the servo motors will be positioned in the "safety" position: M1 = 90°, M2 = 45°, M3 = 180°, M4 = 180°, M5 = 90°, M6 = 10°.

The sketch will position the Braccio in the upright position.

Step Delay: a milliseconds delay between the movement of each servo. Allowed values: from 10 to 30 msec.

M1 allowed values from 0° to 180°
M2 allowed values from 15° to 165°
M3 allowed values from 0° to 180°
M4 allowed values from 0° to 180°
M5 allowed values from 0° to 180°
M6 allowed values from 10° to 73°. (10°: the gripper is open, 73°: the gripper is closed).

```
1 #include <Braccio.h>
2 #include <Servo.h>
```

```
3 Servo base;
4 Servo shoulder;
5 Servo elbow;
6 Servo wrist_ver;
7 Servo wrist_rot;
8 Servo gripper;
```

```
9 void setup() {
10   Braccio.begin();
11 }
```

```
12 void loop() {
13     //(step delay M1, M2, M3, M4, M5, M6);
14   Braccio.ServoMovement(20, 90, 90, 90, 90, 90, 73);
15 }
```

2 SIMPLEMENTMENTS

The “simpleMovements” sketch shows you how each servo motor of the Braccio moves.

M1 = base degrees
M2 = shoulder degrees
M3 = elbow degrees
M4 = vertical wrist degrees
M5 = rotatory wrist degrees
M6 = gripper degrees

Braccio.begin();

Initialization functions and set up the initial position for Braccio.

All the servo motors will be positioned in the “safety” position: M1 = 90°, M2= 45°, M3 = 180°, M4 = 180°, M5 = 90°, M6 = 10°.

The **delay()** function lets you stop the Arduino from executing anything for a period of time.

Step Delay: a milliseconds delay between the movement of each servo. Allowed values: from 10 to 30 msec.

M1 allowed values from 0° to 180°
M2 allowed values from 15° to 165°
M3 allowed values from 0° to 180°
M4 allowed values from 0° to 180°
M5 allowed values from 0° to 180°
M6 allowed values from 10° to 73°. (10°: the gripper is open, 73°: the gripper is closed).

```
1 #include <Braccio.h>
2 #include <Servo.h>
```

```
3 Servo base;
4 Servo shoulder;
5 Servo elbow;
6 Servo wrist_ver;
7 Servo wrist_rot;
8 Servo gripper;
```

```
9 void setup() {
10   Braccio.begin();
11 }
```

```
12 void loop() {
13     //(step delay M1, M2, M3, M4, M5, M6);
14   Braccio.ServoMovement(20, 15, 0, 180, 180, 0, 73);
15   delay(1000);
16   Braccio.ServoMovement(20, 165, 180, 0, 0, 180, 10);
17   delay(1000);
18 }
```

3 TAKETHESPONGE

This example tells the Braccio to take the sponge from the table and show it to the user.

M1 = base degrees
M2 = shoulder degrees
M3 = elbow degrees
M4 = vertical wrist degrees
M5 = rotatory wrist degrees
M6 = gripper degrees

Braccio.begin();

Initialization functions and set up the initial position for Braccio.

All the servo motors will be positioned in the "safety" position: M1 = 90°, M2 = 45°, M3 = 180°, M4 = 180°, M5 = 90°, M6 = 10°.

Starting position.

One second **delay**.

The braccio moves to the sponge.

Close the tongue to take the sponge.

Brings the sponge upwards.

Show the sponge.

Return to the start position.

Open the gripper.

For **Step Delay** and **Motors** values please refer to the previous sketches.

```
1 #include <Braccio.h>
2 #include <Servo.h>
```

```
3 Servo base;
4 Servo shoulder;
5 Servo elbow;
6 Servo wrist_ver;
7 Servo wrist_rot;
8 Servo gripper;
```

```
9 void setup() {
10   Braccio.begin();
11 }
```

```
12 void loop() {
13     //(step delay M1, M2, M3, M4, M5, M6);
14   Braccio.ServoMovement(20, 0, 45, 180, 180, 90, 10);
15   delay(1000);
16   Braccio.ServoMovement(20, 0, 90, 180, 180, 90, 10);
17   Braccio.ServoMovement(10, 0, 90, 180, 180, 90, 60);
18   Braccio.ServoMovement(20, 0, 45, 180, 45, 0, 60);
19   Braccio.ServoMovement(20, 180, 45, 180, 45, 0, 60);
20   Braccio.ServoMovement(20, 0, 90, 180, 180, 90, 60);
21   Braccio.ServoMovement(20, 0, 90, 180, 180, 90, 10);
22 }
```


CERTIFICATE OF ORIGIN

Thank you for choosing a Tinkerkit product. This product was produced, assembled and tested in Italy. All parts but the motors come from Italy and all of the manufacturing, assembling, testing and packaging took place entirely in Italy.

MANUFACTURING

All parts used in this product comply with the RoHS Directive. The RoHS Directive prevents all new electrical and electronic equipment placed on the market in the European Economic Area from containing more than agreed levels of lead, cadmium, mercury, hexavalent chromium, poly-brominated biphenyls (PBB) and polybrominated diphenyl ethers (PBDE). The product is labelled with the CE logo, as it meet the electromagnetic compatibility standards set by the EU. All parts are tested to meet EU consumer safety, health and environmental requirements.

RETURN POLICY

We put all of our expertise and care in this product. Should you, despite all our efforts, find any faults in it, please contact your distributor to find out whether you qualify for a product replacement.

LIMITED WARRANTY STATEMENT

1. WARRANTIES.

1.1

ARDUINO warrants that its products will conform to the Specifications. This warranty lasts for one (1) year from the date of the sale. ARDUINO shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by the Customer. Moreover, ARDUINO shall not be liable for any defects that result from the Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent ARDUINO deems necessary.

1.2

If any ARDUINO products fail to conform to the warranty set forth above, ARDUINO's sole liability shall be to replace such products. ARDUINO's liability shall be limited to products that are determined by ARDUINO not to conform to such warranty. If ARDUINO elects to replace such products, ARDUINO shall be given a reasonable time to provide replacements. Replaced products shall be warranted for a new full warranty period.

1.3

EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". ARDUINO DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4

The Customer agrees that prior to using any systems that include ARDUINO products, the Customer will test such systems and the functionality of the products as used in such systems. ARDUINO may provide technical, applications or design advice, quality characterization, reliability data or other services. The Customer acknowledges and agrees that providing these services shall not expand or otherwise alter ARDUINO's warranties, as set forth above, and that no additional obligations or liabilities shall arise from ARDUINO providing such services.

1.5

ARDUINO products are not authorized for use in safety-critical applications where a failure of the ARDUINO product would reasonably be expected to cause severe personal injury or death. Safety-critical applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. ARDUINO products are neither designed nor intended for use in military or aerospace applications or environments, nor for automotive applications or the automotive environment. The Customer acknowledges and agrees that any such use of ARDUINO products is solely at the Customer's risk, and that the Customer is the sole responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6

The Customer acknowledges and agrees that the Customer is the sole responsible for compliance with all legal, regulatory and safety-related requirements concerning the products and any use of ARDUINO products in the Customer's applications, notwithstanding any applications-related information or support that may be provided by ARDUINO.

2. CONSEQUENTIAL DAMAGES WAIVER.

In no event shall ARDUINO be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether ARDUINO has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

3. CHANGES TO SPECIFICATIONS.

ARDUINO may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". ARDUINO reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

FCC COMPLIANCE

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna;
- Increase the separation between the equipment and receiver;
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected;
- Consult the dealer or an experienced radio/TV technician for help.

The following parties are responsible for the compliance of radio frequency equipment with the applicable standards: in the case of equipment subject to authorization under the verification procedure, the manufacturer or, in the case of imported equipment, the importer. If subsequent to manufacture and importation, the radio frequency equipment is modified by any party not working under the authority of the responsible party, the party performing the modification becomes the new responsible party.

Manufactured by
ARDUINO S.r.l.
Via Romano, 12
10010 Scarmagno
Italy



RoHS COMPLIANT

POWERED BY



[ARDUINO.ORG/BRACCIO](https://www.arduino.org/braccio)

ANNEXE C

Programme de détection des couleurs :

```
////////////////////////////////////// déclaration des variables//////////////////////////////////////  
  
const int s0 = 14;  
  
const int s1 = 15;  
  
const int s2 = 16;  
  
const int s3 = 17;  
  
const int out = 18;  
  
int red = 0;  
  
int green = 0;  
  
int blue = 0;  
  
//////////////////////////////////////configuration des entrées / sortie ////////////////////////////////////////  
  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(s0, OUTPUT);  
  pinMode(s1, OUTPUT);  
  pinMode(s2, OUTPUT);  
  pinMode(s3, OUTPUT);  
  pinMode(out, INPUT);  
  pinMode(redLed, OUTPUT);  
  pinMode(greenLed, OUTPUT);  
  pinMode(blueLed, OUTPUT);  
  digitalWrite(s0, HIGH);  
  digitalWrite(s1, HIGH);  
}
```

//la fonction principale//

```
void loop()
{
  color();
  Serial.print("R Intensité:");
  Serial.print(red, DEC);
  Serial.print(" V Intensité: ");
  Serial.print(green, DEC);
  Serial.print(" B Intensité : ");
  Serial.print(blue, DEC);

  if (red < blue && red < green && red < 20)
  {
    Serial.println(" - (couleur du produit : rouge )");
    digitalWrite(redLed, HIGH); // Turn RED LED ON
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, LOW);
    delay(5000);
  }
  else if (blue < red && blue < green )
  {
    Serial.println(" - (couleur du produit: bleu)");
    digitalWrite(redLed, LOW);
    digitalWrite(greenLed, LOW);
    digitalWrite(blueLed, HIGH); // Turn BLUE LED ON
    delay(5000);
  }
  else if (green < red && green < blue )
```

```

{
  Serial.println(" - (couleur du produit : vert )");
  digitalWrite(redLed, LOW);
  digitalWrite(greenLed, HIGH); // Turn GREEN LED ON
  digitalWrite(blueLed, LOW);
  delay(5000);
}
else{
  Serial.println();
}
delay(300);
digitalWrite(redLed, LOW);
digitalWrite(greenLed, LOW);
digitalWrite(blueLed, LOW);
}

//////////////////////////////////// Programme de détection des couleurs //////////////////////////////////////

void color()
{
  digitalWrite(s2, LOW);
  digitalWrite(s3, LOW);

  //count OUT, pRed, RED
  red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s3, HIGH);

  //count OUT, pBLUE, BLUE
  blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);
  digitalWrite(s2, HIGH);

  //count OUT, pGreen, GREEN

```

```
green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);  
}
```

Le programme de notre système de tri :

```
//=====Déclaration des variables=====//  
//////////////////////////////////// bras robotique //////////////////////////////////////  
  
#include <Braccio.h>  
  
#include <Servo.h>  
  
Servo base;  
  
Servo shoulder;  
  
Servo elbow;  
  
Servo wrist_rot;  
  
Servo wrist_ver;  
  
Servo gripper;  
  
////////////////////////////////////capteur de couleur //////////////////////////////////////  
  
const int s0 = 25;  
  
const int s1 = 24;  
  
const int s2 = 36;  
  
const int s3 = 40;  
  
const int out = 32;  
  
//===== Configuration des entrées / sortie =====//  
  
void setup()  
{  
  Serial.begin(9600);  
  
  pinMode(s0, OUTPUT);  
  
  pinMode(s1, OUTPUT);  
  
  pinMode(s2, OUTPUT);  
  
  pinMode(s3, OUTPUT);  
}
```

```

pinMode(out, INPUT);
pinMode(redLed, OUTPUT);
pinMode(greenLed, OUTPUT);
pinMode(blueLed, OUTPUT);
digitalWrite(s0, HIGH);
digitalWrite(s1, HIGH);

//Initialization functions and set up the initial position for Braccio
//All the servo motors will be positioned in the "safety" position:
//Base (M1):90 degrees
//Shoulder (M2): 45 degrees
//Elbow (M3): 180 degrees
//Wrist vertical (M4): 180 degrees
//Wrist rotation (M5): 90 degrees
//gripper (M6): 10 degrees
Braccio.begin();
}
void loop()
{
  /*===== Angle autorisé pour chaque servomoteur =====

  Step Delay: a milliseconds delay between the movement of each servo. Allowed values from 10 to 30
  msec.

  M1=base degrees. Allowed values from 0 to 180 degrees
  M2=shoulder degrees. Allowed values from 15 to 165 degrees
  M3=elbow degrees. Allowed values from 0 to 180 degrees
  M4=wrist vertical degrees. Allowed values from 0 to 180 degrees
  M5=wrist rotation degrees. Allowed values from 0 to 180 degrees
  M6=gripper degrees. Allowed values from 10 to 73 degrees. 10: the tongue is open, 73: the gripper is
  closed.

  */

```

```

color();

Serial.print("R :");

Serial.print(red, DEC);

Serial.print(" v : ");

Serial.print(green, DEC);

Serial.print(" B : ");

Serial.print(blue, DEC);

//Serial.println();

if (red < blue && red < green && red <20)

{

Serial.println(" - (COULEUR DETECTER : ROUGE)");

sred();

digitalWrite(redLed, HIGH); // Turn RED LED ON

digitalWrite(greenLed, LOW);

digitalWrite(blueLed, LOW);

}

else if (blue < red && blue < green && blue <20)

{

Serial.println(" - (COULEUR DETECTER : BLEU)");

sblue();

digitalWrite(redLed, LOW);

digitalWrite(greenLed, LOW);

digitalWrite(blueLed, HIGH); // Turn BLUE LED ON

}

else if (green < red && green < blue && green < 20 )

{

Serial.println(" - ( COUIEUR DETECTER : VERT)");

sgreen();

digitalWrite(redLed, LOW);

```

```

digitalWrite(greenLed, HIGH); // Turn GREEN LED ON

digitalWrite(blueLed, LOW);

}

else{

Serial.println();

}

delay(300);

digitalWrite(redLed, LOW);

digitalWrite(greenLed, LOW);

digitalWrite(blueLed, LOW);

Braccio.ServoMovement(30,      90, 100, 90, 90, 90, 10);

}

//===== Fonction qui permet la détection des 3 couleurs =====//

void color()

{

digitalWrite(s2, LOW);

digitalWrite(s3, LOW);

//count OUT, pRed, RED

red = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s3, HIGH);

//count OUT, pBLUE, BLUE

blue = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

digitalWrite(s2, HIGH);

//count OUT, pGreen, GREEN

green = pulseIn(out, digitalRead(out) == HIGH ? LOW : HIGH);

}

```

```

void sred(){

    //Starting position

    //////////////////////////////////////////////////Mouvement de dépalacement des produits de couleur rouge////////////////////////////////////

        //(step delay  M1 , M2 , M3 , M4 , M5 , M6);

    Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 10);
    delay(1000);

    Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 10);
    delay(1000);

    Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 55 );
    delay(1000);

    Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 55);
    delay(1000);

    Braccio.ServoMovement(20,      0, 100, 140, 100, 90, 55);
    Braccio.ServoMovement(20,      0, 100, 140, 170, 90, 55);
    Braccio.ServoMovement(20,      0, 100, 140, 170, 90, 55 );
    delay(1000);

    Braccio.ServoMovement(20,      0, 100, 140, 170, 90, 10 );
    delay(1000);

}

    //////////////////////////////////////////////////Mouvement dépalacement des produits de couleur bleu////////////////////////////////////

void sblue(){

    //Starting position

        //(step delay  M1 , M2 , M3 , M4 , M5 , M6);

    Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 10);

    //Wait 1 second

    delay(1000);

    Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 10);
    delay(1000);

    Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 55 );
}

```

```

delay(1000);

Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 55);

delay(1000);

Braccio.ServoMovement(20, 180, 100, 140, 100, 90, 55);
Braccio.ServoMovement(20, 180, 100, 140, 170, 90, 55);
Braccio.ServoMovement(20, 180, 100, 140, 170, 90, 55 );
delay(1000);
Braccio.ServoMovement(20, 180, 100, 140, 170, 90, 10 );
delay(1000);
}

void sgreen(){

//Starting position

//(step delay M1 , M2 , M3 , M4 , M5 , M6);

////////////////////Mouvement dépalacement des produits de couleur verte //////////////////////

//(step delay M1 , M2 , M3 , M4 , M5 , M6);

Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 10);

//Wait 1 second

delay(1000);

Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 10);

delay(1000);

Braccio.ServoMovement(20,      90, 78, 180, 180, 90, 55 );

delay(1000);

Braccio.ServoMovement(20,      90, 45, 180, 180, 90, 55);

delay(1000);

Braccio.ServoMovement(20,      90, 90, 30, 30, 90, 55);

delay(1000);

Braccio.ServoMovement(20,      90, 90, 30, 30, 90, 10);

delay(1000);

}

```

