Nomenclature

Les principales notations et abréviations utilisées dans ce mémoire sont explicitées ci-dessous, sous leur forme la plus couramment employée dans le domaine du génie électrique.

Grandeurs électriques et mécaniques

Nom	Symbole	Unité
Temps	t	s
Tension	U	V
Courant	I	A
Puissance	P	W
Fréquence	f	Hz
Couple	C	Nm
Angle (position)	Θ	°, rad

Glossaire

- MCC Moteur à Courant Continu
- MLI Modulation par largeur d'impulsion
- PI correcteur Proportionnel Intégrale
- Ki coefficient du correcteur intégral
- Kp coefficient du correcteur Proportionnel
- PWM Pulse With Modulation
- DC Direct Current
- PC Personnel Computer

Sommaire

Table des figures

Chapitre I : ETAT DE L'ART

Chapitre II : description de projet

Chapitre III : réalisation de projet

Introduction générale

Le développement de la science et de la technologie, dans les domaines de l'électronique, l'informatique, mathématique et la mécanique, a permis à l'homme de construire des robots hautement perfectionnées capables de se substituer à l'humaine dans ses fonctions motrices, sensorielles et intellectuelles en leur conférant une intelligence artificielle, comme les robots d'exploration, les robots d'intervention, les robots ludiques, les robots de service, industriel, etc. Parmi tous ces robots nous nous sommes intéressées au bras manipulateur qui est énormément utilisé dans le milieu industriel car il peut réaliser des tâches de façon répétitive, à hautes vitesses, pour satisfaire des temps d'exécution désirés.

Ces robots ont besoin d'une phase de programmation afin de générer les plans d'actions nécessaires pour l'exécution des tâches. Un système de programmation des robots d'assemblage doit assurer trois fonctions principales. La phase de programmation est une phase très importante pour générés les trajectoires nécessaires pour réaliser les tache désirés.

Nous avons choisi la réalisation d'une interface graphique pour commander un bras manipulateur KSR10 par un joystick à travers une carte Arduino. Pour cela nous avons divisé notre travaille selon le plan suivant :

Dans le premier chapitre, nous commençons par une présentation générale sur les robots, leurs développements, leur évolution dans le temps, leurs constitutions, leurs types et en fin leurs structure.

Le deuxième chapitre est consacré à la description du bras KSR10 (mécanique, électrique), l'architecture de la carte de commande, le développement de l'interface graphique et la programmation de la carte de commande.

Le troisième chapitre comporte les différentes étapes suivies pour la réalisation de l'interface graphique et des circuits de commande.

Enfin notre travail s'achève par une conclusion générale et quelques perspectives.

Chapitre I : ETAT DE L'ART

1. Introduction

Le mot robot a été introduit pour la première fois dans une pièce de théâtre écrite en 1920: « les robots universels de Rossum » (Rossum′s Universel Robots) par le tchèque Karel Capek. C'est également un terme qui est dérivé du verbe « robota » qui signifie en tchèque «travail forcé ». Dans les années quarante, l'invention des transistors et des circuits intégrés a permis de miniaturiser et développer des circuits électronique. Ce qui a ouvert de nouveaux horizons à la fabrication de robots. En 1945 il y a eu l'invention du premier robot programmable. En 1954, il y a eu l'invention du premier robot industriel. Cependant, l'application du robot n'a vu le jour qu'en 1961 dans l'usine de Général Motors pour extraire des pièces d'une cellule de fonderie sous pression. Ce bras manipulateur exécutait des commandes stockées sur un tambour magnétique [XA 11].

FIGURE 1.1 – Scène de R. U. R. montrant les trois robots. [XA 11]

L'organisation internationale de normalisation (ISO) a défini un robot industriel comme étant un manipulateur automatiquement commandé, reprogrammable et possédant trois axes ou plus [XA 11].

FIGURE 1.2 – Premier robot industriel 'Unimate'[W7].

Ces robots manipulateurs sont reprogrammables et peuvent effectuer de nombreuses tâches telles que déplacer des matériaux, des pièces, et certaines autres tâches. Cette définition englobe une grande variété de manipulateurs robotiques. La présence des robots manipulateurs dans l'industrie est aujourd'hui banalisée notamment dans le domaine de l'automobile (Figure 1.3).

FIGURE 1.3 – Ligne de production robotisée [W8].

Ces robots sont conçus pour fonctionner dans des environnements dont tous les paramètres sont précisément contrôlés. Ils répètent les mouvements qu'on leur a programmés sans informations sur l'environnement ou la tâche effectuée. C'est ce qui rend ces systèmes

incapables de faire face à des changements de structure ou d'environnement sans reconfiguration ou reprogrammation. On mentionne aussi les robots « play-back », ces robots qui reproduisent la tâche apprise ; et les robots à commande numérique qui sont des robots qui peuvent être programmés hors-ligne. La génération du mouvement de la trajectoire se fait à l'aide d'un «syntaxeur» (« boite à boutons », « joystick ») qui permet à un opérateur d'amener le robot en un certain nombre de points, qui sont ensuite mémorisés lors de l'exécution de la tâche. Ce qui permettra au robot de suivre une trajectoire passant successivement par tous les points programmés [Fi 09].

2. Types de robots

On peut classer les robots en plusieurs catégories, selon leurs complexités croissantes. Les robots qui répètent les opérations inscrites dans leurs programmes sont les robots les plus simples et les plus courants. Nous trouvons également les robots qui sont capable de reproduire certains mouvements humains enregistrés sur bande magnétique. Il y a aussi les robots qui reproduisent des mouvements enregistrés sur une unité de stockage et qui peuvent accomplir plusieurs opérations d'usinage différentes, ce sont des robots de troisième génération à commande numérique. Enfin, nous avons les robots qui font appel à des capteurs d'environnement et aux techniques d'intelligence artificielle et qui sont dotés d'une grande capacité de traitement de l'information. Ils sont appelés robots évolués et ce sont les robots les plus complexe [Fi 09].

3. Structure d'un robot

Un robot se compose de 7 parties principales : la structure mécanique articulée, la source d'énergie, les actionneurs, les transmissions, les capteurs internes, les capteurs externes et le système de commande [JU 94]:

- Structure mécanique articulée : est en quelque sorte le squelette du robot. Elle se compose de trois grandes parties fonctionnelles : le porteur (doté de degrés de liberté permettant de placer son extrémité en différents lieux), le véhicule (caractéristique des robots mobiles) et l'organe terminal (doté de possibilités d'orientation, et porte l'outil de travail).
- Source d'énergie : qui peut être d'origine hydraulique (développe la force la plus grande cependant elle est cher), pneumatique (simple, non polluant, bon marché,

maintenance facile mais bruyante) ou électrique (précise, fiable mais peu de puissance).

- Transmissions : assurant le lien entre les actionneurs et les articulations à mouvoir (câbles, rubans métalliques, chaîne, engrenage, …)
- Actionneurs : qui convertissent une énergie primaire en énergie mécanique, de type électriques, hydrauliques ou pneumatiques, suivant l'importance du système robotisé et la nature des tâches à accomplir afin de mouvoir ses articulations.
- Capteurs internes : donnant des informations sur la vitesse, la position et l'accélération.
- Capteurs externes : donnant des informations sur l'environnement.
- Système de commande : comportant deux modules fonctionnels (le premier sert à l'apprentissage, le second à l'exécution de la tâche apprise) il peut être décomposé en plusieurs sous-système (ordinateur, interface homme machine, micro-contrôleuse, PLC….) [JU 94].

4. Motorisation

Les moteurs électriques existants ne sont pas tous adaptés à la robotique. Les moteurs dédiés à la robotique sont généralement des dispositifs de faible ou moyenne puissance, typiquement inferieur à 1kW. Ils doivent pouvoir être commandés précisément avec une bonne résolution angulaire. De ce fait, des actionneurs comme les moteurs pas-à-pas ou les moteurs asynchrones sont peut utiliser en robotique. Les premiers sont en effet le plus souvent réservés à une utilisation en boucle ouverte, dotés d'un faible couple et d'une précision médiocre. Les seconds sont le plus souvent réservés à la forte puissance, par exemple pour la traction électrique, et leur électronique est complexe et couteuse. En robotique, les moteurs les plus fréquemment utilisés sont les moteurs à courant continu, avec ou sans balais [Fi 09].

Concernant le moteur asynchrone, il est utilisé depuis 1954, avec un pilotage analogique à fréquence fixe (couple proportionnel au carré de la tension dans les induits). Cette motorisation garantit un couple résistant très faible et régulier, mais les possibilités de l'embarquer sont limitées à cause de leur faible couple unitaire ainsi que leurs faible rendement au pilotage ce qui les oblige à une surveillance étroite de leurs température [W3].

Concernant les moteurs pas à pas qui sont apparus au début des années 30. Ils ont vraiment été développés qu'à partir des années 60, du faite de l'apparition et le développement des microprocesseurs et de l'électronique de puissance. Il existe trois types de moteur pas à pas à savoir à aimant permanent, à reluctance variable et hybrides. Pour tous ces

types de moteur, on positionne le rotor en modifiant la direction du champ magnétique crée par les bobinages du stator. Ils nécessitent un circuit de commande qui comporte une partie logique et un circuit de puissance. La partie logique détermine pour chaque pas quelles sont les bobines alimentées et le sens de rotation. La fréquence de l'horloge du circuit logique détermine la vitesse de rotation. Les moteurs pas à pas sont utilisés en général pour les positionnements angulaires précis (imprimantes, scanners, disques durs ...). De plus, les moteur pas à pas possède un couple à l'arrêt, et contrairement aux moteurs à courant continu, ils fonctionnent en boucle ouverte, c'est-à-dire que dans des conditions normales d'utilisation pour un nombre « n » d'impulsions on obtient un déplacement de « n » pas (Rotation constante pour chaque commande). Donc ils ne nécessitent pas de boucle d'asservissement et sont plus simples à commander. Par ailleurs parmi les inconvénients majeurs des moteurs pas à pas, on a un couple qui décroît rapidement lorsque la vitesse de rotation augmente [W3].

Concernant, le moteur à courant continu classique, il présente un couple massique plus élevé, mais malheureusement aussi un couple résistant plus important. Cependant, certains modèles sont optimisés pour fournir un fort couple à faible vitesse (« moteurs couple » de type plat ou « pan-cake »). Le moteur à courant continu sans fer a l'avantage d'un couple résistant très faible et pratiquement exempt d'ondulation. Son couple spécifique est cependant plus faible ; sa constante de temps thermique réduite oblige à une surveillance étroite de la température d'induit. Le moteur à cloche (type long) à flux radial (ou « faulhaber ») est très utilisé pour les petites puissances, surtout dans les articulations des robots.

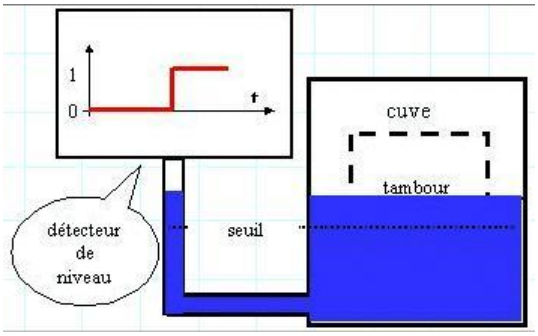
Les moteurs synchrones autopilotés, parfois appelés « moteurs continu sans balais » ou « brushless DC », ont un couple spécifique supérieur aux moteurs à courant continus tout en ne comportant ni collecteur, ni balais. Ces moteurs présentent l'inertie la plus faible de tous les types. Leur couple résistant est plus élevé et moins régulier que celui d'un moteur de type faulhaber (environ 3 % en valeur relative) mais les progrès de la compensation d'ondulation par la commande les rend néanmoins employable dans le contexte exigeant des articulations des robots. Cependant, les meilleures caractéristiques sont en revanche obtenues en utilisant une électronique de commande plus complexe et plus couteuse. L'appellation de ces moteurs (héritée de l'anglais DC brushless motors) est trompeuse. Il ne s'agit en effet pas de moteurs à courant continu, mais de moteurs à courant alternatif [W3].

5. Capteurs

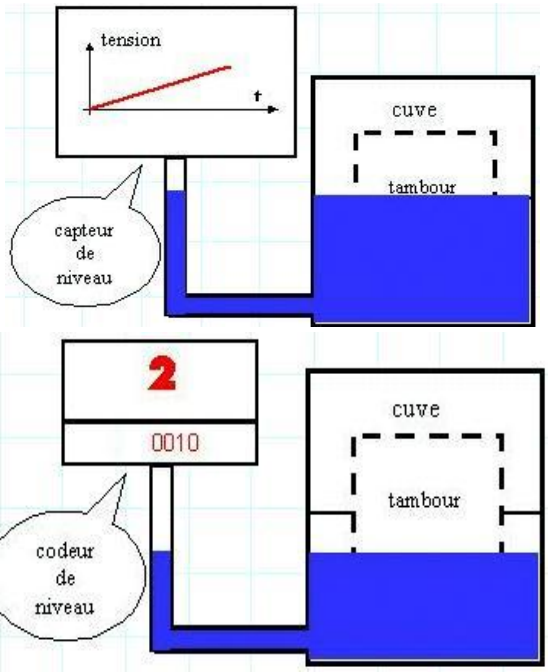
La présence des capteurs augmente les capacités opérationnelles des robots ainsi qu'une amélioration très sensible de ces performances (précision des mouvements …). Les capteurs sont des constituants d'interface entre la partie commande d'un système automatisé et son environnement ou sa partie opérative. Ils peuvent détecter des positions, des pressions, des températures, des vitesses, des accélérations, des niveaux, des présences, …

Tout capteur est composé de deux parties dont l'une est directement sous l'influence de la grandeur à détecter et l'autre est relative à la mise en forme et à la transmission de l'information vers la fonction traitement. De plus, il existe plusieurs types de signaux transmis par les capteurs, à savoir [W11]:

- **Signal Tout Ou Rien :** ce type de capteur peut servir comme un capteur à contacts mécaniques, détecteurs de proximité, détecteur à distance ... qui délivre un signal binaire (0 ou 1) dit tout ou rien. Ce sont les capteurs les plus rependus en automatisation.



Signal analogique: ce type de capteur délivre un signal analogiques (tension ou courant) en traduisant les valeurs de postions, de pressions, de températures…



- **Signal numérique : c**'est capteurs appelés codeurs, transmettent des valeurs des positions, des pressions,..., en signal numériques [W11].

De plus il existe d'autres types de capteurs à savoir :

- **Détecteurs de position :** qui sont des capteurs mécaniques de position (ou interrupteurs de position). Ils sont surtout employés dans les systèmes automatisés pour assurer la détection de position. On parle aussi de détecteurs de présence. Ils sont en général utilisés dans la mécanique et les machines à outils (usinage, manutention, levage, ...), dans l'agro-alimentaire et la chimie.
- **Détecteurs de proximité inductive :** sont des capteurs qui ne nécessitent aucun contact physique avec l'objet à détecter et sont très répandus dans l'industrie du fait de leur très grande robustesse, leur bonne tenue dans les environnements industrielles (atmosphère polluante, température…) et leur possibilité de détecter des objets fragiles …. Leurs fonctionnements dépendent de l'épaisseur des objets (jusqu'à 50mm). Ils sont le plus souvent utilisés dans la machine-outil, la robotique, la chimie fine, et dans les domaines d'application d'usinage.
- **Détecteurs de proximité capacitive :** sont basés sur la variation d'un champ électrique à l'approche d'un objet quelconque. Ainsi, l'objet est à proximité du capteur sans le touché. Ils sont surtout utiliser pour le contrôle de remplissage de liquides des flacons ou des cuves et pour la détection de la présence de matériaux pulvérulents dans des trémies.
- **Détecteurs de proximité photo électrique :** sont basés sur l'association d'un émetteur de lumière et d'un récepteur photosensible. Parmi ces grande avantage il permet de détecter toutes forme d'objet et de toute nature de à très grande distance (dépend du système employé). De plus, l'utilisation de la lumière infrarouge invisible le rend indépendant des conditions d'environnement. Ces types de détecteurs sont surtout utilisés pour la détection des pièces, des objets, des personnes ou même des animaux.
- **Interrupteurs à lame souple (magnétique) :** sont des capteurs constitués d'un boitier qui contient un contacte électrique métallique souple sensible aux champs magnétiques. Lorsqu'un champ est dirigé vers la face sensible du capteur le contacte se ferme. Parmi les avantages de ce capteur, on a la possibilité de détecter des objets fragiles et qui nécessite aucun contact physique avec l'objet à détecter. De plus, il fonctionne avec tout objet magnétique.

6. CONCLUSION

Dans ce chapitre, on a décrit les robots d'une façon générale, leurs structures, leurs commandes, ainsi que les différents types. Dans le prochain chapitre, une description détaillée du bras manipulateur didacticiel KSR10 fera l'objet de notre étude.

Chapitre II: Description matérielle et logicielle

1. Introduction

Le robot est une machine capable d'effectuer automatiquement un certain nombre d'action en exécutant les mouvements décrits par le programme embarqués sur le robot. Ce programme est développé par des méthodes manuelles ou automatiques. Les méthodes manuelles rendent l'utilisation de ces robots difficile car chaque tache doit être décomposée en plusieurs sous taches élémentaires ce qui nécessite trop de calculs. Dans le cas de la méthode automatique, les programmes sont générés automatiquement car ils résultent d'une interaction entre l'operateur et le robot.

L'objet de ce chapitre est de décrire tout d'abord les éléments principaux constitutifs de notre projet, à savoir le bras manipulateur KSR10, la partie matérielle de la commande (le choix du matériel tel que l'arduino et son utilisation). Ensuite, nous aborderons les différents logiciels que nous utiliserons pour programmer la carte de commande et pour terminer nous allons présenter et réaliser l'interface graphique 3D qui affiche et traduit les ordres de commande entre le Joystick et les moteurs du robot.

2. Bras manipulateur KSR10

Le KSR10 est un petit bras manipulateur à Cinq Axes (Cinq degrés de libertés). Ces mouvements sont assurés par cinq moteurs à courant continu (une pince et quatre articulations). Ce bras robotisé peut soulever des poids allons jusqu'à 100g et arrive à atteindre une hauteur de 38cm [W1].

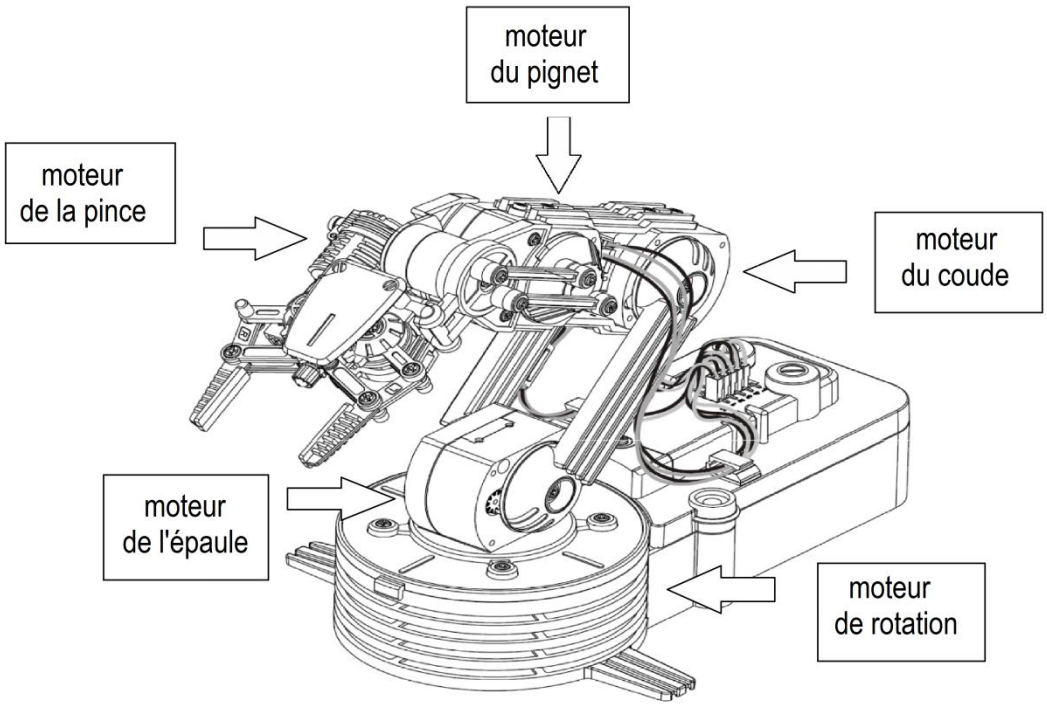


FIGURE 2.1 – Bras manipulateur KSR10 [W1]

Le bras possède quatre articulations de rotation, que je vais appeler : la base, l'épaule, le

coude et le poignet. La distance entre les articulations est représentée en millimètres sur la figure 2.2. La base fait tourner le bras autour de l'axe vertical z, tandis que les trois autres tournent autour de l'axe des x. Aucune articulation tourne autour de l'axe des y, ce qui limite les mouvements du bras, cependant il rend le contrôle beaucoup plus facile. Chaque articulation a une limite de rotation dont les directions avant et arrière pour le poignet, le coude et l'épaule, et de gauche et à droite pour la base comme le montre la figure 2.3.

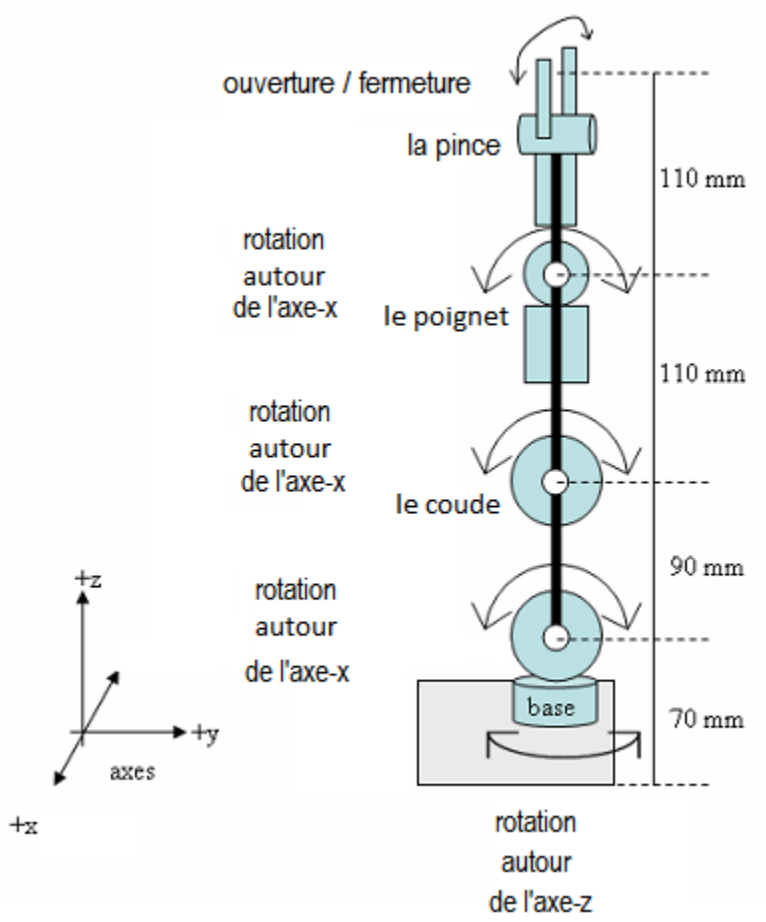


FIGURE 2.2 – Schéma des articulations du bras[NU11].

La pince du bras s'ouvre et se ferme via la rotation des roues dentées, ses dents sont relié aux roues de manière à maintenir les griffes de la pince parallèle les uns aux autres lorsqu'ils se déplacent. La figure 2.3 montre les degrés de liberté de l'articulation du bras le long de l'axe Y.

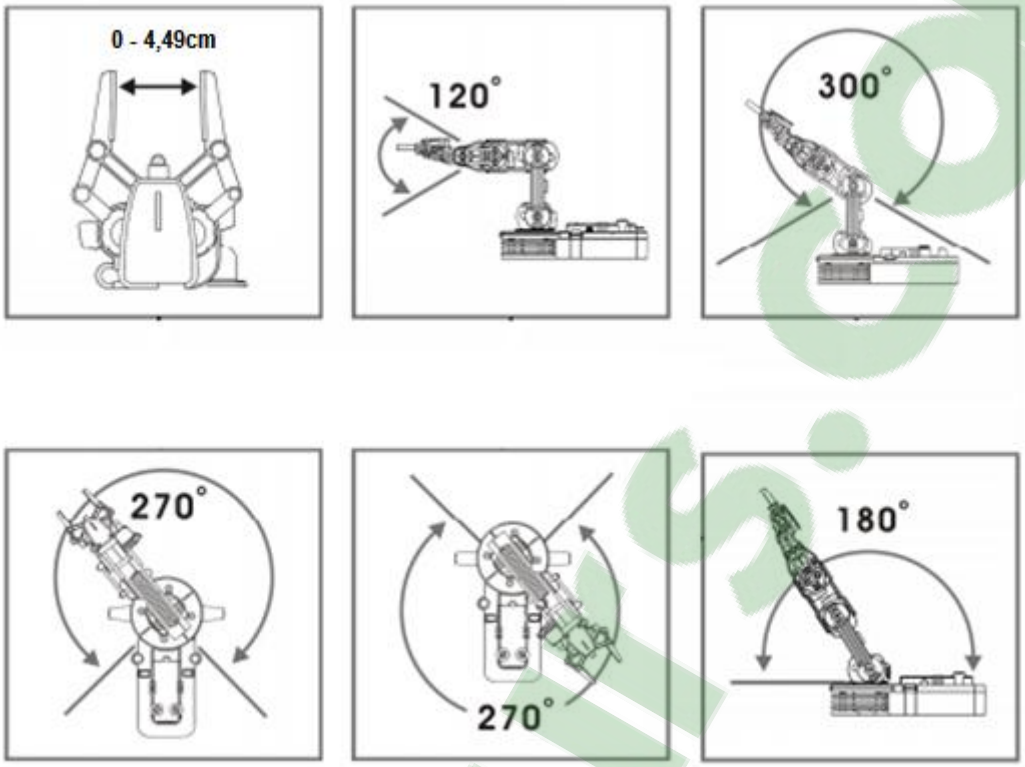


FIGURE 2.3 – vue le long de l'axe Y [W1].

Ce bras n'est pas un robot très précis. La vitesse de rotation n'est pas constante si le bras prend un mouvement au sens même ou opposé du champ gravitationnel. Par exemple si l'épaule tourne vers le bas elle mettra moins du temps que de tournés vers le haut. De plus, ce bras manipulateur à des roues d'engrenage qui sont appelées réducteurs qui servent à réduire la vitesse de rotation de l'axe des moteurs tout en augmentant le couple de sortie.

Le bras est muni de six petits moteurs à courant continu, quatre d'entre eux sont destinés à mouvoir une articulation et le cinquième pour ouvrir et fermer la pince. Donc ces moteurs sont le muscle de ce bras qui sert à convertir l'énergie électrique en énergie mécanique. Les moteurs utilisés dans ce bras sont des moteurs à aimants permanents alimentés par une tension de 3V générés par deux piles de type LR20C. Ces moteur a aimant permanents présentent une très faible inertie mécanique. Ils sont très bien adaptés aux applications qui nécessitent des variations rapides de vitesse de rotation ou une commande en position. Ils sont généralement commandés par un ensemble électronique comportant une alimentation de puissance avec une électronique de commande réalisant un asservissement. Le moteur à courant continu est composé de deux parties principales : le stator (inducteur) et le rotor (induit). Le rotor est composé de fils de cuivre enroulés sur un support lui-même monté sur un axe, constituant plusieurs bobines, toujours en nombres impairs, l'alimentation de ces bobine est assurés par des balais souples placer sur le collecteur. Lorsque l'une des bobines reçoit le courant continu, elle crée un champ magnétique qui attire les pôles des aimants placé sur la carcasse du stator, créant par cela une rotation de l'axe du moteur [An 11].

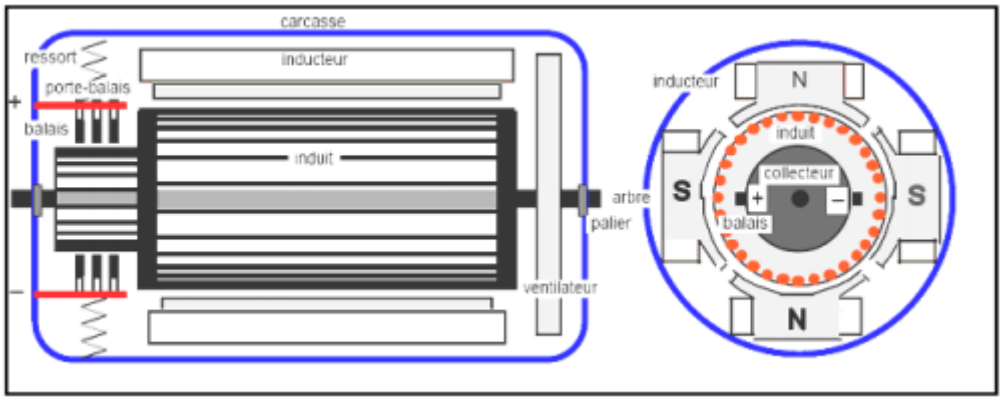


FIGURE 2.4 – Éclaté d'un MCC à aimant permanent

3. Carte Arduino

L'arduino est une plateforme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur, et un logiciel, véritable environnement de développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. C'est un outil qui peut capter et contrôler d'avantage de choses du monde matériel que votre ordinateur de bureau. L'arduino peut être utilisé pour développer des objets qui interagissent avec le milieu qui les entoure, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien communiquer avec des logiciels tournant sur votre ordinateur. Le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant plusieurs avantages pour les enseignants, les étudiants et les amateurs intéressés par les autres systèmes, tel que:

- Le prix : Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes.
- Le Multi-plateforme : Le logiciel Arduino, est écrit en Java et est disponible sous les systèmes d'exploitation Windows, Macintosh et Linux, alors que La plupart des autres systèmes à microcontrôleurs sont limités à Windows.
- Un environnement de programmation simple et clair : L'environnement de programmation Arduino (le logiciel Arduino) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- Le logiciel est Open Source et extensible : Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés.
- Le matériel est Open source et extensible : Les cartes Arduino sont basé sur les microcontrôleurs Atmel ATMEGA8, ATMEGA168, ATMEGA 328, ... Les schémas des modules sont publiés sous licence open source et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et/ou en les améliorant[W9].

De plus le système Arduino offre les fonctionnalités suivantes :

- Communication simple par le port USB
- Terminal texte côté PC qui permet d'afficher des nombres et du texte pendant l'exécution d'un programme permettant une mise au point efficace.
- Calculs sur valeur entière et sur les nombres à virgule. Fonctions math.
- L'utilisation simplifiée de chaines de caractères.
- Communication avec des modules externes.
- La conversion analogique/numérique sur plusieurs voies qui va permettre d'utiliser des capteurs de mesure analo
- giques variés,
- Utilisation possibles des interruptions.
- Nombreuses cartes complémentaires, appelées shields, permettant d'ajouter des fonctions avancées au montage (éthernet, LCD, écran TFT, GPS, SD-CARD, etc...).

La carte Arduino que nous avons utilisé est la carte arduino Uno. Il est basé sur un microcontrôleur ATMEL de référence ATMega328. Ce dernier a une mémoire flash de 32Ko pour stocker le programme (dont 0.5Ko également utilisés par le bootloader). Il possède également 2ko de mémoire SRAM (volatile) et 1Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM) [W9].

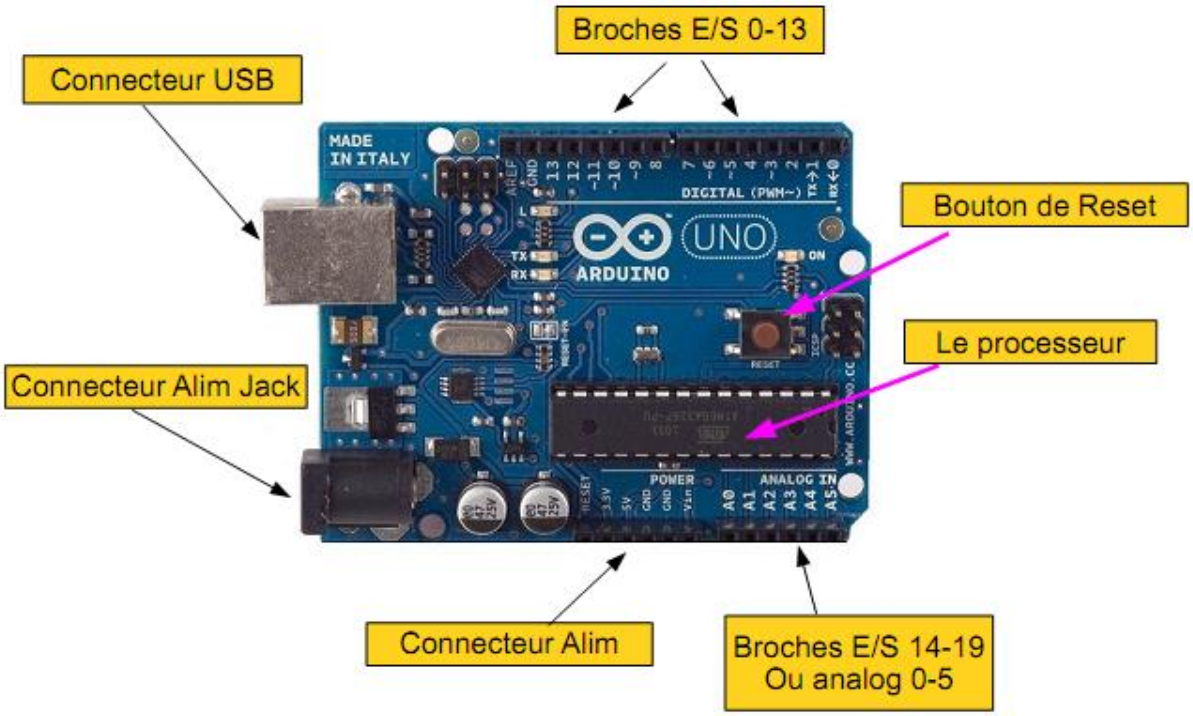


FIGURE 2.5 La Carte Arduino Uno.

Cette carte dispose de:

- 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- un quartz 16Mhz,
- une connexion USB,
- un connecteur d'alimentation jack,
- un connecteur ICSP (programmation "in-circuit"),
- et d'un bouton de réinitialisation (reset).

Elle contient tout ce qui est nécessaire pour le fonctionnement du microcontrôleur ; Pour pouvoir l'utiliser et se lancer, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB ou de l'alimenter avec un adaptateur secteur ou une pile. Les caractéristiques de notre carte arduino peuvent être regroupées dans le tableau suivant :

Microcontrôleur	ATmega328
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	14 (dont 6 disposent d'une sortie PWM)
Broches d'entrées analogiques	6 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA (ATTENTION : 200mA cumulé pour l'ensemble des broches E/S)
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée - 500 mA max si port USB utilisé seul
Mémoire Programme Flash	32 KB (ATmega328) dont **0.5 KB** sont utilisés par le bootloader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

Tableau 2.1. Caractéristiques de la carte Arduino [W9].

Concernant les broches numériques : Chacune des 14 broches numériques de la carte UNO (numérotées des 0 à 13) peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions pinMode(), digitalWrite() et digitalRead() du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité et dispose d'une résistance interne de

"rappel au plus" (pull-up) (déconnectée par défaut) de 20-50 KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite (broche, HIGH). De plus, certaines broches ont des fonctions spécialisées, tel que :

- La communication Série : Broches 0 (RX) et 1 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données sérient 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.
- L'interruption Externe : Broches 2 et 3. 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. Voir l'instruction attachInterrupt() pour plus de détails.
- Impulsion PWM (largeur d'impulsion modulée): Broches 3, 5, 6, 9, 10, et 11. fournissent une impulsion PWM 8-bits à l'aide de l'instruction analogWrite().
- SPI (Interface Série Périphérique): Broches 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Mega.
- I2C: Broches 4 (SDA) et 5 (SCL). Supportent les communications de protocole I2C (ou interface TWI (Two Wire Interface - Interface "2 fils"), disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils") .
- LED: Broche 13. Il y a une led incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la led est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.
- Voir également : Correspondance entre les broches de l'Arduino et les ports de l'ATmega168.

Concernant les broches analogiques : La carte Uno dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c'est à dire sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre 0V (valeur 0) et 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analogReference() du langage Arduino. Ces broches analogiques peuvent être utilisées en tant que broches numériques si elles sont numérotées en tant que broches numériques de 14 à 19 [W10].

Concernant les autres broches : Il y a deux autres broches disponibles sur la carte, à savoir :
AREF : Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference().
Reset : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

Concernant la PWM sur arduino utilise une technique qui permet d'obtenir des effets d'allure analogique avec des broches numériques grâce à la fonction analogWrite (). Le contrôle numérique est utilisé pour créer une onde carrée, un signal basculant entre un

niveau HAUT et un niveau BAS (0V et 5V). Cette succession de niveaux HAUT/BAS peut simuler des tensions entre le niveau HAUT (5 Volts) et le niveau BAS (0 Volts) en faisant varier la proportion du temps où le signal est HAUT sur la proportion de temps où le signal est BAS. La durée du temps du niveau HAUT est appelé largeur d'impulsion, ou encore "duty cycle". Pour obtenir une variation analogique, il suffit de changer ou de modifier cette largeur d'impulsion. Sur l'arduino Uno la fonction analogWrite () fonctionne sur les broches 3, 5, 6, 9, 10 et 11 avec une fréquence d'environ 980 Hz. Dans le graphique ci-dessous, les lignes vertes représentent une période de temps régulière. Un appel de la fonction analogWrite(valeur) utilise une valeur comprise entre 0 et 255, tel que analogWrite(255) utilise 100% du cycle (toujours au niveau HAUT), et analogWrite(127) utilise 50% du duty cycle (la moitié du temps) par exemple. La valeur 0 correspond ainsi à 0% du duty cycle[W3].

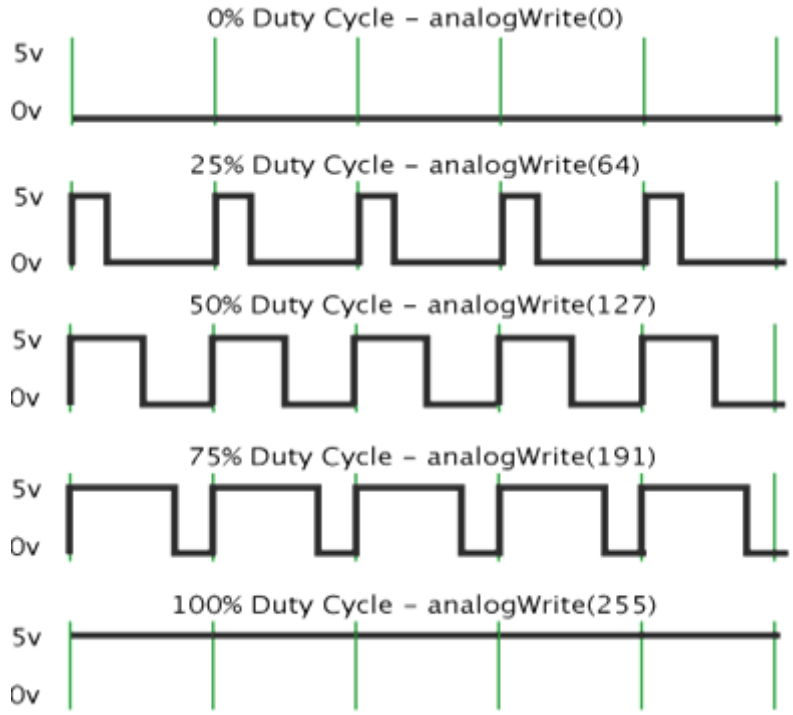


FIGURE 2.6 PWM

4. Environnement de programmation

4.1 Logiciel Arduino

Le logiciel Arduino est un espace de développement intégré (EDI) dédié au langage Arduino et à la programmation des cartes Arduino. Ce logiciel a pour fonctions principales de [W9]:

- pouvoir écrire et compiler des programmes pour la carte Arduino.
- se connecter avec la carte Arduino pour y transférer les programmes.
- communiquer avec la carte Arduino.

Le logiciel Arduino comporte :

- une **BARRE DE MENUS** comme pour tout logiciel avec une interface graphique.
- une **BARRE DE BOUTONS** qui permet un accès direct aux fonctions essentielles du logiciel ce qui fait toute sa simplicité d'utilisation.
- un **EDITEUR** (à coloration syntaxique) pour écrire le code de programme, avec onglets de navigation.
- une **ZONE DE MESSAGES** qui affiche et indique l'état des actions en cours.
- une **CONSOLE TEXTE** qui affiche les messages concernant le résultat de la compilation du programme [W9].

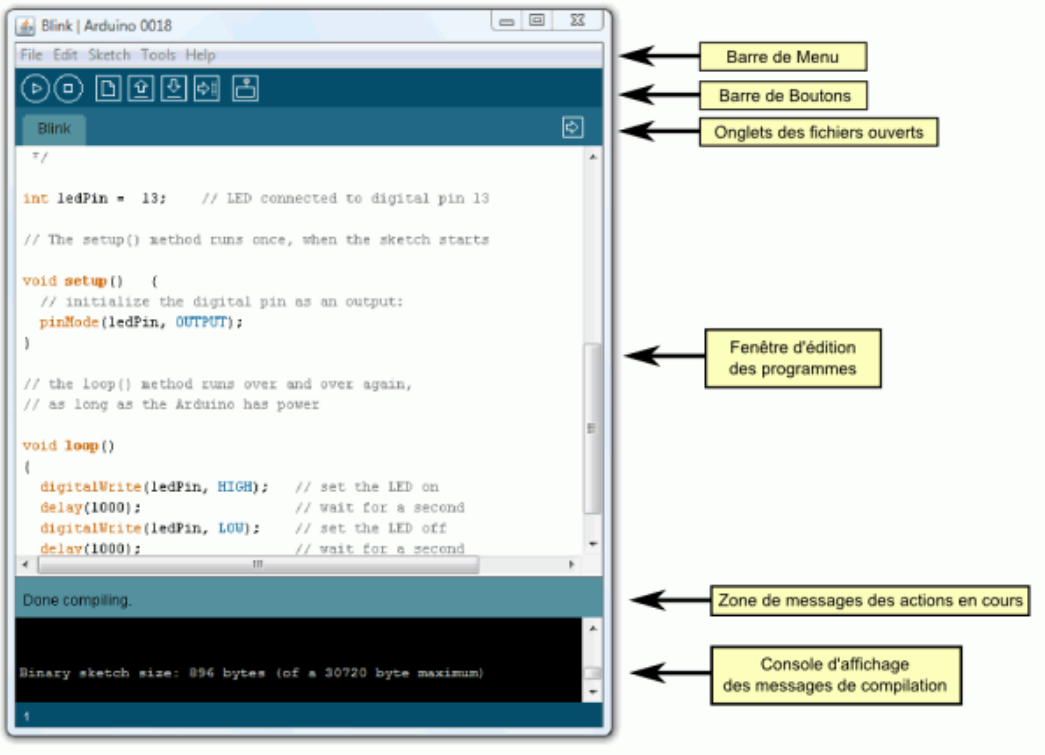


FIGURE 2.7.Présentation de l'EDI

4.2 Langage de programmation Processing

L'interface graphique est très utile pour pouvoir debugger facilement, prendre contrôle du robot et afficher des informations (position, niveau de batterie, messages d'erreurs ...). Bien que de nombreux environnements et outils de programmation sont en concurrence pour animer les robots. Le langage de programmation utilisé pour notre projet est le Processing. La raison de ce choix est fort simple. La majeure partie de notre travail consiste à créer une interface graphique 3 dimensions. Processing nous procure un environnement de design très efficace permettant de réaliser ces interfaces sur Windows, Mac os x et Linux avec un minimum d'efforts et de temps. De plus, il propose à la fois un environnement de création complet et un ensemble de fonctionnalités supplémentaires qui enrichissent les possibilités du logiciel. Il permet de dessiner, réaliser des animations en 2 ou 3 dimensions, créer des œuvres

sonores et visuelles et concevoir des objets électroniques qui interagissent avec leur environnement. Des dizaines de milliers d'artistes, de designers, d'architectes, de chercheurs l'utilisent pour réaliser des projets incroyables dans de multiples domaines tels que les vidéos clips, dessins animés, Publicités, Visualisation de données scientifiques ... [W].

Ce logiciel a été conçu en 2001 au laboratoire Aesthetics + Computation Group (ACG) du MIT Media Lab par Ben Fry et Casey Reas en 2001. Processing fait partie des principaux environnements de création utilisant le code informatique pour générer des œuvres multimédias sur ordinateur. L'interface d'utilisation de Processing est composée de deux fenêtres distinctes : la fenêtre principale dans laquelle vous allez créer votre projet et la fenêtre de visualisation dans laquelle vos créations (dessins, animations, vidéos) apparaissent [W].

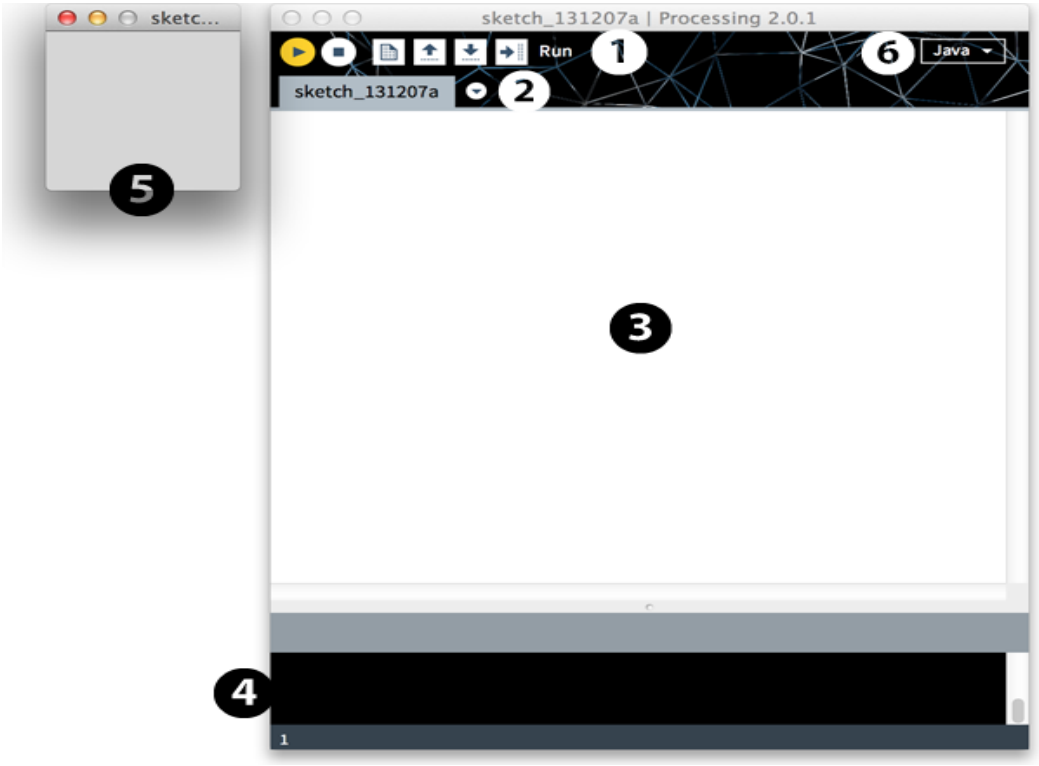


FIGURE 2.8. Interface d'utilisation de Processing [W].

On trouve plus précisément les éléments suivants dans l'interface :

1. Barre d'actions
2. Barre d'onglets
3. Zone d'édition (pour y saisir le programme)
4. Console (destinée aux tests et messages d'erreur)
5. Fenêtre de visualisation (espace de dessin)
6. Liste déroulante pour les modes

De plus, nous avons les boutons :

- "Run" : pour exécute votre sketch (votre programme).
- "Stop" : pour arrête l'exécution de votre sketch.
- "New" : pour Crée un nouveau sketch.
- "Open" : pour ouvre un sketch existant.
- "Save" : pour sauvegarde le sketch en cours.
- "Export" : pour exporte le sketch en application (en mode Java).

6. CONCLUSION

Dans le cadre de ce chapitre, on a pu décrire les différents éléments constitutifs de notre bras manipulateur KSR10, en expliquons les type des moteur. Nous avons également montré les caractéristiques de la carte Arduino que nous avons utilisé pour l'utiliser comme unité de commande pour notre système. Pour terminer nous avons présentés le logiciel utilisé pour l'interface graphique.

Chapitre III: Réalisation matérielle et logicielle

1. Introduction :

Dans ce chapitre nous présentons les différente étapes suivie pour la réalisation de notre projet qui consiste a créé une interface graphique pour commander le bras manipulateur KSR10 à travers une carte arduino UNO comme le montre la figure suivante :

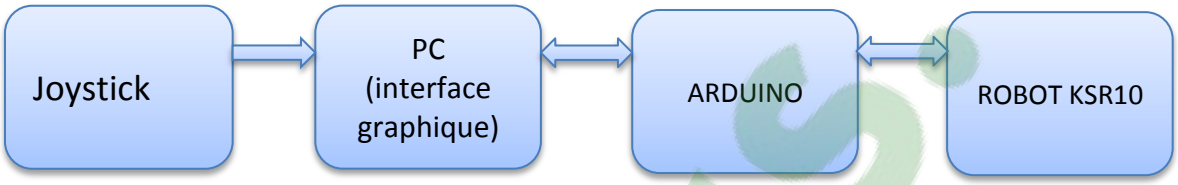


Figure III.1: Schémas synoptique simplifié du système

Le but de ce projet est de réaliser les fonctionnalités comportant :

- L'utilisation d'une interface de commande de type joystick pour la manipulation du bras.
- Visualisation du mouvement du bras manipulateur en trois dimensions.
- Visualisation des valeurs d'entrées / sorties sur le PC (valeurs fournies par le joystick et valeurs transmises par le port RS232 entre le PC et la carte ardiuno).
- Enregistrement et lancement de trois différentes trajectoires.
- L'utilisation d'une carte aduino pour traduire les commande entre PC et robot via le port RS232.

Le schéma synoptique détaillé est représenté par la figure suivante :

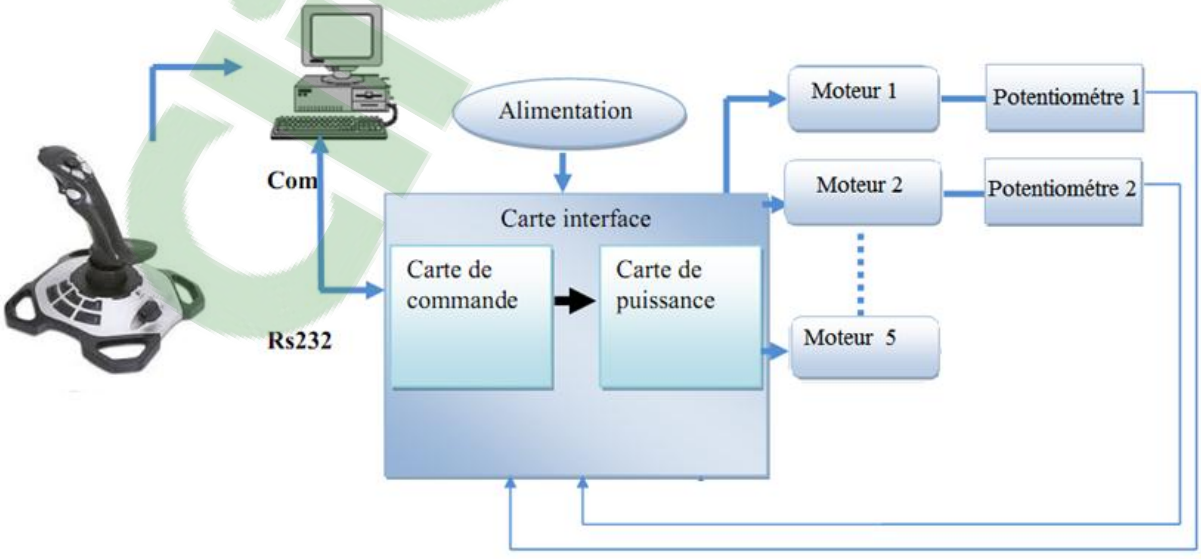


Figure III.2: schéma synoptique du système

À partir du joystick, on transmet une information vers le PC via le port USB, et grâce au logiciel 'Processing' et l'information du joystick, nous pourrons commander et enregistré les mouvements du robot via la carte arduino, les sorties de notre carte arduino ne peuvent être connectées avec notre bras manipulateur qu'à travers une carte de puissance dont ce dernier est connecté à l'actionneur de chaque articulation.

2. Description des différentes parties

A partir de ce qui a été décrit, nous allons présenter en détail chaque partie de notre projet, ainsi nous allons voir :

2.1 Interfaces de commande (joystick)

Le joystick est un périphérique informatique constitué de bouton et d'une manche posé sur un socle. L'utilisateur peut agir sur le périphérique soit en bougeant le manche dans une direction, soit en pressant sur les boutons[W12].

Dans notre projet on a utilisé un joystick nommé « Logitech Force 3D Pro » comme interface de commande, c'est un joystick standard qui a des bonnes caractéristiques en sensibilité et précision.

Figure III.3: Logitech Force 3D Pro

Concernant les boutons numérotés de 1 à 12, nous avons un signal tout ou rien. Par contre pour le bouton nommé HAT, nous avons un signal impulsionnel. En ce qui concerne le bras de joystick, ce dispositif offre un signal analogique.

Les données envoyées par ce périphérique sont brutes et correspondent aux déplacements des axes perçus par les capteurs, et aux différentes actions des boutons. Dans notre cas on a pris des signaux de type toute ou rien, analogique et impulsionnel pour piloter les différentes articulations du bras.

2.2 Réalisation de l'interface graphique :

Les interfaces graphiques qui aident à la maitrise globale d'un système robotique sont souvent indispensable ou même presque exclusivement réservés à des spécialistes. En particulier, ces outils qui permettent l'abordage des aspects de simulation ou de programmation hors-ligne de robots restent d'utilisation complexe. Ainsi, La modélisation d'un nouveau robot nécessite d'habitude un investissement en temps important, ou généralement la définition de tâches n'est pas envisageable pour un non spécialiste. Notre projet fait l'objet de réalisé une interface graphique et qui comblent les inconvénients présentés précédemment. Cette interface graphique, représentée dans la Figure III.4 et développée grâce au logiciel de programmation « Processing » , dispose :

- d'un simulateur en trois dimensions qui représente un bras manipulateur
- d'une zone pour afficher les valeurs du Joystick.
- d'une zone affichant les valeurs envoyées et reçue des différents angles du robot à travers le port RS232.
- des boutons pour enregistrés et lancer des différent trajectoires.

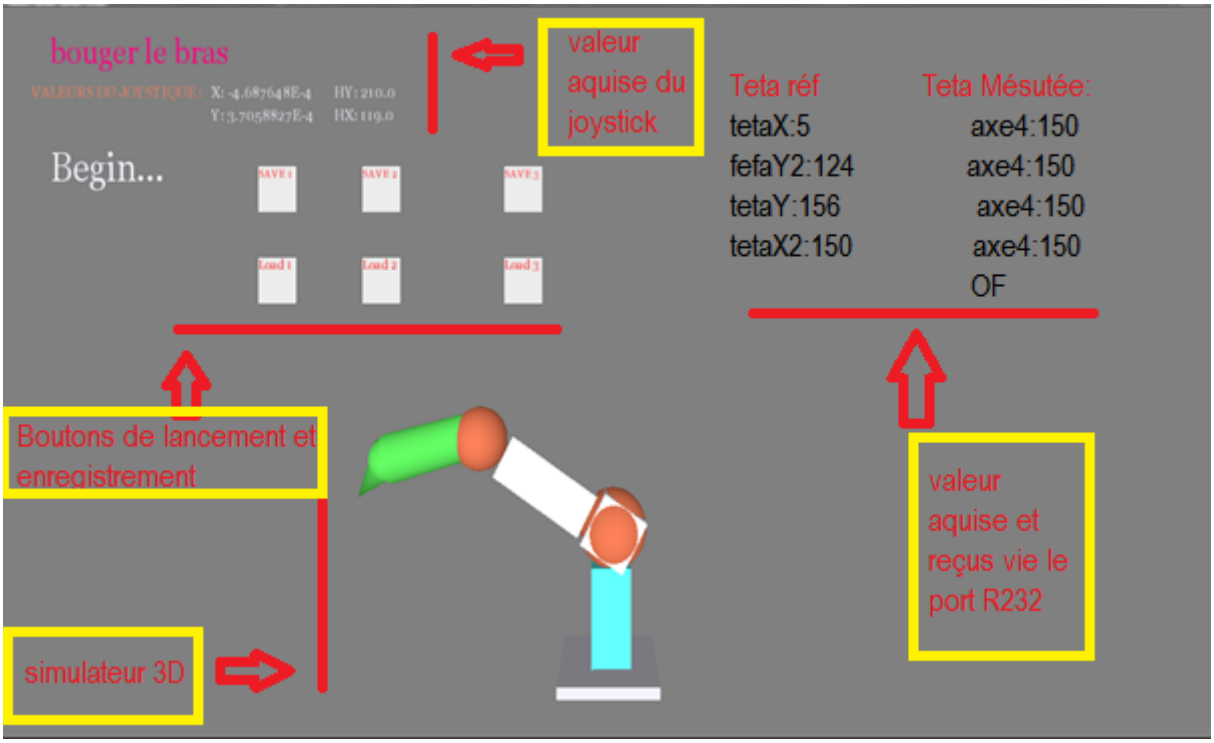


Figure III.4: Interface graphique

L'exécution de l'interface graphique est régie par l'organigramme donnée par la Figure (III.5) ete qui résume les principales étapes à respecter lors de l'exécution de l'algorithme de l'interface de commande.

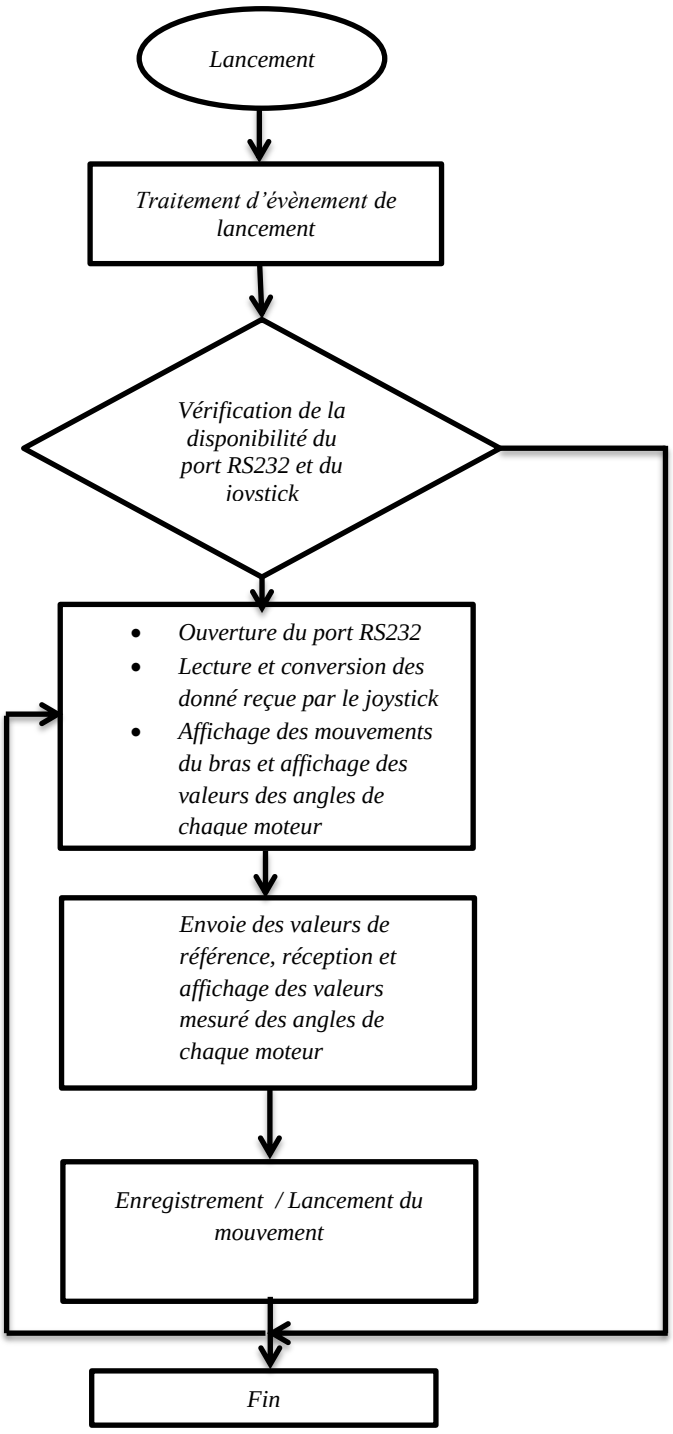


Figure III.5 : Organigramme d'interface de commande

En lançant le programme par l'appui sur le bouton 'RUN', la première étape sera la vérification de la disponibilité du port RS232 et du joystick, l'absence d'un de ces éléments ne permet pas d'exécuter le système. Dans le cas contraire, le programme va établir une connexion avec le RS232 et la connections avec le joystick pour récupérer et convertir les données acquises par ce dernier qui sont affichée et utilisée pour la manipulation du bras manipulateur au niveau de notre interface graphique, nous devons passer par l'enregistrement des trajectoires (action sur un des boutons 'SAVE1, 2 ou 3') afin de mieux Contrôler le bras manipulateur réelle. La deuxième étape consiste au lancement du bras manipulateur réel en actinons sur les boutons 'LOAD1,2 ou 3'

Notre programme développé va nous permettre aussi :

- d'obtenir les valeurs du joystick et leur convertir en degrés.
- de communiquer les valeurs en degrés de chaque articulation via RS232.
- de créer des fichiers en format ' .Txt' comportant les coordonnées des trois différentes trajectoires enregistrées.

2.3. Communication RS232 :

La communication RS232 présente un certain nombre d'atout, tout d'abord elle est présente sur la plupart des pc. Du faite de sa simplicité :

De plus, cette liaison est également utilisé dans les microcontrôleurs, car elle répond très bien au besoin les plus courants (paramétrage, récupération de donnés, etc..).Pour qu'une transmission se fasse sur le port RS232, il est impératif que l**es** transmissions des informations numériques ont:

a) des informations à transmettre qui sont envoyées par l'émetteur sur la ligne Tx, vers le récepteur (ligne Rx) bit par bit.

b) des vitesses de transmission (exprimer en BAUDS) de l'émetteur qui doivent être identique à la vitesse d'acquisition du récepteur.

c) des communications qui peuvent se faire en:
   - simplex, si nous voulons un sens.
   - half-duplex, si nous voulons deux sens, une émission d'abord, puis une réception.
   - full-duplex, si nous voulons une émission et une réception simultanées.

Dans notre cas nous avons opté pour une communication full duplex qui répond à nos besoins.

2.4. Réalisation d'une carte d'interface

Notre carte d'interface se compose de deux parties : une partie commande comportant l'arduino uno et une partie puissance, nécessaire pour piloter les actionneurs et joue un rôle d'amplificateur et d'isolement galvanique.

2.4.1. Partie commande

Grace à la carte arduino qui a servi comme moyen de traduction des valeurs de moteur vers des impulsions de commande de chaque moteur du bras, elle reçoit aussi les valeurs de chaque articulation grâce à la communication série RS232.
On peut dire que la carte auduino gère le fonctionnement de chaque moteur de bras KSR10 en gérant l'asservissement de chaque moteur grâce à ces entrés analogique lié aux potentiomètres qui servent de capteur de position de chaque articulation.

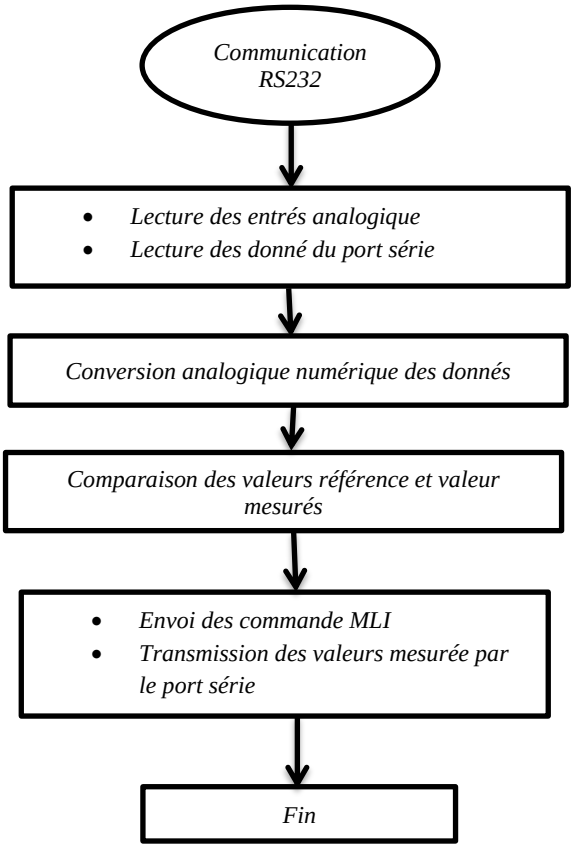


Figure III.6 : Organigramme de fonctionnement de la carte de commande

Le programme flashé dans la carte arduino s'exécute suivant les étapes montrées sur l'organigramme représenté dans la figure III.6. Le fonctionnement de la carte arduino se résume en :

- l'établissement de la communication RS232.
- la lecture des donné du port RS232 et des entrées analogiques (de A0 jusqu'a.A3).
- la conversion analogique numérique des données.

- la comparaison et la correction des erreurs de position (asservissement).
- Commande des différents actionneurs du bras du robot à travers la carte de puissance et transmission des valeurs mesurés par le RS232 vers le PC.

2.4.2. Partie puissance

La carte de puissance est alimentée par une tension de 5V/0,6A fournie par une alimentation externe. L'élément central constituant cette carte de puissance est composé :

- d'un hacheur L293 ; qui sert à alimenter les actionneurs du bras en fonction du signal générés par l'arduino,
- d'un inverseur '74LS04' pour permettre à notre arduino qui est limité en sortie de contrôler les sens de rotation des actionneurs
- d'un optocoupleur pour protéger notre carte de commande arduino.

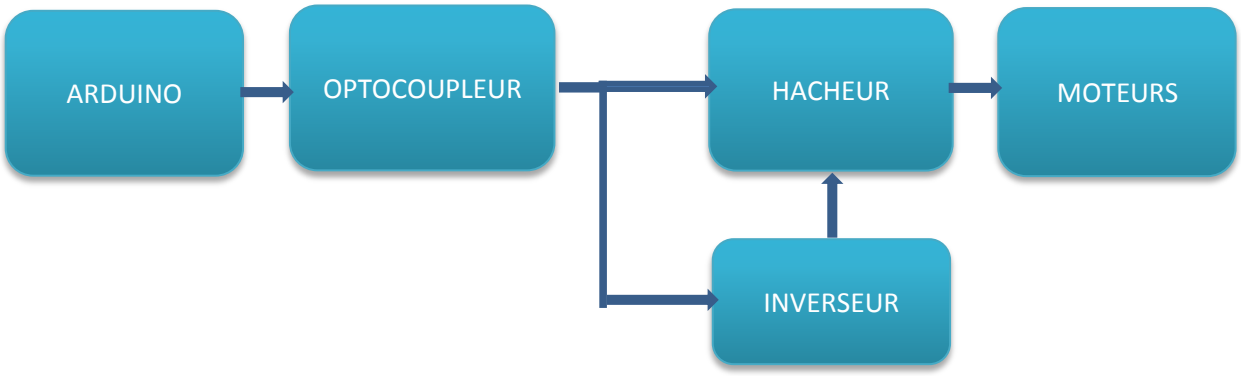


Figure III.7 : Schéma fonctionnel de l'ensemble (commande et puissance)

La Figure III.8 montre l'utilisation de l'inverseur 74LS04 car la carte arduino ne dispose pas des sorties nécessaires suffisantes pour gérer les sens de rotation tous les moteurs.

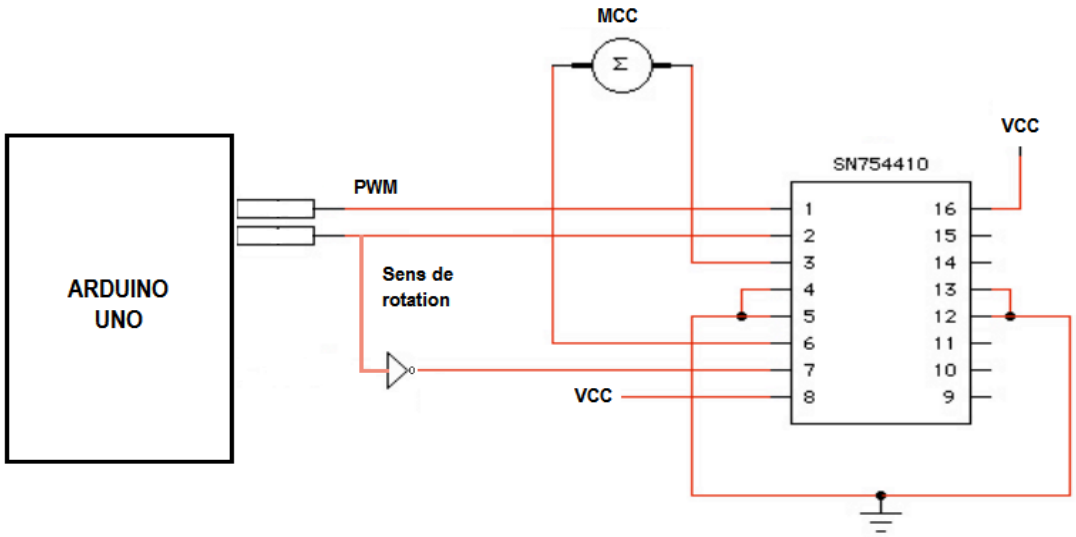


Figure III.8 : commande d'un moteur à C.C dans 2sens de rotation

La figure III.9 reproduite par le logiciel ISIS représente l'une des trois cartes de puissance:

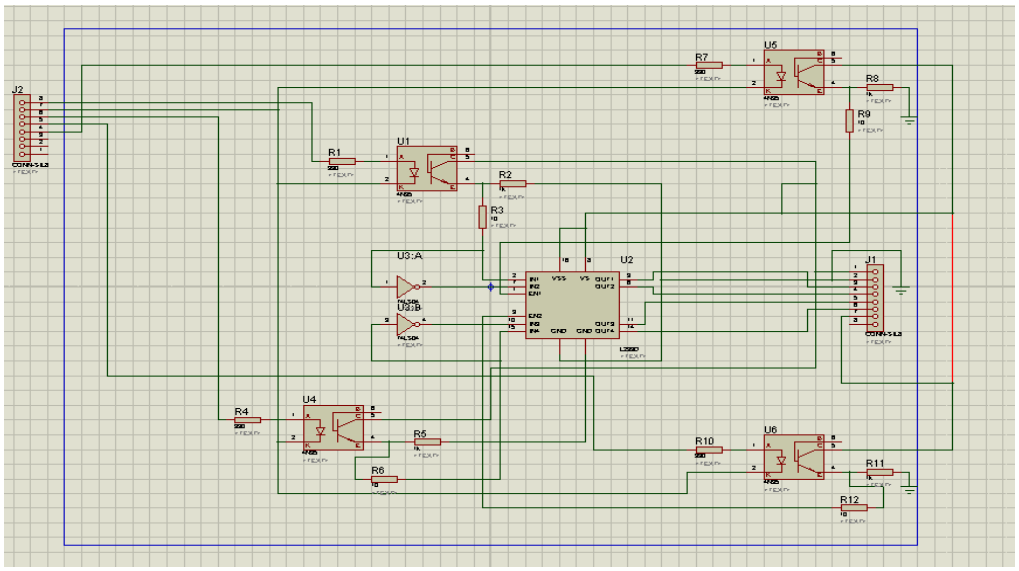


Figure III.9: la carte de puissance dans logiciel ISIS

La figure III.10 représente l'une des trois cartes de puissance réalisée pratiquement

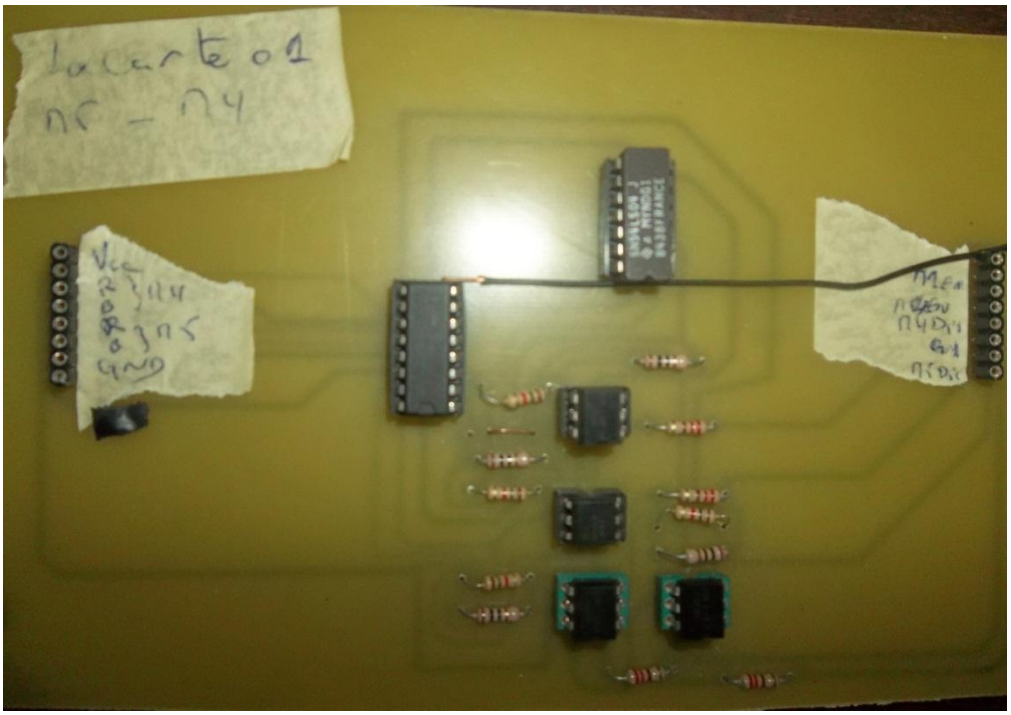


Figure III.10 : Carte de puissance réalisée

3. Commande et Asservissement:

Pour commander les cinq moteurs du bras manipulateur KSR10, on a réalisé deux types de commande, la commande en boucle ouverte et la commande en boucle fermé (commande par les angles des organes du robot) .La commande en boucle fermée repose sur la mesure de la position angulaire de chaque articulation. Le dispositif assurant cette tâche est le potentiomètre rotatif linéaire ou à partir d'un mouvement relatif. Pour connaître la position des axes, on a utilisé des simples potentiomètres angulaire, car ils ont l'avantage d'être bon marché, plutôt faciles à installer et en plus ils fournissent une information absolue sur la position. Le seul inconvénient est qu'ils risquent de s'user avec le temps.la plupart des potentiomètres angulaires ont une étendue de mesure inférieure ou égale à 360°.

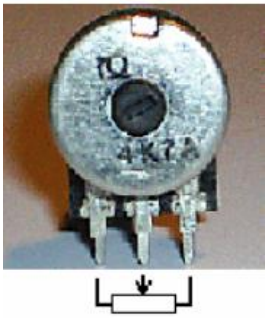


Figure III.11 : potentiomètre angulaire

Bien que on n'a pas une bonne stabilité du système on a eu recoure à l'utilisation d'un correcteur PI, le Proportionnelle qui est ni plus ni moins que la différence entre notre consigne et la valeur actuelle, multiplié par un coefficient de proportionnalité.

Donc la commande de ce régulateur est proportionnelle à l'erreur, mais aussi proportionnelle à l'intégrale de l'erreur.

Commande = Kp * erreur + Ki * somme erreurs

Ki est le coefficient de proportionnalité de la somme des erreurs qu'on a réglé de façon manuelle.

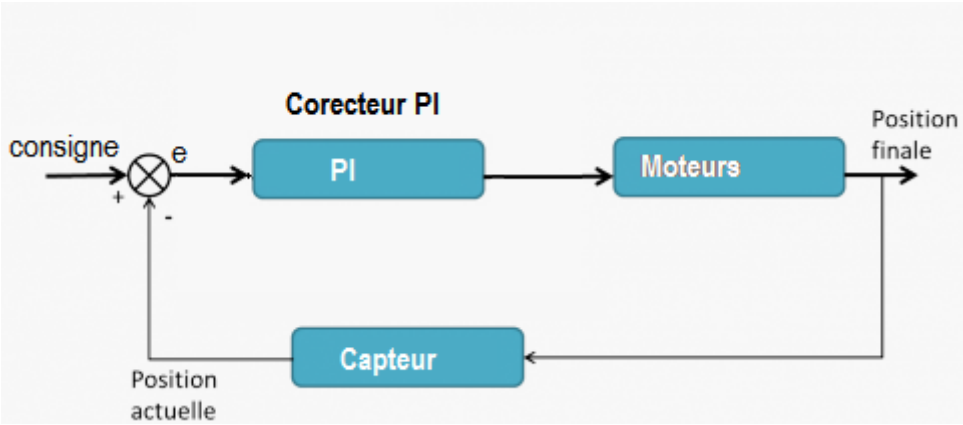


Figure III.12 : Schéma block de la boucle de régulation

4. Sortie PWM :

En utilisant une PWM sur la broche d'activation des hacheurs, on sera en mesure de faire varier la vitesse des moteurs. Le rapport cyclique est défini par un nombre allant de 0 à 255. Cela signifie qu'à 0, le signal de sortie sera nul et à 255, le signal de sortie sera à l'état HAUT.

Pour savoir quel rapport cyclique correspond avec quelle valeur, il faut faire une règle de trois comme le montre le tableau suivant :

Valeur argument	Rapport cyclique (%)
0	0
127	50
255	100

Le calcul donnant la valeur pour chaque portion est défini par cette relation :

$$\text{argument} = \frac{x * 100}{255}$$

Le contrôle de ce rapport cyclique s'applique dans la carte arduino par l'instruction suivante :

analogWrite(Pin, argument); exemple d'un signal PWM généré par l'arduino pour un argument de 127 (50%)

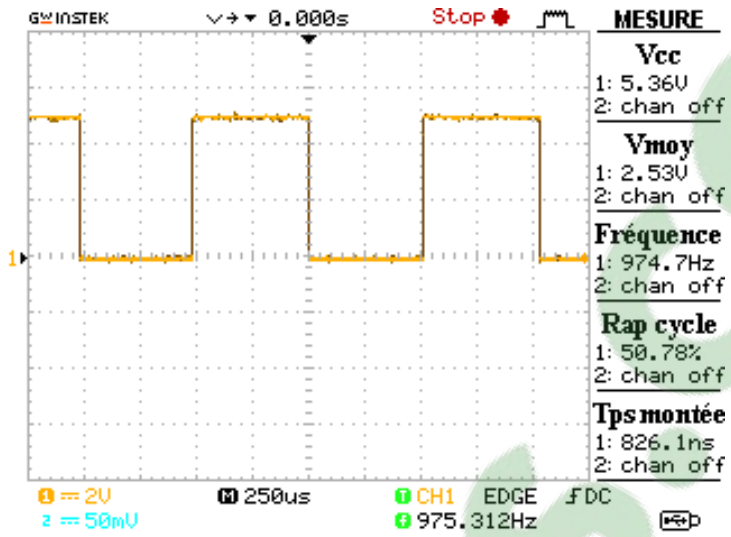


Figure III.13 : un signal PWM pour un rapport cyclique égal à 50%

Et l'exemple d'un signal PWM généré par l'arduino pour un argument de 217 (99%)

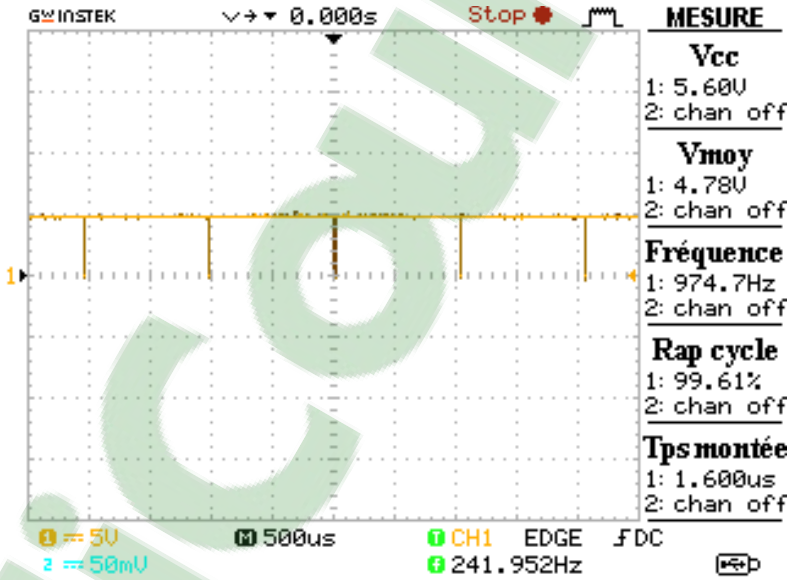


Figure III.14 : un signal PWM pour un rapport cyclique égal à 99%

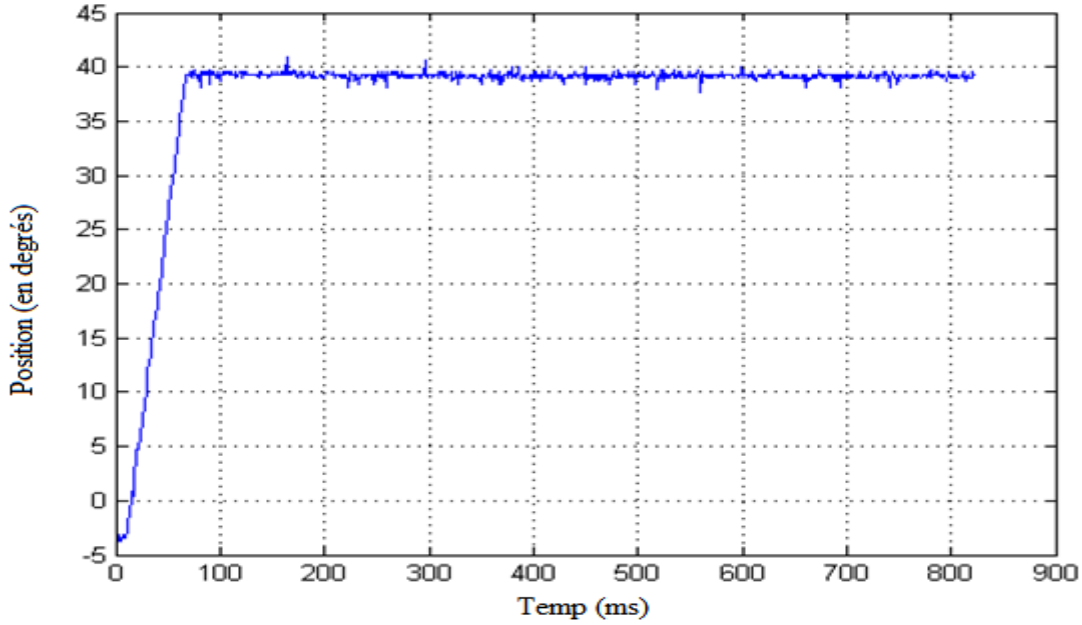


Figure III.15 : résultat obtenue après l'implémentation du correcteur P

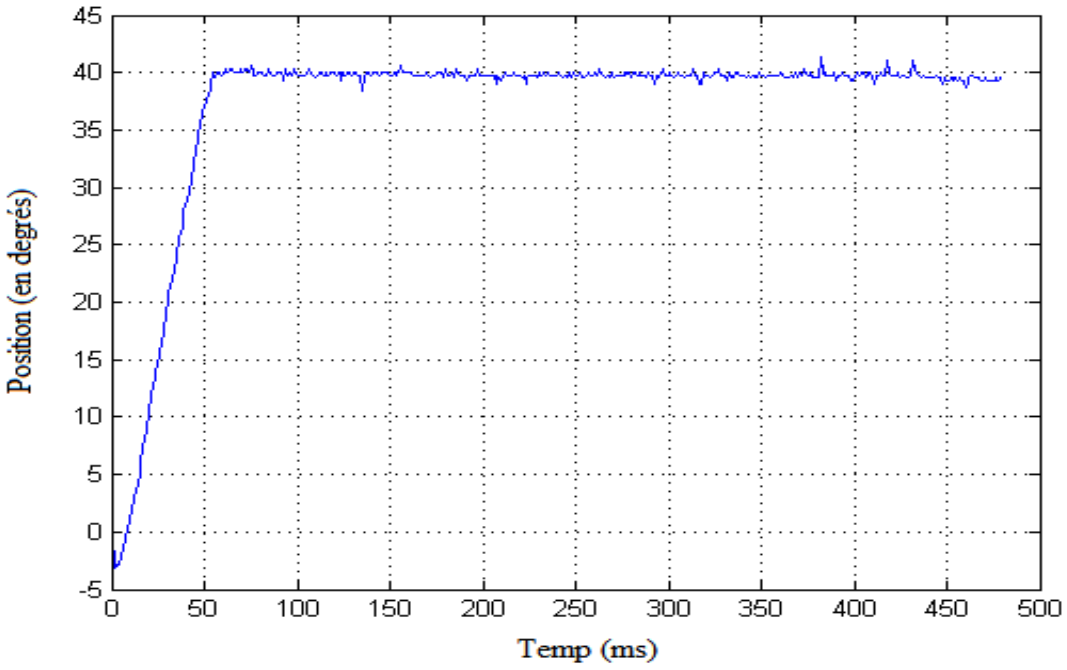


Figure III.16 : résultat obtenue après l'implémentation du correcteur PI

Conclusion :

Dans ce chapitre nous avons décrit en détail toutes les parties importantes de notre réalisation, l'interface de commande (joystick), l'interface graphique, la carte interface, l'amélioration du robot manipulateur KSR10 par l'ajout des capteurs et enfin l'asservissement des positions de notre bras.

Conclusion générale et perspectives

Ce projet nous a permis de nous initier dans le monde pratique de la robotique en nous permettant d'approfondir aussi nos connaissances dans le domaine de la mécanique dédié à la robotique, de l'informatique, de la commande et de l'électronique de puissance.

Dans le premier chapitre, on a donné une vue globale sur les robots, leur historique, leur développement, leur structure et leur constitution.

Une deuxième partie nous a permis la description des différentes parties constituant notre projet, la description du bras KSR10 et ces propriété, ainsi la définition des outils de programmation qui nous a permis de réaliser l'interface graphique et les aspects matériel (carte de commande) qui permet de traduire les différente commande entre le PC et le bras

Le troisième chapitre est dédié principalement à la réalisation de notre projet en expliquant les différentes étapes de conception.

Ce projet nous a permis d'améliorer nos connaissances particulièrement à :

- L'utilisation du logiciel 'Processing' pour le développement de notre interface graphique. Il nous a aussi simplifié la communication entre PC et notre carte d'interfaçage ;
- L'utilisation du protocole de communication de la liaison série (protocole RS232)
- l'utilisation de la carte arduino et sa programmation.

Comme perspectives nous proposons:

- L'utilisation d'une autre carte arduino plus performante que l'Uno pour mieux commander notre bras manipulateur. Cela va permettra d'améliorer les performances de ce dernier,

- De rendre le système plus efficace en rendant le PC travaillant en mode OFF-LINE

- L'amélioration de notre système en le rendant rapide et plus précis.

Bibliographie:

[As12] Astalaseven, Eskimon et olyte " Arduino pour bien commencer en électronique et en programmation " openclassrooms ,2012.
[Vl 02] Vladimir Calderón " Tout simplement JRobot " cybernétique appliquée, 2002.
[Al 02] Ali Kalakech " conception d'un logiciel de programmation graphique des robots d'assemblage " Villeneuve d'Ascq (France), 1991.
[Be 02] Bernard BAYLE robotique : 2ème partie –commande des robots manipulateurs Université de Strasbourg, 2002.
[JU 12] Jules Sery " Bras robotisé "Lycée Jean Perrin, 2012.
[JU 94] Juegouo Josiane nouvelle" méthode de génération des trajectoires pour la commande de bras de robots ", février 1994.
[An 11] Andrew Davison Java Prog. " Techniques for Games ", juillet 2011.
[BO 12] Mme S.Borsali " modélisation des robots " Université Abou Bekr Belkaid de Tlemcen ,2012.
[Di 13] Didier Müller "Robotique" L'informatique au lycée, Avril 2013.
[Lg 07] Ignacio Herrera Aguilar " Commande des bras manipulateurs et retour visuel pour des applications à la robotique de service "Université Toulouse III, 2007.
[Lo 98] Lorenzo FLÜCKIGER "interface pour le pilotage et l'analyse des robots basée sur un générateur de cinématique" école polytechnique fédérale de lausanne, 1998.
[Xa 11] Xavier BROQUERE "Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-robot" Université Toulouse III, 2011.
[Wi 13] Wiley Brand " Arduino Projects For Dummies " John Wiley & Sons, Ltd, Chichester, West Sussex, Angleter, 2013.
[Fi 09] P. Fissette « Introduction à la robotique », France, novembre 2004.

[W] www.fr.flossmanuals.net/processing/
[W1] www.electronicaembajadores.com
[W10]Site officiel Arduino :www.arduino.cc/, consulté en 2015.
[W2] http://www.pobot.org/Asservissement-d-un-moteur-a.html?lang=fr
[W3] http://blog.3sigma.fr/arduino/tutoriel-arduino-asservissement-en-vitesse-dun-moteur-a-courant-continu/
[W4] http://www.zem.fr/arduino-controler-des-moteurs-dc-avec-le-composant-l293d/
[W5] http://www.me.umn.edu/courses/me2011/arduino/technotes/dcmotors/L293/L293.html
[W6] www.conrad.fr
[W7] http://www.pearltrees.com/vitoria/histoire-automates-robots/id6610551
[W8] http://www.galeaindustrie.fr/
[W9] http://www.mon-club-elec.fr/pmwiki_reference_arduino/
[W11] http://www.sbaysite.com/sciences_ingenieur/
[W12] http://www.tomsguide.fr/guide/comparatif-Manette-de-jeu,2-aWRHdWlkZT04NQ==.html

Annexe

L'optocoupleur 4N35 :

Afin de protégé notre carte arduino contre les sur intensité en crée l'isolation électronique entre la carte de commande arduino et la carte de puissance à l'aide des optocoupleurs.

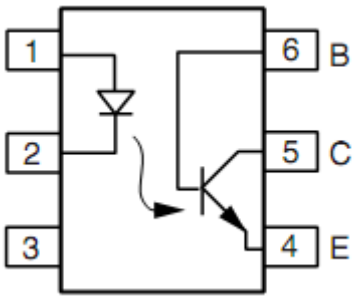
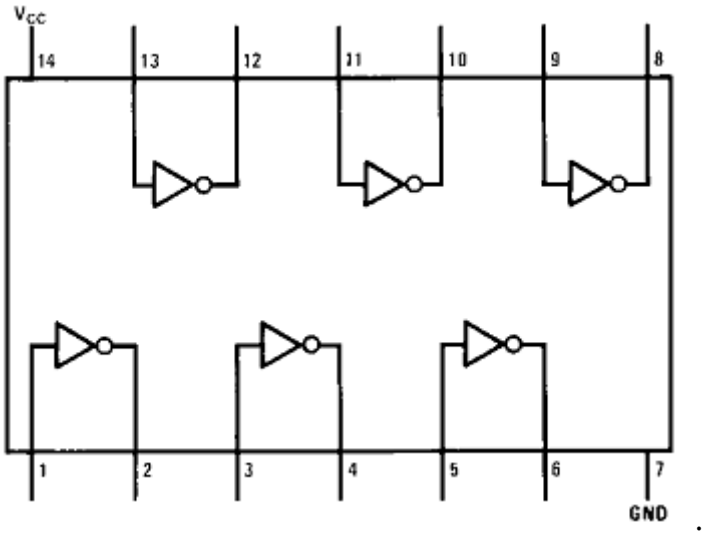


Schéma de l'optocoupleur 4N35

Le hacheur L293

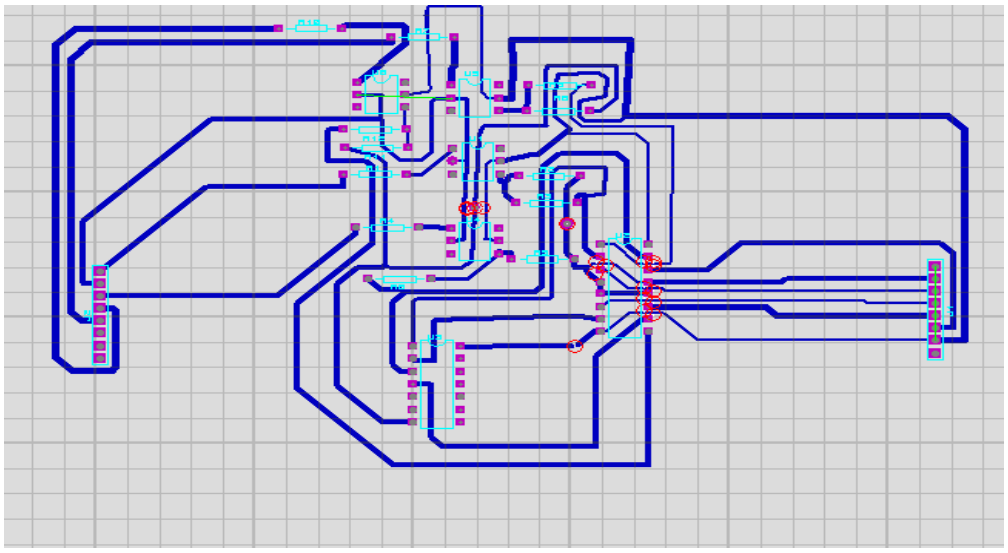
Le L293D est un double pont en H souvent utilisé dans les montages robotiques, il possédant des sorties de courant pour alimenter des moteurs et des entrées de validation. Il ne nécessite donc que très peu de composants externe set sa mise en œuvre est très simple

L'inverseur 74LS04



schémas du hacheur L293

Schéma de la carte dans logiciel ARES :



Interface graphique pour piloter un bras de robot à travers une carte de commande

Résumé

Notre travaille consiste a realiser une interface graphique pour commandé le bras de robot KSR10 atravers une carte ArduinoUNO.Les mouvement du bras seront visuaisé en trois dimetion et sur le PC et enregistrés dans ce dérnier sous format « .txt » .Les valeurs de chaque articulation seras envoyer et recevé via la carte arduino par le port RS232.La carte arduino sért a traduire les commande entre le PC et le Robot KSR10 et faire sont asservissement.

ملخص

عملنا هو تحقيق واجهة رسومية لتحكم في ذراع الروبوت KSR10 من خلال لوحة ARDUINON UNOبحيث تكون حركة الذراع في على جهاز الكمبيوتر ثلاثية الأبعاد وتخزن هذه الحركة في الكمبيوتر في ملف من نوع ". TXT". قيم كل وصلة من وصلات الذراع , ثم يتم إرسال استقبال هذه القيم عبر منفذ الى الأردوينو RS232. يتم استخدام لوحة اردوينو من أجل تحقيق التواصل بين الكمبيوتر والروبوت KSR10 السيطرةعلى الروبوت.

Abstract

Our work is to achieve a graphical interface to the robot arm controlled KSR10 atravers an Arduino UNO board. The arm movement will be viewed in three dimetion and on the PC and stored in the latter in the format ".txt". The values for each link will be send and receve the Arduino board via the RS232 port. The Arduino board is used to translate control between PC and Robot KSR10 and doing are enslavement.

Mots clés

Bras de robot, arduino uno, arduino IDE, proccesing , moteur à courant continue, microcontrôleur, L293