

CHAPITRE IV : CONCEPTION

IV.1 Partie hardware**IV .1.1 Carte de commande (ARDUINO)****IV .1.1.1 Introduction**

Aujourd'hui, l'électronique est de plus en plus remplacée par de l'électronique programmée. On parle aussi de système embarqué ou d'informatique embarquée. Son but est de simplifier les schémas électroniques et par conséquent réduire l'utilisation de composants électroniques, réduisant ainsi le coût de fabrication d'un produit. Il en résulte des systèmes plus complexes et performants pour un espace réduit.

Depuis que l'électronique existe, sa croissance est fulgurante et continue encore aujourd'hui. L'électronique est devenue accessible à toutes personnes en ayant l'envie ce que nous allons apprendre dans ce travail est un mélange d'électronique et de programmation. On va en effet parler d'électronique embarquée qui est un sous-domaine de l'électronique et qui a l'habileté d'unir la puissance de la programmation à la puissance de l'électronique.

IV.1.1.2 Définition du module *Arduino*

Le module *Arduino* est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. *Arduino* est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE *Arduino*. [9]

IV.1.1.3 Les gammes de la carte *Arduino*

Actuellement, il existe plus de 20 versions de module Arduino, nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.

- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPadArduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.
- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'ArduinoDiecimila, avec une interface d'USB et utilise un microcontrôleur

ATmega168.

- L'ArduinoDuemilanove ("2009"), en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- L'ArduinoMega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin deraccordement d'USB et peut être employé comme clavier.
- L'ArduinoEsplora : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Parmi ces types, nous avons choisi une carte Arduino Mega2560.L'intérêt principal de cette carte est de faciliter la mise en oeuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source comme interface de programmation. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur

ATmega2560 et des composants complémentaires. La carte Arduino contient une mémoire morte de 4 kilo. Elle est dotée de 54 entrées/sorties digitales (dont 14 peuvent être utilisées en tant que sortie PWM), 16 entrées analogiques et un cristal a 16 MHz, une connexion USB et Possède un bouton de remise à zéro et une prise jack d'alimentation. La carte est illustrée dans la figure si dessous. [9]

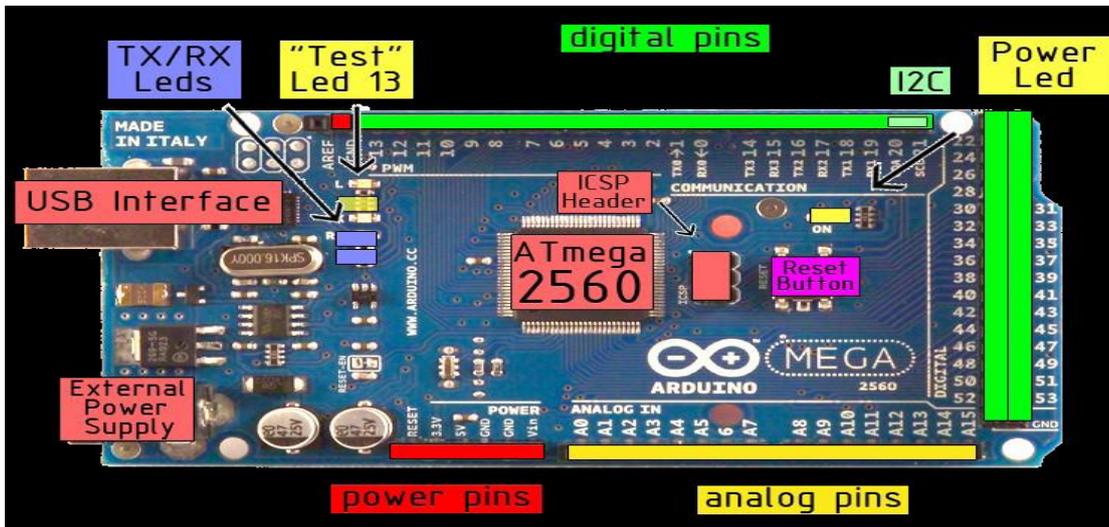


Figure IV.1 La carte Arduino Mega2560

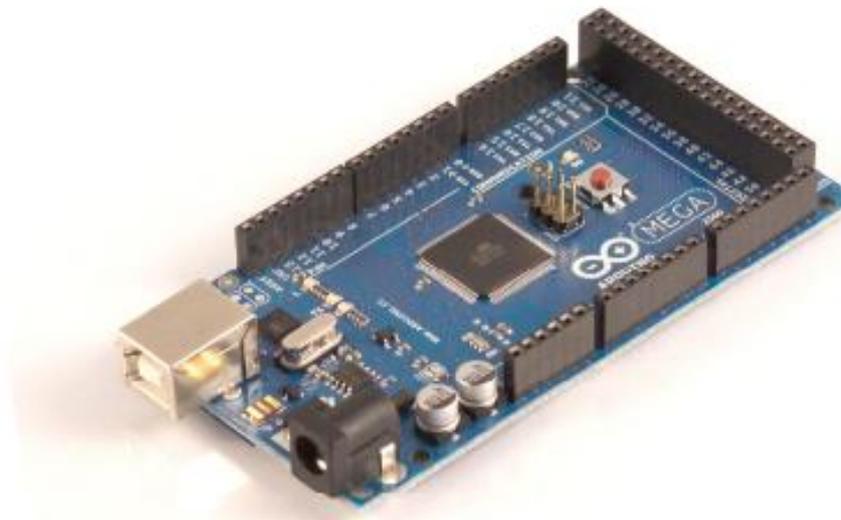


Figure IV.2 La carte Arduino Mega2560

Les Caractéristiques d'ARDUINO Mēga 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Tableau IV.1 Les Caractéristiques d'ARDUINO Mēga 2560

IV.1.1.4 Pourquoi Arduino

Il y a de nombreuses cartes électroniques qui possèdent des plateformes basées sur des microcontrôleurs disponibles pour l'électronique programmée. Tous ces outils prennent en charge les détails compliqués de la programmation et les intègrent dans une présentation facile à utiliser. De la même façon, le système Arduino simplifie la façon de travailler avec les microcontrôleurs tout en offrant à personnes intéressées plusieurs avantages cités comme suit:

- Multi plateforme :

Le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.

- Un environnement de programmation clair et simple :

L'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

- Logiciel Open Source et extensible :

Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

- Matériel Open source et extensible :

Les cartes *Arduino* sont basées sur les Microcontrôleurs *ATMEL ATMEGA8, ATMEGA168, ATMEGA328*, les schémas des modules sont publiés sous une licence créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes *Arduino*, en les complétant et en les améliorant. Même les utilisateurs relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte *Arduino*, dont le but est de comprendre comment elle fonctionne pour économiser le coût.[9]

IV.1.1.5 La constitution de la carte Arduino Mēga 2560

Un module *Arduino* est généralement construit autour d'un microcontrôleur ATMELAVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un bootloader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.[4]

IV.1.1.5.1 Partie matérielle

Généralement tout module électronique qui possède une interface de programmation est basé toujours dans sa construction sur un circuit programmable ou plus.[6]

IV.1.1.5.1.1 Le Microcontrôleur ATmega2560

Un microcontrôleur ATmega2560 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit au temps des pionniers de l'électronique.

Aujourd'hui, en soudant un grand nombre de composants encombrants ; tels que les transistors ; les résistances et les condensateurs tout peut être logé dans un petit boîtier

en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure I.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino.[9]



Figure IV.3 Microcontrôleur ATmega2560

Le microcontrôleur ATmega2560 est constitué par un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de:

- **La mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible mémoire programme de 256Ko (dont **bootloader** de 8 ko).
- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est 8 ko.
- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme. [3]

I.5.1.2 Les sources de l'alimentation de la carte

On peut distinguer deux genres de sources d'alimentation (Entrée Sortie) et cela comme suit :[9]

- **VIN**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche ou, si l'alimentation est fournie par le jack d'alimentation, accéder à la tension d'alimentation sur cette broche.
- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques

nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé) ou de toute autre source d'alimentation régulée.[2]

IV.1.1.5.1.3 Les entrées & sorties

Cette carte possède 54 broches numériques (numérotées de 0 à 53) 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 20-50K Ohms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction **DigitalWrite()** (broche, HIGH). En plus, certaines broches ont des fonctions spécialisées :[7]

- **Interruptions Externes** : 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. - Impulsion PWM (largeur d'impulsion modulée) Ces broches fournissent une impulsion PWM 8-bits à l'aide de l'instruction **analogWrite()**. [9]
- **SPI** (Interface Série Périphérique) : Broches (SS), (MOSI), (MISO), (SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la bibliothèque pour communication SPI. Les broches SPI sont également connectées sur le connecteur ICSP qui est mécaniquement compatible avec les cartes Méga. [9]
- **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. La carte Méga. Dispose 16 entrées analogiques (numérotées de 0 à 15), chacune pouvant fournir une mesure d'une résolution de 10 bits (càd 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 le 0V (valeur 0) et le 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. [9]

La carte Arduino Méga. Intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité). Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche

supplémentaire de protection. Si plus de 500mA sont appliqués au portUSB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuitou la surcharge soit stoppé. [9]

IV.1.1.5.1.4 Les ports de communications

La carte Arduino Méga. à de nombreuses possibilités de communications avec l'extérieur. L'ATmega2560 possède une communication série **UART TTL** (5V), grâce auxbroches numériques **0 (RX)** et **1 (TX)**.On utilise **(RX)** pour recevoir et **(TX)**pourtransmettre (les données séries de niveau **TTL**).

Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega2560 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interfaceentre les niveaux TTL et le port USB de l'ordinateur.Comme un port de communication virtuel pour le logiciel sur l'ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil :[8]



Figure IV.4 les étapes de téléchargement de code

IV .1.2 La carte de puissance

Cette partie du montage est alimentée avec une tension de 9V/3A fournit par l'alimentation du bras de robot **ROB3**. Ce montage se concentre sur l'élément central, le L298N, qui est le cœur du montage. En effet, le circuit amplificateur de puissance L298N de SGS Thomson est une référence en matière d'interface pour les moteurs DC et les moteurs pas à pas. Ces caractéristiques à basses tensions, lui confèrent une place incontestée dans les circuits de puissance intégrée. Il est le plus

utilisés pour les petits robots alimentés par quelques accus. 1. Les signaux de commande entrants du circuit considéré sont :[4]

- Six (06) signaux PWM (Un signal MLI démultiplié sur six voies).
- Six (06) paires de signaux de sens correspondant au sens du moteur sélectionné.
- Une masse commune entre les deux parties puissance et commande.

2. Les signaux de mesures sortant sont :

- Six (06) signaux analogiques fournies par les capteurs à potentiomètres rotatifs situés sur les axes de rotation du robot.[4]

IV.1.2.1 Schéma électronique de la carte de puissance

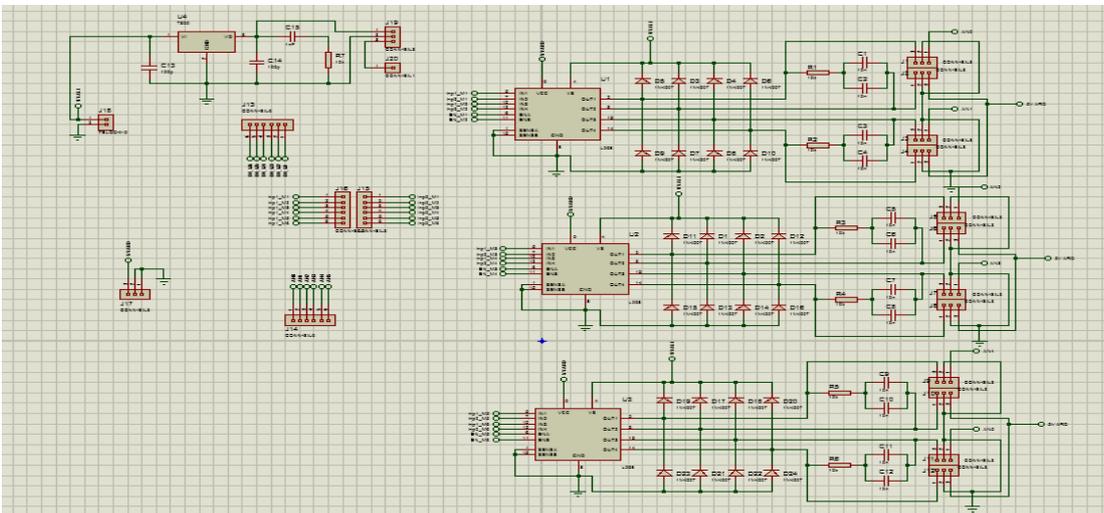


Figure IV.5 : Schéma de la carte de puissance

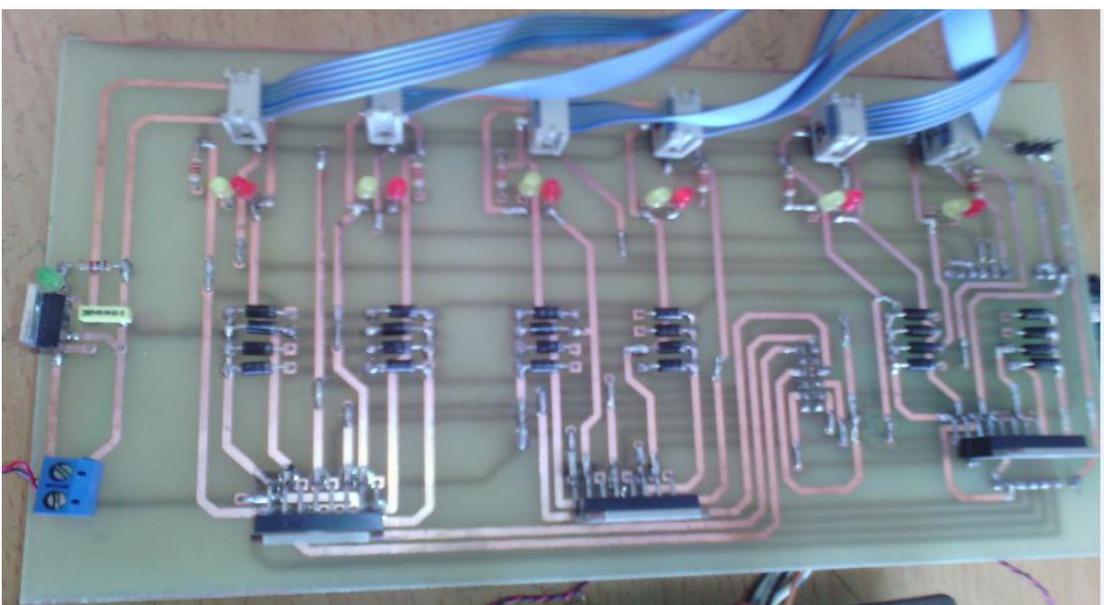


Figure IV.6 : photo de la carte de puissance

IV.1.2.3 Eléments constituant la carte de puissance

IV.1.2.3.1 Le circuit L298N

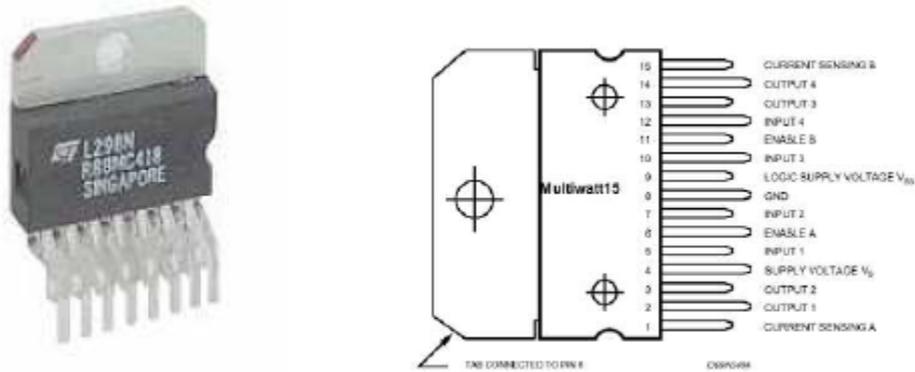


Figure IV.7 Le circuit L298N.

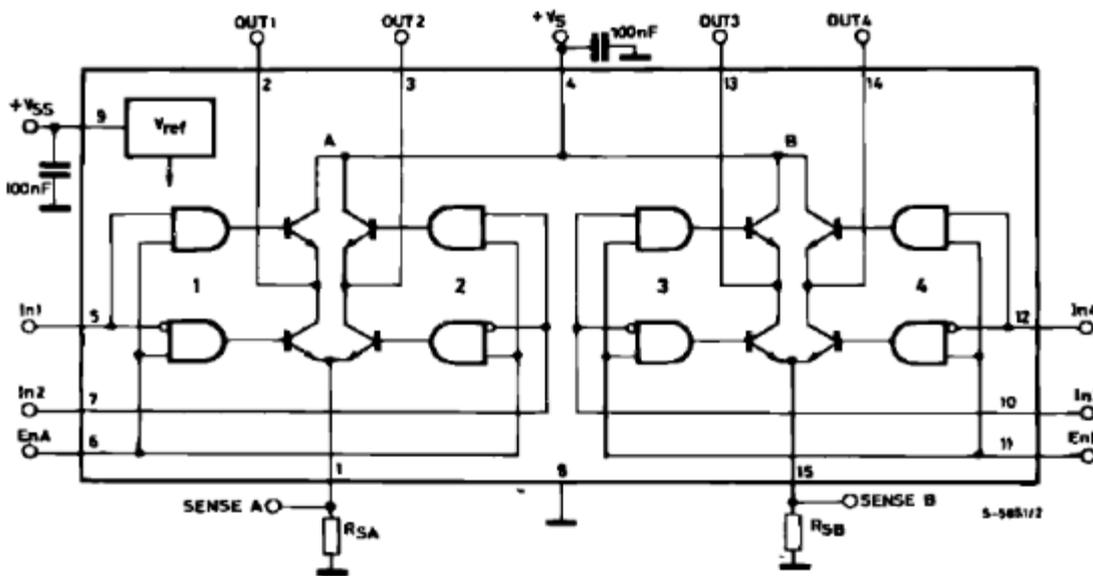


Figure IV.8: Diagramme bloqué du L298.

Les caractéristiques du L298 sont les suivantes :

- Intensité maximale : 2A par pont.
- Alimentation de puissance de 5.5V à 50V.
- Type de boîtier : Multiwatt15.
- Dissipation puissance total : 25w.
- Trois entrées par pont : In1, In2 et ENABLE.

Entrées		Sorties	Fonction
ENA = 1	In1=1 ; In2=0	Out1=1 ; Out2=0	Sens 1
	In1=0 ; In2=1	Out1=0 ; Out2=1	Sens 2
	In1=In2	Out1=1 ; Out2=1 Ou Out1=0 ; Out2=0	Le frein
ENA = 0	In1=X ; In2=X	Rien	roue libre

Tableau IV.2: Logique de commande du LN298 (un pont).

Donc, le pilotage des moteurs se fait par génération d'une MLI en s'appliquant sur les entrées «Enable A» et «Enable B» et le L298N se charge de contrôler la vitesse et le sens de rotation des moteurs indépendamment.

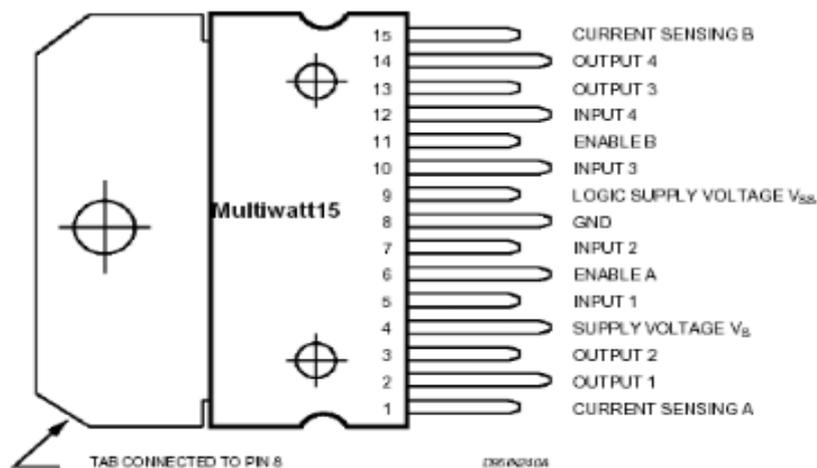


Figure IV.9: Brochage du circuit intégré L298.

IV.1.2.3.2 Les diodes 1N4007 :

Les diodes (1N4007) sont des diodes de puissance présentent un temps de commutation faibles et capables de laisser passer un courant important. Elles servent de protection « roue libre », afin de compenser le courant inverse qui se fera pour chaque déclenchement ON/OFF du moteur. Donc nous avons câblé huit (08) diodes en antiparallèle sur chaque L298N pour la protection de ce circuit parce que contrairement à la majorité des circuits, le L298N n'intègre pas les diodes de protections des transistors internes.[3]



Figure IV.10 : Diode 1N4007

IV.1.2.3.3 Le circuit d'alimentation

Cette partie est très simple. Nous avons besoin de produire une tension stabilisée. Il existe pour cela des composants dédiés, les 78xx. Ces composants sont des composants à trois pattes. Leur fonctionnement est très simple. Ils ont entre les deux premières pattes une tension d'alimentation quelconque (supérieure à la tension de sortie désirée). Ils fournissent alors en sortie une tension stabilisée égale à xx (05, 09, 12,...). On utilise donc ces composants en parallèle avec des condensateurs de découplage comme conseillé dans le Datasheet. Pour notre montage on a besoin de tension stabilisée de 5v donc on utilise le régulateur **LM7805** et pour la protection contre les court-circuit ou les surintensités de courant on ajoute un fusible est disposé directement après l'alimentation. Suivie d'un groupe RESISTANCE-LED de contrôle qui confirme la mise sous tension du circuit et le sens de rotation du chaque moteur.[7]



Figure IV.11.Fusible et son support.

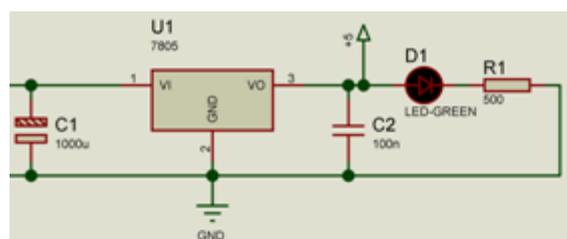


Figure IV.12. Circuit d'alimentation.

IV.1.2.4 Type de commande

Pour commander les six (06) moteurs du bras manipulateur ROB03 on a réalisé deux types de commande, commande en boucle ouverte (commande manuelle) et en boucle fermée (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 ou à partir d'un mouvement relatif, il renseigne sur la position de l'arbre. L'information retournée pourra être traitée dans le module « convertisseur Analogique / Numérique » du ARDUINO.[6]

IV.1.2.4.1 Le signal de commande PWM(MLI)

PWM veut dire Pulse Width Modulation ou en français **MLI** Modulation à Largeur d'Impulsion est un signal auquel on fait varier la tension sans modifier ni l'amplitude ni la fréquence mais la largeur de l'impulsion. En fait le signal PWM n'est pas continu, c'est un signal à impulsions (ressemble au signal carré). Il possède un état haut (un 1 logique) et un état bas (un 0 logique). Voici un schéma représentant un signal PWM :[9]

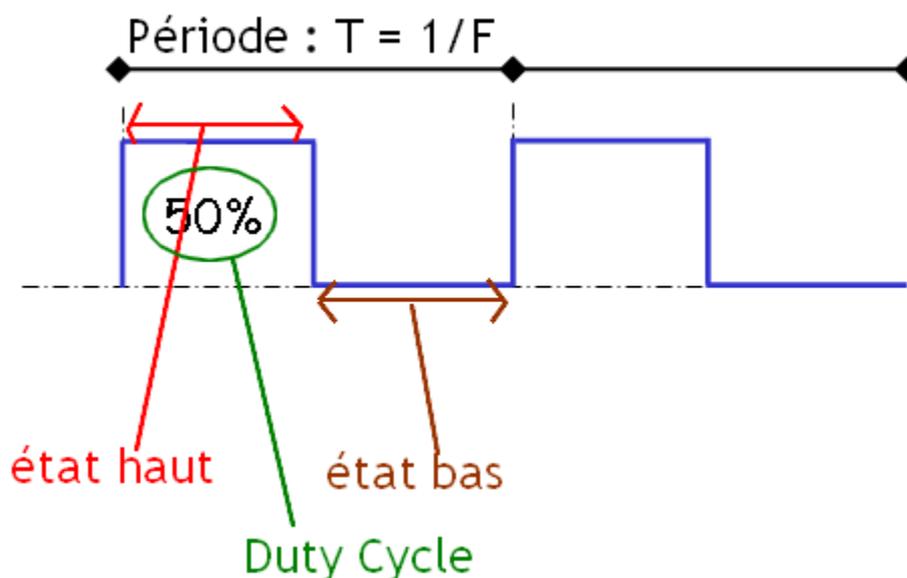


Figure IV.13 : schéma représentant un signal PWM

Dans notre montage Le module Capture/Compare/PWM (CCP) en mode PWM de ARDUINO permet de générer un signal numérique pouvant être utilisé afin de reproduire un signal analogique situé entre 0 et 5V en faisant varier la durée du "Duty Cycle" et en se basant sur le Timer2. Cette tension active et désactive à son tour le pont H faisant varier analogiquement la tension d'alimentation des moteurs de 0 à 9V [9]

IV.2. Partie software

IV.2-1 programmations :

IV.2.1.1 Introduction

La commande en mouvement de notre bras manipulateur ROB3 est essentiellement réalisée par la carte ARDUINO Méga 2560 programmé, les lois de commande sont implémentées sous forme d'algorithmes à partir de langage arduino.[9]

IV.2.1.2. Schéma simplifié du système de commande

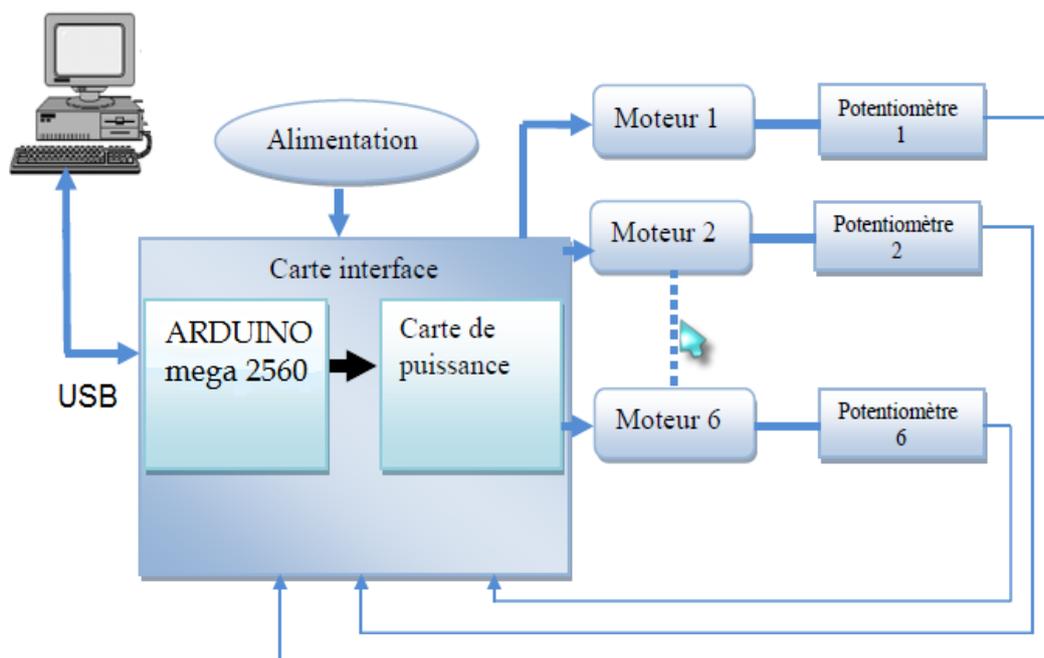


Figure IV.14 : Schéma simplifié du système de commande

IV.2.1.3 La commande :

Pour commander les six (06) moteurs du bras manipulateur ROB03 on a réalisé une 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 ou à partir d'un mouvement relatif, il renseigne sur la position de l'arbre. [7]

Matériel nécessaire :

- PC
- Bras de *ROBOT ROB3*
- Une carte Arduino Méga 2560 :
- Six (06) signaux *PWM* pour la commande des moteurs.
- Deuze (12) signaux pour la direction de la rotation des moteurs.
- Une masse commune entre les deux parties puissance et commande.
- Six (06) signaux analogiques fournies par les capteurs à potentiomètres rotatifs situés sur les axes de rotation du robot.
- Carte de puissance
- Alimentation 9V
- 1 câble USB A-B pour la programmation de la carte ARDUINO

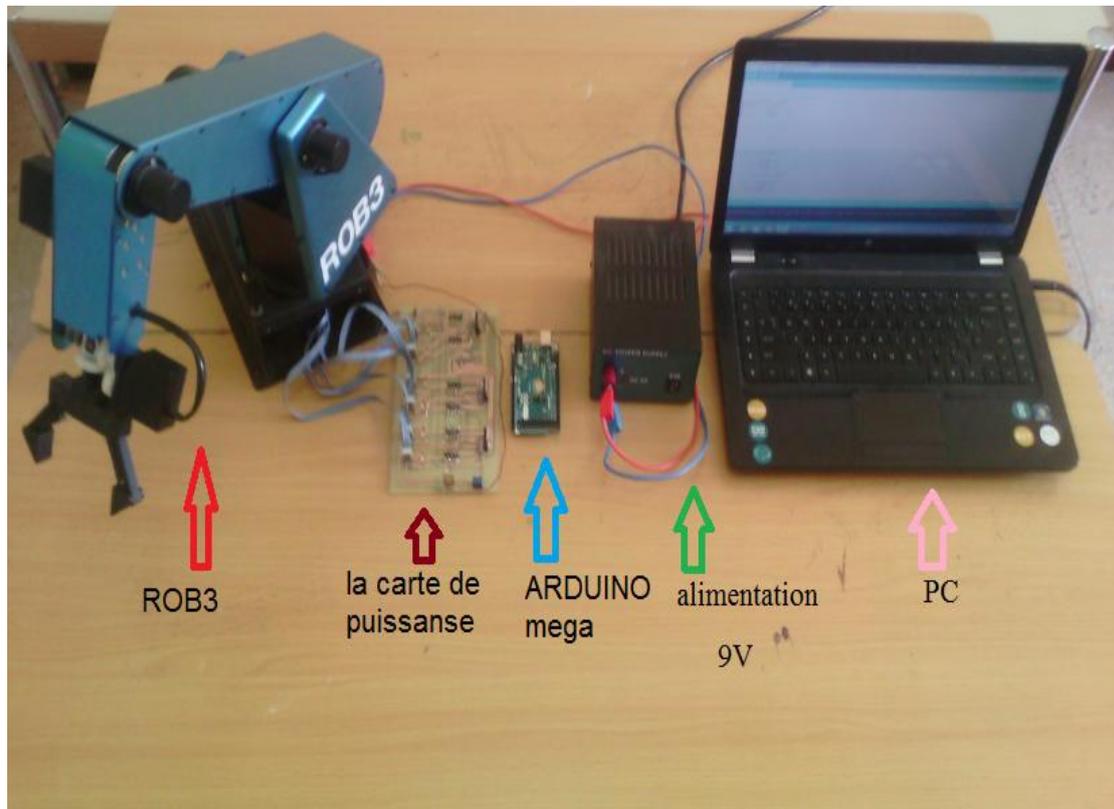


Figure IV.15 les différents moyens utilisés dans la réalisation

IV.2.1.4 Résultats de travail

Pour la commande de notre bras de robot **ROB3** on a appliqué les résultats de la modélisation géométrique qui on a fait dans le chapitre 3, après nous avons implanté un programme dans ARDUINO

IV.2.1.4.2le travail

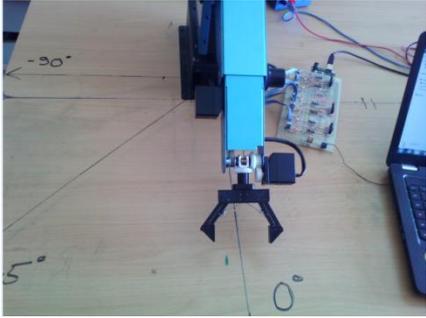
Essais	Position	L'angle	Images
Essai 1 MGD Moteur N°01 $\theta 1$	initiale	$\theta 1 = 0$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	
Essai 2 MGD Moteur N°01 $\theta 1$	Désirée	$\theta 1 = -90$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	

Tableau IV.3 le travail

IV.2.1.4.3 Résultats

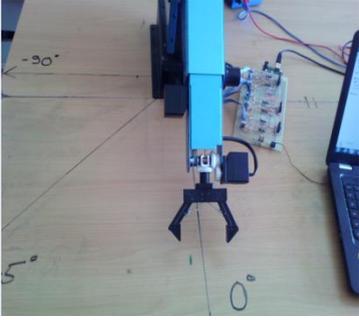
Essais	Position	L'angle désirée	L'angle mesuré	L'erreur	Images
Essai 1 MGD Moteur N°01 $\theta 1$	initiale	$\theta 1 = 0$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	$\theta 1 = 0$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	Erreur=0	
Essai 2 MGD Moteur N°01 $\theta 1$	Désirée	$\theta 1 = -90$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	$\theta 1 = -100$ $\theta 2 = 0$ $\theta 3 = 0$ $\theta 4 = 0$ $\theta 5 = 0$	Erreur=10 Degrée	

Tableau IV.4 les résultats de travail

IV.2.1.5 Conclusion

Dans ce chapitre nous avons présenté la carte de commande ARDUINO avec la carte de puissance de façon détaillée, et on a implantée un programme dans l'Arduino pour commander notre bras de robot **ROB3**, mais nous avons obtenu des résultats négatifs par rapport à ce que nous espérons, à cause de l'erreur entre la mesure et la consigne.

Et pour cette raison nous avons été obligé de passer à l'asservissement de position pour éliminer cette erreur.