

Sommaire:

Introduction Générale.....	5
Chapitre 1 : Présentation du cadre du projet.....	6
Introduction.....	6
I. Robotique	6
I.1 Définition.....	6
I.2 Historique	6
I.3. Types de robots.....	7
I.3.1 Robots Manipulateurs	7
I.3.2 Robots mobiles.....	9
II. Présentation du robot lynx 6.0.....	11
II.1 Définition.....	11
II.2 Composants matériels.....	12
II.2.1. Les Servomoteurs	12
II.2.2 Le Capteur GP2D12	14
II.2.3 La carte SSC 32	15
III. Objectifs et Problématiques.....	16
IV Conclusion	18
Chapitre 2 : Conception.....	19
Introduction.....	19
I. Cahier de charge.....	19
II. Schéma Synoptique	20
III. Organigramme.....	22
IV. La Carte SSC-32	23
IV.1 Détails Techniques	23
IV.2 Le Microcontrôleur ATMEGA168PU20	25
V. La carte Arduino Uno	27
V.1. Caractéristiques	28
V.2 Entrées et Sorties Numériques	29

V.3 Broches Analogiques	30
V.4 Le microcontrôleur ATMEGA328	30
VI. Conclusion	31
Chapitre 3 : Réalisation du projet.....	32
Introduction.....	32
I.1 Hardware :	32
I.1. Présentation :	32
I.2. Fonctionnement du servomoteur :	32
I.3 Fonctionnementdu Capteur GP2D12 :	33
I.4 La connexion entre les cartes ArduinoUno et SSC-32 :	37
II. Software:	39
II.1 Arduino Integrated Development Environment (IDE):	39
II.2 LynxTerm :	40
III. Application :	40
III.1 Montage du système :	41
III.2 Programmation :	41
III.2.1 Commande des servomoteurs :	41
III.2.2 Programmation du capteur GP2D12 :	43
IV. Conclusion :	45
Conclusion Générale	46
Bibliographie	47
Nétographie.....	48

Liste des Figures

Chapitre 1 :

Figure 1 : Le robot Unimate	8
Figure 2 : Le robot médical, DaVinci	8
Figure 3 : Le robot explorateur Spirit	9
Figure 4 : Drone Predator	10
Figure 5 : Le robot Guide Musée	10
Figure 6 : Robot Tondeuse	11
Figure 7 : Voiture Autonome	11
Figure 8 : Le robot Lynx 6	12
Figure 9 : Les composants du servomoteur	13
Figure 10 : Le servomoteur	13
Figure 11 : Le capteur GP2D12	14
Figure 12 : La carte SSC-32	16
Figure 13 : Le logiciel LynxTerm	16
Figure 14 : La carte Arduino Uno.....	17

Chapitre 2 :

Figure 1 : La première source d'alimentation	19
Figure 2 : Transformateur à 9V	20
Figure 3 : Schéma Synoptique	21
Figure 4 : Organigramme.....	22
Figure 5 : La carte SSC-32	23
Figure 6 : L'ATMEGA168PU20	25
Figure 7 : Le bloc diagramme de l'ATMEGA168Pu20.....	27
Figure 8 : Brochage de la carte Arduino Uno	28

Chapitre 3 :

Figure 1 : Types du servomoteur.....	33
Figure 2 : Vue de face du GP2D12	34
Figure 3 : Bloc diagramme du GP2D12.....	34

Figure 4 : Détection de l'objet.....	35
Figure 5 : La courbe distance en fonction de Vout.....	35
Figure 6 : La courbe $[1/(Distance+k)]$ en fonction de Vout.....	36
Figure 7 : Montage de test du capteur GP2D12 avec la carte Arduino.....	37
Figure 8 : Communication série via le port RS232	37
Figure 9 : Mode communication série via les pins RX et TX.....	38
Figure 10 : Communication série via les deux cartes	38
Figure 11 : Arduino IDE.....	39
Figure 12 : LynxTerm	40
Figure 13 : Montage du robot	41
Figure 14 : Robot en position initiale	42
Figure 15 : Serial Monitor.....	45

Liste des tableaux

Tableau 1 : modification des cavaliers (jumpers)	24
Tableau 2 : Caractéristiques de l'Arduino Uno	29

Introduction Générale

La Robotique était depuis des décennies, un domaine qui inspire beaucoup les savants et les chercheurs. Ces dernières années, les robots sont utilisés dans plusieurs domaines surtout dans l'industrie pour assurer des tâches répétitives. Suite à plusieurs évolutions techniques, ils sont employés dans des secteurs de très haut niveau comme le militaire, l'exploration spatial et la médecine.

La grande majorité des robots industriels exécute en mode de répétition une séquence de mouvements bien définie. La réactivité par rapport à un événement externe est très limitée qui se borne souvent à des réactions basiques « tout ou rien ». L'environnement doit donc être constant et connu avec une grande précision ce qui n'est pas toujours le cas.

Afin de donner au robot plus de flexibilité et d'adaptabilité, il est nécessaire d'inclure dans sa commande des retours sensoriels externes. Ainsi, nous avons des véhicules intelligents qui prennent en compte des variations dans l'environnement de travail.

C'est dans ce cadre que s'inscrit notre projet de fin d'études réalisé au sein de l'Institut Supérieur de l'Informatique (ISI). Notre travail consiste à commander le robot Lynx 6 pour récupérer des objets et les déplacer dans un emplacement bien défini.

Le présent rapport présente trois chapitres. Il est articulé comme suit. Le premier chapitre présentera la robotique et les différents types de robots ainsi qu'une présentation générale à notre robot. Le deuxième chapitre nous donnera une idée sur le fonctionnement de notre système sous forme d'un organigramme et présentera les cartes utilisées pour commander notre robot. Le dernier chapitre sera réservé pour la partie réalisation. Nous clôturons notre rapport par une conclusion générale et les perspectives envisageables pour notre projet.

Chapitre 1 : Présentation du cadre du projet

Introduction

Ce chapitre est considéré comme un chapitre introductif. Nous allons mettre l'accent sur la robotique, les robots ainsi que leurs types. Puis, nous passerons à la présentation du robot que nous allons commander « lynx robot 6.0 ». Enfin, nous citons les problématiques vécues et les objectifs de notre travail.

I. Robotique

La robotique peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles. [B1]

I.1 Définition

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à 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

I.2 Historique

La robotique comme ensemble des études, des techniques de conception, de création et de mise en œuvre des robots nous oblige à nous poser la question d'un point de vue historique, question liée à la culture (littérature, cinéma), puis technique de ce que nous entendons par « ROBOT ».

Ce terme que les petits enfants même utilisent aujourd'hui n'a pas existé dans nos langages, il a été inventé par l'écrivain Karel Capek dans la pièce de théâtre Rossum's Universal Robots

connue sous l'abréviation R.U. R en 1920 pour désigner un dispositif mécatronique accomplissant des tâches pour le compte de son créateur, Karel Capek désignait son frère, Joseph, peintre et écrivain comme étant l'inventeur réel de ce mot, en tchèque signifie «esclave » ou « travailleur dévoué ».

Le mot « ROBOTIQUE » a été introduit dans la littérature par Isaac Asimov en 1942 dans son livre « Runaround » dans lequel il évoque « les trois lois de la robotique », les voici :

***Première Loi :**

Un robot ne peut blesser un être humain ni, par son inaction, permettre qu'un humain soit blessé.

***Deuxième Loi :**

Un robot doit obéir aux ordres donnés par les êtres humains, sauf si de tels ordres sont en contradiction avec la Première Loi.

***Troisième Loi :**

Un robot doit protéger sa propre existence aussi longtemps qu'une telle protection n'est la Deuxième Loi.[N1]

I.3. Types de robots

On peut classer les robots en deux catégories. Les robots manipulateurs qui sont des robots à base fixe permettant de manipuler des objets et les robots mobiles qui sont des robots capables de se déplacer pour accomplir une tâche et aussi capable de transporter sur leur plateforme un robot manipulateur.

I.3.1 Robots Manipulateurs

Un robot manipulateur est un robot qui possède une base fixe et qui sert à manipuler des objets et généralement il est dédié pour réaliser une tâche précise ou répétitive. Ce type de robots est généralement utilisé dans l'industrie et les différents types d'usine et aussi dans la médecine.

➤ Les Robots Industriels

Ce sont des robots dédiés aux tâches répétitives et avec rigueur. Un de leurs avantages est de remplacer les ouvriers dans les tâches pénibles et dangereuses. Les robots de l'industrie sont

souvent munis de systèmes de visions qui leur procurent une souplesse d'exécution et des moyens de vérifier la qualité des produits fabriqués.

En 1954, le premier robot industriel nommé « UNIMATE » a été créé par l'inventeur américain « George Charles Devol ». Durant l'année 1961, Unimate a été mis en place dans une usine de General Motors afin de déplacer des objets bien déterminés

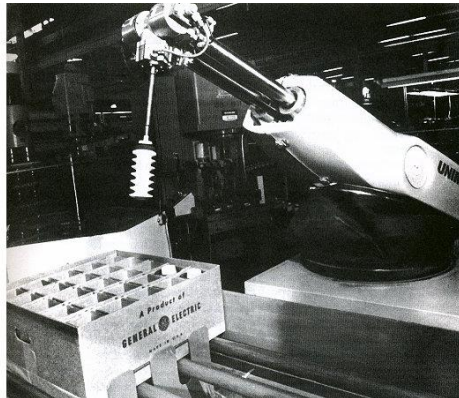


Figure 1 : Le robot Unimate

➤ Robots pour l'assistance médicale

Un robot médical est un robot utilisé dans le cadre d'une application thérapeutique, par exemple lors d'une chirurgie ou au cours d'un programme de réhabilitation neuro-motrice. Citant l'exemple du robot « Da Vinci » qui est utilisé pour les chirurgies. Il s'agit d'une machine dirigée par un chirurgien pour réaliser des opérations principalement au niveau de l'abdomen.

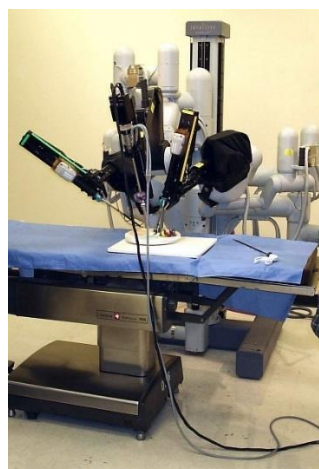


Figure2 :Un robot médical, Da Vinci

Le premier robot chirurgical au monde est le Arthrobot, développé et utilisé pour la première fois à Vancouver en 1983. Le tout premier robot a été utilisé le 12 mars 1984 lors d'une opération de chirurgie orthopédique au UBC Hospital de Vancouver.

I.3.2 Robots mobiles

Nous pouvons définir un robot mobile comme un robot susceptible de se déplacer dans un environnement pour effectuer une tâche précise. Ce type de robot peut avoir un robot manipulateur sur sa plateforme.

Les robots mobiles sont généralement des machines polyvalentes et capables de s'adapter à certaines variations de ses conditions de fonctionnement, c'est pour ça ils sont utilisés pour l'exploration spatiale, les systèmes militaires et les différents types de services (transport de marchandises, ménagers...)

➤ Robots Explorateurs

Ils sont destinés à explorer des environnements où l'homme ne peut pas se rendre, c'est pour cela les scientifiques ont pris le parti d'envoyer des robots pour explorer le système solaire qui est considéré un environnement mortel pour l'être humain. Citons l'exemple de l'astro-mobile « Spirit » envoyé sur « Mars » par la « NASA » en 2003.

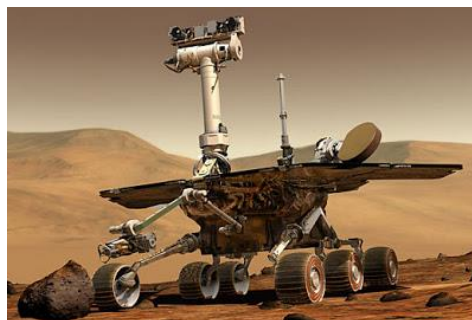


Figure3 :Le robot explorateur« SPIRIT »

Nous pouvons trouver aussi l'emploi des robots explorateurs dans l'exploration d'épaves ou de décombres ainsi que la recherche des victimes par exemple lors de tremblements de terre...

➤ **Robots Militaires**

Les robots militaires sont des robots autonomes ou contrôlés à distance, conçu pour des applications militaires. Des systèmes sont déjà actuellement en service dans un certain nombre de forces armées, avec des succès remarquables, tel que le « drone Predator », qui est capable de prendre des photographies de surveillance, et même à lancer des missiles au sol, pour les drones de combat. Les études se poursuivent car ce type d'engin offre des possibilités prometteuses.

Les drones sont une sous-classe des robots militaires.



Figure 4 : drone Predator

➤ **Robots de Service**

Nous pouvons affirmer que ces robots ont été inventés pour aider l'homme à faire des certaines tâches dans beaucoup de domaines. Par exemple le robot ci-dessous joue le rôle d'un guide de musée à la cité de l'espace à Toulouse.



Figure 5 : Robot guide musée

Nous pouvons trouver aussi des autres robots de service comme les robots ménagers (aspirateur, tondeuse ...) et les robots qui fournissent l'aide aux personnes âgées ou ayant un handicap.



Figure6 :Robot tondeuse



Figure7 :voiture autonome

La robotique semble avoir un avenir très prometteur dans le secteur industriel et la recherche. C'est dans ce cadre que s'inscrit notre projet de fin d'études qui consiste à modéliser et commander le robot « lynx 6.0 ». Commençons tout d'abord par introduire notre robot.

II. Présentation du robot lynx 6.0

II.1 Définition

Le lynx robot 6.0 ou encore « The L6AC-KT Lynx 6 Robotic Arm » est un produit de la fameuse société, Lynxmotion, fondée en 1995 par Jim Frye. Il s'agit d'un bras robotique qui sert à déplacer des objets à un emplacement bien déterminé.

Dans notre cas, ce robot a été amélioré par un capteur de mesure de distance afin de détecter la présence d'un objet puis le déplacer.



Figure8 : Le Robot Lynx 6

II.2 Composants matériels

Pour bien commander notre robot, nous devons d'abord savoir les composants matériels qui le constituent. Le bras robotique se compose d'une base, trois articulations et une pince. A chaque partie, un servomoteur est attaché. Nous distinguons alors que ce robot contient cinq servomoteurs. Pour détecter la présence de la pièce à déplacer, un capteur de mesure de distance a été attaché à la pince. Enfin, nous trouvons la carte « SSC-32 » qui a pour rôle de tester les servomoteurs du robot lynx 6 à travers un logiciel nommé « LynxTerm».

II.2.1. Les Servomoteurs

Un servomoteur (souvent abrégé en « servo », provenant du latin « servus » qui signifie « esclave ») est un moteur capable de maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure [N2]. Il est composé :

- ✓ Un moteur à courant continu
- ✓ Un axe de rotation
- ✓ Un capteur de position de l'angle d'orientation de l'axe (très souvent un Potentiomètre).
- ✓ Une carte électronique pour le contrôle de la position de l'axe et le Pilotage du moteur à courant continu

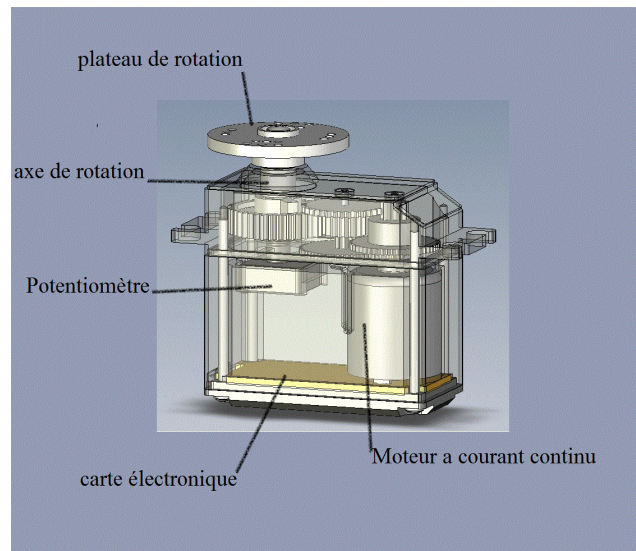


Figure9 : Les composants du servomoteur

Un servomoteur est capable d'attendre des positions prédéterminées dans les instructions qui lui on était donné, puis de les maintenir. Il a l'avantage d'être asservi en position angulaire, cela signifie que l'axe de sortie du servomoteur respectera la consigne d'instruction que vous lui avez envoyée en son entrée.

Même si un obstacle tient sur la route, qui viendrait à lui faire changer l'orientation de sa trajectoire, il essaiera de conserver la position.



Figure 10 : Servomoteur

II.2.2 Le Capteur GP2D12

Le capteur utilisé dans notre robot est le capteur Sharp GP2D12. Commençons d'abord par l'introduire.

➤ Définition

Le GP2D12 est un capteur de distance à infrarouge. Il utilise une triangulation du rayon lumineux afin de calculer la distance à l'objet. C'est un capteur largement répandu dans le milieu de la robotique mobile. Il s'adapte facilement sur tous les robots.

➤ Principe de fonctionnement

Le GP2D12 est composé d'une combinaison intégrée de PSD (Position Sensitive Detector), IRED (InfraRed Emitting Diode) et un circuit de traitement de signal.

Le IRED émet le signal lumineux et le PSD le reçoit après réflexion sur l'obstacle. Puis le rôle du circuit de traitement de signal intervient en traitant l'information en provenance du capteur pour envoyer un compte-rendu au microcontrôleur.

➤ Caractéristiques

Les caractéristiques techniques du capteur GP2D12 sont les suivantes :

- ✓ Intervalle de mesure de distance entre 10 et 80 cm
- ✓ La sortie analogique du voltage correspond à la distance
- ✓ Fonctionne grâce à une alimentation 5V
- ✓ Interface pratique 3-pin
- ✓ 2 trous de fixation espacés de 37 mm



Figure11 : Le Capteur GP2D12

II.2.3 La carte SSC 32

Afin de contrôler notre robot ainsi que tester ses servomoteurs, nous devons utiliser la carte qui lui est attaché. Présentons alors cette carte.

La SSC-32 est une carte électronique de Lynxmotion. Il est possible de commander plusieurs servomoteurs RC. Un connecteur DB9 permet de la relié à un contrôleur par liaison RS232 ou par liaison UART. Grâce à un adaptateur USB2RS232, cette carte peut être commandée à partir d'un PC.

La vitesse de communication peut être changé manuellement de 2400 à 115200bps. Sur cette carte, il y a 32 canaux générateurs de pulsions pour les servomoteurs. Chacun d'eux peut être contrôlé individuellement et configurable sur plusieurs critères afin de faire varier la position et la vitesse angulaire de l'axe du servomoteur.

L'alimentation de la carte se fait par une source de tension externe de 6 à 9V. Pour les servomoteurs deux alimentations secondaires différentes peuvent être nécessaire en fonction de la valeur préconisée par ceux-ci. Sinon, un cavalier permet de lier l'alimentation générale de la carte avec celle des servomoteurs.

La carte est contrôlée grâce à un microcontrôleur Atmel ATMEGA168_20PU

Le logiciel LynxTerm de Lynxmotion est compatible avec la carte SSC-32. Il permet de contrôler manuellement la position et la vitesse des servomoteurs des 32 canaux. Plusieurs configurations sont accessibles pour chaque canal. Il y a aussi la possibilité d'écrire un script afin de tester quelques phases de mouvement pour les servomoteurs. [N3]



Figure12 :La carte SSC-32

III. Objectifs et Problématiques

Notre objectif principal est de commander le robot lynx 6.0 en le rendant capable de détecter la présence d'un objet, l'agripper par la pince puis le déplacer. Mais le problème qui nous a rencontrés est que la carte SSC-32 a été déjà programmée et pour contrôler le robot nous devons utiliser le logiciel lié à cette carte, « LynxTerm ».

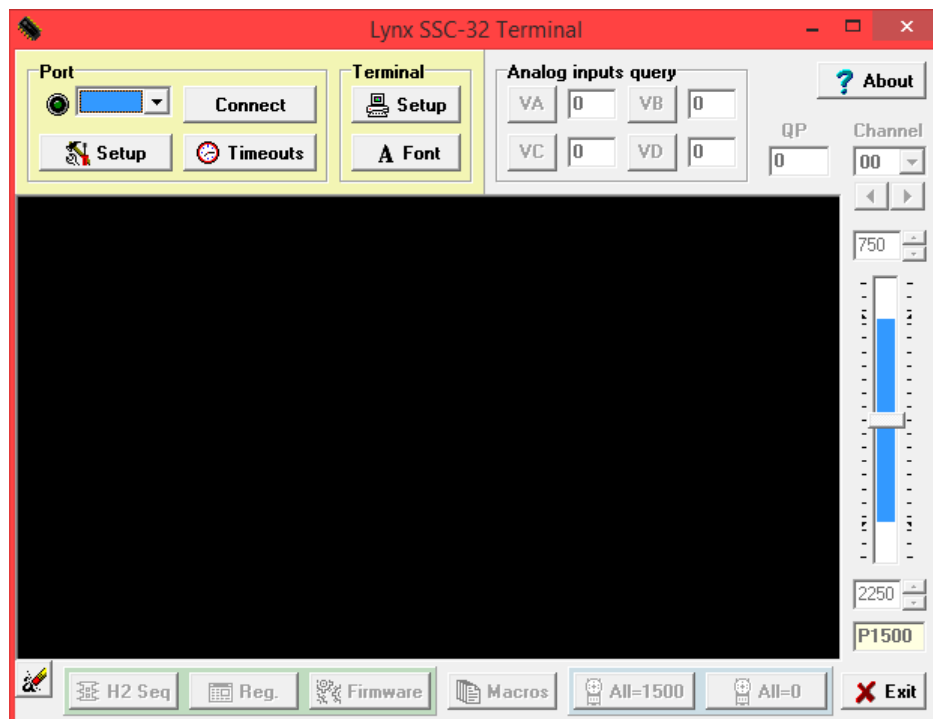


Figure 13 : LynxTerm

La figure ci-dessus nous montre l'interface du logiciel LynxTerm. Nous pouvons dire que son rôle principal est de tester les servomoteurs manuellement ou bien en utilisant des commandes qu'on fait entrer au terminal.

Cette méthode ne nous permet pas de créer des scénarios précis et réels. Ainsi, le cycle de travail de notre robot commence toujours lors de la détection de la présence d'une pièce est cette information ne peut pas introduite à cette interface. Alors nous avons décidé d'utiliser la carte « Arduino Uno » pour manipuler le robot lynx 6.0 mais à travers la carte SSC-32, c'est-à-dire, nous allons utiliser la carte du robot comme un circuit intermédiaire entre la carte Arduino et notre robot.

La carte Arduino Uno, encore appelée Genuino, est un circuit imprimé sur lequel nous pouvons brancher toutes sortes d'appareils. Cette carte se programme sur l'ordinateur via un câble USB (ou autre) et permet ensuite de diriger n'importe quel appareil. Grâce au microcontrôleur « Atmega328p » installé dans la carte, nous pouvons la programmer pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme le contrôle des appareils domestiques, éclairage, chauffage, le pilotage d'un robot...



Figure 14: La Carte Arduino Uno

Alors pour réussir notre projet, nous devons savoir comment faire la communication entre la carte SSC-32 et la carte Arduino afin d'avoir un fonctionnement correct du robot et aussi nous devons trouver la bonne solution pour intégrer l'option de détection des objets à déplacer dans notre système.

IV Conclusion

Ce chapitre représente une vue globale du cadre général du projet. En effet, nous avons présenté la robotique et les différents types de robots. Ensuite, nous avons détaillé la description du robot que nous allons commander « Lynx 6.0 » en citant les composants matériels qui le constituent et leurs caractéristiques. Enfin, nous avons présenté la problématique de notre travail ainsi que la solution proposée.

Chapitre 2 : Conception

Introduction

Dans ce chapitre, nous allons présenter les principaux diagrammes utilisés pour notre système. Ensuite nous parlerons des deux cartes utilisées dans notre projet, ainsi que les caractéristiques de leurs composants.

I. Cahier de charge

Nous souhaitons réaliser un robot qui puisse détecter la présence d'un objet et le déplacer dans un emplacement bien déterminé.

L'alimentation du robot est assurée par une batterie de 9V. Cette source d'énergie n'est pas suffisante et nous aurons des problèmes puisqu'elle va être consommée rapidement ce qui nous oblige d'utiliser deux sources d'alimentation infinies. La première est pour la carte et la deuxième est pour le robot (les servomoteurs) :

- La première source d'alimentation sera l'ordinateur mais nous allons utiliser la carte « Arduino Uno » qui va jouer aussi le rôle d'un circuit intermédiaire qui va fournir une tension égale à 5V pour la carte SSC-32. Comme il est indiqué dans la figure ci-dessous nous avons utilisé deux fils conducteurs pour brancher deux pins de la carte Arduino, 5V et GND, respectivement aux deux pins de la carte SSC-32, un premier portant un signe '+' et un deuxième qui porte le signe '-'.



Figure 1 : Première source d'alimentation

- La deuxième source d'alimentation sera un transformateur fournissant une tension égale à 9V pour les servomoteurs.



Figure 2 : Transformateur à 9V

Contraintes liées au fonctionnement du robot :

- La première contrainte est liée aux caractéristiques du capteur GP2D12. Ainsi, la détection de l'objet à déplacer ne peut se faire que pour une distance comprise entre 10 cm et 80 cm.
- La deuxième contrainte est liée aux servomoteurs ayant des angles de rotation qui se limitent entre 0° et 180°.

II. Schéma Synoptique

En présentant le schéma synoptique ci-dessous, nous allons éclaircir la structure de notre système afin de le mieux comprendre.

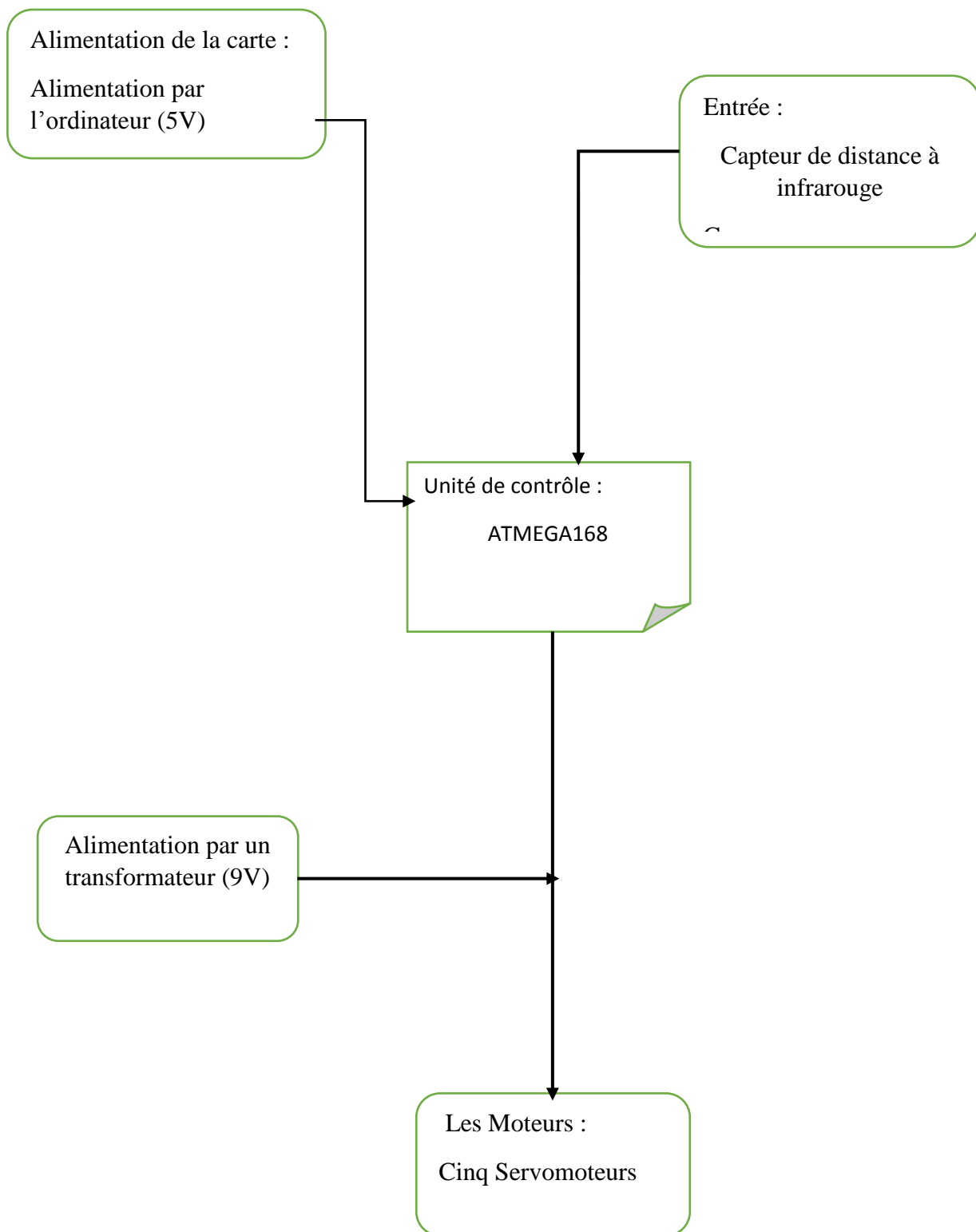


Figure 3 : Schéma synoptique

III. Organigramme

Le diagramme est une suite de directives composées d'actions et de décisions qui doivent être exécutées selon un enchaînement strict pour réaliser une tâche (appelée : séquence).

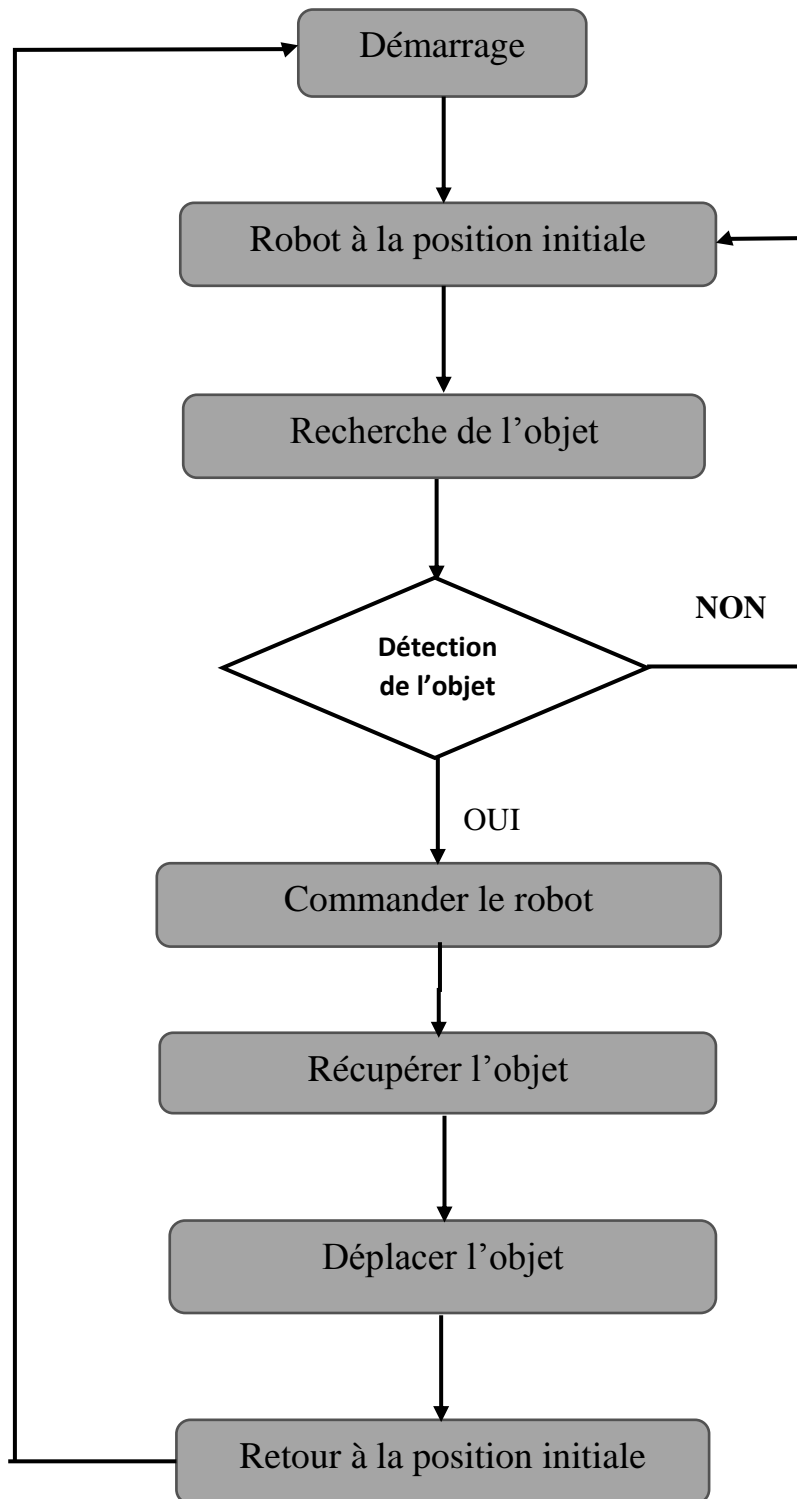


Figure 4 : Organigramme

IV. La Carte SSC-32

Comme nous avons dit dans le premier chapitre, nous allons utiliser la carte SSC-32 comme un circuit intermédiaire entre la carte Arduino et notre robot. C'est pour cela nous devons savoir les détails techniques cette carte afin de déterminer les composants nécessaires pour la liaison entre la carte Arduino et le robot lynx 6.0.

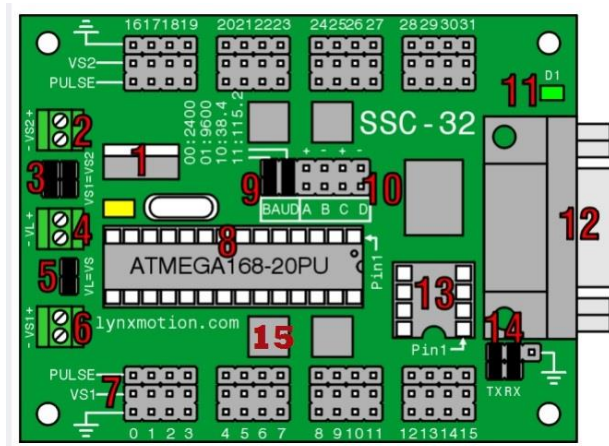


Figure 5 : La Carte SSC32

IV.1 Détails Techniques

Sur cette carte, il y a 32 canaux générateurs de pulsions pour les servomoteurs. Chacun d'eux peut être contrôlé individuellement et configurable afin de faire varier la position angulaire de l'axe du servomoteur.

- 1 - Sur cette carte SSC-32, un régulateur permet de contenir la tension d'alimentation de la carte à 5V. Il peut accepter jusqu'à 9V et 500mA.
- 2 et 6 - L'alimentation des servomoteurs se fait grâce aux borniers « Vs1 » et « Vs2 ». La tension est à régler en fonction de la valeur préconisée par le fournisseur des servomoteurs. Sur la carte, est indiqué l'emplacement de branchement des fils de tension (+) et de la masse (-). Ecart de 2,54mm.
- 3 - Un bornier permet de lier la tension d'alimentation des servomoteurs de 0 à 15 avec celle des servomoteurs de 16 à 31. Si le cavalier n'est pas placé, deux tensions différentes peuvent être branchées.
- 4 - L'alimentation générale de la carte se fait par le biais de ce bornier. La tension peut être de 6 à 9V en lien avec les valeurs autorisées par le régulateur (1).
- 5 - Un cavalier autorise l'utilisateur à lier l'alimentation générale avec celles des servomoteurs si elles sont égales.

- 7 - Les servomoteurs doivent être reliés sur une des deux lignes prévues en fonction des alimentations branchées. Les numéros correspondants sont à retenir pour pouvoir les configurer par liaison RS232 ou UART. Le sens de branchement est indiqué sur la carte (GND, Vs, Pulse). Ecart de 2,54mm.
- 8 - Le microcontrôleur Atmel ATMEGA168_20PU gère les commandes des servomoteurs en fonction des instructions et données envoyées par l'utilisateur. La fréquence de travail est de 14,75MHz. Il possède 32 sorties (PWM ou TTL) ainsi que 4 entrées (analogiques ou numériques).
- 9 - La vitesse de communication avec le PC ou une carte auxiliaire est réglable manuellement grâce aux cavaliers. Les valeurs possibles sont : 2400, 9600, 38400 ou 115200Bauds. Les positions correspondantes au débit sont inscrites sur la carte.

Débit	Cavalier1	Cavalier2
2400	Vide	Vide
9600	Présent	Vide
38400	Vide	Présent
115200	Présent	Présent

Tableau 1 : modification des cavaliers (jumpers)

- 10 - Il y a 4 entrées analogiques (A,B,C et D) de 8bits. Sur le même connecteur, sont présents, 2 alimentations 5V (+) et 2 masses (-). Ceci permet à l'utilisateur de câbler un potentiomètre, un bouton poussoir ou un capteur analogique.

Ces entrées analogiques retournent 0 lorsque la tension d'entrée est nulle, et 255 lorsque celle-ci est à 5V. La résolution est de 20mV.

- 11 - Une LED verte signale lorsqu'elle est allumée que le microcontrôleur est alimenté. Si celle-ci clignote, cela signifie que des données transitent sur la liaison RS232 ou UART avec un débit correct. Si le débit est incorrect, la LED ne clignotera pas.
- 12 - Connecteur DB9 femelle pour la liaison RS232. Un adaptateur RS232toUSB est nécessaire pour communiquer au PC par connexion USB.

- 13 - Une mémoire EEPROM stocke le programme s'il est trop volumineux. C'est un composant 24LC256 de 32Kilo Octets.
- 14 - Deux cavaliers permettent de lier la liaison RS232 au microcontrôleur. Sinon la ligne de connecteur 3 broches est utilisée pour une communication UART avec une carte auxiliaire (Tx, Rx, Gnd) avec Tx pour émettre des données, Rx pour recevoir des données et la masse.
- 15 - C'est un registre de décalage, « Shift Register ». Grâce à ce composant électronique, le microcontrôleur peut contrôler 32 servomoteurs. Chaque registre de décalage est responsable sur 8 servomoteurs. [N4]

IV.2 Le Microcontrôleur ATMEGA168PU20

L'ATMEGA168PU20 est un microcontrôleur 8 bits fabriqué par la société américaine Atmel.



Figure 6 : ATMEGA168PU20

➤ Caractéristiques

C'est un microcontrôleur qui fonctionne à 20 MHz maximum et possède :

- ✓ Style de montage: Through Hole
- ✓ Cœur : AVR
- ✓ Largeur du bus de données : 8 bits
- ✓ Taille de la mémoire du programme : 16 KB
- ✓ Taille de la RAM de données : 1 KB
- ✓ Résolution CAN : 10 bit
- ✓ Tension d'alimentation de fonctionnement : 2.7 V to 5.5 V
- ✓ Température maximale de fonctionnement : +85 °C
- ✓ Température minimale de fonctionnement : -40
- ✓ Gamme de processeur : megaAVR

- ✓ Type de RAM de données : SRAM
- ✓ Taille de la ROM de données : 512 B
- ✓ Type de ROM de données : EEPROM
- ✓ Type d'interface : 2-wire, SPI, Serial,USART
- ✓ Nombre de pins : 23 pins d'entrée et de sortie

➤ **Bloc Diagramme**

Le bloc diagramme est une représentation graphique simplifiée d'un procédé relativement complexe impliquant plusieurs unités ou étapes. Il est composé de blocs connectés par des lignes d'action.

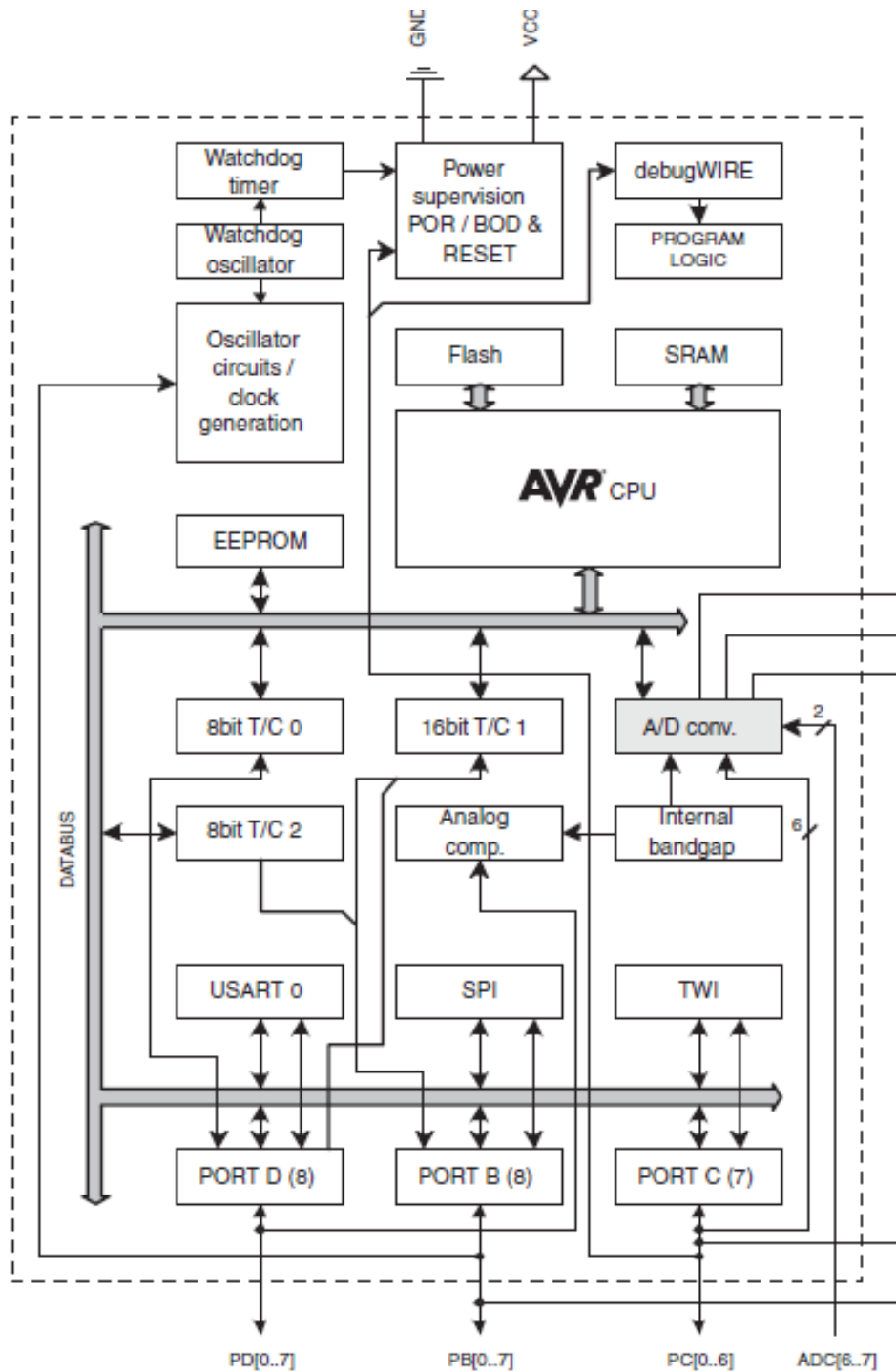


Figure 7 : bloc diagramme L'ATMEGA 168PU20 [B2]

V. La carte Arduino Uno

La carte Arduino Uno est une carte à microcontrôleur basée sur l'ATmega328. Elle dispose :

- ✓ De 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM (Pulse Width Modulation))
- ✓ De 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- ✓ D'un quartz 16Mhz,
- ✓ D'une connexion USB,
- ✓ D'un connecteur d'alimentation jack,
- ✓ D'un connecteur ICSP (programmation "in-circuit"),
- ✓ Et d'un bouton de réinitialisation (reset).

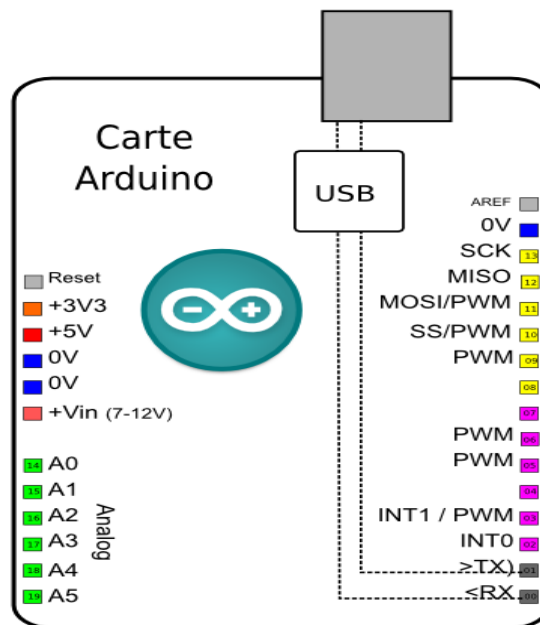


Figure 8 : Brochage de la carte Arduino Uno

V.1. Caractéristiques

Le tableau ci-dessous nous montre la synthèse des caractéristiques de la carte Arduino Uno.

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 Boot Loader
Mémoire SRAM (mémoire volatile)	2 KB (ATmega328)
Mémoire EEPROM (mémoire non volatile)	1 KB (ATmega328)
Vitesse d'horloge	16 MHz

Tableau 2 : Caractéristiques de l'Arduino Uno[N5]

V.2 Entrées et Sorties 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" 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 :

- **Communication Série** : Broches 0 (RX) et 1 (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 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.

V.3 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 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.

V.4 Le microcontrôleur ATMEGA328

Le microcontrôleur ATMEGA328 a la même famille que le microcontrôleur ATMEGA168PU20, de plus ces deux microcontrôleurs ont la même fiche technique (datasheet). Ce qui nous montre que ces deux composants électroniques ont les mêmes caractéristiques et le même bloc diagramme.

VI. Conclusion

Nous avons vu dans ce chapitre l'importance et l'utilité de la conception du système pour respecter le cahier de charge de notre robot. Nous avons présenté les différents composants électroniques utilisés pour accomplir notre travail. Le prochain chapitre s'intéresse à la réalisation de notre projet.

Chapitre 3 : Réalisation du projet

Introduction

Dans ce chapitre, nous allons prendre en compte les exigences du cahier de charge et la conception que nous avons élaboré dans le chapitre précédent pour réaliser la commande du robot Lynx 6.

I.1 Hardware :

I.1. Présentation :

Un servomoteur est en général constitué d'un moteur à courant continu, d'un système de réduction de vitesse à engrenage, d'un capteur de position et d'un circuit électronique d'asservissement commandé par un signal MLI. Il nécessite trois connections :

- Alimentation (6V-9V)
- Masse (0V).
- Contrôle (Modulation à Largeur d'Impulsions ou bien PWM).

I.2. Fonctionnement du servomoteur :

Les servomoteurs sont commandés par modulation de largeur d'impulsion des signaux qui commandent la position de l'actionneur d'asservissement. Les largeurs d'impulsion varient de 500 à 2500 microsecondes, ce qui correspond à un angle de rotation d'environ 0 à 180 °.

Une largeur d'impulsion de 1500 microsecondes va régler le servomoteur en position neutre égale à 90 °. L'amplitude de mouvement et de l'orientation utilisés dans le bras robotique Lynx6 sont résumées ci-dessous.

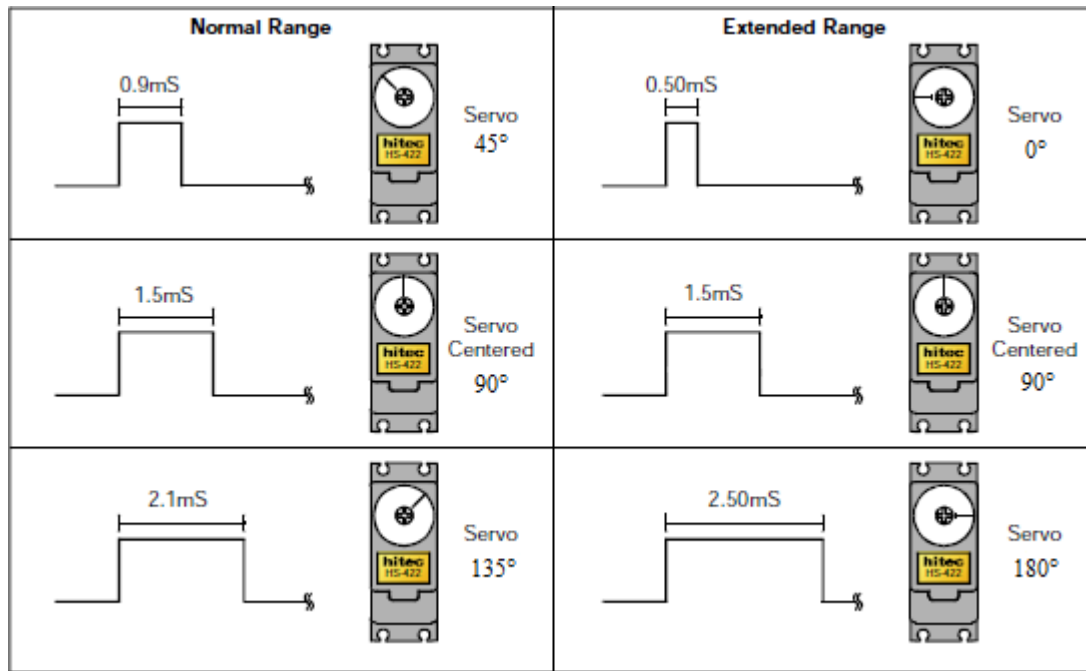


Figure 1 :Types des servomoteurs [B3]

La figure ci-dessus nous montre qu'il existe deux types d'intervalle de largeur d'impulsions.

- ✓ Le premier intervalle varie entre 900µs et 2100µs qui est équivalent à 45° et 135°. C'est l'intervalle normal. Dans notre robot, ce qui correspond au servomoteur est attaché à la pince.
- ✓ Le deuxième cas, c'est l'intervalle étendu qui se limite entre 500µs et 2500µs et qui nous donne des orientations entre 0° et 180°. Le reste des servomoteurs du robot ont ce type d'intervalle de largeur d'impulsions.

I.3 Fonctionnement du Capteur GP2D12 :

Nous allons commencer par décrire la connexion entre le capteur de distance GP2D12 et la carte ARDUINO.

➤ Connection :

- ✓ Le fil issu de l'émetteur de la lumière Vout doit être lié au pin d'entrée ADC (pin A0 de la carte Arduino Uno).
- ✓ Le fil au milieu est relié au GND (0V).
- ✓ Le fil issu du détecteur de la lumière se connecte au Vcc (alimentation).

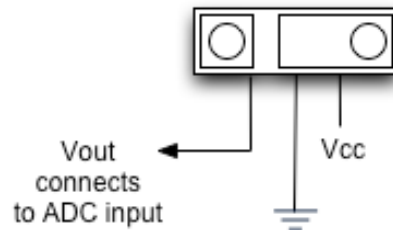


Figure 2 :Vue de face du GP2D12

➤ **Bloc diagramme :**

Le bloc diagramme du capteur GP2D12 est le suivant :

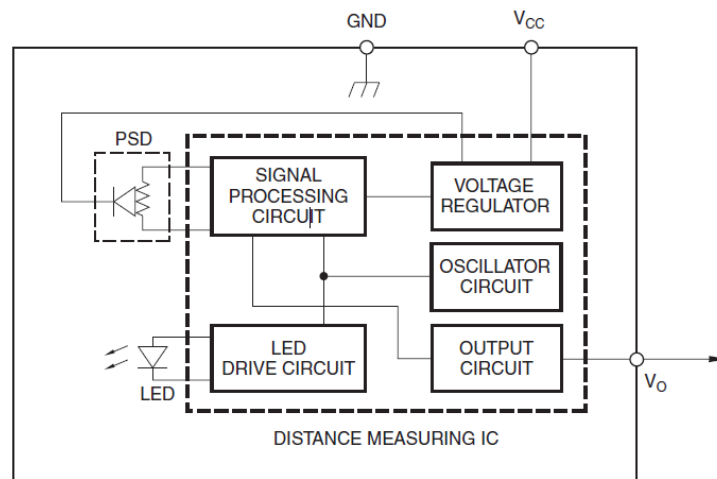


Figure 3 :bloc diagramme du GP2D12 [B4]

➤ **Principe de fonctionnement :**

Ce capteur utilise un ingénieux système optique. Une LED infrarouge émet un rayon infrarouge invisible à l'œil nu qui est réfléchi par l'objet à déplacer. Une barrette photo-réceptrice(PSD) reçoit le rayon réfléchi, ce qui permet d'en déduire l'angle de réflexion et donc la distance.

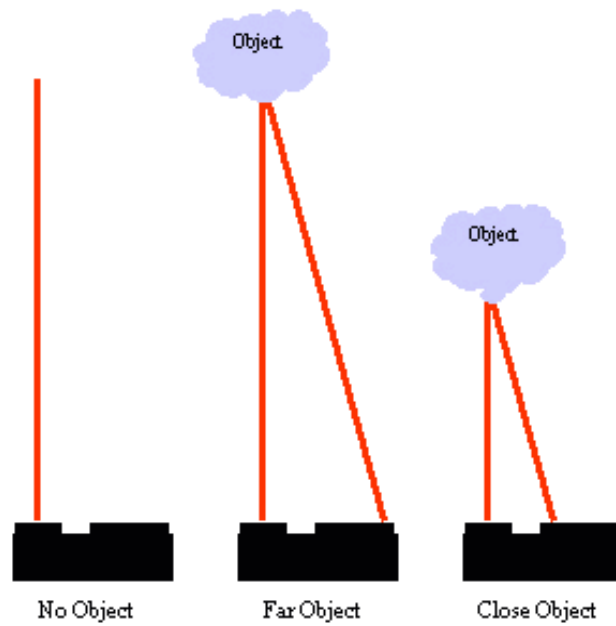


Figure 4 :Détection de l'objet

➤ **Calcul de la distance :**

La courbe liant la tension de sortie du capteur GP2D12 et la distance qui le sépare à l'objet est présentée dans la fiche technique de ce capteur GP2D12 comme il est indiqué dans la figure ci-dessous :

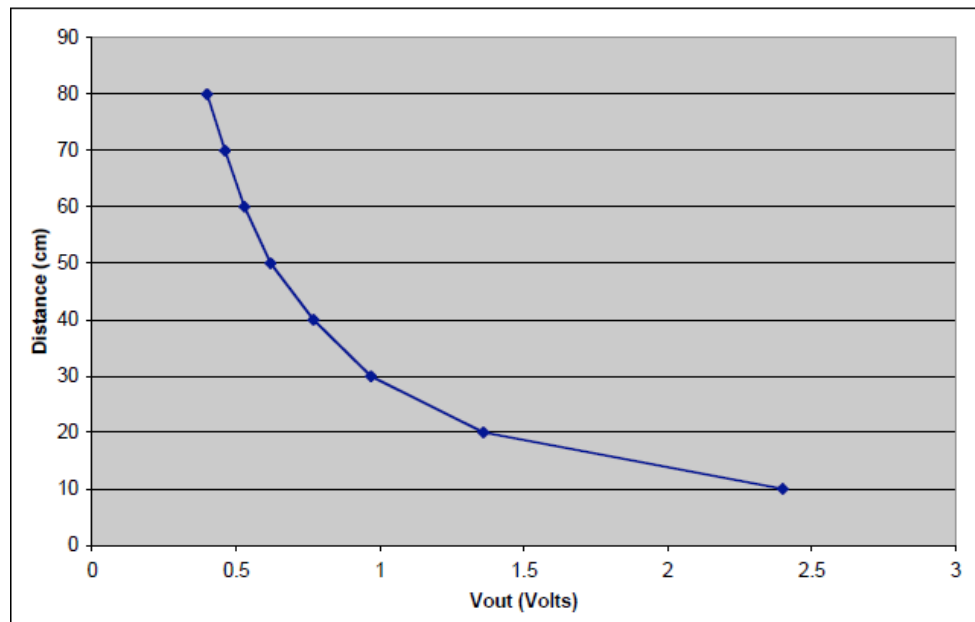


Figure 5 :la courbe Distance en fonction de Vout

Nous remarquons que la courbe ci-dessus n'est pas une droite. Alors nous ne pouvons pas appliquer la formule ($y=ax+b$). C'est pour cela, le fabricant de ce capteur, la société Sharp, a défini une fonction qui transforme la courbe présentée dans la figure précédente en une droite. Ce qui est illustré dans la figure ci-dessous.

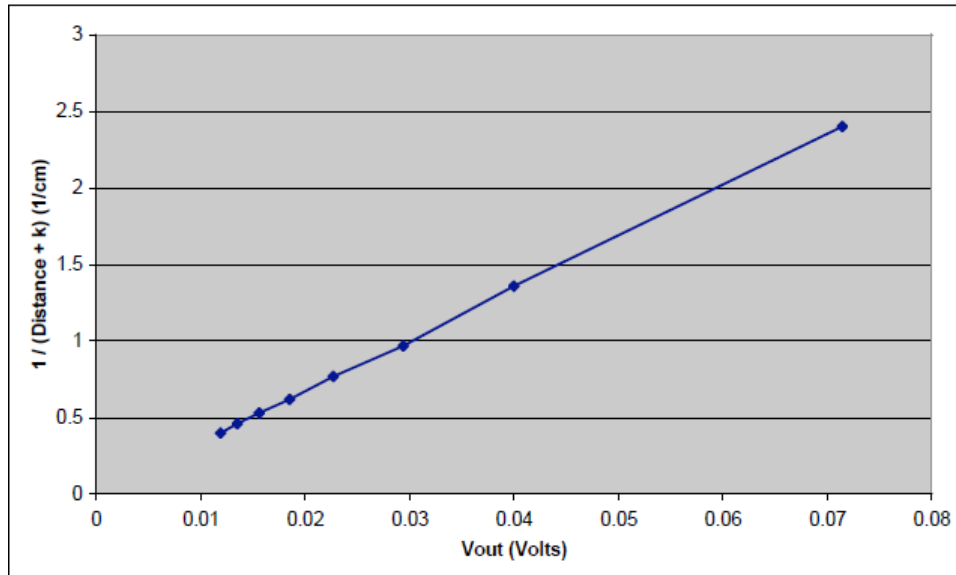


Figure 6 :La courbe $[1/(Distance+k)]$ en fonction de Vout.

Nous avons une droite affine d'équation $y=ax+b$ avec $y=1/(Distance+k)$, ou k est une constante égale à 4 définie par la société Sharp pour le capteur GP2D12. L'abscisse de cette droite, $x=Vout$. En organisant les termes de l'équation nous obtenons la formule suivante :

$$Distance = (1 / (a * Vout + b)) - 4$$

➤ **Branchement du GP2D12 avec la Carte Arduinouno :**

Afin de tester notre capteur, nous avons fait le montage suivant :

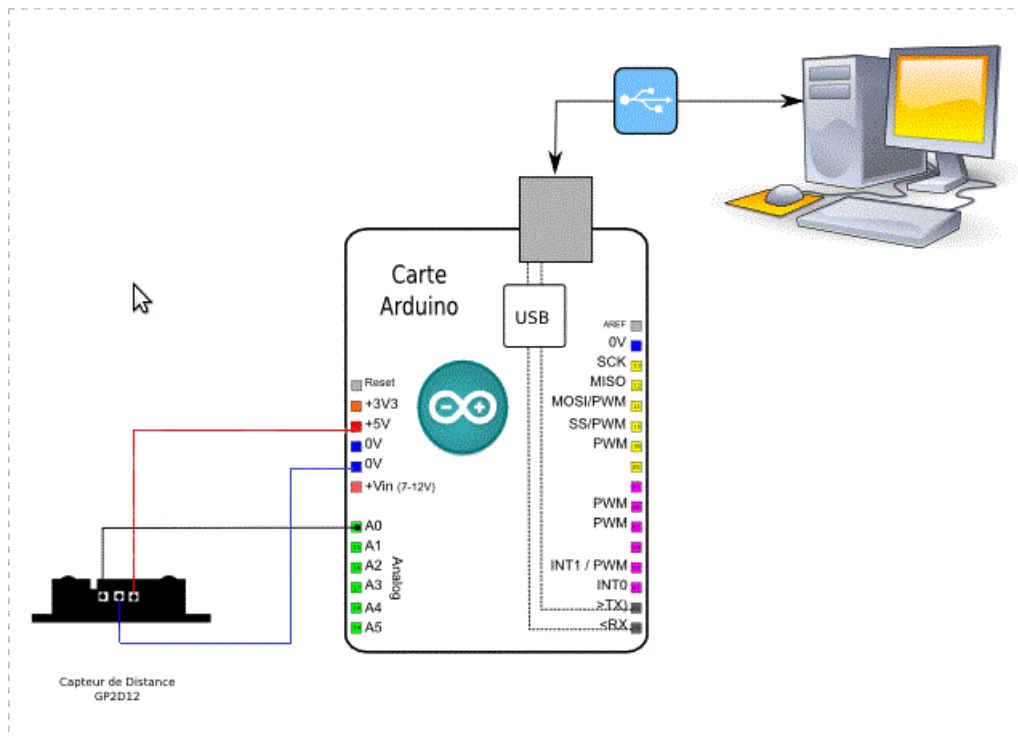


Figure 7 : Montage de test du capteur avec l'ArduinoUno

I.4 La connexion entre les cartes ArduinoUno et SSC-32 :

Afin de commander notre robot, nous devons savoir comment brancher les deux cartes ArduinoUno et SSC-32 et quels pins nous devons utiliser.

Comme nous avons dit dans le chapitre précédent, la carte SSC-32 possède deux types de communication série.

- Le premier s'agit d'un connecteur DB9 femelle pour la liaison série RS232 avec l'ordinateur à travers un adaptateur RS232toUSB. Ce mode de communication série est activé lorsque les deux cavaliers sont mis sur les pins TX et RX.



Figure 8 : communication série via le port RS232

- Le deuxième type est d'utiliser la communication série TTL à travers les deux pins TX et RX. Pour activer ce type de communication série nous devons enlever les deux cavaliers de ces pins ce qui nous permet la communication UART avec une carte auxiliaire.

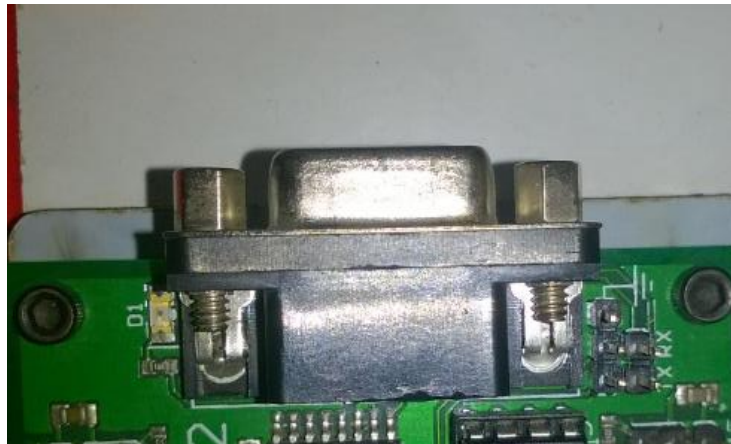


Figure 9 : mode communication série via les pins RX TX

La carte ArduinoUno possède aussi la communication série TTL comme il est mentionné dans le deuxième chapitre. Vu que la carte SSC-32 va se comporter comme un circuit intermédiaire entre le robot et l'Arduino, nous allons utiliser le pin RX (réception des données). Et pour la carte qui va commander le robot, la carte Arduino, nous allons utiliser le pin TX (transmission des données). Alors nous devons brancher le pin TX et le GND de l'Arduino respectivement par le pin RX et le GND de la carte SSC-32.



Figure 10 :Communication série entre les deux cartes

II. Software:

II.1 Arduino Integrated Development Environment (IDE):

Le logiciel de programmation des modules Arduino est une application Java, libre et multiplateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison série (RS-232, Bluetooth ou USB selon le module). Le langage de programmation utilisé est le C++, compilé avec avr-g++ 3, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino, à toute personne maîtrisant le C ou le C++.

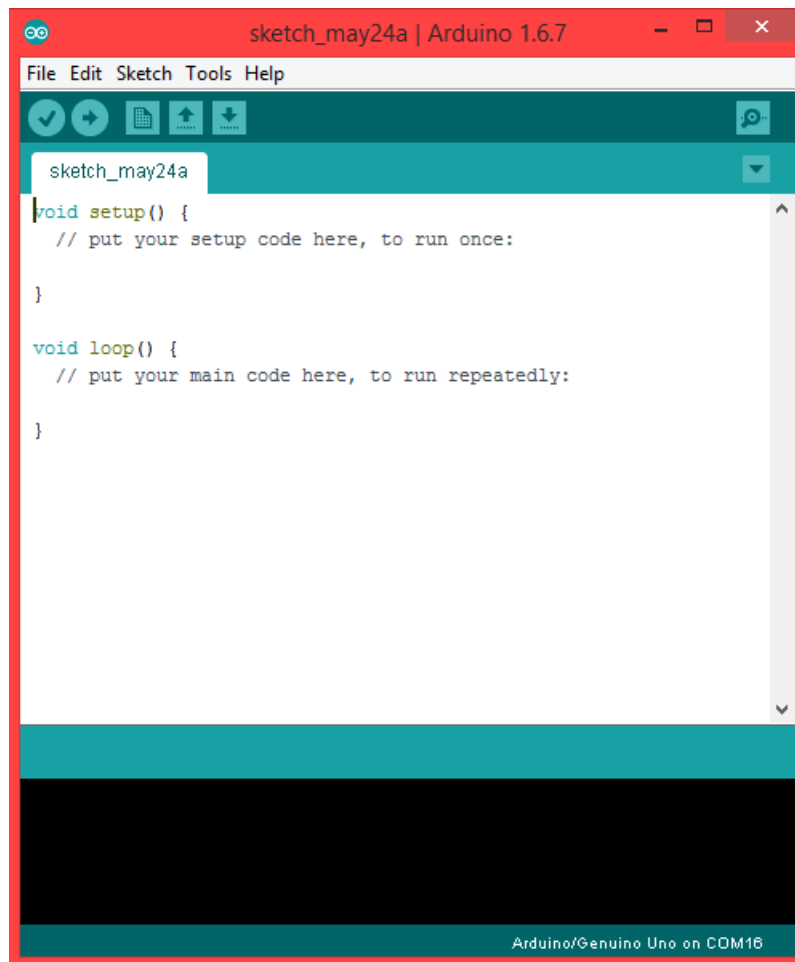


Figure 11 :Arduino IDE

II.2 LynxTerm :

Le LynxTerm est un logiciel développé par la société Lynxmotion. Ce logiciel nous permet de tester les différents servomoteurs branchés sur la carte SSC-32. Une fois la carte reliée à l'ordinateur, il faut sélectionner le port « COM » correspondant et cliquer sur « Connect » ensuite, en cliquant sous « Channel » nous pouvons choisir le servomoteur à tester.

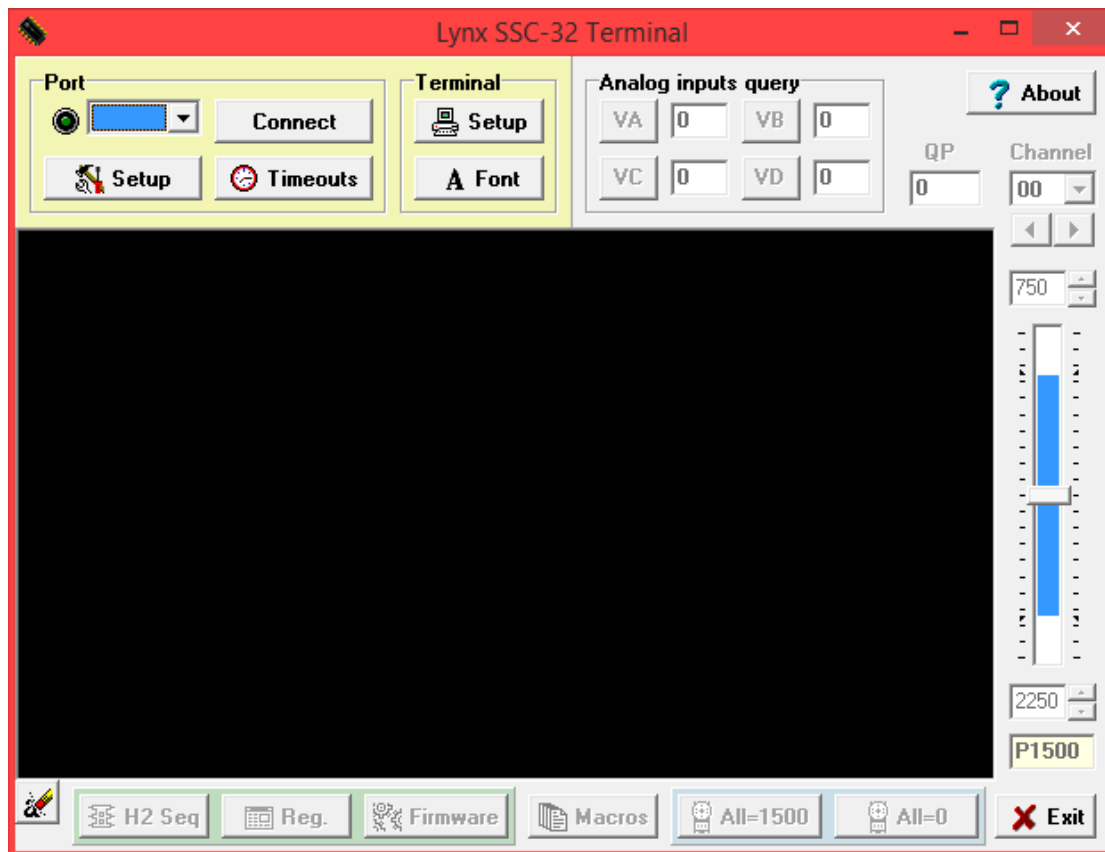


Figure 12 :LynnxTerm

III. Application :

Dans cette partie, nous allons appliquer notre scénario qui a été représenté par l'organigramme dans le deuxième chapitre. Mais d'abord nous devons passer par la réalisation du montage de notre robot.

III.1 Montage du système :

Avant de passer à la partie de la programmation nous devons faire le montage qui est introduit dans la figure ci-dessous.



Figure 13 :Montage du robot

III.2 Programmation :

III.2.1 Commande des servomoteurs :

Nous avons déjà dit au début de ce chapitre que les servomoteurs peuvent être commandé en les envoyant l'angle de rotation ou bien la largeur d'impulsions. Dans notre cas nous allons utiliser la largeur d'impulsions qui varie entre 500 μ s et 2500 μ s.

Commençons par l'ouverture de l'Arduino IDE.

Premièrement, nous allons avoir le « void setup function » dans laquelle nous allons choisir la vitesse de communication de la carte Arduino (Baudrate) qui doit être égale à celle de la carte SSC-32. Nous allons le fixer en 115200 Bauds en écrivant comme suit :

```
void setup()
{
  Serial.begin(115200);
}
```

Maintenant sous la fonction « voidloop » nous allons écrire les commandes avec lesquelles nous contrôlons le robot Lynx 6. Par exemple nous souhaitons positionner le servomoteur en position initiale. Cela signifie que la largeur d'impulsions de chaque moteur va être égale à 1500µs qui est équivalent à 90°.

```
void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.write("#0P1500\r");
  Serial.write("#1P1500\r");
  Serial.write("#2P1500\r");
  Serial.write("#3P1500\r");
}
```

Suite à l'exécution de ce code nous obtenons le résultat suivant :

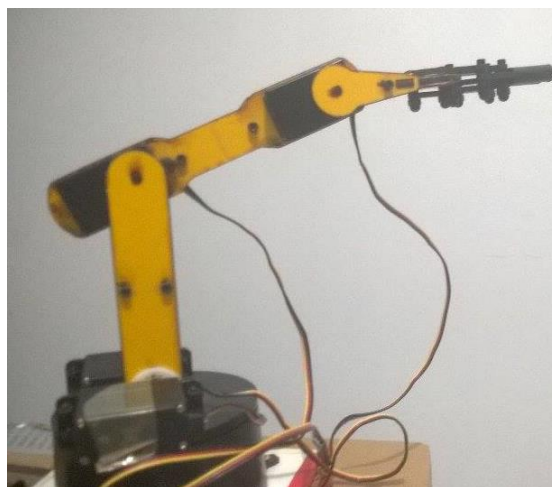


Figure 14 :Robot en position initiale

Si nous prenons la première commande `Serial.write("#0P1500\r");`

Cette commande permet de positionner le premier servomoteur (la base) de maintenir la position neutre qui est égale à 1500µs (90°). Le « #0 » est pour choisir le servomoteur connecté au pin 0 de la carte SSC-32 et « P1500 » représente la largeur d'impulsion choisi.

Le deuxième exemple consiste à créer une boucle infinie qui permet de faire une rotation du 0° à 180° de la base du robot dans une durée égale à 5 secondes puis attendre 3 secondes et faire le contraire et ainsi de suite. Le code est le suivant :

```
#include <SoftwareSerial.h>
void setup()
{
    Serial.begin(115200);
}

void loop()
{
    Serial.write("#0P500 T5000\r");
    delay(3000);

    Serial.write("#0P2500 T5000\r");
    delay(3000);
}
```

« T5000 » signifie la durée de rotation (5000 ms).

III.2.2 Programmation du capteur GP2D12 :

Après le branchement de notre capteur avec la carte ArduinoUno, nous passons pour la partie du code qui va nous permettre de détecter la présence d'un objet en déterminant la distance qui sépare celui-ci et le GP2D12.

Nous commençons par définir des variables qui vont s'intervenir dans la détermination de la distance.

```
int IR_SENSOR = 0;
int intSensorResult = 0;
float fltSensorCalc = 0;
```

La première instruction permet de déclarer une variable « IR_SENSOR » qui représente le détecteur lié au pin d'Analog A0 de la carte ArduinoUno.

La deuxième variable « intSensorResult » est utilisée pour l'enregistrement de la valeur lue au niveau de l'Analog A0 afin de l'utiliser pour calculer « fltSensorCalc » (variable qui représente la distance).

La fonction utilisée pour obtenir la valeur de l'Analog A0 est la suivante :

```
intSensorResult = analogRead(IR_SENSOR);
```

Ensuite nous faisons le calcul de la distance en utilisant la formule définie par la société Sharp et nous terminons par l'affichage des distances obtenus.

```
fltSensorCalc = (6787.0 / (intSensorResult - 3.0)) - 4.0;
```

```
Serial.print(fltSensorCalc);
```

```
Serial.println(" cm");
```

```
delay(200);
```

L'instruction « delay(200) » permet de nous afficher les distances détectées chaque 200ms.

Le code complet du capteur GP2D12 est le suivant :

```
int IR_SENSOR = 0;
```

```
int intSensorResult = 0;
```

```
float fltSensorCalc = 0;
```

```
void setup()
```

```
{
```

```
Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
intSensorResult = analogRead(IR_SENSOR);
```

```
fltSensorCalc = (6787.0 / (intSensorResult - 3.0)) - 4.0;
```

```
Serial.print(fltSensorCalc);
```

```
Serial.println(" cm");
```

```
delay(200);
```

```
}
```

En exécutant le code nous allons visualiser dans le « Serial Monitor » de l'Arduino IDE les différentes valeurs de la distance. Si on bouge le capteur la distance va changer, soit elle augmente soit elle diminue.

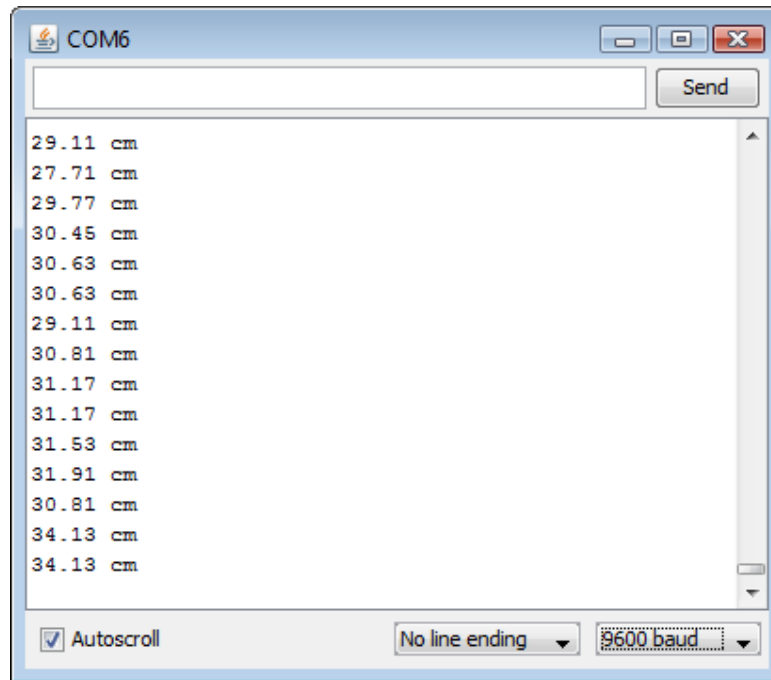


Figure 15 : Serial Monitor

IV. Conclusion :

Nous avons vu dans ce chapitre le matériel et les logiciels utilisés pour réaliser notre travail. Nous avons fait des modifications au niveau du matériel existant ainsi que des expérimentations pour vérifier le bon fonctionnement du robot et améliorer ses performances.

Conclusion Générale

L'objectif de notre projet de fin d'études est de concevoir et développer une solution pour commander le robot Lynx 6. Ce robot doit détecter la présence d'une pièce et par la suite la récupérer et la déplacer dans un emplacement bien défini. Ce travail a été élaboré durant trois mois à l'institut supérieur d'informatique.

Au terme de ce projet, nous pouvons dire que nous avons respecté le cahier de charges et que notre robot a été commandé avec succès. Ce rapport présente une description des étapes que nous avons suivi afin de réaliser notre travail. Nous avons présenté en premier lieu la robotique et spécialement le robot Lynx 6 que nous allons commander. Ensuite nous avons fixé les objectifs à atteindre et la problématique vécue. Puis, nous avons accédé à la conception du robot où nous avons présenté un schéma synoptique qui résume la composition de notre système et un organigramme qui présente le scénario que nous voulons réaliser. Et bien sûr nous avons parlé des deux cartes utilisées en détails afin de réussir à commander notre robot. Dans la dernière partie intitulée « Réalisation », nous avons commandé le robot Lynx 6.

Ce travail nous a permis d'enrichir nos connaissances dans plusieurs domaines et notamment dans la robotique et la conception des cartes électroniques. Nous pouvons enrichir notre projet dans le futur par d'autres fonctionnalités à savoir l'utilisation des coordonnées de l'objet à déplacer dans la commande et à ce moment nous avons recours à la modélisation géométrique inverse du robot.

Finalement, nous sommes satisfaits par le travail en groupe qui nous a apporté le partage du savoir-faire et de la charge du travail en se basant sur nos compétences complémentaires et en fixant des objectifs communs. Cette méthodologie nous a offert une motivation et une grande rapidité d'exécution de travail.

Bibliographie

[B1] Cours de robotique, Jean-Louis Boimond, ISTIA-Université Angers-France :

Error! Hyperlink reference not valid.

[B2] Fiche technique de l'ATMEGA168PU20 :

<http://www.atmel.com/images/doc2545.pdf>

[B3] Fiche technique de la carte SSC-32 :

<http://www.lynxmotion.com/images/data/ssc-32.pdf>

[B4] Fiche technique du capteur GP2D12 :

http://www.sharpsma.com/webfm_send/1203

Nétographie

[N1][Http://tpe-la-robotique.e-monsite.com/pages/intruduction.html](http://tpe-la-robotique.e-monsite.com/pages/intruduction.html)

[N2]<https://fr.wikipedia.org/wiki/Servomoteur>

[N3]<http://wiki.cybedroid.com/index.php?title=SSC-32>

[N4]<http://wiki.cybedroid.com/index.php?title=SSC-32>

[N5]http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php