

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

De Master Académique en Automatique
Option : Commande des systèmes

Thème

*Commande à distance d'un robot rigide
à deux axes à base de la carte Arduino*

Encadré par :

Mr :MELLAH Rabah

Présenté par :

OUAKED Jugurtha

Devant le jury composé de :

Président : MAIDI Ahmed

Examineur : HAMICHE Hamid

Examinatrice : NAIT ABDESSELEM Aldjia

Soutenu le : 17/07/2016

Promotion 2016

DEDICACE

JE DÉDIE LE FRUIT DE CE MODESTE TRAVAIL COMME
UN GESTE DE GRATITUDE À :

MES TRÈS CHERS PARENTS, QUI M'ONT SOUTENU,
ENCOURAGÉ POUR QUE JE PUISSE MENER À BIEN MES
ÉTUDES, ET QUI ATTENDU CE JOUR AVEC IMPATIENCE.

MON FRÈRE
MES TANTES : ZAKIA ET RAZIKA.
LES FAMILLES :OUAKED ET OUHARCHAOU
MA TRÈS CHER GRAND-MÈRE :FÉRROUDJA
MES ENSEIGNANTS ET MES AMIES DE L'ÉTUDE.
À TOUT CEUX QUI ONT CONTRIBUÉ À LA RÉALISATION DE
CE TRAVAIL

Jugurtha

Remerciements

Je tiens à remercier profondément Dieu, le tout puissant de m'avoir donné le courage, la volonté, ainsi que la santé pour réaliser ce modeste travail.

Je tiens à exprimer mes vifs remerciements et mes sincères reconnaissances à mon promoteur Mr MELLAH Rabah, pour sa disponibilité, ces conseils judicieux, ses directives et ses orientations et d'avoir accepté de diriger ce projet et de m'avoir soutenues au long travail tout en m'accordant sa confiance et en me fait profiter de sa large expérience tout au long de la réalisation de ce projet de fin d'étude.

Mes profondes gratitudes, d'avance aux membres du jury qui feront l'honneur d'évaluer ce travail de fin d'études.

Tous mes infinis remerciements vont à tous les enseignants qui ont contribué à ma formation durant mon cursus, pour le riche savoir qu'ils m'ont transmis avec rigueur et dévouement.

Enfin, j'exprime ma reconnaissance envers toute ma famille, tous mes amis, camarades et envers tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail ne serait-ce par un modeste sourire d'encouragement à mon égard.

Sommaire

Introduction général.....	1
Chapitre 1 : généralité sur les robots rigide a deux axes	
I) Introduction.....	2
II.1) définition.....	2
II.2) historique.....	2
III) La télé opération.....	3
III.1) Le protocole de transmission zigbée.....	3
III.2) Présentation de Zigbee.....	3
III.2.1) Les modems.....	3
II.2.1.1) La modulation.....	4
III.2.1.2) Principe de la modulation ASK.....	5
III.2.1.3) Principe de la modulation FSK.....	5
III.3) Le réseau Zigbée.....	6
III.3.1) Fonction « SLEEP ».....	7
III.3.2) Adressage.....	7
III.3.3) Sécurité.....	7
III.3.4) Communication séries.....	7
III.3.5) Adressage des modules.....	7
IV) L'arduino.....	8
IV.1) Origine du nom.....	9
IV.2) Description.....	9
IV.3) Matériel.....	9
IV.4) Matériel libre et logiciel libre.....	10
V) Asservissement visuel.....	10
V.1) Introduction.....	10
V.2) Commande.....	10
VI) Conclusion.....	11

Sommaire

Chapitre 2 : conception d'un robot rigide a deux axes

I) Introduction.....	12
II) Partie mécanique.....	12
II.1) System de guidage.....	12
II.1.1) Douilles à billes.....	12
II.2) Déplacement suivant les axes x y.....	13
II.2.1) Courroie.....	14
III) Partie électrique.....	14
III.1) Les actionneurs.....	14
III.1.1) Moteur pas à pas.....	14
III.2) Les pilotes du moteur pas à pas.....	20
III.2.1) Commande des moteurs pas à pas avec le pilot A4983.....	20
III.2.1.1) Matériel inclus.....	20
III.2.1.2) Connexions de Puissance.....	20
III.2.1.3) Connexions du moteur.....	21
III.2.1.4) Entrées de commande.....	21
III.2.1.5) Structure interne du pilote A4983.....	21
III.2.1.6) Un exemple d'un programme qui va commander un moteur pas à pas à l'aide d'un joystick.....	22
III.2.2) Commande des moteurs pas à pas avec L298N.....	23
III.2.2.1) Caractéristiques.....	24
III.2.2.2) Spécifications.....	24
III.2.2.3) Idées d'applications.....	24
III.2.2.4) Précautions.....	24
III.2.2.5) Détails techniques.....	25
III.2.2.6) raccordement sur une carte de contrôle.....	25
III.2.2.7) fonctionnement d'un moteur pas a pas.....	26
III.3) Les capteurs.....	28

Sommaire

III.3.1) Constitution.....	28
III.3.1.1) Caractéristiques.....	28
III.3.3) Symbolisation.....	29
III.3.4) Les capteurs mécaniques.....	29
III.3.4.1) Utilisation.....	30
III.3.4.2) Les avantages sont les suivants.....	30
III.3.5) Les capteurs inductifs.....	30
III.3.5.1) Avantages et utilisation.....	31
III.3.6) Les capteurs capacitifs.....	31
III.3.6.1) Avantages.....	32
III.3.7) Les cellules photoélectriques, la cellule en barrage.....	32
III.3.8) Les cellules photoélectriques, la cellule reflex.....	32
III.3.8.1) La détection par proximité.....	33
III.3.8.2) Avantages.....	33
III.4) Module Xbee.....	33
III.4.1) Présentation du matériel.....	34
III.4.2) Programmation des modules Xbee.....	36
III.5) Les caméras.....	40
III.5.1) Différents types de caméras.....	40
III.6) Arduino.....	42
III.6.1) Arduino uno.....	42
III.6.2) Arduino méga 2560.....	45
III.6.3) Programmation des cartes arduino.....	48
IV) conclusion.....	57

Chapitre3 : Réalisation pratique du robot.

I) introduction.....	58
II) Structure mécanique.....	58
II.1) Les différentes partie du robot.....	58

Sommaire

III) Assemblage complet de la partie mécanique du robot.....	60
IV) les différentes parties électriques du robot.....	61
IV.1) les cartes de commande du robot.....	61
IV.2) Les modules Xbee.....	62
IV.3) la caméra sans fil.....	63
IV.4) les actionneurs.....	63
IV.4.1) Caractéristiques des moteurs.....	63
V) Structure complète de la réalisation.....	64
VI) Organigramme.....	65
VII) conclusion.....	66
Conclusion général.....	67

Bibliographie.

Annexe.

Liste de figure

Liste des figures :

Figure 1 : robot porteur d'une perceuse.....	1
Figure 2 : communication série entre deux modems	4
Figure 3 : : modulation ASK.....	5
Figure 4 : modulation FSK.....	6
Figure 5 : élément essentiel d'un réseau zigbee.....	7
Figure 6 : planification de trajectoires.....	11
Figure 7 : Asservissement visuel à partir d'une base de données.....	11
Figure 8 : Douilles à billes.....	16
Figure 9 : Douilles à billes avec tige.....	17
Figure 10 : Module de guidage parallèle.....	17
Figure 11 : Courroie.....	18
Figure 12 : courroie avec poulie.....	18
Figure 13 : Aimant permanent.....	18
Figure 14: Hybride.....	19
Figure 15 : Reluctance Variable.....	19
Figure 16 : Vue d'un moteur à aimants permanents.....	19
Figure 17 : Vue d'un moteur hybride.....	20
Figure 18: Vue d'un moteur à reluctance variable.....	21
Figure 19 : Structure d'un moteur pas à pas (montage bipolaire).....	22
Figure 20 : Structure d'un moteur pas à pas (montage unipolaire).....	23
Figure 21 : structure du A4983.....	25
Figure 22 : pilot L298N.....	27
Figure 23 : détail technique du pilot L298N.....	29
Figure 24 : contrôle d'un moteur pas à pas.....	30
Figure 25 : symbolisation des capteurs.....	33
Figure 26 : capteurs mécaniques.....	33
Figure 27 : capteur inductif.....	34

Liste de figure

Figure 28 : capteurs inductifs.....	35
Figure 29 : capteurs capacitifs.....	35
Figure 30 : cellules photoélectriques (cellule de barrage).....	36
Figure 31 : cellules photoélectriques (cellule reflex).....	37
Figure 32: module Xbee.....	38
Figure 33: transmission point à point.....	40
Figure 34 : camera analogique.....	45
Figure 35: caméra discrete.....	45
Figure 36: caméra sans fil.....	46
Figure 37: Carte arduino face arrière.....	47
Figure 38 : face avant de la carte Arduino.....	48
Figure 39: La carte Arduino Mega 2560.....	49
Figure 40 : structure complète du robot.....	64
Figure 41: la structure de la base du robot.....	65
Figure 42: l'axe X.....	65
Figure 43: l'axe Y.....	66
Figure 44: courroie qui entraine l'axe Y.....	66
Figure 45: assemblage complet des deux axes avec la base.....	66
Figure 46: schéma de branchement.....	67
Figure 47: le datasheet du driver L298N.....	68
Figure 48: datasheet du driver A4983.....	68
Figure 49 : transmission d'information avec des Xbee.....	68
Figure 50 : positionnement de la caméra.....	69
Figure 51: moteur pas à pas Sanyo Denki.....	69
Figure 52 : Structure complète du robot.....	70

Liste de figure

Liste de figure

Introduction générale

C'est dernière années, les technologies de l'information et de la communication ont permis d'un côté, pour les activités du tertiaire de ne plus concentrer la production du service dans un lieu unique .D'un autre côté, ces technologies ont permis de dispenser la présence de l'homme dans des milieux dangereux ou tout simplement pour des missions d'exploration.

Dans ce projet, je vais présenter une forme de télétravail, celle liée au contrôle à distance des systèmes robotisés. Pour faire un bon contrôle à distance on doit choisir le bon protocole de transmission entre le maitre qui est l'opérateur et l'esclave qui est le robot, pour cela on doit faire référence à la meilleure carte de commande qui convient à notre travail et le bon actionneur pour avoir une grande précision. Puis faire en sorte que le système soit asservi pour cela on va utiliser les techniques asservissements visuels pour nous informer de la position ou il se trouve notre robot.



Chapitre 1

GENERALITES SUR LES ROBOT ET LA TELE OPERATION

I) Introduction :

Dans ce chapitre on présente un aperçu général sur la technologie de la commande à distance des robots citant le Protocol de transmission qui est basé sur zigbee, et aussi on parle de la carte de commande la plus connu et la plus utiliser dans le monde qui est la carte Arduino, de plus en explique d'une manière générale c'est quoi l'asservissement visuel.

II) définition d'un robot :

Un robot est un système mécanique poly-articulé mu par des actionneurs et commander par un calculateur qui est destiné à effectuer une grande variété de tâches.

II.1) historique :

1947 : premier manipulateur électrique télé-opéré.

1954 : premier robot programmable.

1961 : apparition d'un robot sur une chaine de montage de général Motors, premier robot avec control en effort.

1963 : utilisation de la vision pour commander un robot.

1973 : premier robot à roue.

II.2) différent catégorie de robot :

- Robot mobile à roue.
- Robot sous-marin.
- Robot volant.
- Robot humanoïdes.
- Robot manipulateur.

Domaine d'utilisation d'un robot a deux axes :

En général ce genre de robot il est utilisé dans des usines pour le déplacement des objets dans l'espace 2D. Nous on va l'utiliser pour porter une perceuse a un point bien précis afin d'effectuer sa tâche.



Figure 1 : robot porteur d'une perceuse

III) La télé opération :

La Télé opération désigne les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide d'un système robotique d'intervention (dispositif esclave), commandé à partir d'une station de contrôle (dispositif maître), par l'intermédiaire d'un canal de télécommunication. Pour ça il nous faut un protocole de transmission et notre choix a tombé sur le protocole zigbée.

III.1) Le protocole de transmission zigbée :

C'est quoi ?

– *Un ensemble de protocoles de communications de haut niveau*

- *Utilisant des transmission radio à faible consommation,*
- *Pour une transmission de données à faible débit (250 Kbit/s)*
- *Sur une faible étendue (WPAN)*

=> *basé sur la norme IEEE 802.15.4 ("Low-Rate Wireless Personal Area Network (LR-WPAN) standard")*

III.2) Présentation de Zigbee :

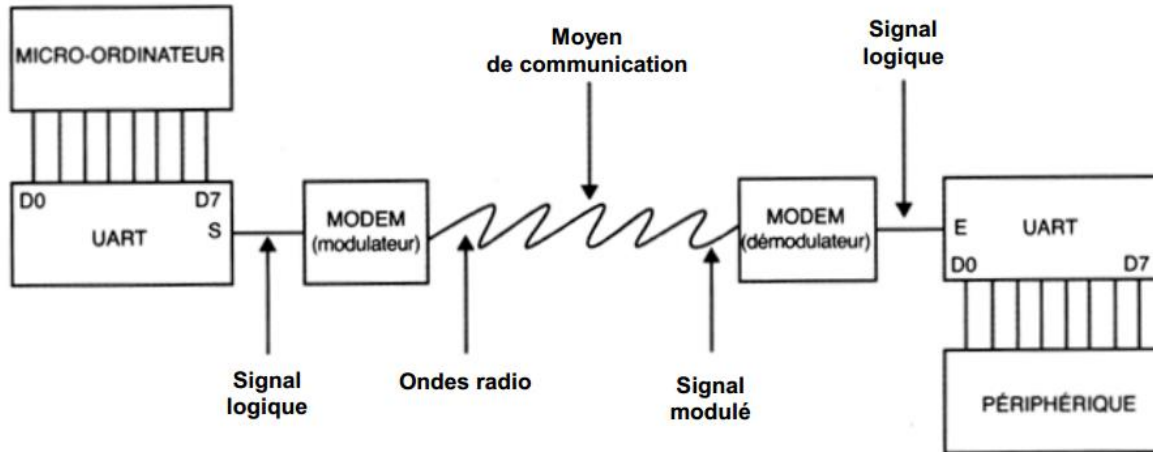
ZigBee est un protocole de haut niveau qui est proposé en 1998 et normalisé en mai 2003, puis 2006 permettant la communication de petites radios, à consommation réduite, basée sur la norme IEEE 802.15.4 pour les réseaux à dimension personnelle (WPANs).

C'est vraiment un très bon protocole, qui est bien plus simple à mettre en œuvre que le protocole Bluetooth.

En effet les appareils qui utilisent ce protocole de transmission on les appelle les modems ont pour rôle d'envoyer l'information à distance.

III.2.1) Les modems:

Un modem qui est l'acronyme de modulateur-démodulateur a pour fonction première de transformer les signaux numériques en signaux analogiques. Comme schématisé sur la figure page suivante, le modem s'intercale au sein d'une liaison numérique série lorsque celle-ci doit parcourir de grandes distances ou utiliser un support de transmission autre qu'un simple fil.



Le modem s'intercale au sein d'une liaison série

Figure 2 : communication série entre deux modems

Si les signaux numériques ne peuvent voyager sur de longues distances sans subir des déformations qui les rendent inutilisables, il n'en est pas de même des signaux analogiques qui peuvent tout à la fois voyager loin mais aussi voyager sur de très nombreux supports physiques : fils bien sûr mais aussi liaison radio, faisceaux infrarouges, fibres optiques, etc.

III.2.1.1) La modulation :

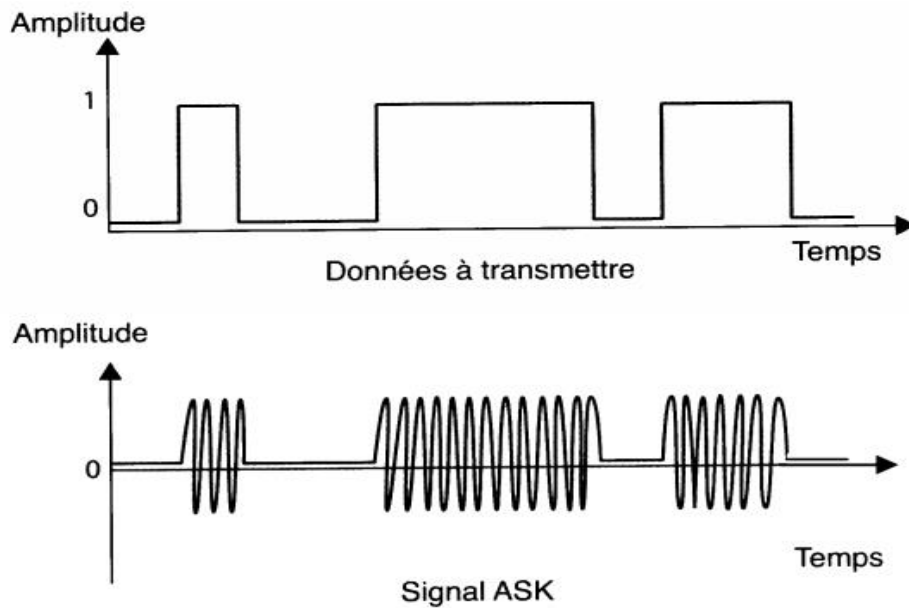
En télécommunications, le signal transportant une information doit passer par un moyen de transmission entre un émetteur et un récepteur. Le signal est rarement adapté à la transmission directe par le canal de communication choisi, hertzien, filaire, ou optique. La modulation peut être définie comme le processus par lequel le signal est transformé de sa forme originale en une forme adaptée au canal de transmission, par exemple en faisant varier les paramètres d'amplitude et d'argument (phase/fréquence) d'une onde sinusoïdale appelée porteuse. Le dispositif qui effectue cette modulation, en général électronique, est un modulateur (voir modem). L'opération inverse permettant d'extraire le signal de la porteuse est la démodulation.

III.2.1.2) Principe de la modulation ASK :

Le modem le plus simple, que l'on rencontre encore aujourd'hui dans nombre de modules de liaison radio aux fréquences autorisées de 433 MHz et 868 MHz, utilisé par exemple pour télécommander des alarmes, des portails ou des portes de garage électrique ou bien encore des volets roulants, est le modem ASK. Cette appellation signifie Amplitude Shift Keying, c'est-à-dire modulation par variation d'amplitude.

Comme le montre la figure ci-dessous, son principe est fort simple puisqu'il consiste à représenter un niveau zéro logique par une absence de signal et un niveau un logique par un signal sinusoïdal à une fréquence qui dépend de la vocation du modem.

Généralement, ce signal est à fréquence relativement basse, de quelques kHz à quelques dizaines de kHz afin de pouvoir voyager sur de nombreux supports.



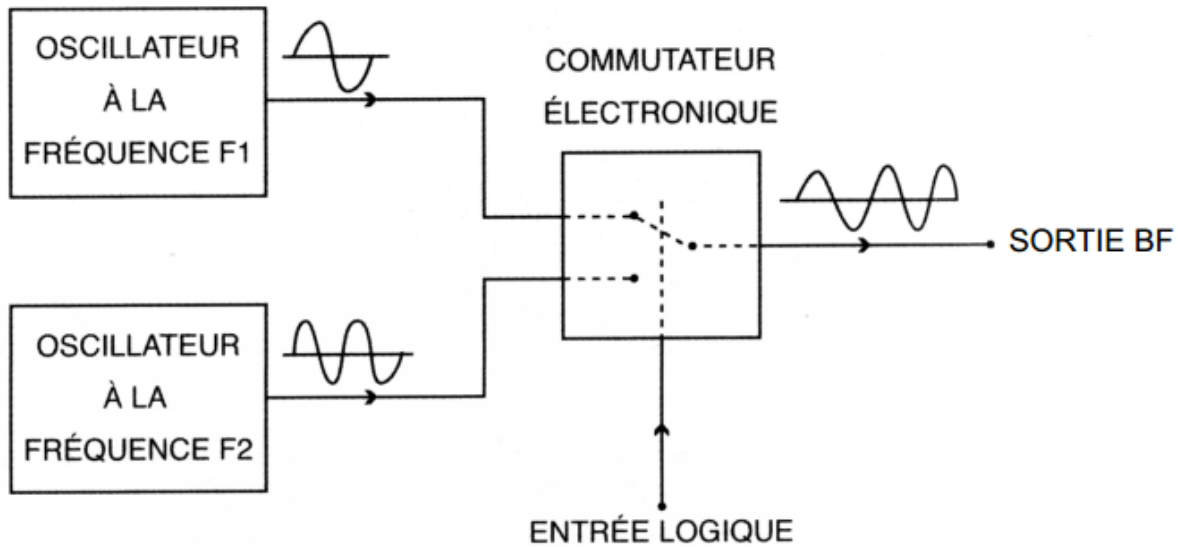
Principe de la modulation ASK

Figure 3 : modulation ASK

Le modem ASK présente cependant quelques inconvénients liés au bruit. Le bruit peut se superposer au signal transmis et peut donc perturber la réception des niveaux zéros logiques, risquant alors d'être confondus avec des niveaux un si ce dernier atteint une amplitude trop importante.

III.2.1.3) Principe de la modulation FSK :

Pour des transmissions plus fiables, on préfère généralement le modem FSK qui signifie Frequency Shift Keying ou encore modulation par variation de fréquence. Comme le montre la figure ci-dessous, un tel modem traduit un zéro logique par un signal à une fréquence $F1$ et un un logique par un signal à une fréquence $F2$. Il est alors beaucoup plus difficile de perturber le signal émis par un tel modem avec des bruits car, contrairement au modem ASK, il n'existe plus de phases de silences pendant la transmission qui se fait à niveau constant.



Principe de la modulation FSK

Figure 4 : modulation FSK

Ces modems se rencontrent aujourd'hui sur des modules similaires à ceux évoqués précédemment pour la modulation ASK, mais lorsque l'on souhaite une télécommande d'une grande fiabilité, ils sont vendus généralement à un prix un peu plus élevé même si, techniquement parlant, cela ne se justifie pas vraiment... Pour efficaces qu'elles soient, ces deux techniques présentent cependant l'inconvénient d'être limitées en termes de vitesse maximum de transmission des données ce qui est assez facile à comprendre.

Considérons par exemple le modèle FSK. Si l'on souhaite que le démodulateur, c'est-à-dire la partie du modem qui traduit les signaux analogiques en signaux logiques, puisse fonctionner correctement, il faut qu'il puisse mesurer la fréquence de chacun des deux signaux transmis, pour chaque niveau logique, zéro ou un et donc qu'il puisse disposer d'au moins une période complète du signal analogique. La durée d'un bit ne peut donc en aucun cas être inférieure à la durée d'une période du signal modulant. En pratique, pour assurer un fonctionnement correct, on préfère prendre un facteur deux et disposer ainsi de deux périodes du signal analogique par bit du signal numérique. Ainsi, un signal basse fréquence de 1 200 Hz par exemple, ne peut pas véhiculer d'information numérique plus rapide que 600 bits par seconde environ. Pour aller vite, il faut donc nécessairement augmenter la fréquence des signaux analogiques transmis ce qui pose rapidement un problème car, plus leur fréquence est élevée, plus leur transmission est difficile pour diverses raisons telles que, par exemple, l'influence des capacités parasites en liaison filaire ou l'augmentation de la fréquence porteuse nécessaire en liaison radio.

III.3) LE RÉSEAU ZIGBEE :

D'après l'exemple de la figure ci-dessous, il existe trois types de périphériques dans le réseau ZigBee : le coordinateur, le routeur et les « End-Devices ».

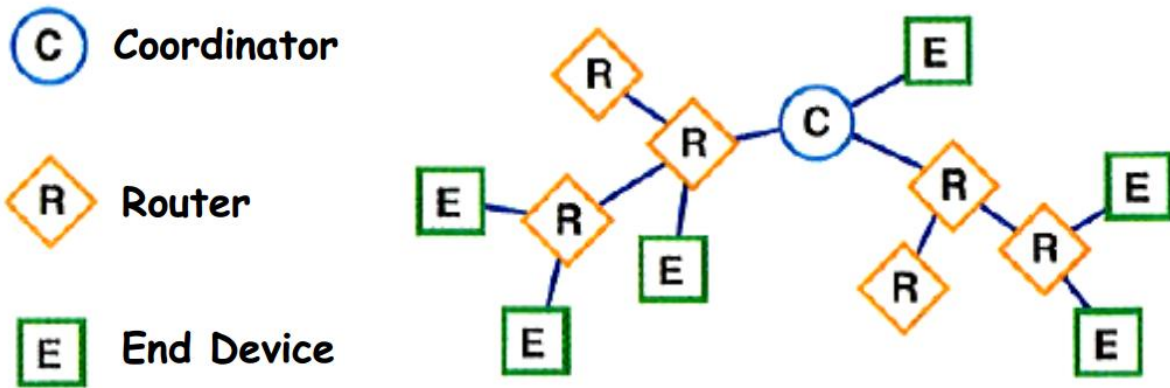


Figure 5 : élément essentiel d'un réseau zigbee

III.3.1) Fonction « SLEEP » :

Seuls les « End Devices » peuvent se mettre en sommeil dans le réseau ZigBee.

III.3.2) Adressage :

L'adressage se fait sur deux couches.

MAC sur 64 bits et adresse « réseau » sur 16 bits.

III.3.3) Sécurité :

Le cryptage utilisé est AES (Advanced Encryption Standard). Le réseau peut également être verrouillé pour empêcher d'autres périphériques de s'y connecter.

III.3.4) Communications séries :

Le module Xbee dispose d'un port série asynchrone. À l'aide de ce port, le module peut communiquer avec n'importe quel système disposant d'un UART compatible avec ses niveaux de tension (3,3 V).

Si vous souhaitez connecter un module Xbee à un ordinateur, un convertisseur de signaux RS232 ou USB doit être utilisé.

Les données entrant dans le module Xbee doivent prendre la forme d'un signal série asynchrone (le signal, au repos est au niveau haut lorsqu'aucune donnée n'est transmise).

Chaque octet de donnée est formée par un bit de départ, huit bits de données (LSB en premier) et un bit de stop. L'UART assurera toutes les tâches comme le timing et le contrôle de la parité.

Par défaut le module Xbee fonctionne en mode transparent.

III.3.5) Adressage des modules :

Il est souvent nécessaire de configurer les modules Xbee. Pour cela, il convient de passer en mode « commande », de leur envoyer des ordres de configuration, puis de procéder à l'écriture de cette suite de paramètres dans leur mémoire.

Il existe deux moyens pour effectuer cette opération : soit utiliser un logiciel émulateur de terminal tel l'HyperTerminal de Windows, soit utiliser le logiciel dédié appelé X-CTU.

Ce dernier est fourni gratuitement par le fabricant des modules, il permet de lire et de configurer tous les modèles de Xbee.

Chapitre I

Chacun des paquets de données envoyés par RF contient une adresse source et une adresse de destination dans son en-tête. Le module Xbee se conforme à la spécification 802.15.4 et supporte aussi bien l'adressage court sur 16 bits que l'adressage long sur 64 bits.

Une adresse unique est assignée à chaque module lors de la fabrication et peut être lue aux moyens des commandes SL (Serial number Low) et SH (Serial number High).

Un module utilisera son adresse unique sur 64 bits si la valeur de son adresse source sur 16 bits est configurée à 0xFFFF ou 0xFFFE.

Pour envoyer un paquet de données à un module, en utilisant son adresse sur 64 bits, il suffit de configurer l'adresse de destination du module émetteur (DL + DH, adresse basse + adresse haute) avec l'adresse source du module récepteur (SL + SH).

Pour envoyer un paquet de données à un module en utilisant cette fois un adressage court sur 16 bits, il convient de paramétrer l'adresse de destination (DL, adresse basse) du module émetteur avec la valeur du paramètre MY (sur X-CTU) du destinataire et de configurer l'adresse haute (DH) à 0.

Le mode unicast est le mode dans lequel le module Xbee opère par défaut. C'est le seul mode où plusieurs tentatives d'envois peuvent avoir lieu. Lors de la réception d'un paquet de données, le module récepteur envoie un accusé de réception (Acknowledge) au module émetteur. Si le module émetteur ne reçoit pas cet accusé, il réitère cet envoi jusqu'à trois fois, ou jusqu'à ce qu'il reçoive l'accusé de réception.

Adresse courte sur 16 bits

Dans le mode unicast, les modules peuvent être configurés avec une adresse courte sur 16 bits ou MY sera inférieur à 0xFFFE.

En configurant le paramètre DH à 0, l'adressage se fera sur 16 bits.

Pour deux modules communicants, l'adresse de destination du module émetteur devra être égale au paramètre MY du module récepteur.

Adresse longue sur 64 bits

Lorsqu'un périphérique de fin « End device » est associé à un périphérique coordinateur, son paramètre MY est configuré à 0xFFFE afin de passer en adressage sur 64 bits.

L'adresse sur 64 bits du module est stockée comme paramètre SL et SH. Afin d'envoyer un paquet de données à un module choisi, l'adresse de destination (DL + DH) d'un des modules doit correspondre à l'adresse source de l'autre (SL + SH).

Le mode broadcast est le mode dans lequel chaque module Xbee accepte le paquet de données reçu qui contient une adresse de broadcast. Configurer dans ce mode, les modules récepteurs n'envoient pas d'accusé de réception et les modules émetteurs ne procèdent pas à une répétition des envois.

Pour envoyer un paquet de données à tous les modules, indépendamment d'un adressage 16 bits ou 64 bits, les adresses de destination de tous les modules devront être configurées de la manière suivante : DL = 0x0000FFFF et DH = 0x00000000.

IV) L'arduino :

Arduino, et son récent synonyme Genuino1, sont des cartes matériellement libres sur lesquelles se trouve un microcontrôleur (d'architecture Atmel AVR comme par exemple l'Atmega328p). Les schémas de ces cartes sont publiés en licence libre, cependant, certains composants, comme le microcontrôleur par exemple, ne sont pas en licence libre.

IV.1) Origine du nom :

L'Arduino emprunte son nom au « Bar di Re Arduino », un pub de la ville d'Ivrée, située en Italie du Nord³. Le nom propre Arduino est un cognat de Ardennes.

IV.2) Matériel :

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328, Atmega32u4 ou ATmega2560 pour les versions récentes, ATmega168, Atmega1280 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5 V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles).

Le microcontrôleur est préprogrammé avec un boot-loader de façon à ce qu'un programmeur dédié ne soit pas nécessaire.

Les modules sont programmés au travers d'une connexion série TTL, mais les connexions permettant cette programmation diffèrent selon les modèles. Les premiers Arduino possédaient un port série RS-232, puis l'USB est apparu sur les modèles Diecimila, tandis que certains modules destinés à une utilisation portable comme le Lillypad ou le Pro-mini se sont affranchis de l'interface de programmation, relocalisée sur un module USB-série dédié (sous forme de carte ou de câble), cela permettait aussi de réduire leur coût, le convertisseur USB-Série TTL (un FTDI232RL de FTDI) coûtant assez cher).

L'Arduino utilise la plupart des entrées/sorties du microcontrôleur pour l'interfaçage avec les autres circuits. Le modèle Diecimila par exemple, possède 14 entrées/sorties numériques, dont 6 peuvent produire des signaux PWM, et 6 entrées analogiques. Les connexions sont établies au travers de connecteurs femelles HE14 situés sur le dessus de la carte, les modules d'extension venant s'empiler sur l'Arduino. Plusieurs sortes d'extensions sont disponibles dans le commerce.

D'autres cartes comme l'Arduino Nano ou le Pro micro utilisent des connecteurs mâles, permettant de les disposer sur une platine d'expérimentation.

IV.3) Logiciel de programmation:

Le logiciel de programmation des modules Arduino est une application Java, libre et multi-plateforme, 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). Il est également possible de se passer de l'interface Arduino, et de compiler et uploader les programmes via l'interface en ligne de commande⁴.

Le langage de programmation utilisé est le C++, compilé avec avr-g++ 5, 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++.

IV.4) Matériel libre et logiciel libre :

Le design matériel de l'Arduino est distribué sous licence Creative Commons Attribution Share-Alike 2.5 et est disponible sur le site d'Arduino. Les schémas ainsi que les typons de circuits sont également disponibles. Le code source de l'environnement de programmation et les bibliothèques embarquées sont disponibles sous licence LGPL.

De nombreuses informations sont disponibles au sujet des circuits intégrés présents sur les cartes arduino (micro-contrôleurs Atmega par exemple) même si ils restent très fermés du fait des fabricants.

V) Asservissement visuel :

Pour avoir un système asservi dans l'asservissement classique on doit utiliser un capteur soit de vitesse ou de température, nous dans l'asservissement visuel on utilise un capteur visuel comme une caméra afin que l'opérateur récupère l'information visuelle et génère une commande pour positionner sans robot à une position bien définie.

V.1) Introduction :

Les techniques d'asservissement visuel consistent à utiliser les informations fournies par une ou plusieurs caméras afin de contrôler les mouvements d'un système robotique. Quelle que soit la configuration du capteur, pouvant aller d'une caméra embarquée sur l'effecteur du robot à plusieurs caméras déportées, il s'agit de sélectionner au mieux un ensemble de mesures, permettant de contrôler les degrés de liberté souhaités, et d'élaborer une loi de commande afin que ces mesures atteignent une valeur désirée ou suivent une trajectoire spécifiée. Le principe de la commande se ramène ensuite à réguler (c'est-à-dire amener et maintenir à zéro) le vecteur d'erreur entre mesure et consigne. Avec un capteur de vision, fournissant à la base des informations 2D, la nature des mesures potentielles est extrêmement riche, puisque l'on peut considérer en asservissement visuel aussi bien des mesures 2D, telles que les coordonnées de points caractéristiques dans l'image par exemple, que des mesures 3D, fournies par un module de localisation exploitant les informations 2D extraites. De cette richesse provient la difficulté majeure de l'asservissement visuel, à savoir, parmi l'ensemble des informations potentielles, comment sélectionner, construire et combiner celles qui fourniront un comportement satisfaisant au système. Les qualités souhaitées sont nombreuses : stabilité locale voire globale, robustesse aux erreurs de mesure.

V.2) Commande :

Sur les aspects proches de la commande, des développements importants ont été réalisés ces dernières années sur le couplage entre planification de trajectoires et asservissement visuel. Ces travaux sont intéressants puisqu'ils permettent d'appréhender de grands déplacements à réaliser entre la position initiale et la position désirée. Ils permettent aussi d'accroître sensiblement la robustesse du système vis-à-vis des erreurs de modélisation et de calibration par la poursuite de ces trajectoires par asservissement visuel.

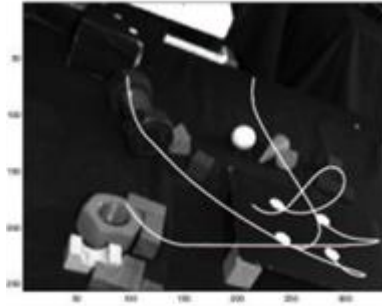


Figure 6 : planification de trajectoires



Figure 7 : Asservissement visuel à partir d'une base de données

VI) Conclusion :

Dans ce chapitre on a pu choisir le protocole de transmission le mieux pour notre projet car le zigbee et le plus utiliser pour transmettre une commande à distance car il consomme moins d'énergie et il transmet pour une longue distance. Et par conséquent la carte de commande qui est choisi et la carte arduino pour sa simplicité et sa disponibilité dans le marcher, et pour minimiser les erreurs entre la consigne et la sortie j'ai proposé l'asservissement visuel comme solution.

Dans le prochain chapitre on va entamer la phase de conception d'un robot rigide a deux axes télé guider ont parlent de ses essentiel partie.



Chapitre 2

CONCEPTION DU ROBOT RIGIDE

I) Introduction :

Le développement majeur dans la technologie informatique nous a permis la fabrication d'un robot rigide à deux axes à faible coût et sa grâce à la disponibilité des matérielles sources, tels que les microcontrôleurs Arduino et les cartes de puissance. Donc le temps de prototypage et de développement sont considérablement réduits. Du point de vue l'utilisant d'un système de commande basé sur Arduino facilite le développement d'un prototype d'un robot rigide. Et aussi dans ce chapitre je vais parler des capteurs qui sont essentiels en automatique et moi le capteur le plus important et le capteur visuel donc c'est la caméra pour cela je vais donner une idée générale sur les caméras et nous la plus importante est la caméra sans fil.

II) Partie mécanique :

Cette partie est pour but de dimensionner les différents éléments fonctionnels de notre dispositif pour assurer le bon fonctionnement de ce dernier. Et cette partie contient le système de guidage et les axes de déplacement.

II.1) System de guidage :

Pour fabriquer le système de guidage il faut prendre en compte qu'il contient deux parties essentielles la première consiste à chercher les douilles à billes et la deuxième partie consiste à concevoir un module de guidage parallèle à base de deux tiges lisses comme le montre la figure 10.

II.1.1) Douilles à billes :

Il s'agit de roulements à billes conçus pour se déplacer le long d'une tige lisse. C'est-à-dire qu'ils ne tournent pas mais se déplacent linéairement (guidage en translation).

Permettent des fonctionnements sans jeux, améliorent la précision et les performances. Valeur du coefficient de frottement de 0.001 à 0.005.

Utilisées sur les machines-outils, robots, systèmes automatisés...

Vitesse de déplacement 5m/s.

Economique pour les arbres lisses, ne supportent que des charges radiales.



Figure 8 : Douilles à billes

Chapitre II



Figure 9: Douilles à billes avec tige

Par contre, on associe toujours cette douille avec une butée angulaire qui peut être réalisée par une autre douille sur une colonne parallèle.

C'est l'ensemble du montage qui les empêche de tourner sur l'axe.



Figure 10 : Module de guidage parallèle

Rail de guidage pour guidage prismatique :

Les guidages de type prismatique associent des surfaces de contact planes. Guidages linéaires sur patins utilisés dans la mécanique de précision (automation, dispositifs de contrôle et de mesure...)

Ils permettent une absence totale de jeux et ils possèdent un très faible coefficient de frottement (0.0005 à 0.003).

Vitesse de déplacement de 3 à 5m/s

Les systèmes linéaires sont généralement équipés de 2 ou 3 pistes de roulements assurant la mise en position et de pistes supplémentaires généralement réglables assurant le maintien en position.

Une recirculation des billes est souvent prévue.

II.2) Déplacement suivant les axes x y :

Pour concevoir un système de déplacement on aura besoin des courroies et des poulies.

II.2.1) Courroie :

Les courroies et les poulies permettent la transmission d'un mouvement de rotation d'un arbre menant à un arbre mené relativement éloignés l'un de l'autre. Le glissement qui résulterait du fonctionnement d'une transmission par friction exclut les courroies plates



Figure 11 : Courroie



Figure 12 : courroie avec poulie

III) Partie électrique :

Dans cette partie on va parler de tous les éléments électrique et électronique nécessaire pour la réalisation d'une commande à distance d'un robot rigide.

III.1) Les actionneurs :

Un actionneur est une chose qui transforme l'énergie qui lui est fournie en un phénomène physique qui fournit un travail, modifie le comportement ou l'état d'un système. Dans notre cas l'actionneur est un moteur pas à pas.

III.1.1) Moteur pas à pas :

Les principaux types de moteur pas-à-pas

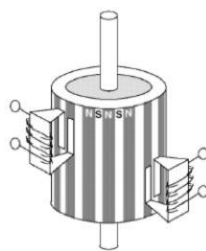


Figure 13 : Aimant permanent

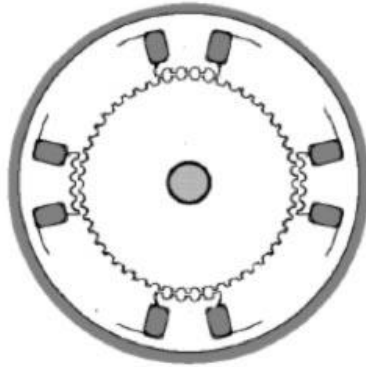


Figure 14 : Hybride

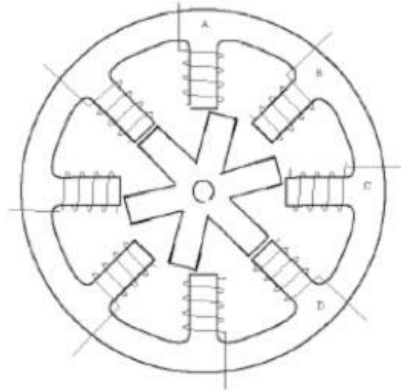


Figure 15 : Reluctance Variable



1) À aimant permanent «tin can»

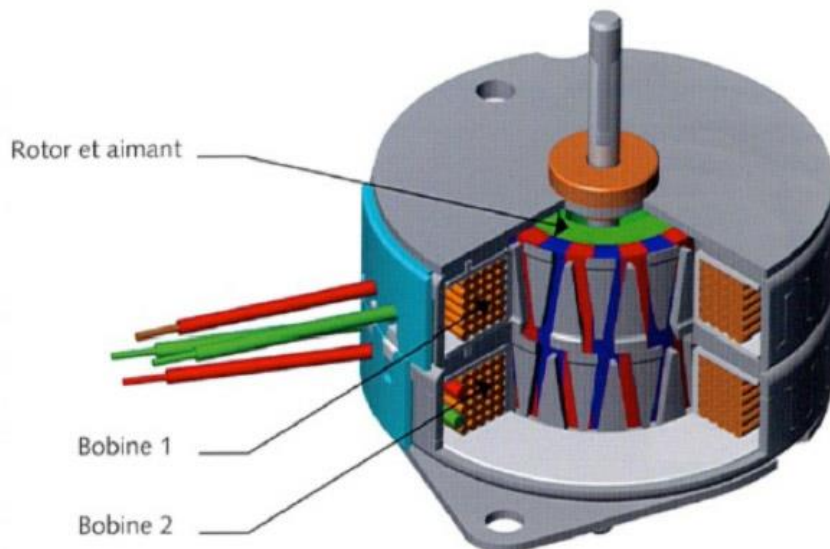


Figure 16 : Vue d'un moteur à aimants permanents

Le nom de ce type de moteur pas à pas est lié à la conception de son stator : une **tôle magnétique découpée et emboutie**.

Sur un diamètre intérieur, les tôles composent une **série de dents qui symbolise les pôles du stator** tout en laissant un espace torique pour une bobine.

Chaque sous-ensemble représente une phase stator (cf. schéma).

Chapitre II

Le rotor est un barreau aimanté radialement ayant plusieurs paires de pôles N-S.

Avantages du moteur à aimant permanent :

- Bon marché
- Dimensions réduites
- Bon rendement
- Bon amortissement des oscillations
- Grand angle de pas (nombre de pas faible : 48)

Inconvénients du moteur à aimant permanent :

- Puissance faible
- Paliers en bronze ou plastique (pas de roulement)
- Couple résiduel sans courant
- Vitesse faible



2) Hybride

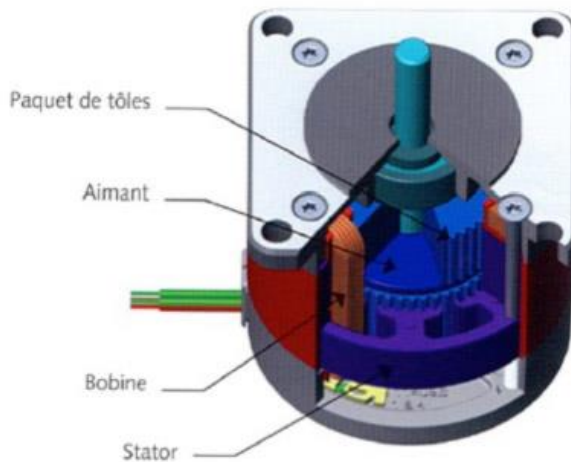


Figure 17 : Vue d'un moteur hybride

Le moteur pas à pas « **hybride** » allie le principe du moteur à réluctance variable à celui du moteur à aimant permanent.

Le rotor du moteur hybride comprend 2 structures régulières de dents.

Ces 2 blocs sont décalés d'une $\frac{1}{2}$ dent l'un par rapport à l'autre et sont fixés de part et d'autre d'un aimant permanent magnétisé axialement.

Le circuit magnétique du stator possède plusieurs pôles constitués de paquets de tôles entourés chacun d'une bobine ; les paquets de tôles se terminant par des dents.

Chapitre II

Une phase est constituée de plusieurs dents ; 4 dans la plupart des cas. Tous les pôles de la phase sont décalés de façon à assurer le déphasage de 90° (quadrature).

Avantages du moteur pas à pas hybride :

- Couple important
- Plus de puissance
- Rendement assez bon
- Courbe start/stop assez élevée
- Bon amortissement
- Adapté au fonctionnement micropas
- Roulement à billes pour une meilleure charge radiale et plus longue durée de vie
- Petit angle de pas

Inconvénients du moteur pas à pas hybride :

- Inertie élevée
- Couple résiduel sans courant
- Plus couteux
- Plus volumineux



3) Réductance variable

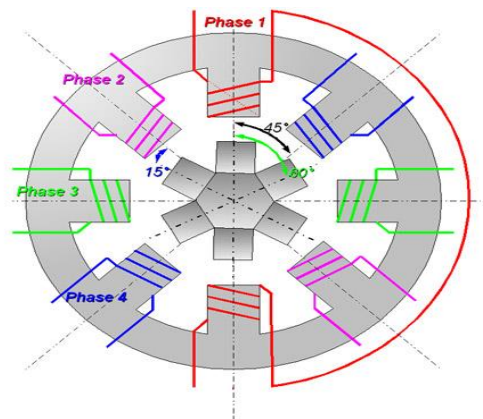


Figure 18 : Vue d'un moteur à reluctance variable

Le principe de fonctionnement de ce type de moteur pas à pas est proche de celui du moteur hybride, avec une structure dentée au rotor et au stator.

Il n'y a pas d'aimant au rotor pour renforcer l'action du flux et donc pas de couple résiduel sans courant.

Ce type de moteur pas à pas n'est presque plus utilisé ni fabriqué.

□ Comment fonctionne un moteur pas à pas ?

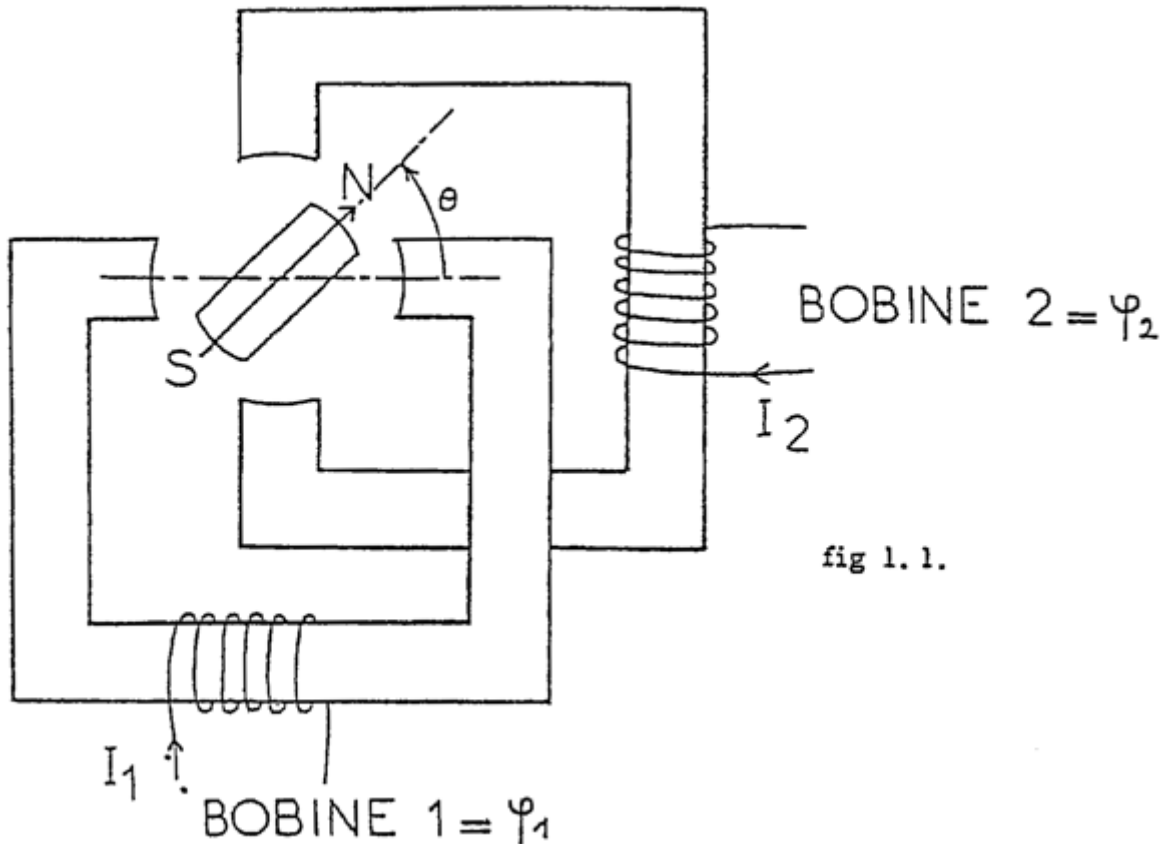


Figure 19: Structure d'un moteur pas à pas (montage bipolaire)

La figure ci-dessous montre la structure de base d'un moteur pas à pas à aimant permanent. Le rotor est un barreau aimanté radialement. Dans ce cas simple, l'aimant est bipolaire (un pôle nord et un pôle sud). Le stator présente 2 phases (bobine 1 et bobine 2).

Si on alimente la bobine 1, dans le sens $+I$, l'aimant va se placer en position

Si l'on supprime le courant dans la bobine 1 et qu'on l'établit dans la bobine 2 dans le sens $-I$, l'aimant va tourner de 90° et va se placer en position

Si l'on supprime le courant dans la bobine 2 et qu'on l'établit dans la bobine 1 dans le sens $-I$, l'aimant va tourner de 90° et va se placer en position

On rétablit enfin le courant dans la bobine 2, dans le sens $+I$ l'aimant va se placer en position

L'aimant a donc 4 positions possibles par tour. On dit que l'angle de pas, ou le pas est de 90° . On a donc un moteur pas à pas de 4 pas par tour.

Cette séquence d'alimentation des phases alimente une seule phase à la fois (1 phase ON). Aussi, en alimentant de manière appropriée 2 phases à la fois (2 phases ON) on obtiendrait également 4 positions stables décalées de 90° entre elles mais à 45° par rapport au cycle 1 phase ON.

1 PHASE ON

Position	Bobine 1	Bobine 2	Angle
1	+1	0	0°
2	0	-1	90°
3	-1	0	180°
4	0	+1	270°

2 PHASE ON

Position	Bobine 1	Bobine 2	Angle
1	+1	+1	45°
2	-1	+1	135°
3	-1	-1	225°
4	+1	-1	315°

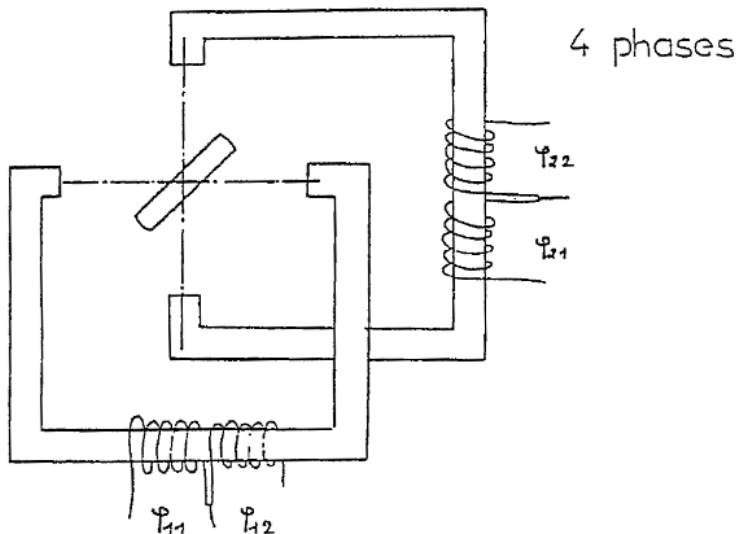


Figure 20 : Structure d'un moteur pas à pas (montage unipolaire)

Une commande mixte (1 phase ON, 2 phases ON, 1 phase ON, etc.) permet un fonctionnement en demi-pas, ce qui double le nombre de positions stables pour un tour et augmente ainsi la résolution du pas à 45° (8 pas par tour).

Il est aisé de voir que le sens de rotation du moteur dépend de la polarité du courant (la permutation des 2 fils d'une phase inverse donc le sens de rotation).

La solution la plus simple pour inverser le sens de rotation consiste à inverser le sens du courant. Ce type de montage est appelé Bipolaire (moteur 2 phases).

Chapitre II

Une autre solution consiste à utiliser un bobinage à point milieu avec le même sens de bobinage pour les 2 enroulements et d'alimenter le point milieu par un courant positif vers une extrémité puis vers l'autre. Ce type de montage est appelé Unipolaire (moteur 4 phases).

III.2) Les pilotes du moteur pas à pas :

Pour commander les moteur pas à pas on doit utiliser des pilotes car la carte Arduino ne peut pas les commander directement car elle délivre seulement 5v donc on a besoin des circuits de puissance, et dans notre cas on va utiliser le A4983 et le L298N.

III.2.1) Commande des moteurs pas à pas avec le pilot A4983 :

Le **A4983 Contrôleur de Moteur Pas à Pas** est une carte de dérivation facile à utiliser. Le conducteur dispose de courant réglable limitant et cinq résolutions micro-pas différents. Il fonctionne de 8 à 35 V et peut fournir jusqu'à 2 A . Les moteurs pas à pas offre un contrôle précis et une précision et sont souvent utilisés dans des projets de prototypage rapide, tels que les imprimantes 3D.

Ce produit est une carte de dérivation pour Allegro A4983 DMOS micro-pas pilote avec le traducteur, nous recommandons donc le datasheet avant d'utiliser ce produit. Ce pilote de moteur pas à pas vous permet de contrôler un moteur pas à pas bipolaire jusqu'à un maximum de sortie de 2A de courant. Voici certaines des principales caractéristiques du A4983 :

- Simple de contrôler de la direction d'interface
- Cinq résolutions différentes : le pas complet , demi-pas, quart de pas, huitième de pas, et seizième de pas
- vous permet de régler la sortie de courant maximum avec un potentiomètre qui vous permet d'utiliser des tensions supérieures à la tension nominale de votre moteur.
- Un Intelligent hacher qui sélectionne automatiquement le mode de désintégration correcte.

III.2.1.1) Matériel inclus :

La SFE A4983 pilote de moteur pas à pas carte de dérivation est livrée avec un 1 × 16-broches 2,54 mm en-tête masculine. Les en-têtes peut être soudés en place pour une utilisation avec des cartes sans soudure ou de 2,54 mm connecteurs femelles. Vous pouvez aussi souder vos fils moteur et autres connexions directement sur la plaque.

III.2.1.2) Connexions de Puissance :

Le A4983 est un Contrôleur de Moteur Pas à Pas nécessite une tension d'alimentation logique (3 - 5,5 V) pour être connecté et une tension d'alimentation pour le moteur de (8 - 35 V). Ces fournitures devraient disposer de condensateurs de découplage à proximité du module, et ils devraient être capables de fournir les courants attendus (des pointes allant jusqu'à 4 A pour l'alimentation du moteur).

III.2.1.3) Connexions du moteur :

Quatre, six, et huit-fils moteurs pas à pas peut être entraîné par l'A4983 si elles sont correctement connectées

Attention: Connexion ou déconnexion d'un moteur pas à pas pendant que le conducteur est alimenté peut détruire le conducteur. (Plus généralement, recâblage pendant qu'il est alimenté il va y'avoir des ennuis.)

Pas (et micro-pas) .

Pour activer la sélection de la résolution en cinq pas selon le tableau ci-dessous on doit envoyer des commandes sur les entrées de sélection (MS1, MS2, MS3). MS2 et MS3 ont une résistance interne de 100kΩ, mais MS1 n'a pas, donc il doit être connecté en externe.

MS1	MS2	MS3	Résolution	Micro-pas
Faible	Faible	Faible	Pas complet	
Haute	Faible	Faible	Demi-pas	
Faible	Haute	Faible	Quatrième-pas	
Haute	Haute	Faible	Huitième-pas	
Haute	Haute	Haute	Seizième-pas	

III.2.1.4) Entrées de commande :

Chaque impulsion d'entrée de STEP correspond à un micro-pas du moteur pas à pas dans la direction choisie par la broche DIR. Si vous voulez juste une rotation dans un seul sens vous pouvez attacher DIR directement à VCC ou GND. La puce dispose de trois entrées différentes pour contrôler ses états d'alimentation multiples: RST, SLP, et EN.

III.2.1.5) Structure interne du pilote A4983 :

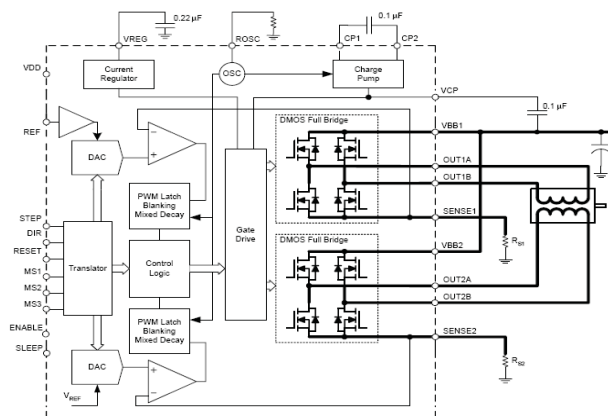


Figure 21 : structure du A4983

III.2.1.6) Un exemple d'un programme qui va commander un moteur pas à pas à l'aide d'un joystick :

```
#define dir 12 // déclaration des pins.

#define step 13 // déclaration des pins.

#define ms1 2 // déclaration des pins.

#define ms2 4 // déclaration des pins.

#define ms3 6 // déclaration des pins.

Int axe-x ; //déclaration de la variable axe-x.

Int joystick=A0 ; //déclaration de la variable joystick.

Void setup(){ // la fonction initial.

pinMode(dir,OUTPUT); // déclaration de la nature de pin (le pin dir est une sortie de
l'arduino).

pinMode(step,OUTPUT); // déclaration de la nature de pin (le pin step est une sortie de
l'arduino).

pinMode(ms1,OUTPUT); // déclaration de la nature de pin (le pin ms1 est une sortie de
l'arduino).

pinMode(ms2,OUTPUT); // déclaration de la nature de pin (le pin ms2 est une sortie de
l'arduino).

pinMode(ms3,OUTPUT); // déclaration de la nature de pin (le pin ms3 est une sortie de
l'arduino).

pinMode(joystick,INPUT) ; //déclaration de la nature de pin(le pin joystick est une entrée de
l'arduino) .}

void loop(){

digitalWrite(ms1,LOW); //on ecrit la valeur 0 sur le pin ms1.

digitalWrite(ms2,LOW); //on ecrit la valeur 0 sur le pin ms2.

digitalWrite(ms3,LOW); //on ecrit la valeur 0 sur le pin ms3.

axe-x= analogRead(joystick) ; // faire la lecture du joystick.

if(axe-x<500){ // donner un condition.

digitalWrite(dir,LOW);
```

```
digitalWrite(step,HIGH);  
  
delay(1);  
  
digitalWrite(step,LOW);}  
  
if(axe-x>600){  
  
digitalWrite(dir,HIGH);  
  
digitalWrite(step,HIGH);  
  
delay(1);  
  
digitalWrite(step,LOW);}  
  
}
```

III.2.2) Commande des moteurs pas à pas avec L298N:

Ce breakout board est un Double Pont-H destiné au contrôle de moteur continu (H-Bridge Motor Driver). Il est basé sur le composant L298N qui est un double Pont-H conçu spécifiquement pour ce cas d'utilisation.

C'est un module extrêmement utile pour le contrôle des moteurs. Il peut contrôler deux moteurs à courant continu ou un moteur pas-à-pas 4 fils 2 phases. il est conçu pour supporter des tensions plus élevées, des courants importants tout en proposant une commande logique TTL (basse tension, courant faibles, idéal donc pour un microcontrôleur).

Il peut piloter des charges inductives comme des relais, solénoïdes, moteurs continus et moteurs pas-à-pas. Les deux types de moteurs peuvent être contrôlés aussi bien en vitesse (PWM) qu'en direction. Toutes les sorties en puissance sont déjà protégées par des diodes anti-retour.

Il s'agit d'un module prêt à l'emploi.

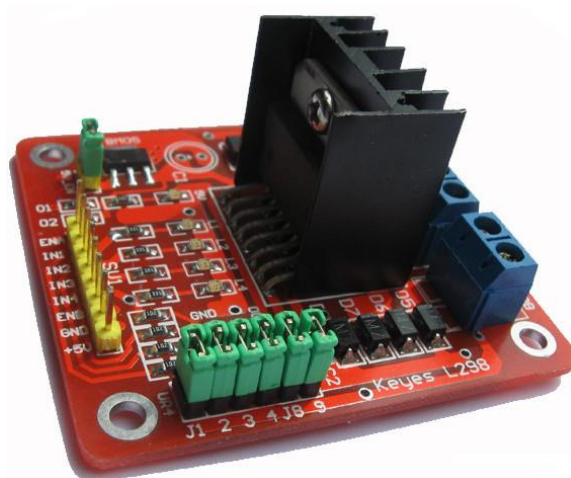


Figure 22 : le pilot L298N

III.2.2.1) Caractéristiques :

- Léger, petit
- Des capacités hors-pair pour contrôle moteur
- Diodes de protections
- Un dissipateur (pour dissiper la chaleur en cas de forte charge)
- Un sélecteur pour sélectionner la source d'alimentation
- 4 Sélecteurs pour les résistances pull up
- Sortie pour 2 moteurs continu/ 1 moteur pas-à-pas (4 bobines, deux phases)
- Indicateur LED pour sens de rotation moteur
- Indicateur LED pour alimentation 5V

4 trous de fixation standard

III.2.2.2) Spécifications :

- Composant de contrôle en puissance: L298N
- Alimentation de la charge: de +6V à +35V
- Courant Max (en pointe): 2A
- Tension de commande logique Vss: de +5 à +7V (alimentation interne de +5V)
- Courant de commande logique: de 0 à 36mA
- Tensions pour contrôle du sens: Low -0.3V~1.5V, high: 2.3V~Vss
- Tensions pour contrôle "Enable": Low -0.3V~1.5V, high: 2.3V~Vss
- Puissance Max: 25W (Température 75 °C)
- Température de fonctionnement: de -25°C à +130°C
- Dimensions: 60mm x 54mm
- Poids: ~48g

III.2.2.3) Idées d'applications :

- Pilotage de moteur continu (ex: voiture téléguidée, montage divers à base de moteurs)
- Pilotage de moteur pas-à-pas 4-fils deux-phase

III.2.2.4) Précautions :

- S'assurer que le pôle positif soit raccorder à VMS et le pôle négatif à GND
- La tension d'entrée (étage de puissance) ne doit pas excéder 35V

III.2.2.5) Détails techniques :

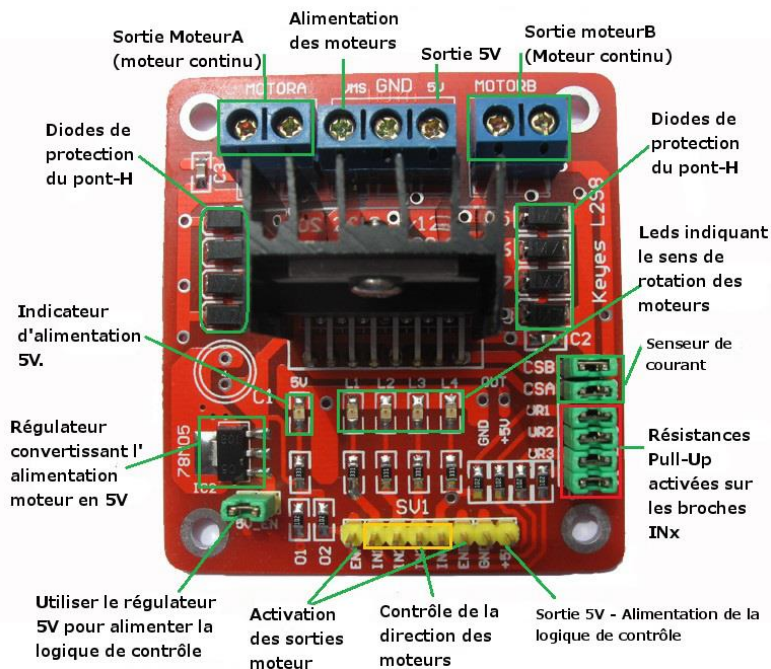


Figure 23 : détail technique du pilote L298N

Activation moteur :

- ENA raccordés à un niveau haut (HIGH) activera MOTOR A.
- ENB raccordés à un niveau haut (HIGH) activera MOTOR B.
- Si vous voulez contrôler la vitesse, vous pouvez connecter ENA(ENB) sur une sortie PWM.

Rotation Moteur A :

- IN1 raccordés à 5V et IN2 a GND MOTOR A tournera dans le sens de l'horloge.
- IN1 raccordés à GND et IN2 à 5V MOTOR A tournera dans le sens Anti-horloge.

Rotation Moteur B :

- IN3 raccordés à 5V et IN4 a GND MOTOR B tournera dans le sens de l'horloge.
- IN3 raccordés à GND et IN4 à 5V MOTOR B tournera dans le sens Anti-horloge.

III.2.2.6) raccordement sur une carte de contrôle :

Cette carte est compatible TTL, ce qui signifie qu'elle fonctionne avec de nombreuses cartes de commande mais aussi avec un Arduino.

Moteur Pas à pas:

Il est aussi possible de commander un moteur Pas-à-pas 4 fils de la même façon que 2 moteurs CC en utilisant les signaux ENA,IN1,IN2 and ENB,IN3,IN4 (voir exemple ci-dessous).

Exemple :

Contrôler un moteur Pas-à-pas (deux phases, 4-fils)

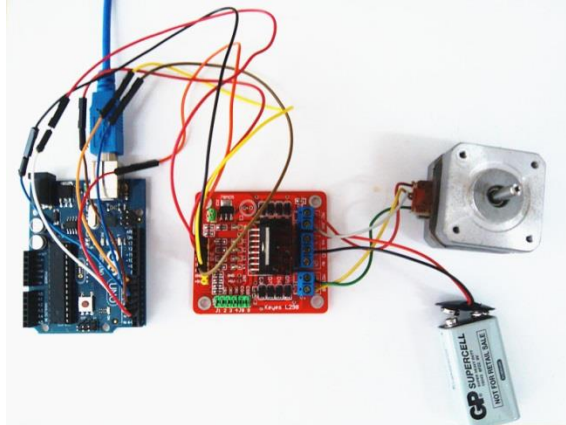


Figure 24 : contrôle d'un moteur pas à pas

Pour réaliser cette démonstration vous aurez besoin:

- Matériel: Un moteur pas-à-pas 4-fils, un Arduino, un breakoutboard L298, une source d'alimentation externe.
- Logiciel: Arduino IDE et le programme ci-dessous.

Premièrement, vous aurez besoin d'un multimètre pour tester les 4 fils et détecter les groupes (bobines). Les deux fils qui sont connectés ensemble (qui présente une résistance) forment un groupe.

Dans cet exemple, le fil rouge et gris forment un groupe (appelé Groupe A). Le fil jaune et vert forment l'autre groupe (appelé Groupe B).

Connectez l'Arduino et le L298 comme présenté sur l'image et documenté dans les commentaires du code ci-dessous. Ensuite, chargez le sketch sur votre Arduino.

III.2.2.7) Fonctionnement d'un moteur Pas-à-pas :

Un moteur Pas-à-pas se contrôle à l'aide de 4 battements (en 4 temps, en 4 rythmes). Le groupe A est connecté à la partie A, le groupe B est connecté à la partie B.

Supposons que:

- A représente le courant circulant dans le groupe A (sens direct).
- A- représente le courant inverse circulant dans le groupe A (lorsque l'on inverse la polarité)
- B représente le courant circulant dans le groupe B (sens direct).
- B- représente le courant inverse circulant dans le groupe B (lorsque l'on inverse la polarité)

Chapitre II

Pour faire tourner le moteur pas à pas, les séquences suivantes sont utilisées :

AB A-B A-B- AB-

Ou

AB AB- A-B- A-B

Le programme qui commande un moteur pas à pas :

```
int ENA=2; //Connecté sur votre Arduino, Pin 2
int IN1=3; //Connecté sur votre Arduino, Pin 3
int IN2=4; //Connecté sur votre Arduino, Pin 4
int ENB=5; //Connecté sur votre Arduino, Pin 5
int IN3=6; //Connecté sur votre Arduino, Pin 6
int IN4=7; //Connecté sur votre Arduino, Pin 7

void setup() {
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
  pinMode(IN1,OUTPUT);
  pinMode(IN2,OUTPUT);
  pinMode(IN3,OUTPUT);
  pinMode(IN4,OUTPUT);
  digitalWrite(ENA,HIGH); // Activer moteur A
  digitalWrite(ENB,HIGH); // Activer moteur B
}

void loop(){
  /* Un moteur Pas-à-pas se contrôle à l'aide de 4 battements (4 rythmes)
  comme décrit ci-dessus.
  Pour faire tourner le moteur pas-à-pas nous pouvons utiliser l'une des
  séquences suivantes:
      AB    A-B    A-B-    AB-
  ou
      AB    AB-    A-B-    A-B
  */
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,HIGH);
  digitalWrite(IN4,LOW);
  delay(10);
  digitalWrite(IN1,LOW);
  digitalWrite(IN2,HIGH);
  digitalWrite(IN3,LOW);
  digitalWrite(IN4,HIGH);
```

```
delay(10);
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,LOW);
digitalWrite(IN4,HIGH);
delay(10);
digitalWrite(IN1,HIGH);
digitalWrite(IN2,LOW);
digitalWrite(IN3,HIGH);
digitalWrite(IN4,LOW);
delay(10);
}
```

III.3) Les capteurs:

Les capteurs sont des composants d'automatisme qui ont pour but de récolter une information sur la partie opérative et de la retransmettre à la partie commande qui pourra ainsi la traiter. Les capteurs Tout ou Rien (TOR) délivrent une information binaire à la partie commande : l'information adopte l'état 0 ou l'état 1. Chaque état possède une signification dans le contexte du système. On distingue essentiellement les capteurs de type mécanique et ceux de proximité (cellules, inductifs ou capacitifs)

III.3.1) Constitution :

ils sont constitués :

- d'un élément sensible mécanique ou électrique
- d'un ou plusieurs contacts de type NF (normalement fermé = à ouverture) ou de type NO (normalement ouvert = à fermeture).

III.3.2) Caractéristiques :

- nature de la détection (mécanique optique, inductif, capacitif).
- distances de détection.
- tension d'alimentation (valeur, continu ou alternatif).
- type de raccordement (2 fils ou 4 fils, PNP ou NPN).
- nombre et types de contacts.
- noyable, non noyable (inductifs ou capacitifs).
- type de raccordement (vis, connecteur).
- taille et encombrement mécanique.

III.3.3) Symbolisation :

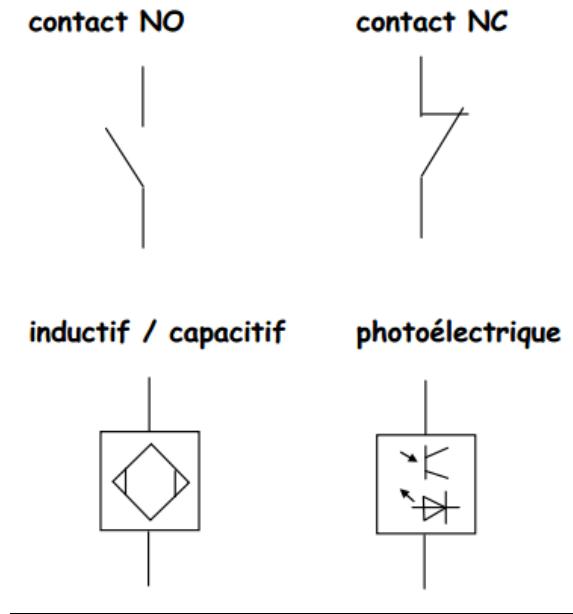


Figure 25 : symbolisation des capteurs

III.3.4) Les capteurs mécaniques : principe

L'objet à détecter touche physiquement l'élément mobile du capteur.

Le contact ouvre ou ferme le circuit d'information.

Un contact est dit "sec" s'il est libre de potentiel : le potentiel est donné par la partie opérative.

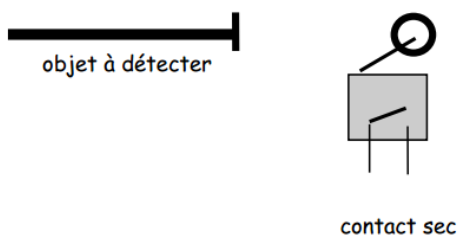


Figure 26 : capteurs mécaniques

III.3.4.1) Utilisation :

- la détection de pièces machines (cames, butées, pignons...).
- la détection de balancelles, chariots, wagons.
- la détection directe d'objets, etc.

III.3.4.2) Les avantages sont les suivants :

- sécurité de fonctionnement élevée : fiabilité des contacts et manœuvre positive d'ouverture.
- bonne fidélité sur les points d'enclenchement (jusqu' à 0,01 mm).
- séparation galvanique des circuits.
- bonne aptitude à commuter les courants faibles combinée à une grande endurance électrique.
- tension d'emploi élevée.
- mise en œuvre simple.
- grande résistance aux ambiances industrielles.

III.3.5) Les capteurs inductifs : principe

La détection de fait sans contact. Un circuit électronique à effet inductif transforme une perturbation magnétique due à la présence de l'objet en commande d'ouverture ou de fermeture statique (par transistor) du circuit d'information. La face sensible crée un champ magnétique local. Lorsque l'objet pénètre dans le champ magnétique, l'oscillateur se met en route et la sortie est activée.

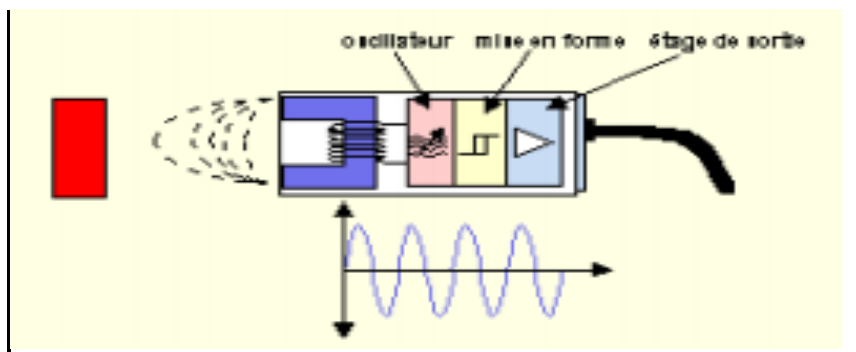


Figure 27 : capteur inductif



Figure 28 : capteurs inductifs

III.3.5.1) avantages et utilisation :

- pas de contact physique avec l'objet (pas d'usure), possibilité de détecter la présence d'objets fraîchement peints ou de surfaces fragiles.
- cadences de fonctionnement élevées en parfaite adéquation avec les modules ou les automatismes électroniques.
- grandes vitesses d'attaque pour la prise en compte d'informations de courte durée.
- produits entièrement enrobés dans une résine, pour une très bonne tenue aux environnements industriels agressifs.
- produits statiques (pas de pièces en mouvement) pour une durée de vie indépendante du nombre de cycles de manœuvres.
- visualisation de l'état de la sortie.

III.3.6) Les capteurs capacitifs : principe

La détection de fait sans contact. Un circuit électronique à effet capacitif transforme une perturbation électrique due à la présence de l'objet en commande d'ouverture ou de fermeture statique (par transistor) du circuit d'information. La face sensible crée un champ électrique local. Lorsque l'objet pénètre dans le champ électrique, l'oscillateur se met en route et la sortie est activée.



Figure 29 : capteurs capacitifs

III.3.6.1) Avantages :

- pas de contact physique avec l'objet (pas d'usure), possibilité de détecter la présence de tous types d'objets.
- cadences de fonctionnement élevées en parfaite adéquation avec les modules ou les automatismes électroniques.
- grandes vitesses d'attaque pour la prise en compte d'informations de courte durée.
- produits entièrement enrobés dans une résine, pour une très bonne tenue aux environnements industriels agressifs.
- produits statiques (pas de pièces en mouvement) pour une durée de vie indépendante du nombre de cycles de manœuvres.
- visualisation de l'état de la sortie.

III.3.7) Les cellules photoélectriques: la cellule en barrage

Les cellules en barrage sont composées d'un émetteur et d'un récepteur séparés. L'émetteur envoie le faisceau vers le récepteur. Le faisceau est coupé par l'objet à détecter. La distance de détection peut atteindre 30 m.

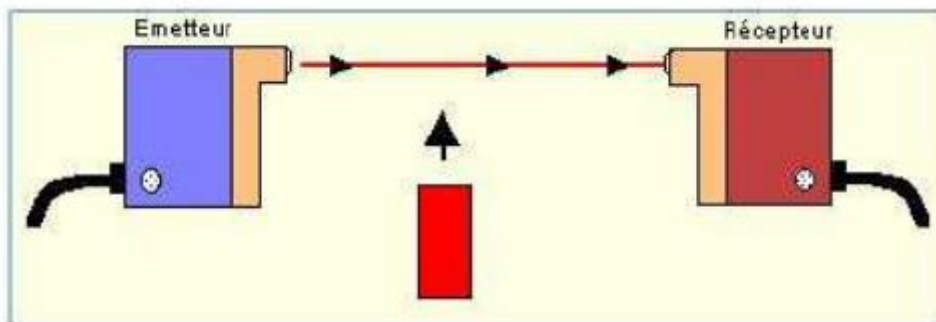


Figure 30 : cellules photoélectriques (cellule de barrage)

III.3.8) Les cellules photoélectriques: la cellule reflex

Les cellules reflex sont composées d'un émetteur/récepteur (dans le même boîtier) et d'un catadioptré (réflecteur).

L'émetteur envoie le faisceau qui revient vers le récepteur après s'être réfléchi sur le catadioptré. L'objet à détecter coupe le faisceau.

Si l'objet à détecter est réfléchissant, il convient d'utiliser un système reflex polarisé : le récepteur n'est pas sensible à la lumière renvoyée par l'objet. La distance de détection

est 2 à 3 fois inférieure au système en barrage.

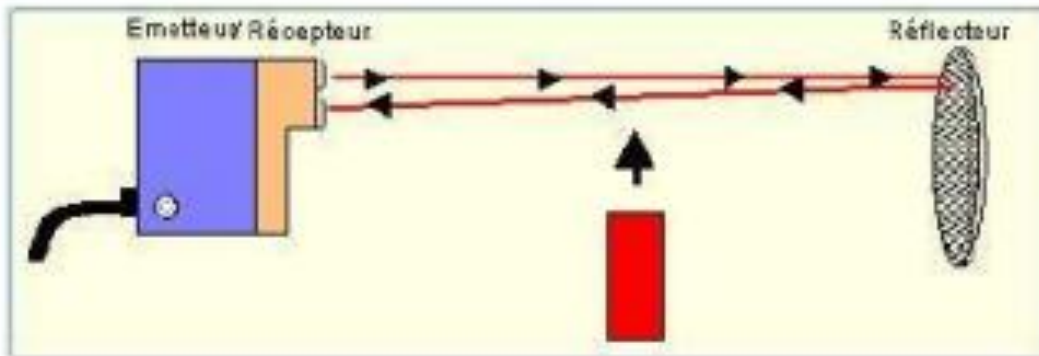


Figure 31 : cellules photoélectriques (cellule reflex)

III.3.8.1) Les cellules photoélectriques: la détection par proximité

Les cellules à détection par proximité sont dotées d'un émetteur qui envoie le faisceau. Celui-ci se réfléchit directement sur l'objet à détecter lui-même avant de retourner au récepteur.

L'objet doit être réfléchissant et guidé.

Variante : système avec effacement de l'arrière-plan : la détection est focalisée (par réglage avec un potentiomètre) en un point précis, évitant ainsi la détection de l'arrière-plan.

La distance de détection (assez faible) et son efficacité dépendent de la couleur et de la taille de l'objet à détecter.

III.3.8.2) Avantages :

- détection d'objets de toutes formes et de matériaux de toutes natures.
- sortie statique pour la rapidité de réponse ou sortie à relais pour la commutation de charges jusqu'à 2 A.
- généralement en lumière infrarouge invisible, indépendante des conditions d'environnement.

III.4) Module Xbee :

Le protocole de communication Zigbee est défini par la norme 802.15.4 de l'IEEE (Institut for Electrical and Electronics Engineers).

Les modules Xbee utilisées répondent à cette norme.

III.4.1) Présentation du matériel :

Les modules XBee sont des modules qui permettent d'envoyer et de recevoir des données, sans fil, performants et accessibles. Ils sont caractérisés par une portée très confortable d'une centaine de mètres en environnement d'intérieur, et jusqu'à plus d'un kilomètre en zone dégagée pour les modules XBee-PRO équipés d'une antenne adaptée. Ils peuvent être utilisés couplés à un microcontrôleur ou de façon indépendante. Ils sont très pratiques pour la réalisation de nombreux montages électroniques qui doivent pouvoir communiquer entre eux, dont des montages à l'utilité plus ou moins discutable.

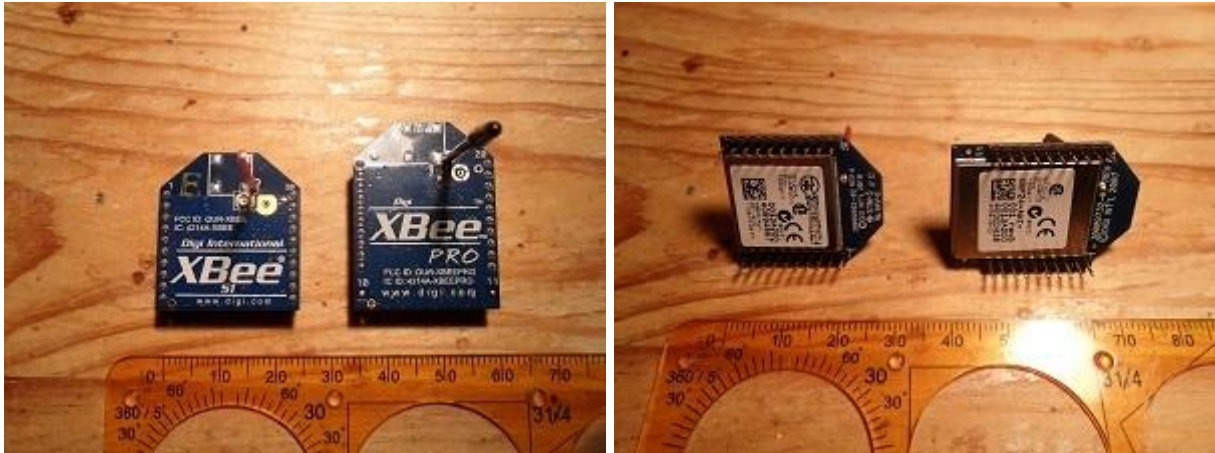


Figure 32 : module Xbee

Module XBee, à gauche sur les photographies, et module XBee-PRO, à droite sur les photographies

Ces modules XBee se basent sur la norme ZigBee pour mettre en place des communications radio. Cette norme est adaptée aux réseaux personnels sans fil à faibles débits.

Elle utilise :

- 16 canaux dans la bande de fréquences 2,4 GHz,
- 10 canaux dans la bande de fréquences 900 MHz,
- 1 canal dans la bande de fréquences 868 MHz.

Il n'est pas nécessaire de connaître en détail ce protocole de communication.

En effet, les composants électroniques présents dans les modules se chargent de la gestion de ce dernier.

Les modules XBee fonctionnent dans la bande de fréquences de 2,4GHZ.

Ils disposent, selon leur modèle, d'une antenne intégrée miniature, intégrée verticale, d'un emplacement pour souder une antenne externe, ou encore d'un connecteur pour raccorder une antenne externe.

La portée des modules dépend bien évidemment de l'antenne, mais aussi de la configuration de la puissance de sortie des modules.

Celle-ci peut être réglée de manière logicielle.

Le module XBee-PRO permet d'obtenir une puissance de sortie de 60 mW, correspondant à 18 dBm.

Chapitre II

Le tableau suivant reprend quelques caractéristiques des modules XBee et XBee-PRO.

Modèle	XBee	XBee-PRO
Portée en intérieur	Jusqu'à 30 mètres	Jusqu'à 100 mètres
Portée en extérieur	Jusqu'à 100 mètres	Jusqu'à 1500 mètres
Puissance de sortie	1 mW	60 mW
Possibilité de régler la puissance de sortie de façon logicielle	Non	Oui
Débit des données RF	250 000 bps	250 000 bps
Débit des données UART	De 1200 bps à 115 200 bps	De 1200 bps à 115 200 bps
Sensibilité	-92 dBm	-100 dBm
Tension d'alimentation	De 2,8 V à 3,4 V	De 2,8 V à 3,4 V
Courant consommé lors d'une émission	45 mA à 3,3 V	<ul style="list-style-type: none">• 10 dBm : 137 mA à 3,3 V• 12 dBm : 155 mA à 3,3 V• 14 dBm : 170 mA à 3,3 V• 16 dBm : 188 mA à 3,3 V• 18 dBm : 215 mA à 3,3 V
Courant consommé lors d'une attente ou d'une réception	50 mA à 3,3 V	50 mA à 3,3 V
Courant consommé en mode « SLEEP »	Moins de 10 μ A	Moins de 10 μ A

La consommation des modules XBee en attente ou en réception est importante. S'ils sont utilisés dans une application où la gestion de l'énergie est prépondérante, par exemple dans le cas d'un système dont l'alimentation électrique provient de piles ou de batteries, il sera judicieux de réduire le courant consommé par les modules.

C'est le rôle du mode « SLEEP », qui se décline en différentes variantes, selon la

Chapitre II

configuration qui a été appliquée aux modules.

La broche 9 des modules peut servir à passer dans le mode « SLEEP », si un niveau logique haut y est présenté.

La sortie du mode « SLEEP » se fait si un niveau logique bas y est présenté.

La consommation peut alors être réduite de 10 μA , et la sortie se fait après un délai de 13,2 ms, ou de 50 μA avec un temps réduit à 2 ms.

Le passage en mode « SLEEP » peut également se faire de façon régulière, et le module en sortira alors à intervalles réguliers, afin de vérifier si des données sont disponibles.

Les transmissions radio peuvent se faire suivant deux modes : le mode « Unicast », ou le mode « Broadcast ».

Dans le premier mode, lorsqu'un module envoie des données, chaque module récepteur concerné envoie un accusé de réception à l'émetteur. Si l'émetteur ne reçoit pas d'accusé de réception, il réitère son envoi, jusqu'à trois fois consécutives.

Dans le second mode, chaque module récepteur concerné reçoit les données, et il n'envoie pas d'accusé de réception en retour. L'envoi des données ne se fait qu'une seule fois.

Schema de principe de la transmission point a point :

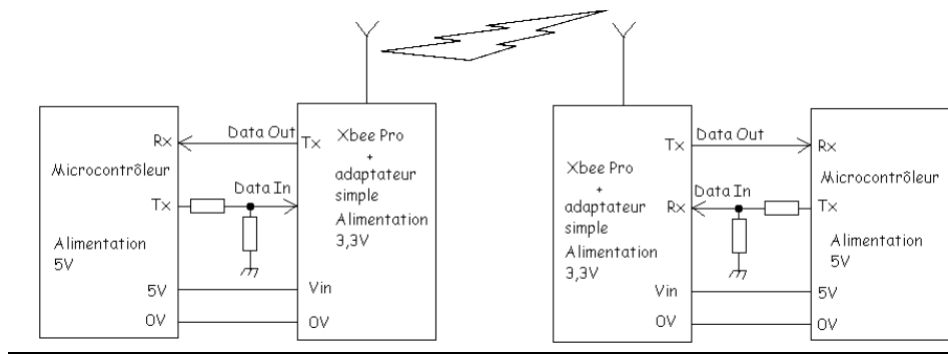


Figure 33 : transmission point à point

III.4.2) Programmation des modules Xbee :

1) Installation du logiciel de configuration X-CTU :

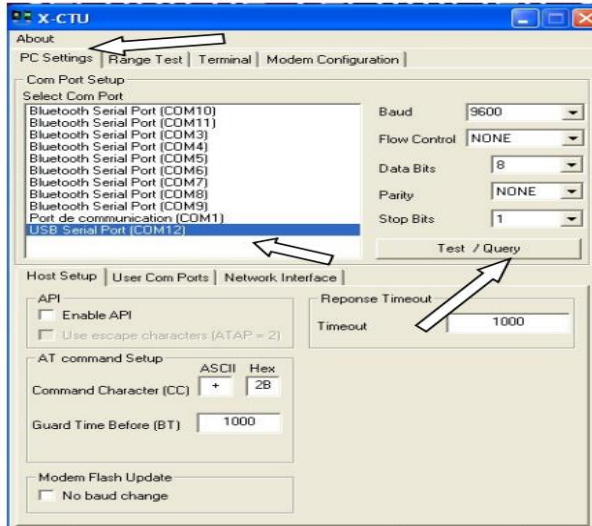
Pour pouvoir programmer les modules Xbee il faut télécharger le programme X-CTU de l'entreprise DIGI et de l'installer sur votre ordinateur.

2) lancement du logiciel :

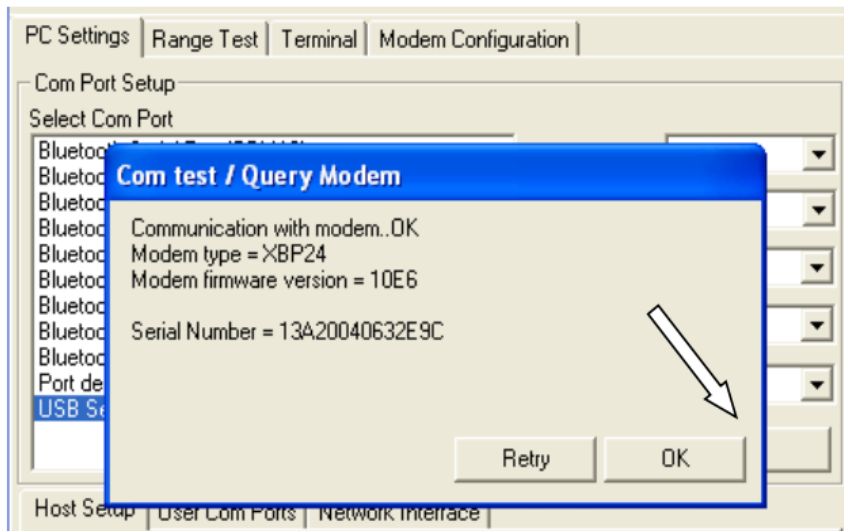


Créer un raccourci sur le bureau et lancez le.

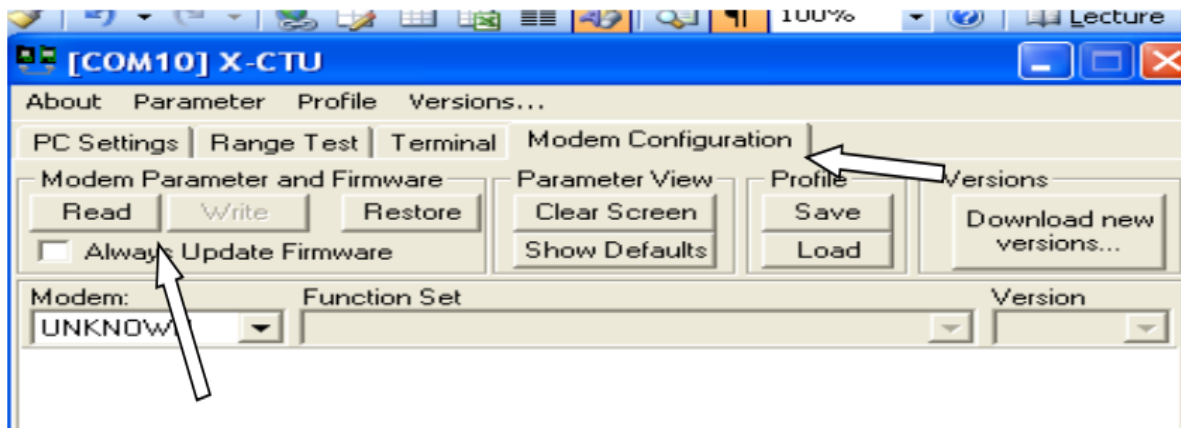
3) déclaration et test du port de communication avec le module :



Par défaut, le réglage des paramètres de communication série sont 9600bauds, 8 bits, pas de stop, pas de control de flux.

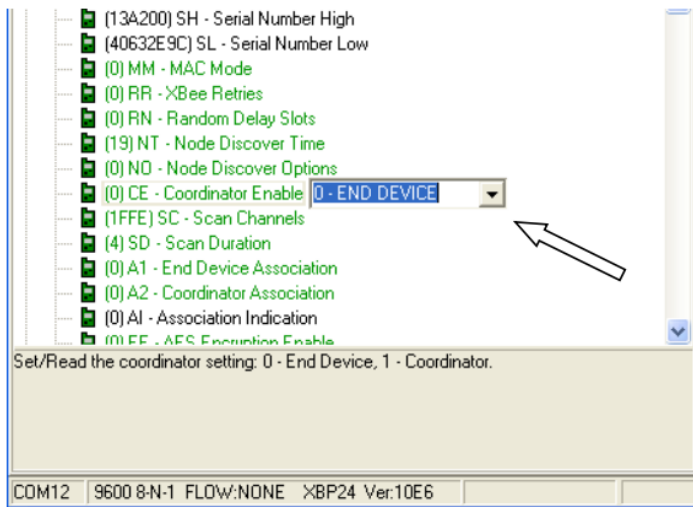


4) paramétrage du module Xbee :



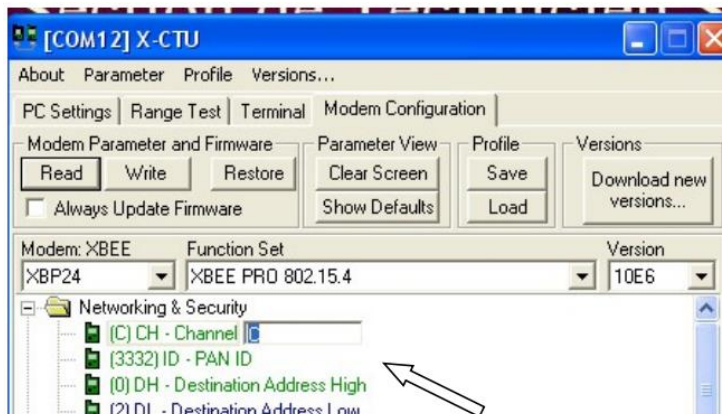
Chapitre II

4.1) Définition du type de module :



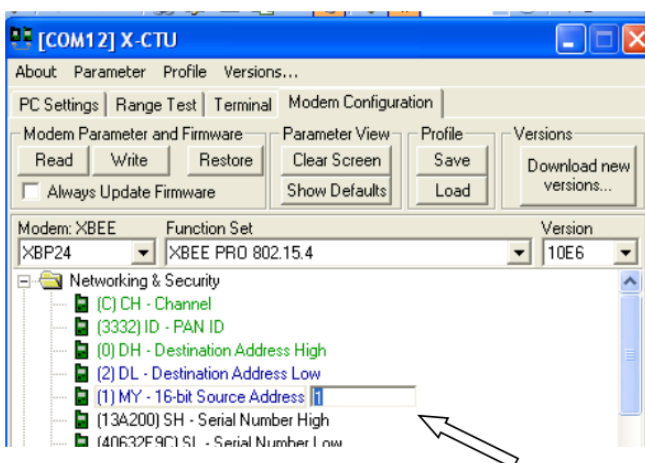
Le module est de type End Device se qui permet une communication partagée

4.2) Sélection du canal RF de communication :



Le module peut émettre en RF sur 10 canaux différents (Cf doc). Le canal doit être le même pour les 2 modules

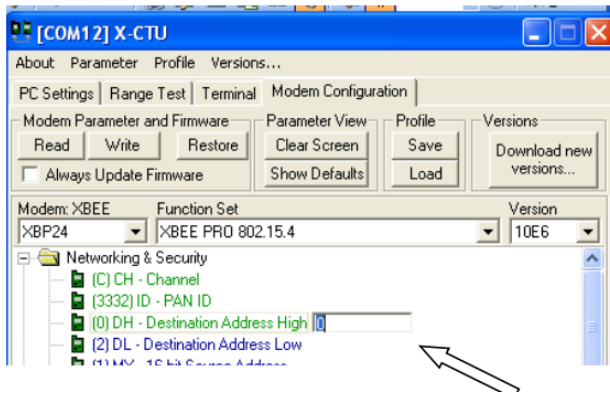
4.3) définition de l'adresse du module en cours de configuration :



L'adresse du module en cours de configuration est donc \$0001. On est en adresse courte (16 bits) car $MY \leq \$FFFE$

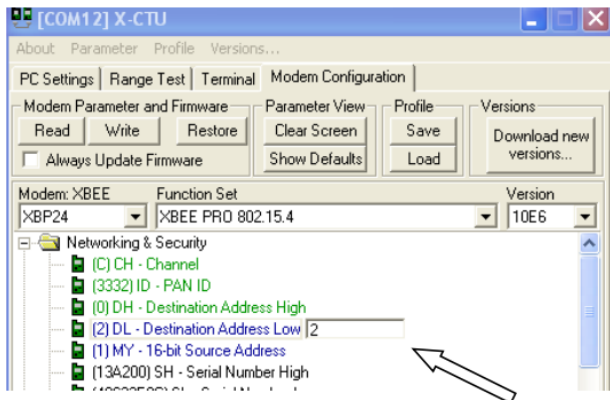
Chapitre II

4.4) définition de la partie haute du module de destination :



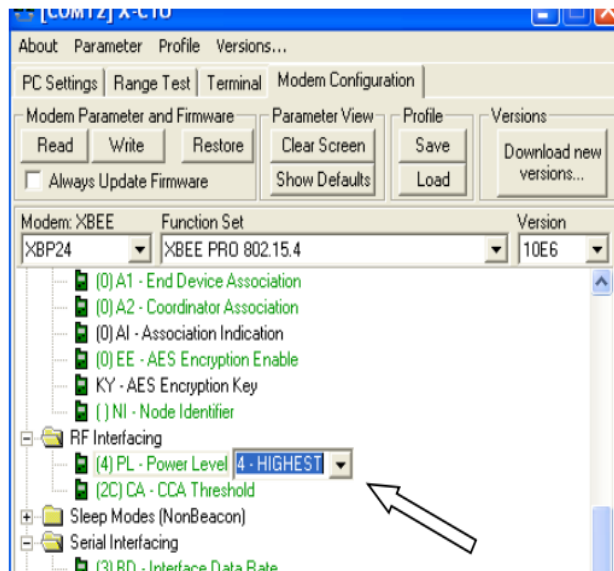
La partie haute de l'adresse du module de destination doit être '0' pour que le module en cours de configuration s'apparie avec le module de destination

4.5) définitions de la partie basse du module de destination :



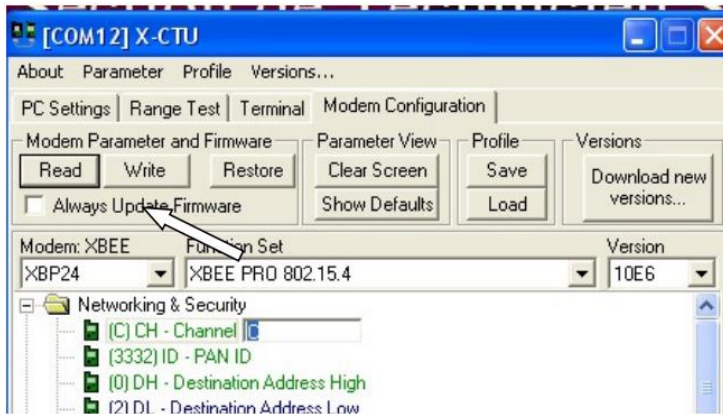
L'adresse du module de destination est donc \$0002

4.6) définition de la puissance d'émission :



La puissance d'émission du module peut être réglée (Elle influe sur la portée du module)

4.7) sauvegarde de la configuration :



Ne pas oublier d'écrire la nouvelle configuration dans le module Xbee.

Après avoir suivie tous les étapes les modules Xbee sont prêts à l'utilisation (transmettre des données de la carte arduino uno vers la carte arduino méga).

III.5) les caméras :

La caméra est un système de prise d'images animées qui génère un signal vidéo noir et blanc ou couleur. La caméra capte la lumière pour la transformer en signal électrique. Elle se compose parfois d'un capteur d'image et d'une électronique de traitement permettant de générer le signal vidéo.

III.5.1) Différents types de caméras :

- **Caméra CCD** :(Charge Coupled Device)

Le capteur CCD ne fait pas de distinction entre les couleurs. L'analyse des couleurs se fait via des filtres qui permettent de récupérer les signaux RVB avant traitement.

- **Caméra analogique :**

Ces caméras sont facilement reconnaissables ; elles ont une sortie de type BNC, la liaison se fait via le câble coaxial

- **Caméra infrarouge :**

Si l'on se réfère à la représentation de la courbe photopique concernant les longueurs d'onde, on s'aperçoit que l'infrarouge est une lumière invisible.

Les infrarouges ne se mesurent pas en lux.

Quand utilise-t-on des caméras infrarouges ? Lorsqu'il n'y a pas de lumière.

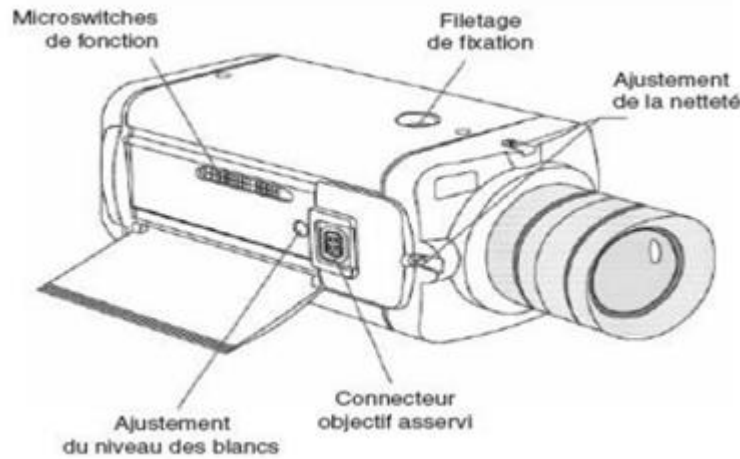


Figure 34 : camera analogique

•CaméraCMOS :

La fabrication des capteurs CCD requiert une fabrication spécifique engendrant des coûts importants. En revanche, la technologie CMOS est très utilisée dans la fabrication de composants électroniques, ce qui la rend économique pour la fabrication de caméras CMOS. Elles permettent une intégration à très grande échelle et consomment une quantité minimale d'énergie par rapport aux CCD ; une caméra peut fonctionner avec une batterie au NiCd pendant une semaine tandis que la caméra CCD ne fonctionnerait que quelques heures. La caméra CMOS génère un signal comportant du bruit nuisant à la fourniture d'une image de qualité. Cependant des progrès sensibles se font sentir et la caméra CMOS gagne inexorablement du terrain sur sa consœur la caméra CCD.

• Caméra discrète :

Les progrès des technologies font que de nos jours une caméra peut se loger dans n'importe quel accessoire. Les plus courants sont le détecteur d'intrusion, l'horloge et la tête de détection incendie (figure 5.3).



Figure 35: camera discrète

- **Caméra sans fil** : Une caméra sans fil peut être de deux types : caméra RF (radiofréquence) ou caméra HF (haute fréquence).

Le système est commercialisé sous forme de :

- kit (transmission, réception, connectique, alimentations) ;
- caméra seule et son alimentation adaptée (tension, puissance).

La caméra sans fil est utilisée principalement en informatique sous forme de webcam, et pour les fonctions de vidéosurveillance professionnelle et/ou domestique. On peut aussi l'utiliser en tant qu'astro cam, météo Cam, ou en modélisme (aéromodélisme et modélisme naval notamment).

La fonction d'une caméra sans fil est de transmettre à distance des prises de vues vidéo et photographiques, qui peuvent être aériennes - à l'aide d'un modèle réduit ou aérocam, ou embarquée dans un dispositif complexe, locomotive, véhicule radiocommandé (avion, voiture ou bateau etc..).

Par le biais d'une liaison sans fil, la caméra est reliée à un récepteur, lui-même connecté à un ordinateur personnel, téléviseur ou tout autre moniteur de visualisation. La liaison se fait par ondes radio grâce à la technologie sans fil, en utilisant la bande radio libre et gratuite ISM.



Figure 36: camera sans fil

III.6)Arduino:

Son des modules généralement construit autour d'un microcontrôleur Atmel. On a plusieurs types de carte Arduino et on va utiliser que la carte Arduino Uno et la Carte Arduino Méga 2560.

III.6.1)Arduino uno :

Description :

- Module microcontrôleur Arduino avec connexion USB
- Grand choix de « blindages » optionnels disponibles
- Conçue pour les roboticiens, artistes, concepteurs et amateurs
- Grand nombre de broches E/S : analogiques, numériques, MLI etc.

Chapitre II

- Basée sur l'ATmega328 (CI DIP amovible)
- Révision 3

L'Arduino Uno est une carte microcontrôleur basée sur l'ATmega328 Il est doté de 14 broches d'entrée/sortie numériques (dont 6 peuvent être utilisées comme sorties MLI), de 6 entrées analogiques, d'un oscillateur en cristal de 16 MHz, d'une connexion USB, d'une prise de courant, d'une embase ICSP et d'un bouton de réinitialisation. Il comprend tout ce qui est nécessaire pour prendre en charge le microcontrôleur ; connectez-le simplement à un ordinateur doté d'un Câble USB de 1.5m Type A vers B ou alimentez-le avec une Transformateur Mural SFE - 9VDC 650mA ou une Batterie LiPo 7.4V 2200mAh (Adaptateur Arduino) DFRobot pour qu'il soit opérationnel.

L'Arduino Uno diffère de toutes les cartes précédentes en ce qu'il n'utilise pas la puce contrôleur FTDI USB/série. Au lieu de cela, il dispose de l'Atmega16U2 programmé en tant que convertisseur USB/série. "Uno", signifie Un en italien et est ainsi nommé à l'occasion de la sortie prochaine de l'Arduino 1.0. Le Uno et la version 1.0 seront les versions de référence d'Arduino, pour continuer à progresser. Le Uno est le dernier d'une série de cartes Arduino USB, et le modèle de référence de plate-forme Arduino.

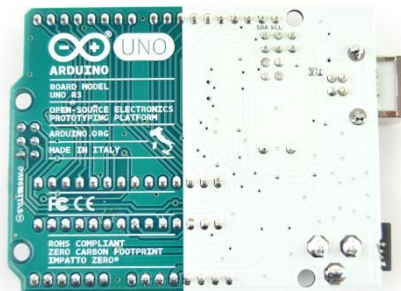


Figure 37: Carte arduino face arrière

L'Arduino Uno peut être alimenté par la connexion USB ou par une alimentation électrique externe. L'alimentation électrique est automatiquement sélectionnée. L'alimentation externe (non-USB) peut provenir d'un adaptateur CA/CC (mural) ou d'une batterie. L'adaptateur peut être connecté en branchant une prise de centre positif de 2,1 mm dans la prise de courant de la carte. Les broches de raccordement d'une batterie peuvent être insérées dans les embases mâles Gnd et Vin (tension en entrée) du connecteur POWER (alimentation). La carte peut fonctionner avec une alimentation externe de 6 à 20 volts.

L'Arduino Uno peut être programmé avec le logiciel Arduino. Sélectionnez « Arduino Uno » à partir du menu Outils > Menu de la carte (selon le microcontrôleur sur votre carte). Pour

Chapitre II

plus de détails, reportez-vous à la référence et aux tutoriels. L'ATmega328 sur l'Arduino Uno est livré avec un chargeur initial de programme préchargé qui vous permet de télécharger son nouveau code sans l'utilisation d'un programmeur physique externe.



Figure 38: face avant de la carte Arduino

Les broches d'alimentation se répartissent comme suit :

- VIN. La tension en entrée vers la carte Arduino quand on utilise une source d'alimentation externe (par opposition aux 5 V provenant de la connexion USB ou d'une autre source d'alimentation régulée). Vous pouvez fournir une tension au travers de cette broche, ou, si l'alimentation passe par la prise électrique, y accéder au travers de cette broche.
- 5V. La broche émet du 5 V régulé depuis le régulateur de la carte. La carte peut être alimentée avec du courant depuis la prise électrique CC (7 à 12 V), le connecteur USB (5 V), ou la broche VIN de la carte (7 à 12 V). La tension d'alimentation au travers des broches 5 ou 3,3 V contourne le régulateur et peut endommager votre carte. Nous ne vous le conseillons pas.
- 3V3. Une alimentation de 3,3 V générée par le régulateur intégré. Le flux maximum de courant est de 50 mA.
- GND. Les broches de Terre (masse).
- IOREF. Cette broche de la carte Arduino fournit la tension de référence à laquelle le microcontrôleur fonctionne. Un blindage correctement configuré peut lire la tension de la broche IOREF et sélectionner la source d'alimentation appropriée ou activer les convertisseurs de tension sur les sorties pour travailler à 5 ou 3,3 V.

Mémoire :

L'ATmega328 dispose de 32 Ko (dont 0,5 Ko utilisés pour le chargeur initial de programme). Il dispose également de 2 Ko de SRAM et d'1 Ko de mémoire EEPROM (qui peut être lue et écrite avec la bibliothèque de l'EEPROM).

III.6.2) Arduino méga 2560 :



Figure 39: La carte Arduino Mega 2560

Vue d'ensemble :

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560 ([fiche technique](#)).

Cette carte dispose :

- de 54 (!) broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties PWM (largeur d'impulsion modulée)),
- de 16 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques),
- de 4 UART (port série matériel),
- 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).

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, mais ceci n'est pas indispensable, l'alimentation étant fournie par le port USB).

Chapitre II

La carte Arduino Mega 2560 est compatible avec les circuits imprimés prévus pour les cartes Arduino Uno, Duemilanove ou Diecimila.

Alimentation :

La carte Arduino Mega 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500mA) ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte.

L'alimentation externe (non-USB) peut être soit un adaptateur secteur (pouvant fournir typiquement de 3V à 12V sous 500mA) ou des piles (ou des accus). L'adaptateur secteur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte. Les fils en provenance d'un bloc de piles ou d'accus peuvent être insérés dans les connecteurs des broches de la carte appelées Gnd (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation.

La carte peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Uno est entre 7V et 12V.

La carte Arduino Mega2560 diffère de toutes les cartes précédentes car elle n'utilise pas le circuit intégré FTDI usb-vers-série. A la place, elle utilise un Atmega8U2 programmé en convertisseur USB-vers-série.

Les broches d'alimentation sont les suivantes :

- 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). Vous pouvez 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.
- 3V3. Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V). L'intensité maximale disponible sur cette broche est de 50mA
- GND. Broche de masse (ou 0V).

Mémoire :

L'ATmega 2560 a 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le bootloader). L'ATmega 2560 a également 8 ko de mémoire

Chapitre II

SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM) .

Pour info : Le bootloader est un programme préprogrammé une fois pour toute dans l'ATméga et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte.

Entrées et sorties numériques :

Chacune des 54 broches numériques de la carte Mega 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 :

- **Communication Serie:** Port Serie Serial : 0 (RX) and 1 (TX); Port Serie Serial 1: 19 (RX) and 18 (TX); Port Serie Serial 2: 17 (RX) and 16 (TX); Port Serie Serial 3: 15 (RX) and 14 (TX). Utilisées pour recevoir (RX) et transmettre (TX) les données séries de niveau TTL. Les broches 0 (RX) and 1 (TX) 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 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), et 21 (interrupt 2). 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 0 à 13. Fournissent une impulsion PWM 8-bits à l'aide de l'instruction `analogWrite()`.
- **SPI (Interface Série Périphérique):** Broches 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). 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 Uno, Duemilanove et Diecimila.
- **I2C:** Broches 20 (SDA) et 21 (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"`). Noter que ces broches n'ont pas le même emplacement que sur les cartes Uno, Duemilanove ou Diecimila.
- **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.

Broches analogique :

Chapitre II

La carte Mega2560 dispose de 16 entrées analogiques, 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.

Note : les broches analogiques peuvent être utilisées en tant que broches numériques.

III.6.3) Programmation des cartes arduino :

Interface du logiciel

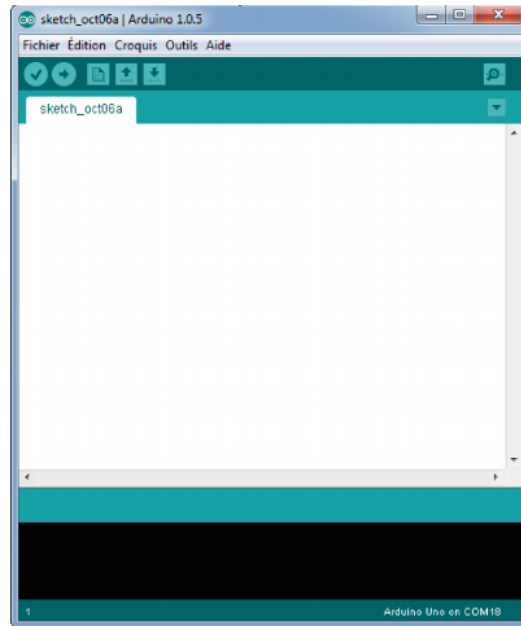
Lancement du logiciel

Lançons le logiciel en double-cliquant sur l'icône avec le symbole "infinie" en vert. C'est l'exécutable du logiciel. Après un léger temps de réflexion, une image s'affiche :



Cette fois, après quelques secondes, le logiciel s'ouvre. Une fenêtre se présente à nous :

Chapitre II

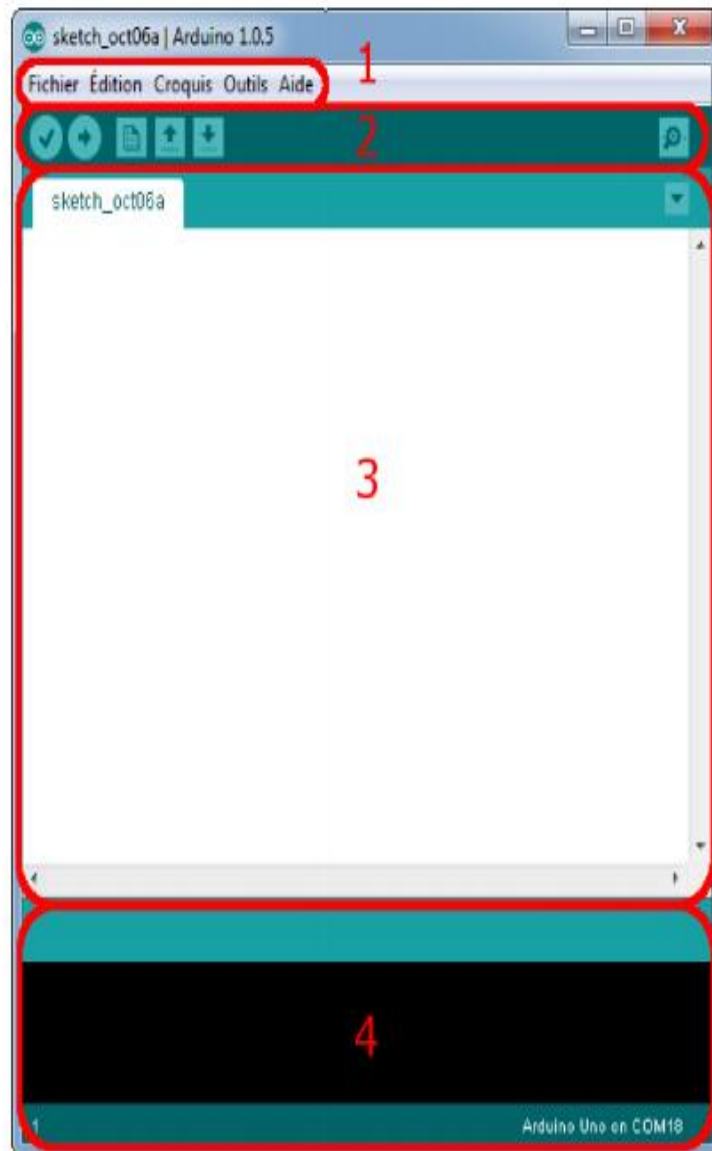


Ce qui saute aux yeux en premier, c'est la clarté de présentation du logiciel. On voit tout de suite son interface intuitive. Voyons comment se compose cette interface.

Chapitre II

Présentation du logiciel

J'ai découpé, grâce à mon ami paint.net, l'image précédente en plusieurs parties :



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

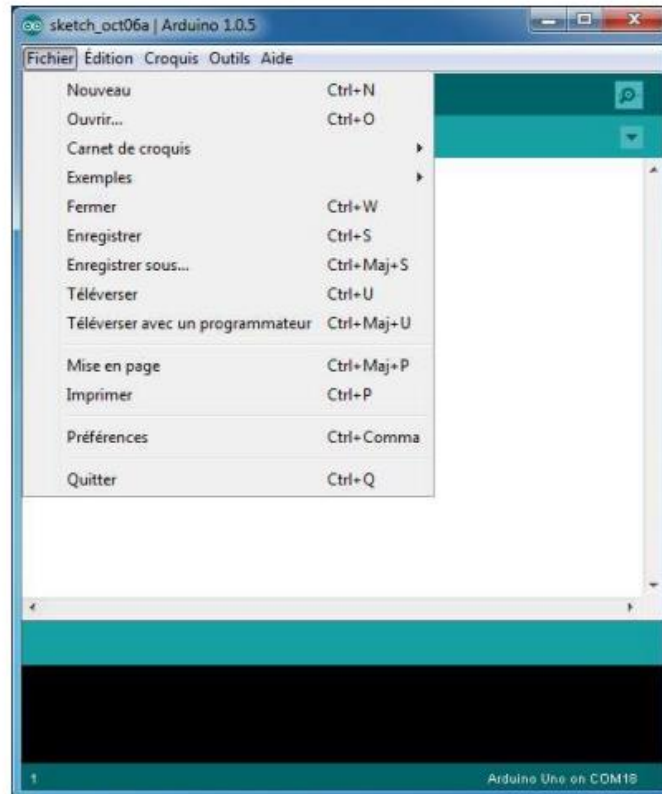
Chapitre II

Approche et utilisation du logiciel

Attaquons-nous plus sérieusement à l'utilisation du logiciel. La barre des menus est entourée en rouge et numérotée par le chiffre 1.

Le menu *File*

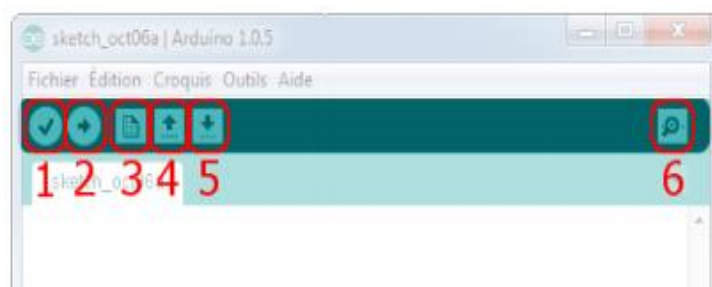
C'est principalement ce menu que l'on va utiliser le plus. Il dispose d'un certain nombre de choses qui vont nous être très utiles. Il a été traduit en français progressivement, nous allons donc voir les quelques options qui sortent de l'ordinaire :



- Carnet de croquis : ce menu regroupe tous les fichiers que vous avez pu faire jusqu'à maintenant.
- Exemples : ceci est important, toute une liste se déroule pour afficher les noms d'exemple de programme existant, avec ça on peut aider nos même a créé des programmes.
- Téléverser : permet d'envoyer sur la carte arduino.
- Préférences : vous pourrez régler ici quelque paramètre du logiciel.

Les boutons

Voyons à présent à quoi servent les boutons, encadrés en rouge et numérotés par le chiffre 2.



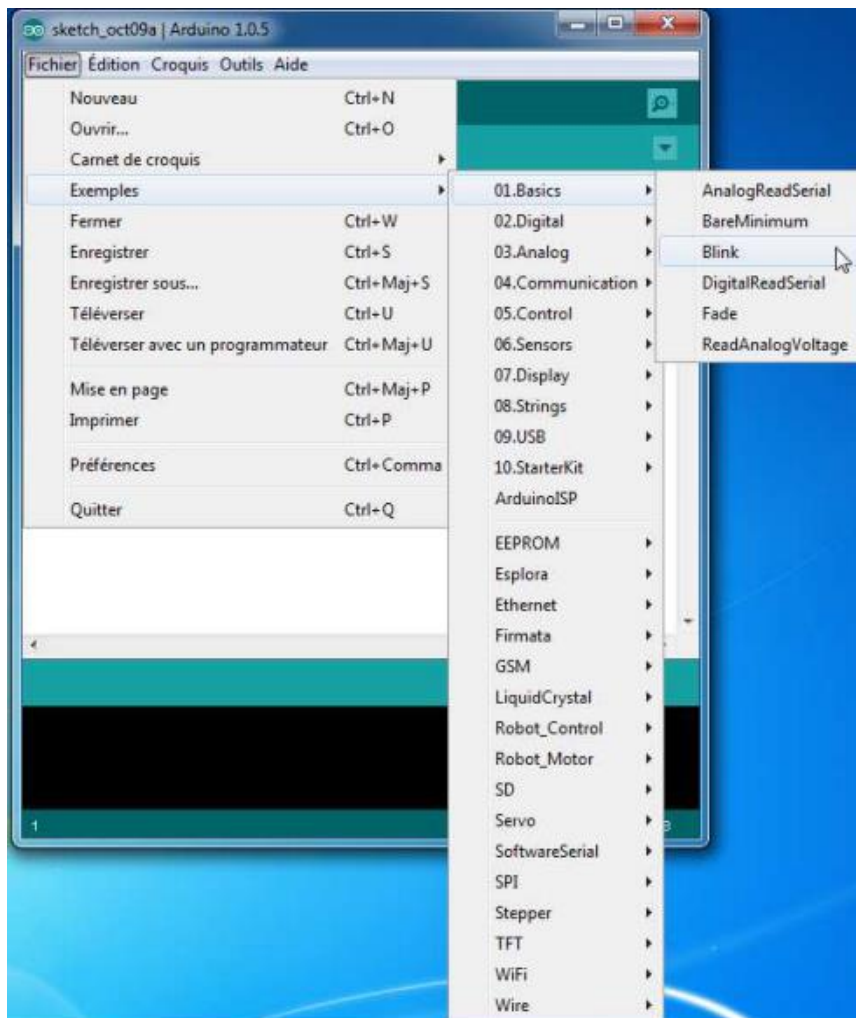
Chapitre II

Les boutons :

- Bouton 1 : ce bouton permet de vérifier le programme, il actionne un module qui cherche les erreurs dans votre programme.
- Bouton 2 : charger le programme dans la carte arduino.
- Bouton 3 : créé un nouveau fichier.
- Bouton 4 : ouvre un fichier.
- Bouton 5 : enregistrée un fichier.
- Bouton 6 : ouvre le moniteur série.

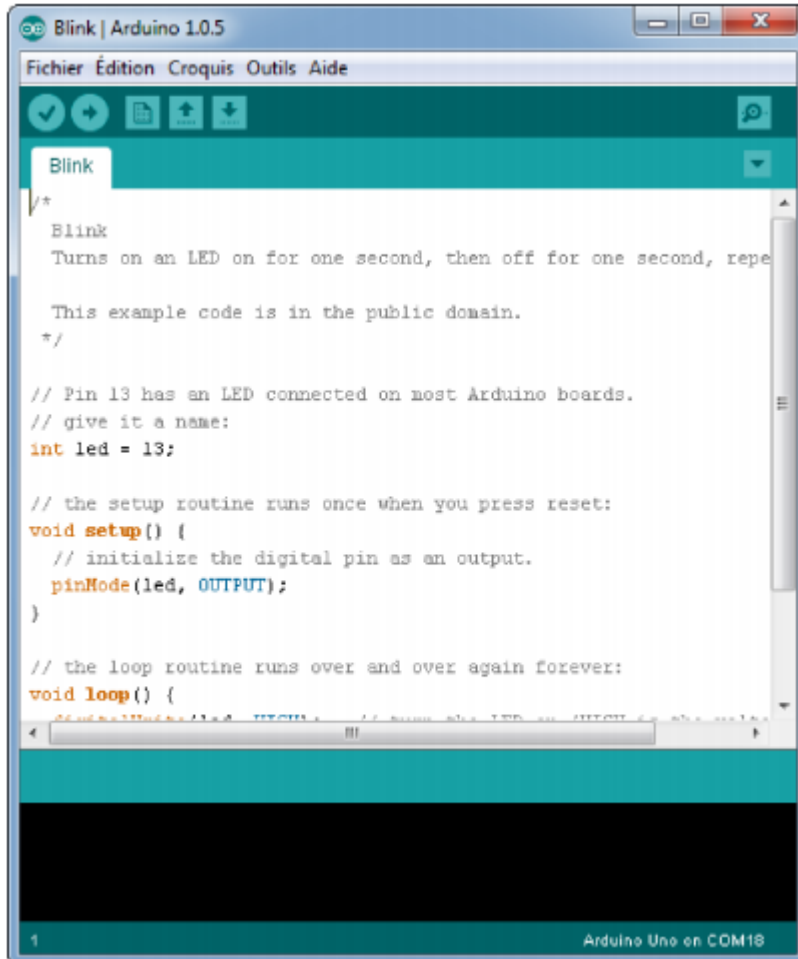
1^{er} étape : ouvrir un programme

Nous allons choisir un exemple tout simple qui consiste à faire clignoter une LED. Son nom est blink vous le trouverez dans la catégorie basics.



Chapitre II

Une fois que vous avez cliqué sur Blink, une nouvelle fenêtre va apparaître. Elle va contenir le programme Blink. Vous pouvez fermer l'ancienne fenêtre qui va ne nous servir plus à rien.



```
Blink | Arduino 1.0.5
Fichier Édition Croquis Outils Aide
Blink
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

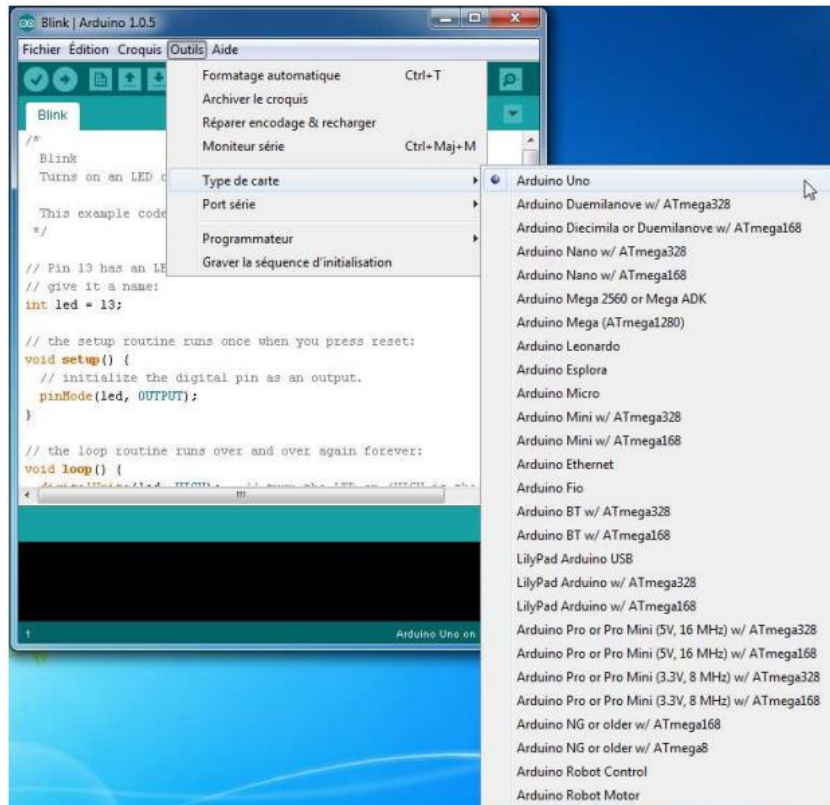
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(led, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

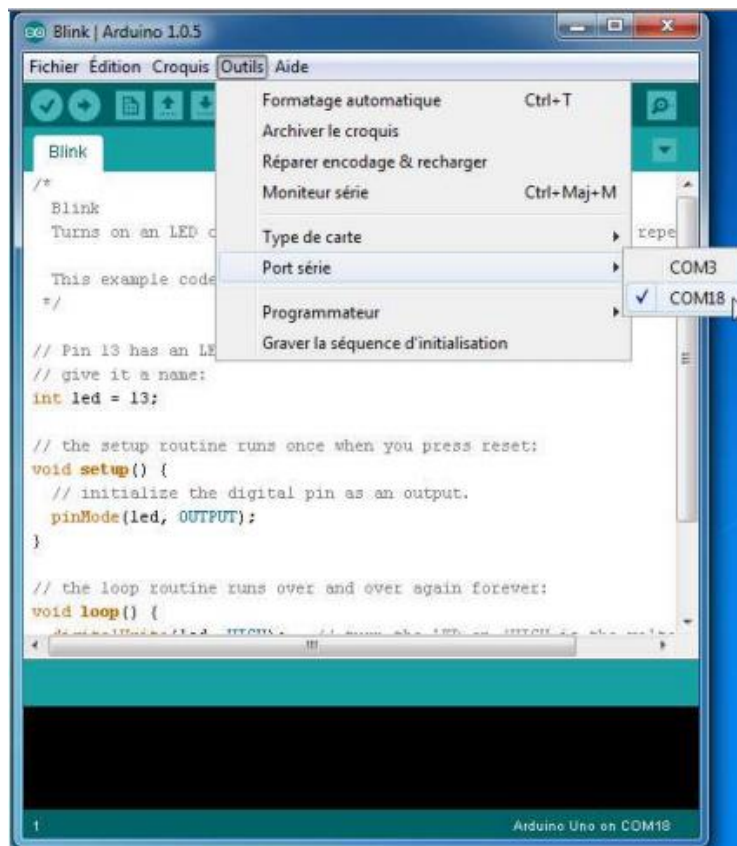
2ème étape :

Avant d'envoyer le programme *Blink* vers la carte, il faut dire au logiciel quel est le nom de la carte et sur quel port elle est branchée. Choisir la carte que l'on va programmer. Ce n'est pas très compliqué, le nom de votre carte est indiqué sur elle. Pour nous, il s'agit de la carte "Uno". Allez dans le menu "Tools" ("outils" en français) puis dans "Board" ("carte" en français). Vérifiez que c'est bien le nom "Arduin Uno" qui est coché. Si ce n'est pas le cas, cochez-le.

Chapitre II

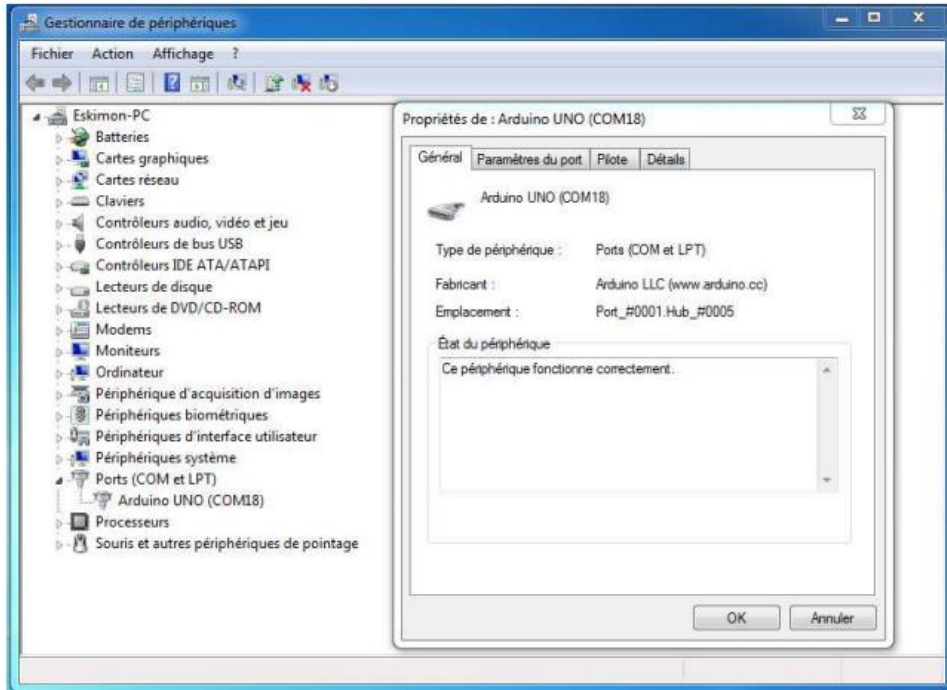


Choisissez le port de connexion de la carte. Allez dans le menu Tools, puis Serial port. Là, vous choisissez le port COM X, X étant le numéro du port qui est affiché. Ne choisissez pas COM1 car il n'est quasiment jamais connecté à la carte. Dans mon cas, il s'agit de COM18 :



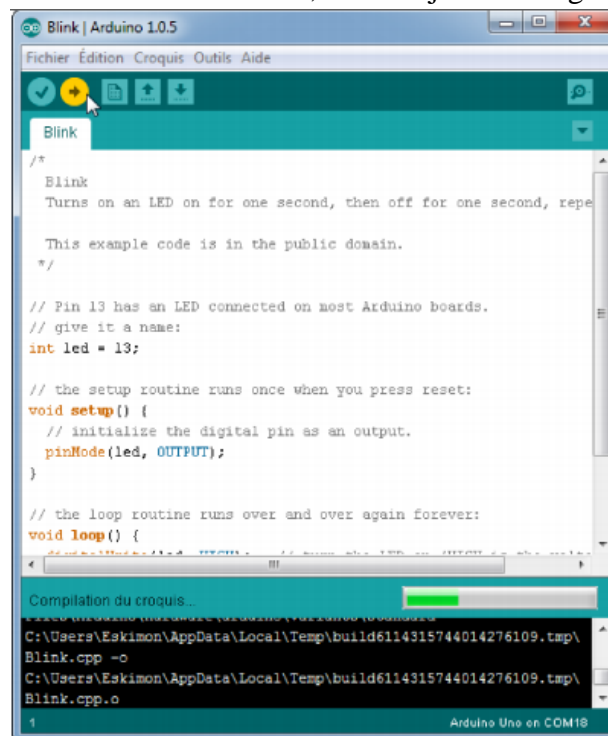
Chapitre II

Pour trouver le port de connexion de la carte, vous pouvez aller dans le gestionnaire de périphérique qui se trouve dans le panneau de configuration. Regardez à la ligne Ports (COM et LPT) et là, vous devriez avoir Arduino Uno (COMX). Aller, une image pour le plaisir :



Dernière étape :

Très bien. Maintenant, il va falloir envoyer le programme dans la carte. Pour ce faire, il suffit de cliquer sur le bouton Téléverser, en jaune-orangé sur la photo :

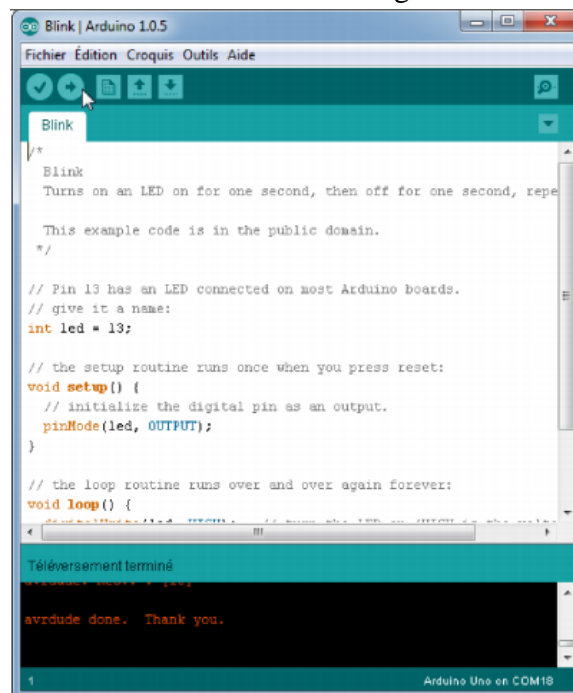


Chapitre II

Vous verrez tout d’abord le message “Compilation du croquis en cours.....” pour vous informer que le programme est en train d’être compilé en langage machine avant d’être envoyé. Ensuite vous aurez ceci :



En bas dans l’image, vous voyez le texte : “ Téléversement“, cela signifie que le logiciel est en train d’envoyer le programme dans la carte. Une fois qu’il a fini, il affiche un autre message :



Le message afficher : “ Téléversement terminé” signale que le programme à bien été chargé dans la carte. Si votre matériel fonctionne, vous devriez avoir une LED sur la carte qui clignote.

IV) conclusion :

Dans ce chapitre j'ai montré les étapes de conception que j'ai répartie en deux parties, partie mécanique, partie électrique.

Dans la partie mécanique j'ai parlé des éléments essentiels constitutifs du robot tel que le système de guidage et du déplacement. Et dans la partie électrique j'ai mentionné l'actionneur essentiel qui est le moteur pas à pas et j'ai expliqué la commande de ce moteur avec une carte de commande basé sur Arduino et les circuits de puissance le A4983 et le L298N. Par la suite j'ai donné une explication de la façon à programmer les modules Xbee qui sert à envoyer l'information.

Dans le chapitre suivant je vais présenter la fabrication du prototype suivant la démarche de conception présenté précédemment.



Chapitre 3

REALISATION PRATIQUE

I) introduction :

Soit pour l'usinage 2D, la maîtrise du déplacement d'un outil dans les 2 dimensions de l'espace est devenue primordiale. Ce déplacement dans les 2 axes de l'espace (X, Y) nécessite de maîtriser 3 grands domaines techniques <<électronique, mécanique et informatique>>

Dans ce chapitre nous allons d'écrire les solutions adoptées par détail pour répondre aux spécifications de notre cahier de charge et ce en abordant la conception détaillée de chaque partie du système.

II) Structure mécanique :

Pour réaliser un système mécanique permettant de déplacer un objet dans l'espace, nous avons opté pour l'aluminium pour la structure des axes et de la base car il est plus léger que l'acier et pour les vis sans fin et les glissières on les a faites avec de l'acier.

On a ici dans la figure suivante la structure complète du robot qui est composée de deux parties, partie de commande (partie électrique), partie mécanique.

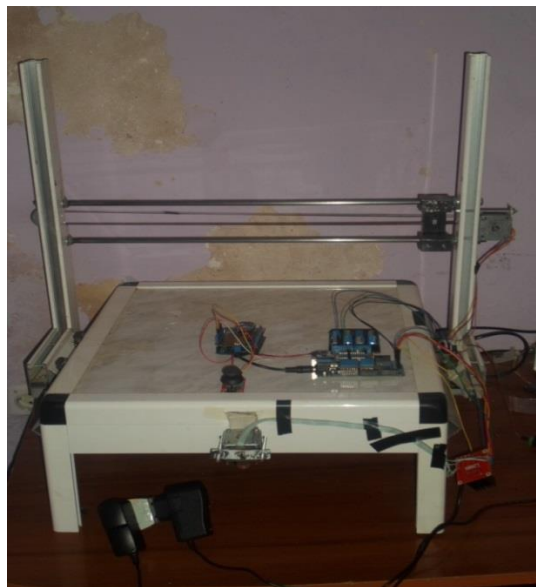


Figure 40 : structure complète du robot

II.1) Les différentes parties mécaniques du robot :

La partie mécanique du robot est constituée de trois parties essentielles qui sont : la base, l'axe Y, l'axe X.

a) La base



Figure 41: la structure de la base du robot

b) L'axe X

L'axe X est constitué d'une plaque métal et de deux glissières entrainé par une courroie



Figure 42: l'axe X

c) L'axe Y

L'axe Y est constitué de deux parties reliées par des vis comme le montre la figure suivante et il est entrainé par une courroie.



Figure 43: l'axe Y



Figure 44: courroie qui entraine l'axe Y

III) Assemblage complet de la partie mécanique du robot

Après avoir cité les différentes parties, nous allons les assembler pour avoir un robot complet, la figure ci-dessous le montre.



Figure 45: assemblage complet des deux axes avec la base

IV) les différentes parties électriques du robot :

La partie électrique du robot est constituée des cartes de commande basées sur Arduino et des circuits de puissance et des modules de transmission Xbee, pour l'information visuelle on a opté pour une caméra sans fil qui est polarisée d'une manière à voir bien la position du robot. Pour les actionneurs on utilise des moteurs pas à pas.

IV.1) les cartes de commande du robot :

La carte de commande est constituée d'une carte Arduino UNO et d'une Arduino méga 2560, alimentée par l'ordinateur, et de deux drivers de moteur pas à pas.

La carte Arduino délivre les signaux PWM aux drivers de moteur pas à pas pour commander les différents moteurs qui contrôlent les axes du robot, qui sont de leur côté alimentés avec une tension de 12 volts délivrée par un générateur de tension,

La figure ci-dessous montre le schéma de branchement de la partie électrique.

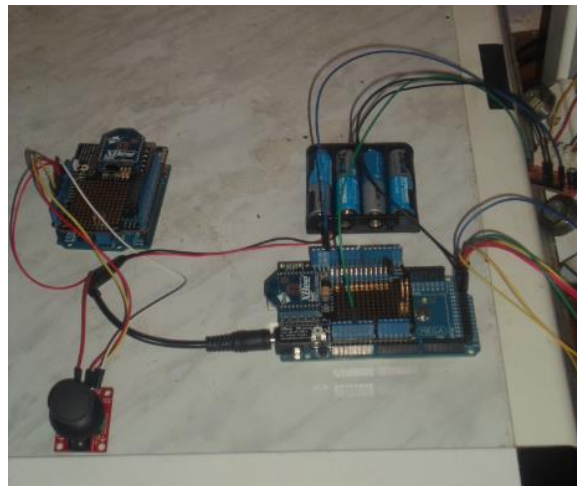


Figure 46: schéma de branchement

Les moteurs pas à pas peuvent être commandés par plusieurs drivers et notre choix c'est porté sur le A4983 et sur le L298N car ils disposent des caractéristiques suivantes

- Simple à utiliser.
- Possède une interface de contrôle de la direction.
- Fonctionne avec 8-35V et fournit jusqu'à 2A par bobine.
- Limite de courant ajustable.
- Cinque résolutions de micros pas différentes : pas complet, demi pas, quart de pas huitième de pas et seizième de pas.
- Protection contre les courts circuits et les surcharges.

Chapitre III

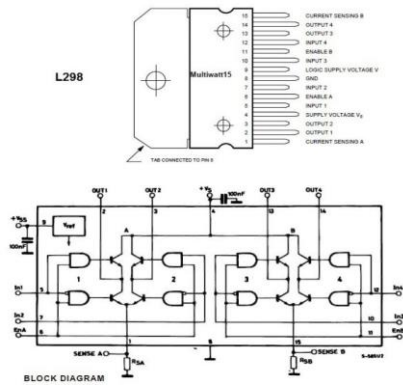
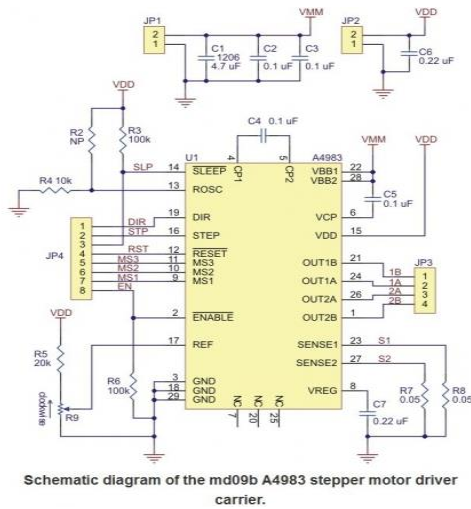


Figure 47: le datasheet du driver L298N



Schematic diagram of the md09b A4983 stepper motor driver carrier.

Figure 48: datasheet du driver A4983

IV.2) Les modules Xbee :

Dans notre cas j'ai utilisé des modules Xbee de référence X24C. Pour assurer la liaison entre les deux module on doit les configuré la première en émetteur et le deuxième en récepteur avec le logiciel Digi.

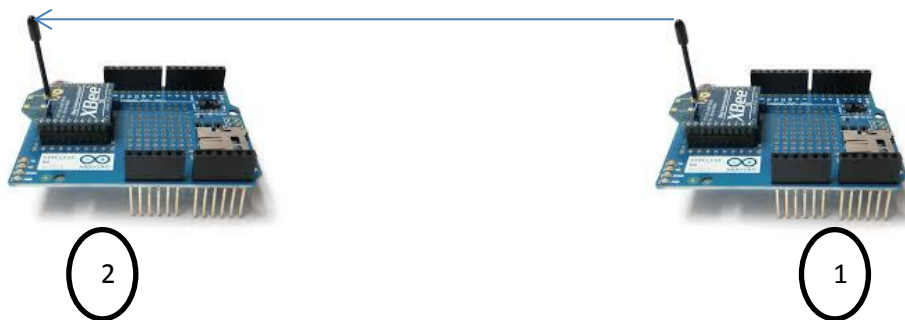


Figure 49 : transmission d'information avec des Xbee

IV.3) la caméra sans fil :

Dans notre projet on a utilisé une caméra sans fil pour récupérer l'information visuel pour assurer la une bonne commande à distance et pour avoir une précision sur la position du robot.

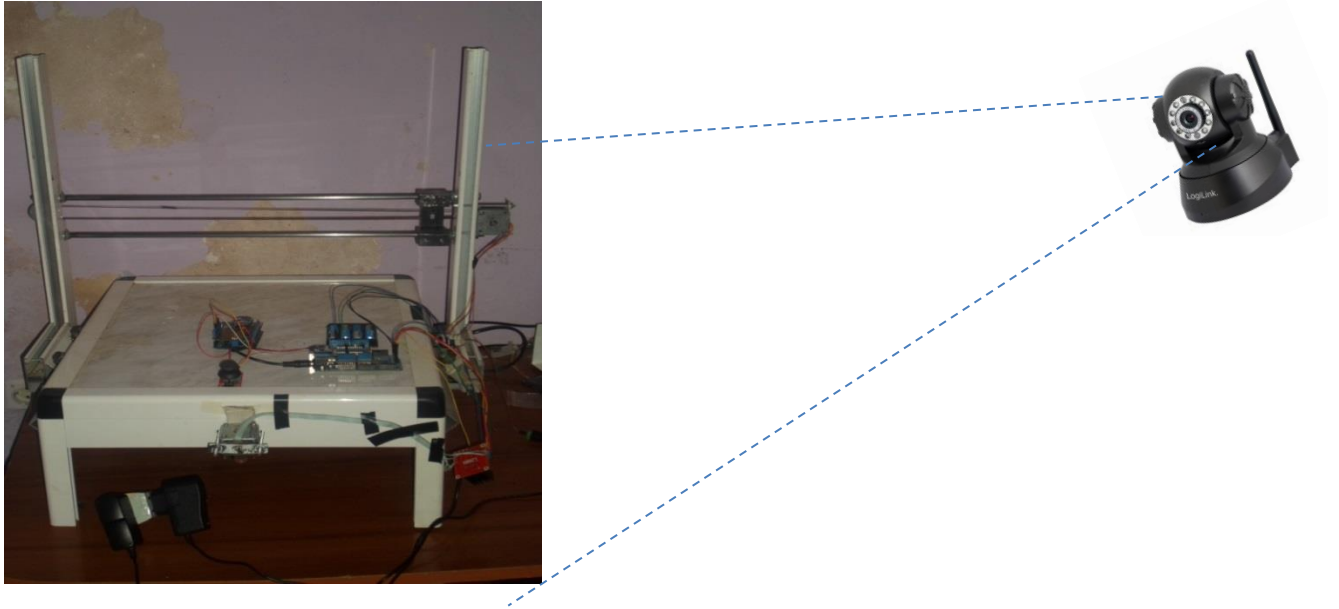


Figure 50 : positionnement de la caméra

IV.4) les actionneurs

Il existe plusieurs types de moteurs pas à pas mais nous avons opté pour des moteurs qui ont une plus grande résolution et un grand couple qui sont de référence <<Sanyo Denki>>

IV.4.1) Caractéristiques des moteurs

- Model 103-591-0155 Sanyo Denki 0,9deg/pas.
- Tension de source 24v.
- Courant opérationnel 0.85A/phase.
- Inductance 3,7 mH/phase.
- Résistance 4,3Ω/phase.

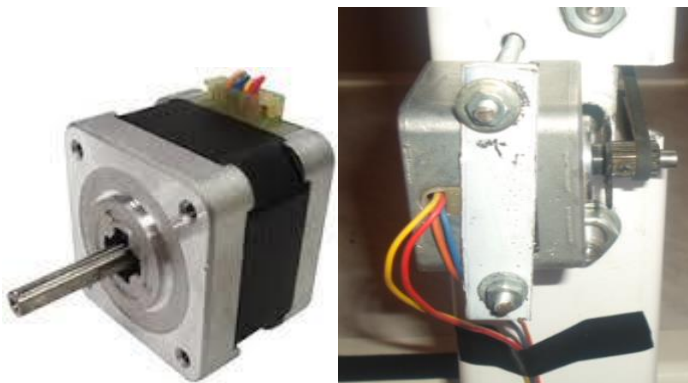


Figure 51: moteur pas à pas Sanyo Denki

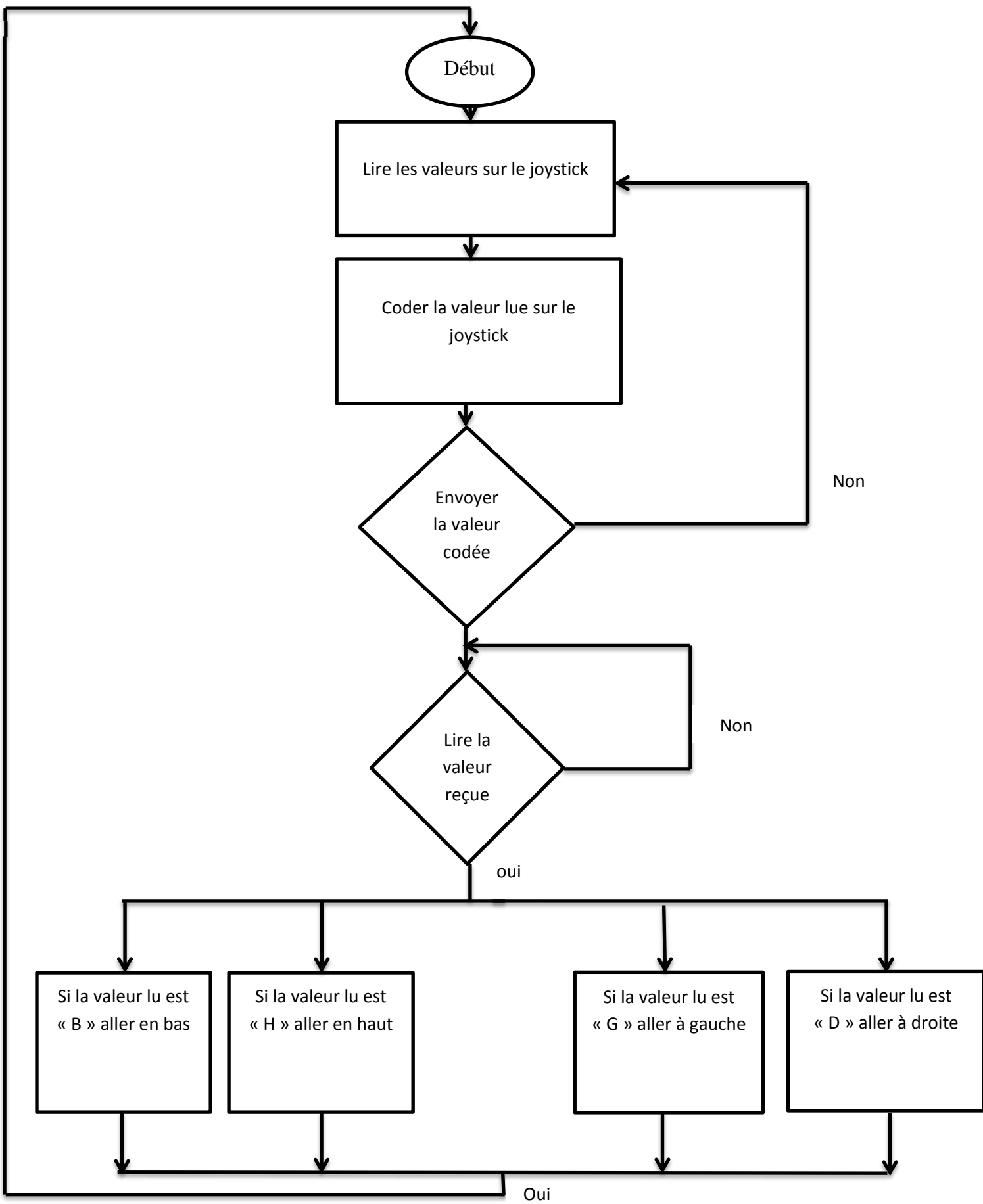
V) Structure complète de la réalisation :

Après l'assemblage des parties mécaniques et électriques on obtient la structure complète du robot.



Figure 52 : Structure complète du robot

VI) L'organigramme



VII) conclusion

Dans ce chapitre, nous avons pu décrire les différents éléments constitutifs du robot et de le construire, puis injecter le programme dans la carte Arduino. Pour commander les axes à distance on a utilisé des Xbee (réalisation d'un système de télé opération) et une caméra sans fil pour la placé dans un endroit voulu pour couvrir tout l'espace ou se trouve le robot pour donner à l'opérateur une bonne information sur l'endroit de la machine.

Conclusion générale

Ce projet m'a permis d'approfondir mes connaissances dans le domaine de la télé opération.

Et j'ai présenté la conception de la commande à distance d'un robot rigide a deux axes en choisissant les bonne carte de commande qui sont basé sur Arduino car elles sont des carte moins cher et disponible sur le marché et simple a programmé, aussi on a utilisé les carte de puissance tel que le A4983 et le L298N. Et ce qui concerne l'envoi des donné j'ai choisi des module a base sur zigbee qui sont simple a utilisé et pour une consommation d'énergie faible. Pour le choix des moteurs on a opté pour les moteurs pas à pas car ils ont une grande précision et simple a commandé en position et en vitesse au contraire des moteurs à courant continu ou on trouve souvent des problèmes de stabilité.

Pour asservir mon système on a besoin d'un capteur visuel pour cela j'ai utilisé une caméra qui va m'informer sur la position du robot à chaque moment.

En conclusion, j'espère que ce modeste travail pourrait servir de référence aux projets futurs des prochaines promotions et les inciter à s'intéresser d'avantage au côté pratique de l'automatique.

Annexe

Datasheet des moteurs pas à pas de type Sanyo Denki

SANMOTION F2
2-phase STEPPING SYSTEM

Stepping motor Specifications



2-phase stepping motor

56mm sq.(2.20inch sq.)

103H712 □
1.8° /step

Unipolar winding

Model		Holding torque at 2-phase energization	Rated current	Wiring resistance	Winding inductance	Rotor inertia	Mass(Weight)
Single shaft	Double shafts	[N·m(oz·in) MIN.]	A/phase	Ω /phase	mH/phase	[×10 ⁻⁴ kg·m ² (oz·in ²)]	[kg(lbs)]
103H7121-0140	-0110	0.39(55.2)	1	4.8	8	0.1(0.55)	0.47(1.04)
103H7121-0440	-0410	0.39(55.2)	2	1.25	1.9	0.1(0.55)	0.47(1.04)
103H7121-0740	-0710	0.39(55.2)	3	0.6	0.8	0.1(0.55)	0.47(1.04)
103H7123-0140	-0110	0.83(117.)	1	6.7	15	0.21(1.15)	0.65(1.43)
103H7123-0440	-0410	0.83(117.5)	2	1.6	3.8	0.21(1.15)	0.65(1.43)
103H7123-0740	-0710	0.78(110.5)	3	0.77	1.58	0.21(1.15)	0.65(1.43)
103H7124-0140	-0110	0.98(138.8)	1	7	12.5	0.245(1.34)	0.8(1.76)
103H7124-0440	-0410	0.98(138.8)	2	1.7	3.1	0.245(1.34)	0.8(1.76)
103H7124-0740	-0710	0.98(138.8)	3	0.74	1.4	0.245(1.34)	0.8(1.76)
103H7126-0140	-0110	1.27(179.8)	1	8.6	19	0.36(1.97)	0.98(2.16)
103H7126-0440	-0410	1.27(179.8)	2	2	4.5	0.36(1.97)	0.98(2.16)
103H7126-0740	-0710	1.27(179.8)	3	0.9	2.2	0.36(1.97)	0.98(2.16)

Bipolar winding

Model		Holding torque at 2-phase energization	Rated current	Wiring resistance	Winding inductance	Rotor inertia	Mass(Weight)
Single shaft	Double shafts	[N·m(oz·in) MIN.]	A/phase	Ω /phase	mH/phase	[×10 ⁻⁴ kg·m ² (oz·in ²)]	[kg(lbs)]
103H7121-5640	-5610	0.55(77.9)	1	4.3	14.5	0.1(0.55)	0.47(1.04)
103H7121-5740	-5710	0.55(77.9)	2	1.1	3.7	0.1(0.55)	0.47(1.04)
103H7121-5840	-5810	0.55(77.9)	3	0.54	1.74	0.1(0.55)	0.47(1.04)
103H7123-5640	-5610	1.0(141.6)	1	5.7	29.4	0.21(1.15)	0.65(1.43)
103H7123-5740	-5710	1.0(141.6)	2	1.5	7.5	0.21(1.15)	0.65(1.43)
103H7123-5840	-5810	1.0(141.6)	3	0.7	3.5	0.21(1.15)	0.65(1.43)
103H7126-5640	-5610	1.6(226.6)	1	7.7	34.8	0.36(1.97)	0.98(2.16)
103H7126-5740	-5710	1.6(226.6)	2	2	9.1	0.36(1.97)	0.98(2.16)
103H7126-5840	-5810	1.6(226.6)	3	0.94	4	0.36(1.97)	0.98(2.16)
103H7128-5640	-5610	2(283.2)	1	8.9	40.1	0.49(2.68)	1.3(2.87)
103H7128-5740	-5710	2(283.2)	2	2.3	10.4	0.49(2.68)	1.3(2.87)
103H7128-5840	-5810	2(283.2)	3	1.03	4.3	0.49(2.68)	1.3(2.87)

Annexe

Schéma des pins d'un ATmega 328P

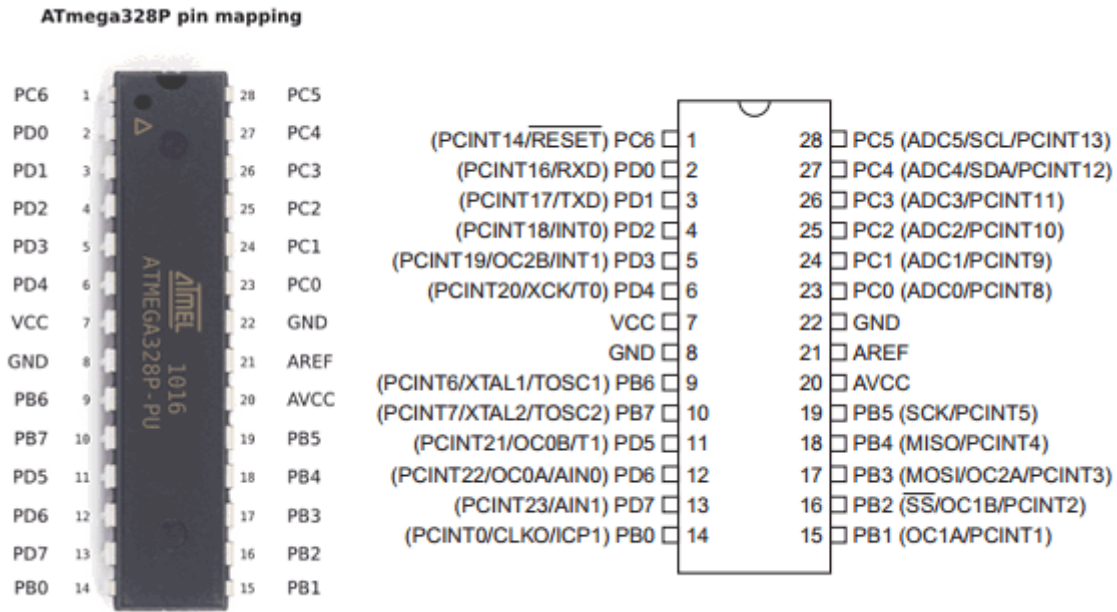
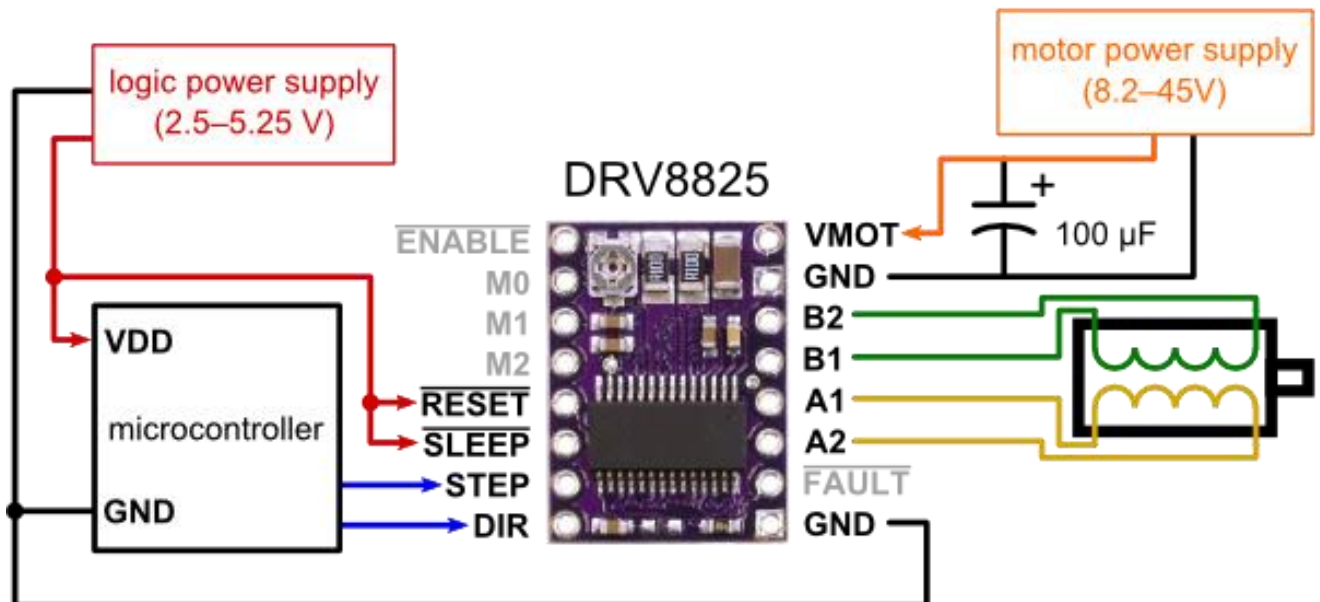


Schéma de branchement d'un moteur pas à pas ver un driver et d'un driver ver un microcontrôleur



Matériel officiel Arduino:

Les modules d'origine des différentes versions de l'Arduino sont fabriqués par la société italienne Smart Projects. Quelques-unes des cartes de marque Arduino ont été conçues par la société américaine SparkFun Electronics.

Dix-sept versions des cartes de type Arduino ont été produites et vendues dans le commerce à ce jour dans l'ordre chronologique ci-dessous :

1. Serial Arduino programmé avec une connexion série par connecteur DB9 et utilisant un ATmega8.
2. Arduino Extreme, programmable via une connexion USB et utilisant un ATmega8.
3. Arduino Mini, une version miniature de l'Arduino utilisant un ATmega168 de type CMS.
4. Arduino Nano, une version encore plus petite de l'Arduino alimenté par USB et utilisant un ATmega168 (avant la version 3) ou ATmega328 (à partir de la version 3.0) de type CMS.
5. LilyPad Arduino, une conception de type minimaliste pour permettre une application portable utilisant un ATmega168 de type CMS, cette carte se remarque par son vernis violet.
6. Arduino NG, programmable via une connexion USB et utilisant un ATmega8.
7. Arduino NG plus, programmable via une connexion USB et utilisant un ATmega168.
8. Arduino Bluetooth(BT), programmable via une connexion Bluetooth et utilisant un ATmega 3286.
9. Arduino Diecimila, possède une interface USB et utilise un ATmega168 dans un boîtier format DIL28. (16 ko flash, 1 ko SRAM, 0,5 ko EEPROM)
10. Arduino Duemilanove (2009) utilise un ATmega168 et est alimenté en électricité par le connecteur USB ou une alimentation externe avec commutation automatique⁷. La nouvelle version⁸ est équipée d'un ATmega328 (32 ko de flash, 2 ko de SRAM, et 1 ko d'EEPROM).
11. Arduino Mega est équipé d'un ATmega1280 de type CMS pour avoir des Entrées/Sorties supplémentaires et de la mémoire (128 ko flash, 8 ko SRAM, 4 ko EEPROM).
12. Arduino Uno utilise un ATmega328 comme les derniers modèles de Duemilanove, mais alors que le Duemilanove utilisait une puce FTDI pour la programmation via un connecteur USB, le Uno utilise une puce ATmega8U2 programmé comme un convertisseur série¹⁰.
13. Arduino Mega2560 est équipé d'un ATmega2560 de type CMS, augmentant la mémoire totale disponible à 256 ko. Il est équipé aussi de la nouvelle puce USB ATmega8U2¹¹.

Annexe

14. Arduino Ethernet est une carte Arduino UNO intégrant un chip Wiznet W5100 pour rajouter la connectivité Ethernet intégrée.
15. Arduino Leonardo est une version bas coût de l'Arduino UNO à base d'un ATmega32U4.
16. Arduino DUE est une évolution de l'Arduino Mega2560 avec un micro-contrôleur 32 bits Atmel SAM3X (ARM 32 bits Cortex-M3).
17. Arduino Esplora est une carte dérivée de l'Arduino Leonardo. Elle a la forme d'une manette de jeu. Contrairement aux autres Arduino, c'est une carte « tout-en-un » qui comporte de nombreux capteurs (température, accélération, lumière, microphone, potentiomètre...) ainsi que 4 boutons poussoirs, un potentiomètre et un joystick analogique.

Bibliographie

- [1] www.technologystudent.com
- [2] **Huglres Austin**, <<Electronic Motors and Driver>> 2013
- [3] **Oqui C Patrice**, <<Moteurs pas à pas et pc>> 2004
- [4] **Kenjo Takashi**, <<Stepping Motors and their Microprocessor controls>> 1995
- [5] **Alain Pruski**, <<robotique générale>> 1988, 207p
- [6] **Nassimo BANZI**, <<Démarrerez avec Arduino>> 2013, 102P
- [7] **Patrick Chantereau et Erik Bartmann**, <<Le grand livre d'Arduino>>.
- [8] **Bruno Affagard, Jean-Michel Gérardan, Jean-Noël Lafargue**, << Projets créatifs avec Arduino>> 2014
- [9] **N. ANDREFF, B. ESPIAU et R. HORAUD**, <<Visual servoing from lines >> Août 2002.
- [10] **J. BARRETO et H. ARAUJO**, <<Geometric properties of central catadioptric line images>> Mai 2002.
- [11] **D. BURSHKA, J. GEIMAN et G. HAGER**, << Optimal landmark configuration for vision based control of mobile robot>> Septembre 2003.
- [12] **A. Alhaj, C. Collewet, F. Chaumette**, << Visual servoing based on dynamic vision>> Septembre 2003.
- [13] **F. Chaumette**, << Asservissement visuel *Commande des robots manipulateurs*>> 2002.
- [14] **R. Murray, z. Li ET S. sastry**, << A mathematical introduction to robotic manipulation, CRC Press>>.