



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

**UNIVERSITE ORAN 2 MOHAMED BEN AHMED**  
INSTITUT DE MAINTENANCE ET DE SECURITE INDUSTRIELLE  
DEPARTEMENT DE MAINTENANCE EN INSTRUMENTATION

**MEMOIRE**  
**DE FIN D'ETUDE POUR OBTENTION DU DIPLOME DE MASTER**  
**Filière : Génie industriel**  
**Spécialité : L'ingénierie de la maintenance en instrumentation**

**THEME**  
**Commande de deux moteurs pas à pas pour un système**  
**D'irrigation**

**Réalisé par** BELHADRI RAJAA Et DOUAIDI Zouheir Khaled

Soutenu publiquement

Le : 13/07/2017

Devant le jury :

|                           |     |                      |
|---------------------------|-----|----------------------|
| Mr MEKKI Ibrahim Elkhalil | MCA | Président            |
| Mr NEKROUF Djilali        | MAA | Encadreur/rapporteur |
| Mr ROUAN Serik Mehdi      | MCB | Examineur            |

**Promotion 2021-2022**

# REMERCIEMENTS

*Nous remercions tout d'abord « LE BON DIEU » tout puissant qui nous a donné le courage, la patience et la volonté.*

*Nous remercions notre promoteur MR NEKROUF. D, pour ses précieuses orientations, son aide et ses conseils tout au long de ce projet.*

*Nos sincères remerciements et notre gratitude s'adressent également au président de jurys ainsi que les membres de jurys d'avoir accepté d'évaluer et juger notre travail.*

*Nos remerciements les plus chaleureux vont à nos chères FAMILLES pour leurs encouragements, leur patience et leur grand soutien durant toutes ces années d'études.*

*Enfin, Il nous est agréable de remercier qui ont contribué de près ou de loin à la réalisation de ce travail.*

*MERCI*

## DEDICACE

*Je dédie ce travail :*

*En Premier Lieu A La Mémoire De Mes Grands Parents Que*

*Dieu Les Recueillera Dans Son Vaste Paradis.*

*Au Meilleur Des Pères BELHADRI Noureddine.*

*A Ma Très Chère Maman.*

*Qu'ils Trouvent En Moi La Source De Leur Fierté.*

*A ma grand-mère*

*A ma sœur ma princesse aicha Malék et mon frère Oussama*

*A mon oncle HORCH BILAL*

*A tous ceux qui me sont chers*

*« RANA »*

## DEDICACE

*Je dédie mon mémoire de fin d'étude de MASTER à tous ceux et celles qui m'ont soutenu à poursuivre mes études supérieures à commencer par mes amis et ma famille, et en particulier, ma mère, ma sœur et mes oncles.*

*Mais je ne remerciais jamais assez mon très cher père DOUALDI Mustapha qui m'a boosté dans mes études sans relâche et toujours de l'avant pour le grand bien de ma personne et pour ma réussite dans mes études.*

*La réussite dans mes études est due au soutien sans faille de mes parents et à un encouragement indéfectible de mon père à qui je lui dois tout et, par la présente, je lui témoigne mon infinie reconnaissance et gratitude.*

*Ma réussite est le fruit de leur sacrifice, de leur dévouement et de l'affection qu'ils m'ont toujours donnée et j'en suis fier et comblé.*

*Que DIEU, dans son infinie bonté, puisse leur accorder la bénédiction et la bonne santé, inshallah !*

«Zouheir »

# Sommaire

|  |    |
|--|----|
| Objectif .....   | 1  |
| Introduction générale.....   | 3  |
| <b>Chapitre 01 : Systèmes embarqués et l'évolution des automates<br/>programmables industriels</b> |    |
| 1-introduction.....  | 5  |
| 2-PRESENTATION DES SYSTEMES EMBARQUES.....   | 5  |
| 2-1-Définition.....  | 5  |
| 2-2-Historique .....   | 5  |
| 2-3-Caractéristiques d'un système embarqué .....   | 5  |
| 2-4-Architecture d'un système embarqué.....  | 6  |
| 2-4-1-Composants matériels d'un système embarqué.....  | 7  |
| a. Introduction .....  | 7  |
| b. Unité de calcul CPU.....  | 7  |
| c. Mémoire.....  | 8  |
| d. Alimentation :.....   | 8  |
| e. Actionneurs :.....  | 9  |
| 2-4-2-Partie logicielle :.....   | 9  |
| 2-5-Classification des systèmes embarqués.....   | 10 |
| 2-6-Domains d'application des systèmes embarqués.....  | 11 |
| 3-L'évolution de l'automate programmable industriel.....   | 11 |
| 3-1-Définitions d'un système automatisé.....   | 11 |
| 3-2-Structure des systèmes automatisés.....  | 11 |
| 3-3-composition d'un système automatisé.....   | 12 |
| 3-3-1-Une partie opérative (P.O).....  | 12 |
| 3-3-2-Une partie commande (P.C.).....  | 12 |
| 3-3-3-La partie dialogue :.....  | 12 |
| 3-4-Les avantages et les inconvénients des systèmes automatisés.....                               | 12 |
| 3-5-Les APIs.....  | 13 |
| 3-5-1-Historique.....  | 13 |
| 3-5-2-Définitions de l'Automate Programmable Industriel.....                                       | 13 |
| 3-5-3-Domains d'emploi des automates.....  | 14 |
| 3-5-4-Nature des informations traitées par l'automate.....   | 14 |

|  |    |
|--|----|
| 3-5-5-L'architecture d'un automate programmable industrielle.....              | 14 |
| 3-6-L'évolution de l'automate programmable industriel pendant les siècles..... | 15 |
| 4-Conclusion.....  | 16 |

## **Chapitre 02 : Le fonctionnement et la commande des moteurs pas à pas**

|  |    |
|--|----|
| 1-Introduction.....  | 18 |
| 2-La présentation d'un actionneur pas à pas.....             | 18 |
| 3-Applications industrielles.....                            | 19 |
| 4-Les avantages et les inconvénient du moteur pas à pas..... | 20 |
| 5-Les types de moteur pas à pas.....                         | 20 |
| 5-1-Moteur à réluctance variable.....                        | 20 |
| 5-1-1Constitution :.....                                     | 20 |
| 5-1-2-Le fonctionnement.....                                 | 20 |
| 5-1-3-Les avantages et Les inconvénient.....                 | 21 |
| 5-2-Moteur hybrides.....                                     | 22 |
| 5-2-1-Constitution.....                                      | 22 |
| 5-2-2-Fonctionnement.....                                    | 22 |
| 5-2-3-Les avantages.....                                     | 22 |
| 5-3-Moteur à aimant permanent.....                           | 23 |
| 5-3-1Constitution.....                                       | 23 |
| 5-3-2-Fonctionnement.....                                    | 23 |
| a. Les moteurs unipolaire.....                               | 24 |
| b. Les moteurs bipolaires.....                               | 24 |
| 6-La commande des moteurs pas à pas.....                     | 26 |
| 6-1-Différents modes de commandes .....                      | 26 |
| 6-1-1-Modes pas entier.....                                  | 26 |
| 6-1-2-Mode demi pas.....                                     | 26 |
| 6-2-Commande de Moteur unipolaire.....                       | 26 |
| 6-2-1-Mode monophasé et biphasé pas entier.....              | 26 |
| a. Monophasé.....  | 27 |
| b. Biphasé.....  | 27 |
| 6-2-2-Principe en mode demi – pas.....                       | 27 |
| 6-3-Moteur bipolaire.....                                    | 28 |
| 6-3-1-mode monophasé et biphasé pas entier.....              | 29 |

|   |    |
|---|----|
| a. Monophasé.....   | 29 |
| b. Biphassé.....  | 29 |
| 6-3-2-Principe en mode demi – pas.....                              | 30 |
| 7-L'alimentation des moteurs pas à pas unipolaire et bipolaire..... | 30 |
| 7-1-L'alimentation du moteur unipolaire.....                        | 31 |
| 7-2-L'alimentation du moteur bipolaire.....                         | 31 |
| 7-2-1-Le pont H.....  | 32 |
| 7-2-2-L'alimentation symétrique.....                                | 32 |
| 8-Conclusion.....   | 32 |

### **Chapitre 03: Description de la carte ARDUINO**

|   |    |
|---|----|
| 1-Introduction.....                                 | 35 |
| 2-Présentation de la carte ARDUINO.....             | 35 |
| 3-Les différentes gammes de la carte ARDUINO.....   | 35 |
| 4-Étude de La carte ARDUINO MEGA 2560.....          | 37 |
| 4-1-Présentation de la carte ARDUINO MEGA 2560..... | 37 |
| 4-2-La constitution de la carte ARDUINO MEGA.....   | 38 |
| 4-2-1La partie matérielle.....                      | 38 |
| a. Alimentation.....                                | 38 |
| b. Le microcontrôleur ATEMEGA2560.....              | 38 |
| c. Les mémoires.....                                | 39 |
| d. Entrés et sorties .....                          | 39 |
| e. La communication.....                            | 39 |
| 4-2-2-Partie logiciel :.....                        | 40 |
| 4-2-2-1-L'interface.....                            | 40 |
| 4-2-2-2-Les boutons de programmation.....           | 41 |
| 4-2-2-3-Le langage ARDUINO.....                     | 42 |
| 4-2-2-4-La structure du code.....                   | 44 |
| 4-2-2-5-Injection du programme.....                 | 45 |
| 5-Conclusion.....                                   | 48 |

### **Chapitre 04 : Principe de fonctionnement du système d'irrigation**

|  |    |
|--|----|
| 1-Introduction.....  | 50 |
| 2-Description de schéma de principe de fonctionnement du système ..... | 50 |
| 3-Les composants utilisés :.....                                       | 51 |

|   |    |
|---|----|
| 3-1-28BYJ-48 - Moteur pas à pas 5V.....                                       | 52 |
| 3-1-1-Quelques détails typiques du moteur pas à pas.....                      | 54 |
| 3-2-Plaque d'essai ELEGOO 830 .....   | 54 |
| 3-3-Driver de pilote de moteur pas à pas ULN2003 à 4 phases :.....            | 54 |
| 3-3-1-Connexion du moteur pas à pas :.....                                    | 55 |
| 3-3-2-LED d'état d'étape :.....   | 56 |
| 3-3-3-Fourniture.....   | 56 |
| 3-3-4-Connexion du microcontrôleur.....                                       | 56 |
| 3-3-5-Opération.....  | 56 |
| 3-3-6-La connexion physique d'un microcontrôleur à la carte de commande ..... | 56 |
| 3-3-7-Connexions typiques.....  | 57 |
| 3-3-8-Code de couleurs :.....   | 57 |
| 4-Conclusion :.....   | 57 |

## **Chapitre 5 : Réalisation pratique du système d'irrigation**

|   |           |
|---|-----------|
| 1-Opportunité.....  | 59        |
| 2-La réalisation du système d'irrigation.....               | 59        |
| 2-1-Montage du système :.....                               | 59        |
| 2-2-Câblage et réalisation du circuit de puissance.....     | 61        |
| 2-2-1-Alimentation .....                                    | 61        |
| 2-2-2-Branchement des pins de commande.....                 | 63        |
| 3-Structure du programme de commande.....                   | 65        |
| 3-1-Au niveau de la partie déclarative.....                 | 65        |
| 3-2Au niveau de la fonction d'initiation setup.....         | 66        |
| 3-3Au niveau de la boucle principe, la fonction loop :..... | 66        |
| 4-Conclusion .....  | 67        |
| <b>Annexe .....</b>   | <b>69</b> |
| <b>Conclusion générale .....</b>                            | <b>78</b> |
| <b>Les problèmes rencontrés .....</b>                       | <b>80</b> |
| <b>Résumé.....</b>  | <b>82</b> |
| <b>La Bibliographie.....</b>                                | <b>84</b> |

## Liste des tableaux

- Tableau 01 : Les modes de commande du moteur pas à pas unipolaire
- Tableau 02 : Le manuel référence de chaque fonction dans IDE
- Tableau 03 : La connexion des branches entre le driver et l'ARDUINO
- Tableau 04 : Les couleurs des fils de bobines et la branche driver correspondante
- Tableau 05 : Les étapes de montage du système d'irrigation

## Liste des figures

- Figure 1: l'Apollo Guidance Computer et son écran d'utilisation
- Figure 2 : ARCHITECTURE D'UN SYSTEME EMBARQUE
- Figure 3: Composants matériels d'un système embarqué
- Figure 4: Premier microprocesseur Intel 4004
- Figure 5: Structure des systèmes automatisés
- Figure 6: automate programmable industriel
- Figure 7: La structure interne d'un API
- Figure 8: symbole du moteur pas à pas
- Figure 9: Moteur à réluctance variable
- Figure 10: fonctionnement d'un moteur pas à pas à reluctance variable
- Figure 11: moteur pas à pas hybride
- Figure 12: moteur pas à pas à aimant permanent
- Figure 13: moteur pas à pas unipolaire
- Figure 14: moteur pas à pas bipolaire
- Figure 15: mode de fonctionnement monophasé ( pas entier) moteur unipolaire
- Figure 16: mode de fonctionnement biphasé(pas entier)moteur unipolaire
- Figure 17: mode de fonctionnement en demi pas moteur unipolaire
- Figure 18 : moteur bipolaire monophasé (pas entier)
- Figure 19 : moteur bipolaire biphasé (pas entier)
- Figure 20 moteur bipolaire demi-pas
- Figure 21 : l'alimentation du moteur pas à pas unipolaire
- Figure 22 : l'alimentation en pont H du moteur pas à pas bipolaire
- Figure 23: l'alimentation symétrique du moteur pas à pas bipolaire
- Figure 24 Figure : Description de la carte ARDUINO MEGA 2560
- Figure 25 Figure : présentation l'interface ARDUINO
- Figure 26 Figure : présentation du menu fichier
- Figure 27 boutons de programmation
- Figure 28: présentation du code
- Figure 29: fenêtre inferieur d'ide
- Figure 30: la sélection du type de la carte
- Figure 31: le numéro du port USB
- Figure 32: compilation et téléchargement du code
- Figure 33: schéma de principe d'irrigation
- Figure 34: la composante liste pour la réalisation

Figure 35: moteur unipolaire à 5 fils

Figure 36: architecture interne du moteur pas à pas

Figure 37: plaque d'essai ELEGOO 830

Figure 38: driver ULN 2003A

Figure 39: l'alimentation des deux drivers

Figure 40: les sortie 5v et GND de la carte ARDUINO vers la plaque d'essai et les deux drivers

Figure 41: l'alimentation du relais de dc+ et dc- vers 5v et GND de l'ARDUINO

Figure 42: l'alimentation 220 v alternative de l'électrovanne

Figure 43 : branchements des pins de commandes du moteur

Figure 44: branchement du relais avec le pins 12 de l'ARDUINO

Figure 45: Réalisation du circuit de puissance

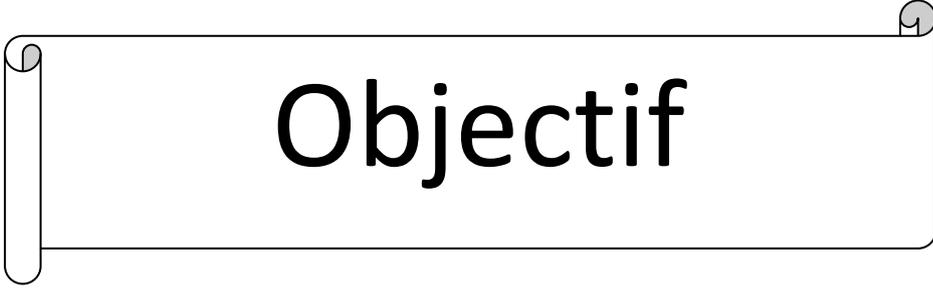
Figure 46: Présentation d'ATEMEGA 2560

Figure 47: Logo ATEMEGA 2560

Figure 48: Schématique ATEMEGA 2560

Figure 49: Schéma réel de la commande de deux moteur pas à pas pour l'irrigation





**Objectif**

## Objectif

Notre objectif Dans ce projet est :

- D'utiliser un interface ARDUINO , d'une programmation facile permettre à un ordinateur de communiquer avec une **carte** électronique pour commander différents composants ( capteur, moteur ) .  
On peut recourir à une quantité de connecteurs disponibles sur la carte qui nous facilite le câblage de ces dernier .afin qu'il puisse interagir avec eux : (déplacement, mouvement , lecture, écriture) , etc.
- Utiliser la robotique pour articuler les objets. pour se faire il est très courant d'utiliser des moteurs électriques pas à pas notamment lorsqu'on veut obtenir une bonne précision de mouvement en boucle ouverte.
- L'agriculture est l'avenir c'est pourquoi on a choisi de mettre en valeur la commande de deux moteurs pas a pas a l'aide d'un ARDUINO pour une irrigation autonome .



# Introduction générale

# Introduction générale

---

## Introduction générale :

Grâce à la technologie et les Systèmes Embarqués, nous vivons désormais dans un monde de plus en plus connecté » Aujourd'hui, les machines savent faire preuve d'intelligence et d'une capacité d'apprentissage.

Une machine intelligente (smart machine) réalise des fonctions mécaniques telles qu'un déplacement ou un positionnement. Pour cela, il a besoin d'un actionneur qui va réaliser à la fois une transmission d'information et une conversion d'énergie. Plusieurs technologies existent : on rencontre des actionneurs électriques, pneumatiques et hydrauliques.

Tous les actionneurs tels que les moteurs électriques, quelle que soit leur taille ou leur fonction, nécessitent un mécanisme de commande.

Les commandes de moteur peuvent remplir de nombreuses fonctions telles que le démarrage ou l'arrêt automatique ou manuel d'un moteur électrique, le réglage de sa vitesse, de sa direction.

Les moteurs pas à pas sont utilisés dans de nombreuses applications, par rapport aux autres moteurs, quoiqu'ils soient moins connus. Or, même si leur commande est beaucoup plus complexe, ils offrent avec un faible coût, un contrôle de vitesse ou de position en boucle ouverte, typiquement dans les systèmes de positionnement. Ils sont ainsi largement utilisés dans le domaine de la cybernétique. Pour commander on a besoin d'un circuit de commande et d'alimentation.

Dans cette optique, on nous a confié un travail dont l'intitulé est « commande de deux moteurs pas à pas » il a pour but de mettre en œuvre une application de commande de positionnement et le sens de deux moteurs pas à pas.

Notre mémoire est structuré en cinq (05) chapitres permettront de présenter les différents aspects de notre projet. Le premier chapitre nommé « les systèmes embarqués & l'évolution des automates programmable » est constitué de deux parties : dans la première partie nous avons parlé généralement sur les systèmes embarqués, la deuxième partie présente l'architecture des automates programmables et leurs évolutions pendant les siècles. Le deuxième chapitre intitulé : « le fonctionnement et la commande des moteurs pas à pas » ou nous avons étudié les différents types du moteur pas à pas et la commande de sens et de positionnement des types aimant permanents ainsi que son alimentation. Le troisième chapitre sous-titre : « description de la carte ARDUINO » Dans ce chapitre nous sommes parlé des Caractéristiques techniques de la carte ARDUINO MEGA et le logiciel de développement intégré (IDE). Par contre, le quatrième contient le schéma de fonctionnement et la liste des matériel utilisé pour la réalisation de notre projet. Le cinquième chapitre représente le fruit de notre travail, la réalisation de la théorie et les résultats obtenus lors de la commande. Nous clôturons ce mémoire par une conclusion générale.



# Chapitre 01

## ***Chapitre 01 : Systèmes embarqués et l'évolution des automates programmables industriels***

### **1-Introduction :**

1- Les systèmes électroniques sont de plus en plus présents dans la vie courante. Les ordinateurs et micro-ordinateurs sont des systèmes électroniques bien connus. Mais l'électronique se trouve maintenant embarquée dans de très nombreux objets usuels : les téléphones, les agendas électroniques, les voitures. Ce sont ces systèmes électroniques enfouis dans les objets usuels qui sont appelés systèmes embarqués.

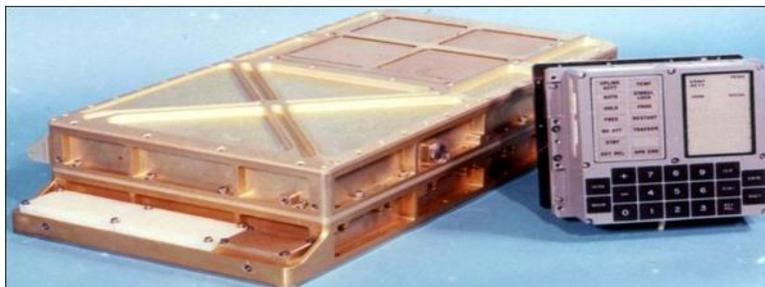
### **2. PRESENTATION DES SYSTEMES EMBARQUES**

#### **2.1. Définition :**

Un système embarqué regroupe à la fois la partie software (logicielle) et la partie hardware (matériaux) étroitement liées afin de produire les résultats escomptés et de manière à pouvoir répondre spécifiquement aux besoins de leur environnement respectif.

#### **2.2. Historique :**

Le tout premier système embarqué c'est-à-dire les premiers ordinateurs autonomes ont été conçu en 1962 dans le but de guider le missile nucléaire MINUTEMAN , contenait plusieurs circuits intégrés afin de réaliser les tâches qui lui étaient dédiées comme le guidage du missile selon le positionnement du missile.



**Figure01 :** l'Apollo Guidance Computer et son écran d'utilisation

#### **2.3. Caractéristiques d'un système embarqué :**

Ces caractéristiques sont uniques dans chaque système embarqué

- Faible coût:
  - Solution optimale entre le prix et la performance
  - À la portée de toute personne
  - Par conséquent, les ressources utilisées sont minimales

- Un système embarqué n'a que peu de mémoire
- Faible consommation:
  - Utilisation d'une batterie d'emmagasinations d'énergie
  - Gérer la consommation pour rester autonome le plus possible
  - Pas de consommation excessive, moins de prix et des batteries de faible capacités
- Faible encombrement et faible poids:
  - Minimiser la taille et le poids pour un système embarqué.
  - Les composants électroniques (analogique et/ou numérique) doivent cohabiter sur une faible surface.
- Fonctionnement en temps réel:

le temps réel est l'une des caractéristiques primordiales au niveau de la performance mais surtout au niveau de la sécurité. Devant un ordinateur classique, quelques minutes de latence ne pourront affecter que l'humeur de l'utilisateur, sur un système embarqué par exemple d'automobile seuls quelques secondes de latence suffisent à provoquer un accident avec des conséquences terribles

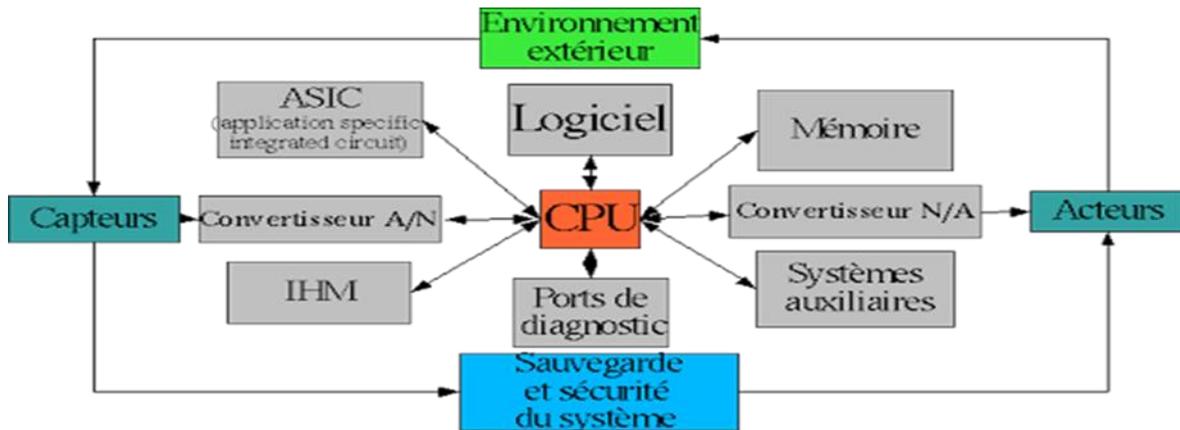
Un système en temps réel est un système qui doit, non seulement, produire un résultat juste mais dans une durée limitée, sans quoi ce résultat deviendrait erroné.
- L'autonomie : Les systèmes embarqués doivent en général être autonomes, c'est-à-dire remplir leur mission pendant de longues périodes sans intervention humaine. Cette autonomie CHAPITRE 01 7 est nécessaire lorsque L'intervention humaine est impossible, mais aussi lorsque la réaction humaine est trop lente ou insuffisamment fiable
- Environnement:
  - Un système embarqué est soumis à des nombreux contraintes d'environnement
  - Il doit s'adapter et fonctionner avec eux.
  - Exemple: La température , L'humidité , Les vibrations , Les chocs ,Les variations d'alimentation, les interférences RF, les radiations... etc.

### **2.4. Architecture d'un système embarqué :**

Quelle que soit la nature et la complexité du système, un système embarqué est composé d'

- Une partie matérielle.
- Une partie logicielle.

L'architecture d'un système embarqué se définit par le schéma ci-dessous (FIGURE 3).



**Figure02 : ARCHITECTURE D'UN SYSTEME EMBARQUE**

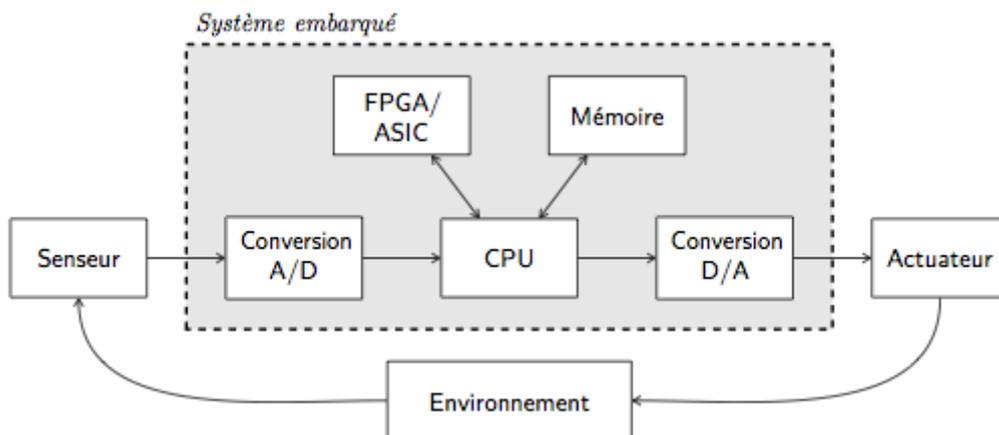
## 2.4.1. Composants matériels d'un système embarqué :

### A. Introduction :

L'architecture matérielle d'un système embarqué liste les composants le constituant ainsi que la manière avec laquelle ils sont agencés entre eux. Il existe plusieurs types de représentations possibles pour décrire une architecture matérielle, la plus courante et pratique étant un diagramme blocs. Ce type de diagramme comporte généralement :

Des blocs qui représentent les différents composants, unités ou sous-systèmes composant le système embarqué, selon le niveau d'abstraction désiré pour la description de l'architecture matérielle.

Des liens entre blocs qui représentent un contrôle, une commande, un échange d'information, avec éventuellement le type de communication utilisé.



**Figure 03: Composants matériels d'un système embarqué**

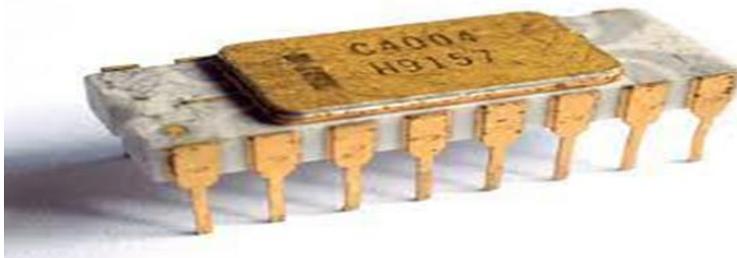
## c. Unité de calcul CPU:

Le calculateur correspond à un boîtier contenant des broches électriques dotées de nombreux ports d'entrées et de sorties afin de permettre à la gestion des instruments de bord.

Une carte programmable composée de circuits imprimés contient tout le traitement informatique du système, principalement codé en langage C++ ou Java.

Dans le cas d'un calculateur moteur, son but précis sera d'assurer les fonctions de pilotage d'un moteur en ajustant en temps réel les besoins du moteur. En recevant des signaux électriques de la part des capteurs, le calculateur peut traiter ces informations pour les transformer en actions par l'intermédiaire d'actionneurs.

- Ceci est le premier microprocesseur ayant la capacité à pouvoir incorporer tous les éléments d'un ordinateur (mémoire, contrôle d'accès, unité de calcul) dans un même circuit intégré.



**FIGURE 04 :** Premier microprocesseur Intel 4004

## c. mémoire :

les systèmes embarqués utilisent principalement trois formes de mémoire: RAM , ROM et EEPROM .

- ROM (READ ONLY MEMORY) :mémoire en lecture seule contient des données stockées de façon permanente et aussi le programme exécuté par le système embarqué.
- RAM :(mémoire vive, ou mémoire à accès aléatoire) Le microprocesseur utilise la mémoire vive, comme un "SCRATCHPAD" temporaire sur lequel pour tenir les

variables de calcul du processeur et d'autres types de données intermédiaires est utilisée comme mémoire tampon d'entrée/sortie pour le traitement de son ou d'images, par exemple.

- EEPROM (ELECTRICALLY ERASABLE Programmable ROM) :est une mémoire permanente, mais dont le contenu peut être modifié en flashant la mémoire à l'aide d'un procédé électrique. On l'utilise notamment comme mémoire cache, pour stocker une copie des instructions et données chargées à l'avance depuis une mémoire externe ou des résultats temporaires de calculs pour du traitement rapide.

### **d. Alimentation :**

Afin de pouvoir fonctionner, un système embarqué a besoin d'être alimenté en énergie électrique. Il doit dispenser d'une source d'énergie qui peut être :

- Le courant du secteur : par exemple machine à laver, récepteur, poste de télévision, système d'alarme...
- Batteries et piles pour les systèmes mobiles : téléphone portable, robot ...
- Générateur de courant électrique : alternateur de voiture pour le cas du GPS, de l'ABS, de l'ESP
- Energie renouvelable : énergie solaire

### **e. Actionneurs**

Dans une machine ou un système de commande à distance, semi automatique ou automatique, un actionneur est l'organe de la partie opérative qui, recevant un ordre de la partie commande via un éventuel pré-actionneur, convertit l'énergie qui lui est fournie en un travail utile à l'exécution de tâches, éventuellement programmées, d'un système automatisé

En d'autres termes, Lorsque le traitement est réalisé par le calculateur, un signal électrique est transmis aux actionneurs permettant une action physique .

Ces actionneurs (ou actuateurs) transforment le signal électrique reçu en énergie mécanique.

### **2.4.2. Partie logicielle :**

Pour le développement informatique des différents programmes présents, ARDUINO qui a été utilisé.

ARDUINO possède son propre langage de programmation applicable seulement sur les cartes de type ARDUINO. Ce langage est en réalité fortement inspiré du langage C++ et du C.

Le logiciel de développement comprend un compilateur permettant de traduire le langage de programmation ARDUINO en langage machine.

# Les systèmes embarqué et l'évolution des automates programmable

---

toutes les instructions programmées sont envoyées au microprocesseur de la carte par l'intermédiaire du câble USB qui y est connecté.

Lors de l'alimentation de la carte, le code enregistré dans le microcontrôleur via sa mémoire flash est automatiquement exécuté et ce en continu. C'est la structure de l'architecture ARDUINO qui impose l'exécution des instructions en boucle.

## 2.5. Classification des systèmes embarqués :

Nous pouvons classier les systèmes embarqués dans trois types comme suit :

- ✓ Systèmes embarqués à petite échelle: Ces systèmes sont conçus avec un microcontrôleur simple de 8 ou 16 bits, ils ont peu de complexité de matériel et de logiciel, Ils peuvent même être à piles.
- ✓ Systèmes embarqués à échelle moyenne : Ces systèmes sont habituellement conçus avec des microcontrôleurs de 16 ou 32 de bit ou des DSPS simples ou RISCs. Ceux-ci ont des complexités de matériel et de logiciel avérées.
- ✓ Systèmes embarqués sophistiqués: Les systèmes embarqués sophistiqués ont d'énormes complexités de matériel et de logiciel et peuvent avoir besoin de processeurs extensibles ou de processeurs configurables et des réseaux logiques programmables (PLA).

## 2.6. Domaines d'application des systèmes embarqués :

Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que :

- le transport (avionique, espace, automobile, ferroviaire).
- les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables).
- la distribution d'énergie.
- l'automatisation.

## 3.L'évolution des automate programmables industriels :

### 3.1. Définitions d'un système automatisé :

Un système de production est dit automatisé lorsqu'il peut gérer de manière autonome un cycle de travail préétabli qui se décompose en séquences et/ou en étapes

Une autre définition : « L'automatisation est considérée comme l'étape d'un progrès technique où apparaissent des dispositifs techniques susceptibles de seconder l'homme, non seulement dans ses efforts musculaires, mais également dans son travail intellectuel de surveillance et de contrôle. »

On est passé d'un système dit manuel, à un système mécanisé, puis au système automatisé. Dans l'industrie, les automatismes sont devenus indispensables : ils permettent d'effectuer quotidiennement les tâches les plus ingrates, répétitives et, dangereuses. Parfois, ces automatismes sont d'une telle rapidité et d'une telle précision, qu'ils réalisent des actions impossibles pour un être humain. L'automatisme est donc synonyme de productivité et de sécurité.

### 3.2. Structure des systèmes automatisés :

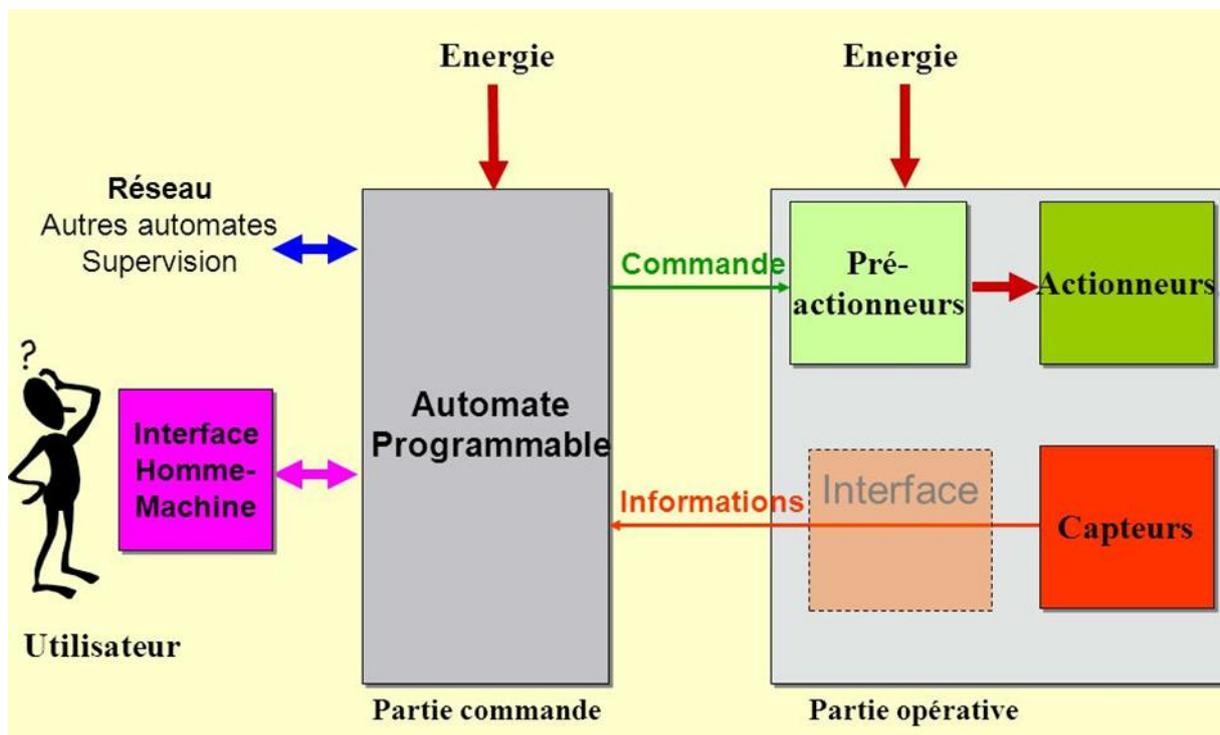


Figure 05: Structure des systèmes automatisés

## 3.3. la composition d'un système automatisé :

Tout système automatisé est composé de

### 3.3.1. Une partie opérative (P.O) :

c'est la partie mécanique et visible du système qui effectue les opérations. Elle comporte les éléments du procédé, c'est à dire :

- des pré-actionneurs (distributeurs, contacteurs) qui reçoivent des ordres de la partie commande.
- des actionneurs (vérins, moteurs, vannes) qui ont pour rôle d'exécuter ces ordres. Ils transforment à partir d'une source d'énergie extérieure : pneumatique (air comprimé), hydraulique (huile sous pression) ou électrique en énergie mécanique
- Les capteurs qui créent, à partir de grandeurs physiques de natures divers (déplacement, température, pression...etc.), des informations utilisables par la partie commande.

### 3.3.2. Une partie commande (P.C.) :

La partie commande d'un système automatisé est destinée à traiter les informations à fin de répondre aux fonctionnalités suivantes :

Cordonner les actions de la partie opérative.

Surveiller le fonctionnement de la P.O

### 3.3.3. La partie dialogue :

La partie dialogue se compose de deux ensembles :

- Les visualisations et avertisseurs qui transforment les informations fournies par l'automate en informations perceptibles par l'homme (informations optiques ou sonores) .
- Les capteurs qui transforment les informations fournies par l'homme (action manuelle sur un bouton poussoir, par exemple) en informations exploitables par l'automate.

## 3.4. Les avantages et les inconvénients des systèmes automatisés:

### a. Les avantages :

- La capacité de production accélérée
- Éliminer les tâches répétitives
- Simplifier le travail de l'humain
- Augmenter la sécurité (responsabilité)
- La création de postes d'automaticiens.

## b. Les inconvénients :

- Le coût élevé du matériels, principalement avec les systèmes hydrauliques .
- La maintenance doit être structurée.

## 3.5. Les APIs

### 3.5.1 Historique :

Les automates programmables industriels ou A.P.I. comme on les appelle le plus souvent ou encore Programmable LOGIC CONTROLLER (PLC en anglais), sont apparut aux Etats-Unis vers 1969 pour les besoins de l'industrie automobile américaine en plein essor . le but rechercher était de remplacer les armoires à relais , utilisées dans la commande des chaines de fabrications pars des équipement moins onéreux,aussi bien au point de vue du cout d'acquisition que du cout de la maintenance, et flexible c'est-à-dire facile à modifier , à utiliser et à entretenir.

### 3.5.2. Définitions de l'Automate Programmable Industriel :

Un Automate Programmable Industriel est une machine électronique spécialisé pour les applications industrielles, programmable par un utilisateur automaticien non informaticien et destinée à piloter en ambiance industrielle et en temps réel des procédés automatiques.



Figure06: automate programmable industriel

### 3.5.3. Domaines d'emploi des automates :

On utilise les API dans tous les secteurs industriels pour la commande des machines (convoyage, emballage...) ou des chaînes de production (automobile, agroalimentaire ...) ou il peut également assurer des fonctions de régulation de processus (métallurgie, chimie ...). Il est de plus en plus utilisé dans le domaine du bâtiment (tertiaire et industriel) pour le contrôle du chauffage, de l'éclairage, de la sécurité ou des alarmes.

### 3.5.4. Nature des informations traitées par l'automate :

Les informations traitées par un API peuvent être de type :

- Tout ou rien (T.O.R.) ou logique : l'information ne peut prendre que deux états (0 ou 1 ...). C'est le type d'information délivrée par un détecteur, un bouton poussoir ...
- Analogique : l'information est continue et peut prendre une valeur comprise dans une plage bien déterminée. C'est le type d'information délivrée par un capteur (pression, température ...)
- Numérique : l'information est contenue dans des mots codés sous forme binaire. C'est le type d'information délivrée par un ordinateur ou un module intelligent.

### 3.5.5. L'architecture d'un automate programmable industrielle :

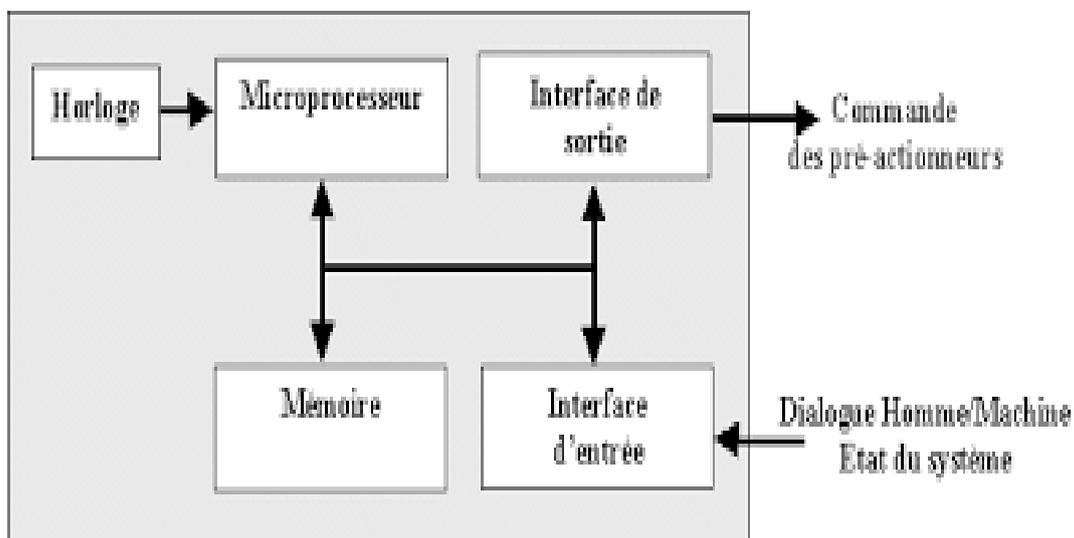


Figure 07: La structure interne d'un API

L'automate programmable reçoit les informations relatives à l'état du système et puis commande les pré-actionneurs suivant le programme inscrit dans sa mémoire.

Les API comportent quatre parties principales :

- Une mémoire ;
- Un processeur ;
- Des interfaces d'Entrées/Sorties ;
- Une alimentation (240 VAC → 24 Vcc).

Ces quatre parties sont reliées entre elles par des bus (ensemble câblé autorisant le passage de l'information entre ces 4 secteurs de l'API). Ces quatre parties réunies forment un ensemble compact appelé automate.

### 3.6. L'évolution de l'automate programmable industriel pendant les siècles :

On sait maintenant que la deuxième partie du XXe siècle a passée à l'histoire comme étant l'ère de l'automatique.

Et c'est à travers l'automatique, d'abord en 1968-69 aux Etats Unis, que les premiers automates industriels ou « contrôleurs programmables » firent leur apparition.

Leurs premières applications furent d'abord le remplacement des horloges de contrôle du temps des employés ; par la suite, leurs multiples utilisations industrielles, en particulier sur les lignes de production des usines, deviennent indispensables non seulement au point de vue contrôle, mais aussi du côté économique pour l'espace et l'entretien. C'est alors que de nombreux systèmes à relais durent céder leur place.

Les premiers automates programmables n'effectuaient que la commutation ON/OFF avec la possibilité de temporisation, comme les relais. Leurs applications étaient limitées seulement aux procédés répétitifs ainsi qu'à certaines machines. Par contre, leurs avantages consistaient dans une installation plus facile, la visualisation des étapes; ils possédaient des indicateurs diagnostiques permettant la localisation des pannes. C'était déjà mieux que les relais, en plus de pouvoir être reprogrammé advenant un changement de fonction ou de procédé.

De 1970 à 1974, la technologie des microprocesseurs ajoutèrent une plus grande flexibilité et une « intelligence » à l'automate programmable. Les capacités d'interface avec l'utilisateur s'améliorent. L'automate programmable peut maintenant exécuter les opérations arithmétiques en plus des opérations logiques; il manipule les données et les adresses ; il effectue la communication avec d'autres automates ou ordinateurs, donnant ainsi une nouvelle dimension aux applications de l'automate programmable.

La console de programmation s'allie avec un moniteur permettant la programmation avec des symboles familiers de relais ce qui facilite beaucoup la compréhension et le dépannage car la logique peut être vue dans la même forme que les dessins à relais.

● Les automates programmables utilisent une mémoire non-volatile (RAM+Pile, EEPROM ou EAPROM par exemple) pour emmagasiner les instructions. Ces derniers accompliront des fonctions logiques, arithmétiques, de temporisation, de comptage et de manipulation des données. En plus, les fonctions de contrôle PID et d'autres fonctions complexes comme le contrôle numérique de processus sont présentes. Puisque les automates programmables ont été conçus pour accomplir des opérations semblables à celles des relais, la programmation

## Les systèmes embarqué et l'évolution des automates programmable

---

est basée généralement sur la nomenclature des diagrammes en échelle (ou schéma à relais). Des langages de haut niveau ont été aussi implantés sur certains automates afin de produire une plus grande flexibilité de programmation.

### **4. conclusion :**

Bien que les systèmes embarqués aient connu une évolution infernale, ils demeurent inadéquats pour n'importe quelle application. On aimerait bien y avoir un qui soit adapté et satisfasse nos besoin et adorable par notre bourse. Heureusement les chercheurs italiens ont pu enlever cette épine et ont conçu un système embarqué adapté s'appelle ARDUINO. Dans le chapitre 3 on présentera ses caractéristiques en détails.



# Chapitre 02

## **Chapitre 02 : Le fonctionnement et la commande des moteurs pas à pas**

### **1-Introduction :**

Les nombreuses applications de la commande numérique ont remis en faveur le moteur pas à pas, appelé aussi moteur à impulsions qui était utilisé depuis déjà longtemps. Les moteurs pas à pas ont été créés en 1930. Leurs développements ont commencé au début des années soixante grâce à l'apparition des microprocesseurs, microcontrôleurs et aux énormes progrès de l'électronique de puissances à transistor. Et encore, pendant des années, l'apparition essentielle fût celle de la commande numérique des machines outil ayant besoin d'un tel type de moteur. Puis avec le développement de la recherche spatiale, de nouveaux moteurs pas à pas firent leur apparition (dimensions réduites et pas angulaires plus grand) et furent largement utilisés grâce à leurs avantages importants de fiabilité et de simplicité de commande qu'ils offraient sur les autres moteurs.

C'est un convertisseur électromécanique qui assure la transformation d'un signal électrique impulsionnel en un déplacement mécanique. Sa structure de base se présente sous la forme de deux pièces séparées mécaniquement, le Stator et le Rotor. L'interaction électromagnétique entre ces deux parties assure la rotation.

Les moteurs pas à pas sont souvent utilisés par les concepteurs qui veulent minimiser le temps de développement du système dans lequel ils ont besoin d'entraînements asservis en position.

### **2. La présentation d'un actionneur pas à pas :**

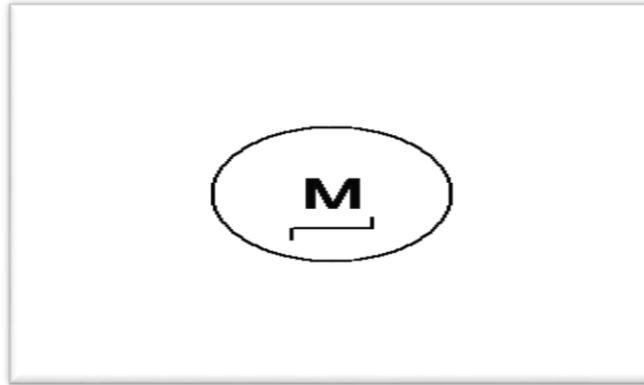
Les moteurs pas à pas permettent de convertir directement un signal électrique numérique en un positionnement angulaire. Chaque succession d'impulsions électriques envoyée par le système de commande au module de puissance se traduit par la rotation d'un ou de plusieurs pas (des fractions de tours) du moteur.

L'excitation du moteur se fait via les bobines du stator.

- La valeur du pas est définie par :
  - un angle par pas.
  - le nombre de pas par tour ( $360^\circ$ ).
- Le nombre de pas dépend du nombre de phases (groupe de bobine), du nombre de pôles du rotor et du stator.

Le moteur pas à pas peut remplir deux fonctions :

- Conversion d'énergie électrique en énergie mécanique (c'est le moteur classique).
- Conversion de l'information numérique en un positionnement angulaire ou linéaire.



**Figure 08:** Symbole du moteur pas à pas

- $\theta$  est appelé « angle de pas » ou simplement « pas », son unité est le degré et s'obtient par la formule suivante :

$$\theta = \frac{360}{Np} [^\circ]$$

Où **NP** est le nombre de pas par tour [ **PAS / TOUR** ]

- La vitesse de rotation du moteur dépend de la fréquence  $f$  des impulsions et s'obtient par la formule suivante :

$$n = \frac{f}{NP} [ \text{TOURS} / \text{S} ]$$

$$n = \frac{f \times 60}{NP} [ \text{tours} / \text{min} ]$$

### **3- Applications industrielles :**

Les moteurs pas à pas sont très utilisés dans toutes les applications mécaniques ou l'on doit contrôler simplement la position ou la vitesse d'un système.

Ces moteurs sont par exemple utilisés en :

- Robotique (positionnement des axes, vitesse variable du robot).
- Informatique (déplacement d'une tête de lecture d'une lecture, déplacement de la tête d'impression, avance du papier pour les imprimantes).

- astronomie(positionnement du télescope pour suivre un astre) et dans la réception TV par satellites (positionnement d'une parabole).

### **4- les avantages et les inconvénient du moteur pas à pas :**

Parmi les **avantages du moteur pas à pas**, on notera la précision que l'on peut obtenir quand il tourne et la facilité de déterminer le nombre de tours ou pas effectués grâce à sa **commande par impulsion**. Il est également aisé de **varier la vitesse du moteur** en modifiant la fréquence d'alimentation des bobines.

Par contre, parmi ses **inconvénients** on peut noter le fait que sa **vitesse maximale** n'est pas élevée ou encore que son principe de commande est plus compliqué qu'un **moteur à courant continu**.

Risque de perte de pas (et donc de dysfonctionnement du contrôle en position) en cas de perturbation anormale notamment de la charge ou de fréquence excessive des impulsions.

### **5. les types de moteur pas à pas :**

Diverses technologies de moteur pas-à-pas sont disponibles:

- Moteur à réluctance variable.
- Moteur à aimant permanent.
- Moteur hybrides.
- Dans chacun de ces types de **moteurs pas à pas** le principe de global de fonctionnement est le même, pour faire tourner le moteur on doit créer un **champ magnétique** dans les **bobines du stator**, ainsi on peut faire tourner le rotor en déplaçant le **champ magnétique** en alternant les bobines alimentées.

#### **5-1- Moteur à réluctance variable :**

##### **5-1-1-Constitution :**

- Dans un moteur pas à pas à réluctance variable Le stator présente un certain nombre de dents ayant un bobinage(les bobines d'excitation) .
- Le rotor est en acier possède un nombre différent de dents, mais sans bobinage.

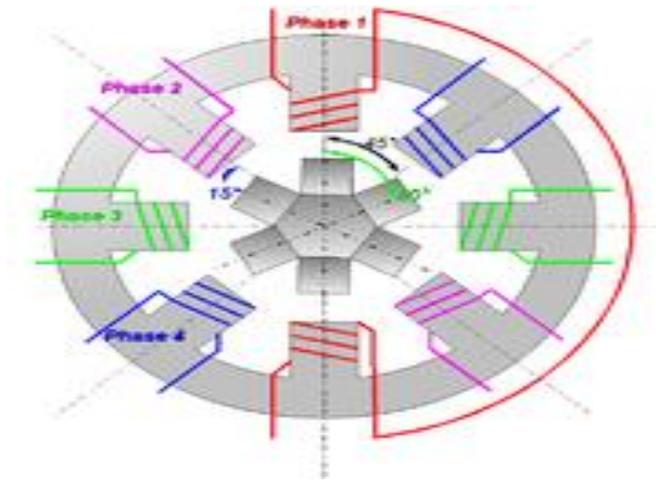


Figure 09: Moteur à réluctance variable

### 5-1-2-le fonctionnement :

Le positionnement se fait suivant le principe de la réluctance minimale. L'axe des deux dents du rotor s'aligne avec l'axe des deux bobines alimentées pour minimiser la réluctance du trajet offert aux lignes de champ. A chaque commutation, ce sont les dents les plus proches de plots alimentés qui s'alignent en face d'eux. On peut avoir les 3 modes de fonctionnement : mode monophasé, mode biphasé et mode demi pas.

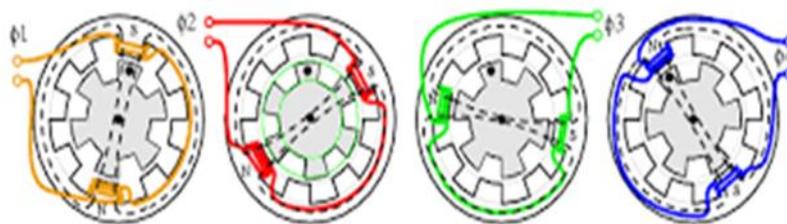


Figure 10: fonctionnement d'un moteur pas à pas à reluctance variable

### 5-1-3-Les avantages Les inconvénient :

#### a- les avantages

Nombre de pas par tour important (bonne résolution).

Construction assez facile.

Sens de rotation lié à l'ordre d'alimentation des bobines.

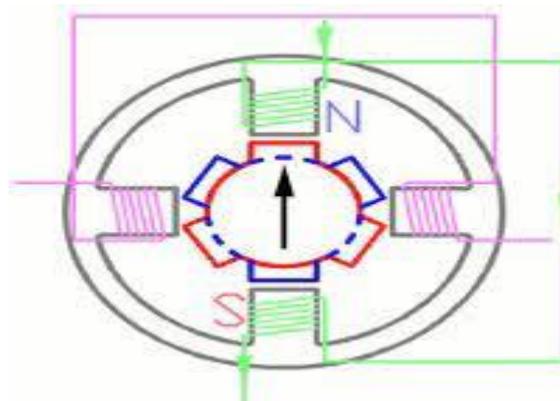
### b- Les inconvénient :

Ce moteur présente une simplicité de construction mais du fait de son faible couple moteur, il est le plus souvent remplacé par des moteurs pas à pas à aimant permanent ou hybrides.

### 5.2. Moteur hybrides :

#### 5.2.1. Constitution :

Combinaison du moteur à réluctance variable et du moteur à aimant permanent.



**Figure 1:** moteur pas à pas hybride

#### 5.2.2. Fonctionnement :

Son fonctionnement est sensiblement identique à celui du moteur à aimant permanent.

Des positions successives du rotor après l'alimentation des bobines du stator.

Le nombre de dents au rotor est différent de celui du stator. Quand on alimente une paire de bobines, le rotor place les dents Nord et Sud de telle façon que le flux traversant le rotor soit maximale.

#### 5.2.3.les avantages :

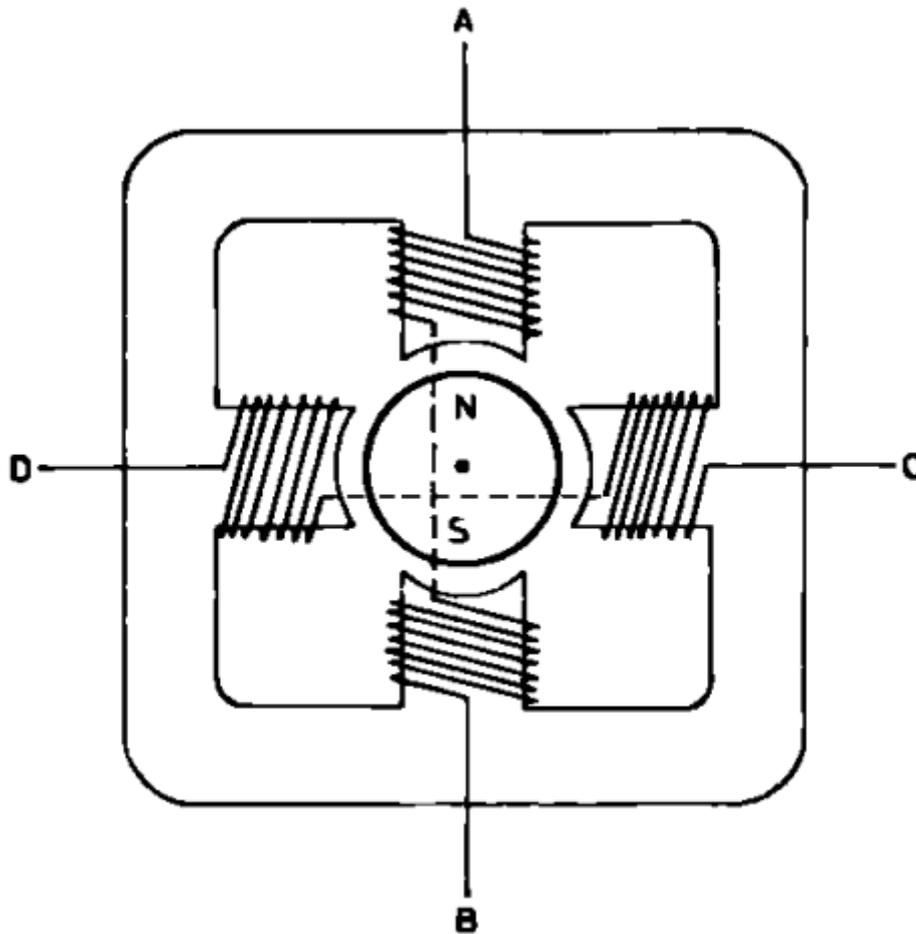
- Couple moteur élevé dû à l'aimantation du rotor
- Sens de rotation lié à l'ordre d'alimentation des bobines et au sens du courant

Dans la suite de ce chapitre on s'intéressera uniquementle plus simple des trois en l'occurrence le moteur pas à pas à aimant permanent [ et surtout l'unipolaire] . Sur ce moteur, le rotor (la partie qui tourne sur laquelle est placée l'axe du moteur)

### 5.3. Moteur à aimant permanent :

#### 5.3.1. Constitution :

- Le stator présente un certain nombre de dents ayant un bobinage.
- Le rotor est un aimant qui s'alignera avec les pôles du stator qui sont alimentés.

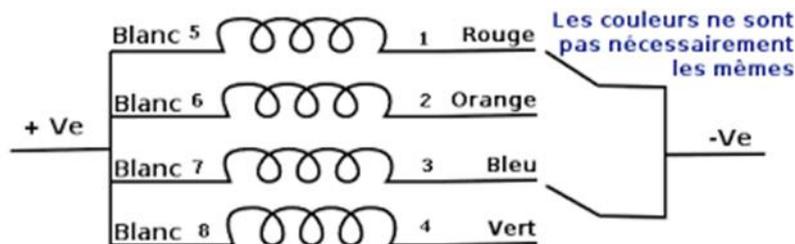


**Figure 2:** moteur pas à pas à aimant permanent

#### 5.3.2. Fonctionnement :

Le moteur à aimant permanent est aussi appelé moteur polarisé. C'est le modelé dont le fonctionnement est le plus simple. Son rotor porte des aimants permanents qui s'alignent suivant son champ magnétique avec celui développé par le stator qui comporte deux paires





**Figure 4:** moteur pas à pas bipolaire

Sur un plan fondamental, ces deux types fonctionnent exactement de la même manière ; des électro-aimants sont activés de façon séquentielle, ce qui induit l'arbre moteur central à tourner. La différence entre les deux types est les niveaux de tension. Un moteur pas à pas unipolaire fonctionne uniquement avec une tension positive, de sorte que les hautes et basses tensions appliquées aux bobines électromagnétiques seraient quelque chose comme 5V et 0 V. Un moteur pas à pas bipolaire a deux polarités, positives et négatives, de sorte que ses hautes et basses tensions seraient quelque chose comme 2,5V et -2,5 V.

### **6- La commande des moteurs pas à pas :**

Le circuit de commande d'un moteur pas à pas se compose de deux parties : le séquenceur, qui commande l'ordre des commutations des courants dans les enroulements, celui-ci peut être un microcontrôleur, un ordinateur ou un circuit intégré spécialisé ; et l'étage de puissance, qui injecte le courant imposé par le séquenceur. L'étage de puissance peut fonctionner en tout ou rien, ou en découpage.

La commande pas à pas est destinée à un contrôle de position, la précision du positionnement est donc un paramètre important. Le nombre de pas par tour d'un système pas à pas correspond à la résolution du positionnement et non à sa précision. Les modes ½ pas ou micro-pas ne font qu'améliorer la résolution mais pas la précision .

#### **6-1-Différents modes de commandes :**

##### **6-1-1-Modes pas entier :**

chacune des phases est alimentée seule par un créneau de courant (ou de tension) sans chevauchement avec la phase voisine. Avec une alimentation bipolaire de chaque phase et compte tenu de la possibilité d'inverser le courant, il y a alors 4 positions stables par période électrique. Ce fonctionnement est appelé par « pas entiers ».

##### **6-1-2- Mode demi pas :**

Si maintenant on combine les 8 possibilités précédemment décrites (une seule phase alimentée, puis deux avec des courants tantôt positifs ou négatifs), on obtient 8 positions

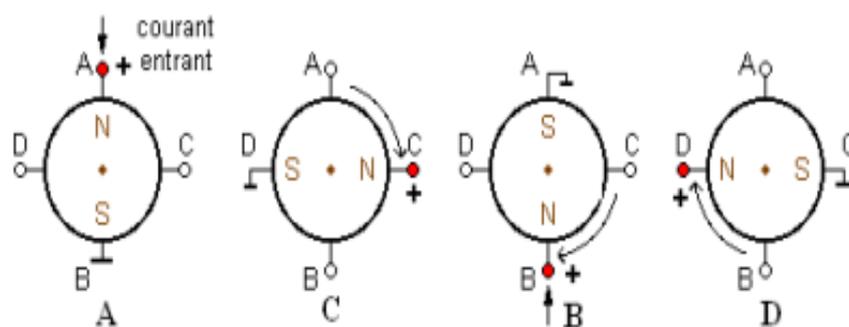
stables par période électrique. Il s'agit du mode  $\frac{1}{2}$  pas, un moteur 48 pas par tour, permet d'obtenir 96  $\frac{1}{2}$  pas par tour. Elles présentent l'avantage de faire comprendre simplement le positionnement du rotor en fonction de l'alimentation des phases.

### 6-2- commande de Moteur unipolaire

#### 6-2-1-mode monophasé et biphasé pas entier :

##### a) Monophasé :

Dans ce mode, les bobines sont alimentées l'une après l'autre dans un sens puis dans l'autre. L'aimant permanent suit le déplacement du champ magnétique créé par ces bobines et s'aligne selon une de ses 4 positions stables (A, B, C, D). Comme le rotor est aimanté, lorsque le moteur n'est pas alimenté le flux magnétique dû à l'aimant permanent va à lui seul créer un couple résiduel ou couple de détente, en se mettant dans l'axe de l'une des bobines.



**Figure 5:** mode de fonctionnement monophasé ( pas entier) moteur unipolaire

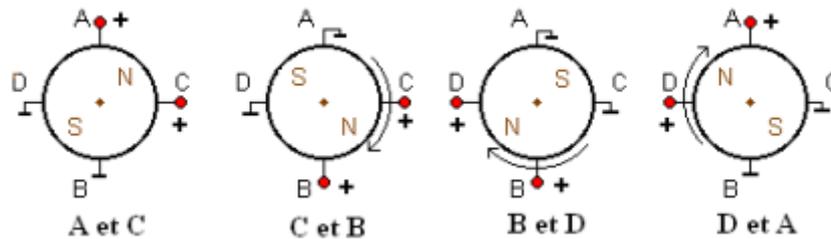
- On alimente successivement chaque demi-enroulement.

La séquence de commande est : A / C / B / D / A . . .

Pour obtenir une rotation dans la direction opposée, il suffit d'inverser les séquences de commande, donc : A / D / B / C / A . . .

##### b) Biphasé :

On alimente les deux bobines en même temps, en faisant varier seulement le sens du courant dans chacune d'entre elles.



**Figure 6:** mode de fonctionnement biphasé(pas entier)moteur unipolaire

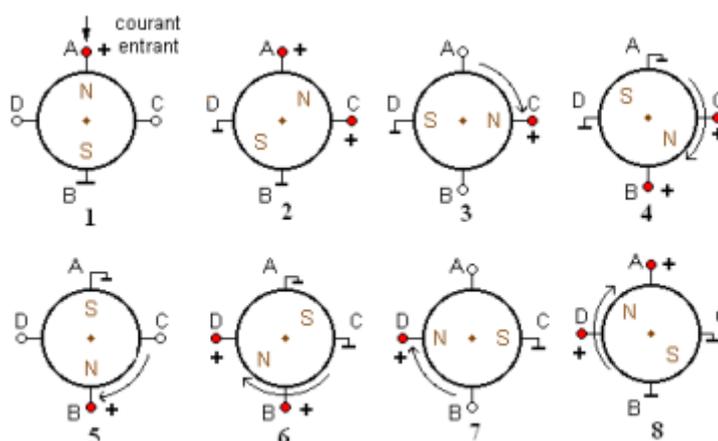
- On alimente successivement deux demi-enroulement.

Voici la séquence à suivre : A et C / C et B / B et D / D et A / A et C ...

et pour l'autre sens : A et C / D et A / B et D / C et B / A et C ...

### 6-2-2-Principe en mode demi – pas :

La combinaison de ces deux derniers modes donne la mode demi pas. Mode monophasé  
Mode biphasé Mode demi pas



**Figure 7:** mode de fonctionnement en demi pas moteur unipolaire

- On obtient le mode demi pas en associant les modes 1 et 2.

## le fonctionnement et la commande des moteurs pas à pas

Ce qui nous donne la séquence suivante : A / A et C / C / C et B / B / B et D / D / D et A / A . . .  
et pour l'autre sens : A / D et A / D / B et D / B / C et B / C / A et C / A . . .

| Mode monophasé |   |   |   | Mode biphasé |   |   |   | Mode demi-pas |   |   |   |
|----------------|---|---|---|--------------|---|---|---|---------------|---|---|---|
| D              | C | B | A | D            | C | B | A | D             | C | B | A |
| 0              | 0 | 0 | 1 | 0            | 0 | 1 | 1 | 0             | 0 | 0 | 1 |
| 0              | 0 | 1 | 0 | 0            | 1 | 1 | 0 | 0             | 0 | 1 | 1 |
| 0              | 1 | 0 | 0 | 1            | 1 | 0 | 0 | 0             | 0 | 1 | 0 |
| 1              | 0 | 0 | 0 | 1            | 0 | 0 | 1 | 0             | 1 | 1 | 0 |
|                |   |   |   |              |   |   |   | 0             | 1 | 0 | 0 |
|                |   |   |   |              |   |   |   | 1             | 1 | 0 | 0 |
|                |   |   |   |              |   |   |   | 1             | 0 | 0 | 0 |
|                |   |   |   |              |   |   |   | 1             | 0 | 0 | 1 |

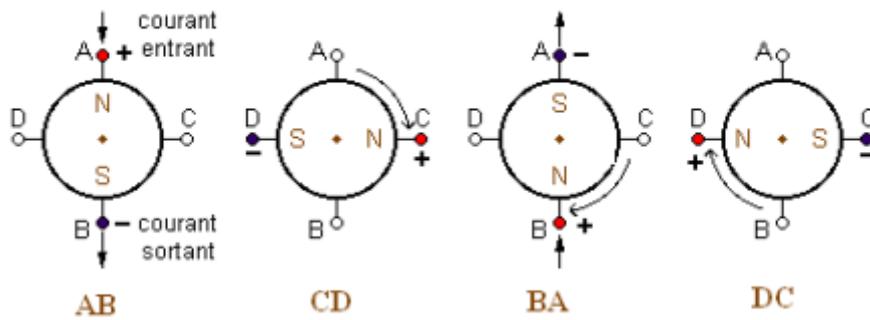
**Tableau 1** : les modes de commandes du moteurs pas à pas unipolaire

### 6-3- Moteur bipolaire

Le principe de ce type de moteur est différent à celui de moteur unipolaire. Le courant doit changer de sens dans les enroulements à chaque pas effectué. Ce moteur, comme le modelé précédent peut être alimenté sous 3 trois séquence différentes .

#### 6-3-1-mode monophasé et biphasé pas entier :

##### a- Mode monophasé :



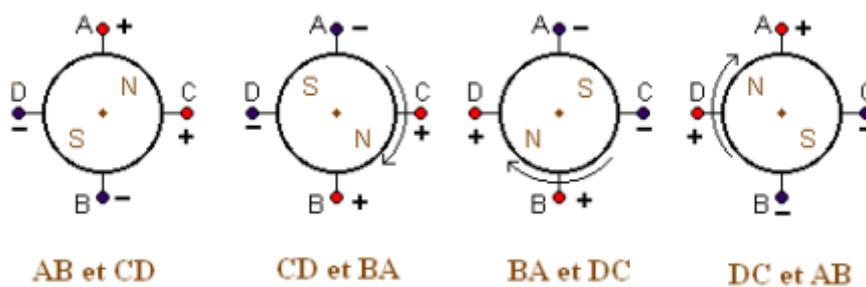
**Figure 8 :** moteur bipolaire monophasé (pas entier)

- Cette séquence de commande ne réclame que l'alimentation d'un seul enroulement à chaque pas.

La séquence de commande est : AB / CD / BA / DC / AB ...

Pour obtenir une rotation dans la direction opposée, il suffit d'inverser les séquences de commande, donc : AB / DC / BA / CD / AB ...

**b- Mode biphasé :**



**Figure 9 :** moteur bipolaire biphasé (pas entier)

- Il consiste à alimenter deux enroulements à la fois de façon à ce que le rotor se positionne entre deux pôles.

La séquence sera : AB et CD / CD et BA / BA et DC / DC et AB / AB et CD . . .

et pour l'autre sens : AB et CD / DC et AB / BA et DC / CD et BA / AB et CD . . .

### 6-3-2-Principe en mode demi – pas :

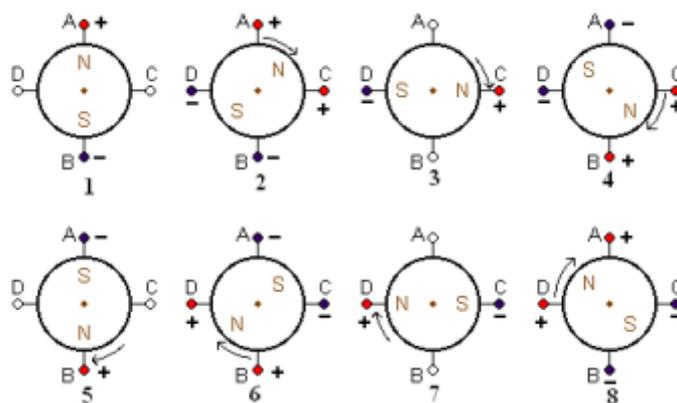


Figure 10 :moteur bipolaire demi-pas

- C'est un mélange de mode biphasé et monophasé. Le moteur est commandé en biphasé, puis en monophasé, puis en biphasé . . .

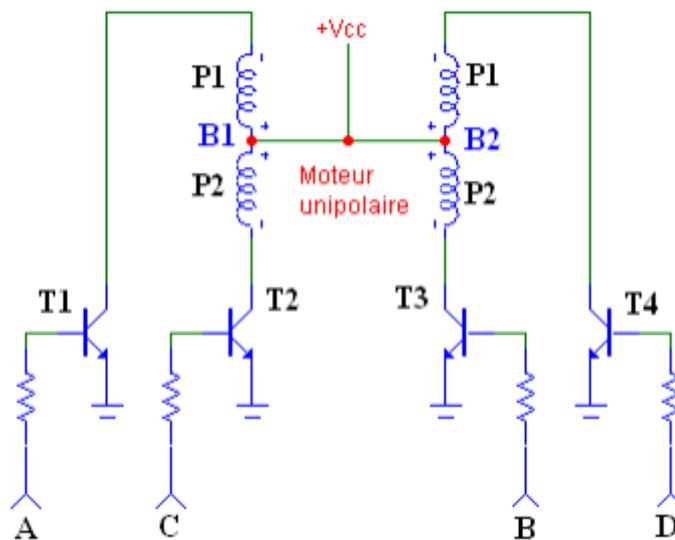
La séquence sera donc : AB / AB et CD / CD / CD et BA / BA / BA et DC / DC / DC et AB . . .

et pour l'autre sens : AB / DC et AB / DC / BA et DC / BA / CD et BA / CD / AB et CD . . .

## 7- l'alimentation des moteurs pas à pas unipolaire et bipolaire :

### 7-1- l'alimentation du moteur unipolaire :

Le circuit électronique de commande de ce moteur est simple puisqu'il ne réclame, dans sa version la plus simple, que quatre transistors utilisés en commutation.



**Figure 11** : l'alimentation du moteur pas à pas unipolaire

Il suffit d'alimenter, en reliant à la masse, l'extrémité de l'une des phases, l'autre extrémité étant connectée en permanence au + de l'alimentation. On voit qu'un seul transistor sera nécessaire pour cette commutation. Le moteur comprenant quatre bobinages donc quatre transistors seront utilisés.

## 7-2- l'alimentation du moteur bipolaire :

Pour la commande d'un moteur bipolaire, c'est un peu plus compliqué puisqu'il faudra inverser le sens du courant dans les bobinages à chaque pas effectué par le moteur. Deux solutions sont envisageables : dans le premier cas, un pont de quatre transistors par phase doit être utilisé ; dans le second, on peut n'utiliser que deux transistors, mais dans ce cas, une alimentation symétrique sera nécessaire.

### 7-2-1- le pont H :

A chaque commande, deux des transistors placés en diagonale de chacun des deux ponts seront commutés dans le même temps. Ainsi, on obtiendra une inversion de la polarité de l'alimentation de la bobine, et donc une inversion du sens de courant.

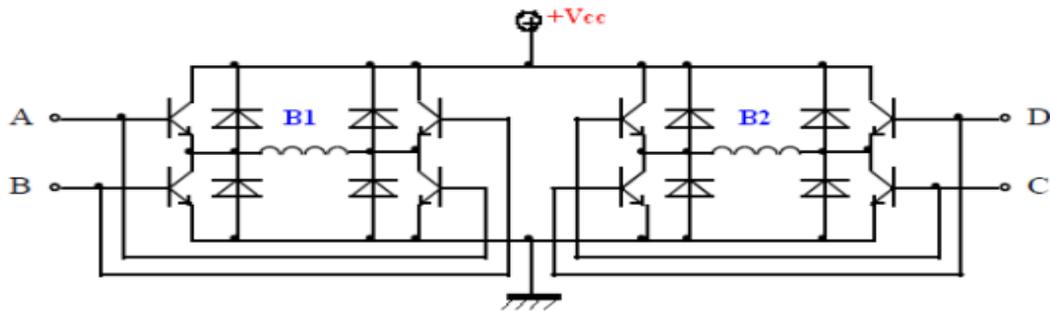


Figure 12 : l'alimentation en pont H du moteur pas à pas bipolaire

### 7-2-2- l'alimentation symétrique :

Ici, l'un des pôles de chaque bobinage est connecté à la masse. L'autre extrémité sera elle reliée à un couple de transistors dont la commande séparée l'amènera soit au potentiel positif de l'alimentation, soit au potentiel négatif. On obtient ainsi une inversion du courant dans la bobine.

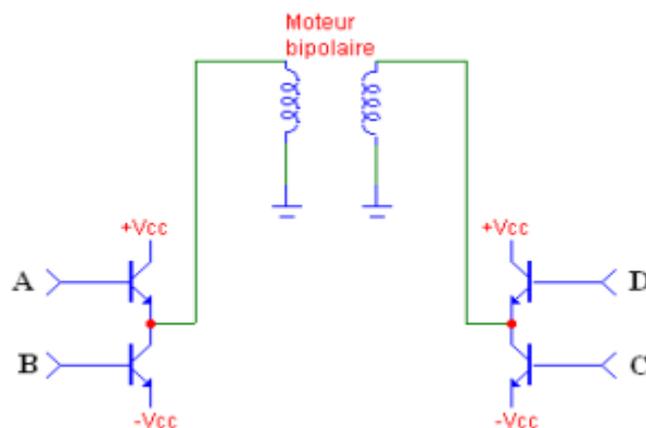


Figure 13: l'alimentation symétrique du moteur pas à pas bipolaire

### 8-Conclusion :

Dans ce chapitre, nous avons étudié les moteurs pas-à-pas, leurs électroniques de puissance et leurs électroniques de commandes. Durant cette étude nous avons travaillé sur les

## *le fonctionnement et la commande des moteurs pas à pas*

---

différents types et technologies de moteurs en générale , et nous avons basé surtout aux moteurs à aimant permanent les signaux électriques qu'ils acceptent pour fonctionner. Ce travail nous a permis de se familiariser avec les domaines comme électronique de puissance et électromécanique.



# Chapitre 03

### Chapitre 03 : Description de la carte ARDUINO

#### 1-introduction :

L'industrie ancienne des circuits électroniques nécessitait une certaine complexité au niveau du plan de projets électroniques aussi dans la recherche de nombreux composants électroniques, mais le développement dans le domaine des circuits intégrés a grandement facilité la fabrication de projets utilisant de petits circuits dont la taille est estimée en nanomètres. Grâce à ce développement, un type particulier de circuit électronique est apparu, à savoir les microcontrôleurs (considéré comme un ordinateur miniature) qui contrôlent un ensemble de services, et ce en le programmant facilement à l'aide d'une solution hardware et une solution software proposé, qui se nomme INTEGRATED DEVELOPMENT ENVIRONMENT (IDE) et qui utilise Le langage ARDUINO qui se distingue parmi les langages utilisés dans l'industrie

En Italie, en 2005, l'idée d'ARDUINO est apparue. est une marque qui fait des cartes conçues pour réaliser des prototypes et des maquettes de cartes électroniques pour l'informatique embarquée. L'objectif est de créer un environnement de développement de microcontrôleur précis permettent un accès simple et open source à faible coût, au profit des étudiants et des passionnés d'électronique.

L'utilisation de pas mal d'outils et d'accessoires de commande à distance a beaucoup simplifier notre contrôle, donc nous chercherons toujours à se concentrer sur la souplesse de la commande.

#### 2-présentation de la carte ARDUINO :

Le module ARDUINO est un petit circuit imprimé (5,33 x 6,85 cm) équipée d'un microcontrôleur c'est à dire une sorte de mini-ordinateur qui sert d'interface entre l'environnement (actions, mesures de grandeurs...) et un utilisateur.

D'une autre façons, ARDUINO est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation dans un langage dérivé du C : le langage « ARDUINO. Sans tout ne connaître ni tout comprendre de l'électronique, cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe avec l'aide de nombreuses ressource

ARDUINO est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée, la domotique mais aussi dans des domaines différents comme le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme).

#### 3-Les différents gammes de la carte ARDUINO :

Actuellement, il existe plus de 20 versions de module ARDUINO, qui se distingue l'une de l'autre principalement par le type du microcontrôleur utilisé, le nombre d'entrées/sorties, la capacité de la mémoire et la vitesse de fonctionnement.

## Description de la carte ARDUINO

---

nous citons quelques un afin d'éclaircir l'évaluation de ce produit scientifique et académique:

- **Carte ARDUINO UNO :**  
le bon choix pour démarrer dans le monde de l'ARDUINO est l'UNO R3. Il possède 14 pins numériques et 6 pins analogiques, un total de 20 Input Output pins qui est un nombre intéressant et suffisant pour des premières expériences.  
Il possède un microcontrôleur ATmega328 , 32 KB de mémoire programmable, 1 KB d'EEPROM et 2 KB de RAM .
- **Carte ARDUINO Nano :**  
L'ARDUINO Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- L'ARDUINO Nano, une petite carte Il ne mesure que 18x45 mm . il utilise un ATMEGA328P et opère à une vitesse de 16 MHz. Il a 14 pins numériques et 6 pins analogiques qui peuvent être alimentés en 5V et 3.3V. En termes de mémoire, il possède 32 KB de mémoire programmable, 1 KB d'EEPROM, 2 KB de RAM. En règle générale, il s'agit d'une des cartes les moins chères ce qui lui donne un avantage pour des projets bon marché.
- **Carte ARDUINO Due :**  
L'ARDUINO Due est une des plus grande carte de la marque, la Due doit être alimentée avec du 3.3V. Cette carte possède un microcontrôleur ATSAM3X8E Cortex-M3 qui tourne à 84 MHz. Il a 512 KB de mémoire ROM et 96 KB de RAM. Concernant ces GPIO il a été construit avec 54 I/O pins numériques, 12 sorties PWM, 12 entrées analogiques et 2 sorties analogiques.
- **Carte ARDUINO MEGA** , avec une interface d'USB pour programmer et usage d'un ATMEGA 2560
- **Le NG d'ARDUINO**, avec une interface d'USB pour programmer et usage d'un ATmega8.
- **L'extrémité d'ARDUINO**, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- **L'ARDUINO Mini**, une version miniature de l'ARDUINO en utilisant un microcontrôleur ATmega168.
- **Le LILYPAD ARDUINO**, une conception de minimaliste pour l'application WEARABLE en utilisant un microcontrôleur ATmega168.
- **Le NG d'ARDUINO plus**, avec une interface d'USB pour programmer et usage d'un ATmega168.
- **L'ARDUINO DIECIMILA**, avec une interface d'USB et utilise un microcontrôleur ATmega168.
- **L'ARDUINO DUEMILANOVE ("2009")**, en utilisant un microcontrôleur l'ATmega168 (ATmega328 pour une plus nouvelle version) et actionné par l'intermédiaire de la puissance d'USB/DC.
- **L'ARDUINO Leonardo**, avec un morceau ATmega328 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.

## Description de la carte ARDUINO

- **L'ARDUINO ESPLORA** : ressemblant à un contrôleur visuel de jeu, avec un manche et des sondes intégrées pour le bruit, la lumière, la température, et l'accélération.

Les cartes ARDUINO les plus répandues pour les applications courantes sont la carte UNO et la carte MEGA .

Dans la suite de ce chapitre on s'intéressera uniquement à la carte ARDUINO méga 2560.

### 4- étude de La carte ARDUINO MEGA 2560 :

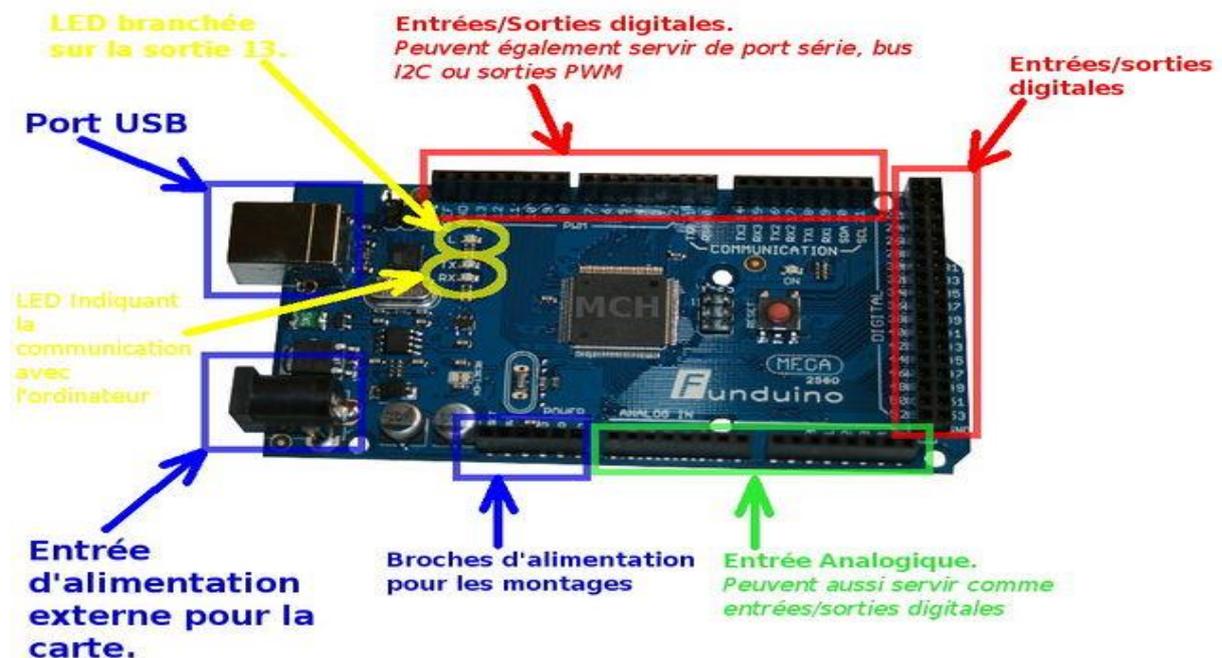


Figure 24 : Description de la carte ARDUINO MEGA 2560

#### 4.1. présentation de la carte ARDUINO MEGA 2560 :

L'ARDUINO MEGA 2560 est un autre produit populaire parmi les cartes ARDUINO. Parfaite pour les applications ARDUINO un peu plus grandes, elle est constituée de tous les éléments nécessaires pour permettre la construction d'objets évolués.

La carte ARDUINO MEGA 2560 est constituée de 54 broches d'entrées/sorties, dont 15 sont utilisables en PWM, de 16 broches d'entrées analogiques, d'une connectique USB, d'une connectique d'alimentation, d'un port ICSP et d'un bouton RESET.

### 4.2. La constitution de la carte ARDUINO MEGA

Un module ARDUINO est constitué d'une :

- Partie matériel
- Partie logiciel

#### 4.2.1. la partie matériel :

- L'alimentation
- Microcontrôleur ATMEGA2560
- Mémoire flash: 256 KB
- Mémoire SRAM: 8 KB
- Mémoire EEPROM: 4 KB
- 54 broches d'E/S dont 14 PWM
- 16 entrées analogiques 10 bits
- Intensité par E/S: 40 mA
- Vitesse d'horloge 16 MHz
- Bus I2C et SPI
- Fiche USB B

#### a-Alimentation

La carte ARDUINO MEGA 2560 peut-être alimentée soit via la connexion USB (qui fournit 5V jusqu'à 500 MA) ou à l'aide d'une alimentation externe.

La tension d'alimentation extérieure (hors USB) peut venir soit d' :

- un adaptateur (pouvant fournir typiquement de 3V à 12V sous 500mA)
- de piles

a - L'adaptateur peut être connecté en branchant une prise 2.1mm positif au centre dans le connecteur jack de la carte

b - Le raccordement vers un bloc de piles peut utiliser les bornes GND (masse ou 0V) et Vin (Tension positive en entrée) du connecteur d'alimentation (POWER). Ce circuit inclut un régulateur de tension à 5 V mais la plage idéale recommandée pour alimenter la carte c'est entre 7 et 12 V pour garder une marge en basse tension et éviter que le circuit ne chauffe trop (car le régulateur de tension disperse toute surtension en chaleur).

#### b. Le Microcontrôleur ATmega2560 :

Un microcontrôleur ATmega2560 est un circuit intégré qui rassemble sur une puce plusieurs éléments complexes dans un espace réduit, c'est le processeur de la carte, s'occupe de tout ce qui est calculs, exécution des instructions du programme et gestion des ports d'entrée/sortie.

### c. Les mémoires :

L'ATMEGA 2560 à 256Ko de mémoire FLASH pour stocker le programme (dont 8Ko également utilisés par le BOOTLOADER), également 8 ko de mémoire SRAM (volatile) et 4Ko d'EEPROM (non volatile - mémoire qui peut être lue à l'aide de la librairie EEPROM)

- La mémoire Flash : C'est celle qui contiendra le programme à exécuter, Cette mémoire est effaçable et réinscriptible.
- RAM : c'est la mémoire dite "vive", elle va contenir les variables de votre programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur.
- EEPROM : C'est le disque dur du microcontrôleur. Vous pourrez y enregistrer des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée. Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme.
- Les registres : c'est un type de mémoire utilisé par le processeur.
- La mémoire cache : c'est une mémoire qui fait la liaison entre les registres et la RAM.

### d. Entrées et sorties :

Chacune des 14 broches **numériques** de la MEGA peut être utilisée en entrée (input) ou en sortie (output) Elles fonctionnent en logique TTL (0V-5V) , chacune pouvant fournir (source) ou recevoir un courant maximal de 40 mA .

Certaines broches ont des fonctions spécialisées :

- Serial : broche 0 (RX) et broche1 (TX). Permet de recevoir (RX) et de transmettre (TX) des données séries .
- Interruptions externes 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur un front montant ou descendant, ou encore sur le changement de valeur.
- PWM: 3, 5, 6, 9, 10 numérique , et 11. Output 8-bit de PWM avec la fonction `ANALOG WRITE()`.
- LED : Il y a une LED connectée à la broche numérique 13.
- I2C : Permettent le support du bus I2C

La carte Mega2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits à l'aide de la très utile fonction [ANALOGREAD\(\)](#) du langage ARDUINO.

#### ➤ Autres broches

- **Reset** : Mettre cette broche au niveau BAS entraîne la réinitialisation (= le redémarrage) du microcontrôleur. Typiquement, cette broche est utilisée pour ajouter un bouton de réinitialisation sur le circuit qui bloque celui présent sur la carte.

### Notation :

**transistor-Transistor LOGIC** ou **TTL** est une famille de [circuits logiques](#) utilisée en [électronique](#) inventée dans les années 1960. Cette famille est réalisée avec la technologie

du [transistor bipolaire](#) .La technologie TTL est normalisée pour une **tension d'alimentation** ( $V_{CC}$ ) de **5 V**.

### f. La communication :

La carte ARDUINO Mega2560 dispose de toute une série de facilités pour communiquer avec un ordinateur, une autre carte ARDUINO, ou avec d'autres microcontrôleurs.

Un circuit intégré ATmega8U2 sur la carte assure la connexion l'un des ports série de l'ATMEGA 2560 vers le port USB de l'ordinateur qui apparaît comme un port COM virtuel pour les logiciels de l'ordinateur. Le code utilisé pour programmer l'ATmega8U2 utilise le driver standard USB COM, et aucun autre driver externe n'est nécessaire.

Le logiciel ARDUINO inclut une fenêtre terminal série (ou moniteur série) sur l'ordinateur et qui permet d'envoyer des textes simples depuis et vers la carte ARDUINO. Les LEDS RX et TX sur la carte clignote lorsque les données sont transmises via le circuit intégré ATmega8U2.

L'ATMEGA2560 supporte également la communication par protocole I2C (ou interface TWI (TWO WIRE Interface - Interface "2 fils") .

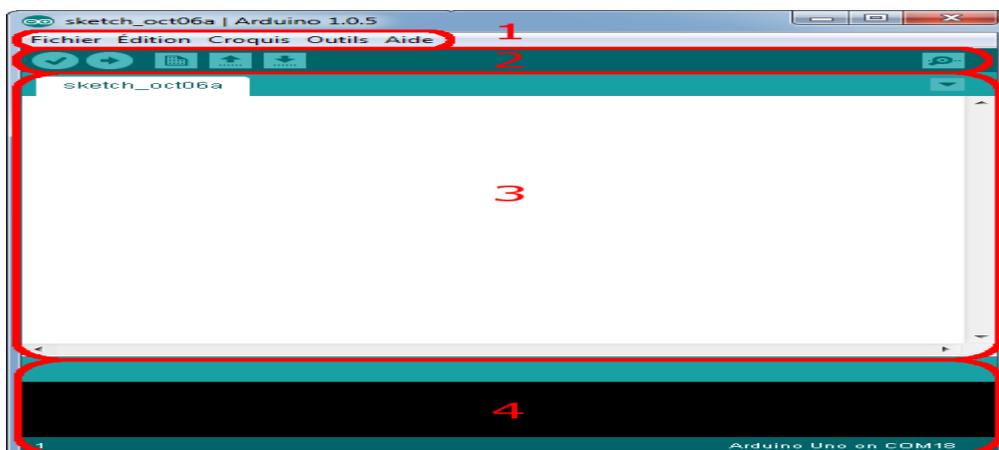
### 4.2.2 partie logiciel :

Pour pouvoir créer nos projets on doit avant tout savoir programmer pour ARDUINO. Le logiciel pour le développement de ARDUINO (ide INTEGRATED DEVELOPMENT ENVIRONNEMENT ) est un interface graphique d'utilisation simple que regroupe tous les instruments nécessaires pour le codage .

L'environnement de programmation open-source pour ARDUINO peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux).

#### 4.2.2.1.L'interface :

L'interface du logiciel ARDUINO se présente de la façon suivante :



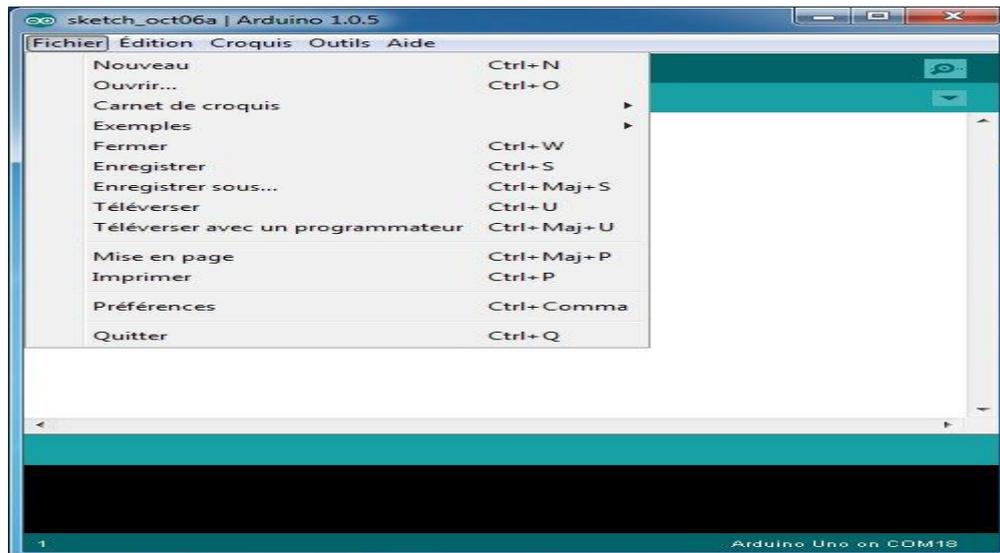
**Figure 14** : présentation l'interface ARDUINO

## Description de la carte ARDUINO

---

1. options de configuration du logiciel
2. boutons pour la programmation des cartes
3. programme à créer
4. débogueur (affichage des erreurs de programmation)

Le menu fichier dispose d'un certain nombre de choses qui vont être très utiles :

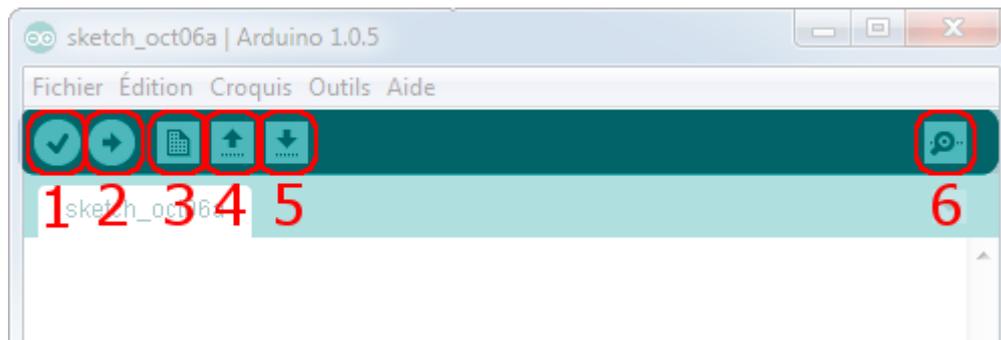


**Figure 15:** présentation du menu fichier

(nouveau) : va permettre de créer un nouveau programme. Quand on appuie sur ce bouton, une nouvelle fenêtre, identique à celle-ci, s'affiche à l'écran.

- (ouvrir) : avec cette commande, on peut ouvrir un programme existant.
- (enregistrer / enregistrer sous...) : enregistre le document en cours / demande où enregistrer le document en cours.
- (exemples) : ceci est important, toute une liste se déroule pour afficher les noms d'exemples de programmes existant.

### 4.2.2.2. Les boutons de programmation :



**Figure 16 :** boutons de programmation

1. permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme
2. Créer un nouveau fichier
3. Sauvegarder le programme en cours
4. Liaison série
5. Stoppe la vérification
6. Charger un programme existant
7. Compiler et envoyer le programme vers la carte

### 4.2.2.3. Le langage ARDUINO

Cependant, le projet ARDUINO a développé des fonctions spécifiques à l'utilisation de la carte qui ont été listées ci-dessous. Vous obtiendrez la description de chacune d'elles dans le manuel de référence

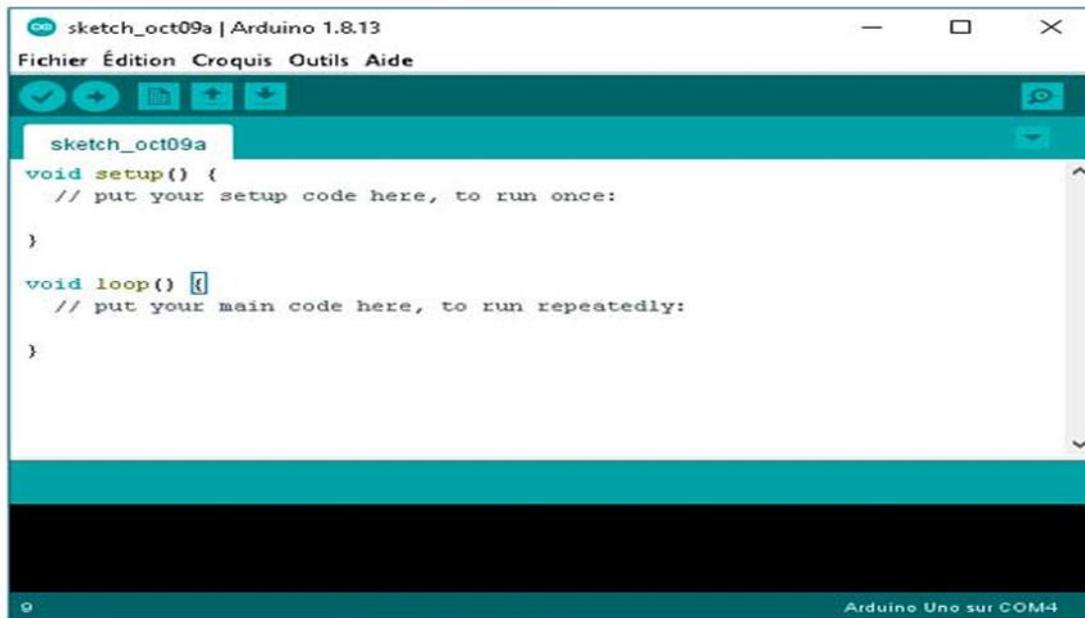
| Structure         | Constants   | Fonctions  |
|-------------------|---|--|
| setup()<br>loop() | HIGH, LOW<br><br>• INPUT, OUTPUT,<br>INPUT_PULLUP | E/S NUMÉRIQUE<br>▪ PINMODE()<br>▪ DIGITALWRITE()<br>▪ DIGITALREAD()<br><br>E/S ANALOGIQUE<br>▪ ANALOGREFERENCE() |

|  |   |   |
|--|---|---|
|  | <ul style="list-style-type: none"> <li>• LED_BUILTIN</li> </ul> | <ul style="list-style-type: none"> <li>▪ ANALOGREAD()</li> <li>▪ ANALOGWRITE() - PWM</li> </ul> <p>E/S AVANCÉE</p> <ul style="list-style-type: none"> <li>▪ TONE()</li> <li>▪ NOTONE()</li> <li>▪ SHIFTOUT()</li> <li>▪ SHIF TIN()</li> <li>▪ PULSEIN()</li> </ul> <p>TEMPS</p> <ul style="list-style-type: none"> <li>▪ MILLIS()</li> <li>▪ MICROS()</li> <li>▪ DELAY()</li> <li>▪ DELAYMICROSECONDS()</li> </ul> <p>Bits et octets</p> <ul style="list-style-type: none"> <li>▪ LOWBYTE()</li> <li>▪ HIGHBYTE()</li> <li>▪ BITREAD()</li> <li>▪ BITWRITE()</li> <li>▪ BITSET()</li> <li>▪ BITCLEAR()</li> </ul> <p>Interruptions externes</p> <ul style="list-style-type: none"> <li>▪ ATTACHINTERRUPT()<br/>DETACHINTERRUPT()</li> </ul> <p>Interruptions</p> <ul style="list-style-type: none"> <li>▪ INTERRUPTS()</li> <li>▪ NOINTERRUPTS()</li> </ul> <p>Communication</p> <ul style="list-style-type: none"> <li>• Serial</li> <li>• Stream</li> </ul> |
|--|---|---|

### Tableaux 02 : le manuel référence de chaque fonction dans IDE

#### 4.2.2.4. La structure du code

La structure du code se place au centre de la fenêtre pour initialiser la programmation



```
sketch_oct09a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Arduino Uno sur COM4

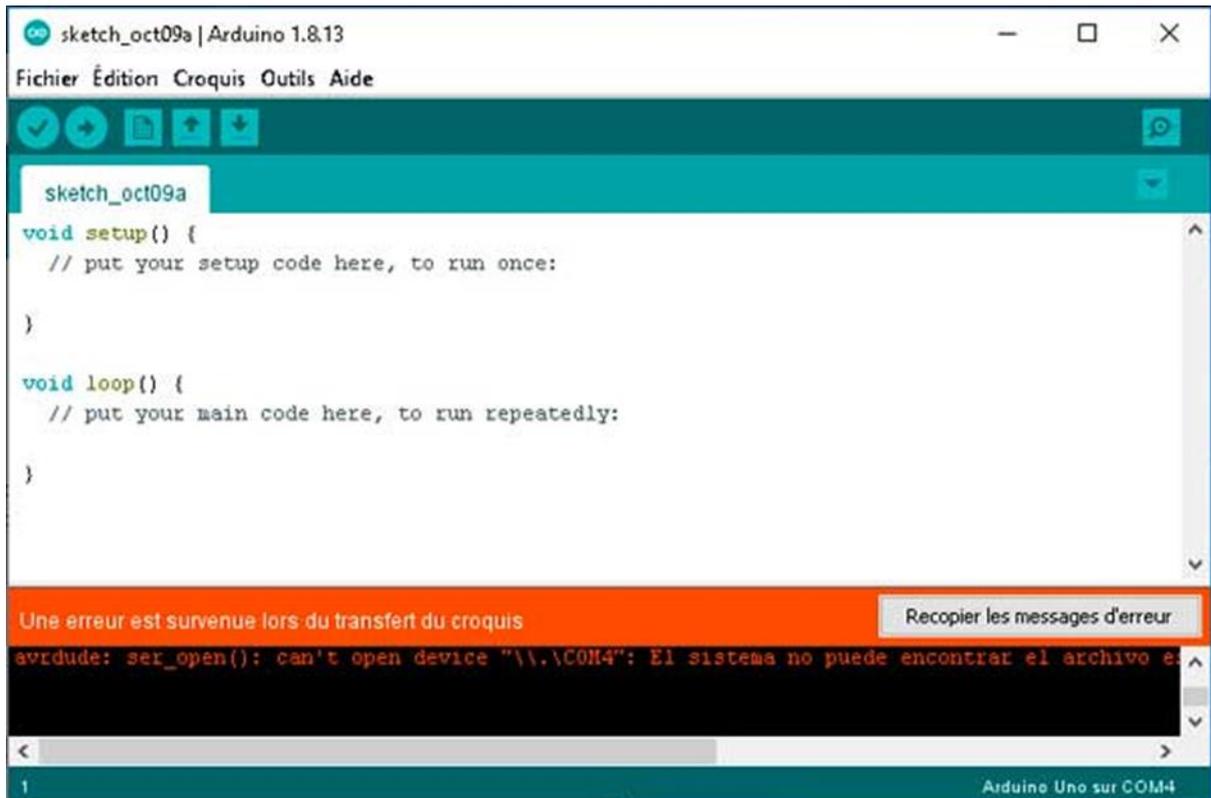
**Figure 17:**présentation du code

Ce code basique est composé de deux structures dans lesquelles on insérera les variables, les constantes, les conditions et les boucles nécessaires pour développer les programmes des projets qu'on voudra réaliser.

- l'instruction VOID *SETUP()*, est considérée la fonction d'initialisation de la carte.
- l'instruction VOID *LOOP()* est la fonction que se répète en boucle.

Ces deux structures ou fonctions sont les plus importantes et elles doivent se configurer de façon précise.

Dans la partie inférieure de la fenêtre de l'IDE on trouve la *console de débogage* où les erreurs d'exécution du code seront remarquées en cas de faille dans la compilation. Grâce à cette console, on est certains que notre code est correctement rédigé avant du télé versement sur la carte.



**Figure 18:** fenêtre inférieur d'ide

### 4.2.2.5. Injection du programme

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte ( ARDUINO UNO) et le numéro de port USB .

## Description de la carte ARDUINO

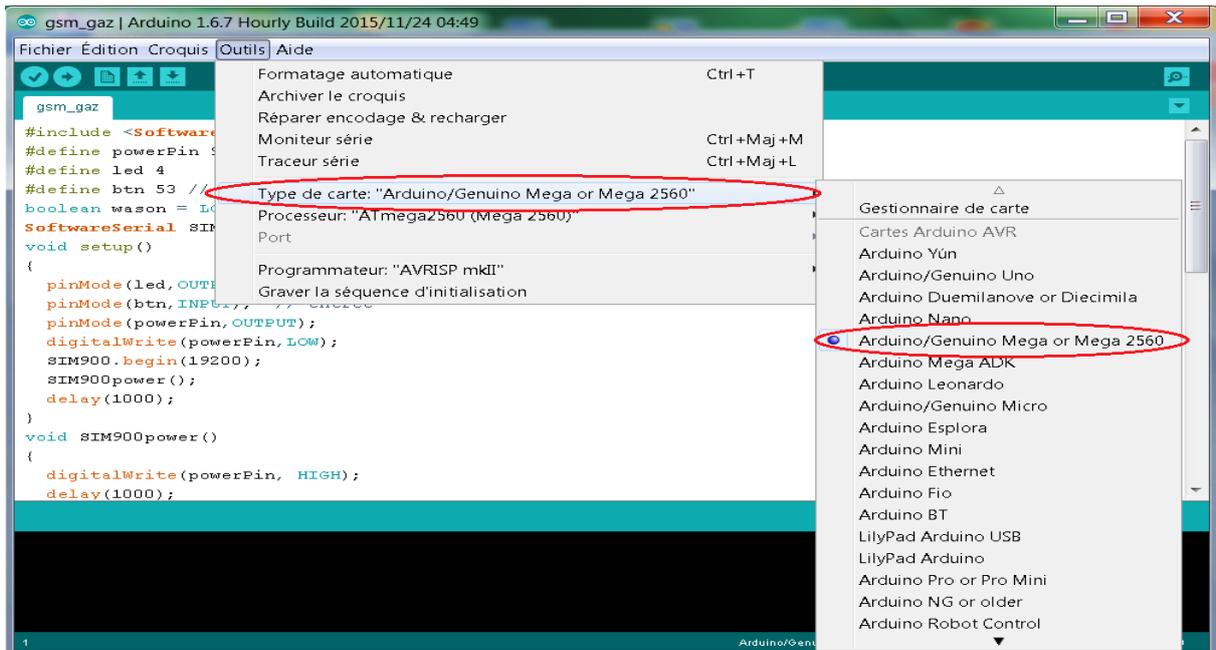


Figure 19: la sélection du type de la carte

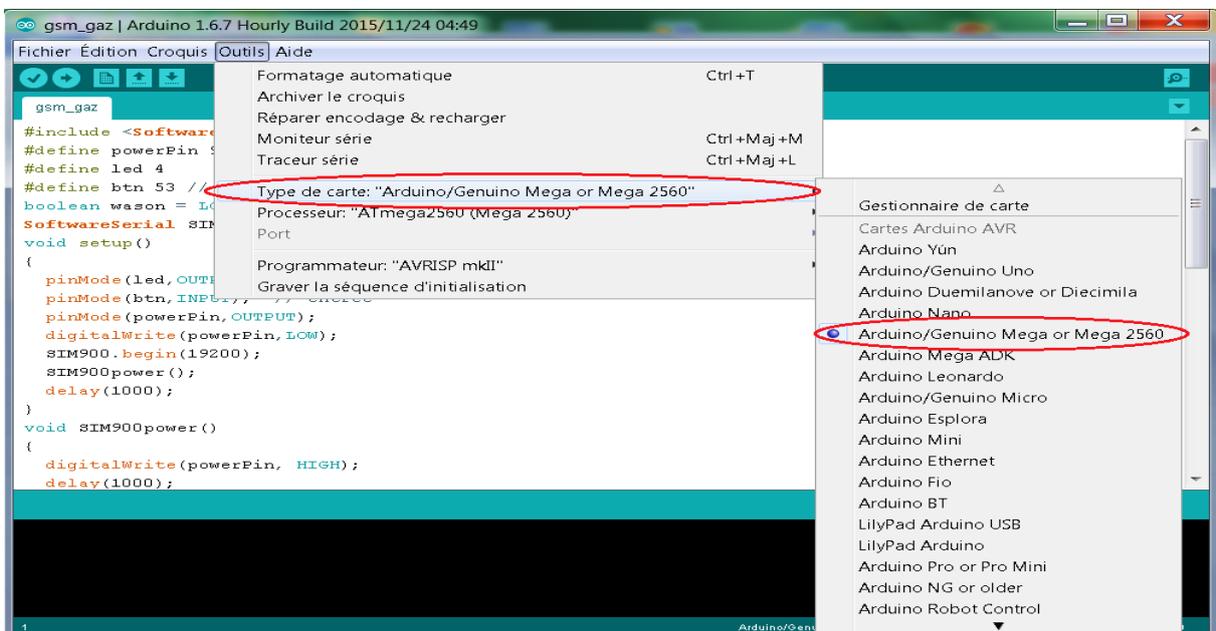


Figure 20: le numéro du port USB

## Description de la carte ARDUINO

Une simple manipulation enchaînée doit être suivie afin d'injecter un code vers la carte ARDUINO via le port USB.

1. On conçoit ou on ouvre un programme existant avec le logiciel IDE ARDUINO.
2. On vérifie ce programme avec le logiciel ARDUINO (compilation).
3. Si des erreurs sont signalées, on modifie le programme
4. On charge le programme sur la carte.
5. On câble le montage électronique.
6. L'exécution du programme est automatique après quelques secondes.
7. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome (pile 9 volts par exemple).
8. On vérifie que notre montage fonctionne.

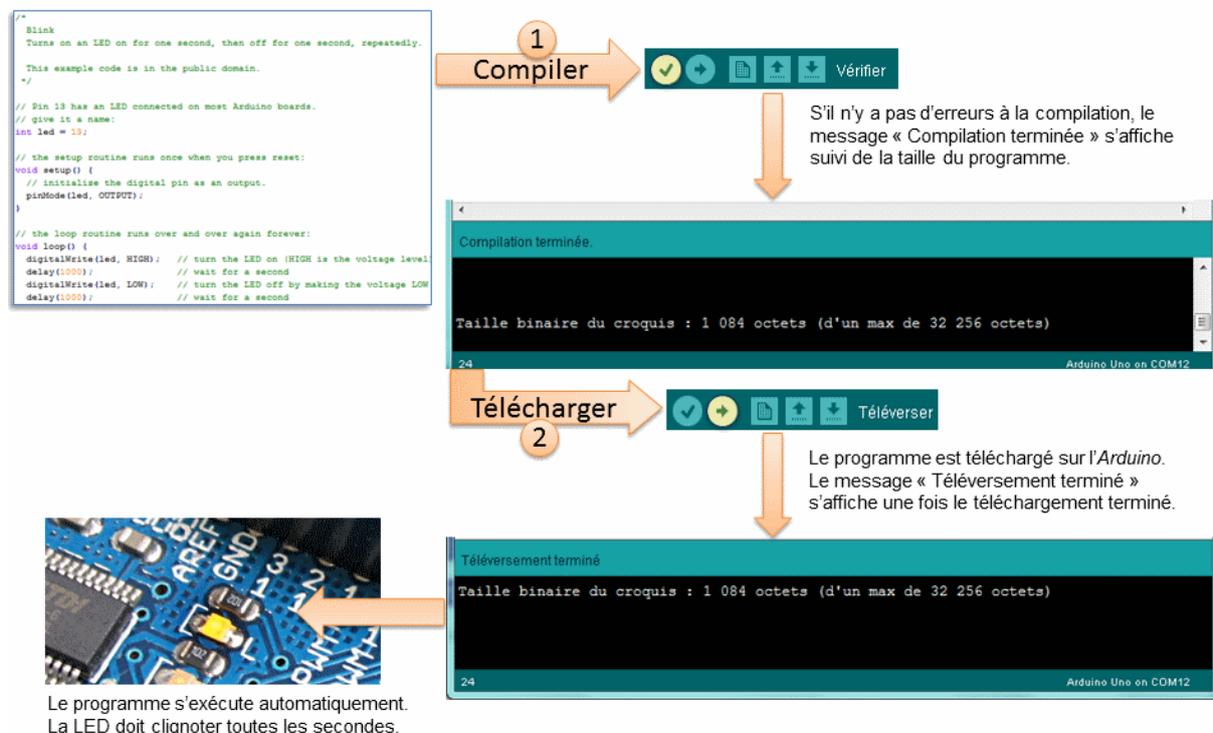
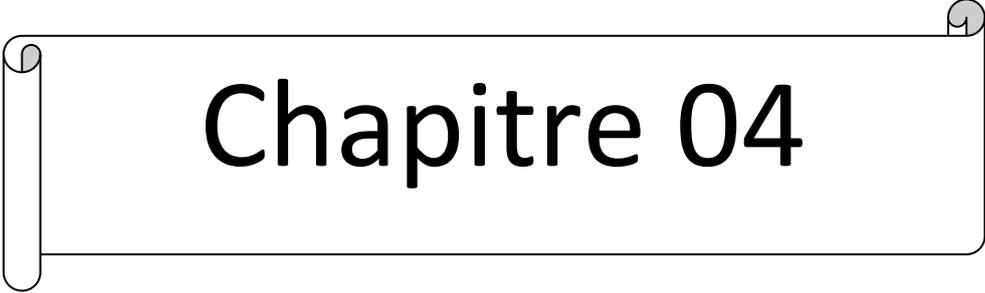


Figure 21: compilation et téléchargement du code

## **5.Conclusion :**

Dans ce chapitre, nous avons donné quelques concepts théoriques , ainsi nous avons projeté la lumière sur une carte d'acquisition qui est l'ARDUINO donnant ainsi les raisons pour lesquelles on l'a choisie, puis nous avons cité des différents types de cette dernière. Ensuite, nous avons expliqué les deux parties essentielles de l'ARDUINO; (la partie matérielle et la partie de programmation) plus précisément. Nous avons également expliqué le principe de fonctionnement de la carte ARDUINO sans oublier ses caractéristiques. Le chapitre suivant sera consacré à l'étude et la réalisation d'un dispositif de commande automatique des moteurs pas à pas.



# Chapitre 04

### **Chapitre 04 : principe de fonctionnement du système d'irrigation**

#### **1-Introduction**

Notre projet principal était de réaliser un système de commande de deux moteurs pas à pas. Mais il s'est avéré qu'il vaudrait mieux le mettre en pratique en créant une application utile dans la vie quotidienne. En effet, on avait le choix entre une chaise roulante ou un système d'irrigation et on s'est convenu sur ce dernier apparemment non disponible d'après notre conception. La mise en relief du système d'irrigation par la commande de deux moteurs pas à pas nécessite un schéma détaillé et précis. Les pièces et les supports dans le projet ont été forgées ; en sorte de mettre le système en marche et garder son bon équilibre.

#### **2-Description de Schéma et principe de fonctionnement du système d'irrigation :**

L'irrigation se fait par bassin au moyen d'une rampe mobile se déplaçant sur deux rails parallèles comme l'indique le schéma présenté ci-dessus.

Cette structure est composée d'une rampe mobile commandée par deux moteurs dit "pas à pas" qui ont pour effet d'assurer son déplacement automatiquement et régulier le long des rails fixées sur les abords du sol dans le but d'effectuer un arrosage d'appoint. Le système mobile d'irrigation que nous concevons est matérialisé par une plaque de commande électronique dite ARDUINO (bloque de contrôle) .

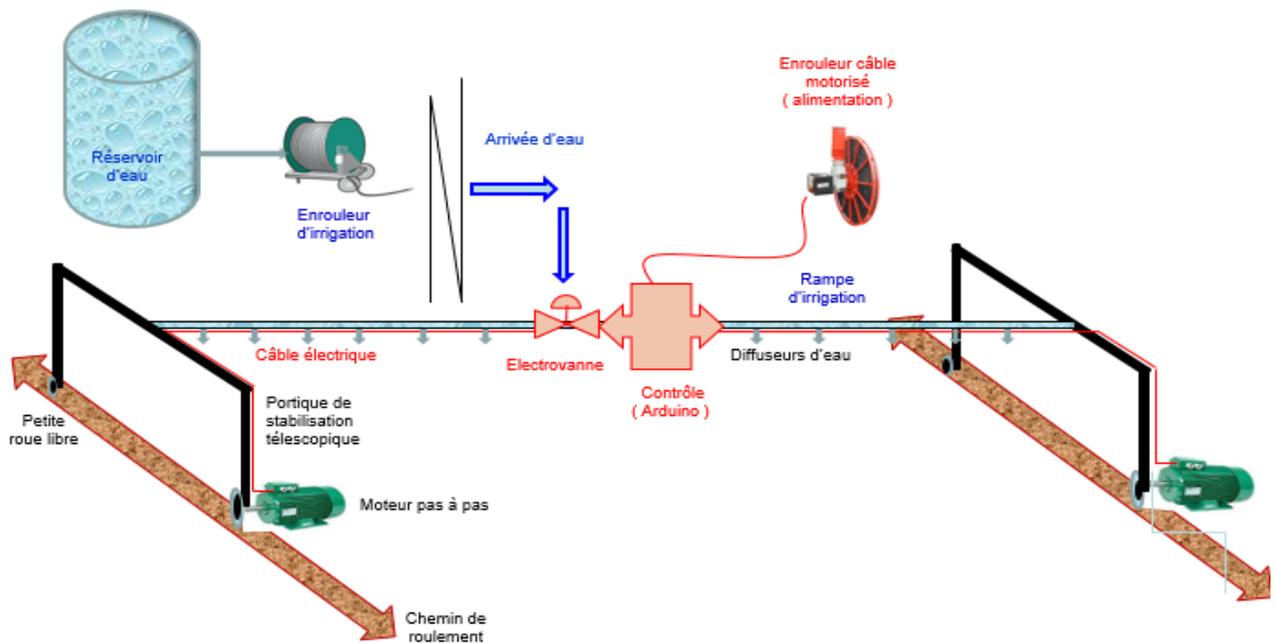
Les asperseurs sont uniformément positionnés le long de la rampe mobile qui elle-même est reliée à un enrouleur nécessaire à l'apport d'eau provenant du puits ou d'un bassin d'eau.

Le matériel d'irrigation par aspersion en modèle de conception réduite que nous proposons serait vraisemblablement efficace en matière de rentabilité économique et ne s'applique pour des exploitations agricoles aux surfaces réduites telles que les potagers ou surfaces maraîchères.

Comme source d'énergie nécessaire au déplacement de notre système à rampe mobile, nous pouvons envisager un réseau électrique pour alimenter les moteurs "pas à pas" mais aussi adapter des panneaux solaires ou des batteries tout simplement, et c'est ce que nous proposons dans notre présent montage de système d'irrigation miniaturisé.

Le système d'irrigation à rampe mobile se déplace grâce à ses deux roues motrices accouplées à deux moteurs "pas à pas". Les roues calibrées de la même manière seront posées sur des rails et c'est tout le système qui déplace doucement selon la trajectoire de leur chemin de fer en aspergeant l'eau nécessaire au potager par le biais de flexible déroulant. Evidemment, on peut intégrer un programmeur d'arrosage automatique pour l'arrêt et la mise en marche de notre système d'irrigation.

L'avantage de ce système d'irrigation peut être monté ou démonté à volonté rapidement et le ranger dans un hangar durant la saison pluvieuse et, son entretien est facile à faire et peu onéreux.



## Système d'irrigation avec deux moteurs pas à pas roulant

Figure 22: Schéma de principe d'irrigation

Le schéma, ci-joint permet d'appréhender assez facilement le concept de notre système d'irrigation "pas à pas" à l'effet de pouvoir l'appliquer à la grandeur nature et suivant la superficie des horticultures maraichères . ce système peut s'appliquer également aux grands jardinages avec ou sans serre .

### 3-Les composants utilisés :

- 1- 2 moteurs pas à pas unipolaire **28BYJ-48**
- 2- Un module **RELAIS 5V DC**
- 3- 1 électrovanne 220 v
- 4- Bloc de contrôle :
  - La carte **ARDUINO MEGA 2560**

## principe de fonctionnement du système d'irrigation

---

➤ Deux drivers **ULN2003A**

5- Des conducteurs ou des fils de prototypage ( malle /malle , malle / femelle, femelle/femelle) .

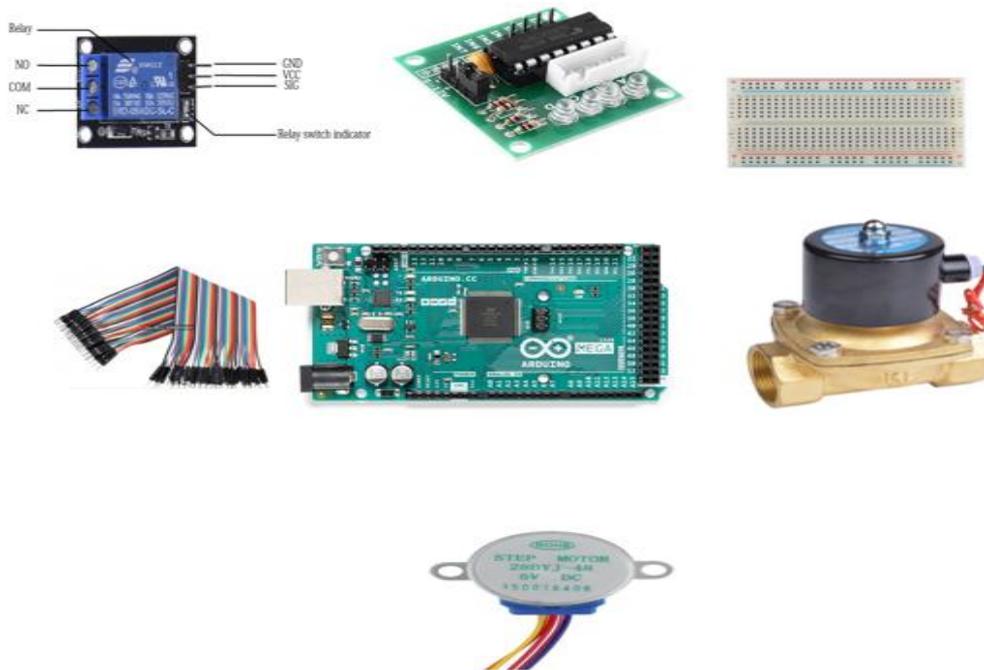
**6- ELEGOO 830 BREADBOARD**

7- 4 roue libre

8- Un enroulement d'irrigation

9- Deux supports

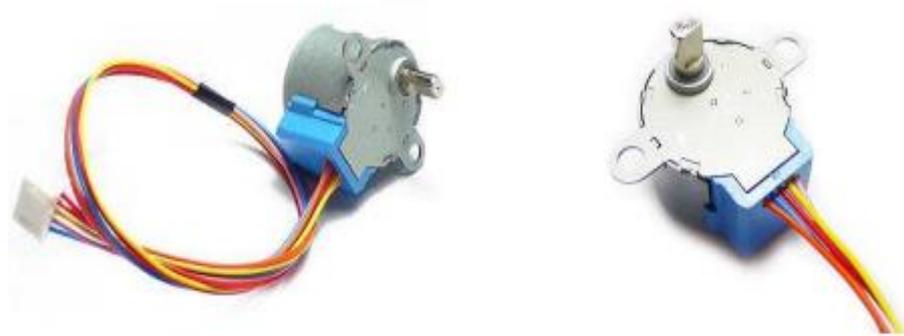
10- Diffuseurs d'eau



**Figure 23:** La composante liste pour la réalisation

### 3-1-28BYJ-48 - Moteur pas à pas 5V

Le 28BYJ-48 est un petit moteur pas à pas adapté à une large gamme d'applications.



**Figure 24:** Moteur unipolaire à 5 fils

#### 3.1.1. Quelques détails typiques du moteur pas à pas :

Modèle 28BYJ-48 – 5V

Bobine : Bobine unipolaire à 5 fils

Tension nominale : 5VDC

Nombre de phase :4

Angle 5,625 ° / 64

Fréquence 100 Hz

Bobine unipolaire à 5 fils

Résistance CC 50  $\Omega$

Augmentation de la température <40K

Bruit <35 dB (120 Hz, sans charge, 10 cm)

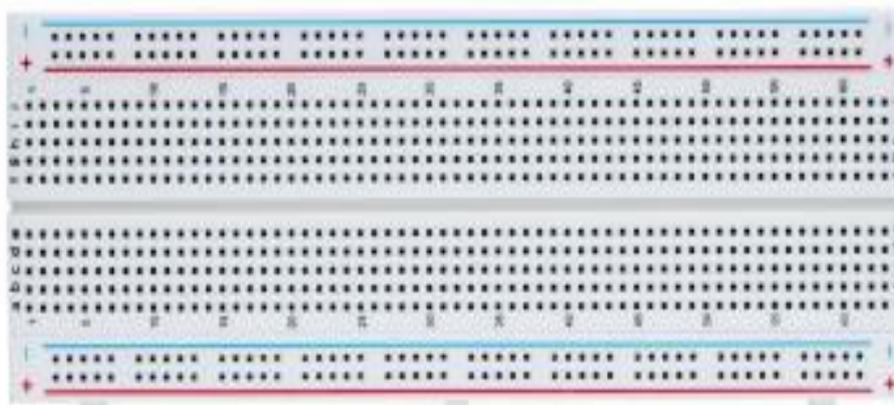
Il faut 4096 pas pour faire tourner la broche à 360°.



**Figure 25:** Architecture interne du moteur pas à pas

### 3-2-plaque d'essai ELEGOO 830 :

Les planches "enfichables" sans soudure à 830 points de liaison offrent un moyen rapide de construire et de tester des circuits pour l'expérimentation ou lors de l'apprentissage de l'électronique.



**Figure 26:** Plaque d'essai ELEGOO 830

### 3-3-Driver de pilote de moteur pas à pas ULN2003 à 4 phases :

Comme nous avons déjà vu les plusieurs gamme de la carte ARDUINO avec ces plusieurs GPIO qui nous permettent de branché plusieurs module mais selon la capacité de la carte .

Concernant les moteurs, on peut brancher directement le moteur pas à pas sur ARDUINO parce qu'il nécessite pas un grand voltage mais si on veut branché plusieurs moteurs dans ce cas on peut pas parce qu'on peut endommagé la carte ARDUINO, donc pour faire piloter le moteur pas à pas on à besoin de l'électronique de puissance pour alimenter les moteurs .

## principe de fonctionnement du système d'irrigation

Le principe de fonctionnement c'est d'appliquer les signaux électriques nécessaires au moteur pas à pas à partir d'un simple signal généré par l'ARDUINO.

Le driver pilote de moteur pas à pas ULN2003 fournit une interface de commande directe entre votre microcontrôleur et le moteur pas à pas. Le PCB fournit 4 entrées pour la connexion à votre microcontrôleur, alimentation pour la tension du moteur pas à pas et cavalier marche/arrêt, un en-tête de moteur pas à pas à connexion directe et 4 LED pour indiquer l'état de progression.

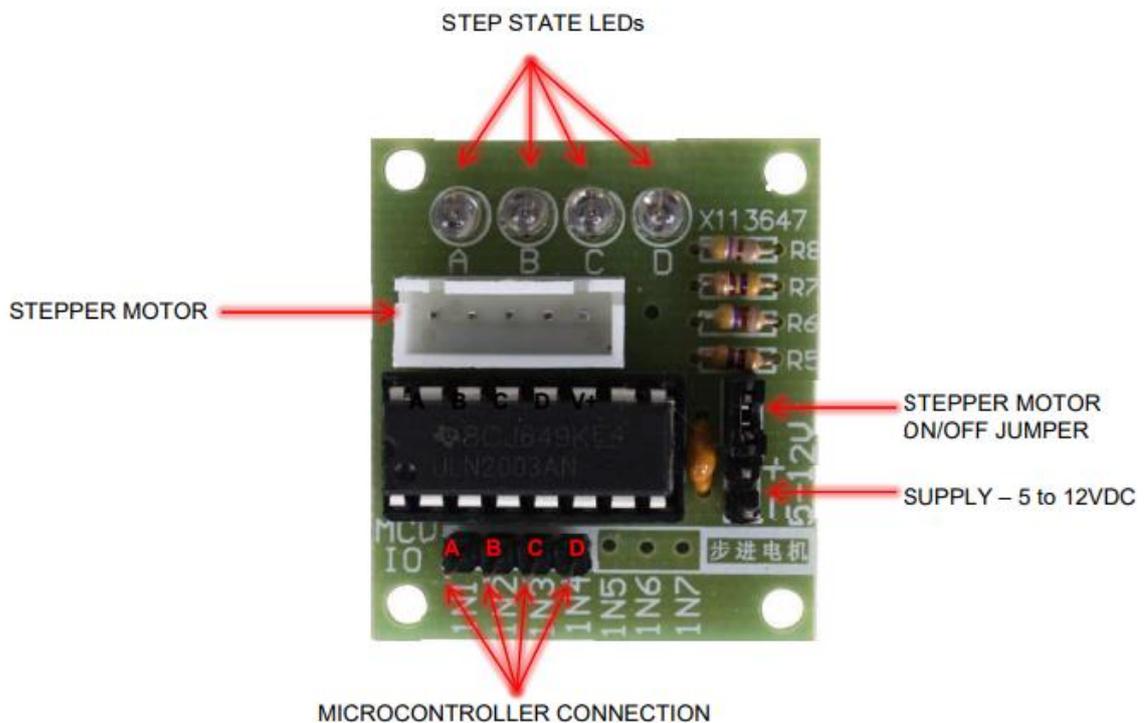


Figure 27: Driver ULN 2003A

### 3-3-1-Connexion du moteur pas à pas :

Connectez le moteur pas à pas à cet en-tête. Si vous utilisez l'un de nos moteurs pas à pas, il se branchera directement sur l'en-tête. Si vous avez une autre marque sans la bonne prise d'en-tête, vous pouvez souder les fils directement à l'arrière du PCB.

### 3-3-2-LED d'état d'étape :

Indiquez quel canal est actif.

### 3-3-3-Fourniture

## principe de fonctionnement du système d'irrigation

---

Puissance pour entraîner le moteur pas à pas. Assurez-vous qu'il correspond à la puissance nominale du moteur pas à pas que vous utilisez.

### 3-3-4-Connexion du microcontrôleur

Connectez 4 sorties de votre microcontrôleur aux quatre entrées.

Volts d'alimentation : 5-12 VDC

Courant maximum par sortie : 500mA

### 3-3-5-Opération :

Le driver accepte une commande à quatre bits de n'importe quel microcontrôleur et applique à son tour la puissance nécessaire pour faire avancer le moteur. Au cœur du driver se trouve un circuit intégré ULN2003AN. Le driver peut fournir entre 5V et 12V au moteur à partir d'une alimentation indépendante. Il a également une banque de LED qui correspondent aux signaux d'entrée reçus du contrôleur. Ils offrent un joli visuel lors de la marche

### 3-3-6-La connexion physique d'un microcontrôleur à la carte de commande :

La connexion physique d'un microcontrôleur à la carte driver est simple. Choisissez une broche GPIO libre sur l'en-tête d'extension et faites passer un fil de celui-ci à l'une des broches d'entrée sur le driver . le driver nécessite Puissance. Assurez-vous que votre alimentation a une puissance suffisante pour entraîner le moteur pas à pas. C'est généralement une bonne idée d'utiliser une source d'alimentation distincte de celle qui pilote le microcontrôleur. Après avoir câblé une broche GPIO au driver, vous pouvez tester l'interface. Définissez la broche GPIO HIGH et la LED correspondante et le driver s'allumera.

### 3-3-7-Connexions typiques

| Nom du signal | Broche du microcontrôleur | Broche du driver |
|---------------|---------------------------|------------------|
| VDD           | 5v                        | +                |
| GND           | GND                       | -                |
| Gpio1         | PORTA.0                   | In1              |
| Gpio2         | PORTA.1                   | In2              |
| Gpio3         | PORTA.2                   | In3              |
| Gpio4         | PORTA.3                   | In4              |

**Tableau 03:** La connexion des branches entre le driver et l'ARDUINO

## principe de fonctionnement du système d'irrigation

---

Le moteur marche lorsqu'une combinaison spécifique d'entrées est pilotée par le microcontrôleur. Ce n'est qu'une impulsion de puissance, juste assez pour faire marcher le moteur. Ce driver utilise un protocole très simple. Application d'un signal à une entrée broche provoque l'envoi de puissance au moteur sur un fil correspondant.

### 3-3-8-Code de couleurs :

| Les broches d'entrées du driver | La bobine | La couleur du fils de bobine |
|---------------------------------|-----------|------------------------------|
| In1                             | 1         | Bleu                         |
| In2                             | 2         | Rose                         |
| In3                             | 3         | Jaune                        |
| In4                             | 4         | Orange                       |

**Tableau04:** Les couleurs des fils des bobine et la branche driver correspondante

### 4-Conclusion :

Nous avons choisi principalement ces composants pour commander les deux moteurs, car ils étaient facilement disponibles pour nous. Aussi, nous avons pensé qu'il serait amusant d'obtenir un système autonome sophistiqué. Mais comme nous sommes arrivés à réaliser, notre ambition a dépassé notre capacité.



# Chapitre 05

### **Chapitre 05 : la réalisation pratique**

#### **1. Opportunité:**

Il est important de rappeler que l'industrie pétrolière et l'agriculture sont deux segments clés, voire, vitaux pour l'économie Algérienne. C'est pourquoi, nous avons porté notre dévolu sur le second secteur en mettant à contribution notre modeste savoir-faire technologique à travers notre présent mémoire Matériel en notre qualité d'électroniciens.

Il est clair qu'en agriculture, l'irrigation constitue un outil de gestion efficace pour pallier aux irrégularités pluviométriques et rendre ainsi les cultures rentables économiquement.

En plus des fertilisants, le sol joue aussi un rôle prépondérant dans l'irrigation et, la régulation s'explique par un apport d'eau sur un terrain à cultiver à l'effet de compenser judicieusement l'insuffisance des précipitations des eaux de pluie et ainsi, permettre le bon développement des plantes potagères.

L'irrigation agricole est donc une technique artificielle utilisée pour augmenter les rendements en cas de sécheresse. Son prix global diffère selon les modèles agricoles qui la composent et là aussi une étude adéquate s'impose.

Autrement dit, la consommation d'eau nous incite constamment à réfléchir sur l'économie d'eau et de l'énergie qui l'accompagne. Cela passera inéluctablement par une gestion ordonnée, mesurée et efficace de l'arrosage ainsi que la maîtrise de l'utilisation et du choix des systèmes d'irrigation à adopter, d'où l'intérêt des électroniciens d'apporter leurs touches dans le concept d'irrigation par aspersion utilisée en maraîchage. Il s'agit donc, de parcelles de terres irrigables de faibles surfaces.

La technique que nous avons choisi est le système d'irrigation automatique du sol qui repose sur un déversement d'eau uniforme de sorte que chaque plante ait la même quantité d'eau dont elle a besoin et sans excédent.

Parmi les types d'irrigation, nous utiliserons la technique semblable à la couverture intégrale mais mobile qui consiste à arroser le sol tout en réduisant considérablement la consommation d'eau. Cette méthode de couverture intégrale améliorée que nous proposons est constituée d'équipement automatisé.

#### **2.la réalisation du système d'irrigation :**

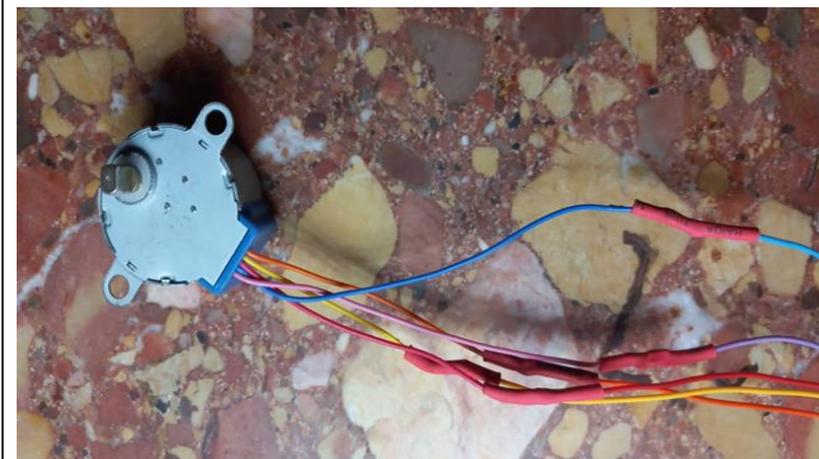
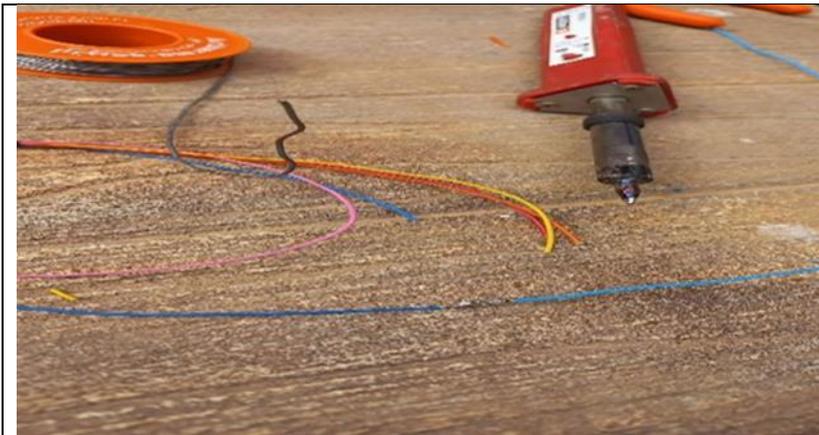
La réalisation d'un système d'irrigation nécessite une conception bien détaillée et précise pour un résultat fiable.

Trois partie distinctes ont été mises en œuvre ; il s'agit de :

## 2.1. Montage du système

A travers le tableau nous allons démontrer les étapes de montage de notre système d'irrigation.

| Composant   | Description du montage  |
|---|---|
|   | <p><b>Etape 1 : Montage de support</b></p> <p>Nous avons fabriqué un support qui contient Ci-dessus un diffuseur d'eau relié à un enroulement transparent pour la circulation d'eau.</p> <p>Le support est basé sur quatre roues, où les deux roues avant sont chacune reliées à un moteur pas à pas.</p> |
|  | <p><b>Etape 2 : installation de la vanne</b></p> <p>Nous avons installé la vanne électrique dans le réservoir afin de contrôler le débit d'eau et l'écoulement pendant l'irrigation.</p>  |



### **Etape 3 : Allongement des fils du moteur**

Nous avons coupé les fils du moteur au milieu afin de les allonger pour s'adapter au projet.

**Tableau 06:** Les étapes de montage du système d'irrigation

## **2.2. Câblage et construction du circuit de puissance**

### **2.2.1. L'alimentation :**

Nous sommes alimenter les deux moteurs pas à pas, brancher sur les deux drivers à partir de la carte ARDUINO avec une alimentation parallèle en utilisant la plaque d'essai :

- Vcc (5v) de la carte ARDUINO le fils blanc, brancher parallèlement avec le fils jaune et le blanc (des deux drivers +5v ) dans le (+) de la plaque d'essai
- GND de la carte ARDUINO, le fils gris brancher parallèlement avec le fils marron et noir (des deux drivers GND) dans le (-) de la plaque d'essai

### Notation :

La carte de contrôle est alimentée par un **adaptateur** (transformateur) qui nous permet d'obtenir une tension continue 5v à partir du 220V alternatif de la prise du secteur.



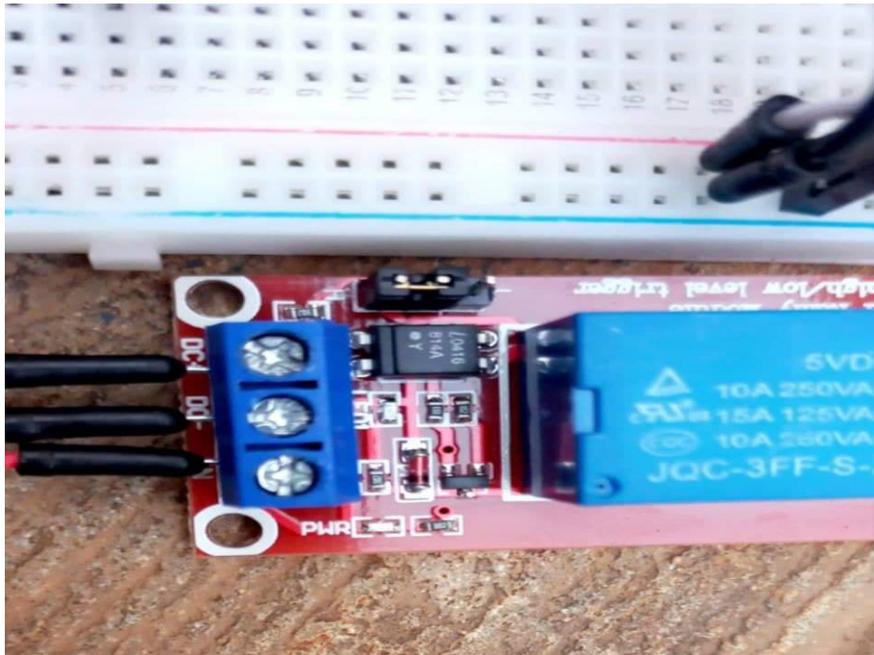
**Figure39** : L'alimentation des deux drivers



**Figure 40:** Les sortie 5v et GND de la carte ARDUINO vers la plaque d'essai et les deux drivers

- l'alimentation du relais :

- DC+ avec le 5v de l'ARDUINO.
- DC- avec le GND de l'ARDUINO.
- NC (normally close) et le commun branché à la vanne.



**Figure 41:** Alimentation du relais de DC+ et DC- vers 5v et GND d'ARDUINO

-L'alimentation de l'électrovanne :

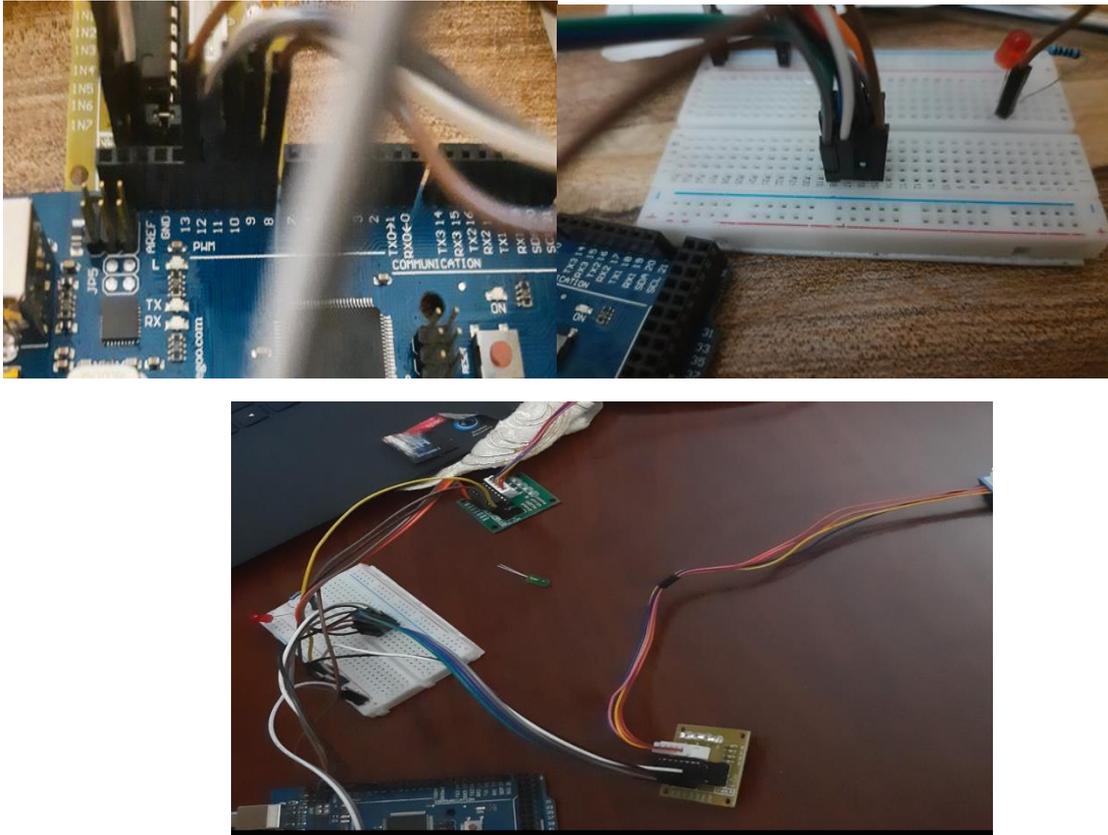
Nous avons branché l'électrovanne à une source de tension de 220 volt alternative.



**Figure 42:** L'alimentation 220 v alternative de l'électrovanne

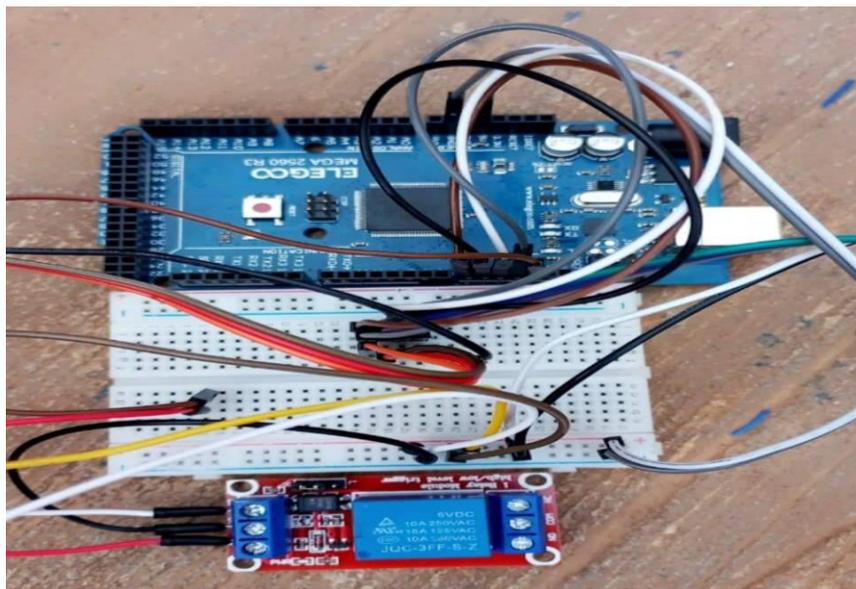
### 2.2.2. Branchement des pins de commande :

On utilise les sorties digitales du microcontrôleur de 8 jusqu'à 11 qui correspond aux pins de ARDUINO en les relie à la plaque d'essai en parallèle (parce que nous avons utilisé 4 pins pour les 8 branche des deux drivers et on les branché en parallèle pour minimiser les connecteur)



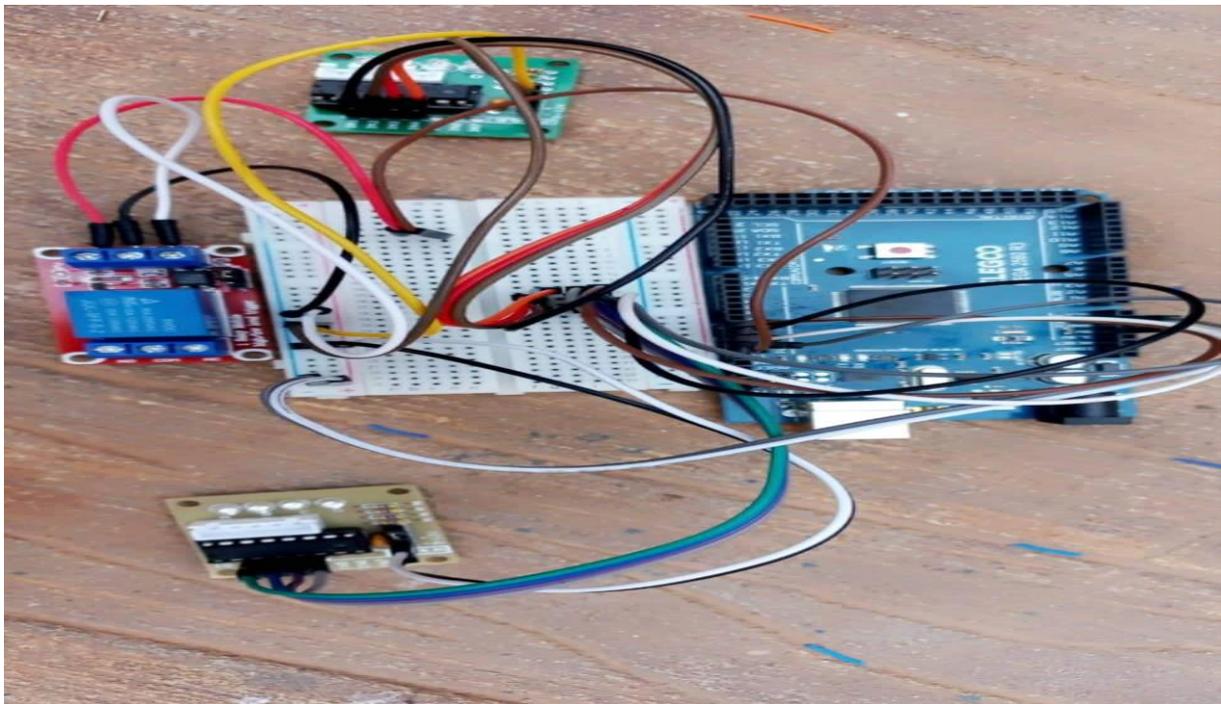
**Figure 43 :** Branchement des pins de commande des moteurs

- Nous avons aussi relié le relais au pin digitale numéro 12 pour commander l'électrovanne à travers ce composant.



**Figure 44 :** branchement du relais avec le pin 12 de l'ARDUINO

- Nous avons soudé et reliés plusieurs bits d'impression et beaucoup des fils entre eux afin d'obtenir un circuit en cours d'exécution adéquat pour contrôler nos bras robotisés.



**Figure 45:** réalisation du circuit de puissance

### 3. structure du programme de commande

#### 3.1. Au niveau de la partie déclarative :

- Nous avons déclaré :

L'instruction `#DEFINE` qui permet au programmeur de donner un nom à une constante

Avant que le programme soit compilé.

```
#define IN1 11
#define IN2 10
#define IN3 9
#define IN4 8
#define relay 12
```

## La réalisation pratique

---

- la variable (NUMBER\_STEPS) de type **Int** (pour **INTEGER**, entier en anglais). Pour le stockage de nombres de pas.

```
int number_steps=512;
```

- déclarer les variables **STEPS** et **DIRECTION** qui montre que le nombre de pas avant l'alimentation et **0** et les moteurs sont en mode arrêt sans aucun pas et aucun sens.

```
int Steps = 0;
int Direction = 0;
```

### 3.2.Au niveau de la fonction d'initialisation setup ( ) :

➤ nous avons:

- identifier la fonction **serial. Begin**, qui Fixe le débit de communication en nombre de caractères par seconde (vitesse de communication entre l'ordinateur et la **carte**).

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

-Identifier la fonction **PINMODE** ,Cette fonction configure les **pins** brancher avec le relais et les driver en mode sortie de tension,

```
pinMode (IN1, OUTPUT);
pinMode (IN2, OUTPUT);
pinMode (IN3, OUTPUT);
pinMode (IN4, OUTPUT);
pinMode (relay, OUTPUT);
```

### 3.3.Au niveau de la boucle principale, la fonction Loop ( ) :

➤ Nous avons utiliser :

1-La fonction **stepper** ( ) Fait tourner le moteur selon nombre de pas indiqué

- Soit Rotation dans le sens antihoraire

```
stepper(number_steps);
```

- Ou Rotation horaire :

```
stepper(-number_steps);
```

2-La structure en boucle for permet d'exécuter un nombre fini de fois, une séquence d'instructions.

```
for(int thisStep=0;thisStep<number_steps;thisStep++)  
    stepper(1);  
  
for(int thisStep=number_steps;thisStep>0;thisStep--)  
    stepper(-1);
```

La présence de parenthèses pour délimiter les conditions de la boucle :

- Variable à itérer et valeur initiale: INT THISSTEP=0, INT THISSTEPS=NUMBER\_NUMBERSTEPS;
- Limiter iteration: THISSTEPS<NUMBER\_NUMBERSTEPS , THISSTEPS>0 ;
- Type d'itération : THISSTEPS++ = incrémentation, THISSTEPS-- =décrémentaion

3- La tâche **DIGITALWRITE** () envoie une valeur **HIGH** (5v) ou **LOW** (0v) au pin digital brancher au relais et préconfiguré comme sortie.

```
//digitalWrite (relay, HIGH) ;  
  
digitalWrite (relay, LOW) ;
```

#### 4. Conclusion :

Dans ce chapitre nous avons fait une étude théorique qui nous a mené à faire une application pratique en passant par :

- le câblage des bloc de puissance ( les drivers , les moteurs ,le relais, la vannes)
- logiciel ARDUINO, nous avons programmé la carte ARDUINO en décrivant les types de programmes implémentés à notre commande .



# Annexes

## Annexes A:

### 1-Programmation pour le système d'irrigation:

```
#define IN1 11
#define IN2 10
#define IN3 9
#define IN4 8
#define relay 12

int Steps = 0;
int Direction = 0;
int number_steps=512;
void setup()
{
  Serial.begin(9600);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(relay, OUTPUT);
  // delay(1000);
}
void loop()
{
  //1 rotation counter clockwise
  stepper(number_steps);
  delay(1000);
  //1 rotation clockwise
  stepper(-number_steps);
```

## Annexes

---

```
delay(1000);
//Keep track of step number
for(int thisStep=0;thisStep<number_steps;thisStep++){
    stepper(1);
}
delay(1000);
for(int thisStep=number_steps;thisStep>0;thisStep--){
    stepper(-1);
}
delay(1000);
}
void stepper(double nbStep){
    if(nbStep>=0){
        Direction=1;
    }else{
        Direction=0;
        nbStep=-nbStep;
    }
    //digitalWrite(relay,HIGH);
    for (int x=0;x<nbStep*8;x++){
        digitalWrite(relay,HIGH);
        switch(Steps){
            case 0:
                digitalWrite(IN1, LOW);
                digitalWrite(IN2, LOW);
                digitalWrite(IN3, LOW);
                digitalWrite(IN4, HIGH);
                break;
            case 1:
```

## Annexes

---

```
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
break;
case 2:
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, LOW);
break;
case 3:
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
break;
case 4:
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
break;
case 5:
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, LOW);
break;
```

## Annexes

---

```
    case 6:
        digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
digitalWrite(IN4, LOW);
break;
case 7:
        digitalWrite(IN1, HIGH);
digitalWrite(IN2, LOW);
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
break;
default:
        digitalWrite(IN1, LOW);
        digitalWrite(IN2, LOW);
        digitalWrite(IN3, LOW);
        digitalWrite(IN4, LOW);
break;

}

digitalWrite(relay,LOW);
delayMicroseconds(1000);
if(Direction==1){ Steps++;}
if(Direction==0){ Steps--;}
if(Steps>7){Steps=0;}
if(Steps<0){Steps=7;}
}
}
```

## Annexe b:

### 1-Microcontrôleur ATEMEGA 2560 :



**Figure 46:** Présentation ATEMEGA 2560

ATMEGA2560 se trouve couramment dans l'ARDUINO MEGA 2560 en tant que microcontrôleur principal. Il s'agit d'un microcontrôleur basé sur AVR RISC qui exécute des instructions puissantes en un seul cycle d'horloge. Cela lui permet de trouver un juste équilibre entre la consommation d'énergie et la vitesse de traitement.

### 2-Fiche technique de ATEMEGA 2560 :

Nom de pièce : atmega2560

Fabricant : ATMEL [ ATMEL CORPORATION ]

REF.FABRICANT : ATEMEGA2560-16AU

Code commande : 1288330



**Figure 47 :** Logo ATEMEGA 2560

### 3-Architecture interne de ATEMEGA 2560 :

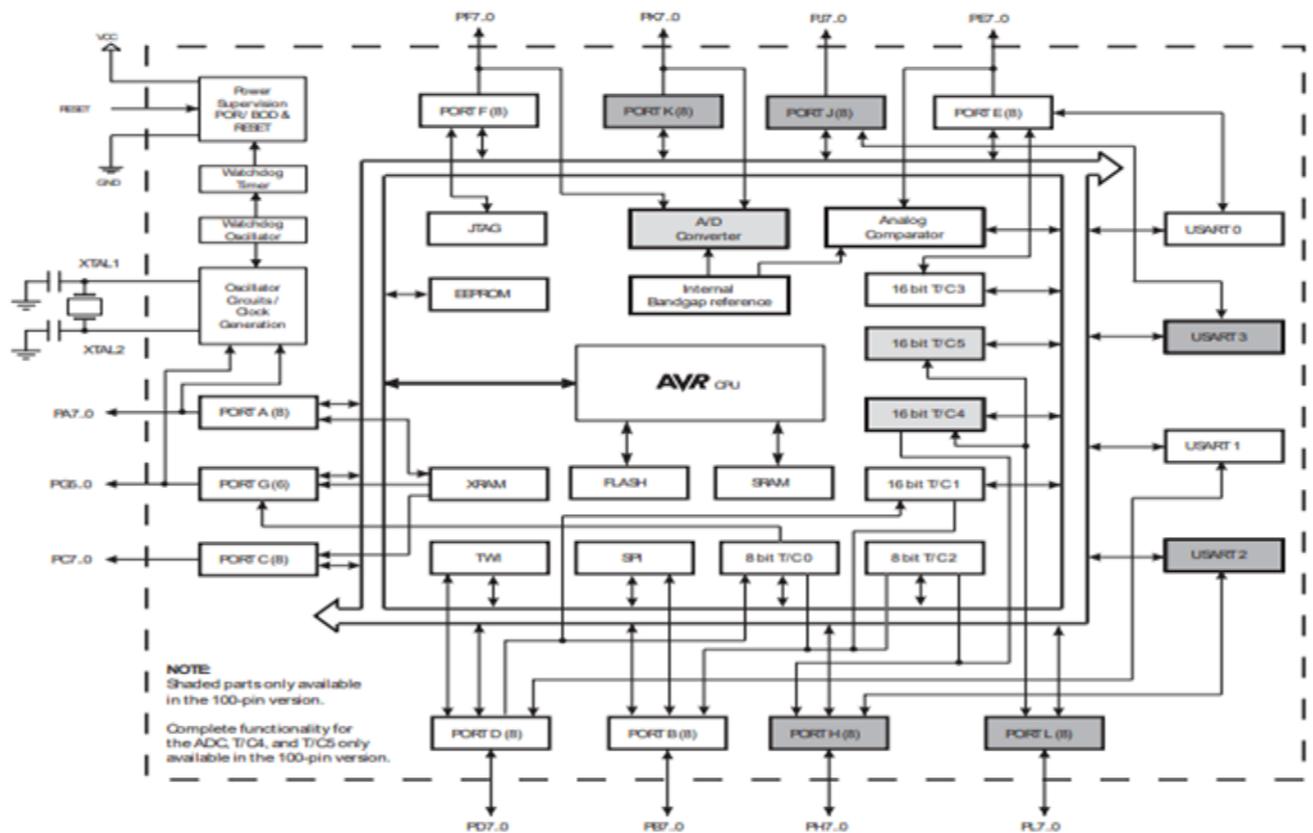


Figure 48 : schématique ATEMEGA 2560

## Annexe c :

### 1.Présentation de la réalisation pratique :



**Figure 49** : Schéma réel de la commande de deux moteur pas à pas pour l'irrigation





# Conclusion générale

### **Conclusion Générale :**

Le travail qui a été effectué dans le cadre de ce mémoire consiste l'irrigation réalisé par la commande de deux moteurs pas à pas à l'aide d'une carte ARDUINO MEGA 2560.

Dans un premier temps nous avons fait une présentation des systèmes embarqués qui sont aujourd'hui au cœur de la convergence électronique, informatique, télécommunication et réseaux.... Les systèmes embarqués nous entourent et nous assistent au quotidien, ils sont dédiés à une tâche bien précise et se distinguent, tant au niveau des applications, qu'au niveau des performances requises.

Ensuite nous avons entamé une étude bien détaillée basée sur les moteurs pas à pas et ces différents types de commande.

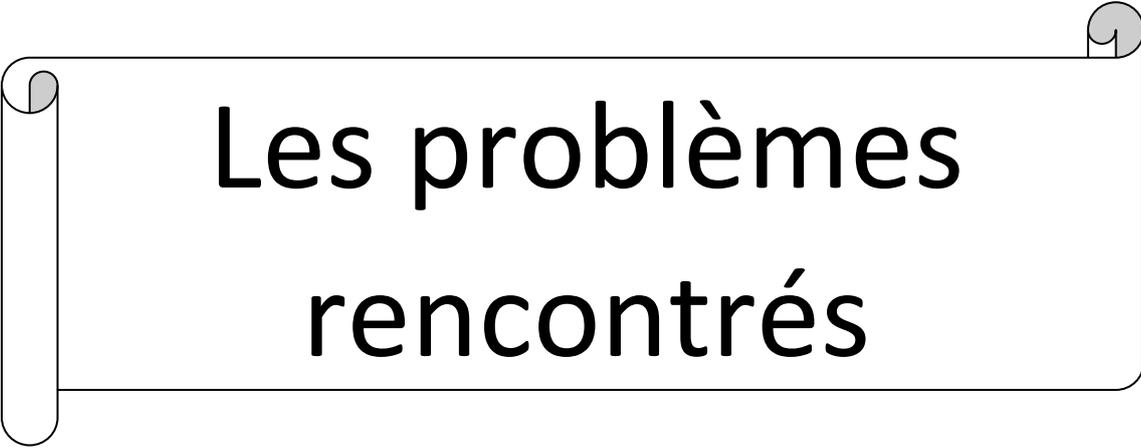
Puis nous nous sommes intéressés à étudier la carte ARDUINO qui est l'élément le plus cruciale. Son apparition a engendré de nombreuses avancées, comme l'augmentation de la vitesse de fonctionnement et de la fiabilité, la réduction de la consommation énergétique....

Nous avons aussi abordé le côté matériel, ou nous avons couvert le matériel de base pour Mettre en relief notre système ainsi que le schéma de principe.

Enfin, dans le dernier chapitre nous avons essayé de rassembler ce qui a été étudié dans les chapitres précédents, Nous avons focalisé notre travail sur la réalisation d'un système d'irrigation, on a pu constater la précision de ces moteurs dans un système de commande de positionnement en boucle ouverte.

Le langage C++ nous a permis de bien gérer le sens de rotation ainsi que la vitesse des deux moteurs, où Nous nous sommes appuyés sur le langage ARDUINO et l'interface IDE effectués au niveau de l'ordinateur et ceci pour former un ensemble homogène qui permettra le bon fonctionnement de notre projet .

A la fin, nous souhaitons que notre travail ait une continuité et il aura envergure et épanouissement dans le domaine d'agriculture à grande surface.



# Les problèmes rencontrés

### **Les Problèmes rencontrés**

Une telle réalisation n'est pas dénuée de difficultés. Il est à noter que nous nous sommes confrontés à plusieurs problèmes surtout dans la partie réalisation du système.

- Ça n'a pas été facile de trouver la matière à utiliser qui soit solide maniable et au même temps légère pour le mouvement des moteurs.
- Le deuxième problème c'est qu'après le cycle d'irrigation l'électrovanne ne se ferme pas au temps de fermeture programmé, pourtant que la vanne est neuve y a pas un soucis d'impuretés par exemple (particule bloqué dans la membrane, problème de nettoyage). Même y a pas de soucis au niveau du relais, on le voit il répond à la commande et même à l'alimentation.



# Résumé

### Résumé

Dans ce travail nous avons conçu un système d'irrigation autonome, il est composé d'un circuit de puissance à base de deux moteurs pas à pas qui sont l'interface idéale entre l'électronique numérique et la mécanique, deux driver et un relais. Ils sont pilotés par un **ATEMEGA 2560** dont le rôle est le traitement des informations, et de donner le résultat des traitements sous forme de rotation du moteur pas à pas pour le déplacement et le positionnement avec grande précision, Et aussi pour la fermeture et l'ouverture de la vanne pour l'écoulement d'eau dans un enroulement d'irrigation.

---

### Abstract

In this work, we have designed an autonomous irrigation system; it is composed of a power circuit based on two stepper motors, which are the ideal interface between digital electronics and mechanics, two drivers and a relay. They are controlled by an **ATEMEGA 2560** whose role is to process the information, and to give the result of the processing in the form of rotation of the stepper motor for displacement and positioning with great precision, and for the closing and opening of the valve for the flow of water in an irrigation coil.

---



# La bibliographie

### La Bibliographiques:

#### Ouvrages :

- 1- FRANCINE KRIEF « Les systèmes embarqués communication, mobilité, sécurité, autonomie » PARIS : HERMES : LAVOISIER 2008
- 2- E. BANGTEGNE, C. BALLE, E. FOURGEAU, PH. OVRAIN, T. CHEVALIER, G. LEBIIZ, E. MITTELETTE ET I. BOUNOURE, "Le livre blanc des systèmes embarqués", SYNTEC INFORMATIQUE, PARIS, FRANCE, 2009
- 3- BOUTEILLE , DANIEL, NOËL , CHANTREUIL , S . , COLLOT , R . , FRACHET , J-P . , LEGRAS , H . , MERLAUD , C . ,SELOSSE , J . , SFAR , a. les automatismes programmables , TOULOUSE , 2<sup>E</sup> ED . , CAPADUES –EDITIONS ,1988,286 P.
- 4- NUSSBAUMER , HENRI . informatique industriel : automate programmable industriel , capteurs , commande numérique , LAUSANNE , PRESSES POLYTECHNIQUE ROMANDES , VOL4 , 1984 , 540 P.
- 5- Sébastien GERARD. Modélisation UML exécutable pour les systèmes embarqués de l'automobile. UNIVERSITE D'EVRY. N° d'ordre : 00EVRY0013. P 12
- 6- FALI JUGURTHA. Etude D'un Système Electronique Embarque Et Réalisation D'une Simulation D'injection Electronique De Carburant. UNIVERSITE MOULOU MAMMERI, TIZI-OUZOU
- 7- JEAN-NOËL MONTAGNE, « Initiation à la mise en OEUVRE matérielle et logicielle de l'ARDUINO » CENTRE DE RESSOURCES ART SENSITIF NOVEMBRE 2006.
- 8- ESKIMON, OLYTE «ARDUINO pour bien commencer en électronique et en programmation »
- 9- TAVERNIER, « ARDUINO applications avancées ».
- 10-ERIK BARTMANN . « Le grand LIVER D'ARDUINO » . 2 EME EDITION , DATE DE PARUTION.
- 11-BERNARD MULTON - Historique des machines électromagnétiques - REVUE 3E.IN°3 - JUIN 1995
- 12-BERNARD MULTON – « Moteurs pas a pas. Structures électromagnétiques et alimentations » - FEVRIER 2006
- 13-Michel ABIGNOLI, Clément GOELDEL – « Moteurs pas à pas » - TECHNIQUES DE L'INGENIEUR - D3 690

#### site web :

- 1- <https://www.synergeek.fr/les-systemes-embarques-dans-automobile/> :04-03-2022
- 2- [www.iut-bm.univ-fcomte.fr](http://www.iut-bm.univ-fcomte.fr) :04 -03-2022

## Bibliographie

---

- 3- [http://lycees.ac-rouen.fr/modeste-leroy/spip/IMG/pdf/\\_Buts\\_de\\_l\\_automatisme.pdf](http://lycees.ac-rouen.fr/modeste-leroy/spip/IMG/pdf/_Buts_de_l_automatisme.pdf) : 04-03-2022.
- 4- [http://sc-st.univ-batna2.dz/sites/default/files/sc\\_st/files/archi\\_des\\_systemes\\_automatisees.pdf](http://sc-st.univ-batna2.dz/sites/default/files/sc_st/files/archi_des_systemes_automatisees.pdf): 20-03-2022
- 5- <https://www.technologuepro.com/cours-automatismes-industriels/chapitre-1-les-automates-programmables-industriels.pdf> : 20-03-2022
- 6- <https://sabtex.files.wordpress.com/2011/11/cours-api-automatique.pdf> :23-03-2022
- 7- <http://www.est-usmba.ac.ma/coursenligne/GE-S2-M8.1-Automatismes%20logiques%20Industriels-CRS-El%20Hammoumi.pdf> : 24-03-2022
- 8- [http://www.techmania.fr/arduino/Decouverte\\_arduino.pdf](http://www.techmania.fr/arduino/Decouverte_arduino.pdf) :17-04-2022
- 9- <http://www-perso.unilim.fr/deneuveille/docs/Info2PC/Cours4.pdf> : 17-04-2022
- 10- [http://www.multimedialab.be/doc/erg/2018-2019/Arduino/Arduino\\_le\\_guide\\_complet\\_JM\\_Hughes\\_O\\_Reilly\\_2018.pdf](http://www.multimedialab.be/doc/erg/2018-2019/Arduino/Arduino_le_guide_complet_JM_Hughes_O_Reilly_2018.pdf) :18-04-2022
- 11- <http://di.univ-blida.dz:8080/jspui/bitstream/123456789/1567/1/021-2016.pdf> :20-04-2022.
- 12- <https://eduscol.education.fr/sti/sites/eduscol.education.fr/sti/files/ressources/pedagogiques/7572/7572-modelisation-dun-moteur-pas-pas-v2.pdf> : 02-05-2022
- 13- [http://les-electroniciens.com/sites/default/files/cours/gpa668\\_pas\\_a\\_pas\\_e2011.pdf](http://les-electroniciens.com/sites/default/files/cours/gpa668_pas_a_pas_e2011.pdf) : 02-05-2022
- 14- [https://edu.ens-rennes.fr/pluginfile.php/1741/mod\\_label/intro/DocPedagogiques/Moteurs-pas-a-pas\\_BM\\_2008.pdf](https://edu.ens-rennes.fr/pluginfile.php/1741/mod_label/intro/DocPedagogiques/Moteurs-pas-a-pas_BM_2008.pdf) : 02-05-2022
- 15- <http://dspace.univ-tlemcen.dz/bitstream/112/11743/1/Ms.ELN.Chigango%2BMapuranga.pdf> : 02-05-2022
- 16- <https://electrotoile.eu/moteur-pas-a-pas-sequence-commande-arduino.php> : 24-05-2022
- 17- <https://www.aranacorp.com/fr/pilotez-un-moteur-pas-a-pas-avec-arduino/> 24-05-2022
- 18- <https://www.elegoo.com/products/elegoo-mega-2560-the-most-complete-starter-kit>
- 19- <https://create.arduino.cc/projecthub/javagoza/elegoo-mega-2650-starter-kit-review-36fa9a>
- 20- <https://www.aranacorp.com/fr/vue-densemble-du-microcontroleur-arduino-mega/>



