الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Centre Universitaire BOUSSOOUF Abdelhafid -Mila

Institut des Sciences et Technologie

Département de Génie Mécanique et Électromécanique

Nº Ref :……

Projet de Fin d'Etude préparé En vue de l'obtention du diplôme de MASTER

Spécialité : Électromécanique

Etude et simulation d'un régulateur de contrôle de niveau d'eau

Réalisé par :
- BENNACER ABDENNOUR
- FERKHA FADI

Soutenu devant le jury :

Dr.	**AOULMIT. S**	**Président**
Dr.	**HADEF. S**	**Examinateur**
Dr.	**GUERFI. N**	**Examinateur**
Dr.	**YESSAD. D**	**Promotrice**

Année universitaire : 2021/2022

Dédicace

À nos chères mères
À nos chers parents
Ceux qui ne se sont jamais arrêtés, priez pour nous, soutenez-nous pour que nous puissions atteindre nos objectifs.
Pour nos sœurs
pour leur soutien moral et leurs précieux conseils
Notre famille
A mon cher binôme pour son sérieux dans ce travail commun
Pour tous nos amis
À tous ceux qui m'ont aidé de près et de loin
Accomplissant ce modeste travail
Tout le monde sans exception.

Remerciements

Avant tout nous remercions Dieu Le tout puissant de nous avoir donné le courage, la volonté, la patience, et la santé durant toutes ces années et que grâce à lui ce travail a pu être réalisé.

Ainsi, nous tenons également à exprimer nos vifs remerciements à notre encadreur D. YESSAD pour avoir d'abord proposé ce thème, pour le suivi tout le long de la réalisation de ce mémoire, et qui n'a pas cessé de nous donner des conseils et des remarques.

Nos vifs remerciements vont également aux membres du jury N.GUERFI, S.AOULMIT et S. Hadaf qui ont accepté d'examiner et de juger notre travail

Nos remerciements vont aussi à tous les enseignants et le chef de département d'Electromécanique H.BENSLIMANE.

Enfin nous tenons à exprimer notre reconnaissance à tous nos amis pour les beaux moments pendant ces cinq années universitaire.

Sommaire

Chapitre IV: Simulation et résultats

Liste des abréviations

EEPROM: Electrically-Erasable Programmable Read-Only Memory

FTDI : Future Technology Devices International

GND: Ground

Hex: Hexadecimal

ICSP: In Circuit Serial Programming

IDE: Integrated development environment

LCD: Liquid Crystal Display

MLI : Modulation de largeur d'impulsion

PID : Proportionnel Intégrale Dérivée

PWM : Pulse Width Modulation

SRAM: Static Random Access Memory

TOR : Régulation tout ou rien

USB: Universal Serial Bus

VCC : Common Collector Voltage (Alimentation Tension Continue)

Liste des figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Liste des tableaux

Chapitre III

Introduction générale

Introduction générale

La gestion des systèmes de stockage de l'eau est importante dans de nombreux secteurs, tels que l'agriculture et les usines de fabrication industrielles et agroalimentaires. Un aspect important à prendre en compte est le niveau d'eau dans le réservoir de stockage utilisé dans les systèmes d'irrigation ou de fabrication. Le plus important sera de contrôler le niveau d'eau afin de répondre aux besoins propres du consommateur. Pour se faire, des nombreuses études et techniques de contrôle automatique ont été menées. Les techniques d'acquisition de données ont considérablement évolue en terme de rapidité, fiabilité et précision, en premier lieu, les capteurs qui bénéficient des nouvelles technologies, sont de plus en plus performants, en second lieu l'avènement des ordinateurs et microcontrôleurs qui, en permettant d'automatiser et de numériser des systèmes, offrant des gains important pour ce qui est de leurs traitement de l'information. La commande numérique est une technique utilisant des données numériques pour représenter des instructions à la conduite d'une machine ou d'un procédé, d'une autre manière, elle est conçue pour piloter le fonctionnement d'une machine à partir des instructions d'un programme.

Notre projet de fin d'étude consiste à concevoir et à simuler un système automatique de contrôle et la régulation de niveau d'eau d'un réservoir. Pour ce faire notre système de contrôle est construit autour d'un microcontrôleur Arduino Mega 2560 qui présente un nombre important de ports d'entrée sortie et un espace mémoire très suffisant. D'une autre manière, on s'intéresse à piloter le fonctionnement d'une moteur à courant continu (pompe) en exploitant de la carte, donc il s'agit de réaliser un système électronique qui commande la marche et l'arrêt des pompes de manière à automatiser les opérations de pompage et de remplissage. Notre travail est présenté dans un mémoire organisé en quatre chapitres :

Le premier chapitre, est une introduction à la régulation, ainsi que la structure d'un système de régulation et les performances exigées.

Dans le deuxième chapitre, nous avons présenté les régulateurs industriels tels que le régulateur TOR, PWM et PID avec ses différentes structures.

Le troisième chapitre sera consacré à la carte Arduino Mega, la partie logicielle, ainsi que une présentation de composants utilisés. La simulation du système et les résultats fait l'objet du dernier chapitre.

Chapitre I

Introduction à la régulation

I.1 Introduction

L'automatique est généralement définie comme la science qui traite des ensembles qui se suffisent à eux-mêmes et où l'intervention humaine est limitée à l'alimentation en énergie et en matière première. L'objectif de cette science est de remplacer l'homme dans la plupart des tâches. Le domaine des applications de l'automatique est très vaste et varié, mais l'observation de l'industrie contemporaine conduit à une certaine classification qui se résume en deux grandes familles selon les données que traitent ces systèmes : l'automatisme séquentiel et l'asservissement, cette dernière se décompose en deux autres sous branches qui sont le système asservi et la régulation.

Dans ce chapitre, nous décrivons la notion de base d'un système, avec son comportement en deux modes asservissement et régulation. Ensuite, nous énonçons les principaux éléments de la Chaine de régulation, ainsi que objectif globale cette science. Nous terminons par la présentation les déférentes performances de système de régulation.

I.2 Notion de base de système

Un système est un ensemble de processus, physique et chimique, en évolution. Des actions sur le système (entrées) sont effectuées dans le but d'obtenir des réponses souhaitées (sorties).Les signaux qui véhiculent dans un système sont de deux types :

- Signaux d'entrées : Ils sont indépendants du système et peuvent être commandés (consignes) ou non commandés (perturbations).
- Signaux de sorties : Ils sont dépendants du système et du signal d'entrée. Pour évaluer les objectifs, ces signaux doivent être observables moyennant des capteurs.

La figure I.1 illustre un système à une entrée de commande, une sortie et une entrée de perturbation [1].

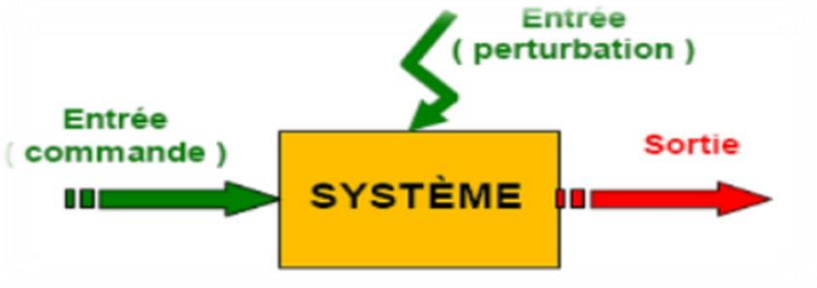

Figure I.1 : Les différents signaux relatifs à un système automatique.

I.2.1 Comportement de système automatique

Le contrôle d'un système automatique ne peut être optimal que si l'on gère correctement son asservissement ou sa régulation [2].

I.2.1.1 Comportement en mode asservissement

Lorsque la consigne d'un système asservi dépend du temps, on parle d'asservissement et de poursuite. Par exemple l'asservissement en position d'une parabole d'un radar de contrôle aérien. Donc en mode asservissement, un changement de la valeur de la consigne correspond à une modification du point de fonctionnement du processus. Si le comportement en asservissement est correct, les performances dynamiques de la boucle de régulation sont conservé, même lorsqu'une perturbation se produit.

I.2.1.2 Comportement en mode régulation

Lorsque la consigne d'un système asservi est indépendante du temps, on parle de régulation. Comme exemple la régulation de température dans un local subissant les variations climatiques ou bien la régulation de PH de rejets d'eau destinés à être déversés dans une rivière. En mode régulation la consigne est maintenue constante sous l'effet d'une modification ou une variation des entrées perturbatrices. L'aspect régulation est l'élément le plus important dans le milieu industriel, car les valeurs des consignes sont souvent fixes.

I.3 Historique de la régulation

L'histoire des automates est intimement liée à celle de l'horlogerie. Afin d'améliorer le principe de l'horloge à eau le physicien grec Ctésibios perfectionne la clepsydre pour améliorer sa fiabilité, il introduit un réservoir supplémentaire dans lequel le volume de liquide reste constant grâce à un mécanisme qui ferme l'entrée du réservoir lorsque celui-ci est trop plein. Donc Il conçoit une horloge hydraulique pouvant fonctionner indéfiniment sans intervention humaine. Le liquide, en s'écoulant, fait tourner des roues qui permettent à une petite statue de monter, en indiquant le passage des heures. Les heures sont tracées soit sur une colonne ou un pilastre. Une figurine sort du bas de la machine et les indique avec une baguette pour toute la durée du jour [3]. Ce system est le premier automate hydraulique qui est considéré comme étant la première réalisation d'un correcteur automatique

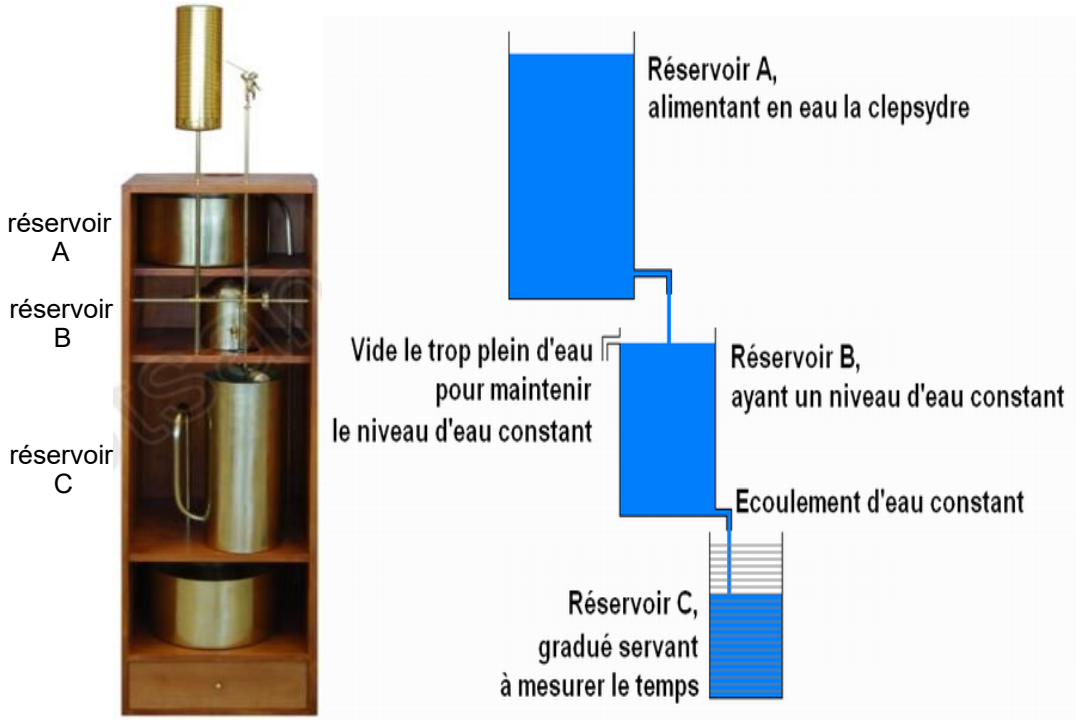

Figure I.2 : Clepsydre de Ctesibios [3].

Les développements de l'horloge conduisent alors à la création d'automates extrêmement compliqués. Citons, à titre d'exemple, le canard de Jacques Vaucanson (1709-1782) qui pouvait boire, se nourrir, caqueter, nager, "digérer" et déféquer [4]. L'idée de la programmation est née à peu près à la même époque. En 1728, Jean- Philippe Falcon crée le premier métier à tisser programmable par cartons perforés. Vaucanson réalise un métier à tisser programme en 1745 [5], et en 1801 le célèbre métier à tisser automatique programmable de Joseph-Marie Jacquard. A la même époque d'autres systèmes voient le jour, poussés par la révolution industrielle qui est en marche. Denis Papin développe la soupape de sécurité pour les systèmes fonctionnant à la vapeur et James Watt invente le régulateur de vitesse à boule. Les réalisations précédentes ne manquent pas d'ingéniosité, néanmoins il faudra attendre l'arrivée de l'électricité puis de l'électronique pour voir apparaitre les premiers correcteurs en tant que tels. Enfin, le développement des microprocesseurs puis celui des microcontrôleurs permet aujourd'hui de mettre en œuvre des principes de commande très élaborés.

I.4 Chaine de la régulation

Dans la plupart des appareils et installations industrielles ou domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Le niveau d'un réservoir d'eau, la température d'une étuve, le débit d'une conduite de gaz, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage (dit en boucle ouverte) permet d'obtenir et de maintenir la valeur demandée (par exemple une action sur un robinet d'eau). Dans la majorité des cas ce type de réglage n'est pas suffisant, parce qu'il est trop grossier ou instable. Il faut alors comparer, en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglant. On a, dans ce cas, a constitué une boucle de régulation et plus généralement une boucle d'asservissement. Cette boucle nécessite la mise en œuvre d'un ensemble de moyens de mesure, de traitement de signal ou de calcul, d'amplification et de commande d'actionneur, constituant une chaine de régulation ou d'asservissement [6]. La figure I.3 présente le schéma principal d'une chaine de régulation.

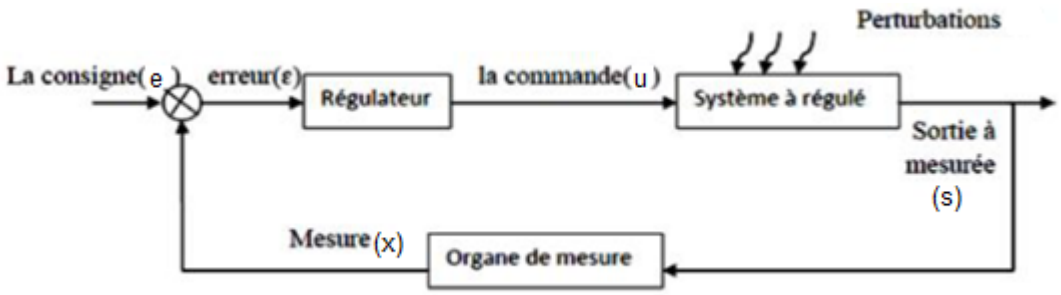

Figure I.3 : Schéma principal de la chaine de régulation.

Toute chaîne soit de régulation ou d'asservissement comprend trois maillons indispensables : l'organe de mesure, l'organe de régulation et l'organe de contrôle. Il faut donc commencer par mesurer les principales grandeurs servant à contrôler le processus. L'organe de régulation récupère ces mesures et les compare aux valeurs souhaitées, plus communément appelées valeurs de consigne. En cas de non-concordance des valeurs de mesure et des valeurs de consigne, l'organe de régulation envoie un signal de

commande à l'organe de contrôle (vanne, moteur, etc.), afin que celui-ci agisse sur le processus. Le choix des éléments de la chaîne de régulation est dicté par les caractéristiques du processus à contrôler, ce qui nécessite de bien connaître le processus en question et son comportement [6].

I.4.1 Structures de commandes

On dit qu'un système physique est commandé lorsqu'il permet de réaliser une relation entre la grandeur d'entrée (cause) et la grandeur de sortie (effet). Quel que soit la nature du système à commander, il est toujours possible de classer les différentes structures de commande en deux grandes familles. Les structures de commande en boucle ouverte et les structures de commande à contre-réaction, appelées structures de commande en boucle fermée [6].

I.4.1.1 Commande en boucle ouverte

En l'absence d'entrées perturbatrices et en supposant que le modèle mathématique du système est parfait, il est concevable de générer un signal de commande produisant le signal de sortie souhaité (Figure I.4). Cela constitue le principe de la commande en boucle ouverte qui exploite la connaissance des dynamiques du système afin de générer les entrées adéquates *e(t)*. Ces derniers ne sont donc pas influencés par la connaissance des signaux de sortie *s(t)*. Cette solution est envisageable dans le cas où le système est parfaitement connu.

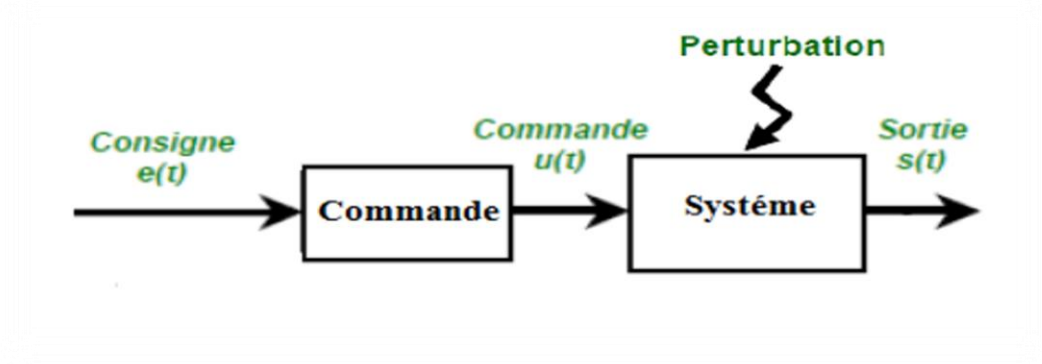

Figure I.4 : Schéma générale d'une commande en boucle ouverte.

Le système en boucle ouvert est insensible aux perturbations, rapide et stable. Ne revanche c'est un système, souvent, qualifié par aveugle, puisque c'est impossible de concevoir une correction [6].

I.4.1.2 Commande en boucle fermée

Si le système à commander n'est pas parfaitement connu ou si les perturbations affectent sensiblement le système, alors les signaux de sortie ne seront plus ceux souhaités. L'introduction d'un retour d'information sur la sortie mesurée s'avère alors nécessaire. Un système bouclé permet, en quelque sorte, que la réponse du système correspond à l'entrée de référence. Tandis qu'un système en boucle ouverte commande sans contrôler l'effet de son action. Les systèmes de commande en boucle fermée sont ainsi préférables quand des perturbations non modélisables et/ou des variations imprévisibles des paramètres sont éventuelles. Cette structure de commande permet ainsi d'améliorer les performances dynamiques du système commandé : rapidité, rejet de perturbation, meilleur suivi des consignes, moindre sensibilité aux variations paramétriques du modèle, stabilisation des systèmes instables en boucle ouverte [6].

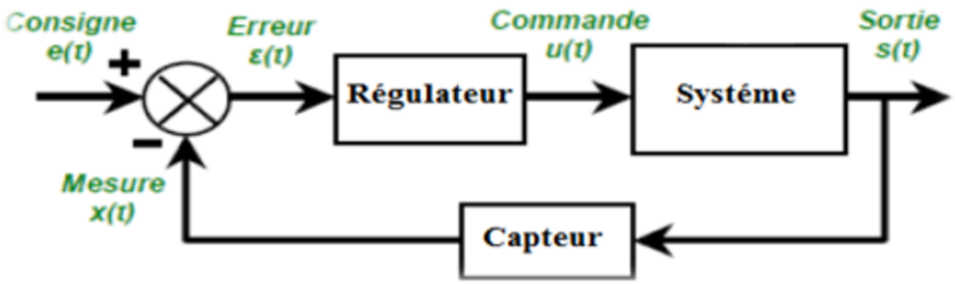

Figure I.5 : Schéma générale d'une commande en boucle fermée.

Un system en boucle fermée s'agit d'un système précis, il y a une correction (sensible aux perturbations), mais peut rendre le système lent et instable.

I.5 Objectif globale de la régulation

Peut se résumer par ces trois mots clés : mesurer, comparer, corriger. Cependant, chaque procédé possède ses exigences propres, chaque appareil possède ses propres conditions de fonctionnement. Il est donc indispensable que la régulation soit conçue pour satisfaire aux besoins particuliers liés à la sécurité, aux impératifs de production et aux matériels. La régulation est l'action de régler automatiquement une grandeur de telle sorte que celle-ci garde constamment sa valeur ou reste proche de la valeur désirée, quelles que soient les perturbations qui peuvent subvenir [7].

I.6 Performances de système de régulation

Les performances et les qualités exigées les plus rencontrées industriellement sont la stabilité, la précision et la rapidité de la grandeur à réguler.

I.6.1 Stabilité

La qualité essentielle pour un système de régulation est la stabilité. Dans une approche simplifiée, un système est considéré comme stable si, pour une grandeur à maîtriser le signale de sortie se stabilise à une valeur finie. Plus le régime transitoire d'un système soumis à une telle variation est amorti plus il est stable. Le degré de stabilité est alors caractérisé par l'amortissement de ce régime transitoire [7].

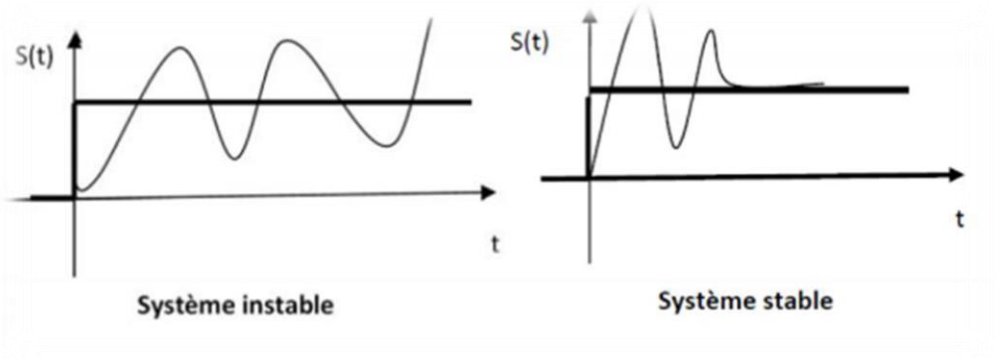

Figure I.6 : Stabilité du système**.**

I.6.2 Rapidité

La rapidité d'un système régulé s'évalue par le temps nécessaire à la mesure pour entrer dans une zone ± 5% de sa valeur finale Le système régulé est d'autant plus rapide que le temps de réponse à 5% est court. [7].

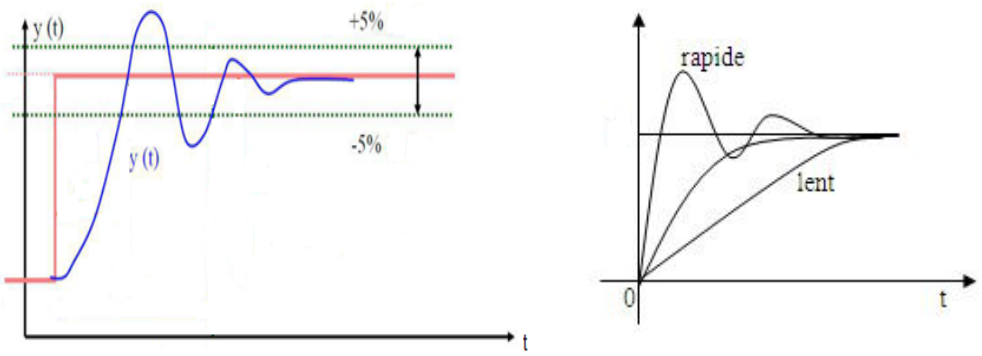

Figure I.7 : Rapidité du système

I.6.3 Précision

La précision d'un système régulé se mesure par l'écart entre la consigne demandé et la mesure en régime permanant de la grandeur réglée, on parle alors de précision statique. Plus l'écart statique est petit, plus le système est précis [7].

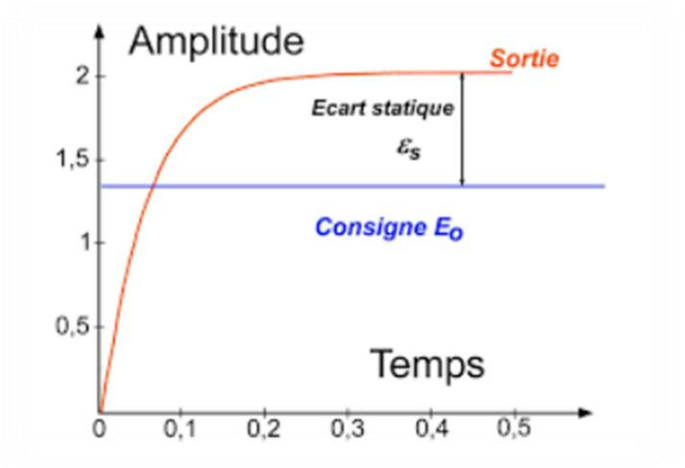

Figure I.8 : Précision d'un système.

I.7 Conclusion

Ce chapitre est une introduction au domaine de la régulation. Il présente les différentes tâches liées aux systèmes de régulation, un bref historique sur la régulation, le schéma principal de la chaine de régulation, l'objectif puis les performances de système de régulation.

Chapitre II

Régulation industrielle

II.1 Introduction

La régulation industrielle occupe une place importante dans le mode moderne, en raison des performances de plus en plus élevées que l'on réclame des commandes automatiques. Pour un technicien ou un ingénieur de régulation dans l'industrie le terme procédé désigne une partie ou un élément d'une unité de production industrielle ; par exemple un échangeur thermique qui comporte une régulation de température ou un réservoir dont le niveau est régulé. Procédé et régulation forment un tout indissociable.

Le choix du type de boucle de régulation et son élaboration impliquent une bonne compréhension du comportement du procédé. Le niveau du réservoir ou de la température sortie échangeur présentent-ils une grande inertie ? Sont –ils stables ou instables ?

Dans ce chapitre, nous décrivons les types de la régulation industrielle, d'abord la régulation TOR, ensuit la régulation MLI, enfin la régulation avec PID.

II.2 Types de la régulation industrielle

La régulation industrielle regroupe l'ensemble des techniques utilisées visant à maintenir constante une grandeur physique appelée grandeur réglée (sortie du système) : A une valeur désirée appelée consigne (entrée), soumise à des perturbations, en agissant sur une grandeur physique appelée grandeur réglant (commande). Pour un minimum d'écart possible (précision), le plus rapidement possible (économie d'énergie), sans déstabilisé la réponse (qualité du produit). Il y a plusieurs types de régulation industrielle, nous citons principalement :

- La régulation TOR : Tout Ou Rien (ON–OFF Control).
- La régulation MLI : Modulation de la Largeur d'Impulsion (PWM) avec PID.
- La régulation PID : Continue Proportionnelle, Intégrateur, Dérivateur.
- La régulation FUZZY : Continue PID avec l'introduction de la logique floue.
- La régulation MULTI-BOUCLES : Maître Esclave, Cascade, Rapport.
- La régulation NUMERIQUE : Modèle de référence [8].

II.3 Régulation TOR (tout ou rien)

Un régulateur « tout ou rien » est un régulateur qui élabore une action de commande discontinue qui prend deux positions ou deux états 0 et 1 (ou 0 et 100%). On les appelle: on-off control ou two steps controller [9].

II.3.1 Domaine d'utilisation

Les régulateurs tout ou rien sont utilisés pour la commande des systèmes ayant une grande inertie où la précision de régulation n'est pas importante. A titre d'exemple la régulation de niveau d'un réservoir.

II.3.2 Fonctionnement d'un régulateur «tout ou rien»

Dans ce type de régulateur, la commande du correcteur agit sur un relais électromécanique à contact. Dans le cas simple, lorsque la commande est en position 1, une bobine est excitée et ferme le contact du relais pour alimenter la résistance de chauffe et est désexcitée lorsque la commande est en position 0 (le contact s'ouvre alors). Les régulateurs tout ou rien classiques sont par exemple les thermostats et les soupapes de sécurité (pressostats) qu'on utilise dans les systèmes de sécurité [9].

II.3.3 Action discontinue

Une action discontinue, dans laquelle la sortie Y du régulateur ne prend que deux valeurs. On appelle aussi le fonctionnement discontinu fonctionnement Tout Ou Rien.

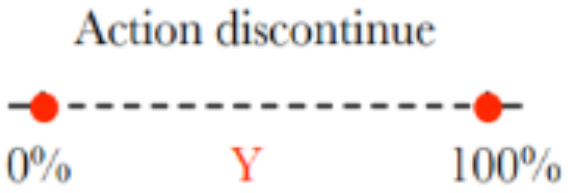

Figure II.1 : Schéma présente l'action discontinue.

II.4 Régulation MLI

Quand on veut faire varier la vitesse d'un moteur, la première idée qui vient à l'esprit est de faire varier la tension aux bornes du moteur. Pour faire varier la vitesse de rotation à tension fixe, on utilise des signaux de la modulation de largeur d'impulsions (MLI ; en anglais : Pulse Width Modulation, soit PWM).

La PWM est une technique couramment utilisée pour synthétiser des signaux pseudo analogiques à l'aide de circuits numériques (tout ou rien, 1 ou 0), ou plus généralement à états discrets. Elle sert à générer un signal pseudo analogique à partir d'un environnement numérique ou analogique pour permettre un traitement de ce signal par des composants en commutation (se comportant comme des interrupteurs ouverts ou fermés).

La PWM est signal carré disponible uniquement pour un train d'impulsions. Donc la tension peut prendre deux valeurs seulement. Le niveau bas correspond généralement à 0 Volt. La période est notée T ; la durée de l'impulsion (pour laquelle la tension est celle de l'état haut) est appelée th [10].

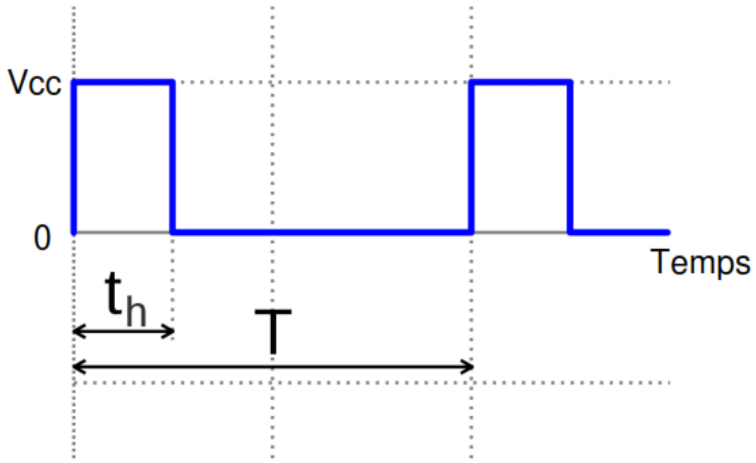

Figure II.2 : La forme du signal MLI [10].

Si la période change, le signal n'est plus vraiment périodique au sens strict. On appelle alors T le pseudo période.

II.4.1 Fréquence

La commande d'actionneurs de puissance par PWM est très liée à la notion de fréquence. Pour que l'impression d'une valeur moyenne constante d'allumage apparaisse, il faut que l'alternance d'allumage/extinction soit suffisamment rapide pour qu'elle ne se

remarque pas. Si par exemple le cycle complet de PWM durait une seconde (cette durée est nommée période), ce qui donne une fréquence de 1 Hertz, les durées d'allumage et d'extinction de l'actionneur seraient réparties proportionnellement sur cette seconde. Imaginons une lampe halogène allumée à 40% de sa puissance. Elle doit être alimentée 40% du temps et éteinte durant 60 % du temps. Avec une fréquence de 1 Hz, elle serait allumée pendant 0,4 seconde puis éteinte pendant 0,6 seconde. L'effet de clignotement résultant est parfaitement perceptible... La fréquence du PWM doit donc être beaucoup plus grande que 1 Hz, ou autrement dit la période doit être bien plus courte qu'une seconde. Selon les utilisations la fréquence de PWM va de 100 Hz (100 cycles par seconde) à 200 kHz. Nos cartes de commande permettent actuellement d'avoir deux fréquences de PWM : 100 Hz et 400 Hz [11].

II.4.2 Rapport cyclique

On appelle rapport cyclique le rapport :

$$\alpha = 100 \times \frac{th}{T}$$; exprimé en pourcentage.

Si th =0 alors α=0% et la tension moyenne de sortie est nulle.

Si th=T alors α=100% et la tension moyenne de sortie est égale à Vcc.

II.5 Régulation PID

Le régulateur PID est considéré comme un régulateur standard car c'est le plus utilisé dans l'industrie (proportionnel intégral dérivé), permettant d'assurer à l'aide de ses trois paramètres les performances souhaitées (amortissement, temps de réponse, …) d'un processus modélisé par un deuxième ordre. Nombreux sont les systèmes physiques qui, même en étant complexes, ont un comportement voisin de celui d'un deuxième ordre. Par conséquent, le régulateur PID est bien adapté à la plupart des processus de type industriel et est relativement robuste par rapport aux variations des paramètres du procédé. Si la dynamique dominante du système est supérieure à un deuxième ordre, ou si le système contient un retard important ou plusieurs modes oscillants, le régulateur PID n'est plus adéquat et un régulateur plus complexe (avec plus de paramètres) doit être utilisé, au dépend de la sensibilité aux variations des paramètres du procédé [12].

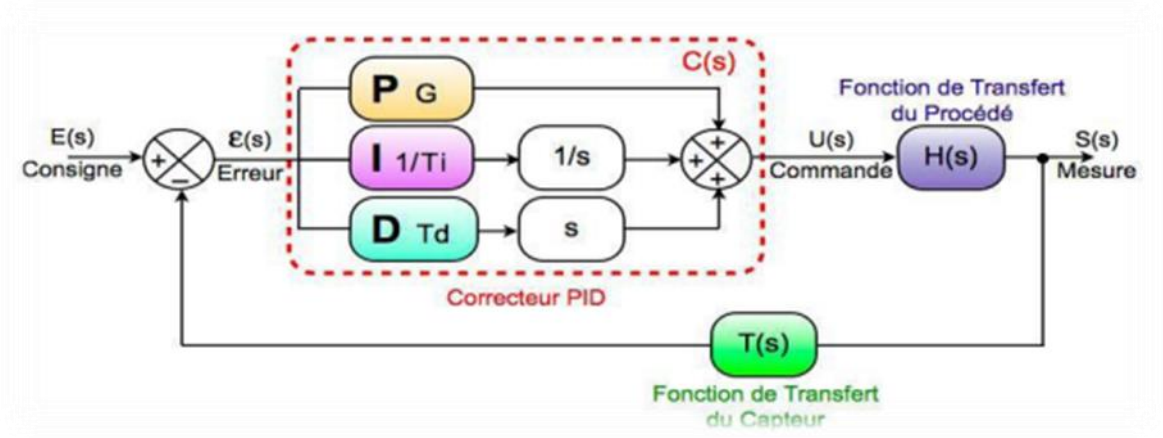

Figure II.3 : Correcteur PID.

Le réglage d'un PID est en général assez complexe, des méthodes pratiques de réglages permettent d'obtenir des bons résultats [13]. L'avantage d'un régulateur PID est sa performance dynamique, sa précision de réglage et sa stabilité. Les éléments de ce régulateur sont une combinaison d'actions P (Proportionnelle), I (Intégrale) et D (Dérivée) choisies en fonction du type d'application.

II.5.1 Action proportionnelle (P)

La commande de type 'Proportionnelle' est la plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée tout écart de la grandeur a réglé.

Son rôle est annulé l'erreur pour que le système réagisse plus rapide, comme si l'erreur était plus grande qu'elle ne l'est en réalité [14].

Elle agit donc principalement sur le gain du système asservi et permet:

- D'entraîner une augmentation du gain, d'où une diminution de l'erreur statique (amélioration de la précision).
- D'améliorer la rapidité du système.
- D'augmenter l'instabilité du système (donne lieu à des oscillations) [15].

II.5.2 Action intégrale (I)

L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul [16].

L'intérêt principal de ce correcteur est d'ajouter dans la chaîne de commande une intégration qui augmente la classe du système et annule, selon le type d'entrée, l'erreur

statique du système.

L'action intégrale pure permet :

- D’améliorer la précision en annulant l'erreur statique.
- D’introduire un déphasage de -90°qui risque de déstabiliser le système (diminution de la marge de phase).

Le régulateur à action exclusivement intégrale n'est pratiquement jamais utilisé, en raison de sa lenteur et des on effet déstabilisant. Il est en général, associé au correcteur Proportionnel [15].

II.5.3 Action dérivée (D)

C’est une action qui tient compte de la vitesse de variation de l’écart entre la consigne et la mesure, elle joue aussi un rôle stabilisateur [16].

Cette action permet :

- D’améliore la stabilité du système par l'introduction d'un déphasage supplémentaire de +90° (augmentation de la marge de phase).
- De faire diminuer la précision du système, et amplifie les bruits de hautes fréquences.
- D’augment la rapidité du système (diminution des temps de réponses) [15].

II.6 Conclusion

Ce chapitre est un avis global sur la régulation industrielle. Il présente les différentes types de régulation qui sont ; la TOR, MLI et PID avec ses différentes versions Proportionnelle, Intégrateur, Dérivateur et le PID FUZZY.

Chapitre III

Carte Arduino Mega 2560 et actionneurs

III.1 Introduction

Chaque projet ou mini-projet doit être complété par un plan pratique contenant plusieurs de ses composants électriques, mécaniques et électroniques. Ces éléments sont divisés en fonction de leur rôle principale ou secondaire, dans notre projet l'arduino et le cerveau qui donne les ordres, le moteur est l'élément épuisé, le capteur ultrason est élément intermédiaire et les autres éléments supplémentaires en fonction de leurs rôles respectifs comme la protection, la source ainsi de suite.

III.2 Carte Arduino

Arduino est un projet électronique qui à vue le jour en Italie en 2005, il a été créé et perfectionné par une équipe de développeurs, composée de six individus:(Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti). C'est une équipe d'enseignants et d'étudiants de l'école de Design d'Interaction d'Ivrea. Le nom Arduino est un nom d'un roi italien, personnage historique de la ville « Arduin d'Ivrée », ou encore un prénom italien masculin qui signifie « l'ami fort ». La carte Arduino est un bon outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes [17].

Le module Arduino est un circuit imprimé en matériel libre, les plans de la carte sont publiés en licence libre dont certains composants de la carte, comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications différents comme les projets de fin d'étude, l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartez 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE (integrated development environment) Arduino. Les projets Arduino peuvent être autonomes, comme ils peuvent communiquer avec d'autres logiciels installés sur l'ordinateur tel que Proteus, Matlab, car ces cartes sont faites à base d'une interface entrée/sortie simple et d'un environnement de développement proche du langage de programmation [18] [19].

III.2.1 Arduino Mega 2560

La carte Arduino Mega 2560 est une carte à microcontrôleur basée sur un ATmega2560, pour pouvoir l'utiliser, il suffit simplement de la connecter à un ordinateur à l'aide d'un câble USB, elle dispose de [20] :

- 54 broches numériques d'entrées/sorties (dont 14 peuvent être utilisées en sorties MLI (Modulation de largeur d'impulsion, en anglais : Pulse Width Modulation, soit PWM);
- 16 entrées analogiques qui peuvent être utilisées en broches entrées/sorties numériques ;
- 4 UART port série matériel ;
- Un quartz de 16Mhz ;
- Une connexion USB ;
- Un connecteur d'alimentation jack ;
- Un connecteur ICSP pour la programmation "in-circuit" ;
- Un bouton de réinitialisation (reset).

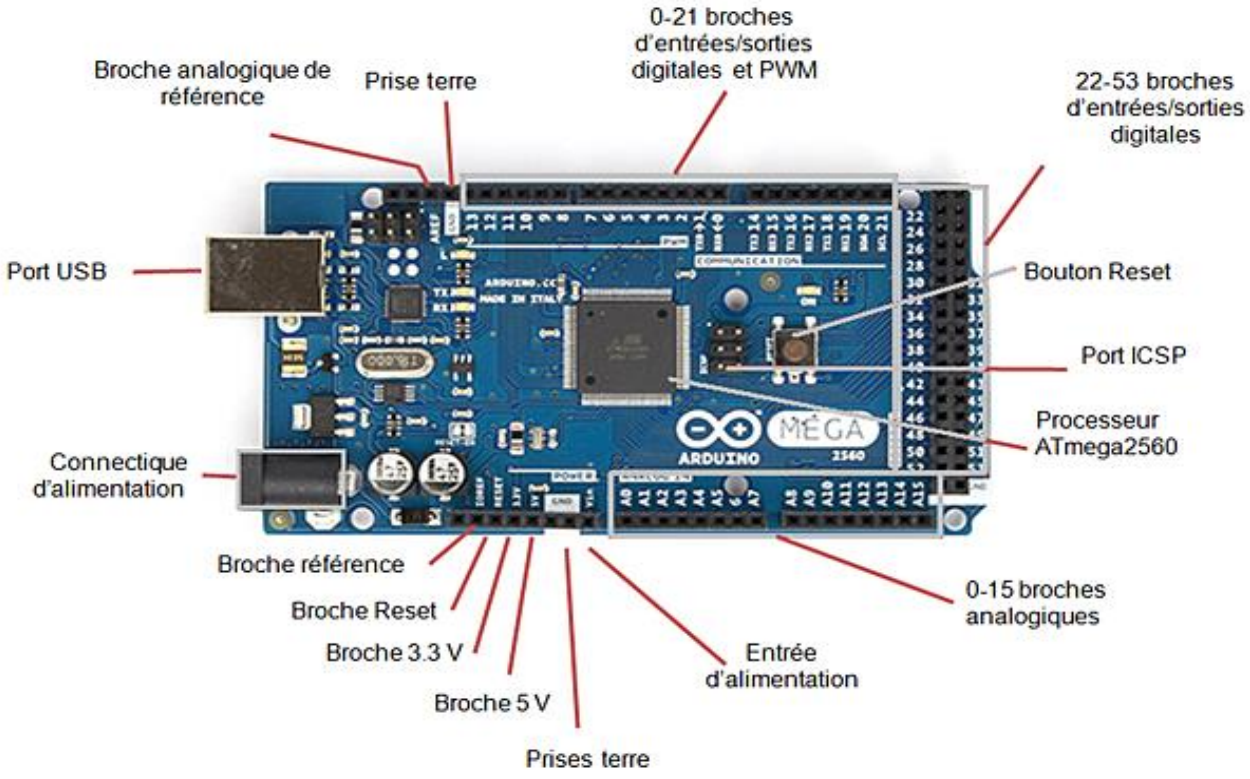

Figure III.1 : Description des entrées/sorties de la carte Arduino Mega2560.

La synthèse des caractéristiques de l'Arduino Mega2560 est présentée dans le tableau III.1.

Tableau III.1 : Caractéristiques de l'Arduino Mega 2560.

Caractéristique	Valeur
Microcontrôleur	Atmega2560
Tension de fonctionnement	5V
Tension d'alimentation (recommandée)	7-12V
Tension d'alimentation (limites)	6-20V
Broches E/S numériques	54 (dont 14 disposent d'une sortie PWM)
Broches d'entrées analogiques	16 (utilisables en broches E/S numériques)
Intensité maxi disponible par broche E/S (5V)	40 mA
Intensité maxi disponible pour la sortie 3.3V	50 mA
Intensité maxi disponible pour la sortie 5V	Fonction de l'alimentation utilisée –500 mA max si le port USB est utilisé seul
Mémoire Programme Flash	256 KB dont 8 KB utilisé par bootloader
Mémoire SRAM (mémoire volatile)	8 KB (Atmega2560)
Mémoire EEPROM (mémoire non volatile)	4 KB (Atmega2560)
Vitesse d'horloge	16 MHz

III.2.1.1 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 qui s'occupe de tout ce qui est calculs, comme l'exécution des instructions du programme et gestion des ports d'entrée/sortie [21].

Figure III.2 : Microcontrôleur ATMega2560.

Les différentes caractéristiques du microcontrôleur ATMega2560 sont :

- 135 instructions puissantes, la plupart d'exécution simple de rythme ;

- 32x8 registres d'usage universel de fonctionnement ;
- Opération entièrement statique ;
- Jusqu'à 16 MIPS de sorties à 16 Mhz ;
- Mémoires non-volatiles de programme et de données ;
- Bytes 64K/128K/256K de flash Individu-Programmable de Dans-Système ;
- Résistance : 10.000 écrire /cycles d'effacement ;
- 4K Bytes EEPROM ;
- Serrure de programmation pour la sécurité de logiciel ;
- Quatre canaux à 8 bits de PWM ;
- Puissance sur la remise et la détection programmable d'arrêt partiel [20].

III.2.1.2 Sources d'alimentation de la carte Mega2560

La carte Arduino Mega2560 peut être alimentée soit via la connexion USB qui fournit 5V jusqu'à 500mA, ou à l'aide d'une alimentation externe. La source d'alimentation est sélectionnée automatiquement par la carte. Elle peut fonctionner avec une alimentation externe de 6 à 20 volts. Cependant, si la carte est alimentée avec moins de 7V, la broche 5V pourrait fournir moins de 5V, et la carte pourrait être instable. Si on utilise plus de 12V, le régulateur de tension de la carte pourrait chauffer et endommager la carte. Aussi, la plage idéale recommandée pour alimenter la carte Arduino est entre 7V et 12V. Les broches d'alimentation sont les suivantes [20] :

- **VIN :** La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée).
- **5V :** La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte. Le 5V régulé fourni par cette broche peut donc provenir soit de la tension d'alimentation VIN via le régulateur de la carte, ou bien de la connexion USB ou de tout autre source d'alimentation régulée ;
- **3.3V :** l'alimentation 3.3V est intéressante pour certains circuits nécessitant cette tension au lieu du 5V, cette alimentation est fournie par une puce FTDI (Future Technology Devices International), c'est un circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATmega2560) de la carte. L'intensité maximale disponible sur cette broche est de 50mA ;
- **GND :** Broche de masse (0V).

III.2.1.3 Mémoire

L'ATmega2560 à 256Ko de mémoire FLASH pour stocker le programme, dont 8Ko également utilisés par le bootloader, qui est un programme préprogrammé une fois pour toute dans l'ATmega et qui permet la communication entre l'ATmega et le logiciel Arduino via le port USB, notamment lors de chaque programmation de la carte. L'ATmega2560 a également 8 ko de mémoire vive statique (SRAM) et 4Ko d'EEPROM (Electrically-erasable programmable read-only memory) un type de mémoire morte non volatile [20].

III.2.1.4 Entrées et sorties numériques

Chacune des 54 broches numériques de la carte Mega 2560 peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40 mA d'intensité et dispose d'une résistance interne de 20-50 KOhms déconnectée par défaut de "rappel au plus" (en anglais : pull-up). Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digitalWrite (broche, HIGH).

III.2.1.5 Broches analogiques

La carte Mega 2560 dispose de 16 entrées analogiques, chacune pouvant fournir une mesure d'une résolution de 10 bits (1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analogRead() du langage Arduino. Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023). Les broches analogiques peuvent être utilisées en tant que broches numériques [20].

III.2.1.6 Autres broches

Il y a deux autres broches disponibles sur la carte :

- **AREF :** Tension de référence pour les entrées analogiques (si différent du 5V). Utilisée avec l'instruction analogReference() ;
- **Reset :** Mettre cette broche au niveau BAS entraîne la réinitialisation du microcontrôleur.Comme un port de communication virtuel pour le logiciel sur l’ordinateur, La connexion série de l'Arduino est très pratique pour communiquer avec un PC, mais son inconvénient est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil [20].

III.2.2 Interface de programmation IDE Arduino

Dans le domaine du développement informatique, l'IDE (en anglais, Integrated Development Environment) regroupe un ensemble d'outils spécifiques. Ceux-ci sont dédiés aux programmeurs afin qu'ils puissent optimiser leur temps de travail et améliorer leur productivité. Autrement dit, l'IDE facilite la mise en œuvre de projets tels que le développement de logiciels ou d'applications. L'Arduino fournit un environnement de développement qui s'appuyant sur des outils open source c'est interface de programmation IDE, cette interface est plutôt simple voir figure III.3, il offre une interface minimale et épurée pour développer un programme sur les cartes Arduino. Il est doté d'un éditeur de code avec coloration syntaxique et d'une barre d'outils rapide. Ce sont les deux éléments les plus importants de l'interface, c'est ceux que l'on utilise le plus souvent. On retrouve aussi une barre de menus, plus classique qui est utilisée pour accéder aux fonctions avancées de l'IDE. Enfin, une console pour afficher les résultats de la compilation du code source, des opérations sur la carte, etc. L'environnement IDE de programmation open-source pour Arduino peut être téléchargé gratuitement (pour Mac OS X, Windows, et Linux). Le langage Arduino est basé sur les langages C/C++. Un programme sur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code. L'injection du programme déjà converti par l'environnement sous forme d'un code « HEX » dans la mémoire du microcontrôleur se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties [22].

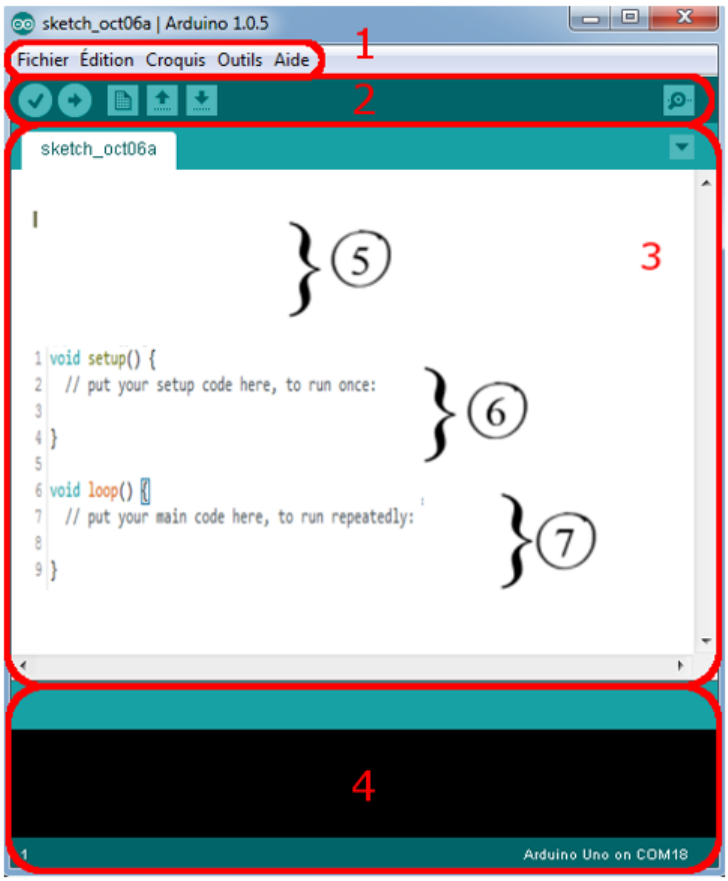

Figure III.3 : Interface IDE du logiciel Arduino.

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

- Le cadre numéro 1 : Ce sont les options de configuration du logiciel ;
- Le cadre numéro 2 : C'est la barre d'outils elle contient les boutons qui vont nous servir lorsque l'on va programmer nos cartes ;
- Le cadre numéro 3 : Ce bloc va contenir le programme que nous allons créer, ce cadre contient 3 parties qui sont :
    1) Partie déclaration de variables (globales) ;
    2) Initialisation «Void Setup () {}»: Au démarrage de l'Arduino toutes les instructions comprises entre les deux accolades serrent exécuter qu'une seul fois ;
    3) Boucle principale «Void loop () {}»: Les instructions sont répéter indéfiniment tant que l'Arduino fonctionne ;
- Le cadre numéro 4 : C'est le débogueur, celui-ci est important, car il va nous aider à corriger les fautes dans notre programme.

La figure III.4 détaille le cadre numéro 2 qui contient les boutons de programmation du logiciel IDE.

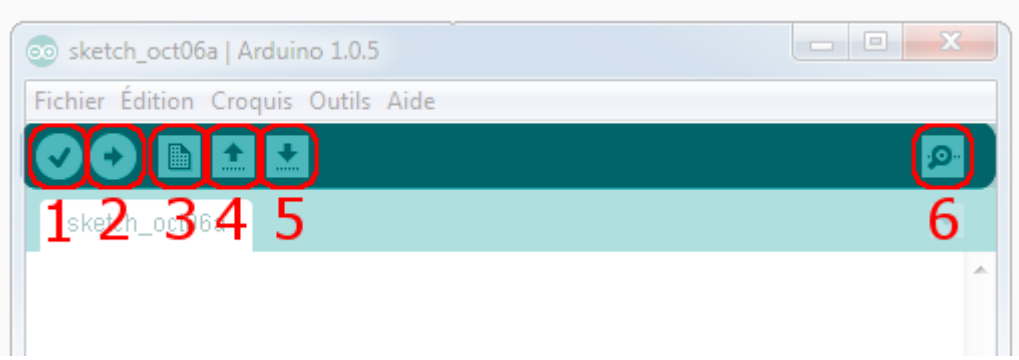

Figure III.4 : Barre d'outils du logiciel IDE.

- **Bouton 1 : Vérifier**, le bouton qui permet de vérifier le programme, il actionne un module qui cherche les erreurs dans le programme ;
- **Bouton 2 : Téléverser**, compiler et envoyer le programme vers la carte ;
- **Bouton 3 : Nouveau**, crée un nouveau fichier ;
- **Bouton 4 : Charger**, ce bouton permet d'ouvrir un programme existant ;
- **Bouton 5 : Sauvegarder** et enregistrer le programme en cours ;
- **Bouton 6 : Ouvrir le moniteur série**, c'est le bouton de base sur la carte Arduino, car on ne peut pas afficher de texte, donc il faut ajouter un module d'affichage ou bien se servir du moniteur série pour utiliser l'écran de notre ordinateur pour savoir où on en est dans l'exécution du programme.

III.3 Actionneurs

III.3.1 Capteur à ultrasons HC-SR04

Le HC-SR04 est un capteur de distance exploite les ultrasons, il offre une excellente plage de détection avec des mesures stables [23]. Son principe est l'utilisation de l'écho pour déterminer la distance à laquelle se trouve un objet, un court signal sonore est envoyé (40kHz) par le capteur puis le son est réfléchi par une surface et repart en direction du capteur ce dernier le détecte une fois revenu à son point de départ. La durée entre l'instant de l'émission et l'instant de la réception peut être mesurée, le signal ayant parcouru 2 fois la distance entre le capteur et la surface c'est un aller-retour, on peut la calculer ainsi :

$$distance = \frac{vitesse\ du\ son}{2} \times durée$$

La vitesse du son est environ égale à 340 m/s.

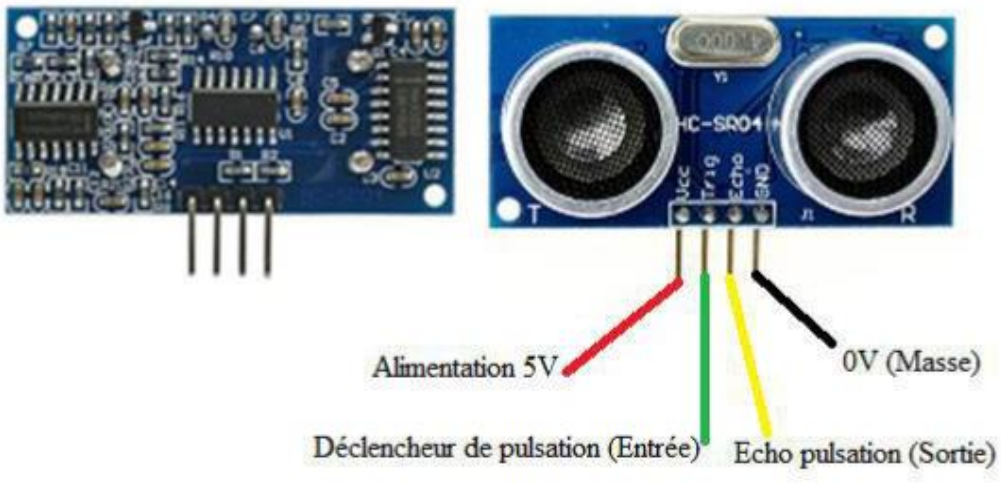

Figure III.5 : Capteur Sonar à Ultrasons HC-SR04.

En générale ce type de capteur peut planter en-dessous de 1 cm et au-dessus de 4 m par rapport à un obstacle, il est alors nécessaire de le réinitialiser par la coupure puis la remise de son alimentation. Les déférentes caractéristiques du capteur HC-SR04 sont illustrées dans le tableau III.2:

Tableau III.2 : Caractéristiques techniques du capteur HC-SR04.

Caractéristique	Valeur
Dimensions	45 mm x 20 mm
Plage de mesure	2 cm à 4m
Résolution de la mesure	0.3 cm
Angle de mesure efficace	<15 °
Largeur d'impulsion	10 µs
Fréquence	40KHZ

III.3.2 Module Afficheur LCD 20x4

L'écran LCD (Liquid Crystal Display) est un module d'affichage électronique, il se trouve dans un large éventail d'applications. Un écran LCD est un petit affichage à faible coût. Il est facile de s'interfacer avec un microcontrôleur (Arduino). L'écran LCD est préféré sur sept segments et d'autres LED multi-segments, car le module LCD est économique ; facilement programmable ; Il n'ya pas de limite à l'affichage de caractères spéciaux et même personnalisés contrairement à sept segments, les animations et ainsi de suite. Un LCD 20x4 signifie qu'il peut afficher 20 caractères par ligne et il y a 4 lignes de ce type [24].

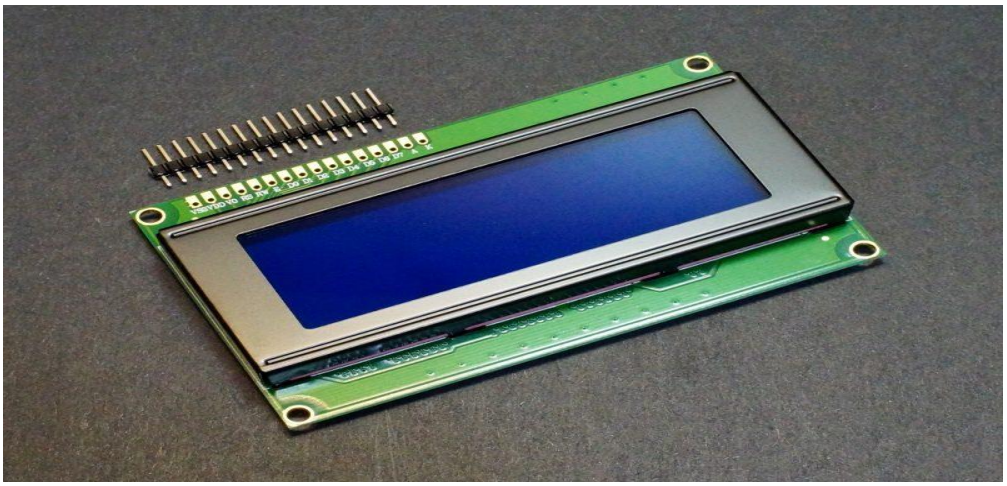

Figure III.6 : Afficheurs LCD (20x4).

Le tableau suivant présente les spécifications des broches de LCD et leur rôle :

Tableau III.3 : Nomenclature du connecteur de l'afficheur LCD.

Numéro	Nom	Rôle
1	GND	Masse 0V
2	VDD	Alimentation +5V
3	VEE	Tension de réglage du contraste
4	RS	Sélection du registre donnée ou commande
5	RW	Lecture ou écriture
6	E	Activation pour un transfert (enable)
7	D0	Bit 0 de la donnée/commande

8	D1	Bit 1 de la donnée/commande
9	D2	Bit 2 de la donnée/commande
10	D3	Bit 3 de la donnée/commande
11	D4	Bit 4 de la donnée/commande
12	D5	Bit 5 de la donnée/commande
13	D6	Bit 6 de la donnée/commande
14	D7	Bit 7 de la donnée/commande

III.3.3 Buzzer

Un buzzer est un élément électromécanique ou électronique qui produit un son quand on lui applique une tension. Certains nécessitent une tension continue comme les buzzers électromécaniques, d'autres nécessitent une tension alternative par exemple les transducteurs piézo- électrique [25].

III.3.4 Circuit L293D

Le L293D est un double pont-H, ce qui signifie qu'il est possible de l'utiliser pour commander quatre moteurs distincts (dans un seul sens) grâce à ses 4 canaux. En raccordant les sorties de façon appropriées, il est possible de constituer deux pont-h. Il est ainsi possible de commander deux moteurs distincts, dans les deux sens et indépendamment l'un de l'autre. Même si cette documentation n'utilise qu'un seul des deux pont-H pour la commande du moteur, il vous sera facile de faire le nécessaire pour commander un deuxième moteur tout aussi simplement. Le L293D un circuit intégré à haut voltage, grand courant et 4 canaux." Cela veut dire que ce circuit intégré peut être utilisé pour des moteurs DC alimentés jusqu'à 36 Volts (ce sont déjà des jolis petits moteurs). Le circuit peut fournir un maximum de 600mA par canal. Le L293D est aussi connu pour être un excellent Pont-H facile à mettre en pratique. Avec deux signaux de commande Input 1 et Input 2 fournit par Arduino, il est possible d'inverser la direction du courant dans le pont-H et donc renverser le sens de rotation du moteur qui y est raccordé. En utilisant différentes combinaisons de Input 1 et input 2 il devient possible de démarrer, stopper ou inverser le courant. Ci-dessous la configuration des broches du L293D et la table de la logique de commande.

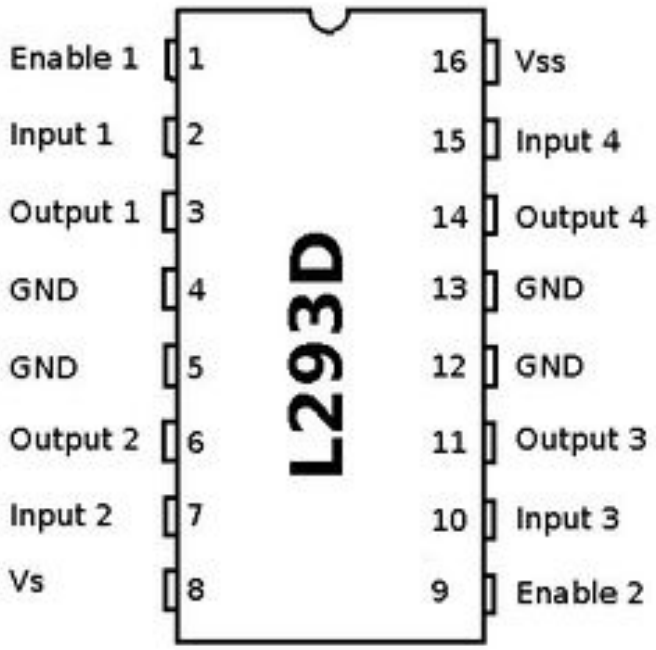

Figure III.7 : Circuit L293D.

Le principe de base de pont H est un assemblage de 4 transistors (2 PNP et 2 NPN) monté de telle façon que le courant puisse passer soit dans un sens, soit dans l'autre au travers de la charge (un moteur continu par exemple). En inversant le sens du courant dans le moteur, ce dernier changera de sens de rotation. En règle générale, lorsque l'on utilise des relais ou moteurs avec un transistor, il faut protéger le transistor à avec une diode anti-retour. La diode à pour but de renvoyer à la masse les surtensions induites par les effets transitoires (lors du déclenchement du bobinage d'un relais... ou bobinage du moteur). Sans cette diode anti-retour (aussi dite "en roue libre"), le transistor ne survit pas bien longtemps. Un pont-H étant constitué de transistor et un moteur étant le siège d'effets transitoire, les différents transistors doivent être protégés à l'aide de diode.

III.3.5 Diode électroluminescente LED

La DEL en français appelée diode électroluminescente est un composant optoélectronique qui émet de la lumière lorsqu'elle est parcourue par un courant électrique dans le sens direct. C'est-à-dire le courant doit la traversée de l'anode ou la tension la plus élevée vers la cathode [26].

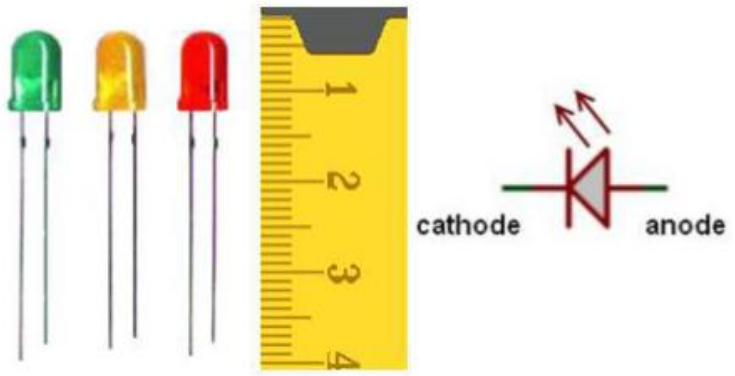

Figure III.8 : Diode électroluminescente LED.

III.3.6 Résistances

Une résistance (mesurée en ohms : **Ω**) est un composant électronique ou électrique dont la principale caractéristique est d'opposer une plus ou moins grande résistance à la circulation du courant électrique.

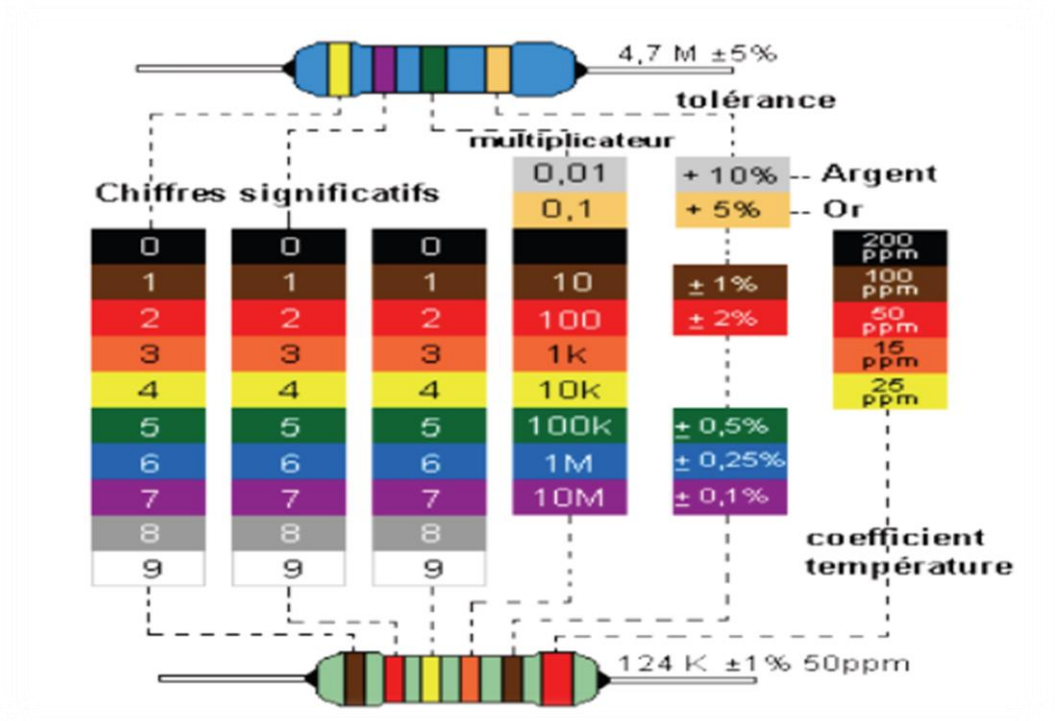

Figure III.9 : Code de couleurs pour déterminer la valeur d'une résistance.

III.3.7 Potentiomètre

Un potentiomètre est un composant électronique à trois bornes qui agit comme une résistance variable. Deux des bornes de bord sont connectées aux extrémités de l'élément résistif.

Figure III.10 : Potentiomètre.

III.3.8 Relais

Un relais électromécanique est un organe électrique permettant de distribuer la puissance à partir d'un ordre émis par la partie commande. Ainsi, un relais permet l'ouverture et la fermeture d'un circuit électrique de puissance à partir d'une information logique. Les 2 circuits, puissance et information, sont complètement isolés par l'isolation galvanique et peuvent avoir des caractéristiques d'alimentation électrique différentes.

Le rôle de la fonction première des relais est le plus souvent de séparer les circuits de commande des circuits de puissance à des fins d'isolement, par exemple pour piloter une tension ou un courant élevé, à partir d'une commande plus faible, et dans certaines applications, assurer aussi la sécurité de l'opérateur.

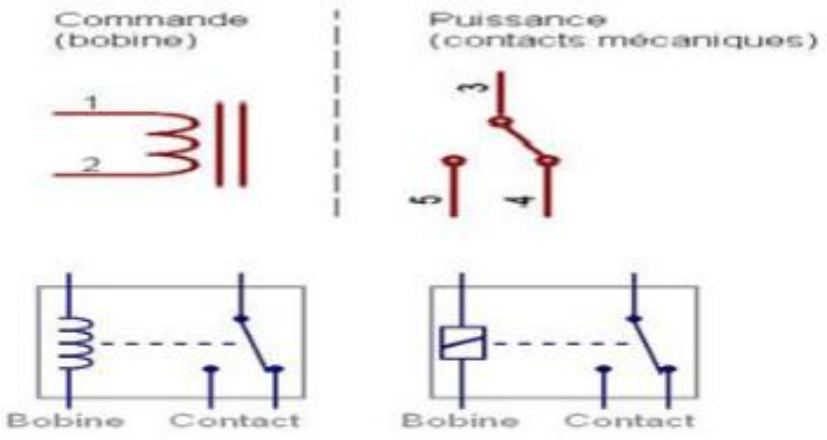

Figure III.11 : Symboles des relais.

III.3.9 Transistor

Le transistor est un composant électronique qui est utilisé dans la plupart des circuits électroniques par exemple circuits logiques, amplificateur, stabilisateur de tension, modulation de signal, etc. aussi bien en basse qu'en haute tension. Un transistor est un

dispositif semi-conducteur à trois électrodes actives, qui permet de contrôler un courant ou une tension sur l'électrode de sortie (le collecteur pour le transistor bipolaire et le drain sur un transistor à effet de champ) grâce à une électrode d'entrée (la base sur un transistor bipolaire et la grille pour un transistor à effet de champ) [27]. La figure III.12 présente les deux types de transistor ; le transistor NPN et le transistor PNP

Figure III.12 : Les types de transistor (NPN et PNP)

III.3.10 Pompe

Une pompe est un dispositif permettant d'aspirer et de refouler un liquide (les compresseurs). Les infiltrations d'eau noyant de façon continuent les galeries souterraines, on utilisa alors de façon régulière des pompes pour évacuer cette eau [28].

Figure III.13 : Pompe Centrifuge.

La mise en œuvre de procédés de transformation de matière nécessite très souvent le transport de fluides. L'acheminement d'un fluide d'un point à un autre peut être réalisé en utilisant les forces de gravité, de vide ou des pressions, mais ces solutions sont limitées dans leurs applications. Dans la plupart des cas, on a recours aux pompes, permettant de travailler à des débits réguliers, contrôlés et sur des distances et des hauteurs importantes. Deux grandes catégories de pompes existent :

- Les pompes volumétriques, qui fonctionnent sur le principe du déplacement d'un volume de fluide.
- Les pompes centrifuges, qui fonctionnent sur le principe du rotor et du stator.

Chaque catégorie présente certains avantages et inconvénients, mais globalement on retrouve plus fréquemment les pompes centrifuges pour les applications industrielles.

III.3.11 Moteur a courant continue

Les moteurs courant continu sont des convertisseurs de puissance, Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur. Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînante. On dit alors qu'ils ont un fonctionnement en générateur [29].

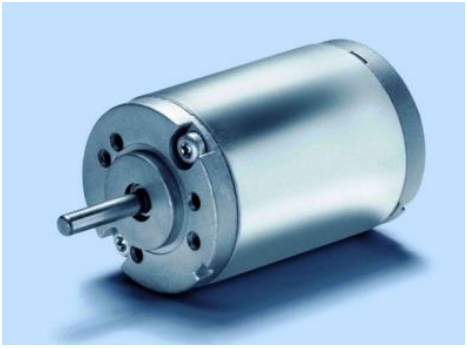

Figure III.15 : Moteur a courant continue.

III.4 Conclusion

Dans ce chapitre nous avons présenté la carte Arduino Mega et ces différentes caractéristiques, puis nous avons la partie logicielle et langage de programmation sur l'interface Arduino IDE. Nous avons également fait une description de tous les composants utilisés pour contrôler le niveau d'eau dans un réservoir et c'est la partie la plus intéressante dans notre travail qui sera développée dans le chapitre suivant.

Chapitre IV

Simulation et résultats

IV.1 Introduction

Ce chapitre traite de la mise en œuvre et la simulation sous Proteus un système de contrôle et de régulation automatique de niveau d'eau d'un réservoir, en utilisant le capteur sonar à Ultrason HC-SR04 avec Arduino Mega 2560. Le niveau de remplissage peut être déterminé lorsque qu'un niveau de remplissage défini est atteint et envoie un signal de commutation.

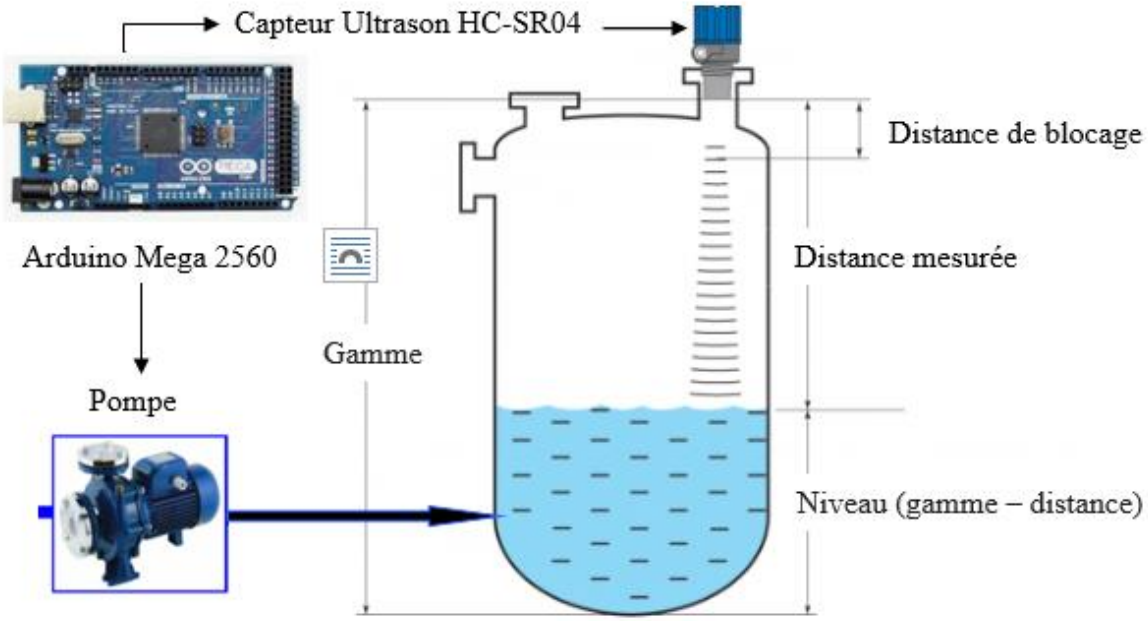

Figure IV.1 : Schéma fonctionnel de contrôle automatique de niveau d'eau d'un réservoir.

La figure illustre le système de remplissage automatique d'un réservoir est constitué essentiellement d'une carte Arduino Mega, une pompe, un capteurHC-SR04 et d'un réservoir source.

IV.2 Fonctionnement et simulation

IV.2.1 Présentation de l'interface d'ISIS PROTEUS

Le logiciel ISIS Proteus qui est un très bon logiciel de simulation en électronique est exploité pour simuler notre système. Isis est un éditeur de schémas qui intègre un simulateur analogique, logique ou mixte. Toutes les opérations se passent dans cet environnement, aussi bien la configuration des différentes sources que le placement des sondes et le tracé des courbes. La simulation permet d'ajuster et de modifier le circuit comme si on manipulait un montage réel. Ceci permet d'accélérer le prototypage et de réduire son coût. Il faut toujours

prendre en considération que les résultats obtenus de la simulation sont un peu différents de celles du monde réel, et cela dépend de la précision des modèles, des composants et de la complication des montages.

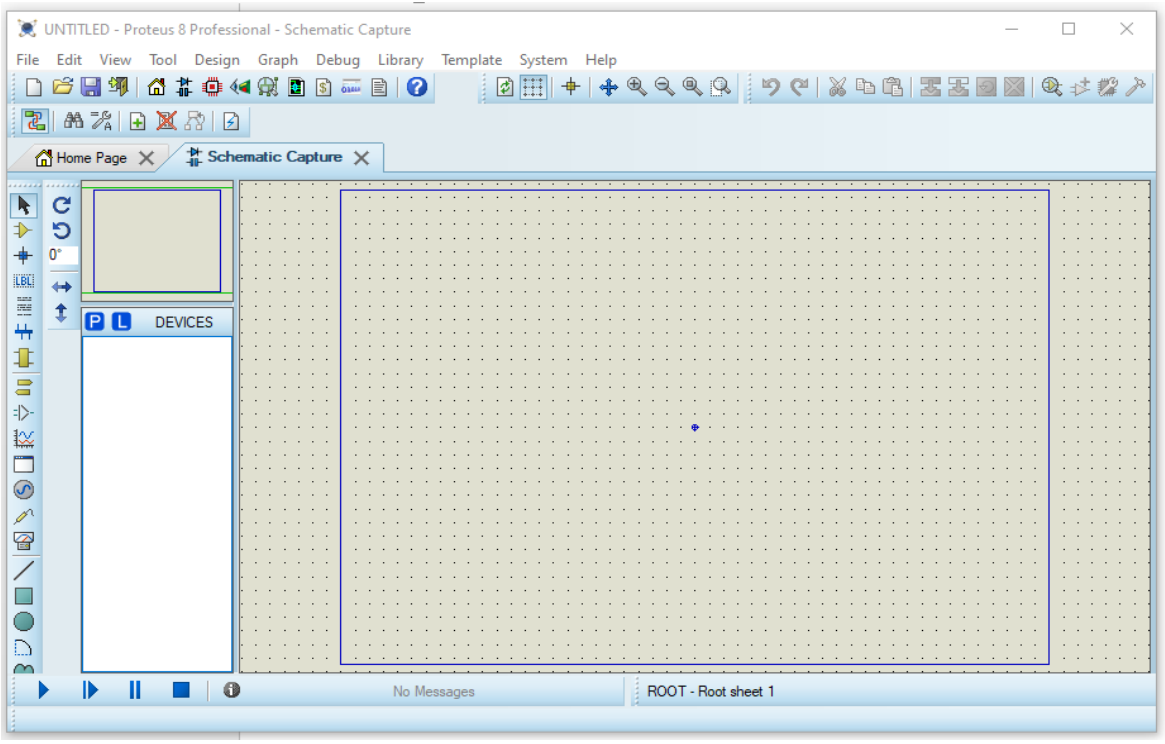

Figure IV.2 : Interface de logiciel de simulation ISIS.

Pour créer un schéma et choisir un composant : La grille qui apparaît en fond d'écran désigne la page de travail dans laquelle vous allez dessiner votre montage de composants électroniques.

Cliquer sur l'icône pour faire apparaître la boîte de dialogue donnant le choix des composants.

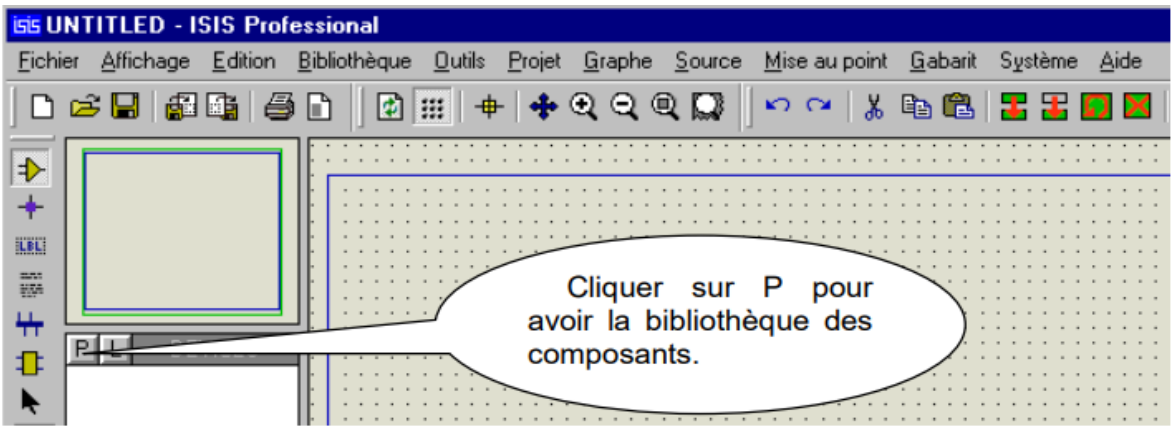

Figure IV.3 : Choix des composants.

IV.2.2 Simulation du circuit du schéma fonctionnel dans Proteus

Les bibliothèques de la carte Arduino Mega 2560 et du capteur Ultrason HC-SR04 doivent être téléchargées et ajoutées dans la bibliothèque de logiciel Proteus Isis, les autres composants utilisés dans la simulation de ce projet sont :

1) Carte Arduino Mega2560 ;
2) Capteur sonar à Ultrason HC-SR04 ;
3) Potentiomètre variable ;
4) Moteur à courant continu MCC ;
5) Écran LCD 20×4 ;
6) Transistor ;
7) Diode ;
8) Résistance (10 K ohms) ;
9) Buzzer (Bipeur) ;
10) Oscilloscope.

Voici le raccordement d'appareils de mesure sur les broches de l'arduino :

- L'alimentation 5V de la carte Arduino va sur la broche VCC du capteur ;
- L'alimentation 5V de la carte Arduino va sur la broche VEE et VDD du LCD ;
- La broche GND de la carte Arduino va sur la broche VSS et RW du LCD ;
- La broche GND de la carte Arduino va sur la broche GND du capteur ;
- La broche 23 de la carte Arduino va sur la broche TRIGGER du capteur ;
- La broche 25 de la carte Arduino va sur la broche ECHO du capteur ;
- La broche 12 de la carte Arduino va au buzzer ;
- La broche 11 de la carte Arduino va sur l'entré 1 du L293D ;
- La broche 10 de la carte Arduino va sur l'entré 2 du L293D
- Les broches 7,6,5,4,3,2 de la carte Arduino va sur les broches D7,D6,D5,D4,E,Rs. De LCD ;
- La broche 9 de la carte Arduino va au bouton poussoir (auto/manual) ;
- La broche 13 de la carte Arduino va au bouton poussoir (on/off pump).

La simulation du projet consiste à établir les différentes connexions entre les différents composants utilisés. Le schéma de câblage établi sur le logiciel Proteus est représenté par la figure IV.4.

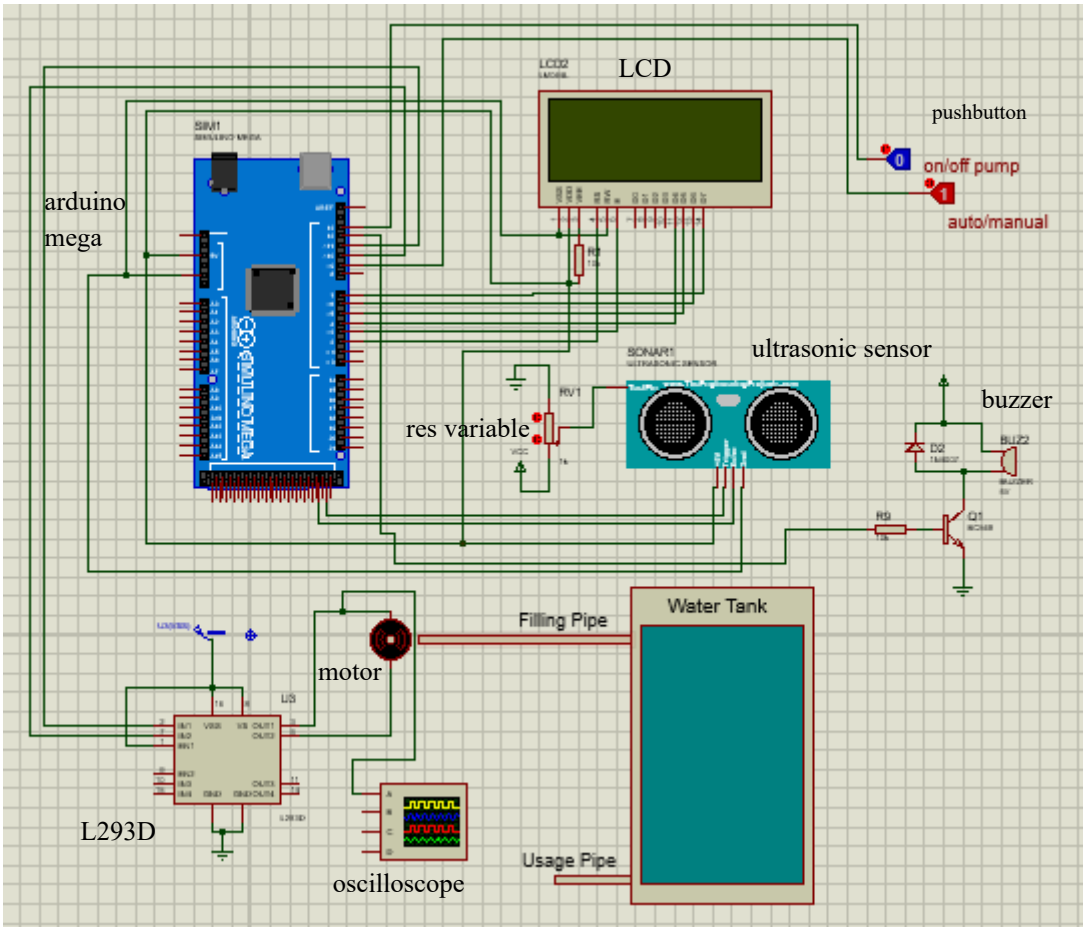

Figure IV.4 : Simulation de control et régulation de niveau d'eau sur Proteus Isis.

Pour exécuter le circuit réalisé sur Proteus, il faut d'abord envoyer le fichier hexadécimal généré après l'implémentation du code sur IDE d'Arduino, ce programme consiste à contrôler la vitesse de moteur MCC selon la distance d'obstacles fourni par le capteur ultrason., donc en doit fournir le code exécutable qui contient tout ce qui est nécessaire au microcontrôleur pour exécuter correctement le programme. La figure suivante illustre le déduit de simulation qui est affiché sur l'écran LCD.

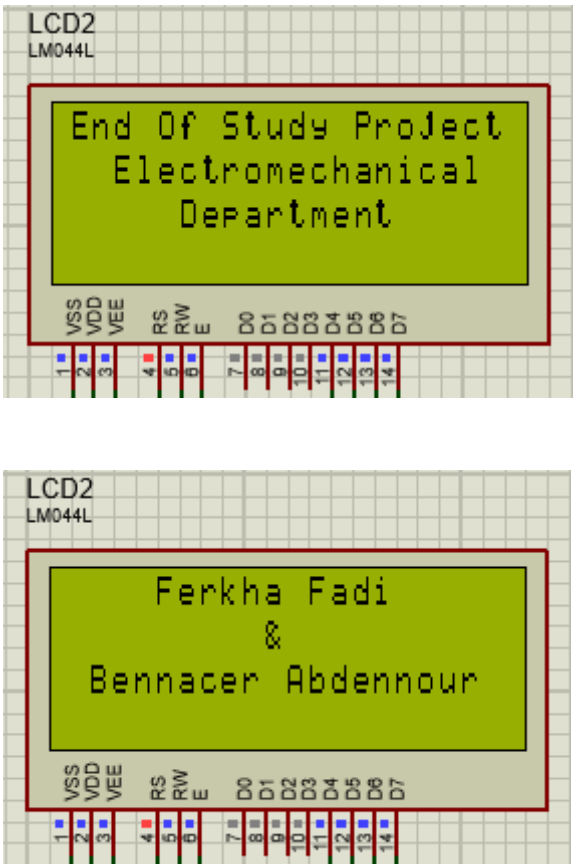

Figure IV.5 : Lancement de simulation et affichage des noms des deux étudiants.

En suit en exécute le mode automatique qui s'affiche en état auto, la figure suivante montre que le réservoir est plein 100%.

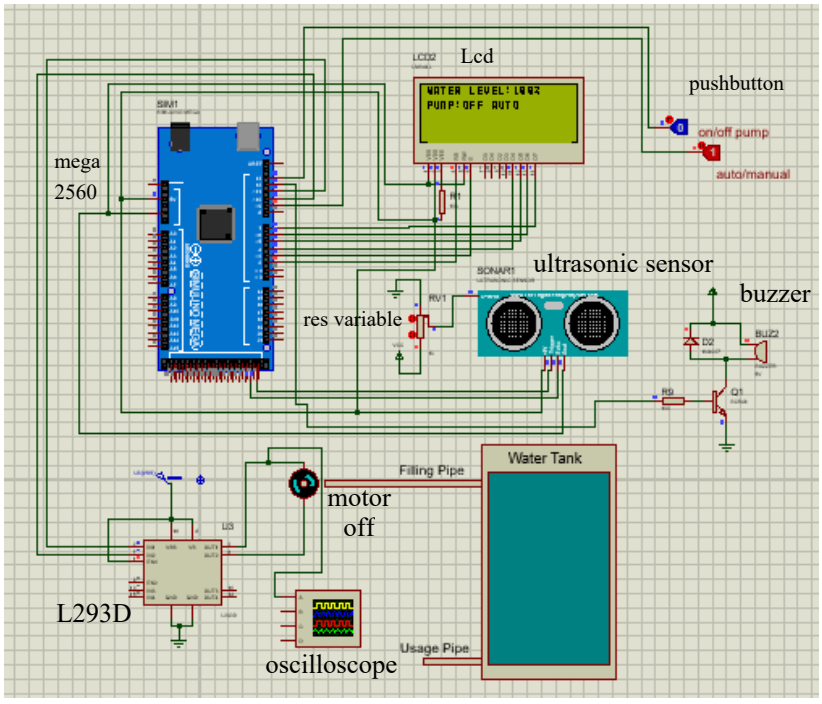

Figure IV.6 : Système en état auto et pompe off et le réservoir est plain100%.

Ce montage de circuit simulé remplir le réservoir automatiquement ou manuel sur tout dans les cas d'urgence et le coté automatique cesse de fonctionner, donc le circuit contient 2 boutons un bouton poussoir pour changer l'état auto au manuel et un autre lorsque l'état manuel est active pour lancer et arrêter la pompe manuellement. D'abord la carte Arduino méga est programmée de piloter et lancer la pompe pour remplir le réservoir lorsque le niveau d'eau est faible ou bien il est diminué au-dessous 20%, la pompe s'arrêter lorsque le niveau est augmenté jusqu'à 100%. Le circuit contient aussi un capteur ultrason pour mesurer le niveau d'eau dans le réservoir, le capteur est branche avec un potentiomètre pour crée un obstacle en simulation, on faire varier le potentiomètre, cette variation est considéré comme un obstacle pour le capteur qui va calculer la distance, et envoie un signal a là la carte Arduino Mega, cette dernier va décider le lancement ou l'arrêt de la pompe, la vitesse du moteur est contrôlé selon les besoin par le changent du rapport cyclique de signal PWM. Lorsque le processus de remplissage est terminé le sonore s'allume pendant quelques secondes. La figure suivante illustre l'affichage de niveau d'eau qui est diminué jusq'au 65% et la pompe est en état OFF.

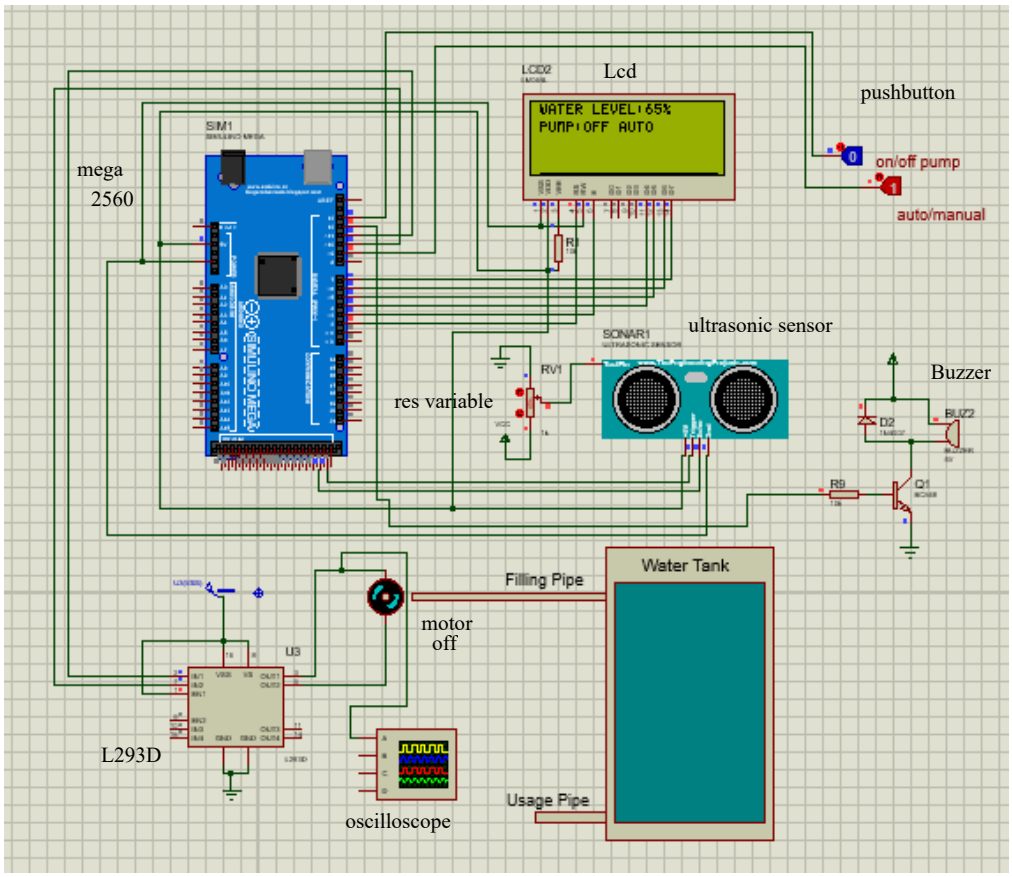

Figure IV.7 : Système en état auto et pompe off et niveau d'eau et de 65%.

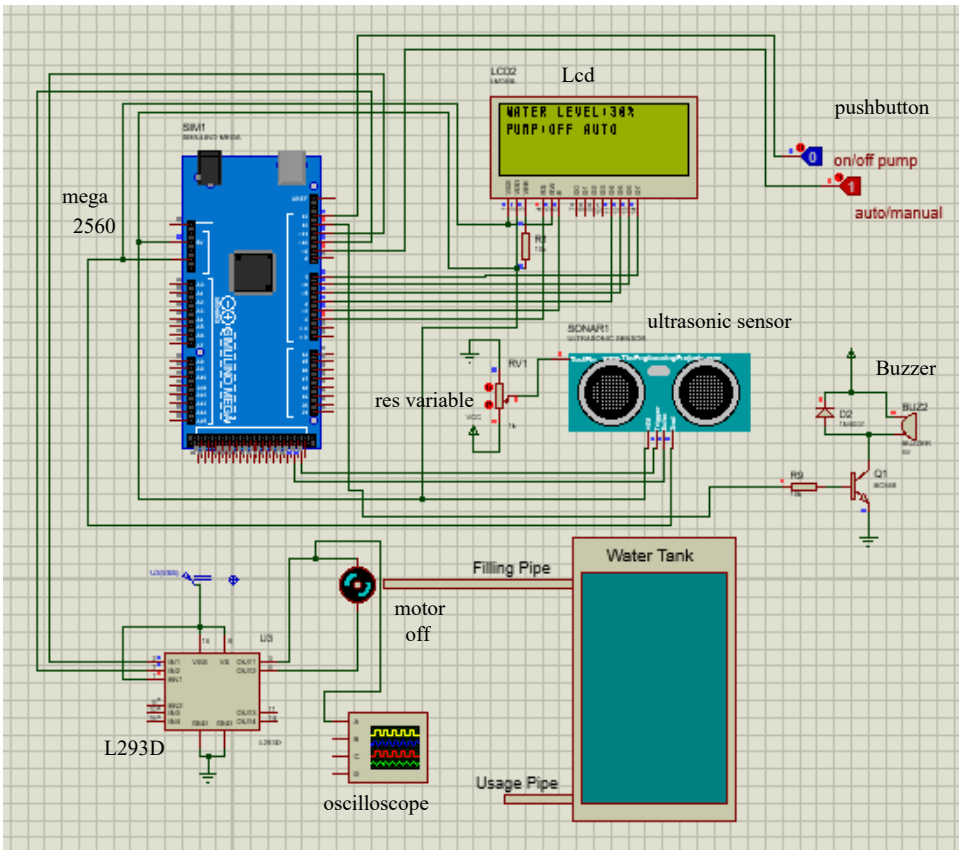

Figure IV.8 : Pompe off et niveau d'eau de réservoir est de 30%.

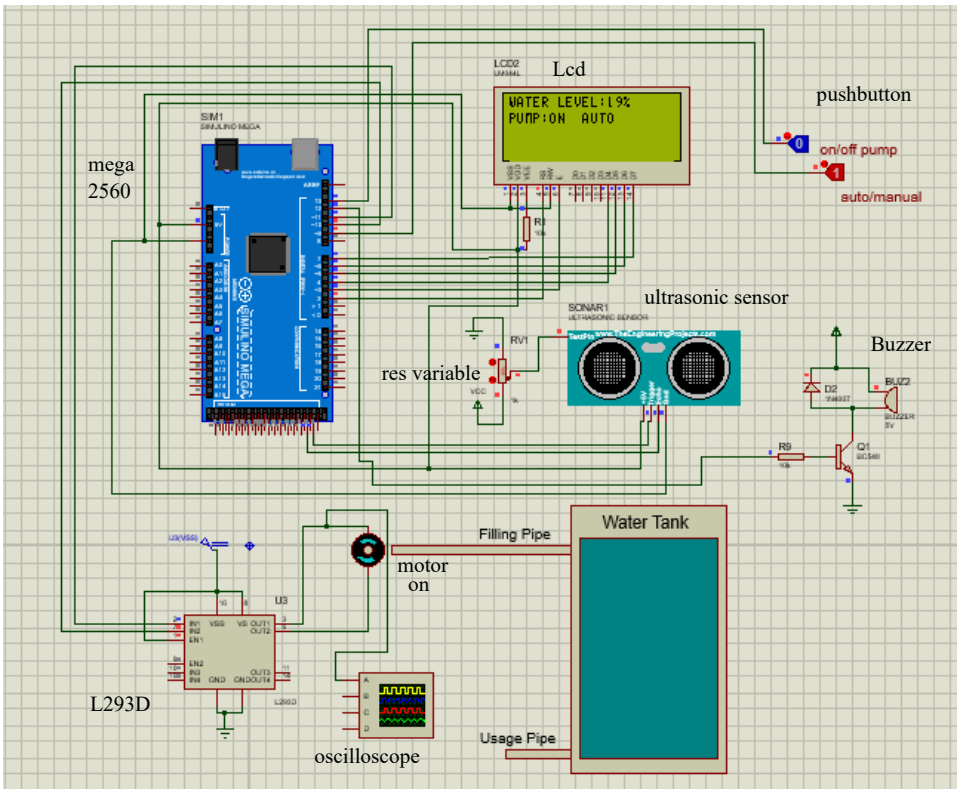

Figure IV.9 : Pompe est en état on et le niveau d'eau de réservoir est de 19%.

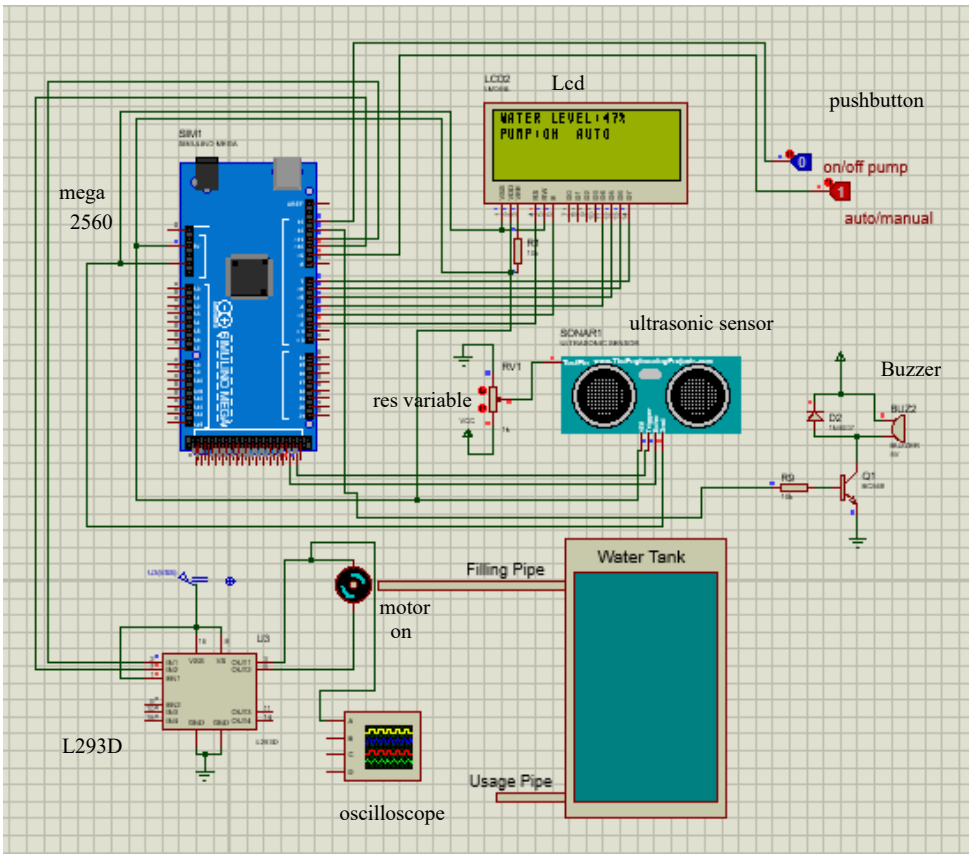

Figure IV.10 : Pompe est en état on et le niveau d'eau de réservoir est de 47%.

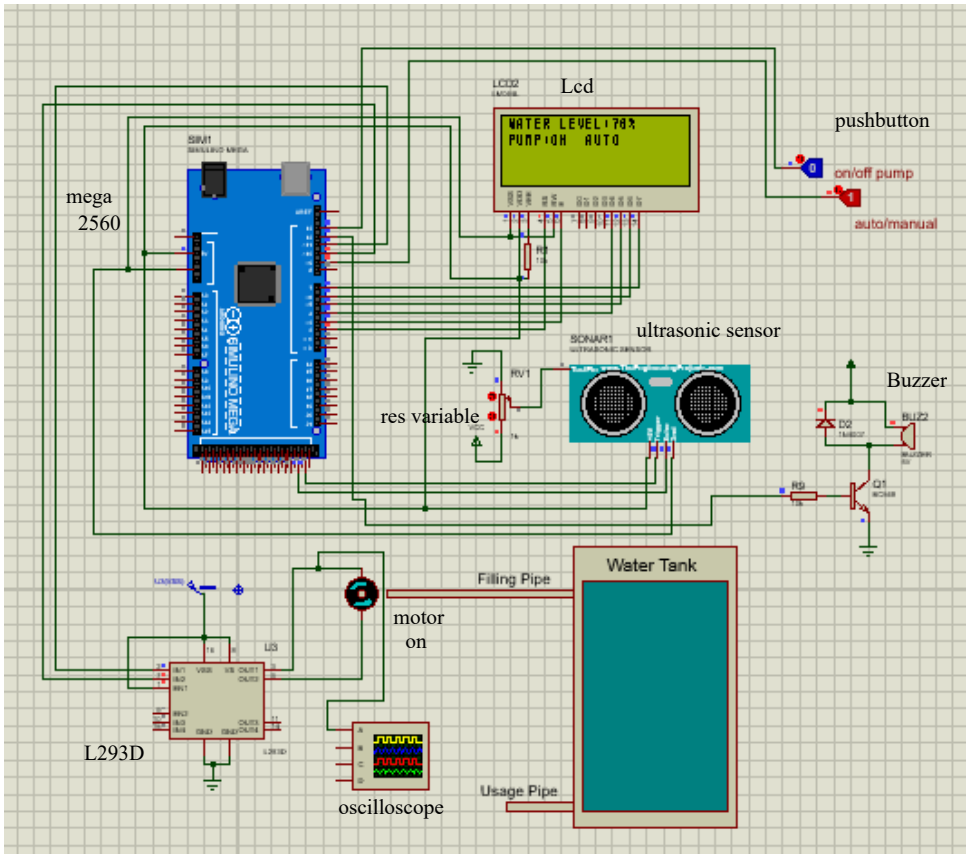

Figure IV.11 : Pompe est en état on et le niveau d'eau de réservoir raugmente à 70%.

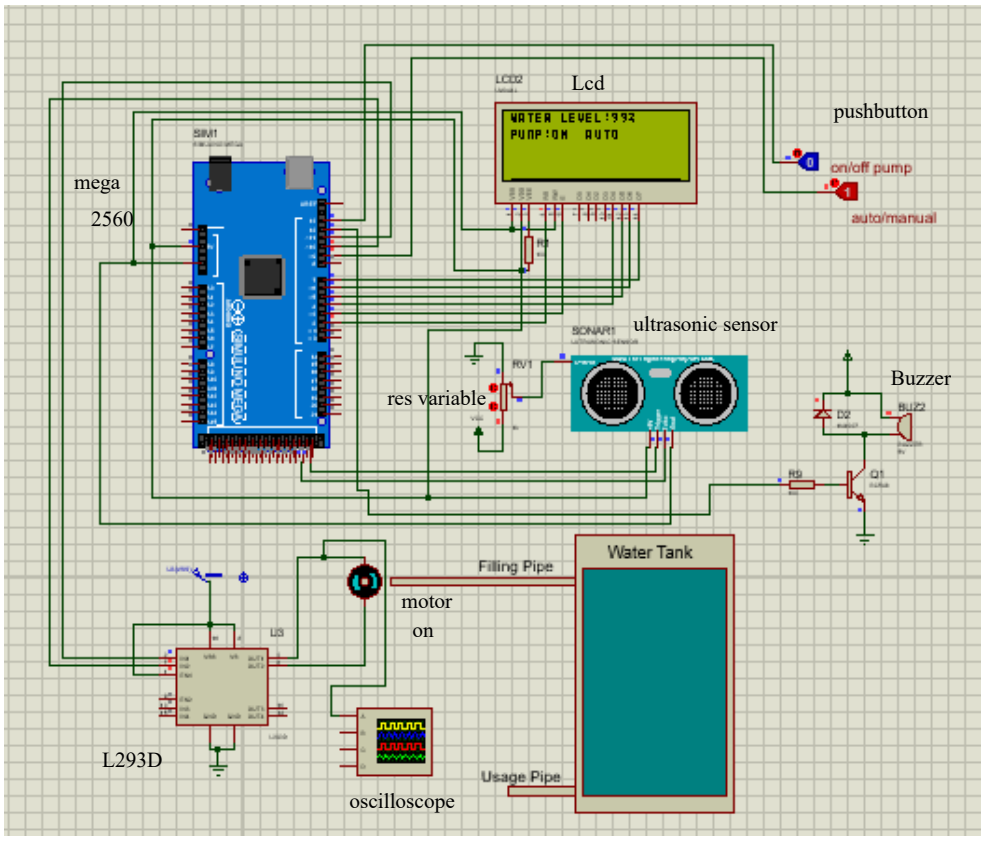

Figure IV.12 : Pompe est en état on et le niveau d'eau de réservoir raugmente à 99%.

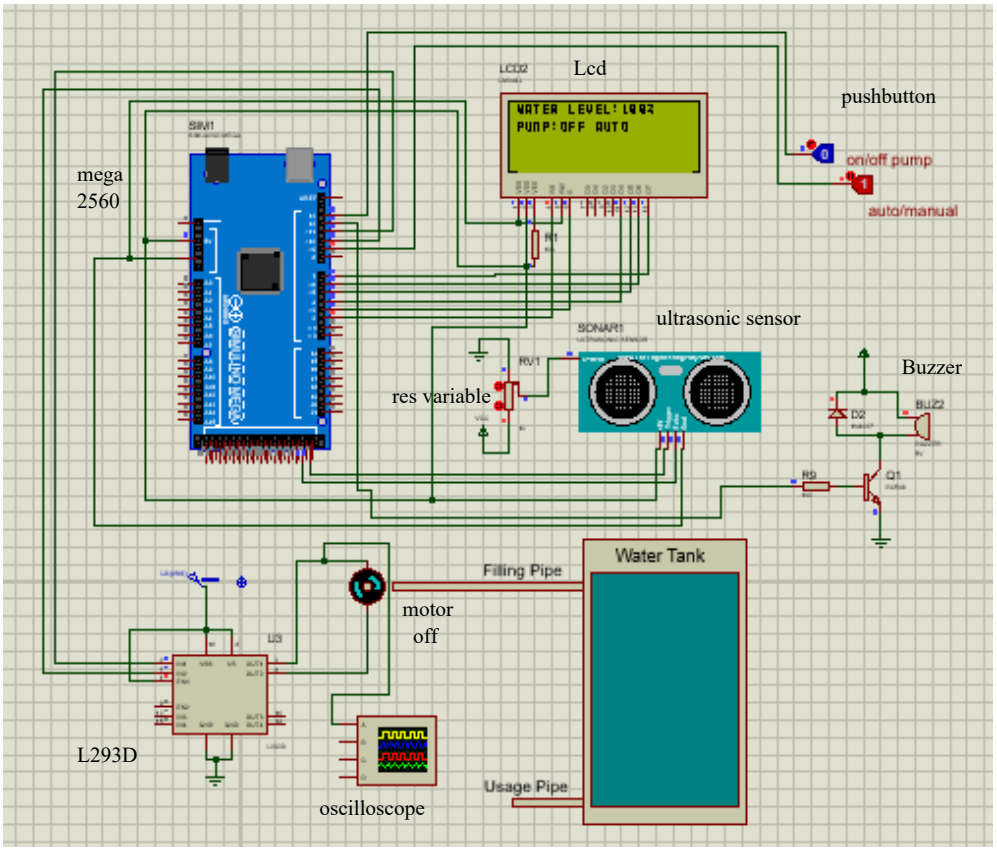

Figure IV.13 : Pompe en état off et le niveau d'eau est à 100% et le de réservoir est plain.

Maintenait-on passe au système en état manuel pour accéder à la fonction de travail manuel en appuyant sur le bouton poussoir (auto/manuel). Les figures suivant affichent différentes états de réservoir et de pompe.

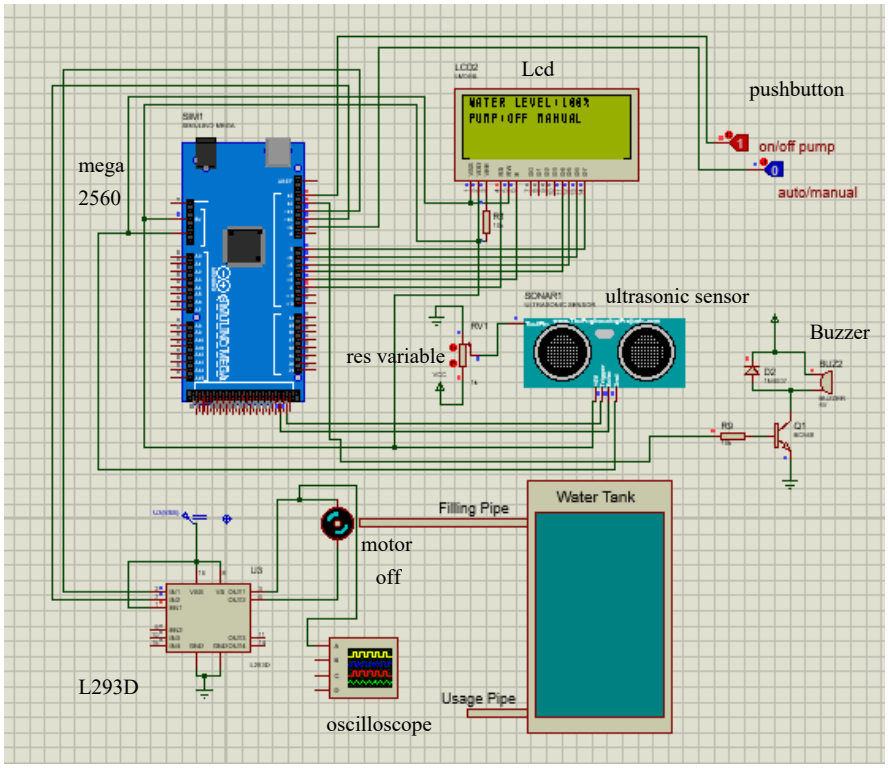

Figure IV. 14 : Système en état manuel et pompe off.

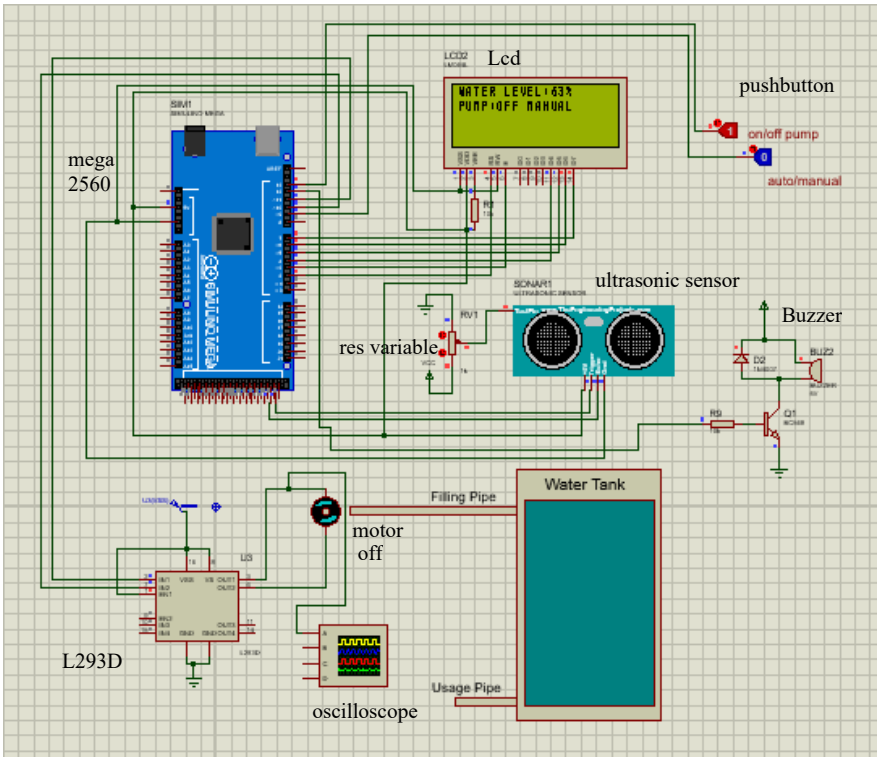

Figure IV.15 : Système en état manuel la pompe off et niveau d'eau à 63%.

Même si on est à état manuel et on veut remplir le réservoir et même si le niveau est 63%, il faut juste appuyer sur le bouton poussoir (on/off pump) et la pompe va au état ON et commence le remplissage d'eau.

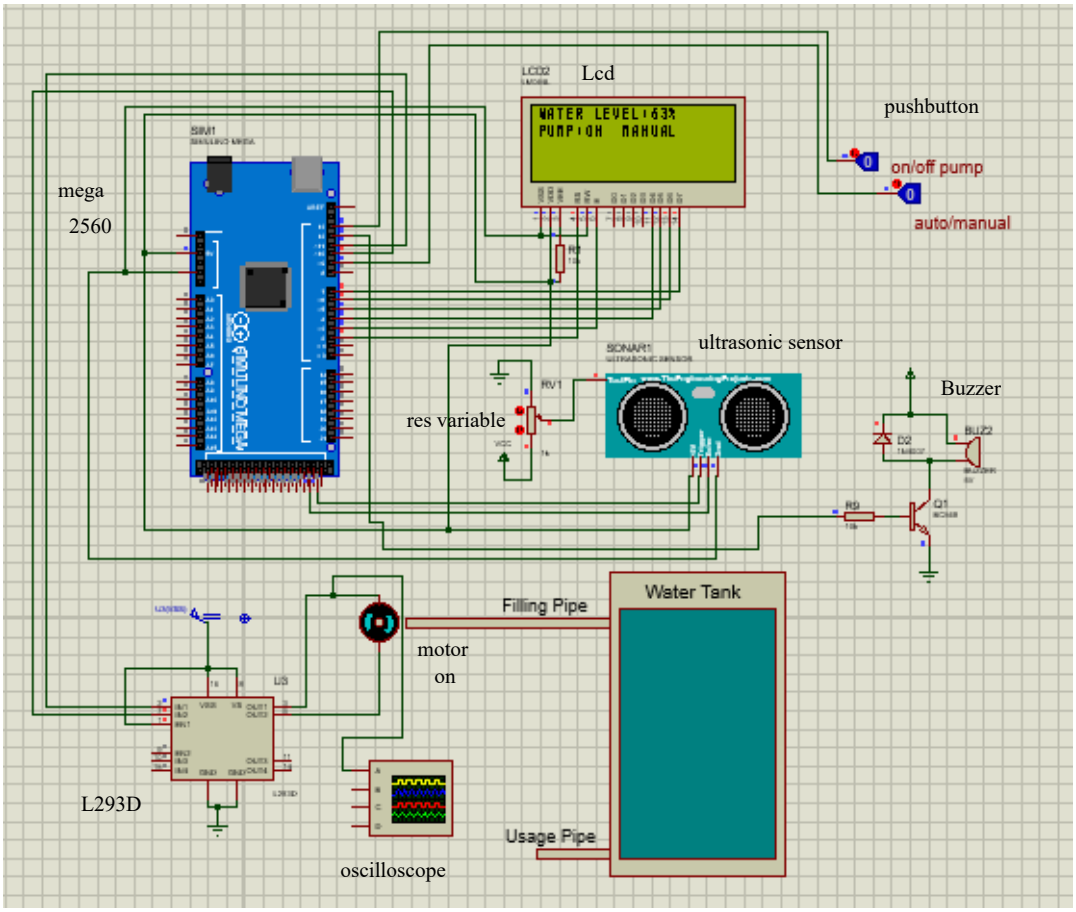

Figure IV.16 : Système en état manuel et pompe ON avec un niveau 63%.

Maniement le niveau d'eau est à 90%, pour arrêter le remplissage du réservoir nous appuyons sur le bouton poussoir on/off, est la pompe s'arrêter.

IV.2.3 Déclencher le Moteur

Nous allons maintenant aborder la commande d'un moteur à courant continu, par une PWM. La manière de s'y prendre diffère pour plusieurs raisons :

- Les moteurs employés en modélisme ferroviaire ont une tension d'alimentation qui est plus élevée que la tension de l'Arduino, généralement 12V ;

- Le courant nécessaire est également plus élevé, environ 10 fois, que ce que peut fournir une sortie de l'Arduino ;
- Un moteur est une charge dite inductive. C'est à dire que lorsque l'alimentation du moteur est coupée, le courant ne s'arrête pas instantanément de circuler. Il diminue progressivement.

La conséquence directe est qu'une sortie PWM de l'Arduino ne peut pas commander directement un moteur. Il est nécessaire d'amplifier le signal, à la fois en tension et en courant. L'amplificateur doit être adapté au courant consommé par le moteur et doit fournir la tension nécessaire. Elle doit aussi être capable de suivre la fréquence de la PWM.
Deux possibilités s'offrent à nous : utiliser un transistor MOSFET de puissance ou bien un pont en H. dons notre projet on a examiné la deuxième solution, en utilisant L293D.

Pour faire de la variation de vitesse d'un moteur DC, il est nécessaire de générer un signal analogique or l'arduino est équipée de broches analogiques en entrées mais ne peuvent pas être utilisées en sorties. Afin de varier artificiellement la tension en sortie de l'Arduino, il nous faudra utiliser certaines broches digitales de la carte. Mais par définition, une broche digitale ne délivre qu'un signal binaire (0 ou 1), c'est-à-dire que le signal électrique en sortie ne peut avoir que deux valeurs soit HIGH **(**niveau Haut = 5V) ou LOW **(**niveau Bas = 0V**).** Les broches digitales compatibles PWM vont permettre d'émettre un signal numérique disposant de 256 valeurs différentes (résolution de 8 bits soit $2^8 = 256$) et non plus seulement 2 valeurs (Tout ou Rien TOR ou binaire).

Pour générer une tension variable ou pseudo analogique en sortie d'une broche digitale de l'Arduino, il va falloir changer très rapidement l'état de la sortie. En effet, le fait de passer d'un état LOW à HIGH très vite cela va engendrer la variation de la valeur moyenne de la tension. Or, c'est ce changement de la valeur moyenne du signal électrique va changer ou varier la vitesse d'un moteur.

Le moteur (récepteur**)** ne percevra plus simplement l'état BAS (0V) ou l'état HAUT (5V), mais une multitude de tensions intermédiaires qui vont varier de 0V à 5V avec une résolution de 256 valeurs possibles. Ce pseudo signal analogique est possible car la fréquence de commutation entre les niveaux HAUT et BAS est élevée : 500 Hz soit 500 fois par seconde. Le changement d'état étant très rapide, le récepteur y est donc insensible, comme notre œil avec le clignotement des tubes fluorescent avec le signal alternatif sinusoïdal 50Hz. Afin de déterminer quelle tension moyenne on souhaite appliquer au récepteur, il faudra modifier le rapport cyclique, c'est-à-dire la durée du maintien au niveau HIGH ou HAUT du signal de sortie. Ainsi, le rapport cyclique pourra varier de 0% (tension

nulle 0V) jusqu'à 100% (Tension égale à 5V). Par contre au niveau de l'arduino, ce n'est pas une valeur comprise entre 0 et 100 qui sera réglée mais une valeur numérique comprise entre 0 et 255 (8 bits). En faisant une règle de trois ou un produit en croix, il est possible de déterminer pour chaque entier (PWM) compris entre 0 et 255 la valeur du rapport cyclique ou la tension moyenne en sortie de l'arduino :

- PWM = 0 ↔ Rapport cyclique = 0% ↔Tension = 0V ;
- PWM = 33 ↔ Rapport cyclique = 10% ↔ Tension = 0,5V ;
- PWM = 127 ↔0 Rapport cyclique = 50% ↔0 Tension = 2,5V ;
- PWM = 230 ↔ Rapport cyclique = 90% ↔ Tension = 4,5V ;
- PWM = 255 ↔ Rapport cyclique = 100% ↔ Tension = 5V.

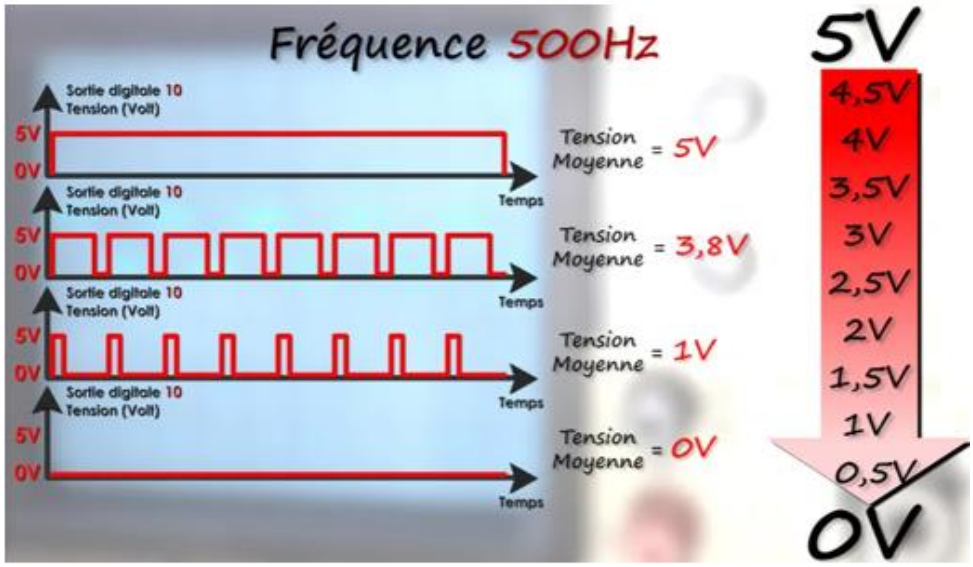

Figure IV.17 : PMW et le signal pseudo analogique en sortie de broche digitale de l'Arduino.

Pour visualiser le signal PWM et le changement du rapport cyclique en utilisont un oscilloscope analogique raccordé sur la sortie digitale 10 configurée en PWM**.**

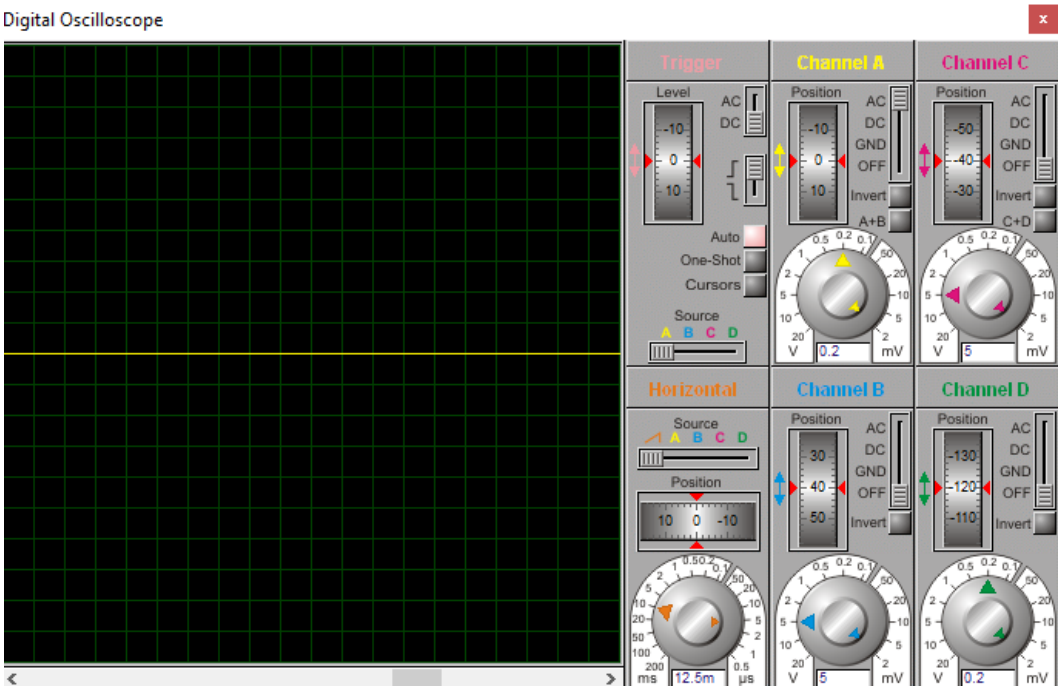

Figure IV.18 : Rapourt cyclique 0% - analogWrite(0).

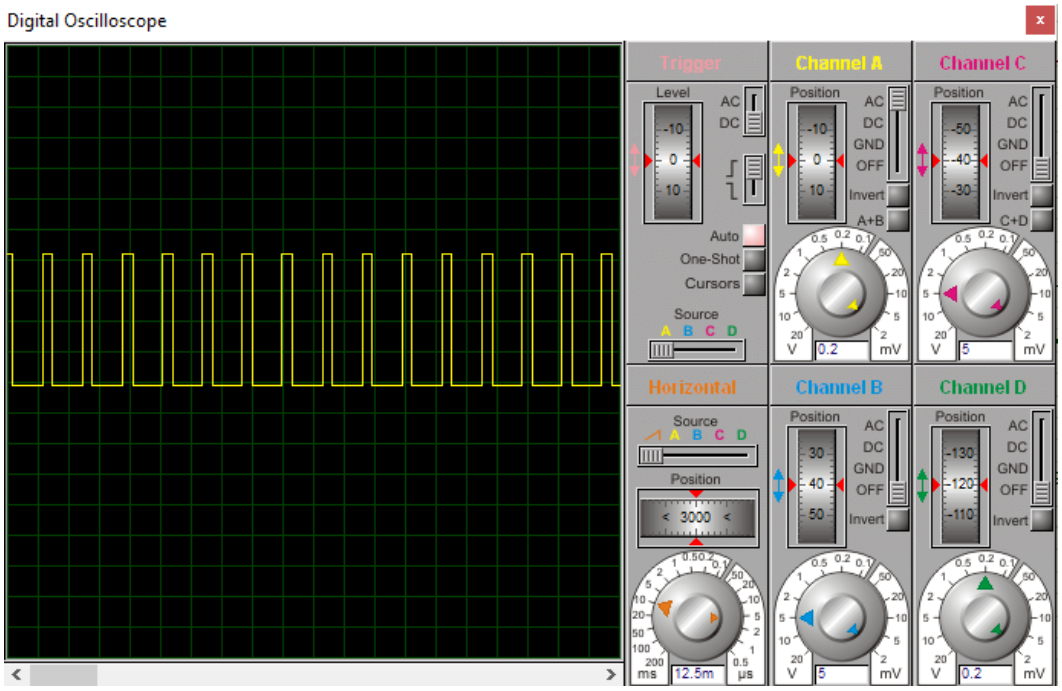

Figure IV.19 : Rapourt cyclique 25% - analogWrite(64).

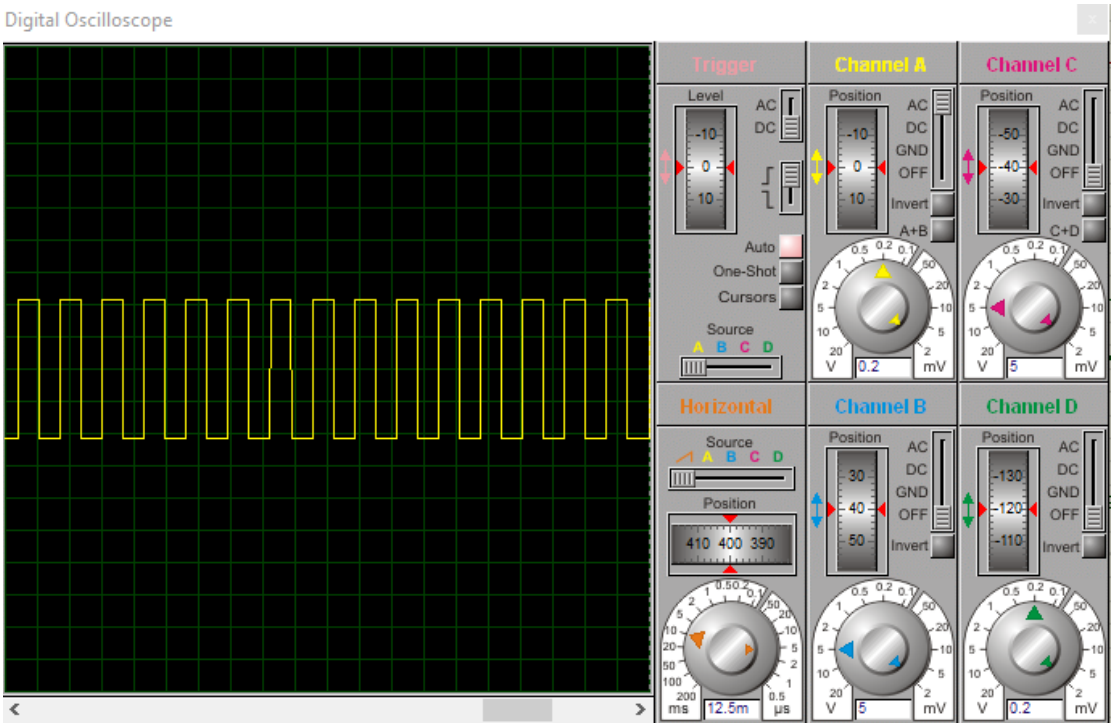

Figure IV.20 : Rapourt cyclique 50% - analogWrite(127).

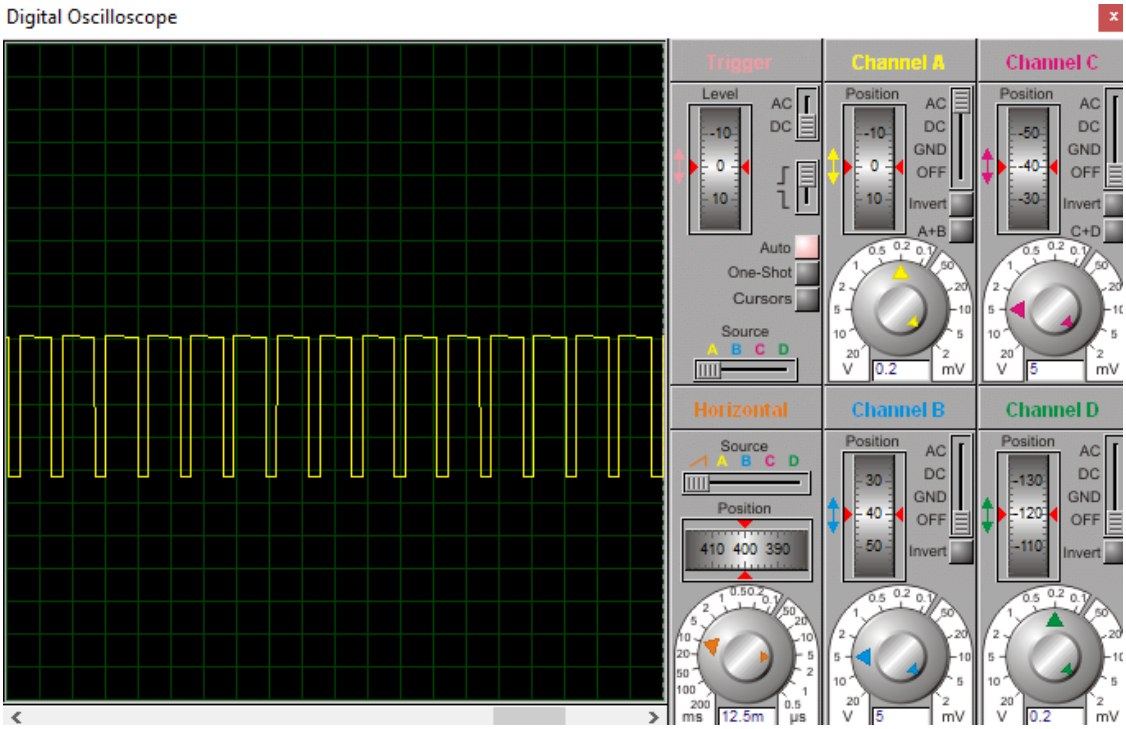

Figure IV.21 : Rapourt cyclique 75% - analogWrite(191).

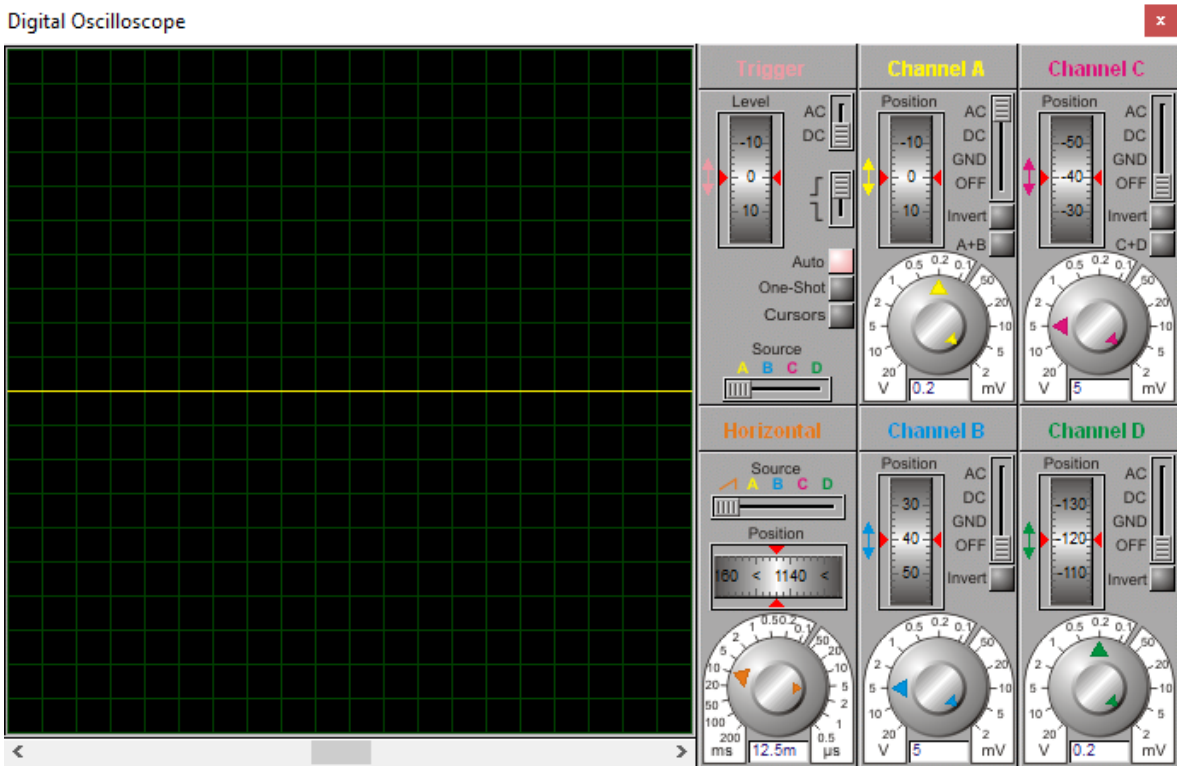

Figure IV.22 : Rapourt cyclique 100% - analogWrite(255).

IV Conclusion

Ce dernier chapitre est une démonstration et présentation de tous les résultats collecter pendent la simulation de notre système, dédie à la régulation de contrôle du niveau de réservoir d'eau. On a démontré en détail la conception et la mise en œuvre de la carte de contrôle. Et on a expliqué les différents bottons que nous avons utilisé pour le contrôle manuel, aussi le capteur que nous ont également aidés au contrôle automatique, le régulateur PWM pour la variation de vitesse du moteur MCC.

Conclusion générale

Conclusion générale

Dans la plupart des installations industriel et même domestique, lorsque l'on souhaite atteindre une certaine vitesse, température, débit, pression, niveau…etc. Il est très souvent nécessaire d'avoir recoure à une régulation et un contrôle.

Dans notre travail, nous avons simulé un système de contrôle automatique et manuel de remplissage d'un réservoir d'eau, pour passer de l'automatique vers le manuel ou l'inverse il faut juste appuyer sur le bouton poussoir (auto/manual). Premièrement le contrôle automatique, lorsque le niveau d'eau dans le réservoir est diminué au-dessous 20% la pompe est alimenté pour remplis le réservoir, après ça lorsque le réservoir est complètement remplis (le niveau est 100%) la pompe est arrêter. Deuxièmement le contrôle manuel, en utilise ce mode de contrôle en cas le niveau est égale ou plus que 20% et nous voulons remplir le réservoir, pour utilisé le il faut juste appuyer sur le bouton poussoir (on/off pump) et la pompe va alimenter et appuyer autre fois pour arrêter la pompe (arrêter le remplissage sur le niveau qui nous voulons).

Dans notre travail on à utilise la carte Arduino Mega pour piloter et commander tout les actions de remplissage, d'une part. Et pour fournir le signal PWM et faire la variation de la vitesse du moteur à courant continu (pompe) (pour annulé les perturbations), d'autre part. Pour cela, on a utilisé le Proteus Isis comme logiciel de simulation destiné à la simulation matérielle et le logiciel IDE Arduino pour programmer l'arduino mega 2560.

Ce projet était une occasion pour acquérir des connaissances sur un outil de programmation et simulation électronique et sur la conception des systèmes automatique de contrôle et de régulation, et voilà les points important de ce travaille :

- Le remplissage de réservoir automatiquement sans intervention humaine.
- Le mode de remplissage manuel pour les cas d'urgence.
- La variation de vitesse du moteur (pompe) pour annulé les perturbations qui entravent le fonctionnement du système.

Bibliographie

[1] Adnene TLILI, Safeyiddine KALLELI. 2014/2015. Asservissement et régulation. Cours Département Génie mécanique, Université de Iset, Nabeul.

[2] PROUVOSTPATRICK, 2010. 2éme édition Dunod. France ; Reims : Automatique contrôle et régulation.

[3] https://atelier-canope-95.canoprof.fr/eleve/Automates%20et%20robots/res/robot.dossierHtml/co/02clepsydreCtesibios_2.html

[4] https://gallica.bnf.fr/essentiels/evenement/automates-vaucanson.

[5] https://hieutr.hypotheses.org/315

[6] SLIMANI Sofiane, HACHOUR Ahcen. 2018. Contribution à l'Implémentation d'un Régulateur Flou Simplifié sous un API S7-300. Mémoire en master, Université Mouloud Mammeri, Tizi-Ouzou-Algérie.

[7] Adnene TLILI, Safeyiddine KALLELI, 2014/2015. Asservissement et régulation. Cours Département Génie mécanique, Université de Iset, Nabeul.

[8] Mohamed BOUASSIDA. 2010. Conception et réalisation. Cours.

[9] RESFA ABBES. 2020-2021. Régulation industrielle. Cours, université Abdelhamid Ibn Badis, Mostaganem, Algérie.

[10] S. Sekhsoukh, K. Oukili, Etude d'une boucle de régulation de niveau implémentation du régulateur et réglage du procédé, Projet de fin d'étude, école supérieur de technologie, 2010/2011.

[11] https://www.interface-z.com/conseil/pwm.php

[12] Vrancic D. et al., A new tuning method for PID Controllers, 4th IFAC conference on system structure and control, Bucharest, october 23-25, 1997.

[13] https://studylibfr.com/doc/2062954/modulation-%C3%A0-largeur-d-impulsion

[14] BOUICHE Hachemi, BRAHIMI Mohamed. Command PID d'un moteur à courant continu. Mini projet, Université Abderrahmane Mira, Bejaia, Algérie.

[15] Dr. BEKAKRA Youcef. Technique de commande. Cours, Université Echahid Hamma Lakhdar, El Oued, Algérie.

[16] Saidi Abdelhamide. 2012. La régulation. Cours.

[17] https://pecquery.wixsite.com/arduino-passion/arduino

[18] Eskimon, Olyte. 2012. Arduino pour bien commencer en électronique et en programmation.

[19] Jean-Noël Montagné. 2006. Initiation à la mise en oeuvre matérielle et logicielle de l'Arduino. Centre de Ressources Art Sensitif.

[20] Landrault S, Weisslinger H. 2014. Arduino : Premiers pas en informatique embarquée.

[21] http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.MaterielMega2560

[22] https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/742_decouverte-de-larduino/3416_le-logiciel/

[23] www.orbit-dz.com/product/capteur-ultrason-hc-sr04/

[24] Jean-Luc. 2015. Les écrans LCD alphanumérique.

[25] http://fr.wikipedia.org/wiki/Bipeur

[26] KRAMA Abdelbasset, GOUGUI Abdelmoumen. 2015. Etude et réalisation d'une carte de contrôle par Arduino via les systèmes Androide. Mémoire de Master Académique, Université Kasdi Merbah, Ourgla.

[27] TOURE Fatoumate Fousseyni REZIK Kheira. 2019. Contrôle et commande d'un système a distance via GSM. Mémoire de Magister, Université d'Abdelhamide Ibn Badis, Mostaganem.

[28] https://www.techno-science.net/definition/5799.html.

[29] AMANI Ahmed amine. 2019. Le contrôle pid d'un moteur. Mémoire de Master, Université de Badji Mokhtar, Annaba.

Annexes

1. Le code programme

```
#include <EEPROM.h>

#include <Wire.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(2,3,4,5,6,7);


long duration, inches;
int set_val,percentage;
bool state,pump;
int val =150; // de 0 à 255 (vitesse du motor)

void setup() {

  lcd.begin(20,4);
  lcd.print("End Of Study Project");
  lcd.setCursor(0,1);
  lcd.print("  Electromechanical  ");
  lcd.setCursor(0,2);
  lcd.print("     Department     ");
  delay(300);
  lcd.clear();
   lcd.print("    Ferkha Fadi ");
   lcd.setCursor(0,1);
   lcd.print("          &       ");
   lcd.setCursor(0,2);
   lcd.print(" Bennacer Abdennour ");
   delay(300);
   lcd.clear();
   lcd.setCursor(0, 0);
   lcd.print("WATER LEVEL:");
   lcd.setCursor(0, 1);
   lcd.print("PUMP:OFF MANUAL");

   pinMode(23, OUTPUT);
   pinMode(25, INPUT);
   pinMode(13, INPUT_PULLUP);
   pinMode(9, INPUT_PULLUP);
   pinMode(12, OUTPUT);
   pinMode(10, OUTPUT);
   pinMode(11, OUTPUT);

    set_val=EEPROM.read(0);
    if(set_val>150)set_val=150 ;
}
```

```
void loop() {
   digitalWrite(23, HIGH);
   delayMicroseconds(10);
   digitalWrite(23, LOW);
   duration = pulseIn(25, HIGH);
   inches = microsecondsToInches(duration);

   percentage=(set_val-inches)*100/set_val;

   lcd.setCursor(12, 0);
   if(percentage>100)percentage=100;
   if(percentage<0)percentage=0;
   lcd.print(percentage);
   lcd.print("%   ");

   if(percentage<20)pump=1;
   if(percentage>99)pump=0;
   if(pump==1)
   {
   analogWrite(10,val);
   analogWrite(11,0);
   digitalWrite(12, LOW);
   }
      if(pump==0)
   {
   analogWrite(10,0);
   analogWrite(11,0);
   digitalWrite(12, HIGH);
   delay(800);
   digitalWrite(12, LOW);
   }

   lcd.setCursor(5, 1);
   if(pump==1)lcd.print("ON ");
   else if(pump==0) lcd.print("OFF");

   lcd.setCursor(9, 1);
   if(!digitalRead(9))lcd.print("MANUAL");
   else lcd.print("AUTO   ");

    if(!digitalRead(13)&!state&digitalRead(9))
    {
      state=1;
      set_val=inches;
      EEPROM.write(0, set_val);
      }
```

```
93
94      if(!digitalRead(13)&!state&!digitalRead(9))
95      {
96         state=1;
97         pump=!pump;
98
99       }
100
101     if(digitalRead(13))state=0;
102
103
104     delay(500);
105
106 }
107 long microsecondsToInches(long microseconds) {
108    return microseconds / 29 / 2;
109 }
```

Résumé

Dans notre travaille nous avons régulé et contrôler le niveau d'eau d'un réservoir de stockage en utilisant la régulation TOR, d'autre manière détecter le niveau d'eau dans le réservoir et de le remplir automatiquement et manuellement. Ce système automatisé est implémenté puis simulé grâce au logiciel « Proteus Isis » avec la carte électronique « Arduino Méga » qui commande la marche et l'arrêt d'un moteur à courant continu MCC pour le remplissage d'eau, pour faire varier la vitesse du moteur MCC en exploitant le signale PWM dans le butest de contrôler le débit d'eau vers le réservoir. Avant tout ça nous avons parlé de la régulation TOR, le logiciel Proteus et la carte Arduino dans la partie théorique.

Mots clé : Régulation TOR, signale PWM, contrôle, moteur MCC, Proteus Isis, Arduino Mega.

ملخص

في عملنا هذا, قمنا بتنظيم و مراقبة مستوى المياه في خزان باستعمال التنظيم TOR, بطريقة أخرى تحديد مستوى المياه داخل الخزان و ملئه أليا و يدويا. هذا النظام الآلي تم إدماجه و محاكاته باستخدام برنامج «Proteus Isis» مع لوحة الكترونية « Arduino Méga », التي تتحكم في بدء و إيقاف المحرك بتيار مستمر لتعبئة الماء, لتغيير سرعة المحرك MCC تم استغلال الإشارة PWM بهدف التحكم في تدفق المياه إلى الخزان. قبل كل هذا قمنا بشرح التنظيم TOR , البرنامج Proteus , اللوحة Arduino , في الجزء النظري.

كلمات مفتاحية : تنظيم TOR , إشارة PWM, تحكم, محرك MCC, Arduino Mega , Proteus Isis.

Abstract

In our work we have regulated and controled the water level of a tank in other ways we detect the water level using the TOR regulation, otherwise detect the water level in the tank and fill it up automatically and manually. This automated system is implemented and simulated using «Proteus Isis» software with the «Arduino Mega» electronic card which controls the start and stop of a DC motor for filling water, the PWM signal is used to varying the speed of theDC motor, in order to control the flow of water to the tank. Before all that we have explained the TOR regulation, the Proteus software and the Arduino card in the theoretical part.

Key words : TOR regulation, PWM signal, control, DC motor, Proteus Isis, Arduino Mega.