

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master

Mention Électronique
Spécialité Instrumentation biomédicale

Présenté par

Achour Hamza

&

Ameur Islam

« Réalisation d'un pousse seringue électrique commandé par Arduino »

Proposé par : Dr. Anou Abderrahmane

Année Universitaire 2018-2019

Remerciements

*Dans le cadre De notre projet De fin D'étude pour
l'obtention Du diplôme de master 2 en Génie
biomédicale*

*Nous remercions tout d'abord Dieu pour ses
bienfaits inestimables, de nous avoir illuminé le
chemin vers le savoir et la science et de nous avoir
permis de terminer nos études ainsi que ce projet
dans de bonnes conditions.*

*Nous remercions les membres De jury qui ont
acceptés D'évaluer ce travail.*

*Nos remerciements à notre promoteur Dr. Anou
Abderrahmane pour leur encadrement ainsi qu'à
tous nos professeurs De Département D'électronique
De l'université SAAD DAHLEB de Blida 1 qui
ont contribués à notre formation.*

*Nos remerciements à nos parents pour leur soutien.
Aussi, nous tenons à remercier nos collègues et nos
amis pour leurs encouragements et à toute la
promotion Master 2.*

*Enfin, nous remercions tous ceux qui ont contribué
de près ou de loin à l'aboutissement De ce mémoire.*

Dédicace

A mes très chers parents

Votre confiance et vos encouragements ont été pour moi la première source de persévérance. Que vous trouvez dans ces modestes mots le témoignage de ma gratitude et ma sincère appréciation.

A mes très chères sœurs

A mon frère

Vous tenez une place immense dans mon cœur.

A toute ma famille

Recevez ici le témoignage de ma gratitude et de mon profond attachement.

A mes chers amis

Merci pour l'amitié, pour votre écoute, gentillesse et altruisme. Les moments que nous avons passé ensemble resteront un agréable souvenir.

Islam.

Dédicace

A mes très chers parents

Votre confiance et vos encouragements ont été pour moi la première source de persévérance. Que vous trouvez dans ces modestes mots le témoignage de ma gratitude et ma sincère appréciation.

A mes très chères sœurs

A mon frère

Vous tenez une place immense dans mon cœur.

A toute ma famille

Recevez ici le témoignage de ma gratitude et de mon profond attachement.

A mes chers amis

Merci pour l'amitié, pour votre écoute, gentillesse et altruisme.

Les moments que nous avons passé ensemble resteront un agréable souvenir.

Hamza.

Résumé

Résumé :

La perfusion effectuée manuellement par le personnel infirmier peut être remplacée par un dispositif médical nommé le pousse seringue électrique (PSE) qui est une pompe à perfusion programmables, en infusant de manière précise et régulière (ml/heures) des solutions injectable (médicaments ou nutriments) dans le système circulatoire d'un patient souvent par voie intraveineuse. Le but de ce travail est donc de développer un prototype de pousse seringue électrique piloté, ce dernier combine des parties mécaniques, électriques et électroniques. La partie mécanique comprend un berceau et un piston qui vont recevoir le corps de la seringue, Cette partie mécanique est pilotée par un moteur électrique pas à pas. Enfin la partie électronique capable de vérifier les débits et les doses pour les quatre modes d'administration et cela a l'aide d'un afficheur LCD, boutons et une carte Arduino.

Mots clés : pousse seringue PSE, moteur pas à pas, Arduino, LCD.

Abstract:

The infusion performed manually by nurses can be replaced by a medical device called the syringe pump, which is a programmable infusion pump that can regularly infuse precise quantities of injectable solutions (Drugs or nutrients) in the circulatory system of a patient, usually intravenously. The purpose of this work is to develop a prototype of a syringe pump, which combines mechanical, electrical and electronic parts. The mechanical part comprises a cradle and a piston that will receive the body of the syringe, this mechanical part is driven by a stepper motor, Finally the electronic part can check rates and doses for the four modes of administration, Using an LCD display, buttons and an Arduino board.

Keywords: Syringe pump, stepper motor, Arduino, LCD.

ملخص:

الحقن اليدوي من قبل الممرض الشخصي قد يستبدل بجهاز طبي مسمى دافع الحقنة الكهربائي PSE الذي هو عبارة عن مضخة حقن مبرمجة بطريقة دقيقة ومنتظمة (مل/ساعة). يتم حقن السوائل القابلة للحقن (أدوية أو مغذيات) في الدورة الدموية للمريض عادة عن طريق الوريد. الهدف من هذا العمل هو تطوير نموذج دافع حقنة كهربائي متحكم فيه، هذا الأخير يجمع أجزاء ميكانيكية، كهربائية وإلكترونية. الجزء الميكانيكي يحوي مكبس وأساس لوضع مجسم الحقنة، هذا الجزء يقوده محرك خطوي. أخيراً الجزء الإلكتروني قادر على التحقق من التدفقات والجرعات لأنماط الحقن وهذا عن طريق شاشة LCD محرك خطوي، أزرار ولوحة أردوينو. كلمات مفتاحية: دافع حقن كهربائي PSE محرك خطوي، لوحة أردوينو

Listes des acronymes et abréviations

PSE : Pousse Seringue Electrique.

AIVOC : Anesthésie Intraveineuse à Objectif de Concentration.

PCA : Patient Controlled Analgesia (analgésie contrôlée par le patient).

PCEA : Patient Controlled Epidural Anesthesia (anesthésie épidurale contrôlée par le patient)

Atm : atmosphère (unité de pression).

PWM : Pulse Width Modulation (modulation de largeur d'impulsion).

LCD : Liquid Crystal Display (affichage à cristaux liquides).

USB : Universal Serial Bus (bus universel en série).

IDE : Integrated Development Environment (environnement de développement intégré).

GND : Ground (la terre).

VCC : Voltage Courant Continu.

EPROM : Erasable Program Read Only Memory (mémoire effaçable en lecture seule).

RAM : Random Acces Memory (mémoire vive).

AC : Alternating Current (Courant alternatif).

DC : Direct Current (courant continu).

Liste des figures

Chapitre 1

Figure 1.1 : Pousse seringue électrique commercialisé model PILOTE A2.	2
Figure 1.2 : Principe de pousse seringue électrique.	3
Figure 1.3 : Seringue simple voie.	5
Figure 1.4 : Exemple de pousse seringue double voies.	6
Figure 1.5 : Mode d'administration des PSE (mode continu) [4].	7
Figure 1.6 : Mode d'administration des PSE (le mode intermittent) [4].	7
Figure 1.7 : Mode d'administration des PSE (le mode séquentiel) [4].	8
Figure 1.8 : Mode d'administration des PSE (mode d'analgésie auto contrôlée) [4]. ..	8
Figure 1.9 : Mode d'administration des PSE (nutrition parentérale totale) [4].	9
Figure 1.10 : Mode d'administration des PSE (mode simultané) [4].	9
Figure 1.11 : Mode d'administration des PSE (mode alterné) [4].	10

Chapitre 2

Figure 2.1 : Module Arduino.	16
Figure 2.2 : Le microcontrôleur ATmega328.	20
Figure 2.3 : Les ports de communications Arduino.	24
Figure 2.4 : l'interface de IDE Arduino.	25
Figure 2.5 : Moteur pas à pas.	25
Figure 2.6 : Structure interne d'un moteur à aimant permanent.	27
Figure 2.7 : Moteur unipolaire à aimant permanent.	27
Figure 2.8 : Moteur bipolaire à aimant permanent.	28
Figure 2.9: Moteur à réluctance variable.	29
Figure 2.10 : Moteur hybride.	30

Chapitre 3

Figure 3.1 : Schéma simplifié de système pousse seringue.	31
--	----

Figure 3.2 : Schéma synoptique du prototype.....	32
Figure 3.3 : Synoptique de l'alimentation stabilisée.....	33
Figure 3.4 : Transformateur monophasé.....	33
Figure 3.5 : Représentation d'un redresseur à double alternance.....	34
Figure 3.6 : Le circuit et le signal de filtrage.....	34
Figure 3.7 : Circuit de branchement des régulateurs.....	35
Figure 3.8 : Schéma électrique de l'alimentation stabilisé linéaire.....	35
Figure 3.9 : Brochage d'Arduino UNO.....	37
Figure 3.10 : Le fonctionnement du moteur dans le prototype.....	38
Figure 3.11 : Le circuit intégré L293D.....	39
Figure 3.12 : Ecran LCD 20X4.....	40
Figure 3.13 : Module d'interface I2C.....	41
Figure 3.14 : Brochage de LCD par le module d'interface I2C.....	41
Figure 3.15 : Boutons poussoir relié à des résistances.....	42
Figure 3.16 : Schéma électrique de notre circuit.....	42
Figure 3.17 Organigrammes.....	46
Figure 3.18 : Déroulement du programme.....	47

Chapitre 4

Figure 4.1 : Initialisation LCD.....	54
Figure 4.2 : Recul du curseur pour avoir la position 0.....	54
Figure 4.3 : Message commencez.....	55
Figure 4.4 : Menu principal.....	55
Figure 4.5 : Type de seringue.....	56
Figure 4.6 : Ajuster le curseur.....	56
Figure 4.7 : Menu mode d'injection.....	57
Figure 4.8 : Mode continue.....	57
Figure 4.9 : Mode intermittent.....	58
Figure 4.10 : Mode direct.....	58
Figure 4.11 : Mode auto contrôle.....	59
Figure 4.12 : Fin d'injection.....	59

Liste des tableaux

Chapitre 2

Tableau 2.1 : Caractéristiques de La carte Arduino UNO.	18
Tableau 2.2 : Mode d'alimentation du moteur unipolaire.	28
Tableau 2.3 : Mode d'alimentation du moteur bipolaire.	29

Chapitre 3

Tableau 3.1 : Brochage du L293D.	39
Tableau 3.2 : Pins du LCD.	40
Tableau 3.3 : Types de variables.	48

Chapitre 4

Tableau 4.1 : Test du mode continue.	53
Tableau 4.2 : Test du mode direct.	53

Sommaire

Introduction générale	1
Chapitre 1.....	1
1.1 Introduction	2
1.2 Définition.....	2
1.3 Principe de fonctionnement	3
1.3.2 Le système de réglage du type de seringue	4
1.3.3 Le système de détection de pression	4
1.4 Les différents types des PES.....	5
1.4.1 Pousse seringue à simple voie	5
1.4.2 Pousse seringue à double voie	5
1.4.3 Pousse seringue hyper-barrique.....	6
1.4.4 Stations de perfusion	6
1.4.5 Pousse seringue entérale.....	6
1.5 Les différents modes d'administration des PSE.....	7
1.5.1 Le mode perfusion continue.....	7
1.5.3 Mode circadien ou séquentiel	8
1.5.4 Mode d'analgésie auto contrôlée.....	8
1.5.6 Mode simultané.....	9
1.5.7 Mode alterné	9
1.6 Applications médicales des PSE	10
1.7.2 Cardiologie et néonatalogie.....	11
1.8 Modalités d'utilisation	11

1.9	Débits des perfusions.....	12
1.10	Avantages et inconvénient d'un PSE.....	12
1.11	Conclusion	13
Chapitre 2.....		2
2.1	Introduction	15
2.2	Partie électronique : la carte Arduino.....	15
2.2.1	Définition du module Arduino	15
2.2.2	Les gammes de la carte Arduino.....	16
2.2.3	Caractéristiques de La carte Arduino UNO.....	18
2.2.4	Pourquoi Arduino UNO ?	19
2.2.5	La constitution de la carte Arduino UNO.....	20
2.3	Partie mécanique : Les moteurs pas à pas.....	25
2.3.1	Utilisation des moteurs pas à pas.....	26
2.3.2	Avantage et inconvénients	26
2.3.3	Les différents types des moteurs pas à pas.....	26
2.4	Conclusion	30
Chapitre 3.....		15
3.1	Introduction	31
3.2	Partie Hardware	32
3.2.1	Circuit d'alimentation stabilisé	32
3.2.2	Schéma électrique de l'alimentation isolée sur ISIS.....	35
3.2.3	La carte Arduino UNO	36
3.2.4	Moteur pas à pas	37
3.2.5	Ecran LCD	40
3.2.6	Le clavier	41

3.3	Partie Software	42
3.3.1	Structure globale du programme	47
3.4	Conclusion	50
	Chapitre 4.....	30
4.1	Introduction	51
4.2	La sensibilité de prototype.....	51
4.3	Phase des tests.....	53
4.3.1	Mode continue.....	53
4.3.2	Mode direct	53
4.4	Mode d'emploi.....	54
4.4.1	Menu principal.....	55
4.4.2	Type de seringue.....	55
4.4.3	Ajuster le curseur	56
4.4.4	Mode d'injection.....	56
4.5	Conclusion	59
	Conclusion générale.....	60
	Bibliographie	62
	Annexes.....	64

Introduction générale

Dans la vie, l'homme a toujours cherché à développer et à améliorer les conditions de vie, en appliquant les progrès des sciences et de la technologie à la médecine.

L'ingénieur biomédical supervise une équipe de techniciens, afin d'assurer la conception, l'optimisation et le fonctionnement des équipements médicaux, en définissant la stratégie de maintenance par rapport à un budget et aux normes en cours.

Dans ce projet, on s'intéresse à une des applications de l'électronique dans le domaine médical, plus exactement dans la réanimation et les blocs opératoires.

Notre travail consiste à étudier et à réaliser un prototype de pousse seringue, avec ces différents éléments, à savoir le moteur pas à pas et la carte Arduino. Il s'articule sur deux parties électronique et mécanique. La partie électronique transforme les commandes envoyées par l'utilisateur en impulsions électriques, pour piloter le moteur, produisant à son tour un mouvement mécanique de translation. Ce mouvement va faire bouger la seringue dans une direction, avec un temps bien déterminé, et un volume d'injection bien choisi. La seringue et le moteur forme notre partie mécanique.

Notre mémoire est organisé en quatre parties. Le premier chapitre, on commencera par définir quelques généralités sur les pousSES seringues, tout ce qui est nécessaire à connaître, et le domaine d'usage, suivi par les différents types existants, et leurs modes d'administrations. Le deuxième chapitre portera sur l'étude des moteurs pas à pas, qui est notre composant de base, et la carte Arduino, qui représente le cerveau de notre projet. Dans le chapitre trois, nous schématiserons notre système sous forme d'un circuit électronique et on essayera d'expliquer son fonctionnement. Finalement un quatrième chapitre qui englobera les différents tests appropriés pour chaque élément de notre circuit, tout en définissant le rôle et le schéma électronique qui lui correspond.

Chapitre 1

Généralité sur le Pousse seringue

1.1 Introduction

Le pousse seringue électrique, ou la seringue auto-pousseuse (anglais : Syringe pump, syringe infusion pump), sont utilisés pour administrer au patient une thérapie à débit constant ou à débit variable, à des vitesses lentes voir très lentes, des solutions concentrées. Ils doivent vaincre la contre-pression due aux résistances, à l'écoulement au sein des lignes de perfusion (tubulure, prolongateur, filtres, cathéter...), et aux pressions intra vasculaires régnant au niveau du site d'injection. Une énergie mécanique fournie par un moteur électrique est utilisée pour pousser sur le piston d'une seringue, afin de transférer la solution contenue vers la circulation sanguine du patient.

L'utilisation d'un pousse seringue électrique, PSE, permet de perfuser un produit médicamenteux injectable, de façon précise et régulière, sur une longue période. On parle également de pompe à débit constant.

1.2 Définition

Un pousse seringue électrique (PSE) est par définition un appareil qui permet d'obtenir un débit constant ou variable d'un médicament avec une vitesse déterminée. La vitesse de déplacement du piston de la seringue varie selon une programmation définie par l'utilisateur.

On peut utiliser plusieurs modèles et tailles de seringue [1]. La figure 1.1 illustre un modèle de pousse seringue électrique commercialisé le **PILOTE A2**.



Figure 1.1 : Pousse seringue électrique commercialisé model PILOTE A2.

1.3 Principe de fonctionnement

Dans le model pilote A2 représenté sur la figure 1.2. Le système combine des parties mécaniques, électriques et électroniques. La partie mécanique sert de support pour les différents types de seringues. Elle comprend un berceau qui va recevoir le corps de la seringue, le berceau est généralement muni d'un capteur et d'une encoche pour verrouiller la seringue. La collerette du piston se fixe sur le chariot du piston de seringue au moyen de griffes. Il comprend également un système décapeur qui va permettre de vérifier la bonne position de fixation du piston. Le piston du PSE se déplace grâce à un système de vis sans fin, qui va littéralement pousser le contenu de la seringue vers le circuit patient, cette partie mécanique est munie d'un moteur électrique pas à pas alimenté soit par le secteur, soit par une batterie. Il est commandé par un système électronique permettant de signaler des dysfonctionnements grâce à des alarmes.

L'utilisation d'une batterie est primordiale pour la sécurité, puisqu'elle permet une administration continue, même en cas de coupure de courant, enfin la partie électronique, gère l'ensemble du système. Cette partie fonctionne comme un véritable petit ordinateur capable non seulement de vérifier les débits, les pressions, mais également d'effectuer de nombreux calculs de doses en fonction de protocoles divers.

[2]

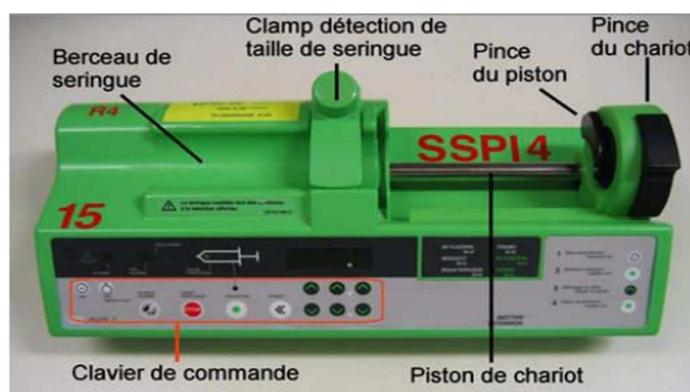


Figure 1.2 : Principe de pousse seringue électrique.

1.3.1 Le système d'entraînement

Le système d'entraînement repose sur le principe d'un ensemble vis/écrou, muni d'un moteur et d'une tige filetée. Il permet de pousser le piston d'une seringue, de diamètre

Chapitre 1 : Généralité sur le Pousse seringue

connu, de manière linéaire. Un capteur potentiométrique entraîné par le mouvement du poussoir permet d'avoir la position en tout moment.

1.3.2 Le système de réglage du type de seringue

Chaque type de seringue nécessite un réglage spécifique du pousse-seringue. Les pousse-seringues qui permettent l'utilisation de plusieurs types de seringues disposent d'un système de détection automatique de la taille de seringue. Il ne reste plus qu'à sélectionner manuellement la marque de la seringue, cela permet de limiter les erreurs de configuration.

Une fois cela fait, le pousse-seringue intègre, par rapport à des abaques, les caractéristiques de la seringue et ajuste les vitesses de déplacement du poussoir. Le réglage du débit d'un pousse seringue peut être effectué soit directement (débit), soit indirectement par le couple de paramètres temps et volume. Quel que soit le mode de réglage, l'erreur sur le débit sera inférieure ou égale à 2%, puisque la précision sur la vitesse d'avance du piston et la précision du diamètre intérieur de seringue sont inférieures ou égales chacune à $\pm 1\%$.

1.3.3 Le système de détection de pression

Les techniques de détection de pression peuvent se diviser en deux :

- Capteur de pression (jauge de contrainte) intégré au poussoir du pousse-seringue, ou il y a génération d'une tension proportionnelle à la force appliquée sur le poussoir ;
- Un système mécanique utilisant un ressort taré dont on règle la compression.

Une surveillance des évolutions de la pression de perfusion est possible, grâce à un indicateur du niveau de pression, pour les appareils utilisant un capteur de pression par jauge de contrainte, permettant ainsi de prévenir l'imminence d'une occlusion. Cette fonction est généralement une option.

Le pousse seringue électrique (PSE) est un moyen pratique et précis permettant l'administration des médicaments à débit continu et constant. Différentes marques sur le

Chapitre 1 : Généralité sur le Pousse seringue

marché sont disponibles, le volume est variable (10, 20, 30, 50 et 60 ml). Le débit peut être réglé entre 0,1 à 99,9 ml/heure. Il existe des appareils à une ou deux seringues. Les plus utilisés en médecine d'urgence sont les appareils à batterie permettant le transport du malade. Le système d'alarme permet de détecter une obstruction de la tubulure, une fuite, ou l'arrêt de la perfusion [3].

1.4 Les différents types des PES

Il existe des pousse-seringues à une simple ou plusieurs voies (généralement deux), ce qui permet une injection en différents sites de différents médicaments à un même patient.

1.4.1 Pousse seringue à simple voie

Un pousse seringue à simple voie ou mono voie (Figure 1.3), est un dispositif qui permet d'injecter une seule solution à la fois par exemple : Dopamine, le support de ce dernier peut acquérir une seule seringue de n'importe quel dosage, on les trouve généralement dans les blocs opératoires [4].



Figure 1.3 : Seringue simple voie.

1.4.2 Pousse seringue à double voie

Le même que celui d'une simple voie, mais ce dernier (Figure 1.4) peut acquérir et gérer deux seringues avec deux solutions, il permet de faire la fusion des solutions ou bien gérer chaque seringue indépendamment [4].



Figure 1.4 : Exemple de pousse seringue double voies.

1.4.3 Pousse seringue hyper-barrique

Les pousSES seringues hyper-barriques répondent aux exigences de sécurité indispensables et spécifiques de la médecine hyperbare (pressions élevées, haute concentration en oxygène). Ils sont conçus et certifiés pour être utilisés sous 6 Atm, et sont soumis aux mêmes contraintes qu'un pousse seringue standard [5].

1.4.4 Stations de perfusion

De composition modulaire, elles servent à organiser au chevet du patient la combinaison de voies de pousse-seringues nécessaires à la thérapie, et de faire évoluer les stations de perfusion au rythme des traitements administrés au patient. Elles peuvent aussi intégrer des pompes à perfusion. Ces systèmes sont composés de pousse-seringues, montés en parallèle, et interconnectés sur une base, qui centralise les commandes, et les informations d'état des perfusions. Ces dispositifs peuvent être utilisés pour l'administration synchronisée d'hypnotique et de morphinique en AIVOC (Anesthésie Intraveineuse à Objectif de Concentration) [5].

1.4.5 Pousse seringue entérale

Il est conçu pour délivrer de petits volumes à faible débit (0,1 à 200 ml/h), utilisé pour la nutrition entérale des bébés et prématurés (il utilise des seringues à usage oral embout conique "cathéter" ou embout "luer lock" femelle) [5].

1.5 Les différents modes d'administration des PSE

Un pousse seringue peut fonctionner selon un ou plusieurs modes :

1.5.1 Le mode perfusion continue

C'est le plus simple, le plus basique et le plus utilisé. La très grande majorité des PSE sont destinés à cet usage. Il suffit de régler un débit en millilitres par heure et l'appareil le délivre. Les PSE modernes proposent de plus en plus de régler une dose/kg/heure (voir par minute ou par 24H), mais sans effectuer le calcul de posologie. C'est à dire que c'est l'opérateur lui-même qui détermine la dose et non le PSE qui va la calculer selon un protocole (Figure 1.5) [4].

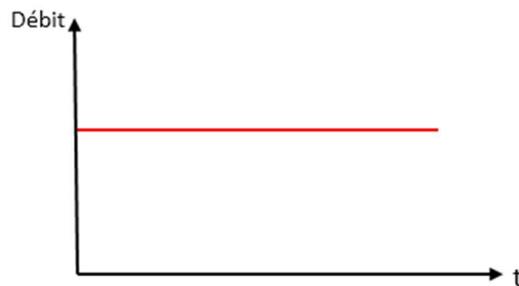


Figure 1.5 : Mode d'administration des PSE (mode continu) [4].

1.5.2 Le mode Intermittent

C'est un mode avec une administration intermittente et une fréquence d'administration régulière ou le volume de dosage est stable. Ce mode est utilisé dans des applications du traitement à intervalles (ex. antibiotiques). La figure 1.6 illustre ce mode.

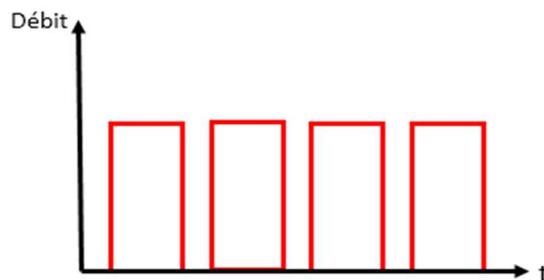
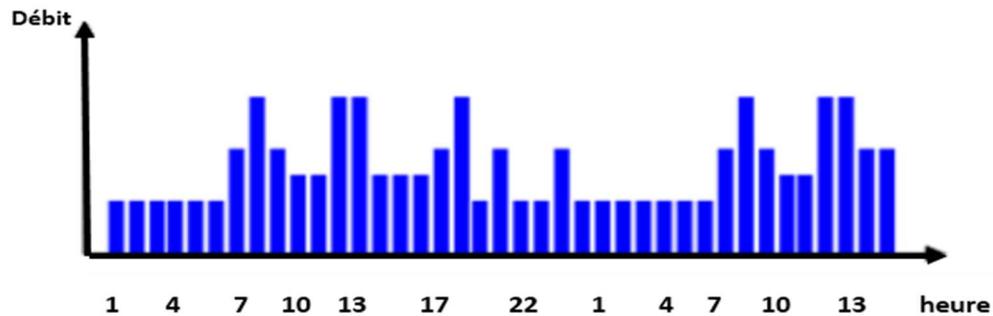


Figure 1.6 : Mode d'administration des PSE (le mode intermittent) [4].

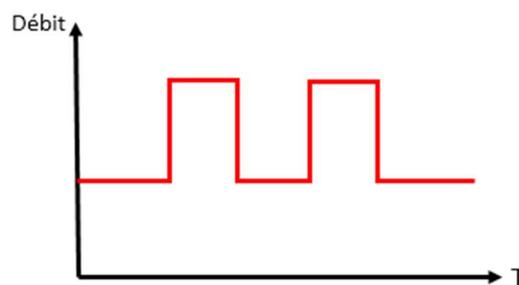
1.5.3 Mode circadien ou séquentiel

C'est un mode avec un débit pour chaque cycle et une unité de programmation qui dure généralement 1h. On le trouve dans les applications hormonales (Hormonothérapie) et la chronothérapie, son intérêt majeur c'est son adaptation au biorythme (Figure 1.7).



1.5.4 Mode d'analgésie auto contrôlée

C'est un mode avec un débit continu (Figure 1.8), programmé avec un ajout ou pas d'un débit qu'on appelle BOLUS à la demande du patient. Cette dose sera administrée chaque fois que le patient effectue une demande acceptée par la pompe (Dose bolus). On utilise ce mode lors d'analgésie contrôlée par le patient, un PCA (douleur aiguë ou chronique) site d'abord d'intraveineux et un PCEA (Patient Controlled Epidural Anesthesia) site d'abord en péri-durale, dont le but est le soulagement de la douleur contrôlée par le patient.



1.5.5 Nutrition Parentérale Totale

La nutrition parentérale est une technique d'assistance nutritive par voie veineuse, exclusive ou complémentaire. Elle peut s'effectuer à l'aide d'une pompe ou d'un pousse seringue. L'augmentation progressive et diminution progressive du débit est possible pour des préconisations thérapeutiques spécifiques (Figure 1.9).

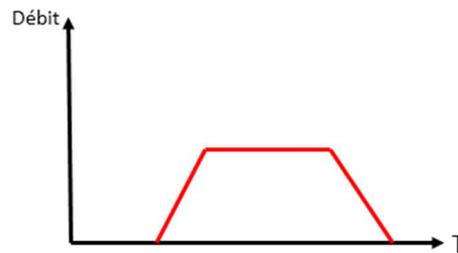


Figure 1.9 : Mode d'administration des PSE (nutrition parentérale totale) [4].

1.5.6 Mode simultané

Il est possible d'administrer plusieurs produits en même temps avec des débits différents (pompe double voies simultanées ou stations de perfusion) (Figure 1.10).



Figure 1.10 : Mode d'administration des PSE (mode simultané) [4].

1.5.7 Mode alterné

Dans le même principe, l'administration peut s'effectuer de manière alternée (Figure 1.11) (avec les pompes à double voies alternées ou les stations de perfusion). Son application peut se voir dans la : Réanimation, Pédiatrie et Anesthésie. Tandis que son Intérêts est de faire une perfusion adaptée en fonction de la masse corporelle du patient.

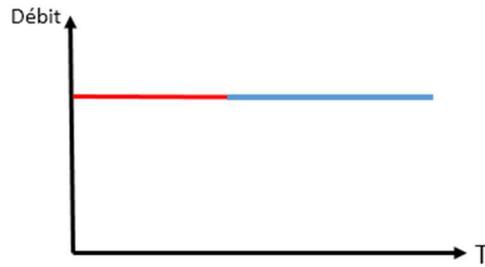


Figure 1.11 : Mode d'administration des PSE (mode alterné) [4].

1.6 Applications médicales des PSE

Cet appareil permet de perfuser en continu, à débit constant, un soluté ou un médicament, dans un large domaine d'application, parmi eux, on peut citer :

- Anesthésie et réanimation
- Bloc opératoire
- Urgences
- Cardiologie
- Néonatalogie

Dans ces domaines d'applications on va injecter différents médicaments pour chaque opération.

1.7 Anesthésie et réanimation

L'anesthésie-réanimation est la branche de la médecine qui se consacre à :

- **L'anesthésie** : abolition partielle ou totale de la sensibilité douloureuse (assortie dans le cas de l'anesthésie générale d'une perte de conscience et d'une assistance vitale) permettant la réalisation sans mémorisation, et sans douleur des interventions chirurgicales et des actes médicaux douloureux ou invasifs.
- **La réanimation** : prise en charge des patients présentant ou susceptibles de présenter une ou plusieurs défaillances viscérales aiguës mettant directement en jeu le pronostic vital.

Chapitre 1 : Généralité sur le Pousse seringue

1.7.1 Bloc opératoire et urgence

Les PSE jouent un très grand rôle dans les applications médical, que ce soit en Bloc opératoire ou en urgence, on peut citer le type de PSE le plus utilisé dans ces domaines : pousses seringues d'anesthésie qui sont du même type que les PSE utilisés dans la réanimation [6].

1.7.2 Cardiologie et néonatalogie

Le Pousse Seringue est moins utilisé dans les applications médicales en cardiologie (traitement d'urgence devant une tachycardie ventriculaire par la xylocaïne), mais très utilisé dans le service néonatalogie.

Les bébés nés prématurément (avant 37 semaines) peuvent être incapables de coordonner la succion, la déglutition et la respiration, et avoir besoin d'être alimentés (recevoir la quantité d'eau, de nutriments et de vitamines dont il a besoin pour grandir). La parentérale sera perfusée par voie intraveineuse grâce aux pousses-seringues. Dès que possible, le lait sera acheminé par intermittence dans l'estomac à travers un tuyau passant par le nez ou la bouche.

Les bolus intermittents de lait peuvent être administrés à l'aide d'un pousse seringue qui pousse doucement le lait dans l'estomac du nourrisson (alimentation par pousse) [7].

1.8 Modalités d'utilisation

Pour les services de soins généraux, l'utilisation des pousses-seringues en routine, reste basique. La perfusion d'un produit via un pousse seringue doit répondre à quelques règles d'utilisation afin d'être réalisée avec efficacité.

Tout d'abord, le matériel est soumis aux règles de matériau vigilance et doit être révisé selon un tableau périodique. Une chute, même sans casse visible de l'extérieur, impose un passage en révision. De nombreux éléments internes sont fragiles et leur altération peut nuire de façon importante au fonctionnement du PSE. Les batteries doivent également faire l'objet d'une maintenance régulière, et d'un remplacement lorsque l'autonomie n'est plus suffisante [8].

Chapitre 1 : Généralité sur le Pousse seringue

Ces appareils sont tarés pour fonctionner à plus ou moins un mètre de hauteur par rapport au cœur du patient. La fiabilité de mesure des pressions d'administration en dépend. Ils ne doivent pas être placés en position verticale. Ceci afin d'éviter le passage de bulles résiduelles dans le système de perfusion. Il n'existe pas de système de détection de bulles comme pour les pompes volumétriques [8].

Le matériel utilisé avec les PSE doit être adapté aux pousses seringues car chaque modèle n'est compatible qu'avec un certain nombre de consommables. Utiliser un consommable non validé par le fabricant expose à des risques d'erreur de dose, et d'alarmes. Les tests de précision et de résistances aux frictions sont effectués pour des matériels spécifiques [8].

1.9 Débits des perfusions

Le débit d'une perfusion est l'action de calculer le volume précis de liquide écoulé dans un temps donné, à l'aide d'un système de perfusion. Le débit de perfusion joue un rôle essentiel dans l'injection, cependant, il peut être limité par l'utilisation de cathéters de faible diamètre. Pour perfuser rapidement de grands volumes, le médecin pourra mettre en place une voie veineuse dite centrale, avec un accès sur de gros vaisseaux.

1.10 Avantages et inconvénient d'un PSE

La fiabilité de ces dispositifs repose essentiellement sur la qualité (constance et précision) des débits annoncés et mesurés. On ne doit enregistrer aucun changement de rythme de perfusion qui n'ait été programmé. La prise en une seule injection du médicament ne permet pas de maintenir un effet optimal et constant de l'action thérapeutique. Au cours des premières minutes qui suivent une injection unique la concentration peut atteindre une valeur élevée, pouvant provoquer dans certains cas des incidents graves.

C'est pourquoi on lui préfère la méthode des injections multiples à doses réduites, administrées en continu ou à intervalles de temps régulièrement espacés. Cependant l'injection à intervalles de temps régulièrement espacés présente les inconvénients suivants :

Chapitre 1 : Généralité sur le Pousse seringue

- Accroissement du nombre de manipulations et des risques d'erreurs ;
- Interventions plus fréquentes du personnel infirmier ;
- Augmentation des risques septiques ;
- Contraintes pour le patient.

De plus, l'utilisation du pousse seringue pour des injections continues permet une injection lente et très précise de l'agent thérapeutique [9]. Généralement les avantages et les inconvénients des PES sont :

- Précision, facilité de mise en place et perfusion de grands volumes.
- Matériel adapté aux médicaments photosensibles (Lasix).
- Les différentes alarmes présentes sur les pousSES seringues :
 - Débrayage mécanique ;
 - Contrôles de l'appareil : déconnexion secteur, erreurs techniques, autotest, rotation moteur, maintenance préventive ;
 - Contrôle perfusion : alarme et pré alarme de fin de perfusion, alarme d'occlusion, alarme de non validation, alarme de fin de pause ; alarme et pré alarme volume limite, seringue vide ; alarme et pré alarme de batterie déchargée, alarme seringue mal installée [5].
- Fonctionnement de l'appareil de façon autonome.
- En urgence, risque d'erreurs de dosage ou de débit d'administration.
- Coût.

1.11 Conclusion

L'administration de médicaments par voie intraveineuse est une pratique courante en milieu hospitalier. Le pousse seringue électrique est un outil très utile et très largement utilisé pour cette application pratique. Que ce soit en Bloc opératoire ou en Urgence, Anesthésie, Réanimation, Cardiologie et Néonatalogie il est vraiment nécessaire pour une

Chapitre 1 : Généralité sur le Pousse seringue

injection commandée ou la dose injecter joue le rôle principal dans le traitement de différentes maladies.

Il existe plusieurs et différents types de pousse seringue avec leurs caractéristiques et mode d'utilisation qui sont appropriés. Mais comme tout outil, il faut en connaître le fonctionnement pour l'utiliser avec sécurité et efficacité.

Chapitre 2

La partie électronique et mécanique du pousse seringue

Chapitre 2 : La partie électronique et mécanique du pousse seringue

2.1 Introduction

Le pousse seringue est constitué de deux parties (électronique et mécanique) :

- La partie électronique représente le cerveau de pousse seringue, et le fonctionnement de cette dernière est basé sur cette partie électronique. Cette partie permet la gestion de l'ensemble des commandes et des instructions appliquées au système, cette partie électronique elle-même est basée principalement sur un microcontrôleur, la famille ATmega implanté dans une carte (Type Arduino) pour son bon fonctionnement et qui facilite sa connexion avec l'environnement extérieur.
- La partie mécanique représente les muscles du pousse seringue, qui est commandée par le microcontrôleur (le cerveau), avec une programmation appropriée, cette partie est composée principalement du moteur et d'autres modules pour le déplacement de la seringue. Elle peut réagir de la façon désirée si on choisit un bon moteur.

Les moteurs pas à pas ont plusieurs avantages par rapport aux autres types de moteurs (à courant continu), parce qu'on peut les faire tourner à des vitesses variables, et la position parcourue sera aussi facile à déterminer, donc ils sont plus précis.

2.2 Partie électronique : la carte Arduino.

2.2.1 Définition du module Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle), dont les plans de la carte elle-même sont publiés en licence libre, dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser, et produire des signaux électriques, de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications, comme l'électrotechnique industrielle et embarquée ; le modélisme ; la domotique, mais aussi dans des domaines différents, comme l'art contemporain, le pilotage d'un robot, commander des moteurs, faire des jeux

Chapitre 2 : La partie électronique et mécanique du pousse seringue

de lumières, communiquer avec l'ordinateur ou commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V, et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino [10].

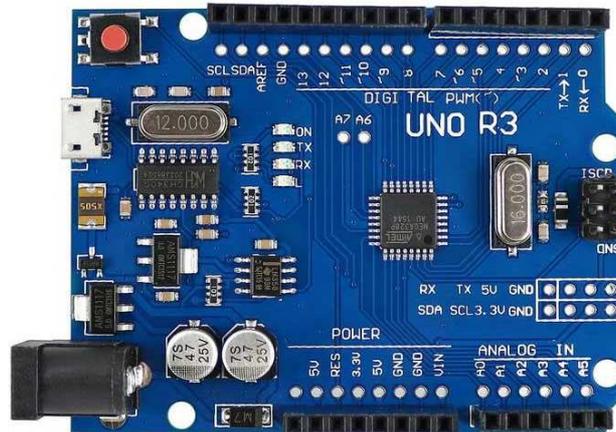


Figure 2.1 : Module Arduino.

2.2.2 Les gammes de la carte Arduino

Actuellement, il existe plus de 20 versions de module Arduino, on cite quelques-uns, afin d'éclaircir l'évolution de ce produit scientifique et académique :

- Le NG d'Arduino, avec une interface d'USB pour programmer et usage d'un ATmega8.
- L'extrémité d'Arduino, avec une interface d'USB pour programmer et usage d'un Microcontrôleur ATmega8.
- L'Arduino Mini, une version miniature de l'Arduino en utilisant un microcontrôleur ATmega168.
- L'Arduino Nano, une petite carte programme à l'aide porte USB cette version utilisant un microcontrôleur ATmega168 (ATmega328 pour une plus nouvelle version).
- Le LilyPad Arduino, une conception de minimaliste pour l'application wearable en utilisant un microcontrôleur ATmega168.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

- Le NG d'Arduino plus, avec une interface d'USB pour programmer et usage d'un ATmega168.
- L'Arduino Bluetooth, avec une interface de Bluetooth pour programmer en utilisant un microcontrôleur ATmega168.
- L'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 Mega, en utilisant un microcontrôleur ATmega1280 pour I/O additionnel et mémoire.
- L'Arduino UNO, utilisations microcontrôleur ATmega328.
- L'Arduino Mega2560, utilisations un microcontrôleur ATmega2560, et possède toute la mémoire à 256 KBS. Elle incorpore également le nouvel ATmega8U2 (ATmega16U2 dans le jeu de puces d'USB de révision 3).
- L'Arduino Leonardo, avec un morceau ATmega3U4 qui élimine le besoin de raccordement d'USB et peut être employé comme clavier.
- 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 [11].

Parmi ces types, on a choisi une carte Arduino UNO (carte Basique). L'intérêt principal de cette carte est de faciliter la mise en œuvre d'une telle commande qui sera détaillée par la suite.

L'Arduino fournit un environnement de développement s'appuyant sur des outils open source, comme interface de programmation. L'injection du programme, déjà converti par l'environnement sous forme d'un code « HEX », dans la mémoire du microcontrôleur, se fait d'une façon très simple par la liaison USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties. Cette carte est basée sur un microcontrôleur ATmega 328 et des composants complémentaires. La carte Arduino contient une mémoire morte de 1 kilo. Elle est dotée de 14 entrées/sorties

Chapitre 2 : La partie électronique et mécanique du pousse seringue

digitales (dont 6 peuvent être utilisées en tant que sortie PWM), 6 entrées analogiques et un cristal à 16 MHz, une connexion USB, et possède un bouton de remise à zéro, et une prise jack d'alimentation. La carte est illustrée dans la figure (2.1) [12].

2.2.3 Caractéristiques de La carte Arduino UNO

Microcontrôleur	ATmega328P
Tension de fonctionnement	5V
Tension d'entrée (recommandée)	7-12V
Tension d'entrée (limite)	6-20V
Broches d'E / S numériques	14 (dont 6 fournissent une sortie PWM)
Broches E / S numériques PWM	6
Broches d'entrée analogiques	6
Courant CC par broche I / O	20 mA
Courant DC pour Pin 3.3V	50 mA
Mémoire flash	32 Ko (ATmega328P) dont 0,5 Ko utilisé par le chargeur de démarrage
SRAM	2 Ko (ATmega328P)
EEPROM	1 Ko (ATmega328P)
Vitesse de l'horloge	16 MHz
LED_BUILTIN	13
Longueur	68,6 mm
Largeur	53,4 mm
Poids	25 g

Tableau 2.1 : Caractéristiques de La carte Arduino UNO.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

2.2.4 Pourquoi Arduino UNO ?

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

- **Le prix (réduits)** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plates-formes. La moins chère des versions du module Arduino peut être assemblée à la main, (les cartes Arduino pré assemblées coûtent moins de 2500 Dinars).
- **Multi plateforme** : le logiciel Arduino, écrit en JAVA, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **Un environnement de programmation clair et simple** : l'environnement de programmation Arduino (le logiciel Arduino IDE) est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **Logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le logiciel de programmation des modules Arduino est une application JAVA multi plateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le programme au travers de la liaison série (RS232, Bluetooth ou USB selon le module).
- **Matériel Open source et extensible** : les cartes Arduino sont basées sur les Microcontrôleur ATMEGA8, ATMEGA168, ATMEGA 328, les schémas des modules sont publiés sous une licence Créative Commons, et les concepteurs des circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant. Même les utilisateurs

Chapitre 2 : La partie électronique et mécanique du pousse seringue

relativement inexpérimentés peuvent fabriquer la version sur plaque d'essai de la carte Arduino, dont le but est de comprendre comment elle fonctionne pour économiser le coût [13].

2.2.5 La constitution de la carte Arduino UNO

Un module Arduino est généralement construit autour d'un microcontrôleur ATMEL AVR, et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmeur dédié ne soit pas nécessaire [12].

a Partie matérielle

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

✓ *Le Microcontrôleur ATmega328*

Un microcontrôleur ATmega328 est un circuit intégré, qui rassemble sur une puce plusieurs éléments complexes, dans un espace réduit au temps des pionniers de l'électronique. Aujourd'hui, en soudant un grand nombre de composants encombrants, tels que les transistors ; les résistances et les condensateurs, tout peut être logé dans un petit boîtier en plastique noir muni d'un certain nombre de broches dont la programmation peut être réalisée en langage C. la figure 2.2 montre un microcontrôleur ATmega 328, qu'on trouve sur la carte Arduino [12].

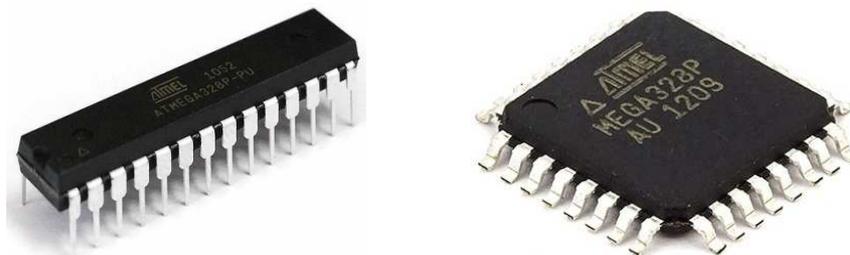


Figure 2.2 : Le microcontrôleur ATmega328.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

Le microcontrôleur ATmega328 est constitué d'un ensemble d'éléments qui ont chacun une fonction bien déterminée. Il est en fait constitué des mêmes éléments que sur la carte mère d'un ordinateur. Globalement, l'architecture interne de ce circuit programmable se compose essentiellement de :

- **Mémoire Flash** : C'est celle qui contiendra le programme à exécuter. Cette mémoire est effaçable et réinscriptible, mémoire programme de 32Ko (dont boot loader de 0.5ko).
- **RAM** : c'est la mémoire dite "vive", elle va contenir les variables du programme. Elle est dite "volatile" car elle s'efface si on coupe l'alimentation du microcontrôleur. Sa capacité est de 2 ko.
- **EEPROM** : C'est le disque dur du microcontrôleur. On y enregistre des infos qui ont besoin de survivre dans le temps, même si la carte doit être arrêtée.

Cette mémoire ne s'efface pas lorsque l'on éteint le microcontrôleur ou lorsqu'on le reprogramme [14].

✓ *Les sources de l'alimentation de la carte*

On peut distinguer plusieurs genres de sources d'alimentation (Entrée /Sortie) et cela comme suit :

- **VIN**. La tension d'entrée positive lorsque la carte Arduino est utilisée avec une source de tension externe (à distinguer du 5V de la connexion USB ou autre source 5V régulée). On peut alimenter la carte à l'aide de cette broche ou, si l'alimentation est fournie par le jack d'alimentation, accède à la tension d'alimentation sur cette broche.
- **5V**. La tension régulée utilisée pour faire fonctionner le microcontrôleur et les autres composants de la carte (pour info : les circuits électroniques numériques nécessitent une tension d'alimentation parfaitement stable dite "tension régulée" obtenue à l'aide d'un composant appelé un régulateur et qui est intégré à la carte Arduino). Le 5V régulé fourni par cette broche, peut donc provenir soit de la tension d'alimentation VIN, via

Chapitre 2 : La partie électronique et mécanique du pousse seringue

le régulateur de la carte, ou bien de la connexion USB (qui fournit du 5V régulé), ou de toute autre source d'alimentation régulée.

- **3.3V.** Une alimentation de 3.3V fournie par le circuit intégré FTDI (circuit intégré faisant l'adaptation du signal entre le port USB de votre ordinateur et le port série de l'ATMega) de la carte est disponible : ceci est intéressant pour certains circuits externes nécessitant cette tension au lieu du 5V, l'intensité maximale disponible sur cette broche est de 50mA [11].

✓ *Les entrées & sorties*

Cette carte possède 14 broches numériques (numérotée de 0 à 13), peut être utilisée soit comme une entrée numérique, soit comme une sortie numérique, en utilisant les instructions : pin Mode (), digital Write () et digital Read () du langage Arduino. Ces broches fonctionnent en 5V. Chaque broche peut fournir ou recevoir un maximum de 40mA d'intensité, et dispose d'une résistance interne de "rappel au plus" (pull-up) (déconnectée par défaut) de 20-50KOhms. Cette résistance interne s'active sur une broche en entrée à l'aide de l'instruction digital Write (broche, HIGH). En plus, certaines broches ont des fonctions spécialisées :

- **Interruptions Externes** : Broches 2 et 3. Ces broches peuvent être configurées pour déclencher une interruption sur une valeur basse, sur un front montant ou descendant, ou sur un changement de valeur.
- **Impulsion PWM** (largeur d'impulsion modulée) : Broches 3, 5, 6, 9, 10, et 11, fournissent une impulsion PWM 8-bits à l'aide de l'instruction analog Write ().
- **SPI** (Interface Série Périphérique) : Broches 10 (SS), 11 (MOSI), 12 (MISO), 13(SCK). Ces broches supportent la communication SPI (Interface Série Périphérique) disponible avec la librairie, pour communication SPI, les broches SPI sont également connectées sur le connecteur ICSP, qui est mécaniquement compatible avec les cartes Méga.
- **I2C**: Broches 4 (SDA) et 5 (SCL), supportent les communications de protocole I2C ou interface TWI (TwoWire Interface - Interface "2 fils"),

Chapitre 2 : La partie électronique et mécanique du pousse seringue

disponible en utilisant la librairie Wire/I2C (ou TWI - Two-Wire interface - interface "2 fils").

- **LED** : Broche 13. Il y a une LED incluse dans la carte connectée à la broche 13. Lorsque la broche est au niveau HAUT, la LED est allumée, lorsque la broche est au niveau BAS, la LED est éteinte.

La carte UNO dispose de 6 entrées analogiques (numérotées de 0 à 5), chacune pouvant fournir une mesure d'une résolution de 10 bits (c à d sur 1024 niveaux soit de 0 à 1023) à l'aide de la très utile fonction analog Read () du langage Arduino.

Par défaut, ces broches mesurent entre le 0V (valeur 0) et le 5V (valeur 1023), mais il est possible de modifier la référence supérieure de la plage de mesure en utilisant la broche AREF et l'instruction analog Référence () du langage Arduino.

La carte Arduino UNO intègre un fusible qui protège le port USB de l'ordinateur contre les surcharges en intensité (le port USB est généralement limité à 500mA en intensité).

Bien que la plupart des ordinateurs aient leur propre protection interne, le fusible de la carte fournit une couche supplémentaire de protection. Si plus de 500mA sont appliqués au port USB, le fusible de la carte coupera automatiquement la connexion jusqu'à ce que le court-circuit ou la surcharge soit stoppé [10].

✓ *Les ports de communications*

La carte Arduino UNO a de nombreuses possibilités de communications avec l'extérieur. L'Atmega328 possède une communication série UART TTL (5V), grâce aux broches numériques 0 (RX) et 1 (TX). On utilise (RX) pour recevoir et (TX) transmettre (les données séries de niveau TTL).

Ces broches sont connectées aux broches correspondantes du circuit intégré ATmega328 programmé en convertisseur USB – vers – série de la carte, composant qui assure l'interface entre les niveaux TTL et le port USB de l'ordinateur (Figure 2.3).

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

Chapitre 2 : La partie électronique et mécanique du pousse seringue

est le câble USB, pour éviter cela, il existe différentes méthodes pour utiliser ce dernier sans fil.

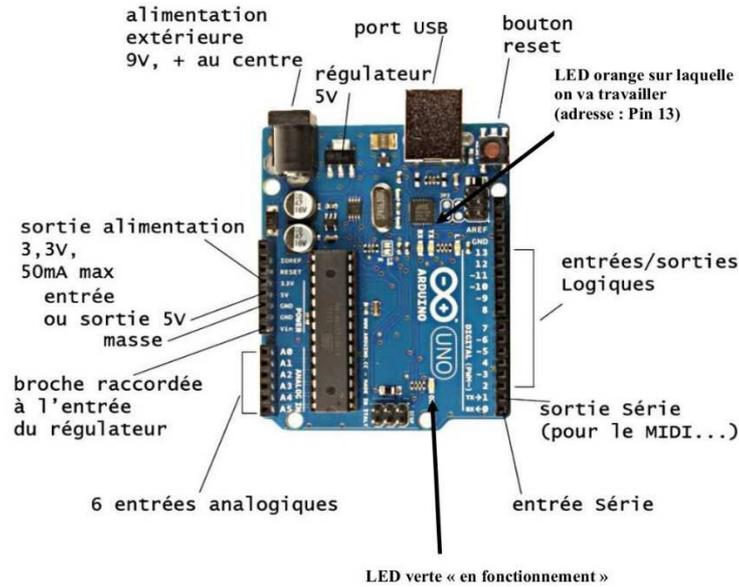


Figure 2.3 : Les ports de communications Arduino.

b Partie programme

Une telle carte d'acquisition qui se base sur sa construction sur un microcontrôleur, doit être dotée d'une interface de programmation comme est le cas de notre carte.

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

✓ ***L'environnement de la programmation***

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino.

✓ ***Structure générale du programme (IDE Arduino)***

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C (Figure 2.4) [15].

Chapitre 2 : La partie électronique et mécanique du pousse seringue

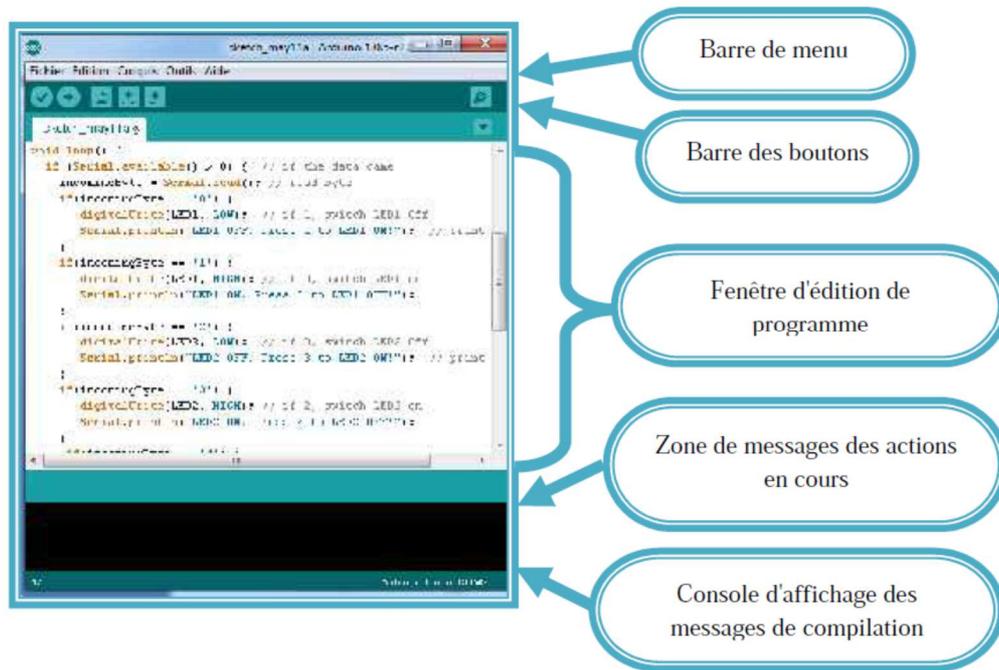


Figure 2.4 : l'interface de IDE Arduino.

2.3 Partie mécanique : Les moteurs pas à pas.

C'est un moteur spécial pour contrôler la précision de l'angle. L'angle entre les deux degrés de mouvement est appelé pas, plus le pas est petit plus le moteur est précis. Les moteurs pas à pas (Figure 2.5) sont couramment utilisés dans les asservissements de vitesse et de position. Le stator se consiste de plusieurs bobines alimentées par des impulsions de courant continu. Le rotor est constitué d'un ou plusieurs aimants permanents qui vont avoir tendance à s'aligner sur les directions successives de l'induction obtenue par les alimentations successives des bobines [16].

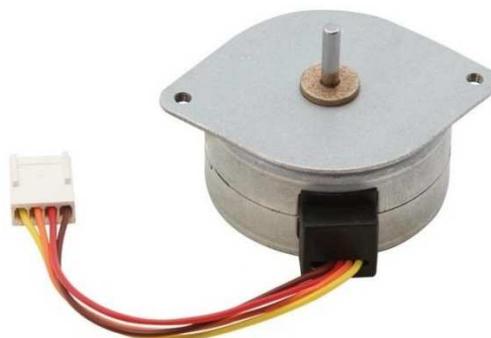


Figure 2.5 : Moteur pas à pas.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

2.3.1 Utilisation des moteurs pas à pas

Les moteurs pas à pas, utilisés pratiquement dans tous les composants d'un système informatique, sont très mal connus, surtout en ce qui concerne leur mode de commande. Ils sont utilisés aussi bien dans les lecteurs de disquettes, les disques durs, que dans les scanners et les imprimantes 3D, conçues pour le prototypage rapide. Dans le domaine de la robotique, ils sont la clef de voûte du système. Ils permettent d'obtenir une précision extraordinaire grâce à leur conception électrique et mécanique. Dans le domaine médical : pousse seringue (le moteur pas à pas permet un débit régulier pour la perfusion).

2.3.2 Avantage et inconvénients

➤ **Les avantages :**

- La vitesse couple et la résolution est caractérisée.
- Pas d'usure donc durée de vie importante.
- Facilité d'intégration dans un système.

➤ **Les inconvénients :**

- Bruyant et grande vibration.
- Vitesse faible si on augmente le nombre de pas par le drive [17].

2.3.3 Les différents types des moteurs pas à pas

Il y a trois types de moteurs pas à pas : Les moteurs à aimant permanent ; Les moteurs à réluctance variable et Les moteurs Hybrides.

✓ **Moteur à aimant permanent**

Un moteur à aimant permanent comprend : Un rotor bipolaire constitué d'un aimant permanent (partie mobile) et un stator à deux paires de pôles (partie fixe) (Figure 2.6) [18].

Chapitre 2 : La partie électronique et mécanique du pousse seringue

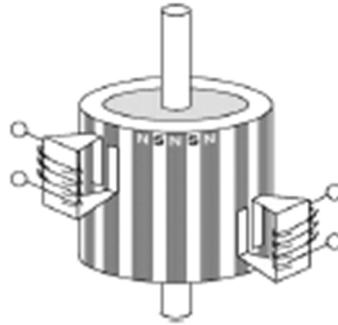


Figure 2.6 : Structure interne d'un moteur à aimant permanent.

Les bobines diamétralement opposées constituent les phases. Elles sont connectées de façon à créer un pôle sud et un pôle nord. En inversant les sens des courants dans une phase, on permute les pôles engendrés par une bobine. Le rotor se déplace alors et prend une nouvelle position d'équilibre stable. Selon la conception des enroulements, on distingue deux grands types de moteurs pas à pas (les moteurs unipolaires et les moteurs bipolaires).

✓ Moteur unipolaire

Le rotor est constitué par un aimant permanent (en ferrite par exemple) comportant une paire de pôles. Le stator, comprend deux circuits magnétiques décalés de 90°. Les enroulements à point milieu sont alimentés avec une polarité toujours du même signe, (Figure 2.7) d'où le nom d'unipolaire [19].

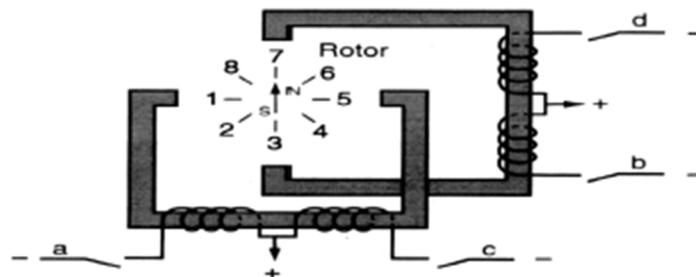


Figure 2.7 : Moteur unipolaire à aimant permanent.

Chaque position du moteur possède une séquence de communication, le tableau 2.2 résume le mode d'alimentation successive qui représente chacun un demi roulement.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

Mode 1 : on alimente successivement chaque demi-enroulement					
Séquence de commutation	a	b	c	d	Position du rotor
1	1	0	0	0	1
2	0	1	0	0	3
3	0	0	1	0	5
4	0	0	0	1	7

Tableau 2.2 : Mode d'alimentation du moteur unipolaire.

✓ Moteur bipolaire

Les enroulements au stator n'ont pas de point de milieu. Chaque borne de chaque enroulement est alimentée successivement par une polarité positive puis négative (d'où le terme bipolaire) (Figure 2.8). En inversant les polarités des enroulements statoriques, on inverse les pôles nord et sud au stator. Le nombre de phase est égal au nombre d'enroulements.

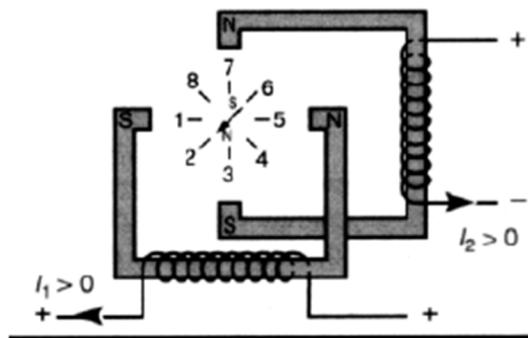


Figure 2.8 : Moteur bipolaire à aimant permanent.

Chapitre 2 : La partie électronique et mécanique du pousse seringue

Le tableau 2.3 présente le mode d'alimentation d'un moteur bipolaire :

Mode 1 : on alimente un enroulement					
Séquence de commutation	I1>0	I1<0	I2>0	I2<0	Position du rotor
1	1	0	0	0	1
2	0	0	1	0	3
3	0	1	0	0	5
4	0	0	0	1	7

Tableau 2.3 : Mode d'alimentation du moteur bipolaire.

Le principe de fonctionnement est identique à celui du moteur unipolaire. Le sens de rotation dépend au sens du courant et de l'ordre d'alimentation des bobinages.

✓ Moteur à réluctance variable

Ce moteur comporte une denture dont le pas n'est pas le même au stator et au rotor; le rotor n'est pas aimanté. Exemple : Stator 8 pôles et rotor 6 pôles ; Pas statorique: $\alpha_s = 360^\circ / 8 = 45^\circ$; Pas rotorique : $\alpha_r = 360^\circ / 6 = 60^\circ$ (Figure 2.9).

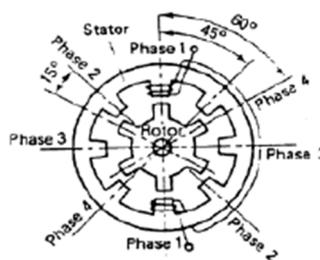


Figure 2.9: Moteur à réluctance variable.

✓ Moteur hybride

Ce moteur superpose le principe de fonctionnement des moteurs à aimant permanent et à réluctance variable et combine leurs avantages. Le rotor est constitué de 2 disques dentés décalés mécaniquement, afin d'obtenir une bonne résolution avec un couple

Chapitre 2 : La partie électronique et mécanique du pousse seringue

moteur assez élevé (Figure 2.10). Entre ces 2 disques est inséré un aimant permanent. La structure du stator reste la même [20].

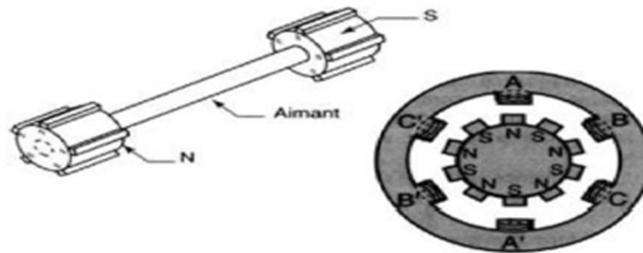


Figure 2.10 : Moteur hybride.

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 maximal. [20].

2.4 Conclusion

Dans ce chapitre, on a projeté la lumière sur la carte Arduino et les moteurs pas à pas expliquant aussi leurs principes de fonctionnement et on a cité leurs différents types.

Parmi les différents types des cartes Arduino on a choisi la carte Arduino de type UNO parce qu'elle est moins chère, et elle offre un nombre d'entrées/sorties suffisant, un processeur puissant doté d'une mémoire vaste, et donc permet d'exploiter des algorithmes plus complexes.

On a choisi le moteur pas à pas à aimant permanent à cause de ses dimensions réduites et son bon marché et en plus il est plus précis.

Ces deux composants sont essentiels dans la réalisation de notre prototype, et on va donner plus de détails sur leurs branchements et utilisations dans le chapitre suivant.

Chapitre 3

Réalisation et étude du Prototype

3.1 Introduction

Le but de notre travail est de réaliser un système qui répond à notre étude théorique, donc un mécanisme qui permet la poussé d'une seringue, en le commandant pour qu'il puisse injecter des doses de médicament bien précis dans un intervalle de temps bien déterminer.

Ainsi notre étude sera menée sur deux parties. Une, sur l'électronique du système, donc l'Arduino qui commande la seringue, et l'autre partie sur le mécanisme qui fait bouger la seringue.

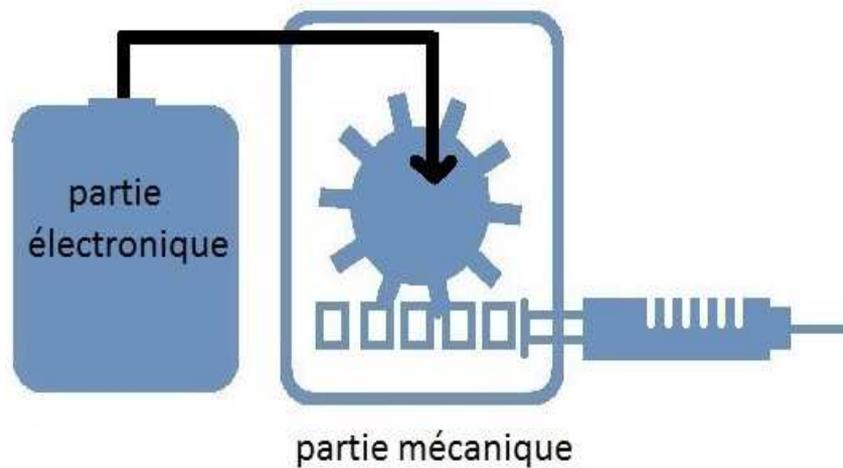


Figure 3.1 : Schéma simplifié de système pousse seringue.

Comme on a cité auparavant, notre prototype est constitué de deux parties mécanique et électronique (Hardware et software) (Figure 3.1). Le schéma bloc générale de notre système est présenté dans la figure 3.2 :

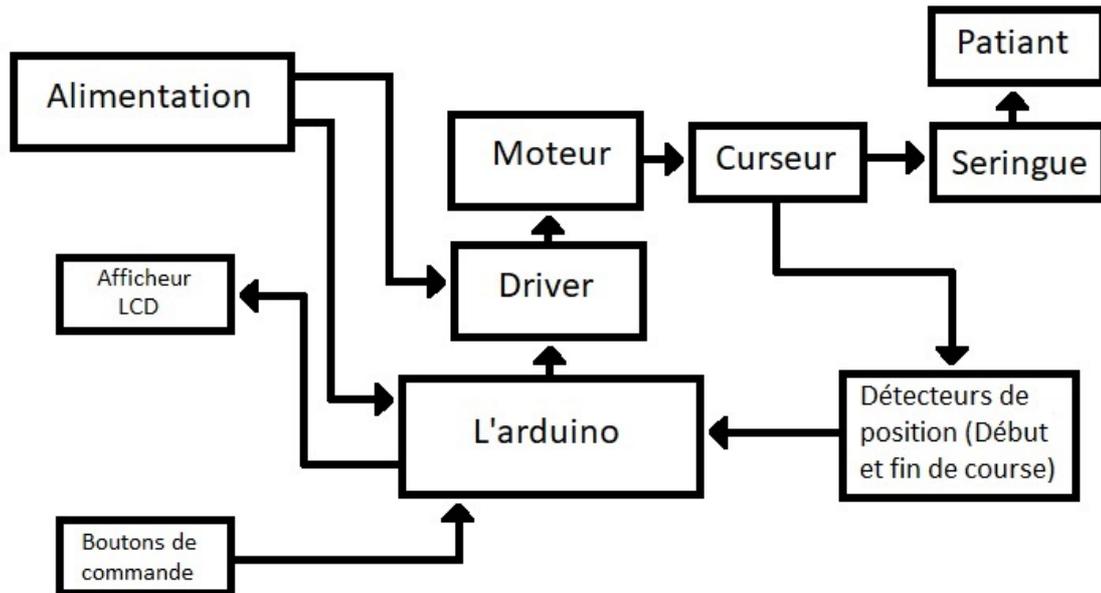


Figure 3.2 : Schéma synoptique du prototype.

3.2 Partie Hardware

3.2.1 Circuit d'alimentation stabilisé

Dans notre travail on a eu besoin d'une source continue de 12 et 5 volts. Pour nous, le disponible était une source de tension de 220 volts en alternatif, ce qui nous a posé un problème.

Pour passer du courant alternatif vers le courant continu, on a décidé de réaliser une carte d'alimentation stabilisée (Figure 3.3), délivrant en sortie des tensions stabilisées 5 volts pour l'Arduino , le LCD et le circuit L293D et 12 volts pour le moteur pas à pas, et comme l'alimentation d'origine est généralement le réseau alternatif fournissant 220 volts, alors on est obligé d'avoir recours à un transformateur abaisseur de tension, puis un redressement double alternance assuré par le pont diode, afin de convertir la tension alternative issue du secondaire du transformateur en une tension unidirectionnelle, ensuite filtrer cette tension par un condensateur, et puis quand on a eu besoin des tensions continues 12 et 5 volts, on a utilisé deux régulateurs.

Chapitre 3 : Réalisation et étude du Prototype

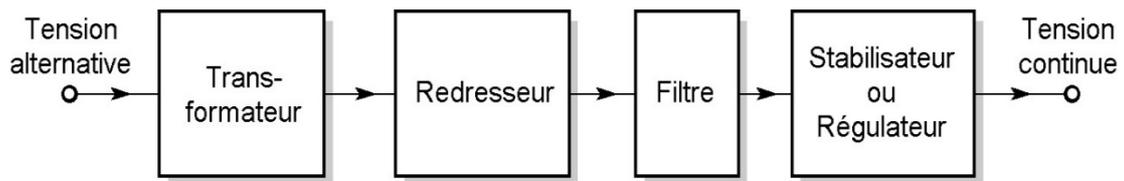


Figure 3.3 : Synoptique de l'alimentation stabilisée.

➤ Le transformateur d'alimentation

Le transformateur électrique est un élément électrique statique, permettant de modifier les valeurs de tension et d'intensité du courant, délivré par une source d'énergie électrique alternative, en un système de tension, et de courant de valeurs différentes, mais de même fréquence et de même forme.

Un transformateur (Figure 3.4) comprend deux bobines différentes couplées par un cadre magnétique fermé. Il comporte un circuit primaire et un circuit secondaire. Le circuit primaire correspond à la bobine alimentée par la tension que l'on veut modifier

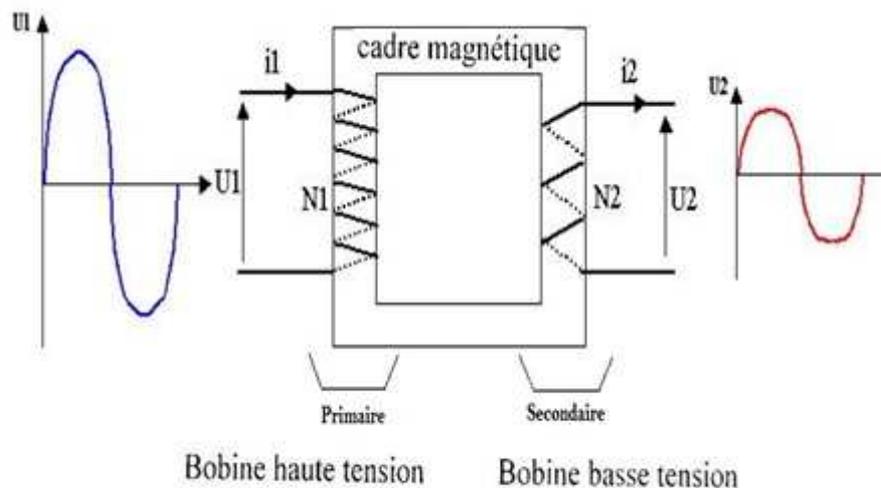


Figure 3.4 : Transformateur monophasé.

Le cadre magnétique sert à canaliser et à transmettre le champ magnétique créé par la première bobine à celle du secondaire. Il apparaît alors dans la seconde bobine un courant électrique induit.

➤ Le Redresseur

Les montages redresseurs (Figure 3.5), souvent appelés simplement redresseurs, sont les convertisseurs de l'électronique de puissance qui assurent directement la conversion

Chapitre 3 : Réalisation et étude du Prototype

alternatif-continu. Alimentés par une source de tension alternative monophasée, ils permettent d'alimenter en courant continu le récepteur branché à leur sortie. Les diodes sont utilisées pour un redresseur non commandé, soit une seule (redresseur mono alternance) ou bien un pont (redresseur double alternance) ce qu'est notre cas.

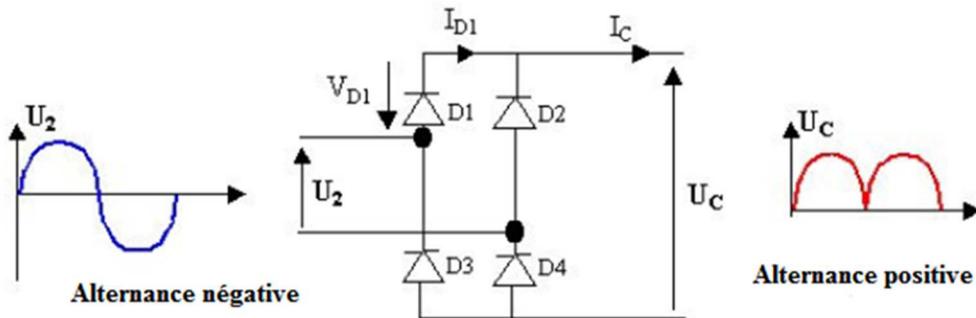


Figure 3.5 : Représentation d'un redresseur à double alternance.

➤ Filtrage

C'est un circuit électrique qui se compose d'un condensateur et une résistance mettent en parallèle avec redresseur (Figure 3.6). Après les étapes précédentes on avait une tension qui garde toujours la même polarité, mais elle varie beaucoup. L'ajout d'un condensateur en parallèle avec notre résistance de charge permettra d'atténuer cette oscillation. Le condensateur se charge lorsque la tension d'entrée est grande et, lorsqu'elle est faible, il se décharge lentement.

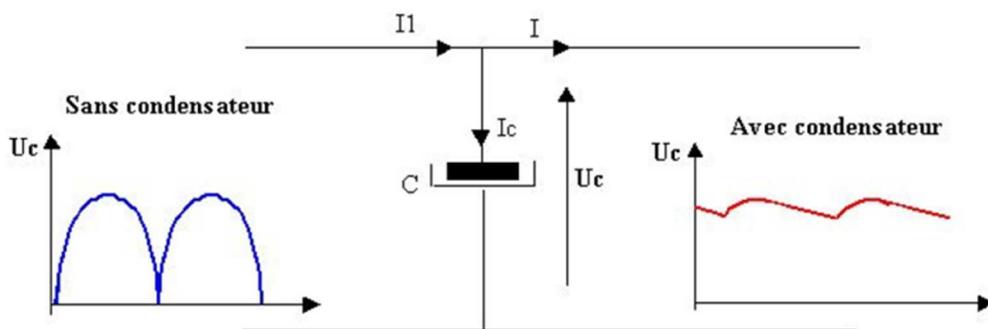


Figure 3.6 : Le circuit et le signal de filtrage.

➤ Régulateur

A la fin, on a utilisé un régulateur de tension (Figure 3.7), élément qui permet de stabiliser une tension à une valeur fixe, et qui est nécessaire pour notre montage

Chapitre 3 : Réalisation et étude du Prototype

électronique car on a besoin d'une tension qui ne fluctue pas. Pour l'appliquer dans les différents blocs qu'on a, on a besoin d'une tension de 5 Volts qui alimente l'Arduino, et une tension de 12Volts qui alimente le moteur pas à pas, pour notre projet, on a utilisé deux régulateurs 12 Volts et 5 Volts à savoir le 7812 et 7805.

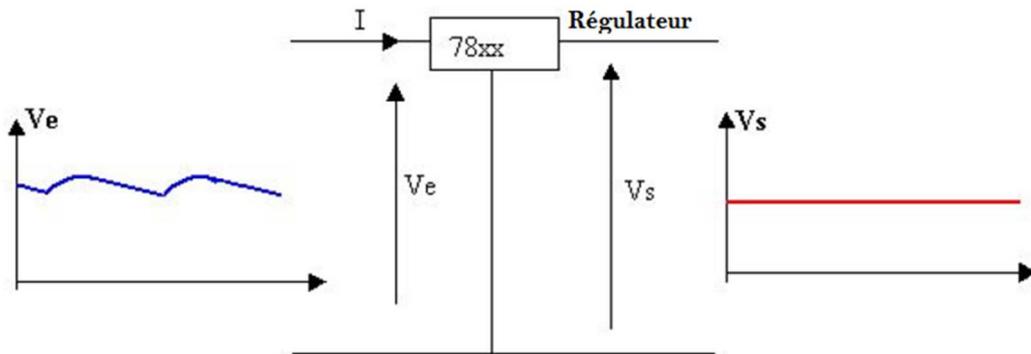


Figure 3.7 : Circuit de branchement des régulateurs.

3.2.2 Schéma électrique de l'alimentation isolée sur ISIS

Ainsi notre circuit électrique de l'alimentation stabilisée et isolée sera simulé par ISIS (Proteus) et réalisé par la suite (Figure 3.8).

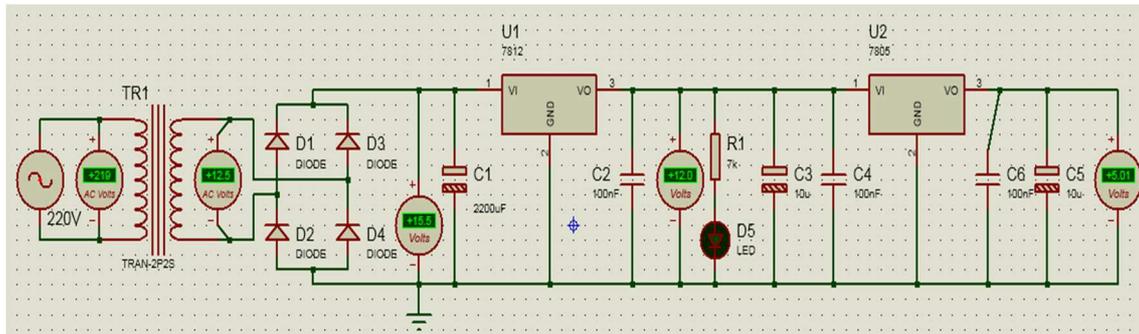


Figure 3.8 : Schéma électrique de l'alimentation stabilisée linéaire.

La tension du secteur est abaissée par le transformateur, puis redressée dans le pont de diodes formé par quatre diodes avant d'être lissée par le filtre, ou le rôle du filtre est de stocker l'énergie pendant la charge et de redistribuer cette énergie pendant la décharge.

Chapitre 3 : Réalisation et étude du Prototype

Pour absorber les imperfections de la tension produite, on a placé un élément qui se comporte comme une résistance variable (transistor de puissance ballast). Ce régulateur de tension permet une réjection de l'ondulation résiduelle et une protection contre les courts circuits par une limitation de l'intensité en sortie. Cette alimentation réalisée est connue sous le nom d'alimentation linéaire, à ballast, ou à régulation série. Les principales caractéristiques de ce genre d'alimentation sont :

Un rendement faible, un temps de maintien en cas de coupure de la tension du secteur de l'ordre de la milliseconde, et une tension résiduelle de l'ordre de quelques mV, et une génération de perturbations électromagnétiques quasi-nulle.

3.2.3 La carte Arduino UNO

Cette unité de contrôle est le cœur de notre projet. Elle permet d'acquérir des données ou envoyer des commandes au reste de circuit. Dans le chapitre précédant on a mentionné tous ce qui concerne son fonctionnement. Dans cette partie, on va montrer son brochage et les pins réservé pour chaque composant utilisé. La figure 3.9 illustre le type de brochage utilisé dans notre projet.

Les pins de port (8, 9, 10,11) sont reliés au driver L293D, le circuit de commande de moteur pas à pas.

Les pins de port (A4, A5) représentent le bus de données reliant le microcontrôleur à l'afficheur LCD.

Les pins de port (2, 3, 4, 5, 7, A0) représentent le bus de données reliant le microcontrôleur aux boutons de commande.

Enfin on a les pins de port (A2, A3, 12, 13), qui sont utilisées pour programmer le microcontrôleur, afin d'avoir les fonctions qui sont respectivement : deux capteurs de fin de course, LED de fonction ON, LED fonction injection.

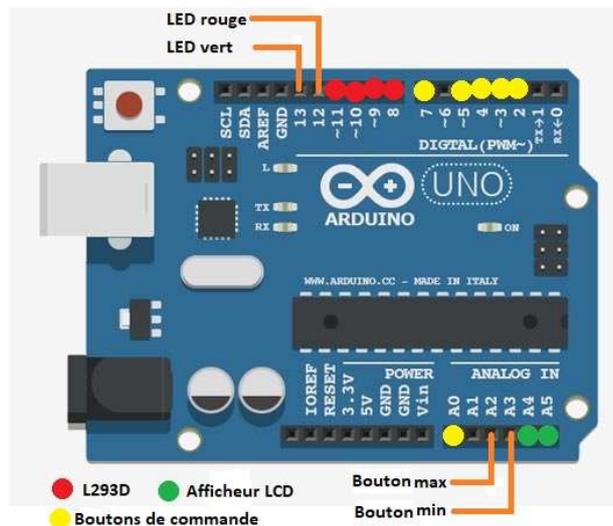


Figure 3.9 : Brochage d'Arduino UNO.

3.2.4 Moteur pas à pas

Il existe différents types de moteurs, alors pourquoi le moteur pas à pas dans notre projet ? La réponse est simple : c'est le seul qui répond à nos besoins, vu qu'on peut commander son déplacement avec précision. Notre moteur on l'a récupéré d'une vieille imprimante.

➤ Le moteur pas à pas bipolaire

Les moteurs pas à pas bipolaires existent en différentes tailles qui varient entre 1 cm et plus d'une dizaine de centimètres (notre moteur est de 42mm).

Signalons que le couple est exprimé en kilogramme par centimètre (kg.cm^{-1}), ce qui définit le poids en kilogrammes que pourra soulever l'axe d'un moteur pourvu d'un bras de longueur exprimée en centimètre, et pour notre cas on peut dire que notre moteur est assez puissant.

La valeur de leur tension d'alimentation varie dans de grandes proportions, elle est comprise entre 6V et plusieurs dizaines de volts (24V). De même, selon la résistance ohmique de leurs bobinages, le courant consommé s'étendra dans une gamme allant de quelques dizaines de milliampères à plusieurs ampères. En simplifiant, on peut dire que plus le courant sera élevé, plus le couple sera important.

Chapitre 3 : Réalisation et étude du Prototype

Ce moteur est assez simple à piloter. En effet, il suffit d'alimenter une à une les bobines pour que le moteur tourne. Chaque bobinage est commandé séparément et ne requiert qu'une alimentation présente ou absente selon que l'on veuille ou non créer un champ magnétique en son sein. Le rôle de moteur dans notre prototype est de faire tourner le pignon ce qui va faire tourner la tige et tirer le poussoir afin de bouger la seringue comme est indiqué dans la figure 3.10 :

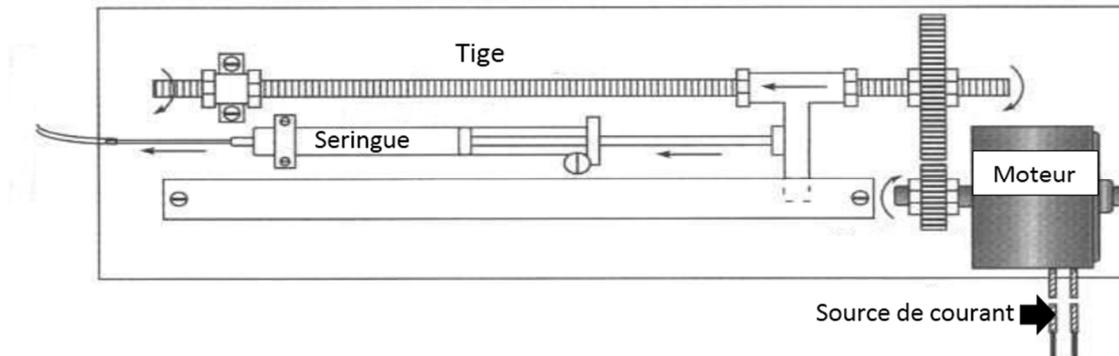


Figure 3.10 : Le fonctionnement du moteur dans le prototype.

➤ La commande de moteur par à pas

Ce composant est souvent utilisé pour piloter le moteur pas à pas à courant continu à partir d'un microcontrôleur. Le L293D est une vraie interface pour moteur avec un circuit de puissance et son alimentation séparée, pour les étages de sortie, et pour un circuit de commande en logique TTL sous 5 Volts. Dans le même esprit que des amplificateurs de puissance. À la base, un circuit pont-H est conçu pour pouvoir dissocier la logique de commande (basse tension) du circuit de puissance.

Un pont-H dispose donc de deux sources d'alimentations :

- ✓ Une source d'alimentation pour la logique du circuit. Elle est nommée **VSS** et nécessite généralement 5 Volts.
- ✓ Une source d'alimentation pour les moteurs. Elle est nommée **VS** et tolère des tensions nettement plus élevées que VSS (logique de commande).

La masse est commune aux deux circuits. Lorsque deux sources de tension sont utilisées il faut raccorder ensemble la masse (GND) d'Arduino et celle de l'alimentation

Chapitre 3 : Réalisation et étude du Prototype

externe. Les masses doivent être communes entre les alimentations (d'Arduino et du Pont-H).

Cette double source d'alimentation permet, par exemple, de commander des moteurs 12 Volts ($V_S=12v$) à partir d'un Arduino (Logique de commande= $V_{SS}=+5v$ Arduino).

Cela ne présente pas seulement un avantage pour des tensions différentes, cela permet aussi de décharger le régulateur de tension du microcontrôleur de l'alimentation des moteurs.

➤ Brochage du L293D

La figure 3.11 représente le schéma de brochage du circuit L293D. Le tableau 3.1 présente une description de chaque broche.

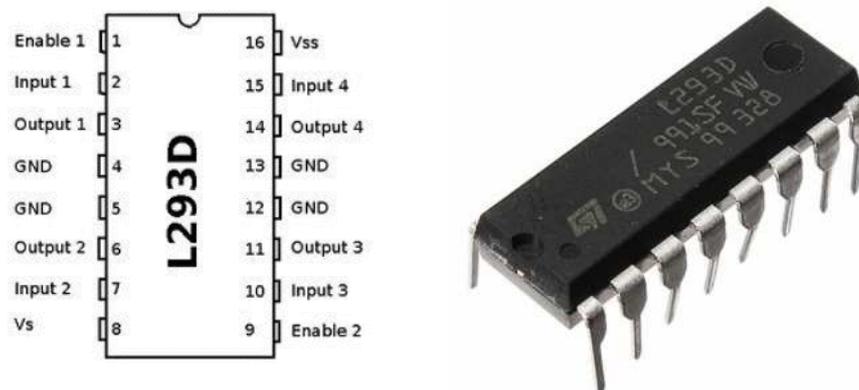


Figure 3.11 : Le circuit intégré L293D.

Broche	Nom	Description
1, 9	Enable1, 2	L'activation/désactivation du premier et second pont H.
2, 7, 10, 15	Input 1, 2, 3, 4	La commande des sorties des Ponts-H.
3, 6, 11, 14	Output1, 2, 3, 4	Raccorder aux charges (moteur).
4, 5, 12, 13	GND	Doit être raccordée à la masse (GND).
8	VS	Alimentation de puissance de moteur.
16	VSS	Alimentation de la logique de commande (5V).

Tableau 3.1 : Brochage du L293D.

Chapitre 3 : Réalisation et étude du Prototype

3.2.5 Ecran LCD

Pour l'affichage il existe plusieurs technologies d'afficheurs, dont les afficheurs LCD ou OLED. Dans ce projet, on a utilisé un afficheur LCD 20x4 (20 colonnes, 4 lignes) (Figure 3.12) car c'est la solution la moins couteuse.

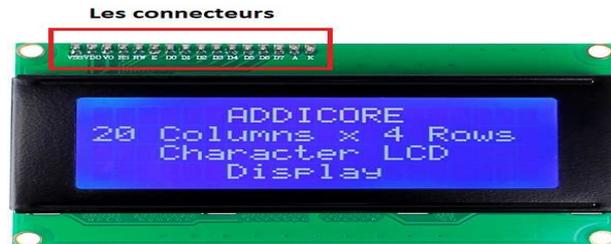


Figure 3.12 : Ecran LCD 20X4.

Un écran LCD possède une grille de carré. Ces carrés sont appelés des pixels (de l'anglais Picture Élément), Chaque pixel est un cristal liquide. Lorsqu'aucun courant ne le traverse, ses molécules sont orientées dans un sens (admettons 0°). En revanche lorsqu'un courant le traverse, ses molécules vont se tourner dans la même direction (90°). Pour pouvoir afficher des caractères sur l'écran il nous faudrait activer individuellement chaque pixel de l'écran (Tableau 3.2).

N°	Nom	Rôle
1	VSS	Masse
2	Vdd	+5V
3	VO	Réglage du contraste
4	RS	Sélection du registre (commande ou donnée)
5	R/W	Lecture ou écriture
6	7E	Entrée de validation
7 à 14	D0 à D7	Bits de données
15	A	Anode du rétroéclairage (+5V)
16	K	Cathode du rétroéclairage (masse)

Tableau 3.2 : Pins du LCD.

➤ Le brochage du LCD

La communication parallèle prend beaucoup de broches c'est pour sa on a utilisé un module d'interface I2C (Figure 3.13).



Figure 3.13 : Module d'interface I2C.

Ce contrôleur permet de contrôler facilement un écran LCD avec uniquement deux broches Arduino (une pour transporter les données SDA, une autre pour l'horloge de synchronisation -SCL).

C'est un outil très appréciable étant donné le nombre de broches très limité sur un Arduino UNO (Figure 3.14).

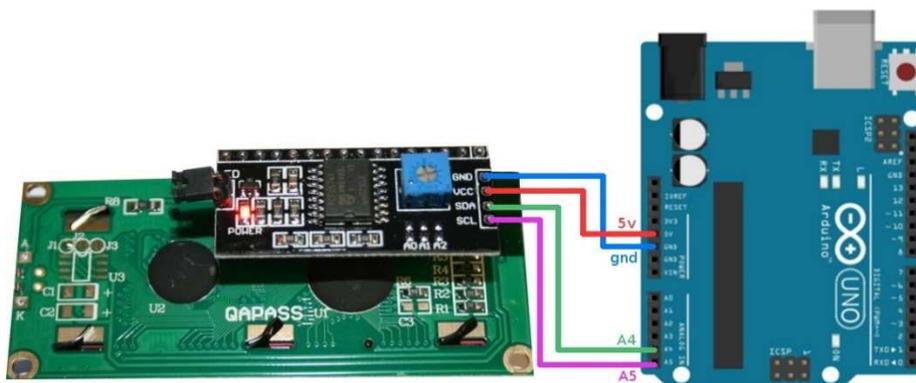


Figure 3.14 : Brochage de LCD par le module d'interface I2C.

3.2.6 Le clavier

C'est un outil qui nous permet de contrôler le prototype (saisir les débits, les volumes la vitesse du moteur...etc.)

Au marché on en trouve plusieurs types comme les claviers digitaux, clavier analogique, claviers à bouton ...etc., mais dans notre cas on a choisi de fabriquer notre clavier à l'aide des boutons poussoir et des résistances pour éliminer le courant de fuit (Figure 3.15).

Chapitre 3 : Réalisation et étude du Prototype

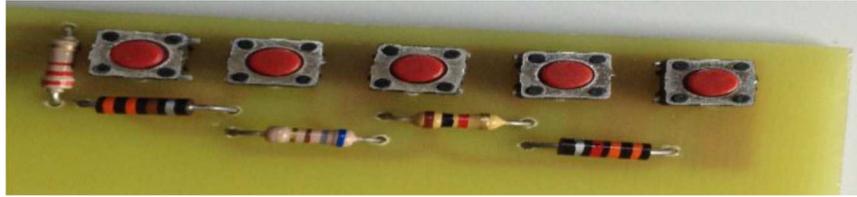


Figure 3.15 : Boutons poussoir relié à des résistances.

La figure 3.16 résume le schéma électrique de notre prototype sur le simulateur ISIS:

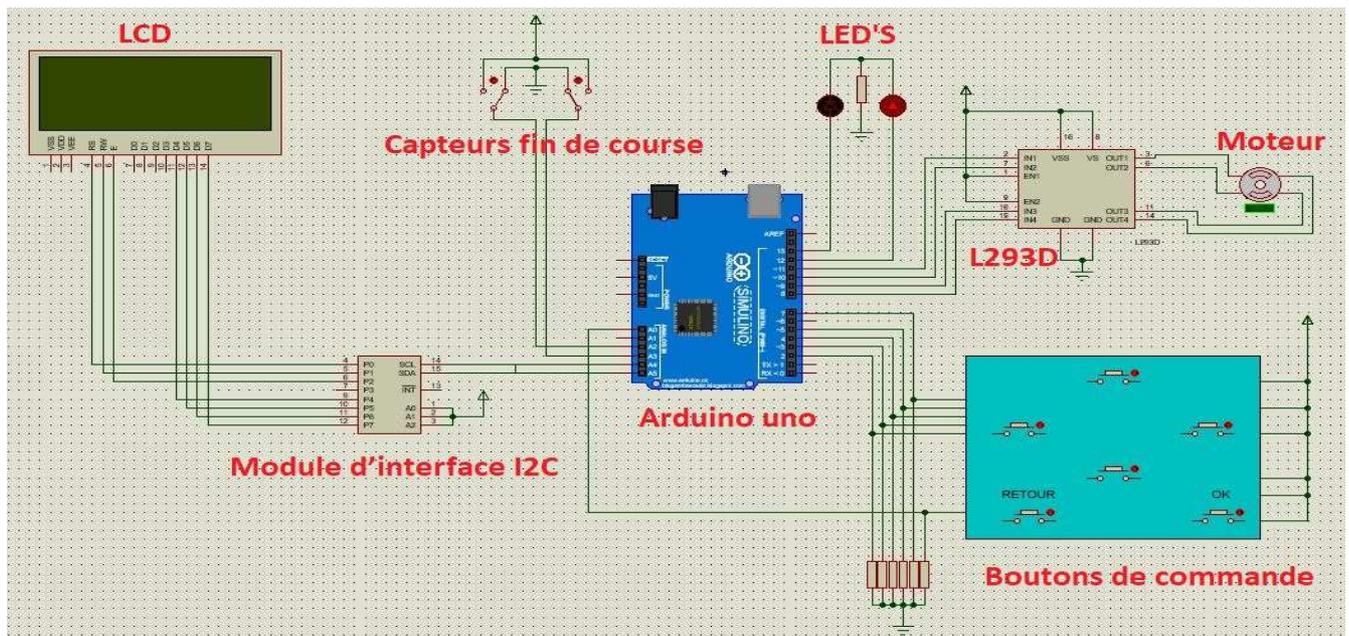
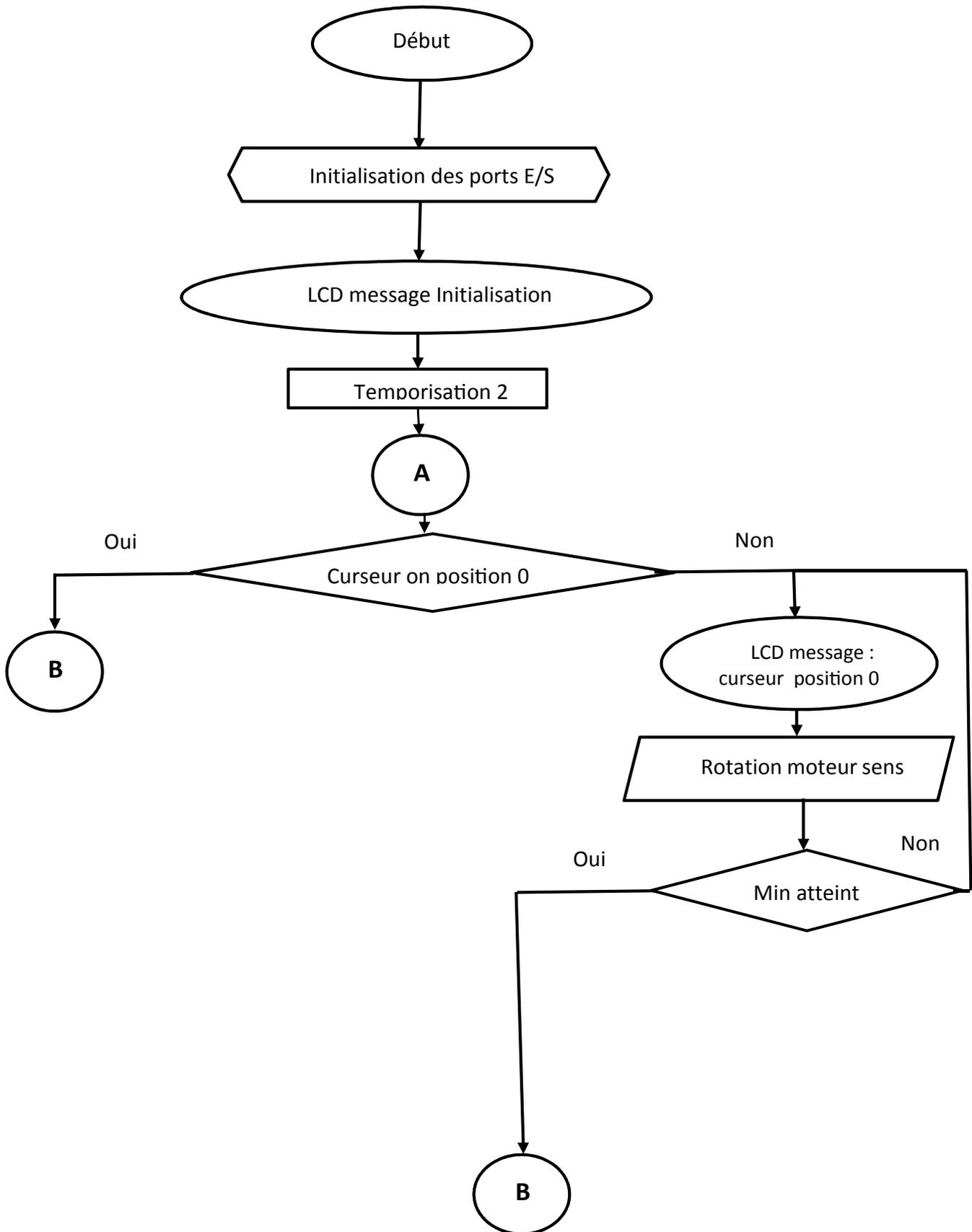


Figure 3.16 : Schéma électrique de notre circuit.

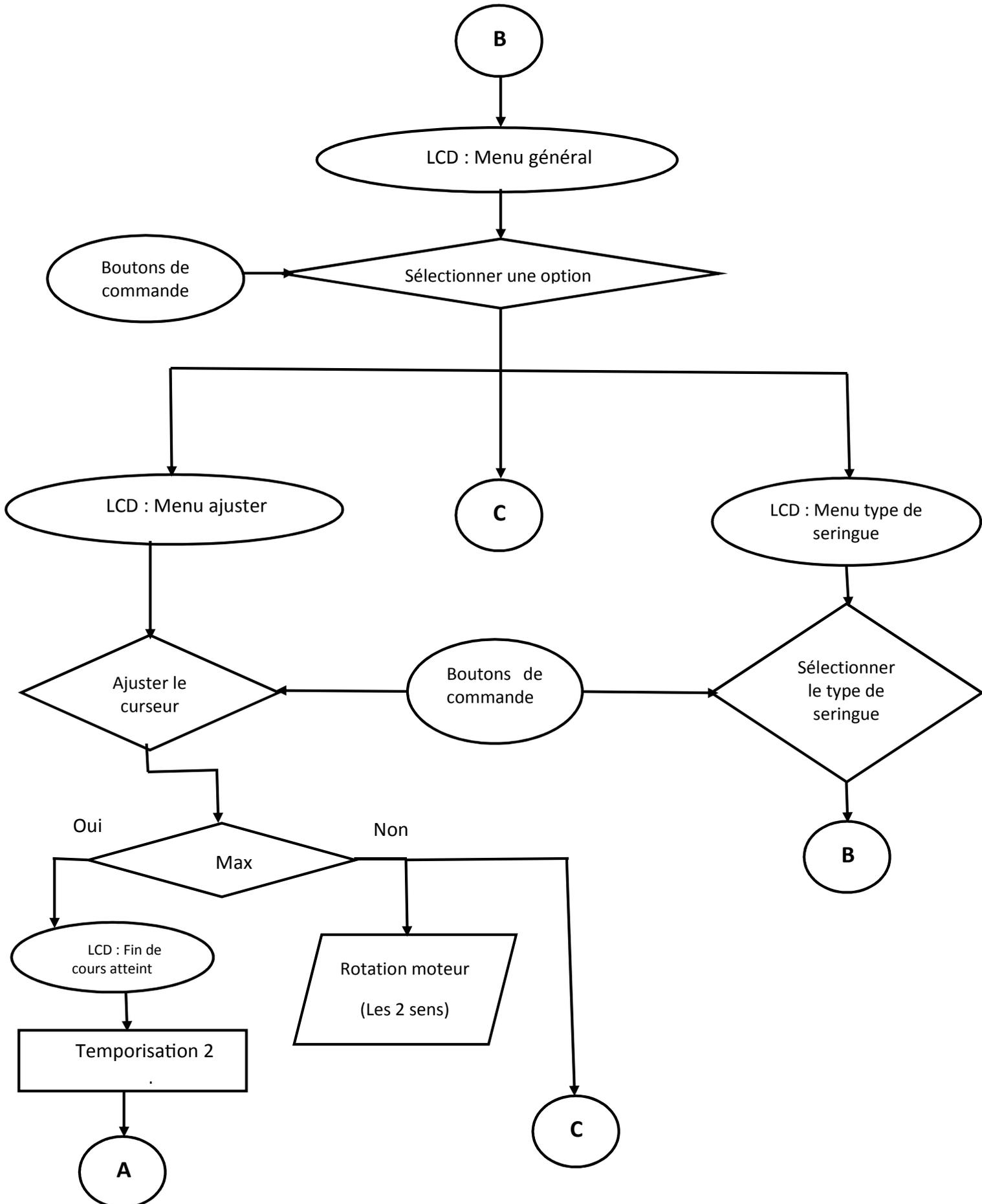
3.3 Partie Software

Dans cette partie de ce chapitre, on va évoquer les différents types de programmation qu'on a utilisée dans notre étude.

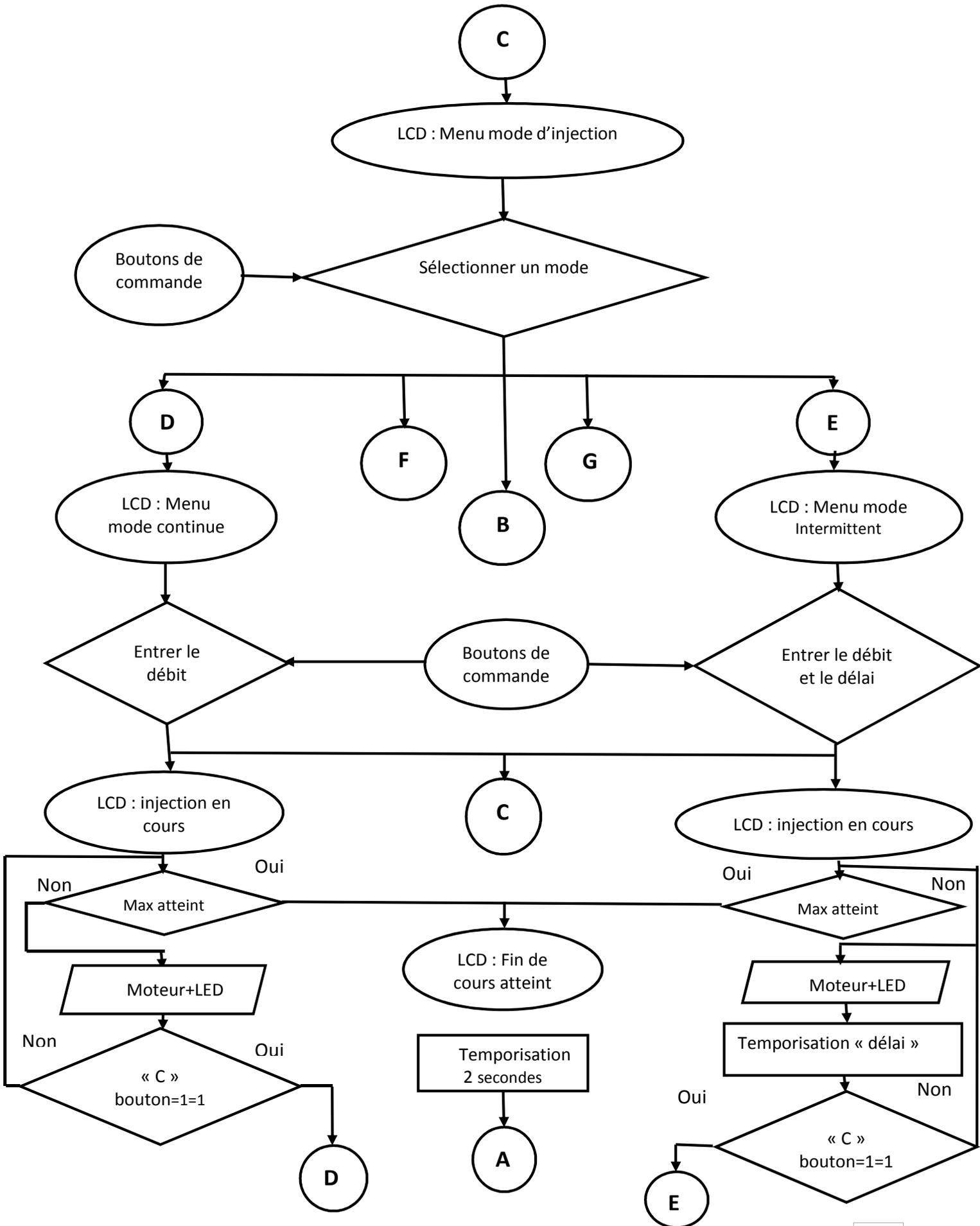
Dans l'organigramme de la figure 3.17, on va essayer de mettre en évidence les différentes parties de notre programme.



Chapitre 3 : Réalisation et étude du Prototype



Chapitre 3 : Réalisation et étude du Prototype



Chapitre 3 : Réalisation et étude du Prototype

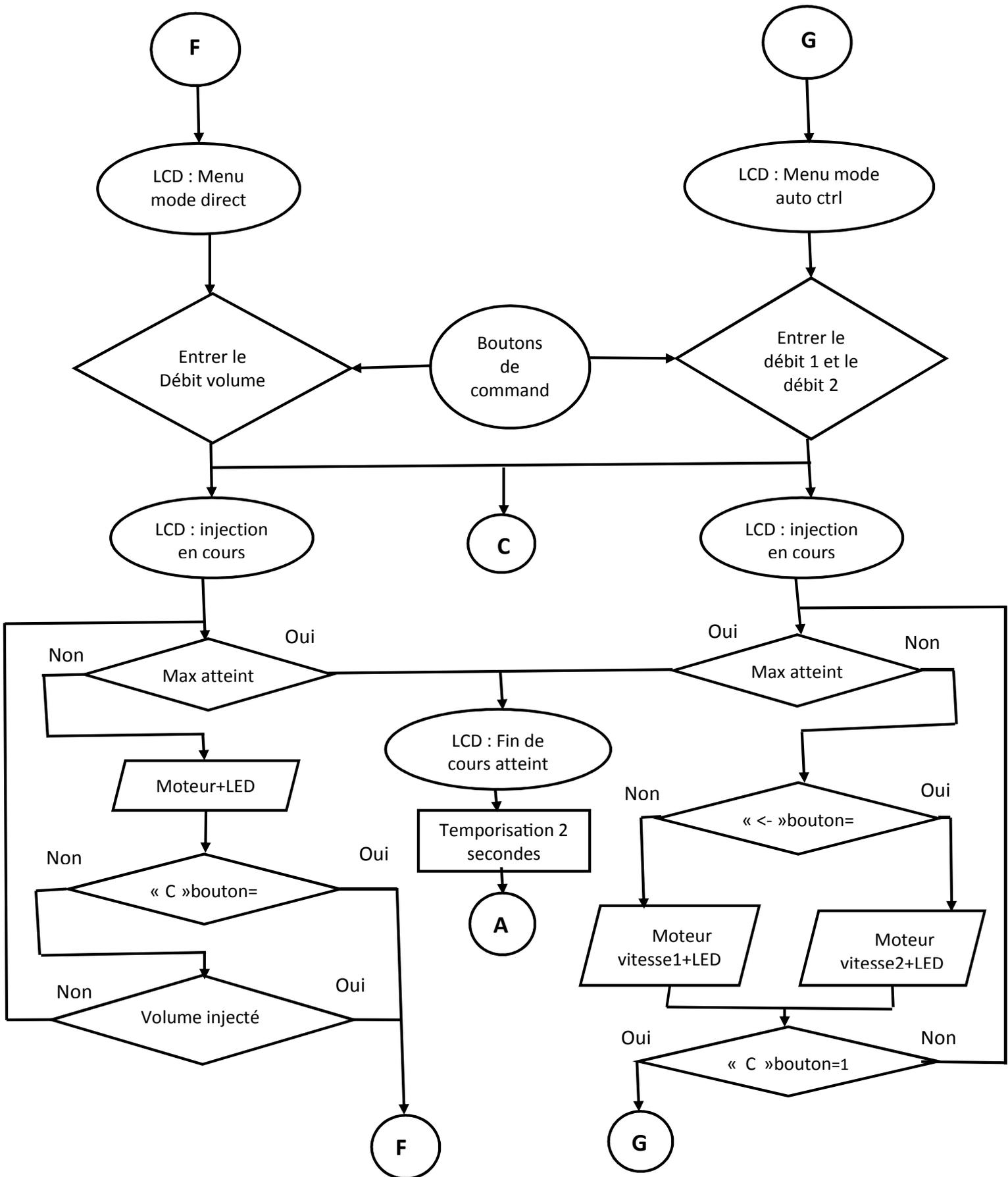


Figure 3.17 Organigrammes.

3.3.1 Structure globale du programme

Ce programme simple comprend :

- Un entête déclaratif
- Une partie « configuration » qui ne sera exécutée qu'une fois (fonction setup ())
- Une partie constituée d'une boucle sans fin que le programme répètera à l'infini (Fonction loop ()) : c'est le cœur du programme.
- **Déroulement du programme**

Le programme se déroule de la façon suivante (Figure 3.18) :

- Après avoir pris en compte les instructions de la partie déclarative,
- Puis après avoir exécuté la partie configuration (fonction setup ()),

Le programme bouclera sans fin (fonction loop ()), exécutant de façon répétée le code compris dans la boucle sans fin.

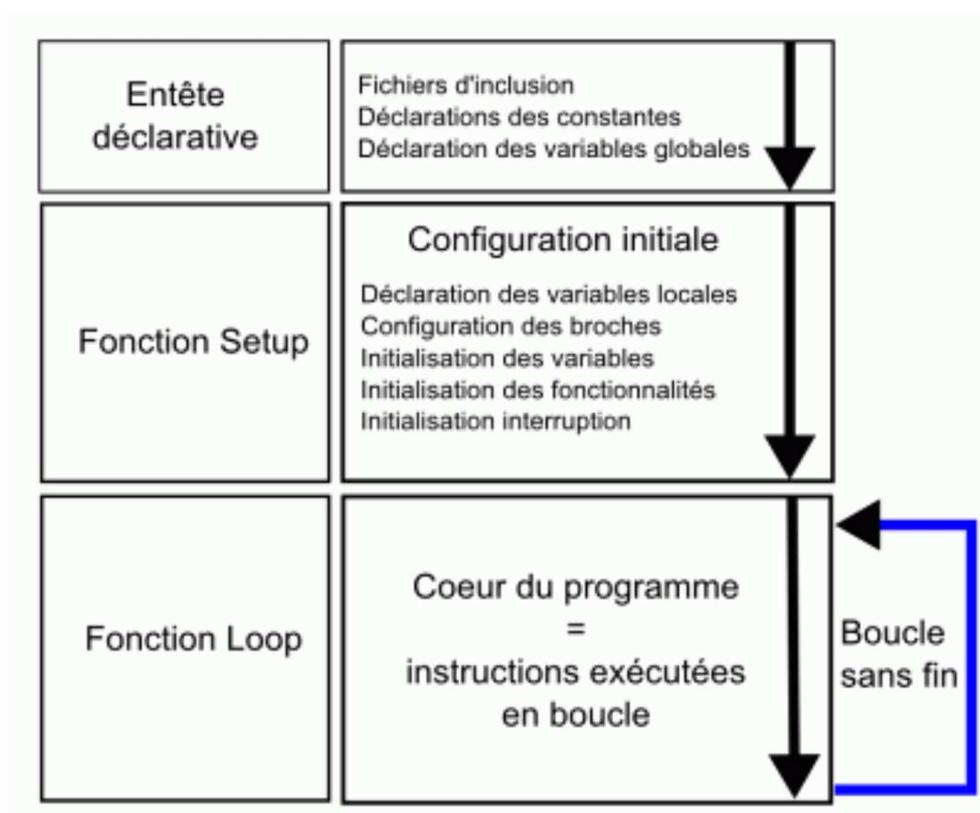


Figure 3.18 : Déroulement du programme.

Chapitre 3 : Réalisation et étude du Prototype

➤ Les variables

Les variables comme son nom l'indique sont conçu pour subir des variations et prendre des valeurs bien distinctes, selon des cas individuels et en fonction des circonstances d'un système.

En programmation une variable est un espace mémoire dans lequel on y stock de façon continue une valeur à exploiter.

Dans notre programme on a utilisé trois types de variables (Tableau 3.3) :

Types	Nature de la variable	Exemple de déclaration	Valeur de donnée
int	Nombre entier pouvant exploiter les valeurs négatives	int variable = 9999	32 768 à -32 767
Float	Nombre à virgule pouvant exploiter les valeurs négatives	float variable = 9999.99;	-3.4*10 ⁻³⁸ à 3.4*10 ³⁸
boolean	1 ou 0, true ou false, HIGH ou LOW.	boolean variable = true; boolean variable = LOW; boolean variable = 0;	

Tableau 3.3 : Types de variables.

Syntaxe :

```
int variable = 100; // commentaires
```

```
Type nom de la variable = valeurs de données ; // Variable.
```

➤ Les boucles

Pour organiser notre code et pour que celui-ci devienne plus lisible et facile à débogger.

On a utilisé des instructions très utiles telles que les boucles et les fonctions. Une boucle est une instruction qui permet de répéter un bout de code. Cela va nous permettre

Chapitre 3 : Réalisation et étude du Prototype

de faire répéter un bout de programme ou un programme entier. Il existe deux types principaux de boucles :

- ✓ La boucle conditionnelle, qui teste une condition et qui exécute les instructions qu'elle contient tant que la condition testée est vraie.
- ✓ La boucle de répétition, qui exécute les instructions qu'elle contient, un nombre de fois prédéterminé.

Exemple de boucles utilisées dans notre programme : while, for

- ✓ La boucle while ()

Ça veut dire Tant que (Vrai).

C'est donc dans cette fonction While () où l'on va écrire le contenu du programme. Il faut savoir que cette fonction est appelée en permanence.

C'est-à-dire qu'elle est exécutée une fois, puis lorsque son exécution est terminée, on la réexécute encore et encore, on parle de boucle Infinie.

Syntaxe :

```
while (condition) {  
    //les instructions entre ces accolades sont répétées TANT  
    QUE la Condition est vrai  
}
```

- ✓ La boucle for :

Une boucle bien particulière. Ce qu'elle va nous permettre de faire est assez simple, cette boucle est exécutée X fois. On doit lui donner trois paramètres.

D'abord, on crée la boucle avec le terme for (signifie "pour que"), ensuite, entre les parenthèses, on doit donner trois paramètres qui sont : ➤ La création et l'assignation de la variable à une valeur de départ. ➤ Suit de la définition de la condition à tester ➤ Suit de l'instruction à exécuter

Syntaxe :

```
for (int compteur = 0; compteur < 5; compteur++)  
{  
    //code à exécuter  
}
```

On a évité de travailler avec boucle loop parce qu'elle se comporte comme une boucle infinie, puisqu'elle se répète après avoir fini d'exécuter ses tâches.

➤ Les sous-programmes (fonction)

Dans un programme, les lignes sont souvent très nombreuses. Il devient alors impératif de séparer le programme en petits bouts afin d'améliorer la lisibilité de celui-ci, en plus d'améliorer le fonctionnement et de faciliter le débogage.

On a créé des fonctions qui nous ont permis de convertir les commandes provenant d'utilisateur en impulsions électriques pour faire tourner le moteur dans un temps bien déterminé. Pour fabriquer notre fonction, on a utilisé trois choses :

- ✓ Le type de la fonction qu'on souhaite créer,
- ✓ Le nom ;
- ✓ Les paramètres nécessaires utilisés par la fonction.

3.4 Conclusion

Dans ce chapitre, nous avons abordé en détail les différentes phases d'élaboration de la carte électronique de commande. On a étudié la partie hardware et ceci en expliquant les différentes parties qui composent le prototype de pousse seringue réalisé, ainsi que leur fonctionnement respectif, et à la fin, on a entamé la partie software composée du programme qui sert à faire fonctionner la carte de commande. Dans le chapitre suivant, on abordera la partie expérimentation du système réalisé avec quelques applications pour vérifier le bon fonctionnement et les différents modes d'utilisation.

Chapitre 4

Test et résultats

4.1 Introduction

Dans ce dernier chapitre, on va aborder l'aspect expérimental de ce qu'on a vu dans les chapitres précédents. On présente par la suite la mise en œuvre du prototype par les tests de ses performances suivi par la présentation de ses différentes configurations et les modes d'injection proposés dans ce travail, car la phase d'essai est très importante et représente une étape cruciale dans la mise en marche du système, et enfin, on terminera par le mode d'emploi de ce prototype, afin d'aider l'utilisateur pour une meilleure maîtrise et avec une grande efficacité.

4.2 La sensibilité de prototype

Lors du pilotage du moteur pas à pas à travers la carte Arduino, deux paramètres sont indispensables :

- Nombres de pas par tour en précisant le sens inverse par un signe moins, sachant que le moteur s'arrête dès que le nombre de pas est atteint.
- Vitesse avec laquelle le moteur tourne exprimée en rotation par minute.

A cet effet, l'exploitation de ces deux paramètres, peut nous conduire à déterminer la sensibilité de notre moteur, qui traduit en fin de compte la quantité de solution injectée en ml par pas, afin d'injecter et d'administrer avec une très grande précision.

Pour cela, on a essayé de déterminer cette précision en faisant tourner ce moteur avec une vitesse constante (80r/min) avec un nombre de pas indéterminé.

On termine par le calcul du temps écoulé pour lequel le piston injecte 5ml.

Nous avons obtenu les résultats suivants :

Le système a besoin de 87s (1min 27s) pour injecter 5 ml de solution.

5 ml \longrightarrow 87 Second

Suivant la vitesse utilisée le moteur effectue 80 tours/min

80 tours \longrightarrow 60 Second

x \longrightarrow 87 Second

$$x = \frac{80 \times 87}{60} = 116 \text{ tours.}$$

Ce qui signifie, que nous avons besoin de 116 tours pour injecter les 5 ml de solution.

Chapitre 4 : Test et résultats

D'après le datasheet du moteur utilisé dans ce prototype, il possède un nombre de pas par tour de 200pa/tour.

1 tour \longrightarrow 200 pas

116 tours \longrightarrow y

$$y = \frac{116 \times 200}{1} = 23200 \text{ pas.}$$

Donc il faut 23200 pas pour pouvoir injecter les 5 ml de solution.

23200 pas \longrightarrow 5 ml

1 pas \longrightarrow b

$$b = \frac{1 \times 5}{23200} = 0.000215 \text{ ml/pas.}$$

Donc la précision du prototype est donnée par le nombre " b " qui vaut 0.000215ml.

Cette sensibilité nous facilite donc l'exploitation des modes cités auparavant, pour mieux comprendre on proposera un exemple.

Exemple :

Supposons qu'on veut injecter avec un débit de 1 ml/min

La sensibilité de notre prototype est de 0.000215ml/pas

1pas \longrightarrow 0.000215ml

X \longrightarrow 1ml

Donc :

$$x = \frac{1 \times 1}{0.000215} = 4651 \text{ pas.}$$

Il faut donc 4651 pas pour injecter 1 ml.

Sachant que le nombre de pas par tour est 200.

1 tours/min \longrightarrow 200 pas

y tours/min \longrightarrow 4651pas

Donc :

$$y = \frac{4651 \times 1}{200} = 23.255 \text{ tours/min.}$$

Donc on doit ajuster la vitesse de moteur à 23 tours/min.

4.3 Phase des tests

On va utiliser de l'eau, l'air et de l'huile dans notre test

On alimente notre dispositif et on effectue notre teste en changeant le produit à injecter (dans notre teste eau, air et l'huile) pour les modes d'injections continu et direct, et on vérifie le temps d'injection et le volume injecté.

4.3.1 Mode continue

Dans la phase de test de mode continu (Tableau 4.1), on a choisi un débit de 1ml/min, et on a utilisé un chronomètre pour déterminer le volume injecté à chaque minute.

Volume (ml)	Temps (min)	Air (ml)	Eau (ml)	Huile (ml)
1	1	0.99	0.97	0.99
2	2	1.99	1.95	1.98
3	3	2.98	2.97	2.97
4	4	3.99	3.96	3.98
5	5	4.98	4.97	4.98

Tableau 4.1 : Test du mode continue.

4.3.2 Mode direct

Pour le mode direct (Tableau 4.1), on a ajusté le débit à la valeur 1 ml/min, et on a comparé les volumes injectés par le prototype (réel) avec ceux qu'on a programmé.

Volume programmé (ml)	Air (ml)	Eau (ml)	Huile (ml)
0.25	0.25	0.24	0.25
0.75	0.74	0.74	0.74
1	0.97	0.96	0.97
1.5	1.48	1.46	1.46
2.25	2.25	2.23	2.23
2.5	2.5	2.48	2.49
3.5	3.47	3.46	3.47

Tableau 4.2 : Test du mode direct.

On remarque qu'il n'y a pas de différence majeure entre la perfusion des fluides injectés par le prototype, et ceci revient à la précision ainsi que la puissance du moteur.

4.4 Mode d'emploi

Dans cette partie, on va présenter en détail le mode d'utilisation afin de faciliter, et d'aider le manipulateur, pour une meilleure exploitation de ce prototype.

Au début et lors de la mise en tension du système, un voyant témoin rouge s'allume, des messages d'initialisation vont apparaître sur l'afficheur LCD (noms et l'année), traduisant le démarrage ainsi que la préparation de l'opération d'injection (Figure 4.1).



Figure 4.1 : Initialisation LCD.

L'étape suivante consiste à la mise en position zéro du curseur, poussé par le piston, qui correspond au volume total de la seringue. Lors du recul du piston, un message sur l'afficheur LCD apparaîtra « Curseur en position initial patienté » (Figure 4.2). Le moteur ne cessera de tourner tant que la fin de course minimale n'est pas actionnée.



Figure 4.2 : Recul du curseur pour avoir la position 0.

Dans le cas où le curseur est déjà positionné sur le point zéro, c'est à dire que la fin de course est déjà actionnée, l'étape de recul est ignorée.

A ce niveau le LCD affichera un message « commencer » (Figure 4.3).



Figure 4.3 : Message commencez.

4.4.1 Menu principal

Par la suite, le LCD affiche un menu (le menu principal), Comme le montre la figure 4.4, trois options sont affichées (ajuster le curseur, mode d'injection et type de seringue).

Pour défiler le curseur entre les trois options, on appuie sur les boutons "haut" et "bas", et pour sélectionner une option, on appuie sur le bouton "ok".



Figure 4.4 : Menu principal.

4.4.2 Type de seringue

Si on sélection l'option « type de seringue » (Figure 4.5), le LCD affiche un menu pour sélectionner la seringue utilisée dans notre expérience, on peut sélectionner un des trois types (5 ml, 10 ml ou 15ml).

Après avoir sélectionné le type de seringue, on appuie sur le bouton "retour" pour revenir au menu principal.



Figure 4.5 : Type de seringue.

4.4.3 Ajuster le curseur

Ce menu (Figure 4.6) permet d'ajuster le curseur, on appuie sur les boutons "gauche" et "bas" pour avancer le curseur (bouton "gauche" pour une vitesse de rotation inferieur, le bouton "bas" pour une vitesse de rotation supérieur), ou on appuie sur les boutons "droit" et "haut" pour faire reculer le curseur (bouton "droit" pour une vitesse de rotation inferieur, le bouton "haut" pour une vitesse de rotation supérieur).

Après avoir ajusté le curseur, on appuie sur le bouton "ok" pour passer au menu "mode d'injection"



Figure 4.6 : Ajuster le curseur.

4.4.4 Mode d'injection

Après avoir ajusté le curseur, on passe au menu mode d'injection (Figure 4.7), dans ce menu, on fait défiler le curseur pour sélectionne un des quatre modes d'injection (continue, intermittent, direct ou auto contrôlé).

On appuie sur le bouton "ok" pour sélectionner un mode, ou sur le bouton "retour" pour revenir au menu principal.



Figure 4.7 : Menu mode d'injection.

a) Mode continue

Au mode continu, le manipulateur utilise les boutons "droit" et "gauche" pour introduire le débit souhaité.

Ensuite, il appuie sur le bouton "ok" pour la validation (début d'injection), ou sur le bouton "retour" pour revenir au menu mode d'injection.



Figure 4.8 : Mode continue.

b) Mode intermittent

Tout comme le mode continu, le manipulateur introduit le débit d'injection, mais aussi le volume à injecter par cycle, qui est exprimé en ml, et délai de repos en seconde (Figure 4.9).

Ensuite, il appuie sur le bouton "ok" pour la validation (début d'injection), ou sur le bouton "retour" pour revenir au menu mode d'injection.



Figure 4.9 : Mode intermittent.

c) Mode direct

Au mode direct, le manipulateur utilise les boutons de commande pour introduire le débit et le volume souhaité à injecter (Figure 4.10).

Ensuite, il appuie sur le bouton "ok" pour la validation (début d'injection), ou sur le bouton "retour" pour revenir au menu mode d'injection.

Après avoir terminé l'injection, le système revient automatiquement au menu mode direct.



Figure 4.10 : Mode direct.

d) Mode auto contrôlée

On utilise ce mode lors d'analgésie contrôlée par le patient, c à d, quand le patient sent des douleurs, il appuie sur un bouton pour faire augmenter le débit d'injection de calmement (morphine).

Au mode auto contrôlé, le manipulateur doit introduire deux débit "débit1" et le "débit2" qui est automatiquement supérieur au "débit1".

Ensuite, le manipulateur appuie sur le bouton "ok" pour le début d'injection.

L'injection fera avec le "débit1", lorsque le patient appuie sur le bouton "gauche", le système passe au "débit2". Le système revient automatiquement au "débit1" lorsque le bouton "gauche" est relâché (Figure 4.11).



Figure 4.11 : Mode auto contrôle.

Eventuellement dans les milieux hospitaliers, des événements imprévus surviennent toujours. Quand on appuie sur le bouton « Retour » ça permet de stopper d'une manière brusque le système.

Dès que le curseur atteint la fin de course maximale, le système arrête d'injecter la solution, et le LCD affiche un message "remplir la seringue", par la suite le curseur revient automatiquement à la position zéro (Figure 4.12).



Figure 4.12 : Fin d'injection.

4.5 Conclusion

Notre réalisation a été faite avec succès, car on l'a entamé par une partie de simulation.

Dans ce chapitre on a présenté notre système en mode opérationnel, en faisant différents tests sur des variétés de fluides, le type de moteur, et la précision du microcontrôleur nous a conduit à des résultats satisfaisants en termes de sensibilité (de l'ordre de 0.000215 ml/pas), en temps de réponse et enfin, en mesures de sécurité.

On a terminé ce paragraphe avec un mode d'emploi afin d'aider le manipulateur à maîtriser notre système le plus correctement possible.

Conclusion générale

Dans certains cas, la quantité de produit administrée par injection à un patient doit être fractionnée dans le temps. La prise en une seule injection du médicament ne permet pas de maintenir un effet optimal et constant de l'action thérapeutique.

Le pousse seringue électrique (PSE) reste le moyen le plus pratique et le plus précis permettant l'administration des médicaments à débit continu et constant. C'est une pompe doseuse capable de délivrer par intraveineuse une dose médicamenteuse définie avec précision.

Durant tout le temps qui nous a été alloué pour effectuer ce travail, on a essayé du mieux qu'on pouvait, d'expliquer avec beaucoup de détail le fonctionnement d'un pousse seringue électrique avec tout ce que ce nom englobe comme étapes à étudier.

D'abord on a expliqué la partie mécanique qui est basée sur le moteur pas à pas, que l'on a étudié en détail dans le chapitre 2. Ensuite et dans le même chapitre, une étude était faite sur la carte Arduino, le cerveau du commande de projet.

Notre prototype est muni des protocoles Bouton/Ecran LCD, qui est un mode de communication moderne et rapide ajouté à nos deux composants de base le moteur pas à pas et la carte Arduino, afin d'avoir une meilleure communication entre l'utilisateur et le prototype pousse seringue. Ceci a été fait en détail dans le chapitre 3 avec toutes les explications nécessaires pour bien comprendre le fonctionnement.

Ensuite on a fait quelques tests de performance par la mesure des différents paramètres essentiels pour le prototype tel que le temps d'injection et le calcul de précision. Un mode d'emploi est fait pour assurer la bonne manipulation en expliquant le rôle et le fonctionnement des boutons, touches et LED, tout ça est bien mentionné dans le chapitre 4.

Pour conclure, on souhaiterait qu'il y'ait une continuation pour cette étude par d'autres étudiants, afin d'apporter d'autres perfectionnements possibles : contrôle à distance par exemple, car chaque conception pratique a besoin d'une amélioration pour

une éventuelle augmentation de ces performances, et c'est le but qu'il faut toujours chercher à atteindre.

Bibliographie

- [1] - Anon, (2019). [En ligne] Disponible à l'adresse : <https://emmed.com/produit/starcare-pousse-seringue-electrique> [Consulté le 24 juin 2019].
- [2]- Dumas.ccsd.cnrs.fr. (2019). [En ligne] Disponible à l'adresse : <https://dumas.ccsd.cnrs.fr/dumas-01654991/document> [Consulté le 24 juin 2019].
- [3] - courantes, P., Acidose métabolique, r. et nébulisation, L. (2019). Formation Médecine d'urgence. [En ligne] Efurgences.net. Disponible à : <http://efurgences.net/> [Consulté le 24 juin 2019].
- [4] - Réservé, C. (2019). fumed1.com - Forum de l'union médicale : Tunisie, Algérie, Maroc. [En ligne] Fumed1.com. Disponible à : <http://fumed1.com.compositesite.com/> [Consulté le 24 juin 2019].
- [5] - Utc.fr. (2019). Formation, Recherche, Innovation - UTC. [En ligne] Disponible à l'adresse : <http://www.utc.fr> [Consulté le 24 juin 2019].
- [6] - Vincent, J., De Backer, D., Creteur, J., Grimaldi, D., Gustot, T. et Jacobs, F. (2017). Manuel de réanimation, soins intensifs et médecine d'urgence.
- [7] - *Bbraun.com. (2019). B. Braun partage son expertise. [En ligne] Disponible à l'adresse : <https://www.bbraun.com> [Consulté le 24 juin 2019].*
- [8] - Infirmiers.com. (2019). Infirmiers.com - communauté infirmière : concours infirmier, cours ifsi, tfe, actu, formations, protocoles. [En ligne] disponible à l'adresse : <https://www.infirmiers.com> [Consulté le 24 juin 2019].
- [9] - Ppe2010.kazeo.com. (2019). EPI : Pousse Seringue Hospal -. [En ligne] Disponible à l'adresse : <http://ppe2010.kazeo.com> [Consulté le 24 juin 2019].
- [10] -Mémoire master académique a.krama et a.gougui « Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde» Université KASDI MERBAH OUARGLA. 2015.
- [11] - Mini rapport de projet intitulé "CONTRÔLEUR PID UTILISANT ARDUINO» qui est soumis par S.V.D.REYVANTH, G.SHIRISH, D.SIVA VARMA, M.RAMU L'Institut d'ingénierie et de technologie de Gokaraju Rangaraju d'état de Telangana, en Inde.
- [12] - Tavernier, C. (2012). Arduino : Applications avancées. Paris : Dunod.

- [13] - Bartmann, E. (2014). Le grand livre d'Arduino. 2^{ème} ed.
- [14] -Mémoire master académique m.zaoui et s.seddiki «Etude et réalisation d'une Interface de mesure de la pression artérielle» Université ABOU BAKR BELKAID TLEMCEM. 2016.
- [15] -Jean- Noël, « livret Arduino en français », centre de ressources art sensitif.
- [16] -Mme Rahmoun « étude et réalisation d'une carte de commande »
- [17] - Transmission-expert.fr. (2019). Fluides & Transmissions - Tous les composants et équipements de transmission de puissance. [En ligne] Disponible à l'adresse : http://www.transmission-expert.fr/pdf_fabricant/00005923.pdf [Consulté le 24 juin 2019].
- [18] -[http://www.technologuepro.com/cours-capteurs-actionneurs_instrumentation industrielle/ch33-les-moteurs-pas-a-pas.pdf](http://www.technologuepro.com/cours-capteurs-actionneurs_instrumentation_industrielle/ch33-les-moteurs-pas-a-pas.pdf)
- [19] - Educyclopedia.karadimov.info. (2019). [En ligne] Disponible à l'adresse suivante : http://educyclopedia.karadimov.info/library/cours_moteur_pas_a_pas.pdf [Consulté le 24 juin 2019].
- [20] -Mémoire fin d'étude « télécommande d'un moteur pas à pas »

Annexes

Liste des composants :

1. le prototype :

- Moteur pas à pas : EM-463
- Carte Arduino UNO
- Afficheur LCD 20*4
- I2C
- 6 Boutons poussoirs
- CI : L293D
- 2 Boutons type fin de course
- 2 LED (Rouge, Verte)
- 8 resistances 10k ohm

2. Circuit d'alimentation :

- Transformateur 220 V /12 V
- 4 Diodes
- 2 régulateurs 7812 et 7809
- Capacités : 2200 uF et 2 capacités 10 uF
- 3 Capacités 100 nF
- LED (verte)
- Résistance 4.7k ohm

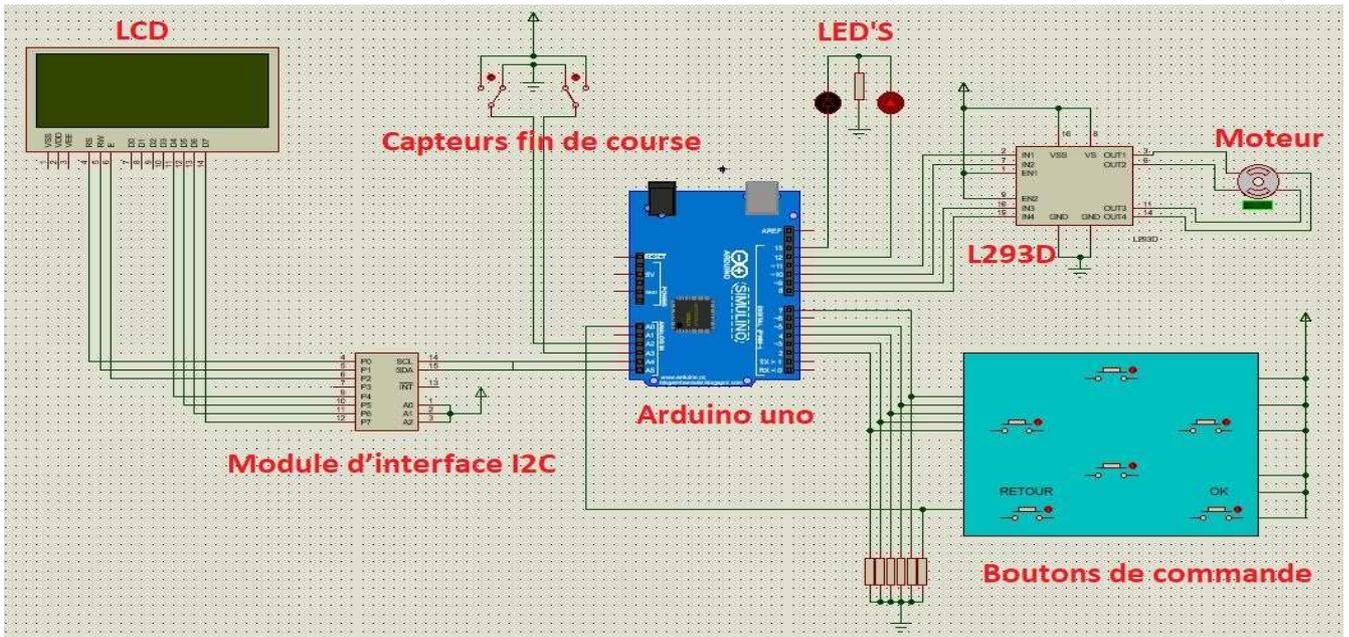
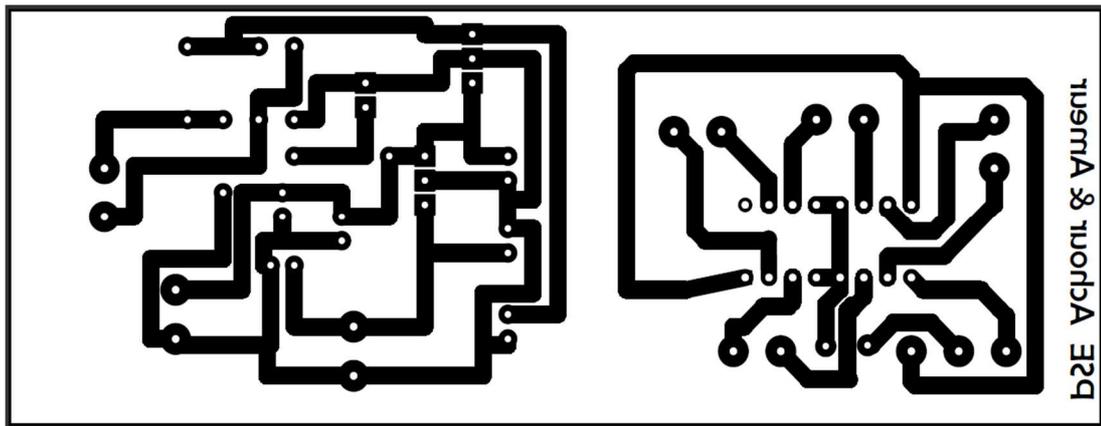
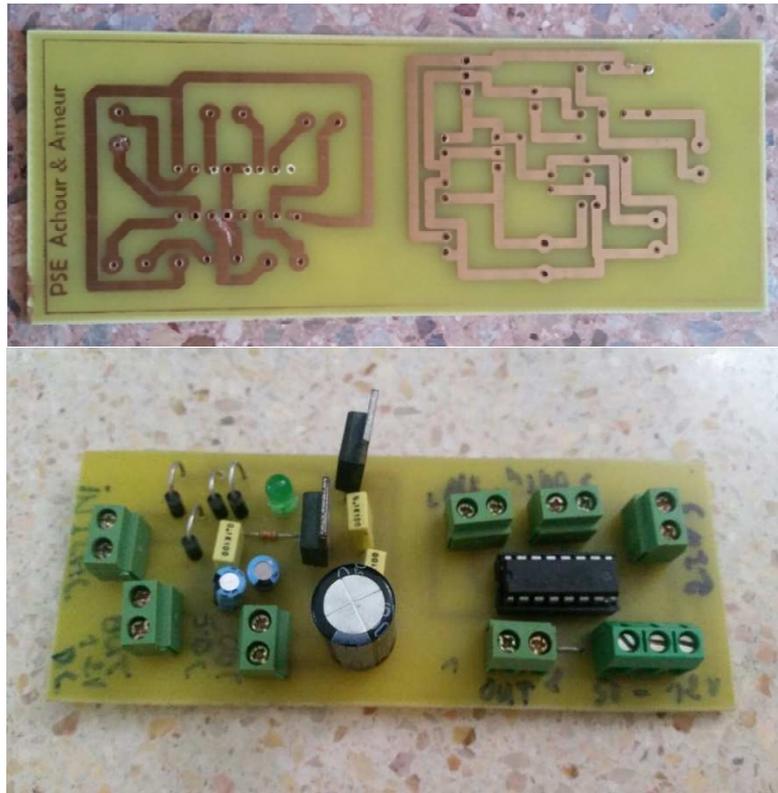


Schéma électrique de notre circuit.



Document transparent pour la commande du moteur (L293D), et l'alimentation.



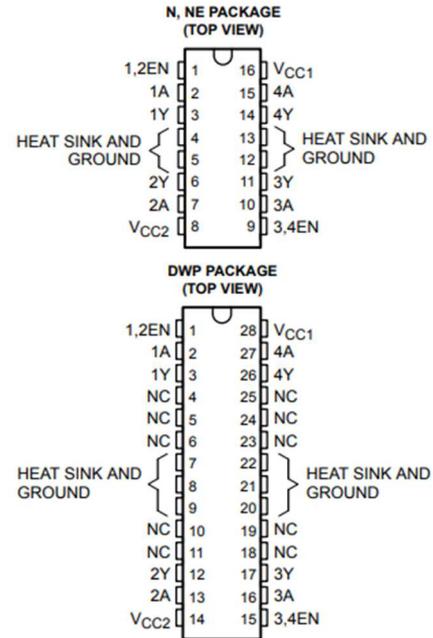
Circuit imprimé pour la commande du moteur (L293D), et l'alimentation .



photo de montage.

1. Datasheet L293D:

- Featuring Unitrode L293 and L293D Products Now from Texas Instruments
- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- Thermal Shutdown
- High-Noise-Immunity Inputs
- Functional Replacements for SGS L293 and SGS L293D
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)



Description: The L293 and L293D are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, dc and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications. All inputs are TTL compatible. Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications. On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. A VCC1 terminal, separate from VCC2, is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C

2. Datasheet: LM7805

✓ Features

- Output Current up to 1.5 A
- Available in Fixed 5-V, 12-V, and 15-V Options
- Output Voltage Tolerances of $\pm 2\%$ at $T_j = 25^\circ\text{C}$ (LM340A)
- Line Regulation of 0.01% / V of at 1-A Load (LM340A)
- Load Regulation of 0.3% / A (LM340A)
- Internal Thermal Overload, Short-Circuit and SOA Protection
- Available in Space-Saving SOT-223 Package
- Output Capacitance Not Required for Stability

✓ Description

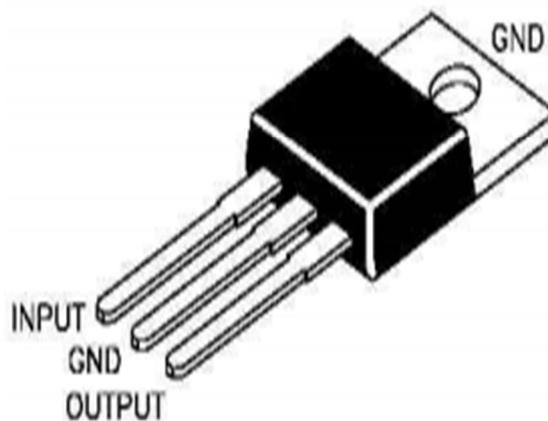
3-TERMINAL POSITIVE VOLTAGE REGULATOR

The LM340 and LM7805 Family monolithic 3-terminal positive voltage regulators employ internal current-limiting, thermal shutdown and safe-area compensation, making them essentially indestructible. If adequate heat sinking is provided, they can deliver over 1.5-A output current. They are intended as fixed voltage regulators in a wide range of applications including local (on-card) regulation for elimination of noise and distribution

problems associated with single-point regulation. In addition to use as fixed voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents.

Considerable effort was expended to make the entire series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

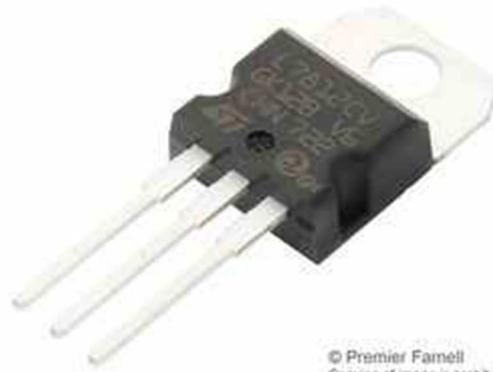
LM7805 is also available in a higher accuracy and better performance version (LM340A). Refer to LM340A specifications in the *LM340A Electrical Characteristics* table.



1. Datasheet: LM7812

✓ FEATURES ·

- Output current in excess of 1.5A
- Output voltage of 12V
- Internal thermal overload protection
- Output transition Safe-Area compensation



© Premier Farnell
Copying of image is prohibited

✓ Table Electrical characteristics of LM7812

Symbol	Parameter	Test conditions	Min.	Typ.	Max.	Unit
V_O	Output voltage	$T_J = 25\text{ }^\circ\text{C}$	11.75	12	12.25	V
V_O	Output voltage	$I_O = 5\text{ mA to }1\text{ A}, V_I = 14.8\text{ to }25\text{ V}$	11.5	12	12.5	V
V_O	Output voltage	$I_O = 1\text{ A}, V_I = 25\text{ to }27\text{ V}, T_J = 25\text{ }^\circ\text{C}$	11.5	12	12.5	V
$\Delta V_O^{(1)}$	Line regulation	$V_I = 14.8\text{ to }30\text{ V}, I_O = 500\text{ mA}, T_J = 25\text{ }^\circ\text{C}$		13	120	mV
		$V_I = 16\text{ to }12\text{ V}$		16	120	mV
		$V_I = 16\text{ to }12\text{ V}, T_J = 25\text{ }^\circ\text{C}$		6	60	mV
		$V_I = 14.5\text{ to }27\text{ V}, T_J = 25\text{ }^\circ\text{C}$		13	120	mV
$\Delta V_O^{(2)}$	Load regulation	$I_O = 5\text{ mA to }1\text{ A}$		25	100	mV
		$I_O = 5\text{ mA to }1.5\text{ A}, T_J = 25\text{ }^\circ\text{C}$		30	100	
		$I_O = 250\text{ to }750\text{ mA}$		10	50	
I_q	Quiescent current	$T_J = 25\text{ }^\circ\text{C}$		4.4	6	mA
					6	mA
DI_q	Quiescent current change	$V_I = 15\text{ to }30\text{ V}, I_O = 500\text{ mA}$			0.8	mA
		$V_I = 14.8\text{ to }27\text{ V}, T_J = 25\text{ }^\circ\text{C}$			0.8	mA
		$I_O = 5\text{ mA to }1\text{ A}$			0.5	mA
SVR	Supply voltage rejection	$V_I = 15\text{ to }25\text{ V}, f = 120\text{ Hz}, I_O = 500\text{ mA}$		60		dB
V_d	Dropout voltage	$I_O = 1\text{ A}, T_J = 25\text{ }^\circ\text{C}$		2		V
eN	Output noise voltage	$T_A = 25\text{ }^\circ\text{C}, B = 10\text{ Hz to }100\text{ kHz}$		10		$\mu\text{V}/V_O$
R_O	Output resistance	$f = 1\text{ kHz}$		18		m Ω
I_{sc}	Short circuit current	$V_I = 35\text{ V}, T_A = 25\text{ }^\circ\text{C}$		0.2		A
I_{scp}	Short circuit peak current	$T_J = 25\text{ }^\circ\text{C}$		2.2		A
$\Delta V_O/\Delta T$	Output voltage drift			-1		mV/ $^\circ\text{C}$