

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR- ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR ANNABA



جامعة باجي مختار - عنابة

Année : 2019

Faculté: Sciences de l'Ingénierat
Département: Electronique

MEMOIRE

Présenté en vue de l'obtention du diplôme de : MASTER

Intitulé :
**Conception et supervision d'un contrôleur
automatique dédié au système d'incubation**

Domaine : Sciences et Technologie

Filière : Automatique

Spécialité : Automatique et informatique industrielle

Par :

HADOUNE Oussama

DEVANT Le JURY

Président :	M. Bekaik	Grade : Dr	UBM Annaba
Directeur de mémoire :	M. Benouaret	Grade : Pr	UBM Annaba
Examineurs :	M. Arbaoui	Grade : Dr	UBM Annaba
	M. Ramdani	Grade : Pr	UBM Annaba

Résumé :

Aujourd'hui, plusieurs espèces animales et végétales ont été disparues, et nombreuses qui sont en voie de disparition, l'écosystème est en déséquilibre, mettre en place un système automatisé pour remédier à ce problème est devenu indispensable. Pour cela, nous allons concevoir et réaliser un système permettant la production des poussins (poule et toute autre espèce ovipare), appelé incubateur, se basant sur le protocole de communication I2C et utilisant des cartes Arduino pour faire la régulation de la température et de l'hygrométrie, un système mécanique est mis en place assurer le retournement des œufs, ce qui rend le système complètement autonome.

ملخص :

اليوم، العديد من الأنواع الحيوانية والنباتية انقرضت، والكثير من الأنواع مهددة بالانقراض، النظام البيئي غير متوازن، وإنشاء نظام آلي لمعالجة هذه المشكلة أصبح لا غنى عنه. لهذا الغرض، سنقوم بتصميم وبناء نظام لإنتاج الكتاكيت (الدجاجة وغيرها من الأنواع البيضية)، يسمى بالحاضنة، بناءً على بروتوكول اتصال I2C واستخدام لوحات أريدينو لتنظيم درجة الحرارة وقياس الرطوبة بالإضافة إلى إعداد نظام ميكانيكي لضمان انقلاب البيض، مما يجعل النظام مستقل تمامًا.

Abstract :

Today, several animal and plant species have disappeared, and many that are endangered, the ecosystem is out of balance, setting up an automated system to remedy this problem has become essential. For this purpose, we will design and build a system allowing the production of chicks (hen and any other oviparous species), called an incubator, based on the I2C communication protocol and using Arduino cards to regulate temperature and humidity, a mechanical system is set up to ensure the turning of the eggs, which makes the system completely autonomous.

Remerciement

Ce mémoire est le résultat d' un travail de recherche qui a duré 06 mois.

En préambule, je tiens tout particulièrement à remercier Monsieur, le professeur Benouaret qui, en tant que directeur de mémoire, s' est toujours montré à l' écoute et très disponible tout au long de la réalisation de ce travail, ainsi que pour l' inspiration, l' effort fourni et le temps qu' il nous a consacré depuis le début de l' année, et sans qui ce mémoire n' aurait jamais vu le jour.

Je remercie également mes parents qui m' ont soutenu, encouragé et qui ont toujours répondu présent, et sans qui je n' aurai jamais pu arriver à ce stade.

Enfin, j' adresse mes plus sincères remerciements à nos proches, amis, et toute personne ayant participé à l' élaboration de ce mémoire.

Je dédie ce mémoire,

A mes chers parents qui m'ont fourni les conditions nécessaires pour la poursuite de mes études et qui se sont sacrifiés pour me voir réussir, leur présence et leur soutien tout au long de mon parcours universitaire m'ont été ma source d'inspiration et de motivation.

A mes chères sœurs, qui elles seules savent me redonner le sourire et qui répondent présent à chaque fois que j'ai besoin d'elles.

Et enfin à toute ma famille que j'aime ainsi qu'à mes amis qui m'avaient toujours soutenu et encouragé durant ces années d'études.

Ce travail vous est dédié

Liste des tableaux :

Tableau 1. Processus du développement de l'embryon	22
Tableau 2. Influence des paramètres K_p , K_i et K_d sur la réponse du système	39
Tableau 3. Calcul des paramètres du contrôleur PID.....	41
Tableau 4. Calcul des paramètres du contrôleur PID par la méthode fréquentielle.....	42
Tableau 5. Rôle des différents éléments d'une carte Arduino	50

Table des figures :

Figure I.1. Incubateur statique	16
Figure I.2. Structure d'un œuf de poule	19
Figure I.3. Technique du mirage sur un œuf de poule	20
Figure I.4. Sens du courant circulant dans la bobine	27
Figure I.5. Polarité de la puissance appliquée sur chaque borne du moteur	27
Figure I.6. Séquence du demi-pas d'un moteur pas à pas	28
Figure II.1. Principe de régulation en boucle ouverte.....	31
Figure II.2. Principe de régulation en boucle fermée.....	32
Figure II.3. Graphe de régulation tout ou rien	33
Figure II.4. Régulation à action continue/discontinue	34
Figure III.5. Configuration d'une boucle de régulation.....	36
Figure II.6. L'effet du paramètre K_p sur la réponse du système	38
Figure II.7. L'effet du paramètre K_i sur la réponse du système	38
Figure II.8. L'effet du paramètre K_d sur la réponse du système	39
Figure II.9. Réponse indicielle d'un système.....	41
Figure II.10. Détermination de la valeur de P_{cr}	42
Figure III.1. Circuit de branchement du capteur DS18B20.....	46
Figure III.2. Circuit de branchement du capteur DHT22	46

Figure III.3. Circuit de branchement du module RTC.....	47
Figure III.4 Afficheur LCD à base du protocole I2C	48
Figure III.5. Driver TB6600	49
Figure III.6. Description des différents composants d'une carte Arduino	50
Figure III.7. Interface d'utilisation du logiciel Arduino	52
Figure III.8. Interface d'utilisation du logiciel Eagle	53
Figure III.9. Interface d'utilisation du logiciel Isis Proteus.....	54
Figure III.10. Circuit de simulation d'un clavier pour un menu déroulant.....	54
Figure III.11 Schéma synoptique du menu déroulant.....	55
Figure III.12. Circuit de simulation du système de régulation	56
Figure III.13 Schéma électrique d'un pont en H	57
Figure III.14. Routage du circuit électronique (version imprimée).....	58
Figure III.15. Passage de la carte sur les rayons ultra-violets	59
Figure III.16. Immersion de la carte électronique dans un bac de Na-OH.....	59
Figure III.17. Passage de la plaque dans un bac de perchlore de fer.....	60
Figure III.18. Réalisation du circuit imprimé	60
Figure III.19 Mécanisme de retournement utilisant les limites siwtch.....	61
Figure III.20. Clavier pour la commande du menu déroulant	62
Figure III.21. Carte de commande	62

Figure III.22. Carte d'affichage et de commande du moteur 63

Figure III.23. Schéma synoptique récapitulant le projet complet..... 64

Table des matières

Introduction générale.....	13
Chapitre I : De l'œuf au poussin	14
I. Introduction.....	15
I.1. Historique	15
I.2. Définition d'un incubateur	15
I.3. Les types d'incubateur	16
I.3.1. L'incubateur ventilé.....	16
I.3.2. L'incubateur statique	16
I.3.3. L'incubateur à contact	17
I.4. Catégorie des appareils d'incubation	17
I.4.1. L'incubateur de table	17
I.4.2. L'incubateur-armoire.....	17
I.4.3. L'incubateur commercial.....	17
I.5. Les matériaux de fabrication	17
I.6. Les différents paramètres à contrôler dans un incubateur.....	17
I.6.1. La température	17
I.6.2. Le taux d'humidité.....	18
I.6.3. Le positionnement d'un œuf.....	18
I.6.4. L'air et la ventilation	18
I.7. De l'œuf au poussin.....	18
I.7.1. La structure d'un œuf de poule.....	18

I.7.2. Technique de mirage	20
I.7.3. Le développement d'un embryon	20
I.8. La température, le taux d'hygrométrie, l'air et le retournement d'œufs dans un incubateur	22
I.8.1. La température	22
I.8.2. Le taux d'hygrométrie	22
I.8.3. Le retournement des œufs.....	23
I.8.4. L'éclosion	24
I.9. Mécanisme de retournement des œufs	24
I.9.1. Moteur pas à pas	25
I.9.2. Type de moteur pas à pas	25
I.10. Conclusion.....	28
Chapitre II : Régulation PID	29
II. Introduction	30
II.1. Définition de la régulation	30
II.2. Les types de régulation	30
II.2.1. Régulation en boucle ouverte	31
II.2.2. Régulation en boucle fermée.....	31
II.3. Méthode de régulation	32
II.3.1. Régulation à action discontinue (tout ou rien)	32
II.3.2. Régulation à action continue	33
II.4. Régulation PID.....	34
II.5. Structure d'une boucle de régulation PID.....	35

II.6. Principe de fonctionnement d'un régulateur PID	37
II.7. L'effet des paramètres d'un régulateur PID.....	37
II.7.1. Correcteur proportionnel	37
II.7.2. Correcteur intégral.....	38
II.7.3. Correcteur dérivé.....	39
II.8. Réglage d'un PID	40
II.9. Méthode de Ziegler Nichols.....	40
II.9.1. Méthode de Ziegler-Nichols temporelle	41
II.9.2. Méthode de Ziegler-Nichols fréquentielle	42
II.10. Conclusion	42
Chapitre IV : Conception et réalisation pratique.....	44
III. Introduction	45
III.1. Matériel utilisé.....	45
III.1.1. Capteur de température DS18B20 :	45
III.1.2. Capteur d'humidité et de température DHT22	46
III.1.3. Module RTC (Real Time Clock)	47
III.1.4. Afficheur LCD I2C	47
III.1.5. Moteur pas à pas	48
III.1.6. Driver TB6600	48
III.1.7. Pompe à eau	49
III.1.8. Une carte Arduino.....	49
III.2. Logiciel utilisé	51
III.2.1. Logiciel de programmation de la carte Arduino IDE	51

III.2.2. Eagle	52
III.2.3. ISIS Proteus	53
III.3. Conception.....	54
III.3.1. Conception d'un clavier à bouton poussoir	54
III.3.2. Conception d'un système de régulation de température :.....	55
III.3.3. Conception du circuit électronique d'un pont H.....	56
III.4. Réalisation Pratique.....	58
III.4.1. Réalisation du circuit imprimé.....	58
III.4.2. Réalisation du mécanisme du retournement des œufs	60
III.4.3. Réalisation d'un clavier à bouton poussoir.....	61
III.4.4. Réalisation du système de régulation de température/humidité	62
III.4.5. Affichage de la date/heure, le taux d'humidité et l'état du moteur.	63
III.5. Conclusion	64
Bibliographie.....	66
Annexe :	68

Introduction générale

De nos jours, les systèmes d'incubation occupent une place de plus en plus importante dans plusieurs et différents domaines (scientifiques, agriculture...etc.), l'incubateur d'œufs est un système de couvaie basé sur le principe de la création d'un environnement semblable à celui créé par l'espèce ovipare tout au long de la période de l'éclosion.

Le système que nous allons concevoir peut présenter une solution pour les animaux ovipares qui sont en voie de disparition, et ce, en augmentant le nombre de naissances des poussins, ce qui peut aider au rééquilibrage de l'écosystème.

Le système d'incubation d'œufs aviaire présente une bonne alternative pour remplacer la poule sur son nid, grâce à l'évolution de l'électronique et de l'automatique, on peut augmenter le nombre d'œufs qui éclosent en 21 jours sans avoir besoin d'aucune intervention de poules.

Plusieurs paramètres sont à ajustés pour pouvoir assurer une bonne éclosion, le taux d'humidité et le degré de la température sont les deux piliers sur lesquels se base la plateforme permettant d'assurer un pourcentage de réussite élevé.

Notre projet a donc pour vocation la réalisation d'un système d'incubation permettant d'augmenter le nombre de naissances des poussins, et ce, en créant un environnement semblable à celui d'une poule sur son nid.

Il consiste en la conception d'un incubateur d'œufs à base de protocole de communication I2C avec le maximum de matériel recyclable possible tout en appliquant les notions de l'automatique (régulation PID, TOR...etc.) et de l'électronique (circuit imprimé...etc.) acquises durant le parcours universitaire.

Chapitre I :

De l'œuf au poussin

I. Introduction

Un incubateur est une enceinte thermostatée qu'on trouve aujourd'hui dans les laboratoires scientifiques, les champs agricoles et ils ont même envahi le monde industriel, grâce à leur environnement intérieur offrant la possibilité de créer un environnement dont la température et le taux d'humidité sont facilement paramétrés.

L'incubateur que nous allons concevoir se spécialise dans l'aviculture, plus précisément dans l'incubation artificielle des œufs de poule. Pour ce faire, il faut étudier les conditions dans lesquelles un embryon peut se développer, son processus de croissance et les différents mécanismes qu'un incubateur doit équiper afin d'assurer une bonne éclosion.

I.1. Historique

Les débuts de l'incubation artificielle remontent à trois mille ans avant l'ère chrétienne, les chinois et les égyptiens étaient les premiers à l'utiliser, en Chine, ils utilisaient les œufs de canard vu que les œufs de canards étaient leur favori en cuisine.

Aujourd'hui, en Egypte, en Alexandrie plus précisément on trouve encore les Manals, des monuments construits il y a quatre ou cinq mille ans qui ont le rôle d'un incubateur artificiel d'une capacité de 90.000 œufs. Ils utilisaient la fiente des chameaux desséchée pour faire augmenter la température, et le contact de la coquille avec la paupière pour la mesurer, pour faire le mirage, ils mettaient la coquille dans la trajectoire des rayons de soleils.

Malgré que les moyens étaient très limités mais ils ont réussi quand même à obtenir un pourcentage de réussite qui varie entre 60 et 65%, l'équivalent d'un incubateur automatique industriel aujourd'hui.

En Roumanie, à cette époque ils excellaient dans la plomberie et le chauffage central, ce qui leur a facilité la fabrication des étuves dans lesquelles ils faisaient l'éclore à la vapeur d'eau.

I.2. Définition d'un incubateur

Un incubateur est un appareil qui permet de simuler un environnement où les conditions d'air et de température sont contrôlées.

Un incubateur d'œufs est un dispositif permettant de garder les œufs à une température et un taux d'humidité fixes et appropriés, afin de permettre le développement de l'embryon jusqu'à ce qu'il se détache de sa coquille.

I.3. Les types d'incubateur

Il existe trois types d'appareils d'incubation :

I.3.1. L'incubateur ventilé

Ce type d'appareils va être étudié dans ce mémoire, il est équipé :

- d'un seul ou plusieurs ventilateurs d'extraction permettant de faire diminuer la température dans le cas où elle dépasse la valeur de consigne ;
- des ventilos donnant la possibilité de dispatcher l'air et la température au sein de l'incubateur.

I.3.2. L'incubateur statique

Cet appareil n'est pas équipé d'un système de ventilation, la résistance chauffante se met au-dessus des œufs, le thermomètre soit presque à la même ligne que la hauteur des œufs qui sont placés dans un support dans le bas, ces derniers reçoivent un taux de température sur la partie supérieure élevé par rapport à la partie inférieure tout comme dans un nid, cet appareil doit impérativement être équipé de deux trous, une entrée pour faire introduire de l'air de l'extérieur à l'intérieur de la couveuse et une sortie d'évacuation.

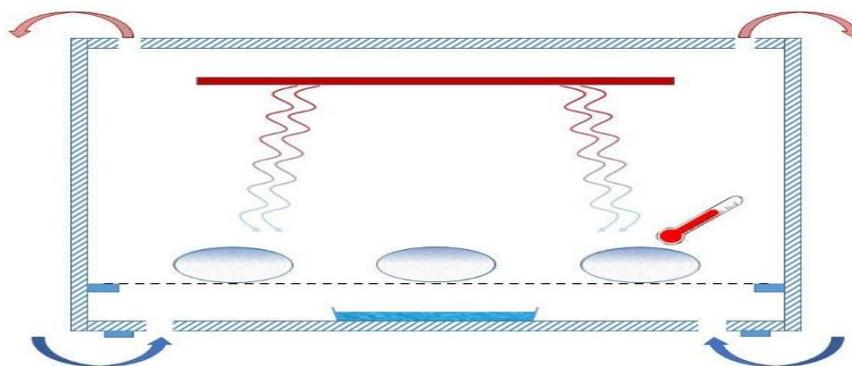


Figure I.1.Incubateur statique

I.3.3. L'incubateur à contact

C'est la dernière génération des couveuses, elle est équipée d'une membrane gonflable qui contrôle la température, située juste au-dessus des œufs, le gonflement et le dégonflement de cette membrane entraîne le changement de position des œufs.

I.4. Catégorie des appareils d'incubation

Ces appareils d'incubation peuvent être classés selon leur taille, on distingue trois catégories :

I.4.1. L'incubateur de table

Ces types de couveuses sont conçues de sorte à ce qu'elles soient placées sur une table, elles ont une capacité limitée à quelques dizaines d'œufs.

I.4.2. L'incubateur-armoire

Comme son nom l'indique, cette catégorie se caractérise par la présence d'une porte permettant et facilitant l'accès aux œufs, il peut contenir jusqu'à quelques centaines d'œufs.

I.4.3. L'incubateur commercial

Il s'agit d'un incubateur armoire d'une capacité énorme qui peut atteindre les 120000 œufs, il occupe généralement la place d'un local, contrôlé par ordinateur.

I.5. Les matériaux de fabrication

Pour la réalisation de ces appareils on peut utiliser plusieurs types de matériaux : le bois, des feuilles de métal séparées par du polystyrène expansé, du plastique ou du verre au niveau des hublots souvent double pour une bonne isolation.

I.6. Les différents paramètres à contrôler dans un incubateur

I.6.1. La température

La température joue un rôle primordial dans le processus d'éclosion d'un œuf, une petite différence entre la température de référence et la température réelle peut être fatale.

I.6.2. Le taux d'humidité

Ce paramètre est également indispensable au développement de l'embryon, le contrôle se fait par, soit un humidificateur ou bien la vapeur due à l'élévation de la température de l'eau par l'utilisation d'un chauffe-eau.

I.6.3. Le positionnement d'un œuf

Afin que l'embryon ne se colle pas à la coquille, un système mécanique qui permet le retournement des œufs est mis en place. L'explication de son principe de fonctionnement sera abordée dans les prochaines pages.

I.6.4. L'air et la ventilation

Tout incubateur doit avoir au moins deux trous permettant de faire un échange d'air entre l'intérieur et l'extérieur de l'incubateur ; l'oxygène est nécessaire pour l'éclosion de l'œuf.

La présence des ventilos dans un incubateur jouera le rôle :

-d'un répartiteur pour s'assurer d'avoir le même degré de température dans les différents coins de l'appareil ;

-d'un extracteur pour faciliter l'échange d'air dans la couveuse.

I.7. De l'œuf au poussin

Dans cette partie de mémoire, nous allons étudier l'évolution de l'embryon jusqu'à ce qu'il se décolle de sa coquille.

I.7.1. La structure d'un œuf de poule

Avant de passer aux études du développement de l'embryon, nous commencerons par définir les différents composants d'un œuf de poule, un œuf n'est pas juste un œuf. Chaque partie a sa raison d'être.

- Chalaze : les chalazes ont pour fonction de maintenir le jaune au centre de l'œuf. Cela est important, car le jaune adhérerait autrement à la paroi, ce qui entraînerait une malformation du poussin ;
- Coquille : la coquille est l'enveloppe protectrice de l'œuf. Beaucoup de gens ignorent que la coquille n'est pas entièrement close et qu'elle présente au total près de 10000 minuscules petits trous (pores). Ils permettent au poussin d'avoir suffisamment d'air pour vivre ;
- Membrane coquillière : elle protège l'intérieur de l'œuf contre les contaminations extérieures telles que les bactéries ;
- Blanc fluide : il s'agit du blanc externe, entre la membrane coquillière et le jaune ;
- Blanc visqueux : il s'agit du blanc intermédiaire, visqueux et trouble, qui enveloppe le jaune ;
- Jaune : le jaune fournit la nourriture au poussin ;
- Vésicule germinative : c'est ici qu'a lieu l'éventuelle fécondation de l'œuf ;
- Chambre à air : dans cette partie de l'œuf, le poussin peut respirer pour la première fois avant de casser la coquille.

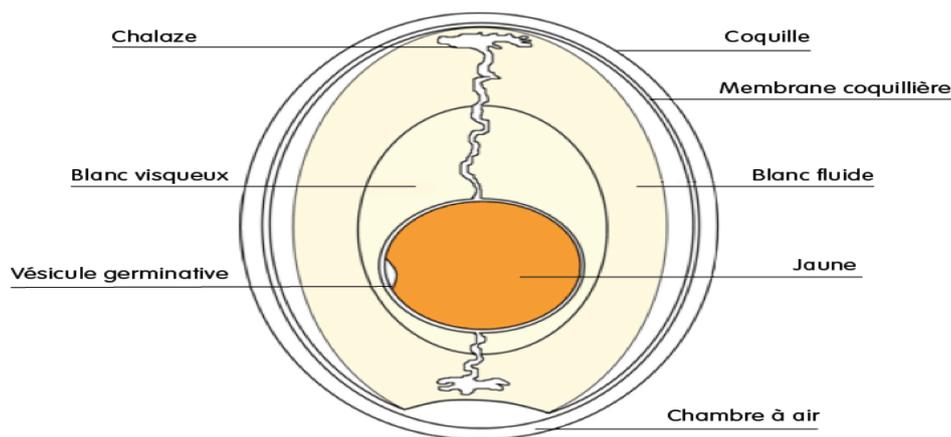


Figure I.2. Structure d'un œuf de poule

I.7.2. Technique de mirage

Tout d'abord, il faut noter que seuls les œufs fécondés sont utilisables. On appelle œuf fécondé lorsque la poule a déjà vécu avec un coq les jours qui ont précédé la ponte.

On ne peut voir toutes les transformations qui se font à l'intérieur de l'œuf qu'en utilisant le mirage. Mirer est voir l'évolution de l'embryon à l'intérieur de l'œuf en utilisant un mire-œuf (torche à une puissance importante) dans le noir, tout en faisant attention à ne pas avoir un contact direct entre l'œuf et le mire-œuf afin de ne pas augmenter sa température.

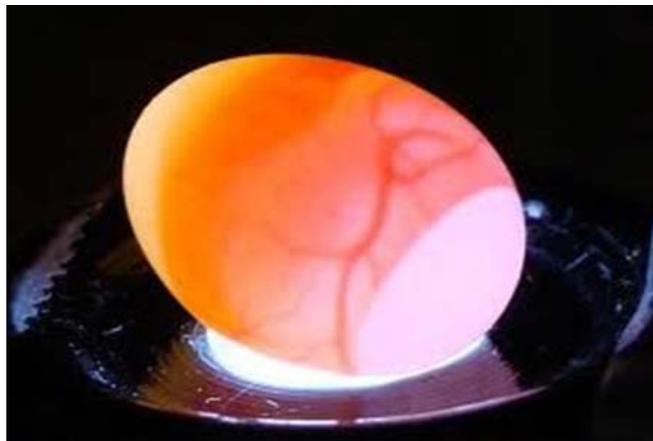


Figure I.3. Technique du mirage sur un œuf de poule

Un œuf se compose de trois niveaux, du jaune, du blanc et une poche d'air, parfois on marque la présence d'un point noir (germe) sur le jaune qui représente la cellule qui donnera l'embryon, aussi on peut les différencier à l'aide de la poche d'air, si elle est bien claire et occupe la partie supérieure on peut dire qu'il s'agit d'un œuf fécondé.

Quelques jours plus tard et dans des conditions de conservation de l'œuf, ce point noir commence à grandir et prend de plus en plus une place entre le jaune et le blanc, c'est ce qu'on appelle le développement d'un embryon.

I.7.3. Le développement d'un embryon

Un embryon se transforme en poussin en 21 jours, et ceci ne peut se faire qu'en présence de plusieurs paramètres dont on a déjà parlé dans la partie précédente.

Pendant la durée de la couvaison, l'opérateur doit mirer au moins deux fois, une fois au huitième jour et une fois aux dix huitièmes jours, ce contrôle permet de visualiser l'évolution de la poche d'air qui fait 1/10 du volume de l'œuf au début de l'incubation et qui doit en faire 1/3 à la fin. Les premiers jours d'incubation, l'embryon utilise les trous se trouvant dans la coquille pour respirer et les nutriments du jaune et du blanc pour s'alimenter.

En effet, tout se base sur la taille de la chambre à air, au cours de l'incubation, l'embryon perdra 13% de son poids en eau par les pores de la coquille, cette perte de quantité d'eau sera occupée par de l'air dans la chambre à air, qui va servir d'Oxygène quelques jours avant l'éclosion du poussin.

Deux cas sont envisageables, si la poche d'air est petite, cela veut dire l'embryon obtenu est d'une grande taille et qui ne pourra percer un trou interne vu qu'il occupe toute la place dans l'œuf, c'est-à-dire que même en lui libérant l'espace de la poche d'air, ce ne serait pas suffisant pour lui permettre de percer dans la coquille.

Dans un deuxième cas où l'embryon perd une importante quantité d'eau (plus de 13%) par les pores de la coquille qui sera remplacée par de l'air ce qui donne une poche d'air très évoluée (large) et un embryon chétif qui n'aura pas assez de force pour percer un trou dans la coquille.

Il existe une deuxième méthode qui permet de voir l'évolution de l'embryon, mais elle est beaucoup moins utilisée, elle consiste à peser le poids des œufs avant de les mettre dans la couveuse, et de réagir sur le taux d'humidité dont on parlera plus tard afin d'avoir une perte de 13% quelques jours avant l'éclosion.

La température, l'hygrométrie, le retournement et l'aération, sont des paramètres qui doivent être présents et maintenus dans le cas d'une couvaison naturelle ou automatique, dans les deux cas le processus d'obtention d'un poussin sera présenté dans le tableau 1.

1 -ier jour	Développement des tissus.
2 -ème jour	Formation du cœur et les premiers battements apparaissent.
3 -ème jour	Formation des vaisseaux sanguins.
4-7 -ème jour	Développement des membres.
8-14 -ème jours	Développement des plumes, les écailles sur les pieds et les pattes ainsi que le tournement de la tête vers l'extrémité la plus large de l'œuf.
14 -18 -ème jour	Développement des plumes, les écailles sur les pieds et les pattes ainsi que le tournement de la tête vers l'extrémité la plus large de l'œuf.
19 ème jour	L'embryon perce un conduit interne.
20 ème jour	L'embryon perce un conduit externe.
21 ème jour	L'éclosion, le poussin est maintenant à l'air libre et est prêt maintenant pour affronter de nouvelles épreuves.

Tableau 1. Processus du développement de l'embryon

I.8. La température, le taux d'hygrométrie, l'air et le retournement d'œufs dans un incubateur

I.8.1. La température

Dans une couveuse, la température idéale qui doit être maintenue fixe doit être entre 37.2 et 37.5°C, Si la température s'écarte de 0.5 degré de plus de 37.7 °C, une éclosion médiocre est probable. La température est le paramètre le plus important avec l'hygrométrie, car il permet la perte de l'eau dans un œuf, ce qu'il en résulte, une bonne évolution de la poche d'air.

I.8.2. Le taux d'hygrométrie

Le taux d'humidité est l'un des paramètres les plus indispensables dans un incubateur, il a un impact direct sur la poche d'air. Si l'humidité dans l'incubateur est trop basse ou trop élevée, la trappe échouera.

Une humidité insuffisante entraîne :

-la cellule d'air est très grande au moment de l'éclosion ;

- le contenu de l'œuf est très visqueux pour que le poussin se retourne ;
- les membranes sont très difficiles à casser ;

Un excès d'humidité provoque :

- très peu d'eau pour s'évaporer de l'œuf ;
- la cellule d'air est trop petite pour que le poussin puisse l'atteindre pendant l'éclosion.

La poche d'air de l'œuf devrait évoluer au fur et à mesure que l'incubation progresse en raison de l'équilibre entre la température et l'humidité pendant la période de l'incubation. Ci-après, le taux d'humidité que doit assurer l'incubateur.

Du 1^{er} au 19 jours : le taux d'humidité doit être entre 55 et 60%. Ce n'est pas un pourcentage fixe, mais il ajustable en fonction du développement de la chambre à air.

Du 20 au 21^{ème} jours ou plus (dans le cas d'un retard) : le taux d'humidité doit être au minimum à 65%, pour une si simple raison, à cette période se fait l'éclore des œufs, pour cela, la membrane interne de la coquille doit être souple et n'adhère pas au poussin afin de faciliter l'éclore.

Pour notre cas on contrôlera le taux d'humidité en utilisant une plaque à LED chauffantes fixée face à un bac à eau. L'augmentation du taux d'hygrométrie se fait par le contrôle de la plaque à LED chauffantes qui permet l'évaporation de l'eau, par contre sa diminution se fait à l'aide des ventilos d'évacuation.

Le niveau du bac à eau est contrôlé par une pompe qui se charge de le remplir dans le cas où le niveau de l'eau diminue.

I.8.3. Le retournement des œufs

Il se fait d'une façon régulière du 3^{ème} au 19^{ième} jour, 2 fois par jour au minimum, il est à noter qu'il ne faut pas retourner les œufs les 2 premiers jours et les 2 derniers jours avant l'éclosion.

Il existe 3 types d'incubateur classés selon la façon dont les œufs sont retournés :

Incubateur manuel : le retournement se fait d'une façon manuelle.

Incubateur semi-automatique : le retournement se fait automatiquement en appuyant sur un bouton.

Incubateur automatique : le retournement se fait d'une façon automatique sans avoir besoin d'une intervention humaine.

Pour le cas de notre incubateur, le retournement se fera d'une manière complètement autonome en utilisant un mécanisme qu'on en parlera dans la partie du mécanisme de retournement.

I.8.4. L'éclosion

Durant cette étape, il ne faut guère intervenir afin d'aider un poussin à sortir de sa coquille pour éviter d'obtenir un poussin chétif et handicapé, par contre il faut lui laisser le temps de le faire tout seul.

Une fois l'éclosion s'est faite, il faut garder les nouveaux nés sans boire ni manger dans la couveuse à une température de 35°C pendant 24 heures et qu'elle soit bien aérée pour sécher les poussins et vider leur tube digestif.

I.9. Mécanisme de retournement des œufs

Dans la nature, la poule fait bouger son œuf régulièrement, ce mouvement permet à l'embryon de bien se placer afin de pouvoir percer plus facilement la coquille. L'incubateur qu'on va réaliser est automatique, c'est-à-dire qu'il n'aura besoin d'aucune intervention humaine, de ce fait, nous avons pensé à utiliser un mécanisme permettant le retournement automatique des œufs en une fréquence prédéfinie. Selon certaines études qui ont été menées à des œufs de poule, le retournement doit se faire au moins 3 fois par jours, pour cela, nous allons compter sur le module RTC que nous définirons dans le dernier chapitre.

Le mécanisme adopté pour faire retourner les poules se compose d'une tige filetée, portée sur des supports qui joueront le rôle du guide, le plateau sera mis sur des roues glissières pour faciliter son déplacement, et sera fixé sur la tige, la rotation de cette dernière entraînera un

mouvement de translation du plateau. La tige est commandée par un moteur pas à pas que nous allons étudier son principe de fonctionnement ci-après.

I.9.1. Moteur pas à pas

Les moteurs pas à pas occupent une place unique dans le monde du contrôle moteur. Ces moteurs sont couramment utilisés dans les applications de mesure et de contrôle. Les exemples d'applications incluent les imprimantes à jet d'encre, les machines à commande numérique et les pompes volumétriques. Plusieurs caractéristiques communes à tous les moteurs pas à pas les rendent idéalement adaptés à ces types d'applications. Ces caractéristiques sont les suivantes:

- Brushless, les moteurs pas à pas sont brushless. Le collecteur et les balais des moteurs conventionnels sont parmi les composants les plus susceptibles de tomber en panne, et ils créent des arcs électriques indésirables ou dangereux dans certains environnements ;
- Indépendant de la charge, les moteurs pas à pas tournent à une vitesse définie, quelle que soit la charge, à condition qu'elle ne dépasse pas le couple nominal du moteur ;
- Positionnement en boucle ouverte, les moteurs pas à pas se déplacent par incréments ou étapes quantifiés. Tant que le moteur respecte les spécifications de couple, la position de l'arbre est connue à tout moment sans nécessiter de mécanisme de rétroaction ;
- Couple de maintien, les moteurs pas à pas sont capables de maintenir l'arbre à l'arrêt ;
- Excellente réponse au démarrage, à l'arrêt et au retour, cette caractéristique figure parmi les plus importantes qui nous a poussé à choisir ce type de moteur, en présence d'une charge considérable sur le plateau (une capacité de 100 œufs), son couple de démarrage important lui permet de vaincre le couple résistant.

I.9.2. Type de moteur pas à pas

Il existe trois types de moteurs pas à pas : aimant permanent, à réluctance variable et hybride. Les moteurs à aimants permanents ont un rotor aimanté, tandis que les moteurs à réluctance variable ont des rotors dentés en fer doux.

Les moteurs pas à pas hybrides combinent les aspects de la technologie des aimants permanents et de la réluctance variable. Le stator, ou partie fixe du moteur pas à pas, contient plusieurs

enroulements. La disposition de ces enroulements est le facteur principal qui distingue différents types de moteurs pas à pas du point de vue électrique. Du point de vue du système électrique et du système de contrôle, les moteurs à réluctance variable sont éloignés des autres types. Les moteurs à aimants permanents et hybrides peuvent être enroulés en utilisant des enroulements unipolaires, bipolaires ou bifilaires.

Pour notre projet, on utilisera un moteur pas à pas bipolaire dont le principe de fonctionnement sera expliqué ci-après.

I.9.2.1. Moteur pas à pas bipolaire

Les moteurs pas à pas bipolaires sont composés de deux enroulements et de quatre fils. Contrairement aux moteurs unipolaires, les moteurs bipolaires ne disposent pas de point milieu. L'avantage de ne pas avoir de point au centre est que le courant traverse un enroulement entier à la fois au lieu de seulement la moitié de l'enroulement. En conséquence, les moteurs bipolaires produisent plus de couple que les moteurs unipolaires de la même taille. L'inconvénient des moteurs bipolaires par rapport aux moteurs unipolaires est que les moteurs bipolaires nécessitent des circuits de commande plus complexes.

Le courant dans l'enroulement d'un moteur bipolaire est bidirectionnel. Cela nécessite de changer la polarité de chaque extrémité des enroulements. Comme le montre la Figure I.4, le courant circule de gauche à droite dans le bobinage 1 lorsque $1a$ est positif et $1b$ est négatif. Le courant circulera dans le sens opposé lorsque la polarité de chaque extrémité sera inversée. Un circuit de commande, appelé pont en H, est utilisé pour changer la polarité aux extrémités d'un enroulement. Chaque moteur bipolaire ayant deux enroulements, 2 ponts en H sont nécessaires pour commander chaque moteur.

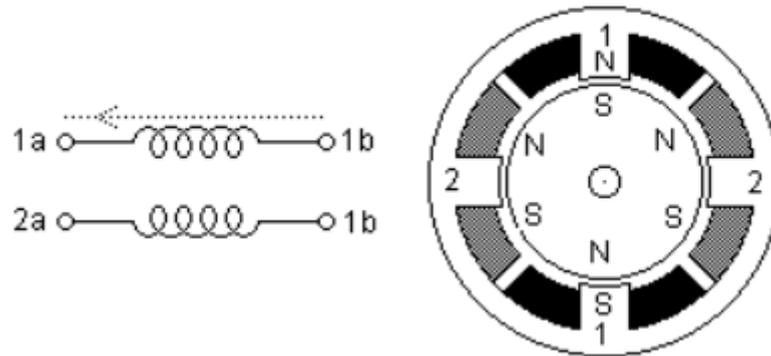


Figure I.4.Sens du courant circulant dans la bobine

La figure I.4 illustre un moteur bipolaire à 30 degrés par étape. L'enroulement 1 du moteur est réparti entre les pôles supérieurs et inférieurs du stator, tandis que l'enroulement 2 du moteur est réparti entre les pôles gauches et droits du stator. Le rotor est un aimant permanent à 6 pôles, 3 sud et 3 nord disposés autour de sa circonférence.

Comme un moteur unipolaire, les moteurs bipolaires peuvent être à pas unique avec deux séquences de commande différentes. En utilisant + et - pour indiquer la polarité de la puissance appliquée à chaque borne du moteur et à 0 pour indiquer qu'aucune puissance n'est appliquée, ces séquences sont présentées dans la figure 1.5 pour un tour ou 12 pas. La première séquence minimise la consommation d'énergie en alimentant un seul enroulement à la fois, tandis que la deuxième séquence optimise le couple en alimentant les deux enroulements alternativement.

Terminal 1a:	+	0	-	0	+	0	-	0	+	0	-	0
Terminal 1b:	-	0	+	0	-	0	+	0	-	0	+	0
Terminal 2a:	0	+	0	-	0	+	0	-	0	+	0	-
Terminal 2b:	0	-	0	+	0	-	0	+	0	-	0	+
					time →							
Terminal 1a:	+	+	-	-	+	+	-	-	+	+	-	-
Terminal 1b:	-	-	+	+	-	-	+	+	-	-	+	+
Terminal 2a:	-	+	+	-	-	+	+	-	-	+	+	-
Terminal 2b:	+	-	-	+	+	-	-	+	+	-	-	+
					time →							

Figure I.5. Polarité de la puissance appliquée sur chaque borne du moteur

La combinaison de ces deux séquences en une seule séquence entraînera un demi-pas du moteur ce qui lui permet de se déplacer par incréments de 15 degrés. La séquence de demi-pas du moteur est montrée à la Figure I. 6 pour un tour ou 24pas.

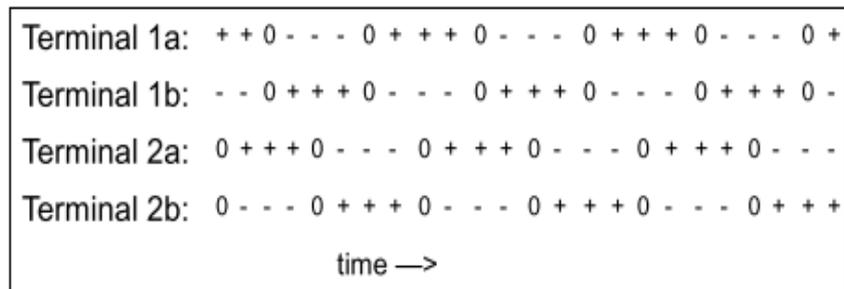


Figure I.6. Séquence du demi-pas d'un moteur pas à pas

I.10. Conclusion

Pour assurer un taux de réussite élevé, il est nécessaire d'avoir des connaissances approfondies dans le domaine de l'incubation artificielle des œufs, plus précisément dans le développement de l'embryon et de l'aviculture. Pour parvenir à un tel résultat, nous nous sommes intéressés, dans ce chapitre, aux œufs de poules, leur structure, le développement de l'embryon, les mécanismes et les périphériques dont un incubateur doit être équipé.

Chapitre II :

Régulation PID

II. Introduction

Les systèmes asservis peuvent présenter des défauts en présence des perturbations : une instabilité, une précision insuffisante, un temps de réponse trop lent, au regard des spécifications du cahier des charges. Il est souvent nécessaire d'incorporer dans la boucle de régulation un élément appelé correcteur (régulateur) en vue d'améliorer les performances de la boucle fermée.

Le régulateur le plus répandu est le régulateur 'Proportionnel, Intégral, Dérivateur' ou PID. Compte tenu de sa simplicité et de sa performance, il est de ce fait utilisé dans divers domaines tels que l'industrie, l'aéronautique, les équipements médicaux et l'automobile. En particulier dans l'automatisation du process PA, dans l'industrie des équipements tels que l'acier, le pétrole, les produits chimiques et l'alimentation, il occupe une place majeure en tant qu'élément indispensable de la technologie d'automatisation.

L'objectif de ce chapitre est de collecter les informations nécessaires pour l'utilisation d'un régulateur, sans oublier de mettre l'accent sur la différence entre les types de régulation et l'impact des différents paramètres du PID sur la réponse du système.

II.1. Définition de la régulation

La régulation est une technique qui permet de maintenir une grandeur physique à une valeur constante quelques soient les perturbations extérieures. L'objectif global de la régulation peut se résumer par ces trois mots clefs : Mesurer, Comparer et Corriger. Nous sommes donc amenés à effectuer des mesures pour obtenir certaines connaissances avant d'entreprendre une action. Ces mesures seront obtenues par l'intermédiaire d'appareillages spécifiques.

II.2. Les types de régulation

Sur beaucoup de machines et d'installations, des grandeurs physiques (température, pression, force, déplacement...etc.) doivent atteindre une valeur déterminée. Pour ce faire, il faut réaliser deux opérations liées l'une à l'autre : la comparaison et l'ajustage. Le cycle d'opérations nécessaires s'effectue dans une boucle de régulation. On peut distinguer deux types de boucles: "boucle de régulation ouverte" et "boucle de régulation fermée".

II.2.1. Régulation en boucle ouverte

Une régulation est dite en boucle ouverte lorsque le système ne comporte pas de rétroaction (feedback), en d'autres termes, la valeur désirée n'est pas comparée à la valeur réelle.

Dans une chaîne de régulation à BO on trouve le processus physique, un capteur se chargeant de la mesure en sortie et un actionneur pour agir sur la grandeur physique, par exemple un radiateur dans lequel l'alimentation en eau chaude et, ainsi, la température, sont réglées ou plus précisément ajustées. En fonction des variations de la température ambiante, on agit manuellement (ouverture/fermeture) sur la vanne. Il s'agit d'une régulation sans boucle de contrôle entre l'entrée et la sortie du système, ni retour d'informations. On entend ainsi par régulation l'utilisation de méthodes et d'équipements spéciaux qui influent sur des cycles opératoires ou des processus.

Un système de régulation est mis en place, si un processus est influencé par une relation à un état désiré sans tenir compte de l'état des mesures. Une caractéristique particulière de la régulation en BO est la séquence ouverte des actions.

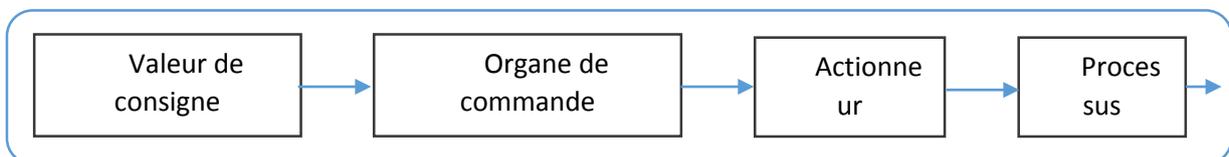


Figure II.1. Principe de régulation en boucle ouverte

II.2.2. Régulation en boucle fermée

Dans une boucle fermée, la valeur souhaitée est constamment comparée à la valeur effective. La norme DIN 19226 définit les termes "réglage et régulation" comme suit : "Le réglage / la régulation" est une opération dans laquelle une grandeur physique (température, pression...etc.) est continuellement mesurée et comparée à une valeur déterminée, dans le but d'aligner ces variables. La séquence fermée des actions qui en résulte s'effectue dans une boucle de régulation fermée. En prenant pour exemple la régulation de la température d'un radiateur, la température réelle est mesurée par un capteur de température et comparée à la consigne. Tout écart entre la valeur désirée et la valeur mesurée provoque un signal qui est transmis à la vanne pour l'ouvrir (si la température est inférieure à la valeur souhaitée) ou la fermer (si la

température est supérieure à la valeur souhaitée). Ainsi, la température est maintenue à la consigne, et ce, de manière stable et indépendamment des conditions extérieures ou d'éléments perturbateurs.

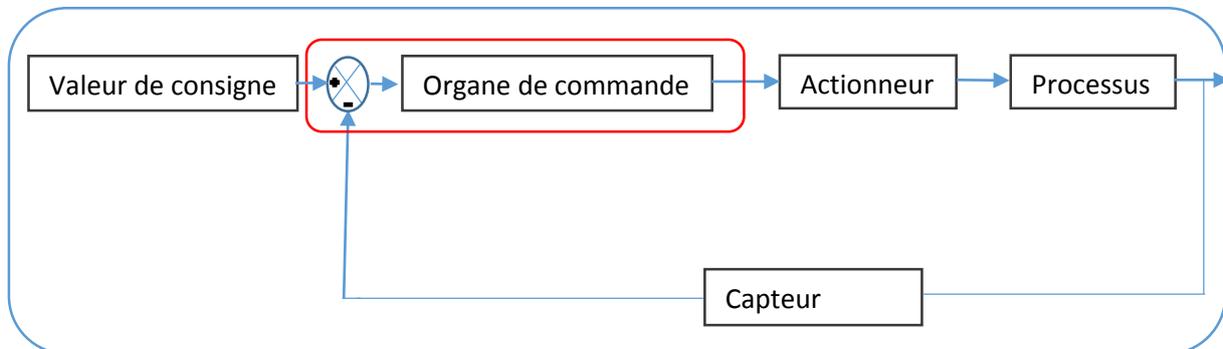


Figure II.2. Principe de régulation en boucle fermée

II.3. Méthode de régulation

Lorsqu'un écart est observé dans un processus, les méthodes de régulation peuvent être distinctes selon le comportement d'un régulateur. Pour les applications en production, il est important de connaître l'influence du temps sur la variable de régulation. Les régulateurs peuvent avoir un comportement continu ou discontinu. On distingue deux types de régulation : à action discontinue (régulation à deux ou plusieurs niveaux) et à action continue (régulation proportionnelle).

II.3.1. Régulation à action discontinue (tout ou rien)

Un processus s'effectuant pas à pas est appelé discontinu. Un régulateur à action discontinue, ou encore appelé régulateur à commutation, influe sur le processus par une commande tout ou rien à un niveau d'énergie constant. Ces régulateurs assurent la fonction de commutation en déclenchant une séquence d'impulsions à un niveau d'énergie fixe. Ces séquences de commutation en circuit et hors circuit ont des temps d'influence variables sur le processus à réguler. Les inconvénients d'un tel système sont les à-coups déclenchés lors de la mise en route. En outre, il n'est pas possible d'éviter les variations de la valeur instantanée autour de la valeur de consigne. Le niveau d'intervalles entre lesquels la grandeur à régler oscille entre l'état "en circuit" et l'état "hors circuit" est appelé amplitude de variation. Cette amplitude est la fonction caractéristique de l'action discontinue. Les régulateurs à trois ou multiples positions disposent

d'au moins une étape intermédiaire supplémentaire par rapport aux régulateurs tout ou rien (appareil de conditionnement d'air, chauffage position neutre - refroidissement).

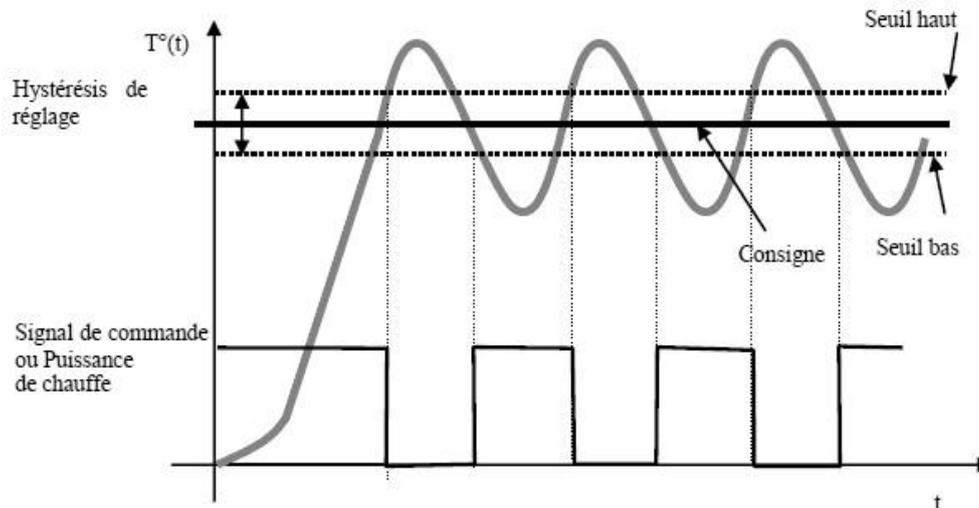


Figure II.3. Graphe de régulation tout ou rien

II.3.2. Régulation à action continue

Les régulateurs à action continue assurent la fonction de régulation en exerçant une influence permanente sur le processus. La valeur de la variable contrôlée peut se situer à n'importe quel niveau de la plage de régulation, ce qui peut générer des signaux entre 0 et 100%. Exemple : Une charge lourde est à accélérer et décélérer en douceur. Dans le cas d'un régulateur à action discontinue, il faut d'abord démarrer la charge à la vitesse V_1 puis à la vitesse V_2 . Celle-ci est ensuite transportée à une vitesse constante V_3 , puis elle décélère aux vitesses V_4 et V_5 (voir Figure III.4). La vitesse est contrôlée en pas à pas. Les paliers individuels des échelons de vitesse sont légèrement nivelés par le débit volumétrique et l'inertie du vérin. Il est toutefois difficile d'en réduire les fluctuations, c.à.d. d'obtenir des échelons plus "lisses". L'une des méthodes pour pallier ces inconvénients est d'utiliser une vanne proportionnelle qui régule en permanence le processus, c.à.d. les vitesses des vérins et des moteurs, évitant ainsi les à-coups des commutations. De plus, il est possible de prédéfinir les vitesses de ces matériels.

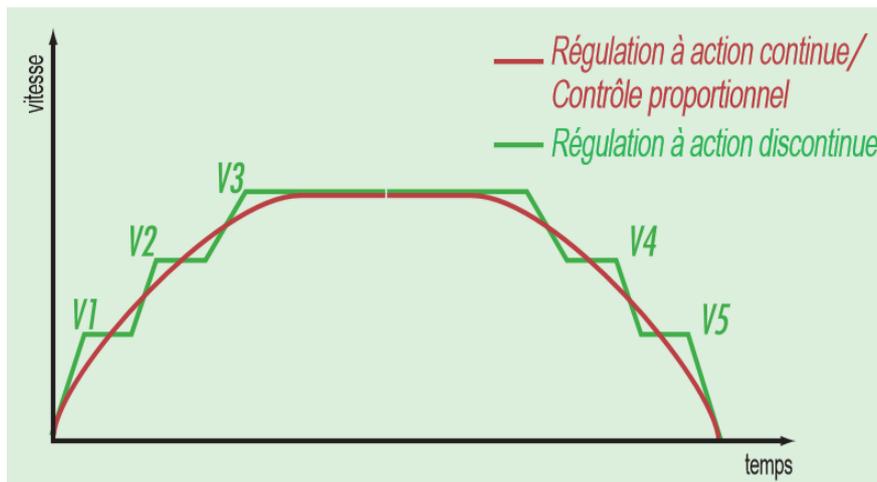


Figure II.4. Régulation à action continue/discontinue

II.4. Régulation PID

C'est l'acronyme de Proportionnel Intégral Dérivée, un régulateur PID est un organe de contrôle permettant d'effectuer une régulation en boucle fermée d'un système industriel. C'est le régulateur le plus utilisé dans l'industrie, il permet de contrôler un grand nombre de procédés. L'erreur observée dans la figure III.4 est la différence entre la consigne et la mesure. Le PID permet 3 actions en fonction de cette erreur :

Une action Proportionnelle : l'erreur est multipliée par un gain K_p .

Une action Intégrale : l'erreur est intégrée sur un intervalle de temps s , puis divisée par un gain T_i .

Une action Dérivée : l'erreur est dérivée suivant un temps s , puis multipliée par un gain T_d .

Ces trois actions seront abordées en détails dans les pages suivantes.

Le régulateur PID classique relie directement le signal de commande $u(t)$ au signal d'écart $e(t)$.

Sa description temporelle est la suivante :

$$u(t) = Kp(e(t) + \frac{1}{Ti} \int_0^t e(t) dt + Td \frac{de(t)}{dt}) \quad (\text{III.1})$$

Avec l'écart défini comme suit :

$$e(t) = w(t) - y(t) \quad (\text{III.2})$$

La fonction de transfert s'écrit :

$$Gc(s) = \frac{U(s)}{E(s)} = Kp(1 + \frac{1}{sTi} + sTd) \quad (\text{III.3})$$

Où Kp, Ti et Td sont les gains du correcteur proportionnel, intégral et dérivée respectivement.

On constate que la variable manipulée est la somme pondérée de l'erreur (P), de l'intégration de l'erreur (I) et de la dérivée de l'erreur (D) ensemble. Une interprétation du PID est qu'il agit selon les erreurs présentes, passées et futures. En effet la partie proportionnelle de la commande se base sur l'erreur du présent, la commande intégrale cumule les erreurs du passé, quant à la dérivée, elle prédit en quelque sorte l'évolution de l'erreur dans le futur.

II.5. Structure d'une boucle de régulation PID

La structure du système de commande est décrite par la Figure III.5

La fonction de transfert du régulateur est notée C(s) et celle du procédé G(s), où S est la variable de Laplace.

$$H_{bo}(s) = C(s)G(s) \quad (\text{III.4})$$

$$H_{bf}(s) = \frac{H_{bo}(s)}{H_{bo}(s) + 1} \quad (\text{III.5})$$

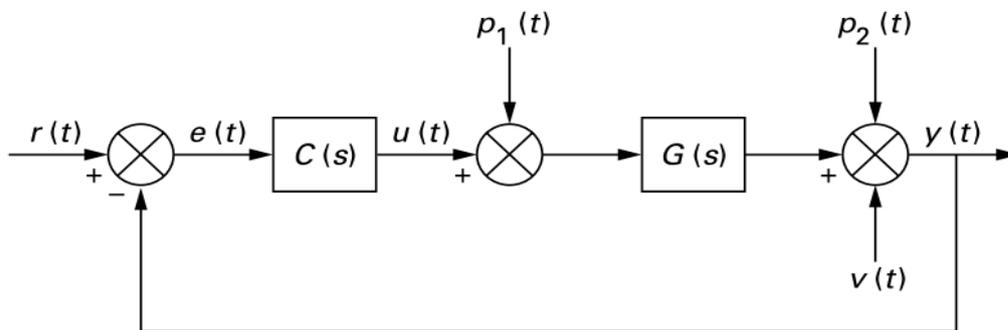


Figure III.5. Configuration d'une boucle de régulation

Avec :

$r(t)$: signal de référence ou consigne (introduit par l'utilisateur ou un autre régulateur),

$e(t)$: erreur (entrée du régulateur).

$$e(t) = r(t) - y(t),$$

$u(t)$: Signal de commande (sortie du régulateur),

$P_1(t), P_2(t)$: perturbations à l'entrée et à la sortie du procédé,

$v(t)$: bruit à la sortie du procédé (par exemple bruit de mesure),

$y(t)$: sortie mesurée du procédé,

$C(s)$: fonction de transfert du régulateur,

$G(s)$: fonction de transfert du procédé.

Il existe 3 architectures possibles pour de combiner les trois effets d'un PID :

-Architecture série.

-Architecture parallèle.

-Architecture mixte (parallèle et série).

II.6. Principe de fonctionnement d'un régulateur PID

Le régulateur PID est un élément indispensable dans la boucle de régulation, en lui introduisant la valeur de l'écart entre la valeur désirée et la valeur réelle, il agit en délivrant le signal de commande adéquat appliqué au système permettant d'atteindre la valeur de consigne.

Dans une boucle de régulation le principal rôle d'un régulateur PID est de minimiser l'écart entre la valeur d'entrée et la valeur de sortie du système, et ce, en choisissant la combinaison d'actions adaptées en fonction d'un cahier des charges préétabli, pour cela, on doit connaître l'influence de chaque type de correction sur la réponse du système.

II.7. L'effet des paramètres d'un régulateur PID

Dans cette partie nous expliquerons clairement l'effet de chaque correcteur sur la réponse du système.

II.7.1. Correcteur proportionnel

Dans le cas de l'utilisation d'un proportionnel, l'erreur est multipliée par une constante K_p .

$$u(t) = K_p \times e(t) \quad (\text{III.6})$$

$$u(p) = K_p \times e(p) \quad (\text{III.7})$$

On utilise le correcteur proportionnel pour augmenter la rapidité de la réponse du système (réduire le temps de montée), mais en contrepartie, l'augmentation de ce dernier engendre une instabilité de la réponse du système.

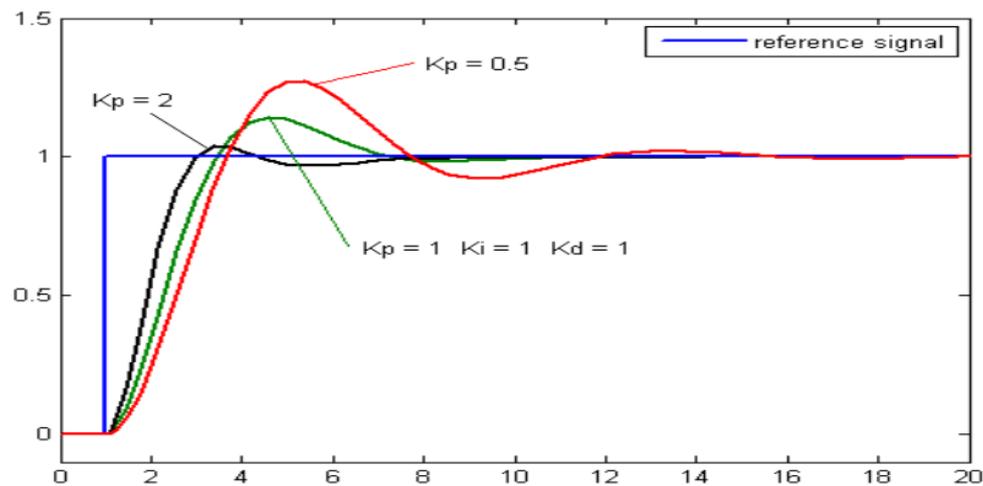


Figure II.6. L'effet du paramètre K_p sur la réponse du système

II.7.2. Correcteur intégral

L'erreur est intégrée sur un intervalle de temps, puis multipliée par une constante K_i .

$$u(t) = K_i \times \int_0^t e(t) dt \quad (\text{III.8})$$

$$u(p) = K_i \times \frac{e(p)}{p} \quad (\text{III.9})$$

Le correcteur intégral élimine l'erreur statique (résiduelle en régime permanent) ce qui augmente la précision et rend le système plus dynamique. Il est utilisé dans le contrôle des procédés dont la variation est bruitée (pression...etc.).

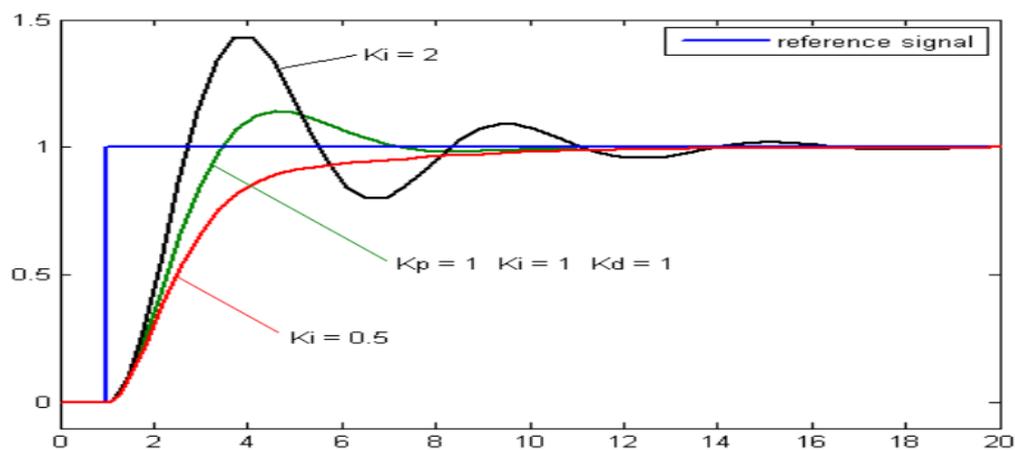


Figure II.7. L'effet du paramètre K_i sur la réponse du système

II.7.3. Correcteur dérivé

L'erreur est dérivée par rapport au temps, puis multipliée par une constante K_d .

$$u(t) = K_d \times \frac{d.e(t)}{d(t)} \quad (\text{III.10})$$

$$u(p) = K_d \times e(p) \times p \quad (\text{III.11})$$

Contrairement au correcteur intégral, le correcteur dérivé est utilisé pour les procédés dont la variation est lente (température...etc.). Il réduit les dépassements et rend le système plus stable.

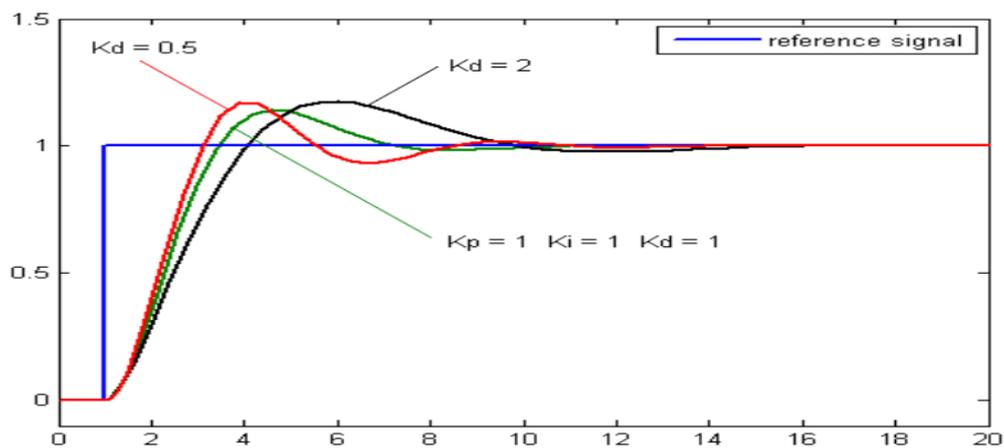


Figure II.8. L'effet du paramètre K_d sur la réponse du système

Le Tableau III-1 montre l'impact de l'utilisation des différents paramètres P, I et D d'un régulateur PID, avant d'entamer le réglage d'un PID, il faut tout d'abord avoir une idée sur l'effet des différents coefficients sur la réponse du système.

Coefficient	Temps de montée	Temps de stabilisation	Dépassement	Erreur statique
K_p	Diminue	Augmente	Augmente	Diminue
K_i	Diminue	Augmente	Augmente	Annule
K_d	-	Diminue	Diminue	-

Tableau 2. Influence des paramètres K_p , K_i et K_d sur la réponse du système

II.8. Réglage d'un PID

Le choix du correcteur (P, PI, I, D, PID) dépend de la nature du procédé et la réponse voulue, régler un correcteur PID revient à déterminer les valeurs des coefficients G, T_i et T_d qui seront appliquées au procédé afin d'obtenir la réponse désirée.

L'objectif principal de la régulation est d'obtenir un système dont les 3 paramètres qu'on citera par la suite sont optimisés, ces paramètres sont comme suit :

La robustesse est le paramètre le plus important, on dit qu'un système est robuste si la régulation fonctionne toujours même si le procédé change d'état (vieillesse mécanique, condition d'environnement extrêmes...etc.).

La précision est liée à l'erreur statique, si on arrive à éliminer ou limiter l'influence de cette dernière sur la réponse du système, on obtiendra un système précis.

La rapidité est un paramètre qui dépend du temps de montée et d'établissement du régime stationnaire de la réponse du système, moins ces deux temps sont réduits ; plus le système est rapide.

Le point le plus difficile dans la conception d'un correcteur PID est de trouver le meilleur compromis, car un régulateur PID parfait n'existe pas, pour cela, il faut fixer en amont un cahier des charges sur lequel il faut travailler pour obtenir les résultats souhaités.

Il existe plusieurs méthodes permettant de calculer les valeurs des coefficients G, T_i et T_d , parmi lesquels on peut citer la méthode de Ziegler nichols que nous allons expliquer dans la partie en-dessous.

II.9. Méthode de Ziegler Nichols

C'est une technique très utilisée pour la détermination des paramètres PID, elle s'applique pour les systèmes dont le type est apériodique, elle offre la possibilité de calculer les paramètres P, I, et D avec deux méthodes différentes que nous allons aborder de la façon suivante.

II.9.1. Méthode de Ziegler-Nichols temporelle

Pour une première approche, on excite le système par un échelon pour avoir en sortie une réponse indicielle, à partir de cette dernière on tire les paramètres mesurés dans la Figure III.9, pour ensuite les remplacer dans le tableau afin de déterminer le réglage des paramètres du régulateur PID.

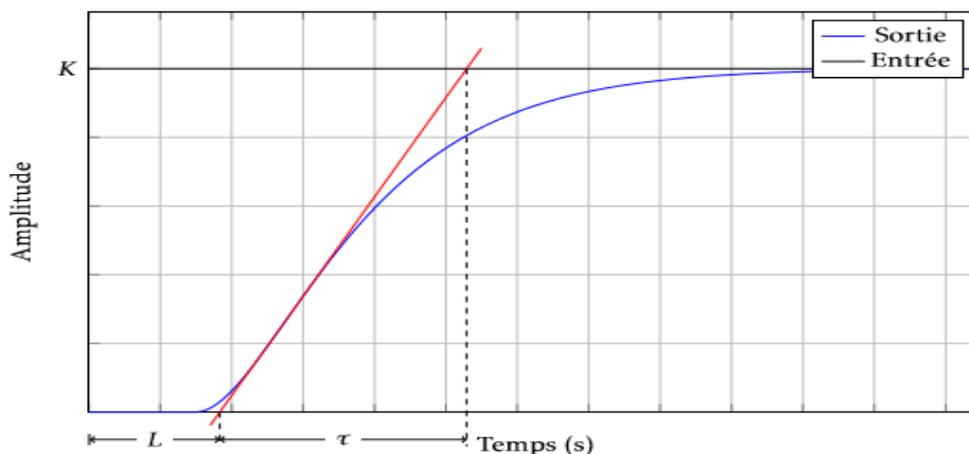


Figure II.9. Réponse indicielle d'un système

On trace la tangente au point d'inflexion, puis on tire du graphe la valeur de la distance commençant de l'origine au point d'intersection de la tangente avec l'axe des abscisses et on la nomme L .

La distance entre les deux points d'intersections de la tangente avec l'axe des abscisses et le gain K correspond à une valeur appelée τ .

On peut alors calculer les coefficients du régulateur choisi à l'aide du tableau suivant :

Paramètre	K_p	T_i	T_d
P	$\frac{\tau}{L}$	∞	0
PI	$0.9 \times \frac{\tau}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \times \frac{\tau}{L}$	$2 \times L$	$0.5 \times L$

Tableau 3. Calcul des paramètres du contrôleur PID

II.9.2. Méthode de Ziegler-Nichols fréquentielle

Dans cette méthode, on détermine le K critique, et ce, en fixant les deux paramètres K_i et K_d à l'infini (∞) et à 0 de suite, en faisant augmenter le paramètre K_p jusqu'au point où le système devient marginalement stable (la réponse commence à présenter des petites oscillations), à partir de ce moment-là, on prend la valeur de $K_p=K_{cr}$, P_{cr} est défini sur la Figure III.10.

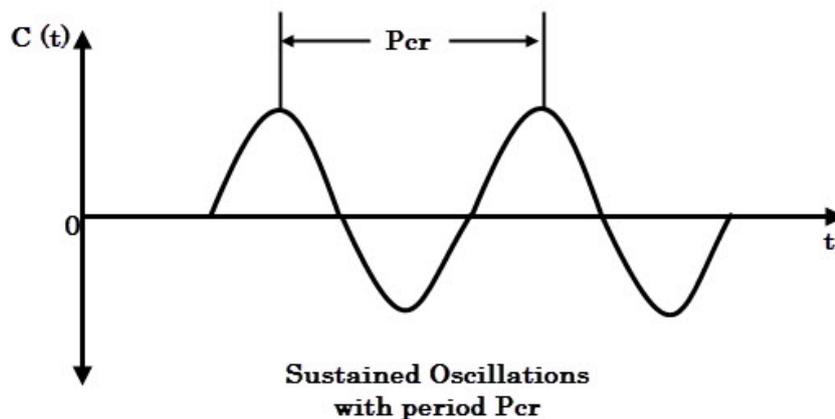


Figure II.10. Détermination de la valeur de P_{cr}

Une fois les deux valeurs de P_{cr} et K_{cr} sont déterminées, il ne nous reste que de les remplacer dans le tableau suivant afin de trouver les paramètres du régulateur.

Paramètre	K_p	T_i	T_d
P	$0.5 \times K_{cr}$	∞	0
PI	$0.45 \times K_{cr}$	$0.8 \times P_{cr}$	0
PID	$0.6 \times K_{cr}$	$0.5 \times P_{cr}$	$0.125 \times P_{cr}$

Tableau 4. Calcul des paramètres du contrôleur PID par la méthode fréquentielle

II.10. Conclusion

L'objectif de ce chapitre était de faire un tour d'horizon sur la régulation PID à partir d'une lecture d'ingénieur de la littérature sur le domaine. Une structure générale d'une boucle de régulation a été utilisée pour ce faire. Les types de régulation ont été élaborés tout en citant des exemples pratiques pour montrer les points de différence.

Les méthodes de régulation, les différents avantages et inconvénients ont également été présentés. Un ensemble de simulation a été exposé pour mieux appréhender l'effet de chaque correcteur sur la sortie du système.

Pour apprendre à paramétrer un régulateur PID, deux méthodes de Ziegler nichols ont été abordées dans le but de maîtriser le réglage des coefficients du régulateur PID.

Pour conclure, un régulateur PID ne pouvant promettre une réponse idéale dont la précision, la rapidité et la robustesse sont exemplaires, il va falloir donc chercher la réponse satisfaisant le cahier des charges. Pour cela on utilise des règles "simples" :

- Augmentation de K_p : montée plus rapide mais plus de dépassement.
- Augmentation de K_i : montée plus rapide mais régime stationnaire plus long, erreur statique plus faible.
- Augmentation de K_d : diminution du dépassement et du temps d'établissement du régime stationnaire, mais augmentation de la sensibilité au bruit.

Chapitre IV :

Conception et réalisation pratique

III. Introduction

Dans ce chapitre nous allons parler de notre réalisation complète, tout en commençant par la conception, pour bien structurer le projet il est nécessaire de le découper en plusieurs tâches, ainsi le projet devient plus facilement gérable et maîtrisable.

Pour une première ébauche, nous allons citer le matériel et les différents composants essentielles à la réalisation du projet, puis vient la partie de la conception qui est une étape très importante dans le cycle de la réalisation d'un projet, elle consiste à tracer le chemin qui mène à la phase finale et permet d'éviter tout imprévu pendant la période de réalisation du projet. Pour cela, nous allons utiliser un logiciel de simulation sur lequel on vérifiera le bon fonctionnement de chaque partie avant d'entamer sa réalisation pratique.

Pour une deuxième ébauche, nous enchaînerons la partie de réalisation pratique où nous allons mettre en œuvre tous les circuits de simulation, les mécanismes que doit équiper notre incubateur un incubateur et finir par une implémentation temps réel de l'algorithme.

III.1. Matériel utilisé

III.1.1. Capteur de température DS18B20 :

C'est un thermomètre numérique qui fournit une mesure dont la résolution est de 9 à 12 bits, il communique sur un bus (1 Wire) qui, par définition, nécessite seulement une ligne de données (et la masse) pour faire la mesure dans une plage allant de -55°C jusqu'à $+125^{\circ}\text{C}$. Il a une précision de $\pm 0.5^{\circ}\text{C}$ dans un intervalle de -10°C à $+85^{\circ}\text{C}$ ce qui rend ce capteur très intéressant pour une utilisation normale. Il est alimenté par tension continue qui varie entre 3.3 et 6v.

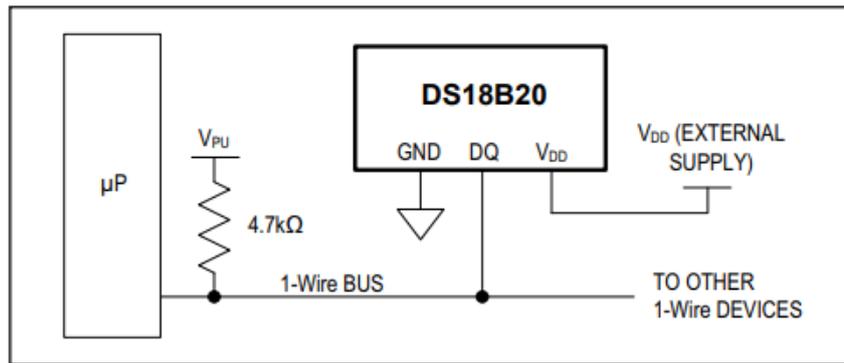


Figure III.1. Circuit de branchement du capteur DS18B20

III.1.2. Capteur d'humidité et de température DHT22

Digital-output relative humidity & temperature sensor/module, appelé AM2302 est un capteur de température et d'humidité numérique capable de mesurer des températures allant de -40°C jusqu'à 125°C avec une précision de $\pm 0.5^{\circ}\text{C}$ et des taux d'humidité relative allant de 0 jusqu'à 100% avec une précision de $\pm 2\%$ ($\pm 5\%$ aux extrêmes, à 10% et 90%).

Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde). Idem pour ce type de capteur sa tension d'alimentation varie entre 3.3 et 6v et, pour une meilleure précision il est recommandé qu'il soit alimenté avec une tension continue de 5v.

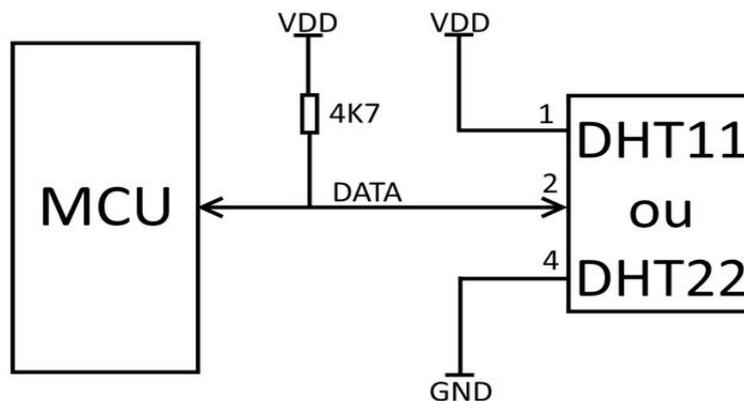


Figure III.2. Circuit de branchement du capteur DHT22

III.1.3. Module RTC (Real Time Clock)

L'horloge temps réel série (RTC) DS1307 est une horloge / calendrier qui fournit les secondes, les minutes, les heures, informations sur le jour, la date, le mois et l'année, toutes ces données sont transférées en série via un bus bidirectionnel I2C.

L'horloge fonctionne en mode 24 heures ou 12 heures. Format heure avec indicateur AM / PM. Le DS1307 a un circuit de détection de puissance intégré qui détecte les coupures de courant et qui lui permet de passer automatiquement à l'alimentation de secours, l'opération de chronométrage se poursuit tant que la pièce fonctionne à partir de l'alimentation de secours.

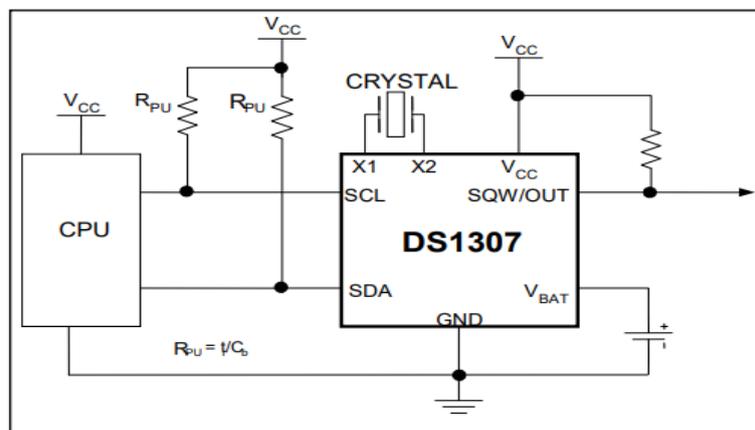


Figure III.3. Circuit de branchement du module RTC

III.1.4. Afficheur LCD I2C

C'est un afficheur à cristaux liquides (Liquid Crystal Display), utilisant le protocole de communication I2C pour afficher les données, dans notre projet, nous allons utiliser 2 afficheurs LCD dont la taille est de 2×16 (2 lignes × 16 colonnes), un LCD 16×2 sera destiné à l'affichage de l'humidité, la date/l'heure et l'état du moteur (right moving & left moving). Un autre LCD d'une taille de 16×2 sera consacré à l'affichage du niveau d'eau se trouvant dans le bac à eau. Un troisième écran LCD d'une taille de 4×20 sera utilisé pour afficher la température et l'effort du régulateur PID.



Figure III.4 Afficheur LCD à base du protocole I2C

III.1.5. Moteur pas à pas

Un moteur pas à pas est une machine qui transforme l'énergie électrique en énergie mécanique, et qui divise une rotation complète en un nombre d'étapes égales. Le couple moteur et la précision sont les deux paramètres les plus importants qui nous ont poussé à opter pour ce type de moteur.

Ce moteur va être utilisé dans un mécanisme permettant de faire translater le plateau afin de retourner les œufs.

III.1.6. Driver TB6600

Le TB6600 est un driver utilisé dans le but de contrôler un moteur pas à pas à deux phases, compatible avec Arduino et d'autres microcontrôleurs qui peuvent émettre un signal d'impulsion numérique 5V. Le conducteur de moteur pas à pas de TB6600 a une entrée de puissance de gamme large, alimentation de 9 ~ 42VDC. Il est capable de produire un courant de crête 4A, ce qui est suffisant pour la plupart des moteurs pas à pas.

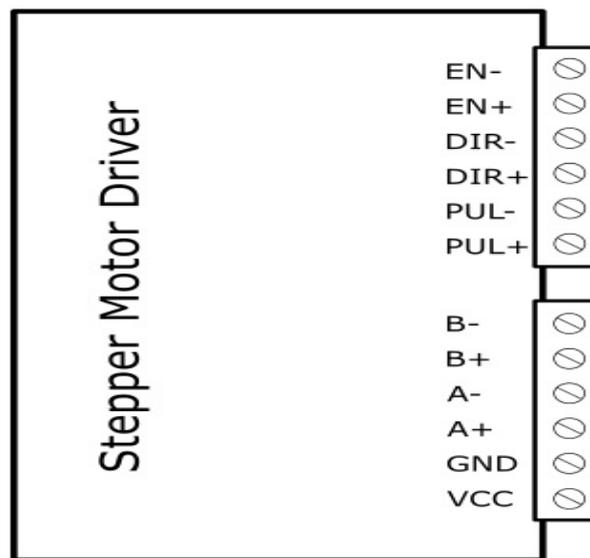


Figure III.5. Driver TB6600

III.1.7. Pompe à eau

L'utilisation d'un bac à eau dans un incubateur complètement autonome nécessite un appareil qui permet de le remplir dans le cas où le niveau d'eau baisse. D'où l'importance de l'utilisation d'une pompe. Le rôle de la pompe est d'aspirer l'eau se trouvant dans un réservoir en dehors de l'incubateur pour le refouler dans le bac à eau à l'intérieur. Cette technique nous assurera la présence d'un certain niveau d'eau dans le bac à eau.

III.1.8. Une carte Arduino

Un Arduino est une carte électronique équipée d'un microcontrôleur programmable qui a pour rôle recevoir des signaux externes/internes, les traiter, et prendre des décisions suivant un programme qu'on écrit et qu'on lui téléverse.

Il existe plusieurs types de cartes Arduino, parmi lesquelles on peut citer Arduino Uno, Yun, Méga, Nano...etc. La carte la plus utilisée et la plus connue est la carte Arduino Uno, la figure II.1 la représente avec ces différents éléments.

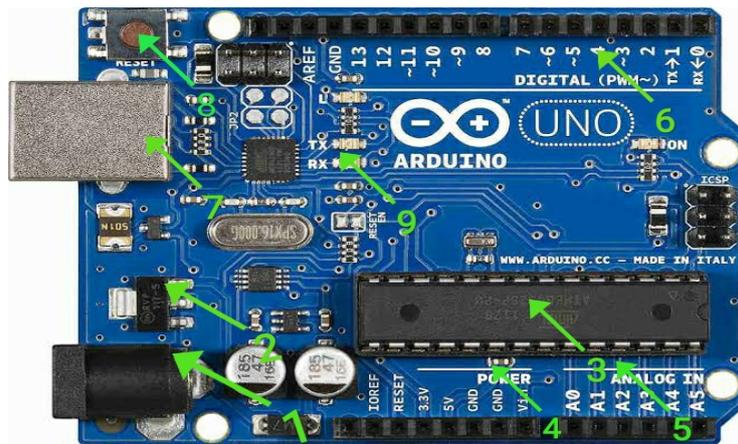


Figure III.6. Description des différents composants d'une carte Arduino

Elément	Rôle
1	Alimenter Arduino par un adaptateur externe, dont la tension continue doit être comprise entre 7v - 12v (recommandée).
2	Régulateur de tension qui a pour tâche, faire stabiliser la tension d'alimentation au tour de 12v.
3	Microcontrôleur, c'est le cerveau au niveau duquel se fait le traitement des données, le μ C utilisé dans Arduino est un AtmelAVR du fabricant Atmel (Atmega328, Atmega32u4...etc.).
4	La Partie puissance.
5	Les différents pins (entrées) Analogiques utilisés pour lire les valeurs analogiques d'un capteur, potentiomètre...etc.
6	Les E/S numériques (tout ou rien), ce sont des pins qu'on peut utiliser pour la PWM (en français, modulation à largeur d'impulsion), les pins Rx et Tx sont responsables de la transmission série.
7	Port USB, il peut être utilisé pour alimenter Arduino ainsi que pour le téléversement du programme.
8	Bouton poussoir qui sert à redémarrer la carte en cas de problème.
9	2 LED qui servent de voyants et qui s'allument pour dire que la carte transmet des données.

Tableau 5. Rôle des différents éléments d'une carte Arduino

Les autres types d'Arduino ont le même principe, la différence se fait au niveau de la taille, un Arduino méga contient plus de pin E/S qu'un Arduino Uno qui quant à lui contient plus de pins E/S par rapport à un Arduino nano. Aussi il faut noter que la EEPROM d'un Arduino diffère d'une carte à une autre.

III.1.8.1. Définition d'une EEPROM

C'est une mémoire morte effaçable électriquement (Electrically Erasable Programmable Read-Only Memory), contrairement aux mémoires volatiles, une mémoire morte est une mémoire dont les données ne se perdent pas en l'absence de l'alimentation électrique.

La taille de la mémoire d'une carte Arduino diffère d'une carte à une autre, la EEPROM d'un :

- Arduino Uno a une taille de 1kB ;
- Arduino méga a une taille de 4 kB ;
- Arduino nano a une taille de 1kB ou 0.512kB (deux versions différentes).

La spécificité de la non perte de données en l'absence de l'alimentation électrique nous sera utile pour la sauvegarde de quelques paramètres nécessaires lors du redémarrage du système d'incubation.

Pour programmer une carte Arduino on a besoin d'un logiciel de programmation.

III.2. Logiciel utilisé

III.2.1. Logiciel de programmation de la carte Arduino IDE

C'est un logiciel sous une licence open source écrit en java et qui se base sur un langage de programmation proche du C, il peut être utilisé avec toutes les cartes Arduino. Pour l'utilisation d'autres modules comme : un écran LCD, capteur de température...etc, on doit inclure une bibliothèque qui définit ce périphérique.

Il intègre une interface d'utilisation facile comme le montre la figure suivante :

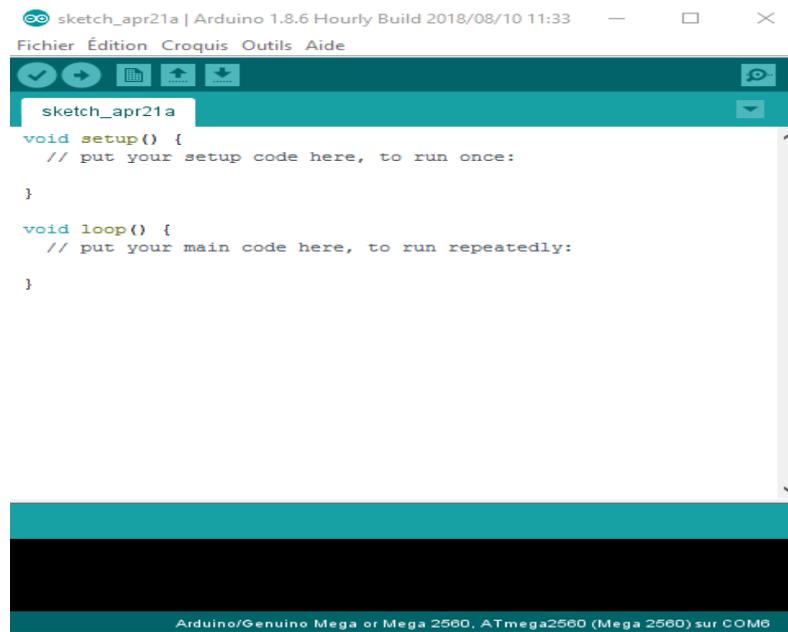


Figure III.7. Interface d'utilisation du logiciel Arduino

Pour notre cas, comme on a tout un processus à contrôler, on a opté de départager les tâches à 3 Arduino qui seront chargés de réaliser les tâches suivantes :

- Arduino UNO : il se charge commander le moteur, de recevoir les informations du capteur dht22, afficher les informations de l'état du moteur, le taux d'humidité, la date et l'heure.
- Arduino nano : il a pour tâche la commande d'une pompe et afficher le niveau de l'eau dans le bac à eau.
- Arduino MEGA : qui aura pour rôle la régulation PID, l'affichage de la température sur un écran LCD ainsi que le contrôle du menu déroulant.

En départageant les tâches ainsi, notre système sera beaucoup moins encombré, aussi en cas de problème, sa détection sera plus facile tant que chaque sous-système a une tâche précise et bien définie.

III.2.2. Eagle

Eagle est un logiciel de conception des circuits imprimés assistée par ordinateur, le nom du logiciel est l'acronyme de " Easily Applicable Graphical Layout Editor ", il offre la possibilité de schématiser un circuit imprimé et de le personnaliser avec une interface facile à utiliser.

Il existe sous forme de deux versions, une version payante illimitée, et une version gratuite limitée à une dimension de 100x80mm pour les cartes à deux couches.

Nous avons utilisé ce logiciel pour faire la conception du schéma électronique d'un Pont H (circuit imprimé) dont le but est de contrôler un moteur à courant continu en direction et en PWM.

Pour une première ébauche, on a utilisé un moteur à courant continu pour le contrôle du mécanisme de retournement des œufs. Le moteur à courant continu a été remplacé par un moteur pas à pas dont la puissance et la précision sont deux paramètres nécessaires pour le déplacement du plateau, malgré cela, nous allons partager avec vous notre expérience avec l'utilisation des circuits imprimés.

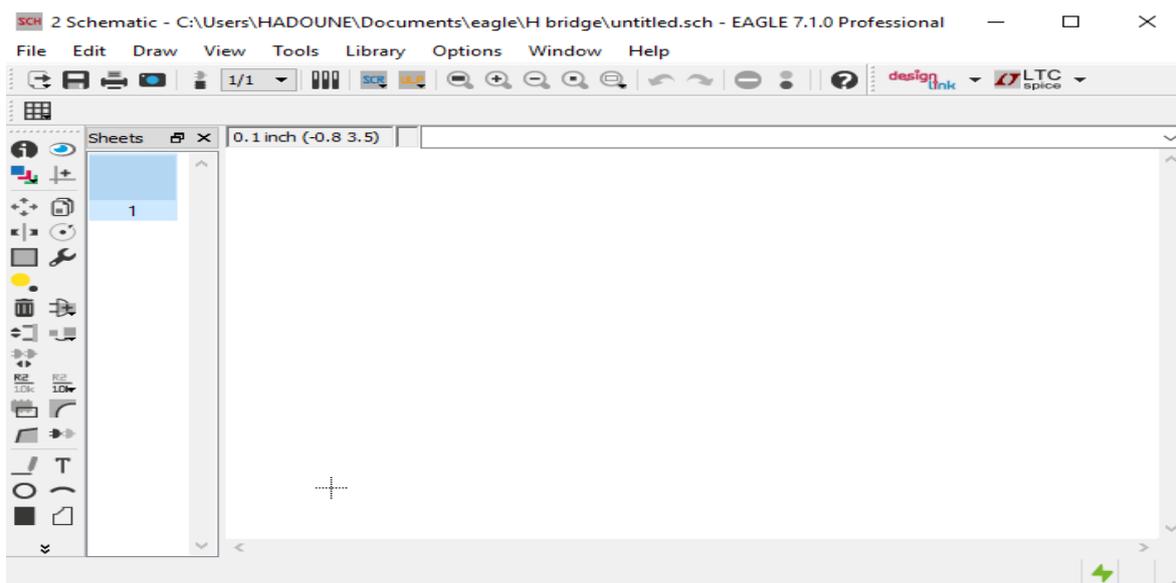


Figure III.8. Interface d'utilisation du logiciel Eagle

III.2.3. ISIS Proteus

C'est un logiciel de conception, test et de mise en page des circuits imprimés. Il associe une facilité d'utilisation et des fonctionnalités diverses permettant une bonne simulation des PCB.

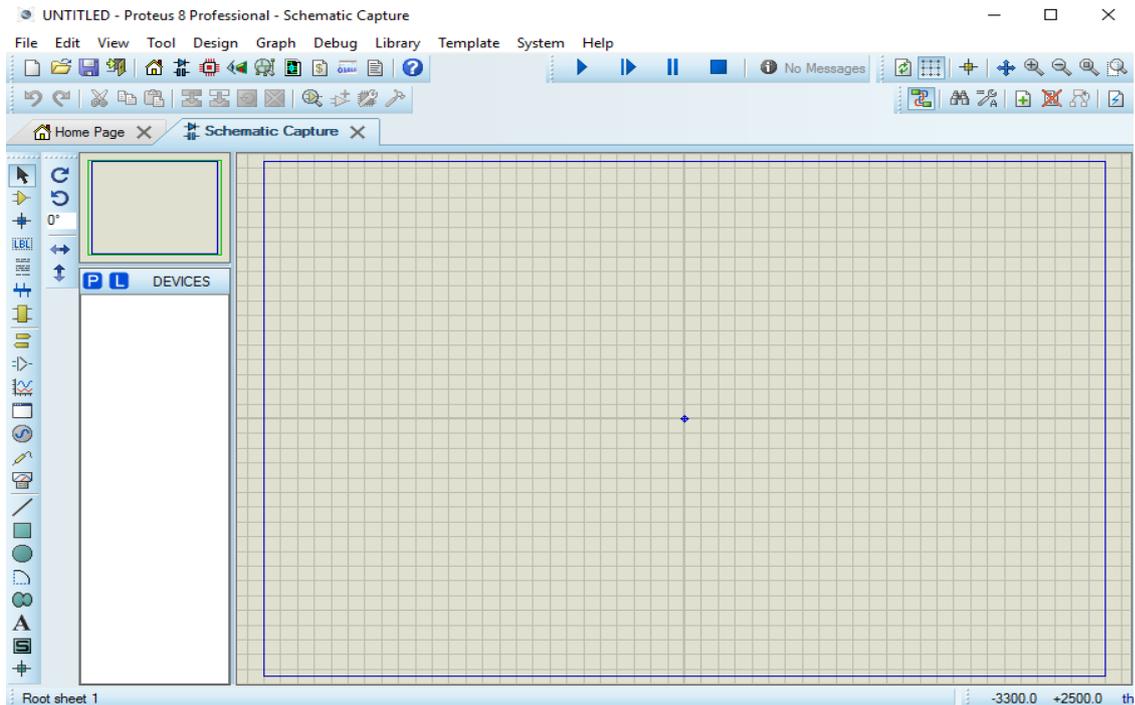


Figure III.9. Interface d'utilisation du logiciel Isis Proteus

III.3. Conception

III.3.1. Conception d'un clavier à bouton poussoir

En utilisant des boutons poussoir des résistances, nous avons pu simuler le fonctionnement d'un clavier qui nous offre la possibilité de contrôler un menu déroulant, paramétrer le régulateur PID, et introduire la température et le taux d'humidité de consigne.

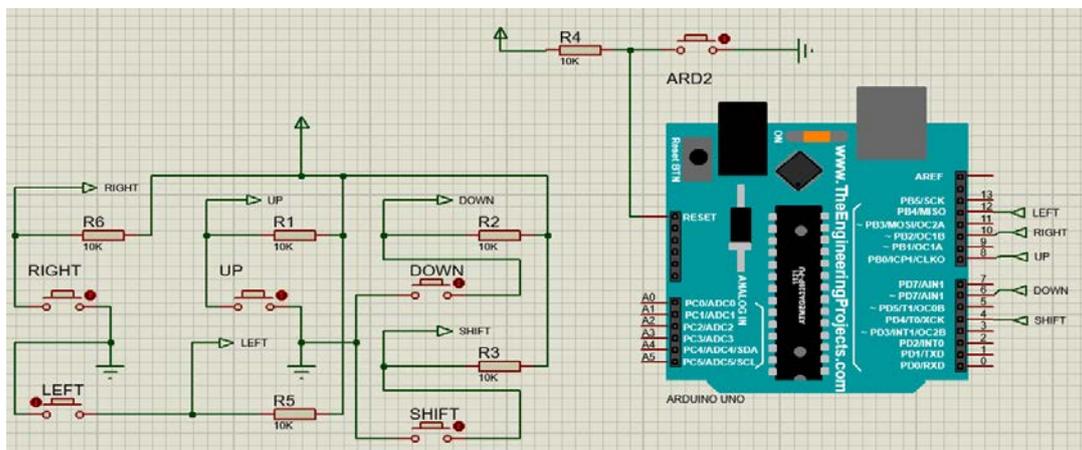


Figure III.10. Circuit de simulation d'un clavier pour un menu déroulant

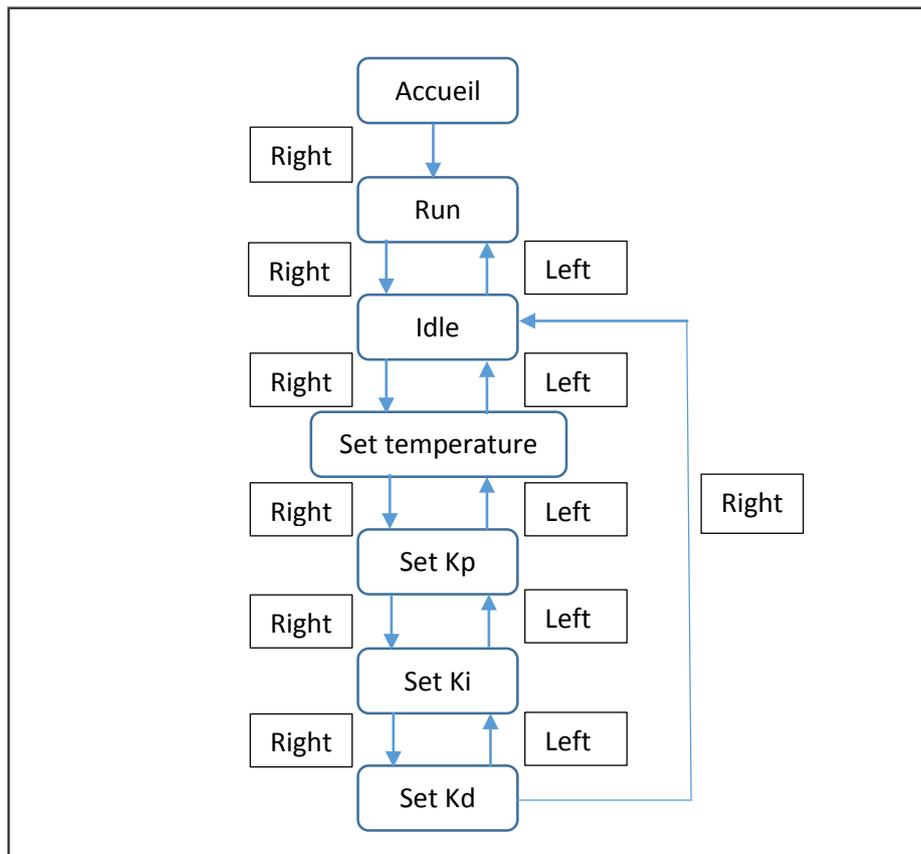


Figure III.11 Schéma synoptique du menu déroulant

III.3.2. Conception d'un système de régulation de température :

Le système se chargeant de réguler la température se compose de deux relais dont l'objectif est de commander un ventilateur et une plaque chauffante. La commande des relais se fait par Arduino et est proportionnelle à l'effort fourni par le régulateur PID.

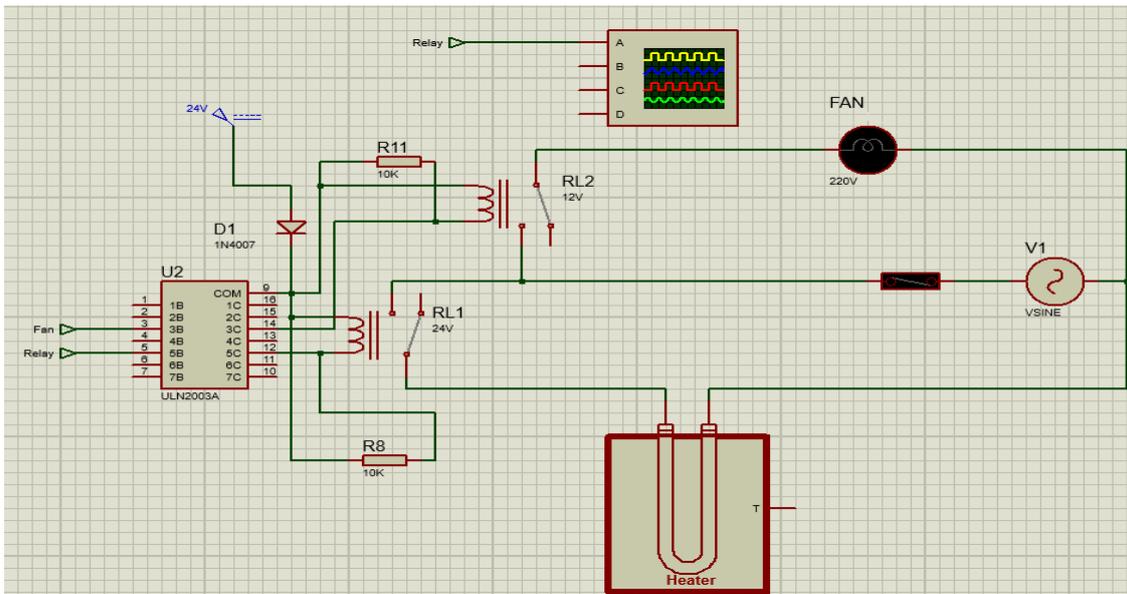


Figure III.12. Circuit de simulation du système de régulation

III.3.3. Conception du circuit électronique d'un pont H

C'est une association de quatre commutateurs réalisant la forme d'une lettre H, ce circuit est utilisé pour inverser le sens du courant traversant généralement un moteur.

Dans cette partie, nous allons aborder les étapes que nous avons suivi pour la réalisation d'un circuit imprimé afin de l'utiliser pour le contrôle du moteur à courant continu (ce moteur à courant continu a été remplacé plus tard par un moteur pas à pas).

1^{ère} étape

On utilise la bibliothèque intégrée initialement au logiciel pour insérer les différents composants électroniques, les placer, et les connecter entre eux.

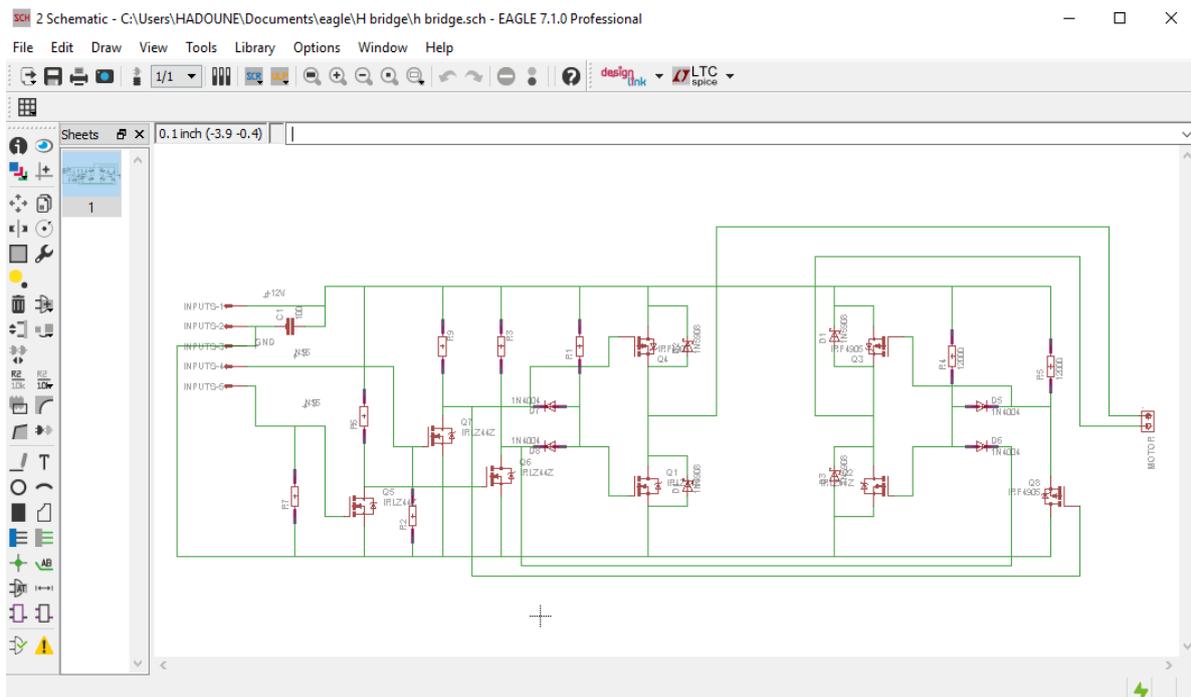


Figure III.13 Schéma électrique d'un pont en H

2^{ème} étape

On passe à la version imprimée du circuit électronique, où on replace les différents composants à notre façon pour les router et finir par l'imprimer avec un papier dédié à ce type d'opérations.

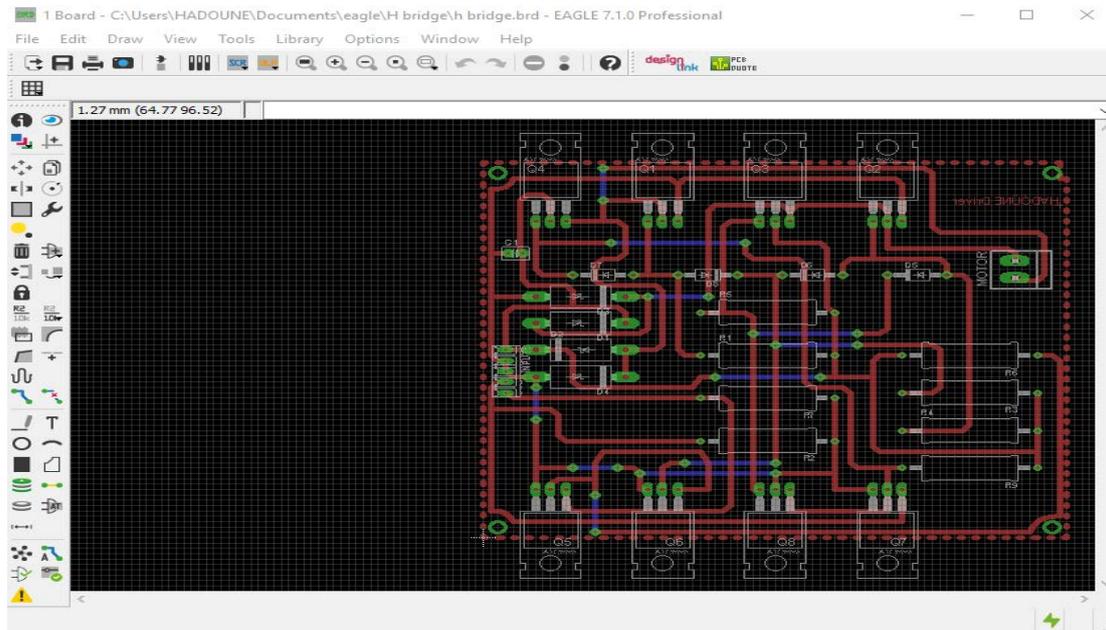


Figure III.14. Routage du circuit électronique (version imprimée)

Maintenant, la partie conception du circuit imprimé se termine et, dans les pages suivantes commence la partie pratique dans laquelle on enchainera avec le reste des étapes.

III.4. Réalisation Pratique

III.4.1. Réalisation du circuit imprimé

Nous vous conseillons d'utiliser des gants afin de vous protéger contre les effets potentiels du NaOH et d'hyper chlorure de fer. Cette partie se compose de 3 étapes qu'on expliquera comme suit :

1^{er} étape

Dépôt de la feuille imprimée sur la carte électronique pour les faire passer sur les rayons ultraviolets.

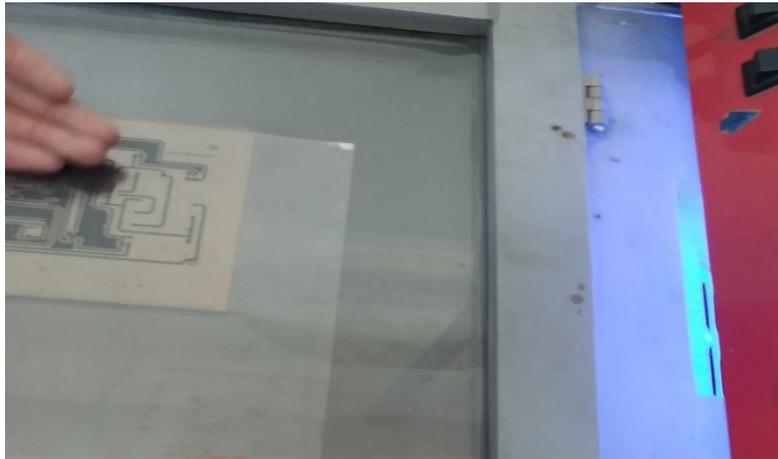


Figure III.15. Passage de la carte sur les rayons ultra-violets

2^{ème} étape

On fait plonger la plaque dans une solution de Na-OH. On renforce les endroits les moins foncés avec un feutre permanent pour obtenir le circuit suivant.

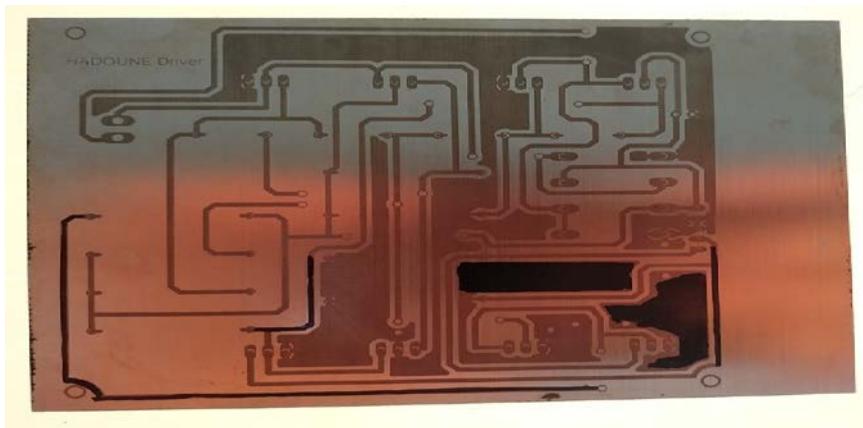


Figure III.16. Immersion de la carte électronique dans un bac de Na-OH

3^{ème} étape

Dans cette dernière étape, on fait passer la plaque dans un bac de perchlorure de fer pour une période de 10 minutes, une fois le temps s'est écoulé, on retire notre plaque et on y jette une petite quantité d'eau pour obtenir une carte prête à utiliser.

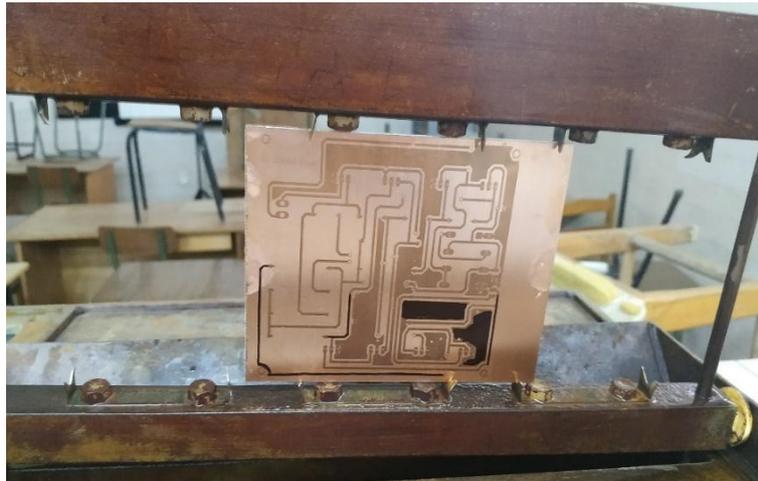


Figure III.17. Passage de la plaque dans un bac de perchlorure de fer

Une fois le processus est terminé, il ne nous reste que de percer l'emplacement des composants électroniques, les placer et les souder.

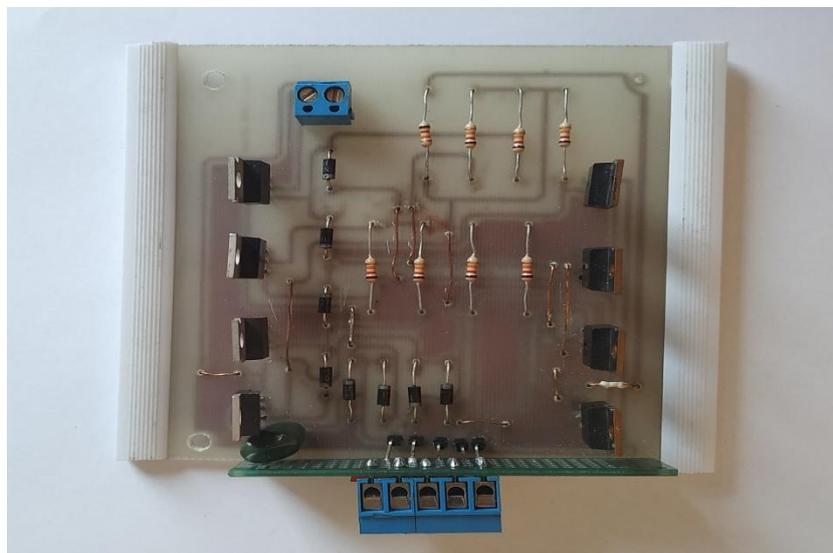


Figure III.18. Réalisation du circuit imprimé

III.4.2. Réalisation du mécanisme du retournement des œufs

Pour aboutir à réaliser un mécanisme de retournement des œufs, nous avons été confrontés au problème du poids (une centaine d'œufs), pour cela, on a opté pour des roulettes dont le rôle

est de porter le plateau (libérer le moteur de la charge), le reste de la conception mécanique se chargera de faire translater le plateau (principe d'une visse sans fin).

Nous avons calculé la distance nécessaire pour un retournement complet d'un œuf dans le but de fixer les limites switch, qui ont pour rôle limiter la translation du plateau dans une distance de 8 cm, cette limitation permettra aux œufs de se retourner à 180° à partir leur position initiale.



Figure III.19 Mécanisme de retournement utilisant les limites switch

III.4.3. Réalisation d'un clavier à bouton poussoir

La manipulation d'un menu déroulant se fait par un clavier à bouton poussoir, dont le rôle est d'introduire les différents paramètres du régulateur PID ainsi que les valeurs de consigne pour la température et le taux d'humidité.

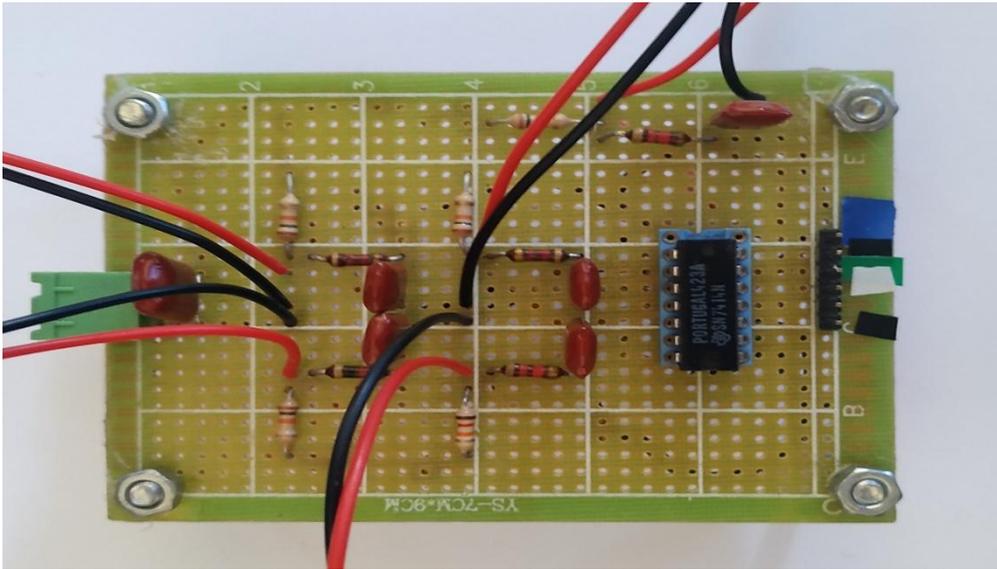


Figure III.20. Clavier pour la commande du menu déroulant

III.4.4. Réalisation du système de régulation de température/humidité

Pour l'alimentation des différentes cartes Arduino, on a eu recours à des régulateurs dont les références sont comme suit : 7805, 7809 et 7812. Chacun des trois régulateurs se charge d'alimenter le matériel suivant : microcontrôleurs, moteur, capteur et écran. Cette carte permet également la commande des ventilos et de la plaque chauffante, elle représente la carte mère de notre projet.

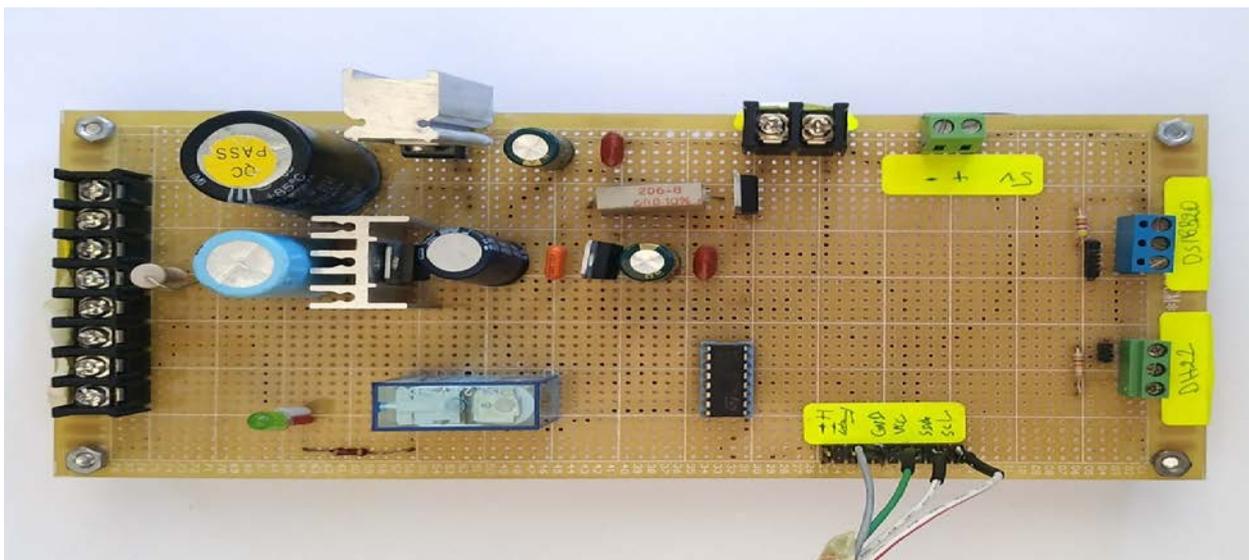


Figure III.21. Carte de commande

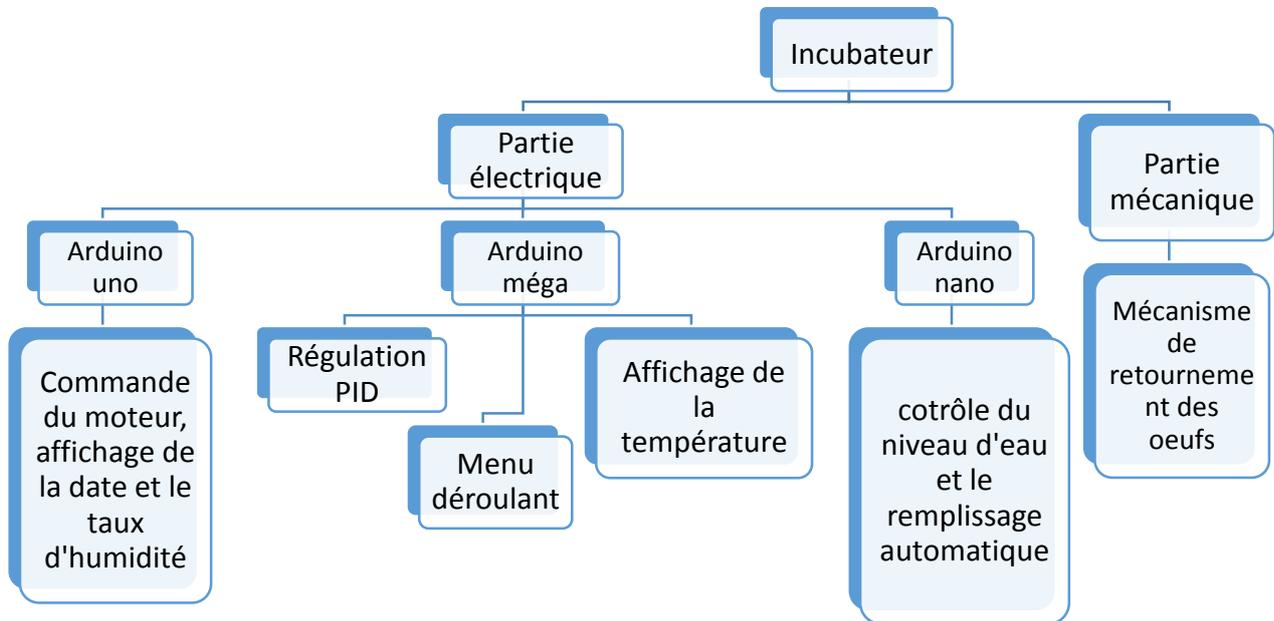


Figure III.23. Schéma synoptique récapitulant le projet complet

III.5. Conclusion

Dans ce chapitre, nous avons commencé par citer le matériel et les logiciels utilisés pour la réalisation du projet et puis nous avons passé à la partie conception et simulation avant de finir par la partie de réalisation. En fait, ce chapitre reflète l'effort fourni dans la partie pratique du projet, et plusieurs sont les connaissances qu'on a pu développer et qu'on peut les énumérer comme suit :

- La conception d'un circuit imprimé, le réaliser en suivant plusieurs étapes qui ont été minutieusement expliquées dans la troisième et la quatrième partie du chapitre ;
- Se familiariser avec l'environnement de programmation Arduino ;
- Enrichir nos connaissances en mécanique et surtout en électronique.

Conclusion Générale :

Le but de ce projet étant la conception et la réalisation d'un système d'incubation spécialisé dans la production des œufs de poule. Nous avons pu atteindre cet objectif et nombreuses sont les connaissances qu'on a pu mettre en pratique à travers ce travail, partant du domaine de l'aviculture et arrivant à la conception d'un système mécanique et tout en passant par les connaissances acquises durant le parcours universitaire plus précisément dans le domaine de l'électronique et de l'automatique.

Pour pouvoir mener à bien ce projet, nous avons développé nos compétences en automatique, en électronique et en mécanique. Ces compétences ont été requises pour mettre en place un mécanisme permettant de retourner les œufs tout en limitant l'influence des vibrations lors de la translation du plateau. Nous ne sommes pas limités au domaine technique mais nous l'avons dépassé en enrichissant nos connaissances dans le domaine de l'aviculture, le développement de l'embryon et son processus de croissance.

Pour finir, la conception et la réalisation d'un système d'incubation complètement autonome nous a apporté une véritable plus-value pluridisciplinaire. Ce travail peut être amélioré en remplaçant le régulateur PID par un régulateur à logique floue qui est souvent plus précis et plus performant qu'un système de régulation classique.

Bibliographie

- 1) *Poulesetcie*. (2017, 06 22). Récupéré sur <https://poulesetcie.com/>
- 2) *Couveuse oeuf -Incubation artificielle des oeufs de poule*. (2018, 07 24). Récupéré sur ovo-site: <https://www.ovo-site.net/incubation-artificielle-oeufs-poule/>
- 3) Cat, A. K. (2011, 11 19). *Technique pour savoir si un oeuf est fécondé*. Récupéré sur techniques d'élevage: <http://www.techniquesdelevage.fr/article-technique-pour-savoir-si-un-oeuf-est-feconde-83445951.html>
- 4) *Développement d'un embryon de poulet*. (2016, 11 30). Récupéré sur <https://www.futura-sciences.com/planete/actualites/zoologie-16-h-developpement-poussin-oeuf-vu-interieur-52137/>
- 5) *Arduino Uno Rev3*. (s.d.). Récupéré sur store.arduino: <https://store.arduino.cc/arduino-uno-rev3>
- 6) Camille, D. (s.d.). Introduction au bus I2C. Metz, Lorraine, France.
- 7) *cours-systèmes-embarqués-Bus-I2C*. (2017, 11 18). Récupéré sur technologuepro: <http://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-Bus-I2C.htm>
- 8) diou, C. (s.d.). Introduction au bus I2C. Metz, Lorraine, France. Récupéré sur technologuerpo: https://les-electroniciens.com/sites/default/files/cours/cours_i2c.pdf
- 9) JARNO, A. (s.d.). *i2c*. Récupéré sur aurel32: <https://www.aurel32.net/elec/i2c.php>
- 10) Dr. Douglas , & W. Jones. (2004). *Stepping Motors Fundamentals / Microchip Technology Inc*. Récupéré sur <https://homepage.divms.uiowa.edu/~jones/step/an907a.pdf>
- 11) Mickaël , C., Etienne , D., & Daniel , R. (2010, 02 26). Régulation par PID. Paris, Paris, France.
- 12) Ould-Bouamama, B. (1998, 03 19). *Régulation Automatique, Actions de régulateurs*. Récupéré sur polytech-lille: <http://www.polytech-lille.fr/cours-regulation-automatique/tr425.htm>
- 13) RACHED, D. (2014/2015). COURS ET EXERCICES DE. Oran, Oran, Algérie.
- 14) *RegulateurPID*. (s.d.). Récupéré sur université de Cadi Ayyad Merrakech: www1.ucam.ac.ma/cneree/RegulateurPID.pdf

- 15) F.Mudry. (2006). "*NotesApplications*" dont le titre est : *Ajustage des Paramètres d'un Régulateur PID*. Récupéré sur "freddy.mudry" de l'école d'ingénieur du Canton de Vaud: freddy.mudry.org/public/NotesApplications/NAPidAj_06.pdf
- 16) Tehrani, K. A., & Mpanda, A. (2012). *PID Control Theory, Introduction to PID Controllers Theory, Tuning and Application to Frontier Areas*. Nancy, Amiens, Lorraine, France.
- 17) Alina BESANÇON-VODA, & Sylviane , G. (1999). *Régulateurs PID analogiques et numériques*. Grenoble, Paris.
- 18) *Index of/TP*. (2012, 09 12). Récupéré sur specialautom:
http://www.specialautom.net/TP/Dimensionnement-de_PID_par_les_methodes_empiriques.pdf
- 19) *eagle*. (s.d.). Récupéré sur elektronique: <http://www.elektronique.fr/logiciels/eagle.php>
- 20) Liu, T. (s.d.). *Digital-output relative humidity & temperature sensor/module DHT22*. Récupéré sur sparkfun:
<https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- 21) datasheets.maximintegrated. (2015). *DS1307*. Récupéré sur [datasheets.maximintegrated](https://datasheets.maximintegrated.com/en/ds/DS1307.pdf):
<https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>
- 22) ada, l. (2018, 11 22). *i2c-spi-lcd-backpack*. Récupéré sur adafruit: <https://cdn-learn.adafruit.com/downloads/pdf/i2c-spi-lcd-backpack.pdf>
- 23) maximintegrated. (2015). *DS18S20 High-Precision 1-Wire Digital Thermometer*. Retrieved from [datasheets.maximintegrated](https://datasheets.maximintegrated.com/en/ds/DS18S20.pdf):
<https://datasheets.maximintegrated.com/en/ds/DS18S20.pdf>
- 24) TamiaLab. (2016). *Utiliser un capteur de température et d'humidité DHT11 / DHT22 avec une carte Arduino / Genuino*. Récupéré sur carnetdumaker:
<https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>
- 25) abra-electronics. (s.d.). *DRI0043 TB6600 Pilote de moteur pas à pas*. Récupéré sur abra-electronics: <https://abra-electronics.com/electromechanical/motors/motor-controllers/dri0043-tb6600-stepper-motor-driver.html?sl=fr>

Annexe :**Programmes utilisés :**

Pour la commande du moteur, l'affichage de la date/heure et le taux d'humidité :

```
#include "Wire.h"

#include <TroykaDHT.h>

#include <LiquidCrystal_I2C.h>

#define DS1307_I2C_ADDRESS 0x68 // the I2C address of Tiny RTC

#define DHTPIN 7 // what pin we're connected to

#define DHTTYPE DHT22 // DHT 22 (AM2302)

#define StepPin 4

#define DirPin 5

#define LISR 8

#define LISL 9

float humd; //Stores humidity value

float tempt; //Stores temperature value

unsigned int ticks=0;

unsigned int Cnt=0;

unsigned int speed_rate =3000;//microseconds

unsigned int pulse =10; //microseconds

bool flag=false;

byte second, minute, hour, dayOfWeek, dayOfMonth, month, year;

// Convert normal decimal numbers to binary coded decimal

DHT dht(DHTPIN, DHTTYPE); /// Initialize DHT sensor

LiquidCrystal_I2C lcd(0x27,16,2);
```

```
uint8_t clock[8]={0x0,0xe,0x15,0x17,0x11,0xe,0x0};
```

```
byte decToBcd(byte val)
```

```
{
```

```
return ( (val/10*16) + (val% 10) );
```

```
}
```

```
// Convert binary coded decimal to normal decimal numbers
```

```
byte bcdToDec(byte val)
```

```
{
```

```
return ( (val/16*10) + (val% 16) );
```

```
}
```

```
// Function to set the currnt time, change the second&minute&hour to the right time
```

```
void setDateDs1307(){
```

```
second =45; minute = 45; hour = 15; dayOfWeek = 5; dayOfMonth =13; month =6; year= 19;
```

```
Wire.beginTransmission(DS1307_I2C_ADDRESS);
```

```
Wire.write(decToBcd(0));
```

```
Wire.write(decToBcd(second)); // 0 to bit 7 starts the clock
```

```
Wire.write(decToBcd(minute));
```

```
Wire.write(decToBcd(hour)); // If you want 12 hour am/pm you need to set
```

```
// bit 6 (also need to change readDateDs1307)
```

```
Wire.write(decToBcd(dayOfWeek));
```

```
Wire.write(decToBcd(dayOfMonth));
```

```
Wire.write(decToBcd(month));
```

```
Wire.write(decToBcd(year));
```

```
Wire.endTransmission();

}

// Function to gets the date and time from the ds1307 and prints result

void getDateDs1307()

{

// Reset the register pointer

Wire.beginTransmission(DS1307_I2C_ADDRESS);

Wire.write(decToBcd(0));

Wire.endTransmission();

Wire.requestFrom(DS1307_I2C_ADDRESS, 7);

second = bcdToDec(Wire.read() & 0x7f);

minute = bcdToDec(Wire.read());

hour = bcdToDec(Wire.read() & 0x3f); // Need to change this if 12 hour am/pm

dayOfWeek = bcdToDec(Wire.read());

dayOfMonth = bcdToDec(Wire.read());

month = bcdToDec(Wire.read());

year = bcdToDec(Wire.read());

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Time : ");

lcd.print(hour, DEC);

lcd.print(":");

lcd.print(minute, DEC);

lcd.print(":");
```

```
lcd.print(second, DEC);

lcd.setCursor(0,1);

lcd.print("Date : ");

lcd.print(month, DEC);

lcd.print("/");

lcd.print(dayOfMonth, DEC);

lcd.print("/");

lcd.print(year,DEC);

lcd.print(" ");

delay(1000);

}

void setup() {

Wire.begin();

Serial.begin(9600);

dht.begin();

pinMode(DHTPIN ,INPUT);

pinMode(StepPin ,OUTPUT);

pinMode(DirPin ,OUTPUT);

pinMode(LISR ,INPUT_PULLUP);

pinMode(LISL ,INPUT_PULLUP);

setDateDs1307(); //Set current time;

setSQW(0x10);// freq=1Hz

lcd.init();

lcd.backlight();
```

```
lcd.print("Initialising .....");

lcd.createChar(2,clock);

attachInterrupt(0,handleInt,FALLING);

}

void setSQW(uint8_t value){

    Wire.beginTransmission(DS1307_I2C_ADDRESS);

    Wire.write(0x7);

    Wire.write(value);

    Wire.endTransmission();

}

void handleInt(){

    ticks++;

    Cnt++;

}

void DH22_display(){

    lcd.clear();

    lcd.setCursor(0,0);

    dht.read();

    switch(dht.getState()) {

    case DHT_OK:

        lcd.print("Temperature = ");

        lcd.print(dht.getTemperatureC());

        lcd.println(" C \t");

        lcd.setCursor(0,1);
```

```
    lcd.print("Humidity = ");  
  
    lcd.print(dht.getHumidity());  
  
    lcd.println(" %");  
  
    break;  
  
case DHT_ERROR_CHECKSUM:  
  
    lcd.println("Checksum error");  
  
    break;  
  
case DHT_ERROR_TIMEOUT:  
  
    lcd.println("Time out error");  
  
    break;  
  
case DHT_ERROR_NO_REPLY:  
  
    lcd.println("Sensor not connected");  
  
    break;  
  
}  
  
delay(3000); //Delay 2 sec.  
  
}  
  
void Motor_state_display(){  
  
    lcd.clear();  
  
    lcd.setCursor(0,0);  
  
    if(digitalRead(LISR)==LOW){  
  
        digitalWrite(DirPin,HIGH);  
  
        lcd.print("RIGHT MOVING ->");  
  
        while(digitalRead(LISL)!=0){  
  
            digitalWrite(StepPin,HIGH);delayMicroseconds(pulse);
```

```

digitalWrite(StepPin,LOW);delayMicroseconds(speed_rate-pulse);
}
if(digitalRead(LISL==LOW)) {return;}
}
if(digitalRead(LISL)==LOW){
digitalWrite(DirPin,LOW);
lcd.print("LEFT MOVING ->");
while(digitalRead(LISR)!=0){
digitalWrite(StepPin,HIGH);delayMicroseconds(pulse);
digitalWrite(StepPin,LOW);delayMicroseconds(speed_rate-pulse);
}
if(digitalRead(LISR==LOW)) {return;}
}
}
void loop()
{
delay(2000);
getDateDs1307();//get the time data from tiny RTC
Serial.print("Ticks= ");Serial.println(ticks);
Serial.print("Cnt= ");Serial.println(Cnt);
if(ticks >=15){DH22_display(); ticks=0;}
if(Cnt>=60){ Motor_state_display();Cnt=0; delay(200);}
}

```