REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA



Faculté de **T**echnologie

Mémoire de fin d'études de Master

Filière : Electronique.

Spécialité : Electronique des Systèmes Embarqués.

Etude et Réalisation d'un Contrôleur d'Incubateur d'œufs.

Présenté par :

BELLABD Yasmina

TCHIKOU Leila

Proposé et Dirigé par :

M. TOUBAL Abdelmoughni

Promotion 2019


بسم الله الرحمن الرحيم

الملخص:

الحاضنة هي جهاز للحضانة الصناعية للبيض من أجل الحصول على كميات كبيرة من الكتاكيت . الغرض من هذا المشروع هو دراسة وتحقيق وحدة تحكم لحاضنة البيض، يتمثل دورها في تنظيم درجة الحرارة والرطوبة والتحكم في التقليب الآلي البيض .لقد صممنا لوحة تحكم إلكترونية قائمة على متحكم Atmega328 مع شاشة LCD وأزرار لتغيير نقاط الضبط، كما تم تزويدها بأجهزة استشعار لقياس درجة الحرارة والرطوبة ونسبة ثاني اكسيد الكربون داخل الحاضنة. هذه البطاقة الالكترونية قادرة على التحكم وتنظيم شروط التحضين في الحاضنة وكذلك تسيير التقليب الآلي للبيض.

الكلمات الأساسية: الحضانة الصناعية ، Atmega328 ، DHT22 ، LM35 ، MQ135 ، triacs.

The abstract:

An incubator is an apparatus for the artificial incubation of eggs to obtain large quantities of chicks. The purpose of this project is the study and the realization of an egg incubator controller whose role is the regulation of temperature, humidity and control of automatic egg turning. We designed an electronic control board based on the Atmega328 microcontroller with an LCD display, push buttons to change set points and sensors to measure temperature, humidity and carbon dioxide. This electronic board is able to control and regulate the incubation parameters in the incubators as well as the management of automatic egg reversal.

Key words: artificial incubation, Atmega328, DHT22, LM35, MQ135, triacs.

Le résumé:

Un incubateur est un appareil destiné à l'incubation artificiel des œufs afin d'obtenir des grandes quantités de poussins. Le but de ce projet est l'étude et la réalisation d'un contrôleur d'incubateur d'œufs dont le rôle est la régulation de la température, de l'humidité et la commande de retournement automatique des œufs. Nous avons réalisé une carte de commande électronique basée sur le microcontrôleur Atmega328 menue d'un afficheur LCD, des boutons poussoirs pour modifier les valeurs consignes, des capteurs pour mesurer la température, l'humidité et le taux de dioxyde de carbone. Cette carte est capable de contrôler et réguler les paramètres d'incubation dans les incubateurs, ainsi que la gestion du retournement automatique des œufs.

Mots clé : incubation artificiel, Atmega328, DHT22, LM35, MQ135, les triacs.

Dédicace

Tous les mots ne sauraient exprime la gratitude ; l'amour, le respect, la reconnaissance, c'est tous simplement que je dédie ce travail :

A la lumière de mes jours, la source de mes effort, la flamme de mon cœur, ma vie et mon bonheur ; Maman que j'adore que dieu vous procure bonne santé et long vie.

Ce travail est dédié à mon père décédé, qui m'a toujours poussé et motivé dans mes études. J'espère que, du monde qui est sien maintenant, il apprécie cet humble geste comme preuve de reconnaissance de la part d'une fille qui a toujours prié pour le salut de son âme. Puisse Dieu, le tout puissant, l'avoir en sa sainte miséricorde.

Aux personnes dont j'ai bien aimé la présence dans ce jour, a tous mes frères : Smain, Islam, Ibrahim, Med Isaak et a ma sœur Amina.

A toute la famille : « BELLABD » et « ALI TURKI ».

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours a mes cotés, et qui m'ont accompagnaient durant mon chemin de vie et d'études : Ssou et ma binôme : Leila

Yasmina

Dédicace

Je dédié ce lapidaire travail :

A mes très chers parents qui m'ont soutenu durant les moments les plus délicates de ce long chemin, mon père qui a sacrifié toute sa vie afin de me voir devenir ce que je suis, et ma mère qui a été à mes cotes et m'a soutenu durant toute ma vie, merci chères parents.

A mes adorables sœurs : Lamia, Moufida et Sirin.

A mon fiancé : Noureddine.

A toute ma famille sans exception.

En fin, je remercie mon binôme BELLABED YASMINA qui a contribué à la réalisation de ce modeste travail et sa famille.

A tous les étudiants (es) de ma promotion.

Leila

Sommaire :

Résumé

Remerciements

Dédicaces

Liste des tableaux

Liste des figures

Introduction générale ……...01

Chapitre I : Généralités sur l'incubation artificiel

1. Introduction ……02
2. Définition et historique ……02
3. L'Incubation artificielle……03

3.1 Stockage des œufs à couver ……03

3.2 Transport des œufs à couver ……04

3.3 Installation de la couveuse ……04

3.4 Préchauffage des œufs……05

3.5 La température d'incubation ……05

3.6 Le niveau d'humidité relative ……06

3.7 La ventilation ……07

3.8 Retournement des œufs ……07

3.9 Mirage des œufs à couver ……08

3.10 L'éclosion ……09

4. L'incubateur ……10

4.1 Les types de l'incubateur ……11

4.1.1 L'incubateur manuel ……11

4.1.2 L'incubateur semi-automatique ……11

4.1.3 L'incubateur automatique ……12

5. Conclusion ……13

Chapitre II : Conception et réalisation du contrôleur

1. Introduction ……14
2. Les paramètres à régler dans l'incubateur automatique des œufs……14

2.1 Contrôle de température …….....….....14
2.2 Contrôle de l'humidité …….....……..14
2.3 Retournement des œufs incubés 14
2.4 La ventilation …….....……15
3. Le contrôleur d'incubateur ……..15
4. Les composants du contrôleur d'incubateur des œufs ……..……..16
5. Spécifications de chaque composant électrique ……..……..17
5.1 Le microcontrôleur ATmega328 …….....17
5.1.2 ATmega328 sans Arduino …….....…….....……18
5.2 Module Afficheur LCD ……..…….....…….....……20
5.3 Le capteur de température et humidité DHT22 …….....……..21
5.4 Le capteur de température LM35 …….....…….....……..22
5.5 Capteur de qualité d'air MQ135 …….....……...23
6. La commande des sorties 220V ……..…….....……25
6.1 Le triac BTA16 ……..…….....……27
6.2 L'optotriac MOC3023 …….....…….....……..28
6.3 La commande du triac en commutation …….....…….....……..30
6.4 Le passage par le zéro secteur ……..…….....……..31
6.5 L'optocoupleur 4N25 ……..…….....……...32
7. conception de la carte électronique ……...……..33
7.1 Présentation générales de Eagle …….....…….33
7.1.1 Etapes de conception ……...…….....…….…33
7.1.2 Travail sous forme de projets …….....….34
7.1.3 Elaboration du schéma structurel (schematic)…….....…….....…..34
7.1.4 Choix des composants …….....……..35
8. Création schémas des circuits ……...…….…….35
8.1 Routage (Création du typon) …….....…….38
9. Conclusion ……...…….…….……..……39

Chapitre III : La programmation du contrôleur d'incubateur

1. Introduction……..…..…40
2. L'environnement de la programmation de l'arduino ……..…….....……..40
2.1. Structure générale du programme (IDE Arduino) ……..……...…..……40

2.2. Transfert du programme ……...……..…….41
2.3. Description du programme ……....…….……..41
2.3.a Les commentaries ……....……..42
2.3.b Definition des variables ……..…....…..……42
2.3.c Configuration des entrées et des sorties void setup …….....……..42
2.3.d Programmation des interactions void loop…….……....…….42
2.4 Les étapes de téléchargement du programme…….....……42
3. Structure du programme du contrôleur d'incubateur d'œufs ……..……..……..43
3.1 Acquisition de l'humidité…….....……...43
3.2 Acquisition de température …….....……46
3.3 Acquisition de la qualité d'air…….……..…...……47
3.4 La communication avec la carte interface utilisateur ……..……....……..47
4. La régulation et la commande des sorties 220v …….……..……54
4.1 La commande tout ou rien ……...…….54
4.2. La commande PID……..……...…...…55
4.2.a : Le régulateur proportionnel (P) …….....…...56
4.2.b Le régulateur proportionnel intégral (PI) …….....……..…….56
4.2.c Le régulateur proportionnel intégral dérivé (PID)…… …….....…….56
4.2.d Réglage des coefficients d'un PID ……..…….....……...57
5.Conclusion……..……...…..60
Conclusion générale……..……..……...61
Référence bibliographique

Liste des tableaux :

Tableau	Titre	Page
Tableau I.1	Tableau d'incubation des œufs……...……	**5**
Tableau I.2	Taux d'humidité optimal……...……...	**7**
Tableau II.1	Caractéristiques techniques de DHT22 (AM2302)……...…….	**22**

Liste des figures :

Figure	Titre	Page
Figure I.1	Les fours d'incubation d'œufs……..	**2**
Figure I.2	Couveuse au pétrole……..	**3**
Figure I.3	Schéma des solutions pour avoir une température stable……	**6**
Figure I.4	Le retournement des œufs……..	**8**
Figure I.5	Les deux méthodes de retournement des œufs……...	**8**
Figure I.6	Mirage d'œufs après 9 jours d'incubation……..…	**9**
Figure I.7	Eclosion d'un œuf ……...……....	**10**
Figure I.8	Couveuse manuel Banbury cross 96 œufs ……..……..	**11**
Figure I.9	FIEM25 Semi-automatique…….	**12**
Figure I.10	Couveuse automatique T4800C de 4800œufs……	**13**
Figure II.1	Système de moteur d'incubateur d'œufs 220V……	**15**
Figure II.2	Le schéma synoptique de contrôleur d'incubateur ……	**16**
Figure II.3	Microcontrôleur ATmega328……..……	**17**
Figure II.4	ATmega328 + Quartz et deux condensateurs……...…….	**19**
Figure II.5	Brochage de l'ATmega328…….....	**19**
Figure II.6	Afficheur LCD 20x4 ……..……..	**20**
Figure II.7	Montage du Module I2C avec l'afficheur LCD……	**21**
Figure II.8	Capteur de température et humidité DHT22……..……	**22**
Figure II.9	Capteur de température LM35.…….	**23**
Figure II.10	Courbes de précision des capteurs LM35. ……	**23**
Figure II.11	Capteur de qualité d'air : module MQ135……..	**24**
Figure II.12	Caractéristiques de sensibilité de le MQ-135 pour plusieurs gaz……...	**24**
Figure II.13	Le relais…….....	**25**
Figure II.14	Constitution et symbole d'un triac……..	**26**
Figure II.15	Fonctionnement de triac ……..……	**26**
Figure II.16	Le triac BTA16……....	**27**
Figure II.17	Principe de commande en Tout ou Rien par Triac …….	**28**
Figure II.18	Le triac BTA16 avec isolation électrique renforcée par le MOC3023……	**28**
Figure II.19	Symbole d'un optotriac.....	**29**
Figure II.20	L'optotriac MOC3023……	**30**

Figure II.21	Temps de conduction du triac…….	**30**
Figure II.22	Le passage par zéro secteur……..…….	**31**
Figure II.23	Détecteur de passage par zéro avec pont de diodes et l'optocoupleur 4N25…..	**32**
Figure II.24	Optocoupleur 4N25…….	**32**
Figure II.25	Etapes de conception.……...……..	**33**
Figure II.26	Creation d'un projet……...	**34**
Figure II.27	Création du schéma structurel.…….	**35**
Figure II.28	Schéma de Circuit interface utilisateur.…..…..…..…..…..…..…..…..…..…..…..…..…	**36**
Figure II.29	Schéma de circuit des capteurs. …….....	**36**
Figure II.30	Schéma de Circuit principal de commande. ……..	**37**
Figure II.31	Typon de Circuit principal de commande…….…….	**38**
Figure II.32	Typon de Circuit interface utilisateur……..……..	**38**
Figure II.33	Typon de circuit des capteurs.…….……...……	**38**
Figure III.1	Interface IDE Arduino……	**41**
Figure III.2	Paramétrage de la carte.……...…...	**41**
Figure III.3	Trame de communication entre DHT22 et ATmega328……..…….….	**43**
Figure III.4	ATmega envoi le signal au DHT22……..…….....	**44**
Figure III.5	La réponse de DHT22……	**44**
Figure III.6	Bibliothèque de DHT22…….……..	**45**
Figure III.7	La tension AREF…….....	**46**
Figure III.8	Organigramme de sélection de mode.….....	**48**
Figure III.9	Bibliothèque de l'afficheur LCD……	**49**
Figure III.10	Organigramme de l'afficheur LCD..……...	**50**
Figure III.11	Organigramme de bouton de sélection UP…….	**51**
Figure III.12	Organigramme de bouton de sélection DOWN……..…	**52**
Figure III.13	Organigramme de programme complet. ……...……	**53**
Figure III.14	La commande en tout ou rien..……...	**54**
Figure III.15	La régulation PID.....	**55**
Figure III.16	Organigramme de l'ISR de l'interruption externe.....…….	**58**
Figure III.17	Organigramme de l'ISR de l'interruption interne……..….	**58**
Figure III.18	Graphes des interruptions …...……...……	**59**
Figure III.19	Organigramme de la commande PID……...……...	**59**

INTRODUCTION GENERALE

Introduction générale :

L'incubation artificielle est l'ensemble des opérations qui, à partir d'œufs pondus, permet d'obtenir le maximum de poussins viables. Cette technique utilise des incubateurs, qui sont des machines reproduisant les conditions de développement fœtal. Elles sont conçues pour régulariser la température, l'humidité la ventilation, et la rotation des œufs. Pour ajuster et régler ces quatre paramètres importants, on a conçu et réalisé un contrôleur qui est le lien entre l'utilisateur et la couveuse, afin d'obtenir un développement embryonnaire normal et pour avoir de meilleur résultats.

Ce mémoire présente un travail de fin d'étude master, ayant pour but l'étude et la réalisation d'un contrôleur universel pour les incubateurs d'œufs. Nous avons utilisé l'ATmega328 comme le cerveau de ce contrôleur pour lire les données qui viennent des capteurs de l'humidité, de la température et le capteur du niveau de CO2 dans l'incubateur. Le contrôle de ces paramètres est fait à travers des sorties 220v pour placer les résistances de chauffage, l'humidificateur, le ventilateur et le moteur de retournement automatique des œufs. Les sorties sont tous commandées via des triacs pour assurer la robustesse du contrôleur.

On commence notre travail par une introduction générale suivi de trois chapitres.
Dans le premier chapitre nous nous présentons des généralités sur l'historique de l'incubation artificielle, ses caractéristiques, et les types des incubateurs qui existent.

Le deuxième chapitre est consacré à la réalisation de la partie électronique du contrôleur d'incubateurs avec une explication détaillée de chaque composant utilisé dans cette réalisation.

Le troisième chapitre s'intéresse à l'aspect programmation et la réalisation de la carte du contrôleur d'incubateur.

Enfin, une conclusion générale résume les principaux résultats auxquels nous avons abouti.

Chapitre I

Généralités sur l’incubation artificielle

1. Introduction :

L'incubateur des œufs est l'une des inventions qui offrent des opportunités, en particulier au cultivateurs et les entreprises fournisseurs du poussin dans le but de couver un nombre d'œufs relativement important, qu'une seule poule peut gérer à la fois.

La fonction de l'incubateur d'œufs est de prendre en charge le travail des animaux pour incuber un œuf à couver. Les œufs sont incubés artificiellement depuis des milliers d'années. L'incubation artificielle consiste à contrôler automatiquement la température, l'humidité, la ventilation de l'incubateur et le retournement automatique des œufs.

2. Définition et historique :

Le mot incubateur peut désigner plusieurs choses :

- Un projet de création d'entreprises : incubateur d'entreprises.
- Un appareil chauffant servant à accueillir des bébés prématurés (couveuse)
- Une armoire thermo-statée dans les laboratoires de biologie.
- L'incubateur qui serve à l'incubation artificiel des œufs de poule, d'oie, reptile... etc.

Les premières couveuses artificielles ou incubatrices sont utilisées par les égyptiens et les chinois. Les égyptiens ont exploité sur une grande échelle l'incubation artificielle dans des fours spéciaux parfois encore en usage de nos jours. Ces fours étaient de véritables monuments comprenant un nombre variable de fours chauffés aux excréments de chameaux desséchés. L'entrée de chaque chambre de l'incubateur était un passage, à travers un petit trou, d'homme.

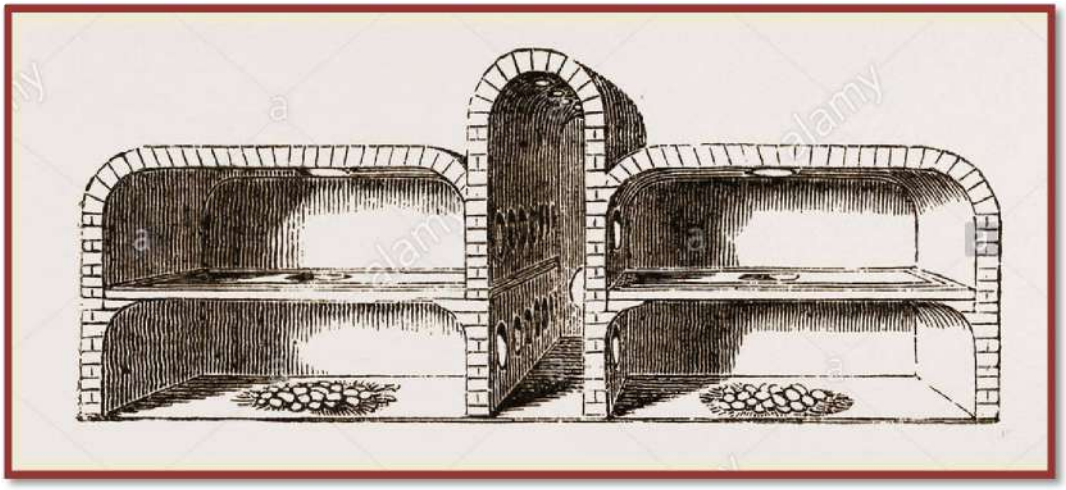


Figure I.1 : Les fours d'incubation d'œufs**.**

Des milliers d'œufs étaient placés sur le sol de chaque pièce et ils étaient retournés 2 fois par jour. Le control de la température était obtenu en contrôlant la force du feu ou l'ouverture des trous de passage d'homme et l'ouverture de trous de ventilation dans le toit des fours. Certains avaient la contenance de 90 000 œufs et des rendements de 70% d'éclosion.

Dès la deuxième moitié du dix-neuvième siècle, des cou0veuses aux pétroles étaient présentées sur le marché et obtenaient des résultats forts satisfaisants **[1]**.



Figure I.2: Couveuse au pétrole.

3. L'Incubation artificielle :

L'incubation artificielle est l'ensemble des opérations qui, à partir d'une quantité d'œufs pondus, permet d'obtenir le maximum des poussins viables au coût le plus bas possible. Cette technique utilise des incubateurs qui sont conçus pour régulariser la chaleur, l'humidité, la ventilation et la rotation des œufs afin que s'accomplisse un développement embryonnaire normal. Une éclosion optimale et une bonne qualité du poussin ne peuvent être obtenues que si l'œuf est maintenu dans des conditions optimales, de la ponte à la mise en incubation **[2].**

3.1 Stockage des œufs à couver

Le stockage est une étape indispensable durant laquelle la chambre à air va se former, du côté le plus large de l'œuf où les pores sont plus nombreux et laissent passer l'air entre la membrane interne et la membrane externe. Lorsque l'œuf est laissé au repos quelques jours (au minimum 2 jours), l'albumen se dégrade et permet à l'embryon de se rapprocher de la chambre à air, pour autant que l'œuf soit maintenu en position verticale, côté large vers le haut, la densité du jaune étant inférieure à celle de l'albumen. L'embryon

minuscule entrera dans un état de latence et pourra être conservé sans danger jusqu'à un maximum de 10 jours. Ses chances d'évoluer correctement ne seront pas diminuées par cette latence pour autant qu'elle se fasse dans de bonnes conditions de température, d'humidité et de retournement de l'œuf. Le stockage favorise même la perte en eau durant l'incubation sans pour autant avoir d'effets négatifs sur l'éclosabilité.

Le local de stockage des œufs doit être propre, bien ventilé et à l'abri de toute condensation et prédation. Sa température optimale sera de 15 °C. Il n'y a cependant pas de différence observée à l'éclosion pour des températures comprises entre 13 et 18 °C. Le taux d'humidité relative doit être assez élevé sans pour autant atteindre le point de rosée ou condensation. Aucune vapeur ne peut être produite dans ce local. Le retournement des œufs pendant cette période est conseillée et sera effectué une fois toutes les 12 heures. Souvent les œufs sont stockés en position horizontale mais certains éleveurs les conservent en position dressée avec la chambre à air vers le haut. Enfin, le stockage des œufs permet de constituer des lots d'œufs qui seront mis en même temps dans l'incubateur.

3.2 Transport des œufs à couver :

Les œufs sont transportés dans des caisses conditionnées avec des alvéoles en mousse afin d'amortir les trépidations pendant le transport. Il faut toujours éviter que les œufs ne soient en contact entre eux pour prévenir les fêlures de coquilles dues aux chocs et la propagation des germes pathogènes. Les trépidations pendant les trajets, surtout si les œufs doivent subir un long voyage, ont pour conséquence de diminuer le taux de fécondité. Une diminution de plus de 10% a été constatée lors de transports de quelques heures en voiture (par exemple lors du transport des œufs vers un centre d'accouvage) **[3].**

3.3 Installation de la couveuse :

La couveuse doit être placée dans une pièce bien isolée dans laquelle la température reste le plus constant possible jour et nuit La température de la salle doit être comprise entre 18 et 20°C et l'hygrométrie supérieure à 70%. Il faut aussi assurer une bonne ventilation tout en évitant les courants d'air. Par ailleurs, avant la mise en service de la couveuse, il est préférable de la faire fonctionner pendant une semaine afin de pouvoir effectuer tous les réglages (stabilité de la température et de l'hygrométrie). On considère que le réglage est au point lorsque la couveuse maintient une température stable pendant une période de 24 heures sans réajustement du thermostat.

La durée d'incubation des œufs est variée en fonction de facteurs propres à l'œuf (souche, âge de l'œuf au moment de sa mise en incubation, poids). La durée et surtout les résultats d'incubation sont liés à un ensemble de paramètres dont les principaux sont :

La température, l'hygrométrie, les teneurs en oxygène et en gaz carbonique de l'air et le retournement.

Tableau I.1: Tableau d'incubation des œufs.

Oiseaux Les espèces d'oiseaux	Température (°C) Couveuse	Température (°C) Eclosoir	Humidité (RH %) Couveuse	Humidité (RH %) Eclosoir	à Piailler (Jours)	Eclosion d'un œuf (Jours)
Poulets	37,5	37,0	52	75	18	21
Canards	37,5	37,0	58	75	25	28
Barbarie	37,5	37,0	60	75	31	34
d'Ornement	37,5	37,0	55	75	19-27	22-30*
Oies	37,5	37,0	55	75	28	31
guinée	37,5	37.0	55	75	27	30
Oies blanches	37,5	37.0	45	75	27-30	30-33*
Oies lourdes	37,5	37,0	50	75	31-33	34-36*
Dindes	37,5	37,0	55	75	25	28
pintades	37,5	37,0	55	75	25	28
Cailles						
japon	37,5	37,0	45	75	15	18
Colin de Virginie	37,5	37,0	45	75	20	23
De chine	37,5	37,0	45	75	12	16
Faisan						
Sauvage	37,5	37,0	50	75	21	24
d'Ornement	37,5	37,0	50	75	20-25	23-28*
Perdrix	37,5	37,0	47	75	20	23
Paon	37,5	37,0	50	75	25	28
Pigeons / colombes	37,5	37,0	50	75	12-13	15-16*
Cygnes	37,5	37,0	50	75	26-32	30-36*
Autruche	36,0	35,5	30	75	35	42
Émeu	36,0	35,5	40	75	46	50-52
Nandou	36,0	35,5	40	75	33	36
Perroquets	37,5	36,6	50	75	15-26	18-29*
* Varie selon	les espèces					

3.4 Préchauffage des œufs :

Pour obtenir un bon taux d'éclosion, il est nécessaire de réchauffer les œufs avant leur mise en incubateur. Ce procédé permet d'éviter un choc thermique aux œufs mais également il permet un démarrage plus rapide et plus homogène du développement embryonnaire. Il est recommandé de préchauffe les œufs pendant 56 à une température compris entre 25 et 28°C **[4].**

3.5 La température d'incubation :

Une couveuse doit fournir une température comparable à celle d'une Poule qui chauffe ses œufs avec son corps. Si la température ambiante Descend en dessous de 35 °C, il faut se servir d'une couveuse équipée d'un chauffage.

Un thermomètre (0-50 °C) est indispensable de même que le maintien de la bonne température Nécessite une bonne isolation.

La température obtenue par chauffage se règle manuellement ou à L'aide d'un thermostat. Ne faites pas incuber d'œufs pendant les saisons Où la température ambiante de la pièce ne descend pas au-dessous De 40 °C. Une telle chaleur tue les embryons. Dans ce cas, on ne peut pas incuber, sauf dans une chambre climatisée, ce qui entraîne d'autres Complications car il faut maintenir la température exacte et l'humidité Relative.

Une hausse de température et une baisse d'humidité peuvent donner ensemble des résultats désastreux. Une température trop basse retarde l'éclosion des œufs, mais est toutefois moins dangereuse qu'une température trop élevée. A partir du 19eme jour de l'incubation, la température doit baisser de 1°C, car les poussins produisent déjà eux - mêmes de la chaleur. **[5].**

La figure suivante montre un schéma des solutions pour stabiliser la température dans un incubateur :

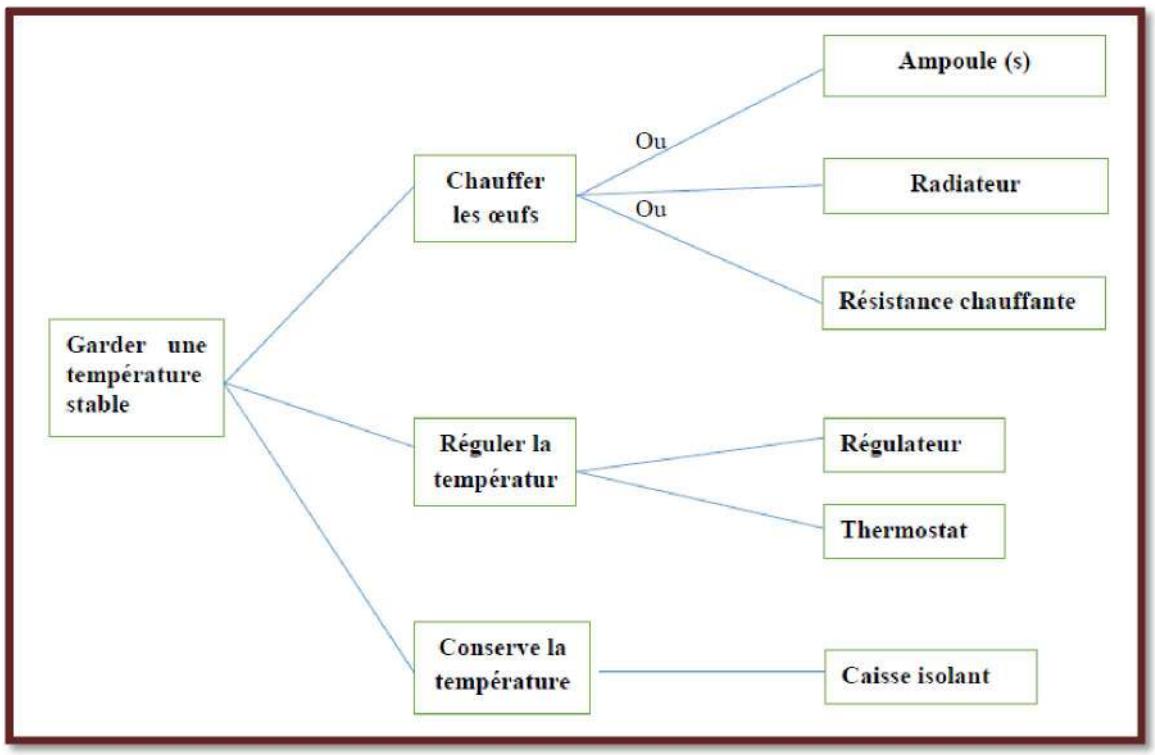


Figure I.3 : Schéma des solutions pour avoir une température stable.

3.6 Le niveau d'humidité relative :

L'humidité relative (HR) varie fortement avec la température. Si ce niveau n'est pas bon, les embryons se déshydratent dans l'œuf ou ne parviennent pas à éliminer les gaz toxiques qu'ils produisent. Un bon niveau d'humidité s'obtient en installant dans la

couveuse des récipients remplis d'eau. L'humidité relative se mesure à l'aide d'un thermomètre à cuvette humide ou avec un hygromètre

Le tableau suivant montre les conséquences d'une mauvaise hygrométrie dans l'incubateur **[2].**

Tableau I.2 : Taux d'humidité optimal.

	Hygrométrie trop élevée	Hygrométrie trop basse
Œuf	Augmentation de la contamination et développement des bactéries	Fuite hydrique entraînant des adhérences
Embryon	mort tardive par défaut de perte d'eau	
Éclosion	Prématurée	éclosion tardive
Poussin	Faible, non mobile, poisseux, œdèmes sous cutanés	sec, petit, déshydraté, adhérant à la coquille

3.7 La ventilation :

L'œuf à couver est un être vivant qui respire, une bonne ventilation est donc nécessaire pour assurer le bon développement de l'embryon. La ventilation apporte de l'oxygène à l'embryon et élimine le dioxyde de carbone excédentaire. Le renouvellement d'air est réalisé par réglage des trappes d'entrée et de sortie d'air au niveau des couveuses. En outre, la ventilation permet de régulariser la température et l'humidité dans la couveuse. Entre le 19ème et le 21ème jour, il est préférable de laisser la teneur en dioxyde de carbone croître pour stimuler le déclenchement de la respiration. Par ailleurs, une ventilation déficiente peut provoquer l'étouffement des poussins dans l'œuf **[4].**

3.8 Retournement des œufs :

Le retournement des œufs est un comportement naturel des oiseaux lors de l'incubation. Cette pratique a donc été incluse dans le processus d'incubation artificielle. La rotation des œufs permet la diffusion des gaz à l'intérieur des œufs et entre ceux-ci et l'environnement extérieur. Les œufs doivent être retournés régulièrement, sinon les résultats D'incubation seront très bas. Le nombre de fois que les œufs ont été retournés est en relation directe avec le nombre d'œufs qui éclosent. Il est indispensable de retourner les œufs trois fois par jour pendant les premiers 19 jours, après, le retournement n'est plus nécessaire. Le retournement des œufs à l'aide d'un mécanisme automatique est une opération très facile. Il peut également se faire à la main, agissez doucement mais assez rapidement. Il est bon de retirer totalement le plateau incubation, de retourner les œufs sur 180° et de replacer le plateau en position inverse **[6].**

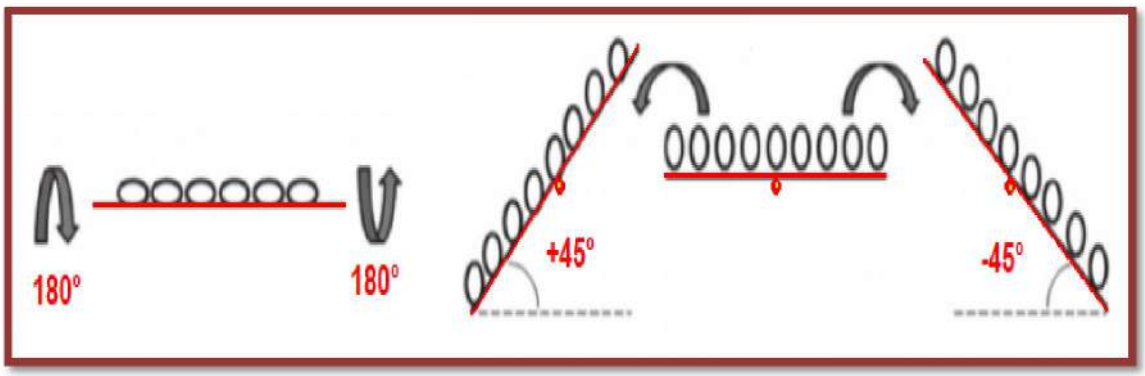


Figure I.4 : Le retournement des œufs.





Retournement 180° Retournement +45°/-45°

Figure I.5 : Les deux méthodes de retournement des œufs.

3.9 Mirage des œufs à couver :

Le mirage des œufs permet de détecter les œufs stériles et les embryons morts. Le mirage des œufs est une opération très utile pendant l'incubation car il permet de voir si le développement du poussin dans l'œuf se déroule à souhait. En cas de développement anormal, retirez l'œuf avant qu'il ne dégage des gaz in désirés. Il faut mirer la première fois le 9ème jour de l'incubation. Faites-le quelques jours auparavant, seulement si vous avez assez d'expérience. Retirez les œufs stériles ou contenant des embryons morts. Ne mirez plus les œufs après le 18ème jour pour ne pas perturber les poussins dans l'œuf. Comme la sortie de l'œuf est une expérience très fatigante pour les poussins, il faut les laisser tranquilles pendant les derniers jours **[7].**

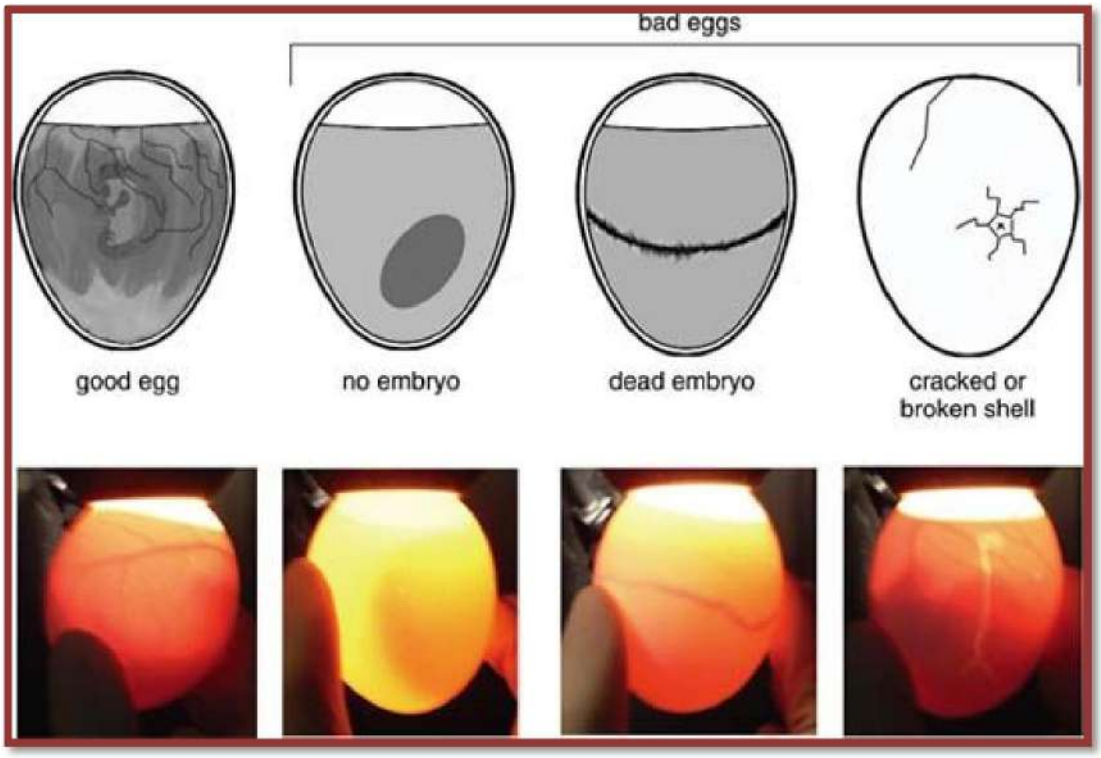


Figure I.6 : Mirage d'œufs après 9 jours d'incubation.

3.10 L'éclosion :

L'éclosion est déclenchée par la demande en oxygène des poussins, cette demande ne pouvant plus être satisfaite par la diffusion.

Afin de ne pas suffoquer, le poussin met sa tête au-dessus de ses pattes et pousse son bec dans la poche d'air pour commencer sa respiration pulmonaire (internal pipping). Son entrée dans la chambre à air lui permet de vivre 24 heures.

Cette ouverture initiale lui donne accès à une quantité limitée d'oxygène, lui permettant de poursuivre ses efforts. Une petite excroissance dure se développe au bout de son bec qui lui permet de casser la coquille. Cette excroissance disparaîtra quelques jours après sa naissance. Les muscles du cou se raidissent et permettent à la tête de casser la coquille. Les premiers poussins sortis émettent un son qui encourage les autres à faire de même. Les poussins doivent être maintenus dans l'éclosion tant qu'ils ne sont pas complètement secs et tant qu'ils ne peuvent pas se tenir debout et marcher **[3].**



Figure I.7: Eclosion d'un œuf.

4. L'incubateur :

Ils existent deux types d'incubateurs : les incubateurs à ventilation naturelle ou statique et les incubateurs à ventilation forcée ou dynamique où l'air est brassé par un ventilateur.

Les incubateurs fonctionnent soit à l'électricité, au gaz ou au pétrole et ont des capacités variant entre 8 et 200 œufs pour les petits producteurs à plus de 100000 œufs pour les grandes fermes.

L'incubateur est un appareil consistant essentiellement en une enceinte isolé, aseptique, maintenue à température et humidité constantes par des systèmes de régulation, et où sont placés les œufs pour la durée de leur incubation.

Les matériaux utilisés pour l'enceinte doivent être non poreux et d'entretien aisé pour le nettoyage, le PVC est idéal pour cet usage. Ils doivent aussi être exothermiques pour éviter les déperditions de calories.

Les systèmes de régulation de la température (thermostats) sont de plusieurs types : bilames, tubes de mercure à contacts électriques, ou thermostats électroniques à micro-processeurs. Ces derniers sont préférables aux autres du fait de leur précision de régulation.

Un système de retournement automatique des œufs sera plus commode que le retournement manuel.

L'incubateur disposera d'ouïes réglables pour permettre l'entrée et la sortie d'air pour le renouvellement de celui-ci dans l'enceinte.

Les incubateurs peuvent être de capacité différentes (50, 100, 250, 1000, … œufs) allant même jusqu'à la taille d'un local entier.

En résumé, l'incubateur imite l'incubation par les poules grâce à un bon réglage de la température, de l'humidité relative, de la ventilation et du retournement régulier des œufs.

Il existe plusieurs modèles avec des différentes tailles par exemple :

4.1 Les types de l'incubateur :

On en distingue de différentes sortes. Elles sont souvent en plastique, de différentes formes et tailles.

4.1.1 L'incubateur manuel :

Idéale pour les débutants et/ou pour couver de petites quantités d'œufs, l'incubateur manuel est simple à utiliser.

• Le retournement des œufs se fait manuellement.

• Elle est généralement équipée d'une petite résistance, d'un thermostat et d'un ventilateur pour répartir la chaleur.



Figure I.8 : Couveuse manuel Banbury cross 96 œufs.

4.1.2 L'incubateur semi-automatique :

La couveuse semi-automatique est préréglée, il suffit de brancher la couveuse et d'ajouter de l'eau dans le réceptacle.

• Thermostat à affichage digital.

• Grille pour placer les œufs.

• Résistance chauffante pour le chauffage.

• Un ventilateur répartit la chaleur.

• Le retournement des œufs est 'effectué par un système de bascule ou autres.

Certaines couveuses peuvent être équipées de deux résistances, de plusieurs tiroirs avec des séparations réglables et adaptables aux œufs



Figure I.9 : FIEM25 Semi-automatique.

4.1.3 L'incubateur automatique :

La couveuse automatique convient aux éleveurs confirmés possédant de nombreuses poules.

• Le retournement des œufs est automatique.

• Thermostat à affichage digital précis.

• L'hygrométrie est régulée.

• Grille pour placer les œufs.

• Plusieurs bacs prévus pour les œufs.

• Bonne répartition de la chaleur.

• Interrupteur d'éclairage indépendant.

• Les autres options pour les plus performantes : matériaux stratifiés avec de bonnes finitions, hygrométrie à bulbe humide, hublot de contrôle, renouvellement de l'air au moyen de trous d'aération, ventilation réglable, thermostat électronique, calendrier d'incubation programmé… **[2]**.



Figure I.10 : Couveuse automatique T4800C de 4800œufs.

5. Conclusion :

Le but de la couvaison artificielle est d'obtenir rapidement des poussins ou continuer de couver des œufs pour avoir une grande quantité d'œufs. Grâce à la couvaison artificielle, on peut palier à l'absence de la poule et avoir tout sous contrôle pour assurer la couvaison des œufs. Nous avons vu dans ce premier chapitre les différentes étapes et les principaux paramètres à prendre en considération pour assure une bonne couvaison. Nous avons faire aussi un aperçu sur l'historique et les différents types des couveuses qui existe. Dans le prochain chapitre nous allons voir la partie matérielle du contrôleur d'incubateur que nous avons conçu pour qu'il soit compatible avec la majorité de couveuses existantes.

Chapitre II

Réalisation de la partie électronique du contrôleur d'incubateur

1. Introduction :

L'incubateur artificiel automatique d'œuf se fait dans une enceinte isolée et équipée de certains appareillages. Cet appareillage consiste généralement à un mécanisme de retournement des œufs, une ou des sources de chaleur pour augmenter la température, une source d'humidité pour humidifier l'air dans l'incubateur et un extracteur d'air pour abaisser la température en cas où elle dépasse les valeurs acceptées. Ces composants sont tous sous le contrôle d'un contrôleur d'incubateur automatique pour réguler la température et l'humidité à des valeurs consignes fixées par l'utilisateur, ainsi que le retournement des œufs pendant des intervalles de temps fixées par l'utilisateur.

Dans ce chapitre, nous allons étudier et réaliser la partie matérielle de notre contrôleur d'incubateur, et voir ses principaux composants électroniques.

2. Les paramètres à régler dans l'incubateur automatique des œufs :

2.1 Contrôle de température :

Dans la plupart des incubateurs le contrôle de la température se fait à l'aide de deux résistances pour le chauffer et ces deux sont :

- Une résistance primaire : a pour but de régler la température dans la couveuse. Dans notre contrôleur, cette opération est faite en utilisant un PID numérique (proportionnel intégral dérivé).
- Une résistance secondaire : a pour but de soutenue la résistance primaire en cas où cette dernière devient insuffisante. La commande de cette résistance se fait par une régulation de type tout ou rien.

2.2 Contrôle de l'humidité :

L'humidité est réglée par des récipients remplis d'eau posé sur le fond de la couveuse est se chauffer à l'aide d'une résistance, plus la surface de l'eau est grande plus l'eau s'évapore. Le contrôle de l'humidité peut aussi fait automatiquement par un brumisateur qui est un appareil innovant permettant de transformer l'eau en microgouttelette, ces derniers vont s'évaporer dans la couveuse. Le réglage d'humidité se fait par une régulation de type tout ou rien.

2.3 Retournement des œufs incubés :

Le retournement des œufs se fait automatiquement au moins deux fois par jours à l'aide d'un moteur qui est un élément indispensable dans une couveuse électrique, car il fournit la puissance nécessaire pour mettre les supports en mouvement il se compose de 3 entrées une entrée pour le retournement des œufs dans un sens, une deuxième pour le retournement dans le

sens inverse et une troisième sortie commun. Le contrôle de retournement des œufs se fait par une régulation de type tout ou rien.

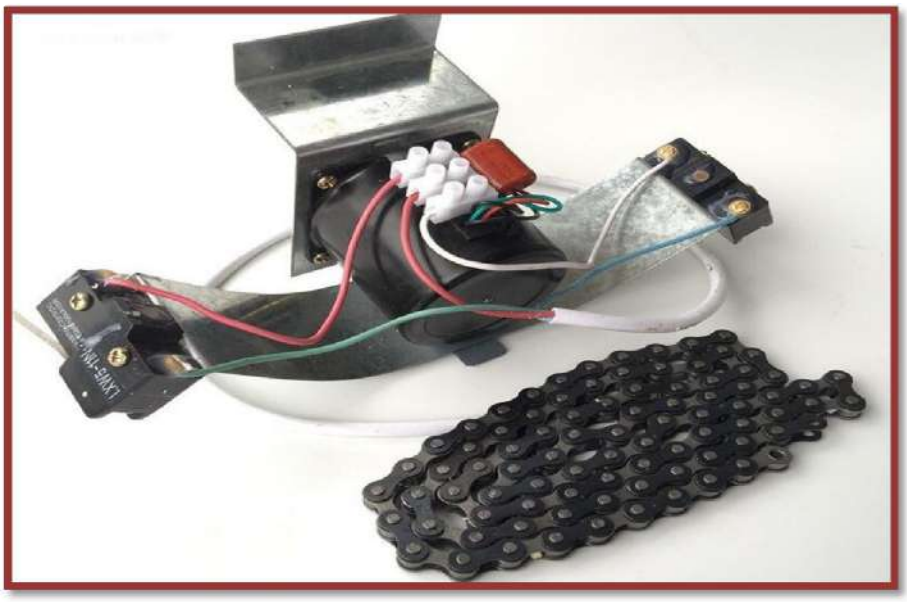


Figure II.1 : Système de moteur d'incubateur d'œufs 220V.

2.4 La ventilation :

Lorsque la température dans la couveuse augmente nous devons mettre hors tension la résistance primaire et la résistance secondaire. Si la température reste toujours élevée on a besoin de faire fonctionner le ventilateur afin de baisser la température.

Il est aussi indispensable de démarrer le ventilateur temps en temps pour changer et rafraîchit l'air dans l'incubateur pour diminuer le taux de co2. La ventilation est aussi contrôlée en tout ou rien.

3. Le contrôleur d'incubateur :

La figure II.2 représente le schéma synoptique du contrôleur d'incubateur qui se compose deux étages :

Un étage d'entre qui comporte

Une alimentation de 5v.

- Un capteur de température.
- Un capteur d'humidité.
- Un capteur de qualité d'air.
- Un détecteur de passage à zéro du secteur pour la commande du triac.

Un étage de sortie qui comporte

- Un afficheur LCD pour visualiser les valeurs de la température et l'humidité.
- Une alarme pour sonner en cas d'anomalies.
- Des sorties de commandes de plusieurs triacs pour commander la résistance principale,

- la résistance secondaire, la ventilation, l'humidification et le moteur.

La commande de la résistance primaire R1 est faite par une régulation de type PID. Par contre, la commande de la résistance secondaire, la ventilation, l'humidification et le moteur de retournement des œufs sont faits : en tout ou rien.

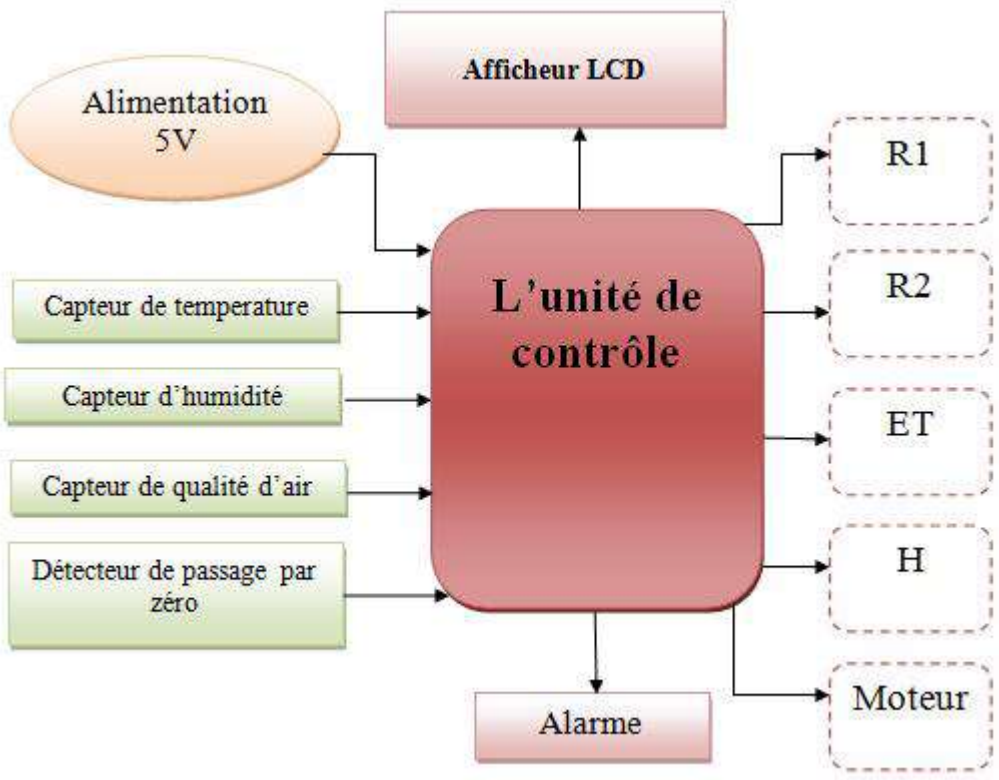


Figure II.2 : Le schéma synoptique de contrôleur d'incubateur.

4. Les composants du contrôleur d'incubateur des œufs :

Les composants utilisés dans notre contrôleur d'incubateurs sont :

- Etage d'alimentation à basse tension se compose d'un (transformateur – un pont de redressement – un régulateur 7805).
- Un microcontrôleur ATmega328.
- Des capteurs de température, d'humidité et qualité d'air.
- Un optocoupleur 4N25 pour la Détection du passage à zéro du secteur
- Trois boutons poussoirs pour le réglage des paramètres d'incubation.
- Un afficheur LCD 20x4 standard.
- Un buzzer comme alarme.
- Des optotriac MOC3023 pour assurer L'isolation entre la partie basse tension et haute tension afin de commander les Triacs.

L'alimentation du contrôleur vienne de la tension alternative du secteur de 220V.

5. Spécifications de chaque composant électrique :

5.1 Le microcontrôleur ATmega328 :

Durant toutes la phase de développement de ce projet nous avons utilisé la carte Arduino Uno comme l'unité de contrôle de notre système, néanmoins, dans le circuit finale nous n'utilisons que le ATMega328, qui est le cœur de la carte Arduino Uno .

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un Microcontrôleur ATMEL de référence ATMega328 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).



Figure II.3 : Microcontrôleur ATmega328.

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 à personnes intéressées plusieurs avantages cités comme suit :

- Le prix (réduits).
- Multi plateforme.
- Un Logiciel Open Source et extensible.
- Environnement de programmation clair et simple.
- Matériel Open source et extensible.

Les caractéristiques principales du microcontrôleur ATmega328 sont :

- FLASH = mémoire programme de 32Ko
- SRAM = données (volatiles) 2Ko
- EEPROM = données (non volatiles) 1Ko

- Digital I/O (entrées-sorties Tout Ou Rien) = 3 ports Port B, Port C, Port D (soit 23 broches en tout I/O).
- Timers / Counters : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits) Chaque timer peut être utilise pour générer deux signaux PWM. (6 brooches OCxA/OCxB).
- PWM = 6 brooches OC0A (PD6), OC0B (PD5), 0C1A (PB1), OC1B (PB3), OC2A (PB3), OC2B (PD3).
- Analog to Digital Converter (resolution 10bits) = 6 entrees multiplexes ADC0 (PC0) a ADC5 (PC5).
- Gestion bus I2C (TWI Two Wire Interface) = le bus est exploite via les broches SDA(PC5)/SCL(PC4).
- Port série (USART) = émission/réception série via les broches TXD(PD1)/RXD(PD0).
- Comparateur Analogique = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption.
- Watchdog Timer programmable.
- Gestion d'interruptions (24 sources possibles (cf. interrupt vectors)) : en résume
- Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).
- Interruptions sur changement d'état des broches PCINT0 a PCINT23.
- Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables).
- Interruption liée au comparateur analogique.
- Interruption de fin de conversion ADC.
- Interruptions du port série USART.
- Interruption du bus TWI (I2C) **[8].**

5.1.2 ATmega328 sans Arduino :

L'ATmega328 C'est le cerveau de notre projet. Il va recevoir le programme que nous allons créer et va le stocker dans sa mémoire avant de l'exécuter. Grâce à ce programme, il va être capable de faire les tâches nécessaires pour gérer l'incubateur.

On peut utiliser le ATmega328 seul Arduino en l'insérant dans un circuit électronique minimal consiste d'une alimentation 5V avec quartz externe de 16MHz et deux condensateurs de 22 pF en céramique.

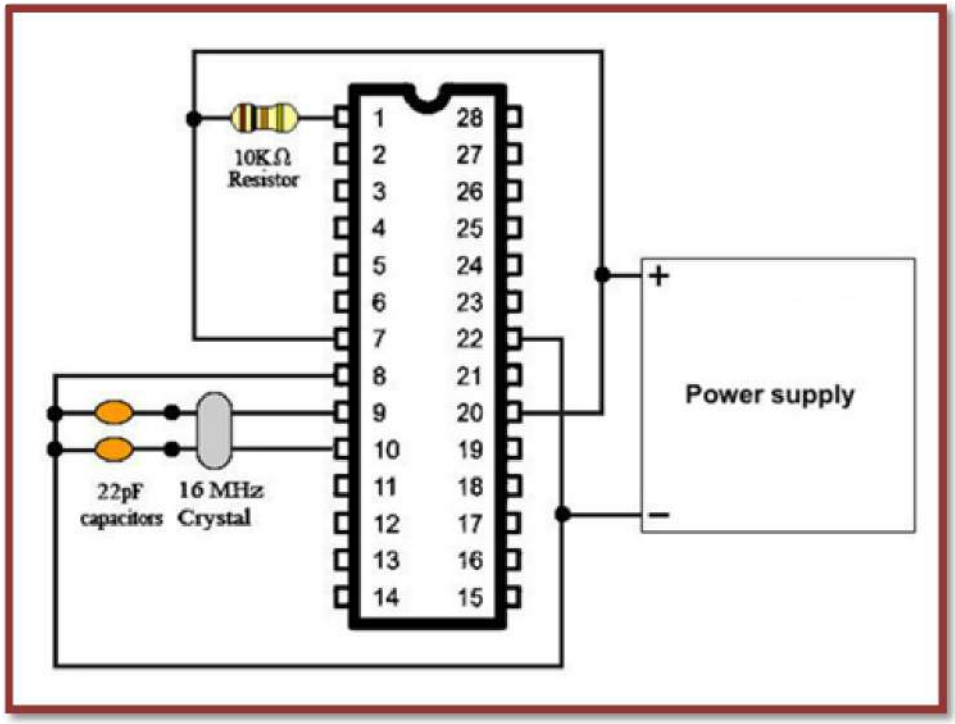


Figure II.4 : ATmega328 + Quartz et deux condensateurs.

La figure suivante montre la correspondance entre les broches de l'Atmega328 et les broches de l'Arduino :

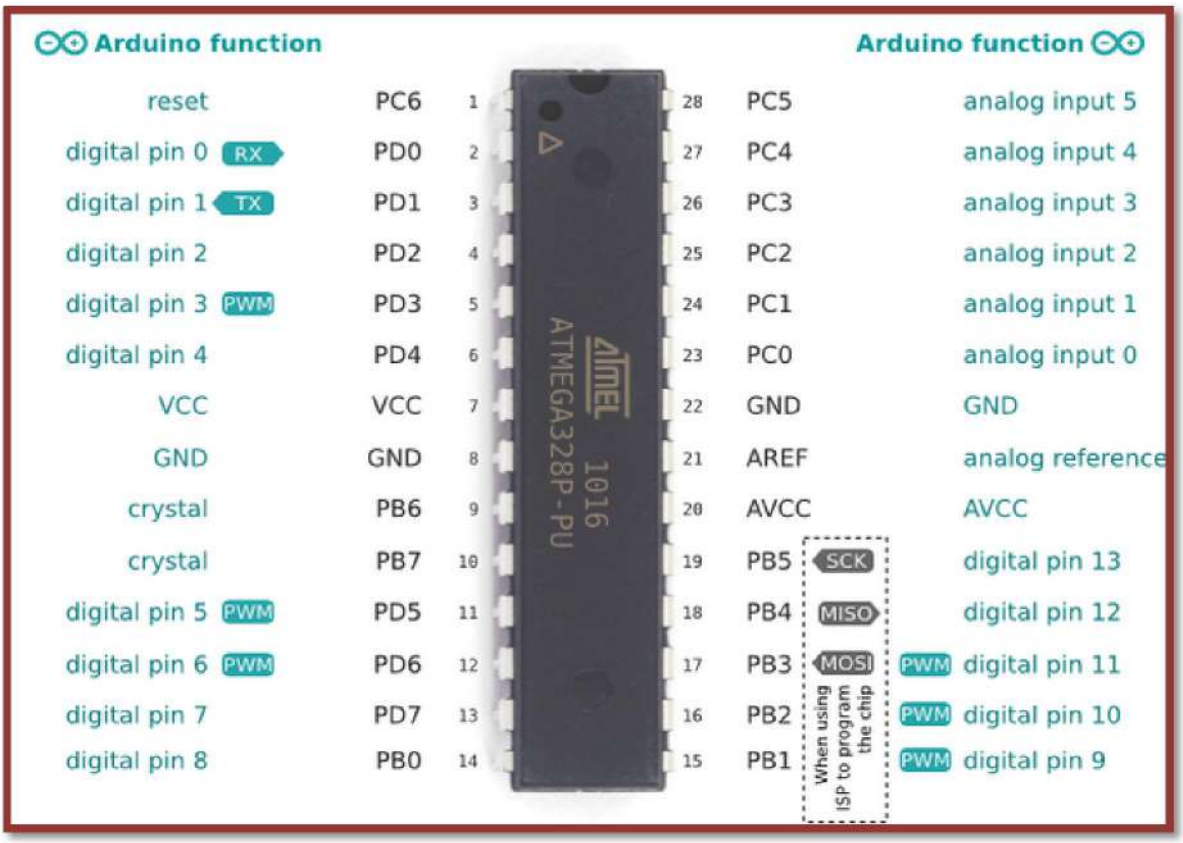


Figure II.5 : Brochage de l'ATmega328.

5.2 Module Afficheur LCD :

Les afficheurs LCD (Affichage à cristaux liquides) alphanumériques présentent une solution facile d'emploi et bon marché de doter les projets électroniques d'une interface indépendante de votre PC.

Les écrans LCD alphanumériques sont essentiellement caractérisés par leur taille. Deux modèles se rencontrent très fréquemment et sont les meilleurs marchés, celui ayant 2 lignes et 16 colonnes d'affichage et celui ayant 4 lignes et 20 colonnes d'affichage.

Dans notre projet, nous avons choisi un afficheur LCD de 20x4 pour l'affichage des variations des valeurs mesurées de température, de l'humidité et le taux de co2.

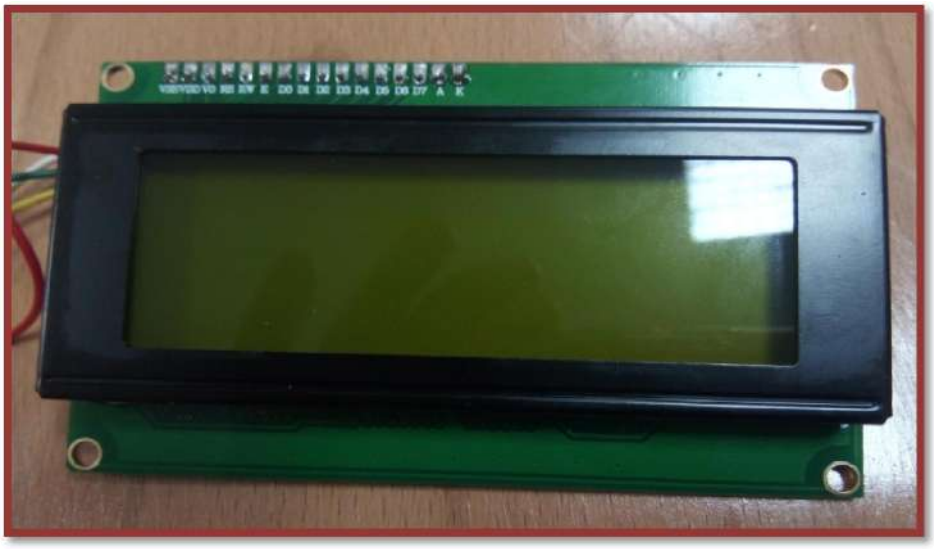


Figure II.6 : Afficheur LCD 20x4.

L'afficheur LCD 20x4 dispose les caractéristiques techniques suivantes :

- Affichage : 20 colonnes, 4 lignes.
- Tension d'alimentation : 5V.
- Courant : 135 mA type. avec rétro-éclairage allumé.
- Modes : I2C ou 9 600 baud de communication série.
- Caractères personnalisés : Jusqu'à 8 caractères personnalisés faciles à définir.

Il existe également de petits modules permettant d'interfacer un afficheur LCD avec un bus I2C. Cette solution peut être intéressante si on manque désespérément des broches sur la carte Arduino, c'est le cas de notre projet puisqu'au lieu de monopoliser 6 à 7 broches, l'écran n'en utilisera plus que

Pour se connecter à un bus I2C il faut donc deux fils d'alimentation et deux fils de communication.

- SDA (Signal DATA) est utilisé pour transmettre les données.
- SCL (Signal CLOCK) est utilisé pour transmettre un signal d'horloge synchrone (signal qui indique le rythme d'évolution de la ligne SDA) **[9].**

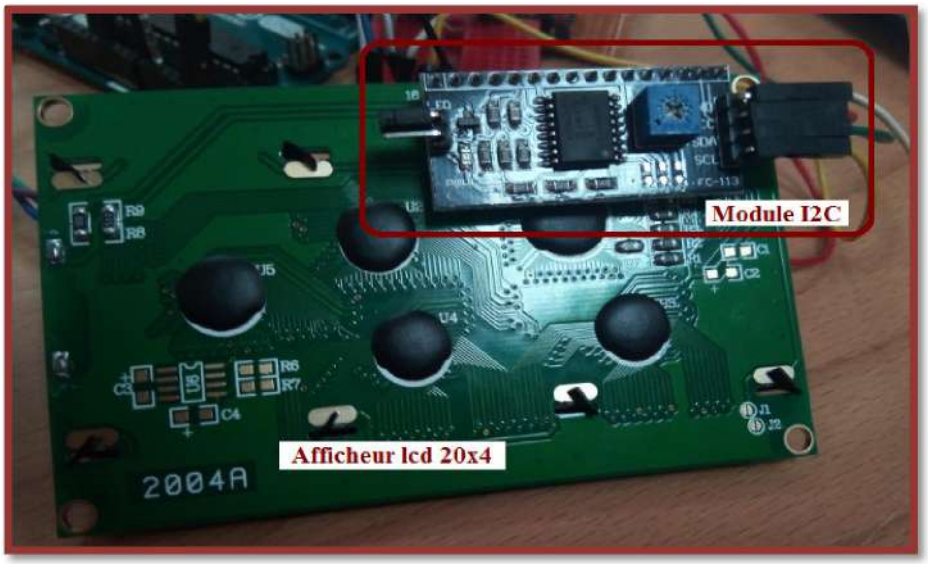


Figure II.7 : Montage du Module I2C avec l'afficheur LCD.

5.3 Le capteur de température et humidité DHT22 :

Le capteur DHT22 (aussi connu sous la référence AM2302) est un capteur de température et d'humidité à sortie numérique série. Il utilise une technique exclusive de collecte de signaux numériques et d'humidité technologie de détection, assurant sa fiabilité et sa stabilité. Ses éléments de détection sont connectés à une puce 8 bits. Chaque capteur de ce modèle est compensé en température et étalonné dans une chambre d'étalonnage précise et le coefficient d'étalonnage est enregistré dans le type de programme dans la mémoire OTP.

Le capteur DHT22 / AM2302 est capable de mesurer des températures de -40 à +125°C avec une précision de +/- 0.5°C et des taux d'humidité relative de 0 à 100% avec une précision de +/- 2% (+/- 5% aux extrêmes, à 10% et 90%). Une mesure peut être réalisée toutes les 500 millisecondes (soit deux fois par seconde) **[10].**

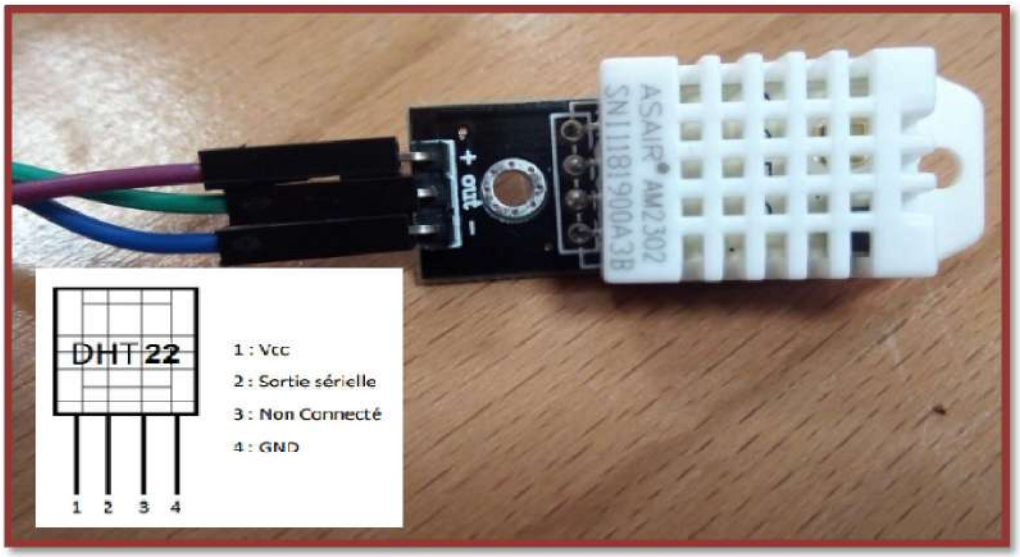


Figure II.8 : Capteur de température et humidité DHT22.

Remarque :

Dans ce projet, on a utilisé le capteur DHT22 seulement pour la mesure d'humidité car lorsque on a fait des tests pratiques de mesure de température nous avons remarqué qu'il prend beaucoup de temps pour répondre à la variation de la température (temps de réponse relativement long), donc on a préféré d'utiliser un autre capteur pour mesurer la température qui est le LM35 et qui a un temps de réponse acceptable.

Tableau II.1 : Caractéristiques techniques de DHT22 (AM2302).

Alimentation	3.3-5.5V
Signal de sortie	Signal numérique via bus 1 fil
Element sensible	Condensateur d'humidité polymère
Plage de fonctionnement	Humidité 0-100% température -40 ~ 80 Celsius
Précision	Humidité +/- 2% température +/- 0.5 Celsius
Résolution et sensibilité	Humidité 1% température 0.1celsius
Stabilité à long terme	+/- 0,5% HR / an
Interchangeabilité	Entièrement interchangeable

5.4 Le capteur de température LM35 :

Le LM35 est un capteur à circuit intégré qui peut être utilisé pour mesurer la température avec un signal électrique proportionnel à la température (en ° C). Il fournit un moyen idéal pour mesurer avec précision la température. Il est également très facile de lire le LM35 en utilisant un convertisseur analogique numérique.

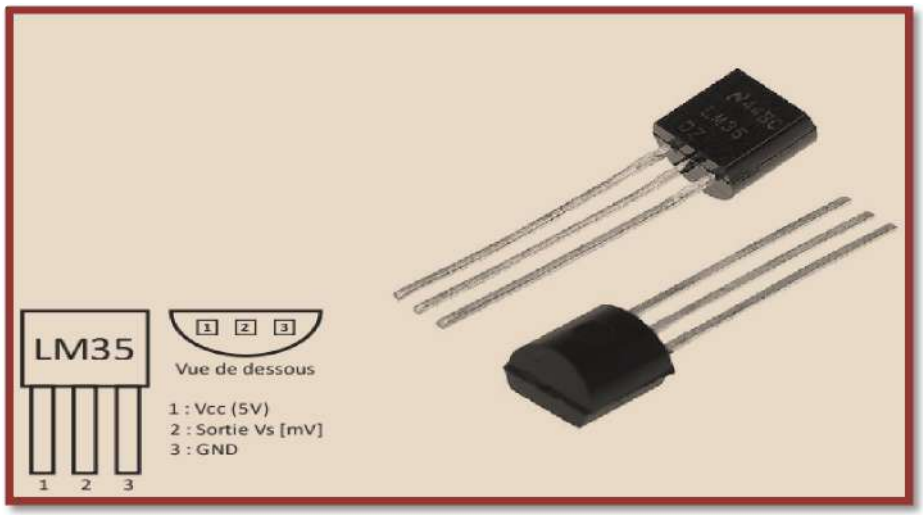


Figure II.9 : Capteur de température LM35.

Le LM35 à une tension de sortie est proportionnelle à la température Celsius. Le facteur d'échelle est 0.1V/°C, le LM35 ne nécessite pas de calibration externe ou de taille et maintient une précision de + / -0,4 °C à température ambiante et + / - 0,8 ° C sur une plage de 0 °C à +100 °C **[11].**

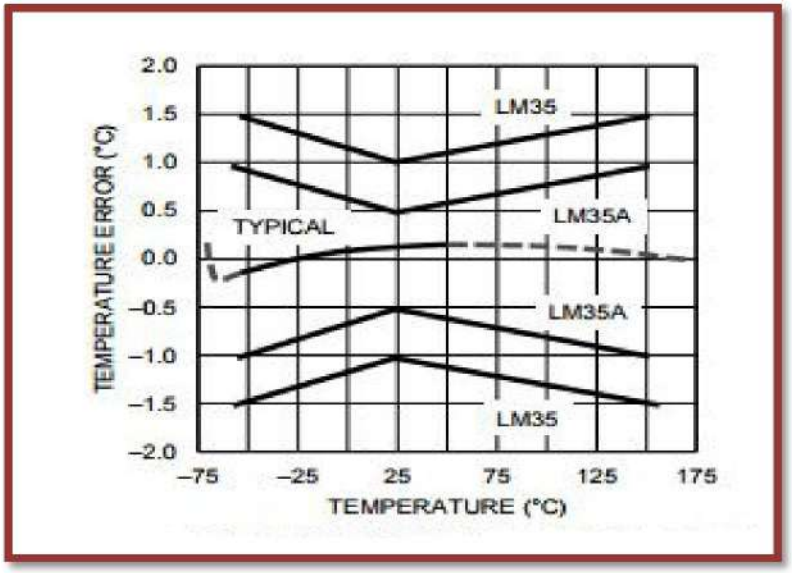


Figure II.10: Courbes de précision des capteurs LM35.

5.5 Capteur de qualité d'air MQ135 :

Le MQ135 est un capteur qui permet de mesurer la qualité de l'air. Le MQ135 est sensible à aux principaux polluants présents dans l'atmosphère. Ce capteur est sensible au CO2, à l'alcool, au Benzène, à l'oxyde d'azote (NOx) et à l'ammoniac (NH3).

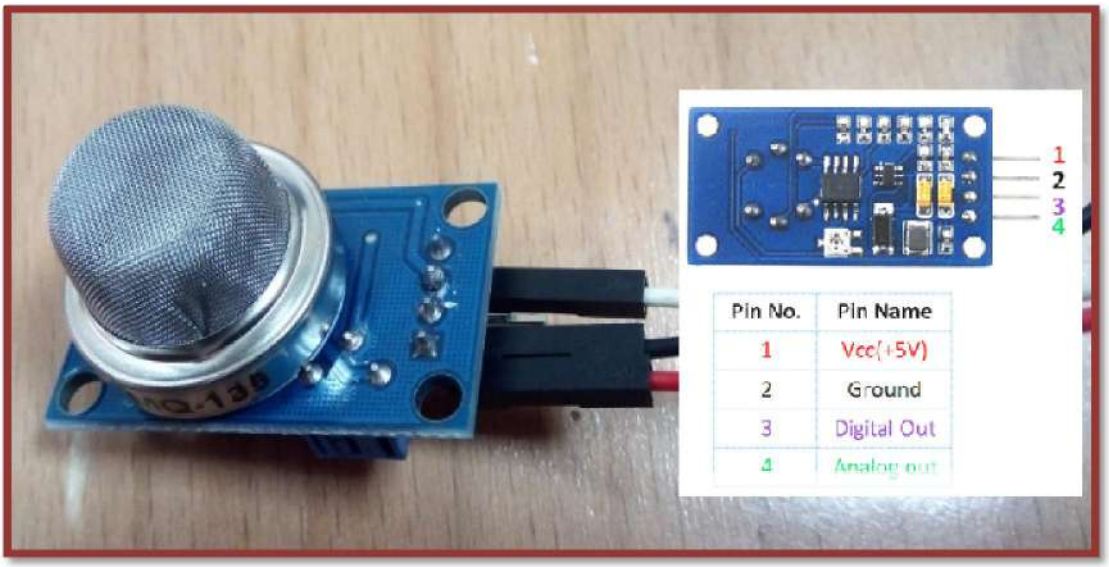


Figure II.11 : Capteur de qualité d'air : module MQ135.

Le module MQ135 possède les caractéristiques techniques suivantes :

- Tension nominale : 5V.
- Une sortie analogique.
- Une sortie numérique tout ou rien selon un seuil réglable.
- Le module possède un indicateur LED de l'état de la sortie numérique.
- Détecte NH3, NOx, alcool, benzene, fumée et CO2.
- Haute sensibilité: 10 - 300 ppm NH3, 10 - 1000 ppm (partie par million) Benzene, 10 - 300 Alcool.
- Plus la concentration est importante plus la sortie analogique est élevée.
- Stable **[12].**

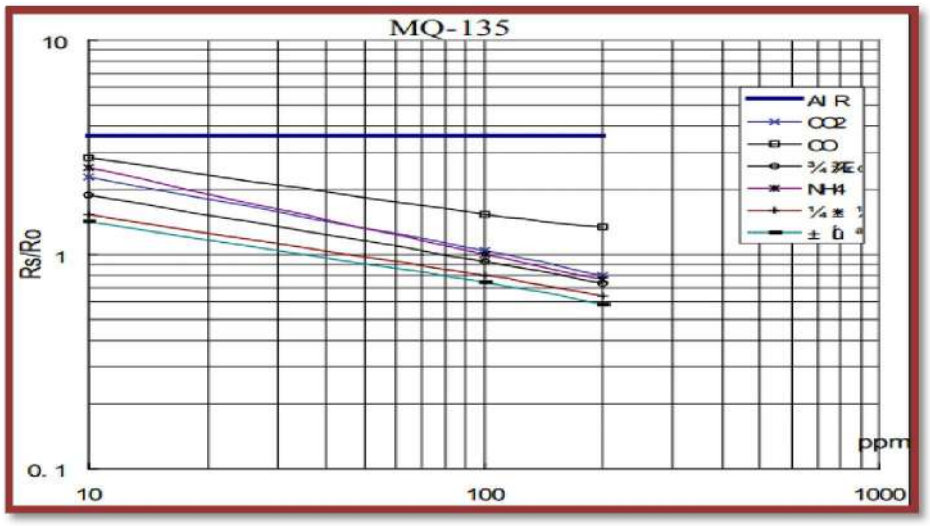


Figure II.12 : Caractéristiques de sensibilité de le MQ-135 pour plusieurs gaz.

La figure II.12 montre les caractéristiques de sensibilité typiques du MQ-135 pour plusieurs gaz. Dans leur : Température : 20 Humidité : 65%, concentration en O2 21%, RL = 20kΩ, Ro: résistance du capteur à 100 ppm de NH3 dans l'air pur. Rs: résistance du capteur à différentes concentrations de gaz.

Remarque : pour convertir de ppm en pourcentage, divisez la ppm par 1000 :

$$x\ (\%) = x\ (ppm) / 1000.$$

Exemple : 400ppm de CO2 représente une concentration de co2 dans l'air de 0.4%.

6. La commande des sorties 220V :

Les pins de sorties de l'arduino ou de l'Atmega328 supportent des faibles courants et des tensions maximales de 5V. Pour cela il est indispensable de placer à chaque sortie un relais 5V/220V ou bien des triacs avec une isolation assurée par des optocoupleurs.

Il est possible, pour les dispositifs commandés en tout ou rien, de travaillé avec des relais. Le relais est une solution à la commande en puissance. Il assure en outre une isolation galvanique en mettant en œuvre un mouvement mécanique.

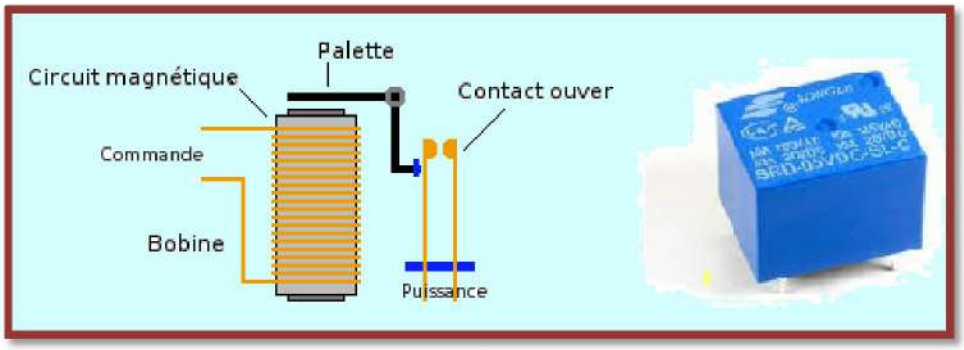


Figure II.13 : le relais.

Le triac (Triode Alternating Current, en anglais) est un dispositif semi-conducteur à trois électrodes qui autorise la mise en conduction et le blocage des deux alternances d'une tension alternative, en général celle du secteur 230 V. Le triac peut passer d'un état bloqué à un régime conducteur dans les deux sens de polarisation, et repasser à l'état bloqué. Par inversion de tension (passage par le "zéro secteur", zero crossing en anglais) ou par diminution de la valeur du courant de maintien.

Par analogie on pourrait dire qu'un triac est constitué de deux thyristors montés "tête-bêche" :

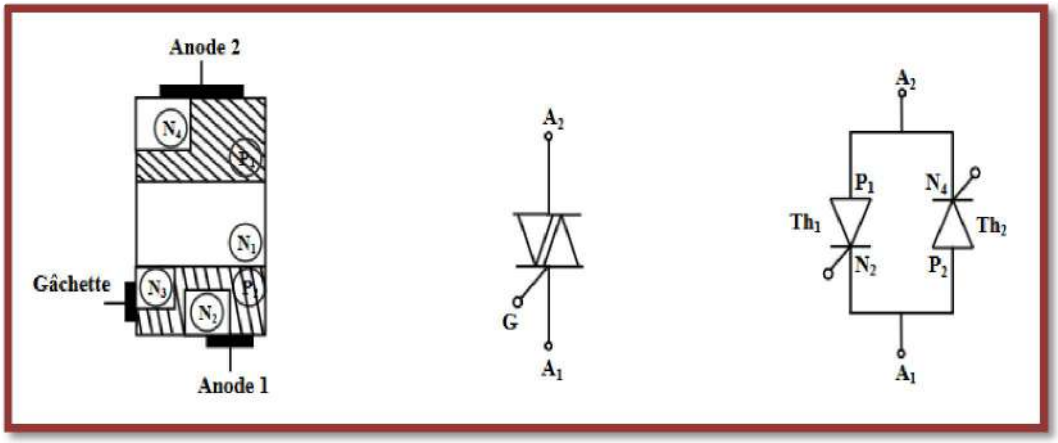


Figure II.14 : Constitution et symbole d'un triac.

Les trois électrodes du triac sont dénommées gâchette (électrode de commande, appelée gate en anglais), et A1 et A2 (pour Anodes 1 et 2) ou, en anglais, MT1 et MT2 (Main Terminals). Ces deux dernières électrodes assurent la conduction principale.

Un courant de commande très faible (environ 50 mA) déclenche le triac, qui reste amorcé jusqu'au passage par zéro des sinusoïdaux secteurs **[13].**

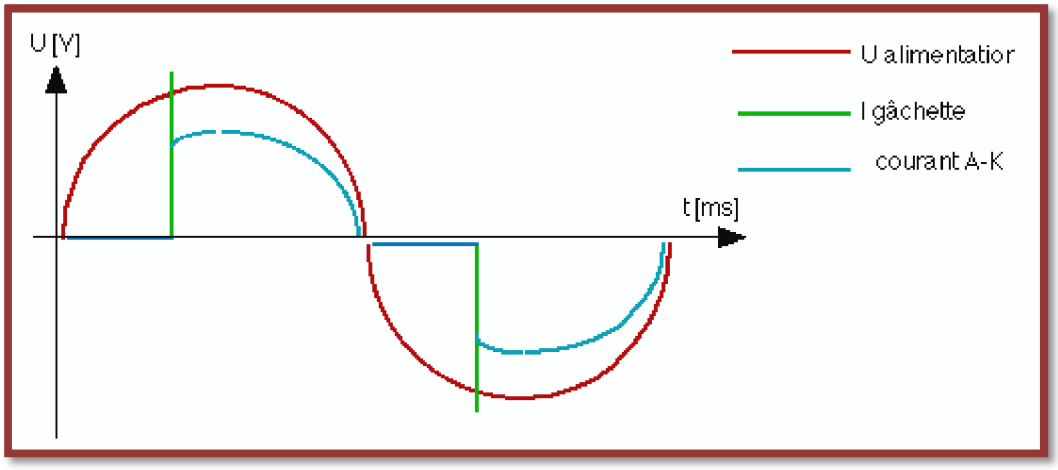


Figure II.15 : Fonctionnement de triac.

La puissance fournie à la charge est maximale lorsque le déclenchement a lieu juste après le passage par zéro de la tension alternative, mais en retardant l'impulsion de déclenchement, on peut faire varier à volonté l'intensité appliquée à la charge.

Pour notre projet, nous avons choisi d'utiliser des triac au lieu des relais, car la durée de vie des relais est courte en cas d'un nombre de commutation important (fatigue des contacts et du ressort de rappel, qui peut se "ramollir" ou même casser) et le brochage pas vraiment normalisé, malgré quelques efforts faits pour certaines catégories de relais (relais en boitier DIL et relais norme "européenne").La compatibilité des relais pas toujours assurée avec les circuits numériques, notamment pour les relais de forte puissance, qui peuvent nécessiter un circuit d'interface spécifique. En outre, pour réguler la température, on ait besoin d'une commande en

commutation (pour le régulateur PID), par contre, les relais sont destinés seulement à la commande tout ou rien.

6.1 Le triac BTA16 :

Nous avons utilisé le BTA16 pour gérer les commandes de puissance des sorties en Tout ou Rien et en commutation pour la commande PID.

Les caractéristiques de triac BTA16 sont :

- Courant moyen (16A max).
- Faible résistance thermique avec clip de liaison.
- Isolant céramique à faible résistance thermique pour isolé le BTA.
- Tab isolé (Série BTA, à 2500VRMS) **[14].**

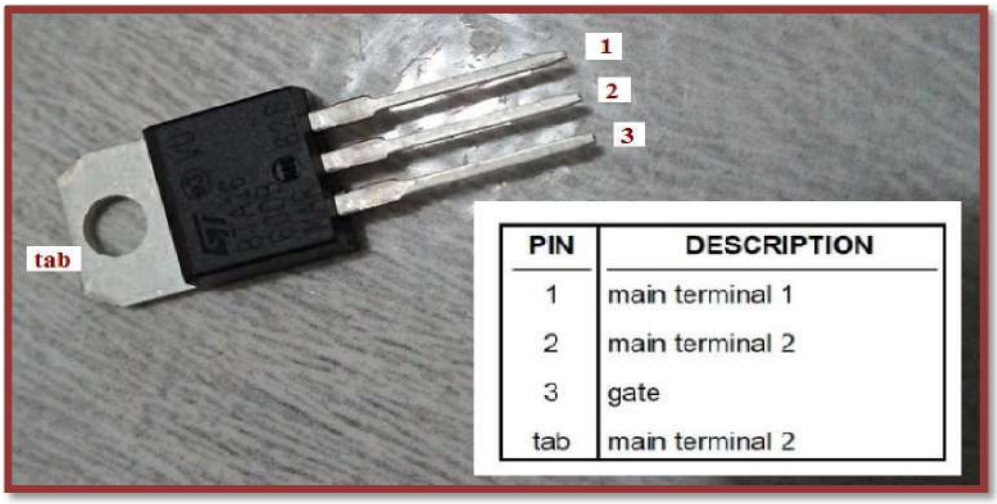


Figure II.16 : Le triac BTA16.

La Commande des sortie 220V pour la résistance secondaire R2, l'humidité H, la ventilation ET et les sorties M1;M2 pour le moteur du retournement des œufs sont faite par la commande Tout ou Rien. Si on maintient la commande active en permanence, le triac reste conducteur tant que la commande est active, il est alors comparable à un interrupteur qui est en position fermée. Lorsque on supprime la commande le triac se re-bloque à la fin de la demi alternance (annulation du courant).

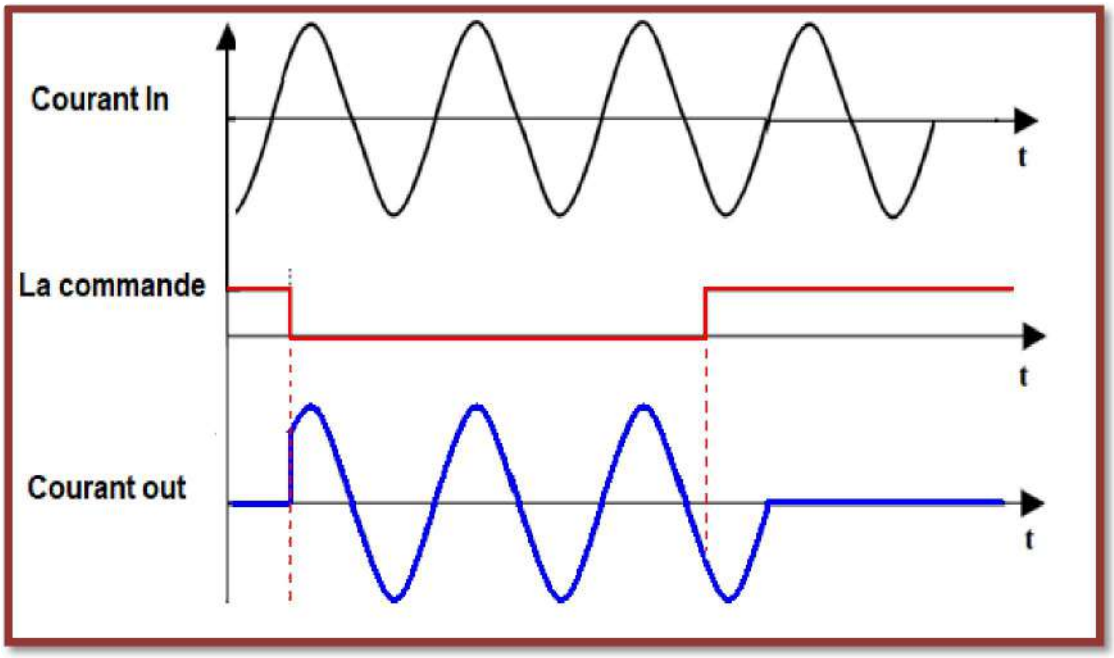


Figure II.17 : principe de commande en Tout ou Rien par Triac.

Dans notre projet on a utilisé un Triac à niveau logique standard Le BTA16, et pour renforcer l'isolation, nous avons placé un optotriac de type MOC3023 entre le microcontrôleur et le BTA16.

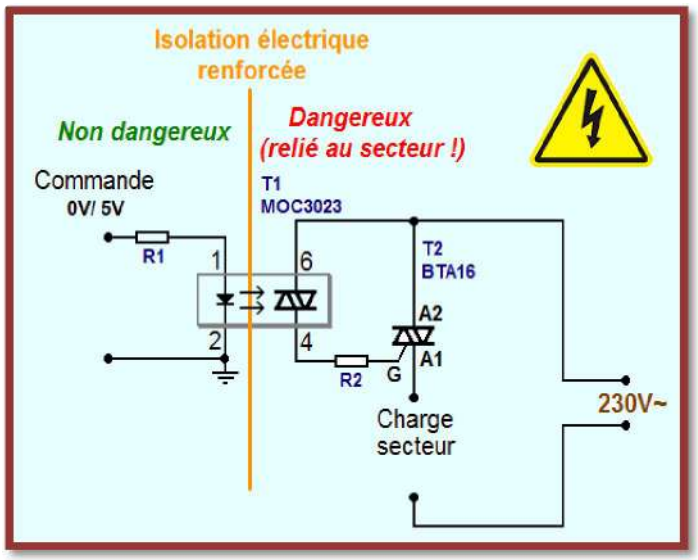


Figure II.18 : Le triac BTA16 avec isolation électrique renforcée par le MOC3023.

6.2 L'optotriac MOC3023 :

Optotriac est un montage qui intègre un triac et un optocoupleur. La mise en œuvre de ce dispositif est très simple, puisqu'il suffit d'appliquer un niveau positif bas tension sur l'entrée de l'opto-coupleur, qui commande à son tour le triac. L'opto-triac fait donc figure d'interface, en

quelque sorte, entre le circuit de commande et le circuit commandé. On peut donc le comparer à un interrupteur. Ce composant offre une isolation galvanique entre la partie "commande" et la partie "puissance".

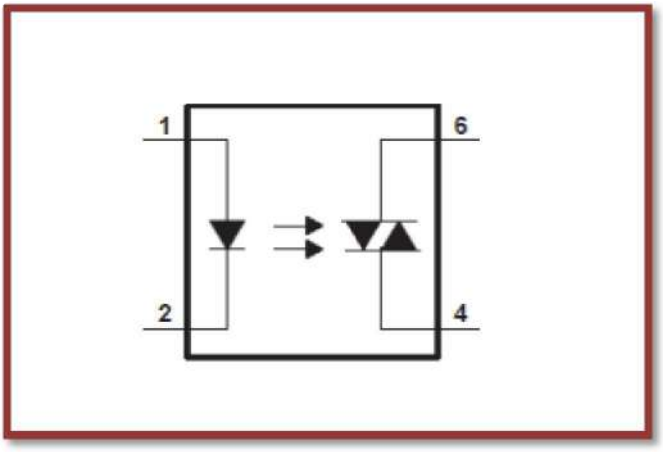


Figure II.19 : Symbole d'un optotriac.

Le MOC3023 est un optocoupleur de sortie triac. Ce dispositif contient une diode à émission infrarouge et un commutateur activé par la lumière qui fonctionne comme un triac. Ce dispositif est conçu pour servir d'interface entre des commandes électroniques et des triacs de puissance, le principe de fonctionnement d l'optotriac est simple, à chaque passage du zéro de la sinusoïde détecter entre les pattes 4 et 6, et lorsque aucune tension est présente en entrée de l'optotriac, le triac (interne au MOC3023) ne sera pas amorcé, dans ce cas la tension du réseau se retrouve au borne du triac qui lui est représenté comme un interrupteur ouvert.

Lorsqu'une tension sera présente sur l'entrée de l'optotriac, le triac va s'amorcé, la tension entre les pattes 4 et 6 étant quasiment nulle, le triac est représenté comme un interrupteur fermé.

Les caractéristiques de l'optotriac MOC3023 sont :

- Tension de blocage maximale de 400V.
- Tension d'isolement de 4.17KV.
- Mode de fonctionnement Triac de passage à zero.
- Plage de température de fonctionnement de -40 ° C à 85 ° C.
- Le courant de déclenchement de la LED est de 5 mA **[15].**

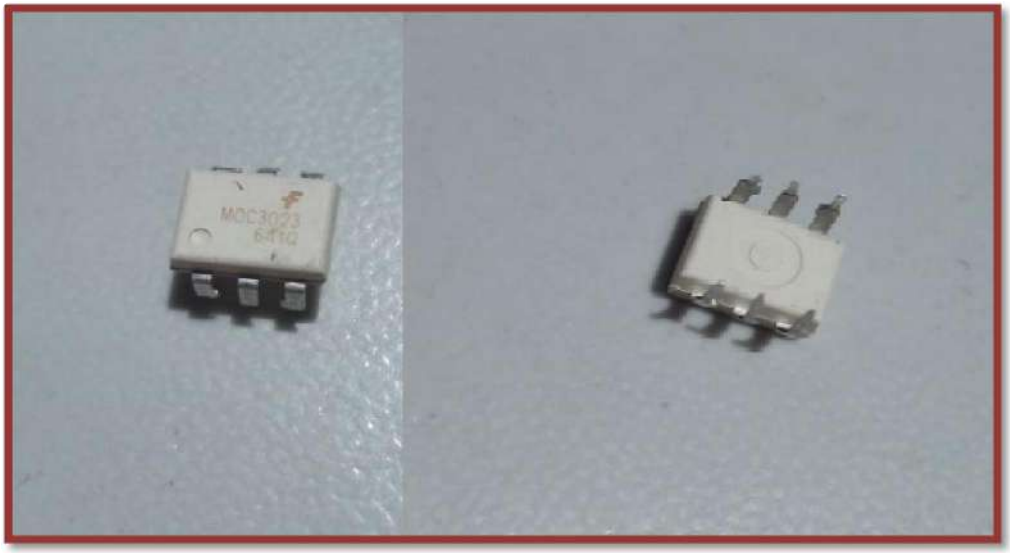


Figure II.20 : L'optotriac MOC3023.

6.3 La commande du triac en commutation :

Le principe consiste à mettre le triac en conduction avec un certain retard par rapport au début de la demi-alternance. Puis il se re-bloque à la fin de cette même demi alternance. On recommence à chaque demi-alternance. Pour réaliser ce type de commande il faut donc envoyer une impulsion de mise en conduction avec un certain temps de retard par rapport au passage par zéro de l'onde secteur.

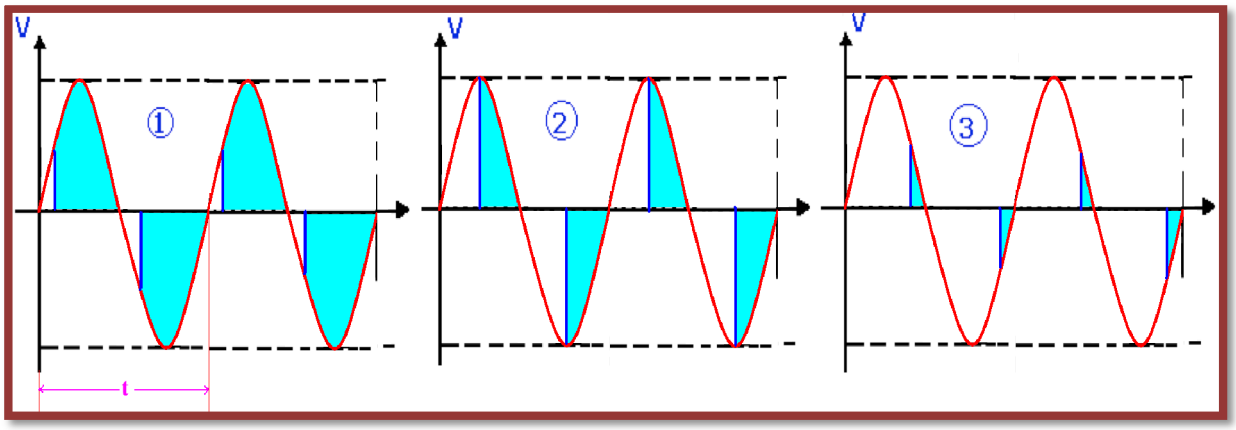


Figure II.21 : Temps de conduction du triac.

La figure II.21 montre trois cas de conductions différents pour trois valeurs de puissance différentes.

- Pour le premier cas, la charge électrique 220v est alimentée presque normalement. Le temps de conduction du TRIAC est presque égal à **t**. (Temps de conduction long partie en bleu).

- Pour le deuxième cas, la charge électrique 220v n'est alimentée que la moitié du temps durant chaque demi-alternance. (Temps de conduction moyen).
- Pour le deuxième cas, la charge électrique 220v n'est alimentée que pendant une fraction de temps durant chaque demi-alternance. (Temps de conduction court).

Notons bien que dans les Triacs, on agit sur le temps de conduction et non sur la valeur de la tension.

6.4 Le passage par le zéro secteur :

En courant alternatif, le passage à zéro est le point instantané sans tension. Dans une onde sinusoïdale ou une autre forme d'onde simple, cela se produit normalement deux fois par cycle. La détection de l'instant de passage à zéro de la tension de secteur est très importante pour la commande des triacs, il nous permet de détecter l'instant zéro pour calculer le temps de retard entre ce dernier et l'instant où on doit générer l'impulsion d'ouverture de triac **[16].**

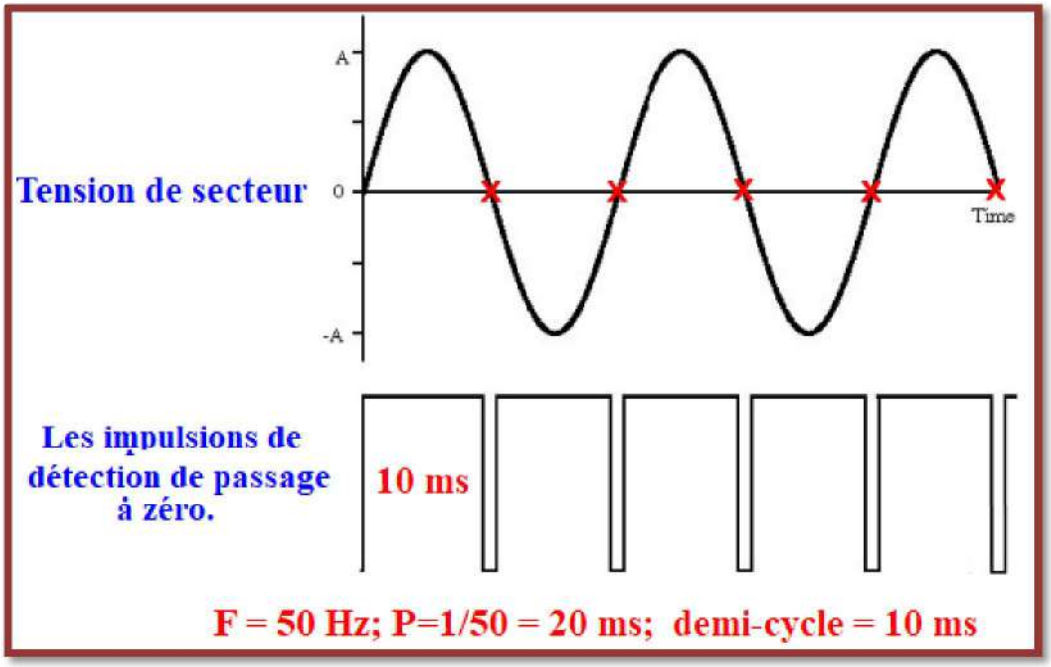


Figure II.22 : le passage par zéro secteur.

Ils existent différents schémas des circuits pour la détection de passage à zéro. Il existe des schémas à base des ampli op, des schémas a base des optocoupleurs et des schémas a base des optocoupleur avec pont de redressement dans le cas d'utilisation des microcontrôleurs. Ce dernier type est utilisé dans notre projet comme illustre dans la figure II.23.

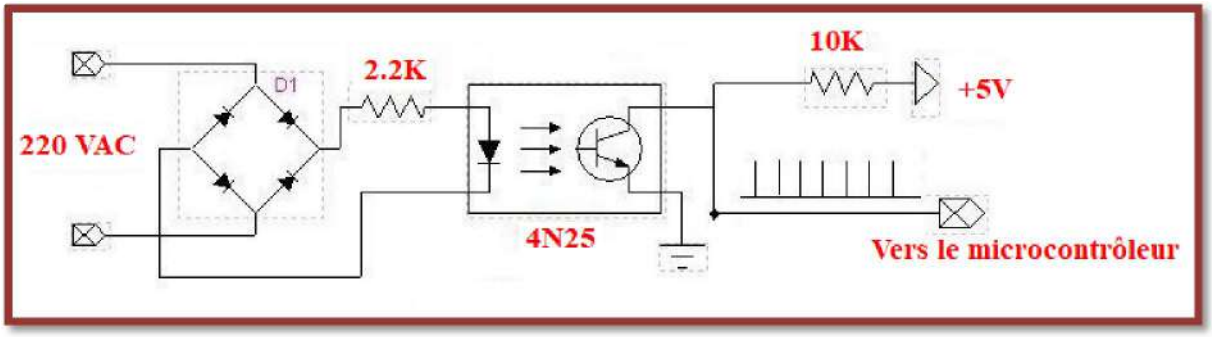


Figure II.23 : Détecteur de passage par zéro avec pont de diodes et l'optocoupleur 4N25.

6.5 L'optocoupleur 4N25 :

Nous avons besoin d'isoler la partie commande de la partie puissance. Pour cela, nous utilisons des optocoupleurs. Nous avons donc deux masses bien distinctes dans notre carte.

La première : la masse de 220v qui est la masse de puissance. Ainsi est la masse de commande et de l'alimentation 5V donc on a utilisé un optocoupleur model 4N25.

Le 4N25 est un coupleur de phototransistor standard à une voie avec une connexion de base. L'optocoupleur est constitué d'une LED infrarouge en arséniure de gallium et d'un phototransistor silicium NPN.

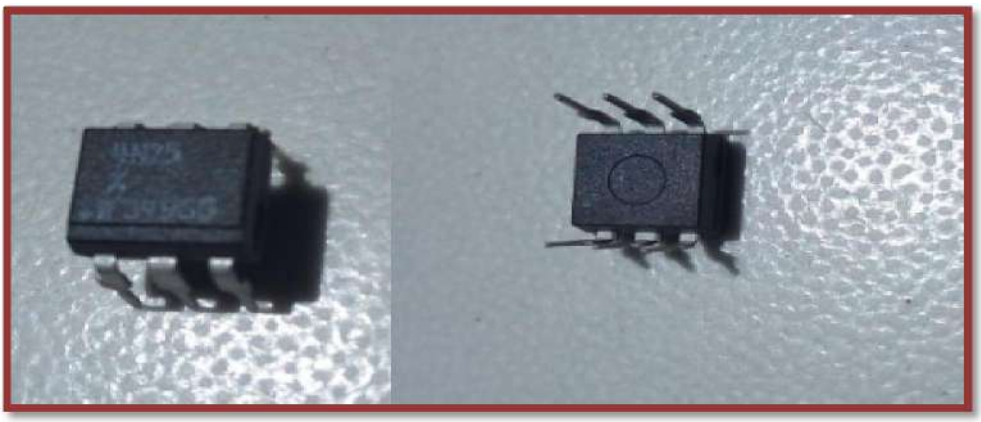


Figure II.24 : optocoupleur 4N25.

Caractéristique de l'optocoupleur 4N25 :

- Tension de test d'isolation de 5000VRMS.
- S'interface avec les familles logiques les plus communes.
- Capacité de couplage entrée-sortie <0.5pF.
- Commutation plus rapide que les coupleurs standards **[17].**

7. Conception de la carte électronique :

Afin de réaliser la carte électronique de notre projet on utilise le logiciel de conception Eagle (Easily Applicable Graphical Layout Editor) qui est un logiciel propriétaire de conception assistée par ordinateur de circuits imprimés.

7.1 Présentation générales de Eagle :

Il est fort populaire auprès des amateurs parce qu'il existe une version de démonstration forte utilisable. Multiplateforme, ce logiciel existe pour Microsoft Windows, Linux et Mac OS X. Il comprend un éditeur de layout, de routage, un routeur automatique et une librairie extensible de composants.

7.1.1 Etapes de conception :

Ce schéma résume l'ensemble des étapes de conception d'une carte électronique

- Elaboration du schematic.
- Elaboration du typon avec chevelu pour une cohérence totale entre schéma et typon.
- Routage automatique possible.

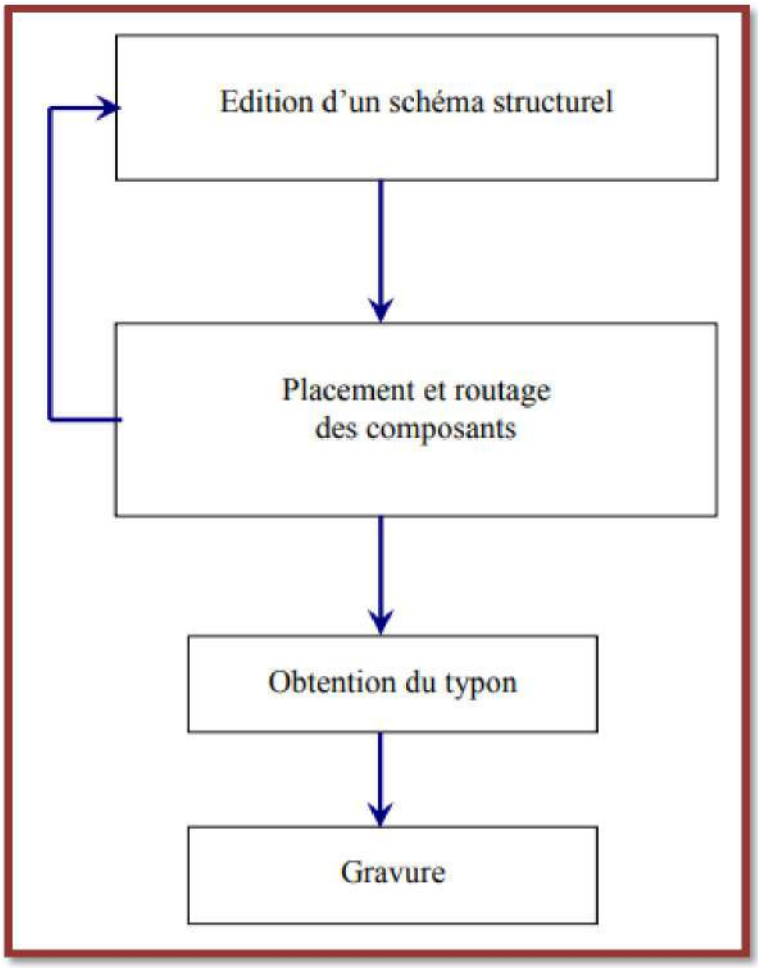


Figure II.25: Etapes de conception.

7.1.2 Travail sous forme de projets :

Eagle oblige les utilisateurs à travailler sous forme de projets ce qui permet :

- De choisir les bibliothèques de composants à utiliser.
- De regrouper tous les circuits qui appartiennent au même thème de travail.

Le projet sera créé dans c:\Users\Admin\Document\eagle\New_Project_2 Les librairies utilisées dans le projet se trouvent toutes dans EAGLE\lbr répertoire par défaut d'Eagle

Pour créer un nouveau projet dont le nom est contrôleur d'incubateur.

1-Aller dans File.

2- Choisir New puis project.

3- Donner le nom que vous souhaitez à votre nouveau projet (ici: contrôleur d'incubateur).

4- Cliquer sur <ok> le nouveau projet est créé.

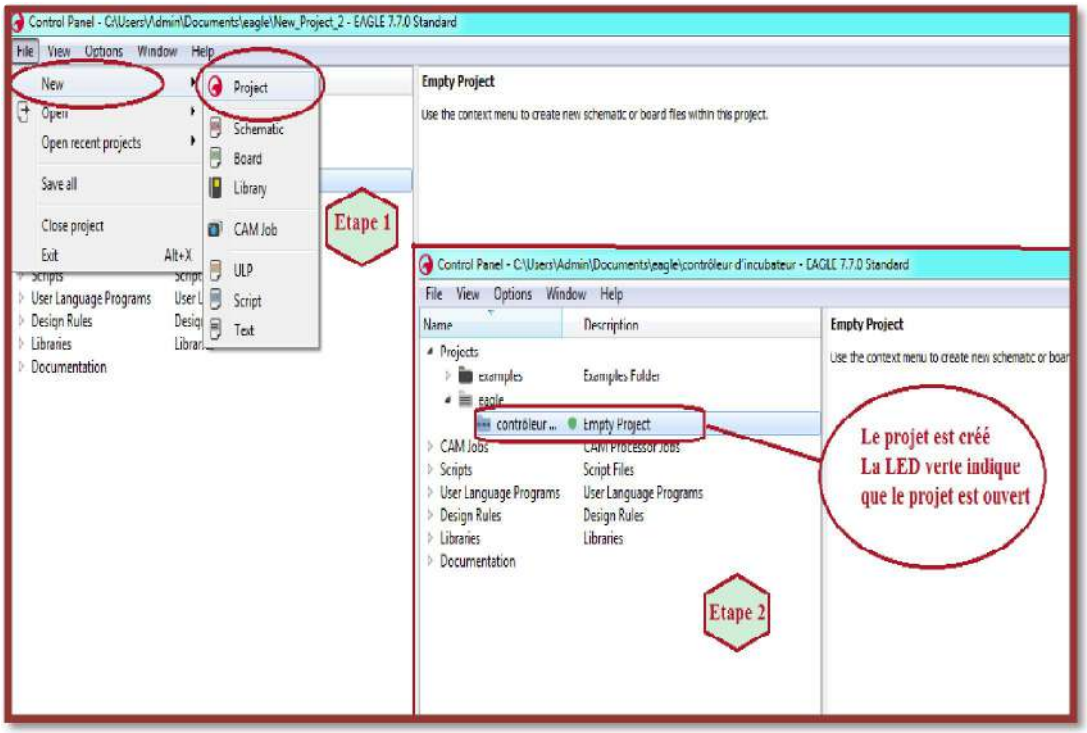


Figure II.26: Création d'un projet.

7.1.3 Elaboration du schéma structurel (schematic) :

Dans le Control Panel, aller dans File, puis New, schematic. La fenêtre suivante apparaît :

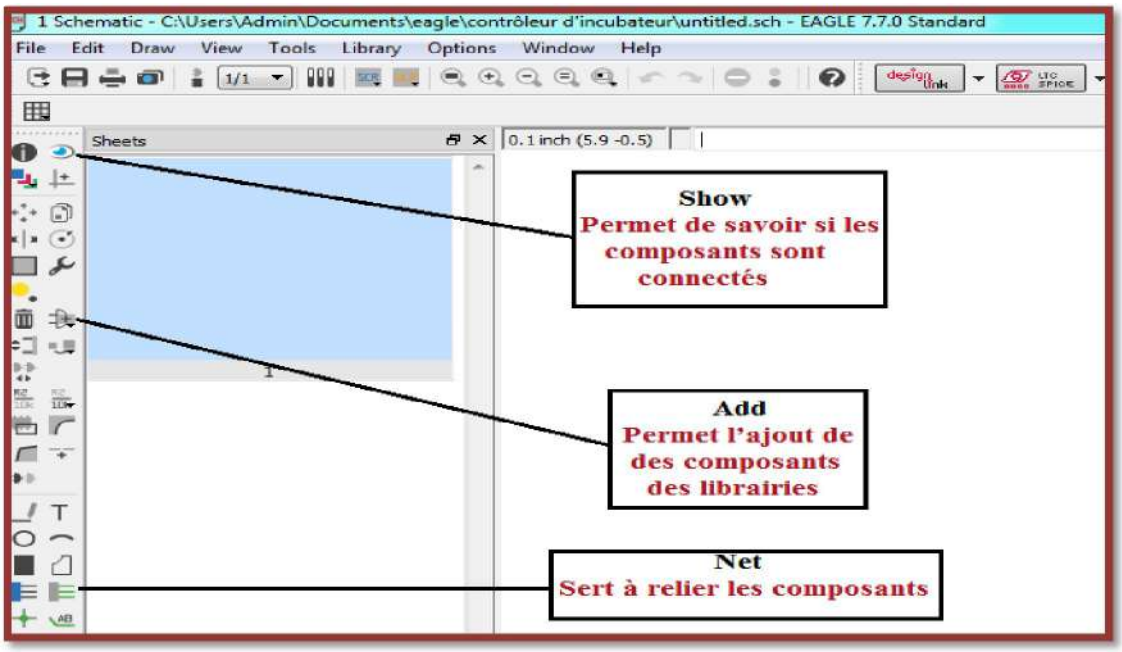


Figure II.27: Création du schéma structurel.

C'est à l'intérieur de cette fenêtre que l'on va dessiner le schéma structurel à l'aide des bibliothèques library qui contiennent les composants device. Il suffira de les relier grâce au bouton Net. La fonction associée aux différents boutons est indiquée par le logiciel en venant pointer sur ceux-ci.

7.1.4 Choix des composants :

• Utiliser des composants que vous pouvez acheter et souder. Ne pas hésiter à acheter les composants avant pour vérifier leur taille, brochage (connecteurs, notamment),...

• Bien lire les datasheets et tester tous les montages avant sur une plaque d'essais pour éviter tous les problèmes auxquels on ne pense pas.

• Utiliser de préférence des composants "standardisés" : toujours les mêmes connecteurs, les mêmes valeurs de condensateurs, résistances, sauf besoin particulier **[18].**

8. Création schémas des circuits :

Maintenant que le schéma structurel est réalisé, il reste à créer le typon en cohérence totale avec ce schéma. Dans la fenêtre schematic, cliquez sur le bouton switch to board . Il permet de générer automatiquement une platine (board) sur laquelle sont « jetés » les composants constitutifs du schéma.

Dans notre mémoire on a utilisé le microcontrôleur ATmega328 pour l'acquisition de la température, de l'humidité, du taux de CO2 et pour commander les diffèrent actionneur qui contrôle l'incubateur. On a devisé notre travail en trois schémas (circuits) :

- Circuit principal de commande qui contient le microcontrôleur et l'étage de puissance.

- Circuit interface utilisateur qui contient l'afficheur LCD et les boutons poussoirs pour changer les paramètres internes (T_{max}, T_{min}, H_{max}, H_{min}…).
- Un circuit séparé pour implémenter les trois capteurs et qui doit être relie a la carte de commande via 5 fils. Ce dernier circuit doit être placé dans l'incubateur.

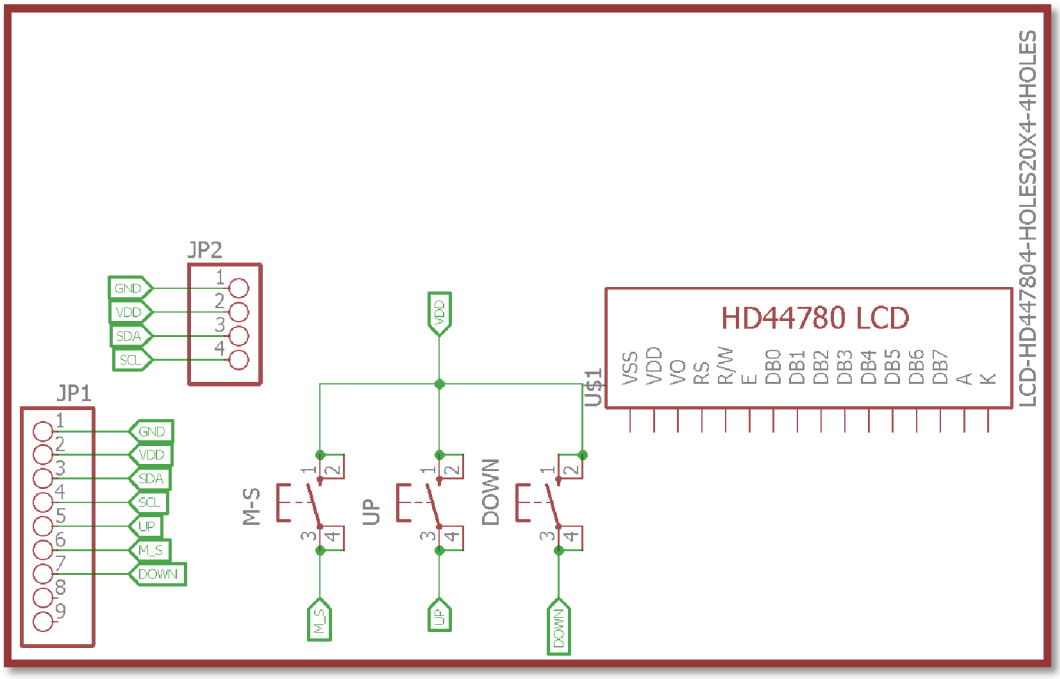


Figure II.28: Schéma de Circuit interface utilisateur.

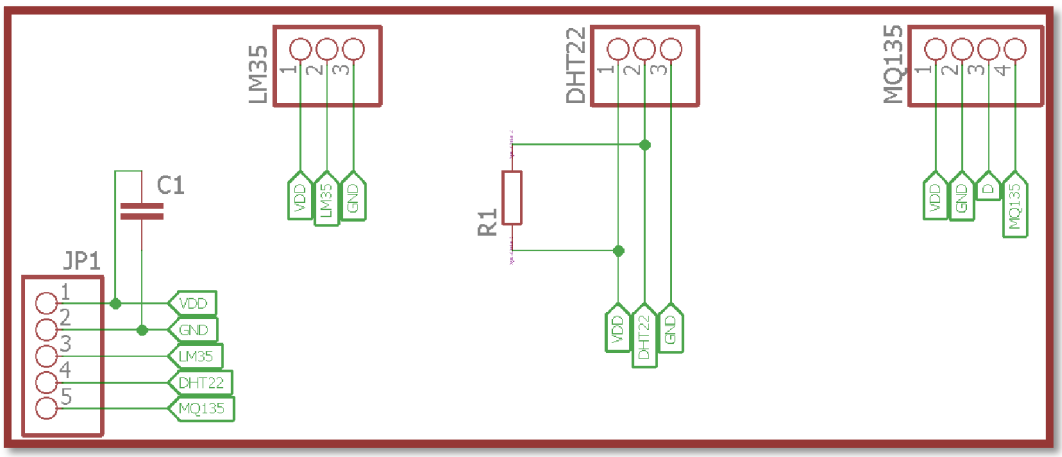


Figure II.29: Schéma de circuit des capteurs.

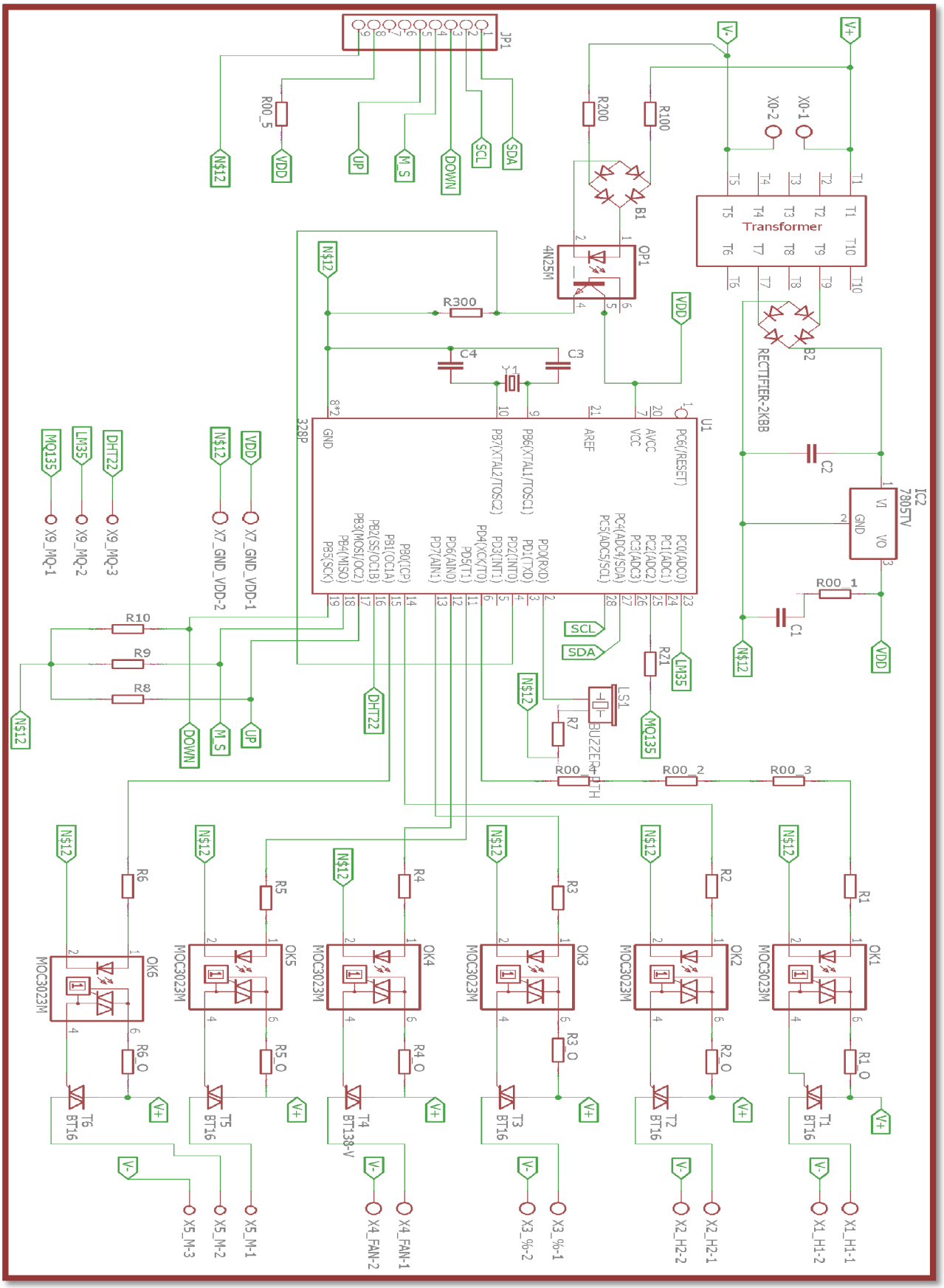


Figure II.30: Schéma de Circuit principal de commande.

8.1 Routage (Création du typon) :

Le schéma structurel est maintenant réalisé, après utilisation du routage automatique suive de quelques modifications manuelles on a obtenu les typons des figures (II.31), (II.32) et (II.33).

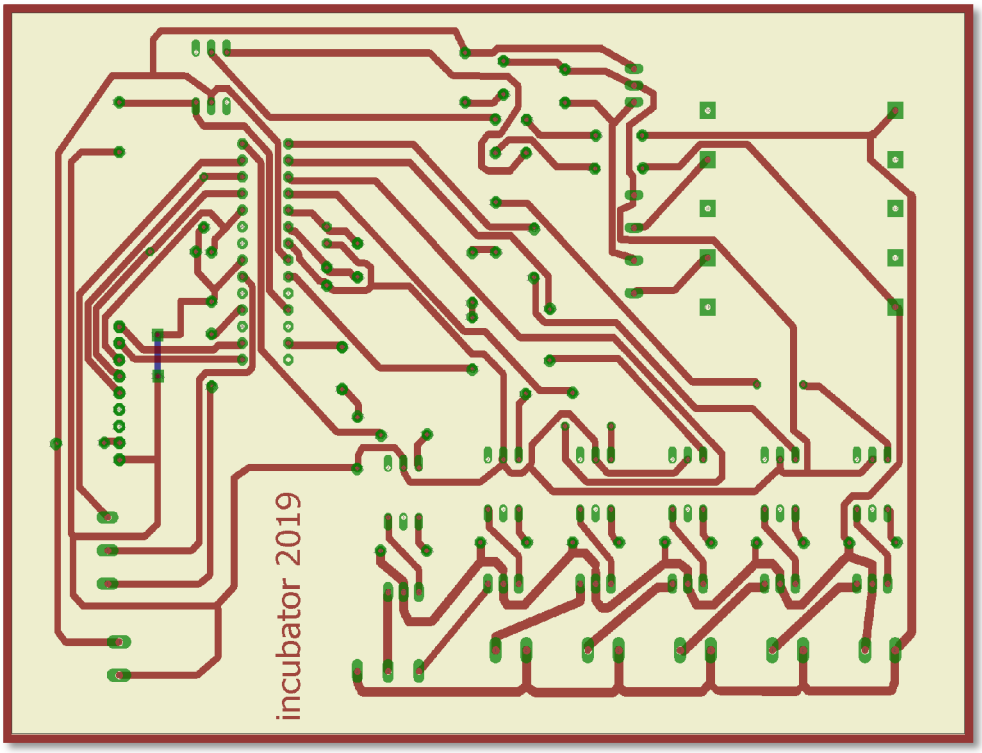


Figure II.31: Typon de Circuit principal de commande.

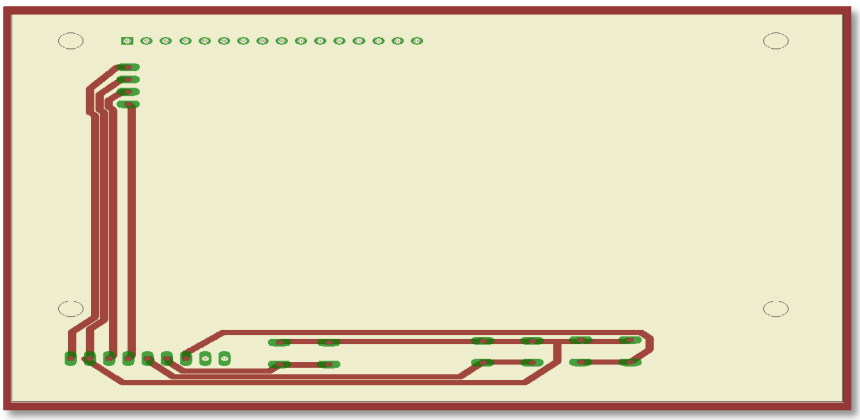


Figure II.32: Typon de Circuit interface utilisateur.

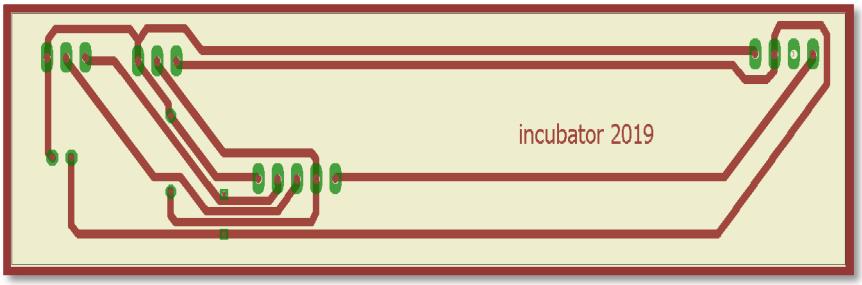


Figure II.33: Typon de circuit des capteurs.

9. Conclusion :

Dans ce chapitre nous avons illustré la partie électronique de notre contrôleur d'incubateur. Une explication détaillée du rôle de chaque composant et de ses caractéristiques techniques a était donnée. La programmation de cette carte électronique avec des organigrammes détaillés de chaque tâche du contrôleur est le but du chapitre suivant.

Chapitre III

La programmation du contrôleur d’incubateur

1. Introduction:

Après avoir mentionné dans le chapitre précédent les composants électroniques, le schéma et le circuit du notre contrôleur des incubateurs artificielle des œufs, on va consacrer ce chapitre à l'aspect software du contrôleur. Dans cette partie nous illustreront la programmation du microcontrôleur Atmega328 qui est le cœur du contrôleur, et afin de simplifier sa programmation, nous avons l'utiliser comme une carte Arduino Uno programmée à base d'Arduino IDE, ce qui nous permet d'utiliser les bibliothèques Arduino. à la fin, on place seulement l'ATmega328 dans le circuit de l'incubateur sans besoin de la carte Arduino Uno.

2. L'environnement de la programmation de l'arduino :

Le logiciel de programmation des cartes Arduino sert d'éditer le code en un langage proche du langage C. Une fois, le programme tapé et vérifié, l'environnement nous permet de transféré et le mémorisé dans la mémoire du programme du microcontrôleur de la carte via la liaison USB. Le câble USB alimente, à la fois, la carte et transporte aussi l'information entre le PC et la carte. L'environnement de développement des cartes Arduino est appelé Arduino IDE.

2.1 Structure générale du programme (IDE Arduino) :

Comme tous les langages de programmation, L'Arduino IDE possède une interface souple, simple est exécutable sur n'importe quel système.

L'interface de l'Arduino IDE consiste de :

- Une barre de menu.
- Une barre de boutons.
- Une Fenêtre d'édition du programme.
- Une zone de message des actions en cours.
- Une console d'affichage des messages de compilation.

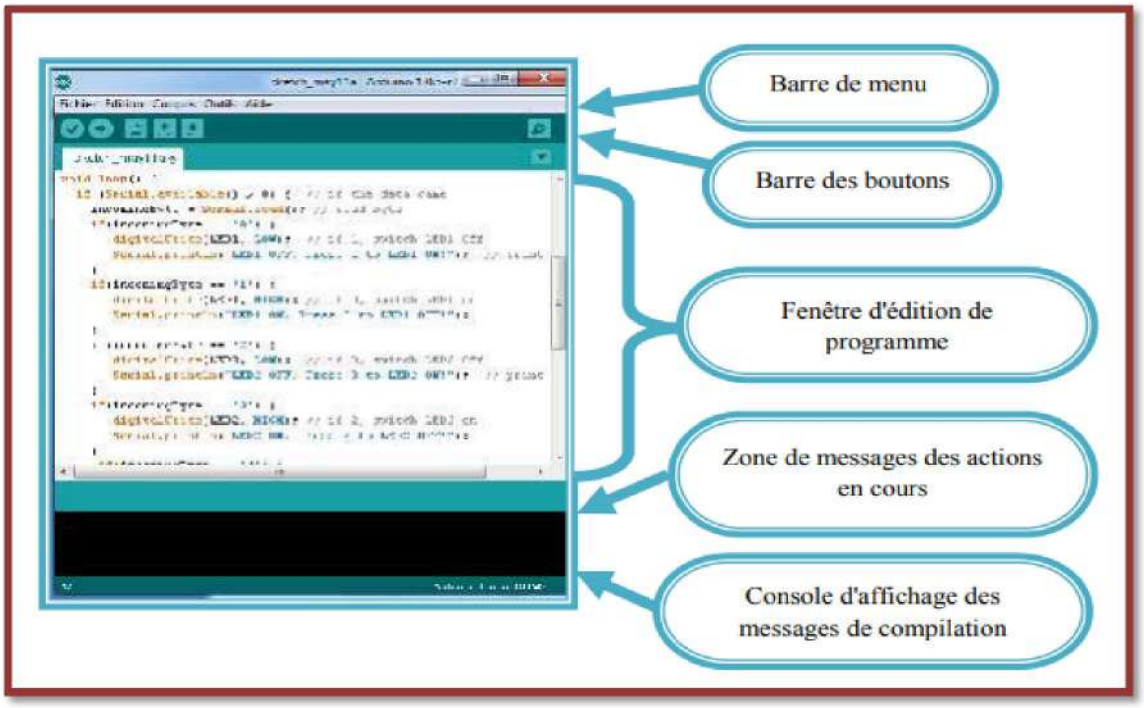


Figure III.1 : Interface IDE Arduino.

2.2 Transfert du programme :

Avant d'envoyer un programme dans la carte, il est nécessaire de sélectionner le type de la carte (Arduino UNO) et le numéro de port USB (Com 3) comme l'exemple de la figure III.2.

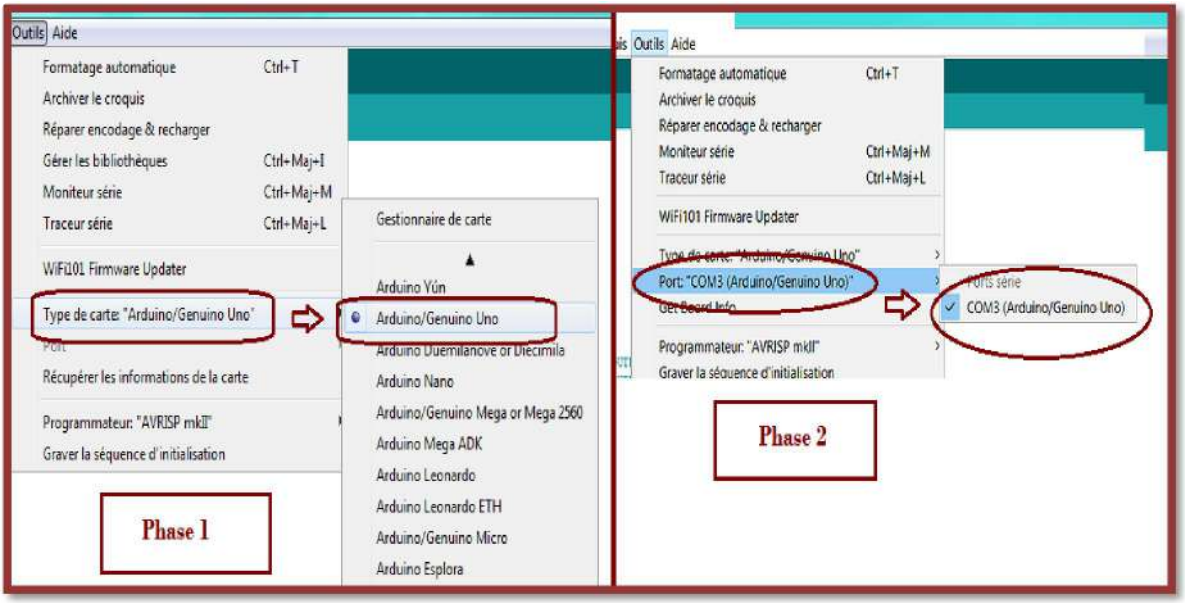


Figure III.2 : Paramétrage de la carte.

2.3 Description du programme :

Un programme Arduino est une suite d'instructions élémentaires sous forme textuelle (ligne par ligne). La carte lit puis effectue les instructions les unes après les autres dans l'ordre défini par les lignes de codes.

2.3.a Les commentaires :

Les commentaires sont, en programmation informatique, des portions du code source ignorées par le compilateur ou l'interpréteur, car ils ne sont pas censés influencer l'exécution du programme.

2.3.b Définition des variables :

Une variable est un moyen de nommer et de stocker une valeur pour une utilisation ultérieure par le programme, telle que des données provenant d'un capteur ou une valeur intermédiaire utilisée dans un calcul.

Avant de les utiliser, toutes les variables doivent être déclarées. Déclarer une variable signifie la définition de son type et éventuellement sa valeur initiale (initialisation de la variable). Les variables ne doivent pas nécessairement être initialisées (attribuer une valeur) lors de leur déclaration, mais cela est souvent utile.

2.3.c Configuration des entrées et des sorties void setup ()

Les broches numériques de l'arduino peuvent aussi bien être configurées en entrées numériques ou en sorties numériques, on peut configurer une pin en entrée; pinMode(nom, INPUT); ou en sortie pinMode(nom, OUTPUT).

2.3.d Programmation des interactions void loop():

Après avoir créé une fonction setup (), qui initialise et définit les valeurs initiales, la fonction loop () fait exactement ce que son nom suggère et boucle successivement, permettant ainsi au programme de contrôler la carte Arduino dans une boucle infinie.

2.4 Les étapes de téléchargement du programme :

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

1. On conçoit ou on ouvre un programme existant avec le logiciel Arduino IDE.
2. On vérifie ce programme avec le logiciel Arduino (compilation).
3. Si des erreurs sont signalées, on doit corriger le programme.
4. On charge le programme sur la carte.

5. On câble le montage électronique.
6. On alimente la carte soit par le port USB, soit par une source d'alimentation autonome
7. L'exécution du programme est automatique après redémarrage du microcontrôleur.
8. On vérifie que notre montage fonctionne **[19].**

3. Structure du programme du contrôleur d'incubateur d'œufs :

Le programme de ce contrôleur doit accomplir les tâches suivantes :

- L'acquisition des données concernant la température, l'humidité et le taux de co2 dans l'incubateur via les capteurs placés dans la barre des capteurs qui est reliée a la carte principale a travers un câble de 5 fils.
- Etre capable de communiquer avec la carte interface utilisateur pour afficher les donnes sur l'afficheur LCD et modifier, selon les commandes reçus via les boutons poussoirs, les valeurs consignes des températures, humidité ... etc
- Détecter les instants de passage par zéro de la tension du secteur.
- Commander les sorties 220v qui commande la résistance principale, la résistance secondaire, l'humidification, la ventilation et le moteur de retournement automatique des œufs.

3.1 Acquisition de l'humidité :

On a utilisé le capteur DHT22 qui produit un signal numérique étalonné, Il communique en série 1 fil avec un protocole spécial, et fournit une indication de température sur 16 bits et une indication d'humidité sur 16 bits également mais dans notre projet on a utilisé le DHT22 pour la mesure d'humidité seulement parce que on remarquer que le temps de réponse de la mesure de température est peu long, donc on a utilisé le LM35 qui à un temps de réponse plus court pour mesurer la température.

On peut redemander une mesure toute les deux secondes environ avec une fonction bloquante de 5ms.

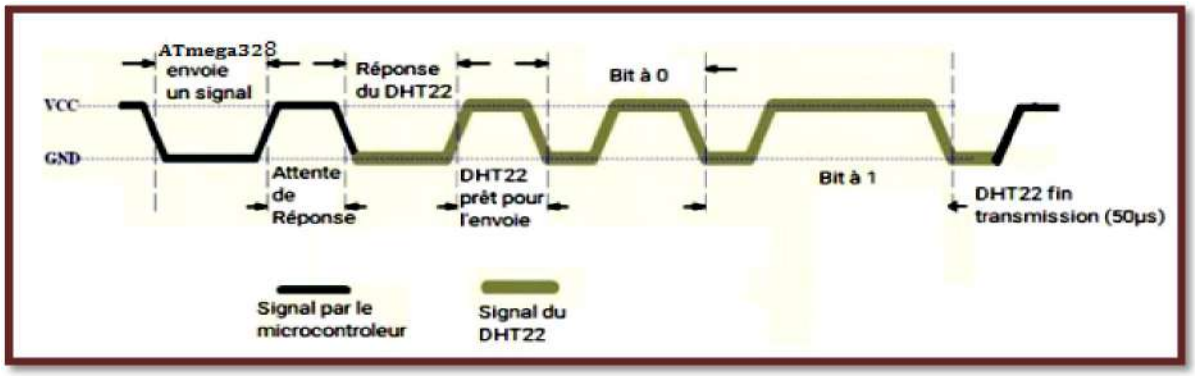


Figure III.3 : La trame de communication entre DHT22 et ATmega328.

La communication entre le capteur DHT22 et l'ATmega328 se fait en 2 étapes :

- Étape 1: le microcontrôleur ATmega328 envoie le signal de démarrage au DHT22

L'état libre du bus de données est le niveau de tension élevé. Lorsque la communication entre ATmega328 et DHT22 commence, programme de ATmega328 transformera le niveau de tension du bus de données de haut en bas et ce processus doit durer au moins 1 ms pour

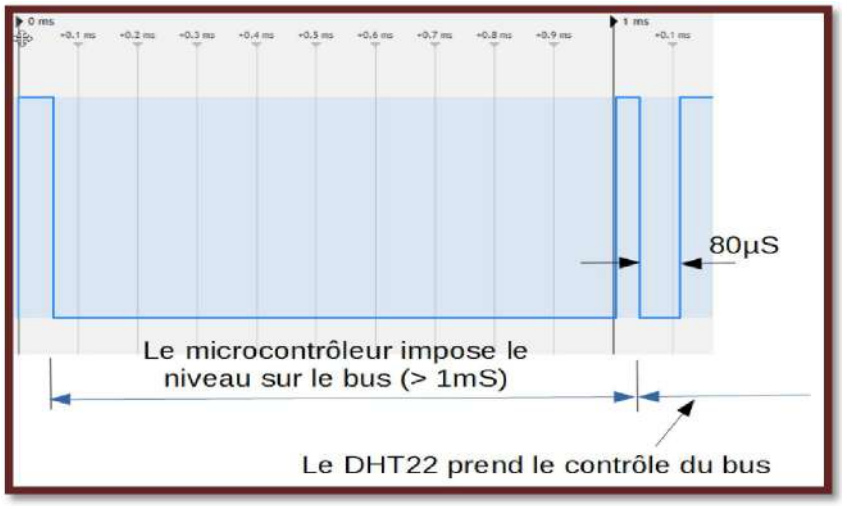


Figure III.4: ATmega envoi le signal au DHT22.

- Étape 2: DHT22 envoie un signal de réponse à l'ATmega328. Lorsque DHT22 détecte le signal de démarrage, DHT22 envoie un signal de niveau de tension bas et ce signal dure 80us comme signal de réponse, puis programme de DHT22 transformer le niveau de tension du bus de données de bas en haut et le dernier 80us pour la préparation de DHT22 à envoyer des données **[20].**

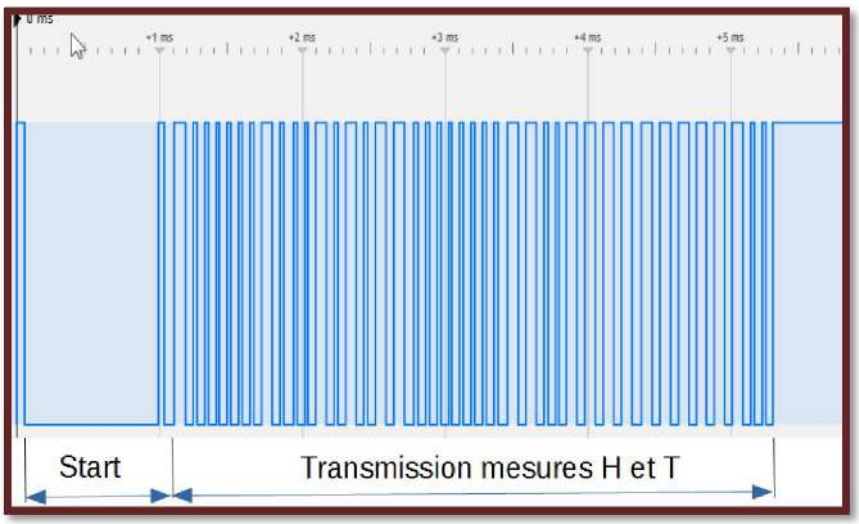


Figure III.5 : la réponse de dht22.

A chaque demande du microcontrôleur le circuit répond par un bit de présence et 40 bits de données suivant l'ordre :

a) 16 bits pour mesure humidité.

b) 16 bits pour mesure température.

c) 8 bits pour le CRC.

Exemple : 00000010 00001110 00000001 10010101 10100110

16 bits humidité*10 | 16 bits température*10 | CRC

Pour obtenir les valeurs d'humidité et de température il suffit de diviser par 10 les valeurs transmises par le circuit DHT22 soit dans notre exemple :

Humidité = 52,6 %.

Température = 40,5 °C.

Le CRC = 00000010 + 00001110 + 00000001 + 10010101 = 10100110

Dans notre programme nous avons utilisé la bibliothèque de DHT22 qui est déjà existée dans Arduino IDE.

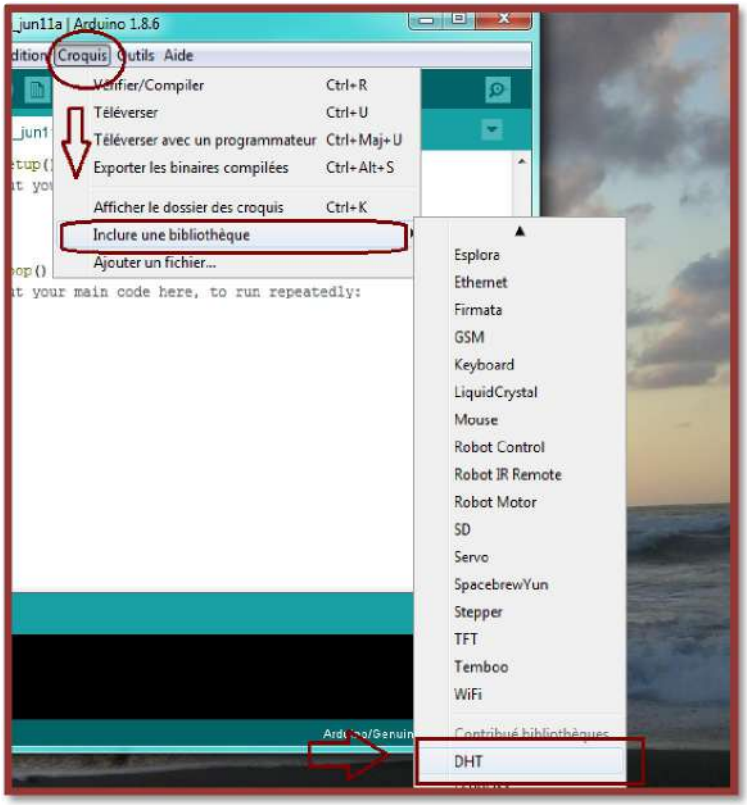


Figure III.6 : bibliothèque de DHT22.

Lorsqu'on a fait l'étape précédente, nous pouvons utiliser donc la bibliothèque de dht22

```
#include <DHT.h>
#define DHTPIN 10
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
H = dht.readHumidity();
```

3.2 Acquisition de température :

Comme mentionné précédemment, Le LM35 est un capteur de température analogique. Cela signifie que la sortie du LM35 est un signal analogique. L'ATmega328 est capable de lire directement les signaux analogiques de ses entrées analogiques grâce au convertisseur analogique numérique 10 bits qu'il possède.

Pour qu'un convertisseur analogique / numérique puisse faire ce travail, il a besoin d'une tension maximum de référence interne ou externe via le pin AREF.

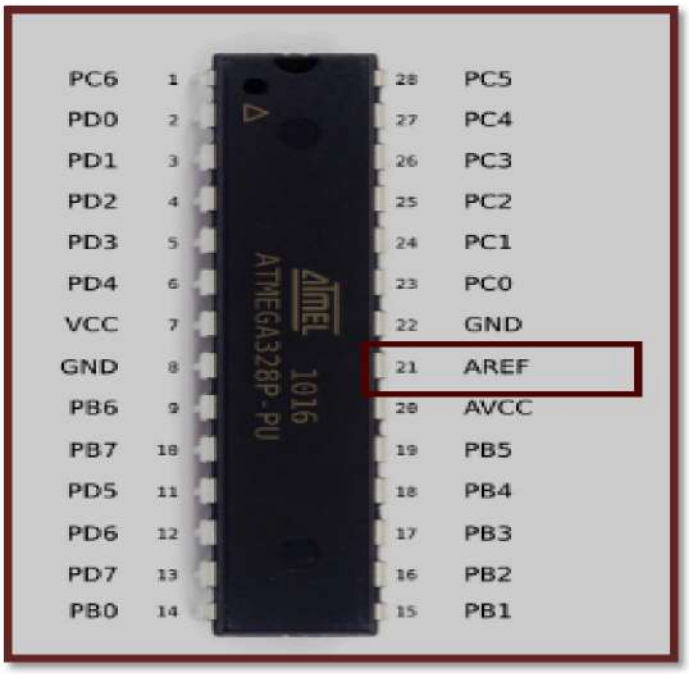


Figure III.7 : la tension AREF.

Pouvoir choisir cette tension est très intéressant, car cela permet de réduire la plage de tension de mesure et donc d'améliorer drastiquement la précision des mesures.

Si on mesure un signal analogique toujours compris entre 0 volt et 1,5 volt, comme c'est le cas avec un capteur LM35, il est dommage d'avoir une référence à 5v, car plus de la moitié des points de mesure ne sont jamais utilisés.

Dans notre cas la tension maximum est même de 1.1v (= 110°C), avoir une tension de référence à 5v est donc dramatique pour la précision de la mesure. Pour rappel, 5 volts / 1024 points (10 bits) = 4.88mV par point.

On peut aussi configurer la tension de référence utilisée pour l'entrée analogique (c'est-à-dire la valeur utilisée en haut de la plage d'entrée) a 1.1V par l'utilisation du paramètre INTERNEL de la fonction analogRefrence().

La fonction analogRead() permet de mesurer une tension sur une broche analogique() retourne un nombre entier (int) compris entre 0 et 1023 et accepte un paramètre obligatoire : le numéro de broche analogique à lire, ce paramètre prend la valeur A0 qui remplace la broche 23 dans notre ATmega328.

```
#define LM35 A0
 •
 •
 •
temperatur = analogRead(LM35);
temperatur =(temperatur *110)/(1023);
```

3.3 Acquisition de la qualité d'air :

Comme on a mentionné précédemment, on a utilisé MQ135 qui est capteur de la qualité de l'air, dans notre cas on voulait mesurer la qualité d'air (co2) dans l'incubateur d'œuf. Ce capteur utilise le dioxyde d'étain (sno2) dont la conductivité électrique varie en fonction de la présence de polluant. Ce dernier a une sortie analogique, la communication entre lui et L'ATmega328 est au niveau de la broche 25 qui remplace la broche A2 dans l'Arduino, se fait à l'aide d'un convertisseur analogique numérique de 10 bits il a le même principe de communication que le capteur lm35. Nous avons lire la tension sur la broche A2 et mesure cette tension avec analogRead().

```
#define MQ135 A2
co2 = analogRead(MQ135);
```

3.4 La communication avec la carte interface utilisateur :

Pour modifier les valeurs consignes on a besoin de 3 boutons : bouton M_S pour sélectionner le mode, et par conséquence, la grandeur à afficher (température et humidité…) et 2 boutons (UP et Down) qui permettre d'incrémenter ou décrémenter les valeurs.

Quand un bouton pousoir est enfoncé, il laisse passer le courant et l'Atmega reçoit +5V, il indique donc HIGH (ou 1). Si un bouton poussoir est libre, l'Atmega ne reçoit rien, donc c'est LOW (ou 0). L'Atmega peut lire une valeur de +5V ou de 0V. Afin d'éviter le cas d'une entrée flottante on a placé une résistance pull down entre l'entrée et la masse (GND) pour assurer que l'entrée est à 0 V lorsque le bouton poussoir est libre.

Pour une alimentation de 5 V, une bonne valeur pour R est d'au moins 4.7 Kilo-Ohm, le courant sera alors de 1.06 mA.

On va expliquer le déroulement des différentes séquences de programmation par des organigrammes.

Organigramme de bouton de M_S de la sélection de mode :

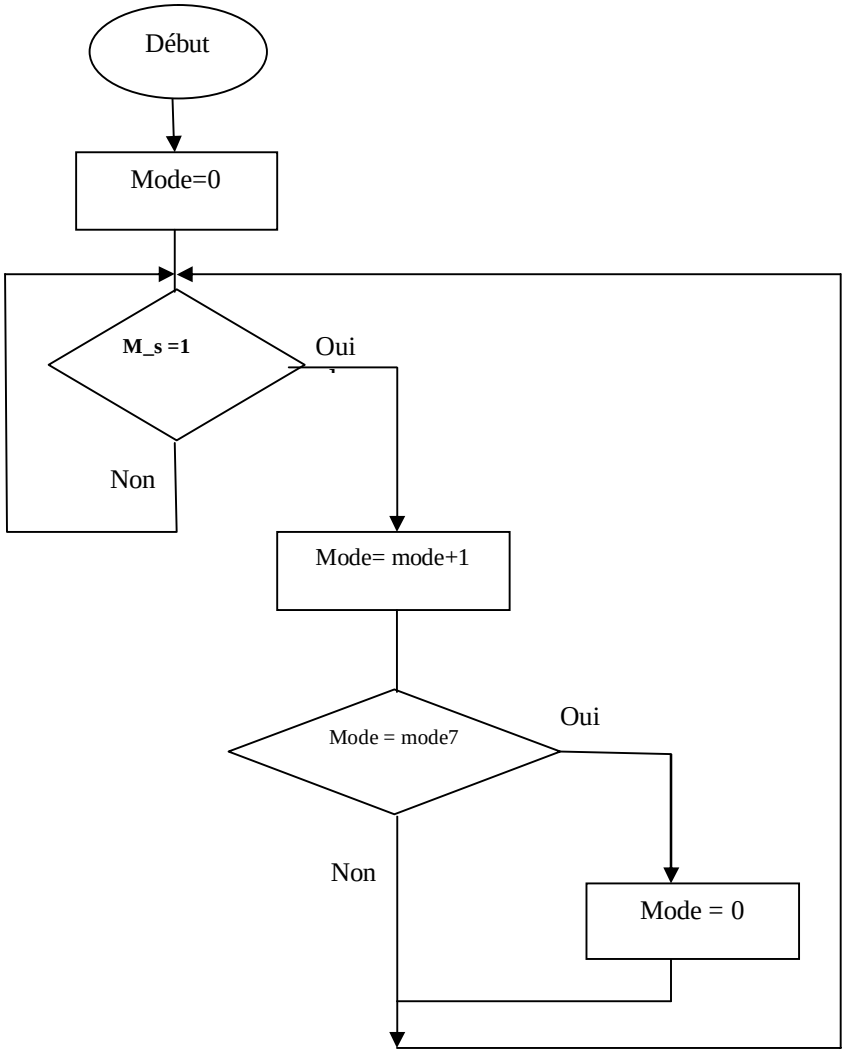


Figure III.8 : organigramme de sélection de mode.

La figure III.8 représente un organigramme des 6 modes qui constituent notre programme s'est les suivants :

Le mode_0 par défaut.

Le mode_1 et le mode 2 pour modifier les valeurs de température (min, max).

Le mode_3 et le mode 4 pour modifier les valeurs de l'humidité (min, max).

Le mode_5 pour modifier la valeur de co2 max.

Le mode_6 pour modifier la valeur la température consigne.

Le mode_0 est le mode initialisé par défaut. Chaque fois qu'on appuis sur le bouton de sélection de mode (M_s), le mode change du mode_I au mode_I+1 jusqu'au mode 6 et se retourne autre fois au mode_0.

Pour l'affichage de la température et l'humidité on a utilisé un afficheur LCD 20*4, il communique avec le microcontrôleur via le module I2C qui est bronché avec ATmega au niveau des broches 27 et 28 (SDA, SCL), la bibliothèque de Lcd_I2C est par défaut inclus dans l'installation de l'Arduino IDE.

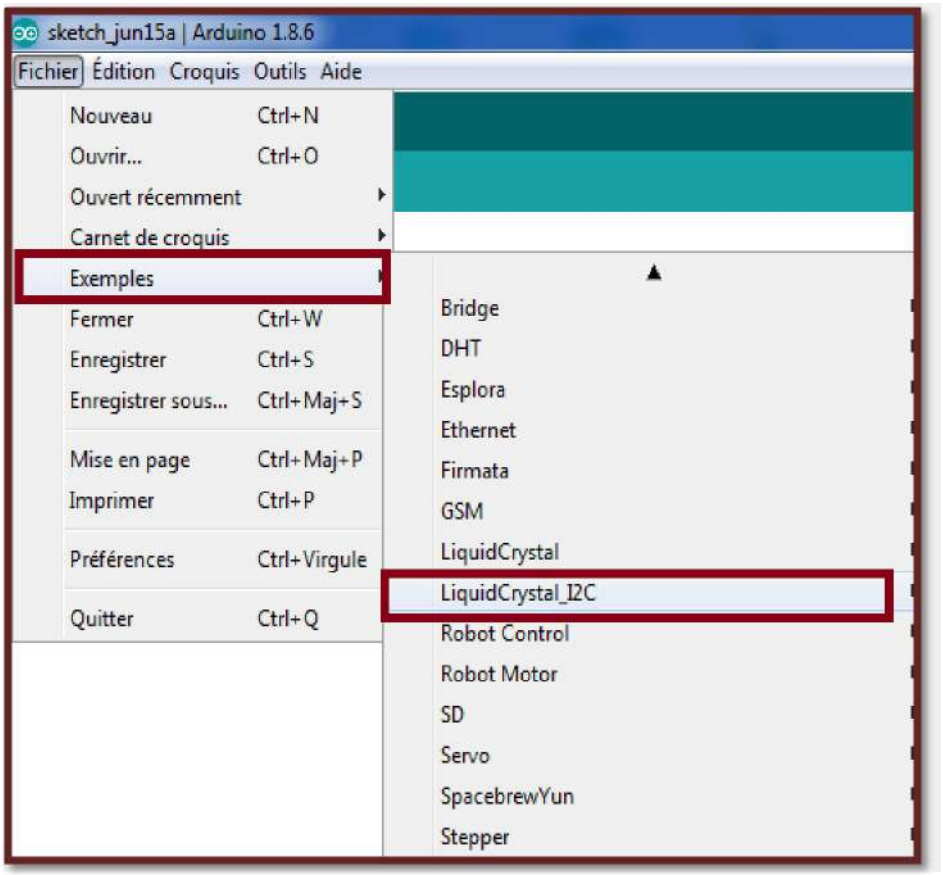


Figure III.9 : bibliothèque de l'afficheur LCD.

Organigramme de l'affichage sur le LCD :

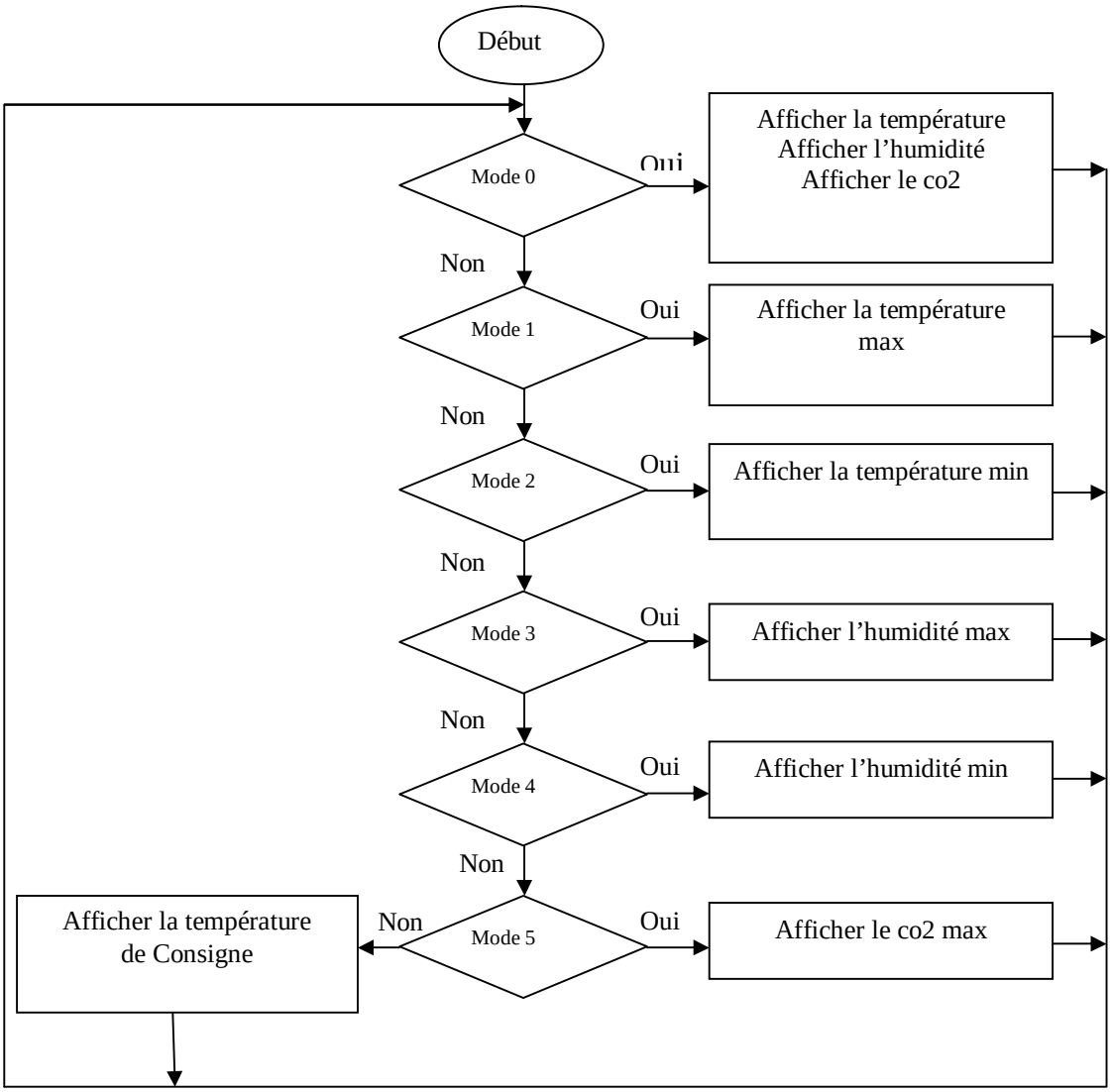


Figure III.10 : Organigramme de l'afficheur LCD.

L'organigramme de la figure III.10 explique comment afficher les grandeurs souhaitées sur un écran LCD de façon que a chaque fois on appuis sur le bouton "mode" on ajoute (+1) a la boucle, donc l'écran affiche le mode qui suit le mode précédent.

Organigramme de bouton de sélection up :

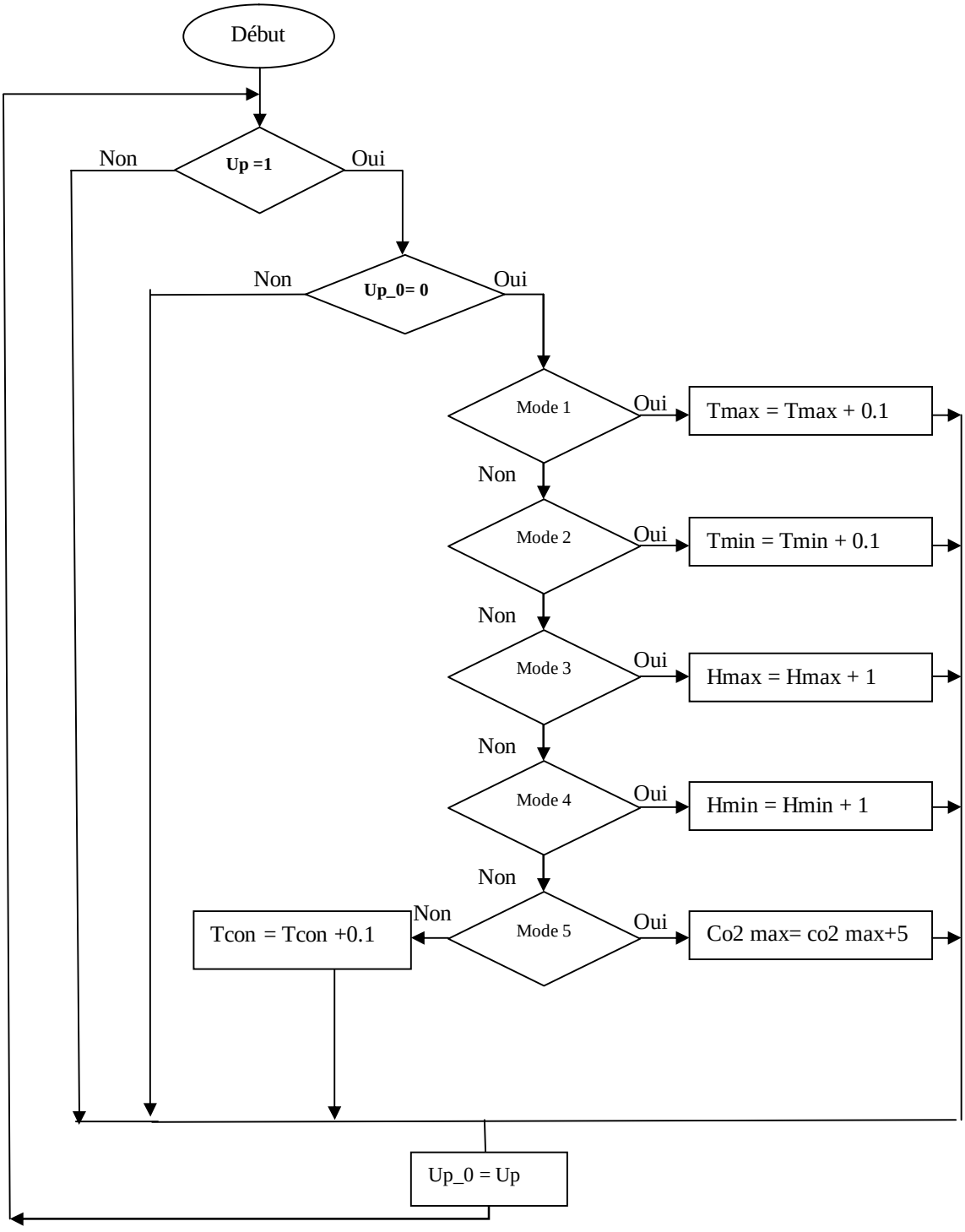


Figure III.11 : Organigramme de bouton de sélection UP.

La figure III.11 représente un organigramme qui nous montre comment marche la fonction "up", cette fonction a pour but d'incrémenté une des grandeurs qu'on souhaite modifier selon le mode sélectionne.

Organigramme de bouton de sélection down :

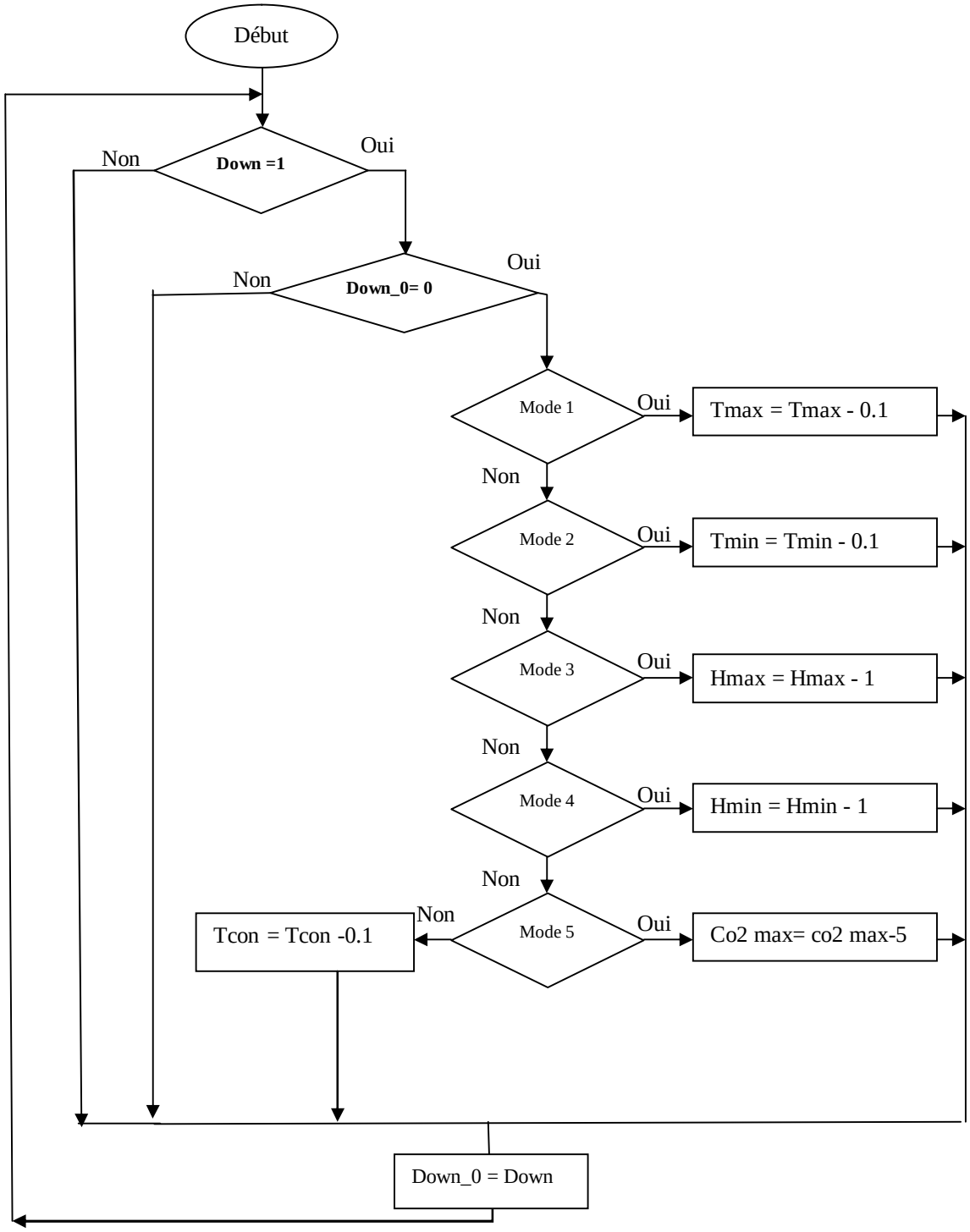


Figure III.12 : Organigramme de bouton de sélection DOWN.

L'organigramme de la figure III.12 montre la fonction inverse. La fonction " Down", qui a pour but de décrémenté une des grandeurs souhaitées selon le mode sélectionné.

Organigramme de programme complet :

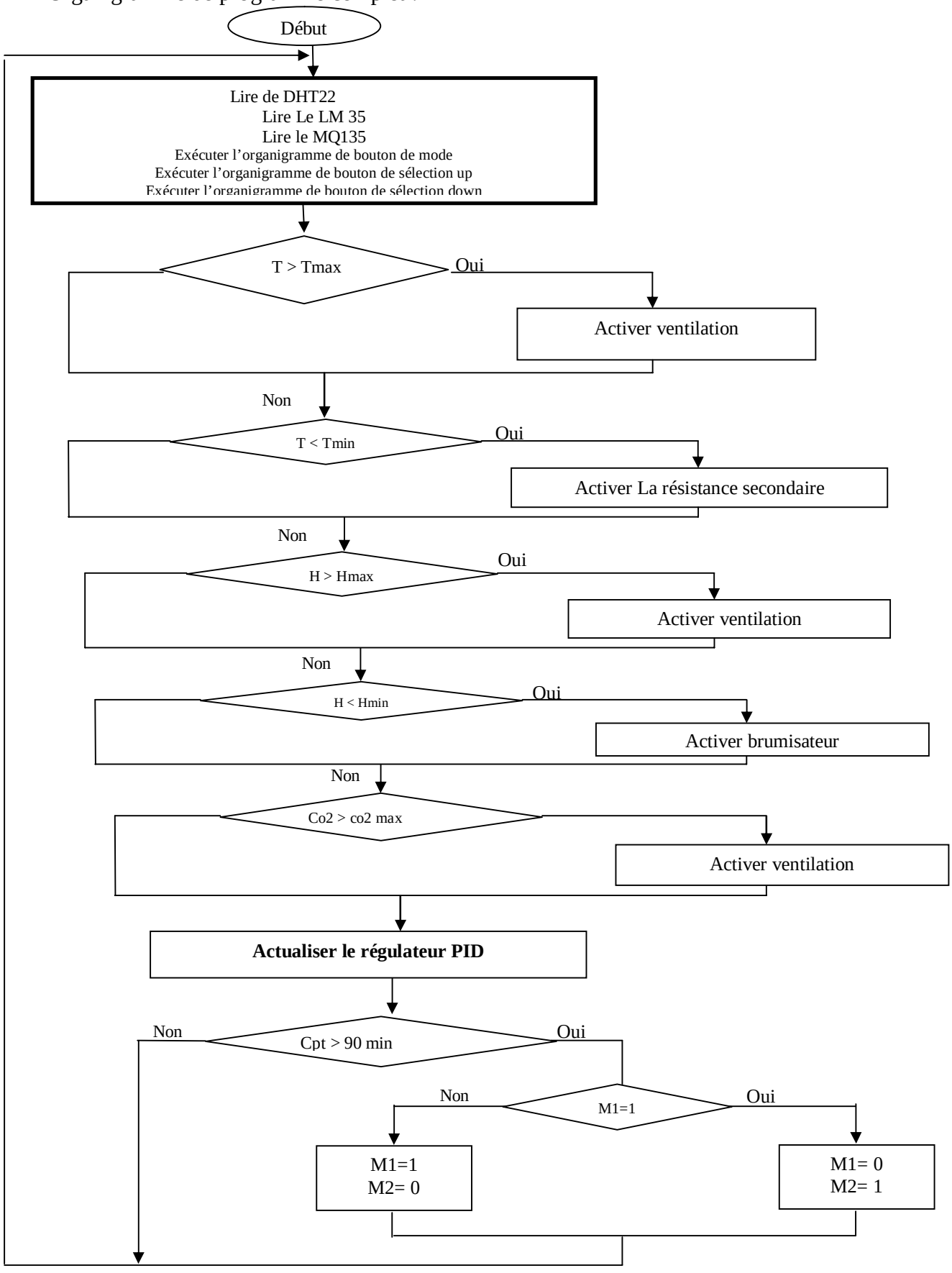


Figure III.13 : Organigramme de programme complet.

La figure III.13 exprime un organigramme de la boucle infinie de l'ATmega qui permet le maintien de la température, l'humidité, et niveau de CO2 dans les plages fixées par l'utilisateur. Les grandeurs fixées à des chiffres constants, en parle d'un régulateur, ce dernier teste les valeurs comme la température ; l'humidité et la compare avec des constants comme T max, T min ... par exemple, si la T> Tmax la ventilation s'active si T<Tmin ; la résistance s'active ainsi de suite. La machine contient deux moteurs qui dois inverser le positionnement des œufs, donc en teste le compteur si il atteint les 90 min les deux moteurs passe de la rotation 1 à 2 et 2 à 1 afin d'assurer le retournement des œufs.

4. La régulation et la commande des sorties 220v :

La tâche principale de ce contrôleur est de commander les triacs des sorties 220v pour maintenir les conditions nécessaires dans l'incubateur afin d'assurer une bonne incubation des œufs. Tous les sorties, (R2, l'humidification, la ventilation, et le moteur de retournement des œufs), sont tous commandées par une régulation tout ou rien. Et vu la sensibilité du processus d'incubation a la variation de température, la résistance de chauffage principale est commandée par une régulation PID.

4.1 La commande tout ou rien :

Le concept TOR (tout ou rien) se ramène au binaire : 0 ou 1. Cela signifie que l'information à traiter ne peut prendre que deux états (marche-arrêt). Seuls ces deux niveaux logiques sont possibles, le triac reste conducteur tant que la commande est active, il est alors comparable à un interrupteur qui est en position fermée. Lorsque on supprime la commande le triac se re-bloque à la fin de la demi alternance (annulation du courant).

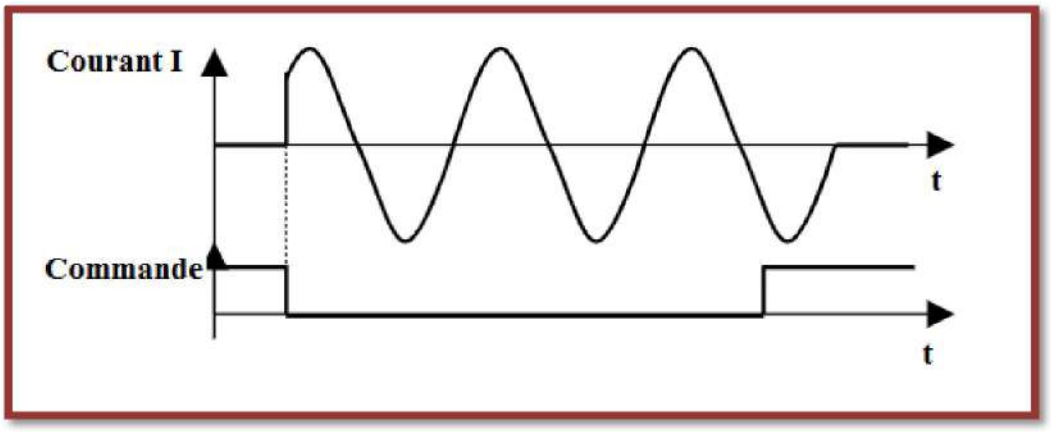


Figure III.14 : la commande en tout ou rien.

4.2. La commande PID :

Le réglage d'un PID consiste à déterminer les cœfficients Kp, Ki et Kd afin d'obtenir une réponse correcte du procédé et de la régulation. L'objectif est d'être robuste, rapide et précis. Il faut pour cela limiter les éventuels dépassements (overshoot).

- La robustesse est probablement le paramètre principal et délicat. On dit qu'un dispositif est robuste si la régulation fonctionne toujours même si le modèle change légèrement. Un régulateur doit être capable d'assurer sa tâche même avec ces changements pour s'adapter à des usages non prévus/testés (dérive de production, vieillissement mécanique, environnements extrêmes...).

- La rapidité du régulateur dépend du temps de montée et du temps d'établissement du régime stationnaire.
- Le critère de précision est basé sur l'erreur statique.

La réponse type d'un procédé stable est la suivante :

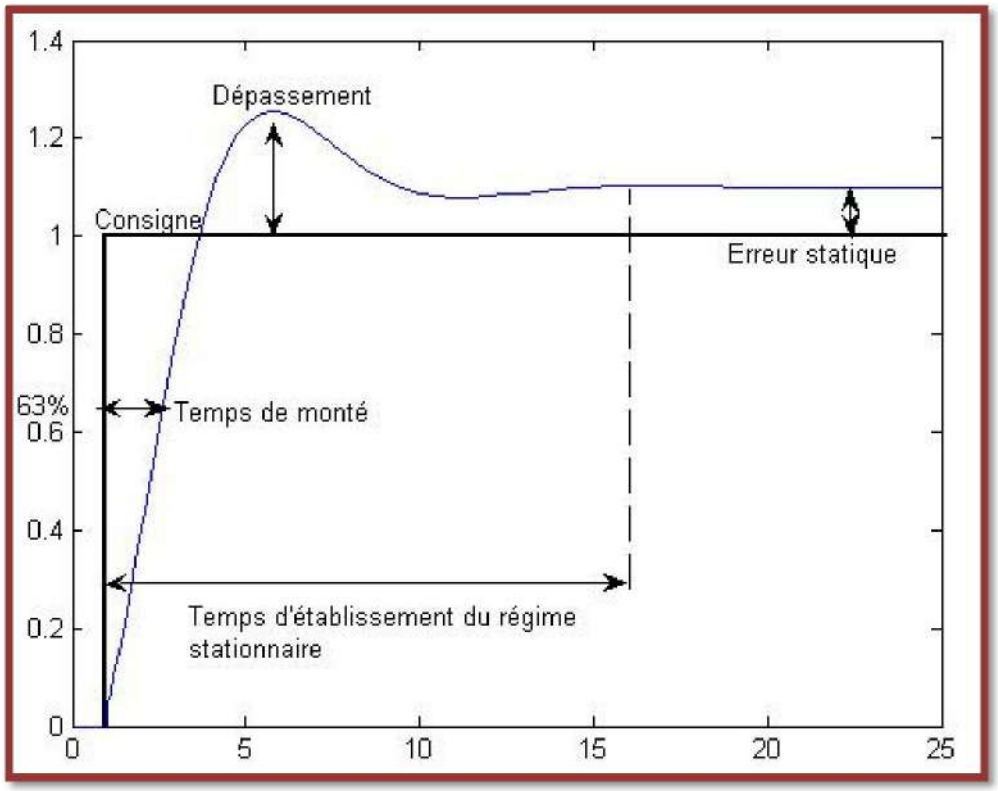


Figure III.15 : la régulation PID.

Dans cette partie, on va expliquer, en une simple manière, le principe de fonctionnement et l'utilisation d'une régulation PID, sans entrer dans les calculs mathématiques et sans utiliser les fonctions de transfert du système. Un asservissement PID consiste à mémoriser l'erreur, la somme des erreurs et la différence de l'erreur courante avec l'erreur précédente, puis générer une commande selon les valeurs calculées et les coefficients Kp,Ki et Kd du régulateur PID.

- Erreur = consigne – mesurée ;
- Somme_erreurs += erreur;
- Variation_erreur = erreur - erreur_précédente;
- Erreur_précédente = erreur.

4.2.a : Le régulateur proportionnel (P) :

La commande de ce régulateur est proportionnelle à l'erreur.

Commande = **Kp** * erreur.

Kp : est le coefficient de proportionnalité de l'erreur à régler de façon manuelle.

4.2.b Le régulateur proportionnel intégral (PI) :

La commande de ce régulateur est proportionnelle à l'erreur, mais aussi proportionnelle à l'intégrale de l'erreur. On rajoute donc à la commande générée par le régulateur proportionnel, la somme des erreurs commises au cours du temps.

Commande = **Kp** * erreur + **Ki** * somme_erreurs.

Ki : est le coefficient de proportionnalité de la somme des erreurs. Il faut aussi le régler de façon manuelle.

4.2.c Le régulateur proportionnel intégral dérivé (PID) :

La commande est à la fois proportionnelle à l'erreur, proportionnelle à la somme des erreurs et proportionnelle à la variation de l'erreur.

Commande = **Kp** * erreur + **Ki** * somme_erreurs + **Kd** * (erreur -erreur_précédente).

On doit donc faire une mesure sur notre système pour pouvoir calculer l'erreur et ainsi appliquer le PID. Cette mesure est à faire régulièrement à une certaine fréquence d'échantillonnage.

Puisque la température dans les couveuses varie lentement, on actualise la commande PID tous les x secondes, faire :

1. Erreur = **T**consigne – **T**mesurée ;

2. somme_erreurs += erreur;
3. variation_erreur = erreur - erreur_précédente;
4. $C._{PID}$ = **Kp** * erreur + **Ki** * somme_erreurs + **Kd** * variation_erreur;
5. erreur_précédente = erreur.

4.2.d Réglage des coefficients d'un PID :

Le réglage des coefficients Kp, Ki et Kd d'un PID peut se faire expérimentalement par essais/erreurs. Il y a plusieurs combinaisons possibles, et pour trouver un triplet performant relèverait de l'exploit. Il vaut mieux y aller par étape.

• Etape1 : il faut mettre en place un simple régulateur proportionnel (les coefficients Ki et Kd sont donc nuls). Par essais/erreurs, il faut régler le coefficient Kp afin d'améliorer le temps de réponse du système. C'est-à-dire qu'il faut trouver un Kp qui permette au système de se rapprocher très vite de la consigne tout en faisant attention de garder la stabilité du système : il ne faut pas que le système réponde très vite tout en oscillant beaucoup.

• Etape2 : Une fois ce coefficient réglé, on peut passer au coefficient Ki. Celui-là va permettre d'annuler l'erreur finale du système afin que celui-ci respecte exactement la consigne. Il faut donc régler Ki pour avoir une réponse exacte en peu de temps tout en essayant de minimiser les oscillations apportées par l'intégrateur **[21].**

• Etape3 : on peut passer au dernier coefficient Kd qui permet de rendre le système plus stable. Son réglage permet donc de diminuer les oscillations.

La commande PID nous permet de contrôler les impulsions de rapport cyclique pour qu'on puisse gérer le temps de retard d'un signal, cela nous autorise de commander l'énergie selon nos besoins.

Notre triac le BTA16 contrôle la tension alternative sur la sortie de la résistance principale R1. La première chose à faire est de détecter le passage par zéro, car notre impulsion doit être en phase avec la tension alternative. Donc, nous devons détecter le moment où la tension passe de positif à négatif ou de négatif à positif et synchroniser notre impulsion (avec un temps de retard calculé) avec celle-ci afin qu'elle se déclenche toujours au même endroit. Pour cela, nous avons utilisé un redresseur en pont complet.

Dans notre projet, la commande de rapport cyclique est gérée par deux interruptions. Une interruption interne et une autre externe, le brochage de l'interruption avec l'atmega328 est sur la broche 4.

- L'interruption externe permet la détection du passage par zéro, et sa routine d'interruption sert à reseter le timer TCNT2 (TCNT2=0).
- L'interruption interne : Le timer continue à compter, quand il aura la même valeur que le registre OCR2A, il génère une interruption. Et sa routine d'interruption sert à ouvrir le triac qui devient en mode conduction.

Le triac reste en mode conduction jusqu'à l'annulation du courant.

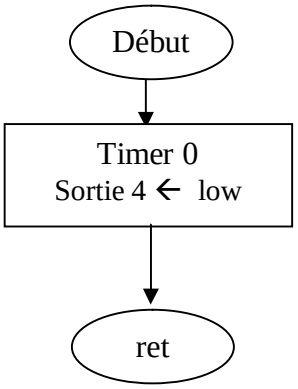


Figure III.16 : Organigramme de l'ISR de l'interruption externe.

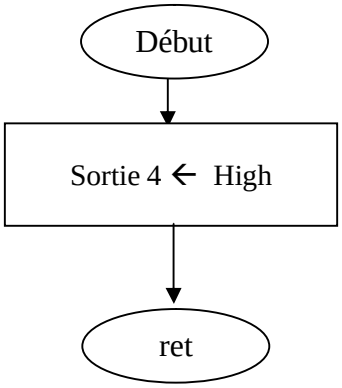


Figure III.17 : Organigramme de l'ISR de l'interruption interne.

On peut augmenter l'énergie qui passe à la résistance principale en plaçant des petites valeurs dans le registre OCR2A (temps de retard de la commande), par contre, on peut réduire l'énergie qui passe à la résistance principale en plaçant des petites valeurs dans le registre OCR2A.

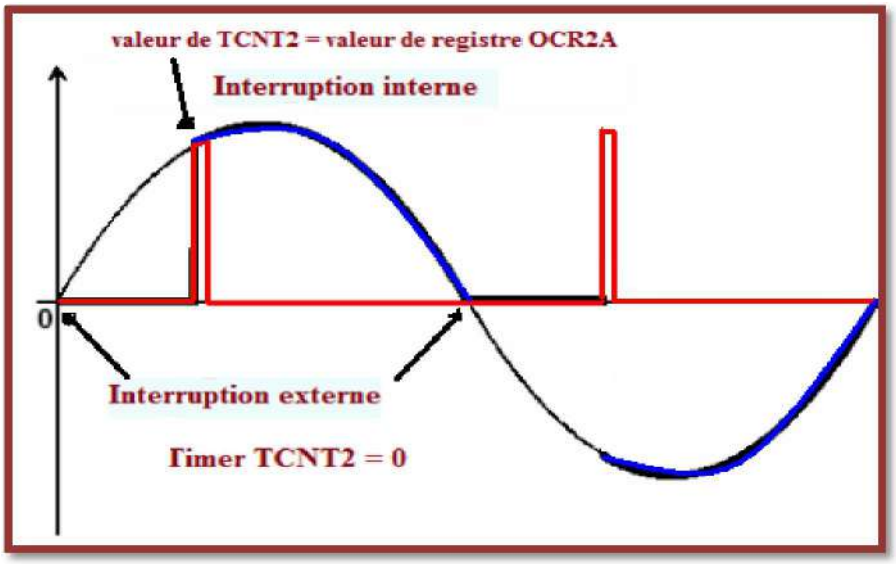


Figure III.18 : Graphe des interruptions.

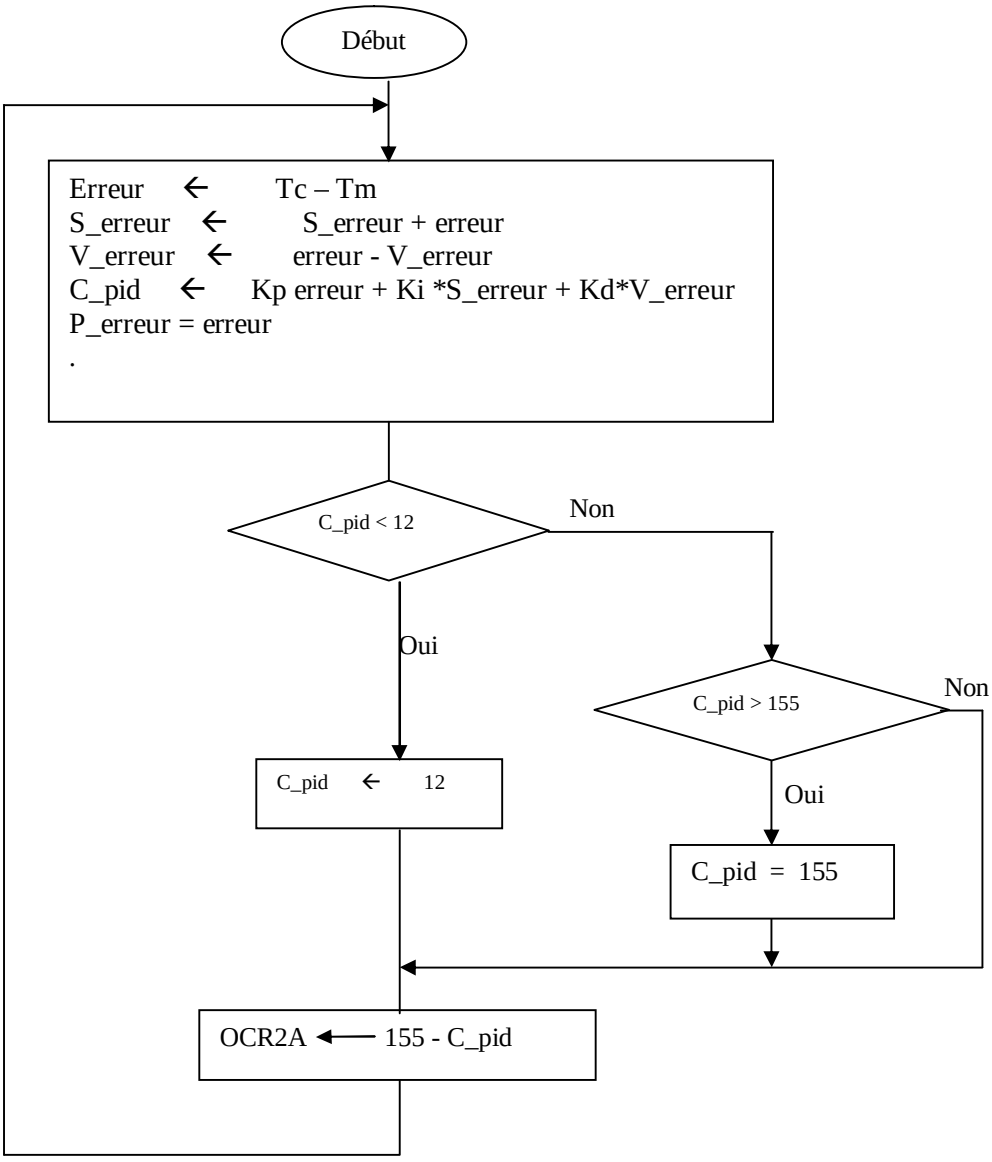


Figure III.19 : Organigramme de la commande PID.

L'organigramme de la figure III.19 indique la commande PID : si on a choisi une valeur de commande pid (qui sera placer dans OCR2A) inferieur a 12 l'énergie est maximale et lorsque on a choisi une valeur supérieure à 155 l'énergie est à zéro.
Remarque : les valeurs de 12 et 155 d'OCR2A sont trouvées expérimentalement.

5. Conclusion :

La programmation de l'Atmega328 comme la carte Arduino Uno nous a permet de simplifier la parties programmation et de bénéficier des bibliothèques préexistantes de l'Arduino IDE pour gérer la communication entre le microcontrôleur et l'afficheur LCD via le bus I2C, il nous a permet aussi de rendre la communication avec le capteur DHT22 plus simple. Le contrôleur réalisé est capable de piloter et ajuster tous les paramètres du control des incubateurs selon nos besoins, afin d'obtenir de meilleurs résultats d'une incubation artificiel.

CONCLUSION GENERALE

Conclusion générale :

L'invention de l'incubateur artificiel a permis à ses utilisateurs de faire incuber au moment choisi par eux, et avec le nombre des œufs qu'ils veulent couver, des dizaines jusqu'à centaines d'œufs selon le modèle d'incubateur. Afin d'obtenir un nombre d'œufs relativement important, il faut garantir le bon fonctionnement de l'incubateur, et cela dépend du réglage des grandeurs nécessaires afin d'avoir les meilleurs résultats possibles. En résumé l'incubateur idéal doit répondre aux critères suivants :

- Contrôle et régulation de la température.
- Contrôle et régulation de l'humidité.
- Retournement automatique des œufs a fréquence réglable.

Le contrôleur d'incubateur réalisé est fiable et adaptable à la plupart des incubateurs automatiques existants. L'utilisation des triacs de puissances (16A) assure la robustesse de ce contrôleur. La programmation à base d'Arduino IDE de l'Atmega328 nous permet d'exploiter le maximum possible les bibliothèques Arduino existants qui a reflétée sur la simplicité de la programmation et qui nous a permet d'atteindre les objectifs planifiés au début de ce travail.

Ce travail nous a donné la chance d'avoir un contact direct avec les composants électroniques, à savoir la carte Arduino Uno, les microcontrôleurs, les afficheurs LCD, l'adaptateur I2C, les optocoupleurs, les optotriac, les triacs etc… l'apport aussi des rectificatifs et des modifications entre la simulation et le fonctionnement réelle. Ce travail nous a donné aussi la chance de pratiquer la programmation en langage C, de maîtriser et d'utiliser des logiciels de compilation et de simulation (Eagle, Arduino IDE, ISIS…).

Nous espérons que les travaux que nous avons réalisés seront utiles pour le domaine agricole en Algérie et utilisés, avec des améliorations, pour réaliser des incubateurs industriels.

Réference bibligraphique

Références bibliographiques :

[1] *http://aviculture85.e-monsite.com/pages/les-couveuses.html. consulté* le 1 juillet 2019.

[2] ELHADI LAB Sidahmed, FEKIRI Khadija« Etude et réalisation d'un incubateur automatique a œufs » memoire de master Université de Ain-Temouchent, année 2015/2016.

[3] Edouard Hansets « De l'œuf à l'autruchon L'incubation des œufs d'autruche »
Les presses agronomiques de Gembloux, Année 2013.

[4] Yoda Scydou « Etude des performances d'une couveuse solaire dans l'incubation des œufs de poule » memoire de diplôme université polytechnique de Bobo-dioulasso, Juine 2011.

[5] NDJECK LAURENT, TCHINDA JOEL CONSTANTIN, BIDJOGO MARIE MICHELE « Etude et réalisation d'une couveuse d'œuf électrique ».

[6] Nico van Wageningen,Johan Meinderts,Puck Bonnier,Henk Kasper« Hatching eggs by hens or in an incubator» Agrodok 34,Année 2004.

[7] Nico van Wageningen,Johan Meinderts,Puck Bonnier,Henk Kasper **«** L'incubation des œufs par les poules et en couveuse » Série-Agrodok No. 34, Annéé2000.

[8] ISTIA « Carte ARDUINO UNO Microcontrôleur ATMega328 **»** Microcontrôleurs EI3 AGI 2016-2017.

[9] HAMID HAMOUCHI « Conception et réalisation d'une centrale embarquée de la domotique Smart Home » memoire de master Université Mohammed V, Rabat Morocco,Juillet 2015.

[10] Temperature and humidity module AM2302 Product Manual
https://akizukidenshi.com/download/ds/aosong/AM2302.pdf consulté le: 10/05/2019

[11] TERTAG Fatima et FERNANE Kheira « Réalisation d'un thermomètre électronique »MASTER en GENIE BIOMEDICAL, Université Abou Bakr Belkaïd de Tlemcen, juin 2016.

[12] MQ-135 GAS SENSOR DATASHEET *https://www.datasheet4u.com/datasheet-pdf/Hanwei/MQ-135/pdf.php consulté le: 15/05/2019*

[13] Moez HAJJI « Support de cours électronique analogique» ISET DE NABEUL (2014).

[14] Datassent de bta16 *http://www.alldatasheet.com consulté le: 15/05/2019*

[15] *http://www.soloelectronica.net/PDF/moc3020.pdf consulté le: 15/05/2019*

[16] *https://en.wikipedia.org/wiki/Zero_crossing*. Consulté en juin 2019.

[17] Texas Instruments Datasheet de 4n25
http://www.alldatasheet.com/view.jsp?Searchword=4N25 Consulté en juin 2019.

[18] Club RoboInsat « Logiciel Eagle 4.11 Initiation au routage » 2007/2008.

[19] KRAMA Abdelbasset, GOUGUI Abdelmoumen « Etude et réalisation d'une carte de contrôle par Arduino via le système Androïde » memoire de master UNIVERSITE KASDI MERBAH OUARGLA,Année 2014/2015.

[20] *https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf* Consulté en juin 2019

[21] Cours Implémenter un PID sans calcul Classe de terminale SI.

ANNEXE

Annexe A : FABRICATION DE CONTROLEUR D4INCUBATEUR





Annexe B : LE PROGRAMME D'UN CONTROLEUR D4INCUBATEUR.

```
#include <DHT.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// Set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x3F, 16, 2);
#define DHTPIN 10
#define LM35 A0
#define MQ135 A2
#define R1 4
#define R2 8
#define Humidity 7
#define Fan 6
#define M1 5
#define M2 9
#define alarm 0
#define DHTTYPE DHT22
const byte interruptPin = 2;
DHT dht(DHTPIN, DHTTYPE);
float temperatur=0;
unsigned long cpt=0;
unsigned long cpt_M=0;
float H = 80;
float T = 42;
float T_c = 37.5;
float T_max = 40;
float T_min = 36;
float H_max = 60;
float H_min = 55;
float CO2_max = 900;
int M_s = 0;
int Up =0;
int Down = 0;
int M_s_0 = 0;
int Up_0 = 0;
int Down_0 = 0;
int MODE = 0;
int co2 = 0;
```

```
38. int M=0;
39. float Erreur =0;
40. float S_erreur =0;
41. float V_erreur=0;
42. float p_erreur=0;
43. float Kp=5;
44. float Ki=2;
45. float Kd=2;
46. float C_pid=0;
47. int C_p=0;
48. unsigned long time0 = 0;
49. void setup()
50. {
51. SREG = SREG | B10000000;
52. TIMSK2 = (TIMSK2 & B11111001) | 0x06;
53. TCCR2B = (TCCR2B & B11111000) | 0x07;
54.  //OCR2A = 150;//min=12; max=155;
                        // OCR2B = OCR2A+5;
55. OCR2A = 75;//min=12; max=155;
56. OCR2B = OCR2A+5;
57. pinMode(interruptPin, INPUT);
58. pinMode(M_s, INPUT);
59. pinMode(Up, INPUT);
60.  pinMode(Down, INPUT);
61. pinMode(R1, OUTPUT);
62. pinMode(R2, OUTPUT);
63. pinMode(Humidity, OUTPUT);
64. pinMode(Fan, OUTPUT);
65. pinMode(M1, OUTPUT);
66. pinMode(M2, OUTPUT);
67. pinMode(alarm, OUTPUT);
68. analogReference(INTERNAL);
69. lcd.begin();
70. lcd.backlight();
71. dht.begin();
72. digitalWrite(R1,LOW);
73. lcd.clear();
74. digitalWrite(M1,LOW) ;
75. digitalWrite(M2,HIGH);
76. }
```

```
77. void loop() {
78. M_s = digitalRead(12);
79. Up = digitalRead(11);
80. Down = digitalRead(13);
81.  if (millis()- cpt  >2000)
82.   {
83.    cpt = millis();
84.  temperatur = analogRead(LM35);
85.  temperatur =(temperatur *110)/(1023);
86.  analogReference(DEFAULT);
87. H = dht.readHumidity();
88.  T = dht.readTemperature(false);
89.    co2 = analogRead(MQ135);
90.  analogReference(INTERNAL);
91. Erreur = T_c - temperatur ;
92.      S_erreur =S_erreur + Erreur ;
93.      V_erreur=Erreur - p_erreur ;
94.      p_erreur=Erreur;
95.      C_pid = (Kp*Erreur) + (Ki *S_erreur) + (Kd*V_erreur);
96.       C_p = round(C_pid);
97.       if(C_p < 0) {C_p = 0;};
98.       if(C_p > 155) {C_p = 155;};

99.  detachInterrupt(digitalPinToInterrupt(interruptPin));
100.          noInterrupts();
101.                OCR2A =155 - C_p  ;//min=12; max=155;
102.               if(OCR2A < 12) {OCR2A = 12;};
103.               OCR2B = OCR2A+5;
104.                delay(3000);
105.              interrupts();
106.            attachInterrupt(digitalPinToInterrupt(interruptPin), IISR, FALLING);
107.    }


108.    digitalWrite(Fan , LOW);
109.    if (temperatur > T_max){digitalWrite(Fan , HIGH);}
110.    if (temperatur < T_min){digitalWrite(R2 , HIGH);} else {digitalWrite(R2 , LOW);}
111.    if (H > H_max){digitalWrite(Fan , HIGH);}
112. if (H < H_min){digitalWrite(Humidity , HIGH);} else {digitalWrite(Humidity  , LOW);}
113.  if (H > H_max){digitalWrite(Fan , HIGH);}
```

```
114.  if (co2 > CO2_max){digitalWrite(Fan , HIGH);}
115. digitalWrite(alarm , LOW);
116. if (temperatur > T_max+1){digitalWrite(alarm , HIGH);}
117. if (temperatur < T_min-1){digitalWrite(alarm, HIGH);}
118. if (H > H_max+10){digitalWrite(alarm , HIGH);}
119. if (H < H_min-10){digitalWrite(alarm , HIGH);}
120. if (co2 > CO2_max+20){digitalWrite(alarm , HIGH);}

121. if ((M_s == 1)&&(M_s_0 == 0)){ lcd.clear();MODE= MODE + 1;  if(MODE ==7){MODE=0;}}
122. M_s_0 = M_s;

123.        if ((Up == 1)&&(Up_0 == 0)){   lcd.clear();
124.              if(MODE ==1){T_max = T_max + 0.1; }
125.                    if(MODE ==2){T_min = T_min + 0.1;}
126.                    if(MODE ==3){H_max = H_max + 1;}
127.                    if(MODE ==4){H_min = H_min + 1;}
128.                    if(MODE ==5){CO2_max = CO2_max + 5;}
129.                    if(MODE ==6){T_c = T_c + 0.1;}
130.               }
131.Up_0 = Up;
132.  if ((Down == 1)&&(Down_0 == 0)){   lcd.clear();
133.                    if(MODE ==1){T_max = T_max - 0.1;}
134.                  if(MODE ==2){T_min = T_min - 0.1;}
135.                  if(MODE ==3){H_max = H_max - 1;}
136.                  if(MODE ==4){H_min = H_min - 1;}
137.                  if(MODE ==5){CO2_max = CO2_max - 5;}
138.                  if(MODE ==6){T_c = T_c - 0.1;}
139.
140.               }
141.   Down_0 = Down;
142.  if (millis()-cpt_M > 54000000)// 90 mn
143.          {
144.            cpt_M = millis();
145.            if (M==1){M=0;}else{M=1;}
146.            if  (M==1){digitalWrite(M1,HIGH);digitalWrite(M2,LOW);}
147.            if  (M==0){digitalWrite(M1,LOW) ; digitalWrite(M2,HIGH);}
148.            }
149.  if (MODE == 0)
150.  {
151.  lcd.setCursor (0,0);
```

```
lcd.print("T:");   lcd.print(T);    lcd.print(" H:");   lcd.print(H);
lcd.setCursor(0,1);
lcd.print("T35:");   lcd.print(temperatur);
//   lcd.setCursor(0,3);
   lcd.print(" CO2:");   lcd.print(co2);
//   lcd.setCursor (0,0);
//   lcd.print("Temperature: ");   lcd.print(T);  lcd.setCursor(18,0);  lcd.print("C");
//   lcd.setCursor(0,1);
//   lcd.print(" Humidity  : ");   lcd.print(H);  lcd.setCursor(18,1);  lcd.print("%");
//   lcd.setCursor(0,2);
//   lcd.print("Temp35  : ");   lcd.print(temperatur);  lcd.setCursor(18,2);  lcd.print("C");
//   lcd.setCursor(0,3);
//   lcd.print("CO2  : ");   lcd.print(co2);  lcd.setCursor(15,3);  lcd.print("PPM");
}
   if (MODE == 1)
{
lcd.setCursor (0,0);
lcd.println ("T_max:"); //lcd.println ("Temperature max:");
lcd.setCursor(3,1);
lcd.print(T_max);  //lcd.setCursor(10,1);  lcd.print("C");
  }
if (MODE == 2)
{
  lcd.setCursor (0,0);
  lcd.println ("T_min:");//  lcd.println ("Temperature min:");
lcd.setCursor(3,1);
lcd.print(T_min); // lcd.setCursor(18,1);  lcd.print("C");
  }
  if (MODE == 3)
  {
    lcd.setCursor (0,0);
    lcd.println ("H_max:");// lcd.println ("Humidity max:");
    lcd.setCursor(3,1);
    lcd.print(H_max); // lcd.setCursor(18,1);  lcd.print("C");
  }
  if (MODE == 4)
  {
  lcd.setCursor (0,0);
  lcd.println ("H_min:"); //lcd.println ("Humidity min:");
  lcd.setCursor(7,1);
```

```
192.  lcd.print(H_min); // lcd.setCursor(18,1);  lcd.print("C");
193.  }
194.  if (MODE == 5)
195.  {
196. lcd.setCursor (0,0);
197. lcd.println ("CO2_max:"); //lcd.println ("Humidity min:");
198.lcd.setCursor(7,1);
199.lcd.print(CO2_max); // lcd.setCursor(18,1);  lcd.print("C");
200.  }
201. if (MODE == 6)
202. {
203. lcd.setCursor (0,0);
204. lcd.println ("T_Consigne:"); //lcd.println ("Humidity min:");
205. lcd.setCursor(7,1);
206. lcd.print(T_c); // lcd.setCursor(18,1);  lcd.print("C");
207. }
208.  }
209.  void IISR()
210.  {
211.  TCNT2=0;
212. digitalWrite(R1,LOW);
213.  }
214.  ISR(TIMER2_COMPA_vect)
215. {
216. digitalWrite(R1,HIGH);
217.  }
218. ISR(TIMER2_COMPB_vect)
219. {
220. digitalWrite(R1,LOW);
221. }
```