

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE
SCIENTIFIQUE



UNIVERSITÉ DJELALI BOUNAAMA
(KHMIS MILIANA)



Faculté des Sciences et de la Technologie
Département

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master en Automatique et Informatique Industriel
Spécialité Automatique et Informatique Industriel

Intitulé :

Etude et réalisation d'un mini système aquaponique à base d'un Raspberry Pi

Encadré par :

Mr. FEKIR Mohamed

Présenté par :

Mr. TAIBOUNI Benyoucef

Mr. MOKRANE Mohamed Cherif

Examineurs :

- Mr. IKNI Samir
- Mr. MAHDAB Salim

Année Universitaire : 2019/2020

Remercîment:

Nous remercions dieu , Allah le tout puissant clément de nous avoir aidé et guidé tout au long de ce parcours et nous tenons a exprimer nos profonds remerciements a l'encadrer de notre mémoire de fin d'études de master , monsieur Fekir Mohamed enseignant à l'université Djilali Bounaama qui c'est toujours montré à l'écoute et a montré disponibilité et bien vaillance tout au long de la réalisation de de ce travail aussi pour son aide et temps qu 'il a bien voulu nous consacrer ses conseils pertinents , amabilité et patience qui ont permis à notre travail d'aboutir malgré les conditions difficiles lié à la pandémie covid-19.

Nous tenons remercier également toutes les personnes qui ont contribué au succès de notre mémoire fin d'études et qui nous ont aidée lors de la rédaction de celle ci

Nos sincères remerciements vont aussi à tous les professeurs, toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé nos réflexions et ont permis à évoluer notre cursus universitaire.

Une pensée particulière à nos très chers parents, nos familles et nos amis pour leurs encouragements leur soutien inconditionnel qui nous ont été d'une grande aide

À tous ces intervenants, nous vous présentons notre gratitude, et notre profond respect.

Résumé :

Ce travail est considéré comme étant une étude, une planification, et la conception d'une commande à distance d'un mini-système aquaponique en utilisant le microordinateur raspberry pi.

Nous avons donc étudié le système et ses composants afin de connaître notre besoin en capteurs et actionneurs pour faire fonctionner le système désiré.

Après nous passerons au choix du microcontrôleur qui aura la vitesse et les performances nécessaire à exécuter notre programme (raspberry pi).

Nous avons créé une application mobile et l'avons associée au Raspberry Pi afin de contrôler le système et d'en recevoir diverses informations.

Enfin, nous avons créé un modèle d'une serre contenant un bac d'élevage de poisson et un lit de culture, ce qui nous a permis de tester les performances du projet que nous avons réalisé.

Abstract:

This work is considered to be a study, planning, and design of a remote control of an aquaponics mini-system using the micro-computer raspberry pi.

We therefore studied the system and its components to determine our needs in term of sensors and actuators to operate the desired system.

Then we will choose the microcontroller that will have the speed and performance necessary to run our program (raspberry pi).

We created a mobile app and combined it with the Raspberry Pi to control the system and receive various information.

Finally, we created a model of a greenhouse containing a fish farm tray and a growing bed, which allowed us to test the performance of the project we carried out.

ملخص:

يُعتبر هذا العمل دراسة، تصميم وتنفيذ تطبيق من أجل التحكم الذاتي والتحكم عن بعد في نظام

اكوابونيك مصغر باستعمال الرازبيري باي

حيث قمنا في الاول بدراسة النظام و مكوناته و هذا لمعرفة ما نحتاجه كمستقبلات و محركات لتسيير

النظام ,وبعدها اخترنا المتحكم الذي بدوره تكون له سرعة الاستجابة في تنفيذ البرامج (رازبيري باي).

قمنا بانشاء تطبيق هاتف نقال وقمنا بربطه بالرازبيري باي . وقاعدة بيانات من أجل التحكم في النظام

و استقبال مختلف المعلومات منه.

وأخيرا قمنا بانشاء نموذج لبيت زراعي الذي يحتوي على حوض الاسماك و حوض الزراعة ,حيث

سمح لنا باختبار أداء المشروع الذي قمنا به.

Le sommaire

REMERCIEMENT:.....	
RESUME :.....	
LE SOMMAIRE	
LISTE DES FIGURES.....	
LISTE DES TABLEAUX.....	
LISTE DES ABREVIATIONS ET SYMBOLES :	
INTRODUCTION GENERALE.....	1
CHAPITRE I.....	
I.1 INTRODUCTION.....	3
I.2 DEFINITION DES SYSTEMES AQUAPONIE	3
I.3 HISTORIQUE DE L'AQUAPONIE	3
I.4 POUR QUOI PRATIQUER L'AQUAPONIE ?	4
I.5 COMMENT FONCTIONNE L'AQUAPONIE ?	4
I.6 LES AVANTAGES DE L'AQUAPONIE	5
I.7 TYPES DE SYSTEMES D'AQUAPONIE	6
I.7.1 Les systèmes de culture en eaux profondes (CEP).....	7
I.7.2 Les systèmes de lit de culture à substrat (système d'inondation et de drainage).....	7
I.7.3 La technique du film nutritif (NFT).....	8
I.8 LES ELEMENTS QUI COMPOSENT LE SYSTEME AQUAPONIQUE	8
I.8.1 L'azote	8
I.8.2 Nitrites	10
I.8.3 Nitrates	10
I.8.4 Les plantes	10
I.8.5 Les poissons.....	13
I.8.6 L'aquarium.....	15
I.8.7 Lit de culture aquaponique	16
I.8.8 L'eau de système.....	17
I.9 CONCLUSION	18
CHAPITRE II	
II.1 INTRODUCTION.....	19
II.2 LE MICROCONTROLEUR RASPBERRY PI.....	19
II.2.1 Les modèles de Raspberry pi.....	20
II.2.1.1 Modèles A.....	20
II.2.1.2 Modèles B	21
II.2.1.3 Caractéristiques du Raspberry Pi utilisé 3B+.....	22
II.2.2 Accessoires du Raspberry Pi.....	23
II.2.2.1 Alimentation.....	23
II.2.2.2 Carte microSD.....	23
II.2.2.3 Câble HDMI.....	24
II.2.2.4 L'écran	24
II.2.2.5 Clavier et souris.....	24
II.2.2.6 Boîtier	24
II.2.3 Systèmes d'exploitation pour Raspberry pi	25
II.2.3.1 Raspbian.....	25
II.2.3.2- Kali Linux.....	25
II.2.3.3 Windows 10 IoT Core.....	25
II.2.3.4 Ubuntu Core.....	25
II.2.4 Fonctionnalité et connectique du Raspberry Pi 3B+.....	26
II.2.5 Connecteurs et ports disponibles	26
II.3 LES CAPTEURS	27

II.3.1 Classification des capteurs	28
II.3.2 Les capteurs utilisés.....	29
II.3.2.1 Capteur de température DHT22.....	29
II.3.2.2 Capteur de température Ds18B20.....	30
II.3.2.3 Module de détection de lumière LDR.....	31
II.4 LES ACTIONNEURS	32
II.4.1- classification des actionneurs	33
II.4.2 Les actionneurs utilisés	33
II.4.2.1 Le moteur à courant continu.....	33
II.4.2.2- Les lampes – LED –.....	35
II.4.2.3- La résistance chauffante.....	35
II.4.2.4 L’extracteur d’air.....	35
II.4.2.5 Distributeur de nourriture pour poissons	36
II.4.2.6 Pompe à eau	36
II.4.2.7 Le diffuseur	36
II.5 LES PREACTIONNEURS.....	36
II.5.1 les relais	36
II.5.2 L’interface de puissance l298	37
II.6 LES LOGICIELLES, LES SITES WEB ET LES ENVIRONNEMENTS DE DEVELOPPEMENTS UTILISENT DANS LA PREPARATION DE L’INTERFACE DE CONTROLE A DISTANCE.....	37
II.6.1 Android studio.....	37
II.6.2 Firebase	38
II.6.3- Kotlin	38
II.6.4 Python.....	38
II.7 CONCLUSION	39
CHAPITRE III.....
III.1 INTRODUCTION	40
III.2 CONCEPTION ET REALISATION DU SYSTEME AQUAPONIQUE	40
III.2 .1 Fabrication de prototype	40
III.2.3 Partie opérative	42
III.2.3.1- Le capteur DHT22	42
III.2.3.2 le capteur LM393.....	42
III.2.3.3 le capteur Ds18b20.....	43
III.2.3.4 le ventilateur	44
III.2.3.5- Moteur DC.....	44
III.2.3.6 Actionneurs commandés par relais	45
III.3 PARTIE COMMANDE	46
III.3.1 Préparation de la base des données	46
III.3.2- Programmation d’application	50
III.3.2.1- Création d’un projet.....	50
III.3.2.2 Explication de programme utilisé	53
III.2.4 Préparation et programmation de Raspberry pi.....	59
III.2.4.1- Installation de Raspbian	59
III.2.4.2 Configuration de Raspberry pi.....	61
III.2.4.3- Programmation de raspberry pi en python	64
III.3 CONCLUSION	73
CONCLUSION GENERALE	75
REFERENCES BIBLIOGRAPHIQUES.....	76
ANNEXES :.....

Liste des figures

Figure I. 1 : un système aquaponique	3
Figure I. 2: une image qui illustre la méthode CEP.....	7
Figure I. 3: une image qui illustre la méthode de lit de culture a substrat	7
Figure I. 4: une image qui illustre la méthode du film nutritif	8
Figure I. 5: cycle d'azote dans l'aquaponie	9
Figure I. 6: Équations de transformation de l'ammoniac en nitrite puis en nitrate.....	10
Figure I. 7: plante tomate.....	10
Figure I. 8: plante laitue	11
Figure I. 9: plante cresson de fontaine	11
Figure I. 10: plante poivrons	11
Figure I. 11: plant concombres	12
Figure I. 12: plante chou-fleur	12
Figure I. 13: plante Chau	12
Figure I. 14: plante fraise.....	12
Figure I. 15: poisson tilapia.....	13
Figure I. 16: poisson truite	13
Figure I. 17: poisson perche soleil	14
Figure I. 18: poisson la carpe	14
Figure I. 19: poisson black-bass.....	14
Figure I. 20: poisson la brème	14
Figure I. 21: poisson-chat	15
Figure I. 22: une image qui illustre un aquarium	15
Figure I. 23: une image qui illustre un lit de culture.....	16
Figure II. 1: une image qui illustre quelques modèles de Raspberry pi	20
Figure II. 2 : une image qui illustre les différent ports de Raspberry pi B+	23
Figure II. 3:Alimentation 5v	23
Figure II. 4: une image qui illustre les classes de la carte microSD	23
Figure II. 5: un câble HDMI	24
Figure II. 6: Ecran de pc.....	24
Figure II. 7: clavier et souris.....	24
Figure II. 8: Boites de protection Raspberry pi	24
Figure II. 9: Logo système d'exploitation Raspbian	25
Figure II. 10: Logo système d'exploitation Kali linux	25
Figure II. 11: Logo système d'exploitation Windows 10 IoT core	25
Figure II. 12: Logo système d'exploitation Ubuntu	26
Figure II. 13: Une image qui illustre les entrees/sorties d'un Raspberry pi 'GPIO'	27
Figure II. 14: Principe de fonctionnement d'un capteur	28
Figure II. 15: exemple d'un capteur analogique "capteur de son"	28
Figure II. 16: exemple d'un capteur numérique "encodeur"	28
Figure II. 17: exemple d'un capteur logique " fin de cours"	28
Figure II. 18: capteur de température et humidité DHT22.....	29
Figure II. 19: méthode de câblage de DHT11	30
Figure II. 20: capteur de température DS18B20	30
Figure II. 21: mode Bus 1-wire	30
Figure II. 22: le module de détection de lumière.....	31
Figure II. 23: Caractéristique du module de détection de lumière.....	31
Figure II. 24: schéma électronique de module de détection de lumière.....	32
Figure II. 25: moteur à courant continu	33
Figure II. 26: comportement d'un moteur à courant continu.....	34
Figure II. 27: lampes LED	35

Figure II. 28: Résistance chauffante	35
Figure II. 29: Extracteur d'air	35
Figure II. 30: Distributeur d'aliment	36
Figure II. 31: pompe à eau	36
Figure II. 32: le diffuseur	36
Figure II. 33: Relais électromagnétique.....	37
Figure II. 34: module L298	37
Figure II. 35: logo de logiciel Android studio	37
Figure II. 36: logo de site web Firebase	38
Figure II. 37: logo de langage de programmation Kotlin.....	38
Figure II. 38: logo de langage de programmation python	38
Figure III 1: les démentions de notre prototype	40
Figure III 2: le prototype à l'état final	41
Figure III 3: Interconnexion DHT11 et Raspberry pi	42
Figure III 4: interconnexions le capteur LM393 et le Raspberry pi.....	43
Figure III 5: interconnexion Ds18B20 et Raspberry pi	43
Figure III 6: Interconnexion Raspberry pi et L298 et le moteur DC	45
Figure III 7: brochage relais Raspberry pi.....	45
Figure III 8: brochage des relais des actionneurs	46
Figure III 9: l'état final de la base de données.....	49
Figure III 10: interface initial de logiciel Android Studio	50
Figure III 11: Activité Main d'application.....	53
Figure III 12: Activité Serre d'application	56
Figure III 13: Activité Aquariume d'application.....	59
Figure III 14: Interface de logiciel Etcher	60
Figure III 15: Interface de logiciel Etcher au cours de flash.....	61
Figure III 16: Interface de configuration Raspberry pi	61
Figure III 17: Interface de configuration avancée	63
Figure III 18: interface de logiciel PuTTY	63
Figure III 19 : organigramme représente le fonctionnement de la porte.....	73
Figure III 20 : organigramme représente le fonctionnement de système dans l'état automatique.....	74

Liste des tableaux

Tableau I. 1: Les valeurs idéales de l'eau pour la culture aquaponique17

Tableau III. 1 : la table des childs et variables utilisés dans notre base des données49

Liste des abréviations et symboles :

µs : micro secondes

A : Ampère

API : Automate Programmable industriel

ARM: Advanced Risc Machine

CAN: Controller Area Network

CEC: Consumer Electronics Control

CSI: Camera Serial Interface

DSI: Display Serial Interface

GND: ground

Go: gigaoctet

GPIO: General Purpose Input/Output

HDMI: High-Definition Multimedia Interface

I2C: Inter Integrated Circuit Bus

IoT: Internet of Things

LDR: Dependent Resistor

LED: light-emitting diode

LVDT: Linear Variable Differential Transformer

MHz : megahertz

PPM: Pulse Position Modulation

PWM: Pulse Width Modulation

PXE: Preboot Execution Environment

RAM: Random Access Memory

RCA: Radio corporation of America

SDHC: Secure Digital high capacity

SMSC: Short Message Service Center

SPI: Serial Peripheral Interface

TOR: Tout ou rien

USB: Universal Serial Bus

UWP: Universal Windows Platform

VCC: Voltage Common Collecto

Wifi : Wireless Fidelity

DC : direct curren

Id : Identifiant

MLI : modulation de largeur d'impulsions

SSH: Secure Shell

Introduction générale :

L'agriculture a, depuis la nuit des temps, était au centre des préoccupations de l'homme. Indispensable à sa survie, l'être humain n'a eu cesse de développer cette science.

Les années défilent au gré des saisons, les hommes se multiplient, puis les siècles passent, les civilisations se succèdent et l'agriculture reste l'unique moyen d'assurer la nourriture du nombre grandissant d'individus. Et aujourd'hui, avec l'explosion démographique, il est primordial d'assurer les ressources afin de subvenir aux besoins toujours plus importants des populations.

En se référant au site Compteur.net, on observe que le nombre d'êtres humains sur terre aurait atteint les 7.7 milliards en début de l'année 2020. L'industrie n'a d'autre choix que de suivre cette évolution fulgurante, on assiste alors à l'apparition de machines modernes toujours plus performantes, des robots infatigables capables de répondre à la forte demande en constante croissance.

Selon le site Latribun.fr, la Fédération de robotique internationale a déclaré lors de sa dernière étude que le nombre de robots industriels dans le monde allait atteindre le nombre approximatif de 3.05 millions de robots au début de 2020. Cette augmentation impactera positivement la rentabilité de la production.

L'évolution de l'automatique a contribué au développement de tous les secteurs, tels que le secteur médical, l'éducation, le divertissement mais aussi et bien-sûr le secteur agricole qui connaît, en plus du progrès dans le domaine de la génétique qui touche les animaux et les plantes, l'introduction de robots se chargeant de nourrir, soigner les animaux et de surveiller la croissance des plantes sans aucune intervention humaine.

Cependant, on ne peut nier les différents aspects négatifs d'une évolution aussi rapide. L'aspect le plus important étant celui qui touche directement à la santé des individus et la dégradation de l'environnement : il s'agit bien entendu de l'utilisation abusive des engrais chimiques ainsi que l'intervention au niveau de la génétique ; ce qui a causé, et en est la preuve aujourd'hui, de nombreuses maladies incurables tel que le cancer.

Aujourd'hui, nous prenons conscience des dégâts qui découlent de l'exploitation démesurée et immorale des ressources naturelles. C'est pour cela que des scientifiques travaillent sans relâche à trouver des alternatives pour une agriculture plus saine pour les consommateurs et bien évidemment pour la planète. L'une de ces alternatives est l'Aquaponie, une technique récente et plus respectueuse de la nature que nous découvrirons dans ce travail.

Les systèmes aquaponiques se sont développés récemment et ont été adoptés dans de nombreux pays comme source simultanée de poissons et de plantes, en particulier les légumes, ce qui en faisait une grande dépendance, Ils ont donc dû être développés et convertis en un système industriel intelligent, dans lequel des processeurs, des capteurs et des actionneurs sont utilisés.

Dans cette optique, des questions récurrentes relatives à cette technique sont posées, à savoir ;

- Qu'est-ce que l'Aquaponie ?
- Et comment nous automaticiens pouvons laisser notre empreinte dans un tel domaine ?
- Et comment ceci peut-il affecter la rentabilité de ce système ?

Tous au long de ce travail, nous allons répondre à toutes ces interrogations. De ce fait, ce mémoire est scindé en trois principaux chapitres :

Dans le premier chapitre, nous avons essayé de donner une vision générale des systèmes aquaponiques en parlant de leurs types, composants et applications.

Dans le deuxième, nous avons parlé de tout ce qu'on va utiliser dans notre réalisation. Au début le contrôleur a été choisi « Raspberry pi », on le définit, et déclare ses différents modèles, accessoires, et fonctionnalités. Après, on a parlé des capteurs et actionneurs, leur types, classification, et le mode de fonctionnalité de chaque composant. Et dans la dernière partie, on a défini les différents logiciels et applications utilisés dans les étapes de la réalisation.

Dans le dernier chapitre, on a commencé par la fabrication du prototype, sa dimension, ensuite on déclare tous les capteurs et les actionneurs utilisés, son association avec le Raspberry pi. La partie suivante c'est la partie commande, nous avons créé la base de données qui va porter toutes les données de ce système, après nous avons créé aussi une application Android qui va permettre de contrôler le système à travers une base de données. A la fin, nous avons préparé le Raspberry pi du début jusqu'à l'installation de système d'exploitation, sa configuration, et la programmation de notre système.

Nous terminerons ce travail par une conclusion générale.

Chapitre I

I.1 Introduction :

L'Aquaponie, bien que peu répandue, en particulier dans notre pays, est une technique qui présente de nombreux avantages. C'est pourquoi une bonne compréhension de son principe ainsi qu'une étude approfondie de son fonctionnement nous permettront de profiter au mieux de ce système.

Ce premier chapitre sera dédié à la présentation de l'Aquaponie : sa définition, son historique, son fonctionnement, ses avantages, ses types ...etc.

I.2 Définition des systèmes Aquaponie : QU'EST-CE QUE L'AQUAPONIE ?

C'est la culture des plantes et des animaux aquatiques dans un environnement de recirculation. Il s'agit d'une synergie entre poissons et plantes. Le terme trouve son origine dans deux mots, aquaculture (l'élevage de poissons dans un environnement clos) et hydroponie (la culture de plantes hors sol) [1].

Les systèmes aquaporines sont de dimensions variées : de petites unités d'intérieur aux grandes surfaces commerciales. Il peut s'agir de systèmes d'eau douce ou de systèmes contenant du sel ou de l'eau saumâtre.

Les systèmes aquaponiques peuvent produire des aliments sains (poissons / herbes / légumes), utilisant un minimum d'eau pour des rendements élevés, ce qui peut contribuer à la sécurité alimentaire et nutritionnelle. Ils peuvent également représenter une entreprise commerciale réussie, tant pour les pays en développement que pour les pays développés [1].

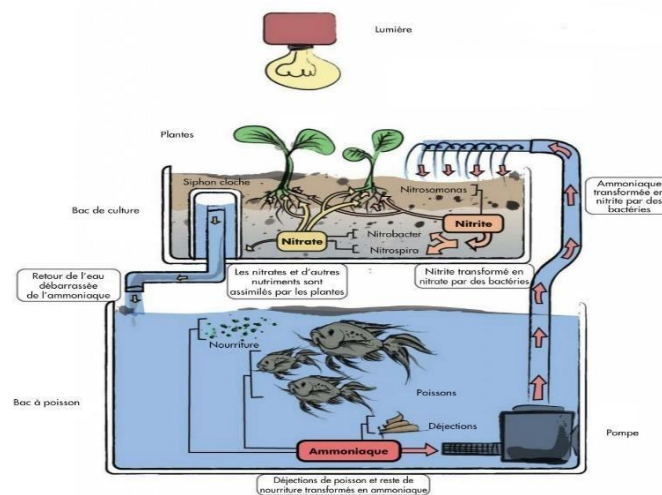


Figure I. 1 : un système aquaponique

I.3 Historique de l'Aquaponie :

L'Aquaponie, bien que largement développée par l'homme aujourd'hui, exploite un phénomène naturel, omniprésent sur notre planète. Que ce soit en eau douce ou en milieu marin, les excréments produits aquatiques sont ensuite minéralisés par les populations bactériennes. Les végétaux absorbent ensuite les éléments minéraux et l'eau s'en trouve purifiée. [1]

Les premières traces d'Aquaponie domestique remontent toutefois à 1500 ans, en Asie et en Amérique du Sud, sous des formes beaucoup plus simples que celles utilisées aujourd'hui. A cette époque, les asiatiques élevaient leurs canards dans des cages au-dessus de leurs élevages de

poissons, afin que les déchets des uns nourrissent les autres. L'eau enrichie des bassins servait ensuite à irriguer les cultures de riz. [1]

Au Pérou, les incas créaient des étangs à poissons avec des îles cultivables. Dans les régions de hauts plateaux, le climat y est froid avec des étés pluvieux et des hivers secs et gelés. Les îlots de culture, gorgés d'eau enrichie d'excréments d'oiseaux attirés par ce milieu artificiel, offraient des rendements impossibles à atteindre autrement dans cette région.

Ce n'est qu'autour de la fin des années 1970 que l'Aquaponie regagna l'intérêt du public et de la communauté scientifique, encouragée par la recherche de solutions aux problématiques environnementales et grâce à l'amélioration des techniques d'hydroponie et d'aquaculture. [1]

I.4 Pour quoi pratiquer l'Aquaponie ?

L'Aquaponie est un système de production alimentaire par recirculation qui utilise moins de 10 % de l'eau normalement nécessaire pour la pisciculture et la production végétale. Ainsi, elle convient tant pour une consommation à petite échelle ou domestique que pour une production commerciale d'aliments frais, notamment dans les collectivités où l'eau est rare. Il y a très peu de perte en eau, les pertes sont principalement causées par l'évaporation et la transpiration des plantes. Il n'y a aucun besoin de produits chimiques pour les plantes, ces produits entraîneraient la mort des poissons. Cependant, certaines mesures de protection organique des plantes pourraient s'avérer utiles, surtout s'il s'agit d'une monoculture. [1]

L'un des principaux avantages de l'Aquaponie est qu'elle est un excellent moyen de produire des aliments riches en protéines, comme le poisson et les légumes, à la fois dans les régions les plus pauvres (soutien à la sécurité alimentaire) et dans les zones où il existe une forte demande de produits de bonne qualité. Par exemple, des denrées biologiques à prix élevé peuvent être produites pour les marchés urbains. Les systèmes aquaponiques économisent l'eau et les plantes qui poussent dans ces systèmes se développent plus rapidement, sont plus grandes et ont un rendement plus élevé (15 %) que celles qui poussent dans un système de culture hydroponique courant. [1]

Certains disent même que ces produits ont meilleur goût ! Les plantes n'ont pas besoin de réapprovisionnement en eau. Parce qu'ils utilisent constamment de l'eau recyclée, ces systèmes peuvent être installés pratiquement n'importe où, même à proximité des marchés. [1]

D'un point de vue commercial, l'Aquaponie présente l'avantage de réduire les coûts de distribution et de baisser l'empreinte carbone de l'entreprise. De plus, les besoins en intrants sont faibles une fois que le système a été mis en place, il suffit d'une simple maintenance des systèmes et des poissons, des aliments pour poissons, des graines (ou semis) et de l'eau pour approvisionner l'unité. Les plantes n'ont besoin pratiquement d'aucun engrais chimique, et tous les nutriments dont elles ont besoin proviennent de déchets des poissons. [1]

I.5 Comment fonctionne l'Aquaponie ?

L'Aquaponie est l'amalgame de l'aquaculture et de la culture hydroponique dans un système unique de production. Cette méthode repose sur des déchets de poissons utilisés en tant que solution nutritive organique pour cultiver des légumes. Dans un tel système, l'eau s'écoule du réservoir de poissons vers un filtre biologique où des bactéries décomposent les déchets de poissons en une solution nutritive organique pour la culture des légumes. Les plantes absorbent alors les éléments

nutritifs de l'eau qui est ainsi nettoyée avant d'être remise en circulation pour un retour dans les bassins à poissons. [2]

Les bactéries jouent un rôle fondamental dans ce processus. Elles convertissent l'ammoniaque, qui est un composant majeur des déchets de poisson, en nitrate (une forme plus accessible de l'azote pour les plantes), ce qui évite que l'eau ne devienne toxique pour les poissons (cette transformation de l'ammoniaque en nitrate est connue comme le processus de nitrification). Il est essentiel que chaque unité aquaponique ait un composant de filtration biologique pour abriter les bactéries, ce qui permet à ces dernières de convertir constamment l'ammoniaque en nitrate. [2]

Les unités d'Aquaponie peuvent être de diverses formes et tailles, du petit appareil posé sur un banc de cuisine et ayant recours aux poissons rouges et à la culture d'herbes, aux systèmes plus importants avec la perche d'argent ou le tilapia et des tomates, courgettes, laitues, etc. Il existe aussi des unités encore plus complexes pour produire plusieurs tonnes de poisson et des milliers de plantes mensuellement, ce qui est idéal pour une production à l'échelle commerciale. [2]

I.6 Les avantages de l'Aquaponie :

1. Pour que le système d'Aquaponie reste fonctionnel, les seuls intrants nécessaires sont la nourriture des poissons et un peu d'eau de temps en temps. Il n'y a aucun besoin d'ajouter de l'engrais organiques, pas de traitement ni rien d'autre, donc la culture en Aquaponie est organique. En d'autre terme, L'Aquaponie est à 100 % un processus naturel. [3]

2. L'une des forces les plus importantes du système aquaponique est qu'il fournit une diversité de plantes cultivées : Il est possible de cultiver de grandes variétés de végétaux, ce système n'est pas limité à des plantes spécifiques. Il est connu que les plantes ont besoin d'air (oxygène et dioxyde de carbone), d'eau et de sels minéraux pour une croissance normale et c'est ce que fournit notre système Aquaponie. Ce système d'Aquaponie ne se limite pas à la variété des produits, mais nous donne aussi un grand rendement de cultures mixtes : nous donne par exemple de la laitue et des tomates dans un même espace cultivé.

3. La méthode d'Aquaponie permet d'assainir et de conserver l'eau, et les seules pertes d'eau que l'on peut observer sont dues à l'eau absorbée par les plantes et à l'évaporation. Ce système de production a l'avantage de valoriser les effluents chargés en composés dissous venant de la production aquacole, en les rendant utilisables en tant que nutriments pour la production de plantes hydroponiques. Il n'y a pas des déchets car nous pourrions récupérer les boues de filtration pour ajouter de la fertilité à nos plantes en terre. Sachons également qu'en Aquaponie, nourrir les poissons est gratuit car nous pouvons les nourrir avec des vers issus de notre compost. Même les déchets végétaux peuvent servir pour nourrir les poissons. Il est aussi possible de recycler les restes de viande et de poisson. En Aquaponie, il y a une économie d'eau et pas de déchets, rien ne se perd donc ! [3]

4. Les légumes, les fruits et les poissons produits dans le système d'Aquaponie sont très savoureux, en termes simples le système d'Aquaponie fournit des produits délicieux et sains, et donc répondant à certains besoins de légumes et de poissons.

5. Notre système aquaponique est un système fermé, alors les traitements faits aux plantes seraient directement transmis aux poissons par l'eau, et pour cela nous ne pouvons pas utiliser de produits chimiques sous peine de voir nos poissons mourir immédiatement. L'Aquaponie est donc

une méthode de culture raisonnable qui ne permet que l'utilisation de produits organiques en l'absence totale de pesticides. [3]

6. L'installation de système aquaponique est un peu facile, contrairement à l'agriculture dans le sol qui a besoin de préparer le terrain et d'autres difficultés que nous rencontrons. Aussi un système Aquaponie est facile à démonter.

7. Élever des poissons d'ornement est une activité de détente et de loisir. Nous pouvons élever cette espèce de poisson dans notre système d'Aquaponie. Mais nous pouvons également élever des poissons comestibles et profiter du poisson frais et savoureux grâce à notre système. Tous les poissons d'eau douce sont compatibles avec l'Aquaponie. [3]

8. L'Aquaponie présente également un avantage au niveau du confort car, l'effort physique étant minime, les agriculteurs sont épargnés par certains maux tels que le mal de dos.

9. L'Aquaponie fournit une formidable occasion de jardiner et peut aussi facilement être installée dans les villes et à l'intérieur des maisons surtout pour les personnes ayant peu de place. Le système aquaponique fonctionne en alimentant des cultures à partir d'un milieu aqueux, ce qui nous donne l'avantage d'éviter le forage et la présence de mauvaises herbes ce qui fait gagner énormément de temps tout en augmentant la productivité de du système. [3]

10. L'Aquaponie est la solution idéale pour les zones difficiles, les patios, les sols pauvres ou même les jardins sur les toits. En prenant en compte une lumière appropriée et une source d'énergie, un système aquaponique peut être installé pratiquement partout c'est donc une très bonne solution pour promouvoir l'agriculture dans le désert Algérien. C'est un excellent moyen d'accroître la productivité de systèmes existants. Deux cultures sont produites au sein d'une infrastructure unique.

11. Une fois que nous savons comment le système d'Aquaponie fonctionne, le processus de maintenance devient très facile et prend seulement quelques minutes pour redémarrer le système fonctionne.

12. Les systèmes aquaponique peuvent être conçus et adaptés à nos besoins spécifiques. L'Aquaponie fournit une excellente démonstration pratique de la vie au travers des techniques de culture modernes. C'est une vraie prise de conscience qui peut être apportée dès le plus jeune âge. [3]

1.7 Types de systèmes d'Aquaponie :

Il existe de nombreuses configurations différentes de systèmes aquaponiques. Les composants communs à chaque système sont les aquariums et un lit de plantes sans sol. Les variables comprennent les composants de la filtration, les composants de la tuyauterie, le type de lit de la centrale et la quantité et la fréquence de la circulation de l'eau et de l'aération.

Il existe trois principales méthodes aquaponiques émergentes dans l'industrie : la technique de film nutritif, le système de culture en eaux profondes, et le système de lit de culture à substrat. Chacune de ces méthodes est basée sur une conception de système hydroponique, avec des adaptations pour le poisson et la filtration : [4]

I.7.1 Les systèmes de culture en eaux profondes (CEP) :

Cette méthode consiste à suspendre des plantes à des feuilles de polystyrène qui flottent dans des canaux avec de l'air qui arrive du fond du bassin à peu près à chaque mètre carré. Cette méthode est la plus utilisée dans les grandes exploitations aquaponiques commerciales qui traitent d'une culture spécifique (de manière stéréotypée, la laitue, les feuilles de salade ou le basilic). [4][2]

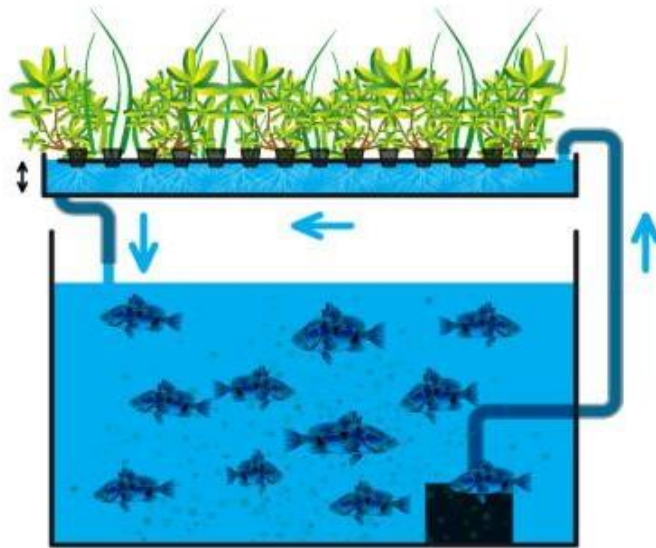


Figure I. 2: une image qui illustre la méthode CEP

I.7.2 Les systèmes de lit de culture à substrat (système d'inondation et de drainage) :

Les unités de lit de culture à substrat sont la conception la plus populaire pour l'Aquaponie à petite échelle car elles sont efficaces en termes d'espace, ont un coût relativement faible et conviennent aux débutants en raison de leur conception très simple. Dans les unités de lit de culture, le substrat utilisé pour soutenir les racines des plantes et agit également comme moyen de filtration. Cette double fonction est la principale raison pour laquelle les unités de lit de culture sont les plus simples. [4] [2]

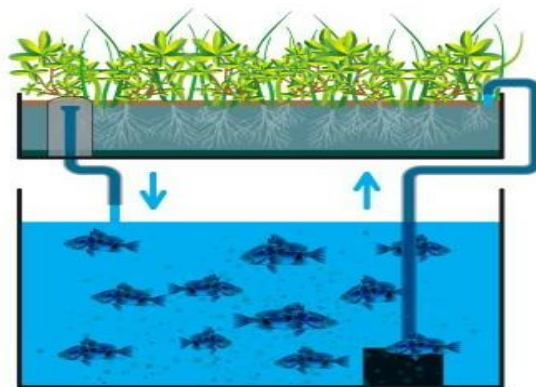


Figure I. 3: une image qui illustre la méthode de lit de culture à substrat

I.7.3 La technique du film nutritif (NFT) :

Nécessite l'utilisation de tuyaux en plastique disposés horizontalement pour faire pousser les légumes. L'eau est pompée dans chacun des tuyaux hydroponiques depuis le filtre biologique, en un petit débit régulier qui crée un flux d'eau aquaponique peu profonde riche en nutriments et qui ruisselle au fond du bassin. Sur le dessus, le tube est percé d'un certain nombre de trous dans lesquels sont placées les plantes pour croître. [4] [2]



Figure I. 4: une image qui illustre la méthode du film nutritif

I.8 Les éléments qui composent le système aquaponique :

Chaque élément du système aquaponique a une fonction et un objectif, soit pour la pisciculture soit pour la culture des plantes dans de meilleures conditions. Il est essentiel pour nous d'essayer d'équilibrer notre système aquaponique en faisant des études approfondies sur chaque élément de ce système, et certainement les éléments les plus importants sont l'azote, les plantes et les poissons. L'élevage de poissons (et donc, l'alimentation donnée aux poissons) doit fournir les nutriments nécessaires au développement des plantes et en contrepartie, les plantes doivent suffisamment filtrer l'eau pour les poissons. De plus, le filtre biologique intermédiaire entre les poissons et les plantes, doit également être suffisamment grand et efficace pour pouvoir convertir l'ensemble des déchets rejetés par les poissons, et pour cela nous allons donc mentionner l'importance de chacun de ces éléments et les conditions optimales que nous choisirons pour obtenir un meilleur résultat. [6] [5]

I.8.1 L'azote

L'Aquaponie est une méthode de culture complexe qui a évolué comme une solution efficace pour le recyclage des déchets générés dans l'aquaculture. Les déchets azotés produits dans l'aquaculture des poissons ou écrevisses sont utilisés comme engrais pour les plantes cultivées en hydroponie. L'aquaponie repose sur le cycle de l'azote présent dans la nature. Un système aquaponique est naturellement plein d'azote. Des bactéries et des vers aident à la dégradation des déchets des poissons en nitrates et nitrites qui peuvent être absorbés par les plantes. L'eau d'un

système aquaponique est propre, n'est pas recyclée et tourne en circuit fermé. Pour cette raison, l'azote est l'élément le plus important du système d'aquaponie. [6] [5]

A cet effet, L'azote, est un des éléments constitutifs des protéines. Il est donc vital, pour tous les êtres vivants, aussi bien animaux que végétaux. Les engrais hydroponiques complets sont composés de 20% à 30% d'éléments azotés et sont responsables pour plus de 10% des coûts de production. En aquaculture seule, l'azote représente 50% à 70% des coûts de production et 70% de l'azote se retrouve dans les rejets, sous forme d'ammoniac.

L'aquaponie permet de récupérer les éléments nutritifs relâchés par les poissons afin de nourrir la croissance des plantes, tout en réduisant les coûts et les impacts environnementaux. L'azote peut, quant à lui, être utilisé comme indicateur de l'équilibre du système et de l'état des solutions nutritives. [6] [5]

Le cycle de l'azote d'un système aquaponique commence par le nourrissage des poissons. Plus la nourriture est riche en protéines, plus celle-ci contient de l'azote. Une partie des protéines consommées par les poissons est absorbée pour la croissance des poissons, le reste est rejeté par l'urine, sous forme d'ammonium. Cette forme de l'azote est ensuite consommée et transformée en nitrites par une première génération de bactéries présente dans l'eau et concentrée dans le bio filtre. L'ammonium et les nitrites sont hautement toxiques à partir de basses concentrations pour les poissons. Ils doivent donc rapidement être dégradés. Une seconde génération de bactéries transforme les nitrites en nitrates, Cette dernière configuration chimique est toxique à des concentrations 100 fois plus élevées que les deux précédentes et est la plus accessible pour les plantes.

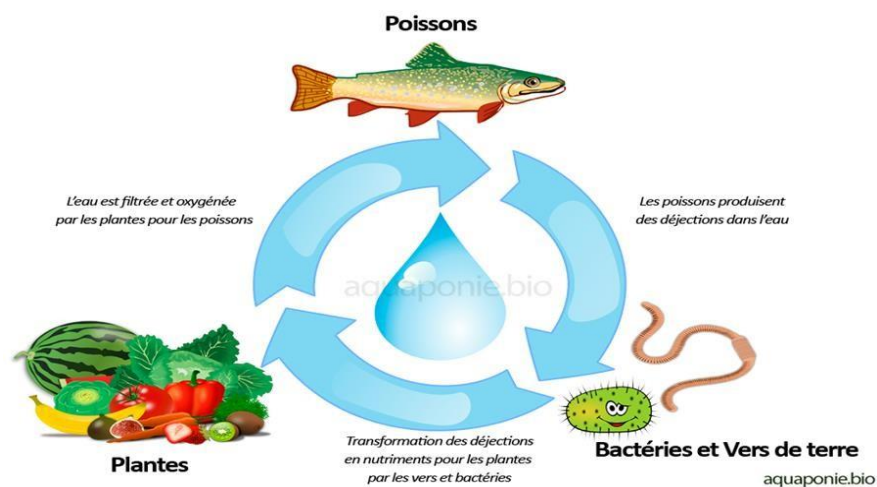


Figure I. 5: cycle d'azote dans l'aquaponie

Les bactéries jouent aussi un rôle capital, en sens inverse, dans le recyclage et la minéralisation de l'azote organique avec notamment Nitrosomonas pour la nitrification ou nitritation (transformation de l'ammoniac NH_3 ou NH_4^+ en NO_2^-) et Nitrobacter pour la nitratation (transformation de NO_2^- en NO_3^-). [7]

Le processus de conversion se déroule en deux étapes, chacune impliquant un groupe de bactéries spécifiques, tel qu'indiqué sur la (Figure I. 6). Par conséquent un bon établissement de ces deux types de colonies bactériennes est crucial pour la réussite de la production aquaponique.

1. Ammoniac → Nitrite

2. Nitrite → Nitrate

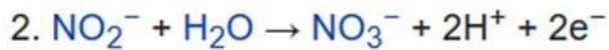


Figure I. 6: Équations de transformation de l'ammoniac en nitrite puis en nitrate

I.8.2 Nitrites

Les nitrites (NO_2) sont des sels de l'acide nitreux qui résultent de la réduction de l'ammoniaque par les nitrobactéries. Ces mêmes nitrites vont ensuite être transformés en nitrates par ces mêmes nitrobactéries. Trop de nitrites dans l'eau et c'est l'asphyxie assurée pour nos poissons. Sachons que chez l'homme et les mammifères, la présence de nitrites dans le sang empêche l'hémoglobine de fixer convenablement l'oxygène. [8]

I.8.3 Nitrates

Les nitrates (NO_3) constituent le produit final essentiel du cycle de l'azote mais, à la différence de l'ammoniaque et des nitrites, ils ne commencent vraiment à être toxiques qu'à des concentrations relativement élevées et sur un plus long terme. Les poissons résistent mieux aux forts taux de nitrates qu'aux forts taux de nitrites, ce n'est pas pareil. Tout comme les questions de température ou de PH, la résistance des poissons face à une surcharge de nitrates est propre à l'espèce des poissons élevés. [8]

I.8.4 Les plantes :

La vérité est qu'un système aquaponique peut faire pousser pratiquement n'importe quel type de plante : légumes, plantes fruitières, herbes, plantes à fleurs ou micro-vert. Il est possible de mettre en place différents systèmes afin de cultiver un éventail de plantes différentes. Mais vous pouvez aussi vous concentrer sur les légumes ou herbes aromatiques les plus efficaces en culture aquaponique, et permet Les meilleurs : [9]

I.8.4.1 Tomates :

La tomate se débrouille exceptionnellement bien dans un système aquaponique. Il sera plus facile de contrôler la température et même la quantité de soleil que reçoivent les plantes. Cependant, il convient de noter que les plants de tomates ont tendance à attirer des ravageurs dont il peut être exceptionnellement difficile de se débarrasser. [9]



Figure I. 7: plante tomate

I.8.4.2 Laitue :

La laitue cultivée en Aquaponie est en fait l'un des légumes feuilles les plus productifs. La laitue pousse bien dans l'eau dont la température se situe dans la zone de confort pour le tilapia.

Tout ce que vous avez à faire est de décider si vous commencez vos semis dans le système aquaponique ou en terre. Bien sûr, si le semis se fait dans un bac de culture traditionnelle, vous devrez laver les racines avant de les

introduire dans le système aquaponique.

Cependant, cela peut être bénéfique pour prévenir la perte de graines dans vos milieux de culture. La laitue est une excellente culture aquaponique pour les débutants. [9]

I.8.4.3 Cresson de fontaine :

Ce légume est l'une des meilleures plantes pour l'aquaponie parce qu'il pousse très vite. Vous pouvez planter une très petite quantité qui se multipliera à une vitesse étonnante.

Cependant, vous devez vous interroger sur l'opportunité de planter ce légume. Vous ne pouvez manger qu'une quantité limitée de cresson de fontaine alors que sa capacité à se multiplier peut potentiellement causer des problèmes en colmatant le lit de culture et même avec le système de circulation d'eau. [9]

I.8.4.4 Poivrons :

Les poivrons peuvent être extrêmement difficiles à cultiver classiquement dans le sol du fait de leur grand besoin en eau et en ensoleillement.

Il est logique de les cultiver dans un petit système aquaponique. Nous pourrions surveiller la température de l'installation et nous assurer que les niveaux de nutriments dans l'eau sont optimisés [9].

Les poivrons ne réussissent pas bien dans un système de radeau, en revanche, ils prospèrent exceptionnellement bien dans un lit de substrat.



Figure I. 8: plante laitue



Figure I. 9: plante cresson de fontaine



Figure I. 10: plante poivrons

I.8.4.5 Concombres :

Le concombre est une plante avide d'eau. Un système aquaponique lui donne donc tout ce dont il a besoin pour s'épanouir, pas seulement pour survivre.

Les concombres ont tendance à avoir de gros systèmes racinaires ; nous devons surveiller les racines pour nous assurer qu'elles n'envahissent pas et ne bloquent pas le système de circulation d'eau.

Ils sont très efficaces pour capter l'azote, ce qui peut priver d'autres plantes dans le système. Pour cette raison, il est conseillé de laisser un espace de 30 à 60 cm entre les plantes et d'éviter de sur planter. [9]



Figure I. 11: plant concombres

I.8.4.6 Chou-fleur :

Les choux fleurs sont très résistantes et nécessitent très peu d'entretien. Ils sont généralement résistants aux insectes et aux maladies, ce qui en fait l'une des meilleures plantes pour l'aquaponie.

Le chou-fleur a un cycle d'environ 12 semaines. Il n'aime pas la lumière directe du soleil. Pour éviter de l'endommager, il est préférable de lui recouvrir la tête avec ses propres feuilles. [9]



Figure I. 12: plante chou-fleur

I.8.4.7 Chou :

Le chou est un aliment de base dans de nombreux pays à travers le monde. C'est aussi l'une des meilleures plantes pour l'aquaponie. En général, cette culture nécessite très peu d'entretien. Nous devons faire attention aux ravageurs habituels : pucerons, maladies fongiques et pourriture noire. Le chou peut être prêt à être récolté à partir de 9 semaines. [9]



Figure I. 13: plante Chau

I.8.4.8 Fraise :

Ces petits fruits rouges se mangent toute l'année. (Techniquement, une fraise n'est pas un fruit, un légume ou une baie, mais c'est une autre histoire). Il s'agit en fait d'une excellente plante aquaponique à cultiver dans votre système. Grâce à l'apport constant d'eau riche en nutriments et à la capacité de contrôler leur environnement, les fraises peuvent être récoltées toute l'année. [9]

La plupart des fraisiers ne produisent que quelques



Figure I. 14: plante fraise

fraises ; si vous voulez les vendre, vous avez besoin d'avoir beaucoup de plantes pour.

I.8.5 Les poissons :

Les poissons sont les éléments qui font la différence entre le système aquacole et le système piscicole, donc son importance dans ce domaine est évidente.

La sélection des poissons est l'une des étapes de base de la réalisation de notre système, et nous la choisissons en fonction de normes, dont les plus importantes sont :

- Adaptation facile dans les divers environnements
- Propre à la consommation
- Bonnes capacités de reproduction
- Taux de croissance très rapide

Les poissons listés dans ce top sont des poissons approuvés par aquaponics.com qui figure parmi les leaders de l'aquaponie depuis une vingtaine d'années. Ils ont en effet testé tous ces poissons dans leurs systèmes aquaponique et ce sont ceux qui ont été retenus parmi tous les poissons utilisés durant leurs essais. [9]

I.8.5.2 Le tilapia :

Est le poisson qu'on voit sans doute le plus en aquaponie. Il peut mesurer jusqu'à 50 centimètres de long. Le tilapia est un poisson d'eau douce ou d'eau saumâtre mais sur les centaines d'espèces de tilapias, seules un petit nombre est compatible en aquaponie.

En fait, le tilapia c'est une sorte de carpe exotique à croissance très rapide. De fait, les tilapias aiment l'eau chaude. [10]



Figure I. 15: poisson tilapia

I.8.5.3 La truite :

Est aussi très courante en aquaponie. Sa chair est appréciée partout dans le monde et sa croissance est elle aussi rapide. On peut arriver à passer d'une truitelle à une truite de taille respectable (on l'appelle alors la truite portion) pour être consommée en moins d'un an. En général, en aquaponie, on voit souvent l'espèce arc-en-ciel issue de la pisciculture est une espèce de truite hybride obtenue à partir de croisement et qui ne se reproduit pas naturellement. Pour qu'elle se reproduise, il faut qu'elle soit dans son milieu naturel, en Amérique du Nord par exemple, autrement il faut avoir recours à l'insémination artificielle en laboratoire avec des éprouvettes et des accessoires adaptés... fait surprenant : la truite arc-en-ciel peut dépasser les 120 cm et les 25 kilos et a une espérance de vie de plus de dix ans![10]



Figure I. 16: poisson truite

I.8.5.4 La perche soleil, argent ou dorée :

Les perches sont magnifiques et commencent à prendre de plus en plus de place dans le monde de l'aquaponie. Les aquaponistes les apprécient beaucoup et nombreux sont ceux à se lancer dans l'élevage de perches. Les perches se pêchent aussi bien dans des eaux douces que dans des eaux saumâtres. Il existe un nombre impressionnant de dérivés de la perche à travers le monde.

[10]



Figure I. 17: poisson perche soleil

I.8.5.5 Les carpes communes et carpes amour ou même carpe Koi :

La carpe regroupe sous son nom plusieurs espèces. Elle vit environ une vingtaine d'années mais certaines sont âgées de plus de 70 ans. Certains disent même avoir déjà vu des carpes centenaires. Elle est appréciée en pisciculture car elle n'a pas son goût de vase caractéristique dès lors qu'elle est élevée dans un bassin propre, sans vase. Une légende urbaine affirme que la

carpe Koi ne serait pas comestible. [10]



Figure I. 18: poisson la carpe

I.8.5.6 Le black-bass :

Est un poisson très vif et combatif. En Europe il ne dépasse jamais les 30 à 50 centimètres et peut peser entre 500 grammes et 3 kilos. On les voit peu en aquaponie mais ils peuvent être élevés en aquaponie.

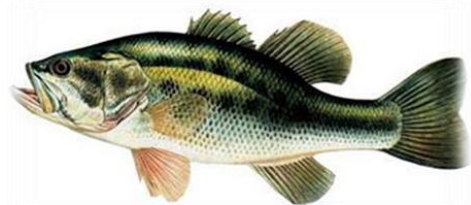


Figure I. 19: poisson black-bass

I.8.5.7 La brème :

Ces poissons étant des bios accumulateurs, ils sont souvent interdits à la consommation car dans leurs habitats naturels pollués, ils se chargent de métaux lourds et deviennent alors impropres à la consommation. Heureusement en aquaponie, ils peuvent être élevés et sont très appréciés. [10]



Figure I. 20: poisson la brème

I.8.5.8 Le poisson-chat :

Ou le silure est souvent mal vu en Europe par les pêcheurs mais il est bien adapté pour l'élevage en bassins à condition de ne pas l'élever jusqu'à l'âge adulte où il peut facilement dépasser les 1,5 mètres de long.



Figure I. 21: poisson-chat

I.8.6 L'aquarium :

Il s'agit d'un réservoir rempli d'eau dans lequel vivent des animaux et/ou des plantes aquatiques, par exemple des poissons, des mollusques, des crustacés, des tortues aquatiques ou des coraux, ainsi que des algues, mais aussi de nombreux microorganismes invisibles à l'œil nu.

L'aquarium communautaire décoratif, d'eau douce ou d'eau de mer, est la forme la plus populaire. Il s'agit d'un bac vitré de petite dimension placé dans une habitation ou un lieu public. Il est une reproduction de l'habitat naturel de l'espèce ou des espèces qui habite(ent) cet aquarium. Il peut aussi contenir des décorations diverses à l'apparence moins naturelle. Il demande des soins particuliers qui peuvent devenir un loisir, l'aquariophilie.



Figure I. 22: une image qui illustre un aquarium

Mais dans notre étude l'aquarium ne sera pas utilisé comme un élément de décoration seulement, mais il servira pour l'élevage des poissons de système aquaponie. Par conséquent, nous devons définir les paramètres que nous allons traiter :

- **Aération :**

L'oxygène est aussi important pour les poissons que pour nous, il faut donc assurer l'oxygène afin qu'ils puissent exercer leurs diverses activités

- **Eclairage :**

La lumière est un facteur de grande importance en aquariophilie, il faut bien choisir son éclairage en fonction des besoins de l'aquarium (poissons et plantes). Durée de l'éclairage Les plantes tropicales ont une durée d'éclairage de 12 heures. La durée maximale d'éclairage est de 15 heures, au-delà les plantes (et les poissons !) ont besoin d'une période de repos. En aquarium, on recommande généralement de démarrer avec 6 à 8 heures d'éclairage maximum, car le temps que les plantes s'adaptent au milieu aquatique, une partie importante de la lumière profite aux algues. [12]

- **Thermorégulation :**

Le maintien en aquarium de poissons dits « tropicaux » requiert la mise en place d'un système de chauffage dans le bac (thermorégulation). La température moyenne à maintenir pour un aquarium standard de poissons tropicaux (bac d'ensemble) se situe aux alentours de 25° à 27°C, Dans le cas de variétés particulières, de traitement ou de reproductions spécifiques, les températures peuvent éventuellement évoluer entre 18° et 33° C selon l'espèce et le biotope originel. [11]

- **Nutrition :**

La plupart des aquariophiles s'accordent sur deux à trois distributions par jour, chaque distribution ne dépassant pas les 3-4 minutes. Ceci bien-sûr à l'exception des poissons de fond qui ont une manière de s'alimenter complètement différente. [13]

I.8.7 Lit de culture aquaponique :

Les lits de culture (ou bacs de culture) sont une part importante d'un système aquaponique car ils effectuent de nombreuses tâches essentielles. Plus important encore, les bactéries s'y développant agissent comme un bio filtre, la collecte et le traitement de l'ammoniac et des déchets solides du poisson. Après quoi, il retourne l'eau propre filtrée vers l'eau des poissons. Deuxièmement, vous êtes récompensé avec une excellente production de légumes frais grâce à l'apport de nutriments et de nitrates lors du processus « de nettoyage de l'eau ».



Figure I. 23: une image qui illustre un lit de culture

Et pour les plantes on va définir comme paramètres à traiter :

- **Température et humidité d'environnement :**

La température et l'humidité sont des paramètres essentiels pour une croissance rapide des plantes, surtout en hiver, C'est pourquoi il est préférable d'utiliser une serre, afin d'avoir un bon contrôle de la température

- **Circulation d'eau :**

L'eau charge des éléments nutritifs est acheminée jusqu'au bac de culture. Les plantes assimilent les éléments nutritifs nécessaires à leurs croissances et purifient l'eau.

La circulation se fait dans un circuit ferme.

- **Eclairage :**

La lumière est l'un des éléments essentiels de la vie végétale pour réaliser la photosynthèse. Et en termes de quantité d'énergie consommée, « la plupart des jardiniers utilisent au moins 25 W pour 30 cm² d'espaces de culture quel que soit le type de lumière. Nous pouvons diminuer cette puissance si nous pouvons compléter avec de la lumière naturelle ou que nous cultivons une plante qui ne nécessite pas beaucoup de lumière comme la laitue par exemple. Cependant, de nombreux jardiniers préfèrent doubler voire tripler la puissance recommandée pour atteindre des taux de croissance plus rapide. Il n'y a vraiment rien de mieux que d'utiliser trop de lumière mais il faut rester vigilant car dans de petits espaces cela génère énormément de chaleur qui peut être difficile à contrôler. La plupart des jardiniers d'intérieur utilisent un éclairage compris entre 12 et 18 heures par jour » [12]

I.8.8 L'eau de système

Voici les valeurs que nous devons viser pour avoir une eau idéale pour la culture aquaponique:

Les éléments	Les valeurs idéales
PH	Entre 6 et 7
Température eau	Entre 18 et 30°C
Oxygène dissous	Entre 5 et 8 mg/litre (ou plus)
Ammoniac	0 mg/litre
Nitrites	0 mg/litre
Nitrate	Entre 5 et 150 mg/litre
KH	Entre 60 et 140 mg/litre

Tableau I. 1: Les valeurs idéales de l'eau pour la culture aquaponique

Choix de l'eau :

Les eaux sont très souvent polluées et pour cela, dans les systèmes aquaponiques, de nombreuses types d'eaux sont utilisées. Dans la suite nous le reconnaissons et les caractéristiques de chaque type : [14]

- **Eau de robinet :**

L'utilisation de l'eau du robinet est plus facile que d'autres sources d'eau mais il reste cependant des inconvénients majeurs, Lié à l'ajout de nombreux suppléments. Un autre problème majeur avec l'eau du robinet est qu'elle est souvent calcaire et difficile d'absorption par les plantes.

- **Eau de pluie**

La meilleure eau est évidemment l'eau de pluie, l'eau de pluie doit être correctement récoltée et stockée. Elle restera la source d'eau la plus saine et naturelle qui convient à notre système aquaponique. Cette méthode fait de belles économies d'eau et en plus elle est de bien meilleure qualité que l'eau du robinet. [14]

- **Eau de puits**

L'eau peut être tirée d'un puits. Cependant, l'eau que nous extrairons du puits doit être testée car elle peut être polluée et l'eau est sélectionnée en testant la proportion des matériaux dont nous avons parlé plus tôt.

I.9 Conclusion :

Dans ce chapitre, nous avons introduit la notion d'aquaponie et nous avons insisté sur les avantages et les bienfaits qu'elle offre à l'agriculture, à l'économie, à l'écologie et à l'être humain de manière générale.

Nous avons listé les différentes techniques utilisées pour fabriquer un système aquaponique, nous avons présenté un type particulier des systèmes aquaponiques : le système de culture en eaux profondes, et nous avons présenté tous les composants de ce système. Dans le chapitre suivant, nous allons aborder les éléments électroniques et automatiques du système aquaponique qui seront utilisés dans notre système.

Chapitre II

II.1 Introduction :

L'un des points les plus importants pour la réussite de notre projet est le choix de matériels. Ainsi, les contraintes suivantes doivent être prise en considération à savoir ;

- Disponible en cas de dysfonctionnements.
- Facile à utiliser
- Longue durée de vie et prix raisonnable.
- Haute précision dans la réception et l'exécution des tâches.
- Rapidité dans le traitement des données.
- Ils donnent des bons résultats.

Dans ce chapitre, nous parlerons en détail de tous les composants électriques et électroniques que nous avons utilisés pour contrôler notre système, et garantir de bons résultats à la fin.

Pour chaque système automatique, de nombreux matériaux et outils sont utilisés afin d'assurer une bonne méthode de travail et de bons résultats, et parmi ce que nous utilisons:

- Les contrôleurs (microcontrôleurs, Arduino, Raspberry pi, API ...)
- les capteurs (température, humidité, lumière ...)
- les actionneurs (moteurs, Vérins ...)
- les éléments de transmission des informations (câbles, Bluetooth, wifi ...)
- les interfaces de control et de supervision (applications, pages web, ...)

Dans notre projet, nous avons choisi une variété de composants, que nous expliquerons a commencer par le processeur (un microcontrôleur dans notre cas) jusqu'au dernier composant du projet.

II.2 Le microcontrôleur Raspberry pi :

Sorti en 2012, l'ordinateur mono carte Raspberry Pi est devenu en quelques années l'un des outils les plus populaires dans le domaine du développement. Les bricoleurs et les nouveaux programmeurs du monde entier utilisent le mini-ordinateur pour réaliser les projets les plus fous comme des machines à café à commande vocale ou des consoles de jeux vidéo. [18]

Le Raspberry Pi est un nano-ordinateur mono carte à processeur ARM conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi. [15]

Cet ordinateur, de la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux, notamment Debian, et des logiciels compatibles. Mais il fonctionne également avec le système d'exploitation Microsoft Windows : Windows 10 IoT Core

Il est fourni nu, c'est-à-dire la carte mère seule, sans boîtier, alimentation, clavier, souris ni écran, dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Le Raspberry Pi est utilisé par des « makers » du monde entier car son prix le rend très attirant. [15]

Le Raspberry Pi est bien un nano-ordinateur et non une carte électronique comme le propose Arduino.



Figure II. 1: une image qui illustre quelques modèles de Raspberry pi

II.2.1 Les modèles de Raspberry pi :

II.2.1.1 Modèles A

▪ **Modèle 1 A**

- ✓ Processeur : ARM1176JZF-S (ARMv6) 700 MHz Broadcom 2835 (dispose d'un décodeur Broadcom VideoCore IV, permettant le décodage H.264 FullHD 1080P et d'un VFPv2 pour le calcul des opérations à virgule) ;
- ✓ RAM : 256 Mo ;
- ✓ 2 Sorties vidéo : Composite et HDMI ;
- ✓ 1 Sortie audio stéréo Jack 3,5 mm (sortie son 5.1 sur la prise HDMI) ;
- ✓ Unité de lecture-écriture de carte mémoire : SDHC / MMC / SDIO ;
- ✓ 1 Port USB 2.0 ;
- ✓ Prise pour alimentation Micro-USB (consommation : 400 mA + périphériques) ;
- ✓ Des entrées / sorties supplémentaires sont accessibles directement sur la carte mère via des pins 3v3⁶¹ : GPIO, S2C, I2C, SPI ;
- ✓ API logicielle vidéo : OpenGL : version embarquée OpenGL ES 2.0 ;
- ✓ Décodage vidéo : 1080p30 H.264 high-profile. [19]

▪ **Modèle 1 A+**

- ✓ Plus petit : 65 mm de long (contre 86 mm)
- ✓ Lecteur de carte microSD en lieu et place du lecteur SD
- ✓ GPIO 40 broches
- ✓ Nouveau chipset audio
- ✓ Consommation électrique moindre [19]

II.2.1.2 Modèles B

Il existe plusieurs révisions du modèle B

▪ **Modèle 1 B**

Spécificités du modèle B rev1 :

- ✓ 2 ports USB 2.0 au lieu de l'unique port du modèle A, mais sur un seul bus, via le composant SMSC LAN9512 ;
- ✓ 1 port réseau Fast Ethernet (10/100 Mbit/s) via le même composant SMSC.

Spécificités du modèle Rev1 + ECN0003 :

- ✓ Suppression des fusibles protégeant les sorties USB
- ✓ Suppression de la diode D14, qui pouvait provoquer des interférences avec des périphériques possédant une broche CEC, lorsque le Raspberry restait connecté sans être alimenté. [19]

Spécificités du Raspberry Pi B Rev2 :

- ✓ Implantation du reset (en reliant les broches 1 et 2 de P6)
- ✓ Support JTAG (deux broches GPIO interchangées)
- ✓ Support I2C (canaux primaire et secondaire inversés)
- ✓ Suppression de quatre signaux GPIO utilisés pour l'identification de version, et réaffectation à d'autres rôles
- ✓ SMSC +1V8
- ✓ Deux trous de fixation
- ✓ Correction du marquage des LED sur la platine. [19]

Spécificités du Raspberry Pi B 512 Mo :

- ✓ Prise pour alimentation micro-USB (consommation : 700 mA) ;
- ✓ La RAM passe à 512 Mo (au lieu de 256 Mo sur les modèles précédents) ;
- ✓ Le modèle est estampillé avec la référence 4G en lieu et place de l'ancienne référence 2G.

▪ **Modèle 1 B+**

Ce modèle est annoncé en juillet 2014.

Différences par rapport au modèle initial :

- ✓ GPIO 40 broches
- ✓ 4 ports USB 2.0 et meilleur comportement en cas de surcharge
- ✓ micro SD
- ✓ réduction de consommation de 3,5 W à 3 W
- ✓ meilleur circuit audio

- ✓ suppression de la prise RCA au profit d'une prise mini-jack 4 points, comprenant une sortie sonore et vidéo [19]

- **Raspberry Pi 2 modèle B**

Le 2 février 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 2, plus puissant, il est équipé d'un processeur Broadcom BCM2836, quatre cœurs ARMv7 à 900 MHz, accompagné de 1 Go de RAM.

- **Raspberry Pi 3 modèle B**

Le 29 février 2016, pour le quatrième anniversaire de la commercialisation du premier modèle, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 3. Comparé au Pi 2, il dispose d'un processeur Broadcom BCM2837 64 bit à quatre cœurs ARM Cortex-A53 à 1,2 GHz, d'une puce Wifi 802.11n et Bluetooth 4.1 intégrée. Il possède les mêmes dimensions et connectiques que les modèles 2 et B+. La vitesse d'horloge est 33 % plus rapide que le Pi 2, ce qui permet d'avoir un gain d'environ 50-60 % de performance en mode 32 bits. Il est recommandé d'utiliser un adaptateur de 2,5 A. Tous les travaux et tutoriels du Pi 2 sont parfaitement compatibles avec le Pi 3. [19]

II.2.1.3 Caractéristiques du Raspberry Pi utilisé 3B+:

Le Raspberry Pi est un nano-ordinateur de la taille d'une carte de crédit. . Sur ce petit bout de carte électronique, il y a :

- ✓ Carte mère Raspberry Pi 3 type B+
- ✓ Processeur Broadcom BCM2837B0 Cortex-A53 64-bit SoC @ 1,4 GHz
- ✓ RAM : 1 Go LPDDR2 SDRAM
- ✓ Connexions sans fil : Bluetooth 4.2 BLE, Wi-Fi Dual Band b/g/n/ac
- ✓ Connexion filaire: Gigabit Ethernet
- ✓ Décodage et encodage H.264 1080p30
- ✓ Support PoE (PoE HAT non fourni)
- ✓ Lecteur de carte micro-SD
- ✓ Header GPIO 40 broches
- ✓ Support du boot PXE
- ✓ Port caméra CSI pour connecter la caméra Raspberry Pi
- ✓ Port d'affichage DSI pour connecter l'écran tactile Raspberry Pi
- ✓ 4 x USB 2.0
- ✓ Compatible avec les accessoires du Pi 3 Model B
- ✓ Alimentation 5V / 2.5A (transformateur non fourni) [16]

Raspberry Pi Model B+

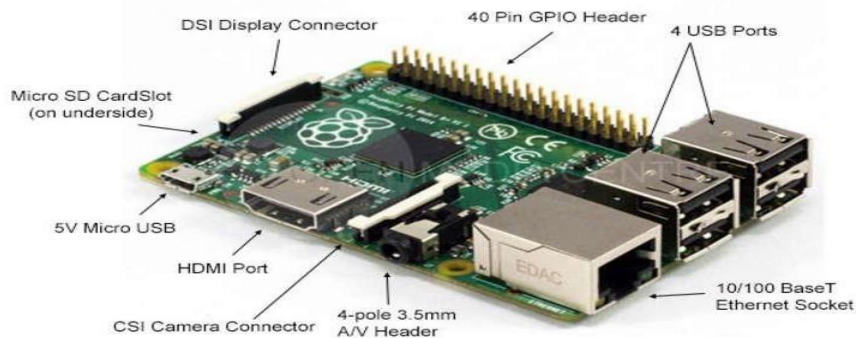


Figure II. 2 : une image qui illustre les différents ports de Raspberry pi B+

Dans notre travail, nous avons opté pour cette dernière version en l'occurrence la Raspberry Pi 3B+.

II.2.2 Accessoires du Raspberry Pi :

II.2.2.1 Alimentation :

Le Raspberry Pi s'alimente sous tension unique de 5 volts, tension sur laquelle il peut consommer jusqu'à 1.8A selon les tâches qu'il exécute. Cette alimentation doit être normalement fournie via le connecteur micro USB placé dans un angle de la carte. En effet, l'utilisation d'un chargeur pour Smartphone équipé d'un micro USB qui délivre une tension de 5 volts avec au minimum 1.8A est suffisante pour alimenter notre Raspberry Pi. [17]

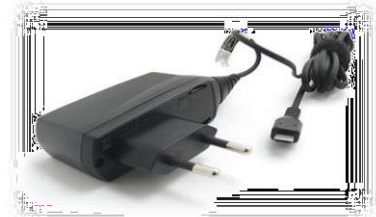


Figure II. 3:Alimentation 5v

II. 2.2.2 Carte microSD :

Pour héberger le système d'exploitation (Raspbian) nécessaire au bon fonctionnement de Raspberry Pi, et tous les fichiers nous aurons besoin d'espace de stockage, c'est pour cela qu'une microSD est requise. La qualité de cette carte, et tout particulièrement sa vitesse, va influencer la vitesse d'écriture des fichiers. Il convient donc de bien la choisir, Pour nous aider, des « classes » sont définies pour nous indiquer la vitesse minimale des cartes en écriture, exprimée en Mo/s. Les cartes vendues en 2010 sont de classes 2, 4, 6 ou 10. Au niveau de la taille, Raspbian étant très léger, il n'y a pas besoin d'une très grosse carte SD. 4Go suffisent, mais il existe sur le marché des cartes de 16 Go. [17]



Figure II. 4: une image qui illustre les classes de la carte microSD

II.2.2.3 Câble HDMI:

Pour le premier démarrage il faut un câble HDMI pour installer le système d'exploitation. On connecte le câble HDMI depuis le Raspberry Pi à l'écran directement, car les écrans modernes ont des ports pour HDMI. [17]

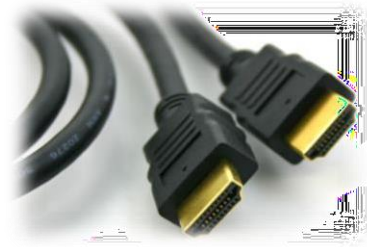


Figure II. 5: un câble HDMI

II.2.2.4 L'écran :

L'écran doit être disponible la première fois pour pouvoir installer le système et visualiser le bureau, cependant, nous pouvons parfois nous passer d'écran, en ayant recours à d'autres moyens comme la communication à distance.



Figure II. 6: Ecran de pc

II.2.2.5 Clavier et souris :

Le Raspberry Pi a besoin d'un clavier et d'une souris pour contrôler le système d'exploitation comme n'importe quel autre ordinateur, cependant certains projets ne nécessitent ni clavier ni souris, mais ils sont nécessaires lors de la première exécution de Raspberry pi. [17]



Figure II. 7: clavier et souris

II.2.2.6 Boîtier

Le Raspberry Pi est livré sans protection. Cela permet de baisser son prix, mais il est plus susceptible à être endommager. Par conséquent, il est préférable d'acheter un boîtier ou bien d'en réaliser un soi-même.



Figure II. 8: Boites de protection Raspberry pi

II.2.3 Systèmes d'exploitation pour Raspberry pi :

Le concept de base reste toujours un système d'exploitation qui gère le système. Pour ce qui est du cas Raspberry, diverses distributions open source et autres applications gratuites sont disponibles et parmi les systèmes les plus connus :

II.2.3.1 Raspbian :

Raspbian est un système d'exploitation libre basé sur la distribution Linux Debian et optimisé pour le matériel de Raspberry Pi. Une première version du portage Debian, initié principalement par les développeurs Mike Thompson et Peter Green, a été publiée peu après la sortie de Raspberry Pi en juin

2012 et depuis lors, Raspbian est considéré comme le système d'exploitation officiel de Raspberry Pi. [18]

II.2.3.2- Kali Linux

Un autre dérivé de Debian qui peut être utilisé dans son édition ARM comme distribution Raspberry Pi est Kali Linux. Le programme open source a été développé par Mati Ahoni et Devon Kearns de la société américaine Offensive Security et publié le 13 mars 2013 dans une première version. La distribution peut être utilisée comme un logiciel de système d'exploitation ordinaire, mais elle est principalement utilisée pour des tests de sécurité et des tests d'intrusion étendus des systèmes et réseaux informatiques. [18]

II.2.3.3 Windows 10 IoT Core :

Avec Windows 10 IoT Core, Microsoft a lancé son premier système d'exploitation pour les appareils IoT comme le Raspberry Pi (2 ou 3) en 2015. L'application s'adresse principalement aux développeurs et aux amateurs qui souhaitent mettre en réseau des objets du quotidien avec Internet ou bien créer de nouveaux objets en réseau. Pour ce faire, Windows 10 IoT Core s'appuie sur l'API interne « Universal Windows Platform » (UWP), qui permet justement d'écrire des applications pour ses propres appareils. L'édition communautaire gratuite de Microsoft Visual Studio sert de logiciel de développement. De plus, le système d'exploitation propriétaire de Raspberry Pi dispose du cryptage Bitlocker et des fonctions « Secure Boot », qui ont été adoptés à partir de la version de bureau. Grâce au support de la modulation de largeur d'impulsion (PWM), le logiciel du système peut également être utilisé pour la commande de moteurs électriques, parmi d'autres choses.

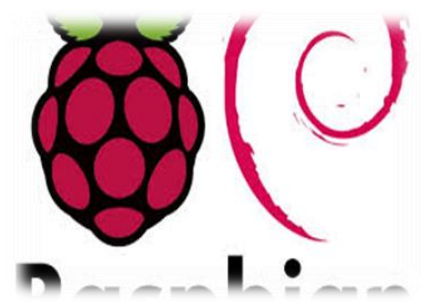


Figure II. 9: Logo système d'exploitation Raspbian

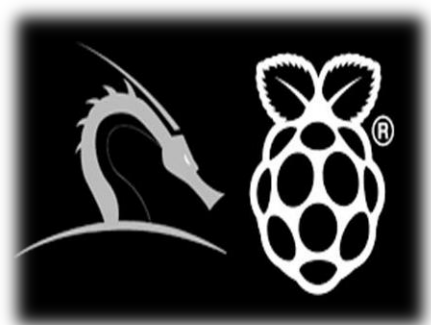


Figure II. 10: Logo système d'exploitation Kali linux



Figure II. 11: Logo système d'exploitation Windows 10 IoT core

II.2.3.4 Ubuntu Core :

Ubuntu est l'une des distributions Linux les plus populaires depuis des années : Ubuntu convient aussi bien comme système d'exploitation pour un ordinateur personnel que pour un serveur, il est également utilisé sur les stations spatiales internationales et est l'unité centrale d'exploitation du BYU Mars Rover. Le logiciel, basé sur Debian et développé par Canonical depuis 2004, se caractérise principalement par son haut degré d'adaptabilité et de convivialité.



Figure II. 12: Logo système d'exploitation Ubuntu

Sous le nom d'Ubuntu Core, les développeurs ont publié en 2014 une variante qui représente une modification minimaliste de l'édition serveur et peut également être utilisée comme système d'exploitation Raspberry Pi. [18]

II.2.4 Fonctionnalité et connectique du Raspberry Pi 3B+

On peut pratiquement faire avec un Raspberry Pi tout ce que l'on peut faire avec un ordinateur de bureau sous Linux, à quelques exceptions près. Le Raspberry Pi utilise une carte SD à la place d'un disque dur, bien que l'on puisse le brancher à un disque dur USB. On peut modifier des documents bureautiques, surfer sur internet, jouer à des jeux, etc. Le Raspberry Pi est un appareil extrêmement souple qu'on peut utiliser dans toutes sortes de situations, que ce soit pour remplacer un ordinateur de bureau, profiter d'un media center ou contrôler un système à l'aide d'un ordinateur embarqué.[38]

II.2.5 Connecteurs et ports disponibles :

❖ HDMI :

« High Definition Multimedia Interface » permet de relier le Raspberry Pi à un dispositif compatible : écran LCD ou un vidéoprojecteur...

❖ GPIO :

Est l'acronyme anglais de "General Purpose Input Output" ce que l'on ne traduirait pas Entrée/sortie pour usage général. Le connecteur GPIO dispose actuellement de différents types de connexions. Elles sont :

- * De vraies broches GPIO avec lesquels vous pouvez allumer/éteindre des LED, etc.
- * Des broches pour une interface I2C qui permet de se connecter sur du matériel en utilisant uniquement 2 broches/pins de contrôle.
- * Une interface SPI 1 pour les périphériques SPI, à concept similaire à I2C 2 mais avec un standard différent.
- * Les broches Rx et Tx pour la communication avec les périphériques séries.
- * De plus, certaines broches peuvent être utilisées en :
 - PWM (acronyme de "Pulse Width Modulation" pour modulation de largeur d'impulsion) permettant le contrôle de puissance
 - PPM (acronyme de "Pulse Position Modulation" pour modulation positionnelle d'impulsion) permettant de contrôler des servo-moteurs.

Raspberry Pi GPIO Header				
Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I2C)		DC Power 5v	04
05	GPIO03 (SCL1 , I2C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN5) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40




Figure II. 13: Une image qui illustre les entrées/sorties d'un Raspberry pi 'GPIO'

❖ Port USB 2.0 :

Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques, et il y a 4 ports USB. Le Raspberry Pi fonctionne avec pratiquement n'importe quels claviers et souris USB. Il est également possible d'utiliser la plupart des claviers et des souris sans fil (ceux qui marchent avec un dongle que l'on branche sur un port USB).

❖ Port Ethernet :

C'est un port qui correspond au protocole international Ethernet de réseau local à commutation de paquets.

❖ Un slot pour cartes microSD:

Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permet de connecter la mémoire externe.

Nous pouvons utiliser notre propre carte SD dans le Raspberry Pi, mais elle devra être préparée avec une image disque d'un système d'exploitation.

II.3 Les capteurs :

Dans de nombreux domaines (industrie, recherche scientifique, services, loisirs...), on a besoin de contrôler des paramètres physiques (température, force, position, vitesse, luminosité, etc...). Le capteur est l'élément indispensable à la détection de ces grandeurs physiques.

Un capteur est un organe de prélèvement d'informations qui élabore, à partir d'une grandeur physique, une autre grandeur physique de nature différente (souvent électrique). Cette grandeur représentative de la grandeur prélevée est utilisable à des fins de mesure ou de commande. [20]

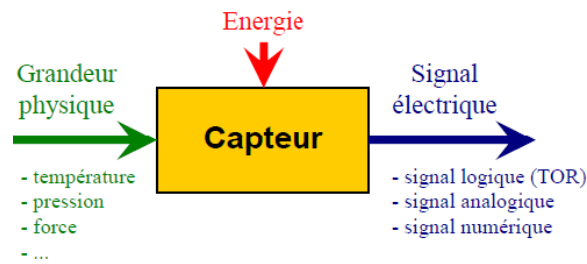


Figure II. 14: Principe de fonctionnement d'un capteur

II.3.1 Classification des capteurs :

On peut classer les capteurs en 3 groupes en fonction de la nature de l'information délivrée en sortie :

➤ Capteurs analogiques :

La sortie est une grandeur électrique dont la valeur est une fonction de la grandeur physique mesurée par le capteur. La sortie peut prendre une infinité de valeurs continues. Le signal des capteurs analogiques peut être du type : sortie tension, sortie courant, règle

graduée, cadran, jauge (avec une aiguille ou un fluide). [21]

Quelques capteurs analogiques typiques :

- capteur à jauge de contrainte ;
- LVDT ;
- thermocouple.

➤ Capteurs numériques :

La sortie est une séquence d'états logiques qui, en se suivant, forment un nombre. La sortie peut prendre une infinité de valeurs discrètes. Le signal des capteurs numériques peut être du type : train d'impulsions, avec un nombre précis d'impulsions ou avec une fréquence précise, code numérique binaire, bus de terrain.

Quelques capteurs numériques typiques :

- codeur rotatif incrémental.
- codeurs référentiels AA34. [21]

➤ Capteurs logiques :

Ou capteurs TOR. La sortie est un état logique que l'on note 1 ou 0. La sortie peut prendre ces deux valeurs. Il y a notamment 4 types de capteurs logiques : courant présent/absent dans un circuit, potentiel, souvent 5 V/0 V, DEL allumée/éteinte, signal pneumatique (pression normale/forte pression).

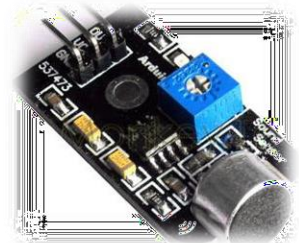


Figure II. 15: exemple d'un capteur analogique "capteur de son"



Figure II. 16: exemple d'un capteur numérique "encodeur"



Figure II. 17: exemple d'un capteur logique "fin de cours"

Quelques capteurs logiques typiques :

- capteurs de fin de course.
- capteurs de rupture d'un faisceau lumineux. [21]

Type de détection :

Détection avec contact (le capteur doit entrer en contact physique avec un phénomène pour le détecter).

Détection sans contact (le capteur détecte le phénomène à proximité de celui-ci).

Caractéristiques

Un capteur est caractérisé selon plusieurs critères dont les plus courants sont :

- la grandeur physique observée ;
- son étendue de mesure (gamme de mesure) ;
- sa sensibilité.
- sa résolution.
- sa précision.
- sa reproductibilité.
- sa linéarité.
- son temps de réponse.
- sa bande passante.
- son hystérésis.
- sa gamme de température d'utilisation.

Remarque : Pour utiliser un capteur dans les meilleures conditions, il est souvent utile de pratiquer un étalonnage et de connaître les incertitudes de mesure relatives à celui-ci. [21]

II.3.2 Les capteurs utilisés :

Dans notre projet, nous avons utilisé une variété de capteurs, et nous les mentionnerons un par un :

II.3.2.1 Capteur de température DHT22 :

Le DHT22 est un capteur numérique de base, à faible coût permettant de mesurer de manière efficace la température et l'humidité de l'air ambiant grâce à sa combinaison de deux en un d'un capteur d'humidité capacitif et d'une thermistance.

De plus il se connecte facilement à votre microcontrôleur préféré grâce à un signal numérique sur un seul fil.

Le seul véritable inconvénient de ce capteur est que vous ne pouvez obtenir de nouvelles données à partir d'une fois par secondes environ.

Ce capteur fournit une information numérique en sortie, information communiquée régulièrement. Nul besoin de broche analogique, une simple broche digitale fera l'affaire.

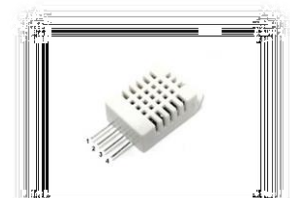


Figure II. 18: capteur de température et humidité DHT22

Utiliser le capteur est vraiment simple: connecter la première broche à gauche sur une alimentation 3V/5V, la seconde broche sur l'entrée numérique du microcontrôleur et la broche de droite sur la masse/GND. Une résistance pull-up de 4.7KOhms à 10 KOhms doit être placée entre la broche de donnée et VCC. [22]

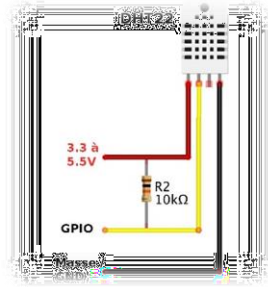


Figure II. 19: méthode de câblage de DHT22

II.3.2.2 Capteur de température Ds18B20 :

C'est le capteur pratique si vous devez réaliser une mesure de température éloignée ou dans des conditions humides. Bien que le capteur peut mesurer jusqu'à 125°C, Comme ce capteur est numérique, vous pouvez l'utiliser sur une longue distance sans détérioration de l'information. Ce capteur One Wire est précis à 0,5°C sur une partie de l'étendue de mesure avec une numérisation sur 12 bits issu du CAN interne. Il fonctionne sur la plupart des microcontrôleurs et nécessite qu'un seul pin, vous pouvez même en brancher plusieurs sur le même pin car chaque capteur dispose d'une adresse unique 64 bits écrite en dur en usine. Ce capteur fonctionne aussi bien en 3,3V qu'en 5V. [23]



Figure II. 20: capteur de température DS18B20

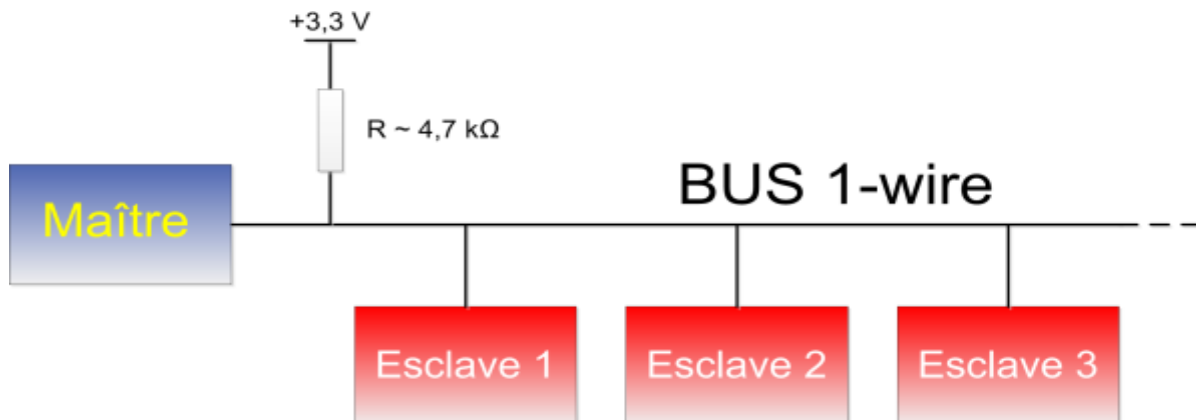


Figure II. 21: mode Bus 1-wire

Le bus 1-wire est basé sur une architecture maître-esclave. Le maître est l'équipement qui contrôle le bus, interroge les périphériques, ou leur envoie des ordres. [35]

Dans notre cas, le maître sera le Raspberry Pi 3 B+ et il n'y aura qu'un esclave, le capteur de température DS18B20.

Sur la figure, on voit que le bus 1-wire est relié au +3,3 V par une résistance de tirage de 4,7 kΩ. En fonction de la longueur des fils de liaison, il sera parfois nécessaire d'ajuster sa valeur. Tous

les appareils sont branchés en parallèle sur le bus et c'est en mettant (ou pas) leur sortie à la masse qu'ils envoient les données au maître.

Le maître initialise le bus en forçant un zéro pendant plus de 480 μ s. Les esclaves lui répondent en mettant eux aussi leur sortie à zéro pendant un certain temps, indiquant ainsi leur présence. [35]

Le maître déclenche une lecture dite ROM et l'esclave renvoie son identifiant unique, gravé lors de la fabrication. Chaque identifiant a une longueur de 48 bits (8 octets) encadrés par un octet indiquant le type de matériel (ici le type est : sonde de température = 28h) et un octet de CRC qui permet au maître de vérifier qu'il a correctement reçu les informations d'identification du composant. Les 48 bits permettent d'individualiser à peu près 280 000 milliards de composants... [35]

Dans notre projet l'utilisation de ce capteur a pour but de mesurer la température dans le bac d'élevage des poissons.

II.3.2.3 Module de détection de lumière LDR :

Comparateur LM393 :

le comparateur LM393 chargé de comparer la tension aux bornes de la LDR avec une tension de référence réglable. Le potentiomètre permet d'ajuster le seuil de déclenchement en fonction du niveau de lumière. Tourné dans le sens horaire, il augmente la sensibilité. Dans le sens antihoraire la sensibilité diminue. Les flèches indiquent le sens de réglage. La LED D1, située à proximité du potentiomètre, s'allume quand le module est alimenté. La LED D2 indique le niveau de sortie sur la broche D0. Elle s'allume quand l'intensité lumineuse dépasse le seuil fixé grâce au potentiomètre. [24]

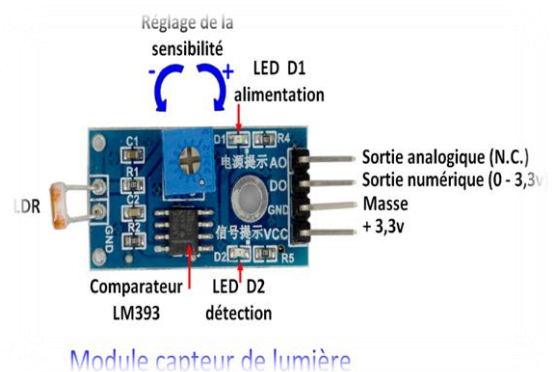


Figure II. 22: le module de détection de lumière

Les broches sont, de haut en bas :

- A0 La sortie analogique, reflet de l'intensité lumineuse reçue par la LDR
- D0 Sortie numérique, passe à 1 (+3,3v ou +5v) lorsque l'intensité lumineuse dépasse le seuil
- GND Masse
- Vcc Alimentation +3,3v

Photorésistance –LDR–:

Une photorésistance est composée d'un semi-conducteur à haute résistivité. Deux électrodes sont séparées par ce matériau photoconducteur (en général du CdS sulfure

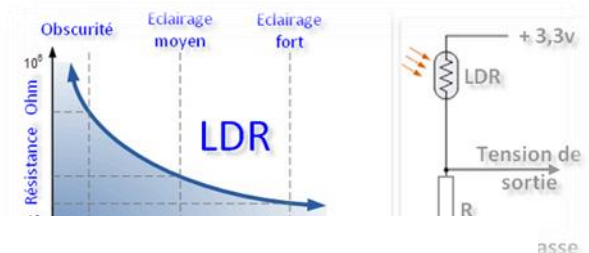


Figure II. 23: Caractéristique du module de détection de lumière

cadmium). La piste de CdS prend la forme d'un ruban présentant de nombreux virages, pour allonger la zone réagissant à la lumière. Ceci améliore la réponse de la photorésistance LDR. [24]

Dans l'obscurité, la résistance d'une LDR est proche de 1 M Ω . Avec un éclairage intense, la résistance chute fortement (quelques K Ω). Un pont diviseur permet de récupérer une tension qui sera directement le reflet de la lumière arrivant sur la LDR.

Comme les GPIO du Raspberry Pi sont des entrées-sorties numériques, ce signal n'est pas directement utilisable. Il faut passer par un "comparateur" de type LM393 pour détecter le niveau de luminosité et mettre la sortie D0 à 1 lorsque la lumière dépasse le seuil (réglé par le potentiomètre). [24]

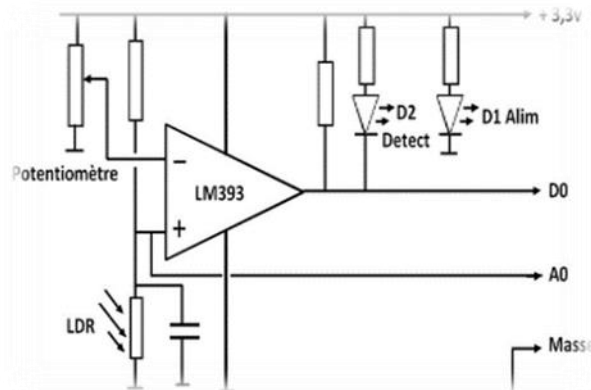


Figure II. 24: schéma électronique de module de détection de lumière

Fonctionnement dans l'obscurité :

Lorsque la LDR est dans l'obscurité, la tension sur l'entrée + est supérieure à celle présente sur l'entrée -. La tension de sortie du comparateur vaut alors +3,3v. C'est cette tension qui est présente sur D0. La LED D2 reçoit sur son anode une tension de 3,3v et sur sa cathode la même tension de 3,3v. Il n'y a aucune différence de tension à ses bornes, elle est donc éteinte. [24]

Fonctionnement à la lumière :

A la lumière, la résistance de la LDR est très faible. L'entrée + est pratiquement à la masse. La tension sur l'entrée + est inférieure à celle présente sur l'entrée -. La tension de sortie du comparateur vaut alors 0v. C'est cette tension qui est présente sur D0. La LED D2 reçoit sur son anode une tension de 3,3v (via une résistance de limitation) et sur sa cathode une tension de 0v, C'est comme si la cathode était reliée à la masse du montage. Un courant circule dans la LED, limité par la résistance en série avec la LED. La LED D2 s'allume.

L'utilisation de ce capteur assure la détection des jours sombres et jours bien éclairés.

II.4 Les actionneurs :

Les actionneurs sont des dispositifs qui permettent à un système automatisé d'effectuer la tâche pour laquelle il a été programmé. Ils constituent, avec les capteurs, la partie opérative d'un système automatisé.

Les actionneurs transforment l'énergie qu'ils reçoivent en un autre phénomène physique (un déplacement, un dégagement de chaleur, une émission de lumière ou de sons...). Tout comme l'être humain, le système automatisé doit être capable de « s'exprimer », c'est-à-dire d'afficher un message ou d'indiquer un point particulier, d'émettre des sons ou encore d'effectuer un mouvement. Il existe donc, pour chacune de ces actions, un type d'actionneur. [25]

II.4.1- classification des actionneurs :

On peut classer les actionneurs suivant différents critères : [25]

- énergie utilisée
 - Pneumatique
 - Hydraulique (eau ou huile)
 - Électrique
 - Mécanique
 - Thermique
 - Combustible fossile

 - phénomène physique utilisable
 - Déplacement ou freinage
 - Chaleur ou Froid
 - Lumière
 - Son

 - principe mis en œuvre
Selon le principe mis en œuvre, les propriétés sont différentes :
 - utilisation de l'énergie pneumatique, avec les propriétés de compression et de dilatation
 - utilisation de l'énergie hydraulique, avec la propriété d'incompressibilité du fluide.
- Dans le cas de l'énergie électrique, les principes sont nombreux :
- effet Joule
 - effet magnétique
 - effet Peltier
 - effet piézoélectrique tel que ceux utilisé pour l'injection de carburant.
 - électroluminescence, incandescence, fluorescence.

II.4.2 Les actionneurs utilisés :

Dans ce travail nous avons utilisé plusieurs actionneurs.

II.4.2.1 Le moteur à courant continu :

Les moteurs courant continus sont des convertisseurs de puissance :

Soit ils convertissent l'énergie électrique absorbée en énergie mécanique lorsqu'ils sont capables de fournir une



Figure II. 25: moteur à courant continu

puissance mécanique suffisante pour démarrer puis entraîner une charge en mouvement. On dit alors qu'ils ont un fonctionnement en moteur.

Soit ils convertissent l'énergie mécanique reçue en énergie électrique lorsqu'ils subissent l'action d'une charge entraînée. On dit alors qu'ils ont un fonctionnement en générateur.

- En mode « moteur », l'énergie électrique est convertie en énergie mécanique.
- En mode « générateur » ou « dynamo », l'énergie mécanique est convertie en une énergie électrique capable de se comporter comme un frein.

Principe de fonctionnement du moteur à courant continu :

Un moteur à courant continu est constitué de deux parties électriques : le stator et le rotor. Lorsqu'on alimente le moteur, il se crée une interaction magnétique qui met le moteur en mouvement. Lorsqu'on inverse le sens de la tension qui alimente le moteur, il tourne en sens inverse. [26]

Le stator :

Le stator d'un moteur à courant continu est la partie fixe du moteur (statique= qui ne bouge pas). Le stator est aussi nommé l'inducteur ou l'excitation on fait passer un courant dans le bobinage du stator et c'est lui qui crée (qui induit) un champ magnétique. Le stator pose le décor pour le rotor qui se retrouve ainsi plongé dans ce champ magnétique. Le stator (inducteur) crée un champ magnétique B appelé champ inducteur, ou encore champ statoriques.

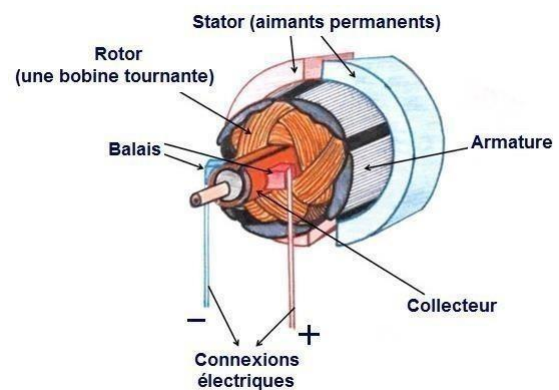


Figure II. 26: comportement d'un moteur à courant continu

Le rotor :

Le rotor est la partie en rotation du moteur. C'est lui qui tourne. Il est constitué du bobinage induit. Il faut alimenter cette bobine pour la transformer en électroaimant qui entrera en interaction avec le stator. Si on n'alimente pas le rotor, il ne serait l'objet d'aucune force et ne tournerait pas.

Un système de frottement spécial permet d'alimenter le rotor : des balais (ou charbons montés sur des ressorts) frottent sur les contacts en rotation : le collecteur. Le collecteur est un ensemble de plages métalliques qui font contact avec les charbons.

Il appartient au rotor. Le frottement des charbons font qu'ils s'usent : ils sont montés sur ressort pour garantir un contact même lorsqu'ils raccourcissent à cause de l'usure

On a utilisé ce moteur pour faire pousser la porte de la serre dans les deux sens ouverture et fermeture.

II.4.2.2- Les Lampes – LED – :

La lampe électroluminescente, dite lampe LED (abréviation de l'anglais Light-Emitting Diode) est un type de lampe électrique qui utilise l'électroluminescence, phénomène optoélectronique issu de la technologie des diodes électroluminescentes.

Avantage des LED :

- Durée de vie plus longue qu'une lampe à incandescence ou fluorescente,
- Faible consommation électrique due à une bonne efficacité lumineuse.
- Sécurité de fonctionnement en très basse tension pour certaines.
- Faible production de chaleur.
- Pas de production d'UV.
- Possibilité de produire une grande variété de couleurs par addition de LED de couleurs différentes.
- Grand choix de la température de couleur pour les LED blanches allant des blancs chauds aux blancs froids. [28]



Figure II. 27: lampes LED

Dans notre système Aquaponie, nous les utilisons pour assurer un éclairage supplémentaire pour les plantes et les poissons, en particulier les jours sombres.

II.4.2.3- La résistance chauffante :

Une résistance chauffante est un matériau conducteur à base de métal présentant une résistance électrique au passage d'un courant qui la porte à une température donnée par effet joule.

Les résistances chauffantes sont étudiées pour délivrer une certaine température soit par rayonnement infrarouge, soit par conduction thermique en fonction de l'application de chauffage.

Certaines résistances sont conçues pour délivrer peu de chaleur afin de maintenir une surface hors gel (gouttières, rampes d'accès, etc.) ou de maintenir une température de base dans un local (câbles de traçage, plancher chauffant électrique). [27]

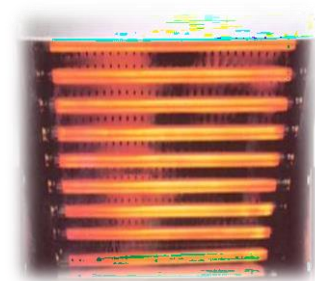


Figure II. 28: Résistance chauffante

II.4.2.4 L'extracteur d'air :

L'extracteur d'air (ou aérateur) consiste en un dispositif dont la vocation est de débarrasser les pièces équipées de leur humidité ainsi que des éventuelles mauvaises odeurs (notamment en cuisine) et de garantir un bon renouvellement de l'air ambiant. Avec un tel équipement, il est donc possible d'obtenir des pièces assainies, délesté de leur pollution. Véritable alternative à la VMC (Ventilation Mécanique Contrôlée) qu'il n'est pas toujours possible de mettre en œuvre, ce système s'avère



Figure II. 29: Extracteur d'air

moins onéreux (aussi bien à l'achat qu'à la pose), plus économe et moins bruyant. [29]

Les ventilateurs extracteurs permettent de forcer l'aération dans les serres aussi lorsque la ventilation naturelle ne permet pas d'atteindre le taux de renouvellement d'air désiré.

Il s'agit d'un besoin essentiel dans la production de cultures et dans les fermes d'élevage.

Ils sont indispensables pour un système cooling avec des panneaux évaporateurs et souvent aussi avec des systèmes de brumisation d'eau afin d'obtenir un certain niveau de réfrigération adéquat.

II.4.2.5 Distributeur de nourriture pour poissons :

Distributeurs automatiques de nourriture pour poissons d'aquarium permettant de programmer et d'assurer le nourrissage lors des absences. Mais en raison du manque de ce composant, nous avons recouru à le fabriquer en utilisant un visse sans fin et un moteur à courant continu.



Figure II. 30: Distributeur d'aliment

II.4.2.6 Pompe à eau :

Pour notre prototype, nous avons besoin d'une petite pompe à eau, que nous allons utiliser pour faire circuler l'eau entre le bac de culture et le bac d'élevage de poissons.



Figure II. 31: pompe à eau

II.4.2.7 Le diffuseur :

Aérateur pour aquarium permettant l'oxygénation et l'amélioration de la vie des poissons et des plantes. Une pompe à air doit être associée à un diffuseur à air pour permettre de créer des remous en surface, évitant ainsi la formation d'un film gras et apportant l'oxygène nécessaire. [30]



Figure II. 32: le diffuseur

II.5 Les préactionneurs :

Un pré actionneur est un constituant dont le rôle est de distribuer l'énergie utile aux actionneurs. Et parmi les préactionneurs utilisés dans notre prototype :

II.5.1 les relais :

Le relais électromagnétique est un élément qui assure la fonction de commutation. Il distribue de l'énergie électrique à un élément de puissance (moteur, résistance chauffante, lampe...)

Le relais est constitué d'un noyau en matériau magnétique autour duquel est enroulé un bobinage de cuivre (électro-aimant), d'un levier et d'un jeu de contacts. Quand le bobinage est mis sous tension, le levier en métal pivote et le contact est fermé.

Dans notre prototype on a utilisé les relais pour alimenter :

- Le distributeur d'alimentation.
- L'éclairage.
- La pompe à eau.
- Le diffuseur.
- La résistance chauffante.



Figure II. 33: Relais électromagnétique

II.5.2 L'interface de puissance L298 :

L'interface de puissance L298 est un module comportant un double pont en H capable de piloter deux moteurs à courant continu ou un moteur pas à pas sous 36 V et consommant 2 A, des diodes de roue libre pour protéger le L298, un régulateur de tension 5 V, des condensateurs de filtrage, des borniers à vis pour y brancher les moteurs et des broches pour y relier le contrôleurs . Le L298 possède trois entrées de commande par pont en H pour piloter le moteur. [30]

Cette interface utilise pour but de alimenté le moteur à courant continu de la porte de la serre, et mm le ventilateur d'aération de serre.



Figure II. 34: module L298

II.6 Les logiciels, les sites web et les environnements de développements utilisent dans la préparation de l'interface de contrôle à distance :

II.6.1 Android studio :

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux. [30]



Figure II. 35: logo de logiciel Android studio

Fonctionnalités :

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur. [30]

II.6.2 Firebase :

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application (Android, iOS, JavaScript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel. Lancé en 2011 sous le nom d'Envolv, par Andrew Lee et par James Templin, le service est racheté par Google en octobre 2014. Il appartient aujourd'hui à la maison mère de Google Alphabet.



Figure II. 36: logo de site web
Firebase

Toute l'implémentation et la gestion serveur de Firebase est à la charge exclusive de la société Alphabet. Les applications qui utilisent Firebase intègrent une bibliothèque qui permet les diverses interactions possibles. [32]

On a utilisé Firebase comme une base des données au temps réel qui va relier notre application avec le contrôleur.

II.6.3- Kotlin :

Kotlin est un langage de programmation orienté et fonctionnel, avec un typage statique qui permet de compiler pour la machine virtuelle Java, JavaScript, et vers plusieurs plateformes en natif (grâce à LLVM). Son développement provient principalement d'une équipe de programmeurs chez JetBrains basée à Saint-Petersbourg en Russie (son nom vient de l'île de Kotlin, près de St. Pétersbourg).



Figure II. 37: logo de langage
de programmation Kotlin

Google annonce pendant la conférence Google I/O 2017 que Kotlin devient le second langage de programmation officiellement pris en charge par Android après Java. Le 8 mai 2019, toujours lors de la conférence Google I/O, Kotlin devient officiellement le langage de programmation voulu et recommandé par le géant américain Google pour le développement des applications Android. [34]. On utilise ce langage pour développer l'application Android.

II.6.4 Python :

Python est un langage de script de haut niveau, structuré et open source. Il est multi-paradigme et multi-usage.

Développé à l'origine par Guido van Rossum en 1989, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative ; c'est d'ailleurs une des raisons qui lui vaut son



Figure II. 38: logo de langage de
programmation python

appellation de « langage agile ».

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en C, C++ ou Fortran. Il est également apprécié pour la clarté de sa syntaxe, ce qui l'oppose au langage Perl. [34]

L'utilisation de python dans notre projet est pour but de rédiger des programme a exécuté sur Raspberry pi, ces programmes capables de faire notre système à l'état de marche.

II.7 Conclusion :

Dans ce chapitre, on a illustré en détaille tous les composants ainsi que les logiciels utilises dans ce travail:

- les composants et éléments électroniques et électriques.
- l'environnement de développement des applications.
- différents langages de programmation.

Chapitre III

III.1 Introduction :

Comme déjà indiqué dans les chapitres précédents, nous avons choisi le système de culture en eaux profondes comme étant un modèle de référence afin de fabriquer notre propre système aquaponique. La simplicité de ce modèle ainsi que la disponibilité de ses composants en Algérie nous ont été avantageux.

On va diviser le travail en trois parties :

- Le prototype.
- Partie opérative (actionneurs-capteurs-câblages)
- Partie commande (contrôleur – programmes)

III.2 Conception et Réalisation du système aquaponique

III.2 .1 Fabrication de prototype :

Au début on va décrire notre système, il se compose de deux parties:

- Lit de culture aquaponique
- Bac d'élevage des poissons

Nous utiliserons des boîtes en plastique à la place des bacs en verre, afin de réduire les coûts et éviter d'éclatement du verre, notamment lors des déplacements d'un endroit à un autre.

Nous fabriquerons une petite serre en verre, le rôle de cette serre est de protéger nôtres plantes et à favoriser la croissance des cultures en créant des conditions climatiques plus favorables que le climat local, et même de protéger notre bac de culture.

Après avoir coupé le verre aux mesures (longueur : 50cm, hauteur : 40cm, largeur : 40cm), nous façonnons la serre avec de la silicone, puis la posons sur un support solide.

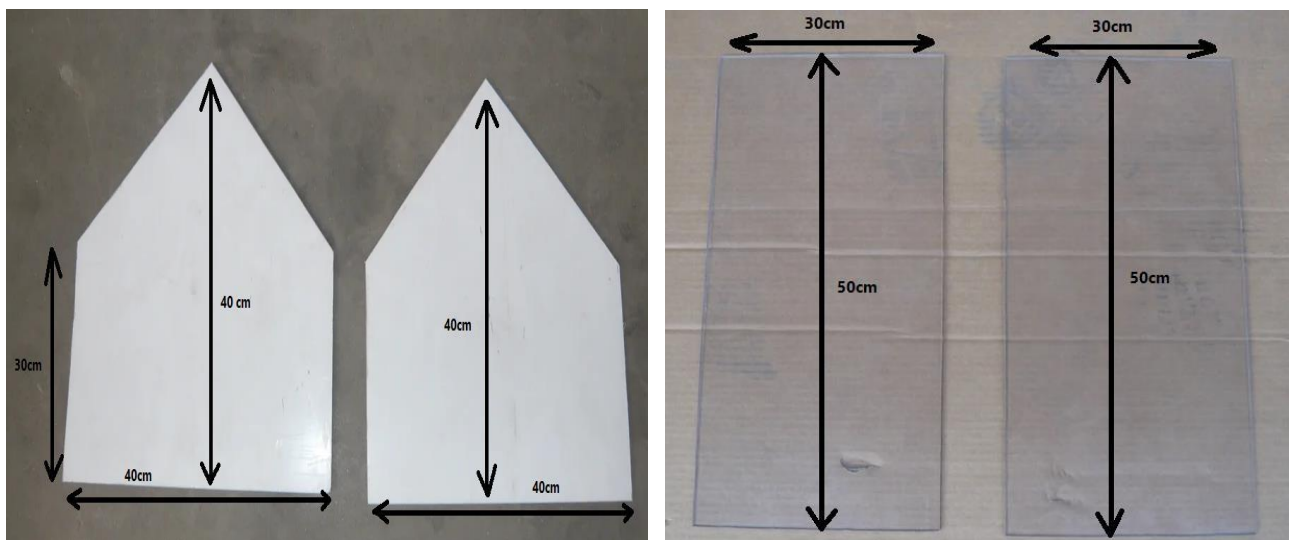


Figure III 1: les démentions de notre prototype

Nous faisons un trou dans l'un des murs de la serre pour le système de ventilation et n'oublions pas l'emplacement de la porte, qui sera une porte coulissante

À l'intérieur de la serre, nous mettons deux boîtes en plastique, que nous utiliserons une comme un lit de culture et l'autre comme un bac d'élevage de poissons.

Le lit va porter des plantes plantées dans des gobelets en plastique, le fond de gobelets sera en contact avec l'eau.



Figure III 2: le prototype à l'état final

III.2.3 Partie opérative :

Dans cette partie, on va placer chaque capteur et chaque actionneur dans sa place.

III.2.3.1- Le capteur DHT22 :

Le capteur de température et d'humidité DHT22 sera à l'intérieur de la serre, son rôle est de relever la valeur de la température et d'humidité dans la serre. La figure suivante illustre le schéma de câblage.

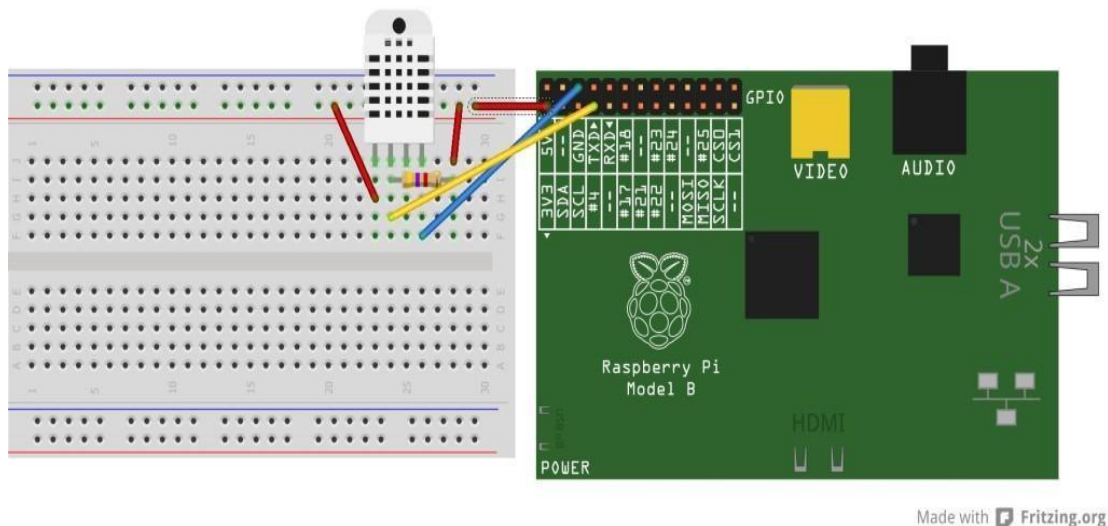


Figure III 3: Interconnexion DHT11 et Raspberry pi

Les branchements sont les suivants :

- Le Pin 1 de la sonde va vers l'alim 3,3V du Raspberry
- Le Pin 2 va vers le pin 3 du Raspberry Pi
- Le Pin 3 va vers le Ground du Raspberry Pi

III.2.3.2 le capteur LM393 :

Le capteur détecteur de lumière sera au plafond de la serre, pour être devant la lumière du soleil et détecte sa présence et son absence.

La connexion du module au Raspberry Pi nécessite 3 fils :

- La broche Vcc est connectée au +3,3v sur le GPIO
- La broche GND est connectée à la masse sur le GPIO
- La sortie D0 est connectée au pin 5 de GPIO

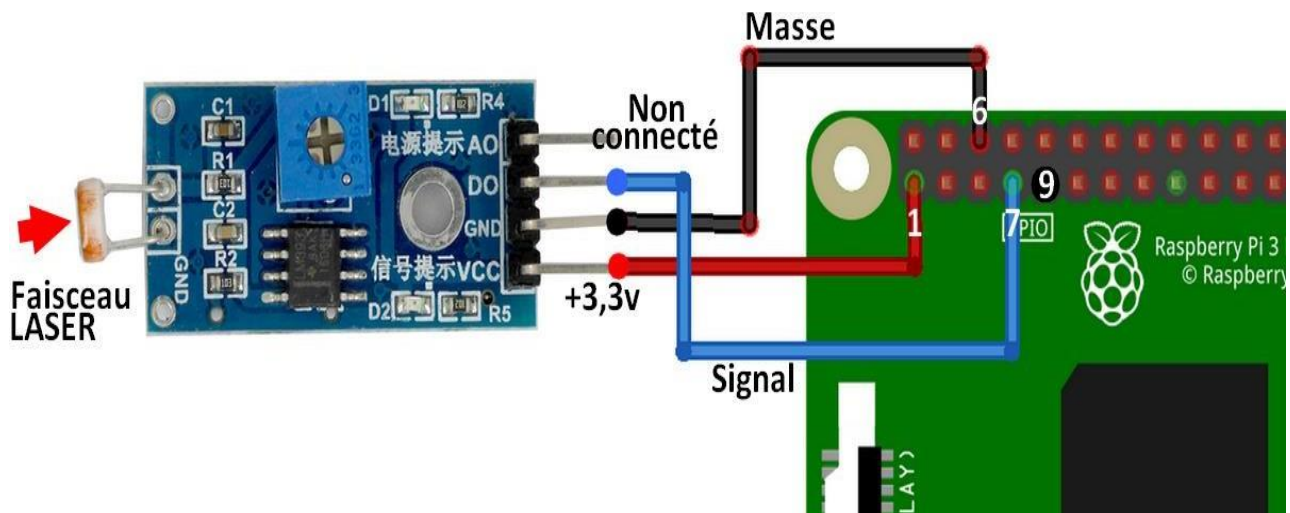


Figure III 4.: interconnexions le capteur LM393 et le Raspberry pi

III.2.3.3 le capteur Ds18b20 :

Ce capteur est isolé, son utilisation est donc dans les endroits humides, alors on va le poser dans le bac d'élevage de poisson.

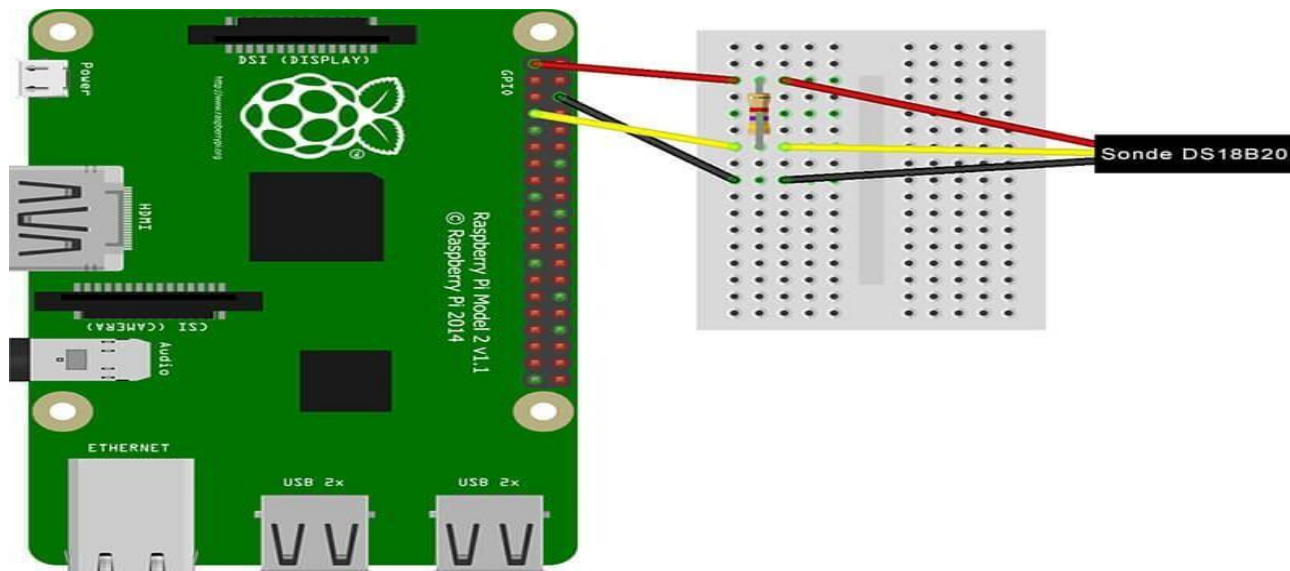


Figure III 5: interconnexion Ds18B20 et Raspberry pi

On va donc logiquement connecter la broche GND à la masse du Raspberry Pi, la broche VCC au +3.3V, et la broche signal au pin 7 de GPIO. Mais il faudra également connecter la broche signal à VCC, via une résistance de 4.7KOhms.

III.2.3.4 le ventilateur :

C'est un ventilateur destiné à créer un courant d'air à l'intérieur de la serre ayant pour but de changer l'air et refroidir le climat au cas de hautes températures.

Ce ventilateur est piloter par le circuit de puissance l298, car ce circuit est habilité a commander la vitesse du ventilateur. [30]

Dans ce cas, nous n'avons besoin que d'un seul sens de rotation, avec une alimentation de 12 v.

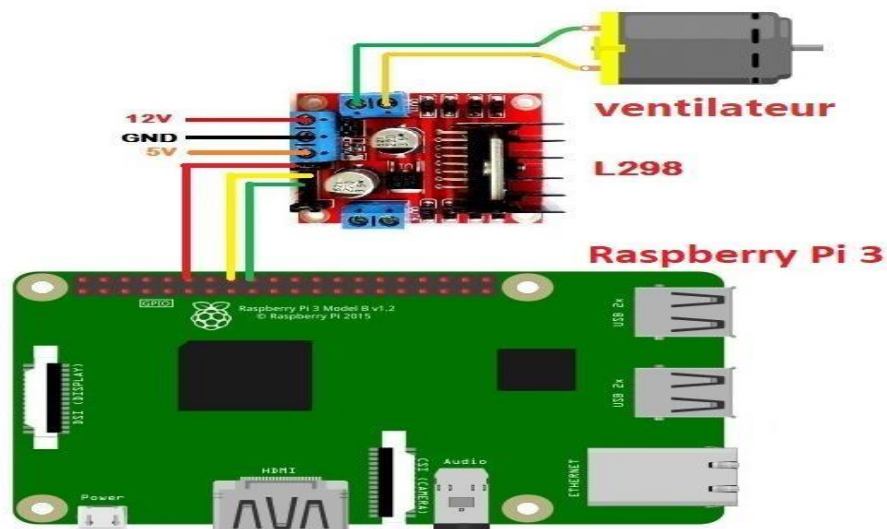


Figure III.6: interconnexion L298 et Raspberry et ventilateur

Les branchements sont les suivants :

- ENA avec le pin 22
- IN1 avec le pin 18
- IN2 avec le pin 16

III.2.3.5- Moteur DC :

Le rôle de moteur dans notre système est de faire pousser la porte de la serre dans les deux sens (ouverture – fermenteur). A cet effet, le montage devient comme suit :

- ENB avec le pin 37
- IN3 avec le pin 35
- IN4 avec le pin 33

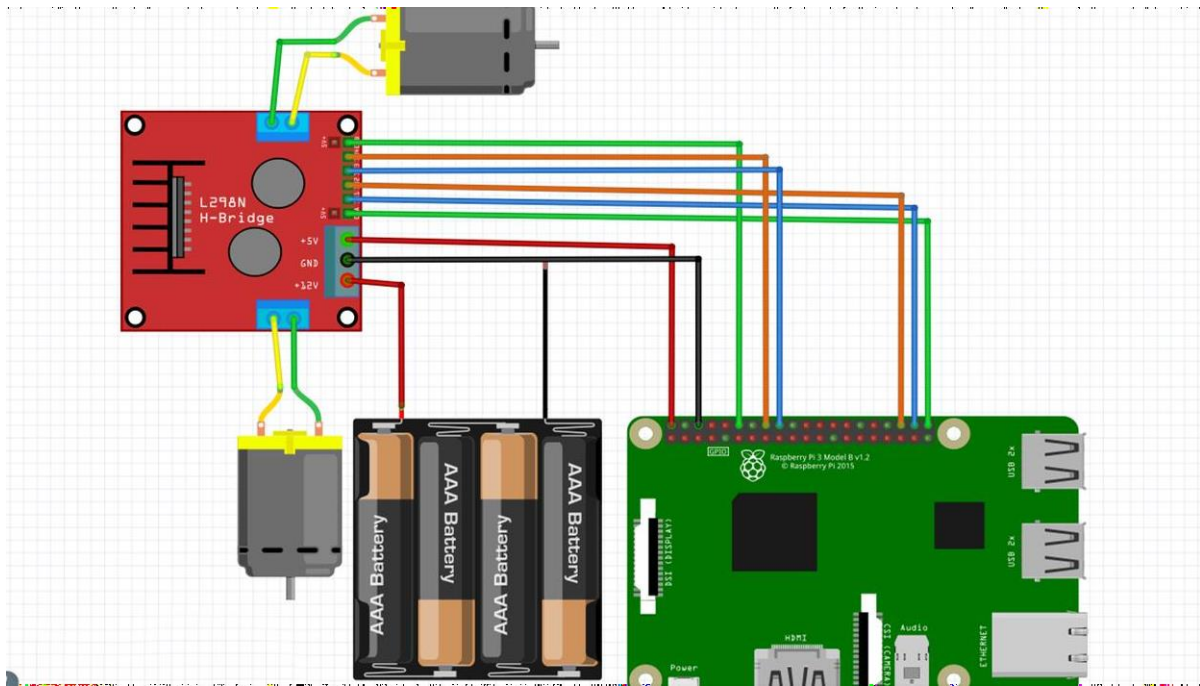


Figure III 6: Interconnexion Raspberry pi et L298 et le moteur DC

III.2.3.6 Actionneurs commandés par relais :

La Lampe, la résistance chauffante, distributeur de nourriture, pompe à eau et le diffuseur sont des actionneurs, ils sont commandés par les relais, chaque relais commandé par un pin de

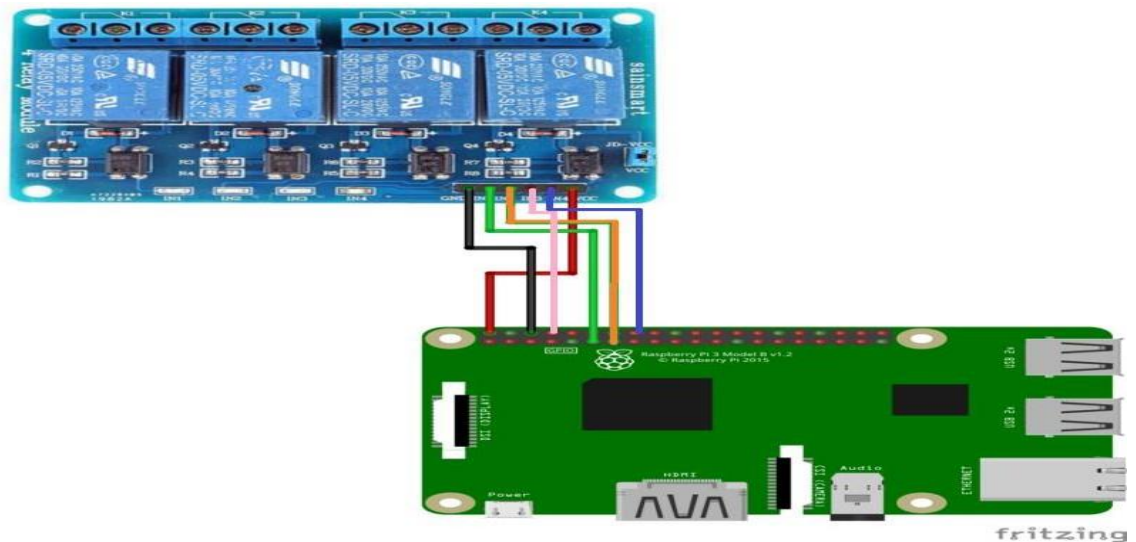


Figure III 7: brochage relais Raspberry pi

Raspberry pi. Le brochage de ces relais devient comme sa :

- Vcc des relais avec le 5v de Raspberry pi
- Ground des relais avec le GND de Raspberry pi
- Pin1 avec le pin 32 de GPIO

- Pin2 avec le pin 36 de GPIO
- Pin3 avec le pin 26 de GPIO
- Pin4 avec le pin 38 de GPIO

Dans l'autre côté (coté puissance) les actionneurs doivent connecter comme suit:

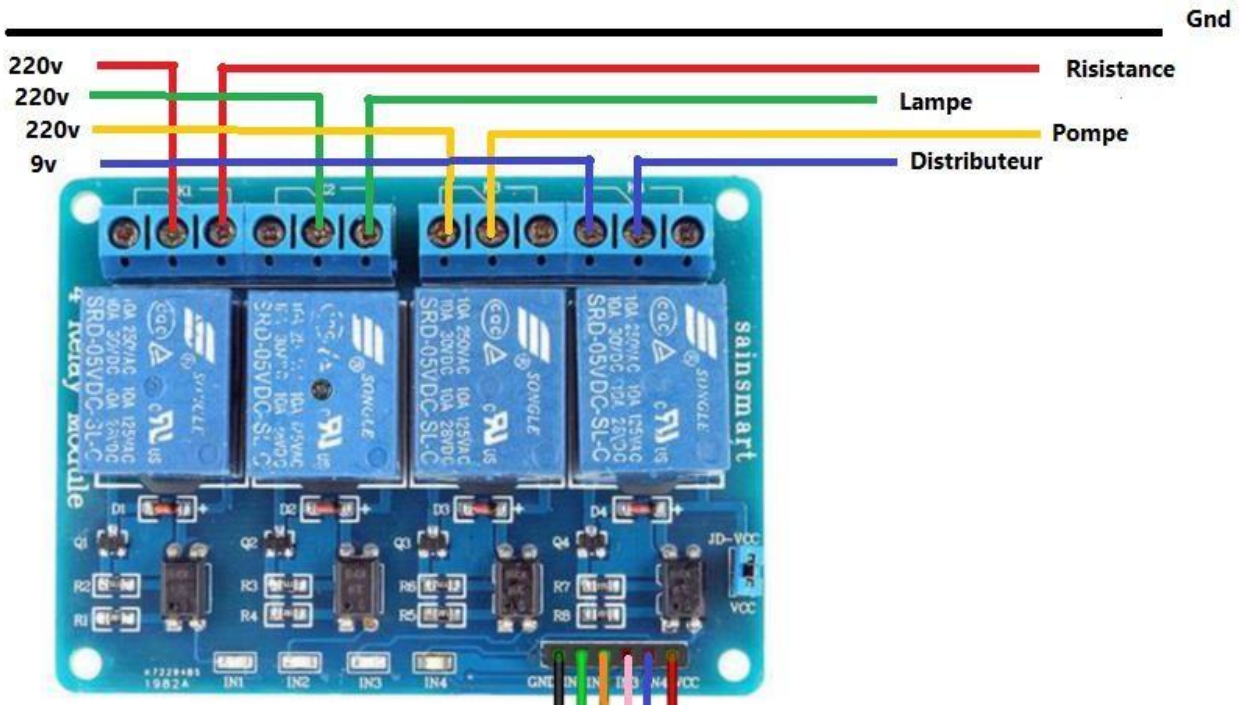


Figure III 8: brochage des relais des actionneurs

Il existe deux contacts dans chaque relais :

- Normalement fermé
- Normalement ouvert

Dans notre cas on a utilisé normalement ouvert c'est-à-dire, dans l'état de repos de relais tous les actionneurs sont éteints.

III.3 Partie commande :

On va diviser la partie commande en trois parties :

- Préparation de la base des données.
- Programmation d'application.
- Programmation de contrôleur.

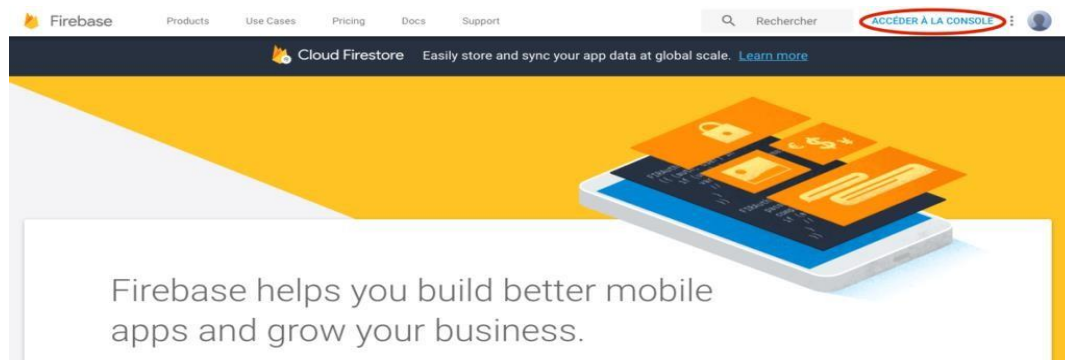
III.3.1 Préparation de la base des données :

Comme on a déclaré dans le chapitre précédant, Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application, alors on va l'utiliser pour notre application Android, on va créer une base de données en temps réel, cette base de données sera un lien entre l'application et le contrôleur, nous l'utiliserons pour échanger des données entre les deux à distance.

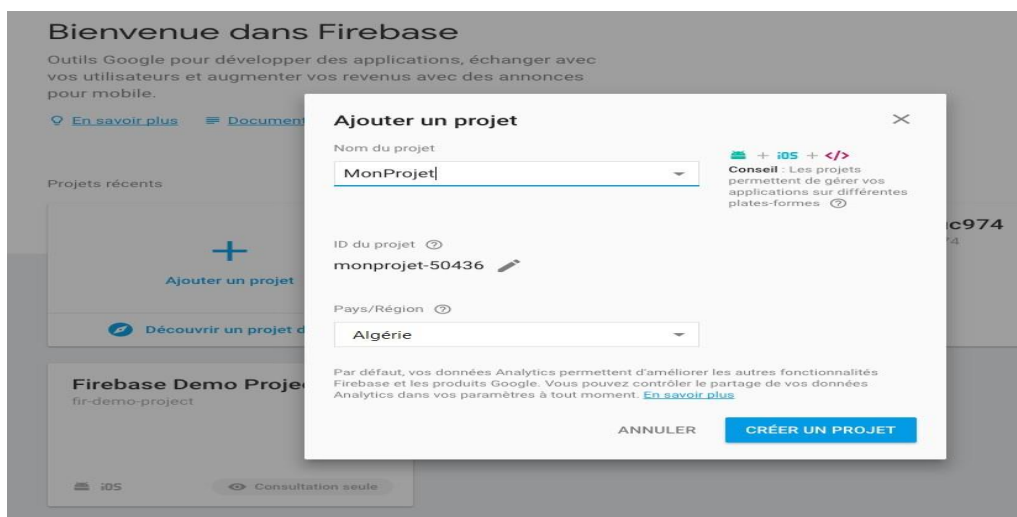
En premier lieu, il faut un email, cet email sera utilisé pour la création de projet Google Firebase.

On connecte sur Console Google Firebase par : <https://console.firebase.google.com/>

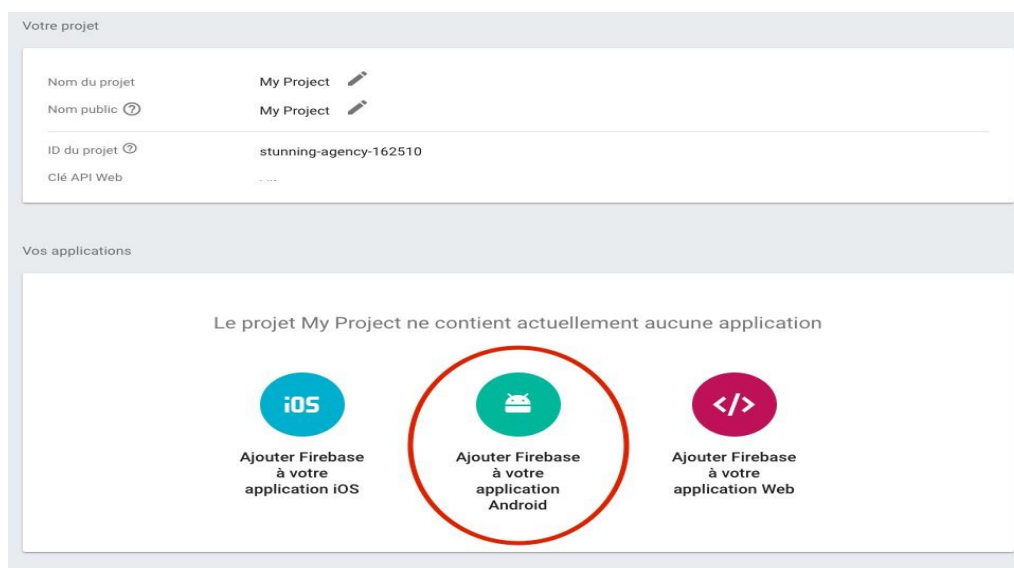
On clique en haut à droite sur « accéder à la console »



On Sélectionne le projet ou bien on ajoute un projet, pour ajouter un projet : on clique sur « ajouter un projet », on va remplir de nom du projet et on sélectionne le pays avec la liste déroulante puis on clique sur « créer un projet »



Une fois sur le projet, une nouvelle page s'affiche. On clique sur l'icône « Ajouter Firebase à notre application Android »



On va Remplir le nom de package de notre application Android, ce package est déclaré dans création d'application, puis on clique sur « Enregistrer l'application »

Ajouter Firebase à votre application Android

1 Enregistrer l'application 2 Télécharger le fichier de configuration 3 Ajouter le SDK Firebase

Nom du package Android ?

Pseudo de l'application (facultatif) ?

Certificat de signature de débogage SHA-1 (facultatif) ?

Requis pour l'assistance relative à Dynamic Links, Invites et Google Sign-In ou l'assistance par téléphone dans Auth. Modifiez les certificats SHA-1 dans les paramètres.

ANNULER ENREGISTRER L'APPLICATION

dans le projet **My Project**


Nous allons maintenant télécharger le fichier "google-service-json", et ce fichier est utilisé lors de la programmation de l'application. Grâce à ce fichier, l'application peut communiquer avec la base des données.

Ajouter Firebase à votre application Android

1 Enregistrer l'application 2 Télécharger le fichier de configuration 3 Ajouter le SDK Firebase

Instructions relatives à Android Studio Alternatives : [Unity](#) [C++](#)

1. [Télécharger google-services.json](#)
2. Accédez à la page **Projet** dans Android Studio pour consulter le répertoire racine de votre projet.
3. Déplacez le fichier **google-services.json** que vous venez de télécharger dans le répertoire racine du module de votre application Android.

 **google-services.json**

Vous avez déjà ajouté les dépendances ? [Accéder directement à la console](#)

CONTINUER

Donc, notre projet est prêt. Maintenant on va créer la base des données et les variables qu'on va les utilisées après.

Dans ce cas, le nom de projet est : systeme-aquaponie-m2

Les variables sont classés dans des 'Childs' et dans chaque 'Child' on trouve les variables qui présentent notre capteurs et actionneurs. La valeur de chaque variable est déclarée dans tous les programmes comme une chaine de caractère 'string'.

Notre base des données est prête a être utilisée.

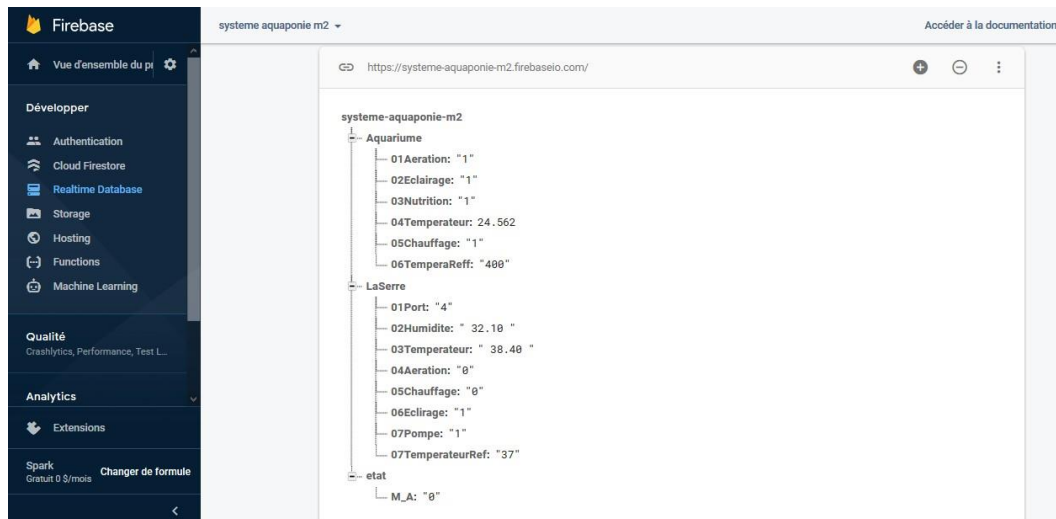


Figure III 9: l'état final de la base de données

Childs	variables
Aquariume	01Aeration
	02Eclairage
	03Nutrition
	04Temperatureur
	05Chauffage
	06TemperatureurReff
LaSerre	01Port
	02Humidite
	03Temperatureur
	04Aeration
	05Chauffage
	06Eclairage
	07Pompe
07TemperatureurRef	
Etat	M_A

Tableau III. 1 : la table des childs et variables utilisés dans notre base des données

III.3.2- Programmation d'application :

III.3.2.1- Création d'un projet :

Après l'installation et la configuration de l'environnement de développement des applications Android (Android studio), on commence à travailler sur notre application.

- Au début on crée un nouveau projet.

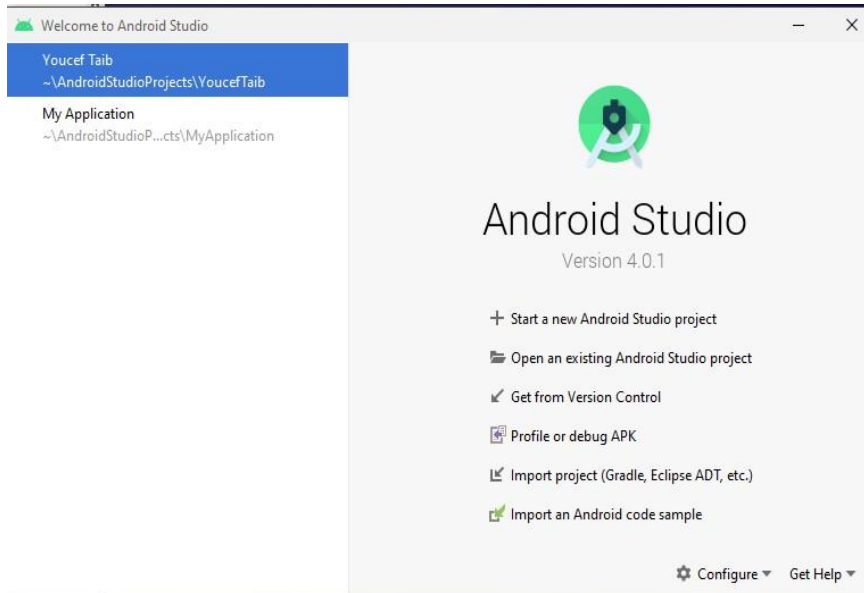
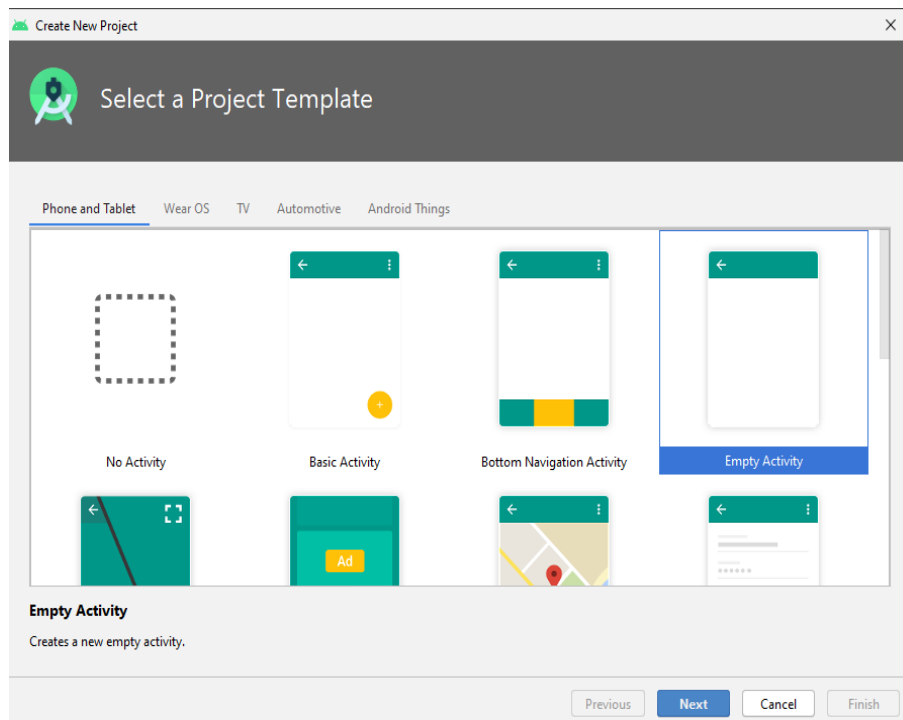
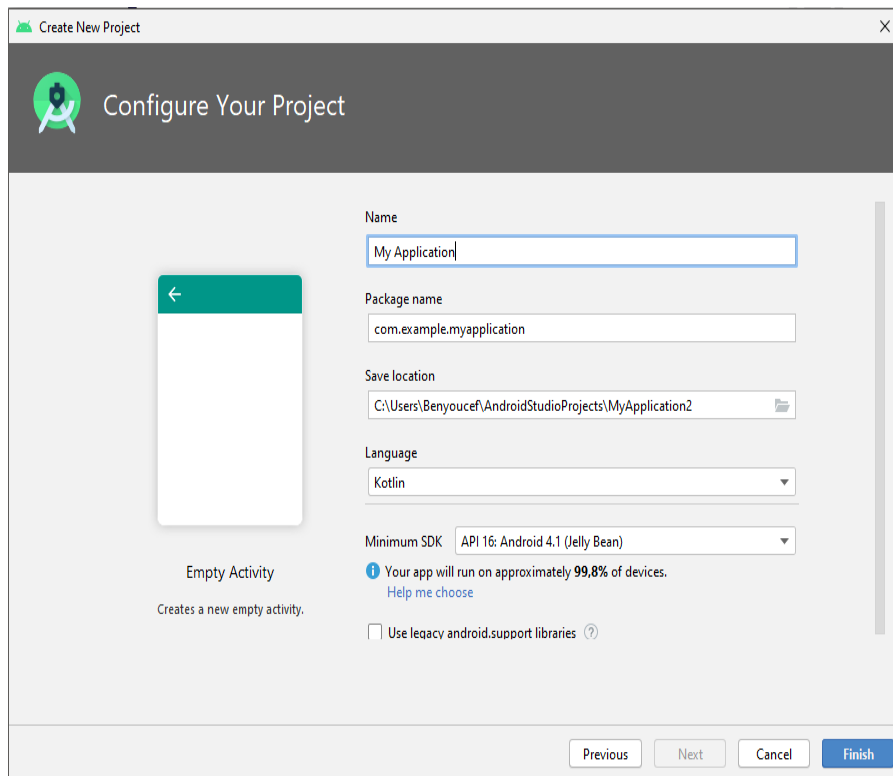


Figure III 10: interface initial de logiciel Android Studio

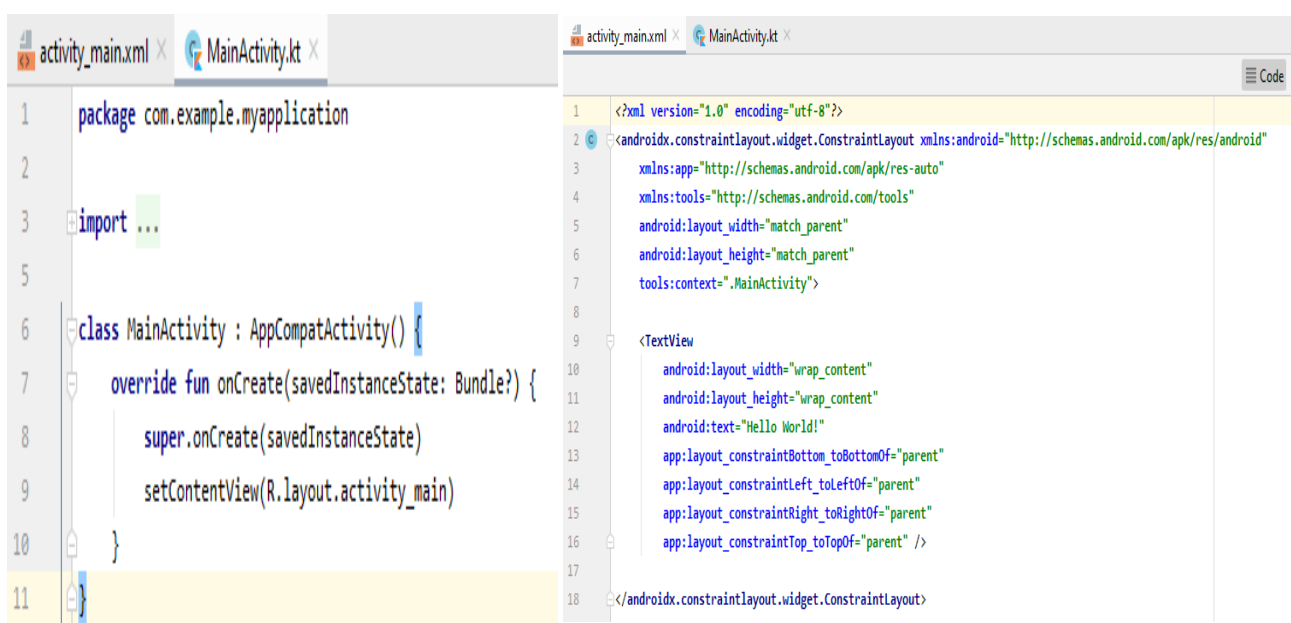
- On choisit un model 'Empty Activity'.



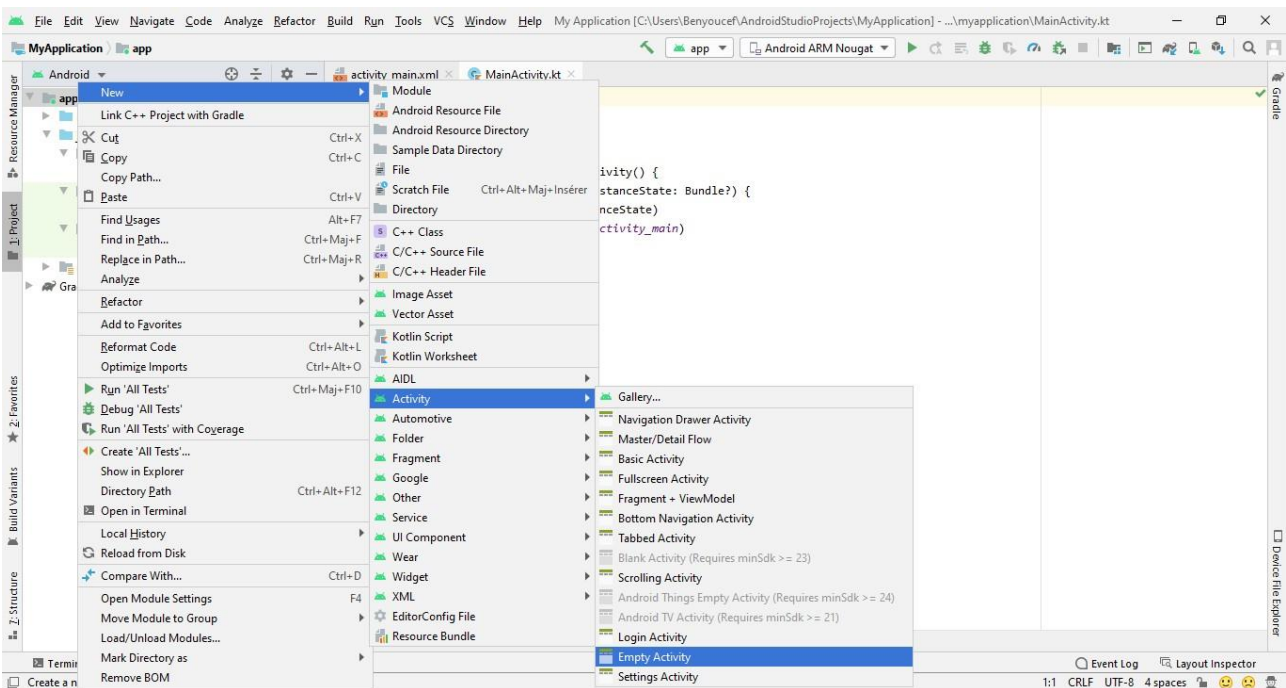
- On donne un nom pour notre projet et on choisit le langage de programmation « Kotlin » ou bien « Java ».



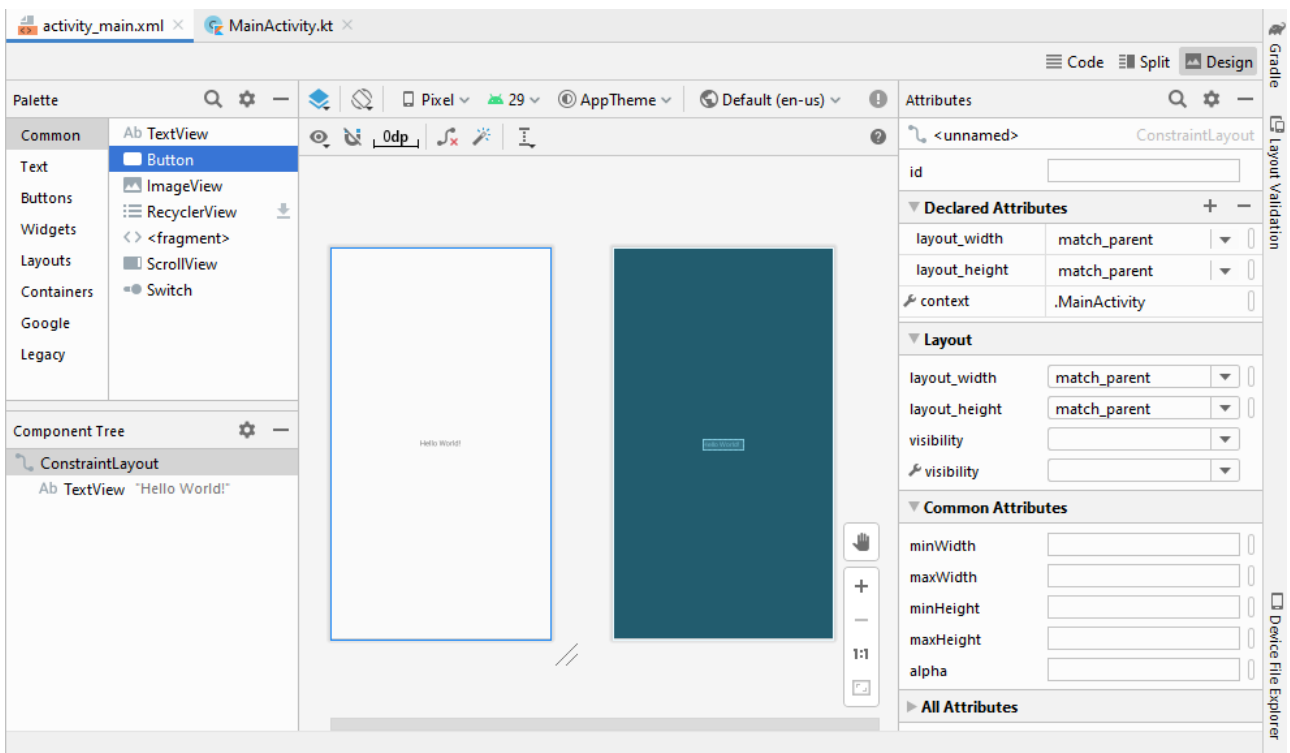
- Le logiciel nous donne deux codes liés à l'activité main de l'application, « MainActivity.kt » c'est le code de différentes tâches qui se produisent au niveau de l'activité main et « Activity_main.xml » c'est le code de dessin de cette activité.



- On ajoute deux autres Activity, une pour la serre et l'autre pour l'aquarium.



- On ajoute les composants de chaque activité, les boutons, les textes et les arrière-plans, etc.
- Chaque composant est spécifié par un 'Id', c'est comme un matricule.
- Chaque composant est programmé séparément via la fenêtre attribues, Longueur, largeur, visibilité, type de police, couleur, etc.



III.3.2.2 Explication de programme utilisé :

- **Activité main :**

Nous avons donc trois activités à programmer, main, aquarium et la serre, chaque activité a un code, code de fonctionnement et un code pour le design, on commence par le design.

Dans l'activité main, on trouve :

- Logo d'université
- Des textes
- Trois boutons
- L'arrière-plan

Les boutons serre et aquarium sont utilisés pour changer l'activité main vers les autres activités, et le bouton Auto / Manuel va changer le mode de fonctionnement de notre système.

On a utilisé textview pour inclure le titre de thème et le nom de l'université.

Et pour la programmation de chaque composant, on déclare au début tous les variables qu'on a utilisés dans un fichier qui porte le nom 'String.xml'.

Le nom 'String' est donné en fonction de la valeur des variables utilisées qui été une chaîne de caractère, par exemple une phrase, et elle doit être déclarée ente " ".

Alors le texte utilisé en haut de l'activité main, le titre de thème et même les boutons sont déclarés comme suit :

```
<string name="algerian_n">République Algérienne démocratique et  
populaire\n\nMinistère de l'enseignement supérieur et de la recherche  
scientifique\n\nÉtablissement Université de Djillali Bounaama Khmis Miliana  
</string>  
<string name="theme">Etude et réalisation d'un mini système Aquaponie a base  
d'un Raspberry Pi</string>  
<string name="la_serre">La serre</string>  
<string name="m_t_o">Auto</string>  
<string name="aquarium">Aquarium</string>
```

L'utilisation de la contre slash \ a pour but de prendre en considération l'apostrophe utilisée dans le mot « l'enseignement ».

L'utilisation de \n et pour sauter la ligne et démarrer un texte dans une nouvelle ligne.



Figure III 11: Activité Main d'application

Ce code est pour la dimension de l'activité main, dont le rôle est celui de l'activité s'adapte avec tous les modèles de téléphones, et l'activité occupe tout l'écran de ce téléphone.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/startbg"
android:padding="15dp"
tools:context=".MainActivity">
```

Ce code est pour le logo de l'université utilisé dans l'activité, le type de ce logo est image, son 'id' est 'logo left', les deux lignes qui sont déclarées 'true' sont utilisées pour garder la position de logo quel que soit la dimension d'écran utilisé. '70dp' C'est la longueur et la largeur de l'image.

```
<ImageView
android:id="@+id/logo_left"
android:background="@drawable/logo"
android:layout_alignParentTop="true"
android:layout_alignParentStart="true"
android:layout_width="70dp"
android:layout_height="70dp"
```

Ce code est destiné à formaliser et positionner le texte en haut de l'activité, le 'id' de texte est 'header_text', la dimensions de zone de texte est "234dp" de largeur et "88dp" de longueur, le texte est appelé de fichier 'string' par "@string/algerian_n" avec algerian_n c'est la variable qui porte le texte.

```
<TextView
android:id="@+id/header_text"
android:layout_width="234dp"
android:layout_height="88dp"
android:layout_alignParentTop="true"
android:layout_marginStart="3dp"
android:layout_marginTop="14dp"
android:layout_marginEnd="4dp"
android:layout_toStartOf="@id/logo_right"
android:layout_toEndOf="@id/logo_left"
android:gravity="center_horizontal"
android:text="@string/algerian_n"
android:textColor="@color/colorPrimary1"
android:textSize="8sp" />
```

La couleur et la taille de police sont déclarées dans les deux dernières lignes

Le titre de thème est programmé par la même méthode qu'algerian_n

Dans ce cas on a le bouton, le texte et sa couleur sont déclarés dans les deux premières lignes, le format de bouton est défini par "@drawable/btnbackground",

La largeur et la longueur de bouton est de '100dp', il est aussi fixé pour tous les démentions des écrans utilisés.

Ce code liée à un bouton, pour les autres boutons on change seulement l'id' et le texte de bouton.

```
<Button
android:textColor="@color/colorWhite"
android:text="@string/la_serre"
android:background="@drawable/btnbackground"
android:layout_marginBottom="40dp"
android:id="@+id/serre"
android:layout_width="100dp"
android:layout_height="100dp"
android:layout_alignParentStart="true"
android:layout_alignParentBottom="true" />
```

Le rôle de code MainActivity.kt est d'effectuer les différents ordres pour changer les activités et les valeurs dans la base des données.

Au début du programme, on déclare la variable 'etat', cette variable est responsable de changer la valeur de la même variable dans la base de données entre "0" et "1", elle va aussi changer le texte de bouton de milieu entre "Auto" et "Manuel"

```
val users: String = "etat"
var firebase =
    FirebaseDatabase.getInstance()
var userRef =
    firebase.getReference(users)
```

Dans la deuxième ligne, on a connecté à la base de données qu'ont été créé sur Firebase.

Dans la troisième ligne, on a connecte avec la variable (ou bien Child) 'etat' directement pour faire changer sa valeur dans la base des données.

Dans cette partie, on a programmé le bouton 'Auto/Manuel'.

Lorsqu'on clique sur le bouton qu'est l'Id "etat" :

Si le texte dans le bouton est 'Auto', on change la valeur de la variable "M_A" de Child "etat" en "0", et la couleur de texte de bouton en noir "#000000"

Si le texte dans le bouton est 'Manuel', on change la valeur de la variable "M_A" de Child "etat" en "1", et la couleur de texte de bouton en blanc "#FFFFFF"

```
etat.setOnClickListener {
    if(etat.text == "Auto"){
        userRef.child("M_A").setValue("0")
        etat.setText("Manuel")
        etat.setTextColor(Color.parseColor("#FFFFFF"))
    }

    else if(etat.text == "Manuel"){
        userRef.child("M_A").setValue("1")
        etat.setText("Auto")
        etat.setTextColor(Color.parseColor("#000000"))
    }
}
```

Les deux boutons Aquarium et Serre, ont le même rôle, qui est le changement de l'activité main vers aquarium activité ou serre activité.

Si on clique sur le bouton aquarium, on démarre l'activité aquarium, et si on clique sur le bouton serre, on démarre l'activité serre.

```
aquarium.setOnClickListener {
    var myintent_aquarium =
        Intent(this.baseContext,
            AquariumeActivity::class.java)
        startActivity(myintent_aquarium)}
serre.setOnClickListener {
    var myintent_serre =
        Intent(this.baseContext,
            SerreActivity::class.java)
        startActivity(myintent_serre)}
```

- **Activité la serre :**

Cette activité représente les paramètres de la serre, des boutons pour l'ouverture et la fermeture de la porte, deux textviews pour afficher la valeur de température et humidité dans la serre, on a aussi des voyants, le voyant est rouge lorsque l'actionneur est éteint, il est vert lorsque l'actionneur est en marche.

Les boutons ON/OFF sont utilisés pour allumer et stopper les actionneurs lorsque le système en mode manuel.



Figure III 12: Activité Serre d'application

Ce code permet d'afficher le nom de l'activité "La Serre", le nom est affiché en gras avec une couleur noire.

Il est localisé en haut de l'activité "match_parent" dans le centre horizontal et vertical, avec une taille de police '36sp'.

```
<TextView
    android:layout_width="match_parent"
    android:textColor="@color/black"
    android:layout_height="50dp"
    android:gravity="center_horizontal|center_vertical"
    android:text="@string/la_serre2"
    android:textSize="36sp"/>
```

Dans les premières lignes de code, <RelativeLayout c'est la partie de réservation d'un espace dans l'activité, cet espace est réservé pour écrire le texte "la porte" et même pour placer les deux boutons "close" et "open".

Les paramètres de <Textview permettent d'afficher "la porte", avec une couleur noire, et taille de police de "18sp".

Le mot "la porte" est fixé dans l'espace réservé au début quel que soit la dimension de l'écran de téléphone.

```
<RelativeLayout
    android:layout_marginTop="10dp"
    android:id="@+id/B"
    android:layout_below="@id/A"
    android:layout_width="match_parent"
    android:layout_height="50dp">

    <TextView
        android:textColor="@color/colorPrimary1"
        android:layout_marginStart="20dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_width="wrap_content"
        android:layout_height="60dp"
        android:gravity="center_vertical"
        android:text="@string/port"
        android:textSize="18sp"/>
```

Le bouton open a un 'Id' "port_status_open", il a des dimensions de 80 / 40dp. La couleur de texte est blanche.

Le bouton close a un 'Id' "port_status_close", il est la même dimension que bouton open, la seule différence c'est leur emplacement dans l'activité.

```
<Button
android:contentDescription="@string/app_name"
android:id="@+id/port_status_open"
android:text="@string/open"
android:layout_width="80dp"
android:layout_height="40dp"
android:textColor="@color/colorWhite"
android:background="@drawable/bt"
android:layout_marginTop="15dp"
android:src="@drawable/one"
android:layout_centerHorizontal="true"/>
```

Cette partie de code permet d'afficher la ligne d'humidité.

Le mot 'humidité' s'affiche avec une couleur noir 'colorPrimary1' et avec une taille de police '18sp'

```
<TextView android:textColor="@color/colorPrimary1"
android:layout_marginStart="20dp"
android:layout_alignParentTop="true"
android:layout_alignParentStart="true"
android:layout_width="wrap_content"
android:layout_height="50dp"
android:gravity="center_vertical"
android:text="@string/Humidit"
android:textSize="18sp"/>
```

De l'autre côté du mot humidité, on trouve la valeur d'humidité en pourcentage, l'id' de la valeur d'humidité est "@+id/humidite_serre_status", sa couleur est noir avec une taille de police '30sp'.

```
<TextView
android:id="@+id/humidite_serre_status"
android:layout_width="wrap_content"
android:layout_height="50dp"
android:layout_alignParentTop="true"
android:layout_alignParentEnd="true"
android:layout_marginTop="0dp"
android:layout_marginEnd="32dp"
android:gravity="center_vertical"
android:text="@string/humi"
android:textColor="@color/colorPrimary1"
android:textSize="30sp" />
```

Le mot 'humidité' et sa valeur est fixe dans sa place quelle que soit la dimension de l'écran utilisé.

Cette partie va créer une zone d'écriture, son "id" est "@ + id / temp_serre_reff", et le rôle de cette région est d'écrire et d'envoyer une température de référence, qui est la température qui doit être à l'intérieur de la serre tout le temps. La transmission se fait via le bouton envoyer.

```
<EditText
android:id="@+id/temp_serre_reff"
android:layout_width="wrap_content"
android:layout_height="60dp"
android:layout_centerHorizontal="true"
android:layout_marginEnd="60dp"

android:gravity="center_vertical|center_horizontal"
android:hint="@string/tempera_reff" />
```

Après avoir expliqué le code Activity_Serre.xml, nous nous tournons maintenant pour expliquer le code SerreActivity.kt. Et puisque on a deux modes de

fonctionnements Auto/Manuel, on a besoin de deux méthodes de fonctionnement des boutons et voyants.

Dans le cas auto, on va recevoir les données seulement, mais dans le cas manuel on va utiliser tous les boutons pour allumer et éteindre les actionneurs.

Comme un premier pas, on déclare le 'Child' "LaSerre" de la base des données, après on connecte notre base de données, et même dans le 'Child' "LaSerre".

```
val users: String = "LaSerre"
var
firebase=FirebaseDatabase.getInstance()
var userRef=firebase.getReference(users)
```

Dans ce cas, si on clique sur le bouton open de la porte, la valeur de 'Child' "01Port" dans la base des données ça va être un "1", et si on clique sur le bouton close, la valeur sera un "0".

```
port_status_open.setOnClickListener {
    userRef.child("01Port").setValue("1")
}

port_status_close.setOnClickListener {
    userRef.child("01Port").setValue("0")
}
```

Les lignes de ce code permettent de télécharger la valeur de température et d'humidité de Child "03Temperatureur" et "02Humidite" vers l'application, et de les afficher sur L'activité d'application 'LaSerre'.

```
temp_serre_status.setText(dataSnapshot.child("03Temperatureur").getValue().toString()
+ "C")
humidite_serre_status.setText(dataSnapshot.child("02Humidite").getValue().toString()
+ "%")
```

Ce code est destiné à commander le bouton et le voyant de la variable Aération de l'activité Serre dans le mode de fonctionnement Manuel, cette variable est déclarée "04Aeration".

Lorsque on clique sur le bouton On/Off de la variable aération :

Si le bouton est ON, on change l'écriture vers OFF, avec une couleur de texte noir, la valeur de la variable devienne "1" et la couleur de voyant devient rouge.

Si le bouton est OFF, l'écriture devienne ON blanche, la valeur de variable "0", et le voyant sera vert.

```
btn_serre_air.setOnClickListener {

    if(btn_serre_air.text == "ON"){

userRef1.child("04Aeration").setValue("0")
serre_air_status.setImageResource(R.drawable.zero)
btn_serre_air.setText("OFF")
btn_serre_air.setTextColor(Color.parseColor("#000000"))    }

else if(btn_serre_air.text == "OFF"){

userRef1.child("04Aeration").setValue("1")
serre_air_status.setImageResource(R.drawable.one)
btn_serre_air.setText("ON")
btn_serre_air.setTextColor(Color.parseColor("#FFFFFF"))    }
```

Ce code relié à la même variable "04Aeration" mais cette fois il fonctionne dans le mode Auto.

Si la valeur de la variable "04Aeration" égale à "0", le texte de bouton sera OFF avec une couleur noir, la couleur de voyant devient rouge.

Si la valeur de la variable est égale a "1", le texte devient ON blanc et le voyant sera vert.

Tous les boutons et les voyants sont programmés de la même manière, la seule différence entre eux c'est la variable utilisée.

- **Activité Aquariume :**

La troisième activité de notre application, c'est l'Aquariume. Cette activité porte tous les paramètres d'un aquarium, qu'on va les utilise pour un élevage de poissons.

Cette activité est reliée avec la basse de données a travers le 'Child' "Aquariume", a l'intérieur de ce 'Child' on trouve tous les variables qui sont reliées avec les paramètres affichées dans l'activité.

La méthode de programmation de cette activité est la même que la précédente, soit pour son design soit pour leur composants destinés aux différentes taches, mais la seule différence c'est les 'Id' utilisés pour définir les différent composants.

```
override fun onDataChange(dataSnapshot:
DataSnapshot)
{if(dataSnapshot.child("04Aeration").getVa
lue().toString() == "0"){
serre_air_status.setImageResource(R.drawab
le.zero)
    btn_serre_air.setText("OFF")
    btn_serre_air.setTextColor(Color.parseColo
r("#000000")) }
else if
(dataSnapshot.child("04Aeration").getValu
e().toString() == "1"){
serre_air_status.setImageResource(R.drawab
le.one)
    btn_serre_air.setText("ON")
    btn_serre_air.setTextColor(Color.parseColo
r("#FFFFFF")) }
```

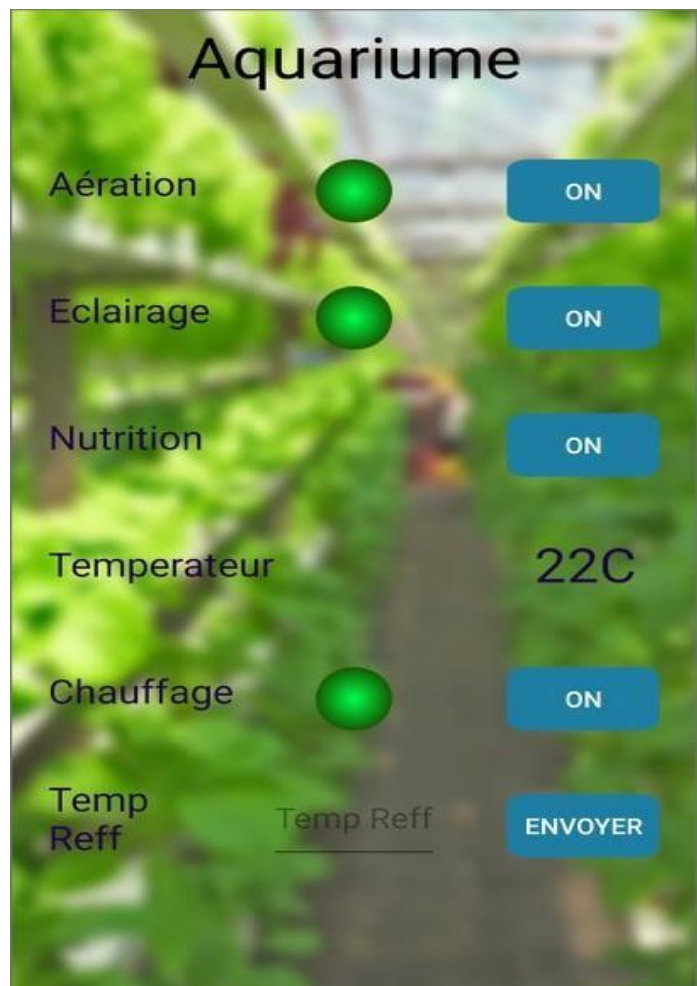


Figure III 13: Activité Aquariume d'application

III.2.4 Préparation et programmation de Raspberry pi :

III.2.4.1- Installation de Raspbian :

Nous aurons besoin dans cette étape de carte SD vierge.

- **Etape 01 : Télécharger la dernière version de Raspbian**

Comme un premier pas on télécharge l'image disque de Raspbian sur le site de la fondation Raspberry Pi : <https://www.raspberrypi.org/downloads/raspbian/> [36]

- **Etape 02 : Décompressez l'archive**

Nous décompresserons l'archive que nous avons téléchargée, nous aurons besoin de l'un des programmes suivants:

- Windows : [7-Zip](#)
- Mac : [The Unarchiver](#)
- Linux : [Unzip](#)

- **Etape 03 : Téléchargez Etcher**

Etcher est un logiciel qui va permettre d'installer Raspbian (ou n'importe quel système d'exploitation) sur la carte SD et de le rendre directement bootable.

Il existe d'autres logiciels mais Etcher a l'avantage d'être compatible avec Mac, Windows, Linux et est recommandé par la fondation Raspberry Pi.

- **Etape 04 : on insère la carte SD dans notre Pc, après, on va lancer Etcher.**

- **Etape 05 : Flashe de la carte SD avec Raspbian**

On va Choisir l'image disque de Raspbian dans Etcher, et on sélectionne notre carte SD et on lance l'écriture en cliquant sur « Flash ». [36]



Figure III 14: Interface de logiciel Etcher

Nous attendrons la fin du flash.



Figure III 15: Interface de logiciel Etcher au cours de flash

Nous pouvons alors insérer la carte SD sur notre Raspberry pi, on branche l'alimentation au Raspberry Pi pour le démarrer, et il va booter sur Raspbian. Il nous reste à configurer Raspbian. [36]

III.2.4.2 Configuration de Raspberry pi :

Une fois que Raspbian est installé sur notre Raspberry Pi, il nous reste quelques paramètres à configurer avant de pouvoir l'utiliser pleinement, comme par exemple la langue de l'OS ou la langue du clavier. [37]

Au premier démarrage de Raspbian, nous devrions voir apparaître l'écran appelé « raspi-config ». Si ce n'est pas le cas, nous pouvons y accéder, on allant dans le terminal et on tapant :

```
sudo raspi-config
```

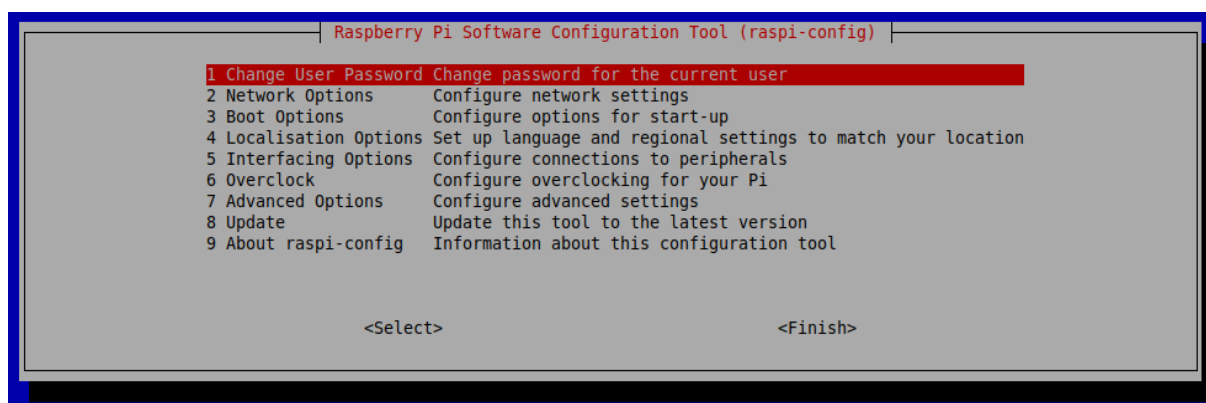


Figure III 16: Interface de configuration Raspberry pi

Voyons tout de suite ce qu'il est possible de configurer via le raspi-config :

1. Changer le mot de passe de l'utilisateur
2. Modifier les paramètres réseaux
3. Configurer les options au démarrage
4. Changer les options de localisation
5. Configurer les connexions des périphériques
6. Configurer l'overclocking du Raspberry Pi
7. Configurations avancées
8. Faire la mise à jour de Raspbian ou de certains paquets

Pour naviguer dans le menu de raspi-config, nous devons utiliser les flèches directionnelles de notre clavier « haut » et « bas ». [37]

L'utilisateur par défaut après l'installation de Raspbian est « pi » et le mot de passe est « raspberry ». Il est fortement conseillé de la changer pour plus de sécurité.

Pour mettre à jour Raspbian, nous allons devoir exécuter 3 commandes dans le terminal :

```
sudo apt-get -y update
```

« sudo » permet d'exécuter la commande comme administrateur, nécessaire car la mise à jour va modifier des fichiers du système.

« apt-get » est l'outil de gestion des paquets.

« update » est une option de apt-get permettant de mettre à jour tous les paquets installés.

```
sudo apt-get -y dist-upgrade
```

```
sudo reboot
```

« dist-upgrade » permet de migrer nos paquets vers la dernière version disponible.

« reboot » redémarre simplement le système.

Dans notre configuration, on a besoin aussi d'activer tous les interfaces de raspberry pi, sont les interfaces utilisées pour communiquer avec l'environnement extérieur de raspberry pi. [37]

Dans le menu démarrer de raspberry pi, on a choisi : Préférences > Raspberry Pi Configuration > Interfaces

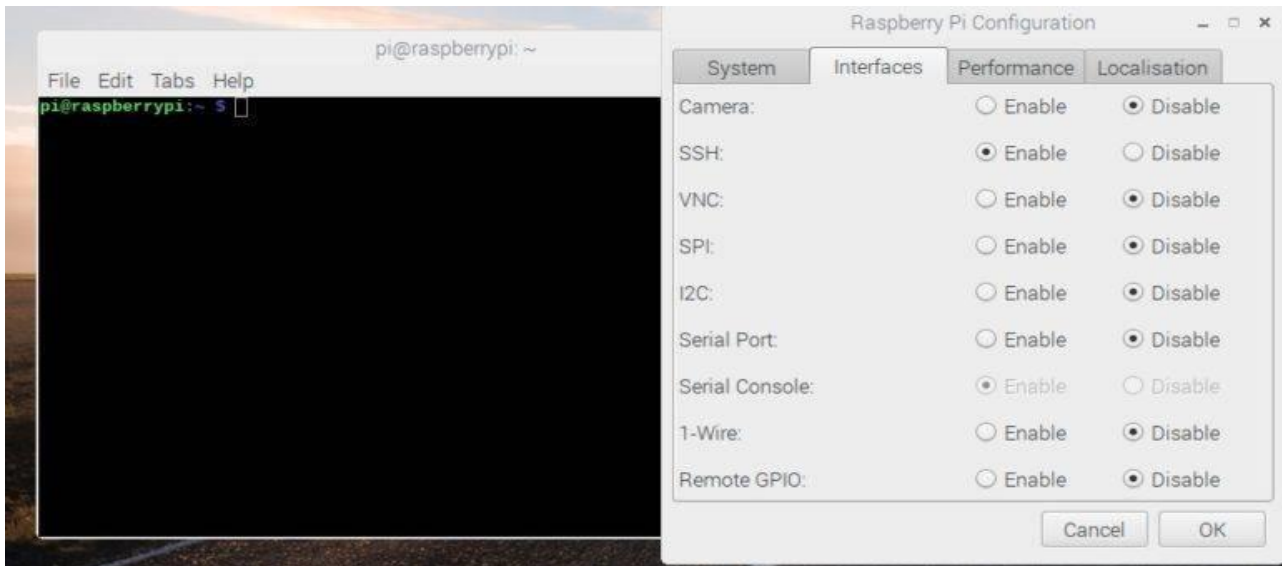


Figure III 17: Interface de configuration avancée

Dans notre cas, on va activer SSH et 1-Wire :

SSH : c'est un logiciel et un protocole de communication informatiques, on va l'utiliser pour contrôler le raspberry pi sans clavier, souris et même écran. Sur Windows, nous allons avoir besoin d'un logiciel pour nous connecter en SSH. Le meilleur logiciel est PuTTY. Une fois PuTTY téléchargé et installé, nous pouvons entrer l'adresse IP de raspberry pi dans le champ « Host Name (or IP address) ». Le champ « Connection Type » doit être « SSH ».

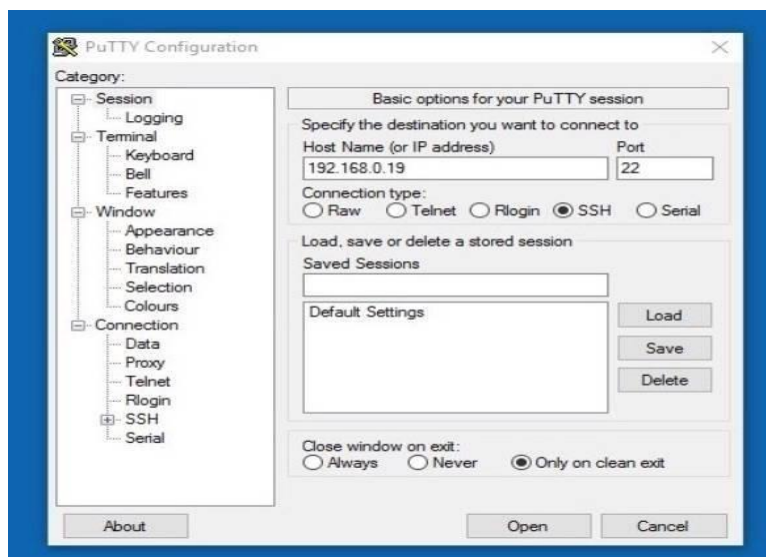


Figure III 18: interface de logiciel PuTTY

Nous pouvons alors cliquer sur « Open » et une fenêtre de sécurité « Security Alert » apparaît, on va cliquer simplement sur « Yes ».

Une fenêtre de ligne de commande devrait alors apparaître nous demandant « login as : ». On va entrer « pi » puis on appuie sur la touche entrée. On va Entrer ensuite notre mot de passe et on va valider avec la touche entrée de notre clavier.

1-Wire : c'est le Protocol utilise pour connecter le capteur ds18b20 avec le raspberry pi.

Après avoir configuré les interfaces, nous allons maintenant en train d'installer Les différentes bibliothèques et paquets, les bibliothèques et les paquets sont utilisés pour définir les capteurs, les protocoles et même les interfaces qu'on a utilisés dans la programmation python :

DHT22 : Pour récupérer les informations de la sonde, une bibliothèque a été développée par Adafruit, alors on va utiliser la bibliothèque Python.

Par la fenêtre de terminal, commençons par installer les fichiers python requis.

```
sudo apt-get install build-essential python-dev git
```

Ensuite, les bibliothèques de capteurs :

```
git clone https://github.com/adafruit/Adafruit_Python_DHT.git
cd Adafruit_Python_DHT/
sudo python setup.py install
```

Les fichiers sont prêts à être utilisés maintenant, par le capteur DHT22.

Le capteur Ds18b20 a besoin aussi d'une bibliothèque, cette bibliothèque va définir ce capteur dans nos programmes.

Puisque nous utilisons Raspbian, nous pouvons l'installer à partir du référentiel officiel à travers la ligne suivante :

```
sudo apt-get install python-w1thermsensor
```

Tout l'assemblage de notre système est prêt, il nous reste que de programmer le microcontrôleur Raspberry pour qu'il puisse reconnaître et fonctionner avec tous nos capteurs et actionneurs. Et enfin notre système sera opérationnel et prêt pour le fonctionnement.

III.2.4.3- Programmation de raspberry pi en python :

Après avoir fait les préparations préliminaires de liaison entre capteurs et actionneurs, et la configuration de raspberry pi, reste seulement la programmation qui va contrôler nos capteurs et actionneurs, dans cette partie on va utiliser le langage python et détailler tous ces programmes.

Nous allons maintenant commencer par le cas automatique de notre système, dans ce cas, nous utilisons notre application pour la supervision seulement.

- **La porte :**

→ Dans les premières lignes, nous appelons les bibliothèques, bibliothèque de temps, bibliothèque de contrôle électronique (GPIO) et la bibliothèque de la base des données Firebase.

→ Dans cette partie, nous déclarons les pins de module L298 (in3, in4, enb) qui sont responsables de contrôler le moteur à courant continu, nous définissons aussi la numérotation au type (BOARD), le type qui permet aux ports d'être organisés en fonction de leur emplacement sur Raspberry Pi.

→ Cette ligne détermine que nous utilisons les pins 35, 33 et 37 comme des sorties (OUT), les sorties 35 et 33 sont fixes dans l'état 0 (LOW), nous fixons la sortie 37 en mode PWM avec une fréquence de 1KHz, le cycle de service au début est fixe dans 0%.

→ La dernière ligne permet de se connecter à Firebase, exactement dans la base des données de notre projet qui est liée au lien "https://systeme-aquaponie-m2.firebaseio.com/ "

→ Il répète les commandes à l'intérieur sans fin ou jusqu'à ce que le programme soit fermé.

→ Accéder à la base de données et télécharger la valeur de la variable "01Port" dans " LaSerre" et la donner à la variable « result ».

→ Si la variable « result=="1" », on fixe la sortie 35 à l'état 0 (LOW) et on change l'état de la sortie 33 vers 1 (HIGH).

```
import time
import RPi.GPIO as GPIO
from firebase import firebase

in3 =35
in4 =33
enb =37

GPIO.setmode(GPIO.BOARD)

GPIO.setup(in3,GPIO.OUT)
GPIO.setup(in4,GPIO.OUT)
GPIO.setup(enb,GPIO.OUT)
GPIO.output(in3,GPIO.LOW)
GPIO.output(in4,GPIO.LOW)
p=GPIO.PWM(enb,1000)
p.start(0)

firebase = firebase.FirebaseApplication('https://systeme-aquaponie-m2.firebaseio.com/', None)
```

```
while True:
    result = firebase.get('/LaSerre/01Port/', "")
    if (result=="1"):
        GPIO.output(in3,GPIO.LOW)
        GPIO.output(in4,GPIO.HIGH)
        p.ChangeDutyCycle(75)
        time.sleep(1)
        firebase.put("/LaSerre", "/01Port", "4")
```

Le cycle de service sera 75%. Après 4 secondes la valeur de variable "01Port" devient "4".

Et si la variable « result=="0" », on change l'état de la sortie 35 vers 1 (HIGH) et on change l'état de la sortie 33 vers 0 (LOW). On garde le cycle de service à 75%, Après 4 secondes la valeur de variable "01Port" devient "4".

Mais si la valeur de la variable « result » est différente de 0 et 1 on fixe les deux sorties 35 et 33 dans un même état 0 (LOW).

Ce programme permet d'ouvrir et fermer la porte en fonction de la valeur de variable « result ». Le cycle de service représente la vitesse d'ouverture et fermeture de porte.

Time.sleep(1) assure le temps nécessaire d'ouverture et fermeture de porte.

- **Le climat – Résistance chauffante et ventilateur – :**

→ Comme le premier programme, on appelle les bibliothèques, mais cette fois ci, on appelle aussi la bibliothèque de capteur DHT22 c'est la bibliothèque Adafruit_DHT

→ Dans cette partie, nous déclarons les pins de module L298 (in1, in2, enA) qui sont responsables à contrôler le ventilateur, nous déclarons aussi le pin 12, c'est le pin responsable à contrôler le relais de résistance chauffante. On déclare le model de capteur "sensor" qui est Adafruit_DHT.DHT22, Nous définissons aussi la numérotation au type (BOARD), le type qui permet aux ports d'être organisés en fonction de leur emplacement sur Raspberry Pi.

On déclare tous les pins comme des sorties, et on fixe les pins in1 et in2 a l'état 0 (LOW).

Nous fixons la sortie 22 en mode PWM avec une fréquence de 1KHz, le cycle de service au début est fixé à 0%.

```
elif (result=="0"):
    GPIO.output(in3,GPIO.HIGH)
    GPIO.output(in4,GPIO.LOW)
    p.ChangeDutyCycle(75)
    time.sleep(1)
    firebase.put("/LaSerre", "/01Port", "4")
else:
    GPIO.output(in3,GPIO.LOW)
    GPIO.output(in4,GPIO.LOW)
    time.sleep(1)
```

```
import time
import RPi.GPIO as GPIO
from time import sleep
from firebase import firebase
import Adafruit_DHT

pin = 12
in1 = 18
in2 = 16
enA = 22
sensor = Adafruit_DHT.DHT22

GPIO.setmode(GPIO.BOARD)
GPIO.setup(12, GPIO.OUT)
GPIO.setup(in1,GPIO.OUT)
GPIO.setup(in2,GPIO.OUT)
GPIO.setup(en,GPIO.OUT)

GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)

p=GPIO.PWM(enA,1000)
p.start(0)
```

→ On a connecté dans notre base des données qu'ont été créé, après on fixe les valeurs des deux variables « 03Temperature » et « 02Humidite » dans la valeur "0.00"

→ Pour assure des commandes répétitives et sans fin.

→ Au début, nous lisons la température et l'humidité du la serre à partir de capteur, on donne les valeurs obtenues pour les variables "temp" et "hum", Nous téléchargeons ces valeurs dans la base de données et les lions à les variables « 03Temperature » et « 02Humidite ».

Nous téléchargeons la valeur de la température de référence de la base des données « 07TemperatureRef », et on la relie avec la variable "tempRef".

Si le capteur ne peut pas mesurer la température, nous obtenons un message d'erreur.

Après que le capteur mesure la température, on va la comparer avec la température de référence, si la température mesuré est inférieure à la température de référence, la sortie 12 sera fixé dans l'état 0, et la valeur de la variable « 05Chauffage » sera "1", mais si la température mesurée est supérieure à la température de référence la sortie 12 sera fixé dans l'état 1, et la valeur de la variable « 05Chauffage » dans la base des données sera "0",

Cette partie du programme permet de contrôler la résistance chauffante.

→ Si la température mesurée supérieur à la température de référence par une degré de Celsius, la sortie in1 sera fixé dans l'état "1" et la sortie in2 dans l'état "0", et la valeur de la variable « 04Aeration » dans la base des données sera "1". Le cycle de service sera 25%, c'est comme une première vitesse de rotation du ventilateur.

```
firebase = firebase.FirebaseApplication("https://systeme-  
aquaponie-m2.firebaseio.com/", None)  
firebase.put("/LaSerre", "/03Temperature", "0.00")  
firebase.put("/LaSerre", "/02Humidite", "0.00")  
while True:  
  
    humidity, temperature =  
    Adafruit_DHT.read_retry(sensor, pin)  
    if humidity is not None and temperature is not  
    None:  
        temp= '{0:0.2f}'.format(temperature)  
        hum= '{0:0.2f}'.format(humidity)  
  
        firebase.put("/LaSerre", "/03Temperature", temp)  
        firebase.put("/LaSerre", "/02Humidite", hum)  
        tempRef =  
        firebase.get('/LaSerre/07TemperatureRef', "")  
        etat= firebase.get('/etat/M_A', "")  
    else:  
        print("Failed to retrieve data from humidity  
sensor")  
    if (int(float(temp)) < int(float(tempRef))):  
        GPIO.output(12,0)  
        firebase.put("/LaSerre", "/05Chauffage", "1")  
    else:  
        GPIO.output(12,1)  
        firebase.put("/LaSerre", "/05Chauffage", "0")
```

```
if (int(float(temp)) == int(float(tempRef)+1)):  
    GPIO.output(in2,GPIO.LOW)  
    GPIO.output(in1,GPIO.HIGH)  
    p.ChangeDutyCycle(25)  
    firebase.put("/LaSerre", "/04Aeration", "1")
```

Si la température mesurée est supérieure à la température de référence par deux degrés Celsius, les deux sorties gardent les mêmes états, et la valeur de la variable «04Aeration» dans la base des données reste "1". Mais le cycle de service sera 50%, c'est la deuxième vitesse de rotation du ventilateur.

Si la température augmente, le cycle de service sera modifié, jusqu'à ce qu'il atteigne 100% si la température mesurée dépasse la température de référence de quatre degrés ou plus.

Mais si la température mesurée est inférieur à la température de référence, les deux sorties sont fixés à l'état 0 (LOW), et la valeur de la variable «04Aeration» dans la base des données sera "0".

Ce programme permet de contrôler le ventilateur et la résistance chauffante, on peut dire la température à l'intérieur de la serre, la résistance chauffante permet d'élever la température, le ventilateur permet de diminuer la température par quatre vitesses puisque il fonctionne par la technique de MLI (PWM).

Ce programme permet aussi de changer les valeurs des variables, « 03Temperature », « 02Humidite », «05Chauffage » et «04Aeration» dans la base des données.

- **La température de bac d'élevage des poissons :**

→ Comme précédemment, on appelle dans le début de chaque programme les différentes bibliothèques, dans ce programme on appelle la bibliothèque relative au capteur ds18b20 (W1ThermSensor).

```
elif (int(float(temp)) ==
int(float(tempRef)+2)) :
GPIO.output(in2,GPIO.LOW)
GPIO.output(in1,GPIO.HIGH)
p.ChangeDutyCycle(50)
firebase.put("/LaSerre", "/04Aeration",
"1")
elif (int(float(temp)) ==
int(float(tempRef)+3)) :
GPIO.output(in2,GPIO.LOW)
GPIO.output(in1,GPIO.HIGH)
p.ChangeDutyCycle(75)
firebase.put("/LaSerre", "/04Aeration",
"1")
elif (int(float(temp)) >=
int(float(tempRef)+4)) :
GPIO.output(in2,GPIO.LOW)
GPIO.output(in1,GPIO.HIGH)
p.ChangeDutyCycle(100)
firebase.put("/LaSerre", "/04Aeration",
"1")
else :
GPIO.output(in1,GPIO.LOW)
GPIO.output(in2,GPIO.LOW)
firebase.put("/LaSerre", "/04Aeration",
"0")
```

```
import time
import RPi.GPIO as GPIO
from time import
from firebase import firebase
from w1thermsensor import W1ThermSensor
```


→ Nous déclarons le capteur utilisé dans la première ligne, dans ce cas on a besoin de deux pins 26 et 40, sont organisées au type de numérotation BOARD, et sont fixées à l'état 1 (HIGH).

→ L'accès à la base de données est également nécessaire cette fois, car nous modifierons les valeurs de certaines des variables.

→ Sur des itérations infinies, nous appelons le capteur pour lire la température et l'associons à la variable 'temp'. Nous portons la valeur de température mesurée sur la base de données, à l'intérieur de la variable "04Temperature".

Nous téléchargeons la valeur de température de référence à partir de la base de données stockée dans la variable "06TemperatureRef".

Si la température mesurée est inférieure à la température de référence la sortie 40 reste à l'état 0 et la sortie 26 sera à l'état 1, la variable "05Heating" portera la valeur "1".

Mais si la température mesurée est supérieure à la température de référence, la sortie 40 sera 1 et la sortie 26 sera à l'état 0, la variable "05Heating" portera la valeur "0".

Ce programme permet de mesurer la température de bac d'élevage des poissons, il va la comparer avec une température de référence, il doit prendre la décision, soit d'alimenter une résistance chauffante, soit d'alimenter la pompe pour refroidir l'eau.

• Eclairage – Aération – Nutrition :

→ Au début on va appeler les bibliothèques qu'on va utiliser dans ce programme, on a également besoin d'une autre bibliothèque de temps « datetime », Cette bibliothèque nous permet d'obtenir la date et l'heure exactes du Raspberry Pi.

→ La numérotation des pins de raspberry pi est déclarée comme BOARD, on a utilisé le pin 5 comme entrée (IN), et les pins 36, 24 et 38 comme des sorties (OUT). Puisque nous

```
import time
import RPi.GPIO as GPIO
from time import sleep
from firebase import firebase
from w1thermsensor import W1ThermSensor
sensor = W1ThermSensor()
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40, GPIO.OUT)
GPIO.setup(26, GPIO.OUT)
GPIO.output(40,GPIO.HIGH)
GPIO.output(26,GPIO.HIGH)
firebase = firebase.FirebaseApplication('https://systeme-aquaponie-m2.firebaseio.com/', None)
while True:
temp = sensor.get_temperature()
firebase.put("/Aquariume", "/04Temperature", temp)
tempRef = firebase.get('/Aquariume/06TemperatureRef', "")
if (int(float(temp)) < int(float(tempRef))):
GPIO.output(40,0)
GPIO.output(26,1)
firebase.put("/Aquariume", "/05Heating", "1")
else:
GPIO.output(40,1)
firebase.put("/Aquariume", "/05Heating", "0")
GPIO.output(26,0)
```

```
import RPi.GPIO as GPIO
import datetime
import time
from firebase import firebase
GPIO.setmode(GPIO.BOARD)
GPIO.setup(5, GPIO.IN)
GPIO.setup(36, GPIO.OUT)
```

allons modifier certaines des valeurs de notre base de données sur Firebase, il convient de la contacter.

→ À l'intérieur de la boucle infinie "while true", Nous déterminons l'heure instantanée dans le système Raspberry Pi et l'attribuons à la variable 'date', On extrait de la variable 'date' l'heure et On attribue sa valeur à la variable 'h'.

Si l'état d'entrée 5 est 0 (LOW), et la valeur de la variable 'h' qui représente l'heure confiné entre 5 et 19, l'état des sorties 26 et 24 sera 1 (HIGH), et nous envoyons aux variables de la base de données « 06Eclairage, 02Eclairage, 01Aeration » la valeur "1", mais s'il y a pas les conditions précédentes, l'état des sorties deviennent 0 (LOW), Et nous envoyons aux variables la valeur "1".

Si la valeur de la variable 'h' égal à 12, l'état de la sortie 38 sera 1 (HIGH), et nous envoyons a la variable de la base de données « 03Nutrition» la valeur "1", après 4 secondes, La valeur de sortie 38 revient à 0 (LOW), et nous envoyons un "0" à la variable « 03Nutrition».

Ce programme détermine la présence et l'absence de lumière, en cas d'absence il allume la lumière artificielle entre 5 h et 19 h, et l'éteint en cas de lumière, De plus, il allume le souffleur, le distributeur de nourriture tous les jours à 12 h pendant 4 s pour distribuer une quantité de nourriture aux poissons.

- **Circulation d'eau :**

→ Comme les programmes précédents, on déclare au début les bibliothèques qu'on va utiliser.

→ La numérotation des pins de raspberry pi est déclarée comme BOARD, on a utilisé le pin 36 comme sortie (OUT), et on fixe leur état à 1 (HIGH), puis on se connecte sur notre base de données.

```
GPIO.setup(24, GPIO.OUT)
GPIO.setup(38, GPIO.OUT)

firebase = firebase.FirebaseApplication('https://systeme-aquaponie-m2.firebaseio.com/', None)

while True:
    date = datetime.datetime.now()
    h=date.hour
    if ((5<=h<19) and GPIO.input(5)) == 0:
        GPIO.output(36,1)
        GPIO.output(24,1)
        firebase.put("/Aquariume", "/02Eclairage", "1")
        firebase.put("/LaSerre", "/06Eclairage", "1")
        firebase.put("/Aquariume", "/01Aeration", "1")
    else:
        GPIO.output(36,0)
        GPIO.output(24,0)
        firebase.put("/Aquariume", "/02Eclairage", "0")
        firebase.put("/LaSerre", "/06Eclairage", "0")
        firebase.put("/Aquariume", "/01Aeration", "0")
    if (h==12):
        GPIO.output(38,1)
        firebase.put("/Aquariume", "/03Nutrition", "1")
        time.sleep(4)
        GPIO.output(38,0)
        firebase.put("/Aquariume", "/03Nutrition", "0")
```

```
import RPi.GPIO as GPIO
import time
from firebase import firebase

GPIO.setmode(GPIO.BOARD)
GPIO.setup(36, GPIO.OUT)
GPIO.output(36,1)

firebase = firebase.FirebaseApplication('https://systeme-aquaponie-m2.firebaseio.com/', None)
```

La variable 't' représente un nombre d'heures, donc pour convertir les heures en secondes, nous les multiplions par le nombre 3600, pour les utiliser dans la fonction « time.sleep », t1 représente un nombre de minutes. Et pour l'obtenir par seconde, on la multiplie par 60.

→ Initialement, la sortie 36 reste à l'état 1 pendant 'f' s, ce qui représente 5h, dans cette période, nous envoyons un 0 à la variable « 07Pompe » dans la base de données.

→ Après 5 heures, l'état de la sortie 36 passe de 1 à 0, et un 1 est envoyé à la variable « 07Pompe »

```
t=5
f=t*3600
t1=5
f1=t1*60
while True:
    GPIO.output(36,1)
    firebase.put("/LaSerre", "/07Pompe", "0")
    time.sleep(f)
    GPIO.output(36,0)
    firebase.put("/LaSerre", "/07Pompe", "1")
    time.sleep(f1)
```

Ce programme fait fonctionner la pompe à eau comme suit:

- 5 heures de repos.
- 5 heures de course.

À l'état manuel, nous contrôlons notre système via notre application mobile, que ce soit pour démarrer ou arrêter tous les actionneurs, de plus, le fonctionnement des capteurs reste le même que dans le cas automatique. Concernant les programmes, on ne changera rien au début, le changement se fera dans la partie qui suit la boucle "while".

Le programme qui contrôle la porte est le même, ce qui signifie que la porte est contrôlée par l'application dans les deux cas.

• Le climat – Résistance chauffante et ventilateur – :

→ Dans cette partie, le capteur est appelé pour lire la température et l'humidité, Nous donnons les valeurs obtenues aux variables 'temp' et 'hum'.

Nous envoyons les valeurs mesurées par le capteur à la base de données et les mettons dans les variables « 03Temperatureur » et « 02Humidite » respectivement, en même temps, nous téléchargeons les valeurs des variables « 04Aeration » et « 05Chauffage » de la base de données et nous les affectons aux variables 'Air' et 'Chau' respectivement.

Si le capteur ne lit pas les grandeurs, un message d'erreur est envoyé pour trouver le problème.

```
humidity, temperature =
Adafruit_DHT.read_retry(sensor, pin)
if humidity is not None and temperature is not None:
    temp= '{0:0.2f}'.format(temperature)
    hum= '{0:0.2f}'.format(humidity)
    firebase.put("/LaSerre", "/03Temperatureur", temp)
    firebase.put("/LaSerre", "/02Humidite", hum)
    Air = firebase.get("/LaSerre/04Aeration", "")
    Chau = firebase.get("/LaSerre/05Chauffage", "")
else:
    print("Failed to retrieve data from humidity sensor")
```

→ Si la valeur de la variable 'Air' est égale à 1, la sortie in1 sera fixée dans l'état "1" (HIGH) et la sortie in2 dans l'état "0" (LOW), Le cycle de service 100%

Mais si la valeur de variable 'Air' est différente de 1, on fixe les deux sorties in1 et in2 dans le même état 0 (LOW).

→ Si la valeur de la variable 'Air' est égale à 1, la sortie 12 sera fixée dans l'état "0" (Low), mais si elle est différente de 1 la sortie sera dans l'état 1 (HIGH).

Ce programme allume et éteint le ventilateur et la résistance chauffante à travers l'application, et permet également d'afficher la valeur de la température et de l'humidité sur l'application.

- **La température de bac d'élevage des poissons :**

→ La température est lue par le capteur puis envoyée à la base de données pour être attribuée à la variable « 04Temperature », au même temps, la valeur de la variable « 05Chauffage » est téléchargée depuis la base de données, pour être attribuée à la variable 'ch'. Si 'ch' est égal à 1, la sortie 21 sera à l'état 0 (LOW), sinon sera à l'état 1 (HIGH).

```
if (Air == "1"):
    GPIO.output(in2,GPIO.LOW)
    GPIO.output(in1,GPIO.HIGH)
    p.ChangeDutyCycle(100)
else:
    GPIO.output(in1,GPIO.LOW)
    GPIO.output(in2,GPIO.LOW)
if (Chau == "1"):
    GPIO.output(12,0)
else:
    GPIO.output(12,1)
```

```
temp = sensor.get_temperature()
firebase.put("/Aquarium", "/04Temperature", temp)
ch = firebase.get("/Aquarium/05Chauffage", "")
if (ch == "1"):
    GPIO.output(21,0)
else:
    GPIO.output(21,1)
```

Ce programme permet d'afficher la température de bac d'élevage des poissons, ainsi que d'activer et désactiver manuellement sa résistance chauffante, le tout en utilisant notre application.

- **Eclairage – Aération – Nutrition :**

→ Après avoir accéder à la base de données, nous téléchargeons les valeurs des variables « 02Eclairage », « 06Eclairage », « 03Nutrition » et « 01Aération », et les attribuons respectivement aux variables 'ec', 'ec1', 'nu' et 'ar'.

→ Si la valeur de la variable 'ec' ou bien la variable 'ec1' sont égale à 1, la sortie 36 sera à l'état 0 (LOW), sinon elle sera à l'état 1 (HIGH).

```
ec = firebase.get("/Aquarium/02Eclairage", "")
ec1 = firebase.get("/LaSerre/06Eclairage", "")
nu = firebase.get("/Aquarium/03Nutrition", "")
ar = firebase.get("/Aquarium/01Aération", "")
if ((ec == 1) or (ec1 == 1)):
    GPIO.output(36,0)
else:
    GPIO.output(36,1)
```

→ Si la valeur de la variable 'nu' est égale à 1, la sortie 38 sera à l'état 0 (LOW), sinon elle sera à l'état 1 (HIGH).

→ Si la valeur de la variable 'ar' est égale à 1, la sortie 24 sera à l'état 0 (LOW), sinon elle sera à l'état 1 (HIGH).

Ce programme démarre / arrête manuellement trois actionneurs (LED, diffuseur, distributeur de nourriture) en utilisant les boutons de l'application.

```
if (nu == "1"):
    GPIO.output(38,0)
else:
    GPIO.output(38,1)
if (ar == "1"):
    GPIO.output(24,0)
else:
    GPIO.output(24,1)
```

• Circulation d'eau :

→ Nous téléchargeons la valeur de la variable « 07Pompe » depuis notre base de données, et l'attribuons à la variable 'pm'.

Si la valeur de 'pm' est égale à 1, la sortie 26 sera à l'état 0 (LOW), sinon elle sera à l'état 1 (HIGH).

```
pm = firebase.get("/LaSerre/07Pompe", "")
if (pm == "1"):
    GPIO.output(26,0)
else:
    GPIO.output(26,1)
```

Ce programme démarre/arrête manuellement la pompe d'eau responsable à la circulation d'eau entre le bac de culture et le bac d'élevage des poissons.

Pour plus de clarté, nous avons expliqué le fonctionnement de la porte et de l'ensemble du système dans les deux organigrammes suivants :

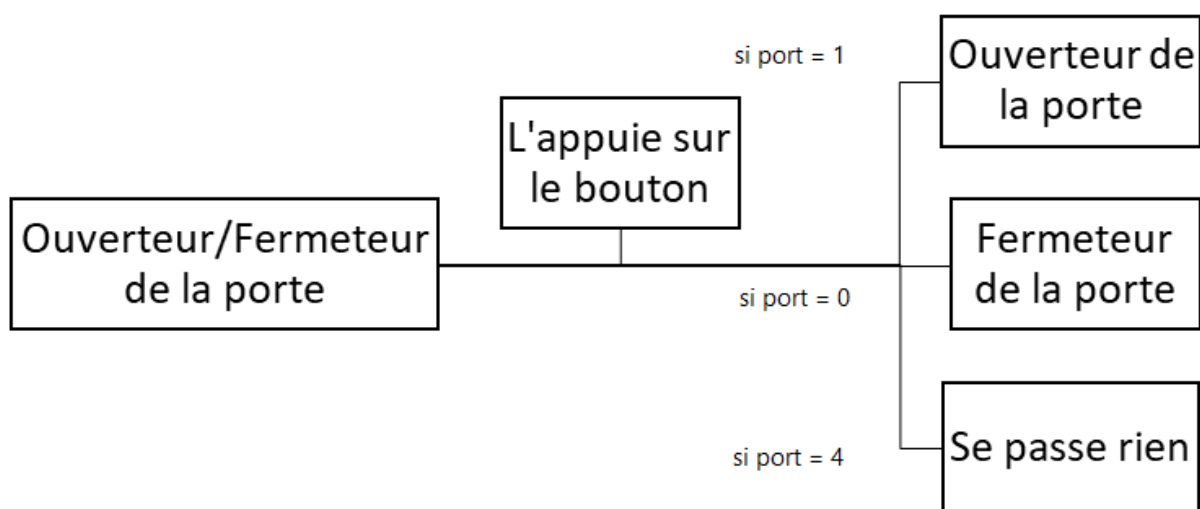


Figure III 19: organigramme représente le fonctionnement de la porte

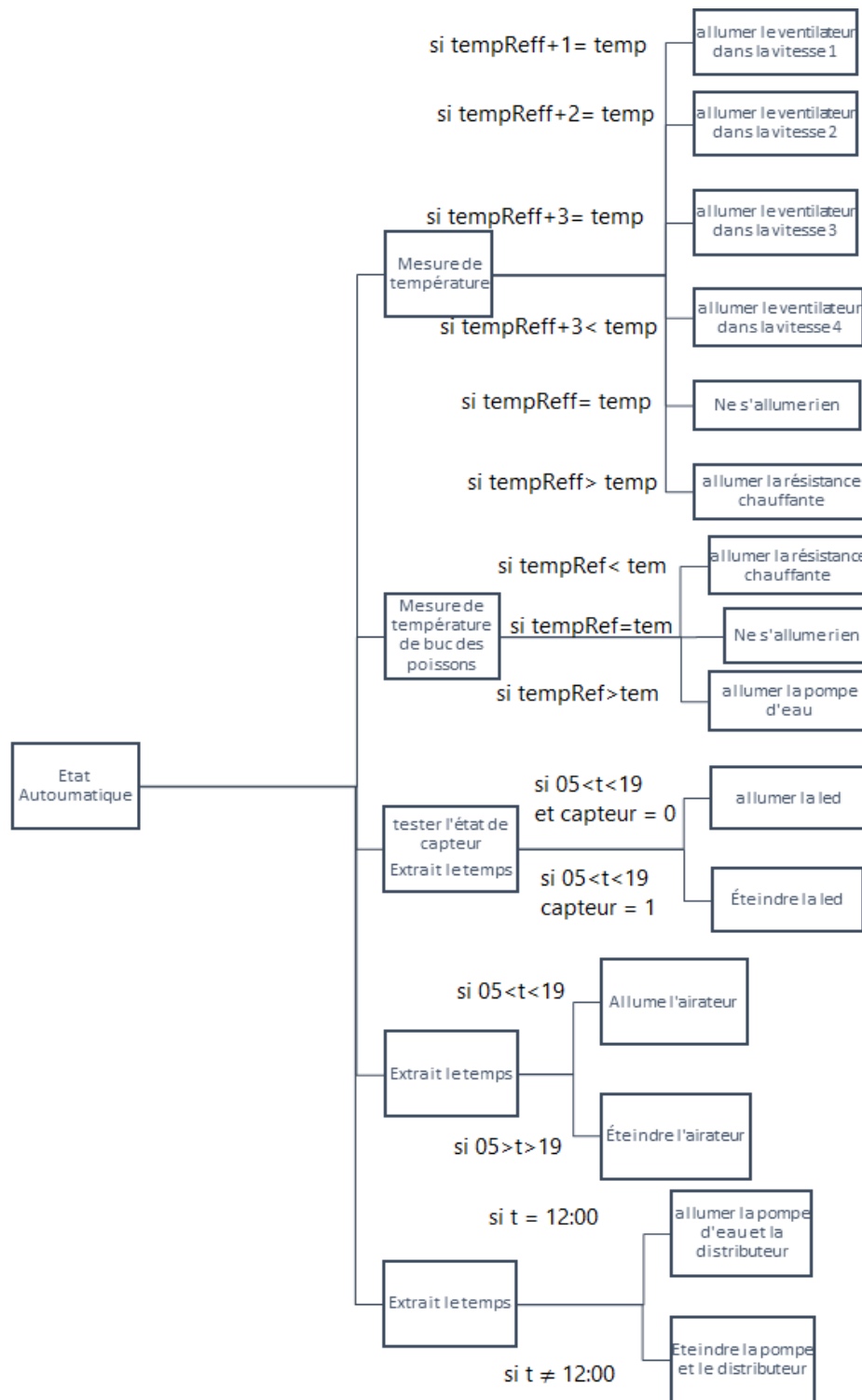


Figure III 20: organigramme représente le fonctionnement de système dans l'état automatique

Et pour l'état manuel, il suffit de présenter qu'on va appuyer sur des boutons pour changer l'état de chaque actionneurs.

III.3 Conclusion :

Dans ce chapitre nous nous attarder principalement sur pratique ayant trait à notre travail. C'est à dire nous avons expliqué en détail les différentes étapes de conception, de réalisation et de programmation de notre prototype du système aquaponique.

Dans ce contexte, nous avons commencé par la fabrication et d'installation de la serre, puis nous sommes passés à la partie opérationnelle, où nous avons expliqué la méthode d'installation de tous les capteurs, les actionneurs et les préactionneurs avec le raspberry pi.

Pour la partie de configuration, nous avons expliqué la méthode générale de configuration d'un Raspberry Pi pour le rendre utilisable en tant que pilote du système, nous avons donc téléchargé toutes les bibliothèques dont il avait besoin pour lire les capteurs, faire fonctionner les actionneurs.

Pour la partie programmation, Nous avons créé une application et une base de données Android, le but des deux est d'obtenir des informations de Raspberry Pi et de les afficher sur les activités de l'application à distance, et en même temps d'envoyer des commandes au Raspberry à exécute.

Afin d'obtenir des informations des capteurs, et pour démarrer et arrêter les moteurs, nous avons programmé le Raspberry Pi en langage Python pour qu'il fonctionne en deux modes, automatique et manuel.

Conclusion générale

L'objectif de ce travail est de concevoir et de réaliser un prototype d'un système aquaponique et lui associer une application androïde ainsi qu'une base de données dans le but de contrôler à distance ce système.

Nous avons structuré notre mémoire en trois chapitres, Dans le premier chapitre, nous avons essayé de donner une vision générale des systèmes aquaponiques en parlant de leurs types, composants et applications.

Dans le deuxième, nous avons abordé tout ce qui sera utilisé dans la réalisation du prototype. Au début nous avons présenté le contrôleur choisi il s'agit du « Raspberry pi », nous l'avons défini et présenté ses différents modèles, accessoires, et fonctionnalités. Ensuite, nous avons parlé des capteurs et actionneurs, leurs types, leur classification, ainsi que le mode de fonctionnement et la fonctionnalité de chaque composant. Enfin, nous avons défini les différents logiciels et applications utilisés lors des étapes de la réalisation.

Dans le dernier chapitre, nous avons commencé par la fabrication du prototype, sa démontage, en suite nous déclarons tous les capteurs et les actionneurs utilisés, leur association avec le raspberry pi. La partie suivante c'est la partie commande, nous avons créé la base de données qui va porter toutes les données de ce système, puis nous avons créé aussi une application Android qui va permettre de contrôler le système à travers une base de données. A la fin, nous avons préparé le raspberry pi de début, l'installation de système d'exploitation, sa configuration, et la programmation de notre système.

Pour conclure, notre projet ayant prouvé sa performance dans le domaine agricole et celui de l'aquaculture en créant un système écologique innovant doté d'un flux hydrique qui assure la vitalité de la flore et la faune en assurant un besoin vitale permanent et recyclable. Ce système ne se limite pas uniquement au domaine sus cité, son potentiel pratique peut être applicable à grande échelle notamment dans le domaine industriel, la production agricole et l'industrie du poisson, ce mini système Aquaponique doté d'un microcontrôleur Raspberry pi d'une vitesse maximale de 1.5GHZ aura toujours un espace pour innovation et amélioration en effet, divers type de système Aquaponique peuvent être produit à des échelles différentes en fonction de la taille de la serre, l'aquarium et la dimension et l'importance du projet pouvant assurer la production allant du petit aquarium aux grandes superficies tel qu'un barrage.

Références bibliographiques :

- [1]: aquaponie-Année de publication: 2014-Pages: #2-Auteur: Short, C.;Piccolo, A.;Sommerville, C-Éditeur: FAO/IOC
- [2]: (FAO 2014 ; Scott 2002).
- [3]: fédération France d'aquaponie / <http://www.aquaponiefrance.com> / Publié le septembre 21, 2015 par AquaponieFrance (**13/06/2020**)
- [4]: Nelson and Pade, Inc.
- [5]: FAO, 2014 ; Zhen, et al., 2015
- [6]: Treftz & Omaye, 2015
- [7]: <https://www.superprof.fr/ressources/scolaire/svt/schema-svt15/tous-niveaux-svt15/legender-annoter-schema.html> (**22/06/2020**)
- [8]: Pierre HARLAUT www.aquaponie.biz
- [9]: <https://www.agriculture-afrique.com/les-meilleures-plantes-pour-laquaponie/>
<http://aquaponie.bio/top-10-des-poissons-a-elever-en-aquaponie/> (**23/06/2020**)
- [10]: <http://aquaponie.bio/top-10-des-poissons-a-elever-en-aquaponie/> (**03/07/2020**)
- [11]: <https://www.zolux.com/conseils-de-pros/chauffage> (**03/07/2020**)
- [12]: <https://www.aquariophilie.org/articles/eclairage-221.html> (**01/08/2020**)
- [13]: https://www.aquariophilie-pratique.net/alimentation_quatre.htm (**01/08/2020**)
- [14]: Pierre HARLAUT www.aquaponie.biz
- [15]: https://fr.wikipedia.org/wiki/Raspberry_Pi (**03/08/2020**)
- [16]: <https://www.ldlc.com/fiche/PB00246555.html> (**03/08/2020**)
- [17]: <https://www.kubii.fr/content/77-demarrer-son-rpi> (**03/08/2020**) // Mémoire de fin d'études intitulée
« **conception et réalisation d'une maison intelligente via raspberry pi 3** »
- [18]: <https://www.ionos.fr/digitalguide/serveur/know-how/systeme-exploitation-raspberry-pi/>
(**07/08/2020**)
- [19]: https://fr.wikipedia.org/wiki/Raspberry_Pi#Sp%C3%A9cifications_mat%C3%A9rielles_et_architectures (**08/08/2020**)
- [20]: http://www.lycee-ferry-versailles.fr/si-new/4_2_aquis_info/tp_decouverte_capteurs/TP_decouverte_capteurs/index.html?Introduction.html
- [21]: <https://fr.wikipedia.org/wiki/Capteur> (**10/08/2020**)
- [22]: <https://shop.mchobby.be/fr/environnemental-press-temp-hrel-gaz/1535-dht22-extra-capteur-humidite-temperature-3232100015357-adafruit.html> (**11/08/2020**)
- [23]: <https://fr.hobbytronics.co.uk/ds18b20-temperature-sensor> (**11/08/2020**)

- [24] : <https://www.framboise314.fr/scratch-raspberry-pi-composants/module-detecteur-de-lumiere-a-ldr/> (12/08/2020)
- [25] : <https://junior.universalis.fr/encyclopedie/actionneur/> (12/08/2020)
- [26] : <https://www.astuces-pratiques.fr/electronique/le-moteur-a-courant-continu-principe> (13/08/2020)
- [27] : <https://www.climamaison.com/lexique/resistance.htm> (13/08/2020)
- [28] : https://fr.wikipedia.org/wiki/Lampe_%C3%A9lectroluminescente(13/08/2020)
- [29] : <https://www.ulmaagricola.com/fr/serres/equipements/extracteur/> (13/08/2020)
- [30] : <https://www.aqua-store.fr/s/821/aerateur-pour-aquarium> (15/08/2020)
- [31] : <https://pecquery.wixsite.com/arduino-passion/copie-de-le-detecteur-a-ultrasons-h>(15/08/2020)
- [32] : https://fr.wikipedia.org/wiki/Android_Studio (15/08/2020)
- [33] : <https://fr.wikipedia.org/wiki/Firebase> (15/08/2020)
- [34] : https://fr.wikipedia.org/wiki/Discussion_Wikip%C3%A9dia:Accueil_principal(16/08/2020)
- [35] : <https://www.framboise314.fr/mesure-de-temperature-1-wire-ds18b20-avec-le-raspberry-pi/> (17/08/2020)
- [36] : <https://www.raspberrypi-france.fr/guide/installer-raspbian-raspberry-pi/> (23/08/2020)
- [37] : <https://www.raspberrypi-france.fr/guide/configurer-raspbian/> (23/08/2020)
- [38] : <https://www.electronique-mixte.fr/kits-de-developpement-processeurs-et-microcontrolleurs/raspberry-pi/> (13/08/2020)

Annexes :

Annexe 01 : la technique PWM :

La modulation de largeur d'impulsions (MLI ; en anglais : Pulse Width Modulation, soit PWM), est une technique couramment utilisée pour synthétiser des signaux pseudo analogiques à l'aide de circuits à fonctionnement tout ou rien, ou plus généralement à états discrets.

En numérique, les signaux ont toujours une valeur LOW ou HIGH et le principe est de construire un signal qui est alternativement LOW et HIGH et de répéter très vite cette alternance.

Prenons par exemple LED, sur une période de 10ms, soit une fréquence de 100Hz. Si la LED est allumée pendant 5ms et éteinte pendant 5ms, comme sur la figure ci-dessous, l'impression sera une luminosité de 50% de la luminosité maximum.

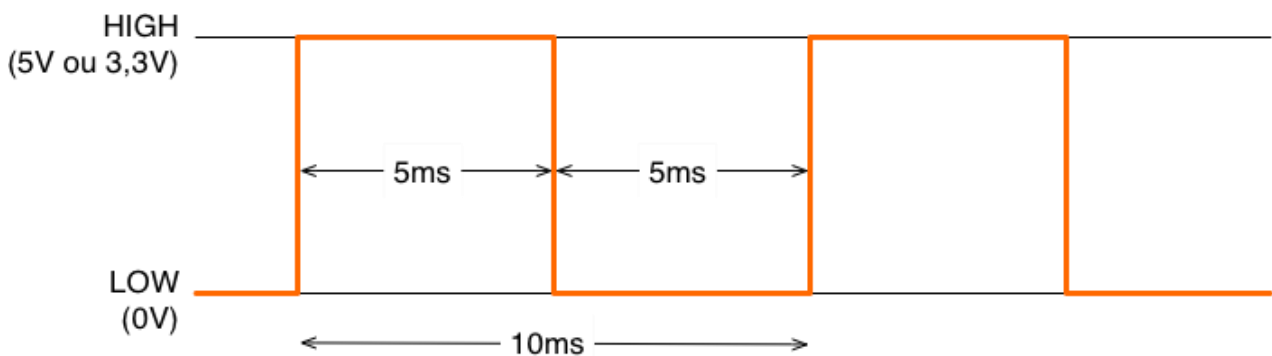


Figure 1 : PWM à 50%

Si la DEL est allumée pendant 1ms et éteinte pendant 9ms, l'impression sera une luminosité de 10% comme sur la figure ci-dessous.

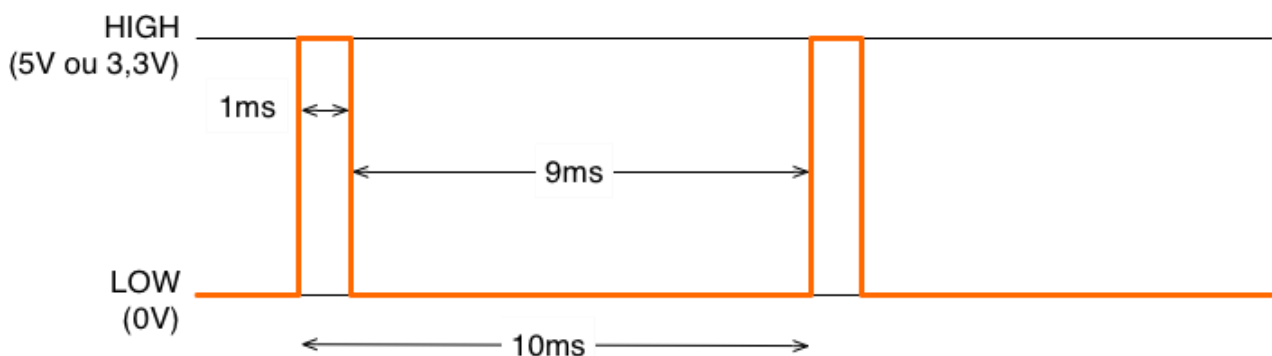


Figure 2: PWM a 10%

Le pourcentage de temps passé à l'état HIGH sur la période du signal est appelé le *rapport cyclique*. Il varie donc de 0%, le signal est tout le temps LOW, à 100%, le signal est tout le temps HIGH.