



وزارة البحث العلمي والتعليم العالي
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
جامعة عبد الحميد بن باديس مستغانم
Université Abdelhamid Ibn Badis Mostaganem
كلية العلوم و التكنولوجيا
Faculté des Sciences et de la Technologie
DEPARTEMENT DE GENIE ELECTRIQUE



N° d'ordre : M...../GE/2020

MEMOIRE

Présenté pour obtenir le diplôme de

MASTER EN ELECTRONIQUE

Option: Electronique des Systèmes Embarqués

Par

BENAHMED Roumaïssa

Intitulé du sujet
Conception et Réalisation d'un Distributeur Automatique de
Médicaments Reconfigurable à Distance

Soutenu en septembre 2020 devant le jury composé de :

Président :	MIMI Malika	Professeur	U.M.A.B
Examineur :	BENDANI Djazia	M.A.A	U.M.A.B
Rapporteur :	BENTOUMI Mohamed	M.C.B	U.M.A.B
Co-Rapporteur	BENOUALI Abdelhak	M.A.A	U.M.A.B

Année Universitaire 2019/2020

Remerciements

Remerciements

Je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

En second lieu, je tiens à remercier mes encadreurs monsieur BENTOUMI M. et monsieur BENOUALI A., pour leurs précieux conseil et leurs aides durant toute la période du travail.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à ce mémoire en acceptant d'examiner mon travail et de l'enrichir par leurs propositions.

Enfin, je tiens également à remercier ma famille et mes amis pour leur soutien et toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

Dédicaces

Je dédie ce mémoire à

Mes chers parents

Ceux qui ont été à mes côtés depuis ma naissance et ont été le premier soutien dans mon parcours académique et n'ont cessé de m'encourager et de me souhaiter du succès, considérant que ma réussite est leur réussite.

Mes chers sœurs Hanane et Sarah et mon chers petit frère Rayane

Ceux qui partagent l'amour et la fraternité avec moi et ont été à mes côtés dans les bons et les mauvais moments.

Mes chers amis

Qui je les considérais comme ma deuxième famille car ils m'ont soutenu dans mes études et ont partagé avec moi la plupart des temps d'étude.

Résumé

Résumé en Arabe, français et anglais

ملخص

احد اهم الاشياء عند الانسان هي الصحة و يجب عليه الحفاظ عليها لكن العالم لا يخلو من الامراض حتى انه يوجد امراض مزمنة تحتاج لمتابعة مستمر واخذ العلاج بانتظام، لكن هناك بعض المرضى يتهاونون في ذلك. هذا المشروع سيساعد على اصلاح ذلك حيث سيقوم بتوزيع الدواء بشكل منظم و آلي معتمدا على الوقت حيث التحكم بهذا الاخير يكون عبر هاتف ذكي بواسطة تطبيق Android الذي أنشأناه

كلمات مفتاحية :

موزع آلي، موزع حبوب، التحكم الآلي، نظام ال Android، بطاقة esp32، الأدوية، التحكم عبر ال WIFI.

Résumé

L'une des choses les plus importantes pour l'être humain est la santé, et il doit la préserver, mais le monde n'est pas sans maladies, de sorte qu'il y a des maladies chroniques qui nécessitent un suivi constant et de prendre un traitement régulièrement, mais il y a des patients qui négligent cela.

Dans ce projet, on a réalisé un distributeur automatique de médicament qui aidera à résoudre ce problème, car il permet de distribuer des médicaments de manière automatique et ordonnée en fonction du temps où ce dernier est contrôlé par un smart phone via une application Android qu'on a créé.

Mots clés :

Distributeur automatique, distributeur de comprimé, contrôle automatique, système Android, la carte esp32, les médicaments, contrôle par WIFI.

Abstract

One of the most important things for the human being's health, and he must preserve it, but the world is not without diseases, so there are chronic diseases which need to be followed up. Constant and take treatment regularly, but there are patients who neglect this.

This project will help solve this problem, as it will dispense the medication in an organized and automatic way, depending on the time, where it is controlled by a smart phone through an Android application that we have created.

Keywords:

Vending machine, pill dispenser, automatic control, Android system, esp32 card, the medication, Control via WIFI.

Liste des figures

Liste des figures

Figures	Page
Figure I-1-Des distributeurs automatiques dans une station du métro parisien.	4
Figure I-2-Distributeur automatique de chocolats Stollwerck, Compagnie générale des distributeurs automatiques de Berne, 1889.	5
Figure II-1- Schéma de la partie commande.	9
Figure II-2- L'interface de mon distributeur.	10
Figure II-3-le boîtier pour les médicaments en comprimé.	10
Figure II-4- Partie 1	11
Figure II-5- Partie 2.	11
Figure II-6- L'interface de l'Arduino.	13
Figure II-7- Liste de fichier.	13
Figure II-8- Liste des croquis.	14
Figure II-9- Liste d'outils.	14
Figure II-10- Barre d'outils.	14
Figure II-11-Environnement de développement du Android studio.	15
Figure II-12-Liste de fils.	16
Figure II-13- Première étape pour crée un nouveau projet.	16
Figure II-14- Choix du type de projet.	17
Figure II-15- La configuration du projet.	17
Figure II-16- L'interface finale.	18
Figure II-17- Barre d'outils.	18
Figure II-18- Image d'un moteur pas à pas.	19
Figure II-19-Composition du moteur pas à pas.	20
Figure II-20- Image d'un ESP32.	21
Figure II-21- Le schéma fonctionnel d'un ESP32.	24
Figure II-22- Les pin d'un ESP32.	24

Figure II-23- Image d'un ultrason.	25
Figure II-24-Mise en marche de capteur.	25
Figure II-25- Image d'un afficheur LCD.	25
Figure III-1- Schéma du branchement.	28
Figure III-2-Schéma du branchement.	29

Figures	Page
Figure III-3- Le prototype du projet.	33
Figure III-4- Vue du prototype à l'intérieur.	34
Figure III-5-Le résultats.	35
Figure III-5-Première étape de configuration.	37
Figure III-6- image des informations du smartphone brancher.	38
Figure III-7- Description de la méthode d'activation du mode développeur.	39
Figure III-8- Indication que le branchement et bon.	39
Figure III-9- L'application sur l'interface du smartphone.	40
Figure III-10- l'accueil de l'application.	40
Figure III-11- L'interface 2 de l'application.	41
Figure III-12-L'interface 3 de l'application.	41
Figure III-13- L'architecture de la communication entre l'application et esp32.	42

Sommaire

Remerciements.....	ii
Dédicaces	iv
Résumé en Arabe, français et anglais	vi
Liste des figures	ix
Sommaire	xi
Introduction Générale	2
Chapitre 01 : les distributeurs automatiques.....	4
Introduction :.....	4
Les distributeurs automatiques :	4
Définition :.....	4
Historique :	5
L'automatisation :	6
1.1. Définition :	6
1.2. Techniquesd'automatisation et les distributeurs :.....	6
Les types du distributeur automatique :.....	6
- Le distributeur de boissons chaudes :.....	6
- Le distributeur de boissons froides :.....	6
- Le distributeur automatique de snack :.....	7
- Le distributeur automatique mixte :	7
- Le distributeur automatique de billet :.....	7
Conclusion :	7
Chapitre 02 : Description du projet réalisé	9
Introduction :.....	9
1. Définition de mon distributeur :.....	9

2. Schéma fonctionnel :	9
3. Fonctionnement :	10
4. Les contraintes :	11
5. Autres additifs :	12
6. Les outils :	12
6.1. Les logiciels utilisés :	12
6.2 Matérielsutilisée :	19
Conclusion :	26
Chapitre 03 : Réalisation du projet	28
Introduction :	28
1- Moteur pas à pas :	28
1.1 Montage :	28
1.3 Programmation :	29
2- Le capteur de distance :	29
2.1 Montage :	29
2.2 Programmation :	30
3- Le prototype :	33
3- Application :	34
4- Lier l’application avec Arduino :	42
Conclusion ;	43
Conclusion générale	45
Bibliographie	47

Introduction générale

Introduction Générale

Au fil des années, la science s'est développée et plusieurs méthodes sont apparues qui aident la vie humaine à obtenir du confort grâce à la recherche et à l'invention.

L'électronique a beaucoup aidé dans ce développement auquel grâce à elle on a plusieurs appareils électroniques qui accomplissent le travail humain en peu de temps, comme les robots que l'on trouve dans les usines.

Parmi les machines inventées, il y a le distributeur automatique qui aide l'humain à être autonome dans ces achats, notre projet a une grande liaison avec ce domaine car c'est un distributeur de médicament à domicile qui va aider les patients à ne pas rater le temps de médicament pour assurer la protection de sa santé car elle est vraiment nécessaire.

Dans notre étude, on a débuté par le premier chapitre qui consiste à donner un aperçu sur les distributeurs, qui se trouvent dans le monde, leur définition, leur histoire, ces différents types, etc.

Dans le deuxième chapitre on a présenté la description du distributeur à réaliser, son fonctionnement et les outils utilisés (logiciel et matériel).

Le troisième chapitre englobe la partie pratique du projet, c'est à dire la réalisation. On a mentionné les différentes étapes de la mise en œuvre de notre prototype à côté d'une application Android qui permet la configuration du dispositif à distance (wifi).

Chapitre I

Chapitre 01 : les distributeurs automatiques

Introduction :

Les systèmes automatisés sont des systèmes qui peuvent faire de différentes tâches avec répétition automatiquement sans l'intervention humaine (sauf les cas exceptionnels) suivant un code précis.

Un automate est constitué par des capteurs pour l'envoi des informations, un contrôleur (le cerveau de la machine) qui utilise le code pour traiter les informations et envois des commandes vers les actionneurs qui permettent l'exécution de tâches.

Les distributeurs automatiques font partie de ces systèmes (les systèmes automatisés) mais leur fonctionnement est de délivrer différents produits selon le choix de l'utilisateur.



Figure I-1-Des distributeurs automatiques dans une station du métro parisien.

Les distributeurs automatiques :

Définition :

Un distributeur est un dispositif électromécanique qui offre des produits divers aux consommateurs. Les distributeurs se différencient selon la nature des produits distribués : carburant, aliments, boissons froides ou chaudes, billets bancaires, livres, DVD, ... etc.

Historique :

Il a été utilisé au début du XX^e siècle dans le métro parisien. C'est cependant avec l'industrie et l'urbanisation que l'on a vu se développer ce type de machines qui répondait à une demande sociale.



Figure I-2-Distributeur automatique de chocolats *Stollwerck*, compagnie générale des distributeurs automatiques de Berne, 1889.

Les distributeurs automatiques de boissons ont été mis au point en France par l'industriel marseillais *Henri Émile Schloesing* et *Benjamin Degremont*. Un brevet a été déposé par ces inventeurs en 1889 [1]. En 1889, ce distributeur a été présenté à l'exposition universelle de Paris. Le journal « La science illustrée » a publié un descriptif de cette machine appelée « distributeur automatique de liquides » le 30 janvier 1892. Un article du 29 novembre 1889 précise : « cette invention destinée à procurer des boissons chaudes aux ouvriers du port de Marseille, fonctionne à Paris place de la république. Moyennant 10 centimes, ce distributeur automatique fournit bière, vin ou café. Soixante-dix villes ont autorisé l'établissement de ces appareils schloesing ». Un brevet est déposé au grand-duché du Luxembourg par le représentant des inventeurs, *M. Aug Ligier*, le 14 février 1890. Un brevet canadien est délivré le 3 septembre 1890 pour les inventeurs d'un distributeur.

En juin 1967, le premier distributeur automatique de billet a été installé à l'agence Barclays d'Enfield, en Grande-Bretagne. Il distribuait des billets de 10 livres en échange d'un bon en papier inséré par le client qui tapait ensuite son code à quatre chiffres. Son inventeur, John Shepherd-Barron, travaillait chez « De La Rue » [1].

L'automatisation :

1.1. Définition :

Une automatisation est une technique ou un ensemble de techniques ayant pour but de réduire ou de rendre inutile l'intervention d'opérateurs humains dans un processus où cette intervention était coutumière.

1.2. Techniques d'automatisation et les distributeurs :

Ces machines font appel à un grand nombre de techniques d'automatisation :

- Les monnayeurs, qui permettent la détection et l'identification des pièces de monnaie et/ou des billets de banques.
- Les lecteurs de carte, de clés *cashless*, de puce sans contact (RFID) ou de porte-monnaie électronique. Des solutions de paiement comme *Moneo* ou l'emploi de smartphone commencent à apparaître.
- La mise à disposition des produits, avec des mécaniques en spires par exemple. Elles sont aujourd'hui également renforcées par des solutions d'ascenseur qui viennent collecter le produit libéré par les spires et le restituer en douceur au consommateur.

Les types du distributeur automatique :

Avec le temps les distributeurs automatiques sont beaucoup évolués. Dans nos jours, il en existe plusieurs types dans différents domaines selon les besoins. Chaque distributeur a son propre rôle et ses propres caractéristiques. [2]

Parmi eux, on mentionne les distributeurs suivants :

- Le distributeur de boissons chaudes :

Ce type est considéré comme l'un des distributeurs les plus répandus, on le trouve dans presque tous les lieux et les établissements publics où il supervise la fourniture de boissons chaudes de toutes sortes (café, cappuccino, chocolat chaud, ...) selon les souhaits du client.

- Le distributeur de boissons froides :

Ce type est similaire au type précédent, on le trouve souvent dans les gares de transport, les universités et les centres publics, mais il en diffère en termes de matériaux car ce sont des boissons

froides (jus, boisson gazeuse, ...) prêtes à la prendre immédiatement contrairement à un distributeur de boissons chaudes, la préparation prend du temps.

- Le distributeur automatique de snack :

Ce distributeur sert à distribuer un lunch ou un complément au lunch. Il contient de nombreuses options, car il existe des produits salés (chips, sandwich, salades, gâteaux salés,...) et des produits sucrés (fruits, petits gâteaux, barres chocolatées, sachets de bonbons,...).

- Le distributeur automatique mixte :

Il s'agit de la combinaison d'un distributeur de boissons fraîches et de boissons chaudes, ou de boissons et de snacks. Ils sont simplement constitués de deux appareils juxtaposés.

- Le distributeur automatique de billet :

Il diffère sensiblement des distributeurs précédents (fonctionne en payant et en prenant), cela se fait en plaçant une carte bancaire et en retirant un montant du compte, par conséquent cela contribue à réduire le travail des employés de la banque et la surpopulation.

Conclusion :

Comme on remarque dans ce chapitre il y a des différents types de distributeur ou chaque un a ses propres caractéristiques et mode d'emploi, cette machine a aidé l'homme à être autonome et lui accorde du confort dans sa vie quotidienne.

Chapitre II

Chapitre 02 : Description du projet réalisé

Introduction :

La santé est l'une des choses les plus importantes pour les humains, et en raison de la présence de certaines maladies chroniques qui obligent le propriétaire à prendre au sérieux la prise de médicaments régulièrement, mais certaines personnes négligent cela, ce qui conduit à des résultats négatifs. Donc notre projet consiste à proposer une réalisation d'un distributeur automatique de médicaments pour assister les personnes à domicile qui ont besoin.

1. Définition de mon distributeur :

Ce sera un point de contact entre la médecine et l'automatisation, car il assistera le patient et distribuera le médicament selon un calendrier donné par le médecin de manière ordonnée.

2. Schéma fonctionnel :

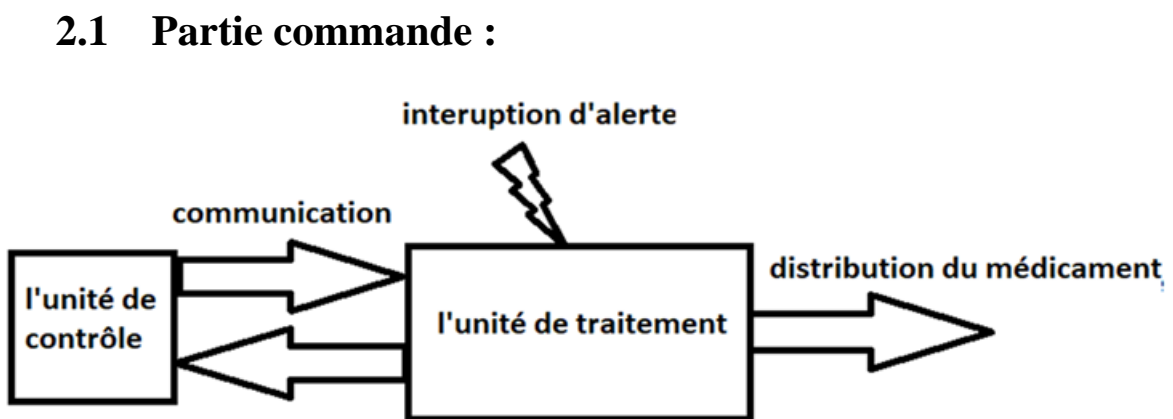


Figure II-1- Schéma de la partie commande.

2.2 Partie opérative :

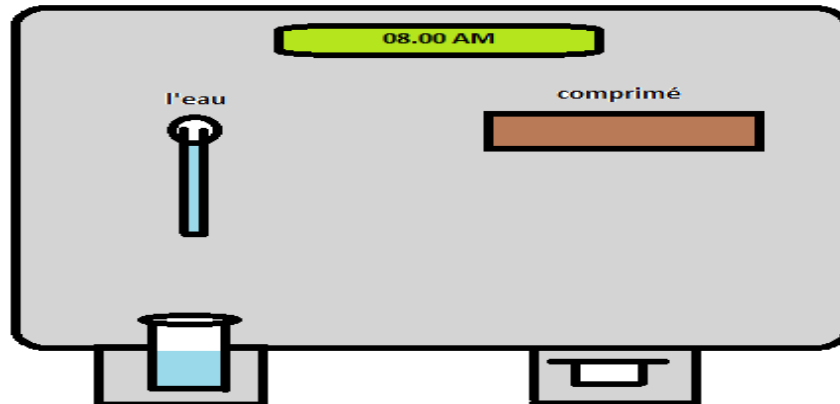


Figure II-2- L'interface de mon distributeur.

3. Fonctionnement :

Étant donné que l'appareil a beaucoup à voir avec le temps, il doit y avoir une unité de contrôle qui envoie des informations de temps à l'unité de traitement et puis le médicament sera distribué selon l'ordonnance donné par le médecin par exemple à 8h le malade doit prendre un comprimé de Glucophage (cas d'un diabète).

Il se trouve de nombreuses formes de médicament (Comprimé, médicament qui coule dans l'eau, sirop) et chaque un a ses propriétés. Pour notre projet on a pris en considération les médicaments en forme de comprimés.

.1 Distribution des comprimés :

A l'aide du boîtier qui se trouve dans la figure II-3 on fait la distribution du comprimé, c'est-à-dire le moteur pas à pas fait tourner la partie 1 qui fait pousser le comprimé vers le trou qui se trouve dans la partie 2 ce qui lui permet de tomber vers un récipient externe. [3]



Figure II-3-Le boîtier pour les médicaments en comprimé.

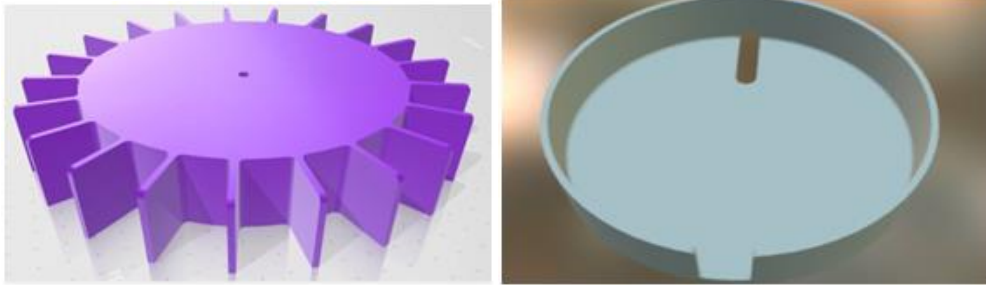


Figure II-4- Partie 1. Figure II-5- Partie 2.

.2 Distribution de l'eau :

A chaque fois, où on a un comprimé jeté, la pompe se met on fonctionnement et fait couler l'eau jusqu'à obtenir la quantité souhaitée. Il y a une autre façon si le patient a besoin de boire de l'eau il peut appuyer sur un bouton qui met la pompe on marche pour servir l'eau.

.3 Le contrôle du temps :

Le temps de distribution sera contrôlé par une application Android qui permet de configurer l'horloge du système par une méthode facile et simple.

.4 L'alerte de danger :

Dans le cas où le patient est exposé à une situation dangereuse, un bouton peut être pressé pour qu'un message soit envoyé au smartphone de la personne en charge et à l'hôpital le plus proche.

4. Les contraintes :

Comme déjà indiqué, c'est un système lié à la santé donc il doit respecter les contraintes de la santé et de la prévention, c'est-à-dire :

- Respectez les règles d'hygiène,
- Respecter les paramètres de conservation tenant compte :
- D'éviter de mélanger les médicaments ;
- -Sur l'aspect d'environnement (humide, frais, sec).
- Le flux de la lumière, certain médicament doit être conservés dans un endroit à l'abri de la lumière.

- Les températures, certains médicaments doivent être conservés selon des fluctuations différentes d'un autre (température ambiante, basse température 4-8 °C)
- Respecter les doses des médicaments selon les unités de mesure.

5. Autres additifs :

- Cas d'une personne diabète, à différents intervalles de temps (selon le médecin) on mesure la glycémie et on sauvegarde l'information.
- La même chose se fait pour un hypertendu, on fait mesurer la tension et on sauvegarde l'information.
- Le malade prend le médicament à l'aide d'un bras robot.
- On peut ajouter une distribution du sirop et/ou des sachets de médicaments qui coule dans l'eau.

6. Les outils :

6.1. Les logiciels utilisés :

a) Arduino :

Le logiciel Arduino est un environnement de développement (IDE) open source et gratuit, téléchargeable sur le site officiel Arduino.

Il se caractérise par le fait que son langage de programmation est facile et simple à apprendre où la syntaxe est similaire au C++ et JAVA. Ce logiciel nous permet de programmer une carte Arduino.

- Environnement de développement :

Lorsqu'on lance le logiciel, on aura l'interface indiquée dans la Figure II-6. On note qu'il y a quatre sections principales.

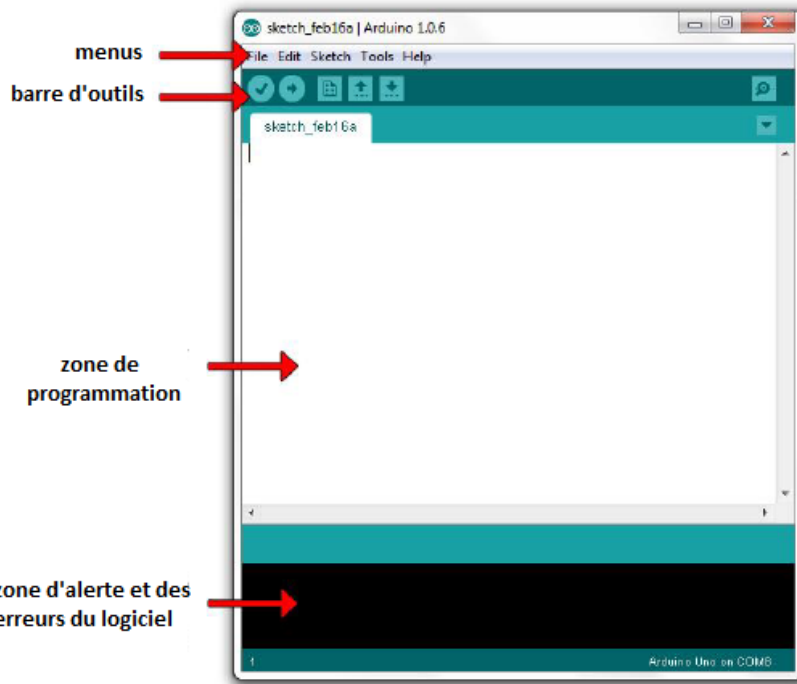


Figure II-6- L'interface de l'Arduino.

- Menus :
- Liste de fichier

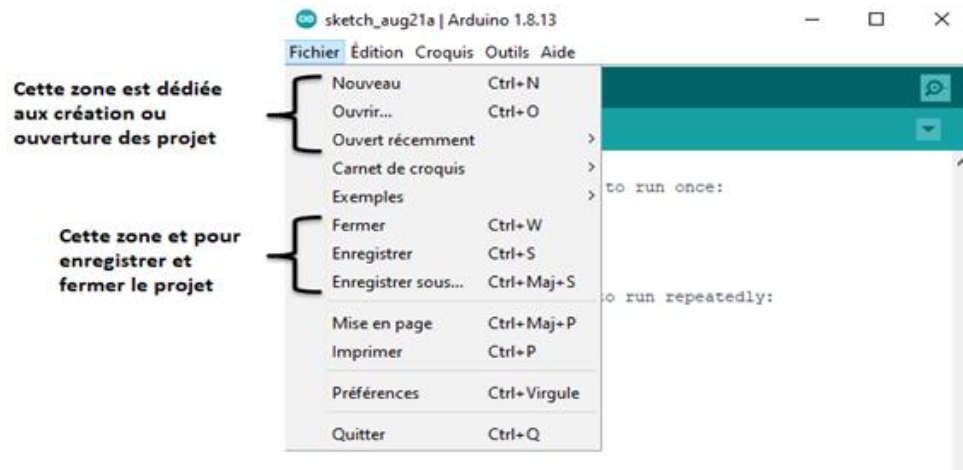


Figure II-7- Liste de fichier.

- Liste des croquis :

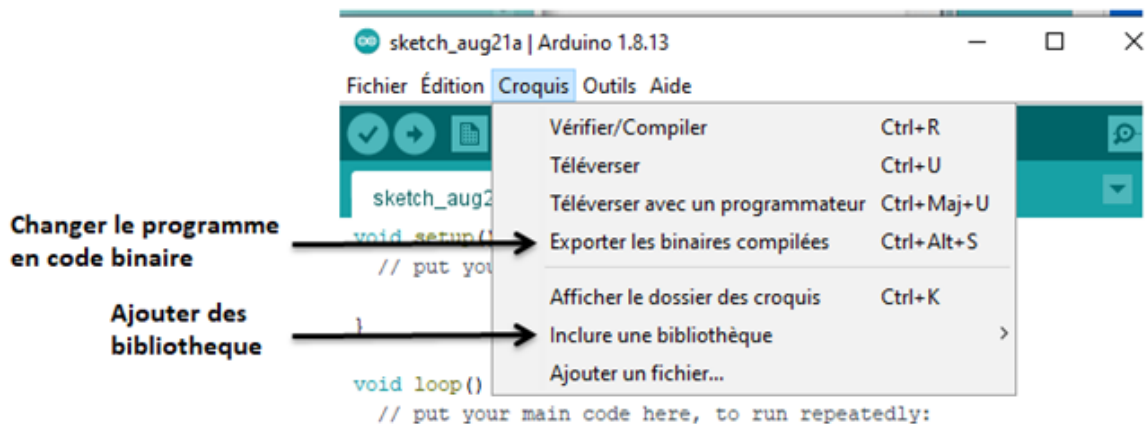


Figure II-8- Liste des croquis.

- Liste d'outils :

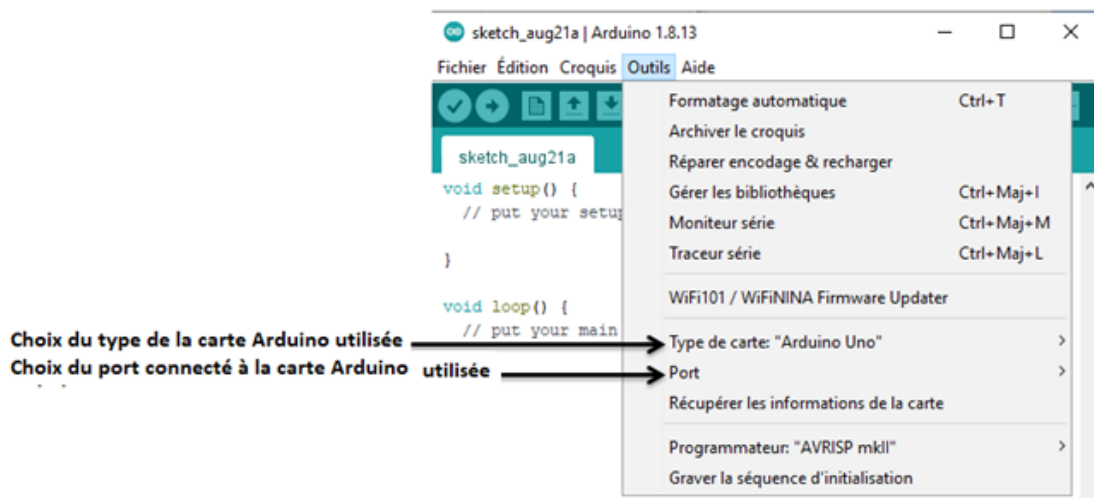


Figure II-9- Liste d'outils.

- Barre d'outils :



Figure II-10- Barre d'outils.

- 1 : Compiler le programme uniquement pour savoir s'il y a des erreurs.
- 2 : Compiler le programme et l'envoyer vers la carte.
- 3 : Créer un nouveau projet.
- 4 : Ouvrir un projet existant.
- 5 : Enregistrer le projet actuel.
- 6 : Ouvrir le terminal de communication série.

Remarque : pour l'installation du logiciel le lien est au niveau de la bibliographie [9].

b) 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. Il possède deux langages de programmation **JAVA** et **Kotlin**.

➤ Environnement de développement :

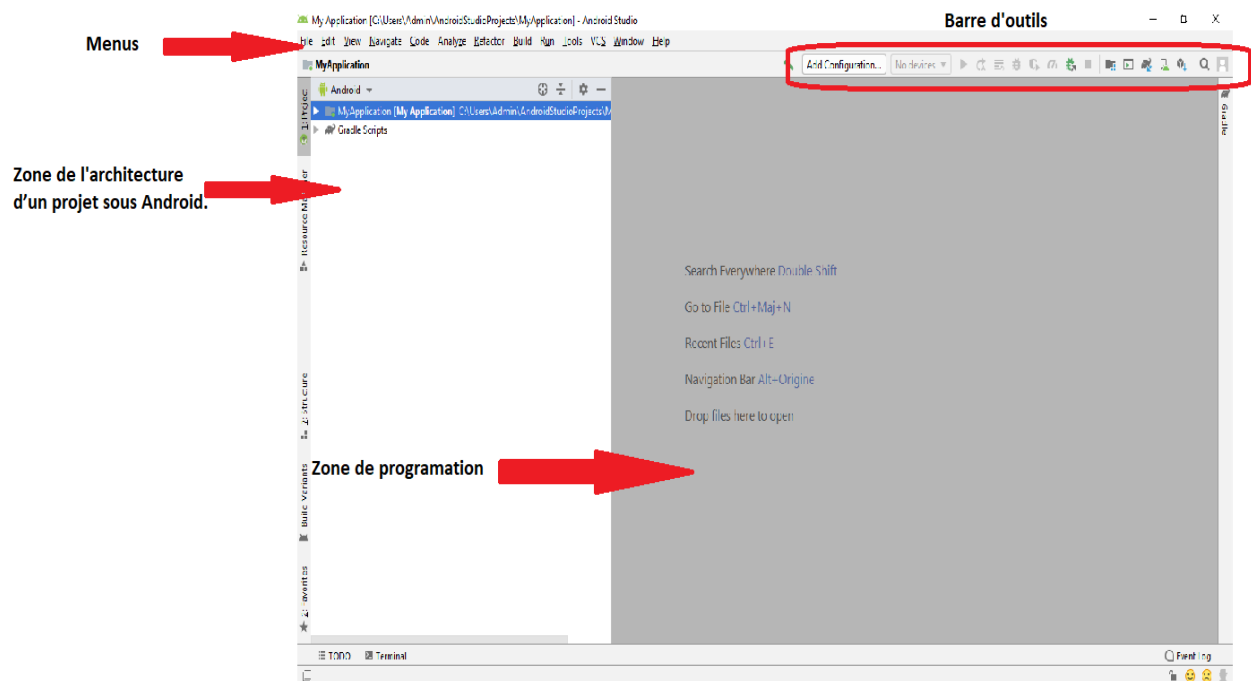


Figure II-11-Environnement de développement de l'Android studio.

- Menus :
- Liste de fils

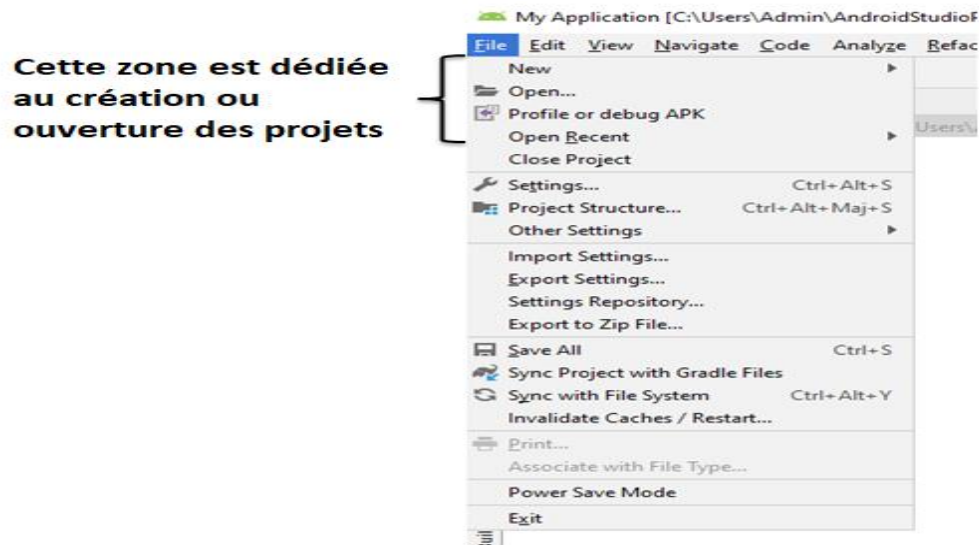


Figure II-12-Liste de fils.

- Création d'un projet :

Pour crée un projet il faut aller au fils --> new --> new project...

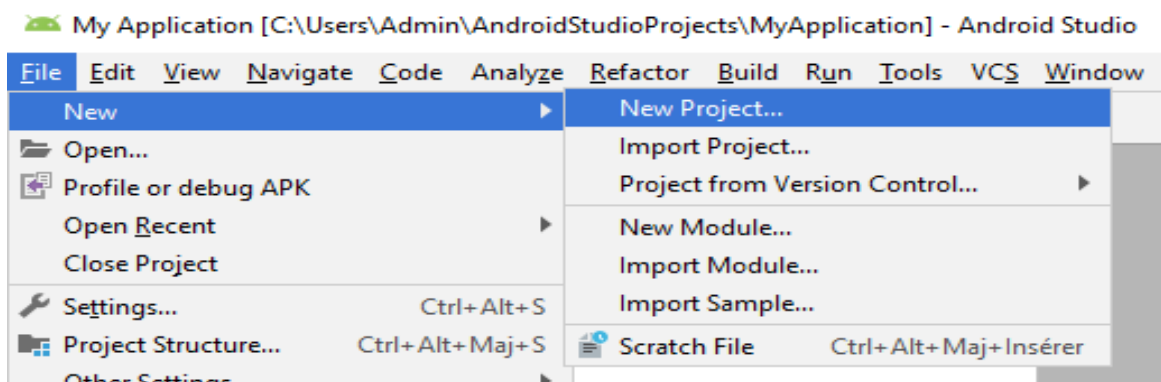


Figure II-13- Première étape pour crée un nouveau projet.

On trouve la fenêtre suivante :

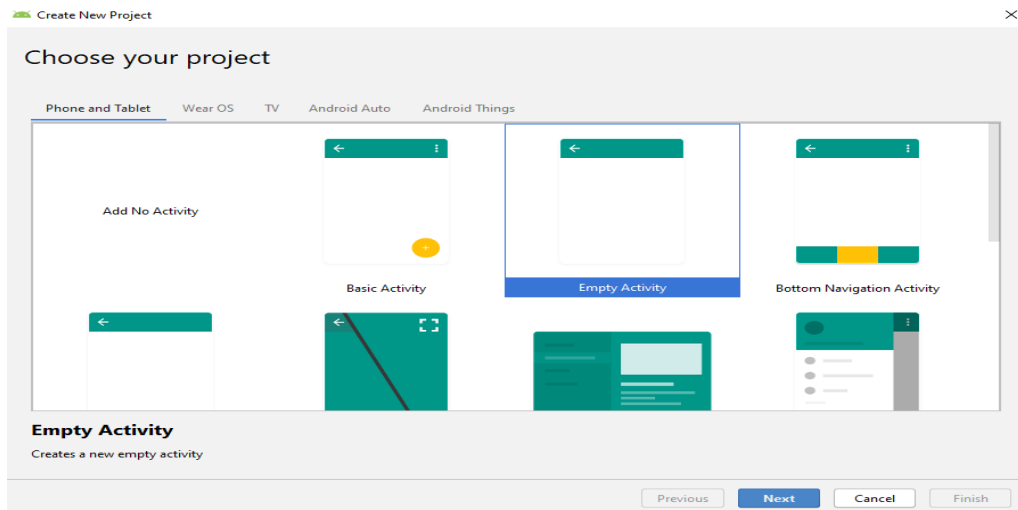


Figure II-14- Choix du type de projet.

On choisit « *emptyactivity* » ensuite « *next* » et on fait la configuration de notre projet c'est-à-dire on spécifie le nom, le package, l'endroit de sauvegarde du projet et le langage utilisé.

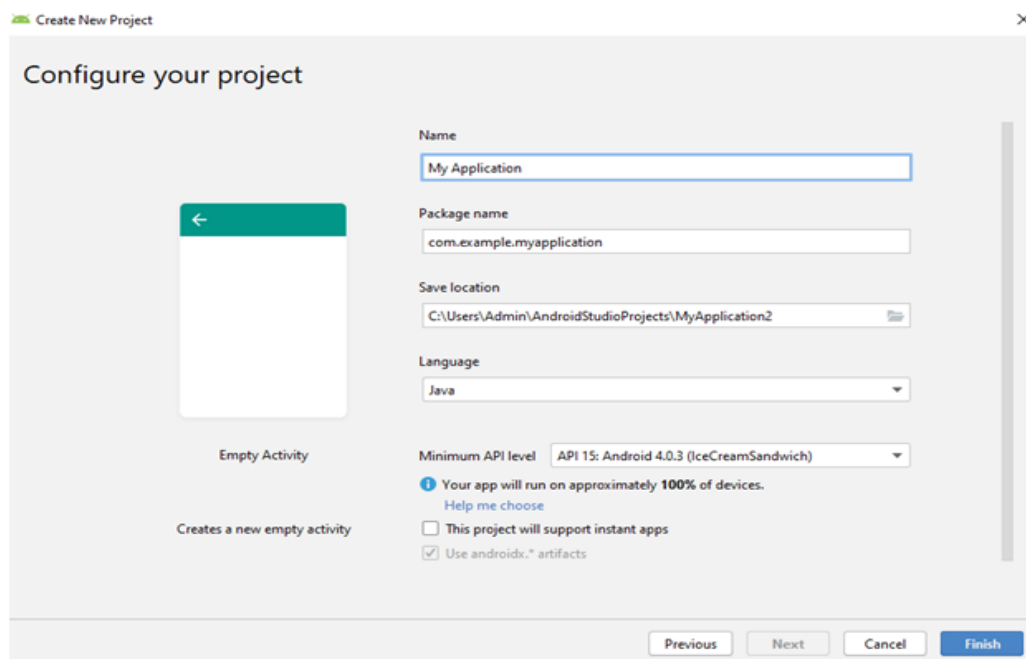


Figure II-15- La configuration du projet.

A la fin on clique sur « *finish* » on trouve la figure suivante.

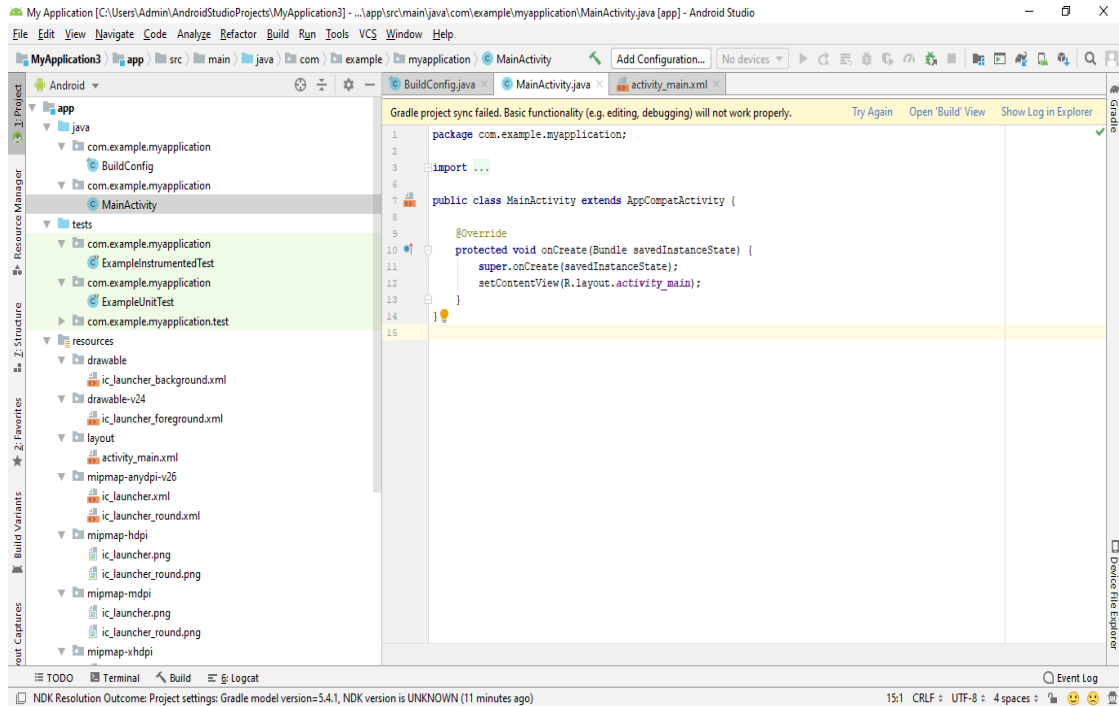


Figure II-16- L'interface finale.

➤ Barre d'outils :



Figure II-17- Barre d'outils.

- 1) : Mise en fonction.
- 2) : Appliquer le changement et redémarrer l'activité.
- 3) : Appliquer le changement du code.
- 4) : Débogage et correction.

- 5) : Mise en marche avec couverture.
- 6) : Le profile.
- 7) : Attachement avec le processus Android.
- 8) : Botton d'arrêter le fonctionnement.
- 9) : Structure du projet.
- 10) : Mis en fonction quoi que ce soit.
- 11) : Synchronisation du projet avec les fichiers Gradle.
- 12) : Gestionnaire de l'AVD.
- 13) : Gestionnaire du SDK.

Remarque : pour l'installation du logiciel le lien est au niveau de la bibliographie [10], avec une vidéo explicative [11].

6.2 Matérielsutilisée :

Le moteur pas à pas : Le moteur pas-à-pas appartient au groupe des moteursynchrones et dispose d'un élément moteur pivotant doté d'un arbre, le rotor, et d'un élément moteur fixe, que l'on appelle le stator. Tandis que le rotor agit comme un aimant permanent, le stator, lui, est composé de bobines d'excitation disposées en quinconce et qui produisent un champ magnétique. Celui-ci est à la base des capacités de positionnement du moteur et marque la différence de ce moteur par rapport aux servomoteurs. Alors que ces derniers utilisent différents capteurs de mesure de position et les valeurs qui en découlent, le moteur pas-à-pas, parfois aussi appelé stepper, fonctionne sans aucun capteur.



Figure II-18- Image d'un moteur pas à pas.

- a) En tant que moteur synchrone sans balai, le moteur pas-à-pas a pour principale tâche et fonction de transformer des impulsions électriques en mouvement mécanique. Le rotor se déplace

ainsi par à-coup, et le nombre des pas ou des positions par tour peuvent être définies. Par conséquent, la quantité des pas de chaque moteur pas-à-pas détermine le changement d'angle par pas. Si une rotation de 360° est divisée par exemple en 200 pas, alors à chaque pas, le rotor bougera précisément de $1,8^\circ$ — ce qui est l'angle de pas le plus couramment utilisé avec les moteurs pas-à-pas. Le changement de position tout à fait régulier est d'une extrême importance et fait de ce moteur ce qu'il est. C'est la raison pour laquelle aucun signal de retour n'est nécessaire et qu'au lieu de cela, la direction peut être mesurée en temps réel au moyen des impulsions données.

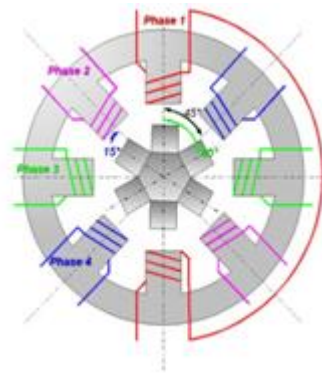


Figure II-19-Composition du moteur pas à pas.

Afin de faire bouger le rotor, autrement dit l'arbre, l'application d'une tension constante, comme c'est le cas pour les moteurs à courant continu, n'est pas suffisante. Au contraire, les bobines d'excitation du moteur pas-à-pas sont alimentées en tension de manière ciblée et temporairement non alimentées (unipolaire) ou par inversion des pôles (bipolaire). Ces pilotages multiples des moteurs permettent de commander à la fois des modèles unipolaires et bipolaires.[4][5]

➤ Type du moteur pas à pas :

Il existe trois types de moteurs pas à pas :

- Le moteur à réluctance variable.
- Le moteur à aimants permanents.
- Le moteur hybride, qui est une combinaison des deux technologies précédentes.

➤ Les domaines d'application du moteur pas-à-pas :

Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position en boucle ouverte, typiquement dans les systèmes de positionnement et d'indexation.

L'usage le plus connu du grand public est dans les imprimantes classiques et imprimantes 3D, les scanner et les platines vinyles de DJ. Mais ils sont présents dans de nombreuses applications telles : les photocopieurs, imprimantes bancaires, robotique, instrumentation, pompes à perfusion, pousse-seringues, système de positionnement sur machine industrielle et machine-outil.

b) ESP32 :

ESP32 est une série de microcontrôleurs de type système sur une puce (SoC) d'*EspressifSystems*, basé sur l'architecture *XtensaLX6* de *Tensilica* (en), intégrant la gestion du Wi-Fi et du Bluetooth en mode double, et un DSP. Son support Wi-Fi et Bluetooth, en fait un système apprécié dans le domaine de l'internet des objets.



Figure II-20- Image d'un ESP32.

Ce SoC rencontre un certain succès depuis quelques années à la fois pour son coût, ses capacités et son intégration dans un nombre croissant de systèmes. [6]

➤ Les caractéristiques techniques des ESP32 :

- Processeurs :
 - CPU : Xtensa double-cœur (ou simple-cœur), microprocesseur LX 32 bits, fonctionnant à 160 ou 240 MHz et fournissant jusqu'à 600 DMIPS.
 - Coprocesseur ultra basse consommation (ULP).
- Mémoire : 520 KiO SRAM.
- Connectivités sans-fil :
 - Wi-Fi : 802.11 b/g/n.
 - Bluetooth : v 4.2 BR/EDR and BLE jusqu'à v 5.0 et v 5.1.
- Interfaces de périphériques :
 - 12-bit Segmentation sur les ADC (SAR ADC) jusqu'à 18 canaux.
 - 2 × 8 bit DAC.
 - 10 × capteurs de touché (GPIO de capteur capacitif (en)).
 - × SPI.
 - 2 × interfaces I²S.
 - 2 × interfaces I²C.
 - × UART.
 - contrôleur hôte SD/SDIO/CE-ATA (en)/MMC/eMMC.
 - contrôleur esclave SDIO/SPI.
 - interface MAC Ethernet avec DMA dédié et support du protocole de temps précis IEEE 1588.
 - Bus de données CAN 2.0.
 - contrôleur infrarouge distant (TX/RX, jusqu'à 8 canaux).
 - Moteur PWM.
 - LED PWM (jusqu'à 16 canaux).
 - Capteur à effet Hall.
 - préamplificateur analogique ultra-basse consommation.
- Sécurité :
 - Standard de sécurité supportant complètement IEEE 802.11, incluant WPA/WPA2 et WAPI de WFA.
 - Secure boot (démarrage sécurisé).
 - Chiffrement de la Flash.

- 1024-bit OTP, jusqu'à 768 bit pour les clients.
- Accélération matérielle du chiffrement : AES, SHA-2, RSA, ellipticcurvecryptography (ECC), générateur de nombre aléatoire (en) (RNG).
- Gestion de l'énergie :
 - low-dropout regulator (en) interne.
 - Domaines d'alimentation individuels pour le RTC.
 - Alimentation en sommeil profond de 5 μ A.
 - Réveil depuis des interruptions GPIO, timer, mesure ADC, interruption du capteur de touché capacitif.
- Outils de développement :

Le ESP32 peut être programmé à partir de plusieurs outils de programmation, on mentionne les outils suivants :

- Arduino IDE avec le module ESP32 *ArduinoCore*.
- Espruino.
- FAUST, langage de programmation de traitement de données audio, utilisant son DSP3.
- Lua RTOS pour ESP32.
- MicroPython, une variante pour l'embarqué du langage Python.
- mruby (en), une variante pour l'embarqué du langage Ruby.
- NodeMCU.
- Schéma fonctionnel :

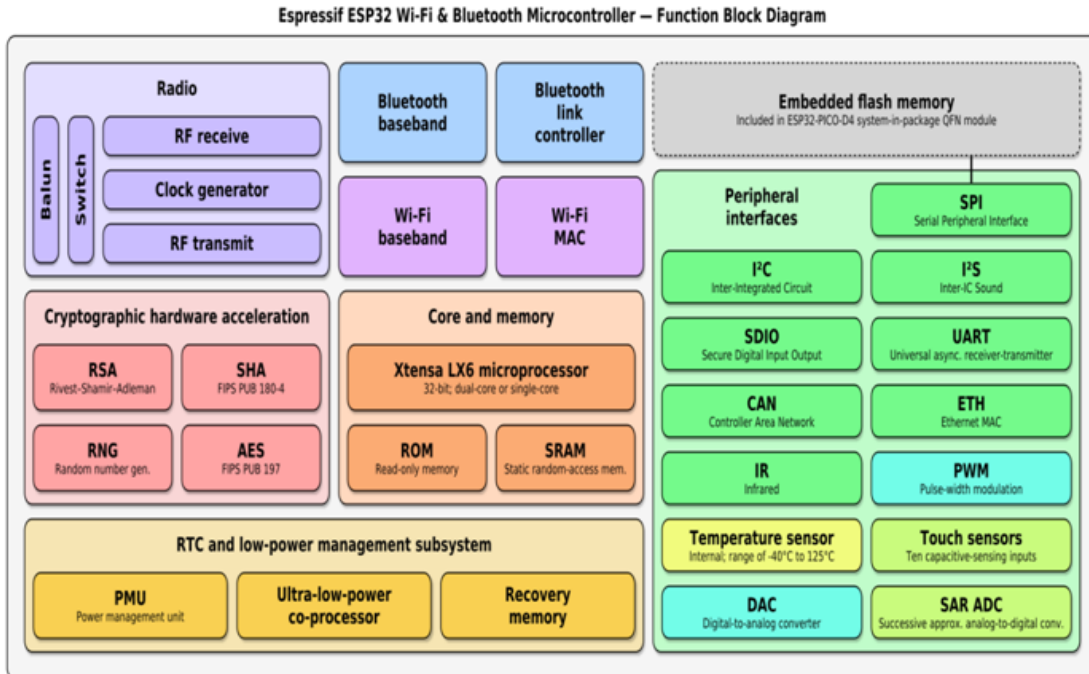


Figure II-21- Le schéma fonctionnel d'un ESP32.

➤ Description des ports :

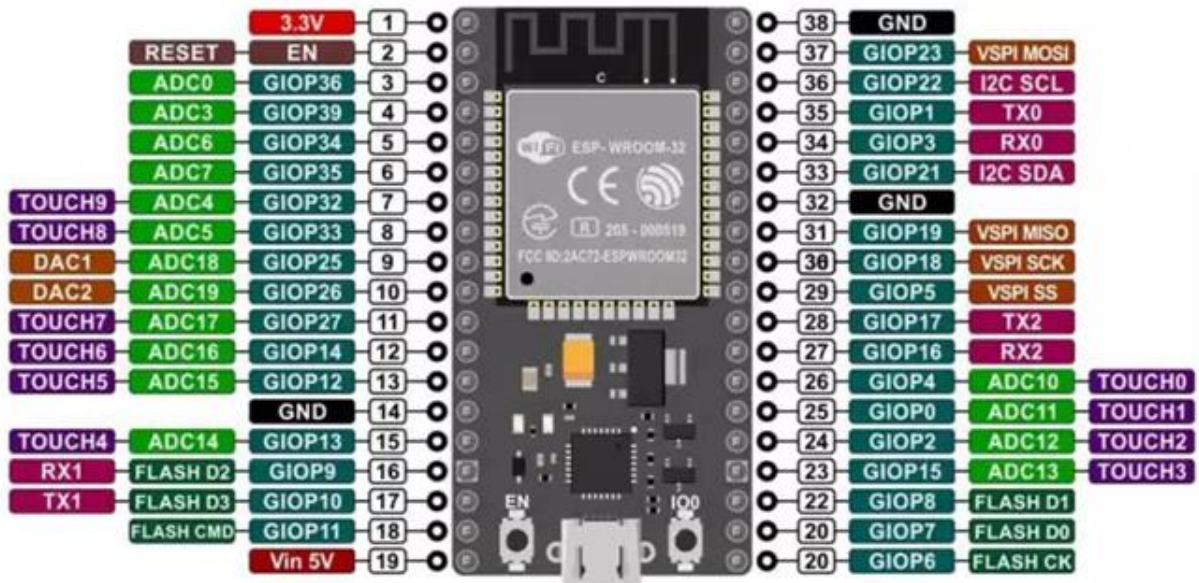


Figure II-22- Les pin d'un ESP32.

c) Capteur de distance à ultrason :

C'est un composant électronique qui aide à capter la distance entre l'objet et l'obstacle, il a été inventé par l'inspiration de la façon de déplacement des chauves-souris.



Figure II-23- Image d'un ultrason.

➤ Le principe des ultrasons :

Un capteur à ultrasons émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho. [7]

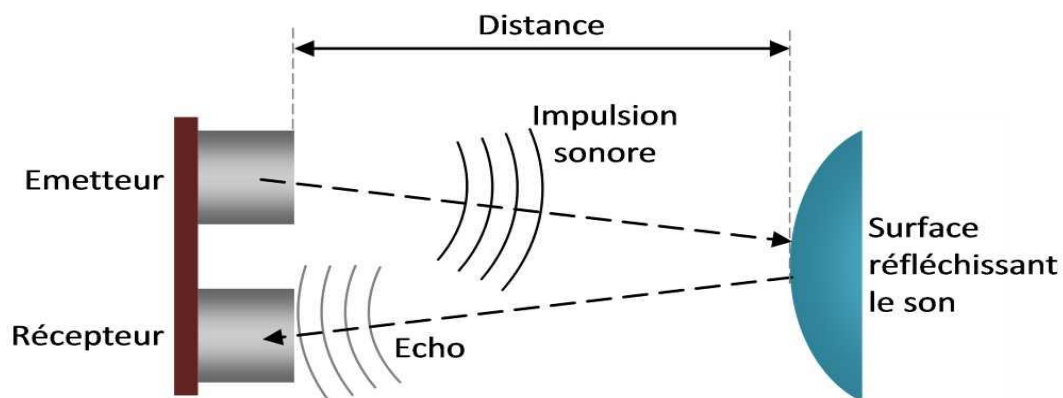


Figure II-24-Mise en marche de capteur

d) Afficheur LCD :

L'afficheur LCD (*Liquid Crystal Display*), autrement appelé afficheur à cristaux liquides est en particulier une interface visuelle entre un système (projet) et l'homme (utilisateur). Son rôle est de transmettre les informations utiles d'un système à un utilisateur. Il affichera donc des données susceptibles d'être exploiter par l'utilisateur d'un système



Figure II-25- Image d'un afficheur LCD.

e) Autre

- Plaque d'essais.
- Fils de connectivité.
- LED.
- Pompe à eau.
- Fer a soudé et l'étain.
- Résistances de protection.
- Buzzer.

Conclusion :

Dans ce chapitre on a fait une description pour notre projet et on a mentionné les outils utilisé pour la réalisation, on remarque que chaque outil a ces propres caractéristiques.

Chapitre III

Chapitre 03 : Réalisation du projet

Introduction :

Après avoir une vue générale et le cahier de charge du projet, il faut passer à la réalisation et la programmation sans oublier les tests et essais pour évaluer la réalisation. Donc dans ce chapitre on va présenter le travail à réaliser dans notre projet de fin d'étude.

Remarque : Vu que l'ESP32 peut être programmé avec Arduino IDE, et la gestion des entrées/sorties est similaire à un *ArduinoUno*, alors les tests ont été réalisés sur un *ArduinoUno*.

1- Moteur pas à pas :

1.1 Montage :

Mon moteur a besoin d'un driver pour le contrôler, il a également un branchement facile comme c'est indiqué dans la figure ci-dessous.

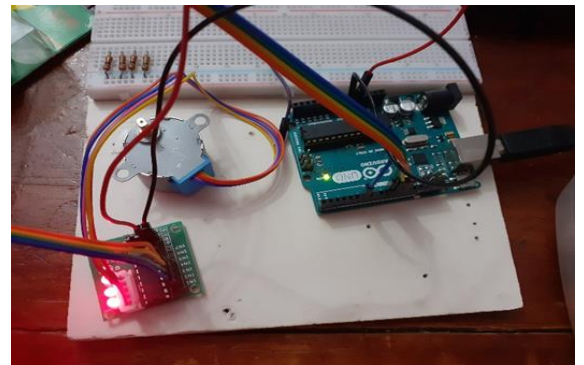
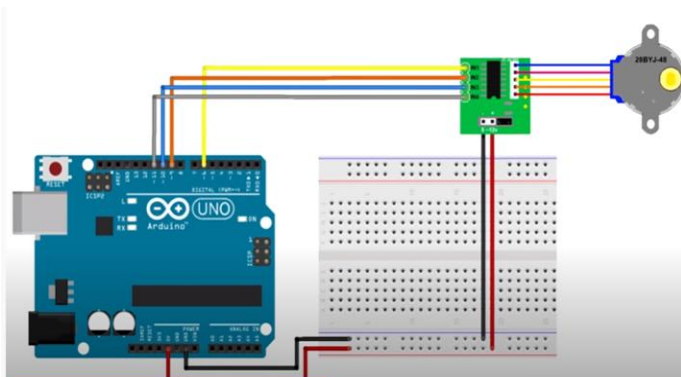


Figure III-1- Schéma du branchement.

1.3 Programmation :

```
#include <Stepper.h> //Inclure la bibliothèque
int nmb=48*64;      //nombre de pas
Stepper mot(nmb,9,11,10,6); // Attachement des pins
void setup() {
  mot.setSpeed(9);
  Serial.begin(9600);
}

void loop() {
  Serial.println("sens");
  mot.step(300); //nombre de rotation des bobine
  delay(1000);
}
```

Résultat : On remarque que le moteur tourne avec un certain degré après chaque instant choisi par delay.

2- Le capteur de distance :

2.1 Montage :

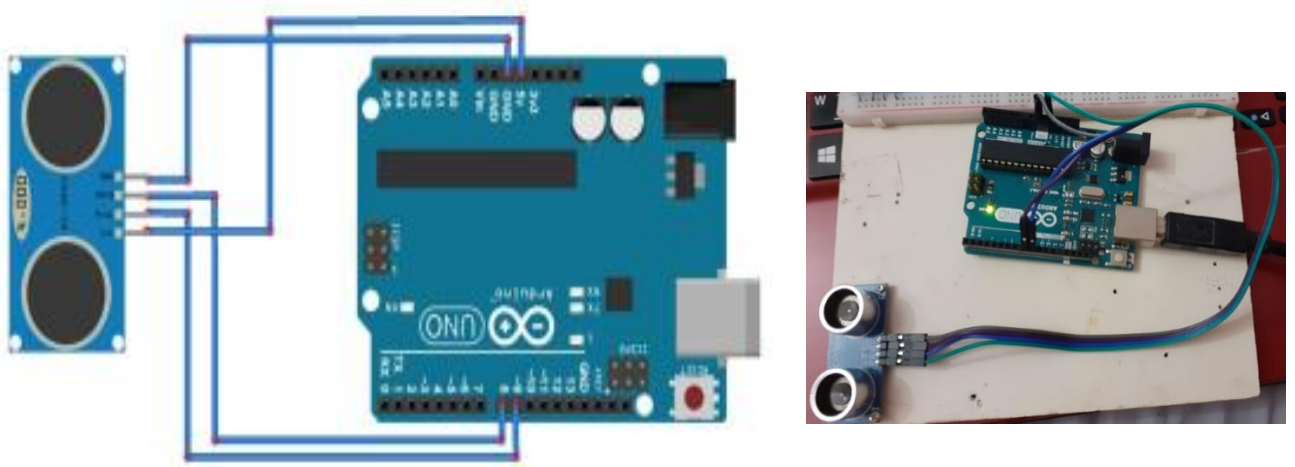


Figure III-2-Schéma du branchement.

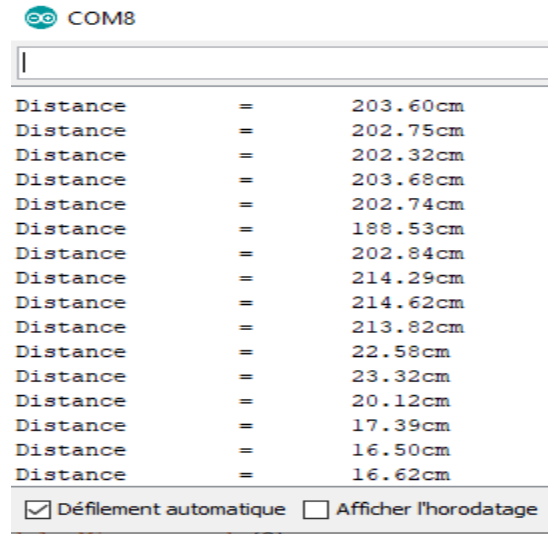
2.2 Programmation :

- Lecture de la distance :

```
const int TrigPin = 9;
const int EchoPin = 8;
float cm;
void setup()
{
  Serial.begin(9600);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}
void loop()
{
  digitalWrite(TrigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin, HIGH); //l'envoi du signal
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW);
  cm = pulseIn(EchoPin, HIGH) / 58.0; //le temps d'echo est converti en cm
  cm = (int(cm * 100.0)) / 100.0; // garder deux décimales

  //affichage de la distance
  Serial.print("Distance\t=\t");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(1000);
}
```

Résultat : après faire avancer un obstacle et le reculé on remarque que dans le moniteur série on obtient un message qui affiche la distance comme le montre la figure ci-dessous.



```
COM8
|
Distance      =      203.60cm
Distance      =      202.75cm
Distance      =      202.32cm
Distance      =      203.68cm
Distance      =      202.74cm
Distance      =      188.53cm
Distance      =      202.84cm
Distance      =      214.29cm
Distance      =      214.62cm
Distance      =      213.82cm
Distance      =      22.58cm
Distance      =      23.32cm
Distance      =      20.12cm
Distance      =      17.39cm
Distance      =      16.50cm
Distance      =      16.62cm
 Défilement automatique  Afficher l'horodatage
```

➤ Obtenir une distance précise :

```
const int TrigPin = 9;
const int EchoPin = 8;
float cm=20;
void setup()
{
  Serial.begin(9600);

  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}
void loop()
{
  //tests de distance
  while(cm>5){
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin, LOW);
    cm = pulseIn(EchoPin, HIGH) / 58.0;
    cm = (int(cm * 100.0)) / 100.0;
    Serial.print("Distance\t=\t");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
  }
}
```

Résultat : On remarque qu'à l'instant où la distance soit inférieure ou égale à 5 cm la boucle d'affichage s'arrête.

➤ Contrôle de la pompe à eau :

Pour le montage on ajout uniquement une pompe on mettons le pin de Vcc dans l'entre 13 de l'Arduino et le pin GND avec le GND du montage. Le code est le même que le précédant uniquement on ajout quelque ligne montré dans la figure ci-dessous.

```
const int TrigPin = 9;
const int EchoPin = 8;
const int pompPin =13;
float cm=20;
void setup()
{
  Serial.begin(9600);
  pinMode(pompPin, OUTPUT);
  pinMode(TrigPin, OUTPUT);
  pinMode(EchoPin, INPUT);
}
void loop(){
  digitalWrite(pompPin,LOW);
  //tests de distance
  while(cm>5){
    digitalWrite(pompPin,HIGH);
    digitalWrite(TrigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPin, LOW);
    cm = pulseIn(EchoPin, HIGH) / 58.0;
    cm = (int(cm * 100.0)) / 100.0;
    Serial.print("Distance\t=\t");
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
  }
}
```

Résultat : Lorsque la distance entre la surface de l'eau dans le Goblet (verre) et le capteur soit inferieur ou égale à 5 cm la pompe s'arrête.

➤ Utilisation de l'ESP32 :

En premier il faut ajouter la bibliothèque du ESP32 pour pouvoir l'utilisé en suite changer la numérotation des pins.

3- Le prototype :

Après les tests des matériels utilisé il faut englober tout c'est à dire rassembler tout, lorsque le temps arrive le moteur tourne avec un certain degré pour faire tomber le comprimé et au même temps l'eau coule dans le verre jusqu'à obtenir le niveau voulu. Pour une alarme on ajoute un buzzer ou quand le traitement fini on obtient un bip qui l'indique, pour l'affichage on peut ajouter un LCD pour mettre la communication facile.

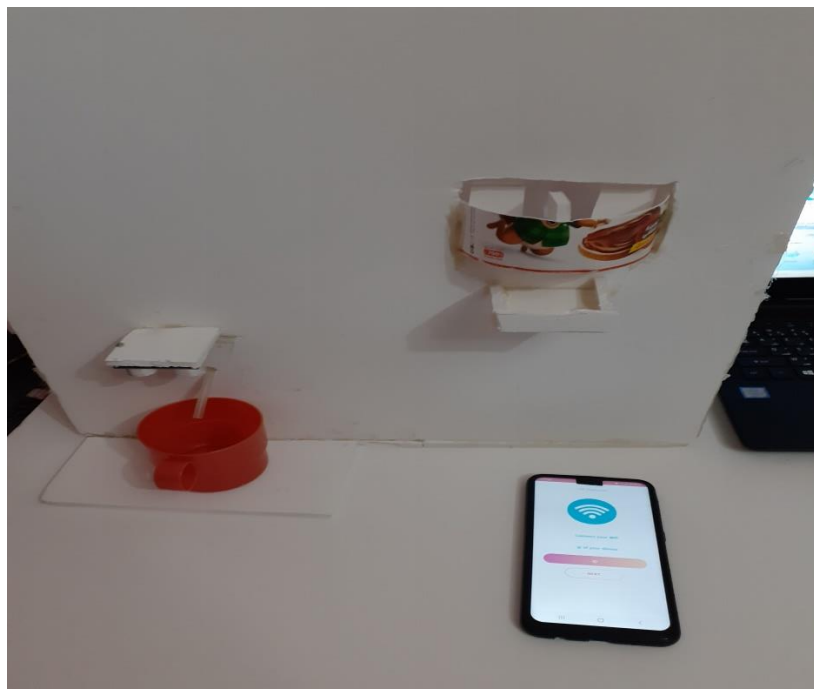


Figure III-3- Le prototype du projet.



Figure III-4- Vue du prototype à l'intérieur.

3- Application :

Cette application a été créée avec l'aide d'un informaticien, avec les étapes suivantes.

➤ **Création des designs :**

On premier il faut savoir le design de l'application et le décorer des interfaces.

➤ **Codage par XML :**

Dans cette étape on fait la création de l'application par :

- Modification du Style de l'application.

```
styles.xml x
1 <resources>
2
3 <!-- Base application theme. -->
4 <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
5     <!-- Customize your theme here. -->
6     <item name="colorPrimary">@color/rose</item>
7     <item name="colorPrimaryDark">@color/rose</item>
8     <item name="colorAccent">@color/colorAccent</item>
9 </style>
10 <style name="AppTheme.NoActionBar" >
11     <item name="windowActionBar">>false</item>
12     <item name="windowNoTitle">>true</item>
13 </style>
14
15
16
17 </resources>
```

- Création de *layout* et ajout des boutons, les *textview* et les édits textes.

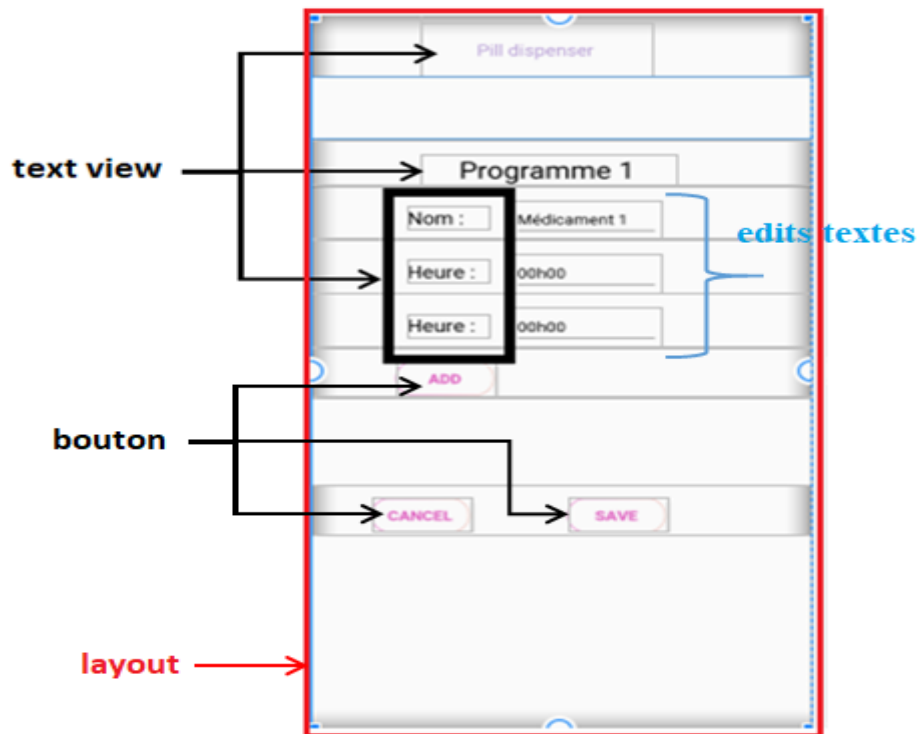


Figure III-5-Le résultats.

- Ajouter des couleurs spécialement pour l'application

```
colors.xml
1 <?xml version="1.0" encoding="utf-8" ?>
2 <resources>
3   <color name="colorPrimary">#008577</color>
4   <color name="colorPrimaryDark">#00574B</color>
5   <color name="colorAccent">#D81B60</color>
6   <color name="rose">#d773a0</color>
7   <color name="rose1">#E069C3</color>
8   <color name="rose3">#b59cd4</color>
9   <color name="rose2">#EDDBB0</color>
10  <color name="vert">#4DC8D6</color>
11
12 </resources>
```

➤ **Codage de *AndroidManifest* :**

C'est le code XML globale où on définit le nom et l'icône de l'application qui s'affiche dans le Smartphone, aussi elle fait appel aux styles quant à travailler avec avant.

```
AndroidManifest.xml
1 <?xml version="1.0" encoding="utf-8" ?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3   package="com.example.myapplication">
4
5   <application
6     android:allowBackup="true"
7     android:icon="@mipmap/logo"
8     android:label="Pill dispenser"
9     android:roundIcon="@mipmap/logo"
10    android:supportsRtl="true"
11    android:theme="@style/AppTheme">
12     <activity android:name=".Main4Activity"></activity>
13     <activity
14       android:name=".Main3Activity"
15       android:theme="@style/AppTheme.NoActionBar" />
16     <activity
17       android:name=".Main2Activity"
18       android:theme="@style/AppTheme.NoActionBar" />
19     <activity
20       android:name=".MainActivity"
21       android:theme="@style/AppTheme.NoActionBar">
22       <intent-filter>
23         <action android:name="android.intent.action.MAIN" />
24
25         <category android:name="android.intent.category.LAUNCHER" />
26       </intent-filter>
27     </activity>
28   </application>
29
30 </manifest>
```

➤ **Codage par JAVA :**

Dans cette partie on passe à l'étape dynamique c'est à dire on relie la base de donnée avec l'application, la mise on marche des boutons, création des méthodes, vérification, ect. Comme la montre l'exemple ci-dessous.

```
1 package com.example.myapplication;
2
3 import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11     private Button button;
12     @Override
13     public void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16         button=findViewById(R.id.button);
17         button.setOnClickListener(new View.OnClickListener() {
18             @Override
19             public void onClick(View view) {
20                 openActivity2();
21             }
22         });
23     }
24     public void openActivity2() {
25         Intent intent=new Intent( packageContext: this,Main2Activity.class);
26         startActivity(intent);
27     }
28 }
29 }
```

➤ **L'installation :**

On premier on branche notre Smartphone avec l'ordinateur et le configuré on commence par ce qui est montré dans la figure ci-dessous.

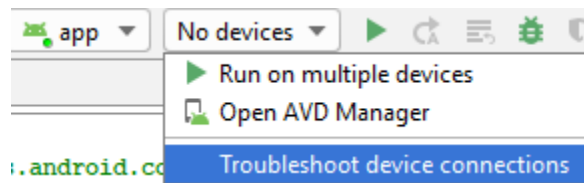


Figure III-6-Première étape de configuration.

On obtient la page suivante qui indique les informations du Smartphone utilisé.

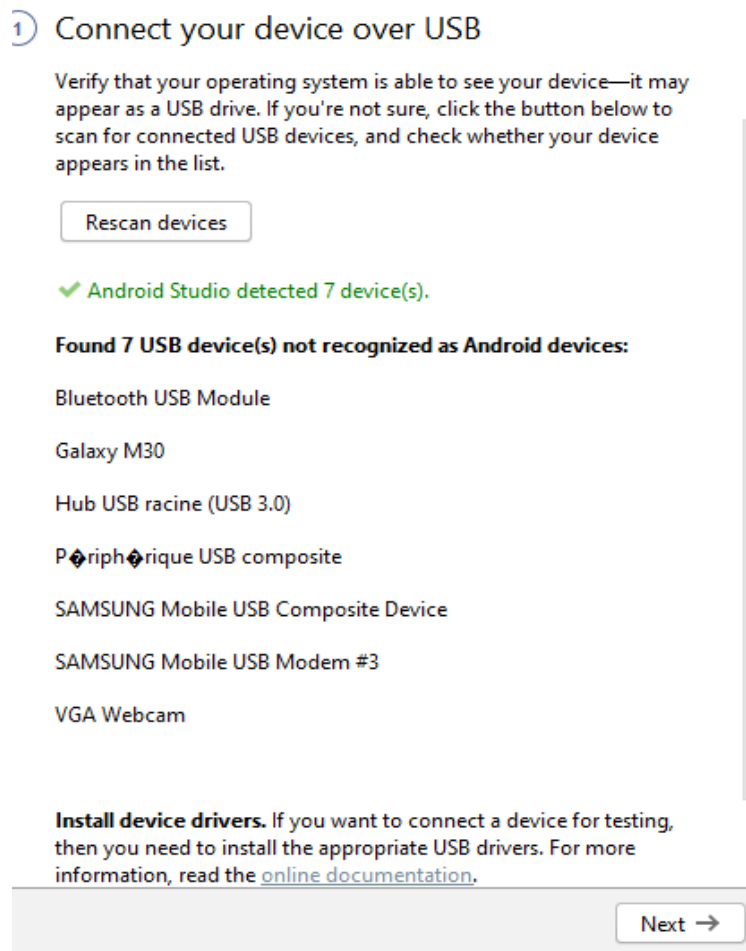


Figure III-7- image des informations du smartphone brancher.

Ensuite on clique sur *next* on obtient la page suivante

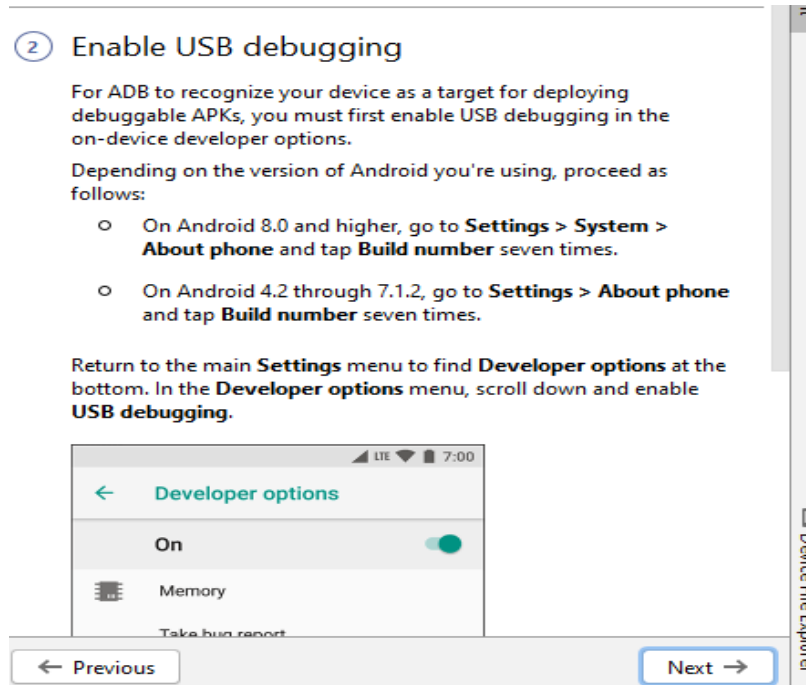


Figure III-8- Description de la méthode d'activation du mode développeur.

Cette page possède les étapes pour mettre le Smartphone on mode développeur et activé le débogage USB. Ensuite on appuis sur *next* et on obtient le résultat suivant.

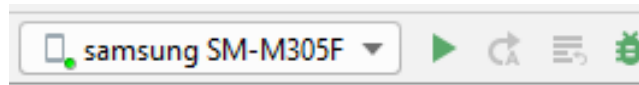


Figure III-9- Indication que le branchement et bon.

Finalement on appuis sur  pour l'installer sur le Smartphone.

➤ Le résultat :

Après l'installation on obtient l'icône de l'application au niveau du menu du Smartphone avec les autres applications comme l'indique la figure.

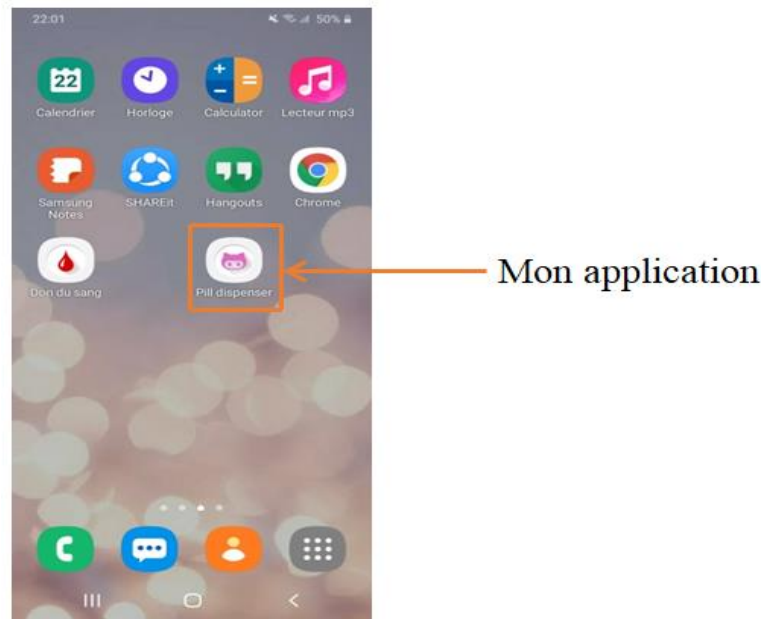


Figure III-10- L'application sur l'interface du smartphone.

En entrant dans l'application, on obtient les interfaces programmées avant.

- **Interface 1 :**

C'est l'interface qui fait connecter l'application avec une adresse IP d'un wifi, elle est représentée comme le montre la figure suivante.



Figure III-11- l'accueil de l'application.

- **Interface 2 :**

Cette interface englobe les différents boutons pour la configuration et réglage des médicaments, la figure suivante indique sa surface.



Figure III-12- L'interface 2 de l'application.

- **Interface 3 :**

C'est l'interface où on peut changer le nom et les différents moments de distribution de médicament (l'heur de prendre le médicament qui se trouve dans la notice). Comme a figure suivante le clarifier.



Figure III-13-L'interface 3 de l'application.

4- Lier l'application avec Arduino :

Comme c'est indiqué avant la liaison est faite par un module wifi dans le but de pouvoir faire une communication entre l'application et le ESP.

Donc faire communiquer l'appareil Android avec l'Arduino, on utilise un langage commun c'est à dire le protocole MQTT. L'avantage de ce protocole, c'est qu'il existe une librairie pour Android et une librairie pour Arduino. Ainsi nos deux mondes communiqueront de la même façon.

Le protocole MQTT est un protocole de messagerie de type *Publish-Subscribe* basé sur le protocole TCP/IP (d'où la nécessité du *shield Ethernet*). Le protocole se décompose donc en trois grandes parties :

- Les « *publishers* » : ils envoient un ou des message(s) sur un ou plusieurs « Topic »
- Le « *broker* » MQTT : il fait le lien entre les « *publishers* » et les « *subscribers* »
- Les « *subscribers* » : ils s'abonnent à un ou plusieurs « Topic ». Lorsque qu'un message est publié sur un topic, tous les *subscribers* de ce topic reçoivent le message.

Le broker MQTT permet de faire le lien entre le *publisher* (appareil Android) et le *subscriber* (Arduino) et permet de faire communiquer l'appareil Android avec l'Arduino. Donc c'est le cœur de notre architecture de la figure ci-dessous. [8]

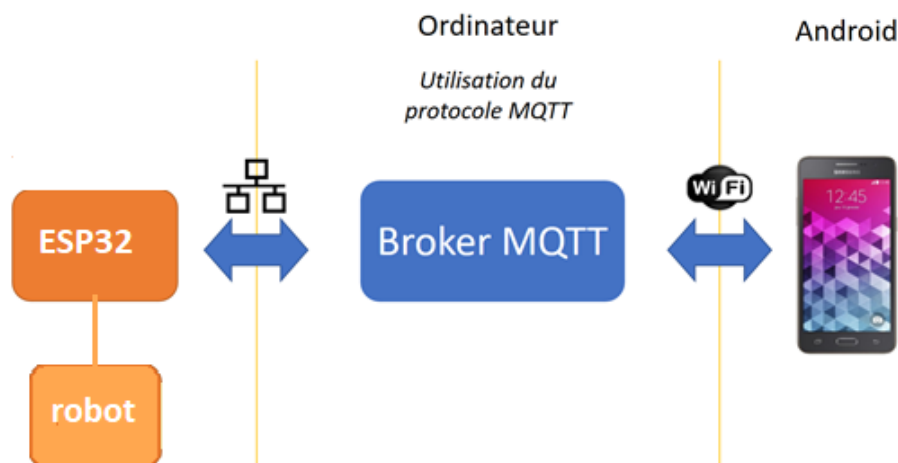


Figure III-14- L'architecture de la communication entre l'application et esp32.

Conclusion ;

Dans ce chapitre on a englobé le travail mais vu que la situation dans laquelle on vit on a trouvé beaucoup de difficulté là où on n'a pas pu faire la réalisation complète du projet et obtenir un bon fonctionnement du projet.

**Conclusion
générale**

Conclusion générale

L'objectif de ce projet est de réaliser un prototype d'un système d'assistance et de préparation médicale à domicile. En raison de l'état dans lequel on vit avec le virus covid-19, on a pu réaliser certaines parties de ce projet mais on a trouvé plusieurs obstacles dans la partie pratique à cause du confinement.

Avec ce système, on peut aider des personnes qui ont des maladies chroniques (comme les diabétiques, les hypertendues, ...) à prendre leur médicament correctement et à l'heure précisée par le médecin pour protéger leur vie.

La réalisation de ce projet peut donner beaucoup d'aide en ce moment, en le déposant dans les hôpitaux où se trouve des patients du covid-19 ce qui leur permet de prendre les médicaments sans l'intervention des infirmiers et de réduire le contacte entre eux. Mais cela nécessite à apporter une solution pour le déplacement vers la machine (le distributeur).

Dans ce projet on a considéré le cas des personnes pouvant se déplacer, concernant les personnes qui souffrent d'un handicap physique (ils ne peuvent pas utiliser la machine seul) il y a d'autre solution par exemple ajouter un bras robot qui lui donne le médicament.

Bibliographie

Bibliographie

- [1] https://fr.wikipedia.org/wiki/Distribution_automatique
- [2] <https://www.guide-distributeur-boissons.be/distributeur-de-boissons/les-differents-types-de-distributeur-automatique/>
- [3] <https://create.arduino.cc/projecthub/audalogy/smart-auto-dispensing-pillbox-b1c089>
- [4] <https://harmonicdrive.de/fr/glossaire/moteur-pas-a-pas>
- [5] https://fr.wikipedia.org/wiki/Moteur_pas_%C3%A0_pas#Application
- [6] <https://fr.wikipedia.org/wiki/ESP32>
- [7] <https://www.microsonic.de/fr/support/capteurs-%C3%A0-ultrasons/principe.htm#:~:text=Le%20principe%20des%20ultrasons%3A,forme%20d'%C3%A9cho%20au%20capteur.>
- [8] <https://zestedesavoir.com/tutoriels/2069/faites-communiquer-votre-arduino-avec-votre-appareil-android/>

Les références pour téléchargement des applications :

- [9] <https://www.arduino.cc/en/Main/Software>
- [10] <https://developer.android.com/studio/index.html>
- [11] <https://www.youtube.com/watch?v=M6pi6jXpRrs&t=395s>