



**REPUBLIQUE ALGERIENNE DEMOCRATIQUE
ET POPULAIRE**



**UNIVERSITÉ ABDELHAMID IBN
BADIS_MOSTAGANEM
FACULTÉ DES SCIENCES ET DE TECHNOLOGIE**

DÉPARTEMENT D'ÉLECTRONIQUE

MÉMOIRE DE FIN D'ÉTUDES
Pour l'obtention du diplôme de **Master en**
SYSTÈME EMBARQUÉ

Présenté par

**TSRIAT TAHANI NOR EL HOUDA
BOUZIANE ZAKARIA**

**Spécialité : SYSTÈME
EMBARQUÉ**

Thème :

**CONTRÔLE À DISTANCE
D'UN MICROSCOPE
POUR LES TRAVAUX
PRATIQUES**

ENCADRANT : BANECHENHOU ABDALHALIM

DEVANT LE JURY

Président :

- Mr. BENYAMINA

LES EXAMINATEURS :

- Mme. ABAD

- Mr. BENSTALLI

ANNÉE UNIVERSITAIRE : 2021/2022

DÉDICACE

Je dédie ce modeste travail à :

Celle qui a consacré sa vie et souffert

Pour veiller à mon bien être, à la source de ma

Réussite, à ma chère mère. Que Dieu la garde.

*Mon père, qui m'a donné l'espoir et le courage nécessaire pour
l'accomplissement de ce travail, et qui m'a encouragée dans mes études.*

Je souhaite qu'ils soient fièrs de moi.

*Mon frère et Mes sœurs, Lokman, Amina, Meriem pour leur soutien moral
et leurs sacrifices tout au long de ma formation.*

*Ma binôme Tahani Nour el houda pour son soutien moral, sa patience et
sa compréhension tout au long de ce mémoire et à toute sa famille.*

Mes amis (es) proches.

ZAKARIA

DÉDICACE

En témoignage d'amour et d'affection, je dédie ce travail avec une grande fierté à :

Mes parents qui ont été d'un dévouement exemplaire et d'un réconfort inestimable. Ma mère, source de tendresse et d'amour pour son soutien tout le long de ma vie scolaire. Que dieu te protège Mon père, qui m'a toujours soutenu et qui a fait tout possible pour m'aider.

Que dieu te garde Mes chères sœurs Amani et Hanadi pour leur soutien moral et leurs sacrifices tout au long de ma formation et toute ma famille en reconnaissance de leur encouragement.

Mon binôme Zakaria, Merci pour la confiance que tu m'as accordé et pour ton accord de réaliser ce mémoire ensemble. Et à toute sa famille Mes chers (es) amis (es) avec qui j'ai partagé les meilleurs et les moments les plus agréables tout au long de mon parcours universitaire.

Tous mes professeurs pour leurs générosités et la grande patience dont ils ont su faire preuve. Veillez, tous, accepter mes hautes salutations et considérations.

Que Dieu puisse vous protéger

Tahani

Remerciements

Et voilà, la fin d'un long périple que nous poncturons volontairement par la rédaction des remerciements. Il est toujours difficile de terminer une étape, partagé entre la joie d'apercevoir le point d'arrivée, promoteur de nouveaux horizons, de nouvelle expérience et le regret de laisser derrière soi, un parcours souvent éprouvant autant physiquement que moralement, mais tellement riche de sensations. Dieu merci d'avoir fait de mon parcours de thèse un parcours réussit.

Ce travail a été effectué aux laboratoires d'Électromagnétisme et Optique Guidée (LEOG), université Abdelhamid Ibn Badis de Mostaganem.

Beaucoup de reconnaissance et de gratitude à notre enseignant et promoteur le Professeur Benachenhou Abdelhalim, pour ses qualités humaines, sa précieuse orientation et ses conseils pour accomplir ce travail. Merci pour tout, avec notre plus profond respect.

On exprime notre profonde gratitude à Mr Benyammina Et Mr Benstalli et Mme abad pour l'honneur qu'ils nous ont fait en acceptant de présider nos jurys de mémoires.

Nos vifs remerciements s'adressent au Mr Azzedine Mohamed ainsi qu'envers tous nos enseignants pour le savoir transmis durant nos cinq ans d'études, leurs soutiens, leurs esprits communicatifs performants ainsi que pour les fructueuses discussions scientifiques que nous avons eues tout au long de ce travail.

J'exprime également ma reconnaissance à Mr Boumehdi Abderrahmane et Mr Mebrouka Abdelhamid pour leurs disponibilités à tout moment au laboratoire et pour leurs contributions au bon déroulement de ce mémoire.

Enfin nous tenons à remercier très chaleureusement nos familles en particulier nos parents pour leur amour et leur soutien constant qui ont contribué à la réussite de ce travail, aussi bien dans les bons moments que dans ceux de doutes et d'impatience. C'est à eux que nous dédions ce modeste travail.

Table des matières

Table des matières :

Introduction Générale

1. Introduction :	8
2. Le microscope :	9
3. Les applications du microscope USB :	11
4. Les travaux pratiques distants en biologie :	12

Chapitre1 : Architecture matérielle

1. Introduction :	14
2. Le schéma synoptique :	14
3. La partie hardware :	15
3.1 le châssis :	25
3.2 Raspberry Pi :	17
3.2.1 Les types de base de Raspberry Pi :	18
3.2.2 Raspberry Pi 4:	18
3.2.3 PINS de Raspberry Pi :	19
3.3 Le moteur pas à pas :	20
3.3.1 Le choix de moteur pas à pas :	21
3.4 DRV_8825 :	23
3.4.1 Brochage et description des pins de DRV_8825 :	24
3.5 Le capteur de fin de course :	25
3.6 Le microscope USB :	26
4. Conclusion :	27

Chapitre2 : Architecture logicielle

1. Introduction :	29
2. La partie software :	29
2.1 Backend :	29
2.1.1 Python :	29
A. Les versions python :	29
2.1.2 Node.js :	30
B. Npm :	30
2.1.3 Child process :	30
2.2 Frontend :	31
2.2.1 HTML :	31
2.2.2 CSS :	31
2.2.3 Java Script :	32

Table des matières

2.3	Application client-serveur :	32
3.	Conclusion :	33

Chapitre3 : Mise en œuvre

1.	Introduction :	35
2.	Hardware :	35
2.1	Les composants utilisés :	35
2.2	Le schéma de câblage :	35
2.3	EasyEDA :	38
2.3.1	Schéma de branchement :	38
2.3.2	PCB :	39
3.	Software :	39
3.1	Architecture logicielle:	39
3.2	Dans la page web :	40
3.2.1	L'interface :	41
3.3	Dans le serveur d'application :	42
3.3.1	LiveCam :	42
3.3.2	Programme de python :	42
3.3.2.1	L'organigramme :	42
4.	Conclusion :	45
Conclusion générale :		47

Annexe

1.	PuTTY :	53
2.	Serveur d'écho :	53
3.	Commandes SSH de base :	53
4.	Programme python :	54

Introduction générale

Introduction générale

1. Introduction :

La pandémie de COVID-19 a conduit l'enseignement supérieur vers l'enseignement en ligne. C'est quoi un enseignement en ligne ?

Un enseignement en ligne est défini comme étant un processus visant à concevoir, à construire et à dispenser un enseignement lorsque la source du savoir et les apprenants sont séparés par le temps ou la distance, ou les deux. Une définition encore plus simple l'enseignement à distance est l'apprentissage structuré qui se déroule sans la présence physique de l'enseignant, il peut être simplement décrit comme un apprentissage accessible électroniquement à partir d'un dispositif informatique (généralement connecté à Internet). Cela permet à l'apprenant de s'engager dans l'apprentissage à tout moment et depuis n'importe quel endroit. Quelques conditions de base sont toutefois requises : une connexion Internet, un ordinateur et l'accès à un fournisseur d'apprentissage en ligne sur le web [1]. L'apprentissage en ligne (également appelé e-learning, formation en ligne ou enseignement basé sur le Web) fait référence à l'enseignement dispensé via Internet, où l'instructeur et l'apprenant sont éloignés l'un de l'autre. Il existe deux types d'apprentissage en ligne :

- Asynchrone (forum, courrier électronique, remise de devoir...).
- Synchrone (diffusion en continu de vidéos et d'audio) [2].

Au début de la pandémie la plupart des cours de l'enseignement supérieur ont trouvé la transition vers l'enseignement en ligne difficile. De nombreux professeurs ont transféré leurs classes en ligne en une ou deux semaines seulement. Ils ont intentionnellement appris eux-mêmes comment enseigner en ligne, ils ont appris à générer des ressources pédagogiques, à organiser des classes en ligne et à mener des sessions en direct [3]. Après les premiers bouleversements, les universités ont commencé à fournir un soutien plus méthodique. Des équipes de soutien à l'apprentissage en ligne ont été créées, des programmes de formation à l'apprentissage en ligne ont été proposés et des outils et ressources d'apprentissage en ligne ont été mis à disposition. La confiance des enseignants dans l'enseignement en ligne s'est rapidement développée, tout comme leur opinion sur l'apprentissage en ligne [1].

Quelle est la situation actuelle de l'enseignement à distance dans l'enseignement des sciences et des ingénieurs ? L'enseignement à distance a explosé en popularité au cours de la dernière décennie. Le développement de l'apprentissage à distance a été l'une des dix réalisations les plus importantes de l'enseignement des ingénieurs au cours du siècle dernier. Il est difficile de ne pas être d'accord, compte tenu de la croissance spectaculaire actuelle et des excellents résultats d'apprentissage obtenus grâce à l'enseignement et à la formation en ligne [4]. Cependant, les cours qui nécessitent un apprentissage

Introduction générale

pratique, comme les matières scientifiques, technologie ingénierie et mathématiques, la médecine, la biologie, la chimie...etc sont particulièrement difficile à porter à distance. Beaucoup croyaient que c'était impossible car L'apprentissage est un processus qui exige la participation. Écouter ne nous apprend rien, mais "faire des choses" nous apprend beaucoup. Les niveaux supérieurs de réflexion (par exemple, la synthèse et l'analyse) sont absents de l'apprentissage en ligne parce que l'étudiant ne fait qu'écouter pour cela de nombreux enseignants ont commenté l'importance des travaux pratiques dans le processus d'apprentissage [1]. L'expérimentation avec des équipements réels dans les domaines de la science et de l'ingénierie aide la construction des connaissances et fourni aux étudiants l'expertise nécessaire pour s'attaquer aux problèmes du monde réel. Un laboratoire est communément reconnu comme étant composé d'une pièce contenant des équipements spécialisés sur lesquels des expériences sont menées, et après lesquelles les résultats sont enregistrés. Les compétences typiques acquises dans les laboratoires vont de l'observation à la manipulation et à l'interprétation. D'autres attributs positifs comprennent un intérêt et un plaisir accrus pour le sujet et une conversion de la théorie en réalité [3].

Aujourd'hui avec une planification minutieuse, un personnel dévoué, des étudiants enthousiastes et des approches pédagogiques créatives il est possible de faire des travaux pratiques à la maison. Les laboratoires distants sont une sorte d'environnement de simulation pédagogique qui permet aux étudiants de participer à des expériences de laboratoire en ligne et d'étudier des concepts et des théories sans avoir à se rendre dans un laboratoire physique [1].

Avec les laboratoires en ligne ou Les laboratoires distants nous pouvons utiliser des équipements réels situés dans une autre région [2]. L'absence d'équipement réel à proximité de l'apprenant et la rapidité de réaction sont deux obstacles potentiels. Ce projet est une tentative d'apporter une solution pour les travaux pratiques nécessitant une observation au microscope. Au niveau de base, les étudiants peuvent recevoir des images numérisées à partir d'un microscope USB. Cette approche est simple à déployer et à mettre en œuvre mais est limitée en termes de fonctionnalités. C'est pour quoi une fonctionnalité de télécommande pourrait être intégrée à un microscope. Lorsqu'il est équipé d'une platine électronique et de commandes de mise au point, un utilisateur peut contrôler le microscope depuis un emplacement hors laboratoire.

2. Le microscope :

Un microscope est un appareil qui grossit des éléments qui seraient autrement trop petits pour être vus, ce qui donne une image qui semble plus grande. La majorité des photographies de cellules sont obtenues à l'aide d'un microscope, et ces images sont appelées micrographies [5].

Introduction générale

D'après la définition ci-dessus, un microscope peut sembler n'être rien d'autre qu'une loupe. En réalité, les loupes sont des microscopes ; elles sont appelées microscopes simples car elles ne possèdent qu'une seule lentille. Les microscopes composés, qui contiennent plusieurs lentilles, sont les outils les plus avancés que nous considérons comme des microscopes. Ces lentilles peuvent courber la lumière pour produire une image considérablement plus agrandie qu'une loupe, en raison de la façon dont elles sont disposées [5].

Le but du microscope est de donner accès à la structure microscopique des objets observés. Un microscope sert donc à voir des détails plus fins de l'objet et pas seulement à en faire une image agrandie [6]. Le paramètre principal de cet instrument est donc sa résolution, c'est-à-dire la distance la plus faible à laquelle deux points peuvent être séparés tout en étant reconnus comme des objets indépendants. Plus cette valeur est faible, plus le pouvoir de résolution du microscope est élevé et plus l'image est claire et détaillée, le deuxième paramètre est son grossissement qui désigne la taille d'un objet qui apparaît plus grand lorsqu'il est observé à l'aide d'un microscope (ou d'une paire de lentilles dans un microscope). Les microscopes optiques (MO), par exemple, couramment utilisés dans les lycées et les universités, grossissent les objets jusqu'à 400 fois leur taille réelle. Un objet qui fait 1 mm de large dans la réalité, fera 400 mm de large dans la vue du microscope [5].



Figure1 : Microscope optique [1]

Introduction générale

Le type de microscope le plus connu est le microscope optique, est un type de microscope qui agrandit les images de petits matériaux en faisant passer de la lumière visible à travers le spécimen (l'échantillon biologique) et en la courbant via le système de lentilles [6].

Les microscopes dans les laboratoires universitaires ont tendance à être des microscopes à fond clair, ce qui signifie que la lumière visible traverse l'échantillon et est utilisée pour générer une image sans aucune altération. La microscopie optique, qui est un peu plus avancée, utilise des méthodes optiques pour augmenter le contraste, ce qui permet de visualiser plus facilement les caractéristiques des cellules et des tissus [7].

Un autre type de microscope est le microscope électronique (ME). Son fonctionnement ressemble un peu à celui d'un microscope optique sauf que :

- Les photons sont remplacés par des électrons.
- Les lentilles de verre sont remplacées par des lentilles électromagnétiques.
- La limite de résolution du microscope électronique est plus élevée à celle du microscope optique. Elle est de 2 nm (c'est-à-dire 1000 fois plus élevé que le MO). L'agrandissement du ME peut atteindre jusqu'à 500.000 contre 2000 pour un MO [8].

Le type que nous avons utilisé est le microscope USB. Un microscope informatique ou un microscope connecté à un ordinateur est un autre nom pour un microscope USB. Il s'agit d'un microscope qui se branche sur le port USB d'un ordinateur ou d'un téléviseur et qui utilise des capteurs CMOS [9].

Au lieu de regarder à travers un oculaire, l'observateur voit le spécimen sur un écran d'ordinateur ou de télévision, l'objectif du microscope USB peut être utilisé pour agrandir un objet ou pour voir des objets à une courte distance. La plupart coûtent moins de 200 \$ et offrent un grossissement de 20 à 300 fois [9].

3. Les applications du microscope USB :

Le microscope USB peut être utilisé dans différents domaines, notamment :

- Observation microbiologique
- Inspection industrielle :PCB, Machines de précision
- Observation des plantes
- Examen de la peau
- Observation des cellules

Introduction générale

- Observation des pièces mécaniques

Bien que les exemples d'utilisation de cette solution se rapportent à la biologie, le produit peut être utilisé dans tout travail nécessitant l'observation au microscope avec un grossissement de 50 à 500.

4. Les travaux pratiques distants en biologie :

L'un des problèmes de la pandémie de COVID-19 est la fermeture indéfinie des établissements d'enseignement dans les pays touchés afin de minimiser la propagation de la maladie [1]. Pendant ce temps, l'enseignement en ligne est devenu de plus en plus utilisé dans de nombreuses structures éducatives. Mais ce passage a vécu plusieurs obstacles, un de ces obstacles est les travaux pratiques ou les cours en laboratoires [3].

Les travaux pratiques dans l'enseignement sont une forme d'activité d'apprentissage qui se fait dans un laboratoire et qui permet à un groupe d'étudiants d'expérimenter les principes théoriques appris en suivant les consignes de l'enseignant [10].

Les cours en laboratoire sont l'un des espaces les plus compliqués et les plus difficiles à recréer ou à enseigner par le biais de l'apprentissage à distance car ils nécessitent souvent des équipements et des fournitures spécifiques surtout les matières scientifiques comme la biologie [3].

L'objectif de notre initiative est de donner aux étudiants un accès à la microscopie à distance, ce qui leur permet de contrôler le microscope USB et de réaliser leurs travaux sans se déplacer au laboratoire.

La solution développée dans ce projet nécessite la réalisation matérielle du microscope avec une platine de trois degrés de déplacement, le développement logiciel du contrôle à distance via internet.

Ce mémoire est donc ainsi structuré :

Le premier chapitre décrit l'architecture matérielle et les différents composants utilisés. Le second chapitre décrit l'architecture logicielle avec une description des langages de programmation utilisés.

Le dernier chapitre est pour la mise en œuvre.

Chapitre 1 :
Architecture
matérielle

Chapitre 1 : Architecture matérielle

1. Introduction :

Dans ce chapitre, nous présentons le schéma synoptique et la partie hardware de projet. Dans la partie hardware nous présenterons une description des composants utilisés.

Ces éléments constitutifs sont :

- Raspberry pi 4.
- Les moteurs pas à pas.
- Le driver du moteur pas à pas : DRV8825.
- Capteur fin de course.
- Microscope USB.

2. Le schéma synoptique :

Le système est composé d'un microscope USB, d'une platine permettant des déplacements X, Y, Z et d'un contrôleur Raspberry Pi 4 connecté au Réseau local (LAN).

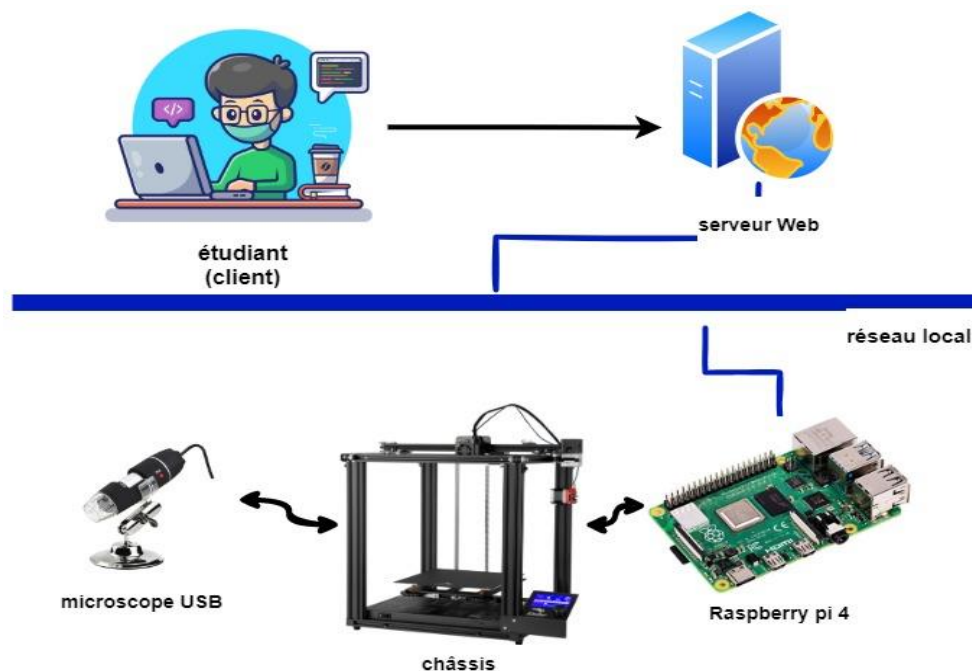


Figure2 : schéma synoptique

Le châssis est composé de trois moteurs, chacun contrôlant un axe distinct, les axes X et Y contrôlant le support des spécimens et l'axe Z contrôle la hauteur du microscope USB pour la mise au point. L'étudiant inscrit doit avoir accès à une interface Web où il peut utiliser des boutons pour contrôler les axes de translation. L'interface Web est composée de neuf boutons au total :

Chapitre 1 : Architecture matérielle

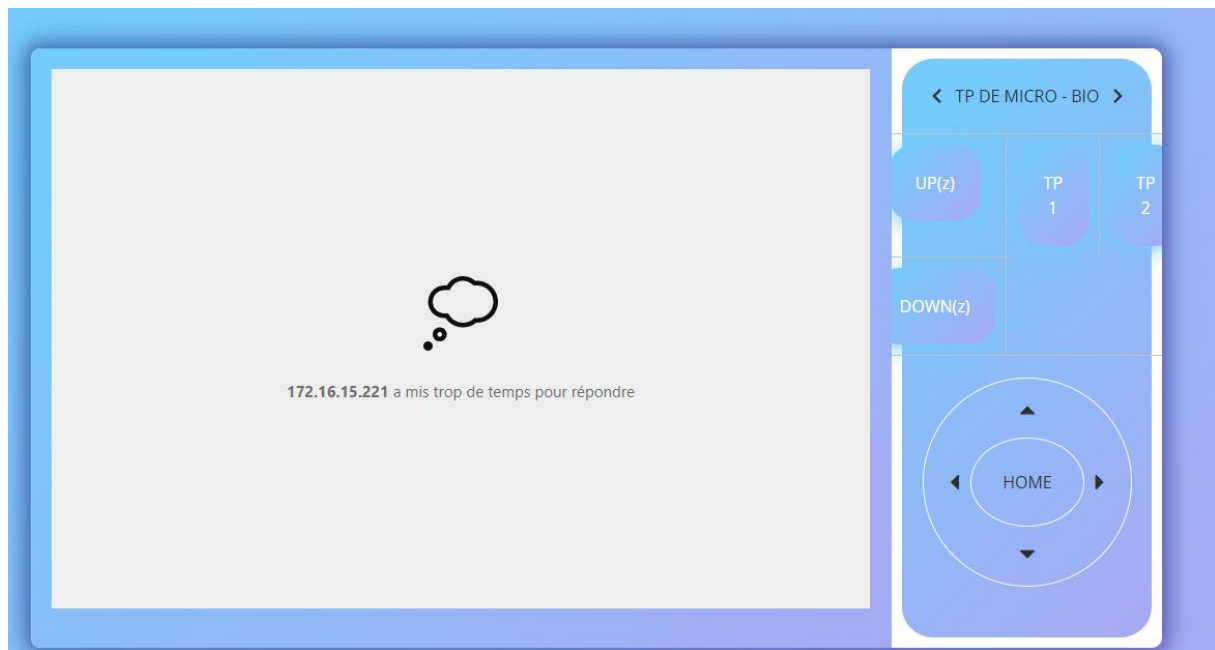


Figure 3: l'interface

- Le bouton Zone 1, qui ramène le support au milieu de l'emplacement de spécimen 1.
- Le bouton Zone 2, qui ramène le support au milieu de l'emplacement de spécimen 2.
- L'axe X comporte deux boutons permettant de contrôler le support de spécimen à gauche (X-) et à droite (X+).
- L'axe Y comporte deux boutons aussi permettant de contrôler le support de spécimen en bas (Y-) et en haut (Y+).
- La montée (Z-) et la descente (Z+) du microscope USB sont contrôlées par deux boutons situés sur l'axe Z ils se présentent le focus du microscope.
- Le bouton HOME, qui ramène le support de spécimen à son emplacement initial, est le dernier bouton. L'action d'appuyer sur ce bouton agit sur le moteur X qui commence à tourner vers la gauche jusqu'à ce que le capteur fin de course le détecte, puis il s'arrête et le moteur Y commence à tourner vers la gauche jusqu'à ce que son capteur fin de course le détecte aussi. En ce moment Le microscope USB commence alors à se déplacer jusqu'à ce qu'il atteigne la fin de sa trajectoire. Le site web est composé d'une fenêtre de caméra en temps réel permet à l'apprenant de voir tous les détails qu'il désire.

3. La partie hardware :

Le hardware désigne la partie physique du dispositif. Ce terme permet de distinguer le montage du circuit électronique et ses composants [11]. Un montage électronique est souvent composé de

Chapitre 1 : Architecture matérielle

l'assemblage de plusieurs composants, connectés les uns aux autres de manière très précise, pour permettre d'exécuter la tâche désirée [12].

3.1 Le châssis:

Dans la partie mécanique nous avons réutilisé le châssis d'une imprimante 3D qui était au Laboratoire Électromagnétisme et Optique Guidée (LEOG) de Mostaganem où nous avons placé notre circuit, ce châssis est composé de :

- Vis et écrou à billes : La vis sans-écrou est une tige filetée qui permet à l'aide d'un écrou de transformer la vitesse de rotation de moteur en un mouvement de translation et assure des déplacements de grande précision.
- L'accouplement : est un dispositif de liaison entre deux arbres rotatifs, permettant la transmission du couple.
- Rail de guidage V-slot : C'est un profil de rail linéaire de haute qualité en aluminium avec une rainure en V. extrêmement lisse sur les 4 côtés. Elle permet de guider le mouvement de translation des roulements selon les trois axes de la machine.
- Support des axes de guidage : Se relie aux rails pour fixer les axes.
- L'axe de guidage : Permet de guider le mouvement de translation des roulements selon les deux axes de la machine.
- Les roulements : Les roulements sont des organes mécaniques servant à guider un arbre de transmission en rotation [13].

La figure 4 présente le châssis utilisé et réadapté pour utilisation en tant que microscope avec une platine porte échantillon trois degrés de liberté.

Chapitre 1 : Architecture matérielle

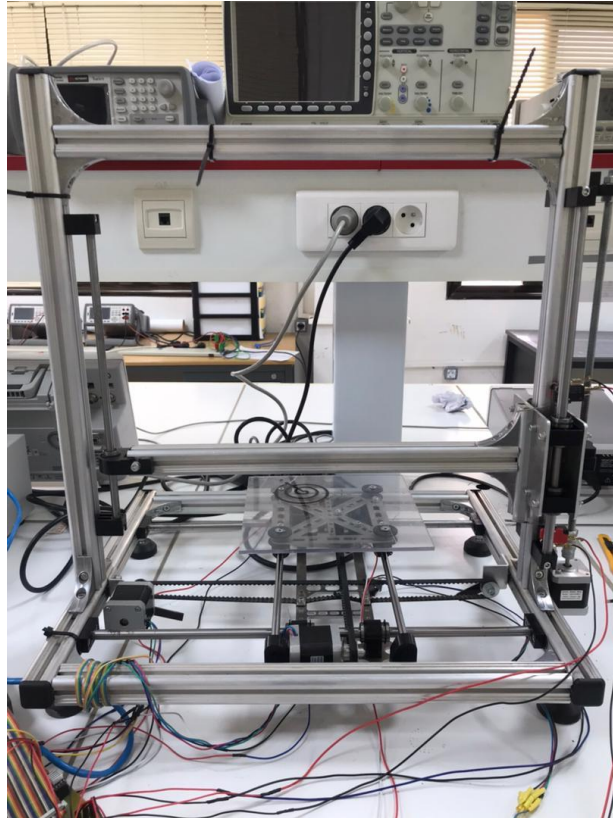


Figure 4: Châssis

3.2 Raspberry Pi:

Le Raspberry Pi est un appareil remarquable : un ordinateur entièrement fonctionnel dans un boîtier minuscule et peu coûteux [14]. Contrairement à un ordinateur traditionnel, qui cache son fonctionnement interne dans un boîtier, un Raspberry Pi a tout ce qu'il faut pour être efficace (ses composants, ses ports et ses fonctions) [15].

Le Raspberry Pi est connu comme un ordinateur mono-carte (single board computer), est un nano-computeur basé sur une carte de circuit imprimé unique. Le Raspberry Pi, comme la plupart des ordinateurs mono-carte est petit mais cela ne veut pas dire qu'il n'est pas puissant. Un Raspberry Pi peut accomplir tout ce qu'un ordinateur plus gros et plus énergivore peut faire, mais pas nécessairement aussi vite [16].

À son cœur se trouve une puce Broad-com BCM283X (X :5,6,7...) qui contient un processeur central ARM, une unité de traitement (CPU) et une unité de traitement graphique (GPU) de vidéo-Core 4 partagent la mémoire entre elles. Le GPU est suffisamment puissant pour pouvoir gérer une lecture vidéo de qualité Blu-ray [17].

Chapitre 1 : Architecture matérielle

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 l'OS Microsoft Windows : Windows 10 IoT Core et celui de Google, Android Pi [18].

3.2.1 Les types de base de Raspberry Pi :

Il existe différents modèles de Raspberry Pi, chacun d'entre eux apportant soit des spécifications améliorées, soit des caractéristiques spécifiques à un cas d'utilisation particulier. Depuis sa sortie, Raspberry Pi est proposé dans différentes versions, avec des mises à jour petites et grandes et des améliorations apportées à l'unité Raspberry Pi Model B d'origine, il existe trois types de base de Raspberry Pi :

- ❖ Raspberry Pi model B.
- ❖ Raspberry Pi modèle A.
- ❖ Raspberry Pi model Zéro.

Tous les modèles Raspberry Pi ont des propriétés communes : ils sont compatibles, ce qui signifie que les logiciels écrits pour un modèle fonctionneront sur n'importe quel autre modèle. Dans notre projet, nous avons utilisé la **carte Raspberry Pi4 model B** du laboratoire de génie électrique de l'université de Mostaganem, sous système d'exploitation Raspbian qu'on a chargé et installé en utilisant une carte mémoire. L'affichage de l'interface peut se faire sur un écran tactile compatible [19].

3.2.2 Raspberry Pi 4:

Le Raspberry Pi 4 Modèle B illustré par la figure 5 est le plus récent ajout à la célèbre gamme d'ordinateurs Raspberry Pi. Comparé à la génération précédente Raspberry Pi 3 modèle B+, il offre des améliorations révolutionnaires en termes de vitesse de traitement, de performances multimédias, de mémoire et de connexion, une rétrocompatibilité et une consommation d'énergie. Les principales caractéristiques de ce modèle sont :

- ❖ **Processeur** : Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) SoC 64 bits @ 1,5 GHz
- ❖ **Mémoire** : 1GB, 2GB ou 4GB LPDDR4(selon le modèle)
- ❖ **Connectivité** : 2,4 GHz et 5,0 GHz

LAN, Bluetooth 5. 0, BLE

Gigabit Ethernet

2× ports USB 3. 0

Chapitre 1 : Architecture matérielle

2 × ports USB 2.0

- ❖ **GPIO** : GPIO standard 40 broches
- ❖ **Carte SD** : fente pour carte Micro SD pour le système d'exploitation de chargement et stockage des données

Raspberry Pi 4 modèle B restera en production au moins jusqu'en janvier 2026 [20].

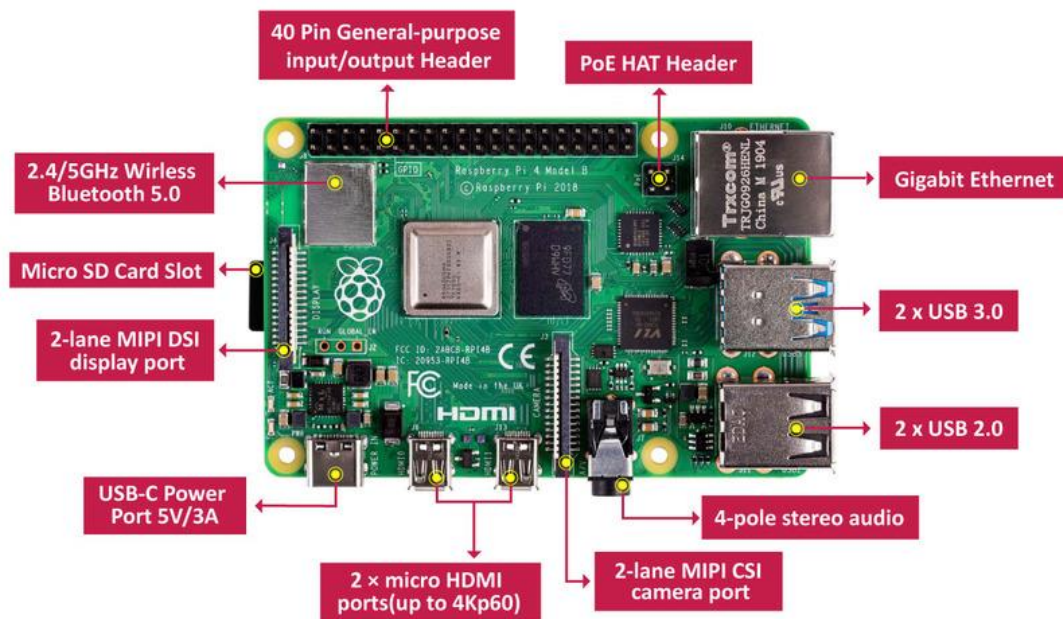


Figure 5: Raspberry Pi 4 [2]

3.2.3 PINS de Raspberry Pi :

GPIO est l'abréviation de General Purpose Input/Output. Ce sont des broches qui peuvent être configurées pour exécuter différentes fonctions. Comme leur nom l'indique, elles sont donc destinées à un usage large plutôt qu'à une application spécifique [21].

Ce sera l'utilisateur au moment de l'exécution qui pourra configurer ces broches GPIO pour qu'ils fassent ce qu'il veut. Cela peut être fait de différentes manières, par exemple avec certains codes ou scripts de la console ou avec le programme Python, qui est l'un des moyens les plus simples et les plus préférés en raison de la quantité d'options disponibles [21].

Il est nécessaire de câbler correctement les périphériques sur ce connecteur. Le connecteur GPIO supporte les **GPIO** (entrées/sorties binaires) mais également les sorties **PWM**, les périphériques de

Chapitre 1 : Architecture matérielle

communication (UART, I2C, SPI) et les alimentations 5V et 3V3. Les broches peuvent avoir des fonctions différentes suivant qu'elles sont activées en tant que GPIO ou périphérique de communication. Certaines possèdent des résistances de pull-up donnant un bit à 0 dominant et un bit à 1 récessif [22]. La figure 6 présente la disposition des différents GPIO sur le connecteur d'extension ainsi que leurs fonctions.

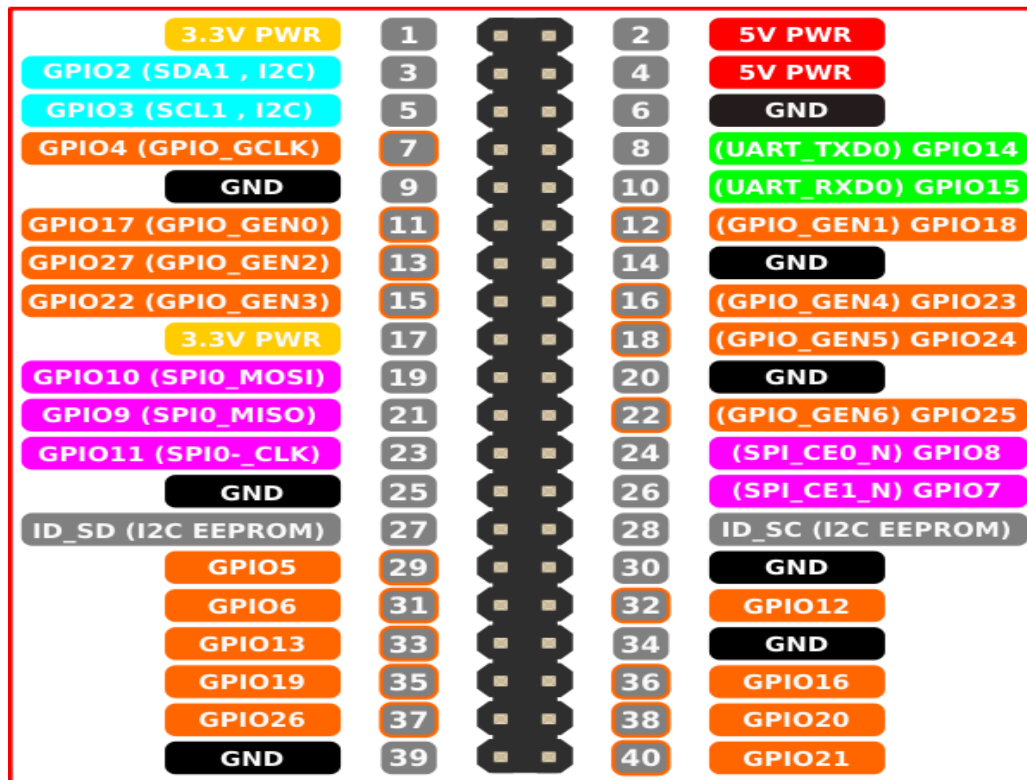


Figure 6 : pins de Raspberry Pi 4 [3]

3.3 Le moteur pas à pas:

Un moteur pas à pas est un système électromécanique qui convertit des signaux électriques en mouvement mécanique. Il est destiné à effectuer un mouvement discret (concept de pas) et à atteindre une position précise. Le mouvement est accompli par des bobines produisant un champ magnétique qui est détecté par des aimants. Lorsque l'une des bobines est alimentée, un champ magnétique se forme. Le champ magnétique fluctue si l'énergie est fournie de manière cyclique (par des impulsions d'entrée). Lorsqu'un aimant est placé dans un champ magnétique fluctuant, il recherche l'état d'énergie le plus bas (équilibre) et fournit ainsi un mouvement [23].

Les moteurs pas à pas sont largement utilisés dans les applications médicales, satellitaires, robotiques et de commande en raison de leur commande précise. Tous les moteurs pas à pas présentent certaines caractéristiques qui les rendent appropriés pour ces applications [24]. Un moteur pas à pas possède quelques composants de base. Tout d'abord, nous avons un stator en fer doux. Chaque stator est

Chapitre 1 : Architecture matérielle

entouré de plusieurs phases qui sont mises sous tension à l'aide d'une source de tension, ce qui fait circuler le courant dans la phase pour produire une polarité à chaque extrémité ou pôle du stator. Le rotor est le composant rotatif réel du moteur. Il peut être magnétisé ou non magnétisé, selon le type de moteur choisi [25].

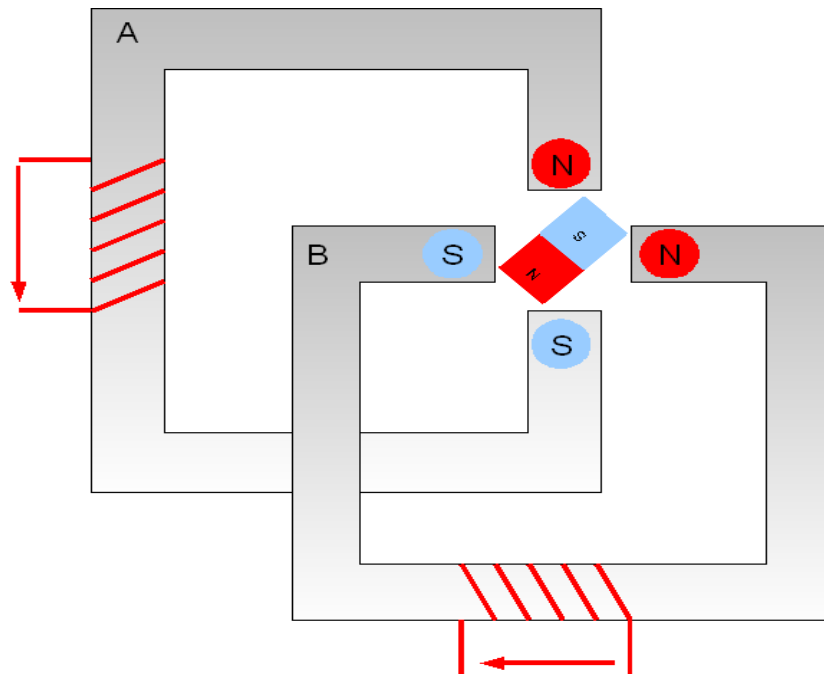


Figure 7 : Schéma d'un moteur pas à pas [4]

3.3.1 Le choix de moteur pas à pas :

Lors de la sélection d'un moteur pas-à-pas pour une application particulière, il faut tenir compte de plusieurs exigences de performance :

- La taille de pas.
- Le poids.
- La vitesse maximale [24].

Pour notre projet nous avons choisi le 42BYGHW811 illustré par la figure 8, c'est un moteur pas à pas bipolaire, il est très populaire dans les applications CNC telles que l'impression 3D. Ce moteur peut se déplacer par pas de 1,8 degré. L'arbre de ce moteur a une épaisseur de 5 mm et une longueur de 24 mm. Le moteur a également un couple de maintien allant jusqu'à 4,5 kg par centimètre. Ce moteur est très facile à utiliser et peut être contrôlé jusqu'à 1/32 ou 1/16 micropas avec un driver comme le DRV8825 ou A4988 [26].

Ce moteur a les caractéristiques électriques suivantes [27] :

Chapitre 1 : Architecture matérielle

- Précision de pas : $\pm 5 \%$
- Plage de température ambiante : $20^{\circ}\text{C} \sim 50^{\circ}\text{C}$
- Résistance d'isolement : $100 \text{ M}\Omega$ Min. 50 V CC
- Angle de pas : $1,8^{\circ}$ (200 pas / rotation)
- Tension nominale : $3,1 \text{ V}$
- Courant nominal : $2,5 \text{ A}$
- Résistance : $1,25 \Omega$ par bobine
- Inductance : $1,8 \text{ mH}$
- Tolérance à la montée en température : 80°C
- Couple de maintien : $4800 \text{ g}\cdot\text{cm}$
- Couple de détente : $280 \text{ g}\cdot\text{cm}$
- Inertie du rotor : $68 \text{ g}\cdot\text{cm}^2$
- Nombre de pistes : 4
- Longueur du moteur : $4,8 \text{ cm}$
- Poids : 340 grammes



Figure 8 : Moteur pas à pas 42BYGHW811

Chapitre 1 : Architecture matérielle

3.4 DRV_8825 :

Pour les imprimantes, les scanners et d'autres applications d'équipement automatisé, le DRV8825 fournit une solution de pilotage motorisé intégrée. L'appareil est conçu pour faire fonctionner un moteur pas à pas bipolaire et comprend deux pilotes H-bridge et un indicateur de micro-pas. Pour piloter les enroulements du moteur, le bloc d'entraînement de sortie est constitué de MOSFET de puissance N-canal configurés comme des ponts H complets. À 24 V et 25°C, le DRV8825 peut alimenter jusqu'à 2,5 A de courant de chaque sortie (avec une dissipation thermique suffisante) [28].

Les circuits du contrôleur peuvent être facilement interfacés à l'aide d'une simple interface STEP/DIR. Le moteur peut être configuré en mode full_step à 1/32-step à l'aide des broches de mode [28].

Le DRV8825 intègre un limiteur de courant actif et réglable. Il est doté également d'une protection contre les surintensités, la surchauffe avec 6 résolutions de micro_step (jusqu'au 1/32 de pas). Il fonctionne entre 8.2 et 45 V et sais délivrer approximativement 1.5 A par phase sans refroidissement (par refroidisseur ou air forcé -conçu pour 2.2A par bobine en utilisant un système de refroidissement efficace/adéquat) [29].

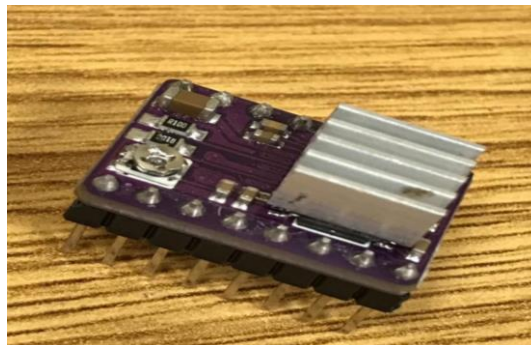


Figure 9 : DRV_8825 et son refroidisseur

Les moteurs pas à pas ont souvent une spécification de taille de pas qui s'applique à des pas entiers (par exemple, 1,8° ou 200 pas par tour). Des résolutions plus élevées sont possibles avec un pilote micro-pas comme le DRV8825, car il permet des emplacements de pas intermédiaires, obtenus en alimentant les bobines avec des niveaux de courant intermédiaires. Les entrées de sélection de la résolution (taille du pas) (MODE0, MODE1 et MODE2) permettent de choisir parmi les six résolutions de pas répertoriées dans le tableau 1. Comme les trois entrées de sélection ont des résistances pull-down de 100k intégrées, le mode pas complet (Full_step) est obtenu en laissant ces trois broches de sélection de micro-pas déconnectées. La limite de courant doit être réglée suffisamment bas pour que la limitation de courant s'engage afin que les modes micro-pas fonctionnent correctement. Sinon, le moteur sautera des micro-pas car les niveaux de courant intermédiaires ne seront pas correctement maintenus [29].

Chapitre 1 : Architecture matérielle

Tableau 1 : configuration du registre de résolution

MODE 0 (M0)	MODE 1 (M1)	MODE 2 (M2)	Résolution du micro pas
0	0	0	Pas complet
1	0	0	1/2 pas
0	1	0	1/4 pas
1	1	0	1/8 pas
0	0	1	1/16 pas
1	0	1	1/32 pas
0	1	1	1/32 pas
1	1	1	1/32 pas

Le DRV8825 possède les circuits nécessaires pour commander un moteur pas à pas bipolaire. Les moteurs pas à pas bipolaires ont généralement quatre fils, deux pour chaque bobine.

3.4.1 Brochage du DRV_8825 :

- Enable : Logique Inversée, ce pin permet d'activer ou désactiver le moteur, quand il est mis dans un état haut (en logique 1) le moteur est désactivé et quand il est mis en état bas (en logique 0) le moteur est activé.
- M0, M1, M2 : Ces pins permettent de sélectionner la configuration de mode de résolution de moteur pas à pas, Ces broches disposent de résistances Pull-Down ramenant le potentiel à 0v lorsque rien n'est connecté sur ces broches.
- RESET : Logique inversée. Permet de faire une réinitialisation du module. Généralement connecté sur la broche "SLEEP".
- SLEEP : Logique inversée. Généralement connecté sur la broche "Reset" du module
- STEP : En l'envoyant un signal d'horloge (haut puis bas) le moteur avancera d'un pas
- DIR : Permet d'indiquer la direction de rotation du moteur. Etat Haut=High pour tourner dans un sens, Etat bas=Low pour tourner dans l'autre sens
- VMOT : Tension d'alimentation du moteur, tension entre 8.2 et 45v.
- GND : Sous "VMOT", masse pour l'alimentation du moteur.
- A1 A2 : Première bobine du moteur pas à pas bipolaire.
- B1 B2 : Deuxième bobine du moteur pas à pas bipolaire.

Chapitre 1 : Architecture matérielle

- FAULT : Logique inversée. Brancher une tension entre 2.5 et 5.25v, celle qui alimente le microcontrôleur.
- GND : sous FAULT. Masse commune avec le microcontrôleur [30].

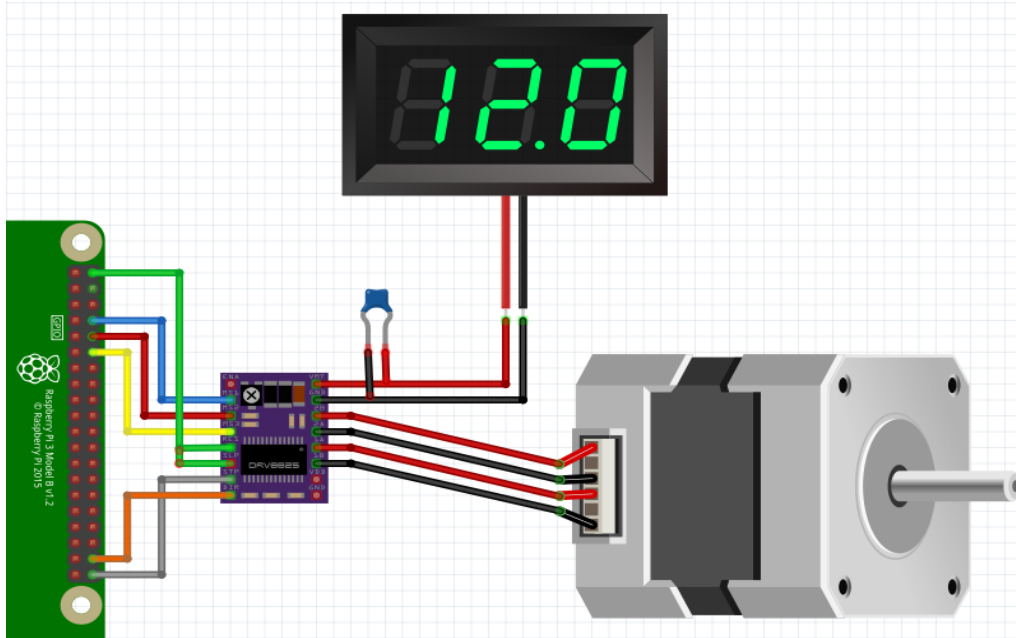


Figure 10 : Câblage du DRV8825 avec un moteur pas à pas

3.5 Le capteur de fin de course :

Un capteur de fin de course est un dispositif mécanique qui modifie l'état de l'interrupteur (ouvert/fermé) en entrant en contact physique avec l'actionneur de l'interrupteur. Ils sont réalisés à base de microcontacts placés dans un corps de protection et muni d'un système de commande ou tête de commande [31].

C'est un commutateur, commandé par le déplacement d'un organe de commande (corps d'épreuve). Lorsque le corps d'épreuve est actionné, un contact électrique est ouvert ou fermé. Le corps peut être couplé à une variété de modèles, y compris une tête droite, inclinée ou multidirectionnelle avec divers dispositifs d'attaque (à poussoir, à levier, à tige) [32].



Figure 11: fin de course

Chapitre 1 : Architecture matérielle

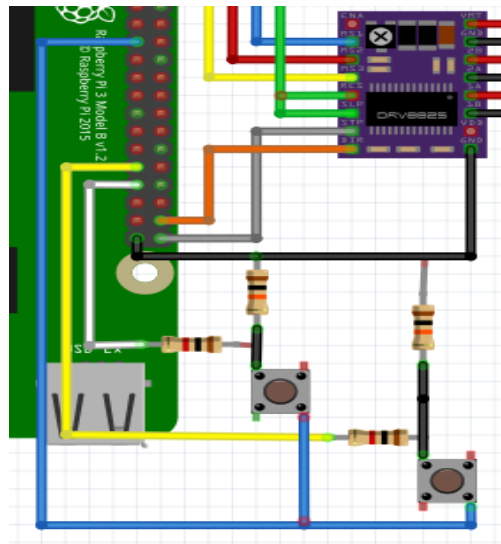


Figure 12 : le brochage du capteur de fin de course avec le Raspberry PI 4

3.6 Le microscope USB :

Le microscope USB est utilisé pour la visualisation en temps réel ou la documentation sur l'ordinateur jusqu'à 500 grossissements / 1280 x 1024 pixels. L'analyse avec ce microscope USB se fait de façon précise et fait apparaître des détails invisibles à l'œil nu, des plantes, des objets ou de petits organismes est utile pour des travaux en sciences ou pour l'enseignement [33]. Le Microscope USB a les caractéristiques suivantes [34] :

Tableau 2 : Caractéristiques du microscope USB

Résolution	2 Mégapixel (1280x1024 pixels)
Grossissement	De 50 à 500
Champ de vision	Min : 19mm x1.5mm Max : 82mm x65mm
Couleurs	24 bits RGB
Connexion au PC	USB 2.0
Alimentation	USB 2.0
Dimensions	110 x33mm
Poids	90 g

Chapitre 1 : Architecture matérielle



Figure 13: le microscope USB [6]

4. Conclusion :

Ce chapitre est dédié à la partie hardware de notre projet. Dans la partie hardware, nous avons présenté les composants utilisés :

- Châssis.
- Raspberry pi de façon générale puis le Raspberry pi 4 de façon précise.
- Le moteur pas à pas.
- Le DRV_8825 et leur câblage avec un moteur bipolaire 42BYGHW811.
- Le capteur fin de course.
- Le microscope USB.

Chapitre 2 :

Architecture

Logicielle

Chapitre 2 : Architecture logicielle

1. Introduction :

Dans ce chapitre nous présenterons ensuite la partie software où nous présenterons une description des langages de programmation utilisés dans la partie backend et la partie frontend et La relation entre eux.

2. La partie software :

Front-end et Back-end sont les deux termes les plus utilisés dans le développement Web. Ces termes sont très cruciaux pour le développement Web mais sont assez différents les uns des autres. Chaque côté doit communiquer et fonctionner efficacement avec l'autre comme une seule unité pour améliorer la fonctionnalité du site Web [35].

2.1 Back-end :

Le back-end est le côté serveur. Il stocke et organise les données et s'assurer que tout fonctionne correctement du côté client du site (Front-end). Il s'agit de la section du site Web à laquelle l'accès est interdit et avec laquelle c'est impossible d'interagir. C'est la partie du logiciel qui n'est pas en contact direct avec les utilisateurs. Les utilisateurs peuvent accéder aux pièces et fonctionnalités développées par les concepteurs de back-end directement via une application front-end. La partie back-end est construite à l'aide de certains langages comme le Python et le Node.js [35].

2.1.1 Python :

À la fin des années 1980 python a été développé par Guido van Rossum, Aujourd'hui, il est développé par la Python Software-Fondation [36]. Python est un langage de programmation polyvalent, ce qui signifie qu'il peut être utilisé pour une variété de tâches telles que la science des données, le développement de logiciels et de sites Web, l'automatisation et la réalisation de tâches en général. Python est récemment devenu l'un des langages de programmation les plus utilisés sur la planète [37].

A. Les versions python :

Python a deux versions : 2.x et 3.x dans notre projet nous avons utilisés le python 3.9.2 qui était déjà inclus dans le Raspberry

Python 3 est la version la plus récente du langage de programmation Python. Cette version a été principalement publiée pour résoudre les problèmes de Python 2. Python 3 a évolué au fil du temps pour être utile à un large éventail de secteurs comme le développement web, la science des données et les scripts analytiques, entre autres. Python 3 est livré avec un grand nombre de bibliothèques et est facile à combiner avec d'autres langages, des efforts ont été déployés pour que Python 3 prenne en charge un grand nombre des fonctionnalités importantes que Python 2 offrait, et Python 2 a été retiré en 2020 [38].

Chapitre 2 : Architecture logicielle

2.1.2 Node.js :

JavaScript est un langage de script bien connu qui a été créé pour gérer l'interaction entre les utilisateurs du navigateur [39]. Grâce à node.js, il est désormais possible d'utiliser JavaScript en dehors du navigateur, et même de s'interfacer avec le matériel sur des appareils comme le Raspberry Pi [40]. Node JS est un serveur web JavaScript. Il s'agit d'une plateforme qui permet de créer et d'exécuter des applications JavaScript. Elle est surtout utilisée pour créer des applications web qui permettent d'échanger des données. Une application web est un programme qui s'exécute uniquement sur un serveur et auquel on accède via Internet à l'aide d'un navigateur client. Une application web est composée de trois parties : un client, un serveur et une base de données [39]. Node.js fonctionne sur l'architecture Single Threaded Event Loop ce qui lui permet de gérer plusieurs clients à la fois. Pour l'installer il suffit de suivre les instructions données dans ce site <https://nodejs.org/en/> [41].

B. Node Package Manager (npm) :

L'une des meilleures choses qui sont arrivées à node.js est Npm. Npm est le gestionnaire de paquets de Node.js.

NPM est un outil intégré qui est inclus par défaut avec chaque installation de Node. NPM facilite la gestion des modules dans les projets Node en téléchargeant des paquets, en résolvant des dépendances, en exécutant des tests et en installant des utilitaires en ligne de commande. Le système NPM est destiné à faciliter la disponibilité et l'installation d'un ensemble de composants réutilisables accessibles au public via un référentiel en ligne, avec gestion des versions et des dépendances [42].

2.1.3 Child process :

Une requête du serveur est exécutée sur un processus par Node.js, qui fonctionne sur single thread. Ce processus, en particulier, est le processus principal, et c'est lui qui répond aux demandes du client. Cependant, Node.js propose également des Child process, qui sont créés par le processus principal. L'une des raisons d'être Child process est d'éviter que quoi que ce soit ne bloque le serveur principal [43].

En utilisant le module de Child process de Node, nous pouvons rapidement créer un processus enfant, et ces processus enfants peuvent facilement communiquer entre eux en utilisant un système de messages [44].

Le module de Child process nous permet d'accéder aux fonctionnalités du système d'exploitation en exécutant n'importe quelle commande système dans un processus enfant. Nous pouvons contrôler et écouter le flux de sortie de ce processus [43].

Chapitre 2 : Architecture logicielle

2.2 Front-end :

La partie d'un site Web avec laquelle l'utilisateur interagit directement est appelée le front-end. Il est également appelé « côté client » de l'application. Il comprend tout ce que les utilisateurs voient directement. Pour la présentation des pages nous utilisons le HTML, pour la charte graphique nous utilisons le CSS et pour les interactions exécutées par le client nous utilisons le JavaScript. Sont les langages utilisés pour La structure, la conception, le comportement et le contenu de tout ce qui s'affiche sur les écrans de navigateur en général sont les langages utilisés pour le développement front-end [45].

2.2.1 HTML :

HyperText Markup Language, communément appelé HTML, est un langage de balisage standard utilisé pour créer une page Web et son contenu. Par exemple, le contenu peut être structuré dans un ensemble de paragraphes, une liste de points à puces ou à l'aide d'images et de tableaux de données. Un fichier HTML est lu par un navigateur web et rendu sous forme de page web visible ou audible. Le HTML est un langage de balisage, et non un langage de programmation, car il représente sémantiquement la structure d'une page web ainsi que des indices de présentation.

Grâce au Html il est possible d'afficher du texte, des images, des hyperliens (liens vers d'autres pages web), des formulaires, et même des vidéos (grâce au HTML5 la dernière version de Html) sur un navigateur web (Firefox, Chrome, Opera, etc.) [46].

2.2.2 CSS :

Signifie une "feuille de style en cascade". Les feuilles de style en cascade sont utilisées pour mettre en forme la mise en page des pages Web. Elles permettent de définir les styles de texte, la taille des tableaux et d'autres éléments des pages Web qui ne pouvaient auparavant être définis que dans une seule page HTML.

Le CSS facilite aux concepteurs de sites Web la modification des styles sur plusieurs pages à la fois. Au lieu de définir le style de chaque élément et de chaque bloc de texte dans le code HTML d'une page, les styles actuellement utilisés ne doivent être définis qu'une seule fois dans un document CSS. Une fois qu'un style est défini dans une feuille de style en cascade, il peut être utilisé sur toutes les pages qui font référence au fichier CSS.

Le CSS donne aux développeurs web un contrôle plus précis de l'apparence des pages web que le HTML car le CSS peut être utilisé pour définir le remplissage des cellules d'un élément, ainsi que le style, l'épaisseur et la couleur de la bordure d'élément, de même que le remplissage des photos ou d'autres objets. C'est pourquoi, de nos jours, la grande majorité des pages web comportent des feuilles de style en cascade [47].

Chapitre 2 : Architecture logicielle

2.2.3 Java Script :

JavaScript (souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web.

JavaScript est un langage de programmation principalement utilisé pour créer des pages Web interactives. Ce langage, incorporé dans un document HTML, n'est pas visible dans la fenêtre du navigateur. Il sert à améliorer le langage html : en effet, il permet d'exécuter des commandes du côté client (c'est-à-dire au niveau du navigateur et non du serveur web).

Ce code qui est exécuté par le navigateur Web est utile pour toutes les interactions du client sur la page Web. Il permet d'améliorer la présentation et l'interactivité des pages web [48].

2.3 Application client-serveur :

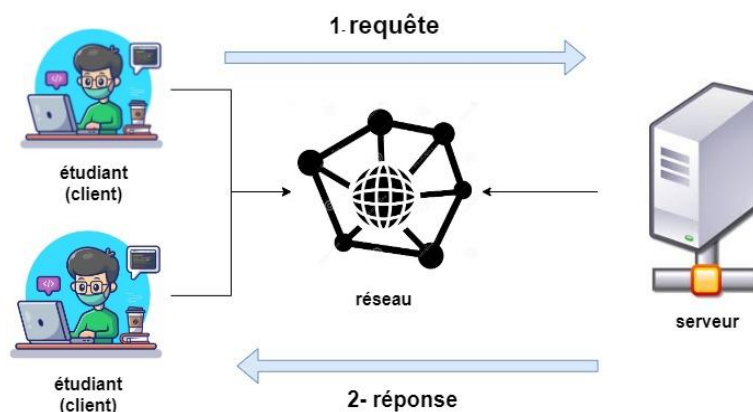


Figure 14 : schéma explicatif d'application client serveur

WebSocket est un protocole de couche applicative qui permet à un client et un serveur de communiquer dans les deux sens ce qui est l'objectif de notre projet.

L'étudiant inscrit doit disposer d'une page Web (côté client) où il peut gérer les axes de rotation à l'aide de boutons sur la surface du site, ainsi d'une fenêtre de caméra avec l'image appropriée pour contrôler le microscope depuis l'extérieur du laboratoire cette partie est la partie Front-end avec laquelle l'utilisateur interagit directement.

La partie du logiciel qui n'est pas en communication directe avec les utilisateurs est le côté serveur. C'est là que nous stockons et organisons les données pour garantir que tout le côté client du site fonctionne correctement.

Alors comment construire un WebSocket en Python :

Commençons par configurer l'environnement :

Chapitre 2 : Architecture logicielle

Installation de l'API WebSocket en Python, une fois que nous avons installé le package WebSockets nous pouvons passer à construire le server-client pour se connecter.

Créons maintenant un serveur :

Création d'un fichier serveur par exemple « server_python.py ». (Annexe 1)

Dans le fichier du serveur, nous importons les packages requis comme **WebSockets**

Créons un client maintenant :

Création d'un fichier HTML par exemple « client.html » [49]. (Annexe 2)

La page web contient une zone pour afficher l'image observée et des boutons pour déplacer la platine.



Figure 15 : Création du serveur websocket

3. Conclusion :

Ce chapitre est dédié à la partie software de notre projet, où nous avons passé à la description des langages de programmation utilisés dans la partie back-end qui sont le python, node.js. Puis, nous avons décrit Html, CSS et java script qui sont les langages utilisés dans la partie front-end. La communication entre back-end et front-end a été expliquée à la fin ce chapitre.

Chapitre 3 :

Mise en œuvre

Chapitre 3 : Mise en œuvre

1. Introduction :

Après avoir choisi les bons composants, on va maintenant passer à la partie mise en œuvre qui consiste à contrôler le microscope de telle sorte qu'il soit commandé par Internet.

La mise en œuvre est divisée en deux parties, la partie Hardware où on va réaliser la carte de circuit imprimé en précisant l'emplacement de chaque composant et la partie Software où on va réaliser le côté client et côté serveur.

2. Hardware :

La partie Hardware est la partie de matériel et de composants, elle propose une architecture fonctionnelle de la carte électronique en précisant l'emplacement de chaque composant électronique associé et le routage entre eux. Une fois fabriquée, la carte électronique est réceptionnée pour de nouveaux tests ainsi que sa mise au point finale, avant l'utilisation finale [50].

2.1 Les composants utilisés :

Tableau 3: les composants utilisés

Les composants utilisés :	Quantité :
Raspberry Pi 4	1
Moteur pas à pas	3
Driver de moteur pas à pas :DRV-8825	3
Capteur fin de course	6
Résistance 10k	6
Résistance 1k	6
Condensateur 100μF	3
Alimentation 12v	1

2.2 Le schéma de câblage :

Pour manipuler le microscope USB nous avons utilisé trois moteurs pas à pas de la référence 42BYGHW811 c'est un moteur bipolaire qu'il a 4 fils (2 pour chaque bobine), pour changer la polarité des bobines il nécessite un DRV_8825, un driver peut contrôler un seul moteur pas à pas.

Chapitre 3 : Mise en œuvre

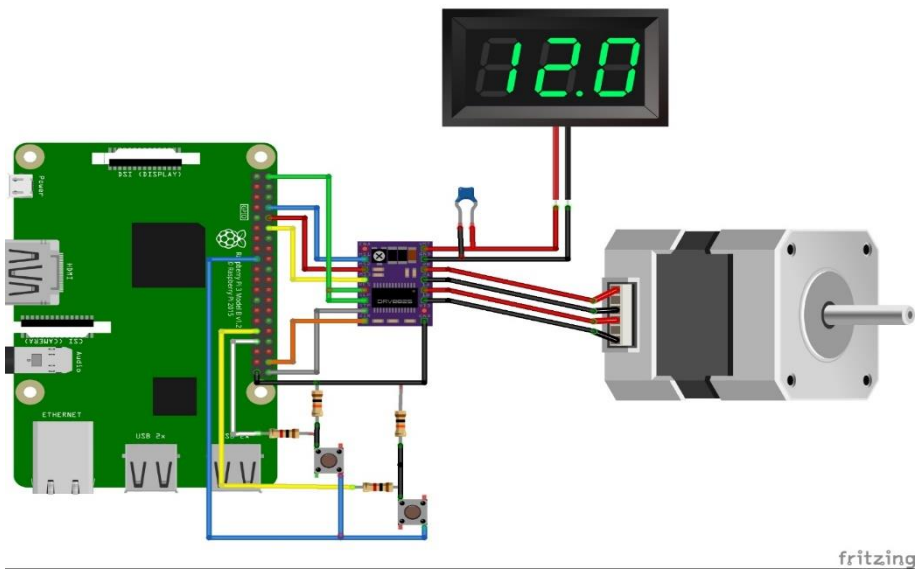


Figure 16 : le schéma de câblage d'un seul moteur pas à pas

Le câblage du DRV_8825 est très simple :

- V_MOT est relié à une alimentation de 12 V.
- GND est relié à la masse de l'alimentation.
- A1.A2.B1.B2 sont reliés au 4 fils de moteur pas à pas (A1 et A2 pour une bobine/ B1 et B2 pour la deuxième bobine).
- RESET et SLEEP sont reliés au 5V du Raspberry pi 4.
- STEP est relié au :
 - ❖ GPIO18 pour le moteur de l'axe X
 - ❖ GPIO25 pour le moteur de l'axe Y
 - ❖ GPIO20 pour le moteur de l'axe Z
- DIR est relié au :
 - ❖ GPIO24 pour le moteur de l'axe X
 - ❖ GPIO8 pour le moteur de l'axe Y
 - ❖ GPIO21 pour le moteur de l'axe Z
- GND est relié au GND du Raspberry pi 4
- M0.M1.M2 sont les pins de configuration de la résolution.

Chapitre 3 : Mise en œuvre

Le DRV8825 est sensible aux pics de tension destructeurs. Pour protéger la carte, nous avons placé un condensateur électrolytique de 100 μ F sur l'alimentation du moteur aussi près que possible du DRV8255.

Les capteurs de fin de course ont pour rôle de définir les extrémités des axes et la position HOME, un capteur de fin de course est composé de trois pattes :

- Com (pour Commun)
- NC ou NF (pour Normalement Fermé)
- NO (pour Normalement Ouvert)

La patte commune est utilisée dans tous les cas, elle est reliée au GND du Raspberry Pi4. Pour la seconde patte nous avons choisi NO (pull up), elle est reliée au 3.3v du Raspberry (passant par une résistance de 10k) plus un GPIO relié aussi à une résistance de protection de 1k.

- **Le réglage de Vref de DRV_8825 :**

Vref est une référence de tension qui correspond au courant maximum qui circulera pour les moteurs pas à pas, il est très important de la régler correctement car un excès de courant surchauffera et finira par brûler les moteurs et les drivers, également pour obtenir une meilleure performance de moteurs.

La formule de Vref : $Vref = 5 \times Courant\ max \times Rsens$ pour $Rsens = 0.1\Omega$, $Vref = Courant\ max / 2$.

Nous avons fixé la sonde négative de multimètre à la masse du DRV-8825 et la sonde positive du multimètre à un petit tournevis métallique. Pour faire ajuster la Vref en effectuant des ajustements comme montre la figure dessous.

Le courant maximal du moteur pas à pas 42BYGHW811 est 2,5 A pour un poids de 4,8 kg, mais nous n'avons utilisé que 1,5 A pour notre châssis en raison de son poids inférieur., ce qu'il fait $Vref = 0.75v$.

Chapitre 3 : Mise en œuvre

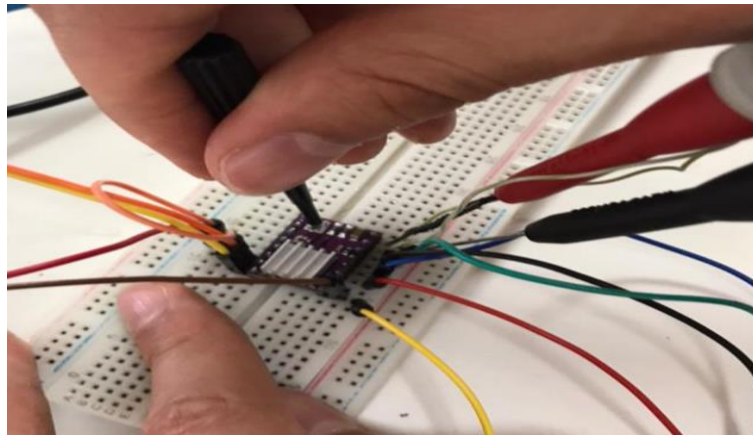


Figure 17 : Réglage de V_{ref}

2.3 EasyEDA :

EasyEDA est un outil en ligne gratuit qui aide les ingénieurs électriques, les apprentis, les étudiants en ingénierie et les passionnés d'électronique à acquérir une expertise utile dans le domaine de l'EDA (Electronics Design Automation). À l'aide de EasyEDA La conception et la simulation des circuits, ainsi que la conception du PCB (printed circuit board) sont simples et se font à l'aide d'un navigateur web [51].

2.3.1 Schéma de branchement :

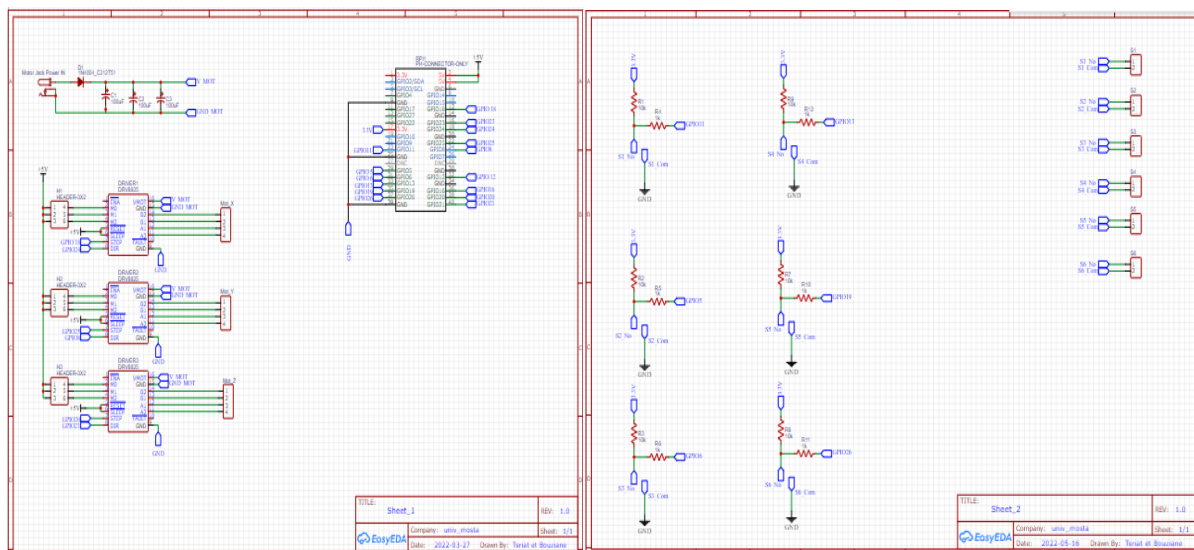


Figure 18: la carte mère

Chapitre 3 : Mise en œuvre

2.3.2 PCB :

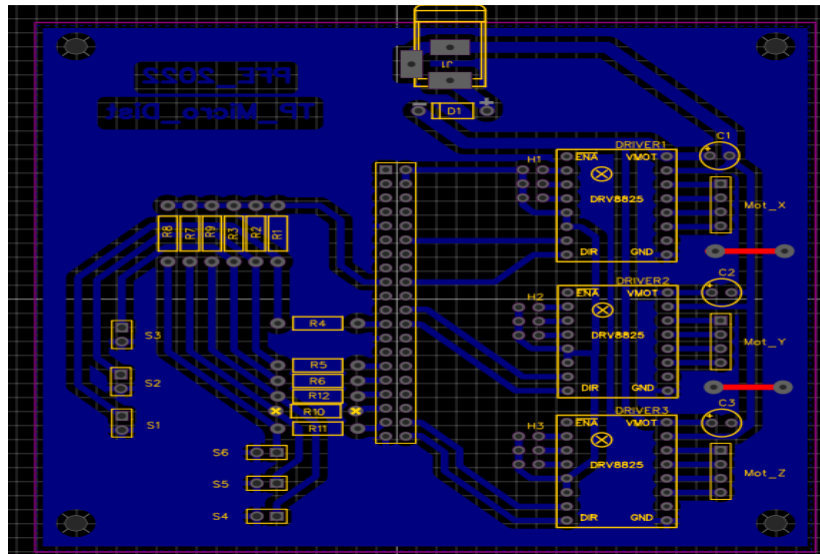


Figure 19: PCB

3. Software :

La solution logicielle est une application client-serveur. Le client et le serveur peuvent envoyer librement des données l'un à l'autre par le protocole Websocket. Le protocole Websocket permet une communication bidirectionnelle entre le client et le serveur [52].

Dans cette partie nous allons expliquer le fonctionnement de notre système et comment l'étudiant peut contrôler le microscope USB à distance.

3.1 Architecture logicielle :

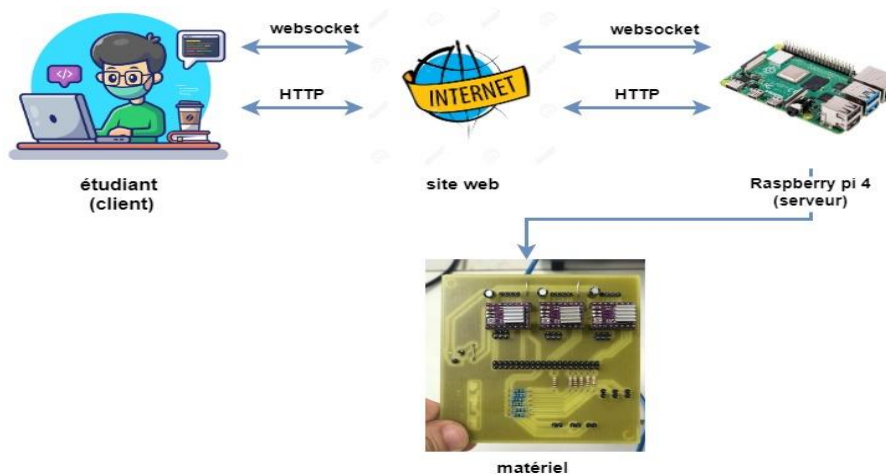


Figure 20 : le schéma explicatif d'Architecture logicielle

Ce système comporte quatre éléments :

- Le client : l'étudiant devant son PC.

Chapitre 3 : Mise en œuvre

- Le site web (internet).
- Le serveur : Raspberry pi 4.
- Le matériel.

Ce système utilise à la fois **HTTP** et **WebSocket** où :

- HTTP est utilisé pour servir un site Web qui comprend les fichiers HTML statiques et JavaScript.
- Une WebSocket est ensuite utilisée pour transmettre les commandes d'emplacement des axes au serveur.

Une fois la page web statique chargée dans le navigateur, il s'affiche deux boutons pour choisir la zone de TP désirée, six boutons pour contrôler les trois axes et un bouton HOME et une fenêtre de caméra. Après le chargement de la page, un code **JavaScript** exécuté dans le navigateur ouvrira une connexion **WebSocket** vers l'application serveur exécutée sur le Raspberry Pi4. Le fait de cliquer sur l'un des boutons, déclenche la transmission au serveur d'une **chaîne de caractères** pour changer l'emplacement de support de spécimen ou pour changer la position de microscope USB, où elle est traitée et l'état de GPIO est ajusté. Tout cela en temps réel, et avec une communication dans les deux sens.

3.2 Dans la page web:

- Les fichiers HTML et JavaScript sont transmis par la connexion HTTP.
- Le navigateur rend le HTML, puis le JavaScript.
- Le JavaScript en cours d'exécution crée une connexion WebSocket.

Le client choisit l'échantillon à observer et exécute des commandes de déplacement de l'échantillon. Ces instructions sont codées et envoyées au serveur :

- La déclaration d'une instance avec en paramètre l'URL contenant le protocole ws: dans le cas d'une connexion locale, on précisera le port : 8080.
 - `var gateway = `ws://172.16.15.221:8080``
- **onopen** détecte l'ouverture d'une connexion avec le serveur.
 - ```
function onOpen(event) {
 console.log('Connection opened');
}
```

## Chapitre 3 : Mise en œuvre

---

- L'envoi des données structurées, il suffit de les convertir en chaîne de caractères pour la transmission. Par exemple :

- ```
function stepper(axe,direction){  
    var data = {"axe":axe,  
                "direction": direction }  
  
    websocket.send(JSON.stringify(data)); } 
```

- **onclose** détecte la fermeture de la connexion à l'initiative du serveur.

- ```
function onClose(event) {
 console.log('Connection closed');
}
```

### 3.2.1 L'interface :

Dès que l'étudiant se connecte à l'interface, il doit sélectionner la zone appropriée pour son TP en cliquant sur le bouton correspondant (Zone1, Zone2). Chaque Zone (1 ou 2) est un échantillon à étudier. Une fois qu'il a choisi la zone, le support sera déplacé au centre de la zone, où se trouve le spécimen désiré, et il pourra utiliser les boutons restants où chaque bouton a une valeur qui empêche le support de dépasser les extrémités de la zone sélectionnée, les boutons sont :

- Le bouton Right(x)
- Le bouton Left(x)
- Le bouton Right(y)
- Le bouton Left(y)
- Le bouton Up(z)
- Le bouton Down(z)
- Le bouton Home

À travers une fenêtre de caméra en temps réel, les boutons Up(z) et Down(z) peuvent être utilisés pour contrôler le microscope USB.

Cette image est obtenue d'une observation d'une bactérie avec un grossissement de  $\times 50$ .

## Chapitre 3 : Mise en œuvre

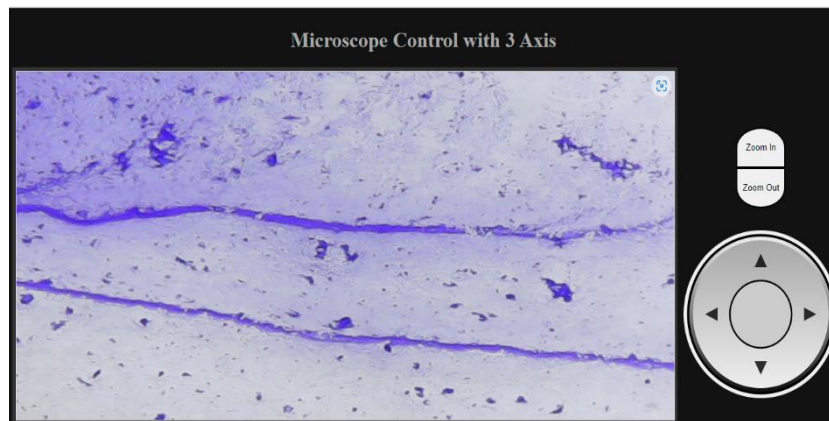


Figure 21 : capture d'écran d'une tentative

### 3.3 Dans le serveur d'application:

Nous avons deux parties de code serveur :

#### 3.3.1 LiveCam:

Est la solution de diffusion en temps réel de l'image Microscope USB utilisant GStreamer et Node.js

Il suffit d'installer correctement les bibliothèques nécessaires depuis ce site <https://www.npmjs.com/package/livecam>

Le serveur LiveCam donne une image en Streaming. Il délivre en temps réel l'image d'une web cam qui est ici un microscope.

L'image est accessible sous forme de page web avec adresse IP du serveur qui est ici le Raspberry.

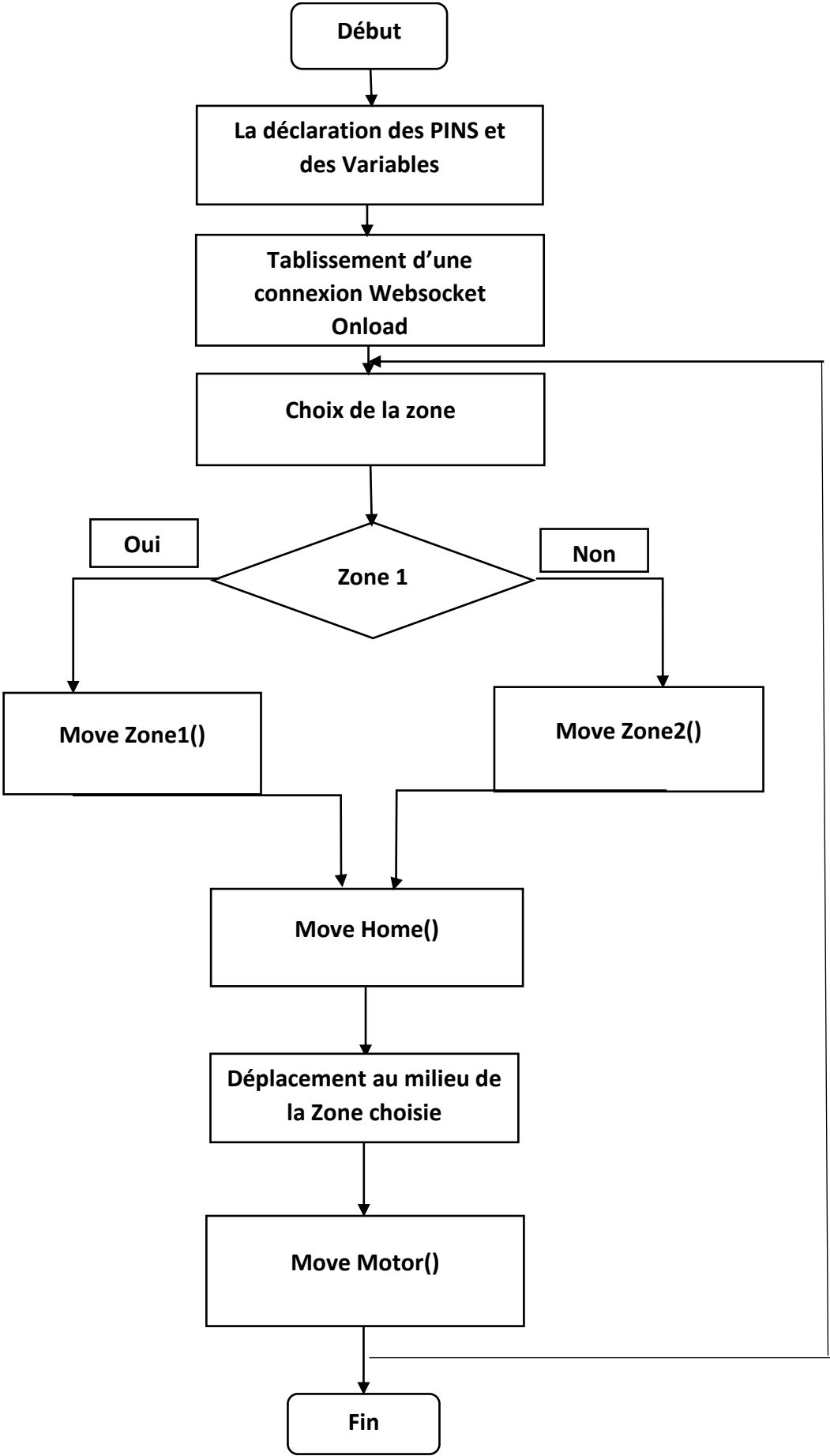
La page Web est intégrée dans le client sous forme d'une rame.

#### 3.3.2 Programme de Python:

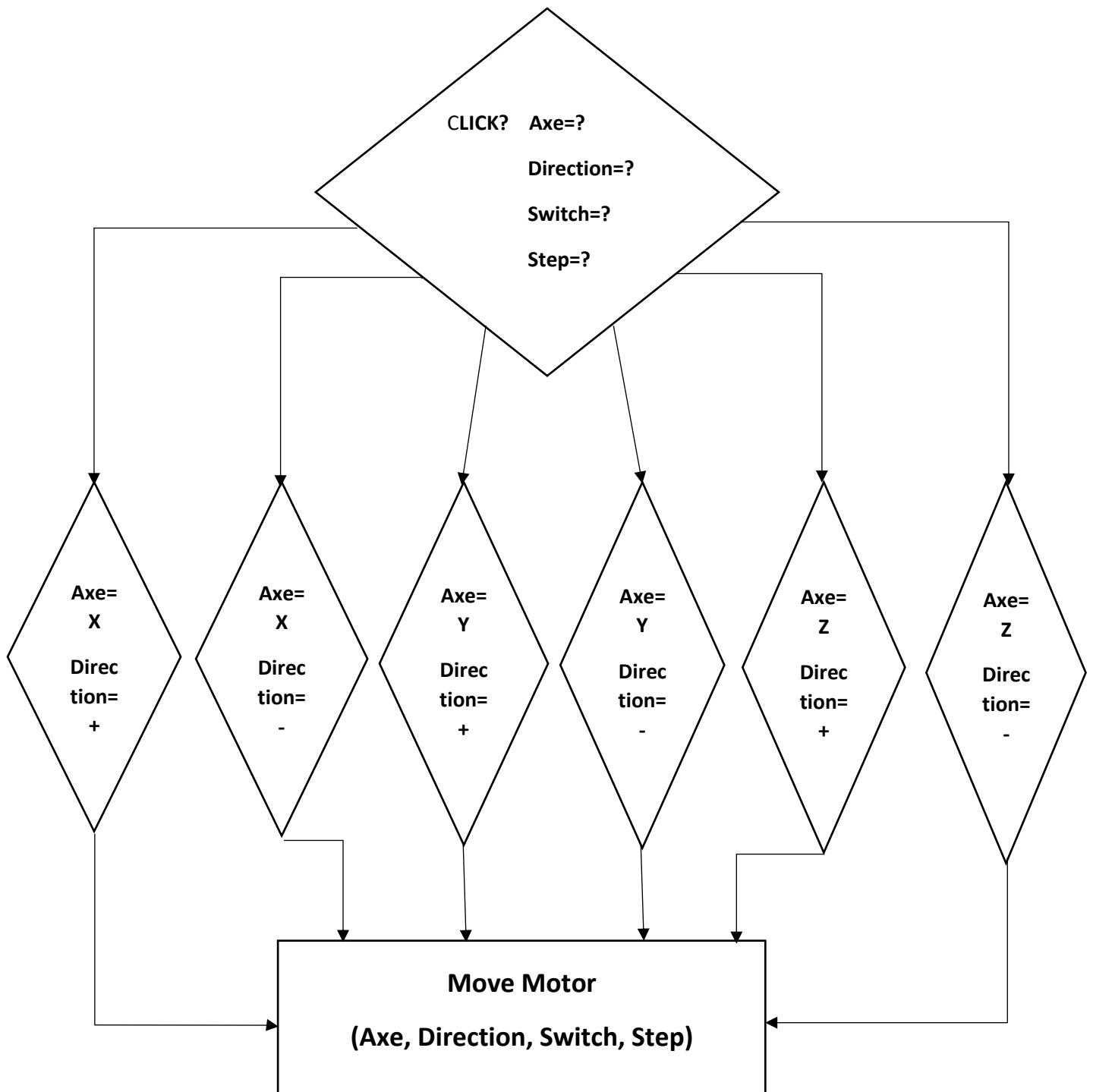
Le programme (code) de la commande du microscope USB distant est exécuté dans un fichier Python, le serveur HTTP écoute les connexions des clients et les transmet à l'application centrale. Lorsque le navigateur envoie une demande ou un événement, les gestionnaires (sont l'endroit où ces demandes ou événements seront traités) sont chargés de renvoyer les fichiers statiques au navigateur Web sur demande, mais aussi d'exploiter l'extrémité réceptrice de la WebSocket et de contrôler les GPIO locaux.

##### 3.3.2.1 L'Organigramme :

### Chapitre 3 : Mise en œuvre



## Chapitre 3 : Mise en œuvre



Une fois la connexion websocket est ouverte, l'étudiant doit choisir la zone désirée si c'était la zone 1, une fonction Move Zone1() sera exécutée.

Le support se déplacera au début à la position Home par la fonction Move Home() puis se déplacera au milieu de la zone choisie, à ce moment l'étudiant pourra déplacer la position de support et la hauteur du microscope USB en utilisant les boutons restants :

- RIGHT (X), LEFT(X)

## Chapitre 3 : Mise en œuvre

---

- RIGHT (Y), LEFT(Y)
- UP(Z), DOWN(Z)

Le fait de cliquer sur l'un des boutons une fonction MoveMotor(Axe, Direction, Switch, Step) sera exécutée et ses paramètres (Axe, Direction, Switch, Step) seront changés en fonction du bouton choisit.

### 4. Conclusion :

Ce chapitre est dédié à la partie pratique de notre projet. Nous avons commencé par le schéma de câblage des composants constitutifs puis la description de logiciel EasyEDA et la réalisation de la carte de circuit imprimé. Ensuite, nous avons décrit le fonctionnement de système dans la partie software avec un schéma explicatif. Nous avons même expliqué le rôle de chaque bouton dans l'interface.

---

# ***Conclusion générale***

---

# Conclusion générale

---

## Conclusion générale :

Le travail présenté dans ce mémoire a été consacré à la mise en œuvre d'un environnement de travaux pratiques à distance via le réseau Internet dans le domaine de la biologie c'est la microscopie à distance.

Pour mettre en œuvre un microscope télécommandé à distance, nous étions chargés de mettre en place une architecture matérielle spécifique et de développer un site web performant et facile à utiliser permettant à un étudiant distant de réaliser un TP à distance.

Ce projet nous a offert l'occasion d'approfondir nos connaissances en :

- L'utilisation de software PUTTY.
- La maîtrise de plusieurs langages de programmation :
  - Python.
  - Node.js.
- La création d'un site web en utilisant :
  - Html.
  - CSS.
  - Javascript.
- La connexion Websocket.
- L'utilisation de différentes bibliothèques comme :
  - SimpleWebSocketServer.
  - RPi.GPIO.
  - Livecam.
- La création des cartes à circuits imprimés PCB en utilisant l'application EasyEDA.
- La maîtrise des différentes étapes de l'impression de PCB :
  - Utilisation de la machine LPKF laser et Electronics.
- La découverte du Raspberry PI 4.

Lors de la réalisation de notre projet nous avons vécu quelques problèmes parmi lesquels nous citons :



## Conclusion générale

---

- L'échauffement de DRV\_8825 quand nous l'utilisons pendant une période prolongée.
- La fixation de l'agrandissement à une certaine valeur car ce n'est pas possible de le changer que manuellement.

L'enseignement devrait conserver cette culture de soutien au changement. Les développeurs qui font preuve d'originalité et d'adaptabilité dans la conception des travaux pratiques à distance et dans l'interactions avec les étudiants devraient être soutenus et encouragés par les universités. Notre projet reste toujours extensible par d'autres développeurs pour d'autres améliorations et ajout de nouvelles fonctionnalités pour qu'il serait plus développé nous proposons d'ajouter :

- Un fichier Json où l'enseignant peut ajouter d'autres zones sans accéder et changer le programme original en mettant les paramètres de chaque zone.
- L'utilisation de plusieurs microscopes USB avec différents agrandissements où l'étudiant peut choisir le zoom le plus adapté à son TP.
- Une calibration des pas (steps) en centimètres.

## Références bibliographiques

---

- [1] Steve Mackay and Darrell Fisher, Practical Online Learning and Laboratories, for Engineering, Science and Technology, IDC Technologies 2014, accessible à <https://eitdc.com/files/Practical Online Learning and Laboratories 5 April.pdf>
- [2]<https://theconversation.com/universities-after-covid-as-lecture-theatres-reopen-some-pandemic-teaching-methods-should-live-on-174652>
- [3] <https://www.mdpi.com/2227-7102/10/10/291/htm>
- [4]<https://universityservices.wiley.com/future-of-online-education-in-the-wake-of-covid19/>
- [5]<https://www.khanacademy.org/science/biology/structure-of-a-cell/introduction-to-cells/a/microscopy>
- [6] <https://www.britannica.com/technology/microscope>
- [7] <https://www.sciencedirect.com/topics/engineering/optical-microscope>
- [8]<https://fac.umc.edu.dz/snv/faculte/tc/2020/TD%20N%C2%B01%20M%C3%A9thodes%20de%20microscopie%20optique%20et%20%C3%A9lectronique.pdf>
- [9] <https://www.microscopemaster.com/usb-computer-microscope.html>
- [10] <https://prepas.org/index.php?article=74>
- [11] <https://www.lemagit.fr/definition/Hardware>
- [12] <http://www.elektronique.fr/cours/composants.php>
- [13]<https://123dok.net/document/q7w0knoz-conception-et-r%C3%A9alisation-d-une-machine-cnc.html>
- [14] <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [15] <https://www.piday.org/whats-a-raspberry-pi-and-how-does-it-work/>
- [16][https://magazines-attachments.raspberrypi.org/books/full\\_pdfs/000/000/038/original/BeginnersGuide-4thEd-Eng\\_v2.pdf](https://magazines-attachments.raspberrypi.org/books/full_pdfs/000/000/038/original/BeginnersGuide-4thEd-Eng_v2.pdf)
- [17]<http://bedford-computing.co.uk/learning/wp-content/uploads/2015/10/Raspberry-Pi-For-Dummies.pdf>
- [18] [https://fr.wikipedia.org/wiki/Raspberry\\_Pi](https://fr.wikipedia.org/wiki/Raspberry_Pi)

## Références bibliographiques

---

- [19] <https://www.efxkits.us/different-types-of-raspberry-pi-boards-its-application/>
- [20] <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-4-Product-Brief.pdf>
- [21] <https://www.hwlibre.com/fr/gpio-raspberry-pi/?msclkid=5725c1b4a54611eca7d2b605f01ac84b>
- [22] <https://www.framboise314.fr/scratch-raspberry-pi-composants/gpio/>
- [23] [https://www.faulhaber.com/fileadmin/user\\_upload\\_global/support/MC\\_Support/Motors/AppNotes/Faulhaber\\_AN001\\_EN.pdf](https://www.faulhaber.com/fileadmin/user_upload_global/support/MC_Support/Motors/AppNotes/Faulhaber_AN001_EN.pdf)
- [24] <http://www.t-es-t.hu/download/microchip/an907a.pdf>
- [25] [https://www.microchip.com/stellent/groups/sitecomm\\_sg/documents/devicedoc/en543047.pdf](https://www.microchip.com/stellent/groups/sitecomm_sg/documents/devicedoc/en543047.pdf)
- [26] <https://opencircuit.fr/produit/moteur-nema-17-pas-48mm-12v-24v-1.7a>
- [27] [https://www.phidgets.com/productfiles/3303/3303\\_0/Documentation/3303\\_0\\_Datasheet.pdf](https://www.phidgets.com/productfiles/3303/3303_0/Documentation/3303_0_Datasheet.pdf)
- [28] [https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1650533594097&ref\\_url=https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1650533594097&ref_url=https%253A%252F%252Fwww.google.com%252F)
- [29] <https://www.tme.eu/Document/1dd18faf1196df48619105e397146fdf/POLOLU-2133.pdf>
- [30] <https://wiki.mchobby.be/index.php?title=DRV8825>
- [31] <https://pobot.org/Interrupteur-Microswitch.html>
- [32] [http://lycees.ac-rouen.fr/maupassant/Melec/co/Techno/Detecteur/web/co/grain\\_Detect\\_meca.html](http://lycees.ac-rouen.fr/maupassant/Melec/co/Techno/Detecteur/web/co/grain_Detect_meca.html)
- [33] <https://www.microscopemaster.com/different-types-of-microscopes.html>
- [34] <https://www.pce-france.fr/fiches-techniques/fiche-technique-pce-mm200.pdf>
- [35] <https://www.geeksforgeeks.org/frontend-vs-backend/>
- [36] <https://juser.fz-juelich.de/record/891478/files/python-2021.pdf>
- [37] <https://www.techopedia.com/definition/3533/python>
- [38] <https://www.guru99.com/python-2-vs-python-3.html>
- [39] <https://www.data-transitionnumerique.com/nodejs-guide-complet/>
- [40] [https://www.academia.edu/19259375/Raspberry\\_Pi\\_The\\_Complete\\_Manual](https://www.academia.edu/19259375/Raspberry_Pi_The_Complete_Manual)
- [41] <https://core.ac.uk/download/pdf/232792216.pdf>

## Références bibliographiques

---

- [42] <https://cdn-learn.adafruit.com/downloads/pdf/node-embedded-development.pdf>
- [43] <https://blog.sessionstack.com/how-javascript-works-building-a-child-process-in-node-js-776c8c51a8b0>
- [44] <https://www.freecodecamp.org/news/node-js-child-processes-everything-you-need-to-know-e69498fe970a/>
- [45] <https://frontendmasters.com/guides/front-end-handbook/2019/>
- [46] [https://pixees.fr/informatiquelycee/n\\_site/snt\\_web\\_html.html](https://pixees.fr/informatiquelycee/n_site/snt_web_html.html)
- [47] <https://techlib.fr/definition/css.html>
- [48] <https://www.gralon.net/articles/internet-et-webmaster/creation-site-internet/article-javascript--presentation-et-applications-1776.htm>
- [49] <https://www.piesocket.com/blog/python-websocket/>
- [50] <https://www.essentracomponents.com/en-gb/news/guides/the-ultimate-guide-to-pcb-hardware>
- [51] <https://www.electrobrahim.com/2016/09/easyeda-pcb.html>
- [52] <http://remy-manu.no-ip.biz/Java/Tutoriels/J2EE/websocket.html>

# Listes des figures et tableaux

---

## Listes des figures :

|                                                                                        |    |
|----------------------------------------------------------------------------------------|----|
| <b>Figure1</b> : Microscope optique [1] .....                                          | 10 |
| <b>Figure2</b> : Schéma synoptique .....                                               | 14 |
| <b>Figure 3</b> : L'interface.....                                                     | 15 |
| <b>Figure 4</b> : Châssis.....                                                         | 17 |
| <b>Figure 5</b> : Raspberry Pi 4 [2].....                                              | 19 |
| <b>Figure 6</b> : pins de Raspberry Pi 4 [3] .....                                     | 20 |
| <b>Figure 7</b> : Schéma d'un moteur pas à pas [4] .....                               | 21 |
| <b>Figure 8</b> : moteur pas à pas 42BYGHW811.....                                     | 22 |
| <b>Figure 9</b> : DRV_8825 et son refroidisseur.....                                   | 23 |
| <b>Figure 10</b> : Câblage du DRV8825 avec un moteur pas à pas.....                    | 25 |
| <b>Figure 11</b> : fin de course.....                                                  | 25 |
| <b>Figure 12</b> : le brochage du capteur de fin de course avec le Raspberry PI 4..... | 26 |
| <b>Figure 13</b> : le microscope USB [6] .....                                         | 27 |
| <b>Figure 14</b> : schéma explicatif d'application client serveur.....                 | 32 |
| <b>Figure 15</b> : Création du serveur websocket.....                                  | 33 |
| <b>Figure 16</b> : le schéma de câblage d'un seul moteur pas à pas.....                | 36 |
| <b>Figure 17</b> : le réglage de Vref.....                                             | 38 |
| <b>Figure 18</b> : la carte mère.....                                                  | 38 |
| <b>Figure 19</b> : PCB.....                                                            | 39 |
| <b>Figure 20</b> : Schéma explicatif d'Architecture logicielle.....                    | 39 |
| <b>Figure 21</b> : capture d'écran d'une tentative.....                                | 42 |

## Listes des tableaux :

|                                                                                |    |
|--------------------------------------------------------------------------------|----|
| <b>Tableau 1</b> : les bits de mode de fonctionnement selon la résolution..... | 24 |
| <b>Tableau 2</b> : Caractéristiques du microscope USB.....                     | 26 |
| <b>Tableau 3</b> : les composants utilisés.....                                | 35 |

## Annexe

---

### 1. PuTTY :

PuTTY est une implémentation gratuite de SSH (et telnet) pour les PC exécutant Microsoft Windows. C'est un client d'accès à distance à un compte sur un Unix ou sur autre système multi-utilisateurs à partir d'un PC. Son principal avantage est que SSH fournit une connexion sécurisée et cryptée au système distant.

Après d'avoir installé correctement le PuTTY, passant au comment connecter PuTTY :

- Le lancement de client PuTTY SSH, entrer l'adresse IP SSH et le port SSH de serveur. Cliquer sur le bouton **OPEN** pour continuer.
- Un message de connexion **LOGIN AS** : apparaîtra demandera d'entrer le nom d'utilisateur SSH.
- Entrer le mot de passe. Pour des raisons de sécurité, l'écran n'affichera pas le mot de passe mais l'enregistrera.

[https://sitelec.org/download\\_page.php?filename=cours/abati/download/putty.pdf](https://sitelec.org/download_page.php?filename=cours/abati/download/putty.pdf)

### 2. Serveur d'écho :

Echo Server est quelque chose qui reflète le message envoyé au client.

Les serveurs Echo sont assez simples à fabriquer. Cela implique 3 étapes faciles :

- Créer un serveur.
- Accepter le message.
- Renvoyer le message au Client.

<https://soshace.com/node-js-lessons-echo-server/>

### 3. Commandes linux de base :

| Commande SSH        | Utilisation                                                            |
|---------------------|------------------------------------------------------------------------|
| ls                  | Pour lister tous les fichiers et répertoires.                          |
| cd [directory]      | Pour passer d'un répertoire à un autre.                                |
| cd ../..            | Pour revenir au répertoire d'origine.                                  |
| mkdir [folder name] | Pour créer un répertoire.                                              |
| rm [file name]      | Pour supprimer un fichier.                                             |
| nano [file name]    | Pour modifier un fichier directement depuis la ligne de commande [ ]]. |

[ ] <https://kinsta.com/fr/blog/commandes-ssh/>

## Annexe

---

### 4. Programme python :

```
import json
from time import sleep
import RPi.GPIO as GPIO
from SimpleWebSocketServer import SimpleWebSocketServer, WebSocket

CW = 1 # Clockwise Rotation
CCW = 0 # Counterclockwise Rotation
SPR = 25 # Steps per Revolution
SPR_Z = 300 #Steps per Revolution for the axe Z

Max_steps_X = 859 #Maximum steps for the axe X
Max_steps_Y = 740 #Maximum steps for the axe Y
Max_steps_Z = 4846 #Maximum steps for the axe Z
step_count = SPR
delay = .0008 #delay

X_Axe motor:
STEP_X = 18 # Step GPIO Pin
DIR_X = 24 # Direction GPIO Pin
SW_X_S = 6 # Start limit switch
SW_X_E = 26 # End limit switch

Y_Axe motor:
STEP_Y = 25 # Step GPIO Pin
DIR_Y = 8 # Direction GPIO Pin
SW_Y_S = 5 # Start limit switch
SW_Y_E = 19 # End limit switch

Z_Axe motor:
STEP_Z = 20 # Step GPIO Pin
DIR_Z = 21 # Direction GPIO Pin
SW_Z_S = 11 # Start limit switch
SW_Z_E = 13 # End limit switch

#Position Trackers
position_x=0
position_y=0
position_z=0
#Current Range
limited_x_max = 500 #limited maximum steps for the axe X
limited_x_min = 0 #limited minimum steps for the axe X
limited_y_max = 500 #limited maximum steps for the axe Y
limited_y_min = 0 #limited minimum steps for the axe Y
#Zone_1 range
Zone_1_x_max = 400 #maximum for the axe X in Zone 1
Zone_1_x_min = 0 #minimum for the axe X in Zone 1
Zone_1_y_max = 500 #maximum for the axe Y in Zone 1
Zone_1_y_min = 0 #minimum for the axe Y in Zone 1
```

## Annexe

```
#Zone_2 range
Zone_2_x_max = 800 #maximum for the axe X in Zone 2
Zone_2_x_min = 400 #minimum for the axe X in Zone 2
Zone_2_y_max = 500 #maximum for the axe Y in Zone 2
Zone_2_y_min = 0 #minimum for the axe Y in Zone 2

GPIO.setmode(GPIO.BCM)

X_Axe motor GPIO:
GPIO.setup(STEP_X, GPIO.OUT)
GPIO.setup(DIR_X , GPIO.OUT)
GPIO.setup(SW_X_S, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(SW_X_E, GPIO.IN, pull_up_down=GPIO.PUD_UP)
Y_Axe motor GPIO:
GPIO.setup(STEP_Y, GPIO.OUT)
GPIO.setup(DIR_Y , GPIO.OUT)
GPIO.setup(SW_Y_S, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(SW_Y_E, GPIO.IN, pull_up_down=GPIO.PUD_UP)
Z_Axe motor GPIO:
GPIO.setup(STEP_Z, GPIO.OUT)
GPIO.setup(DIR_Z , GPIO.OUT)
GPIO.setup(SW_Z_S, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.setup(SW_Z_E, GPIO.IN, pull_up_down=GPIO.PUD_UP)

class SimpleEcho(WebSocket):
 def handleMessage(self):
 Data = json.loads(self.data)
 print(f'Data: {Data}')
 axe = Data["axe"]
 direction = Data["direction"]
 global step_count
 if Data["axe"] in ['X'] and direction in ['+']:
 axe = STEP_X
 dir_pin = DIR_X
 direction = CW
 switch = SW_X_E
 step_count = SPR
 elif Data["axe"] in ['X'] and direction in ['-']:
 axe = STEP_X
 dir_pin = DIR_X
 direction = CCW
 switch = SW_X_S
 step_count = SPR
 elif Data["axe"] in ['Y'] and direction in ['+']:
 axe = STEP_Y
 dir_pin = DIR_Y
 direction = CW
 switch = SW_Y_E
```



## Annexe

```
 step_count = SPR
 elif Data["axe"] in ['Y'] and direction in ['-']:
 axe = STEP_Y
 dir_pin = DIR_Y
 direction = CCW
 switch = SW_Y_S
 step_count = SPR
 elif Data["axe"] in ['Z'] and direction in ['+']:
 axe = STEP_Z
 dir_pin = DIR_Z
 direction = CW
 switch = SW_Z_E
 step_count = SPR_Z
 elif Data["axe"] in ['Z'] and direction in ['-']:
 axe = STEP_Z
 dir_pin = DIR_Z
 direction = CCW
 switch = SW_Z_S
 step_count = SPR_Z
 elif Data["axe"] in 'zone_1':
 move_zone_1()
 print('After Zone 1 ')
 return
 elif Data["axe"] in 'zone_2':
 move_zone_2()
 print('After Zone 2 ')
 return
 elif Data["axe"] in 'HOME':
 axe = 'Home'
 move_home()
 print('after home')

 if axe != 'Home':
 print('Axe pin:', axe)
 print('Direction pin:', dir_pin)
 print('Switch pin: ', switch)
 print('Direction: ', direction)
 move_motor(axe,dir_pin,switch,direction)

 def handleConnected(self):
 print(self.address, 'WebSocket connected')

 def handleClose(self):
 print(self.address, 'WebSocket closed')

def move_motor(step_pin,dir_pin,sw_pin,direction=0):
 GPIO.output(dir_pin, direction)
 print('status SW: ',GPIO.input(sw_pin))
```

## Annexe

```
print('SPR: ',step_count)
global position_x
global position_y
if(step_pin==STEP_X and direction == 1):

 if(position_x >= limited_x_max):
 print('Zone X Max reached')
 return
 else:
 position_x = position_x + step_count

elif(step_pin==STEP_X and direction == 0):

 if(position_x <= limited_x_min):
 print('Zone X Min reached')
 return
 else:
 position_x = position_x - step_count

elif(step_pin==STEP_Y and direction == 1):

 if(position_y >= limited_y_max):
 print('Zone Y Max reached')
 return
 else:
 position_y = position_y + step_count

elif(step_pin==STEP_Y and direction == 0):

 if(position_y <= limited_y_min):
 print('Zone Y Min reached')
 return
 else:
 position_y = position_y - step_count

for x in range(step_count):
 if GPIO.input(sw_pin) == 1 :
 GPIO.output(step_pin, GPIO.HIGH)
 sleep(delay)
 GPIO.output(step_pin, GPIO.LOW)
 sleep(delay)
 else:
 print('End limit switch reached:',sw_pin)
 GPIO.output(step_pin, GPIO.LOW)
 sleep(delay)
print(f'Posistion_X: {position_x}')
print(f'Posistion_Y: {position_y}')

def move_home():
```

## Annexe

```
print('Home Call')
global position_x
global position_y
global position_z
Move X_Axe to home position
GPIO.output(DIR_X, 0)
max_steps_count=Max_steps_X
while (GPIO.input(SW_X_S) and max_steps_count):
 GPIO.output(STEP_X, GPIO.HIGH)
 sleep(delay)
 GPIO.output(STEP_X, GPIO.LOW)
 sleep(delay)
 max_steps_count = max_steps_count - 1
 print(f'max steps count X: {max_steps_count}')
 position_x=0
print("X Axe is back to starting position")
print("Step Count: ",max_steps_count)
Move Y_Axe to home position
GPIO.output(DIR_Y, 0)
max_steps_count=Max_steps_Y
while (GPIO.input(SW_Y_S) and max_steps_count):
 GPIO.output(STEP_Y, GPIO.HIGH)
 sleep(delay)
 GPIO.output(STEP_Y, GPIO.LOW)
 sleep(delay)
 max_steps_count = max_steps_count - 1
 position_y=0
print("Y Axe is back to starting position")
print("Step Count: ",max_steps_count)
Move Z_Axe to home position
GPIO.output(DIR_Z, 1)
max_steps_count=Max_steps_Z
while (GPIO.input(SW_Z_E) and max_steps_count):
 GPIO.output(STEP_Z, GPIO.HIGH)
 sleep(delay)
 GPIO.output(STEP_Z, GPIO.LOW)
 sleep(delay)
 max_steps_count = max_steps_count - 1
 position_z=0
print("Z Axe is back to starting position")
print("Step Count: ",max_steps_count)

def move_zone_1():
 global limited_x_max
 global limited_x_min
 global limited_y_max
 global limited_y_min

 global Zone_1_x_max
```

## Annexe

```
global Zone_1_x_min
global Zone_1_y_max
global Zone_1_y_min

global position_x
global position_y

limited_x_max = Zone_1_x_max
limited_x_min = Zone_1_x_min
limited_y_max = Zone_1_y_max
limited_y_min = Zone_1_y_min

move_home()
while(position_x < ((limited_x_max + limited_x_min) // 2)):
 move_motor(STEP_X, DIR_X, SW_X_E, direction=1)
while(position_y < ((limited_y_max + limited_y_min) // 2)):
 move_motor(STEP_Y, DIR_Y, SW_Y_E, direction=1)

print("In Zone 1")
print(f'limit x max: {limited_x_max}')
print(f'limit x min: {limited_x_min}')
print(f'limit y max: {limited_y_max}')
print(f'limit y min: {limited_y_min}')

def move_zone_2():
 global limited_x_max
 global limited_x_min
 global limited_y_max
 global limited_y_min

 global Zone_2_x_max
 global Zone_2_x_min
 global Zone_2_y_max
 global Zone_2_y_min

 global position_x
 global position_y

 limited_x_max = Zone_2_x_max
 limited_x_min = Zone_2_x_min
 limited_y_max = Zone_2_y_max
 limited_y_min = Zone_2_y_min

 move_home()
 while(position_x < ((limited_x_max + limited_x_min) // 2)):
 move_motor(STEP_X, DIR_X, SW_X_E, direction=1)
 while(position_y < ((limited_y_max + limited_y_min) // 2)):
 move_motor(STEP_Y, DIR_Y, SW_Y_E, direction=1)
```

## Annexe

```
print("In Zone 2")
print(f'limit x max: {limited_x_max}')
print(f'limit x min: {limited_x_min}')
print(f'limit y max: {limited_y_max}')
print(f'limit y min: {limited_y_min}')

server = SimpleWebSocketServer('', 8080, SimpleEcho)
server.serveforever()
```