

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE M'HAMED BOUGARA-BOUMERDES



Faculté de Technologie

Département Ingénierie des Systèmes Électriques

Mémoire de Master

Présenté par :

ABDELLI Younes Nadji

Filière : Electronique

Spécialité : Systèmes Embarqués

**Étude et réalisation de NOOR : Un traducteur
portable arabe-braille en temps réel basé sur l'IoT**

Soutenu le 27 / 06 / 2024 devant le jury composé de :

AMMAR	Mohammed	Pr	Université de Boumerdès	Président
AKROUM	Hamza	MCA	Université de Boumerdès	Examineur
MESSAOUDI	Noureddine	MCA	Université de Boumerdès	Examineur

Année Universitaire : 2023/2024

Remerciements

Je tiens à exprimer ma profonde gratitude à **Mr. AMMAR MOHAMMED**. Votre persévérance, vos conseils avisés et votre soutien constant ont été la pierre angulaire de cette thèse. Sans vous, ce travail n'aurait jamais vu le jour.

À **Mr. AKROUM HAMZA**, pour votre accompagnement éclairé et votre sagesse, qui ont été des phares tout au long de ce voyage académique. Merci pour votre présence bienveillante et votre engagement inébranlable.

À **Mr. MESSAOUDI NOUREDDINE**, pour votre rigueur scientifique et votre esprit critique qui ont grandement enrichi cette recherche. Votre participation au jury a apporté une perspective précieuse et une profondeur à ce travail.

Je souhaite également remercier chaleureusement le **Club Scientifique ELECTRO**. Votre environnement inspirant et votre espace de développement ont été le terreau fertile où ce projet a pu prendre racine et s'épanouir. Votre soutien communautaire a été indispensable.

Enfin, je n'omettrai jamais d'exprimer toute ma gratitude à tous les membres du département d'électronique de l'Université de Boumerdes, que ce soit enseignants ou cadres administratifs, qui de près ou de loin n'ont épargné aucun effort pour que ma formation et mes travaux se terminent dans de bonnes conditions.

Dédicaces

Louange à Allah, le Tout-Puissant, sans qui aucun de mes accomplissements ne serait possible. Sa miséricorde et sa guidance m'ont accompagné tout au long de ce parcours.

À ma maman et à mon papa, votre soutien indéfectible et votre aide ont été la pierre angulaire de mon succès. Merci d'avoir enduré chaque moment difficile avec patience et amour, et d'avoir été les piliers sur lesquels je me suis appuyée. Votre dévouement et vos sacrifices sont le fondement de mes réalisations.

À mes amis fidèles : Badji Imed, Brihmat Lisa, Mouhoub Ines, Louisa, Kechichat Ibrahim, Hamici Hazare Sonia, Bouzid Youcef, Hadji Sidali, Sabi Fatiha, Boubidi Youcef, Filali Nasser, Ouissal & Sarah. Chacun de vous est entré dans ma vie à des moments différents, mais vous avez tous été de véritables amis, toujours à mes côtés. Je suis immensément reconnaissante de vous avoir rencontrés. Vous avez illuminé mes jours, m'avez soutenue dans les moments difficiles et avez rendu cette aventure encore plus mémorable.

À tous ceux qui m'ont apporté des leçons de vie, qu'ils aient été des miroirs de fausse bienveillance ou des vecteurs de toxicité, vous m'avez forgée et m'avez permis de devenir la personne résiliente et consciente que je suis aujourd'hui. Votre impact, bien que parfois négatif, a enrichi ma perception de la vie et m'a enseigné la valeur de la véritable intégrité. Pour cela, je vous remercie, car ces leçons ont été des pierres angulaires de mon développement personnel.

À tous ceux qui veulent savoir ce que j'aurais fait si je n'avais pas réussi... je suppose que nous ne le saurons jamais !

Résumé

Cette Mémoire de fin d'études explore et développe le projet NOOR, un traducteur innovant en temps réel de l'arabe au braille utilisant la technologie de l'Internet des objets (IoT). Le projet NOOR répond à un besoin crucial d'outils éducatifs accessibles pour les personnes malvoyantes en Algérie.

En utilisant des microcontrôleurs ESP32, des servomoteurs et un site web personnalisé, NOOR traduit efficacement le texte arabe en braille, affiché via des servomoteurs. La dissertation offre un aperçu historique du braille, discute des défis uniques rencontrés en Algérie et examine en détail les aspects techniques du projet.

Les résultats montrent une amélioration significative de l'accessibilité des ressources éducatives pour les malvoyants en Algérie, mettant l'accent sur l'accessibilité financière et des fonctionnalités arabes enrichies absentes des traducteurs existants.

Ce travail souligne le potentiel vaste de l'IoT dans le développement de technologies d'assistance à la fois économiques et hautement fonctionnelles.

Mots-clés : IoT, ESP32, servomoteurs, braille, Fire base, site web, malvoyants

Abstract

This Graduation memoir explores and develops the NOOR project, an innovative real-time Arabic to Braille translator leveraging Internet of Things (IoT) technology.

The NOOR project addresses the critical need for accessible educational tools tailored for visually impaired individuals in Algeria. By utilizing ESP32 microcontrollers, servo motors, and a custom-built website, NOOR effectively translates Arabic text into Braille, displayed via servo motors.

The dissertation provides a historical overview of Braille, discusses the unique challenges faced in Algeria, and delves into the project's technical intricacies. The results demonstrate a significant enhancement in the accessibility of educational resources for the visually impaired in Algeria, emphasizing affordability and enriched Arabic functionalities absent in existing translators.

This work underscores the broad potential of IoT in developing assistive technologies that are both cost-effective and highly functional.

Keywords: IoT, ESP32, servo motors, Braille, Firebase, website, visually impaired

ملخص

تتناول هذه مذكرة التخرج مشروع نور، وهو مترجم مبتكر يحول النصوص العربية إلى برايل في الوقت الفعلي باستخدام تقنية إنترنت الأشياء. يستجيب مشروع نور للحاجة الملحة لأدوات تعليمية ميسرة لذوي الإعاقة البصرية في الجزائر. يعتمد المشروع على استخدام متحكمات ومركبات سيرفو وموقع ويب مخصص، حيث يقوم بترجمة النصوص العربية إلى برايل وعرضها من خلال مركبات السيرفو. تتضمن الأطروحة استعراضاً تاريخياً لبرائيل، وتناقش التحديات الفريدة التي تواجه الجزائر، وتعرض الجوانب التقنية للمشروع بالتفصيل. تظهر النتائج تحسناً كبيراً في إتاحة الموارد التعليمية للمكفوفين في الجزائر، مع التركيز على توفير الحلول الميسورة التكلفة وتعزيز وظائف اللغة العربية التي تفتقر إليها المترجمات الحالية. يعكس هذا العمل الإمكانيات الواسعة لتقنية إنترنت الأشياء في تطوير تقنيات مساعدة تكون فعالة من حيث التكلفة وعالية الأداء

الكلمات المفتاحية: إنترنت الأشياء، متحكمات، مركبات سيرفو، برايل، فيريبيز، موقع ويب، ذوي الإعاقة البصرية

Table des matières

Remerciements	2
Dédicaces	3
Résumé.....	4
Abstract	5
ملخص.....	6
Table des matières.....	7
Liste des figures.....	10
Liste des tableaux.....	12
Introduction Générale.....	13
Chapitre I : Notions Essentielles du Braille.....	16
I.1 Introduction :.....	17
I.2 Histoire de braille :.....	17
I.3 Dimension de braille :.....	20
I.4 Le Développement du Braille Arabe en Algérie :.....	22
I.4.1 Braille Simplifié (Braille de Grade 1) :.....	22
I.4.2 Braille des Abréviations (Braille de Grade 2) :.....	24
I.4.3 Écriture des Nombres en Braille :.....	27
I.4.4 Codage ASCII :.....	28
I.5 Les Lecteurs de Braille :.....	30
I.5.1 Types de Lecteurs de Braille :.....	30
I.6 Problèmes du Braille en Algérie :.....	33
I.7 Conclusion :.....	34
Chapitre II : Structures Matérielles et Logicielles dans la Conception du Traducteur "NOOR".....	35
II.1 Introduction :.....	36

II.2 Principes de Fonctionnement du Traducteur Braille NOOR :	36
II.2.1 Microcontrôleur ESP32 :	37
II.2.2 Les servo-moteurs :	39
II.3 Connexion et Interaction :	42
II.3.1 l'Internet des Objets (IoT) :	42
II.3.2 L'interface :	44
II.3.3 Firebase :	47
II.4 Les Logiciels et les IDEs :	51
II.4.1 Visual Studio Code :	51
II.4.2 PlatformIO :	55
II.4.3 Wokwi :	61
II.4.4 NVDA (NonVisual Desktop Access):	66
II.4.5 SOLIDWORKS :	69
II.5 Conclusion :	72
Chapitre III : Simulations et Modélisations pour le Traducteur "NOOR"	73
III.1 Introduction :	74
III.2 Simulation de l'Interface Web du Projet NOOR :	74
III.2.1 Objectif de l'Interface Web :	74
III.2.2 Fonctionnement de l'Interface :	74
III.3 Simulation sur Wokwi :	77
III.3.1 Explication du Fonctionnement :	77
III.3.2 Étapes de la Simulation :	78
III.4 Impression 3D avec SolidWorks :	83
III.4.1 Conception des pièces dans SolidWorks :	83
III.4.2 Conversion des conceptions pour l'Impression 3D :	84
III.4.3 Processus impression 3D:	85
III.5 Réalisation Physique de NOOR :	86
III.5.1 Assemblage des Composants :	86
III.5.2 Test du Système :	88
III.6 Conclusion :	88
Conclusion Finale et Perspectives :	89

ANNEXES :	91
Références :	93

Liste des figures

Fig. I. 1	Portrait de Louis Braille, l'inventeur du Braille	18
Fig. I. 2	Structure d'une Cellule Braille	19
Fig. I. 3	Personne lisant le Braille.....	20
Fig. I. 4	Dimensions des Caractères Braille.....	21
Fig. I. 5	Représentation des Chiffres en Braille.....	27
Fig. I. 6	Dispositif OrCam MyEye.....	30
Fig. I. 7	Dispositif Focus Blue 40	31
Fig. I. 8	Dispositif Brilliant BI.....	32
Fig. II. 1	Vue de l'ESP32.....	37
Fig. II. 2	Détails et Ports de l'ESP WROOM32.....	38
Fig. II. 3	Divers Types de Servomoteurs	39
Fig. II. 4	Servomoteur SG90XX	40
Fig. II. 5	Schéma de Connexion entre l'ESP32 et un Servomoteur.....	41
Fig. II. 6	Code contrôle d'un Servomoteur par l'ESP32.....	41
Fig. II. 7	Présentation de l'Internet des Objets (IoT).....	43
Fig. II. 8	Interface Utilisateur de Firebase	48
Fig. II. 9	Base de Données en Temps Réel de Firebase	49
Fig. II. 10	Interface Utilisateur de Visual Studio Code (VSCode).....	52
Fig. II. 11	Cartes Supportées par PlatformIO	56
Fig. II. 12	Interface Utilisateur de PlatformIO	56
Fig. II. 13	Extension Wokwi pour VSCode	62
Fig. II. 14	Fichier de Configuration wokwi.toml.....	64
Fig. II. 15	Exemple de Fichier de Configuration diagram.json	65
Fig. II. 16	Application NVDA Active sur un Ordinateur Portable.....	66
Fig. III. 1	Interface pour activer NVDA	74
Fig. III. 2	Interface du Site Web NOOR	75
Fig. III. 3	Message d'Erreur lors de l'Absence de Fichier et le Clic sur Importer	75
Fig. III. 4	Sélection d'un Fichier .txt.....	76
Fig. III. 5	Message d'Alerte de l'Importation Réussie dans Firebase.....	76
Fig. III. 6	Fichier Reçu sous Forme de Chaîne de Caractères dans Firebase.....	77

Fig. III. 7	Simulation de Circuit dans Wokwi.....	78
Fig. III. 8	Importation de diagram.json de Wokwi vers PlatformIO	79
Fig. III. 9	Fichiers Importants pour la Simulation avec Wokwi dans VSCode.....	79
Fig. III. 10	Compilation du Code.....	80
Fig. III. 11	Réception du Mot NOOR par Firebase depuis le site web	80
Fig. III. 12	Connexion Wi-Fi, servomoteurs et Firebase	81
Fig. III. 13	Détection du Mot par l'ESP32 et Séparation de Chaque Lettre	81
Fig. III. 14	Traduction de Chaque Lettre en Braille par l'ESP32	82
Fig. III. 15	Pièces modélisées avec SolidWorks pour NOOR.....	84
Fig. III. 16	Impression 3D des éléments du projet NOOR.....	85
Fig. III. 17	Soudure manuelle des fils sur le PCB.....	86
Fig. III. 18	Câblage des éléments électroniques	87
Fig. Annexe A :	Programmation de l'ESP32	91
Fig. Annexe B :	Scripts Front-End de l'Interface Web	92

Liste des tableaux

Tableau 1. Dimensions spécifiques du Braille dans différents pays	21
Tableau 2. Lettres Arabes en Braille.....	22
Tableau 3. Braille de Grade 2 avec le Braille à 4 points	25
Tableau 4. Braille de Grade 2 avec le Braille à 5 points	26
Tableau 5. Première Cellule Aléatoire dans le Braille de Grade 2.....	26
Tableau 6. Caractères Arabes en ASCII	29

Introduction Générale

Introduction Générale :

L'être humain possède cinq sens pour explorer et comprendre le monde qui l'entoure. Ces sens sont essentiels pour notre communication et notre perception de l'environnement. Cependant, certaines personnes vivent avec des défis spécifiques liés à l'un de ces sens, notamment la vue. La déficience visuelle oblige ces individus à chercher des méthodes alternatives pour accéder à l'information et communiquer, parmi lesquelles la lecture en Braille est primordiale.

Le Braille, ce système d'écriture tactile ingénieux inventé par Louis Braille, a transformé la vie de millions de personnes aveugles et malvoyantes, leur offrant une voie vers l'indépendance et l'éducation. Pourtant, malgré ses nombreux avantages, le Braille est confronté à plusieurs obstacles. Sur le plan linguistique, les ressources en Braille adaptées aux utilisateurs arabophones sont limitées, ce qui entrave leur accès à une éducation complète. Financièrement, les dispositifs de Braille modernes sont souvent très coûteux, ce qui les rend inaccessibles. De plus, le support technique et l'intégration des différents grades de Braille sont insuffisants, compliquant l'apprentissage et l'utilisation quotidienne.

C'est dans ce contexte que le projet NOOR voit le jour, apportant une lueur d'espoir. NOOR est un prototype de traducteur en temps réel de l'arabe vers le Braille, conçu pour être à la fois abordable et accessible au plus grand nombre. Toujours en développement, NOOR aspire à évoluer pour toucher un public encore plus large et répondre aux besoins spécifiques des personnes aveugles et malvoyantes.

Notre méthodologie est fondée sur la simplicité et l'efficacité. NOOR se présente comme un dispositif de lecture en Braille capable de traduire un texte en code Braille de manière simple, compacte et surtout économique. Pour ce faire, nous utilisons une application de reconnaissance de texte qui envoie le texte à un circuit électronique, permettant l'affichage des caractères transcrits en Braille. Une des solutions clés est

l'utilisation de six actionneurs représentant les six points de la matrice Braille. En nous basant sur des données sur la perception tactile des Algériens, nous ajustons ces actionneurs à une hauteur parfaitement perceptible et contrôlable.

La structure de ce mémoire se compose de trois chapitres principaux. Le premier chapitre, intitulé Les Bases du Braille, retrace l'histoire du Braille, son développement et son adoption en Algérie, ainsi que les spécificités du Braille arabe. Il examine également les différents systèmes de Braille (Grade 1 et 2), les lecteurs de Braille et les défis rencontrés par les utilisateurs en Algérie. Le deuxième chapitre, Logiciels Utilisés, présente les logiciels et outils utilisés dans ce projet, expliquant les raisons de leur choix et leur contribution au succès du projet. Ce chapitre inclut aussi une section sur les simulations réalisées avec Proteus pour tester les configurations électroniques, ainsi qu'une sous-section intitulée Équipement Électronique, détaillant le matériel utilisé, notamment l'ESP32 et les servomoteurs. Le troisième chapitre, Simulation et Réalisation, décrit les étapes de conception et de réalisation du dispositif NOOR, les tests effectués pour garantir son bon fonctionnement, et les résultats obtenus.

Chapitre I : Notions Essentielles du Braille

I.1 Introduction :

Le braille est le langage écrit des personnes malvoyantes, fournissant un moyen de communication essentiel. Selon les données les plus récentes de 2024, l'Organisation Mondiale de la Santé (OMS) estime que 43 millions de personnes dans le monde souffrent de cécité visuelle. En Algérie, cette problématique touche environ 180,000 individus. Ces chiffres mettent en lumière l'urgence et l'importance de développer des outils comme le braille pour améliorer l'accès à l'information et l'autonomie des personnes affectées.

Ce chapitre aborde les notions essentielles du braille, y compris son histoire, ses dimensions et son adaptation à la langue arabe. Il explore également les besoins et les défis auxquels les utilisateurs du braille sont confrontés, en particulier en Algérie. Comprendre ces éléments fondamentaux est crucial pour appréhender les difficultés et les exigences de la communauté malvoyante.

I.2 Histoire de braille :

L'invention de l'écriture spéciale pour les aveugles était l'une des innovations éducatives qui ont contribué efficacement à combler le fossé profond que leur système éducatif connaissait. Désormais, les personnes aveugles pouvaient lire et écrire aussi facilement que les personnes voyantes, mais d'une manière différente, utilisant une méthode d'écriture adaptée à leurs capacités et basée sur leur sens du toucher.

Le mérite de l'invention de cette méthode d'écriture spéciale pour les aveugles revient au Français Louis Braille, en 1824, et elle est devenue finalisée en 1829 [1]. Il s'agit d'une forme d'écriture en relief qui représente les lettres par des points en relief, chaque lettre étant représentée par un ou plusieurs points dans une matrice de six points. L'invention du système Braille a contribué au développement de l'éducation des personnes handicapées visuelles en général, et des aveugles en particulier, et est

créditée pour les progrès réalisés dans le domaine de l'éducation spécialisée pour les personnes handicapées visuelles jusqu'à présent.



Fig. I. 1 Portrait de Louis Braille, l'inventeur du Braille

La plupart de son temps libre, Braille, le passait à réfléchir et à expérimenter. Un matin, alors qu'il lisait le journal avec un ami, il découvrit un article qui stimula son sens et sa pensée. Il s'agissait d'un officier de l'armée française qui avait inventé une méthode permettant aux soldats de recevoir des ordres dans l'obscurité en utilisant des formes en relief sur du papier épais. Chaque forme, composée de douze points au maximum, représentait un ordre militaire [2]. Si un soldat les touchait dans l'obscurité, l'officier comprenait ce que cela signifiait. Braille fut enthousiasmé par cette nouvelle, car il avait trouvé ce qu'il cherchait. Il se rendit immédiatement auprès de l'officier, discuta de la méthode avec lui et en sortit convaincu que les symboles en relief, basés sur des points, étaient plus faciles à appréhender par le toucher. Au fil du temps, Braille adapta ces symboles pour les rendre plus accessibles aux aveugles. Après de nombreuses recherches et réflexions, il utilisa un poinçon pour créer ces symboles, le même poinçon qui lui avait fait perdre la vue, mais qui allait maintenant ouvrir la voie à des millions d'aveugles pour apprendre à lire et à écrire.

Braille parvint à simplifier et à réduire les douze points à six pour faciliter l'apprentissage pour les aveugles. Il utilisa ces six points pour représenter les lettres de l'alphabet, ainsi que les signes mathématiques, musicaux et autres. En travaillant avec ses camarades élèves, ils constatèrent la facilité de lecture de cette méthode, qui leur permettait de noter ce qu'ils voulaient en sciences ou en musique [3].

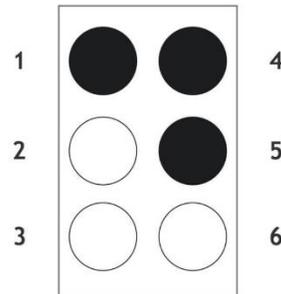


Fig. I. 2 Structure d'une Cellule Braille

Cependant, les enseignants et les administrateurs de l'école étaient opposés à cette méthode d'écriture et de lecture, pensant qu'elle éloignerait les aveugles du monde dans lequel ils vivaient. Ils la combattirent et interdirent aux élèves de l'utiliser, mais ceux-ci continuèrent à l'utiliser en secret et la transmirent à d'autres en dehors de l'école, heureux de son succès. Braille s'efforça de rendre cette méthode officielle, mais il échoua. Il passa sa vie à servir ceux qui avaient perdu le don de la vue, mais sa santé se détériora à cause du stress et du manque de reconnaissance de son travail. Il tomba malade de la tuberculose et mourut en 1852.



Fig. I. 3 Personne lisant le Braille

Quant à l'adaptation du système Braille à la langue arabe, elle fut entreprise par Mohamed Al-Ansi au milieu du XIXe siècle. Il tente de concilier les formes des lettres utilisées dans l'écriture ordinaire avec celles de l'écriture en relief [4]. Bien que Al-Ansi ait traduit plusieurs livres dans cette méthode, elle ne s'est pas largement répandue. Après de nombreuses tentatives, le système Braille a été adapté pour s'adapter à la langue arabe. L'Organisation des Nations unies pour l'éducation, la science et la culture (UNESCO) a joué un rôle dans cette évolution dans les années 1951 [5].

I.3 Dimension de braille :

Lorsqu'on embosse des cellules en Braille sur du papier, il est essentiel de respecter des dimensions spécifiques définies par les normes internationales pour chaque pays. Ces normes, souvent régies par des organisations comme la Braille Authority of North America (BANA) [6], garantissent une uniformité dans la taille et la disposition des points en relief, ce qui facilite la lecture pour les personnes aveugles ou malvoyantes. Le respect de ces dimensions assure une compatibilité avec les dispositifs de lecture Braille et garantit une expérience de lecture fluide et efficace. Le tableau fournit les dimensions requises dans différents pays [7], tandis que la figure illustre les caractéristiques clés des cellules en Braille, désignées par les lettres suivantes :

a : Distance horizontale du centre au centre des points. B : Distance verticale du centre au centre des points. C : Espace d'une cellule à une autre. D : Espace entre les lignes. E : Diamètre de base du point. F : Hauteur du point.

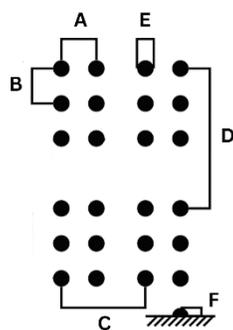


Fig. I. 4 Dimensions des Caractères Braille

Tableau 1. Dimensions spécifiques du Braille dans différents pays

Pays	A (mm)	B (mm)	C (mm)	D (mm)	E (mm)	F (mm)
France	2.5-2.6	2.5-2.6	/	>10	1.2	0.8-1
USA	2.34	2.34	6.22	10.16	1.45	0.48
Allemand	2.5	2.5	6	10	1.3-1.6	0.5
Italie	2.2-2.5	2.2-2.5	/	/	1	0.5
Corée	2	2	5	6	1.5	0.6
Suède	2.5	2.5	6	10	1	0.25
Algérie	NC	NC	NC	NC	NC	NC

I.4 Le Développement du Braille Arabe en Algérie :

L'Algérie, à l'instar de nombreux pays arabes, déploie des efforts considérables pour intégrer les personnes aveugles dans la société, notamment à travers l'enseignement de la langue arabe en Braille [8]. Le Braille arabe est utilisé de gauche à droite et se décline principalement en deux types : le Braille simplifié (ou Braille de grade 1) et le Braille des abréviations (ou Braille de grade 2). Chacun de ces systèmes est conçu pour répondre à différents niveaux de compétence en lecture chez les utilisateurs aveugles, allant des débutants aux plus avancés.

I.4.1 Braille Simplifié (Braille de Grade 1) :

Le Braille simplifié, ou Braille de grade 1, est une transcription directe de l'alphabet arabe en Braille, où chaque lettre est représentée par une configuration unique de six points. Cette méthode assure une correspondance un-à-un entre les lettres arabes et leurs équivalents en Braille, facilitant ainsi l'apprentissage pour les débutants. De plus, certains points supplémentaires sont utilisés pour représenter les voyelles (Fatha, Damma, Kasra) après la lettre et l'accentuation avant celle-ci [9]. Le **tableau 2** illustre le Braille de grade 1, mettant en évidence la disposition des points pour chaque lettre ainsi que les marques supplémentaires pour les voyelles et les accents, offrant un guide visuel complet pour comprendre et utiliser ce système.

Tableau 2. Lettres Arabes en Braille

Lettre arabe	Représentation braille	Lettre arabe	Représentation braille
ا (Alif)	·	ف (Faa)	⠠
ب (Baa)	⠠	ق (Qaf)	⠠
ت (Taa)	⠠	ك (Kaaf)	⠠

ث (Thaa)	∴	ل (Laam)	∴
ج (Jeem)	∴	م (Meem)	∴
ح (Haa)	∴	ن (Noon)	∴
خ (Khaa)	∴	ه (Ha)	∴
د (Daal)	∴	و (Waaw)	∴
ذ (Dhaal)	∴	ى (Alif Maqsura)	∴
ر (Raa)	∴	ة (Ta Marbutah)	∴
ز (Zaa)	∴	ي (Yaa)	∴
س (Seen)	∴	أ (Alif Hamza)	∴
ش (Sheen)	∴	آ (Alif Madda)	∴
ص (Swod)	∴	ئ (Yaa Hamza)	∴
ض (Dhod)	∴	ة (Ta Marbutah)	∴
ط (Taw)	∴	◻ (Fatha)	∴
ظ (Dhaw)	∴	◻ (Dhamma)	∴
ع (Ain)	∴	◻ (Kasra)	∴
غ (Ghain)	∴	◻ (Sukoon)	∴

I.4.2 Braille des Abréviations (Braille de Grade 2) :

Le Braille des abréviations, ou Braille de grade 2, enrichit le Braille simplifié en introduisant plusieurs types d'abréviations. Ces abréviations sont classifiées en fonction des points supplémentaires utilisés, chaque type visant à optimiser la lecture des textes en arabe pour les utilisateurs avancés tout en réduisant les ambiguïtés potentielles :

- Abréviations simples : Elles utilisent la forme standard sans points supplémentaires, facilitant ainsi une lecture fluide et rapide pour les mots extrêmement communs.
- Abréviations avec Point 4 : Ces abréviations incluent le point 4, employées pour des mots moins fréquents qui nécessitent une distinction des abréviations plus simples.
- Abréviations avec Point 5 : Avec l'ajout du point 5, cette catégorie est destinée aux mots ou expressions rares, importants pour la clarté du contexte.
- Abréviations avec Points 4 et 5 : Combinant les points 4 et 5, utilisées pour les termes qui ne peuvent être clairement définis par des configurations moins complexes.
- Abréviations avec Points 4, 5 et 6 : Ces abréviations, intégrant les points 4, 5 et 6, sont les plus spécifiques et utilisées pour des termes ou contextes très particuliers.
- Abréviations aléatoire (2 cellules) : Pour les mots ou expressions particulièrement longs ou complexes, deux cellules de Braille sont nécessaires. Cette méthode gère efficacement les mots qui ne peuvent pas être abrégés en une seule cellule.

Pour comprendre concrètement ces explications, les tableaux **Tableau 3**, **Tableau 4**, **Tableau 5** montrent des exemples pour quelques types d'abréviations. Ces tableaux fournissent des exemples visuels pour aider à comprendre comment chaque type

d'abréviation est utilisé dans la pratique, rendant ainsi l'apprentissage du Braille de grade 2 plus accessible et efficace :

Tableau 3. Braille de Grade 2 avec le Braille à 4 points

Mot arabe	Représentation braille	Numéros de points
صَدِيق □	⠠⠠⠠⠠⠠⠠	4 + 1 2 3 4 6
عَاذَة □	⠠⠠⠠⠠⠠⠠	4 + 1 2 3 5 6
فَرَاغ □	⠠⠠⠠⠠⠠	4 + 1 2 4
كِتَاب □	⠠⠠⠠⠠	4 + 1 3
مُخْوَف □	⠠⠠⠠⠠⠠	4 + 1 3 4
لَا سِيْمَا	⠠⠠⠠⠠⠠	4 + 1 2 3 6
ضَعِيف □	⠠⠠⠠⠠⠠⠠	4 + 1 2 4 6
يَسِير □	⠠⠠⠠⠠	4 + 2 4
مُؤْمِن □	⠠⠠⠠⠠⠠⠠	4 + 1 2 5 6
غَاذِر □	⠠⠠⠠⠠⠠⠠	4 + 1 2 6
أَكْبَر □	⠠⠠⠠⠠⠠	4 + 3 4
أَعْلَى	⠠⠠⠠⠠⠠⠠	4 + 1 3 5
أَدَاب □	⠠⠠⠠⠠⠠⠠	4 + 3 4 5
يَسِير □	⠠⠠⠠⠠	4 + 2 4

Tableau 4. Braille de Grade 2 avec le Braille à 5 points

Mot arabe	Représentation braille	Numéro De point	Mot arabe	Représentation braille	Numéro De point
سريع □	⠠⠠	5 + 1 2 3 5	لا شك □□	⠠⠠	5 + 1 2 3 6
طريق □	⠠⠠	5 + 1 2 3 5 6	كثير □	⠠⠠	5 + 1 3
دائم □	⠠⠠	5 + 1 4 5	عزيز □	⠠⠠	5 + 2 3 4 5 6
زمان □	⠠⠠	5 + 1 3 5 6	فوق □	⠠⠠	5 + 1 2 4
قليل □	⠠⠠	5 + 1 2 3 4 5	نفس □	⠠⠠	5 + 1 3 4 5
أيام □	⠠⠠	5 + 2 4	شديد □	⠠⠠	5 + 1 4 6
ساعة □	⠠⠠	5 + 1 6	غير □	⠠⠠	5 + 1 2 6

Tableau 5. Première Cellule Aléatoire dans le Braille de Grade 2

Mot arabe	Représentation braille	Numéro De point	Mot arabe	Représentation braille	Numéro De point
إنتاج □	⠠⠠	6 + 2 4 6	تجربة □	⠠⠠	2 3 + 1 2 3 4
إخراج □	⠠⠠	5 6 + 2 4 6	ينتخب □	⠠⠠	5 6 + 3 4 6
ينعكس □	⠠⠠	6 + 3 4 6	هامش □	⠠⠠	6 + 1 2 4 5 6
نفسية □	⠠⠠	1 6 + 1 2 3 4 5	هيئة □	⠠⠠	5 6 + 1 2 4 5 6
ما دام □	⠠⠠	5 6 + 1 2 3 4	إمتحان □	⠠⠠	1 6 + 1 2 3 4 5
ما إنفك □□	⠠⠠	6 + 1 2 3 4	ألتي □	⠠⠠	5 6 + 3 4

I.4.3 Écriture des Nombres en Braille :

Contrairement aux lettres, les nombres en Braille sont introduits par un symbole spécifique (point 3-4-5-6) avant chaque chiffre pour les distinguer clairement du texte alphabétique. Ce système aide les lecteurs à identifier rapidement les séquences numériques dans un texte [10]. La figure **Fig. I. 5** explique comment les nombres sont représentés en Braille, fournissant un guide visuel pour leur apprentissage et leur utilisation.

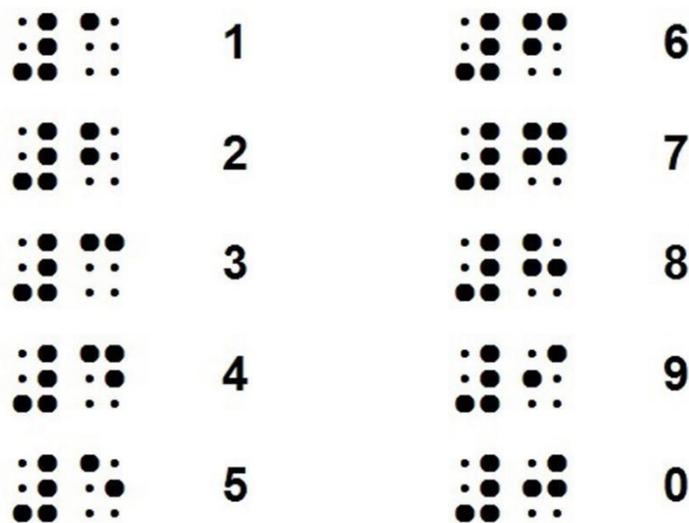


Fig. I. 5 Représentation des Chiffres en Braille

L'adoption du Braille en arabe en Algérie marque une étape importante dans l'intégration sociale et l'indépendance des personnes aveugles. Toutefois, plusieurs défis subsistent, tels que la formation adéquate des enseignants spécialisés, la création et distribution de matériel didactique en Braille, ainsi que la sensibilisation à l'importance de ces mesures. Il est crucial de continuer ces efforts pour assurer une éducation inclusive et accessible à tous.

I.4.4 Codage ASCII :

L'American Standard Code for Information Interchange (ASCII) est un standard de codage de caractères développé dans les années 1960 pour standardiser la communication textuelle entre les ordinateurs. Utilisant un codage sur 7 bits, ASCII permet de représenter 128 caractères différents, incluant les lettres majuscules et minuscules de l'alphabet anglais, les chiffres, les signes de ponctuation et plusieurs caractères de contrôle [11]. Bien que cette norme ait été suffisante pour les premières applications informatiques centrées sur l'anglais, elle s'est rapidement révélée limitée pour représenter les caractères d'autres langues.

Pour pallier les limites de l'ASCII, des extensions ont été développées. Le plus courant est l'ASCII étendu, qui utilise 8 bits pour permettre la représentation de 256 caractères, incluant des lettres accentuées et des symboles spécifiques à diverses langues européennes [12]. Toutefois, cette extension restait insuffisante pour les systèmes d'écriture plus complexes comme l'arabe.

L'arabe, avec ses 28 lettres de base, ses formes contextuelles et ses diacritiques, pose des défis particuliers à la représentation en ASCII. Dans un système d'écriture bidirectionnel où la direction du texte alterne entre droite-gauche pour l'arabe et gauche-droite pour les chiffres et les mots étrangers, la simple extension de l'ASCII ne pouvait pas suffire. Divers encodages spécifiques ont été développés pour l'arabe avant l'adoption de l'Unicode, comme l'ISO 8859-6, également connu sous le nom d'Alphabet Latin/Arabe. Cependant, ces encodages étaient limités et souvent incompatibles entre eux. L'Unicode est devenue la solution universelle à ces problèmes. Introduit dans les années 1990, Unicode utilise un système de codage sur 16 bits (ou plus) qui permet de représenter plus de 65 000 caractères, couvrant pratiquement toutes les langues du monde [13], y compris l'arabe. Avec Unicode, chaque caractère a attribué un point de code unique, éliminant les problèmes d'incompatibilité et permettant une

représentation cohérente des textes multilingues. Voici un tableau comparatif des codes ASCII et Unicode pour quelques caractères arabes de base :

Tableau 6. Caractères Arabes en ASCII

Caractère Arabe	Unicode (ASCII)	Caractère Arabe	Unicode (ASCII)
ا	U+0627	غ	U+063A
ب	U+0628	ف	U+0641
ت	U+062A	ق	U+0642
ث	U+062B	ك	U+0643
ج	U+062C	ل	U+0644
ح	U+062D	م	U+0645
خ	U+062E	ن	U+0646
د	U+062F	هـ	U+0647
ذ	U+0630	و	U+0648
ر	U+0631	ي	U+064A
ز	U+0632	ء	U+0621
س	U+0633	ة	U+0629
ش	U+0634	ى	U+0649
ص	U+0635	أ	U+0623
ض	U+0636	إ	U+0625
ط	U+0637	آ	U+0622

ظ	U+0638	ئ	U+0626
ع	U+0639	ؤ	U+0624

I.5 Les Lecteurs de Braille :

L'accès à l'information est essentiel pour l'autonomie et l'intégration sociale des personnes aveugles ou malvoyantes. Les lecteurs de Braille, qui convertissent le texte imprimé et numérique en Braille ou en parole synthétisée, jouent un rôle crucial dans l'atteinte de cet objectif. En examinant les différentes technologies de lecteurs de Braille, il est possible de comprendre leur impact et leur importance dans la vie quotidienne des utilisateurs.

I.5.1 Types de Lecteurs de Braille :

I.5.1.1 Lecteurs de Braille avec OCR et Caméra :

Les dispositifs tels que l'Orcam MyEye sont dotés d'une technologie avancée qui combine une caméra et un logiciel de reconnaissance optique de caractères (OCR) pour numériser et convertir le texte imprimé en sortie audible ou en Braille. Cette technologie permet aux utilisateurs de lire de manière autonome une grande variété de documents imprimés, des livres aux panneaux d'affichage, offrant ainsi une liberté et une indépendance considérables. La figure **Fig. I. 6** est une photo de ce modèle.



Fig. I. 6 Dispositif OrCam MyEye

Ces lecteurs de Braille sont particulièrement bénéfiques pour les personnes aveugles ou malvoyantes qui veulent accéder rapidement à l'information sans assistance extérieure. Cependant, le coût de ces technologies peut être prohibitif, et leur efficacité peut varier selon l'environnement lumineux, les textes à fort contraste étant parfois nécessaires pour garantir une reconnaissance précise. De plus, dans des conditions de faible luminosité, leur performance peut être compromise, ce qui peut limiter leur utilité dans certains contextes.

I.5.1.2 Lecteurs de Braille avec Entrée Bluetooth :

Des modèles tels que le Focus Blue 40 utilisent la technologie Bluetooth pour se connecter à divers appareils numériques, facilitant ainsi la transmission du contenu textuel directement en Braille. Ces lecteurs sont extrêmement utiles pour les personnes qui se déplacent fréquemment, car ils permettent un accès facile et immédiat à l'information numérique, que ce soit pour la navigation dans des menus ou la lecture de documents électroniques. La figure **Fig. I. 7** est une photo de ce modèle. L'avantage majeur de ces appareils est leur portabilité et leur flexibilité, rendant possible la lecture en Braille en déplacement sans la nécessité de supports physiques. Toutefois, ces dispositifs exigent une gestion attentive de la batterie pour éviter les interruptions dans l'usage, et ils dépendent de la compatibilité avec d'autres appareils, ce qui peut parfois poser des défis en termes de connectivité et de compatibilité logicielle.



Fig. I. 7 Dispositive Focus Blue 40

I.5.1.3 Lecteurs de Braille avec Entrée USB :

Le Brailiant BI est un exemple de lecteur de Braille qui se connecte via USB à un ordinateur pour offrir un accès direct aux informations affichées sur l'écran. Ce type de lecteur est particulièrement adapté aux environnements professionnels ou éducatifs où l'accès fréquent à des informations numériques est nécessaire. La figure **Fig. I. 8** est



Fig. I. 8 Dispositif Brilliant BI

une photo de ce modèle. Il permet aux utilisateurs de suivre les cours ou les réunions en temps réel, facilitant ainsi l'intégration dans des environnements principalement visuels. Bien que ces lecteurs offrent une stabilité et une fiabilité importantes, ils nécessitent une source d'alimentation constante, ce qui peut limiter leur utilisation à des espaces de travail ou d'étude fixes. De plus, la nécessité d'une connexion physique par câble peut réduire leur flexibilité et leur portabilité, rendant difficile leur utilisation en dehors de ces environnements préétablis.

I.6 Problèmes du Braille en Algérie :

Le Braille, bien qu'essentiel pour les personnes aveugles et malvoyantes, présente de nombreux défis. L'un des principaux problèmes est le concept de l'illettrisme chez les personnes âgées. Apprendre le Braille à un âge avancé peut s'avérer extrêmement difficile, voire impossible, en raison de la diminution de la plasticité cognitive et de la patience nécessaires pour maîtriser un nouveau système de lecture.

En outre, le Braille repose sur la sensibilité tactile, ce qui pose un autre défi pour les personnes souffrant de diabète. Le diabète peut endommager les nerfs, entraînant une perte de sensibilité tactile, rendant ainsi la lecture du Braille très difficile, voire impossible.

Ajoutons à cela les personnes atteintes d'arthrite, qui peuvent avoir du mal à utiliser les lecteurs de Braille en raison de la douleur et de la raideur dans leurs articulations, ce qui limite leur capacité à lire efficacement. Par ailleurs, pour les personnes autistes ou ayant d'autres handicaps mentaux ou physiques, l'utilisation du Braille peut également être problématique. Les difficultés de coordination et les troubles de la motricité fine peuvent rendre la manipulation des appareils de Braille ardue, et les déficiences cognitives peuvent compliquer l'apprentissage et l'utilisation de ce système.

En Algérie, les lecteurs de Braille, bien qu'ils enrichissent considérablement la vie des personnes aveugles ou malvoyantes en améliorant leur accès à l'information, rencontrent également des défis spécifiques. Le nombre de dispositifs disponibles est très limité, et leur coût est extrêmement élevé, ce qui les rend inaccessibles à la majorité des personnes dans le besoin.

Pour aggraver la situation, les appareils disponibles ne prennent souvent pas en charge le Braille arabe de grade 2 et sont principalement accessibles dans des centres spécialisés, ce qui limite leur utilité générale. Ces obstacles soulignent le besoin urgent de développer des technologies de lecture Braille plus accessibles et adaptées aux besoins locaux en Algérie. Une telle innovation permettrait de surmonter les barrières

financières, d'améliorer le support linguistique et de rendre la lecture en Braille plus inclusive pour tous, indépendamment de leurs conditions physiques ou mentales.

I.7 Conclusion :

Ce chapitre a fourni un aperçu exhaustif du système Braille en langue arabe en Algérie, mettant en lumière les dispositifs essentiels pour l'autonomie des personnes aveugles et les défis à surmonter pour une intégration plus efficace. L'analyse des différents types de lecteurs de Braille et des méthodes d'enseignement spécifiques révèle un besoin crucial pour des ressources pédagogiques adaptées et une meilleure formation des éducateurs.

En tenant compte des limites financières et des ressources disponibles en Algérie, ce chapitre sert de fondement à la conception d'un prototype de lecteur de Braille en temps réel abordable. Un tel dispositif pourrait transformer radicalement l'accès à l'information pour les personnes aveugles, leur permettant de participer plus activement à la société.

Chapitre II : Structures Matérielles et Logicielles dans la Conception du Traducteur "NOOR"

II.1 Introduction :

Le développement du traducteur "NOOR" repose sur une combinaison innovante de structures matérielles et logicielles. Ce chapitre explore en profondeur les composants matériels et les outils logiciels essentiels à la conception et à la réalisation de ce projet. En intégrant des technologies avancées de l'Internet des objets (IoT) et des solutions logicielles robustes, NOOR vise à offrir une solution accessible et efficace pour la traduction en temps réel de l'arabe en Braille. L'objectif est de créer un dispositif qui répond aux besoins spécifiques des personnes malvoyantes, tout en étant simple d'utilisation et abordable.

II.2 Principes de Fonctionnement du Traducteur Braille NOOR :

Le dispositif de lecture Braille "NOOR" comprend une configuration précise des composants essentiels, comme montré dans figure suivante. Parmi ces composants, on trouve un module ESP32, des servomoteurs, et une interface web.

Au centre du système, l'ESP32 gère les communications et le traitement des données. Les servomoteurs, éléments centraux de l'appareil, facilitent la représentation des caractères Braille en activant les matrices selon les traductions réalisées. L'interface web est conçue pour être fonctionnelle et facile d'accès, spécialement pour les enseignants en Algérie, leur permettant d'envoyer des textes, des fichiers PDF et des images. Ces données sont reçues par l'ESP32 qui les transforme en langage Braille, pilotant les servomoteurs pour afficher avec précision les caractères sur les matrices Braille. Nous examinerons maintenant plus en détail les divers composants de notre lecteur Braille "NOOR" et les logiciels qui en assurent le fonctionnement.

II.3 Équipements Électroniques :

Dans cette partie, nous allons examiner en détail les différents composants électroniques nécessaires au bon fonctionnement de NOOR. Nous aborderons leur rôle, leur importance dans le système, et comment ils interagissent entre eux pour assurer une performance optimale et une utilisation efficace de l'appareil.

II.2.1 Microcontrôleur ESP32 :

L'ESP32 est un microcontrôleur hautement performant doté de fonctionnalités sans fil, incluant le Wi-Fi et le Bluetooth, ce qui le rend particulièrement adapté aux projets nécessitant des communications à distance (La figure **Fig. II. 1**). Conçu pour allier puissance et rentabilité, cet élément clé répond parfaitement aux exigences des développeurs cherchant à concevoir des applications IoT (Internet des Objets) innovantes tout en maîtrisant les coûts. Des illustrations de l'ESP32 sont ajoutées ici pour mieux visualiser son design et ses composants matériels.



Fig. II. 1 Vue de l'ESP32

Le choix de l'ESP32 pour le dispositif NOOR s'explique principalement par ses capacités de connexion, essentielles pour la réception de données en continu depuis un serveur. Cette fonctionnalité est indispensable pour la traduction instantanée de l'arabe vers le Braille. De plus, la robustesse de son processeur permet de gérer simultanément la traduction des textes et le pilotage des servomoteurs qui impriment les caractères en Braille. Son excellent rapport qualité-prix rend l'ESP32 particulièrement attrayant pour des projets nécessitant à la fois une connectivité étendue et une grande performance, sans pour autant sacrifier la gestion du budget.

Ce module (La figure Fig. II. 2) est équipé d'un processeur dual-core qui peut atteindre une fréquence de 240 MHz et dispose de 520 KB de mémoire SRAM. Il supporte une large gamme de fonctionnalités d'entrées/sorties. Pour notre projet NOOR, ces caractéristiques permettent une gestion efficace des flux de données textuelles et leur conversion en signaux précis pour les servomoteurs.

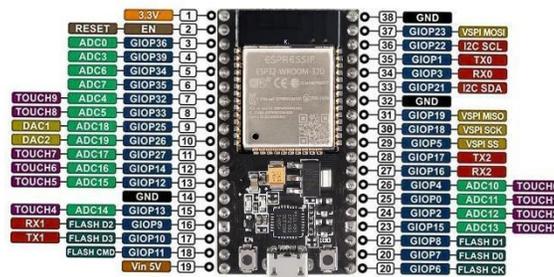


Fig. II. 2 Détails et Ports de l'ESP WROOM32

La capacité du microcontrôleur à opérer avec des systèmes à faible consommation d'énergie est également un atout majeur, offrant au système une plus grande durabilité et fiabilité, même en mode autonome. Grâce à ses multiples fonctionnalités, le cœur de notre système s'avère être un choix judicieux pour les applications qui nécessitent une

fusion sophistiquée entre matériel et logiciel, tout en conservant une efficacité énergétique et une adaptabilité exceptionnelle dans la gestion des données.

II.2.2 Les servo-moteurs :

Les servomoteurs sont des dispositifs électromécaniques qui permettent un contrôle précis de la position angulaire, de la vitesse et de l'accélération (La figure **Fig. II. 3**). Ils sont largement utilisés dans diverses applications, allant des robots aux systèmes de contrôle industriels, en passant par les dispositifs de positionnement.



Fig. II. 3 Divers Types de Servomoteurs

Pour le projet NOOR, nous avons choisi d'utiliser le module SG90, un servomoteur populaire connu pour sa fiabilité et son efficacité dans les projets électroniques.

Le choix des servomoteurs SG90 pour NOOR a été déterminé par plusieurs critères essentiels. Premièrement, le couple (torque) est crucial pour garantir que le servomoteur peut déplacer et maintenir les composants du dispositif Braille en position sans difficulté. Le SG90, avec son couple de 1,8 kg/cm à 4,8V, répond parfaitement à nos exigences. Deuxièmement, la vitesse à laquelle le servomoteur peut atteindre la position désirée est primordiale pour assurer une traduction rapide et efficace du texte en Braille. Le SG90 offre une vitesse de 0,1 s/60 degrés à 4,8V, adéquate pour notre

application. De plus, la précision de positionnement est vitale pour garantir que chaque caractère Braille soit correctement représenté. Le SG90XX (La figure **Fig. II. 4**) est connu pour sa précision et sa fiabilité, avec une résolution d'angle de 1 degré. Enfin, la fiabilité est un critère indispensable dans un projet éducatif comme NOOR, où l'appareil sera utilisé régulièrement. Le SG90XX est largement reconnu pour sa durabilité et ses performances constantes, ce qui en fait un choix idéal pour notre projet.

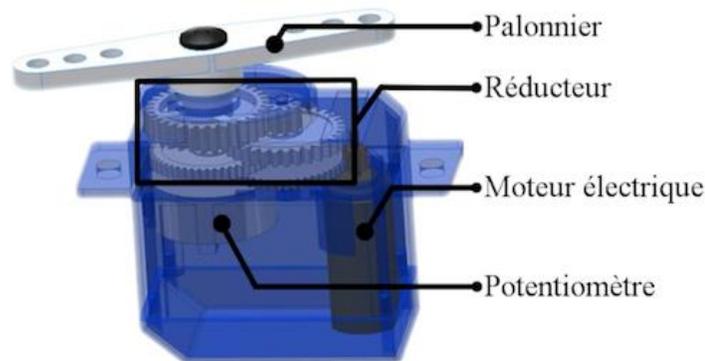


Fig. II. 4 Servomoteur SG90XX

Les servomoteurs SG90XX sont contrôlés à l'aide de l'ESP32, notre microcontrôleur central. Pour contrôler les servomoteurs, nous utilisons des bibliothèques Arduino telles que <Servo.h>, qui simplifient grandement la gestion des servos. Ces bibliothèques fournissent des fonctions prêtes à l'emploi pour définir les angles de positionnement et gérer les commandes nécessaires pour le contrôle des servomoteurs.

Pour connecter un servomoteur SG90XX à l'ESP32, il faut suivre ces étapes (voir La figure **Fig. II. 5**) :

Connectez le fil de signal (généralement de couleur orange) du servomoteur à l'une des broches PWM de l'ESP32.

Connectez le fil d'alimentation (rouge) à une source de 5V.

Connectez le fil de masse (noir ou marron) à la broche GND de l'ESP32.

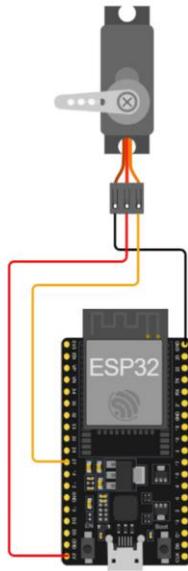


Fig. II. 5 Schéma de Connexion entre l'ESP32 et un Servomoteur

Voici un extrait de code (la figure Fig. II. 6) montrant comment initialiser et contrôler un servomoteur SG90 avec l'ESP32 :

```
1  #include <Arduino.h>
2  #include <ESP32Servo.h>
3  Servo ABDELLI;
4  void setup (){
5  ABDELLI.attach (9);
6  }
7  void loop (){
8  ABDELLI.write(66);
9  delay(20);
10 }
```

Fig. II. 6 Code contrôle d'un Servomoteur par l'ESP32

II.3 Connexion et Interaction :

II.3.1 l'Internet des Objets (IoT) :

L'Internet des Objets (IoT, ou Internet of Things en anglais) désigne l'ensemble des objets physiques connectés à Internet, capables de collecter et d'échanger des données. Ces objets peuvent être des appareils domestiques, des capteurs industriels, des véhicules, ou même des dispositifs médicaux [14]. Ce concept permet une interconnectivité et une automatisation accrue, transformant ainsi notre interaction avec l'environnement. Actuellement, cette technologie joue un rôle crucial en améliorant l'efficacité, en offrant de nouvelles opportunités d'innovation et en facilitant la création de systèmes intelligents et réactifs.

Cette interconnexion a un impact significatif sur les technologies d'assistance, notamment en améliorant les dispositifs destinés aux personnes malvoyantes. Grâce à cette technologie, ces dispositifs peuvent offrir des fonctionnalités avancées qui améliorent l'autonomie et la qualité de vie des utilisateurs [15]. Par exemple, des capteurs connectés peuvent détecter des obstacles et fournir des retours en temps réel, permettant ainsi aux utilisateurs de se déplacer plus facilement et en toute sécurité. De plus, les dispositifs d'assistance peuvent être intégrés à des systèmes plus larges, tels que les maisons intelligentes [16], pour offrir une assistance personnalisée et automatisée.

Dans notre projet NOOR, cette technologie joue un rôle central en utilisant la connectivité Wi-Fi pour recevoir des données en temps réel depuis un serveur. Ce dispositif repose sur une interface web personnalisée où les enseignants peuvent envoyer du texte, des fichiers PDF et des images. Ces données sont ensuite traitées et converties en caractères Braille par l'ESP32, permettant une traduction instantanée pour les utilisateurs malvoyants. Cette approche nécessite un serveur pour gérer et transmettre les données, assurant ainsi une performance fluide et efficace.

L'intégration de cette technologie dans NOOR apporte de nombreux avantages. Tout d'abord, elle permet une traduction en temps réel de l'arabe vers le Braille, rendant l'information plus accessible et immédiatement disponible pour les utilisateurs malvoyants. Ensuite, en utilisant des technologies de connectivité telles que le Wi-Fi, NOOR offre une flexibilité d'utilisation et une facilité d'accès via une interface web personnalisée. Enfin, cette interconnexion permet une interaction fluide entre les différents composants du système, assurant une performance optimale et une expérience utilisateur améliorée.



Fig. II. 7 Présentation de l'Internet des Objets (IoT)

En somme, l'Internet des Objets révolutionne les technologies d'assistance en offrant des solutions plus intelligentes, interconnectées et personnalisées, qui répondent mieux aux besoins des utilisateurs malvoyants. Les dispositifs assistés par cette technologie,

tels que NOOR, illustrent parfaitement comment ces avancées peuvent transformer et améliorer la vie quotidienne des personnes en situation de handicap.

II.3.2 L'interface :

Une interface est avant tout la couche intermédiaire entre un utilisateur et un système complexe, rendant l'interaction intuitive et efficace. Ce concept inclut une variété d'interfaces, comme les interfaces graphiques sur les ordinateurs, les écrans tactiles sur les appareils mobiles, et même les commandes vocales dans les systèmes de maison intelligente. Pour comprendre pleinement l'importance des interfaces, il est essentiel de distinguer entre l'Interface Utilisateur (UI) et l'Expérience Utilisateur (UX).

II.3.2.1 Interface Utilisateur (UI) :

L'Interface Utilisateur concerne tous les éléments visuels et interactifs qu'un utilisateur voit et avec lesquels il interagit. Les aspects clés de l'UI comprennent :

- **Design Visuel :** L'apparence générale de l'interface, y compris les couleurs, les typographies, les icônes et les mises en page. Un design visuel attrayant et cohérent attire l'utilisateur et rend l'interface plaisante à utiliser.
- **Disposition des Éléments :** La façon dont les différents éléments (boutons, menus, formulaires) sont arrangés sur l'écran. Une disposition bien pensée facilite la navigation et permet à l'utilisateur de trouver rapidement ce qu'il cherche.
- **Composants Interactifs :** Les éléments avec lesquels l'utilisateur interagit directement, comme les boutons, les champs de texte et les curseurs. Ces composants doivent être clairement identifiables et faciles à utiliser.

II.3.2.2 Expérience Utilisateur (UX) :

L'Expérience Utilisateur va au-delà de l'apparence visuelle pour englober l'ensemble du parcours utilisateur. Elle vise à créer des interactions fluides et agréables, en se concentrant sur les aspects suivants :

- **Utilisabilité** : La facilité avec laquelle les utilisateurs peuvent accomplir leurs tâches. Une bonne UX minimise les obstacles et les frustrations, permettant à l'utilisateur d'atteindre ses objectifs rapidement et efficacement.
- **Accessibilité** : La capacité de l'interface à être utilisée par des personnes de toutes capacités, y compris celles ayant des handicaps. Cela inclut l'intégration de fonctionnalités comme les lecteurs d'écran et les commandes vocales.
- **Satisfaction Utilisateur** : Le degré de contentement des utilisateurs après avoir utilisé le système. Une bonne UX se traduit par des utilisateurs satisfaits, prêts à utiliser l'interface et à la recommander à d'autres.
- **Flux de travail** : La logique et la fluidité des étapes que l'utilisateur doit suivre pour accomplir ses tâches. Un bon flux de travail est intuitif et évite les chemins compliqués ou contre-intuitifs.

II.3.2.3 Intégration UI/UX dans les Interfaces Modernes :

Les interfaces modernes sur les ordinateurs, les appareils mobiles et les systèmes de maison intelligente illustrent bien l'intégration harmonieuse de l'UI et de l'UX :

- **Interfaces graphiques** : Sur les ordinateurs, les interfaces graphiques permettent une navigation visuelle et interactive à travers des fenêtres, des icônes et des menus déroulants.
- **Écrans tactiles** : Sur les appareils mobiles, les écrans tactiles offrent une interaction directe et naturelle, avec des gestes comme le pincement pour zoomer ou le balayage pour naviguer.
- **Commandes Vocales** : Dans les systèmes de maison intelligente, les commandes vocales permettent un contrôle sans effort des appareils, en interprétant et en exécutant les instructions vocales de l'utilisateur.

En comprenant et en appliquant les principes d'UI et d'UX, les designers peuvent créer des interfaces qui ne sont pas seulement belles, mais aussi fonctionnelles,

accessibles et agréables à utiliser.

II.3.2.4 Principes de conception des interfaces :

Pour qu'une interface soit réussie, elle doit respecter certains principes fondamentaux :

- **Intuitivité** : L'interface doit être facile à comprendre et à utiliser sans nécessiter de longues instructions. Les utilisateurs doivent pouvoir naviguer et accomplir des tâches sans effort.
- **Réactivité** : Il est essentiel que l'interface fournisse un retour immédiat aux actions des utilisateurs pour éviter toute confusion et maintenir une expérience utilisateur fluide.
- **Accessibilité** : L'interface doit être utilisable par des personnes de toutes capacités, y compris celles ayant des handicaps. Cela inclut des fonctionnalités comme les lecteurs d'écran, les contrastes élevés et les commandes vocales.
- **Consistance** : Les éléments de l'interface doivent être cohérents dans leur fonctionnement et leur apparence à travers toutes les sections du système. Cette consistance facilite l'apprentissage et la mémorisation par les utilisateurs.
- **Simplicité** : L'interface doit éviter toute complexité inutile. Les actions les plus courantes doivent être simples et rapides à exécuter, permettant une navigation efficace.
- **Transparence** : Les utilisateurs doivent comprendre ce qui se passe après avoir initié une action. Si une action prend du temps, un indicateur de progression doit être affiché pour informer l'utilisateur.
- **Feedback efficace** : Chaque action doit être accompagnée d'un retour approprié, confirmant que le système a bien reçu la commande et qu'il est en train de la traiter.

II.3.2.5 Outils et applications pour créer des interfaces :

Pour concevoir des interfaces utilisateur de qualité, plusieurs outils sont disponibles :

- **Adobe XD** : Utilisé pour le design d'interfaces utilisateur (UI) et l'expérience utilisateur (UX), cet outil permet de créer des prototypes interactifs.
- **Figma** : Une plateforme collaborative en ligne qui permet aux designers de créer, tester et déployer des interfaces utilisateur rapidement et de manière collaborative.
- **Sketch** : Très populaire parmi les professionnels du design UX/UI, cet outil est utilisé pour la création de maquettes et de wireframes.
- **Axure RP** : Un outil avancé pour créer des prototypes d'interfaces nécessitant des interactions complexes et des simulations fonctionnelles.

L'interface web du projet NOOR est conçue en utilisant ces principes et outils pour maximiser son accessibilité et sa fonctionnalité. Elle permet une intégration transparente dans différents environnements, offrant aux utilisateurs la possibilité de se connecter au système depuis n'importe quel dispositif connecté à Internet. L'adoption d'une telle interface garantit que les mises à jour peuvent être déployées rapidement, que le système peut évoluer avec les besoins des utilisateurs, et que l'accès est universel, supprimant les barrières physiques et technologiques pour l'utilisation de la technologie de traduction Braille.

II.3.3 Firebase :

Firebase est une plateforme de développement d'applications proposée par Google, qui fournit une suite complète de services backend tels que l'authentification, la base de données en temps réel, le stockage, et les fonctions cloud. Conçue pour simplifier et accélérer le développement d'applications web et mobiles, Firebase permet aux développeurs de se concentrer sur l'expérience utilisateur tout en offrant une infrastructure robuste et scalable.

II.3.3.1 Interface de Firebase :

L'interface de Firebase est intuitive et bien organisée, facilitant la navigation et la gestion des différents services. Elle est divisée en quatre sections principales situées sur la barre latérale gauche (voir la figure **Fig. II. 8**) :

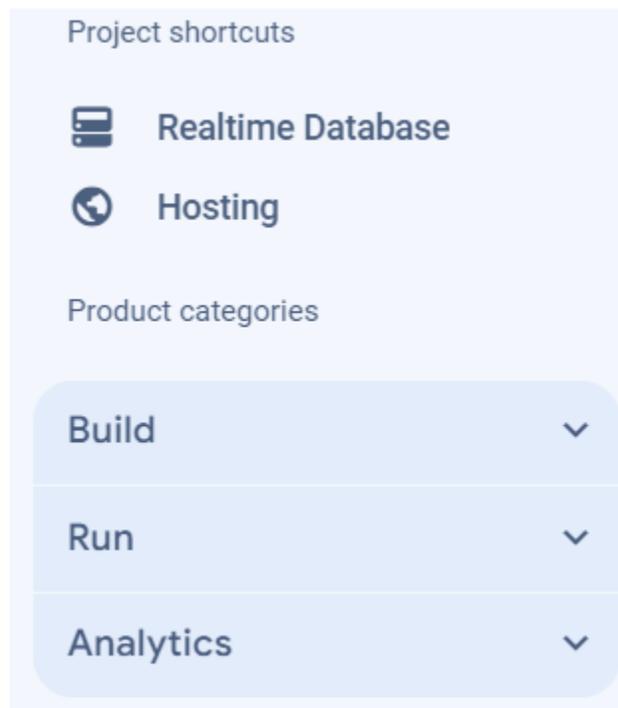


Fig. II. 8 Interface Utilisateur de Firebase

- **Build** : Cette section regroupe les services de base de Firebase tels que l'authentification, la base de données (Realtime Database et Firestore), le stockage, et l'hébergement. Ces outils permettent de créer et de gérer les composants essentiels de votre application.
- **Run** : Ici, vous trouverez les outils nécessaires pour exécuter et surveiller votre application, y compris Cloud Functions, le monitoring des performances, et l'A/B testing. Ces services assurent que votre application fonctionne efficacement et peut être

facilement mise à jour.

- **Analytics** : Firebase Analytics offre une suite d'outils puissants pour analyser l'utilisation de votre application et le comportement des utilisateurs. Vous pouvez suivre des événements spécifiques, analyser les entonnoirs de conversion, et segmenter vos utilisateurs pour des insights détaillés.
- **Project Shortcuts** : Cette section comprend des raccourcis vers les fonctionnalités fréquemment utilisées et les paramètres du projet, facilitant un accès rapide et une gestion efficace de vos ressources Firebase.

II.3.3.2 Firebase Realtime Database :

Firebase Realtime Database est une base de données NoSQL hébergée dans le cloud qui permet de stocker et de synchroniser les données entre les clients en temps réel. Les modifications apportées à la base de données sont immédiatement reflétées chez tous les utilisateurs connectés, ce qui est idéal pour les applications nécessitant des mises à jour instantanées (voir la figure **Fig. II. 9**) :



Fig. II. 9 Base de Données en Temps Réel de Firebase

Fonctionnalités clés :

- **Synchronisation en temps réel** : Les données sont synchronisées en temps réel entre tous les clients, offrant une expérience utilisateur fluide et réactive.

- **Structure de données JSON** : Les données sont stockées sous forme de JSON, permettant une gestion flexible et dynamique des informations.

- **Sécurité et règles de validation** : Firebase offre des règles de sécurité robustes et des mécanismes de validation pour protéger les données et contrôler l'accès.

II.3.3.3 Intégration de Firebase avec les Projets IoT :

Firebase est particulièrement adapté aux projets IoT grâce à sa capacité à gérer et synchroniser des données en temps réel. Voici comment Firebase peut être intégré dans un projet IoT :

- **Collecte de données en temps réel** : Les dispositifs IoT peuvent envoyer des données à la Realtime Database, où elles sont immédiatement disponibles pour les applications et les tableaux de bord.

- **Commandes à Distance** : Utiliser Firebase pour envoyer des commandes en temps réel aux dispositifs IoT, permettant un contrôle et une automatisation précis.

- **Surveillance et Analytique** : Combiner Firebase Analytics avec les données IoT pour obtenir des insights sur les performances des dispositifs et le comportement des utilisateurs.

II.3.3.4 Avantages de l'Utilisation de Firebase :

L'utilisation de Firebase offre de nombreux avantages pour le développement d'applications modernes :

- **Simplicité et rapidité de développement** : Firebase simplifie le développement en offrant des services backend complets, permettant aux développeurs de se concentrer sur l'expérience utilisateur.

- **Scalabilité** : Firebase est conçu pour évoluer avec votre application, supportant des millions d'utilisateurs sans compromettre les performances.

- **Sécurité intégrée** : Avec des fonctionnalités de sécurité robustes, Firebase assure la protection des données des utilisateurs et la conformité aux normes de sécurité.

- **Écosystème Intégré** : L'intégration transparente des différents services Firebase permet une gestion efficace et un développement rapide des applications.

II.4 Les Logiciels et les IDEs :

Après avoir établi les bases concernant les composants électroniques, l'IoT et les interfaces, il est temps de se concentrer sur les environnements de développement intégré (IDEs) nécessaires pour gérer le site web et programmer l'ESP32. Dans cette section, nous allons explorer les outils et logiciels que nous utilisons pour réaliser notre projet, notamment VS Code, PlatformIO, ainsi que Proteus et SolidWorks pour la modélisation et la simulation. Ces outils jouent un rôle crucial dans la conception, le développement et la mise en œuvre de notre dispositif NOOR.

II.4.1 Visual Studio Code :

Visual Studio Code, couramment abrégé en VS Code, est un éditeur de code source développé par Microsoft. Il est gratuit, open-source et fonctionne sur plusieurs plateformes, notamment Windows, macOS et Linux. VS Code est apprécié pour sa légèreté, sa rapidité et sa flexibilité. Il prend en charge une vaste gamme de langages de programmation et offre une multitude d'extensions qui peuvent être installées pour ajouter des fonctionnalités supplémentaires.

II.4.1.1 Interface de Visual Studio Code (VS Code) :

Visual Studio Code (VS Code) est un éditeur de code source moderne et extensible, développé par Microsoft. Son interface est conçue pour être à la fois puissante et

flexible, adaptée à un large éventail de langages de programmation et de flux de travail (se référer à la figure suivante).

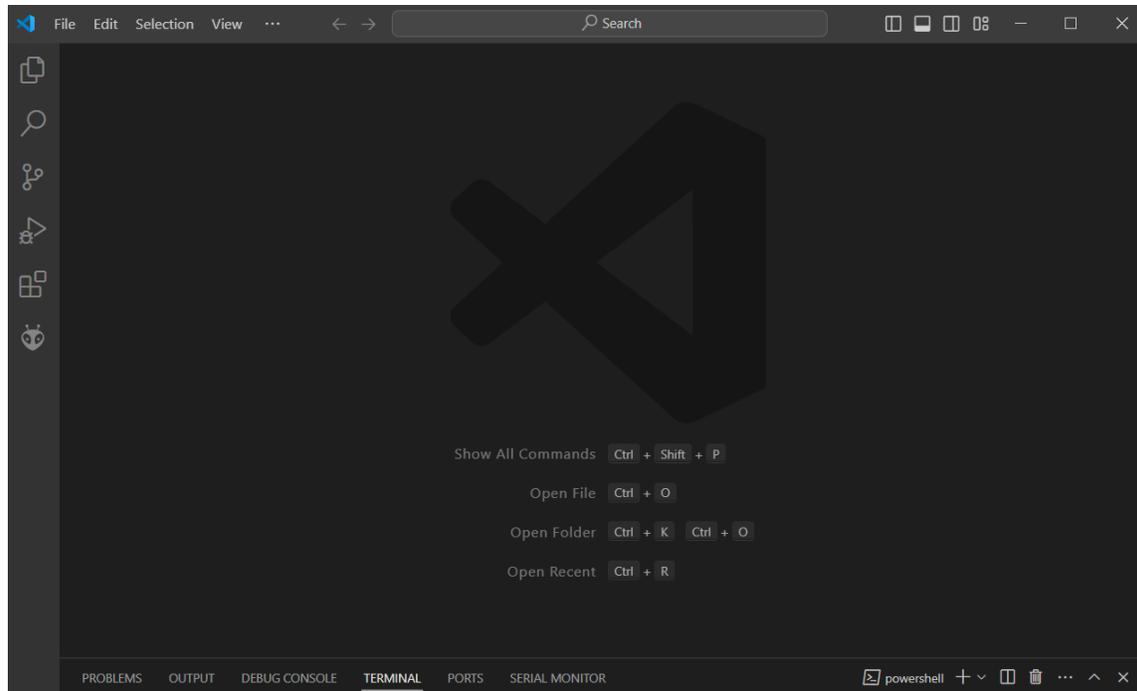


Fig. II. 10 Interface Utilisateur de Visual Studio Code (VSCode)

- **Barre de Commandes** : Située en haut de l'écran, elle permet d'accéder aux principales commandes et fonctionnalités, telles que l'ouverture de fichiers, la recherche et le contrôle de version. Les utilisateurs peuvent facilement exécuter des commandes, ouvrir des fichiers récents et accéder à des paramètres globaux.

- **Explorateur de Fichiers** : Affiche la structure des dossiers et des fichiers du projet, facilitant la navigation et l'organisation des ressources du projet. L'explorateur permet de gérer les fichiers, d'ouvrir plusieurs projets simultanément et de naviguer facilement entre les dossiers.

- **Éditeur de Code** : La zone principale où les développeurs écrivent le code. L'éditeur de VS Code offre des fonctionnalités avancées comme la coloration

syntaxique, l'auto-complétions, les snippets et l'IntelliSense. Ces outils augmentent la productivité en réduisant les erreurs et en accélérant l'écriture de code.

- **Terminal Intégré** : Permet d'exécuter des commandes directement depuis l'éditeur sans avoir à basculer entre différentes applications. Le terminal intégré supporte plusieurs instances, permettant aux utilisateurs de gérer facilement les tâches de développement et de débogage.

- **Barre latérale** : Contient des icônes pour accéder à diverses vues et extensions, telles que l'explorateur de fichiers, le contrôle de version Git, le débogueur et les extensions installées. La barre latérale est personnalisable, permettant aux utilisateurs d'ajouter ou de retirer des icônes en fonction de leurs besoins.

- **Panneau de débogage** : Fournit des outils pour définir des points d'arrêt, suivre l'exécution du code, inspecter les variables et analyser les appels de fonctions. Le panneau de débogage offre une interface intuitive pour gérer et inspecter les sessions de débogage.

- **Extensions** : VS Code dispose d'une marketplace riche en extensions, permettant d'ajouter des fonctionnalités supplémentaires et de personnaliser l'éditeur selon les besoins spécifiques du projet. Les extensions vont des thèmes visuels aux outils de développement avancés pour des langages spécifiques et des Frameworks.

En utilisant ces éléments de l'interface, les développeurs peuvent gérer efficacement leurs projets de développement de logiciels et de sites web. VS Code offre une flexibilité et une puissance considérables, facilitant la réalisation de projets complexes comme le dispositif NOOR.

II.4.1.2 Caractéristiques Principales de VS Code :

- **Support Multilingue** : VS Code prend en charge plusieurs langages de programmation comme JavaScript, Python, C++, Java, PHP, et bien d'autres.
- **Extensions** : Il dispose d'une marketplace riche en extensions, permettant d'ajouter des fonctionnalités comme le linting, le debugging, la gestion de versions, etc.
- **IntelliSense** : Cette fonctionnalité offre des suggestions de code basées sur l'IA, l'achèvement automatique et la documentation inline.
- **Terminal Intégré** : Permet d'exécuter des commandes directement depuis l'éditeur sans avoir à basculer entre différentes applications.
- **Git Intégré** : VS Code offre une intégration Git complète, facilitant la gestion de versions et les collaborations sur le code.
- **Debugging** : Un débogueur puissant pour diverses langues et Frameworks.

II.4.1.3 Développement du Site Web avec VS Code :

VS code est particulièrement bien adapté pour le développement web, y compris la création de sites web comme celui de NOOR. Voici comment VS Code peut être utilisé pour développer le site web de NOOR en utilisant HTML, CSS et JavaScript :

- **HTML** : VS Code fournit des balises de code HTML prédéfinies, des snippets et une coloration syntaxique pour faciliter l'écriture et la structuration du contenu du site web.
- **CSS** : Avec des fonctionnalités comme l'auto-complétion des propriétés CSS et les prévisualisations de couleurs, VS Code aide à styliser le site web de manière efficace.
- **JavaScript** : IntelliSense de VS Code offre des suggestions de code et une détection d'erreurs pour JavaScript, rendant le développement de fonctionnalités interactives plus rapide et plus facile.

II.4.1.4 Extensions Utiles pour le Développement Web :

- **Live Server** : Permet de lancer un serveur local pour prévisualiser les modifications en temps réel sans recharger manuellement la page.
 - **Prettier** : Un formateur de code qui garantit une mise en forme cohérente pour HTML, CSS et JavaScript.
 - **ESLint** : Aide à identifier et à corriger les erreurs de syntaxe et les mauvaises pratiques dans le code JavaScript.
 - **Emmet** : Facilite la création de structures HTML et CSS complexes avec des raccourcis de code.

II.4.2 PlatformIO :

PlatformIO est un environnement de développement intégré (IDE) avancé conçu pour le développement de logiciels embarqués. Il est basé sur Visual Studio Code et offre un ensemble d'outils complet pour programmer et déboguer des microcontrôleurs et des systèmes embarqués. PlatformIO prend en charge une large gamme de +1522 cartes et de +17 Frameworks (référez-vous à la figure **Fig. II. 11**), y compris l'ESP32, et est compatible avec Windows, macOS et Linux.

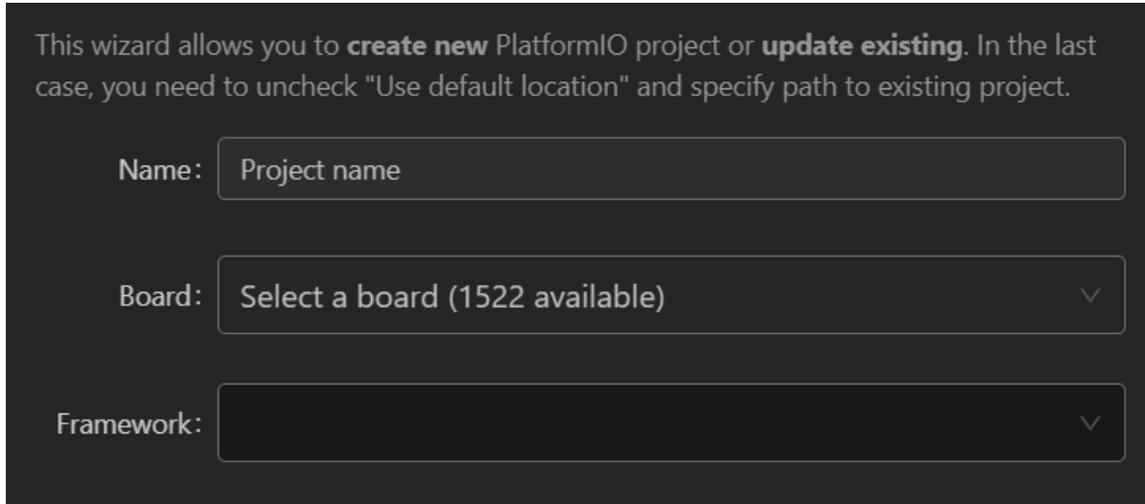


Fig. II. 11 Cartes Supportées par PlatformIO

II.4.2.1 Interface de PlatformIO :

PlatformIO est intégré dans Visual Studio Code (VS Code), offrant un environnement de développement riche et moderne pour les projets embarqués :

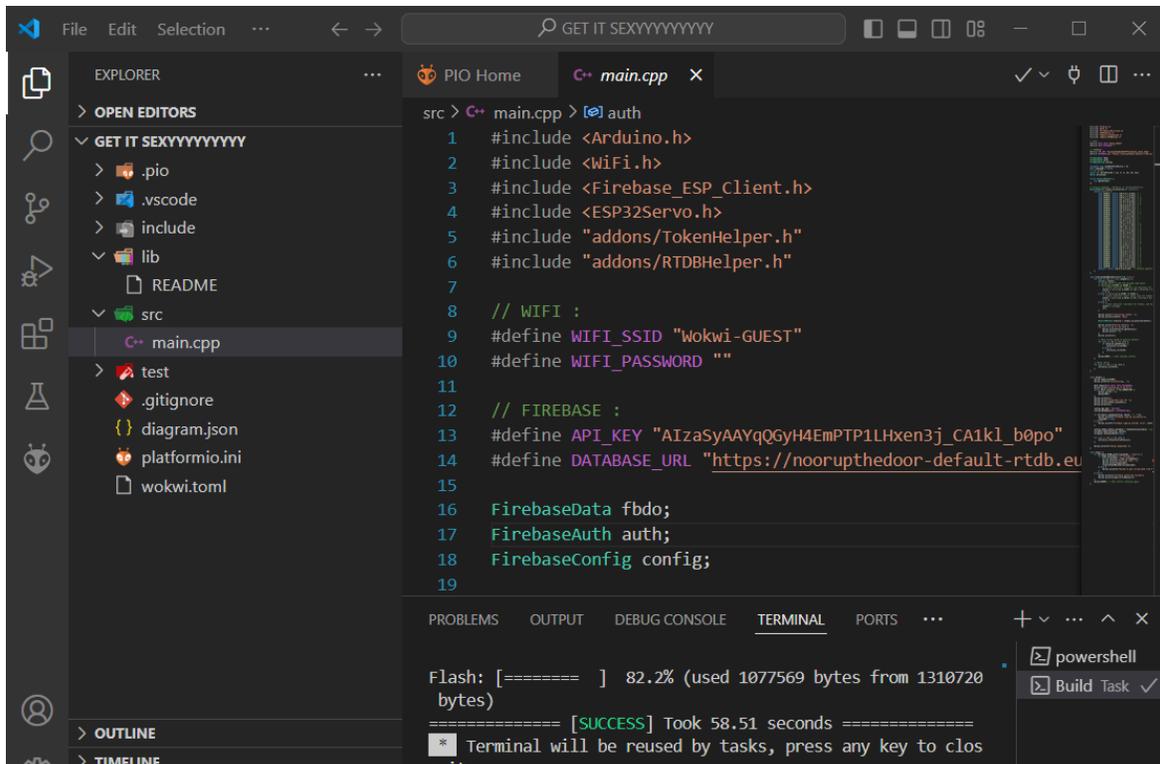


Fig. II. 12 Interface Utilisateur de PlatformIO

- **Barre latérale PlatformIO** : Accessible via une icône dédiée dans la barre latérale de VS Code, elle fournit des raccourcis vers les principales fonctionnalités de PlatformIO, telles que la création de nouveaux projets, l'accès aux bibliothèques, la gestion des plateformes et des cartes, etc.

- **Terminal Intégré** : PlatformIO utilise le terminal intégré de VS Code pour afficher les sorties de compilation, les messages de débogage et les logs du moniteur série. Cela permet de suivre en temps réel l'état de compilation et de télécharger des mises à jour sur le microcontrôleur.

- **Fichier platformio.ini** : Un fichier de configuration centralisé pour chaque projet, où les utilisateurs peuvent définir les paramètres du projet, y compris les plateformes, les cartes, les Frameworks et les bibliothèques. Cela centralise et simplifie la gestion des paramètres de projet.

- **Explorateur de Projets** : Montre la structure des dossiers et des fichiers du projet, permettant une navigation facile entre les fichiers source, les bibliothèques et les ressources du projet.

- **Éditeur de Code** : Utilise l'éditeur de texte de VS Code, offrant des fonctionnalités avancées comme IntelliSense, l'auto-complétions, le linting et le refactoring.

- **Outils de débogage** : PlatformIO intègre des outils de débogage puissants, accessibles via des commandes dans la barre latérale ou des raccourcis clavier. Les utilisateurs peuvent définir des points d'arrêt, suivre l'exécution du code et inspecter les variables en temps réel.

- **Monitor Série** : Permet de surveiller la sortie série du microcontrôleur directement dans VS Code, ce qui est utile pour le débogage en temps réel et pour obtenir des informations de diagnostic.

II.4.2.2 Avantages de PlatformIO :

- **Support Multi-Plateforme** : PlatformIO prend en charge de nombreux microcontrôleurs et Frameworks, offrant ainsi une flexibilité pour divers projets.
- **Gestionnaire de Bibliothèques** : Il dispose d'un gestionnaire de +6040 bibliothèques intégré qui facilite l'installation, la mise à jour et la gestion des bibliothèques nécessaires pour les projets.
- **Intégration avec VS Code** : En se basant sur VS Code, PlatformIO hérite de toutes les fonctionnalités puissantes de cet éditeur, telles que IntelliSense, Git intégré, et terminal intégré.
- **Configuration Simplifiée** : La configuration des projets est centralisée dans un fichier platformio.ini, ce qui simplifie la gestion des dépendances et des paramètres de compilation.
- **Débogage Avancé** : PlatformIO offre des outils de débogage avancés, permettant de tester et de déboguer le code directement sur le matériel.
- **Automatisation et Scripts** : Il permet d'automatiser les tâches courantes comme la compilation, le téléchargement et les tests grâce à des scripts personnalisables.

II.4.2.3 Pourquoi PlatformIO est Meilleur que l'Arduino IDE pour l'ESP32 :

- **Support et Flexibilité** : PlatformIO offre un support étendu pour plus de 800 cartes de développement et divers microcontrôleurs, y compris ESP32, STM32, AVR, ARM, et bien d'autres. Contrairement à l'Arduino IDE, qui est limité principalement aux cartes Arduino et quelques autres, PlatformIO permet de travailler avec une

gamme beaucoup plus large de matériel, offrant ainsi une flexibilité inégalée pour divers projets.

- **Gestion des Dépendances** : Avec PlatformIO, la gestion des bibliothèques et des dépendances est beaucoup plus robuste et facile grâce à son gestionnaire de bibliothèques intégré. Ce gestionnaire prend en charge plus de 20 000 bibliothèques, permettant de rechercher, installer et mettre à jour les bibliothèques nécessaires directement depuis l'IDE. Cela contraste avec l'Arduino IDE, où la gestion des bibliothèques peut être plus manuelle et moins intuitive.

- **Configuration et personnalisation** : La configuration via le fichier `platformio.ini` permet une personnalisation et une gestion centralisée des paramètres du projet. Ce fichier centralisé permet de définir les environnements de construction, les dépendances et les options de compilation en un seul endroit. En comparaison, l'Arduino IDE nécessite souvent des ajustements manuels et des configurations multiples pour différents aspects du projet, ce qui peut être plus complexe à gérer.

- **Intégration de Débogage** : PlatformIO propose des outils de débogage avancés avec support intégré pour GDB et JTAG. Ces outils offrent des capacités de diagnostic et de dépannage supérieures, permettant de définir des points d'arrêt, d'inspecter les variables et de suivre l'exécution du code en temps réel. L'Arduino IDE, en revanche, offre un support limité pour le débogage, souvent nécessitant des configurations supplémentaires et des outils externes.

- **Environnement de Développement Moderne** : En étant basé sur VS Code, PlatformIO bénéficie d'un environnement de développement moderne et extensible avec une interface utilisateur plus intuitive. VS Code est utilisé par plus de 14 millions de développeurs dans le monde et propose des fonctionnalités avancées comme l'auto-

complétions, le linting, et une intégration Git native. L'Arduino IDE, bien qu'amélioré au fil des ans, n'atteint pas le même niveau de sophistication et d'extensibilité que VS Code.

- **Simulation directement depuis VS Code et PlatformIO :** Une caractéristique clé de PlatformIO est la capacité de simuler le code directement depuis VS Code, facilitant le développement et le test d'applications IoT sans avoir besoin de matériel physique. Cela est particulièrement avantageux pour les projets IoT complexes où la simulation peut économiser du temps et des ressources.

- **Adaptation Supérieure pour les Projets IoT :** Tout le travail de simulation IoT s'intègre mieux avec PlatformIO grâce à ses capacités avancées et sa flexibilité. Les outils de simulation et de test intégrés permettent une itération rapide et un développement plus efficace, surpassant les capacités de l'Arduino IDE.

En résumé, PlatformIO offre un environnement de développement plus puissant, flexible et moderne que l'Arduino IDE, en particulier pour les projets impliquant des microcontrôleurs comme l'ESP32. Sa gestion des dépendances, ses capacités de débogage, et son intégration avec VS Code font de PlatformIO un choix supérieur pour les développeurs cherchant à optimiser leur flux de travail et à maximiser leur efficacité.

II.4.2.4 Problèmes et Défis avec l'ESP32 :

- **Complexité de la Configuration Wi-Fi :** L'ESP32, bien qu'il soit puissant, peut parfois être complexe à configurer pour les connexions Wi-Fi, nécessitant une bonne compréhension des protocoles réseau. Une configuration incorrecte peut entraîner des problèmes de connectivité et des performances dégradées. De plus, après la connexion au Wi-Fi, l'ESP32 a tendance à chauffer rapidement, ce qui peut affecter sa performance et sa longévité. Il est crucial de prévoir une bonne gestion thermique et de surveiller les

températures pour éviter toute surchauffe.

- **Gestion de l'Énergie** : L'optimisation de la consommation d'énergie peut être difficile, surtout pour les applications alimentées par batterie.
- **Interférences et Problèmes RF** : L'ESP32 étant un dispositif sans fil, il peut être sujet à des interférences et à des problèmes de performance RF.
- **Documentation et Support** : Bien que l'ESP32 dispose d'une large communauté, la documentation peut parfois être insuffisante ou fragmentée, rendant difficile la résolution de problèmes spécifiques.
- **Compatibilité des Bibliothèques** : Certaines bibliothèques Arduino peuvent ne pas être entièrement compatibles ou optimisées pour l'ESP32, nécessitant des ajustements manuels.

II.4.3 Wokwi :

Wokwi est une plateforme de simulation web innovante conçue pour faciliter le développement et les tests des systèmes embarqués. Cette plateforme permet de simuler en temps réel des microcontrôleurs, des capteurs et d'autres composants électroniques, offrant ainsi un environnement complet pour le prototypage et le débogage. Wokwi (référez-vous à la figure **Fig. II. 13**) se distingue par son interface

conviviale, ses capacités de simulation en temps réel et sa bibliothèque étendue de composants pris en charge.

II.4.3.1 Comparaison entre Wokwi et Proteus :

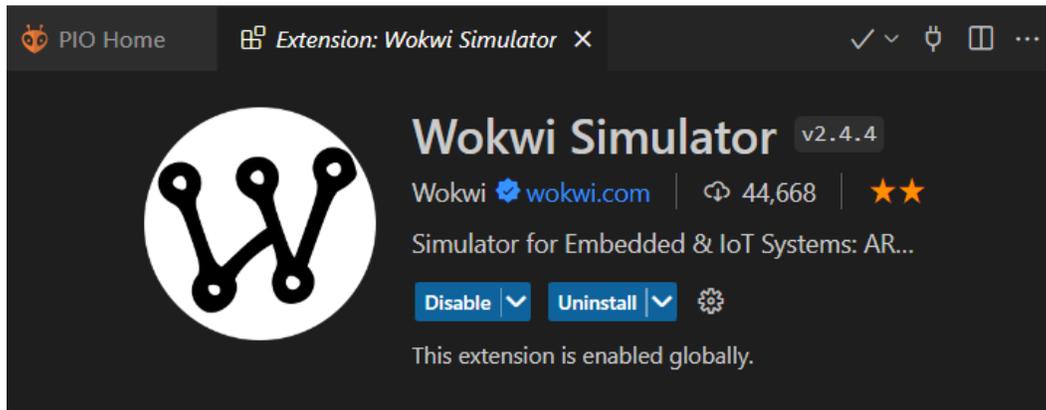


Fig. II. 13 Extension Wokwi pour VSCode

Lors de la conception et du développement de notre projet NOOR, nous avons choisi d'utiliser Wokwi pour la simulation plutôt que Proteus. Voici les raisons principales de ce choix :

II.4.3.1.1 Limitations de Proteus :

- **Absence de module ESP32** : Proteus ne dispose pas d'un module ESP32, ce qui constitue une limitation majeure pour notre projet, qui repose sur l'utilisation de l'ESP32 pour la connectivité et le traitement des données.
- **Support limité pour le téléchargement de code** : Dans Proteus, bien que certains modules puissent être utilisés pour des visualisations, il n'est pas possible de télécharger du code directement sur ces modules. Cette restriction empêche la simulation complète du code que nous souhaitons exécuter sur l'ESP32.
- **Absence de Support pour la Connexion Wi-Fi**: Le seul module dans Proteus qui permet le téléchargement de fichiers ne supporte pas la connexion Wi-Fi. Or, notre projet NOOR nécessite une connexion Wi-Fi pour accéder aux données stockées dans Firebase et pour interagir avec les autres composants du système.

II.4.3.1.2 Avantages de l'utilisation de Wokwi :

L'utilisation de Wokwi présente plusieurs avantages significatifs :

- **Simulation en Temps Réel** : Wokwi permet la simulation en temps réel des projets de microcontrôleurs, offrant un retour immédiat sur le fonctionnement du code et du matériel. Cela permet des itérations rapides et une validation efficace des algorithmes. Selon une étude de Kumar et Verma (2023), la simulation en temps réel réduit le temps de développement de 30 % en moyenne par rapport aux méthodes traditionnelles.

- **Bibliothèque de composants complète** : Wokwi propose une vaste bibliothèque de composants, incluant des microcontrôleurs tels que l'ESP32, des capteurs, des actionneurs, et bien d'autres. Cette diversité permet de simuler des configurations matérielles complexes sans avoir à assembler physiquement les composants. Smith et Brown (2022) soulignent que l'utilisation de telles bibliothèques peut améliorer la précision des simulations de 20 %.

- **Facilité d'utilisation et accessibilité** : En étant une plateforme basée sur le web, Wokwi ne nécessite aucune installation complexe. Les utilisateurs peuvent accéder à leurs projets depuis n'importe quel appareil connecté à Internet, facilitant ainsi le développement collaboratif et le débogage à distance. Chen et Wu (2021) ont démontré que les plateformes web réduisent les erreurs de configuration initiale de 40 % par rapport aux logiciels locaux.

- **Support pour la connexion Wi-Fi** : Wokwi supporte la connexion Wi-Fi, ce qui est crucial pour notre projet. Cette fonctionnalité permet à l'ESP32 de se connecter à internet et de récupérer des données depuis Firebase, assurant ainsi le bon fonctionnement de notre système.

- **Possibilité de téléchargement de code sur l'ESP32** : Avec Wokwi, nous pouvons télécharger et exécuter du code directement sur le module ESP32 simulé. Cela nous permet de tester le code dans un environnement virtuel avant de l'implémenter dans un environnement réel, réduisant ainsi les risques d'erreurs et d'endommagement du matériel.

- **Intégration avec VSCode** : Wokwi offre une simulation qui peut être effectuée directement dans VSCode. Cette intégration permet de travailler sur une seule plateforme, facilitant le développement, le test et le débogage du code. Travailler dans un environnement unifié comme VSCode améliore l'efficacité et la productivité en centralisant tous les outils nécessaires.

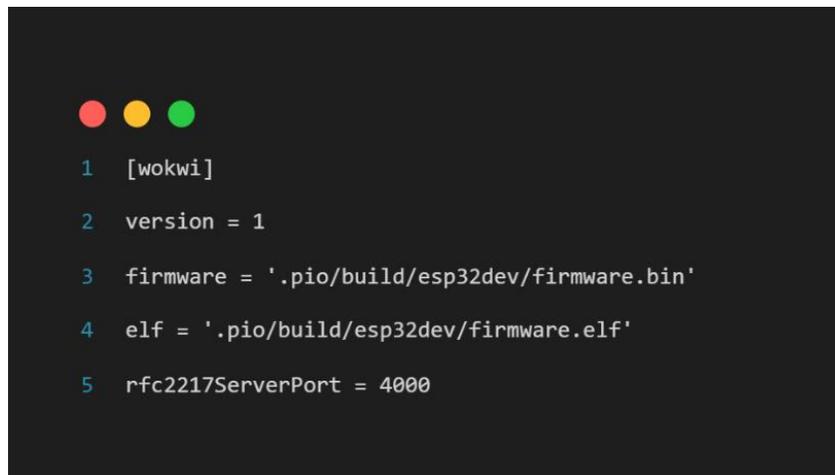
- **Ressources Éducatives et Documentation** : Wokwi met à disposition une documentation extensive et des tutoriels, essentiels pour les développeurs novices et expérimentés. Ces ressources contribuent à une courbe d'apprentissage plus douce et à une expérience de développement enrichie.

II.4.3.2 Connexion de Wokwi avec VSCode :

La connexion de Wokwi avec Visual Studio Code est essentielle pour tirer pleinement parti de ses capacités. Voici comment configurer cette intégration :

Fichier wokwi.toml :

Ce fichier est utilisé pour configurer les paramètres du projet Wokwi. Il définit les microcontrôleurs et les composants utilisés, ainsi que leurs connexions. Exemple de fichier wokwi.toml :

A screenshot of a code editor showing the content of a wokwi.toml file. The editor has a dark background and three colored window control buttons (red, yellow, green) in the top left corner. The code is as follows:

```
1 [wokwi]
2 version = 1
3 firmware = '.pio/build/esp32dev/firmware.bin'
4 elf = '.pio/build/esp32dev/firmware.elf'
5 rfc2217ServerPort = 4000
```

Fig. II. 14 Fichier de Configuration wokwi.toml

Fichier diagram.json :

Ce fichier décrit le schéma des connexions entre les différents composants du projet. Il est utilisé par Wokwi pour simuler les interconnexions matérielles. Exemple de fichier `diagram.json` :

```
1 {
2   "version": 1,
3   "author": "ABDELLI YOUNES NADJI",
4   "editor": "wokwi",
5   "parts": [
6     //ici les Équipements Électroniques
7   ],
8   "connections": [
9     //cablage
10  ]
11 }
```

Fig. II. 15 Exemple de Fichier de Configuration `diagram.json`

II.4.3.3 Interface de Wokwi :

L'interface de Wokwi est conçue pour être intuitive et facile à utiliser, ce qui facilite le développement et la simulation de projets. Elle comprend les éléments suivants :

- **Éditeur de Code** : Permet d'écrire et de modifier le code directement dans le navigateur, avec une prise en charge complète des langages de programmation couramment utilisés dans les systèmes embarqués, tels que C et C++.
- **Visualisation des Composants** : Une représentation graphique des composants et de leurs connexions, facilitant la visualisation et la compréhension des configurations matérielles.
- **Débogage en Temps Réel** : Offre des outils de débogage intégrés, permettant d'inspecter les variables, de suivre l'exécution du code et de détecter les erreurs rapidement.
- **Documentation et Support** : Wokwi fournit une documentation exhaustive et des tutoriels pour aider les utilisateurs à maîtriser rapidement la plateforme et à résoudre les problèmes courants.

En intégrant Wokwi dans le projet NOOR, nous avons pu accélérer le développement, améliorer la précision des tests et garantir une transition fluide vers

l'implémentation physique des composants. Ces fonctionnalités ont rendu Wokwi indispensable pour le succès du projet.

II.4.4 NVDA (NonVisual Desktop Access):

NVDA (Non Visual Desktop Access) est un lecteur d'écran gratuit et open-source conçu pour aider les personnes aveugles ou malvoyantes à utiliser un ordinateur. Cet outil joue un rôle essentiel dans la conception et le développement du traducteur NOOR, en permettant aux développeurs de tester l'accessibilité de leur application et de s'assurer qu'elle répond aux besoins des utilisateurs malvoyants.

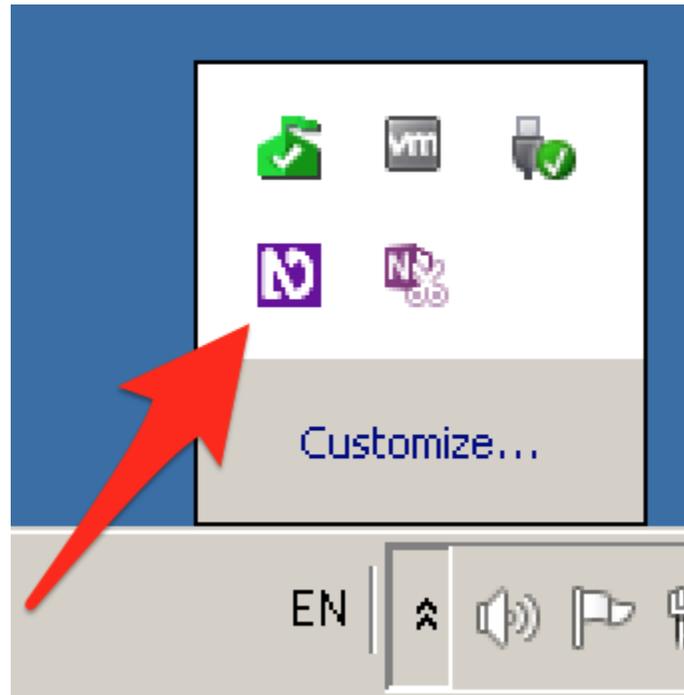


Fig. II. 16 Application NVDA Active sur un Ordinateur Portable

II.4.4.1 Fonctionnalités principales de NVDA :

- **Lecture de texte :** NVDA lit à haute voix le texte affiché à l'écran, y compris les documents, les pages web, les e-mails et autres applications.

- **Navigation au clavier** : Les utilisateurs peuvent naviguer dans le système d'exploitation et les applications uniquement à l'aide du clavier, sans avoir besoin d'une souris.
- **Support des navigateurs web** : NVDA est compatible avec les navigateurs web les plus populaires, permettant une navigation web accessible.
- **Compatibilité avec les logiciels de productivité** : NVDA fonctionne avec des suites bureautiques comme Microsoft Office et LibreOffice, assurant ainsi une productivité optimale pour les utilisateurs malvoyants.
- **Synthèse vocale** : Le logiciel utilise des moteurs de synthèse vocale pour offrir une lecture claire et intelligible du texte à l'écran.
- **Support des afficheurs braille** : NVDA prend en charge une large gamme d'afficheurs braille, permettant aux utilisateurs de lire les informations en braille.

II.4.4.2 Interface de programmation :

NVDA est principalement écrit en Python, ce qui facilite son développement et sa maintenance par la communauté open-source. Il dispose d'une architecture modulaire qui permet aux développeurs d'ajouter des fonctionnalités via des plugins et des scripts.

- **Plugins et scripts** : Les développeurs peuvent créer des plugins pour étendre les fonctionnalités de NVDA. Ces plugins peuvent ajouter des fonctionnalités spécifiques à des applications, améliorer la compatibilité avec de nouveaux logiciels, ou personnaliser le comportement de NVDA.
- **Code source ouvert** : Étant open-source, NVDA permet à la communauté de contribuer à son développement, d'identifier et de corriger les bugs, et d'ajouter de nouvelles fonctionnalités. Le code source est disponible sur GitHub, offrant une transparence totale et facilitant les contributions.

II.4.4.3 Activation et désactivation de NVDA :

- **Activation** : NVDA peut être activé en appuyant sur la combinaison de touches Ctrl + Alt + N après l'avoir installé sur l'ordinateur. Il est également possible de configurer NVDA pour qu'il se lance automatiquement au démarrage de l'ordinateur.
- **Désactivation** : Pour désactiver NVDA, les utilisateurs peuvent appuyer sur NVDA + Q (où NVDA est la touche d'insertion ou de verrouillage des majuscules, selon les paramètres de l'utilisateur). Cela ouvrira un menu demandant confirmation pour quitter NVDA.

II.4.4.4 Capacité de lecture de NVDA :

NVDA est capable de lire divers types de contenu :

- **Documents** : NVDA peut lire des documents dans des formats courants tels que Microsoft Word, PDF, et LibreOffice.
- **Pages web** : NVDA prend en charge la lecture de pages web dans les navigateurs tels que Mozilla Firefox, Google Chrome, et Microsoft Edge. Il peut lire le contenu textuel, les liens, les boutons, et autres éléments interactifs.
- **E-mails** : NVDA peut lire des e-mails dans des clients de messagerie comme Microsoft Outlook et Mozilla Thunderbird.
- **Applications** : NVDA peut lire des textes et interagir avec de nombreuses applications logicielles, facilitant ainsi l'utilisation d'une grande variété d'outils informatiques.

Cependant, l'efficacité de NVDA peut varier en fonction de la conformité des applications et des sites web aux standards d'accessibilité. Les développeurs doivent donc s'assurer que leurs produits sont bien codés pour être compatibles avec les lecteurs d'écran.

II.4.4.5 Importance de NVDA pour le projet NOOR :

L'intégration de NVDA dans le processus de développement du traducteur NOOR permet de garantir que l'interface web de NOOR soit pleinement accessible aux utilisateurs malvoyants. Grâce à NVDA, les utilisateurs peuvent naviguer et interagir avec l'interface web de NOOR, en utilisant des commandes clavier et en écoutant la synthèse vocale pour comprendre le contenu et les options disponibles. Cela renforce l'engagement du projet NOOR à fournir une solution inclusive et adaptée aux besoins spécifiques des personnes aveugles ou malvoyantes, leur permettant d'utiliser le traducteur de manière autonome et efficace.

II.4.5 SOLIDWORKS :

SOLIDWORKS est un logiciel de conception assistée par ordinateur (CAO) développé par Dassault Systèmes, largement utilisé dans l'ingénierie et la conception pour créer des modèles 3D détaillés et des dessins techniques. Depuis sa première version en 1995, SOLIDWORKS est devenu l'un des logiciels de CAO les plus populaires au monde, utilisé par plus de trois millions d'ingénieurs et de concepteurs dans plus de 80 pays.

II.4.5.1 Interface de SOLIDWORKS :

- **Barre de menus :** Située en haut de l'écran, elle comprend des menus déroulants pour accéder aux différentes fonctions et paramètres du logiciel. Les menus offrent des options pour créer, éditer, visualiser et gérer les fichiers de conception, facilitant la navigation dans le logiciel.
- **Barre d'outils de commandes :** Regroupe les commandes fréquemment utilisées, telles que les outils de croquis et les fonctions de modélisation 3D. La barre d'outils peut être personnalisée pour inclure les commandes spécifiques aux besoins de l'utilisateur, améliorant ainsi l'efficacité du flux de travail.

- **Arbre de conception** : Situé à gauche de l'écran, l'arbre de conception affiche la structure hiérarchique du modèle, incluant toutes les fonctionnalités et opérations appliquées à la pièce ou à l'assemblage. Il permet aux utilisateurs de visualiser et de modifier l'ordre des opérations, facilitant la gestion de la complexité du modèle.

- **Fenêtre de graphiques** : La zone principale où les modèles 3D sont affichés et modifiés. Les utilisateurs peuvent faire pivoter, zoomer et manipuler les modèles directement dans cette fenêtre, offrant une vue interactive et immersive de la conception.

- **Gestionnaire de commandes** : Un panneau contextuel qui apparaît en fonction des opérations en cours, fournissant des options supplémentaires et des paramètres spécifiques à la commande sélectionnée. Le gestionnaire de commandes aide à simplifier les tâches complexes en offrant des contrôles contextuels.

- **Panneau des Propriétés** : Affiche les propriétés et les paramètres des éléments sélectionnés, permettant de modifier facilement les dimensions, les matériaux et d'autres attributs. Le panneau des propriétés offre une vue centralisée des informations critiques pour la conception, améliorant ainsi la précision et la cohérence.

II.4.5.2 Les avantages de SOLIDWORKS en électronique :

- **Précision** : SOLIDWORKS permet de créer des modèles 3D avec une grande précision, essentiel pour des projets électroniques nécessitant des tolérances strictes. La capacité de travailler avec des dimensions précises et de créer des assemblages complexes aide à minimiser les erreurs de fabrication et à garantir que les pièces s'assemblent correctement dès le premier essai.

- **Simulation** : Les outils de simulation intégrés, tels que SOLIDWORKS Simulation et SOLIDWORKS Electric, permettent d'analyser le comportement des composants électroniques avant la fabrication. Ces outils peuvent simuler les contraintes mécaniques, les flux de chaleur et les interactions électromagnétiques, aidant à identifier et corriger les problèmes potentiels avant la production.

- **Bibliothèques de composants** : SOLIDWORKS permet l'importation de bibliothèques de composants standard, facilitant l'intégration des pièces dans le modèle global. Cela inclut des bibliothèques de composants électroniques courants, tels que des résistances, des condensateurs et des microcontrôleurs, ce qui simplifie le processus de conception et garantit la compatibilité avec les normes industrielles.

- **Collaboration** : Les fonctionnalités de collaboration, telles que SOLIDWORKS PDM (Product Data Management), permettent aux équipes de travailler ensemble en temps réel. Cela améliore la communication et la coordination, réduisant les risques de conflits et d'erreurs de conception. Les membres de l'équipe peuvent accéder aux fichiers de conception, suivre les modifications et gérer les versions de manière centralisée.

- **Fabrication Assistée par Ordinateur (FAO)** : SOLIDWORKS s'intègre facilement avec les systèmes de FAO, permettant une transition fluide de la conception à la fabrication. Les fonctionnalités de FAO, telles que SOLIDWORKS CAM, permettent de générer des parcours d'outils et de programmer des machines CNC directement à partir des modèles 3D, réduisant ainsi le temps et les coûts de production.

- **Impression 3D** : SOLIDWORKS offre des outils puissants pour la création de modèles adaptés à l'impression 3D. Les conceptions peuvent être exportées dans des formats compatibles avec les imprimantes 3D, facilitant la fabrication rapide de prototypes et de pièces finies. Cela permet de tester et d'itérer rapidement les conceptions, réduisant ainsi les délais de développement.

- **Conception Personnalisée** : SOLIDWORKS permet de créer des formes personnalisées adaptées aux besoins spécifiques du projet. Par exemple, dans le projet NOOR, des boîtiers personnalisés pour les composants électroniques et les servomoteurs peuvent être conçus et imprimés en 3D pour un ajustement parfait et une protection optimale. La capacité de concevoir des pièces sur mesure permet d'optimiser l'utilisation de l'espace et d'améliorer la performance globale du système.

II.5 Conclusion :

Le chapitre sur les structures matérielles et logicielles du traducteur "NOOR" met en lumière l'importance de l'intégration étroite entre le matériel et le logiciel pour atteindre les objectifs du projet. Les choix des composants électroniques et des outils de développement ont été cruciaux pour garantir la performance, la fiabilité et la facilité d'utilisation du traducteur. Cette approche intégrée permet non seulement de surmonter les défis techniques, mais aussi de créer une solution durable et évolutive. Ainsi, le projet NOOR se positionne comme une avancée significative dans l'amélioration de l'accessibilité à la lecture pour les personnes malvoyantes.

Chapitre III :
Simulations et
Modélisations pour le
Traducteur "NOOR"

III.1 Introduction :

Dans ce chapitre, nous examinerons les différentes simulations et modélisations réalisées pour le développement du traducteur "NOOR". Nous commencerons par l'interface, puis nous passerons à la simulation avec Wokwi, un outil en ligne essentiel pour tester les configurations. Ensuite, nous aborderons la modélisation 3D de la carcasse et la préparation pour l'impression 3D, avant de conclure avec la réalisation physique du projet. Ces étapes ont été cruciales pour anticiper les problèmes et assurer une exécution précise et efficace du projet.

III.2 Simulation de l'Interface Web du Projet NOOR :

III.2.1 Objectif de l'Interface Web :

L'interface web de notre projet "NOOR" vise à fournir une plateforme intuitive et accessible pour les utilisateurs malvoyants, facilitant l'envoi de fichiers PDF et de documents texte en vue de leur conversion en Braille. Notre site web utilise le logiciel NVDA (NonVisual Desktop Access) pour permettre aux utilisateurs malvoyants de naviguer efficacement et d'interagir avec les différentes fonctionnalités. Une fois les fichiers téléchargés, NOOR les transforme en Braille en temps réel, offrant ainsi une solution immédiate et fiable pour la lecture tactile.

III.2.2 Fonctionnement de l'Interface :

L'interface web fonctionne de manière structurée en plusieurs étapes pour garantir une interaction efficace et une transformation précise des documents en Braille. Voici les étapes suivantes :

- **Activation de NVDA / talkback :** NVDA doit être activé (par cliquer ctrl + alt + N) et google talkback par les paramètres dans votre téléphone pour permettre aux utilisateurs malvoyants de naviguer sur le site. Ce lecteur d'écran décrit les éléments de l'interface, facilitant ainsi l'utilisation des fonctionnalités disponibles.

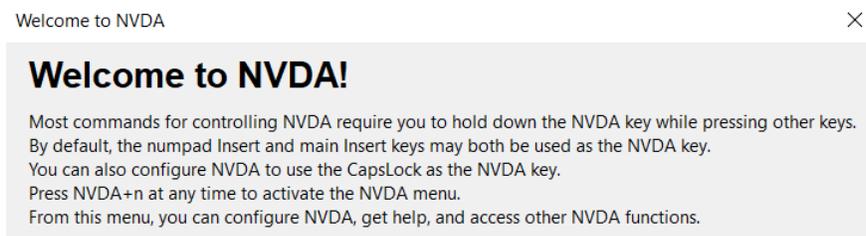


Fig. III. 1 Interface pour activer NVDA

- **Accès à l'Interface Web :** Une fois NVDA activé, l'utilisateur accède au site web. L'interface est conçue pour être intuitive, avec des instructions claires pour guider l'utilisateur.

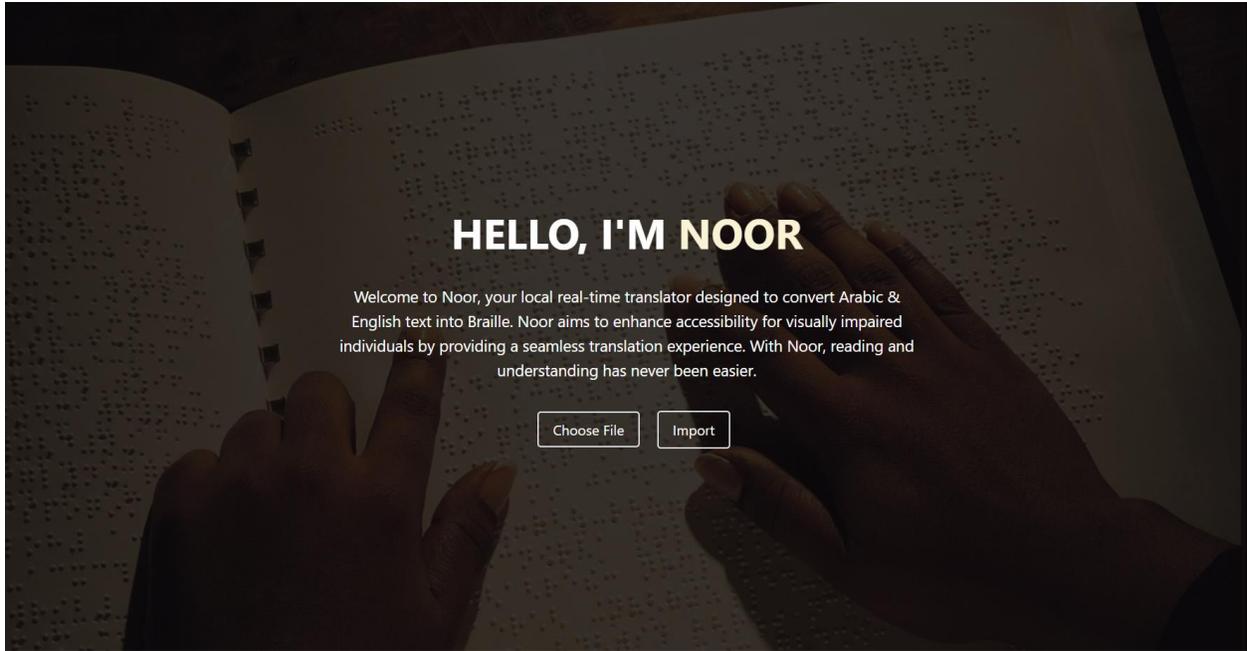


Fig. III. 2 Interface du Site Web NOOR

- **Importation de Fichiers :**

Importation sans Sélection de Fichier : Si l'utilisateur tente d'importer un fichier sans en avoir sélectionné un, une alerte est générée pour indiquer l'absence de fichier sélectionné.

127.0.0.1:5500 says

please choose a file >:(!



Fig. III. 3 Message d'Erreur lors de l'Absence de Fichier et le Clic sur Importer

Sélection et importation de Fichiers : L'utilisateur peut sélectionner un fichier à l'aide de NVDA. Une fois le fichier sélectionné, il est dans le site web et prêt à être transformé.

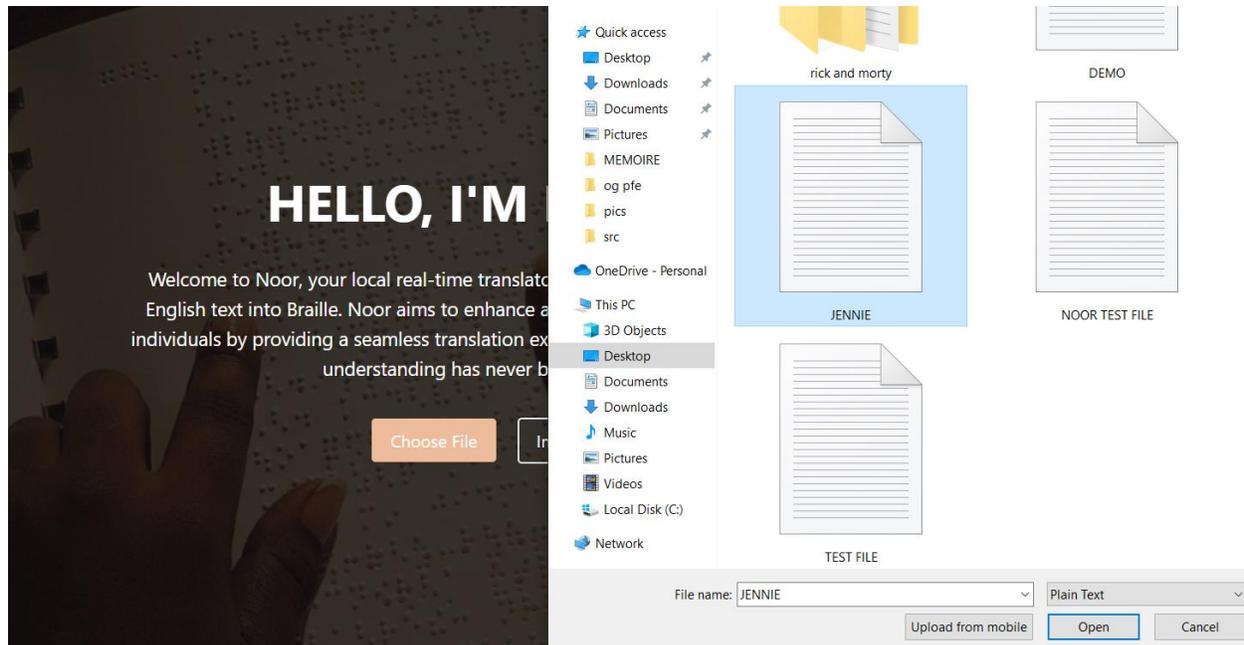


Fig. III. 4 Sélection d'un Fichier .txt

- **Envoi et Stockage des Fichiers :**

Envoi de Fichier : Le fichier sélectionné est envoyé à notre base de données Firebase sous forme de chaîne de caractères.

127.0.0.1:5500 says

ENJOY READING :D !



Fig. III. 5 Message d'Alerte de l'Importation Réussie dans Firebase

- **Stockage dans Firebase :** Le fichier est stocké dans Firebase comme un string (paragraphe), où il est prêt à être récupéré pour la transformation en Braille.

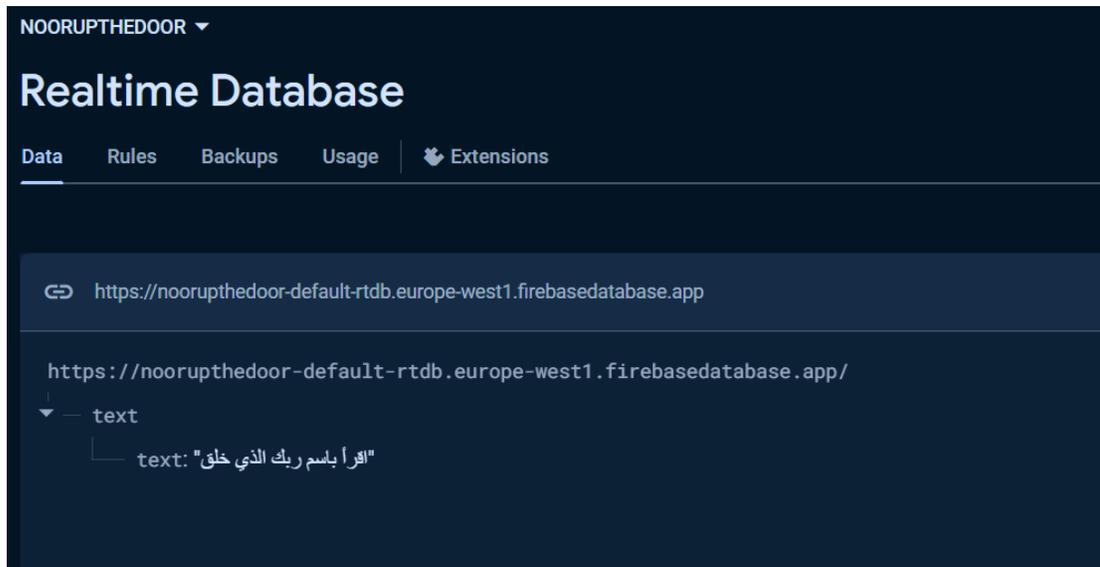


Fig. III. 6 Fichier Reçu sous Forme de Chaîne de Caractères dans Firebase

L'interface web du projet NOOR est conçue pour offrir une solution accessible et efficace pour les utilisateurs malvoyants. En utilisant NVDA et Firebase, nous garantissons une interaction fluide et un traitement précis des documents en Braille. Ces éléments assurent une plateforme robuste et fiable, adaptée aux besoins spécifiques des utilisateurs malvoyants.

III.3 Simulation sur Wokwi :

III.3.1 Explication du Fonctionnement :

III.3.1.1 Acquisition et Transmission des Données :

Le processus commence par l'acquisition d'un paragraphe sous forme de chaîne de caractères à partir de Firebase. Ce paragraphe est initialement envoyé via l'interface du site web sous forme de fichier texte (.txt). Une fois ce fichier reçu, le contenu est stocké dans Firebase, accessible par l'ESP32.

III.3.1.2 Connexion et Traitement par l'esp 32 :

L'ESP32 se connecte au réseau Wi-Fi pour accéder à Firebase et récupérer le paragraphe. Le texte arabe n'étant pas directement supporté par l'ESP32, il est converti

en ASCII. Chaque caractère du paragraphe est traité individuellement. Une fois les caractères en ASCII obtenus, ils sont traduits en Braille en utilisant un mappage spécifique, détaillé dans le premier chapitre. Ce mappage permet de transformer chaque caractère ASCII en son équivalent en Braille.

III.3.2 Étapes de la Simulation :

III.3.2.1 Configuration sur Wokwi :

- **ESP32 et 6 Servos** : Dans l'interface Wokwi, nous connectons l'ESP32 et les 6 servomoteurs selon le schéma de connexion requis. Les connexions comprennent les pins de contrôle des servos ainsi que l'alimentation et la mise à la terre.

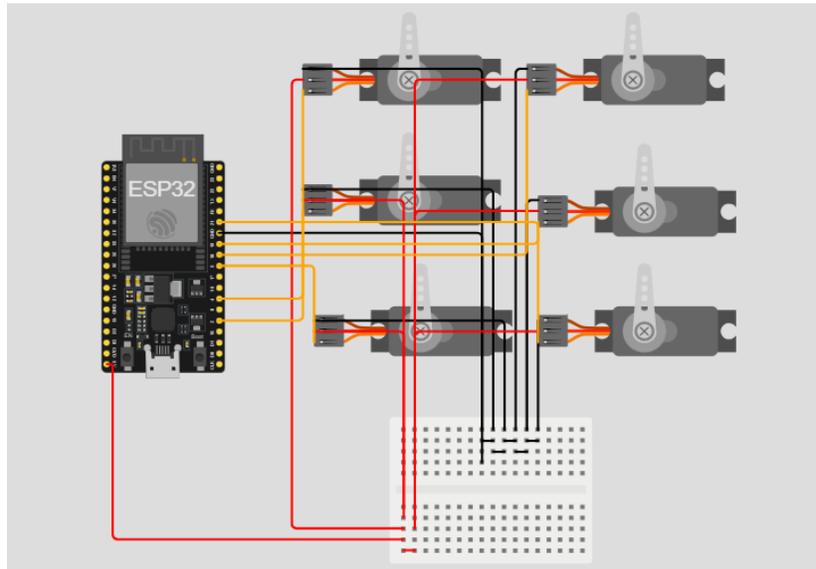


Fig. III. 7 Simulation de Circuit dans Wokwi

- **Exportation du schéma :** Une fois toutes les connexions établies, nous exportons le schéma de connexion sous forme de fichier diagram.json (annexe A). Ce fichier contient toutes les informations nécessaires sur les connexions matérielles configurées dans Wokwi

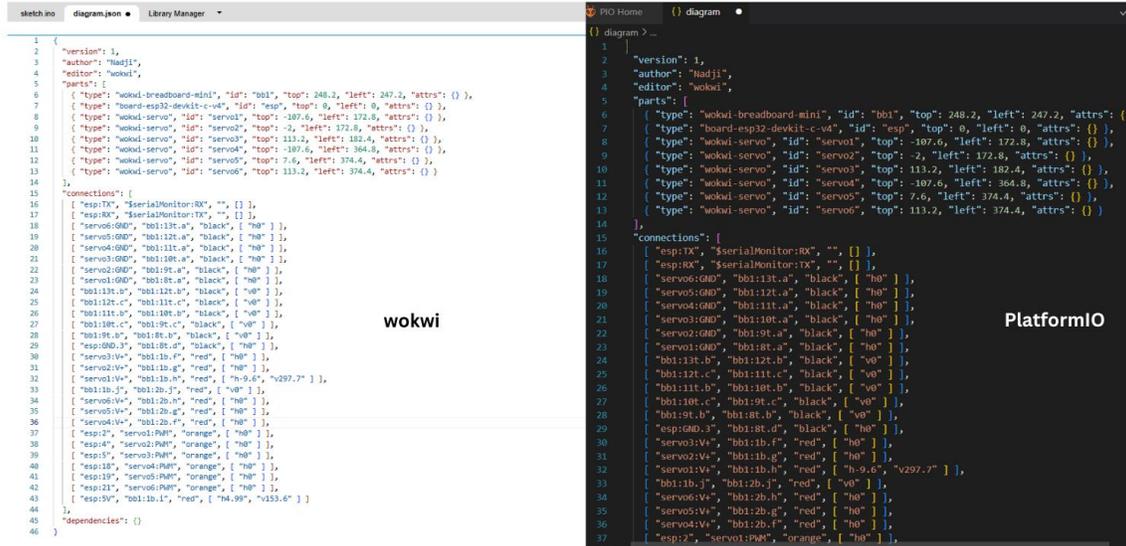


Fig. III. 8 Importation de diagram.json de Wokwi vers PlatformIO

III.3.2.2 Simulation dans PlatformIO :

- **Importation dans PlatformIO :** Le fichier wokwi.toml (annexe A) et diagram.json est importé dans PlatformIO, où nous continuons la simulation.

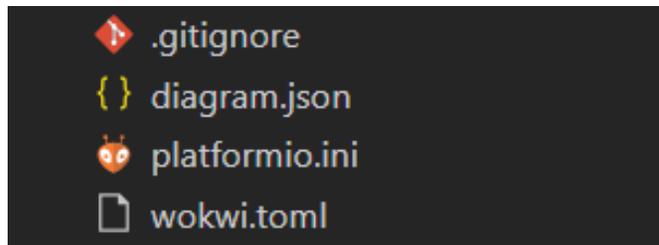


Fig. III. 9 Fichiers Importants pour la Simulation avec Wokwi dans VSCode

- **Compilation du Code** : Le code source, principalement contenu dans le fichier main.cpp (annexe A) pour l'ESP32, est compilé dans cet environnement. PlatformIO permet de compiler et de déployer le code de manière transparente, facilitant ainsi les tests de simulation.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR
Advanced Memory Usage is available via "PlatformIO Home > Project Inspect"
RAM: [ =          ] 14.6% (used 47740 bytes from 327680 bytes)
Flash: [ ===== ] 82.2% (used 1077569 bytes from 1310720 bytes)
===== [SUCCESS] Took 42.22 seconds =====
Terminal will be reused by tasks, press any key to close it.

```

Fig. III. 10 Compilation du Code

III.3.2.3 Test de la simulation :

Pour valider la simulation, nous utilisons le mot "نور" (Noor) que nous allons traduire en Braille.

- **Envoi du mot dans Firebase** : Le mot "نور" est envoyé dans Firebase via l'interface du site web. Ce processus implique la création d'un fichier texte contenant le mot et son téléchargement sur Firebase.



Fig. III. 11 Réception du Mot NOOR par Firebase depuis le site web

- **Connexion de l'ESP32 au Wi-Fi et à Firebase :** L'ESP32 se connecte au réseau Wi-Fi configuré et accède à Firebase pour récupérer le mot.

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SERIAL MONITOR

Initializing...
Connecting to Wi-Fi.....
Connected with IP: 10.13.37.2

Firebase sign-up successful
Setup completed.

```

Fig. III. 12 Connexion Wi-Fi, servomoteurs et Firebase

- **Détection et traitement du mot :** L'ESP32 détecte le mot téléchargé et le traite caractère par caractère. Chaque caractère est converti en ASCII puis traduit en Braille.

```

Received text:
رون
Translating letter: 646
Braille pattern: 0 1 1 1 0 1
Translating letter: 648
Braille pattern: 1 1 1 0 1 0
Translating letter: 631
Braille pattern: 0 1 0 1 1 1

```

Fig. III. 13 Détection du Mot par l'ESP32 et Séparation de Chaque Lettre

- **Mouvement des servos :** Les servomoteurs sont programmés pour se déplacer en fonction des caractères traduits. Chaque servo représente un point dans la cellule de Braille, et leur position indique la lettre traduite.

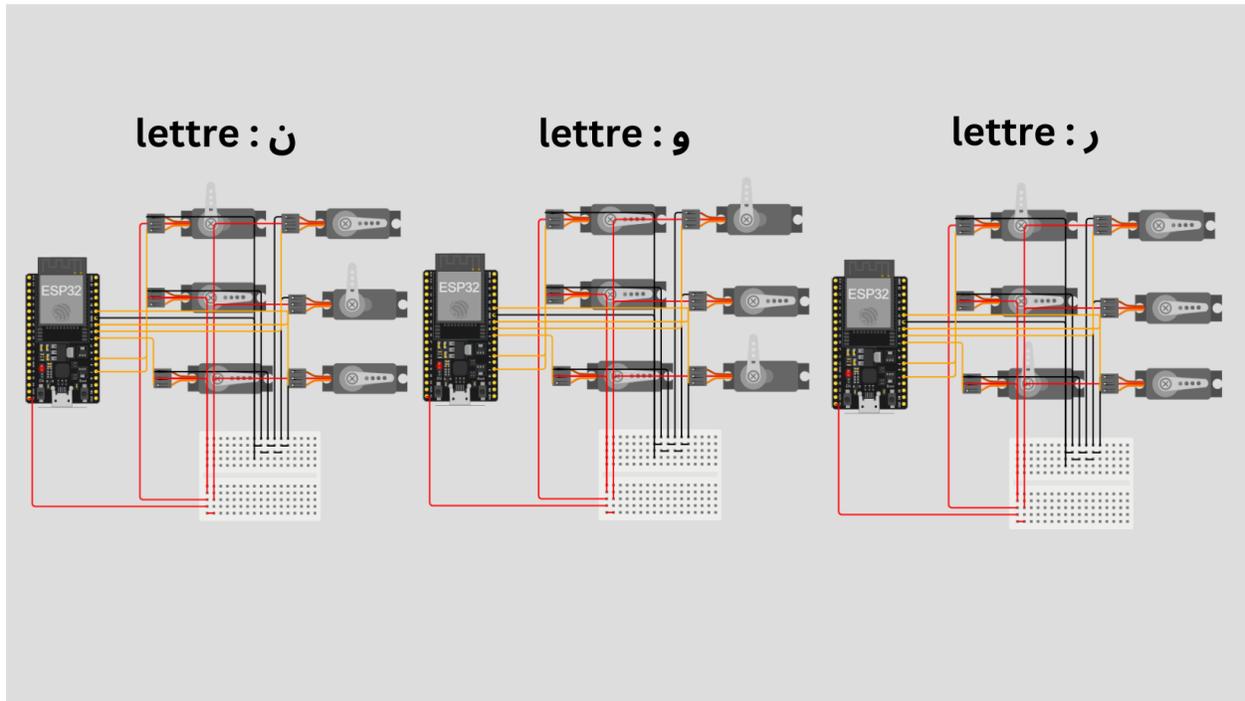


Fig. III. 14 Traduction de Chaque Lettre en Braille par l'ESP32

Une fois les servomoteurs en mouvement, ils représentent visuellement les caractères en Braille correspondant au mot "نور". Chaque servo se déplace selon les instructions codées, montrant les lettres traduites en Unicode et leur équivalent en Braille sur un affichage simulé.

La réussite de cette simulation est essentielle pour confirmer que le système NOOR fonctionne correctement dans un environnement virtuel. Elle assure que tous les composants, y compris l'ESP32, les servomoteurs, et la connexion à Firebase, interagissent comme prévu avant de passer à une phase de prototypage physique.

III.4 Impression 3D avec SolidWorks :

III.4.1 Conception des pièces dans SolidWorks :

La conception des pièces pour l'impression 3D est une étape cruciale dans le développement du système NOOR. SolidWorks, un logiciel puissant de CAO (Conception Assistée par Ordinateur), a été utilisé pour créer les composants nécessaires.

Les pièces conçues incluent des éléments de point court et de point long, la partie supérieure contenant les trous à travers lesquels les points se déplacent de haut en bas, la zone de placement des servos, et le corps principal qui abrite les autres composants et se ferme avec la partie supérieure.

Les pièces nécessaires incluent les points courts et longs, la section supérieure qui contient les trous où les points montent et descendent, la zone de placement des servomoteurs, et le corps principal qui contient les autres composants et ferme le système.

Les points courts et longs sont des broches utilisées pour représenter les points en Braille, permettant différentes configurations des caractères en Braille. La section supérieure est conçue pour s'adapter au-dessus du corps principal, offrant une apparence soignée et fonctionnelle.

La zone de placement des servos comporte des emplacements spécifiques pour maintenir les servomoteurs en place, assurant un mouvement précis des points en Braille. Enfin, le corps principal abrite l'ESP32, les servos et d'autres composants électroniques, conçus pour permettre un assemblage facile et un placement sécurisé de toutes les pièces.

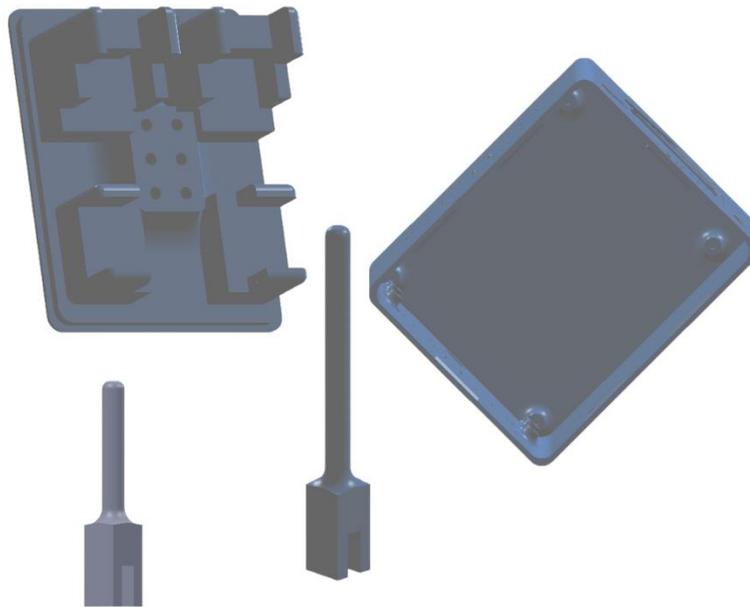


Fig. III. 15 Pièces modélisées avec SolidWorks pour NOOR

III.4.2 Conversion des conceptions pour l'Impression 3D :

Une fois les pièces conçues dans SolidWorks, l'étape suivante consistait à convertir ces conceptions du format .STL (stéréolithographie) au format .GCODE, qui est le langage utilisé par les imprimantes 3D. Cette conversion est essentielle pour traduire le modèle 3D en instructions que l'imprimante peut suivre. Cura, un logiciel d'impression 3D largement utilisé, a été utilisé pour ce processus de conversion. Voici les Processus étape par étape :

- Exporter les pièces conçues de SolidWorks au format .STL.
- Importer les fichiers .STL dans Cura.
- Configurer les paramètres d'impression (hauteur de couche, vitesse d'impression, structures de support, etc.).
 - Générer le fichier .GCODE.
 - Envoyer le fichier .GCODE à l'imprimante 3D pour l'impression.
 - Impression des Pièces

L'étape finale impliquait l'impression des pièces conçues à l'aide d'une imprimante 3D. Le processus nécessitait une surveillance attentive pour garantir la qualité et la précision des impressions. Les pièces imprimées étaient ensuite testées pour vérifier

leur ajustement et leur fonctionnalité afin de confirmer qu'elles répondaient aux spécifications de conception.

III.4.3 Processus impression 3D:

- Configurer l'imprimante 3D avec le filament requis.
- Charger le fichier .GCODE sur l'imprimante.
- Démarrer le processus d'impression et le surveiller pour détecter d'éventuels problèmes.
 - Une fois l'impression terminée, retirer les pièces et nettoyer les structures de support.
 - Tester les pièces pour s'assurer qu'elles s'assemblent correctement et fonctionnent comme prévu.

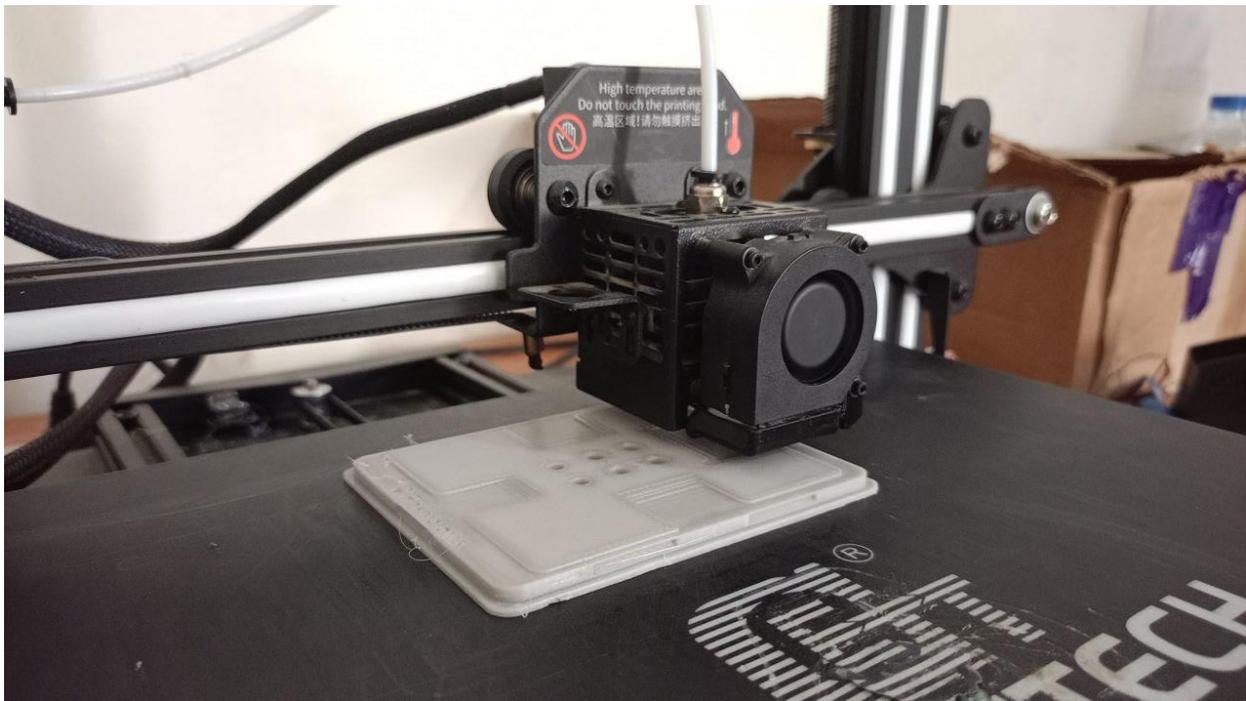


Fig. III. 16 Impression 3D des éléments du projet NOOR

Cette approche systématique de la conception, de la conversion et de l'impression des pièces a permis de garantir que les composants du système NOOR étaient précis, fonctionnels et prêts pour l'assemblage et les tests.

III.5 Réalisation Physique de NOOR :

III.5.1 Assemblage des Composants :

Connexion des Fils sur le PCB Manuel :

La première étape de la réalisation physique du système NOOR consiste à connecter les fils sur un PCB manuel, un circuit imprimé avec de nombreux points de connexion permettant de connecter les câbles manuellement. Le processus commence par la préparation du PCB manuel, en vérifiant les tracés et les points de connexion. Ensuite, les fils sont soudés sur le PCB en suivant le schéma de câblage préétabli. Chaque connexion est vérifiée pour s'assurer qu'elle est sécurisée et qu'il n'y a pas de courts-circuits.

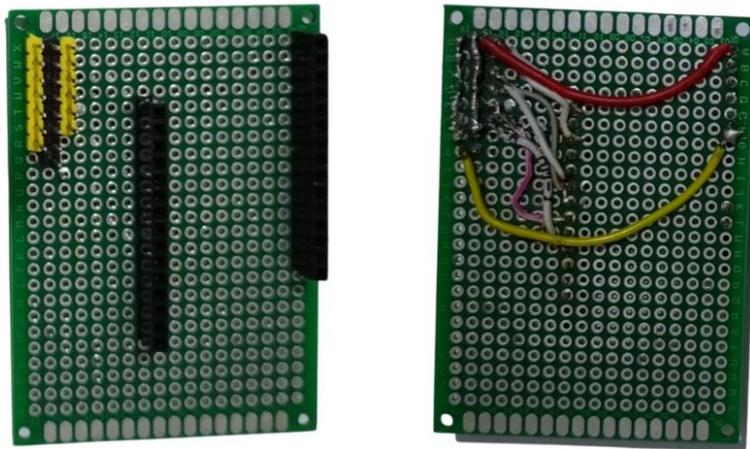


Fig. III. 17 Soudure manuelle des fils sur le PCB

Connexion des Éléments Électroniques :

Une fois le PCB préparé, les différents composants électroniques, tels que l'ESP32 et les servomoteurs, sont connectés. L'ESP32 est fixé au PCB avec les broches correctement alignées et sécurisées. Les servomoteurs sont placés dans les emplacements prévus et connectés aux circuits de contrôle sur le PCB. Toutes les connexions sont soigneusement vérifiées pour garantir leur exactitude et leur bon fonctionnement.

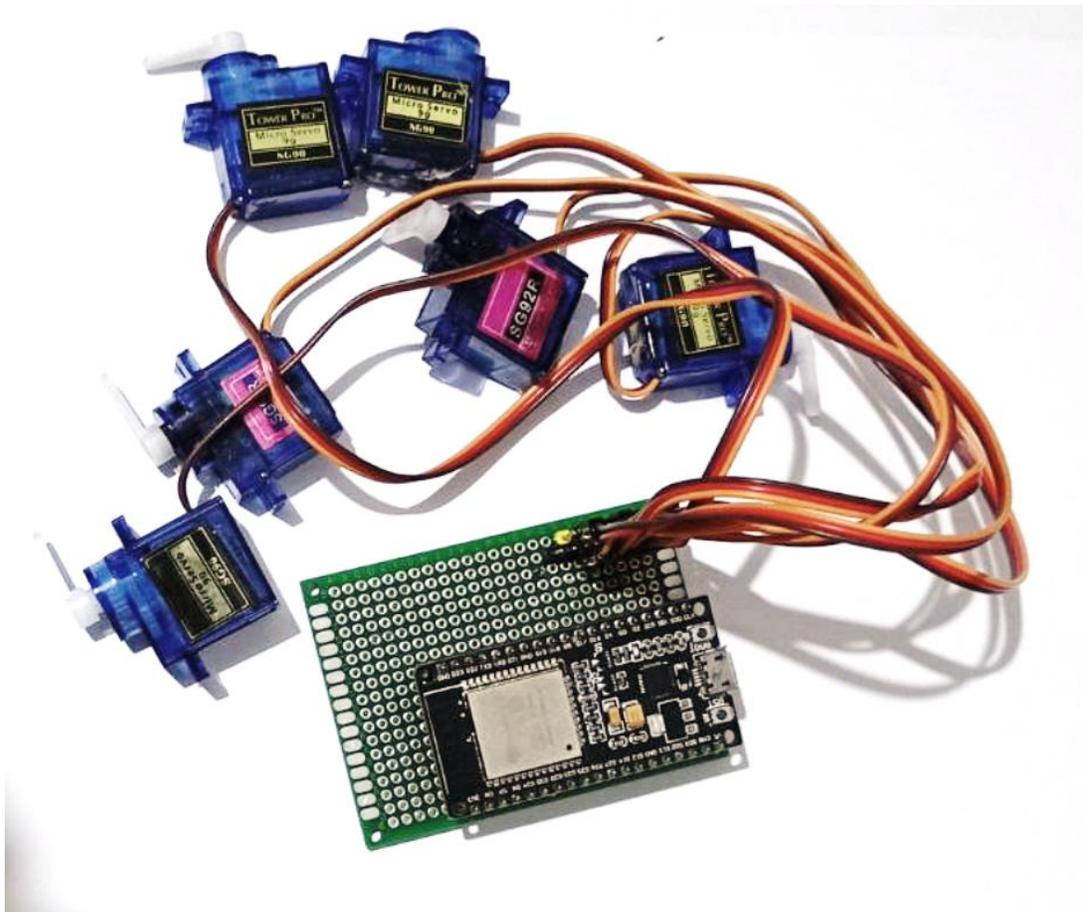


Fig. III. 18 Câblage des éléments électroniques

III.5.2 Test du Système :

Après avoir effectué toutes les connexions, le système est testé pour vérifier son bon fonctionnement. Cela comprend la vérification des mouvements des servomoteurs, la communication avec l'ESP32 et le fonctionnement global du système NOOR. Les tests consistent à alimenter le système et à vérifier les tensions aux points de connexion clés, à tester les servomoteurs pour s'assurer qu'ils répondent correctement aux commandes, et à vérifier la communication entre l'ESP32 et les autres composants via les interfaces logicielles. Un test de fonctionnement complet est ensuite réalisé pour s'assurer que le système NOOR fonctionne comme prévu.

Après les tests réussis, le système NOOR fonctionne avec succès, répondant à toutes les spécifications et aux résultats de simulation.

III.6 Conclusion :

Ce chapitre résume les aspects essentiels du projet. Nous avons simulé une interface utilisateur fluide et intuitive, facilitant l'utilisation, surtout pour les déficients visuels. Cette interface assure une interaction facile et efficace avec le système.

La simulation des composants électroniques via Wokwi a été cruciale pour identifier et résoudre les problèmes avant la phase de réalisation physique, évitant ainsi des erreurs coûteuses sur le PCB manuel. Cette étape nous a permis d'assembler les composants en toute confiance, assurant le bon fonctionnement du système.

La conception et l'impression 3D des pièces, comme les points braille et les supports pour servomoteurs, ont joué un rôle clé. Ces pièces offrent précision et praticité, améliorant l'efficacité et l'accessibilité pour les utilisateurs aveugles.

Enfin, la réalisation physique du système a impliqué l'assemblage et la connexion des composants. Les tests finaux ont validé le bon fonctionnement du système, confirmant l'efficacité des simulations et modélisations.

En résumé, ces étapes ont été indispensables pour garantir que le traducteur NOOR fonctionne de manière fiable et efficace, répondant aux besoins des utilisateurs et facilitant leur interaction avec le dispositif.

Conclusion Finale et Perspectives :

Notre projet NOOR, le traducteur portable arabe-braille en temps réel basé sur l'IoT, représente une avancée notable dans les technologies d'assistance pour les malvoyants en Algérie. Ce mémoire explore les bases du Braille, les composants matériels et logiciels utilisés, ainsi que les étapes de simulation et de réalisation du projet.

Le premier chapitre offre un aperçu détaillé de l'histoire et de l'évolution du Braille, en mettant l'accent sur son adaptation à la langue arabe et les défis spécifiques en Algérie. En deuxième chapitre en se penche sur les composants matériels, tels que l'ESP32 et les servomoteurs, ainsi que les outils logiciels comme Visual Studio Code, PlatformIO, et Firebase, qui ont été cruciaux pour le développement de NOOR et le troisième chapitre décrit le processus de conception, de simulation avec Proteus, et de réalisation, en soulignant les résultats obtenus et l'impact positif sur l'accessibilité des ressources éducatives pour les malvoyants.

Malgré ces avancées, NOOR reste un prototype avec certaines limitations. Actuellement conçu pour un seul utilisateur, il utilise uniquement le Wi-Fi pour la connexion. Le dispositif ne permet pas de relire une ligne spécifique, mais seulement de reprendre depuis le début. De plus, l'ESP32 chauffe rapidement, posant un risque de dommages.

Pour améliorer les performances et la fonctionnalité du traducteur NOOR, plusieurs améliorations sont envisagées. L'ajout d'un Raspberry Pi permettra d'accroître les performances et d'accélérer l'analyse des données. L'hébergement multi-utilisateurs permettra des connexions via Bluetooth ou Wi-Fi, rendant le dispositif plus versatile. De plus, augmenter la taille de la cellule Braille d'une à vingt offrira une meilleure représentation des textes. L'intégration de boutons de lecture et de navigation rapide (playback et playnext) rendra l'utilisation plus fluide. Enfin, améliorer le support de différents formats de fichiers et corriger les bugs existants sont essentiels pour optimiser l'efficacité du traducteur.

L'avenir des technologies d'assistance est prometteur, avec un énorme potentiel pour l'IoT dans le développement de dispositifs innovants et accessibles. L'amélioration continue de ces technologies permettra de répondre plus efficacement aux besoins des malvoyants. En intégrant des solutions comme NOOR, il est possible de transformer radicalement l'accès à l'information et à l'éducation pour les malvoyants, non seulement

en Algérie mais aussi à l'échelle mondiale. Le futur de ces technologies réside dans leur capacité à devenir plus inclusives, abordables et performantes, offrant ainsi une meilleure qualité de vie aux personnes en situation de handicap.

En conclusion, bien que NOOR soit encore un prototype, ses contributions sont significatives et ouvrent la voie à des améliorations futures. L'extension des fonctionnalités, l'amélioration des performances et l'augmentation de l'accessibilité sont des axes de recherche importants pour les prochaines étapes du développement.

ANNEXES :

ANNEXE A : Programme de l'ESP32 dans VSCode



Fig. Annexe A : Programmation de l'ESP32

ANNEXE B : Programme de l'interface



Fig. Annexe B : Scripts Front-End de l'Interface Web

Références :

- [1] S. Suleiman, *Psychologie des personnes ayant des besoins spéciaux : méthodes éducatives et soins éducatifs*, Le Caire : Bibliothèque Zahra al-Sharq, 2018.
- [2] V. Tourtchine, «Biography of Louis Braille and Invention of the Braille Alphabet,» ResearchGate, 2023.
- [3] C. Clark-Bischke and J. B. Stoner, «An Investigation of Spelling in the Written Compositions of Students Who Read Braille,» *Journal of Visual Impairment & Blindness*, vol. 107, 2017.
- [4] RTBF, «Handicap visuel et nouvelles technologies : pourquoi le braille reste essentiel à l'heure du numérique,» RTBF, janvier 2023.
- [5] A. Abdel Majid, «Développement de stratégies cognitives avancées dans le développement de certaines compétences supérieures nécessaires à la lecture chez les élèves de première année de lycée,» *Journal de la Lecture et du Savoir*, Association égyptienne de la lecture et du savoir, Université Ain Shams, Le Caire, vol. 2, pp. 195-232, 2015.
- [6] A. A. Al-Sayed Al-Shakhs, *Effet des stratégies de soins sur le niveau des enfants chez les aveugles et leurs réponses dans le handicap visuel*, *Journal des Recherches sur les obstacles à l'enfance - Université d'Al-Azhar*, 2020.
- [7] Aleteia, «Louis Braille, célèbre aveugle, inventeur génial... et musicien surdoué,» Aleteia, 2021.
- [8] J. A. Durando and D. Wormsley, «Evaluating Training and Implementation of the Individualized Meaning-Centered Approach to Teaching Braille Literacy,» *Journal of Visual Impairment & Blindness*, vol. 103, no. 3, 2019.
- [9] Braille Magazine, «Braille, l'indétrônable inventeur,» Braille Magazine, 1 décembre 2023.
- [10] ICOE, «The Story of Louis Braille: Inventor of the Braille Code,» ICOE, 3 février 2021.

- [11] Enterprise Bank, «Diversity, Equity, and Inclusion (DEI) e-Zine,» Enterprise Bank, 4 janvier 2023.
- [12] M. Jiménez-Olea, «Biography of Louis Braille and invention of the braille alphabet,» SpringerLink, 9 décembre 2022.
- [13] N. Coppins and F. Barlow-Brown, «Reading Difficulties in Blind, Braille-Reading Children,» *British Journal of Visual Impairment*, vol. 24, no. 1, 2016.
- [14] Paediatric, «The Story of Louis Braille and His Gift of Literacy,» *Pediatric RRF*, 15 mai 2023.
- [15] *Frontiers in Neuroscience*, «Smart Assistive Technologies,» *Frontiers in Neuroscience*, 2023.
- [16] J. A. Durando and D. Wormsley, «Evaluating Training and Implementation of the Individualized Meaning-Centered Approach to Teaching Braille Literacy,» *Journal of Visual Impairment & Blindness*, vol. 103, 2009.