

DOSSIER DE CONCEPTION

Projet MAGIMBA

Maître d'ouvrage (enseignant responsable) :

Andres ARCINIEGAS MOSQUERA

Titulaire (équipe de conception) :

Gabriel ERBE

Djibril EL KARANI

Enes YGIT

Date de rédaction :

09/02/2025

Table des matières

I Introduction.....	5
1. Objectifs du document	5
2. Portée du document.....	5
II Domaines d'application	5
1. Objectifs du système	5
2. Explication globale du système	6
2.1 Composition du Magimba	6
2.2 Analyse fonctionnelle technique.....	7
3. Contraintes générales de conception.....	8
III Documents de référence	10
IV Outils et notations.....	11
1. Environnement et outils utilisés.....	11
1.1 Outils logiciels.....	11
1.2 Outils physiques	12
1.3 Micro-contrôleur	12
2. Notations utilisées.....	13
V Conception générale	14
1. Les actionneurs.....	14
1.1 Présentation d'un actionneur type	14
1.2 Schéma électrique d'un actionneur	16
1.3 Composants d'un actionneur	17
2. L'alimentation	20
2.1 Présentation	20
2.2 Conception technique	20
2.3 Composants.....	21
2.4 Schéma électrique simplifié sur le projet.....	22
3. L'affichage des modes de fonctionnement.....	22
3.1 Présentation de l'écran.....	22
3.2 Schéma électronique simple	22
4. Langages utilisés.....	23

5.	La programmation	25
5.1	Fonction principale	26
5.2	Fonction Automatique.....	27
5.3	Fonction Semi-automatique	27
5.4	Logique de l'enregistrement.....	28
VII.	Tables des illustrations	29

Historique des révisions de ce document

Référence	Révision	Date	Auteur(s)	Description
Dossier de Conception MAGIMBA	00	03/02/2025	Equipe de conception	Version projet initial
Dossier de Conception MAGIMBA	005	08/02/2025	Equipe de conception	Conception Générale PANFi
Dossier de Conception MAGIMBA	01	10/02/2025	Equipe de conception	Conception Générale PAFi
Dossier de Conception MAGIMBA	02	13/02/2025	Equipe de conception	Révision batterie et écran PAFi

Barème d'évaluation du document

Critères	Étudiant(s)	Binome 1	Binome ...	Binome n
Présentation du document	4	0	0	0
Document pdf	0,5			
Maitrise du traitement de texte	1			
Langage technique	1			
Référencement des figures et tableaux	1			
Gestion des versions	0,5			
Introduction - mise en contexte	7	0	0	0
Structure du document (par ss-système, ou thématique)	2			
Explication globale du système (analyse fonctionnelle tech)	4			
Identification des contraintes de conception	1			
Justification des choix de conception	9	0	0	0
Justification des choix de composants élec	3			
Justification des choix d'architecture information embarquée	3			
Justification des outils logiciels utilisés (langage de prog)	3			
Détails de conception	10	0	0	0
Détails de conception élec (calculs, fiches tech, etc)	4			
Détails de conception info (algo, protocoles de comm, brochures)	4			
Détails de conception méca (prototypage)	2			
Total, ramené à 20 et arrondi	20	0	0	0

Notation à viser pour validation par niveau	
0	Non acquis
4	Insuffisant
8	En cours d'acquisition
12	Acquis
16	Maîtrisé
20	Expert

I Introduction

1. Objectifs du document

Ce document présente la conception détaillée du projet de 3ème année du BUT GEII : la création d'un instrument automatisé, aussi appelé « orchestrion ». Cette phase de conception intervient après A REMPLIR, qui consiste en la réalisation d'un prototype d'orchestrion.

2. Portée du document

Ce document participe à la construction des prototypes « Alpha » et « Beta » du Magimba et servira de support dans la phase de fabrication physique et informatique de ces prototypes.

II Domaines d'application

1. Objectifs du système

L'objectif de ce projet est de fournir à l'IUT de Cergy-Pontoise un orchestrion fonctionnel qui prendra ici la forme d'un kalimba magnétique : le Magimba.

Cet orchestrion contient 4 **modes de fonctionnement** principaux : un **mode manuel**, où les utilisateurs peuvent jouer de l'instrument de façon normale, un **mode semi-automatique** où ils peuvent enregistrer les musiques qu'ils produisent et les faire rejouer, un **mode automatique** qui permettra de faire jouer une partition pré-enregistrée (destiné aux tests du PA et PB), un **mode à distance**, accessible depuis une interface web (destiné à être présent dans le PB), et un **mode chef d'orchestre** (destiné à être implémenté pour le PB).

Le **mode manuel** permet à un utilisateur de jouer de l'instrument en ignorant l'utilisation de ses propriétés supplémentaires qui font de lui un orchestrion.

Le **mode semi-automatique** permet de récupérer les touches jouées en conservant la note et sa durée, puis de pouvoir faire jouer à l'instrument le morceau ainsi enregistrer.

Le mode **automatique** est assez similaire au mode semi-automatique puisqu'il permet de restituer un morceau pré-enregistré (codé en dur dans le code) par l'orchestrion.

Le **mode à distance** permet de jouer Magimba en se passant de l'interface physique prévue en utilisant un appareil électronique pouvant se connecter au wifi et comportant une IHM suffisante pour pouvoir contrôler le Magimba comme il est prévu.

Le **mode chef d'orchestre** permettra au Magimba de se connecter à une table de mixage audio afin de pouvoir permettre aux utilisateurs de ce mixeur audio de traiter les signaux de plusieurs instruments simultanément pour recréer un « orchestre ».

2. Explication globale du système

2.1 Composition du Magimba

Le projet de Magimba est composé des éléments suivants :

- Un micro-contrôleur.
- Des lames de kalimba spéciales.
- Un module d'alimentation.
- Des solénoïdes.
- Dse composants supplémentaires mineurs dans l'explication du principe de fonctionnement (comme des vis ou des résistances).

Ces **éléments** peuvent être modélisés de manière simplifiée comme sur la figure suivante pour une meilleure compréhension du système :

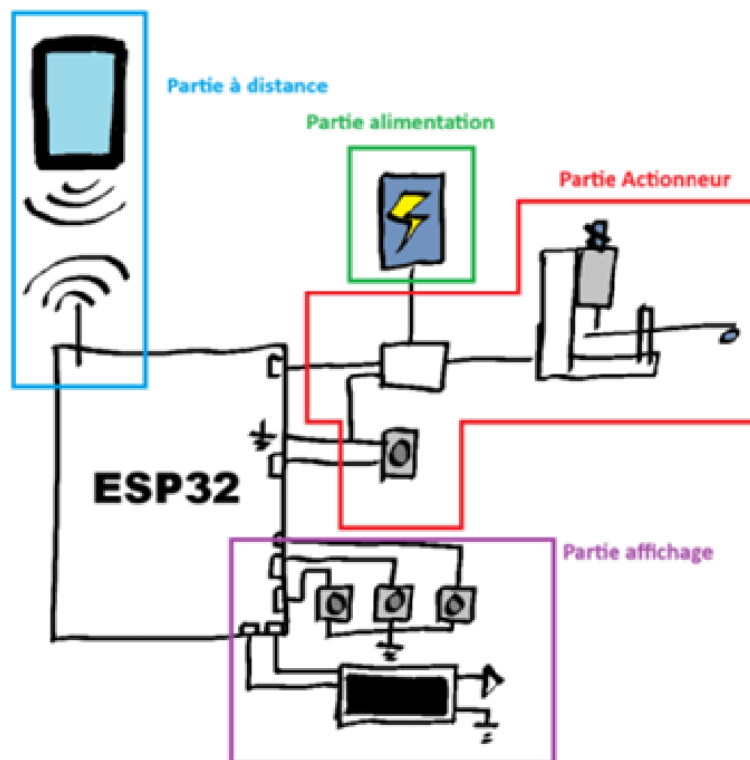


Figure 1 : Croquis simplifié du système global pour le PB

2.2 Analyse fonctionnelle technique

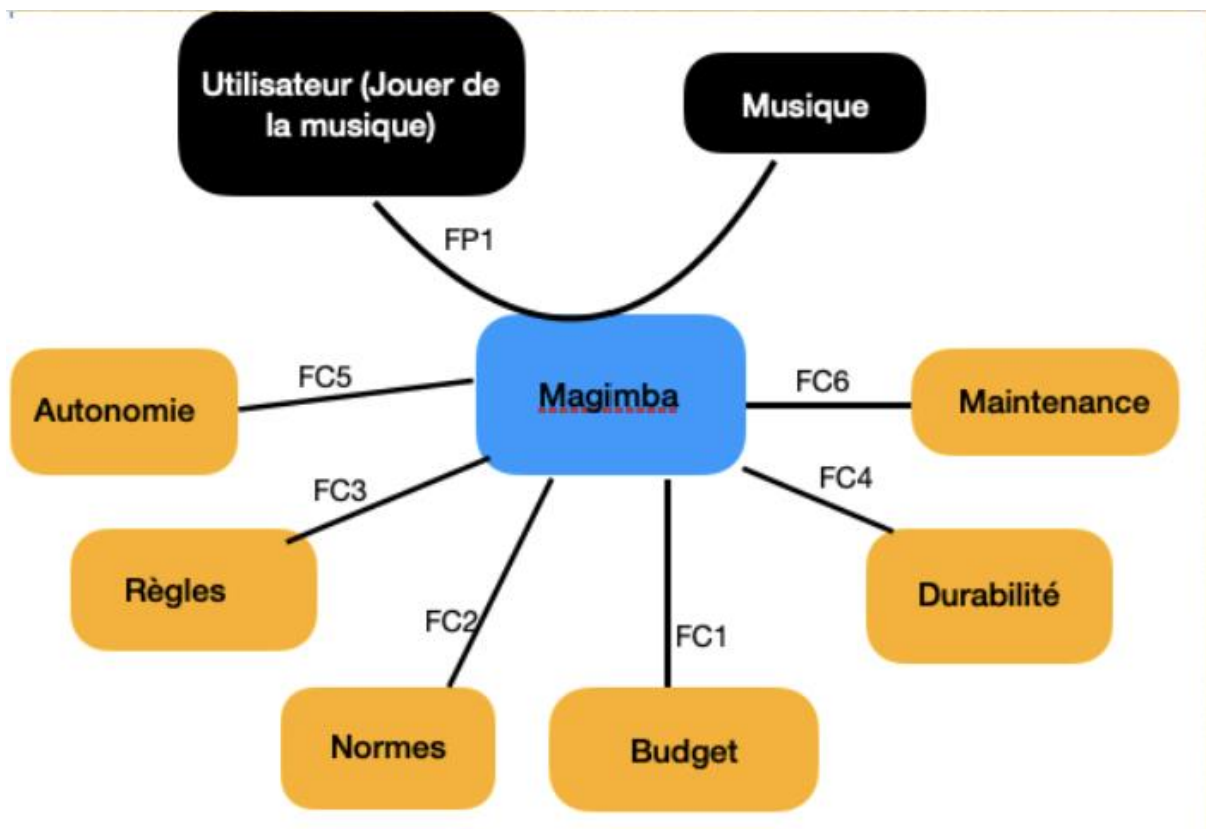


Figure 2 : fonctions principales et les fonctions de contraintes du projet

Fonctions	Énoncé de la fonction	Explication de la fonction
FP1	Utilisateur (Jouer de la musique)	
FC1	Budget	L'ensemble des composants non-fournis par l'IUT ne doivent (à l'achat) pas dépasser un certain seuil.
FC2	Normes	L'objet doit correspondre aux normes (Ex : CEM)
FC3	Règles	Le projet doit éviter d'enfreindre les lois françaises et internationale lors de sa réalisation

FC4	Durabilité	Les différentes parties de l'orchestration doivent être suffisamment solides pour éviter un dysfonctionnement dû à une mauvaise manipulation
FC5	Autonomie	L'orchestration ne doit pas dépendre d'une prise secteur pour pouvoir fonctionner, d'où la prévision d'une batterie
FC6	Maintenance	L'objet doit pouvoir rendre sa maintenance accessible aux personnes tierces (Ex : Documentation associée)

3. Contraintes générales de conception

Plusieurs **contraintes** provenant de différentes sources sont à prendre en compte dans la phase de conception du système. Ci-dessous, un récapitulatif des contraintes imposées par le maître d'ouvrage dans le cahier des charges :

- Contraintes de **fonctionnalités** :

- Pour éviter des interférences avec d'autres appareils électroniques, le Magimba doit être installé à une certaine distance de dispositifs sensibles aux champs électromagnétiques.
- Le produit devra être conçu pour faciliter la **maintenance** et les réparations, notamment en cas de défaillance des composants comme les **électroaimants**, les transistors **MOSFET**, ou le **microcontrôleur**.
- Le Magimba doit être fabriqué sans partir d'un instrument acheté.
- L'instrument doit être capable de fonctionner sous plusieurs **modes** (manuel, semi-automatique, automatique, et à distance ...).
- L'instrument doit comporter au minimum une octave avec les dièses. Ce qui fait un total de **12 notes a minima**.
- L'orchestration doit être **autonome en énergie** et respecter les normes de sécurité électrique. Ce qui signifie qu'il devrait pouvoir ne pas dépendre d'une prise secteur à tout temps.

- Contraintes de **coûts** :

- Les **matériaux** et les **composants** (tant électroniques que physiques) doivent au maximum être déjà présent dans l'enceinte de l'IUT afin d'éviter des dépenses supplémentaires.

III Documents de référence

Les documents suivants sont à utiliser en référence avec la lecture de ce document :

- Le **Cahier des Charges** dans sa version la plus récente : il contient les spécifications initiales des exigences du **maître d'ouvrage**. Sa référence actuelle est : **MAGIMBA-CC-V03NF**
- Le **Dossier de Fabrication** dans sa version la plus récente : il contient les données utilisées pour construire actuellement le PA. Sa référence actuelle est : **MAGIMBA-DF-V01NF**
- Le **Dossier de Planification** dans sa version la plus récente : il contient les données utilisées pour prévoir l'organisation du projet dans le temps ainsi que les rôles des différents membres de l'équipe dans celui-ci. Sa référence actuelle est : **MAGIMBA-DP-V01NF**

IV Outils et notations

1. Environnement et outils utilisés

Dans le cadre de la réalisation du projet de Magimba, il est nécessaire de savoir que celui-ci se compose de plusieurs parties qui nécessitent des outils différents, c'est de cette manière que nous allons les catégoriser entre « **Outils logiciels** » qui regroupe les outils principalement utilisés pour le développement informatique et la création de schémas électriques, et « **Outils physiques** » qui regroupe alors les outils employés dans un contexte plus manuel.

1.1 Outils logiciels

Pour mener à bien le projet de **Magimba**, il est nécessaire d'avoir des logiciels capables de mener à bien certaines actions. Bien qu'il en existe pour chacune des catégories auxquelles les prochains logiciels appartiennent, notre expérience dans leur utilisation, leur accès facile (dans notre situation) et leurs spécificités nous ont guidé dans nos choix.

a. Logiciel Proteus

Le logiciel **Proteus** est l'outil principal que nous utiliserons pour la conception de circuit électronique du projet. Celui-ci, au contraire d'autres logiciels du même type comme **Kicad**, permet de faire des simulations et des schémas en simultané. Sans compter, en restant sur l'exemple **Kicad**, que la bibliothèque de simulation de **Proteus** est complète, et qu'il n'y a pas de besoin répété de trouver par soi-même les modèles de simulation.

Nous pouvons par ailleurs directement faire des circuits de simulation et la conception de circuit une fois ce point réglé.

L'un des autres points forts du logiciel **Proteus** est que contrairement à **Altium** par exemple, il n'y a pas besoin de prendre une licence pour le projet uniquement, puisque tous les ordinateurs de l'IUT (département GEII) possèdent.

b. Logiciel Visual Studio Code

Visual Studio Code est un éditeur de code extensible et léger développé par Microsoft. Nous avons choisi cet environnement de développement plutôt qu'ArduinoIDE pour les fonctionnalités supplémentaires que VSCode apportait. Celle-ci peuvent se lister de la manière suivante :

- Mise en évidence de la syntaxe
- Fonctionnalité de débogage
- Programmation dans plusieurs langages (C++, C#, Java, Python, PHP, etc.)

c. Autodek fusion

Autodesk fusion pour la modélisation 3D qui nous servira pour concevoir des éléments sur mesure pour notre Magimba. Ce logiciel a l'avantage d'être gratuit et suffit amplement pour l'utilisation que nous souhaitons en faire.

1.2 Outils physiques

Afin de pouvoir réaliser, entre autres, la partie physique du Magimba et ces circuits électroniques originaux. L'IUT nous a mis à disposition un atelier de travail et donné un accès à un set d'outils. Voici une liste des principaux instruments qui sont prévus d'être utilisés :

- **Scie** : Pour les travaux manuels sur la partie physique du Magimba.
- **Marteau** : Pour les travaux manuels sur la partie physique du Magimba.
- **Tournevis** : Pour les travaux manuels sur la partie physique du Magimba.
- **Perceuse** : Pour les travaux manuels sur la partie physique du Magimba.
- **Plaque Lab** : Pour contenir les circuits électronique (Analogique).
- **Fer à souder** : Pour passer d'une plaque lab à un format de carte imprimée.
- **Imprimante 3D** (Ultimaker 2) : Cette machine nous permettra de réaliser des « morceaux » sur mesure de la partie physique, indispensable pour faciliter une implémentation correcte des solénoïdes par exemple.

1.3 Micro-contrôleur

Pour la partie de l'informatique embarquée de ce projet, nous avons décidé de nous baser sur un **micro-contrôleur ESP32**.

Pour comprendre le choix de la carte, nous devons prendre en compte certains éléments :

- **Le nombre de broches utilisées** :

Puisque nous souhaitons avoir **12 notes**, nous aurons besoin de 12 entrées/sorties reliées aux solénoïdes, le même nombre pour les **boutons d'activation**. Deux pour **l'écran** que nous voulons rajouter, et trois supplémentaires (au minimum) pour les **boutons du menu**. Ce qui nous fait un total de 29 entrées/sorties au minimum.

- **La taille de la mémoire :**

Nous avons prévu, dans le cadre de l'enregistrement du mode semi-automatique, que la RAM (nul besoin de garder cet enregistrement après l'arrêt de l'appareil) doit être d'au moins 3.2 Kbit pour un enregistrement de 10 secondes avec un temps d'échantillonnage de 0.05 secondes.

- **La connectivité sans-fil de la carte :**

Nous recherchons une carte micro-processeur qui intègre la gestion du Wifi et Bluetooth.

L'**ESP32** étant suffisant pour toutes ces conditions et étant accessible à l'IUT, notre choix s'est donc arrêté sur ce modèle de carte.

Référence : <https://fr.wikipedia.org/wiki/ESP32>

2. Notations utilisées

La terminologie utilisée dans le projet MAGIMBA est disponible dans le Cahier des Charges à partir de sa seconde version. Les acronymes qui sont les plus couramment utilisés sont PA (Prototype Alpha), PB (Prototype Beta), Fi (Fini), et NFi (Non fini).

Chaque dossier respecte la notation suivante pour le référencement :

MAGIMBA-[Initiales du dossier]-V[Numéro de version][NF (Non Finale) ou F (Finale)]

*Exemple de notation pour la deuxième version du Cahier des Charges non finale : **MAGIMBA-CC-V02NF***

V Conception générale

1. Les actionneurs

1.1 Présentation d'un actionneur type

Dans le projet de Magimba, nous avons conçu un système modulaire d'actionneurs, de telle manière à ce qu'ils puissent être facilement entretenus par la suite lors de la maintenance. Cependant, cette conception ne sera qu'intégrée sur le PB et non à partir du PA. Vous pourrez retrouver ci-dessous une représentation de l'idée d'un actionneur type dans ce système modulaire :

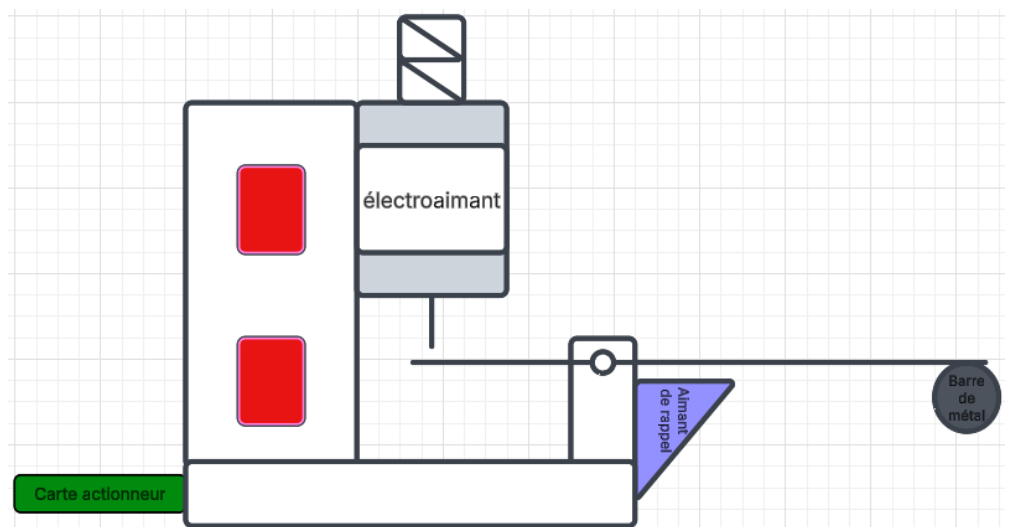
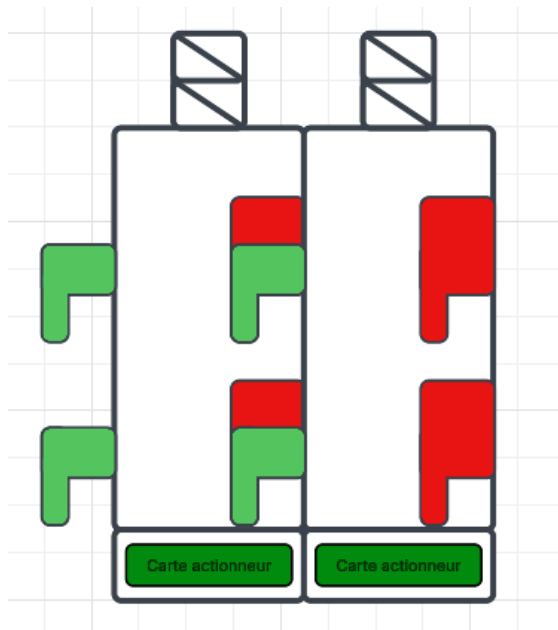


Figure 3 : : Croquis visuel du dispositif, vue de côté (Réalisé sur LucidChart)



Les parties rouges correspondent aux connecteurs dits « femelles » et les parties vert-clair, aux connecteurs « mâles », afin de pouvoir avoir un système facilement gérable dans les actions de maintenance prévues.

Figure 4 : Croquis visuel du dispositif, vue de derrière (Réalisé sur LucidChart)

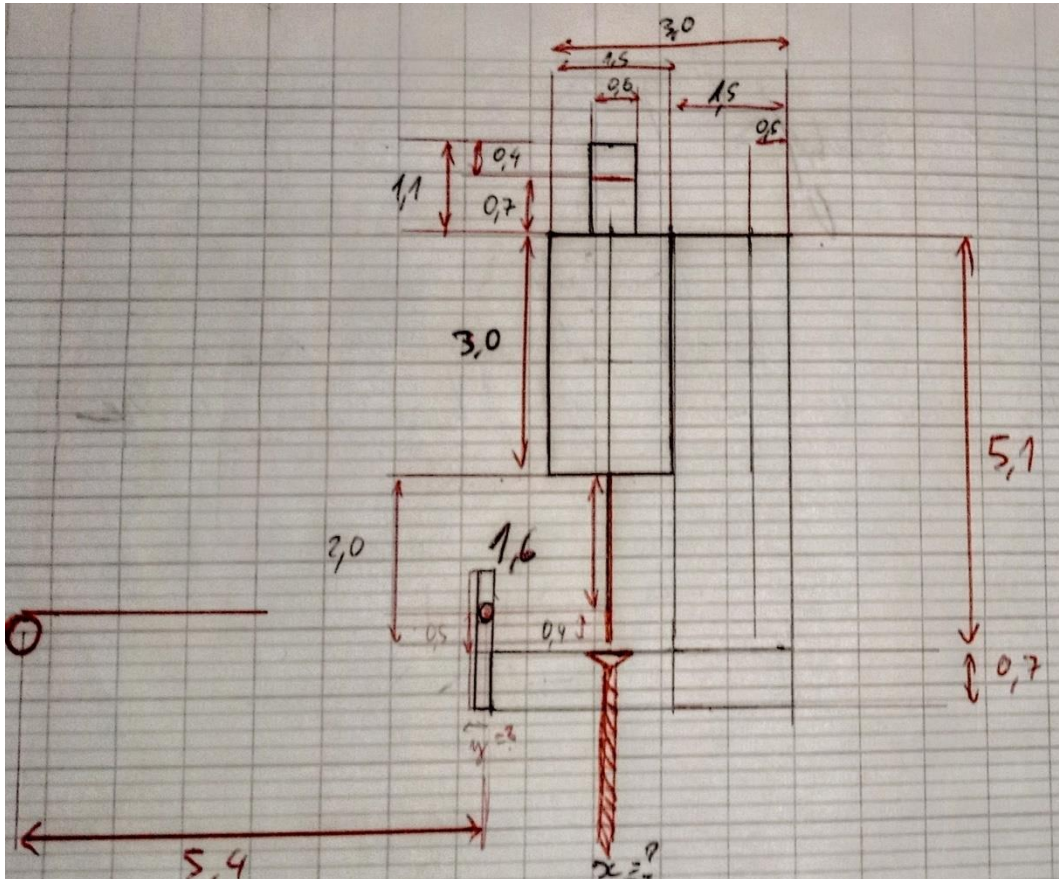


Figure 5 : Croquis de dimensions du dispositif, vue de côté

(Les chiffres donnés sont en centimètre (cm))

Ainsi, dans un cas où nous ne pouvons accéder à une gamme complète d'actionneurs, nous pouvons en rajouter plus tard sans problèmes majeurs comme la révision complète du système physique, et seulement quelques changements de variables générales dans le code afin d'accueillir ces modifications.

1.2 Schéma électrique d'un actionneur

Voici le schéma électrique d'un actionneur type :

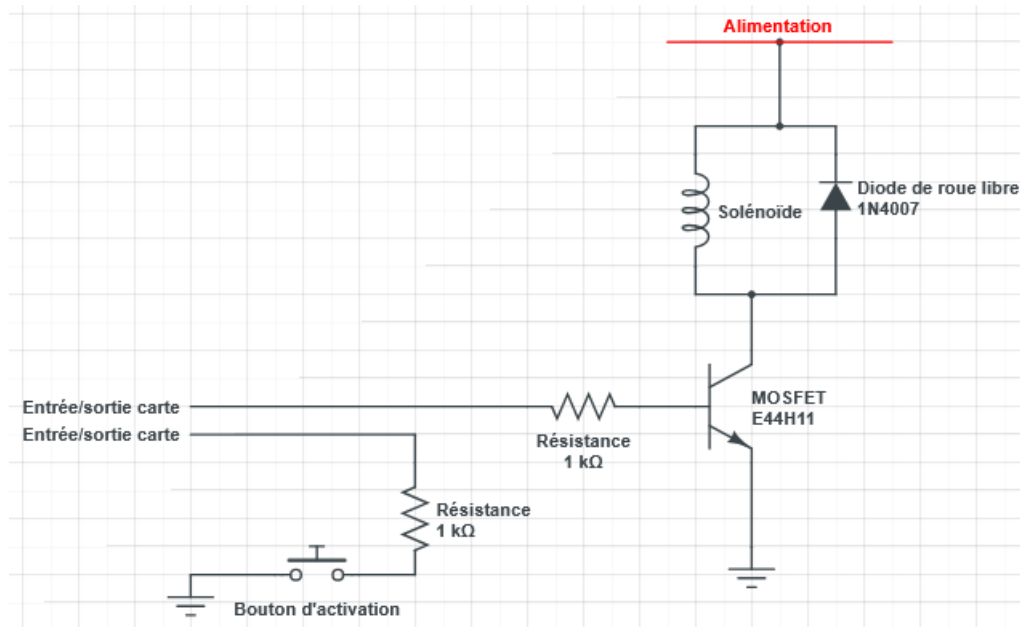


Figure 6 : Schéma électrique d'un actionneur type (réalisé avec CircuitLab)

Nous pouvons alors faire l'inventaire de ce qui compose ce système :

- 2 **résistances** de 1k Ohms.
- Un **bouton**.
- Un **solénoïde**.
- Un **transistor MOSFET**.
- Une **diode**.

Il est prévu que nous retrouvions la partie liée au MOSFET sur la carte actionneur que nous pouvons voir dans sur le croquis visuel (rectangle vert-sombre) des croquis précédents. En grande partie puisque cela permettra de faciliter les opérations de maintenance de la partie électronique analogique de ce morceau du système d'actionneur.

1.3 Composants d'un actionneur

a. Composants physiques principaux

Les lames de métal

Le principe d'émission de sons dans un kalimba repose sur la vibration d'une lame de métal dont le matériau et les dimensions physiques de la partie vibrante déterminent la fréquence de son émis (et ainsi sa note).

Les lames de kalimba sont généralement en **acier inoxydable** ou en **acier carboné**. Bien que la différence sonore entre ces deux types de métal soit minime, l'**acier inoxydable** a l'avantage d'être **plus résistant** à la corrosion, ce qui lui permettra d'avoir une **meilleure pérennité**. C'est ce en quoi nos lames seront donc faites.

Concernant leurs dimensions, cet instrument n'étant que peu commun, nous n'avons pu justifier les dimensions des lames d'une autre manière que celle empirique, empruntée par le créateur du modèle de Magimba dont est inspiré le projet.

Un autre point limitant est le fait qu'il manque les informations nécessaires pour créer les lames des dièses pour respecter le cahier des charges donné.

Note (gamme de Do majeur)	Do	Ré	Mi	Fa	So	La	Si	Do
Longueur (mm)	101	94	88	86	81	77	73	71
Largeur (mm)	5	5	5	5	5	5	5	5
Épaisseur (mm)	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7

Tableau répertoriant les dimensions des lames de Magimba et leur note associée

Référence : <https://www.youtube.com/watch?v=7LK3aFS5-vQ>

Les aimants de rappel

Le principe du Magimba repose sur l'idée d'utiliser des aimants pour remplacer les barres de maintien des kalimbas traditionnel. Ces aimants-ci auront donc comme principale fonction d'attirer à eux les lames de métal utilisées (puisqu'elles sont dans un matériau magnétique). L'idéal serait ainsi de trouver un modèle qui serait suffisamment puissant pour

être efficace dans l'attraction des lames, mais pas assez pour que les actionneurs ne puissent pas faire leur travail.

b. Composants électroniques principaux

Les électroaimants des lames

Dans la réalisation de ce projet, nous avons imaginé plusieurs moyens différents pour « activer » les lames du Magimba. Jusqu'à tenter d'utiliser des électroaimants véritables (C'est-à-dire à la différence de solénoïdes, sans partie mouvante).

Cependant, nous confrontant à des problématiques physiques et un rendement insuffisant, nous avons dû retourner sur une base moins expérimentale et plus mécanique : Les solénoïdes.

Ce choix nous permet alors de régler les difficultés rencontrées auparavant, et nous permet d'avancer et de commencer leur intégration sur le circuit.

Cela dit, la « puissance » que ces **solénoïdes** (12V – 2A), bien que moindre que les électroaimants véritables, demandent pour fonctionner se révèle être importante pour ce que la carte utilisée peut fournir. Surtout lorsque nous prévoyons d'en avoir autant que nous avons de notes. Cette situation augmente nos contraintes pour la conception de la batterie du Magimba.



Figure 7 : Apparence des électroaimants dans le projet

Les diodes de roue libre

Cela dit, nous ne pouvons pas vraiment nous permettre d'utiliser un électroaimant seulement comme cela. En effet, le problème que nous pouvons rencontrer dans cette situation est une surtension, due au fait que nous observons une charge inductive (solénoïde) susceptible d'avoir un des interruptions rapides du courant dans son circuit. Pour éviter

d'abîmer le transistor lorsqu'il coupera le circuit, il est essentiel de rajouter une diode de roue libre. La diode que nous utiliserons sera alors **la diode 1N4007**.

Les transistors

Dans notre conception du projet, puisque la carte utilisée ne peut pas alimenter autant de solénoïdes. Nous avons décidé de les relier « directement » à la batterie. Cela dit, pour pouvoir tout de même avoir un certain encadrement sur leurs actions, l'utilisation de transistors s'est révélée être une évidence, tant que le modèle qui correspondra aux attentes existe.

C'est en grande partie pourquoi **le transistor E44H11** (NPN) est un choix optimal pour notre Magimba puisqu'il répond parfaitement aux besoins en puissance et contrôle électronique des électroaimants que nous avons choisis.

- **Haute capacité de courant** : Il peut supporter jusqu'à 10A, suffisant pour activer nos électroaimants (12V - 2A).
- **Facilité d'intégration avec ESP32** : En tant que transistor NPN, il s'active en appliquant une tension positive sur la base, ce qui simplifie son pilotage par un microcontrôleur.
- **Commutation efficace** : Son faible V_{CEsat} ($< 0.5V$) réduit la perte d'énergie et améliore l'efficacité du circuit.
- **Protection et durabilité** : Combiné avec une diode de roue libre (1N4007), il protège le circuit contre les surtensions générées par l'électroaimant.

En résumé, le **E44H11** est un choix robuste, simple à utiliser et bien adapté aux exigences de notre système d'électroaimants.

2. L'alimentation

2.1 Présentation

Depuis la présentation des différents **sous-systèmes** du **Magimba**, un besoin évident d'énergie s'est fait ressentir. Ce besoin d'énergie peut être réglé par l'utilisation de prises secteur, mais dans la partie des contraintes, il est voulu que l'orchestron soit « autonome » en énergie. C'est pourquoi, l'intégration d'une batterie est prévue pour éviter de dépendre uniquement des prises secteurs.

Ainsi ce sous-système d'alimentation est conçu pour fournir une tension stable de 5V 10A aux 12 électro-aimants (5V 1,1A chacun) et au microcontrôleur ESP32. Deux sources d'alimentation sont disponibles :

- **Mode secteur** : une alimentation à découpage 5V 10A permet d'alimenter directement le système lorsqu'une prise secteur est disponible.
- **Mode batterie** : une batterie Li-Ion 3S 12Ah associée à un convertisseur DC-DC 12V → 5V 10A prend le relais en mode mobile.

Le passage d'une source à l'autre est assuré manuellement via un switch qui sélectionne l'alimentation active.

2.2 Conception technique

a. Attendus techniques

Suite à la liste des différents éléments que nous retrouverons dans notre projet, il est primordial de lister l'ensemble des besoins que nous aurons dans la conception de la batterie. Cette liste est la suivante :

- **Double mode d'alimentation** : secteur (5V 10A) ou batterie (via DC-DC). Afin de rester polyvalent dans la gestion de l'énergie du projet.
- **Tension de sortie stable** : $5V \pm 5\%$.
- **Capacité** à fournir un **courant** de 10A max. Pour l'alimentation d'au moins 8 actionneurs de manière simultanée.
- **Autonomie** suffisante en mode batterie (environ 2,5 heures à 50W). Point important pour pouvoir faciliter la partie d'utilisation de l'orchestron sans prise secteur à proximité.
- **Sécurité** : protections contre surcharge, court-circuit et inversion de polarité.
- Commutation **simple** et **fiable** via un switch manuel pour le changement de mode d'alimentation (secteur ou batterie).

b. Données techniques

Voici un ensemble de données techniques du module d'alimentation du projet, réparti selon les parties du sous-système :

Alimentation Secteur (5V 10A) :

- **Entrée** : 220V AC.
- **Sortie** : 5V DC, 10A.
- **Puissance** : 50W.
- **Efficacité** : >85%.
- **Protections** : Court-circuit, surcharge.

Batterie Li-Ion 3S 12Ah (pour le mode d'alimentation mobile) :

- **Tension nominale** : 11,1V (max 12,6V, min 9V).
- **Capacité** : 12Ah.
- **Énergie stockée** : 133Wh.
- **Courant de décharge max** : $\geq 10A$.
- **Autonomie estimée** : environ 2,5 heures à 50W.
- **BMS (Battery Management System) intégré** : Pour la protection contre des surcharges, des décharges profondes et des courts-circuits.

Convertisseur DC-DC (Batterie → 5V) :

- **Entrée** : 9-14V.
- **Sortie** : 5V 10A.
- **Efficacité** : >90%.
- **Protections** : Court-circuit, surtension.

Switch Manuel :

- **Type** : Interrupteur bipolaire ON-ON (DPDT).
- **Fonctionnement** : Sélectionne entre l'alimentation secteur et la batterie.

2.3 Composants

Le sous-système de l'alimentation est composé :

- D'une **alimentation à découpage** 5V 10A, pour réduire la tension de sortie de la prise secteur et alimenter l'orchestrier dans une plage convenable (5V 10A).
- D'une **batterie** Li-Ion (Lithium) 3S 12Ah avec BMS (un système de contrôle) , qui offrira une bonne autonomie et une densité énergétique élevée au sous-système.
- D'un **convertisseur DC-DC** 12V → 5V 10A, qui assurera une tension stable de 5V avec une bonne efficacité.
- D'un **switch manuel** (DPDT ON-ON), présent pour basculer sans risque de coupure brutale entre l'alimentation « secteur » et l'alimentation « autonome ».

2.4 Schéma électrique simplifié sur le projet

Voici le schéma électrique simplifié du sous-système de la batterie :

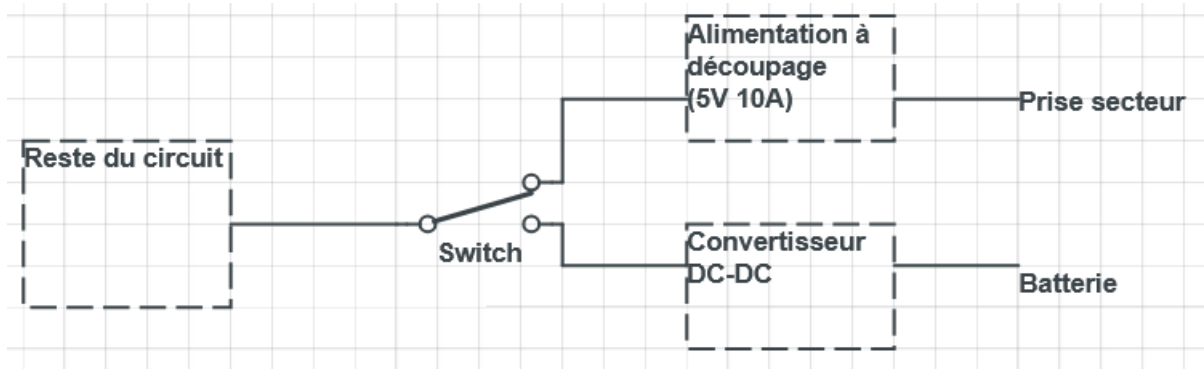


Figure 8: Schéma électrique simplifié du sous-système d'alimentation (réalisé avec CircuitLab)

Position 1 du switch : Alimentation secteur.

Position 2 du switch : Batterie active.

3. L'affichage des modes de fonctionnement

3.1 Présentation de l'écran

Nous avons choisi pour afficher les différents modes de fonctionnement en temps réel sans passer par la console de l'environnement de développement, d'utiliser un écran OLED 128x64.

L'ESP32 contrôlera alors l'affichage des modes de fonctionnements sur l'écran OLED en utilisant le protocole I2C.

3.2 Schéma électronique simple

L'intégration de l'écran dans le circuit se fait de manière simple et rapide. Celui-ci comporte 4 broches qu'on relie aux broches de même appellation :

- **+5V** : Broche d'alimentation de l'écran, sur la broche du même nom de la carte.
- **Gnd** : La masse de l'écran à la masse de l'ESP32.

Puisque nous utilisons le protocole I2C pour l'écran, et que celui-ci se base sur l'emploi de deux fils de communication. Nous avons :

- Une broche **SCL** : (Ligne d'horloge série) Pour permettre au micro-contrôleur de synchroniser les données entrantes ou sortantes de l'écran.

- Une broche **SDA** : (Ligne de donnée série) Pour l'envoi des données, qui serviront ici à afficher des éléments pour l'écran.

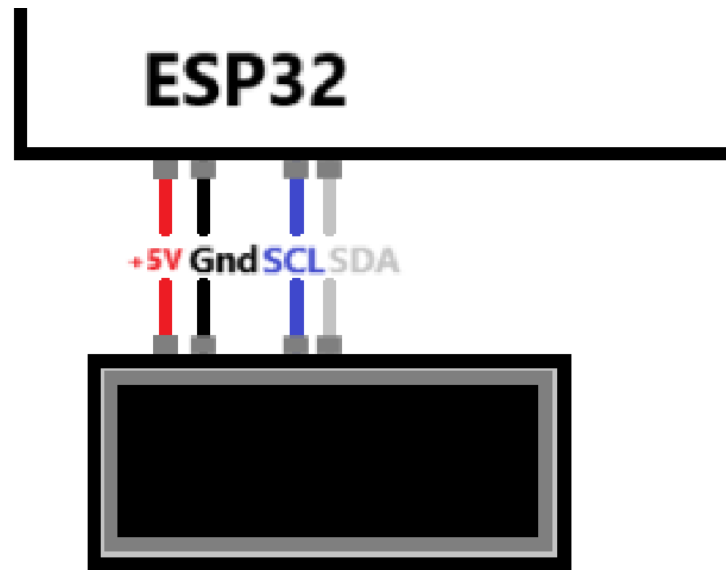


Figure 9 : Schéma électronique simple de l'écran intégré à l'ESP32

4. Langages utilisés

Voici les différents langages de programmation utilisés dans le projet MAGIMBA :

- Le langage **Arduino/C** : Pour programmer la carte **ESP32** et les différentes fonctions principales correspondant aux modes de fonctionnement de l'orchestrier.
- Le langage **Javascript** : Pour la communication avec le site web à l'aide de requête http.
- Le langage **CSS** : Pour la conception du site web.
- Le langage **HTML** : Pour la conception du site web.

La raison pour laquelle nous avons fait le choix de coder la carte choisie (**ESP32**) en C (**Arduino/C**) plutôt qu'en python (**micropython** dans le cadre des micro-contrôleurs) comme il peut parfois être d'usage peut s'expliquer de la manière suivante :

- **Gain de temps et économie de batterie** : Le langage **C** est plus rapide que le **micropython**. Et cela permet donc en quelque sorte de moins utiliser d'énergie pour effectuer les mêmes tâches (puisqu'elles prendront moins de temps).
- **Utilisation de mémoire** : Le langage **C** utilise moins de mémoire que **mpython** et **cpython**. Un point important pour le mode semi-automatique du Magimba tel que nous avons pu le voir dans la partie du choix de la carte.

Afin de faire l'interface web dont nous avons besoin pour pouvoir réaliser la fonctionnalité attendue de l'orchestron, le mode à distance. Nous devons :


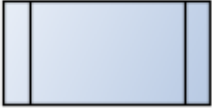

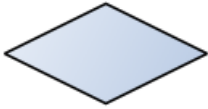

- La **créer** dans un premier temps, ce pourquoi nous utiliserons le langage **HTML** (structure du site web) et le langage **CSS** (présentation du site web).
- **Interagir** rapidement et facilement avec le contenu de cette interface. C'est pourquoi nous avons décidé de nous occuper de cette partie avec le langage **Javascript**. Nous l'utiliserons également pour envoyer des requêtes HTTP au micro-contrôleur via une liaison wifi. Ce qui nous permettra de compléter la demande d'un mode à distance du cahier des charges

Référence : <https://www.youtube.com/watch?v=u9UfKTOcYNs&t=134s>

5. La programmation

Dans cette partie, nous allons présenter les **algorigrammes** permettant d'expliquer les principes du code lié au projet du **Magimba**.

Durant la lecture de cette partie, vous pourrez voir après l'explication des programmes, un algorigramme lié à celui présenté. Voici les blocs utilisés et leur fonction :

Type de blocs utilisés	Fonction du bloc
	Début ou fin d'une fonction
	Initialisation (variables, constantes, ...)
	Action(s) faites dans le programme (opérations, envois de données, ...)
	Choix logiques (boucles, vérifications, ...)
	Affichage sur console ou écran physique

5.1 Fonction principale

Cet algorithme représente l'idée du code tel que nous l'avons imaginé pour le prototype final du Magimba. Cependant nous n'avons pas encore considéré l'ajout de la fonction « Chef d'orchestre » liée au projet tutoré ainsi que la fonction « Distance » pour le PA.

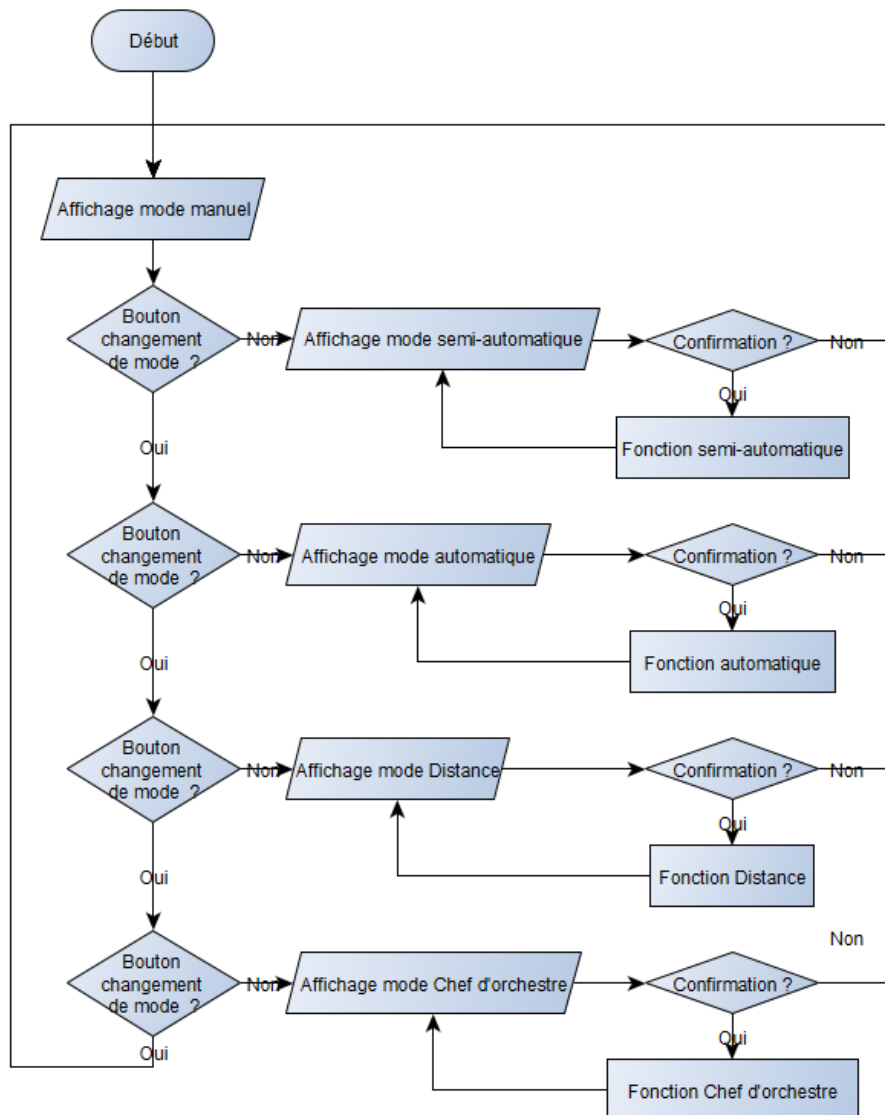


Figure 10 : Algorithme de la fonction principale du programme

5.2 Fonction Automatique

La fonction Automatique, plus simple que la fonction précédente, repose sur l'utilisation de 2 boutons pour le joueur :

- Le bouton **PLAY** : Qui permettra d'écouter un morceau pré-enregistré. Ce mode de fonctionnement est essentiellement présent pour des batteries de tests sur le contrôle des lames.
- Le bouton de sortie : Qui fait sortir de la fonction et permet de revenir au mode manuel.

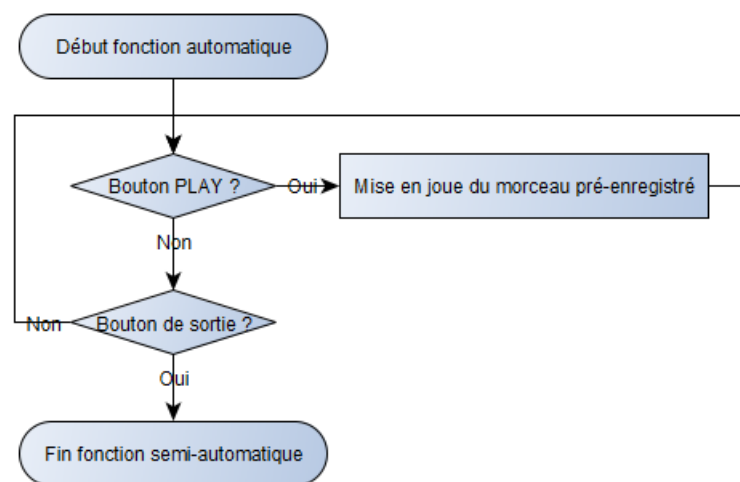


Figure 11 : Algorithme de la fonction Automatique

5.3 Fonction Semi-automatique

La fonction Semi-automatique fonctionne sur l'utilisation de 3 boutons pour le joueur :

- Le bouton **REC** : Qui débutera l'enregistrement
- Le bouton **PLAY** : Qui permettra d'écouter ce qui a été enregistré. Si rien n'est disponible, un message l'indiquant s'affichera sur l'écran.
- Le bouton de **sortie** : Qui fait sortir de la fonction et permet de revenir au mode manuel.

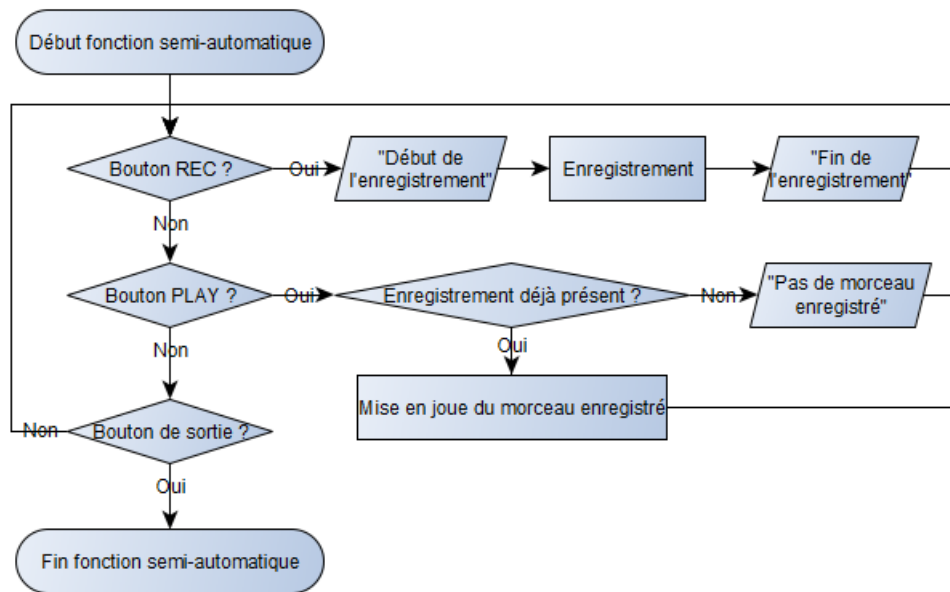


Figure 12 : Algorithme de la fonction Semi-automatique

5.4 Logique de l'enregistrement

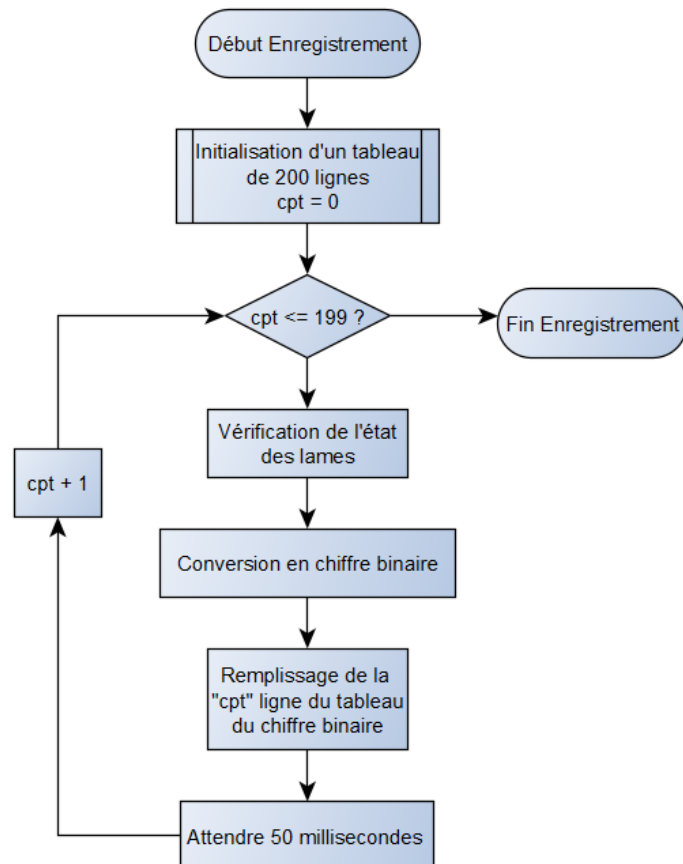


Figure 13 : Algorithme du fonctionnement de l'enregistrement

Ainsi nous dispensons à la fin de la fonction d'un tableau qui répertorie toutes les notes jouées toutes les 0.05 secondes pendant 10 secondes d'enregistrement, ce qui nous donne un ensemble de 200 lignes, illustrée comme le tableau suivant :

Numéro de ligne	Contenu de la ligne
1	0b000100001000

Où le chiffre binaire : 0b000100001000 correspond aux notes jouées et enregistrées au moment de l'enregistrement de cette ligne du tableau

Et peut être décomposé de la manière suivante :

Notes	Do	Ré	Mi	Fa	So	...
Correspondance	0	0	0	1	0	...

Calcul de la mémoire :

Puisque nous prévoyons d'enregistrer 10 secondes avec un temps d'échantillonnage de 0.05 seconde. Nous pouvons donc en déduire par la formule suivante :

$$\frac{t_{total}}{t_{\text{échantillonnage}}} = \frac{10}{0.05} = 200 \text{ bits si nous ne voulons qu'enregistrer l'état d'une note.}$$

Cependant, nous voulons avoir l'état de 12 notes et l'ESP32 ne « contrôle » les données que par paquet de 8 bits. Nous devons donc prendre en compte, non pas 12 bits mais 2 octets de données pour l'enregistrement à chaque ligne du tableau. Notre mémoire nécessaire sur l'ESP32 devra donc être au minimum de :

$$200 * 16 = 3.2 \text{ Kbits}$$

VII. Tables des illustrations

Figure 1 : Croquis simplifié du système global pour le PB.....	6
Figure 2 : fonctions principales et les fonctions de contraintes du projet	7
Figure 3 : : Croquis visuel du dispositif, vue de côté (Réalisé sur LucidChart)	14
Figure 4 : Croquis visuel du dispositif, vue de derrière (Réalisé sur LucidChart)	14
Figure 5 : Croquis de dimensions du dispositif, vue de côté	15
Figure 6 : Schéma électrique d'un actionneur type (réalisé avec CircuitLab)	16
Figure 7 : Apparence des électroaimants dans le projet.....	18
Figure 8: Schéma électrique simplifié du sous-système d'alimentation (réalisé avec CircuitLab).....	22
Figure 9 : Schéma électronique simple de l'écran intégré à l'ESP32.....	23
Figure 10 : Algorithme de la fonction principale du programme	26
Figure 11 : Algorithme de la fonction Automatique.....	27
Figure 12 : Algorithme de la fonction Semi-automatique	28
Figure 13 : Algorithme du fonctionnement de l'enregistrement	28