

Dossier de Conception

Table de Matières

I - Avant propos - Mise en contexte	2
II - Rappel des besoins et des contraintes	2
A - Explication du fonctionnement	2
B - Gestion de l'alimentation	2
B - Analyse Fonctionnelle Technique	3
C - Contraintes	3
III - Justification des choix et Explication	4
A - Programmation	4
1 - L'IDE	4
figure 1: avantages de PlatformIO par rapport à Arduino IDE	4
2 - La carte de développement	4
figure 2: détails de la datasheet de l'Arduino Uno R4 Minima	5
figure 2.5: Conceptualisation des broches reliées au microcontrôleur	5
B - Interface graphique	6
1 - Les commandes	6
2 - L'affichage	6
figure 3: Écran GME128128	7
figure 3.5: Conceptualisation du câblage de l'écran oled GME128128	7
C - Fonctionnalités du produit	7
1 - Menu	7
figure 4: Organigramme du Menu	8
2 - Mode Automatique/Manuel	8
figure 5: Organigramme des Modes	9
3 - Fréquencemètre	9
figure 6: Organigramme du Fréquencemètre	9
4 - Batterie	9
figure 7: Organigramme de la Batterie	10
5 - Rappel des fréquences des Notes Arrêt du programme	10
figure 8: Organigramme du Rappel des notes	11
D - Partie Mécanique	12
2. Croquis du projet	12
3. Étude de conception	13

3.1 Contraintes techniques	13
3.2 composant et méthodes	13
4. Conception CAO et intégration	13
4.1 Étapes de modélisation	13
4.2 Intégration mécanique	13
4.3 système manuel de tension	14
5. Fabrication et assemblage	14
5.1 Fabrication des pièces	14
5.2 Montage final	14
6. Perspectives d'évolution	14
6.1 Automatisation de la tension	14
6.2 Optimisation du système électronique	14
E - Alimentation	15
1. Objectifs	15
2. Analyse des besoins	15
3. Architecture globale	16
4. Composants utilisés	18
5. Justification des choix techniques	18
5.1 Choix du Transformateur 24V	18
5.2 Choix des convertisseurs Buck	19
5.3 Choix du Basculement Secteur/Batterie	20
5.4 Choix du Switch ON/OFF	21
5.5 Choix du redresseur et filtre	21
6. Protection du Système	22
IV - Jalon des versions du document	22

I - Avant propos - Mise en contexte

Ce document va retracer toutes les étapes de conception du produit, ainsi que les détails de de résolution des besoins auxquels la guitare électro-acoustique doit répondre. Des figurés, des calculs et des schémas seront disponibles pour schématiser les fonctions du système.

Nous commencerons avec des rappels des besoins pour recentrer le cadre et à quelles questions répond la guitare électro-acoustique. Ensuite, nous continuerons sur les détails de conception de chaque sous système de la guitare: affichage, mécanique, programmation. Nous terminerons ensuite par une synthèse et

II - Rappel des besoins et des contraintes

A - Explication du fonctionnement

Notre guitare électro-acoustique se distingue par deux modes de fonctionnement :

1. Mode manuel

L'utilisateur joue de manière classique, sans intervention électronique. Seule l'amplification est activée.

2. Mode automatique

Les moteurs et actionneurs gèrent totalement l'accordage et la mise en vibration des cordes selon une partition ou un signal pré-enregistré.

B - Gestion de l'alimentation

Le système fonctionne avec deux sources d'alimentation :

- Mode Secteur : Le transformateur 24V AC convertit le courant en tensions adaptées aux composants. Il recharge également la batterie.
- Mode Batterie : Lorsque le secteur est coupé, la batterie alimente le système de façon autonome.

Un switch ON/OFF permet de désactiver complètement l'alimentation, protégeant ainsi les composants et économisant l'énergie.

L'alimentation fournit différentes tensions :

- 12V pour les moteurs pas à pas.
- 6V pour l'Arduino UNO R4 Minima.
- 5V pour les electroaimants.

Le passage entre secteur et batterie est géré automatiquement via des diodes Schottky.

B - Analyse Fonctionnelle Technique

On peut résumer toutes les fonctionnalités dans un tableau d'analyse fonctionnelle technique:

Fonction	Entrées	Sorties	Contraintes	Critères de performances
Alimentation du système	Secteur (24V AC) ou batterie	12V, 6V, 5V	Sécurité, rendement > 80%	Autonomie \geq 2h, pertes minimales
Contrôler l'accordage	Signal utilisateur ou capteur	Commandes moteur	Rapidité, précision	Temps d'accord < 3s par corde
Faire vibrer les cordes	Signal de commande	Activation des électroaimants	Réactivité, puissance adaptée	Force suffisante pour activer les cordes
Amplifier le son	Signal du microphone	Signal amplifié	Distorsion minimale, consommation réduite	Puissance audio \geq 5W
Gérer les modes d'utilisation	Sélection mode utilisateur	Activation du mode correspondant	Ergonomie, fiabilité	Temps de réponse < 1s

C - Contraintes

- conso du système (moteurs);
- tension des cordes;
- résistance des matériaux (bois, plastique 3D);
- limite des broches (arduino r4 minima)
- alimentation stable et efficace
- compatibilité avec les composants sélectionnés
- sécurité électrique et protection du système
- budget limité
- taille et encombrement limité
- fiabilité et durabilité

III - Justification des choix et Explication

A - Programmation

1 - L'IDE

Visual Studio Code (VS Code) est un environnement très simple à utiliser et dans lequel on peut ajouter multitude de modules et de librairies pour le projet qu'on veut réaliser.

Dans notre cas, cet environnement fut utilisé à de nombreuses reprises dans des projets d'électronique embarquée et il est tout à fait intéressant de le réutiliser sachant que nous sommes déjà habitués à manipuler VS Code.

D'un point de vue plus sur la programmation, puisqu'on utilise une Arduino R4 Minima, il nous faut ainsi coder en Arduino, c'est-à-dire en C/C++.

Dans ce cas, un module nommé PlatformIO permet de créer un environnement de développement comme sur l'IDE d'Arduino tout en restant sur la plateforme de VS Code.

Ce module a des avantages que la plateforme de développement Arduino IDE ne possède pas, comme l'organisation du dossier où les fichiers résident, IntelliSense, la possibilité de faire des breakpoints et traquer les modifications des variables durant le programme, ou encore une vitesse plus grande pour le "build" d'un programme:

PlatformIO for Arduino?								
Feature	VSCode + PlatformIO	Arduino IDE	Library Management			Build Speed		
Code Editing Features	✓ Advanced (IntelliSense, refactoring, linting)	✗ Basic	✓ Automatic, centralized	✗ Manual		✓ Fast with incremental builds	✗ Slow	
Project Structure	✓ Multi-file, well-organized	✗ Sketch-based (flat structure)	✓ Full debugging (breakpoints, real-time inspection)	✗ Serial print only		✓ Easy setup, extensions	✓ Very simple	
						Cost	✓ Free	✓ Free

figure 1: avantages de PlatformIO par rapport à Arduino IDE

2 - La carte de développement

Le projet nécessite la création de 6 signaux PWM qui vont commander 6 moteurs différents sauf si les contraintes nous empêchent de faire ainsi. Avec l'Arduino R4 Minima, nous avons accès à 6 broches PWM qui vont être envoyées sur les moteurs. La carte peut être alimentée en 3.3 V et en 5 V.

En plus de la mémoire bien plus que suffisante de 32kB, nous possédons 6 broches pouvant envoyer un signal PWM essentiel pour les signaux de commandes pour les moteurs:

■ Memory

- 256 kB Flash Memory
 - 32 kB SRAM
 - 8 kB Data Memory (EEPROM)
-
- 14x digital pins (GPIO), D0-D13
 - 6x analog input pins (ADC), A0-A5
 - 6x PWM pins: D3,D5,D6,D9,D10,D11

figure 2: détails de la datasheet de l'Arduino Uno R4 Minima

Dans notre cas, nous avons besoin d'une quantité conséquente de broches:

- quatre broches d'entrées de commande (valider une action, retourner en arrière ou annuler une action, monter dans le menu ou descendre dans le menu),
- le nombre de broches de sorties reliées au nombre de moteurs disponible (6 signaux pwm à envoyer sur 6 actionneurs disponibles en théorie, cependant le problème de la quantité des moteurs soulève une possibilité de surconsommation; une solution avec un seul moteur est en phase d'étude),
- 2 broches d'entrées pour les informations de batterie et de fréquence du signal audio (signal audio directement originaire des vibrations de la corde et informations de la batterie sous forme d'un signal DC ou un nombre binaire de 8 bits; partie en attente de l'alimentation),

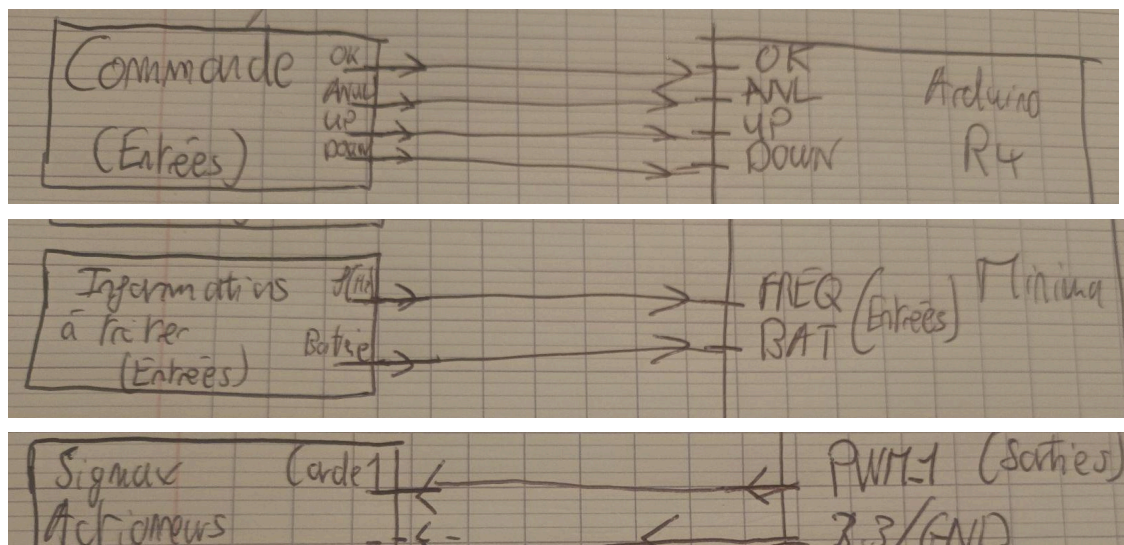


figure 2.5: Conceptualisation des broches reliées au microcontrôleur

B - Interface graphique

1 - Les commandes

Dans une mesure de simplicité, les commandes seront gérées par un petit réseau de boutons poussoir reliés avec des LEDs et une bascule D pour deux de ses boutons (le bouton de validation et le bouton d'annulation d'action).

La bascule D ici, en reliant les broches D et Q_bar, permettra de retenir les états des deux actions pour faciliter le programme pour l'affichage de l'information sur l'écran oled.

Les boutons poussoirs sont en cours de révision pour le prototype B, puisque ces mêmes éléments ont un problème de rebond, empêchant la confirmation d'une demande d'action et, logiquement, une perte de temps dans le cas où l'utilisateur doit appuyer plusieurs fois sur le bouton pour une seule action.

2 - L'affichage

Au début de la conception et des premières tentatives, un écran oled 128 pixels sur 64 pixels fut sélectionné: le nombre d'informations disponible à ce moment de la conceptualisation et la réalisation du code. Cependant, au fur et à mesure que les besoins se sont précisés, il est devenu évident qu'il nous faudrait un écran 128x128 pixels pour l'affichage.

Ainsi, nous avons un écran 128x128 pixels en communication I2C qui a une couleur: le blanc. Ce composant, se reposant sur l'I2C, nécessite deux broches de communication: la broche SDA pour l'envoi des données et la broche SCL pour l'horloge imposée par le microcontrôleur, ici la R4 Minima. Les deux autres broches alimentation sont aux 3.3 V et à la masse, le GND.



figure 3: Écran GME128128

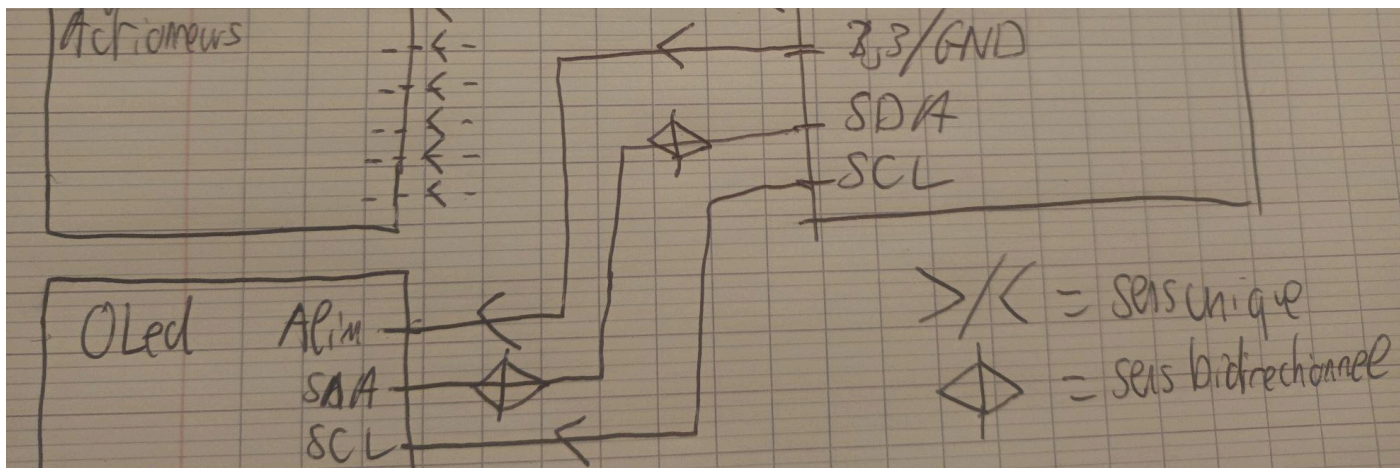


figure 3.5: Conceptualisation du câblage de l'écran oled GME128128

C - Fonctionnalités du produit

1 - Menu

Lors du lancement du programme, l'écran oled va afficher les images de démarrage pour celle de l'écran et aussi pour l'introduction du produit. Ensuite, un menu sera disponible à l'utilisateur.

Quatre boutons permettent de naviguer sur le menu et choisir quelle fonctionnalité on veut utiliser. Dans le cas où on appuie sur le bouton d'annulation, on lance la fonction d'arrêt du programme entier. Le programme retient l'état des boutons de sélection et d'annulation (boutons OK et ANNUL) et les deux boutons de navigations sont des impulsions.

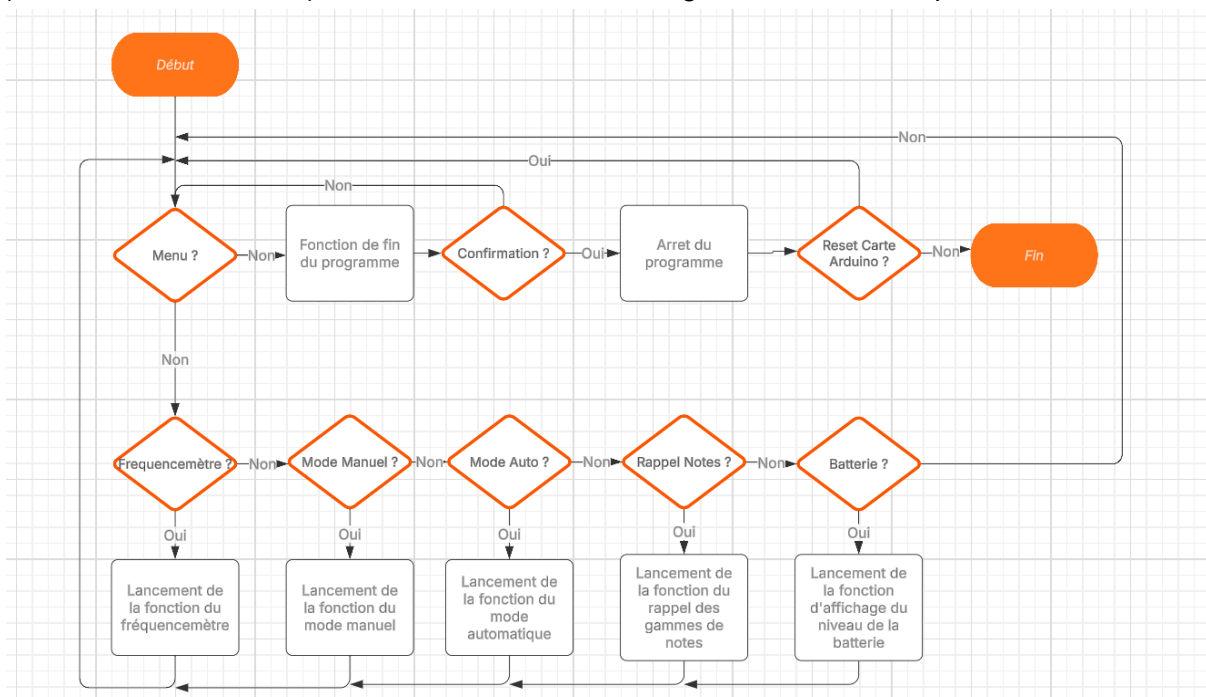


figure 4: Organigramme du Menu

2 - Mode Automatique/Manuel

Les deux modes principaux de la guitare électro-acoustique qui vont déterminer son utilisation sont les modes manuel et automatique.

Le mode manuel consiste à jouer de la guitare de manière acoustique: c'est-à-dire avec les doigts, le pincement des cordes et leur mouvement au niveau de la caisse de résonance. De plus, puisqu'on utilise que l'écran oled, pour le prototype B, il est envisagé que le programme la consommation du système électronique entier, relié à la batterie et aux broches d'alimentation des composants les plus énergivores.

Le mode automatique va insister à envoyer des commandes pré-enregistrées sélectionnables, au préalable, sur des actionneurs qui vont eux même pincer la corde et la frotter au niveau de la caisse de résonance. Des signaux PWM seront envoyés sur les actionneurs de manière précise, c'est-à-dire avec des temps de delay particulier qui vont permettre de simuler de manière efficace un musicien s'entraînant ou utilisant une guitare.

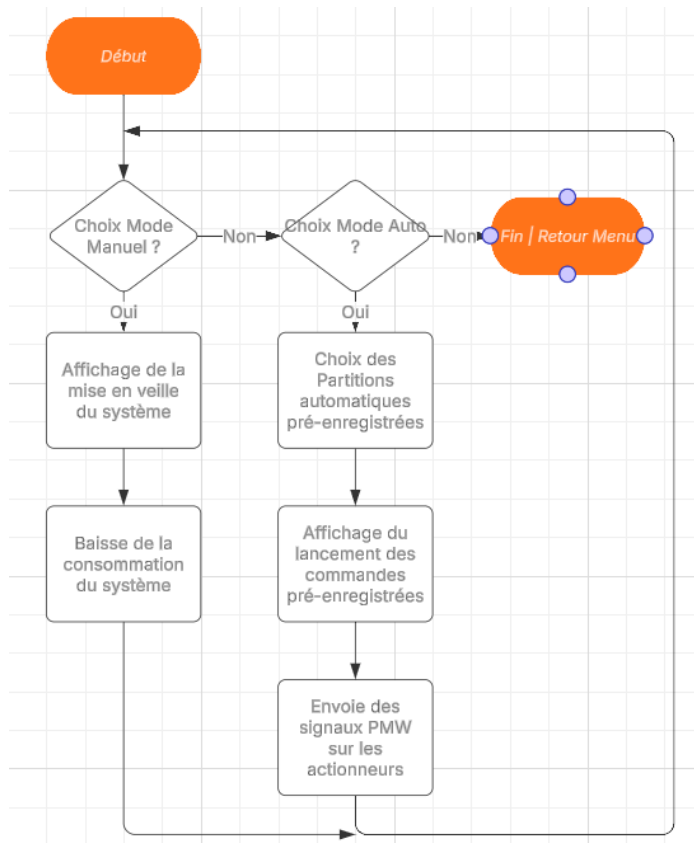


figure 5: Organigramme des Modes

3 - Fréquencemètre

Le fréquencemètre consiste à mesurer les fréquences émises par la guitare, plus précisément les fréquences des cordes pincées. Le but ici est de déterminer si la corde est accordée ou si un réajustement de la tension physique de la corde est nécessaire.

Le système va récupérer la fréquence via un microphone qui va envoyer un signal sur un comparateur qui va déterminer le dépassement d'un seuil. Puis, les temps des états haut et bas seront calculés sur l'Arduino R4 Minima et, en faisant une moyenne, on obtient la période moyenne et, indirectement, la fréquence moyenne.

Au niveau de l'affichage, il y aura un sprite, une image en 64 par 64 pixels et en code hexa, qui représente une note de musique, puis en dessous de l'image on pourra voir l'affichage de la fréquence en Hertz (Hz). Si la fréquence calculée est une valeur particulière, c'est-à-dire que si elle correspond à la valeur d'une des notes des cordes de guitare, stockées dans des tableaux, alors le programme devra montrer le nom de la note. Sinon, dans le cas contraire, un texte hors cadre sera affiché.

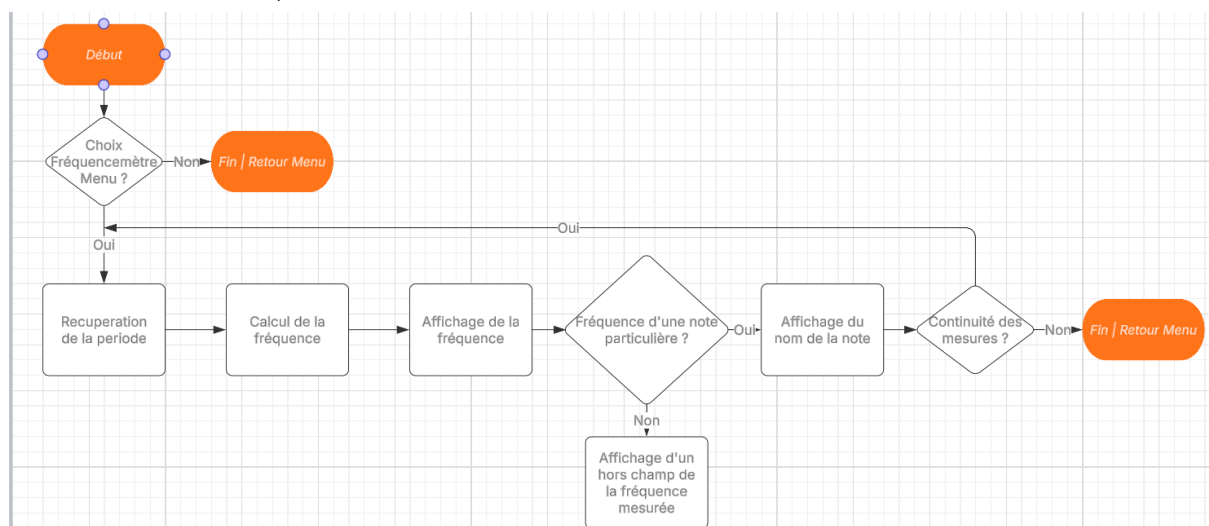


figure 6: Organigramme du Fréquencemètre

4 - Batterie

Le système doit répondre à un besoin essentiel: il doit être capable de fonctionner de manière embarqué: c'est-à-dire que le produit doit être capable d'avoir une alimentation intégrée dedans. Ainsi, on doit intégrer une batterie dans le système et les différents niveaux de tension de la batterie atteste de son niveau de charge.

Cependant, il faut pouvoir traquer l'évolution de la batterie si la guitare n'est pas utilisée en mode secteur, pour pouvoir prévoir une éventuelle recharge de la batterie qui serait nécessaire. Ainsi, il faudrait mesurer le niveau de la batterie en temps réel et pouvoir montrer le pourcentage de charge.

Cette partie est encore en conception mais la piste envisagée est que directement en sortie de la batterie, en parallèle de la piste qui va alimenter tout le circuit, il y aura un module de conversion qui va soit envoyer des trames binaires avec une suite de porte logique, pour avoir un octet représentant la tension et une programme convertissant ce nombre binaire en nombre numérique avec le taux de charge de la batterie. Il est aussi probable que toute la partie calcul soit faite en hardware pour maximiser la vitesse de la carte.

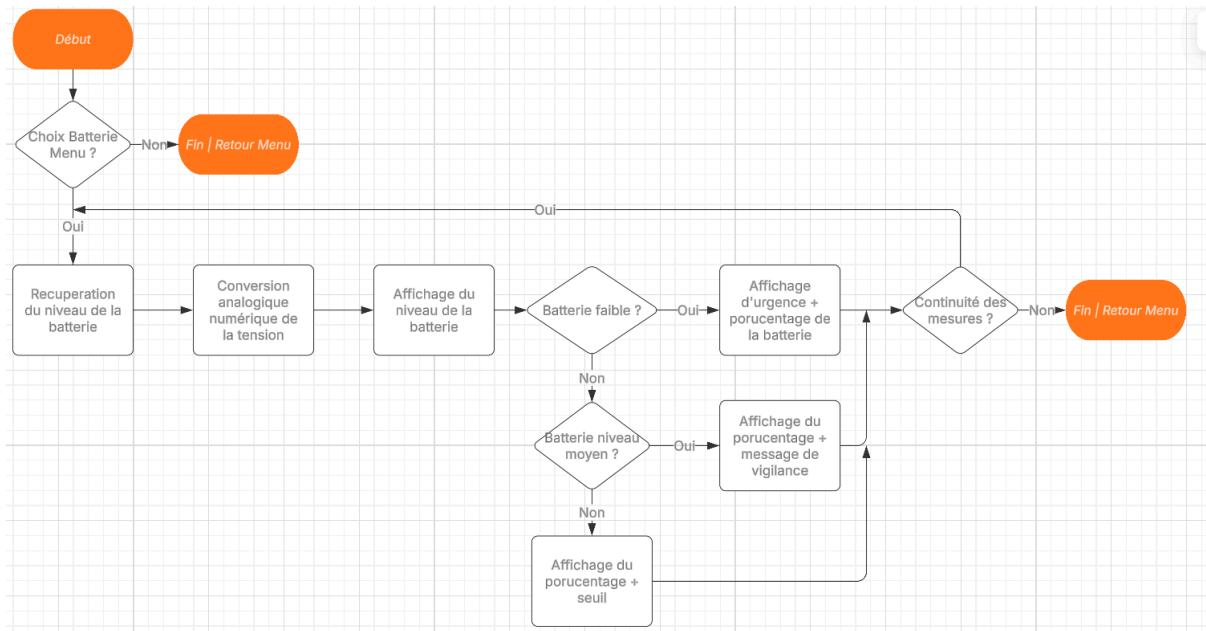


figure 7: Organigramme de la Batterie

5 - Rappel des fréquences des Notes | Arrêt du programme

Les deux dernières parties s'occupent d'un simple rappel des notes des cordes et aussi de l'arrêt du programme, dans le cas où l'utilisateur veut arrêter d'utiliser la guitare, ou veut seulement la jouer en acoustique sans utilisation d'énergie quelconque.

Le processus doit gérer plusieurs conditions et éviter les états incohérents. L'une des contraintes majeures est de s'assurer qu'un arrêt du programme ne se fasse pas accidentellement sans confirmation. De plus, la gestion des boutons (OK, ANNUL) doit être robuste pour éviter des erreurs logiques, notamment lorsqu'un utilisateur souhaite annuler une action déjà en cours. Enfin, un mécanisme de reset est prévu pour permettre à l'utilisateur de relancer le programme après un arrêt.

Le logigramme propose une solution en guidant l'utilisateur à travers des choix successifs. Si l'utilisateur demande un rappel des notes, le programme affiche les gammes et les fréquences. En cas de demande d'arrêt, une confirmation est requise. Si l'utilisateur active le bouton "ANNUL", le programme empêche l'arrêt. En revanche, si "OK" est activé, le programme affiche une option pour réinitialiser la carte Arduino, permettant ainsi une relance propre du système. Cette structure assure une interaction fluide et sécurisée.

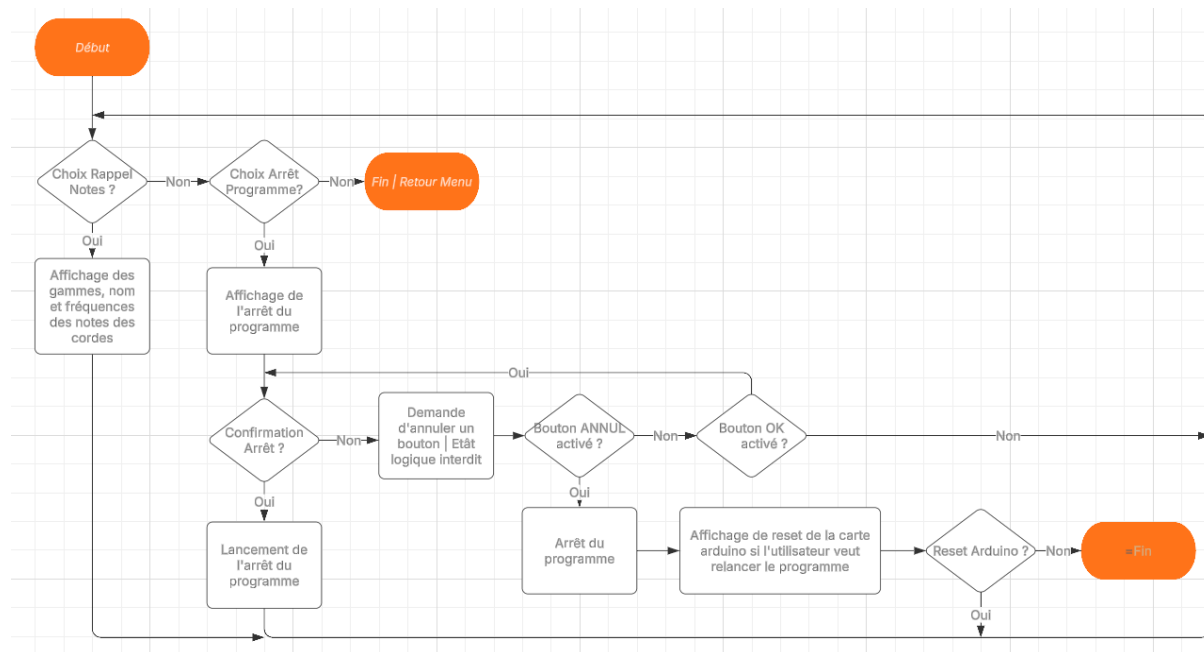


figure 8: Organigramme du Rappel des notes

D - Partie Mécanique

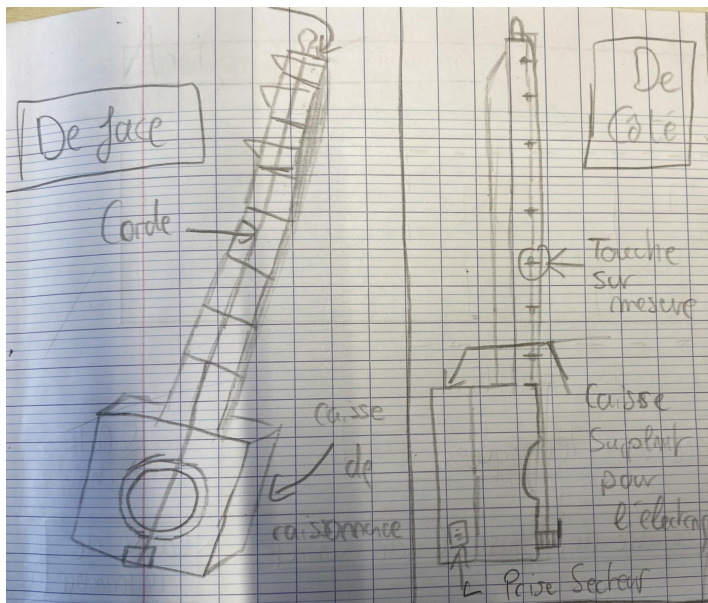
Cette partie a pour objectif de concevoir une monocorde électro-acoustique, basée sur un corps en bois fourni par le professeur. La conception inclut :

- La fabrication de pièces mécaniques en impression 3D pour le guidage et la tension de la corde.
- L'ajout d'un compartiment électronique à la caisse de résonance, afin d'intégrer un moteur et un système de contrôle pour l'automatisation future de la tension de la corde.

Le monocorde doit produire un son stable (Mi) tout en offrant des possibilités d'évolution vers une automatisation complète.

2. Croquis du projet

Voici le croquis initial, montrant les éléments principaux de la conception :



Description du croquis :

Vue de face :

- Une corde tendue au-dessus de la caisse de résonance.
- Une touche sur mesure pour guider la corde.

Vue de côté :

- Une caisse de résonance déjà existante.
- Un compartiment supplémentaire dédié au système électronique et à l'alimentation.
- Un système mécanique pour ajuster la tension de la corde.

3. Étude de conception

3.1 Contraintes techniques

- Le corps en bois ne peut pas être modifié, mais un compartiment interne ou externe sera ajouté pour accueillir les composants électroniques.
- Les pièces mécaniques nécessaires (guide de corde, système de maintien, support pour l'électronique) seront fabriquées en impression 3D.
- La tension de la corde sera ajustable manuellement dans un premier temps, avec une prévision pour l'intégration d'un moteur électrique.

3.2 composant et méthodes

Composant	Méthode de fabrication
Guide de corde	Impression 3D
Système de maintien	Impression 3D
Compartiment électronique	Impression 3D

4. Conception CAO et intégration

4.1 Étapes de modélisation

1. Analyse du corps fourni pour déterminer les dimensions disponibles pour le compartiment électronique.
2. Modélisation des pièces imprimées (guide de corde, système de maintien, compartiment électronique).
3. Préparation des supports pour le moteur et l'alimentation.

4.2 Intégration mécanique

- Installation du système manuel de tension (roue ou vis de serrage).
- Prévision d'un logement pour intégrer le moteur sans modification majeure de la structure.
- Intégration du compartiment électronique pour protéger les circuits et faciliter l'accès à l'alimentation.

4.3 système manuel de tension

Pour étirer la corde manuellement et ajuster sa tension, un système simple mais efficace a été conçu. La corde est fixée à la base de la guitare à l'aide d'une vis de maintien. À l'autre extrémité, un mécanisme de serrage a été installé. Ce mécanisme utilise une roue qui permet d'enrouler la corde autour d'elle. En tournant cette roue, la corde est tendue progressivement, ce qui permet de régler sa tension jusqu'à atteindre la note souhaitée (Mi). Un guide a été ajouté pour maintenir la corde alignée tout au long de la manipulation. Ce système est entièrement mécanique et permet un ajustement précis et manuel de la tension de la corde sans nécessiter d'outils supplémentaires. Ce mécanisme servira également de base pour l'intégration future d'un moteur, qui automatisera cette opération.

5. Fabrication et assemblage

5.1 Fabrication des pièces

- Impression 3D des éléments mécaniques et des supports électroniques.
- Tests de compatibilité des pièces imprimées avec le corps existant.

5.2 Montage final

- Fixation du guide de corde et du système de maintien.
- Ajout du compartiment électronique et préparation pour l'intégration future du moteur et des circuits.

6. Perspectives d'évolution

6.1 Automatisation de la tension

- Intégration d'un moteur électrique (moteur pas à pas).
- Programmation du microcontrôleur pour contrôler la tension automatiquement.

6.2 Optimisation du système électronique

- Installation d'un capteur pour un réglage précis et automatisé.
- Ajout d'une interface utilisateur pour surveiller la note liée à la corde.

Prochaines étapes :

1. Finalisation des pièces en CAO.
2. Impression et assemblage des éléments mécaniques.
3. Intégration des supports électroniques.

E - Alimentation

L'alimentation doit assurer une distribution stable et fiable des tensions requises pour les différents composants électroniques du système, tout en offrant un mode de fonctionnement sur secteur et un mode sur batterie rechargeable.

1. Objectifs

- Fournir les tensions nécessaires aux circuits électroniques de la guitare.
- Assurer une alimentation stable à partir d'une source secteur (transformateur 24V) et d'une batterie.
- Permettre la recharge de la batterie via l'alimentation secteur.
- Intégrer un switch ON/OFF pour désactiver complètement l'alimentation.
- Inclure des protections contre les surtensions, les surintensités et les courts-circuits.

2. Analyse des besoins

D'après les composants utilisés, voici les tensions requises :

- 12V pour les moteurs
- 6V pour l'arduino UNO R4 minima
- 5V pour les actionneurs(electroaimants)

Tension	Composants associés	Courant min (A)	Courant max (A)	Puissance min (W)	Puissance max (W)
12V	Moteur	0A	3.4A	0W	40.8W
6V	Arduino UNO R4 Minima	0.07A	0.07A	0.042W	0.042W
5V	8 electroaimants	0A	8A	0W	40W

Ces valeurs sont exactes sur une seule guitare monocorde.

- Puissance minimum (Arduino seul) : 0.42W
- Puissance maximum (moteur + 8 electroaimants + Arduino) : 81.22W

3. Architecture globale

L'alimentation doit distribuer les tensions nécessaires aux différents composants tout en permettant un fonctionnement en mode secteur et en mode batterie.

- Mode secteur

Un transformateur 24V AC est utilisé pour alimenter le système :

- Redressement avec un pont de diodes.
- Filtrage via condensateurs électrolytiques.
- Convertisseurs Buck pour abaisser la tension :
 - 12V pour les moteurs pas à pas
 - 6V pour l'Arduino UNO R4 Minima.
 - 5V pour les actionneurs
- Recharge de la batterie en parallèle.

- Mode Batterie

Lorsque le secteur est déconnecté, la batterie prend le relais et fournit directement 12V.

- Deux convertisseurs Buck :
 - 12V → 6V alimente l'Arduino.
 - 12V → 5V alimente les actionneurs

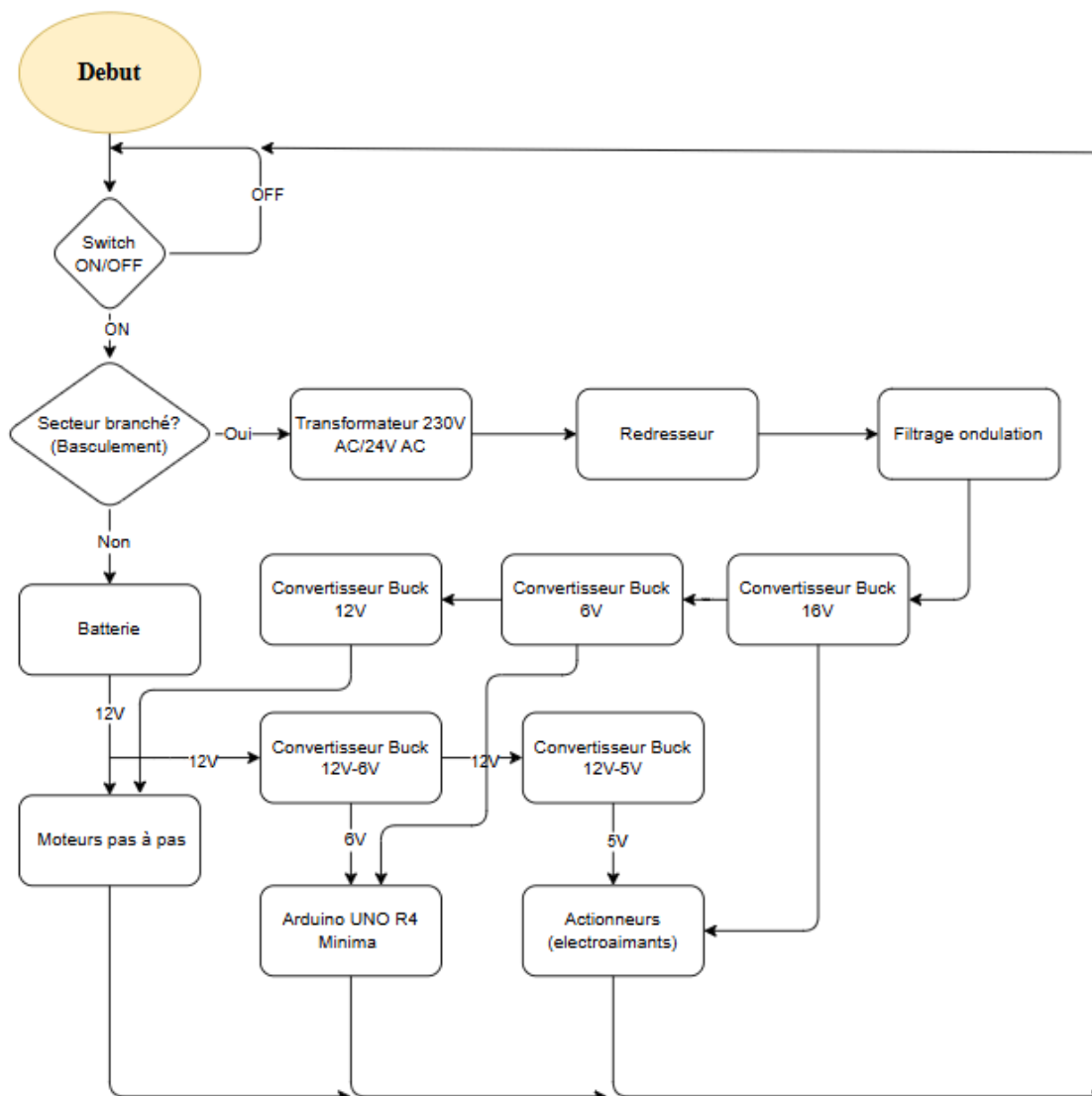
Les diodes Schottky permettent un basculement automatique entre secteur et batterie.

- Switch ON/OFF

Un interrupteur DPST (Double Pôle) est placé avant la distribution pour couper complètement toutes les alimentations (secteur et batterie).

- Mode ON → Le circuit est alimenté par secteur ou batterie (priorité au secteur).
- Mode OFF → Plus aucune tension n'est distribuée dans le circuit.

3.1 Diagramme globale de l'alimentation



4. Composants utilisés

Composant	Rôle	Valeur / Modèle
Transformateur	Convertit 230V AC en 24V AC	24V AC
Pont de diodes	Redressement AC → DC	*
Condensateurs	Filtrage des ondulations	*
Convertisseurs Buck	Régulation de tension	12V, 6V, 5V (NE555 + MOSFET)
Diodes Schottky	Protection et basculement secteur/batterie	*
Fusibles	Protection contre surintensités	*
Interrupteur ON/OFF	Coupure totale du circuit	DPST (Double Pôle)
Régulateur de charge	Charger la batterie	*

* : Composants pas choisis ou pas disponibles à l'IUT.

5. Justification des choix techniques

5.1 Choix du Transformateur 24V

- Alimentation secteur standardisée : Disponible à l'IUT et compatible avec les convertisseurs de tension.
- Tension suffisante pour générer les rails 12V, 6V et 5V après redressement et régulation.
- Permet la recharge de la batterie avec une régulation adaptée.

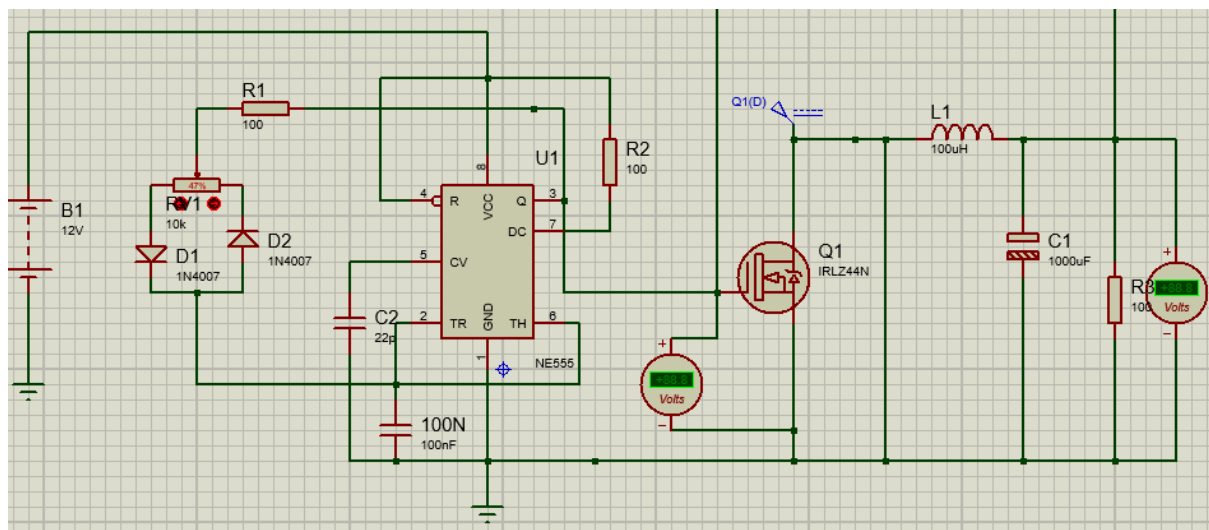
5.2 Choix des convertisseurs Buck

Ce convertisseur permet de réduire la tension d'alimentation à la valeur nécessaire aux différents composants.

Pourquoi utiliser des convertisseur Buck plutôt que des régulateurs linéaires?

Un convertisseur BUCK est plus efficace qu'un régulateur linéaire car il fonctionne en mode découpage, limitant les pertes d'énergie sous forme de chaleur.

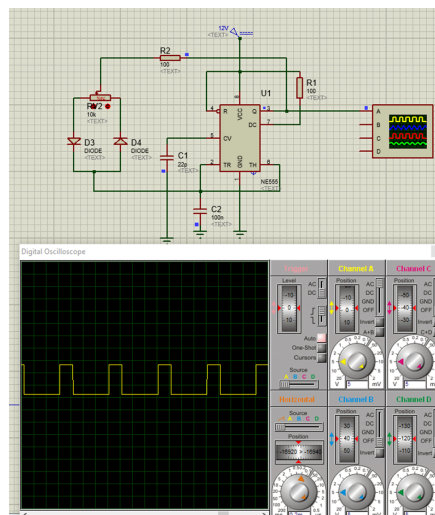
Voici le schéma du convertisseur BUCK géré par un signal PWM:



Fonctionnement :

Le signal PWM active et désactive le transistor (MOSFET). Quand le transistor est fermé, **l'énergie est stockée dans l'inductance**. Quand il est ouvert, **l'inductance restitue l'énergie vers la charge**. La diode et le condensateur assurent le lissage de la tension de sortie.

Le PWM est généré grâce à un circuit NE555 configuré en mode astable.



Pourquoi un NE555 pour générer le signal PWM?

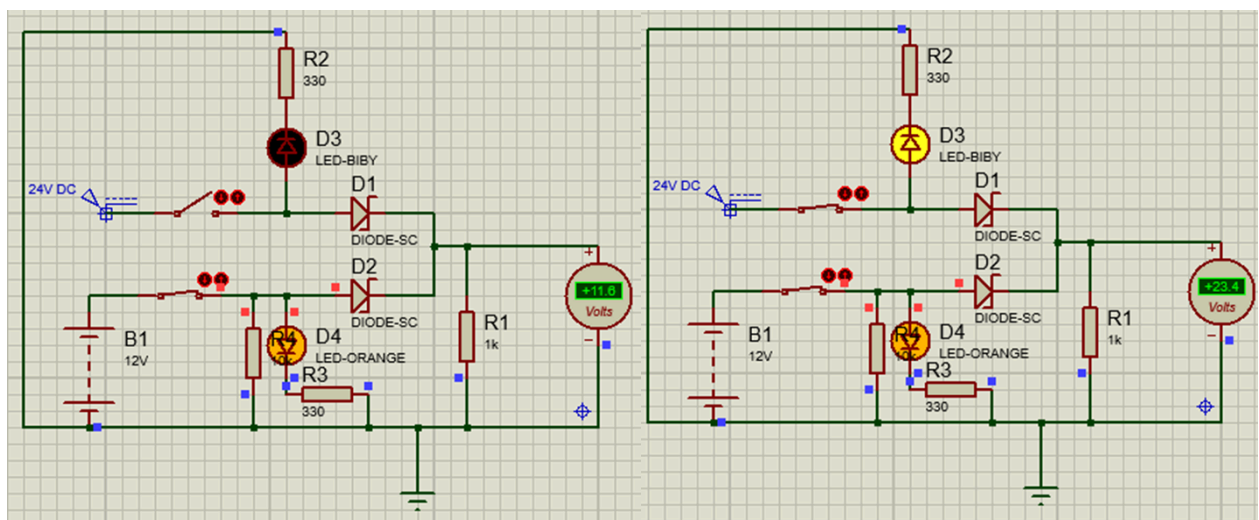
Il est simple à mettre en œuvre et permet de générer un signal PWM stable avec peu de composants.

- **Configuration en mode astable :**

Deux résistances et un condensateur déterminent la fréquence et le rapport cyclique du signal. Le rapport cyclique est ajustable grâce à un potentiomètre pour contrôler la tension de sortie du Buck.

5.3 Choix du Basculement Secteur/Batterie

Ce circuit permet de passer automatiquement sur la batterie en cas de coupure secteur.



Pourquoi utiliser des diodes Schottky pour gérer le basculement?

Les diodes Schottky sont utilisées pour le basculement secteur/batterie car elles ont une **faible chute de tension** (0.2V à 0.5V), limitant les pertes d'énergie et maintenant une tension stable. De plus, leur **commutation rapide** permet un passage instantané entre le secteur et la batterie en cas de coupure, garantissant ainsi une alimentation continue et efficace.

Fonctionnement :

Deux diodes sont utilisées pour sélectionner automatiquement la source ayant la tension la plus élevée. Lorsque le secteur est présent, la diode reliée au secteur conduit et alimente le circuit. Lorsque le secteur disparaît, la diode reliée à la batterie prend le relais instantanément. Cela empêche toute coupure d'alimentation du système.

Alternative possible: Un MOSFET en commutation pour encore réduire la chute de tension, mais cette option est plus complexe à mettre à l'œuvre.

5.4 Choix du Switch ON/OFF

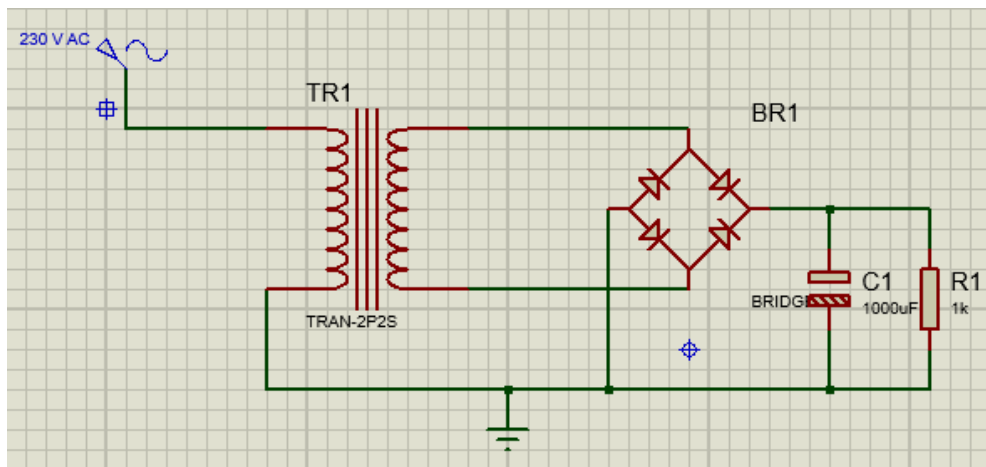
Pourquoi utiliser un interrupteur DPST (Double Pôle)?

- Permet de couper totalement l'alimentation, que ce soit sur secteur ou batterie.
- Isolation complète des circuits en mode OFF ce qui accroît la sécurité.

Alternative possible: Un interrupteur MOSFET piloté électroniquement, mais plus compliqué à intégrer.

5.5 Choix du redresseur et filtre

Le redresseur convertit la tension alternative du secteur en tension continue grâce à un pont de diodes.

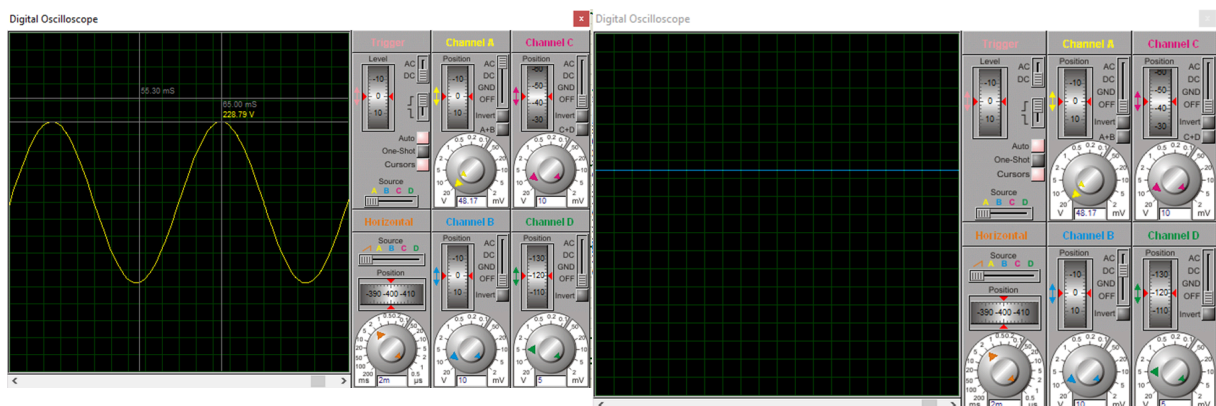


Pourquoi un pont de diode?

Un pont de diode permet de redresser les deux alternances du signal alternatif, garantissant ainsi une tension continue plus stable qu'un simple redresseur à une seule diode.

Fonctionnement :

La tension alternative est appliquée à l'entrée du pont. Pendant l'alternance positive, deux diodes conduisent et permettent au courant de passer dans un sens. Pendant l'alternance négative, les deux autres diodes conduisent, redressant l'autre moitié du signal. On obtient alors une tension pulsée qu'il faut lisser avec un condensateur. Ci-dessous les résultats de simulation du redresseur :



6. Protection du Système

- Diodes TVS pour absorber les surtensions.
- Varistances (MOV) pour protéger contre les pics de tension secteur.
- Fusibles en entrée pour éviter les surintensités.
- Protection contre la décharge excessive pour la batterie.

IV - Jalon des versions du document

Version du document	Contenu ajouté	Chargé du document	Date
1.0	Tableau de matières + Avant propos + Rappel des besoins	Lathro-Seri	01/02/2025
1.1	Justification des choix pour la programmation (IDE + Carte de Développement)	Lathro-Seri	02/02/2025
1.2	Justification des choix pour la programmation (Interface graphique)	Lathro-Seri	06/02/2025
1.3	Justification des choix pour le projet entier (Fonctionnalités du produit + Partie Mécanique + Alimentation)	Lathro-Seri Ramirez Victor Idmond Benjy	09/02/2025
1.4	Ajustement du rendu du texte + Modification des informations sur les composants	Lathro-Seri Ramirez Victor	09/02/2025
1.5	Amélioration de la partie Alimentation	Ramirez Victor	08/03/2025

D'après le Règlement des Examens de CY Cergy-Paris, soumis à l'approbation du Conseil d'établissement du 3 octobre 2023, l'article 6-1-3 nous autorise à utiliser de l'intelligence artificielle tant que c'est explicitement cité dans le document et que dans notre cas, l'intelligence artificielle n'a pas été expressément interdit.

***6-1-3** Tout recours à l'intelligence artificielle, quand il n'est pas expressément interdit, devra être mentionné de façon explicite en tant qu'emprunt ou citation d'une source externe. Le non-respect de cette règle sera considéré comme une fraude.*