

CY IUT – GEII Neuville

Document de Conception

Projet tutoré Mixeur Sous-système Entrées

Document rédigé par : MUHAMMAD Asad
Version : 1.1 [16 février 2025]

Historique des modifications et révisions de ce document :

N° de version	Date	Auteur	Description et circonstances de la modification
V 0		MUHAMMAD Asad	Brouillon : première version, modèle fourni.
V 1.0	09/02/2025	MUHAMMAD Asad	Première rédaction complète du document.
V1.1	16/02/2025	MUHAMMAD Asad	Corrections (valeurs numériques ...etc)

Grilles d'évaluations

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

Avant-propos¹

Ce document présente la conception d'une table de mixage audio destiné à offrir une solution compacte et performante pour le traitement et la gestion de signaux sonores.

L'approche adoptée dans ce document vise à fournir une description claire de l'architecture du système, des choix techniques et des principales fonctionnalités, sans entrer dans les détails de mise en œuvre. Il sert de base à la compréhension globale du projet avant d'aborder les aspects plus spécifiques liés à la fabrication et au développement logiciel.

¹ Référence : Gestion de projet, 50 outils pour agir ; F. Bouchaoui, Y. Dentinger, O. Englander ; Vuibert ; 2014. Expression du besoin et cahier des charges fonctionnel ; J. Bernard-Bousières ; AFNOR ; 2012.

Table des matières

Avant-propos.....	1
I. Contexte du projet	6
A. <i>Situation et description</i>	6
B. Besoins et objectifs.....	6
II. Rappel des fonctions	7
III. Choix des Outils de travail	8
A. Conception du PCB avec Altium Designer	8
1. Altium Designer : Outil Principale :	8
2. Alternatives : Ki-Cad et Proteus pour certaines parties du projet	8
B. Simulations circuits LTSPICE	9
1. LTSPICE : Outil principal pour la simulation de circuits.....	9
2. Alternative : Proteus pour des simulations et conception PCB	9
C. Choix de « cerveau » de notre système.	10
1. Caractéristiques de la Raspberry Pi 4 :	10
2. Pourquoi une Raspberry Pi 4 et non un microcontrôleur ?	10
D. Choix de langage de programmation.....	11
IV. Conception détaillée des fonctions.....	12
A. Assurer la Fiabilité et maintenance.....	12
1. Système modulaire avec sous-systèmes indépendants.....	12
1.1. Justification du découpage en modules	12
1.2. Organisation des cartes	12
1.3. Connectique et assemblage	12
2. Protection des entrées audio contre les surtensions (FT9-1)	13
3. Protection des alimentations contre les courts-circuits	17
B. Gestion de l'entrée audio analogique	18
1. Conditionner le signal analogique (filtrage).....	18
1.1. Choix du filtre : un filtre passe-bas actif d'ordre 2	18
2. Régulation du gain du signal	21
2.1. Montage de l'amplificateur	21
3. Application d'un décalage sur le signal analogique	23
3.1. Source de tension stable et fixe	23
3.2. Montage Offset.....	24
3.3. Justification et mise en œuvre	25
4. Conversion du signal analogique en numérique	26

4.1.	Choix du Convertisseur Analogique-Numérique	27
C.	Gestion de l'entrée audio numérique	28
1.1.	Connectique et interface MIDI	28
1.2.	Transport des données MIDI	28
1.3.	Canaux MIDI	29
1.4.	Structure des messages MIDI	29
2.	Synchronisation des horloges MIDI avec le traitement audio	31
3.	Conversion des données MIDI en signal audio	32
D.	Traitement numérique du signal audio	33
1.	Application de filtres et algorithmes de mixage numérique.....	33
2.	Gestion du mixage audio en temps réel	33
3.	Fournir un IHM pour la gestion des paramètres de mixage et effets.	33
4.	Restitution du signal audio traité.....	33
5.	Sortie stéréo ou ligne pour restitution sonore.....	33
E.	Amplification du signal audio	33
1.	Amplification en tension et en courant pour restitution optimale.....	33
F.	Gestion de l'alimentation électrique.....	33
1.	Alimentation via batterie rechargeable	33
2.	Recharge via adaptateur secteur ou alimentation DC	33
G.	Format compact et ergonomie	33
1.	Conception d'un boîtier robuste et léger.....	33
2.	Optimisation des dimensions pour transport et intégration	33
V.	Développement en cours et prochaines étapes	34

Répartition des responsabilités dans la rédaction du dossier

Partie du dossier	Responsable
Avant-propos I. Contexte du projet A. Situation et description B. Besoins et objectifs II. Rappel des fonctions	Muhammad Asad & VRONSKYI Volodymyr
III. Choix des outils de travail	Assignation en fonction des outils utilisés dans chaque sous-système
IV. Conception détaillée des fonctions A. Assurer la fiabilité et la maintenance 1. Système modulaire avec sous-systèmes indépendants 2. Protection des entrées audio contre les surtensions	Muhammad Asad & VRONSKYI Volodymyr
3. Protection des alimentations contre les courts-circuits	RAMIRES MATOS Victor & IDMOND Benjy
B. Gestion de l'entrée audio analogique 1. Conditionner le signal analogique (pré-amplification et filtrage) 2. Régulation dynamique du gain du signal 3. Application d'un décalage sur le signal analogique 4. Conversion du signal analogique en numérique	Muhammad Asad & VRONSKYI Volodymyr
C. Gestion de l'entrée audio numérique 1. Réception des signaux MIDI via interface DIN ou USB 2. Synchronisation des horloges MIDI avec le traitement audio 3. Conversion des données MIDI en signal audio	Muhammad Asad & VRONSKYI Volodymyr
D. Traitement numérique du signal audio 1. Application de filtres et algorithmes de mixage numérique 2. Gestion du mixage audio en temps réel ...3. Fournir un IHM pour la gestion des paramètres de mixage et effets.	YIGIT Enes & EL KARANI Djibril
4. Restitution du signal audio traité 5. Sortie stéréo ou ligne pour restitution sonore	LATHRO-SERI Nathan & ERBE Gabriel
E. Amplification du signal audio 1. Amplification en tension et en courant pour restitution optimale	LATHRO-SERI Nathan & ERBE Gabriel
F. Gestion de l'alimentation électrique 1. Alimentation via batterie rechargeable 2. Recharge via adaptateur secteur ou alimentation DC G. Format compact et ergonomie 1. Conception d'un boîtier robuste et léger 2. Optimisation des dimensions pour transport et intégration	RAMIRES MATOS Victor & IDMOND Benjy

Table des Figures

Figure 1 Organisation des cartes en montage verticale	12
Figure 2 Connecteur femelle 2.54mm	12
Figure 3 Connecteur mâle angle droit 2.54mm.....	12
Figure 4 Vue 3D de montage verticale avec les connecteurs	12
Figure 5 Montage protection pics de surtension, avec doide TVS bidirectionnelle	13
Figure 6 Montage protection pics de surtension avec 2 diodes Zeners	13
Figure 7 Circuit de clamping entrée analogique avec diode Zener	14
Figure 8 Circuit de clamping entrée analogique avec AOP.....	14
Figure 9 Montage simple optocoupleur à phototransistor, $F_{max} \leq 25\text{kHz}$	15
Figure 10 Montage améliorée optocoupleur à phototransistor. $F_{max} \leq 125\text{kHz}$	15
Figure 11 Montage Filtre Sallen-Key d'ordre 2	18
Figure 12 Analog filter wizard.....	19
Figure 13 Paramètres de filtre sur Analog filter wizard.....	19
Figure 14 Filtre proposé par analog filter wizard en fonction des parametres	20
Figure 15 Diagramme de Bode (Gain) de filtre.....	20
Figure 16 Diagramme de Bode (Phase) de filtre	20
Figure 17 Montage Amplificateur Inverseur.....	21
Figure 18 Montage 2 régulation du gain du signal	22
Figure 19 Régulateur de tension variable/programmable pour une référence de tension stable.....	23
Figure 20 Montage simplifié regulateur de tension variable/programable.	23
Figure 21 Montage amplificateur différentiel	24
Figure 22 Simulation Offset avec montage différentiel	25
Figure 23 Numérisation d'un signal et impact de fréquence d'échantillonnage.....	26
Figure 24 Tableau présentant quelques exemples des standard qualité audio	27
Figure 25 Trame MIDI	29
Figure 26 Format et le contenu des principaux types de message MIDI.	30
Figure 27 Notes MIDI.....	30
Figure 28 Format message MIDI Timecode	31

I. Contexte du projet

A. *Situation et description*

Le projet consiste à concevoir et réaliser une table de mixage audio automatique capable de traiter et de synchroniser plusieurs flux audios provenant d'instruments de musique. Ce dispositif sera intégré à l'Orchestrion, un système automatisé conçu pour reproduire les performances d'un orchestre en combinant différents instruments.

Développé dans le cadre de la SAE (Situation d'Apprentissage et d'Évaluation), ce projet sera présenté lors des Journées Portes Ouvertes de l'IUT de Cergy-Pontoise. Il met en avant la conception d'un dispositif électronique physique tout en respectant des contraintes écologiques et économiques.

B. Besoins et objectifs

L'objectif principal de ce projet est de valoriser les enseignements du BUT GEII (Bachelor Universitaire de Technologie en Génie Électrique et Informatique Industrielle) à travers la création d'une unité de contrôle audio automatisée.

Ce mixeur audio doit permettre :

- Le traitement et la synchronisation de plusieurs sources sonores en temps réel.
- Une intégration optimisée au sein de l'Orchestrion, garantissant une interaction fluide avec les instruments automatisés.
- Une gestion efficace du signal audio tout en maintenant une faible consommation énergétique.

En plus des contraintes techniques, le projet répond également à des enjeux écologiques et économiques, en favorisant l'utilisation de composants optimisés et durables.

II. Rappel des fonctions

- FT9 : Assurer la fiabilité et la maintenance
 - FT9-0 : Concevoir un système modulaire avec des sous-systèmes indépendants (entrée audio, traitement, interface utilisateur, alimentation).
 - FT9-1 : Garantir la sécurité électrique grâce à une protection contre les surtensions et courts-circuits sur les entrées Analogiques et Numériques.
 - ~~FT9-2 : Garantir la sécurité électrique grâce à une protection contre les surtensions et courts-circuits sur les Alimentations.~~
- FT2, FT3 : Gérer l'entrée audio analogique
 - FT2-0 : Conditionner le signal analogique via pré amplification et filtrage.
 - FT2-1 : Réguler le gain du signal d'entrée.
 - FT2-2 : Appliquer un décalage (offset) sur le signal analogique AC.
 - FT3 : Convertir le signal analogique en numérique (ADC) avec une résolution minimale de 12 à 16 bits et une fréquence d'échantillonnage de 44,1 kHz.
- FT1, FT6 : Gérer l'entrée audio numérique
 - FT1-0 : Réceptionner les signaux MIDI.
 - FT6 : Synchroniser les horloges MIDI avec le traitement audio interne.
 - FT1-1 : Convertir les données MIDI en audio grâce à une synthèse numérique.
- ~~FT5, FT6, FT7, FS1, FS2 : Assurer le traitement numérique du signal et produire un signal sonore~~
 - ~~FT5-0 : Appliquer des algorithmes de filtrage numérique (IIR/FIR) et de mixage pour ajuster les niveaux sonores et ajouter des effets.~~
 - ~~FT5-1 : Gérer le mixage audio en temps réel avec une allocation dynamique des canaux.~~
 - ~~FT7, FS1, FS2 : Fournir une IHM pour la gestion des paramètres de mixage et effets.~~
 - ~~FT6-0 : Restituer le signal audio~~
 - ~~FT6-1 : Fournir une sortie stéréo pour deux haut-parleurs intégrés ou une sortie ligne (non amplifiée).~~
- ~~FT6-2 : Amplifier le signal audio~~
 - ~~Assurer l'amplification en tension et en courant du signal audio pour une restitution optimale.~~
- ~~FT8 : Gérer l'alimentation électrique~~
 - ~~Assurer une alimentation via une batterie rechargeable.~~
 - ~~Permettre la recharge via un adaptateur secteur, une alimentation DC.~~
- ~~FC6, FT9 : Garantir un format compact et une bonne ergonomie~~
 - ~~Concevoir un boîtier robuste et léger, en matériaux résistants aux chocs et adaptés aux manipulations fréquentes.~~
 - ~~Optimiser les dimensions pour faciliter le transport et l'intégration dans un setup mobile.¹~~

¹ Les Fonction barrées seront rédigées dans le document de fabrication de sous système

III. Choix des Outils de travail

A. Conception du PCB avec Altium Designer

1. Altium Designer : Outil Principale :



Pour la réalisation du circuit imprimé, Altium Designer a été choisi en raison de ses capacités avancées en conception multi-cartes, ce qui est particulièrement adapté à l'architecture modulaire du projet.

La conception du PCB suit plusieurs étapes :

- **Schématisation du circuit** : Création des schémas électriques des différentes cartes en définissant les composants et leurs connexions.
- **Placement des composants** : Positionnement des composants sur le PCB en prenant en compte l'optimisation de l'espace et les contraintes thermiques.
- **Routage des pistes** : Traçage des connexions électriques en respectant les règles de conception pour minimiser les interférences et assurer la qualité du signal.
- **Vérifications** : Analyse du design via les outils de validation d'Altium Designer afin d'éviter les erreurs électriques et mécaniques.
- **Intégration dans un projet multi-cartes** : Ajout et gestion des différentes cartes (carte mère et cartes modules) dans un projet multi-cartes d'Altium Designer, facilitant ainsi la cohérence des connexions et l'optimisation du routage inter-cartes.
- **Génération des fichiers de fabrication** : Production des fichiers Gerber et des documents nécessaires à la fabrication et à l'assemblage du circuit imprimé.

Grâce à Altium Designer, l'architecture **modulaire et verticale** du système est optimisée, assurant une intégration mécanique et électrique efficace entre la carte mère et les modules.

2. Alternatives : Ki-Cad et Proteus pour certaines parties du projet

Pour les circuits qui ne nécessitent pas une architecture modulaire et qui doivent uniquement gérer des entrées et sorties, d'autres logiciels peuvent être employés :

Ki-Cad : Open-source, idéal pour des conceptions simples et collaboratives sans restriction de licence.

Proteus : Disponible à l'université, utile pour des schémas rapides et des simulations avant fabrication.

Ces outils offrent une flexibilité supplémentaire en fonction des besoins spécifiques du projet, tout en maintenant Altium Designer comme principal logiciel pour la conception des modules et de la carte mère.

B. Simulations circuits LTSPICE

1. LTSPICE : Outil principal pour la simulation de circuits



LTSpice est privilégié pour la simulation des circuits en raison de son large choix de composants et de la facilité de création des modèles de simulation. Il offre une flexibilité optimale pour concevoir des simulations précises et réalistes de circuits complexes. L'outil permet également d'ajuster facilement les paramètres des composants, ce qui facilite la modélisation fine des comportements électriques. Grâce à sa puissance et à ses capacités de simulation rapide, LTSpice permet d'identifier et de corriger d'éventuelles erreurs avant la fabrication, réduisant ainsi les risques et les coûts.

2. Alternative : Proteus pour des simulations et conception PCB



PROTEUS

En alternative, Proteus peut être utilisé pour les simulations de circuits. Il permet de passer facilement de la simulation à la conception du PCB, ce qui simplifie l'ensemble du processus de conception. Toutefois, Proteus est limité par un choix restreint de composants par rapport à LTSpice, ce qui peut rendre son utilisation moins flexible pour certaines parties complexes nécessitant des composants spécifiques.

C. Choix de « cerveau » de notre système.

Notre système doit gérer plusieurs signaux analogiques et numériques en entrée et en sortie, tout en fournissant une interface utilisateur permettant de contrôler et d'ajuster différents paramètres. Pour cela, nous avons besoin d'un dispositif capable d'exécuter plusieurs tâches en parallèle, de traiter des signaux en temps réel et de communiquer avec divers périphériques.

Après analyse des différentes options, nous avons opté pour une Raspberry Pi 4, un micro-ordinateur monocarte (SBC) puissant et polyvalent, développé par la Fondation Raspberry Pi.

1. Caractéristiques de la Raspberry Pi 4 :

La Raspberry Pi 4 est équipée d'un processeur Broadcom BCM2711, un quad-core Cortex-A72 (ARM v8) cadencé à 1,5 GHz, offrant des performances nettement supérieures aux générations précédentes. Elle dispose de plusieurs interfaces de communication et d'entrées/sorties essentielles pour notre application :

- **40 broches GPIO** configurables pour différents usages.
 - UART : 6 interfaces (dont certaines partagées avec d'autres fonctions)
 - SPI : 6 interfaces matérielles.
 - I²C : 6 interfaces matérielles.
 - PWM : Jusqu'à 2 canaux matériels, plus d'autres via software.
 - GPIO : 28 broches utilisables pour des entrées/sorties générales.
- **Connectivité avancée :**
 - 2× USB 3.0 + 2× USB 2.0
 - 2× micro-HDMI (résolution jusqu'à 4K)
 - Ethernet Gigabit et Wi-Fi 5 GHz pour la communication réseau
 - Slot microSD pour le stockage du système et des données

2. Pourquoi une Raspberry Pi 4 et non un microcontrôleur ?

Contrairement aux microcontrôleurs comme les STM32 ou ESP32, la Raspberry Pi 4 dispose d'un système d'exploitation (Linux) qui permet d'exécuter plusieurs processus simultanément, ce qui est essentiel pour notre projet. Elle facilite également le développement grâce à la prise en charge de multiples langages de programmation (Python, C, C++, JavaScript, etc.), tandis que les microcontrôleurs nécessitent souvent un code unique compilé en firmware.

En comparaison avec les FPGA, bien que ceux-ci soient plus performants pour le traitement parallèle, ils nécessitent une programmation bas niveau (VHDL/Verilog) et ne disposent pas d'un système d'exploitation intégré, rendant la gestion des tâches complexes plus contraignante.

Avantages pour notre projet :

- Puissance de calcul suffisante pour le traitement du signal et l'analyse audio.
- Compatibilité avec un large éventail de capteurs et périphériques via ses nombreuses interfaces GPIO.
- Possibilité d'exécuter plusieurs scripts et processus en parallèle, notamment pour la gestion des signaux et l'interface utilisateur.
- Facilité d'utilisation et de développement (comparé à FPGA) grâce à un environnement Linux complet.

Grâce à ces caractéristiques, la Raspberry Pi 4 s'impose comme le choix idéal pour notre projet, offrant un équilibre optimal entre performance, flexibilité et accessibilité.

D. Choix de langage de programmation

Pour programmer notre Raspberry Pi, nous avons décidé d'utiliser le langage C comme langage principal. Ce choix est principalement motivé par plusieurs raisons :

- **Performance** : Le langage C permet un contrôle précis sur les ressources matérielles et optimise l'utilisation du processeur, ce qui est essentiel dans un projet embarqué où la gestion des performances est cruciale, notamment pour le traitement audio en temps réel.
- **Portabilité** : Le C est largement utilisé dans les environnements embarqués et supporté par de nombreuses plateformes, ce qui garantit une bonne compatibilité et pérennité du projet.
- **Bibliothèques et support** : Le Raspberry Pi dispose de nombreuses bibliothèques en C, notamment pour interagir avec les périphériques GPIO, les interfaces série (comme UART pour MIDI), et d'autres modules nécessaires au bon fonctionnement du projet.

Par ailleurs, **Bash** est également utilisé dans le projet pour automatiser certaines tâches, telles que la gestion des processus de démarrage du système, la configuration initiale des périphériques ou l'exécution de scripts de maintenance. L'utilisation de Bash permet de simplifier l'administration et la configuration du système sans avoir à développer des scripts complexes dans des langages de plus haut niveau.

Enfin, **Node.js** est utilisé pour certaines parties du projet nécessitant une gestion d'événements asynchrones, notamment pour l'interface utilisateur ou la gestion de la communication réseau. Node.js est particulièrement adapté pour ces tâches en raison de sa **nature non-bloquante** et de sa capacité à gérer des entrées/sorties simultanées de manière efficace. Il permet également de développer des applications Web ou des interfaces simples pour interagir avec notre système via un navigateur, ce qui facilite la gestion et le contrôle du mixeur audio à distance.

IV. Conception détaillée des fonctions

A. Assurer la Fiabilité et maintenance

1. Système modulaire avec sous-systèmes indépendants

L'architecture du système repose sur une conception modulaire visant à améliorer la flexibilité, la maintenance et l'évolutivité du dispositif. Cette approche permet d'individualiser les différentes fonctions du système en les répartissant sur plusieurs cartes distinctes.

1.1. Justification du découpage en modules

Plutôt qu'un circuit unique intégrant l'ensemble des fonctionnalités, le choix d'une architecture modulaire a été retenu pour plusieurs raisons :

- Maintenance facilitée : En cas de panne ou d'évolution, il est possible de remplacer un module spécifique sans impacter l'ensemble du système.
- Optimisation de l'espace : Le montage avec des modules permet une conception plus compacte et structurée.
- Facilité d'assemblage et d'évolution : Des améliorations futures peuvent être intégrées en remplaçant ou en ajoutant de nouveaux modules.

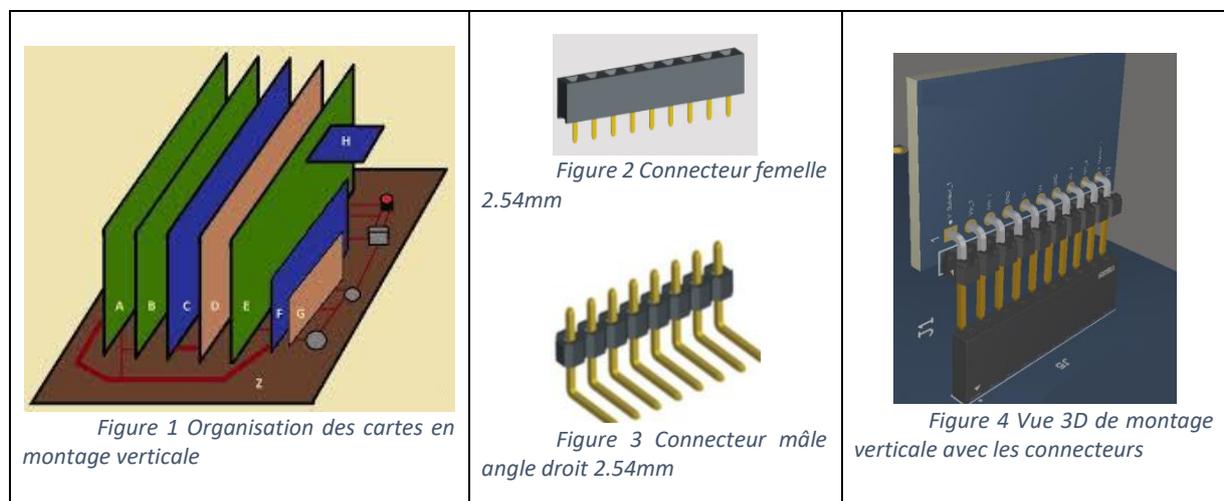
1.2. Organisation des cartes

Le système est structuré autour de deux types de cartes principales :

- Carte mère : Elle assure l'interconnexion entre les différents modules et gère l'alimentation et la distribution des signaux.
- Cartes modules : Chacune d'elles remplit une fonction spécifique et se connecte à la carte mère via un connecteur dédié. Ces cartes peuvent inclure des fonctions comme le conditionnement du signal, la conversion analogique-numérique (ADC), le traitement numérique ou l'amplification.

1.3. Connectique et assemblage

Les cartes modules sont montées verticalement sur la carte mère à l'aide de connecteurs headers mâles coudés sur les modules et connecteurs femelles 2,54 mm sur la carte mère. Cette disposition permet un gain de place, une meilleure accessibilité aux composants et une maintenance facilitée, tout en garantissant une connexion robuste entre les différents éléments du système.



2. Protection des entrées audio contre les surtensions (FT9-1)

Pour garantir la fiabilité et la durabilité du système, plusieurs protections électriques sont prises en compte dans la conception :

- **Protection contre les surtensions** : les entrées du système intègrent des diodes TVS (Transient Voltage Suppression) afin d'absorber les pics de tension et de préserver les composants sensibles.

Deux circuits à diodes sont proposés : l'un utilisant une diode TVS, l'autre basé sur deux diodes Zener offrant un fonctionnement similaire.

Dans la simulation ci-dessous, une diode TVS bidirectionnelle est utilisée. On observe que celle-ci absorbe efficacement les pics de tension dépassant sa tension de claquage (bleu : entrée, vert : sortie).

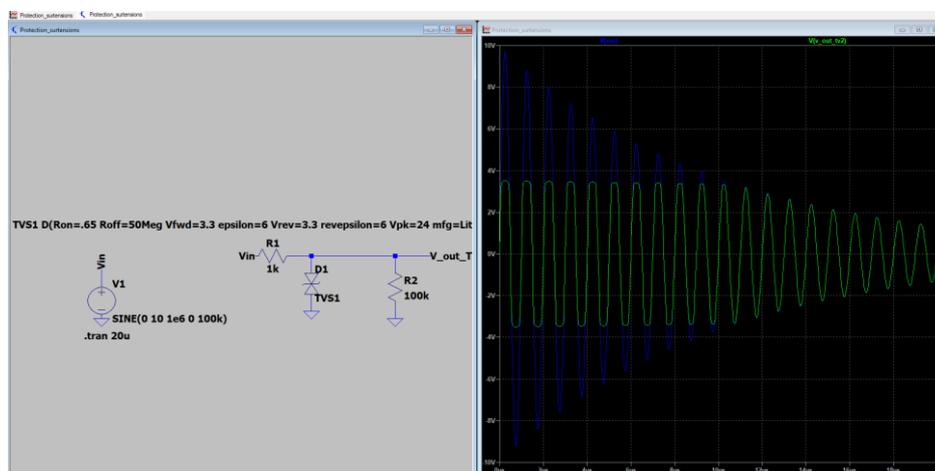


Figure 5 Montage protection pics de surtension, avec diode TVS bidirectionnelle

De même, la simulation suivante illustre le comportement d'un circuit avec deux diodes Zener, permettant d'absorber les surtensions au-delà de la tension de claquage.

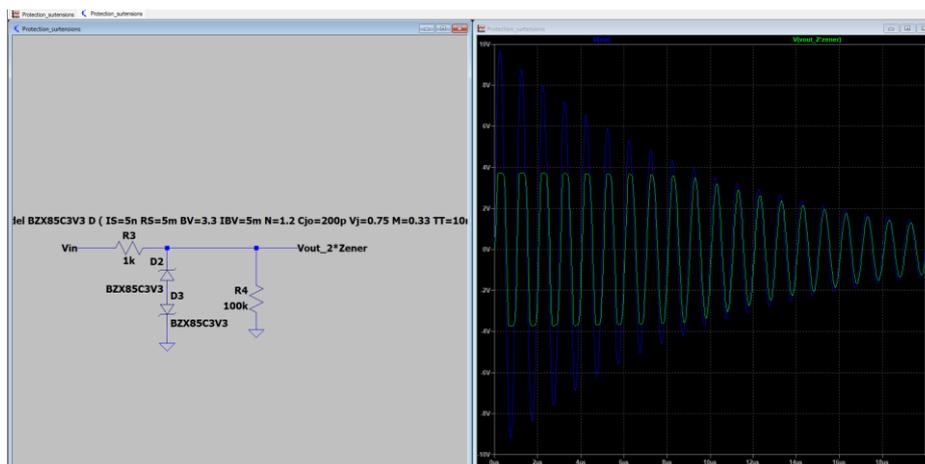


Figure 6 Montage protection pics de surtension avec 2 diodes Zeners

Ces circuits permettent de protéger les entrées de notre système contre les pics de surtensions, le 2eme on peut réaliser le 2eme montage avec les composants disponible à l'IUT, donc on choisit le 2eme montage.

- **Circuits de clantage / Limitation de la tension des entrées analogiques** : L'intégration de diodes Zener, Schottky ou d'un montage à amplificateur opérationnel permet de limiter la tension d'entrée des ADC à une valeur prédéfinie, assurant ainsi leur protection contre les surtensions (exemple : 3.3V pour les entrées audio).

Deux montages sont proposés : l'un utilisant une diode Zener, l'autre un amplificateur opérationnel.

La simulation suivante illustre le circuit de clantage avec une diode Zener (basé sur le même principe que montage en Figure 5 et Figure 6). Lorsque la tension aux bornes de la diode dépasse sa tension de clantage, celle-ci devient conductrice et limite la tension en sortie.

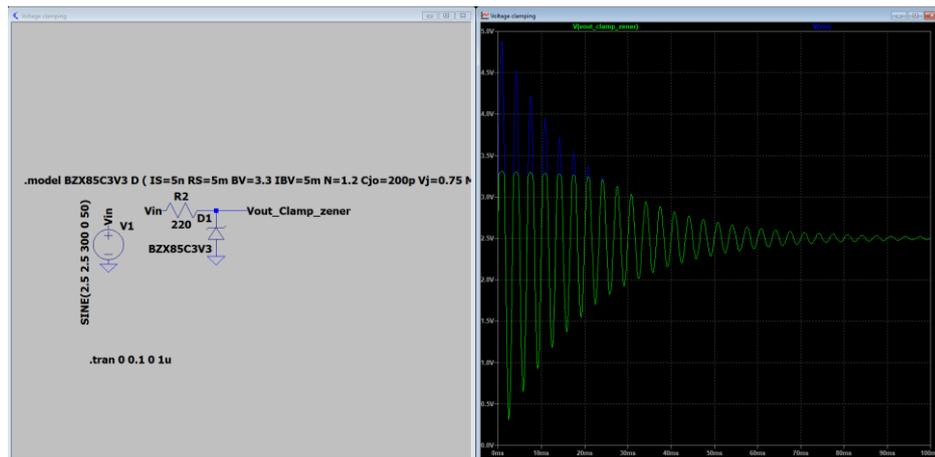


Figure 7 Circuit de clantage entrée analogique avec diode Zener

La simulation suivante montre le circuit de clantage utilisant un amplificateur opérationnel en montage non inverseur avec un gain de 1. La sortie de l'AOP ne pouvant pas dépasser sa tension d'alimentation, elle est automatiquement limitée lorsque la tension d'entrée excède cette valeur maximale.

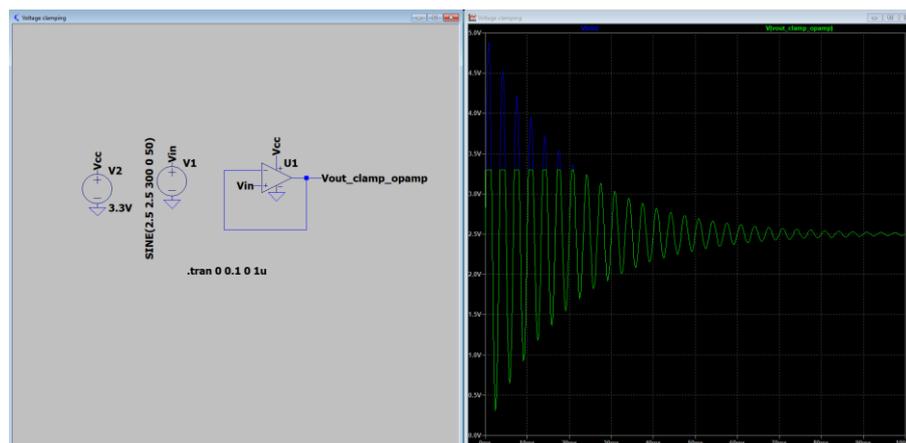


Figure 8 Circuit de clantage entrée analogique avec AOP

Nous avons choisi d'utiliser le deuxième montage avec un AOP, car notre circuit intègre déjà des amplificateurs opérationnels. En limitant la tension d'alimentation de l'AOP, nous pouvons réaliser cette fonction. L'AOP sélectionné doit avoir une sortie Rail-to-Rail afin de garantir un fonctionnement optimal. L'un des modèles disponibles répondant à ces critères est le MCP6022, qui est également disponible dans le stock de l'université.

- **Isolation galvanique** : Afin de protéger les circuits sensibles et d'éviter les interférences, une isolation galvanique est envisagée sur certaines interfaces critiques, comme l'entrée MIDI et d'autres interfaces numériques.

Deux montages sont proposés, basés sur un optocoupleur à phototransistor.

Le premier montage, simple, est limité à une fréquence maximale d'environ 25 kHz. En effet, les caractéristiques internes du phototransistor imposent des temps de montée et de descente qui restreignent la fréquence de fonctionnement.

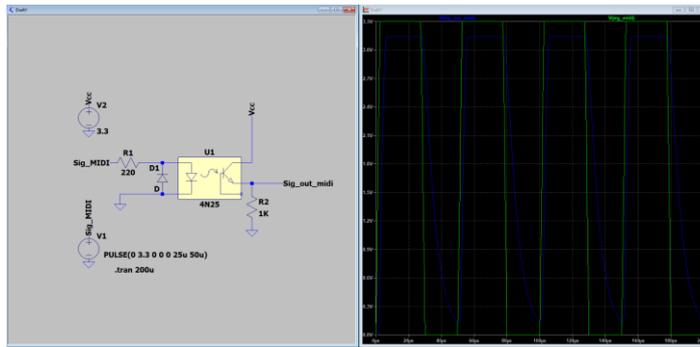


Figure 9 Montage simple optocoupleur à phototransistor, $F_{max} \leq 25\text{kHz}$

Le montage suivant améliore les performances en utilisant le même optocoupleur, mais avec un circuit optimisé inspiré d'un schéma publié dans le livre *Troubleshooting Analog Circuits* de Bob Pease, le montage proposé avec un inverseur en sortie permet d'avoir une fréquence max d'environ 125kHz.

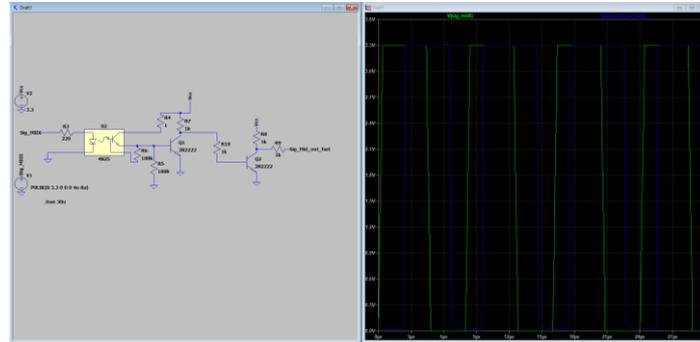


Figure 10 Montage améliorée optocoupleur à phototransistor. $F_{max} \leq 125\text{kHz}$

Ce circuit améliore le transfert de courant effectif (CTR) en ajoutant un transistor en sortie, permettant une meilleure commutation et une tension de sortie allant de rail à rail. Deux résistances supplémentaires optimisent également le temps de réponse :

- **R1** accélère le temps de descente en tirant la base du transistor de sortie vers le bas.
- **R2** applique une rétroaction positive sur la base du phototransistor, augmentant son courant de base lorsqu'il commence à conduire, ce qui accélère son activation.

Dans cette configuration, Pease recommande des valeurs de **2 MΩ pour R1 et 1 kΩ pour R2**.

Ce circuit introduit un léger retard sur le signal de sortie, mais celui-ci reste négligeable dans le cadre de notre application et ne pose pas de problème de performance.

Le choix du montage utilisé dépendra de la fréquence des signaux à isoler afin d'assurer une transmission fiable et adaptée aux exigences du système.

Pour les signaux basés sur les spécifications MIDI, dont le baud rate est de 31 250 bauds, le montage de Buson amélioré basé sur l'optocoupleur à phototransistor 4N25 est un bon choix. Il peut également être remplacé par un 6N138, ces deux composants étant des optocoupleurs de type Darlington.

En revanche, pour les signaux à fréquences plus élevées, tels que I2C et SPI, ce montage n'est pas adapté. D'autres optocoupleurs, comme le 6N137, peuvent être utilisés. Ce dernier dispose d'une sortie numérique et peut gérer des signaux allant jusqu'à 10 MBd, ce qui le rend parfaitement compatible avec ces signaux.

3. Protection des alimentations contre les courts-circuits

Rédaction dans le dossier de sous système

B. Gestion de l'entrée audio analogique

1. Conditionner le signal analogique (filtrage)

Le filtrage est une étape essentielle du conditionnement du signal analogique. Il permet d'éliminer les composantes indésirables, telles que le bruit haute fréquence ou les interférences, garantissant ainsi une meilleure qualité audio.

En plus de son rôle dans la purification du signal, le filtrage assure une entrée haute impédance, ce qui est crucial pour préserver l'intégrité du signal. Une haute impédance limite le courant entrant et empêche la source audio d'être « chargée », réduisant ainsi l'atténuation et la distorsion pour une transmission fidèle et sans perte de qualité.

Enfin, cette étape est essentielle pour l'adaptation du signal au traitement numérique, notamment en supprimant les fréquences supérieures à la bande utile afin d'éviter les artefacts tels que l'aliasing lors de la conversion analogique-numérique (CAN).

1.1. Choix du filtre : un filtre passe-bas actif d'ordre 2

Pour notre application, nous avons opté pour un filtre passe-bas actif de type Sallen-Key d'ordre 2. Ce choix offre un bon compromis entre simplicité de mise en œuvre, stabilité et efficacité dans l'atténuation des hautes fréquences.

Justification du choix d'un filtre d'ordre 4 :

Un filtre d'ordre 2 présente une pente d'atténuation de -12 dB/octave, ce qui est suffisant pour limiter les hautes fréquences au-delà de 20 kHz, tout en minimisant la complexité du circuit. Contrairement à un filtre d'ordre 4, il nécessite moins de composants actifs et réduit ainsi la consommation d'énergie et les erreurs de phase potentielles.

Caractéristiques du filtre :

- Type : Passe-bas actif (Sallen-Key)
- Ordre : 2
- Fréquence de coupure (f_c) : 20 kHz
- Gain : Unité (1x) pour éviter toute amplification ou atténuation non souhaitée

Voici le montage du filtre d'ordre 2 :

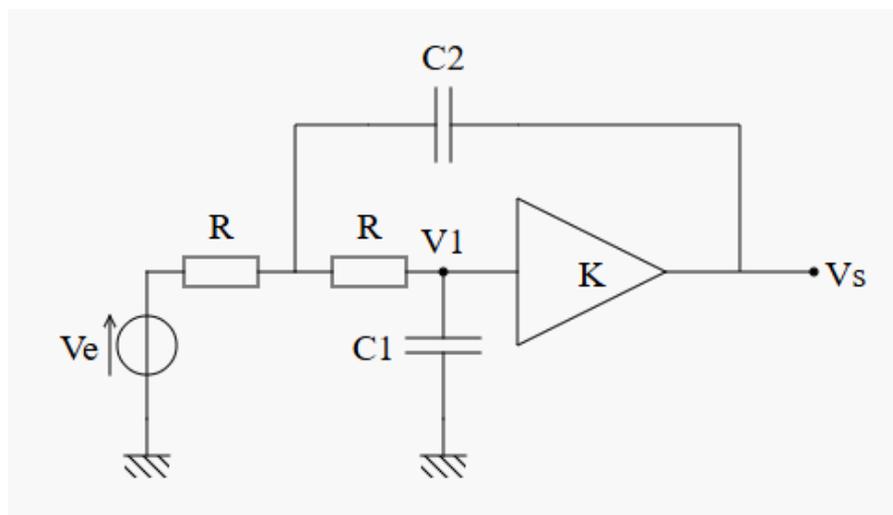


Figure 11 Montage Filtre Sallen-Key d'ordre 2

L'élément actif est un amplificateur de tension de gain K . Idéalement, l'amplificateur doit avoir une impédance d'entrée assez grande pour pouvoir être considérée comme infinie, et une impédance de sortie nulle. Il réalise la fonction suivante :

$$V_s(t) = KV_1(t)$$

Pour un ampli-op idéal, la fonction de transfert est de la forme suivante :

$$H(\omega) = A \frac{K}{1 + mj \frac{\omega}{\omega_c} + \left(j \frac{\omega}{\omega_c}\right)^2}$$

Avec :

$$\omega_0 = \frac{1}{R\sqrt{C_1 C_2}}$$

$$m = 2 \sqrt{\frac{C_1}{C_2}} + \sqrt{\frac{C_2}{C_1}} (1 - K)$$

Pour concevoir le filtre et calculer précisément les valeurs des composants, nous allons utiliser Analog Filter Wizard développé par Analog Devices. Cet outil permet de réaliser des filtres en fonction des paramètres souhaités et fournit des suggestions optimisées pour garantir des performances optimales. Il facilite également la simulation et l'ajustement des valeurs afin d'obtenir la meilleure réponse en fréquence possible.

Lien vers le site : <https://tools.analog.com/en/filterwizard/>

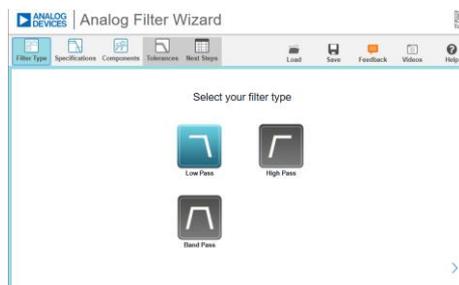


Figure 12 Analog filter wizard

Passband	
Gain:	0 dB
-1 dB	20k Hz
Stopband	
-10 dB	60k Hz
Filter Response	
<input type="range"/>	
Fewest Stages	Fastest Settling
2nd order Chebyshev 0.07 dB (1 stage)	

Figure 13 Paramètres de filtre sur Analog filter wizard

En utilisant le logiciel de conception de filtres, nous avons conçu le filtre suivant :

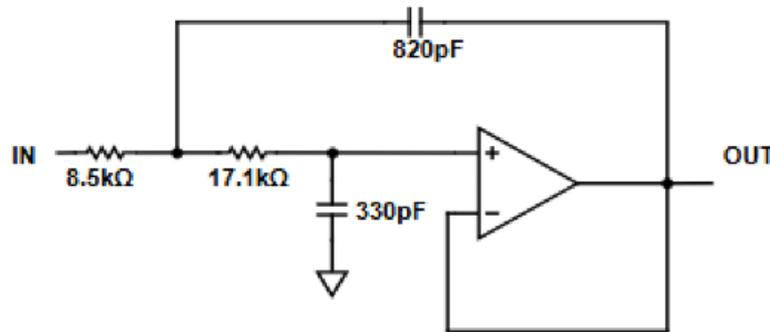


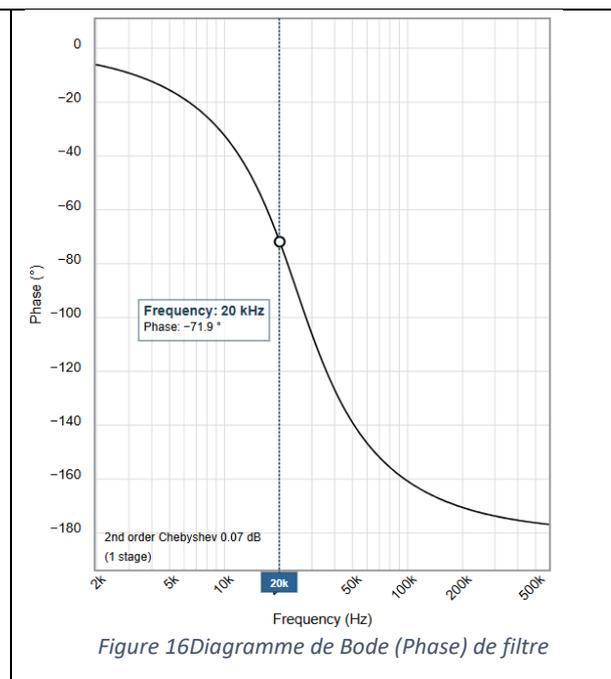
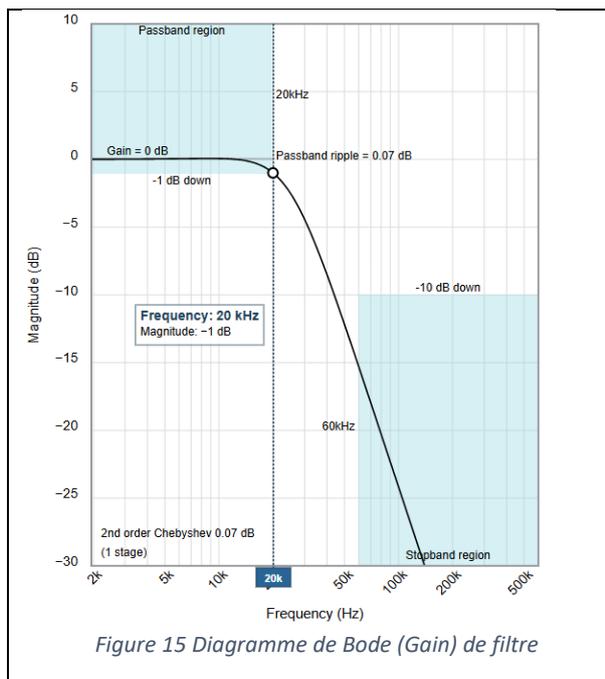
Figure 14 Filtre proposé par analog filter wizard en fonction des paramètres

Il s'agit d'un filtre de second ordre Chebyshev, avec une ondulation de 0,07 dB dans la bande passante. Ce filtre présente des caractéristiques de réponse en fréquence telles que :

- Une atténuation de -1 dB à 20 kHz.
- Une atténuation de -3 dB à 26,5 kHz.

En ce qui concerne la phase, le filtre présente une variation qui atteint -72° à la fréquence maximale de 20 kHz. Cette phase montre l'effet de la réactivité du filtre aux hautes fréquences, un aspect important à prendre en compte pour éviter des distorsions indésirables dans le signal, surtout dans les applications audio où la précision de phase est cruciale pour la qualité du son.

Le choix du filtre Chebyshev de second ordre a été motivé par son excellent compromis entre la réponse en fréquence et la régularité de la phase dans les applications audio. Bien que ce type de filtre introduise une petite ondulation dans la bande passante, il permet d'obtenir des caractéristiques de coupure plus nettes et d'atteindre un niveau d'atténuation suffisant dans la bande de transition, ce qui est essentiel pour le traitement des signaux audio dans notre projet.



2. Régulation du gain du signal

Le signal provenant des sources telles que les instruments de musique n'est pas suffisamment amplifié pour être utilisé avec un convertisseur analogique-numérique (ADC), et pour exploiter pleinement la plage de mesure de ce dernier, il doit être adapté. Nous allons donc utiliser un amplificateur de tension pour adapter ce signal à l'entrée de l'ADC.

Les sources musicales ont une amplitude variante entre 50 mVpp et 1,65 Vpp, mais nous souhaitons obtenir un signal de sortie maximum de 0 à 3,3 V (soit un signal équivalent de 1,65 Vpp). Pour cela, un gain variable est nécessaire :

- Gain de 1 (pas d'amplification) si le signal d'entrée est de 1,65 Vpp,
- Gain de 33 si le signal d'entrée est de 50 mVpp, ce qui permettra d'obtenir 1,65 Vpp en sortie.

Pour obtenir ce gain réglable, nous allons utiliser un potentiomètre afin de permettre un ajustement manuel du gain en fonction des besoins.

2.1. Montage de l'amplificateur

Le circuit d'amplification sera constitué de deux amplificateurs :

- Un amplificateur inverseur avec un gain variable allant de 0 à 33.
- Un second amplificateur inverseur avec un gain de 1.

Le second amplificateur est utilisé pour conserver la phase entre l'entrée et la sortie du système. Le premier amplificateur permet de régler le gain en fonction de l'amplitude du signal d'entrée, et le second amplificateur rétablit la phase tout en maintenant un gain unitaire pour éviter toute inversion de phase indésirable dans le signal amplifié.

Voici le montage d'un amplificateur inverseur :

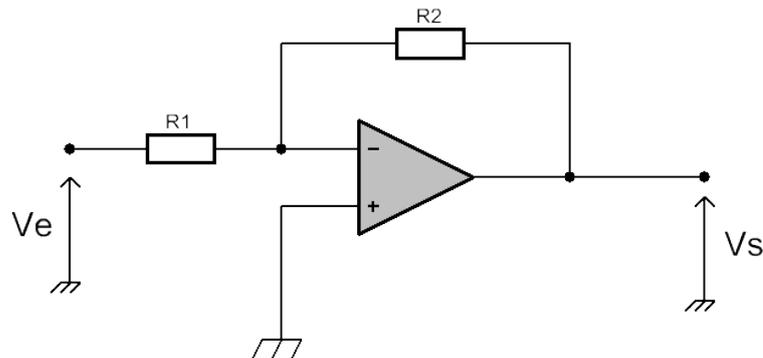


Figure 17 Montage Amplificateur Inverseur

Le gain d'un amplificateur inverseur est de :

$$\frac{V_e}{V_s} = G = -\frac{R_2}{R_1}$$

En mettant en série deux montages inverseurs, nous pouvons éliminer le signe négatif. L'avantage de cette configuration, comparée à d'autres comme le montage non-inverseur, est qu'elle permet d'obtenir un gain minimum de 0.

Voici le schéma de notre système :

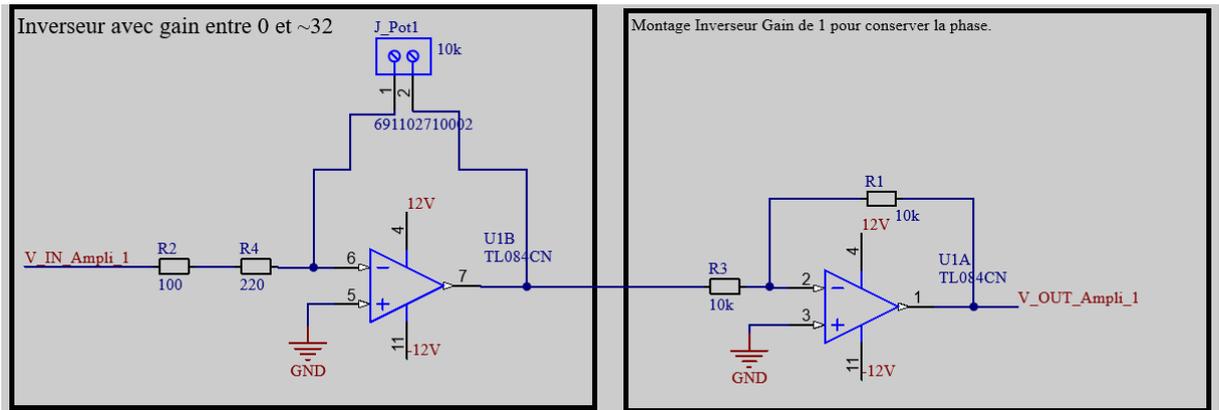


Figure 18 Montage 2 régulation du gain du signal

Un potentiomètre est utilisé dans le premier montage, permettant de régler le gain.

3. Application d'un décalage sur le signal analogique

Le signal audio provenant des instruments de musique est un signal alternatif (AC), ce qui signifie qu'il comporte des parties négatives. Or, ces signaux ne sont pas adaptés aux circuits numériques, tels que les convertisseurs analogique-numérique (CAN), les microcontrôleurs et d'autres composants numériques, qui pourraient être endommagés par des tensions négatives.

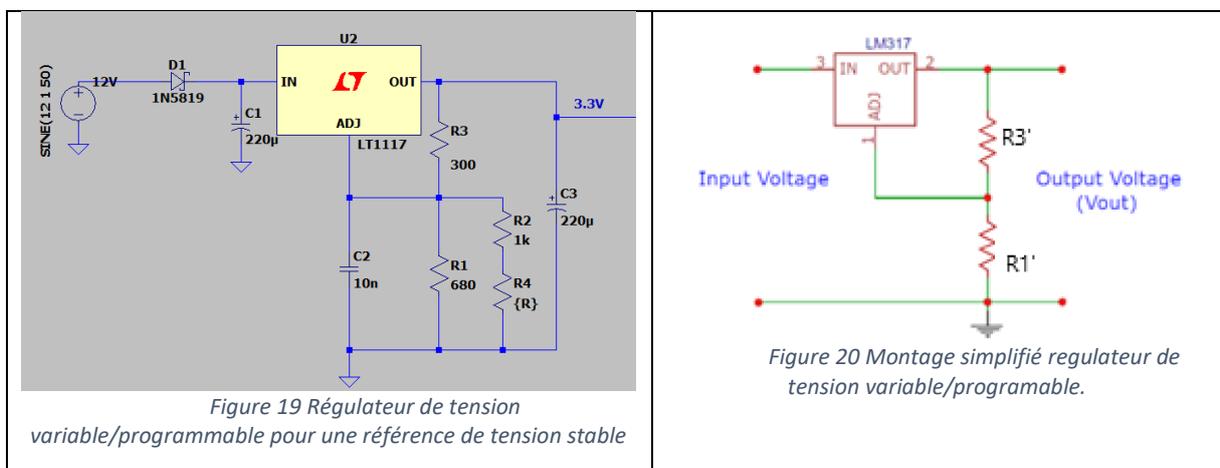
Pour éviter cela, nous devons ajouter une composante continue (DC) à notre signal, également appelée offset ou décalage DC. Cet ajout permet de décaler le signal vers des valeurs strictement positives, le rendant compatible avec les circuits numériques tout en préservant son intégrité.

3.1. Source de tension stable et fixe

L'ajout d'un décalage DC nécessite une tension de référence stable et fixe pour assurer un décalage constant du signal, sans fluctuations indésirables.

Nous utilisons un régulateur de tension pour générer cette tension de référence. Comparé à d'autres solutions, comme une référence basée sur une diode Zener ou des circuits similaires, le régulateur de tension offre une meilleure stabilité, une précision accrue, et réduit la sensibilité aux variations de charge et de température. En effet, les diodes Zener peuvent être sujettes à des variations de tension en fonction du courant qui les traverse, ce qui peut entraîner une instabilité de l'offset.

Le montage suivant utilise un potentiomètre multi tour de précision, permettant d'ajuster précisément la tension de référence et donc l'offset appliqué au signal.



Pour que la référence de tension soit réglable, la tension en sortie de montage d'un régulateur de tension de variable se définit par la relation suivante :

$$V_{out} = 1.25 * \left(1 + \frac{R1'}{R3'} \right)$$

Dans le montage simplifié (Figure 20), la valeur de R3' est fixée, tandis que R1' est réglable, ce qui permet un ajustement précis du circuit. Cependant, si R1' était un potentiomètre unique, Vout pourrait descendre jusqu'à 1.25V, ce qui n'est pas idéal pour un réglage de précision.

Pour pallier ce problème, le montage présenté en Figure 19 propose une solution améliorée. Il utilise une résistance fixe R2 en parallèle avec un réseau de deux résistances en série (R2 + R4), où R4 est un potentiomètre. Cette configuration permet de régler la tension dans une plage définie autour de 3.3V, évitant ainsi d'atteindre des valeurs trop basses qui compromettraient la précision du réglage.

Cela nous donne l'expression suivante :

$$V_{out} = 1.25 * \left(1 + \frac{R1' = \frac{R1(R2 + R4)}{R1 + (R2 + R4)}}{R3 = R3'} \right)$$

3.2. Montage Offset

Pour ajouter un offset au signal audio, nous utilisons un amplificateur différentiel, qui permet d'ajouter la tension de référence à notre signal tout en maintenant une bonne linéarité et une faible distorsion. Ce montage garantit que le décalage DC est appliqué de manière uniforme, sans altérer les caractéristiques du signal d'origine.

Principe du montage différentiel :

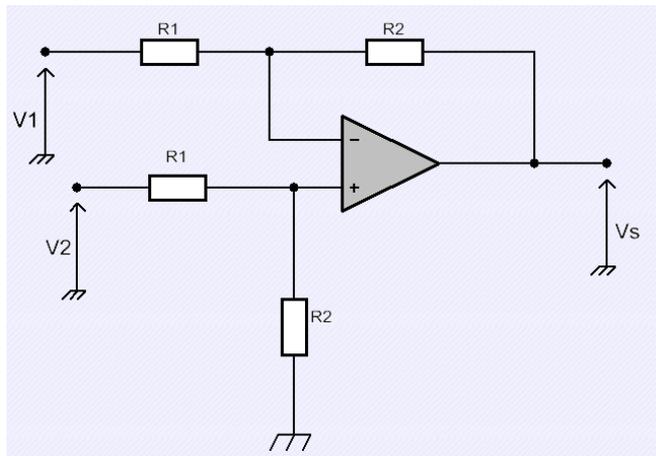


Figure 21 Montage amplificateur différentiel

L'amplificateur opérationnel (AOP) est supposé idéal et fonctionne en régime linéaire, ce qui signifie que ses entrées sont à des tensions égales :

$$V_+ = V_-$$

La tension V_+ est obtenue via un pont diviseur de tension, tandis que V_- est déterminée par le **théorème de Millman** :

$$V_+ = V_2 \left(\frac{R_2}{R_1 + R_2} \right)$$

$$V_- = \frac{\frac{V_1}{R_1} + \frac{V_s}{R_2}}{\frac{1}{R_1} + \frac{1}{R_2}} = \frac{V_1 R_2 + V_s R_1}{R_1 + R_2}$$

or, puisque $V_+ = V_- = \frac{V_2 R_2}{R_1 + R_2} = \frac{V_1 R_2 + V_s R_1}{R_1 + R_2}$

$$\Leftrightarrow \frac{V_2 R_2}{R_1 + R_2} = (V_2 - V_1) \frac{R_2}{R_1 + R_2}$$

$$\Leftrightarrow V_s = \frac{R_2}{R_1} (V_2 - V_1)$$

En choisissant $R_1 = R_2$, nous obtenons l'expression suivante :

$$V_s = V_2 - V_1$$

3.3. Justification et mise en œuvre

Nous avons choisi V_2 comme tension de référence, car nous souhaitons ajouter un offset positif au signal. Cela signifie que le signal d'entrée doit être appliqué sur V_1 (entrée inverseuse).

Ce choix entraîne une inversion de phase du signal, ce qui n'est pas problématique en soi. Si nécessaire, un montage inverseur en amont permettrait de rétablir la phase d'origine.

Pour notre application, nous devons ajouter un offset de 1.65V à un signal audio de 1.63Vpp, afin d'obtenir un signal compatible avec un système numérique en 0 - 3.3V.

En choisissant $R_1 = R_2$, la tension sur l'entrée non inverseuse V_+ est divisée par 2, ce qui signifie que pour obtenir un décalage de 1.65V, il faut appliquer une tension de $1.65\text{ V} / 2 = 825\text{ mV}$ sur l'entrée V_+ .

Le pont diviseur de tension doit donc réduire la tension de référence de 3.3V à 825mV, soit :

$$\frac{3.3\text{V}}{4} = 825\text{mV}$$

Le montage réalisé suit donc la fonction :

$$V_s = -V_{in} + 1.65$$

Dans la simulation suivante, nous pouvons observer que l'offset est bien appliqué à notre signal, bien que celui-ci soit inversé.

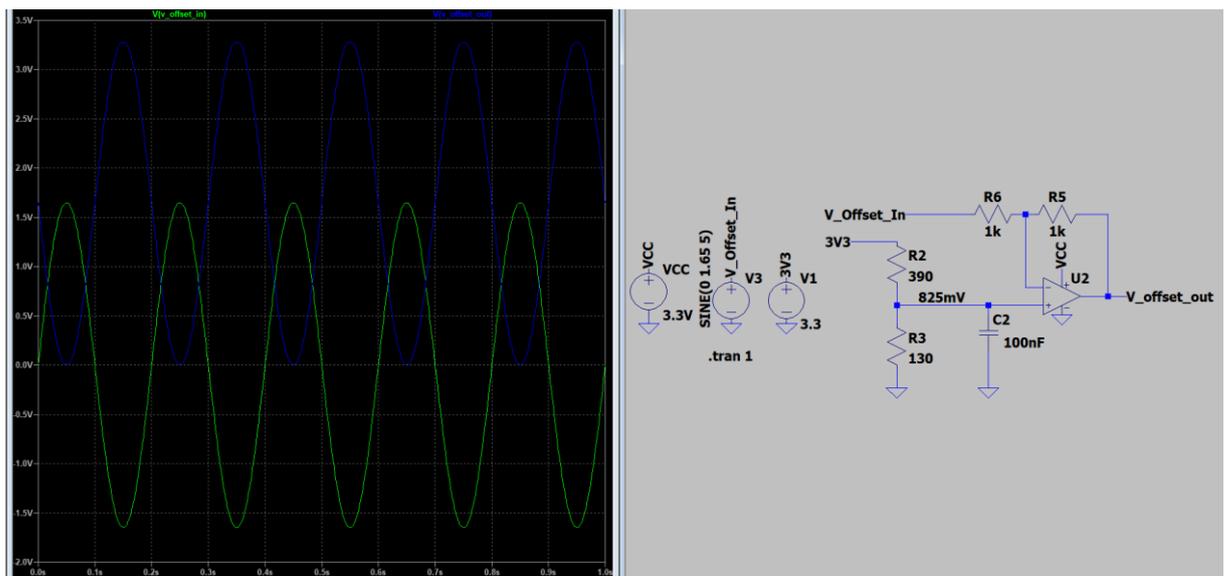


Figure 22 Simulation Offset avec montage différentiel

4. Conversion du signal analogique en numérique

Après avoir conditionné le signal analogique, il est nécessaire de le numériser à l'aide d'un convertisseur analogique-numérique (ADC).

Contrairement à la plupart des microcontrôleurs, notre micro-ordinateur ne possède pas d'ADC intégré, nous devons donc utiliser un convertisseur externe.

Pour un signal audio, la qualité minimale recommandée est de 16 bits avec une fréquence d'échantillonnage de 44,1 kHz, soit un débit de 705 kbps sans compression. La fréquence maximale audible par l'oreille humaine est d'environ 20 kHz. Selon le théorème de Nyquist, la fréquence d'échantillonnage doit être au moins deux fois supérieure pour éviter la perte d'information. En choisissant 44,1 kHz, nous respectons cette condition tout en optimisant la taille des données à traiter.

Représentation de numérisation d'un signal et impact selon la fréquence d'échantillonnage :

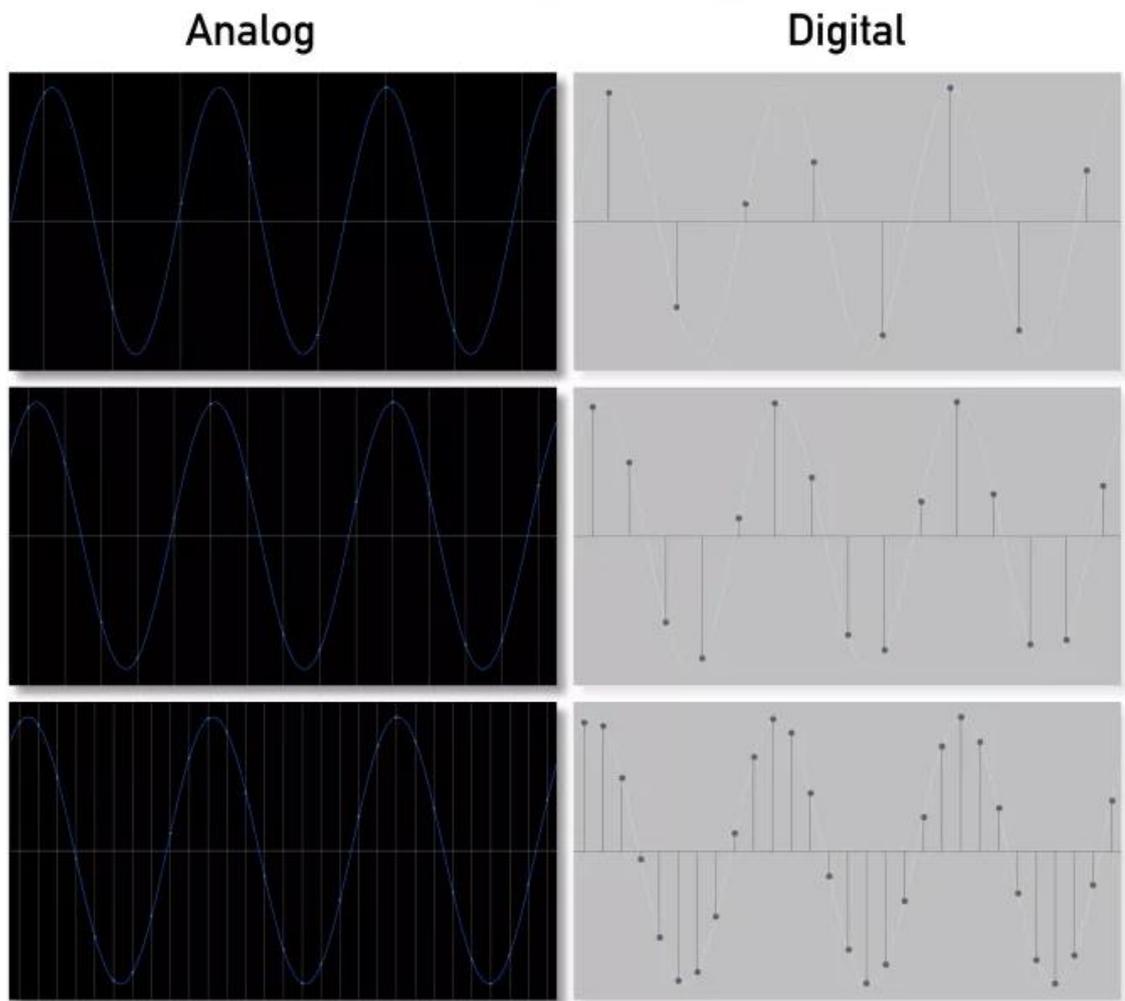


Figure 23 Numérisation d'un signal et impact de fréquence d'échantillonnage

Voici un tableau présentant les différentes qualités standards :

Sound Quality Required	Bandwidth	Sampling rate	Number of bits	Data rate (bits/sec)	Comments
High fidelity music (compact disc)	5 Hz to 20 kHz	44.1 kHz	16 bit	706k	Satisfies even the most picky audiophile. Better than human hearing.
Telephone quality speech (with companding)	200 Hz to 3.2 kHz	8 kHz	12 bit	96k	Good speech quality, but very poor for music.
	200 Hz to 3.2 kHz	8 kHz	8 bit	64k	Nonlinear ADC reduces the data rate by 50%. A very common technique.
Speech encoded by Linear Predictive Coding	200 Hz to 3.2 kHz	8 kHz	12 bit	4k	DSP speech compression technique. Very low data rates, poor voice quality.

Figure 24 Tableau présentant quelques exemples des standard qualité audio

Ce choix correspond à une qualité "High Fidelity" (44,1 kHz, 16 bits). Toutefois, une qualité inférieure avec une fréquence d'échantillonnage et une résolution réduites reste envisageable si la priorité est de minimiser la quantité de données.

4.1. Choix du Convertisseur Analogique-Numérique

Nous avons sélectionné deux ADC répondant à ces critères et communiquant via SPI :

- ADS8331IPW : ADC 16 bits, un seul canal, basé sur une architecture SAR (Successive Approximation Register). Il est optimisé pour des mesures rapides et précises avec une faible consommation d'énergie.
- MCP3564 : ADC 24 bits, jusqu'à 8 canaux, utilisant une architecture $\Delta\Sigma$ (Delta-Sigma). Il offre une meilleure résolution et une faible distorsion, idéal pour des applications nécessitant une grande précision.

Le choix entre ces deux ADC dépend du nombre de signaux à traiter ainsi que des contraintes de vitesse et de précision du système.

C. Gestion de l'entrée audio numérique

Le MIDI (Musical Instrument Digital Interface) intervient dans le contrôle des instruments de musique tels que les échantillonneurs ou synthétiseurs dédiés ou intégrés aux ordinateurs sous forme de cartes. Le MIDI permet également de contrôler les consoles de mixage, des processeurs d'effets et des systèmes d'enregistrement.

Le MIDI est une interface de contrôle sérielle originellement dédiée aux systèmes musicaux. Mais il a rapidement débordé de sa vocation première et intervient aujourd'hui dans le contrôle d'un large éventail de matériel non seulement audio, mais destinés à l'éclairage du spectacle.

Le protocole MIDI intègre de nombreux paramètres de commande, la majorité d'entre eux étant librement affectables par l'utilisateur en fonction de ses besoins. Il est possible de contrôler des instruments de musique polyphonique en temps dit pseudo-réel : les temps de transmission ne sont pas perceptibles dans la majorité des cas. Il est également possible d'adresser plusieurs appareils différents sur le même flot de données MIDI.

Le MIDI est un moyen économique de contrôler et stocker des informations sonores. En comparaison avec de l'audio numérique échantillonné (16 bits/44100 Hz) qui peut requérir un espace de stockage très important, les informations MIDI pour une minute de musique peuvent tout à fait ne pas excéder 5 ko. De même la charge de travail imposée à un ordinateur est beaucoup plus faible avec du MIDI qu'avec de l'audio.

1.1. *Connectique et interface MIDI*

Historiquement, le MIDI utilise un connecteur DIN 5 broches, mais depuis MIDI 2.0, une alternative est le mini-jack TRS 2,5 mm, plus compact et couramment utilisé sur les appareils récents. L'adaptation entre DIN-5 et TRS est possible avec un câblage approprié.

Afin d'utiliser un ordinateur comme contrôleur central dans un système MIDI, il faut disposer d'une interface MIDI. Chaque interface MIDI possède au moins un port IN et un port OUT. Le port THRU n'est pas toujours nécessaire.

1.2. *Transport des données MIDI*

Le standard MIDI spécifie une interface sérielle unidirectionnelle fonctionnant à 31.24 kbits (à + ou - 1 %). Ce taux a été fixé à l'époque où la vitesse des microprocesseurs était beaucoup plus basse qu'aujourd'hui : ce taux était une division pratique d'un taux d'horloge maître type de 1 ou 2 MHz. Le taux devait de plus être suffisamment bas pour assurer une transmission sans perte sur des câbles et interfaces simples, tout en autorisant un délai de transmission raisonnable entre instruments.

La transmission des données est unidirectionnelle entre le transmetteur et le récepteur. Le retour requiert une liaison MIDI séparée.

Les messages de contrôle sont transmis par groupes d'octets précédés par un bit de début et suivis par un bit de fin à chaque octet afin de synchroniser la réception des données, la transmission étant asynchrone.

L'addition des bits de début et de fin indique que chaque mot de 8 bits occupe en fait une période de 10 bits à la transmission (pour une durée totale de 320 microsecondes). Un message MIDI typique consiste en un, deux ou trois octets. Certains messages spéciaux peuvent toutefois être plus longs.

1.3. Canaux MIDI

Les messages MIDI sont constitués d'un certain nombre d'octets. Chaque partie du message a une vocation spécifique, l'une d'entre elles étant de définir le canal de réception auquel le message est destiné. De cette façon, un appareil de contrôle peut adresser ses données à un appareil de réception de manière spécifique.

Un appareil réglé pour recevoir un ou plusieurs canaux donnés réagit uniquement aux informations qui sont adressées à ces canaux et ignorent les autres.

A la base, il existe 16 canaux MIDI et les instruments peuvent être réglés en vue de recevoir spécifiquement sur un ou plusieurs de ces canaux, ou sur tous les canaux à la fois.

La limite de 16 canaux peut être dépassée en utilisant une interface multiport connectée à un ordinateur. Chacun des ports physique de l'interface peut alors prendre en charge 16 canaux MIDI.

1.4. Structure des messages MIDI

A la base, il existe deux types d'octets de message MIDI : l'octet de statut et l'octet de données.

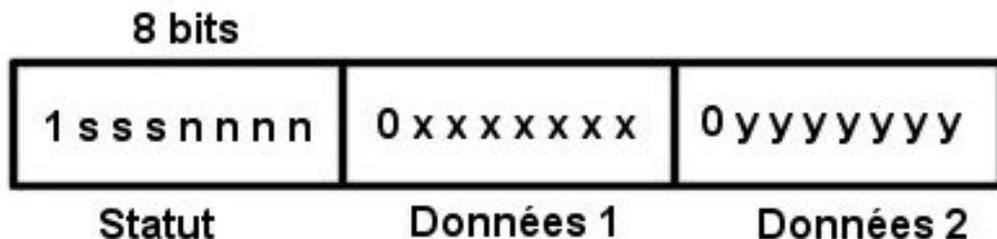
Sur les huit bits de ces octets, seuls sept sont actifs car le premier bit de chaque octet est réservé à la qualification du type (statut ou données) ce qui offre donc 2 puissance 7 (128) valeurs possibles.

Les octets de statut débutent toujours par un 0.

La première moitié (3 bits) de l'octet de statut indique le type de message. Cela donne donc 8 types de messages possibles. Il existe cependant des cas particuliers avec les message systèmes.

La seconde moitié le numéro de canal (4 bits) est attribuée à la définition du numéro de canal auquel le message est adressé. Cela offre 2 puissance 4 (16) canaux possibles.

Format général de message MIDI



Bits "sss" : type de message

Bits "nnnn" : numéro de canal

Bits "xxxxxxx" et "yyyyyyy" : données du message.

Figure 25 Trame MIDI

Les messages MIDI standards peuvent avoir une longueur de 3 octets mais ce n'est pas le cas général, et certains cas assez communs font exception à la règle.

Le tableau suivant résume le format et le contenu des principaux types de message MIDI.

Résumé des messages MIDI			
Message	Statut	Données 1	Données 2
<i>Messages de canal</i>			
Note off	&8n	Numéro de note	Vélocité
Note on	&9n	Numéro de note	Vélocité
Aftertouch polyphonique	&An	Numéro de note	Pression
Changement de contrôle	&Bn	Numéro de contrôleur	Données
Changement de programme	&Cn	Numéro du programme	.
Aftertouch de canal	&Dn	Pression	.
Roue de hauteur	&En	LSbyte (octet de poids faible)	MSbyte (octet de poids fort)
<i>Système exclusif</i>			
Début	&F0	ID de constructeur	Octets de données
Fin	&F7	.	.
<i>Système commun</i>			
Quart de trame (MTC)	&F1	Données	.
Pointeur du morceau	&F2	LSbyte	MSbyte
Sélection du morceau	&F3	Numéro du morceau	.
Ordre d'accordage	&F6	.	.
<i>Système temps réel</i>			
Horloge	&F8	.	.
Début	&FA	.	.
Continuer	&FB	.	.
Stop	&FC	.	.
Senseur actif	&FE	.	.
Réinitialisation	&FF	.	.

Figure 26 Format et le contenu des principaux types de message MIDI.

Le tableau des notes dans un message MIDI :

Octave (-2 à 8 ou -1 à 9)	Do	Do#	Re	Re#	Mi	Fa	Fa#	Sol	Sol#	La	La#	Si
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-2 (ou -1)	0	1	2	3	4	5	6	7	8	9	10	11
-1 (ou 0)	12	13	14	15	16	17	18	19	20	21	22	23
0 (ou 1)	24	25	26	27	28	29	30	31	32	33	34	35
1 (ou 2)	36	37	38	39	40	41	42	43	44	45	46	47
2 (ou 3)	48	49	50	51	52	53	54	55	56	57	58	59
3 (ou 4)	60	61	62	63	64	65	66	67	68	69	70	71
4 (ou 5)	72	73	74	75	76	77	78	79	80	81	82	83
5 (ou 6)	84	85	86	87	88	89	90	91	92	93	94	95
6 (ou 7)	96	97	98	99	100	101	102	103	104	105	106	107
7 (ou 8)	108	109	110	111	112	113	114	115	116	117	118	119
8 (ou 9)	120	121	122	123	124	125	126	127	-	-	-	-

Figure 27 Notes MIDI

2. Synchronisation des horloges MIDI avec le traitement audio

La synchronisation des horloges MIDI avec le traitement audio est un aspect crucial pour assurer que les événements MIDI (comme le déclenchement de notes ou de changements de contrôleurs) se produisent au moment exact dans le traitement audio.

Le MIDI TimeCode (MTC) est très exploité comme moyen de synchroniser des équipements contrôlés par MIDI avec une véritable référence temporelle. Les interfaces MIDI multiports disposent généralement d'un port de timecode SMPTE/EBU qui permet de lire et d'écrire du timecode longitudinal (LTC). Le LTC est converti en MTC et peut ainsi être géré par un logiciel MIDI.

Dans une trame de timecode LTC, deux groupes de données binaires sont alloués aux heures, minutes, secondes et images, ces groupes représentant respectivement les dizaines et les unités de chaque section, ce qui donne au total huit groupes binaires. Ce format doit être converti en un format compatible avec les autres données MIDI (octet de statut suivi par les octets de données).

Il existe deux types de message de synchronisation MTC :

Le premier (full-frame) met à jour le récepteur régulièrement. Il est transmis sous la forme d'un message en système exclusif universel en temps réel.

Le second (quarter-frame) met à jour la position du timecode en une fois, dans des situations comme l'avance rapide. Il est dénoté par l'octet de statut (&1F).

Comme une trame de timecode comporte trop d'informations pour pouvoir être transmise sous la forme standard du message MIDI à 3 octets. Elle est donc tronçonnée en 8 messages séparées (quarter-frame). Quatre de ces messages sont transmis à chaque trame de timecode afin de limiter le taux de transmission. Chaque message du groupe de huit représente une partie de la valeur du timecode et prend la forme générale : & [F1] [Données].

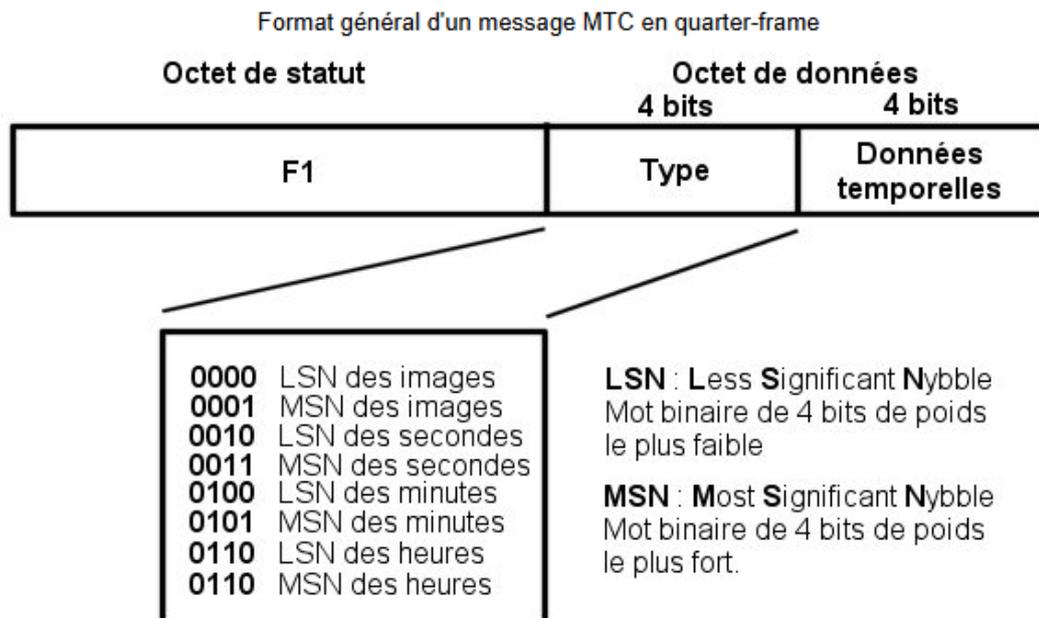


Figure 28 Format message MIDI Timecode

3. Conversion des données MIDI en signal audio

La conversion des données MIDI en un signal audio réel est un processus clé qui permet de transformer les informations de contrôle MIDI en une sortie sonore. MIDI étant un protocole de données numériques ne générant pas directement de son, il faut un moteur de synthèse pour traduire ces données en un signal audio analogique ou numérique. Ce processus est réalisé grâce à un moteur de synthèse audio qui peut interpréter les messages MIDI reçus et générer un signal sonore en conséquence.

Les données MIDI, telles que les événements de note, les contrôleurs et les messages de changement de programme, sont utilisées pour activer des instruments virtuels ou des échantillons audio. Cela se fait généralement par deux approches principales : la synthèse par échantillonnage et la synthèse par modélisation physique. La synthèse par échantillonnage consiste à jouer des enregistrements préenregistrés de sons instrumentaux, alors que la synthèse par modélisation physique génère des sons à partir de modèles mathématiques des instruments.

Le signal audio généré peut être au format .wav, qui est un format standard pour les fichiers audio non compressés, ou un autre format approprié selon les besoins du système (par exemple, .aiff ou .flac pour la gestion d'audio compressé ou de haute qualité). Une fois les données MIDI converties en audio, ces signaux sont traités par le système de mixage, où ils peuvent être ajustés, modifiés et fusionnés avec d'autres signaux audio dans un environnement en temps réel.

La conversion MIDI vers audio dans ce projet est un élément essentiel pour permettre à l'Orchestrion de recréer des performances musicales en combinant les divers instruments MIDI et en les intégrant dans une composition sonore cohérente. Cela ouvre également la possibilité d'utiliser des effets audio comme la réverbération, la modulation ou le filtrage, afin d'ajuster le son final généré par les signaux MIDI.

D. Traitement numérique du signal audio

1. Application de filtres et algorithmes de mixage numérique
2. Gestion du mixage audio en temps réel
3. Fournir un IHM pour la gestion des paramètres de mixage et effets.
4. Restitution du signal audio traité
5. Sortie stéréo ou ligne pour restitution sonore

E. Amplification du signal audio

1. Amplification en tension et en courant pour restitution optimale

F. Gestion de l'alimentation électrique

1. Alimentation via batterie rechargeable
2. Recharge via adaptateur secteur ou alimentation DC

G. Format compact et ergonomie

1. Conception d'un boîtier robuste et léger
2. Optimisation des dimensions pour transport et intégration

Rédaction dans le dossier de sous-système

V. Développement en cours et prochaines étapes

Le développement du projet est encore en cours, et certaines parties, notamment la gestion du signal MIDI, sont en phase de développement. Actuellement, l'implémentation des interfaces MIDI sur le Raspberry Pi est en train d'être finalisée, avec une attention particulière portée à la synchronisation des horloges MIDI et à la conversion des données MIDI en signaux audio. En raison de la complexité de la gestion du MIDI Timecode et des diverses stratégies de synchronisation audio, cette section est en cours de validation et de tests. Des informations supplémentaires concernant ces développements seront intégrées dans la prochaine version du dossier de conception, qui présentera l'avancement de ces fonctionnalités et leurs intégrations finales.

De plus, la conversion analogique-numérique (ADC) fait l'objet de tests et d'optimisations. Actuellement, nous travaillons sur le choix du convertisseur adapté aux besoins spécifiques du projet, avec un accent particulier sur la résolution et la fréquence d'échantillonnage pour garantir une restitution sonore fidèle et sans artefacts. Le système de traitement des entrées analogiques sera précisé et détaillé dans la mise à jour suivante du dossier, qui fournira une vue d'ensemble complète des choix techniques, des schémas de connexion et des résultats des tests de validation du système ADC.

Les versions futures du dossier de conception fourniront des informations supplémentaires sur la validation des performances, les tests unitaires et les améliorations logicielles en cours de développement. Elles incluront également des détails sur les aspects techniques associés aux algorithmes de traitement audio et sur l'intégration du système dans l'environnement Orchestrion.

ANNEX

Simulations :

Les Fichiers de simulations seront disponibles sur l'espace d'hébergement git (github et gitlab).