

# Table des matières

<b>Introduction générale</b>	<b>1</b>
<b>1 Présentation de l'entreprise et cadre du projet</b>	<b>3</b>
1.1 Organisme d'accueil . . . . .	3
1.1.1 Aperçu général . . . . .	3
1.1.2 Historique . . . . .	4
1.1.3 Les tendances stratégiques de l'entreprise . . . . .	5
1.1.4 Diverses clientèles . . . . .	6
1.1.5 Site de Tunis . . . . .	6
1.2 Cadre du projet . . . . .	7
1.2.1 Contexte générale . . . . .	8
1.2.2 Problématique . . . . .	8
1.2.3 Solution Proposée . . . . .	9
<b>2 Introduction aux Microcontrôleurs STM32 et à l'Architecture STM32Cube</b>	<b>11</b>
2.1 Généralités sur les microcontrôleurs . . . . .	11
2.1.1 Familles des Microcontrôleurs STM32 . . . . .	12
2.2 Périphériques Spécifiques . . . . .	14
2.2.1 SAI (Serial Audio Interface) . . . . .	15
2.2.2 DFSDM (Digital Filter for Sigma-Delta Modulators) . . . . .	15
2.3 Architecture STM32Cube . . . . .	16

2.3.1	Niveaux de l'Architecture . . . . .	16
2.3.2	Board Support Package (BSP) . . . . .	17
<b>3</b>	<b>Implémentation d'une Démonstration Audio sur la Carte</b>	
	<b>STM32H735G-DK</b>	<b>20</b>
3.1	Carte de développement STM32H735G-DK . . . . .	20
3.2	Fonctionnalités Audio de la STM32H735G-DK . . . . .	21
3.2.1	codec audio WM8994 . . . . .	21
3.2.2	Enregistrement Audio numérique . . . . .	22
3.2.3	Enregistrement Audio Analogique . . . . .	24
3.2.4	Comparaison entre l'enregistrement audio analogique et numérique	25
3.2.5	Fonctions BSP Audio IN pour l'enregistrement numérique et analogique . . . . .	26
3.2.6	Lecture Audio . . . . .	27
3.2.7	Fonctions BSP Audio OUT pour la lecture Audio . . . . .	28
3.2.8	Démonstrations BSP . . . . .	29
3.2.9	Fonction d'enregistrement audio analogique . . . . .	30
	<b>Conclusion générale</b>	<b>32</b>

# Table des figures

1.1	STMicroelectronics diffusion mondiale . . . . .	5
1.2	La gamme de produits de STMicroelectronics . . . . .	6
1.3	STMicroelectronics Tunis . . . . .	7
2.1	STM32 MCU sous-familles . . . . .	13
2.2	Structure de base d'un MCU STM32 . . . . .	14
2.3	Schéma du SAI . . . . .	15
2.4	Schéma du DFSDM . . . . .	16
2.5	Architecture des packages STM32Cube . . . . .	17
2.6	Organisation des pilotes BSP . . . . .	18
3.1	Carte de développement STM32H735G-DK . . . . .	21
3.2	Codec audio WM8994 . . . . .	22
3.3	Exemple d'acquisition de son dans une application audio . . . . .	23
3.4	Microphone numérique IMP34DT05 . . . . .	23
3.5	Entrée jack pour l'enregistrement . . . . .	24
3.6	Sortie jack pour la lecture . . . . .	27
3.7	Organigramme des exemples de démonstrations BSP . . . . .	29
3.8	Organigramme de la fonction d'enregistrement audio analogique . . . . .	31

# Liste des tableaux

3.1	Comparaison entre l'enregistrement audio analogique et numérique . . . .	25
3.2	BSP AUDIO IN class APIs . . . . .	26
3.3	BSP AUDIO IN class APIs . . . . .	28

# Introduction générale

Les systèmes embarqués sont au cœur de nombreuses technologies modernes, intégrant des logiciels et du matériel pour réaliser des fonctions spécifiques au sein d'un dispositif. De l'électronique grand public aux applications industrielles, ces systèmes jouent un rôle crucial dans l'optimisation des performances, la réduction des coûts et l'amélioration de l'efficacité énergétique. Grâce à leur flexibilité et leur capacité d'adaptation, les systèmes embarqués permettent de répondre à des besoins variés dans des domaines tels que l'automobile, la santé, et l'Internet des objets.

Dans ce contexte, STMicroelectronics se positionne comme un acteur clé dans le développement de solutions basées sur les microcontrôleurs STM32. L'entreprise, reconnue pour son innovation et sa recherche constante d'excellence, propose une large gamme de produits adaptés aux exigences des systèmes embarqués. Mon stage d'été chez STMicroelectronics m'a offert l'opportunité de m'immerger dans cet environnement dynamique et d'explorer les défis liés à l'intégration de ces technologies avancées.

Ce rapport se divise en trois chapitres. Le premier chapitre présente une vue d'ensemble des systèmes embarqués, en mettant en lumière leur architecture et leurs applications. Le deuxième chapitre se concentre sur STMicroelectronics, décrivant son rôle dans le secteur des semi-conducteurs et ses contributions significatives au développement des microcontrôleurs STM32. Enfin, le troisième chapitre aborde la mise en œuvre pratique d'une démonstration audio sur la carte STM32H735G-DK, illustrant les compétences techniques acquises tout au long de mon stage.

Cette expérience enrichissante m'a permis d'approfondir mes connaissances dans le

domaine des systèmes embarqués et d'appliquer mes compétences techniques dans un cadre industriel.

# Chapitre 1

## Présentation de l'entreprise et cadre du projet

### Introduction

Dans ce premier chapitre, je me concentrerai sur la présentation de l'entreprise d'accueil, STMicroelectronics, ainsi que sur le cadre général du projet que j'ai réalisé au cours de mon stage. Je commencerai par une vue d'ensemble de l'organisme d'accueil, en abordant son historique, ses tendances stratégiques et son importance dans le secteur des semi-conducteurs. Ensuite, je détaillerai le contexte du projet, en mettant en lumière les défis rencontrés, notamment dans le domaine de l'enregistrement audio embarqué, et la solution que j'ai proposée pour y répondre, en m'appuyant sur les technologies avancées de STMicroelectronics.

### 1.1 Organisme d'accueil

#### 1.1.1 Aperçu général

STMicroelectronics, communément appelé ST, est une société multinationale néerlandaise de technologie franco-italienne dont le siège est basé à Genève en Suisse.

Elle est spécialisée dans la conception et la fabrication des composants électroniques, particulièrement les microcontrôleurs, les circuits intégrés, les capteurs et les dispositifs de puissance. Sa puissance industrielle, son expertise en silicium, son portefeuille de propriétés intellectuelles et ses alliances stratégiques a fait d'elle un leader mondial dans la fabrication de semi-conducteurs. En effet, elle a été classée pendant plusieurs années parmi les 20 premiers fabricants de semi-conducteur depuis 1987 et occupait le rang 14 pour le classement fait en 2021. Les produits proposés par la société sont utilisés dans divers domaines d'applications, tel que l'automobile, les équipements industriels, les communications sans fil, les dispositifs médicaux et l'électronique dédié au grand public. D'autre part, en plus de concentrer ses efforts sur les solutions technologiques innovatrices et écologiques, STMicroelectronics veille à travailler sur les initiatives environnementales à l'instar de la réduction de l'empreinte carbone, la gestion d'énergie et la conception de produits électroniques durables.

### 1.1.2 Historique

STMicroelectronics est l'un des leaders mondiaux dans la conception, le développement et la fabrication des circuits intégrés et de dispositifs discrets à semi-conducteurs. Elle a été créée en 1987 par fusionnement de la société italienne SGS et la société française Thomson Semi-conducteurs sous le nom de SGS-Thomson, renommée STMicroelectronics en 1998 à la suite du retrait de Thomson du capital. La nouvelle société a poursuivi une stratégie de croissance accélérée dans l'objectif était de devenir un leader mondial de la microélectronique. Aujourd'hui, STMicroelectronics est classée parmi les 15 premiers fabricants mondiaux de semi-conducteurs et la première au niveau européen avec un chiffre d'affaires de 16.1 billions de milliards en 2022. Au total, ST possède environ 50 000 employés, dispose de 16 unités de recherche et développement avancées, 39 centres de conception, 14 sites de production et de 80 bureaux de vente dans 35 pays. STMicroelectronics est considérée parmi l'une des sociétés les plus innovantes grâce à un riche portefeuille d'environ 20 390 brevets à l'échelle mondiale.



Ces derniers appartiennent à 8 721 familles de brevets uniques. Sur les 20 390 brevets, 16 430 sont actifs . La figure suivante illustre la diffusion mondiale de ST et met en évidence son avantage d'être proche des consommateurs à travers le monde.



FIGURE 1.1 – STMicroelectronics diffusion mondiale

### 1.1.3 Les tendances stratégiques de l'entreprise

Les tendances stratégiques de STMicroelectronics se concentrent sur les domaines suivants :

- Automobile : les produits de ST sont dédiés pour une large gamme d'applications automobiles, tel que la gestion des moteurs, la sécurité et l'électronique embarquée.
- Industrie : La demande mondiale d'énergie est en hausse et il existe un besoin croissant de produits semi-conducteurs économes en énergie. Les produits de la société répondent à cette demande et sont exploités dans plusieurs applications industrielles telles que l'automatisation, l'énergie et le secteur médical.
- Internet des objets, IOT : L'IoT est un marché en plein expansion qui devrait générer des milliards de dollars de revenus dans les années à venir. STMicroelectronics est un fournisseur leader de produits semi-conducteurs pour ce marché et est bien positionnée pour bénéficier de cette croissance. Pour cela

l'entreprise développe une large gamme de produits recourus dans la domotique, les dispositifs portables et l'automatisation industrielle. STMicroelectronics est bien positionnée pour bénéficier des principales tendances stratégiques qui stimulent la croissance de l'industrie des semi-conducteurs. L'entreprise est un fournisseur leader de produits semi-conducteurs pour une large gamme de marchés et est engagée dans l'innovation et la durabilité. La figure 1.2 expose la gamme de produits de STMicroelectronics.

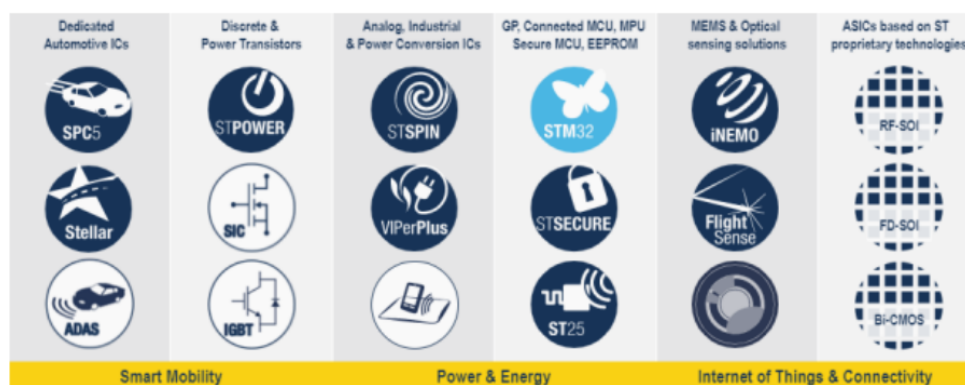


FIGURE 1.2 – La gamme de produits de STMicroelectronics

### 1.1.4 Diverses clientèles

La responsabilité concernant la qualité et la diversité des produits ont permis à STMicroelectronics de servir plus de 100 000 clients de différents domaines. Les cinq principaux clients de ST dans le domaine automobile sont BMW, Bosch, Tesla, Honda, Toyota. Ceux du domaine industriel sont : ABB, Emerson Electric, Schneider Electric, Alstom, Tetra Pak. Et pour finir dans le domaine d'internet des objets on peut citer : Azon, Apple, IBM, Intel, Huawei.

### 1.1.5 Site de Tunis

Le site fait partie intégrante du réseau mondial de fabrication de STMicroelectronics et joue un rôle important dans la capacité de l'entreprise à répondre aux besoins de ses clients dans le monde entier. Il a été implanté en décembre 2001 avec 9 ingénieurs à la

cit  technologique de communication d'El Ghazela. Il est en expansion continue et compte actuellement 250 ing nieurs qui collaborent avec divers groupes.

Le site de Tunis est l'un des sites les plus diversifi s de l'entreprise en termes d'expertise et de comp tence. Refl tant son r le multifonctionnel et multidisciplinaire. Le site ST Tunis comporte deux divisions :

- Microcontroller Division (MCD).
- Digital Information Technology (DIT).



FIGURE 1.3 – STMicroelectronics Tunis

## 1.2 Cadre du projet

"Dans cette section, je vais commencer par  tudier le contexte g n ral du projet afin de mieux comprendre son environnement. Ensuite, j'analyserai la probl matique rencontr e, en soulignant les principaux d fis techniques. Enfin, je proposerai une solution adapt e, en d taillant les approches et les choix technologiques que j'ai adopt s pour relever ces d fis."

### 1.2.1 Contexte générale

Dans le domaine des systèmes embarqués, l'audio est un élément clé dans de nombreuses applications, telles que les appareils portables, les systèmes de communication et les produits électroniques grand public. Ces systèmes nécessitent une gestion précise de l'acquisition, du traitement et de la restitution des signaux audio, qu'ils soient analogiques ou numériques.

L'enregistrement audio analogique, bien que traditionnel, reste pertinent, car il permet l'intégration de sources audio classiques comme les microphones analogiques ou les instruments de musique. Cependant, l'implémentation d'une solution d'enregistrement analogique sur des microcontrôleurs modernes, comme ceux de la série STM32, présente plusieurs défis techniques. Parmi ces défis, on trouve la conversion des signaux analogiques en numériques, la gestion des ressources matérielles limitées, et le traitement audio en temps réel.

Le projet vise à utiliser la carte de développement STM32H735G-DK, équipée du codec audio WM8994, pour concevoir une solution d'enregistrement audio analogique. Cette carte offre des interfaces audio robustes et des capacités de traitement en temps réel, adaptées aux applications audio embarquées. Ainsi, l'objectif est de développer un exemple fonctionnel d'enregistrement analogique qui pourra être utilisé dans divers contextes applicatifs.

### 1.2.2 Problématique

L'enregistrement audio analogique sur une plateforme embarquée telle que la STM32H735G-DK présente plusieurs défis techniques. D'une part, il est crucial de garantir une conversion efficace du signal analogique en numérique, tout en minimisant la perte de qualité sonore. Cette conversion nécessite une gestion précise de la synchronisation entre les interfaces du microcontrôleur, telles que le module SAI (Serial Audio Interface), et le codec WM8994, tout en tenant compte de la latence et des

ressources matérielles limitées.

D'autre part, il est nécessaire d'optimiser les performances pour assurer un traitement fluide des signaux audio en temps réel, en particulier sur des plateformes embarquées à faible consommation d'énergie. L'intégration des composants matériels doit également garantir la compatibilité et la stabilité du système, notamment entre le codec et le microcontrôleur STM32. Enfin, une analyse comparative avec l'enregistrement numérique est cruciale pour identifier les avantages et les limites de l'approche analogique, en fonction des exigences des différentes applications.

### 1.2.3 Solution Proposée

Pour relever les défis identifiés, la solution proposée se concentre sur l'utilisation optimisée des composants audio de la carte STM32H735G-DK, tout en assurant une gestion efficace des ressources matérielles.

#### **Configuration du codec WM8994 et de l'interface SAI :**

Le codec WM8994 sera configuré pour convertir les signaux analogiques en données numériques, tandis que l'interface SAI gèrera le flux audio avec une synchronisation optimale pour éviter toute latence. Le DMA (Direct Memory Access) sera utilisé pour transférer les données audio sans surcharger le processeur.

#### **Utilisation des pilotes BSP (Board Support Package) :**

Les pilotes BSP permettront une intégration simplifiée des interfaces matérielles, facilitant ainsi l'initialisation, la capture et la gestion des flux audio. Cela permettra de démarrer rapidement l'enregistrement et de gérer les fonctions essentielles comme la pause et l'arrêt.

### **Démonstration d'enregistrement audio analogique :**

Une démonstration pratique d'enregistrement audio analogique sera développée pour capturer des signaux via une entrée jack, les convertir, et les stocker ou les lire en temps réel. L'accent sera mis sur la qualité sonore et la minimisation des délais.

Cette solution permet d'implémenter une démonstration d'enregistrement audio analogique fonctionnelle, stable, et optimisée pour une plateforme embarquée telle que la STM32H735G-DK, tout en offrant une base pour des améliorations futures.

## **Conclusion**

Ce chapitre m'a permis de mettre en évidence le rôle de STMicroelectronics en tant qu'acteur majeur de l'industrie des semi-conducteurs, ainsi que l'importance de son site à Tunis dans la réalisation des objectifs de mon projet. J'ai présenté les défis techniques liés à l'intégration d'une solution d'enregistrement audio analogique sur une plateforme STM32 et détaillé la solution que j'ai élaborée. Cette première étape pose les bases pour les chapitres suivants, où j'explorerai plus en détail les microcontrôleurs STM32 et l'architecture utilisée dans la mise en œuvre pratique de mon projet.

# Chapitre 2

## Introduction aux Microcontrôleurs STM32 et à l'Architecture STM32Cube

### Introduction

Les microcontrôleurs STM32 sont une référence dans le domaine des systèmes embarqués en raison de leur flexibilité et de leurs performances. Ce chapitre examine en profondeur les composants essentiels des microcontrôleurs STM32, en détaillant leur architecture ainsi que les périphériques associés. Nous aborderons également l'architecture STM32Cube, un cadre logiciel modulaire qui facilite le développement d'applications sur les microcontrôleurs STM32.

### 2.1 Généralités sur les microcontrôleurs

Un microcontrôleur est un circuit intégré programmable qui combine les fonctionnalités d'un processeur, de la mémoire et des périphériques d'entrée/sortie sur une seule puce. Il est souvent qualifié de "système sur puce" (SoC) en raison de sa capacité à effectuer de

nombreuses tâches de contrôle en temps réel dans des systèmes embarqués [1].

Composants principaux d'un microcontrôleur :

**Unité centrale de traitement (CPU) :** C'est le cerveau du microcontrôleur. Il exécute les instructions du programme stocké en mémoire, effectue les calculs et gère les opérations logiques.

**Mémoire :**

- **Mémoire Flash :** Stocke le programme que le microcontrôleur doit exécuter.
- **Mémoire RAM :** Mémoire volatile utilisée pour stocker temporairement des données pendant l'exécution des programmes.

**Périphériques d'entrée/sortie (I/O) :** Ils permettent au microcontrôleur de communiquer avec le monde extérieur (capteurs, actionneurs, interfaces utilisateurs). Ces périphériques incluent des ports GPIO, des interfaces de communication (UART, SPI, I2C), et des convertisseurs analogiques-numériques (ADC).

**Horloges :** Des oscillateurs ou circuits de cadencement sont utilisés pour synchroniser les opérations à l'intérieur du microcontrôleur.

**Bus interne :** Les différents composants du microcontrôleur sont reliés entre eux via un bus interne, permettant le transfert de données entre la mémoire, le CPU et les périphériques.

### 2.1.1 Familles des Microcontrôleurs STM32

La famille STM32 de STMicroelectronics est une série de microcontrôleurs développés depuis 2006, cette famille est basée sur des processeurs d'architecture ARM 32 bits. Actuellement, STMicroelectronics est un leader mondial dans le domaine des microcontrôleurs, avec une large gamme de microcontrôleurs subdivisés en plus de dix sous-familles de produits distinctes dont chacune présente des caractéristiques spécifiques.

- **High Performance :** Idéal pour des systèmes embarqués avancés nécessitant une



puissance de calcul élevée, comme les STM32F7 et STM32H7 avec des fréquences allant jusqu'à 480 MHz.

- **Mainstream** : Microcontrôleurs équilibrés entre performance et coût, tels que les STM32F0 et STM32F3, adaptés aux applications de signalisation ou grand public.
- **Ultra-low-power** : Optimisés pour les dispositifs alimentés par batterie, comme les STM32L4 et STM32L5, offrant des performances énergétiques efficaces.
- **Wireless** : Incluent des fonctionnalités de connectivité sans fil, comme les STM32WB, adaptés aux applications IoT.

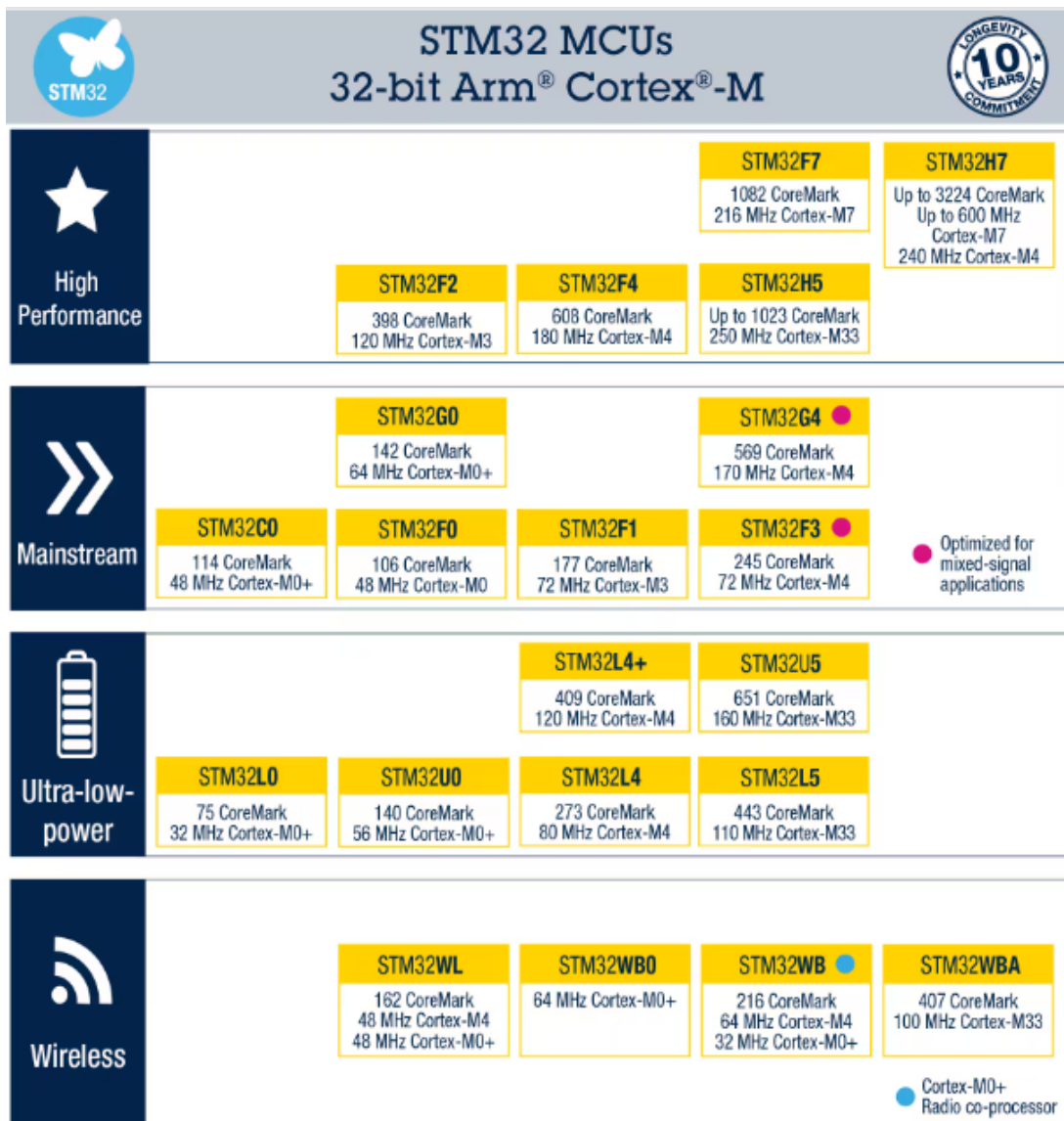


FIGURE 2.1 – STM32 MCU sous-familles

D'un point de vue fonctionnel, dans cette partie, je décris l'architecture système des STM32. La structure de base d'un microcontrôleur STM32 comprend principalement une unité centrale de traitement « CPU », des périphériques internes/externes, une mémoire pour stocker les programmes et les données, ainsi que des bus de communication, comme le montre la figure suivante.

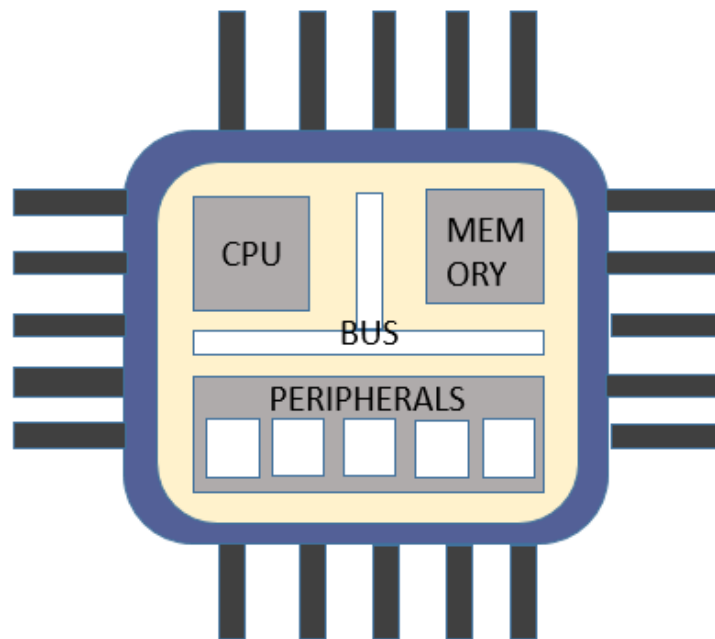


FIGURE 2.2 – Structure de base d'un MCU STM32

## 2.2 Périphériques Spécifiques

Dans mon projet, je me concentre sur certains périphériques spécifiques tels que le SAI (Serial Audio Interface) et le DFSDM (Digital Filter for Sigma-Delta Modulators). Ces périphériques jouent un rôle central dans le traitement et la transmission de données audio en série.

### 2.2.1 SAI (Serial Audio Interface)

Le SAI est utilisé pour transmettre et recevoir des données audio numériques en série. Il supporte plusieurs protocoles, notamment I2S et TDM, et offre une grande flexibilité en mode maître ou esclave. Il peut aussi fonctionner en mode synchrone pour optimiser l'utilisation des broches disponibles.

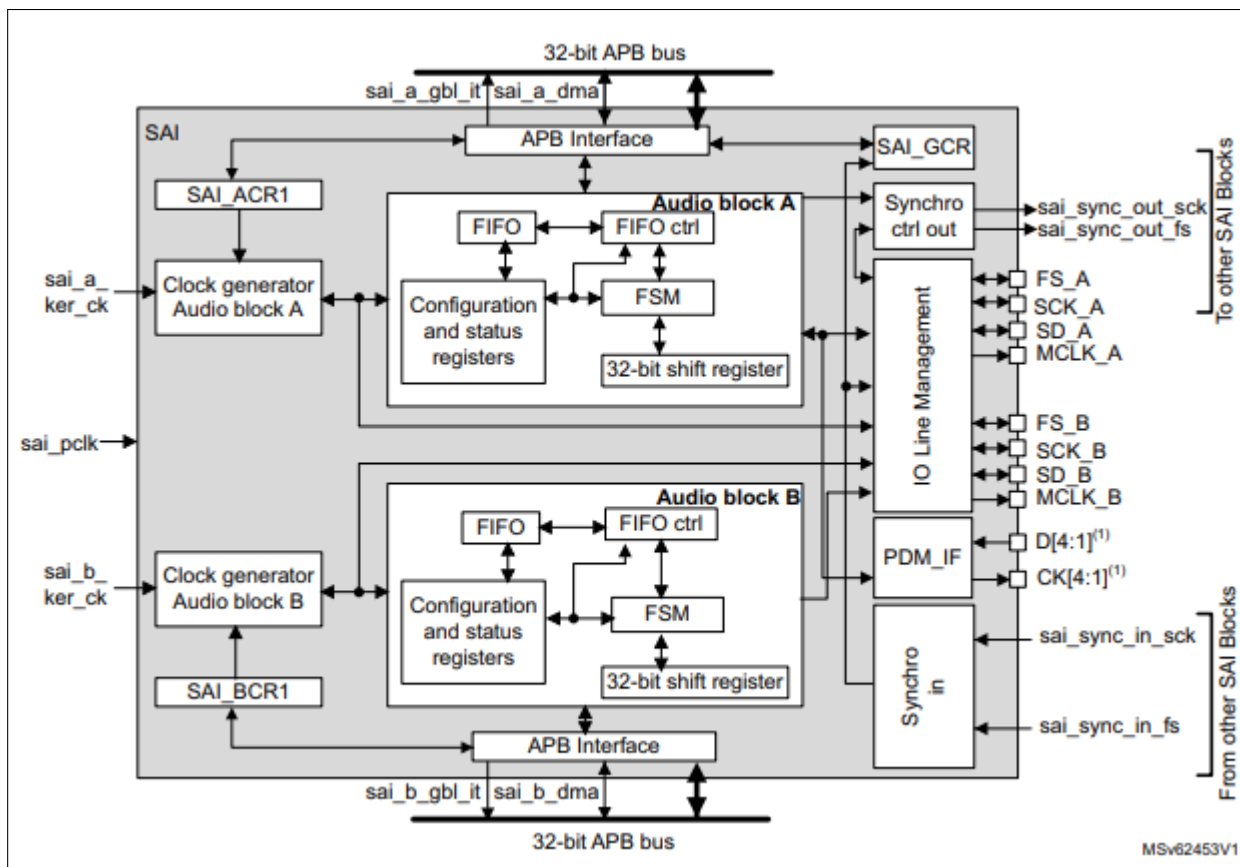


FIGURE 2.3 – Schéma du SAI

### 2.2.2 DFSDM (Digital Filter for Sigma-Delta Modulators)

Le DFSDM est utilisé pour traiter les signaux numériques provenant des modulateurs Sigma-Delta. Ce périphérique est capable de filtrer et de décoder ces signaux, ce qui est essentiel dans les applications embarquées nécessitant un traitement rapide et précis des signaux.

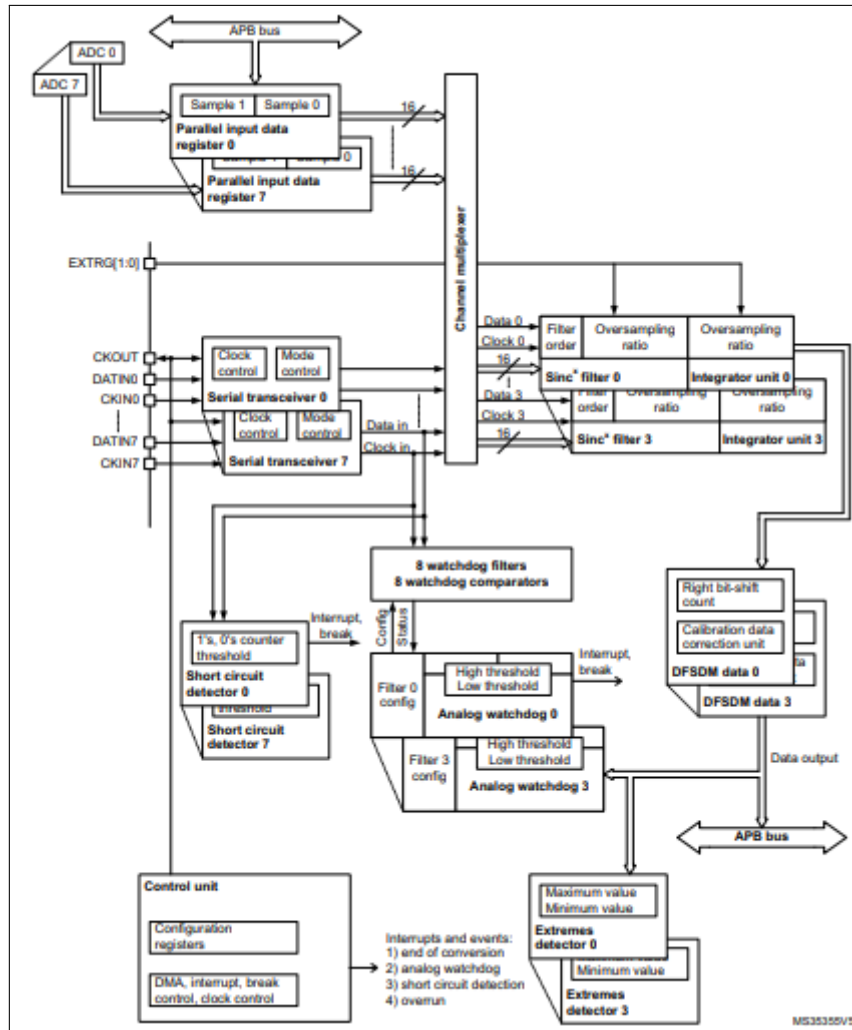


FIGURE 2.4 – Schéma du DFSDM

## 2.3 Architecture STM32Cube

L'architecture STM32Cube facilite le développement des applications sur les microcontrôleurs STM32. Elle est modulaire et comporte plusieurs couches logicielles, allant des pilotes matériels aux bibliothèques intermédiaires [2].

### 2.3.1 Niveaux de l'Architecture

L'architecture est subdivisée en quatre niveaux

- **Niveau matériel** : Ce niveau comprend les différentes cartes STM32, telles que les cartes d'évaluation (Evaluation Boards), les cartes de découverte (Discovery

Boards), et les Nucleo Boards.

- **Niveau 0 (Drivers)** : Il contient les pilotes spécifiques à chaque carte STM32, notamment le Board Support Package (BSP) et les couches d'abstraction matérielle (HAL).
- **Niveau 1 (Middleware)** : Ce niveau regroupe des bibliothèques intermédiaires, telles que FreeRTOS et USB, qui facilitent l'intégration d'applications avancées.
- **Niveau 2 (Applications)** : Il inclut des exemples et des démonstrations pour aider les utilisateurs à se familiariser avec STM32Cube.
- **Couches logicielles externes** : Comprend des utilitaires et des logiciels pour PC ainsi que la suite CMSIS.

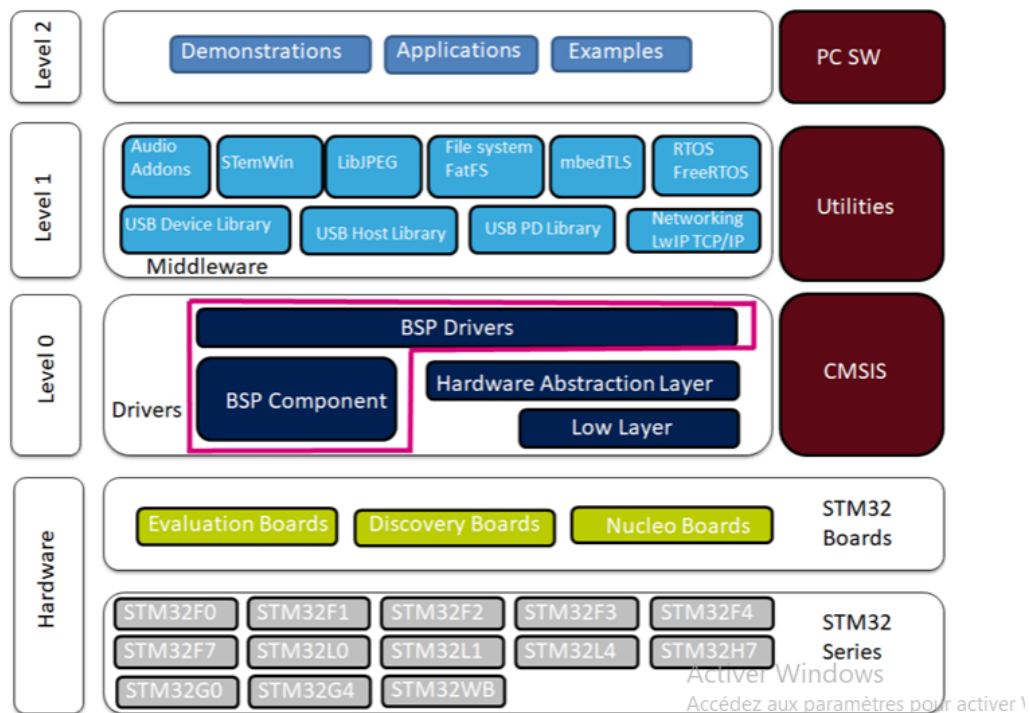


FIGURE 2.5 – Architecture des packages STM32Cube

### 2.3.2 Board Support Package (BSP)

Le BSP est une collection de pilotes qui permet une interface efficace entre le matériel et les logiciels. Il est essentiel pour simplifier le développement sur les microcontrôleurs STM32, en fournissant des outils et des pilotes spécifiques pour chaque carte.

Les pilotes BSP (Board Support Package) des cartes STM32 sont organisés en quatre catégories principales pour simplifier la gestion des périphériques et composants matériels.

Les pilotes de fonctions fournissent des API de haut niveau pour des fonctionnalités spécifiques telles que les écrans LCD, l'audio, et les écrans tactiles, facilitant leur utilisation.

Les pilotes communs offrent des API pour gérer les interfaces homme-machine (HMI) comme les boutons, LEDs, et joysticks, ainsi que les services de communication, rendant l'intégration de ces éléments plus simple.

Les pilotes de composants sont dédiés aux dispositifs externes, indépendants de la couche HAL, et comprennent des API spécifiques aux circuits intégrés externes. Ils sont portables et se composent de fichiers principaux et de fichiers de registres, parfois complétés par un fichier de configuration.

Le pilote de bus fournit une interface pour le transport des données via des protocoles comme I2C ou SPI, assurant une communication standardisée entre la carte et les périphériques.

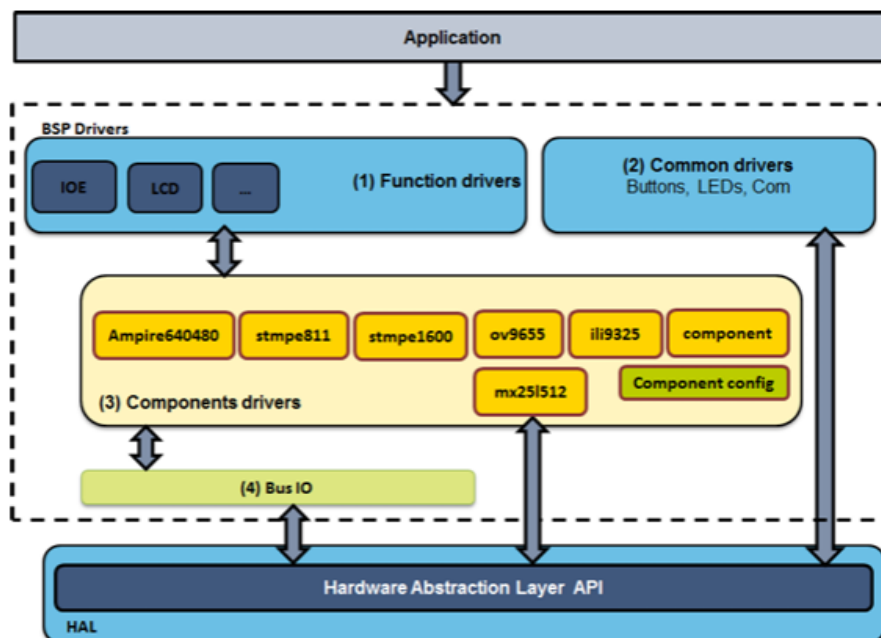


FIGURE 2.6 – Organisation des pilotes BSP

# Conclusion

Dans ce chapitre, j'ai exploré les composants essentiels au développement des systèmes embarqués STM32, incluant les microcontrôleurs, les périphériques spécifiques, l'architecture STM32Cube et le Board Support Package. Ensemble, ces éléments forment une base solide pour le développement efficace de projets utilisant les microcontrôleurs STM32, en assurant flexibilité, modularité et performances optimisées.

# Chapitre 3

## Implémentation d'une Démonstration Audio sur la Carte STM32H735G-DK

### Introduction

Ce chapitre présente l'implémentation pratique d'une démonstration audio utilisant la carte STM32H735G-DK. Ce projet vise à exploiter les capacités avancées de cette carte, en particulier son codec audio WM8994, pour mettre en place une solution complète d'enregistrement et de lecture audio. Les fonctionnalités analogiques et numériques seront comparées, avec une attention particulière portée aux défis techniques liés à l'enregistrement audio embarqué.

### 3.1 Carte de développement STM32H735G-DK

Dans ce projet, j'ai travaillé sur la carte STM32H735G-DK pour intégrer une démonstration analogique. Cette carte est équipée du microcontrôleur STM32H735IG, basé sur le Cortex-M7 cadencé à 550 MHz, offrant des performances élevées pour les



applications embarquées. Elle inclut également un codec audio, essentiel pour le traitement des signaux sonores. En plus, elle propose un large éventail de fonctionnalités, telles qu'un écran LCD 4,3 pouces avec interface tactile, des interfaces Ethernet, USB OTG, ainsi que des extensions compatibles avec Arduino et Pmod [3].

Ces caractéristiques m'ont permis de développer et tester efficacement ma démo analogique dans un environnement optimisé.



FIGURE 3.1 – Carte de développement STM32H735G-DK

## 3.2 Fonctionnalités Audio de la STM32H735G-DK

Dans cette section, nous présentons les capacités audio de la carte STM32H735G-DK, notamment le codec audio WM8994 ainsi que les différentes périphériques et microphones

### 3.2.1 codec audio WM8994

Le codec audio WM8994, utilisé dans la carte STM32H735G-DK, permet de gérer efficacement les signaux audio. Il dispose de 4 canaux DAC 24 bits pour la lecture audio, offrant une conversion numérique-analogique de haute qualité. Il intègre également 2 canaux ADC 24 bits pour l'enregistrement, permettant de convertir les signaux analogiques en données numériques avec une grande précision [4]. Ces capacités font du WM8994 un composant clé pour les applications nécessitant des performances audio avancées, comme la lecture et l'enregistrement en temps réel.

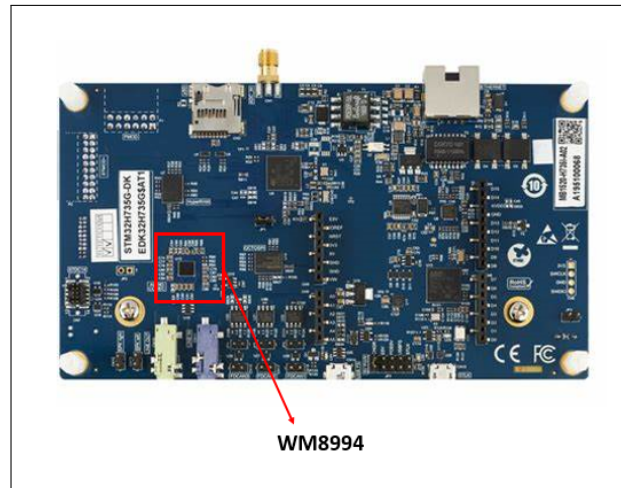


FIGURE 3.2 – Codec audio WM8994

#### 3.2.2 Enregistrement Audio numérique

L'enregistrement numérique sur la carte STM32H735G-DK repose sur l'utilisation de microphones MEMS numériques, émettant un signal PDM (Pulse Density Modulation). Ce signal est capturé via des interfaces telles que SPI/I2S, SAI ou DFSDM, qui assurent la gestion et le traitement du signal par le microcontrôleur STM32.

Le processus de conversion du PDM en PCM (Pulse Code Modulation) se fait par un traitement logiciel via un filtre numérique. Le filtrage élimine les hautes fréquences indésirables et la décimation réduit la fréquence d'échantillonnage, passant de la fréquence élevée PDM à la fréquence audio cible (par exemple 16 kHz ou 48 kHz). Ce traitement produit un flux audio au format PCM, utilisable dans les applications audio.

Les interfaces DFSDM et SAI sont particulièrement adaptées à ce processus, car elles permettent de traiter plusieurs microphones en parallèle, offrant ainsi une solution efficace pour des applications nécessitant plusieurs sources audio (comme la capture stéréo ou le beamforming). De plus, l'interface DFSDM permet de filtrer et de décimer directement les données PDM, sans besoin de traitement externe supplémentaire.

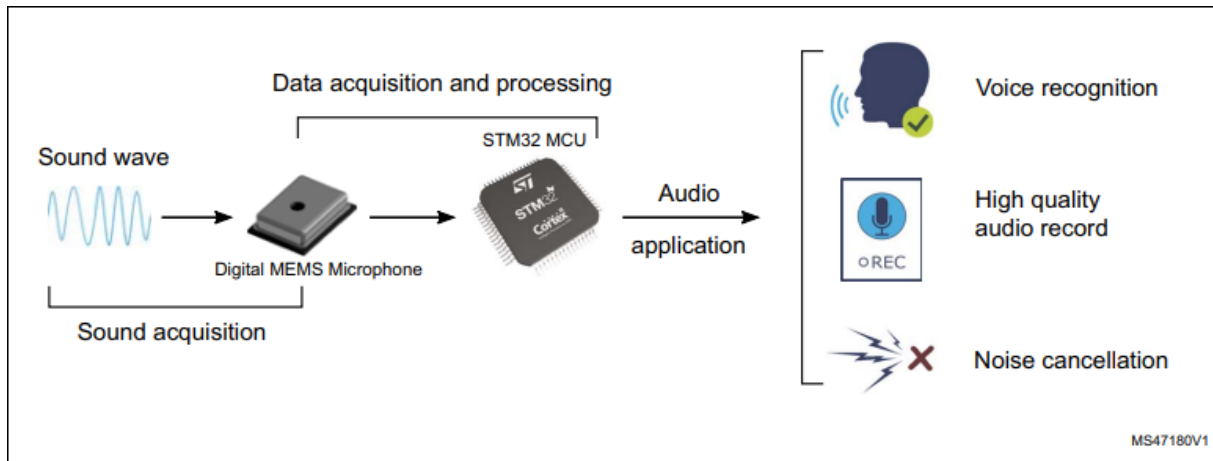


FIGURE 3.3 – Exemple d’acquisition de son dans une application audio

#### Microphone numérique MEMS IMP34DT05

IMP34DT05 est un microphone numérique MEMS (Micro-Electro-Mechanical System) développé par STMicroelectronics, conçu pour capturer des signaux audio avec une sortie numérique PDM (Pulse Density Modulation). Il offre une bonne qualité sonore avec un rapport signal/bruit de 64 dB, idéal pour des applications telles que les smartphones, les assistants vocaux et autres systèmes embarqués nécessitant de l’enregistrement ou détection sonore. Compact et à faible consommation (1.8V à 3.6V), il est sensible aux sons faibles avec une sensibilité d’environ -26 dBFS, ce qui en fait un capteur audio performant et adapté à des environnements limités en espace [5].



FIGURE 3.4 – Microphone numérique IMP34DT05

### 3.2.3 Enregistrement Audio Analogique

Le codec WM8994 joue un rôle central dans l'enregistrement audio analogique sur une carte STM32 via une entrée jack. Il permet de convertir un signal audio analogique, capté par un microphone ou un autre périphérique connecté à l'entrée jack, en un signal numérique que le microcontrôleur STM32 peut traiter. Ce codec intègre un convertisseur analogique-numérique (ADC), qui prend en charge cette conversion.

Lorsque le signal analogique entre dans le WM8994, il est immédiatement converti en un flux numérique, généralement au format I2S (ou TDM). Ce format numérique est ensuite transmis au SAI (Serial Audio Interface) du STM32, qui gère la réception et l'enregistrement des données. Le SAI, étant spécifiquement conçu pour traiter des flux numériques, ne peut pas fonctionner avec un signal analogique directement. C'est pourquoi le WM8994 est indispensable dans ce processus. En plus de la conversion, le WM8994 peut aussi effectuer des traitements audio tels que le contrôle du volume, l'amplification, et le filtrage des signaux avant qu'ils ne soient envoyés au STM32.

Le processus d'enregistrement audio avec le WM8994 et le SAI permet de capturer des sons via une entrée analogique, de les convertir en numérique, puis de les traiter ou de les stocker pour des applications audio variées.

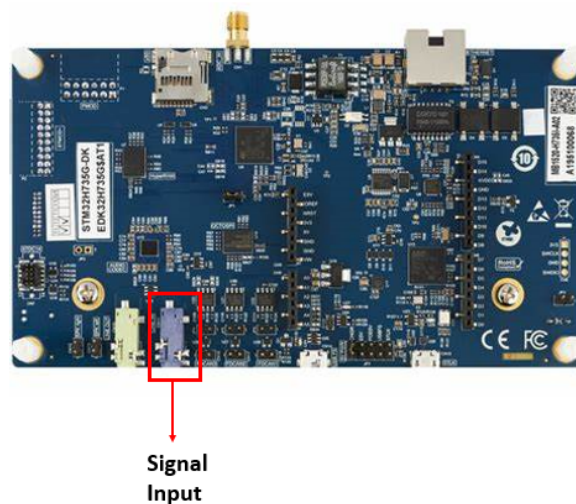


FIGURE 3.5 – Entrée jack pour l'enregistrement

### 3.2.4 Comparaison entre l'enregistrement audio analogique et numérique

TABLE 3.1 – Comparaison entre l'enregistrement audio analogique et numérique

Critère	Enregistrement Analogique	Enregistrement Numérique
Source	Microphones analogiques via une entrée jack	Microphones MEMS numériques
Conversion	Utilise un codec audio (WM8994) pour convertir en numérique	Conversion PDM en PCM via un traitement logiciel
Qualité du signal	Dépend de la qualité de conversion ADC	Haute fidélité sans conversion ADC
Interfaces utilisées	SAI, I2S	SAI, DFSDM, I2S
Applications	Adapté aux sources analogiques traditionnelles	déal pour des applications stéréo ou multicanal complexes .

L'enregistrement analogique repose sur l'utilisation de microphones traditionnels, dont le signal est converti en numérique à l'aide d'un codec audio, comme le WM8994 . Ce mode est particulièrement adapté aux sources audio analogiques, telles que les instruments ou microphones classiques. En revanche, l'enregistrement numérique utilise des microphones MEMS, qui génèrent directement un signal PDM (Pulse Density Modulation), éliminant ainsi le besoin d'une conversion analogique-numérique [6] .

### 3.2.5 Fonctions BSP Audio IN pour l'enregistrement numérique et analogique

L'enregistrement audio, qu'il soit numérique ou analogique, utilise les mêmes fonctions fournies par l'API BSP AUDIO IN de la carte STM32H735G. Le tableau ci-dessous présente les principales fonctions disponibles pour l'enregistrement audio, y compris l'initialisation, la pause, la reprise et l'arrêt de l'enregistrement.

TABLE 3.2 – BSP AUDIO IN class APIs

Function	Arguments	Description
int32_t BSP_AUDIO_IN_Init	uint32_t Instance BSP_AUDIO_Init_t *AUDIO_Init	AUDIO IN Initialization
int32_t BSP_AUDIO_IN_DeInit	uint32_t Instance	AUDIO IN De-Initialization
__weak HAL_StatusTypeDef MX_PPP_Init	PPP_HandleTypeDef *hPPP MX_PPP_TypeDef *Init	AUDIO IN periph configuration
static HAL_StatusTypeDef PPP_MspInit	PPP_HandleTypeDef *hPPP	Initializes default AUDIO OUT MSP part
static HAL_StatusTypeDef PPP_MspDeInit	PPP_HandleTypeDef *hPPP	De-Initializes AUDIO IN MSP part
__weak HAL_StatusTypeDef MX_PPP_ClockConfig	uint32_t SampleRate	Set AUDIO IN clock configuration
int32_t BSP_AUDIO_IN_Record	uint8_t *pBuf uint32_t NbrOfBytes	Start audio Recording in default mode
int32_t BSP_AUDIO_IN_Pause	uint32_t Instance	Pause audio recording
int32_t BSP_AUDIO_IN_Resume	uint32_t Instance	Resume audio recording
int32_t BSP_AUDIO_IN_Stop	uint32_t Instance	Stop audio recording
int32_t BSP_AUDIO_IN_SetDevice	uint32_t Instance uint32_t Device	Set AUDIO IN input device
int32_t BSP_AUDIO_IN_GetDevice	uint32_t Instance uint32_t *Device	Get AUDIO IN input device
int32_t BSP_AUDIO_IN_SetSampleRate	uint32_t Instance uint32_t SampleRate	Set AUDIO IN frequency
int32_t BSP_AUDIO_IN_GetSampleRate	uint32_t Instance uint32_t *SampleRate	Get AUDIO IN frequency
int32_t BSP_AUDIO_IN_SetBitsPerSample	uint32_t Instance uint32_t BitsPerSample	Set AUDIO IN sample resolution
int32_t BSP_AUDIO_IN_GetBitsPerSample	uint32_t Instance uint32_t *BitsPerSample	Get AUDIO IN sample resolution
int32_t BSP_AUDIO_IN_SetChannelsNbr	uint32_t Instance uint32_t ChannelNbr	Set AUDIO IN channels number
int32_t BSP_AUDIO_IN_GetChannelsNbr	uint32_t Instance uint32_t *ChannelNbr	Get AUDIO IN channels number
int32_t BSP_AUDIO_IN_GetState	uint32_t Instance, uint32_t *State	Get AUDIO IN state : AUDIO_IN_STATE_RESET AUDIO_IN_STATE_RECORDING AUDIO_IN_STATE_STOP AUDIO_IN_STATE_PAUSE
void BSP_AUDIO_IN_DMA_IRQHandler	uint32_t Instance uint32_t Device	Handles audio IN DMA transfer request depending on used instance and device
void BSP_AUDIO_IN_TransferComplete_Callback	uint32_t Instance	Transfer Complete callback
void BSP_AUDIO_IN_HalfTransfer_Callback	uint32_t Instance	Half Transfer callback
void BSP_AUDIO_IN_Error_Callback	uint32_t Instance	Error callback
int32_t BSP_AUDIO_IN_RegisterMspCallbacks	uint32_t Instance BSP_AUDIO_IN_Cb_t *Callbacks	Register AUDIO IN Msp callbacks
int32_t BSP_AUDIO_IN_RegisterDefaultMspCallbacks	uint32_t Instance	Register AUDIO IN default Msp callbacks

### 3.2.6 Lecture Audio

Le processus de lecture audio sur une carte STM32, via un codec comme le WM8994, repose sur la conversion d'un flux numérique en signal analogique. Les données audio, généralement en format I2S, sont transmises par le microcontrôleur STM32 au codec via le SAI (Serial Audio Interface). Le WM8994, qui intègre un convertisseur numérique-analogique (DAC), joue un rôle central en transformant ces données numériques en un signal analogique, prêt à être reproduit sur des périphériques audio comme des haut-parleurs ou des écouteurs via une sortie jack. Le SAI, conçu pour gérer des flux numériques, ne peut pas directement traiter les signaux analogiques, d'où l'importance du WM8994 dans ce processus. En plus de la conversion, le codec peut effectuer des traitements supplémentaires comme le contrôle du volume et l'amplification du signal pour garantir une qualité audio optimale. Ce mécanisme de lecture audio permet ainsi de décoder les fichiers numériques, de les convertir en sons analogiques, puis de les diffuser à travers une sortie audio, offrant une solution efficace pour la lecture de contenu sonore sur des plateformes STM32.

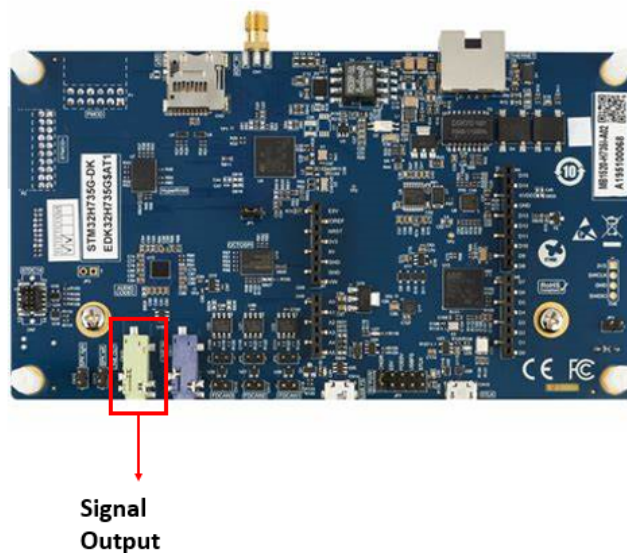


FIGURE 3.6 – Sortie jack pour la lecture

### 3.2.7 Fonctions BSP Audio OUT pour la lecture Audio

API BSP AUDIO OUT fournit un ensemble de fonctions permettant de gérer la lecture audio. Ces fonctions sont utilisées pour contrôler le flux de sortie audio, que ce soit pour un fichier audio numérique ou pour la conversion d'un signal analogique.

Les principales fonctionnalités de l'API BSP AUDIO OUT incluent l'initialisation des périphériques de sortie audio, le démarrage de la lecture, ainsi que la gestion du volume et des actions comme la pause ou l'arrêt de la lecture. Le tableau ci-dessous présente les fonctions disponibles pour la gestion de la sortie audio.

TABLE 3.3 – BSP AUDIO IN class APIs

Function	Arguments	Description
int32_t BSP_AUDIO_IN_Init	uint32_t Instance BSP_AUDIO_Init_t *AUDIO_Init	AUDIO IN Initialization
int32_t BSP_AUDIO_IN_DeInit	uint32_t Instance	AUDIO IN De-Initialization
int32_t BSP_AUDIO_IN_Record	uint32_t Instance uint8_t *pBuf uint32_t NbrOfBytes	Start audio Recording in default mode
int32_t BSP_AUDIO_IN_Pause	uint32_t Instance	Pause audio recording
int32_t BSP_AUDIO_IN_Resume	uint32_t Instance	Resume audio recording
int32_t BSP_AUDIO_IN_Stop	uint32_t Instance	Stop audio recording
int32_t BSP_AUDIO_IN_SetDevice	uint32_t Instance uint32_t Device	Set AUDIO IN input device
int32_t BSP_AUDIO_IN_GetDevice	uint32_t Instance uint32_t *Device	Get AUDIO IN input device
int32_t BSP_AUDIO_IN_SetSampleRate	uint32_t Instance uint32_t SampleRate	Set AUDIO IN frequency
int32_t BSP_AUDIO_IN_GetSampleRate	uint32_t Instance uint32_t *SampleRate	Get AUDIO IN frequency
int32_t BSP_AUDIO_IN_SetBitsPerSample	uint32_t Instance uint32_t BitsPerSample	Set AUDIO IN sample resolution
int32_t BSP_AUDIO_IN_GetBitsPerSample	uint32_t Instance uint32_t *BitsPerSample	Get AUDIO IN sample resolution
int32_t BSP_AUDIO_IN_SetChannelsNbr	uint32_t Instance uint32_t ChannelNbr	Set AUDIO IN channels number
int32_t BSP_AUDIO_IN_GetChannelsNbr	uint32_t Instance uint32_t *ChannelNbr	Get AUDIO IN channels number
void BSP_AUDIO_IN_DMA_IRQHandler	uint32_t Instance uint32_t Device	Handles audio IN DMA transfer request depending on used instance and device
void BSP_AUDIO_IN_TransferComplete_Callback	uint32_t Instance	Transfer Complete callback
void BSP_AUDIO_IN_HalfTransfer_Callback	uint32_t Instance	Half Transfer callback
void BSP_AUDIO_IN_Error_Callback	uint32_t Instance	Error callback
int32_t BSP_AUDIO_IN_RegisterMspCallbacks	uint32_t Instance BSP_AUDIO_IN_Cb_t *Callbacks	Register AUDIO IN Msp callbacks
int32_t BSP_AUDIO_IN_RegisterDefaultMspCallbacks	uint32_t Instance	Register AUDIO IN default Msp callbacks



### 3.2.8 Démonstrations BSP

Le BSP démo comporte plusieurs fonctionnalités, notamment la gestion du LCD, du TS (Touchscreen), de l'OCTOSPI, de l'Audio, et des SD Cards. Le projet se concentre principalement sur l'ajout d'une démo spécifique : l'Audio Analog Recording. Cette démo vise à permettre l'enregistrement audio analogique et la lecture en temps réel, en capturant un signal via une entrée Jack et en le traitant immédiatement pour un usage direct.

L'organigramme suivant présente les différentes démonstrations BSP disponibles sur la carte STM32H735G, avec une attention particulière portée à l'enregistrement audio analogique (en rouge), qui a été intégré dans ce projet. Ce flux met en évidence le processus de capture, traitement et lecture du signal audio en temps réel.

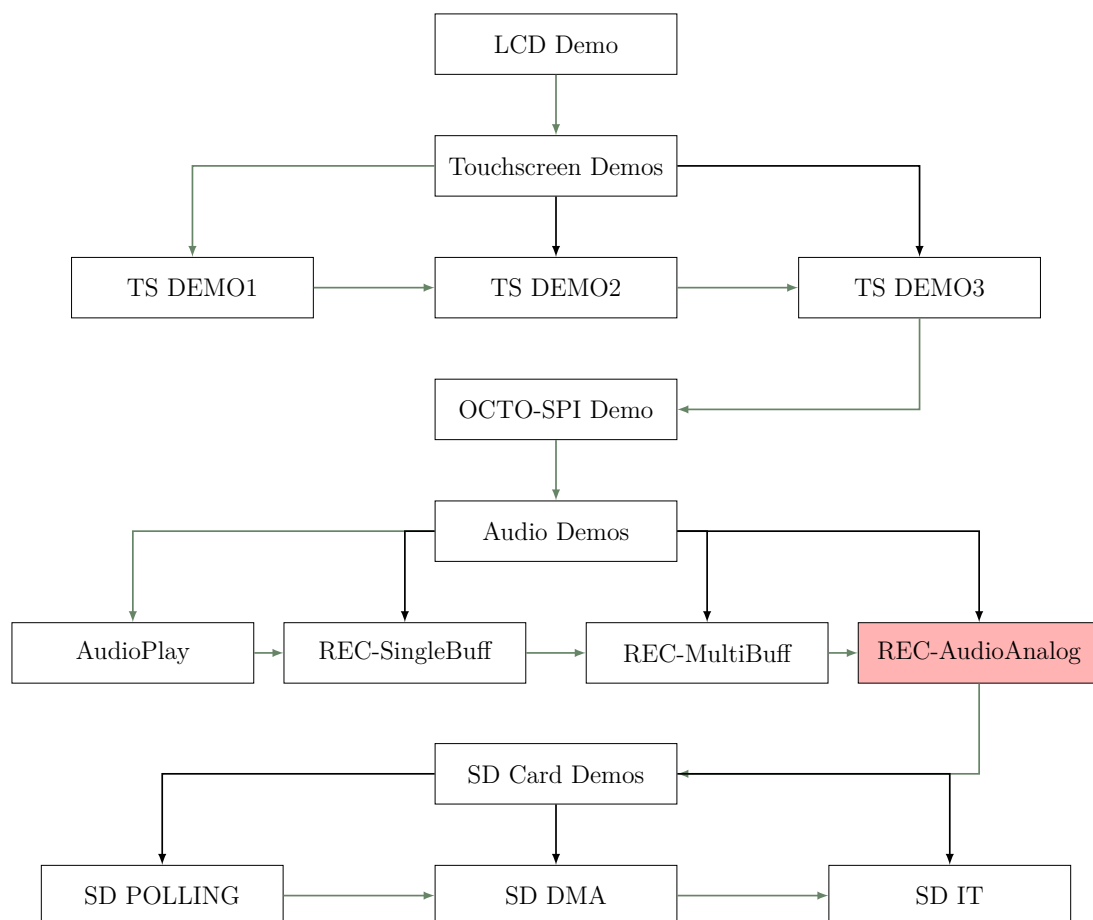


FIGURE 3.7 – Organigramme des exemples de démonstrations BSP

### 3.2.9 Fonction d'enregistrement audio analogique

Cette démonstration illustre le processus d'enregistrement audio analogique sur la carte STM32H735G, permettant de capturer un signal audio analogique provenant d'une source comme un microphone ou une entrée Jack. Dès que l'utilisateur lance l'enregistrement, les périphériques audio sont initialisés grâce à la fonction `BSP_AUDIO_IN_Init`, qui configure les paramètres tels que le taux d'échantillonnage, la résolution et le nombre de canaux.

Une fois l'initialisation terminée, la fonction `BSP_AUDIO_IN_Record` démarre l'enregistrement. Le signal analogique est capturé par l'entrée Jack et converti en numérique à l'aide du convertisseur analogique-numérique (ADC) intégré à la carte. Cette conversion est essentielle pour traiter le signal dans un environnement numérique, afin de le stocker ou de l'utiliser en temps réel.

Après la capture du signal, la fonction `BSP_AUDIO_OUT_Play` est utilisée pour lire le flux audio enregistré. Cette fonction permet de transférer les données audio vers le périphérique de sortie, comme des haut-parleurs ou des écouteurs, offrant à l'utilisateur la possibilité d'écouter immédiatement l'audio capturé.

Pendant l'enregistrement ou la lecture, si l'utilisateur appuie sur le bouton User, une interruption est déclenchée. Pour l'enregistrement, la fonction `BSP_AUDIO_IN_Stop` est appelée pour stopper la capture du signal audio. Pour la lecture, la fonction `BSP_AUDIO_OUT_Stop` est utilisée pour interrompre la lecture. Ces actions assurent un contrôle en temps réel du processus d'enregistrement et de lecture audio.

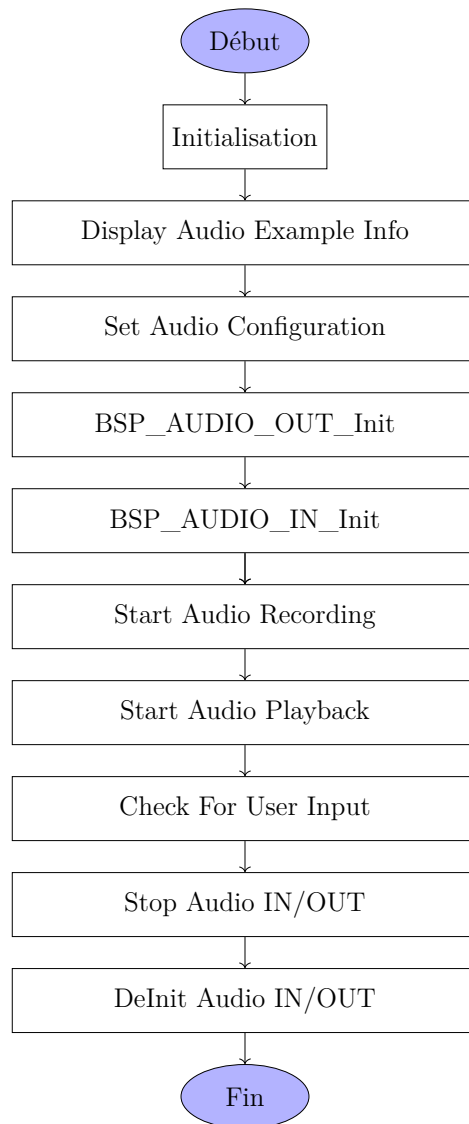


FIGURE 3.8 – Organigramme de la fonction d’enregistrement audio analogique

## Conclusion

Dans ce chapitre, j’ai mis en œuvre l’intégration d’une démonstration audio sur la carte STM32H735G-DK, en exploitant les capacités du codec audio WM8994 pour l’enregistrement et la lecture audio. J’ai comparé les approches d’enregistrement analogique et numérique, en utilisant des microphones MEMS pour le numérique et une entrée jack pour l’analogique. Les interfaces SAI et DFSDM, ainsi que les fonctions BSP AUDIO IN et OUT, ont permis de gérer efficacement les flux audio sur cette plateforme.

# Conclusion générale

Mon expérience de stage chez STMicroelectronics a été un tournant décisif dans ma formation professionnelle. En me plongeant dans le domaine des systèmes embarqués, j'ai eu l'occasion d'explorer les subtilités des microcontrôleurs STM32 et de comprendre leur rôle vital dans le développement de solutions innovantes. La mise en œuvre d'une démonstration audio a été particulièrement formatrice, me permettant d'appliquer des concepts théoriques à des situations réelles.

Cette immersion m'a également permis de renforcer mes compétences en travail d'équipe, en apprenant à collaborer efficacement avec des ingénieurs de divers horizons. Les échanges et les défis rencontrés m'ont aidé à développer une approche analytique face à des problèmes complexes, tout en stimulant ma créativité.

En résumé, ce stage a non seulement enrichi mes connaissances techniques, mais il a aussi été un catalyseur pour ma croissance personnelle. Je suis désormais mieux préparé à affronter les défis futurs dans le domaine des systèmes embarqués, avec une confiance renouvelée et une passion pour l'innovation technologique.

# Références bibliographiques

[1] : <https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.html>

[2] : <https://www.st.com/en/development-tools/stm32cube.htm>

[3] : <https://www.st.com/en/evaluation-tools/stm32h735g-dk.html>

[4] : <https://www.cirrus.com/products/wm8994/>

[5] : <https://www.st.com/en/mems-and-sensors/imp34dt05.html>

[6] : <https://www.circuitdigest.com/article/analog-vs-digital-audio>