

# Rapport soutenance finale

- Super Simon -



Eve GRESSE  
Maxime GOËLLER  
Gauthier MAUPAS

- 5 Février 2025 -

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Le jeu du Simon</b>	<b>2</b>
<b>3</b>	<b>Réalisation du projet</b>	<b>3</b>
3.1	Spécifications du projet . . . . .	3
3.2	Contraintes . . . . .	4
3.3	Réalisation hardware . . . . .	5
3.3.1	Première version de la PCB . . . . .	5
3.3.2	Seconde version de la PCB . . . . .	6
3.3.3	Fabrication et soudure . . . . .	8
3.4	Réalisation firmware . . . . .	8
3.5	Problèmes rencontrés . . . . .	9
<b>4</b>	<b>Test et validation</b>	<b>10</b>
<b>5</b>	<b>Perspectives d'évolution</b>	<b>11</b>
<b>6</b>	<b>Conclusion</b>	<b>12</b>
	<b>Glossaire</b>	<b>13</b>
	<b>Annexes</b>	<b>16</b>

# 1 Introduction

Les jeux électroniques ont toujours occupé une place importante dans le divertissement, alliant mémoire, réflexe et stratégie. Parmi eux, le jeu du Simon, conçu en 1978, a marqué plusieurs générations en proposant un défi simple mais addictif : mémoriser et reproduire une séquence lumineuse et sonore.

Dans le cadre de notre projet de fin d'études, nous avons choisi de revisiter ce jeu emblématique en développant notre propre version : le Super Simon. Notre objectif est de modifier le jeu en intégrant de nouvelles fonctionnalités, tout en conservant le principe fondamental qui a fait le succès du Simon original. Nous avons ainsi conçu et réalisé notre propre version du jeu en développant une carte électronique dédiée, en sélectionnant et intégrant les composants adaptés et en programmant l'ensemble des fonctionnalités.

Ce rapport détaille les différentes étapes de conception et de réalisation de notre projet, depuis la définition des spécifications jusqu'à l'implémentation des composants matériels et logiciels. Nous présenterons également les défis rencontrés, les tests effectués pour valider notre solution, ainsi que les perspectives d'évolution possibles.

## 2 Le jeu du Simon

Le jeu du Simon est un jeu électronique de mémoire, inventé en 1978, qui met à l'épreuve la capacité des joueurs à mémoriser et reproduire des séquences lumineuses et sonores. Il se compose généralement de quatre boutons colorés, chacun associé à un son distinct. Le jeu commence par l'illumination aléatoire d'un bouton, que le joueur doit répéter. À chaque tour réussi, la séquence s'allonge, augmentant ainsi la difficulté.

Au fil des années, plusieurs variantes du jeu ont vu le jour. Certaines versions ajoutent des couleurs et des sons supplémentaires, d'autres introduisent des modes de jeu alternatifs, comme des défis en coopération ou en compétition. Il existe aussi des adaptations sous forme de jeux vidéo et d'applications mobiles, intégrant des fonctionnalités interactives et des graphismes modernes.

L'objectif principal du jeu reste le même : tester et améliorer la mémoire des joueurs de manière ludique et progressive.



FIGURE 1 – Simon

## 3 Réalisation du projet

Le principe de ce PFE est donc de réaliser le jeu du Simon tout en le revisitant. Nous avions pour objectif de réaliser tant la partie hardware (PCB, choix des composants, soudure) que la partie software (implémentation du jeu et des différentes options).

### 3.1 Spécifications du projet

Le projet que nous avons réalisé est composé des éléments hardwares suivants :

- **STM32 Nucleo** : Gestion du jeu, génération des séquences et détection des entrées utilisateur.
- **Boutons** : Huit servant pour la saisie des séquences.
- **LEDs** : Indication visuelle de l'état du jeu et des séquences à reproduire.
- **Module sonore** : Émission de sons distinctifs associés à chaque bouton ainsi que des musiques de fin pour renforcer l'expérience utilisateur.
- **Alimentation par pile** : Source d'énergie adaptée au fonctionnement du circuit.
- **Interrupteur d'activation** : Permet d'allumer et d'éteindre le jeu.
- **Potentiomètre de réglage du son** : Ajuste le volume sonore en fonction des préférences de l'utilisateur.
- **Driver** : Composant communiquant en SPI avec le microcontrôleur afin de gérer les LEDs.

#### Modes de jeu

Le projet propose plusieurs modes de jeu pour diversifier l'expérience utilisateur :

- **Mode Classique** : Le Simon affiche une séquence de touches en ajoutant une nouvelle touche à chaque manche. Le joueur doit la reproduire sans erreur. Seules 4 couleurs sont utilisées.
- **Mode 8 couleurs** : Identique au mode Classique, mais avec 8 couleurs différentes.
- **Mode Fantôme** : Après l'affichage de la séquence, les LED s'éteignent, obligeant le joueur à mémoriser la position ou le son des touches.
- **Mode Color Switch** : La séquence est affichée normalement, mais les couleurs changent de place avant que le joueur ne la reproduise. Il faut donc retenir la couleur plutôt que la position.

- **Mode Partie rapide** : Similaire au mode Classique, mais deux nouvelles touches sont ajoutées à chaque manche, rendant le jeu plus dynamique.
- **Mode Duel** : Inspiré du mode existant sur le Super Simon de 1980, ce mode oppose deux joueurs. Une séquence est diffusée et le premier joueur à la reproduire correctement gagne la manche. En cas d'erreur, l'adversaire peut tenter sa chance.

### Niveaux de difficulté

Pour s'adapter aux différents profils de joueurs, plusieurs niveaux de difficulté sont intégrés :

- **Trivial** : La vitesse des séquences est réduite.
- **Normal** : Mode standard avec une vitesse de séquence modérée.
- **Légende** : La vitesse des séquences augmente rapidement.

Ces options permettent de personnaliser l'expérience de jeu en fonction des préférences et du niveau de compétence des joueurs.

## 3.2 Contraintes

La première contrainte rencontrée fut notre budget, pour deux raisons principales. La première - détaillée davantage par la suite - concerne le driver de LED. Comme la PCB n'était pas encore finalisée, nous avons dû utiliser un Dev Kit. Cependant, le Dev Kit du composant initialement choisi s'est révélé trop onéreux pour le PFE, nous obligeant à nous tourner vers un autre composant dont le Dev Kit était plus abordable. Pour des raisons similaires, lors du soudage de notre seconde PCB, nous avons dû nous limiter quant à la production et à l'utilisation d'un stencil afin de respecter notre budget, au prix d'une légère perte de précision.

Une autre contrainte majeure a été la taille de notre PCB, restreinte à 10cm x 10cm, une fois de plus pour limiter les frais. En effet à plusieurs reprises le "câblage" de nos composants a été repensé afin d'optimiser la lisibilité et la compréhension de la carte, facilitant ainsi la correction d'éventuelles erreurs pouvant être détecté par les outils de conception de PCB. La suppression de l'écran a toutefois réduit l'importance de cette problématique.

Au-delà des contraintes budgétaires, le facteur temps s'est également révélé un défi de taille. En plus d'un planning serré tout au long de l'année, nous avons parfois été contraints d'attendre des avancées dans la partie Hardware avant de poursuivre le développement software. Deux exemples illustrent bien ces limitations : L'intégration de l'écran, qui nous a fait perdre plusieurs semaines à tenter de le faire fonctionner (plus d'explications dans la suite du rapport), et l'attente de la réception du Dev Kit et de la PCB, rendant les tests difficiles à mener à bien.

### 3.3 Réalisation hardware

Nous avons réalisé deux versions de notre PCB. Durant le premier semestre, nous avons conçu, commandé et soudé notre première version de la PCB. Durant le deuxième semestre, nous avons réalisé une deuxième version de notre PCB afin de corriger les différents problèmes rencontrés sur la première version.

#### 3.3.1 Première version de la PCB

Durant le premier semestre, nous avons conçu une première version de la PCB. Pour cette dernière, nous avons utilisé une STM32F103C8T6 (Blue Pill) comme microcontrôleur pour gérer les entrées des boutons, le contrôle des LEDs et la logique du jeu. Cette PCB était composée de huit boutons poussoirs, huit LEDs, un buzzer, un driver de LED (LP5024) et un connecteur pour interfaçer un écran via une nappe.

Lors de la soudure des composants et des premiers tests, nous avons identifié plusieurs problèmes sur cette version de la PCB :

- Erreur d'empreinte du LP5024 : L'empreinte du composant était trop petite ce qui a rendu la soudure du composant difficile.
- Inversion de deux pins des LEDs RGB : Deux pins des LEDs étaient inversées, ce qui empêchait un affichage correct des couleurs.

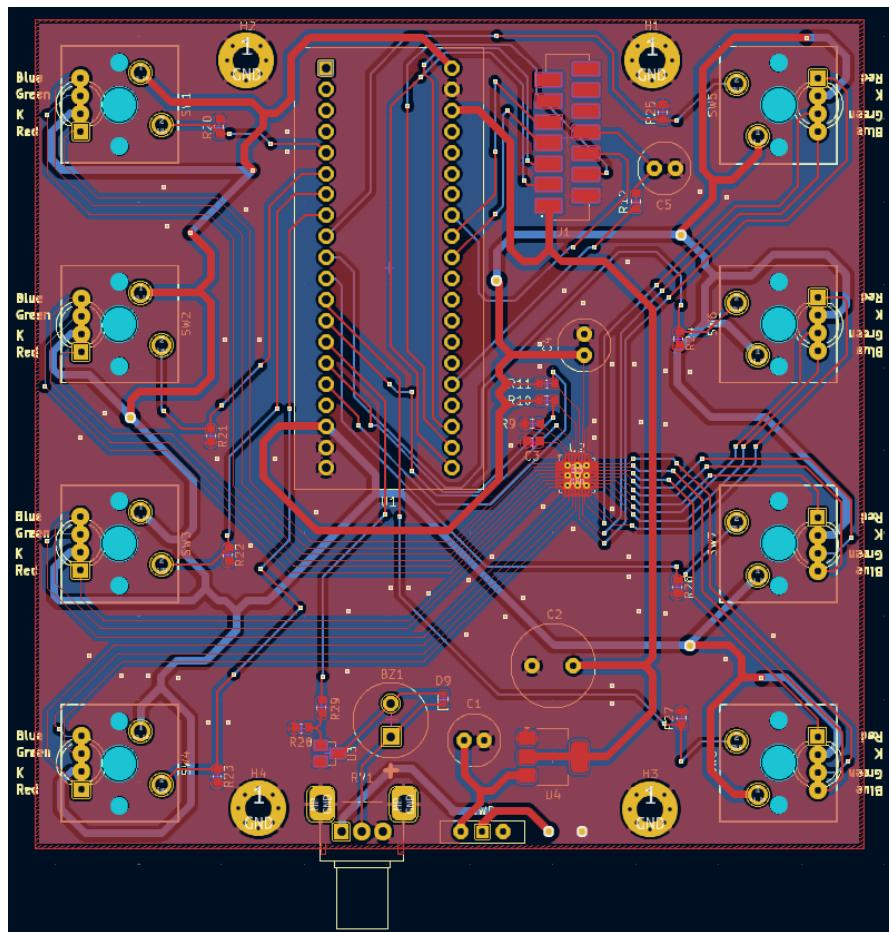


FIGURE 2 – Première version de la PCB sous KiCad

### 3.3.2 Seconde version de la PCB

Durant le deuxième semestre, en parallèle du développement du software, nous avons fait une refonte de la PCB afin de corriger les erreurs précédentes et d'optimiser notre architecture.

L'un des principaux changements apportés concerne le driver. Le coût du kit de développement du LP5024 était hors budget ( $\sim 170\text{€}$ ). Afin de pouvoir poursuivre le développement du software et de réduire les coûts du projet, nous avons choisi d'utiliser un autre driver, le TLC5947, dont le kit de développement est disponible à un prix environ dix fois inférieur ( $\sim 17\text{€}$ ). Grâce à ce changement, nous avons pu continuer le développement sur une breadboard, évitant ainsi un blocage lié à la réalisation hardware.

Nous avons également, pour cette nouvelle version, changé de microcontrôleur. Nous avons remplacé la STM32F103C8T6 par une STM32F303K8, intégrée à une carte Nucleo qui possède un ST-Link embarqué. Ce choix nous a permis d'éliminer les problèmes de flashage rencontrés avec la Blue Pill.

Enfin, nous avons supprimé la connectique pour l'écran. Ce choix a été fait suite à des contraintes identifiées au cours du développement et sera détaillé dans la section Problèmes rencontrés.

Cette nouvelle PCB est une version simplifiée et optimisée, elle est focalisée sur les fonctionnalités essentielles de notre jeu du Simon.

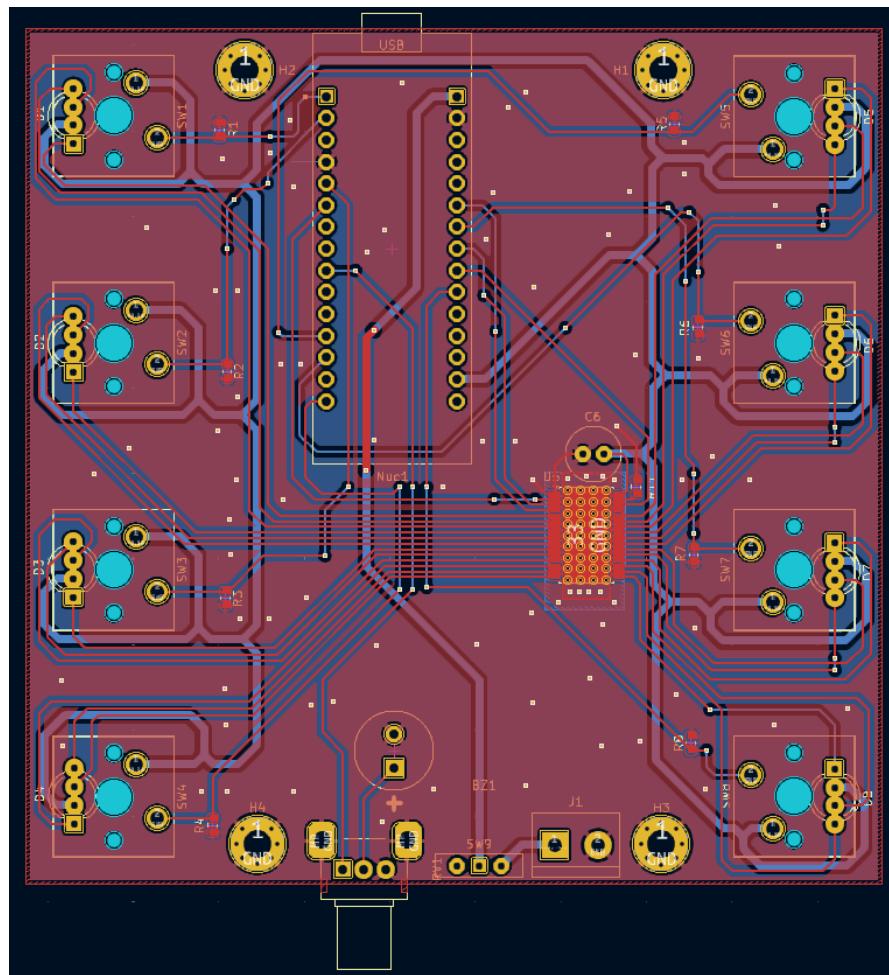


FIGURE 3 – Deuxième version de la PCB sous KiCad

### 3.3.3 Fabrication et soudure

La conception des deux versions de la PCB a été réalisée sur KiCad.

Pour la soudure des composants, deux méthodes ont été utilisées en fonction des versions de la PCB :

- Première version : Nous avons soudé les composants montés en surface (CMS) au four en utilisant un stencil. Les composants traversants et certains composants CMS ont ensuite été soudés manuellement au fer à souder.
- Seconde version : L'ensemble des composants a été soudé manuellement au fer à souder.

## 3.4 Réalisation firmware

Le développement du firmware a été réalisé en parallèle de la conception de notre PCB. Afin de ne pas être bloqués par l'attente de la version finale de la PCB, nous avons utilisé une breadboard pour tester et valider les fonctionnalités de notre firmware.

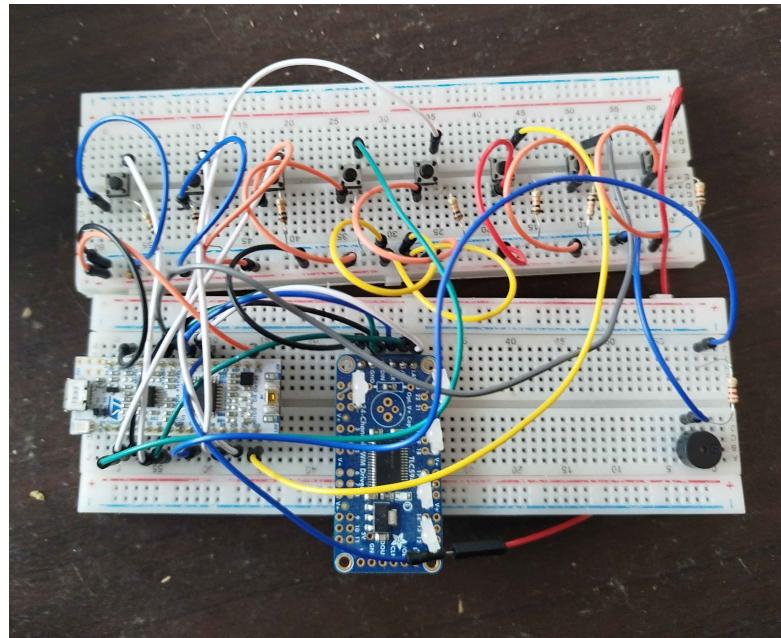


FIGURE 4 – Super Simon sur breadboard

Pour le contrôle des LEDs, nous n'avons trouvé aucune bibliothèque existante pour le TLC5947 compatible avec notre environnement, uniquement une petite bibliothèque Arduino développée par Adafruit. Nous avons donc commencé le développement du firmware en créant notre propre bibliothèque pour contrôler les huit LEDs RGB.

L'implémentation du firmware a été conçue de manière modulaire afin de pouvoir ajouter facilement de nouveaux modes de jeu. Chaque mode de jeu suit la même structure et intègre une fonction qui gère la logique du mode de jeu pour la génération de la séquence suivante. Une fois que nous avions posé la structure du firmware, nous avons pu commencer le développement des différents modes de jeu, en commençant par le mode classique.

Le jeu propose plusieurs niveaux de difficulté, influençant la vitesse d'affichage des séquences lumineuses. Nous avons défini un temps d'affichage des LEDs qui diminue progressivement lorsque la séquence augmente.

Enfin, nous avons finis le développement de notre firmware par le code du buzzer, permettant de jouer une tonalité précise ou une mélodie composée de différentes tonalités avec différentes durées.

### 3.5 Problèmes rencontrés

Durant ce projet, nous avons naturellement rencontré plusieurs problèmes ayant entraîné des retards et de nouvelles difficultés pour notre PFE. Lors du rendu de fin de S8, nous avions présenté une PCB soudée, mais nous n'avions pas encore pu y flasher de code. Cependant il s'est avéré que cette PCB était défectueuse, ce qui nous a empêchés de tester notre code avant l'acquisition d'un dev kit, puis, plus tard, de concevoir une seconde version de notre PCB.

Le plus gros obstacle que nous avons rencontré concernait l'écran que nous souhaitions utiliser. Nous avons d'abord testé une variété de bibliothèques, cartes et écrans pour tenter de le faire fonctionner notre projet, mais seuls certains essais ont été concluants, et uniquement avec une carte Arduino. Ne souhaitant pas utiliser cette carte dans notre PFE, nous avons essayé d'analyser les données envoyées en I2C à l'écran afin de les reproduire avec une carte plus adaptée. Cependant malgré nos efforts, nous n'avons pas réussi à générer les signaux attendus. Ce problème nous a contraints à reporter plusieurs fonctionnalités que nous aurions souhaité intégrer, telles que le tableau des scores, des modes de jeu dépendants de l'écran, un chronomètre et bien d'autres. Nous avons également dû adapter certains aspects

essentiels du projet, comme le choix du mode de jeu, désormais représenté par un code couleur.

Enfin, une autre difficulté est survenue lors de l'utilisation temporaire de breadboard, en raison de la présence de faux-contacts. Le circuit devenant de plus en plus complexe au fil des tests, il devenait difficile de déterminer si une erreur provenait de notre code, d'une mauvaise utilisation du dev kit, ou d'un simple faux contact entre la LED et le kit.

## 4 Test et validation

Afin de tester et valider notre Super Simon, nous avons mis en place un plan de test, que vous pouvez retrouver avec les livrables rendus. Ce plan de test comporte dix scénarios différents permettant de vérifier l'ensemble des fonctionnalités que nous avons développées pour ce projet. Nous avons également créé quatre tests spécifiques pour tester notre Super Simon sur breadboard.

Lors de notre dernière session de test, nous avons validé 11 tests, partiellement validé 2 tests et nous n'avons pas validé un test. Les tests non validés sont dus à des fonctionnalités que nous n'avons pas eu le temps de terminer d'implémenter. Vous pouvez retrouver un rapport plus détaillé de ces tests dans les livrables rendus.

Cependant, il nous manque des tests unitaires permettant de vérifier différentes parties de notre firmware indépendamment les unes des autres. Ces tests nous permettraient de nous assurer qu'après l'ajout d'une nouvelle fonctionnalité, nous n'avons pas modifié le bon fonctionnement d'autres parties du firmware.



FIGURE 5 – Version finale du Super Simon

## 5 Perspectives d'évolution

Dans l'optique d'améliorer notre projet, plusieurs ajouts ont été pensé. Dans un premier temps, un écran augmenterait la compréhension des informations envoyées à l'utilisateur : La victoire ou la défaite, le choix du mode de jeu et de la difficulté, ainsi que des aides ou des informations plus facile à transmettre par écrit allant d'un tableau des scores à un tutoriel.

Un autre point intéressant à ajouter serait des modes de jeu dépendant de l'écran. Par exemple un mode de jeu dans lequel les couleurs au lieu d'être annoncées avec les LED le serait sur l'écran, dans une autre couleur afin de piéger le joueur.

Le dernier ajout auquel nous avons pensé est la possibilité de modifier le code de façon simplifié, par le biais de mise à jour. L'ajout d'un système externe au boîtier afin d'éviter une connexion directe à la Nucléo, permettrait de ne pas avoir à démonter le boîtier lors d'une modification du code.

## 6 Conclusion

En conclusion, ce PFE nous a permis de revisiter le jeu du Simon en concevant et en réalisant notre Super Simon avec des fonctionnalités et modes de jeu originaux. Nous avons réalisé ce projet de la définition des spécifications à la conception et la fabrication de la carte électronique, en passant par l'implémentation hardware et software.

Malgré les contraintes budgétaires et les problèmes techniques rencontrés, nous avons réussi à concevoir un prototype fonctionnel. Les différents modes de jeu implémentés, tels que le mode classique, le mode 8 couleurs, le mode fantôme, le mode Color Switch, le mode partie rapide et le mode duel, sont fonctionnels.

Ce projet nous a permis de mettre en pratique nos connaissances en électronique, en programmation embarquée et en conception de PCB. Ce projet nous a appris à travailler en équipe sur un projet de longue durée, avec la réalisation de toutes les étapes d'un tel projet.

Bien que notre Super Simon soit aujourd'hui un prototype fonctionnel, plusieurs améliorations pourraient être envisagées, comme l'ajout d'un écran ou de nouveaux modes de jeu. Il serait même possible d'en développer une version proche de la commercialisation.

# Glossaire

**Breadboard** : Plaque d'essai en français, est une plaque plastique avec des trous alignés horizontalement et verticalement. Elle permet de prototyper des circuits électroniques sans avoir à souder les composants. [6](#), [13](#)

**Buzzer** : Composant électronique qui produit un signal sonore lorsqu'il est alimenté en courant électrique. [5](#), [13](#)

**CMS** : Composant Monté en Surface. Il s'agit d'un type de composant électronique conçu pour être monté directement sur la surface d'un circuit imprimé (PCB). [8](#), [13](#)

**Dev Kit** : Ensemble de matériel et de logiciels fourni par les fabricants pour faciliter le développement de nouveaux produits électroniques. [4](#), [13](#)

**Driver** : Désigne un circuit intégré conçu pour contrôler et gérer les LEDs connectées. [3](#), [5](#), [13](#)

**Empreinte** : Désigne le dessin ou le modèle physique d'une composante électronique tel qu'il doit être disposé sur une carte de circuit imprimé. Cela inclut les dimensions exactes, l'espacement des pins, les trous de montage, et d'autres détails qui permettent à la composante d'être correctement montée et soudée sur la carte PCB. Chaque composant électronique a généralement une empreinte spécifique qui lui est associée, conçue pour s'adapter à ses dimensions et à ses spécifications de montage. [5](#), [13](#)

**Firmware** : Logiciel embarqué dans un matériel électronique. [8](#), [13](#)

**Hardware** : Composants physiques d'un système informatique ou électronique. [3](#), [13](#)

**I2C** : Inter-Integrated Circuit, un protocole de communication série à faible coût et basse vitesse, utilisé pour la communication entre microcontrôleurs et périphériques. I2C permet la communication bidirectionnelle avec seulement deux fils : un pour les données (SDA) et un pour l'horloge (SCL). [9](#), [13](#)

**KiCad** : Logiciel de conception assistée par ordinateur open-source utilisé pour la conception de circuits électroniques et de cartes de circuit imprimé (PCB). [8](#), [13](#)

**LED** : Acronyme anglais de "Light-Emitting Diode", ce qui signifie en français "diode électroluminescente". C'est un composant électronique qui émet de la lumière lorsqu'il est traversé par un courant électrique. [3](#), [13](#)

**Manche** : Représente un cycle complet de jeu où le Simon affiche une séquence de couleurs et sons que le joueur doit reproduire. [3](#), [13](#)

**Microcontrôleur** : Petit composant électronique intégrant un processeur, de la mémoire et des interfaces de communication. Il est conçu pour contrôler des appareils et des systèmes, souvent utilisé dans des applications embarquées. [3](#), [5](#), [13](#)

**Nappe** : Ensemble de fils plats et parallèles maintenus ensemble. [5](#), [13](#)

**PCB** : (Printed Circuit Boards) Plaques utilisés pour fournir un support mécanique et électrique aux composants électroniques, en permettant la connexion et la transmission des signaux électriques entre eux de manière fiable et efficace. Permettant ainsi la fabrication de divers dispositifs électroniques, allant des smartphones aux ordinateurs en passant par les équipements industriels et domestiques. [3](#), [13](#)

**PFE** : Projet de fin d'études. [3](#), [13](#)

**Pin** : Désigne une petite tige métallique ou une patte qui dépasse d'un composant électronique ou d'un connecteur. Les pins sont utilisés pour établir des connexions électriques entre différents composants ou entre un composant et un circuit imprimé. [5](#), [13](#)

**Potentiomètre** : Composant électronique ajustable qui permet de régler la résistance électrique dans un circuit. Dans notre cas il permettra de régler le son. [3](#), [13](#)

**RGB** : Système de couleur utilisé dans l'affichage numérique et sur les écrans. Il se compose de trois couleurs de base : rouge (Red), vert (Green), et bleu (Blue), combinées en différentes intensités pour créer une large gamme de couleurs. [5](#), [13](#)

**Software** : (logiciel en français) Désigne les programmes informatiques et les instructions qui permettent à un ordinateur ou à un système électronique de fonctionner. Il englobe les applications, les systèmes d'exploitation, et tout autre code exécutable qui contrôle le matériel. [3](#), [13](#)

**SPI** Protocole de communication série synchrone utilisé pour échanger des données entre un microcontrôleur et divers périphériques comme des capteurs ou des mémoires.. [3](#), [13](#)

**ST-Link** Outil de débogage et de programmation pour les microcontrôleurs STM32, intégré dans les cartes STM32 Nucleo. [7](#), [13](#)

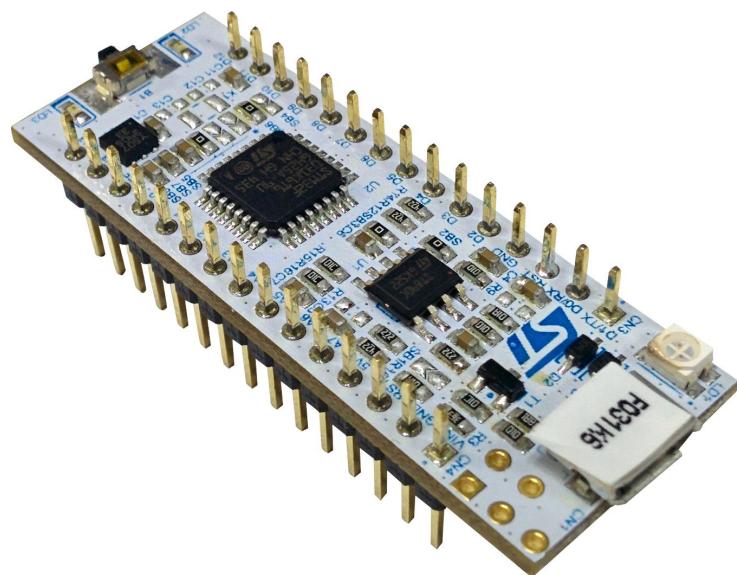
**Stencil** : Un stencil est un pochoir permettant de déposer uniformément la pâte à braser sur un circuit imprimé (PCB) avant l'assemblage des composants CMS. [4](#), [13](#)

**Switches** : Interrupteurs physiques utilisés dans les claviers d'ordinateurs et d'autres dispositifs électroniques. [13](#)

## Annexes



ANNEXE 1 – STM32F103C8T6 Blue Pill



ANNEXE 2 – STM32F303K8 Nucleo