

Conception et réalisation d'une console Space Invaders portable avec un RP2040

GEN-363-09 CAO ÉLECTRIQUE - H2026

YANN CANTIN TAMBA LEMOGO



Introduction

Space Invaders est un jeu vidéo de tir emblématique développé et publié par Taito en 1978 pour les arcades. Ce projet propose une version personnelle et moderne du classique en s'appuyant sur le projet open-source Pico Space Invaders (GitHub). L'objectif est de créer une console électronique compacte et portable qui reproduit l'expérience de jeu originale tout en intégrant des technologies actuelles: microcontrôleur RP2040, écran OLED I²C, commandes potentiométriques, buzzer et alimentation mixte USB/batterie LiPo 3,7 V.

Objectifs

- Analyser le projet de référence Pico Space Invaders (GitHub) et les documents techniques fournis.
- Concevoir le schéma électrique et le circuit imprimé à deux couches.
- Sélectionner, valider et intégrer les composants CMS de l'inventaire.
- Assembler, tester, programmer le prototype (Micro-Python) et Réaliser le boîtier.

Problématique

Comment transformer le jeu d'arcade mythique Space Invaders (1978) en une console électronique moderne, compacte et autonome, en utilisant principalement des composants CMS, un microcontrôleur RP2040, un écran OLED I²C, des commandes analogiques et une alimentation USB/batterie LiPo, tout en respectant les contraintes d'un circuit imprimé à deux couches et de l'inventaire du laboratoire ?

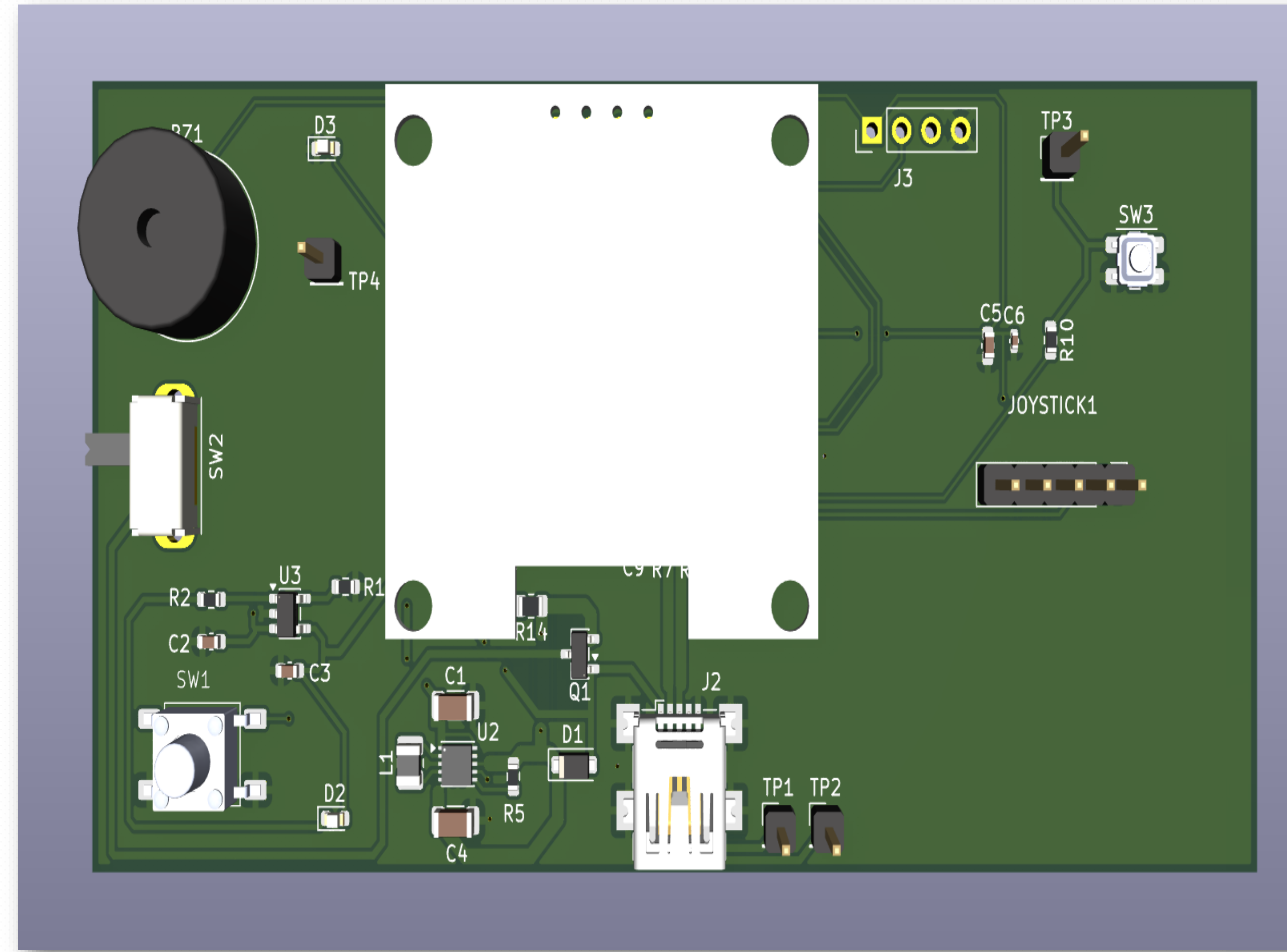


Figure 2: Schématisation 3D du PCB

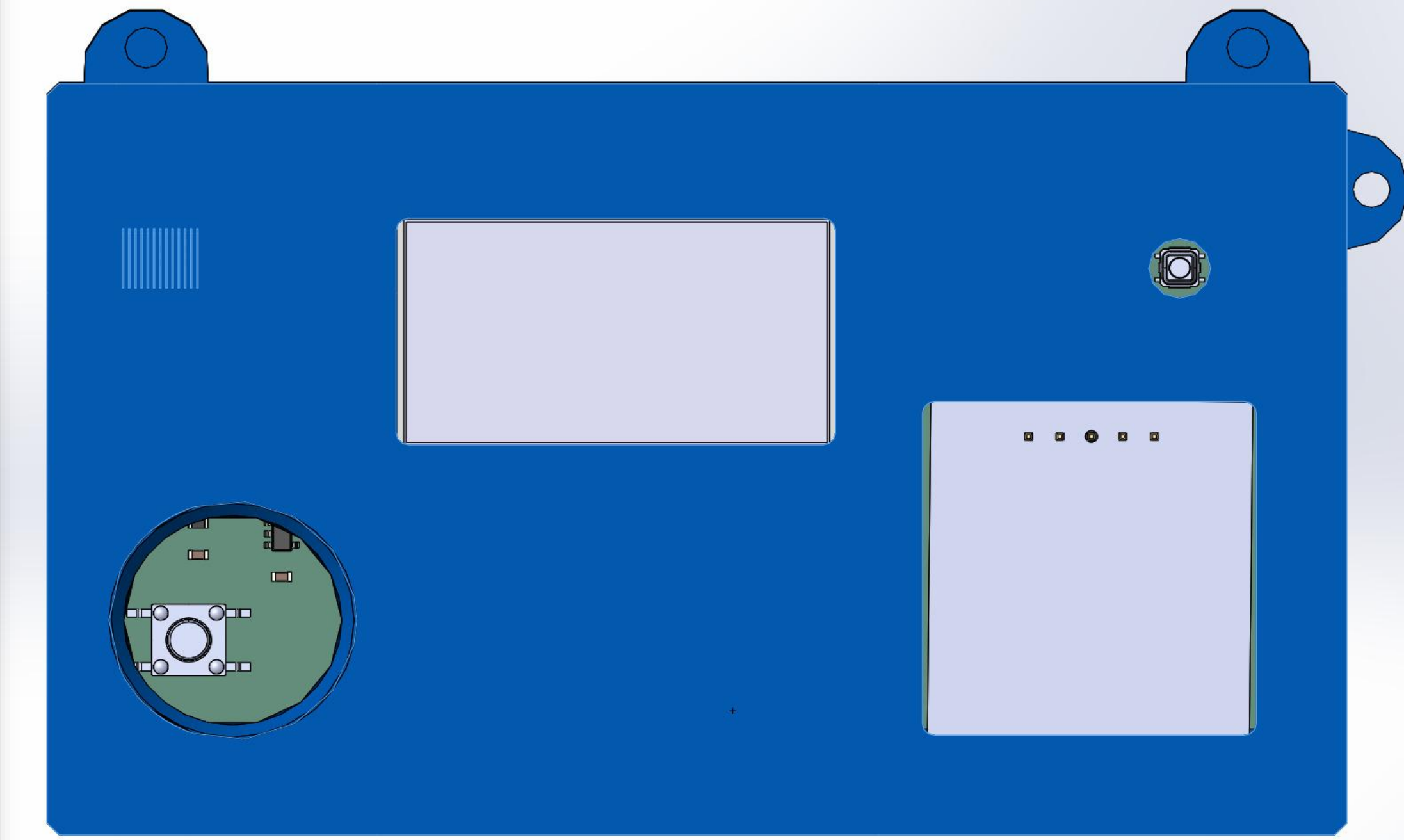


Figure 3: Boîtier

Méthodologie et concepts développés

- Analyse détaillée du projet de référence Pico Space Invaders (GitHub) et des documents techniques du Raspberry Pi Pico (datasheets, schémas, pinout et SDK Micro-Python).
- Analyse des fiches techniques des composants (TR6150AB-06, Ts21, OLED I²C, etc.)
- Sélection exclusive des composants CMS dans l'inventaire du laboratoire (RP2040, RT6150B buck-boost 3,3 V, MCP73831, W25Q16, MBR120, DMP3125, cristal 12 MHz, TS21, connecteur JST, etc.)
- Conception du schéma électrique dans KiCad avec contrôle des règles électriques (ERC) et évaluation de la consommation du circuit.
- Validation des empreintes et des modèles 3D STEP de tous les composants.
- Revue de conception du schéma.
- Conception du circuit imprimé à deux couches : placement optimisé des composants CMS, routage des traces et contrôle des règles de conception (DRC).
- Génération des fichiers de fabrication Gerber et de l'Interactive BOM.
- Assemblage manuel des composants CMS sur le PCB.
- Tests d'alimentation USB/batterie LiPo, charge de la batterie et programmation de l'amorce Micro-Python.
- Implantation et test du code du jeu Space Invaders en conditions réelles.

Bibliographie

- Raspberry Pi Pico – Hardware design, datasheet, pinout, schematic (Moodle).
- RP2040 datasheet et Pico Python SDK.
- Fiches techniques : TR6150AB-06, Ts21, OLED 1,3"/1,54" I²C, MBR120VLSFT1G, DMP3125L, MCP73831, W25Q16, etc.
- Projet de référence : Pico Space Invaders (GitHub).

Résultat

Le prototype a été assemblé avec succès et est entièrement fonctionnel.

- Le jeu Space Invaders s'affiche correctement sur l'écran OLED avec des graphismes fluides et une bonne réactivité.
- Les deux potentiomètres 10 kΩ assurent un contrôle précis du vaisseau.
- Le buzzer reproduit les effets sonores d'origine.
- L'alimentation USB et batterie LiPo 3,7 V fonctionne parfaitement (charge et commutation validées).
- Les contrôles ERC/DRC du schéma et du PCB ont été passés avec succès.
- La consommation du circuit est conforme aux attentes grâce au régulateur buck-boost RT6150B.

Suites à donner au projet:

- Améliorer les graphismes et les animations du jeu (ajout de niveaux, explosions, high-score).
- Optimiser l'autonomie de la batterie en réduisant encore la consommation.
- Concevoir et imprimer en 3D un boîtier plus ergonomique et compact.
- Ajouter des fonctionnalités avancées (Bluetooth pour scores en ligne, mode deux joueurs, son amélioré).
- Réaliser une version finale plus compacte ou intégrée dans un boîtier rétro.

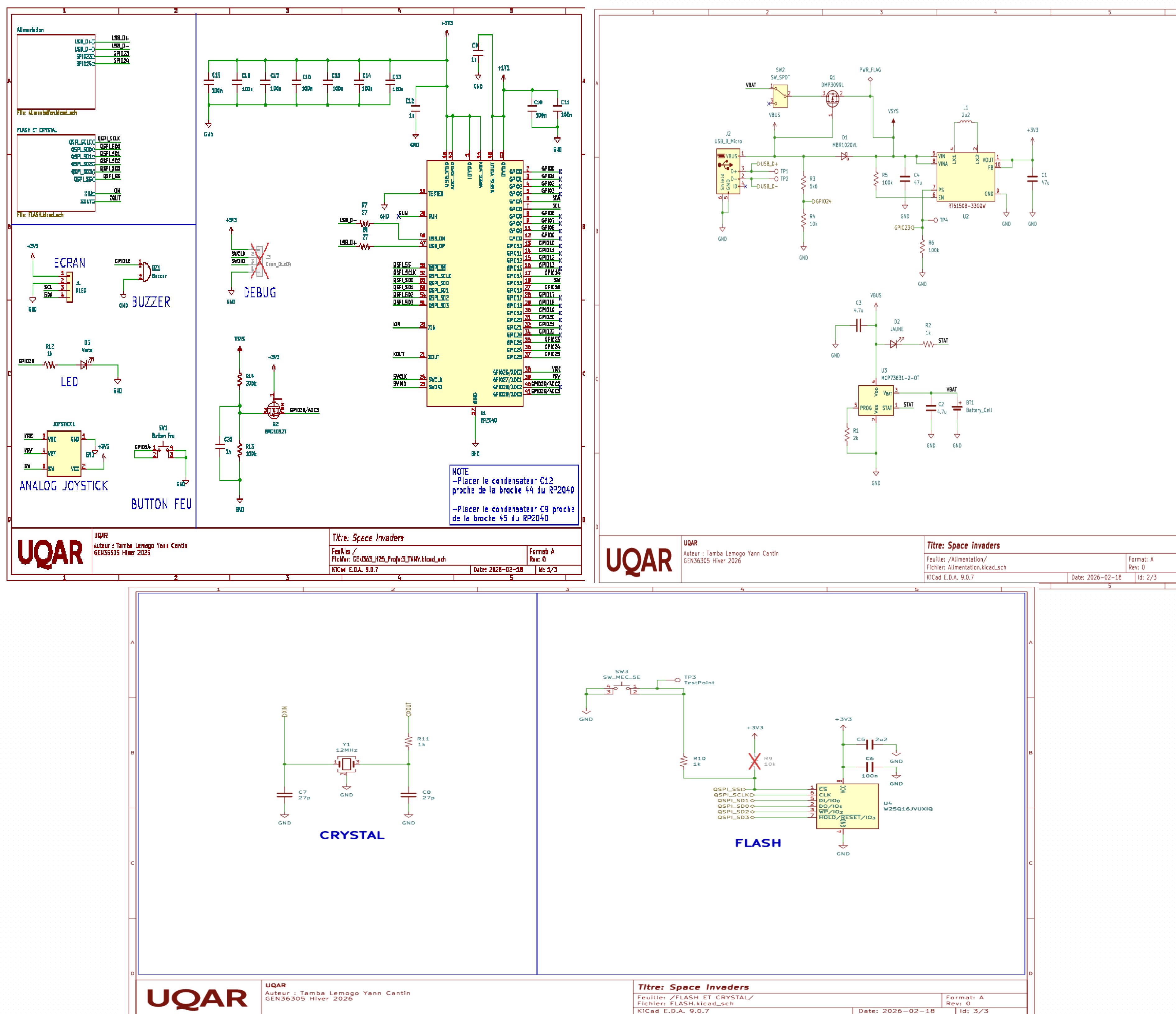


Figure 1: Circuit de conception