

In.Ra.Ci

Avenue Jupiter, 188

1190 Bruxelles

Technique de qualification

Electronique

Année scolaire 2024-2025

GAMEBOY

Achraf El Amrani

À l'issue de ce Travail de Fin d'Études, je tiens à exprimer ma profonde gratitude à toutes les personnes qui m'ont soutenu, guidé et inspiré tout au long de ce parcours.

Je remercie tout d'abord Monsieur Kapita et Monsieur Mazzeo pour leur accompagnement académique, leurs conseils éclairés et leur disponibilité constante. Leur expertise a été précieuse dans l'élaboration et la réalisation de ce travail.

Je tiens également à exprimer ma reconnaissance à Monsieur Senga, Monsieur Harmach, et Monsieur Zemraoui, dont les enseignements et les encouragements ont grandement contribué à mon développement tout au long de mes études.

Un merci particulier à Gabriel Burgho et Quentin Hordies pour leur aide et leur précieuse collaboration durant ce projet. Leur implication m'a été d'une grande aide.

Je ne saurais terminer sans adresser mes remerciements les plus sincères à mes parents. Ma mère et mon père, pour leur amour inconditionnel, leur soutien moral et matériel, ainsi que leur confiance tout au long de mon parcours.

À toutes ces personnes, je vous dis merci du fond du cœur.

Sommaire :

1. L'introduction :	4
1.1. Photo/3D :	5
2. Les caractéristiques :	8
3. Les principes mis en jeu :	9
3.1. Le principe du bus I ² C :	9
3.2. Le principe du bus SPI :	12
3.3. Le principe du bus I ² S :	13
4. Le schéma bloc :	14
4.1. Description fonctionnelle du schéma :	15
5. L'étude détaillée et les mesures	17
5.1. Schéma de principe	17
5.2. Schéma de l'écran :	18
5.4. Schéma de l'ampli class D :	36
5.5. Schéma du MCP73831/2:	42
5.6. Schéma du MCP1826 :	48
6. L'ordinogramme:	53
6.2. Programmation :	54
7.Fabrication :	55
8.Mise au point :	60
9.Conclusion :	61
Webographie :	62
Annexe :	64

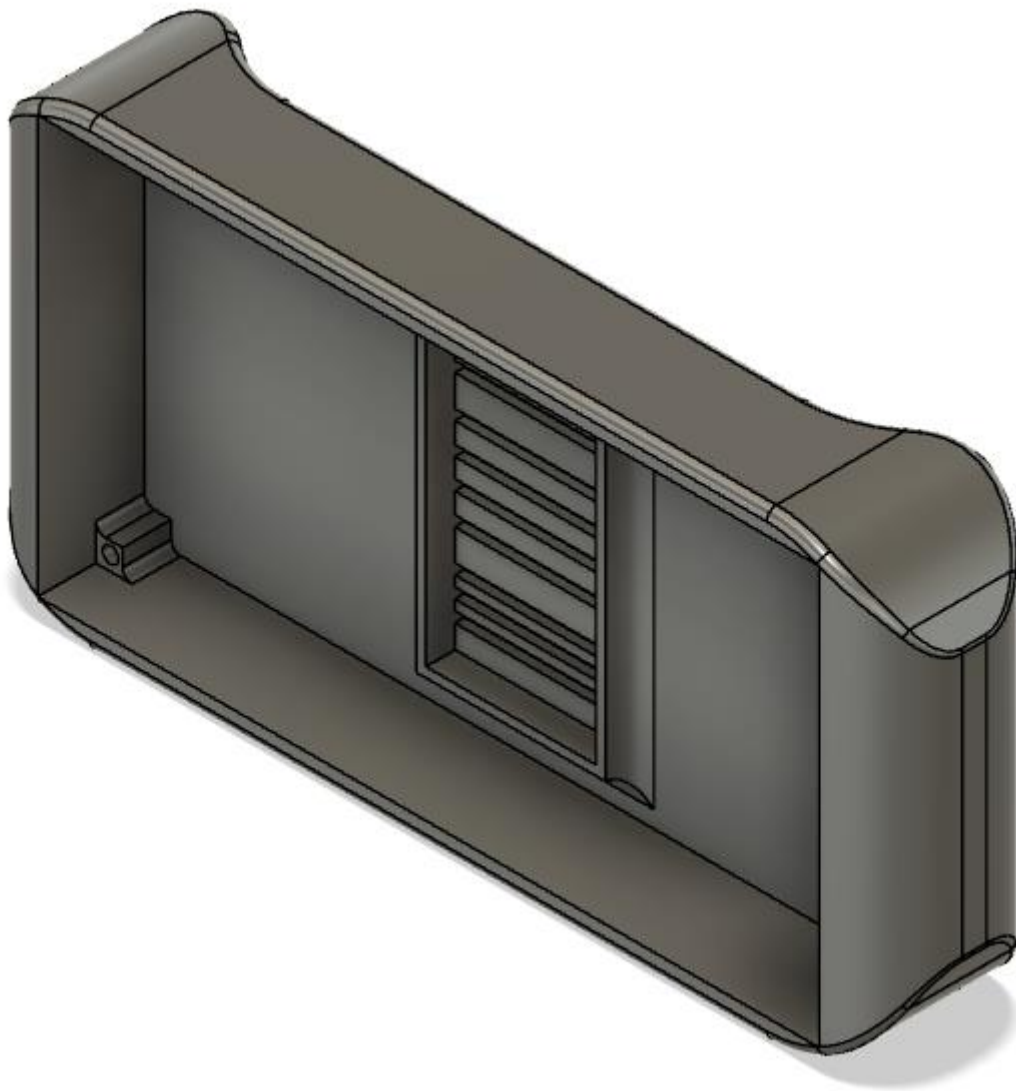
1. L'introduction :

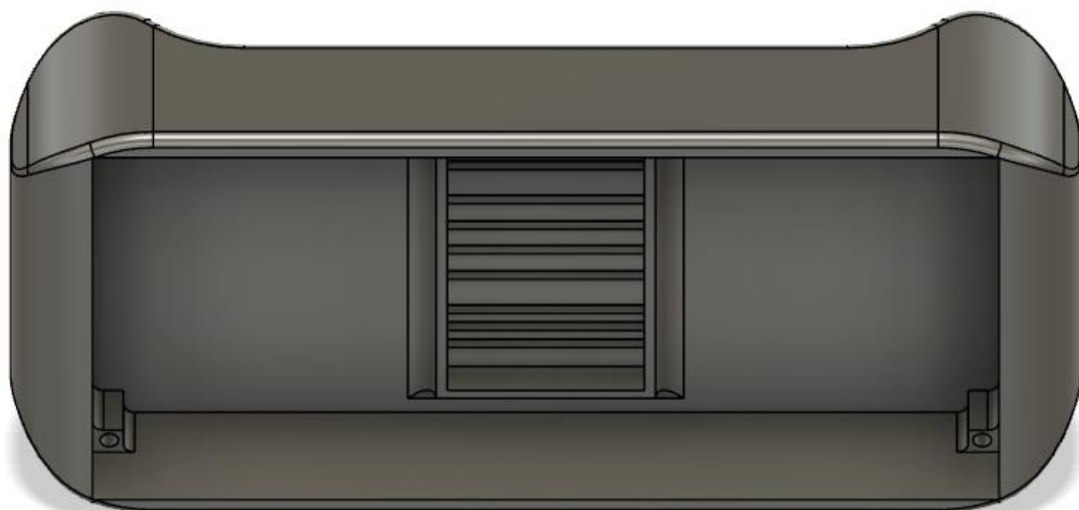
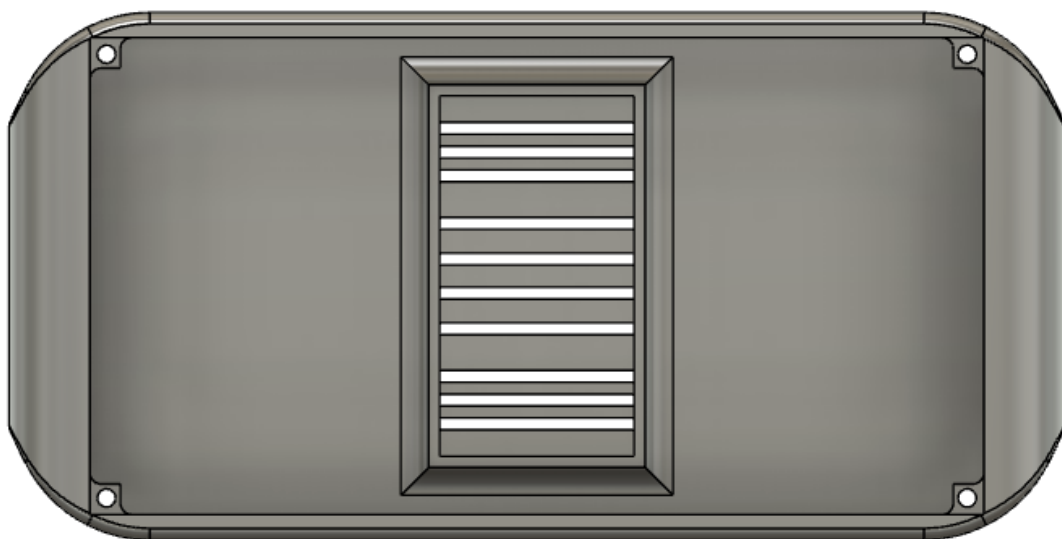
Passionné par les jeux vidéo, j'ai décidé de consacrer mon projet de fin d'études à la création d'une console de jeux portable. L'inspiration m'est venue après avoir découvert le film Tetris, qui met en lumière le célèbre jeu ainsi que la mythique Game Boy.

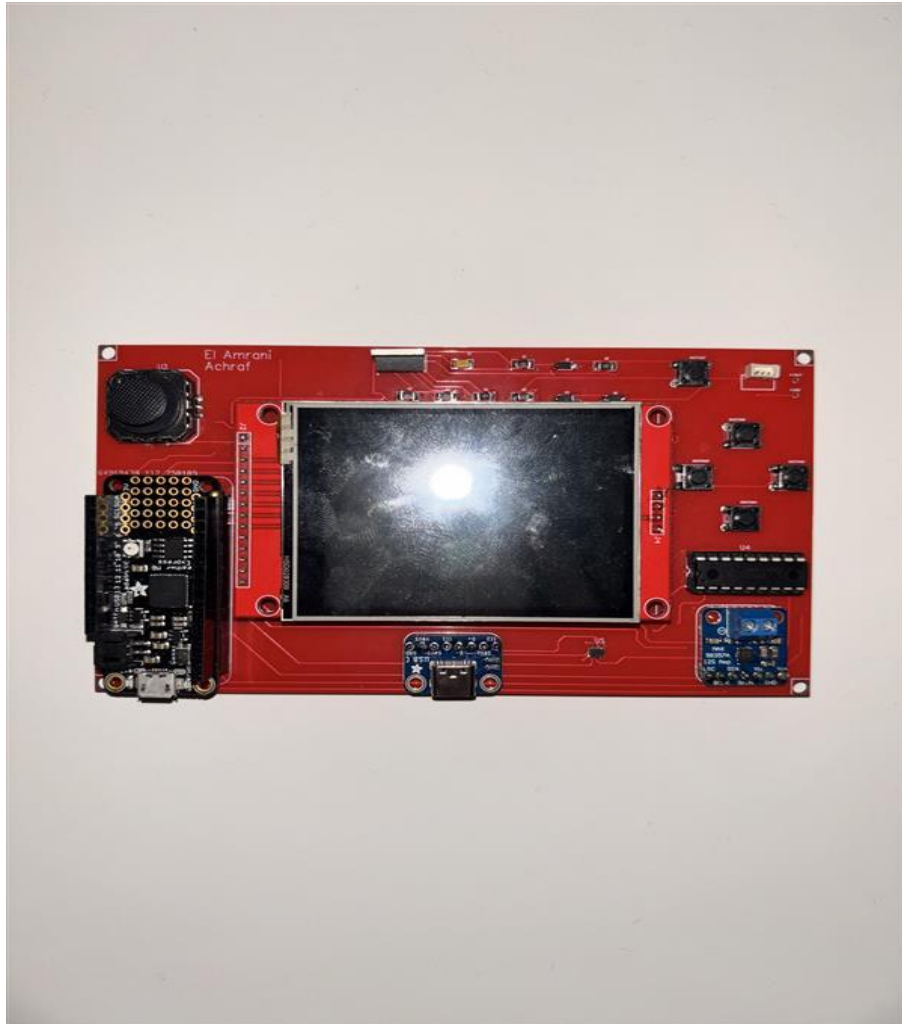
Mon objectif est de concevoir un appareil capable d'exécuter plusieurs jeux, avec Tetris comme titre principal. L'expérience de jeu sera optimisée grâce à un système de contrôle intuitif, combinant joystick et boutons physiques.

Ce projet est bien plus qu'un simple exercice technique : il représente l'aboutissement de mon parcours académique et l'opportunité d'appliquer concrètement les compétences que j'ai développées au fil des années. Il reflète mon engagement dans le domaine de l'électronique et du jeu vidéo ainsi que mon envie de relever des défis techniques stimulants

1.1. Photo/3D :







2. Les caractéristiques :

Générales :

Un écran SPI / ILI9341 TFT 2.8"

Un haut-parleur

Une entrée de charge USB C

Un Mini Joystick Analogique 2 Axes / Joystick 2765

Un interrupteur ON/OFF

4 boutons

Électriques :

Alimentation : via USB ou batterie Li-Po 3.7V

Sortie audio : Mono, puissance jusqu'à 3,2W

Tension de charge : 4.2V (pour batteries Li-Ion / Li-Po)

Fréquence SPI : Jusqu'à 40MHz, permettant un affichage fluide

Fréquence : 48 MHz du μ C

Mécanique :

Poids : +-200g

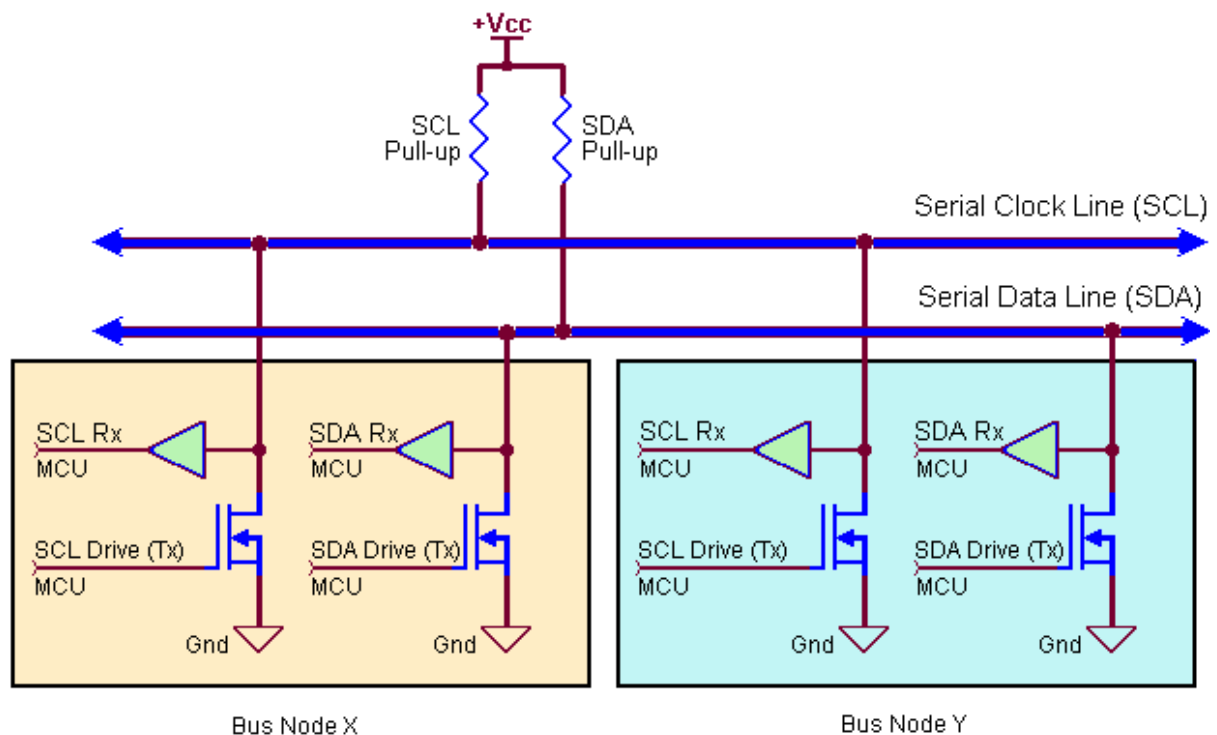
Taille : 150 x 80

Matériaux : PLA

3. Les principes mis en jeu :

3.1. Le principe du bus I²C :

Le bus I²C (Inter-Integrated Circuit) est un bus de communication série synchrone permettant à plusieurs circuits intégrés de communiquer sur deux fils, ce qui veut dire C'est un type de communication dans lequel les données sont échangées en se basant sur un signal d'horloge partagé.



IIC Bus and Interface Hardware

© Mosaic Industries, Inc., www.mosaic-industries.com/embedded-systems/

- Les appareils communiquent "au rythme" d'une horloge
- Chaque bit est envoyé au bon moment, en suivant les impulsions d'horloge
- L'horloge (clock) est un signal carré (1, 0, 1, 0...) envoyé régulièrement
- Le SCL (Serial Clock Line) est l'horloge
- Le maître envoie les impulsions de SCL
- À chaque front montant de SCL, un bit est lu ou écrit sur SDA (Serial Data Line)
- La synchronisation se fait automatiquement

Les lignes du bus I²C :

Nom	Signification	Rôle
SDA	Serial Data	Ligne de données (bidirectionnelle)
SCL	Serial Clock	Ligne d'horloge (pilotée par le maître)

Ces lignes doivent avoir des résistances de pull-up car le protocole utilise des sorties à collecteur-ouvert.

Collecteur ouvert :

Qui ne peut tirer une ligne qu'à l'état bas (0V) mais jamais à l'état haut (Vcc).

Elle agit comme un interrupteur vers la masse.

Sorties collecteur-ouvert	État logique	Ce qu'elle fait
Tiré vers le bas	0	Connecte la ligne à la masse
Relâchée	1	Laisse la ligne flotter

Elle ne fournit jamais de tension → c'est pour ça qu'on met une résistance de pull-up.

Pull-up sur les lignes SDA et SCL en I²C :

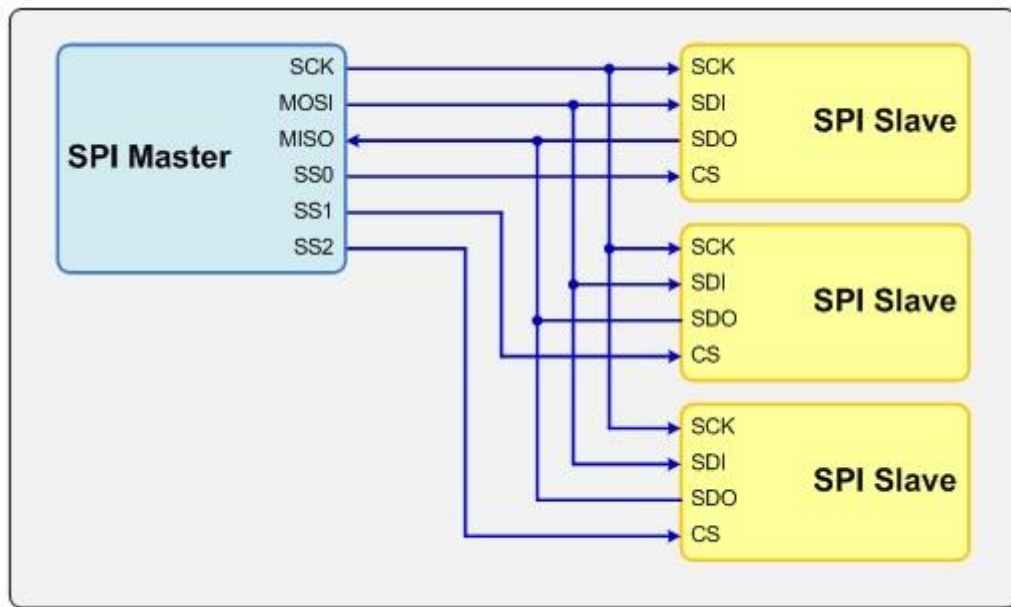
- Les périphériques ne poussent jamais les lignes à l'état haut
- Ils peuvent seulement tirer les lignes vers le bas (0V)

Du coup, si personne ne tire la ligne vers le bas, on a besoin de résistance pour ramener la ligne à l'état haut (3.3 V ou 5 V).

- Quand aucun périphérique ne communique, les lignes SDA et SCL sont tirées à VCC par les résistances de pull-up.
- Quand un périphérique veut transmettre un 0, il connecte la ligne à la masse (GND).
- Pour un 1, il relâche la ligne, et la résistance de pull-up ramène le signal à VCC.
- Sans résistances de pull-up SDA et SCL risquent de rester en état indéterminé. Le microcontrôleur peut ne pas détecter les bits correctement. Les signaux peuvent être déformés ou instables à l'oscilloscope.

3.2. Le principe du bus SPI :

Le SPI (Serial Peripheral Interface) est un protocole de communication série, synchrone et full-duplex, conçu pour permettre un échange rapide de données entre un maître et un ou plusieurs périphériques esclaves.

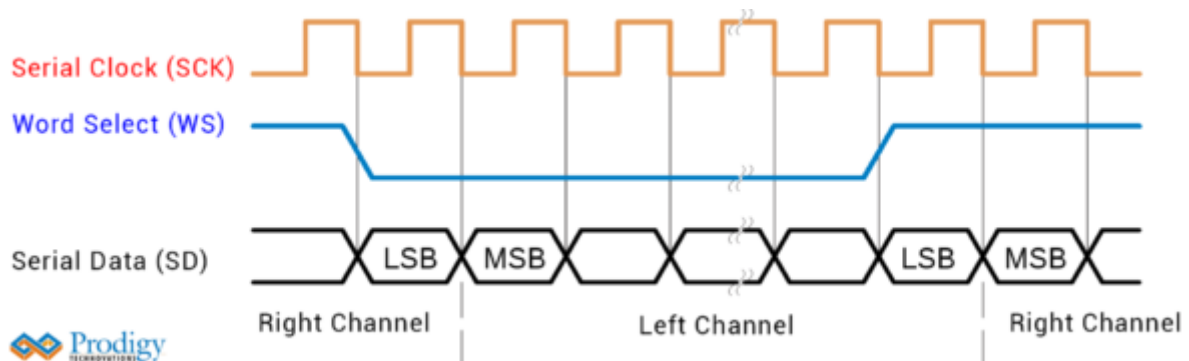


<https://www.corelis.com/images/screenshots/spi-multipleslaves.jpg>

- **Série** : Les données sont transmises bit par bit sur une ou plusieurs lignes de données, à la différence d'une communication parallèle (où plusieurs bits sont envoyés simultanément sur plusieurs fils) cela permet de réduire le nombre de fils nécessaires.
- **Synchrone** : La communication est synchronisée par une horloge (signal d'horloge appelé SCLK) générée par le maître. Cela signifie que le maître et l'esclave sont parfaitement coordonnés pour l'envoi et la réception de bits.
- **Full-duplex** : Cela signifie que le SPI permet d'envoyer et de recevoir des données simultanément. Le maître peut envoyer un octet pendant que l'esclave lui en renvoie un autre sur une ligne dédiée (MOSI pour l'envoi, MISO pour la réception).
- **Maître (Master)** : C'est l'appareil qui contrôle la communication. Il génère le signal d'horloge et sélectionne à quel esclave il souhaite communiquer.
- **Esclave (Slave)** : C'est le périphérique qui reçoit les instructions du maître et y répond. Il n'envoie rien de lui-même sans y avoir l'autorisation.

3.3. Le principe du bus I²S :

L'I²S (Inter-IC Sound) est un protocole de communication série, synchrone, spécialement conçu pour transmettre de l'audio numérique entre des composants électroniques, à l'intérieur d'un même appareil. On le retrouve souvent dans des systèmes comme les ordinateurs, smartphones, cartes son, DAC (convertisseurs numérique/analogique), ou encore des microcontrôleurs avec modules audio.



<https://prodigytechno.com/wp-content/uploads/2021/03/i2s2-600x183.png>

À quoi sert l'I²S ?

Il permet à deux puces de s'échanger des données audios numériques avec précision et en temps réel. L'I²S est optimisé uniquement pour le son, avec une gestion spéciale du timing des échantillons audio.

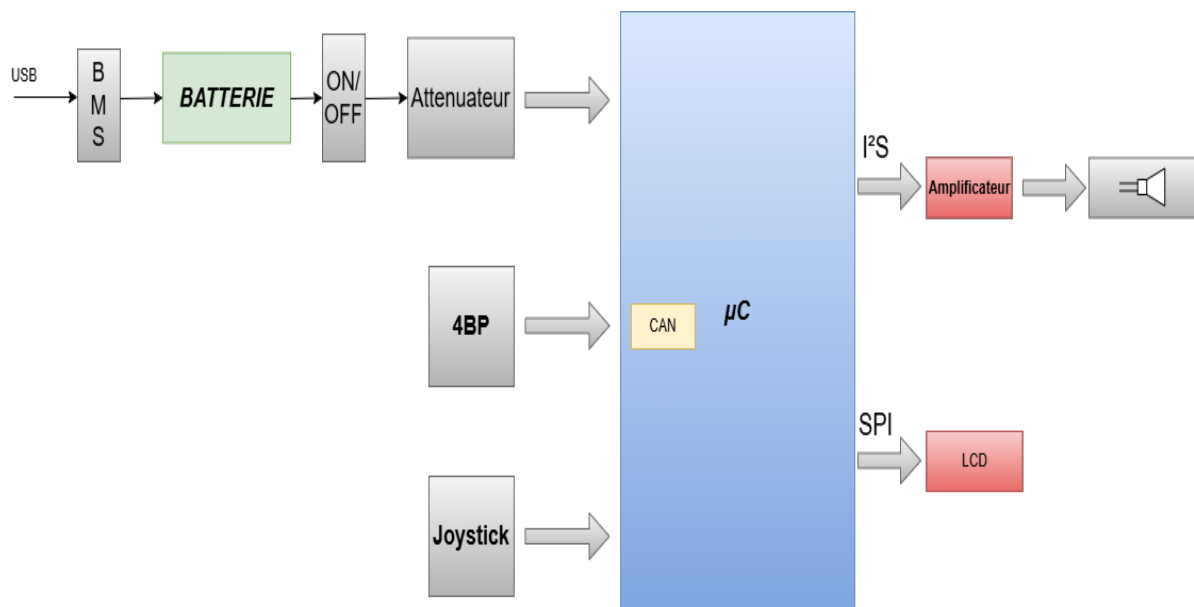
Rôles : Maître / Esclave

- Le maître génère les horloges et contrôle le rythme de transmission.
- L'esclave reçoit les données et les horloges pour reconstruire le son.

Les données sont transmises :

- **En série** : les bits sont envoyés un à un.
- **En synchrone** : tout est cadencé par un signal d'horloge.
- **En MSB First** : le Bit de Poids Fort (Most Significant Bit) est transmis en premier pour la bonne reconstruction du signal audio.

4. Le schéma bloc :



4.1. Description fonctionnelle du schéma :

Bloc BATTERIE :

Fonction : Fournit l'énergie électrique nécessaire au système.

Type de signal : Tension continue (DC).

Bloc USB 5V :

Fonction : Convertit la tension de la batterie en une tension stabilisée de 5V.

Bloc +BMS (Battery Management System) :

Fonction : Assure la protection, la gestion de charge/décharge et la sécurité de la batterie.

Bloc 4BP (4 boutons poussoirs) :

Fonction : Permet d'envoyer des commandes.

Type de signal : Signal logique (état haut/bas selon l'appui sur les boutons).

Bloc Joystick :

Fonction : Permet un contrôle directionnel ou analogique par l'utilisateur.

Type de signal : Signal analogique ou numérique selon le déplacement.

Bloc ON/OFF :

Fonction : Permet d'allumer ou d'éteindre le système.

Type de signal : Signal logique (interrupteur ouvert ou fermé).

Bloc μ P (microprocesseur) :

Fonction : Cœur du système. Il traite les signaux reçus, exécute des instructions stockées en mémoire et pilote les sorties audio et visuelles.

Type de signal :

- Entrée : signaux logiques/analogiques des commandes ;
- Sortie : signaux de commande vers les modules de sortie (audio/visuel).

Bloc Amplificateur :

Fonction : Amplifie le signal audio envoyer par la μ P pour piloter le haut-parleur.

Type de signal :

- Entrée : signal audio faible.
- Sortie : signal audio amplifié.

Bloc Haut-parleur :

Fonction : Ressort le son demander.

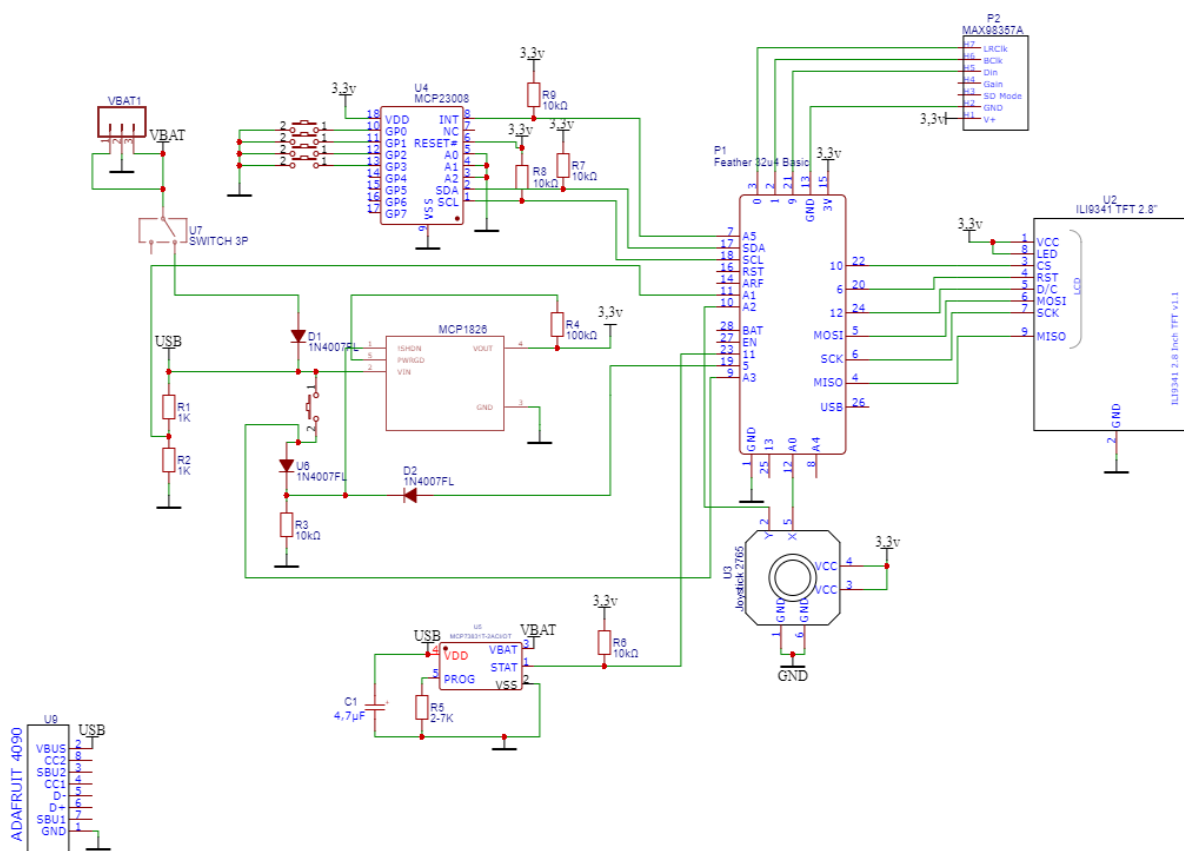
Bloc LCD :

Fonction : Affiche des informations visuelles.

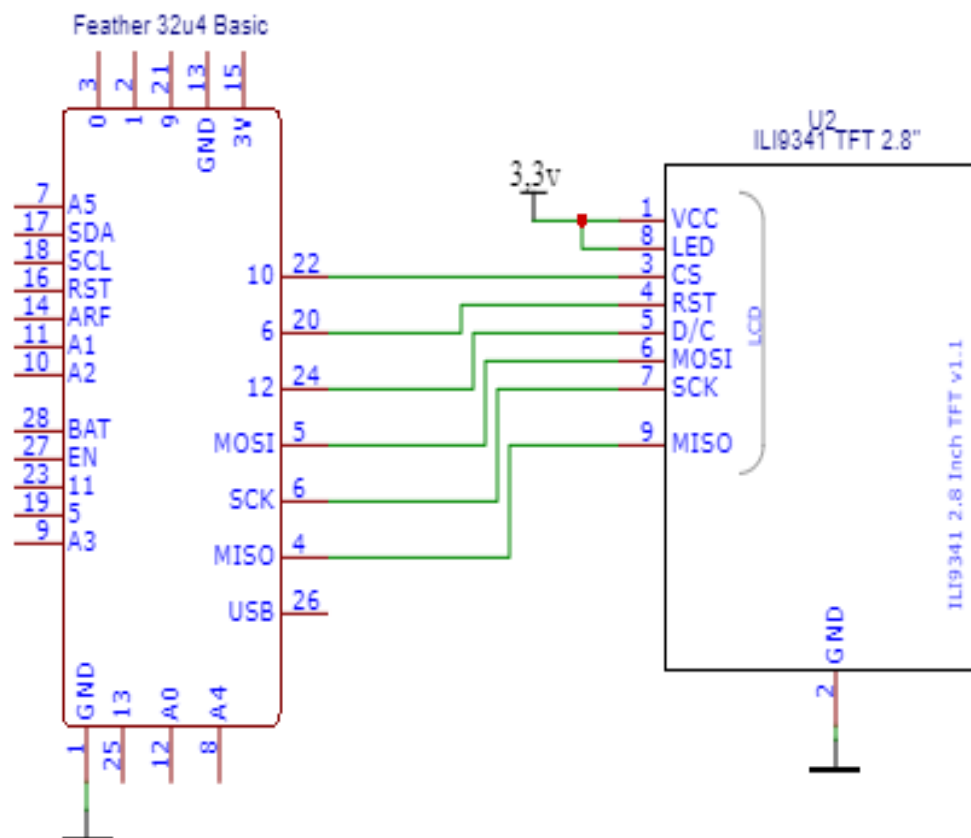
Type de signal : Signal série.

5. L'étude détaillée et les mesures

5.1. Schéma de principe



5.2. Schéma de l'écran :



Présentation du ILI9341 :



<https://fr.led-diode.com/uploads/202220221/small/2-8-inch-ili9341-240x320-spi-tft-lcd-display16375845236.jpg>

L'ILI9341 est une puce contrôleur pour écran TFT LCD de 240 x 320 pixels capable d'afficher 262 144 couleurs (format RGB 18 bits).

Elle intègre les circuits nécessaires pour piloter l'écran ainsi qu'une mémoire interne (GRAM) de 172 800 octets permettant de stocker l'image affichée.

L'ILI9341 permet aussi de définir une zone spécifique de l'écran à mettre à jour, ce qui est utile pour animer une partie de l'image sans rafraîchir l'ensemble de l'écran.

Elle fonctionne avec des tensions d'interface comprises entre 1,65 V et 3,3 V grâce à un régulateur intégré, et propose un mode veille pour réduire la consommation d'énergie.

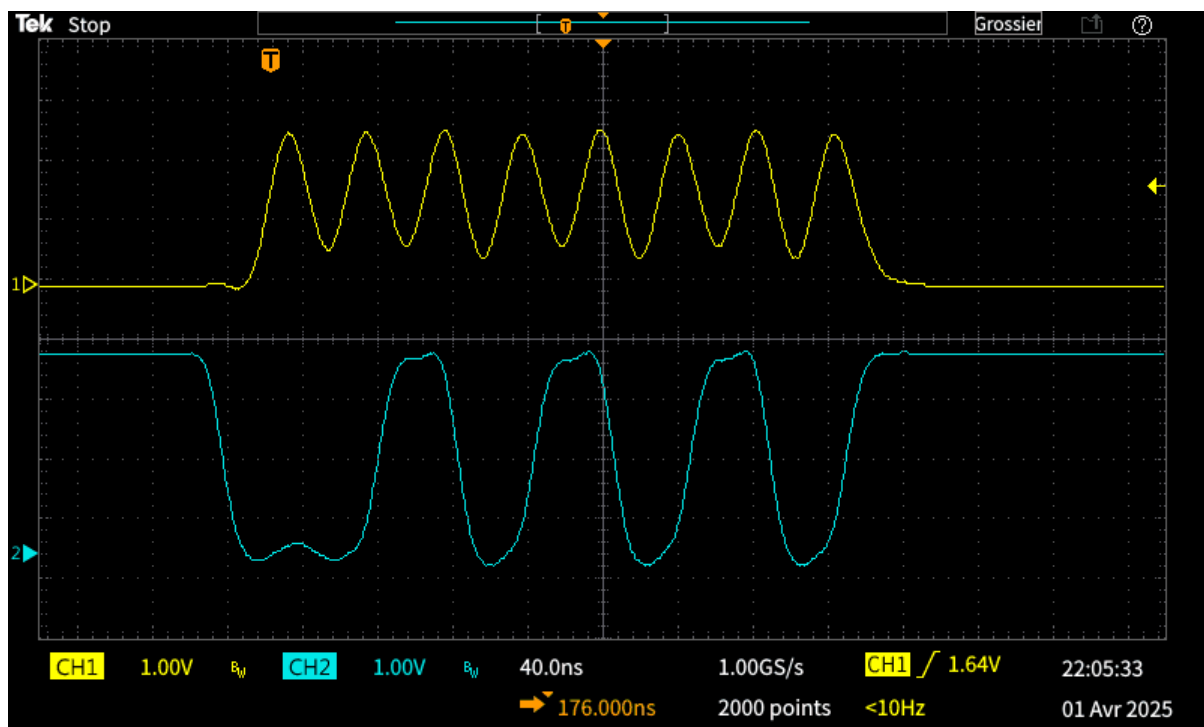
Rôles et lignes de communication :

- **Maître (Master)** : C'est l'appareil qui contrôle la communication. Il génère le signal d'horloge et sélectionne quel esclave il souhaite communiquer.
- **Esclave (Slave)** : C'est le périphérique qui reçoit les instructions du maître et y répond. Il n'envoie rien de lui-même sans y avoir l'autorisation.
- **SCLK (Serial Clock)** : Ligne d'horloge générée par le maître pour cadencer l'échange des bits.
- **MOSI (Master Out Slave In)** : Ligne de données sur laquelle le maître envoie des informations vers l'esclave.
- **MISO (Master In Slave Out)** : Ligne utilisée par l'esclave pour répondre au maître.
- **CS (Chip Select)** ou **SS (Slave Select)** : Ligne dédiée pour sélectionner un esclave spécifique. Lorsqu'elle est à l'état bas (LOW), le périphérique est activé pour la communication.
- **VCC** : Alimentation positive pour l'écran
- **GND** : Masse
- **RESET** : Réinitialisation de l'écran.
- **DC (Data/Command)** : Indique si l'information transmise est une donnée (1) ou une commande (0).
- **LED** : Commande le rétroéclairage. Souvent connecté à une alimentation ou à un signal PWM pour ajuster la luminosité.

Programme de test :

```
tft.begin();    // Initialise l'écran
tft.setRotation(1); // Définissez l'orientation de l'écran
// Efface l'écran avec une couleur de fond
tft.fillScreen(ILI9341_BLACK);
// Afficher du texte
tft.setTextColor(ILI9341_WHITE);
tft.setTextSize(2);
tft.setCursor(10, 10);
tft.println("Hello,World!");
// Dessinez quelques formes
tft.fillRect(10, 30, 100, 50, ILI9341_RED);    // Rectangle rouge
}
void loop() {
}
```

Mesure :



Analyse de l'image :

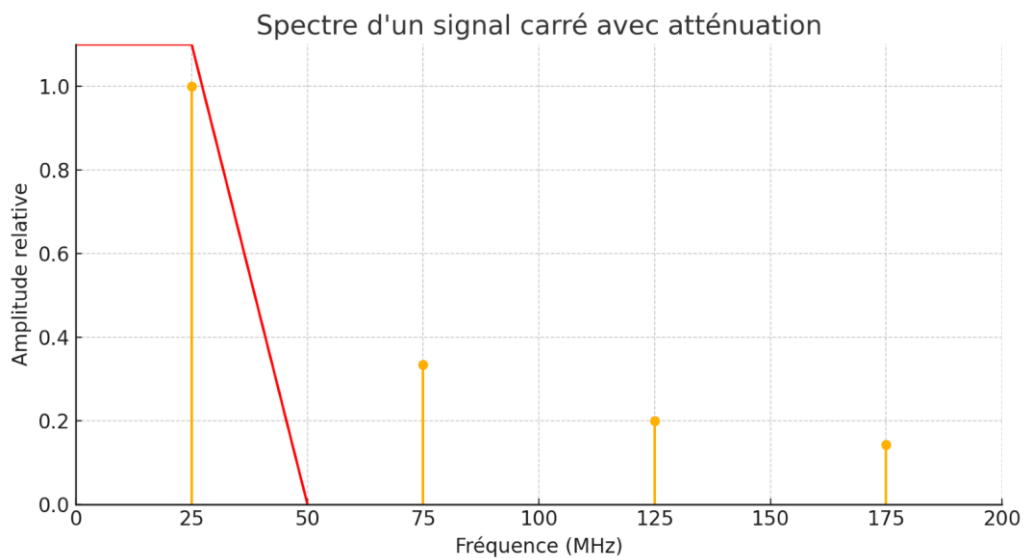
- **Jaune (CH1) :** Signal SCK (horloge)
- **Bleu (CH2) :** Signal MOSI (données)

Sur l'oscilloscope, on remarque que les signaux ont une forme arrondie, proche d'une sinusoïde, alors qu'on s'attend à voir un signal carré.

Pourquoi cette forme sinusoïdale ?

Cela est dû à une limitation de la bande passante de la sonde et de l'oscilloscope.

- La fréquence du signal est de 25 MHz (période = 40 ns).
- Une sonde avec une bande passante de 50 MHz ne peut pas capturer correctement toutes les composantes du signal carré.
- Un signal carré est composé de la fréquence fondamentale (25 MHz) et de ses harmoniques impairs (75 MHz, 125 MHz, etc.).
- Comme la bande passante de la sonde et de l'oscilloscope est limitée à 50 MHz, elle ne laisse passer que la fondamentale (25 MHz) et filtre les harmoniques.
- Résultat : le signal observé est lissé, il ressemble à une sinusoïde au lieu d'un carré.

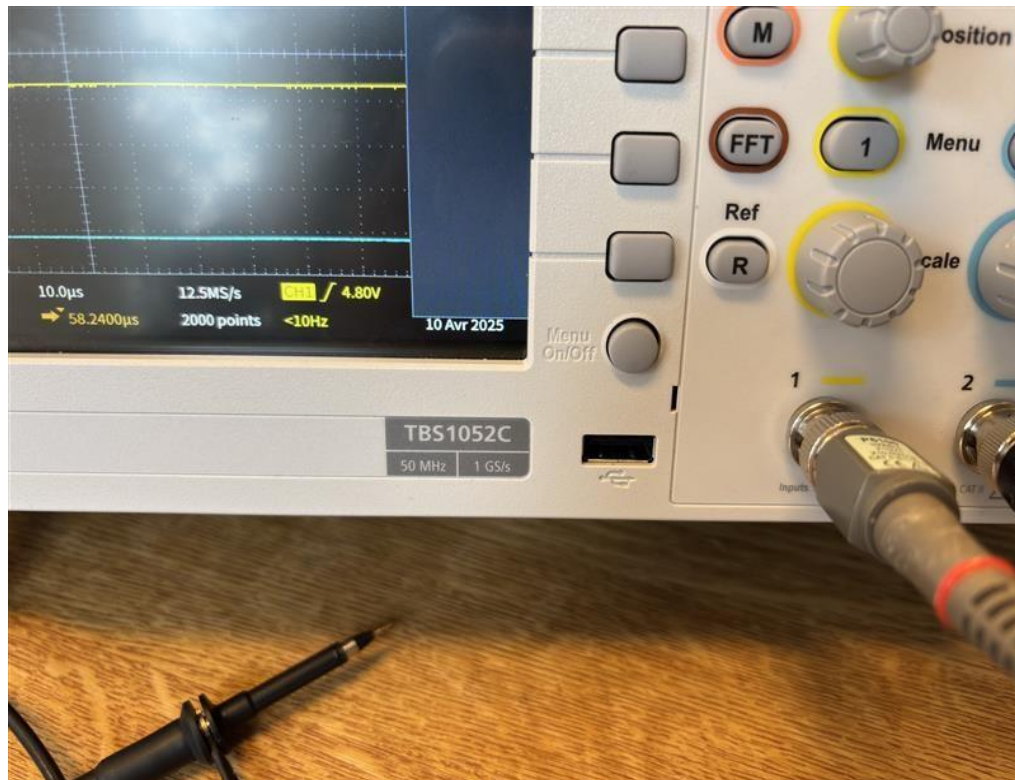


Conséquences

- On ne voit pas la vraie forme du signal.

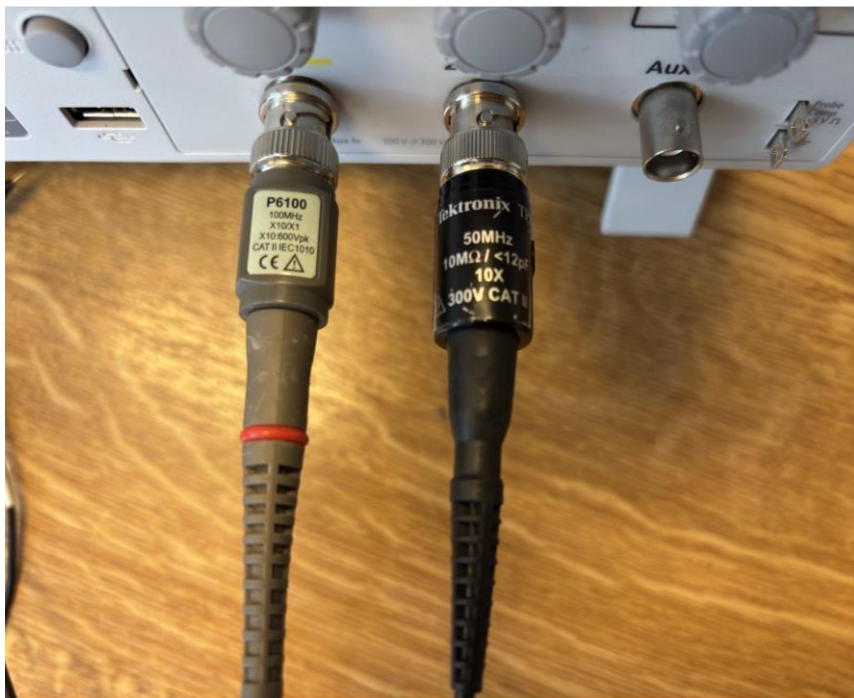
Information sur le matérielle :

Oscilloscope :



Fréquence max : 50MHz

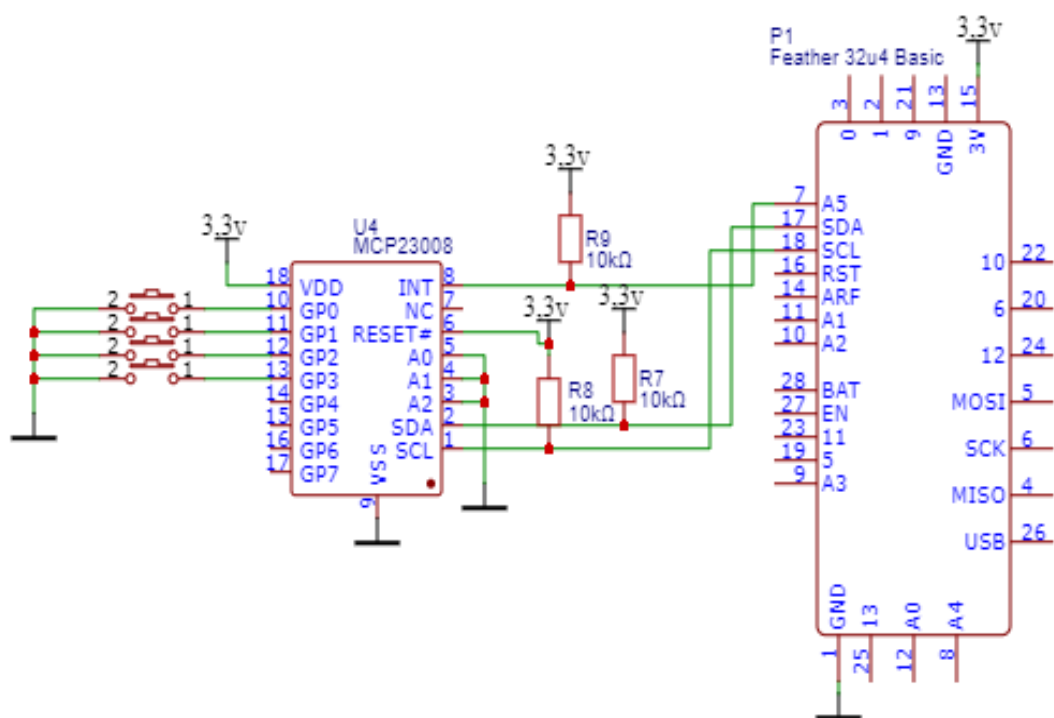
Sonde :



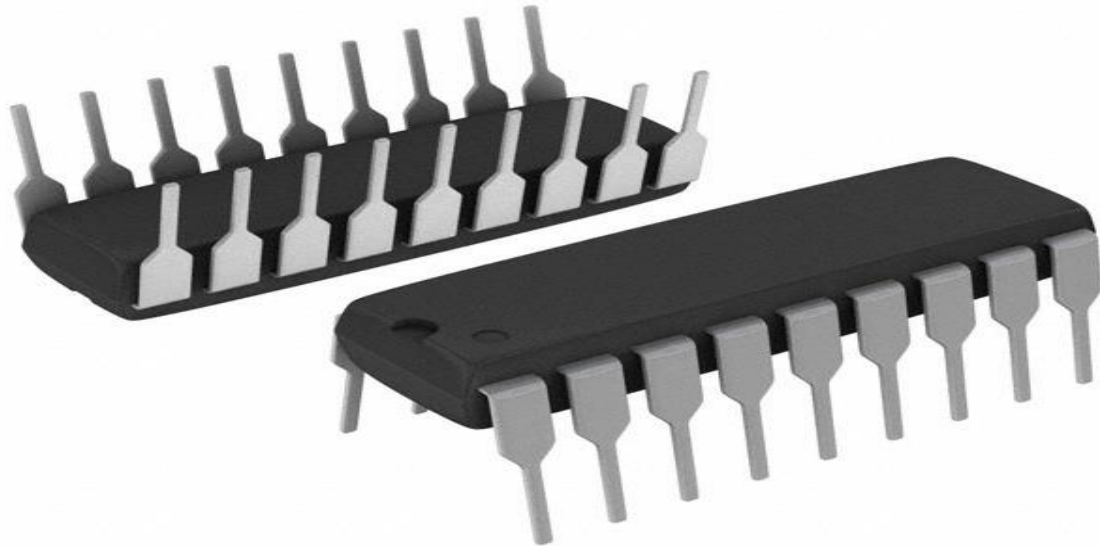
Sonde 1 : 100MHz

Sonde 2 : 50MHz

5.3. Schéma du MCP23008 :



Présentation du MCP23008 :



<https://mm.digikey.com/Volume0/opasdata/d220001/medias/images/376/18-Dip.jpg>

Le MCP23008 est un circuit intégré d'extension d'entrées/sorties (I/O expander) conçu par Microchip.

Il permet d'ajouter jusqu'à 8 ports d'E/S numériques supplémentaires à un microcontrôleur via une interface I²C (bus série à deux fils).

Il est compact et facile à utiliser, le MCP23008 est particulièrement utile lorsque le nombre de broches disponibles sur un microcontrôleur est insuffisant pour gérer tous les périphériques.

Rôles des pattes :

- **VDD** : Alimentation (+3.3 V ou +5 V selon le système)
- **GPA0–GPA7** : 8 GPIO programmables (Entrées ou sorties numériques)
- **INT** : Interruption (sortie) → devient LOW si un événement GPIO se produit
- **NC** : Non connectée (Not Connected)
- **RESET** : Réinitialisation active à l'état bas
- **A0** : Bit 0 de l'adresse I²C
- **A1** : Bit 1 de l'adresse I²C
- **A2** : Bit 2 de l'adresse I²C
- **SDA** : Données I²C – échanges de bits avec le maître
- **SCL** : Horloge I²C – reçoit les impulsions du maître
- **VSS** : Masse (GND)

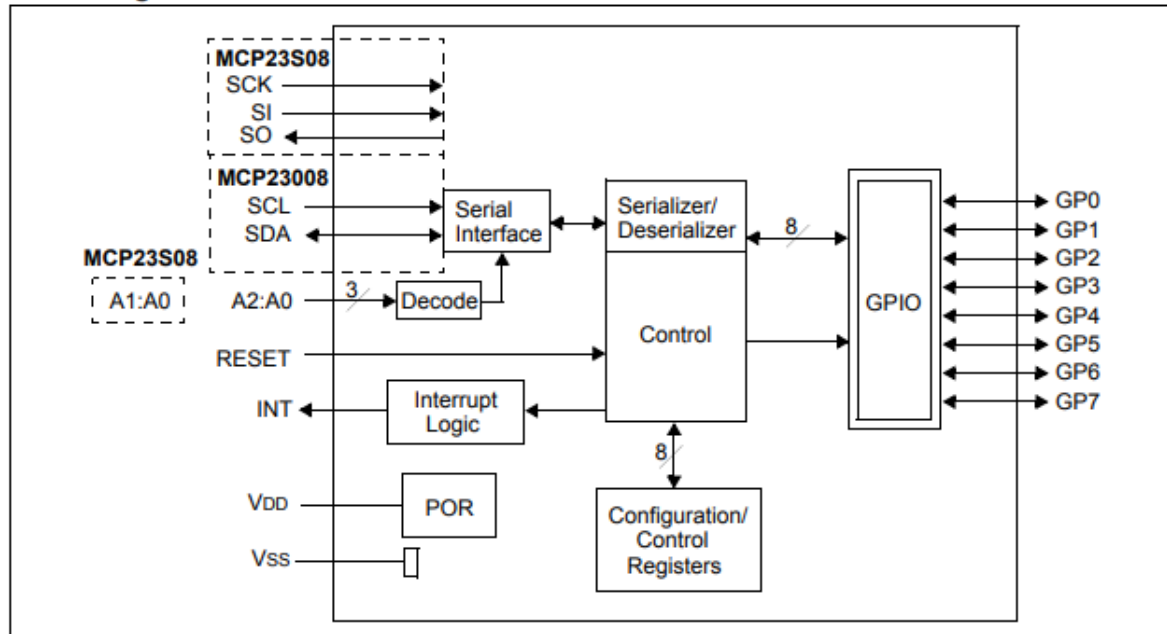
Tableau des adresses possibles :

A2	A1	A0	Adresse I2C
0	0	0	0x20
0	0	1	0x21
0	1	0	0x22
0	1	1	0x23
1	0	0	0x24
1	0	1	0x25
1	1	0	0x26
1	1	1	0x27

Analyse d'extraits de la datasheet :

Diagramme fonctionnel interne :

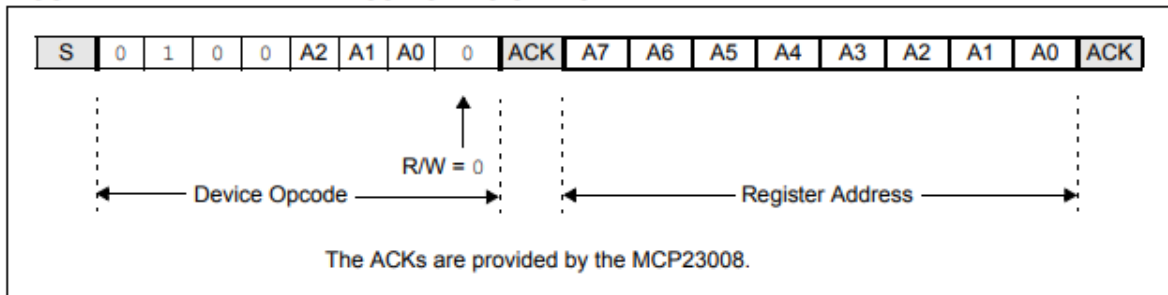
Block Diagram



- **Interface série I²C (SCL/SDA) :** gère la communication avec le microcontrôleur. C'est par ce canal que transitent les commandes et données.
- **Décodeur d'adresse (Decode) :** les broches A0 à A2 permettent de choisir l'adresse I2C de l'expandeur (jusqu'à 8 sur le même bus).
- **Contrôle & Serialiseur/Désérialiseur :** ce bloc convertit les données série (venant d'I2C) en données parallèles pour les GPIO (et inversement).
- **GPIO (GP0 à GP7) :** ce sont les 8 broches configurables individuellement en entrée ou en sortie.
- **Registres de configuration :** mémoire interne utilisée pour stocker la direction des broches, les valeurs de sortie, les états d'interruption, etc.
- **Interruption (INT) :** broche qui peut signaler au microcontrôleur un changement d'état sur un ou plusieurs GPIO.
- **RESET :** permet de réinitialiser le composant.
- **POR (Power-On Reset) :** système de mise à zéro automatique au démarrage pour garantir un état stable.

Registres d'adressage I²C :

FIGURE 1-4: I²C ADDRESSING REGISTERS



- Octet 1 : Adresse de l'appareil (Device Opcode)
- Composé de bits fixes 0100 suivis de A2, A1, A0 (configurés par des broches physiques sur le MCP23008).
- Le dernier bit indique si l'on souhaite lire (1) ou écrire (0).

Exemple : Si A2=A1=A0=0, l'adresse sera 0100 0000 soit 0x40 en écriture, et 0x41 en lecture.

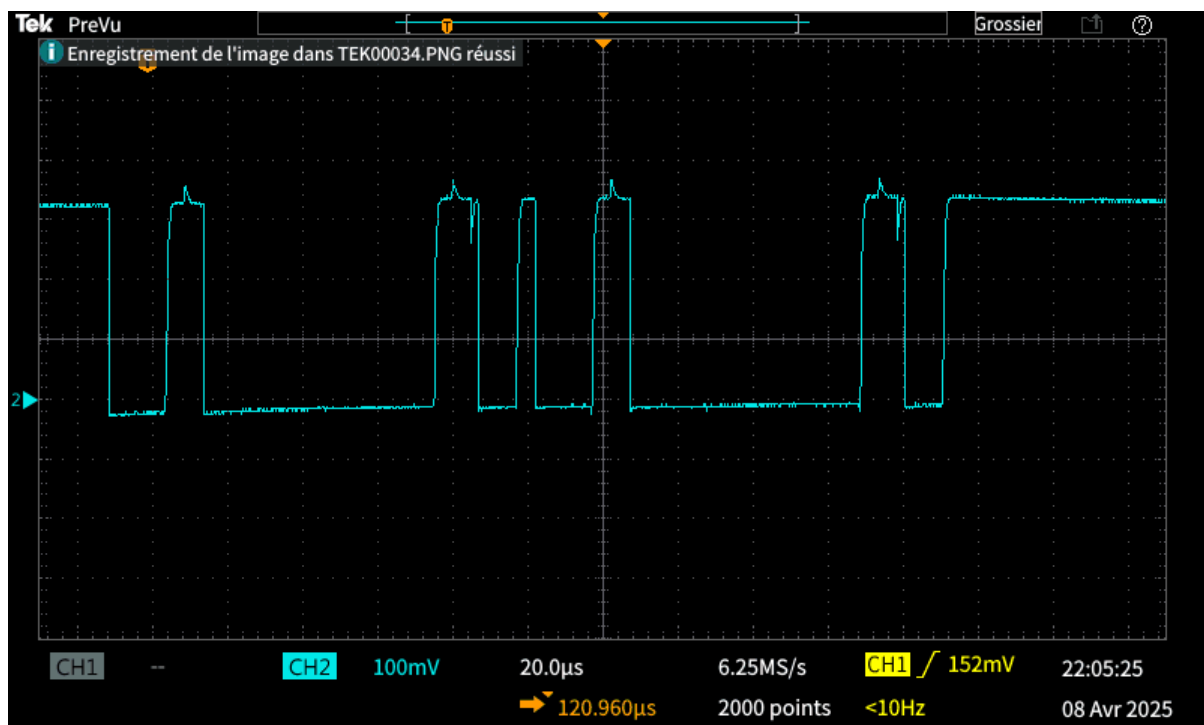
- ACK : Le périphérique MCP23008 confirme avoir reçu chaque octet en renvoyant un bit d'acquittement.
- Octet 2 : Adresse du registre interne
- Le maître envoie ensuite l'adresse du registre à lire/écrire (ex. 0x00 pour IODIR, 0x09 pour GPIO...).

Programme de test :

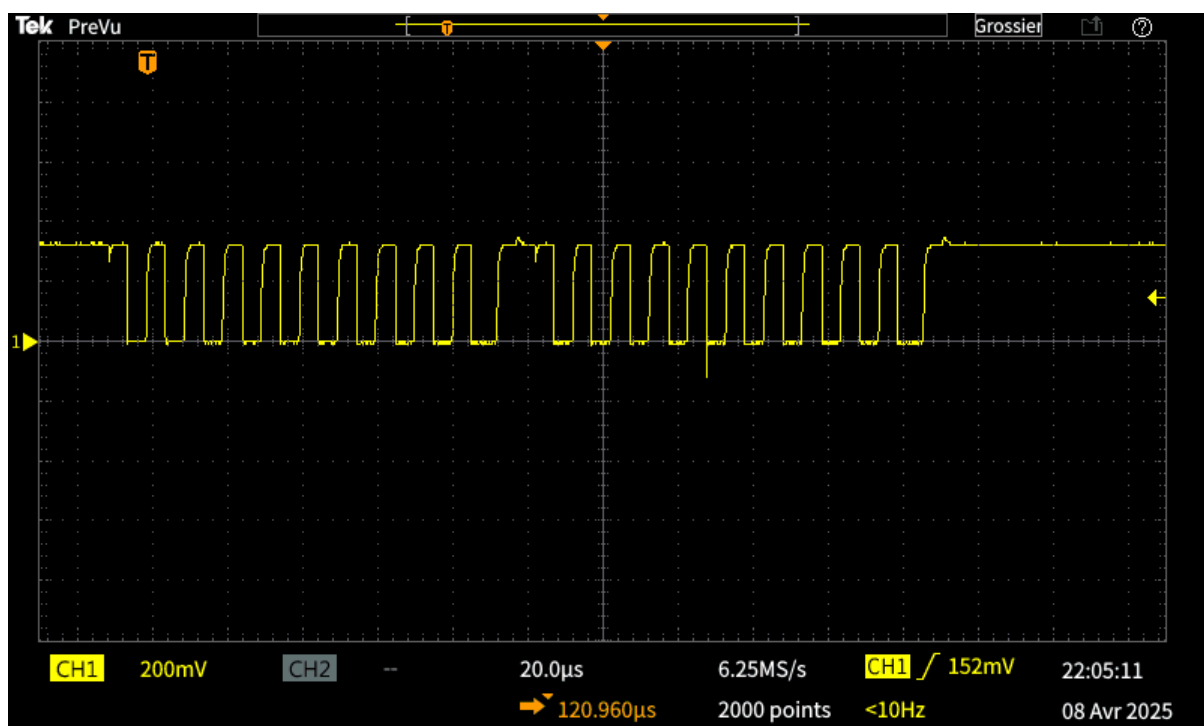
```
void setup() {  
  while (!Serial) delay(10);  
  Serial.println("Détection du MCP23008...");  
  // Initialisation avec l'adresse 0x20 (A0 = 0, A1 = 0, A2 = 0)  
  mcp.begin(0); // 0x20 est l'adresse par défaut  
  Serial.println("MCP23008 détecté avec succès à l'adresse 0x20 !");  
  // Configuration de la broche 0 comme sortie  
  mcp.pinMode(0, OUTPUT);  
  // Mise à HIGH de la broche 0  
  mcp.digitalWrite(0, HIGH);  
}  
void loop() {  
  // Clignotement de la broche 0  
  mcp.digitalWrite(0, HIGH);  
  delay(1000);  
  mcp.digitalWrite(0, LOW);  
  delay(1000);  
}
```

Mesure :

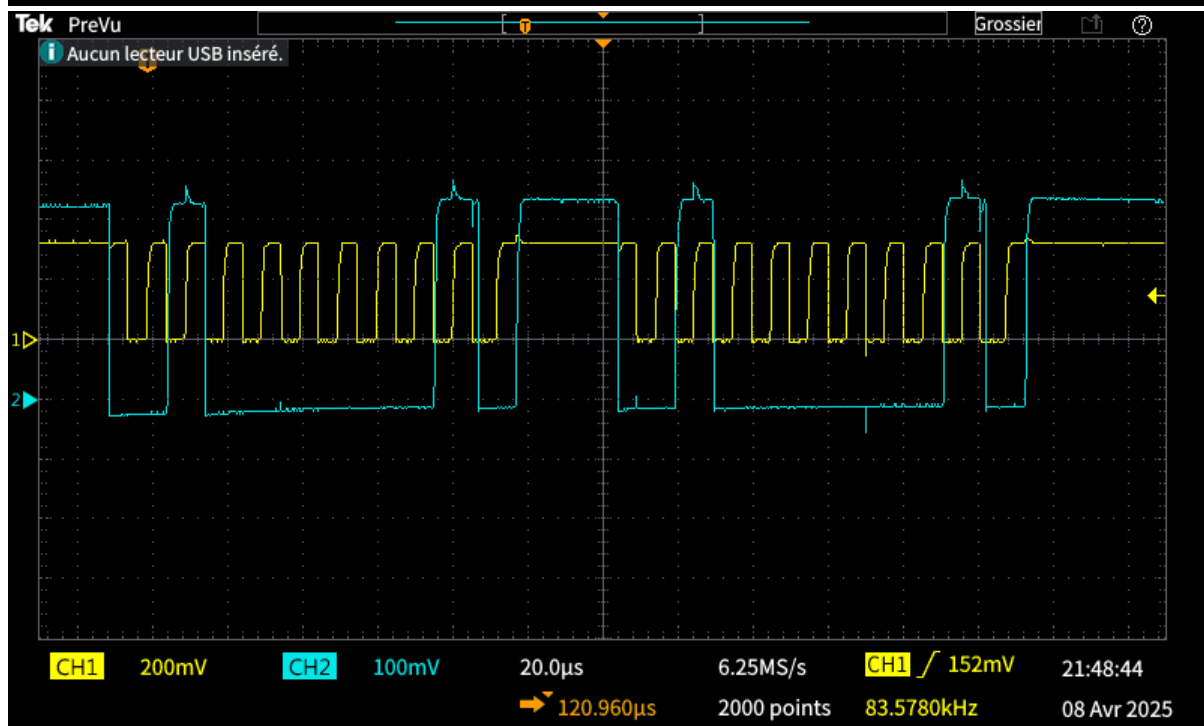
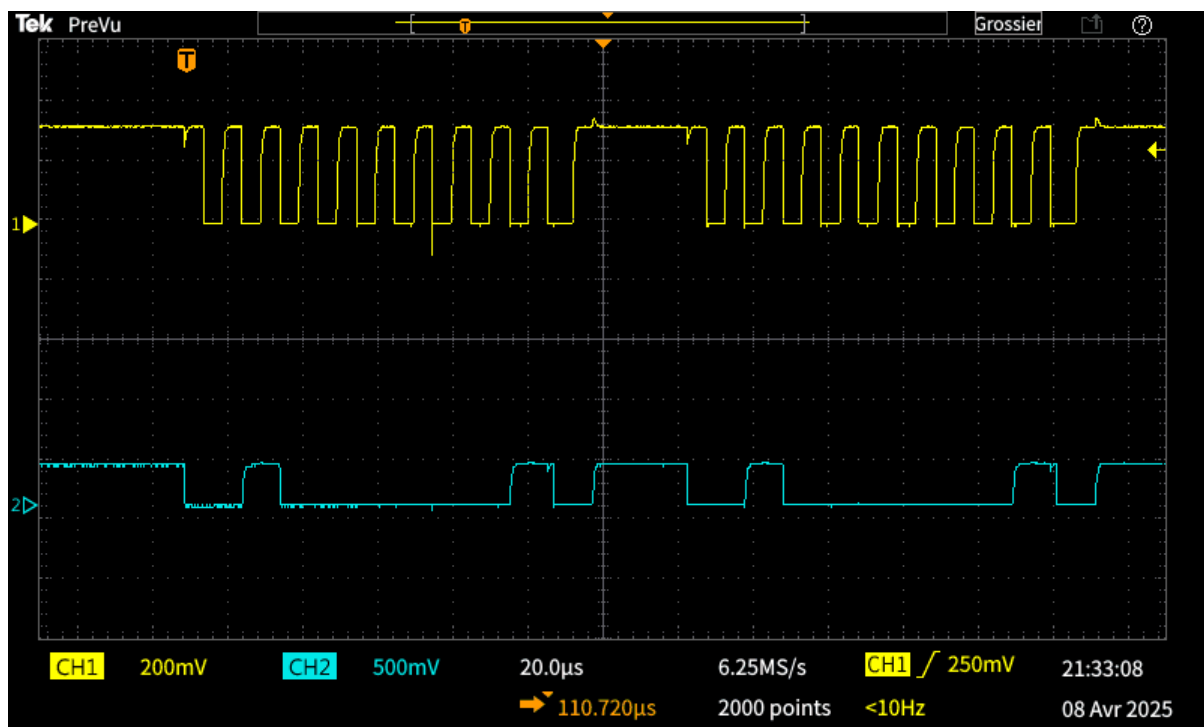
Canal bleu (CH2) : SDA (données)



Canal jaune (CH1) : SCL (horloge)



Les 2 signaux :

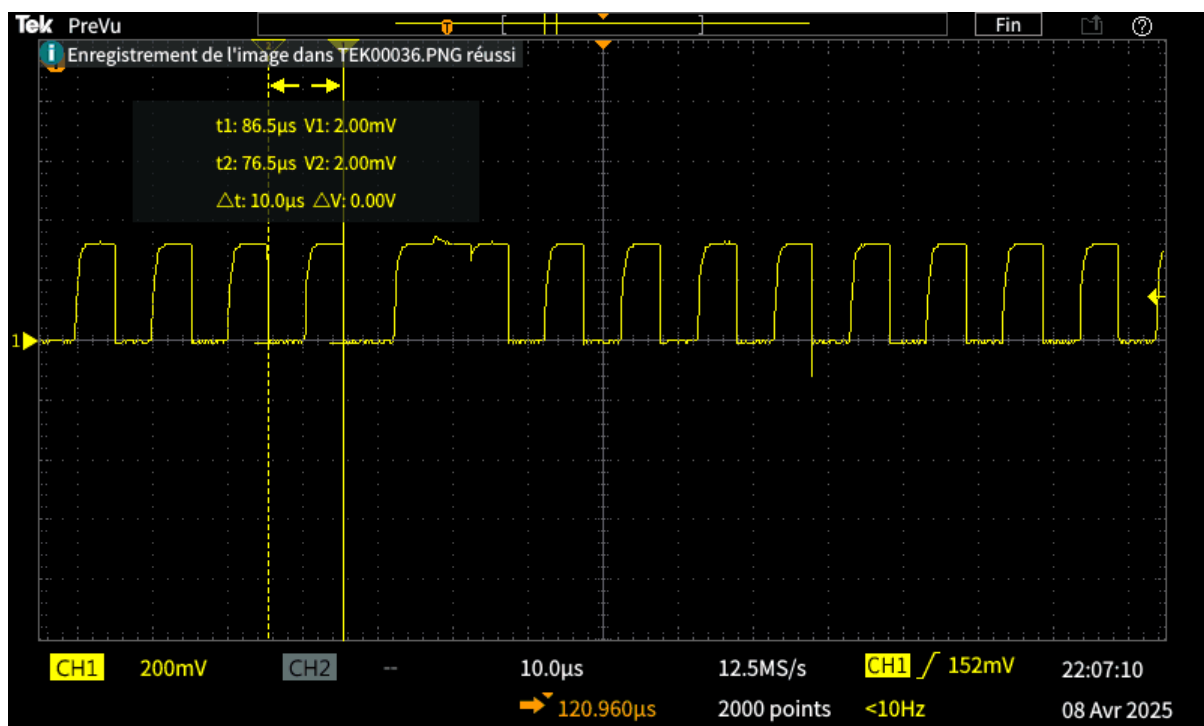


Interprétation du signal :

Les changements d'état SDA pendant SCL haut indiquent :

- **Start condition** (SDA ↓ pendant SCL ↑ au début du signal)
- **Stop condition** (SDA ↑ pendant SCL ↑ a la fin du signal)

Calcul des mesures :



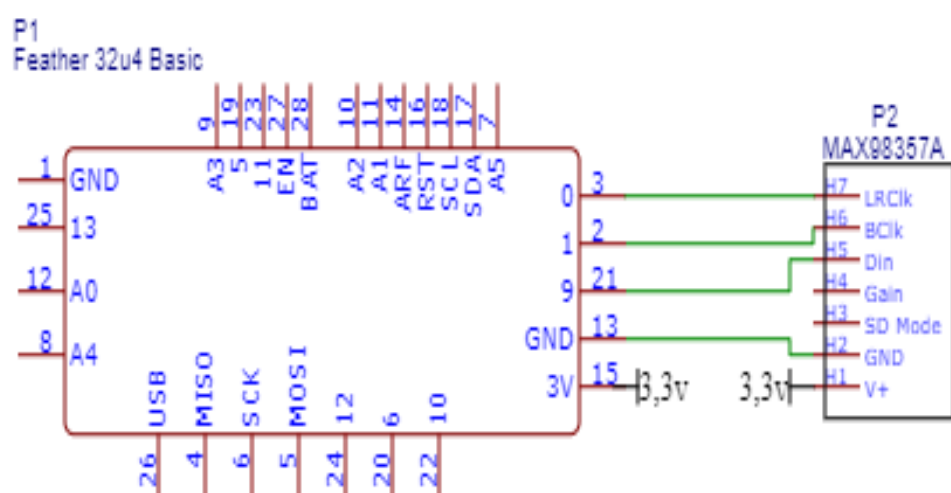
Durée d'un bit

$$\Delta t = 10 \mu s$$

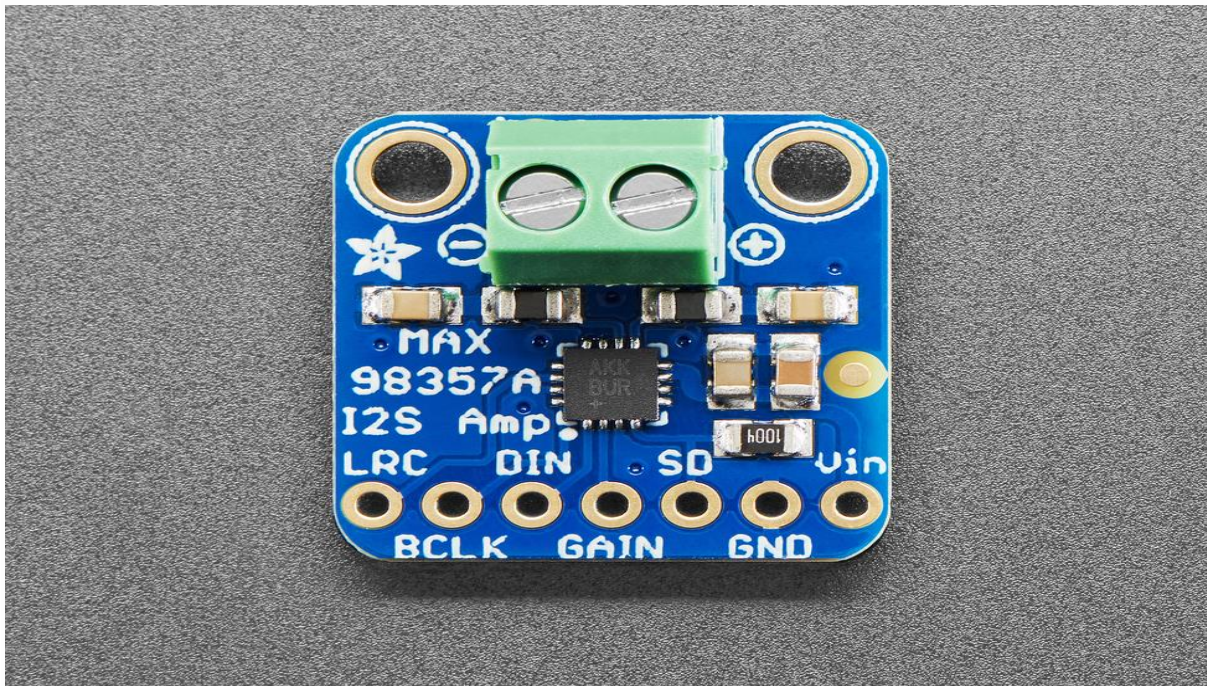
Baud rate

$$1 / \Delta t = 1 / 10 \times 10^{-6} s = 100\,000 \text{ bauds}$$

5.4. Schéma de l'ampli class D :



Présentation du MAX98357A :



<https://cdn-shop.adafruit.com/970x728/3006-08.jpg>

Le MAX98357 I2S Amp Breakout est une carte d'amplification audio numérique tout-en-un, compacte et efficace. Cette carte permet de recevoir un signal audio numérique via l'interface I2S — un protocole audio différent de l'I2C, utilisé pour transmettre des données sonores entre circuits intégrés. Ce breakout ne se contente pas de convertir le signal numérique en audio analogique, il l'amplifie également, fournissant jusqu'à 3,2 watts sur un haut-parleur de 4 ohms, avec une alimentation de 5V.

Il prend en charge les niveaux logiques 3,3V et 5V pour l'entrée I2S, et utilise une sortie audio en mode Bridge Tied Load (BTL), sans masse partagée, ce qui permet de connecter directement un haut-parleur (mais pas un autre ampli).

Le breakout inclut également des fonctionnalités de personnalisation : une broche de gain permet de régler l'amplification de 3 dB à 15 dB, tandis que la broche Mode permet de choisir entre un mix stéréo (L+R)/2 ou une sortie uniquement sur le canal gauche ou droit. Avec ses protections intégrées contre la surchauffe, ce module offre une solution robuste, abordable et facile à intégrer pour ajouter du son amplifié à n'importe quel projet embarqué.

Rôle des signaux :

- LRC (Left/Right Clock) - c'est la broche qui indique à l'amplificateur quand les données sont pour le canal gauche et quand elles sont pour le canal droit
 - ❖ LRC = 0 → Données pour le canal gauche (Left)
 - ❖ LRC = 1 → Données pour le canal droit (Right)
- BCLK (Bit Clock) - Il s'agit de la broche qui indique à l'amplificateur quand lire les données sur la broche de données. Capture les données sur le flanc montant du BCLK.
- DIN (Data In) - Il s'agit de la broche qui contient les données réelles.
- GAIN est, eh bien, le réglage de gain. Vous pouvez avoir un gain de 3dB, 6dB, 9dB, 12dB ou 15dB.
 - ❖ 15dB si une résistance de 100K est connectée entre GAIN et GND
 - ❖ 12dB si GAIN est connecté directement à GND
 - ❖ 9dB si GAIN n'est connecté à rien (c'est la valeur par défaut)
 - ❖ 6dB si GAIN est connecté directement à V+
 - ❖ 3dB si une résistance de 100K est connectée entre GAIN et V+
- SD Cette broche est utilisée pour le mode d'arrêt, mais elle est également utilisée pour définir le canal de sortie.
- V+ Alimentation positive pour l'ampli • GND = Masse

Calcul de la fréquence BCLK :

La fréquence BCLK est donnée par : $\text{Fréquence BCLK} = \text{Taux d'échantillonnage} \times \text{Nombre de bits par canal} \times \text{Nombre de canaux}$

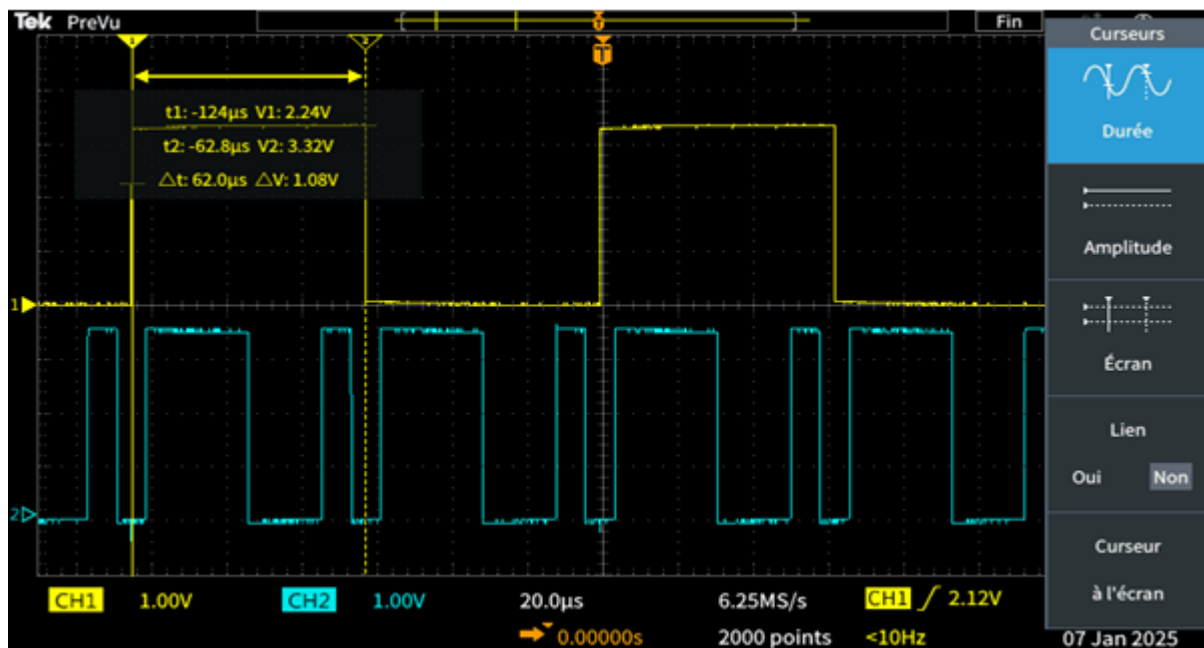
$\text{Fréquence BCLK} = 8000 \times 16 \times 2 = 256\text{kHz}$.

Programme :

```
#include <I2S.h>
const int frequency = 440; // frequency of square wave in Hz
const int amplitude = 500; // amplitude of square wave
const int sampleRate = 8000; // sample rate in Hz
const int halfWavelength = (sampleRate / frequency); // half wavelength of square wave
short sample = amplitude; // current sample value
int count = 0;
void setup() {
  Serial.begin(9600);
  Serial.println("I2S simple tone");
  // start I2S at the sample rate with 16-bits per sample
  if (!I2S.begin(I2S_PHILIPS_MODE, sampleRate, 16)) {
    Serial.println("Failed to initialize I2S!");
    while (1); // do nothing
  }
}
void loop() {
  if (count % halfWavelength == 0) {
    // invert the sample every half wavelength count multiple to generate square wave
    sample = -1 * sample;
  }
  // write the same sample twice, once for left and once for the right channel
  I2S.write(sample);
  I2S.write(sample);
  // increment the counter for the next sample
  count++;
}
```

Mesure :

LRC : Jaune



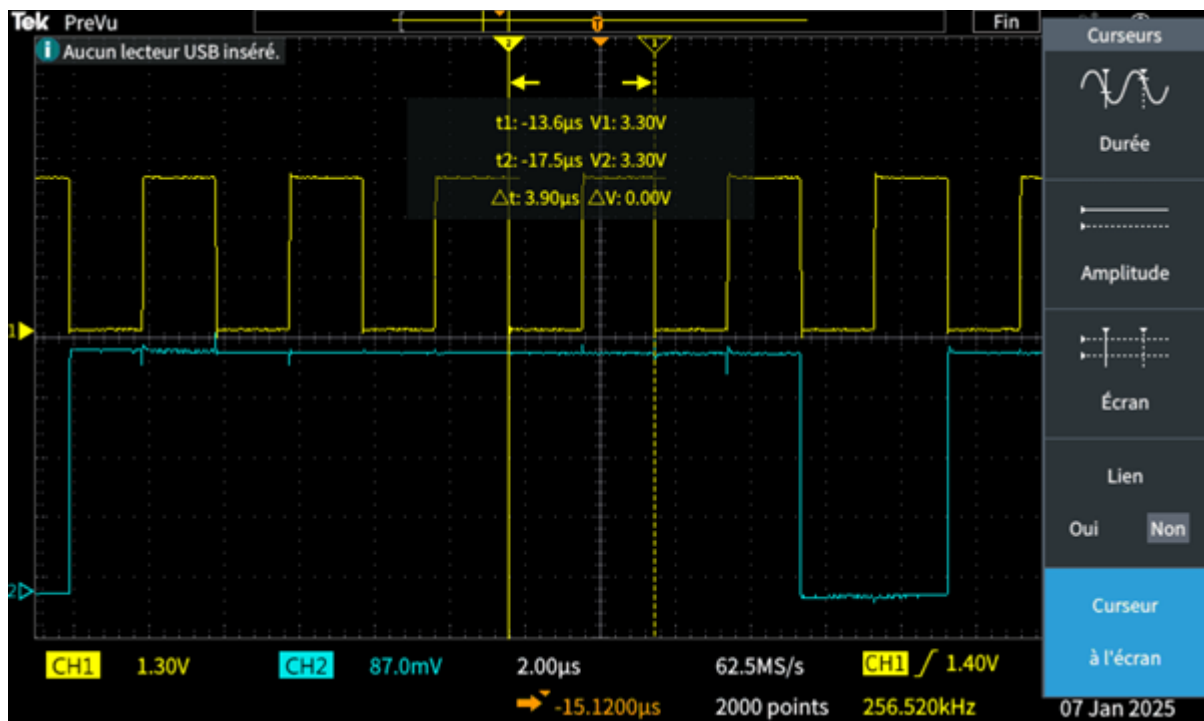
Delta temporel (Delta/Δt) :

$t_1 = -124\mu s$

$t_2 = -62.8\mu s$

$\Delta t = 62.0\mu s$, qui correspond à la période du signal LRCLK.

BLCK : Jaune



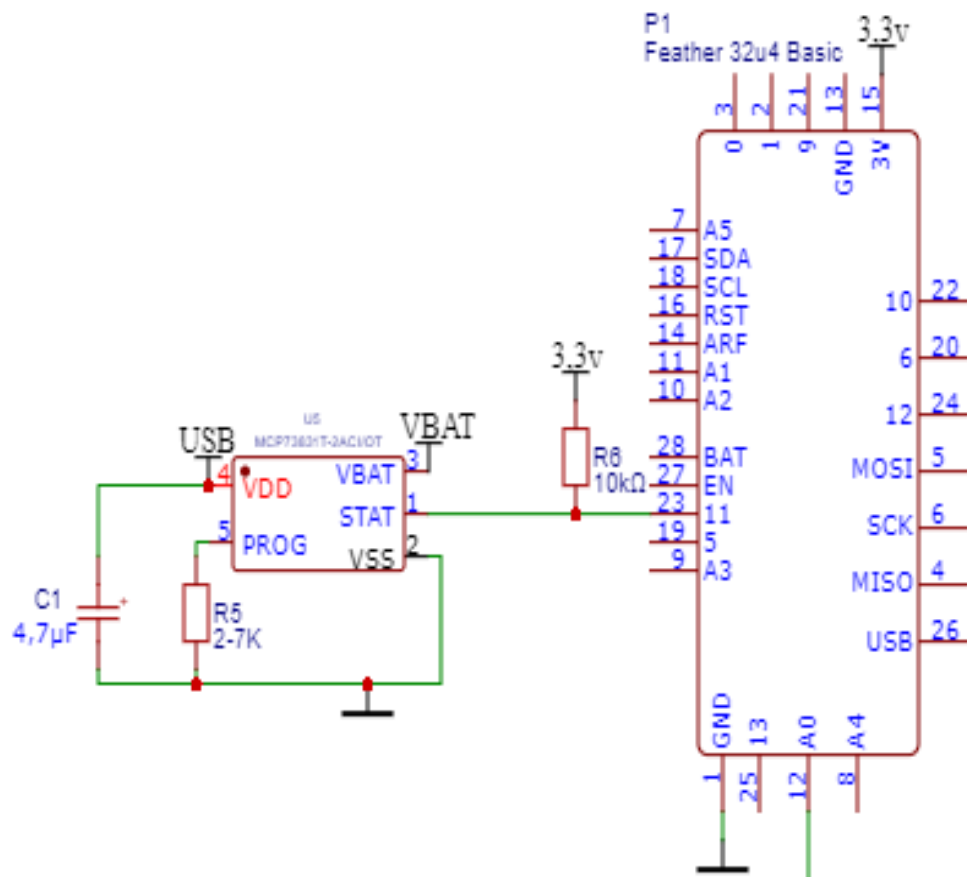
Delta temporel (Delta / Δt) :

t1= -13.6µs

t2 = -17.5µs

Δt=3.90µs

5.5. Schéma du MCP73831/2:



Présentation du MCP73831/2 :



[MCP73831 | Microchip Technology](#)

Le MCP73831 et le MCP73832 sont des circuits intégrés de gestion de charge pour batteries Li-Ion/Li-Polymère, ces composants sont adaptés aux applications portables à faible consommation d'énergie, comme les capteurs embarqués ou modules sans fil.

Ces contrôleurs de charge utilisent une méthodologie de charge en trois étapes (précharge, charge rapide, et terminaison), assurant une charge sûre, fiable pour la durée de vie des batteries.

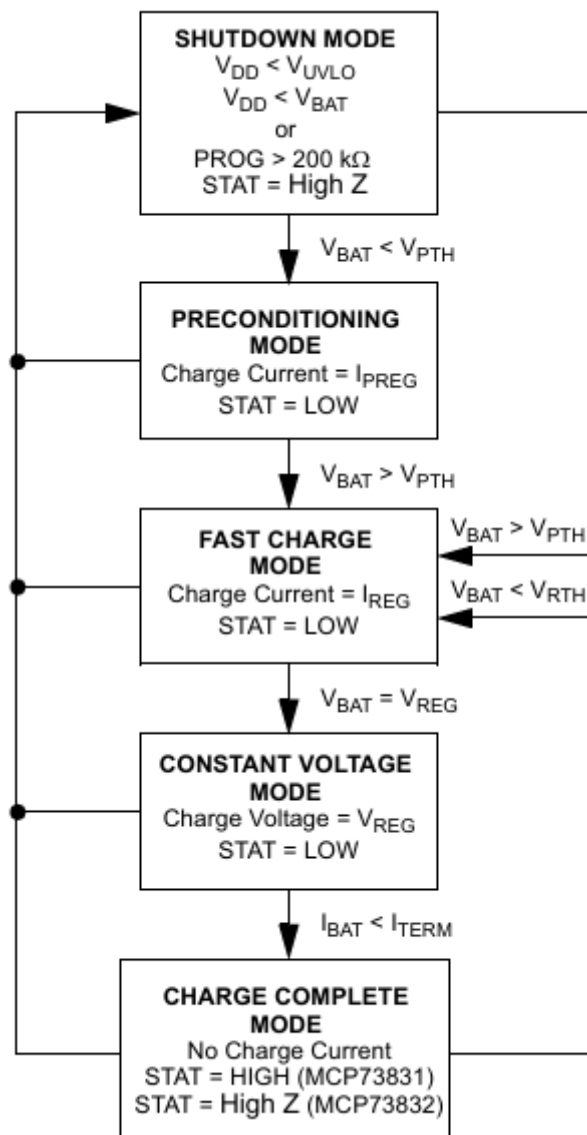
Rôle des pattes :

- **STAT** – Sortie d'état de charge
Cette broche indique l'état de la charge :
- **Basse (LOW)** : batterie en cours de charge
- **Haute impédance (Hi-Z)** : charge terminée ou sans batterie
- **Haute (HIGH)** : erreur ou pas de charge
Elle peut être connectée à une LED pour indiquer visuellement l'état de la charge ou à un microcontrôleur pour une surveillance numérique.
- **VSS** – Référence de masse
- **VBAT** – Sortie de charge vers la batterie
Connectée au pôle positif de la batterie, cette broche fournit le courant de charge. Il est recommandé de placer un condensateur de 4,7 μF entre VBAT et VSS pour assurer la stabilité du circuit, notamment lorsque la batterie est déconnectée.
- **VDD** – Alimentation d'entrée
Cette broche reçoit la tension d'alimentation du chargeur, typiquement entre 3,75 V et 6 V.
- **PROG** – Programmation du courant de charge
En connectant une résistance entre PROG et VSS, on définit le courant de charge maximal. Par exemple :
- **10 k Ω** → environ **100 mA**
- **2 k Ω** → environ **500 mA**
Si PROG est laissée en l'air (non connectée), le circuit de gestion de charge est désactivé.

Caractéristiques principales :

- **Tension d'alimentation :** 3,75 V à 6 V
- **Courant de charge programmable :** jusqu'à 500 mA
- **Indication de charge via LED (pin STAT)**
- **Protection contre la surcharge et la surchauffe**
- **Faible encombrement :** boîtiers SOT-23-5

Analyse d'extraits de la datasheet :

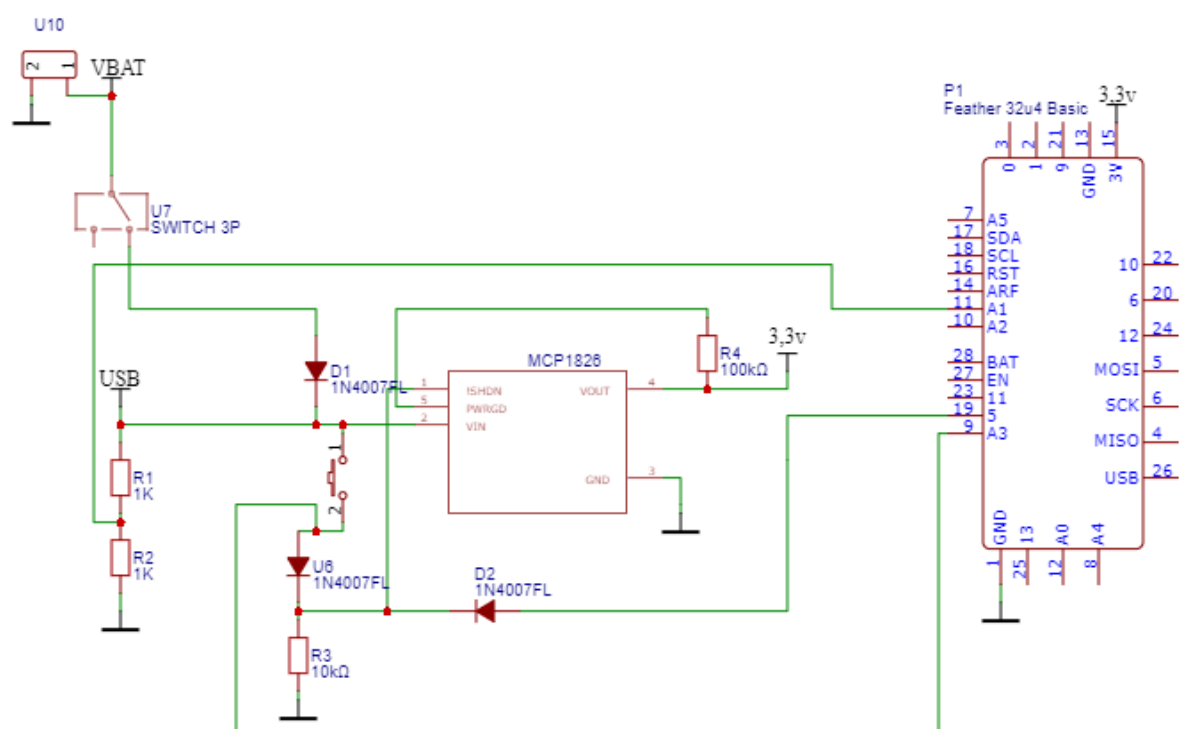


Phases du cycle de charge

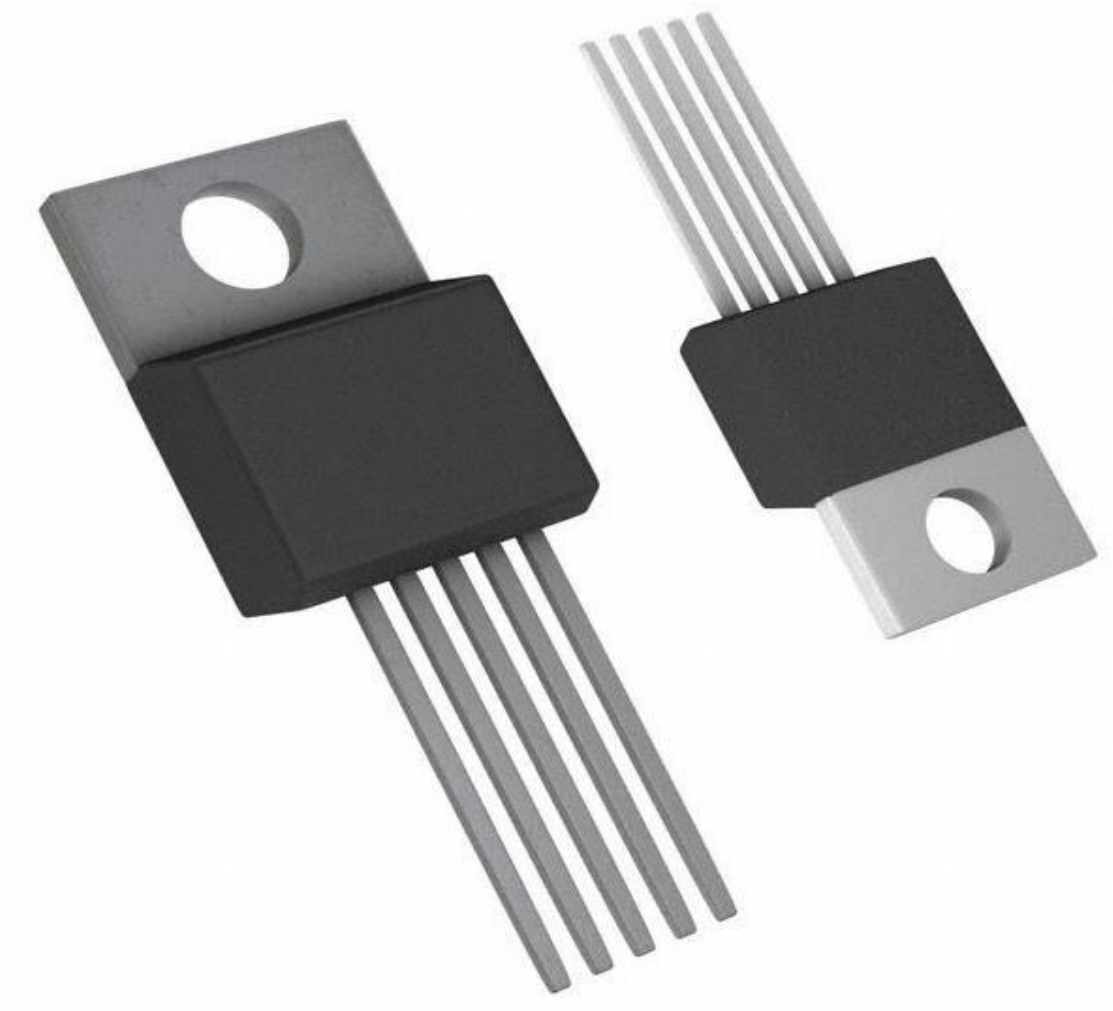
Le MCP73831 utilise un algorithme de charge en courant constant / tension constante (CC/CV), structuré en plusieurs étapes

1. **Pré conditionnement** : Si la tension de la batterie est inférieure à un certain seuil (par exemple, 2.8 V), le circuit applique un courant réduit (généralement 10 % du courant de charge programmé) pour éviter d'endommager la batterie.
2. **Charge rapide (courant constant)** : Une fois la tension minimale atteinte, le circuit fournit un courant constant à la batterie. Ce courant est déterminé par une résistance externe connectée à la broche PROG.
3. **Charge en tension constante** : Lorsque la tension de la batterie atteint la tension de régulation (par exemple, 4.20 V), le circuit maintient cette tension constante. Le courant diminue progressivement à mesure que la batterie se charge complètement.
4. **Fin de charge** : Le processus de charge se termine lorsque le courant tombe en dessous d'un seuil spécifique (par exemple, 10 % du courant de charge programmé).

5.6. Schéma du MCP1826 :



Présentation du MCP1826 :



[MCP1826-ADJE/AT Microchip Technology | Circuits intégrés | DigiKey](#)

Le MCP1826 est un régulateur LDO capable de fournir un courant de sortie jusqu'à 1 A, avec une tension de sortie fixe ou ajustable entre 0,8 V et 5,0 V. Il est conçu pour des applications nécessitant une alimentation stable et efficace, telles que les ordinateurs portables, les cartes réseau et les équipements embarqués.

Caractéristiques principales :

- **Courant de sortie maximal** : 1 A
- **Tension d'entrée** : 2,3 V à 6 V
- **Tension de sortie** : fixe ou ajustable de 0,8 V à 5,0 V
- **Faible courant de repos** : 120 μ A typique
- **Protection contre les surintensités et la surchauffe**

MCP1826 – Boîtier à 5 broches (SOT-223-5, TO-220-5, DDPAK-5)

Broche	Nom	Fonction
1	SHDN	Entrée de mise en veille : Permet d'activer ou de désactiver le régulateur. Niveau bas pour désactiver.
2	VIN	Tension d'entrée : Alimentation principale du régulateur.
3	GND	Masse : Référence de tension du circuit.
4	VOUT	Tension de sortie régulée : Fournit la tension régulée au circuit.
5	PWRGD / ADJ	Sortie "Power Good" (version à tension fixe) : Indique si la sortie est dans la plage de régulation. Entrée d'ajustement (version ajustable) : Permet de définir la tension de sortie via un diviseur de tension externe.

MCP1826S – Boîtier à 3 broches (SOT-223-3, TO-220-3, DDPAK-3)

Broche	Nom	Fonction
1	VIN	Tension d'entrée : Alimentation principale du régulateur.
2	GND	Masse : Référence de tension du circuit.
3	VOUT	Tension de sortie régulée : Fournit la tension régulée au circuit.

Explication complète du schéma :

1. Source principale : VBAT et USB

VBAT : provient de la batterie (connecteur U10).

USB : fourni de l'alimentation via le port micro-USB et charge la batterie.

2. Diodes D1, D2 (1N4007FL)

Ces diodes assurent une protection contre les retours de courant :

D1 bloque les retours du régulateur MCP1826 vers l'USB.

D2 bloque les retours de la batterie VBAT vers le régulateur.

3. MCP1826 : Régulateur de tension

But : abaisser la tension de la batterie (~3.7V) à 3.3V régulée.

Broches :

SHDN (pin 1) : mise en marche/arrêt du régulateur (ici contrôlée par le microcontrôleur ou forcée à l'état actif via R4).

VIN (pin 2) : entrée venant de la source (via D1).

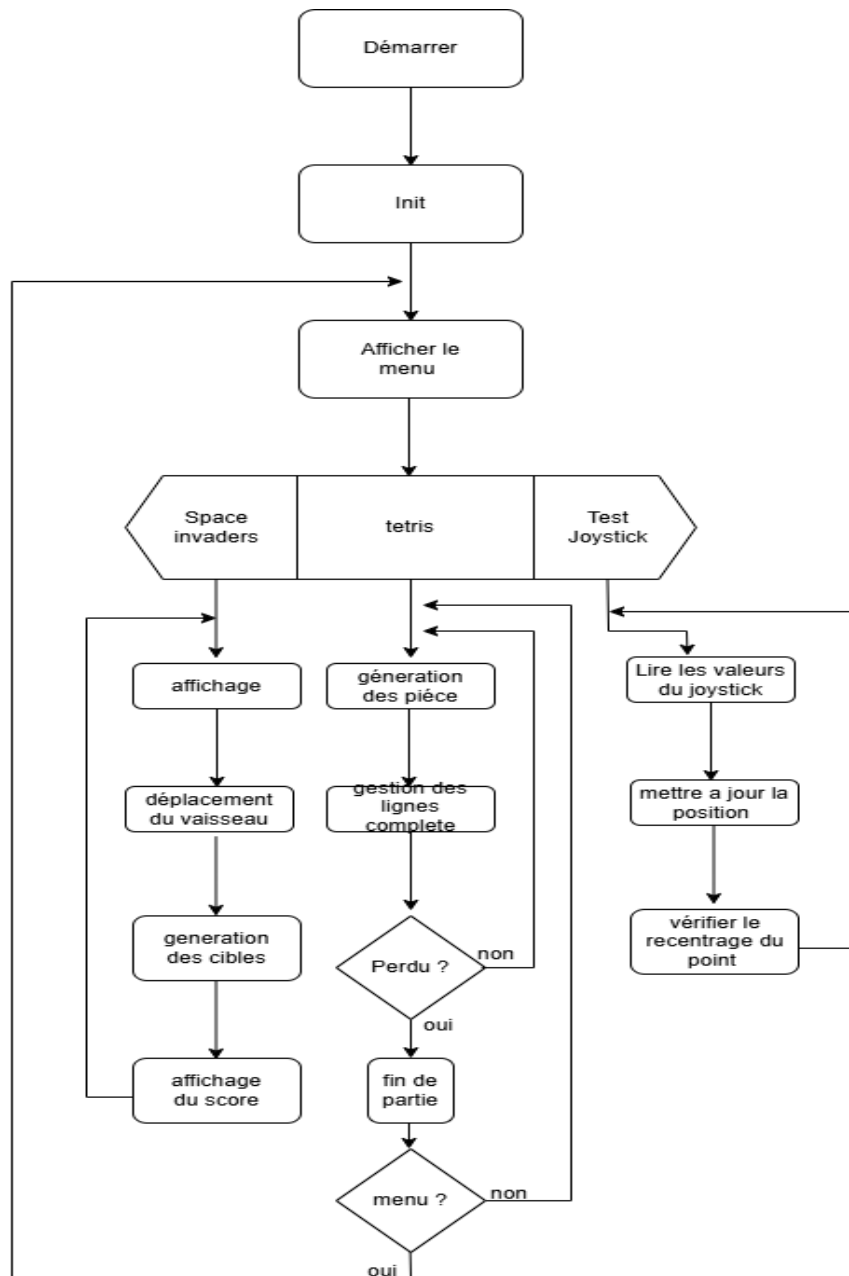
GND (pin 3) : masse

VOUT (pin 4) : sortie régulée à 3.3V..

PWRGD (pin 5) : signal d'état (Power Good).

R4 (100k Ω) tire SHDN vers le haut (vers 3.3V), maintenant le régulateur activé.

6. L'ordinogramme:



6.2. Programmation :



https://github.com/Achraf-ElAmrani/Gamboy_Achraf_2024/tree/main/Gameboy_Achraf_2024

7.Fabrication :

Choix des composants

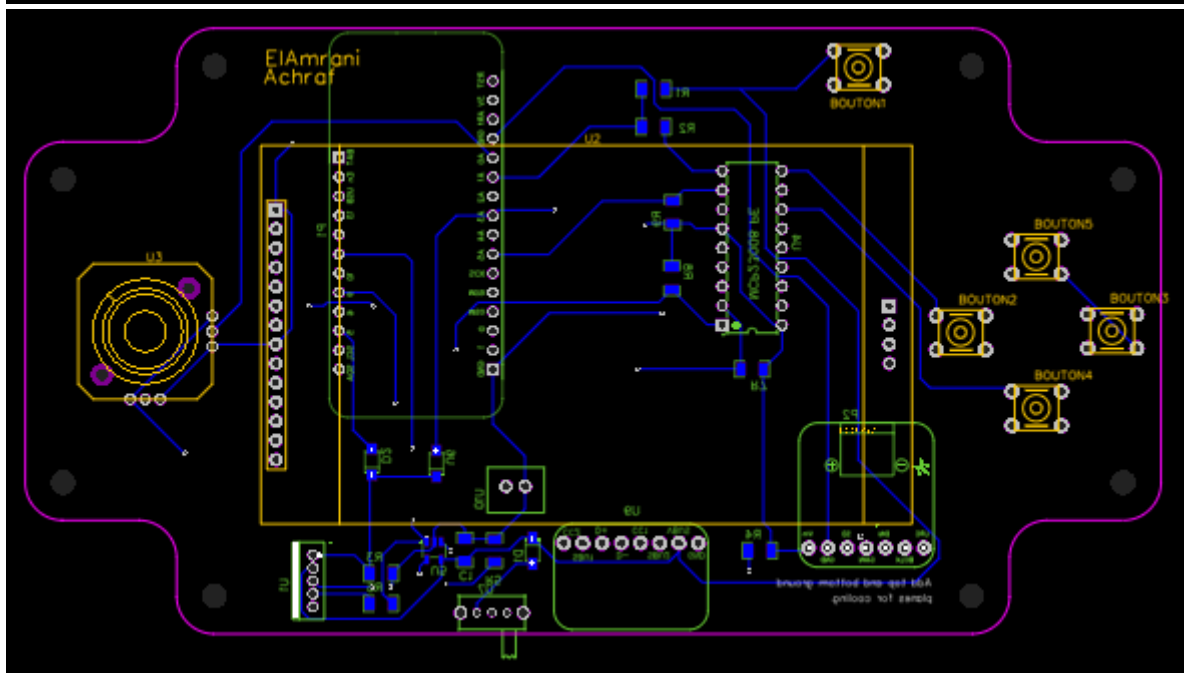
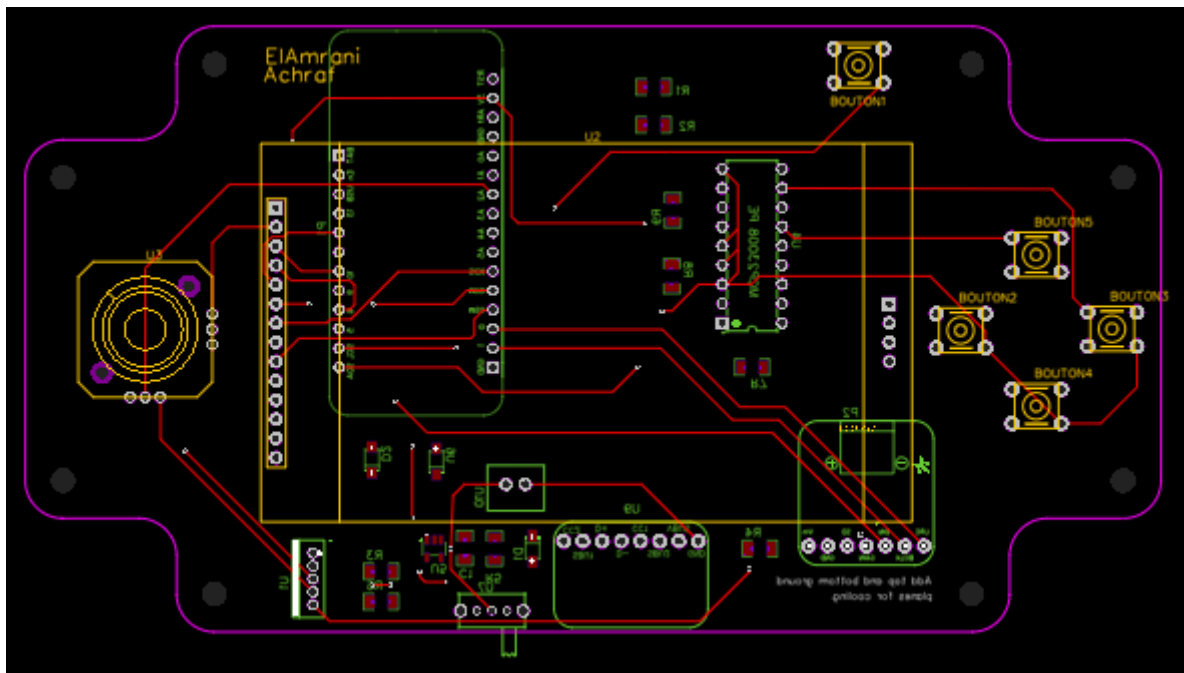
La première étape de la conception de ma console portable a consisté à sélectionner les composants électroniques nécessaires à son bon fonctionnement. J'ai opté pour un microcontrôleur performant et économe en énergie, un écran couleur adapté à un usage nomade, des boutons ergonomiques, une batterie rechargeable lithium-ion.

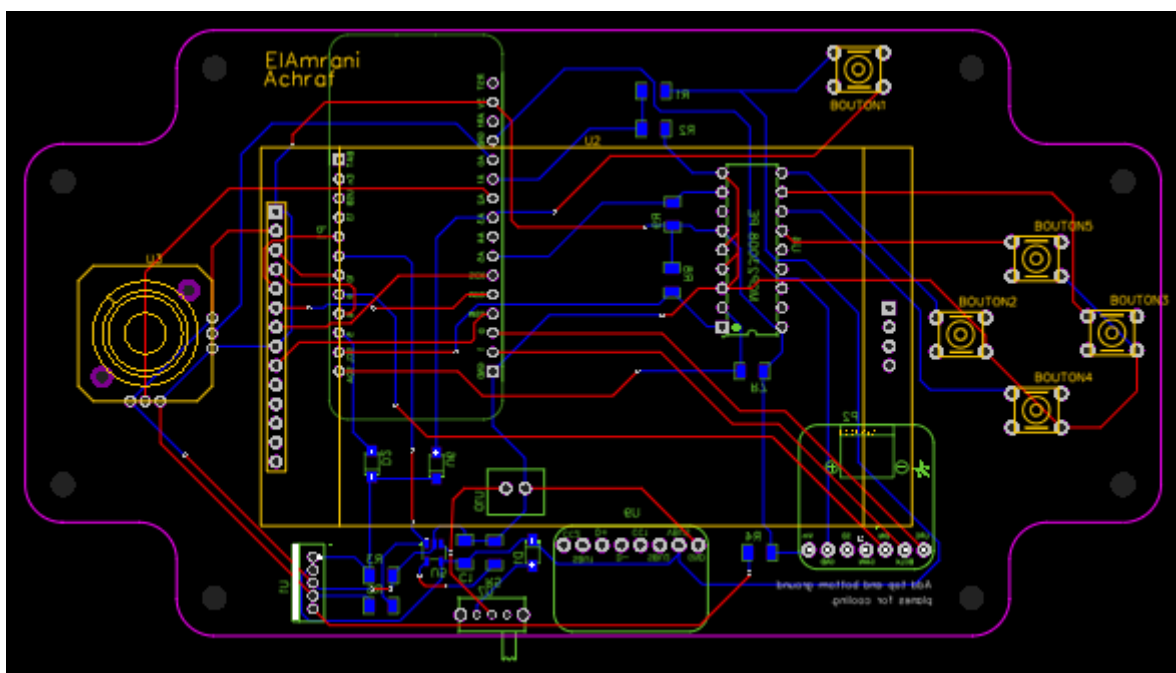
Création du schéma de principe (P15)

Une fois les composants sélectionnés, j'ai réalisé le schéma de principe du projet. Ce schéma représente les connexions électriques entre les différents éléments : alimentation, microcontrôleur, écran, etc. Cette partie est importante pour anticiper les erreurs de conception et optimiser le placement des composants.

Conception du PCB et choix d'une taille compacte

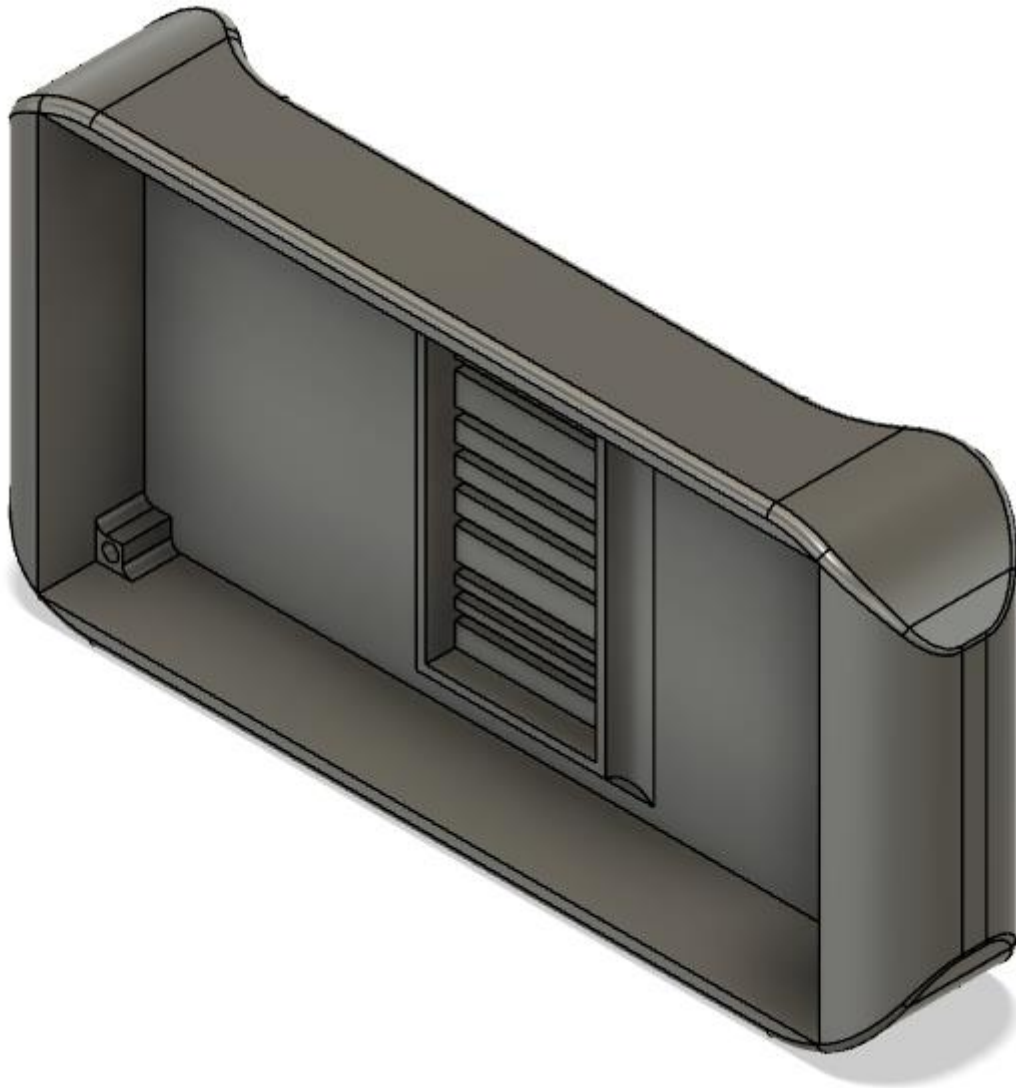
Sur la base du schéma de principe, j'ai conçu le circuit imprimé (PCB) en prenant en compte la taille liées à la portabilité de la console. J'ai minimisé l'encombrement tout en garantissant la fiabilité des connexions électriques. Le format final du PCB a été pensé pour s'intégrer parfaitement dans un boîtier compact.

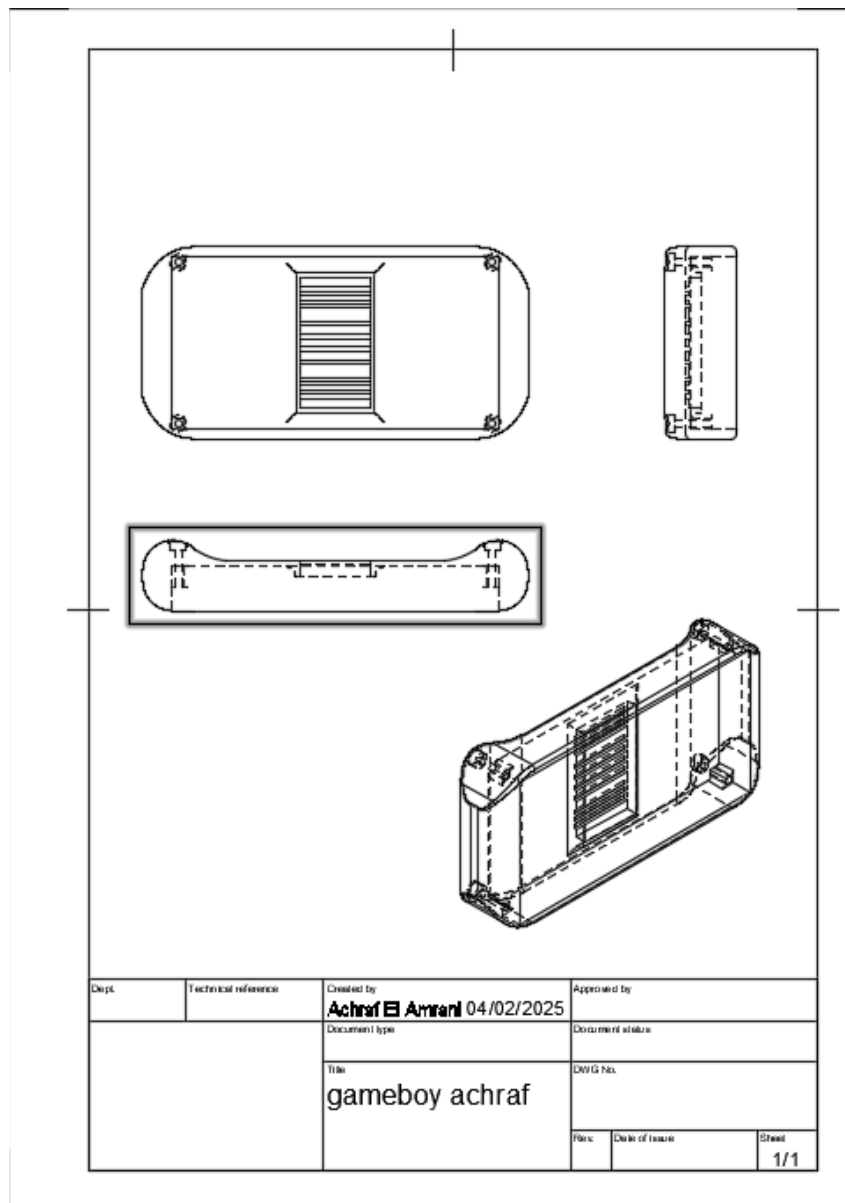




Modélisation 3D de la console portable

Pour garantir une bonne prise en main, j'ai conçu un boîtier 3D adapté. L'objectif était de créer une forme agréable à tenir et un accès confortable aux boutons.





Programmation de la console

Enfin, j'ai développé le logiciel embarqué permettant à la console de fonctionner. Cela inclut la lecture des entrées (boutons), l'affichage sur l'écran, ainsi que le chargement et le lancement des jeux.

8.Mise au point :

Au cours de la réalisation de mon projet, plusieurs difficultés techniques sont apparues, notamment lors de la mise au point de ma carte électronique (PCB) et de la programmation. Voici un aperçu détaillé des problèmes rencontrés et des solutions que j'ai mises en œuvre.

Tout d'abord, lors de la conception de ma PCB, je me suis trompé dans le raccordement de certaines pattes de composants. Cette erreur de connectique m'a obligé à couper certaines pistes à l'aide d'un cutter pour les connexions incorrectes, puis j'ai réparé les liaisons en soudant manuellement des fils sur les bonnes pattes. Cela m'a permis de sauver la carte sans avoir à en refaire une nouvelle.

Un autre problème est survenu avec le composant MCP23008. L'adresse I2C de ce n'était pas reconnue par le microcontrôleur. Après des recherches, j'ai découvert que le problème venait d'erreur d'inattention sur ma PCB : les résistances de pull-up n'étaient pas présentes sur les lignes SDA et SCL, ce qui empêchait une bonne communication I2C. De plus, j'avais complètement oublié de connecter la broche RESET du composant. Une fois ces erreurs corrigées, la communication a pu être rétablie correctement.

J'ai également rencontré un souci d'interférence dans le code lors de la déclaration de la bibliothèque I2S. Ce conflit empêchait la compilation du programme. Grâce à l'aide de Monsieur Kapita, le problème a pu être résolu en modifiant certains paramètres dans les fichiers de la bibliothèque à l'aide du logiciel Notepad++.

Un autre point bloquant était le rafraîchissement de l'écran. Au départ, j'avais programmé un rafraîchissement complet de l'affichage à chaque mise à jour, ce qui ralentissait considérablement l'affichage et provoquait des clignotements. Pour améliorer les performances, j'ai appris à ne rafraîchir que les objets nécessaires, ce qui a rendu l'affichage plus fluide.

Enfin, un dernier problème a été rencontré avec l'impression 3D. Les machines de l'école n'imprimaient pas correctement ma pièce, ce qui m'a retardé dans la fabrication du boîtier de mon projet. J'ai finalement utilisé une imprimante 3D externe.

9.Conclusion :

Le projet fonctionne partiellement : seule la partie alimentation est défectueuse, en raison d'un problème de réglage, mais les autres parties du projet sont fonctionnelles.

Au départ, mon projet avait bien débuté. J'avais une bonne vision d'ensemble et une motivation solide. Cependant, j'ai rencontré des difficultés avec certains composants, ce qui m'a retardé, notamment sur le sujet principal du projet : le codage. Ces imprévus ont perturbé ma gestion du temps et m'ont obligé à adapter mes priorités en cours de route.

Si je devais améliorer ce projet, je me concentrerais en priorité sur la gestion de la batterie et la correction des parties du code qui posent des problèmes. Une meilleure gestion de mon temps aurait également permis d'éviter certaines erreurs.

Concernant la gestion de projet, j'ai fait preuve de persévérance, mais je reconnais qu'une meilleure organisation et un plan plus structuré auraient facilité l'avancement.

Malgré ces obstacles, cette expérience m'a permis d'apprendre de nombreuses choses : j'ai compris comment rafraîchir un écran de manière efficace, créer du son via la programmation, et développer un projet compact. Ces compétences techniques sont des acquis importants pour la suite de mon parcours.

En conclusion, bien que le projet ne soit pas totalement fonctionnel, il m'a permis de progresser, de faire face à des situations complexes et d'en tirer des leçons précieuses. Je termine donc ce travail sur une note positive, avec l'envie de continuer à m'améliorer et à apprendre.

Webographie :

Adafruit MAX98357 I2S Class-D Mono Amp (25/05/2025) :

[Downloads | Adafruit MAX98357 I2S Class-D Mono Amp | Adafruit Learning System](#)

I2S (25/05/2025) :

[I2S — Wikipédia](#)

MAX98357A Class D'Amplifier (25/05/2025) :

<https://youtu.be/At8PDQ3g7FQ>

Max98357 - What is I2S ?

https://www.az-delivery.de/en/blogs/azdelivery-blog-fur-arduino-und-raspberry-pi/max98357-was-ist-eigentlich-i2s?utm_source=chatgpt.com

Interfacing Arduino with ILI9341 color TFT display (25/05/2025) :

[Interfacing Arduino with ILI9341 color TFT display](#)

Introduction to the Serial Peripheral Interface (25/05/2025) :

[Introduction to the Serial Peripheral Interface | Arduino Documentation](#)

Serial Peripheral Interface (25/05/2025) :

[Serial Peripheral Interface - Wikipedia](#)

Adafruit GFX Library (25/05/2025) :

<https://github.com/adafruit/Adafruit-GFX-Library>

Adafruit ILI9341 (25/05/2025) :

https://github.com/adafruit/Adafruit_ILI9341

Adafruit MCP23008 library (25/05/2025) :

<https://github.com/adafruit/Adafruit-MCP23008-library>

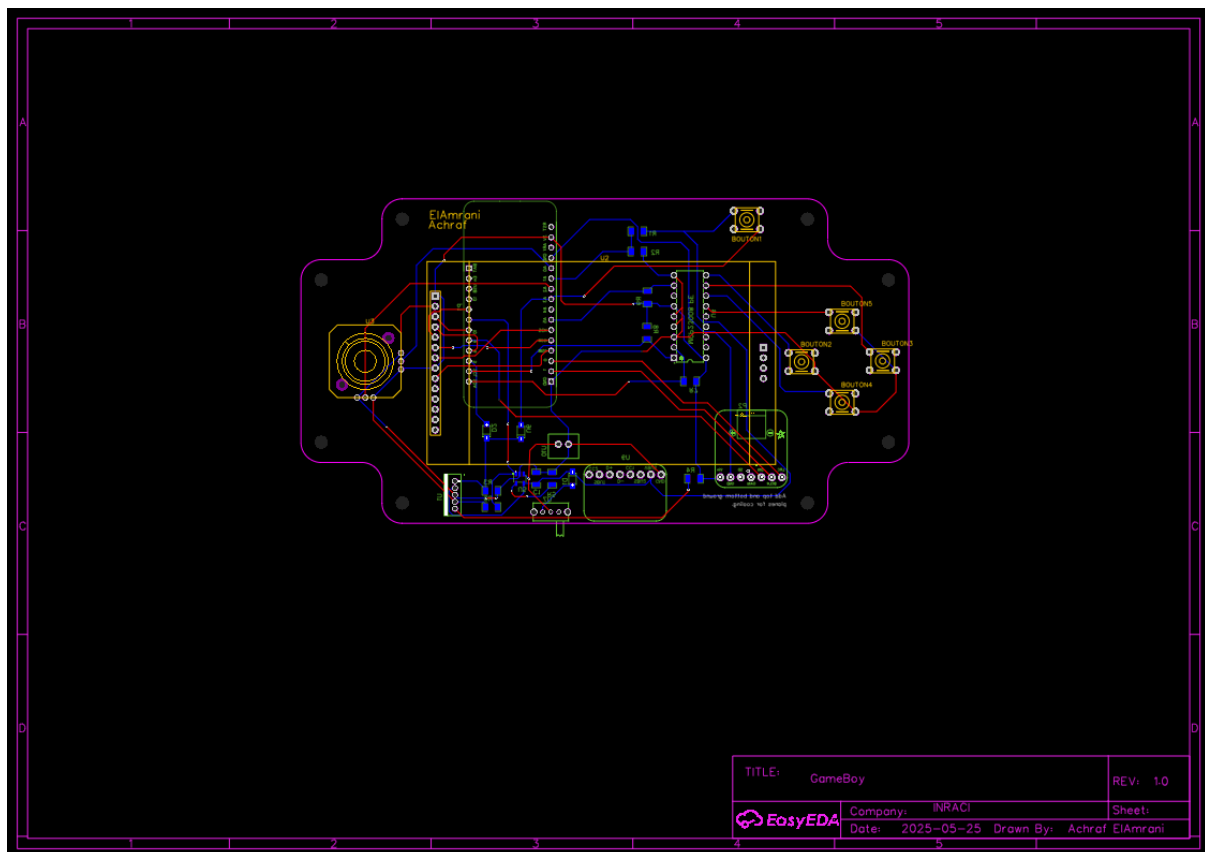
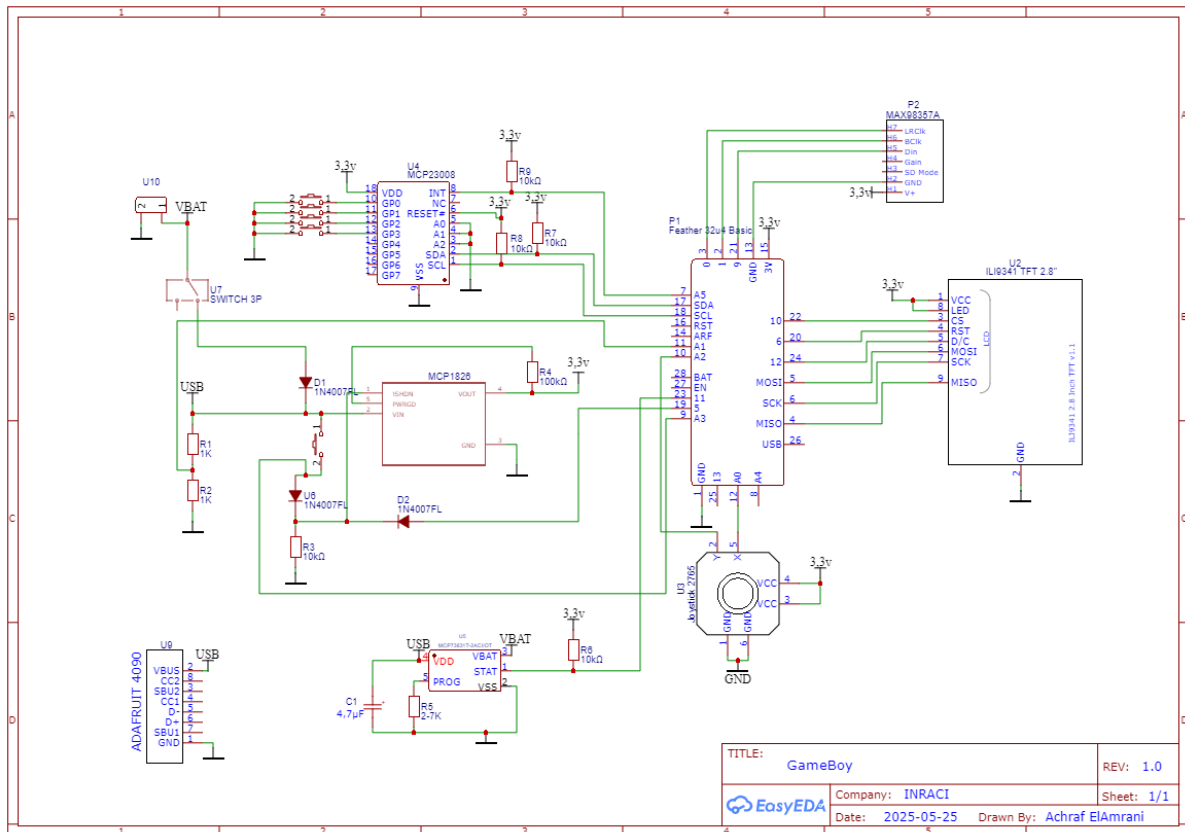
L'ensemble des livres d'électronique et d'informatique de Monsieur Mazzeo :

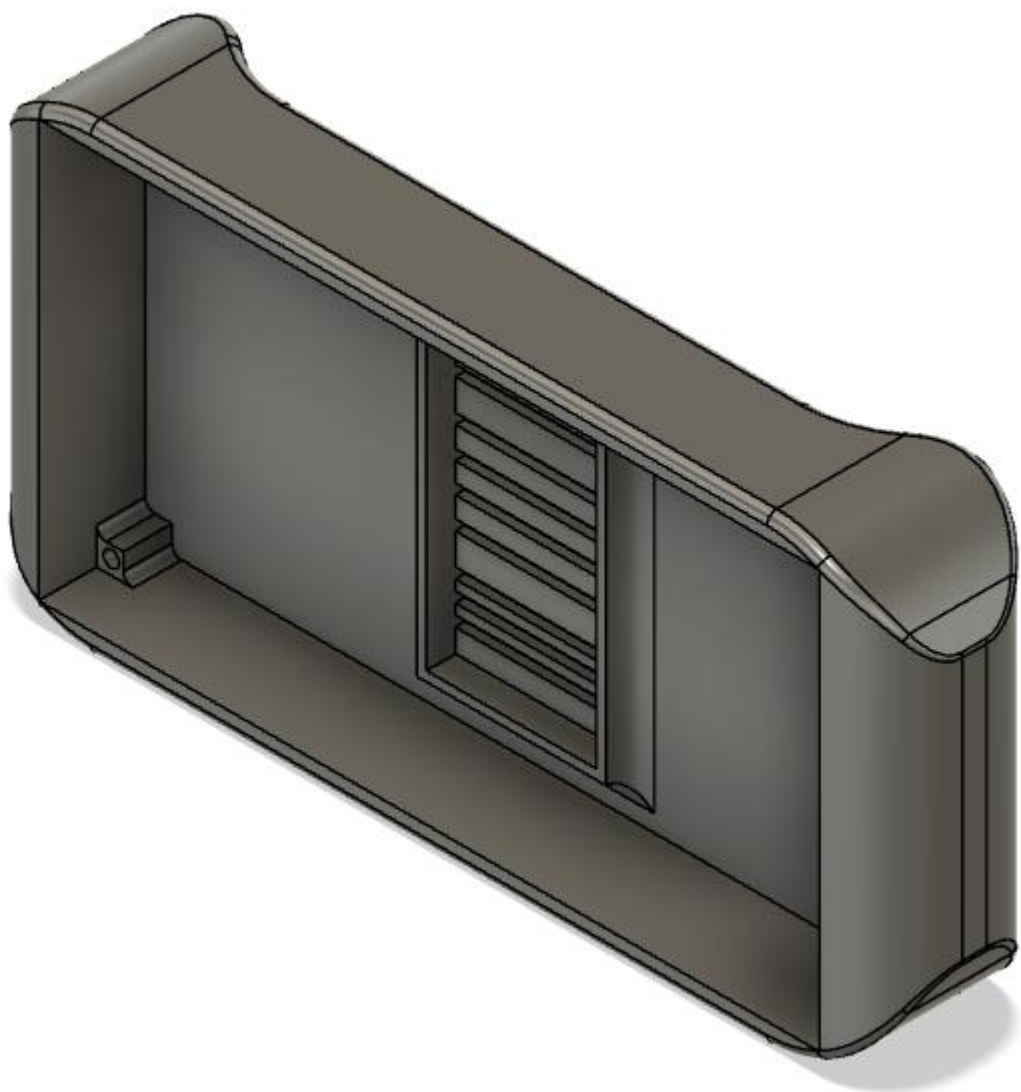
- Electronique 1,2 et 3 ,5èmeEo édition 2018 par R.Mazzeo
- Informatique Embarqué 5èmeEo édition 2023 par R.Mazzeo
- Informatique Embarqué 6èmeEo édition 2024 par R.Mazzeo

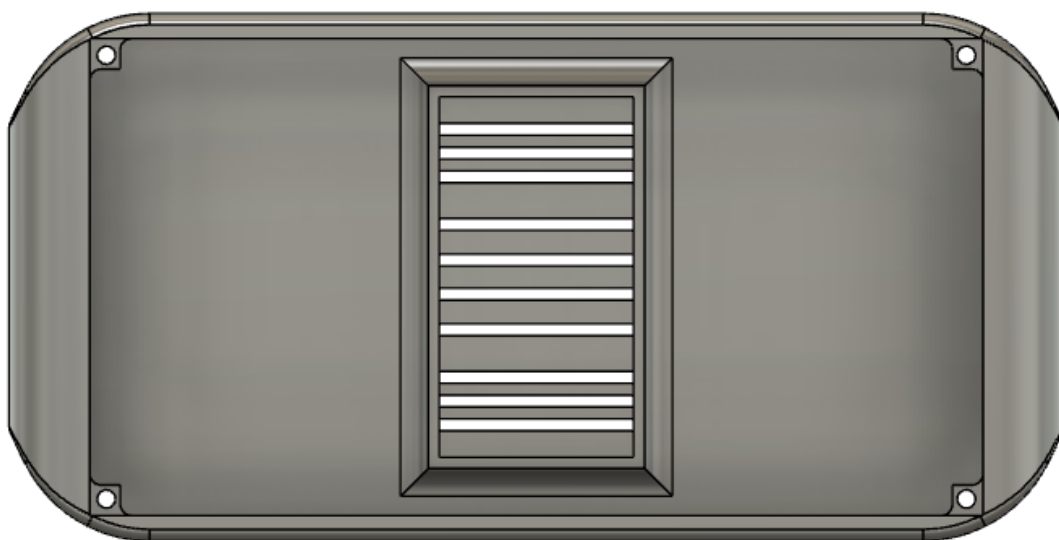
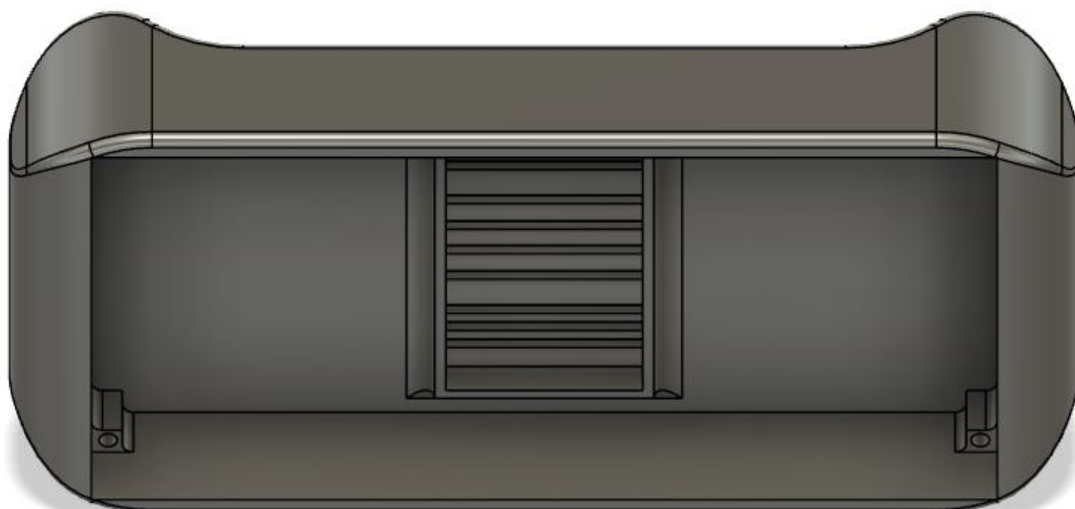
L'ensemble de tous les cours donnés à l'INRACI :

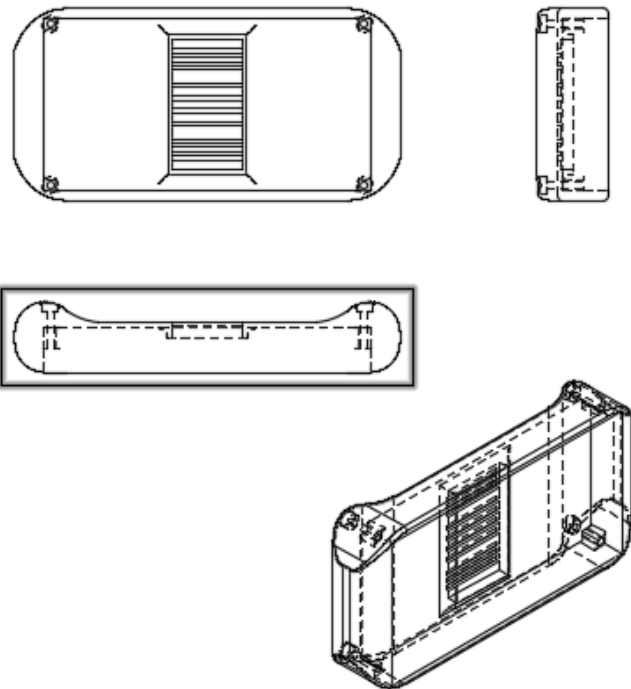
- Le cours de mathématique de 6ème, 2024-25 par Monsieur Senga
- Le cours de C.P de 6ème, 2024-25 par Monsieur Pochet
- Le cours d'électronique de 6ème, 2024-25 par Monsieur Pochet

Annexe :



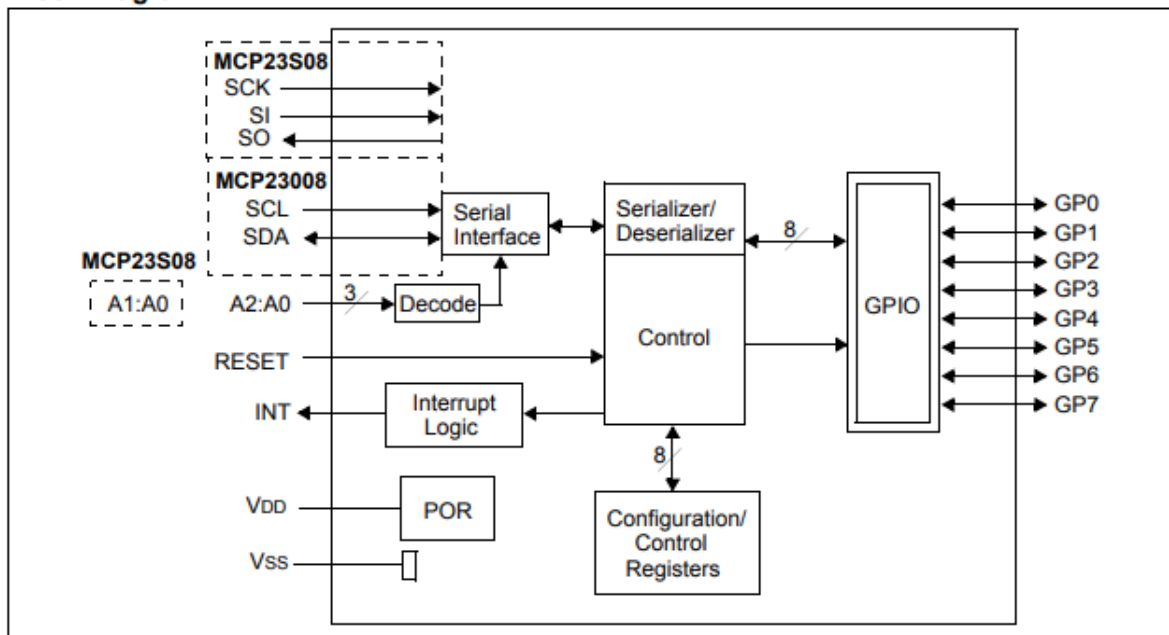




			
Dept.	Technical reference	Created by Achraf B Amrani 04/02/2025	Approved by
		Document type	Document status
		Title gameboy achraf	DWG No.
		File	Date of issue
			Sheet 1/1

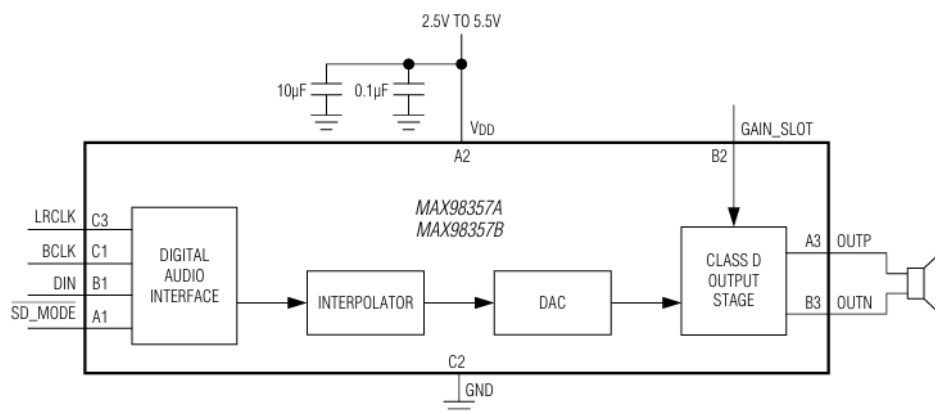
MCP23008 :

Block Diagram



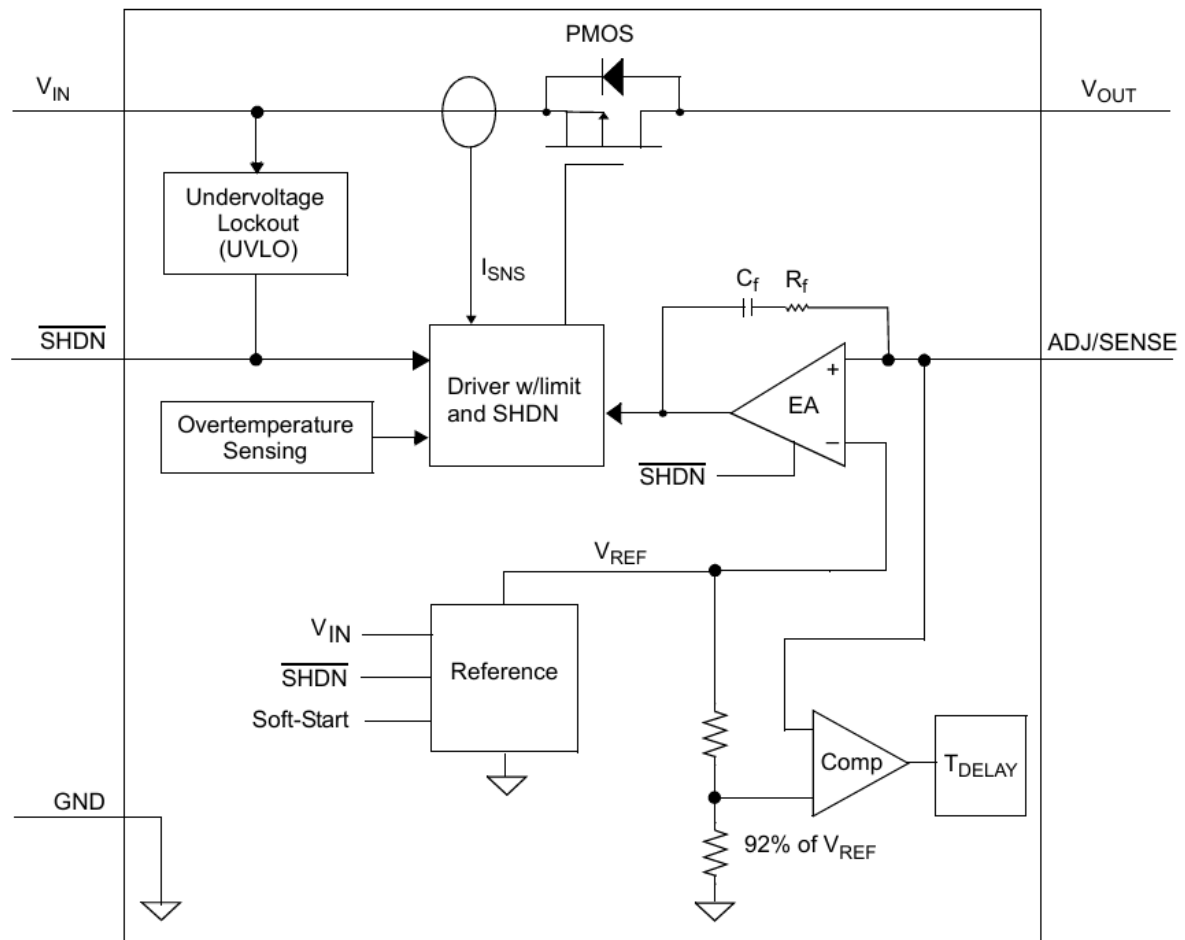
<https://ww1.microchip.com/downloads/en/DeviceDoc/MCP23008-MCP23S08-Data-Sheet-20001919F.pdf>

MAX98357A :



[MAX98357A-MAX98357B.pdf](#)

MCP1826 :

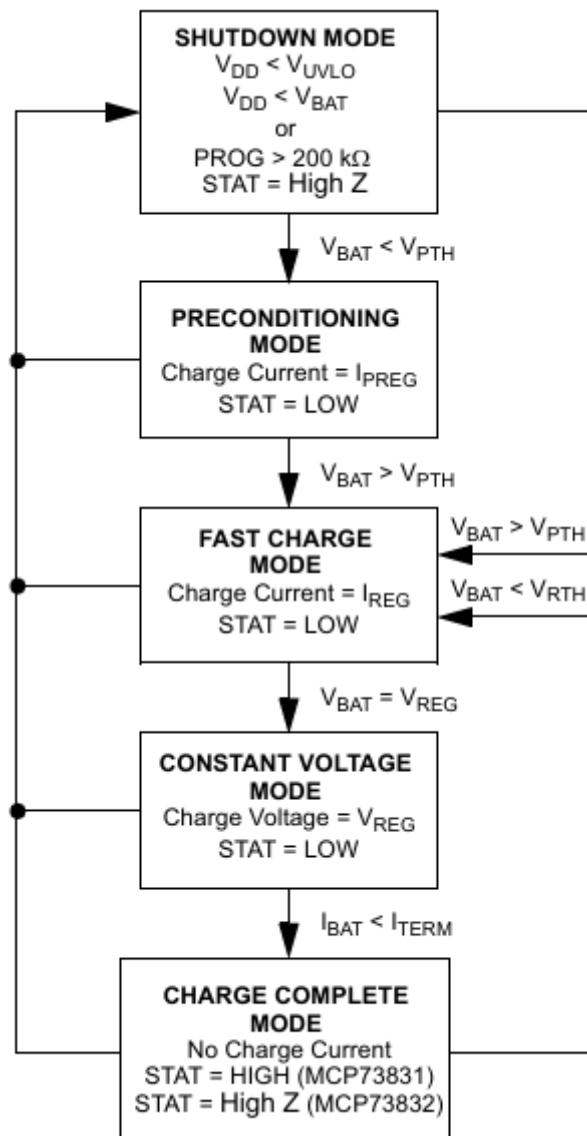


<https://ww1.microchip.com/downloads/en/DeviceDoc/22057B.pdf>

MINI 2-AXIS ANALOG THUMBSTICK :

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/813/2765_Web.pdf

MCP73831/2 :



<https://ww1.microchip.com/downloads/en/DeviceDoc/MCP73831-Family-Data-Sheet-DS20001984H.pdf>

ADAFRUIT USB C BREAKOUT BOARD :

https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/320/4090_Web.pdf

FEATHER 32U4 BASIC PROTO ATMEGA :

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-feather-32u4-basic-proto.pdf>

