

In.Ra.Ci

Avenue Jupiter,188

1190 Bruxelles

Technique de qualification

Electronique

Année scolaire 2024-2025

Radio Réveil

Matheo Patino Loya

Avenue Du Parc 32

1060 Saint-Gilles

Dans un premier temps, j'aimerais remercier tout le personnel éducatif de l'Inraci que j'ai eu durant mon entrée à cette école.

Ensuite, j'aimerais exprimer ma gratitude à M. Mazzeo pour tout le savoir qu'il nous a transmis au fil des deux années passées avec lui.

À mon promoteur M. Kapita, je le remercie de m'avoir tenu la main jusqu'au bout et de m'avoir donné son temps pour mon projet.

En dernier temps, ma famille et mon meilleur ami Grégory de m'avoir aidé et poussé à finir mon projet jusqu'au bout.

Table des matières

1 Introduction	5
2 Les caractéristiques	6
2.1 Générale	6
2.2 Electrique	6
2.3 Mécanique	6
3 Les Principes mis en jeu	7
3.1 Principe de réception d'un signal FM	7
3.2 Principe d'amplificateur Classe D	11
4 Le Schéma bloc	15
5 L'étude détaillée et les mesures	17
5.1 Schéma complet	17
5.2 Etude du Max98357A	18
5.2.1 Mesure du MAX98357A	25
5.3 Etude du RDA5807M	28
5.3.1 Mesure du RDA5807M	33
5.4 Bus I2C	37
5.5 L'encodeur rotatif	41
5.5.1 Mesure de l'encodeur rotatif	46
6 La Programmation	48
6.1 L'ordinogramme général	48
6.2 Le programme	49
7 La fabrication	52
8 La mise en point	60
9 La conclusion	65
Bibliographie	66
Les annexes	70
Schéma de principe sans connecteur	70
PCB Sérigraphie V1	71
PCB Sérigraphie V2	72
Croquis	73
Liste du matériel	75

1 Introduction

Dans le cadre de mon travail de fin d'études à l'INRACI, j'ai choisi de concevoir un radio-réveil en mettant en pratique les compétences techniques acquises tout au long de ma formation.

Ce projet a pour objectif principal de développer de combiner deux fonctionnalités essentielles :

- La réception de la radio FM
- Un système de réveil fonctionnel

En supplément, l'appareil affichera l'heure et la date en temps réel, le tout sur un écran graphique. Des dispositifs tournants permettront de régler le volume et de changer de station radio. Deux boutons seront ajoutés pour notamment afficher l'heure, la date, le jour, et pour activer, désactiver l'alarme.

Ce projet m'est venu en tête pendant un cours avec M. Mazzeo et est resté là depuis. La radio est une technologie qui a révolutionné le monde par sa capacité à envoyer et recevoir de l'information sans fil sur de longues distances.

Donc, pour comprendre cette technologie, j'ai eu envie d'en faire mon projet, pour acquérir les bases de la radio. Certes, la radio est un sujet assez compliqué à comprendre, notamment pour ses aspects techniques, mathématiques et physiques. Mais cela m'a motivé à en faire mon projet.



Lien QR GitHub

2 Les caractéristiques

2.1 Générale

Écran TFT 2.8" (ILI9341)

2 Encodeur Rotatif (KY-040)

Fusible

2 Bouton Poussoir

Antenne

Port USB type C

2.2 Electrique

Précision : ± 2 sec/jour

Autonomie : ~ 3 ans (pile CR1225)

Fréquence : 80 – 108 MHz

Sensibilité : 1.4 – 1.8 μV

Impédance antenne : 50 Ω

Sortie : 3.2W

Résolution : 30 crans/tour

Courant de consommation : ~ 200 mA

Alimentation : 9V

2.3 Mécanique

Matériaux : Planche de bois Multiplex

Dimension : 13.5 cm x 9.5 cm

3 Les Principes mis en jeu

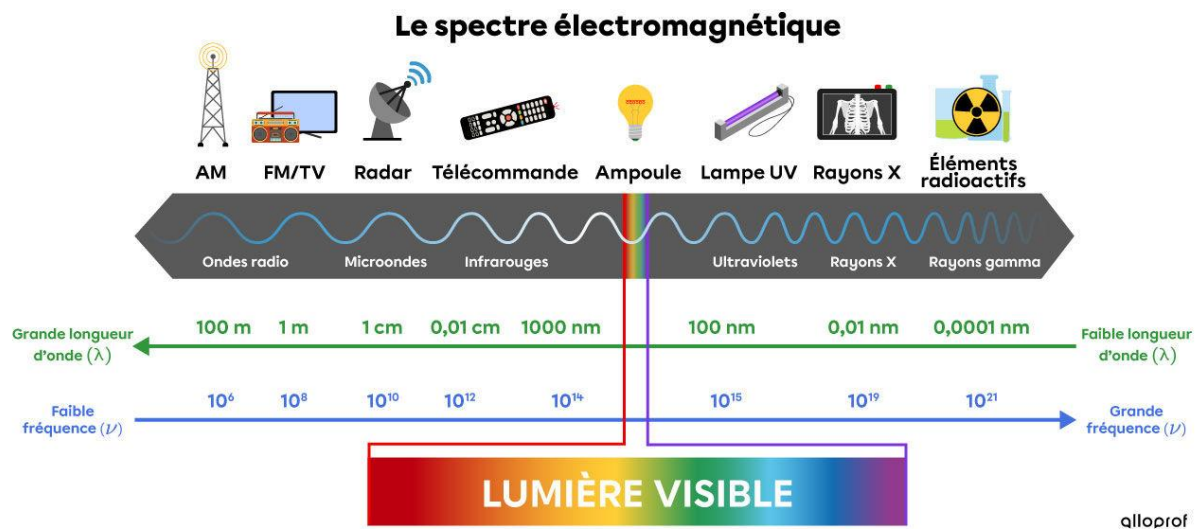
3.1 Principe de réception d'un signal FM

Principe du spectre électromagnétique

Pour comprendre comment fonctionne la réception d'un signal FM, il faut d'abord comprendre les bases du fonctionnement d'une radio.

Le spectre électromagnétique, c'est l'ensemble des ondes électromagnétiques classées selon leur fréquence et leur longueur d'onde.

Il existe de nombreux types d'ondes dans ce spectre, les ondes radio, les micro-ondes, la lumière visible, les rayons X, etc.



Source : <https://cms.alloprof.qc.ca/sites/default/files/styles/1920w/public/2022-03/is1137-2-spectre-electromagnetique.jpg?itok=KwB8EG7k> Date de consultation (24/05/25)

Les ondes radio sont utilisées pour transmettre de l'information sans fil grâce aux ondes électromagnétiques. Celles-ci font partie du spectre électromagnétique. Sa plage de fréquences va de 3 kHz à 300 GHz.

La vitesse de ces ondes dans le vide est de 300 000 km/s, soit à peu près la vitesse de la lumière.

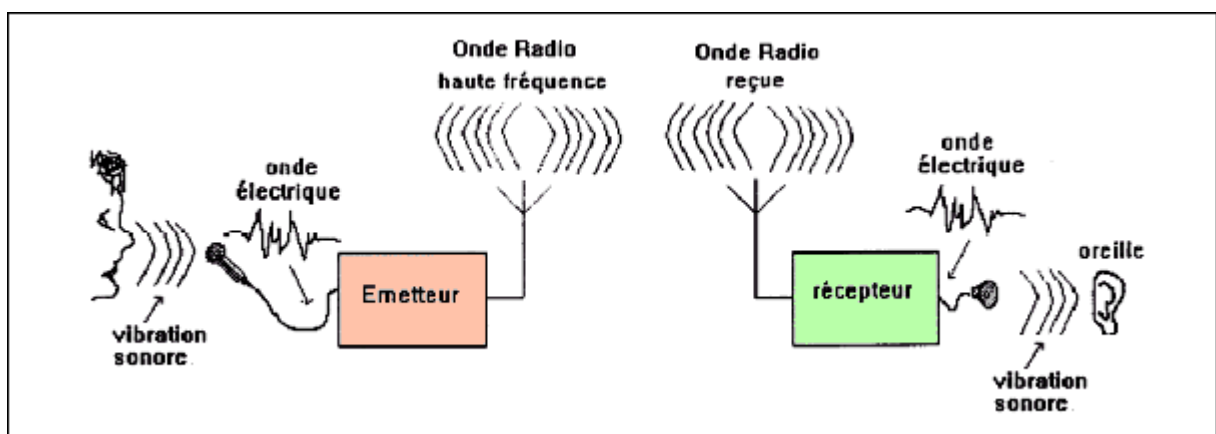
Pour fabriquer une onde électromagnétique, on utilise de l'électricité, lorsque que l'on fait circuler du courant électrique dans une bobine on crée un champ magnétique si on inverse le courant électrique on inverse le champ magnétique, on le fait varier. C'est ce qui se passe à l'intérieur d'une antenne, on y fait circuler un courant alternatif qui produit une onde électromagnétique.

Maintenant pour transporter la voix grâce aux ondes électromagnétiques ce n'est pas très compliqué, lorsque qu'on parle dans un micro on fait vibrer de l'air à une certaine fréquence, ce qui fait vibrer la membrane du micro, et derrière cette membrane il y a une bobine qui se déplace à l'intérieur d'un aimant. Les électrons de la bobine se mettent en mouvement, on crée un courant électrique de même fréquence que celle de la voix.

Donc il suffirait de transporter ce courant électrique dans l'antenne, ce qui émet des ondes électromagnétiques. Malheureusement la voix a une fréquence très basse, de 80 Hz à 180 Hz. Pour régler ce problème, il suffit de faire circuler du courant alternatif de haute fréquence grâce à un oscillateur, ce qui crée une onde porteuse pour transporter la voix.

On mélange le courant électrique de haute fréquence avec le signal électrique obtenu à l'aide de la voix de basse fréquence. On mélange un courant de haute fréquence (onde porteuse) avec le signal audio basse fréquence (voix) pour produire un signal FM. Du coup, l'antenne va transmettre une onde électromagnétique de haute fréquence modulée par la voix (FM).

Schéma



Source : https://i0.wp.com/promocom.r-e-f.org/wp-content/uploads/2015/11/10_1.gif Date de consultation (19/05/25)

Pour la réception d'un signal FM, l'antenne récupère l'onde électromagnétique émise, on obtient donc un courant alternatif de haute fréquence. Le problème, c'est que les fréquences sont tellement grandes, l'oreille humaine peut entendre de 20 Hz à 15 kHz, ce qui est trop faible pour entendre les hautes fréquences.

Pour régler ce problème, on va utiliser un démodulateur pour récupérer que le signal de la voix et ainsi couper les hautes fréquences, et avec ce signal on peut ainsi alimenter le haut-parleur et écouter la voix émise au début.

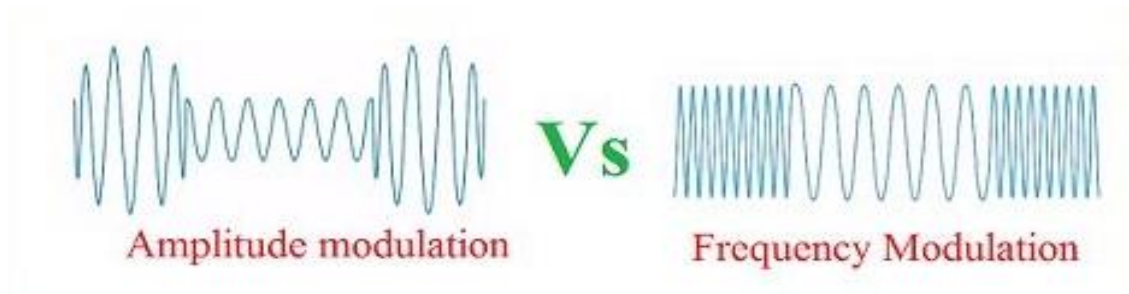
Les ondes électromagnétiques transportent le signal audio sous forme de variations de fréquence, qui sont captées par l'antenne d'un récepteur. Ce signal est ensuite démodulé pour enlever les variations de fréquence et juste récupérer le signal électrique de la voix pour ensuite l'entendre sur un haut-parleur.

Principe de la Fm et Am

De plus, la radio repose sur deux techniques principales de modulation pour transmettre le son la modulation d'amplitude (AM) et la modulation de fréquence (FM).

Ces deux méthodes influencent la qualité du son et la fiabilité de transmission. Avec la modulation d'amplitude (AM), c'est l'amplitude qui varie en fonction du signal audio. Elle permet une large portée de diffusion.

À l'inverse, la modulation de fréquence (FM) c'est la fréquence qui varie, tout en gardant une amplitude constante.



Source : <https://th.bing.com/th/id/R.7a028524f09b79cb0bfb383d5a6a4df5?rik=r%2fPeYTAIPuPyzg&riu=http%3a%2f%2felectronics-club.com%2fwp-content%2fuploads%2f2020%2f09%2fAM-vs-FM.jpg&ehk=FTt6Wd4UWJcLtHDBvMvX2aBUBDZga1KqjhUIGCT3sel%3d&risl=&pid=ImgRaw&r=0>

Date de consultation (19/05/25)

Avantages de la FM

La modulation de fréquence (FM) offre une qualité sonore nettement supérieure. En modulant la fréquence de l'onde porteuse plutôt que son amplitude, la FM réduit considérablement les interférences et parasites, ce qui donne un son de meilleure qualité.

Inconvénients de la FM

Cependant, la portée des émissions FM est limitée, ce qui a pour effets de restreint sa distance de portée

Avantages de l'AM

À l'inverse, la modulation d'amplitude (AM) permet une portée bien plus étendue.

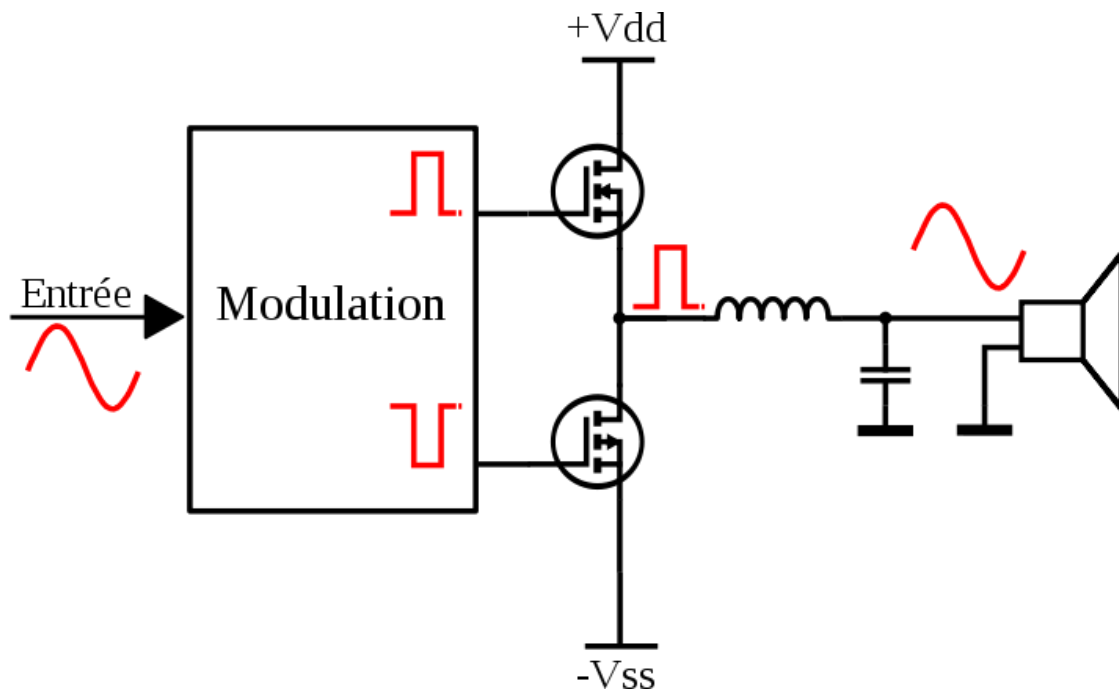
Inconvénients de l'AM

Toutefois, la modulation AM est plus vulnérable aux parasites et aux distorsions, ce qui peut dégrader la qualité sonore.

3.2 Principe d'amplificateur Classe D

L'amplificateur de classe D est un type d'amplificateur audio efficace, souvent utilisé dans les appareils portables, les systèmes audio compacts, il fonctionne selon un principe de commutation à haute fréquence, ce qui permet de réduire considérablement les pertes d'énergie.

Schéma bloc



Source : https://jaazz.files.wordpress.com/2011/01/classe_d.png?w=712 Date de consultation (20/05/25)

Fonctionnement de l'amplificateur de classe D

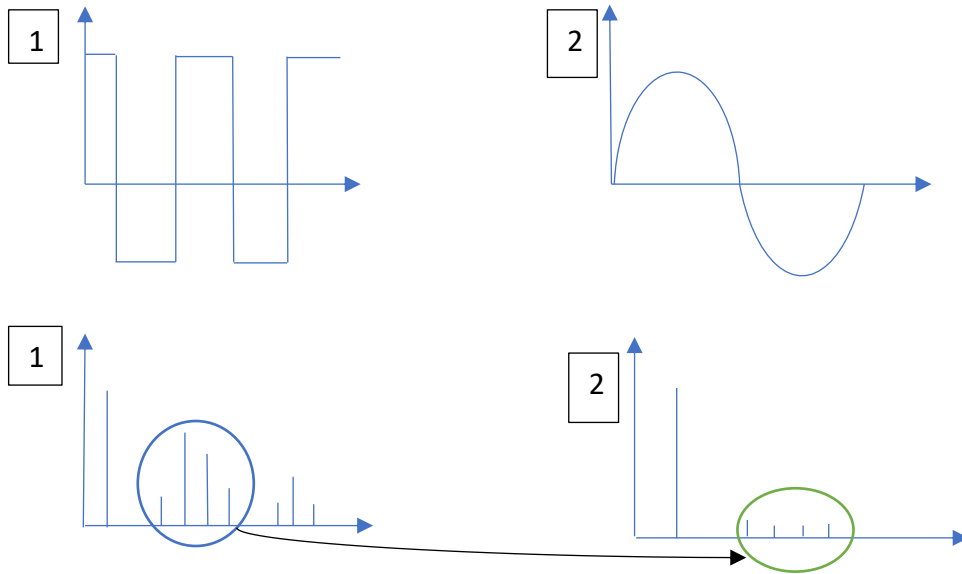
L'amplificateur de classe D transforme le signal audio en un signal à impulsions (PWM), grâce à un modulateur.

Ces impulsions sont envoyées à des transistors qui fonctionnent comme des interrupteurs ils sont soit ouverts, soit fermés, avec une alimentation symétrique.

Ensuite, un filtre passe-bas, composé d'une bobine et d'un condensateur, élimine les hautes fréquences liées à la fréquence de découpage.

Afin de permettre de reconstituer le signal audio d'origine, ce signal sera enfin envoyé dans un haut-parleur.

Schéma spectrale et temporelle



Explication

Sur le premier graphique, nous avons un signal numérique. Juste en dessous, c'est sa représentation spectrale. On remarque qu'on a une fondamentale, celle-ci correspond au signal audio. À la suite de la fondamentale suivent les harmoniques.

Elle correspond à la fréquence de découpage, ce qui, ensemble, crée un signal PWM ou MLI qui est modulé par rapport au signal audio d'origine.

Sur le deuxième graphique, on a filtré le signal pour retirer les harmoniques de la fréquence de découpage afin de n'avoir que le signal audio d'origine. Celle-ci est représentée par un sinus.

On remarque que, dans la représentation spectrale, il reste encore des harmoniques. Cela est dû au filtre, car il n'est pas parfait et n'enlève pas toutes les harmoniques. Cela peut causer de la distorsion dans le signal d'origine, soit une variation ou déformation du sinus.

Avantage

Son plus gros avantage est son rendement, qui est dû au transistor en commutation. Lorsque celui-ci est bloqué, sa consommation est de 0 car $I_c = 0$. À l'inverse, quand le transistor est saturé, sa consommation est de 0 car $V_{CE} = 0$ (0,2 V), ce qui évite la dissipation thermique ou presque.

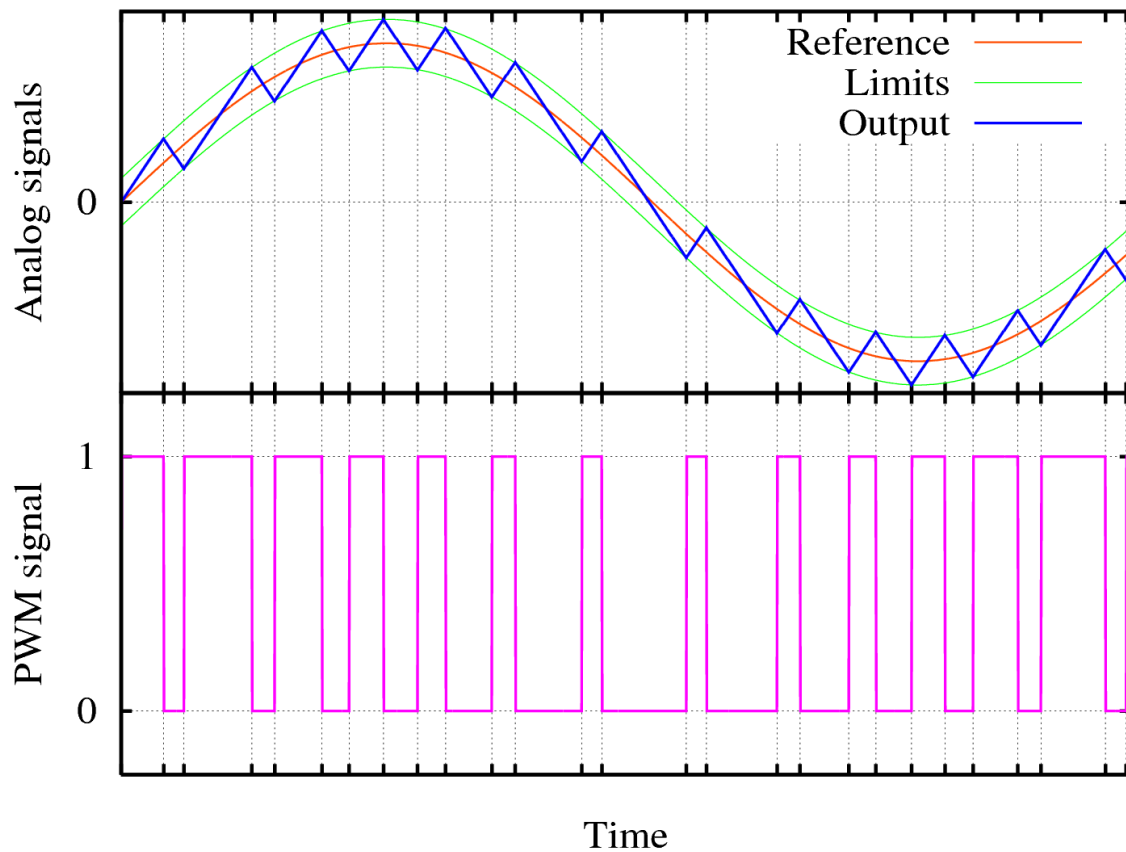
Moins encombrant, parfait pour des petits dispositifs elle occupe le moins d'espace possible, avec une bonne puissance de sortie pour sa taille.

Inconvénient

Distorsion due aux harmoniques qui passent car le filtre n'est pas parfait.

Un des plus grands inconvénients, c'est que l'amplification dépend de l'alimentation.

Schéma

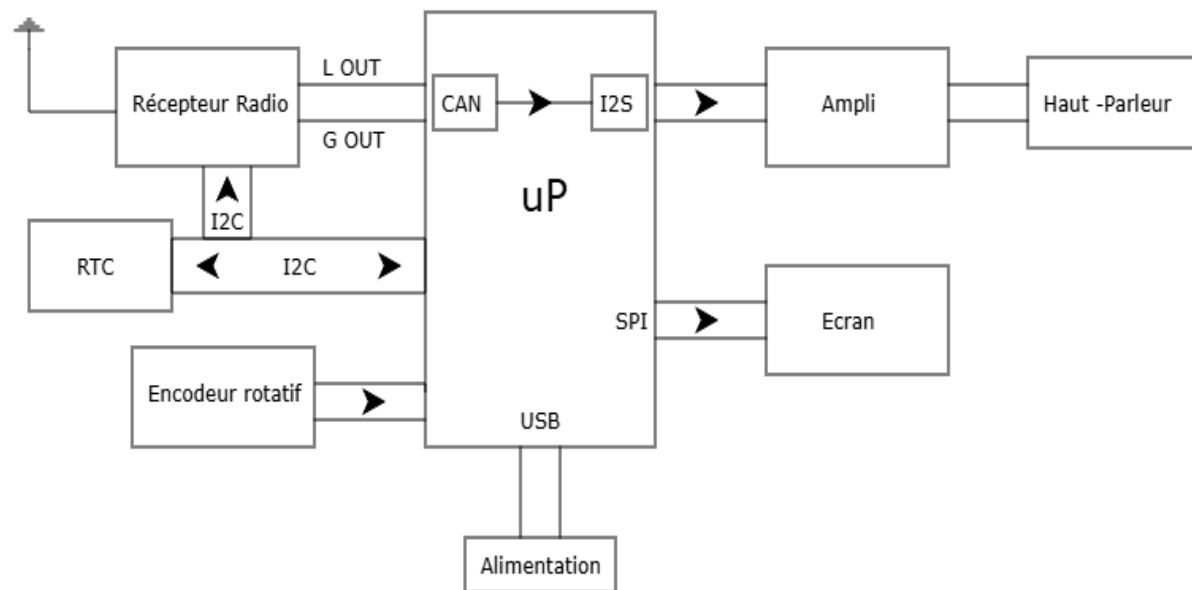


Source : https://upload.wikimedia.org/wikipedia/commons/e/e2/Delta_PWM.png Date de consultation (20/05/25)

Analyse

Sur l'image, on voit en haut un signal analogique, un sinus, et une dent de scie qui sert de référence. Grâce à cette référence, le signal PWM est modulé par rapport au sinus d'origine. La sortie alterne entre impulsions 1 et 0. Le rapport cyclique du signal PWM varie pour suivre la forme du sinus, ce qui permet de conserver l'information du signal analogique.

4 Le Schéma bloc



Explication

Récepteur radio

Ce bloc capte les signaux radio analogiques reçus par l'antenne et les convertit en signaux audio analogiques. Il reçoit un signal analogique provenant de l'antenne et fournit en sortie un signal audio analogique pour le microprocesseur.

RTC

Le bloc RTC conserve l'heure, la date et le jour, même lorsque l'appareil est éteint, grâce à une pile interne. Il communique avec le microprocesseur via un bus I2C bidirectionnel, échangeant des données numériques pour fournir ou recevoir des informations.

Encodeur rotatif

L'encodeur rotatif permet à l'utilisateur d'interagir avec la radio, par exemple pour changer le volume ou la station. Il génère des impulsions numériques en sortie.

Alimentation

L'alimentation fournit une tension continue stable et le courant nécessaire pour faire fonctionner tous les blocs.

Écran

L'écran reçoit des données et des commandes numériques envoyées par le microprocesseur via un bus SPI. Son rôle est d'afficher les informations provenant des différents blocs, comme l'heure, la station radio ou le volume.

Amplificateur audio

L'amplificateur reçoit un signal audio numérique via le bus I2S, puis amplifie ce signal audio pour pouvoir alimenter le haut-parleur. Sa sortie est un signal audio analogique amplifié.

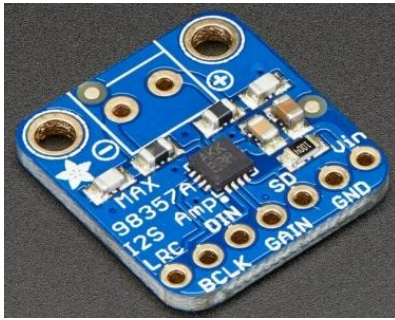
Haut-parleur

Le haut-parleur convertit le signal audio électrique amplifié en ondes sonores audibles. Il reçoit en entrée un signal analogique amplifié et produit un son audible en sortie.

Microprocesseur (μ P)

Le microprocesseur reçoit le signal audio analogique provenant du récepteur radio et le convertit en signal numérique grâce à un convertisseur analogique-numérique (ADC) de 16 bits. Il gère la communication avec le RTC via le bus I2C. Il envoie le signal audio numérique à l'amplificateur via le protocole I2S, et communique avec l'écran par SPI pour afficher les informations. Enfin, il lit les impulsions numériques envoyées par l'encodeur rotatif. Les entrées du microprocesseur sont donc à la fois analogiques (audio) et numériques (encodeur, RTC), tandis que ses sorties sont numériques (audio via I2S, commandes pour l'écran via SPI).

5.2 Etude du Max98357A



Source : https://cdn.myikass.com/images/8b57d900-7b49-4cf7-810a-401e00bd0425/25604e38-825c-41f1-b189-39866da1e7f8/image_900.webp Date de consultation (20/05/25)

Spécifications

Puissance de sortie : 3,2 W à 4 Ω , 10 % THD, 1,8 W à 8 Ω , 10 % THD, avec alimentation 5V

Fréquences d'échantillonnage I2S de 8kHz à 96kHz

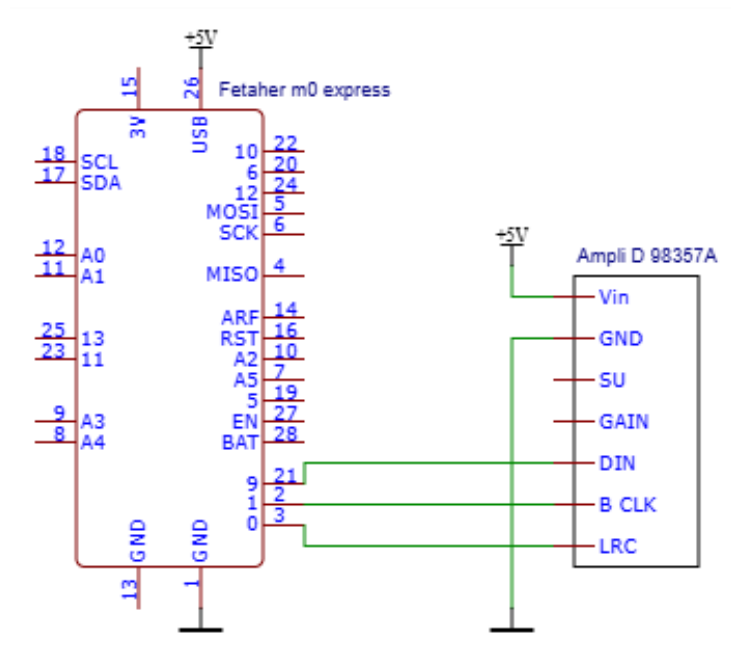
Cinq gains sélectionnables par broche : 3 dB, 6 dB, 9 dB, 12 dB, 15 dB

Protection contre l'arrêt thermique



Lien QR Spécification

Schéma



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 le 1 pour le canal droit et 0 pour le canal gauche.

BCLK (Bit Clock)

Elle sert à indiquer à l'amplificateur quand lire les données sur la broche de données, actif sur flanc montant, synchronise les bits de données.

DIN (Data In)

Cette entrée contient les données audios numériques pour les canaux gauche et droit.

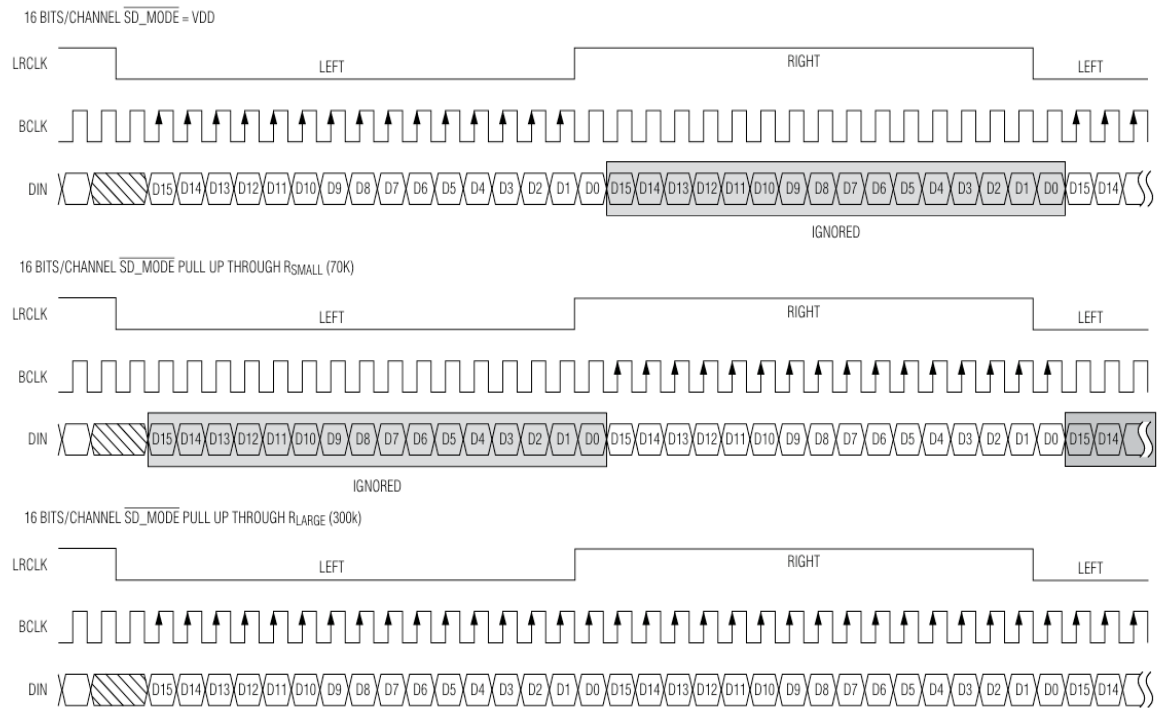
GAIN (Gain Select)

Celle-ci permet de sélectionner le gain de l'amplificateur, c'est-à-dire le niveau d'amplification du signal audio.

SD (Shutdown)

Utilisée pour le mode d'arrêt, mais elle est également utilisée pour définir le canal de sortie.

Datasheet



Analyse

Sur cette image, on constate que trois signaux sont présents : LRC, BCLK, DIN. Chacun d'entre eux correspond bien au rôle qui lui a été attribué sur 16 bits. De plus, la synchronisation se fait sur les flancs montants.

MAX98357A/MAX98357B

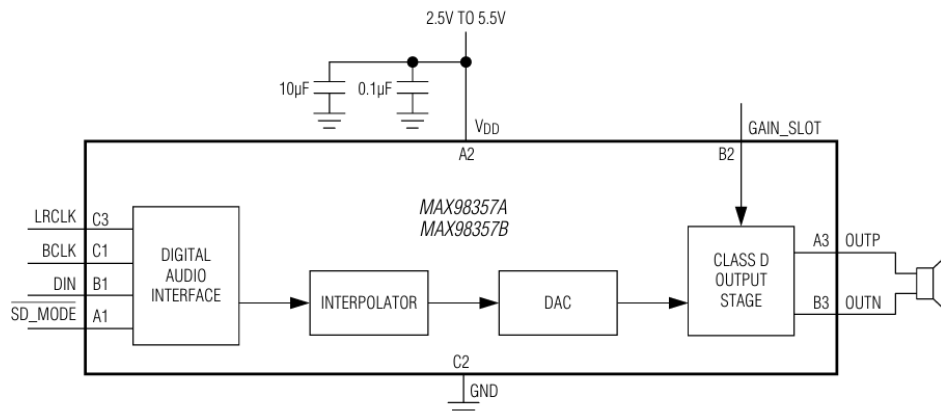
PCM Input Class D Audio Power Amplifiers

ELECTRICAL CHARACTERISTICS (continued)

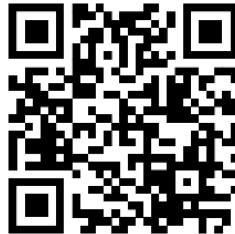
($V_{DD} = 5V$, $V_{GND} = 0V$, $GAIN_SLOT = V_{DD}$, $BCLK = 3.072MHz$, $LRCLK = 48kHz$, speaker loads (Z_{SPK}) connected between OUTP and OUTN, $Z_{SPK} = \infty$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted. Typical values are at $T_A = +25^\circ C$.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Power (Note 3)	P_{OUT}	THD+N 10%, gain = 12dB	$Z_{SPK} = 4\Omega + 33\mu H$	3.2		W
			$Z_{SPK} = 8\Omega + 68\mu H$	1.8		
			$Z_{SPK} = 8\Omega + 68\mu H$, $V_{DD} = 3.7V$	0.93		
		THD+N = 1%, gain = 12dB	$Z_{SPK} = 4\Omega + 33\mu H$	2.5		
			$Z_{SPK} = 8\Omega + 68\mu H$	1.4		
			$Z_{SPK} = 8\Omega + 68\mu H$, $V_{DD} = 3.7V$	0.77		
Total Harmonic Distortion + Noise	THD+N	$f = 1kHz$, $P_{OUT} = 1W$, $T_A = +25^\circ C$, $Z_{SPK} = 4\Omega + 33\mu H$, WLP		0.02	0.06	%
		$f = 1kHz$, $P_{OUT} = 1W$, $T_A = +25^\circ C$, $Z_{SPK} = 4\Omega + 33\mu H$, TQFN		0.02		
		$f = 1kHz$, $P_{OUT} = 0.5W$, $T_A = +25^\circ C$, $Z_{SPK} = 8\Omega + 68\mu H$		0.013		
Dynamic Range	DR	A-weighted, $V_{RMS} = 2.54V$, 24- or 32-bit data		105		dB
Output Noise	V_N	A-weighted, 24- or 32-bit data (Note 4)		25		μV_{RMS}
Gain (Relative to a 2.1dBV Reference Level)	A_V	$GAIN_SLOT = GND$ through 100k Ω	14.4	15	15.6	dB
		$GAIN_SLOT = GND$	11.4	12	12.6	
		$GAIN_SLOT = unconnected$	8.4	9	9.6	
		$GAIN_SLOT = V_{DD}$	5.4	6	6.6	
		$GAIN_SLOT = V_{DD}$ through 100k Ω	2.4	3	3.6	
Current Limit	I_{LIM}			2.8		A
Efficiency	ϵ	$Z_{SPK} = 8\Omega + 68\mu H$, THD+N = 10%, $f = 1kHz$, gain = 12dB		92		%
DAC Gain Error				1		%
Frequency Response			-0.2		+0.2	dB
Class D Switching Frequency	f_{OSC}			330		kHz
Spread-Spectrum Bandwidth				± 20		kHz
DAC DIGITAL FILTERS						
VOICE MODE IIR LOWPASS FILTER (LRCLK < 30kHz)						
Passband Cutoff	f_{PLP}	Ripple limit cutoff	0.443 $\times f_S$			Hz
		-3dB cutoff	0.446 $\times f_S$			
Stopband Cutoff	f_{SLP}			0.464 $\times f_S$		Hz
Stopband Attenuation		$f > f_{SLP}$	75			dB

Schéma bloc interne de l'amplificateur classe D



Source <https://cdn-shop.adafruit.com/product-files/3006/MAX98357A-MAX98357B.pdf> Date de consultation (12/11/2025)



Lien Datasheet

Explication des blocs

Digital Audio Interface

C'est le premier bloc du circuit. Il reçoit le signal audio numérique et le prépare pour traitement.

Interpolateur

Il améliore la qualité du signal numérique avant qu'il ne soit converti en signal analogique.

DAC (Digital-to-Analog Converter)

Ce composant transforme le signal audio numérique (0 et 1) en un signal analogique continu.

Étape de sortie de classe D

Amplifie le signal analogique pour qu'il soit suffisamment puissant pour faire fonctionner un haut-parleur.

Les sorties OUTP (A3) et OUTN (B3) envoient ce signal amplifié vers le haut-parleur connecté.

GAIN_SLOT

Permet de régler l'amplification grâce à la configuration du gain.

Alimentation

Fournit l'énergie au circuit entre 2,5 V et 5,5 V.

Programme Test

```
#include <I2S.h>

const int amplitude = 500; // Amplitude du signal audio
short sample = amplitude; // Valeur actuelle de l'échantillon

void setup() {
  Serial.begin(115200);
  if (!I2S.begin(I2S_PHILIPS_MODE, 44100, 16)) {
    Serial.println("Erreur I2S !");
    while(true);
  }
}

void loop() {
  I2S.write(sample); // Envoi sur canal gauche
  I2S.write(sample); // Envoi sur canal droit
  sample = -sample; // Inversion pour générer une onde carrée
  delayMicroseconds(1136); // Pause pour fréquence ~440 Hz
}
```

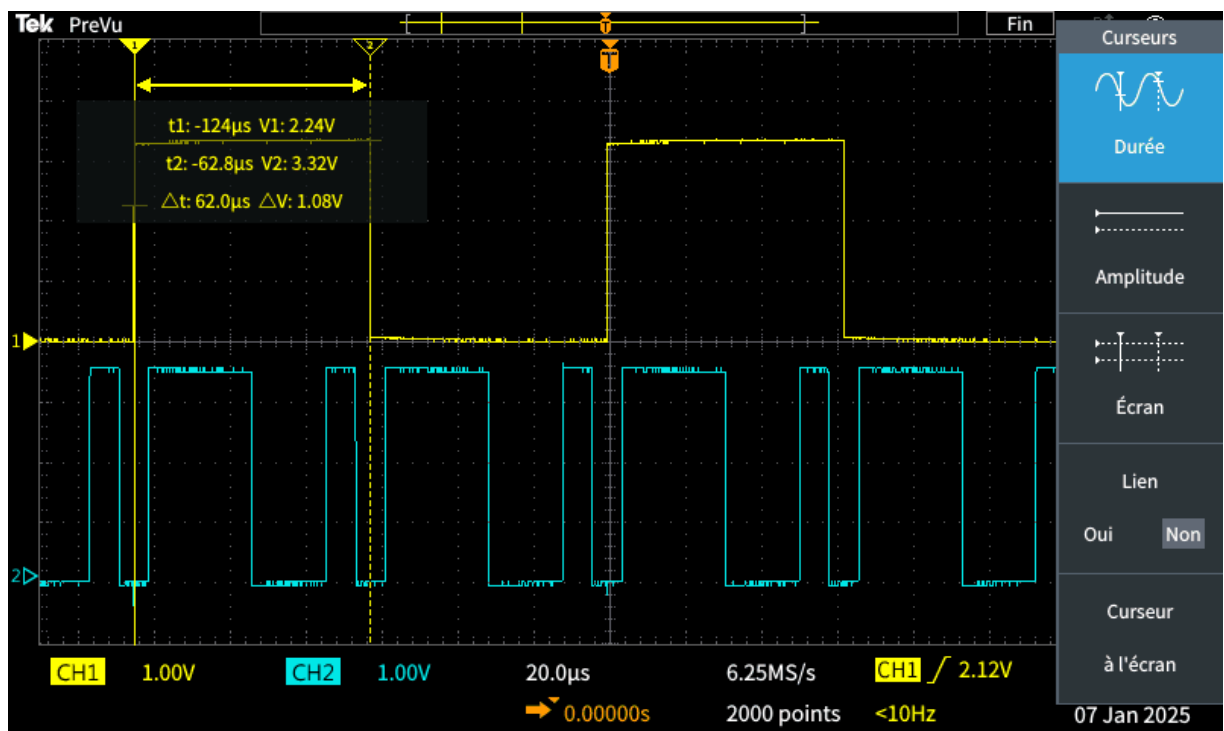
Source : [ChatGPT](#) date de consultation (20/11/2025)

Explication

Ce programme crée un son (la), l'amplitude de ce signal est de 500, et on l'envoie dans les canaux gauche et droit de l'I2S. Comme on a que du positif, on ne peut pas avoir de son, alors on crée du -500 pour avoir une oscillation.

La fréquence qu'on veut est 440 Hz, donc la période d'un cycle complet est d'environ 2270 microsecondes. Dans le code, on fait un délai de 1136 microsecondes, car c'est la moitié de la période puisque qu'il alterne entre -500 et +500.

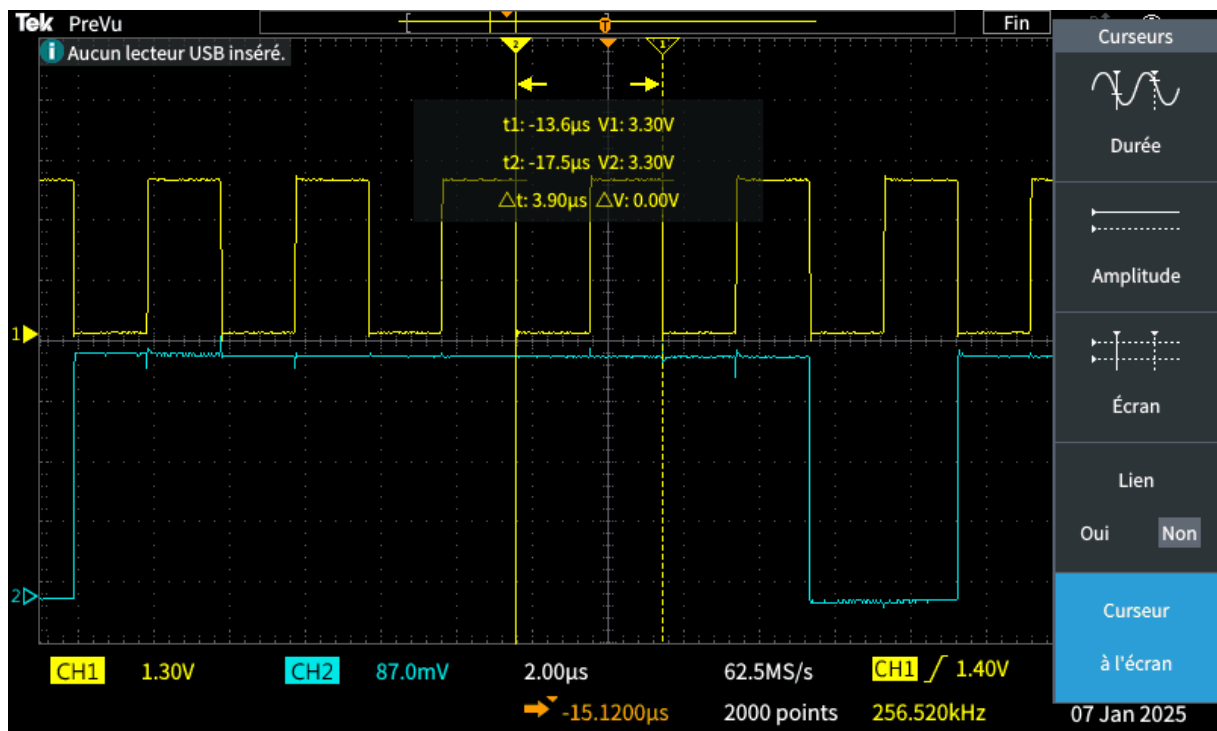
5.2.1 Mesure du MAX98357A



En jaune : LRC

En bleu : DIN

La largeur d'impulsion est de 62µs pour le canal droit



En jaune : Bclk

En bleu : DIN

BCLK a une fréquence de 256,520 KHz, la durée d'un bit dure 3.90µs

Analyse

Le signal BCLK permet d'envoyer chaque bit au bon moment, le signal audio transmis est une onde carrée qui alterne entre +500 et -500 à chaque demi-période, et comme chaque valeur est envoyée deux fois (gauche puis droite), le son est le même des deux côtés.

On voit aussi sur le canal CH1 (jaune) la période BCLK mesurée à 3,90 μ s donc la fréquence est bien $1 / 3,90 \mu\text{s} \approx 256,41 \text{ kHz}$, et sur CH2 (bleu) on voit le motif binaire 01010101 typique d'une onde carrée numérique, avec le signal LRC qui est bas pour le canal gauche et haut pour le canal droit.

Donc en conclusion, les signaux mesurés montrent que la transmission I2S fonctionne bien. De plus, la datasheet nous permet de vérifier l'allure des signaux qui sont en cohérence avec les mesures faites sur l'ampli MAX98357A.

5.3 Etude du RDA5807M



Source: https://s13emagst.akamaized.net/products/51986/51985910/images/res_26ceb7fe6f25d598bae5399645cb6c0b.jpg
Date de consultation (20/05/25)

Spécifications

Type : Récepteur FM stéréo

Fréquence : 50 à 115 MHz

Alimentation : 1.8 V à 3.6 V

Courant de consommation : 20mA

Sortie audio : Analogique stéréo (L/R)

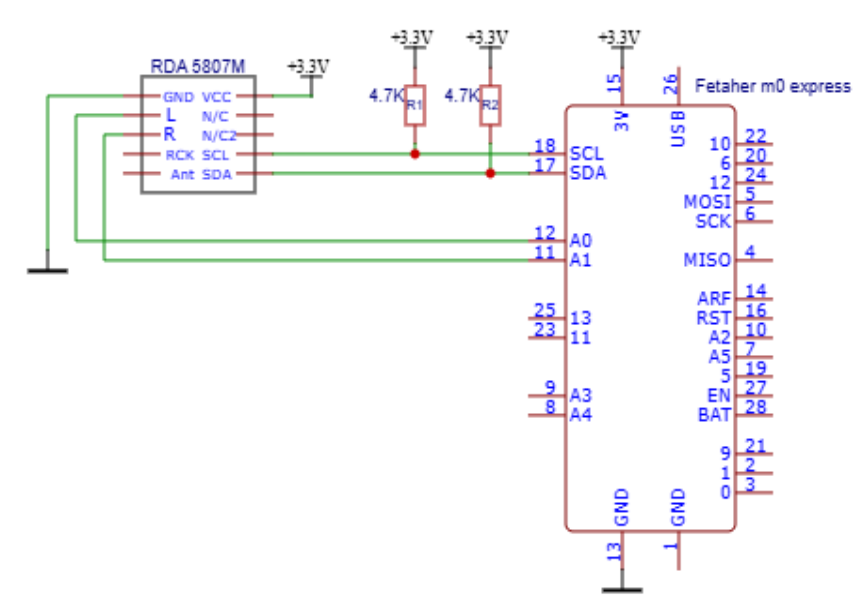
Interface : I2C

Impédance d'entrée : 50 Ω



Lien QR spécifications

Schéma



Rôle des signaux

SCL (Serial Data Line) : SCL est la ligne d'horloge elle est utilisée pour synchroniser tous les transferts de données sur le bus I2C sur flanc montant

SDA (Serial Clock Line) : SDA est la ligne de données

L : Sortie audio canal gauche connecter au μ p pour une conversion Analogique numérique

R : Sortie audio canal droit connecter au μ p pour une conversion Analogique Numérique.

R1 et R2 : les résistances de pull-up assure quel soit au niveau logique haut pour une communication stable

N/C (no connect) : pas de connexion

Ant : Broche du signal analogique de l'antenne

Programme Test

```
include <Arduino.h>
#include <Wire.h>
#include <radio.h>
#include <RDA5807M.h>
// ----- Paramètres fixes -----
#define FIX_BAND RADIO_BAND_FM ///< La bande radio utilisée par ce
programme est la FM.
#define FIX_STATION 8930        ///< La station qui sera syntonisée est
89.30 MHz.
#define FIX_VOLUME 10           ///< Le volume défini par ce programme est
de niveau 10.

RDA5807M radio; // Création d'une instance de la classe pour la puce
RDA5807M

/// Configuration de la radio FM
/// avec affichage des informations de débogage sur le port série
void setup() {

    // Configuration FM pour l'Europe
    radio.setup(RADIO_FMSPACING, RADIO_FMSPACING_100); // Espacement des
stations (Europe)
    radio.setup(RADIO_DEEMPHASIS, RADIO_DEEMPHASIS_50); // Déaccentuation
du signal (Europe)

    // Initialisation de la puce radio
    if (!radio.initWire(Wire)) {
        Serial.println("Aucune puce radio détectée.");
        delay(4000);
    };

    // Appliquer les paramètres fixes de la radio
    radio.setBandFrequency(FIX_BAND, FIX_STATION);
    radio.setVolume(FIX_VOLUME);
    radio.setMono(false);
    radio.setMute(false);
} // Fin de setup

void loop() {

}
```

Source : [GitHub - mathertel/Radio : Une librairie Arduino pour contrôler les puces radio FM comme SI4703, SI4705, RDA5807M, TEA5767](#). Date de consultation (20/11/25)

Datasheet

4 Receiver Characteristics

Table 4-1 Receiver Characteristics

(VDD = 3 V, TA = 25 °C, unless otherwise specified)

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNIT
General specifications							
F _{in}	FM Input Frequency Range	Adjust BAND Register		50		115	MHz
V _{rf}	Sensitivity ^{1,2,3}	S/N=26dB	50MHz	-	1.4	1.8	μ V EMF
			65MHz	-	1.2	1.5	
			88MHz	-	1.2	1.5	
			98MHz	-	1.3	1.5	
			108MHz	-	1.3	1.5	
			115MHz	-	1.3	1.8	
IP3 _{in}	Input IP3 ⁴	AGCD=1		80	-	-	dB μ V
α_{am}	AM Suppression ^{1,2}	m=0.3		60	-	-	dB
S ₂₀₀	Adjacent Channel Selectivity	± 200 KHz		50	70	-	dB
S ₄₀₀	400KHz Selectivity	± 400 KHz		60	85	-	dB
V _{AFL} ; V _{AFR}	Audio L/R Output Voltage ^{1,2} (Pins LOU _T and ROU _T)	Volume [3:0] =1111		-	360	-	mV
S/N	Maximum Signal to Noise Ratio ^{1,2,3,5}		Mono ²	55	57	-	dB
			Stereo ⁶	53	55	-	
α_{SCS}	Stereo Channel Separation			35	-	-	dB
R _L	Audio Output Loading Resistance	Single-ended		32	-	-	Ω
THD	Audio Total Harmonic Distortion ^{1,3,6}	Volume[3:0] =1111	R _{load} =1K Ω	-	0.15	0.2	%
			R _{load} =32 Ω	-	0.2	-	
α_{AOI}	Audio Output L/R Imbalance ^{1,6}			-	-	0.05	dB
R _{mute}	Mute Attenuation Ratio ¹	Volume[3:0]=0000		60	-	-	dB
BW _{audio}	Audio Response ¹	1KHz=0dB	Low Freq ⁹	-	100	-	Hz
		± 3 dB point	High Freq	-	14	-	
Pins FMIN, LOU _T , ROU _T							
V _{com_rfin}	Pins FMIN Input Common Mode Voltage				0		V
V _{com}	Audio Output Common Mode Voltage ⁸			1.0	1.05	1.1	V

Notes: 1. F_{in}=65 to 115MHz; F_{mod}=1KHz; de-emphasis=75 μ s; MONO=1; L=R unless noted otherwise;

2. Δf =22.5KHz; 3. B_{AF} = 300Hz to 15KHz, RBW <=10Hz;

5. P_{RF}=60dB μ V;

6. Δf =75KHz,fpilot=10%

8. At LOU_T and ROU_T pins

4. |f₂-f₁|>1MHz, f₀=2xf₁-f₂, AGC disable, F_{in}=76 to 108MHz;

7. Measured at V_{DMS} = 1 m V, f_{RF} = 65 to 108MHz

9. Adjustable

Figure 5-1. I²C Interface Write Timing Diagram

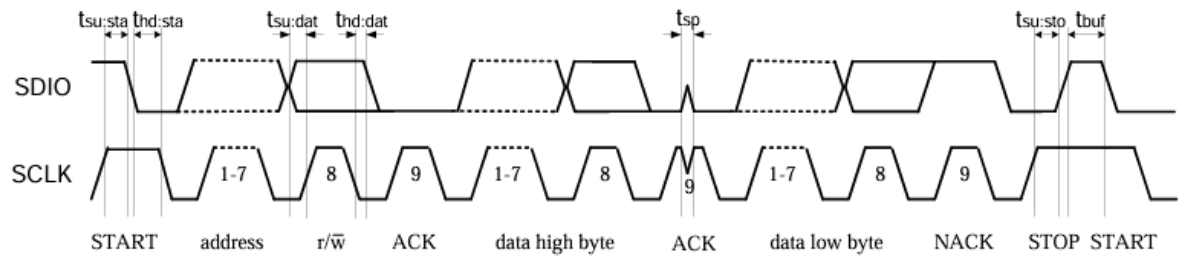


Figure 5-2. I²C Interface Read Timing Diagram

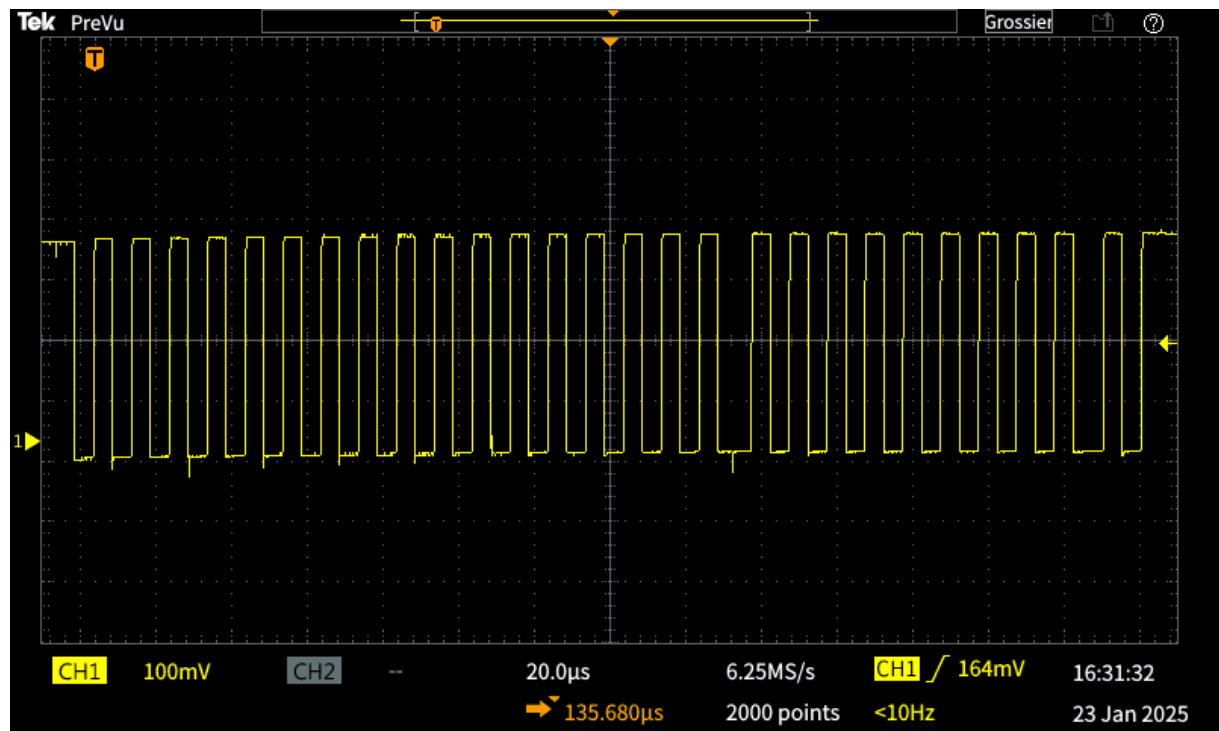


Lien QR datasheet

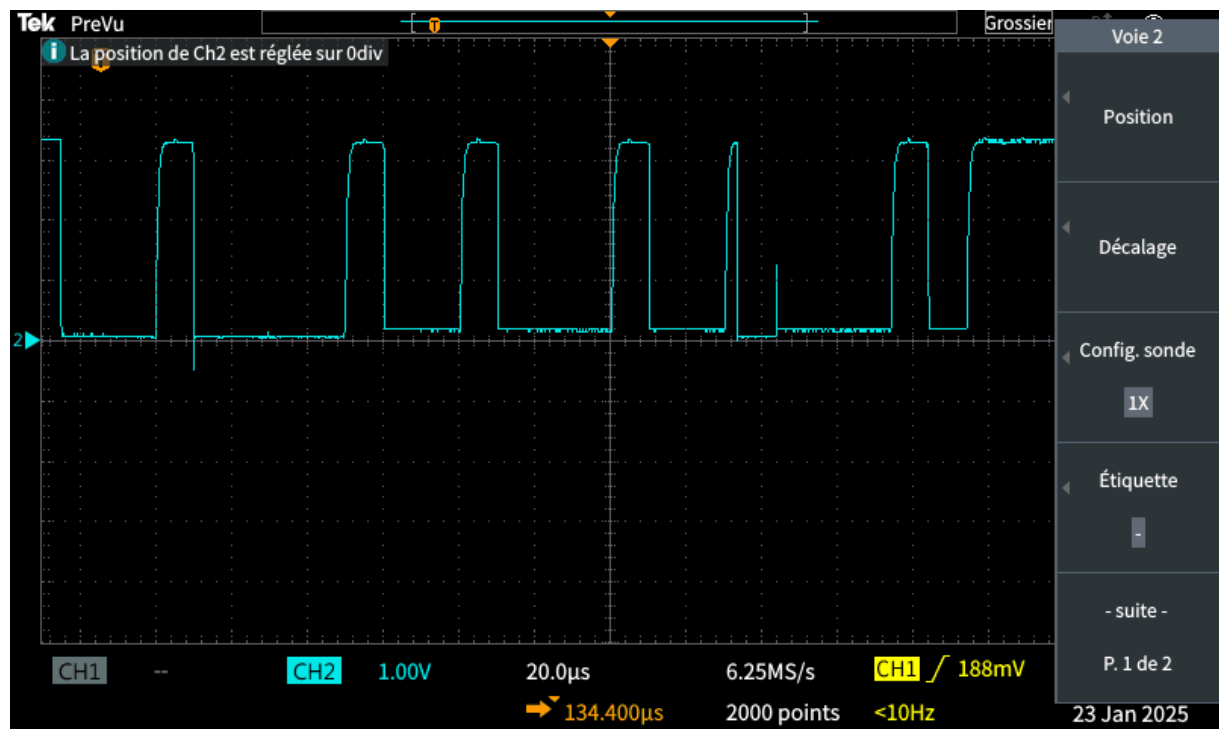
5.3.1 Mesure du RDA5807M

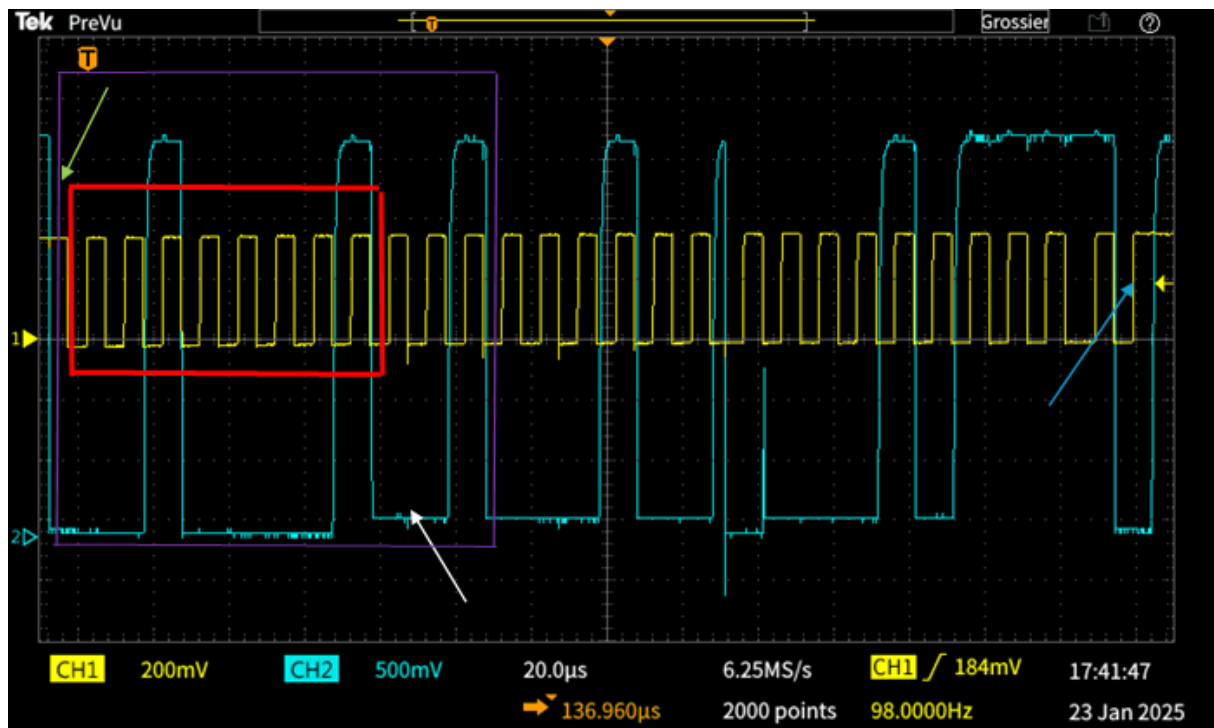
Mesures

SCL



SDA





Analyse

Dans cette trame, la condition de départ est pressante, adresse, R/W, ACK, condition d'arrêt.

En vert, la condition de départ

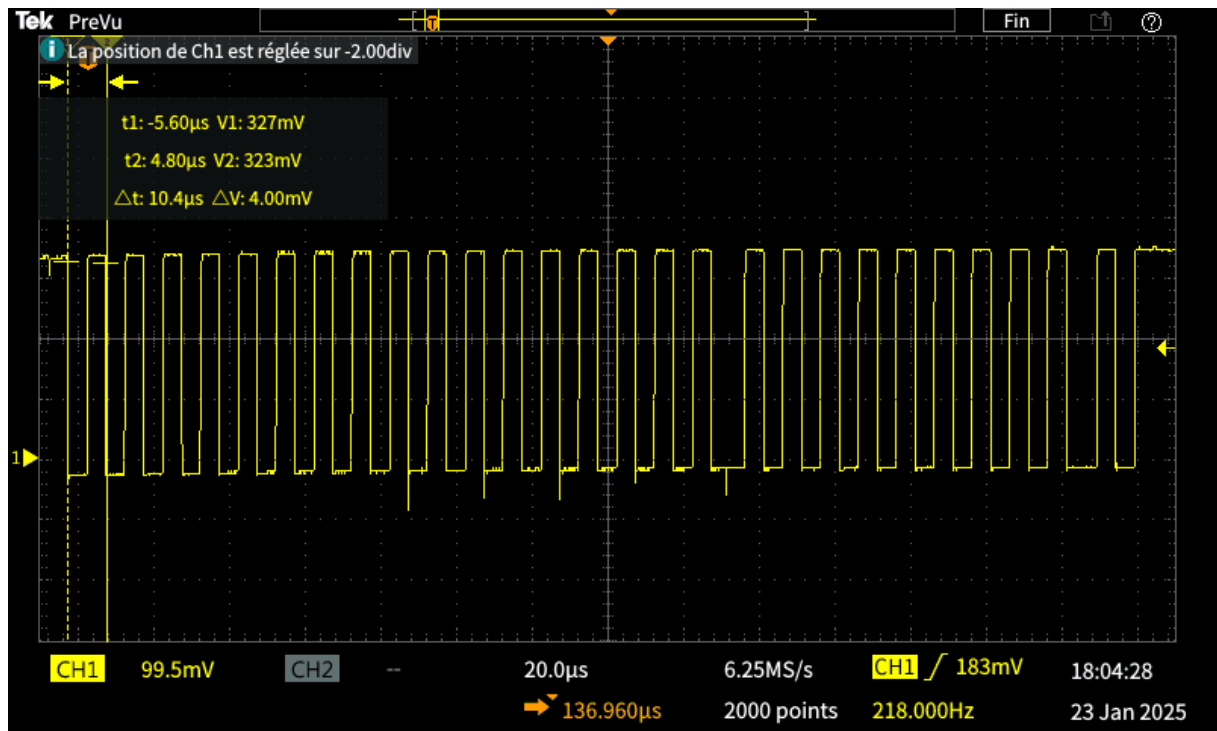
En rouge, l'adresse envoyée bit par bit

En blanc, le R/W et ACK

En bleu, la condition d'arrêt

En mauve, le premier octet

L'adresse en binaire correspond à 0b0010001, contient 7 bits ce qui veut dire que le 8e et 9e bits correspondent à R/W et ACK qui eux sont à 0. D'un côté, cela signifie que la Feather envoie des données (écriture) au RDA5807 (module radio) et puisque ACK est à 0, cela confirme que l'esclave a bien reçu l'info.



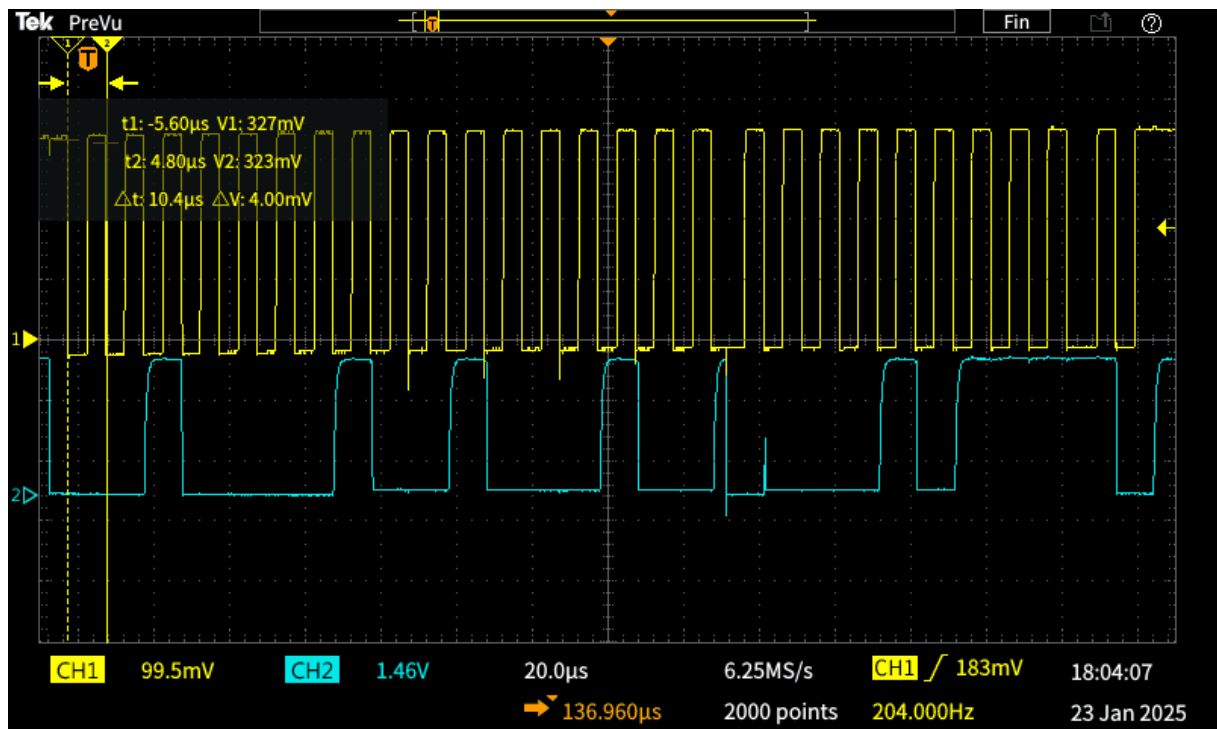
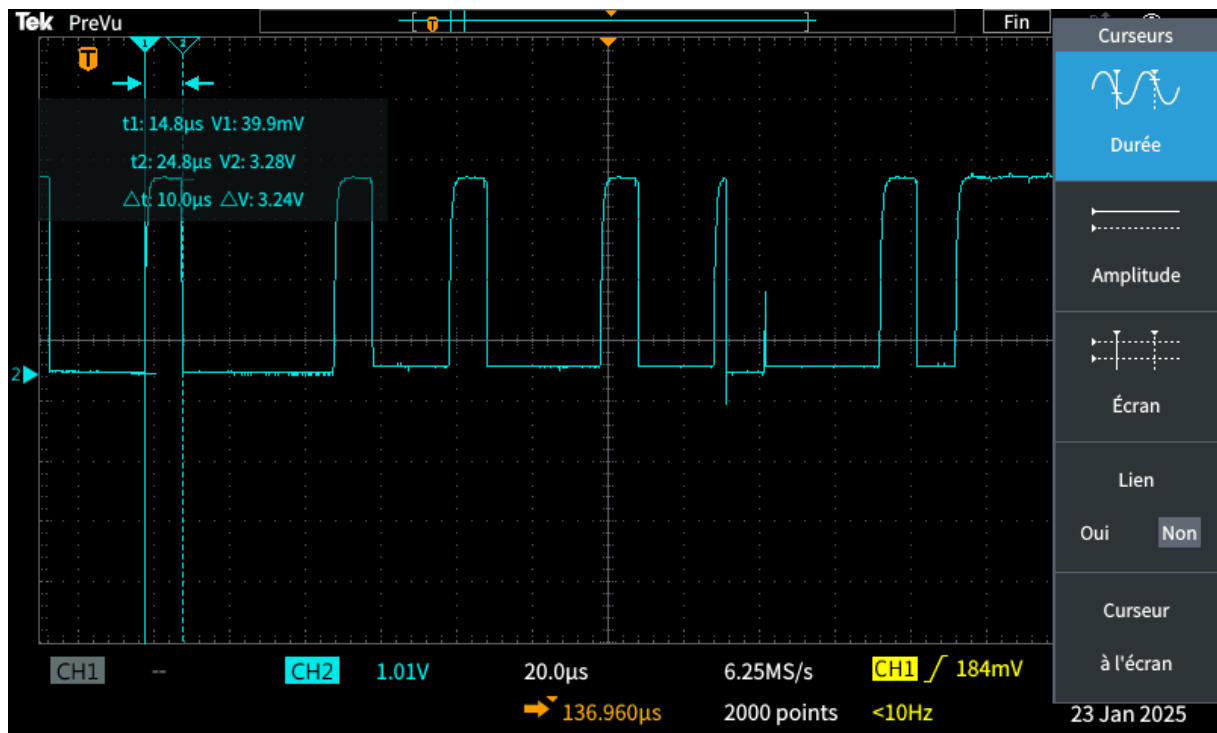
Durée d'un bit

$\Delta t = 10.4 \mu s$

Baud rate

$1 / \Delta t = 1 / 10,4 \times 10^{-6} s = 96\,154 \text{ bauds}$

Images supplémentaires



5.4 Bus I2C

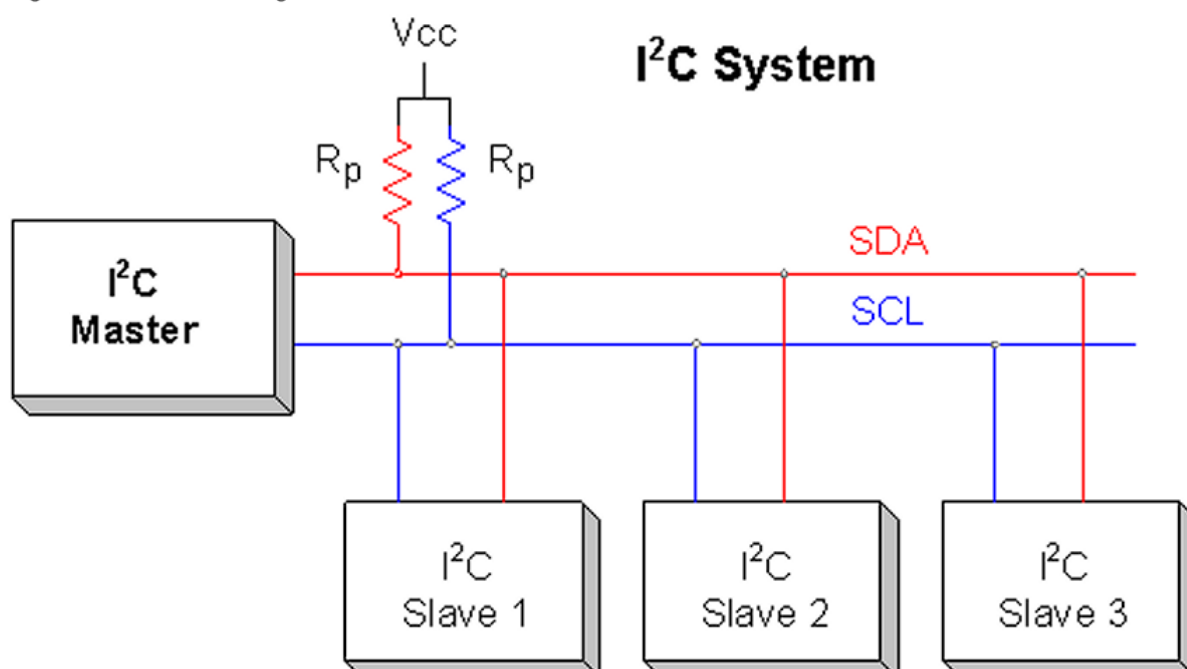
Le I²C (Inter-Integrated Circuit) est un protocole de communication développé par Philips au début des années 1980, c'est un bus série synchrone destiné à permettre à plusieurs circuits intégrés numériques "périphériques" de communiquer avec un ou plusieurs "contrôleurs".

Maître (Master) : Contrôle la communication, en générant les signaux d'horloge et en initiant les transferts de données, "contrôleurs".

Esclaves (Slaves) : Répondent aux commandes du maître. Un bus I2C peut avoir plusieurs esclaves connectés à une paire de lignes SDA et SCL, "périphériques".

Schéma

Figure 1. I²C Block Diagram

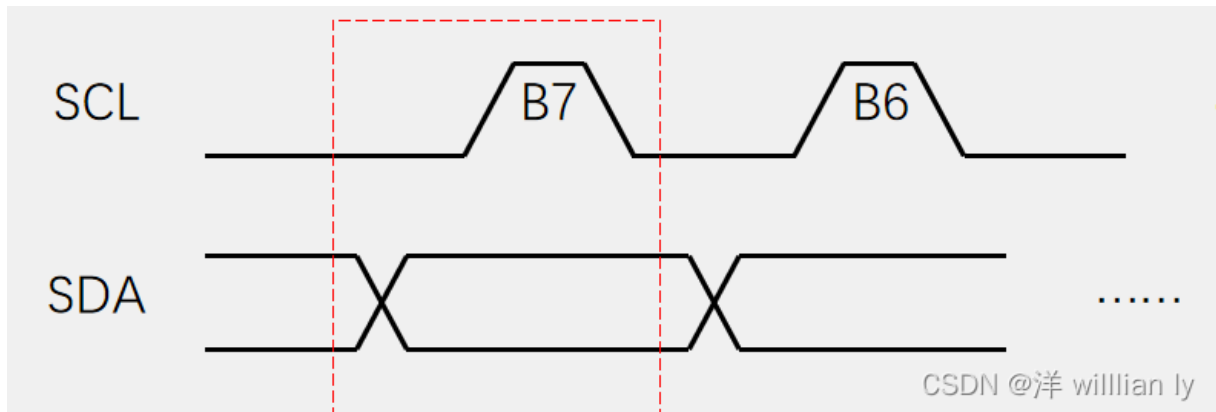


Configuration basic du I2C, un maitre peut avoir plusieurs esclaves de plus seul 2 lignes de communication sont nécessaires.

Source : I2Cm.dita date de consultation (2/05/25)



Lien QR Datasheet



Source : <https://img-blog.csdnimg.cn/5b8198682abf41e4a44646eb8f2e6eb0.png> Date de consultation (2/05/25)

SCL (Serial Clock Line) : SCL est la ligne d'horloge elle est utilisée pour synchroniser tous les transferts de données sur le bus I²C sur flanc montant, scl est généré par le maitre.

SDA (Serial Data Line) : SDA est la ligne de données elle transporte les bits de données envoyés ou reçus entre l'esclaves et le maitre

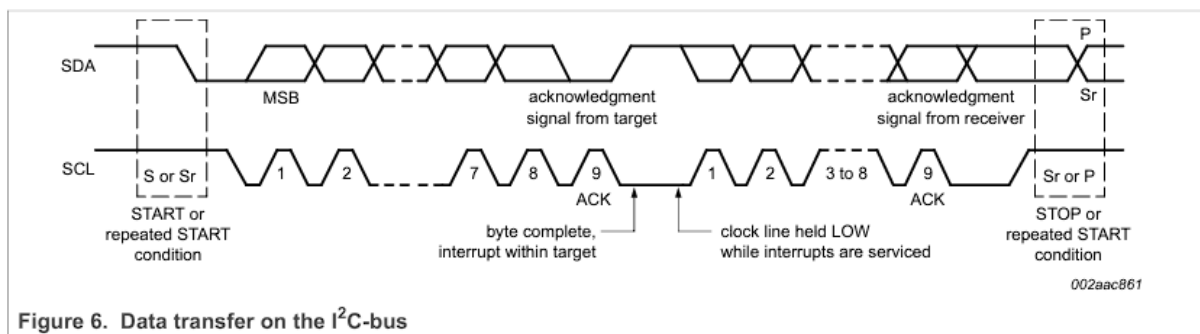
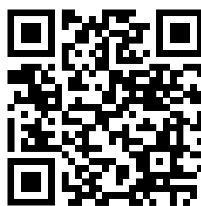
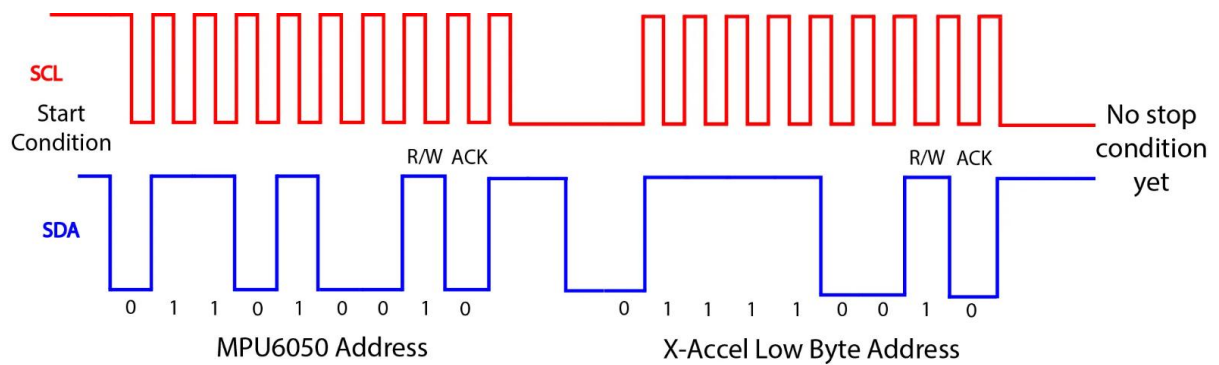


Figure 6. Data transfer on the I²C-bus

Source : [I²C-bus specification and user manual](#) date de consultation (2/05/25)



Lien QR Datasheet



Source : <https://i2.wp.com/www.teachmemicro.com/wp-content/uploads/2018/04/MPU6050-I2C-1.jpg?ssl=1>

Date de consultation (2/05/25)

Explication

Une trame de donnée commence toujours par une condition de départ envoyée par le maître pour signaler le début de la communication. Ensuite, l'adresse est envoyée bit par bit, suivie d'un bit de lecture ou écriture (ex : Lecture = lire des données d'un capteur ou de température. Écriture = écrire des données, configuration d'un capteur ou de température), suivie de l'accusé de réception ACK pour confirmer que l'adresse a bien été reçue, et pour finir la condition d'arrêt pour arrêter la transmission de données. Tout ce cycle correspond à 1 octet.

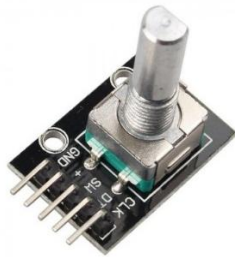
Condition de départ : commence quand la ligne SDA passe de haut à bas lorsque la ligne SCL est toujours en haut.

Condition d'arrêt : commence quand la ligne SDA passe de bas à haut lorsque la ligne SCL est toujours en haut.

Lecture et écriture : le bit de lecture est à 1, ce qui veut dire que le maître veut lire les données de l'esclave. Le bit d'écriture est à 0, ce qui veut dire que le maître veut écrire dans la donnée de l'esclave.

ACK : confirme la réception de donnée entre le maître et l'esclave, il garantit la fiabilité des données entre les deux.

5.5 L'encodeur rotatif



Source : https://arduinko.ru/image/cache/catalog/Enkoderpotenciometr_rotorny KY-040-1000x1340.jpg

Date de consultation (20/05/25)

Spécifications

Type : Codeur rotatif incrémental

Tension d'alimentation : 5v

Courant : 10 mA

Cycles par tour 30

Nombre de battements par tour : 20

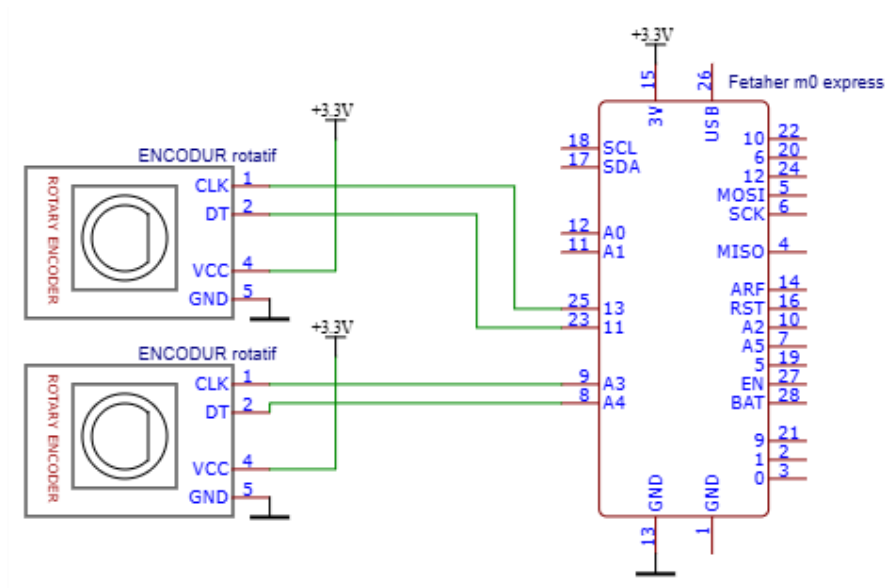
Dimensions : 20 x 30 x 30 mm

Poids : 10g



Lien QR spécification

Schéma de principe



Rôle des signaux

CLK : Signal d'impulsion d'horloge

DT : Signal d'impulsion de donnée

VCC : alimentation

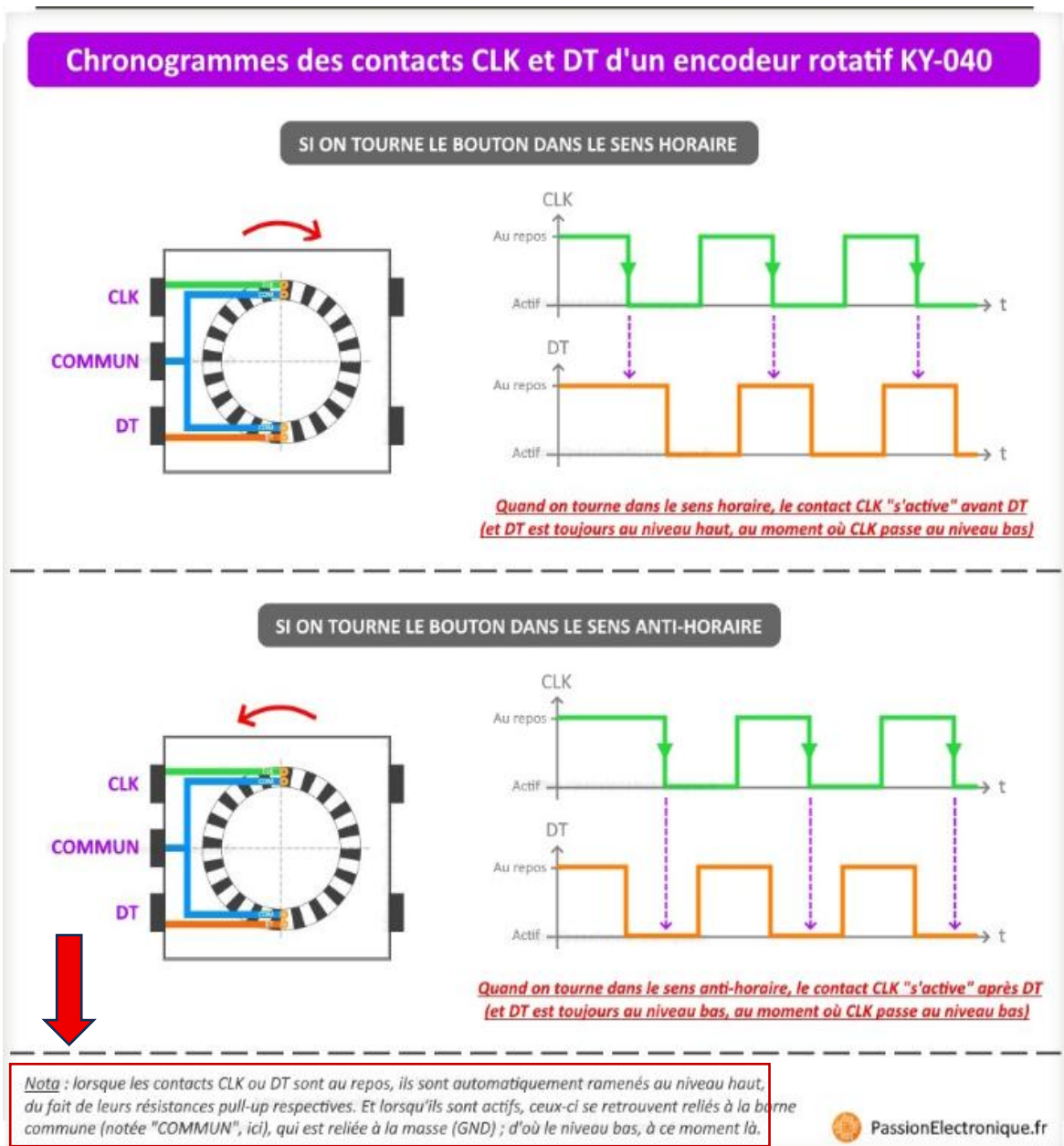
GND : masse

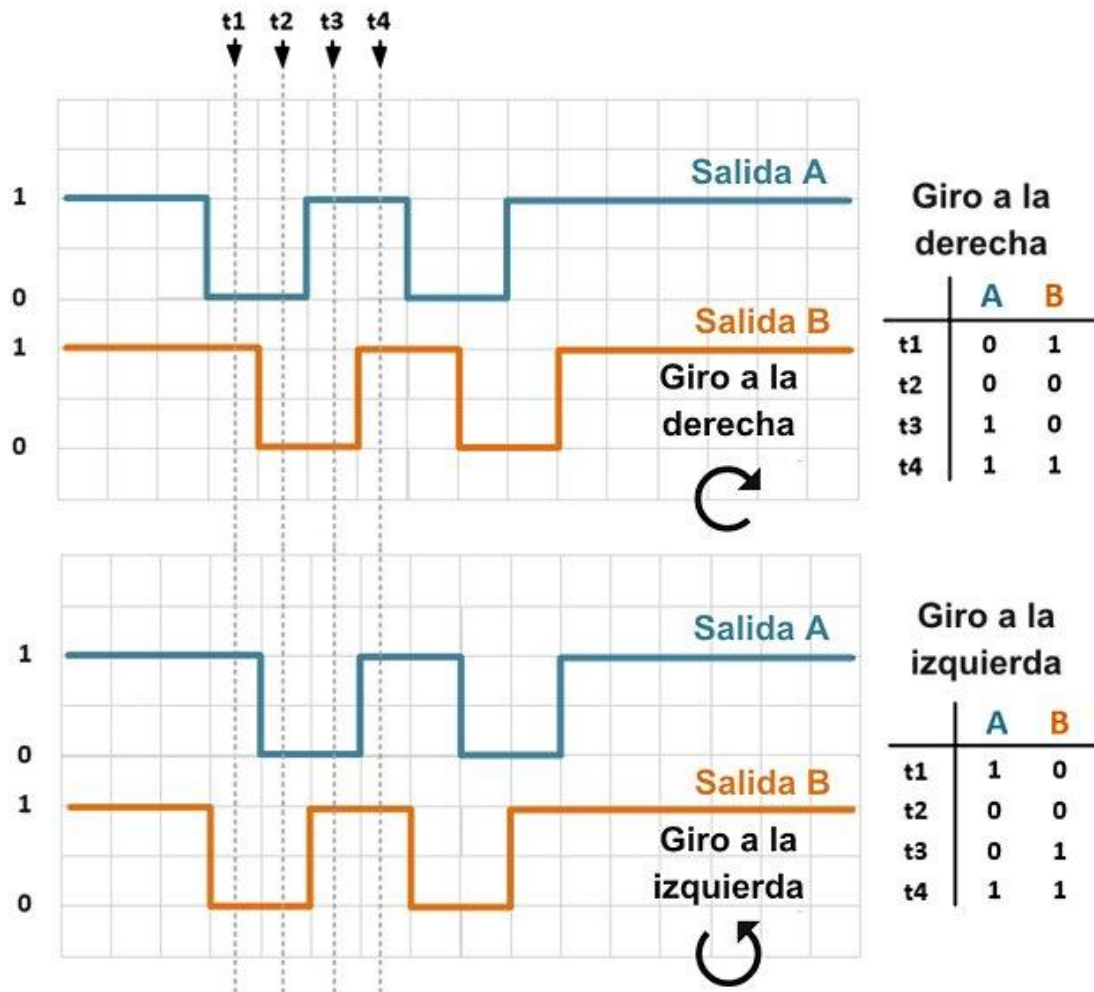
Principe de fonctionnement

L'encodeur KY-040 est un système d'encodage incrémental. Quand on tourne, les broches A (CLK) et B (DT) envoient des signaux carrés qui passent de l'état haut à l'état bas.

Ces signaux s'alternent au fur et à mesure qu'on fait tourner l'encodeur en fonction du sens dans lequel on tourne : l'un des signaux change avant l'autre pour que le microcontrôleur capte le sens de rotation. CLK et DT doivent être en quadrature, ce qui veut dire déphasés de 90°.

Schéma





Source: <https://th.bing.com/th/id/R.01cf5959a329601a70316d1d934d9806?rik=hAzRjl8eRxrZyA&riu=http%3a%2f%2frobots-argentina.com.ar%2fdidactica%2fwp-content%2fuploads%2fOperaci%C3%B3n-seg%C3%BAn-el-giro.jpg&ehk=z%2fuoeEBkK30OS5BbUOfpMi3SRisRjhwBHqyexxLnr4g%3d&risl=&pid=ImgRaw&r=0>

Date de consultation (20/05/25)

Analyse

Quand on commence à tourner l'axe vers la droite, on observe que le signal A change d'état un peu avant le signal B. Par exemple, A passe à 0, puis B passe à 0 juste après, ensuite A repasse à 1 et B le suit un peu plus tard. Ce petit décalage entre les deux signaux permet de savoir que l'encodeur tourne dans un sens précis.

Si on tourne l'encodeur dans l'autre sens, c'est le signal B qui change d'état en premier, avant A.

Programme Test

```
#define CLK 2 // Broche d'horloge
#define DT 3 // Broche de données

int counter = 0; // Compteur pour suivre la rotation
int currentStateCLK;
int lastStateCLK;

void setup() {
    pinMode(CLK, INPUT);
    pinMode(DT, INPUT);

    Serial.begin(9600);

    // Lire l'état initial de la broche CLK
    lastStateCLK = digitalRead(CLK);

    Serial.println("Test de l'encodeur rotatif KY-040");
}

void loop() {
    // Lire l'état actuel de la broche CLK
    currentStateCLK = digitalRead(CLK);

    // Si l'état a changé, l'encodeur a tourné
    if (currentStateCLK != lastStateCLK) {
        // Vérifier la broche DT pour déterminer le sens de rotation
        if (digitalRead(DT) != currentStateCLK) {
            counter++;
        } else {
            counter--;
        }
    }

    Serial.print("Compteur : ");
    Serial.println(counter);
}

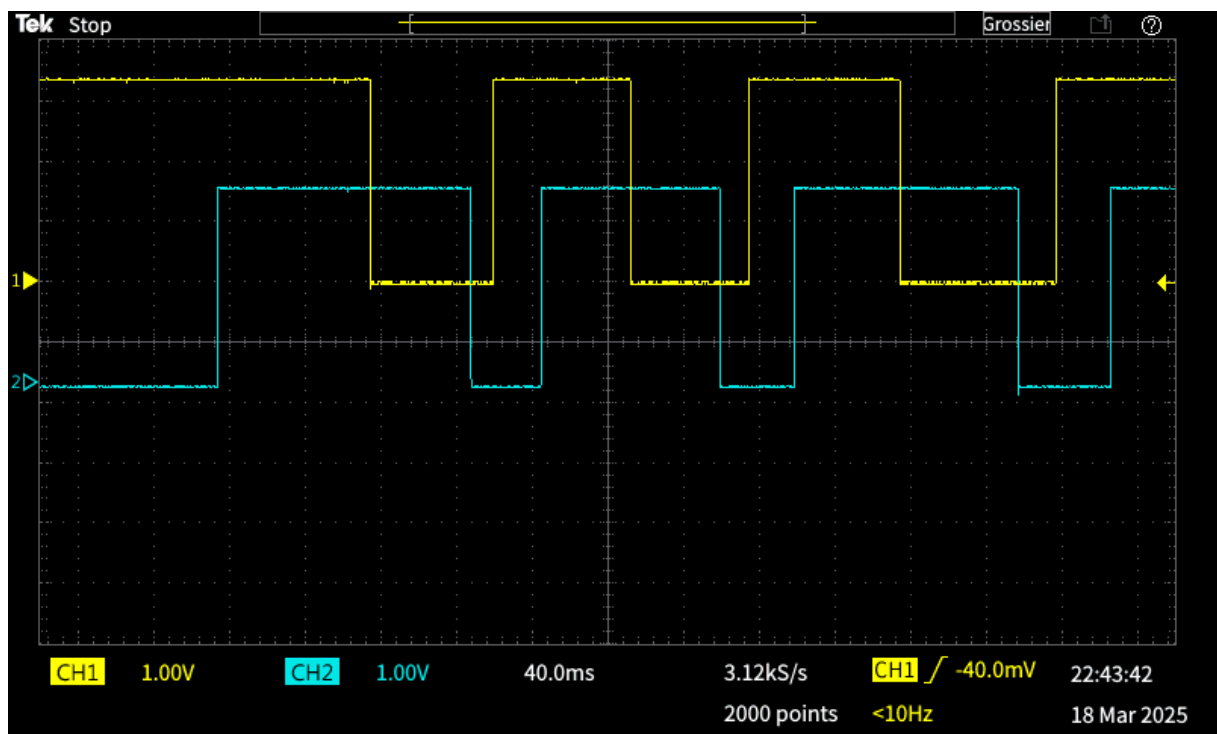
// Sauvegarder le dernier état de CLK
lastStateCLK = currentStateCLK;
}}
```

Source : [Comment utiliser l'encodeur rotatif KY-040 avec Arduino – 38-3D](#) Date de consultation (20/5/2025)



Lien QR code

5.5.1 Mesure de l'encodeur rotatif



CLK : Jaune

DT : Bleu

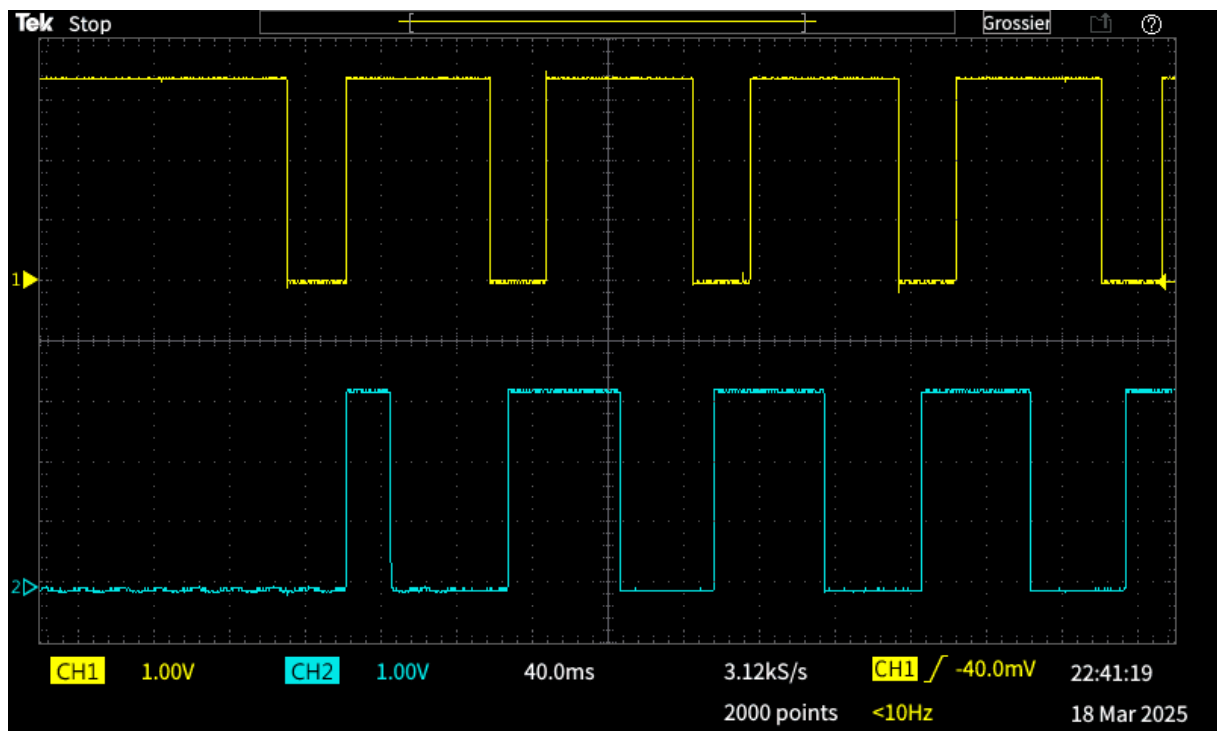
T=40ms

Analyse

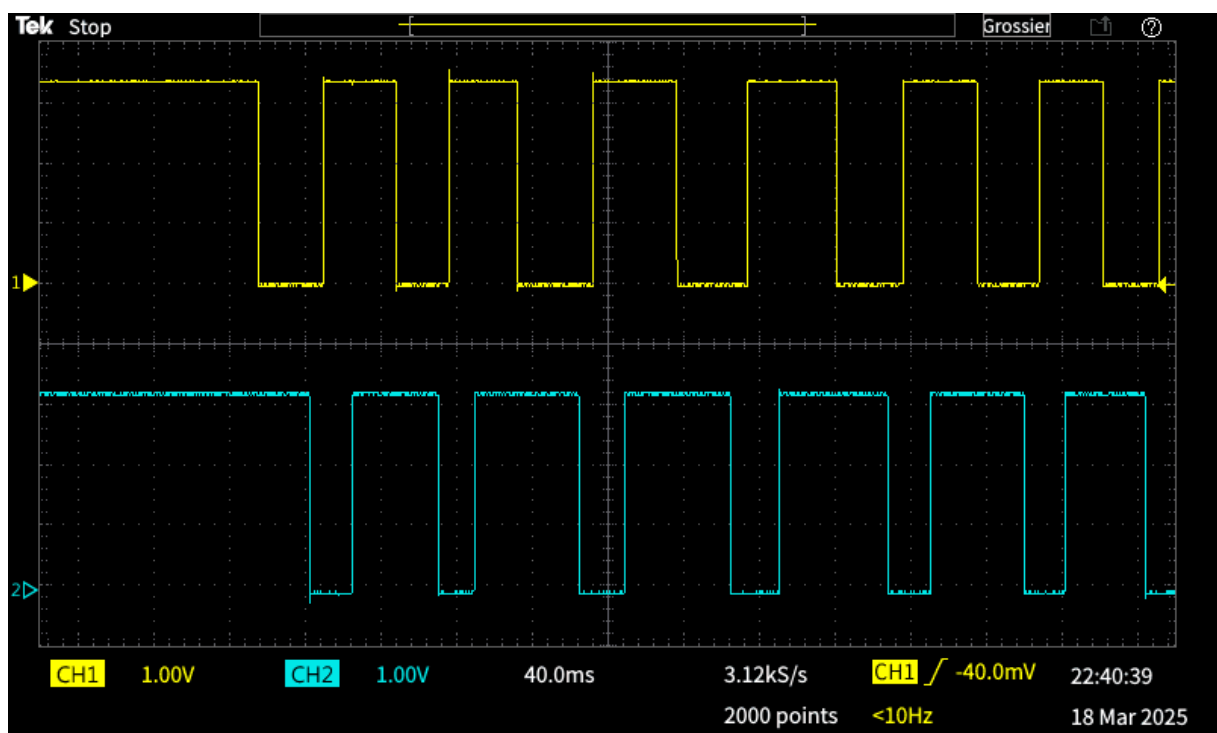
On remarque que quand on tourne l'encodeur, peu importe le sens, il sera déphasé. Malheureusement, le SW de l'encodeur rotatif qui fait office de BP n'est pas mesuré car je ne l'utilise pas dans mon projet. Selon la rotation effectuée, le déphasage ne s'effectue pas de la même manière, soit de 90° ou de -90° .

Si on veut calculer l'impulsion, il nous suffit de faire un petit calcul : $1/T = 1/0,040 \text{ s} = 25 \text{ Hz}$. Cela signifie que l'encodeur génère 25 impulsions par seconde, ce qui est cohérent avec les quelques sources trouvées sur l'encodeur rotatif, qui indiquent 20 impulsions par seconde.

Les rebonds ne sont pas présents sur les images, mais il peut y en avoir, et ils se distinguent par des variations rapides ou des signaux courts qui provoquent une fausse lecture de direction ou une multiplication des impulsions.



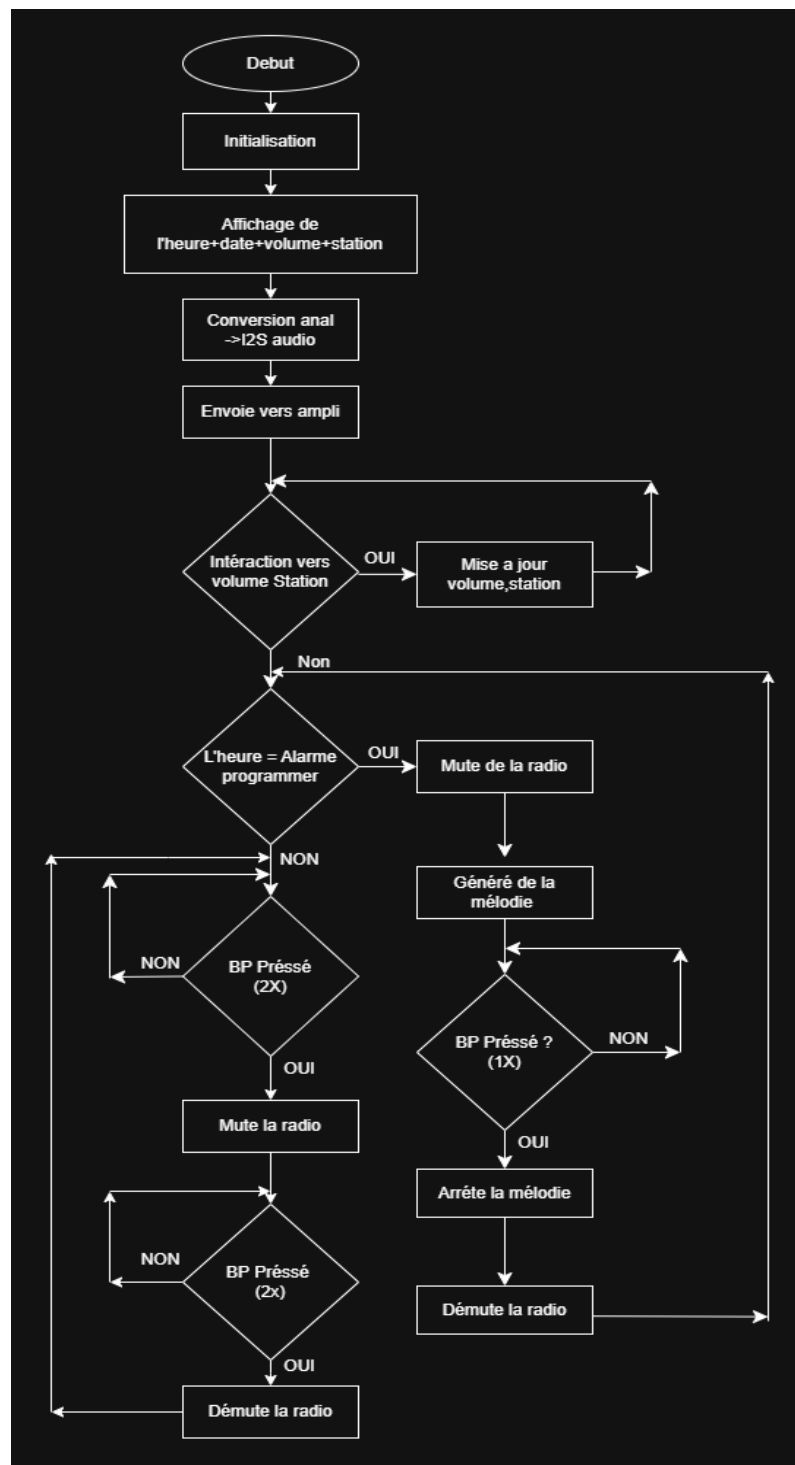
Celui tourne vers le sens anti-horaire, déphasage de -90°



Celui la tourne vers le sens horaire, déphasage de 90°

6 La Programmation

6.1 L'ordinogramme général



6.2 Le programme

```
#include "Peripherique.h"

RDA5807M radio;
RTC_PCF8523 rtc;
//*****Variable Globale*****

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);

int Enregistre_CLK;
int Enregistre2_CLK;

int FREQUENCE_RADIO = 9000;
int VOLUME_RADIO = 12;
// ===== Lecture des signaux rotatifs
=====
void Lecture_CLK(void) {
    Enregistre_CLK = digitalRead(CLK);
    Enregistre2_CLK = digitalRead(CLK1);
}
// ===== Configuration des broches =====
void Pin_Mode(void) {
    pinMode(CLK, INPUT_PULLUP);
    pinMode(DT, INPUT_PULLUP);
    pinMode(CLK1, INPUT_PULLUP);
    pinMode(DT1, INPUT_PULLUP);
}
// ===== Initialisation de la radio =====
void Initi_Radio(void) { //librairie
    radio.initWire(Wire);
    radio.setup(RADIO_FMSPACING, RADIO_FMSPACING_100);
    radio.setup(RADIO_DEEMPHASIS, RADIO_DEEMPHASIS_50);

    radio.setBandFrequency(BANDE_RADIO, FREQUENCE_RADIO); /// parametre pas
fixe
    radio.setVolume(VOLUME_RADIO); // parametre pas
fixe
    radio.setMute(false);
    radio.setMono(true);
}
// ===== Réglage de la fréquence radio via encodeur
=====
void Station(void) { //mon idée, correction de la syntaxe(;;)avec chat
gpt
    int Enregistre1_CLK = digitalRead(CLK);
```

```

if (Enregistre1_CLK != Enregistre_CLK) {
    if (digitalRead(DT) != Enregistre1_CLK) {
        FREQUENCE_RADIO += 10;
        if (FREQUENCE_RADIO > 10800)
            FREQUENCE_RADIO = 8700;

    } else {
        FREQUENCE_RADIO -= 10;
        if (FREQUENCE_RADIO < 8700)
            FREQUENCE_RADIO = 10800;
    }

    radio.setBandFrequency(BANDE_RADIO, FREQUENCE_RADIO);
    Serial.print("Fréquence: "); //TEST
    Serial.println(FREQUENCE_RADIO);
}

Enregistre_CLK = Enregistre1_CLK;
}

// ===== Réglage du volume via encodeur
=====
void Volume(void) { /// MOI SEUL
    int Enregistre3_CLK = digitalRead(CLK1);
    if (Enregistre3_CLK != Enregistre2_CLK) {
        if (digitalRead(DT1) != Enregistre2_CLK) {
            VOLUME_RADIO += 1;
            if (VOLUME_RADIO > 15)
                VOLUME_RADIO = 15;
        } else {
            VOLUME_RADIO -= 1;
            if (VOLUME_RADIO < 0)
                VOLUME_RADIO = 0;
        }
        radio.setVolume(VOLUME_RADIO);
        Serial.print("Volume: ");
        Serial.println(VOLUME_RADIO);
    }
    Enregistre2_CLK = Enregistre3_CLK;
}

// ===== Lecture audio =====
void Audio(void) {/// repris de chat gpt
    int ADC_Gauche = analogRead(GAUCHE);
    int ADC_Droite = analogRead(DROIT);

    int16_t audioGauche = (ADC_Gauche - 275) * 64; // echantillonner sur 16
    bits recentrée sur 0 puis amplifier 275 son de qualité
    int16_t audioDroite = (ADC_Droite - 275) * 64;
    I2S.write(audioGauche);
    I2S.write(audioDroite);
}

```

```

}
// ===== Initialisation de la sortie audio I2S
=====
void Initi_Ampli(void) {///librairie
  I2S.begin(I2S_PHILIPS_MODE, 44100, 16); // Initialisation de l'I2S mode
Philips, 44.1 kHz, 16 bits44100
}
void Initi_Ecran(void){
  tft.begin();
  tft.setRotation(1);
  tft.fillScreen(ILI9341_BLACK);
}
}

```

Code radio au stade actuelle 23/25/25



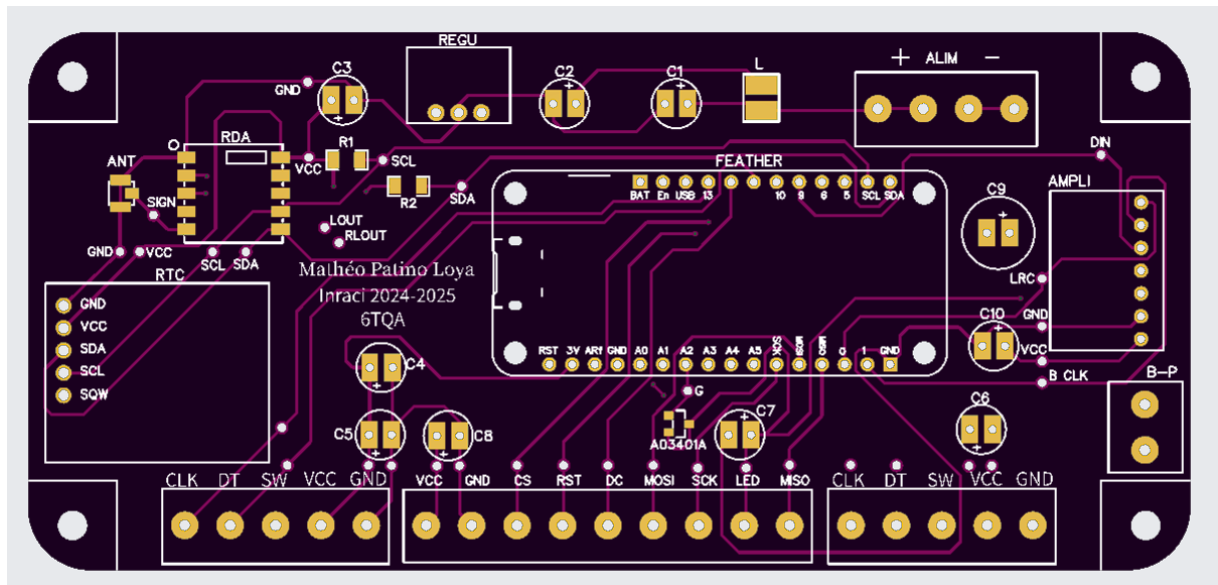
Lien GitHub de mon code

[Code TFE MPL/Code TFE MPL.ino at main · MatheoPatinoLoya/Code TFE MPL](#)

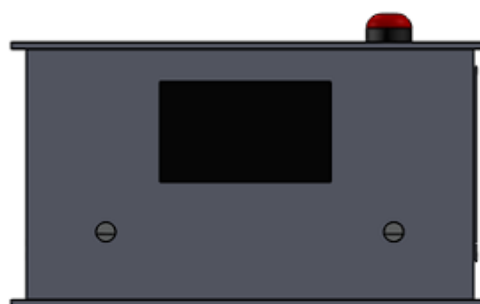
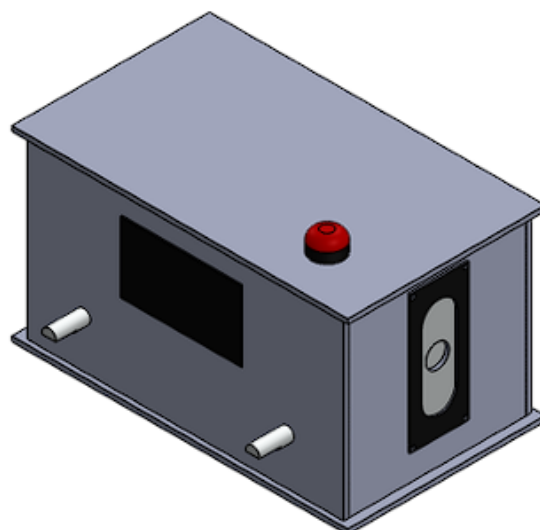
7 La fabrication

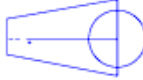
Pour pouvoir réaliser la fabrication, j'ai dû commencer par le PCB pour l'adapter selon la forme et la taille de mon boîtier.

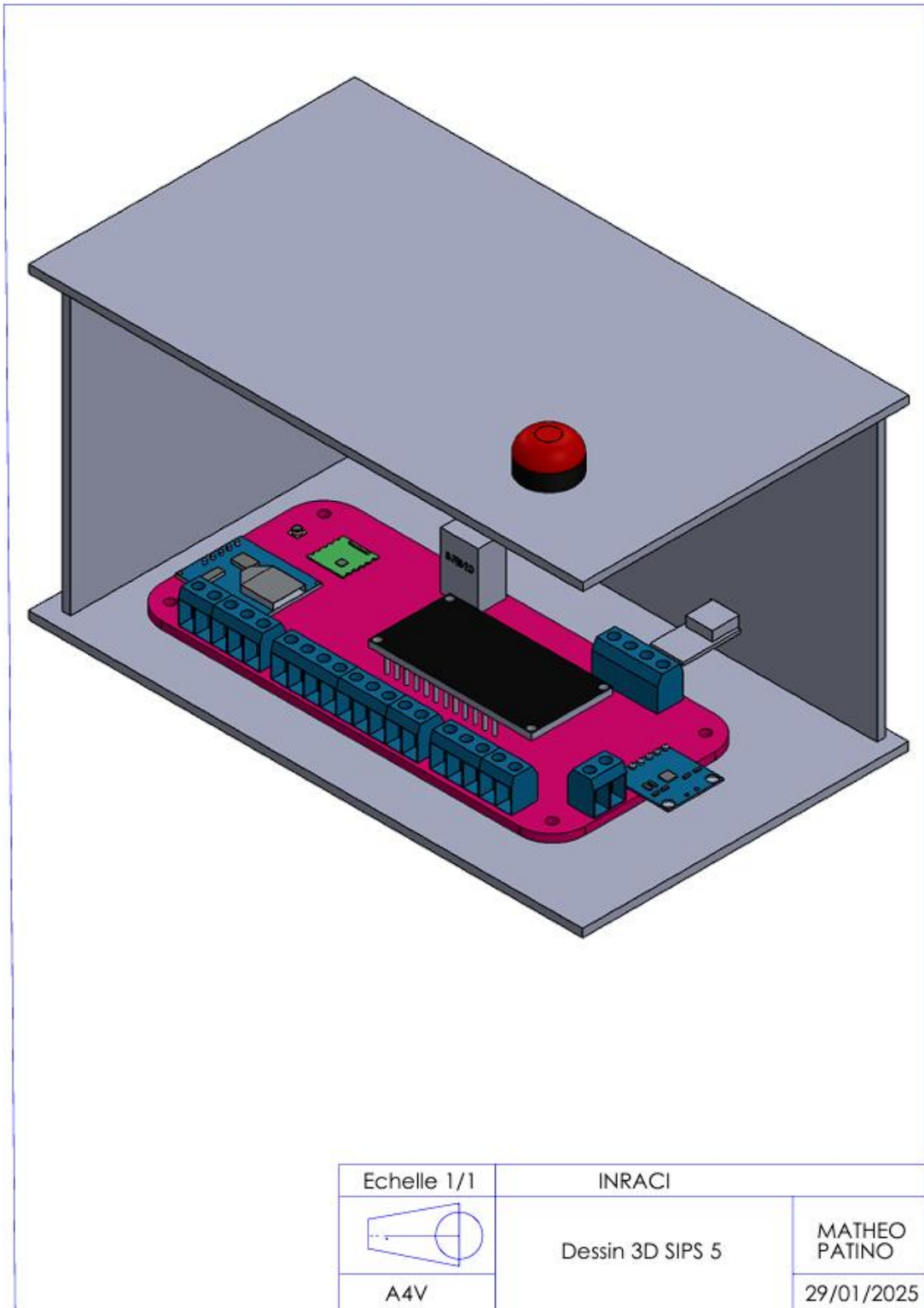
Voici mon PCB sur lequel j'ai fait des tests, notamment des mesures, pour avoir une petite idée de ce à quoi devrait ressembler mon PCB final.

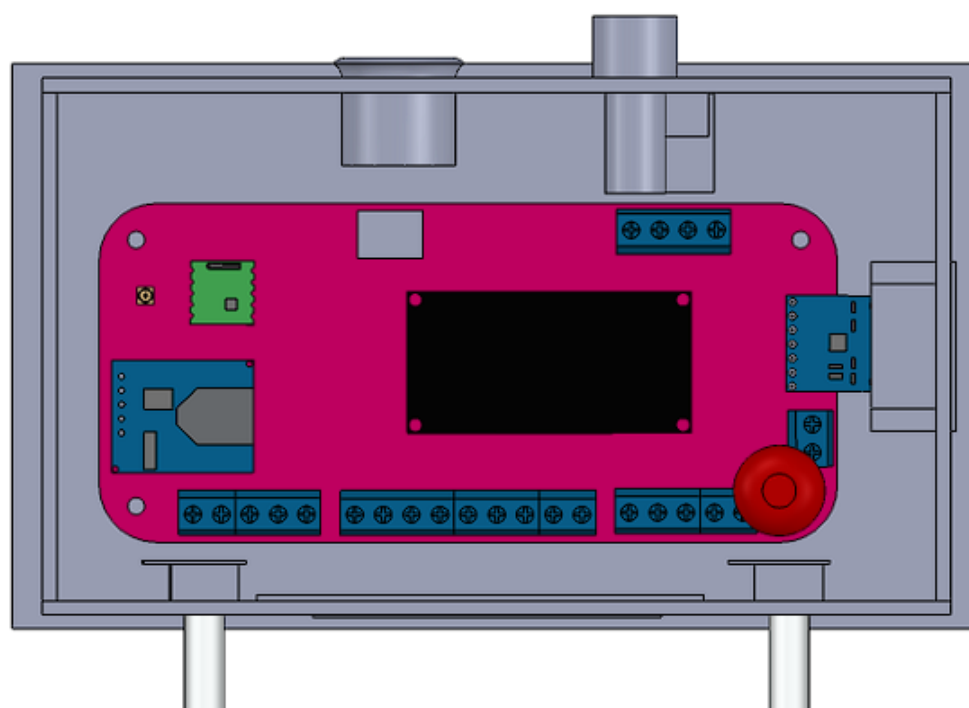



Pour avoir une vue d'ensemble et un peu imaginer mon boîtier idéal j'ai utilisé Solidwork2013 dans lequel j'ai assemblé mon boîtier et mon Pcb.

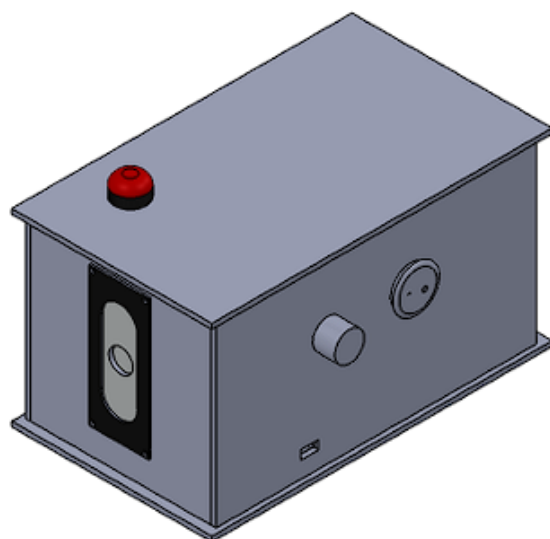


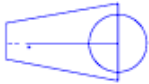
Echelle 1/1	INRACI	
	Dessin 3D SIPS 5	MATHEO PATINO
A4V		29/01/2025



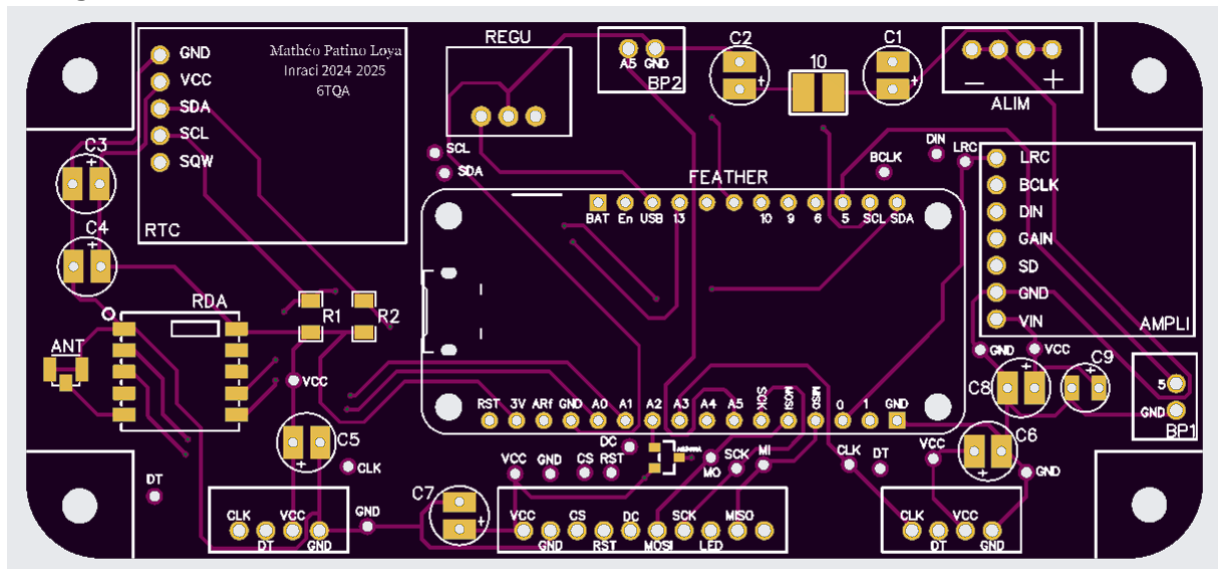


Echelle 1/1	INRACI	
	Dessin 3D SIPS 5	MATHEO PATINO
A4V		29/01/2025

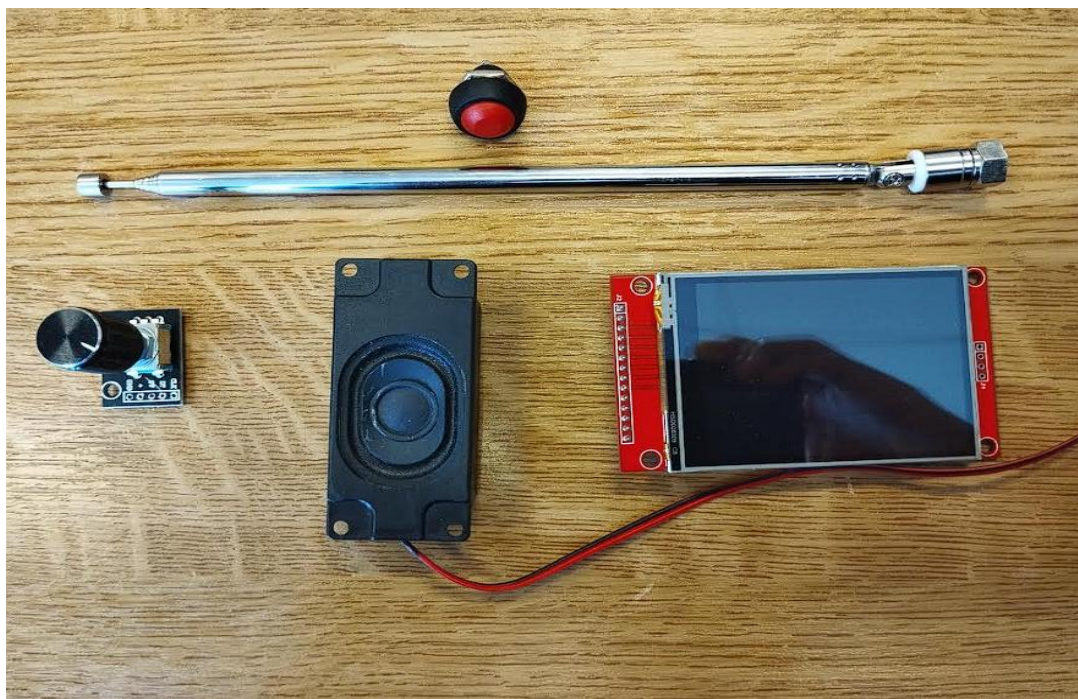


Echelle 1/1	INRACI	
	Dessin 3D SIPS 5	MATHEO PATINO
A4V		29/01/2025

À la suite de l'état d'avancement dans lequel j'étais, M. Kapita a ouvertement critiqué mes borniers. Suite à cette critique constructive, j'ai décidé de refaire un deuxième PCB pour notamment changer les borniers classiques par des connecteurs JST et corriger d'autres défauts.

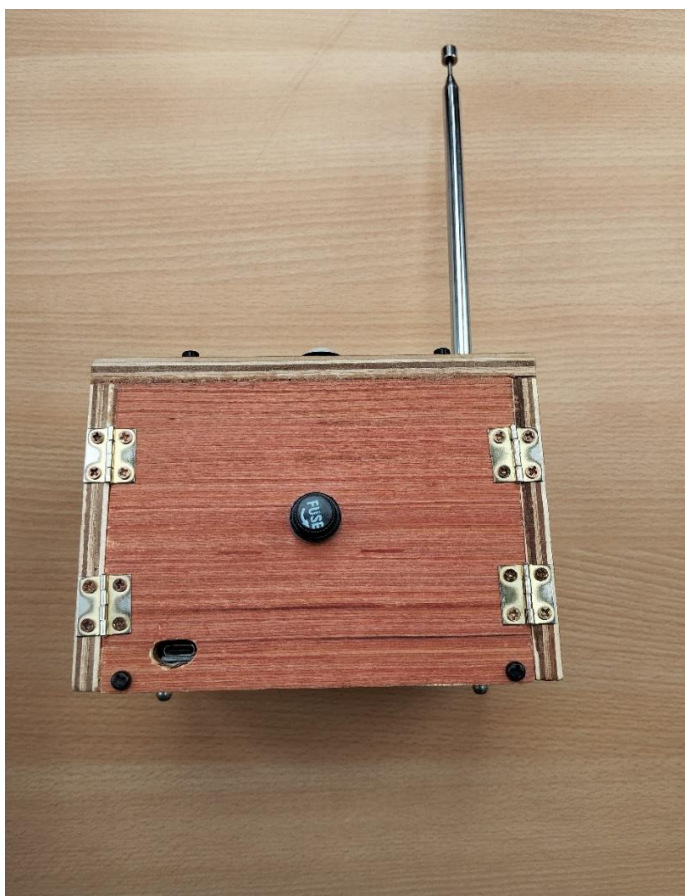


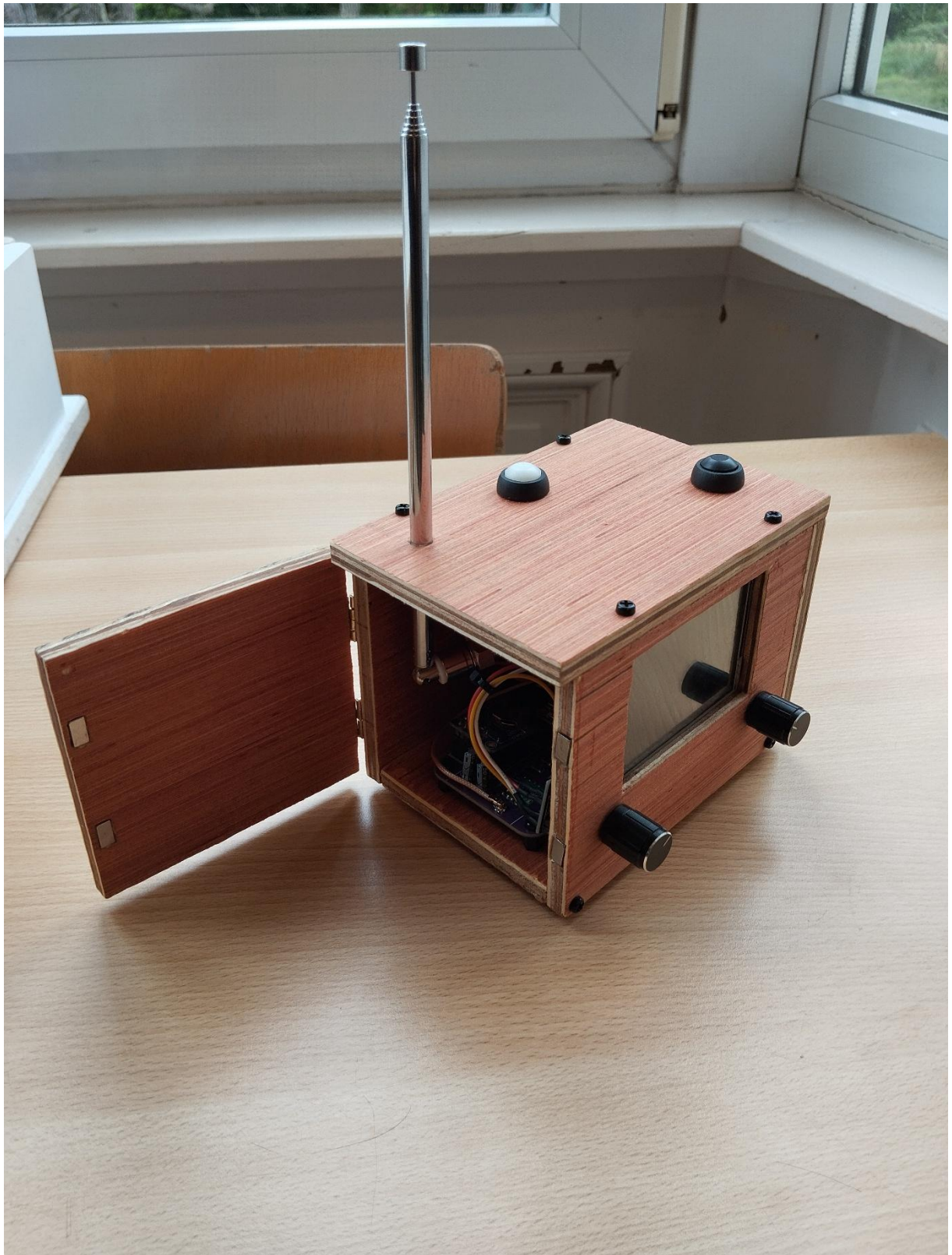
Sur ce PCB j'ai décidé de réduire la taille pour que ce soit le plus compact possible.



Une des étapes les plus dures, physiquement et mentalement, a été de faire les trous nécessaires pour pouvoir assembler ceci dans le bois.

Finalement le résultat attendu a été accomplie.





Pour finir l'état final est satisfaisant, je devrai encore l'améliorer car l'esthétique est vraiment son point faible, mais avant tout mon but a été de le faire en bois pour avoir une radio vintage, donc aucun regret.

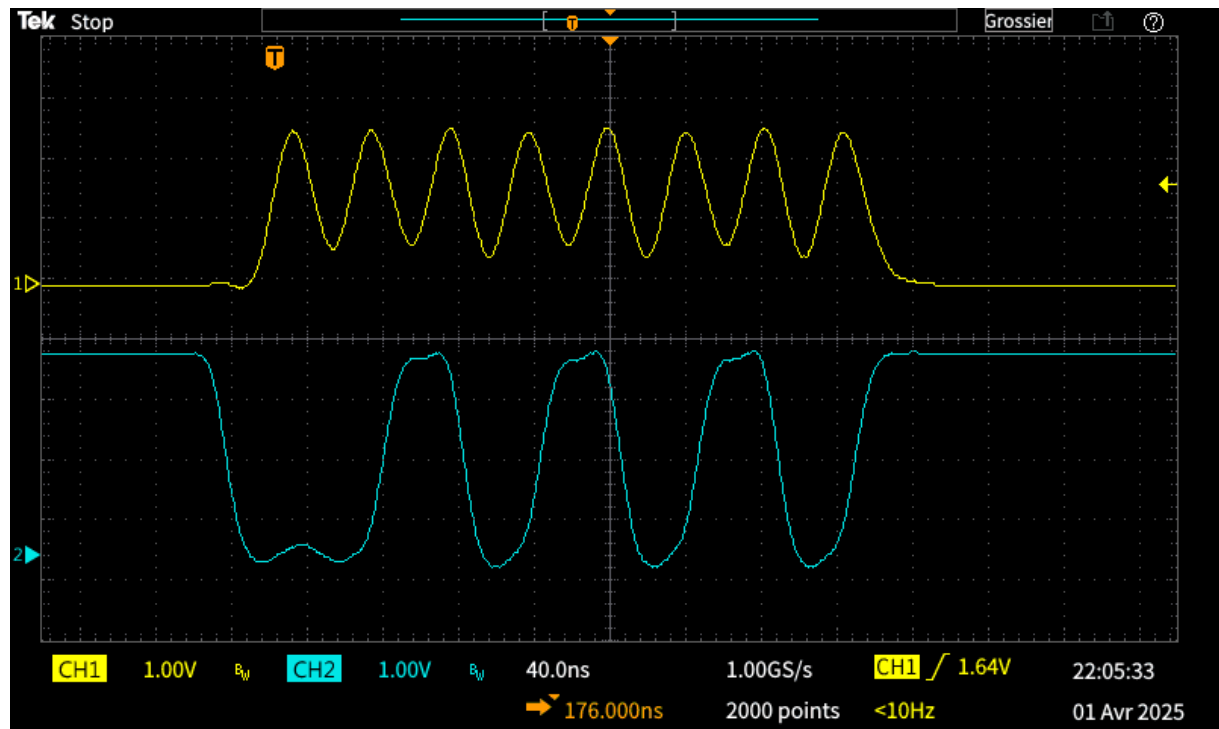
8 La mise en point

Durant la réalisation de mon projet, j'ai rencontré de nombreux soucis en cours de route, aussi bien au niveau du hardware que software.

Les mesures de l'écran spi



Mesure



Jaune : SCK

Bleu : Mosi

Analyse

Sur cette image, on constate que les signaux ressemblent plus à des sinus qu'à un signal carré, par conséquent impossible d'avoir une trame de donnée lisible et correcte. La cause de ce problème est la limitation de l'oscilloscope et de la sonde.

En sachant que la période est de 40 ns, la fréquence est de 25 MHz.

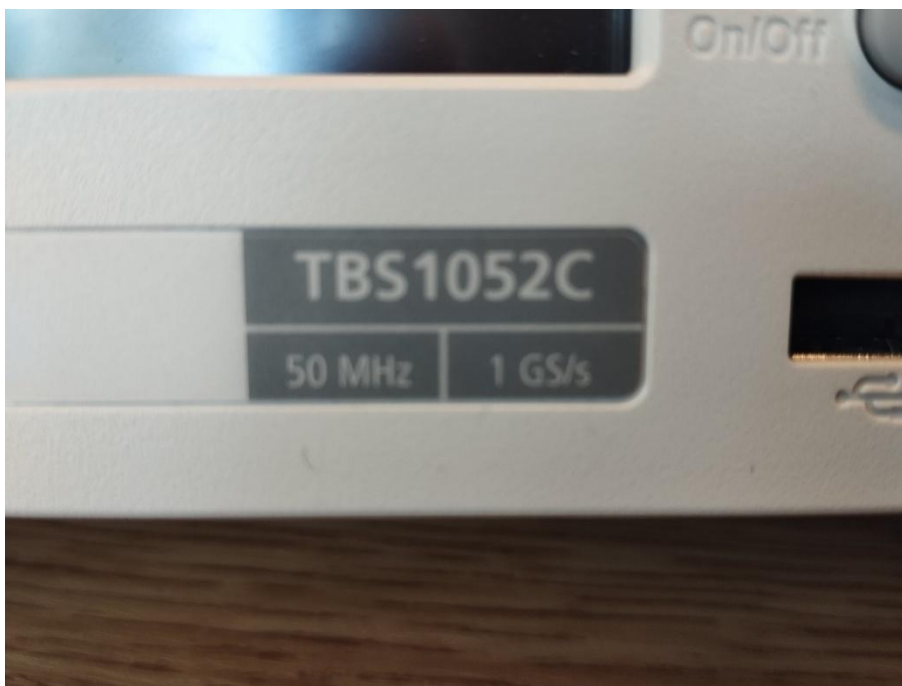
Cette distorsion est principalement due à la bande passante limitée de la sonde (50 MHz), qui ne permet pas de capturer correctement les harmoniques présentes dans un signal carré. Un signal carré contient une fondamentale ainsi que des multiples harmoniques impairs. Donc, si la fondamentale est de 25 MHz et que la sonde a une bande passante limitée à 50 MHz, l'harmonique qui suit est de 75 MHz, ce qui dépasse la bande passante de la sonde. Donc la sonde filtre les harmoniques et ne laisse passer que la fondamentale 25 MHz, ce qui correspond à une forme sinusoïdale.

De plus, la limitation de l'oscilloscope pose un problème, car si la bande passante de celui-ci est de 50 MHz, alors il ne peut afficher les fréquences supérieures à 50 MHz. Or un signal carré contient des harmoniques impairs, ce qui donne une mauvaise image du signal et le lisse.

La solution pour régler ce problème est d'avoir une sonde avec une bande passante supérieure à la fréquence du signal, et aussi un oscilloscope avec la bande passante la plus élevée possible.



On constate qu'une des sondes ici est de 100 MHz et l'autre est de 50 MHz. Le signal jaune a été mesuré par la sonde à 50 MHz, et le signal bleu par celle de 100 MHz. Si on revient sur l'image, on remarque que le signal bleu ressemble moins à un sinus comparé à l'autre. Ceci est dû à la sonde qui a une bande passante plus élevée, par conséquent, elle laisse passer plus d'harmoniques du signal carré.

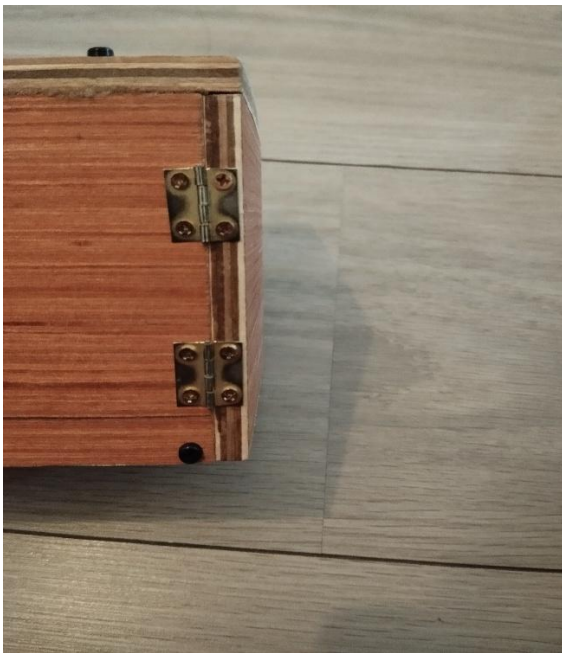
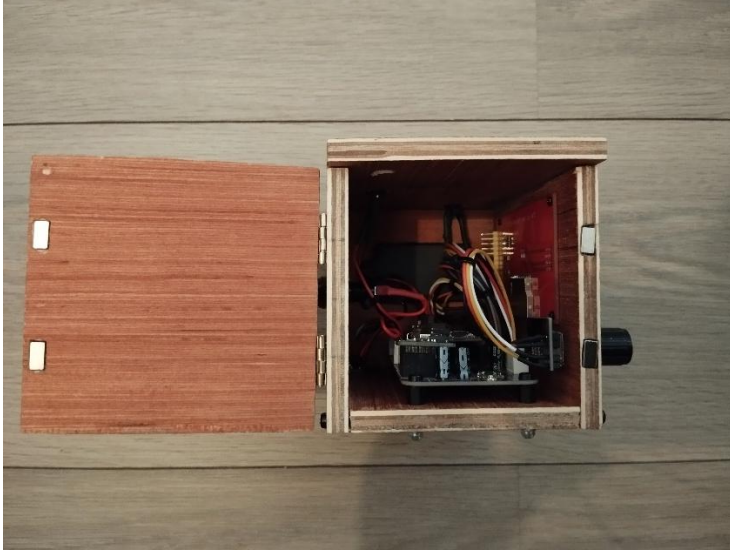


50Mhz correspond à la bande passant de l'oscilloscope (max)

1GS/S correspond à la fréquence d'échantillonnage (max)

Conception du boîtier

J'ai eu un petit problème lors de la conception du boîtier en bois, car à la base mon boîtier devait rester fermé sans aucun moyen de l'ouvrir, ce qui fait que on ne pourrait jamais voir mon beau PCB et mon câblage.



Une des solutions a été de créer une petite porte pour pouvoir accéder de l'intérieure de mon boîtier à l'aide de charnière et d'aimant.

Conflit de librairie

Dans le cadre de mon projet, la partie software a aussi été un gros obstacle. En effet, j'ai eu un problème de librairie de deux bibliothèques qui utilisaient la même fonction appelée `zero_dmac_handler`, ce qui empêchait mon code de se compiler.

De base la fonction `zero_dmac_handler()` sert à gérer les interruptions générées par le contrôleur sur les microcontrôleurs SAMD21.

La solution a été de changer le nom de la fonction dans un des fichiers `.cpp`, pour qu'il n'y ait plus de soucis. Une fois cette modification faite, le code s'est compilé sans problème.

Bien sûr, M. Kapita m'a aidé à régler ce souci, parce que c'était trop compliqué pour un simple débutant comme moi. Sans son aide, j'aurais eu beaucoup de mal à comprendre comment changer l'erreur, malheureusement je n'ai pas de pris de photo de l'erreur de compilation.

9 La conclusion

Dans un premier temps la partie mécanique, assemblage a été finaliser avec succès même s'il reste des petites améliorations esthétiques de mon boîtier Malheureusement, la partie programmation a été une épreuve difficile. Au début de mon projet, j'étais un peu perdu, notamment pour la mise en commun des différents modules, Ducoup j'ai accumuler du retard et la programmation actuelle du réveil n'a pas encore débiter.

Le plus important c'est que j'ai avancé sans jamais abandonner même si au début je ne savais pas comment m'y prendre car c'était un domaine encore inconnu puisque a l'inranci il n'y plus de cour de Télécommunication. Mais pour y arriver j'ai dû me renseigner au sujet de tout, datasheet des composant, librairie, etc. Grace a messieurs Mr kapita je n'ai pas perdu mon chemin et j'ai pu continuer mon projet avec plus de confiance et de clarté.

Avant tous ces difficultés ont été très enrichissante j'ai compris l'importance de mieux s'organiser, de se documenter et de ne pas trop dépendre de chat GPT.

Finalement si tout ne s'est pas passé comme je l'espérais, je suis satisfait du chemin parcouru, car cette expérience m'a permis d'apprendre de mes erreurs et m'a donné de bonnes bases pour mes futurs projets.

Bibliographie

Librairie

Source : [adafruit/Adafruit_ZeroI2S: I2S audio playback library for the Arduino Zero / Adafruit Feather M0 \(SAMD21 processor\).](#)

Date de consultation (2/10/2024)

Source : [GitHub - mathertel/Radio : Une librairie Arduino pour contrôler les puces radio FM comme SI4703, SI4705, RDA5807M, TEA5767.](#)

Date de consultation (5/10/2024)

Source : [pu2clr/RDA5807: Arduino Library for SINGLE-CHIP BROADCAST FM RADIO TUNER RDA5807](#)

Date de consultation (8/10/2024)

Source : [adafruit/Adafruit_ILI9341: Library for Adafruit ILI9341 displays](#)

Date de consultation (15/10/2024)

Source : [adafruit/Adafruit-GFX-Library: Adafruit GFX graphics core Arduino library, this is the 'core' class that all our other graphics libraries derive from](#)

Date de consultation (18/10/2024)

Source : [GitHub - adafruit/RTCLib: A fork of Jeelab's fantastic RTC Arduino library](#)

Date de consultation (24/10/2024)

Datasheet

Source : [adafruit-max98357-i2s-class-d-mono-amp.pdf](#)

Date de consultation (15/11/2024)

Source : [MAX98357A/MAX98357B - Tiny, Low-Cost, PCM Class D Amplifier with Class AB Performance](#)

Date de consultation(12/11/2024)

Source : [KY-040 Datasheet\(PDF\) - JOY-IT](#)

Date de consultation(2/2/2025)

Source : [RDA5807M Datasheet\(PDF\) - RDA Microelectronics, Inc](#)

Date de consultation(13/12/2024)

Source : [R-78E-0.5.pdf](#)

Date de consultation(2/3/2025)

Source : [UMW AO3401A](#)

Date de consultation(26/3/2025)

Sources d'information

Source : [Wikipedia](#)

Date de consultation (1/10/2024 - 26/05/2025)

Source : [ChatGPT](#)

Date de consultation (1/10/2024 – 26/05/2025)

Source : [Simple FM Radio Receiver With Arduino UNO and RDA5807M | Arduino | Maker Pro](#)

Date de consultation (7/10/2024)

Source : [Radio fm basée sur Arduino utilisant rda5807 - l'audio 2025](#)

Date de consultation (8/4/2025)

Source : [Arduino RDA5807M FM Receiver](#)

Date de consultation (20/11/2024)

Source : [Radio FM audio numérique basée sur Arduino utilisant RDA5807 : 15 étapes - Instructables](#)

Date de consultation (2/5/2025)

Source : [Module SPI 2,8 pouces ILI9341 SKU :MSP2807 - Wiki LCD](#)

Date de consultation (9/1/2025)

Source : [Comment utiliser l'écran ILI9341 avec Arduino - Tutoriel complet et détaillé](#) date de consultation(26/1/2025)

Source : [Encodeur rotatif : fonctionnement + exemples de code arduino](#)

Date de consultation (27/1/2025)

Source : [Encodeur rotatif KY-040 | Modules et senseurs avec Arduino](#)

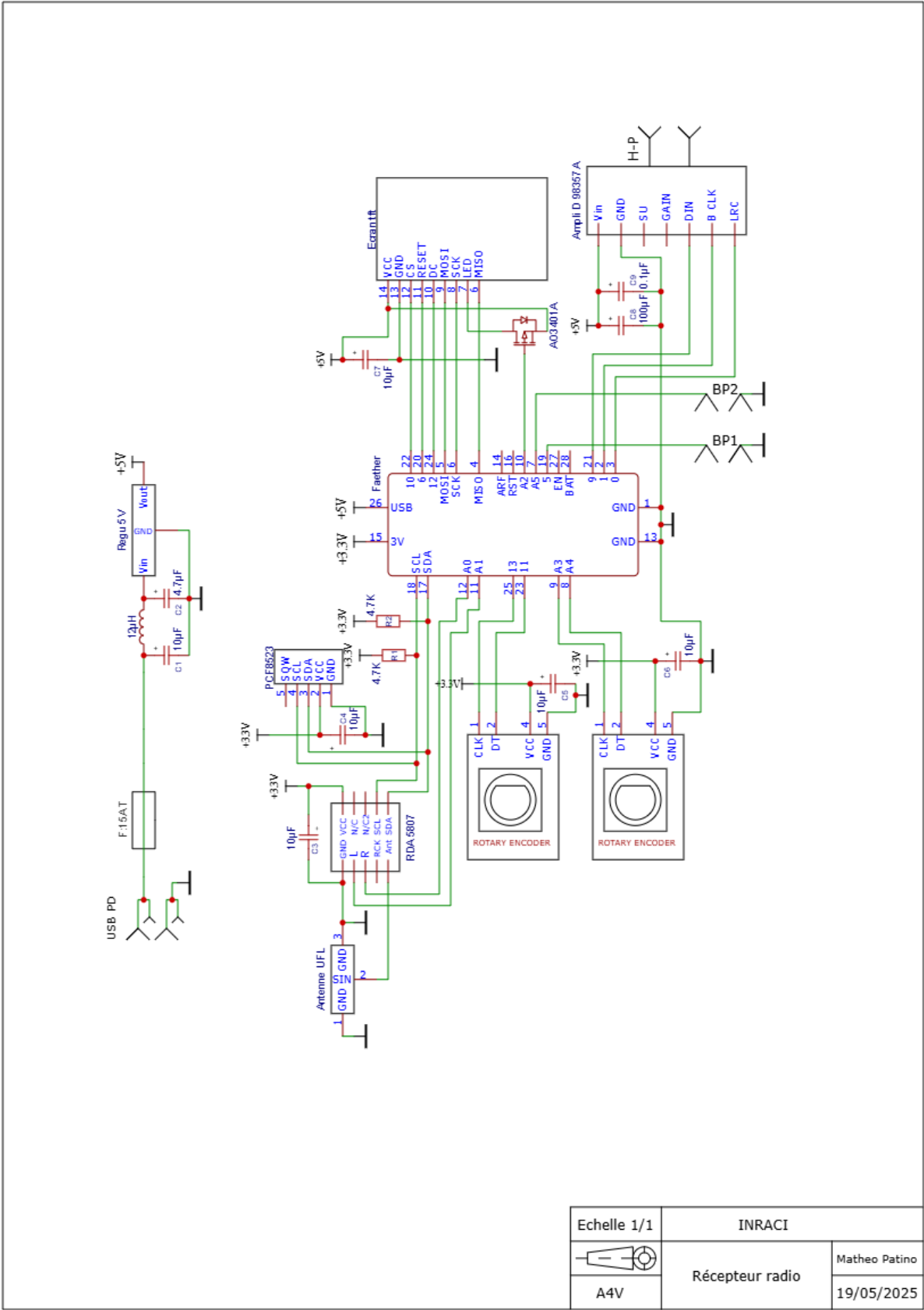
Date de consultation (28/1/2025)

Source : [I2S Library Examples | Arduino Documentation](#)

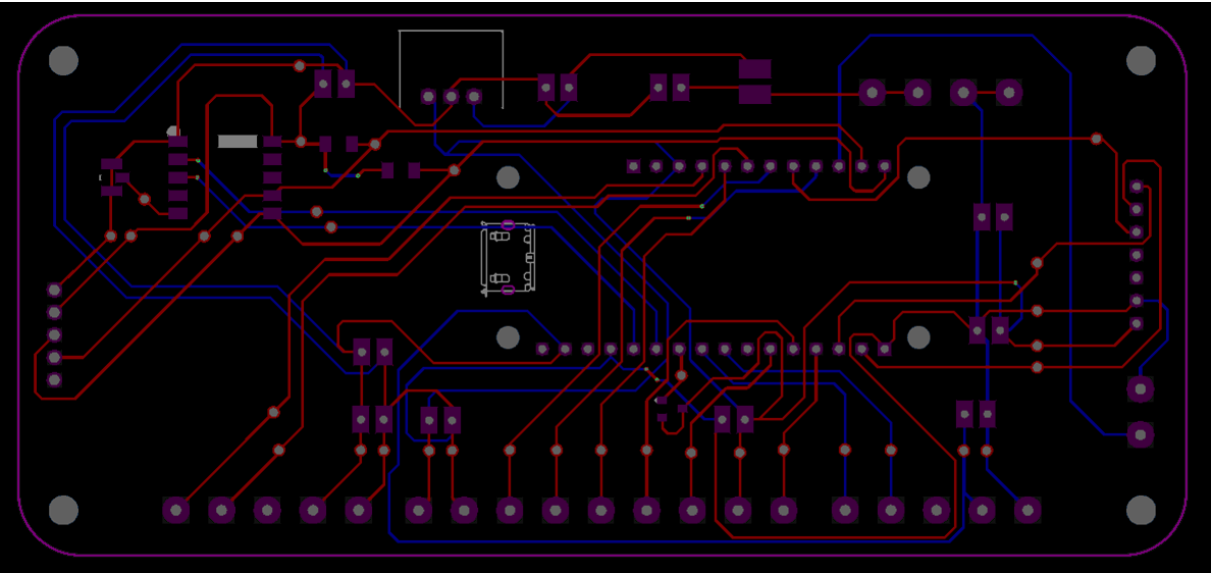
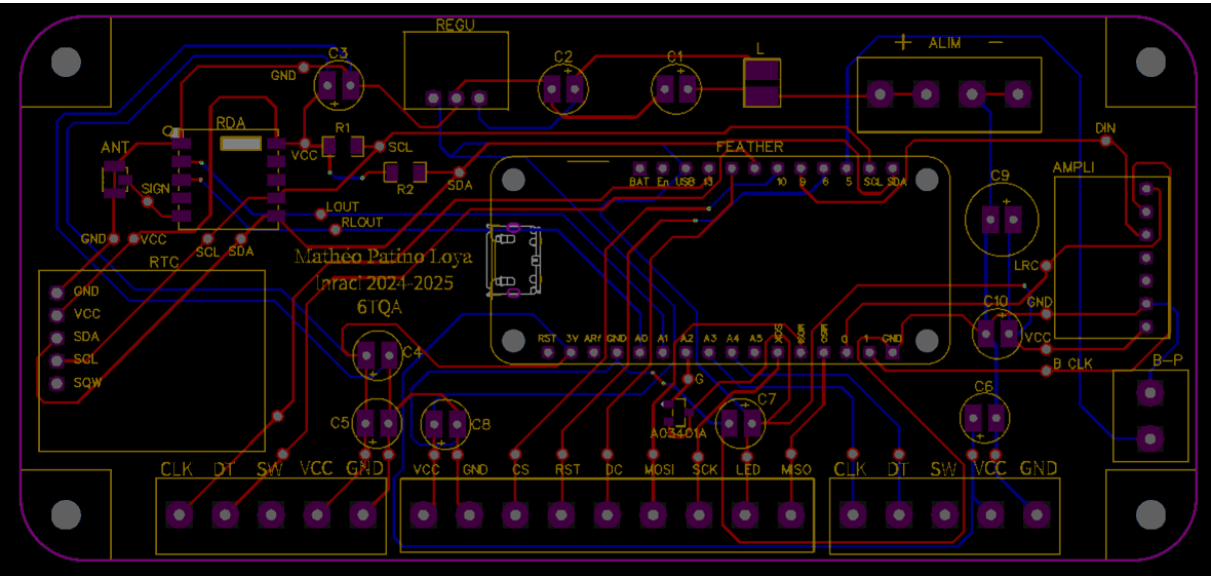
Date de consultation (13/5/2025)

Les annexes

Schéma de principe sans connecteur

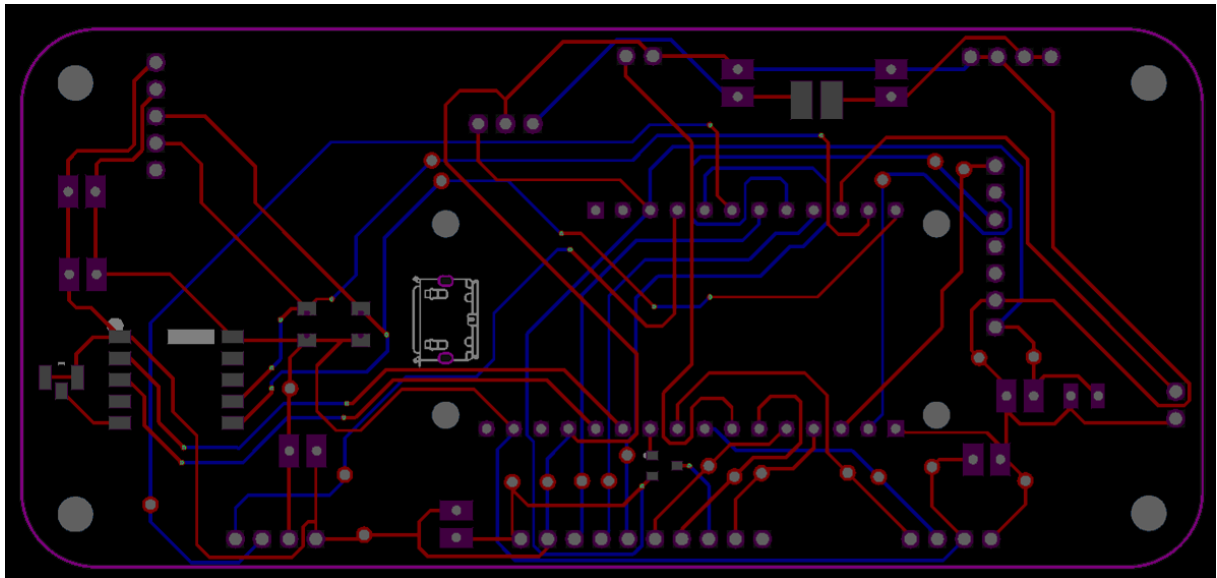
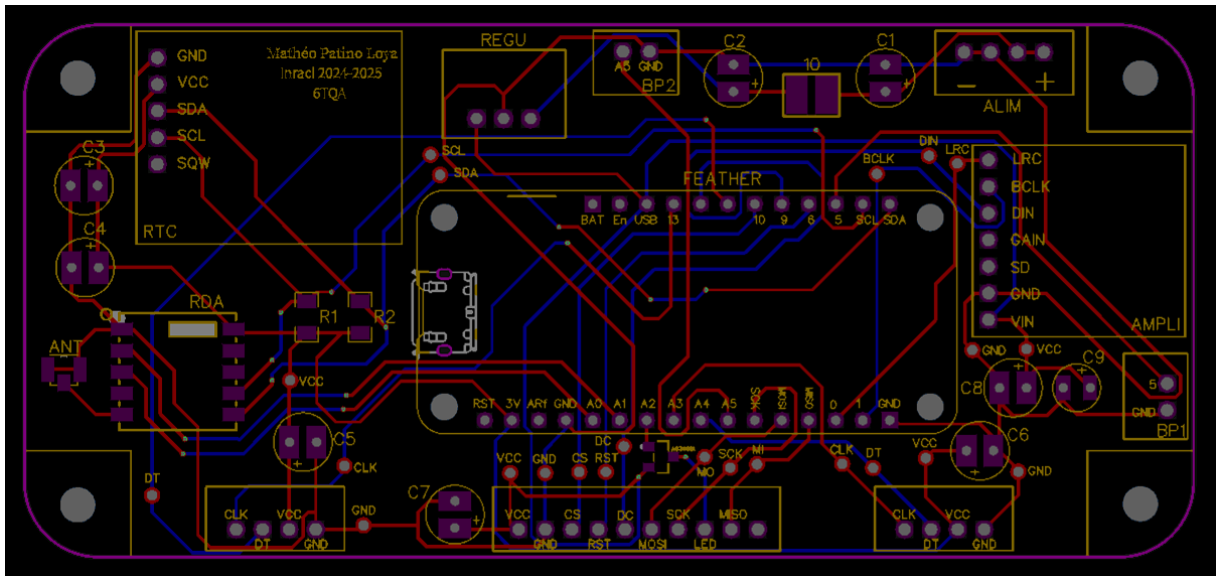


PCB Sérigraphie V1



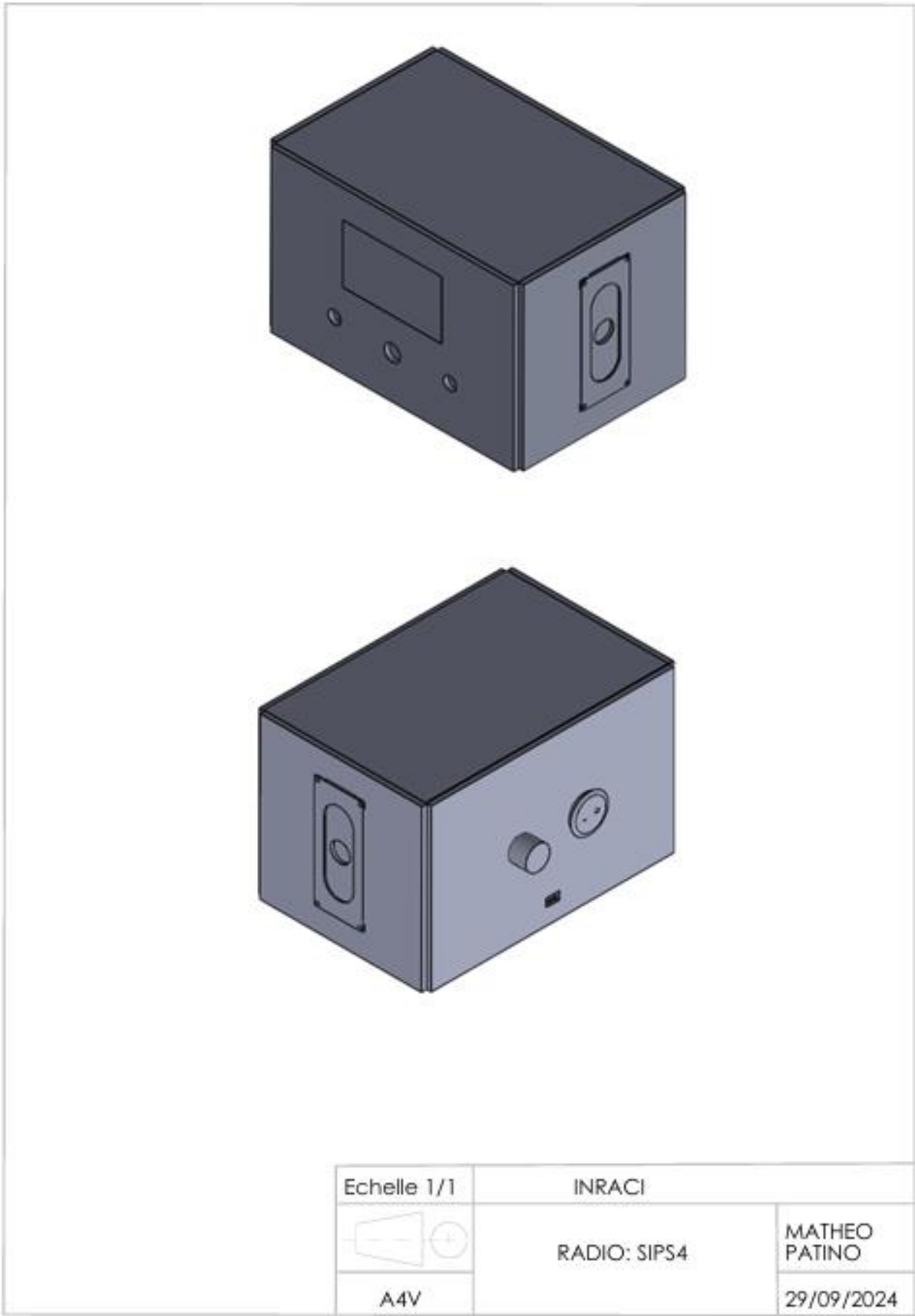
Lien Gerber

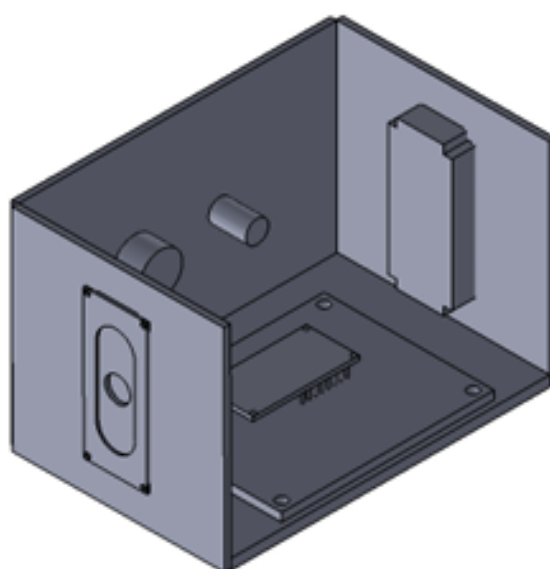
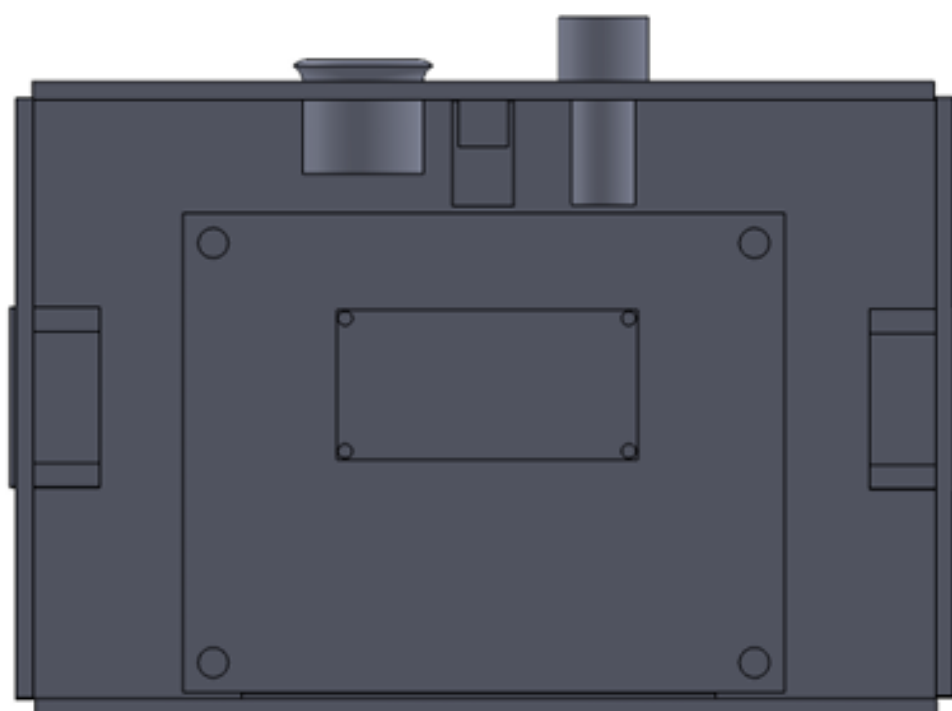
PCB Sérigraphie V2



Lien Gerber

Croquis





Echelle 1/1	INRACI	
	RADIO: SIPS4	MATHEO PATINO
A4V		29/09/2024

Liste du matériel

Régulateur 5V : [R-78E5.0-0.5 Recom Power](#) | Alimentations à montage sur carte | [DigiKey](#)

Bobine : [RLS-126 Recom Power](#) | Inductances, bobines, bobines d'arrêt | [DigiKey](#)

Connecteur UFL : [U.FL-R-SMT-1\(10\) Hirose Electric Co Ltd](#) | Connecteurs, interconnexions | [DigiKey](#)

Resistance : [RC1206FR-074K7L YAGEO](#) | Résistances | [DigiKey](#)

Transistor : [AO3401A UMW](#) | Produits semi-conducteurs discrets | [DigiKey](#)

Ecran : [Hailege 2.8" ILI9341 240 * 320 TFT Écran LCD Écran Tactile SPI Série 5V / 3.3V Raspberry Pi STM32](#) : [Amazon.com.be: Industrie et science](#)

Encodeur linéaire : [QWORK Lot de 5 KY-040 360 Degree Rotatif Encodeur Module avec Capuchon de Bouton pour Arduino](#) : [Amazon.com.be: Industrie et science](#)

Bouton poussoir : [Gebildet 12pcs 12mm Étanche Momentané Bouton Poussoir SPST ON/OFF 3A/3V-5V-9V-12V-24V-230VAC, Débloable Rond Bouton, pour Trompette de Voiture, PC, Lampe de Table, Sonnette de Maison](#) : [Amazon.com.be: Auto et moto](#)

Fusible : [10 Pièces 10a 250 V Porte-fusible Montage sur Panneau 5 X 20mm Porte-fusible Bouchon à Vis pour équipement électrique et Motos](#) : [Amazon.com.be: Auto et moto](#)

Cable coaxial : [TUOLNK IPX MHF1 vers SMA, câble coaxial SMA femelle vers U.FL IPX RG178 pour module GPS Mini PCI UFL vers SMA WiFi Câble d'extension d'antenne 10 cm 5 pièces](#) : [Amazon.com.be: High-tech](#)

RTC : [Hailege PCF8523 Real Time Clock RTC Breakout Module PCF8523 Real Time Clock Assembled Breakout Board 3.3V 5V Time Clock for Arduino](#) : [Amazon.com.be: Industrie et science](#)

Module radio : [Lot de 10 modules radio FM stéréo RDA5807M RRD-102V2.0 sans fil 2,7-3,6 V 76-108 MHz pour téléphone portable, ordinateur portable et MP3](#) : [Amazon.com.be: High-tech](#)

Ampli Max 98357A : [Adafruit I2S 3W Class D Amplifier Breakout - MAX98357A \[ADA3006\]](#) : [Amazon.com.be: High-tech](#)

Condensateur [Voarge Lot de 500 condensateurs électrolytiques 24 valeurs, condensateur électrolytique 0.1uF-1000uF, lot de différents assortiments noir avec boîte en plastique transparent](#) : [Amazon.com.be: Industrie et science](#)

Connecteur JST : [Connecteurs ZH 1,5 mm et câbles en silicone pré-sertis 28 AWG compatibles avec connecteur JST ZH 1,5 2/3/4/5/6/7/8/9/10 broches avec câble adaptateur de boîtier de 160 mm de long](#) : [Amazon.fr: High-Tech](#)

Antenne : [Vecys Lot de 2 antennes radio FM portables télescopiques pour radio TV, réception stéréo FM/AM, 8 sections](#) : [Amazon.com.be: High-tech](#)