INRACI
Avenue Jupiter, 188
1190 Bruxelles

Technique de qualification
Electronique
Année scolaire 2024-2025

Table musicale

Verlinskiy Alexander

Les remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué à la réalisation de ce projet.
Mes professeurs :

À Monsieur Mazzeo, pour ses conseils techniques avisés et son accompagnement tout au long de la phase de conception.

À Monsieur Kapita, dont l'expertise en électronique a été déterminante pour résoudre les défis liés au câblage et à l'optimisation du PCB.

À Monsieur Pochet, pour son soutien méthodologique et ses ressources pédagogiques essentielles.

À Madame Hekster, dont les cours m'ont aidé à écrire en français presque sans fautes.

Ma famille
Un merci particulier à mes parents, pour leur soutien indéfectible, tant moral que financier, et pour leurs précieux conseils lors des choix critiques du projet.

Mes amis
Je remercie mes amis pour leur encouragement constant et leur présence, qui ont rendu ce parcours intellectuellement stimulant et humainement enrichissant.

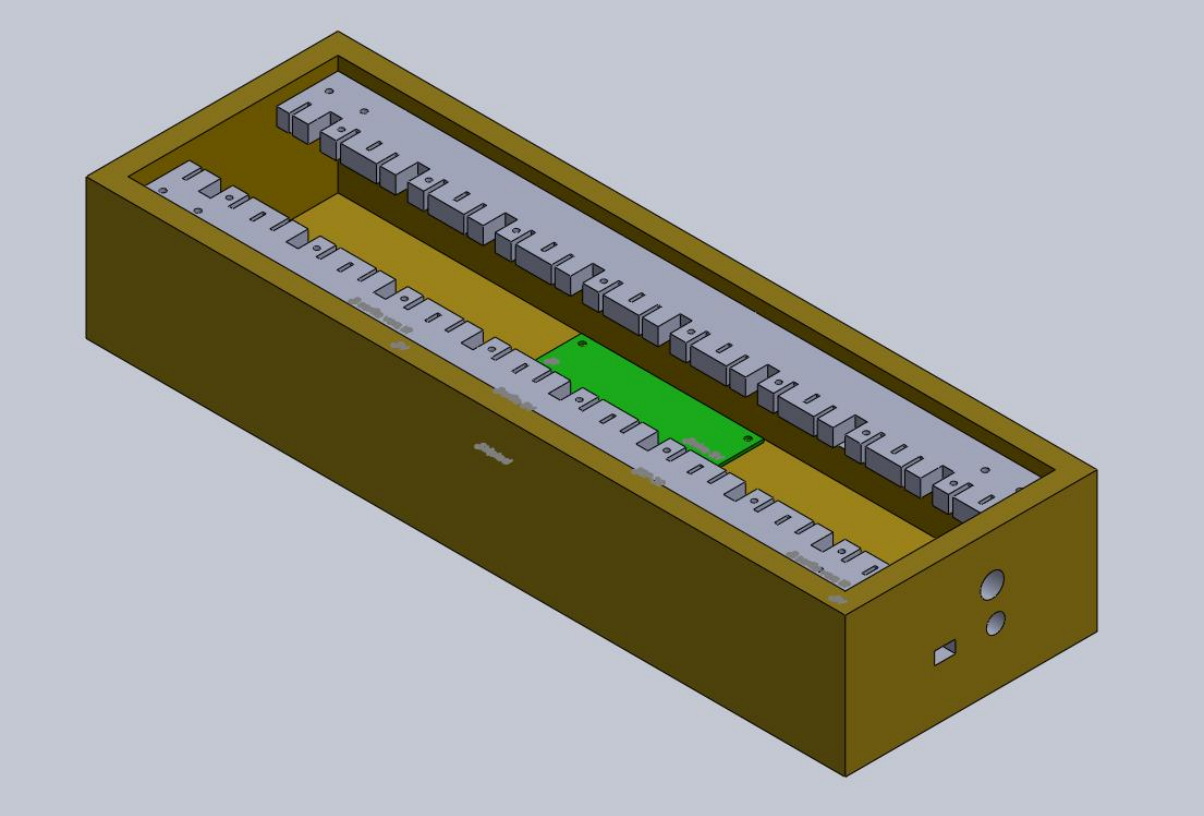
Ce projet n'aurait pu aboutir sans l'apport collectif de chacun d'entre vous.

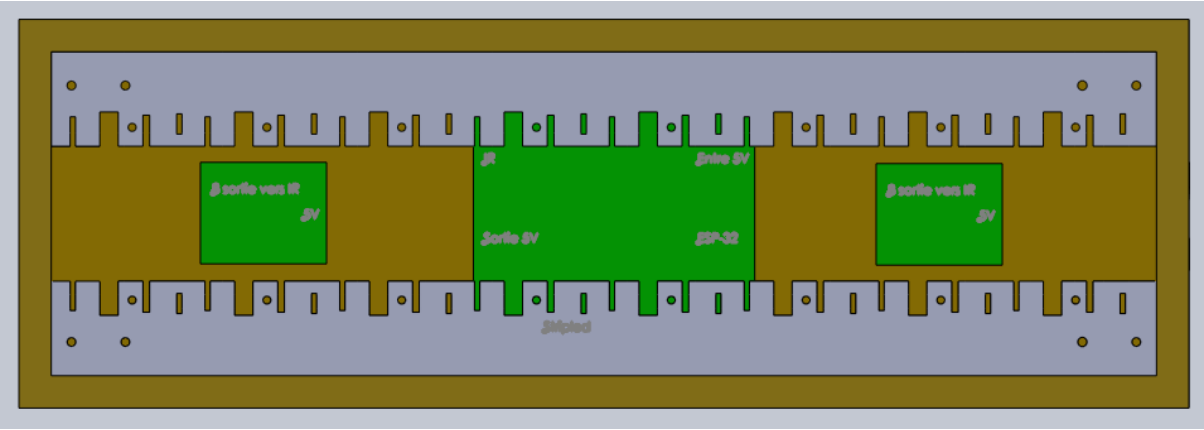
Sommaire

Tout d'abord je voudrai vous prévenir que ça fait 3 ans que je suis en Belgique et avant je n'ai jamais parlé français, j'écris ce message pour vous prévenir et que vous ne soyez pas surpris en lisant le mémoire.

1)L’introduction

Bonjour, je vais vous présenter mon projet, qui est la table musicale. Elle utilise le rayon infrarouge pour allumer les différentes couleurs et pour produire des notes. J’ai choisi ce projet car je suis intéressée par la musique, et, bien que je n’aie pas une raison précise, je suis également fascinée par la lumière.





Malheureusement, je n’ai pas dessiné les infrarouges et les LEDs, mais vous pouvez voir une représentation 3D de mon idée.

2)Les caractéristiques

2.1) General :

Je vous présente les caractéristiques générales de mon projet : une table musicale équipée de 16 capteurs infrarouges, dont 8 notes, 2 réglages de son, 2 fonctions d'enregistrement et de lecture, 2 contrôles pour augmenter ou diminuer la gamme de fréquence, et 2 pour jouer une musique simple (musique de base). J'utilise 32 LED RVB : 2 par capteurs.

2.2) Electrique :

Alimentation en USB-C -5V
Consommation de 0.184 A
Puissance de Haut-Parleur de 4 W
Haut-Parleur de 8 ohm
Résolution signal audio 16 bits
Résolution de volume : 0-9

2.3) Mécanique :

MDF + plexiglas
433x45x114 mm
1.55kg

3)Les principes de mis en joue

3.1) L’infrarouge :

L'infrarouge est une forme de rayonnement électromagnétique qui utilise des fréquences basses pour fonctionner. Son principe de fonctionnement est simple : il détecte la chaleur émise dans son spectre de détection. La fréquence de l'infrarouge est relativement basse. Il existe trois types d'infrarouge : l'infrarouge proche (NIR) (Near-Infrared), l'infrarouge moyen (MIR) (Midl-Infrared) et l'infrarouge lointain (FIR) (Far-infrared). L'infrarouge est utilisé dans divers domaines : par exemple, le NIR est utilisé dans les télécommandes, le MIR pour l'étude des gaz dans l'atmosphère, et le FIR pour des applications associées à des températures plus basses, comme la détection de chaleur corporelle.

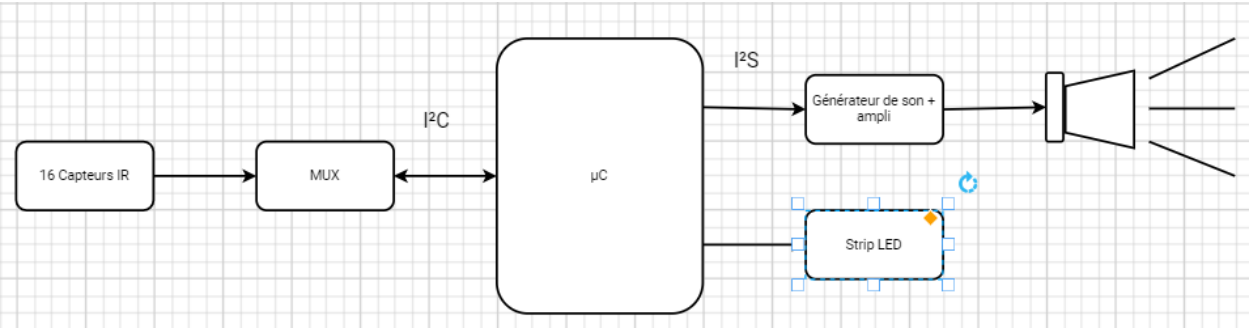
3.2) La lumière :

La lumière que nous voyons n'est qu'une petite partie du spectre électromagnétique. Il y a : les rayons gamma (très courtes longueurs d'onde), les rayons X, l'ultraviolet (UV), la lumière visible (ce que nous voyons), l'infrarouge (IR), les micro-ondes et les ondes radio (très longues longueurs d'onde). La vitesse de la lumière dans le vide est d'environ 300 000 kilomètres par seconde. Toutes les couleurs que vous voyez sont le mélange de trois couleurs de base : le rouge, le bleu et le vert. Par exemple, le blanc est le résultat de la combinaison de toutes les couleurs.

3.3) Le son :

Le son est un phénomène physique qui correspond aux oscillations d’ondes dans l’espace, que l’on peut entendre. La gamme de fréquences audibles par l’être humain se situe entre 20 Hz et 20 kHz. Par exemple, lorsque je parle, je produis un son qui se propage dans l’espace sous forme d’ondes, que l’on peut percevoir.
Il existe également les infrasons et les ultrasons, que nous ne pouvons pas entendre. Le mot ultra indique que le son est au-dessus de la gamme audible, tandis qu’infra signifie qu’il est en dessous.

4)Le schéma bloc



Capteur IR (infrarouge)
Le capteur infrarouge détecte la présence d'un doigt et envoie cette information au microcontrôleur en passant par un multiplexeur (MUX). La communication avec le MUX se fait via un bus I²C.

MUX (Multiplexeur)
Le multiplexeur est bidirectionnel : il reçoit des signaux du capteur et du microcontrôleur. Il transmet les données du capteur vers le microcontrôleur et inversement, toujours via le bus I²C.

Microcontrôleur
Le microcontrôleur reçoit l'information issue du capteur infrarouge (via le MUX). Il traite ces données et envoie les commandes nécessaires :

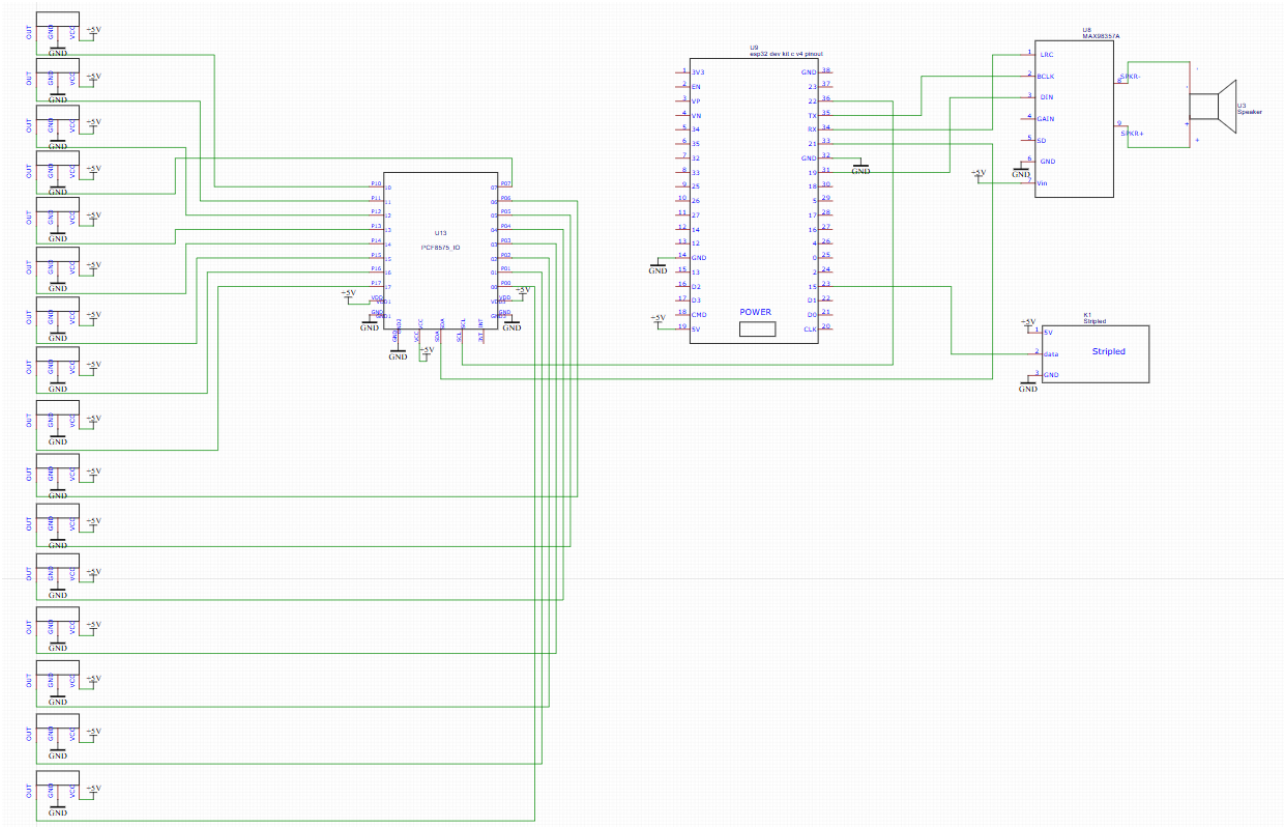
- au strip LED pour allumer la LED correspondante,
- au générateur de son pour produire le son approprié.
  La communication avec le générateur de son se fait via un bus I²S.

Strip LED
Le strip LED reçoit les instructions du microcontrôleur et allume la LED désignée.

Générateur de son
Le générateur de son crée le son demandé, puis l'amplifie.

Haut-parleur
Le haut-parleur diffuse le son généré par le générateur.

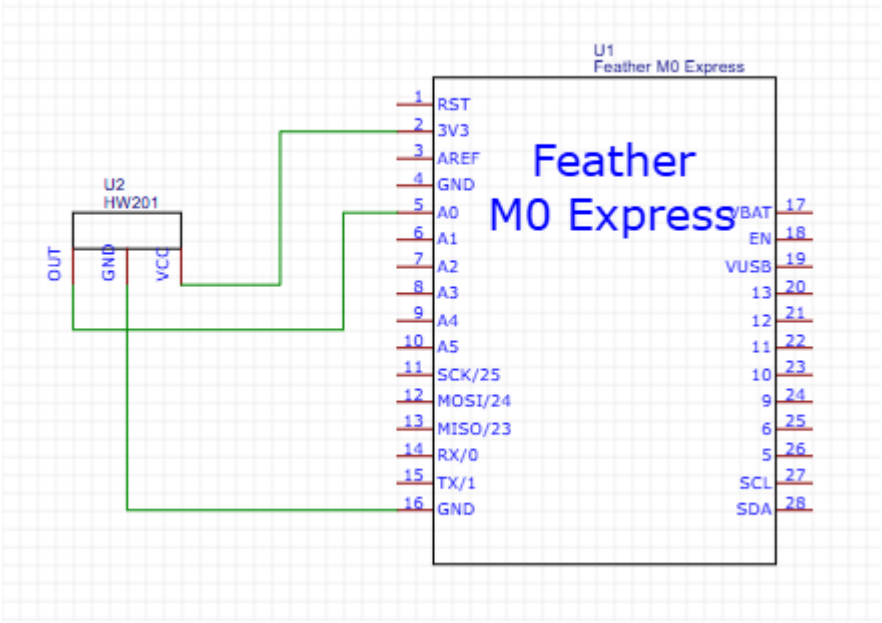
5)Les études détaillées



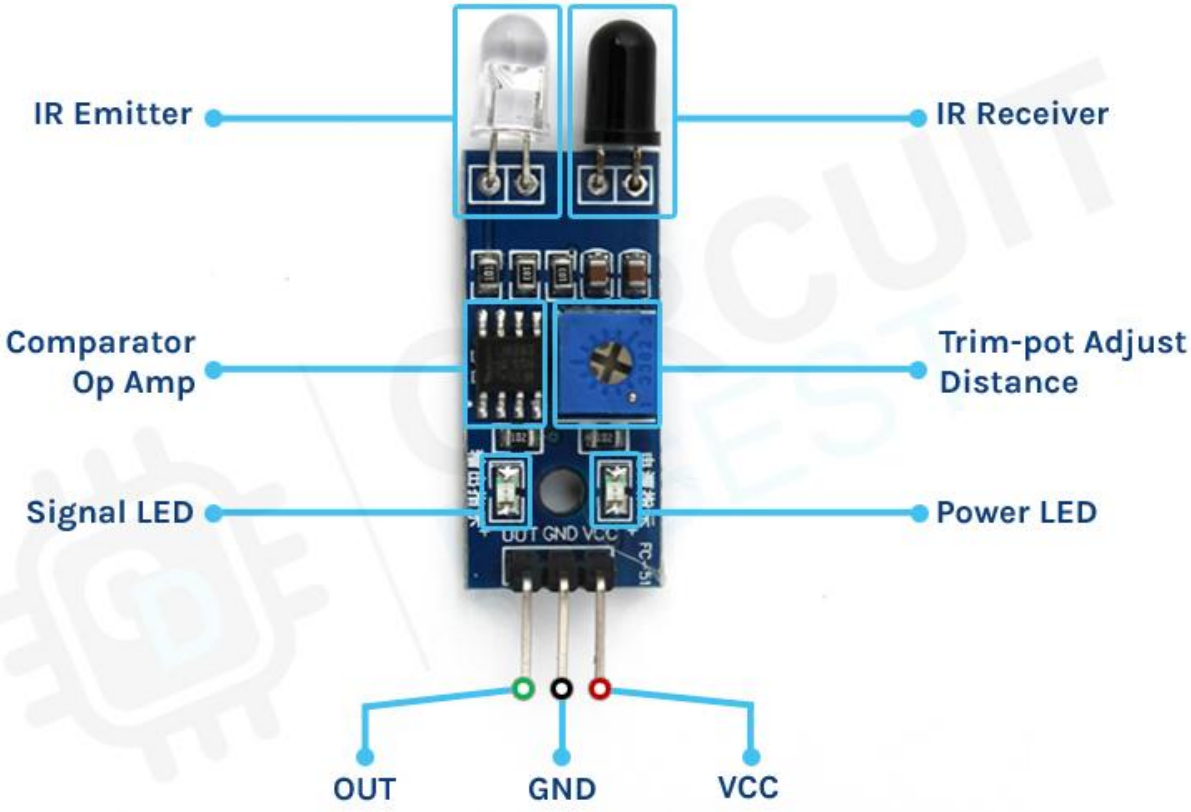
Comme vous pouvez voir sur le schéma de principe. Tout à gauche, vous voyez les petits carrés, ce sont les infrarouges. Ensuite, on a un MUX qui nous indique quel infrarouge est allumé. Un rectangle représente le microcontrôleur (mC). Juste à droite, il y a deux blocs : l'un est le strip LED, l'autre est l'ampli classe D.

Je vais d'abord analyser l'infrarouge

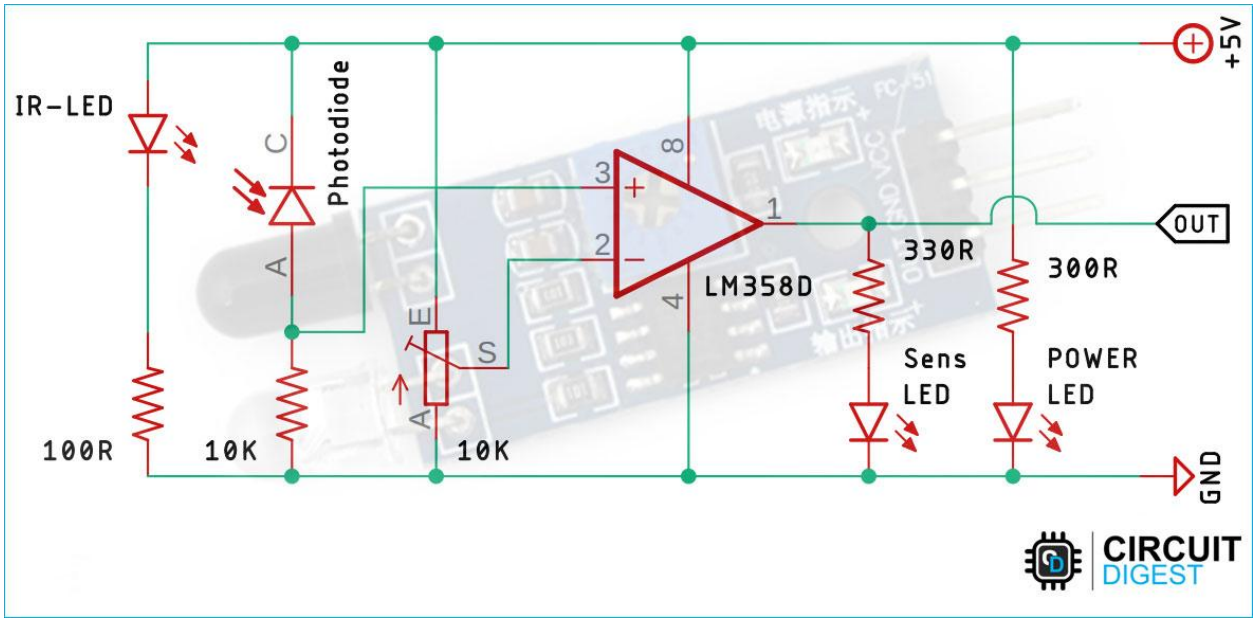
5.1) Infrarouge



Sur cette photo vous pouvez voir les différents composants et leurs explications :



Explication de schéma :



On utilise AOP dans le circuit HW-201. On rentre 5V et connecte avec GND, on allume directement le LED disant nous qu'il a le courant. Le AOP est asymétrique car on connecte 5V et le GND à la place de -5V. Sur le moins de AOP on met la résistance variable qui va définir la distance entre le capteur et l'objet qu'on capte. Sur le plus de AOP on met la photodiode et comme ça on compare si la distance de la résistance variable et dit si la résistance variable > ou < que ce que la photodiode capte. À la sortie de AOP on a le LED qui va nous dire directement si on capte quelque chose ou pas.

Programme de test :

```
#include <Servo.h>

Servo servo;
const int analogInPin = A0;  // Set the 'out' signal of the Sensor to this pin (A0)
int val = 0;                 // Set the 'out' signal of the Servo to this pin (9)
int servoPin = 9;
int angle = 90;
#define LED 13

void setup() {
  pinMode(LED, OUTPUT);

  servo.attach(servoPin);
  Serial.begin(9600);
  while (!Serial)  // Attente de l'ouverture du moniteur série
  {}
}

void loop() {
  val = analogRead(analogInPin);  // This reads the value of the sensor to see if an object is detected then saves in the val variable.

  if (val < 600) {
    while (angle >= 75)  // Command to move from 1 degree to 90 degrees and 1 equals speed. Opens Clockwise.
    {
      servo.write(angle);
      angle--;
      digitalWrite(LED, 1);
      Serial.println("1");
      //delay(8);  // This controls the speed of the Servo (gate) opening. Higher# lower speed.
    }
    while (angle <= 180) {
      servo.write(angle);  // If object detected the Servo rotates to 90 degrees.
      angle++;
      Serial.println("0");
      digitalWrite(LED, 0);
      //delay(8);  // This controls the speed of the Servo (gate) returning. Higher# lower speed.
    }
  }
}
```

Table 1: Absolute Maximum Ratings

Parameter	Name		Conditions	Min	Typical	Max	Units
Terminal Voltage	Vpeak	HW101		0		2.9	V
		HW201				5.8	
Temperature[1]	Tmax			-40		+85	°C

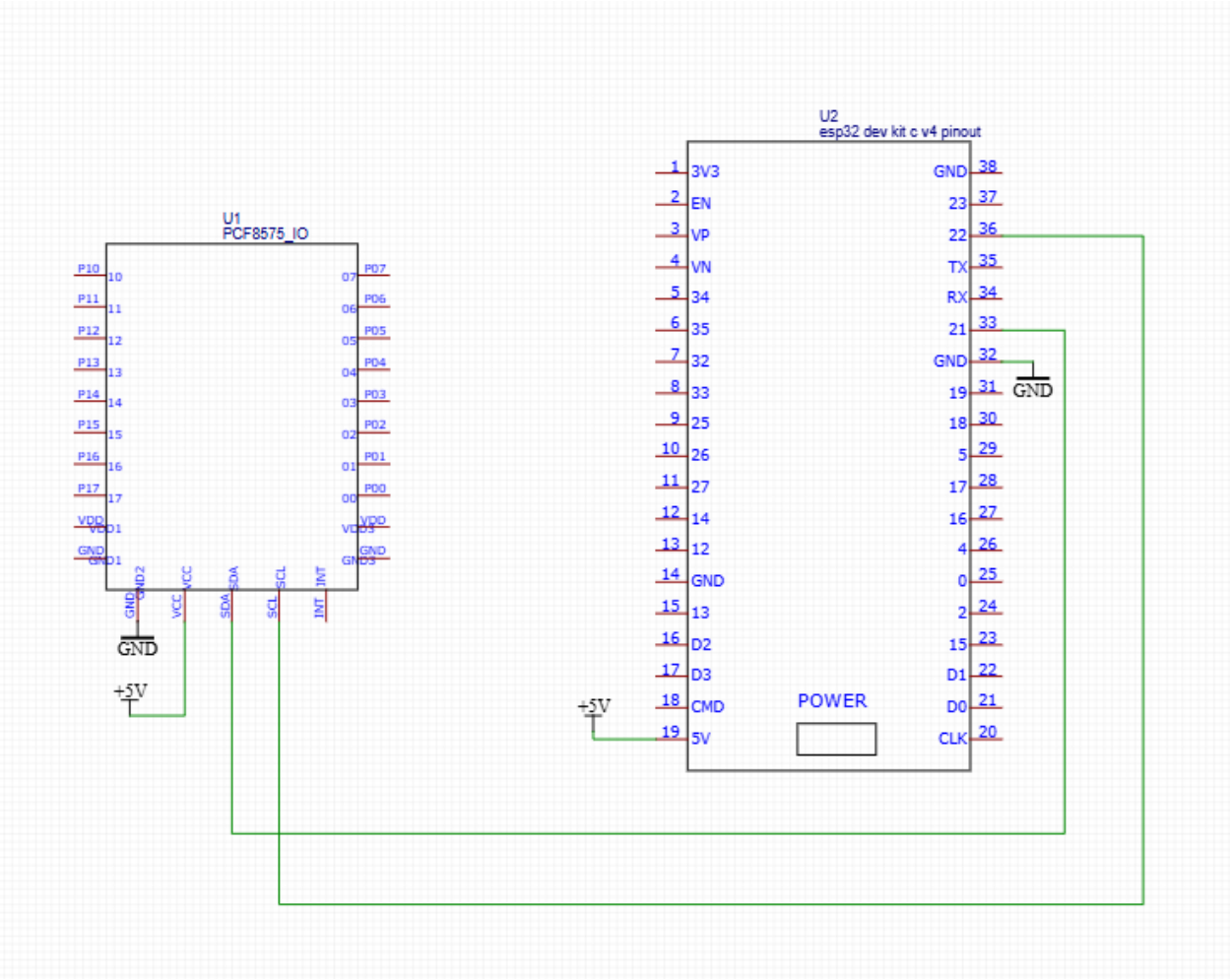
Table 2: Electrical Characteristics

Parameter	Name		Conditions	Min	Typical	Max	Units
Terminal Voltage	Vn	HW101		0		2.75	V
		HW201		0		5.5	
Capacitance	C	HW101	DC, 23°C	608	760	912	mF
		HW201		304	380	456	
ESR	ESR	HW101	DC, 23°C		50	60	mΩ
		HW201			100	120	
Leakage Current	I_L		2.75V, 23°C 120hrs		1	2	µA
RMS Current	I_{RMS}		23°C			3	A
Peak Current[2]	I_P		23°C			30	A

Comme on peut voir sur le datasheet, on peut l'allumer entre 2.75-5.5V. Dans mon cas je l'aliment à 5V.

Deuxièmement c'est le MUX qui va être analysé

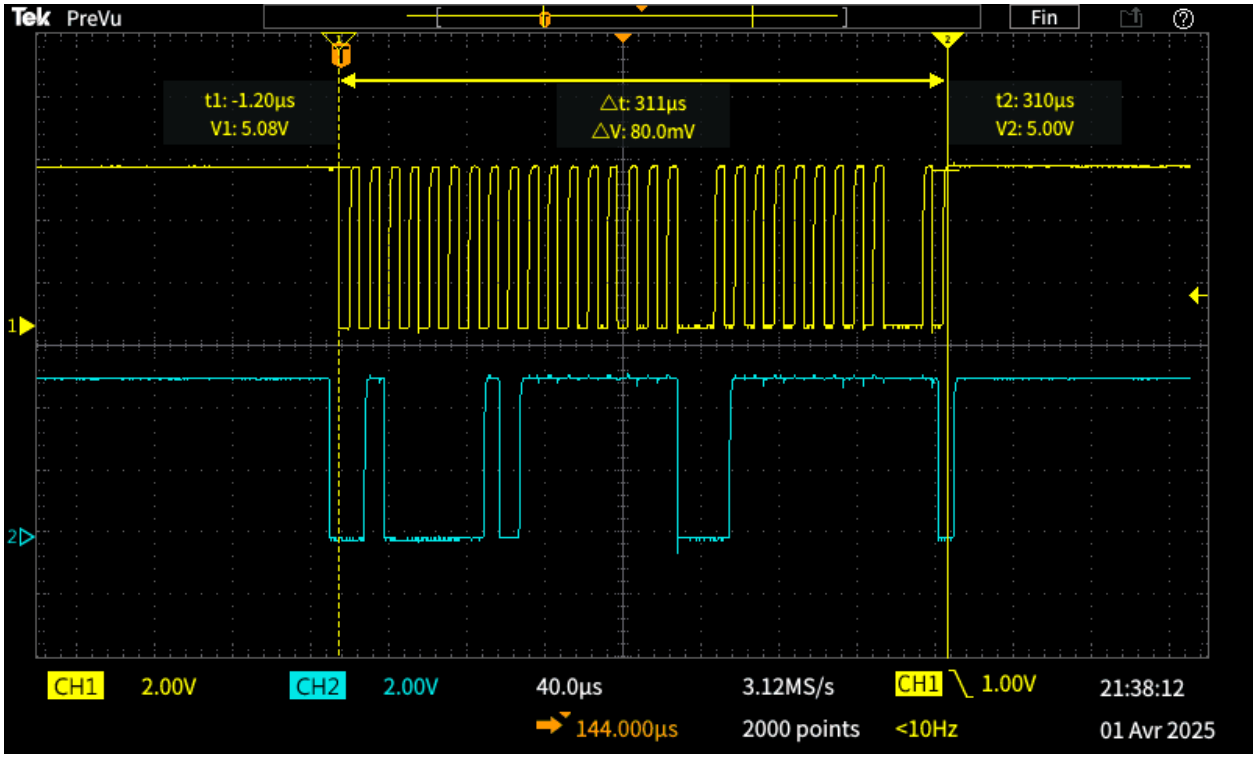
5.2) MUX(I²C)



Le bus I²C est un bus de communication qui fonctionne grâce à une horloge. Je m'explique : chaque bit transmet une information, et lorsque cette information est nécessaire, l'horloge indique au maître de lire le ou les bits.

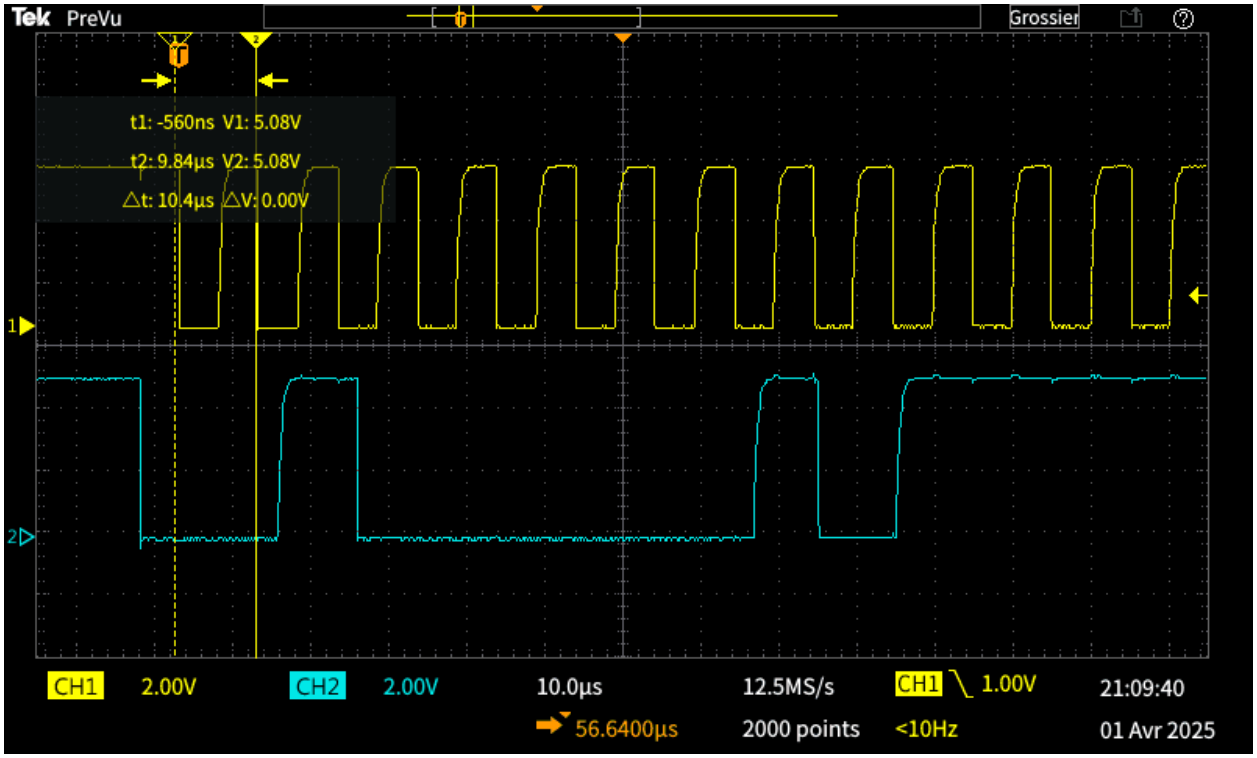
C'est pourquoi vous verrez toutes les mesures avec deux couleurs différentes, que je vais vous indiquer en haut ou en bas.

Vous pouvez voir le full tram des données de I²C sur cette photo. Dans mon exemple c'est PCF8575.



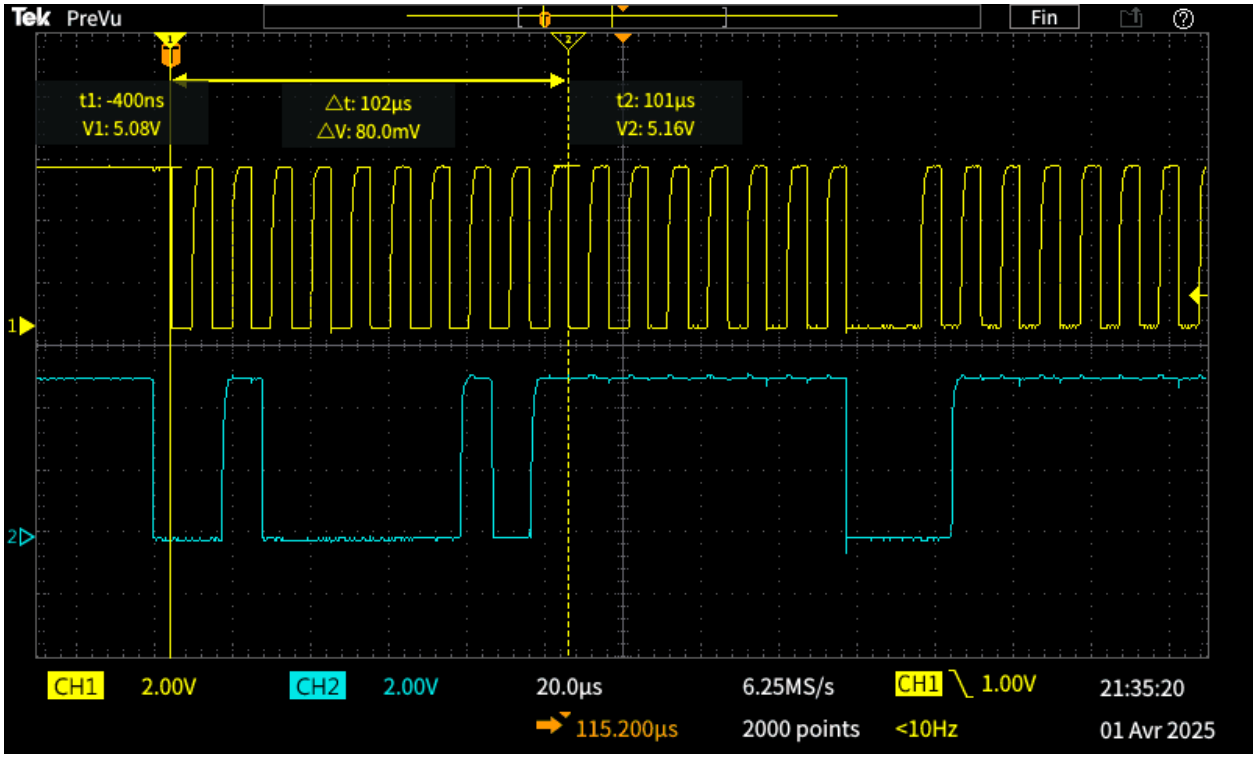
Le bleu ce sont les données et le jaune est clock.

Sur la photo, on voit la trame de données envoyé vers le PCF. Le paquet de données dure 311µs.Comme on peut voir sur cette image, la durée d'un bit est 10.4µs.



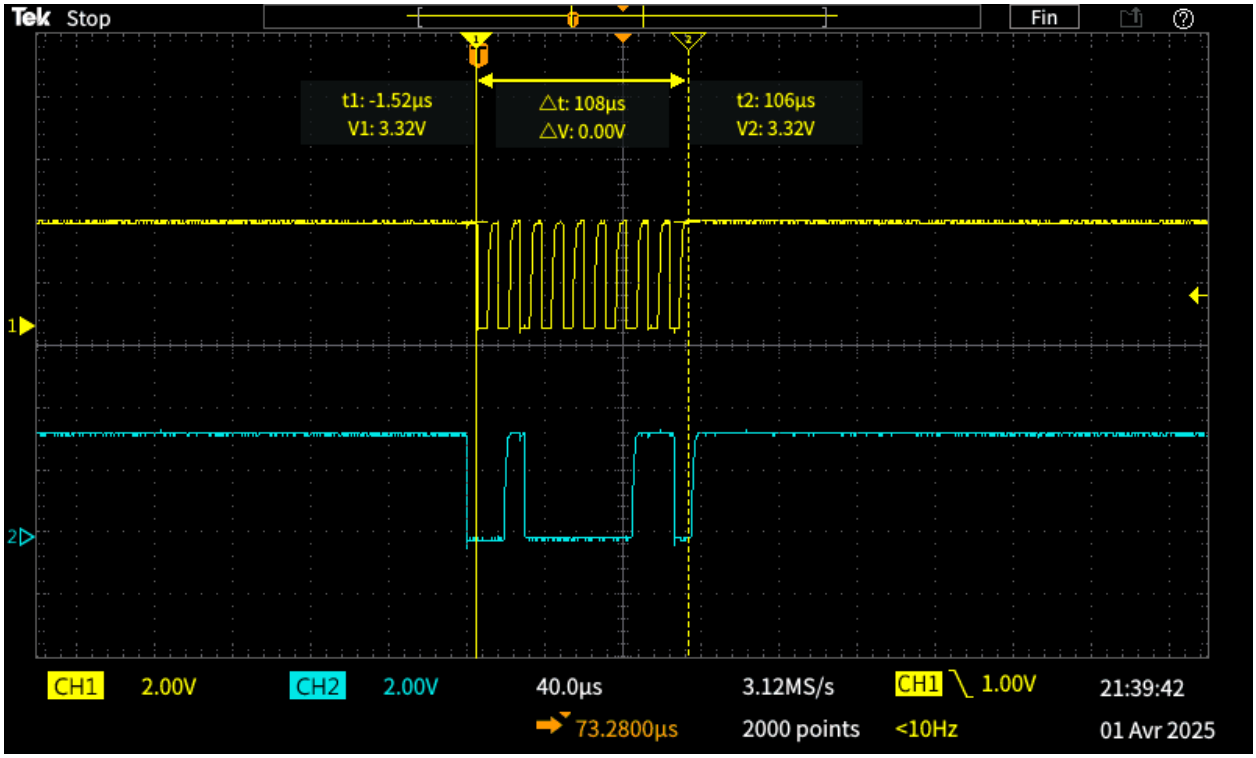
La durée d'un paquet trame donné dure 102µs.

Jaune-clock, bleu-data



Jaune-clock, bleu-data

Ici, on peut observer le tracé des données qui ne sont pas comptées. Le bus envoie les données et en retour il n'a rien, et vu qu'il n'y a rien, il arrête de travailler. Comme on peut voir sur la photo, il travaille 108µs, après quoi le bus s'arrête d'envoyer les données.



Jaune-clock, bleu-data

Voilà le programme:

```ino
#include "PCF8575.h"

PCF8575 PCF(0x20);

int Capteurs;

void setup() {
  Serial.begin(9600);

  Wire.begin();  //Initialisation du bus I2C

  PCF.begin();
}


void loop() {
  Capteurs = PCF.read16();
  Serial.print("Capteurs =  ");
  Serial.println(Capteurs, BIN);
  delay(1000);
}
```

5.6 I²C Interface Timing Requirements

over recommended operating free-air temperature range (unless otherwise noted) (see Figure 6-1)

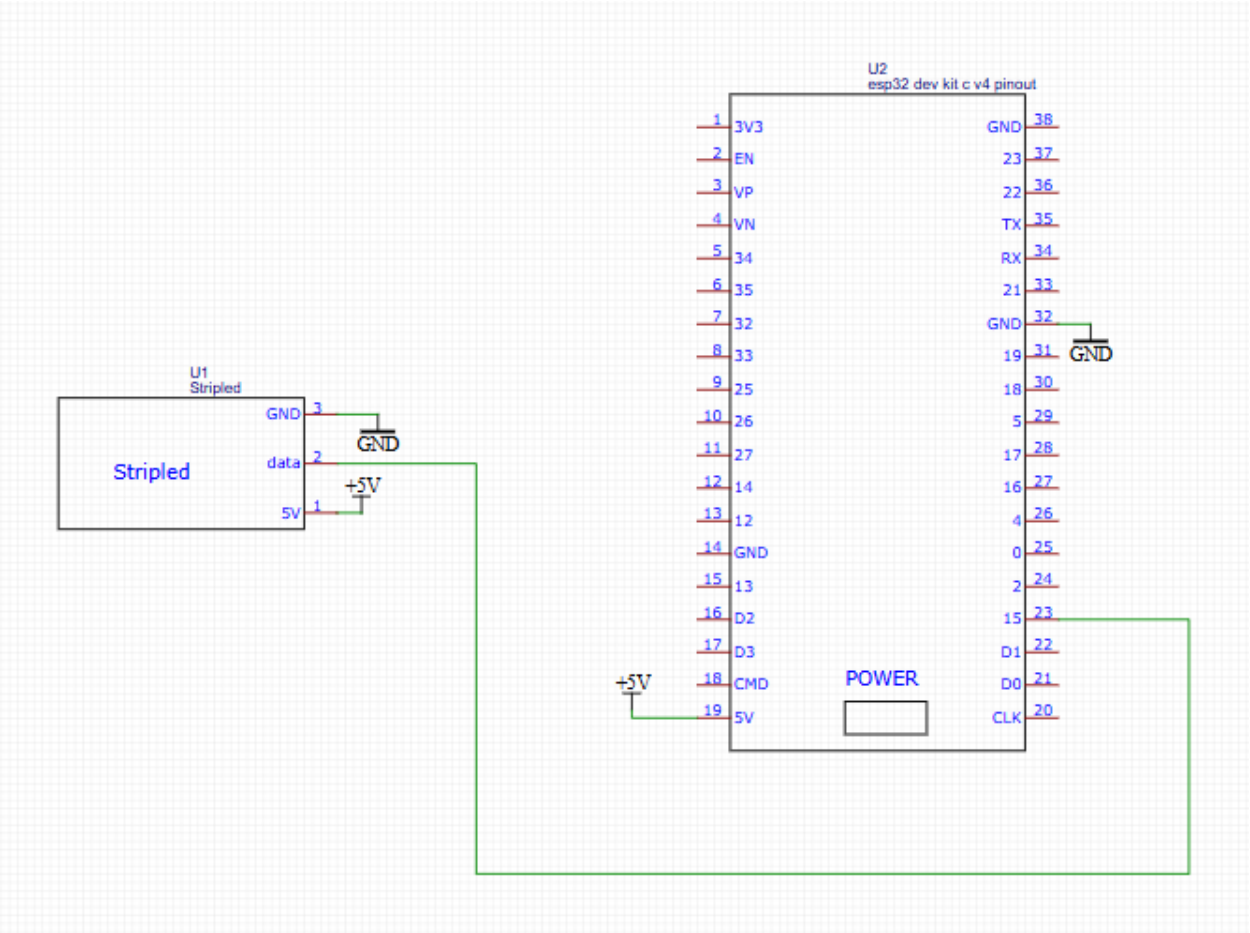
			MIN	MAX	UNIT
f_{scl}	I²C clock frequency			400	kHz
t_{sch}	I²C clock high time		0.6		µs
t_{scl}	I²C clock low time		1.3		µs
t_{sp}	I²C spike time			50	ns
t_{sds}	I²C serial data setup time		100		ns
t_{sdh}	I²C serial data hold time		0		ns
t_{icr}	I²C input rise time		$20 + 0.1C_b$ (1)	300	ns
t_{icf}	I²C input fall time		$20 + 0.1C_b$ (1)	300	ns
t_{ocf}	I²C output fall time	10-pF to 400-pF bus		300	ns
t_{buf}	I²C bus free time between Stop and Start		1.3		µs
t_{sts}	I²C start or repeated Start condition setup		0.6		µs
t_{sth}	I²C start or repeated Start condition hold		0.6		µs
t_{sps}	I²C Stop condition setup		0.6		µs
t_{vd}	Valid-data time	SCL low to SDA output valid		1.2	µs
C_b	I²C bus capacitive load			400	pF

(1) C_b = total bus capacitance of one bus line in pF

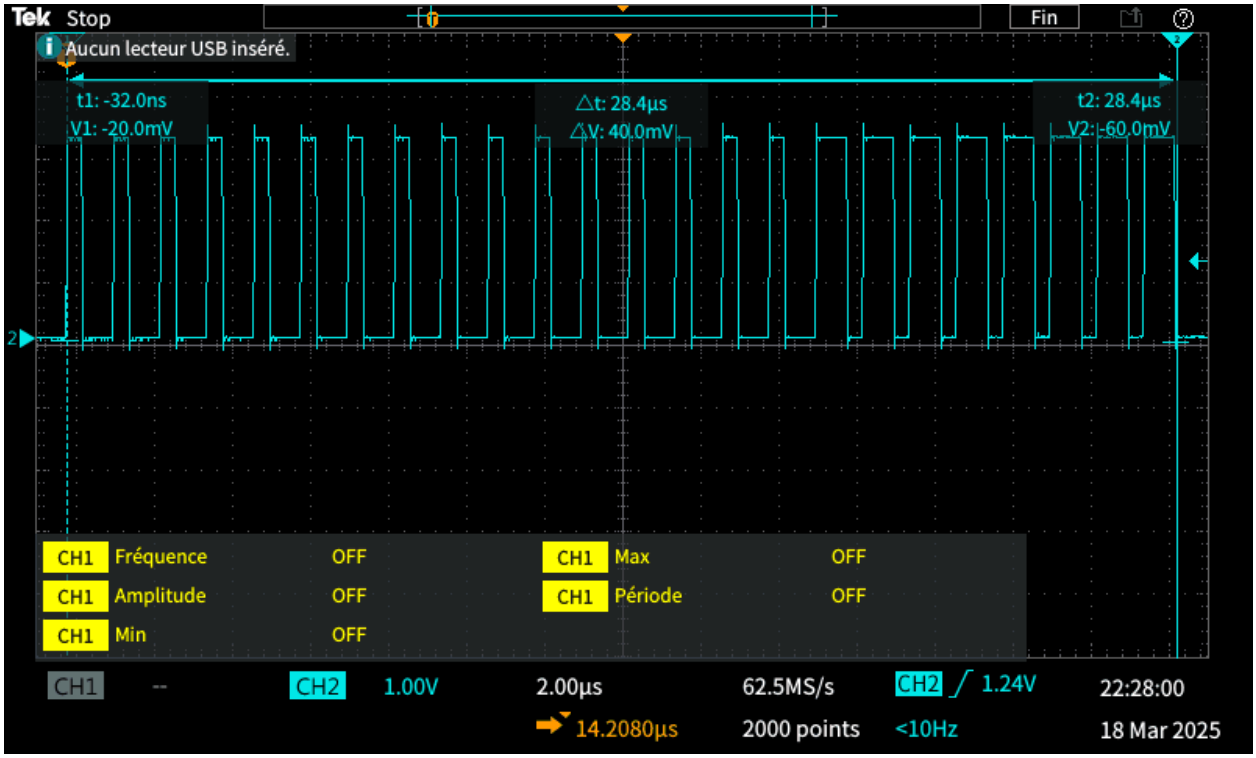
En analysent le datasheet on peut remarquer que la fréquence maximale de clock est 400kHz et dans mon cas c'est environs 100Khz. Pour calculer la fréquence il faut utiliser la formule 1/T, ou le T dans notre cas égale à 10.4µs.
Datasheet : https://www.ti.com/lit/ds/symlink/pcf8575.pdf

Troisièmement, je vais analyser le Neopixel.

5.3) Néopixel



Sur cette photo, vous pouvez voir la trame des données qui dure 28.4µs.

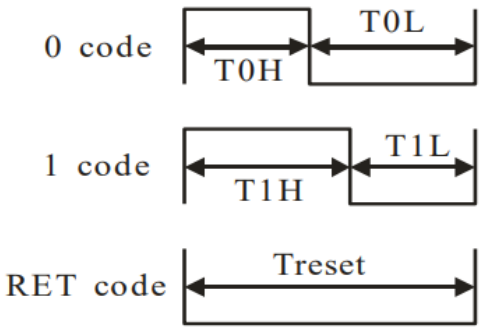


Ici, on peut voir que la durée d'allumage de la LED est de 816 ns. Comme on peut le constater, il y a des impulsions faibles et des impulsions fortes.
Dans les 16 premiers bits, on observe principalement des impulsions faibles, ce qui signifie que les LED sont allumées, mais pas dans l'ordre classique RGB (ou RVB en français), mais dans l'ordre GRB (VRB).

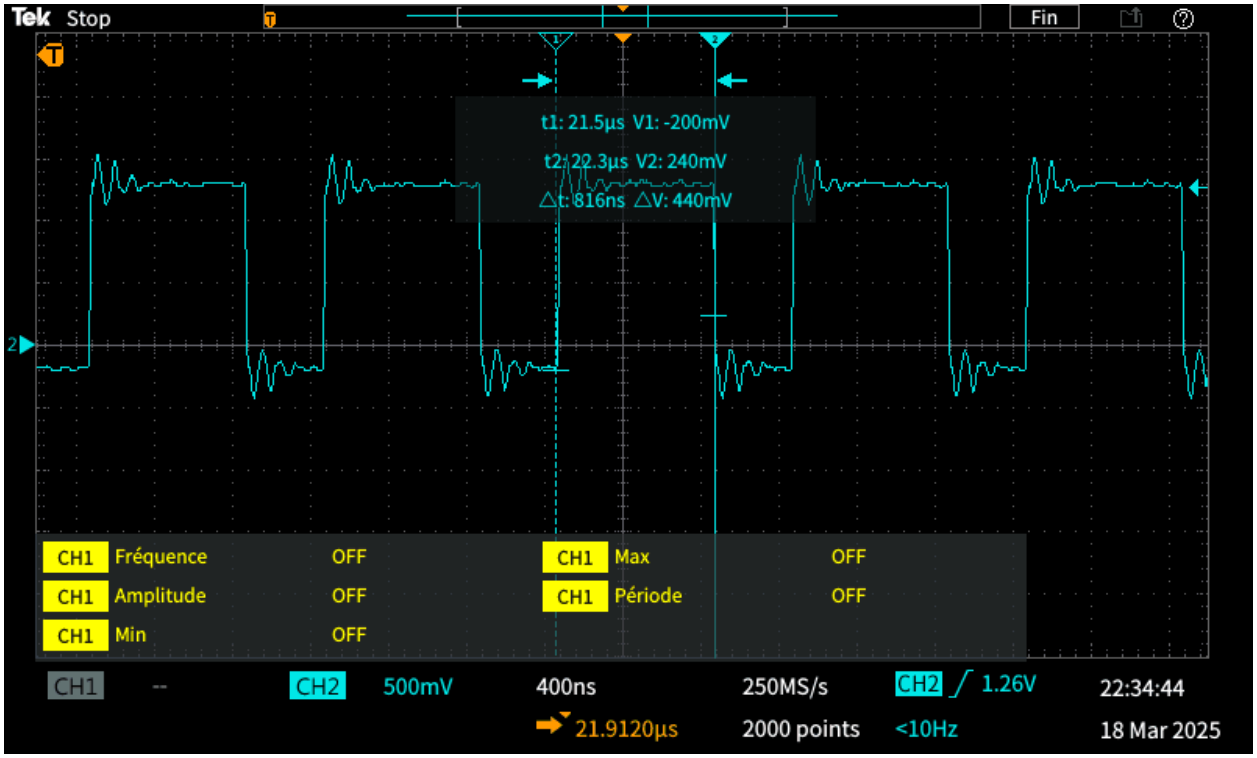
Data transfer time(TH+TL=1.25μs±600ns)

T0H	0 code ,high voltage time	0.4us	±150ns
T1H	1 code ,high voltage time	0.8us	±150ns
T0L	0 code , low voltage time	0.85us	±150ns
T1L	1 code ,low voltage time	0.45us	±150ns
RES	low voltage time	Above 50μs	

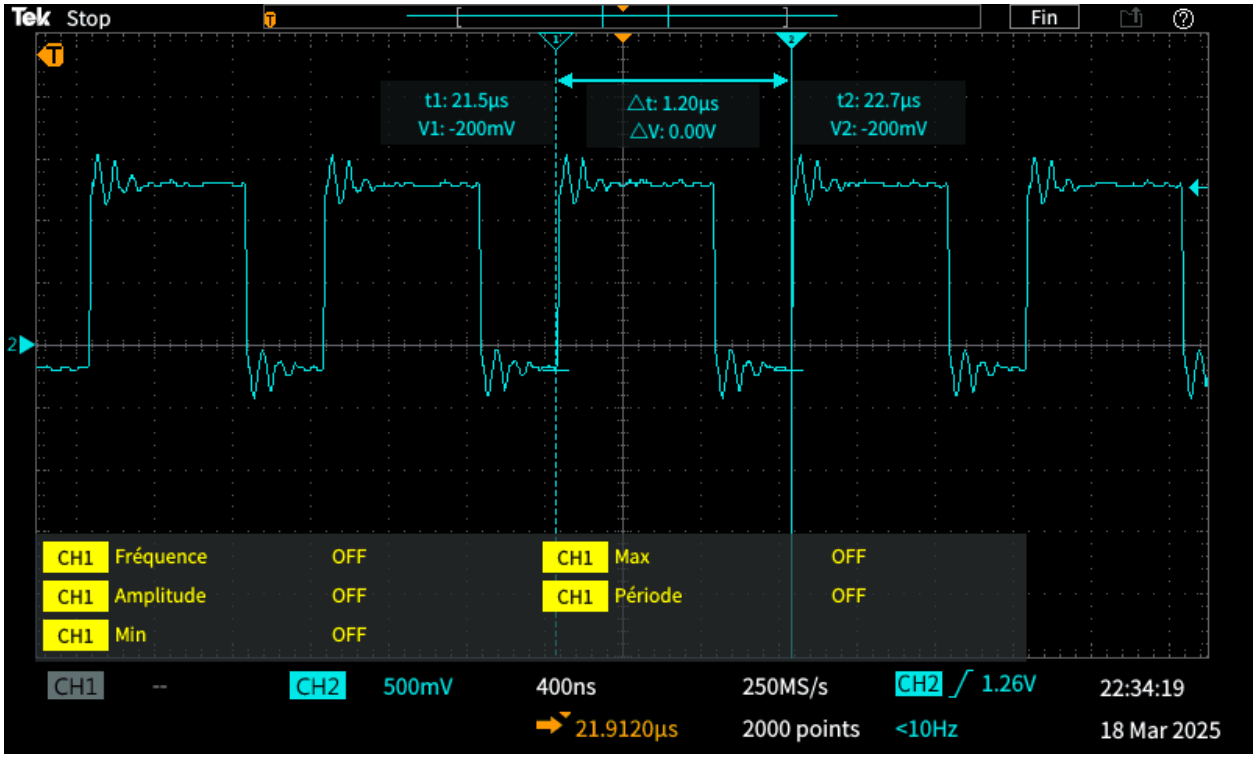
Dans ce tableau, vous pouvez voir que la troisième colonne contient les valeurs typiques, et la quatrième colonne montre les erreurs associées. Le T0H – time 0 high, T1H - time 1 high, T0L – Time 0 low, T1L – Time 1 low, RES – résistor de 300-500ohm pour stabiliser le signal qui rend ou sort.



Le 0 correspond à l'état où la LED est éteinte, et le 1 signifie que la LED est allumée. Et le RET code, ou suivi de Treset que vous voyez, correspond à la valeur de retour.

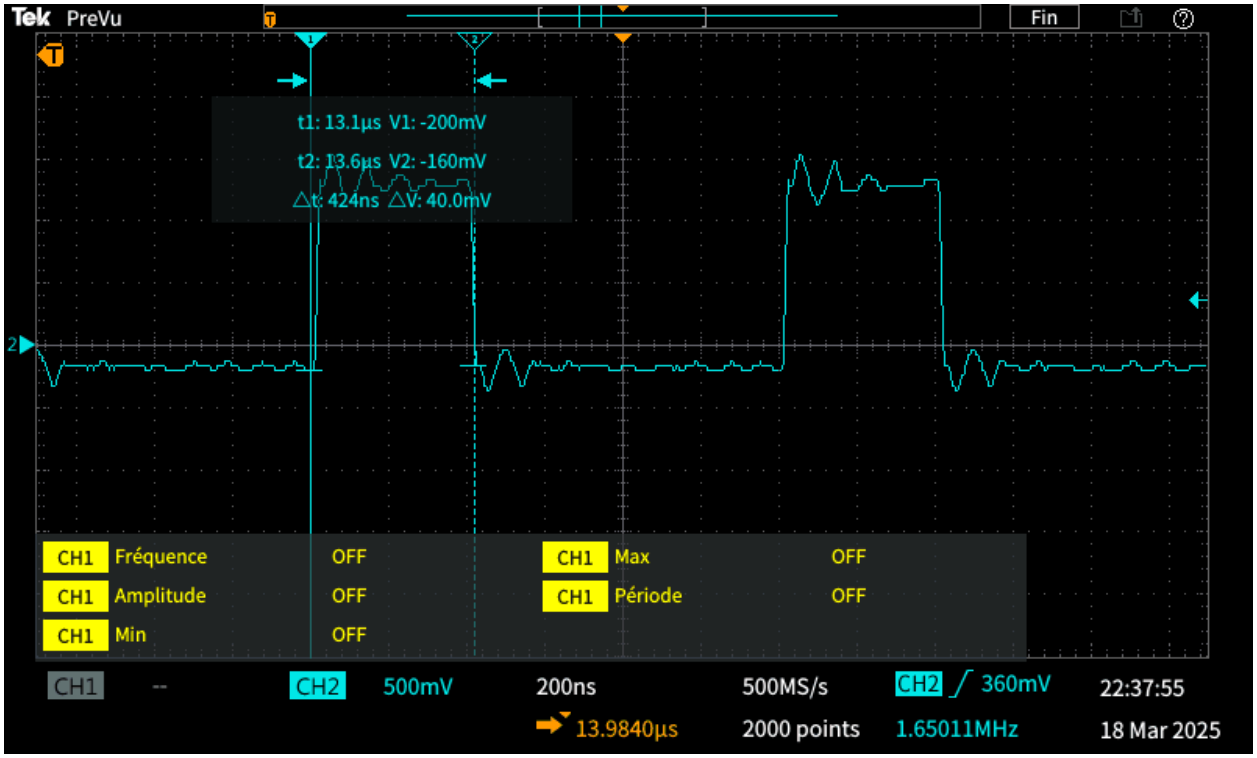


Comme on peut le constater, la durée de T1H est de 816 ns, ce qui correspond à la valeur indiquée dans la datasheet.



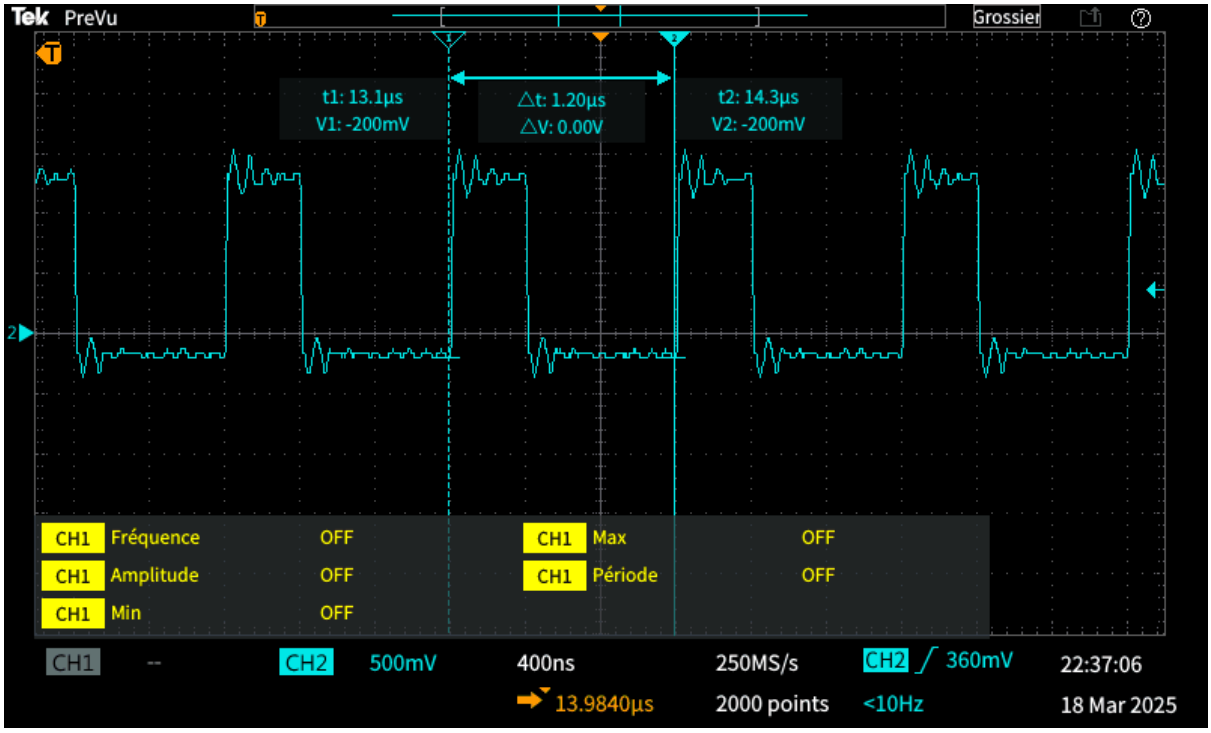
La durée d’un bit est 1.20µs

Et ça c’est le ON de LED



Le temps de T0H est 424ns.

Et enfin je vais vous montrer le LED qui est étant.



Comme on peut voir, le temps est à 1.20µs, qui correspond au datasheet.

Et voilà le programme:

```ino
#include <Adafruit_NeoPixel.h>

#define PIN 6
#define NUMPIXELS 30

// Créer un objet NeoPixel pour contrôler la bande de LED
Adafruit_NeoPixel strip = Adafruit_NeoPixel(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

void setup() {
  strip.begin();
  strip.show();
}

void loop() {
  for (int i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, strip.Color(255, 255, 255));
  }
  strip.show();
  delay(100);
}
```

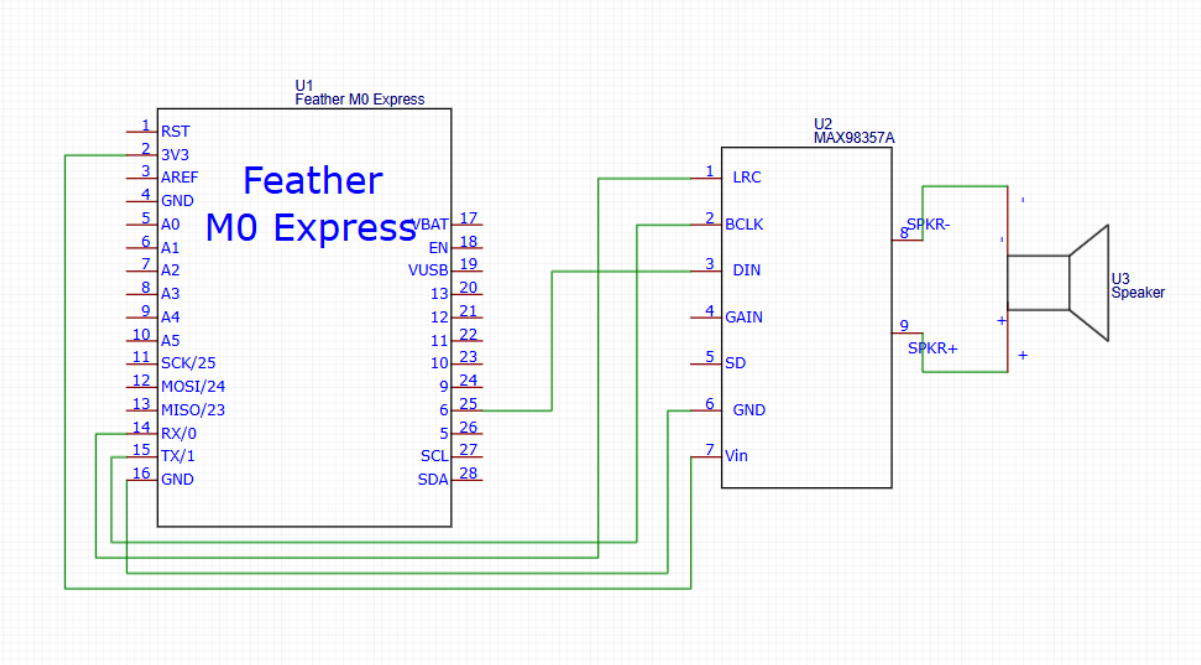
RGB IC characteristic parameter

Emitting color	Model	Wavelength(nm)	Luminous intensity(mcd)	Voltage(V)
Red	13CBAUP	620-625	390-420	2.0-2.2
Green	13CGAUP	522-525	660-720	3.0-3.4
Blue	10R1MUX	465-467	180-200	3.0-3.4

Sur cette extrait de datasheet vous pouvez voir quelle couleur besoin de Volte il besoin pour allumer la couleur, ça luminosité et ça fréquence dans le spectre électromagnétique.
Datasheet : https://cdn-shop.adafruit.com/datasheets/WS2812B.pdf

Et enfin je vous explique c'est que l'ampli classe D.

5.4) Ampli classe D



Rôle des signaux :

LRC (Left-Right Clock) : Signal utilisé pour indiquer si les données actuelles concernent le canal gauche ou le canal droit.

BCLK (Bit Clock) : Signal d'horloge qui synchronise la transmission des bits. Sur flanc montant

DIN (Data In) : Entrée de données, utilisée pour recevoir un signal de données.

GND (Ground) : La masse de circuit.

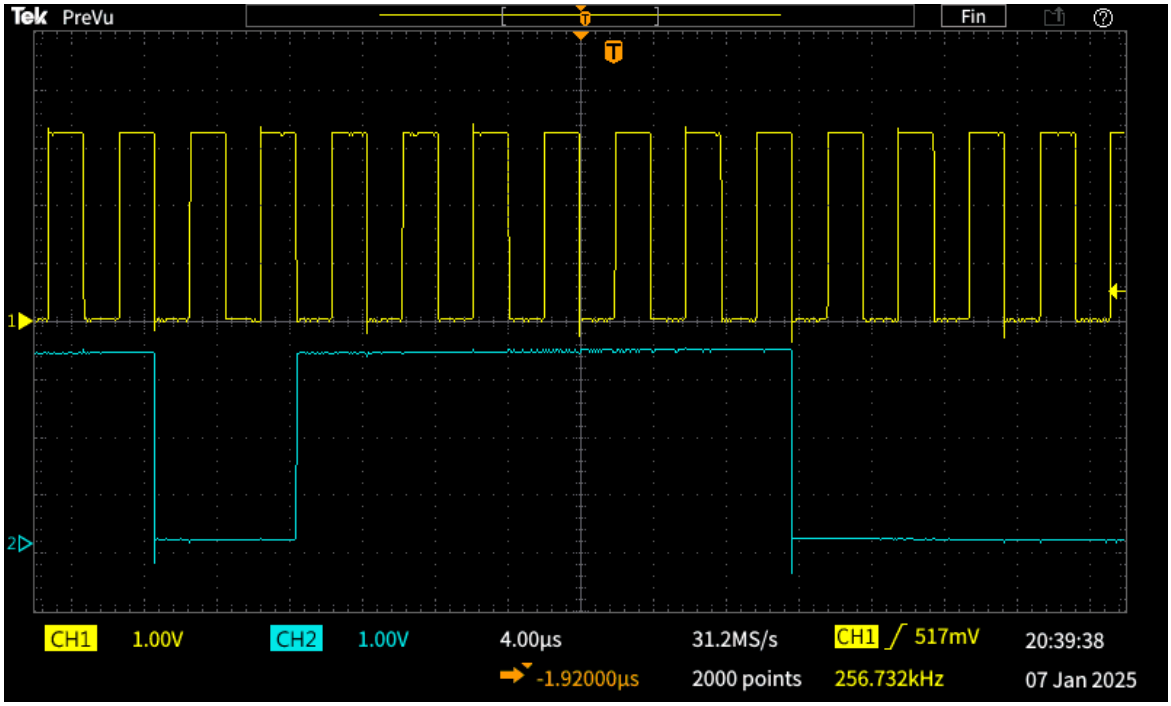
VIN (Voltage Input) : Tension d'entrée d'alimentation.

C'est quoi le bus **I²S** ?

I²S c'est un bus spécial, faite pour transmettre les données dans les appareilles audio.

Ligne d'horloge binaire :

Donc chaque bit de données correspondant à l'audio numérique possède une impulsion.



Clk Data

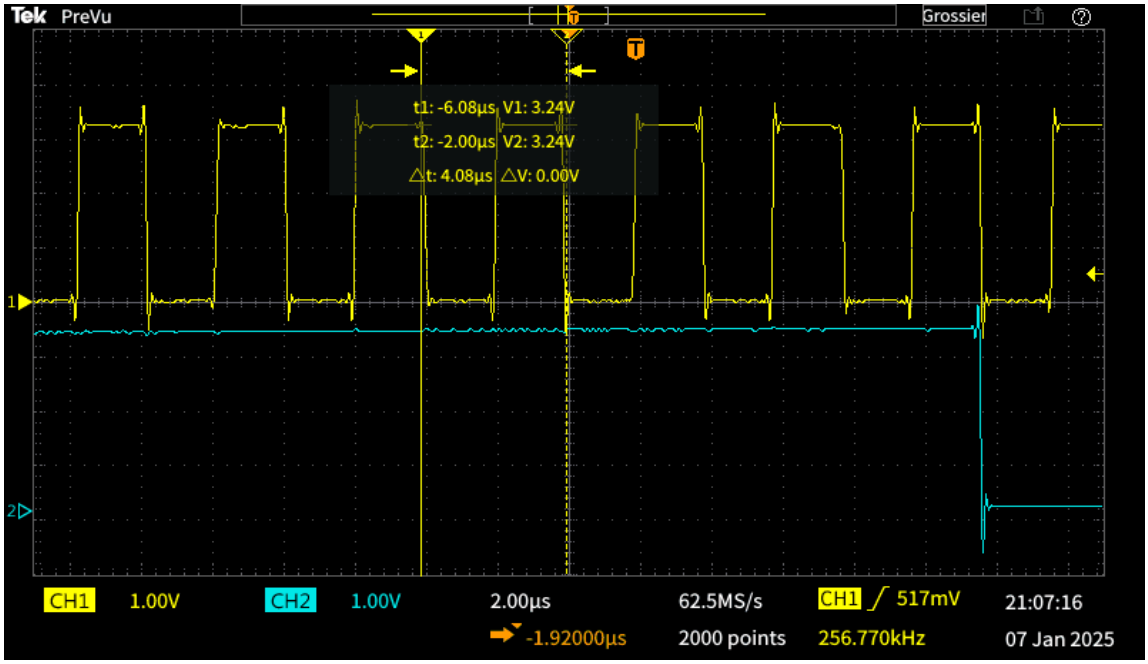
Durées d'1 bit est de 4µs

Vitesse est 250 kbit/s

Data

9 bit . 4µs = 36µs

Durées 1 donner de data est de 36 µs



Sur ce graphique on voit qu'un bit dure 4 µs

La vitesse est les mêmes, comme sur la photo d'avant

De Signal est à 250kbit/s

Le programme :

```ino
#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++;
}
```

Datasheet:

Electrical Characteristics

(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} = ∞, T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.) (Note 2)

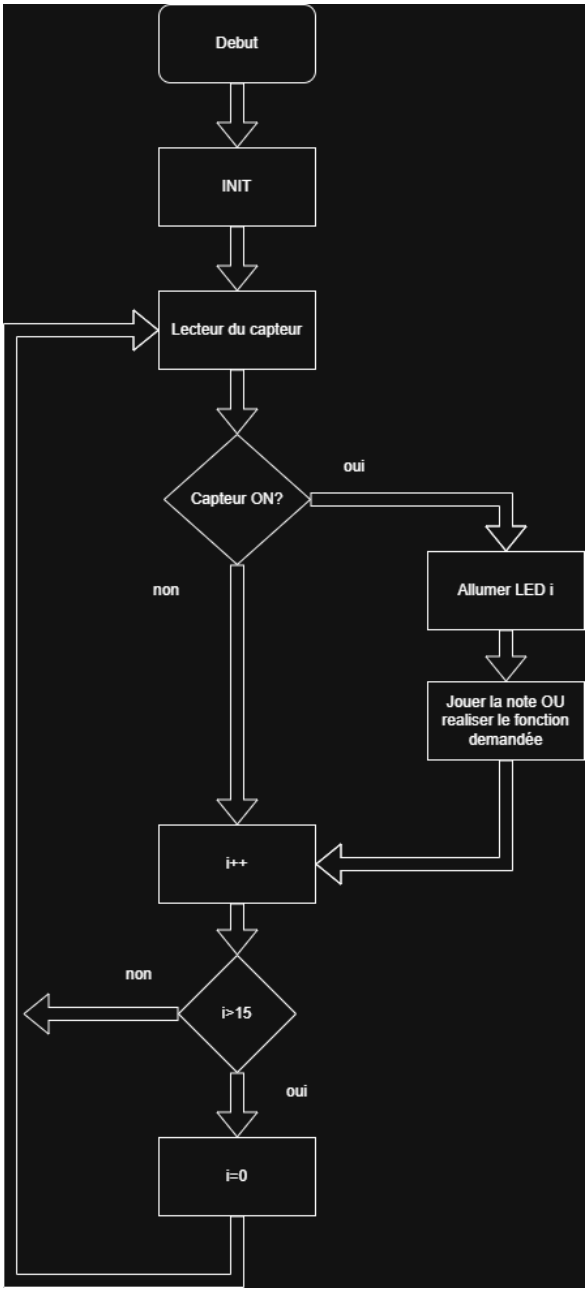
PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
Supply Voltage Range	V_{DD}	Guaranteed by PSSR test		2.5		5.5	V
Undervoltage Lockout	UVLO			1.5	1.8	2.3	V
Quiescent Current	I_{DD}	T_A = +25°C			2.75	3.35	mA
		T_A = +25°C, V_{DD} = 3.7V			2.4	2.85	
Shutdown Current	I_{SHDN}	$\overline{SD\_MODE}$ = 0V, T_A = +25°C			0.6	2	µA
Standby Current	I_{STNDBY}	$\overline{SD\_MODE}$ = 1.8V, no BCLK, T_A = +25°C			340	400	µA
Turn-On Time	t_{ON}				7	7.5	ms
Output Offset Voltage	V_{OS}	T_A = +25°C, gain = 15dB			±0.3	±2.5	mV
Click-and-Pop Level	K_{CP}	Peak voltage, T_A = +25°C, A-weighted, 32 samples per second (Note 3)	Into shutdown		-72		dBV
			Out of shutdown		-66		
Power-Supply Rejection Ratio	PSRR	V_{DD} = 2.5V to 5.5V, T_A = +25°C		60	75		dB
		T_A = +25°C (Notes 3, 4)	f = 217Hz, 200mV_{P-P} ripple		77		
			f = 10kHz, 200mV_{P-P} ripple		60		

Features

- Single-Supply Operation (2.5V to 5.5V)
- 3.2W Output Power into 4Ω at 5V
- 2.4mA Quiescent Current
- 92% Efficiency (R_L = 8Ω, P_{OUT} = 1W)
- 22.8µV_{RMS} Output Noise (A_V = 15dB)
- Low 0.013% THD+N at 1kHz
- No MCLK Required
- Sample Rates of 8kHz to 96kHz
- Supports Left, Right, or (Left/2 + Right/2) Output
- Sophisticated Edge Rate Control Enables Filterless Class D Outputs
- 77dB PSRR at 1kHz
- Low RF Susceptibility Rejects TDMA Noise from GSM Radios
- Extensive Click-and-Pop Reduction Circuitry
- Robust Short-Circuit and Thermal Protection
- Available in Space-Saving Packages: 1.345mm x 1.435mm WLP (0.4mm Pitch) and 3mm x 3mm TQFN
- Solution Size with Single Bypass Capacitor is 4.32mm²

Source: https://www.analog.com/media/en/technical-documentation/data-sheets/max98357a-max98357b.pdf

6) La programation

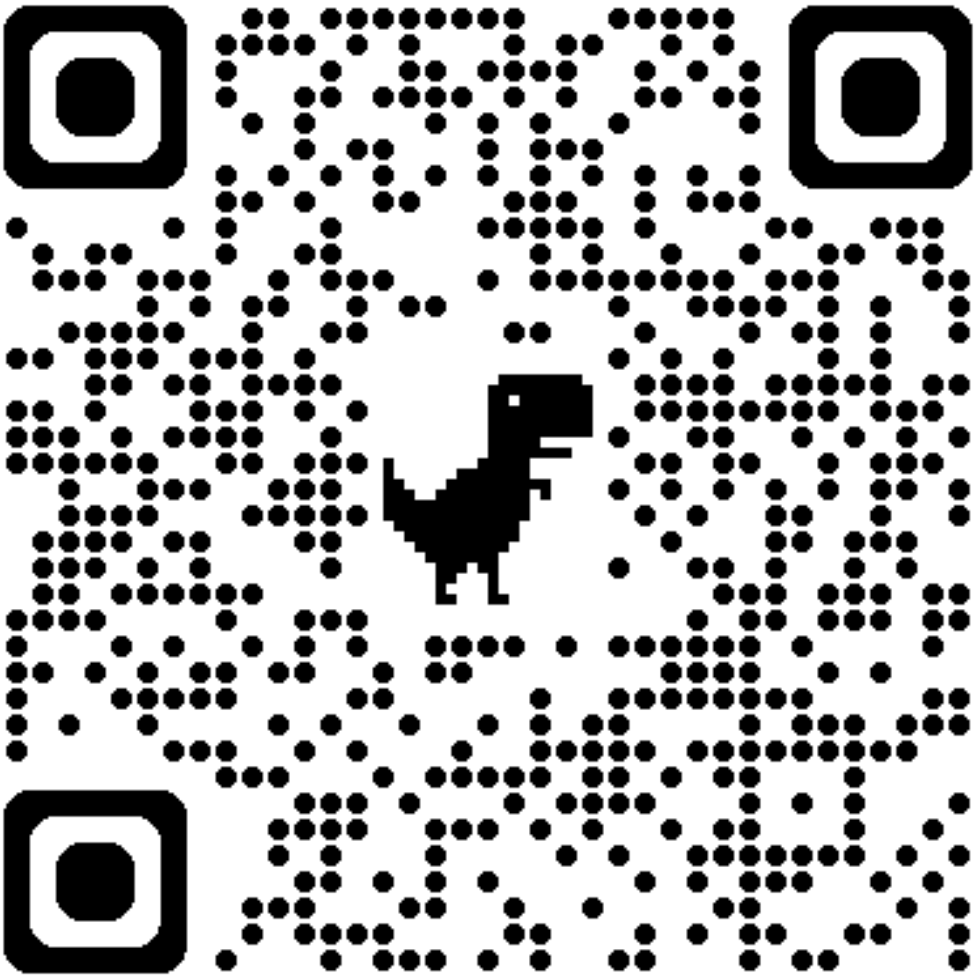


On commence par l'initialisation de tous les composants.
Tout d'abord, on vérifie si les capteurs infrarouges détectent quelque chose ou non.
Si oui, on allume la LED et on joue la note demandée (ou on suit la fonction), puis on revient à la lecture du capteur.

```ino
//Libreries
#include "qqq.h"
#include "PCF8575.h"
#include <Adafruit_NeoPixel.h>

//Constantes
#define NEOPIXELPIN 15
#define NUMPIXELS 48
#define NUMCAPTEUR 16

const float NOTES[] = {    //Les frequences des notes
  261.63, 293.66, 329.63,  // C4, D4, E4
  349.23, 392.00, 440.00,  // F4, G4, A4
  493.88, 523.25, 587.33,  // B4, C5, D5
  659.25, 698.46, 783.99,  // E5, F5, G5
  880.00, 987.77, 1046.50  // A5, B5, C6
};

PCF8575 PCF(0x20);
Adafruit_NeoPixel pixels(NUMPIXELS, NEOPIXELPIN, NEO_GRB + NEO_KHZ800);

void setup() {
  Serial.begin(115200);
  setupCustomI2S(1, 3, 19);  // CLK=1, LRC=3, DATA=19
  Wire.begin();
  PCF.begin();
  pixels.begin();
  pixels.setBrightness(100);
  Serial.println("System Ready");
}

void loop() {
  uint16_t capteurs = PCF.read16();

  // par chaque capteur on a la note+led
  for (int i = 0; i < NUMCAPTEUR; i++) {
    bool isActive = !(capteurs & (1 << i));
    uint32_t color = isActive ? pixels.Color(0, 150, 0) : pixels.Color(0, 0, 0);

    pixels.setPixelColor(i * 3, color);
    pixels.setPixelColor(i * 3 + 1, color);
```

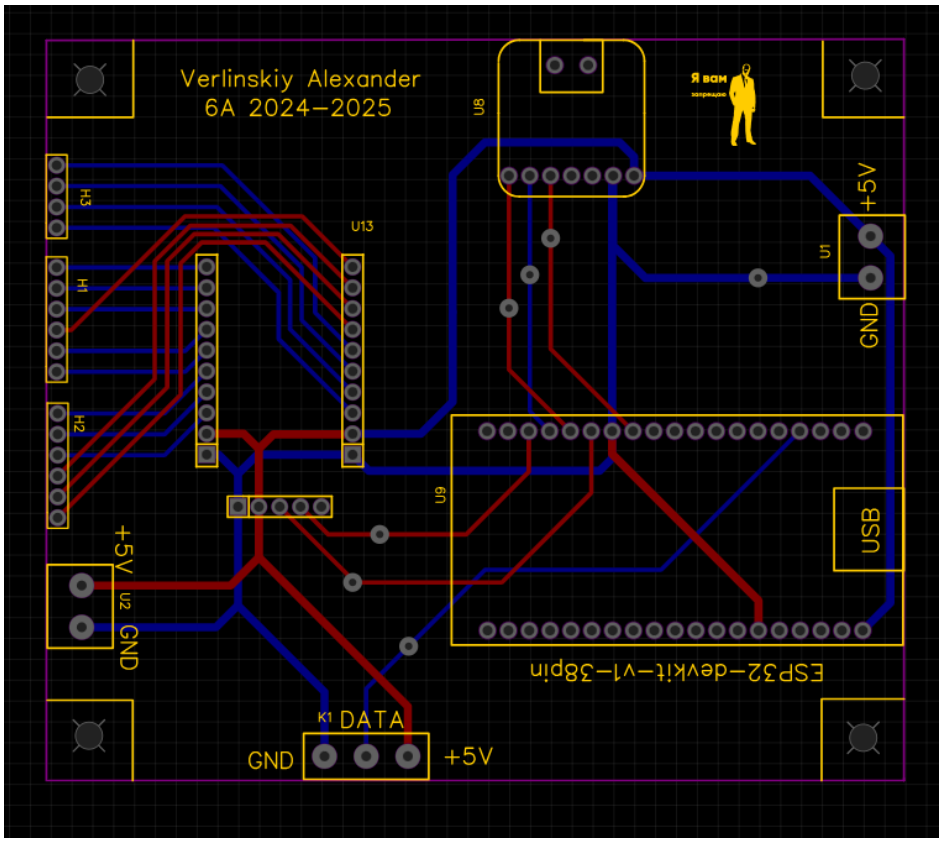
```ino
void loop() {
  uint16_t capteurs = PCF.read16();

  // par chaque capteur on a la note+led
  for (int i = 0; i < NUMCAPTEUR; i++) {
    bool isActive = !(capteurs & (1 << i));
    uint32_t color = isActive ? pixels.Color(0, 150, 0) : pixels.Color(0, 0, 0);

    pixels.setPixelColor(i * 3, color);
    pixels.setPixelColor(i * 3 + 1, color);
    pixels.setPixelColor(i * 3 + 2, color);

    if (isActive && !isPlaying) {
      playNote(NOTES[i], 250, 150);
    }
  }

  pixels.show();
  handleSound();
}
```

Le programme actuel fonctionne à moitié. Vous pouvez voir le code final sur mon GitHub.
https://github.com/AlexanderVerl/Table-Musicale/tree/main

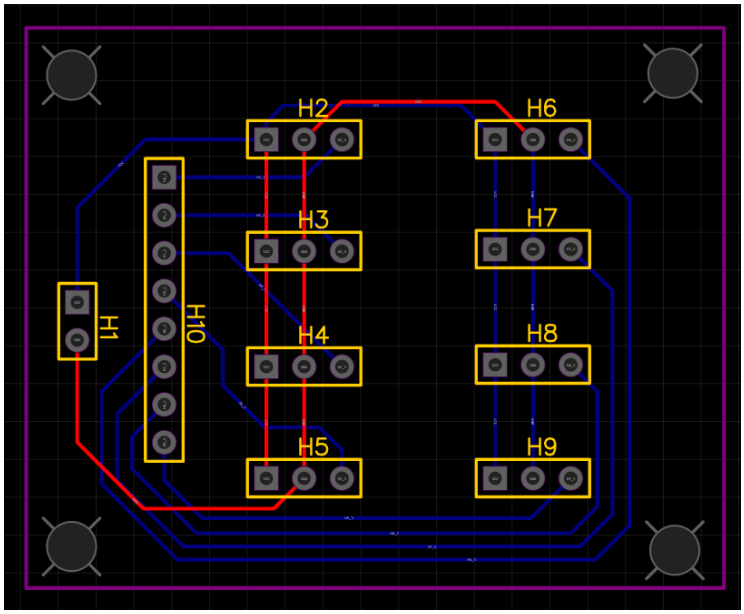


7) La fabrication

Le design du PCB a été réalisé par mes soins. Sur la photo ci-dessous, vous pouvez voir la version actuelle du PCB.

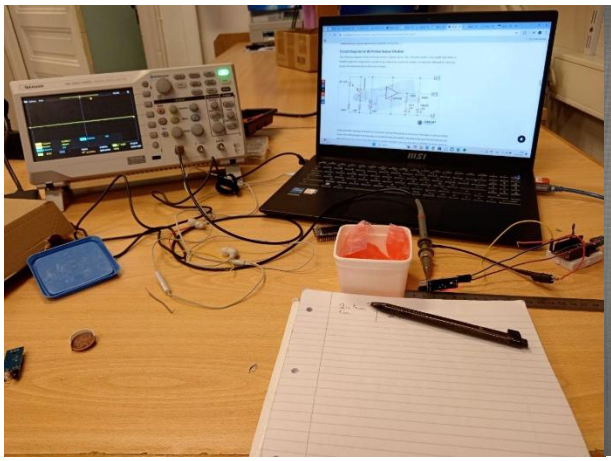


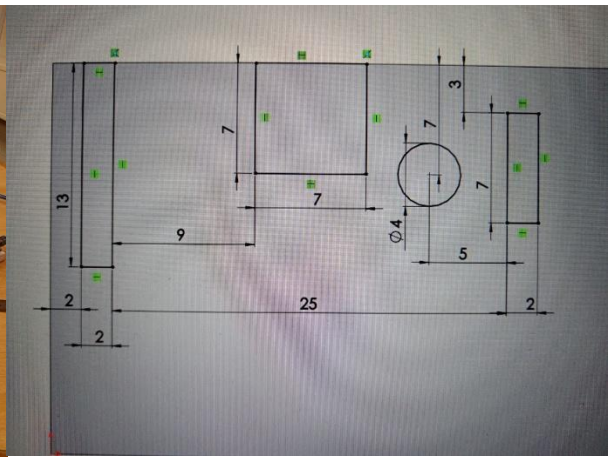
Après un certain temps, j'ai réalisé que j'avais un problème de câblage. Monsieur Kapita m'a proposé de concevoir un PCB plus petit, ce qui m'aidera à mieux gérer le câblage. Ce PCB a une taille vraiment réduite.

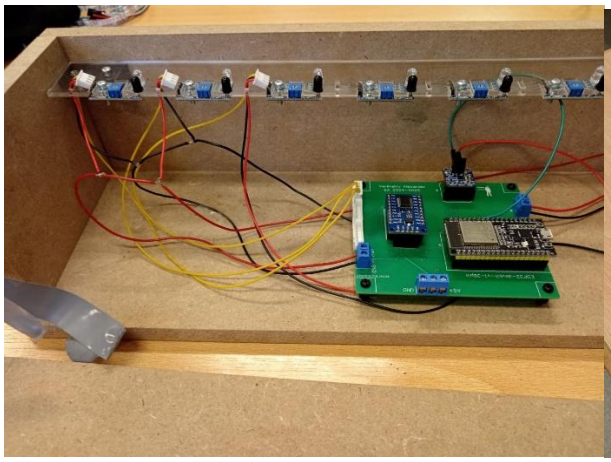


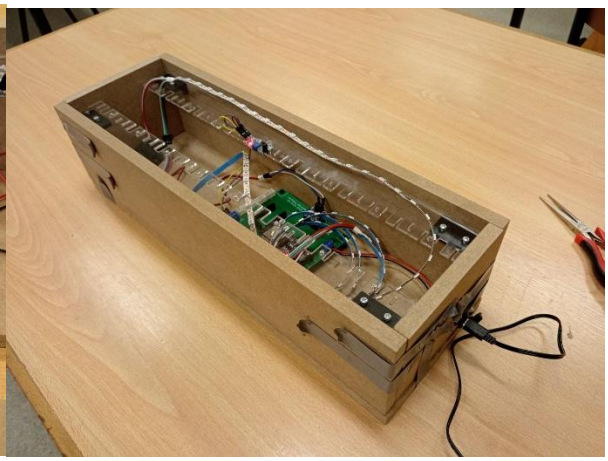
Sur les photos en sous vous pouvez voir comment avançait mon projet.

Le première 2 photo sont des photos de mesures et le 3D. Apres c’est l’assemblage.









8) La mise au point

Tout d'abord, j'ai rencontré un problème de câblage. La solution que j'ai apportée a été d'ajouter deux petits PCB, qui m'ont aidé à connecter correctement tous les capteurs.
J'ai également eu un problème de programmation : le son et la LED réagissaient avec du retard quand je posais le doigt.
Pour résoudre cela, j'ai supprimé la fonction qui appelait le NeoPixel à chaque fois ; je l'ai allumé une seule fois, et le son ainsi que la LED fonctionnaient correctement, en temps réel, c'est-à-dire sans retard.

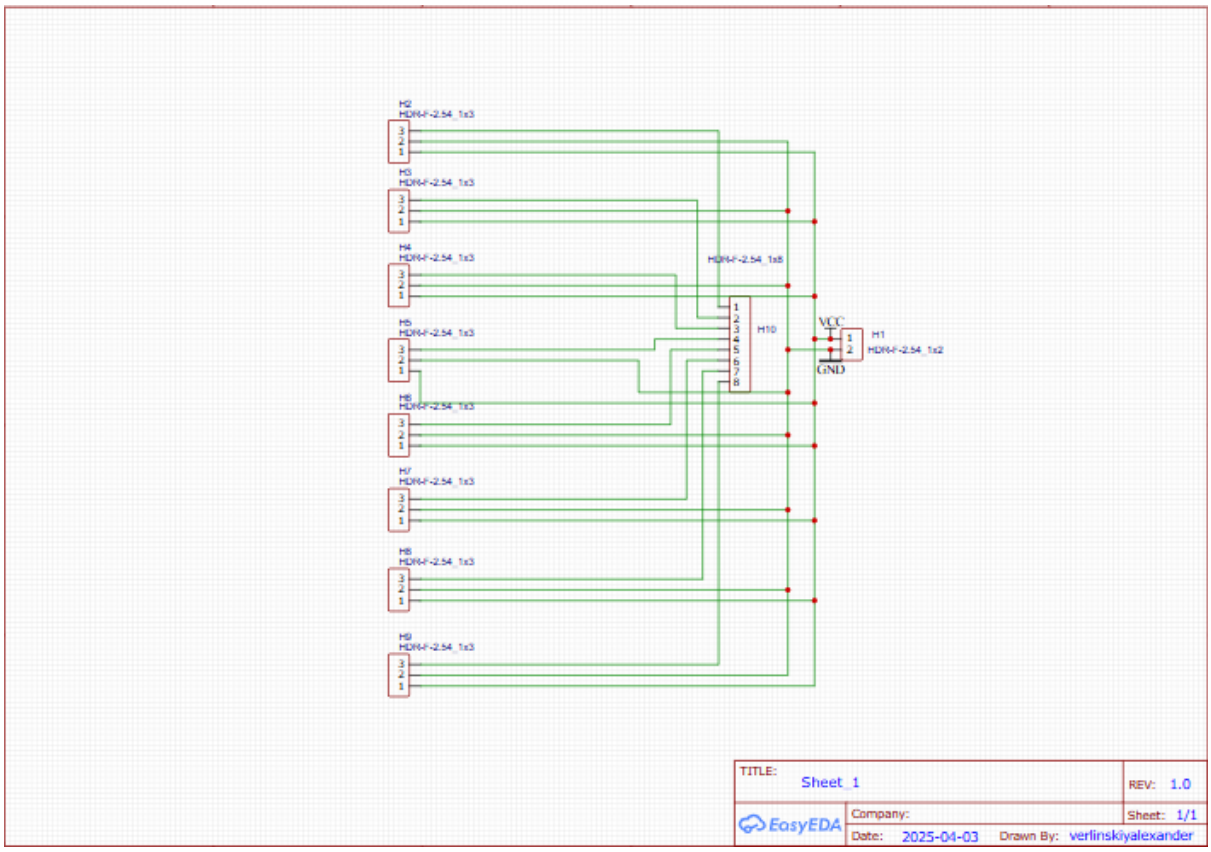
9) La conclusion

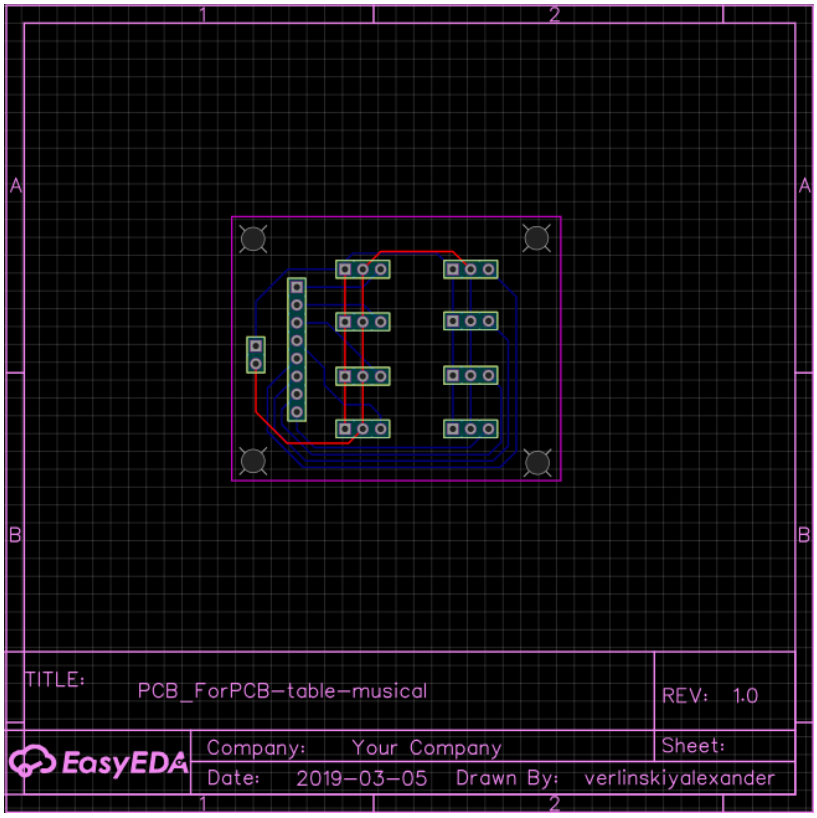
J’ai beaucoup apprécié travailler sur ce projet, que j’ai développé entièrement par moi-même, de A à Z.
Cette expérience m’a permis d’approfondir mes connaissances techniques, notamment sur les bus de communication entre composants, la gestion des capteurs infrarouges et l’utilisation des NeoPixels.
J’ai aussi rencontré des défis concrets, comme des problèmes de câblage et de synchronisation du son avec la LED, que j’ai réussi à résoudre grâce à une réflexion méthodique et des ajustements du code.
À ce stade, le projet est presque terminé ; il reste quelques améliorations à apporter au code pour finaliser l’ensemble.
Ce travail m’a non seulement permis d’appliquer mes compétences théoriques, mais aussi de développer mon autonomie et ma capacité à résoudre des problèmes pratiques.
Malheureusement, j’ai mal géré le temps, et du coup j’ai dû fournir un effort supplémentaire, mais j’ai fait cet effort et terminé mon projet.

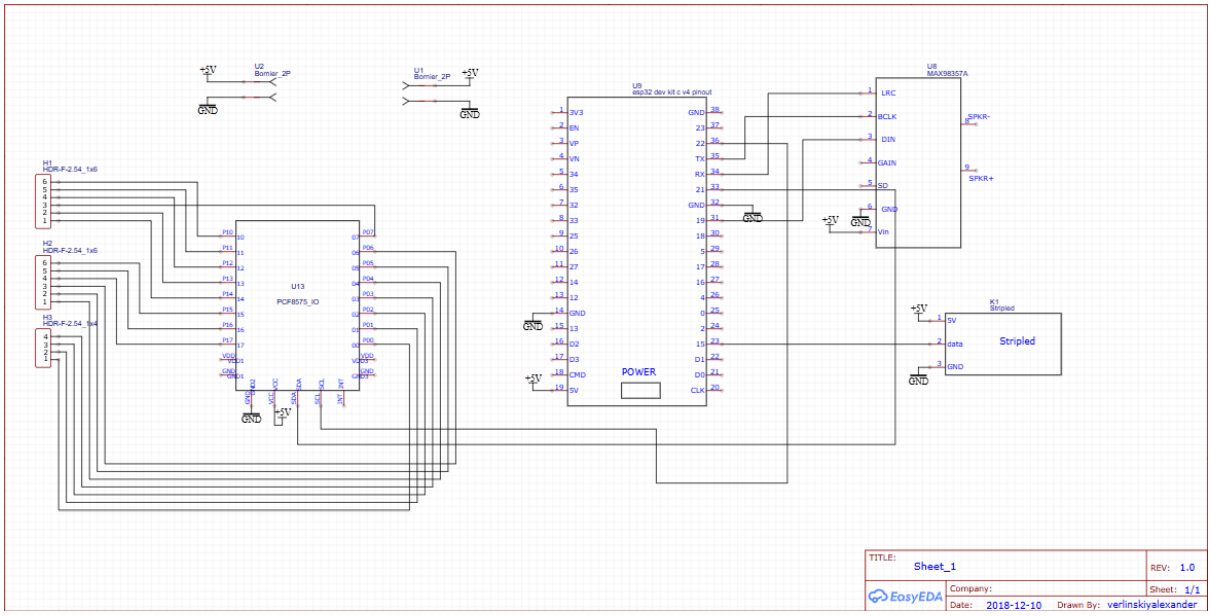
10) La bibliographie

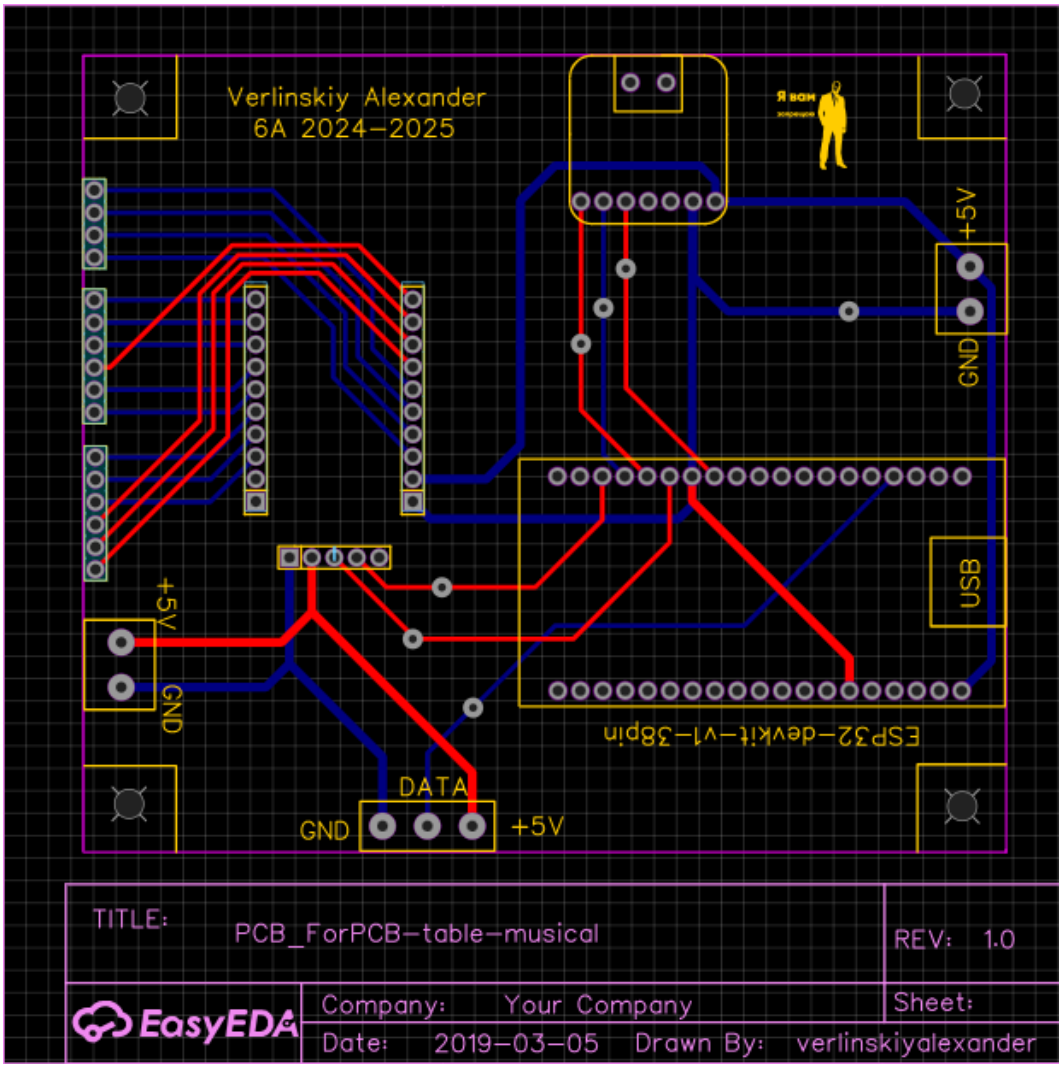
https://wp.josh.com/2014/05/13/ws2812-neopixels-are-not-so-finicky-once-you-get-to-know-them/ (8 /02/2025)
Cours de Monsieur Pochet (24/05/2025)
Cours de Monsieur Slagmolen (24/05/2025)
https://circuitdigest.com/microcontroller-projects/interfacing-ir-sensor-module-with-arduino (15/12/2024)
https://diyi0t.com/i2s-sound-tutorial-for-esp32/ (13/03/2025)
https://www.donskytech.com/esp32-pcf8575-port-expander/ (20/03/2025)
https://www.donskytech.com/interface-esp32-with-infrared-sensor/ (18/02/2025)
Assistance technique
J'ai eu recours à des outils d'intelligence artificielle pour la correction orthographique et l'optimisation de certaines sections du code, tout en garantissant une compréhension approfondie des solutions implémentées.

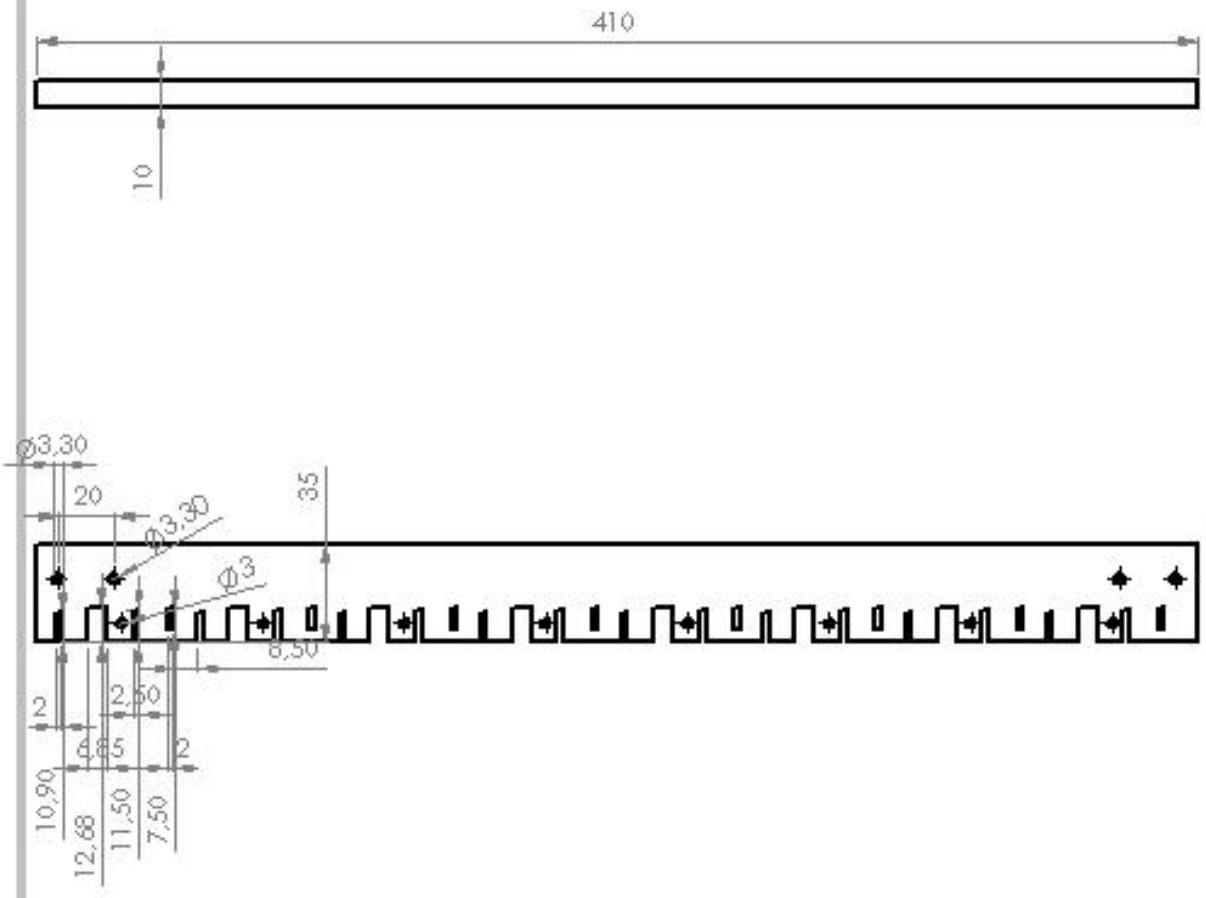
11) Les annexes



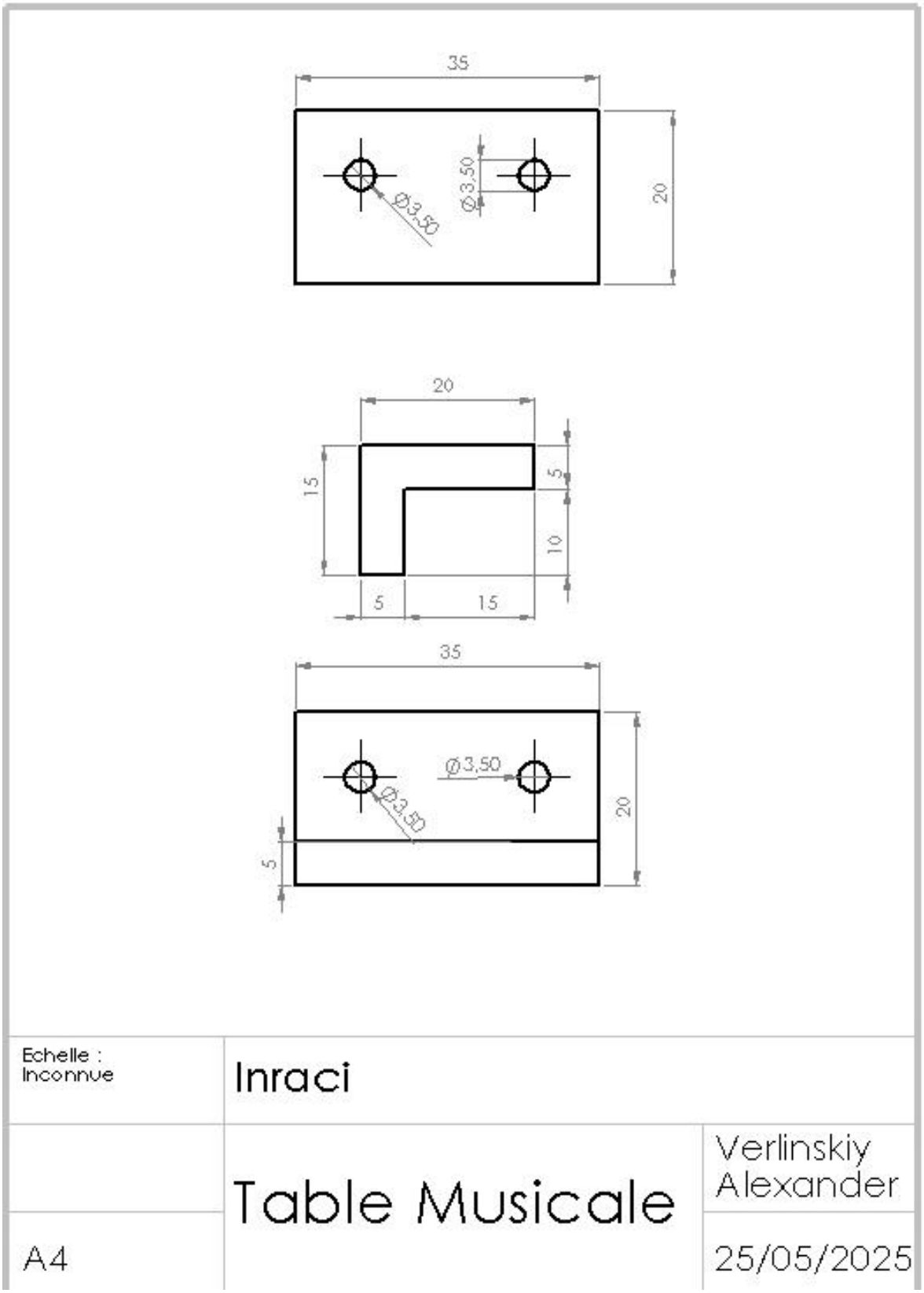



U2
Bornier_2P
U1
Bornier_2P
U9
esp32 dev kit c v4 pinout
U8
MAX98357A
H1
HDR-F-2.54_1x6
H2
HDR-F-2.54_1x6
H3
HDR-F-2.54_1x4
U13
PCF8575_IO
POWER
K1
Stripled
LRC
BCLK
DIN
GAIN
SD
Vin
SPKR-
SPKR+
data
TITLE:
Sheet_1
REV: 1.0
EasyEDA
Company:
Sheet: 1/1
Date: 2018-12-10
Drawn By: verlinskiyalexander


Verlinskiy Alexander
6A 2024-2025
Я вам
+5V
GND
USB
+5V
GND
ESP32-devkit-v1-38pin
DATA
GND
+5V
TITLE:
PCB_ForPCB-table-musical
REV: 1.0
EasyEDA
Company: Your Company
Sheet:
Date: 2019-03-05
Drawn By: verlinskiyalexander

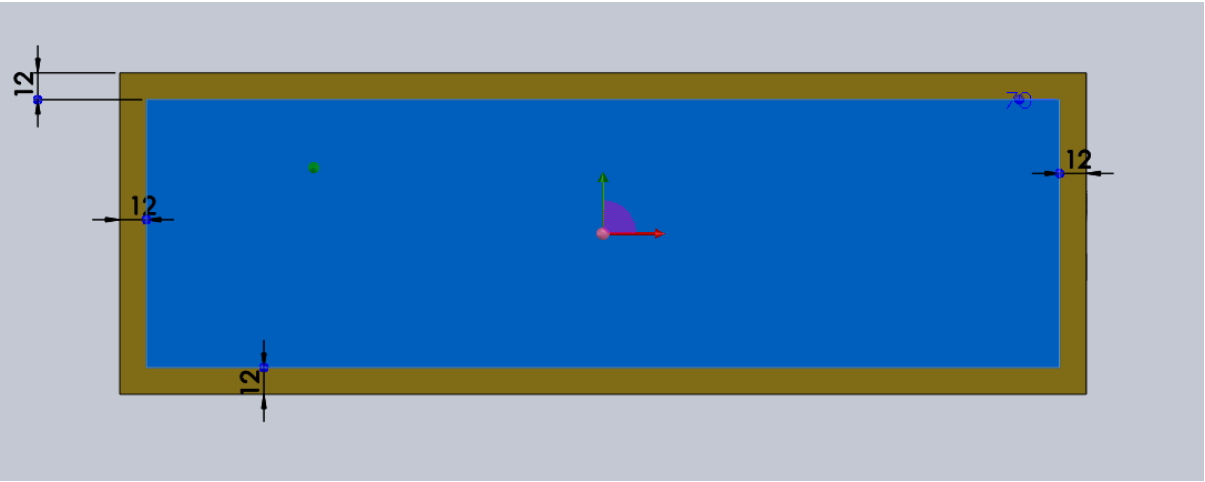


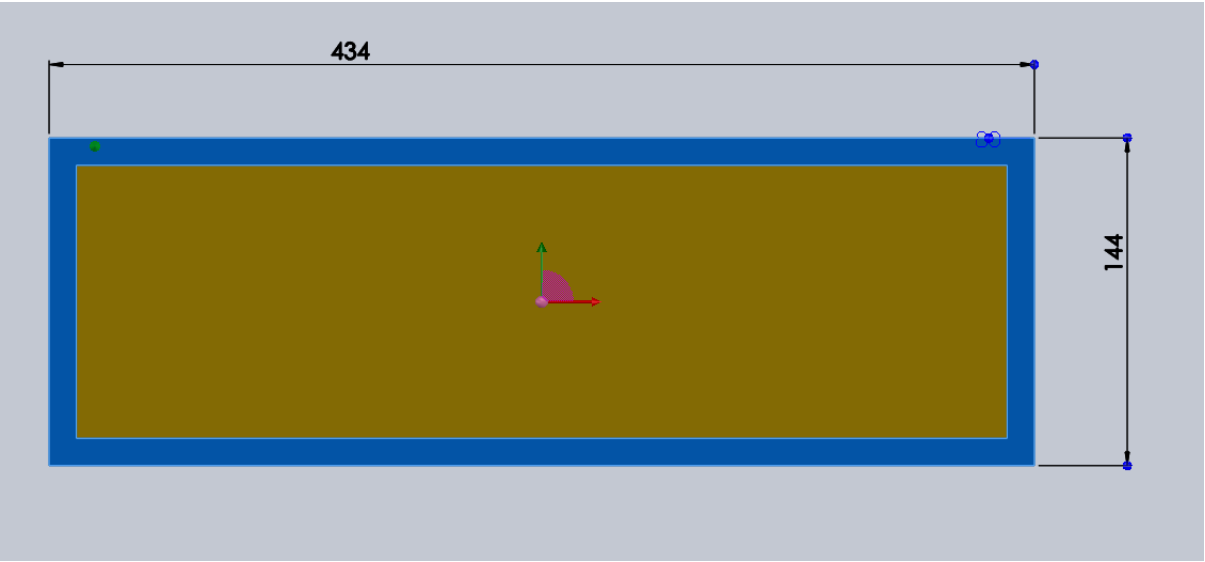
Echelle : Inconnue	Inraci	
	Table Musicale	Verlinskiy Alexander
A4		25/05/2025



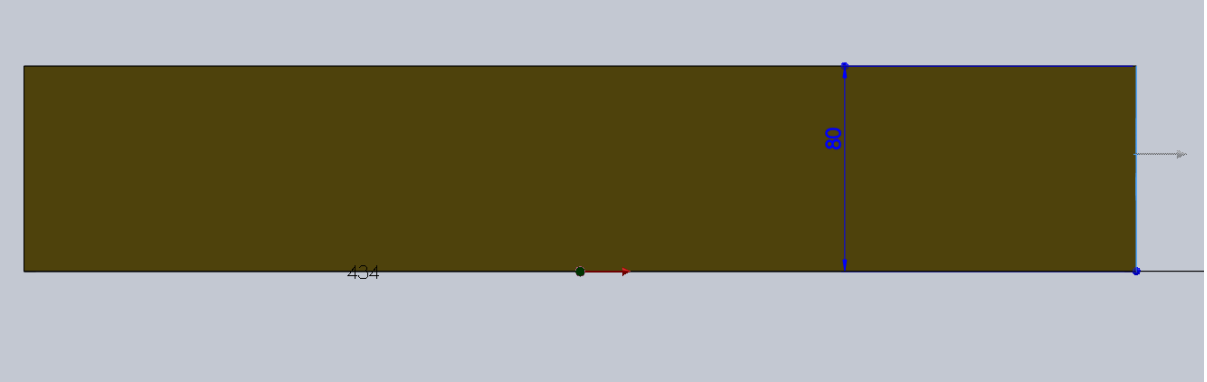
Echelle : Inconnue	Inraci	
	Table Musicale	Verlinskiy Alexander
A4		25/05/2025

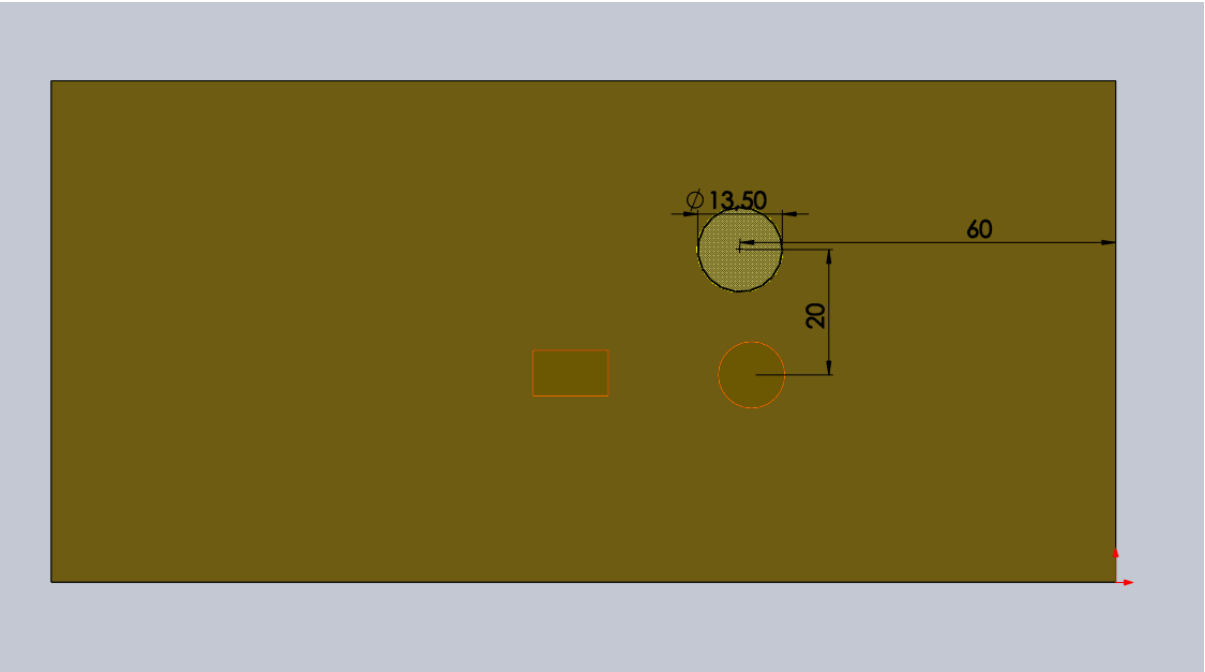
Ici vous pouvez voir la taille de ma boite avec tous les trous sur ma boite.

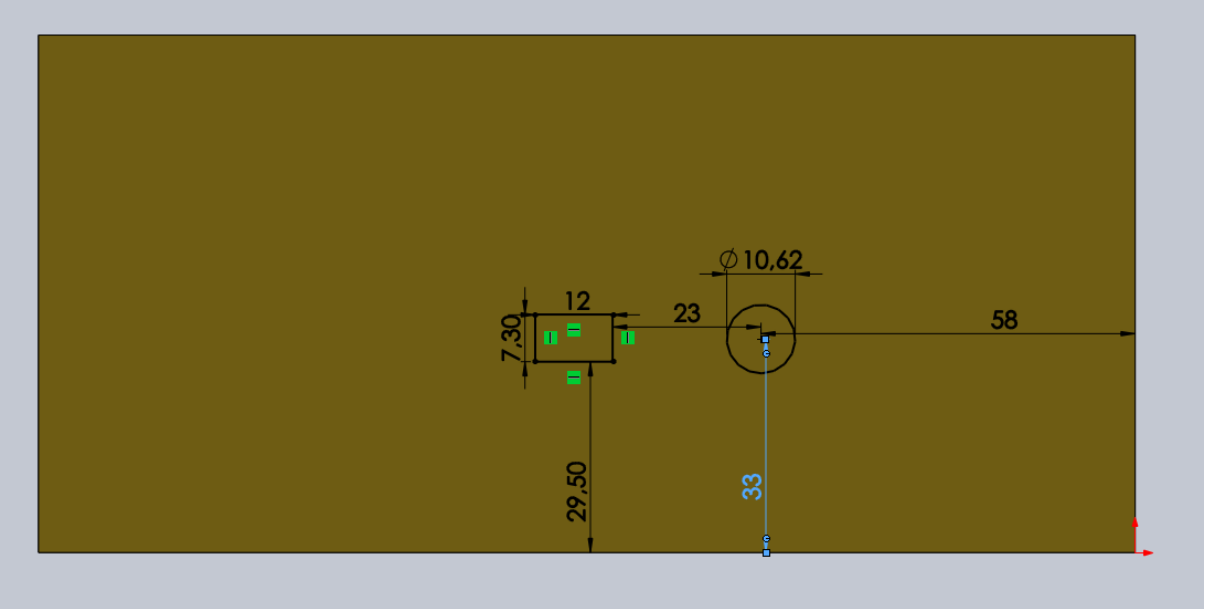

12
12
12
12

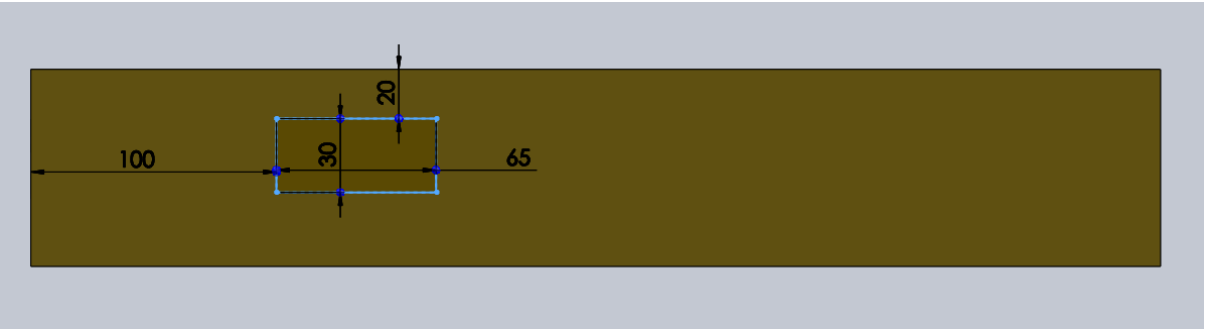

434
144

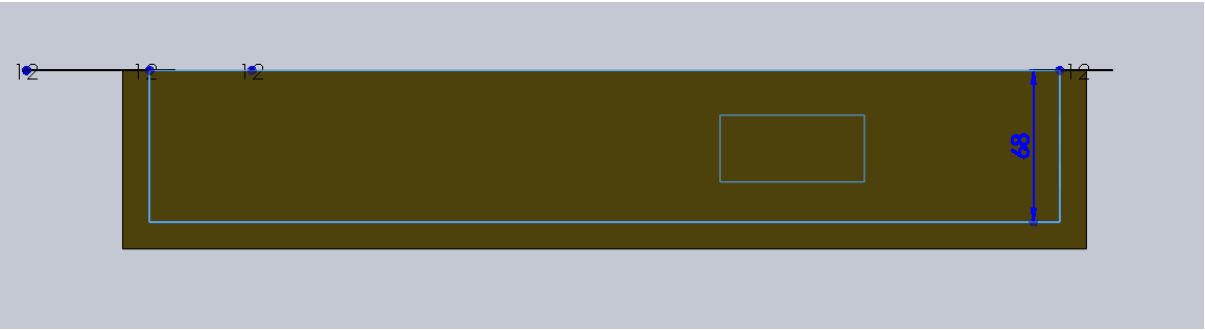

410 ±0,50
120 ±0,50


80
434


∅13,50
60
20


∅10,62
12
23
58
7,30
29,50
33


20
100
30
65


12
12
12
12
48