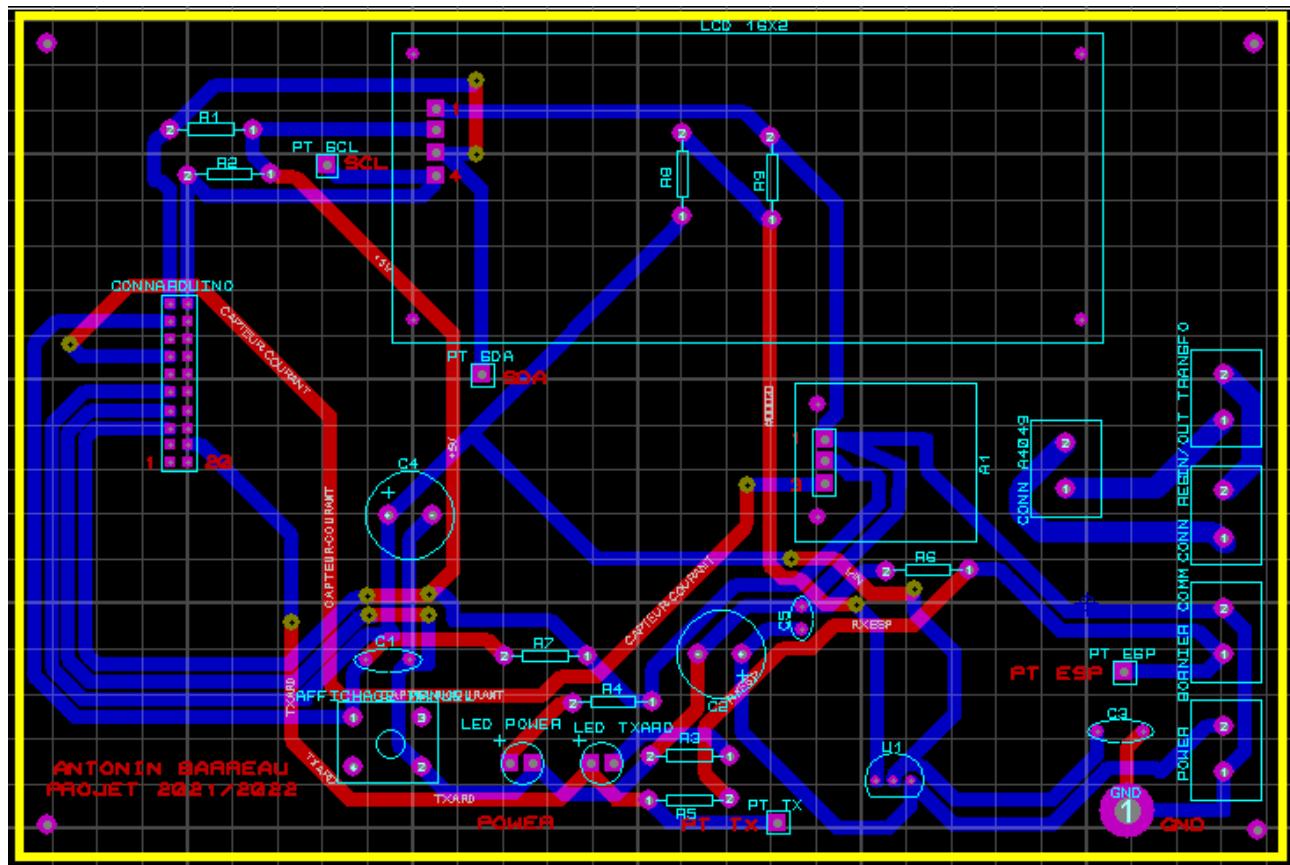


# Revue de projet



Lycée Pierre de Coubertin  
Année 2022-2023



## Liens du projet complet sur Github

☰ README.md

Projet 2023 Captor ASC724 Language C++ Version V1 MADE WITH C++

### Projet bts

### Contenu github

- [Info General](#)
- [Contenue](#)
- [Technologies](#)
- [Schéma structurelle](#)
- [Carte](#)
- [Setup](#)

### Info General

Ce projet consiste à capter le courant et une fois fais le renvoyer sur une passerelle MQTT envoyer en RS232.

### Contenue

Ce github contient:

*Certaines images sont tirées de mes dossiers sur github. L'intégralité des codes, fichiers 3D et mesures sont sur le github et accessibles depuis le QR CODE. Des logos github sont disposés avec l'emplacement du détail github.*



## Sommaires :

### Projet globale :

- *Présentation projet globale* ..... *page 4*
- *Commanditaire du projet* ..... *page 5*

### Sous projet :

- *Présentation sous projet* ..... *page 6*
- *Les composants utilisé* ..... *page 7*
- *A4049* ..... *page 8*
- *Étude des composants* ..... *page 9*
- *La mise en œuvres du projet* ..... *page 10*  
*(phase de test et problème rencontré, mesures)*

### Conception :

- *Schéma structurelle* ..... *page 11*
- *Conception de la carte à l'aide de logiciel assisté* ..... *page 14*
- *Étude du programme* ..... *page 15*
- *Boîtier 3D* ..... *page 20*

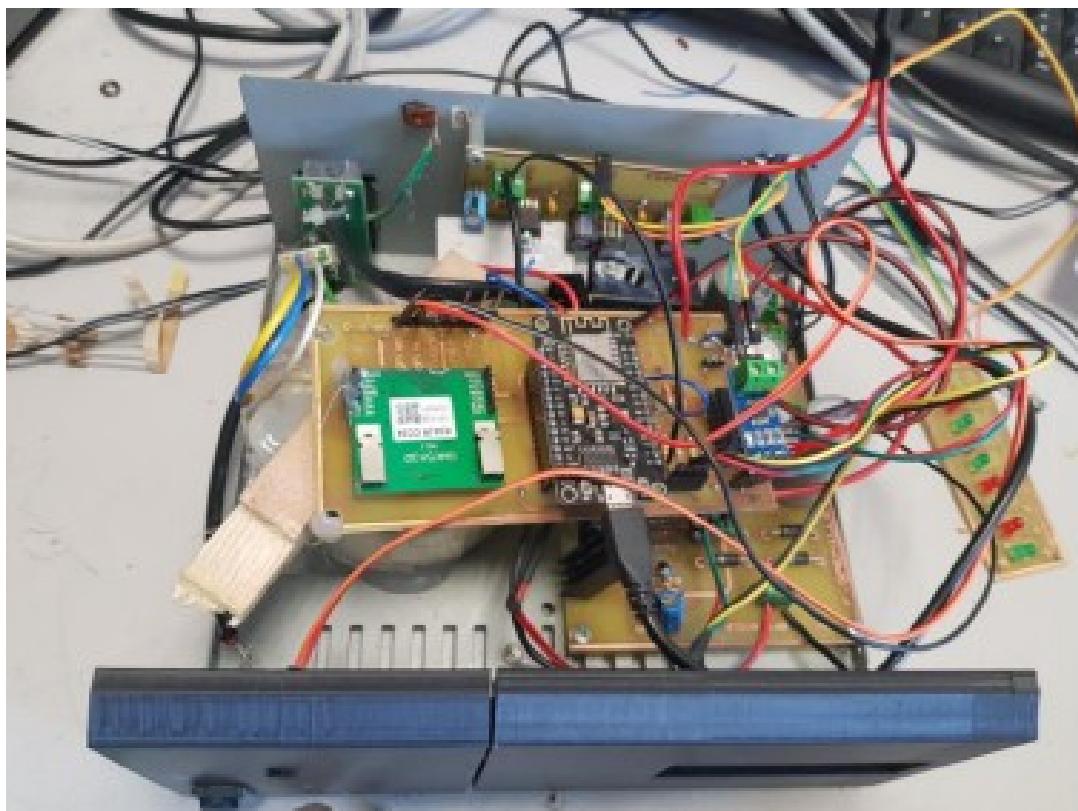
## Projet Globale :

Il faudra réaliser un prototype avec les solutions qui nous seront imposées, ou chercher les plus appropriées (*micro-contrôleur ex : arduino, esp, raspberry*), avec différents capteurs, et ensuite envoyer sur un serveur MQTT réalisé par le BUT.

Le projet est découpé en plusieurs sous-parties :

- *Aide au dimensionnement des dissipateurs thermiques*
- *Aide au dimensionnement du transformateur d'alimentation (mon projet)*

Toutes les cartes seront alimentées par une carte d'alimentation qui fourni du +27V (environ), fournie par l'établissement et qui régule la tension selon le contrôleur et la carte.



## Commanditaire :

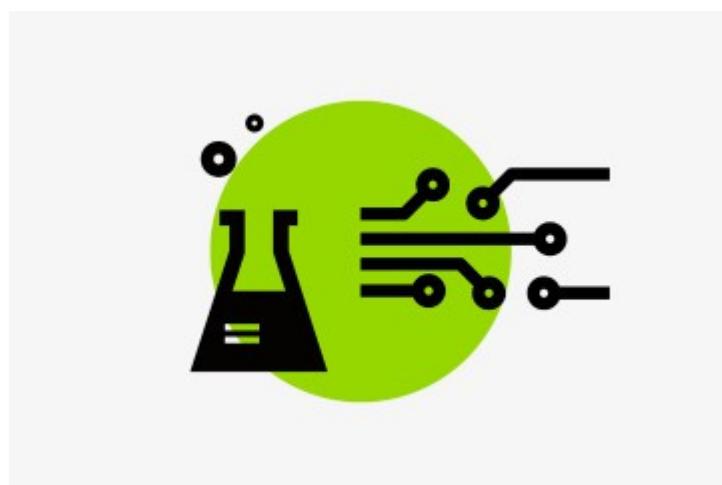
Pour ce projet, le lycée Pierre de Coubertin est en partenariat avec la formation GEII « **Génie électrique et informatique industrielle** ». Monsieur François Gremillet, professeur d'électronique dans le lycée Pierre de Coubertin et qui intervient dans la formation **GEII**.

Un partenariat a vu le jour suite à la présentation de son projet « station audio » qui met en œuvre le fonctionnement de nombreux composants électroniques étudiés dans la formation.

### **La formation :**

Le but GEII est formation de niveau BAC+3 (licence). Cette formation est sur 3 ans avec :

- 600h de projet.
- 30h-33h de cours /semaine
- 22 à 26 semaines de stage.
- Travaux de groupe.

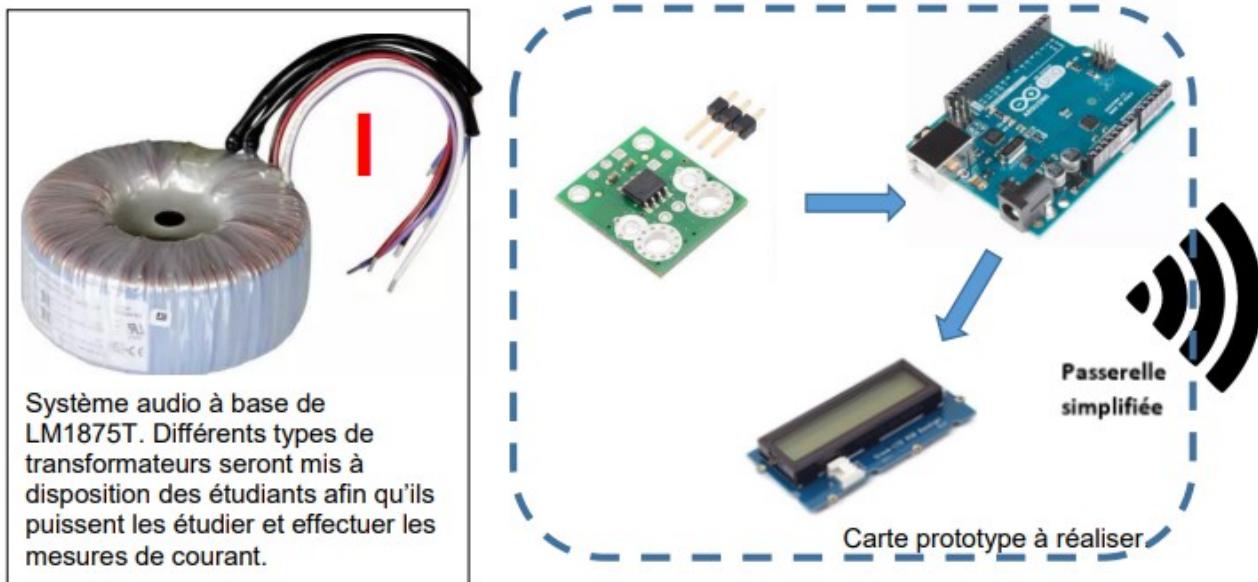


## Présentation de mon sous-projet :

**Extrait cahier des charges :** « *Création d'une carte de mesure de courant sur transformateur avec envoie des données sur le serveur MQTT.* »

En solution imposé :

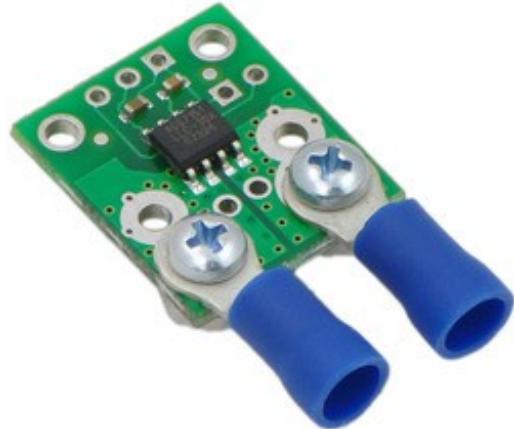
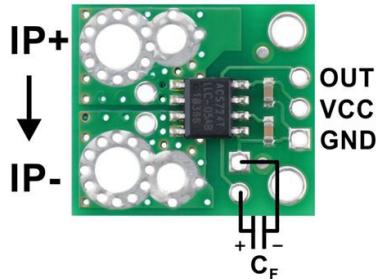
Micro-contrôleur	Arduino
Capteur effet hall	ACS724 (A4049)
Écran	Afficheur I <sup>2</sup> C « Quapass UI »



**But du projet :** Aide au dimensionnement d'un transformateur à l'aide du capteur de courant à effet hall ASC724.

## Les composants utilisé :

<b>Micro-contrôleur</b>	Arduino	Sert pour envoyer le code, et alimente une partie du projet . Il envoi les information à la passerelle MQTT.	
<b>Ecran I2C</b>	Quapass U1	Affiche la tension et le courant.	
<b>Capteur de courant</b>	ASC724 (A4049)	Capte le courant entrant et il envoie une tension sur le pin analogique de l'arduino et le convertira pour avoir le courant.	
<b>Régulateur tension</b>	LM317	Régule le courant entrant (+30V) en +7.40V pour l'arduino	
<b>Résistances et condensateurs</b>			
<b>Bouton</b>		Pour activer la fonction « Prise de courant automatique »	
<b>Borniers (24A)</b>	CTB0700 /2		



## A4049 :

La carte A4049 est un capteur de courant à effet hall. Il existe en plusieurs versions selon le modèle. Il encaisse plus ou moins d'Ampérage (le modèle utilisé est un **+50A/-50A**).

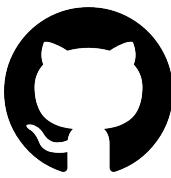
Tension d'entrée	+4,5V / +5V
Sensibilité	40mV / A
Tension de sortie	+2,5V
Erreur typique	$\pm 0,8 \%$
Version	Bidirectionnelle
Plage de température	-40°C à 150°C
Courant	De -50A/+50A

La carte est une « carte porteuse » ou « carte porteuse » pour le capteur ACS724. La carte fonctionne en AC/DC.

Pour connaître le courant envoyé par le A4049, le calcul : **P/U**

Pour le courant : $(X - 2,5) / (0,40)$
--

\*X : Le voltage reçu sur le pin analogique.

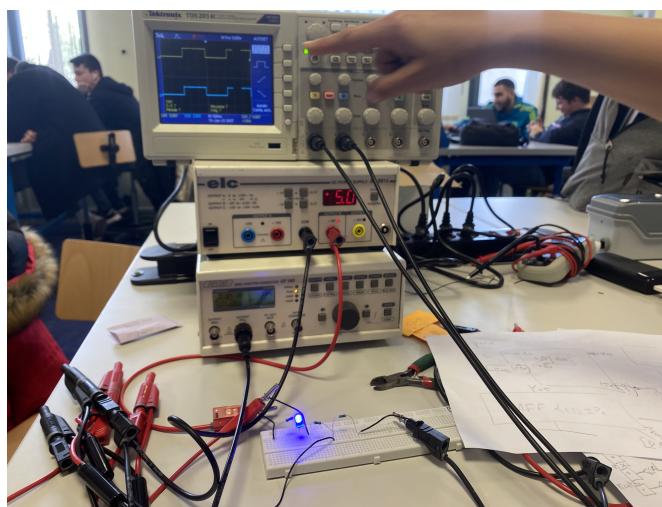
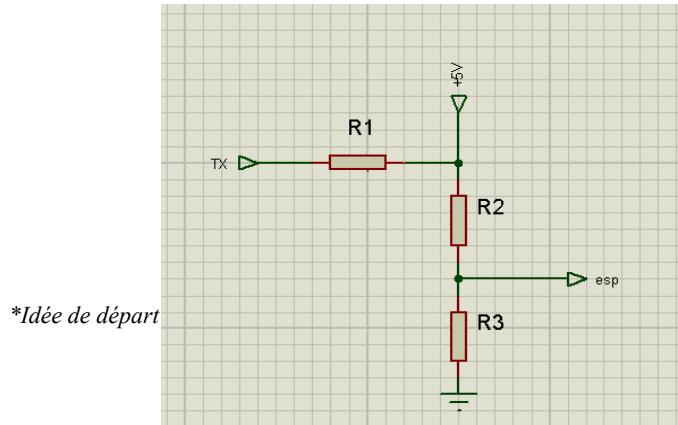
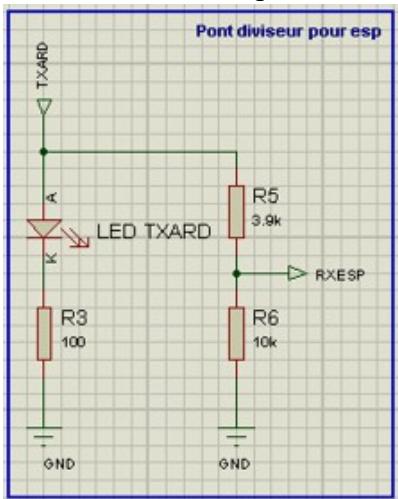


## La mise en œuvres du projet

Mon projet est de mettre en œuvre mon capteur A4049 avec un arduino (solution imposée) comme micro-contrôleur. Je dois ensuite faire une mesure de courant avec celui-ci, le renvoyer sur un afficheur I<sup>2</sup>C, et le renvoyer en RS232.

Les données seront envoyées en RS232. La tension devra être adaptée car la passerelle est gérée par un esp (qui est un équivalent de l'arduino, en version réduite), qui prend du 3.3V. Hors, l'arduino lui envoie du 5V et il y a un risque pour la passerelle.

J'ai donc réalisé un pont diviseur et j'ai rencontré un problème avec celui-ci lors de mes tests



Le problème est que la résistance qui protège la led faisait diminuer la tension avant le pont diviseur qui abaissait la tension bien dessous des 2V.

## Mesures :

- courant(A) x sensibilité(40mV)

Tension	Mesure Capteur(V)	Mesure Capteur(A)	Calcul
+0V	+2,50V	+0A	+0V
+5V	+2,50V	+0.07A	+0.028V
+9V	+2,51V	+0.25A	+0.1V
+15V	+2,51V	+0.31A	+0.124V

- Regulateur

Pins	Tension
ADJ(1)	+6.22V
VO(2)	+7.50V
VIN(3)	+15V (change selon l'alim)

- Mesure Transmission:

	Tension
TX	+4,70V
ESP	+3.37V
SCL	+4.99V
SDA	+4.99V

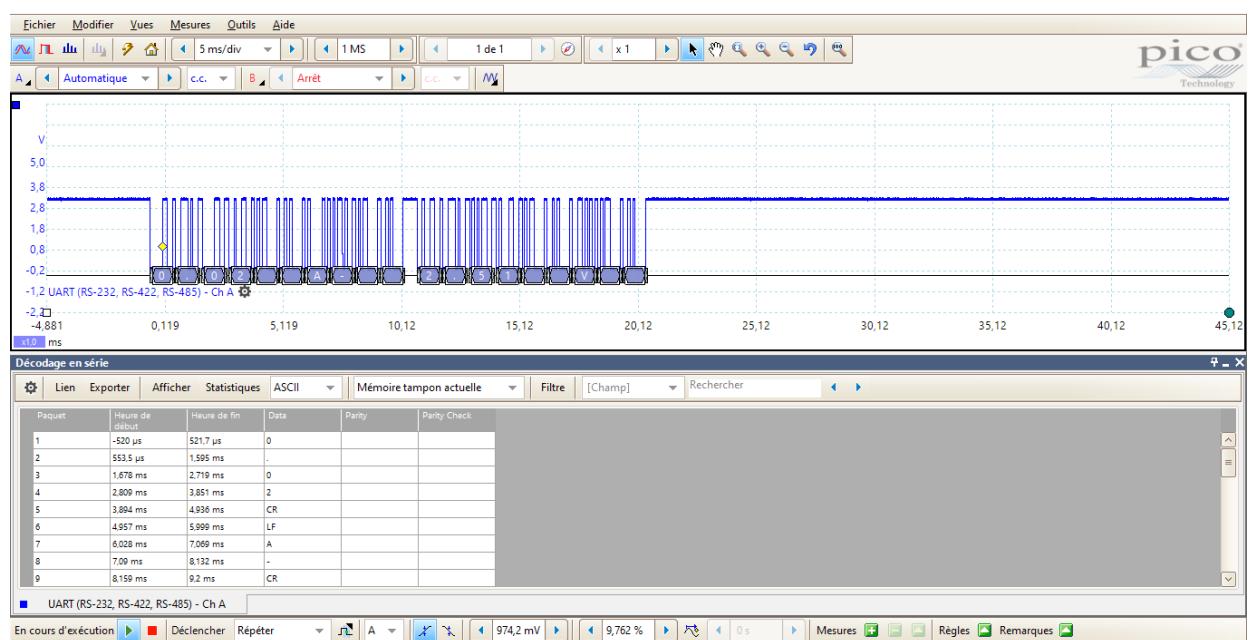
- Capteur:

	Tension
Vcc	+5V
Out	+2.5V voir +2.49V

- Mesure:

	Tension
Vin-ard	+7,47V
+5 ard	+5V
Alim-led	+2.10V
led-led	+4.70V

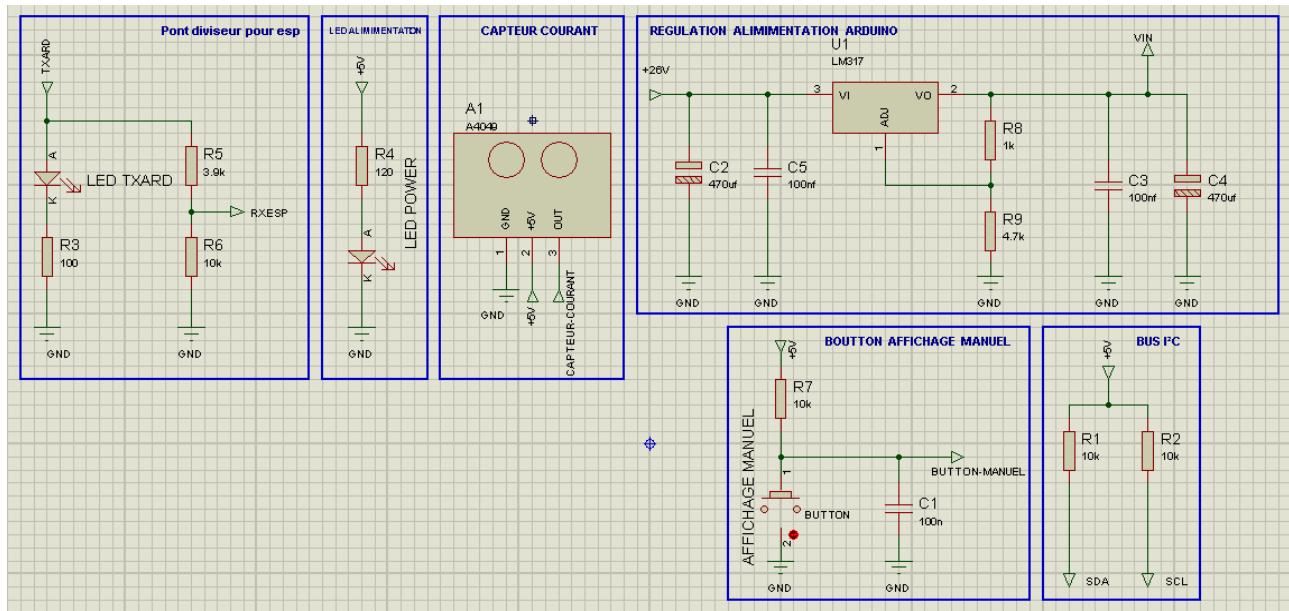
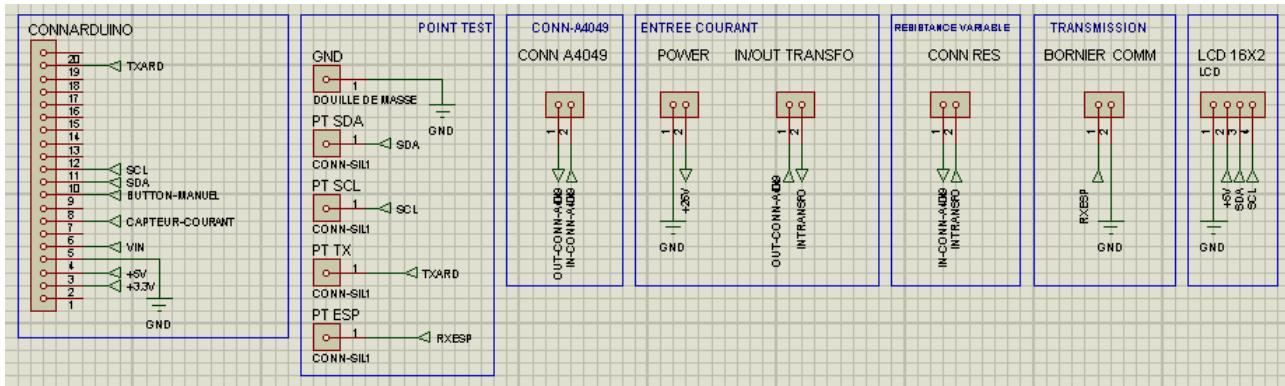
\*Les mesures sont relevé de la carte

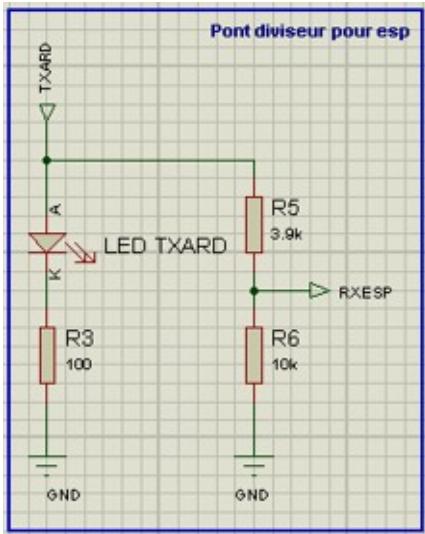


Relevé En Rs232 (avec les informations relevé par la carte est envoyé au projet passerelle)

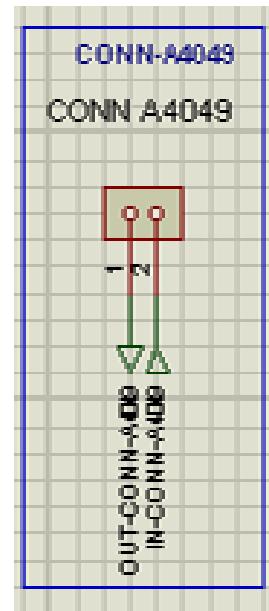
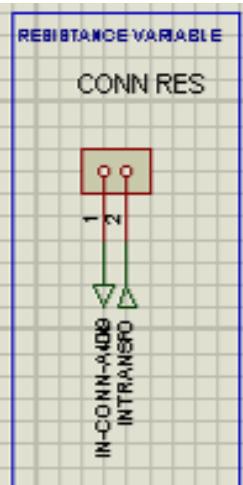
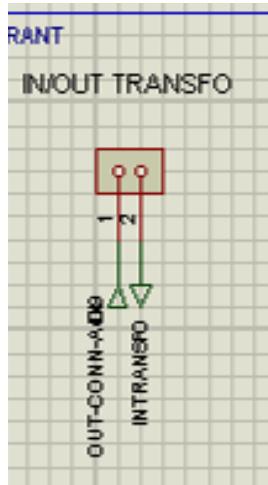
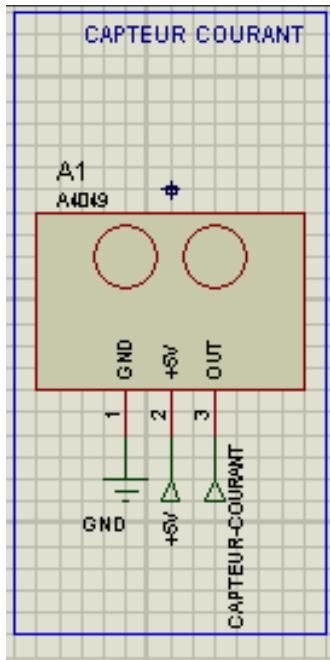
# Conception :

## Schéma structurelle :





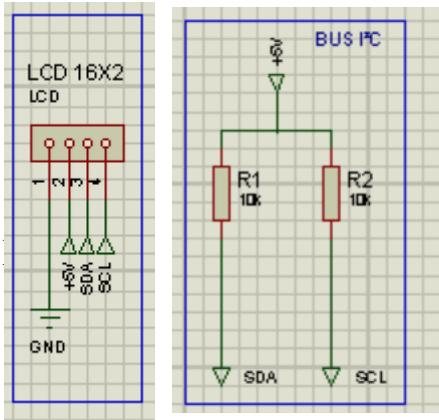
Le pont diviseur pour adapter pour la passerelle



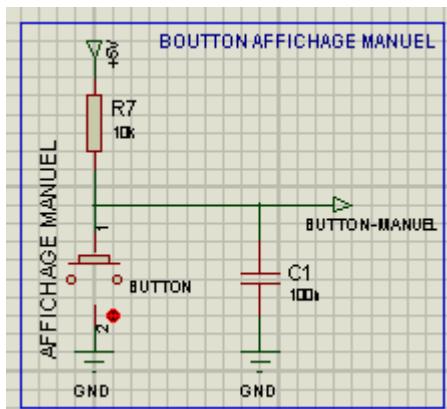
Cette partie est pour le capteur de courant, un premier bornier sert à brancher l'entrée secondaire du transformateur.

Ce bornier va ensuite sur un deuxième bornier où on met une résistance pour faire une consommation sur le système.

Il est branché sur un troisième bornier qui est sur les COS du A4049 et revient sur le bornier du transformateur

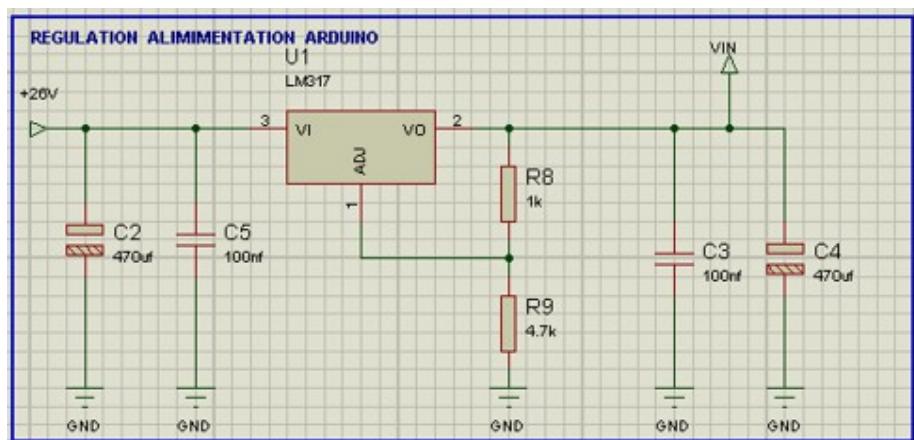


Pour l'écran I<sup>2</sup>C j'ai mis 2 résistances de 10k

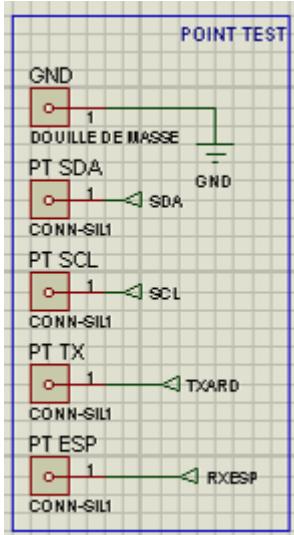


La platine est pourvue d'un bouton qui permet de passer en mode « Mesure automatique »

J'ai mis une résistance pour « lisser » le courant et éviter tout parasite. Le condensateur sert « d'anti-rebond ».



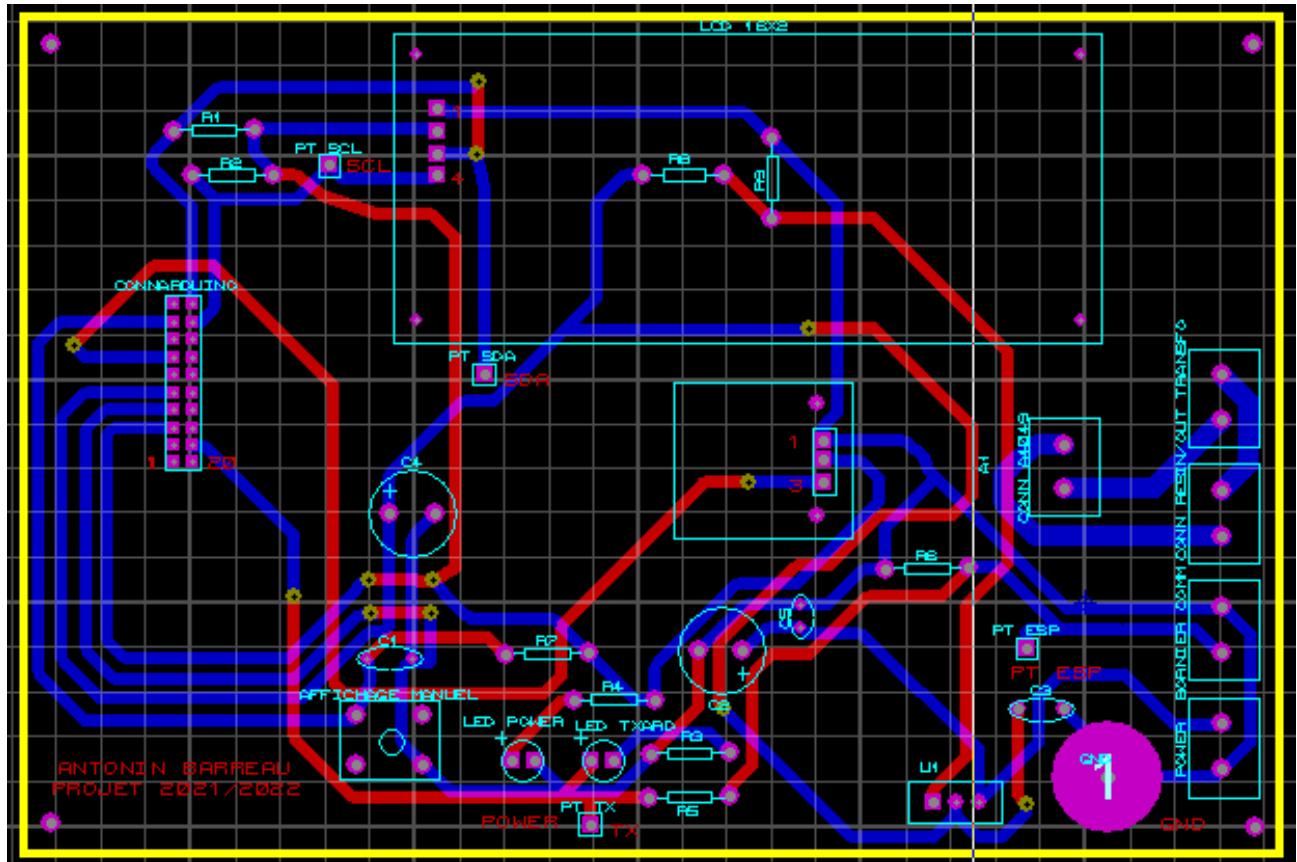
Le régulateur qui régule +26V en +5V pour l'arduino.



Les points de test pour les mesures et la douilles de masses seront dispersées sur la carte.

\*Les points de test pour les mesures et la douilles de masses seront dispersé sur la carte

# Conception de la carte à l'aide de logiciel assisté



L'image-ci dessus est mon schéma d'implantation. La carte était conçue de sorte à ce qu'un boîtier soit intégré sur la carte. Tous les borniers sont du côté droit de la carte pour une raison de pratique. Le régulateur lm317 est quant à lui placé en bordure de carte le plus éloigné possible car il chauffe.

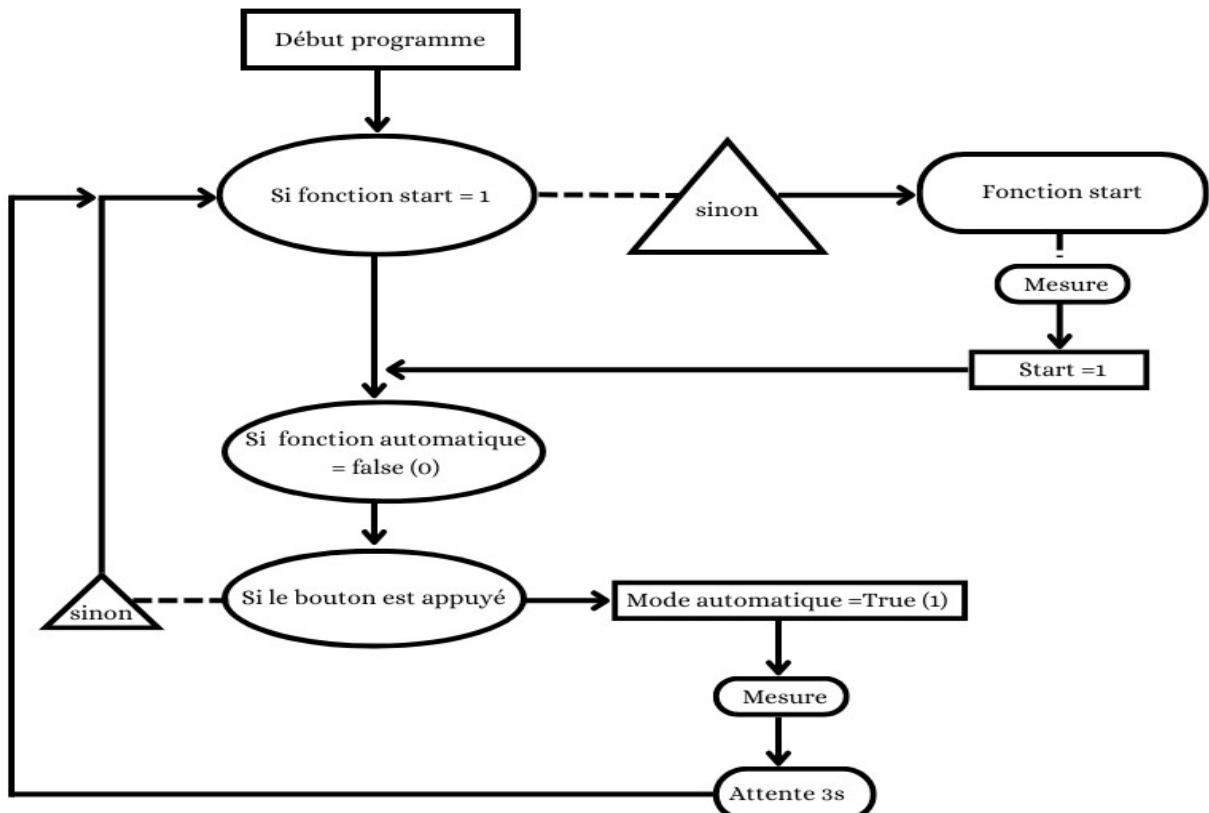
Ce schéma d'implantation n'est pas le même que la carte. Il y a quelques modifications comme le sda, scl qui n'était pas bien relié, les pistes qui étaient trop collées, la douille de masse qui était juste en taille, et l'empreinte du capteur qui avait une légère erreur de taille.

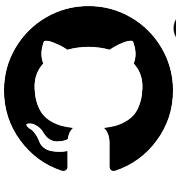
Pour le capteur, il y a en tout 3 borniers, 1 pour l'entrée du courant (transformateur par exemple), un bornier où l'on peut brancher un système (ex : des ampoules) pour tirer du courant et pour que cela soit plus pratique un bornier pour brancher le A4049.

## Le code :

(voir sur github ou doc technique).

Pour comprendre mon code voici un organigramme :





(final/finalV4)



```
// importation librairy
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <Wire.h>

// paramétrage
LiquidCrystal_I2C lcd(0x27, 16, 2);
SoftwareSerial mySerial(4, 3); // RX,TX redéfinit par la librairy

const char buttonPin = A2;
const int debit = 9600; // Débit Tx
char Demarrage; // Boucle demarrage
char MesureLoop = 0;

int buttonState = 1; // Button définit à l'état 1

void setup()
{
    // Ecran initialisé
    lcd.init();
    lcd.backlight();

    pinMode(buttonPin, INPUT); // Entrée bouton

    Serial.begin(debit);
    mySerial.begin(debit);
    Serial.println("Test communication RS232");
    delay(1000);
}

void measure()
{
    int adc = analogRead(A0); // Définit pin de lecture sur le pin
    Analogique A0
    float voltage = adc * 5 / 1023.0; // Converti la tension
    float current = (voltage - 2.50) / 0.04; // Converti le courant (Sensibilité
    40mA)

    if (current <= 0)
    { // Si le courant est inférieur ou égale à 0 il n'affiche que 0

        // Envoie dans la console le courant et tension Toutes les 8s
    }
}
```



```
// affichage écran
lcd.clear();
lcd.print("Tension: ");
lcd.print(voltage);
lcd.print("V");

lcd.setCursor(0, 1);
lcd.print("Courant: ");
lcd.print("0A");

// Envoie passerelle

Serial.println("Envoie Courant sur la passerelle");

mySerial.println(current); // Envoie du courant
mySerial.println("0A-");

// Console retour test uniquement
Serial.println("0A");

mySerial.println(voltage); // Envoie du courant
mySerial.println("V");

// Console retour test uniquement
Serial.println(voltage);
}

else
{
    // affichage écran
    lcd.clear();
    lcd.print("Tension: ");
    lcd.print(voltage);
    lcd.print("V");

    lcd.setCursor(0, 1);
    lcd.print("Courant: ");
    lcd.print(current);
    lcd.print("A");

    // Envoie passerelle

    Serial.println("Envoie Courant sur la passerelle");

    mySerial.println(current); // Envoie du courant
    mySerial.println("A-");
}
```



```
// Console retour test uniquement
Serial.println(current);

mySerial.println(voltage); // Envoie du courant
mySerial.println("V");
// Console retour test uniquement
Serial.println(voltage);
}

void start()
{
    // Démarrage système
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.print("Projet BTS");

    lcd.setCursor(3, 1);
    lcd.print("2022/2023");

    // Attente de 2s
    delay(2000);

    lcd.clear();
    lcd.print("Initialisation");
    lcd.setCursor(6, 1);

    lcd.print(".");
    delay(1000);
    lcd.print(".");
    delay(1000);
    lcd.print(".");
    delay(1000);

    lcd.clear();
    measure();
}

void loop()
{
    // Message départ/fonction pour ne pas l'avoir à chaque fois
    buttonState = digitalRead(buttonPin);
```



```
if (Demarrage == 0)
{
    start();
    Demarrage++;
    Serial.print(Demarrage);
}

if (MesureLoop == 1)
{
    measure();
    delay(1000);
}

}

if (MesureLoop == 0) {

    if (buttonState == LOW){

        lcd.clear();
        lcd.print("Passage en mode");

        lcd.setCursor(3, 1);
        lcd.print("Automatique");

        delay(2000);
        MesureLoop++;
    }
}
}
```



## Boîtier 3D :



Pour ce projet j'ai décidé de faire une boîte en impression 3D.

J'ai utilisé Fusion 360 et crée la boîte entièrement

### Dimension de la carte :

Largeur	Longueur
9,7cm (97mm)	14,5cm (145mm)

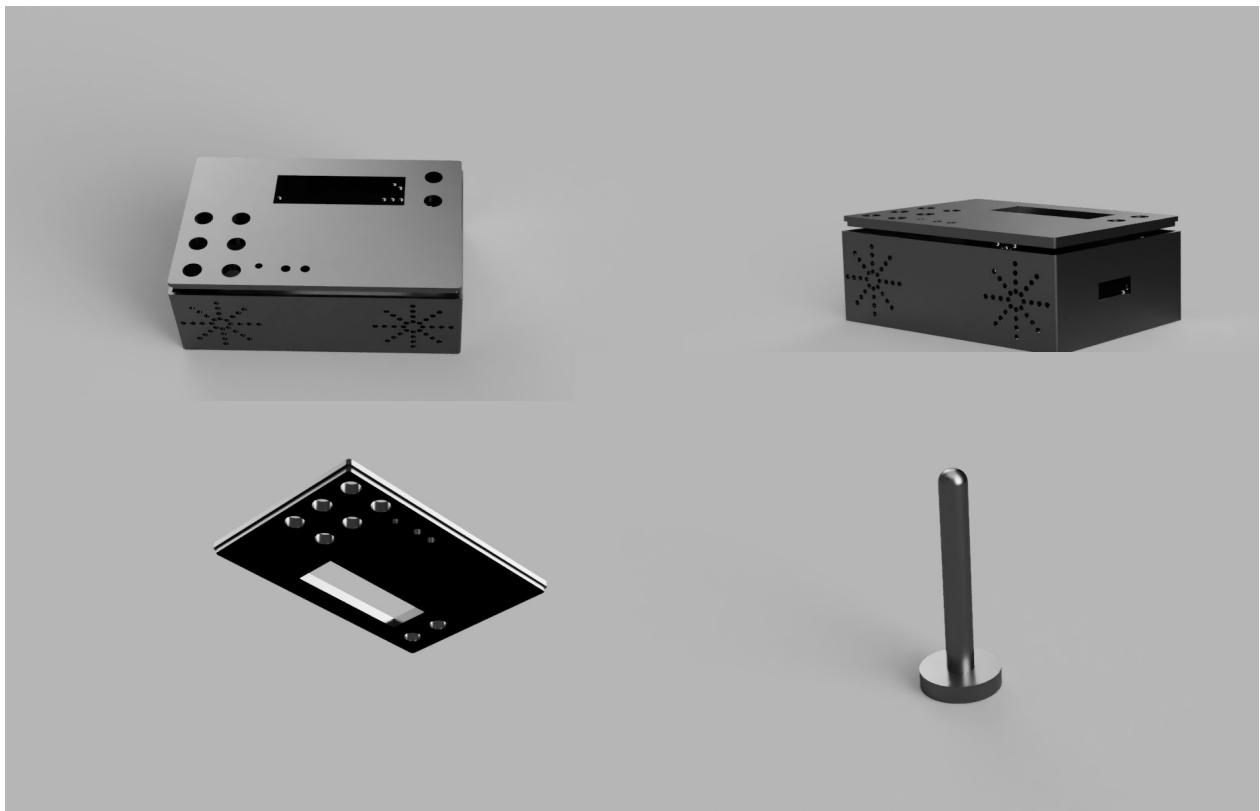
Sur le modèle 3D j'ai ajouté 12cm sur la longueur & largeur. Soit

### Dimension de la boîte :

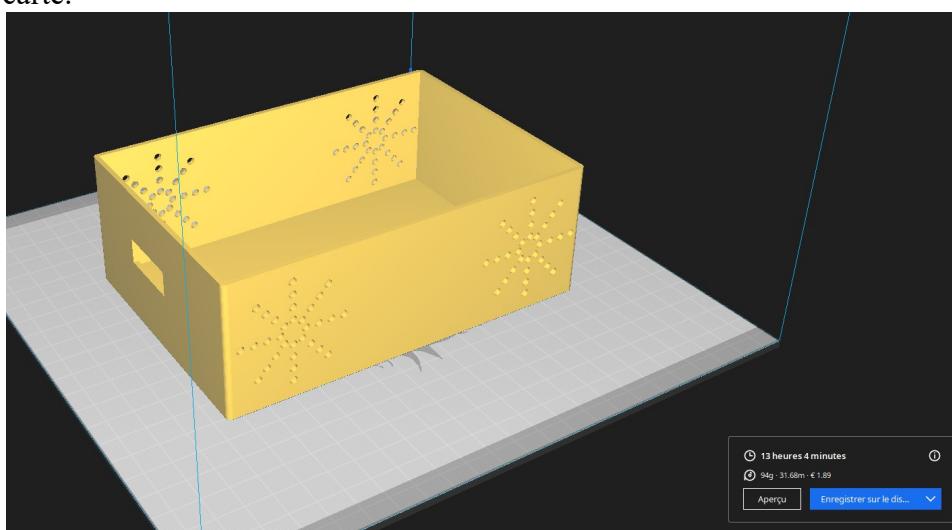
	Largeur	Longueur	Bordure
Couvercle haut	104mm	151mm	2mm
Couvercle bas	99.85mm	146.85mm	
Boîtier bas	104mm	151mm	2mm

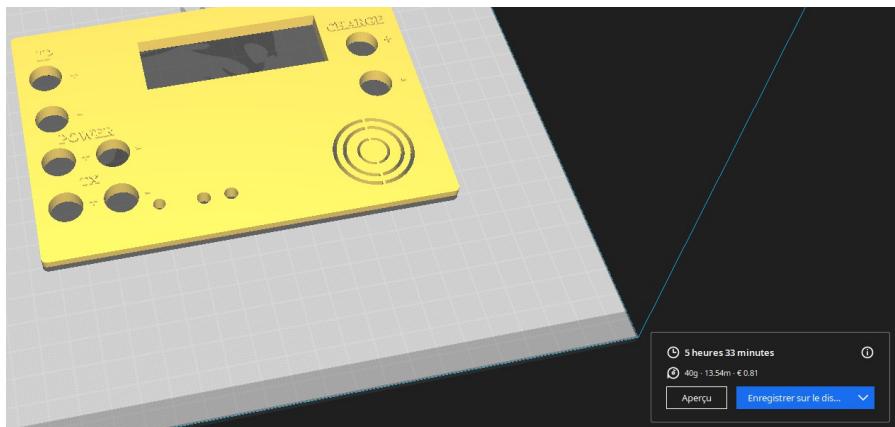
Le boîtier est emboîtable pour que ça soit simple à ouvrir et accéder à la carte.

Un aperçu depuis le logiciel « Fusion 360 » :



Le boîtier est conçu pour rentrer au millimètres près. Il y a un bouton imprimé pour atteindre le bouton de la carte. Des points tests sont reliés par des câbles aux borniers situé sur la droite de la carte.





En slicer j'utilise cura, les paramètres utilisé sont plus lent pour un résultat optimal

