

# **DOSSIER TECHNIQUE**

## **TP - SOLAIRE**

### **Panneau solaire**

*01/06/2021*

**SII / Informatique – AAM**

### **Versions :**

- 25/05/2021 : Création du document
- 01/06/2021 : Ajout de l'introduction & des parties
- 08/06/2021 : Mise en page
- 12/06/2021 : Ajout de contenu & code
- 14/06/2021 : Amélioration Code & ajout sur le compte rendu
- 16/06/2021 : Ajout de la partie « Amélioration »

### **Remerciements :**

- SAINTFELIX Audrey, pour ses informations précieuses au niveau de la carte Arduino
- Mr. CHAZOTTES-LECONTE, notre professeur de sciences industrielles
- Mr. FLANDROIS, notre professeur d'informatique

## Table des matières

Table des matières .....	3
<b>I) Introduction :</b> .....	4
<i>Objectifs :</i> .....	4
<i>Matériel :</i> .....	4
<b>II) Cahier des charges :</b> .....	5
<i>Cahier des charges pour la matière Informatique :</i> .....	5
<i>Cahier des charges la matière SII :</i> .....	5
<b>III) Fabrication du panneau solaire :</b> .....	5
Montage : .....	5
<b>IV) Elaboration du code Arduino :</b> .....	8
<i>1<sup>ère</sup> partie : Faire un programme qui permet de contrôler manuellement l'orientation du panneau solaire.</i> .....	8
<i>2<sup>nd</sup> partie : Faire un programme qui permet de contrôler automatiquement l'orientation du panneau solaire à l'aide de 4 photorésistances.</i> .....	10
<b>V) Améliorations apportées / Optimisations :</b> .....	14
Roulement à bille : .....	14
Data logger : .....	15
<b>VI) Conclusion :</b> .....	16

## I) Introduction :

### *Objectifs :*

Dans le cadre de ce TP, l'objectif général est d'explorer l'efficacité d'une installation disposant d'un système de suivi du soleil par rapport à une installation fixe.

Notre objectif de ce projet est de construire un panneau solaire qui se positionnera de sorte à suivre le soleil. Nous allons lier le domaine de l'informatique enseigné par M. Flandrois et le domaine des sciences industrielles enseigné par M. Chazotte-Lecompte.

En informatique, il nous est proposé d'étudier deux scénarios. Le premier scénario met en scène deux servos moteurs Arduino devant décrire un arc de cercle. Le second scénario concerne notre objectif principal : faire en sorte que notre panneau solaire puisse suivre le soleil à l'aide de capteurs et de servomoteurs.

### *Matériel :*

- 1 microcontrôleur Arduino
- 2 servomoteurs 5.0 v
- 4 capteurs de luminosité : LDR 2,20 k $\Omega$
- Batteries
- Plaque en bois de 1 cm
- Tout le matériel du FabLab

## II) Cahier des charges :

### *Cahier des charges pour la matière Informatique :*

1. Créer un Flowchart pour chacun des scénarios décrit dans « Introduction / Objectifs ».
2. Prendre le control des servomoteurs
3. Faire interagir les moteurs en fonctions des résultats de capteurs de luminosité.

### *Cahier des charges la matière SII :*

1. Réaliser le prototype de panneau solaire
2. Bonus : Apporter des modifications au prototype pour améliorer son efficacité et aussi son apparence.

## III) Fabrication du panneau solaire :

Pour réaliser ce panneau solaire, nous étions fournis de pièces en bois découpé par la découpe laser du FabLab. Nous avons donc assemblé les pièces à l'aide des outils du Fablab. Nous avons débuté par le châssis puis par le haut du panneau solaire.

### Montage :

- 1) Châssis :

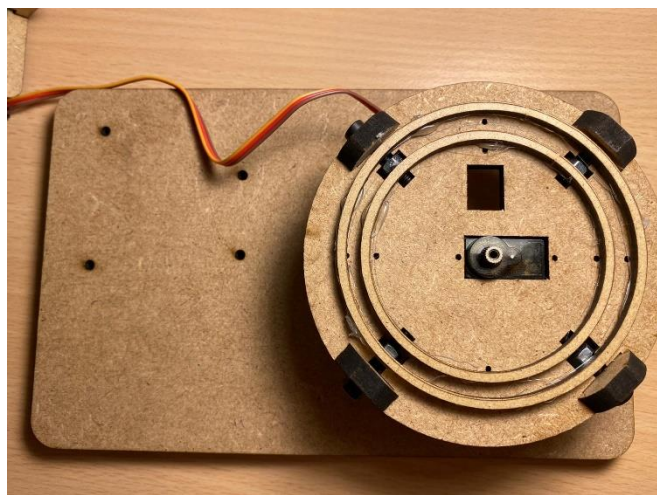


Photo 1 : Bâtis vu de haut

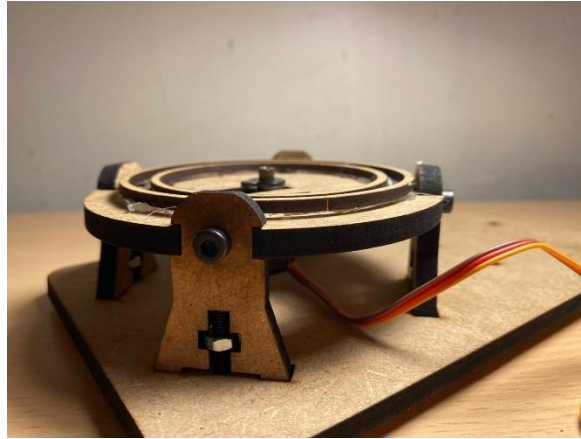


Photo 2 : Vue de gauche

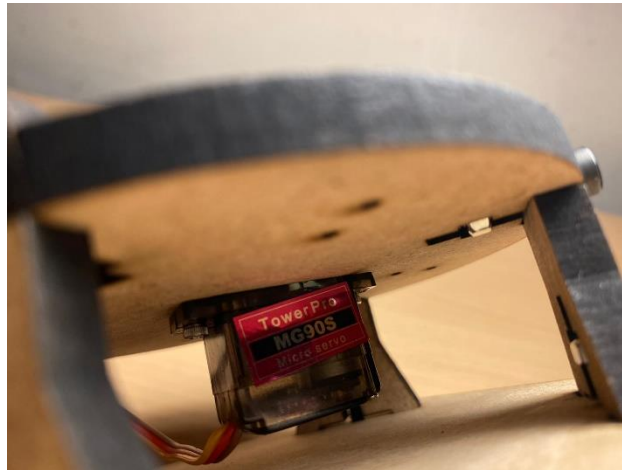


Photo 3 : Vue de droite

Le bâti est assemblé d'un plateau supportant un servomoteur.

## 2) Tourelle

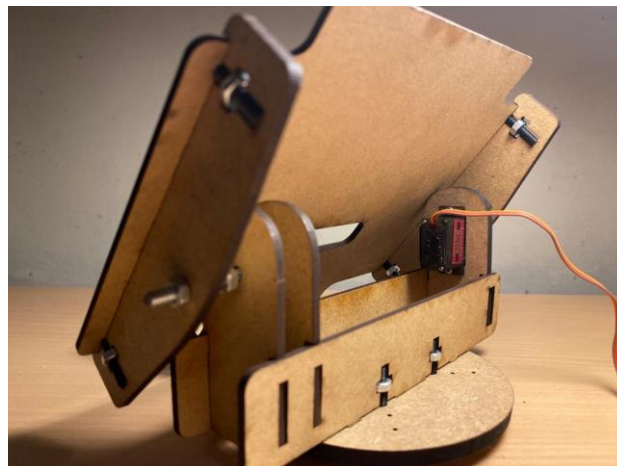


Photo 4 : Vue de derrière

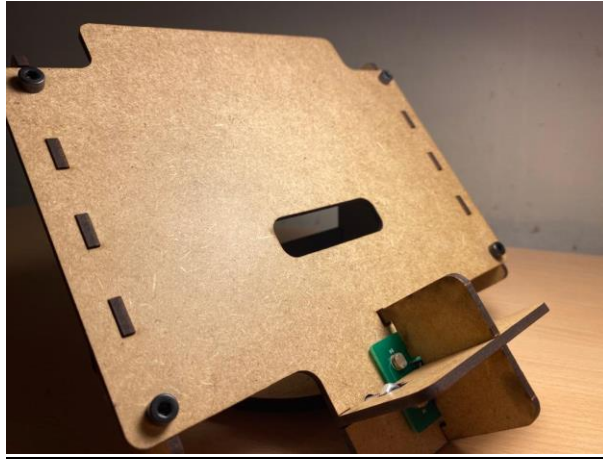


Photo 5 : Vue de devant

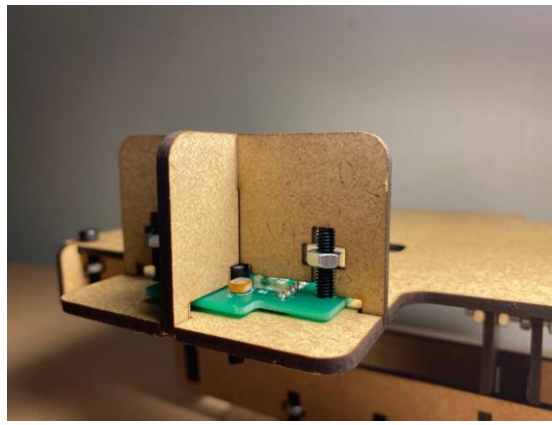


Photo 6 : Photo résistance du panneau

Au niveau de notre support de notre cellule photo voltaïque, nous avons 4 photorésistances permettant l'orientation de ce support.

## IV) Elaboration du code Arduino :

*1<sup>ère</sup> partie : Faire un programme qui permet de contrôler manuellement l'orientation du panneau solaire.*

Pour contrôler les deux servos, nos objectifs a été de les relier à deux potentiomètres permettant chacun, de contrôler un servo respectif.

Le schéma :

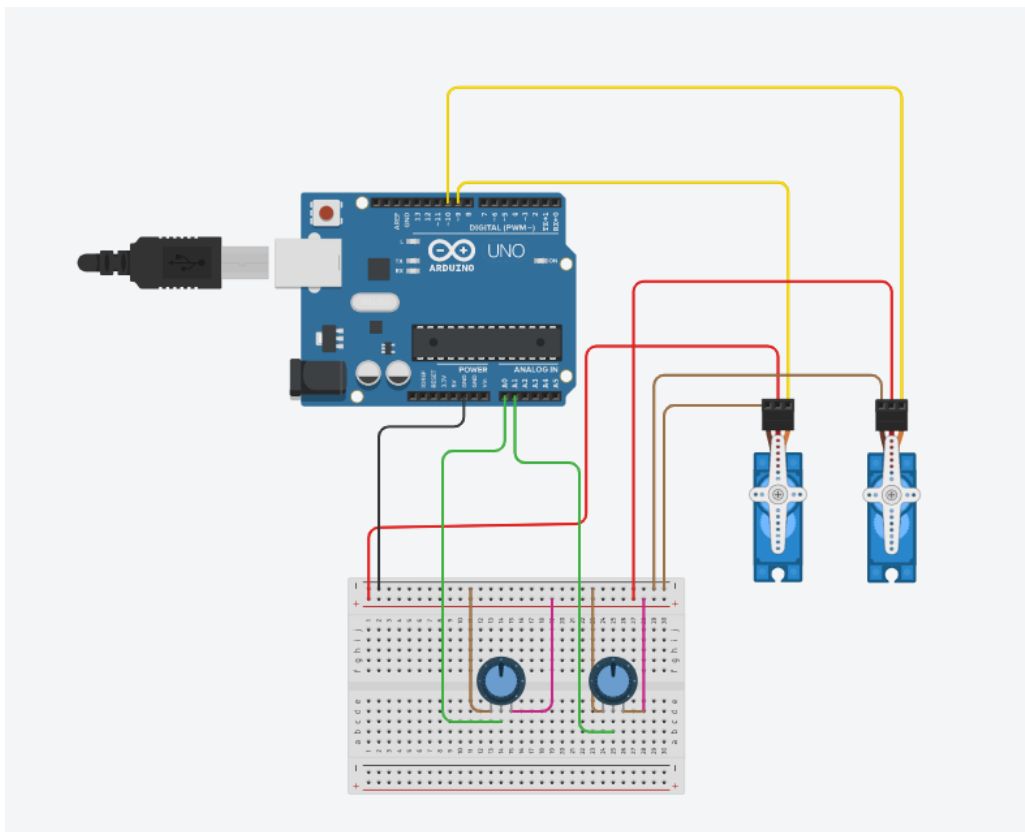


Schéma n°1 : Deux servos en connexion avec deux potentiomètres



Ce code est axé sur le bon fonctionnement de nos servos. En effet, il nous a permis de considérer les améliorations que nous avons pu lui apporter tel que le roulement à bille. Nous avons remarqué avec l'utilisation des potentiomètres. Cette étape nous a aussi permis de relever les potentielles erreurs concernant l'assemblage du panneau solaire ainsi que la carte Arduino.

Voici le code ci-dessous :

```
#include <Servo.h> //On commence par inclure la bibliothèque de commande des servos
#define PIN_POT1  A0
#define PIN_POT2  A1

const int potentiometre = 0; // Initialisation du potentiomètre
Servo Servo1;           // Initialisation du servo-moteur n°1
Servo Servo2;           // Initialisation du servo-moteur n°2

void setup()
{
  Servo1.attach(9);      // on va commander l'angle du servo-moteur avec la pin 9
  Servo1.write(0);        // on place le servo à 0 degré au démarrage
  Servo2.attach(10);     // on va commander l'angle du servo-moteur avec la pin 10
  Servo2.write(0);       // on place le servo à 0 degré au démarrage
}

void loop()
{
  val_1 = analogRead(PIN_POT1);
  pos_1 = map(val_1, 0, 1023, 0, 180);
  Servo1.write(pos_1);    // tell servo to go to position in variable 'pos'
  val_2 = analogRead(PIN_POT2);
  pos_2 = map(val_2, 0, 1023, 0, 180);
  Servo2.write(pos_2);
  Serial.println(pos_1);
  Serial.println(pos_2);
  delay(100);
}
```

*2<sup>nd</sup> partie : Faire un programme qui permet de contrôler automatiquement l'orientation du panneau solaire à l'aide de 4 photorésistances.*

Dans un premier temps, ne commençons par introduire le fonctionnement du capteur de luminosité. Dans notre système, nous en avons un total de 4.

Le code :

Pour contrôler les deux servos, nos objectifs a été de les relier les 4 photorésistances aux deux servos.

Voici une vidéo du fonctionnement de notre panneau solaire en fonctionnement :

Voici le code avec les 4 photorésistances ci-dessous :

```
// Coding in cpp
// TP - Solarius
// LEGER / AUGER / LE-DEVEHAT

// Nous allons appeler la fonction que nous allons utiliser
#include <Servo.h>

// Ici, nous initialison le servo qui décrit l'axe horizontal
Servo horizontal;
int servoh = 90;

// Nous allons définir les limites d'angle de notre servo afin qu'il ne puisse pas
s'endommager
int servohLimitHigh = 180;
int servohLimitLow = 65;

// Ici, nous initialison le servo qui décrit l'axe horizontal
Servo vertical;
int servov = 90;

// Nous allons définir les limites d'angle de notre servo afin qu'il ne puisse pas
s'endommager
int servovLimitHigh = 120;
int servovLimitLow = 15;

// Nous allons définir les pin(s) de connections pour les photos résistances
int ldrTR = 0; // LDR haut droite
int ldrTL = 1; // LDR haut gauche
int ldrBR = 2; // LDR bas droite
int ldrBL = 3; // LDR bas gauche

void setup() {
  Serial.begin(9600);
  // Nous attachons les servos à leur pin respectif
  horizontal.attach(9);
  vertical.attach(10);
  // nous faisons bouger les servos
```

```
horizontal.write(90);
vertical.write(45);
delay(5000); // le délai peut être variable
}

void loop() {

// on implémente les valeurs des photorésistances

int tr = analogRead(ldrTR); // haut droite
int tl = analogRead(ldrTL); // haut gauche
int br = analogRead(ldrBR); // bas droite
int bl = analogRead(ldrBL); // bas gauche

int dtime = 0; // cette fonction permet de débbuguer le système
int tol = 5;

int avt = (tl + tr) / 2; // valeur moyenne des photorésistance du haut
int avd = (bl + br) / 2; // valeur moyenne des photorésistance du bas
int avl = (tl + bl) / 2; // valeur moyenne des photorésistance de gauche
int avr = (tr + br) / 2; // valeur moyenne des photorésistance de droite

int dvert = avt - avd; // on regarde la différence entre les valeurs du haut et
du bas
int dhoriz = avl - avr; // on regarde la différence entre les valeurs de la
droite et de la gauche

// on print l'ensemble des valeurs calculées

Serial.print(tl);
Serial.print(" ");
Serial.print(tr);
Serial.print(" ");
Serial.print(bl);
Serial.print(" ");
Serial.print(br);
Serial.print(" ");
Serial.print(avt);
Serial.print(" ");
Serial.print(avd);
Serial.print(" ");
Serial.print(avl);
Serial.print(" ");
Serial.print(avr);
Serial.print(" ");
Serial.print(dtime);
Serial.print(" ");
Serial.print(tol);
Serial.print(" ");
Serial.print(servov);
Serial.print(" ");
Serial.print(servoh);
Serial.println(" ");

// on regarde la différence entre les capteurs si les angles changes avec la
modification verticale

if (-1 * tol > dvert || dvert > tol) {
    if (avt > avd) {
        servov = ++servov;
        if (servov > servovLimitHigh) {
            servov = servovLimitHigh;
        }
    }
    else if (avt < avd) {
```

```
servov = --servov;
if (servov < servovLimitLow) {
    servov = servovLimitLow;
}
}
vertical.write(servov);
}

// on regarde la différence entre les capteurs si les angles changent avec la
modification horizontale

if (-1 * tol > dhoriz || dhoriz > tol) {
    if (avl > avr) {
        servoh = --servoh;
        if (servoh < servohLimitLow) {
            servoh = servohLimitLow;
        }
    }
    else if (avl < avr) {
        servoh = ++servoh;
        if (servoh > servohLimitHigh) {
            servoh = servohLimitHigh;
        }
    }
    else if (avl = avr) {
        // nothing
    }
    horizontal.write(servoh);
}

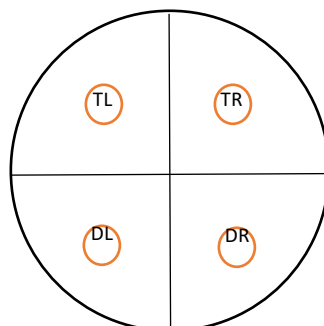
delay(dtime);
}
```

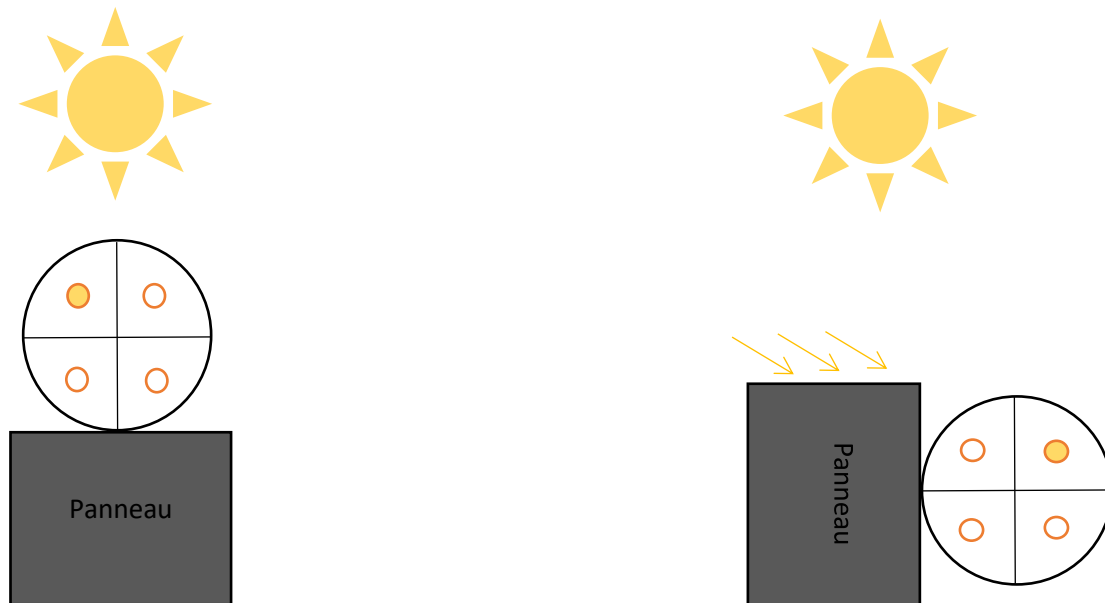
Lors de l'essai du code sur notre système, nous avons dû faire des tests sur la tolérance de nos capteurs. Nos tests ont été réalisés sur une terrasse ensoleillée. Notre problème fut que notre panneau ne faisait pas de rotation en fonction de la luminosité. En divisant la tolérance par 10, nous avons remarqué que notre système admettait mieux les moyennes de luminosité et donc, in fine, fonctionnait « parfaitement » dans le cadre de l'exercice.

### *Photorésistance :*

Afin de capter la lumière et d'orienter le panneau solaire de façon à obtenir un maximum d'énergie, il est nécessaire d'ajouter des photorésistances. Sur ce panneau, nous en avons un total de 4 formants en carré. Leur principe est simple : plus la lumière est élevée, plus la résistance est faible.

Schématisation :



*Schéma de la résistance**Schéma explicatif*

Ici, nous sommes dans le cas où la résistance en haut à gauche reçoit le plus de lumière et donc possède une résistance moins élevée que les autres. Grâce au code et à la carte Arduino, les servos vont s'orienter de façon à tourner le panneau vers la gauche et vers le haut.

## V) Améliorations apportées / Optimisations :

### *Roulement à bille :*

Nous avons eu du mal à le faire car nous avons eu des soucis d'échelle au niveau du coffrage des billes avec la découpe laser. Un souci d'échelle concernant notre protection de notre Arduino ne nous pas permis de réaliser les améliorations voulues.

En effet, Plusieurs découpes Laser ont été effectuées, cependant, ce roulement n'est pas fonctionnel. En effet, lors de la première tentative, nous avons été confrontés à des soucis de mesures suite à l'utilisation du logiciel CREO. De plus, lors du dernier essai, les dimensions étaient bonnes, cependant l'épaisseur des rails contenant la bille étaient trop épais et ne permettait donc pas d'enclencher le servo.

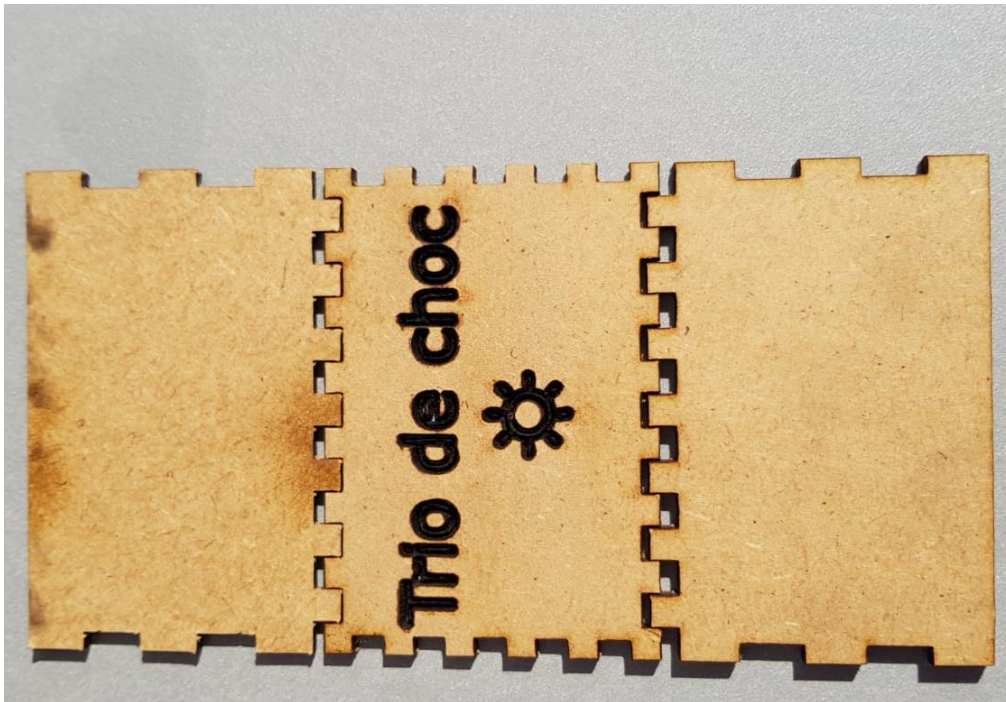


Photo du test de la protection de la carte Arduino

Nous avons prévu de réaliser une boîte pour protéger les principaux câblages. Cependant, Nous avons également rencontré des problèmes de dimensions.

Concernant le panneau solaire, nous avons voulu ajouter un support pour le panneau solaire. Il nous aurait permis de rendre le panneau amovible et de pouvoir l'échanger en cas de problème. Cependant nous avons également été confrontés à des soucis de mesures suite à l'utilisation du logiciel CREO.

### *Data logger :*

Le data logger est un moyen d'enregistrer les caractéristiques de puissance que fournit notre panneau solaire. Avec la relation de la puissance  $P = Ri^2$ , nous devons sauvegarder sur une carte SD les valeurs de puissance fournies par le panneau à intervalle régulier.

#### Code :

```
#include <SPI.h>
#include <SD.h>

const char* filename = "Datalogger_solarius.csv";

File file;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(9600);

    pinMode(10, OUTPUT);

    if (!SD.begin(10)){
        Serial.println("Error : Push the reset button");
        for(;;);
    }

    file = SD.open(filename, FILE_WRITE);
}

void loop() {
    // put your main code here, to run repeatedly:

    measure();
    delay(1000);
}

void measure(){
```

```
int val = analogRead(A0);  
float tension = val/1023 * 5;  
float R= 100.0;  
float intensity = tension/R;  
float power = intensity * tension;  
file.println(power);  
file.flush();  
}
```

## VI) Conclusion :

Ce projet, alliant à la fois les sciences industrielles ainsi que l'informatique nous a permis de nous améliorer dans ces deux disciplines. Nous avons pu utiliser les outils du fablabs et nous avons pu approfondir les connaissances en Arduino acquises tout au long de l'année. Avec du recul et une séance en plus, nous aurions pu ajouter un data logger permettant l'étude de la puissance qu'aurait fourni la cellule durant sa mise en opération.