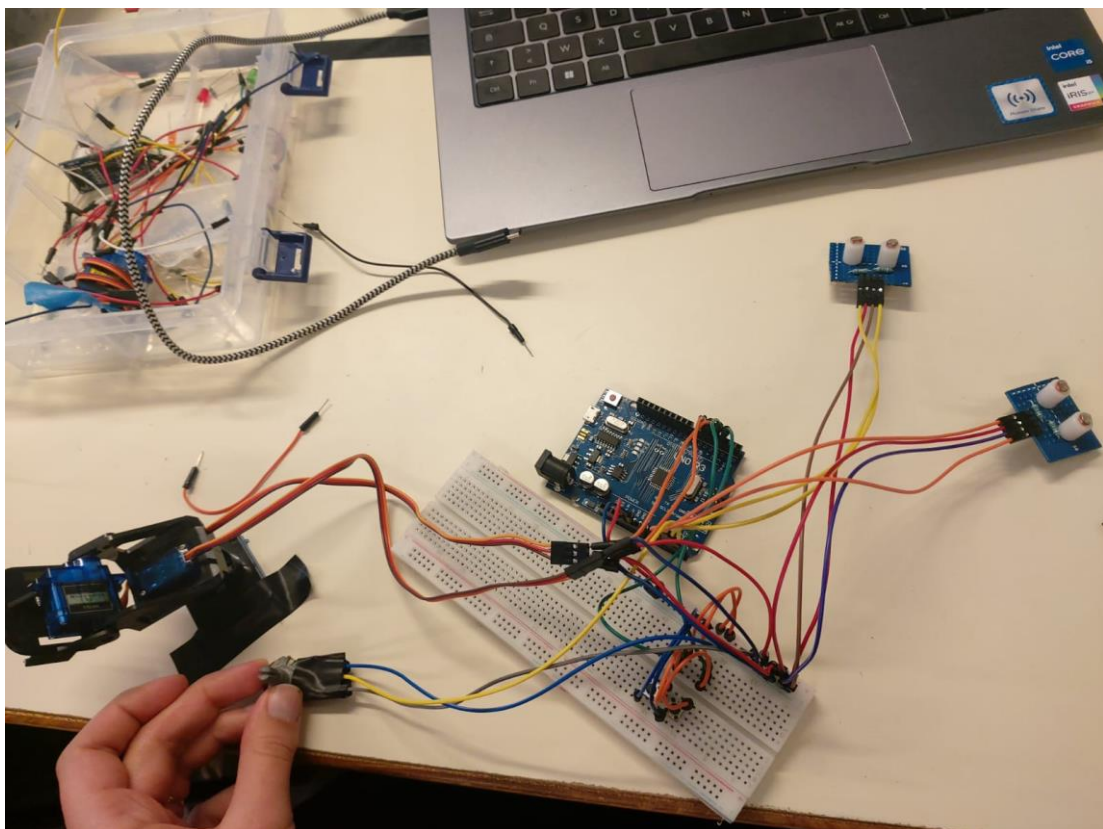


## Rapport de s  ance 6

Au d  but de la s  ance Roxane et moi avons fait le point sur les branchements et avons conclu qu'il faudrait les cacher dans un coffre. Roxane s'est occup  e du design du coffre en tenant compte des indications que je lui donnais, comme avoir les boutons et la molette du potentiom  tre apparents par exemple. Pour « a  rer » un peu plus ce montage nous avons d  cid   de souder les c  bles sur une petite plaquette qui fera office de passerelle entre les entr  es 5V et GND de la carte Arduino et celles de tous les autres composants. De plus, les boutons poussoirs seront tous les deux fix  s sur une petite plaque (comme celles sur lesquelles nous avons mis les LDRs), qui sera coll  es sur le coffre du Solar Tracker.

En plus de ce coffre, il nous manquait une petite pi  ce    placer entre le servo moteur (haut-bas), qui suivrait ses mouvements et sur laquelle on pourrait fixer le support du panneau solaire. Roxane a demand      Mr Masson quelle serait la solution la plus optimale pour concevoir cette pi  ce et finalement le professeur nous a donn   un autre syst  me de servos moteurs qui avaient une pi  ce du m  me genre d  j   fix  e    l'endroit voulu.

Ensuite j'ai pass   le reste de la s  ance    travailler sur le mode manuel du Solar Tracker. J'ai commenc   par revoir mon montage car j'ai remarqu   quelques erreurs sur le montage de la derni  re fois. Voici une photo du montage final :



Puis j'ai recommencé à travailler sur le programme. La dernière fois je pensais que j'avais réussi à faire marcher le passage du mode manuel au mode automatique mais je me suis rendue compte que ce n'était qu'une coïncidence et j'ai dû retravailler cette fonctionnalité. Pour se faire je me suis basée sur le programme que nous avons fait en cours au premier semestre, pour transformer un bouton poussoir en interrupteur. Une fois ce problème réglé il a fallu que je travaille sur la lecture des valeurs du potentiomètre pour faire bouger le servo moteur choisi. Pour ça je lis les valeurs du potentiomètre qui sont comprises entre 0 et 1023, et je les convertis grâce à la fonction « map » pour qu'elles soient comprises entre 0 et 180 (plage de valeurs des servos moteurs). Ensuite je rentre cette valeur dans la fonction write pour les servos. Je n'ai pas pu tester le programme final car je l'ai retravaillé un peu après le cours et je n'avais pas le reste du matériel avec moi.

```
#include <Servo.h>

int haut;
int gauche;
int droite;
int bas;
int val1=1;
int ancien_val1=1;
int etat1=1;
int val2=1;
int ancien_val2=1;
int etat2=1;
int state;
int mode=1;
int ButtonUD=3;
int ButtonLR=4;
int reglage=4; //potentiomètre
int LDR1=0;    //gauche
int LDR2=1;    //droite
int LDR3=2;    //haut
int LDR4=3;    //bas
Servo servoUD;
Servo servoLR;

void setup() {
  pinMode(ButtonUD, INPUT);
  pinMode(ButtonLR, INPUT);
  pinMode(reglage, INPUT);
  servoUD.attach(5);
  servoLR.attach(6);
  Serial.begin(9600);
}
```

```

void setup() {
  pinMode(ButtonUD, INPUT);
  pinMode(ButtonLR, INPUT);
  pinMode(reglage, INPUT);
  servoUD.attach(5);
  servoLR.attach(6);
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  val1=digitalRead(ButtonUD);
  val2=digitalRead(ButtonLR);
  if ((val1==LOW) && (ancien_val1==HIGH)) {
    etat1=1-etat1;
  }
  ancien_val1=val1;
  if ((val2==LOW) && (ancien_val2==HIGH)) {
    etat2=1-etat2;
  }
  ancien_val2=val2;
  if ((etat1==HIGH) && (etat2==HIGH)) {
    automaticTracker();
  }
  else {
    if (etat1==LOW) {
      mode=1;
    }
    else{
      mode=0;
    }
    manualTracker();
  }
}

void manualTracker() {
  state=analogRead(reglage);
  state=map(state,0,1023,0,180);
  if (mode==1){
    servoUD.write(state);
  }
  else{
    servoLR.write(state);
  }
}

```

```

void automaticTracker(){
  gauche=analogRead(LDR1);
  droite=analogRead(LDR2);
  haut=analogRead(LDR3);
  bas=analogRead(LDR4);
  gauche=map(gauche,1,959,0,180); //étalonnage fait
  droite=map(droite,0,977,0,180); //étalonnage fait
  haut=map(haut,2,967,0,180); //en attente de l'étalonnage
  bas=map(bas,0,978,0,180); //en attente de l'étalonnage
  int diffHB=haut-bas;
  int diffGD=gauche-droite;

  if (abs(diffHB)>2){
    if (diffHB<0){
      if (servoUD.read()<180){
        servoUD.write(servoUD.read()+1);
      }
    }
    else{
      if (servoUD.read()>0){
        servoUD.write(servoUD.read()-1);
      }
    }
  }
  if (abs(diffGD)>2){
    if (diffGD<0){
      if (servoLR.read()<180){
        servoLR.write(servoLR.read()+1);
      }
    }
    else{
      if (servoLR.read()>0){
        servoLR.write(servoLR.read()-1);
      }
    }
  }
  delay(100);
}

```

Pour les prochaines séances, il nous restera la soudure des fils à faire, le montage du convertisseur au bout duquel on pourra brancher une batterie portable en USB, le collage du support du panneau solaire et de ses parois, et, bien sûr, vérifier que l'ensemble fonctionne.