

## Compte rendu – Première séance

### Travail réalisé lors de la première séance :

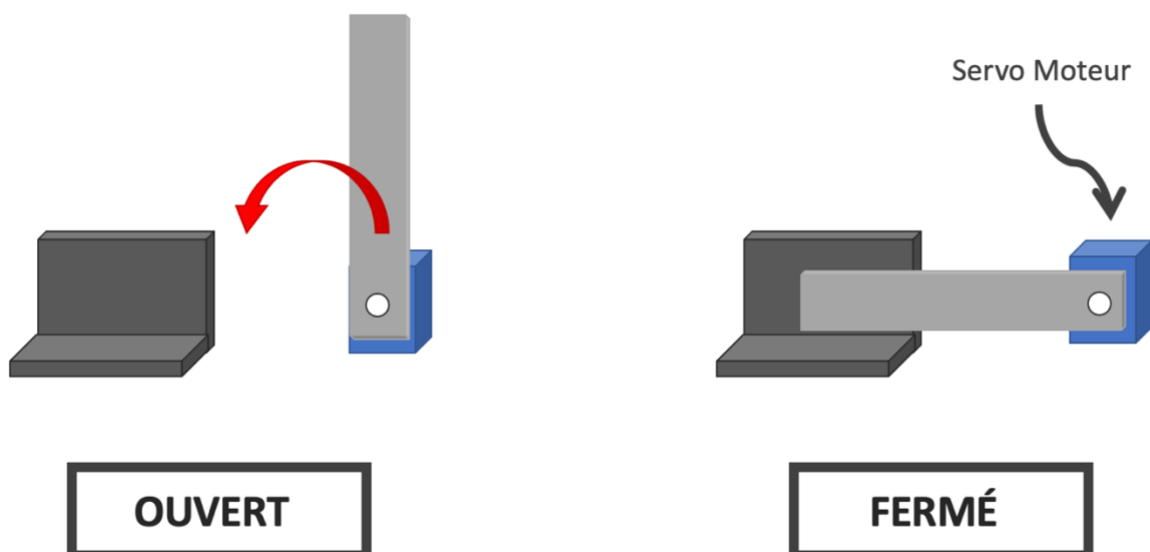
Lors de cette première séance, j'ai travaillé sur le système de fermeture de la boîte. Après réflexion un verrou à glissière s'avère moins pratique qu'un verrou qu'on fait basculer. Il suffit alors de programmer un Servomoteur pour effectuer des rotations de 90° suivant sa position initiale.

### Problèmes :

Le Servomoteur fonctionne et effectue bien les rotations mais ne respecte pas les conditions d'activation. Pour l'instant le bouton ne permet pas d'actionner le Servomoteur mais cela n'étant pas nécessaire à la production j'ai trouvé un autre moyen de l'activer, directement avec le clavier de l'ordinateur sur le terminal (ce qui sera plus utilisé pour lier un smartphone plus tard).

Le programme ne marchait pas également au début puisque la bibliothèque permettant de piloter un servomoteur n'était pas activée. Il fallait alors rajouter la commande `#include <Servo.h>` au début du programme pour l'activer.

### Changement de type de verrou



## Premier code avec bouton.

```
#include <Servo.h> // on inclut la bibliothèque pour le servomoteur

Servo servoM;          // creation de l'objet servomoteur
const int bouton = 6;
int etat_bouton = 1;
int ancienne_val =1;
int val =1;
int pos =0;

void setup(){
  pinMode(bouton, INPUT);

  servoM.attach(9);    //permet d'attacher le servomoteur au Pin d'asservissement

  Serial.begin(9600);
}

void loop(){

  val = digitalRead(bouton;

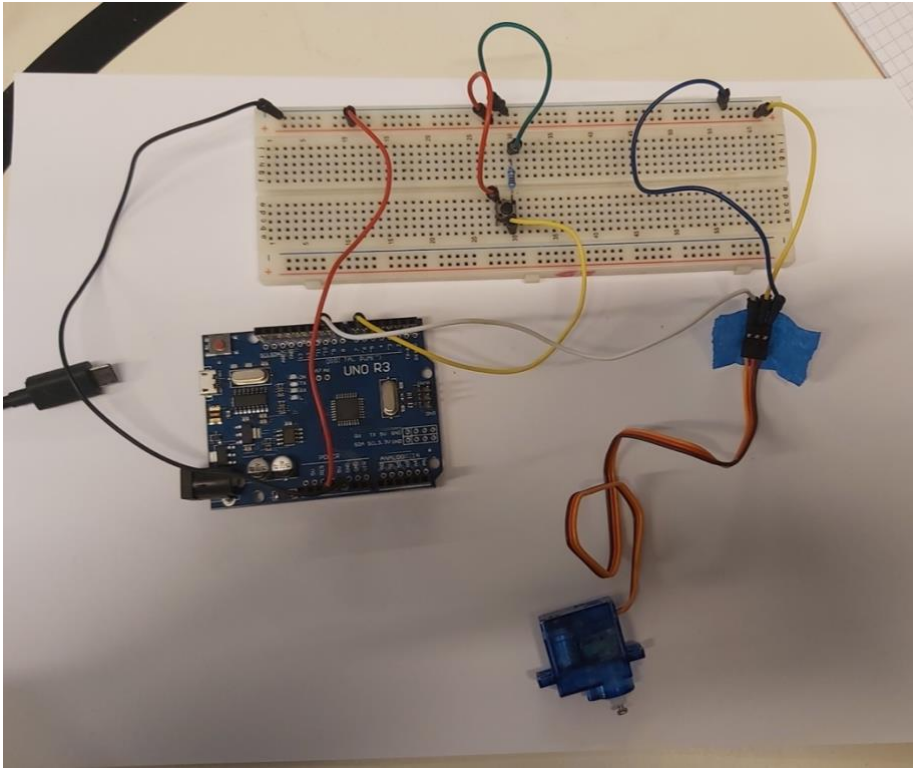
  if ((val == HIGH)&&(ancienne_val == LOW)){
    etat_bouton = 1 - etat_bouton;
    delay(20);
  }

  ancienne_val=val;

  if (etat_bouton == HIGH){
    for (pos = 0; pos <= 90; pos += 1) {
      servoM.write(pos);          //envoi le servo à l'angle spécifié
      delay(15);
      Serial.println(etat_bouton);
    }
  }

  if(etat_bouton == LOW) {
    for (pos = 90; pos >= 0; pos -= 1) {
      servoM.write(pos);
      delay(15);
      Serial.println(etat_bouton);
    }
  }
}
```

## Montage (avec bouton)



Avec ce code et ce montage, on obtenait bien un changement de valeur à l'entrée de la carte lors de l'appuie sur le bouton mais le servomoteur ne bloquait pas ses positions (mouvement constant entre un angle de 90° et de 0°).

```
1
0
1
0
1
0
```

Le problème venait donc du code (surement des deux boucles "for"), et sera rectifié pour la prochaine séance.

En enlevant le bouton poussoir et commandant le servomoteur avec le clavier de l'ordinateur, j'ai réussi à simplifier le code et à le faire fonctionner comme voulu.

**Second code avec contrôle sur le terminal (qui sera utilisé par la suite).**

```
#include <Servo.h>

int incomingByte;
Servo servo_9;

void setup()
{
  servo_9.attach(9);
  Serial.begin(9600);
}

void loop(){

  if (Serial.available() > 0) {
    incomingByte = Serial.read();
```

```

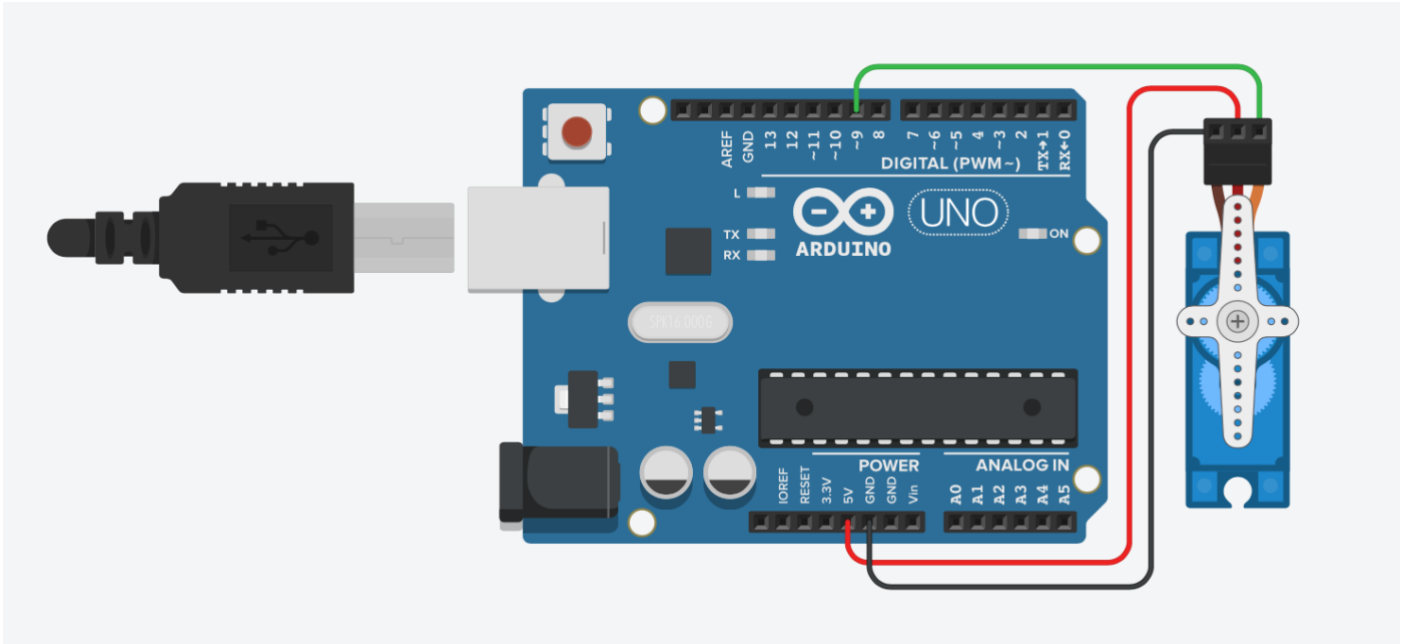
}

if (incomingByte == 'a'){
    servo_9.write(90);
}

if (incomingByte == 'b'){
    servo_9.write(-90);
}
}

```

## Montage (sans bouton)



## Sites internet utilisés

- <https://www.tinkercad.com>
- <https://projetsdiy.fr/piloter-servomoteur-arduino/>
- <https://www.arduino.cc/en/Tutorial/BuiltInExamples/PhysicalPixel>