

FICHE TP N°14

Commande lente d'un aiguillage grâce à un servomoteur

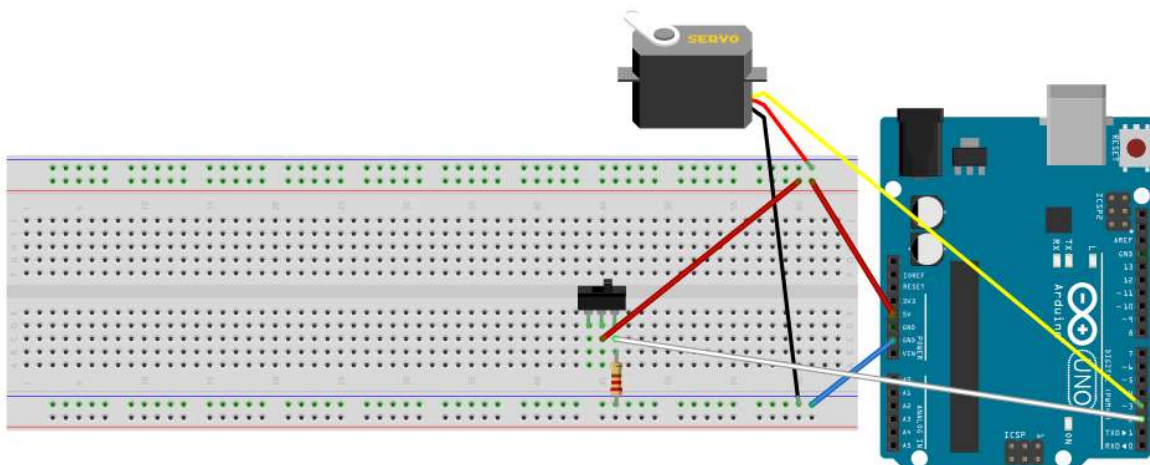
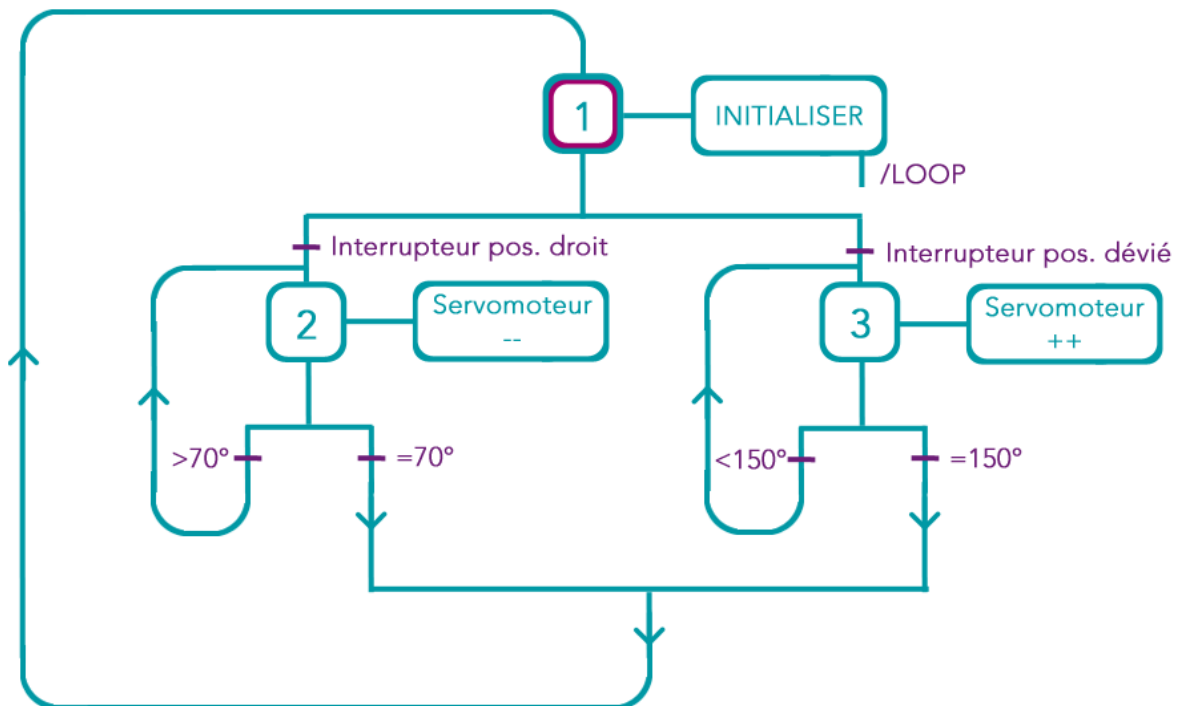
Composants nécessaires :

- ▶ Un servomoteur
- ▶ Une résistance de 10 kΩ
- ▶ Un interrupteur

Objectif :

Grâce au succès du TP13, vous décidez d'optimiser et fiabiliser le fonctionnement de vos moteurs d'aiguillage en modifiant le programme de manière à ce que le mouvement de l'aiguille soit lent, comme en réalité.

Schéma de câblage identique au TP13



```
#include <Servo.h>

#define pin_switch_1 2
#define pin_servo_A1 3
#define millisecondes_par_angle 250 // MODIFIE la vitesse du mouvement

Servo servo_A1;

int angle_A1_droit = 70;
int angle_A1_devie = 150;

int angle_A1_voulu;
int angle_A1_actuel;
unsigned long debut_mouvement_A1 = 0;
```

```
void setup()
{
  // initialisation servo
  pinMode(pin_switch_1, INPUT);
  servo_A1.attach(pin_servo_A1);
  delay(1000);

  // Entrebaillement servo
  servo_A1.write(90);
  angle_A1_actuel = 90;
  delay(1000);
}
```

```
void loop() {
  // Lecture bouton
  if(digitalRead(pin_switch_1) == LOW && debut_mouvement_A1 == 0) {
    angle_A1_voulu = angle_A1_droit;
    debut_mouvement_A1 = millis(); }
  if(digitalRead(pin_switch_1) == HIGH && debut_mouvement_A1 == 0) {
    angle_A1_voulu = angle_A1_devie;
    debut_mouvement_A1 = millis(); }

  // Fonction des mouvements
  if(angle_A1_voulu != angle_A1_actuel) {
    servo_A1.attach(pin_servo_A1);
    if((millis() - debut_mouvement_A1) > millisecondes_par_angle) {
      if(angle_A1_voulu > angle_A1_actuel) {
        angle_A1_actuel++; }
      else {
        angle_A1_actuel--; }
      servo_A1.write(angle_A1_actuel); }
    }
  else {
    servo_A1.detach();
    debut_mouvement_A1 = 0; }

  delay(millisecondes_par_angle/5); }
```