

SERVOMOTEUR**CODE ARDUINO**

```
// Croquis P7-Servo.ino

//----- LIBRAIRIE ADDITIONNELLE -----
#include <Servo.h> // Librairie pour l'utilisation de servomoteurs

//----- CONSTANTES -----
Servo Servo1; // création de l'objet Servo1 (notre SERVOMOTEUR)
#define Pot 0 // POTENTIOMETRE sur la ligne analogique 0 (A0)
#define Led 5 // LED sur la ligne E/S 5

//----- VARIABLES -----
int valPot; // Variable du POTENTIOMETRE
int angle; // valPot converti en angle (entre 0 et 180 degrés)

//----- PROCEDURE D'INITIALISATION -----
void setup() {
    Servo1.attach(6); // Servo1 rattaché à la broche 6
}

//----- BOUCLE PRINCIPALE -----
void loop(){
    valPot = analogRead(Pot); // Lecture du POTENTIOMETRE
    angle = map(valPot, 0, 1023, 0, 180); // conversion
    Servo1.write(angle); // Activation du SERVOMOTEUR
    delay(20); // Temps nécessaire au mouvement du palonnier
}
```



CODE ARDUINO

```
// Croquis P7-Servo-calibration.ino

#include <Servo.h> // Librairie pour l'utilisation de servomoteurs

int temps = 2200; //censé être à mi-chemin entre 1000 et 2000, un bon point de départ

Servo monServo;

void setup()
{
    Serial.begin(115200);
    Serial.println("Hello World");

    monServo.attach(6);
    //on démarre à une valeur censé être la moitié de
    //l'excursion totale de l'angle réalisé par le servomoteur
    monServo.writeMicroseconds(temps);
}

void loop(){
    //des données sur la liaison série ? (lorsque l'on appuie sur 'a' ou 'd')
    if(Serial.available())
    {
        char commande = Serial.read(); //on lit

        //on modifie la consigne si c'est un caractère qui nous intéresse
        if(commande == 'a')
            temps += 10; //ajout de 10 microsecondes au temps HAUT
        else if(commande == 'd')
            temps -= 10; //retrait de 10 microsecondes au temps HAUT

        //on modifie la consigne du servo
        monServo.writeMicroseconds(temps);

        //et on fait un retour sur la console
        Serial.println(temps, DEC);
    }
}
```



CODE ARDUINO

```
// Croquis P7-Servo_Q1.ino

//----- LIBRAIRIE ADDITIONNELLE -----
#include <Servo.h> // Librairie pour l'utilisation de servomoteurs

//----- CONSTANTES -----
Servo Servo1; // création de l'objet Servo1 (notre SERVOMOTEUR)
#define Pot 0 // POTENTIOMETRE sur la ligne analogique 0 (A0)
#define Led 5 // LED sur la ligne E/S 5

//----- VARIABLE -----
int valPot; // Variable du POTENTIOMETRE
int angle; // valPot converti en angle (entre 0 et 180 degrés)

//----- PROCEDURE D'INITIALISATION -----
void setup() {
    Servo1.attach(6); // Servo1 rattaché à la broche 6
    pinMode(Led, OUTPUT);
}

//----- BOUCLE PRINCIPALE -----
void loop(){
    valPot = analogRead(Pot); // Lecture du POTENTIOMETRE
    angle = map(valPot, 0, 1023, 0, 180); // conversion
    if (angle > 90){
        digitalWrite(Led, HIGH);
    }
    else {
        digitalWrite(Led, LOW);
    }
    Servo1.write(angle); // Activation du SERVOMOTEUR
    delay(20); // Temps nécessaire au mouvement du palonnier
}
```



CODE ARDUINO

```
// Croquis P7-Servo_Q2.ino

//----- LIBRAIRIE ADDITIONNELLE -----
#include <Servo.h> // Librairie pour l'utilisation de servomoteurs

//----- CONSTANTES -----
Servo Servo1; // création de l'objet Servo1 (notre SERVOMOTEUR)
#define Led 5 // LED sur la ligne E/S 5

//----- VARIABLE -----
int angle; // valPot converti en angle (entre 0 et 180 degrés)

//----- PROCEDURE D'INITIALISATION -----
void setup() {
    Servo1.attach(6); // Servo1 rattaché à la broche 6
    pinMode(Led, OUTPUT);
}

//----- BOUCLE PRINCIPALE -----
void loop(){
    for (angle = 0; angle <= 180; angle += 2){
        if (angle > 90){
            digitalWrite(Led, HIGH);
        }
        else {
            digitalWrite(Led, LOW);
        }
        Servo1.write(angle); // Activation du SERVOMOTEUR
        delay(20); // Temps nécessaire au mouvement du palonnier
    }

    for (angle = 180; angle >= 0; angle -= 2){
        if (angle > 90){
            digitalWrite(Led, HIGH);
        }
        else {
            digitalWrite(Led, LOW);
        }
        Servo1.write(angle); // Activation du SERVOMOTEUR
        delay(20); // Temps nécessaire au mouvement du palonnier
    }
}
```

