

Siradjo Sy
Louanes Abboud
Gougui Mohamed

TP SERVOMOTEUR

Siradjo SY, Louanes ABBOUD, Gougui Mohamed

Siradjo Sy
Louanes Abboud
Gougui Mohamed

Table des matières

Contrôle d'un Servo SG92R avec Joystick et Interface Python.....	3
🎯 Objectif général.....	3
🔧 Matériel utilisé.....	3
⚙️ Branchement matériel	3
I-) CONTROLE de servo moteur.....	4
A-) Test de base du servo Moteur	4
B-) Observation des trams PMW sur un oscilloscope.....	5
C-) programme complet Python + Arduino qui permet de contrôler le servo-moteur SG92R avec un slider Tkinter en temps réel.....	6
D-) Programme permettant de contrôler un servo-moteur SG92R à l'aide d'un joystick analogique, en traduisant les mouvements horizontaux en signaux de rotation. Le système permet un contrôle précis et en temps réel du moteur via une carte Arduino Uno, avec option d'affichage d'un slider via une interface Python, le tout grâce a une communication en serie.....	9
📊 Résultats obtenus.....	11

Siradjo Sy
Louanes Abboud
Gougui Mohamed

Contrôle d'un Servo SG92R avec Joystick et Interface Python

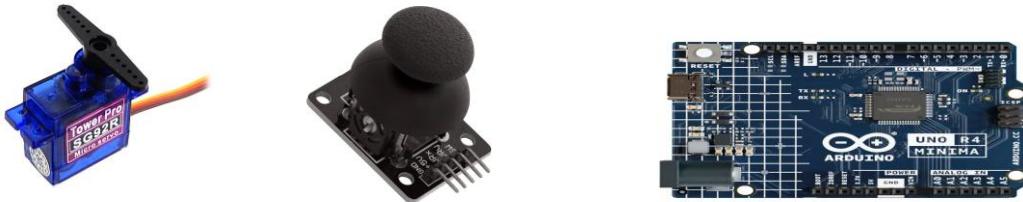
⌚ Objectif général

Développer un système interactif de contrôle d'un servo-moteur SG92R à rotation continue, intégrant :

- Un joystick analogique pour piloter manuellement,
- Une interface graphique Python (curseur Tkinter) pour visualiser et contrôler le mouvement,
- Une communication série temps réel entre un PC et une carte Arduino Uno.

🔧 Matériel utilisé

- Servo-moteur SG92R (rotation continue)
- Joystick analogique KY-023
- Câbles Dupont
- PC avec Python + Tkinter + PySerial
- Oscilloscope (facultatif)
- IDE Arduino 1.8.16



⚙️ Branchement matériel

🎲 Joystick

- GND → GND
- VCC → 5V
- VRx → A0
- VRy → A1 (non utilisé)

🤖 Servo SG92R

Siradjo Sy
Louanes Abboud
Gougui Mohamed

- Rouge (+) → 5V
- Marron/GND → GND
- Orange (Signal) → D9

⚠ Si le servo demande plus de courant que l'Arduino peut fournir, utiliser une alimentation externe 5V (en reliant les GND ensemble).

I-) CONTROLE de servo moteur

A-) Test de base du servo Moteur

```
#include <Servo.h>

Servo monServo;

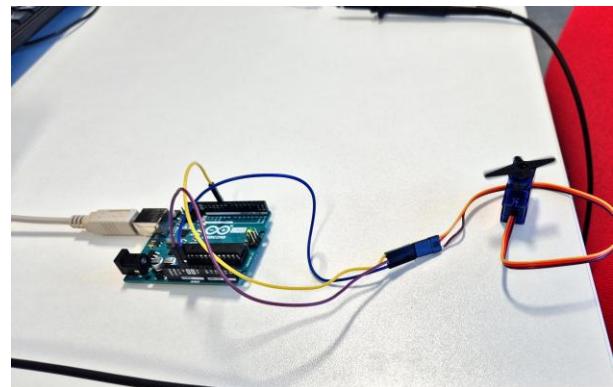
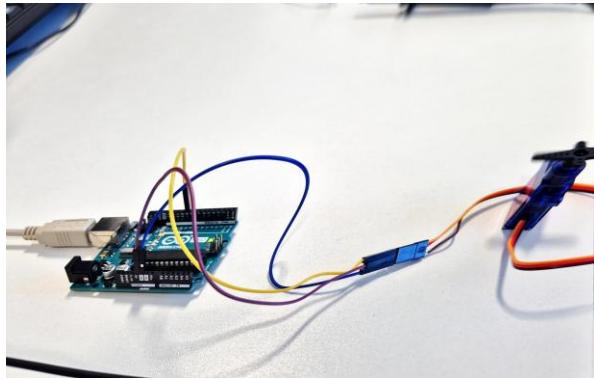
void setup() {
    monServo.attach(9); // Broche de signal du servo (SG92R)
}

void loop() {
    // --- État 1 : ARRÊT ---
    monServo.write(90); // Signal d'arrêt pour servo à rotation continue 1500 ms
    delay(3000); // Attendre 3 secondes

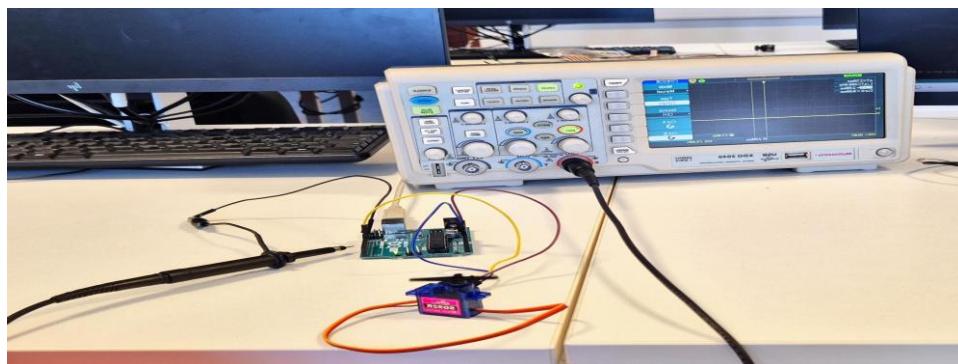
    // --- État 2 : MARCHE AVANT ---
    monServo.write(180); // Rotation dans un sens (avant) 2500 ms
    delay(3000); // Attendre 3 secondes

    // --- État 3 : MARCHE ARRIÈRE ---
    monServo.write(0); // Rotation dans l'autre sens (arrière) 500 ms
    delay(3000); // Attendre 3 secondes
}
```

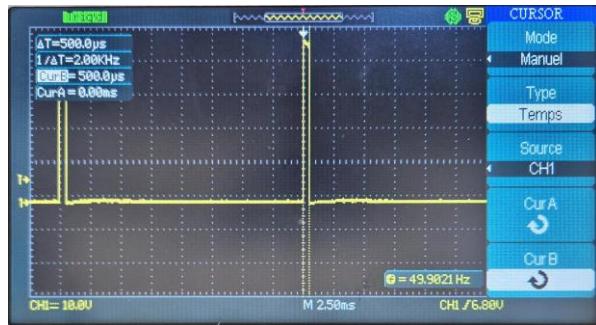
Siradjo Sy
 Louanes Abboud
 Gougui Mohamed



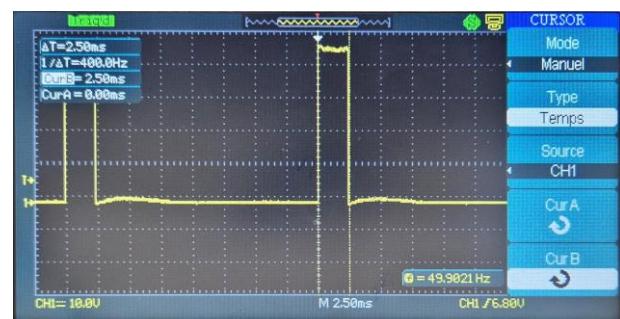
B-) Observation des trams PMW sur un oscilloscope



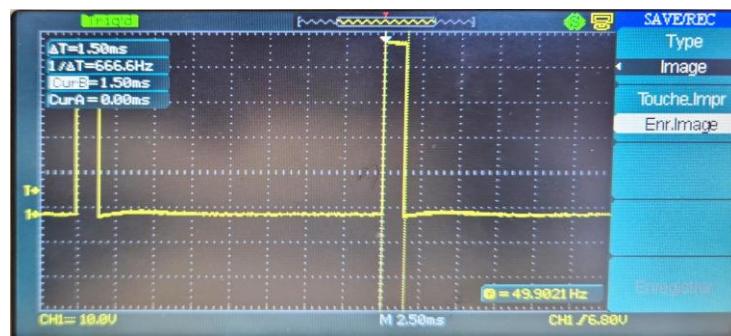
Delta t marche arrière avec 500 μ s



Delta t marche avant avec 2.5 ms



Delta t Arrêt avec 1.5 ms



Siradjo Sy
Louanes Abboud
Gougui Mohamed

C-) programme complet Python + Arduino qui permet de contrôler le servo-moteur SG92R avec un slider Tkinter en temps réel.

Code Python : Contrôle par slider uniquement

```
import tkinter as tk
import serial
import time

# Connexion au port série
try:
    arduino = serial.Serial('COM3', 9600, timeout=1) # Remplace COM3 si nécessaire
    time.sleep(2)
except:
    print("Erreur d'ouverture du port")
    arduino = None

# Création de la fenêtre Tkinter
fenetre = tk.Tk()
fenetre.title("Contrôle Servo via Slider")

# Création du slider
slider = tk.Scale(fenetre, from_=0, to=180, orient=tk.HORIZONTAL,
                  label="Position Servo", length=400)
```

Siradjo Sy
Louanes Abboud
Gougui Mohamed

```
slider.set(90)
slider.pack(padx=20, pady=20)
# Fonction appelée quand le slider change
def envoyer_commande(val):
    if arduino:
        commande = f"{val}\n"
        arduino.write(commande.encode())
# Mise à jour du servo lors du mouvement du slider
slider.config(command=envoyer_commande)

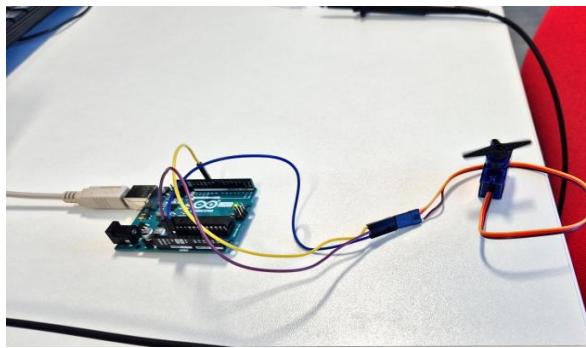
# Lancer l'interface
fenetre.mainloop()

- Code Arduino correspondant
#include <Servo.h>

Servo monServo;
String donnees = "";
void setup() {
    Serial.begin(9600);
    monServo.attach(9); // D9 connecté au fil orange (signal) du SG92R
}
void loop() {
    // Lecture série
```

Siradjo Sy
Louanes Abboud
Gougui Mohamed

```
while (Serial.available()) {  
  
    char c = Serial.read();  
  
    if (c == '\n') {  
  
        int angle = donnees.toInt(); // Conversion string → int  
  
        monServo.write(angle); // Positionner le servo  
  
        donnees = ""; // Réinitialiser la chaîne  
  
    } else {  
  
        donnees += c; // Construction du message  
  
    }  
  
}  
  
}
```



D-) Programme permettant de contrôler un servo-moteur SG92R à l'aide d'un joystick analogique, en traduisant les mouvements horizontaux en signaux de rotation. Le système permet un contrôle précis et en temps réel du moteur via une carte Arduino Uno, avec option d'affichage d'un slider via une interface Python, le tout grâce à une communication en série

1. Contrôle avec le joystick

Code arduino

```
#include <Servo.h> // Inclusion de la bibliothèque Servo

Servo monServo; // Création d'un objet Servo

const int joyX = A0; // Broche analogique utilisée pour lire l'axe X du joystick

void setup() {

    Serial.begin(9600); // Initialisation de la communication série (pour Python)

    monServo.attach(9); // Le servo est connecté à la broche numérique D9

}

void loop() {

    int val = analogRead(joyX); // Lire la position horizontale du joystick (0 à 1023)

    int angle = map(val, 0, 1023, 0, 180); // Convertir la valeur lue en angle (0° à 180°)

    Serial.println(angle); // Envoyer la valeur de l'angle au port série (pour affichage ou interface Python)

    monServo.write(angle); // Appliquer l'angle au servo pour le faire tourner

    delay(50); // Pause de 50 ms pour lisser le signal (éviter les secousses)

}
```

2. **Code Python complet pour afficher la position du joystick lue depuis l'Arduino via le port série, contrôler un slider graphique, et renvoyer la commande au servo**

Siradjo Sy
Louanes Abboud
Gougui Mohamed

Code Python

```
import tkinter as tk

import serial

import threading

import time

# Connexion au port série

try:

    arduino = serial.Serial('COM3', 9600, timeout=1)

    time.sleep(2) # Laisser le temps à l'Arduino de redémarrer

except:

    print("Erreur d'ouverture du port série")

    arduino = None


# Création de la fenêtre principale

fenetre = tk.Tk()

fenetre.title("Contrôle Servo avec Joystick")


# Slider graphique pour afficher la valeur lue

slider = tk.Scale(fenetre, from_=0, to=180, orient=tk.HORIZONTAL,

                   label="Position du Servo (par Joystick)", length=400)

slider.set(90)

slider.pack(padx=20, pady=20)
```

Siradjo Sy
Louanes Abboud
Gougui Mohamed

```
# Fonction exécutée en parallèle pour lire les données série

def lire_joystick():

    while True:

        if arduino and arduino.in_waiting:

            try:

                ligne = arduino.readline().decode().strip()

                if ligne.isdigit():

                    valeur = int(ligne)

                    if 0 <= valeur <= 180:

                        slider.set(valeur)

                        # (Optionnel) renvoyer la commande à l'Arduino

                        arduino.write(f"{valeur}\n".encode())

            except Exception as e:

                print(f"Erreur lecture série : {e}")

# Lancer le thread en arrière-plan

threading.Thread(target=lire_joystick, daemon=True).start()

# Démarrer l'interface

fenetre.mainloop()
```

💡 Résultats obtenus

- Le joystick envoie des valeurs analogiques à Arduino
- Les valeurs sont affichées dynamiquement dans un curseur Tkinter

Siradjo Sy
Louanes Abboud
Gougui Mohamed

- Le curseur renvoie les données vers Arduino
- Le servo-moteur tourne selon la position du joystick

