# Interfacing a Servo Motor with Arduino Nano

Will guide you to interface the Arduino nano with Servo motor.

Difficulté **Facile** | Durée **2 heure(s)** | Catégories **Électronique** | Coût **10 USD ($)**

## Sommaire



# Introduction

Servo motors are widely used in robotics, automation, and DIY projects because they allow precise control of angular position. The Arduino Nano, being compact and versatile, is perfect for controlling servos in space-constrained projects. In this tutorial, we'll learn how to connect and program a servo motor with the Arduino Nano

## Matériaux

## Outils

# Étape 1 - Components Required

- **Arduino Nano** (official or compatible board)
- **Servo Motor** (e.g., SG90 micro servo)
- **Jumper wires**
- **USB cable (Mini-B)** for programming

# Étape 2 - Get PCBs for Your Projects Manufactured


You must check out PCBWAY for ordering PCBs online for cheap! You get 10 good-quality PCBs manufactured and shipped to your doorstep for cheap. You will also get a discount on shipping on your first order. Upload your Gerber files onto PCBWAY to get them manufactured with good quality and quick turnaround time. PCBWay now could provide a complete product solution, from design to enclosure production. Check out their online Gerber viewer function. With reward points, you can get free stuff from their gift shop. Also, check out this useful blog on PCBWay Plugin for KiCad from here. Using this plugin, you can directly order PCBs in just one click after completing your design in KiCad.

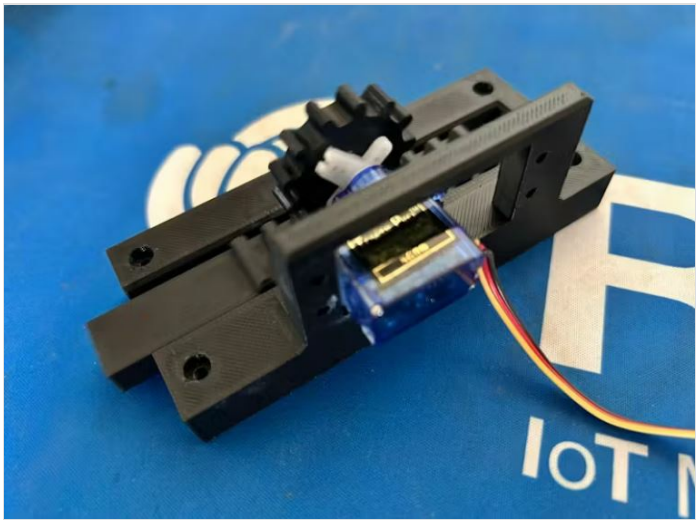
# Étape 3 - Understanding the Servo Motor

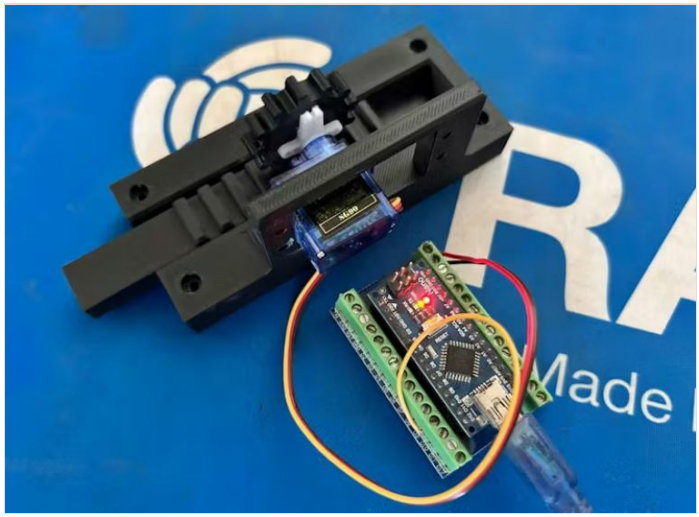
**Wires:**

- **Red** → VCC (typically 5V)
- **Brown/Black** → GND
- **Orange/Yellow** → Signal (PWM input)

**Working Principle:** The servo motor rotates to a specific angle (0°–180°) based on the PWM signal sent from the Arduino
Circuit Connections

- Connect **servo VCC (red)** to **5V pin** of Arduino Nano
- Connect **servo GND (black/brown)** to **GND pin** of Arduino Nano.
- Connect **servo signal (orange/yellow)** to **digital pin D9** of Arduino Nano (you can choose other PWM-capable pins).
- If using a high-torque servo, power it with an **external 5V supply** and connect grounds together to avoid damaging the Nano





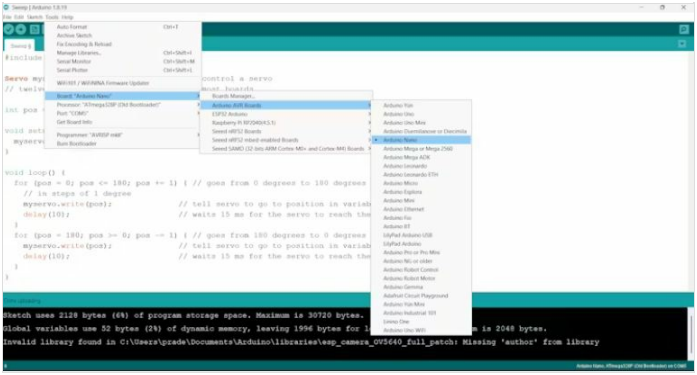
# Étape 4 - Installing the Servo Library

The Arduino IDE comes with a built-in **Servo.h library**.

- Open Arduino IDE → Go to **Sketch > Include Library > Servo**.
- This library simplifies sending PWM signals to the servo.

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 d
egrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'p
os'
    delay(10);                       // waits 15 ms for the servo to reach the posit
ion
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 de
grees
    myservo.write(pos);              // tell servo to go to position in variable 'p
os'
    delay(10);                       // waits 15 ms for the servo to reach the posit
ion
  }
}
```

# Étape 5 - Uploading the Code



- Connect Arduino Nano to your PC via USB.
- Select **Tools > Board > Arduino Nano**.
- Choose the correct **Port**.
- Click **Upload**.
- The servo should start sweeping back and forth.

# Étape 6 - Testing and Troubleshooting

- If the servo **jitters**, provide external power.
- Ensure **common ground** between Arduino and external supply.
- If the servo doesn't move, check wiring and pin assignment.

`myServo.write(angle)` `myServo.write(90)` Use to set specific angles (e.g., for center position).

## Expanding the Project

- Control servo with a **potentiometer** for manual angle adjustment.
- Use **ultrasonic sensors** to make the servo respond to distance.
- Integrate with **Bluetooth/Wi-Fi modules** for remote control.

# Étape 7 - Conclusion

By following these steps, you've successfully interfaced a servo motor with an Arduino Nano. This forms the foundation for robotics arms, pan-tilt camera systems, and countless automation projects.