SERVO MOTOR

When it comes to choosing the right motor for a project, the options can be overwhelming—stepper motors, DC motors, brushless motors, and more. Each type has its own strengths. But if you need precise control over movement, there's one clear choice: the servo motor. Tell a servo motor where to point, and it will go there exactly! It's as simple as that!

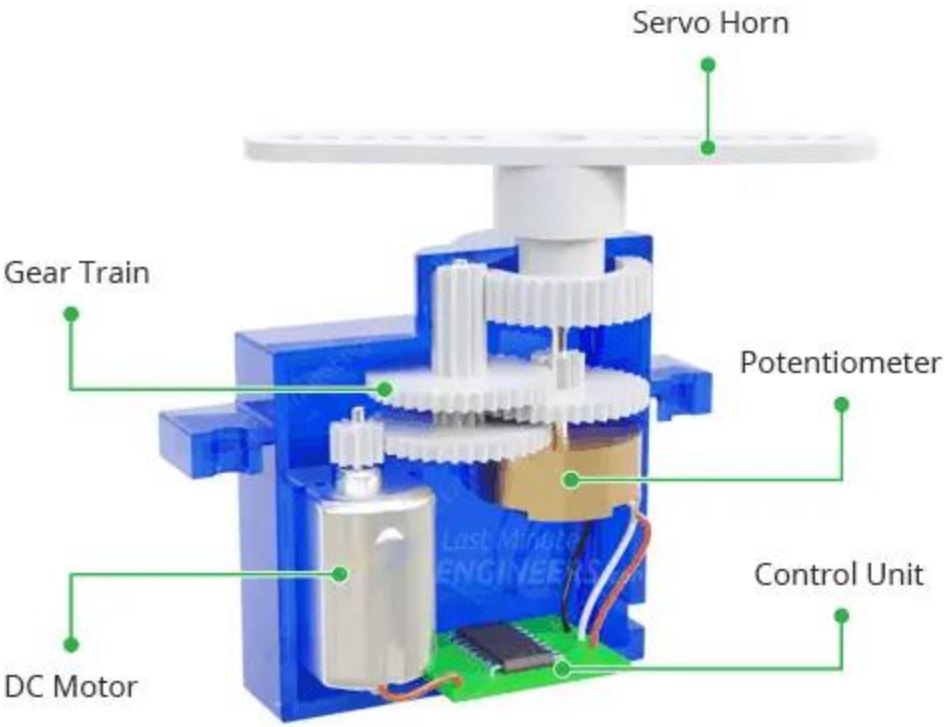
Servo motors first became popular in the Remote Control (RC) world, where they were used to control the steering of RC cars or the flaps on RC planes. Over time, they found their way into robotics, automation, and of course, the Arduino world.

In this guide, you'll learn how a servo works, how to connect it to an Arduino, and how to control its position using simple Arduino code. Whether you're building a robotic arm, an automated door, or just experimenting with motion control, this tutorial will help you get started.

Let's dive in and get that servo moving!

WHAT IS A SERVO AND WHAT MAKES IT PRECISE

Inside a typical hobby servo motor, you'll find five main components working together: a DC motor, a gearbox, a servo horn, a potentiometer, and a control unit.



DC Motor: This is the core component providing the rotational force. It's similar to the hobby motors found in toys, but how it's controlled makes all the difference. The DC motor connects to the output shaft through a series of gears.

Gearbox: The gearbox (a system of gears) plays a crucial role in increasing the motor's torque and improving the precision of its movements. It reduces the speed of the motor but increases its strength.

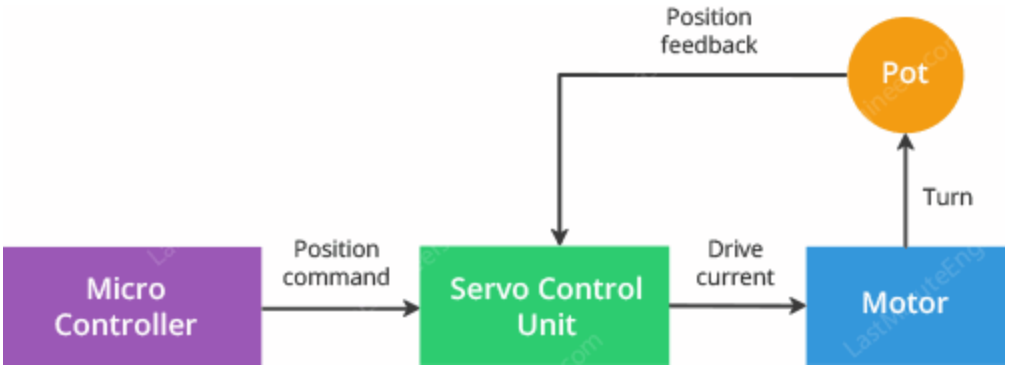
Servo Horn: This is a plastic arm (or wheel) attached to the output shaft (the part that actually rotates). The servo horn gives you a place to attach whatever you want the servo to move – like the arm of a robot or the rudder on a model airplane.

Potentiometer: This is a variable resistor that acts as a position sensor. It's connected directly to the output shaft. As the output shaft rotates, the potentiometer's resistance changes. This tells the control unit exactly where the motor is positioned.

Control Unit: This is the "brain" of the servo motor. It receives signals from an external controller (like an Arduino), checks the potentiometer's position, and controls the DC motor.

HERE'S HOW THESE PARTS WORK TOGETHER:

The servo motor works using what's called a "closed-loop feedback system".



When you send a signal to the servo telling it to move to a specific position, the control unit receives this command.

The potentiometer is physically connected to the output shaft, so it always knows the current position of the servo. When the motor rotates, the output shaft turns, and so does the potentiometer. This rotation changes the potentiometer's resistance, creating a voltage that directly corresponds to the current position of the output shaft (and thus the motor). This voltage serves as feedback to the control unit.

The control unit constantly compares this feedback voltage (representing the motor's current position) with the desired position signal from the microcontroller.

If there's a difference (an "error" signal) between the desired position and the current position, the control unit adjusts the power and direction of the DC motor to fix this difference.

As the motor moves, the potentiometer's feedback voltage changes, continuously updating the control unit about the motor's current position.

Once the motor reaches the desired position you wanted, the error signal becomes zero, and the control unit stops the motor.

This whole process happens very quickly and repeatedly. The servo is always checking its position and making tiny adjustments to stay exactly where it should be. This is why servos are so good at holding their position, even when external forces try to move them.

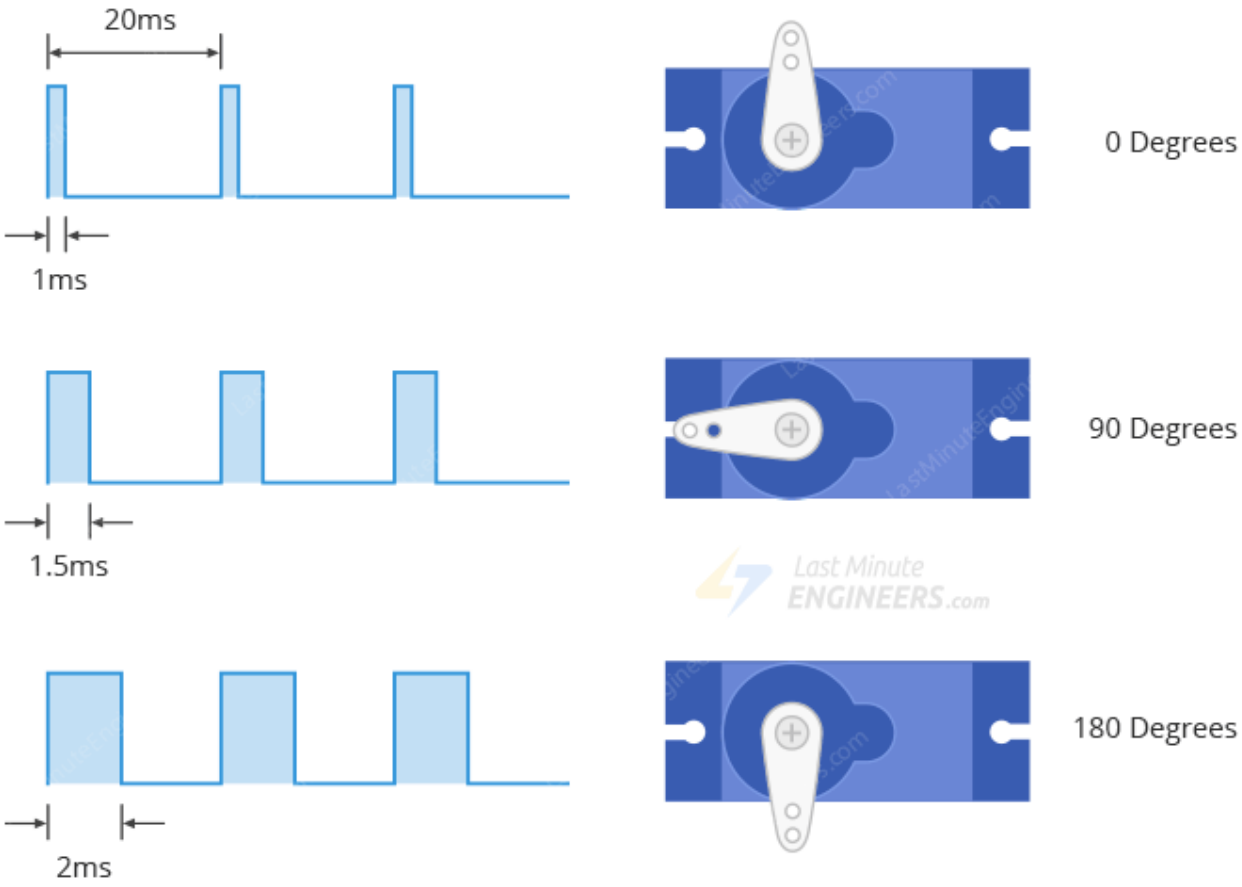
This entire system is called a "servomechanism" or simply a "servo." It works as a closed-loop control system, using negative feedback to precisely control the motor's speed and direction, allowing it to reach and maintain a specific position.

HOW DO SERVO MOTORS WORK

Hobby servo motors are controlled using a clever technique called Pulse Width Modulation (PWM).

PWM works by sending a series of electrical pulses to the servo at regular intervals. For most hobby servos, these pulses are sent about 50 times per second (50 Hz). This means a new pulse arrives every 20 ms (1/50th of a second).

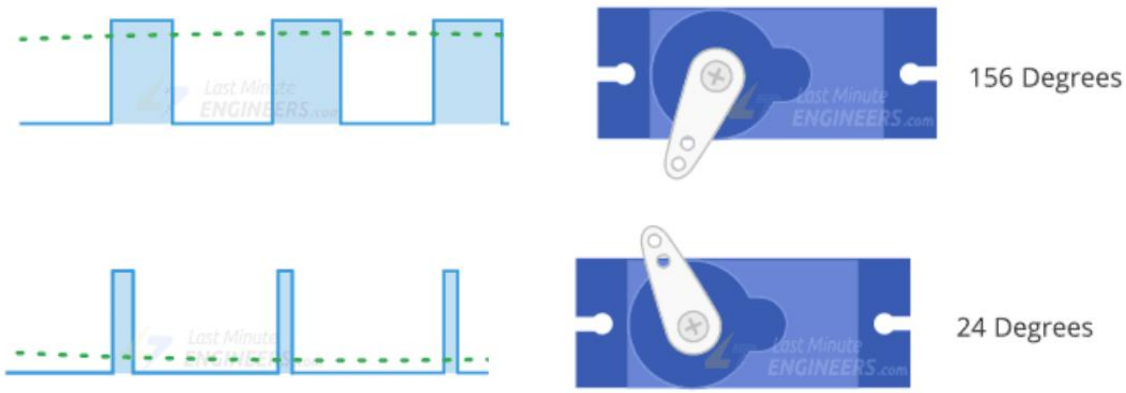
What really matters for controlling the servo's position isn't how often we send pulses, but how long each pulse lasts. This duration is called the "pulse width" (or "duty cycle" of the PWM signal, if you prefer the technical term).



- A short pulse around 1 ms or less tells the servo to move to 0 degrees (all the way to one side)
- A medium pulse of about 1.5 ms positions the servo at 90 degrees (right in the middle)
- A longer pulse of around 2 ms moves the servo to 180 degrees (all the way to the other side)
- For positions in between, we use pulse widths that fall between these values. For example, if we want the servo at the 45-degree position (halfway between 0 and 90 degrees), we would use a pulse that's about 1.25 ms long.

This means we can precisely control exactly where the servo points by carefully controlling how long each pulse lasts. The servo's control unit receives these pulses, measures their width, and moves the motor to the corresponding position.

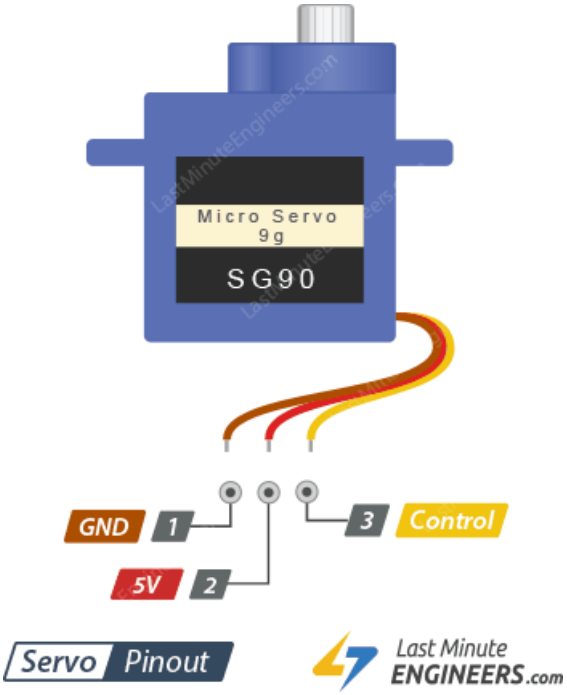
The animation below can help you visualize how changes in pulse width correspond to different servo positions.



SERVO MOTOR PINOUT

Almost all hobby servos come with a standard three-pin, 0.1"-spaced connector. While the color coding of the wires may vary between brands, the pins are generally arranged in the same order.

Here's a breakdown of the pinout:



GND is the ground pin.

5V pin supplies power to the servo motor. Most hobby servos need between 4.8V and 6V to operate properly. Providing the right voltage is important – too little and the servo won't have enough strength, too much and you might damage it.

Control receives the PWM (Pulse Width Modulation) signal from a microcontroller, which determines the servo's position.

WIRING SERVO MOTOR TO ARDUINO UNO

Alright! Let's connect a servo motor to an Arduino UNO and see how it works.

For this experiment, we'll use an SG90 Micro Servo Motor. This small but powerful motor runs on 5V (can work between 4.8V to 6V DC) and can rotate up to 180 degrees—90 degrees in each direction.

It's important to note that when the servo is sitting still, it only needs about 10mA of current. But when it's moving, it needs much more—between 100mA and 250mA. This means that for most simple projects, we can power the servo directly from the Arduino's 5V pin. But if your servo needs more than 250mA, you should use a separate power supply to avoid damaging your Arduino.

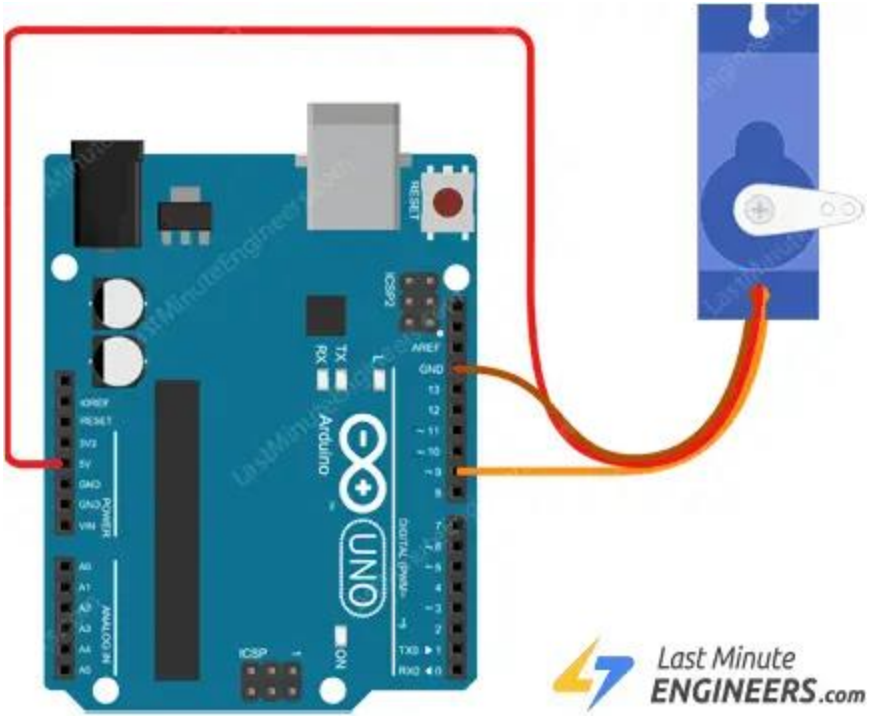
To connect the SG90 servo motor to the Arduino:

- Connect the red wire to the 5V pin on the Arduino.
- Connect the black or brown wire to the GND (ground) pin.
- Connect the orange or yellow wire to pin #9 on the Arduino. This pin is PWM-enabled, meaning it can send the Pulse Width Modulation (PWM) signal needed to control the servo's movement.

Here's a quick reference table for the pin connections

Servo Motor	Arduino
5V	5V
GND	GND
Control	9

Please refer to the image below to see the proper wiring setup.

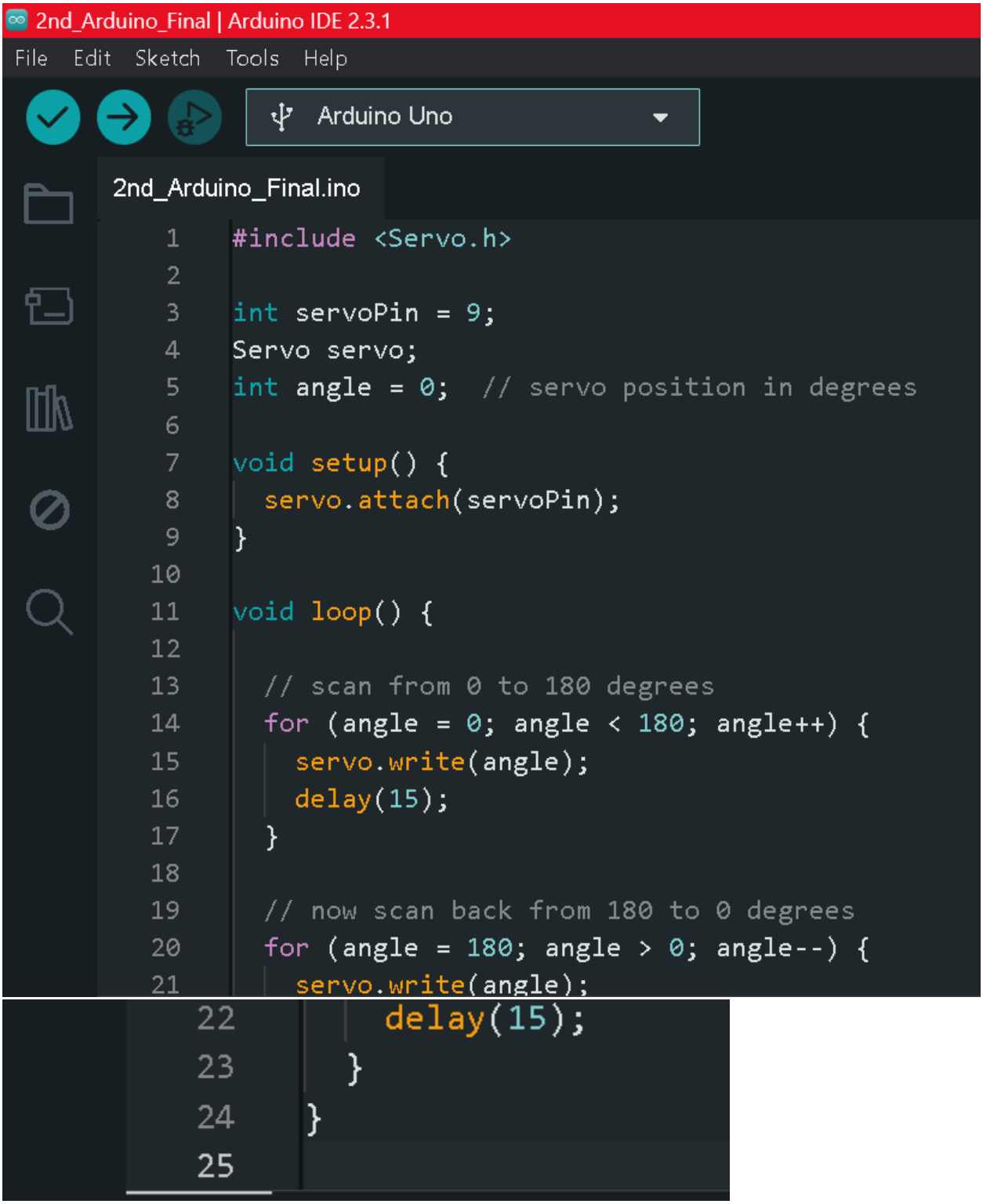


Once everything is connected, you're ready to control the servo motor using Arduino code!

CODE

Let's start by using one of the built-in examples from the Arduino IDE. Open the Examples menu, navigate to Servo, and select the Sweep sketch.

Once the code is loaded, upload it to your Arduino board. If everything is set up correctly, you'll see the servo motor's shaft sweeping back and forth across 180 degrees.



REFERENCE:

https://lastminuteengineers.com/servo-motor-arduino-tutorial/