# VIDEO#25: Electronic Basics #25: Servos and how to use them

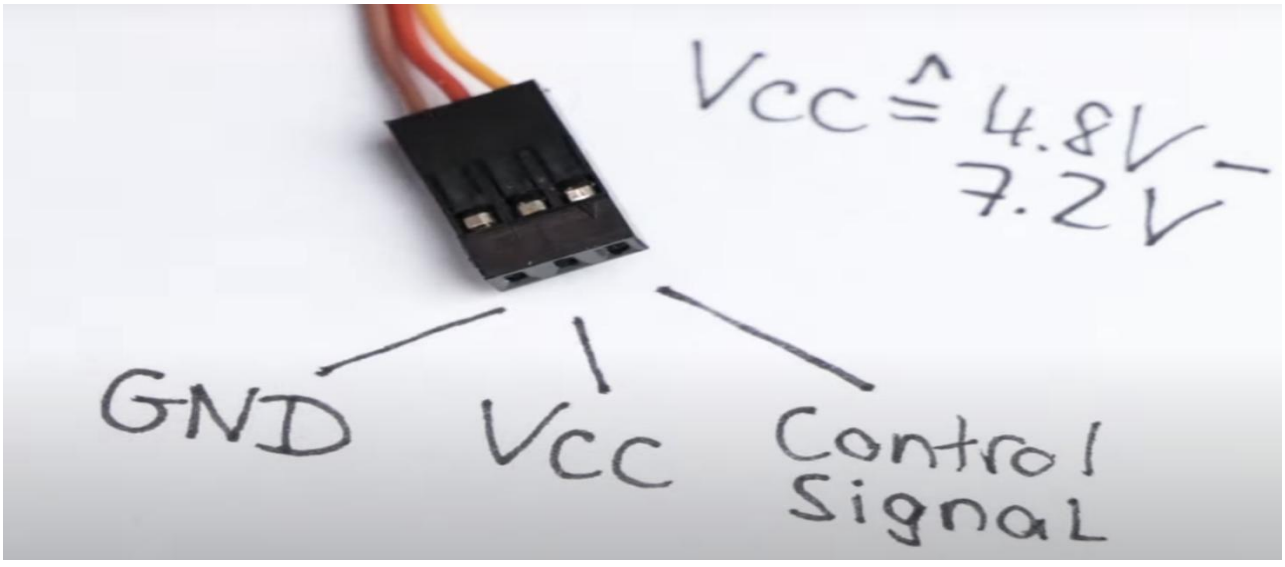
## Servo Motor and Its Working Principle

**Introduction**

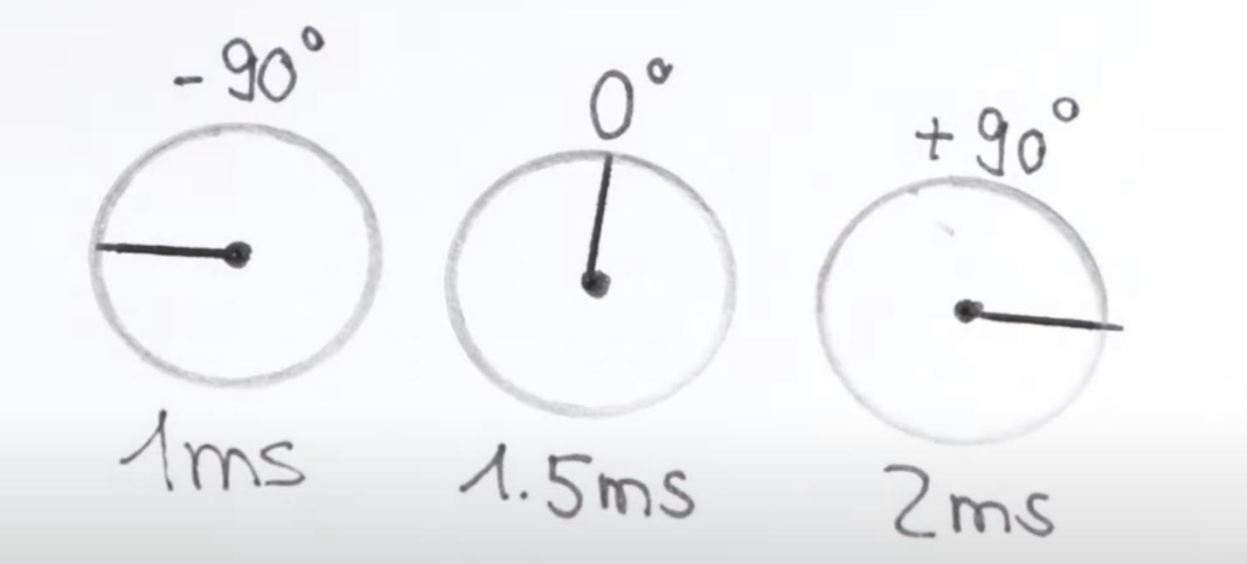
When positioning tasks in a project, stepper motors are commonly used. Another popular motor-driven positioning system is the **Servo motor**, which integrates the motor with control electronics in an easy-to-use package. This lecture explains how a servo motor works and how to use one in a project with or without a microcontroller.

**Common Servo Motor Features**

Most servo motors have a **sawtooth shaft** on top for attaching accessories and three wires: **Brown** for Ground (GND), **Red** for VCC (Voltage supply: 4.8V – 7.2V), and **Orange** for the control signal (PWM signal).




The control signal requires a **PWM signal** with a periodic time of **20 milliseconds (50Hz)** and a duty cycle of **1 to 2 milliseconds (5% to 10%)**. An **on-time of 1 ms** represents a **-90° position**, **1.5 ms represents 0°**, and **2 ms represents +90°**, allowing a total rotation of **180°**.




## Internal Structure of a Servo Motor

Opening the servo reveals key components such as a **gear system**, which reduces the **DC motor's RPM from ~3000 to ~50-100 RPM**, increasing torque. A **potentiometer** acts as a feedback system by creating a voltage divider, helping track the position of the motor shaft. The **control IC (e.g., KC5188)** constantly compares the potentiometer's voltage (current position) with the **PWM signal (target position)** and activates an **H-Bridge** to rotate the motor until the difference is zero.

**Servo Motor Upgrades**

Higher torque and performance servos, like the **MG996R Servo**, feature **metal gears**, a **bigger motor**, and a control IC with **N-channel and P-channel MOSFETs** for better current handling. These upgrades improve durability and efficiency in demanding applications.

**Controlling a Servo Motor**

A servo motor can be controlled using an **Arduino** or a **555 Timer IC**. With an **Arduino**, the **Servo library** can generate the necessary PWM signal. A **potentiometer** connected to an analog input can adjust the signal between **1 ms and 2 ms**, though some servos require a wider range (0.5 ms to 2.5 ms) for a full **180° rotation**.

Alternatively, a **555 Timer IC** circuit with **two resistors, one diode, two capacitors, and a potentiometer** can generate a PWM signal. The **on-time varies between 0.5 ms and 2.45 ms**, and the **off-time is approximately 17.5 ms to 18.5 ms**. While the generated signal may not have a perfect **20 ms periodic time**, the servo functions correctly without issue.

**Modifying a Servo for Continuous Rotation**

Standard servos rotate only **180°**, but they can be modified for **360° continuous rotation**. This involves removing the **mechanical end stop** from the gear system and replacing the **feedback potentiometer** with **two 10kΩ resistors** to create a fixed voltage divider. This modification tricks the control IC into thinking the motor is always at **0°**. By adjusting the PWM signal, **below 1.4 ms results in clockwise rotation, while above 1.4 ms results in counterclockwise rotation**. However, since feedback is lost, the motor rotates continuously without stopping at a specific position. Pre-modified continuous rotation servos are also available for purchase.

Servo motors are versatile and can be used for **precise positioning or continuous rotation**. They can be controlled using **Arduino, a 555 Timer IC, or other circuits**. **Upgraded servos** with metal gears and high torque are available for higher performance. Modifying servos for continuous rotation is possible, but pre-modified versions are also an option.