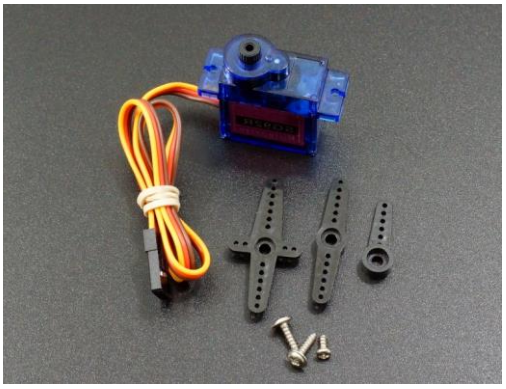
AA171 SERVO MICRO SG92R 2.5kg 4.8V

DESCRIPTION

The Servo Motor Micro SG92R is an upgraded version of the ubiquitous SG90 micro servo with improved gears and torque.

PACKAGE INCLUDES:

- 1 SG92R Servo motor with attached 9.5" control cable
- 3 arms/horns for various interface applications
- Screws for mounting arms to the servo and mounting the servo



ECHNICAL SPECIFICATIONS

Motor Model		Generic SG92R
Drive Type		Analog
Degree Rotation		180° (±15°)
Operating Ratings		
	Voltage	4.8-6VDC (5V Typical)
	Current (idle)	10mA (typical)
	Current (typical during movement)	100-300mA
	Current (stall)	650mA (measured)
	Stall Torque	2.5 kg-cm
	Speed	0.10s / 60 degree (varies with VDC)
Dimensions		
	Cable Length	24cm (9.5")
	Motor Housing L x W x H	23 x 12.2 x 27mm (0.9 x 0.48 x 1.06")
	Motor Height (w/ shaft)	32mm (1.26")

	Motor Housing Width with Mounting Ears	32.5mm (1.28")

KEY FEATURES OF SERVO MOTOR MICRO SG92R:

- Very small micro size
- POM with carbon fiber gears
- Stall torque up to 2.5 kg-cm
- 180 degree rotation
- Analog drive

The Servo Motor Micro SG92R work well for basic servo experimentation and can be used in applications where small size is a virtue and that don't require a huge amount of torque, but they are still pretty strong. Gears are POM plastic with carbon fiber particles that improve strength and durability.

Servo motors can be commanded to go to a specific position and so are the usual go-to motor when accurate positioning is needed, such as for turning the front wheels on an RC model for steering or pivoting a sensor to look around on a robotic vehicle.

Servo motors are comprised of a DC motor, gears, a potentiometer to determine its position and a small electronic control board.

Standard servos have a specified limited range. This is usually specified as 180 degrees. Frequently the actual range is not quite the full 180 degrees and is limited by the mechanical gears and potentiometer used for position sensing that is contained in the device. If the motor is run all the way to 0 or 180, it may start making unhappy sounds and start vibrating as it tries to drive to a position that it cannot get to. This causes a high stall current condition and has the potential of stripping gears and damaging the motor, so it is best to either drive it to a safely reduced range such as 20-160 or experiment a bit to determine the actual usable range if you want to maximize the range.

Servos expect to see a pulse on their PWM pin every 20 mSec. The pulse is active HIGH and the width of the pulse determines the position (angle) of the servos shaft. The pulse can vary between 1mSec and 2mSec. A 1mSec pulse positions the shaft at 0 degrees. A 1.5mSec pulse positions the shaft at 90 degrees (centered in its range). A 2 mSec pulse positions the shaft at 180 degrees. Pulses with values between these can be used to position the shaft arbitrarily.

Motor Connections

The built-in cable has a 3-pin female connector that is usually mated with a standard 0.1" male header

1×3 Female Connector

- **Brown** = Ground
- **Red** = 5V
- **Orange** = PWM Signal

OUR EVALUATION RESULTS:

In our testing these servos can lift about 5.5lbs that is positioned on an arm 1cm out from the shaft , so they are fairly strong little motors. We also didn't have any issues with stripping the gears when pushed to their max.

The servo runs on 5V nominal with a current draw about 10mA at idle and 100mA to 300mA when being commanded to move depending on how it is being operated. Current draw can get up to a maximum of 650mA under a stall condition. One SG92R can typically be driven off the power pin of an Arduino when experimenting, but motors in general are electrically noisy and power hungry devices. It is always better to drive them directly off of a power supply rather than trying to power from the on-board Arduino regulator whenever possible

If you do decide to run it directly off the Arduino, you can help avoid most problems by running the power and ground from the Arduino over to a breadboard and then to the servo. By placing a fairly large electrolytic cap of around 470-1000uF across the power and ground on the breadboard, that will help to insulate the Arduino from some of the power surges of the motor.

The program below can be used to exercise a servo motor by using a potentiometer to set the position of the servo. This setup can also be used to determine the limits of the servos range by running the servo near its end-points and observing where it mechanically stops relative to the position command that is being issued. The constants MIN_VALUE and MAX_VALUE are used to set the 2 end-points in the program below.

Servo Motor Micro SG92R Test Program

```
/*

  Exercise Servo motor

  Use a potentiometer on pin A0 to command a servo attached to pi
n 9 to move to

  a specific position.  The Servo MIN_VALUE and MAX_VALUE can be
adjusted to

  avoid hitting the servo stops

  Uses built-in Servo.h library

*/

#include "Servo.h"

#define SERVO_PIN 9   // Can use any PWM pin

#define POT_PIN A0    // Can use any analog pin

#define MIN_VALUE 0   // Minimum Servo position

#define MAX_VALUE 180 // Maximum Servo position


Servo servo;          // creates servo object used to control the
servo motor

int value_pot = 0;    // Current value of the potentiometer

int value_servo = 0;  // Current servo position

int value_servo_old = 0; // Used to hold old servo value to look
for change.

//================================================================
================

//  Initialization

//================================================================
================

void setup()
```

```
{
  servo.attach(SERVO_PIN); // assigns PWM pin to the servo object
  Serial.begin (9600);     // Set Serial Monitor window comm spee
d
}
//================================================================
================
//  Main
//================================================================
================
void loop()
{
  value_pot = analogRead(POT_PIN); // Reads value of the potentio
meter. Return value = 0 to 1023
  value_servo = map(value_pot, 0, 1023, MIN_VALUE, MAX_VALUE); //
remap pot value to servo value
  if (value_servo != value_servo_old) {   // Only do something if
there's a change in the servo position
    servo.write(value_servo);         // Update servo position
    Serial.print("Pot Value: ");  // Update Serial Monitor window
with what's going on
    Serial.print(value_pot);
    Serial.print("tServo Value: ");
    Serial.println(value_servo);
    value_servo_old = value_servo;
    delay(25); // give servo time to move
  }
}
```

BEFORE THEY ARE SHIPPED, THESE MOTORS ARE:

- Inspected
- Basic operation of servo motor verified