Miniature Linear Servo Actuators - User Guide (Rev 1)

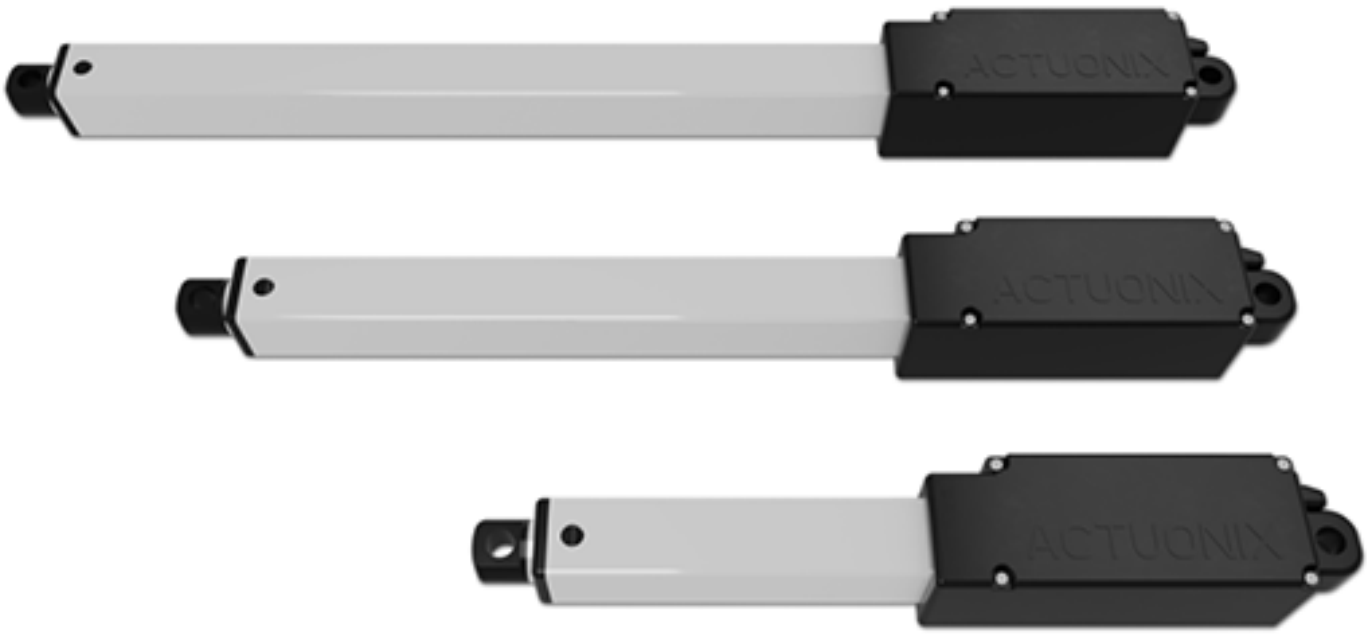


Table of Contents

What are the Miniature Linear Servo Actuators, L16 Series?

Actuonix Motion Devices' unique line of Miniature Linear Actuators enables a new generation of motion-enabled product designs, with capabilities that have never before been combined in a device of this size. These linear actuators are a superior alternative to designing your own push/pull mechanisms.

The L16 actuators are complete, self contained linear motion devices with position feedback for sophisticated position control capabilities, end of stroke limit switches for simple two position automation, or RC servo. Several gear ratios are available to give you varied speed/force configurations.

More information can be found by going to the Actuonix product page here:
https://www.actuonix.com/L16-R-Miniature-Linear-Servo-For-RC-p/l16-r.htm

Sizes Offered by WCP

WCP Part #	Actuonix Part #	Stroke	Ratio
WCP-0407	L16-50-35-6-R	50mm	35:1
WCP-0408	L16-100-35-6-R	100mm	35:1
WCP-0409	L16-140-35-6-R	140mm	35:1
WCP-0410	L16-50-63-6-R	50mm	63:1
WCP-0411	L16-100-63-6-R	100mm	63:1
WCP-0412	L16-140-63-6-R	140mm	63:1
WCP-0413	L16-50-150-6-R	50mm	150:1
WCP-0414	L16-100-150-6-R	100mm	150:1
WCP-0415	L16-140-150-6-R	140mm	150:1

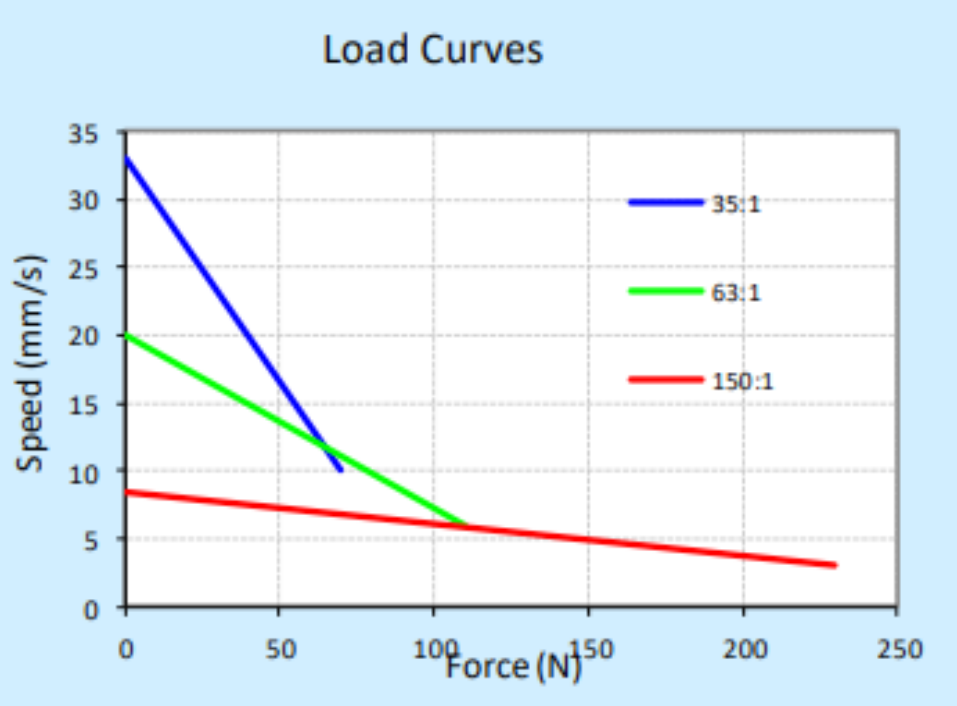
Note: All models are offered in the 6V DC option.

Specifications

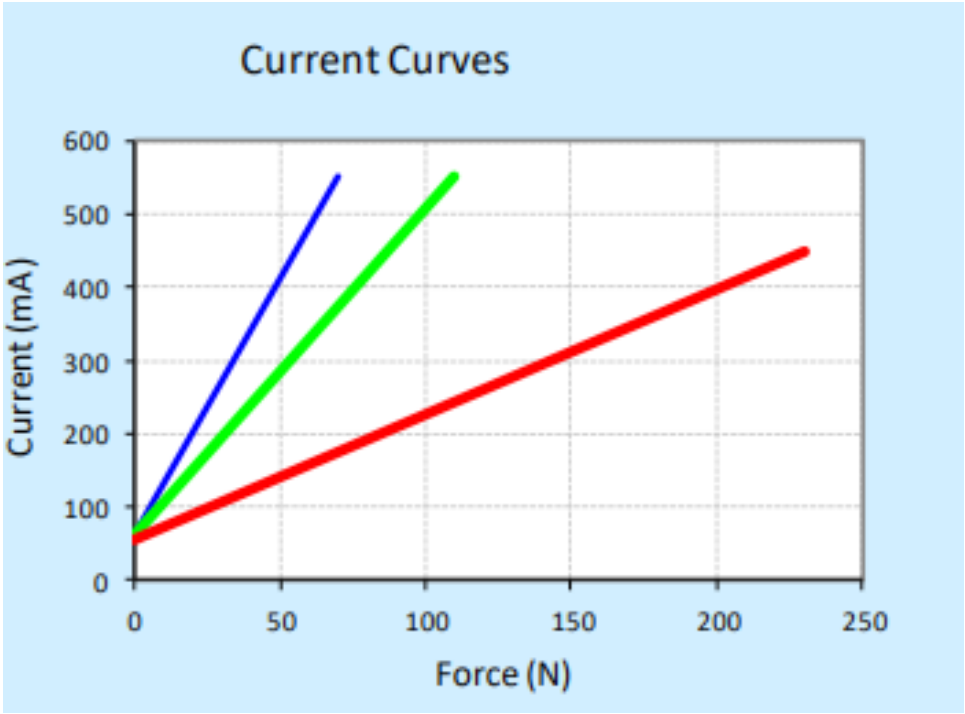
The L16 is designed to push or pull a load along its full stroke length. The speed of travel is determined by the load applied. (See the Load Curves). Actuator speed can be reduced by lowering the drive voltage. When power is removed the actuator will hold its position, unless the applied load exceeds the back drive force. Repeated stalling or stalling for more than a few seconds will shorten the life of the actuator significantly. Stalling is when an actuator is pushing a load that it cannot move. Actuators should be tested in each specific application to determine their effective life under those loading conditions and environment.

Gearing Option	**35:1**	**63:1**	**150:1**
Peak Power Point	50N @16mm/s	75N @10mm/s	175N @4mm/s
Peak Efficiency Point	24N @24mm/s	38N @15mm/s	75N @7mm/s
Max Speed (no load)	32mm/s	20mm/s	8mm/s
Max Force (lifted)	50N	100N	200N
Back Drive Force	31N	46N	102N
Stroke Option	**50mm**	**100mm**	**140mm**
Mass	56g	74g	84g
Max Side Load (extended)	40N	30N	20N
Closed Length (hole to hole)	118mm	168mm	208mm
Feedback Potentiometer	6kΩ+-50%	11kΩ+-50%	16kΩ+-50%
Feedback Linearity	Less than 2.00%		
Input Voltage	6VDC		
Stall Current	650mA @ 12V		
Operating Temperature	-10C to 50C		
Audible Noise	60dB @ 45cm		
Ingress Protection	IP-54		
Mechanical Backlash	0.25mm		
Maximum Static Force	250N		
Maximum Duty Cycle	20%		

Load Curves



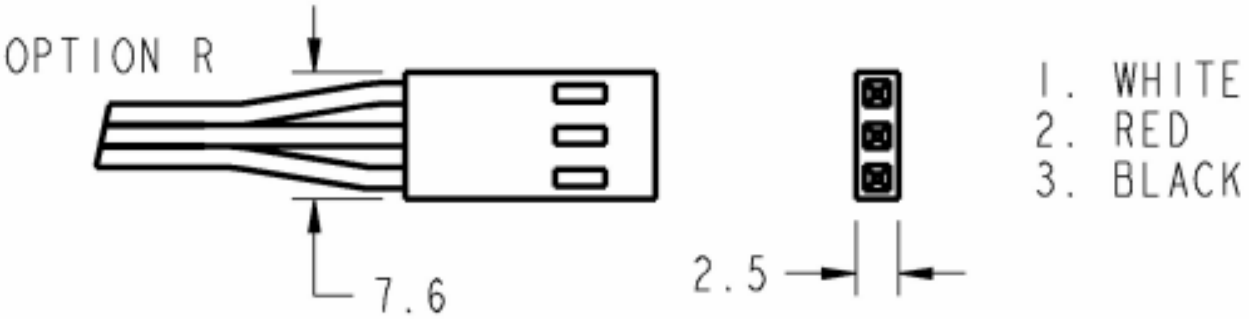
Current Curves



Option R - RC Linear Servo

WIRING: (see below for pin numbering)
1 - White – RC input signal (RC-servo compatible)
2 - Red – Power (+6 VDC)
3 - Black – Ground

Note: Reversing the polarity of pins 2 and 3 may permanently damage the actuator
-R actuators are ideally suited to use in robotics and radio control models. The –R actuators or 'linear servos' are a direct replacement for regular radio controlled hobby servos. The desired actuator position is input to the actuator on lead 1 as a positive 5 Volt pulse width signal. A 1.0 ms pulse commands the controller to fully retract the actuator, and a 2.0 ms pulse signals it to fully extend. If the motion of the actuator, or of other servos in your system, seems erratic, place a 1–4Ω resistor in series with the actuator's red V+ lead wire. L16 –R Linear Servos are the only 6 volt models in the L16 range because they are designed to work with typical RC receivers and battery packs. Consequently they also are compatible with Arduino control boards, VEX Microcontrollers and many other similar boards designed for robotics.



Included Hardware

Each actuator ships with two mounting brackets, #8-32 mounting hardware and male connector pins. The cable length is approximately 300mm and connector is a 0.1” pitch female socket connector.

Sample Code

Sample code provided by FRC 4414. The provided code is to control the position of the L16-R Miniature Linear Servo Actuators.

```
import edu.wpi.first.wpilibj.Servo;
import edu.wpi.first.wpilibj.Timer;
import edu.wpi.first.wpiutil.math.MathUtil;

public class LinearServo extends Servo{

    double m_speed;
    double m_length;

    double setPos;
    double curPos;

    /**
     * Parameters for L16-R Actuonix Linear Actuators
     *
     * @param channel PWM channel used to control the servo
     * @param length max length of the servo [mm]
     * @param speed max speed of the servo [mm/second]
     */
    public LinearServo(int channel, int length, int speed) {
        super(channel);
        setBounds(2.0, 1.8, 1.5, 1.2, 1.0);
        m_length = length;
        m_speed = speed;
    }

    /**
```

```
 * Run this method in any periodic function to update the position estimation of your servo
    *
    * @param setpoint the target position of the servo [mm]
    */
   public void setPosition(double setpoint){
      setPos = MathUtil.clamp(setpoint, 0, m_length);
      setSpeed( (setPos/m_length *2)-1);
   }

   double lastTime = 0;
   /**
    * Run this method in any periodic function to update the position estimation of your servo
    */
   public void updateCurPos(){
      double dt = Timer.getFPGATimestamp() - lastTime;
      if (curPos > setPos + m_speed *dt){
         curPos -= m_speed *dt;
      } else if(curPos < setPos - m_speed *dt){
         curPos += m_speed *dt;
      }else{
         curPos = setPos;
      }
   }

   /**
    * Current position of the servo, must be calling {@link #updateCurPos() updateCurPos()} periodically
    *
    * @return Servo Position [mm]
    */
   public double getPosition(){
      return curPos;
   }
```

```
    /**
     * Checks if the servo is at its target position, must be calling {@link #updateCurPos()
updateCurPos()} periodically
     * @return true when servo is at its target
     */
    public boolean isFinished(){
        return curPos == setPos;
    }

}
```

Revision Table

Revision Date	Revision #	Description
6/23/2021	1.0	First revision created.
6/25/2021	1.1	Added sample code.