

Potentiometers - Measuring joint angle or linear motion

Potentiometers are a common analog sensor used to measure absolute angular rotation or linear motion (string pots) of a mechanism. A potentiometer is a three terminal device that uses a moving contact to form a variable resistor divider. When the outer contacts are connected to 5V and ground and the variable contact is connected to an analog input, the analog input will see an analog voltage that varies as the potentiometer is turned.

Potentiometer Taper

The taper of a potentiometer describes the relationship between the position and the resistance. The two common tapers are linear and logarithmic. A linear taper potentiometer will vary the resistance proportionally to the rotation of the shaft; For example, the shaft will measure 50% of the resistive value at the midpoint of the rotation. A logarithmic taper potentiometer will vary the resistance logarithmically with the rotation of the shaft. Logarithmic potentiometers are commonly used in audio controls due to human perception of audio volume also being logarithmic.

Most or all FRC uses for potentiometers should use linear potentiometers so that angle can be deduced directly from the voltage.

Using Potentiometers with WPILib

Potentiometers can either be read with the `AnalogInput` class (then work with the voltage or perform the scaling in your code) or the `AnalogPotentiometer` class which implements the `Potentiometer` interface. The `AnalogPotentiometer` class will read the sensor ratiometrically (compensate for the Analog supply voltage) and will scale and offset the voltage to return meaningful units.

Constructing a Potentiometer

C++

```
Potentiometer *pot;  
pot = new AnalogPotentiometer(0, 360, 30);  
AnalogInput *ai = new AnalogInput(1);  
pot = new AnalogPotentiometer(ai, 360, 30);
```

Java

```
Potentiometer pot;
```

Potentiometers - Measuring joint angle or linear motion

```
pot = new AnalogPotentiometer(0, 360, 30);  
AnalogInput ai = new AnalogInput(1);  
pot = new AnalogPotentiometer(ai, 360, 30);
```

The Potentiometer constructor takes 3 parameters: a channel number for the analog input, a scale factor to multiply the 0-1 ratiometric value by to return useful units, and an offset to add after the scaling. Generally, the most useful scale factor will be the angular or linear full scale of the potentiometer. For example, let's say you have an ideal single turn linear potentiometer attached to a robot arm. This pot will turn 360 degrees over the 0V-5V range so using that for the scale factor will result in the output being in degrees. In order to prevent the potentiometer from breaking due to minor shifting in alignment, it may be installed with the "zero-point" of the arm a little ways into the potentiometers range, the example above represents the potentiometer being installed with an initial value of 30 degrees, which is negated using the offset so that the output is 0 at the "zero-point" of the mechanism.

You can also pass an existing AnalogInput to the constructor in place of the channel if you wish to share the input with other code.

The Calculations

The potentiometer output is calculated using the following formula: $(\text{Analog Input Voltage} / \text{Analog Supply Voltage}) * \text{FullScale} + \text{Offset}$. As you can see the result of the first part of the calculation is a unitless quantity in the range 0 to 1. This means the units of the output are the same as the units of the scale factor. The offset is added to the scaled quantity so it should have the same units as the scale factor.

Reading the output

C++

```
Potentiometer *pot = new AnalogPotentiometer(0, 360, 30);  
double degrees = pot->Get();
```

Java

```
Potentiometer pot = new AnalogPotentiometer(0, 360, 30);  
double degrees = pot.get()
```

To read the potentiometer output, call the Get() method.