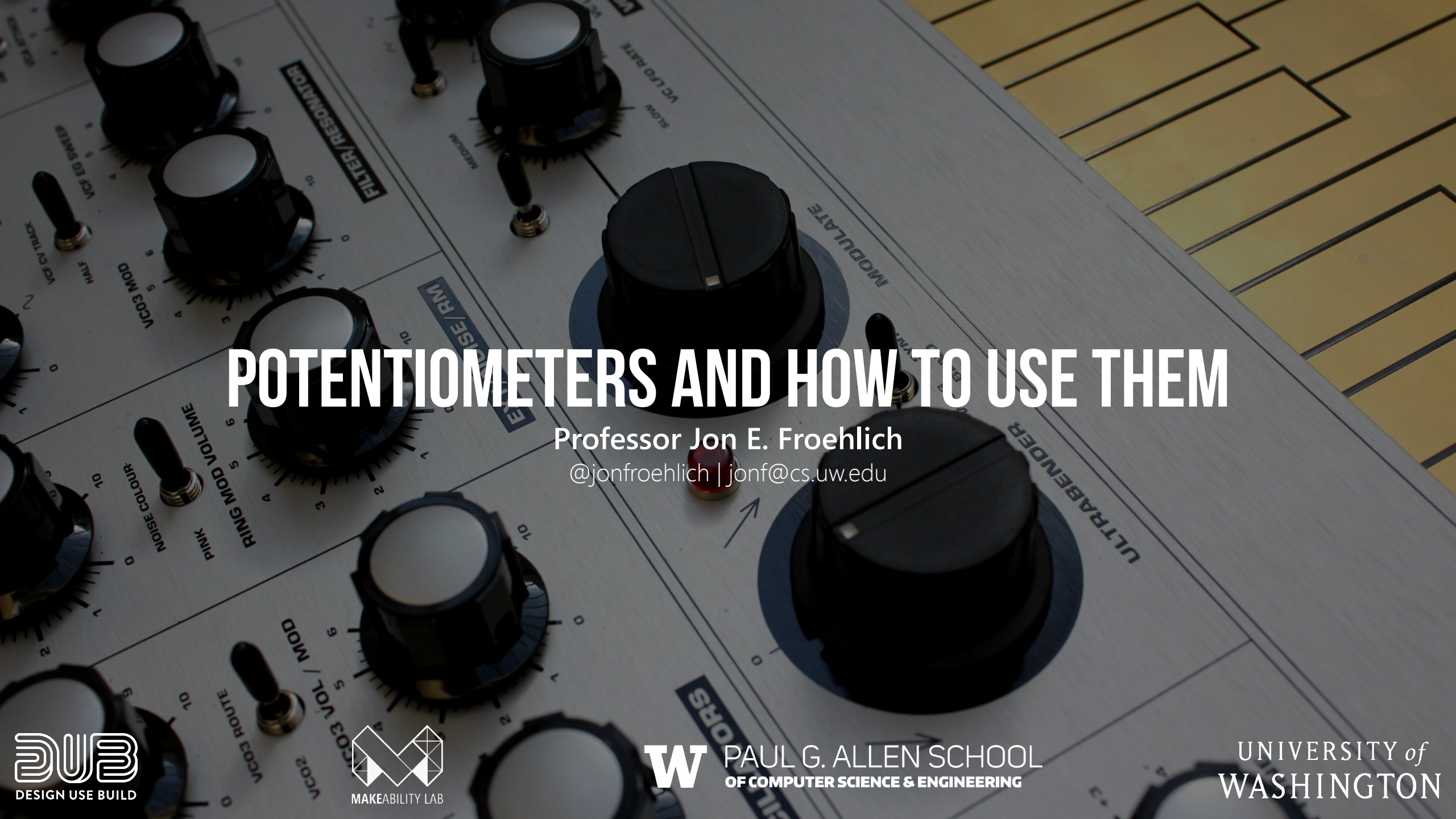


POTENTIOMETERS AND HOW TO USE THEM

Professor Jon E. Froehlich
@jonfroehlich | jonf@cs.uw.edu

DESIGN USE BUILD

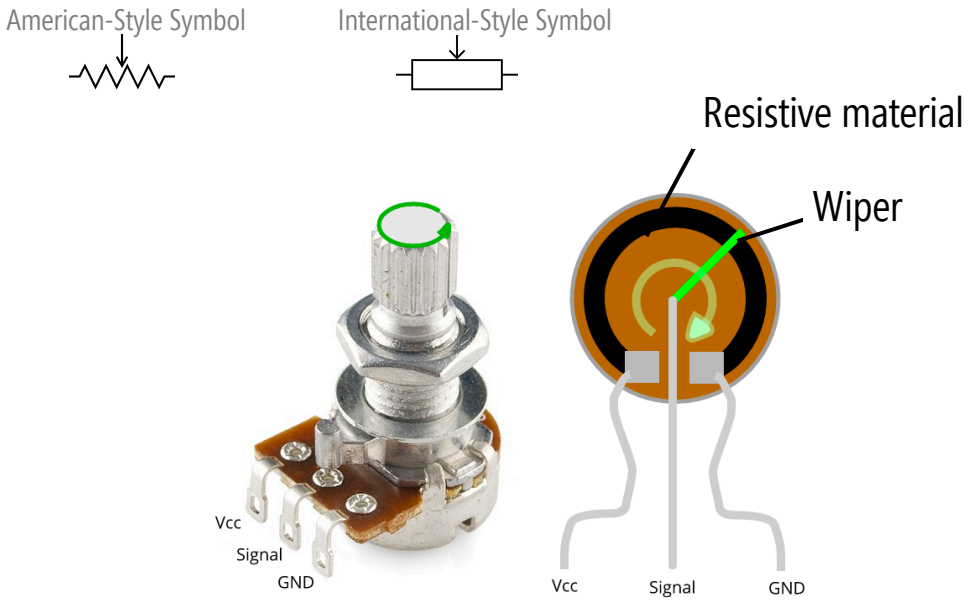
MAKEABILITY LAB

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY of WASHINGTON

VARIABLE RESISTORS

WHAT IS A POTENTIOMETER?

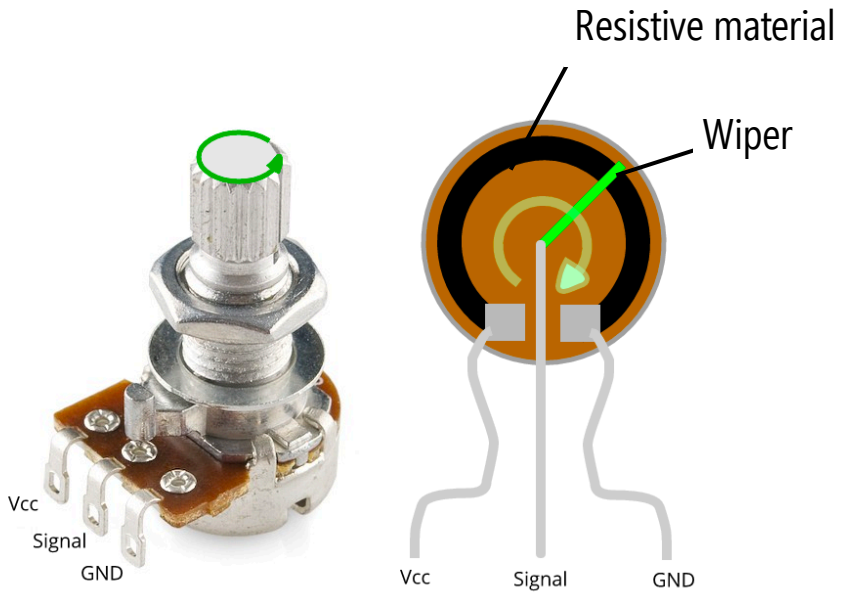


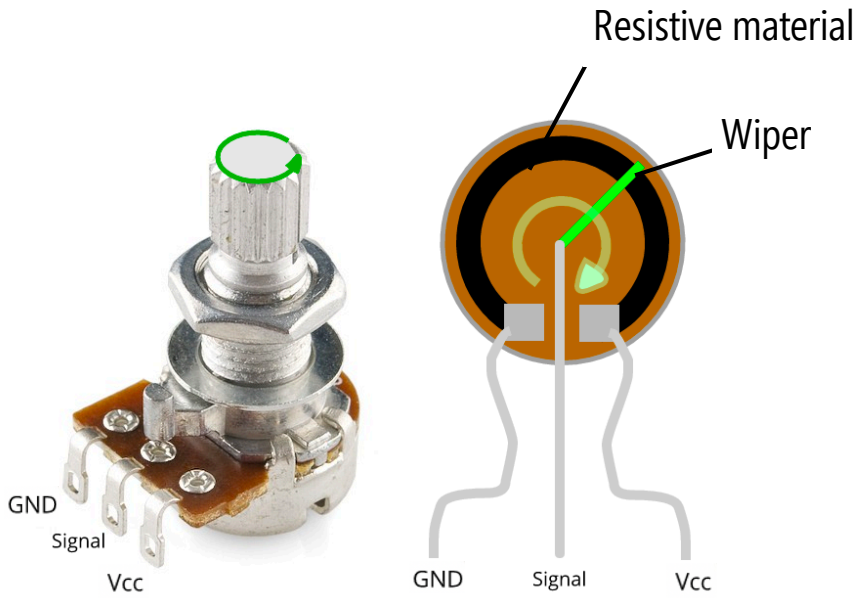
A three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divider

If only two terminals are used, the potentiometer acts as a **rheostat** or a two-terminal variable resistor

Source: https://en.wikipedia.org/wiki/Potentiometer

POTENTIOMETERS CAN BE HOOKED UP IN EITHER ORIENTATION

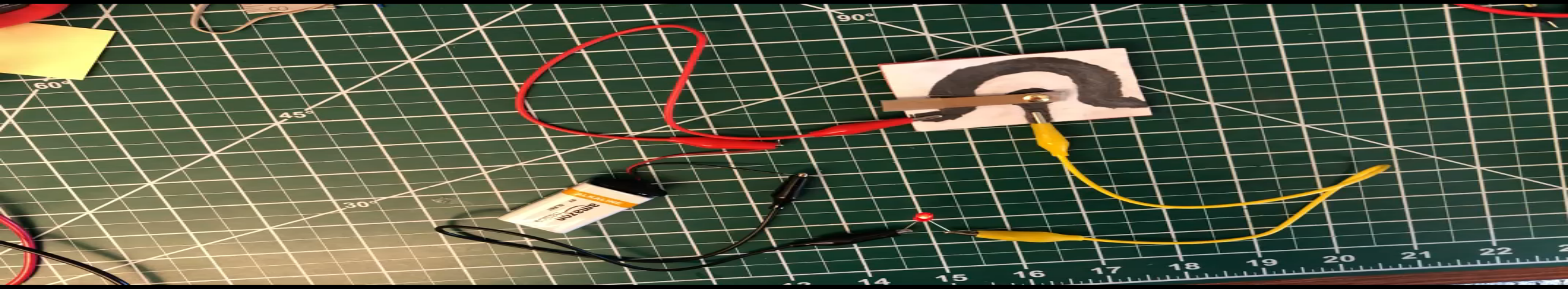




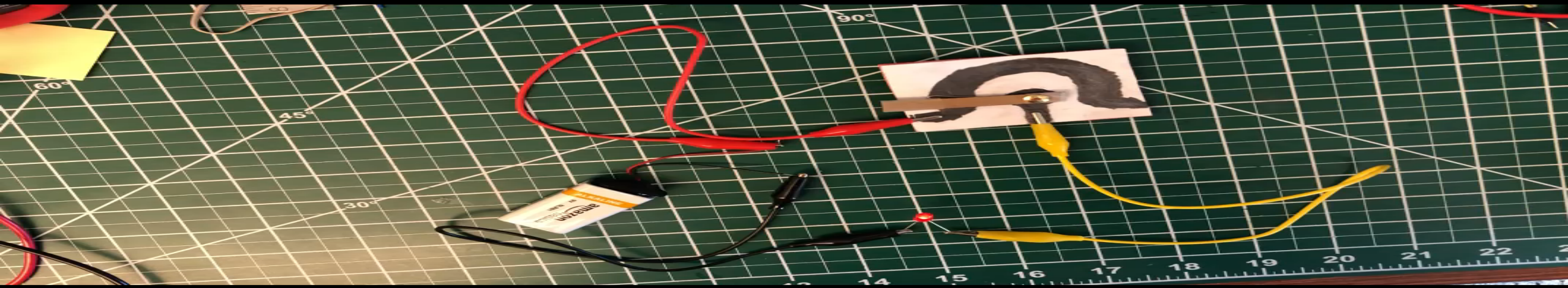
Source: https://en.wikipedia.org/wiki/Potentiometer

POTENTIOMETERS

DEMO OF LO-FI POTENTIOMETER

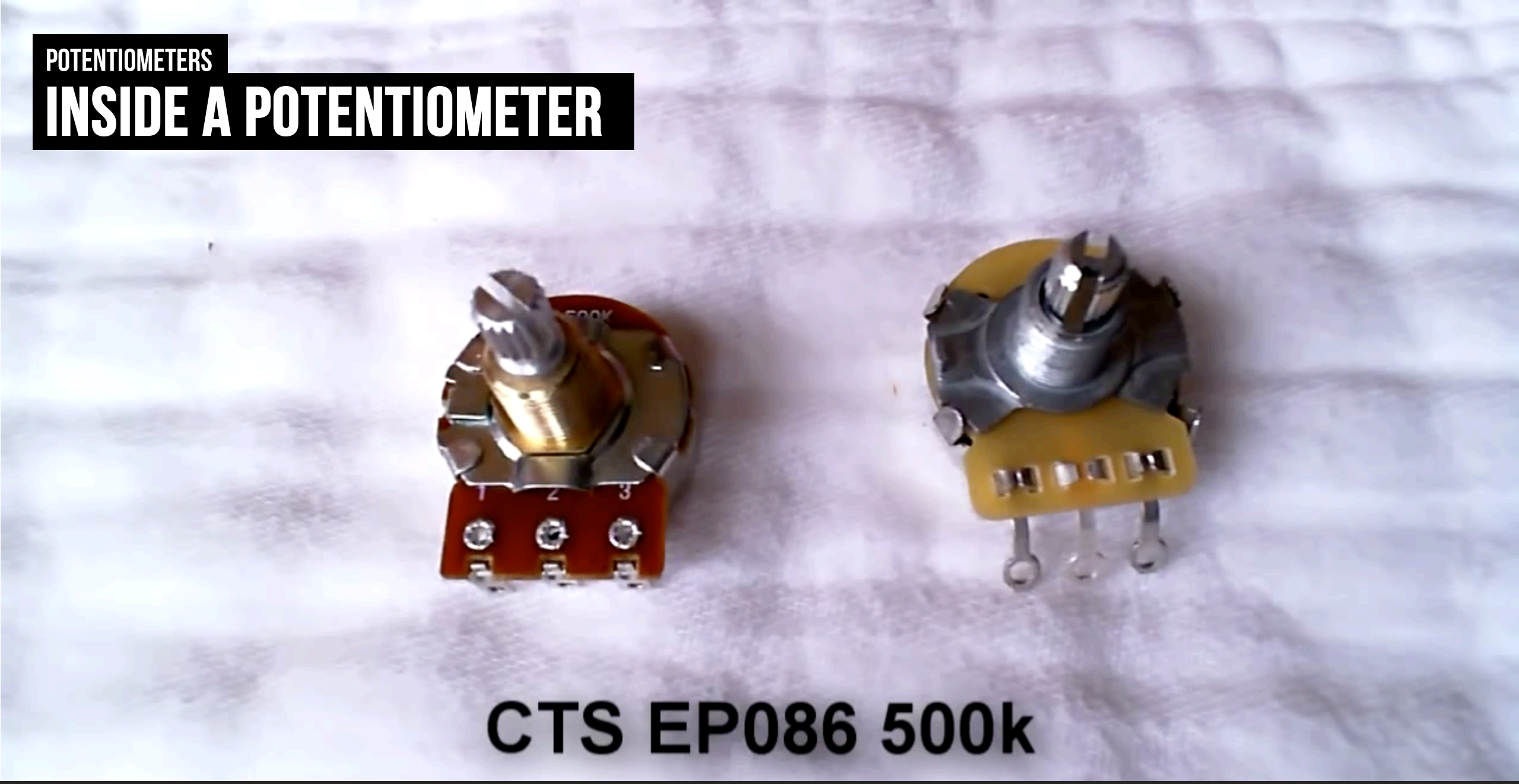


Video by Jon Froehlich

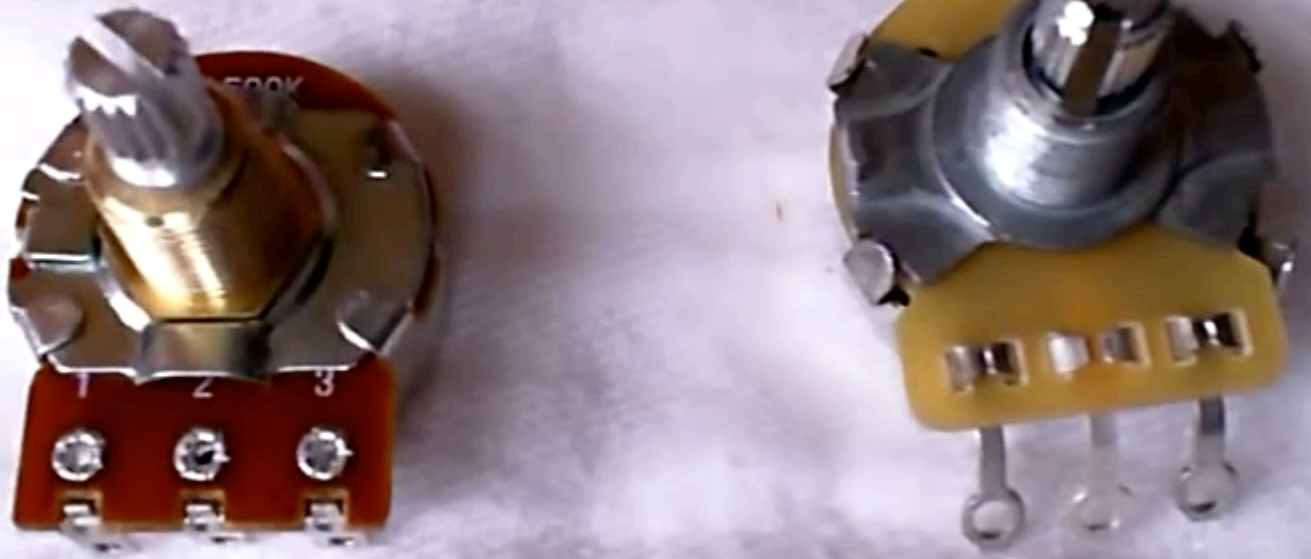

90°
60°
45°
30°
14
15
16
17
18
19
20
21
22

POTENTIOMETERS

INSIDE A POTENTIOMETER

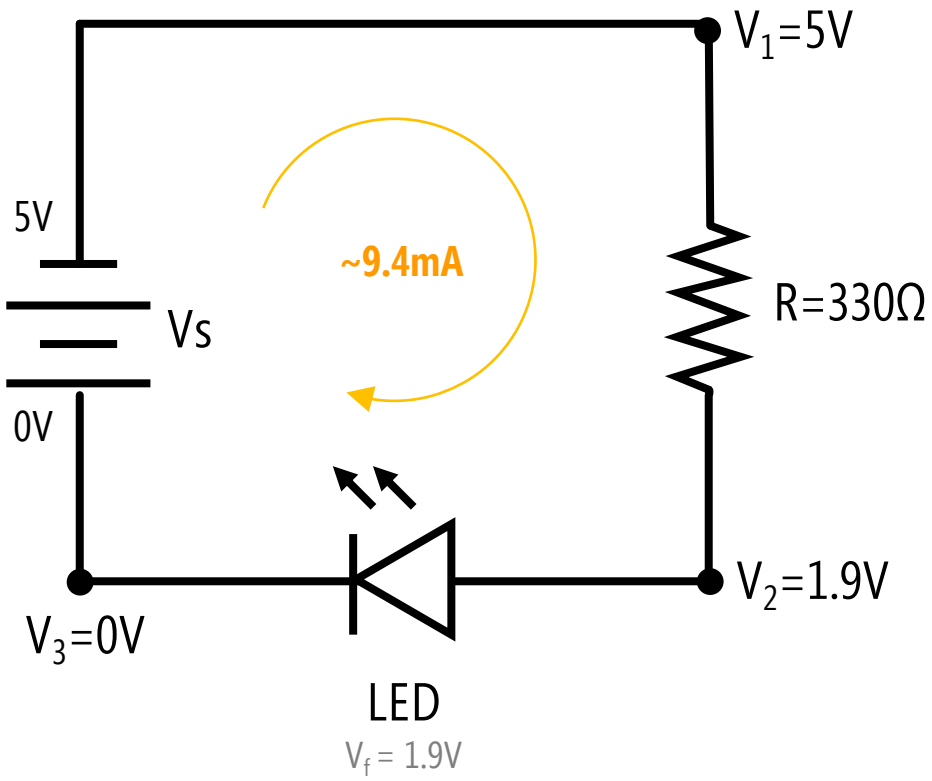


Source: https://youtu.be/rUkrpqEmXb8

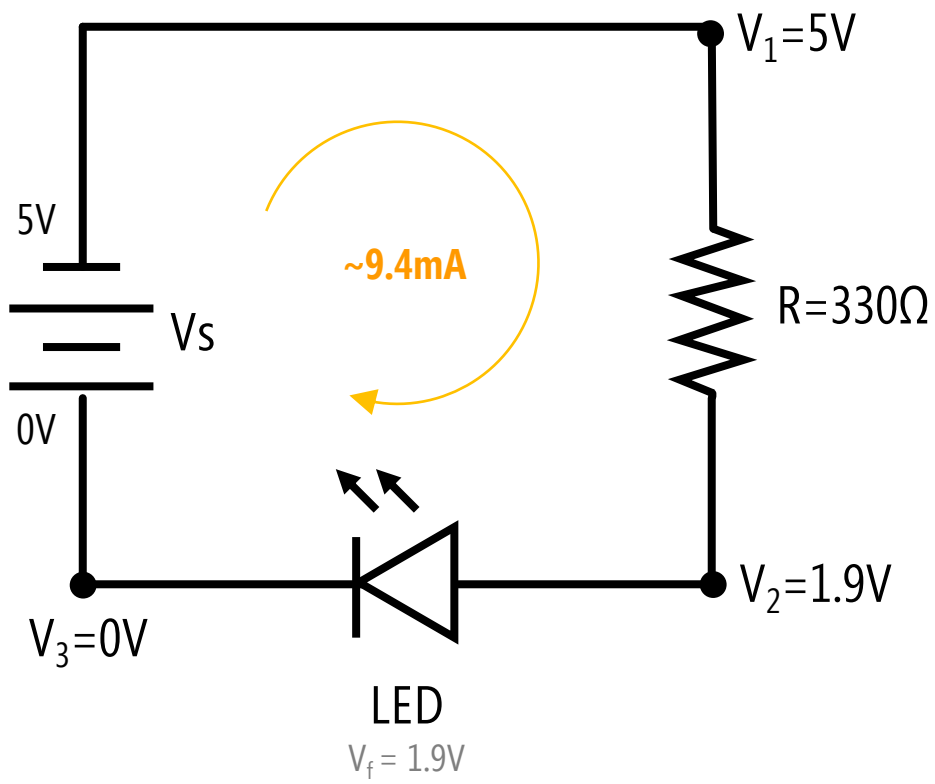


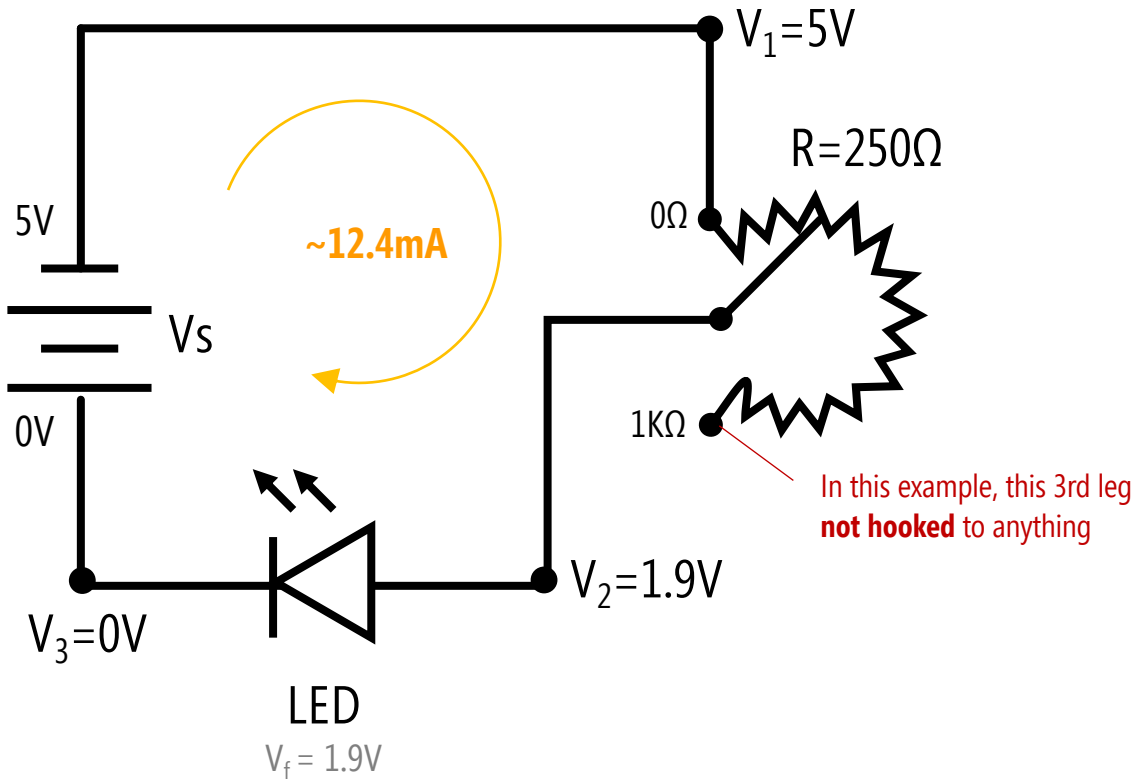
CTS EP086 500k

RESISTOR VS. POTENTIOMETER

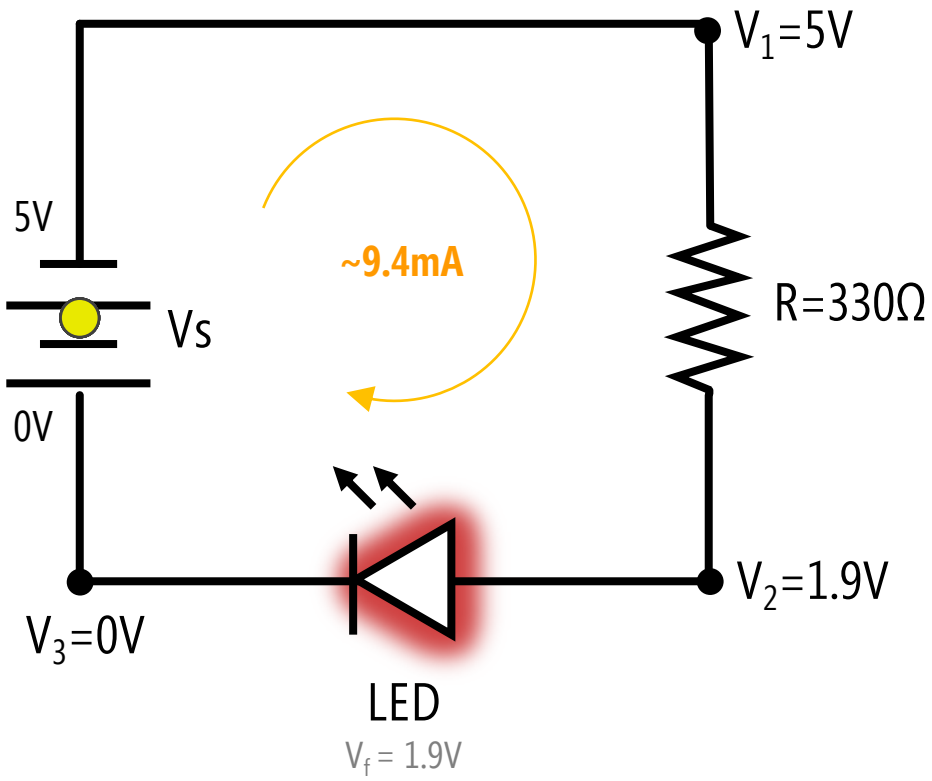


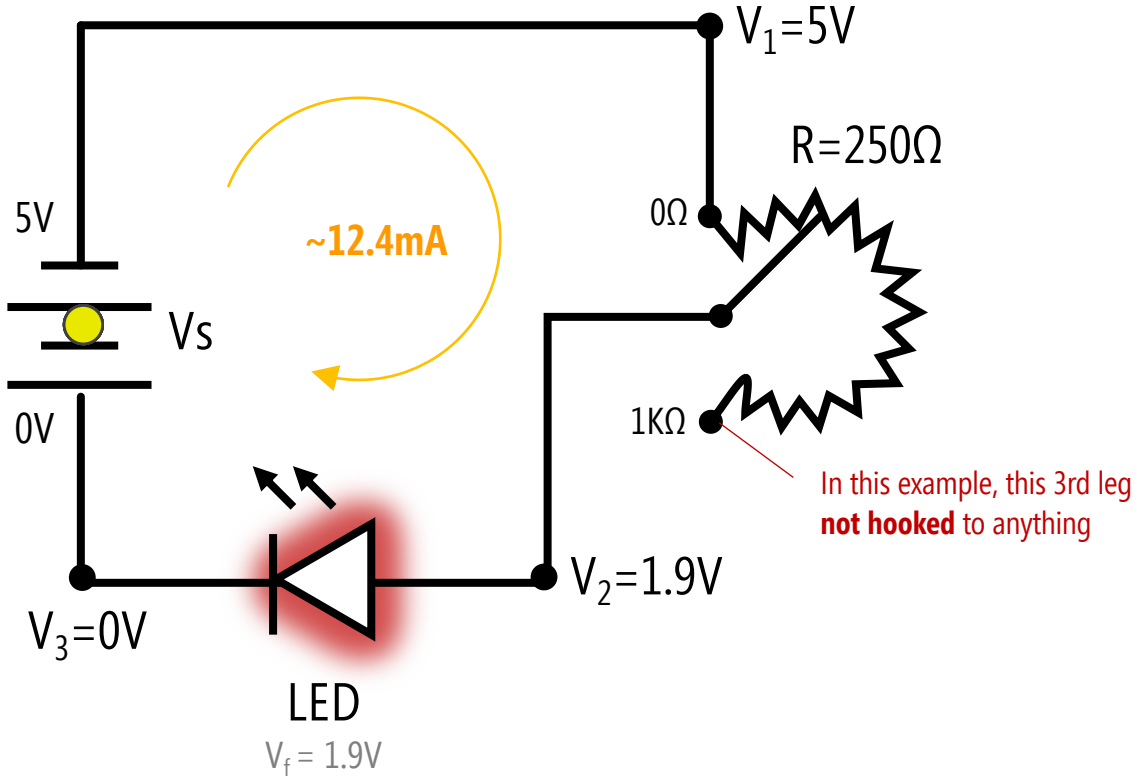
RESISTOR VS. POTENTIOMETER



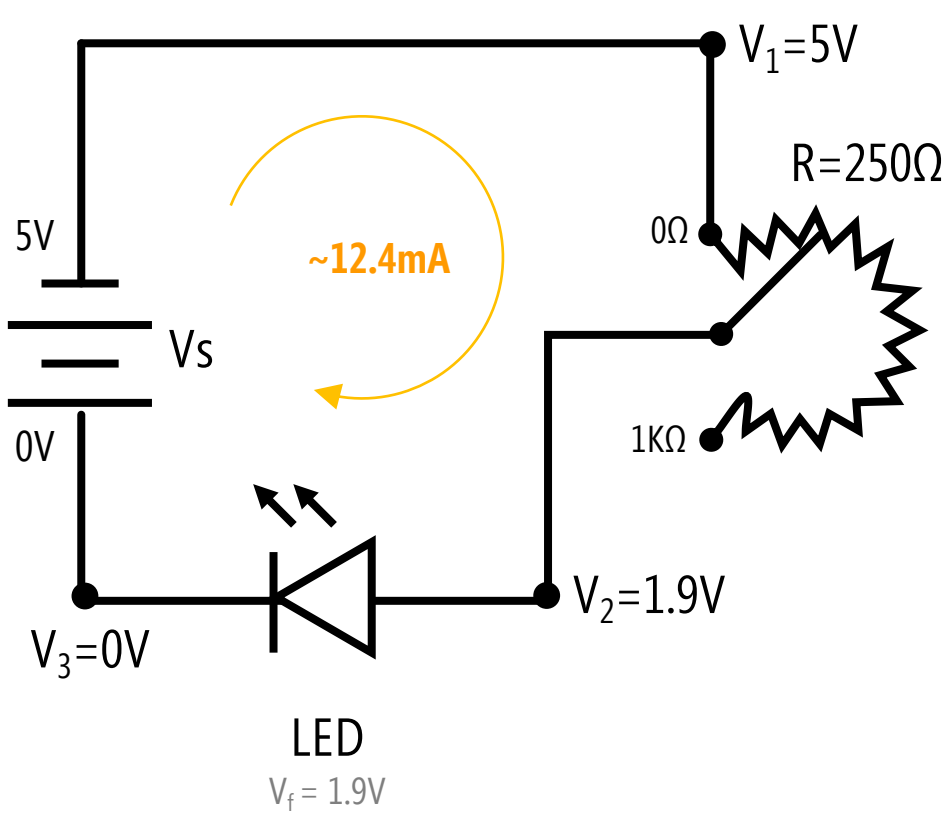


RESISTOR VS. POTENTIOMETER



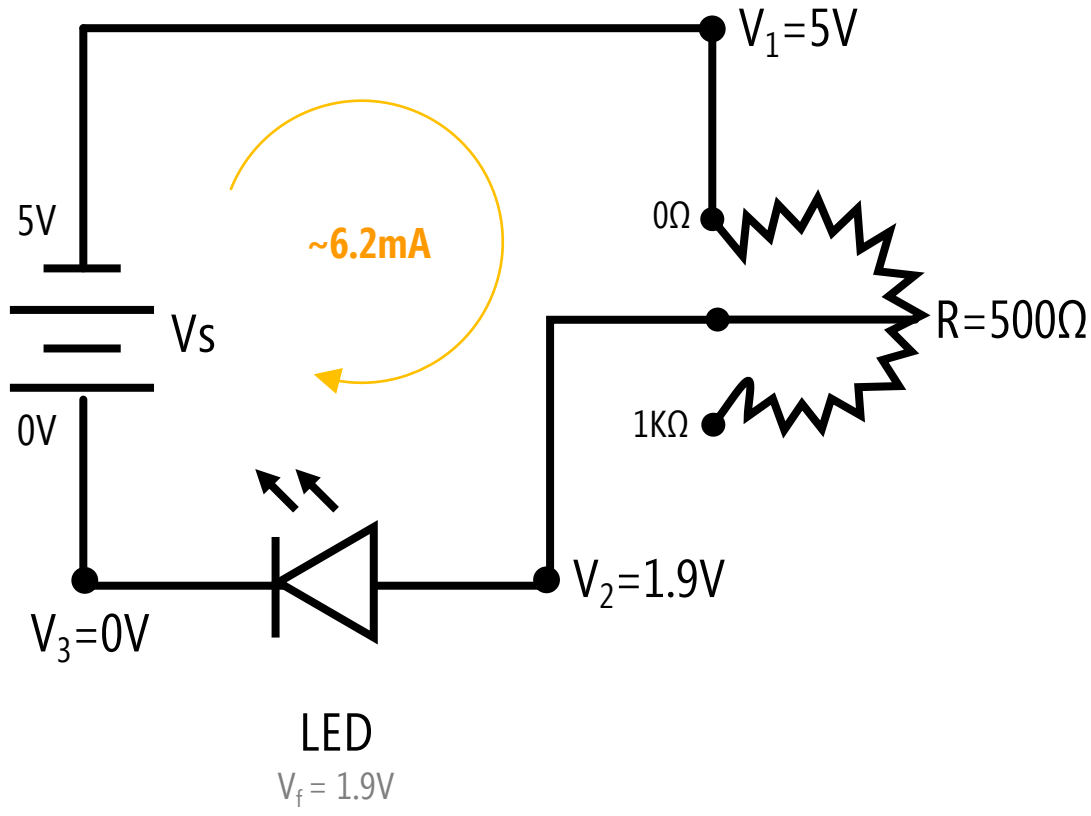


POTENTIOMETER EXAMPLES



LED

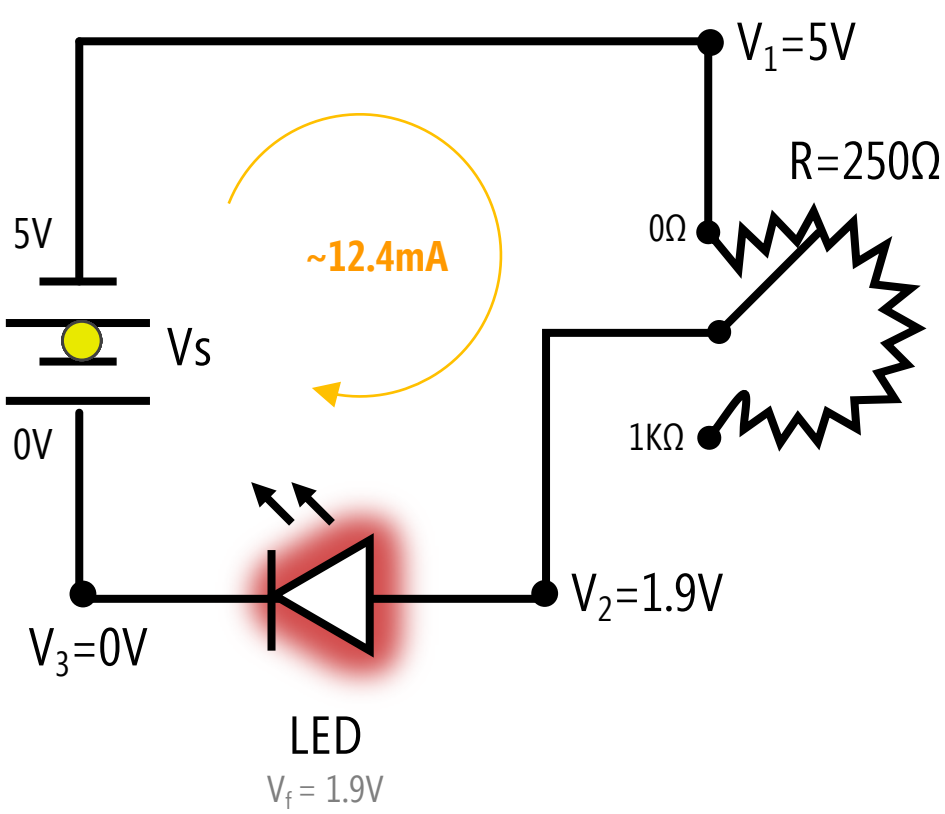
V_f = 1.9V

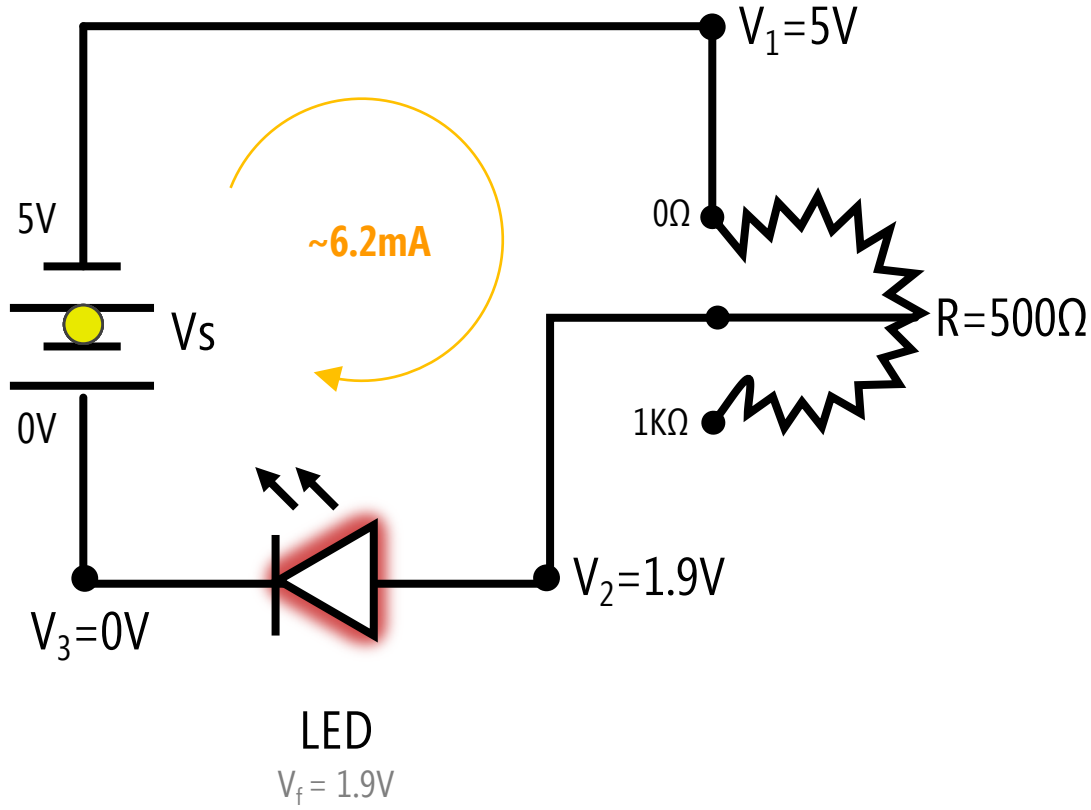


LED

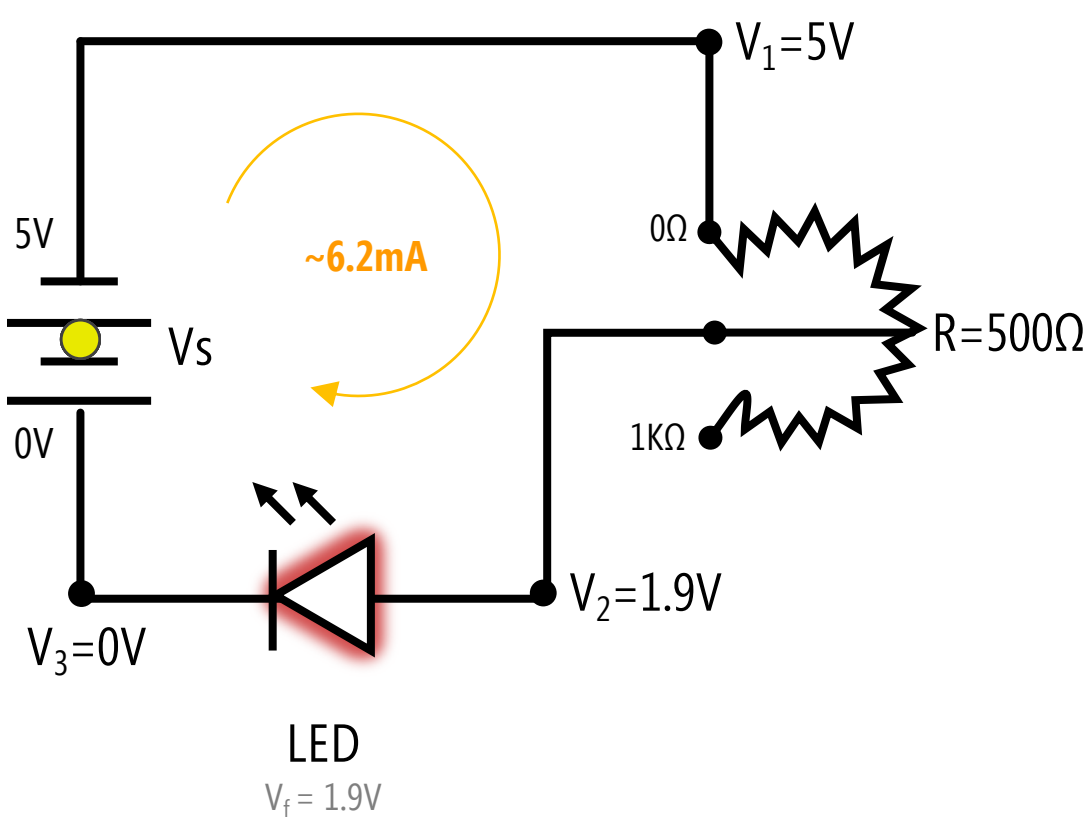
V_f = 1.9V

POTENTIOMETER EXAMPLES



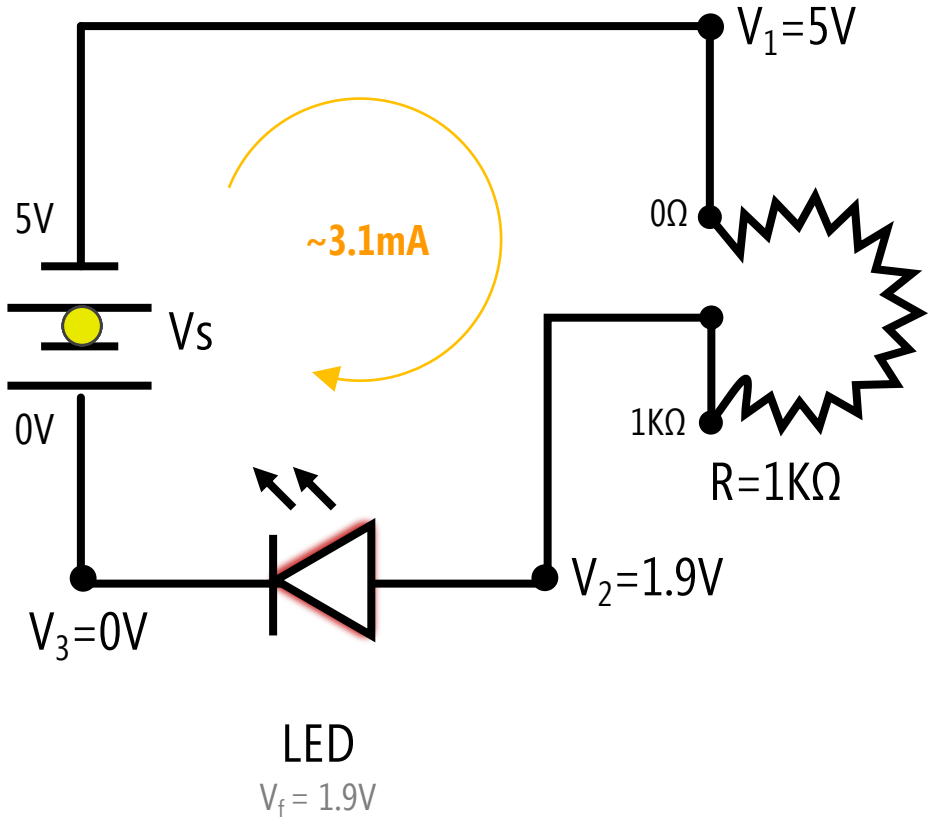


POTENTIOMETER EXAMPLES



LED

V_f = 1.9V



LED

V_f = 1.9V

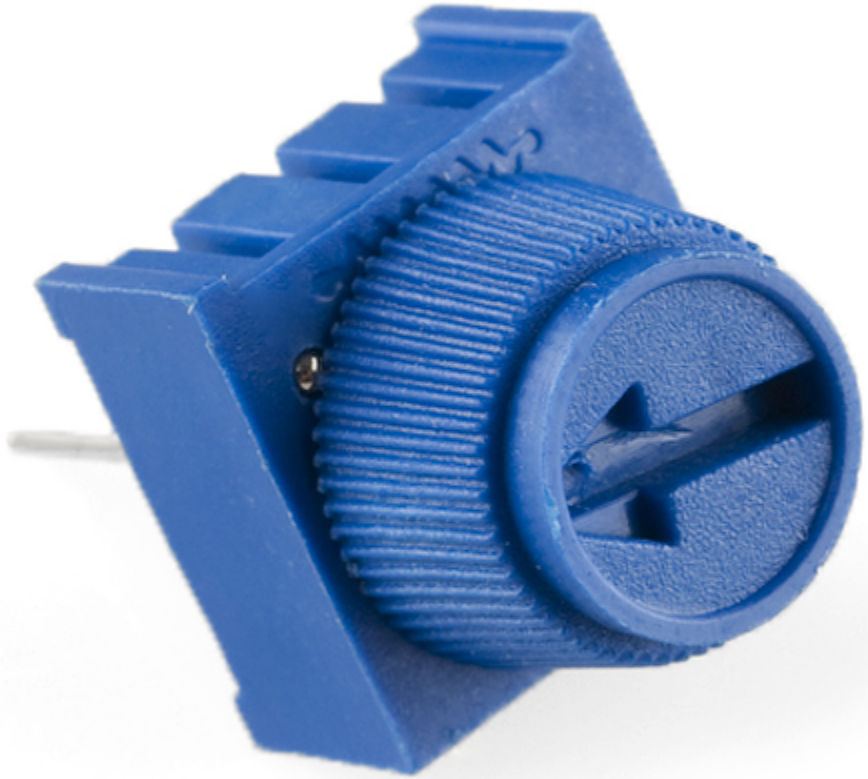
POTENTIOMETERS

LINEAR VS. EXPONENTIAL RESISTANCE

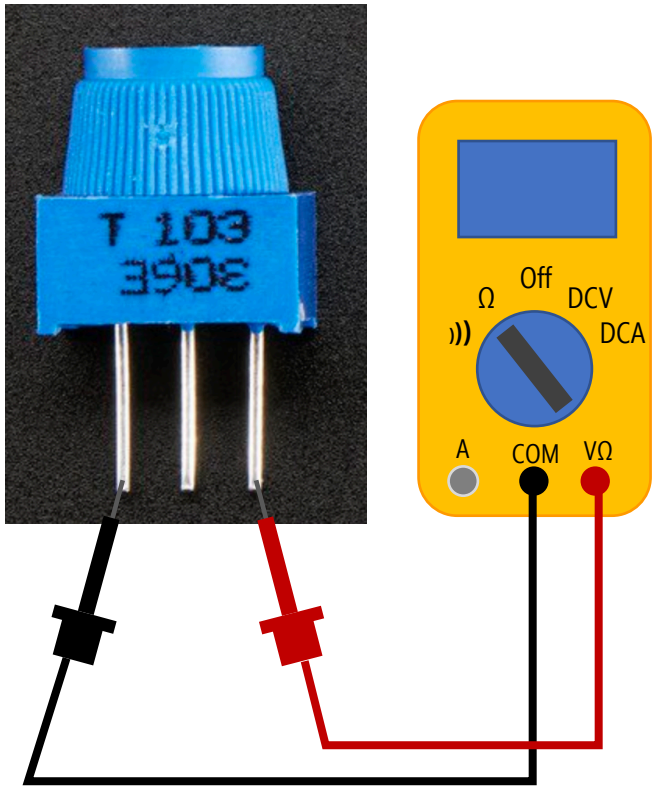
Source: https://youtu.be/rUkrpqEmXb8

VARIABLE RESISTORS

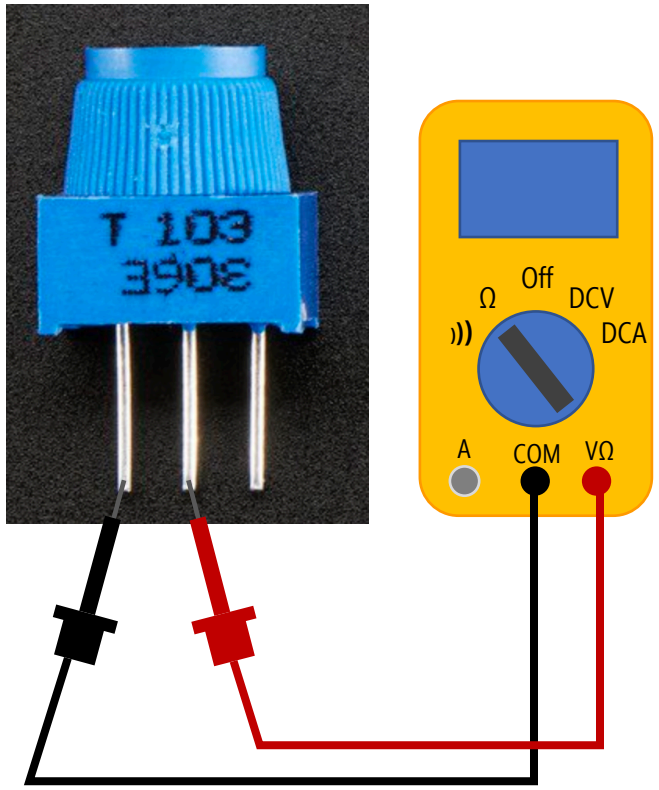
BREADBOARD-FRIENDLY TRIMPOT



ACTIVITY: EXPLORE THE POT WITH A MULTIMETER

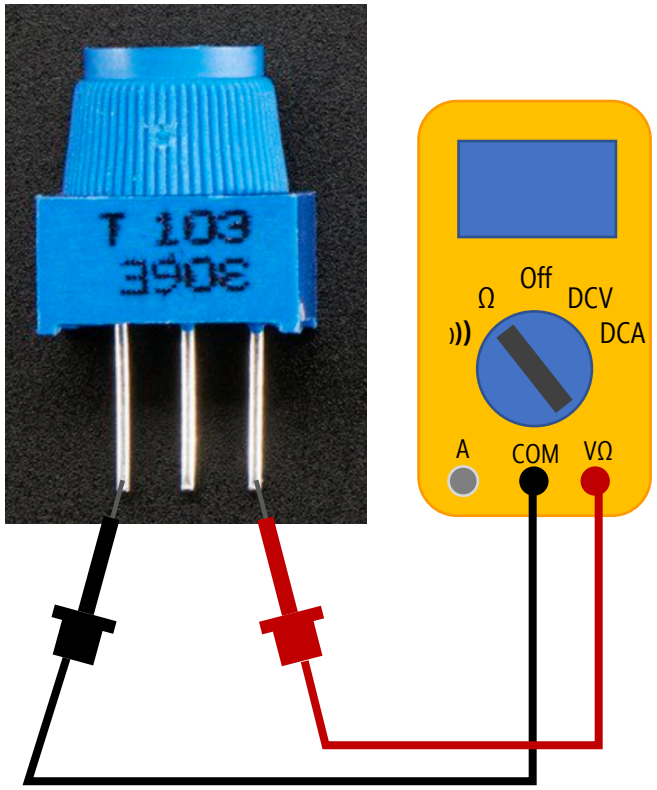


Hook up the **two outer legs** to the multimeter. What happens when you turn the knob?

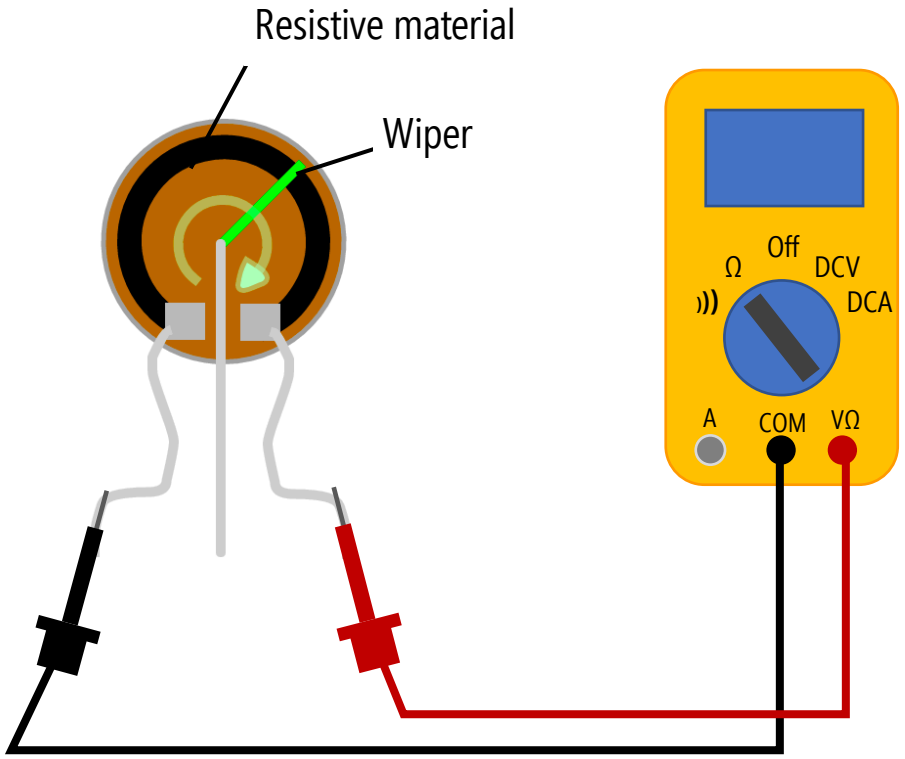


Hook up the **middle** and an **outer leg** to the multimeter. What happens when you turn the knob?

ACTIVITY: EXPLORE THE POT WITH A MULTIMETER

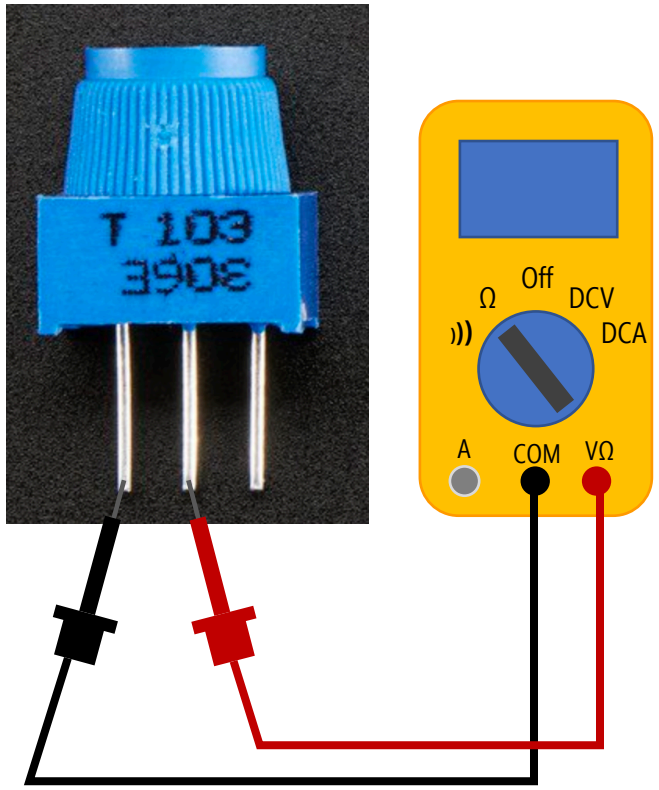


Hook up the **two outer legs** to the multimeter. What happens when you turn the knob?



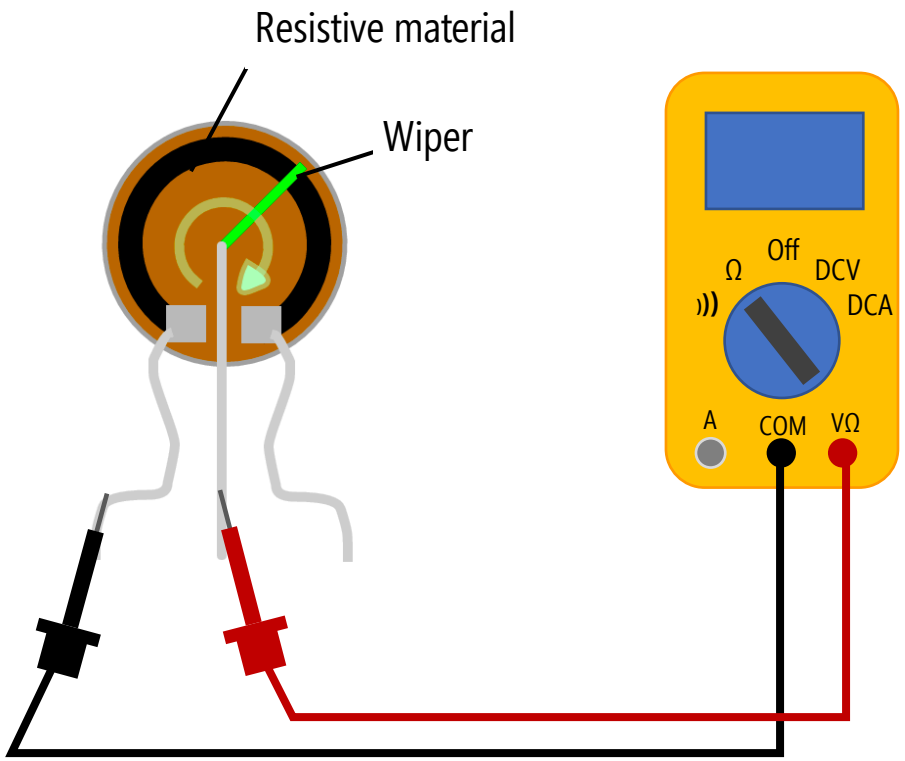
Answer: Nothing! The multimeter will always read the full max resistance of the potentiometer since you are measuring

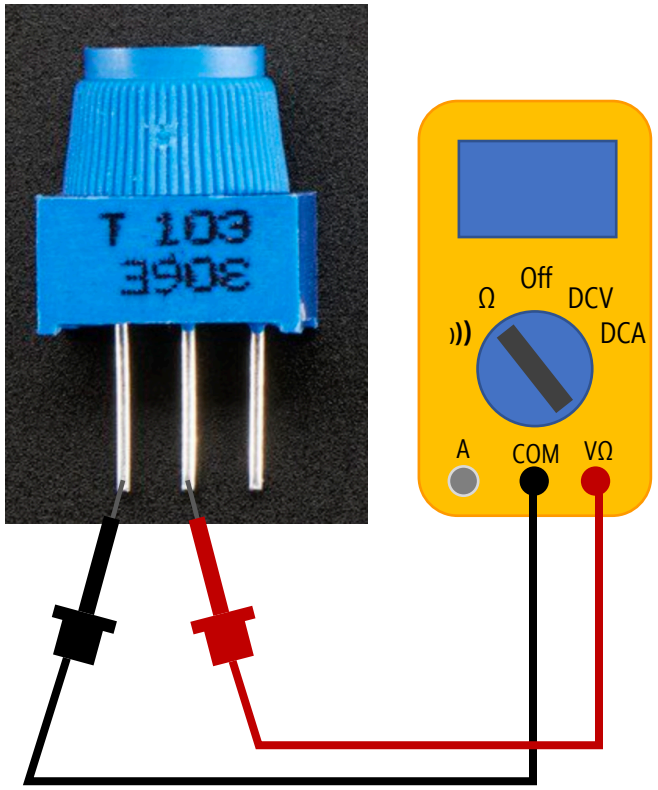
ACTIVITY: EXPLORE THE POT WITH A MULTIMETER



Hook up the **middle** and an **outer leg** to the multimeter. What happens when you turn the knob?

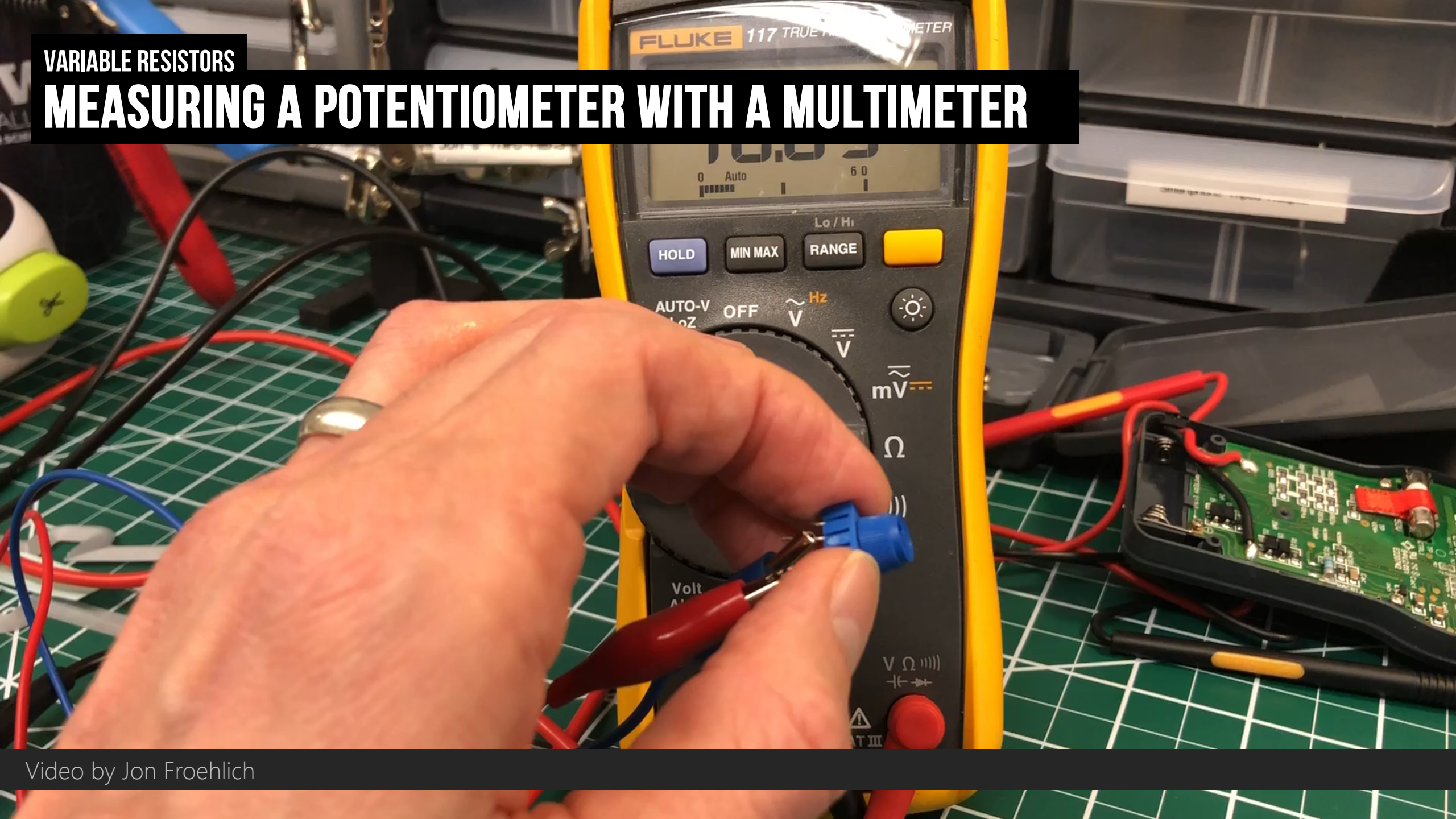
ACTIVITY: EXPLORE THE POT WITH A MULTIMETER

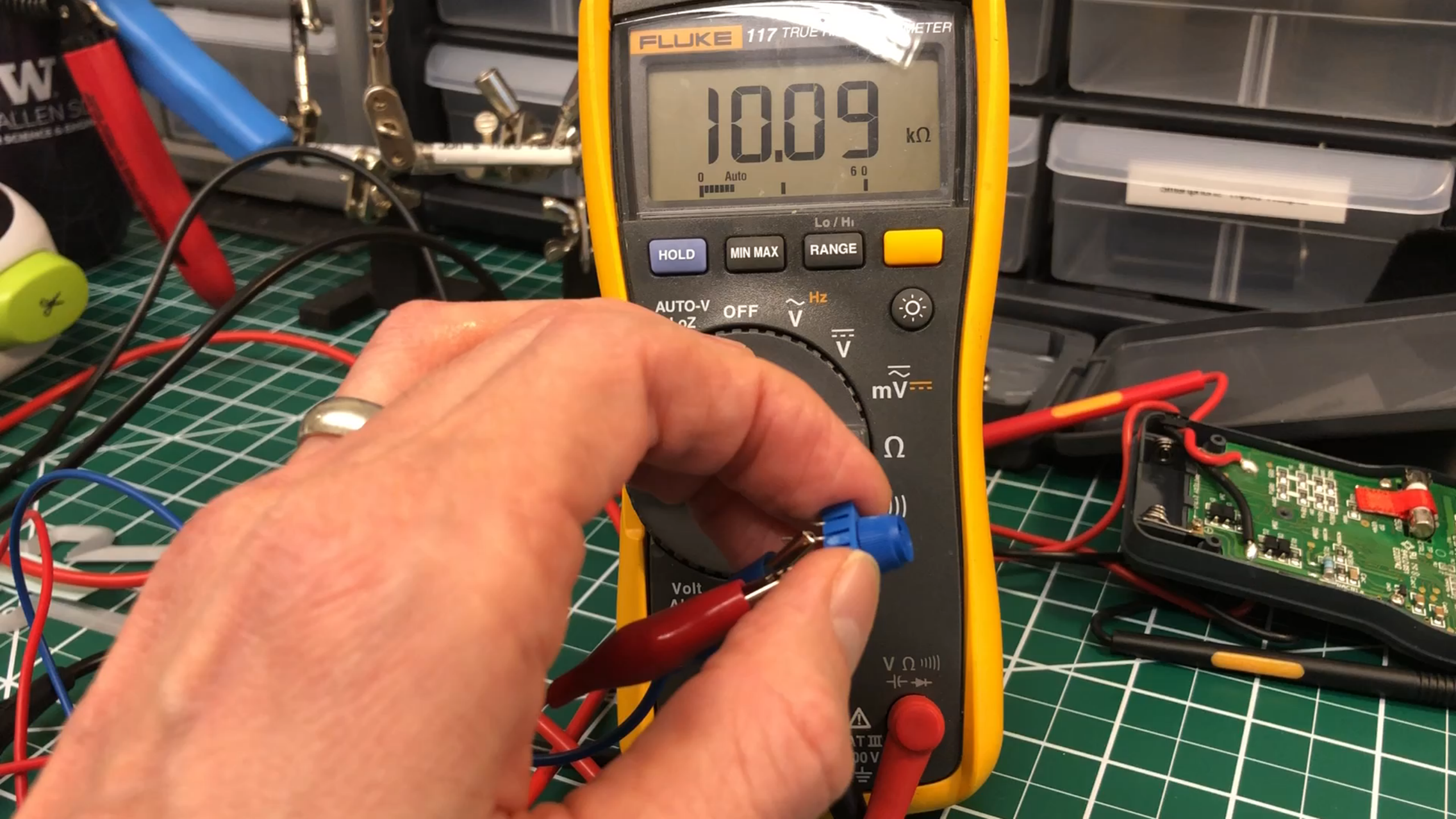




Answer: You'll see the resistance change based on the position of the knob. In this case, 0Ω with the knob all the way to the left, 5K Ω in middle, and 10KΩ with knob all the way to the right

Hook up the **middle** and an **outer leg** to the multimeter. What happens when you turn the knob?

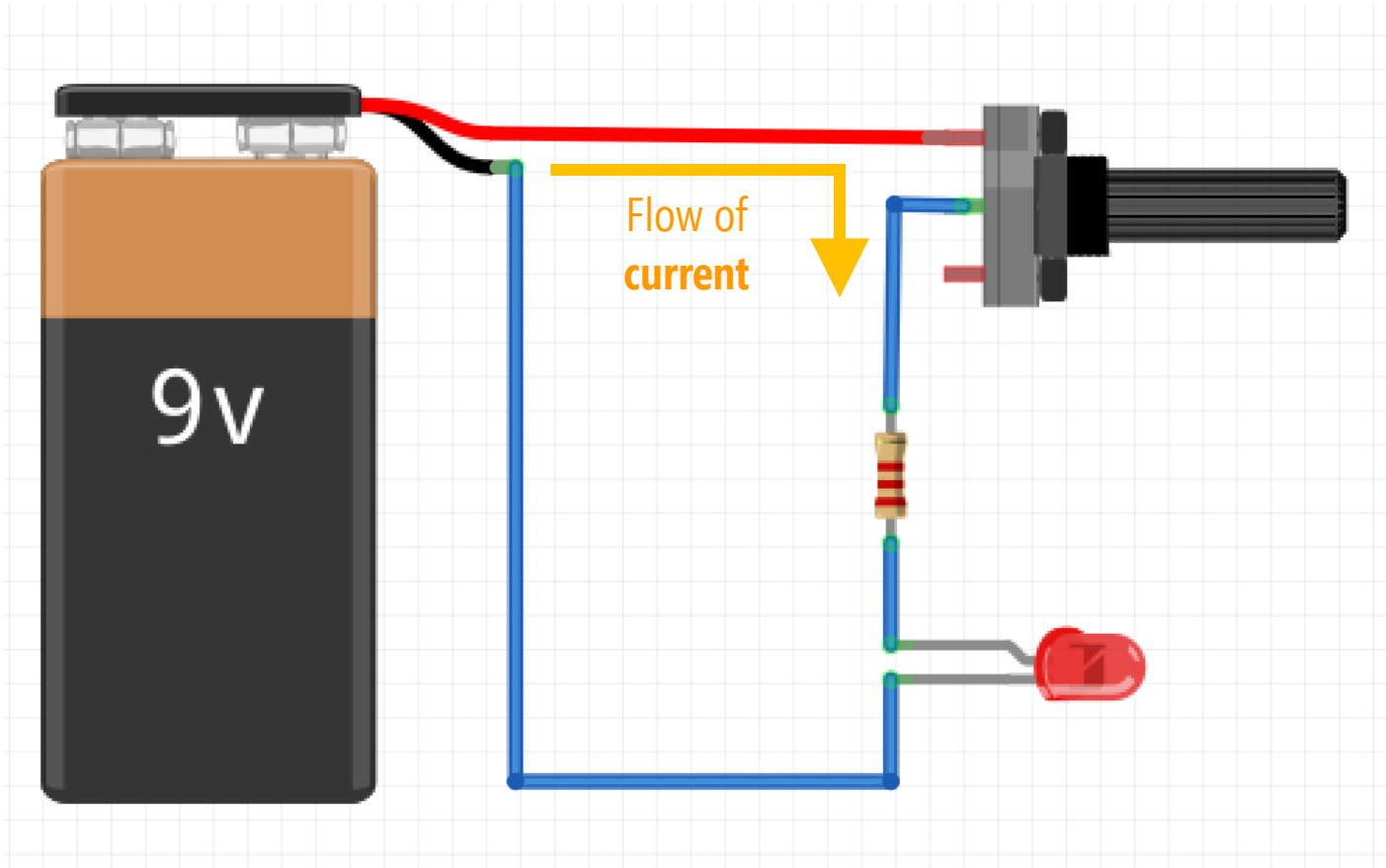

VARIABLE RESISTORS
MEASURING A POTENTIOMETER WITH A MULTIMETER
FLUKE
HOLD
MIN MAX
RANGE
OFF
Video by Jon Froehlich


FLUKE
10.09
kΩ
HOLD
MIN MAX
RANGE
Lo / Hi
AUTO-V
OFF
Hz
V
mV
Ω

POTENTIOMETERS

ACTIVITY HOOK UP AN LED WITH A POT & ALLIGATOR CLIPS

ACTIVITY HOOK UP AN LED WITH A POT & ALLIGATOR CLIPS



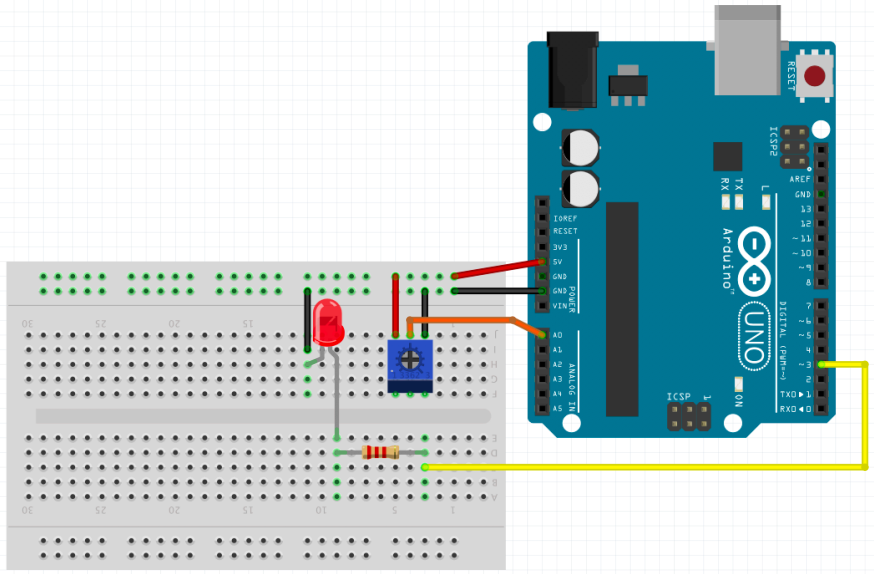
ANALOG INPUT

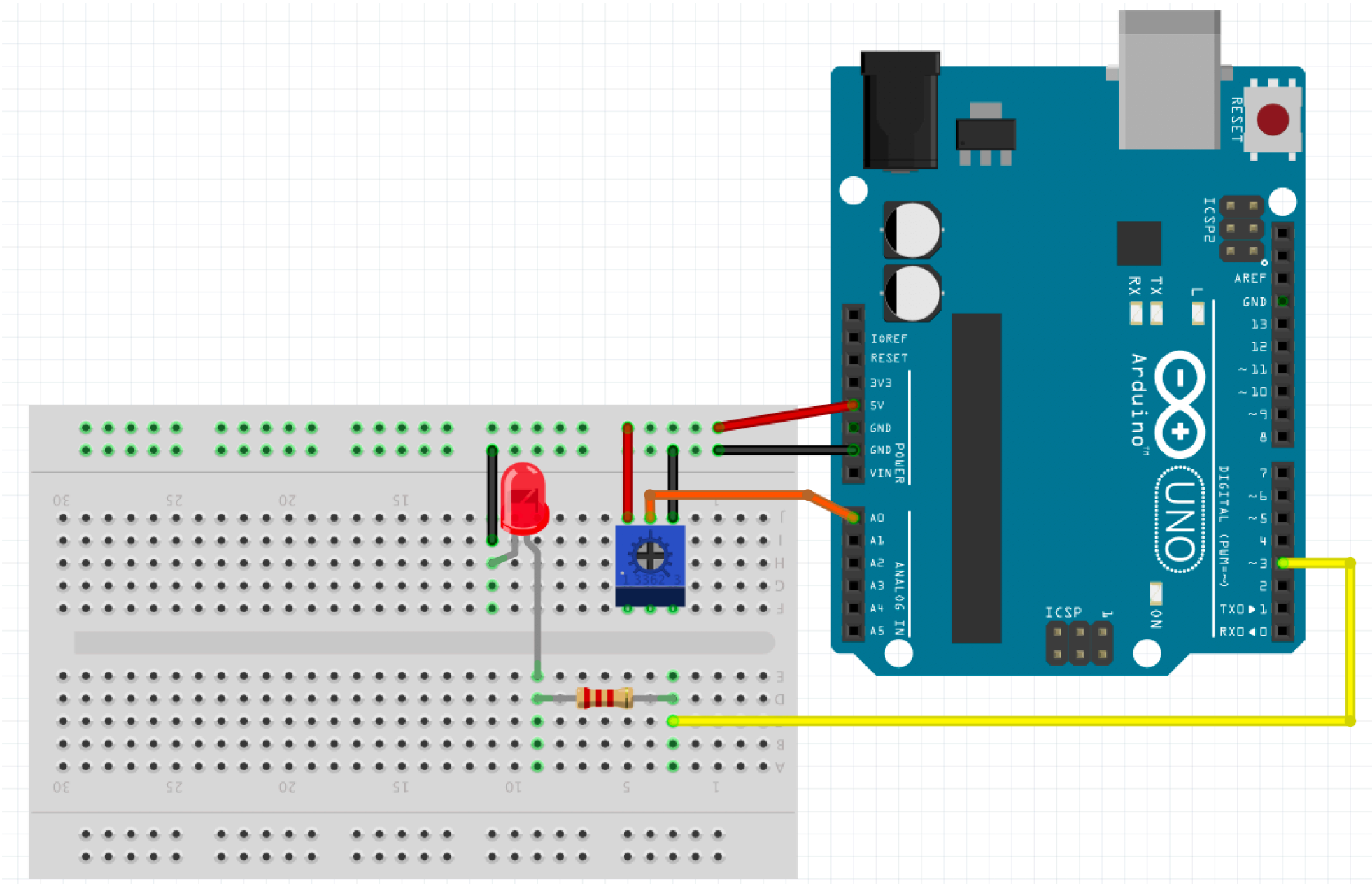
NOW LET'S DO THE SAME THING BUT WITH ARDUINO

Here, we will make **two separate circuits**.

One for input

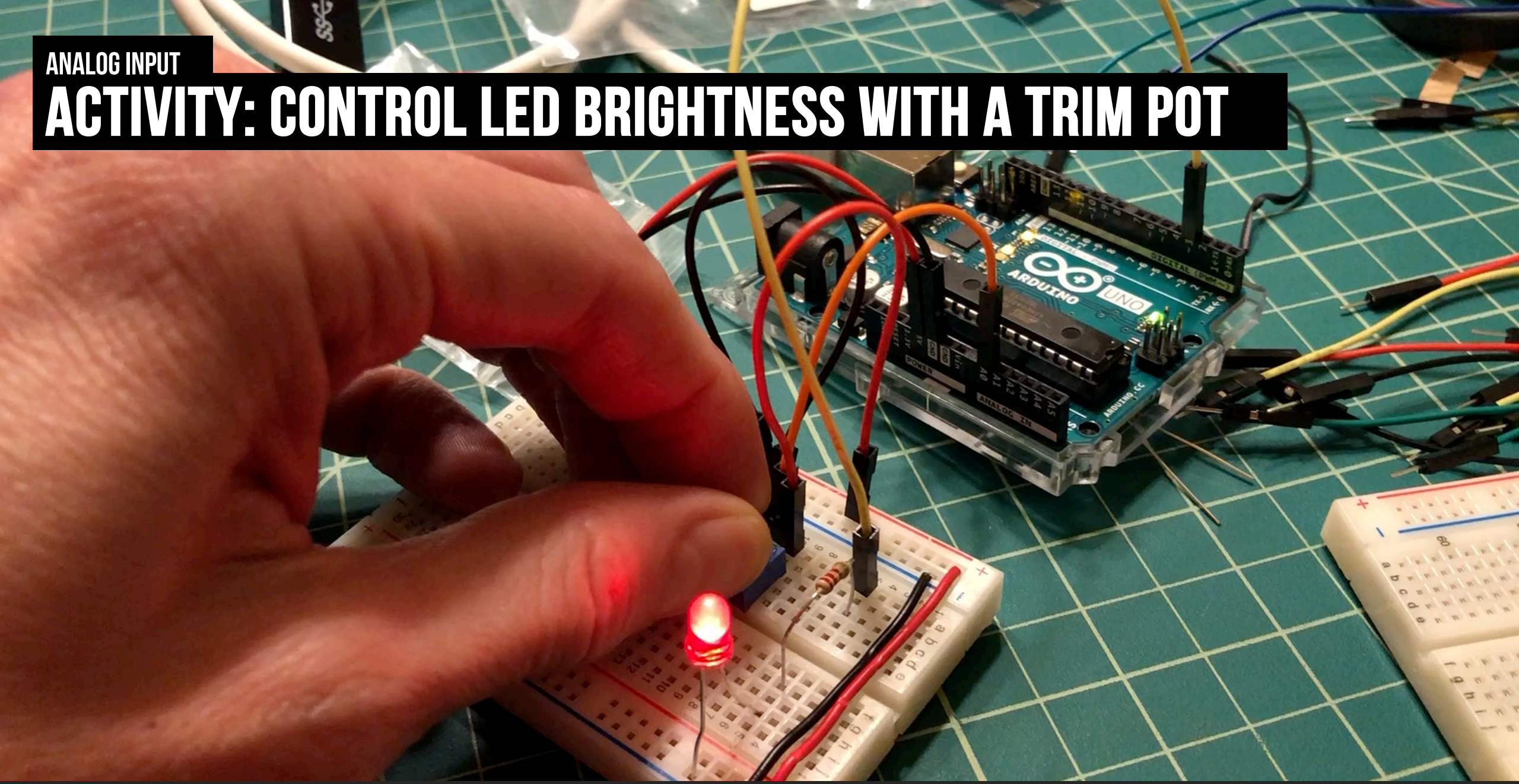
One for output




RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
TX
RX
L
ON
ICSP
1
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
UNO
Arduino™
30
25
20
15
10
5
1
J
I
H
G
F
E
D
C
B
A

ANALOG INPUT

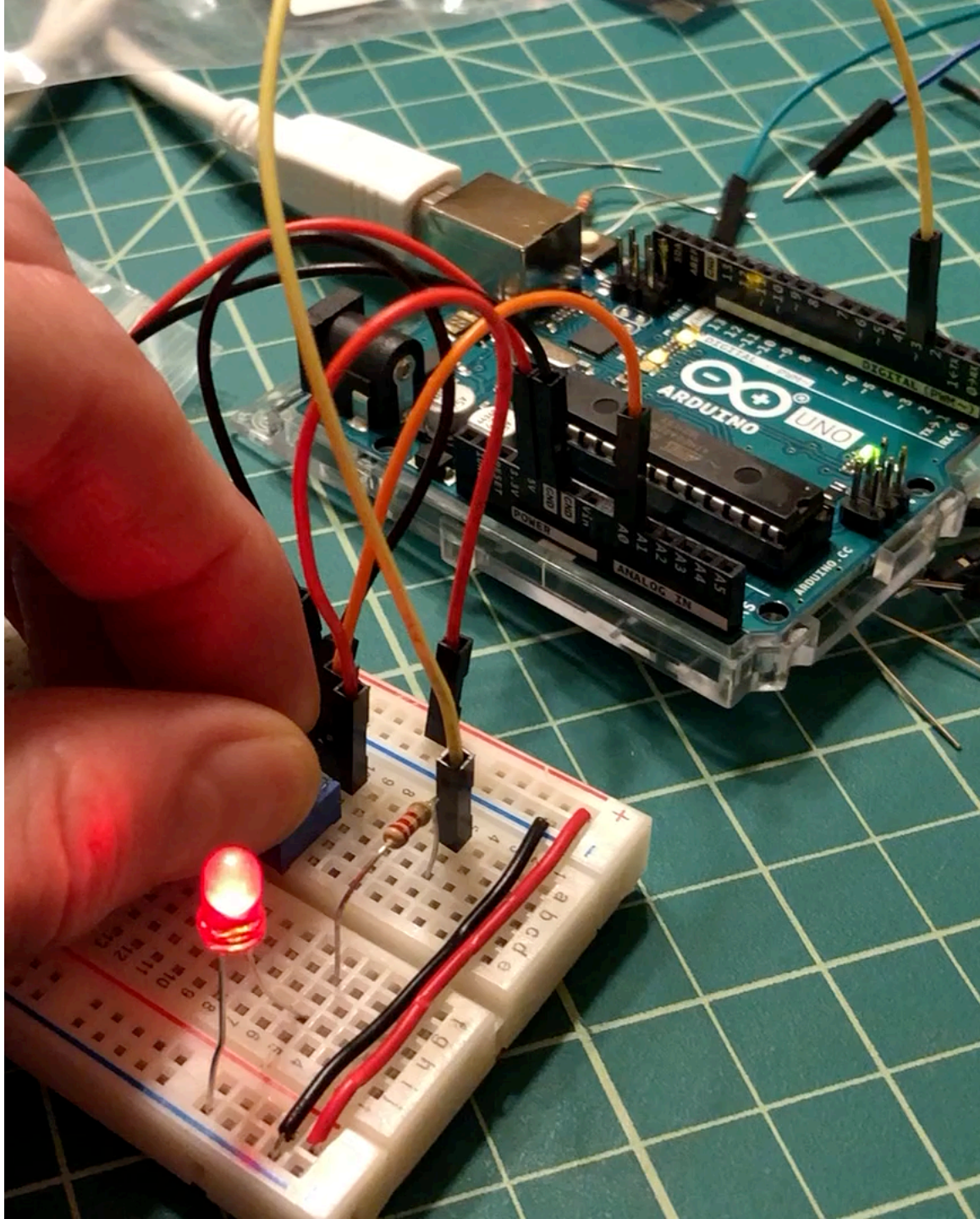
ACTIVITY: CONTROL LED BRIGHTNESS WITH A TRIM POT



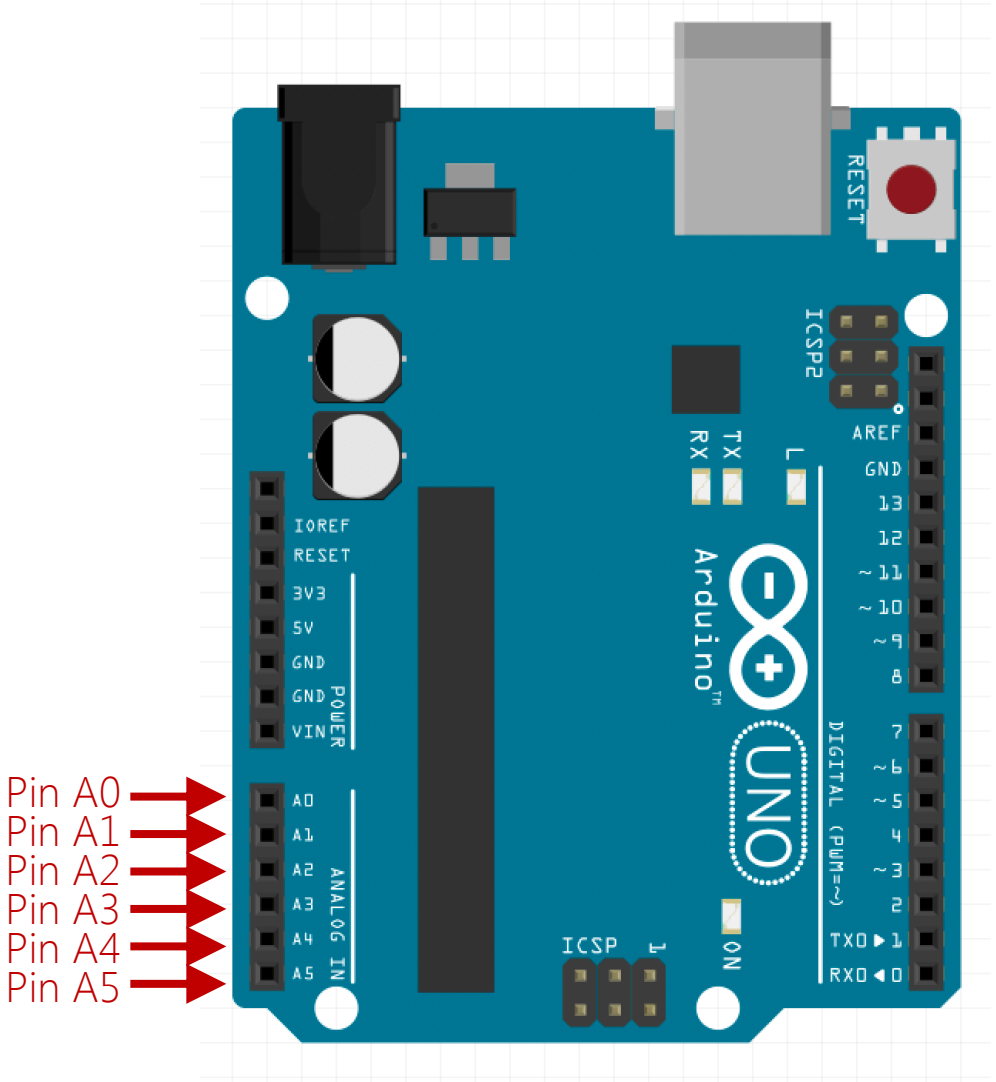
Video by Jon Froehlich

ACTIVITY: CONTROL LED BRIGHTNESS WITH TRIMPOT

Design circuit + software to read the potentiometer value on A0 and proportionally set the brightness of an LED on Pin 3



UNO AND LEONARDO BOARDS HAVE SIX ANALOG INPUTS



ANALOG INPUT

int analogRead(pin)

Reads the voltage from the specified analog pin & returns an integer (0 – 1023)

Syntax

`analogRead(pin)`

Parameters

`pin`: the analog pin to read from

Returns

An integer between 0 and 1023 (inclusive).

TrimpotLED

```
// The Arduino Uno ADC is 10 bits (thus, 0 - 1023 values)
#define MAX_ANALOG_INPUT_VAL 1023

const int LED_OUTPUT_PIN = 3;
const int POT_INPUT_PIN = A0;

void setup() {
  pinMode(LED_OUTPUT_PIN, OUTPUT);
  pinMode(POT_INPUT_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {

  // read the potentiometer value
  int potVal = analogRead(POT_INPUT_PIN);

  // the analogRead on the Arduino Uno goes from 0 to 1023. We need to remap
  // this value to the smaller range (0-255) since the analogWrite function can
  // only write out 0-255 (a byte--2^8). The map function provides a linear
  // mapping to do this (however, a better way would likely be some sort of
  // non-linear mapping given that perceived LED brightness is not linear with current,
  // perhaps logarithmic)
  int ledVal = map(potVal, 0, MAX_ANALOG_INPUT_VAL, 0, 255);

  // print the raw pot value and the converted led value
  Serial.print(potVal);
  Serial.print(",");
  Serial.println(ledVal);

  // write out the LED value. This value is translated to voltage by:
  // voltageVal = max_voltage * val/255 or voltageVal = 3.3V * val/255 in
  // the case of the RedBear Duo
  analogWrite(LED_OUTPUT_PIN, ledVal);

  delay(100);
}
```

https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

ANALOG INPUT

`int analogRead(pin)`

Reads the voltage value from the specified analog pin and returns it as an integer (0 – 1023)

ATmega microcontroller uses a **10-bit analog-to-digital-convert** (so, input voltages are mapped to 0-1023)

On a Uno or Leonardo, this yields a **resolution of 0.0049 volts** (5 volts / 1024 units)

The reading freq is about **~10,000Hz**

Reference > Language > Functions > Analog io > Analogread

analogRead()

[Analog I/O]

Description

Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023. On an Arduino UNO, for example, this yields a resolution between readings of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit. See the table below for the usable pins, operating voltage and maximum resolution for some Arduino boards.

The input range can be changed using analogReference(), while the resolution can be changed (only for Zero, Due and MKR boards) using analogReadResolution().

On ATmega based boards (UNO, Nano, Mini, Mega), it takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.

https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

ANALOG INPUT

int analogRead(pin)

Reads the voltage value from the specified analog pin and returns it as an integer (0 – 1023)

Syntax

`analogRead(pin)`

Parameters

`pin`: the analog pin to read from

Returns

An integer between 0 and 1023 (i

ANALOG OUTPUT

analogWrite(pin, value)

Writes an analog value between 0 and 5V to a pin using pulse-width modulation (PWM).

Syntax

`analogWrite(pin, value)`

Parameters

`pin`: the pin to write to.

`value`: an integer value between 0 & 255 which roughly maps to 0 – 5V on the Arduino Uno.

https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/

https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/

HELPER FUNCTION

map(value, fromLow, fromHigh, toLow, toHigh)

Remaps a number from one range to another. Warning: it does not constrain values to be within the range.

Syntax

`map(value, fromLow, fromHigh, toLow, toHigh)`

Parameters

`value`: the number to map

`fromLow`: the lower bound of the value's current range

`fromHigh`: the upper bound of the value's current range

`toLow`: the lower bound of the value's target range

`toHigh`: the upper bound of the value's target range

Returns

The mapped value

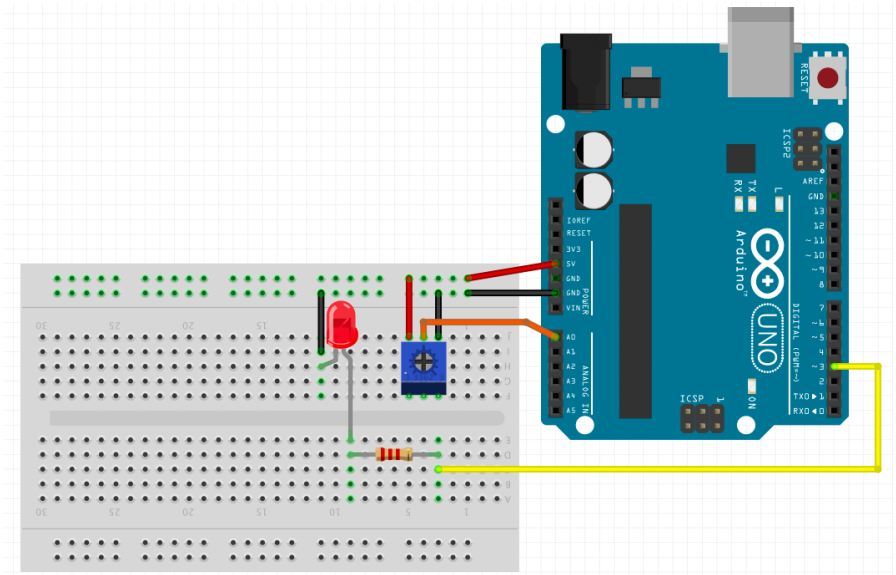
TrimpotLED

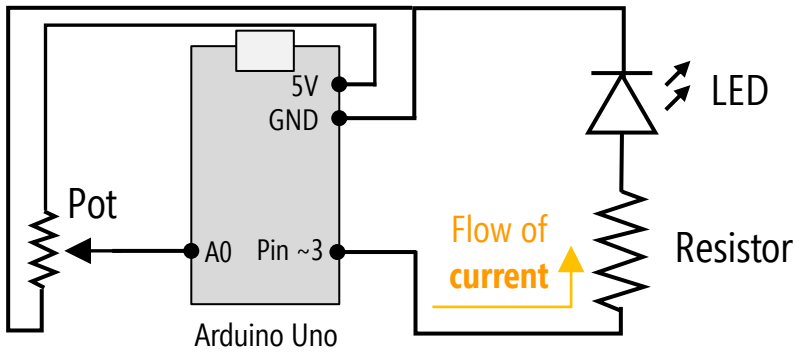
```
// The Arduino Uno ADC is 10 bits (thus, 0 - 1023 values)
#define MAX_ANALOG_INPUT_VAL 1023

const int LED_OUTPUT_PIN = 3;
const int POT_INPUT_PIN = A0;

void setup() {
  pinMode(LED_OUTPUT_PIN, OUTPUT);
  pinMode(POT_INPUT_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {

  // read the potentiometer value
  int potVal = analogRead(POT_INPUT_PIN);

  // the analogRead on the Arduino Uno goes from 0 to 1023. We need to remap
  // this value to the smaller range (0-255) since the analogWrite function can
  // only write out 0-255 (a byte--2^8). The map function provides a linear
  // mapping to do this (however, a better way would likely be some sort of
  // non-linear mapping given that perceived LED brightness is not linear with current,
  // perhaps logarithmic)
  int ledVal = map(potVal, 0, MAX_ANALOG_INPUT_VAL, 0, 255);

  // print the raw pot value and the converted led value
  Serial.print(potVal);
  Serial.print(",");
  Serial.println(ledVal);

  // write out the LED value. This value is translated to voltage by:
  // voltageVal = max_voltage * val/255 or voltageVal = 3.3V * val/255 in
  // the case of the RedBear Duo
  analogWrite(LED_OUTPUT_PIN, ledVal);

  delay(100);
}
```

https://www.arduino.cc/reference/en/language/functions/math/map/

ACTIVITY: CONTROL LED BRIGHTNESS WITH POTENTIOMETER

Circuit: Fade LED on/off via pot value





Code: Fade LED on/off via pot value

TrimpotLED

```
// The Arduino Uno ADC is 10 bits (thus, 0 - 1023 values)
#define MAX_ANALOG_INPUT_VAL 1023

const int LED_OUTPUT_PIN = 3;
const int POT_INPUT_PIN = A0;

void setup() {
  pinMode(LED_OUTPUT_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  // read the potentiometer value
  int potVal = analogRead(POT_INPUT_PIN);

  // the analogRead on the Arduino Uno goes from 0 to 1023. We need to remap
  // this value to the smaller range (0-255) since the analogWrite function can
  // only write out 0-255 (a byte--2^8). The map function provides a linear
  // mapping to do this (however, a better way would likely be some sort of
  // non-linear mapping given that perceived LED brightness is not linear with current,
  // perhaps logarithmic)
  int ledVal = map(potVal, 0, MAX_ANALOG_INPUT_VAL, 0, 255);

  // print the raw pot value and the converted led value
  Serial.print(potVal);
  Serial.print(",");
  Serial.println(ledVal);

  // write out the LED value. This value is translated to voltage by:
  // voltageVal = max_voltage * val/255 or voltageVal = 3.3V * val/255 in
  // the case of the RedBear Duo
  analogWrite(LED_OUTPUT_PIN, ledVal);

  delay(100);
}
```

TrimpotLED

```
// The Arduino Uno ADC is 10 bits (thus, 0 - 1023 values)
#define MAX_ANALOG_INPUT_VAL 1023

const int LED_OUTPUT_PIN = 3;
const int POT_INPUT_PIN = A0;

void setup() {
  pinMode(LED_OUTPUT_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  // read the potentiometer value
  int potVal = analogRead(POT_INPUT_PIN);

  // the analogRead on the Arduino Uno goes from 0 to 1023. We need to remap
  // this value to the smaller range (0-255) since the analogWrite function can
  // only write out 0-255 (a byte--2^8). The map function provides a linear
  // mapping to do this (however, a better way would likely be some sort of
  // non-linear mapping given that perceived LED brightness is not linear with current,
  // perhaps logarithmic)
  int ledVal = map(potVal, 0, MAX_ANALOG_INPUT_VAL, 0, 255);

  // print the raw pot value and the converted led value
  Serial.print(potVal);
  Serial.print(",");
  Serial.println(ledVal);

  // write out the LED value. This value is translated to voltage by:
  // voltageVal = max_voltage * val/255 or voltageVal = 3.3V * val/255 in
  // the case of the RedBear Duo
  analogWrite(LED_OUTPUT_PIN, ledVal);

  delay(100);
}
```

`int analogRead()`:Tips/Warnings

If the analog **input pin is not connected** to anything, the value returned by analogRead() will fluctuate based on a number of factors (*e.g.,* the values of the other analog inputs, how close your hand is to the board, *etc.*). Try it! ☺

https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/; https://github.com/firmata/arduino/issues/334; http://forum.arduino.cc/index.php?topic=69675.0

int analogRead():Tips/Warnings

If the analog **input pin is not connected** to anything, the value returned by analogRead() will fluctuate based on a number of factors (*e.g.,* the values of the other analog inputs, how close your hand is to the board, *etc.*). Try it! ☺

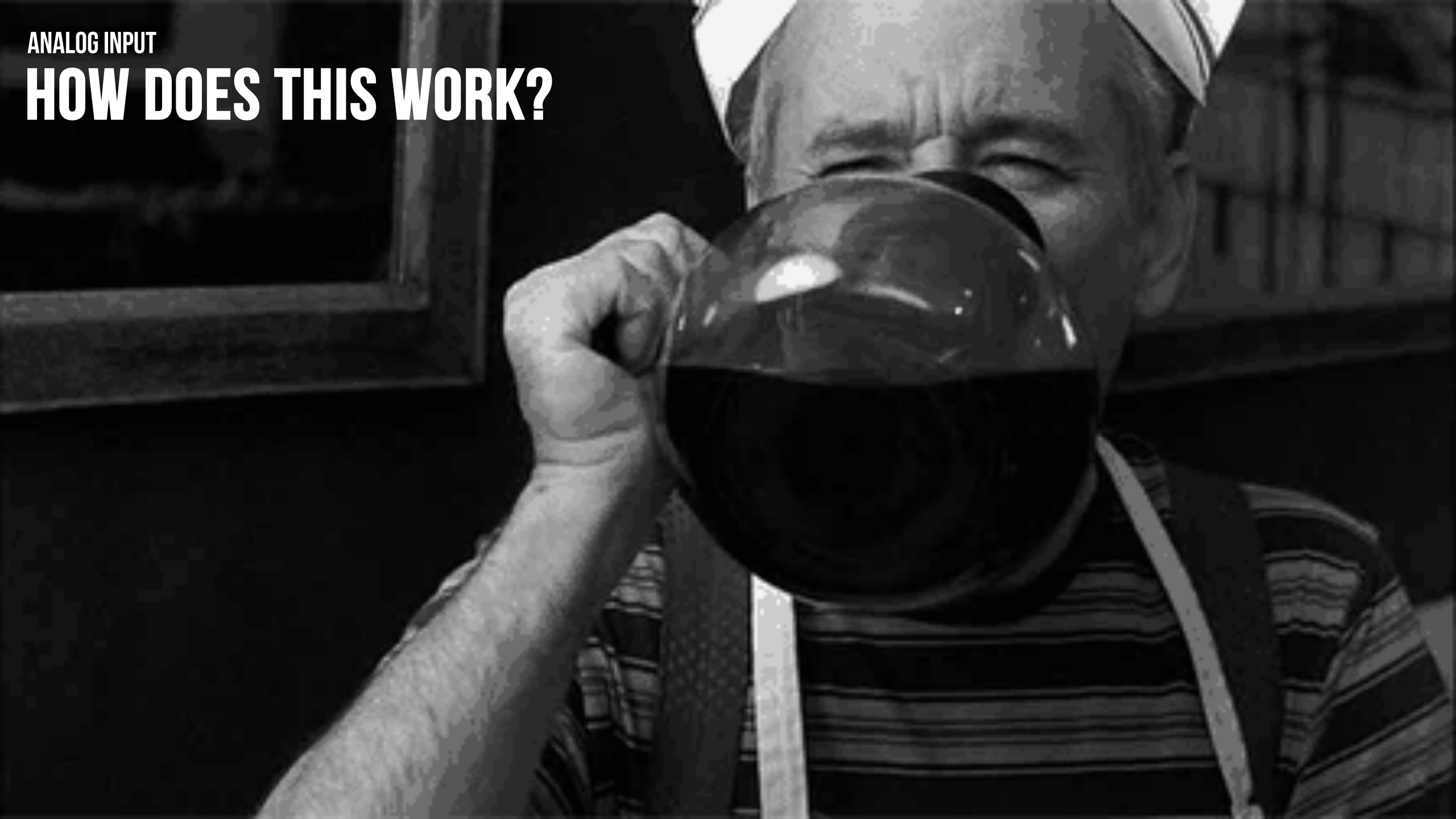
If you **execute multiple, consecutive** analogReads() **across different pins**, you might get noisy values. Instead:

- add a couple of ms delay between measurements on different pins
- perform two analogRead() on the same pin, dump the 1st & keep the 2nd

https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/; https://github.com/firmata/arduino/issues/334; http://forum.arduino.cc/index.php?topic=69675.0

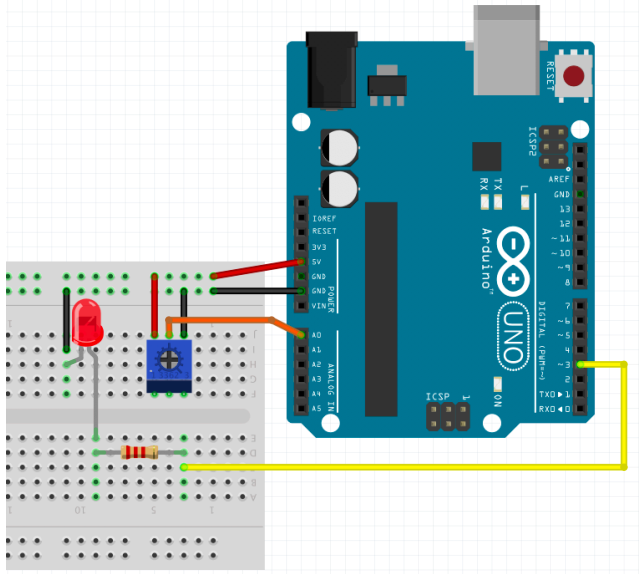
ANALOG INPUT

HOW DOES THIS WORK?

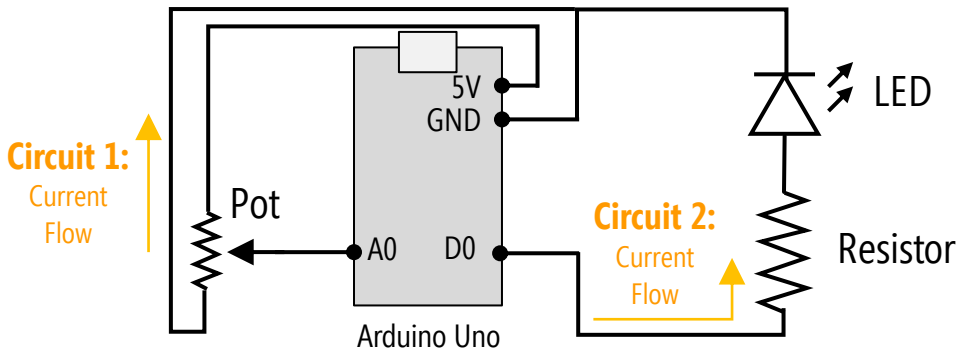


ANALOG INPUT

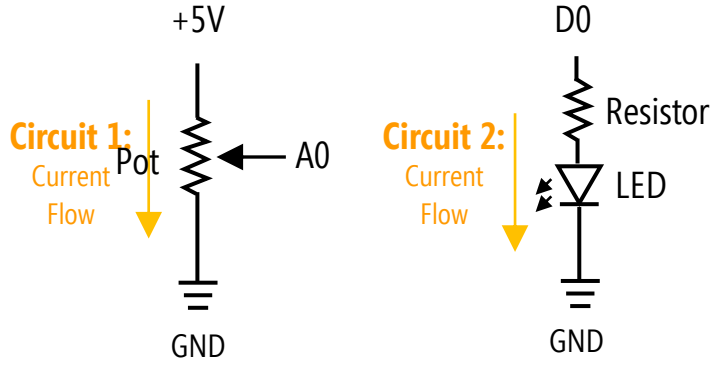
HOW DOES THIS WORK?



Let's redraw our circuit into an equivalent but simpler representation



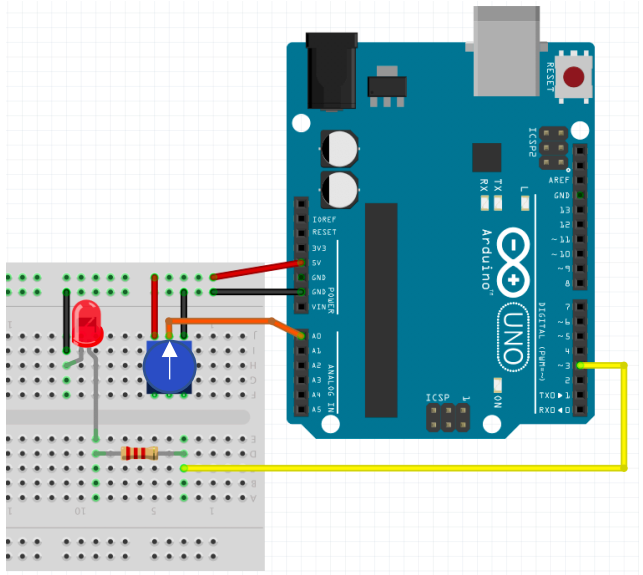
Schematic Representation 1



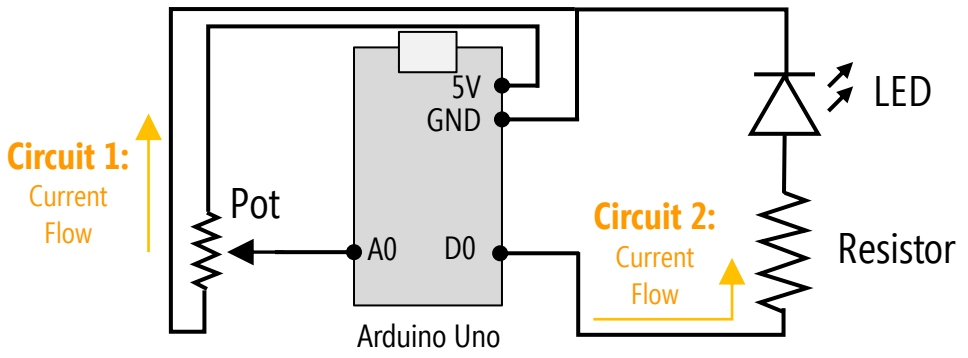
Schematic Representation 2

ANALOG INPUT

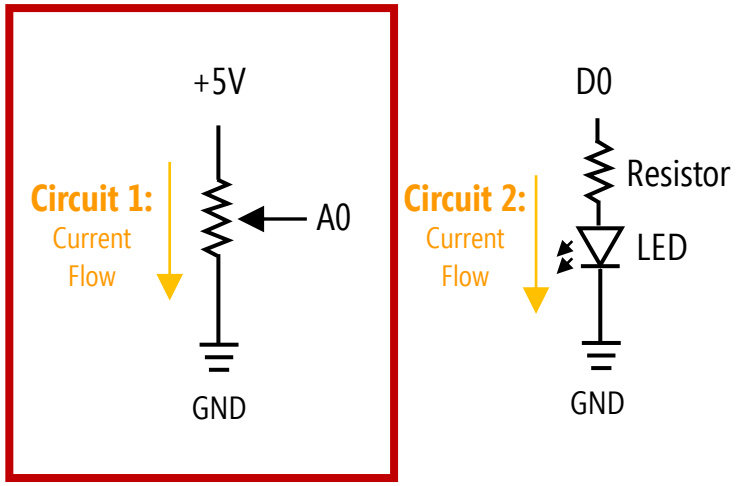
HOW DOES THIS WORK?



Let's redraw our circuit into an equivalent but simpler representation



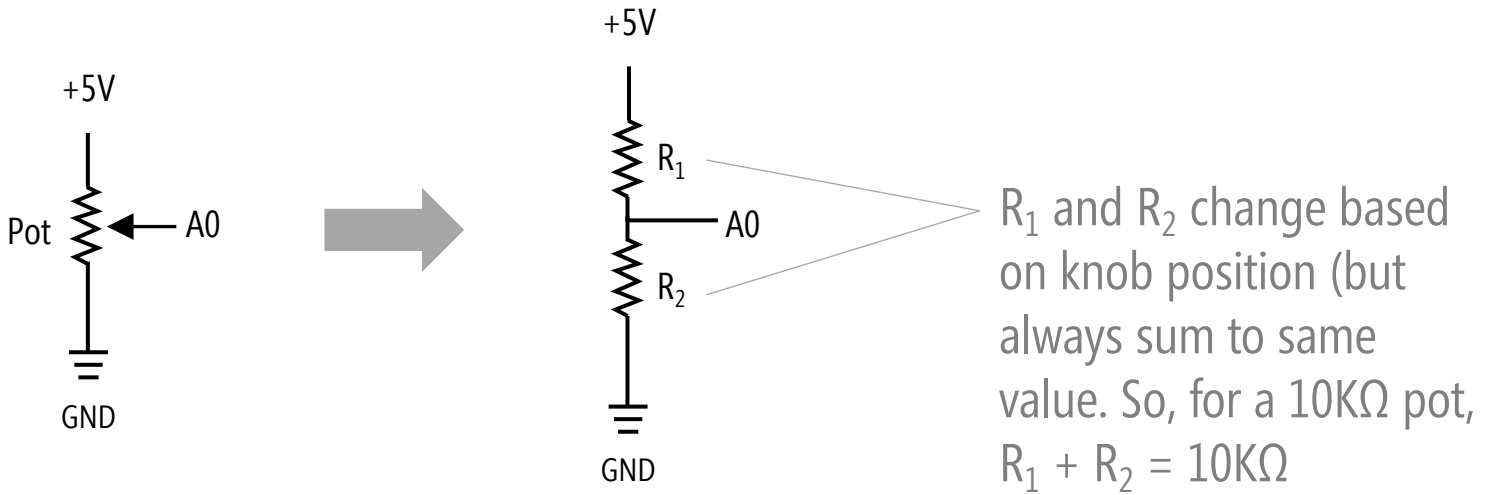
Schematic Representation 1



Schematic Representation 2

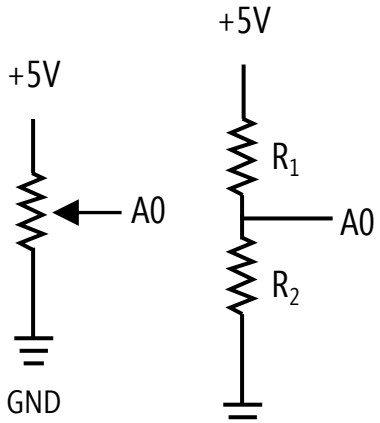
HOW DOES THIS WORK?

Another way to think about the potentiometer

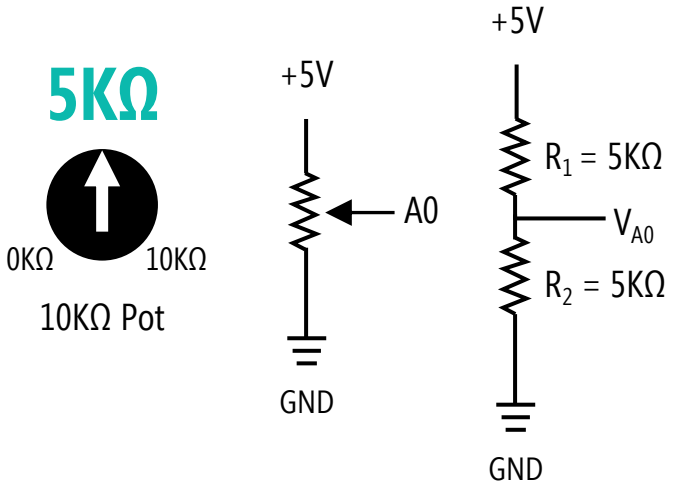


R_1 and R_2 change based on knob position (but always sum to same value. So, for a 10KΩ pot, $R_1 + R_2 = 10K\Omega$

WHAT IF THE POT IS SET TO 5KΩ?



WHAT IF THE POT IS SET TO 5KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{5{,}000\Omega + 5{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

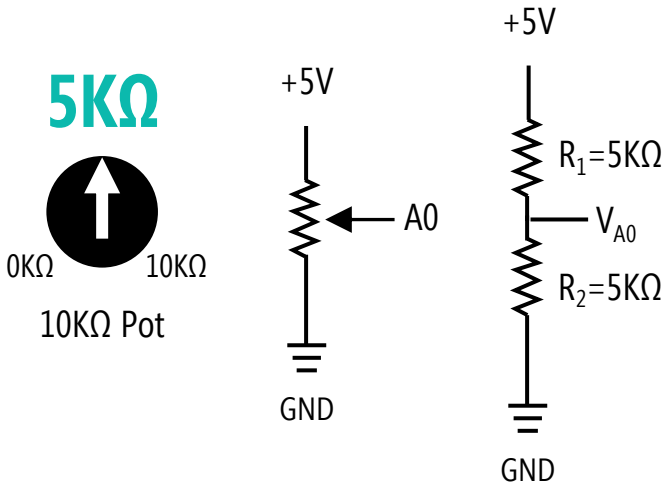
$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 5{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 2.5V$$

WHAT IF THE POT IS SET TO 5KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{5{,}000\Omega + 5{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

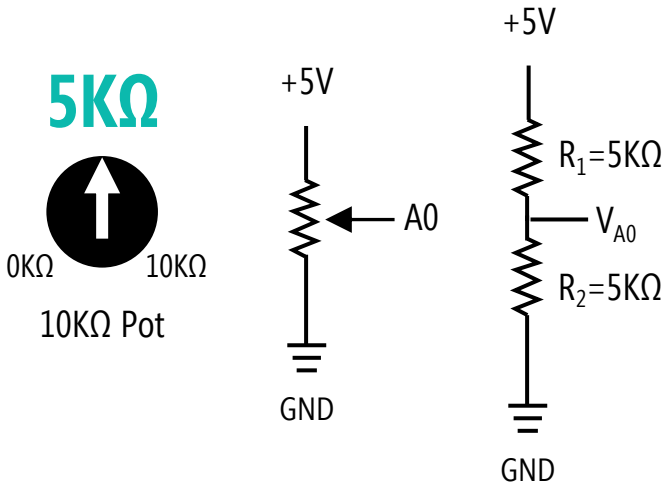
$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 5{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 2.5V$$

analogRead(A0) would return 512

WHAT IF THE POT IS SET TO 5KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{5{,}000\Omega + 5{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 5{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 2.5V$$

Why? Recall that the minimum and maximum analogRead values are 0 and 1023 respectively on the Uno. Thus:

$$\frac{2.5V}{5V} = \frac{x}{1023}$$

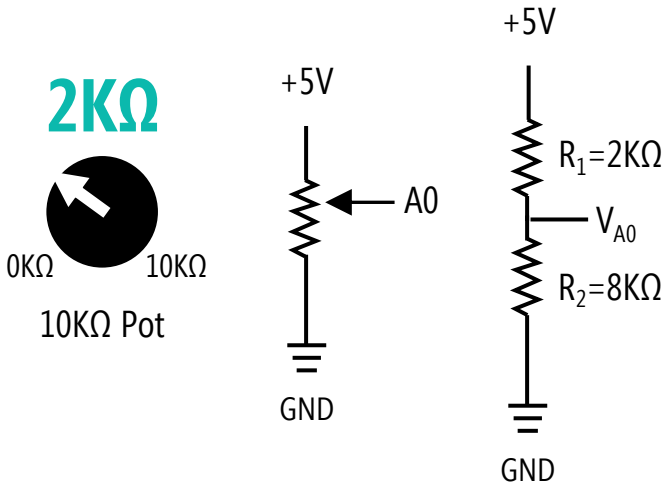
$$0.5 = \frac{x}{1023}$$

$$x = 0.5 * 1023 = 511.5$$

analogRead(A0) would return 512

ANALOG INPUT

WHAT IF THE POT IS SET TO 2KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

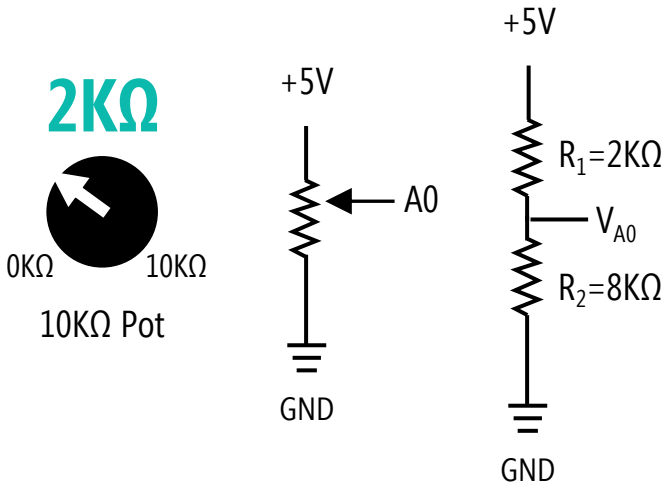
$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 2{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 4V$$

WHAT IF THE POT IS SET TO 2KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

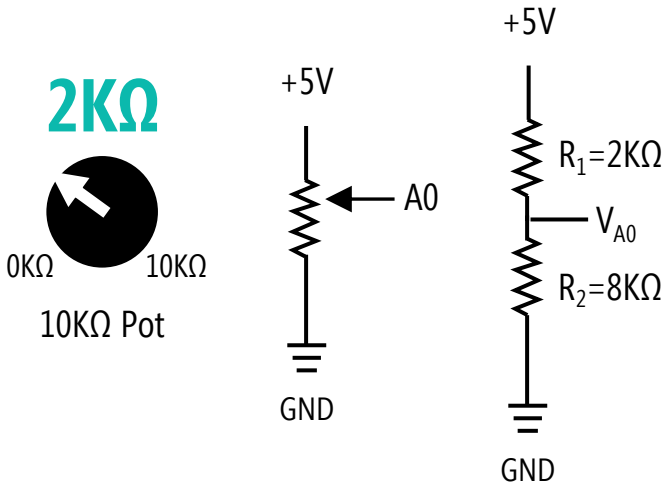
$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 2{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 4V$$

analogRead(A0) would return 818

ANALOG INPUT

WHAT IF THE POT IS SET TO 2KΩ?



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 2{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 4V$$

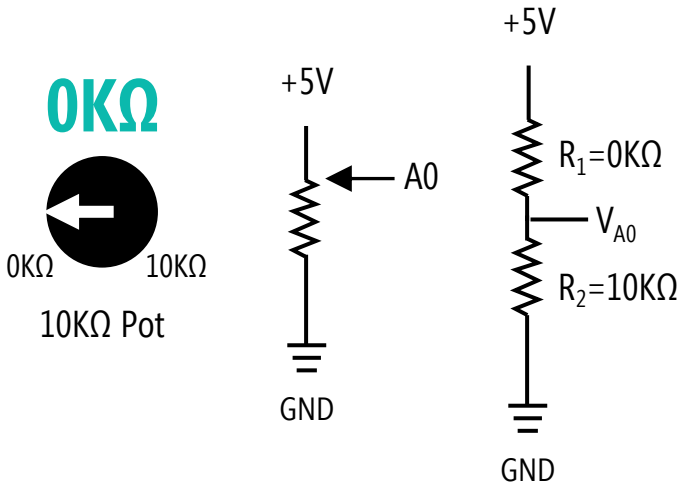
$$\frac{4V}{5V} = \frac{x}{1023}$$

$$0.8 = \frac{x}{1023}$$

$$x = 0.8 * 1023 = 818.4$$

analogRead(A0) would return **818**

WHAT IF THE POT IS SET TO 0KΩ



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

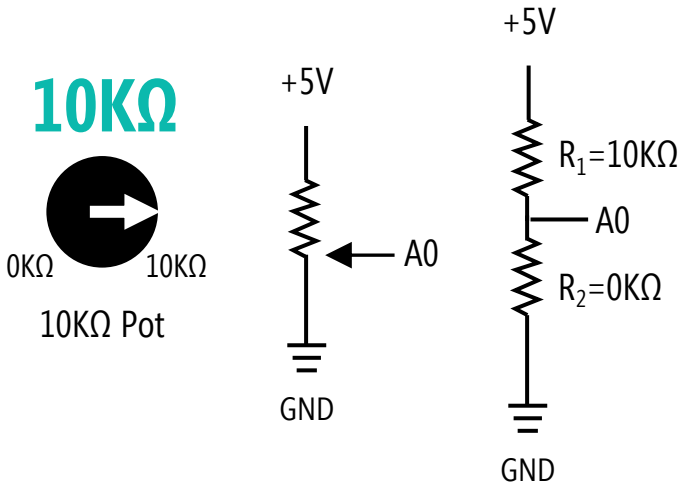
$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 0\Omega)$$

$$Voltage_{A0} = V_{A0} = 5V$$

analogRead(A0) would return **1023**

WHAT IF THE POT IS SET TO 10KΩ



What is A0 equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

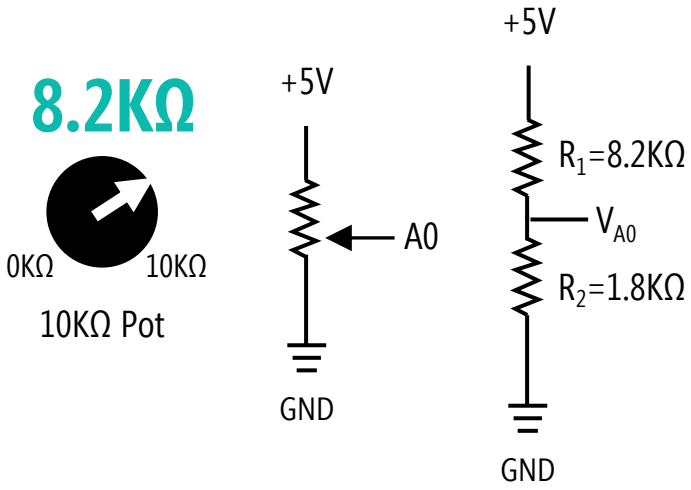
$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 10{,}000\Omega)$$

$$Voltage_{A0} = V_{A0} = 0V$$

analogRead(A0) would return **0**

ANALOG INPUT

WHAT IF THE POT IS SET TO 8.2KΩ



What is V_{A0} equal to?

$$\boldsymbol{Current = I = \frac{V_{high\ potential} - V_{low\ potential}}{R}}$$

$$Current = I = \frac{5V - 0V}{2{,}000\Omega + 8{,}000\Omega}$$

$$Current = I = 0.0005A = 0.5mA$$

$$\boldsymbol{Voltage = V = I * R}$$

$$Voltage_{A0} = V_{A0} = 5V - V_{R1}$$

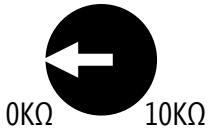
$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 8{,}200\Omega)$$

$$Voltage_{A0} = V_{A0} = 0.9V$$

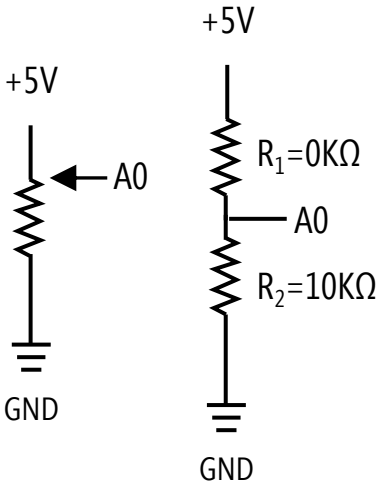
analogRead(A0) would return **184**

LET'S LOOK AT SOME VALUES ALL TOGETHER

0KΩ



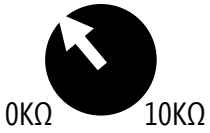
10KΩ Pot



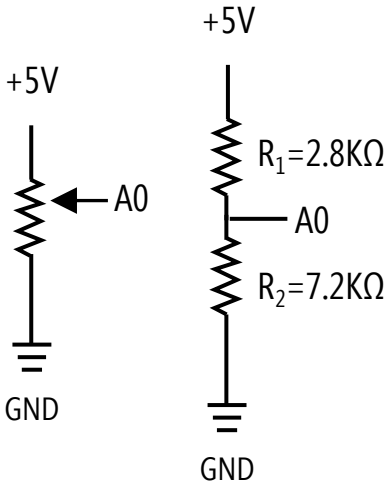
$Voltage_{A0} = V_{A0} = \mathbf{5V}$

analogRead(A0) returns **1023**

2.8KΩ



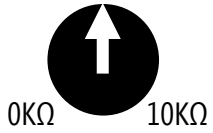
10KΩ Pot



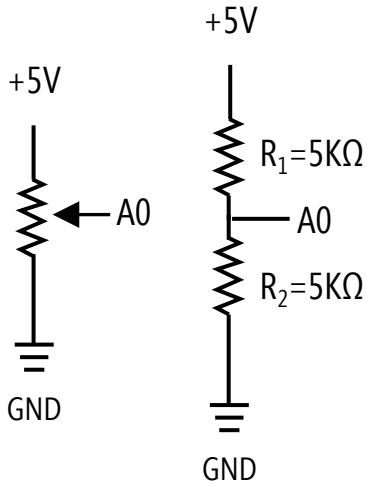
$Voltage_{A0} = V_{A0} = \mathbf{3.6V}$

analogRead(A0) returns **737**

5KΩ



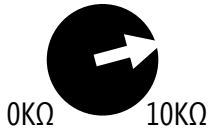
10KΩ Pot



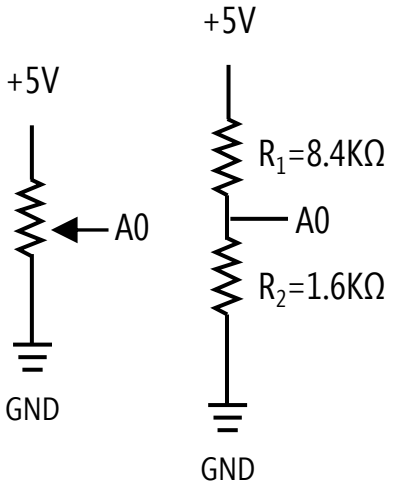
$Voltage_{A0} = V_{A0} = \mathbf{2.5V}$

analogRead(A0) returns **512**

8.4KΩ



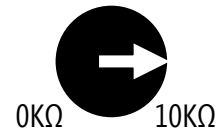
10KΩ Pot



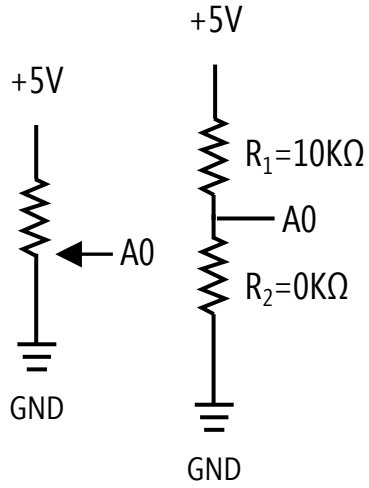
$Voltage_{A0} = V_{A0} = \mathbf{0.8V}$

analogRead(A0) returns **164**

10KΩ



10KΩ Pot



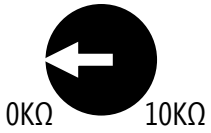
$Voltage_{A0} = V_{A0} = \mathbf{0V}$

analogRead(A0) returns **0**

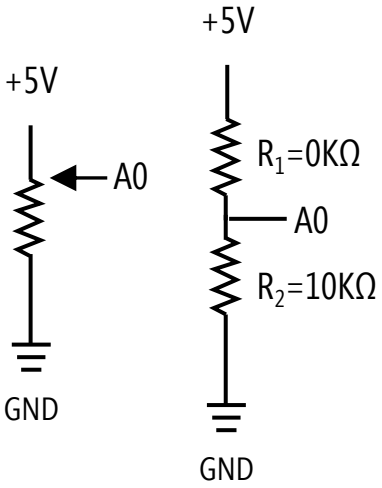
ANALOG INPUT

LET'S LOOK AT SOME VALUES ALL TOGETHER

0KΩ



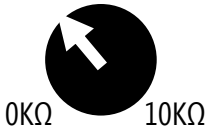
10KΩ Pot



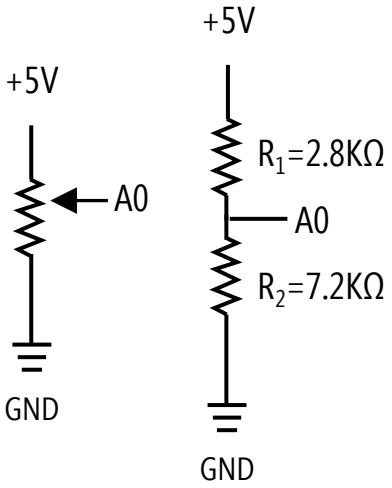
$Voltage_{A0} = V_{A0} = \mathbf{5V}$

analogRead(A0) returns **1023**

2.8KΩ



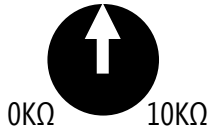
10KΩ Pot



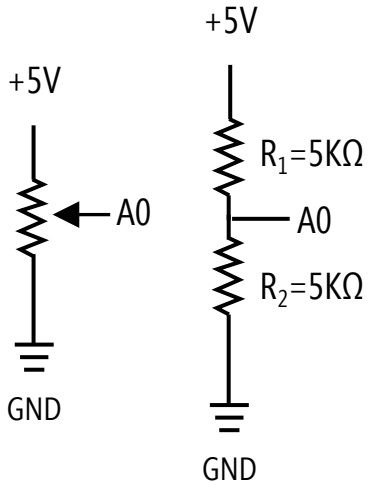
$Voltage_{A0} = V_{A0} = \mathbf{3.6V}$

analogRead(A0) returns **737**

5KΩ



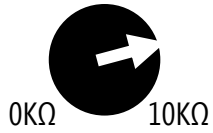
10KΩ Pot



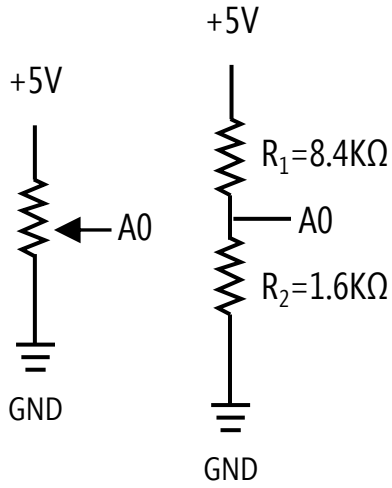
$Voltage_{A0} = V_{A0} = \mathbf{2.5V}$

analogRead(A0) returns **512**

8.4KΩ



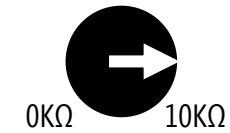
10KΩ Pot



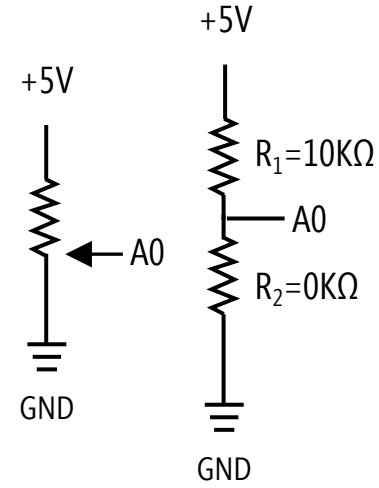
$Voltage_{A0} = V_{A0} = \mathbf{0.8V}$

analogRead(A0) returns **164**

10KΩ



10KΩ Pot



$Voltage_{A0} = V_{A0} = \mathbf{0V}$

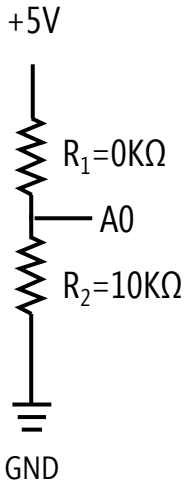
analogRead(A0) returns **0**

As R_1 grows big with respect to R_2, V_{A0} drops to zero

ANALOG INPUT

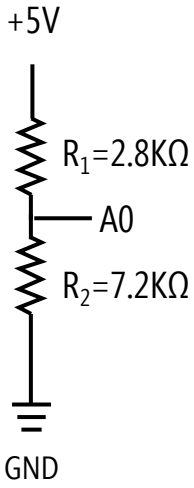
THIS TYPE OF CIRCUIT IS CALLED A “VOLTAGE DIVIDER”

It’s called a voltage divider because the voltage drops are divided across two resistors. A voltage divider is one of the primary ways that we configure our circuits to read sensor data with Arduino.



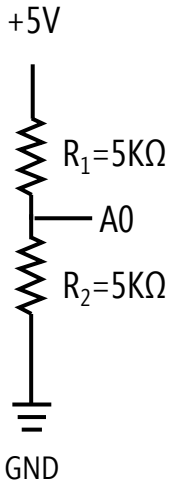
$Voltage_{A0} = V_{A0} = \mathbf{5V}$

analogRead(A0) returns **1023**



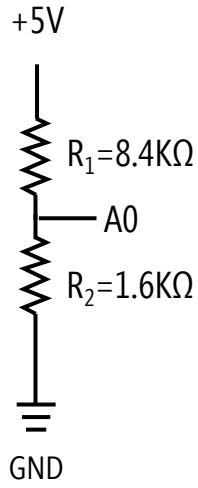
$Voltage_{A0} = V_{A0} = \mathbf{3.6V}$

analogRead(A0) returns **737**



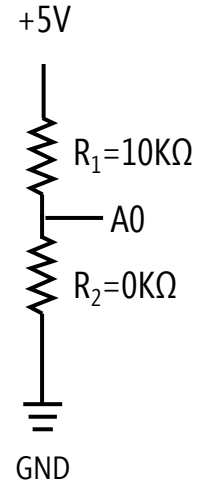
$Voltage_{A0} = V_{A0} = \mathbf{2.5V}$

analogRead(A0) returns **512**



$Voltage_{A0} = V_{A0} = \mathbf{0.8V}$

analogRead(A0) returns **164**

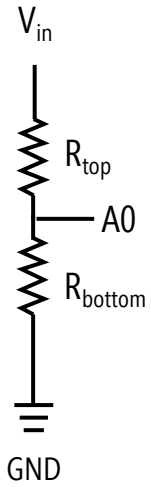


$Voltage_{A0} = V_{A0} = \mathbf{0V}$

analogRead(A0) returns **0**

ANALOG INPUT

VOLTAGE DIVIDER EQUATION



$$Voltage_{A0} = V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

Note that it's the **ratio** between the two resistors that's important.

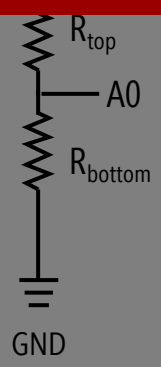
As R_{bottom} grows large with respect to R_{top}, V_{A0} grows proportionally large

Similarly, as R_{top} grows large with respect to R_{bottom}, V_{A0} grows proportionally small

ANALOG INPUT

VOLTAGE DIVIDER EQUATION

We covered this in our **1st Physical Computing** Lecture

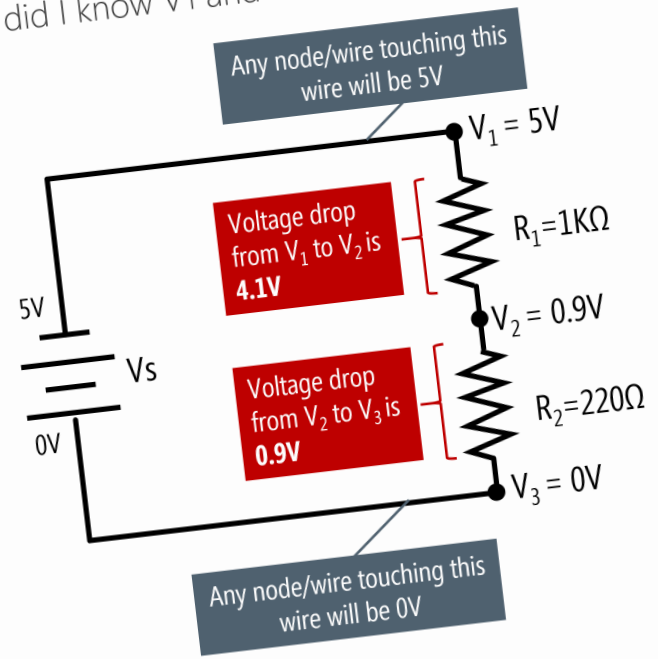


$Voltag$

SERIES VS. PARALLEL RESISTANCE

OHM'S LAW EXERCISE: SOLVE FOR V1, V2, AND V3

How did I know V1 and V3?



This basic circuit is called a **voltage divider** because it "splits" or "divides" voltages across nodes.

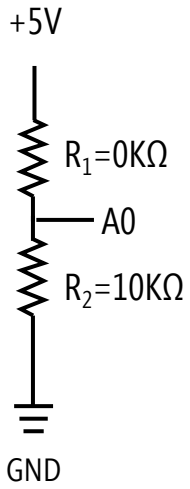
It is one of the **most common** (and useful) circuit configurations when working with microcontrollers

…n the

…ect to …rge

…respect to R_{bottom}, V_{A0} grows proportionally small

APPLYING THE“VOLTAGE DIVIDER” EQUATION



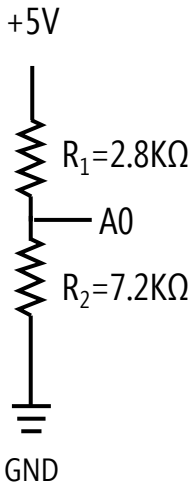
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

$$V_{A0} = 5V * \frac{10{,}000\Omega}{(0\Omega + 10{,}000\Omega)}$$

$$V_{A0} = 5V * 1$$

$$V_{A0} = \mathbf{5V}$$

analogRead(A0) returns **1023**



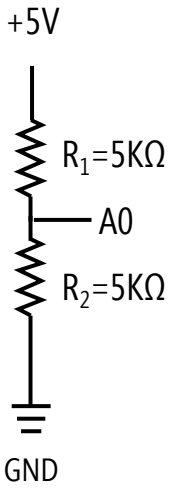
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

$$V_{A0} = 5V * \frac{7{,}200\Omega}{(2{,}800\Omega + 7{,}200\Omega)}$$

$$V_{A0} = 5V * 0.72$$

$$V_{A0} = \mathbf{3.6V}$$

analogRead(A0) returns **737**



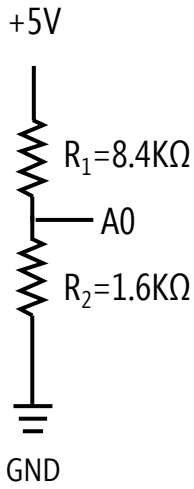
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

$$V_{A0} = 5V * \frac{5{,}000\Omega}{(5{,}000\Omega + 5{,}000\Omega)}$$

$$V_{A0} = 5V * 0.5$$

$$V_{A0} = \mathbf{2.5V}$$

analogRead(A0) returns **512**



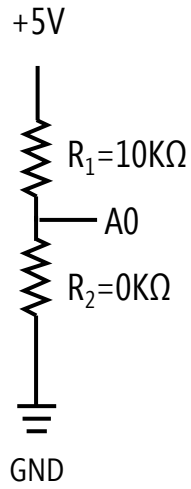
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

$$V_{A0} = 5V * \frac{1{,}600\Omega}{(8{,}400\Omega + 1{,}600\Omega)}$$

$$V_{A0} = 5V * 0.16$$

$$V_{A0} = \mathbf{0.8V}$$

analogRead(A0) returns **164**



$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

$$V_{A0} = 5V * \frac{0\Omega}{(10{,}000\Omega + 0\Omega)}$$

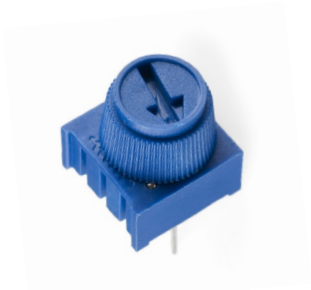
$$V_{A0} = 5V * 0$$

$$V_{A0} = \mathbf{0V}$$

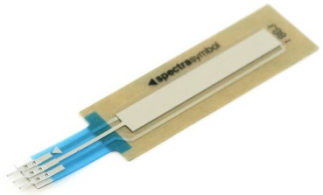
analogRead(A0) returns **0**

VARIABLE RESISTORS

LOTS OF DIFFERENT KINDS OF VARIABLE RESISTORS



Potentiometer 10k; $0.95*

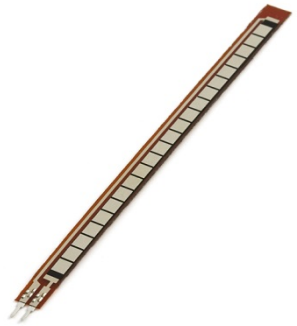


Touch Membrane Potentiometer; $12.95

Photocell (aka photodetector or photo resistor); $1.50

Thermistor 10k; $0.75

Force Resistive Sensor 0.5”; $6.95



Flex Sensor 4.5”; $12.95

*Note: Prices are from Sparkfun.com; parts are cheaper in bulk and often cheaper from Digikey.com