

PHYSCOMP 4: ANALOG INPUT

CSE 599 Prototyping Interactive Systems | Lecture 5 | Oct 10

Jon Froehlich • Liang He (TA)

TODAY'S LEARNING GOALS

Working with **analog input**

What is **a knob/trim potentiometer** and how to use it

What is **a slide potentiometer** and how to use it

What are **variable resistive sensors** and how to use them

Understanding the **importance of voltage dividers** when working with analog input (and the relevancy of Ohm's law!)

INTRODUCTION TO I/O

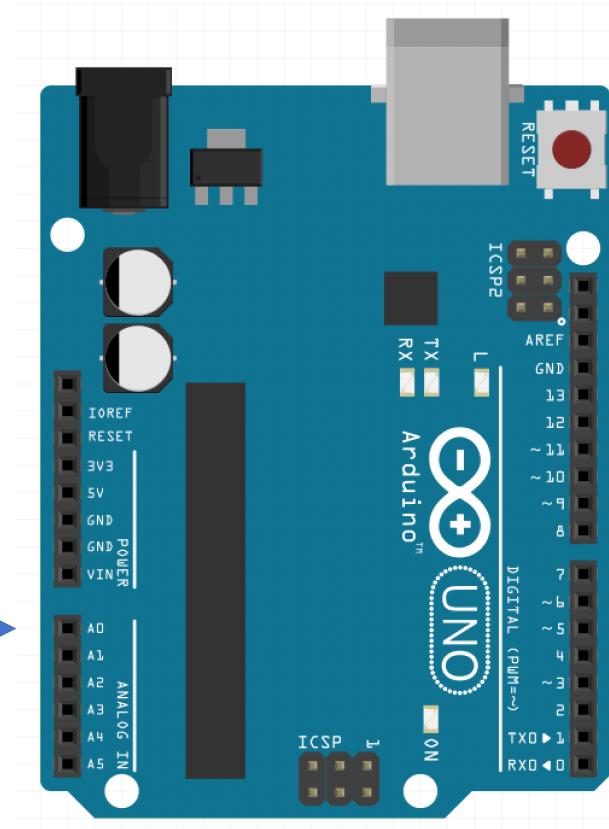
DIGITAL AND ANALOG OUTPUT

Now we are going to switch from talking about output to talking about input



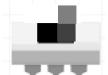
Analog Input on A0

Reads in any value between 0V or 5V using the analogRead function. In this case, the value of a photocell



Digital Input on pin 13

Reads in a digital input signal (anything below 2.5V converted to LOW, anything above 2.5V converted to HIGH). In this case, the value of switch.



Digital Output on pin 8

Writes out 0V or 5V using digitalWrite function. In this case, turning on and off an LED.



Analog Output on pin 3

Writes out any value between 0V or 5V using analogWrite function. In this case, vibrating a motor (where strength of vibration proportional to voltage). Only pins with a tilde ~ can be analog outputs.



REVISITING POTENTIOMETERS



VARIABLE RESISTORS

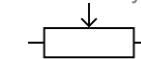
WHAT IS A POTENTIOMETER?



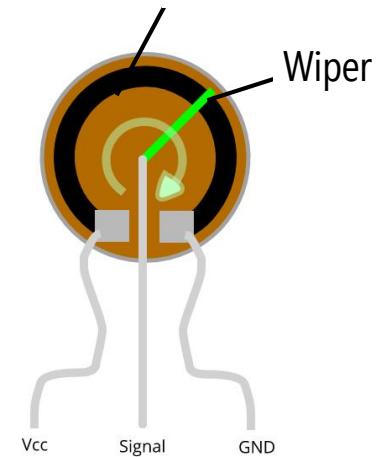
American-Style Symbol



International-Style Symbol



Resistive material

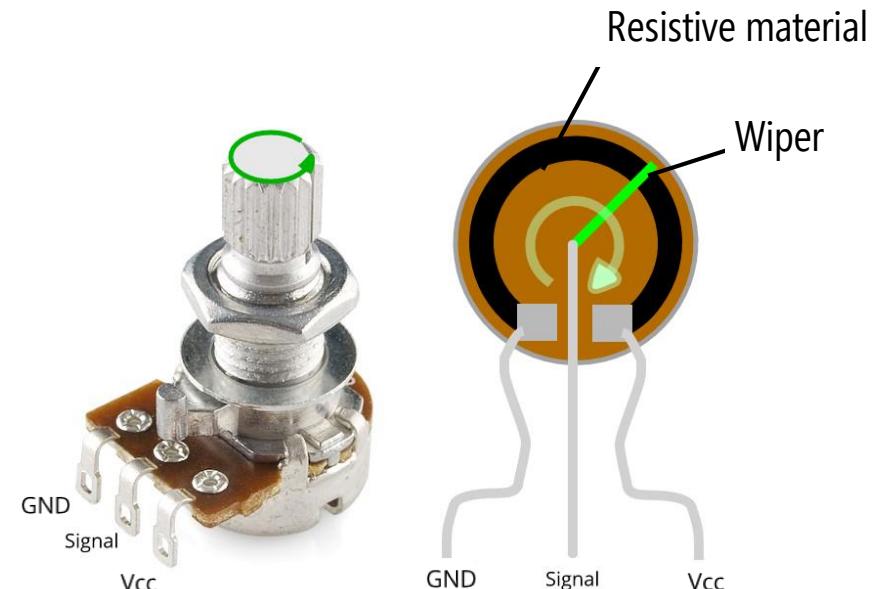
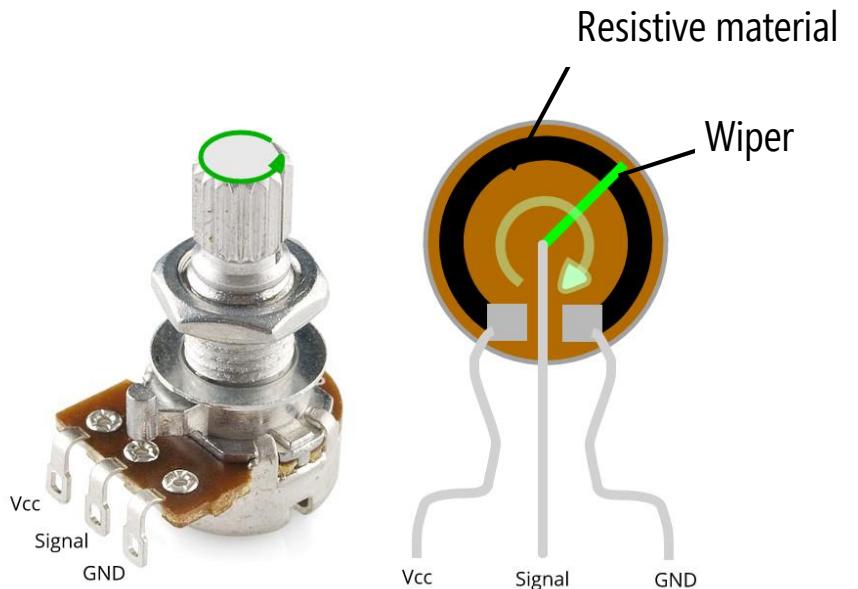


A **three-terminal resistor** with a sliding or rotating contact that forms an adjustable voltage divider

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

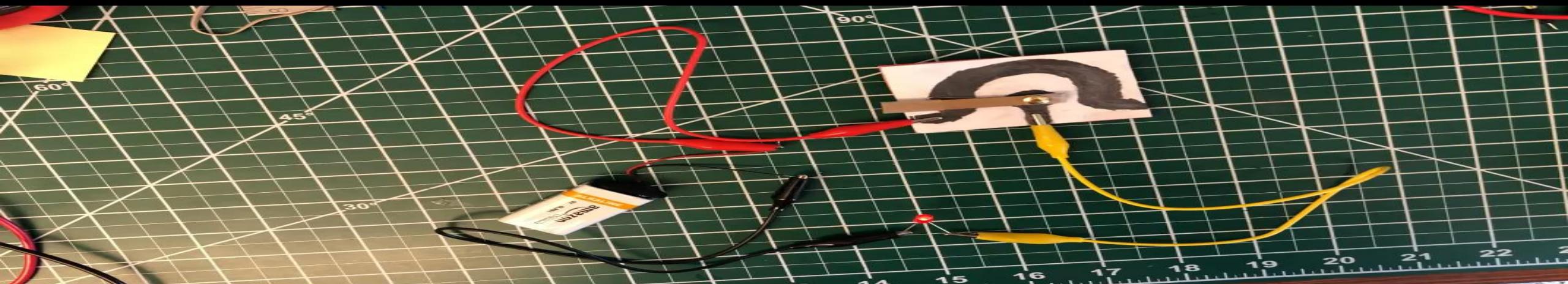
POTENTIOMETERS

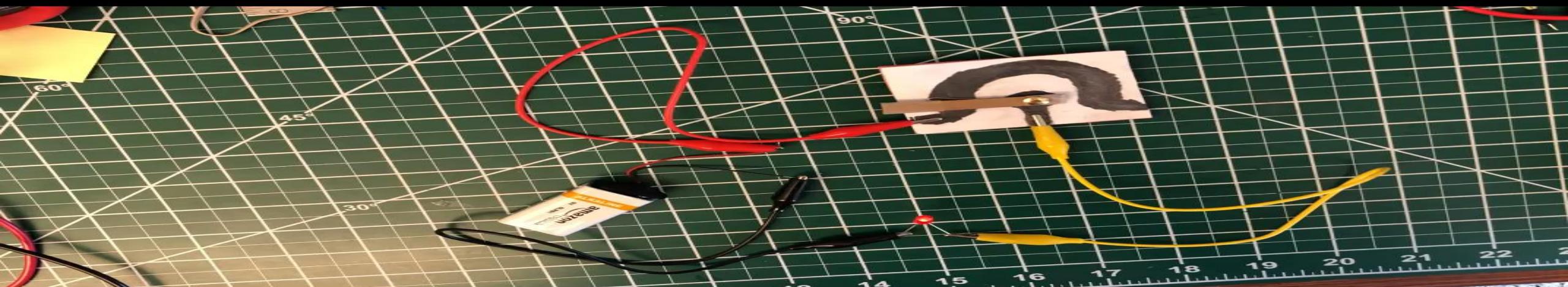
POTENTIOMETERS CAN BE HOOKED UP IN EITHER ORIENTATION



POTENTIOMETERS

DEMO OF LO-FI POTENTIOMETER



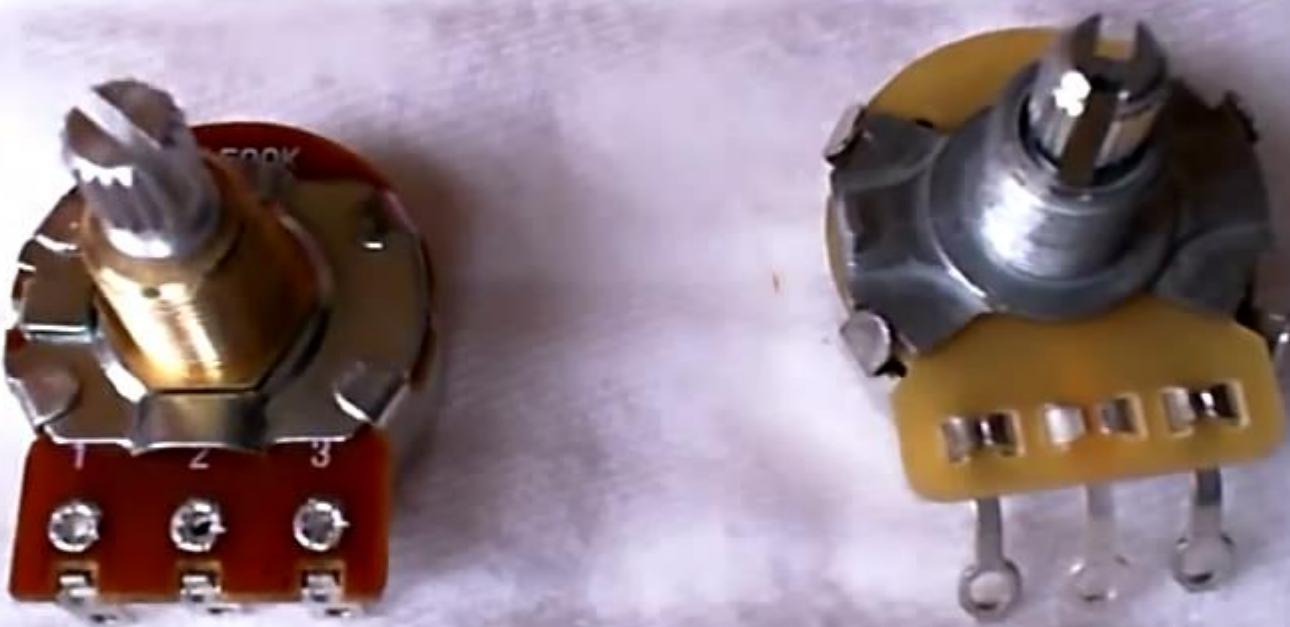


POTENTIOMETERS

INSIDE A POTENTIOMETER



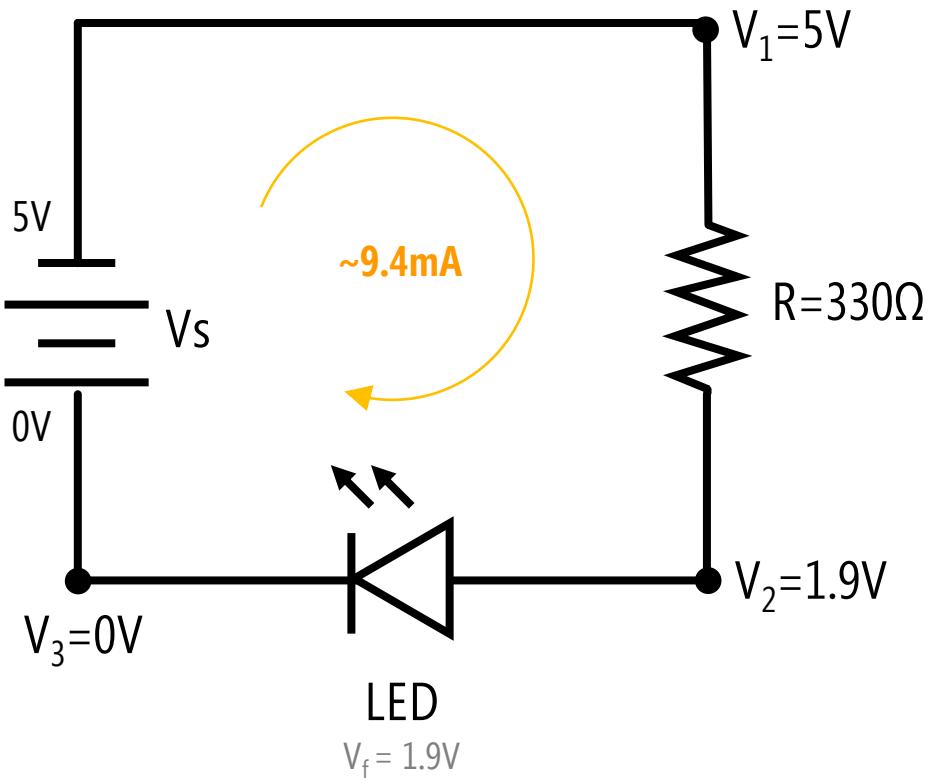
CTS EP086 500k



CTS EP086 500k

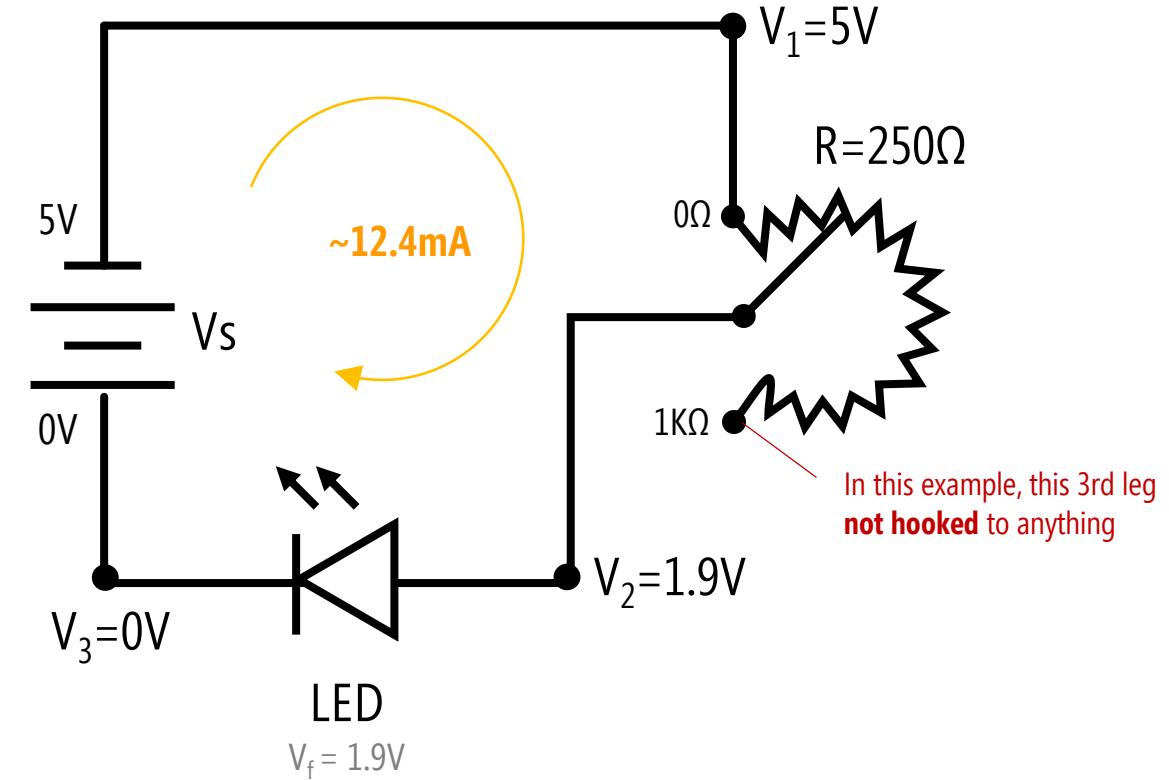
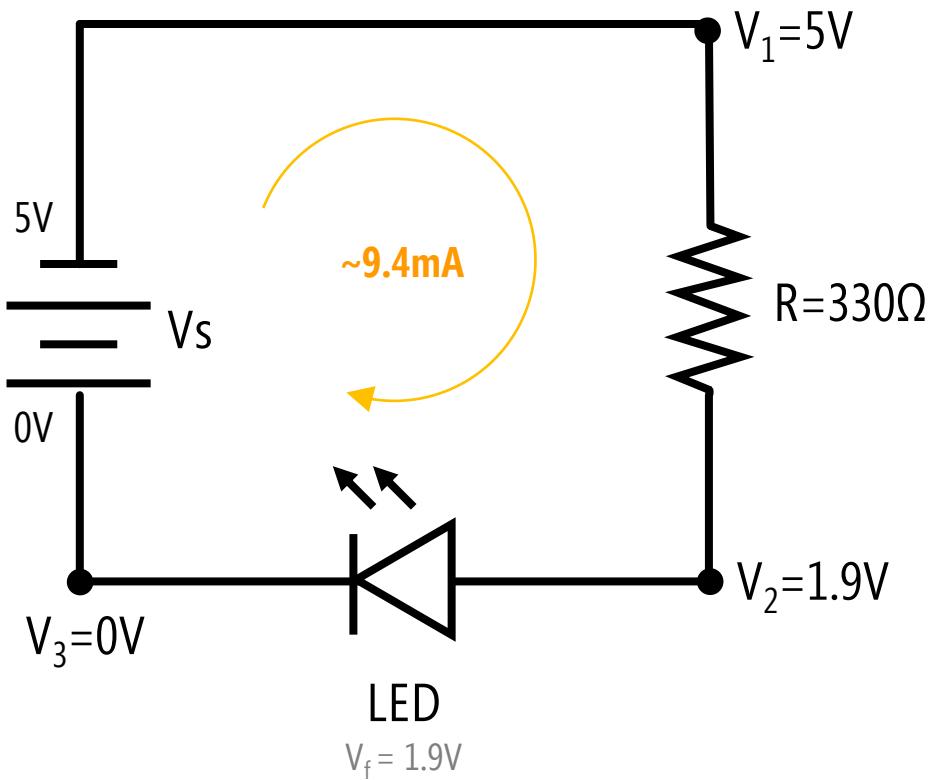
POTENTIOMETER

RESISTOR VS. POTENTIOMETER



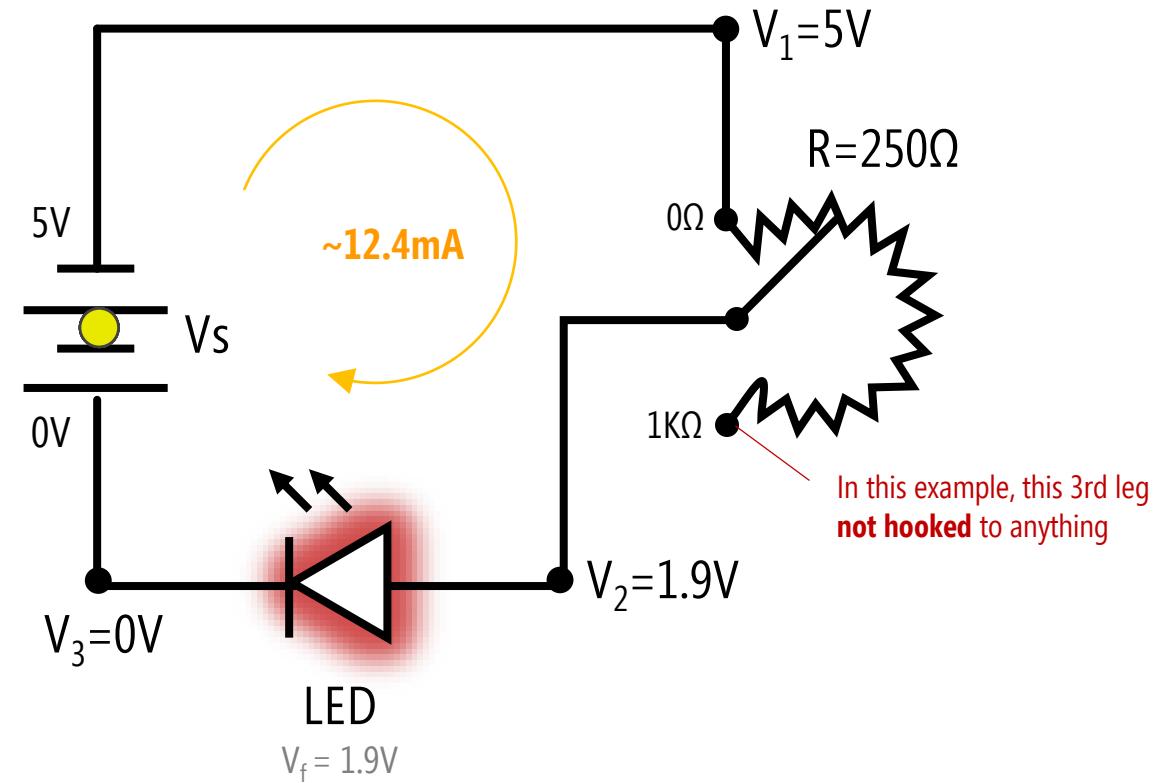
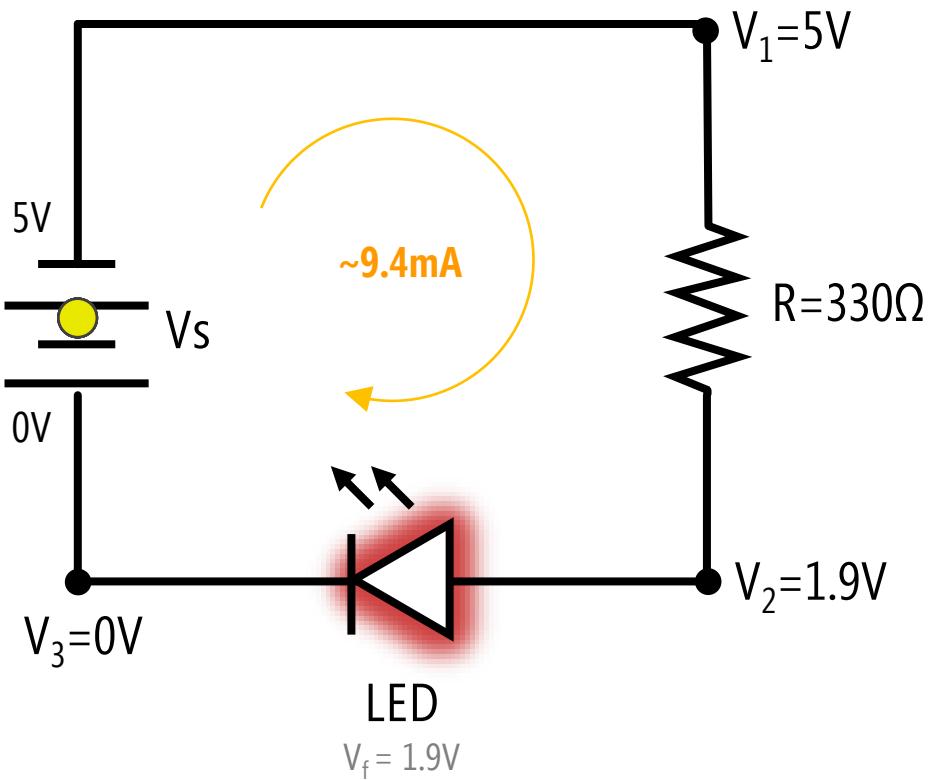
POTENTIOMETER

RESISTOR VS. POTENTIOMETER



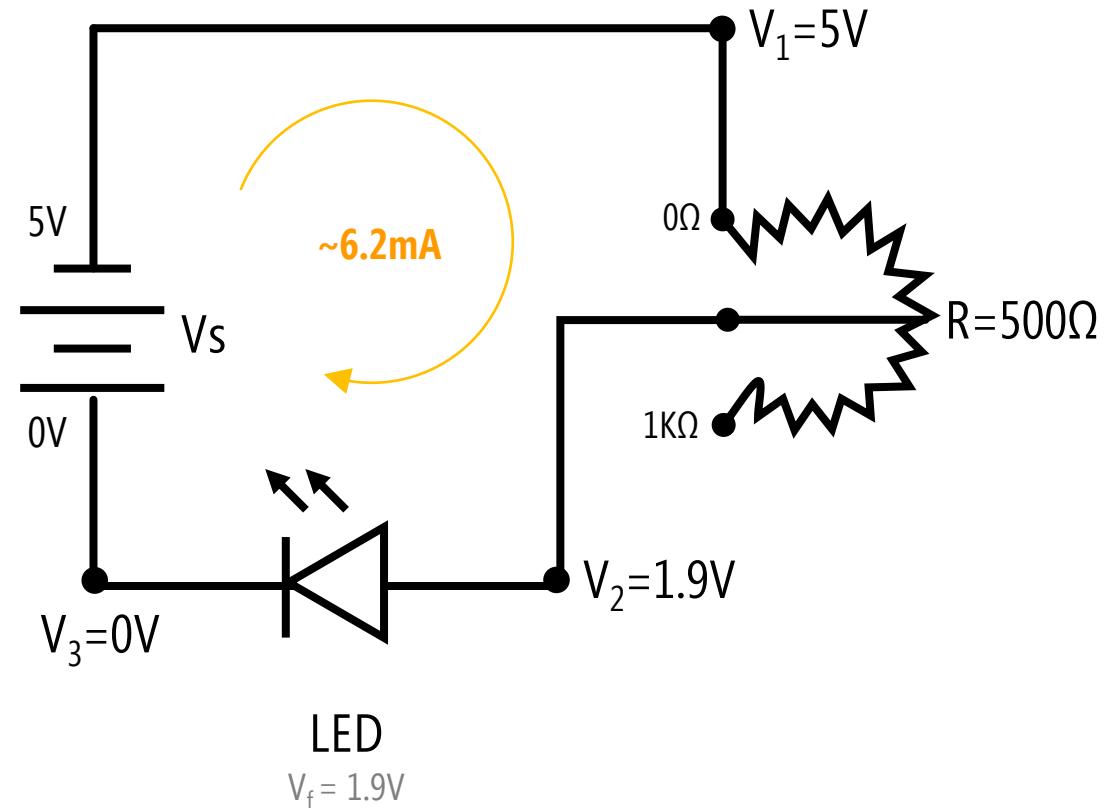
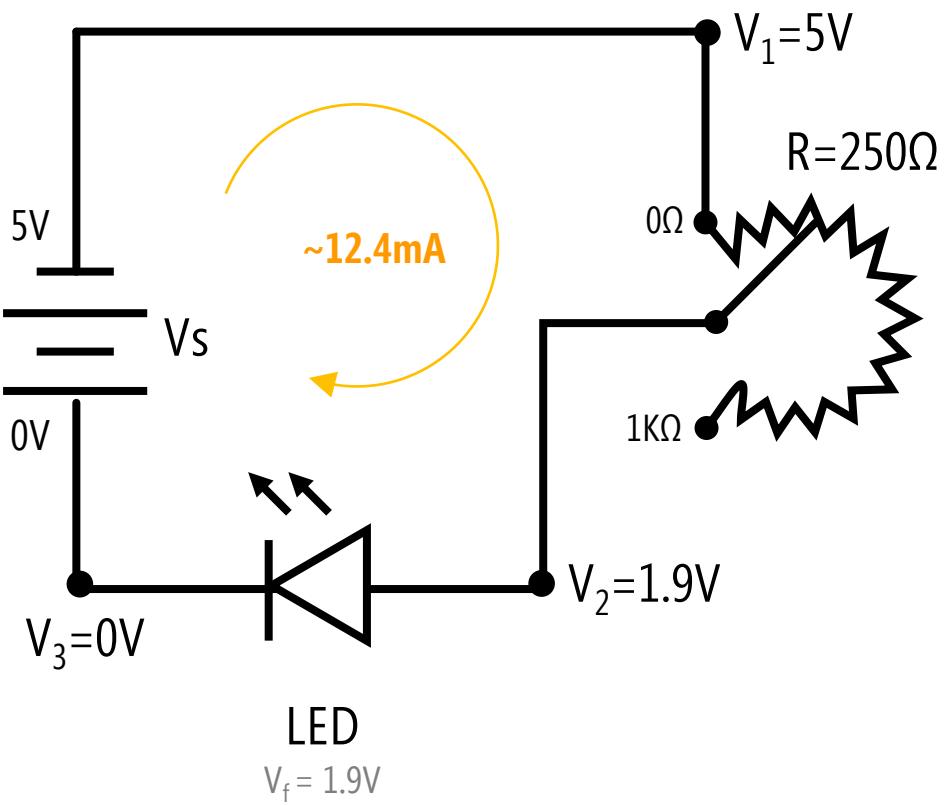
POTENTIOMETER

RESISTOR VS. POTENTIOMETER



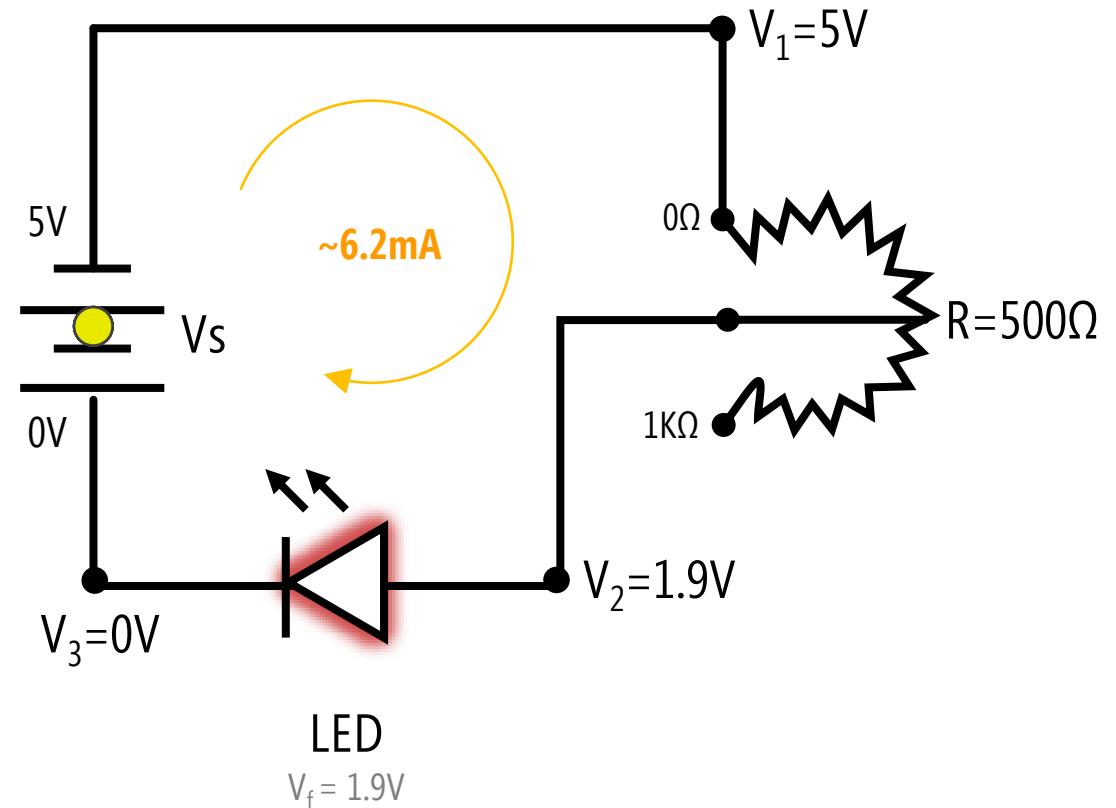
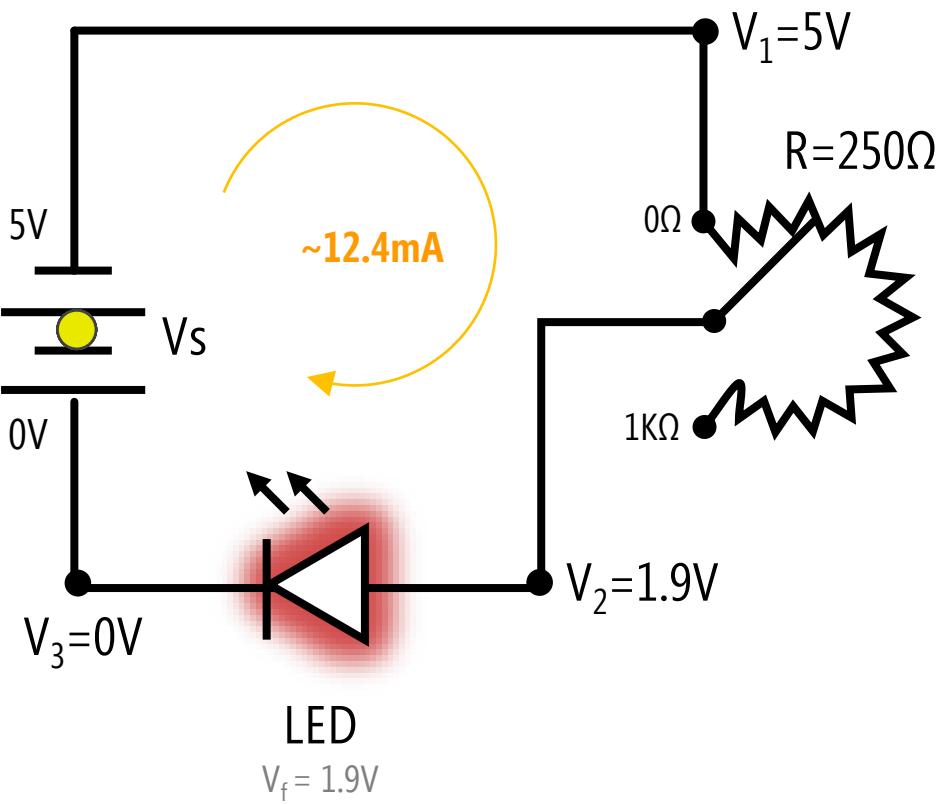
POTENTIOMETER

POTENTIOMETER EXAMPLES



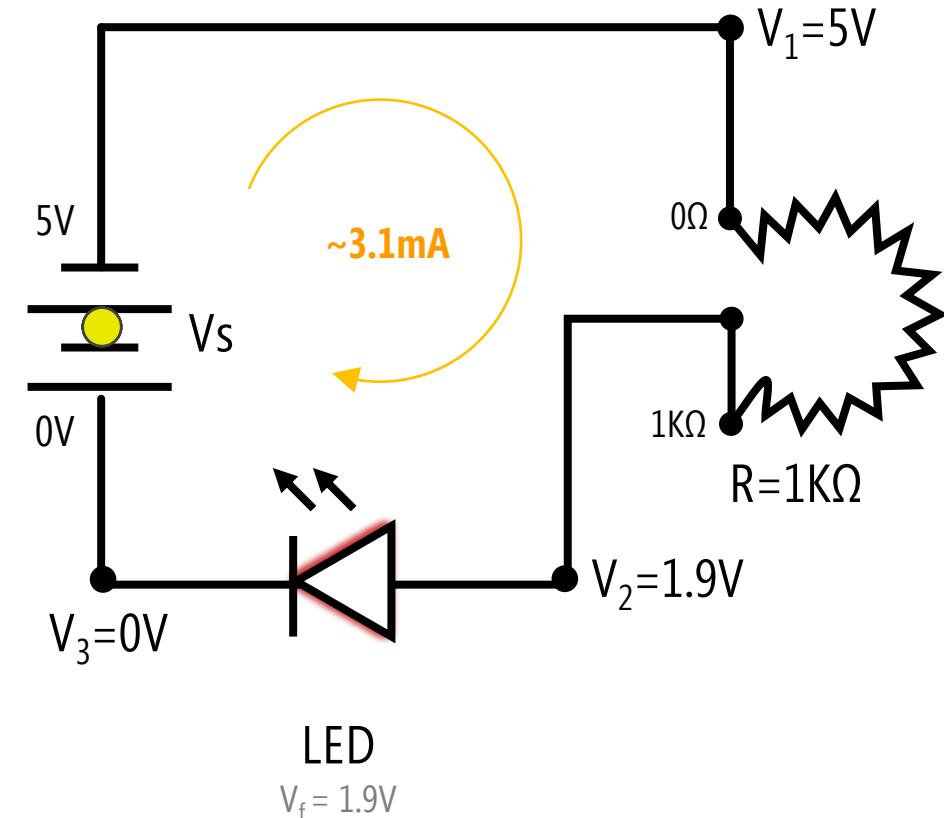
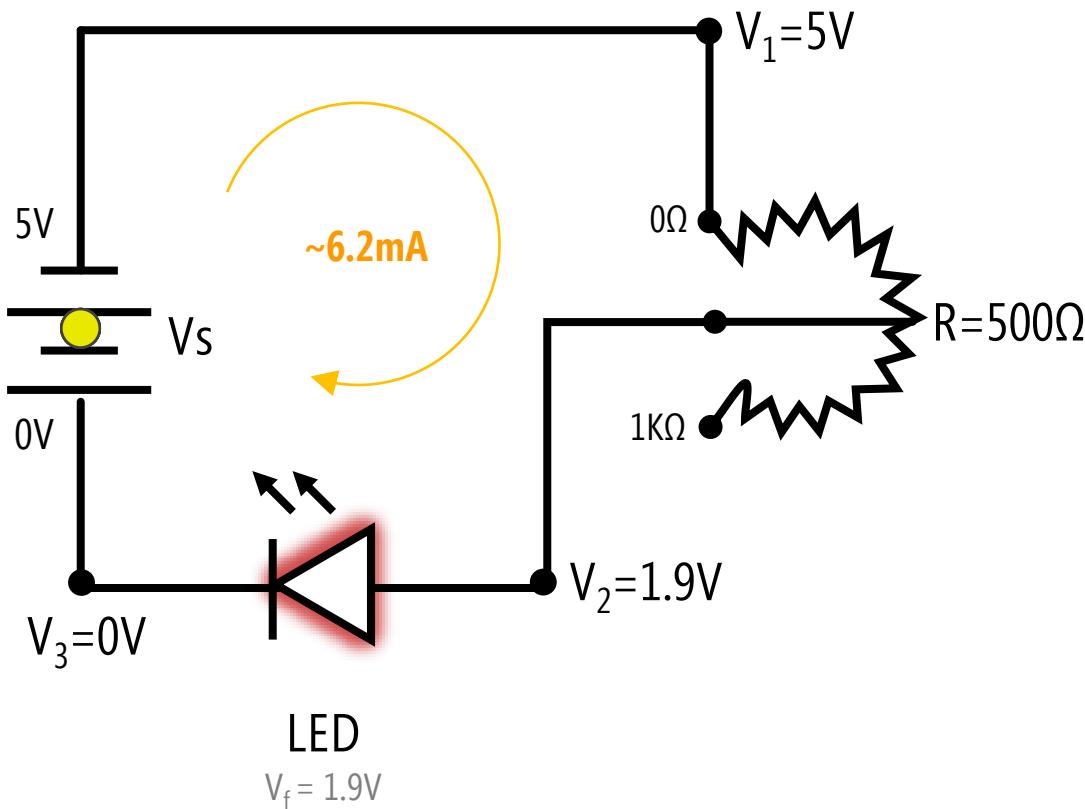
POTENTIOMETER

POTENTIOMETER EXAMPLES



POTENTIOMETER

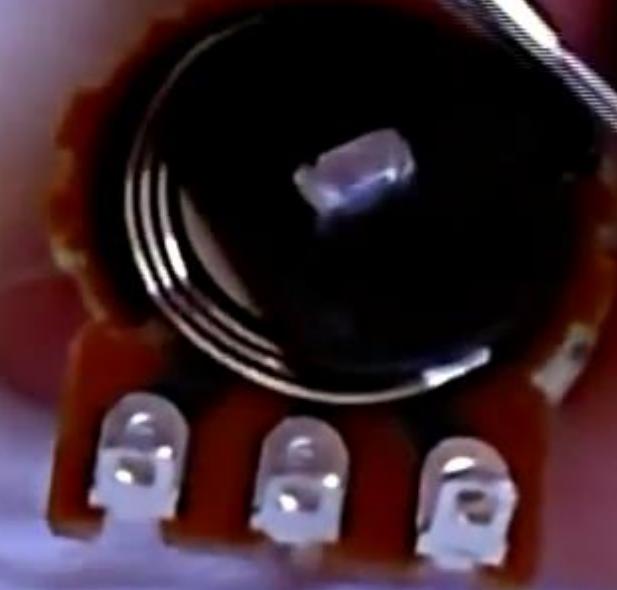
POTENTIOMETER EXAMPLES



POTENTIOMETERS

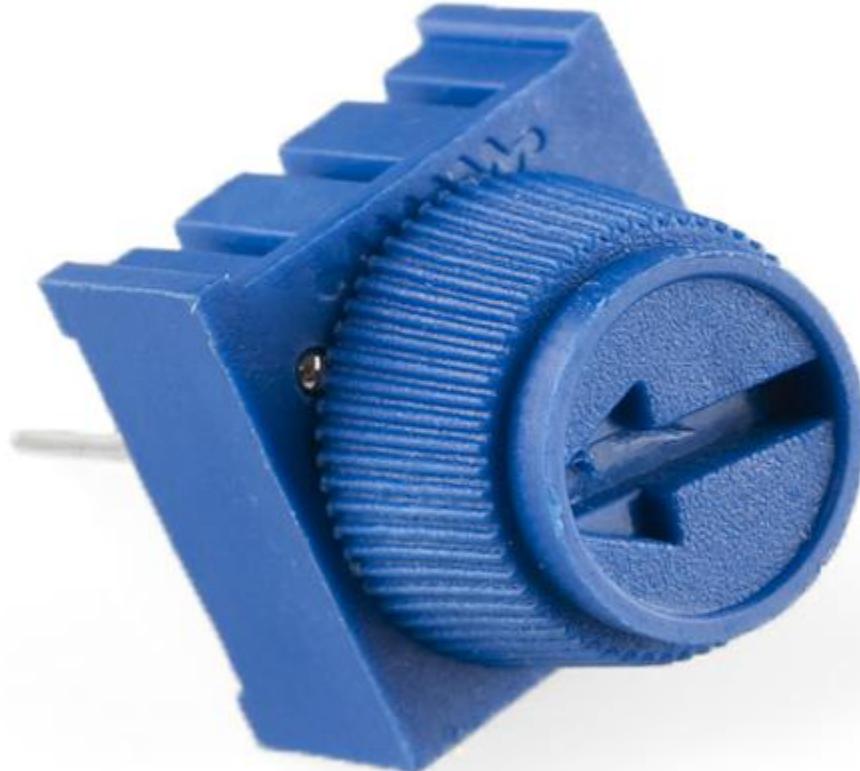
LINEAR VS. EXPONENTIAL RESISTANCE





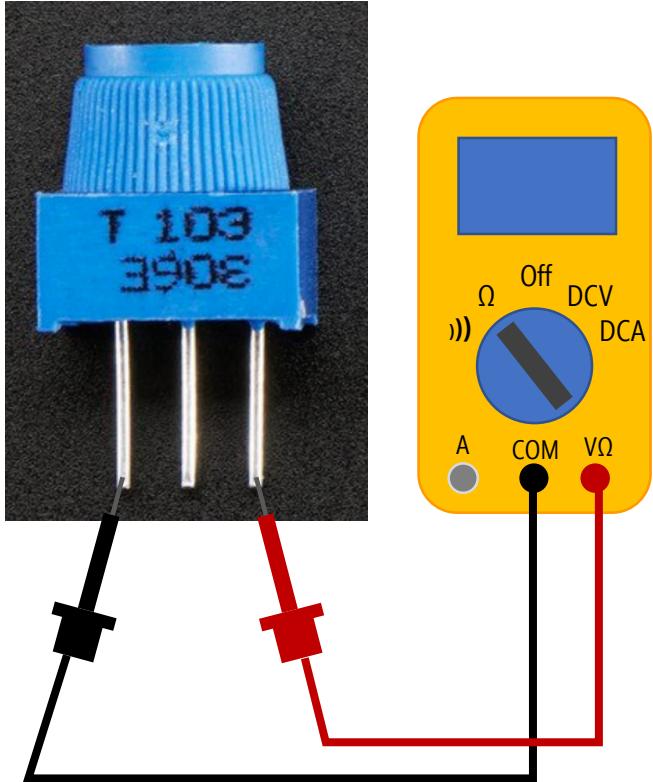
VARIABLE RESISTORS

BREADBOARD-FRIENDLY TRIMPOT

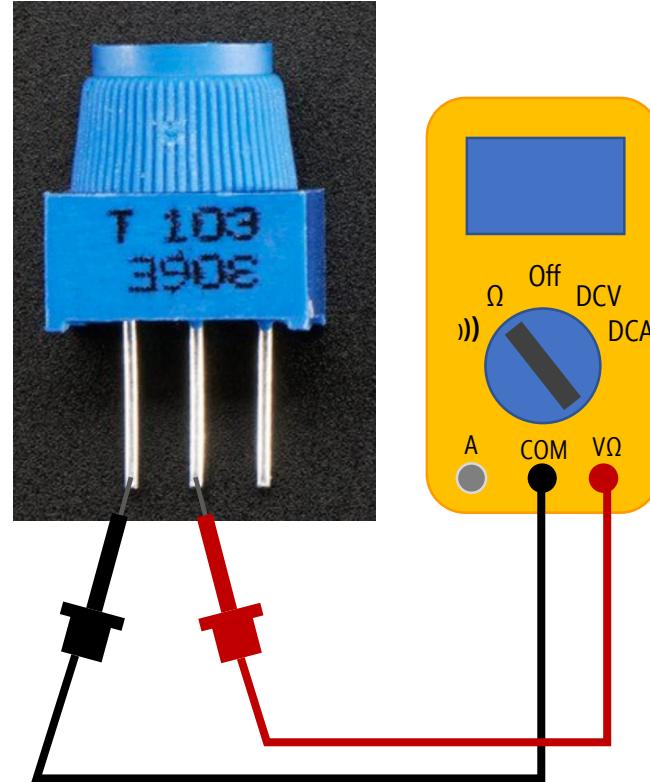


POTENTIOMETERS

ACTIVITY: EXPLORE THE POT WITH A MULTIMETER



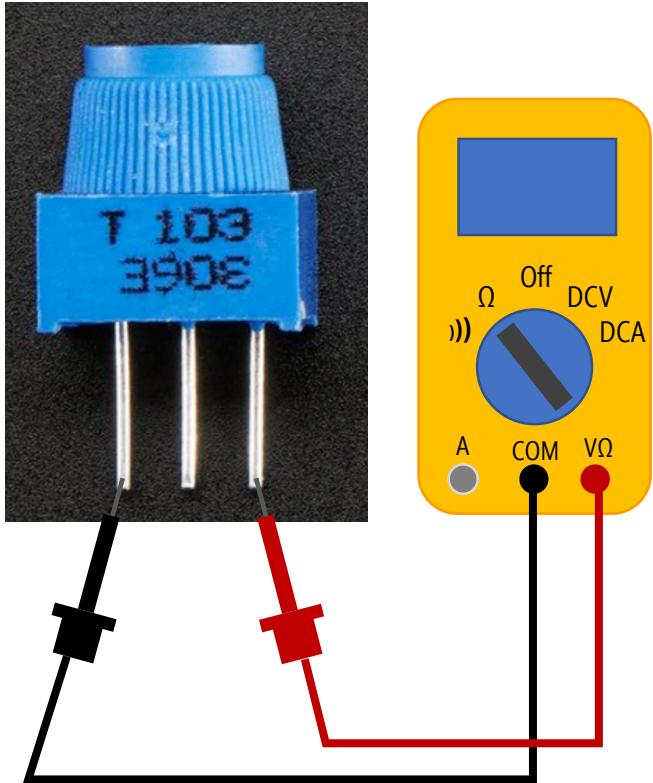
Hook up the **two outer legs** to the multimeter.
What happens when you turn the knob?



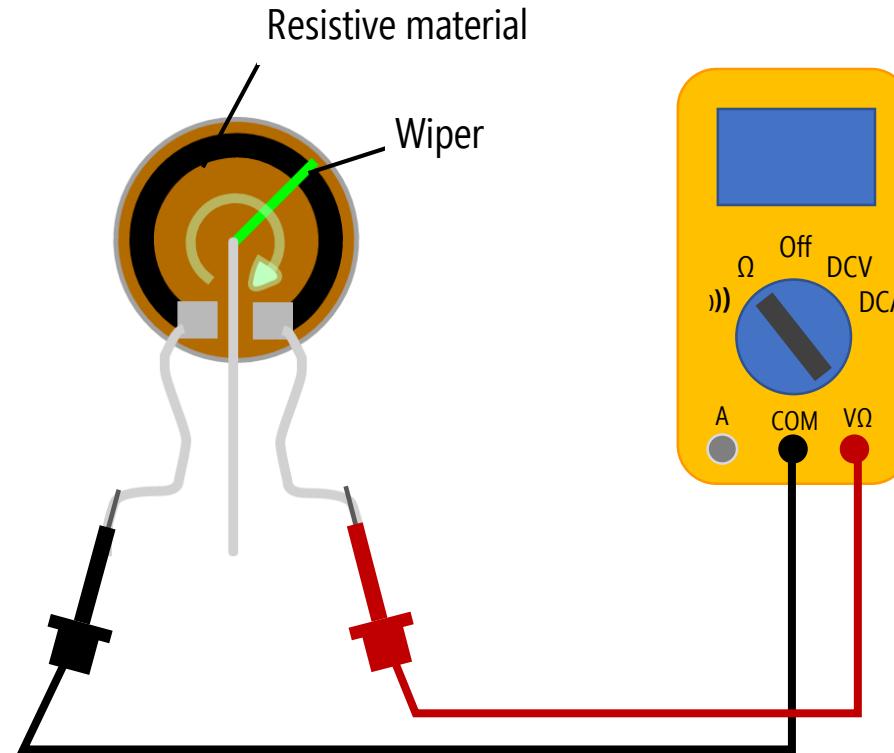
Hook up the **middle** and an **outer leg** to the
multimeter. What happens when you turn the knob?

POTENTIOMETERS

ACTIVITY: EXPLORE THE POT WITH A MULTIMETER



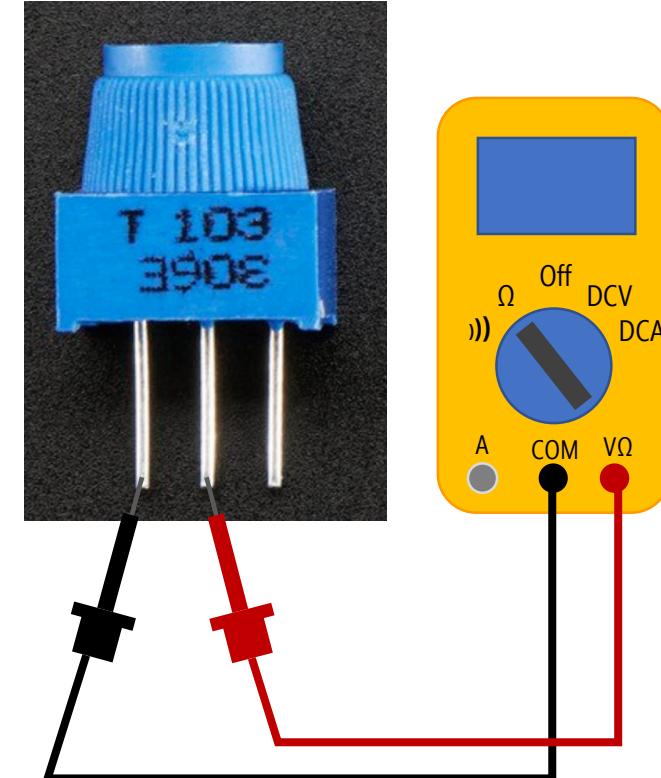
Hook up the **two outer legs** to the multimeter.
What happens when you turn the knob?



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

POTENTIOMETERS

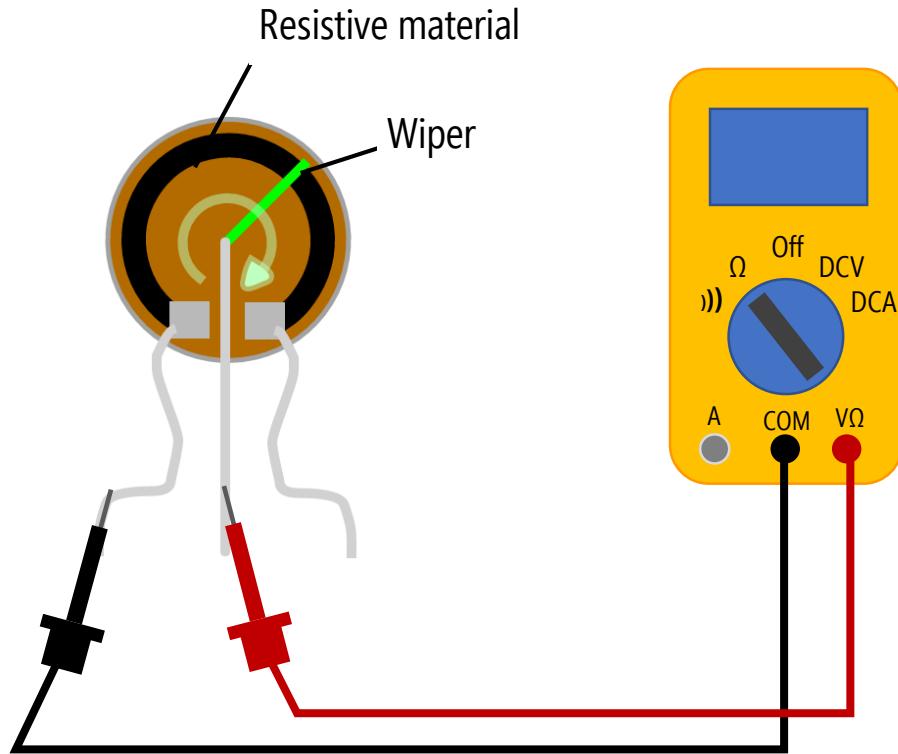
ACTIVITY: EXPLORE THE POT WITH A MULTIMETER



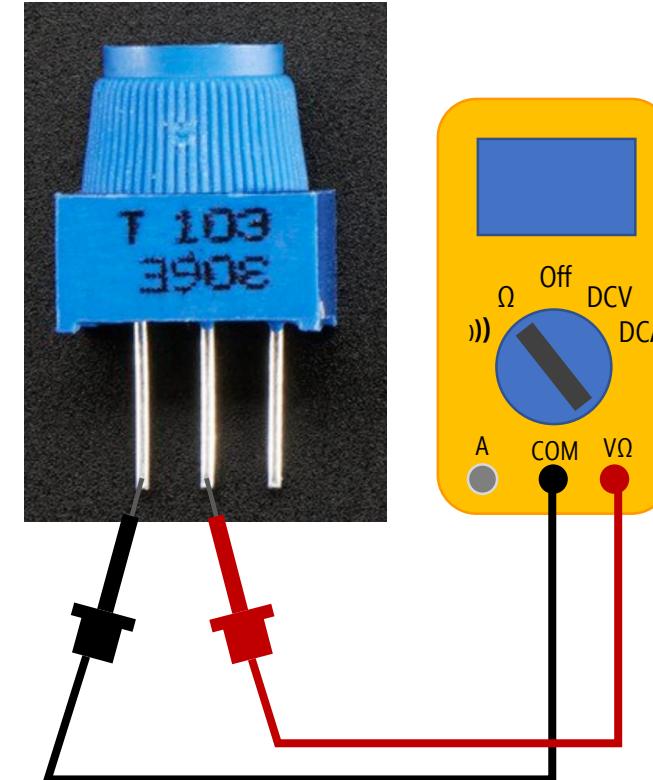
Hook up the **middle** and an **outer leg** to the multimeter. What happens when you turn the knob?

POTENTIOMETERS

ACTIVITY: EXPLORE THE POT WITH A MULTIMETER



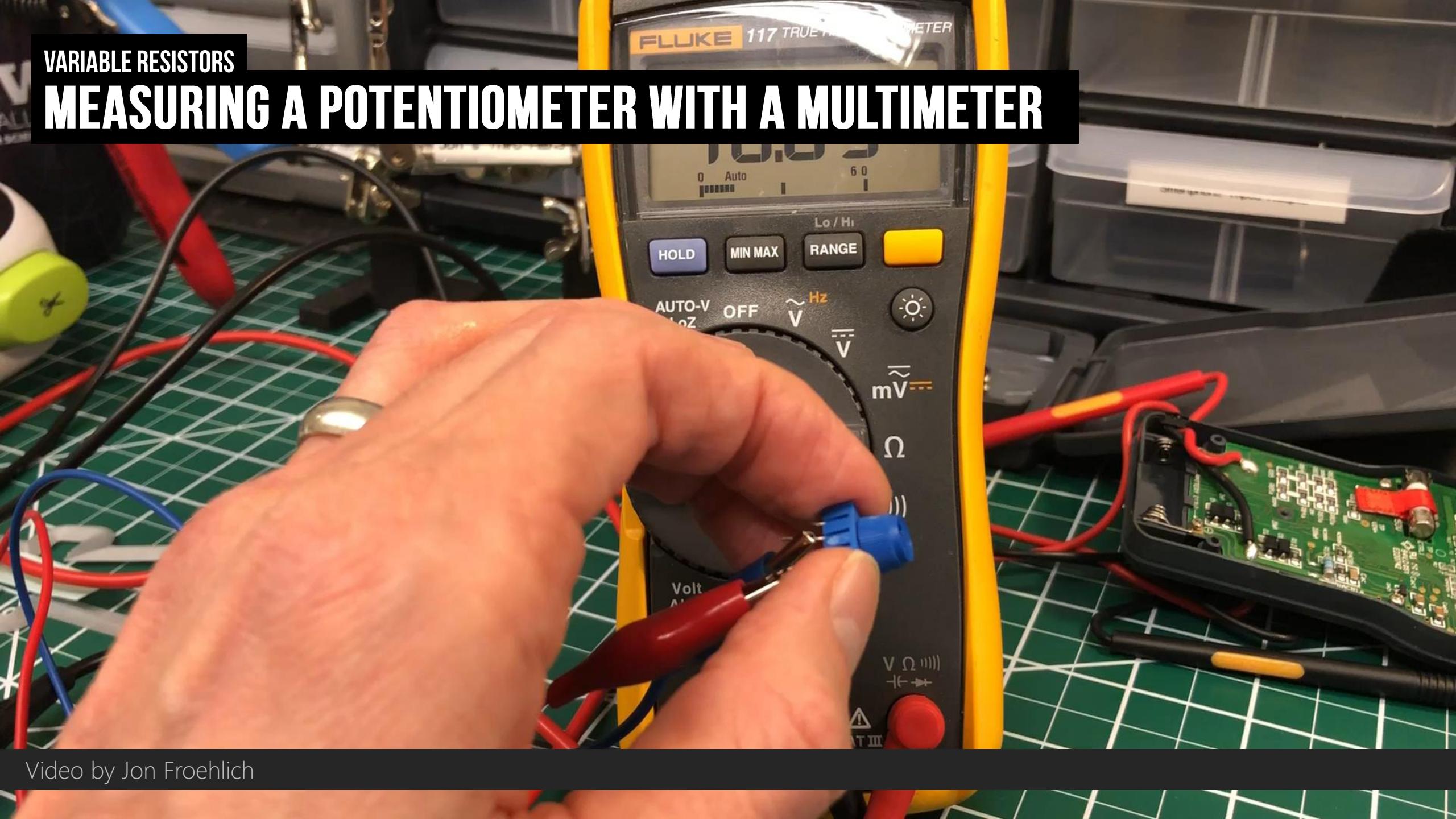
Answer: You'll see the resistance change based on the position of the knob. In this case, 0Ω with the knob all the way to the left, $5K\Omega$ in middle, and $10K\Omega$ with knob all the way to the right



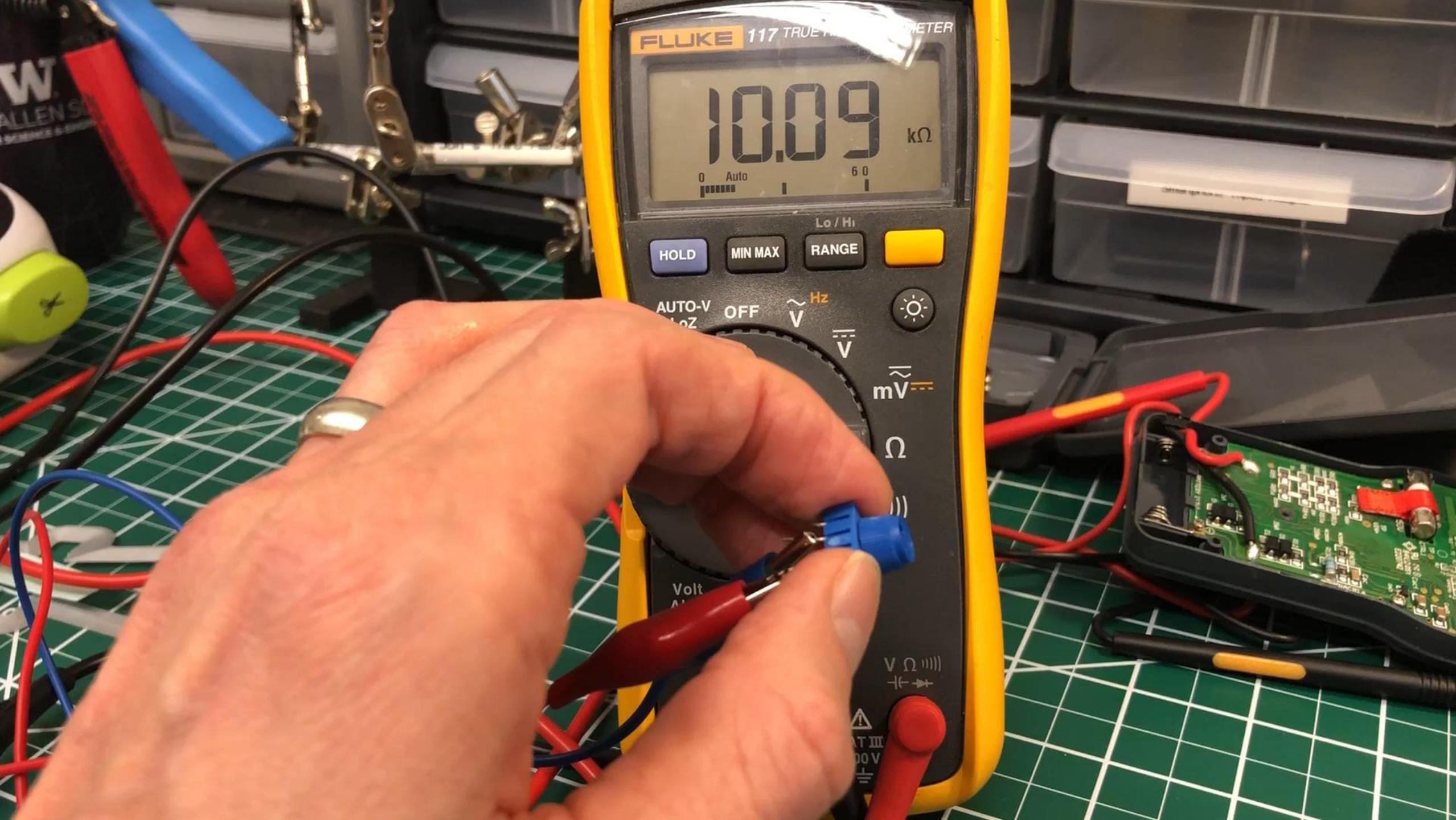
Hook up the **middle** and an **outer leg** to the multimeter. What happens when you turn the knob?

VARIABLE RESISTORS

MEASURING A POTENTIOMETER WITH A MULTIMETER



Video by Jon Froehlich

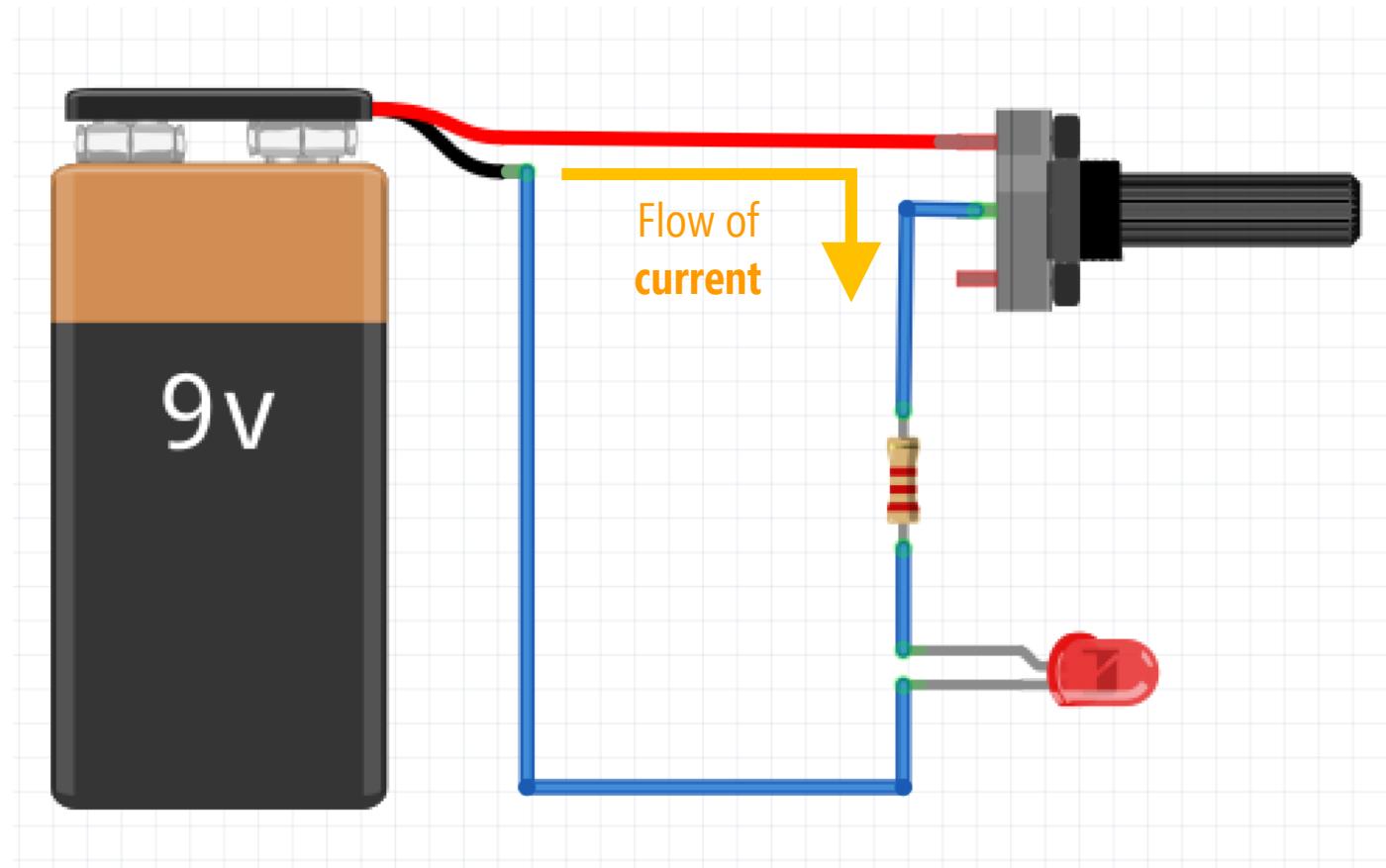


POTENTIOMETERS

ACTIVITY HOOK UP AN LED WITH A POT & ALLIGATOR CLIPS

POTENTIOMETERS

ACTIVITY HOOK UP AN LED WITH A POT & ALLIGATOR CLIPS



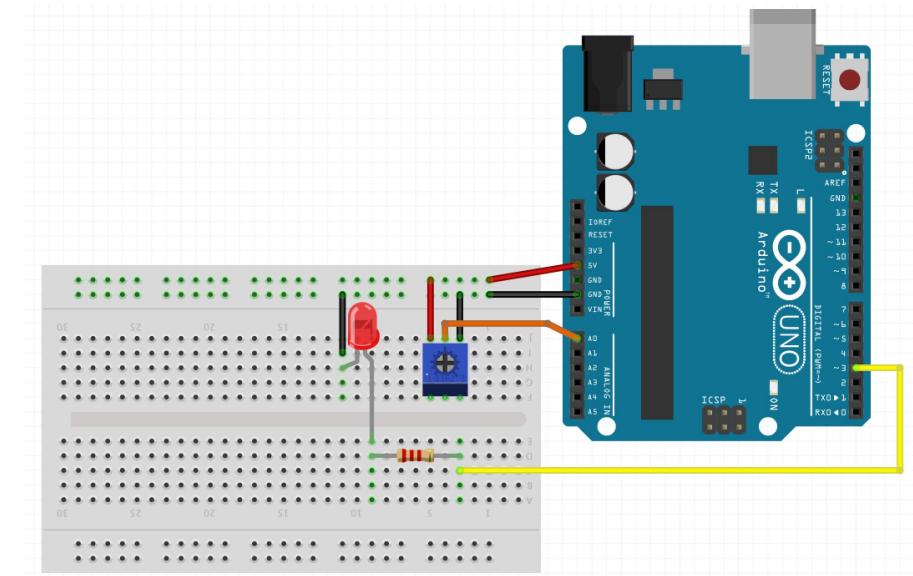
ANALOG INPUT

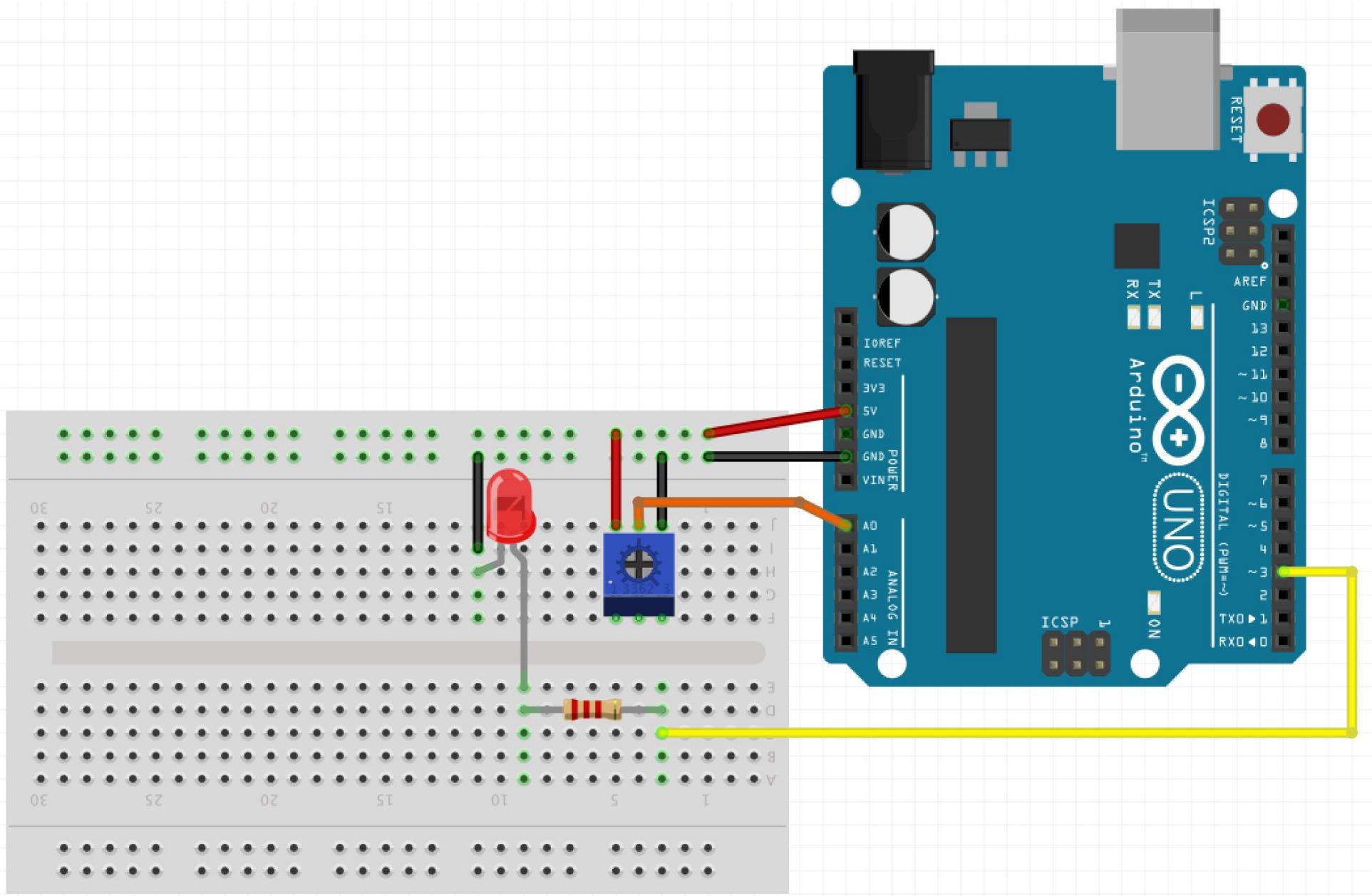
NOW LET'S DO THE SAME THING BUT WITH ARDUINO

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

One for input

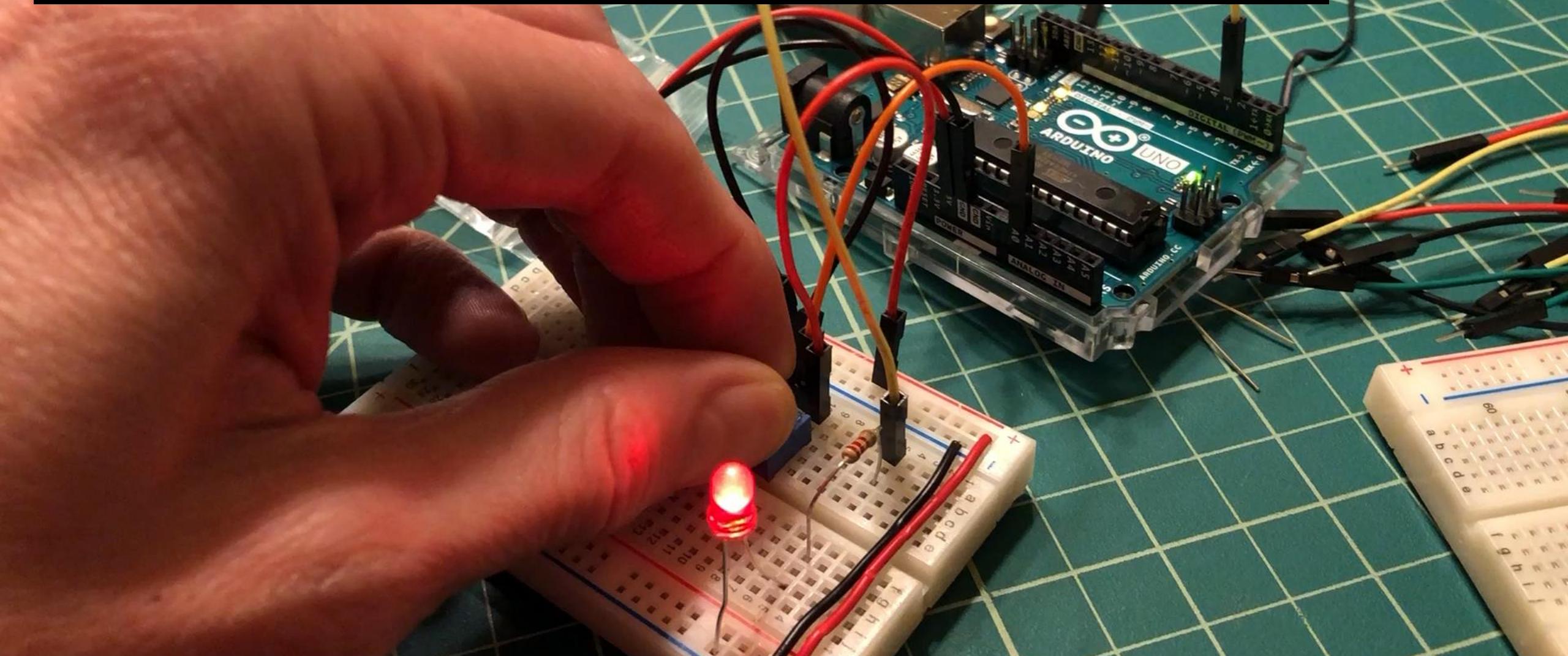
One for output





ANALOG INPUT

ACTIVITY: CONTROL LED BRIGHTNESS WITH A TRIM POT

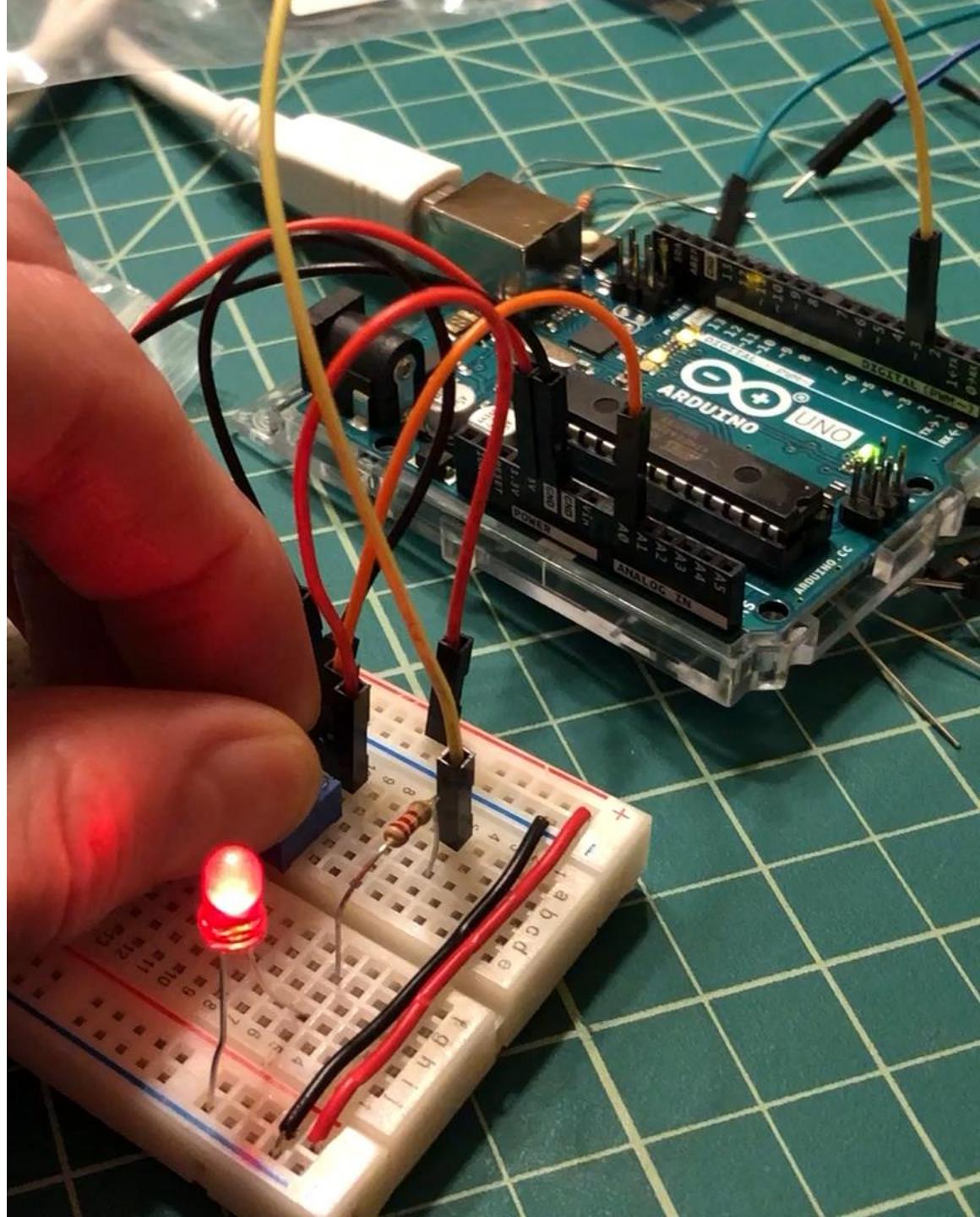


Video by Jon Froehlich

ANALOG INPUT

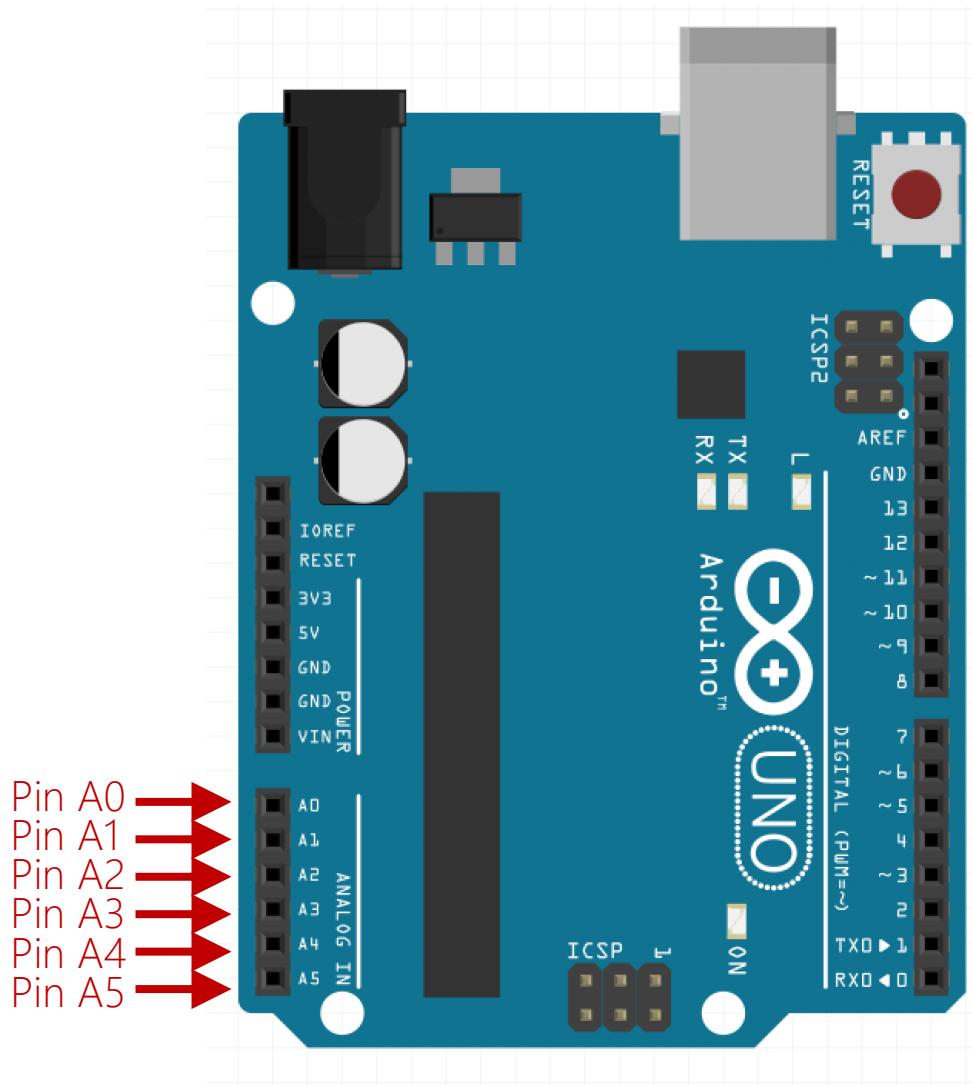
ACTIVITY: CONTROL LED BRIGHTNESS WITH TRIMPOT

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



ANALOG INPUT

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);
}
```

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.

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 (in

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/>

```
FadeOneDirection §
* Adapted from http://www.arduino.cc/en/Tutorial/Fade
*
*/
const int LED_OUTPUT_PIN = 3;
int _curBrightness = 0; // how bright the LED is
// The setup function runs once when you press reset or power the board
void setup() {
  // set the LED pin to as an output
  pinMode(LED_OUTPUT_PIN, OUTPUT);
}
// The loop function runs over and over again forever
void loop() {
  // set the brightness of the LED pin:
  analogWrite(LED_OUTPUT_PIN, _curBrightness);
  // change the brightness for next time through the loop:
  _curBrightness = _curBrightness + 1;
  // the maximum value that we can write out to analogWrite is 255
  // so check to see if the current brightness value is greater than 255
  // and if it is, reset the brightness to 0 (which is off)
  if (_curBrightness > 255){
    _curBrightness = 0;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}
```

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);

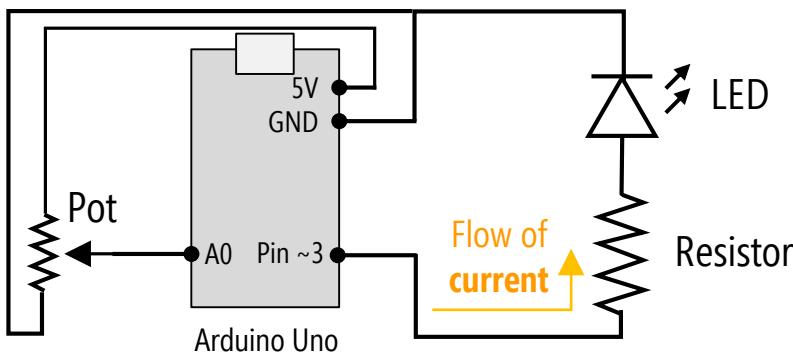
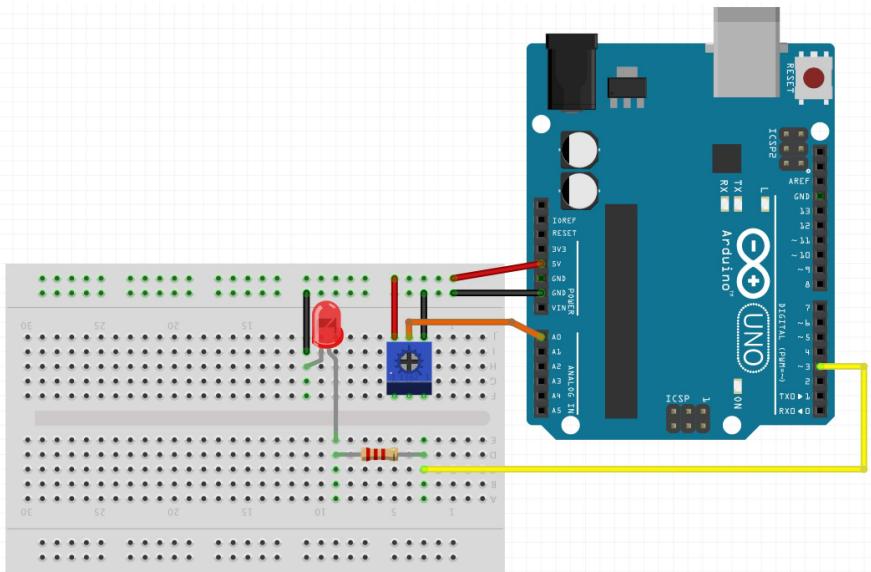
    // write out the LED value. This value is translated to voltage by:
    // voltageVal = max_voltage * val/255 or voltageVal = 3.3V * val/255 in
    // the case of the RedBear Duo
    analogWrite(LED_OUTPUT_PIN, ledVal);

    delay(100);
}
```

ANALOG INPUT

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! ☺

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

ANALOG INPUT

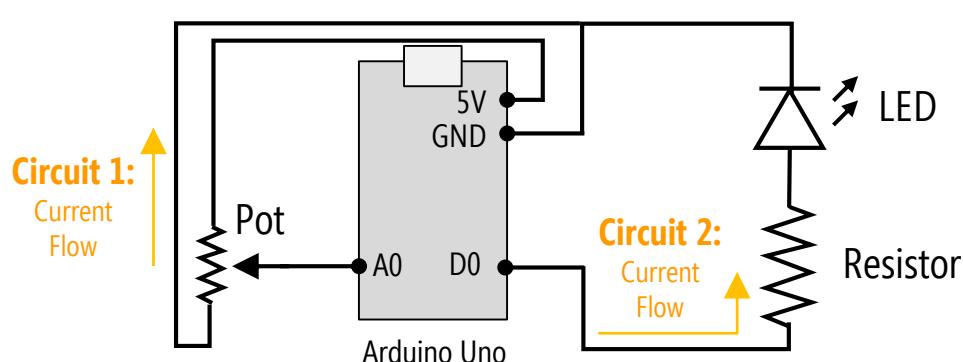
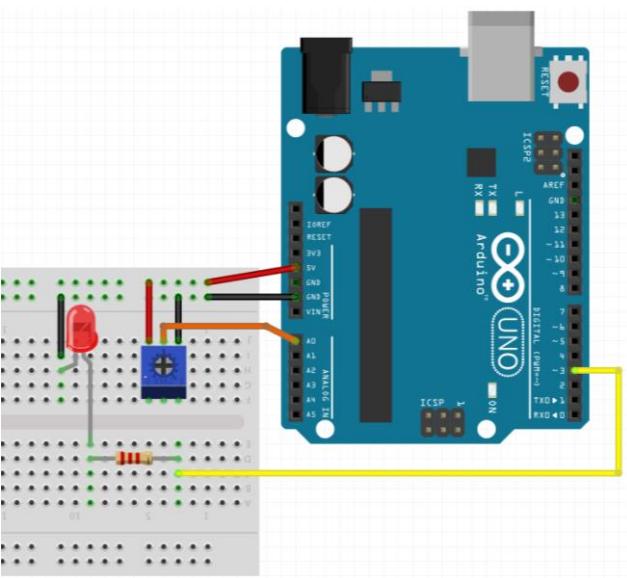
HOW DOES THIS WORK?



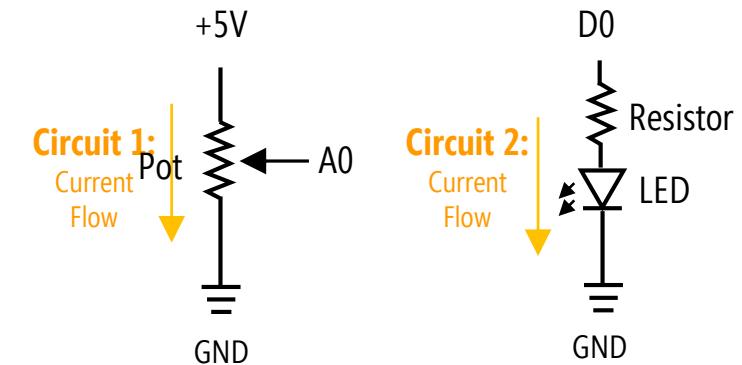
ANALOG INPUT

HOW DOES THIS WORK?

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



Schematic Representation 1

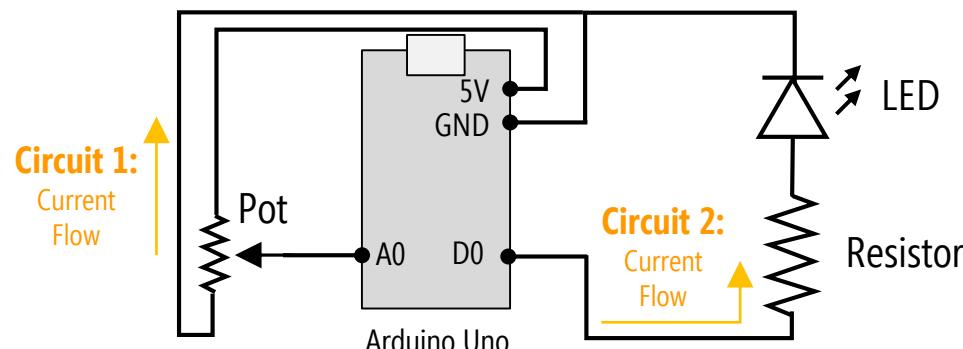
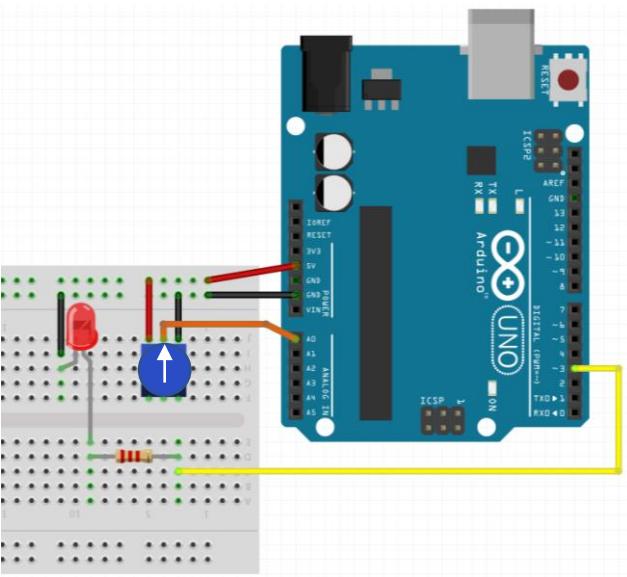


Schematic Representation 2

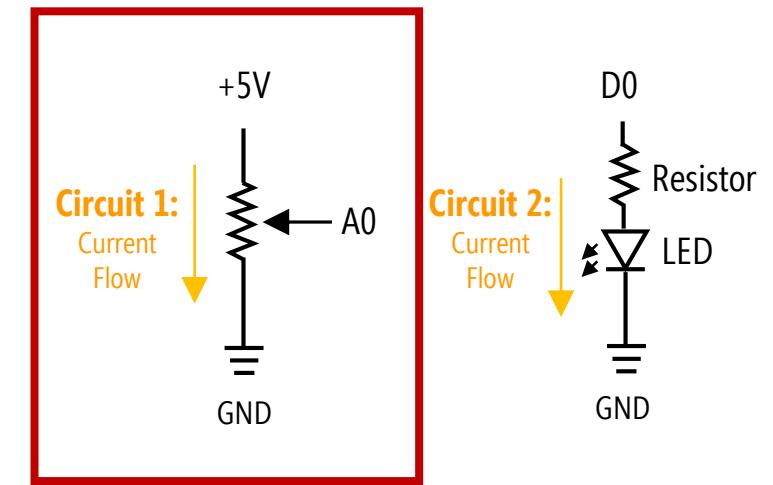
ANALOG INPUT

HOW DOES THIS WORK?

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



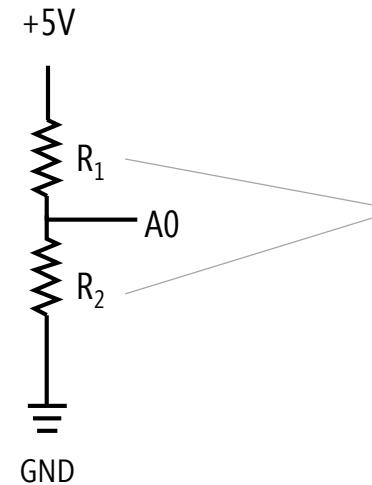
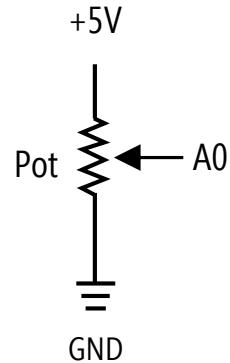
Schematic Representation 1



Schematic Representation 2

HOW DOES THIS WORK?

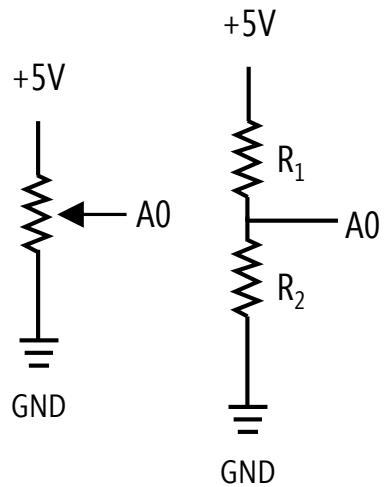
Another way to think about the potentiometer



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

ANALOG INPUT

WHAT IF THE POT IS SET TO $5\text{K}\Omega$?



WHAT IF THE POT IS SET TO 5KΩ?

What is V_{A0} equal to?

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

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

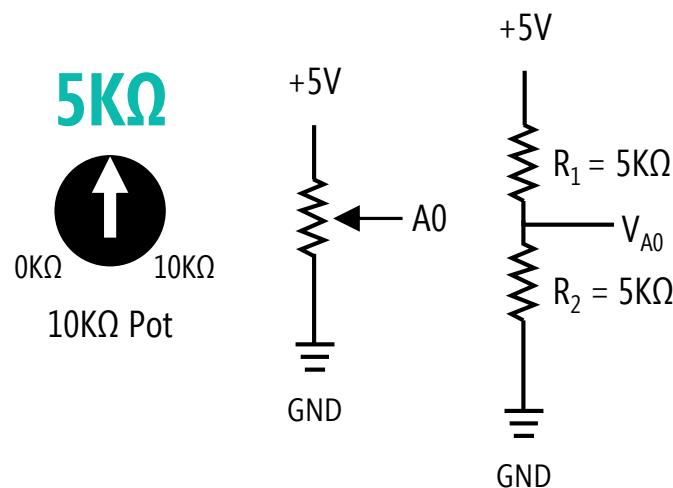
$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

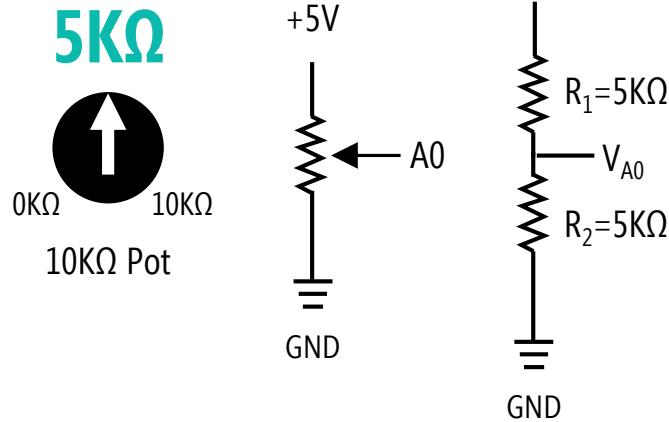
$$\text{Voltage}_{A0} = V_{A0} = 5V - (0.0005A * 5,000\Omega)$$

$$\text{Voltage}_{A0} = V_{A0} = 2.5V$$



ANALOG INPUT

WHAT IF THE POT IS SET TO 5KΩ?



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

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

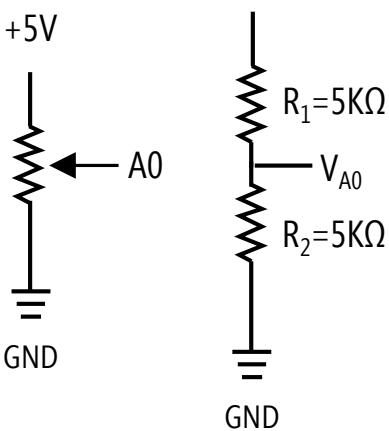
$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 5,000\Omega)$$

$$Voltage_{A0} = V_{A0} = 2.5V \quad \text{analogRead(A0) would return 512}$$

WHAT IF THE POT IS SET TO 5KΩ?

5KΩ

 0KΩ 10KΩ
 10KΩ Pot



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

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

$$\text{Voltage}_{A0} = V_{A0} = 2.5V \quad \text{analogRead(A0) would return 512}$$

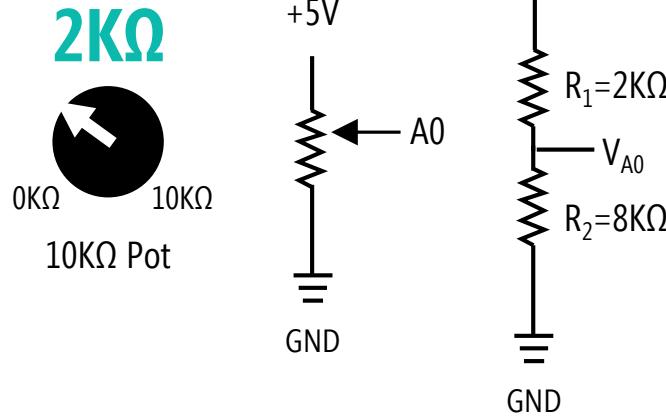
Why? Recall that the minimum and maximum analogRead values are 0 and 1023 respectively on the Uno. Thus:

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

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

$$x = 0.5 * 1023 = 511.5$$

WHAT IF THE POT IS SET TO 2KΩ?



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

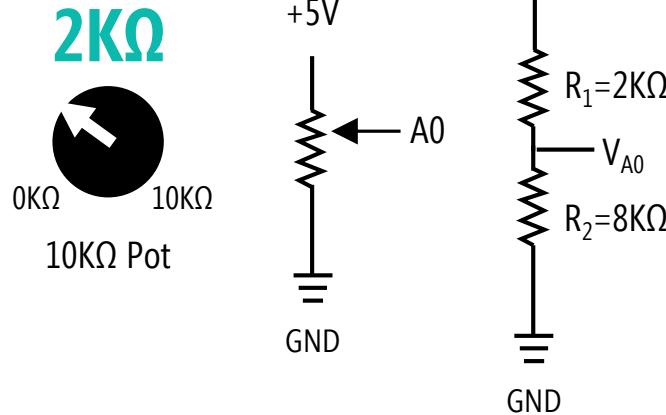
$$\text{Voltage} = V = I * R$$

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

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

$$\text{Voltage}_{A0} = V_{A0} = 4V$$

WHAT IF THE POT IS SET TO 2KΩ?



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

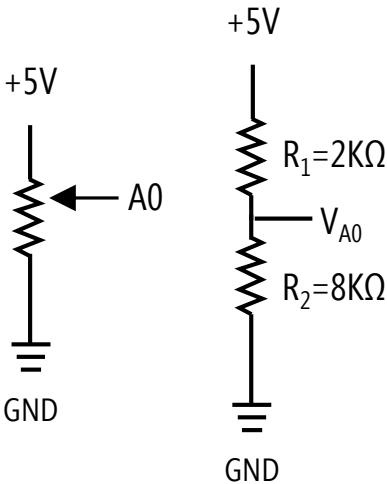
$$\text{Voltage}_{A0} = V_{A0} = 5V - (0.0005A * 2,000\Omega)$$

$$\text{Voltage}_{A0} = V_{A0} = 4V \quad \text{analogRead(A0) would return 818}$$

WHAT IF THE POT IS SET TO 2KΩ?

2KΩ

 0KΩ 10KΩ
 10KΩ Pot



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

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

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

$$x = 0.8 * 1023 = 818.4$$

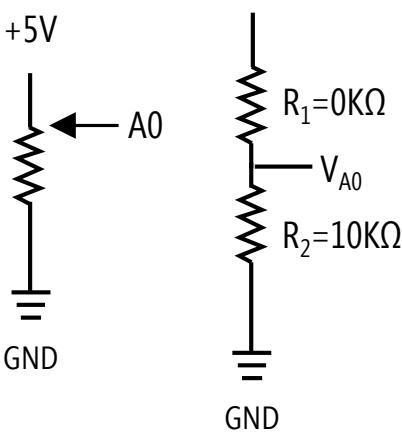
$$\text{Voltage}_{A0} = V_{A0} = 5V - (0.0005A * 2,000\Omega)$$

$$\text{Voltage}_{A0} = V_{A0} = 4V \quad \text{analogRead(A0) would return } \mathbf{818}$$

WHAT IF THE POT IS SET TO $0K\Omega$

$0K\Omega$

 10KΩ Pot



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

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

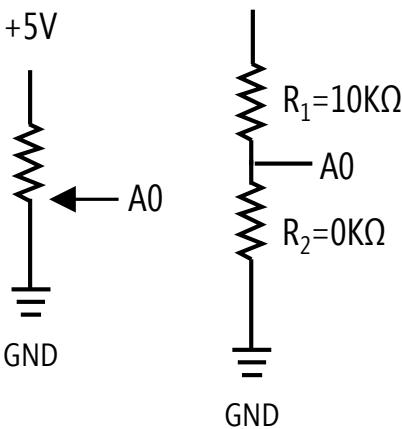
$$Voltage_{A0} = V_{A0} = 5V - (0.0005A * 0\Omega)$$

$$Voltage_{A0} = V_{A0} = 5V \quad \text{analogRead(A0) would return 1023}$$

WHAT IF THE POT IS SET TO 10KΩ

10KΩ

 0KΩ 10KΩ
 10KΩ Pot



What is A0 equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

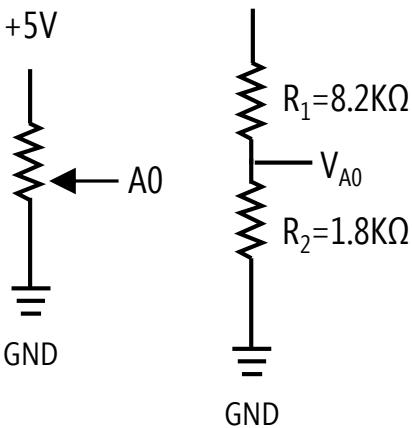
$$\text{Voltage}_{A0} = V_{A0} = 5V - (0.0005A * 10,000\Omega)$$

$$\text{Voltage}_{A0} = V_{A0} = 0V \quad \text{analogRead(A0) would return 0}$$

WHAT IF THE POT IS SET TO 8.2KΩ

8.2KΩ

 0KΩ 10KΩ
 10KΩ Pot



What is V_{A0} equal to?

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

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

$$\text{Current} = I = 0.0005A = 0.5mA$$

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

$$\text{Voltage}_{A0} = V_{A0} = 5V - V_{R1}$$

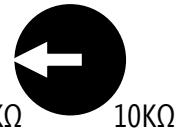
$$\text{Voltage}_{A0} = V_{A0} = 5V - (0.0005A * 8,200\Omega)$$

$$\text{Voltage}_{A0} = V_{A0} = 0.9V \quad \text{analogRead(A0) would return 184}$$

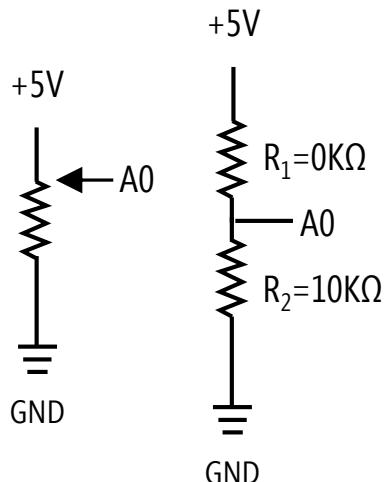
ANALOG INPUT

LET'S LOOK AT SOME VALUES ALL TOGETHER

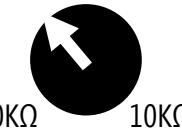
0KΩ



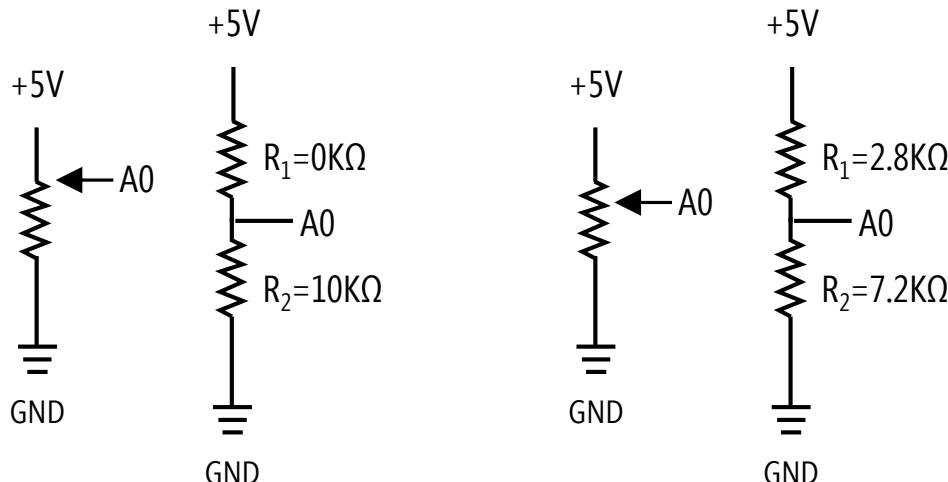
10KΩ Pot



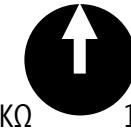
2.8KΩ



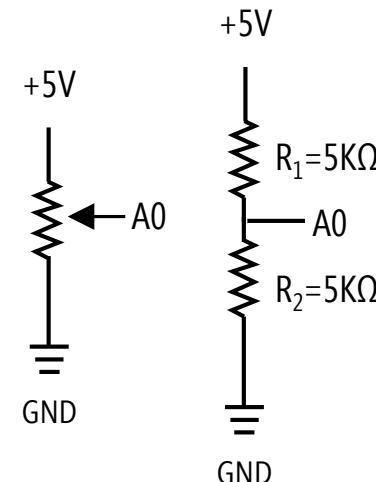
10KΩ Pot



5KΩ



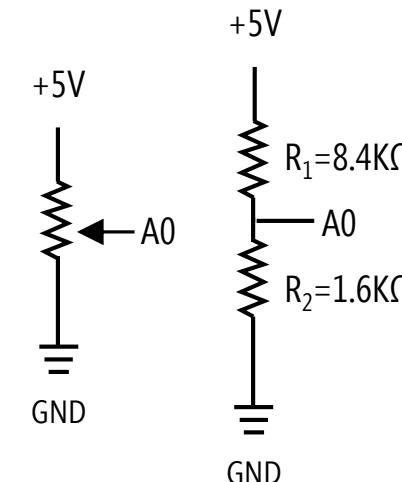
10KΩ Pot



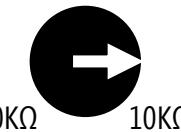
8.4KΩ



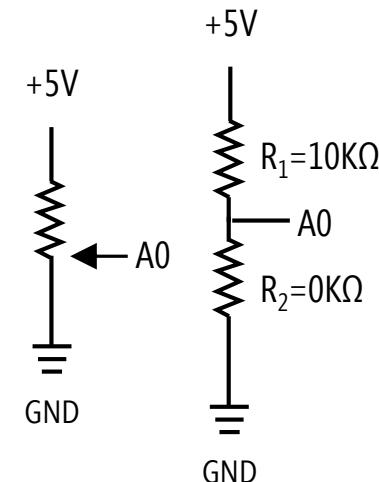
10KΩ Pot



10KΩ



10KΩ Pot



$Voltage_{A0} = V_{A0} = 5V$
analogRead(A0) returns 1023

$Voltage_{A0} = V_{A0} = 3.6V$
analogRead(A0) returns 737

$Voltage_{A0} = V_{A0} = 2.5V$
analogRead(A0) returns 512

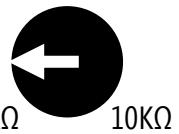
$Voltage_{A0} = V_{A0} = 0.8V$
analogRead(A0) returns 164

$Voltage_{A0} = V_{A0} = 0V$
analogRead(A0) returns 0

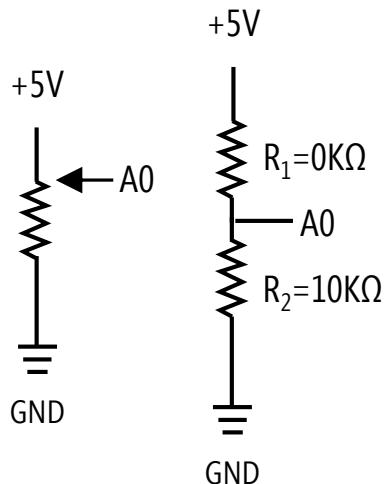
ANALOG INPUT

LET'S LOOK AT SOME VALUES ALL TOGETHER

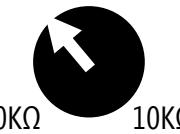
0KΩ



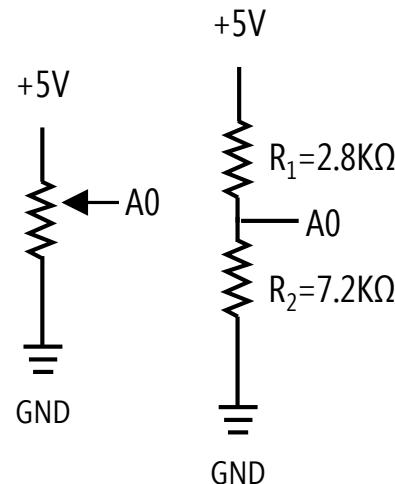
0KΩ 10KΩ
10KΩ Pot



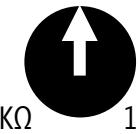
2.8KΩ



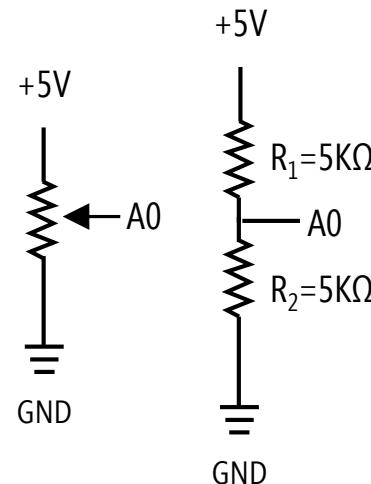
0KΩ 10KΩ
10KΩ Pot



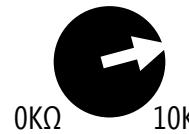
5KΩ



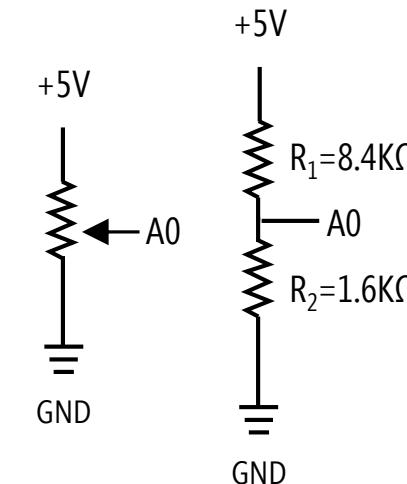
0KΩ 10KΩ
10KΩ Pot



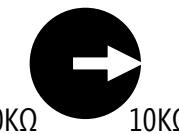
8.4KΩ



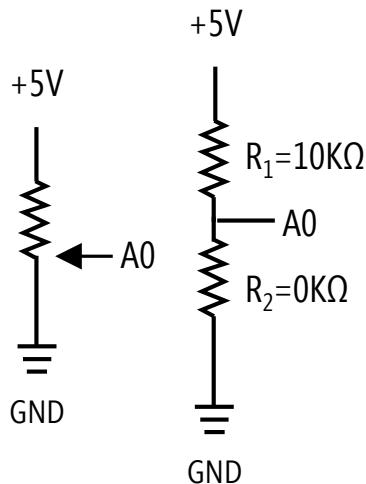
0KΩ 10KΩ
10KΩ Pot



10KΩ



0KΩ 10KΩ
10KΩ Pot



$Voltage_{A0} = V_{A0} = 5V$
analogRead(A0) returns 1023

$Voltage_{A0} = V_{A0} = 3.6V$
analogRead(A0) returns 737

$Voltage_{A0} = V_{A0} = 2.5V$
analogRead(A0) returns 512

$Voltage_{A0} = V_{A0} = 0.8V$
analogRead(A0) returns 164

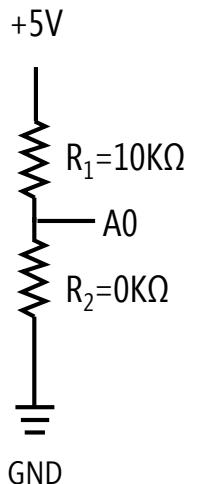
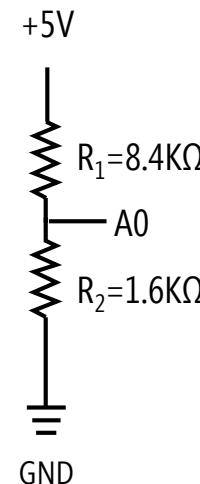
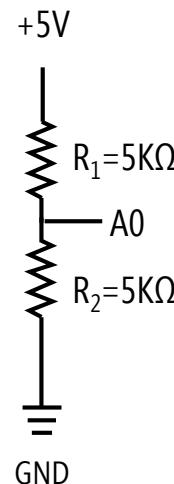
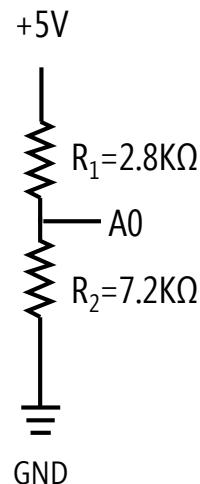
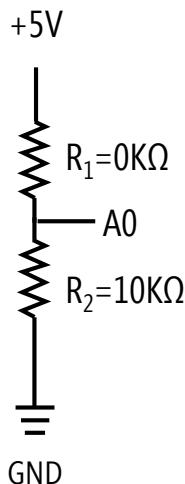
$Voltage_{A0} = V_{A0} = 0V$
analogRead(A0) returns 0

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.



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

analogRead(A0) returns 1023

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

analogRead(A0) returns 737

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

analogRead(A0) returns 512

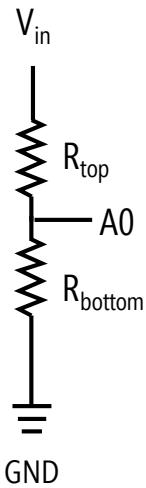
$$Voltage_{A0} = V_{A0} = 0.8V$$

analogRead(A0) returns 164

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

analogRead(A0) returns 0

VOLTAGE DIVIDER EQUATION



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

}

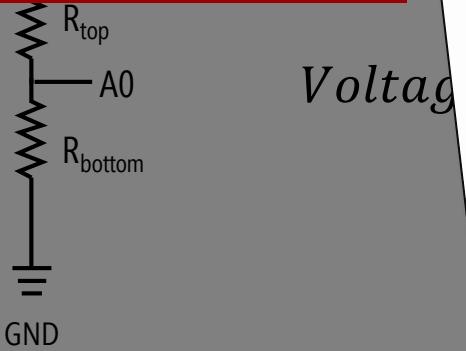
Note that it's the **ratio** between the two resistors that's important.

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

VOLTAGE DIVIDER EQUATION

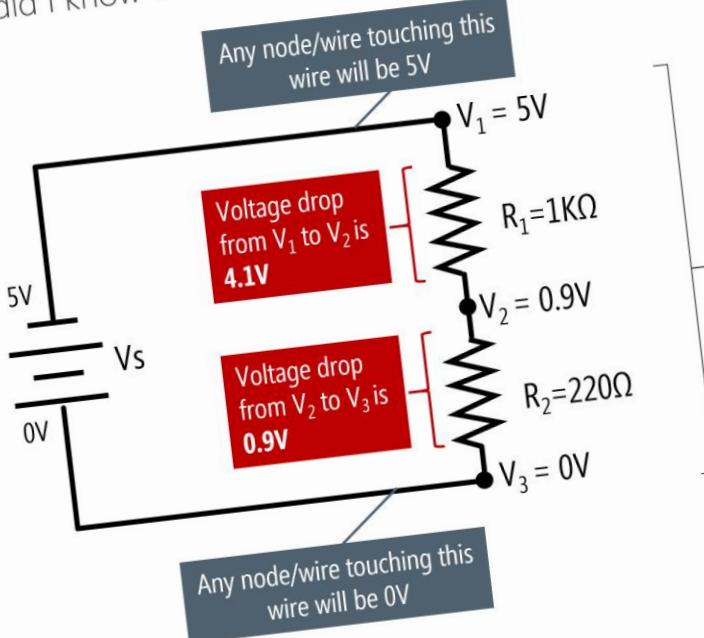
We covered this in our
1st Physical Computing
Lecture



SERIES VS. PARALLEL RESISTANCE

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

How did I know V1 and V3?



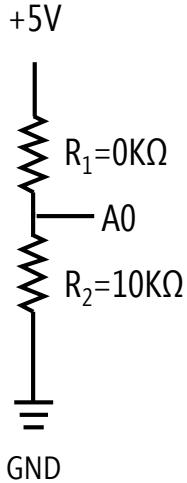
This basic circuit is called a **voltage divider** because it "splits" or "divides" voltages across nodes.

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

As R_{bottom} increases, V_{A0} increases with respect to R_{bottom} , V_{A0} grows proportionally small

ANALOG INPUT

APPLYING THE “VOLTAGE DIVIDER” EQUATION



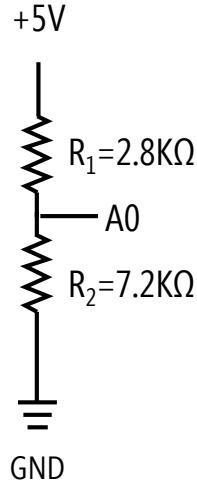
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

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

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

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

`analogRead(A0)` returns **1023**



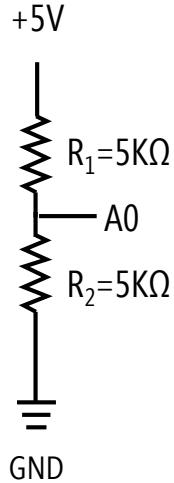
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

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

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

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

`analogRead(A0)` returns **737**



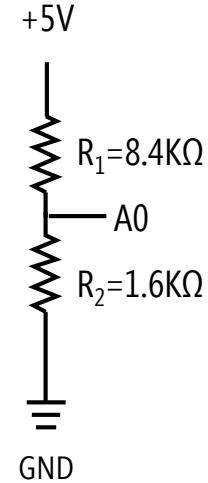
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

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

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

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

`analogRead(A0)` returns **512**



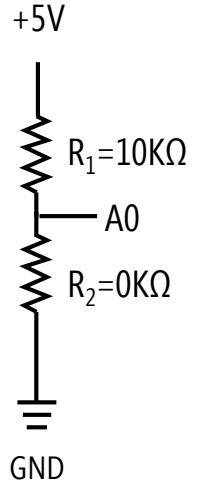
$$V_{A0} = V_{in} * \frac{R_{bottom}}{(R_{top} + R_{bottom})}$$

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

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

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

`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

A close-up, low-angle shot of a person's hands adjusting the faders on a large audio mixing console. The console is covered in numerous circular knobs, red faders, and small digital displays. The brand name "LENNARZETH ZED 436" is visible on the top right of the main control panel. The lighting is dramatic, with strong highlights on the hands and the metallic surfaces of the console, while the background is dark.

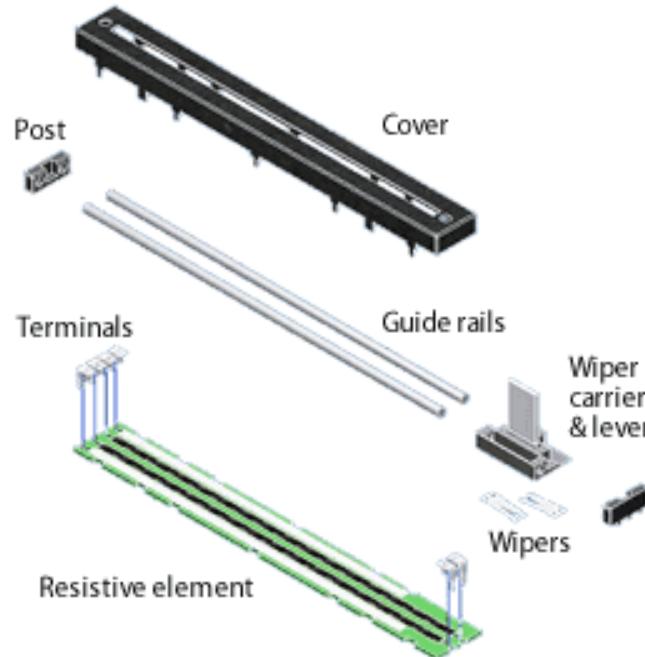
SLIDE POTENTIOMETERS



Slide potentiometers on an audio
mixing board in a music studio

VARIABLE RESISTORS

SLIDE POTENTIOMETERS



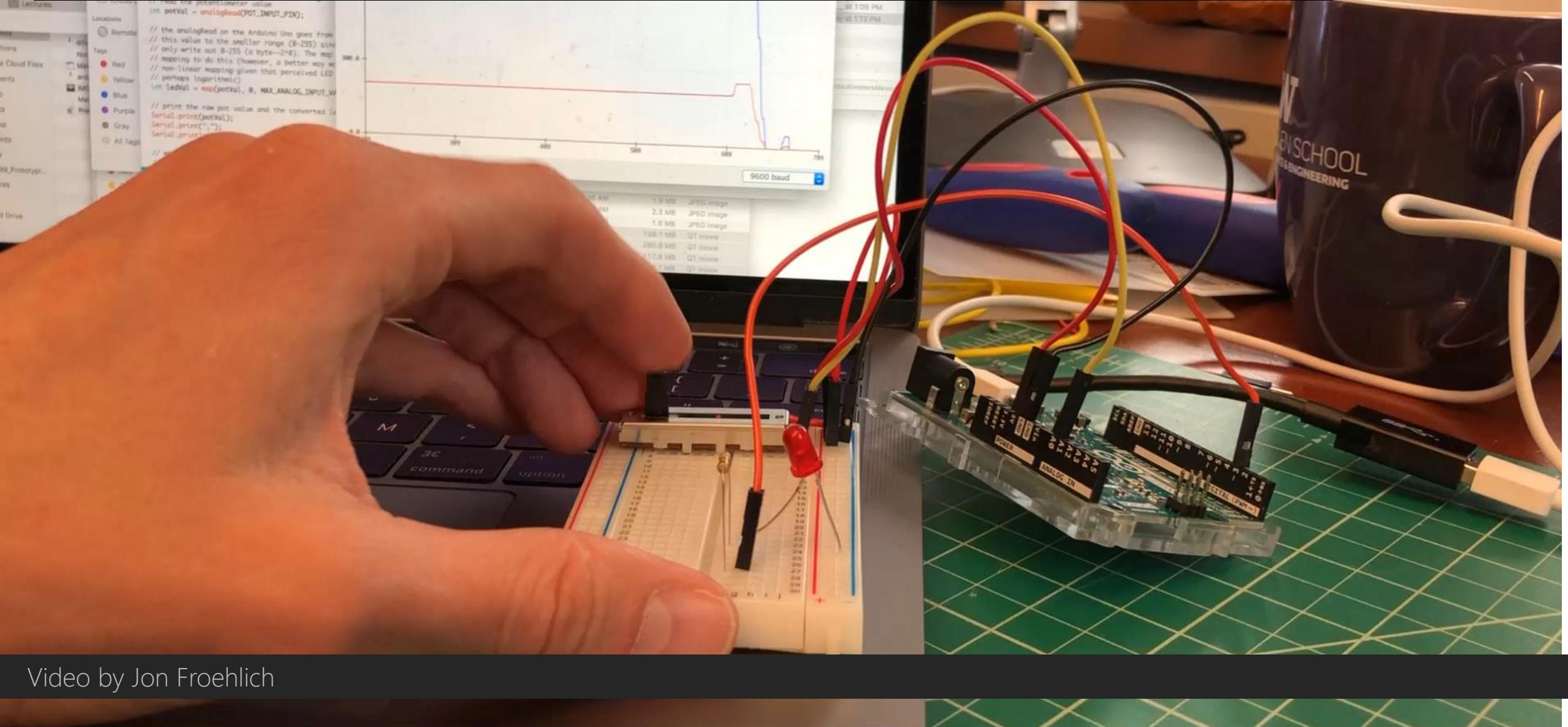
Very similar to a **traditional (rotating) potentiometer** but the resistive material is laid out in a straight line

Can have linear taper, exponential taper, etc.

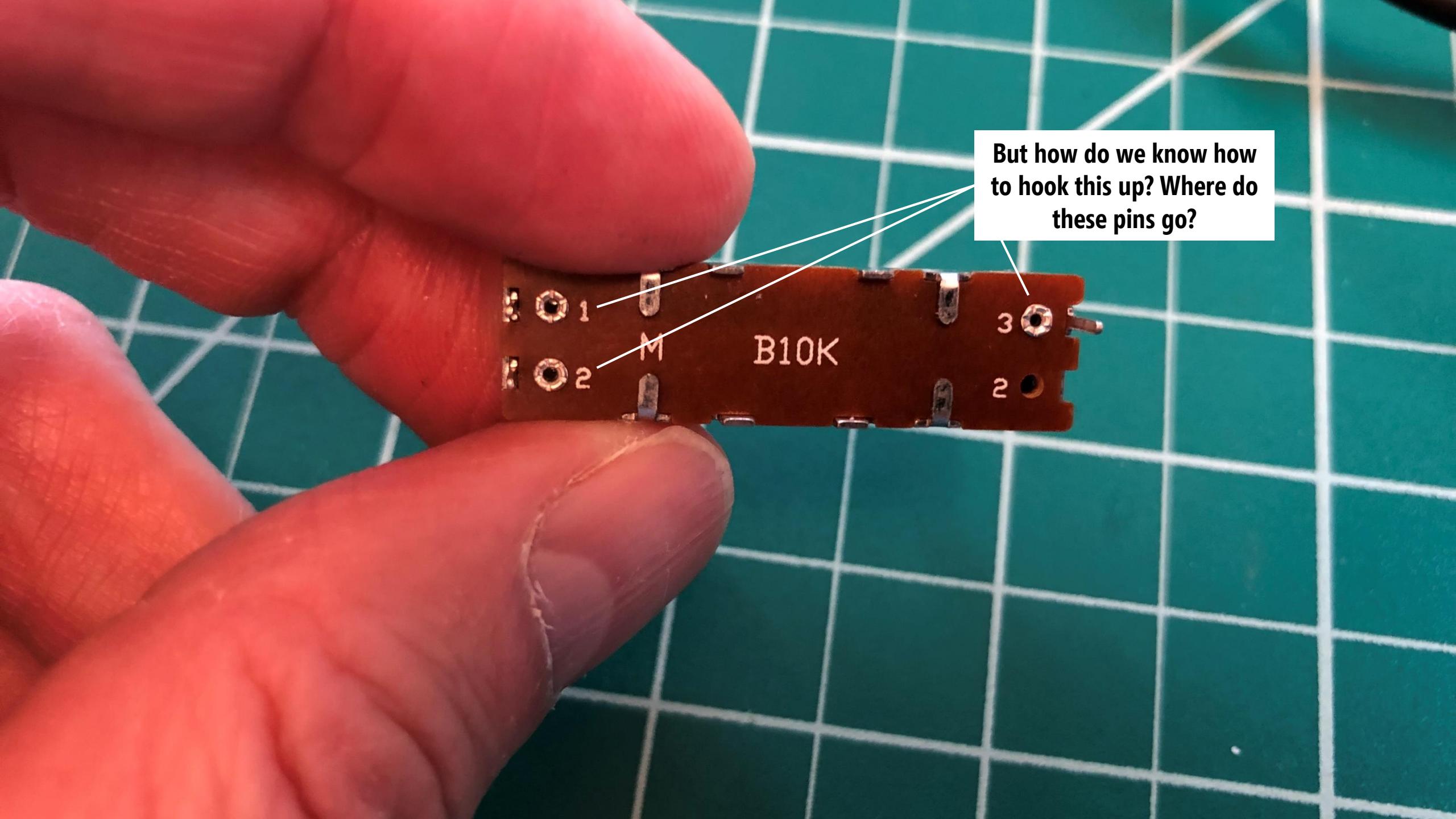
Comes in different sizes

ANALOG INPUT

ACTIVITY: CONTROL LED BRIGHTNESS WITH A SLIDE POT



Video by Jon Froehlich



But how do we know how
to hook this up? Where do
these pins go?



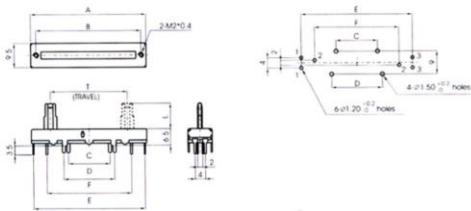


How do we know? To the **data sheet!**

LEVER TYPE

MATERIAL	Insulated lever							
LEVER TYPE	C		CA		B			
DIMENSIONS								
LENGTH(L)	X: 1 2 3 5 6 7	L: 10 12.5 15 18 20 0	X: 4 8	L: 5 10	X: 9	L: 10		
F	5 5 5 5 5 5							

P.C.B. MOUNTING HOLE DETAIL



MODEL	T	A	B	C	D	E	F
C1531	15	30	26		28.5	18	
C2031	20	35	31	6.2	9.8	33.5	23
C3031	30	45	41	16.2	19.8	43.5	33
C4531	45	60	56		58.5	48	

Electrical characteristics:電氣的性能 :

Total resistance 總阻值	5K Ω ~2M Ω						
Total resistance tolerance 總阻偏差	± 20% more than 1M Ω ± 30%						
Resistance taper 阻值規律	A, B, C, D, K, W, RD						
Insulation resistance 絶緣電阻	100MΩ min. at 250V DC.						
Withstand voltage 耐電壓	AC 300V (1 minute)						
	Travel Taper	15 mm	20 mm	30 mm	45 mm		
Rated power 額定功率 (W)	B	Single	0.05	0.1	0.2	0.25	
	Dual	0.025	0.05	0.1	0.125		
Except B (B 以外)	Single	0.025	0.05	0.1	0.125		
	Dual	0.012	0.025	0.05	0.06		
Max. operating voltage(AC V) 最高使用電壓	B	Single	100	200	200	200	
	Except B (B 以外)	Dual	50	150	150	150	
Residual resistance 殘留阻值	Total resistance 總阻值	10K Ω	50K Ω	100K Ω	200K Ω	500K Ω	1M Ω
	Between terminals 1-2 端子 1-2 間	10 Ω	10 Ω	10 Ω	20 Ω	20 Ω	20 Ω
	Between terminals 2-3 端子 2-3 間	10 Ω	10 Ω	20 Ω	20 Ω	50 Ω	50 Ω

Durability:耐久的性能:

Slidinglife 滑動壽命	15,000 cycles
---------------------	---------------

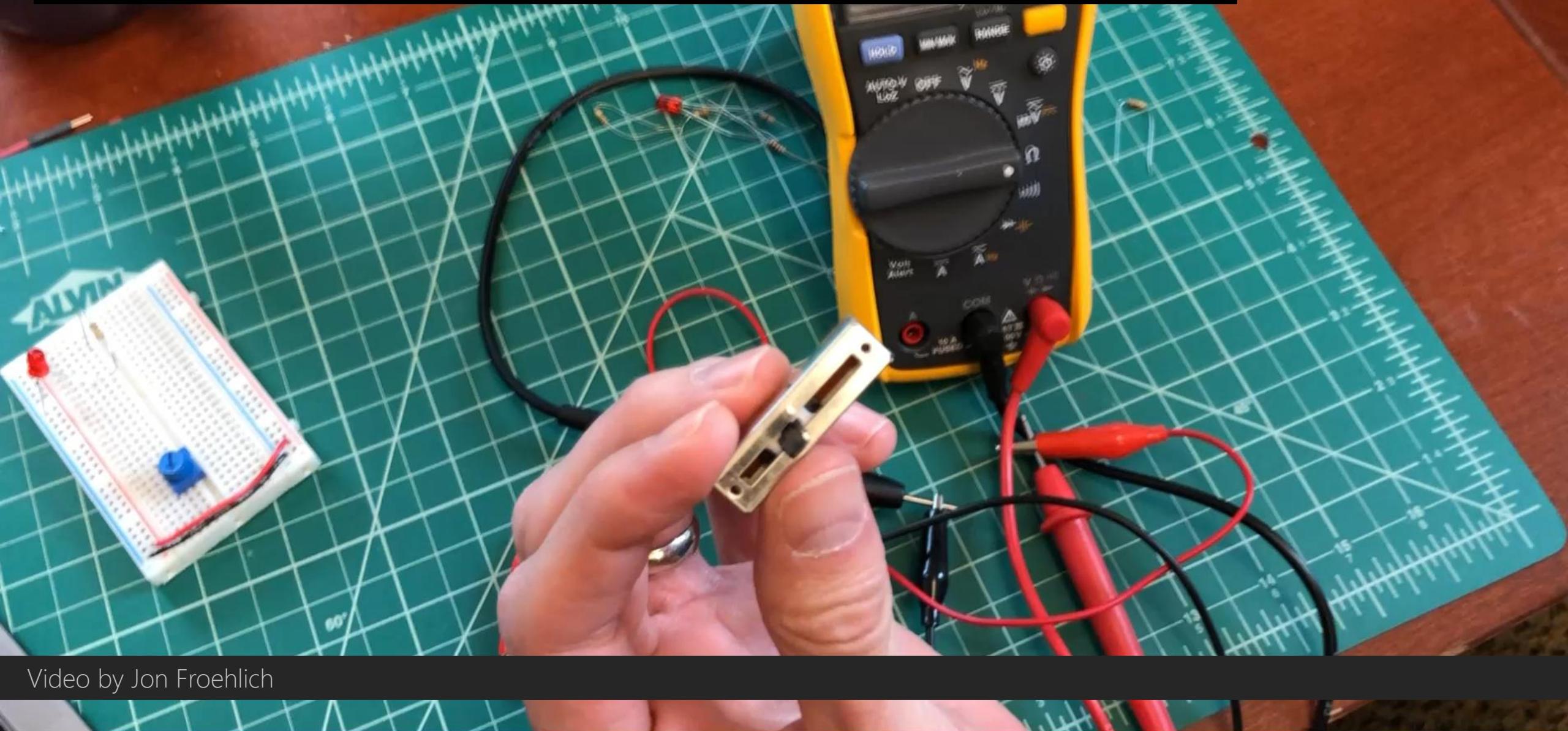
Mechanical characteristics:機械的性能 :

Operating force 滑動力	0.2 ~ 2N (20 ~ 200gf)
Stopper strength 止擋強度	50N (5kgf) measuring point = 5mm
Lever wobble 柄的晃動	2 (2*L) / 20 mm max. (L: Lever length both side) L: 兩側柄長
Bendingmoment 瞬間彎曲	25mN.m (250gf.cm)

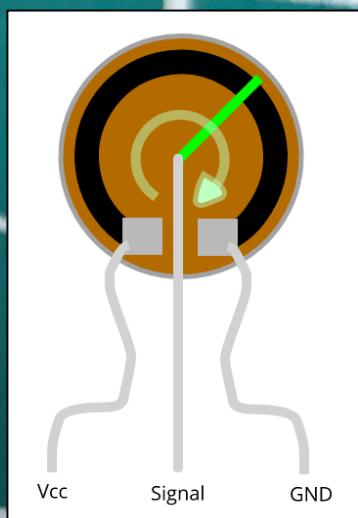
Push pull strength 軸推拉強度	30N (3kgf)
Click slip - out force C.C. 脫出力	0.2 ~ 2N (20 ~ 200gf)
Lever deviation 柄的偏心量	0.5 max. (one side) 単側

ANALOG INPUT

USE A MULTIMETER TO DETERMINE SLIDE POT PINS

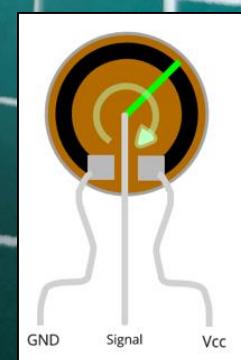
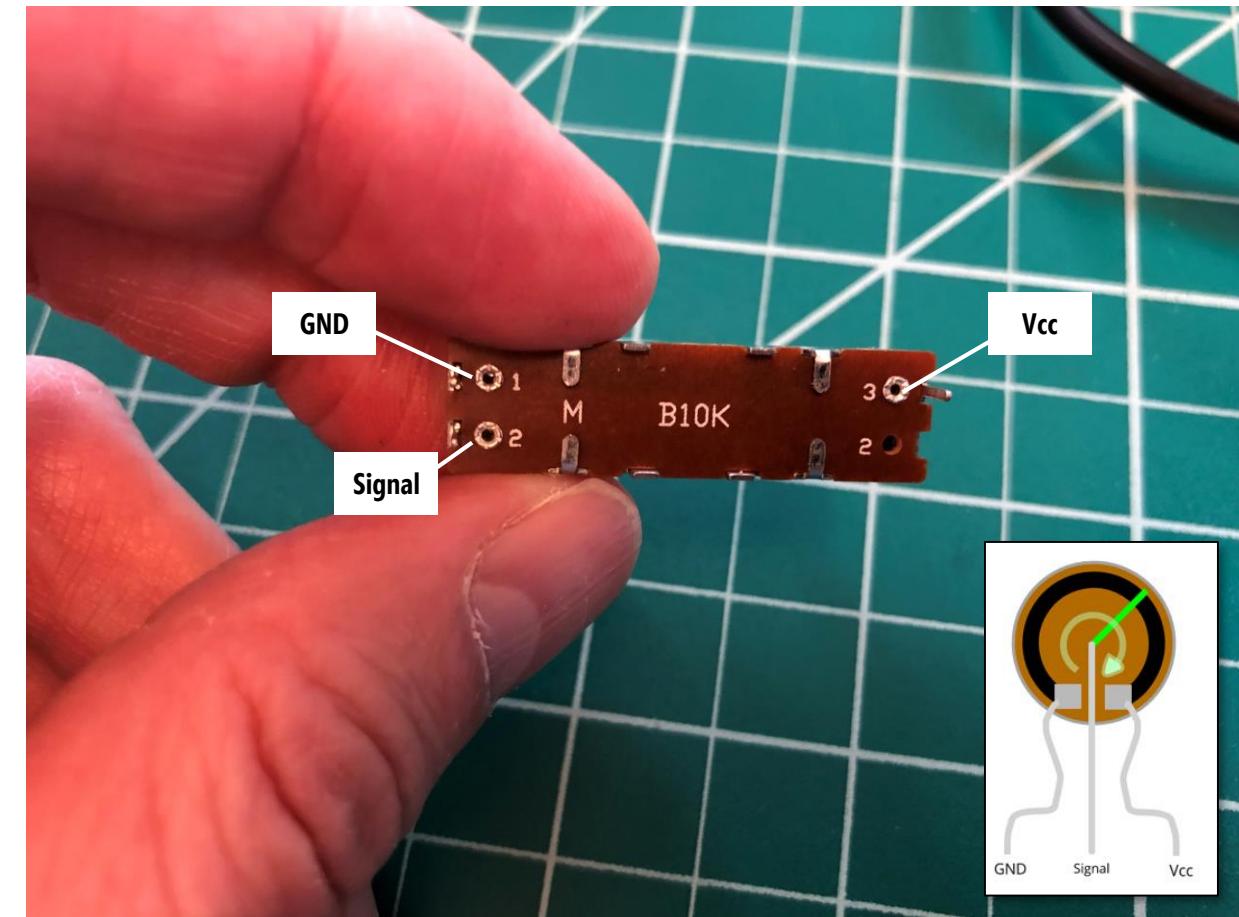
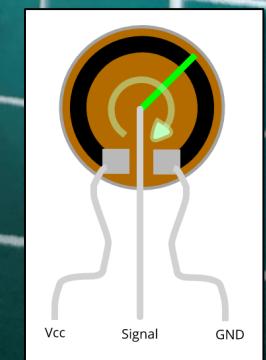
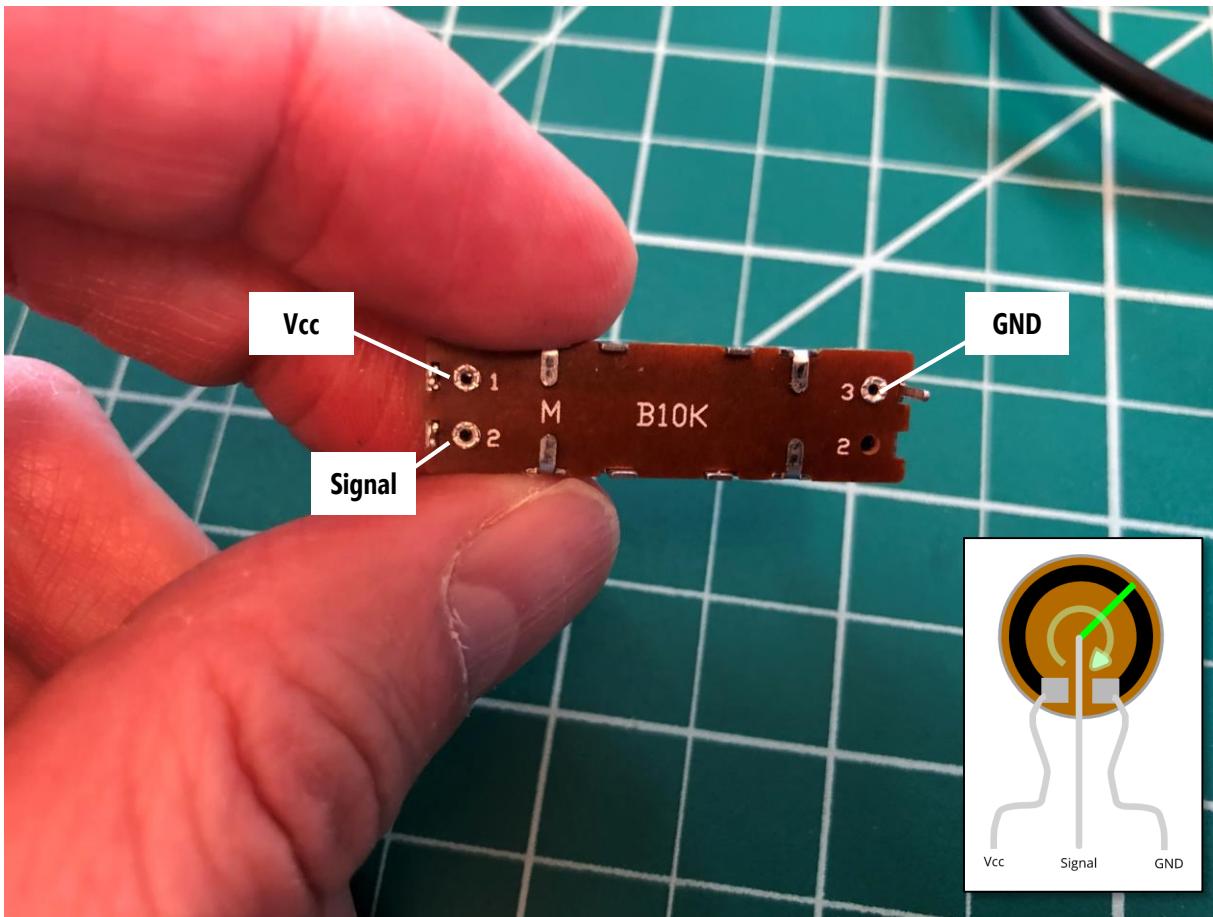






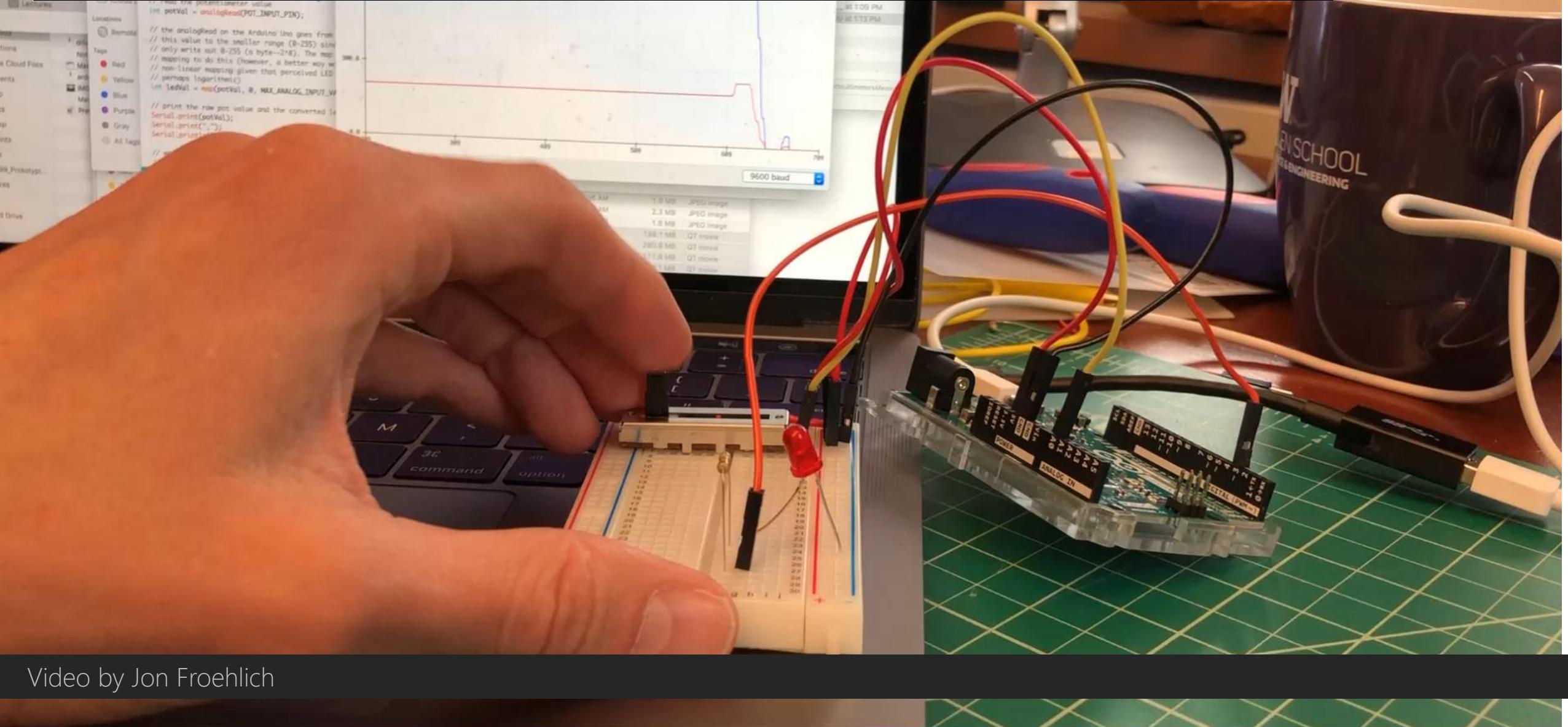
SLIDE POTENTIOMETERS

REMEMBER: EITHER ORIENTATION WORKS!



ANALOG INPUT

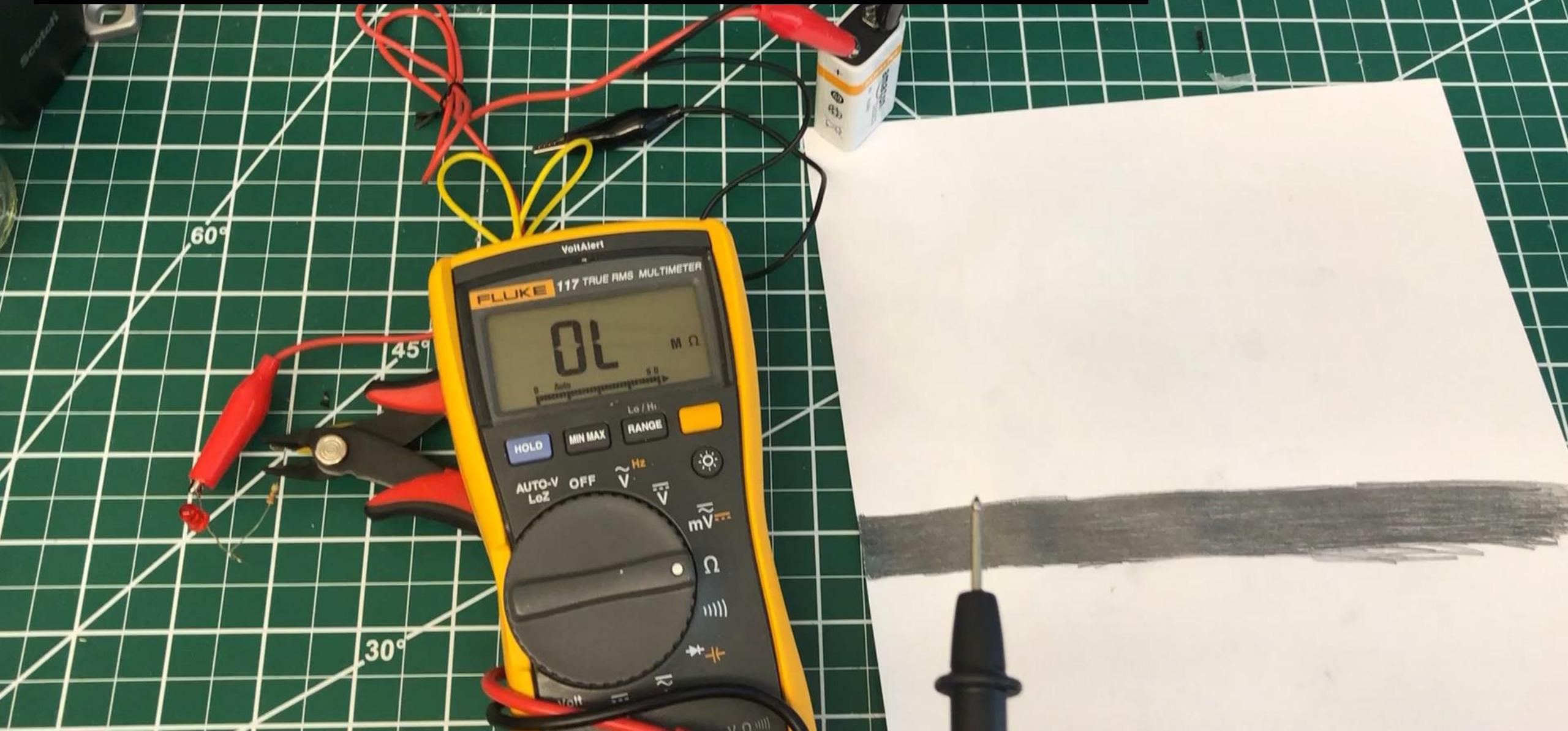
ACTIVITY: CONTROL LED BRIGHTNESS WITH A SLIDE POT



Video by Jon Froehlich

VARIABLE RESISTORS

MAKE A DIY SLIDE POT & HOOK UP TO ARDUINO



VARIABLE RESISTORS

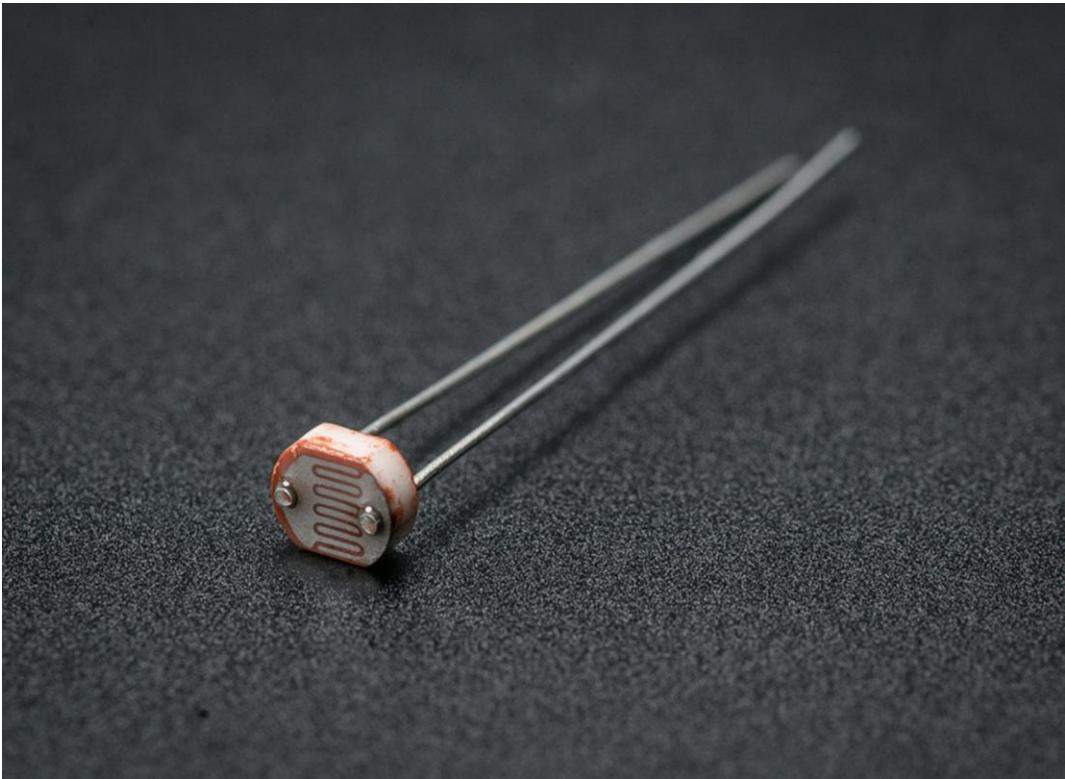
A GREAT RESOURCE: MAKING A LO-FI VARIABLE RESISTIVE INPUT

Source: Jeff Feddersen, Ohm Part 2, <https://vimeo.com/channels/pcomp/76442431>



PHOTORESISTORS AND HOW TO USE THEM

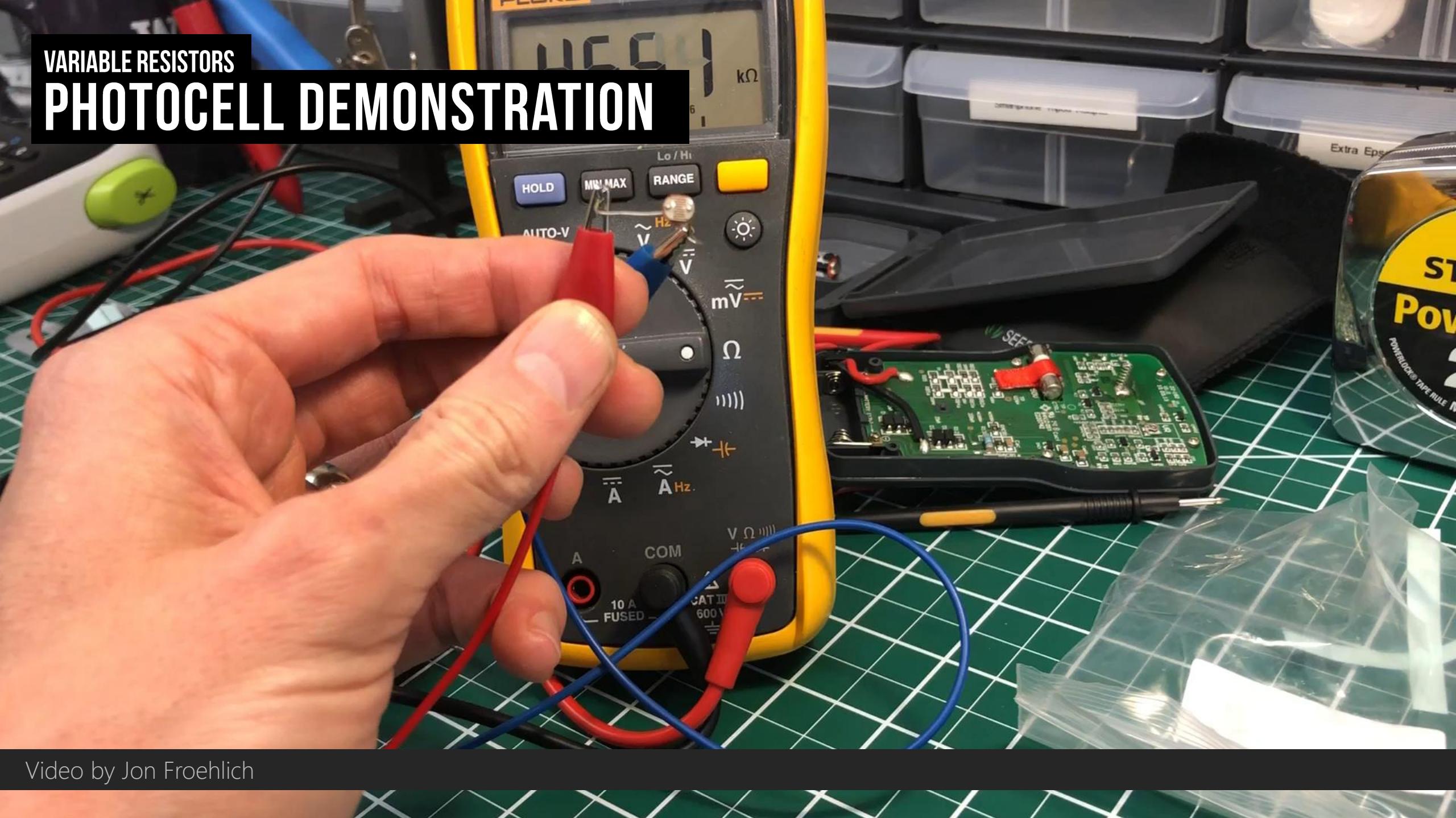
THE PHOTOSENSITIVE RESISTOR



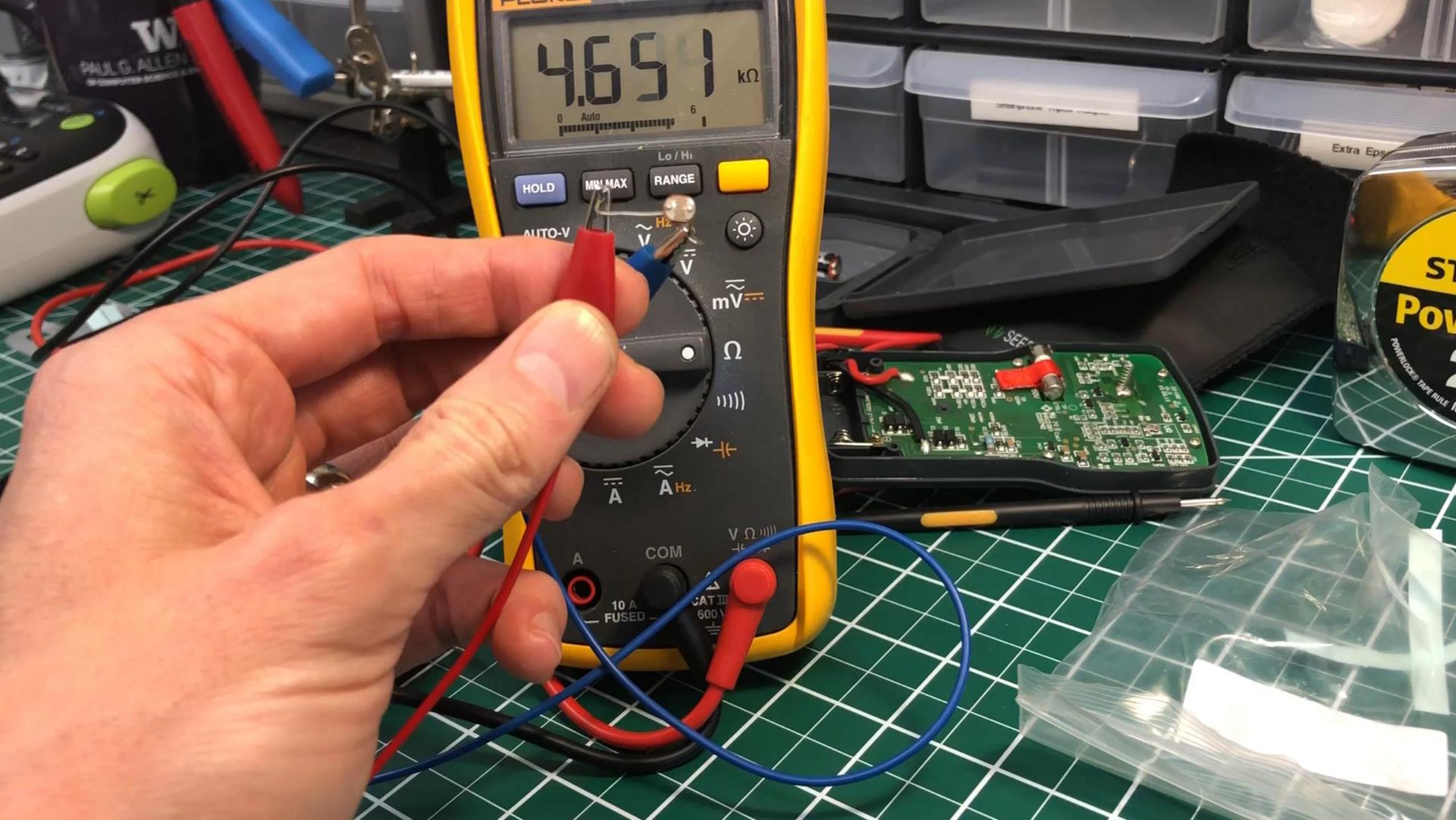
Photosensitive resistors or photocells **change their resistance based on light levels**. When light levels go up, resistance goes down.

VARIABLE RESISTORS

PHOTOCELL DEMONSTRATION



Video by Jon Froehlich



VARIABLE RESISTORS

PHOTOCELLS ARE EVERYWHERE!



Melissa & Doug

~crafted by hand~

SOUND PUZZLE

FIRE TRUCK

COUNTY
FIRE DEPT

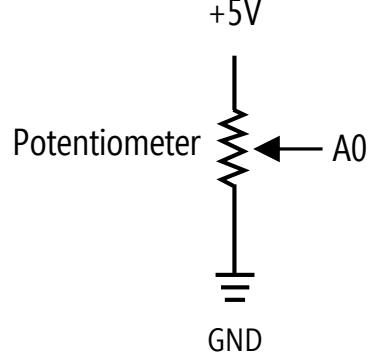
6

9-1-1

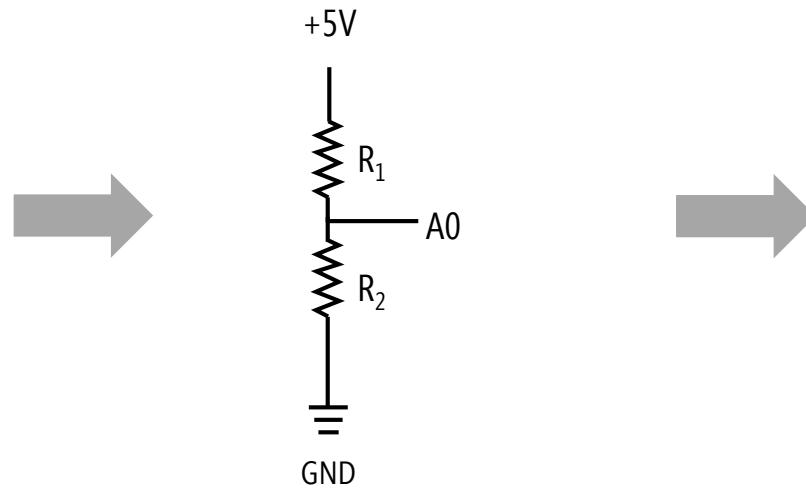


ANALOG INPUT

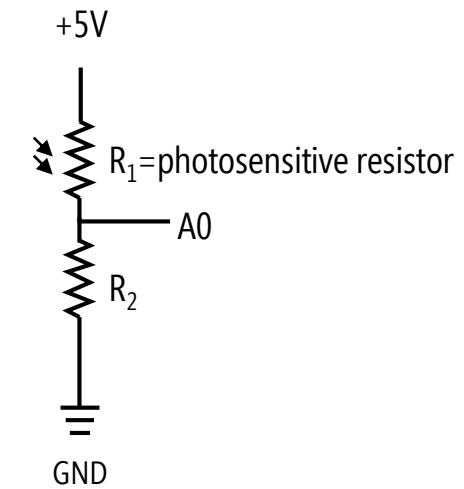
REPLACE POT WITH PHOTOSENSITIVE RESISTOR



Potentiometer
input circuit



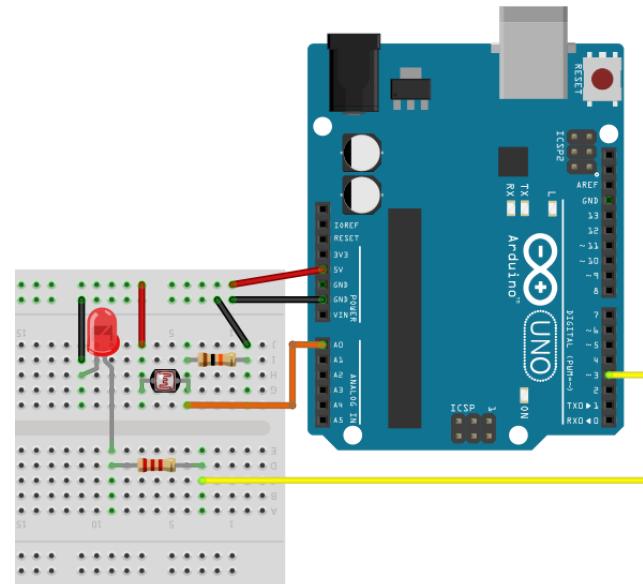
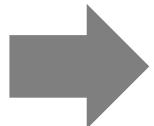
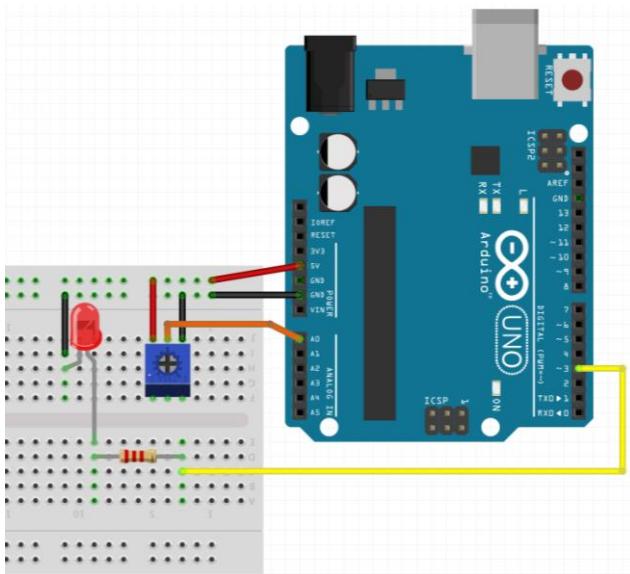
Potentiometer
input circuit is, by
design, a voltage divider



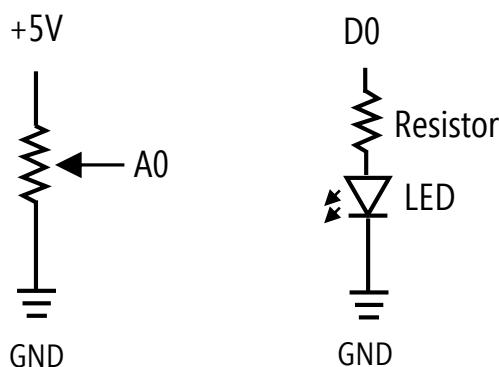
Let's make a similar
circuit but with a
photosensitive resistor

ANALOG INPUT

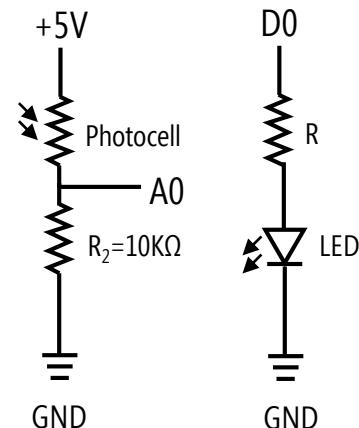
REPLACE POT WITH PHOTOSENSITIVE RESISTOR



Old Circuit with Potentiometer

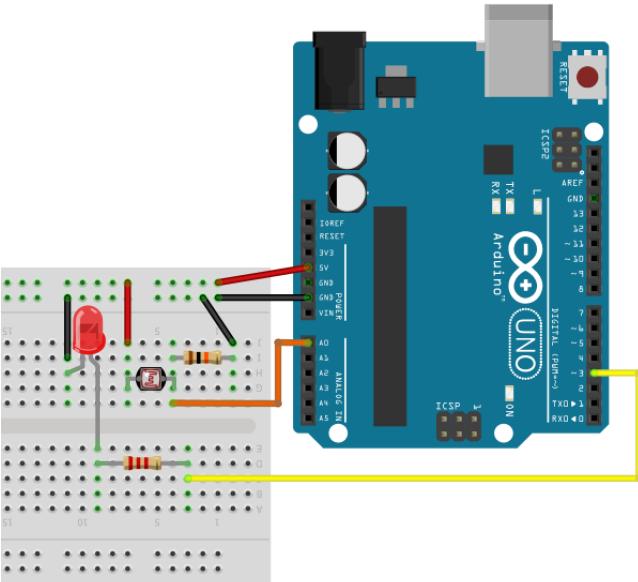


New Circuit with Photocell

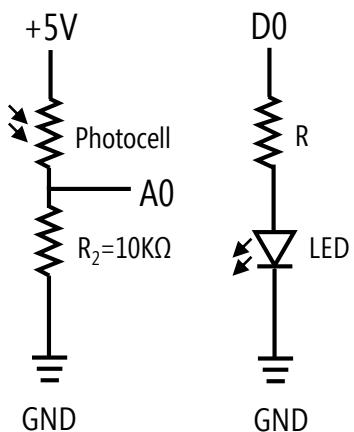


ANALOG INPUT

PHOTOCELL HOOKUP DIAGRAM

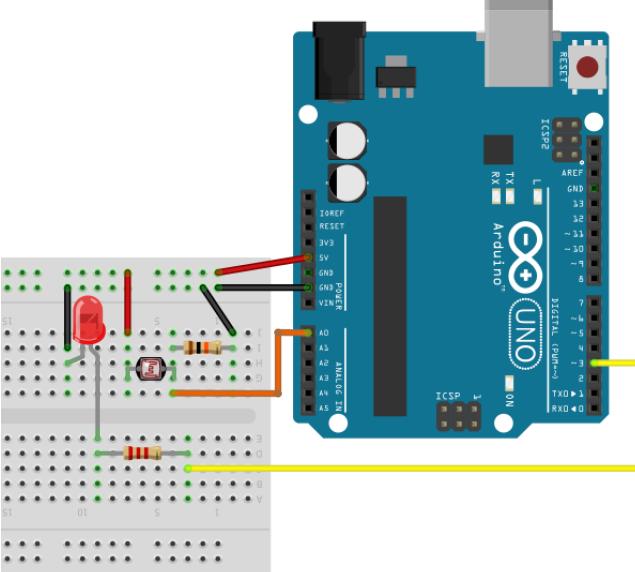


Correct Circuit with Photocell

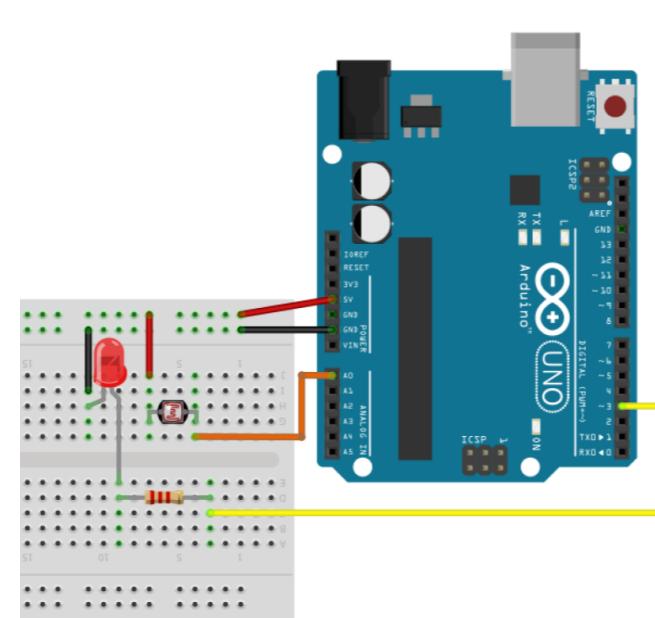
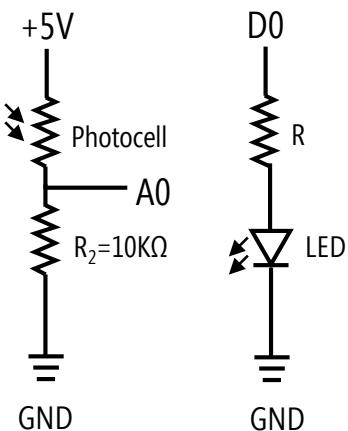


ANALOG INPUT

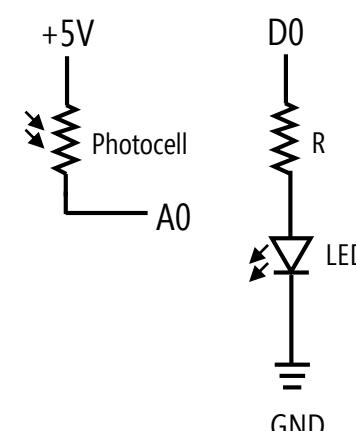
WHY CAN'T WE DO THIS?



Correct Circuit with Photocell

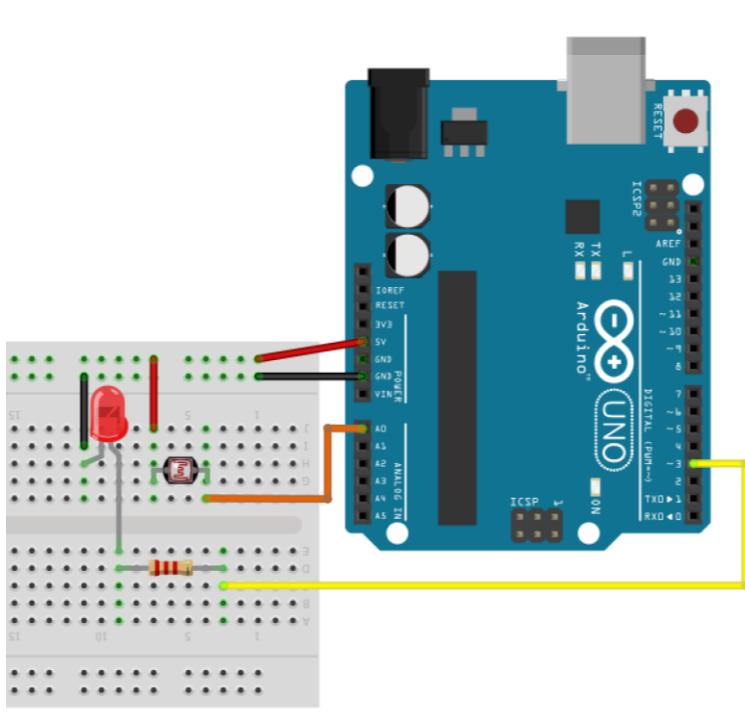


Incorrect Circuit with Photocell

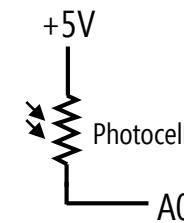


ANALOG INPUT

WHY CAN'T WE JUST DO THIS?

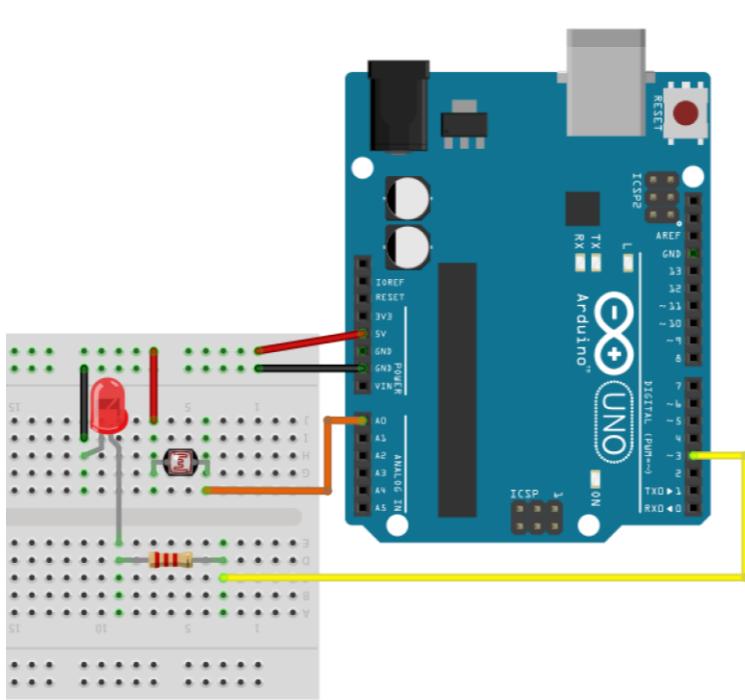


What's wrong with this circuit?

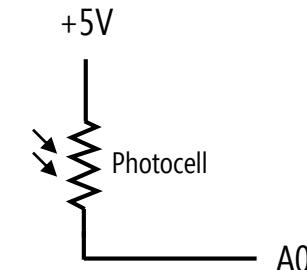


ANALOG INPUT

WHY CAN'T WE JUST DO THIS?

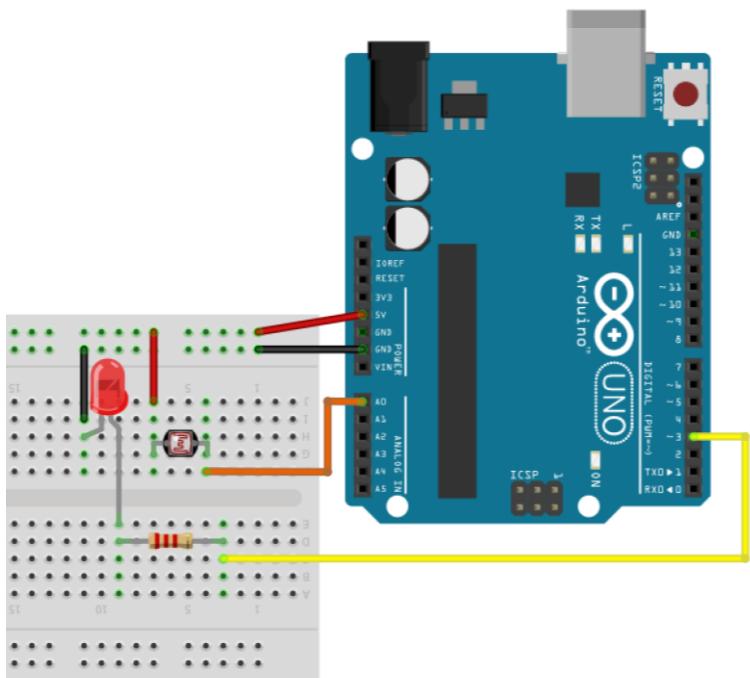


What's wrong with this circuit?

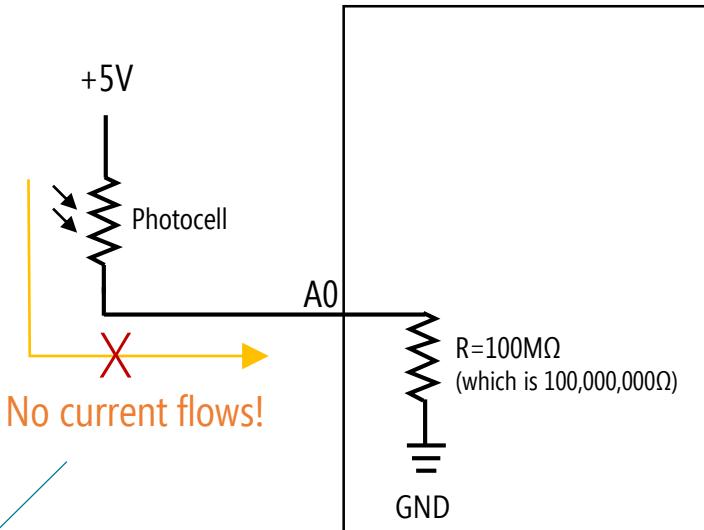


ANALOG INPUT

WHY CAN'T WE JUST DO THIS?



What's wrong with this circuit?



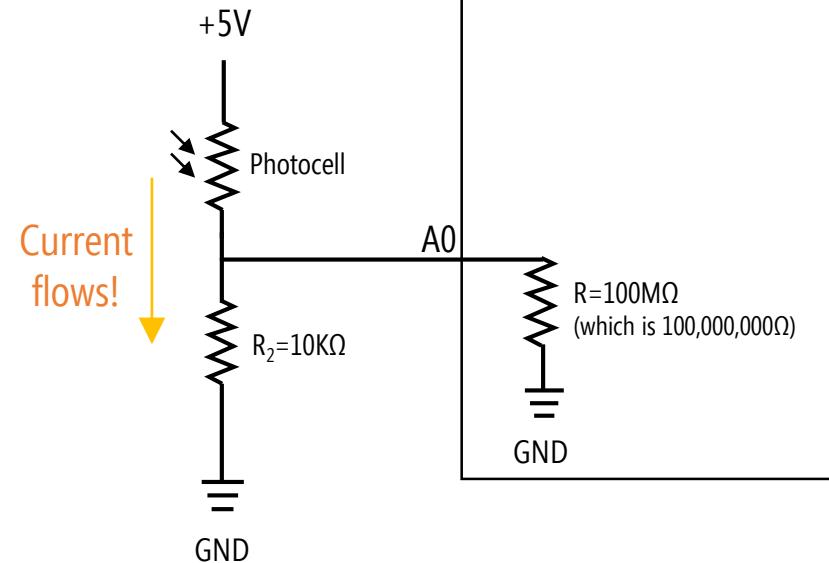
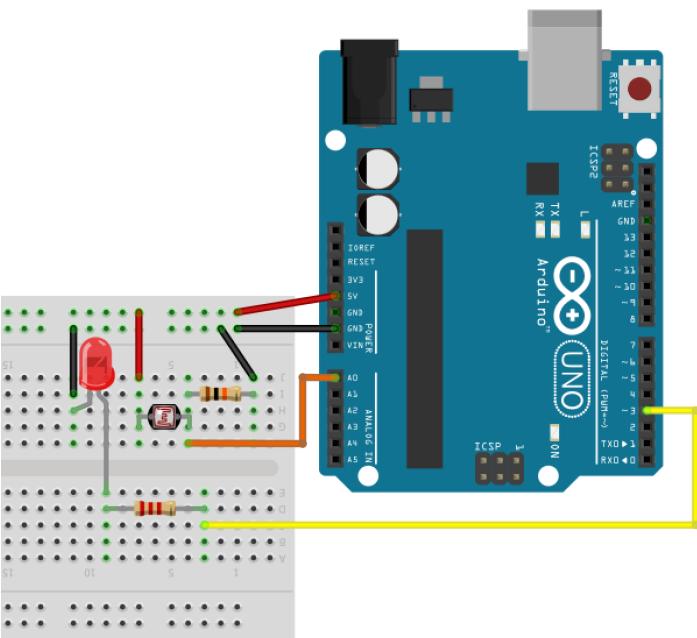
1. No current flows into the input pin because of extremely high resistance in the microcontroller

2. Microcontrollers read changes in voltage & not resistance or current

ANALOG INPUT

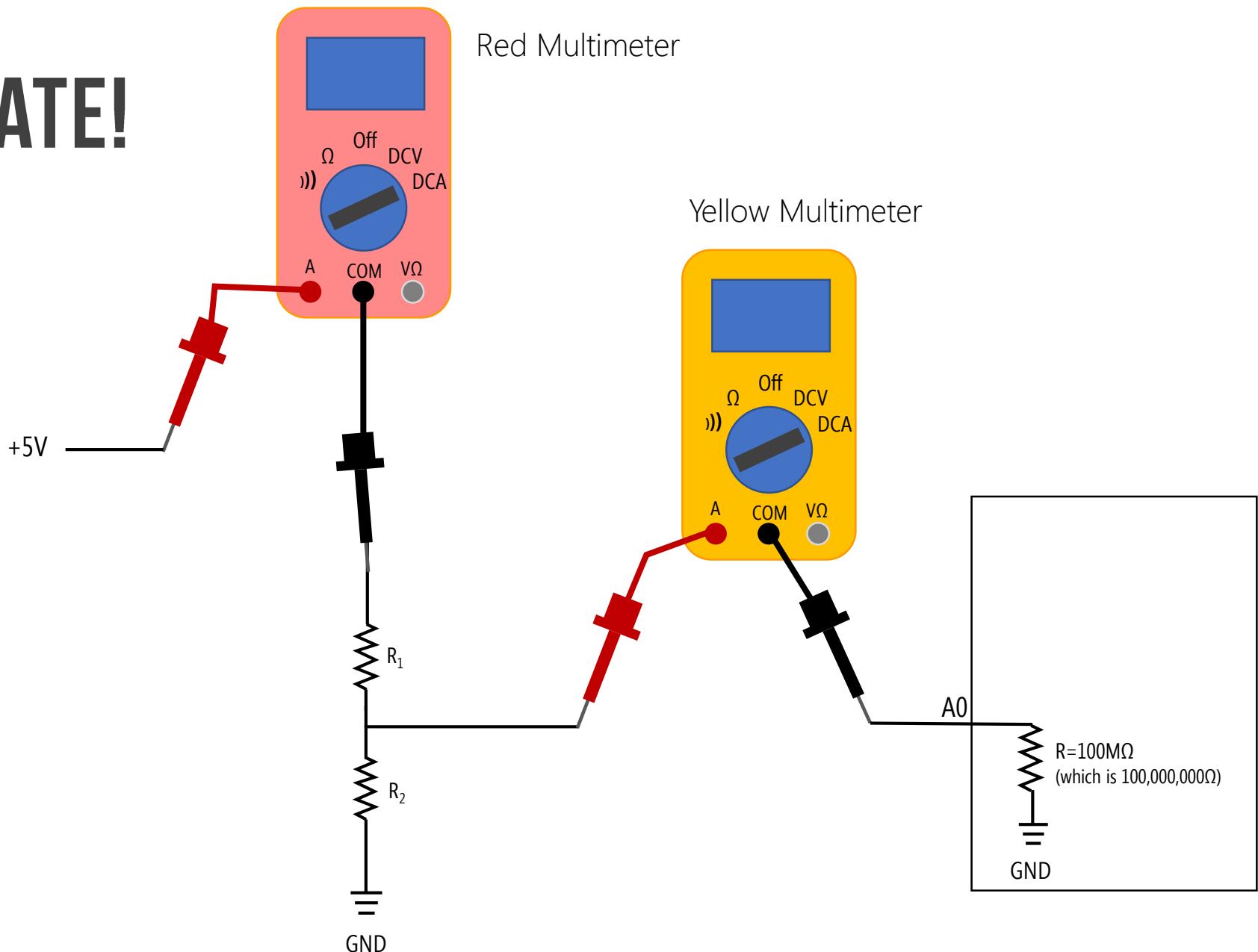
THE CORRECT CIRCUIT DESIGN

The correct circuit design



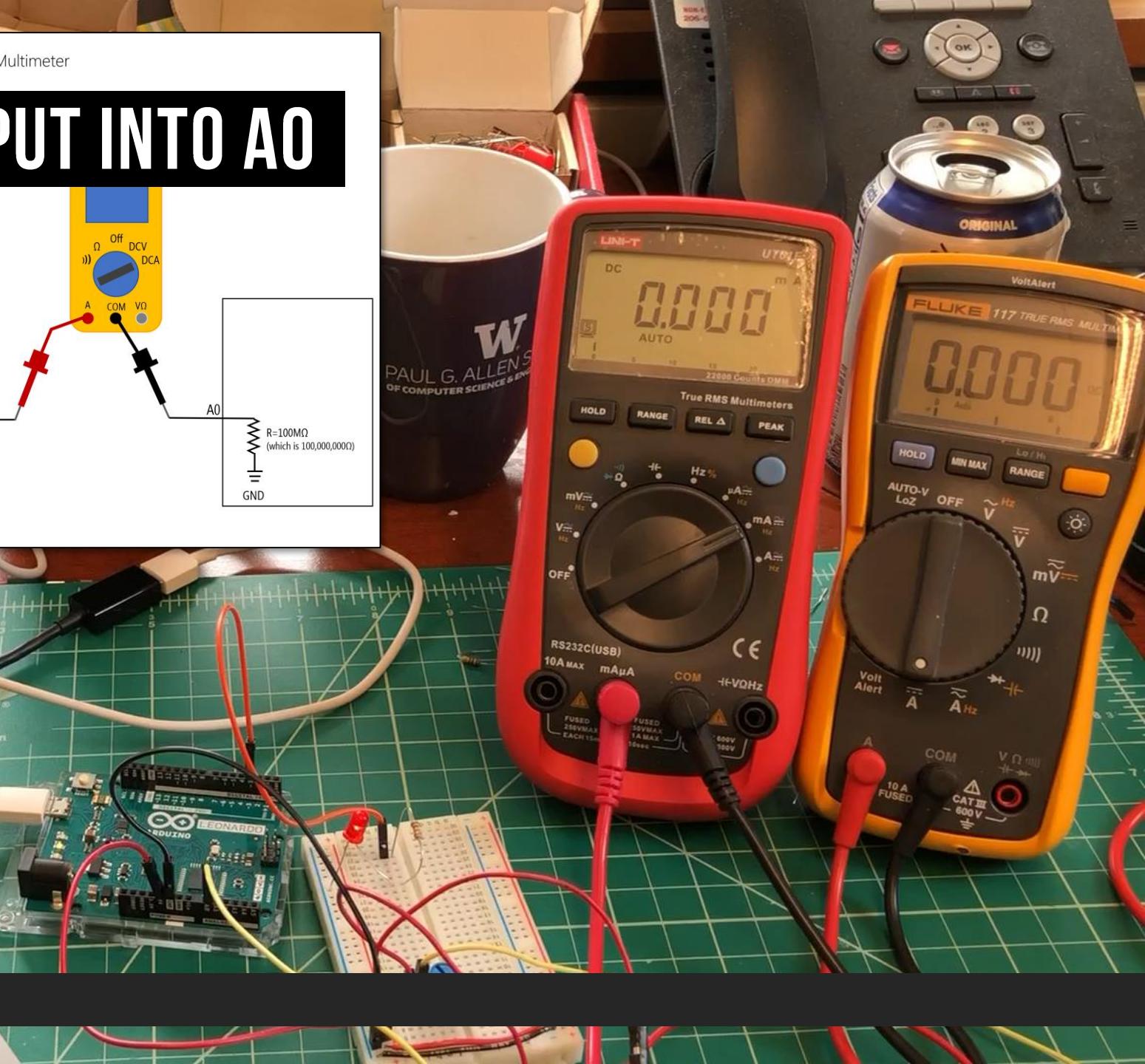
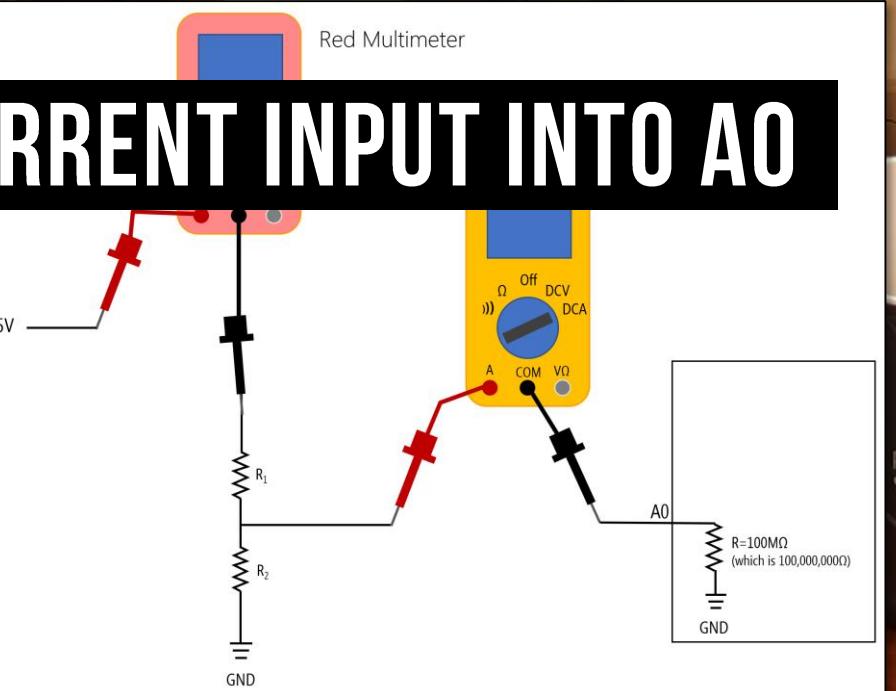
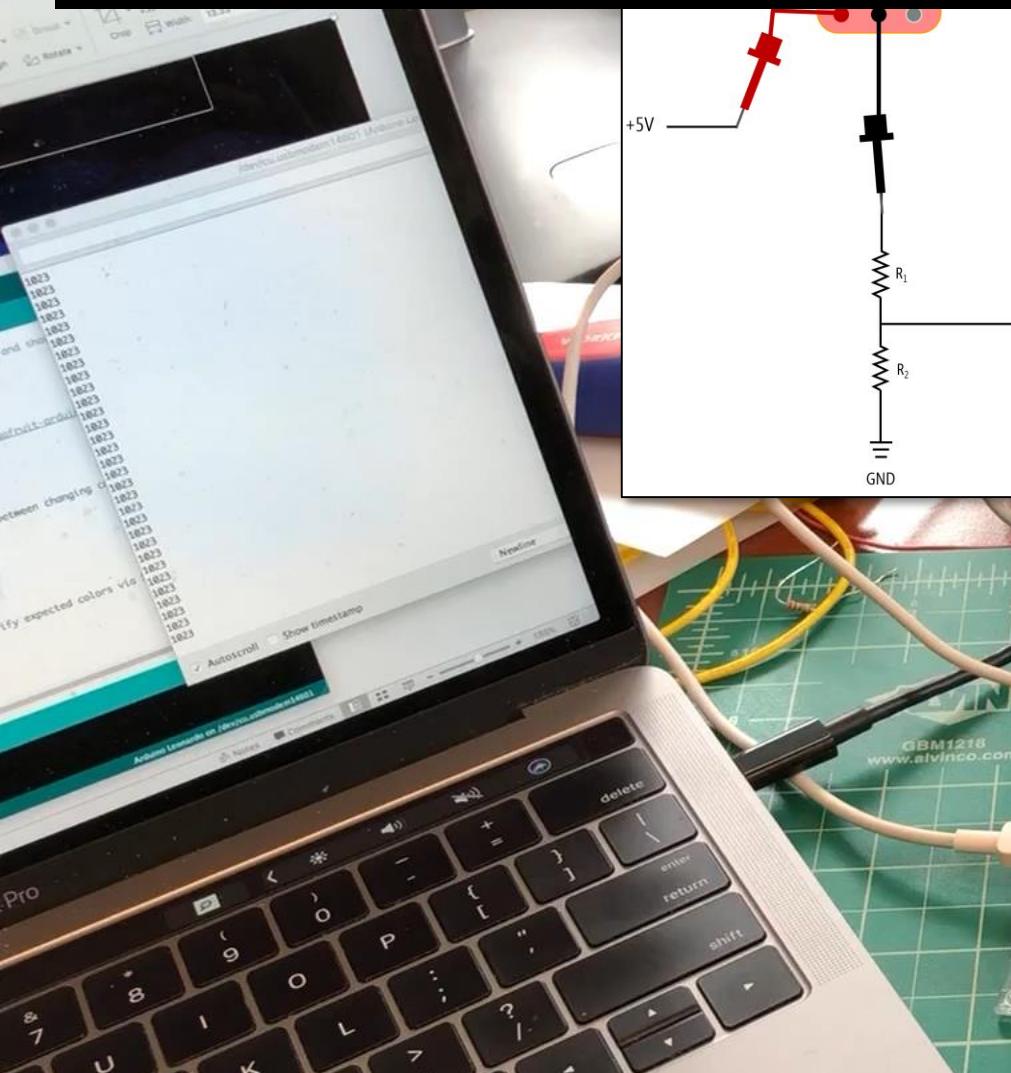
ANALOG INPUT

LET'S INVESTIGATE!

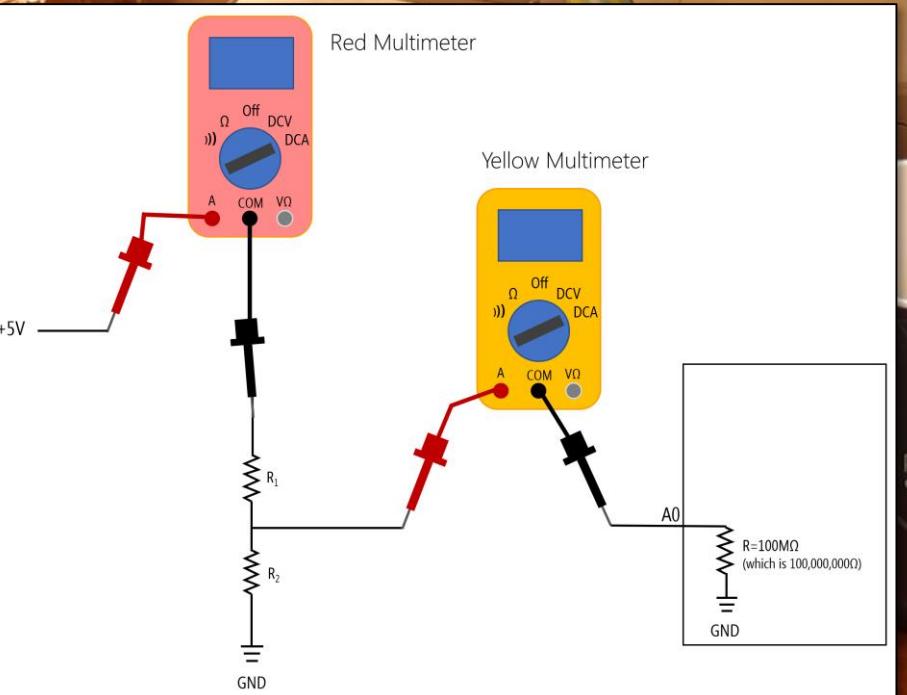
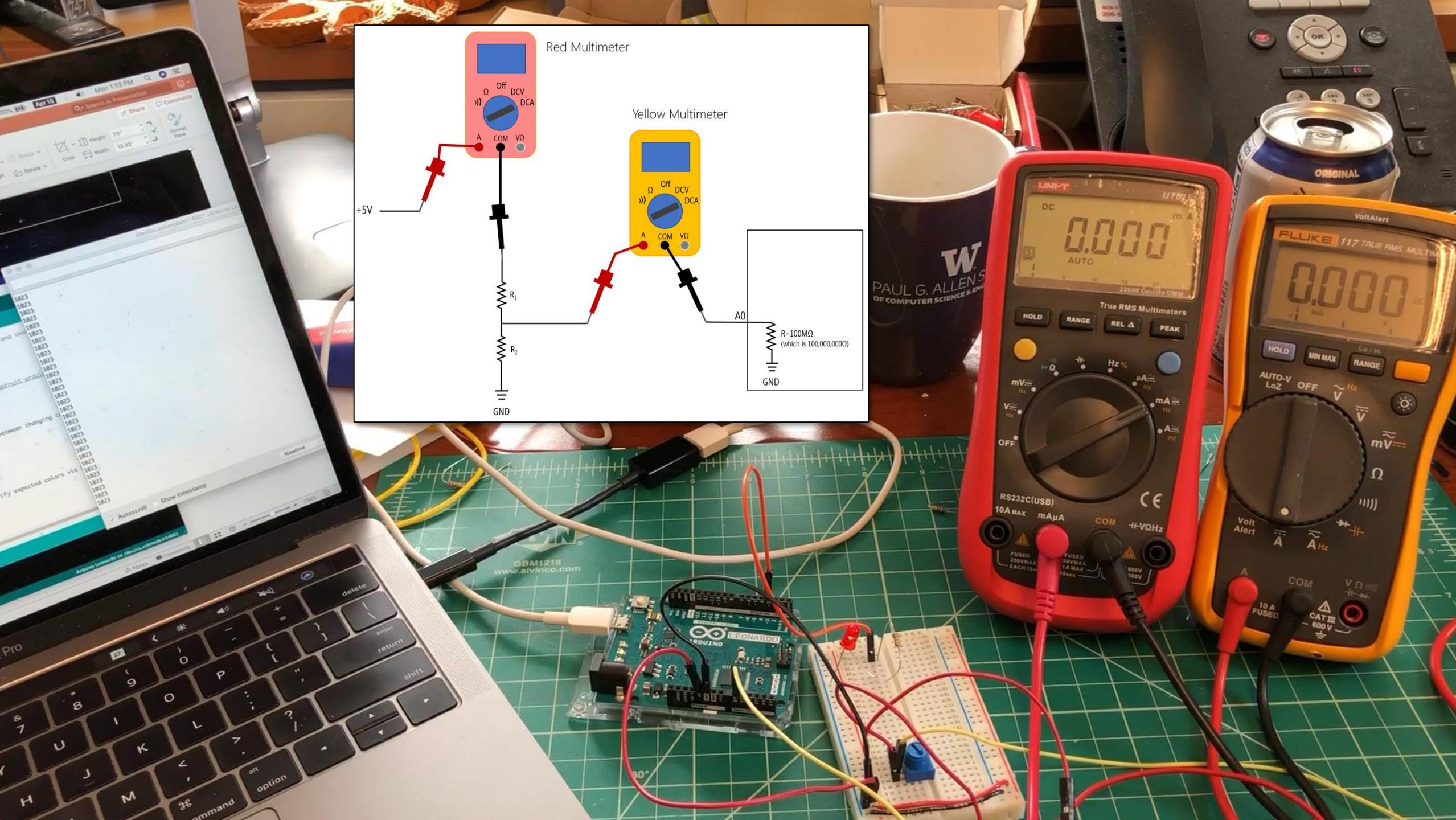


ANALOG INPUT

MEASURING CURRENT INPUT INTO AO

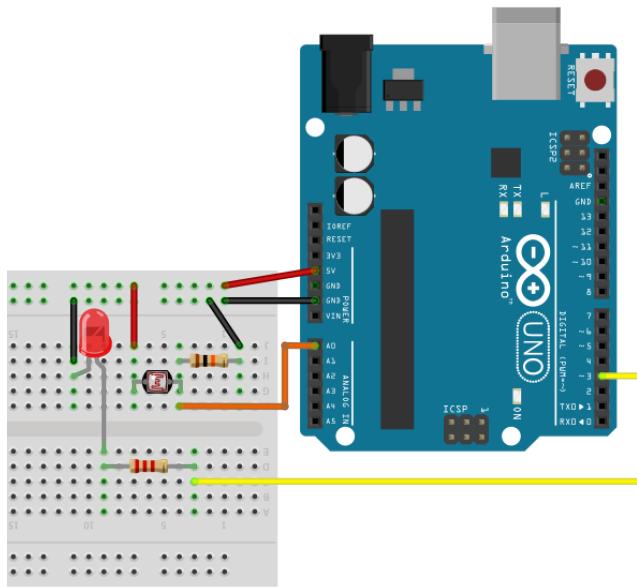


Video by Jon Froehlich

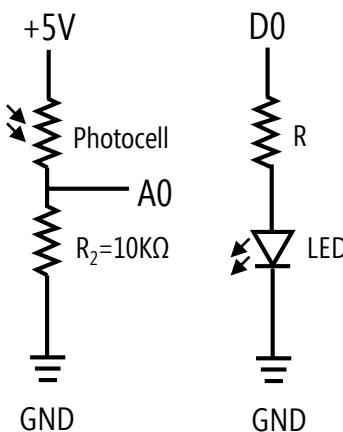


ANALOG INPUT

HOOKING UP PHOTOCELL



Correct Circuit with Photocell



PhotocellLED

```
const int LED_OUTPUT_PIN = 3;
const int PHOTOCELL_INPUT_PIN = A0;

// Set the min and max photocell values (this will be based on
// the brightness of your environment and the size of the voltage-divider
// resistor that you selected). So, the best way to set these values
// is to view the photocellVal in the Serial Monitor or Serial Plotter
// under different expected lighting conditions and observe the values
const int MIN_PHOTOCELL_VAL = 150; // Photocell reading in dark
const int MAX_PHOTOCELL_VAL = 1023; // Photocell reading in ambient light

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

void loop() {

    // Read the photo-sensitive resistor value. If you have the photocell resistor hooked
    // up as Rtop in the voltage divider (that is, one leg of the photocell is connected to 5V),
    // then higher values correspond to brightness. If you have the photocell hooked up as Rbottom
    // in the voltage divider (that is, one leg of the photocell is connected to GND), then
    // higher values correspond to darkness.
    int photocellVal = analogRead(PHOTOCELL_INPUT_PIN);

    // Remap the value for output.
    int ledVal = map(photocellVal, MIN_PHOTOCELL_VAL, MAX_PHOTOCELL_VAL, 0, 255);

    // The map function does not constrain output outside of the provided range
    // so, we need to make sure that things are within range for the led
    ledVal = constrain(ledVal, 0, 255);

    // We want to invert the LED (it should be brighter when environment is darker)
    // This assumes the photocell is Rtop in the voltage divider
    ledVal = 255 - ledVal;

    // Print the raw photocell value and the converted led value (e.g., for Serial Plotter)
    Serial.print(photocellVal);
    Serial.print(",");
    Serial.println(ledVal);

    // Write out the LED value.
    analogWrite(LED_OUTPUT_PIN, ledVal);

    delay(100);
}
```

ANALOG INPUT

CONTROL LED BRIGHTNESS WITH A PHOTOCELL

```
* By Jon Froehlich
* http://makeabilitylab.io
*
*/
const int LED_OUTPUT_PIN = 3;
const int PHOTOCELL_INPUT_PIN = A0;

// Set the min and max photocell values (this will be based on
// the brightness of your environment and the size of the voltage-
// divisor that you selected). So, the best way to set these val-
// ues is to view the photocellVal in the Serial Monitor or Serial
// Plotter under different expected lighting conditions and observe the va-
// lues.
const int MIN_PHOTOCELL_VAL = 500; // Photocell reading in dark
const int MAX_PHOTOCELL_VAL = 850; // Photocell reading in ambient

void setup() {
}

void loop() {
    int photocellVal = analogRead(PHOTOCELL_INPUT_PIN);
    if (photocellVal >= MIN_PHOTOCELL_VAL & photocellVal <= MAX_PHOTOCELL_VAL) {
        analogWrite(LED_OUTPUT_PIN, photocellVal);
    }
}
```

Upload done. Thank you.

Contributors

Simon Monk

The photocell is at the bottom of the breadboard.

```
const int TRIG_PIN = 7;
HO_PIN = 8;
```

/dev/cu.usbmodem14401 (Arduino/Genuino Uno)

Send

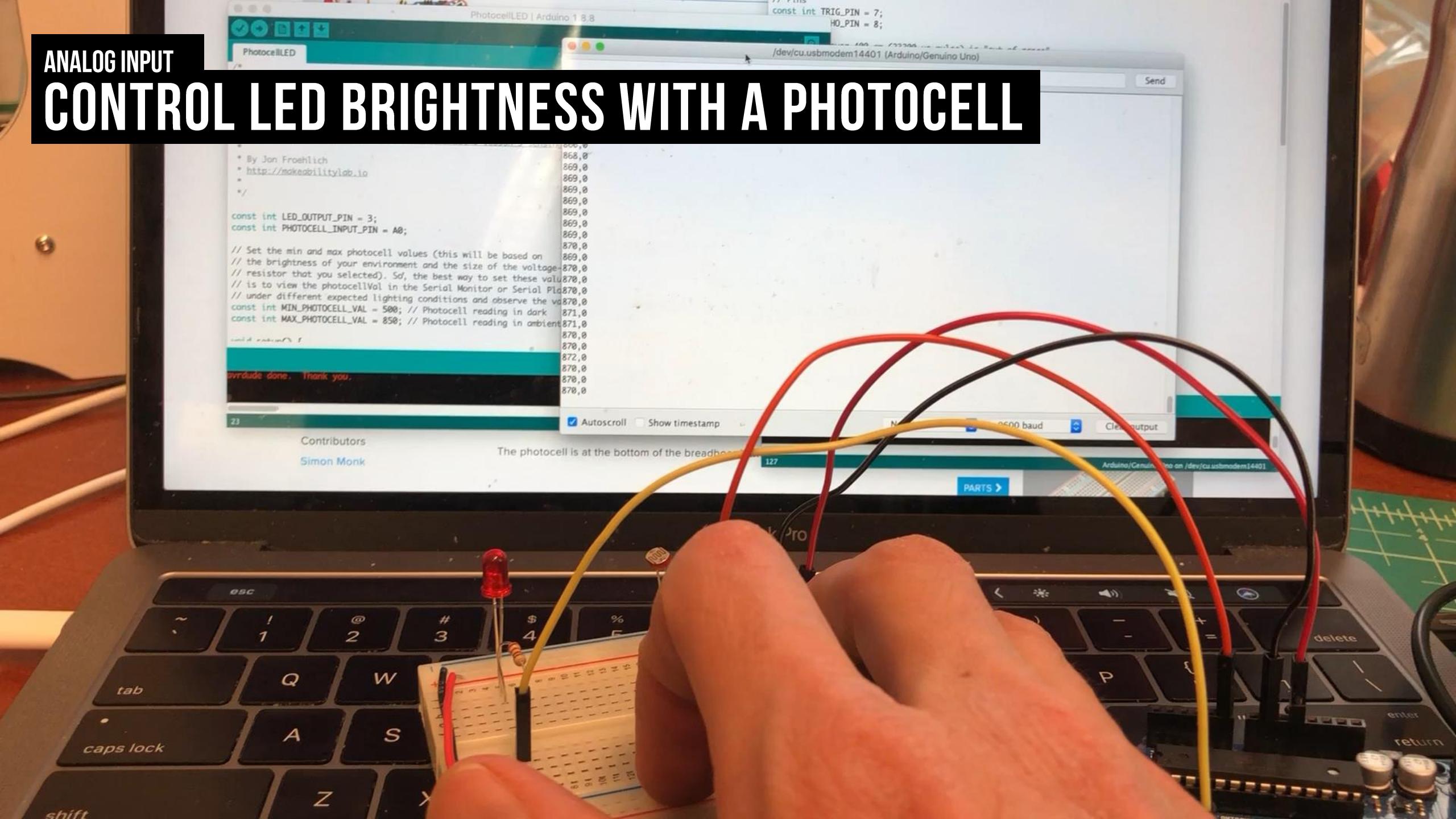
2500 baud

Clear output

Arduino/Genuino Uno on /dev/cu.usbmodem14401

127

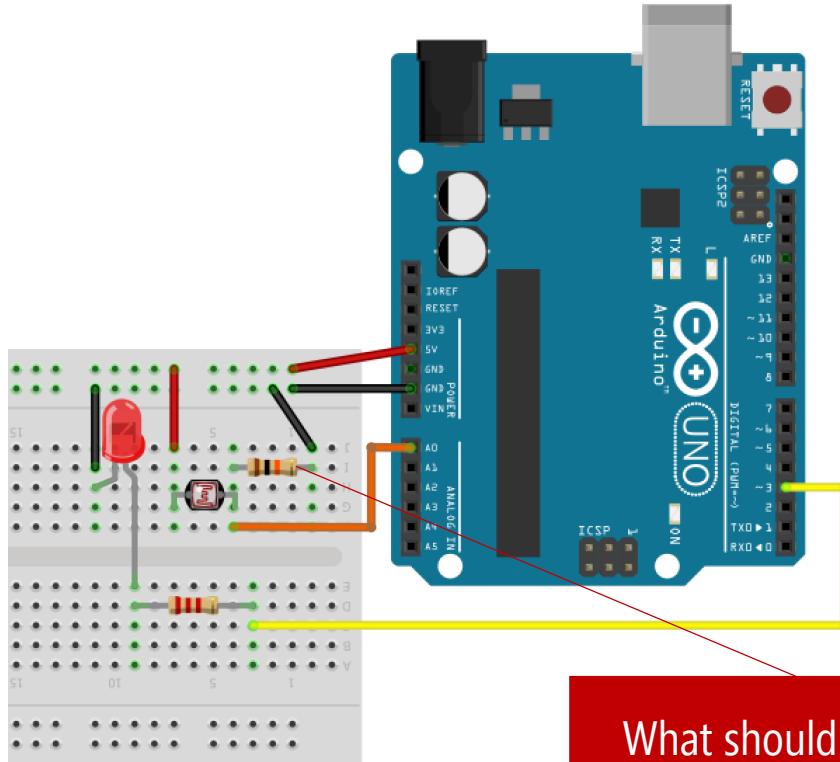
PARTS >



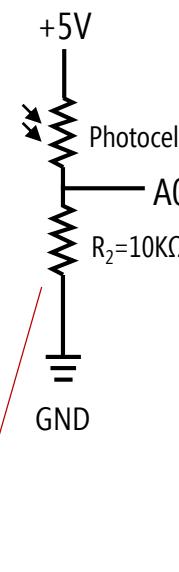
USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2}$$

HOW DO I KNOW WHAT TO SELECT FOR R₂?



What should we make this **R₂ resistor value**? And how do we know?



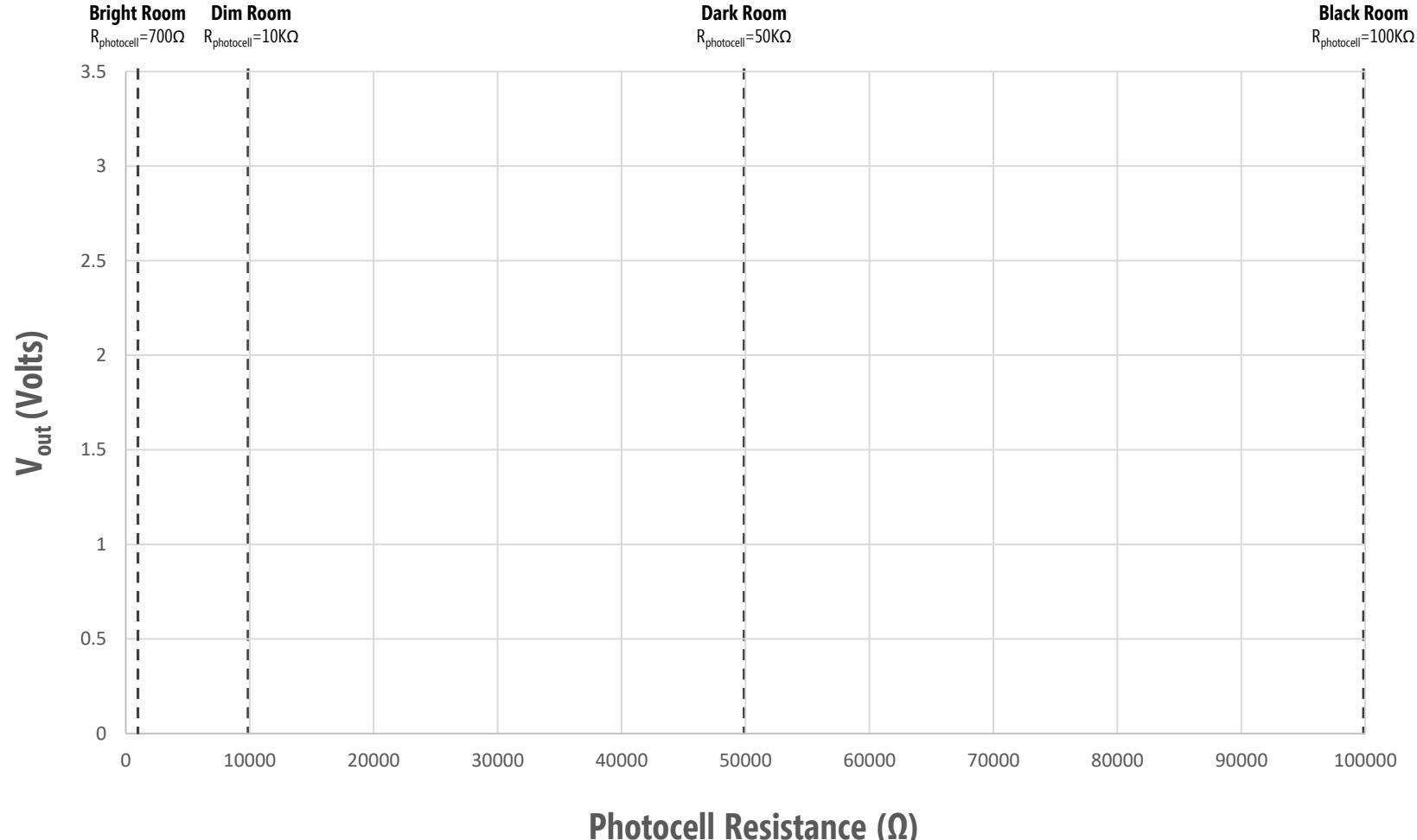
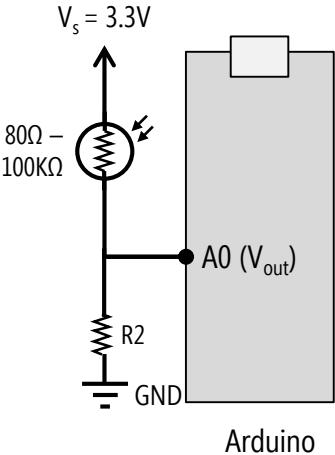
It depends on the **resistance range** of your variable resistor (R1)

It depends on the **context of use** (what's the expected resistance range of R1)

USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

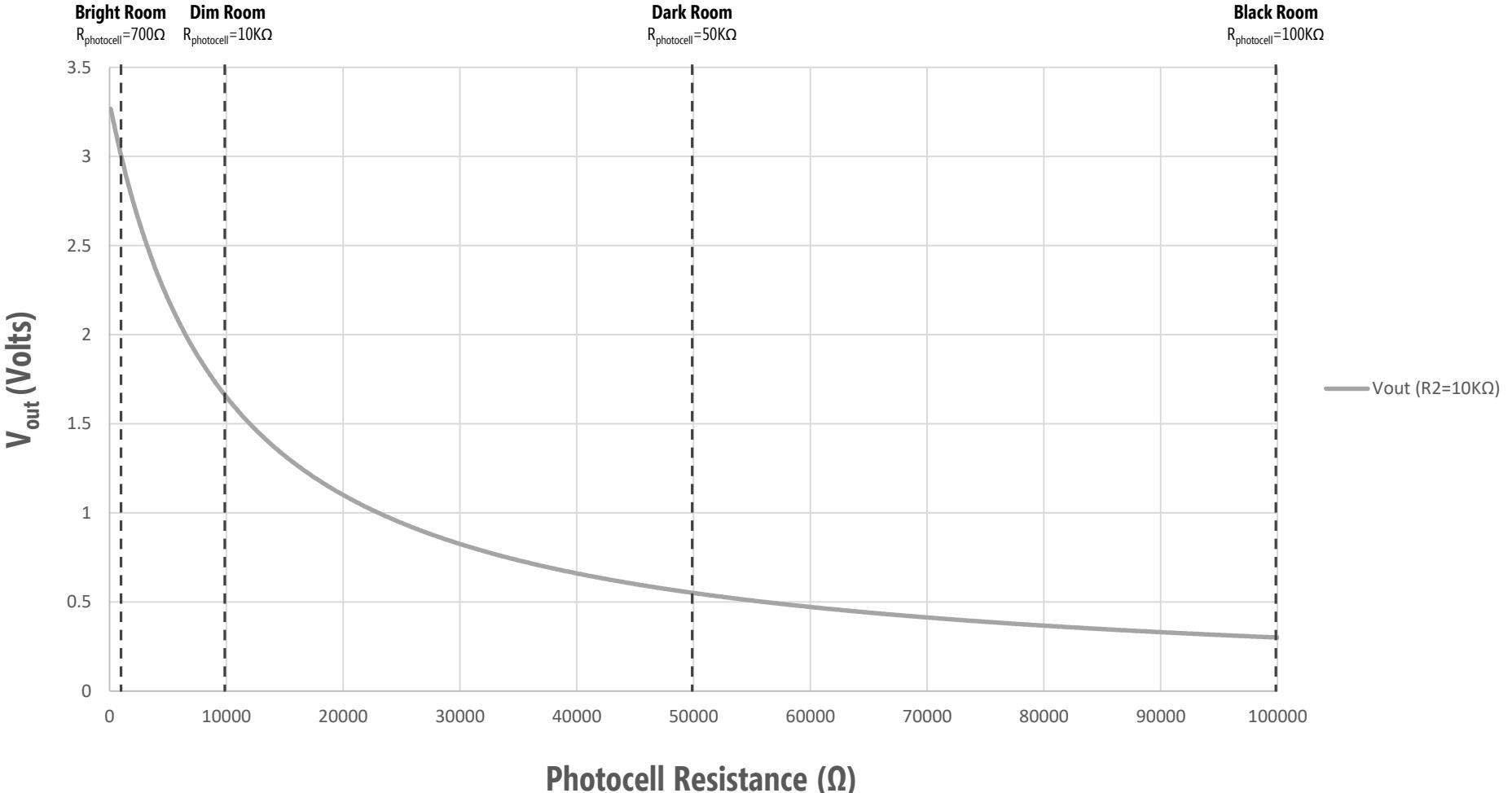
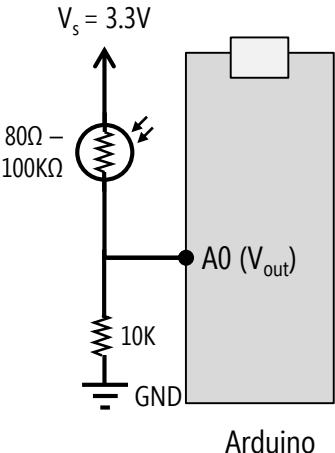
GRAPHING VOUT FOR PHOTOCELL



USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

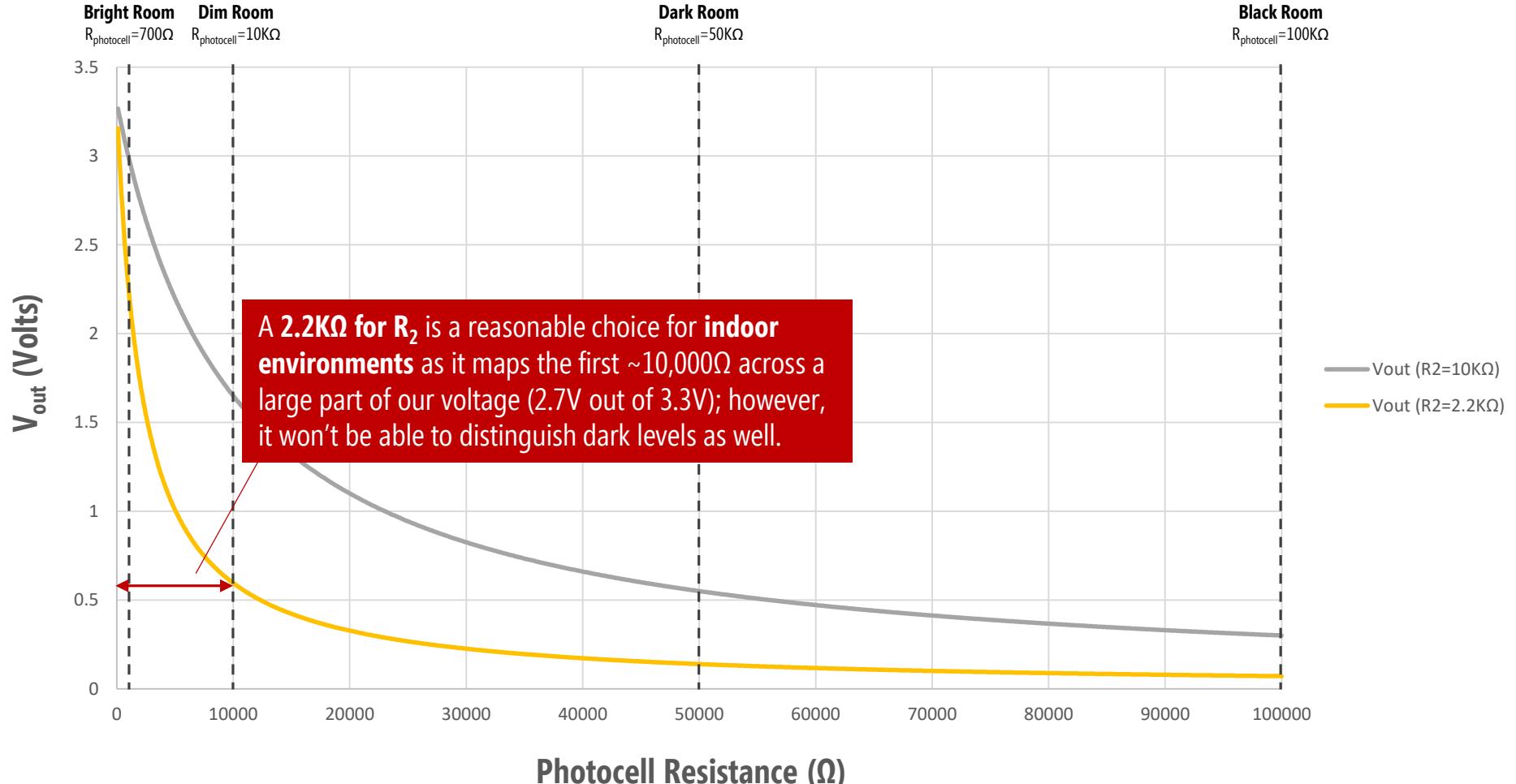
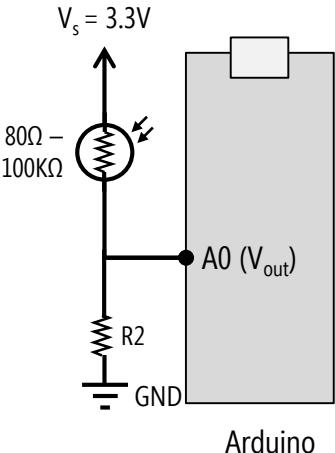
GRAPHING VOUT FOR PHOTOCELL



USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

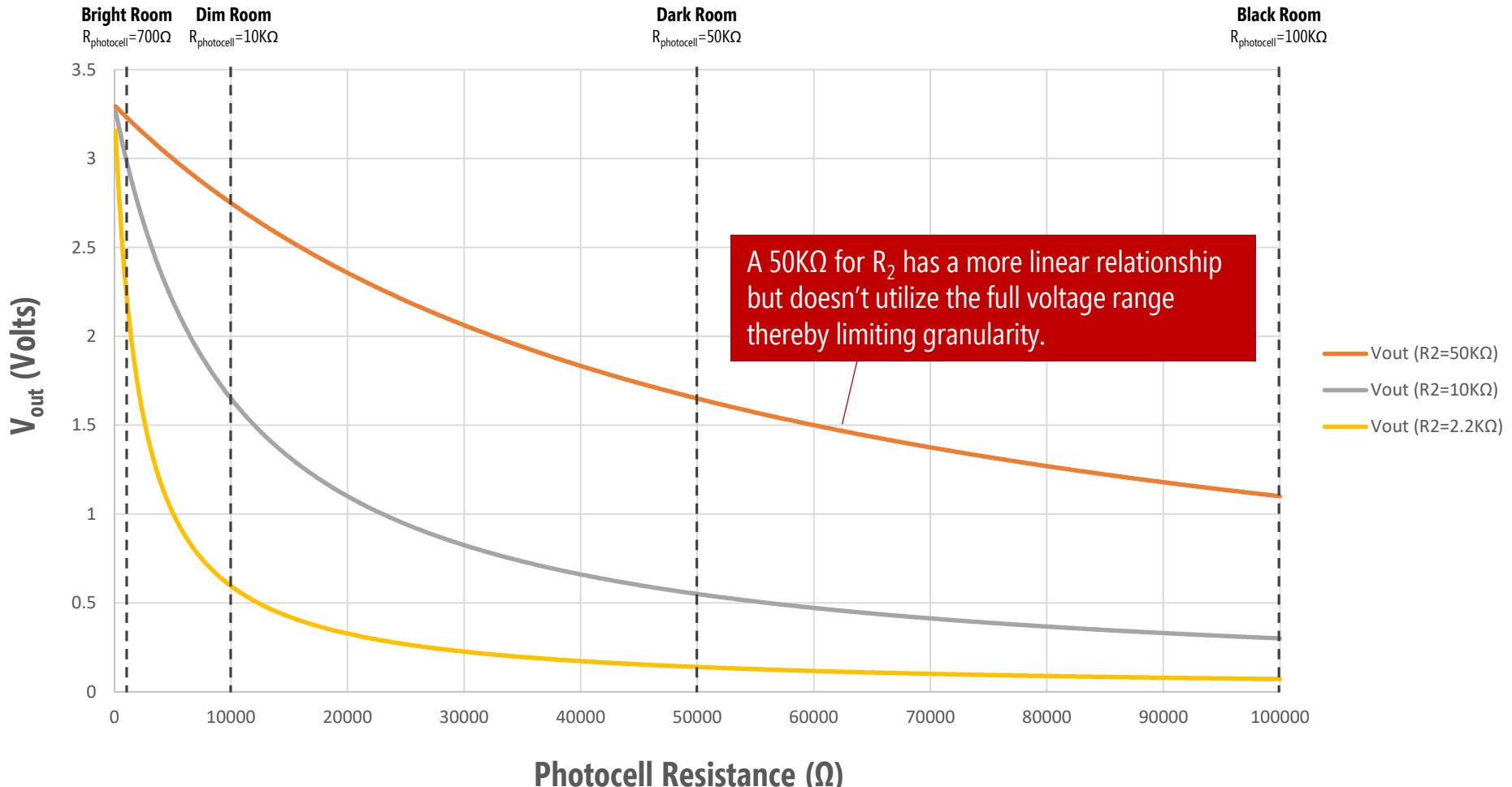
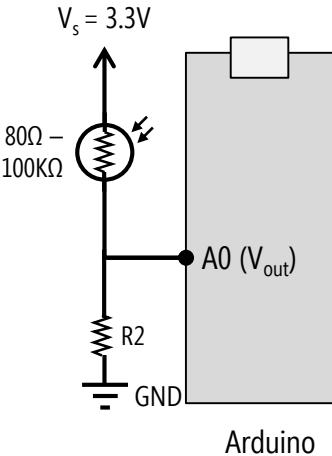
GRAPHING VOUT FOR PHOTOCELL



USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

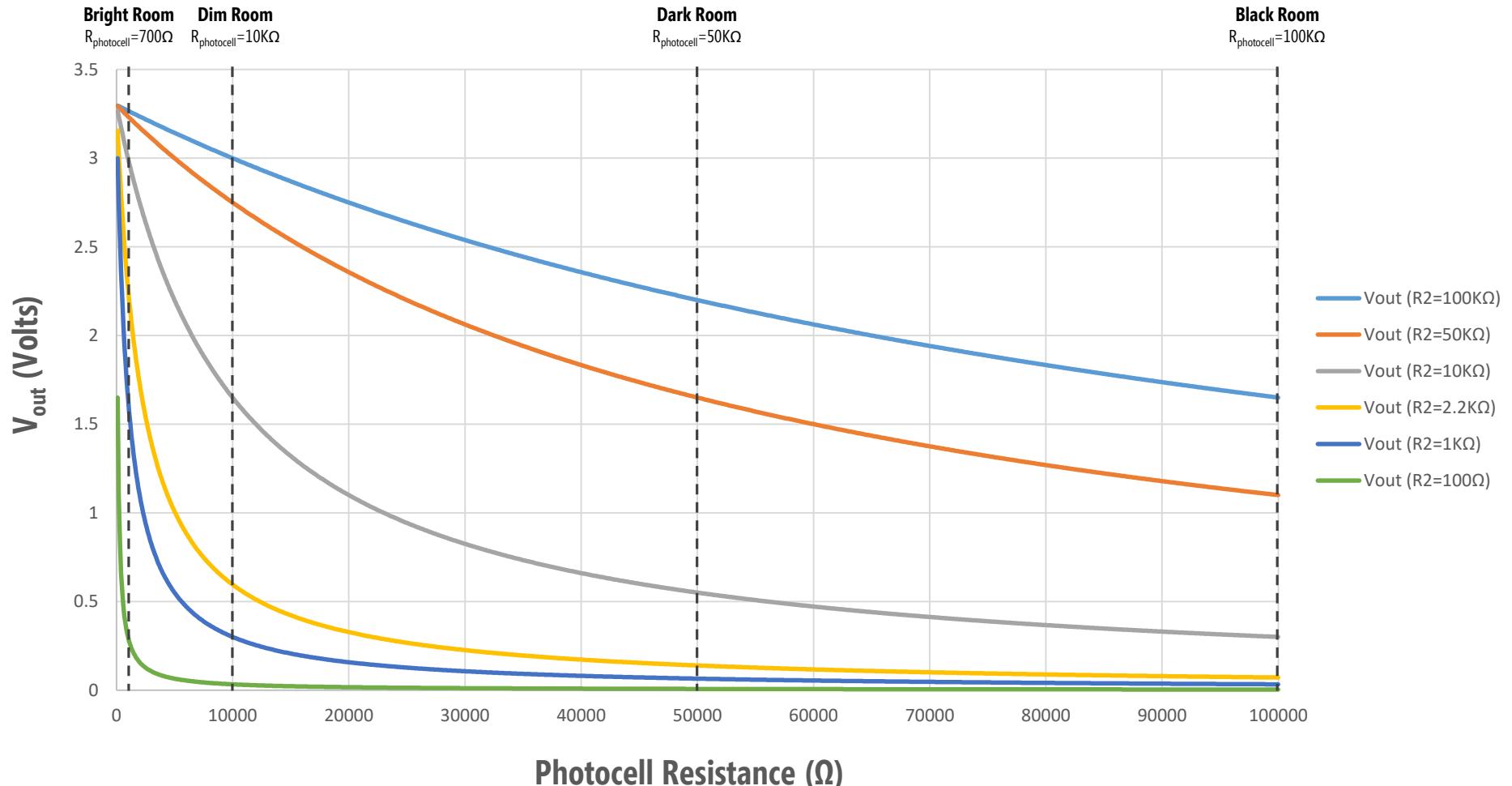
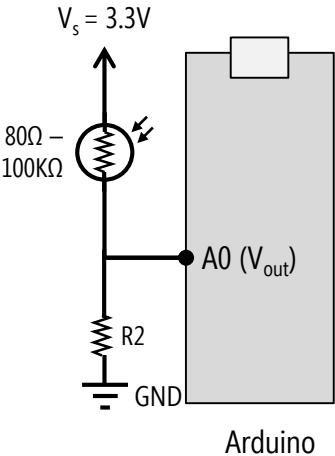
GRAPHING VOUT FOR PHOTOCELL



USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

GRAPHING VOUT FOR PHOTOCELL

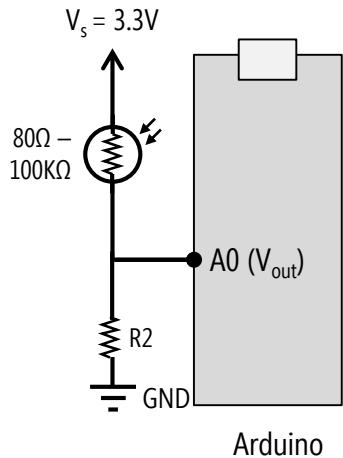


USING A VARIABLE RESISTIVE SENSOR

$$V_{out} = Vin * \frac{R_2}{R_1 + R_2}$$

AXEL BENZ FORMULATION FOR R2

First, measure the R_{min} and R_{max} values for your specific environment using a multimeter



Then use these measurements to find the value for R_2

$$R_2 = \sqrt{R_{1min} * R_{1max}}$$

$$R_2 = \sqrt{80\Omega * 10,000\Omega}$$

$$R_2 = \sim 900\Omega$$

USING A VARIABLE RESISTIVE SENSOR

EXPERIMENTATION

You'll also experiment with setting thresholds in the mapping function based on empirical observation

RedBearDuoReadPhotocellFadeLED

```
SYSTEM_MODE(MANUAL);

const int LED_OUTPUT_PIN = D0;
const int PHOTOCELL_INPUT_PIN = A0;

// Set the min and max photocell values (this will be based on
// the brightness of your environment and the size of the voltage-divider
// resistor that you selected).
const int MIN_PHOTOCELL_VAL = 1200; // Photocell reading in dark
const int MAX_PHOTOCELL_VAL = 3700; // Photocell reading in ambient light (tested in my office)
```

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

```
void loop() {
  // Read the photo-sensitive resistor value
  int photocellVal = analogRead(PHOTOCELL_INPUT_PIN);
```

```
// Remap the value for output.
int ledVal = map(photocellVal, MIN_PHOTOCELL_VAL, MAX_PHOTOCELL_VAL, 0, 255);
```

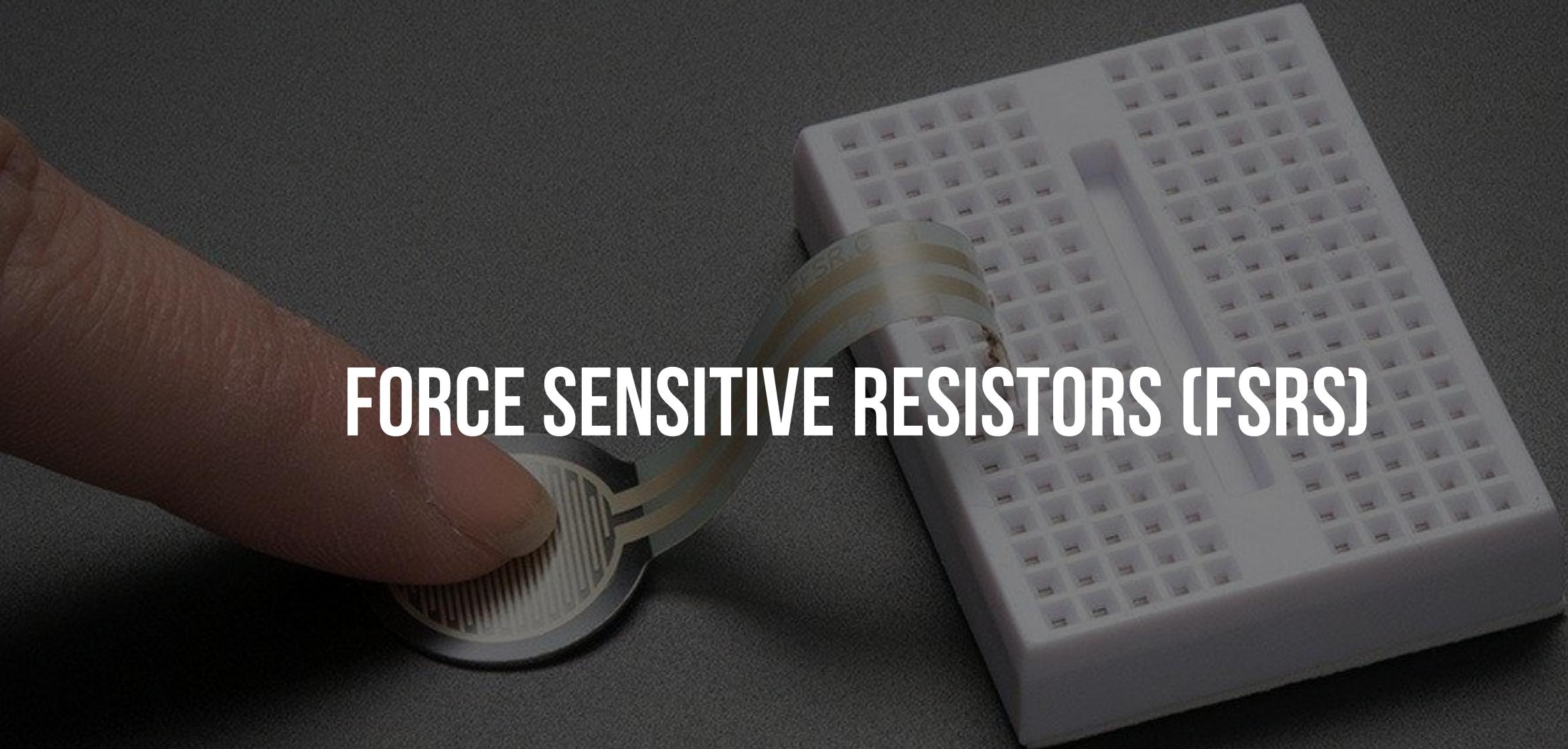
```
// The map function does not constrain output outside of the provided range
// so, we need to make sure that things are within range for the led
ledVal = constrain(ledVal, 0, 255);
```

```
// We want to invert the LED (it should be brighter when environment is darker)
ledVal = 255 - ledVal;
```

```
// Print the raw photocell value and the converted led value (e.g., for Serial Plotter)
Serial.print(photocellVal);
Serial.print(",");
Serial.println(ledVal);
```

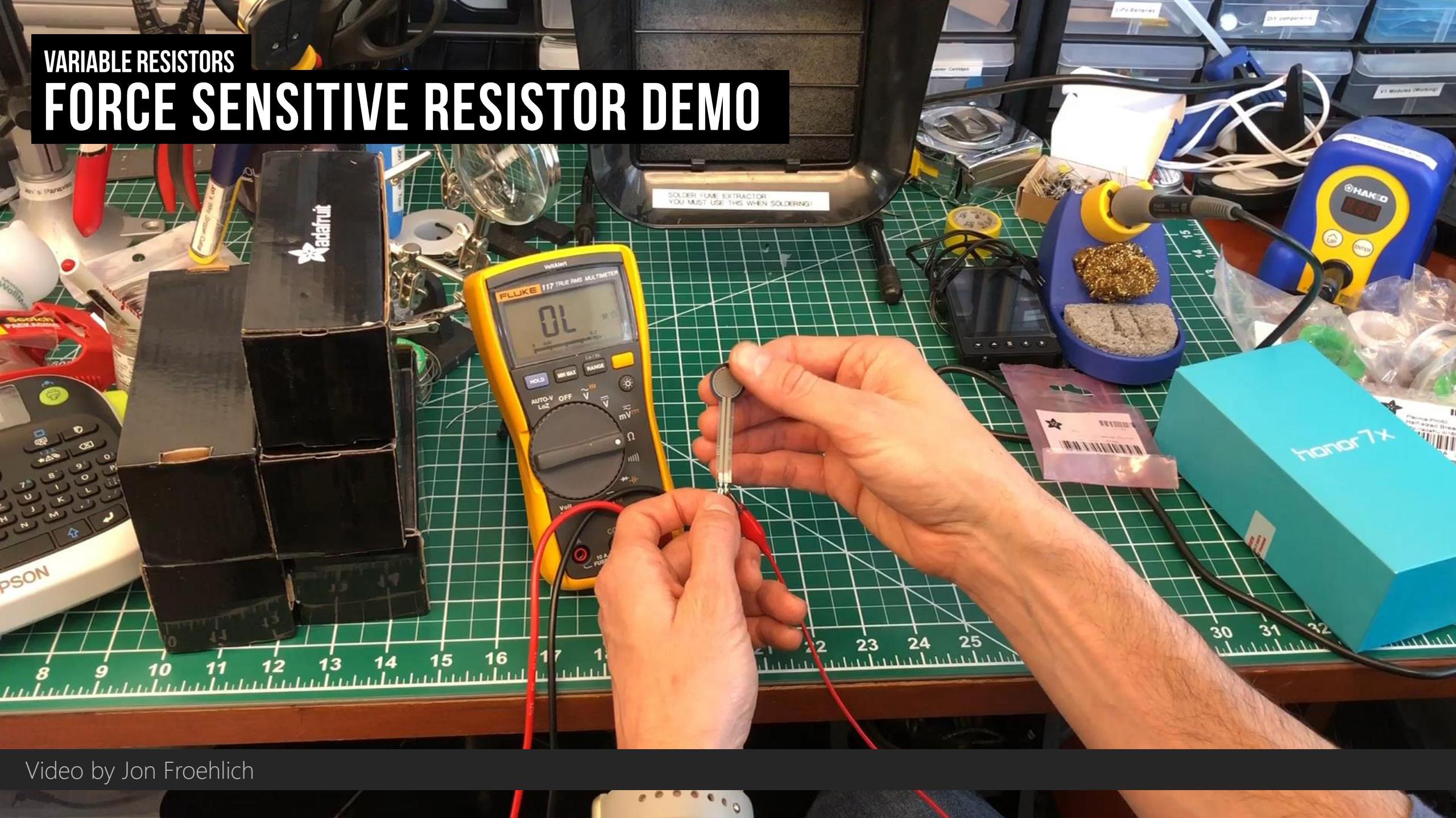
Use Serial Plotter to figure out good min and max values for map

FORCE SENSITIVE RESISTORS (FSRS)

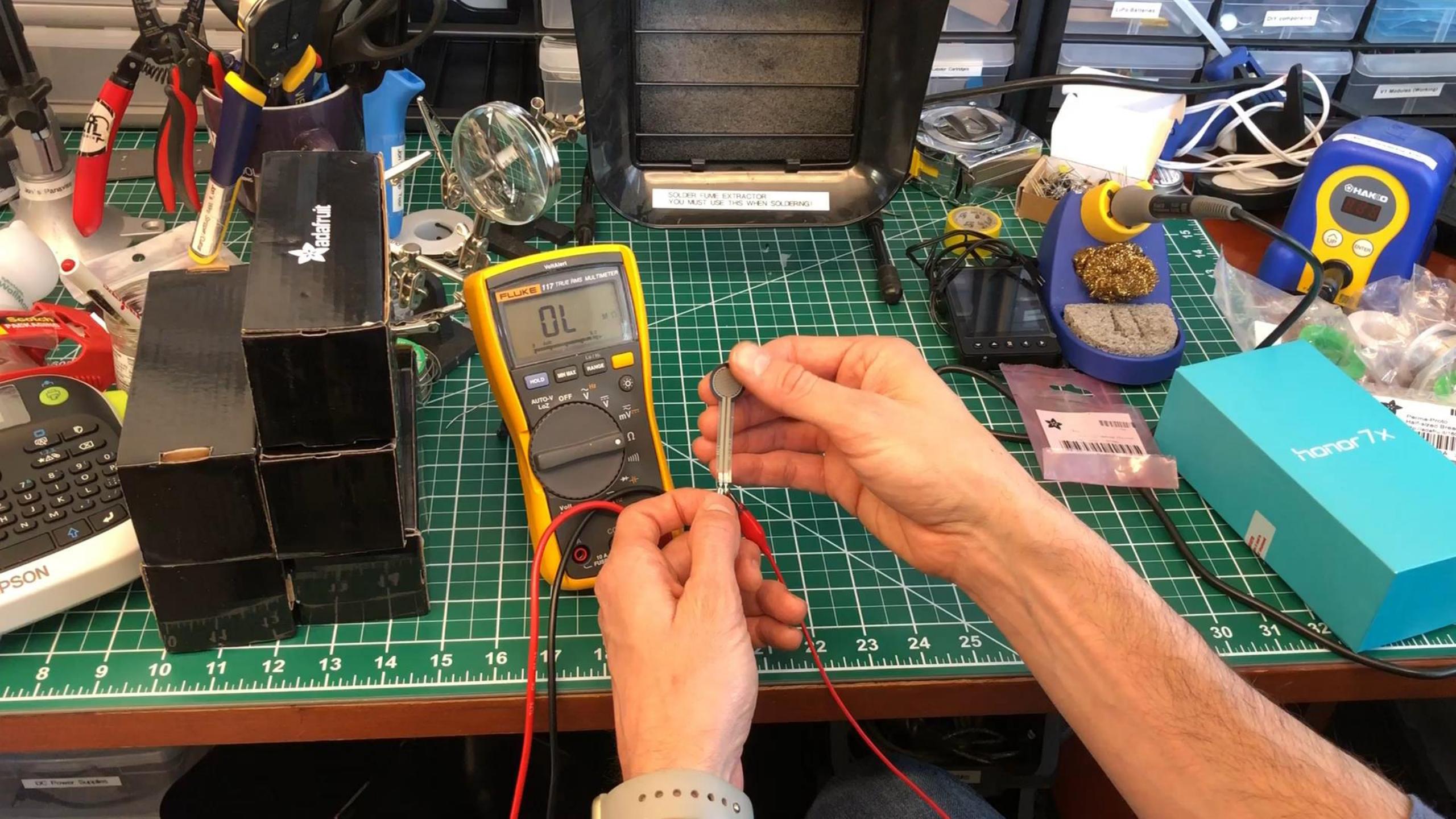


VARIABLE RESISTORS

FORCE SENSITIVE RESISTOR DEMO

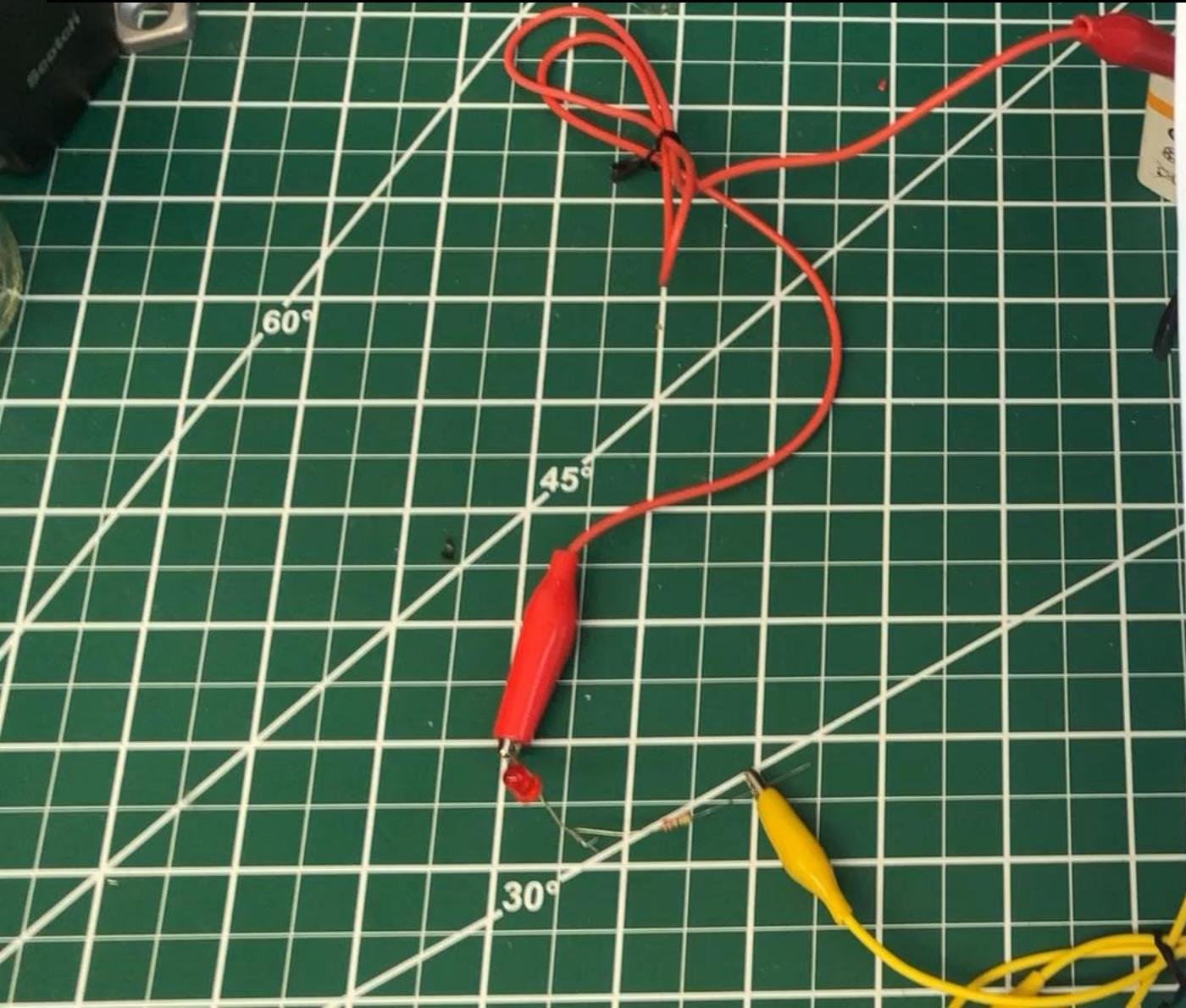


Video by Jon Froehlich



VARIABLE RESISTORS

PAPER SWITCH / PRESSURE SENSOR



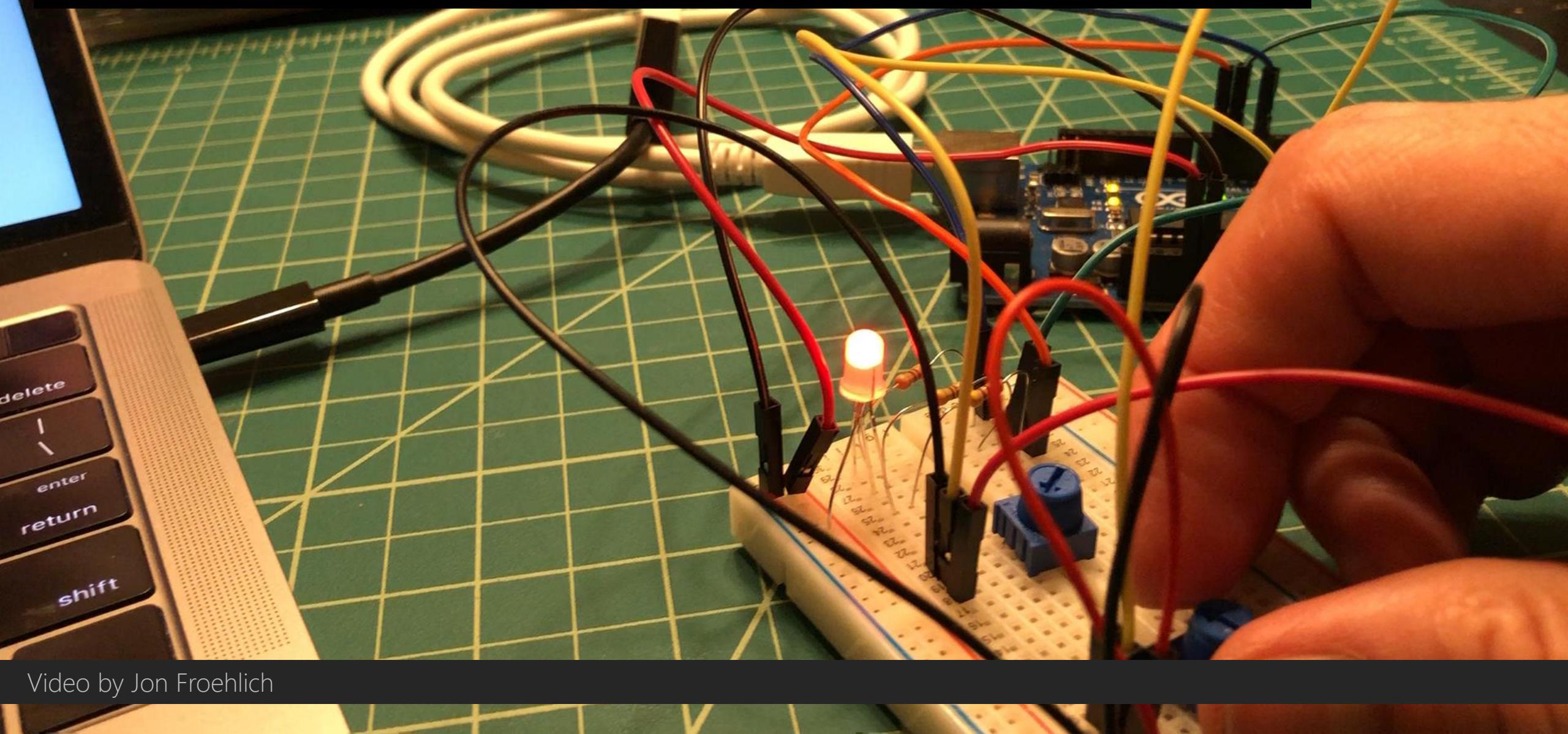
ACTIVITY: FORCE SENSITIVE RESISTOR

Replace your photocell with an FSR

Experiment!

ANALOG INPUT

CHANGING RGB COLOR & BRIGHTNESS WITH TWO POTS



Video by Jon Froehlich

TODAY'S LEARNING GOALS

Working with **analog input**

What is **a knob/trim potentiometer** and how to use it

What is **a slide potentiometer** and how to use it

What are **variable resistive sensors** and how to use them

Understanding the **importance of voltage dividers** when working with analog input (and the relevancy of Ohm's law!)

PHYSCOMP 4: ANALOG INPUT

CSE 599 Prototyping Interactive Systems | Lecture 5 | Oct 10

Jon Froehlich • Liang He (TA)