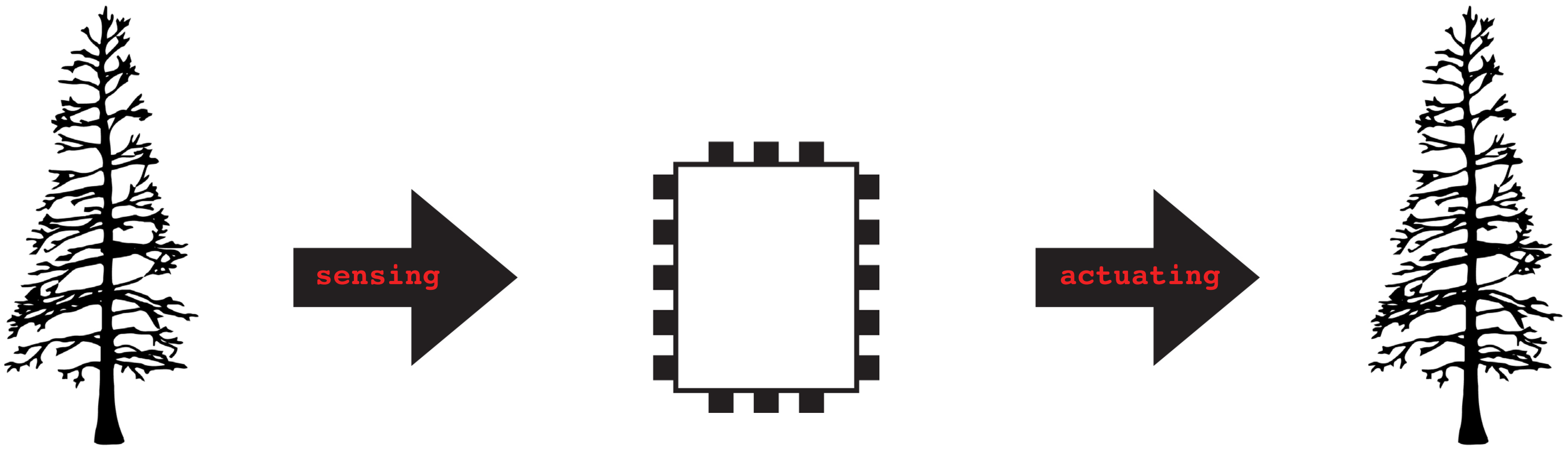
translating

world ---------> <--------------

computer

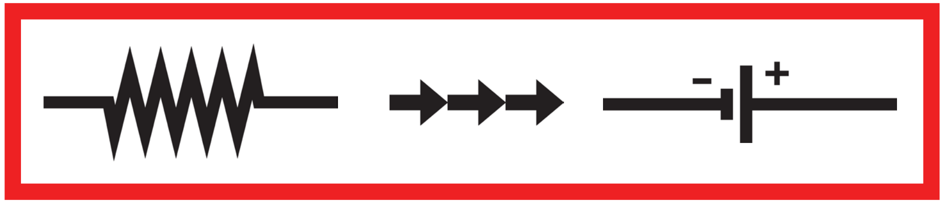
translating between world and computer



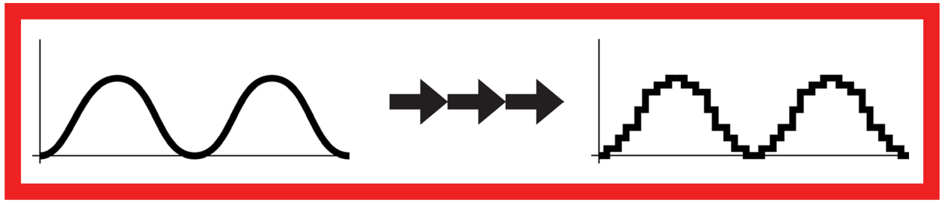
sensing

actuating

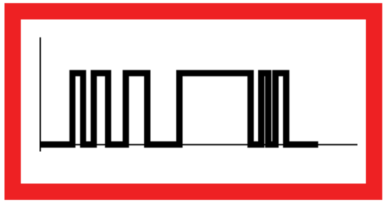
translating resistance to voltage



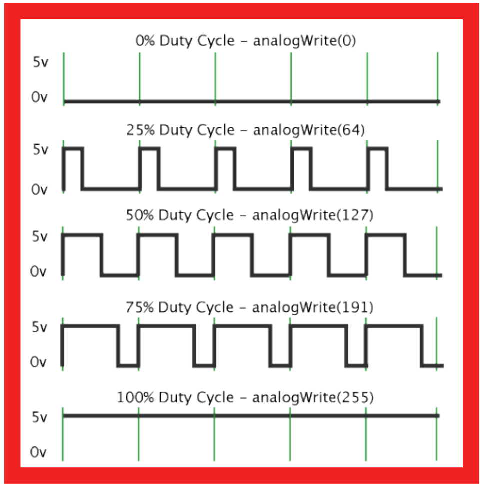
translating analog to digital



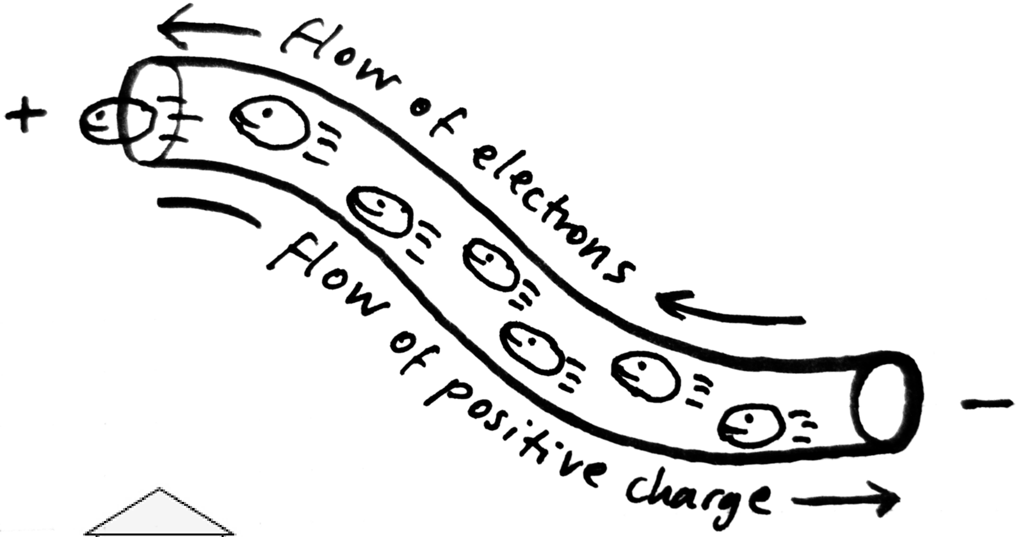
HIGH, LOW

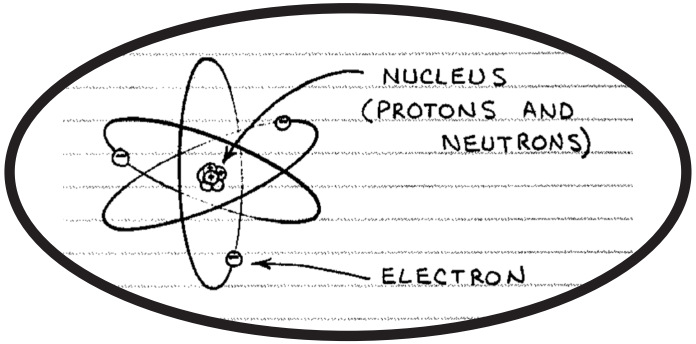


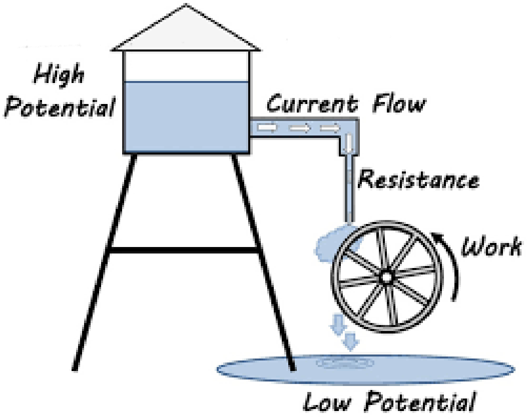
PWM



intro to electricity







Water Analogy

If we compare electricity to water flowing through a pipe, then:
Voltage is the water pressure,
Current is the stream of flow of water,
Resistance is the valve.

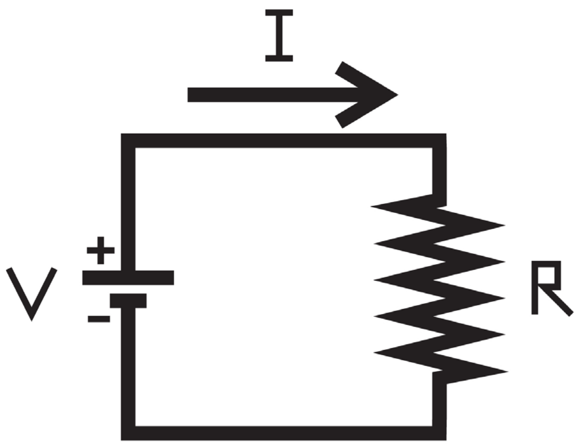
Voltage (V) - is electrical pressure or force. Sometimes referred to as potential. Voltage drop is the difference in voltage between the two ends of a conductor through which current is flowing.

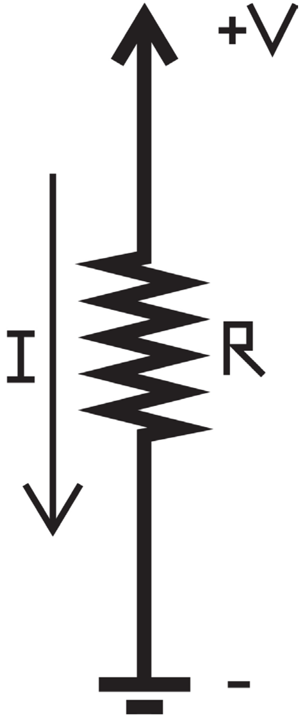
Current (I) - is the quantity of electronics passing a given point.
The unit of current is Ampere.
1 Amp = 6,280,000,000,000,000,000 electronics passing a point in one second.

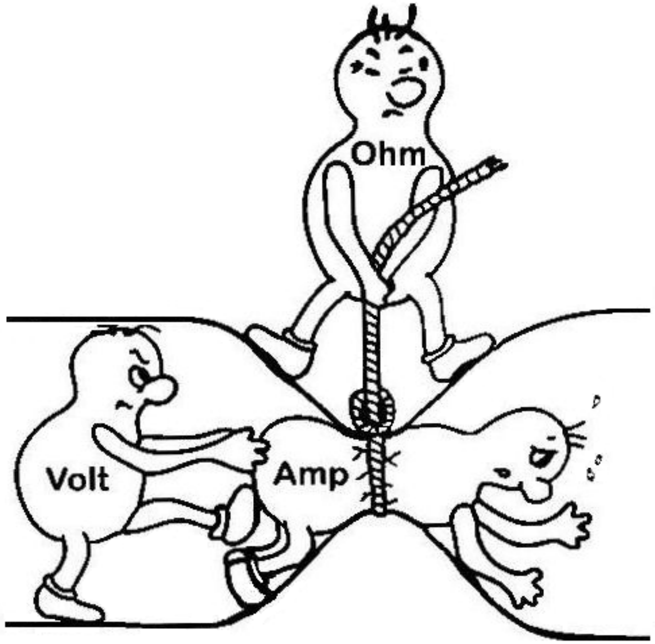
Resistance (R) - conductors are not perfect, they resist the flow of current to some degree.
the unit of resistance is the Ohm (Ω).

book tip: Forrest M. Mims "Getting Started in Electronics"

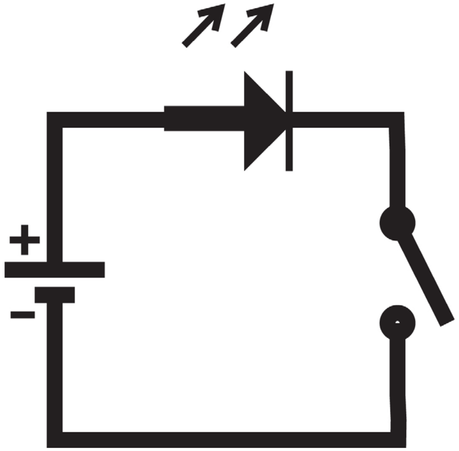
Ohm's Law: $V = I \times R$

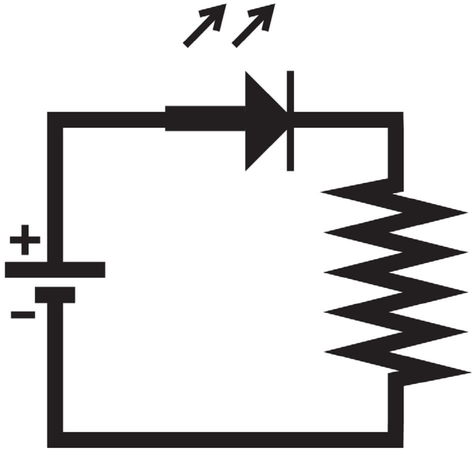


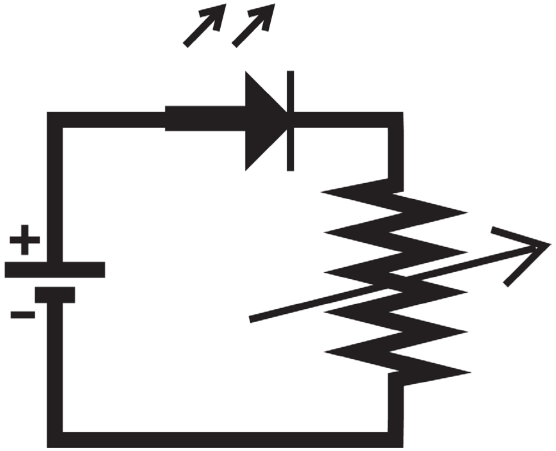




simple LED circuit

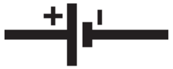




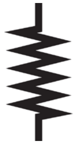


schematic symbols:

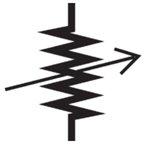
power supply



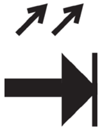
resistor



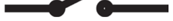
variable resistor



LED



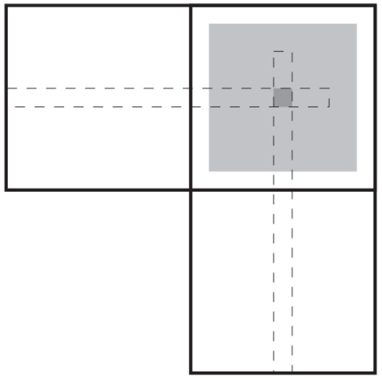
switch



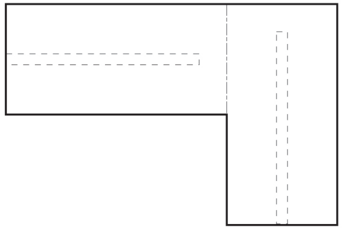
swatch: velostat pressure sensor

analog sensor

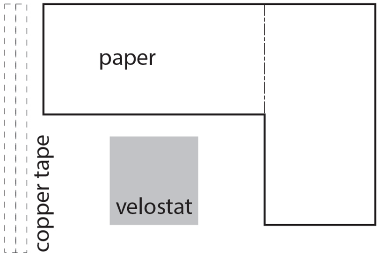
constructed from layering a piece of Velostat between two pieces of copper tape on paper and folded in half.



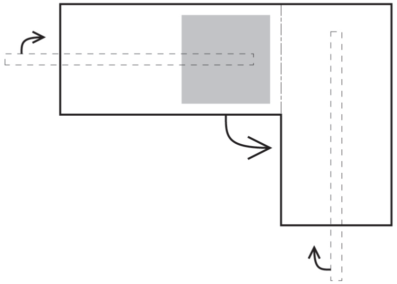
TEMPLATE



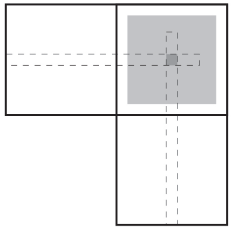
MATERIALS

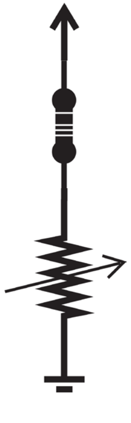


ASSEMBLE



FOLD





Resistor: __________ Ohm

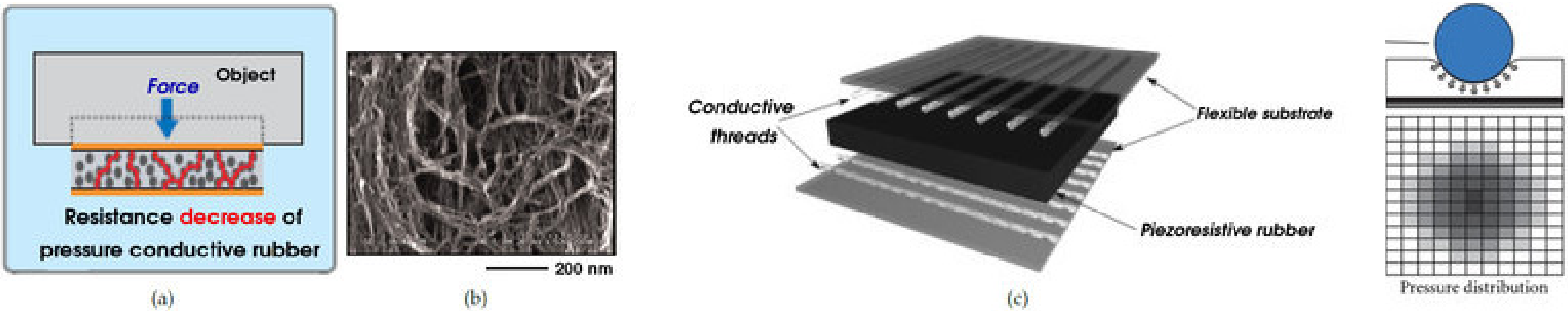
light press: ________ Ohm

hard press: _________ Ohm

Velostat

Piezoresistance

*“**Piezo**”, derived from the Greek piezein, which means to squeeze or press*

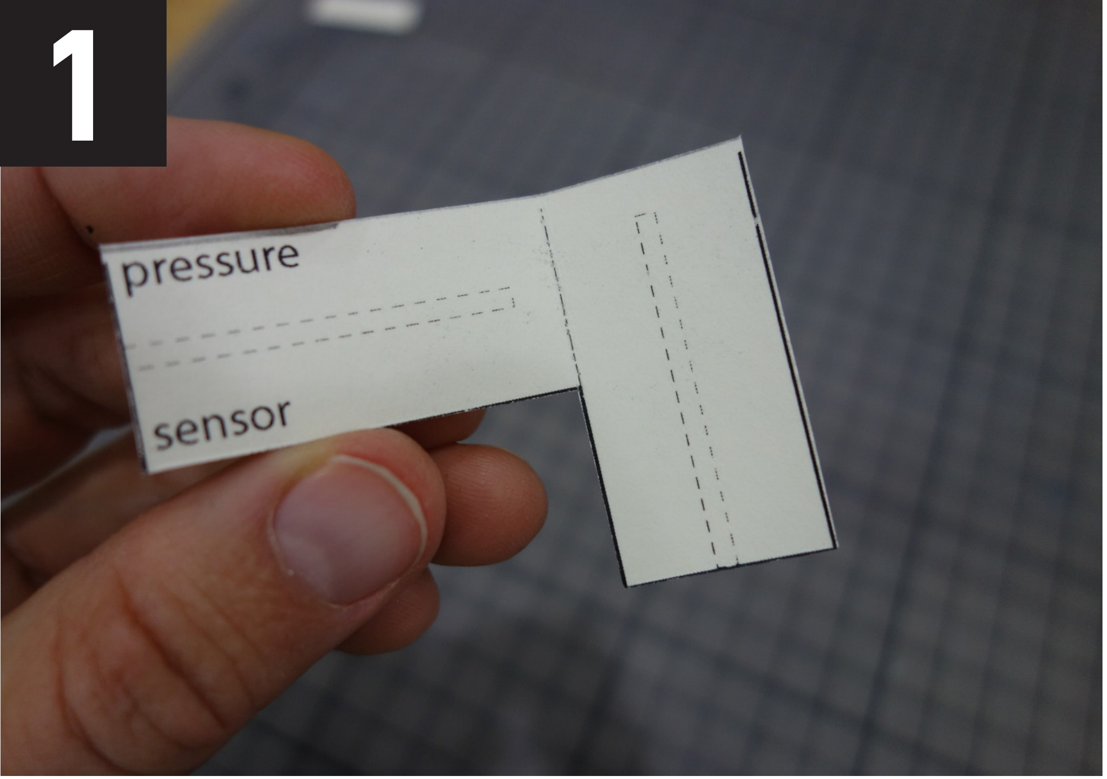


(a) (b) (c)

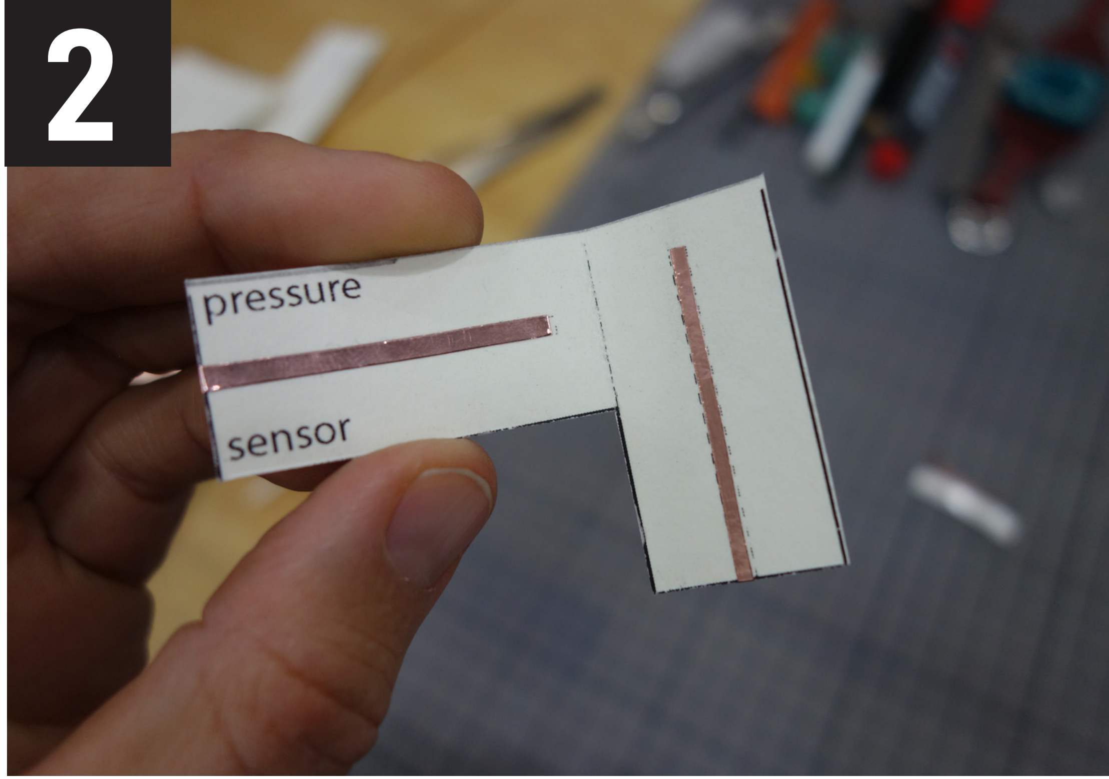
Tactile sensing in dexterous robot hands

https://www.researchgate.net/publication/282557394_Tactile_sensing_in_dexterous_robot_hands_-_Review/figures?lo=1

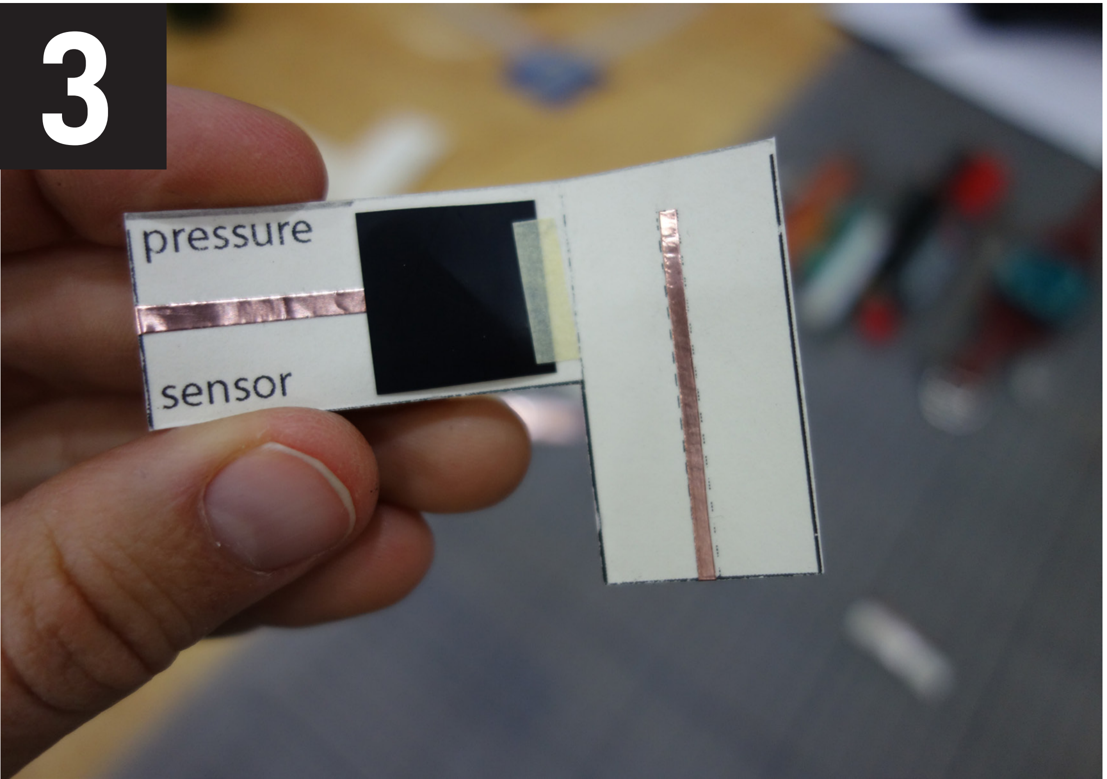
1


pressure
sensor

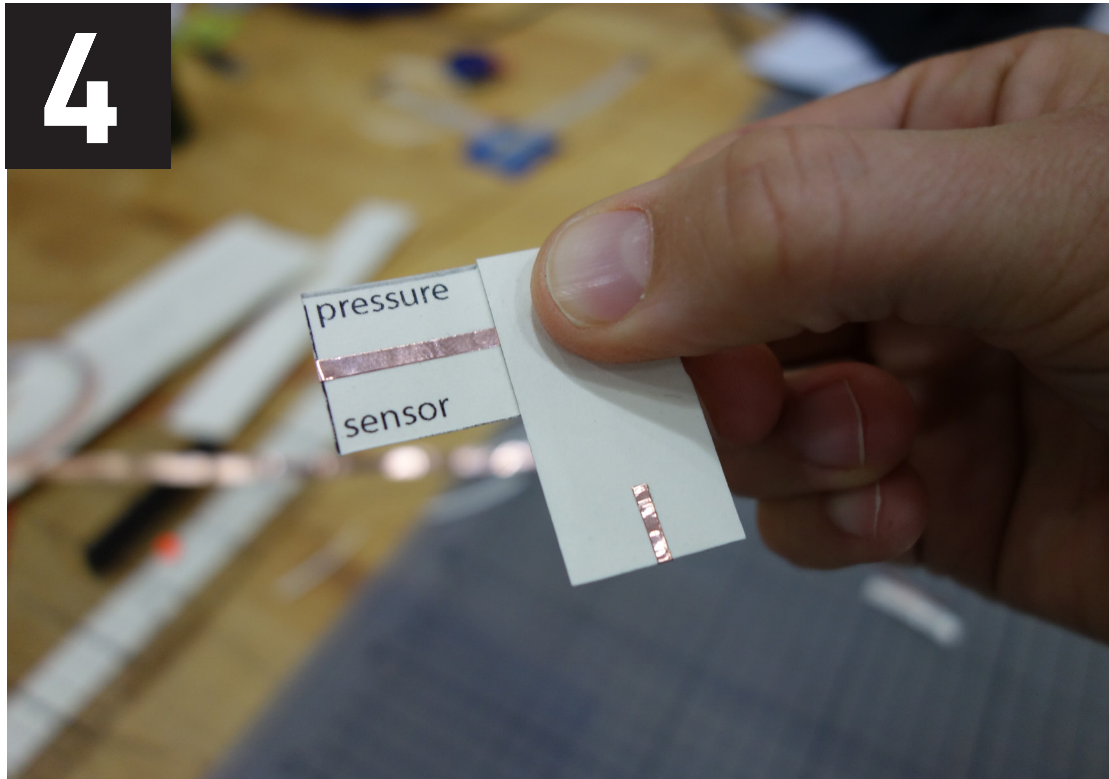
2

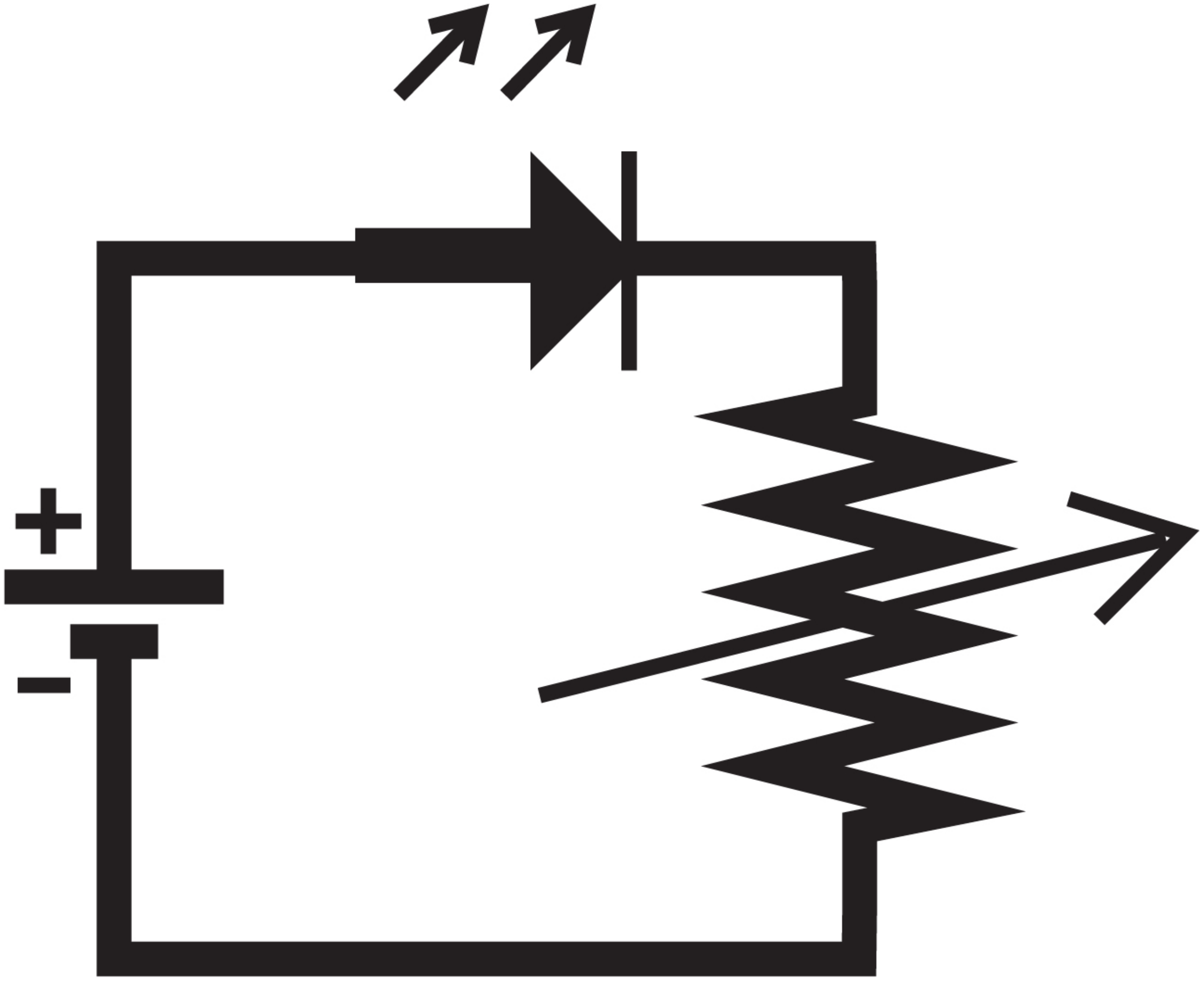

pressure
sensor

3


pressure
sensor

4

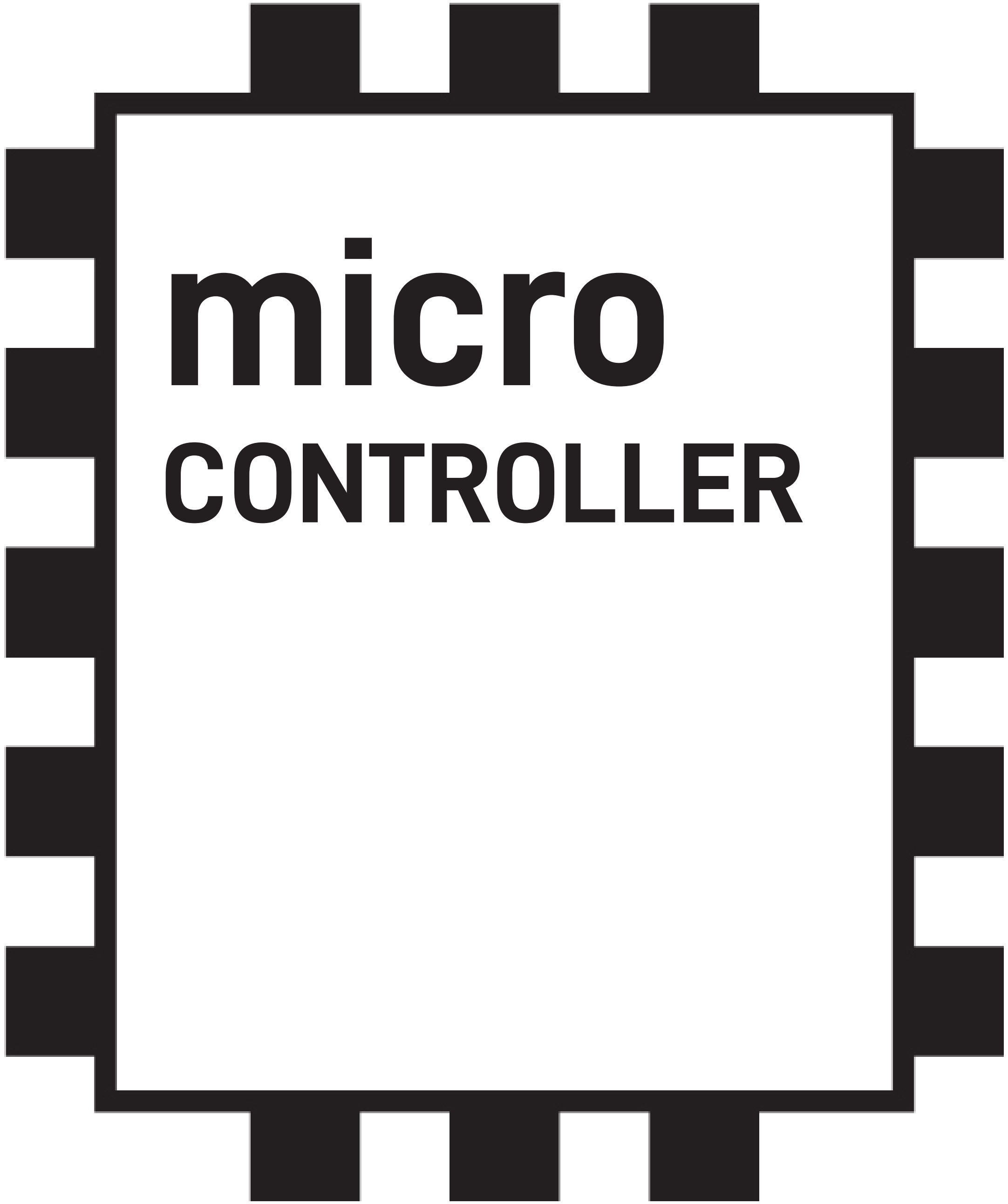

pressure
sensor



but..............what if...... the light

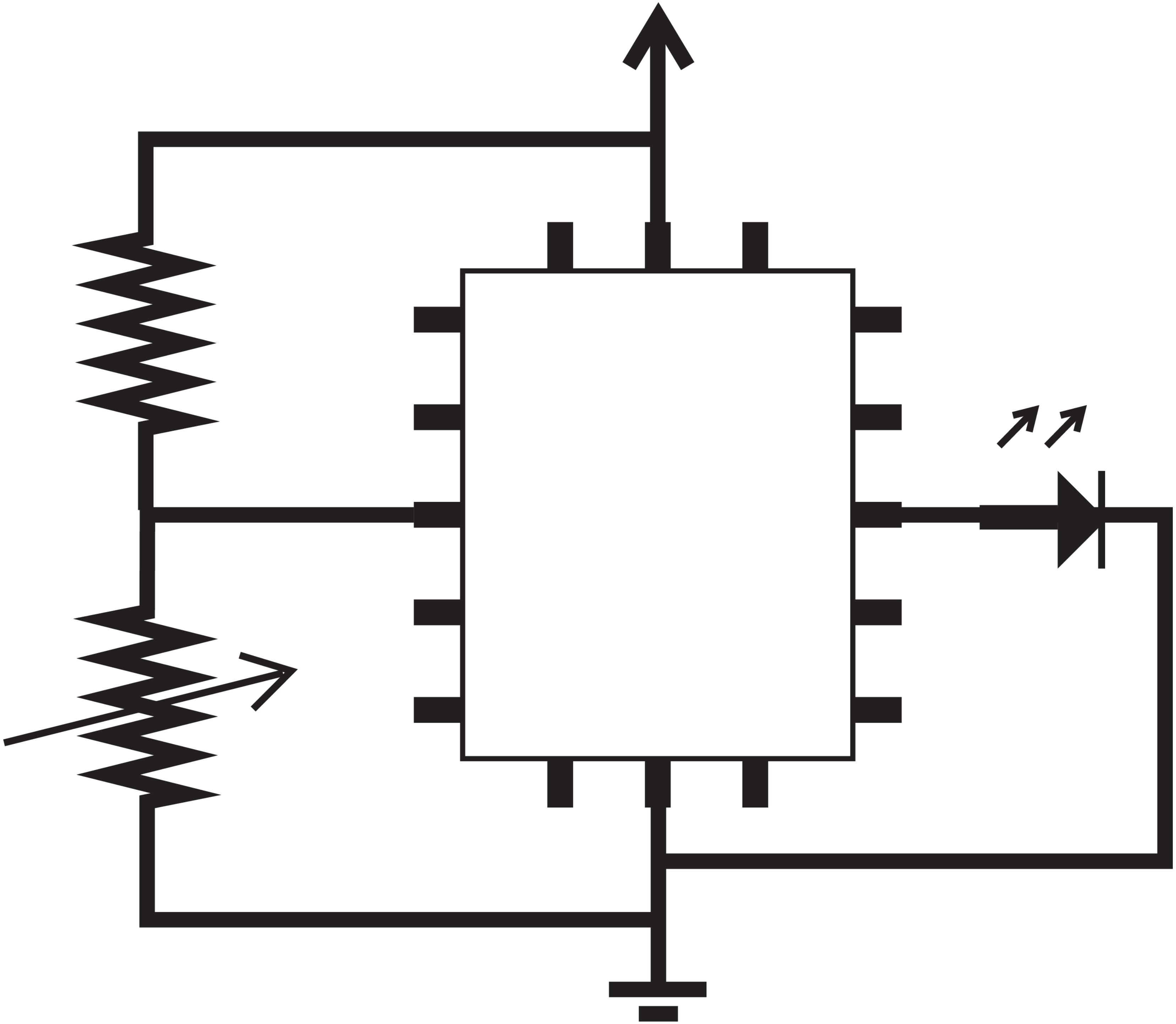
should go off when I

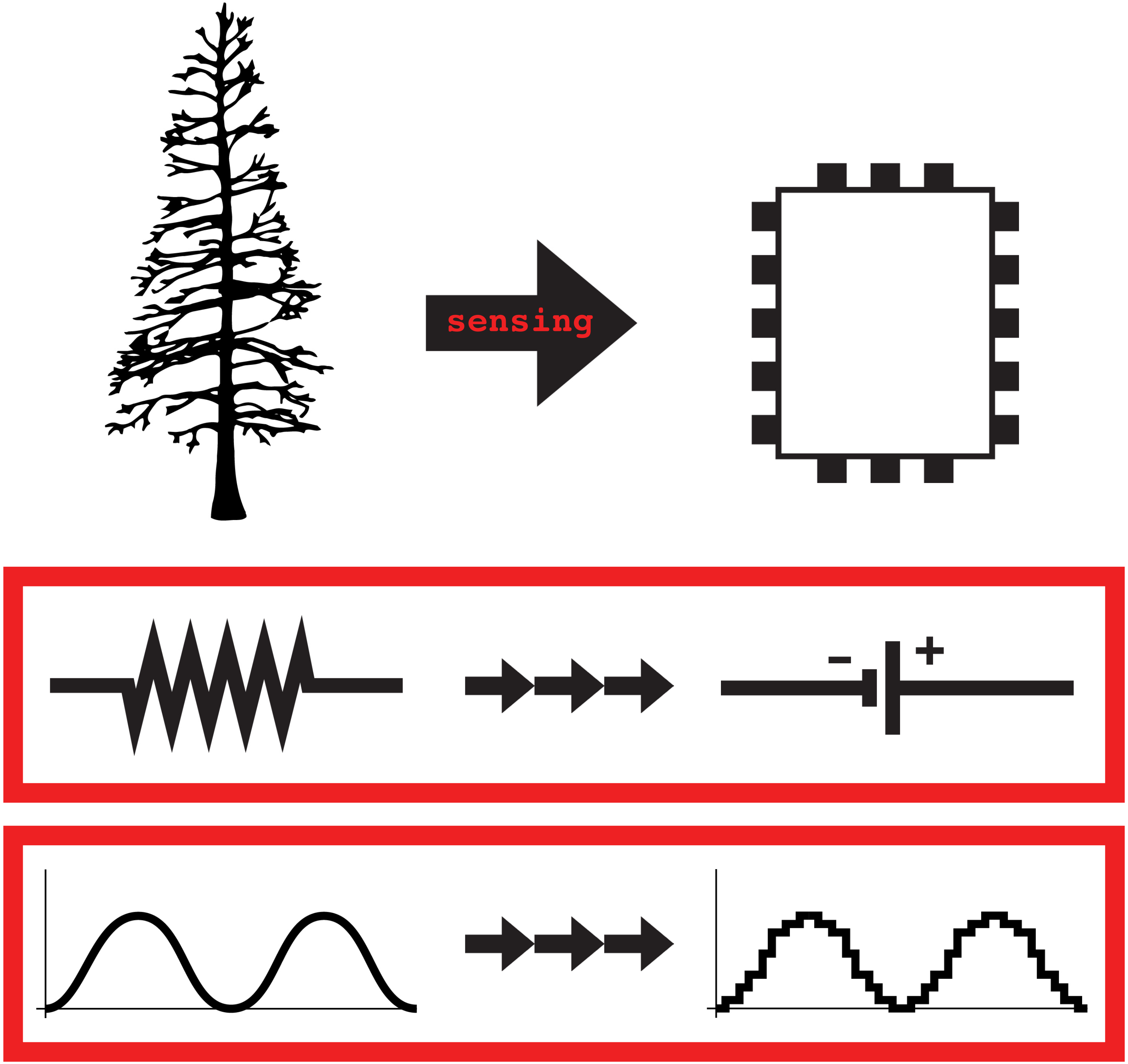
press the sensor ?



micro

CONTROLLER





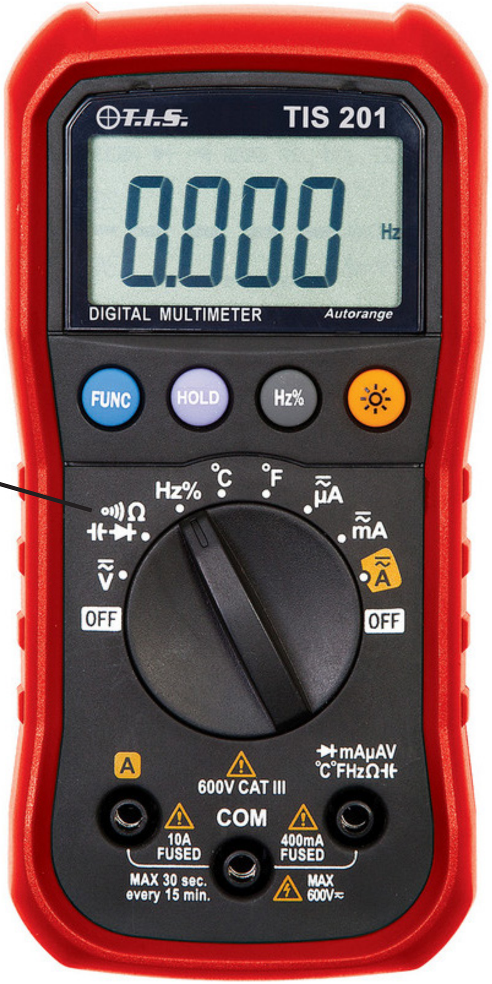
sensing

VOLTAGE
DIVIDER

ANALOG
DIGITAL
CONVERTER
(ADC)

MULTIMETERS

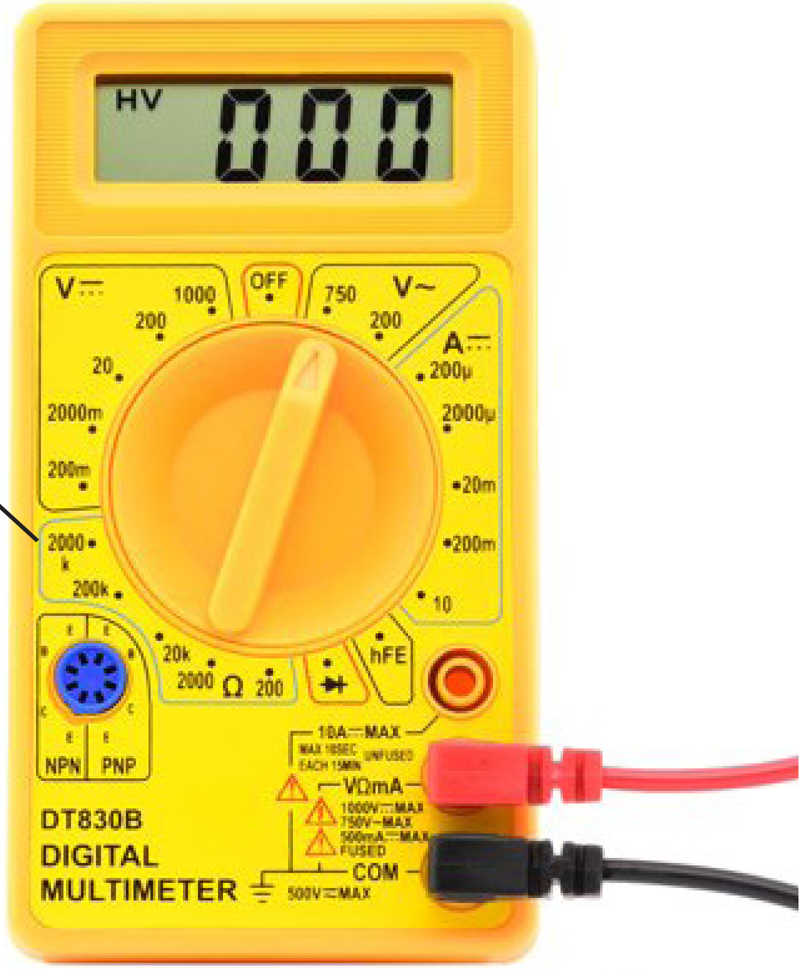
auto-ranging

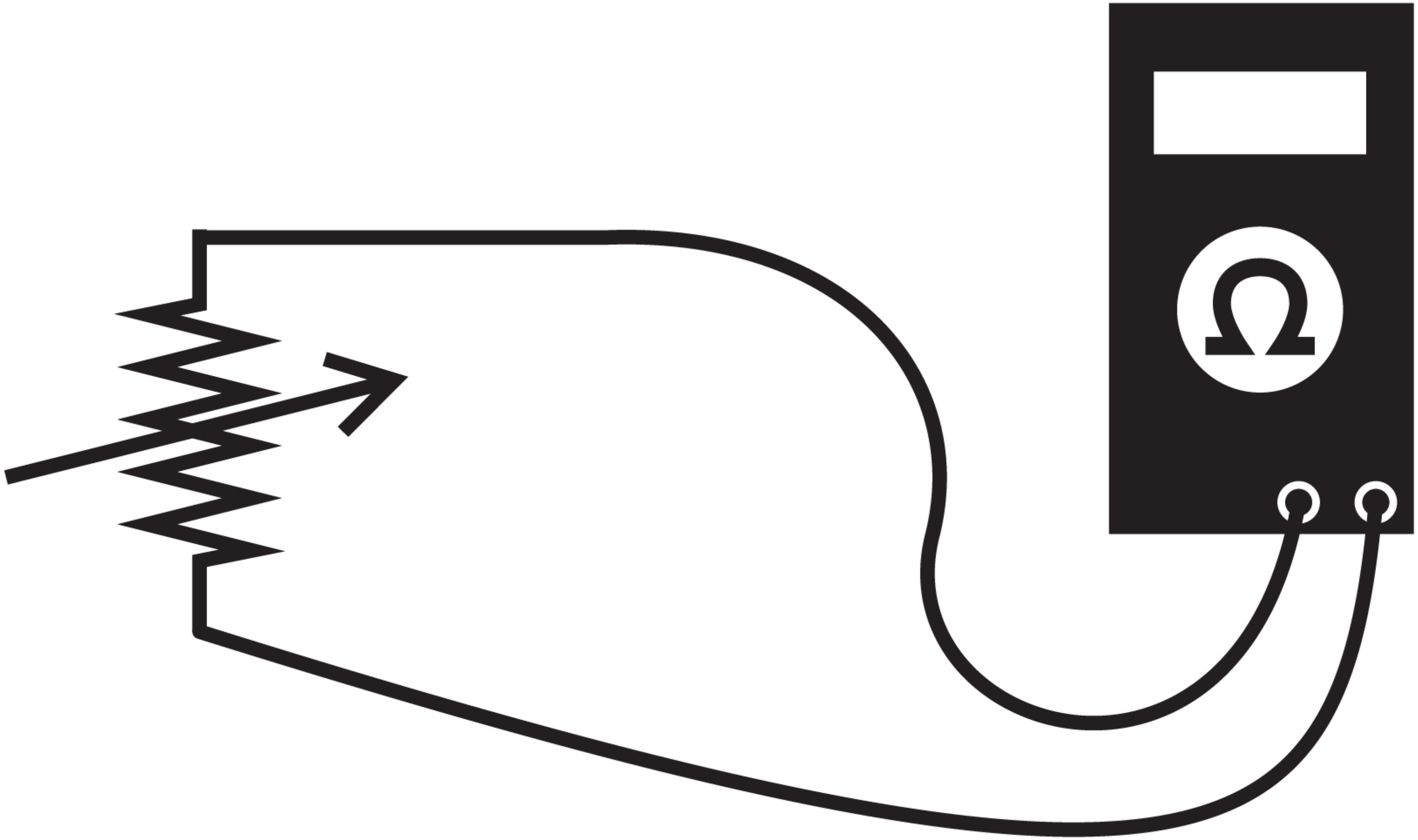


in this mode it will beep when there is a direct connection (<100 ohm)

manual-range

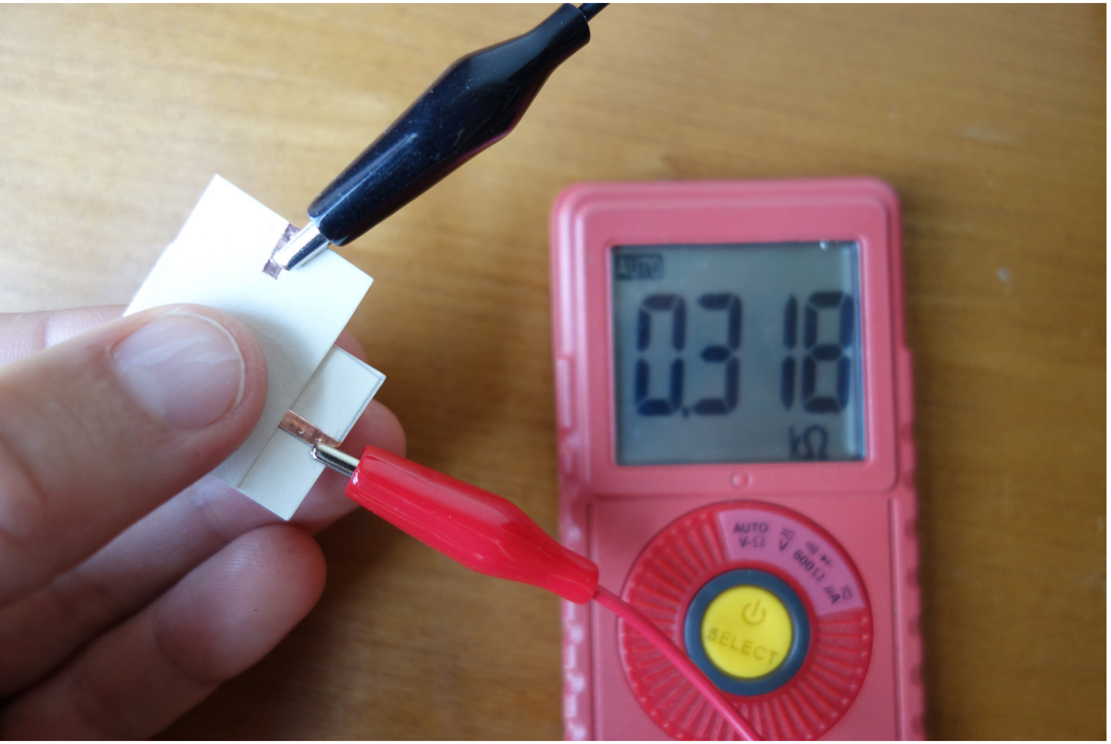
the numbers on the dial are not multipliers, but indicate the maxi-maum reading range.
for example: if the dial is on 20K and the display says "12.4"
then you are reading 12.4K ohm or 12,400 ohm



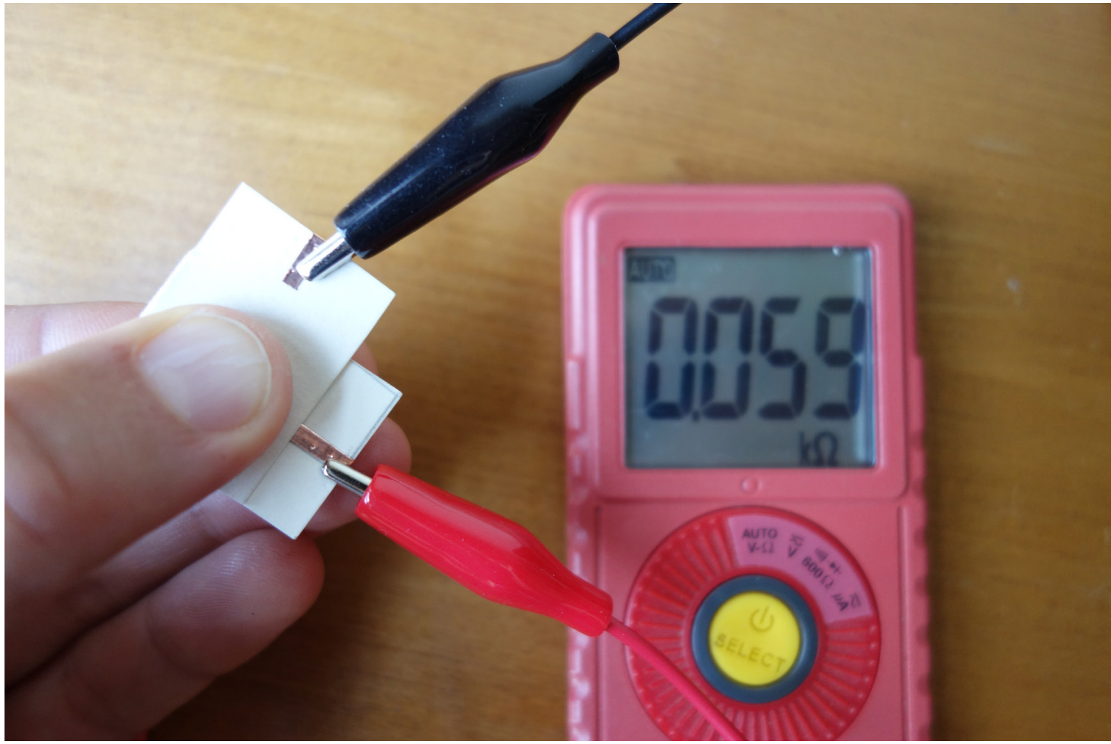


light press: ________ Ohm

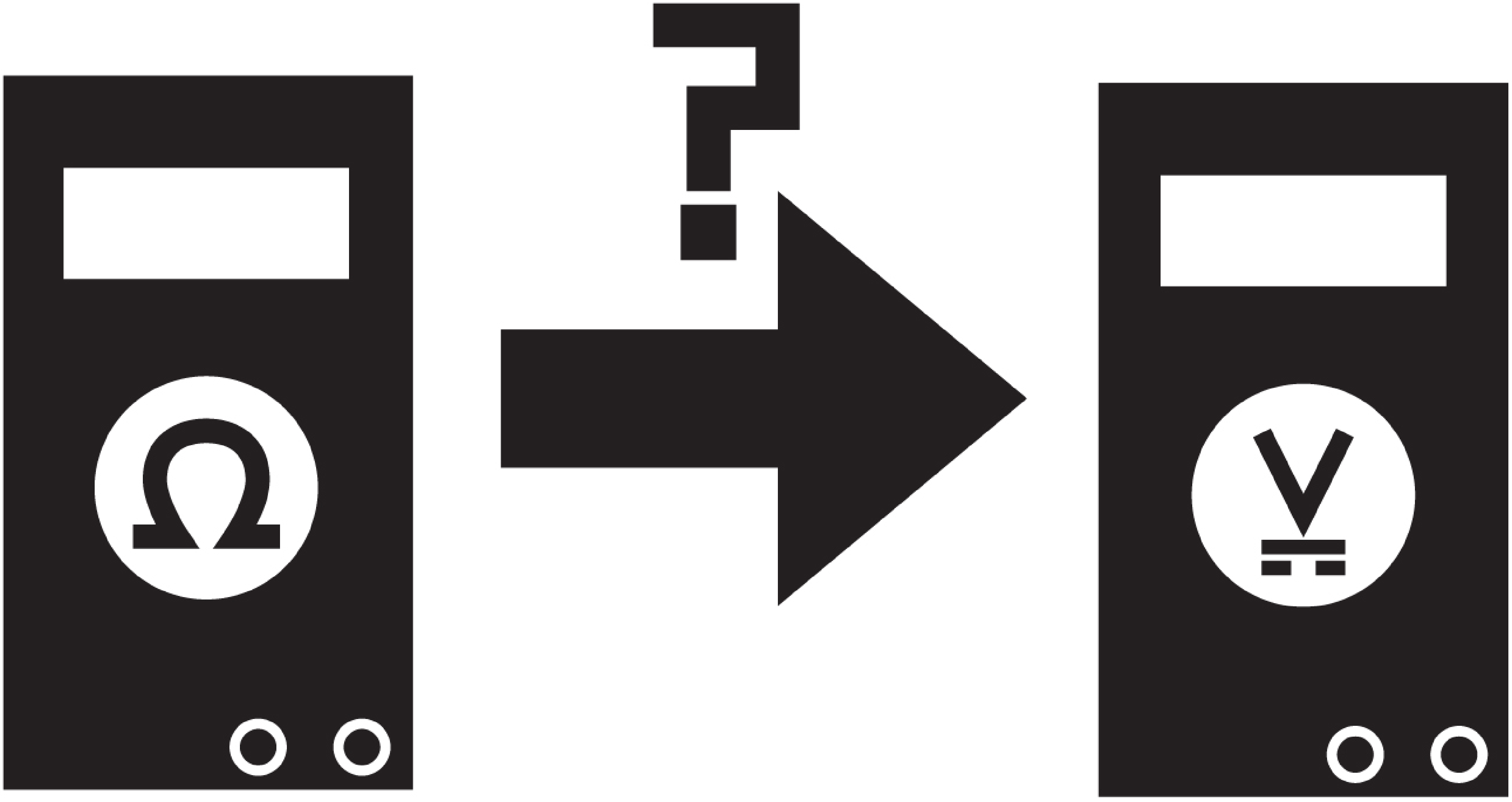
hard press: ________ Ohm

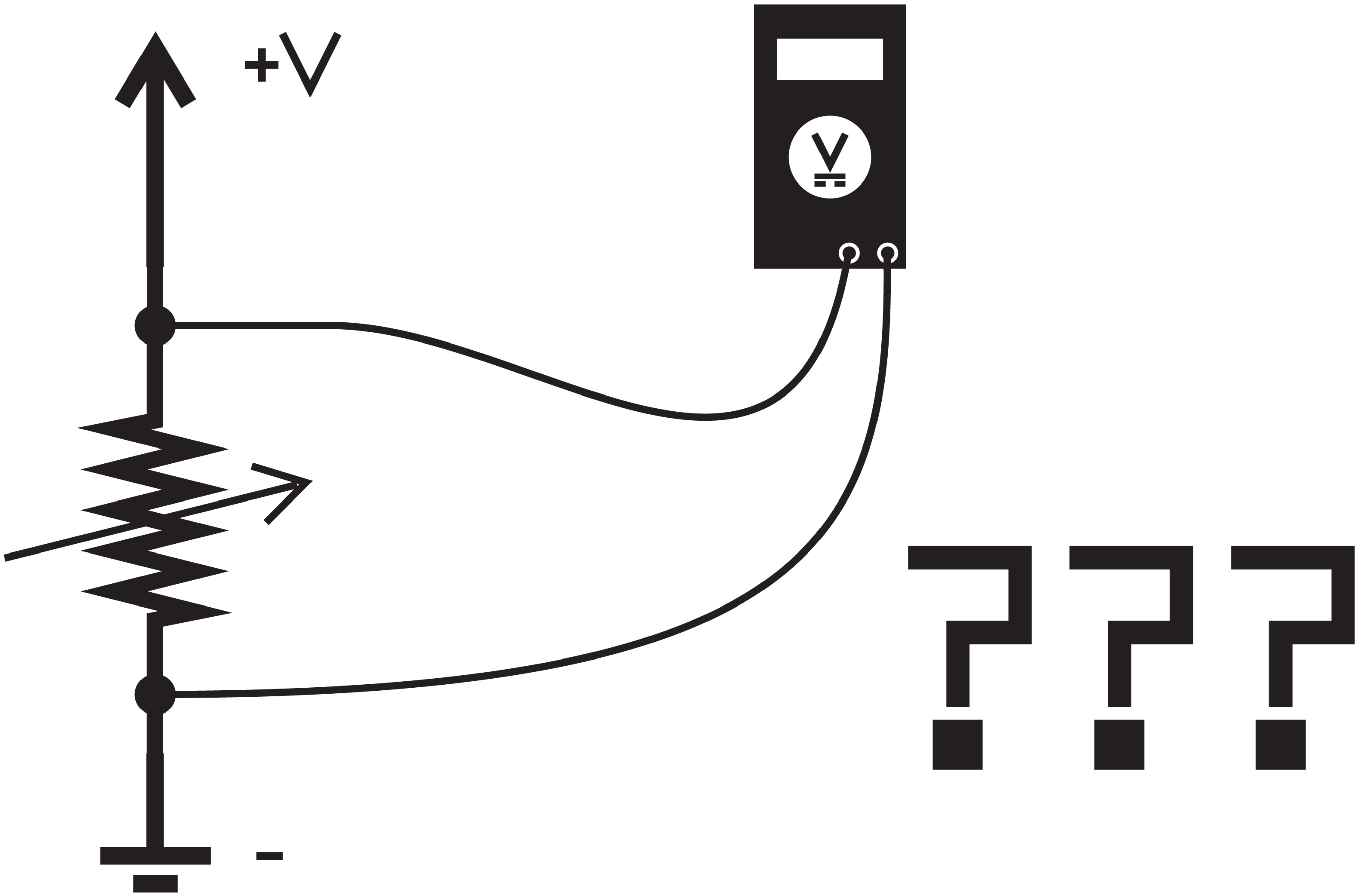


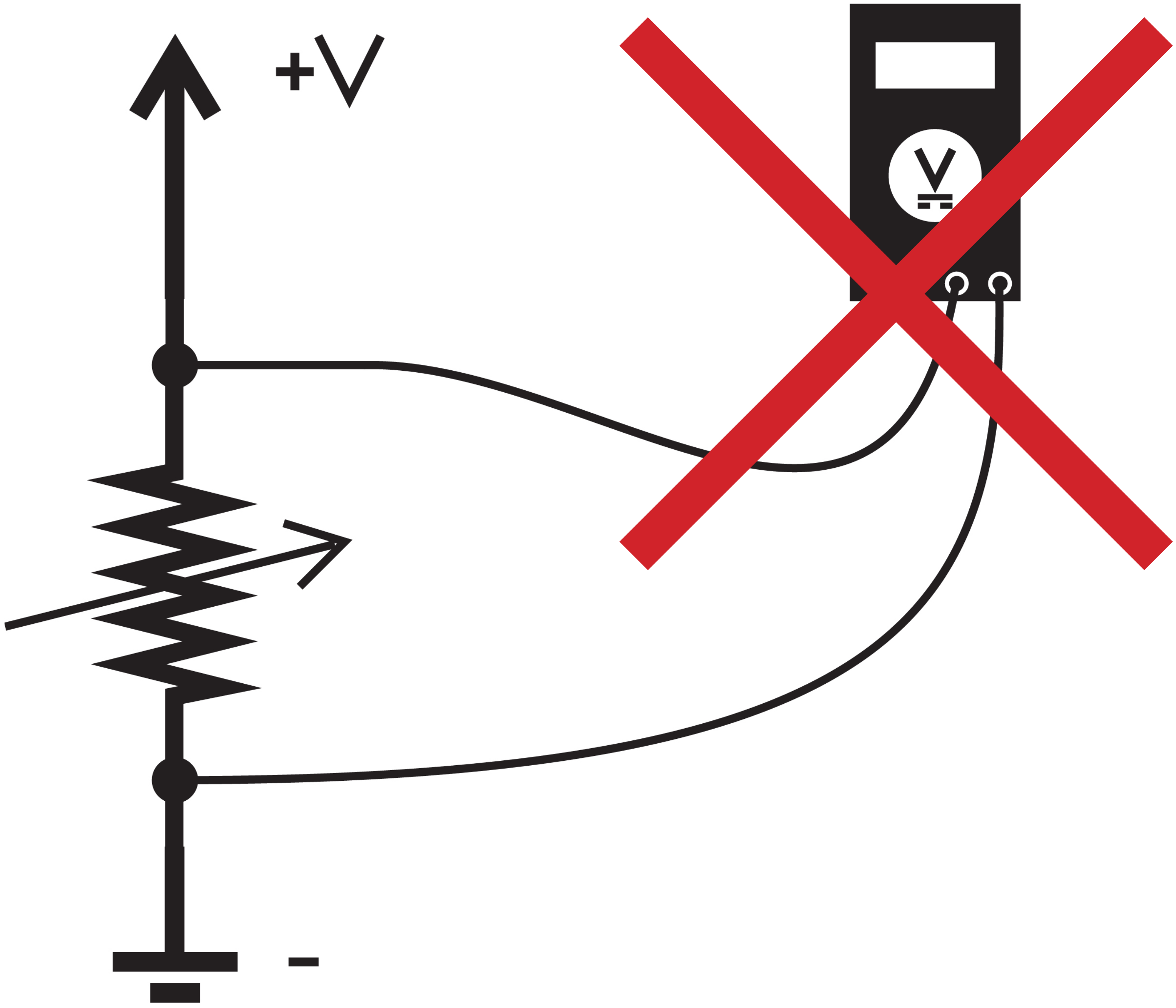
light press: 300 Ohm

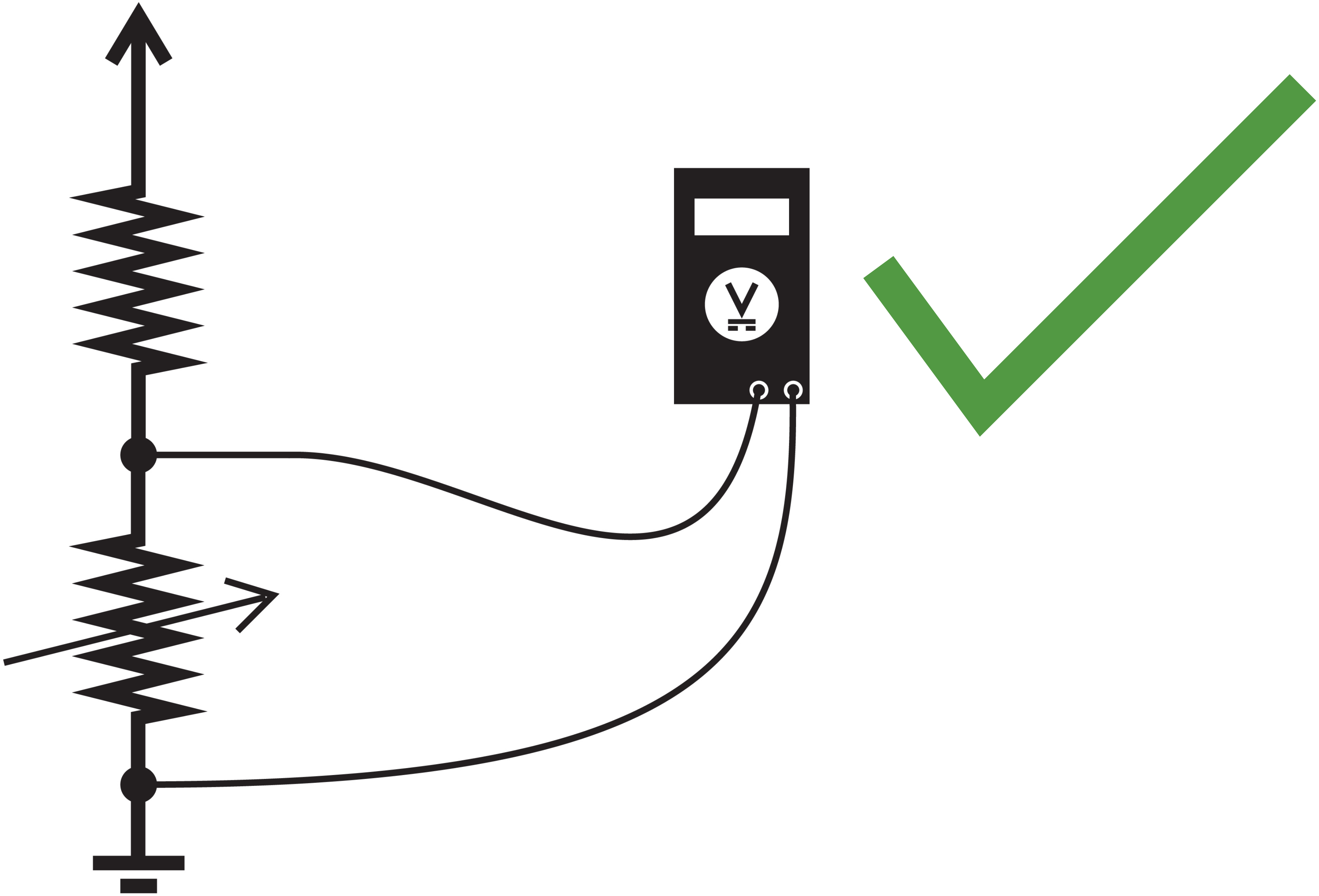


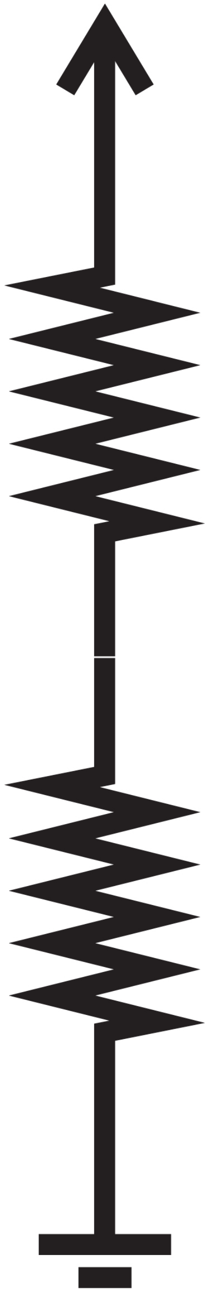
hard press: 50 Ohm

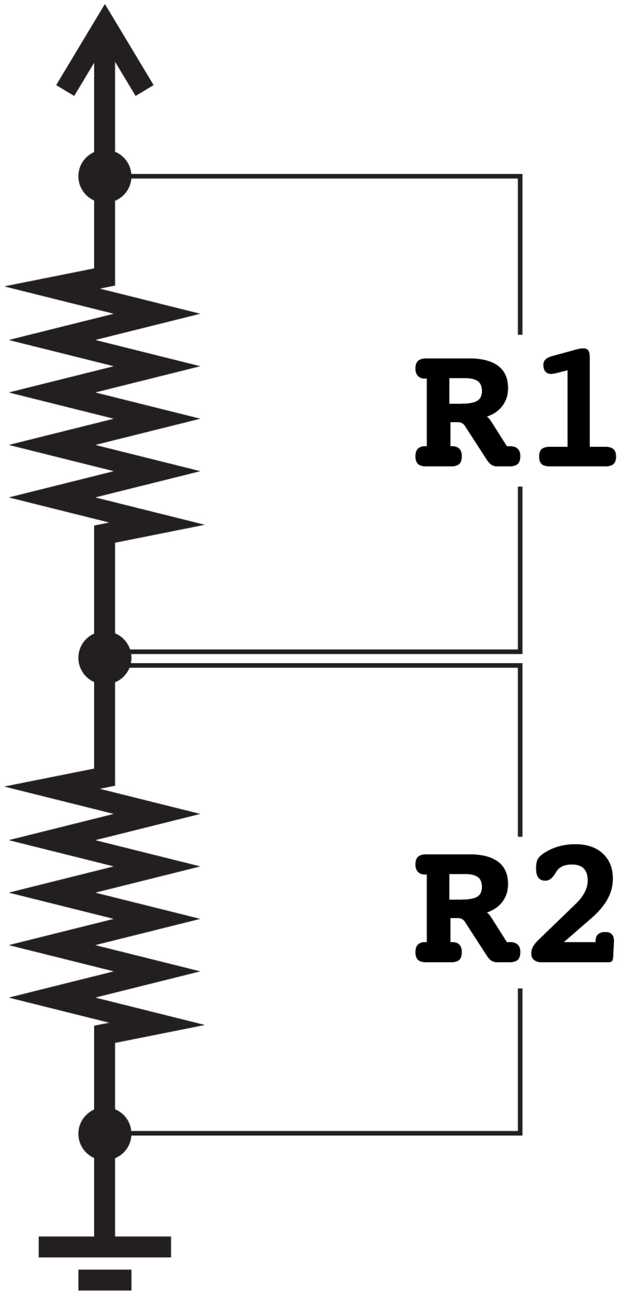


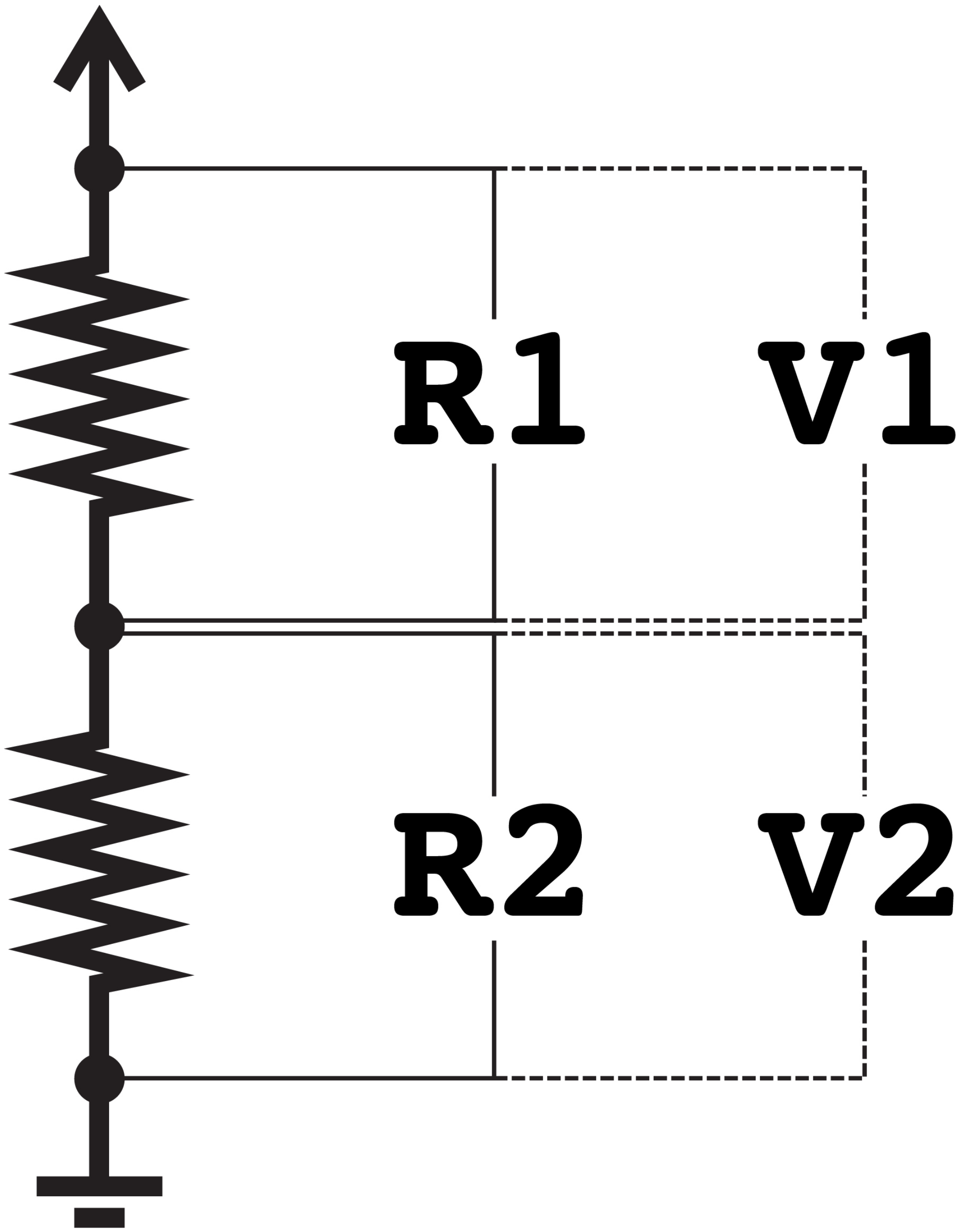

+V
V
-
???


+V
V
-

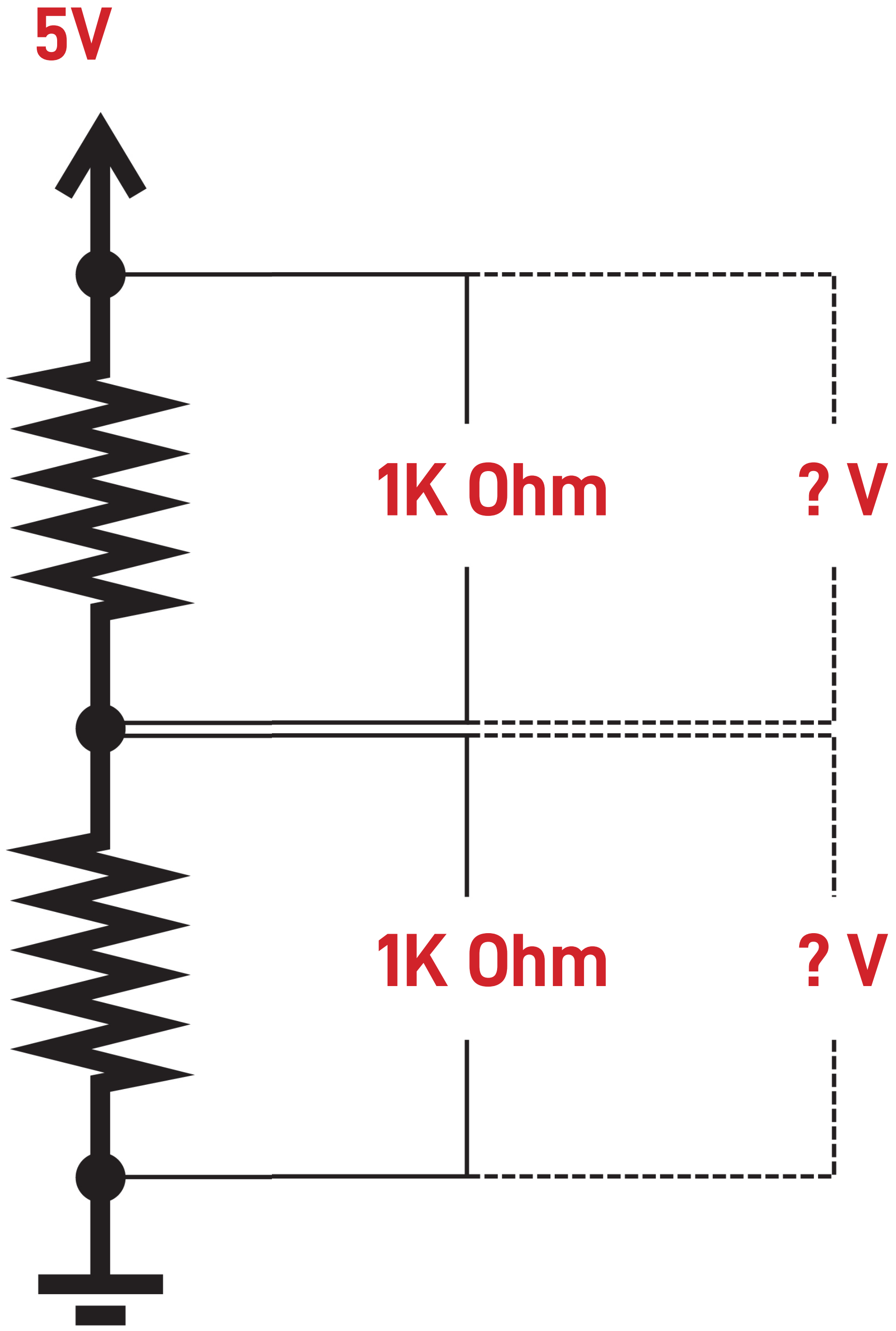

V



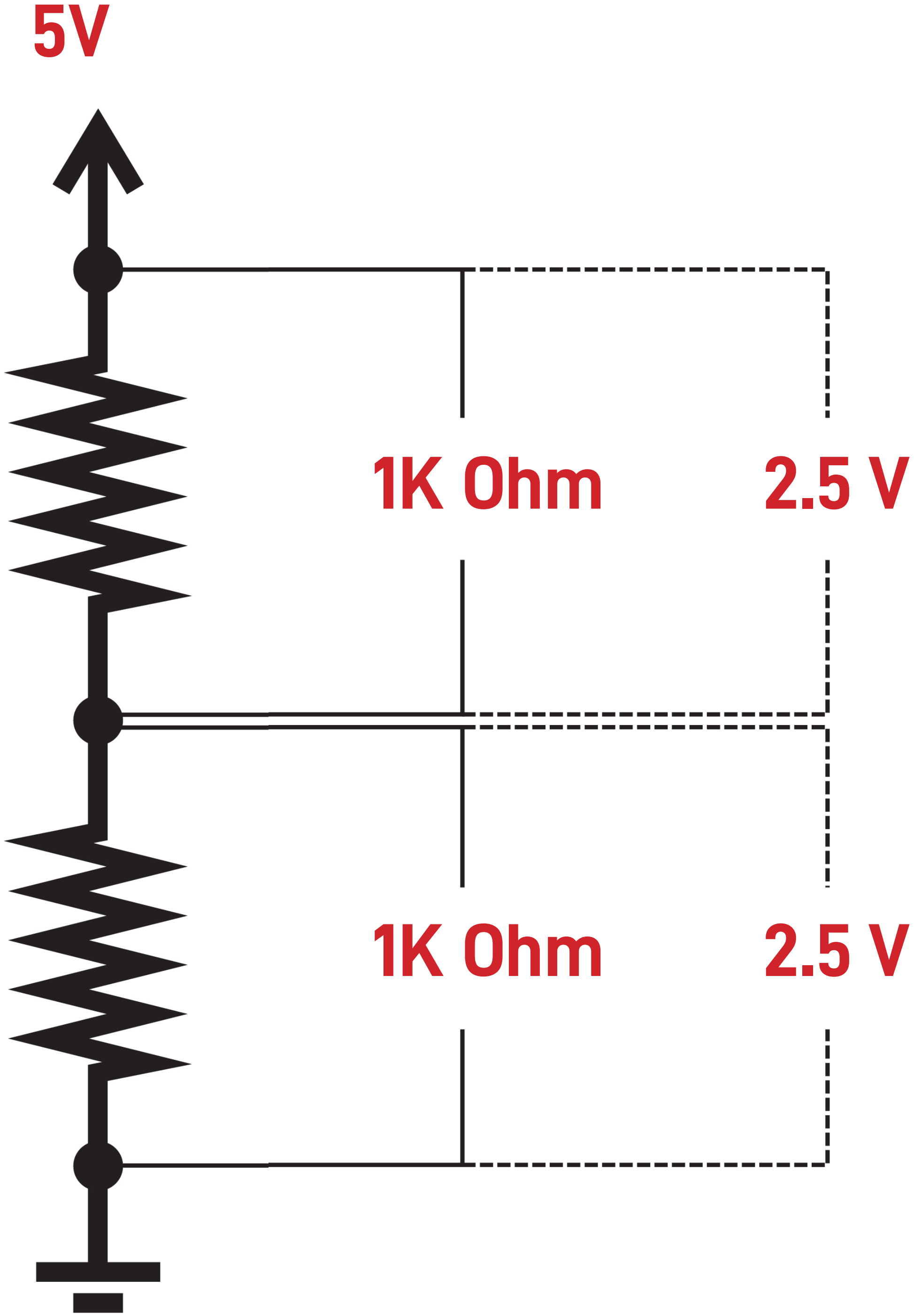

R1
R2



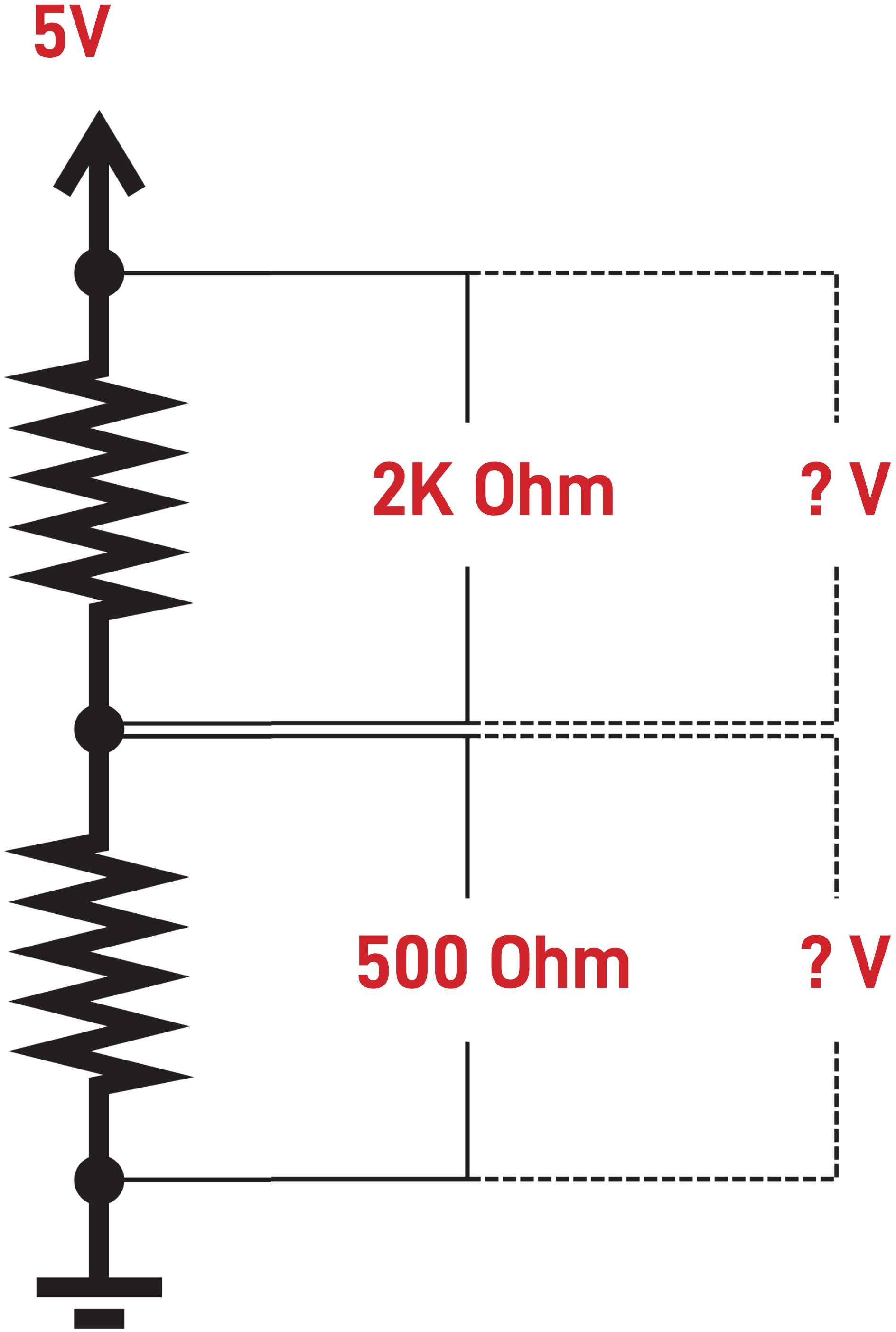
$$\frac{R1}{R2} = \frac{V1}{V2}$$



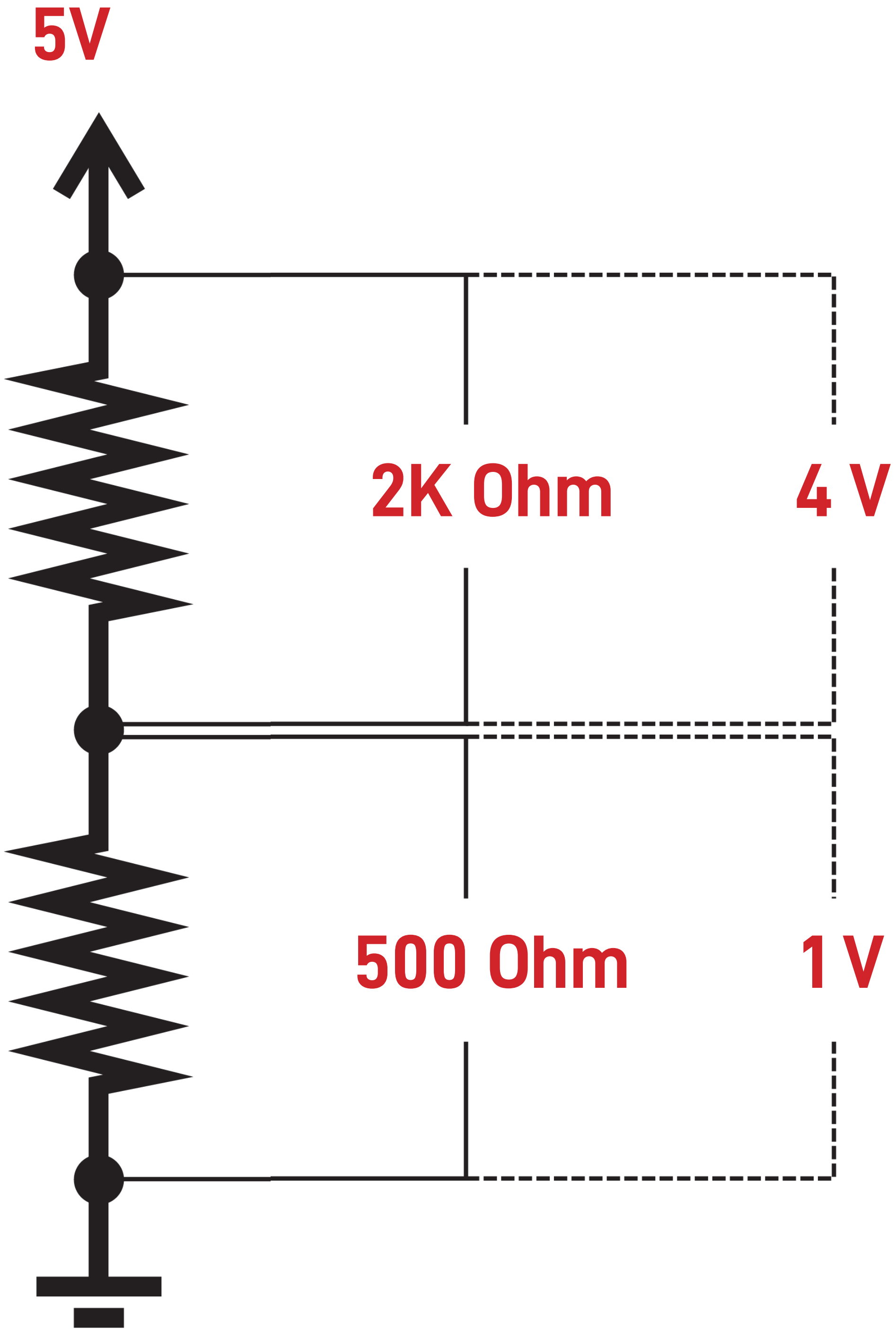
$$\frac{R1}{R2} = \frac{V1}{V2}$$



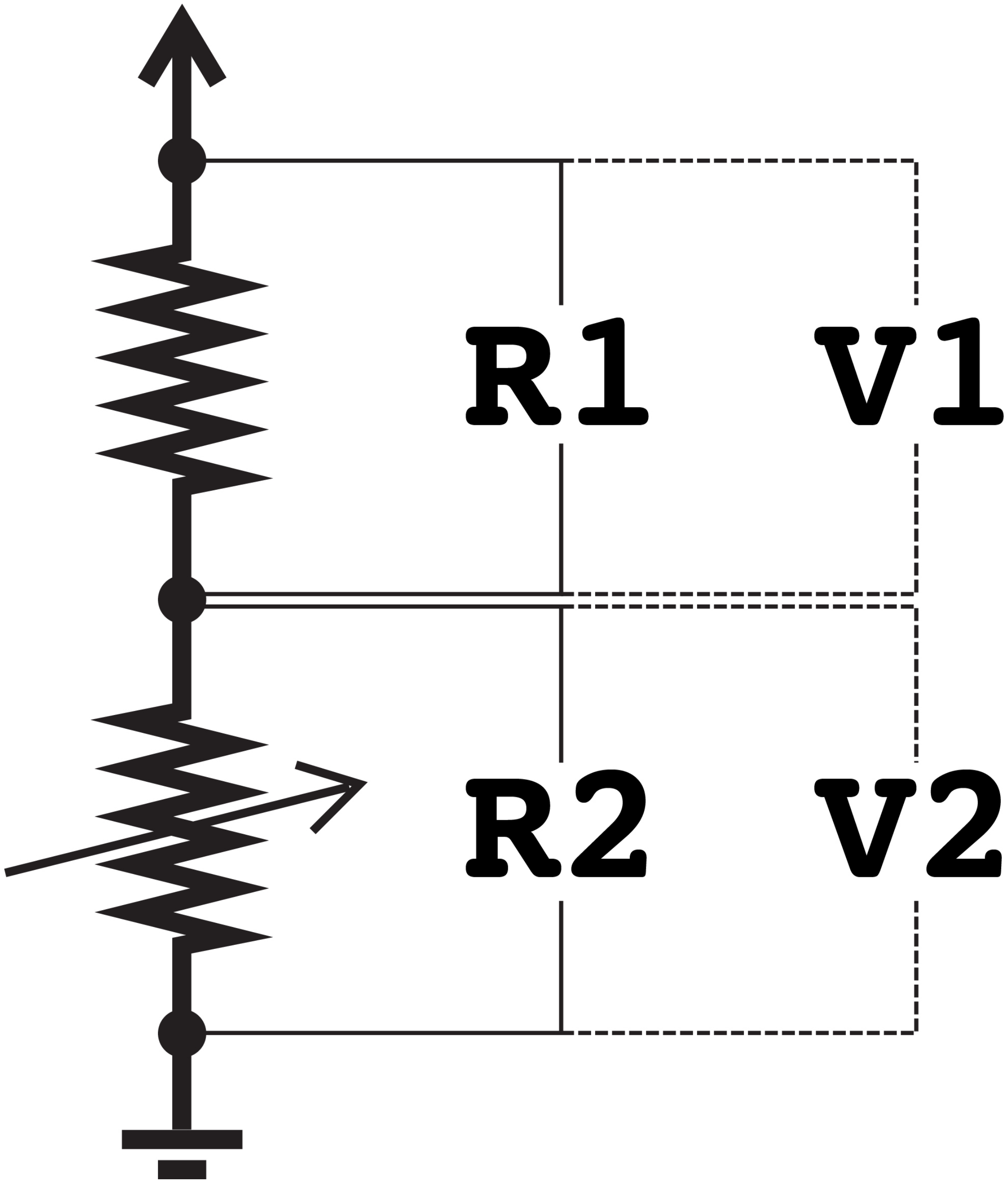
$$\frac{R1}{R2} = \frac{V1}{V2}$$



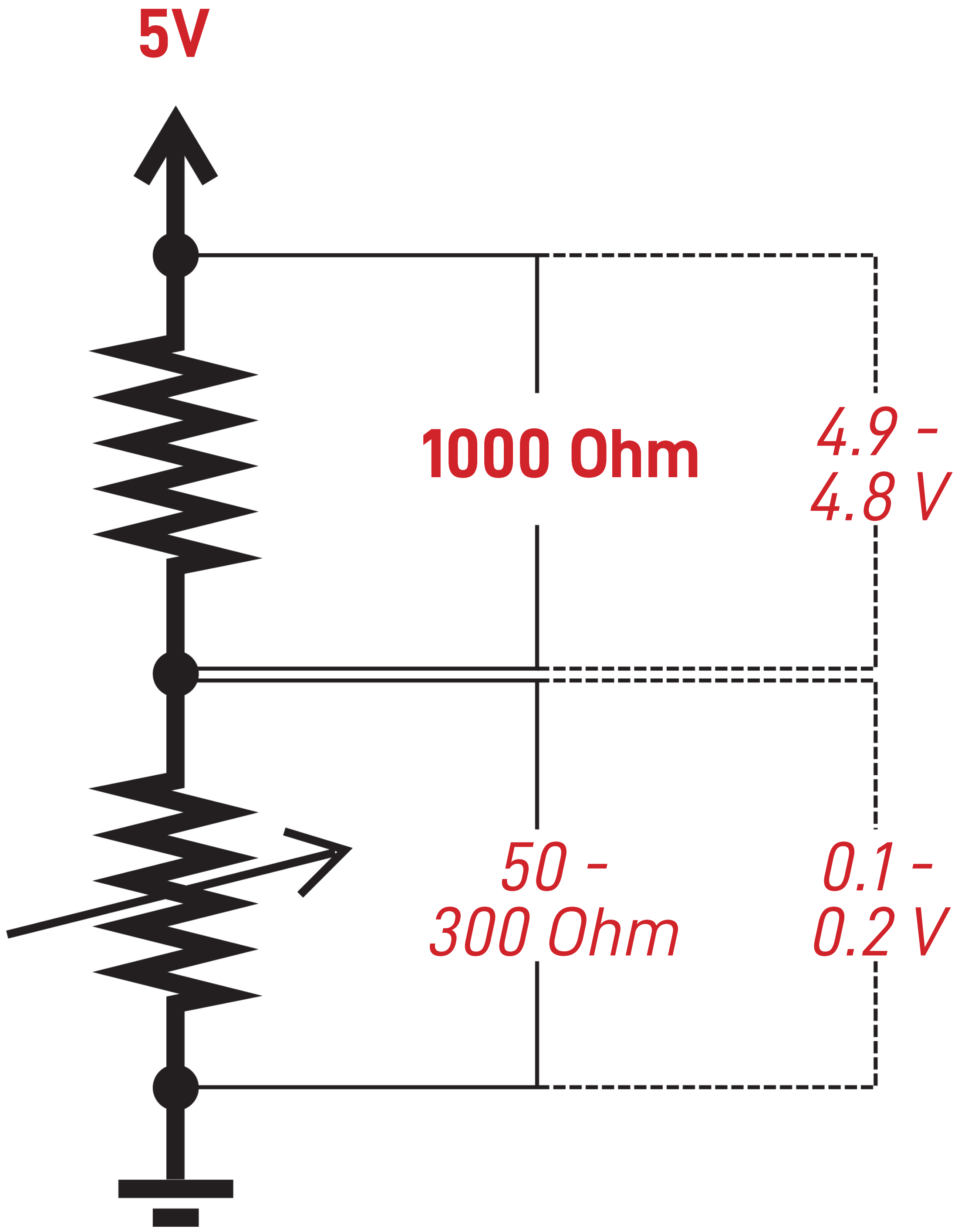
$$\frac{R1}{R2} = \frac{V1}{V2}$$

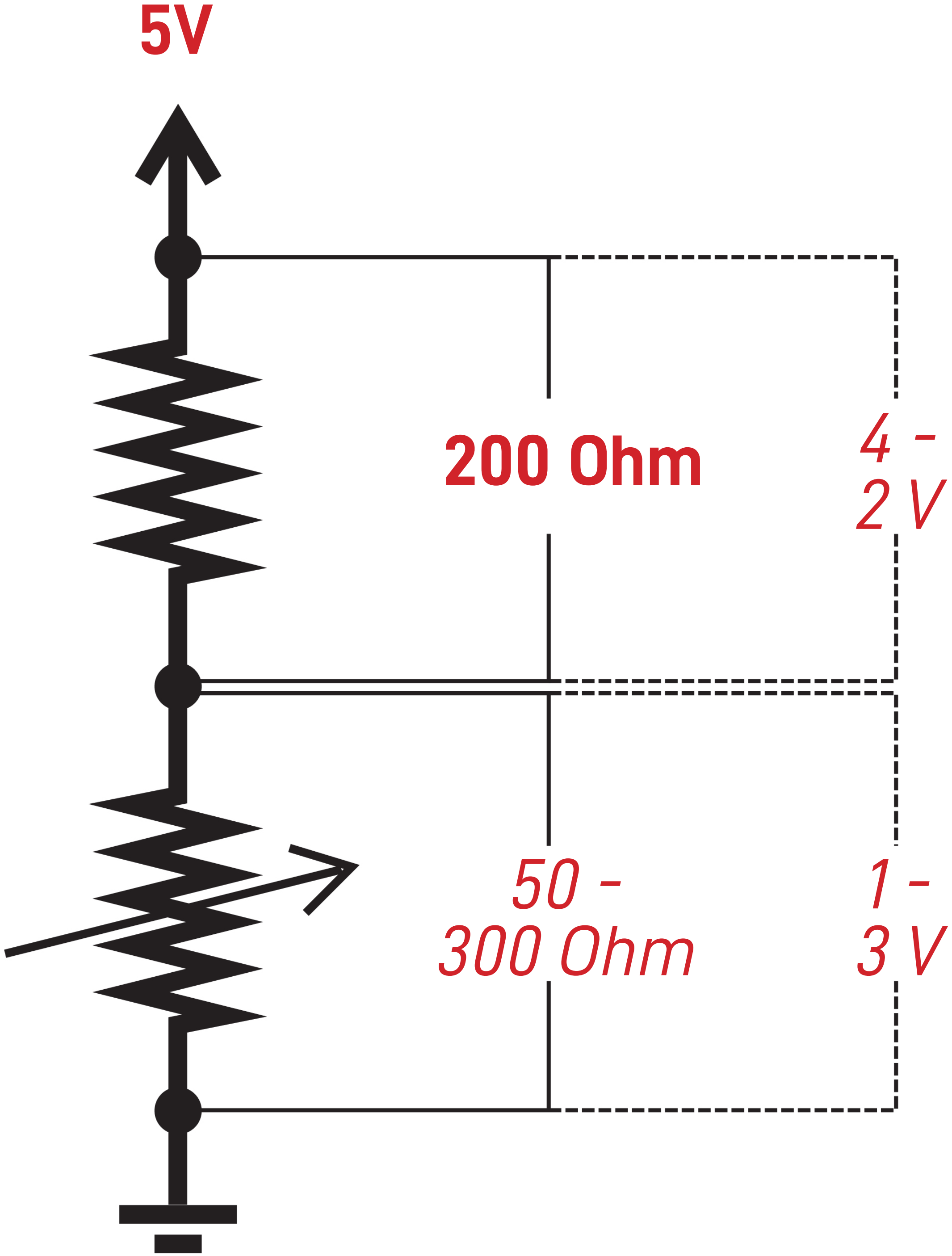


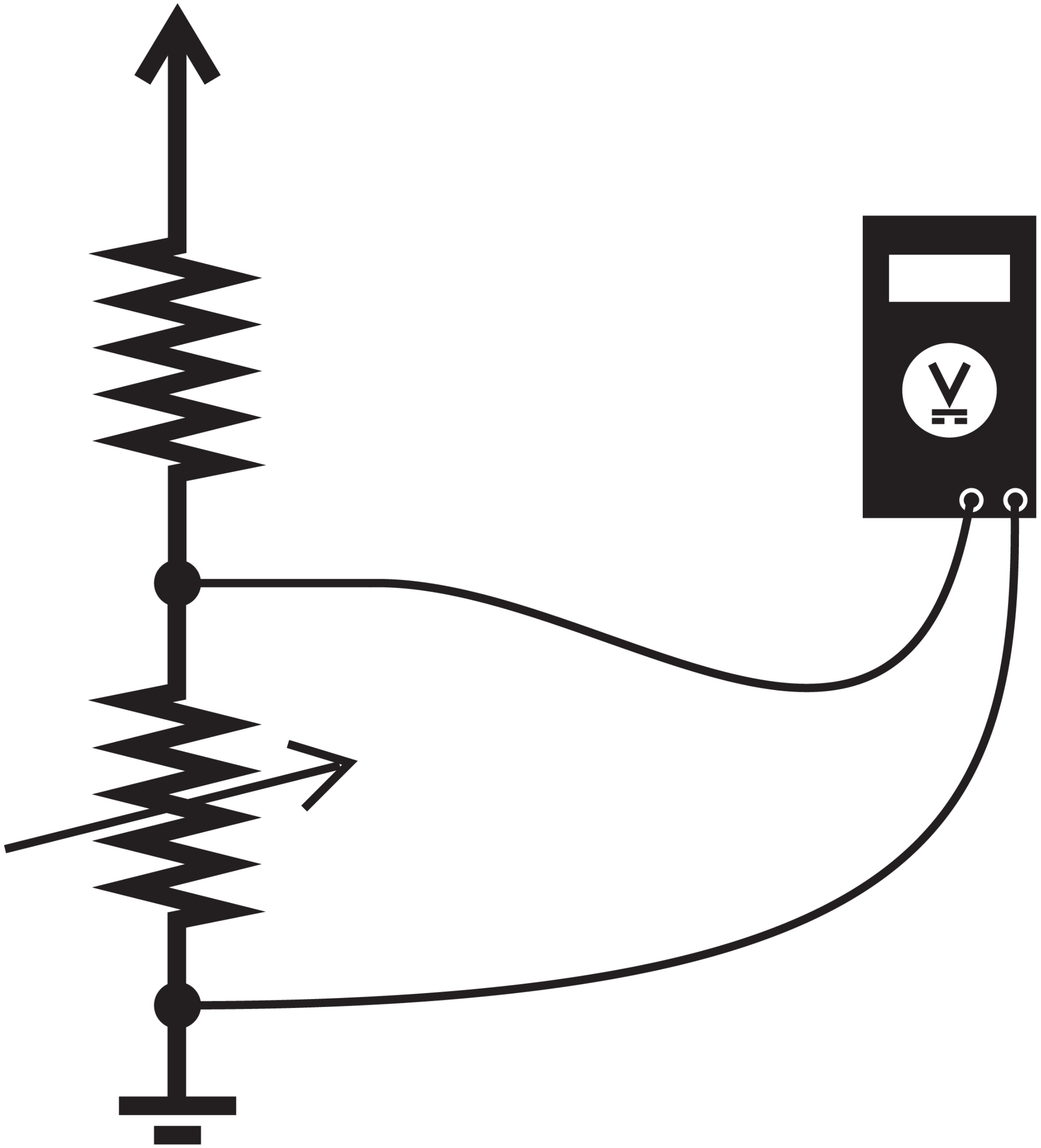
$$\frac{R1}{R2} = \frac{V1}{V2}$$



$$\frac{R1}{R2} = \frac{V1}{V2}$$

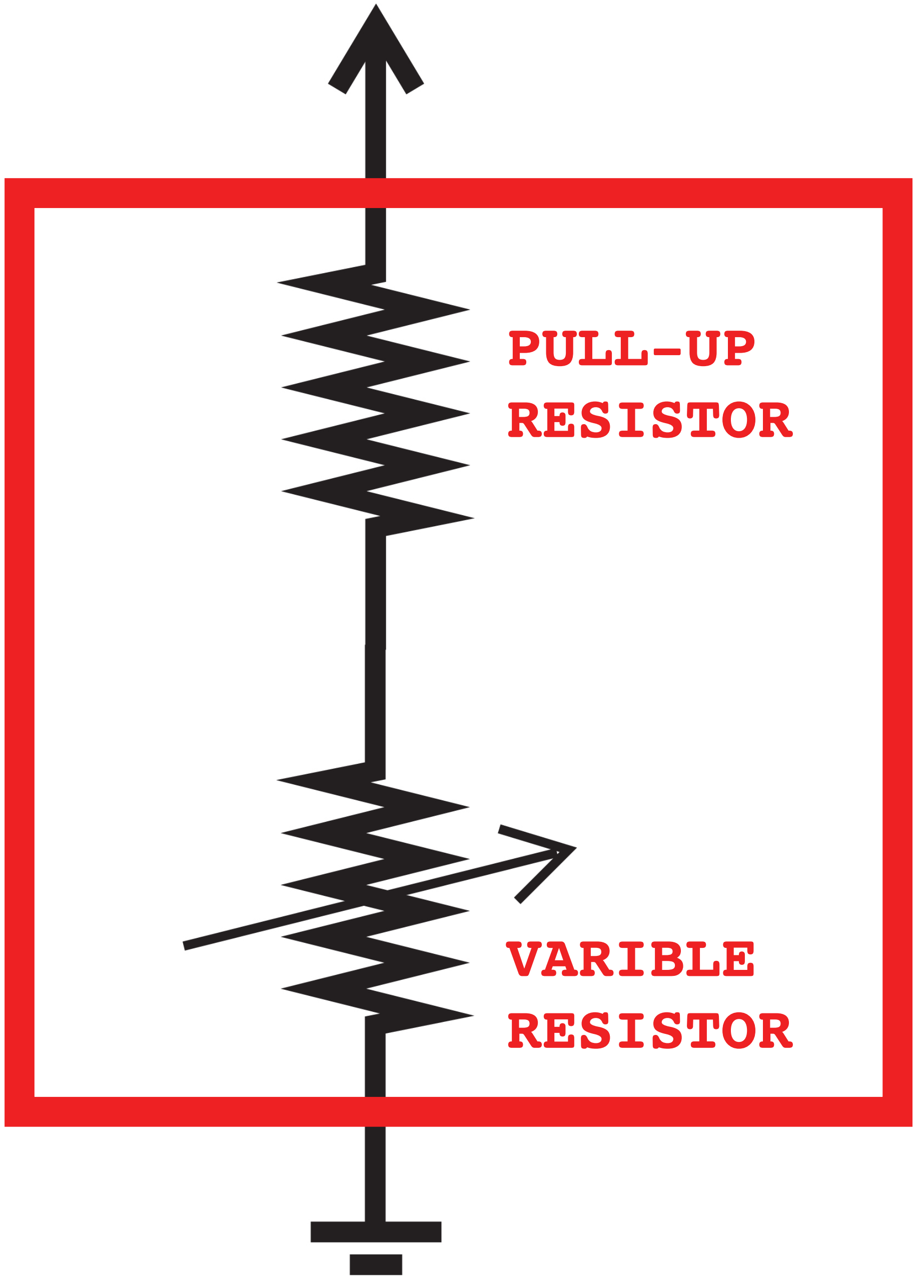

5V
1000 Ohm
4.9 -
4.8 V
50 -
300 Ohm
0.1 -
0.2 V


5V
200 Ohm
4 -
2 V
50 -
300 Ohm
1 -
3 V



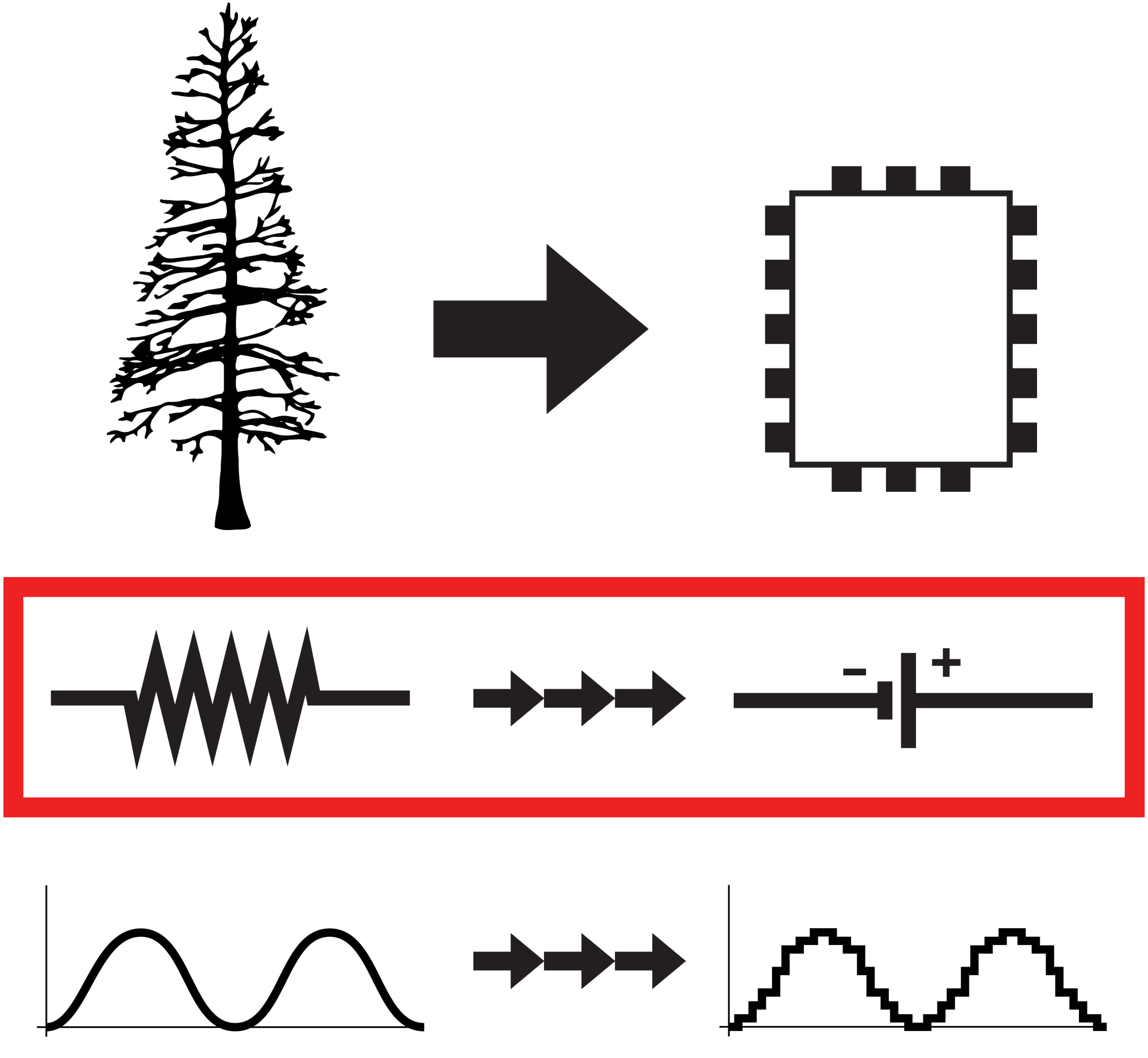
light press: ________ *V*

hard press: ________ *V*

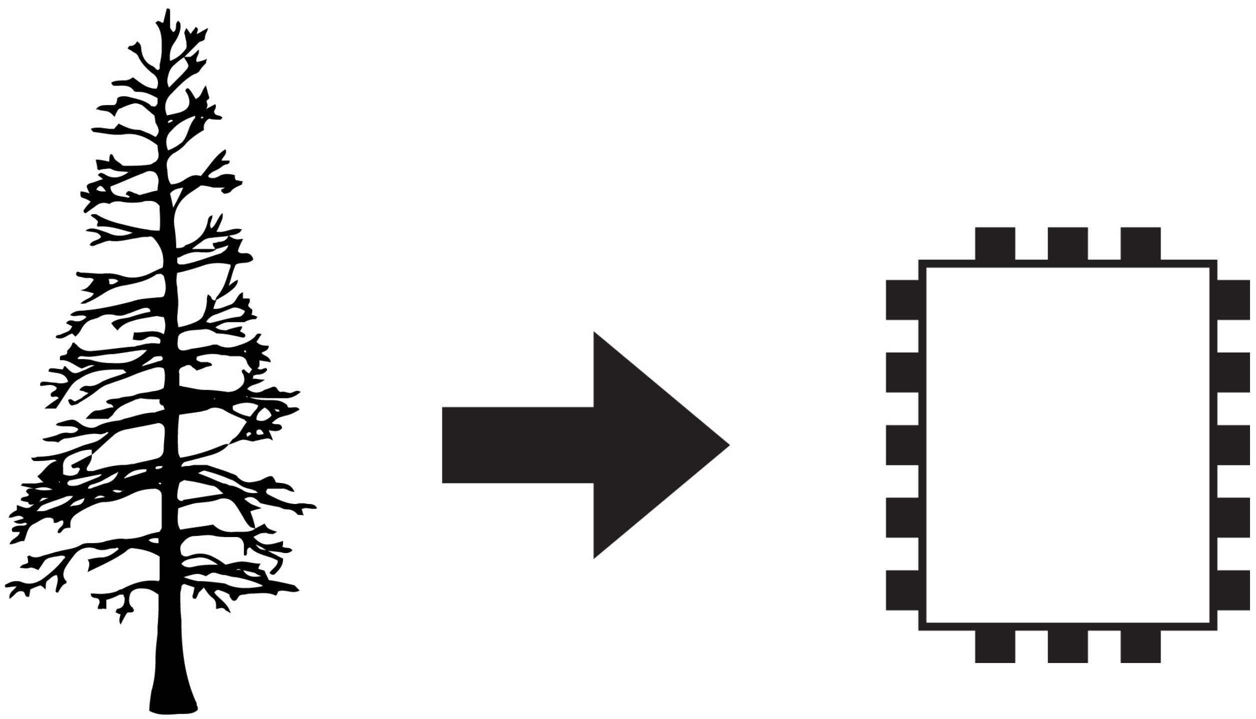


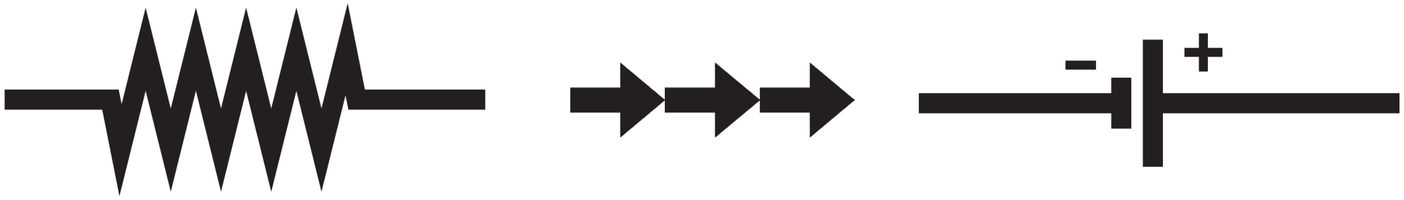
VOLTAGE
DIVIDER

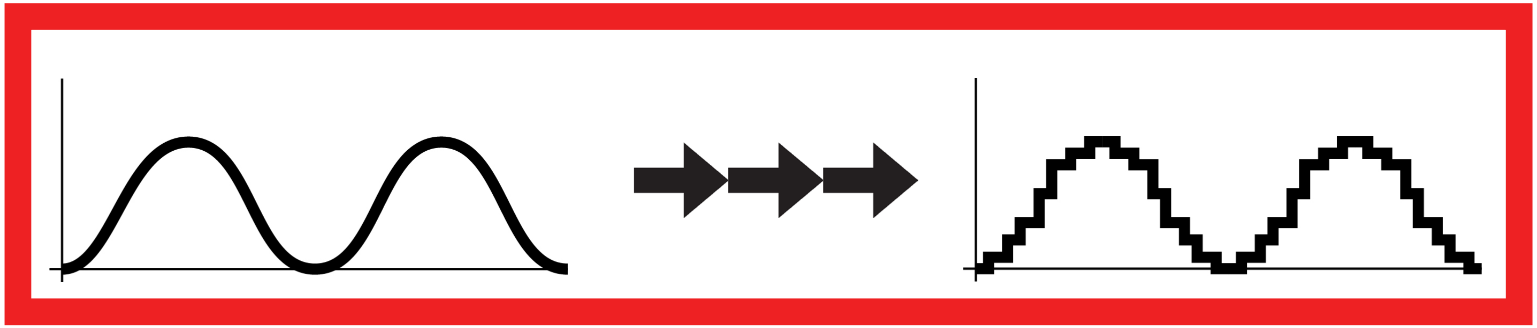
QUESTIONS?



VOLTAGE DIVIDER

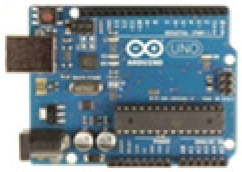




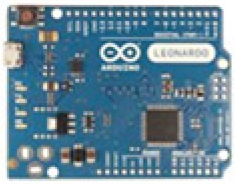


ANALOG TO DIGITAL CONVERSION

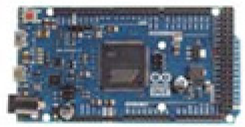
Arduino



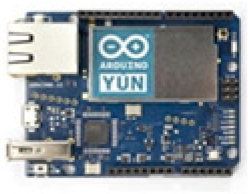
Arduino Uno



Arduino Leonardo

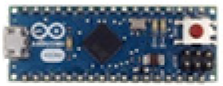


Arduino Due



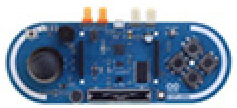
Arduino Yún

Arduino Tre

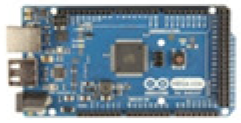


Arduino Micro

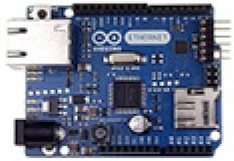
Arduino Robot



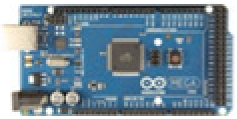
Arduino Esplora



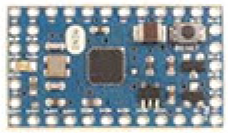
Arduino Mega ADK



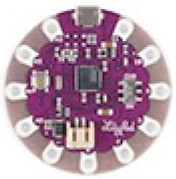
Arduino Ethernet



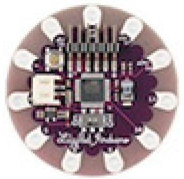
Arduino Mega 2560



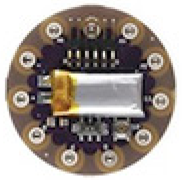
Arduino Mini



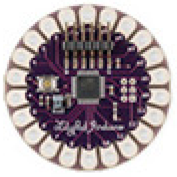
LilyPad Arduino USB



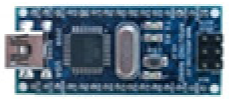
LilyPad Arduino Simple



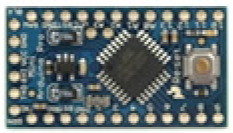
LilyPad Arduino SimpleSnap



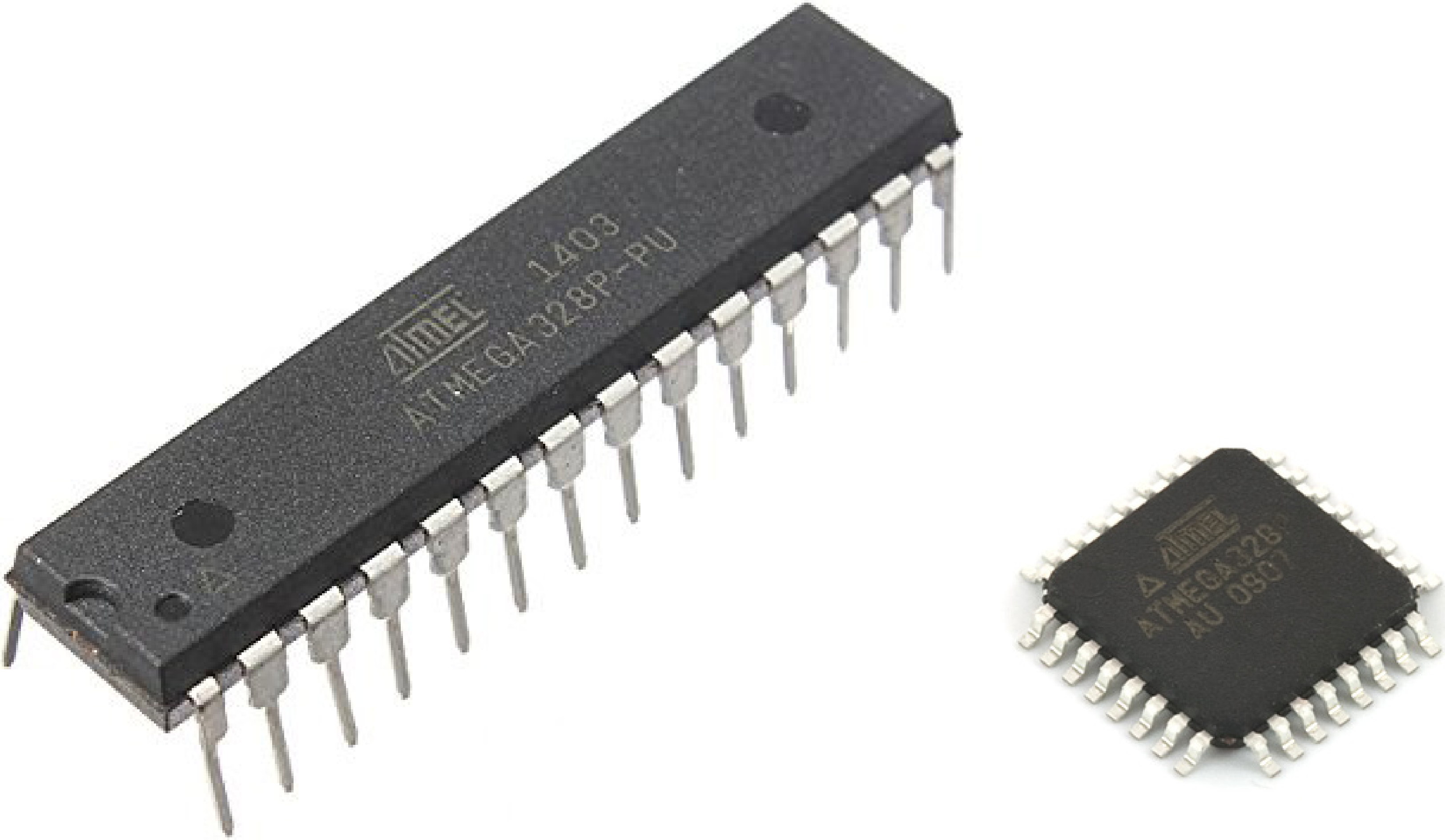
LilyPad Arduino

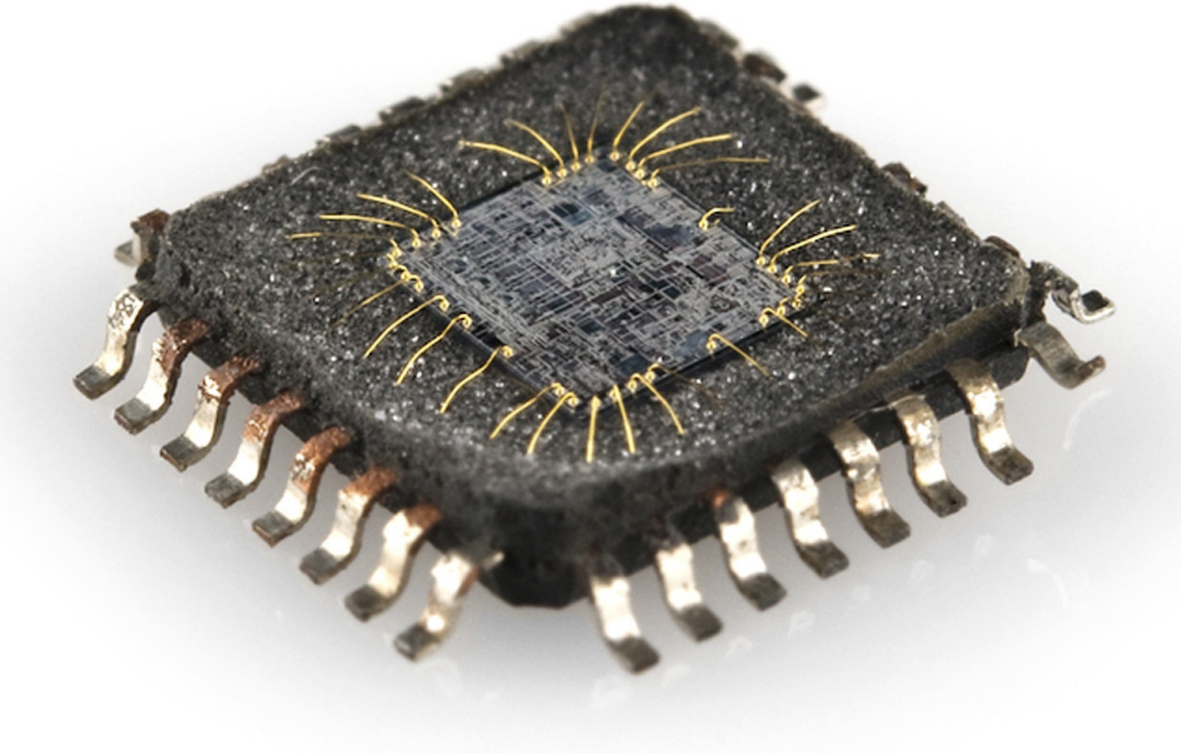


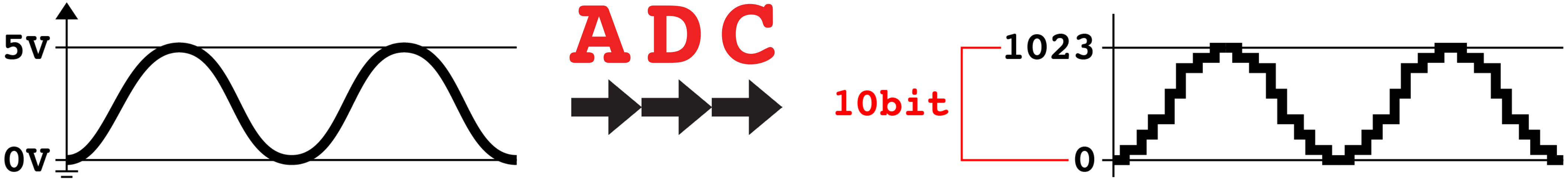
Arduino Nano

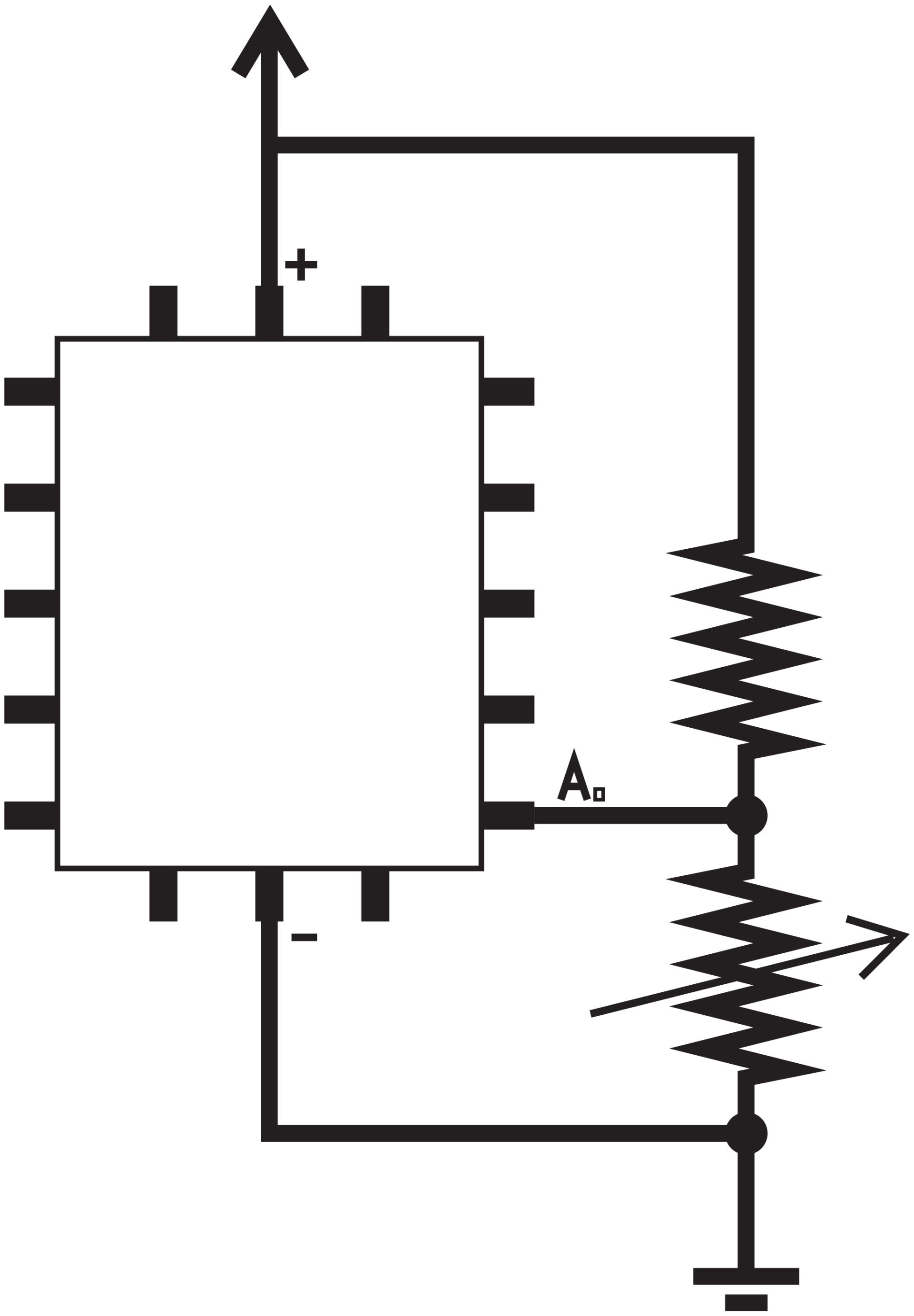


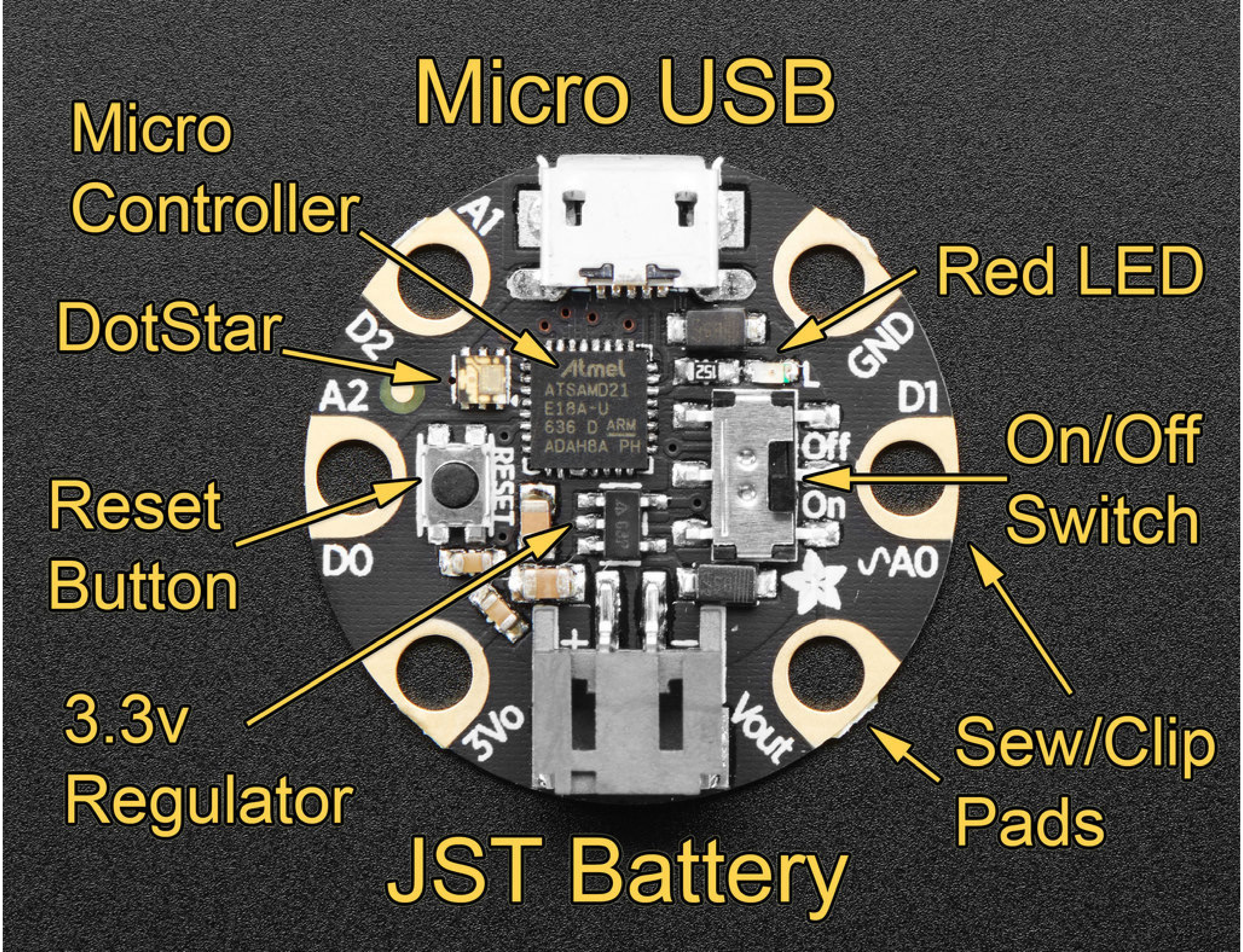
Arduino Pro Mini

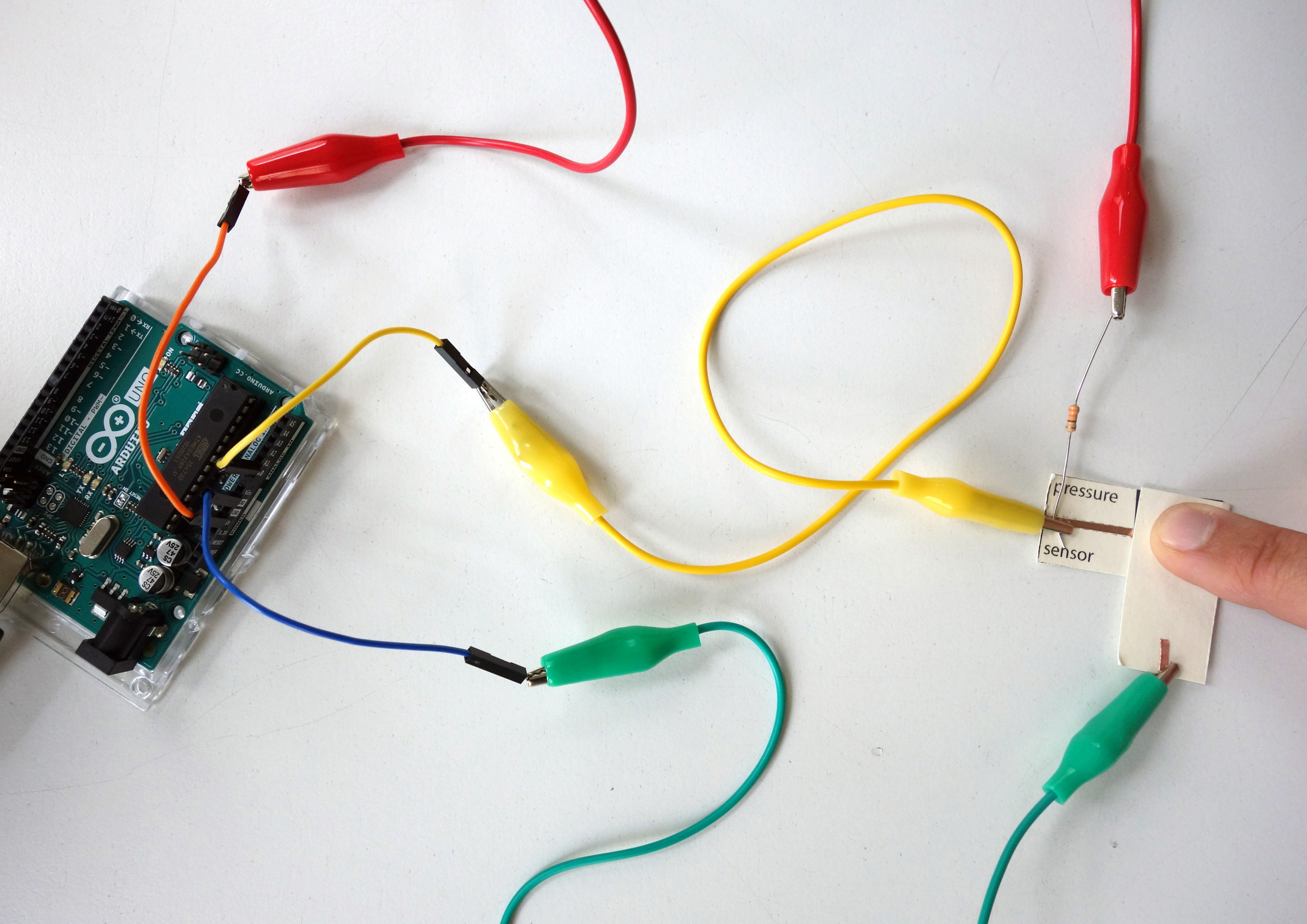

1403
ATMEGA328P-PU
ATMEGA328
AU 0807



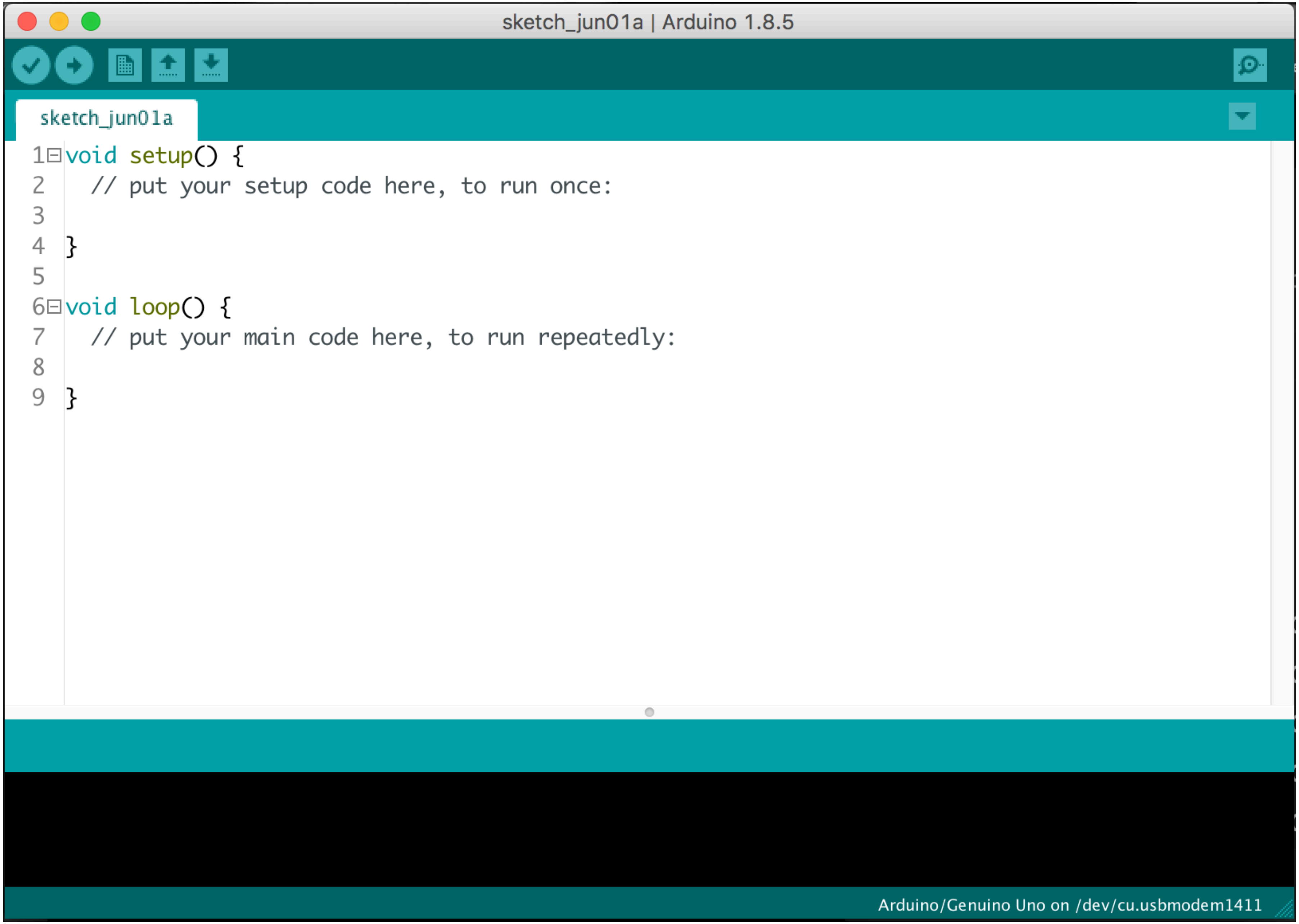

5V
0V
ADC
10bit
1023
0


+
-
A0


Micro USB
Micro Controller
Red LED
DotStar
On/Off Switch
Reset Button
3.3v Regulator
Sew/Clip Pads
JST Battery
A1
D2
A2
D0
3Vo
GND
D1
Off
On
A0
Vout
RESET
Atmel
ATSAMD21
E18A-U
636 D ARM
ADAH8A PH


pressure
sensor
ARDUINO
UNO

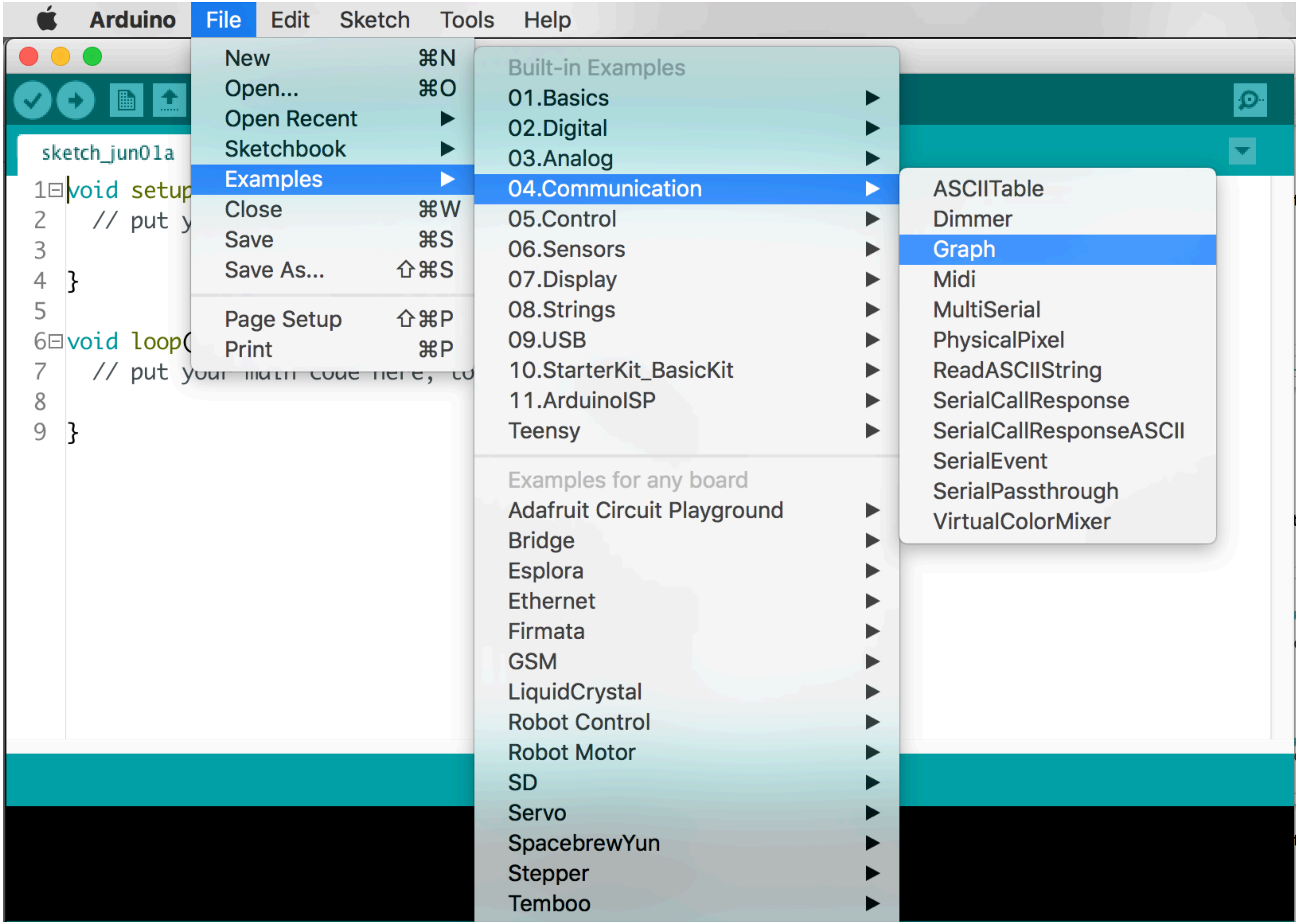
open Arduino IDE...



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

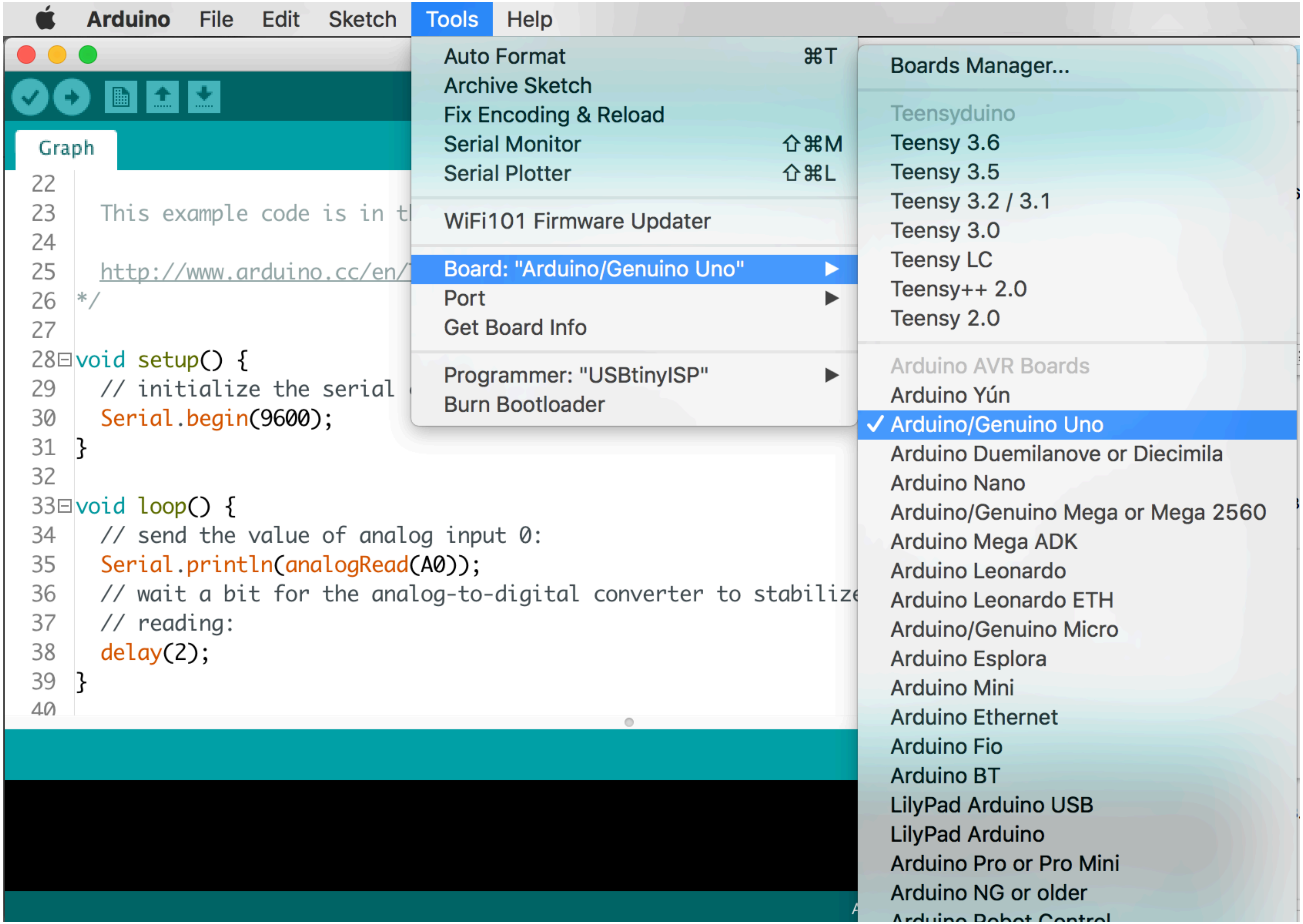
Arduino/Genuino Uno on /dev/cu.usbmodem1411

open: File >> Examples >> Communication >> Graph



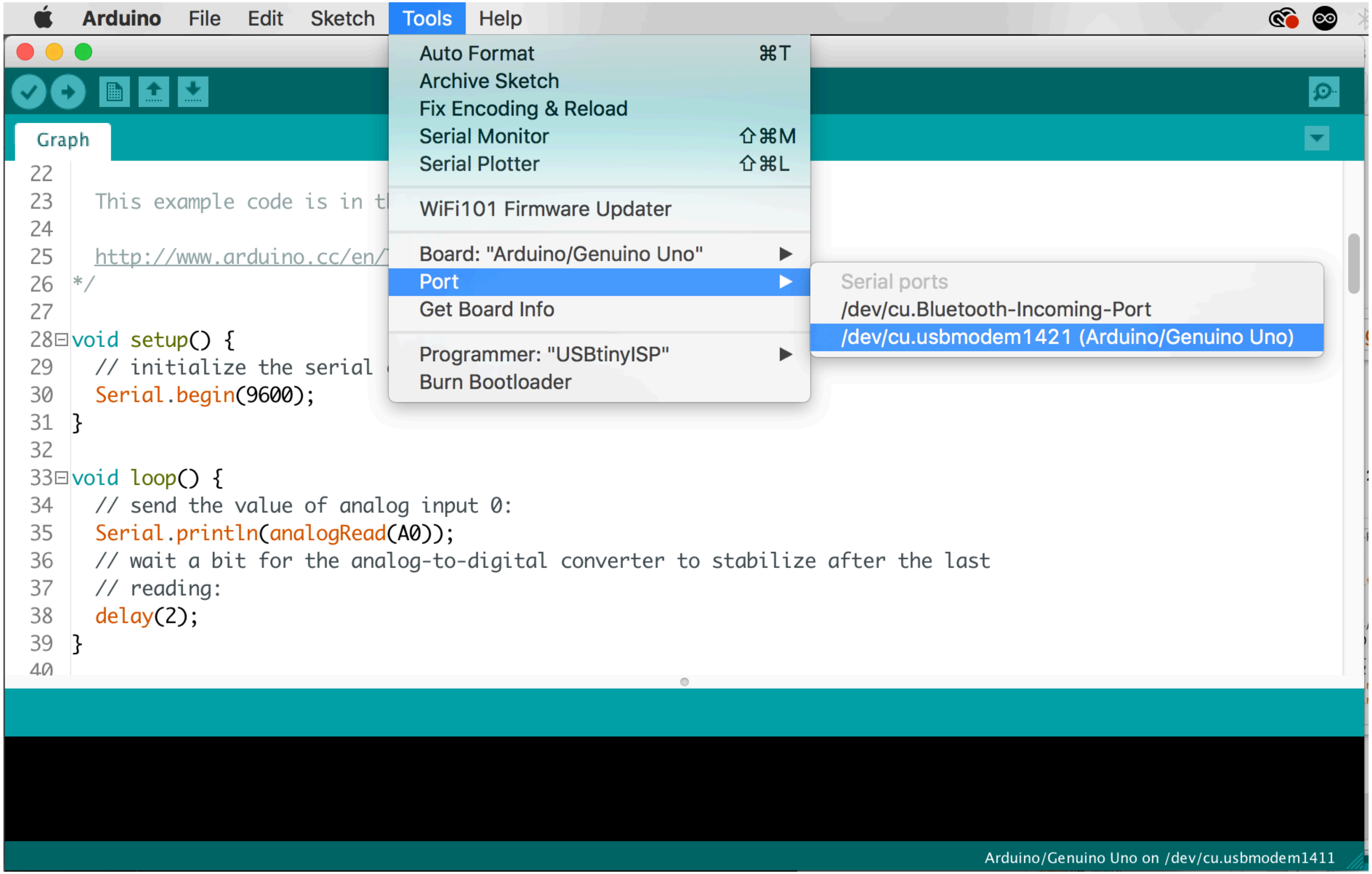
```
void setup
  // put y

}

void loop(
  // put your main code here, to

}
```

select Board



```

  This example code is in t

  http://www.arduino.cc/en/
*/

void setup() {
  // initialize the serial 
  Serial.begin(9600);
}

void loop() {
  // send the value of analog input 0:
  Serial.println(analogRead(A0));
  // wait a bit for the analog-to-digital converter to stabilize
  // reading:
  delay(2);
}
```

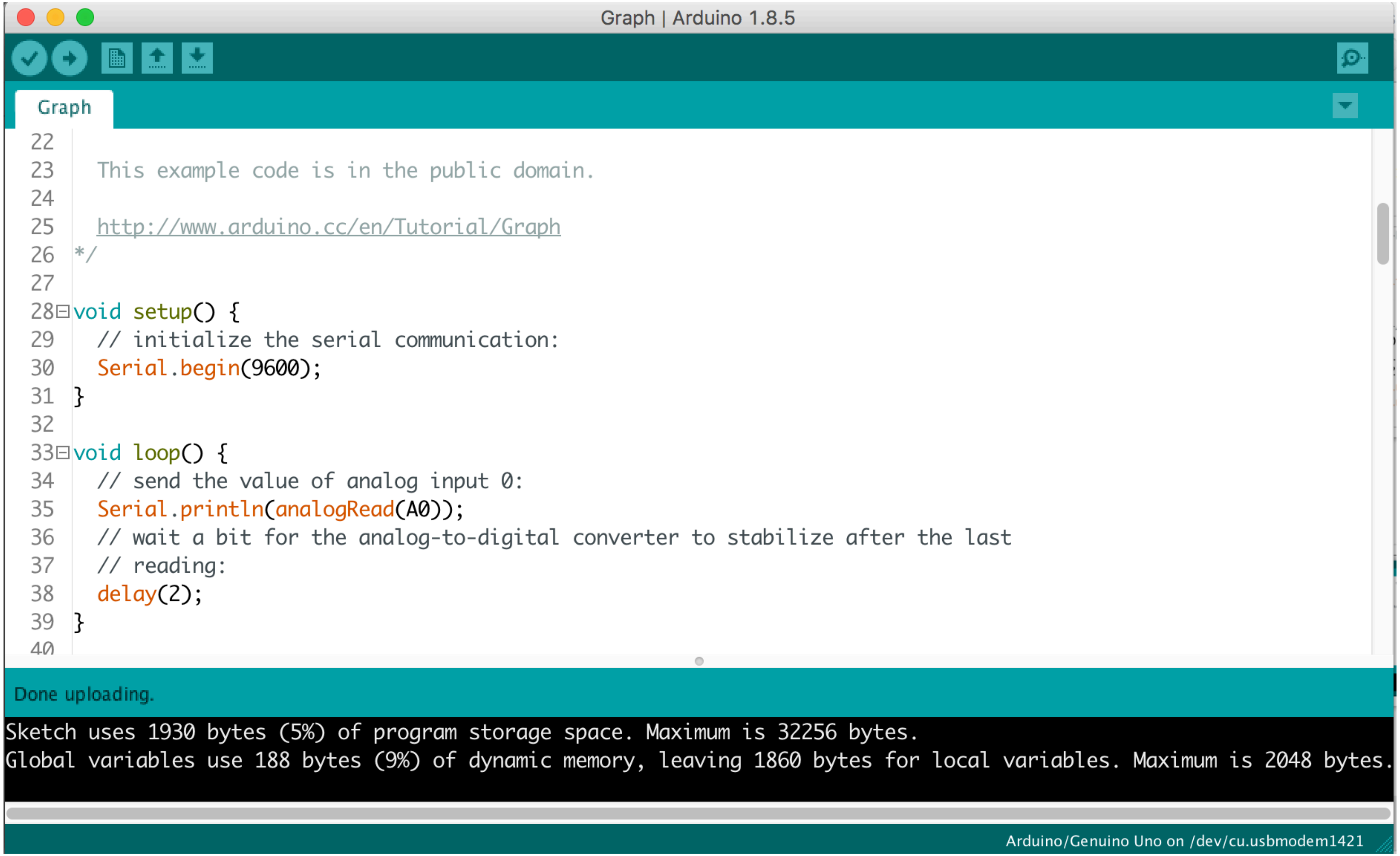
select Port



```

  This example code is in t

  http://www.arduino.cc/en/
*/

void setup() {
  // initialize the serial
  Serial.begin(9600);
}

void loop() {
  // send the value of analog input 0:
  Serial.println(analogRead(A0));
  // wait a bit for the analog-to-digital converter to stabilize after the last
  // reading:
  delay(2);
}

```

Arduino/Genuino Uno on /dev/cu.usbmodem1411

upload

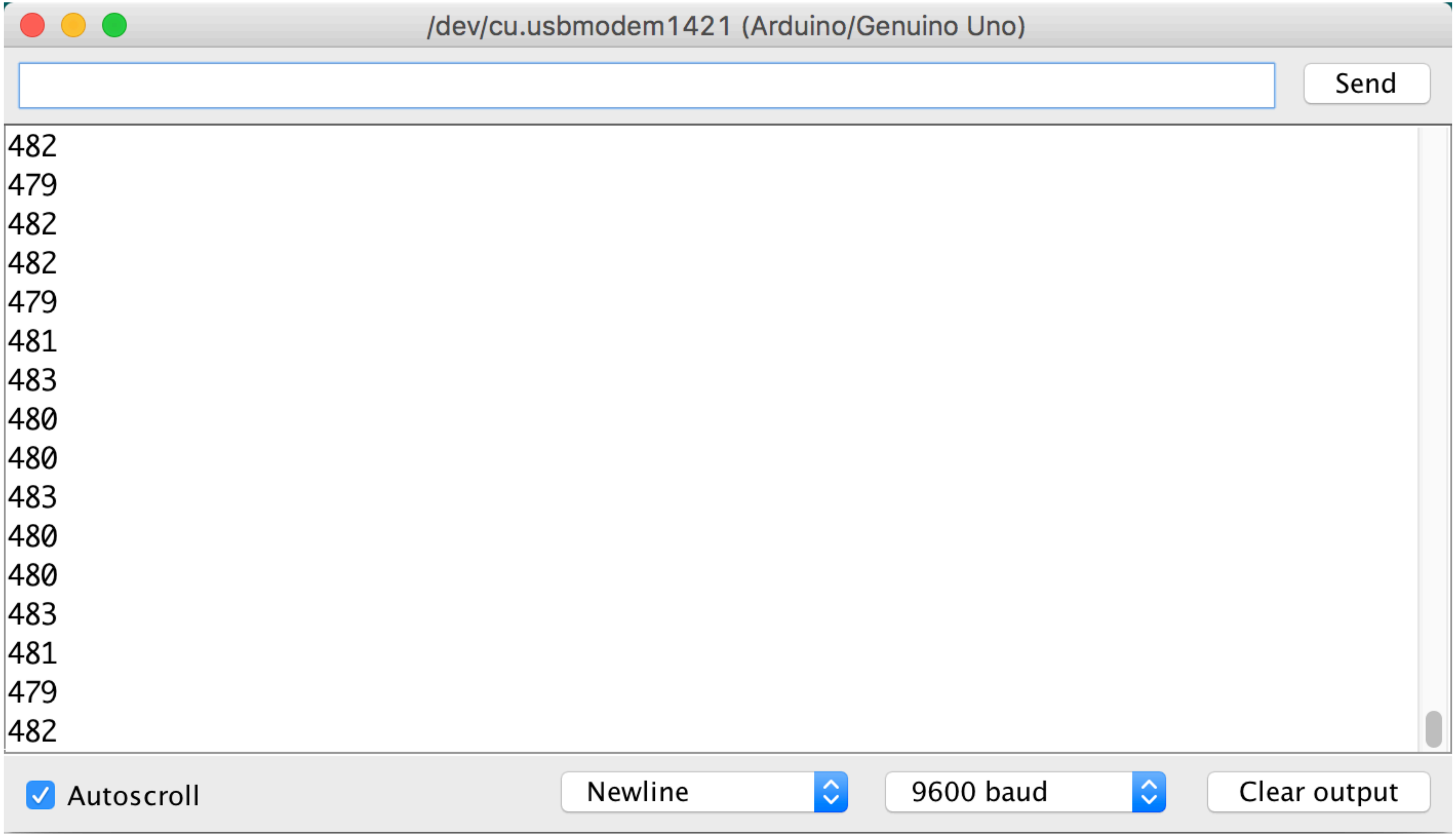


```

  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Graph
*/

void setup() {
  // initialize the serial communication:
  Serial.begin(9600);
}

void loop() {
  // send the value of analog input 0:
  Serial.println(analogRead(A0));
  // wait a bit for the analog-to-digital converter to stabilize after the last
  // reading:
  delay(2);
}

```

Done uploading.

```
Sketch uses 1930 bytes (5%) of program storage space. Maximum is 32256 bytes.
Global variables use 188 bytes (9%) of dynamic memory, leaving 1860 bytes for local variables. Maximum is 2048 bytes.
```

open Serial Monitor

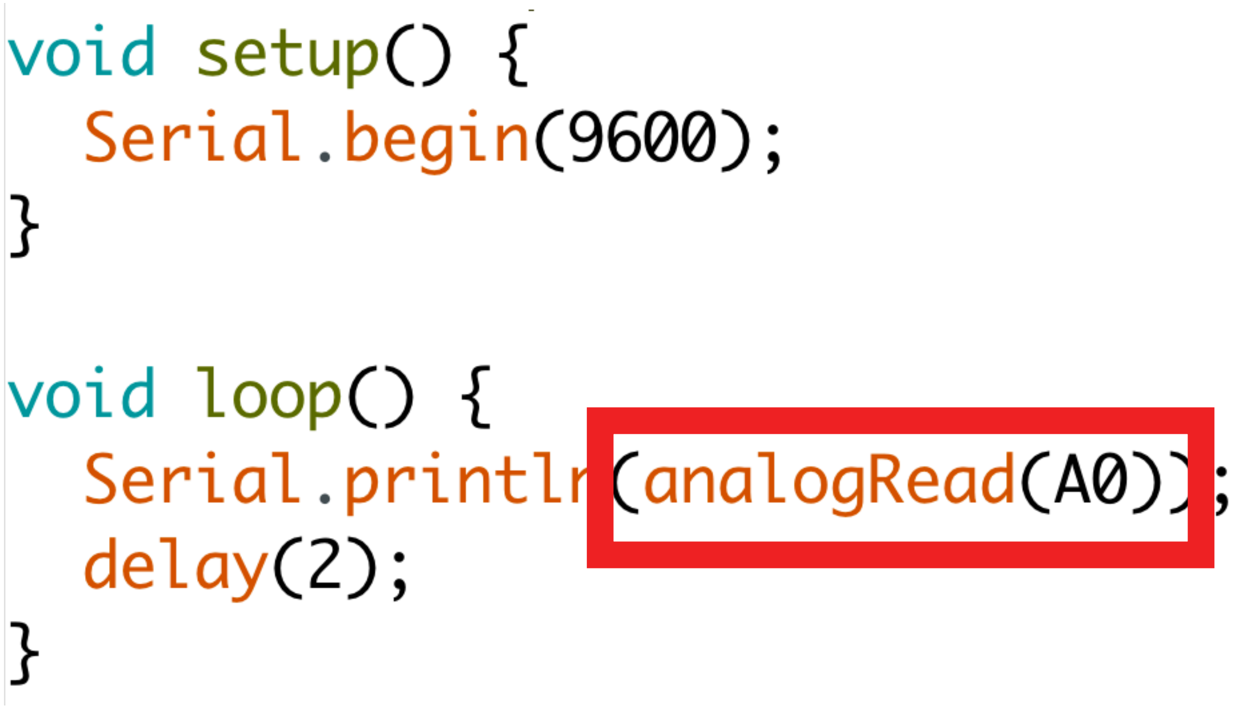


```
482
479
482
482
479
481
483
480
480
483
480
480
483
481
479
482
```

☑ Autoscroll

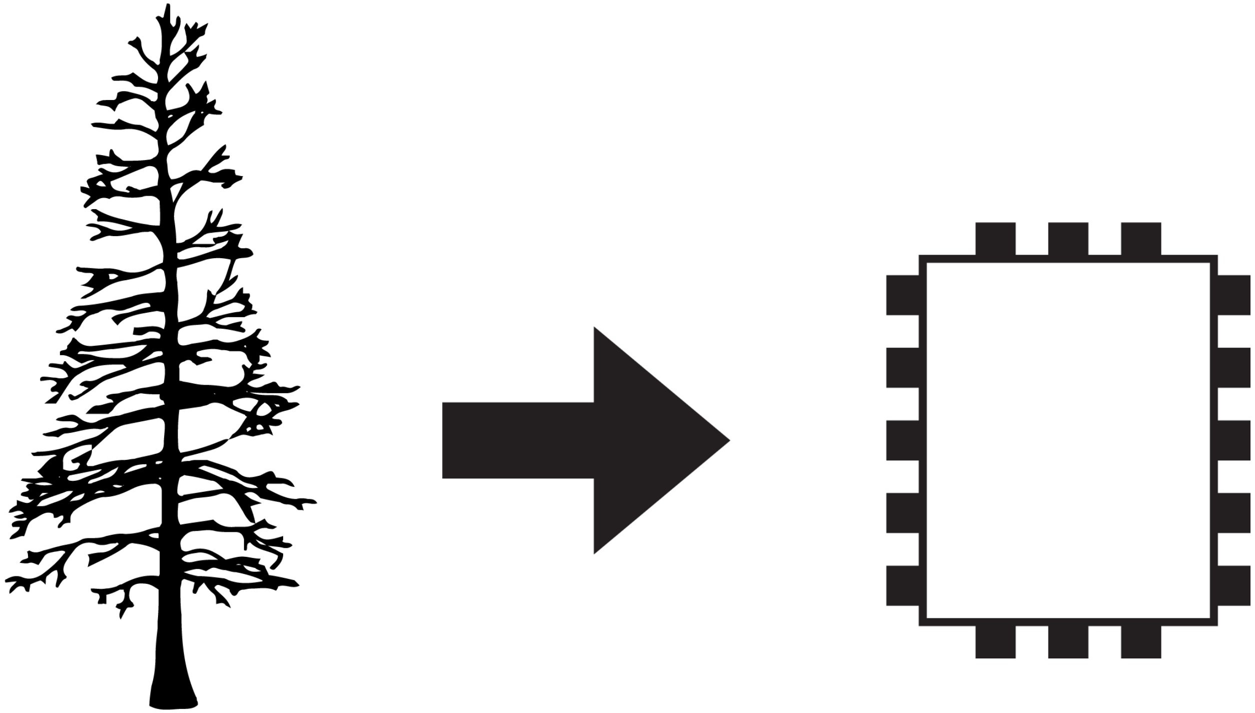
Newline

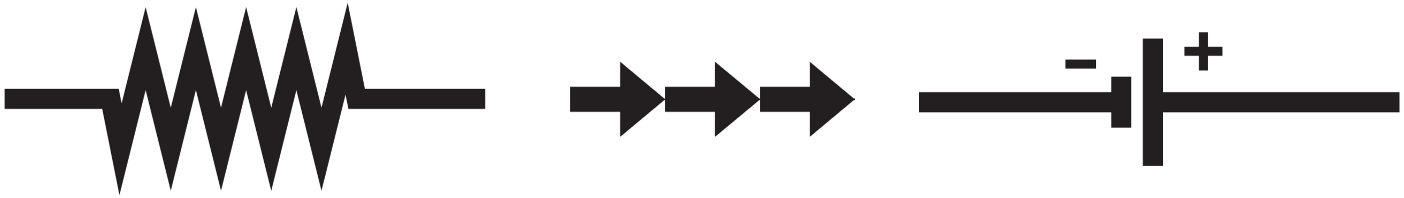
9600 baud

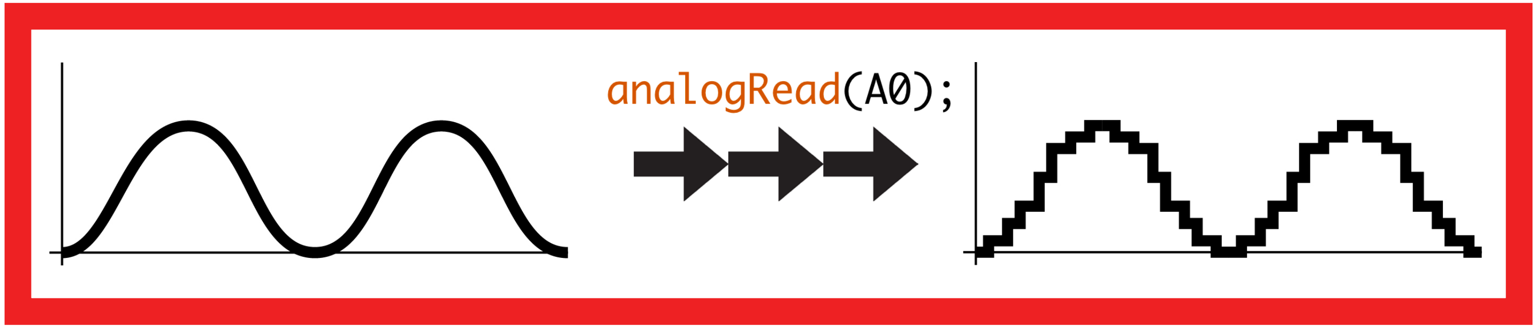


```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
  delay(2);
}
```

ANALOG DIGITAL CONVERTER (ADC)







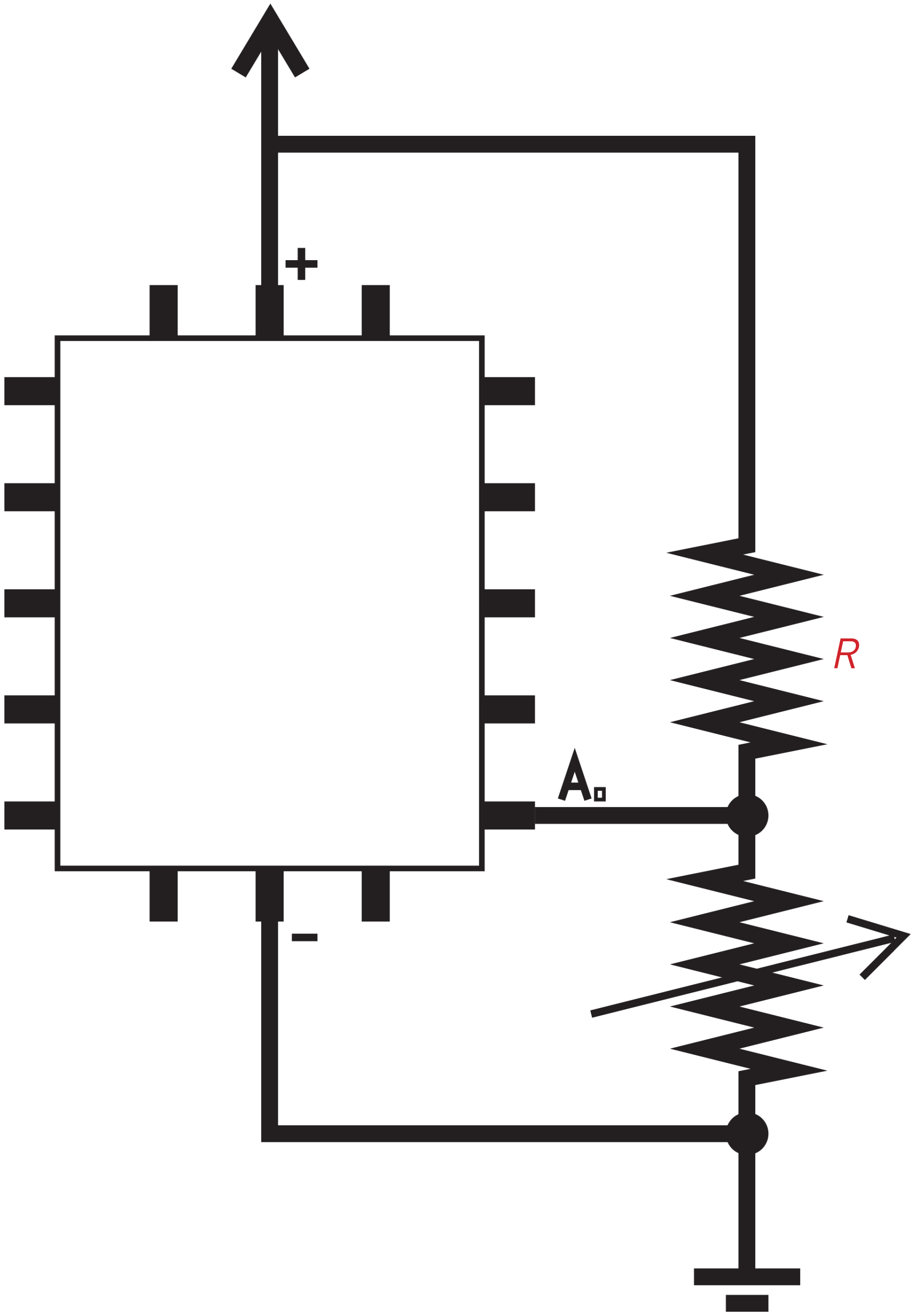
```
analogRead(A0);
```

ANALOG
DIGITAL
CONVERTER
(ADC)

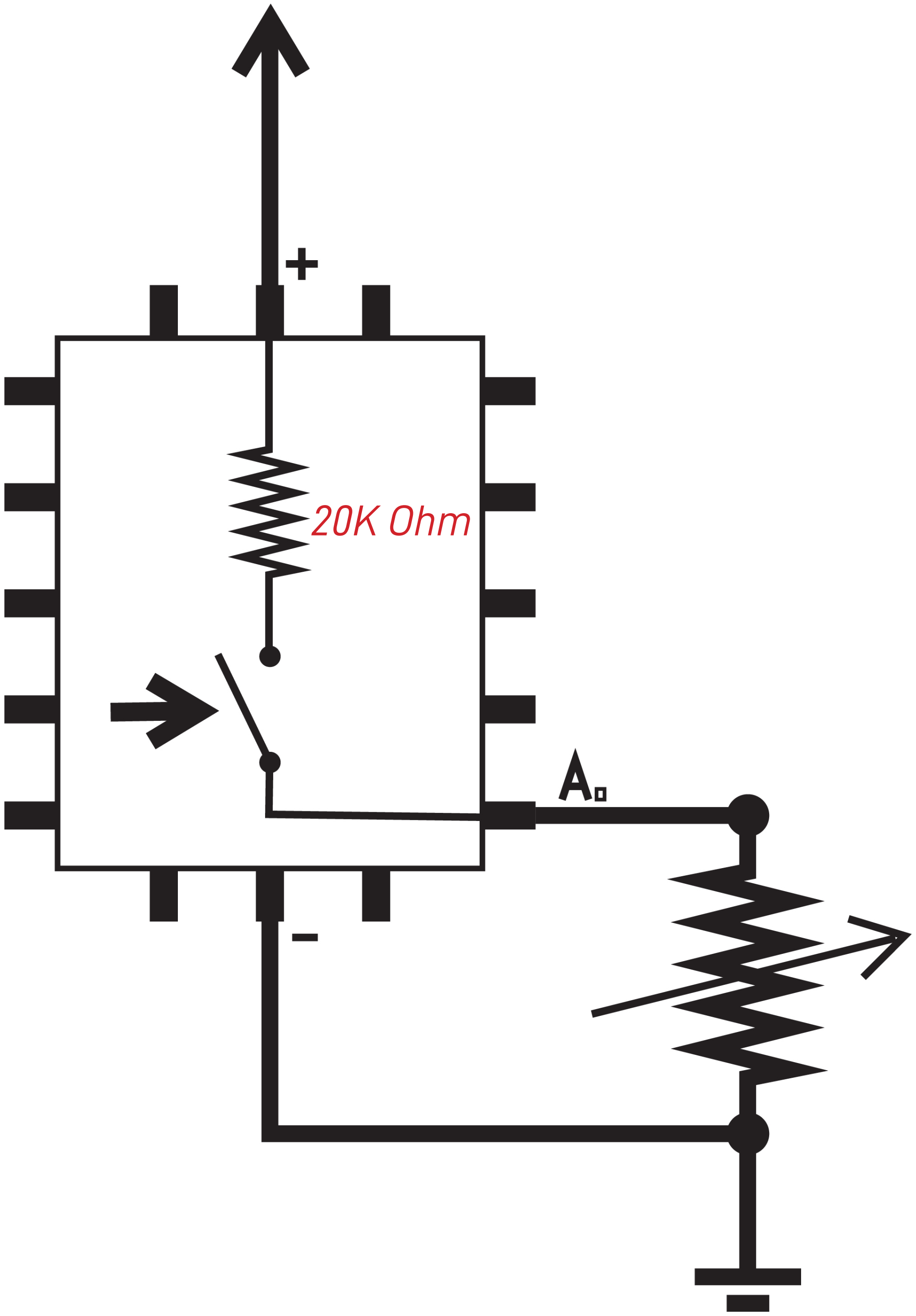
TIP:

internal pull-up resistors!

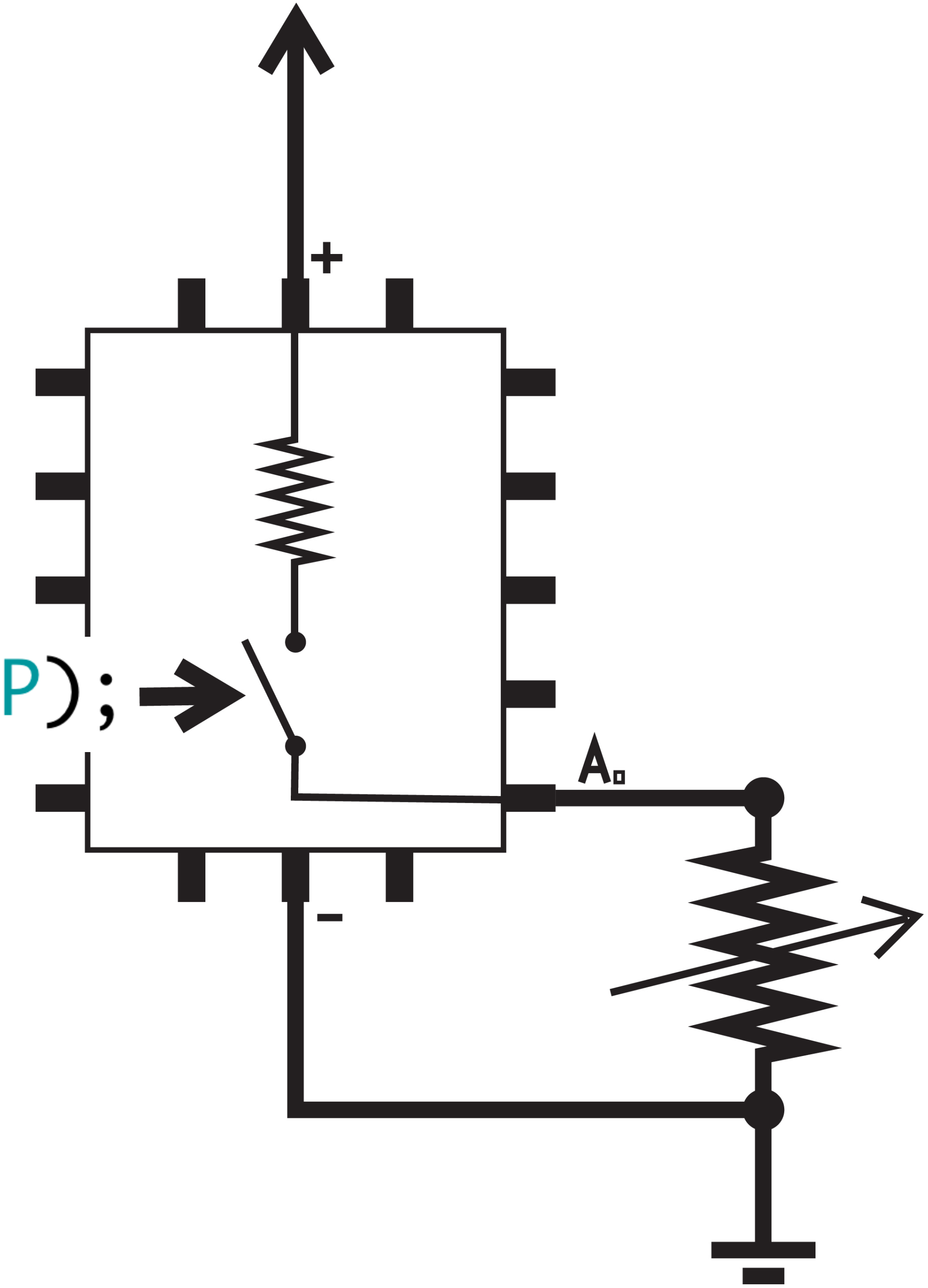
EXTERNAL PULLUP

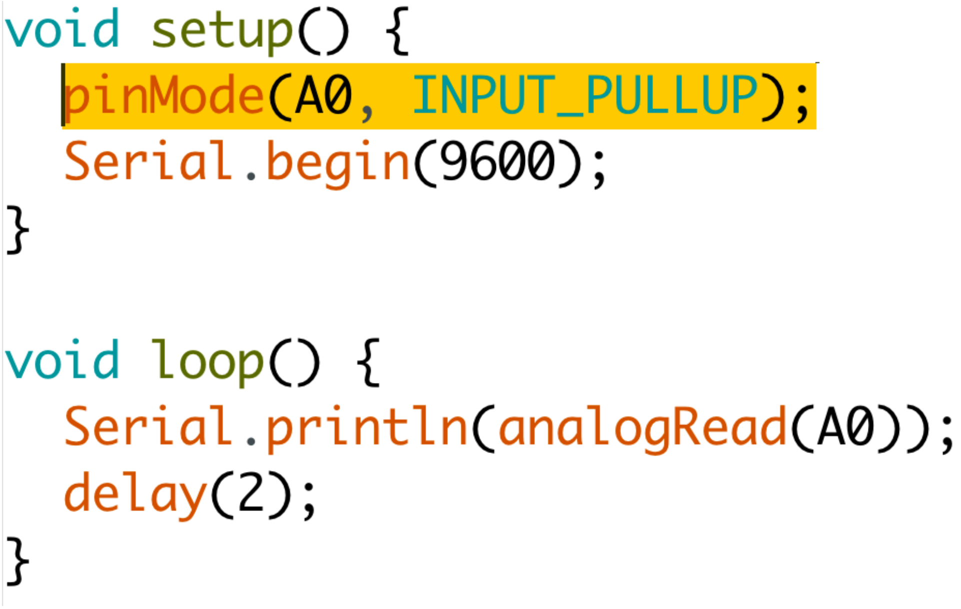


INTERNAL PULLUP (20K)

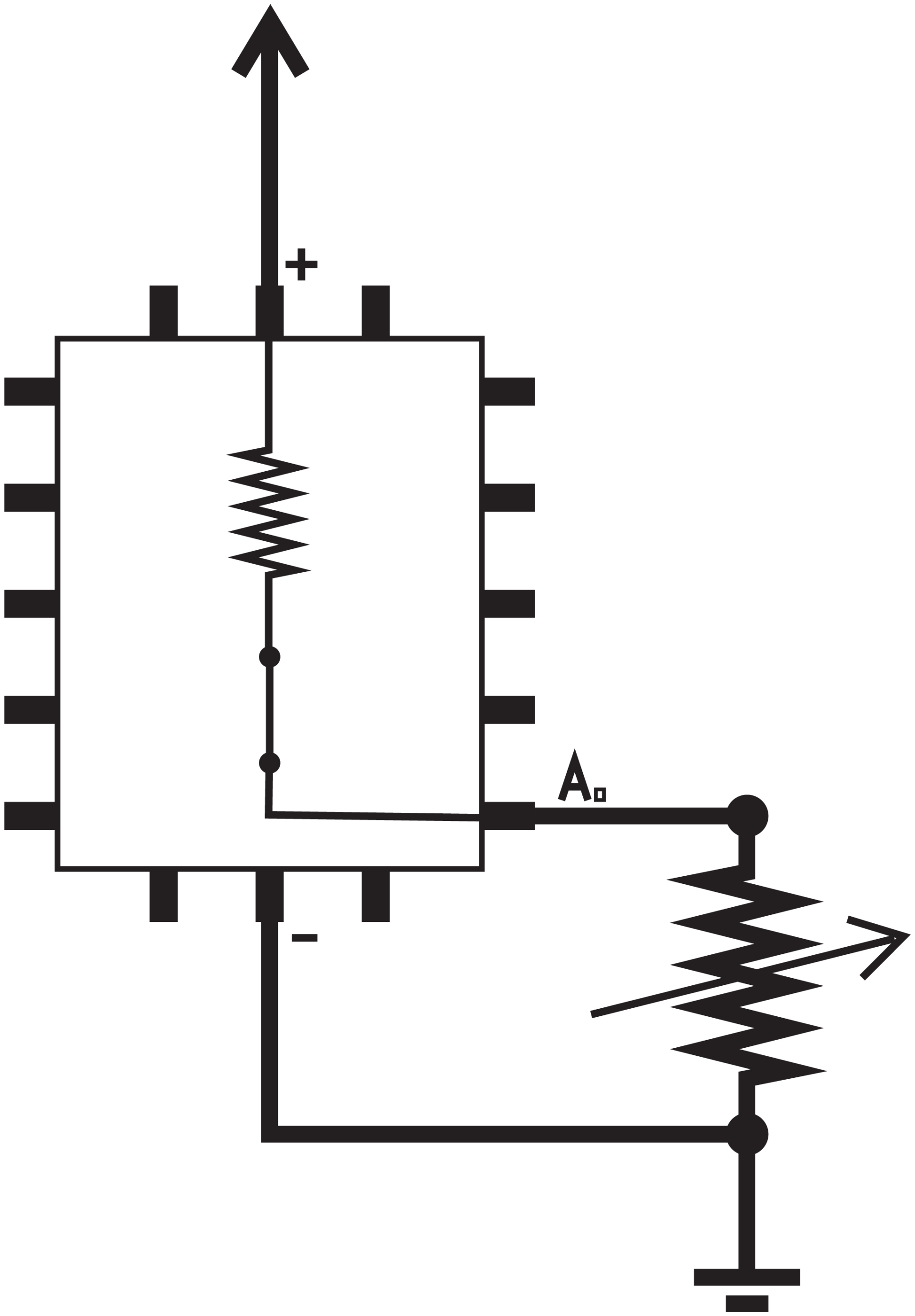


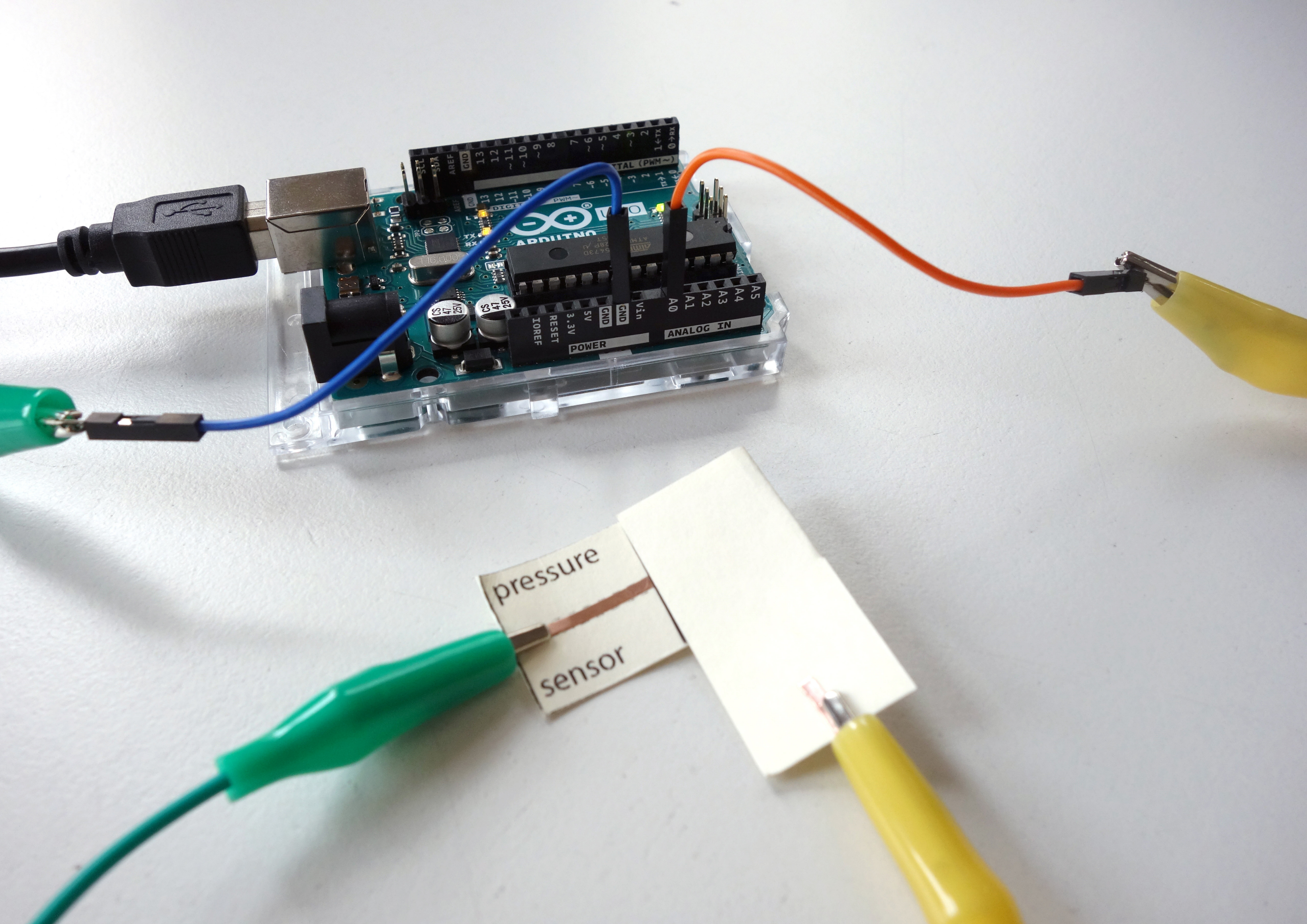
```
pinMode(A0, INPUT_PULLUP);
```



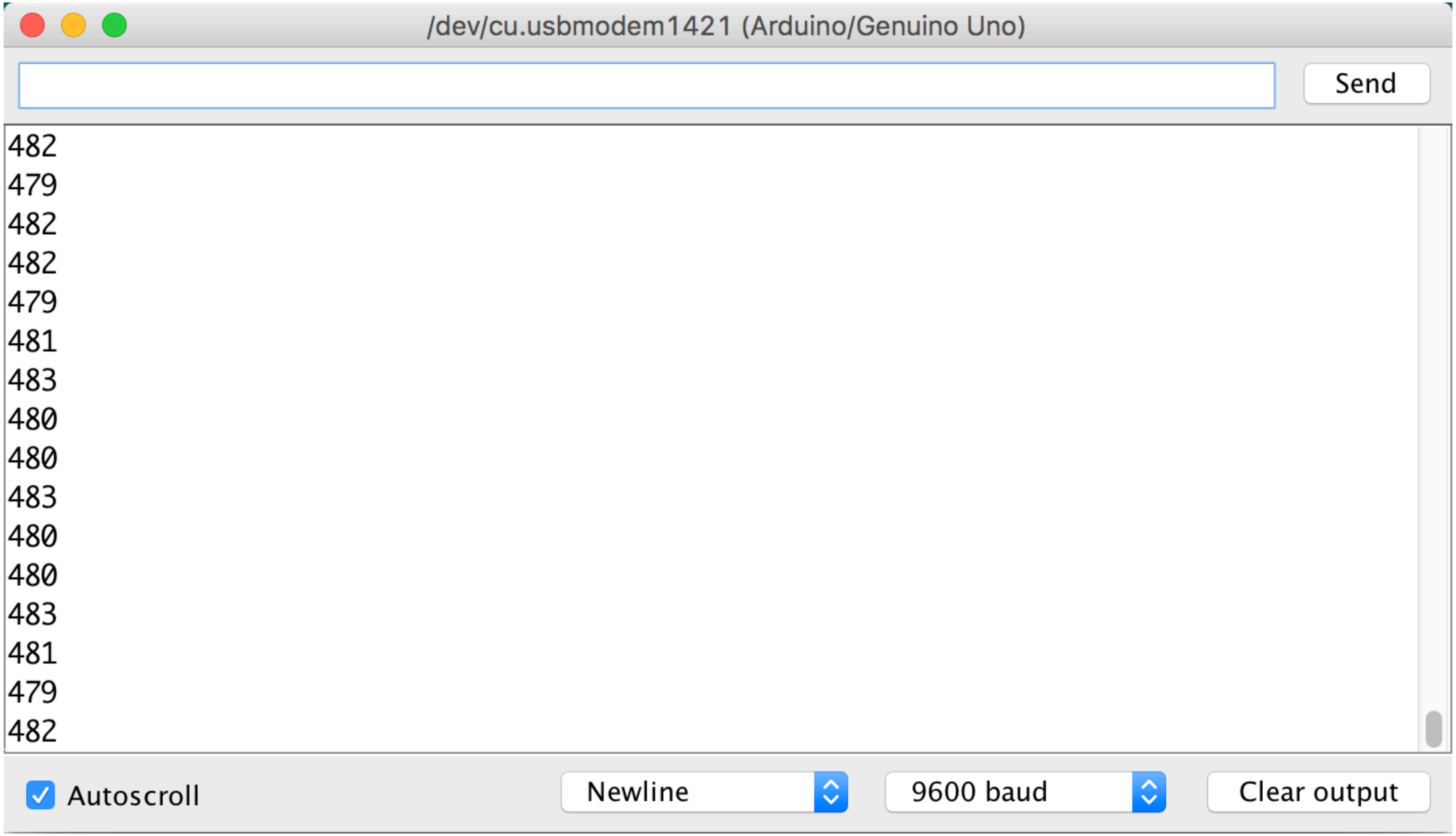


```
void setup() {
  pinMode(A0, INPUT_PULLUP);
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogRead(A0));
  delay(2);
}
```




pressure
sensor
ANALOG IN
POWER
A0
GND
GND
Vin
5V
3.3V
RESET
IOREF

open Serial Monitor



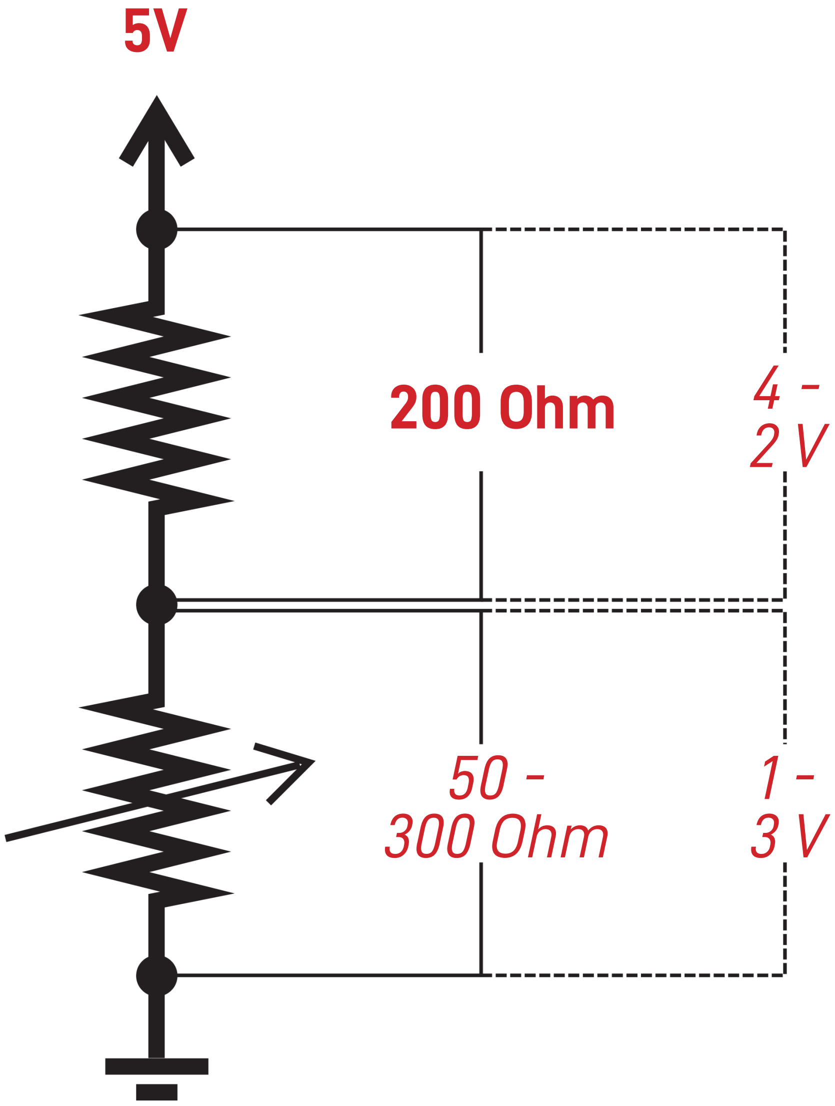
```
482
479
482
482
479
481
483
480
480
483
480
480
483
481
479
482
```

☑ Autoscroll

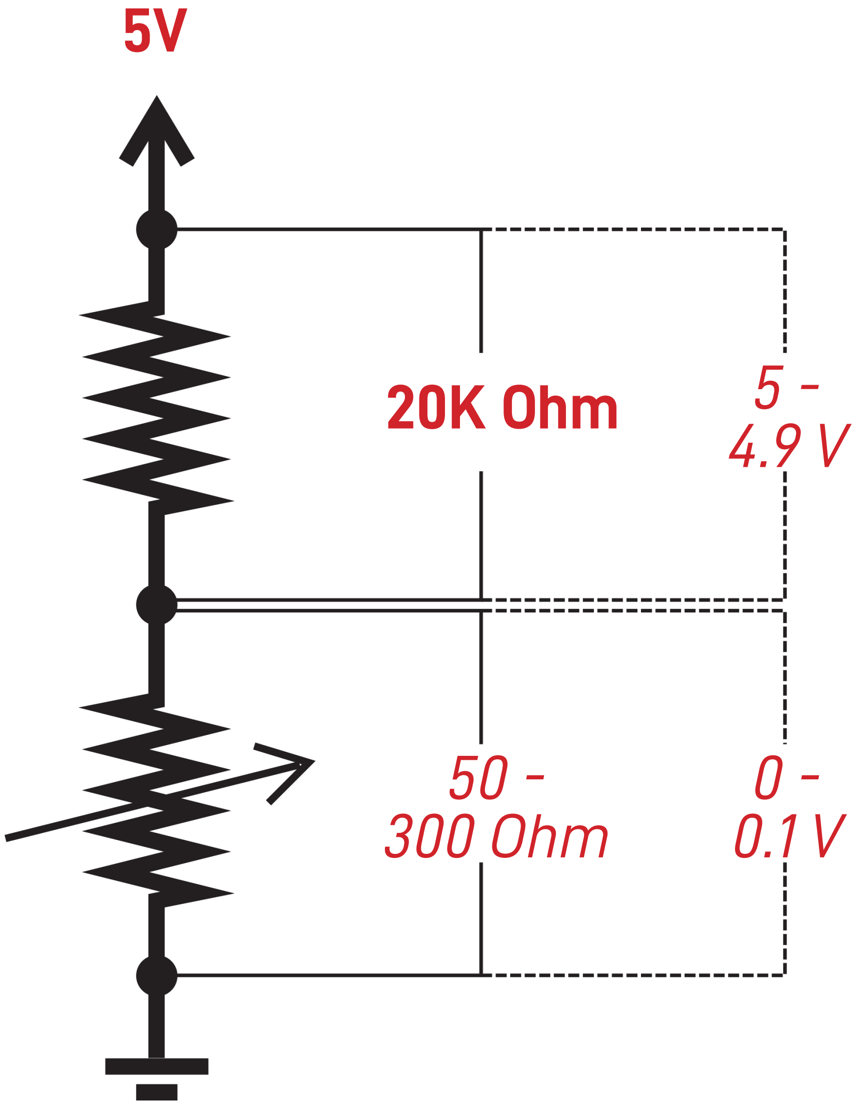
Newline

9600 baud

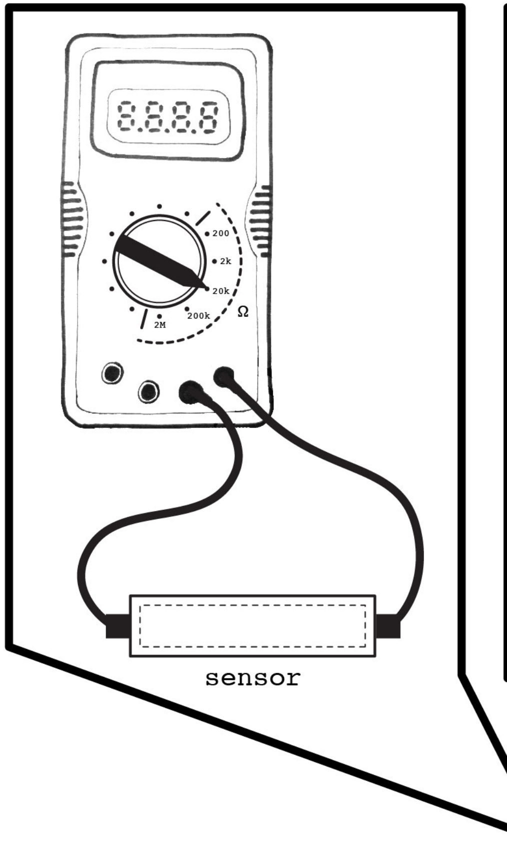
EXTERNAL PULLUP



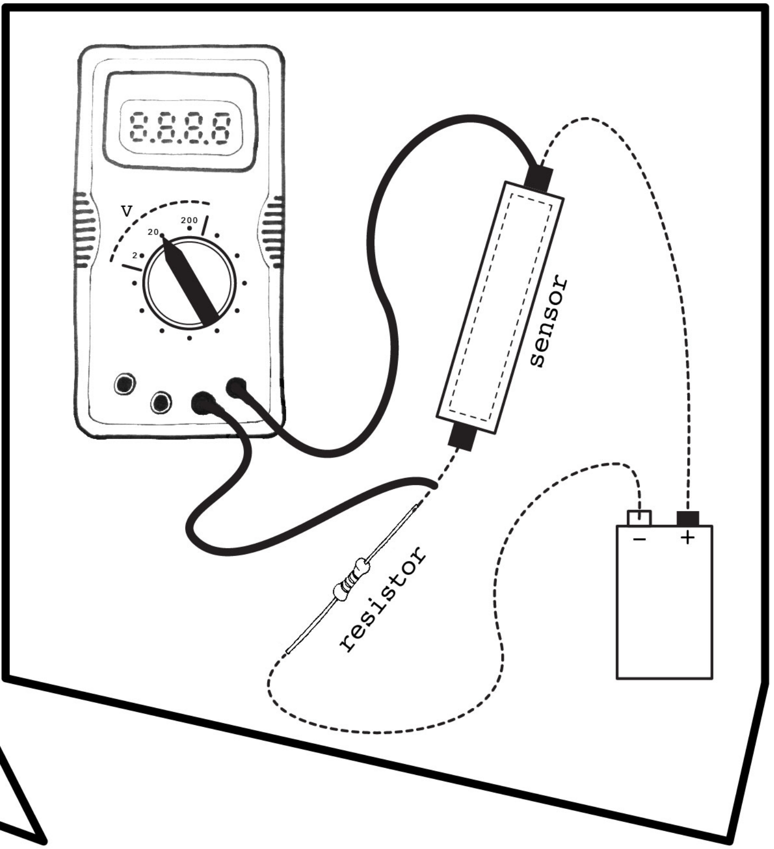
INTERNAL PULLUP (20K)



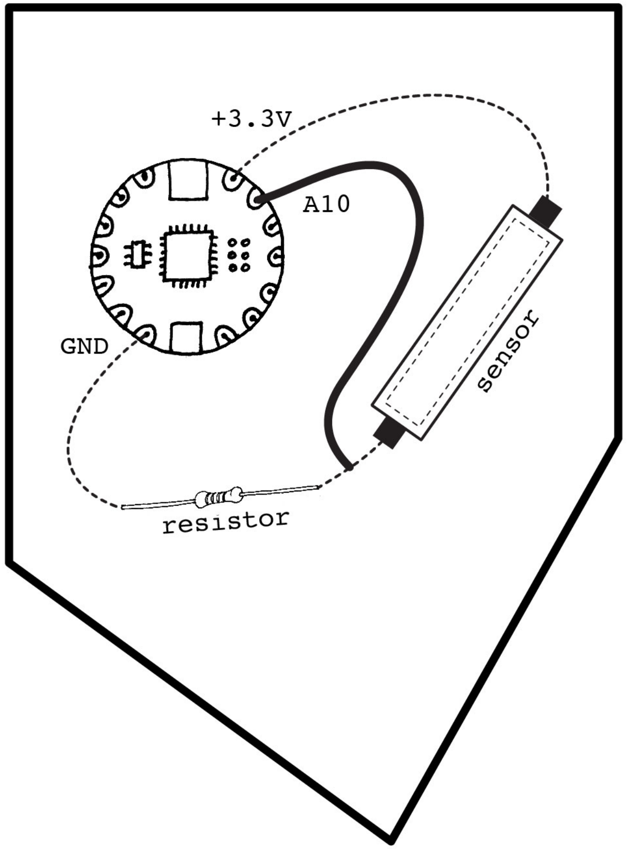
sensor's resistance range



sensor's voltage range

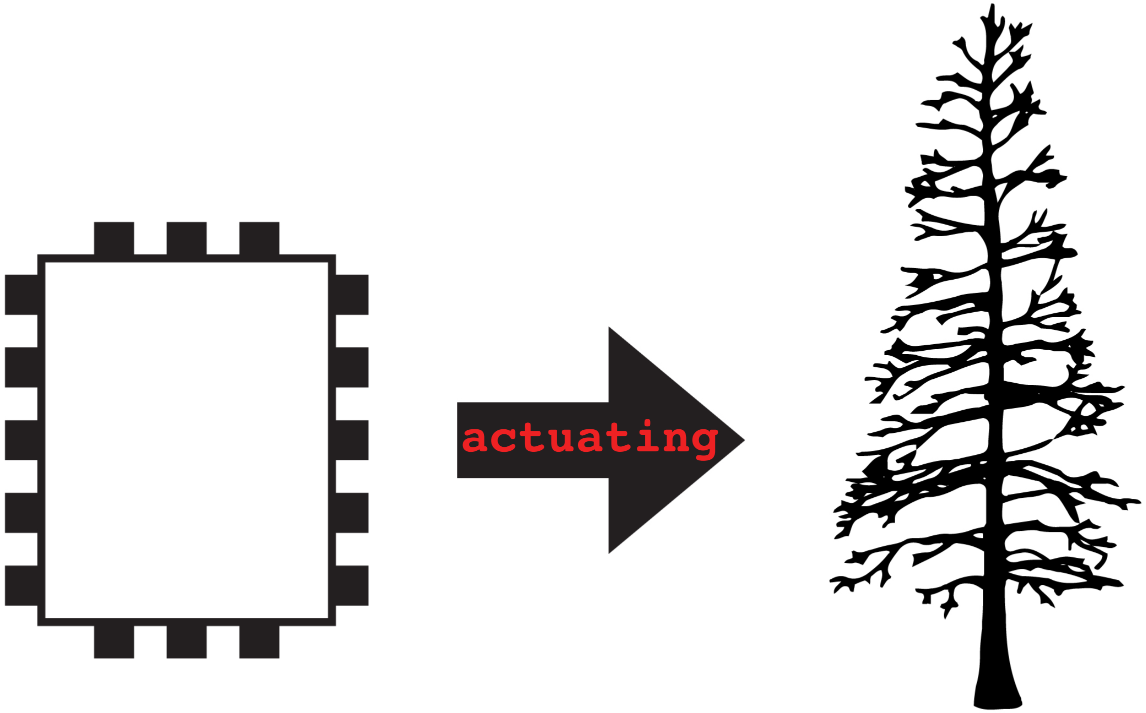


sensor's 10bit ADC range



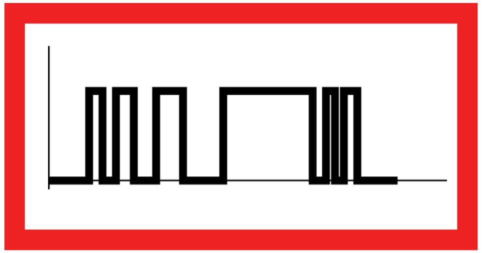
sensor	max Ω resting	min Ω activated	mean Ω resistor	max V resting	min V activated	max ADC resting	min ADC activated

QUESTIONS?

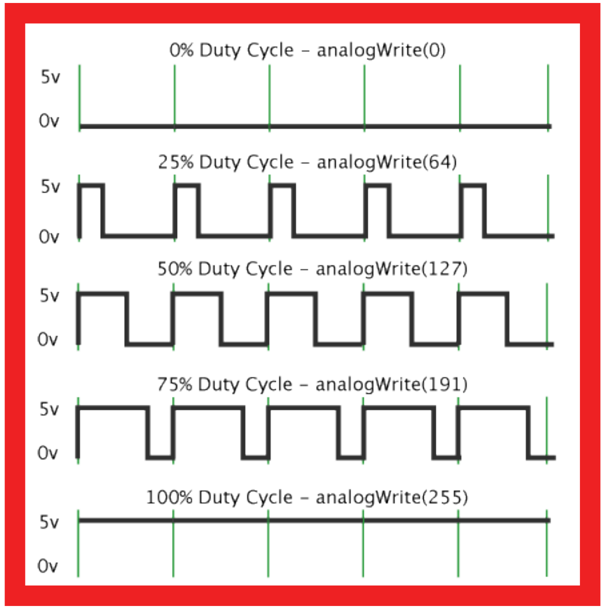


actuating

HIGH, LOW

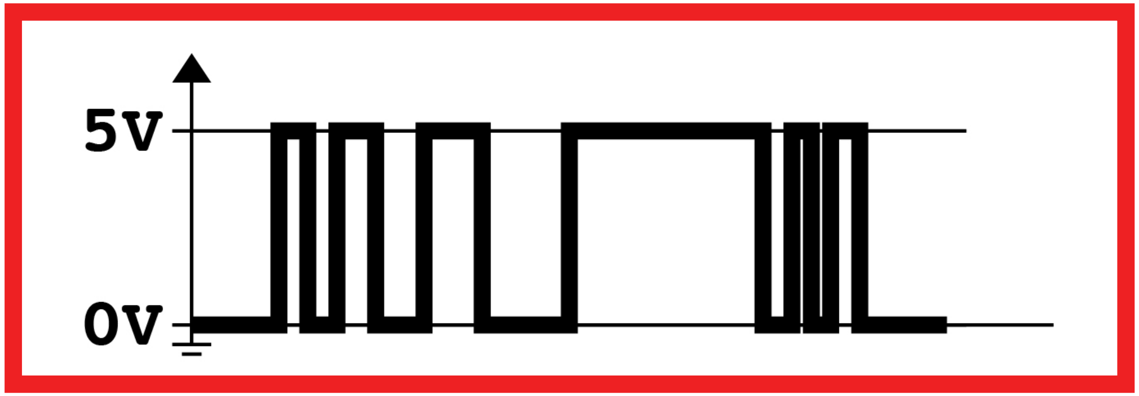


PWM



digital -----------> HIGH / LOW

```
digitalWrite(PIN#, HIGH);
```



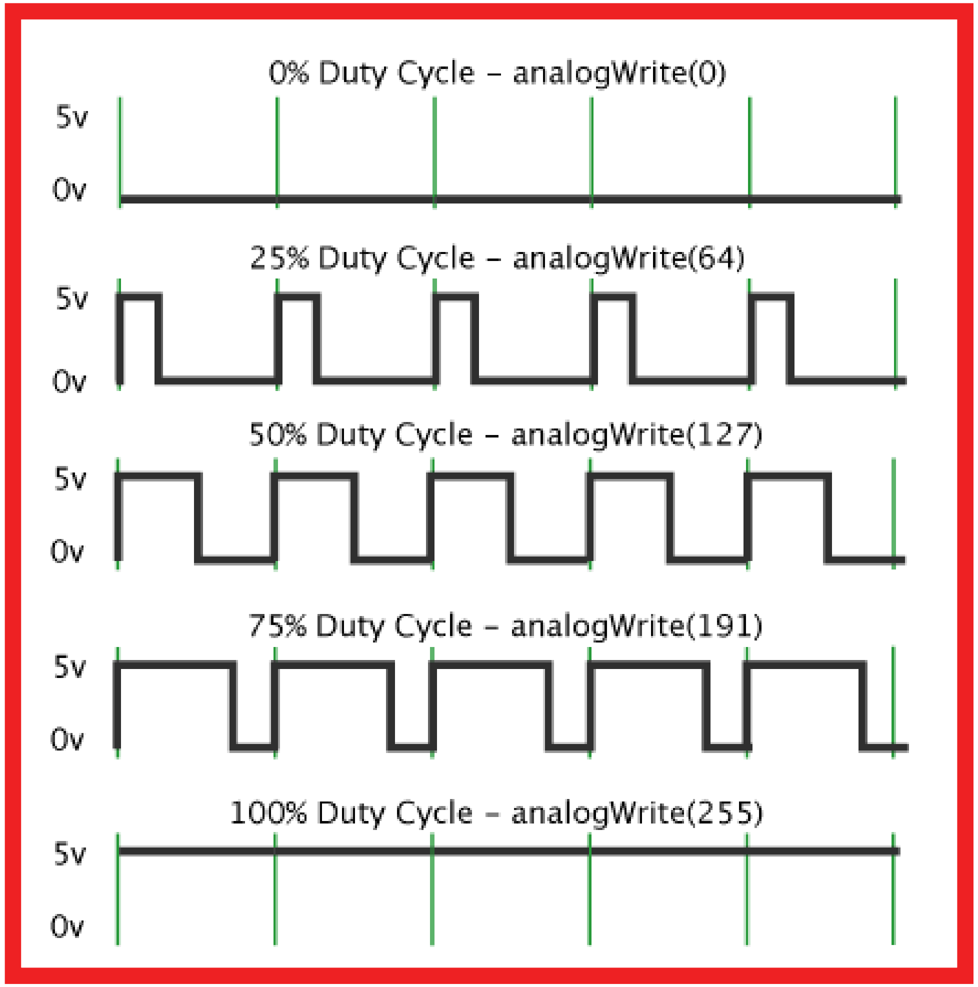
HIGH - 1 - VCC - on

LOW - 0 - GND - off

```
digitalWrite(PIN#, LOW);
```

digital ---------> PWM *"fake analog"*

```
analogWrite(PIN#, [0-255]);
```


0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

PULSE
WIDTH
MODULATION