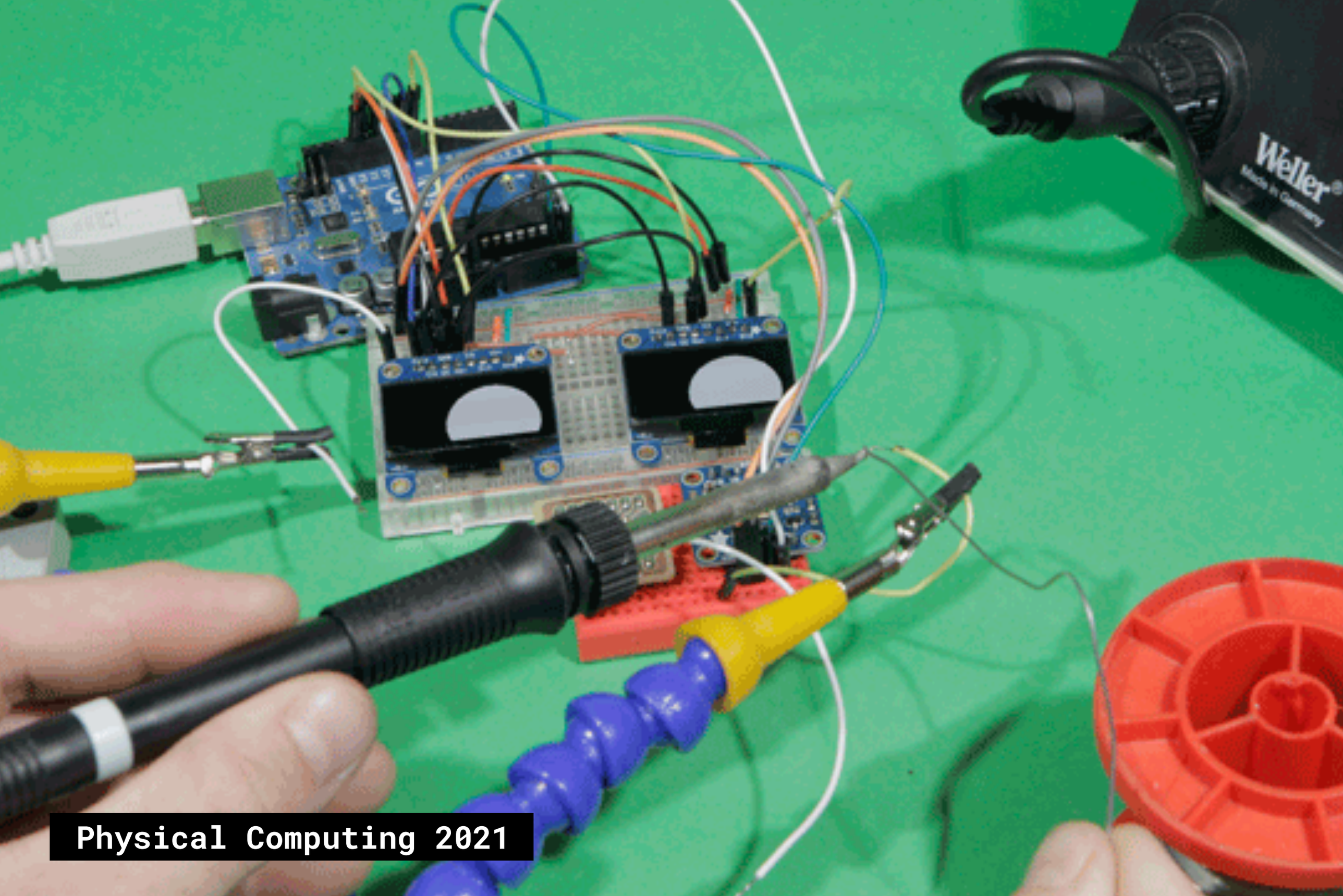
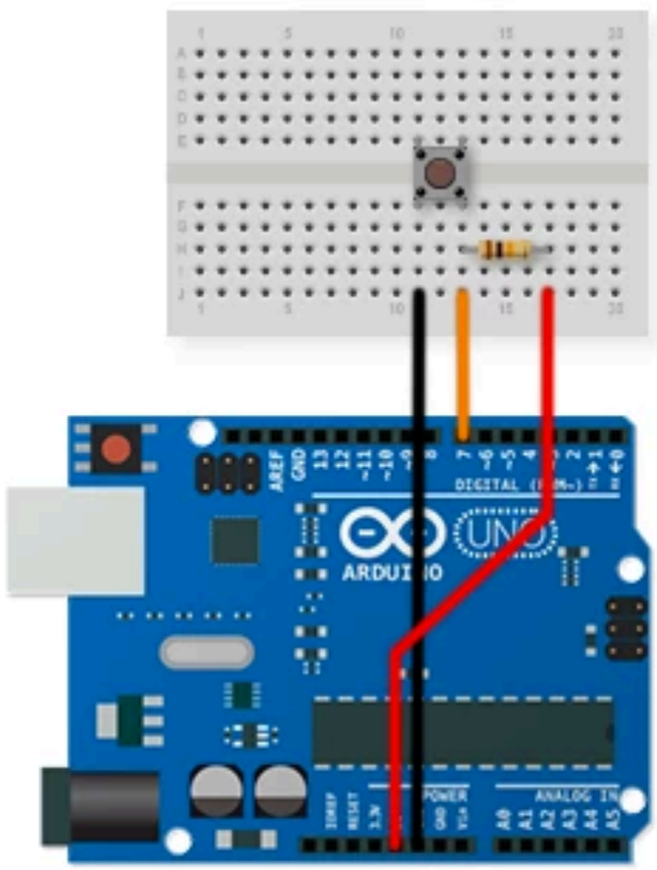
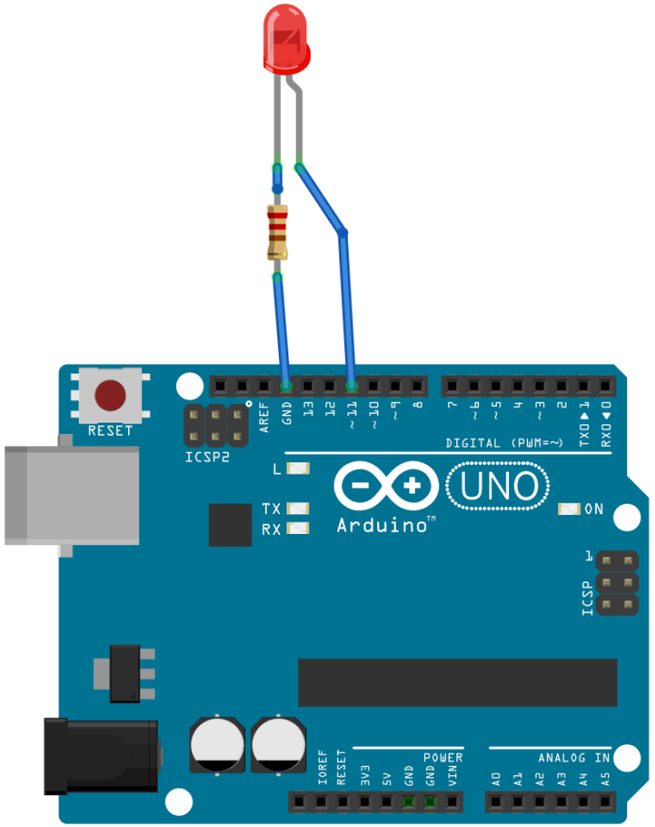



# Physical Computing 2021

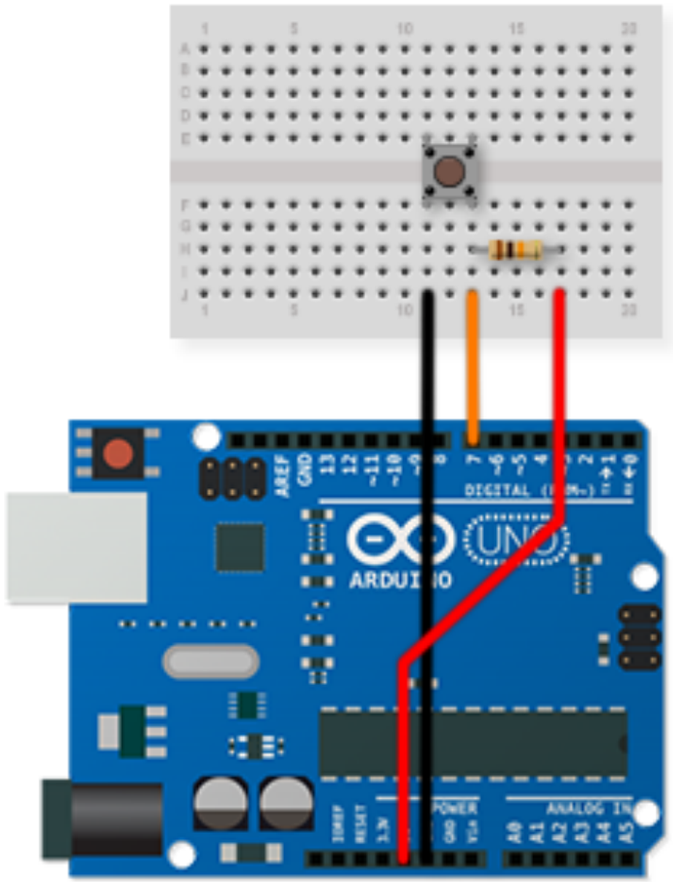




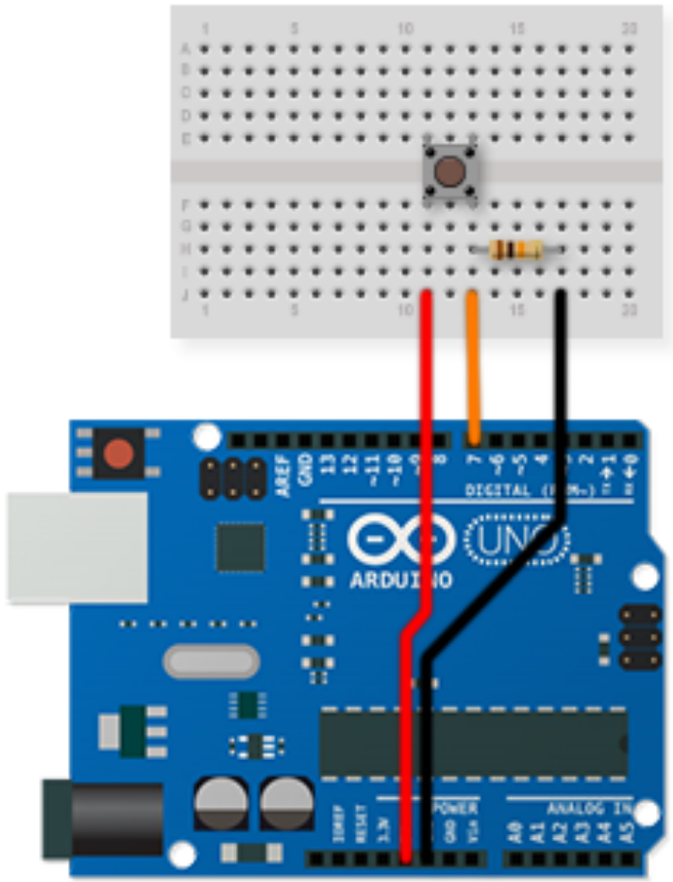
# Exercise 2.3: Digital Input

**Combine the two circuits to create an led you can turn on with your switch**

## Pull Up



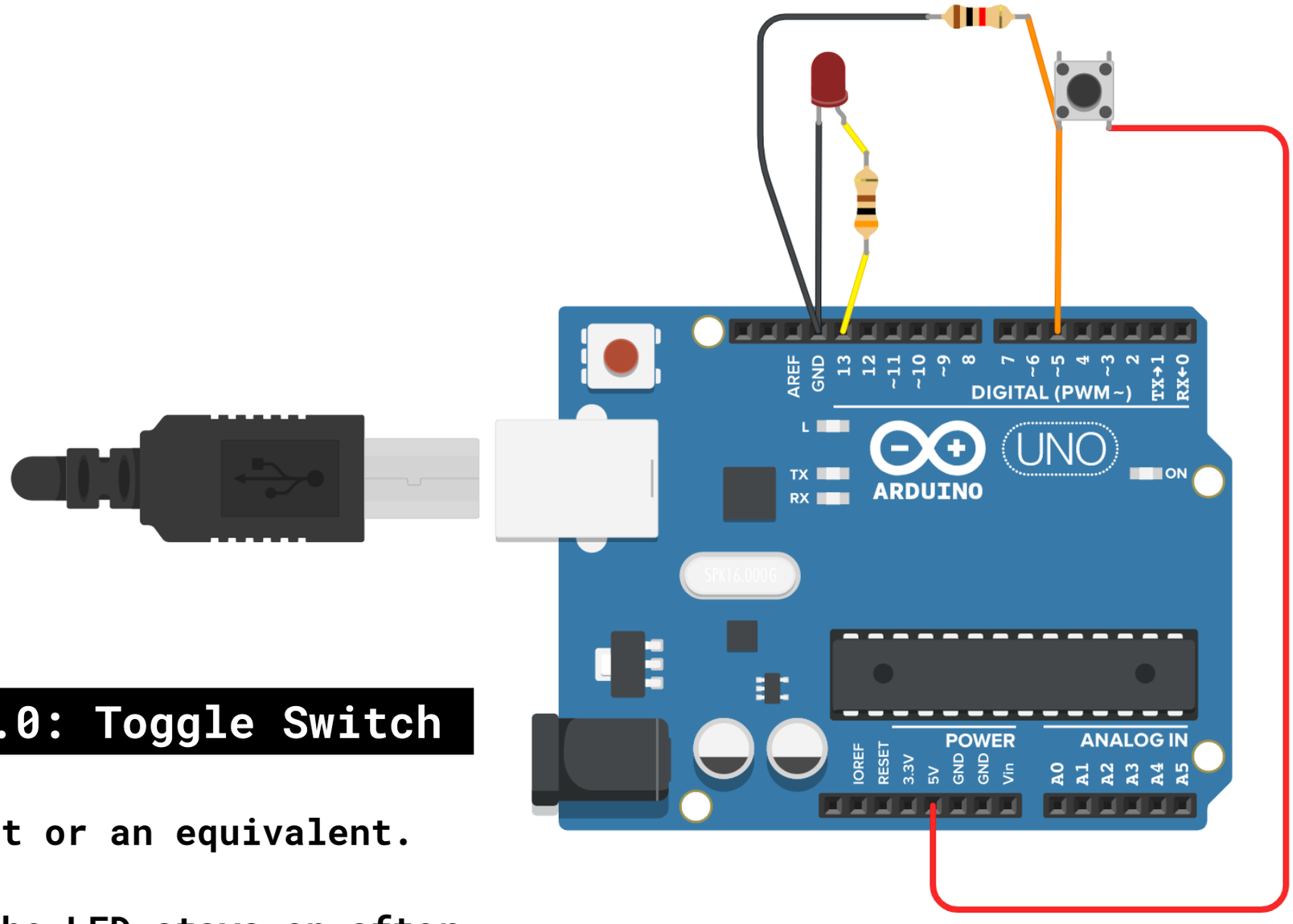
## Pull Down



## Internal Pull Up

```
pinMode(5, INPUT_PULLUP)
```

## Digital Inputs



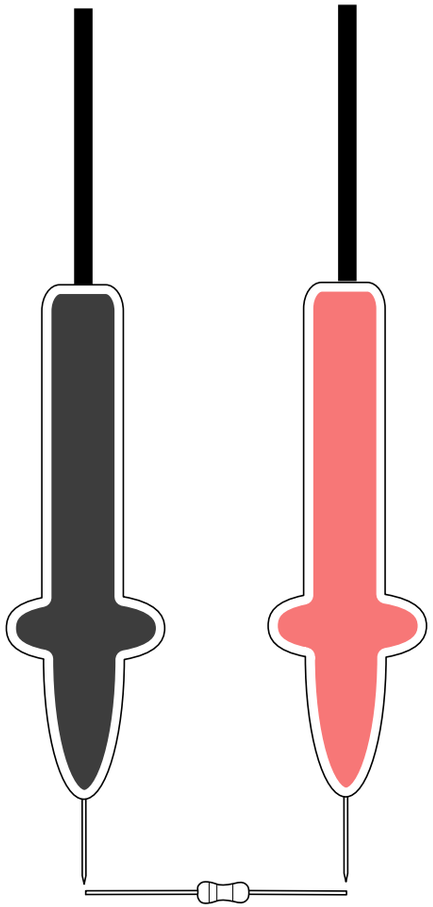

## Exercise PC1.0: Toggle Switch

Build this circuit or an equivalent.

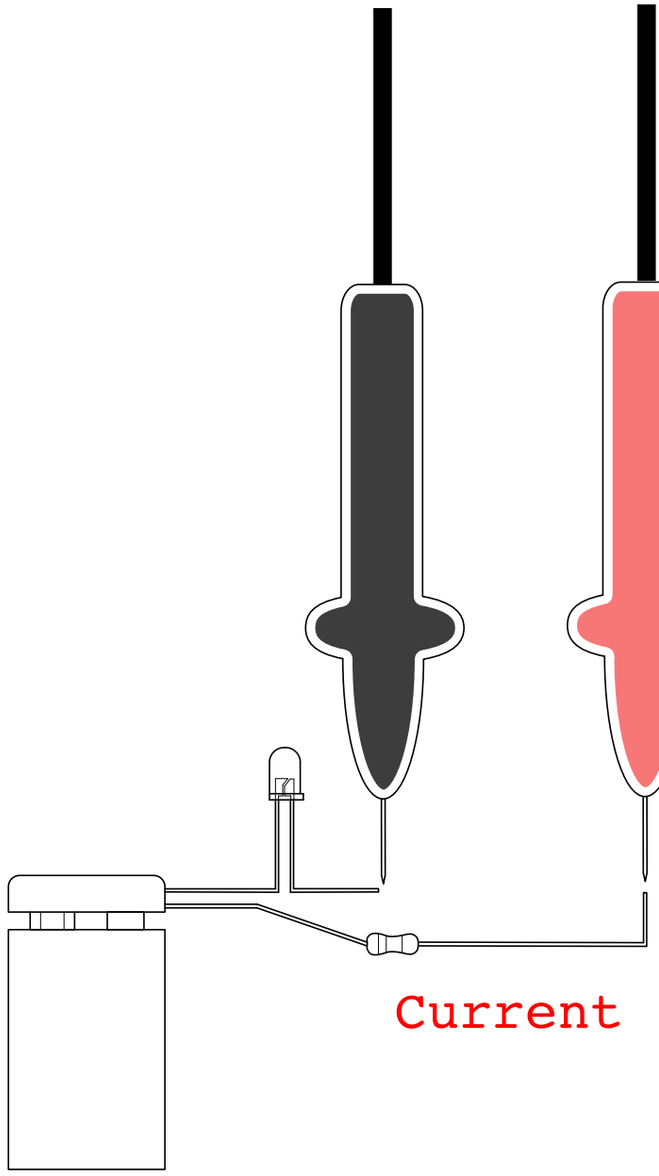
Code it so that the LED stays on after one press, and then turns off after the next press i.e a toggle switch.

The multimeter is an essential tool for problem solving in electronics!

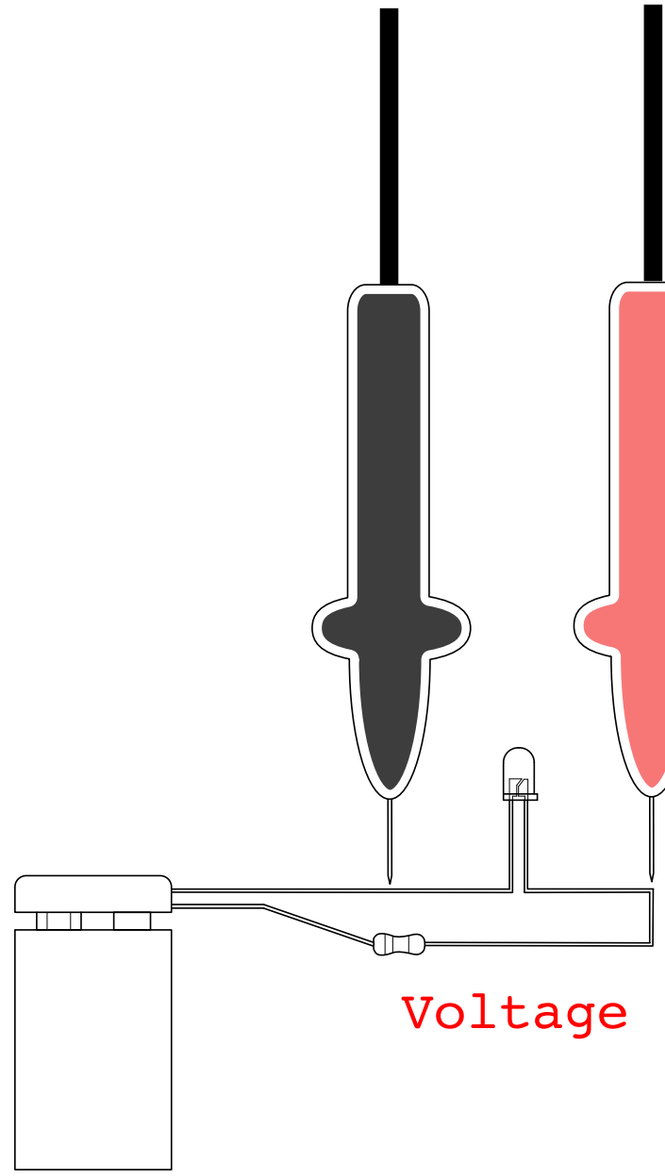
# The Multimeter



Resistance



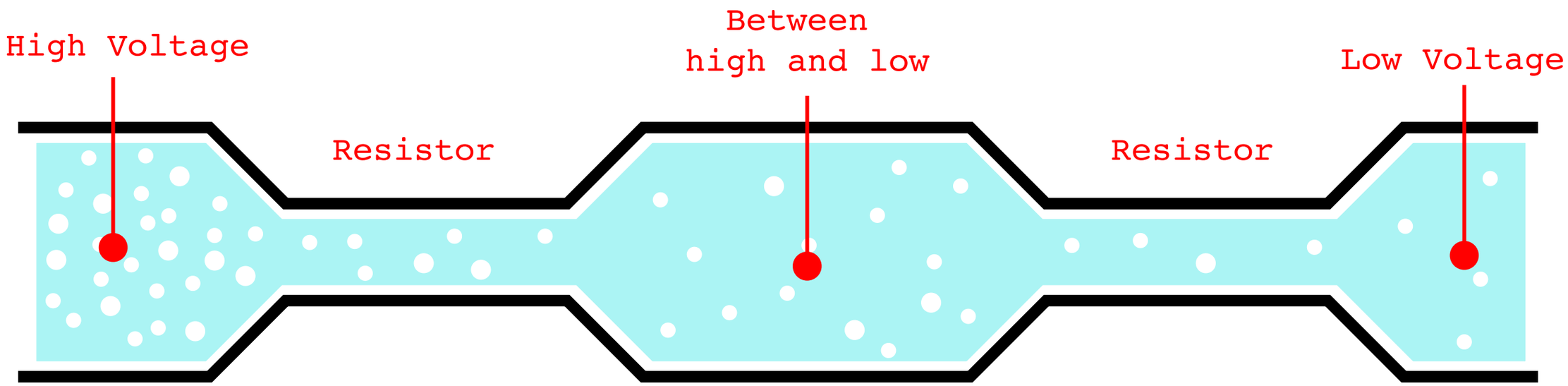
Current



Voltage

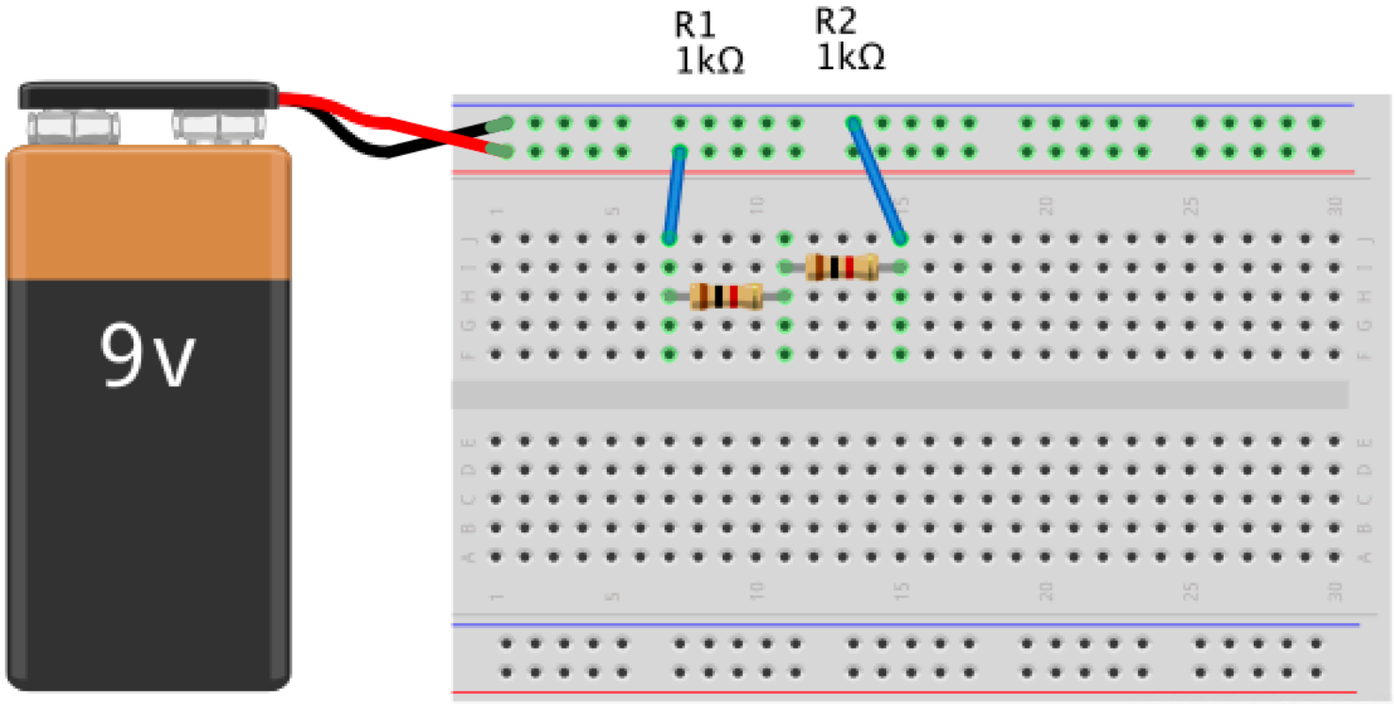
**The way we use the probes on a multimeter depends on what we are testing! Resistance of components should (ideally) be tested outside its circuit. For current we have to break the circuit, and insert the probes to close it again. Voltage, however, can be measure between any two points.**

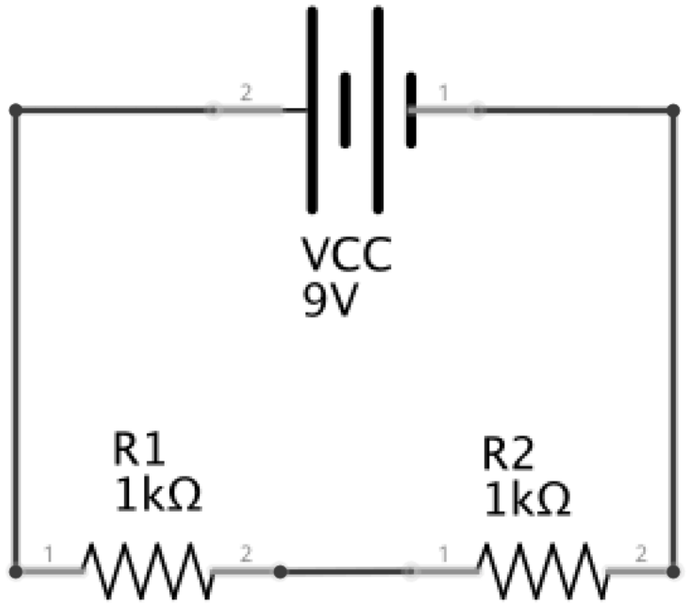
## The Multimeter




**Voltage will drop when current flow though any component that converts electricity to some other form of energy.**

## Voltage Divider



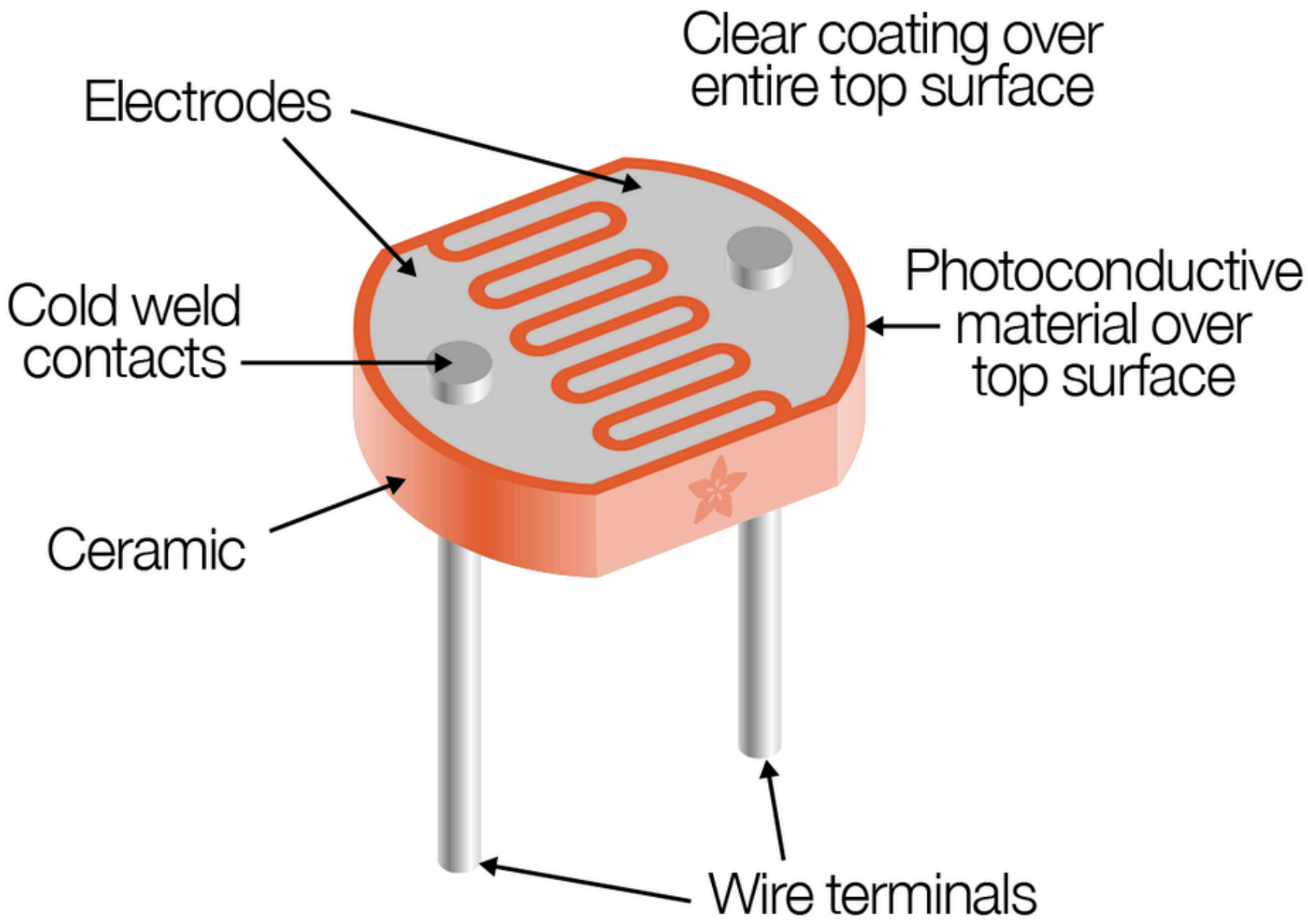




## Exercise PC1.1: Voltage Divider

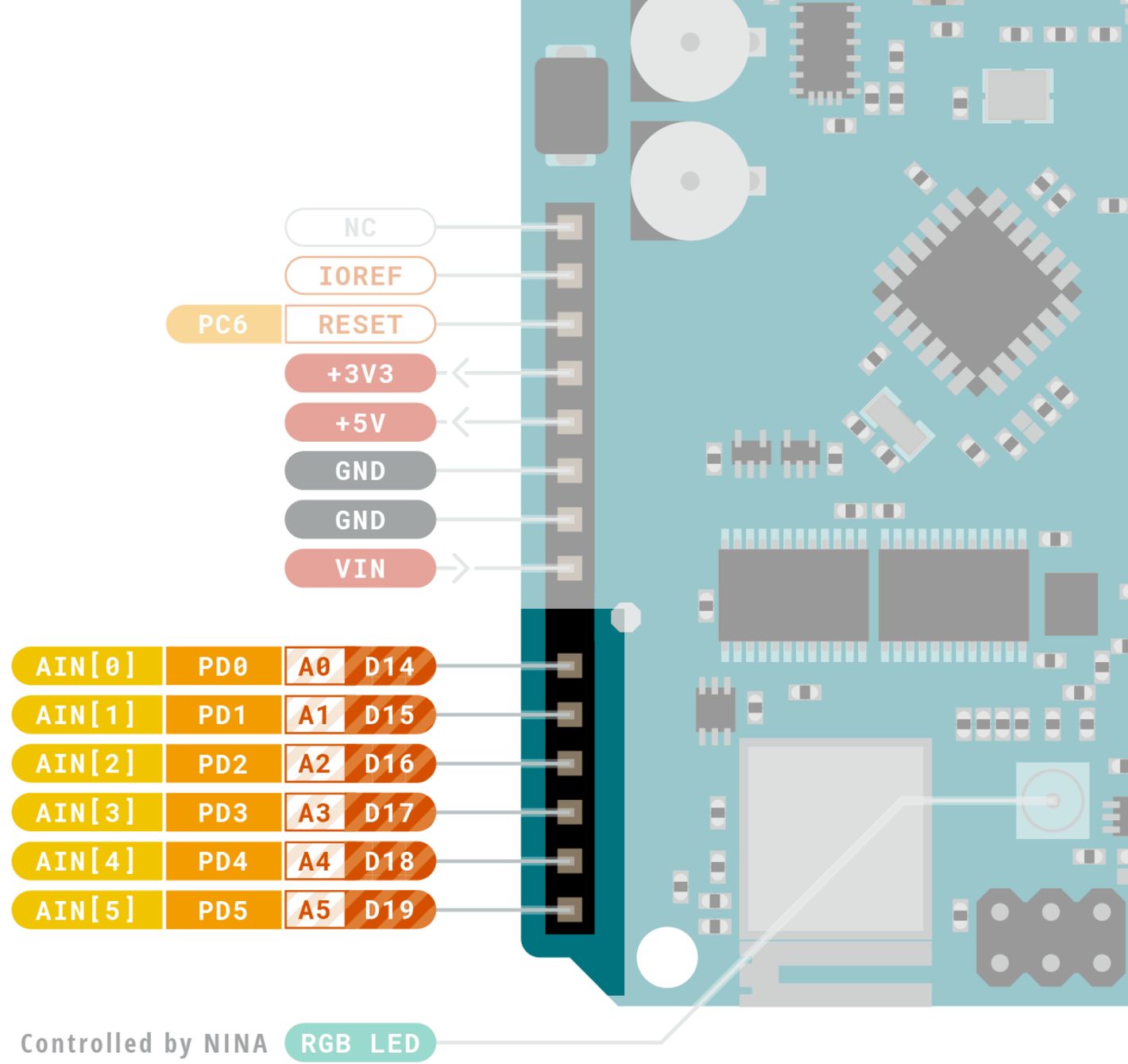
Assembly the circuit with resistors of two different values. Try calculating the voltage between your resistors, and check with the multimeter if it's correct.

$$Vres = Vcc \times \frac{R2}{(R1 + R2)}$$

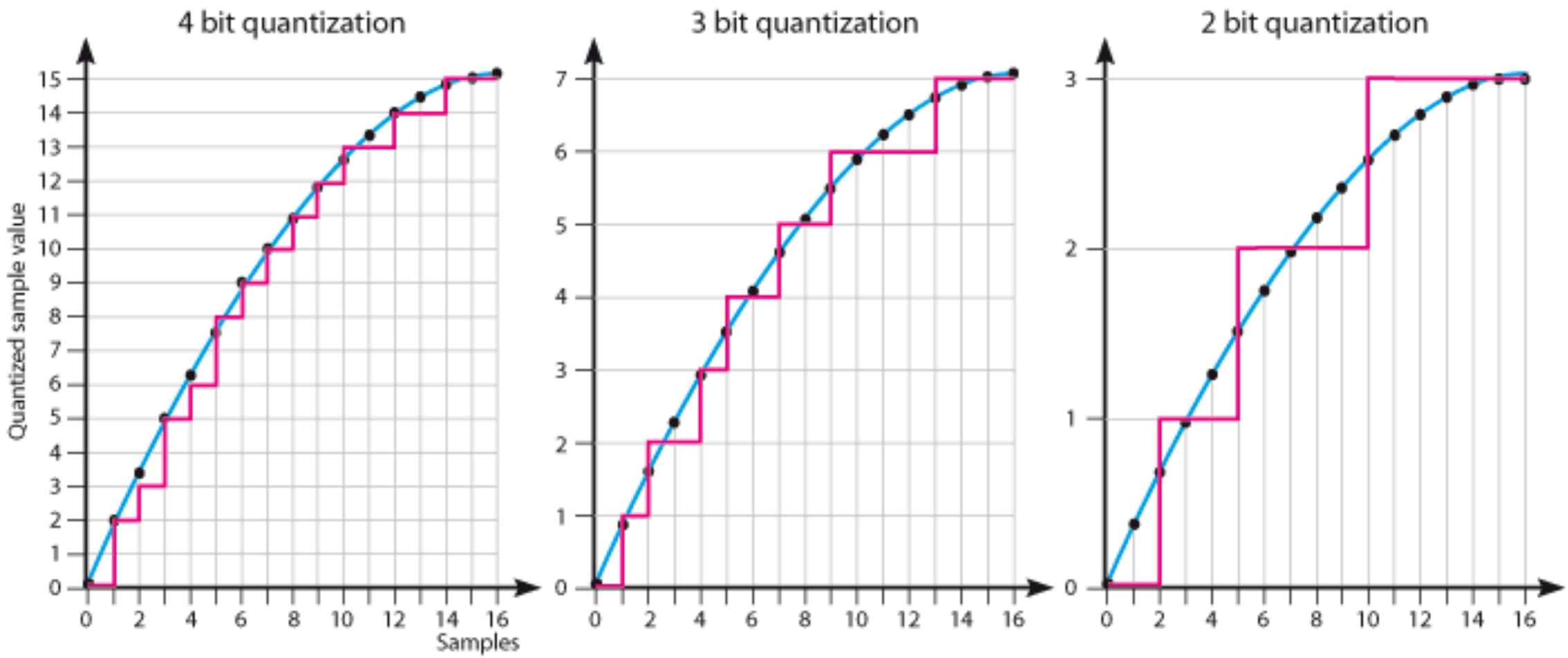
Vres = resulting voltage
Vcc = applied voltage (9V)
R1 = first resistor (1000 ohm)
R2 = second resistor (1000 ohm)




## Analog Sensors


NC
IOREF
PC6
RESET
+3V3
+5V
GND
GND
VIN
AIN[0] PD0 A0 D14
AIN[1] PD1 A1 D15
AIN[2] PD2 A2 D16
AIN[3] PD3 A3 D17
AIN[4] PD4 A4 D18
AIN[5] PD5 A5 D19
Controlled by NINA RGB LED

## Analog Pins

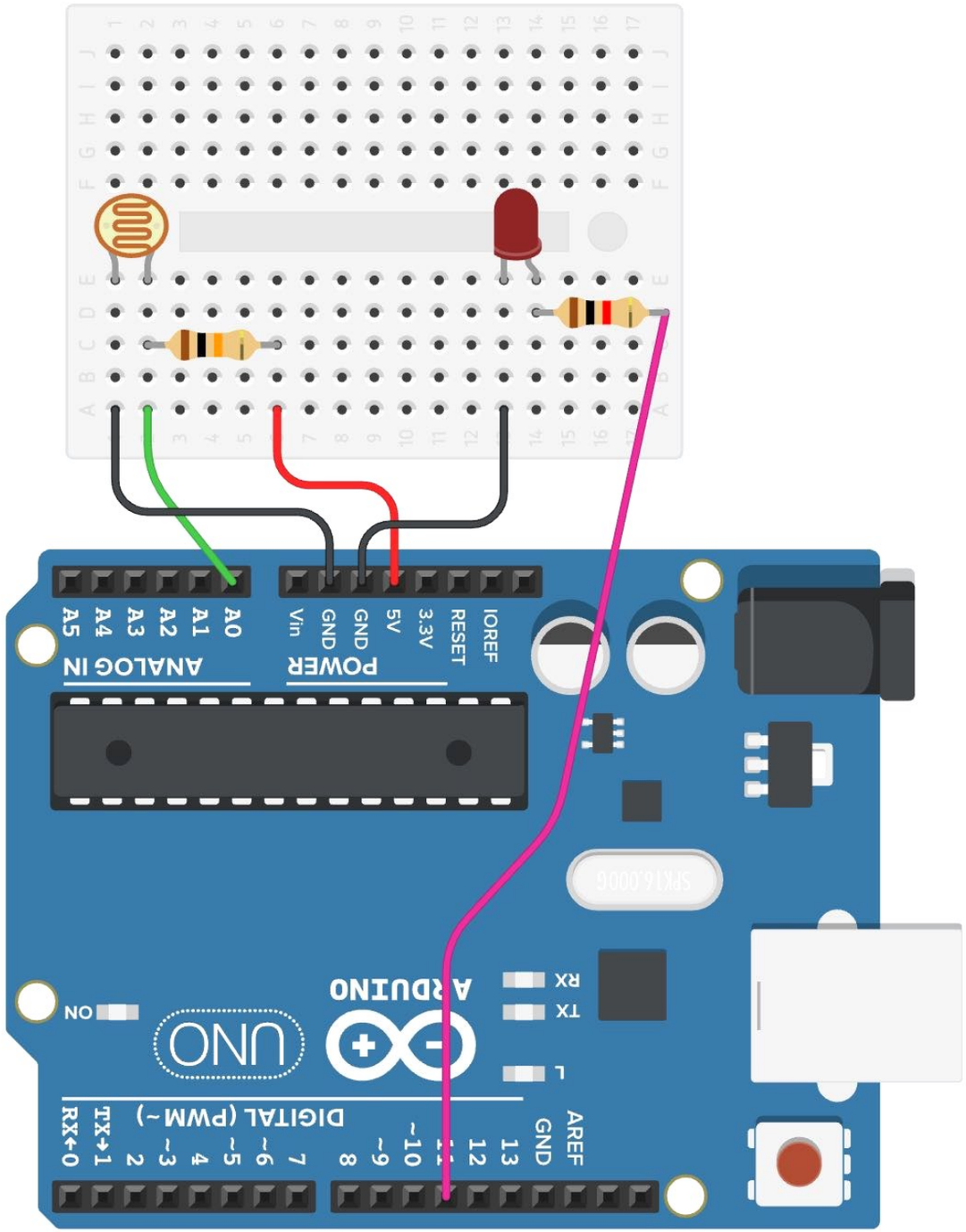



# Analog to Digital Conversion

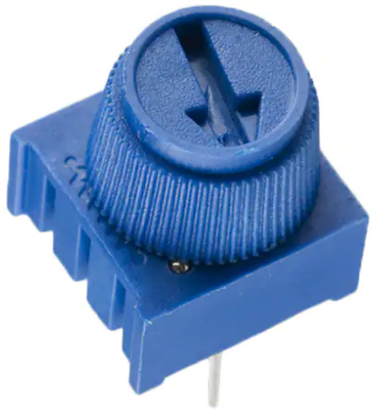
# Exercise PC1.2: Light Sensor

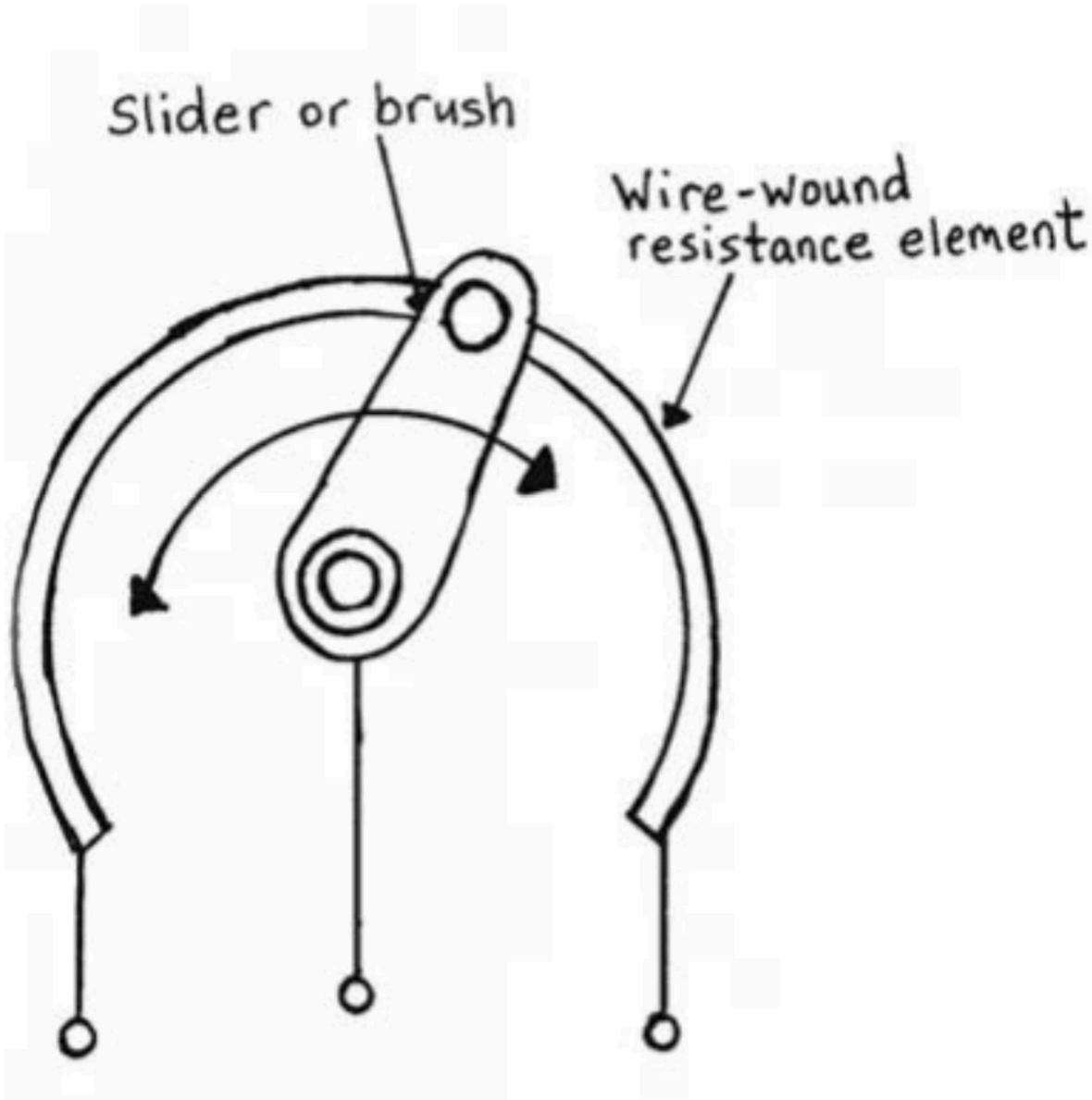
**Build a circuit and code it to turn on an LED when it gets dark using a Photoresistor.**

**Optional: code it so the LED fades smoothly between dark and light states.**



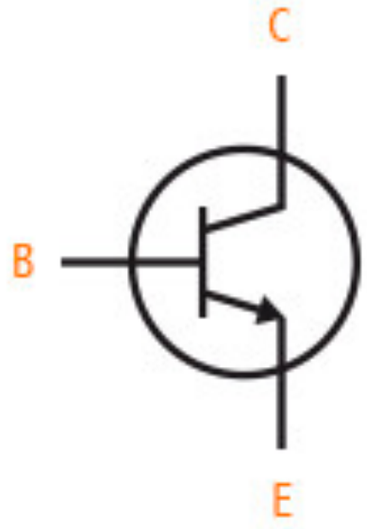
**A Potentiometer is also a voltage divider!**





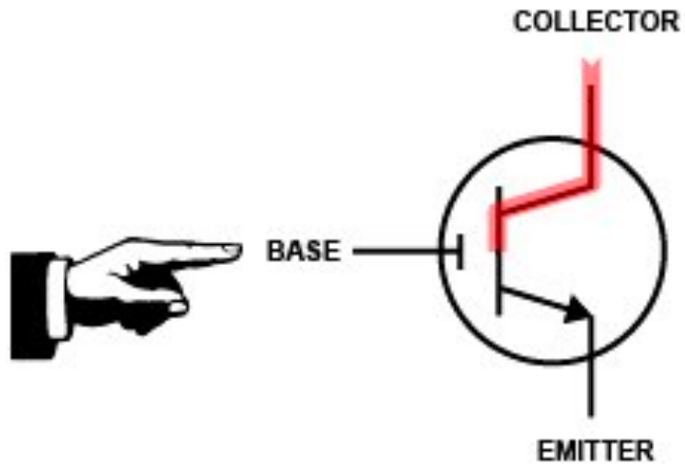

# Potentiometer

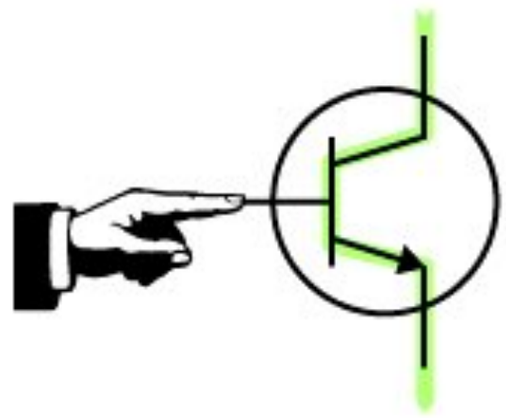
**To power loads greater than the 40mA our Arduino digital pins can provide, we need to use a transistor.**

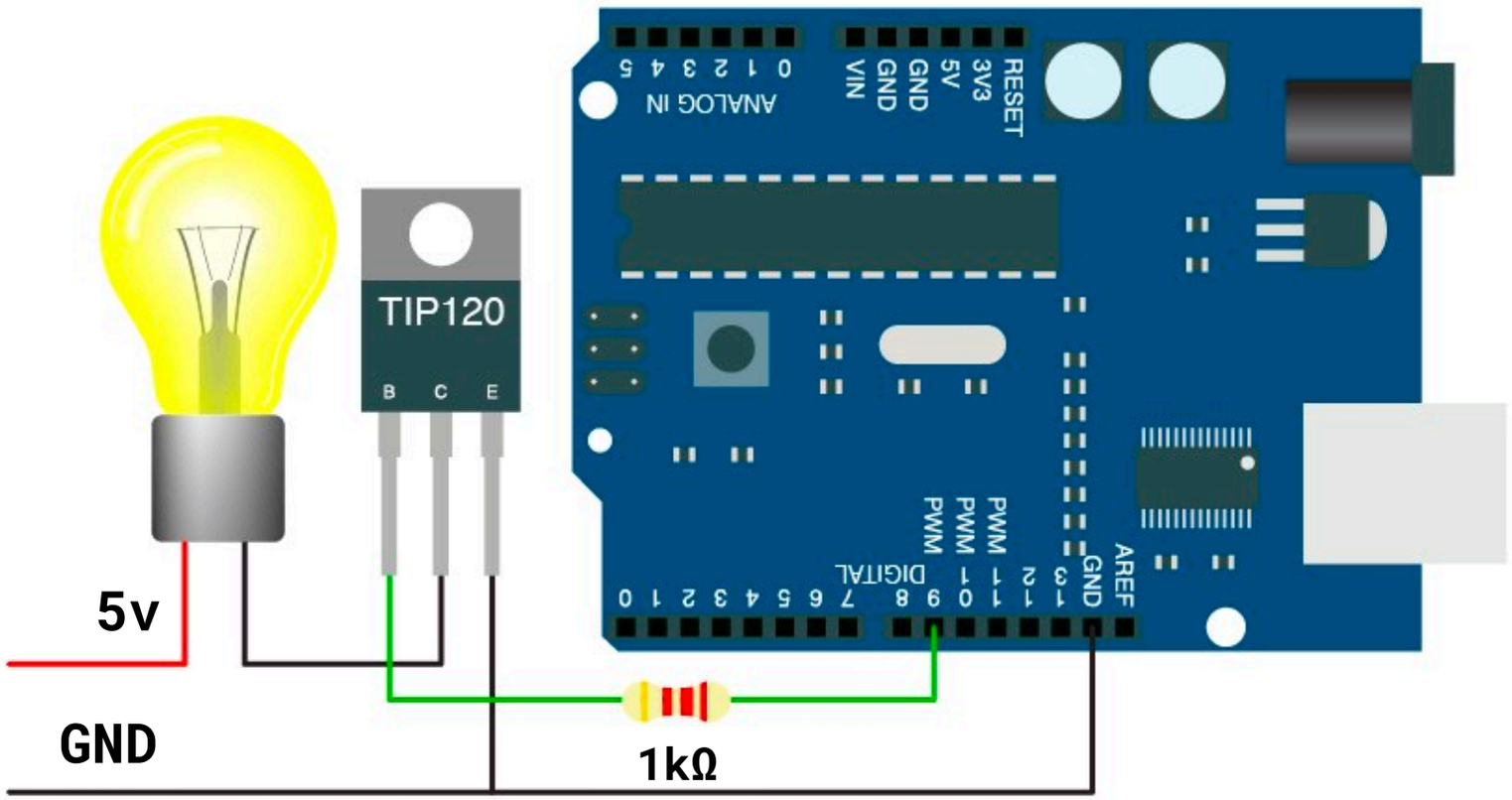

# Transistors

**Transistors work like buttons, with a very small current on the base pin, allowing a larger current to flow between collector and emitter.**



# Transistors



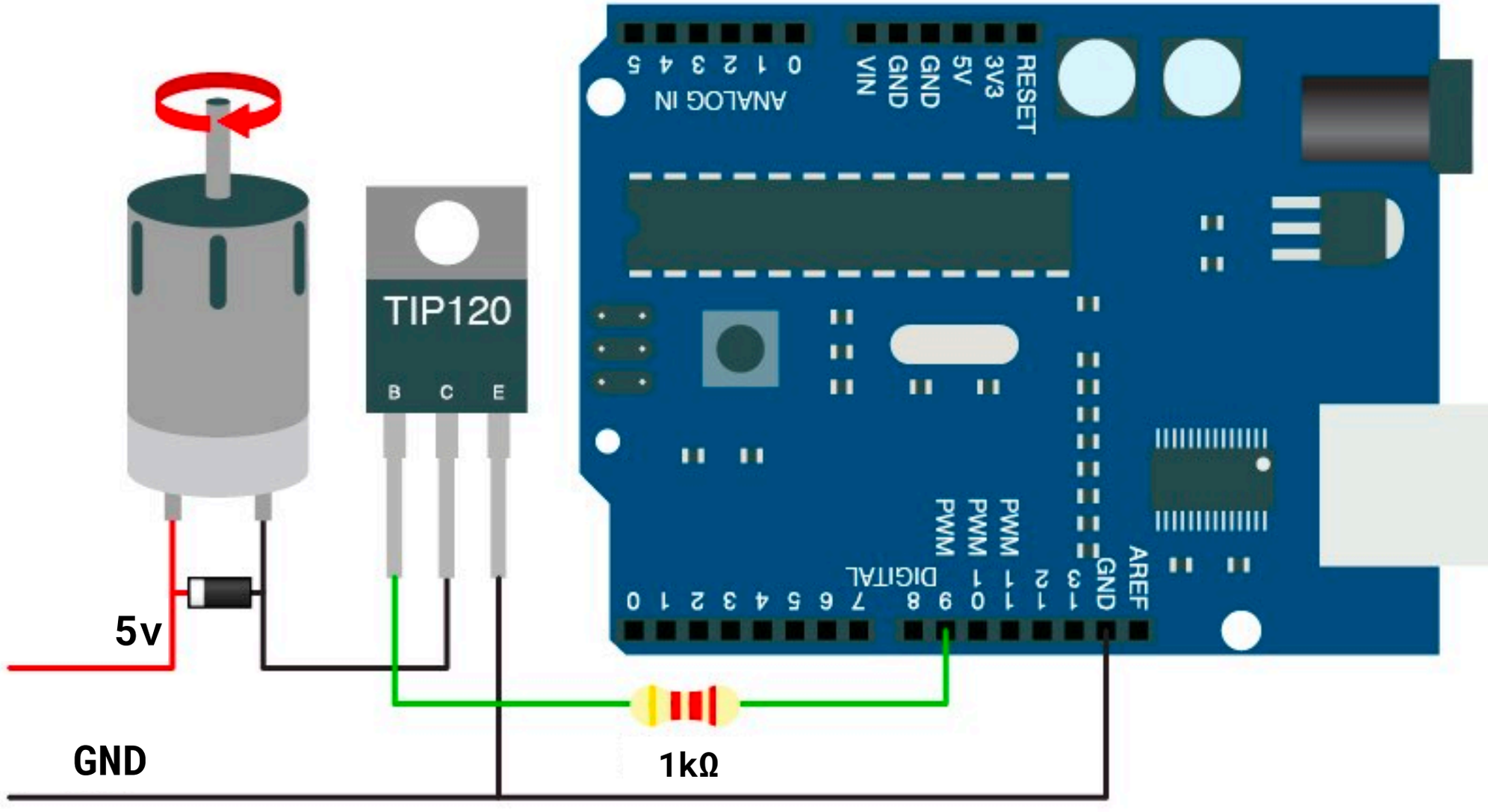

## Exercise PC1.3: Transistor with Lightbulb

Control an incandescent
lightbulb using a transistor.

Can you also use PWM to dim it?

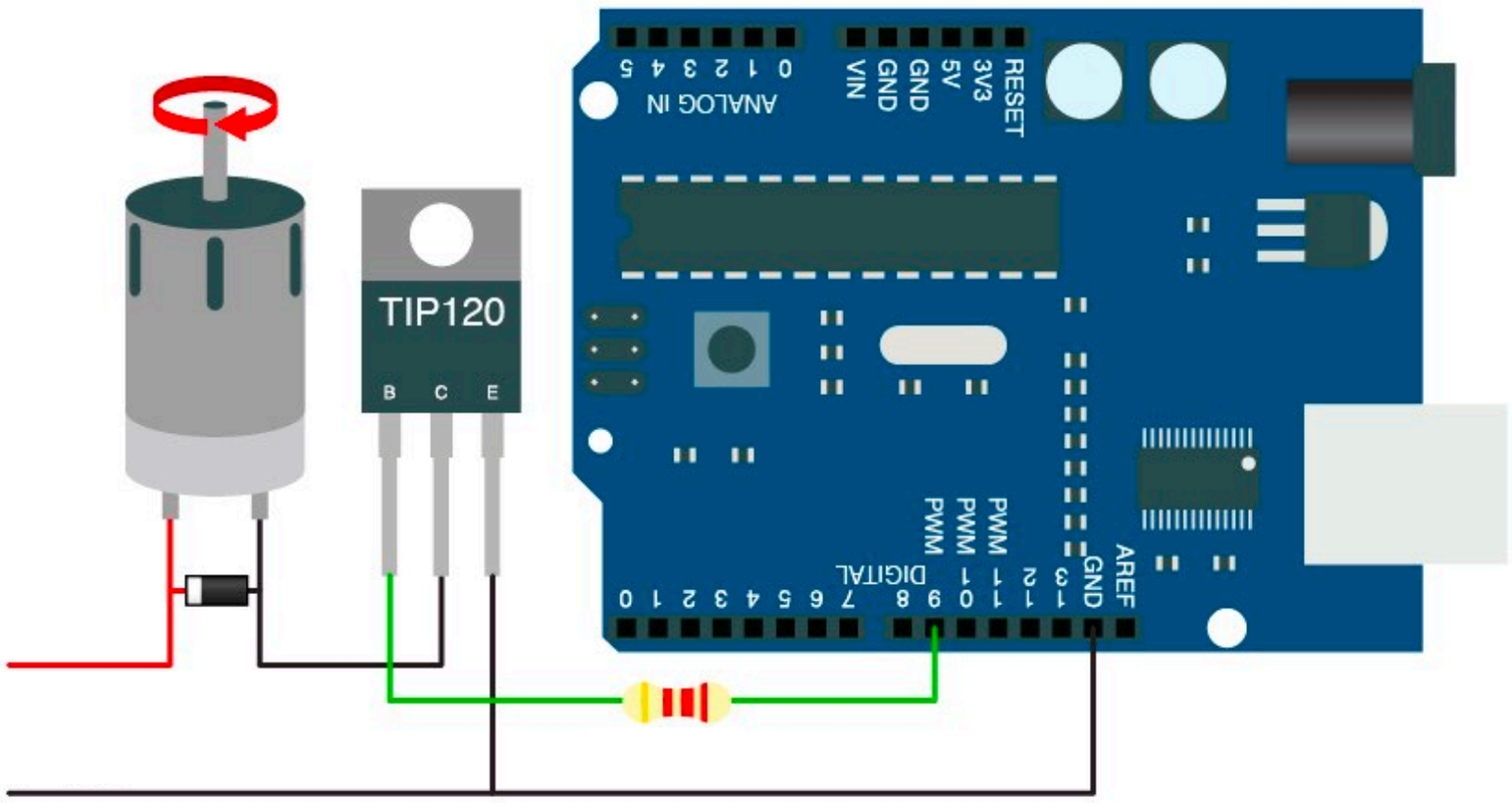
## Transistors

**Adding diode is important for inductive loads (motors for example)**




## Transistors

**Adding diode is important for inductive loads!**

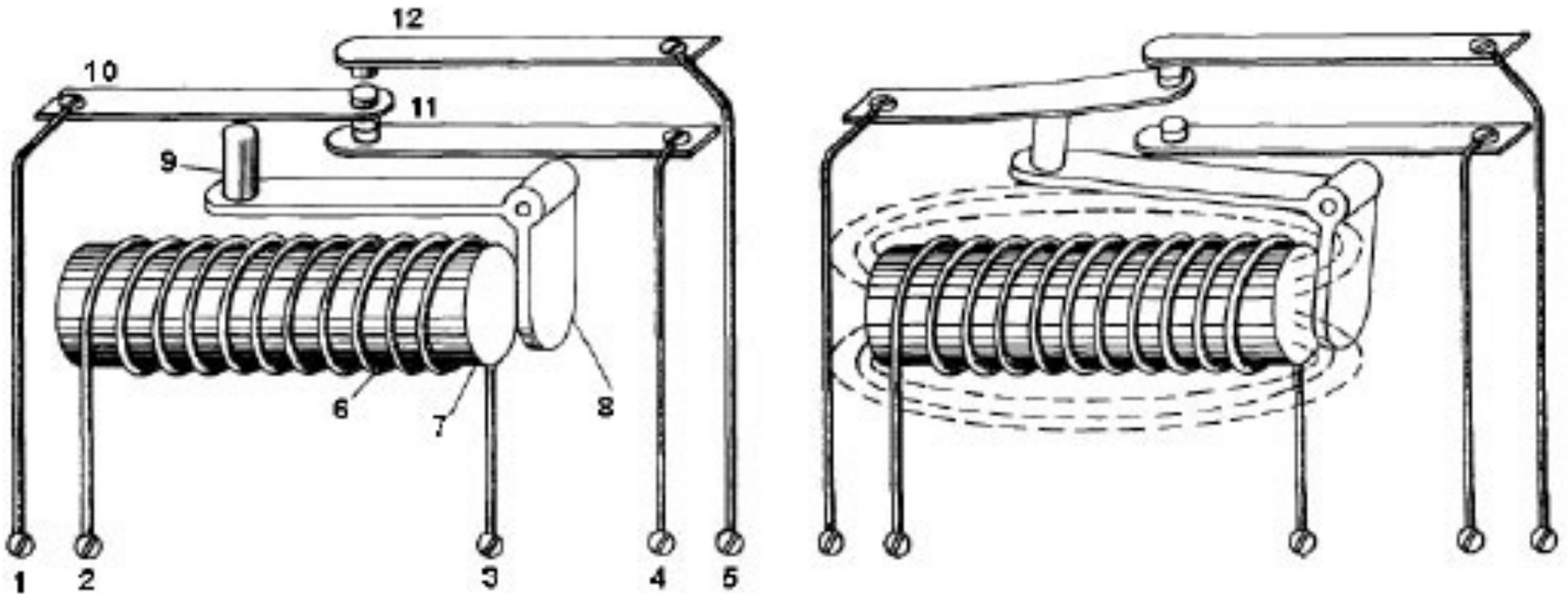



## Exercise PC1.4: Transistor with Motor

Hook up a motor in place of your lightbulb. Add in a potentiometer and write new code to control the motor speed by rotating a potentiometer.

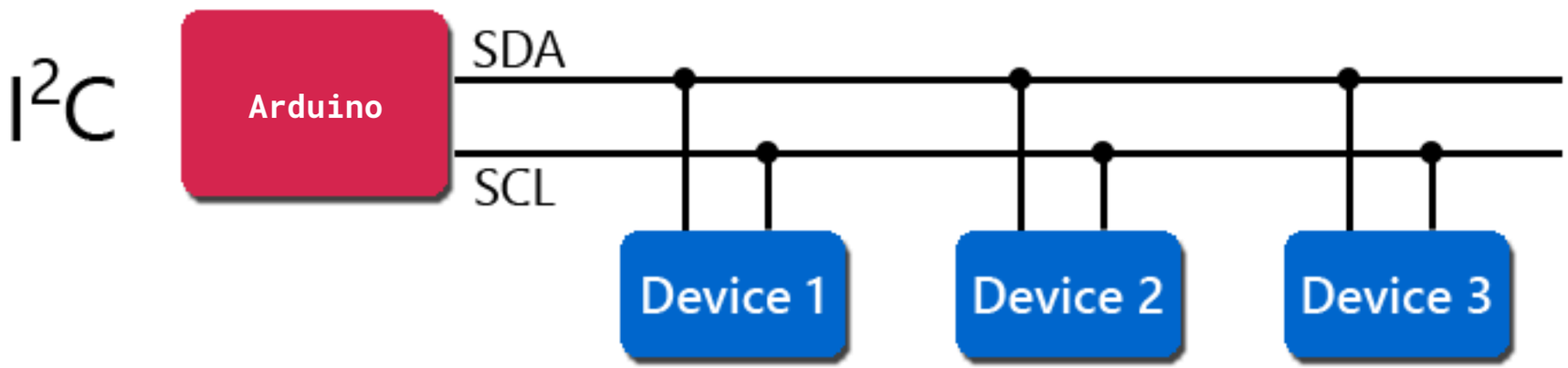
## Transistors

**Relays can sometimes be used instead of transistors (for controlling very high loads, or when the HV circuits need to be completely isolated. But, they are loud and slow!**




# Relays

# Digital Peripherals and Protocols



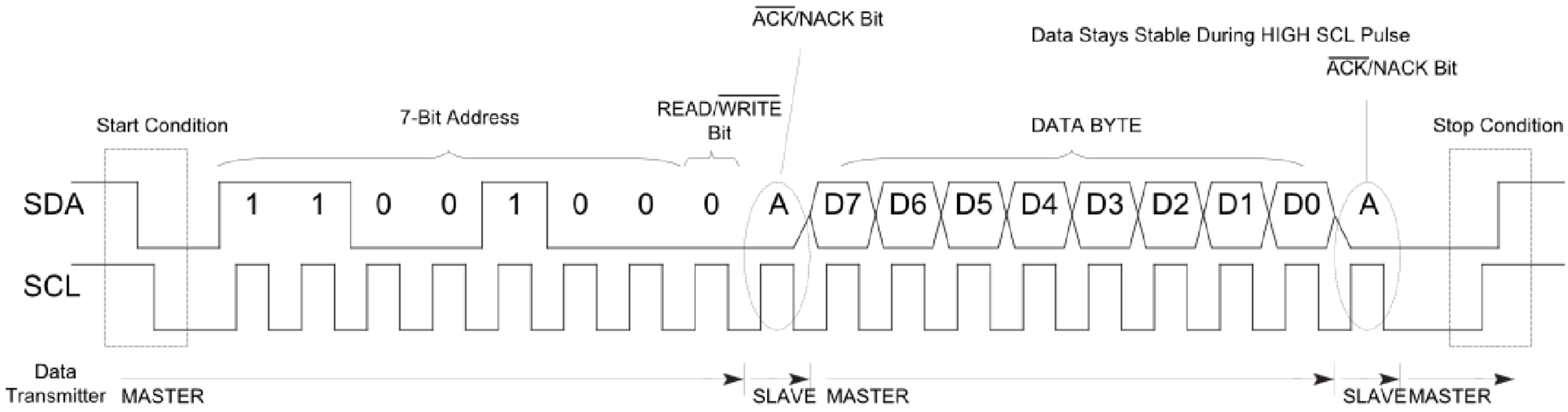

**Uses:**

- **Communicating between Arduino and other chips (sensors or displays for example)**
- **Communicating between several Arduino boards on a small network**

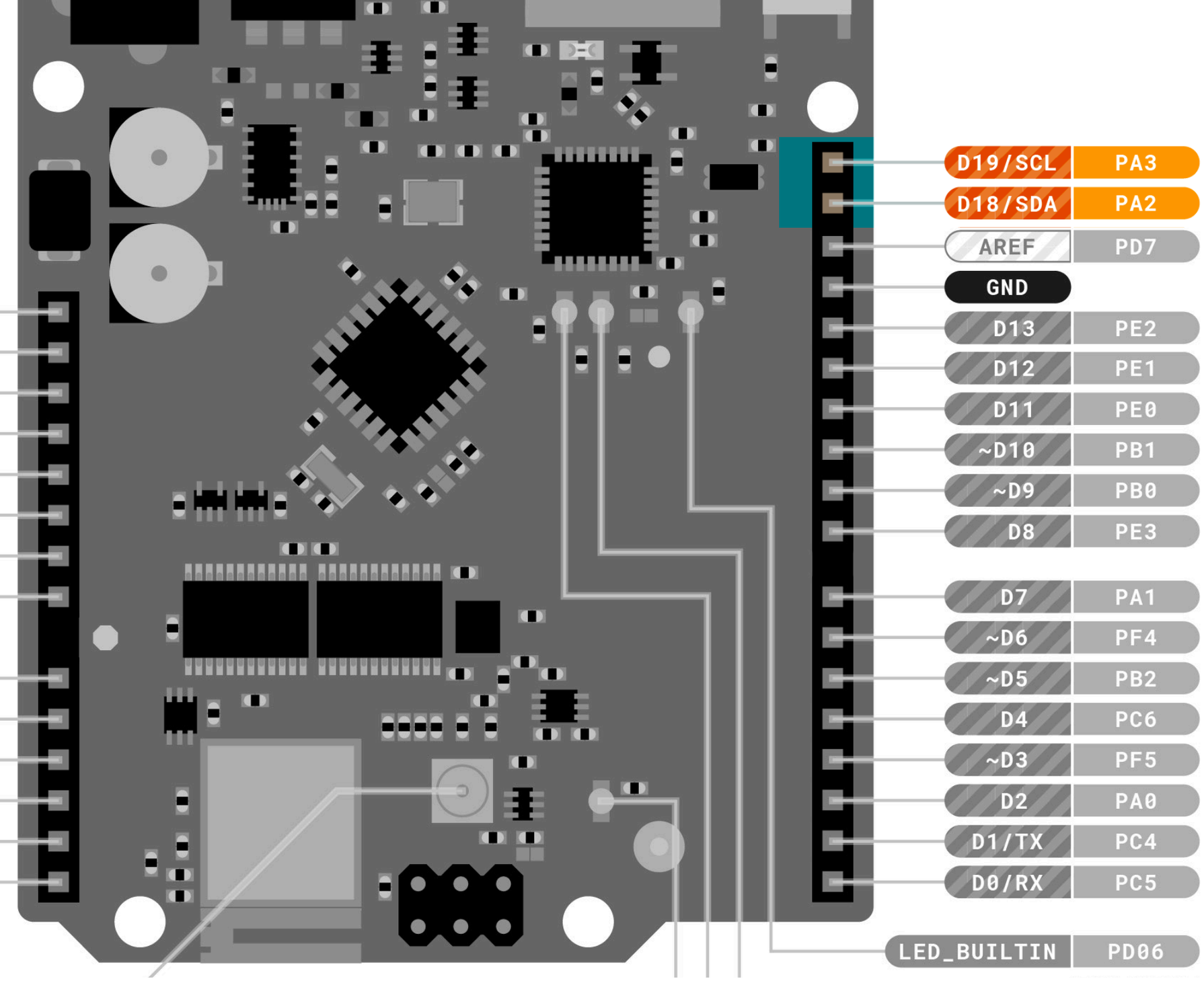
**Limitations:**

- **Fairly slow (probably doesn't matter for our projects)**
- **IC2 only works over short distances**

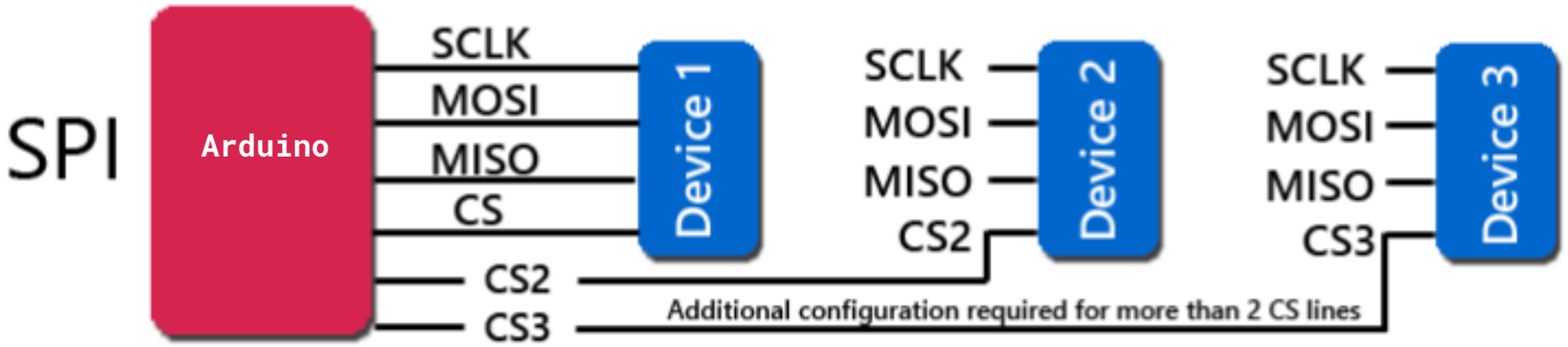
## I²C (Inter Integrated Circuit)




# I²C (Inter Integrated Circuit)



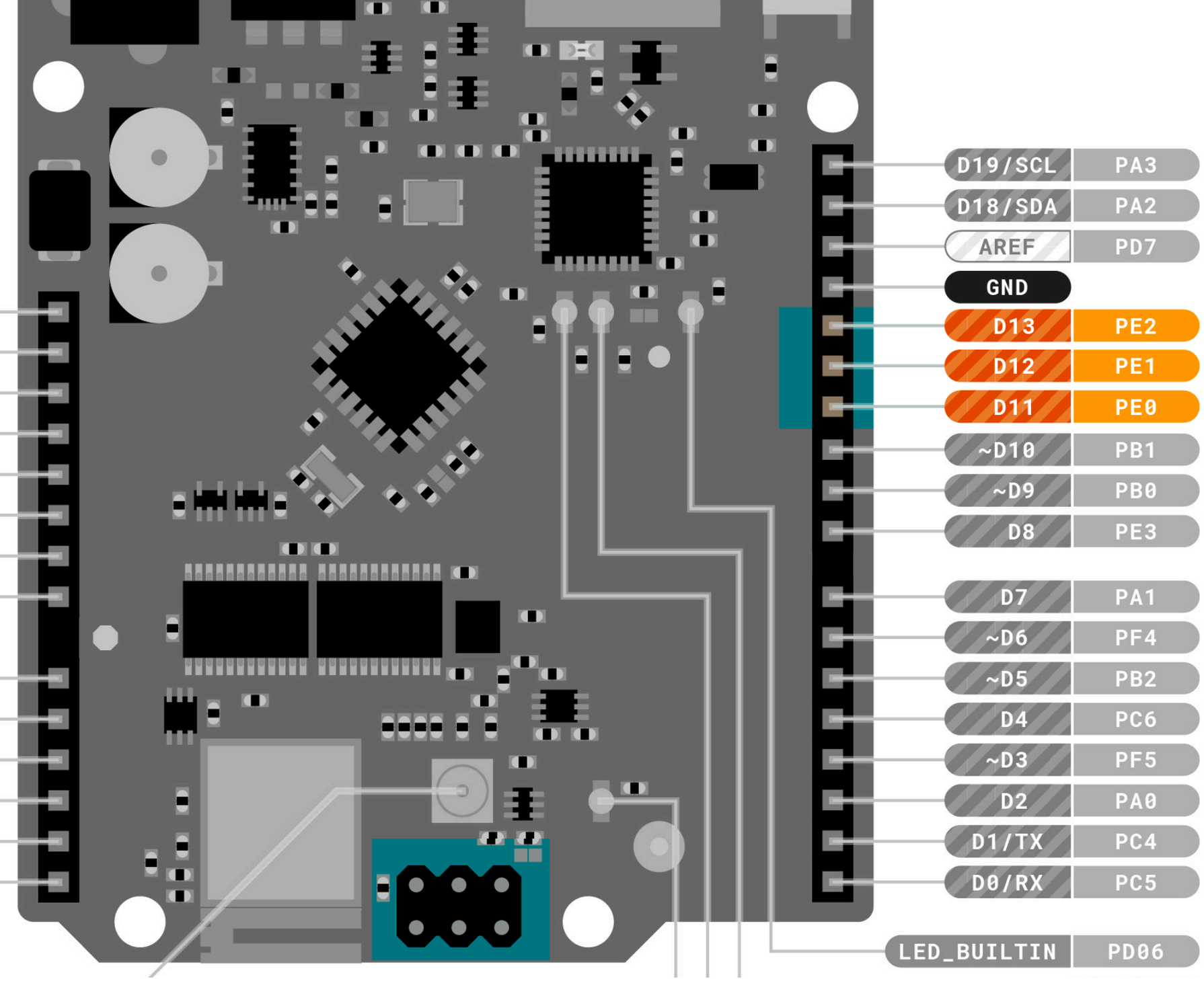

# I²C (Inter Integrated Circuit)



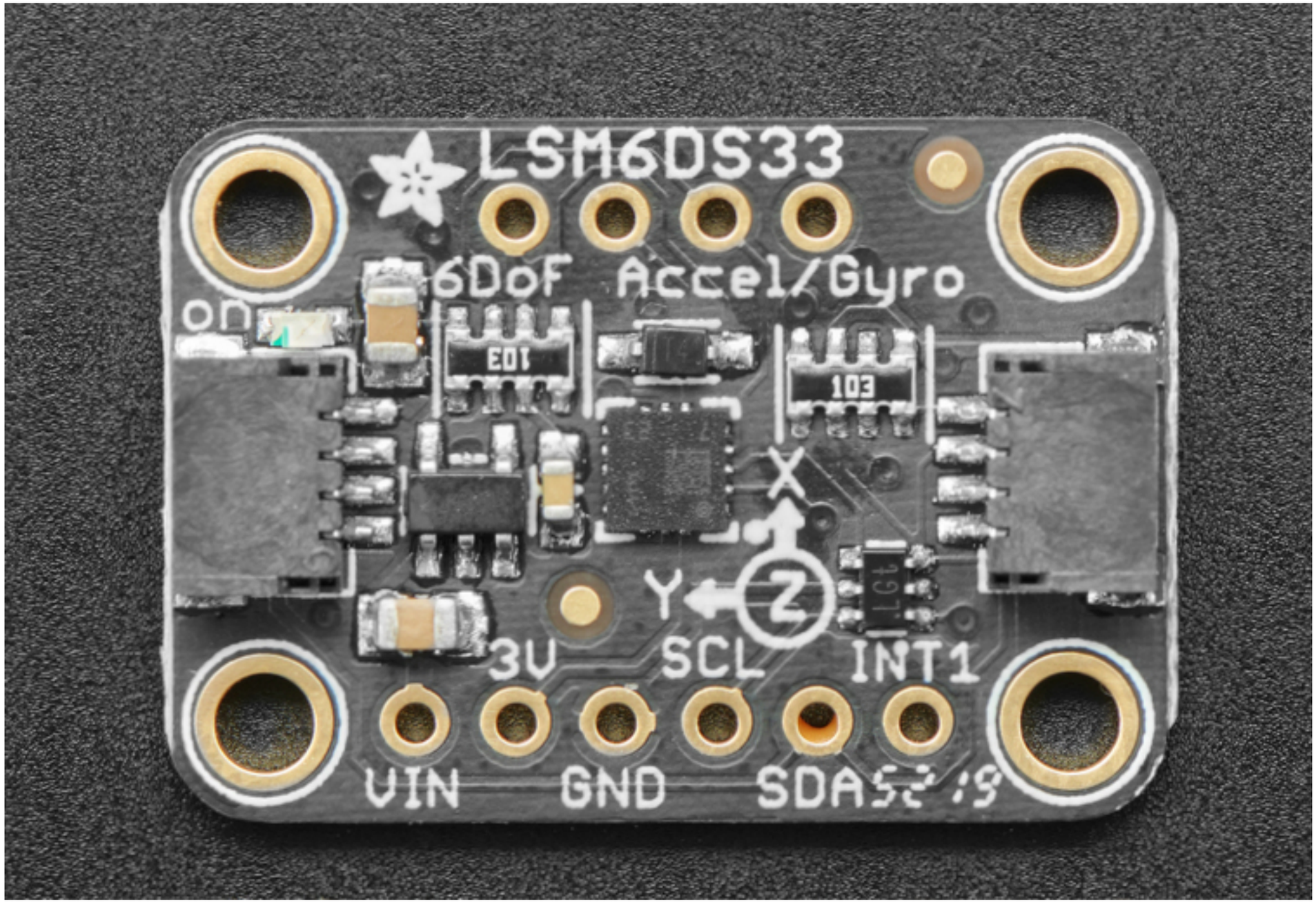

**Pros:**
- **Faster that I²C**

**Cons:**
- **Needs more pins that I²C**

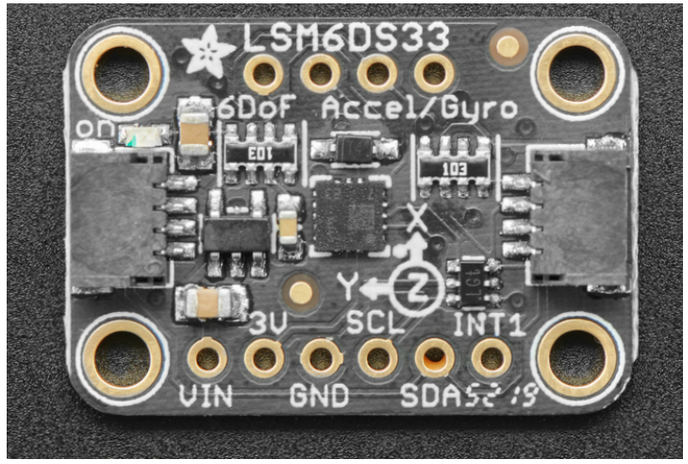
# SPI (Serial Peripheral Interface)



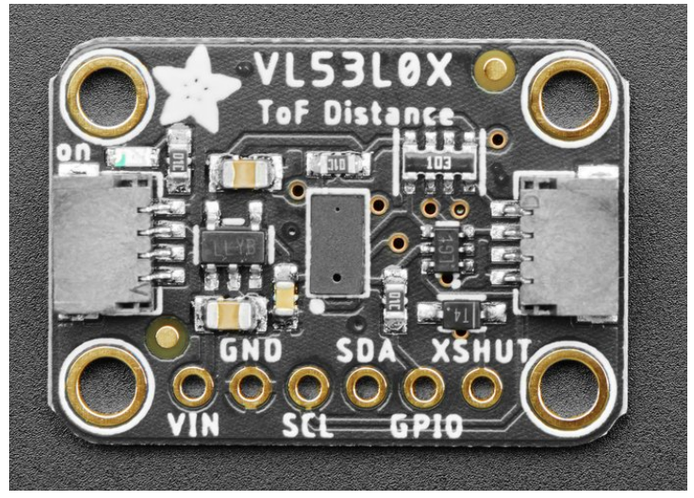

# SPI (Serial Peripheral Interface)



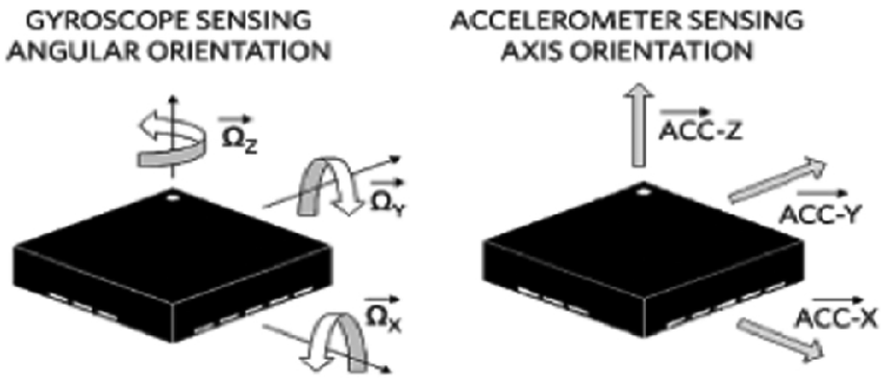

## Digital Peripherals

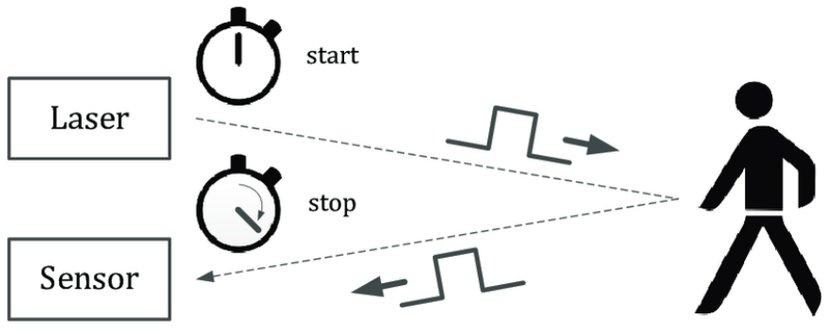



**LSM6DS33**
**IMU**

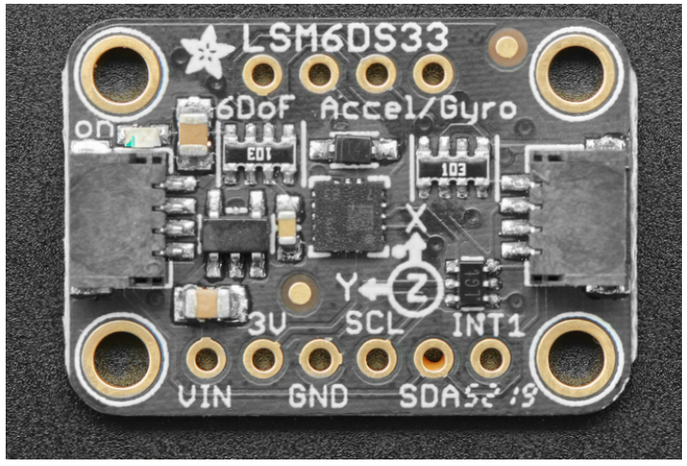


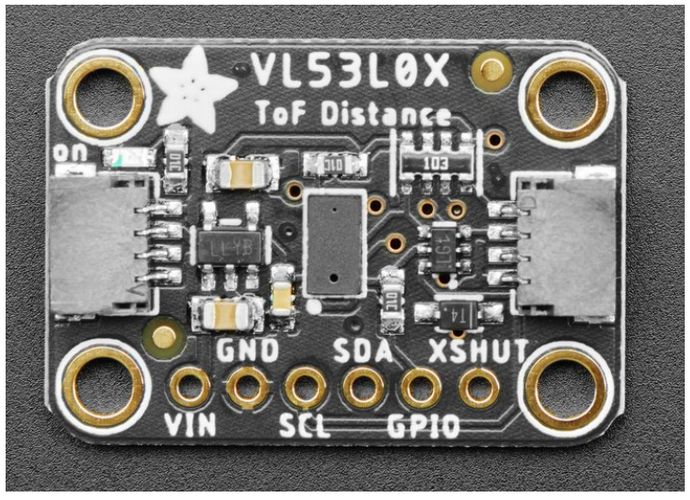

**VL53L0X**
**Distance Sensor**



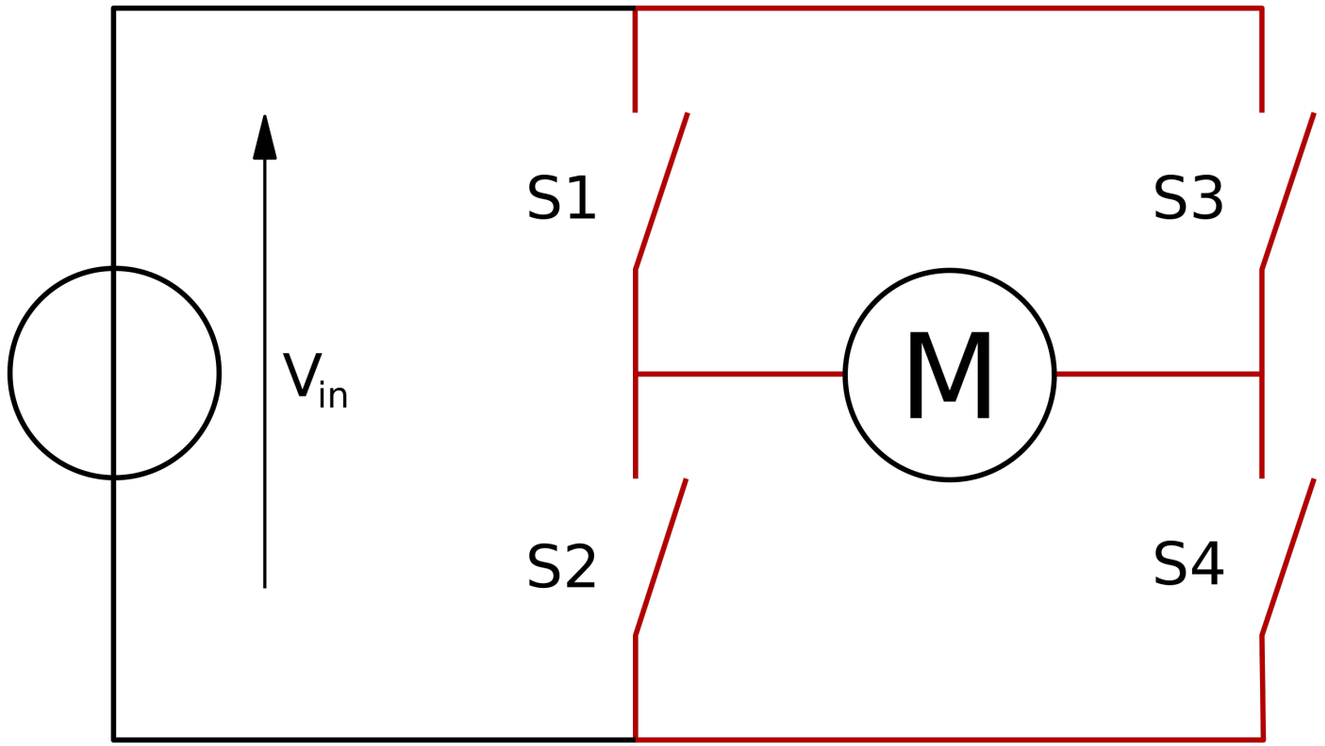

# Digital Sensors



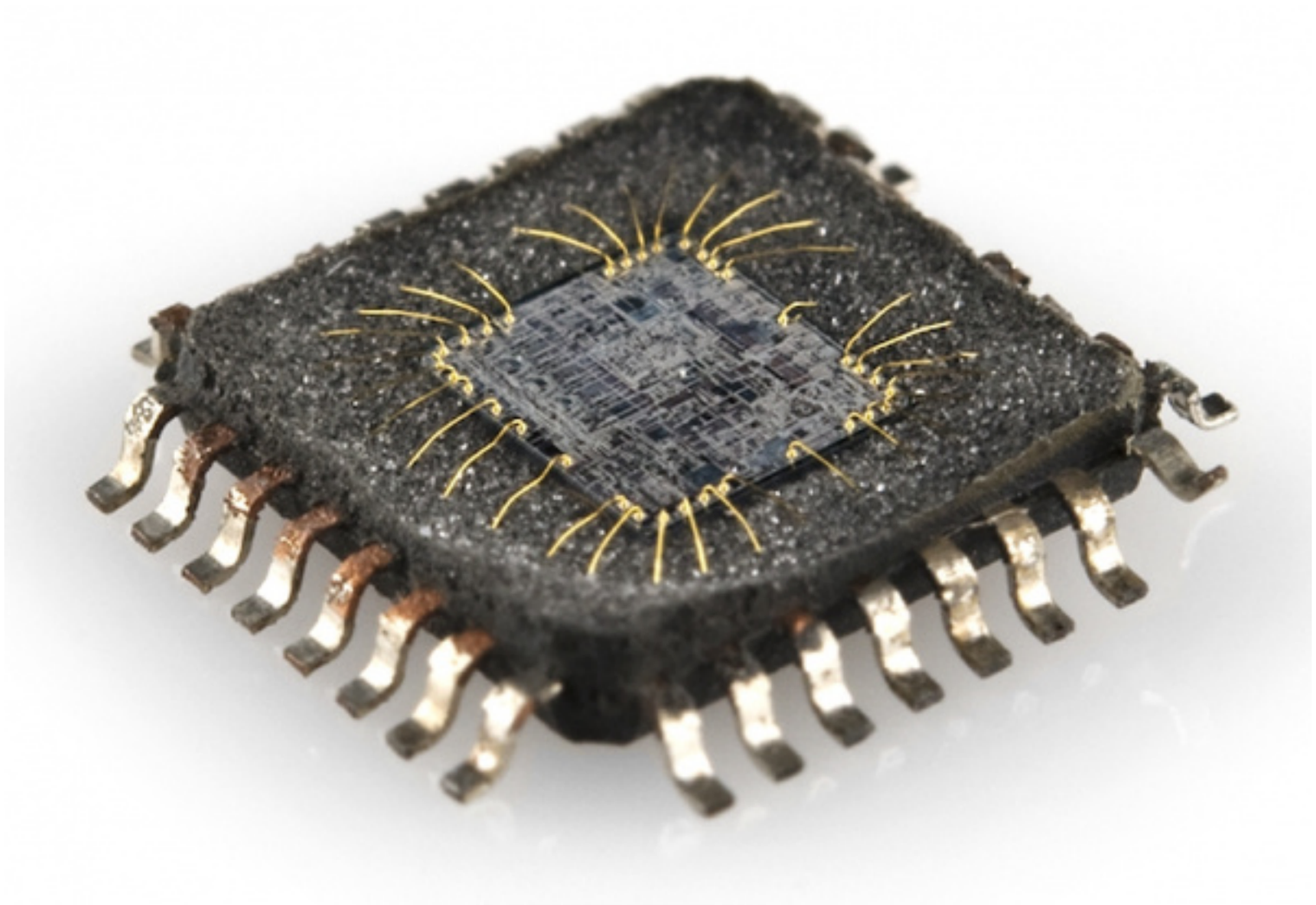

## Exercise PC2.0: Peripherals

**Follow the hook up guide provided by Adafruit for your device. Use the data from your sensor to control one or more outputs, such as LEDs, motors or lightbulbs.**




**H - Bridge**

# Integrated Circuits



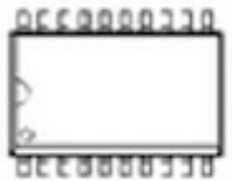
# Integrated Circuits

DDPAK

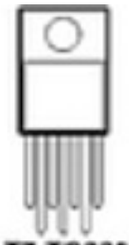
DPAK

DIP

SQP



SW



T7-TO220

FDIP

PDIP

PENTAWATT

TO220

TO220S

TO220ISO

PLCC

QDIP

QFP

TO252

TO263

TO268

SIP

SO

SO8

TO3

TO52

TO99

SOT223

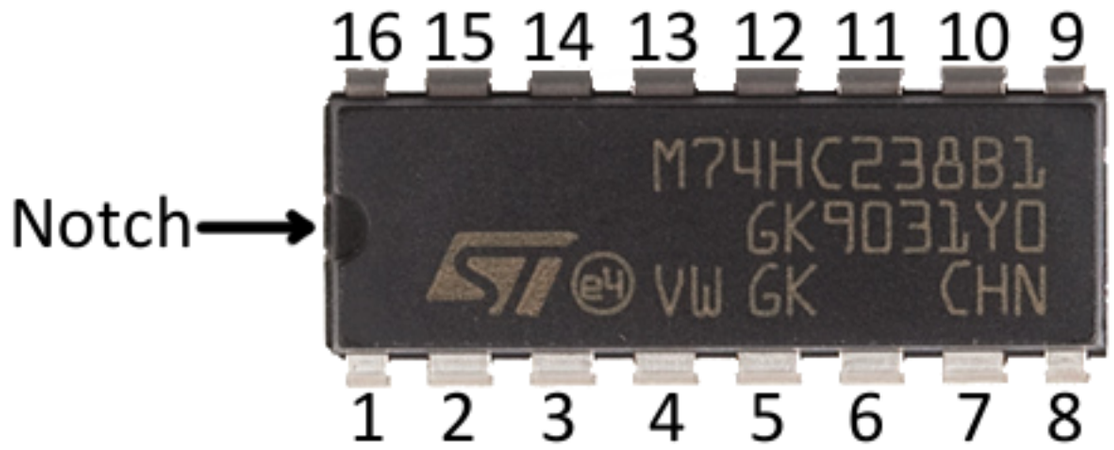
SOT23

SQL

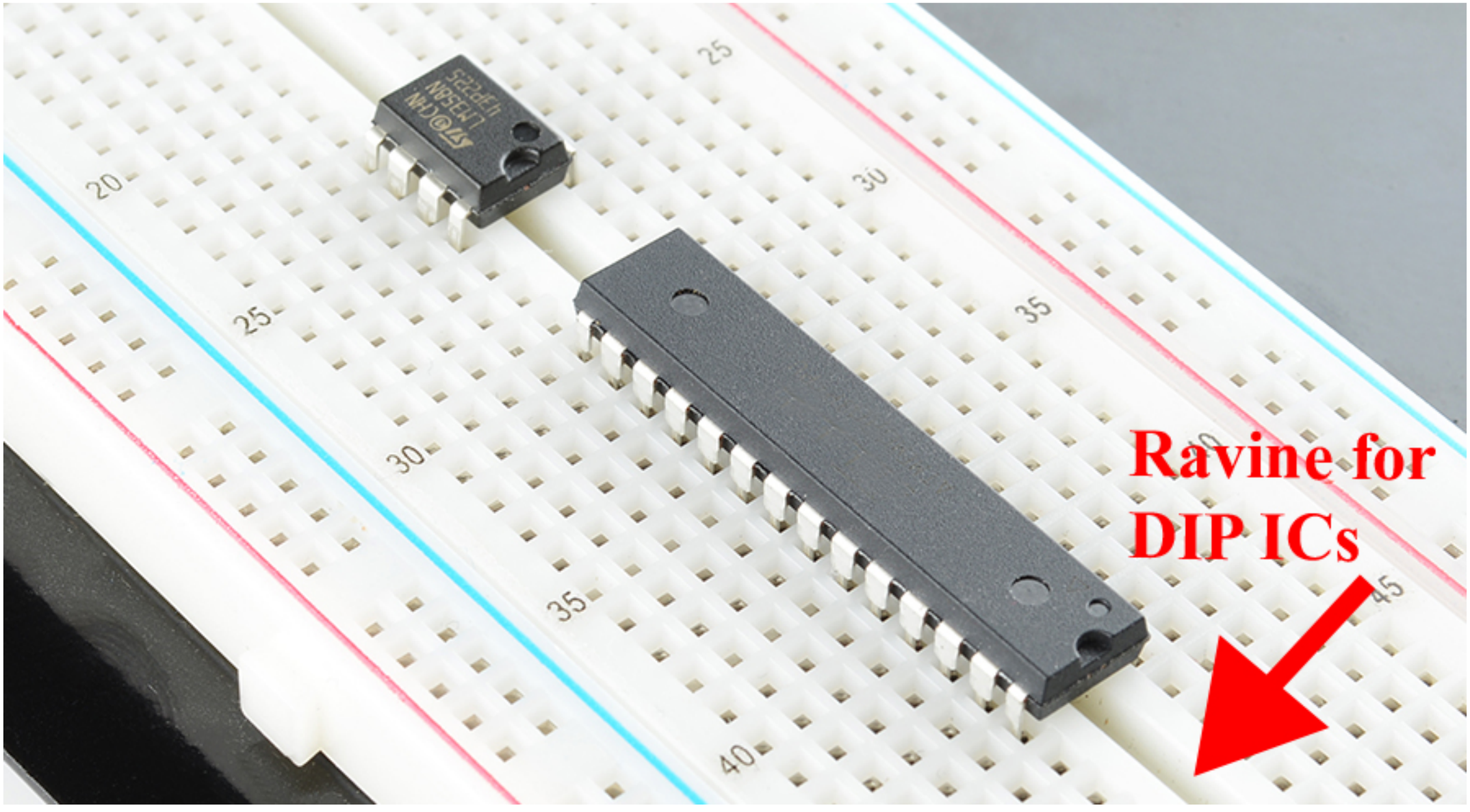
TSOP

ZIP

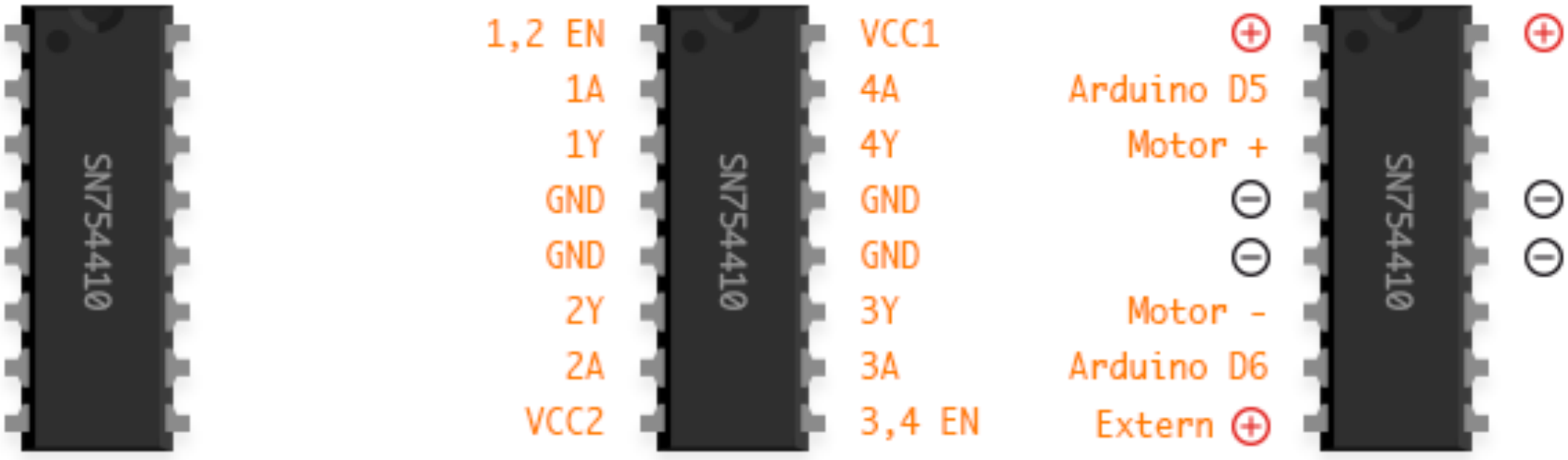
## Packages


16 15 14 13 12 11 10 9
Notch
M74HC238B1
GK9031Y0
VW GK CHN
1 2 3 4 5 6 7 8

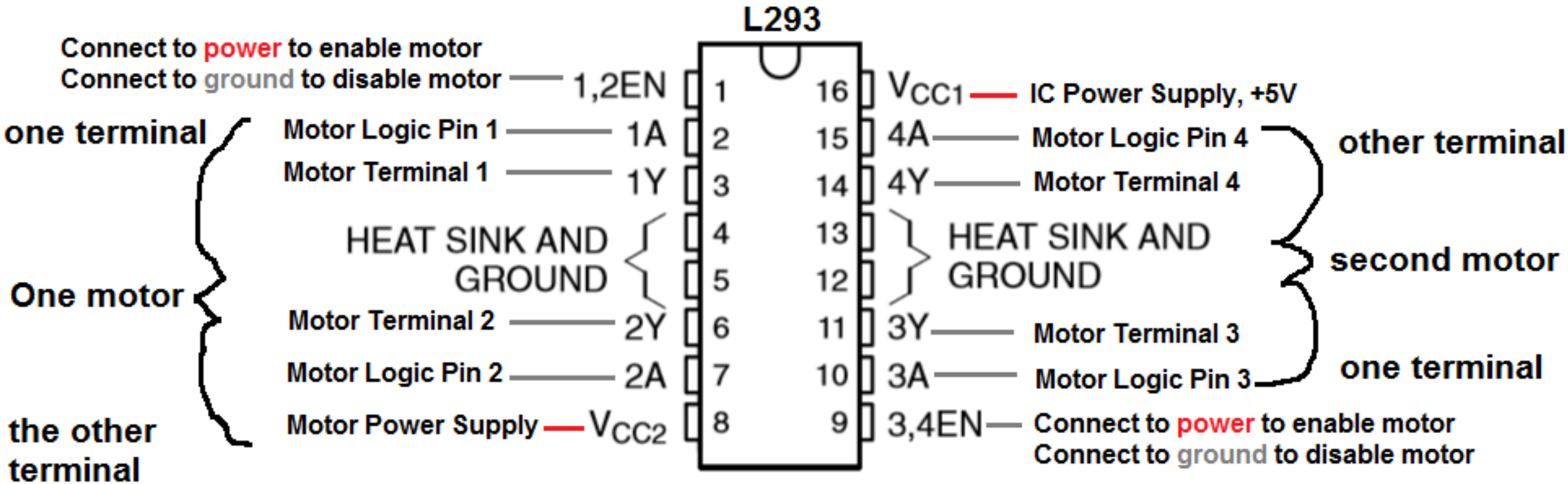
## Packages



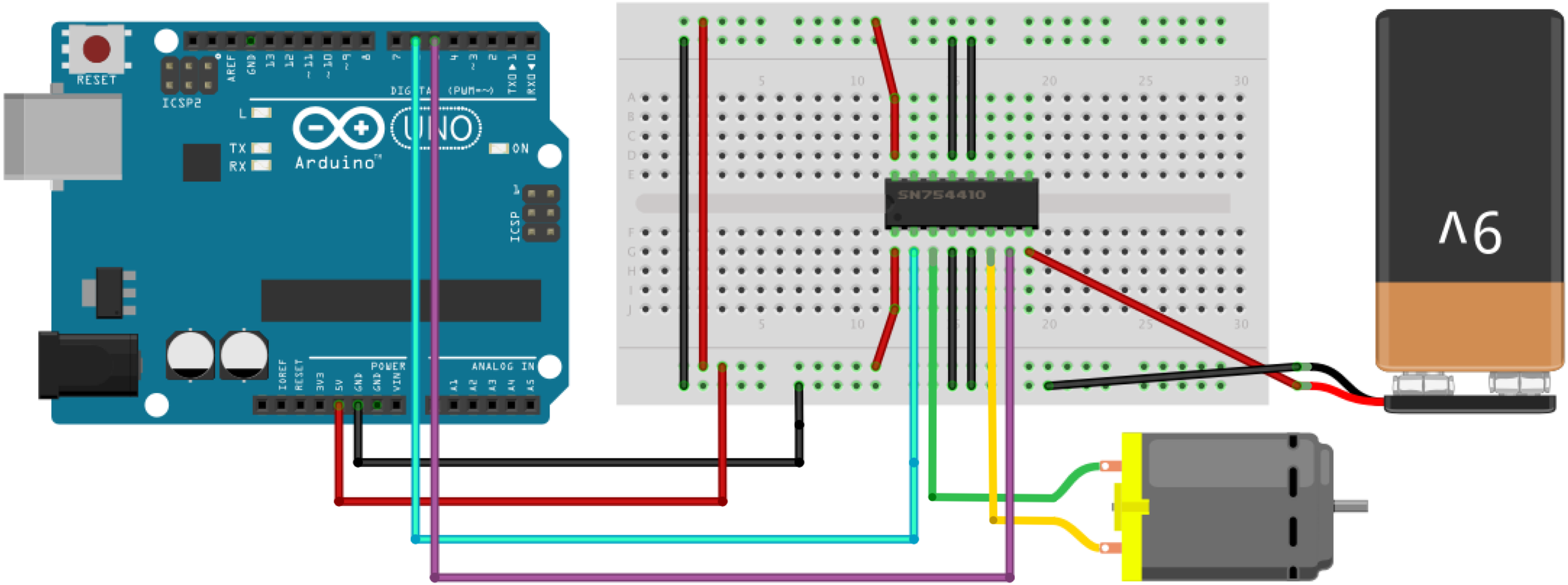

## Packages



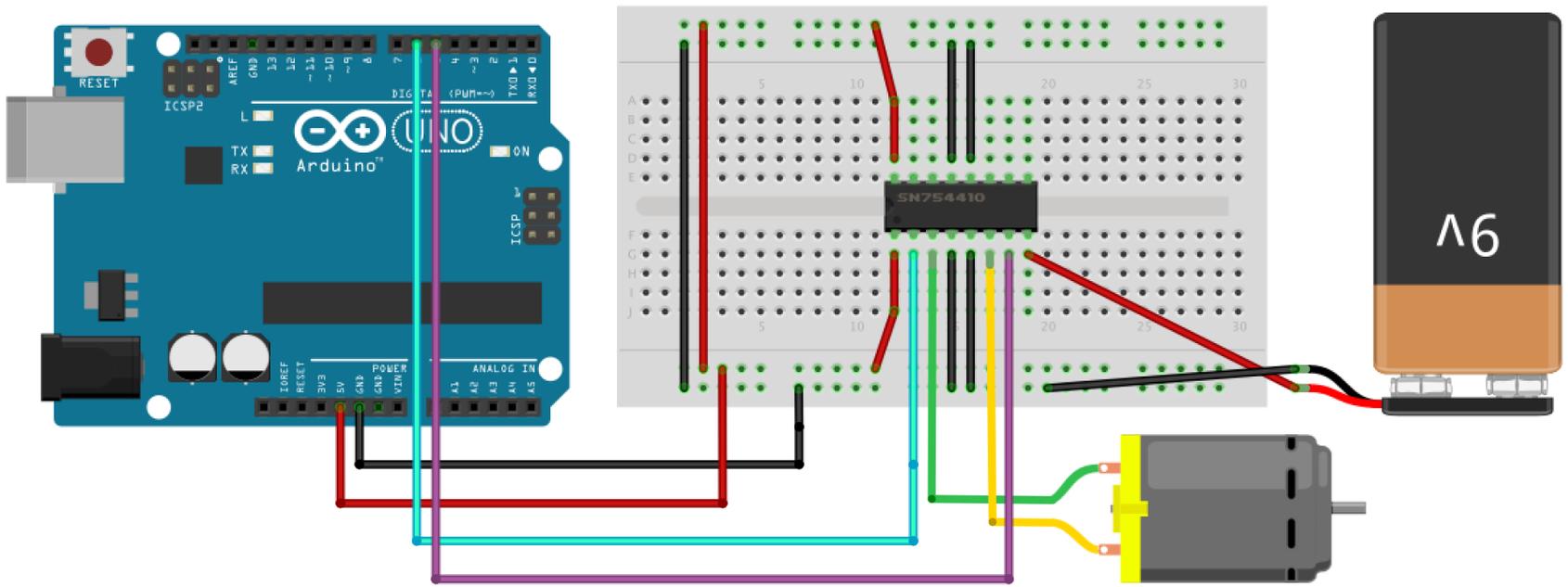

# H Bridge - SN754410




# H Bridge - SN754410




# SN754410 - hook-up




# Exercise PC3.0: H-Bridge

1. Use a push button to change the running direction of the motor using a h-bridge.

2. Add a way of controlling the speed too, such as a potentiometer.

3. Add an additional motor to your h-bridge