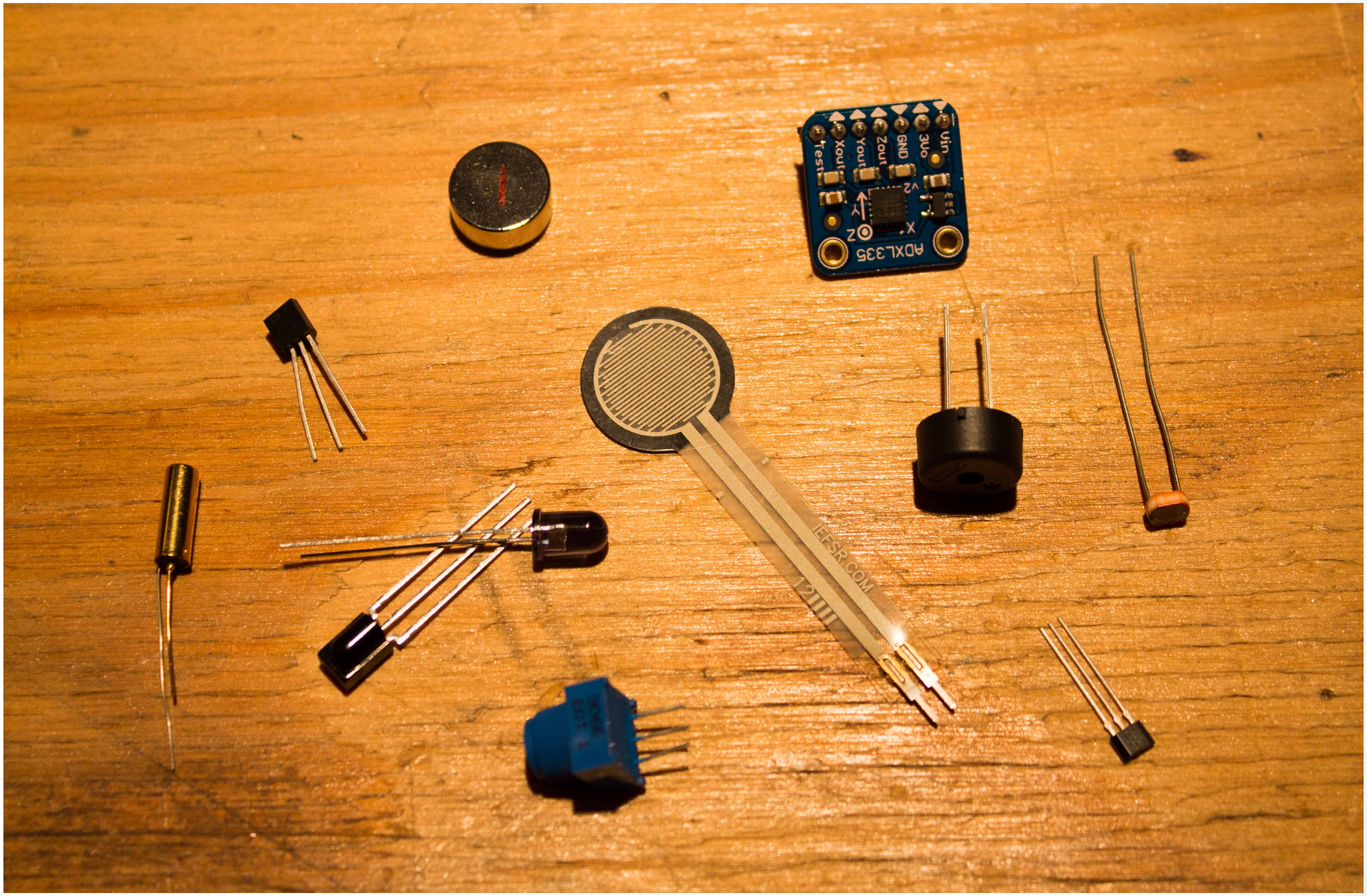
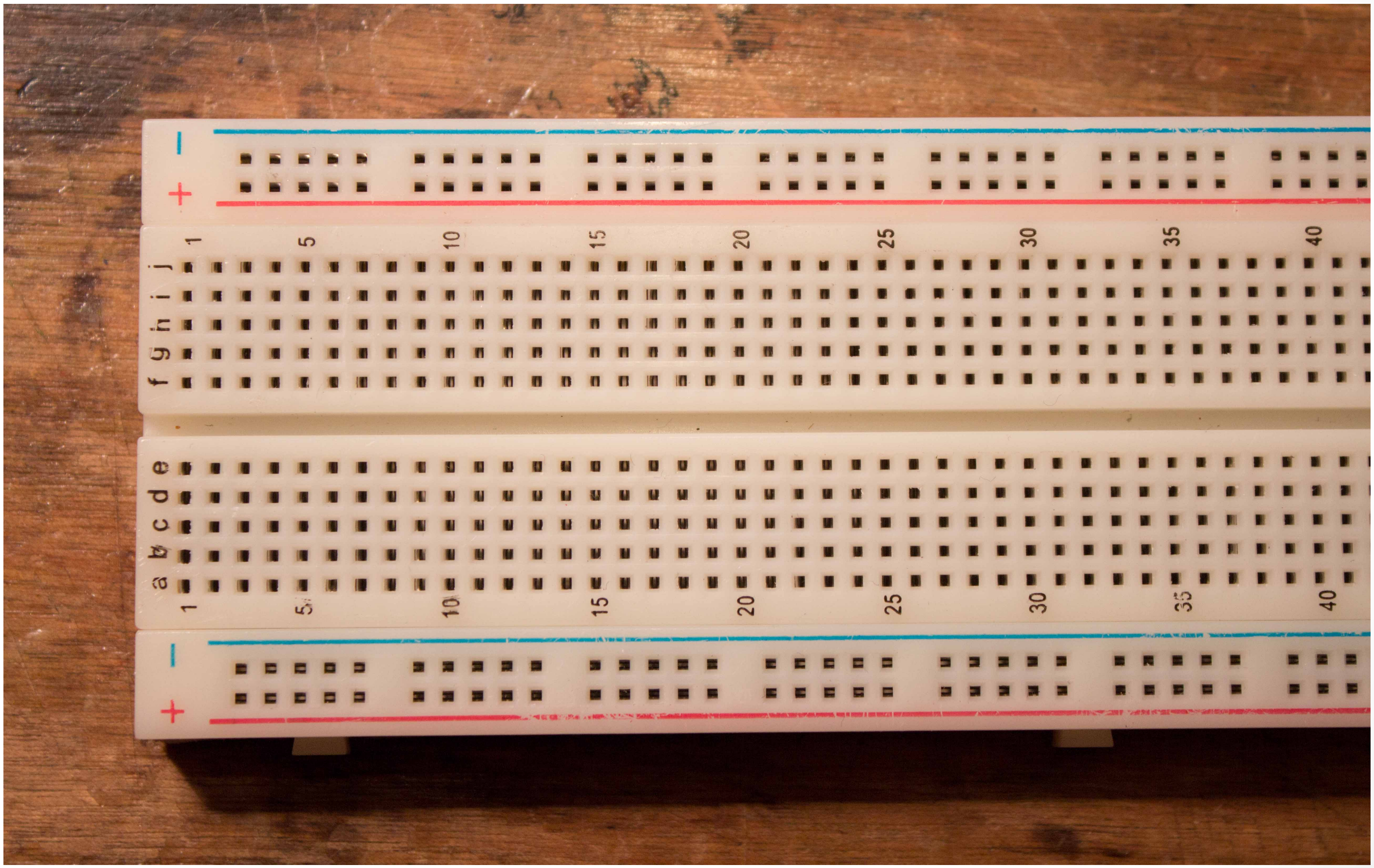
Arduino and Sensors

Miria Grunick
NYC Resistor

Sensors



Breadboard



Resistors

- 1K Ω: brown, black, red, gold
- 10K Ω: brown, black, orange, gold
- 1M Ω: brown, black, green, gold
- Potentiometer: 10k Ω max resistance

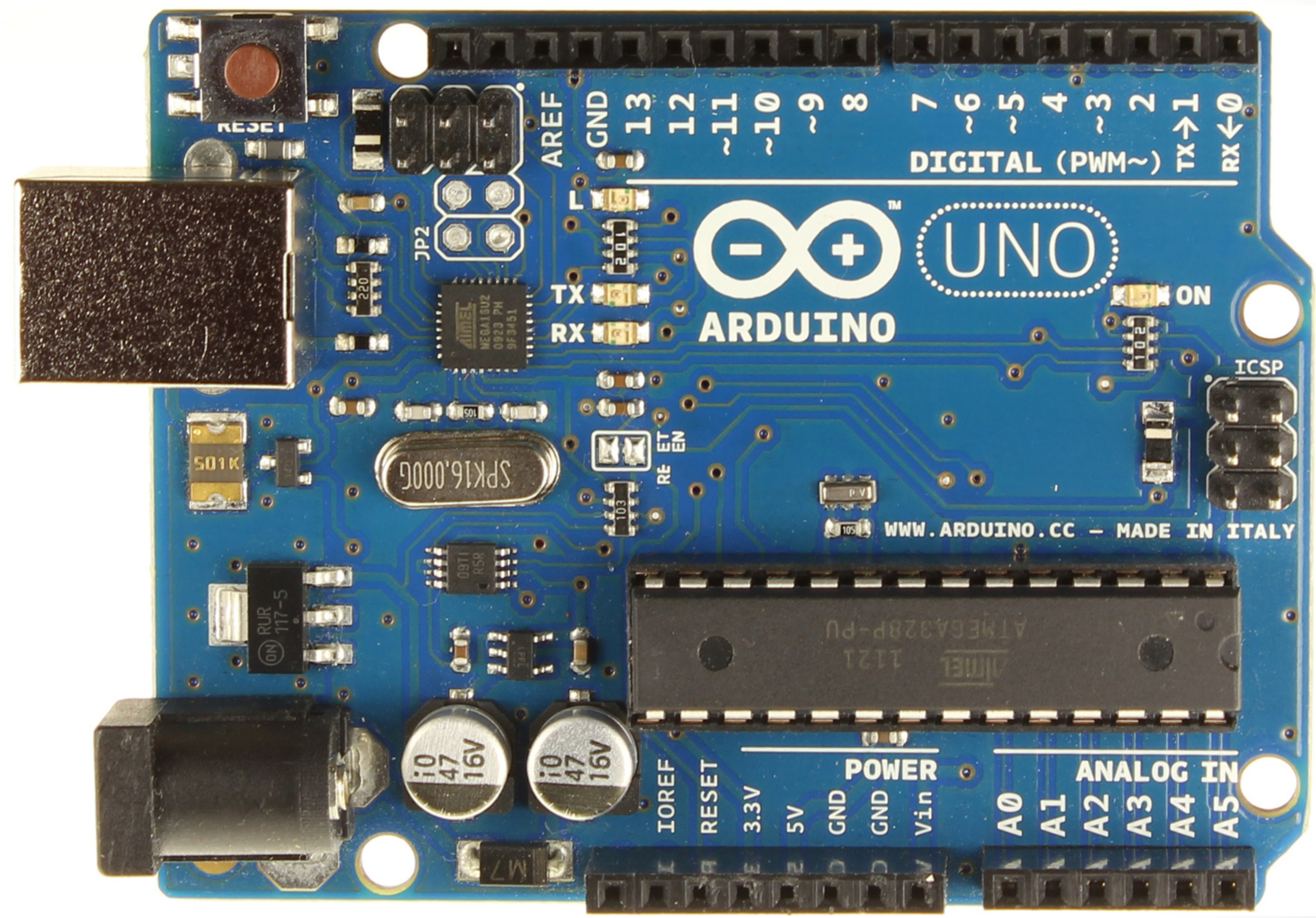
Types of Sensors

- **Digital Signal**
  - Returns either a high signal or a low signal based on whether or not a condition is present. Like a switch, it is on or off.
  - The Arduino software converts this to either a 0 or a 1.
  - Examples: tilt sensor, infrared sensor, hall effect sensor

Types of Sensors

- **Analog Signal**
  - Returns a voltage between 0 and the max voltage based on how much of an effect the sensor detects.
  - The Arduino software converts this to a value between 0 (no voltage) and 1023 (full voltage)
  - Examples: force sensor, accelerometer, temperature sensor

Arduino



RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
ICSP
JP2
WWW.ARDUINO.CC – MADE IN ITALY
ATMEGA328P-PU
IOREF
RESET
3.3V
5V
GND
GND
Vin
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5

Agenda

- Digital signal sensors
  - Ball Tilt Sensor
  - Hall Effect Sensor
  - Infrared Sensor
- Analog signal sensors
  - Force Sensor
  - Knock Sensor
  - Photo Cell Sensor
- Potentiometers
  - Photo Cell Sensor part two

Agenda

- More analog signal sensors
  - Temperature Sensor
- Analog reference voltages
  - Temperature Sensor part two
- Calibrating sensors
  - Accelerometer calibration
  - Accelerometer usage
- Tips and tricks

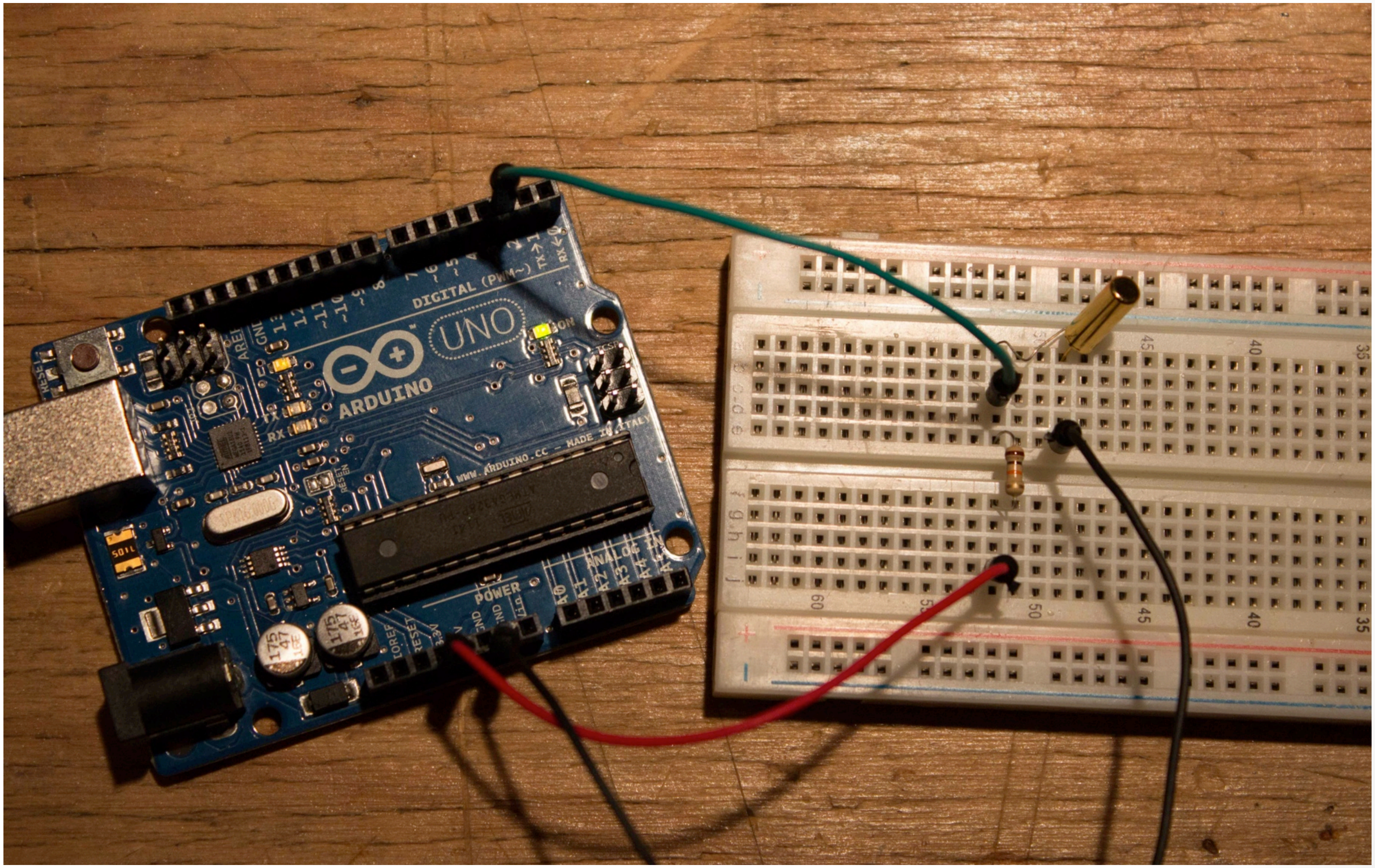
Knowing your sensor!

- Sensors come with a datasheet
- These contain important information
  - Operating voltages
  - What the output signal means
  - Wiring information
  - Polarity information
  - Calibration information
  - Accuracy information
  - Maximum current
  - Maximum sensor conditions
  - …and more!

Ball Tilt Sensor

- Senses tilt
- Digital signal
- Requires 10K Ω resistor between power and sensor
- Order of pins is not important
- Code: ball_tilt_demo

Wiring the ball tilt sensor



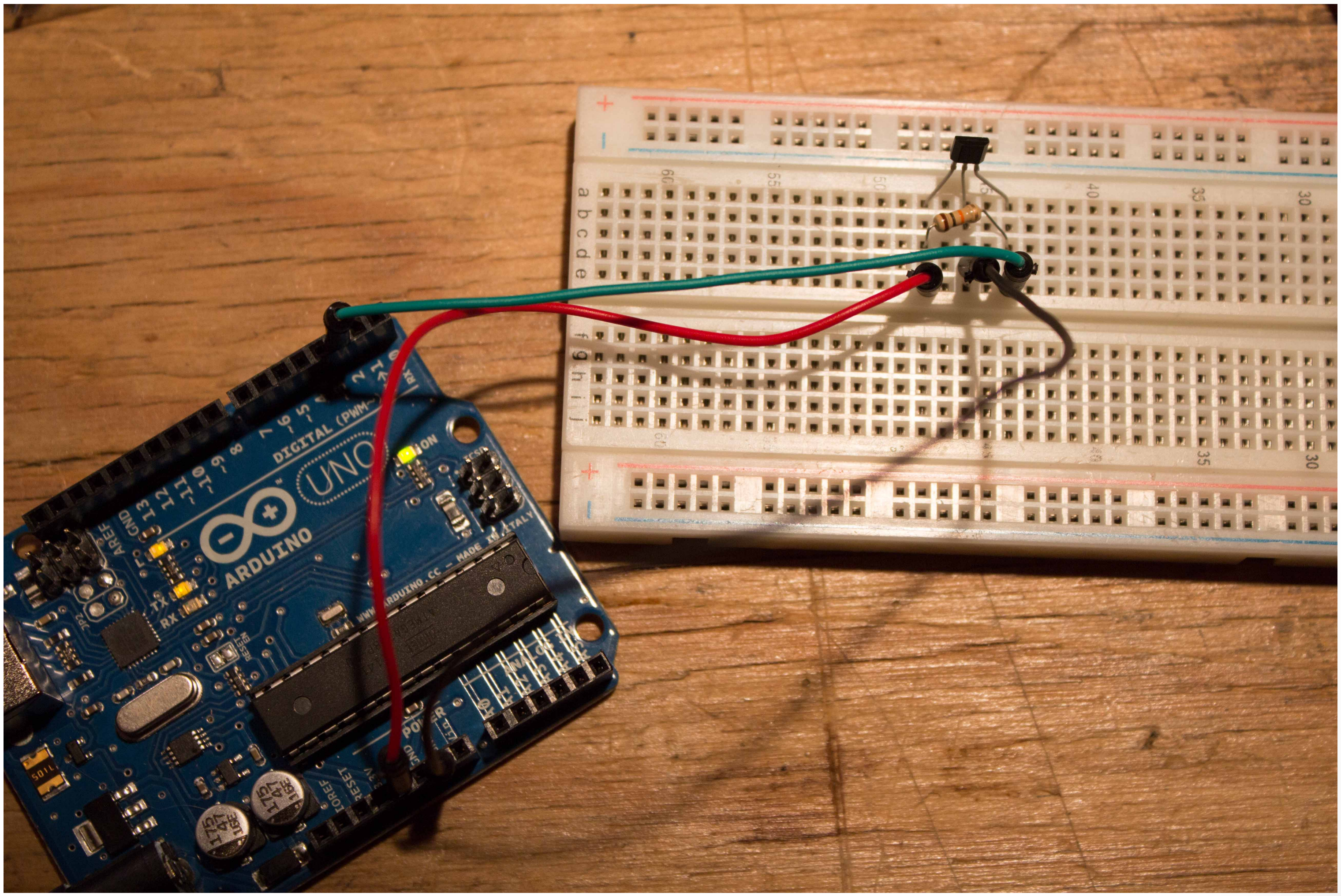
Hall Effect Sensor

- Detects magnets
- Digital signal
- Only senses the south pole of the magnet
- Code: hall_effect_demo

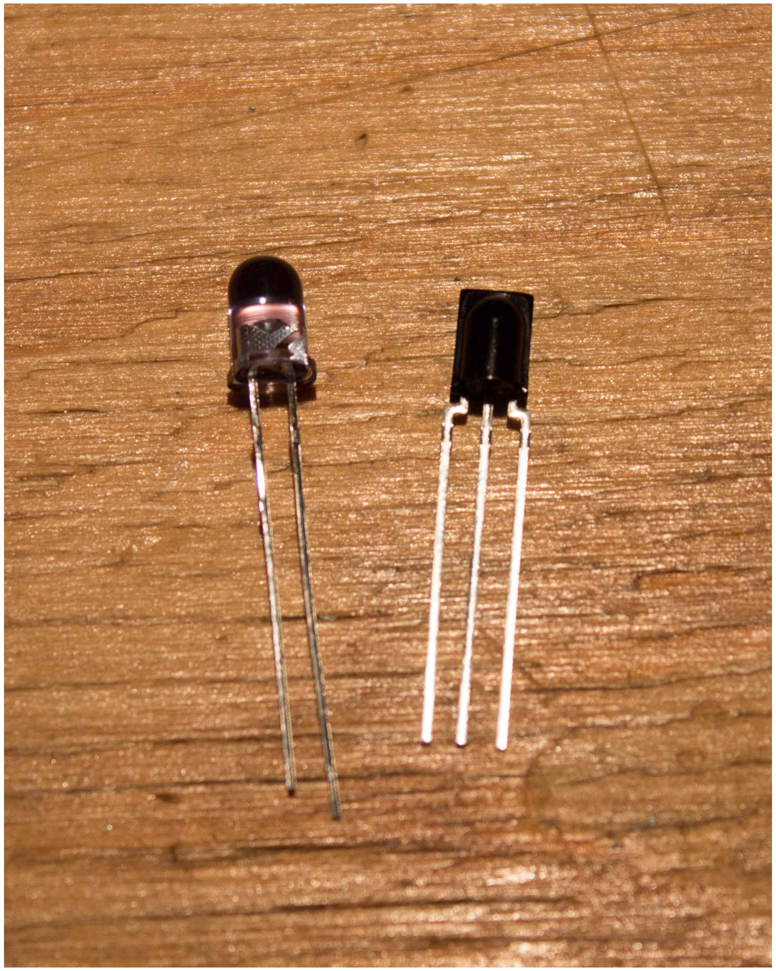
Wiring the hall effect sensor

- Order is important!
- Curved/angled part facing you
- Pin 1: Power in
- Pin 2: Ground
- Pin 3: Signal out
- 10K Ω resistor between pin 1 and 3
- The side facing you is the sensing side

Wiring the hall effect sensor



Infrared (IR) Sensor

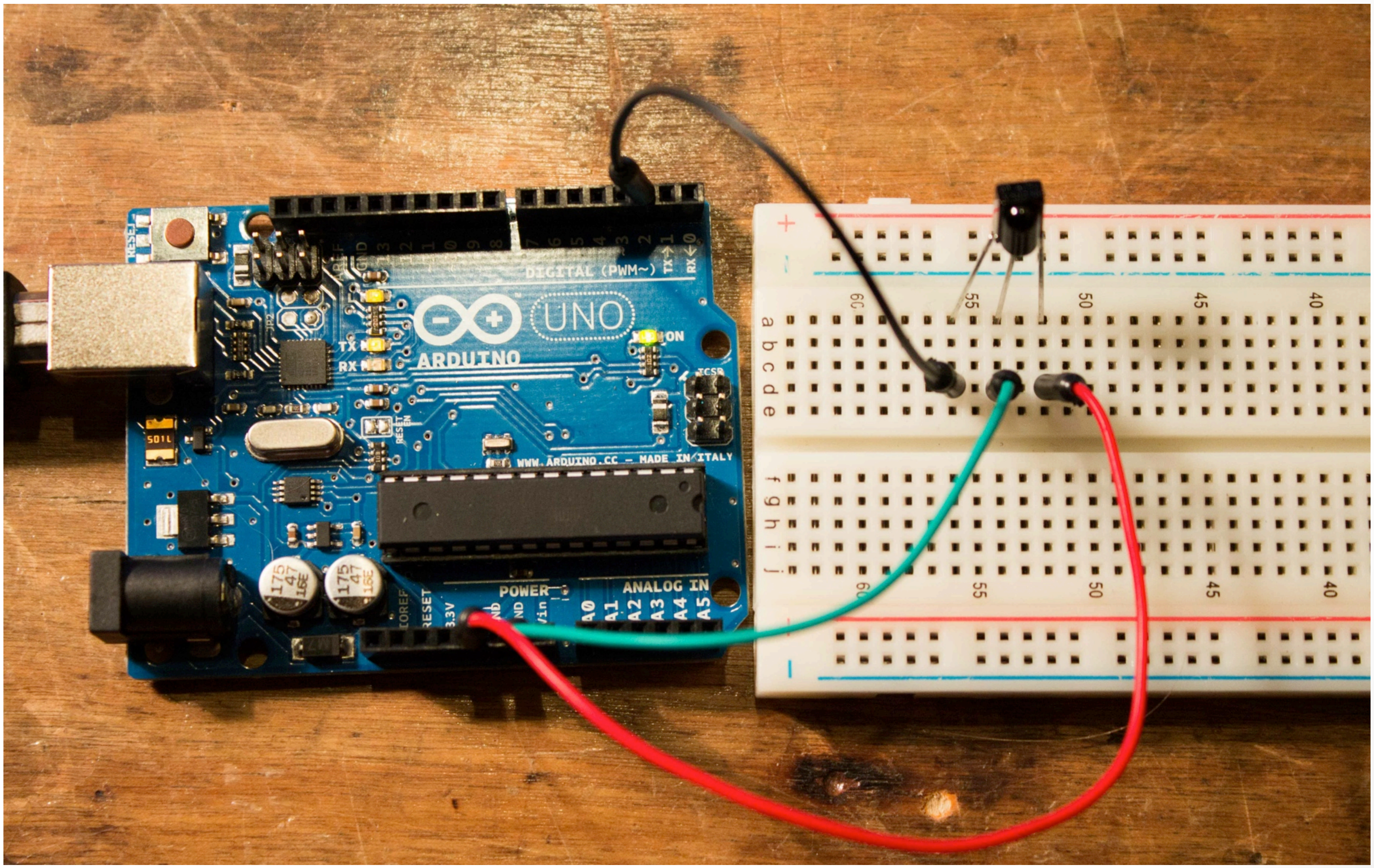


- Senses infrared light
- Digital signal
- Left: IR LED for testing
- Right: IR Sensor
- Code: infrared_demo

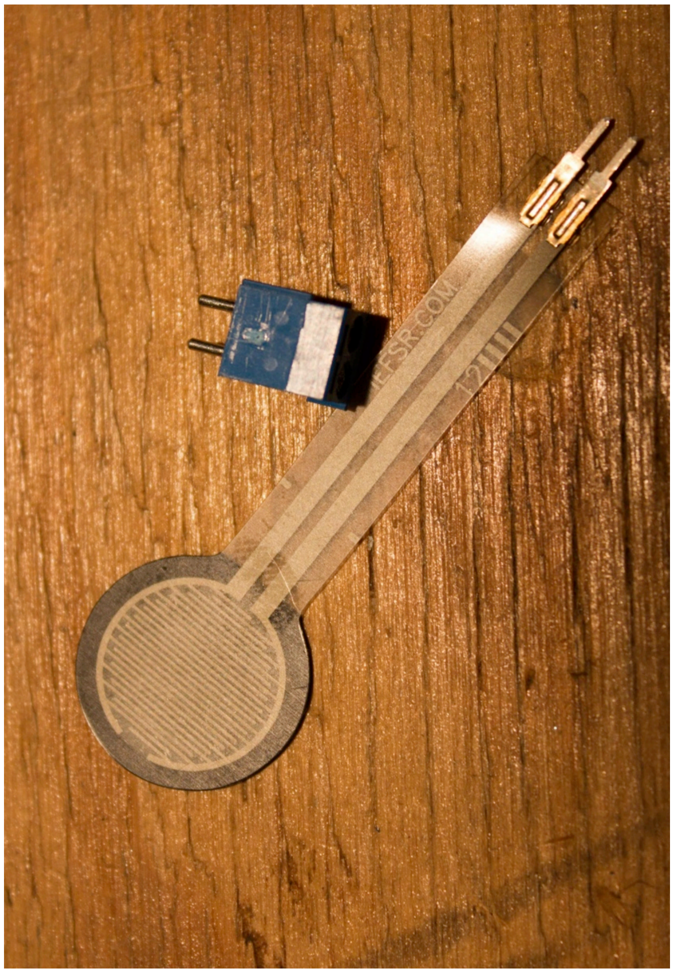
Wiring the IR sensor

- Order is important
- Sensing side is the “bubble” side
- Sensing side facing you
- Pin 1: signal output
- Pin 2: ground
- Pin 3: voltage
- No resistor required!

Wiring the IR sensor

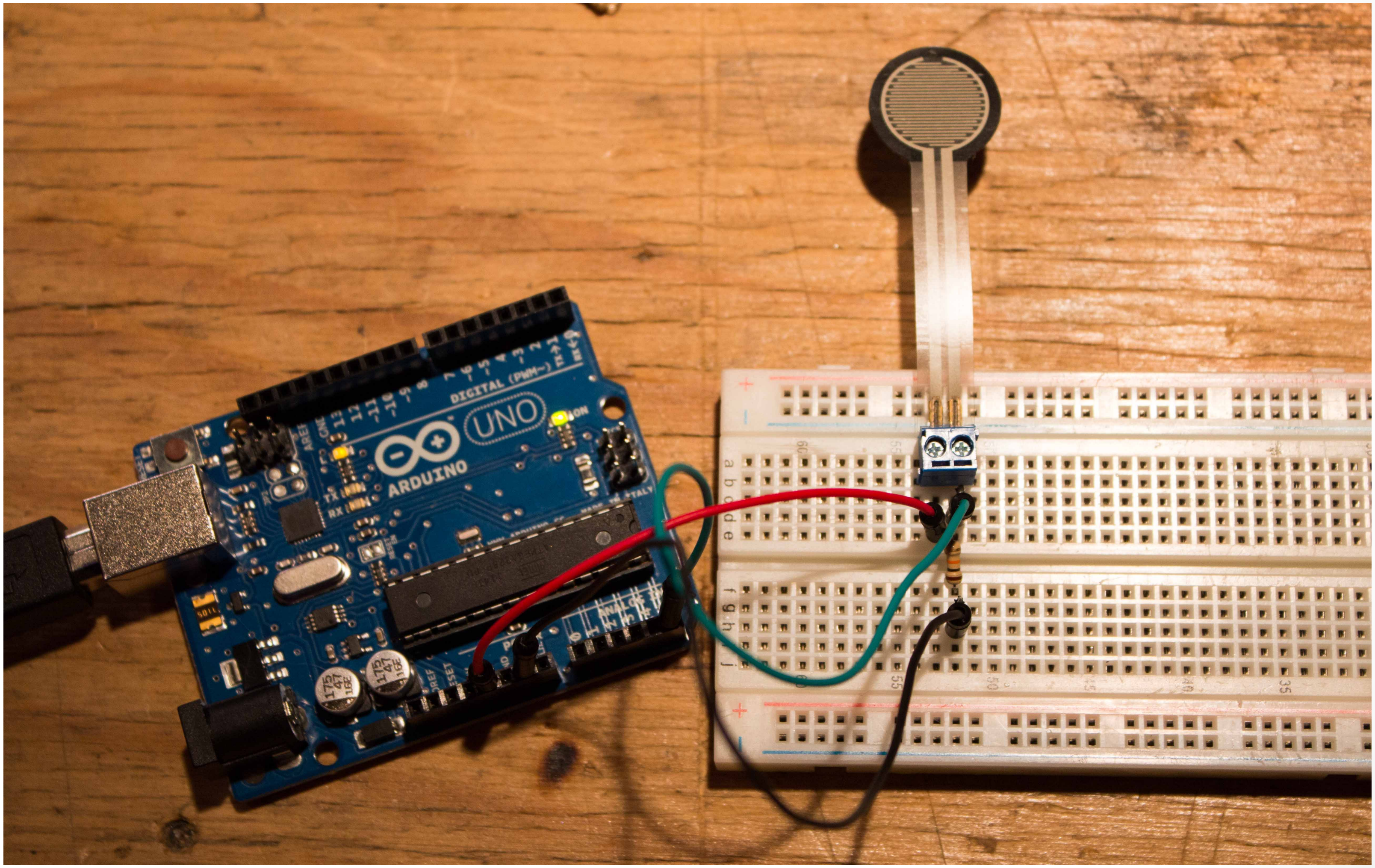


Force Sensor

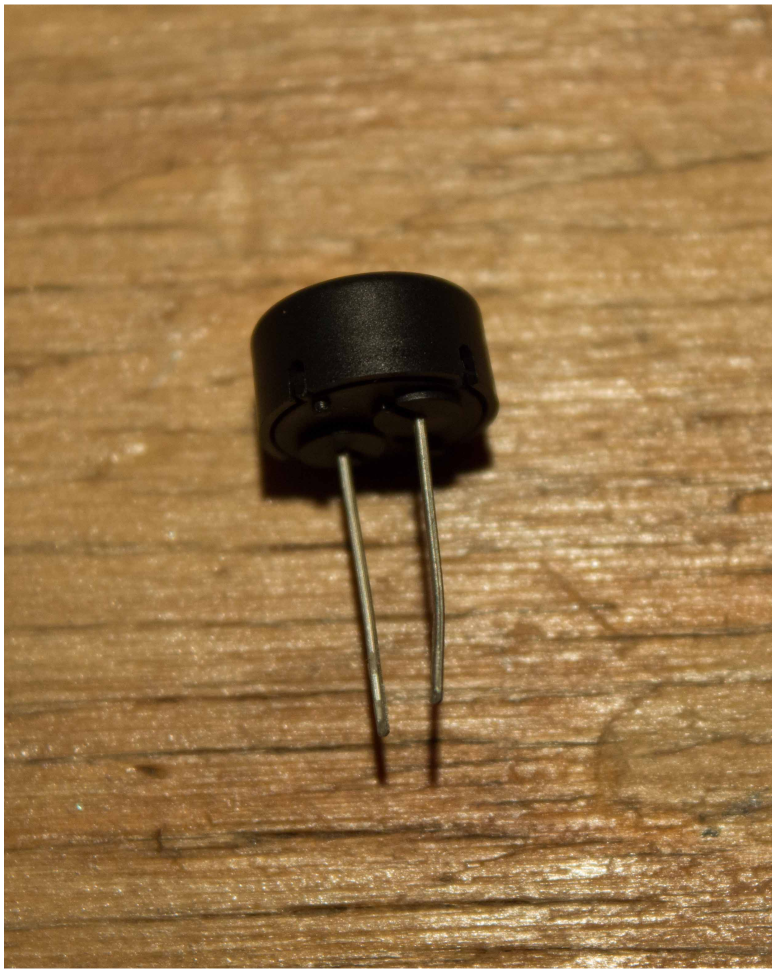


- Senses force
- Analog signal
- Order of pins is not important
- Requires 10K Ω resistor between ground and output
- Code: force_demo

Wiring the force sensor



Knock Sensor/Piezo Buzzer



- Detects vibration or sound
- Analog signal
- Order of pins is not important
- Requires 1M Ω resistor between ground and output
- Code: knock_sensor_demo

Wiring a knock sensor

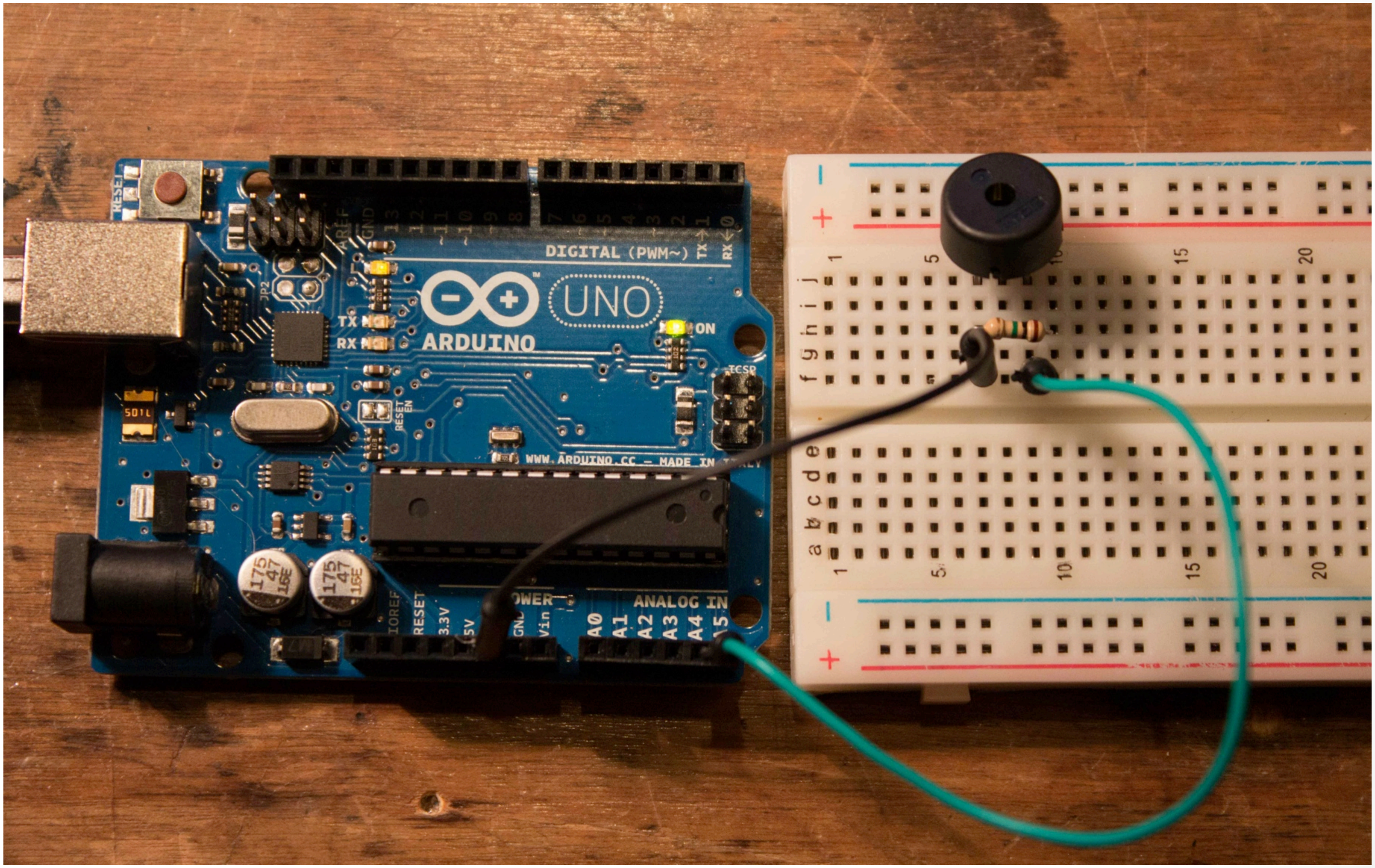
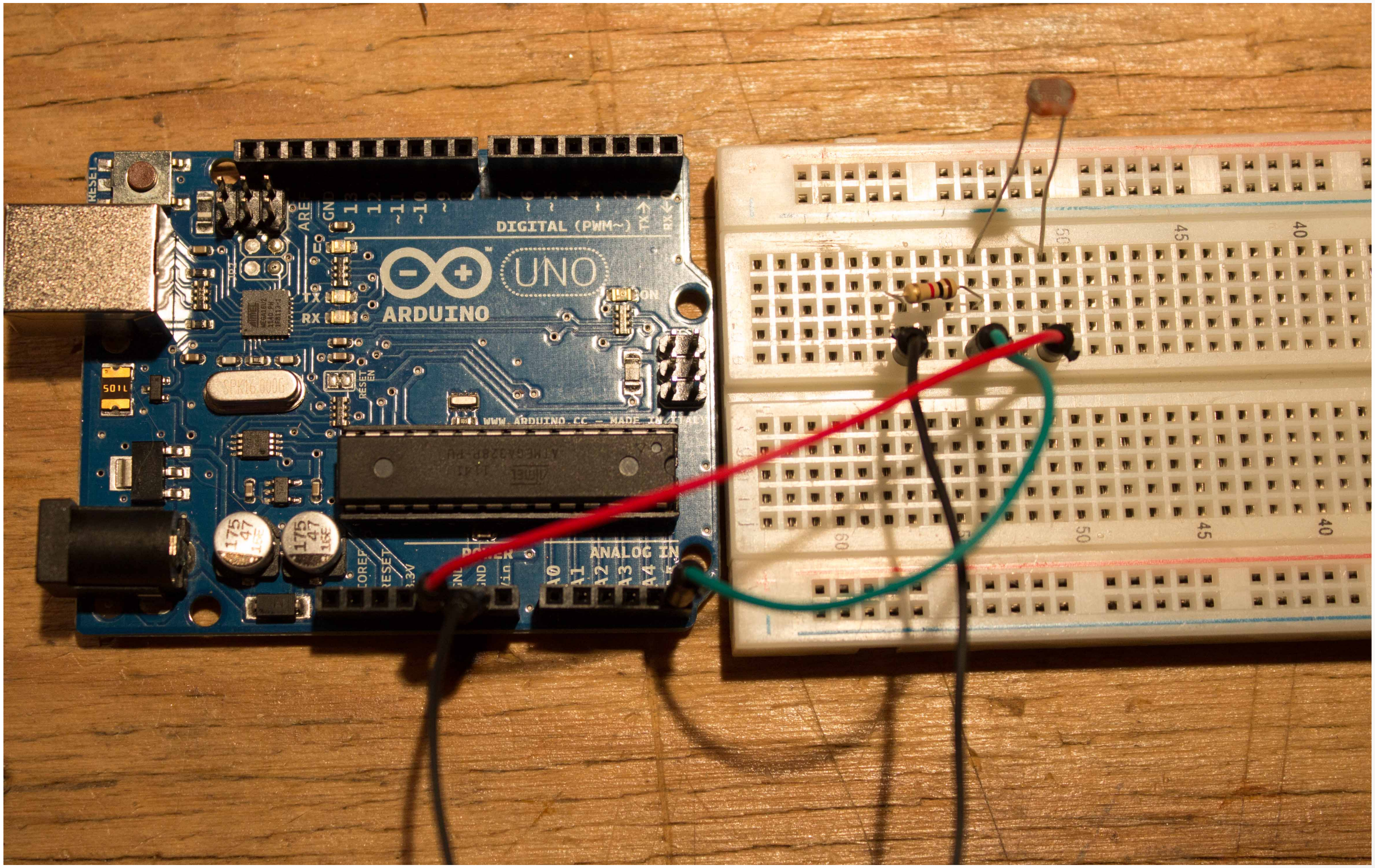


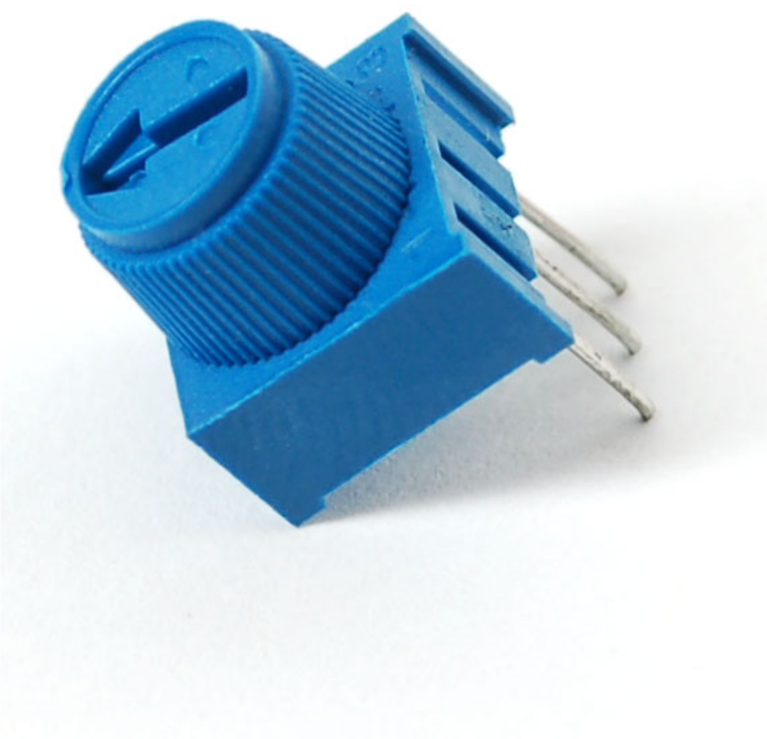
Photo Cell Sensor

- Senses light
- Analog signal
- Pin order not important
- 1K Ω (bright) or 10K Ω (dark) resistor between output and ground
- Code: photo_cell_demo

Wiring the photo cell



Potentiometers

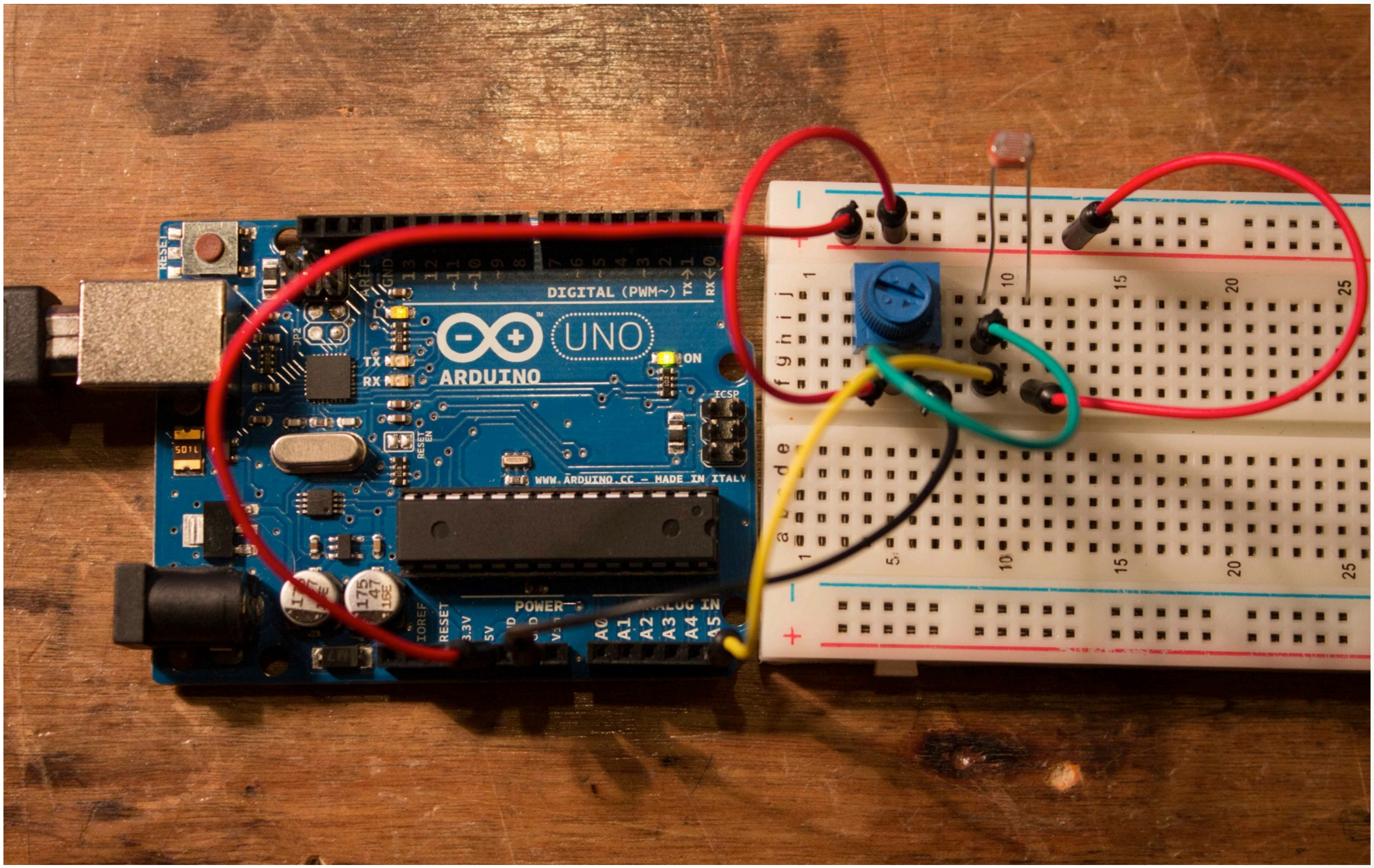


- Variable resistor
- One outer pin is ground and the other is power (order not important)
- Turning the knob changes resistance
- Middle pin is transformed voltage

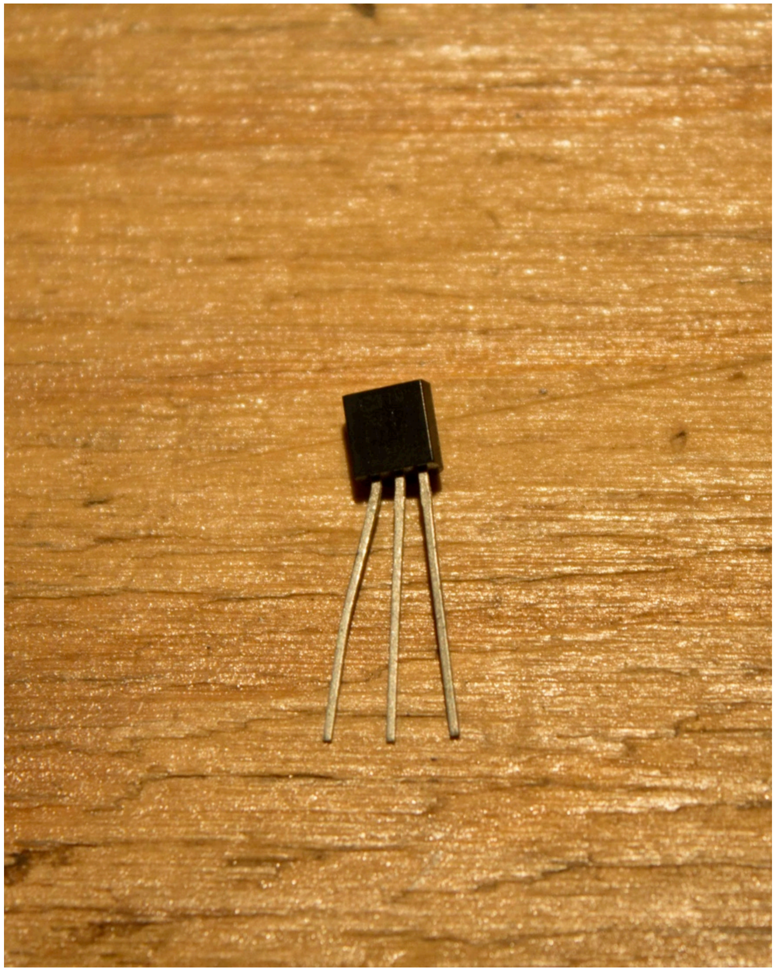
Potentiometers and Sensors

- We can use potentiometers to tune the sensitivity of some types of sensors
- The potentiometer we have is a 10K Ω potentiometer, meaning the maximum resistance is 10K Ω
- Let's see what happens when we connect this to our photo cell sensor…

Wiring the photo cell and pot



Temperature Sensor



- Senses the temperature
- Analog signal
- Flat side is the front
  - Pin 1: Power in
  - Pin 2: Sensor output
  - Pin 3: Ground
- No resistor
- code: temperature_demo

Sensor Datasheet

- Open TMP35_36_37.pdf
- Let's figure out how to calculate the temperature by looking at the datasheet
- Our sensor is the TMP36

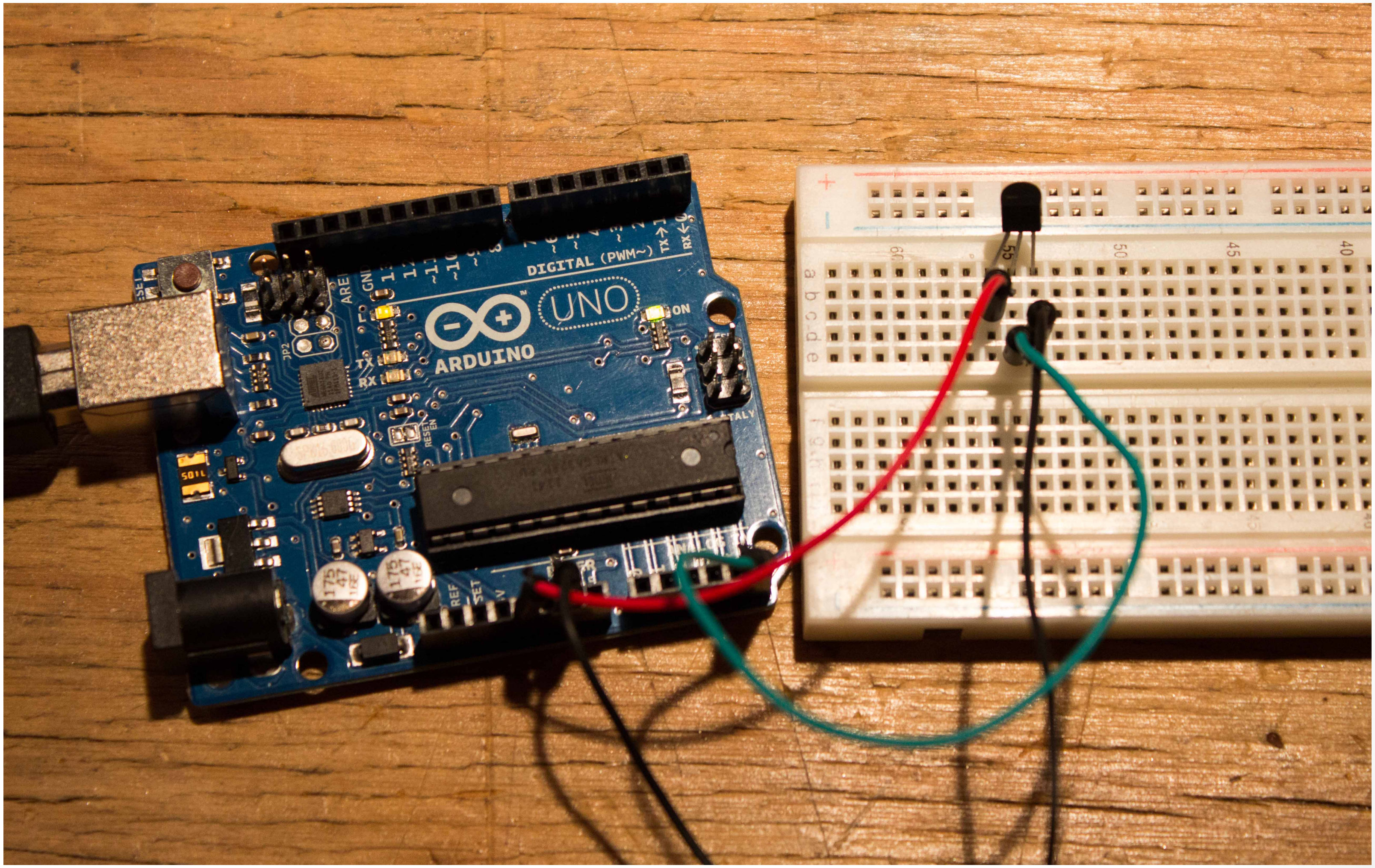
Page 1:

> *The TMP36 is specified from −40°C to +125°C, provides a 750 mV output at 25°C, and operates to 125°C from a single 2.7 V supply. The TMP36 is functionally compatible with the LM50. Both the TMP35 and TMP36 have an output scale factor of 10 mV/°C.*

Calculating Temperature

- At 750 mV of output, the temperature is 25°C.
- Every 10 mV (1/100 V) = 1°C
- Thus, at 0 V, the temperature is 75° C cooler than at 750 mV, so the 0 V temperature is -50° C.
- To get the voltage, take the analogRead value, divide it by 1024 (to get the percentage of the voltage) and multiply that by the max voltage (here 5 volts)
- Multiply the voltage by 100 to convert it to our scale factor (1/100 V = 1°C)
- Subtract 50, since at 0 volts, the temperature is -50° C
- temp = (((reading / 1024.0) * 5.0)) * 100.0 – 50.0

Wiring the temperature sensor



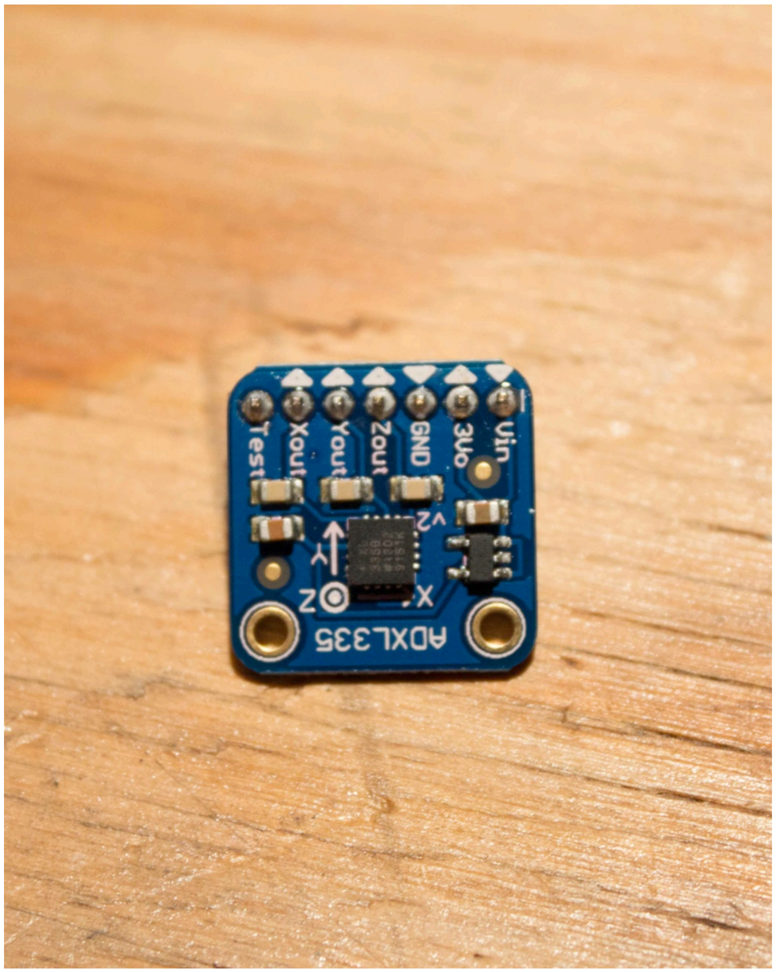
AREF: Analog Reference

- A reference voltage is the maximum voltage returned by an analog read
- By default, the reference voltage is the same voltage as the board (for the Uno, it's 5 volts)
- In a 5 volt board, each digit in the analog read value represents 4.88 millivolts (5 volts / 1024 possible values)
- Setting a lower analog reference voltage means that you can have greater precision.
- Setting your analog reference voltage to 3.3 volts means that each digit in the analog read value represents 3.22 millivolts (3.3 volts / 1024 possible values)

Using AREF

- To set the analog reference voltage to 3.3 volts, run a wire from the regulated 3.3 volt power supply to the AREF pin
- In your code, you need to tell the Arduino that there is a different analog reference voltage
  - In setup(), add the following line:
  - analogReference(EXTERNAL);
- Let’s do this to the temperature sensor!
- Code: temperature_demo_aref

Accelerometer



- Measures gravitational force
- Analog signal
- Pins are conveniently labeled
- Must be calibrated!

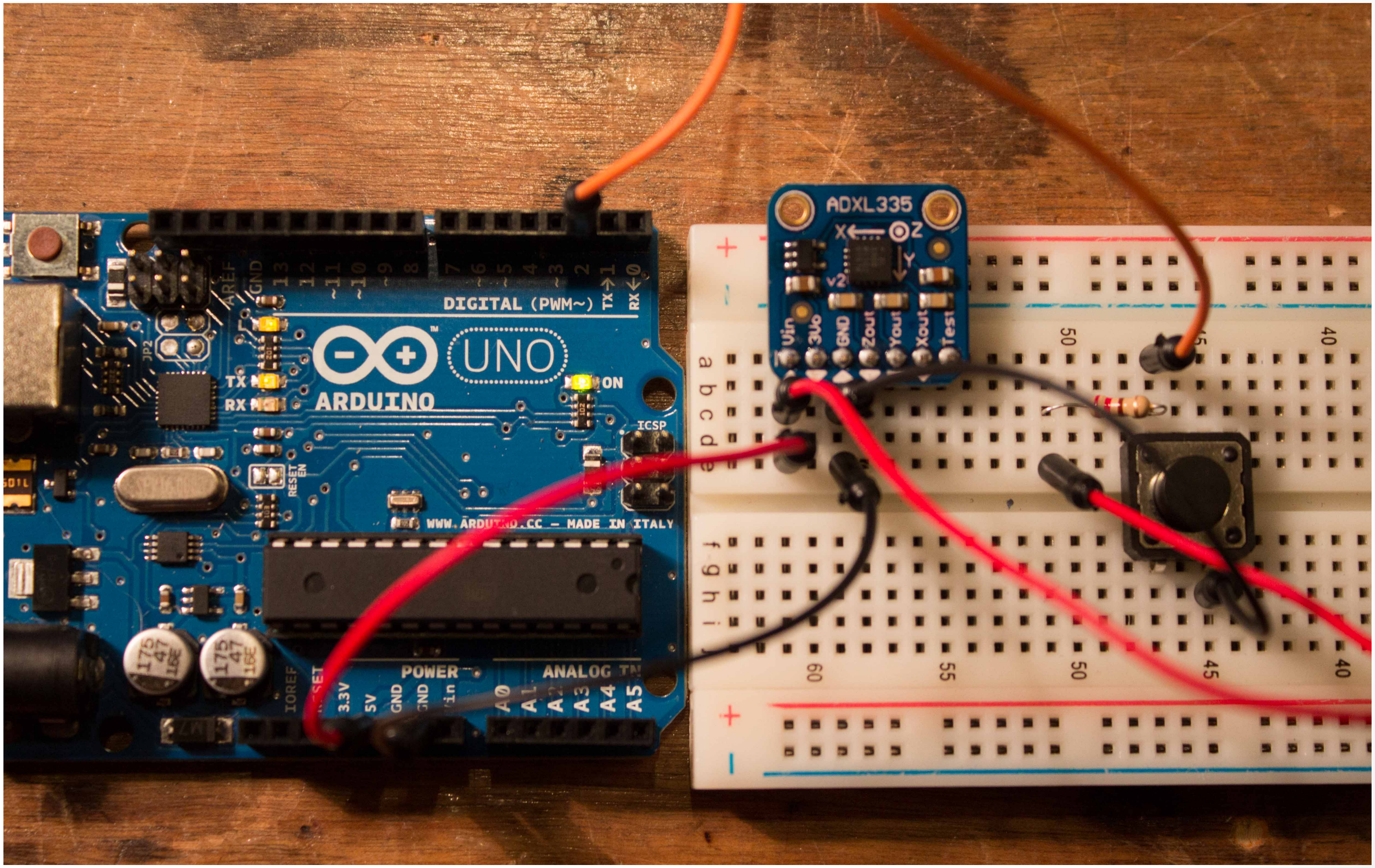
Calibrating Sensors

- Some sensors have natural variances and must be calibrated before use.
- Check the datasheet to see if your sensor needs to be calibrated
- Often the datasheet has information on how to calibrate the sensor

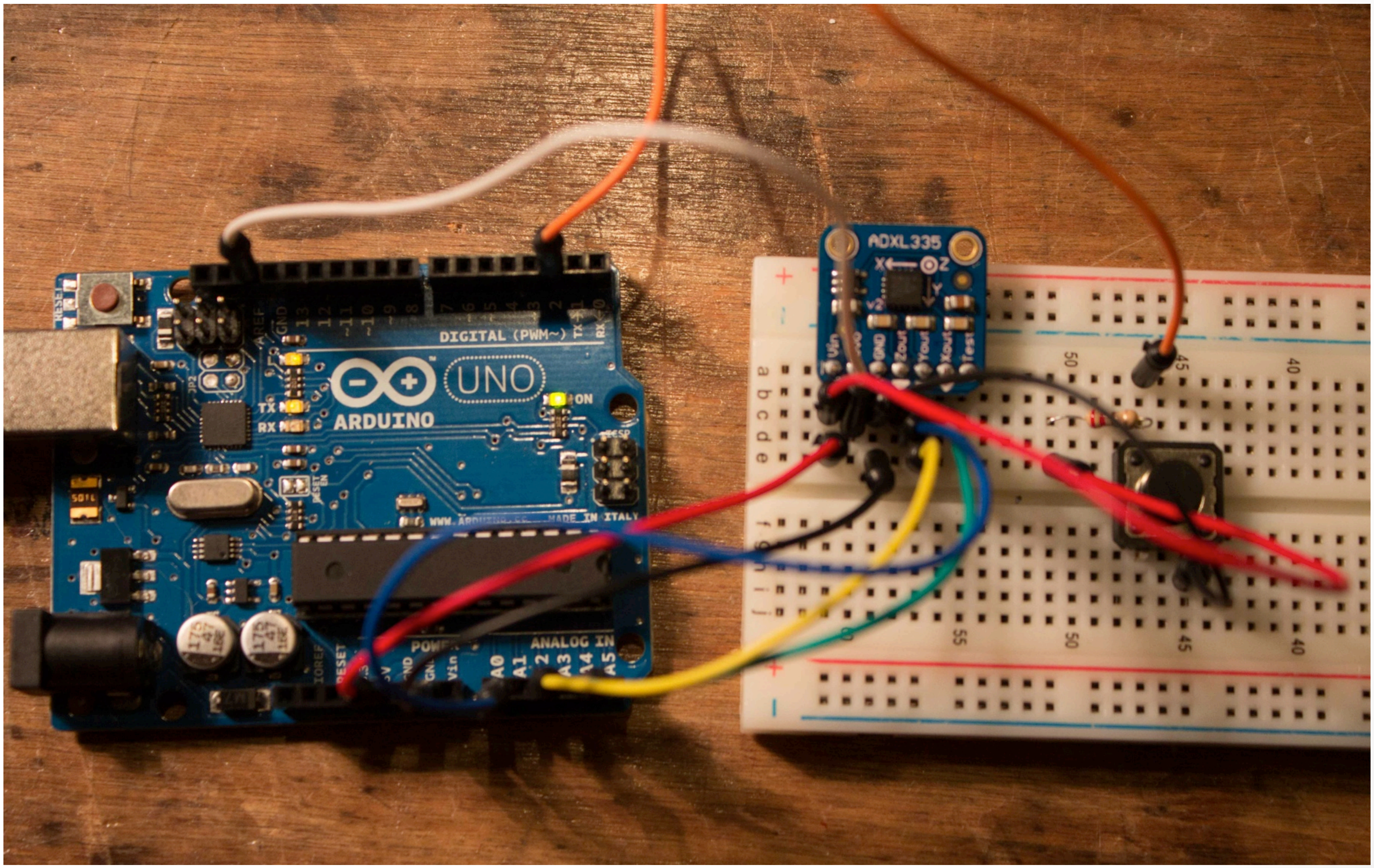
Calibrating the accelerometer

- Use a switch to take samples of the sensor in different orientations
- Test each axis facing up (+1.0g) and facing down (-1.0g)
- Purpose of the switch is to keep the movement of the board does from impacting readings
- Get readings at every axis and use this data in a future sketch
- Code: accelerometer_calibrate

Wiring the switch



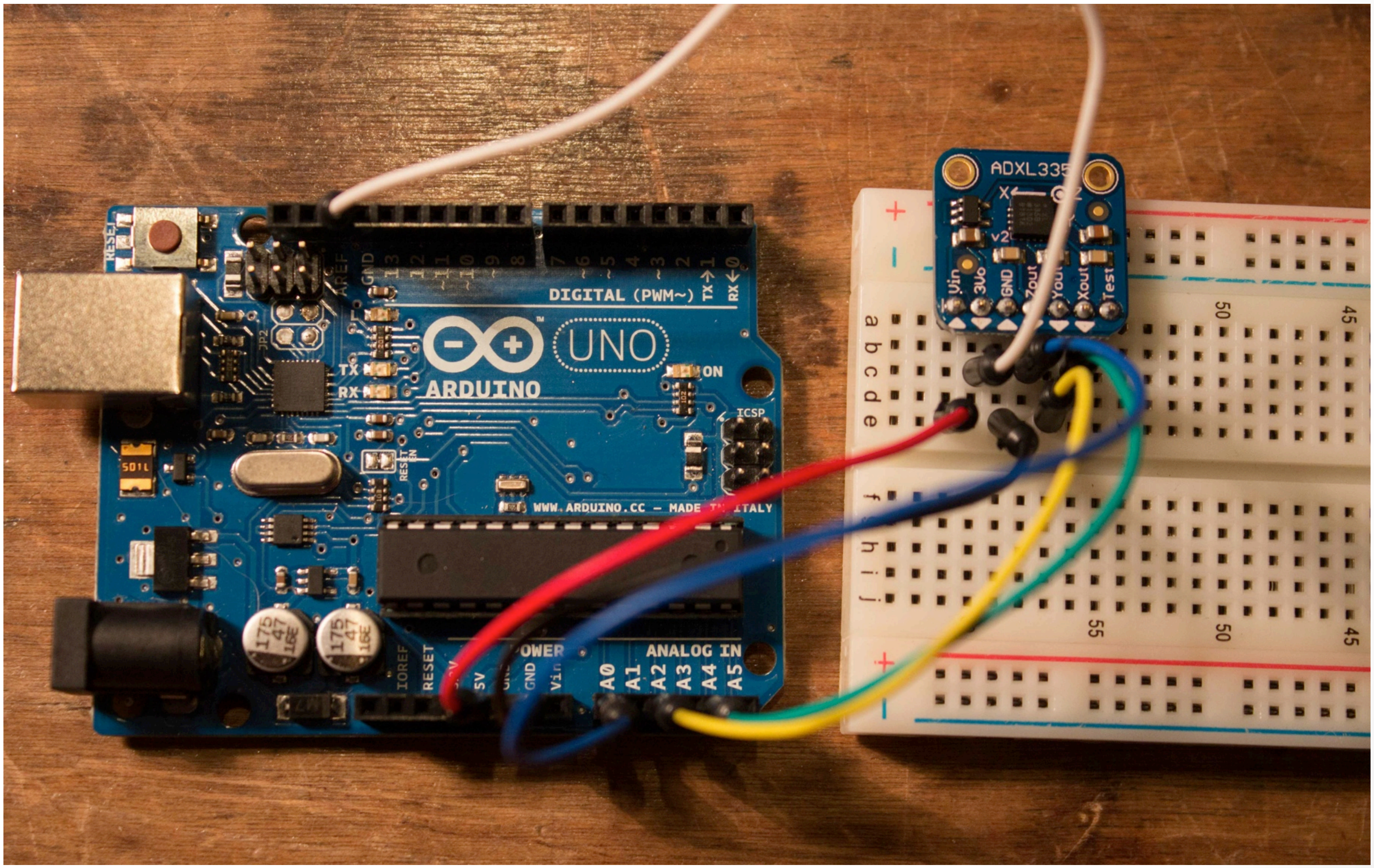
Wiring the accelerometer



Using the accelerometer

- Take the xMin/xMax, yMin/yMax and zMin/zMax values from the calibration sketch.
- Update the values in the accelerometer_demo code to match the values for your accelerometer
- You can (but don't have to) remove the switch

Wiring the accelerometer



Tips for working with sensors

- Read the datasheet!
- Check your wiring. Power? Ground? Sensor output?
- Check the pin you are reading in the code
- Does the pin order matter for this sensor?
- Is the thing the sensor is detecting present?
- Check for tutorials online
- Try, try again

Questions?

- Contact me!
  - miria@nycresistor.com
  - @MiriaGrunick