Photon Analog IO - Input

44-440/640-IoT

Objectives

- Students will be able to:
  - explain how to read analog values using the Photon
  - explain how the TMP36 temperature sensor works, and construct a program that uses it
  - describe how potentiometers and photocells work

Analog Input: ADC

- **analogRead(*pin*)** will convert the voltage (0-3.3V) applied to pin to a digital value in the range 0-4095

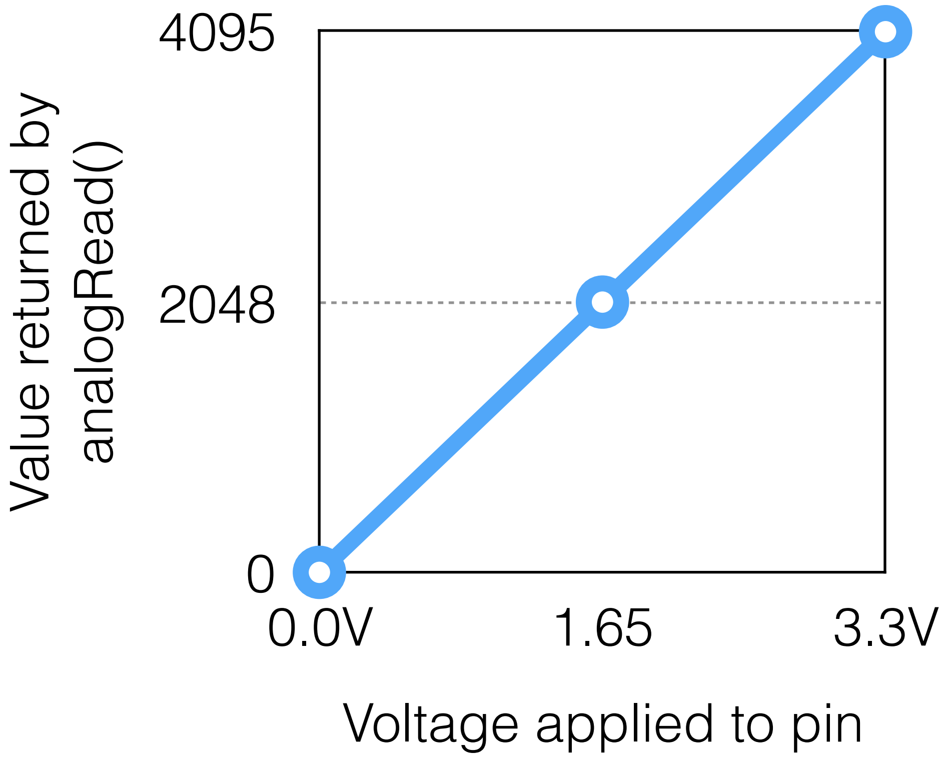
- int analogRead(int *pin*)

  - *pin* - **A0-A7**

  - returns - 0-4095

- Invoke **pinMode(pin, AN_INPUT)** prior to calling analogRead()*

- Lame mnemonic: analogRe<u>ad</u>() -- <u>ADC</u>



Technically not necessary — when analogRead() is called, it automatically calls pinMode(*pin*, AN_INPUT) -- we will do this for consistency's sake. See the docs for details.

Example 1: The TMP36

- The TMP36 is a simple sensor for measuring temperature. It has 3 pins — one connects to Vs, one to ground, and the third outputs a voltage between 0.0V-3.3V.

- Based on two points, (V,C) = (0.2683,-25.0), (1.7,125), we see that, approximately,

  C = 104.6V - 54.4

- **BOTTOM VIEW means that you are "beneath" the TMP36**, looking up: the pins are visible.

- Be sure and hook this up properly -- backwards, and the TMP36 can get dangerously hot 🔥

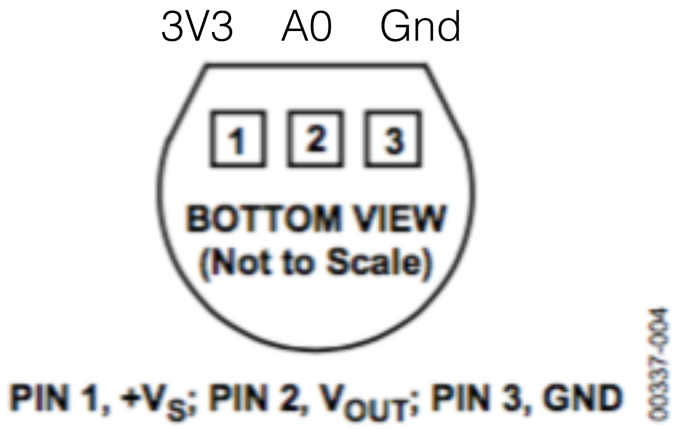


Figure 4. T-3 (TO-92)

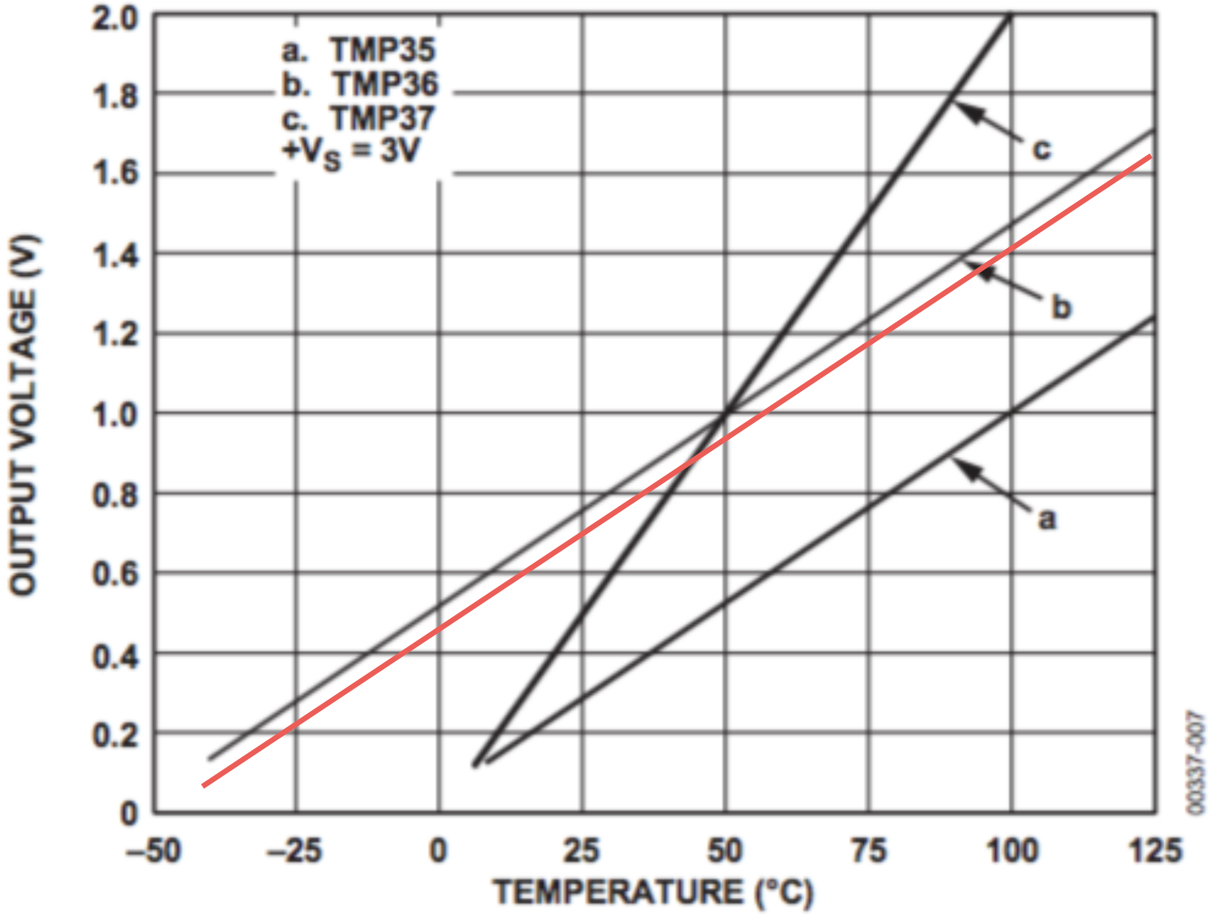


Figure 6. Output Voltage vs. Temperature

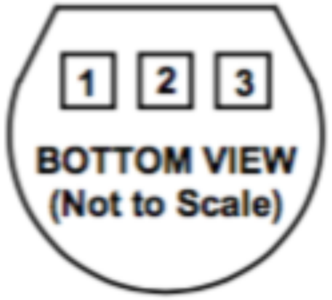
Question: How to Read Voltage?

- analogRead(pin) returns 0-4095 when the (analog) voltage is between 0.0-3.3V. But the TMP 36 graph and formula that we just derived requires that "raw" voltage. Fill in the blanks (do **not** look at the code on the next slide until you have completed this):

```
double voltage = analogRead(pin) / ____ * ____;
```

The TMP36 in Action

3V3 A0 Gnd



PIN 1, +V_S; PIN 2, V_{OUT}; PIN 3, GND

Figure 4. T-3 (TO-92)

```
// tmp36demo.ino


double temperatureC = 0.0;
int analogPin = A0;

void setup() {
    Serial.begin(9600);
    pinMode(analogPin, AN_INPUT);
}

void loop() {
    double voltage = analogRead(analogPin)/4095.0 * 3.3;
    temperatureC = 104.6*voltage - 54.4;
    Serial.println(temperatureC);
}
```

Construct the circuit and implement the code in an app

How the TMP36 Works

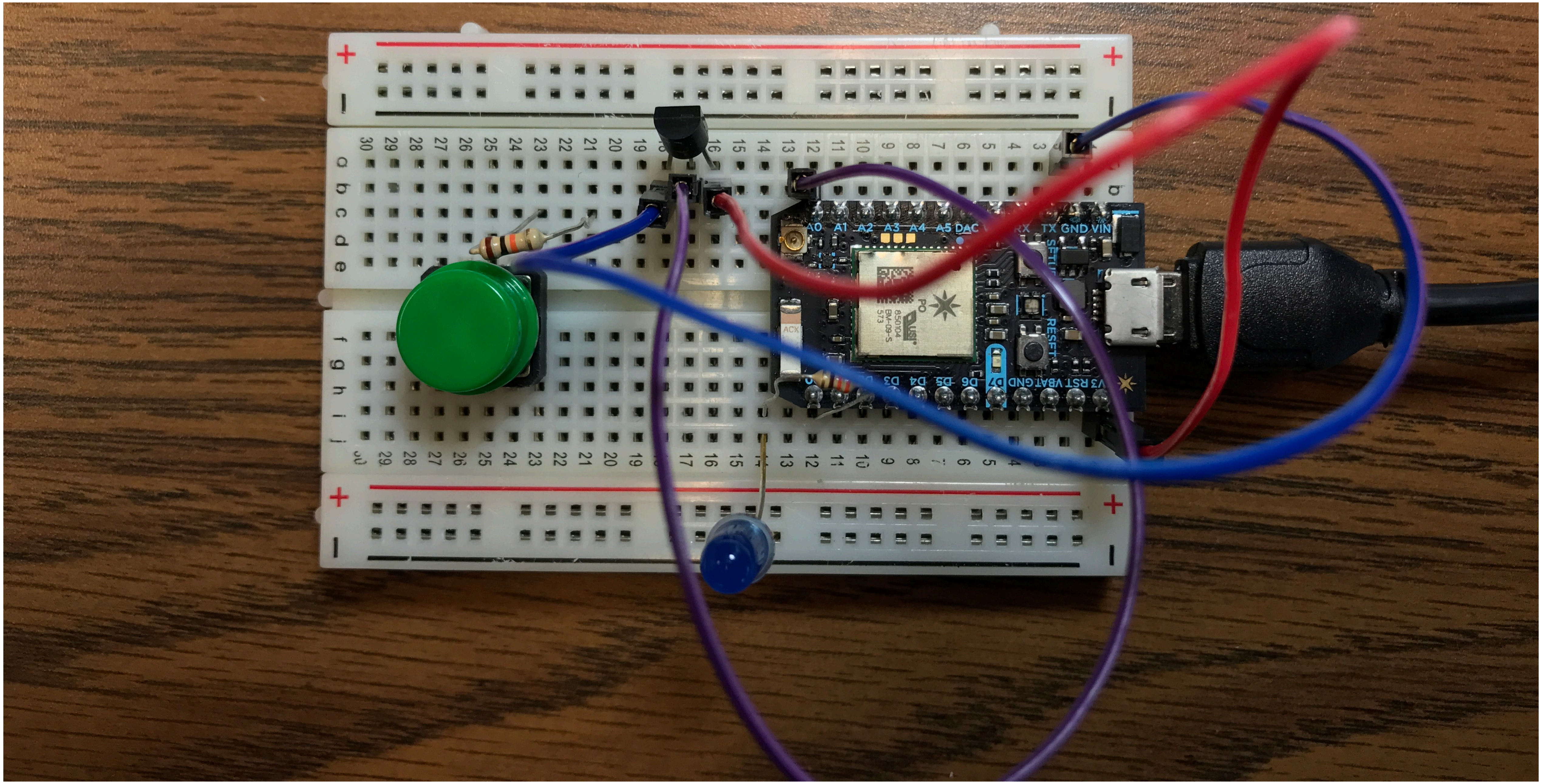
- The TMP36 takes advantage of the behavior of a diode, namely that Vf (the voltage drop across the diode) changes in a quantifiable fashion depending on temperature.

ICE: Is it Cold In Here?

- Construct the circuit to hook up a TMP36, flash the code, and let's see how cold it is
    - in various parts of the room
    - outside 😍, time/weather permitting — but you will have to stay within the confines of the access point* and your computer
- Q: Now that we have data, what can we do with it?
  A: As we shall see, a *lot*

*actually, no: if you wander outside, you will fall off the Particle Cloud, but your firmware should still run

The TMP36, Connected

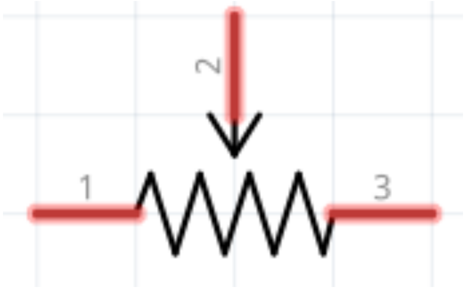


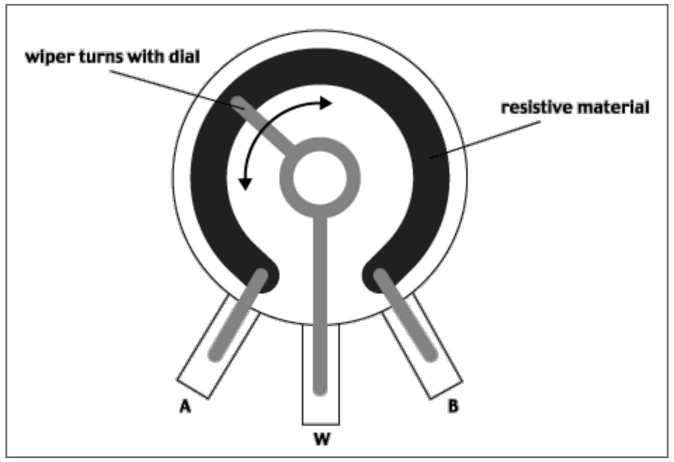
Potentiometers

- Standard resistors have a constant resistance, but *some* electrical components change resistance depending upon certain factors.

- A **potentiometer** basically forms a voltage divider, in which the resistors on both side of the w change (but sum to the same value)

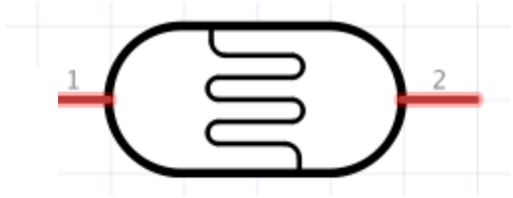
- You've used a pot anytime you've used a dimmer switch

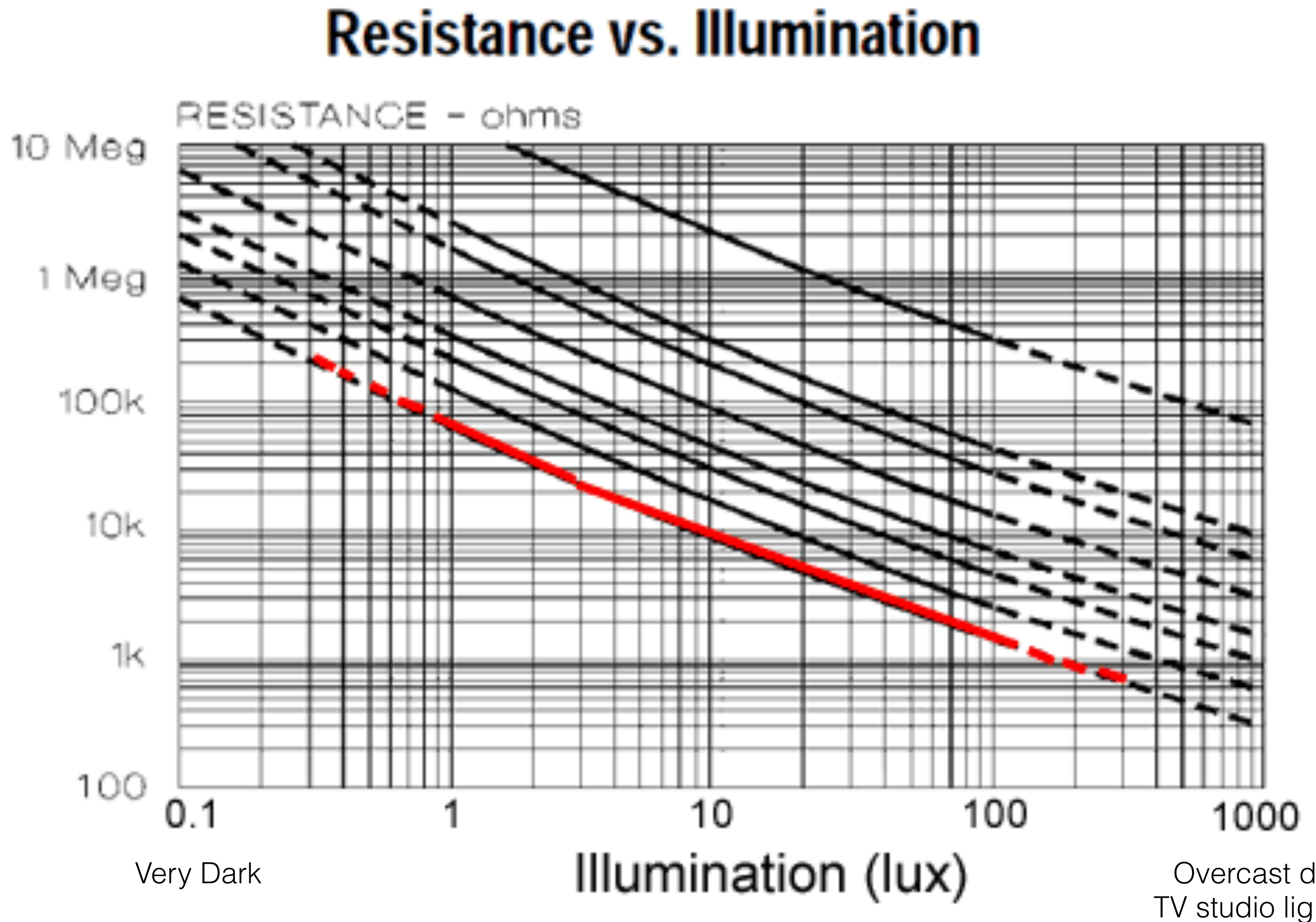




Photoresistors

- A **photoresistor** is a resistor whose resistance decreases with light intensity





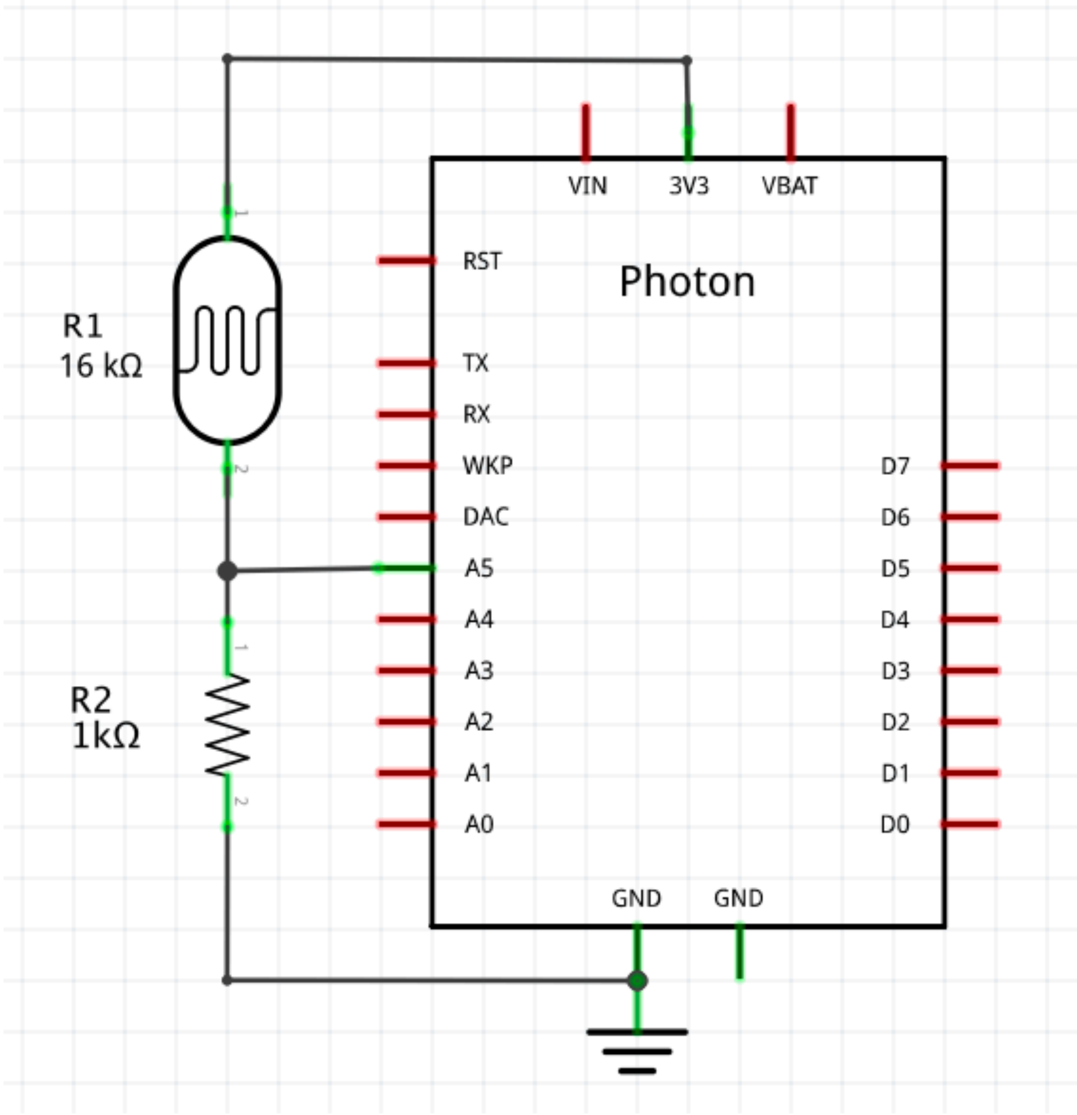
Very Dark

Overcast day;
TV studio lighting

https://learn.adafruit.com/photocells/measuring-light

Example 2: a Light Detector

- We have two resistors in the circuit, with a "tap" between them.
- Q: What is the name of this circuit?
- Q: What is the voltage at A5, assuming 3.3V?
- Q: How much light is currently illuminating the photoresistor? Use the red curve on the graph to estimate this.



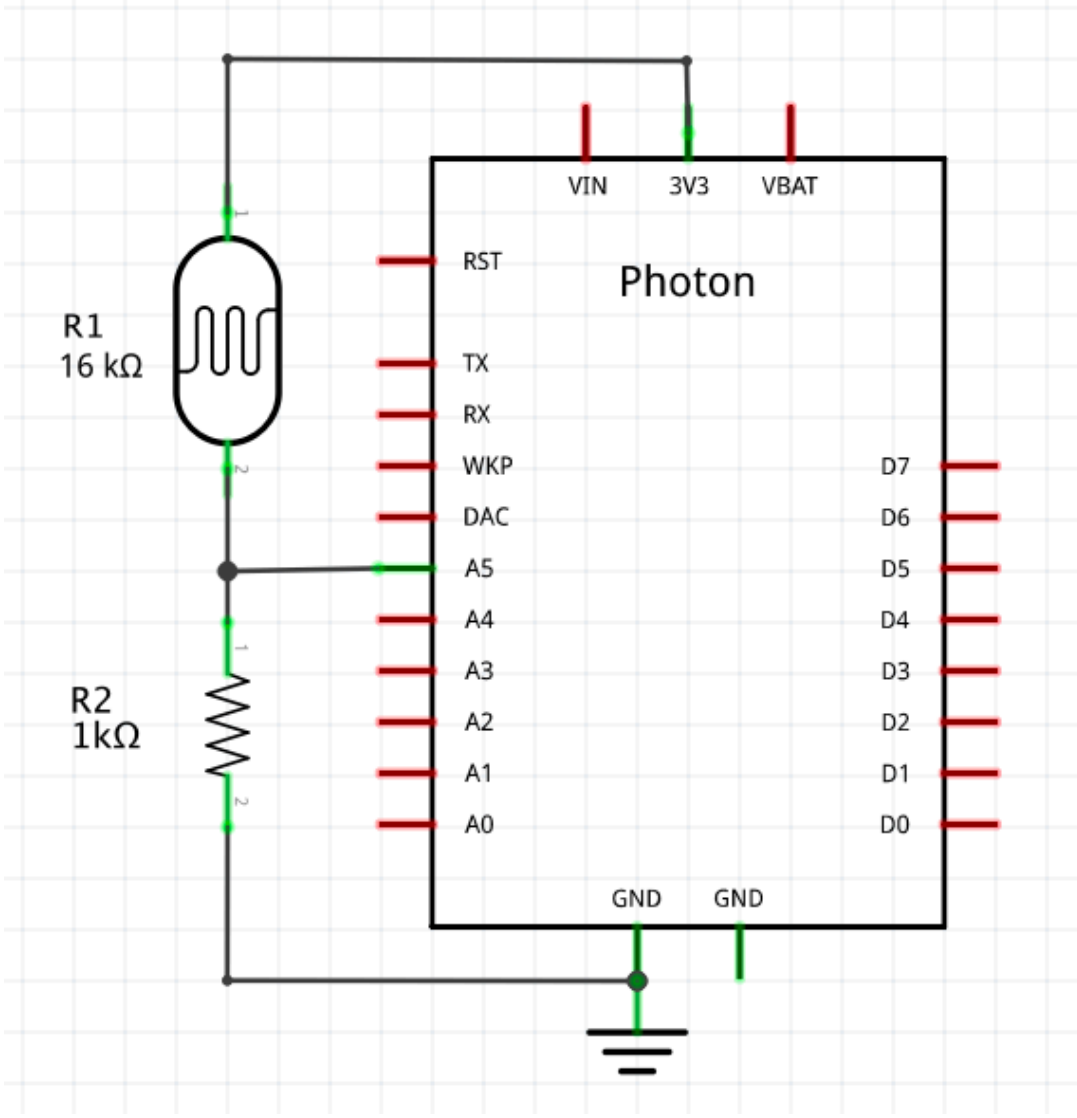
Example 2: a Light Detector

- We have two resistors in the circuit, with a "tap" between them.

- A: Voltage Divider

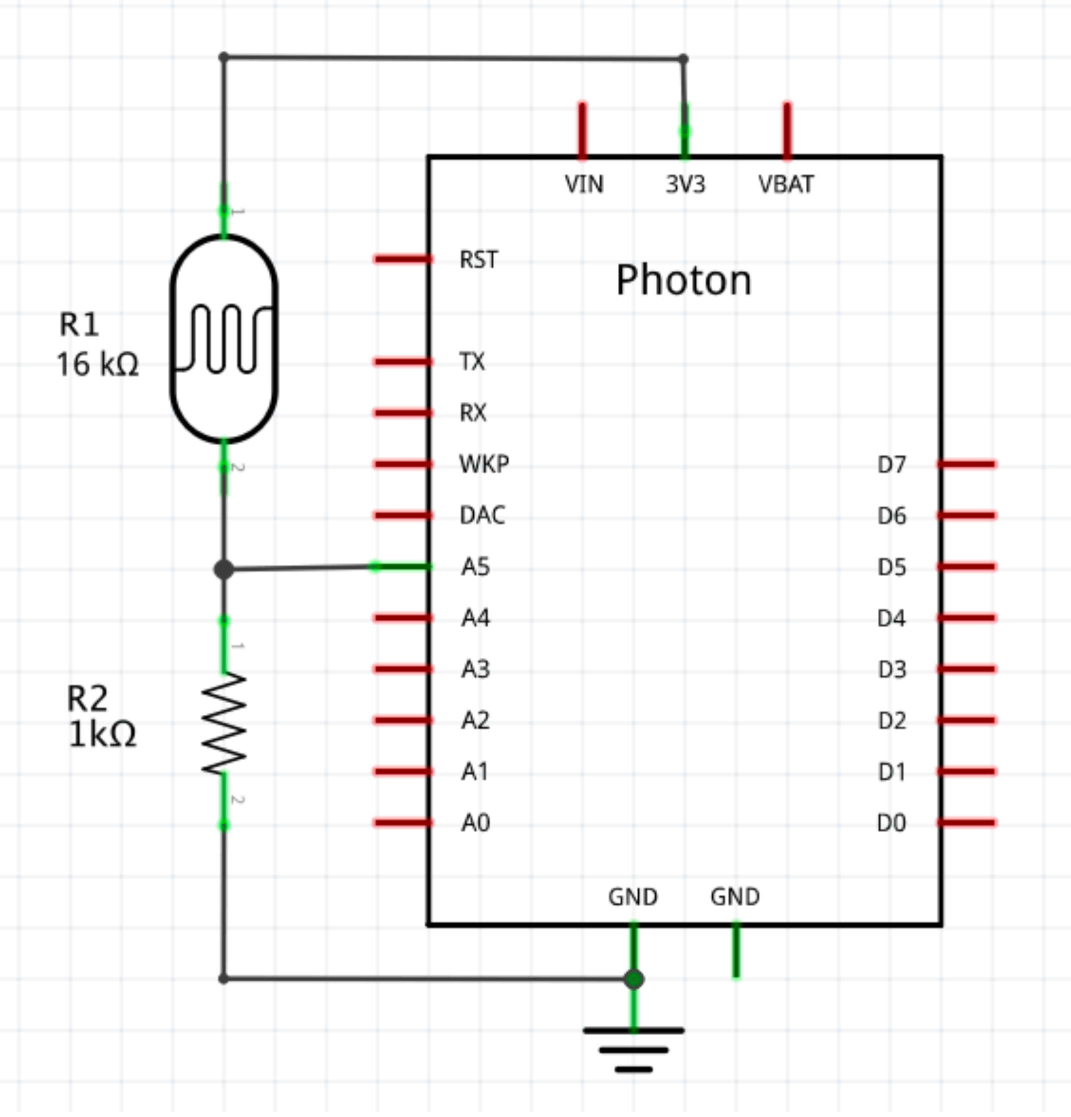
- A: Vout = Vin * R2/(R1 + R2)
  = 3.3 * 1000/(17000)
  = 0.2V

- A: Approximately 10 lux

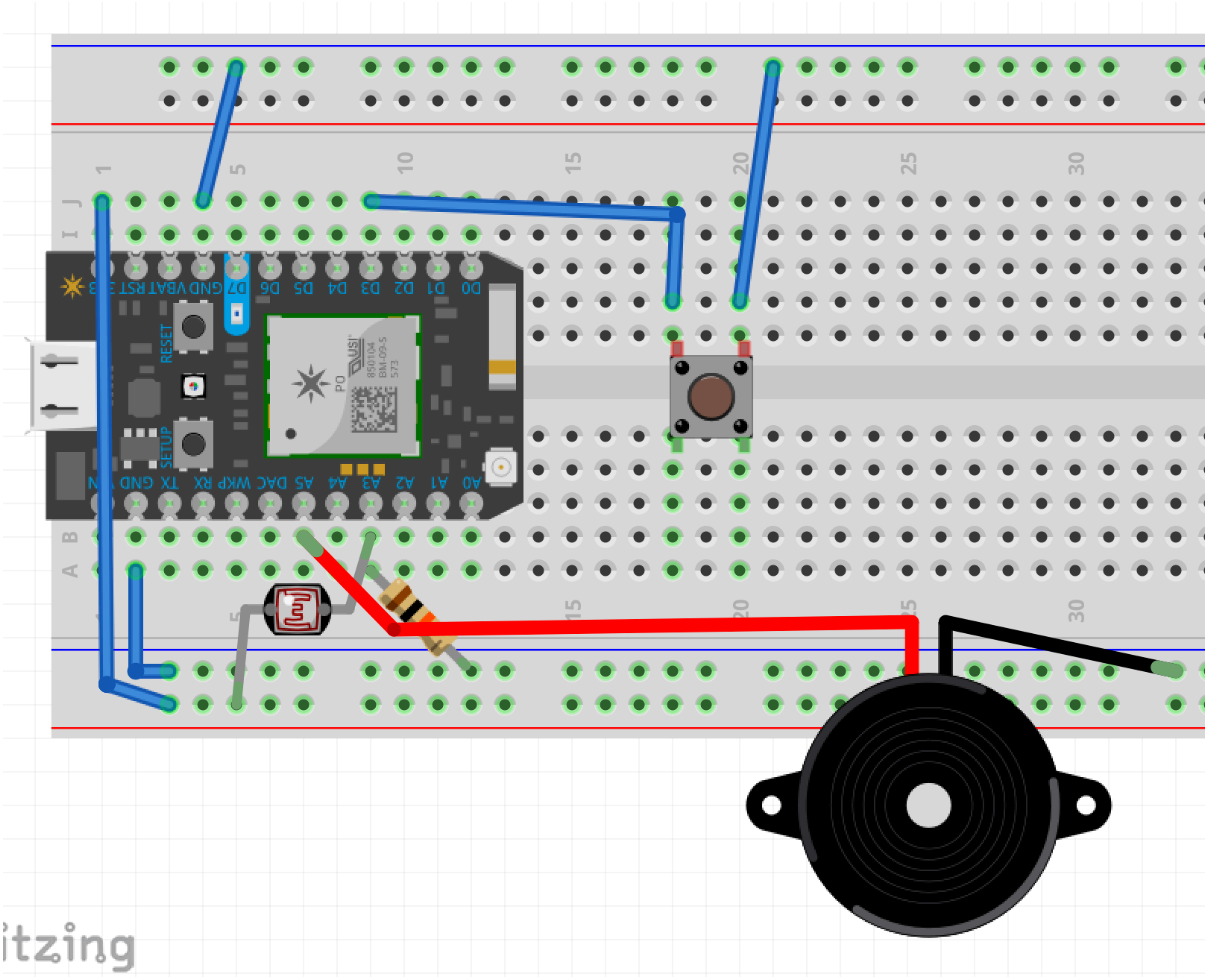


Example 2: a Light Detector

```
// thereallygreatlightmeter.ino
// Make: Getting Started with the Photon, Simon Monk.
int reading = 0;
double volts = 0.0;
int analogPin = A5;

void setup() {
   pinMode(analogPin, AN_INPUT);
   Particle.variable("volts",volts);
    Serial.begin(9600);
}

void loop() {
  reading = analogRead(analogPin);
  volts = reading * 3.3 / 4095.0;
}

int ariseMinions(String command){
    Serial.println(String(volts));
    return 1;
}
```



```
// curl -v https://api.spark.io/v1/devices/xxxxx/reading?access_token=xxxxx
// curl -v https://api.spark.io/v1/devices/xxxxx/volts?access_token=xxxxx
```


1
5
10
15
20
25
30
J
I
B
A
D0
D1
D2
D3
D4
D5
D6
D7
GND
VBAT
RST
3V3
RESET
SETUP
A0
A1
A2
A3
A4
A5
DAC
WKP
RX
TX
GND
VIN
itzing

Buzzer and Voltage

```
double voltage;
const int button = D3;
const int buzzer = A5;
const int photoResistor = A3;
const int builtInLED = D7;

void setup() {
    pinMode(button, INPUT_PULLUP);
    pinMode(builtInLED,OUTPUT);
    pinMode(photoResistor, AN_INPUT);
    pinMode(buzzer, OUTPUT);
    Particle.variable("voltage",voltage);
    Serial.println(9600);
}

void loop() {
    voltage = analogRead(photoResistor)/4095.0 * 3.3;
    Serial.println(voltage);
    if(digitalRead(button) == LOW){
        analogWrite(buzzer, 128, 440);
        delay(500);
        analogWrite(buzzer,0,440);
        delay(500);
    }
}
```

NB: Not all pins work with PWM. See Photon Datasheet, esp. note [3] in the Peripherals and GPIO section.

Exercises

- Add an alarm to your light detector circuit. The mini-speaker in your kit needs alternating current, but fortunately PWM is available!

- See docs.particle.io for details

- Do something amusing and possibly recursive involving a photoresistor and LEDs (a dark room/closet/box may be helpful here)

- Sunflowers point towards the sun -- could you do something with several photoresistors to do the same (or at least identify where the sun is)?

Cheatsheet

	Digital Input	Digital Output	Analog Input	Analog Output	Analog Output (PWM)
Pins	D0-D7, A0-A7, DAC, WKP, RX, TX	D0-D7, A0-A7, DAC, WKP, RX, TX	A0-A7	DAC1, DAC2	D0-D3, A4, A5, WKP, RX, TX*
Pin Mode	INPUT	OUTPUT	AN_INPUT	OUTPUT	OUTPUT
Methods	digitalRead()	digitalWrite()	analogRead()	analogWrite()	analogWrite()

*PWM is duplicated on A5/D2 (can't use both for independent PWM output); likewise with A4/D3)

Resources

- https://learn.sparkfun.com/tutorials/analog-vs-digital
- https://learn.sparkfun.com/tutorials/pulse-width-modulation
- http://www.eetimes.com/document.asp?doc_id=1274544
- https://docs.particle.io/reference/firmware/electron/#analogwrite-pwm-
- https://learn.adafruit.com/tmp36-temperature-sensor/using-a-temp-sensor
- http://www.analog.com/media/en/technical-documentation/data-sheets/TMP35_36_37.pdf