Photon Digital IO -- Output

44-440/640-IoT

Objectives

- Students will be able to:
  - explain the difference between digital and analog
  - explain the purpose of, and incorporate functions pinMode() and digitalWrite() in code
  - realize the importance of reading the documentation
  - attach external LEDs to the Photon

Digital versus Analog

- **Digital** means that a quantity can take on one of a finite number of values

- **Analog** means that a quantity can take on any value in a continuum.

- Classify the following as either digital or analog:

    - Day of the month, Outside temperature, Brightness, Color, Status of an on-off switch, Audio Volume, Status of a dimmer switch

- Give two other examples of things that are digital and analog

- In this presentation we will be dealing strictly with **digital**, specifically, signals that can take on one of two possible values

Digital Circuits

- Circuits built with just **resistors**, **capacitors**, **diodes**, inductors and operational amplifiers are typically **analog**.

- Those built with **transistors and logic gates**, or **microcontrollers** (such as the Photon), are **digital**.

- Digital circuits usually restrict values to **2** logic levels: a **high** voltage (around **3.3V**, in our case) represents one value, and a **low** voltage (around **0V**, usually) a second.

- These logic values are usually called off/on, low/high, false/true, and, curiously, high voltage does not always map to the second of the pair (on, high or true); it depends on the logic system being used.

Digital IO on the Photon

- All the GPIO (General Purpose Input Output) pins (D0-D7, A0-A7, DAC, WKP, RX, TX) can be used for both digital input and output. The values that the pin can take on are **LOW** (0V) or **HIGH** (3.3V).

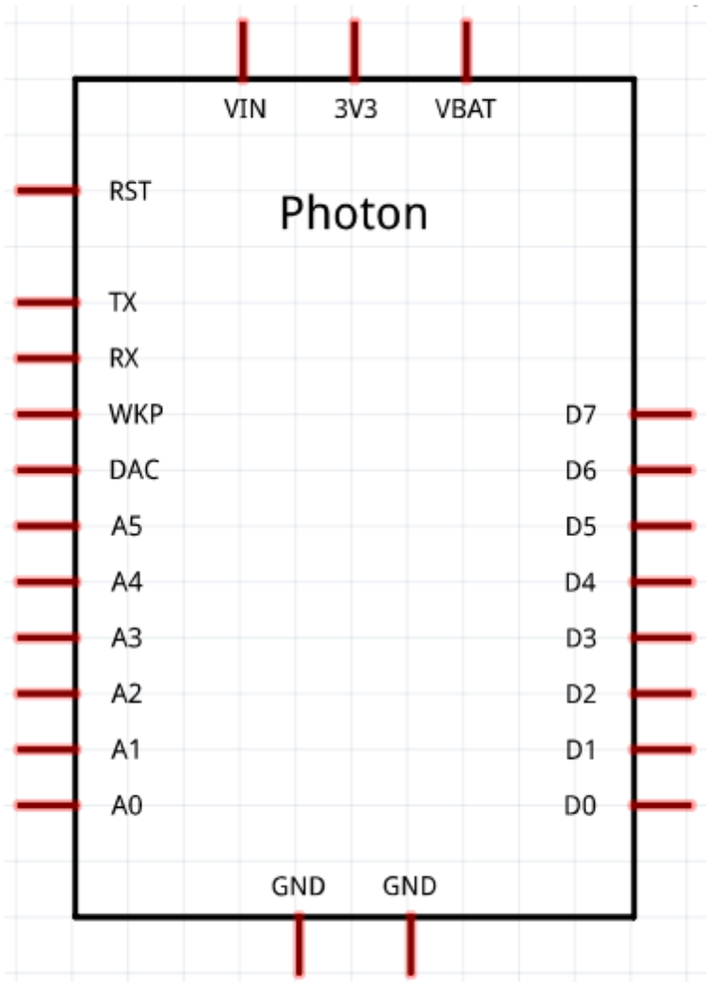
- A pin must be set up to do either input or output, using **pinMode(pin, mode)**. It takes two arguments,

  - **pin**: D0-D7, A0-A7,

  - **mode**: INPUT, INPUT_PULLUP, INPUT_PULLDOWN, OUTPUT

- When in output mode, digitalWrite() can be used to write either LOW (0V) or HIGH (3.3V) to the pin.

- When in input mode, digitalRead() returns either LOW or HIGH

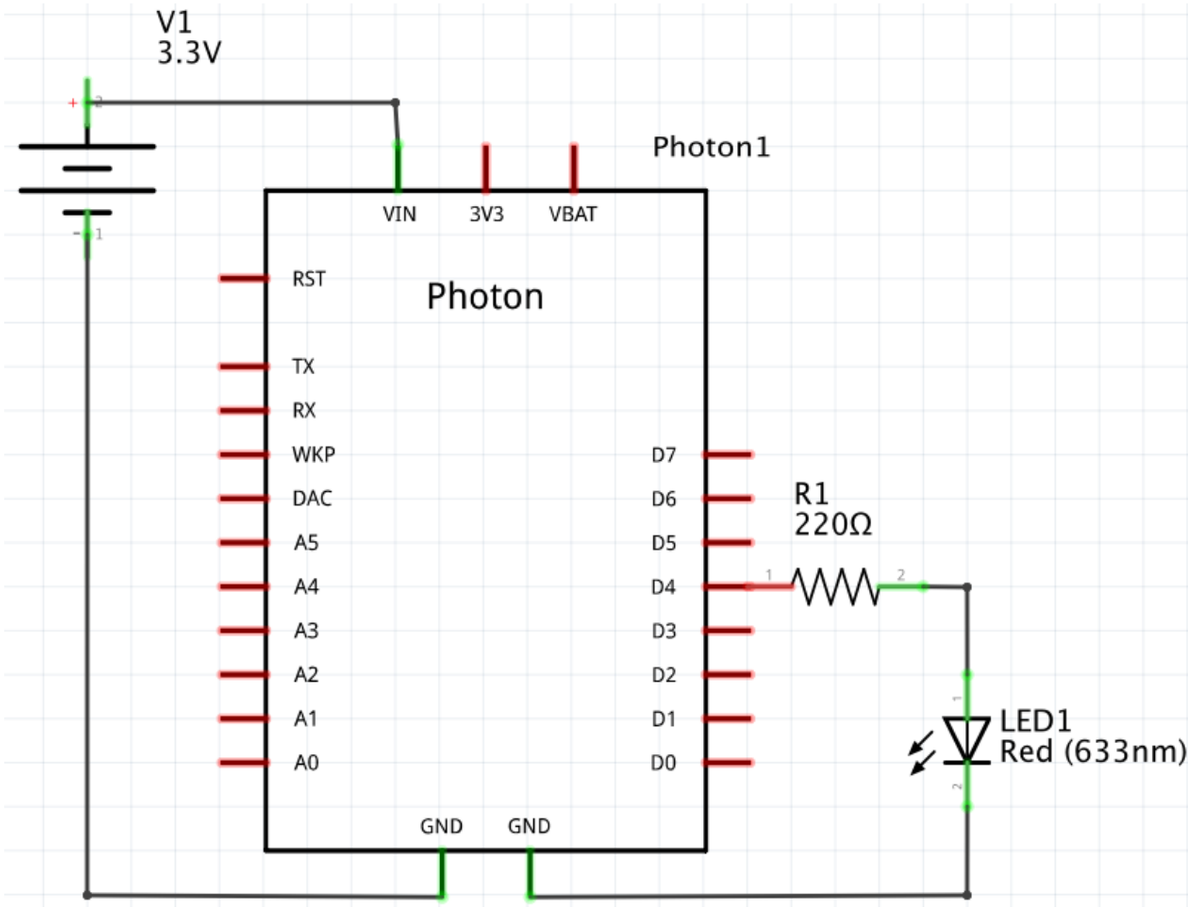


Hook 'em Up!: LEDs

- For output, a digital pin will be at 3.3V when it is HIGH, and 0V when it is low. So, we need to connect the LED's anode to the pin, and the cathode to ground (or something eventually leading to ground.

- LEDs also can be current-hungry, and so we will *always** use them in conjunction with a resistor

- But how do we choose the resistor?



*Eliminating the resistor *may* work, or may damage the LED and/or Photon 😟 by pulling too much current. See this discussion (especially the highly rated answer).

Current Limiting Resistor

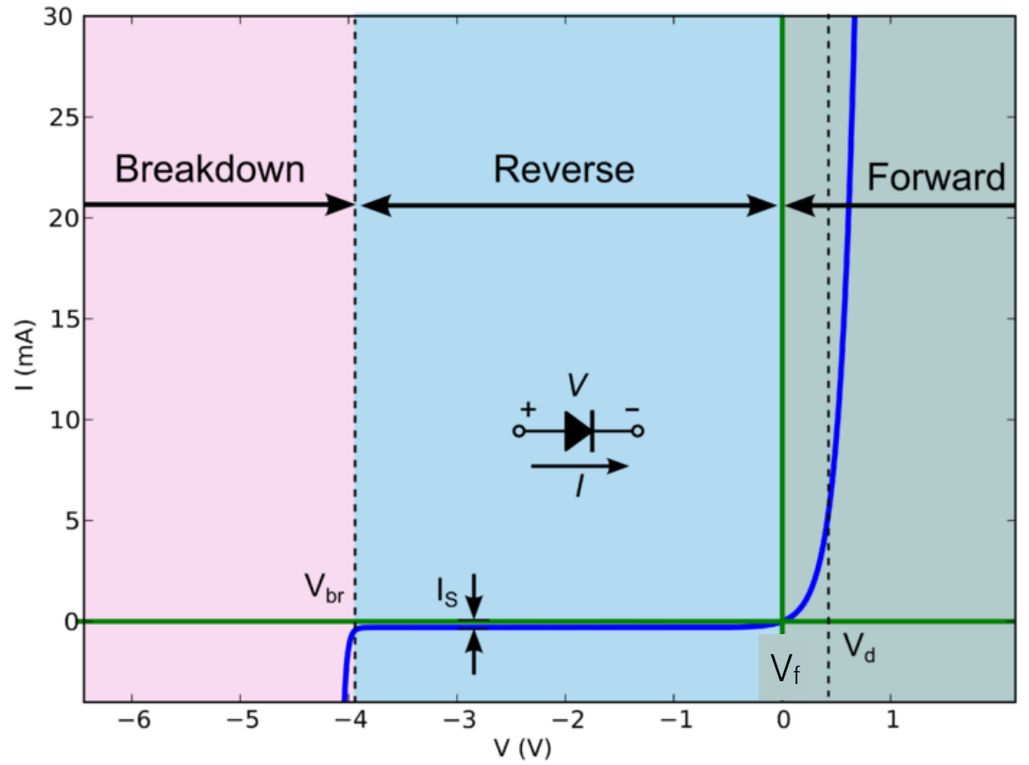
- The forward voltage, **Vf**, is the amount of voltage required to turn on an LED. If voltage exceeds Vf the LED lights up; otherwise, it stays dark.

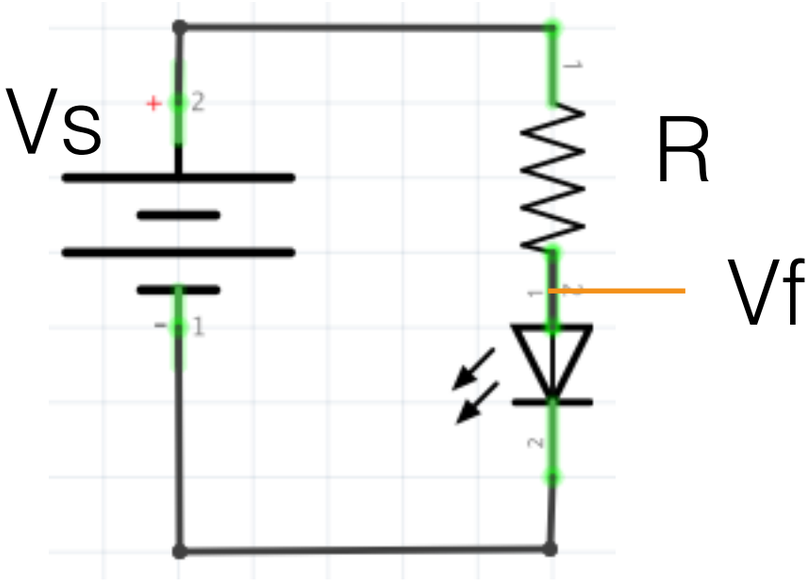
- For our LEDs, Vf is 1.8-2.2V (see the SparkFun datasheet)

- Problem: as voltage increases, so does current, and too much current could melt the LED 😫

- We must therefore limit the current to a safe value, known as **Ir** on the LED data sheet

- That's the job of the **current limiting resistor**.





Choosing a Current Limiting Resistor

1. We need the voltage at B to be Vf; at A it was Vs

2. Voltage at B = Voltage at A - Voltage Drop Across R
   Vf = Vs - Voltage Drop Across R
   VoltageDrop Across R = Vs - Vf

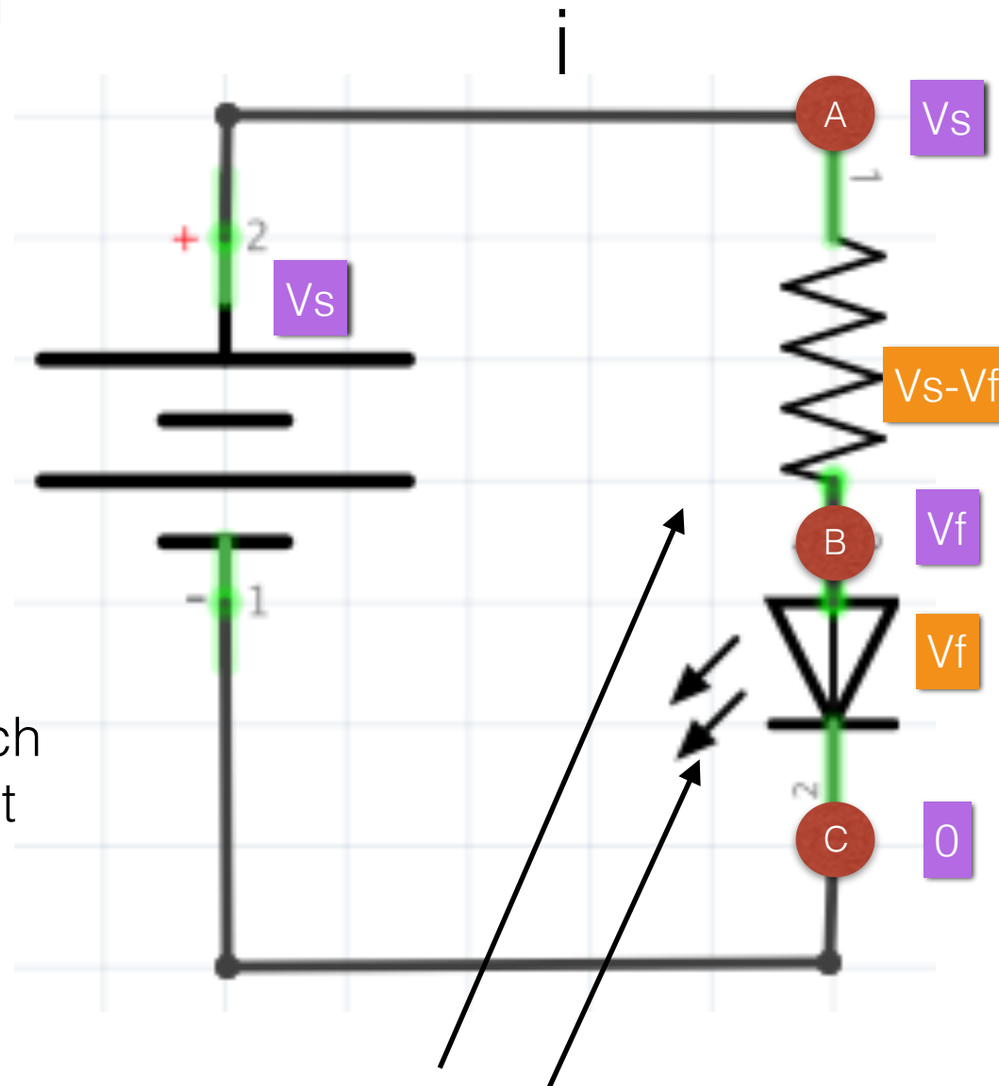
3. The voltage at C needs to be 0 (ground always has a voltage of 0), so the voltage drop across the LED is Vf

4. The resistor and the LED are in series, so current (which we need to be Ir) will be the same throughout the circuit

5. From Ohm's law **R = (Vs-Vf)/Ir**

6. We can get an estimate for red LEDs here, so (3.3-1.8)/0.020 = 75 Ω.

7. We probably can't find 75 Ω: in the class, we will use 330 Ω



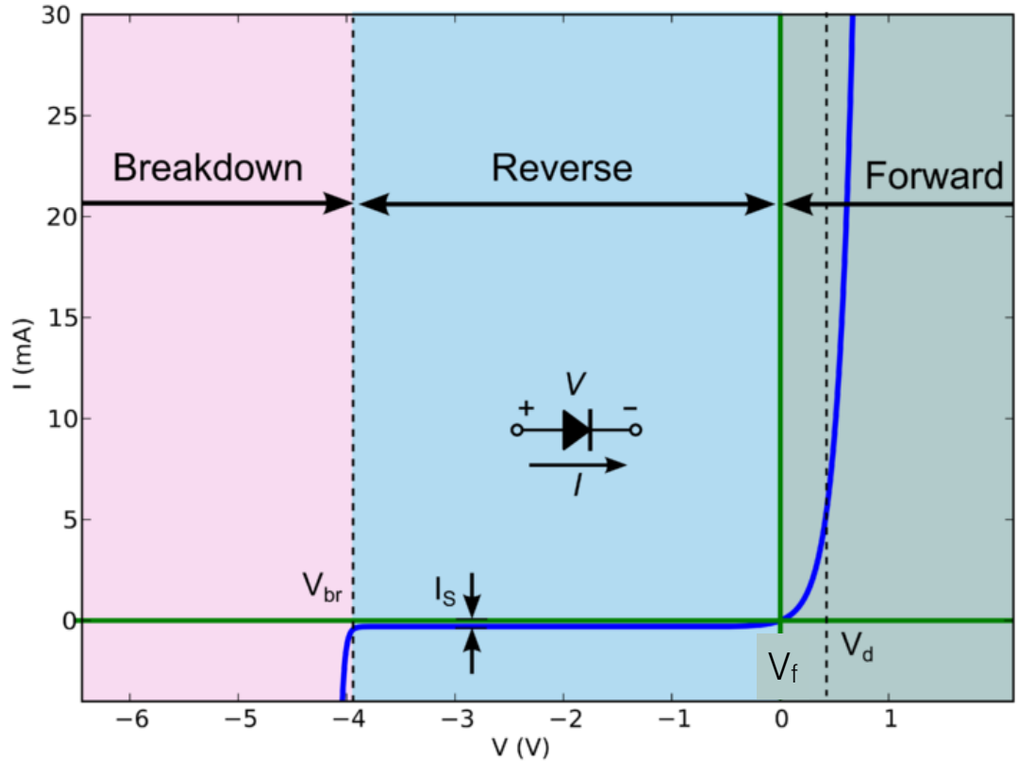
LED Resistance

- Does an LED have resistance? Yes, otherwise we would not have a voltage drop: but it is not very large

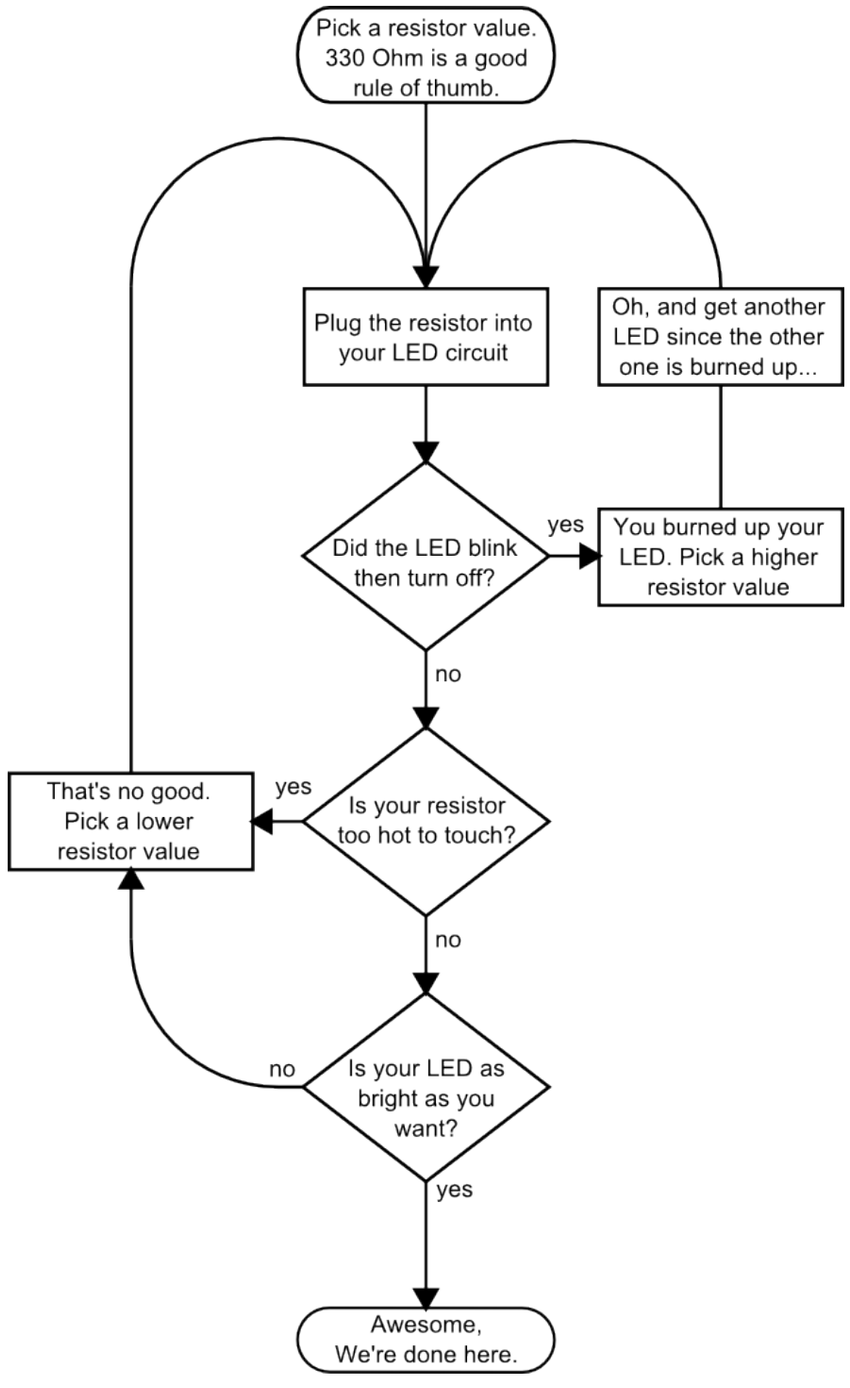
- Unlike a resistor, where R = V / I, the resistance of an LED is not linear, but past Vd, it is approximately so: the current increases to keep the voltage at Vd

- The slope of the line past Vd gives the resistance.

- See this discussion for details



Choosing a Current Limiting Resistor: The Ad hoc Approach

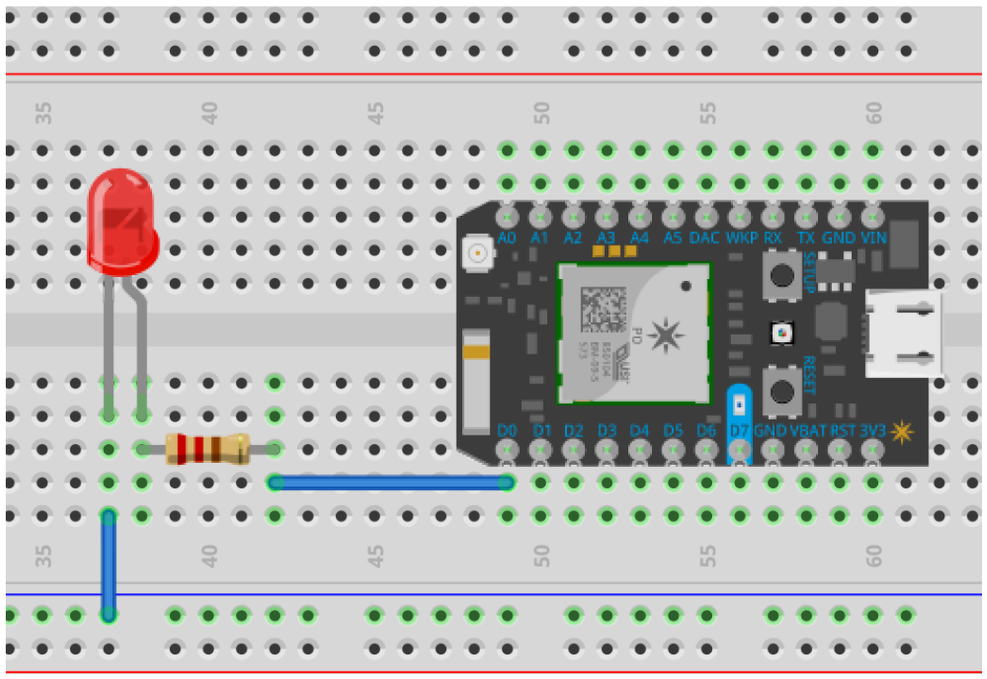


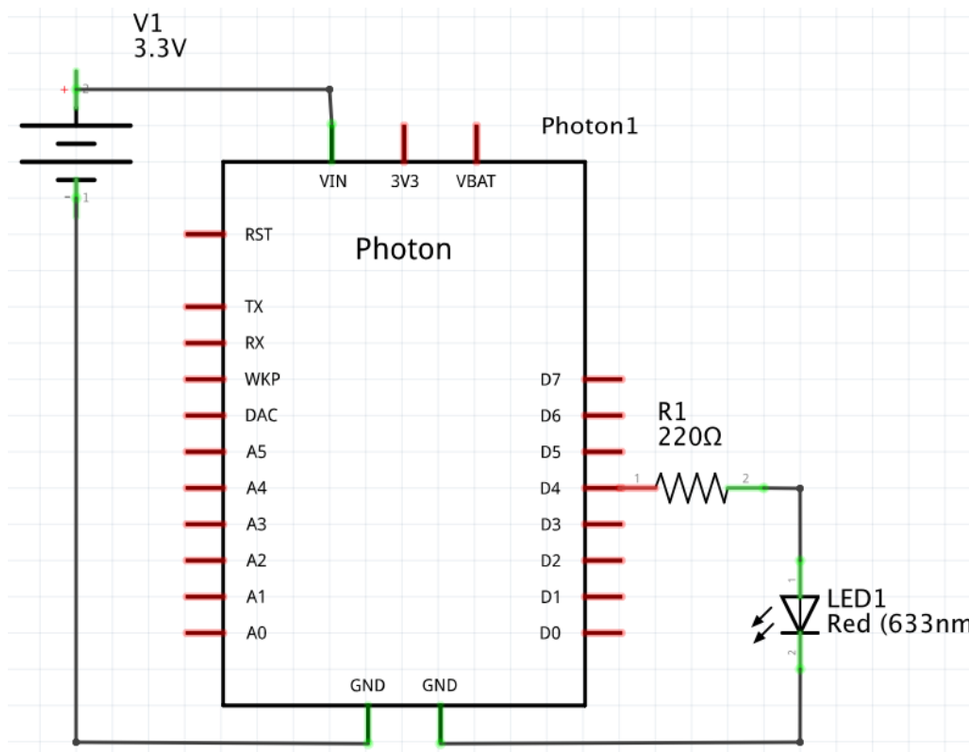
Techy Aside: Typical LED Characteristics

Typical LED Characteristics			
Semiconductor Material	Wavelength	Colour	V_F @ 20mA
GaAs	850-940nm	Infra-Red	1.2v
GaAsP	630-660nm	Red	1.8v
GaAsP	605-620nm	Amber	2.0v
GaAsP:N	585-595nm	Yellow	2.2v
AlGaP	550-570nm	Green	3.5v
SiC	430-505nm	Blue	3.6v
GaInN	450nm	White	4.0v

ICE: Setting up the Breadboard

- Locate the following components:
  - Photon, Resistor, LED, jumper wires, breadboard
- In class, we will work through the mechanics of hooking up the Photon.





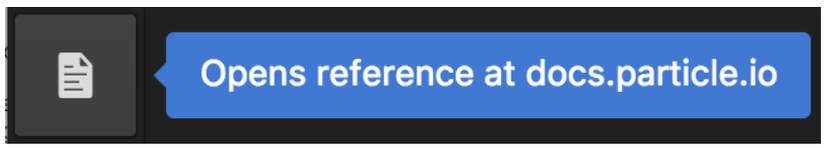
ICE: A Wee Bit of Code

```
int led1 = D0; //D0, D7, OUTPUT, HIGH, etc. are all defined in
int led2 = D7; // an Arduino.h file (?) that gets linked in
int time = 1000;
void setup() {
  pinMode(led1, OUTPUT); // now D0 can only output
  pinMode(led2, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH); // turn it on
  digitalWrite(led2, HIGH);
  delay(time);                  // in ms, of course

  digitalWrite(led1, LOW); // turn it off
  digitalWrite(led2, LOW);
  delay(time);
}
```

Fun fact: Code on a micro controller is called firmware, even though it really isn't.

ICE: Randomness

- Consulting the reference docs, modify the previous so that it blinks on and off for a random amount of time (between 50-500 ms)




```
int led1 = D0; //D0, D7, OUTPUT, HIGH, etc. are all defined in
int led2 = D7; // an Arduino.h file (?) that gets linked in
int time = 1000;
void setup() {
  pinMode(led1, OUTPUT); // now D0 can only output
  pinMode(led2, OUTPUT);
}

void loop() {
  digitalWrite(led1, HIGH); // turn it on
  digitalWrite(led2, HIGH);
  delay(time);              // in ms, of course

  digitalWrite(led1, LOW); // turn it off
  digitalWrite(led2, LOW);
  delay(time);
}
```

Digital Output Exercises

- Hook up 3 LEDs and 3 resistors, using D0-D2.
  - Make them blink simultaneously
  - Make them blink in order
  - Make them blink 10 times then stop
  - Make them count from 0-7 digitally (000, 001, ... 111), where 1 means lit, 0 means unlit)
- More, pending ...

Resources

- https://www.sparkfun.com/tutorials/219
- https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds
- http://www.electronics-tutorials.ws/diode/diode_8.html
- https://learn.sparkfun.com/tutorials/pull-up-resistors
- http://apple.stackexchange.com/questions/124152/how-do-you-paste-syntax-highlighted-code-into-keynote-13
- http://markup.su/highlighter/
- http://www.cc.gatech.edu/~hadi/teaching/cs3220/02-2015fa/doc/debounce.pdf
- https://learn.adafruit.com/all-about-leds/forward-voltage-and-kvl
- http://dangerousprototypes.com/docs/Basic_Light_Emitting_Diode_guide
- https://www.baldengineer.com/led-basics.html
- https://electronics.stackexchange.com/questions/76367/accounting-for-led-resistance [what is an LED's resistance?]