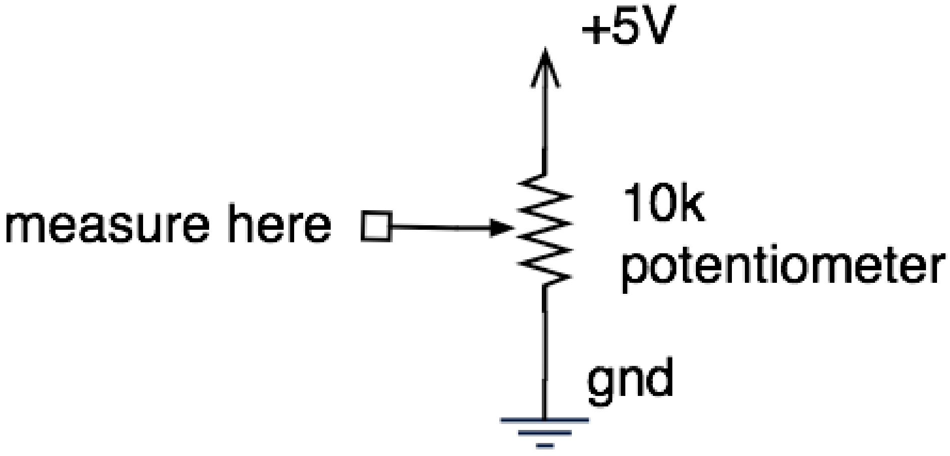
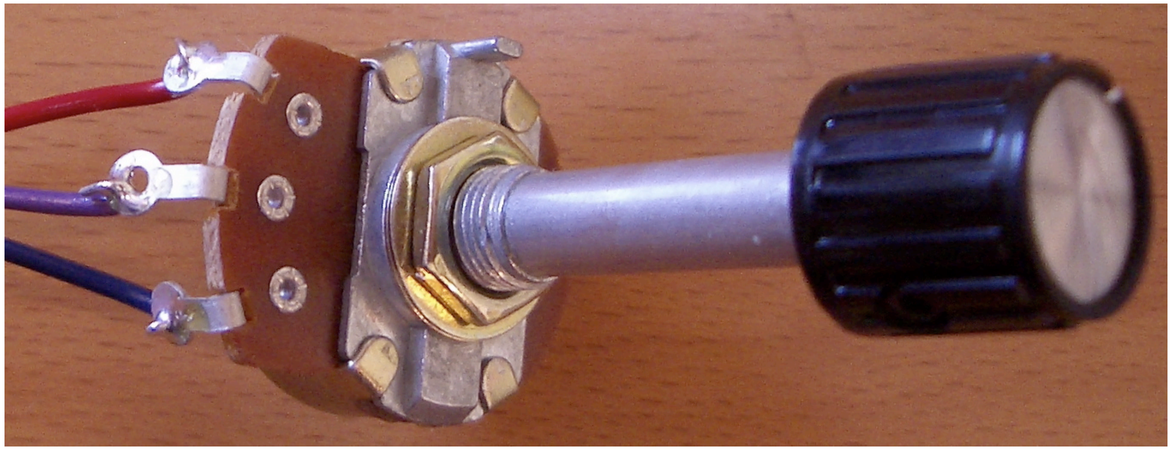
Analog Input

Sure sure, but how to make a varying voltage?

With a *potentiometer*. Or just *pot*.





+5V–

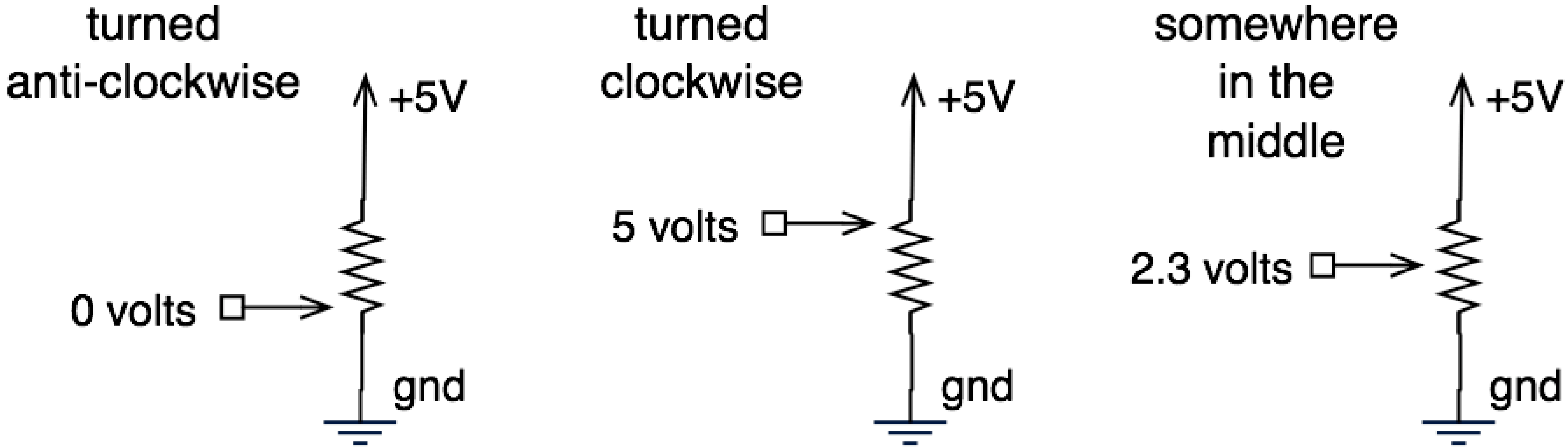
measure–

gnd–

Color coding: red goes to power, blue to ground, purple to ‘measure here’ (it’s a mix, see?)

Potentiometers

Moving the knob is like moving
where the arrow taps the voltage on the resistor



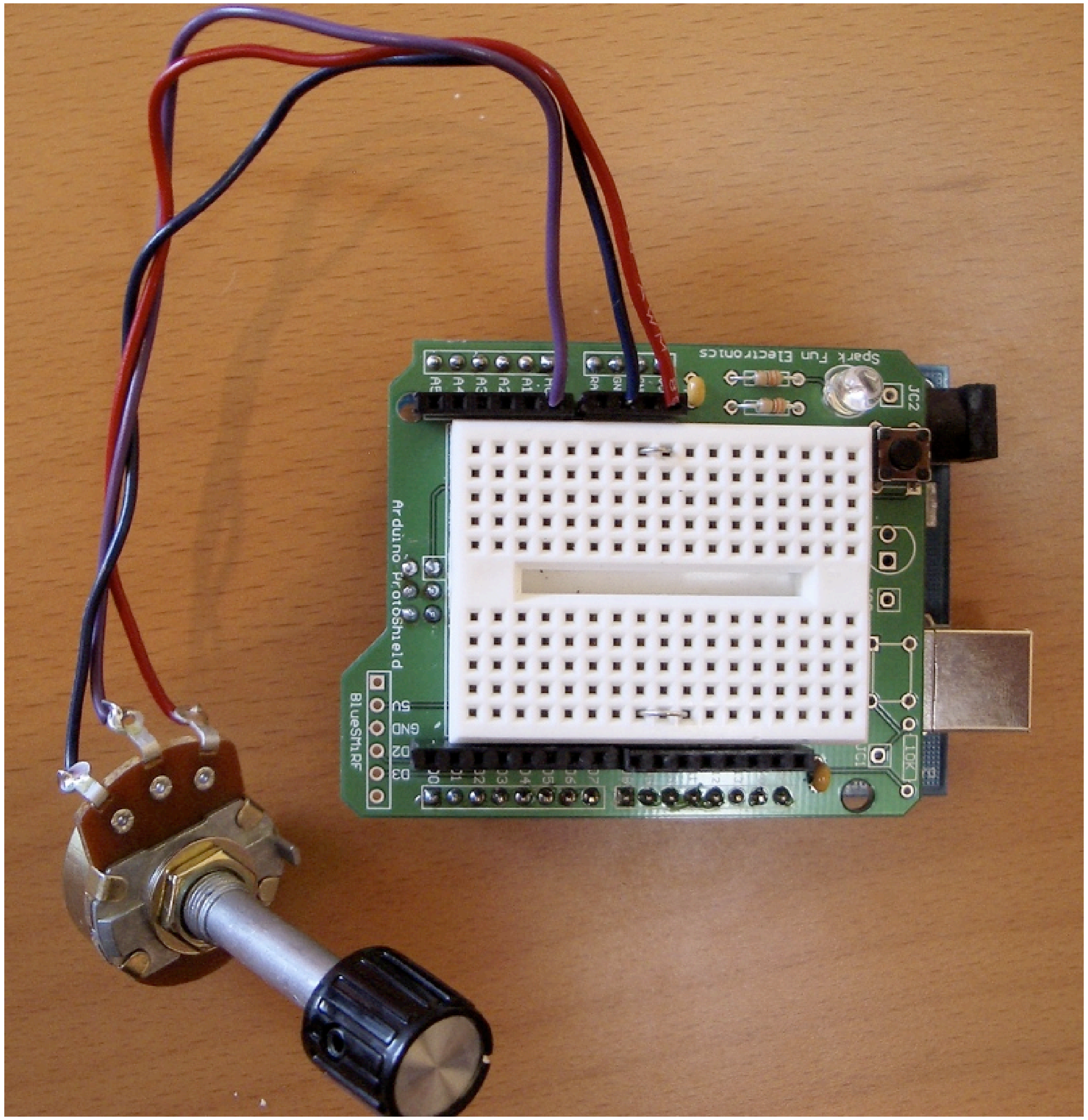
And that's actually how it works, btw, if you take apart a pot.
But I might have the directions reversed (clockwise vs. anti–clockwise).

Arduino Analog Input

Red to Vcc

Purple to A0

Blue to Gnd

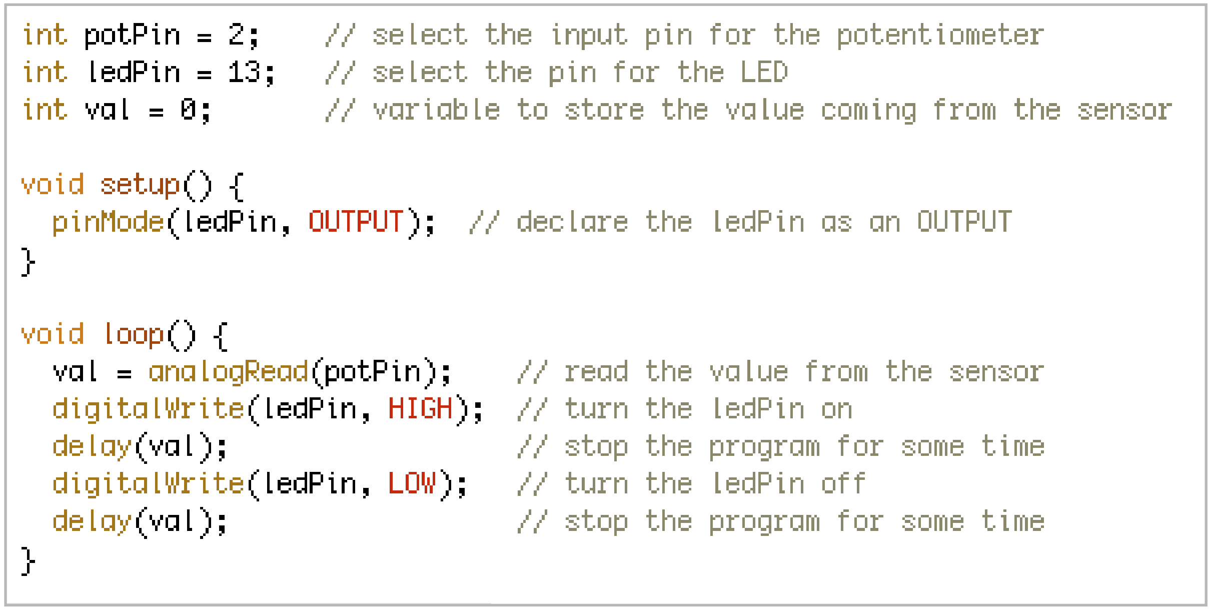


Hook it up, plug in the wires in directly
“Vcc” is alias for +5V.
“Raw” is alias for external power (approx 9V)

Analog Input Sketch

Sketch “Examples/sensors_resistive/analog_read_led”

Change to 0



```
int potPin = 2;    // select the input pin for the potentiometer
int ledPin = 13;   // select the pin for the LED
int val = 0;       // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT);  // declare the ledPin as an OUTPUT
}

void loop() {
  val = analogRead(potPin);    // read the value from the sensor
  digitalWrite(ledPin, HIGH);  // turn the ledPin on
  delay(val);                  // stop the program for some time
  digitalWrite(ledPin, LOW);   // turn the ledPin off
  delay(val);                  // stop the program for some time
}
```

Turn knob to vary blink rate of the LED

Notice no `pinMode()` for analog inputs

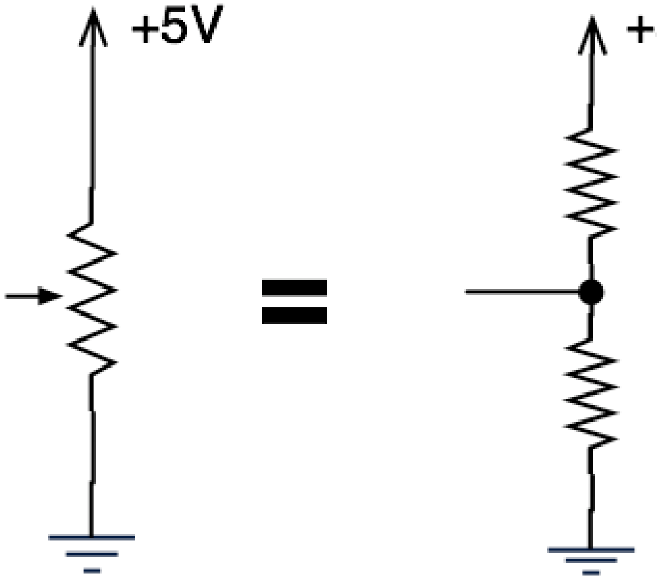
What good are pots?

- Anytime you need a ranged input
  - (we’re used to knobs)
- Measure rotational position
  - steering wheel, etc.

- But more importantly for us, potentiometers are a good example of a *resistive sensor*

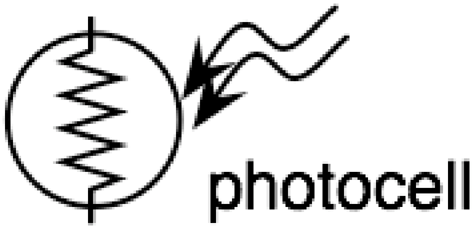
Sensing the Dark

- Pots are example of a *voltage divider*
- Voltage divider splits a voltage in two
- Same as two resistors, but you can vary them

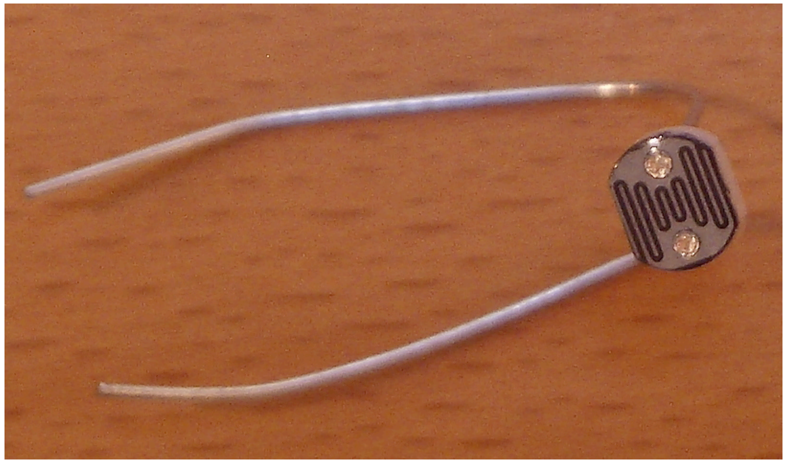


Sensing the Dark: Photocells

- aka. photoresistor, light-dependent resistor
- A *variable* resistor
- Brighter light == lower resistance
- Photocells you have range approx. 0-10k

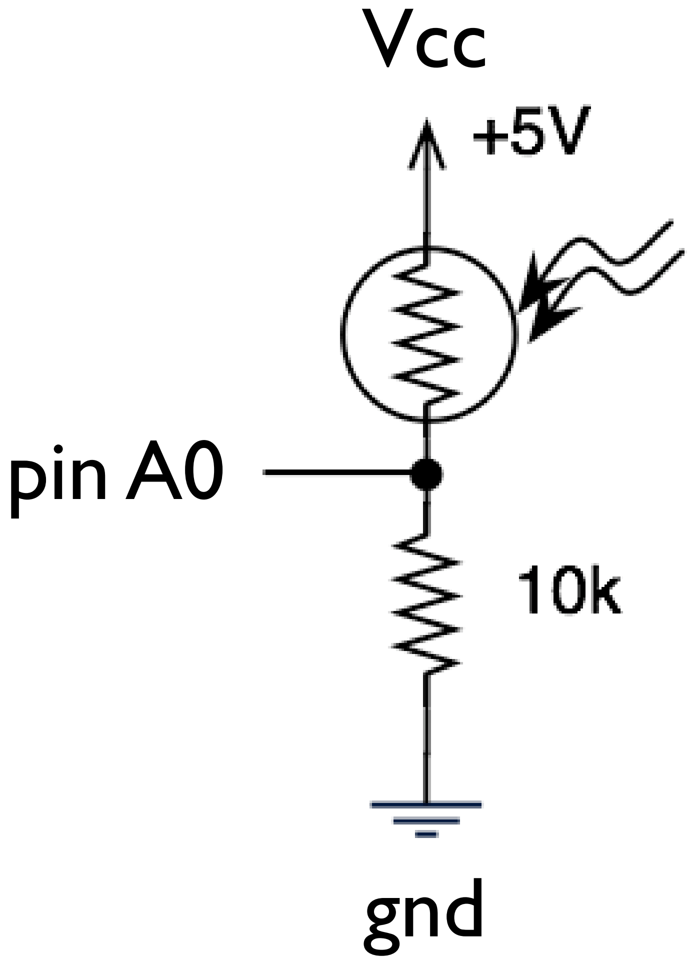


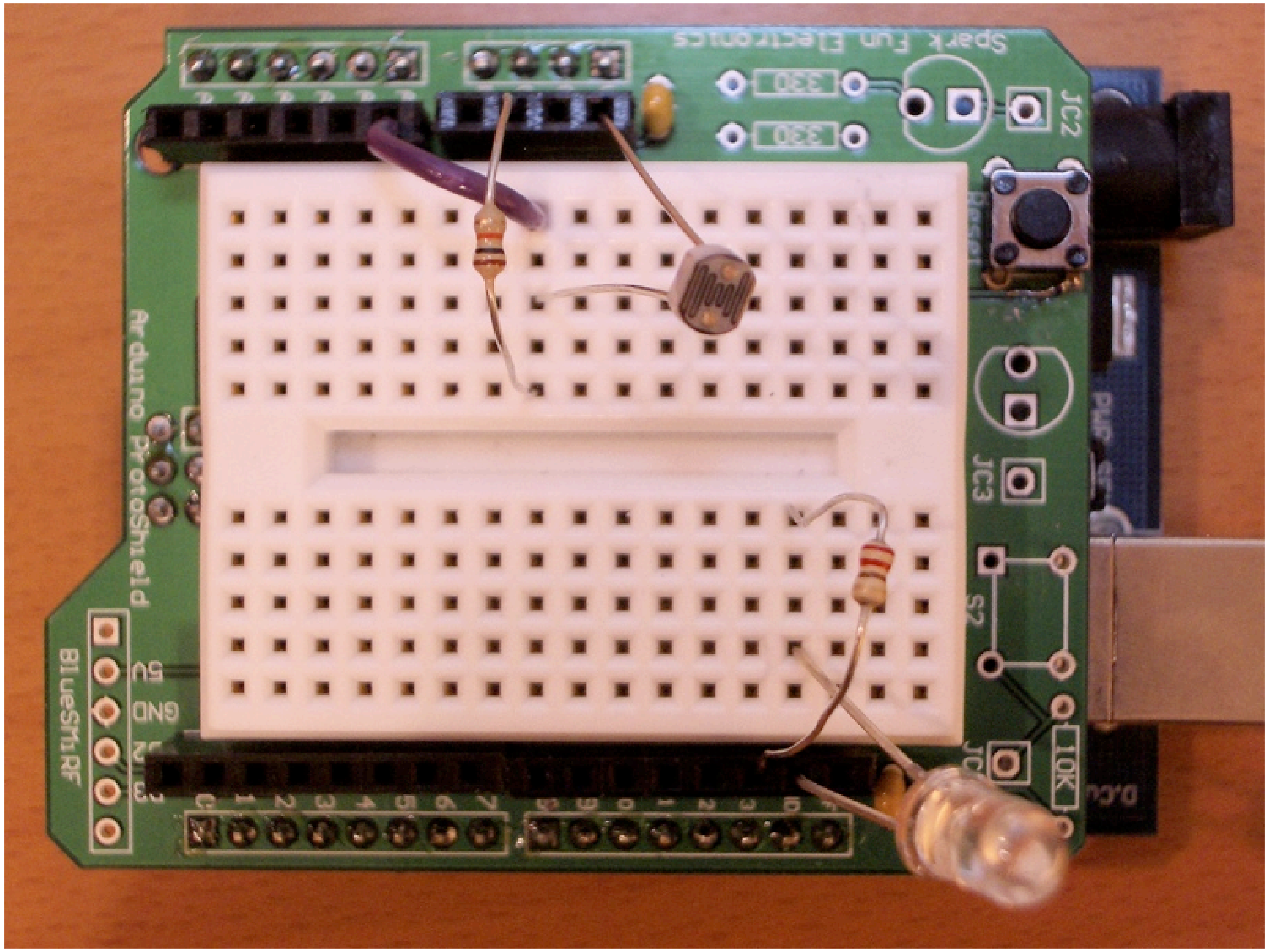
schematic symbol



Pretty cheap too. Can get a grab bag of 100 misc from Jameco for $20

Photocell Circuit



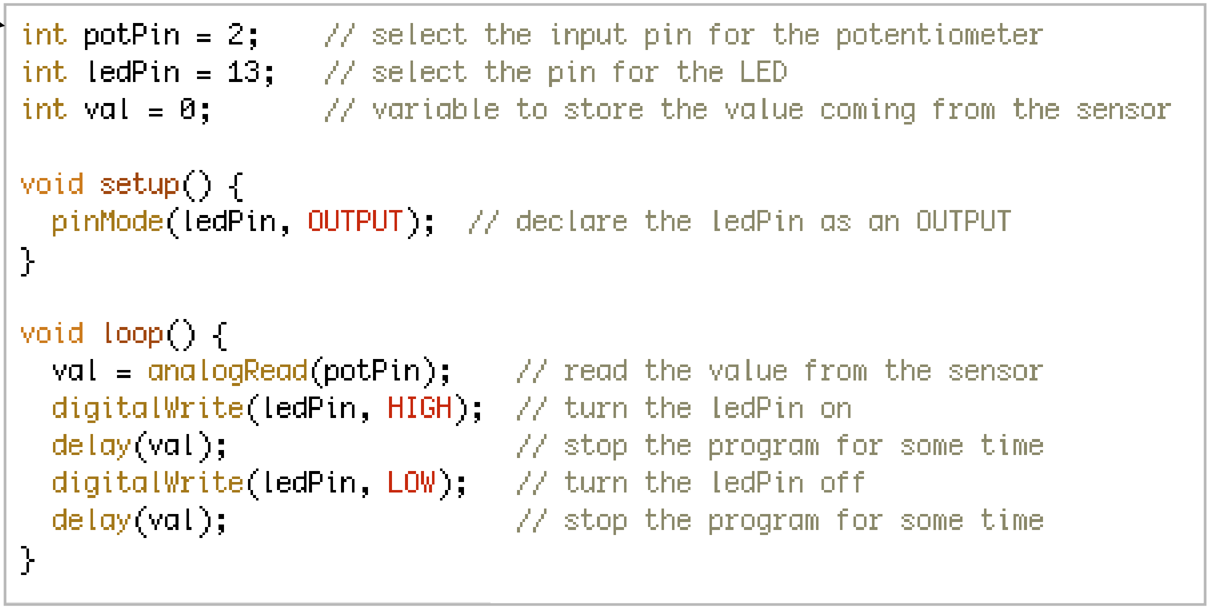


Looks a lot like the pot circuit, doesn't it?

Photocell Arduino Sketch

Can use as before, sketch “analog_read_led”

Change to 0



```
int potPin = 2;    // select the input pin for the potentiometer
int ledPin = 13;   // select the pin for the LED
int val = 0;       // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT);  // declare the ledPin as an OUTPUT
}

void loop() {
  val = analogRead(potPin);    // read the value from the sensor
  digitalWrite(ledPin, HIGH);  // turn the ledPin on
  delay(val);                  // stop the program for some time
  digitalWrite(ledPin, LOW);   // turn the ledPin off
  delay(val);                  // stop the program for some time
}
```

Wave your hand over it = blink faster
Point it towards the light = blink slower

Just like magic!
If circuit was configured the other way (photocell on bottom), then darkness would make it blink slower.

More Spooky, Please

All this blinking is okay, but...

Booo!

I design
your eyes

Okay, so the googly-eyeness of it makes it more Simpsonesque than spooky.

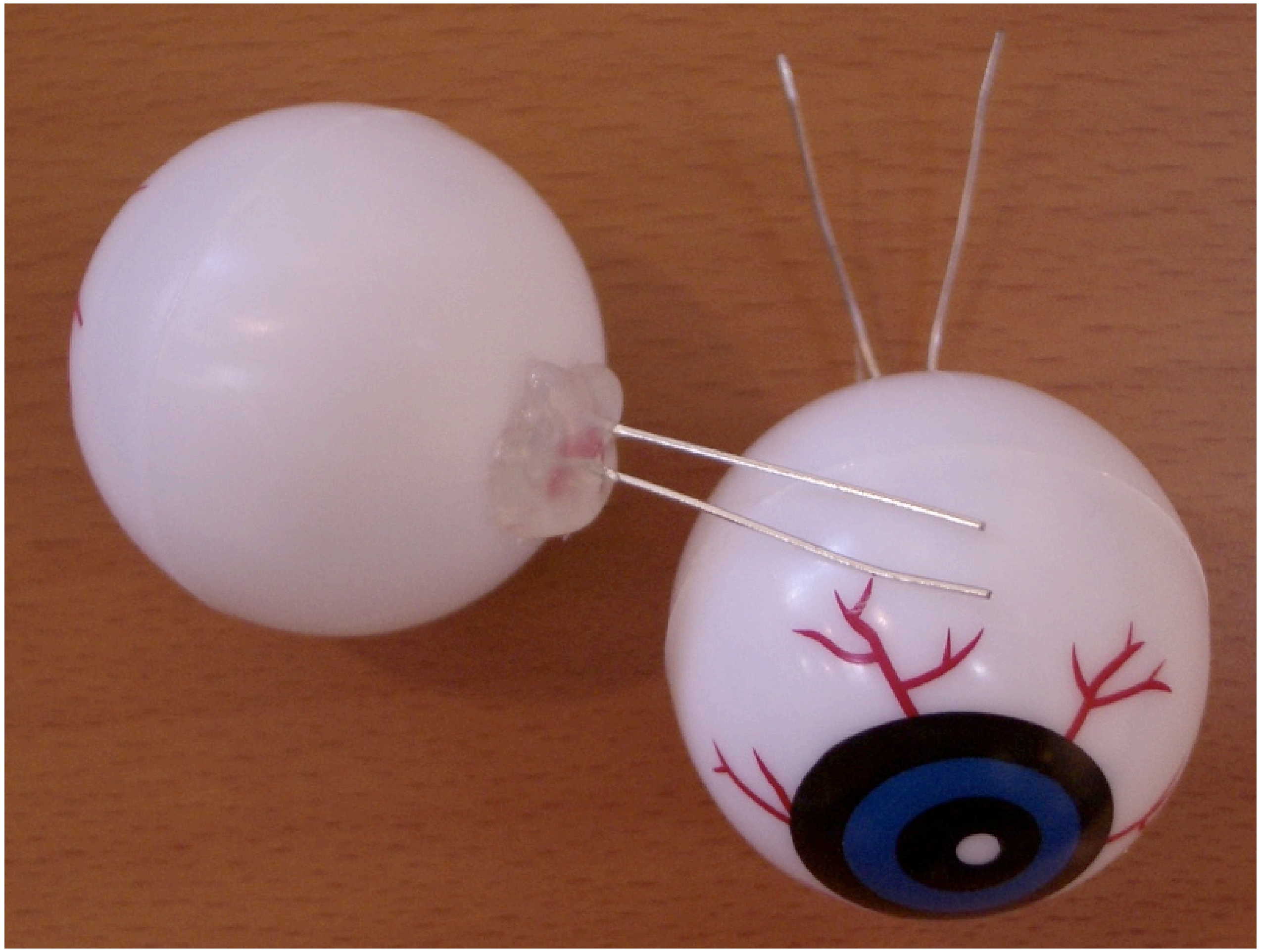
Evil Glowing Eyes

Almost as cool as Roy Batty

LED Eyeballs

Use your two orange LEDs

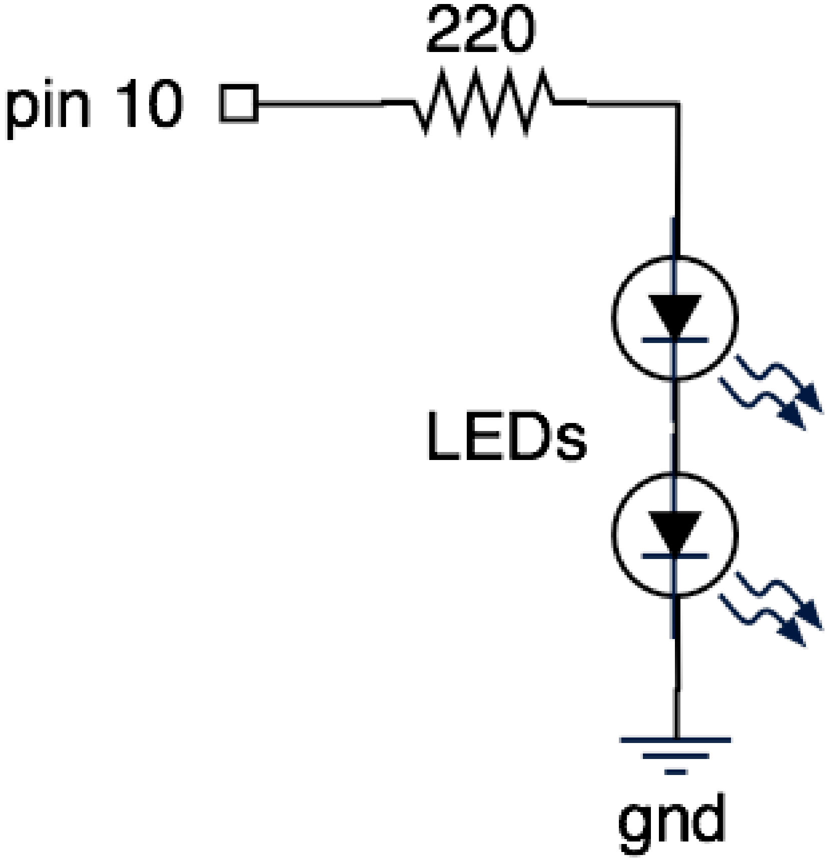
Little bit of hot glue and you’re set



Use the two orange LEDs.
Save the R,G,B LEDs for next week.
Hot glue is the best thing in the world.
I brought my hot glue gun if you want to do this right now

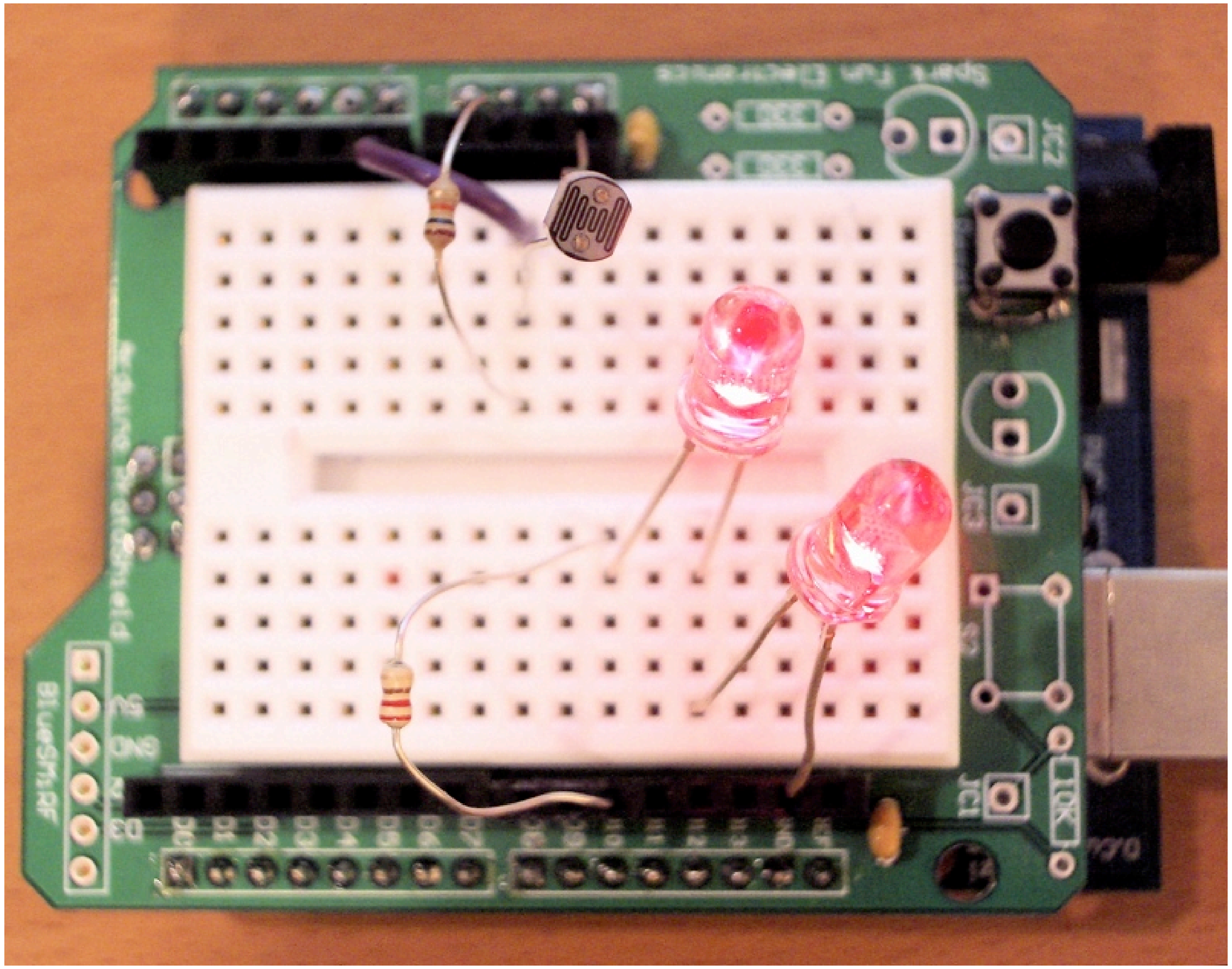
Driving Two LEDs

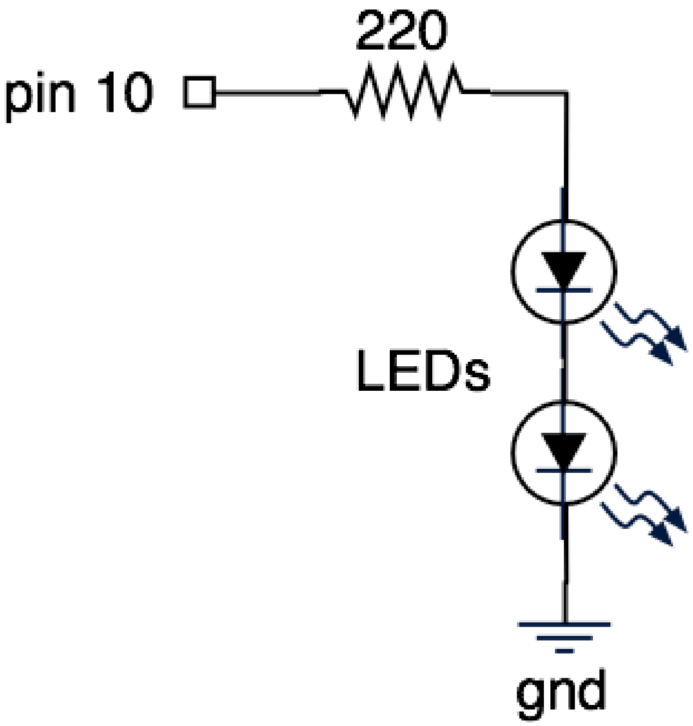
- Could use two Arduino pins. But wasteful.
- Instead, put two in series
- Doesn't work for blue LEDs
  (and white, and some green)



Blue LEDs have a voltage drop of ~3.4V, two in series makes ~6.8V which is greater than the 5V the Arduino puts out.
Don't put LEDs in parallel. http://members.misty.com/don/ledd.html
Notice pin 10. That's important.

LED Eyes





220

pin 10

LEDs

gnd

photocell circuit is as before

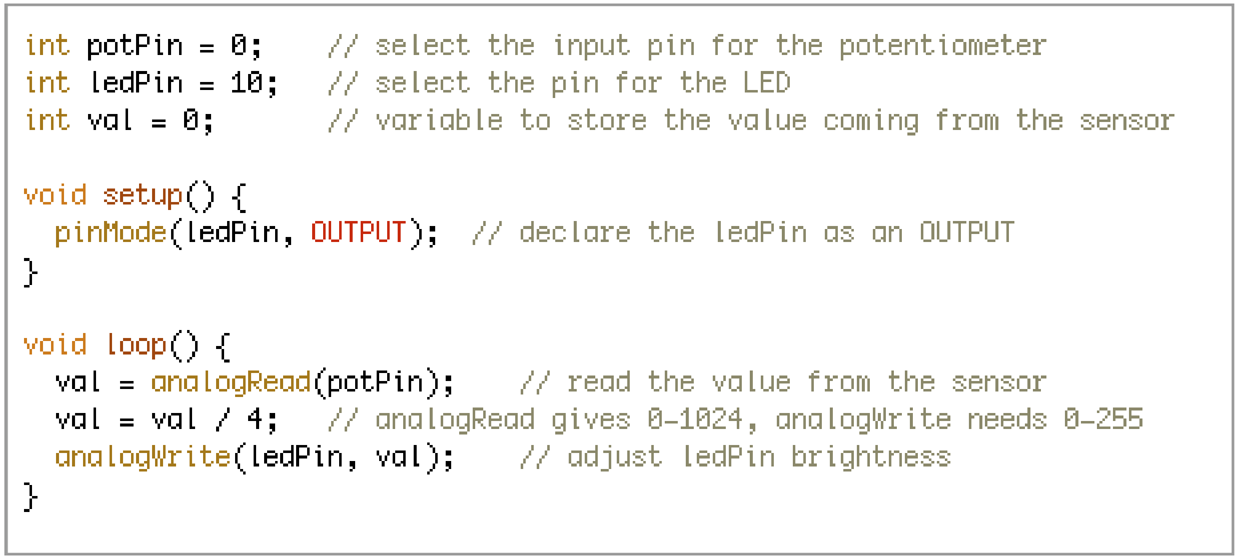
Notice, pin 10. This will become important later.

LED Eyes Brightness

- To complement `analogRead()`, there is `analogWrite()`.
- Only available on digital pins 9,10,11. (yes, a little confusing)
- More next week about how it works.
- Can use it to set <u>brightness</u> of LEDs

LED Eyes Sketch

Sketch “analog_brightness”



```
int potPin = 0;    // select the input pin for the potentiometer
int ledPin = 10;   // select the pin for the LED
int val = 0;       // variable to store the value coming from the sensor

void setup() {
  pinMode(ledPin, OUTPUT);  // declare the ledPin as an OUTPUT
}

void loop() {
  val = analogRead(potPin);    // read the value from the sensor
  val = val / 4;   // analogRead gives 0-1024, analogWrite needs 0-255
  analogWrite(ledPin, val);    // adjust ledPin brightness
}
```

As it gets darker, the LEDs get less bright

You just built an auto-dimmer

This is cool, but still not spooky enough.

Making Eyes Glow

(where “glow” is the throbbing of brightness)

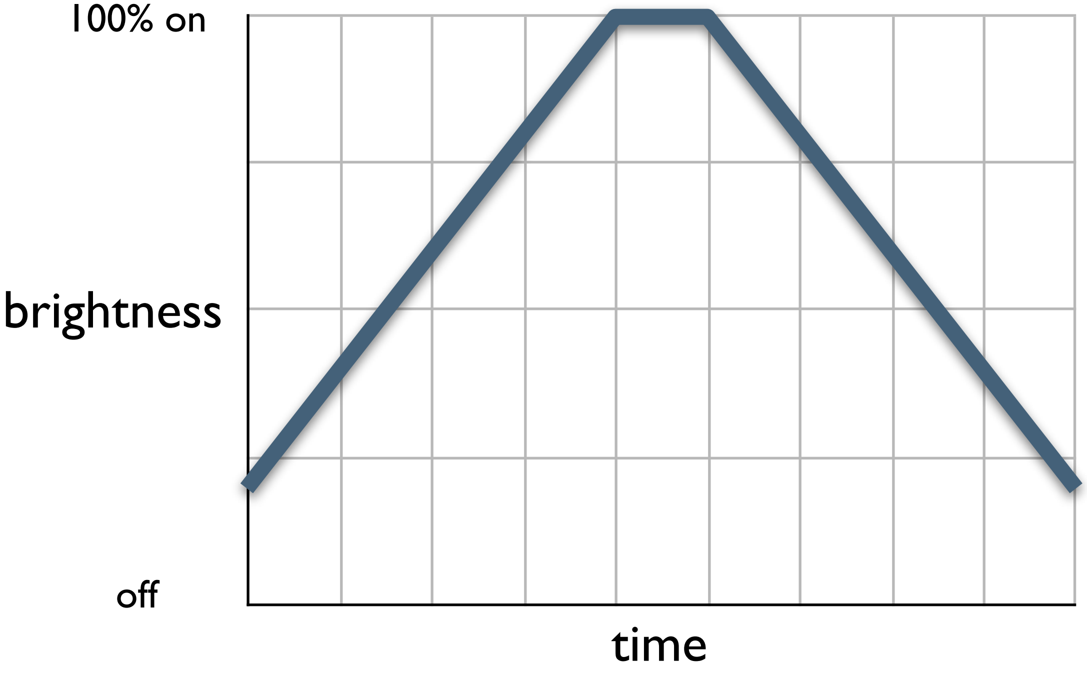
How does that glow throbbing work?

Sleeping laptops do something similar

Need to describe how brightness changes over time

LED Brightness Functions

Brightness over time can be described as a graph

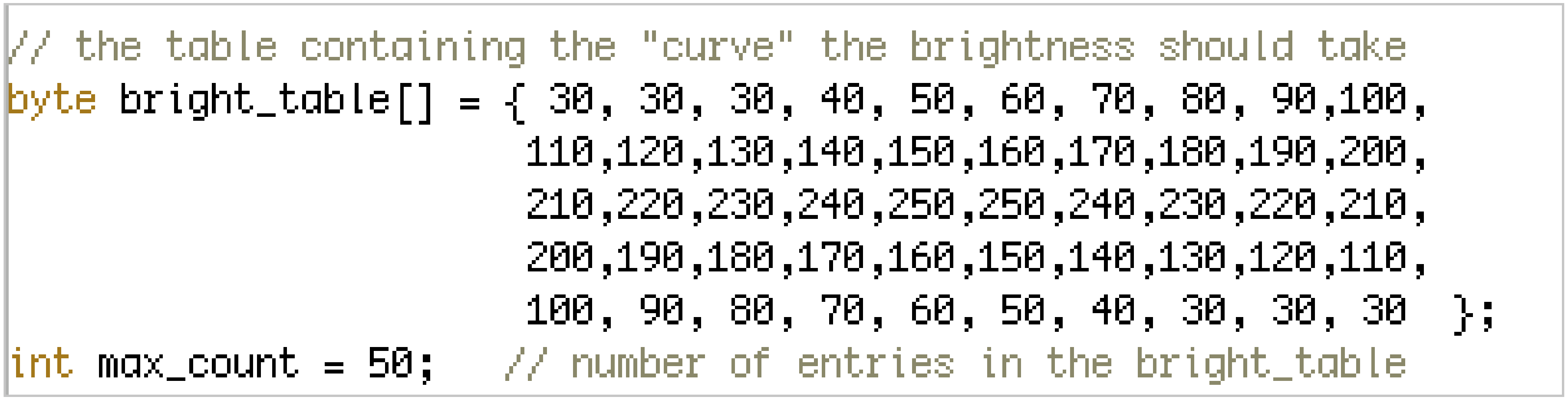


Draw your graph, use the resulting numbers

Doesn’t matter which numbers you choose right now

LED Brightness Functions

Then turn those numbers into an array



```
// the table containing the "curve" the brightness should take
byte bright_table[] = { 30, 30, 30, 40, 50, 60, 70, 80, 90,100,
                       110,120,130,140,150,160,170,180,190,200,
                       210,220,230,240,250,250,240,230,220,210,
                       200,190,180,170,160,150,140,130,120,110,
                       100, 90, 80, 70, 60, 50, 40, 30, 30, 30  };
int max_count = 50;   // number of entries in the bright_table
```

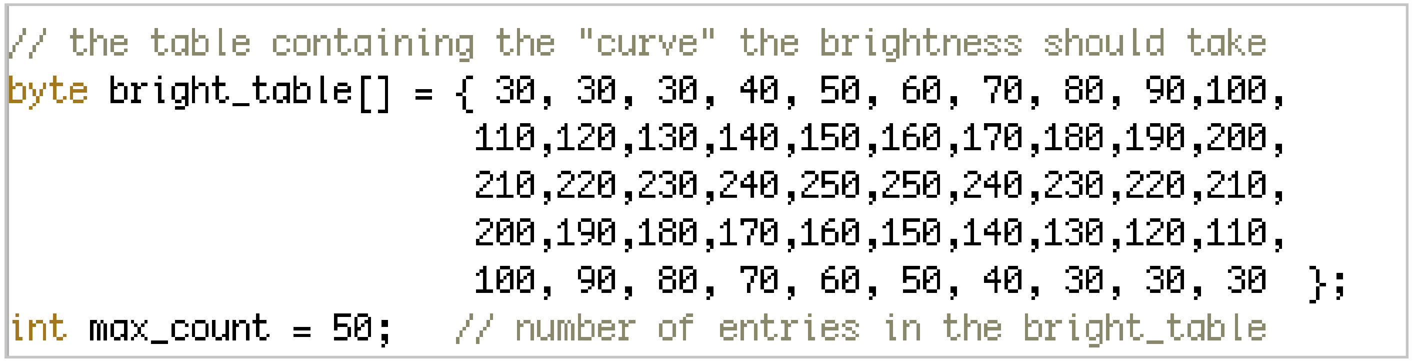
Use any pattern of numbers you like
but they must range between 0-255

```
  0 = full off
127 = half on
255 = full on
```

Make sure max_count is not too large!

LED Brightness Functions

Once you have your table...

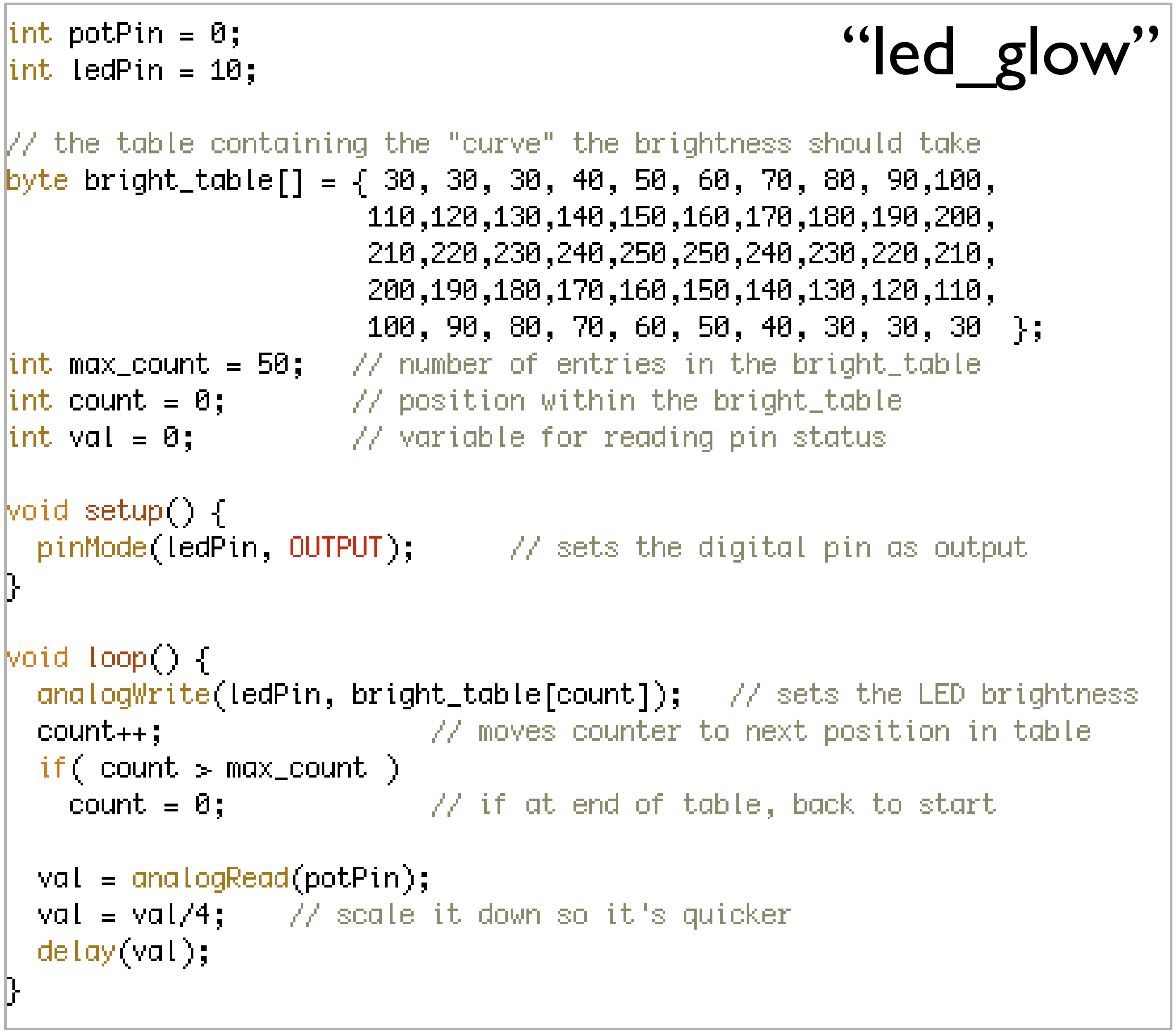


```
// the table containing the "curve" the brightness should take
byte bright_table[] = { 30, 30, 30, 40, 50, 60, 70, 80, 90,100,
                       110,120,130,140,150,160,170,180,190,200,
                       210,220,230,240,250,250,240,230,220,210,
                       200,190,180,170,160,150,140,130,120,110,
                       100, 90, 80, 70, 60, 50, 40, 30, 30, 30  };
int max_count = 50;   // number of entries in the bright_table
```

...the rest is just programming

1. Get a `bright_table` value
2. Send it out with `analogWrite()`
3. Advance counter into `bright_table`
4. Wait a bit
5. Repeat

Glowing Eyes Sketch

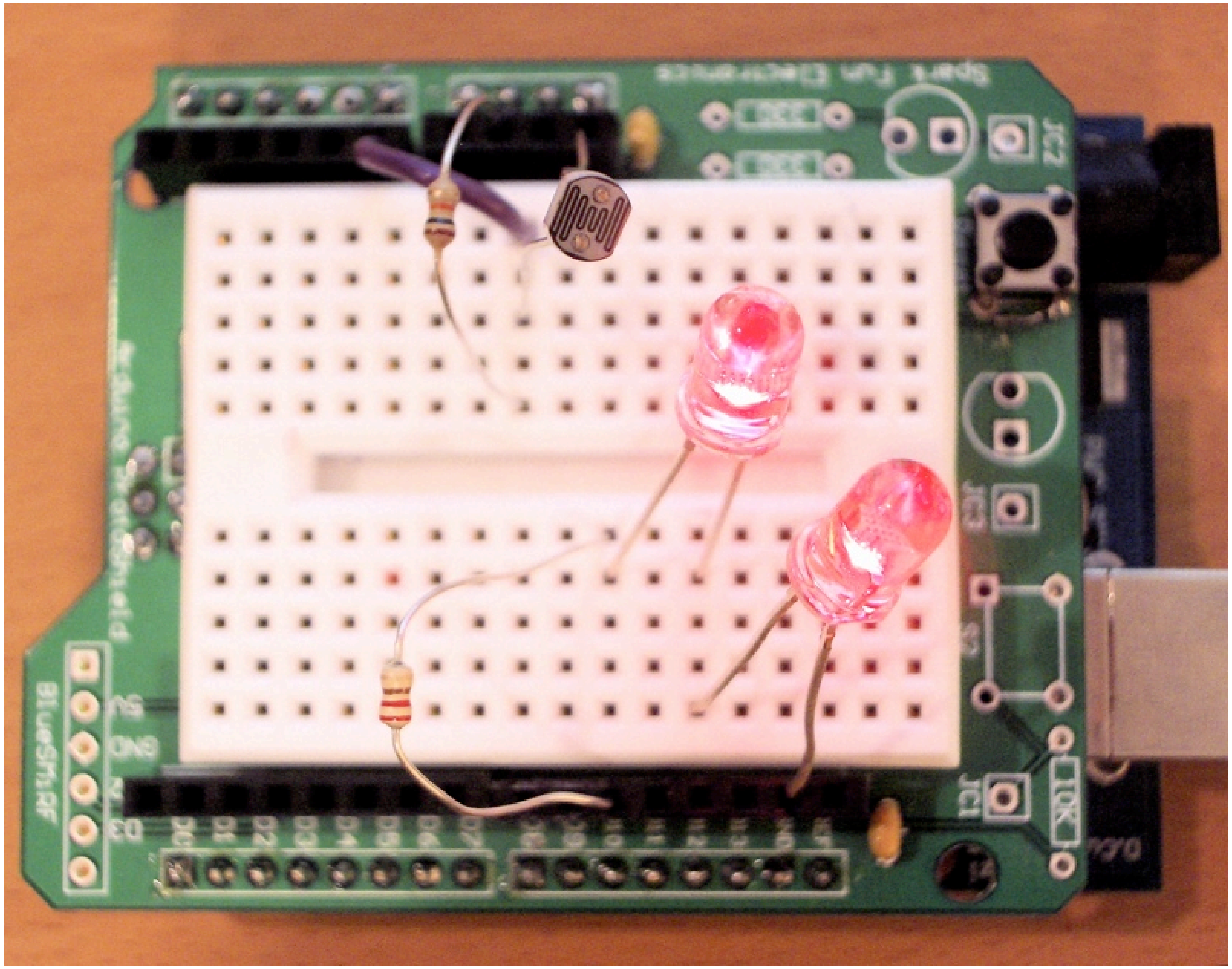


“led_glow”

```
int potPin = 0;
int ledPin = 10;

// the table containing the "curve" the brightness should take
byte bright_table[] = { 30, 30, 30, 40, 50, 60, 70, 80, 90,100,
                       110,120,130,140,150,160,170,180,190,200,
                       210,220,230,240,250,250,240,230,220,210,
                       200,190,180,170,160,150,140,130,120,110,
                       100, 90, 80, 70, 60, 50, 40, 30, 30, 30  };
int max_count = 50;   // number of entries in the bright_table
int count = 0;        // position within the bright_table
int val = 0;          // variable for reading pin status

void setup() {
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop() {
  analogWrite(ledPin, bright_table[count]);   // sets the LED brightness
  count++;                     // moves counter to next position in table
  if( count > max_count )
    count = 0;                 // if at end of table, back to start

  val = analogRead(potPin);
  val = val/4;     // scale it down so it's quicker
  delay(val);
}
```

I can

Glowing Eyes



Going Further

- Glowing LEDs
  - The last sketch is *data driven*
  - So you can plug in any brightness function
  - Make a flickering candle or a bad neon light

Going Further

- Photocells
  - Think of some interesting uses
  - What about *multiple* photocells?
- Homemade Sensors
  - Make some of your own!