Intro to Electronics

Week 5

Build a Larson scanner (red moving Cylon eye)

TODAY'S PROJECT

Analog vs. digital

- Continuous range of voltages
  - Can use any value within certain limits
  - More susceptible to noise
- Useful for sound, light, sensing, etc.

- Two voltages — high and low
  - Can only use these two values (0 and 1)
  - Can lose some information
- Useful for data storage, processing, etc.

Analog vs. digital

- Continuous range of voltages
  - Can use any value within certain limits
  - More susceptible to noise
- Useful for sound, light, sensing, etc.

- Two voltages — high and low
  - Can only use these two values (0 and 1)
  - Can lose some information
- Useful for data storage, processing, etc.

- Plenty of reasons to use both!

Digital logic

- We'll focus here on digital
- Started with this a couple of weeks ago
  - Mainly talked about logic gates

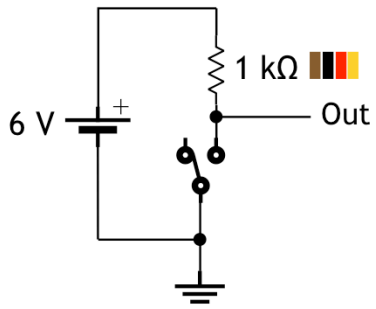
Digital logic

- More complex parts exist than just gates:
  - Multiplexers
    - Use one signal to control several outputs
  - Latches
    - Fundamental storage element — store a bit at a time
  - Adders, multipliers
    - Mathematical elements (add/multiply numbers)
  - etc.

Making zeroes and ones

- Need to hold stable high and low voltages
- How do we do that?
  - Already know how to set a high *or* low voltage
    - Use a switch to connect something to either positive or negative

Pull-up resistor



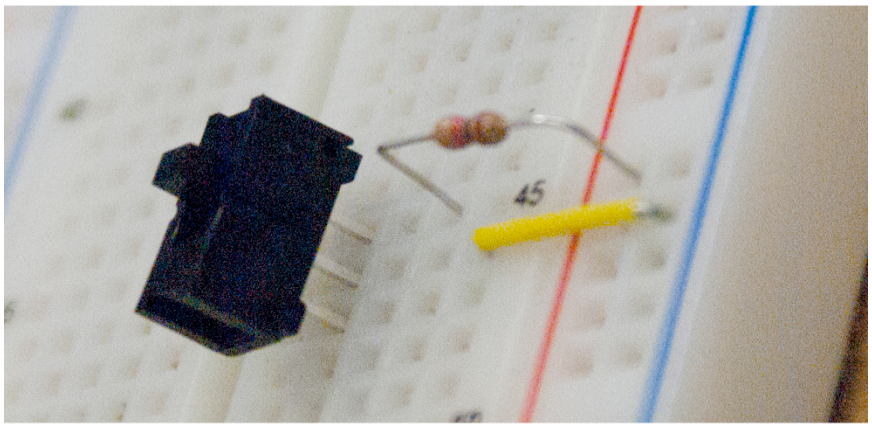
- When switch is connected, output is 0 V (low)
- When switch is disconnected, output is ~6 V (high)
- Avoids "floating" (unpredictable) output

When the switch is connected, the output is at 0 V because all of the batteries' 6 V ends up going across the resistor. (This means it does draw current from the battery — 6 mA, specifically. (Remember how to do that?) Much like the filter capacitor last week that just went between positive and negative, this is the sort of thing where you just pick a reasonable size because (within reason) it doesn't really matter what specific value you use.)

When the switch is disconnected, the output is still connected to *something* — namely, the positive end of the battery through the resistor. This means that while the output could be a little less than 6 V (depending on what it's connected to), it's still considerably higher than 0 V, and all things kept constant, it'll stay there.

Why do we care? If we didn't have that resistor connecting the positive end to the output, then when the switch isn't connected, the output isn't hooked to *anything*, which means its voltage could be just about anything. Digital circuits like things to be predictable: Something's either high or low, and there really isn't much use for in between.

Pull-up resistor



(Yes, I know that because we have an SPDT switch, we could connect the middle terminal to the output and the other two terminals to positive and negative and theoretically be able to get rid of the resistor. If you just had a plain (SPST) switch that has only two terminals, though, and can just connect or disconnect them, you'd definitely need something like this.)

Pull-up resistor

- Measure it with a multimeter!
  - Watch the voltage between the negative end of the batteries and the leg of the resistor that connects to the switch
  - Note the difference at different positions of the switch

Decimal counter

- New component!
- Ten (main) outputs
  - We'll refer to them as outputs 0 through 9
- One (main) input
  - Called the "clock"

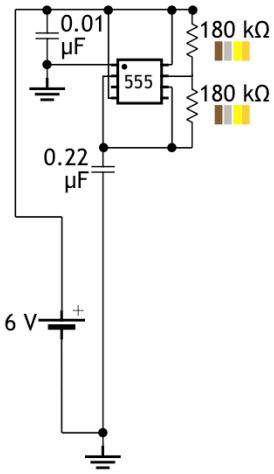
Decimal counter

- Clock oscillates (goes back and forth)
- Each time it goes high:
  - The currently active output turns off
  - The next output turns on
- Counts from 0 to 9 and then loops back to 0
  - Like a wheel in an old odometer

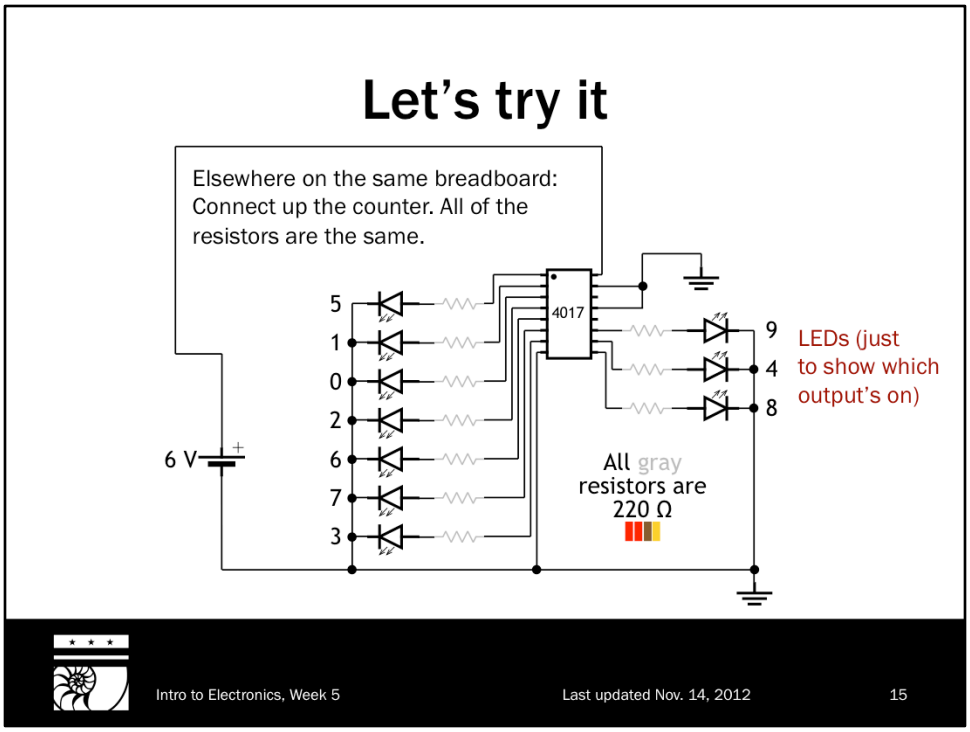
What works as a clock?

- Anything that alternates between high and low
  - Quartz crystals
  - Resonant LC networks
  - Part of your credit cards' magnetic stripes
  - 555 timer outputs!

Let's try it



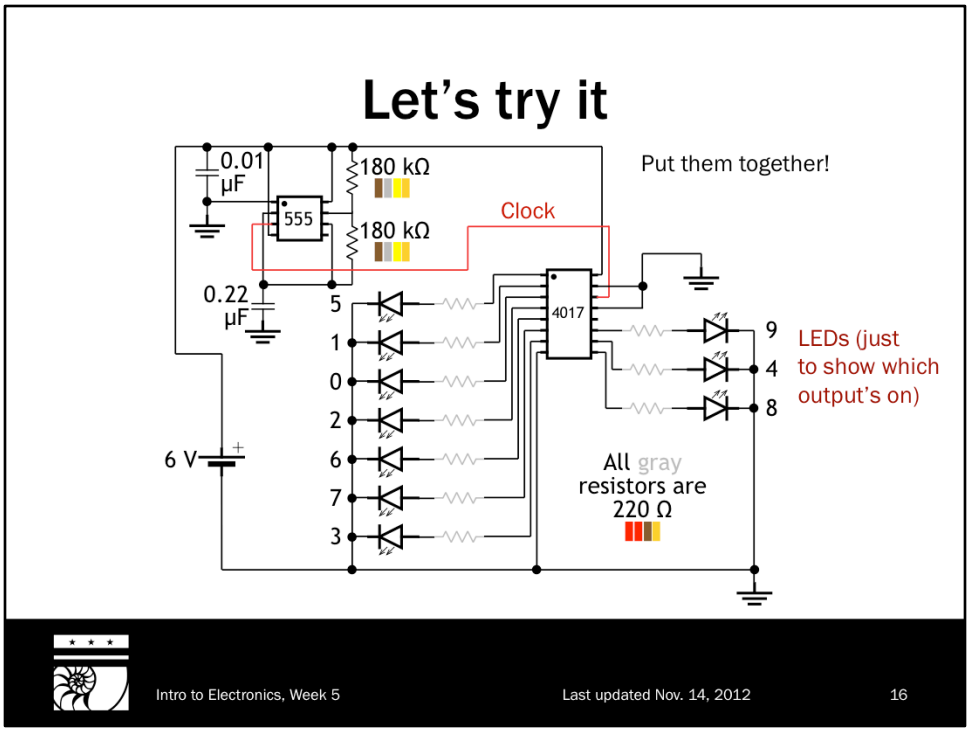
Just the clock first:
This is same 555 circuit as last week, just with different resistors (and therefore a different frequency).



Useful thing to notice: The counter has four connections to power. The top right pin (pin 16) goes to the positive supply, the bottom left pin (pin 8) goes to the negative (ground), and two of the other pins on the chip (pins 13 and 15, which control other details about how the counter runs) also go to ground. (Remember, pins on chips are numbered starting with pin 1 at the dot and going up as you go counterclockwise.)

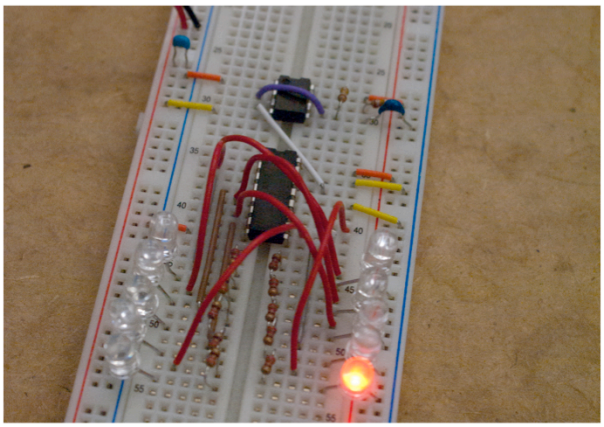
As for all of the other pins we use, each pin connects to one end of a 220 Ω resistor; the other end of that resistor goes to the positive end of an LED (long leg, or non-flat side), and the other end of the LED goes to ground.

For more information on this counter, check out the datasheet: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00000312.pdf



The only thing you need to put these two together is one more wire going from the timer's output pin (pin 3) to the counter's clock pin (pin 14). It's the connection highlighted in red.

Let's try it



For all of you who commented last week on my breadboard technique, I didn't have a good way to make this look good. Sorry about that.

Part of that, though, is that even though the counter has its outputs in some weird order, I wanted to put the LEDs in order anyway — so my wires especially go all over the place to make that happen. It's cool if yours do, too.

What now?

- We have a sequence of flashing lights
- Can we make them go in a line back and forth?

Make a table!

- Let's use six LEDs instead of 10
- At each step, let's pick which LED turns on

Step (output)	Which LED to light?
0	0
1	1
2	2
3	3
4	4
5	5
6	4
7	3
8	2
9	1

The 0-5 part looks pretty straightforward — after all, what we have *already* does this. It's the last four steps (6-9) that make this a bit more complicated.

You can think of this like an animation with 10 frames, if that helps. We just have to figure out how to "draw" the frames.

Make another table!

Step (output)	LED
0	0
1	1
2	2
3	3
4	4
5	5
6	4
7	3
8	2
9	1

LED	Steps when it should be on
0	0
1	1 or 9
2	2 or 8
3	3 or 7
4	4 or 6
5	5

Each LED has one or two times in the sequence when it needs to be turned on.

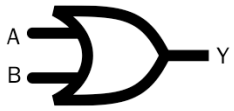
This or that

- Need some way to say “when A or B is on, make Y turn on as well”

Every time I look at this slide, it looks like I wrote the word “bison”.

This or that

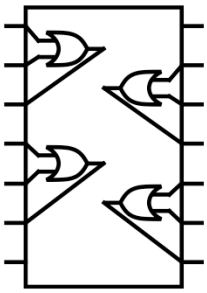
- Already got one: the OR gate



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

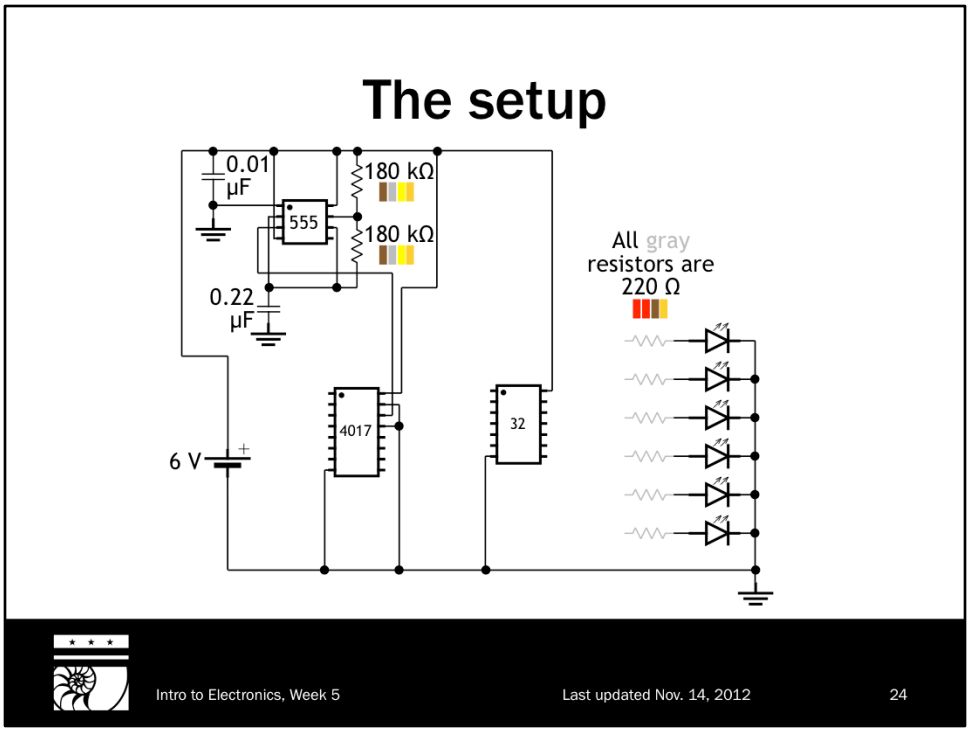
OR gate chip (74HC32)

- Has four OR gates built in
  - How handy — we need four of them



The two remaining pins are for positive (top right, pin 14) and ground (bottom left, pin 7). This is a really common thing for logic chips. (Remember that the counter had the same setup.)

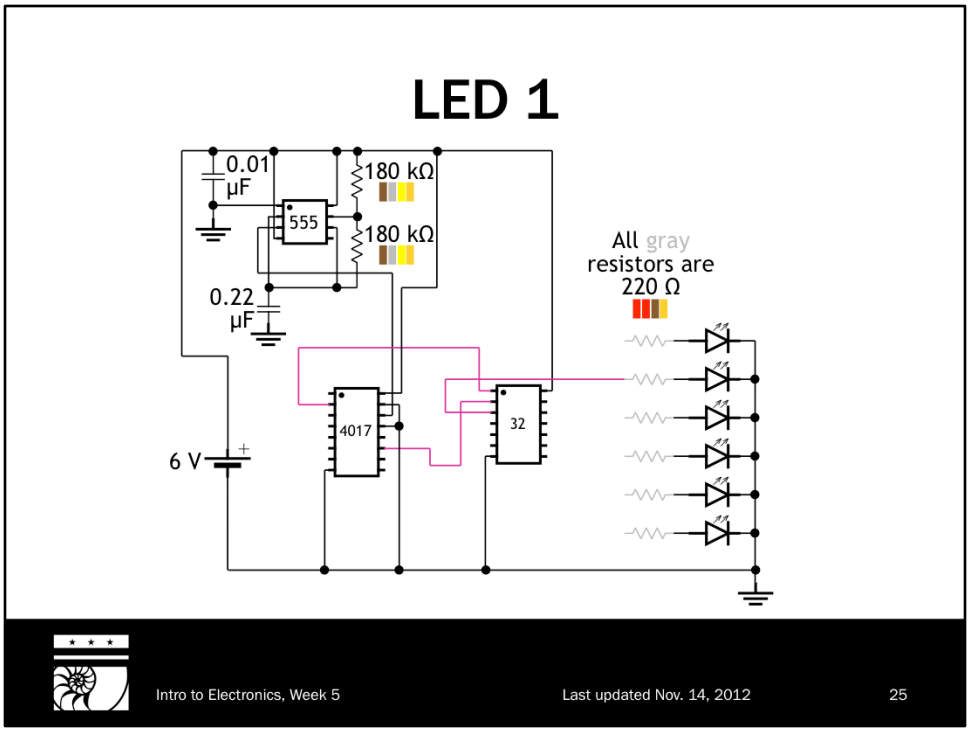
This chip's datasheet is here: http://www.ti.com/lit/ds/symlink/sn74hc32.pdf



Let's start by setting all of the chips and LEDs up where we want them and then focus on the connections between them. (I've broken up all of those connections into the next several slides so it's not so overwhelming.)

Disconnect everything from the counter except power, ground and the 555 circuit.
Add the OR gate chip somewhere and connect it to power and ground.
Add the six LEDs and resistors somewhere and get those connected together.

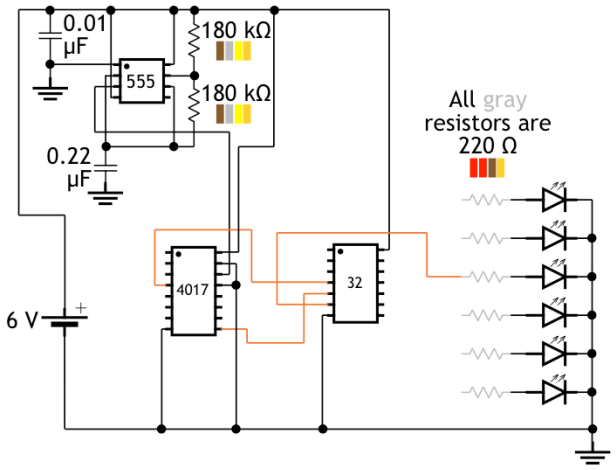
All that's left are the connections between the counter, the OR gates and the LEDs! Onward!



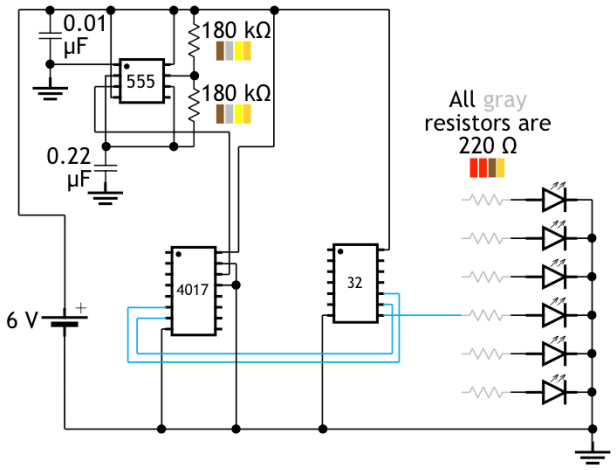
I'll highlight one OR gate's connections at a time. For each of these, there'll be two wires going from the counter to the OR gate and one wire going from the OR gate to an LED's resistor.

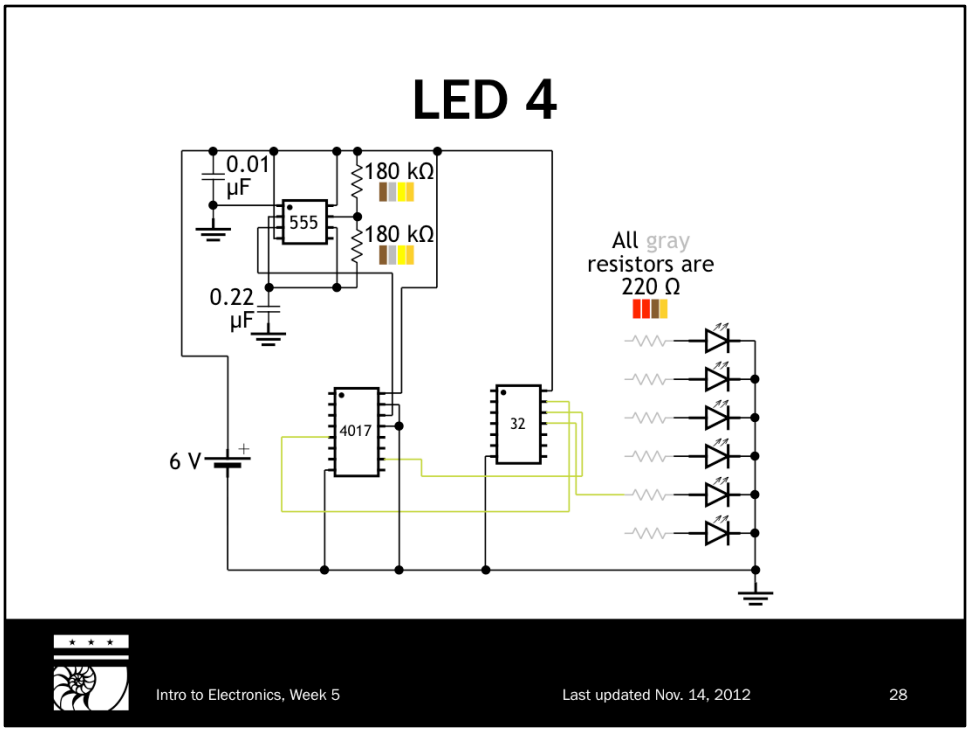
Remember, all this does is set up the circuit so that whenever *either* of those outputs on the counter is turned on, the LED gets turned on, too.

LED 2

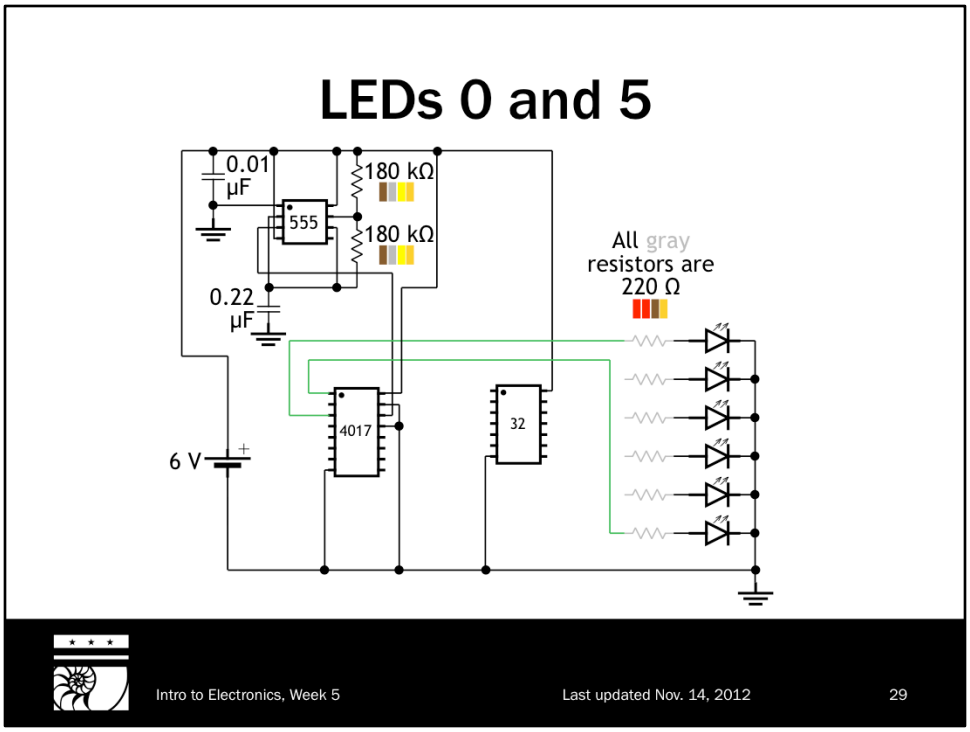

0.01 µF
555
180 kΩ
180 kΩ
0.22 µF
6 V
4017
32
All gray resistors are 220 Ω

LED 3

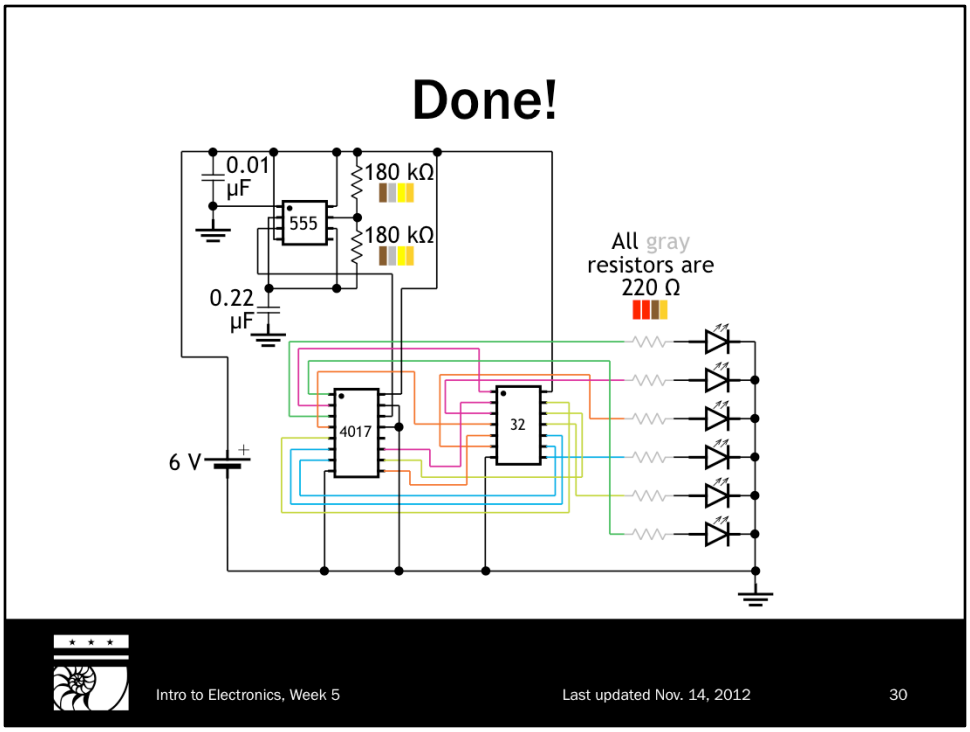

0.01 µF
555
180 kΩ
180 kΩ
0.22 µF
6 V
4017
32
All gray resistors are 220 Ω


LED 4
0.01 µF
555
180 kΩ
180 kΩ
0.22 µF
All gray resistors are 220 Ω
6 V
4017
32
Intro to Electronics, Week 5
Last updated Nov. 14, 2012
28

Last OR gate!

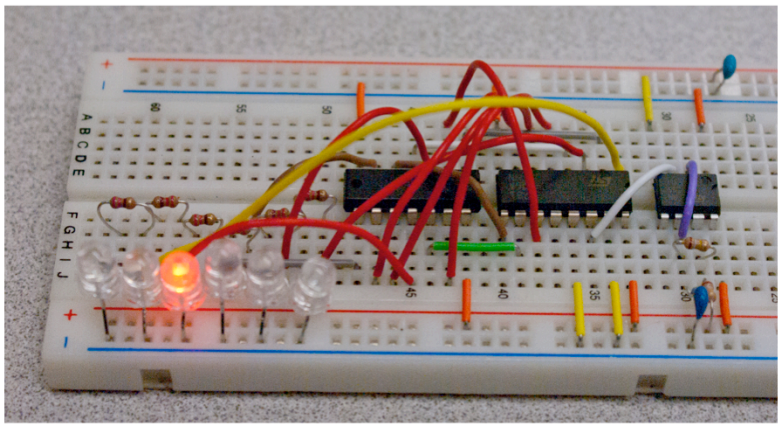


There are two more LEDs to worry about — the ones at either end of the string. These only turn on at one point each in the sequence, so we can just connect them directly to the counter rather than doing anything with OR gates. (That's good, because we're out of OR gates.)



That's everything!

Done!



That's everything!

- Thanks again for coming to the class!