Lecture 05
Prototyping & Digital Circuits

CE346 – Microprocessor System Design

Branden Ghena – Fall 2022

Some slides borrowed from:
Josiah Hester (Northwestern), Prabal Dutta (UC Berkeley)

Northwestern

Administrivia

- Labs
  - Debrief: How did that go?
    - Can use personal computers if preferred

  - See schedule of Lab hours available on Canvas for checkoffs
    - Due by end-of-day Thursday
    - Also due are postlab questions (on Canvas)

- Quiz
  - Today at end of class!
  - Someone remind me at ~4:30 if I don’t stop

Project Proposals

- It is time to start forming teams and working on Proposals
  - Due next week Thursday! (10/13)

  - Project details are posted to Campuswire
  - Specific proposal details are on the Canvas assignment
    - 1-2 pages, with some specific items you MUST include

- Project teams are 2-3 students (4 under rare occasions)
  - You may NOT work alone
  - There is a partnership survey if you want us to match you with someone
    - Due by end-of-day Sunday

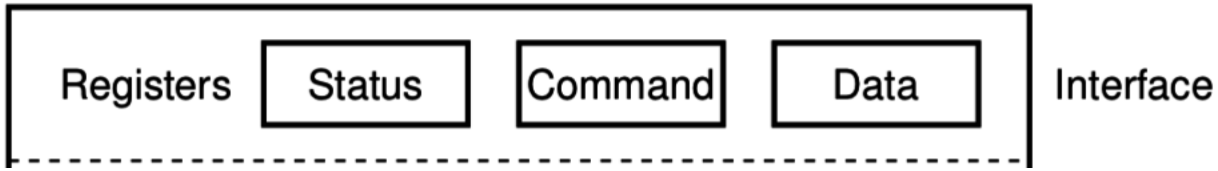
Today's Goals

- Explore another peripheral interaction pattern: DMA
- Discuss prototyping methods and basic circuits components
- Understand the basics of digital circuitry
  - Enough to be able to interact with the Microbit

Outline

- **DMA**
- Prototyping
- Digital Circuits
- Components

Reminder: Polling I/O

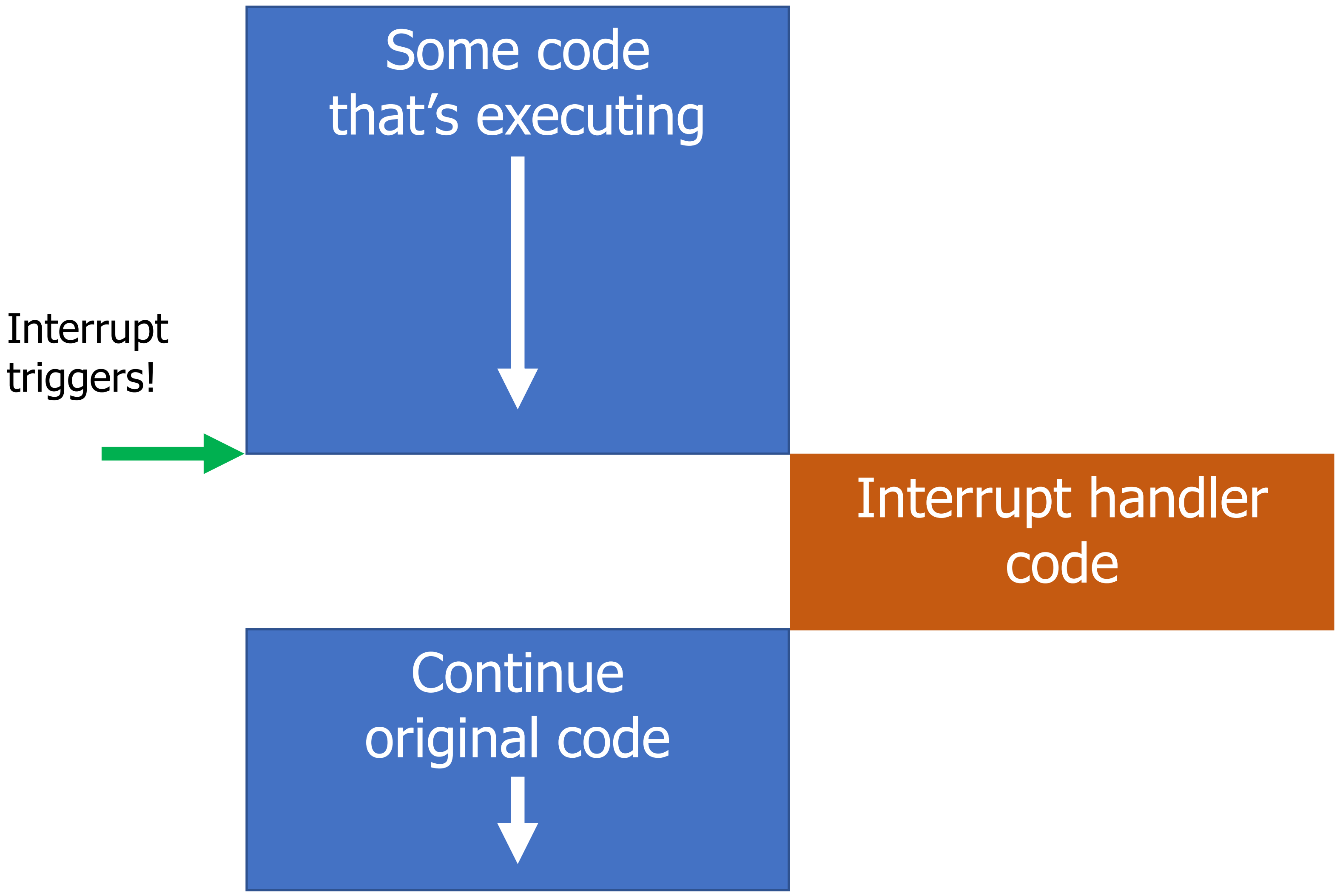


1. while STATUS==BUSY; Wait
   - (Need to make sure device is ready for a command)
2. Write value(s) to DATA
3. Write command(s) to COMMAND
4. while STATUS==BUSY; Wait
   - (Need to make sure device has completed the request)
5. Read value(s) from Data

This is the "polling" model of I/O.

"Poll" the peripheral in software repeatedly to see if it's ready yet.

Reminder: Interrupts, visually

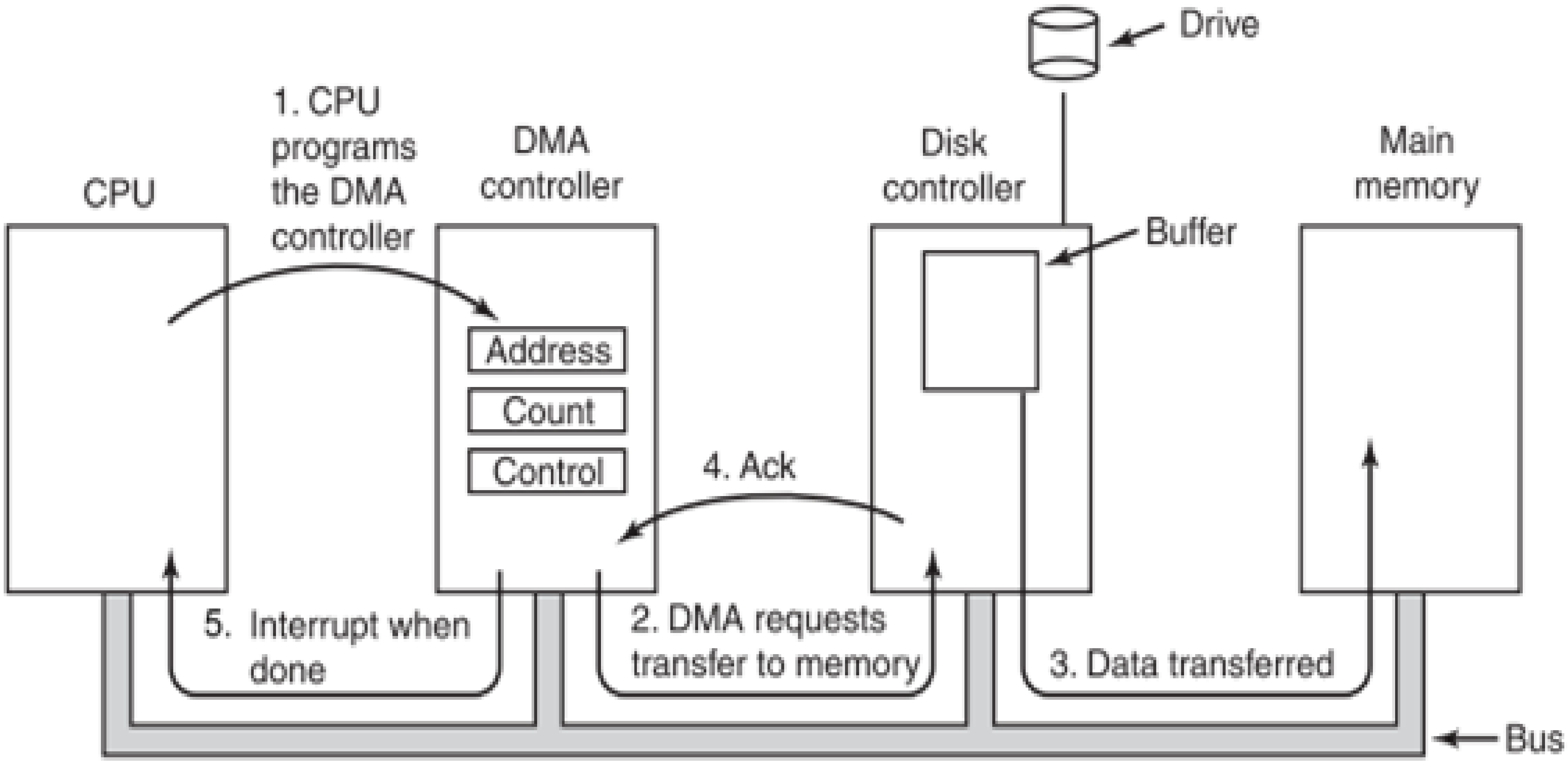


Direct Memory Access (DMA)

- Even with interrupts, providing data to the peripheral is time consuming
  - Need to be interrupted every byte, to copy the next byte over

- DMA is an alternative method that uses hardware to do the memory transfers for the processor
  - Software writes address of the data and the size to the peripheral
  - Peripheral reads data directly from memory
  - Processor can go do other things while read/write is occurring

General-purpose DMA



Full peripheral interaction pattern

1. Configure the peripheral
2. Enable peripheral interrupts
3. Set up peripheral DMA transfer
4. Start peripheral

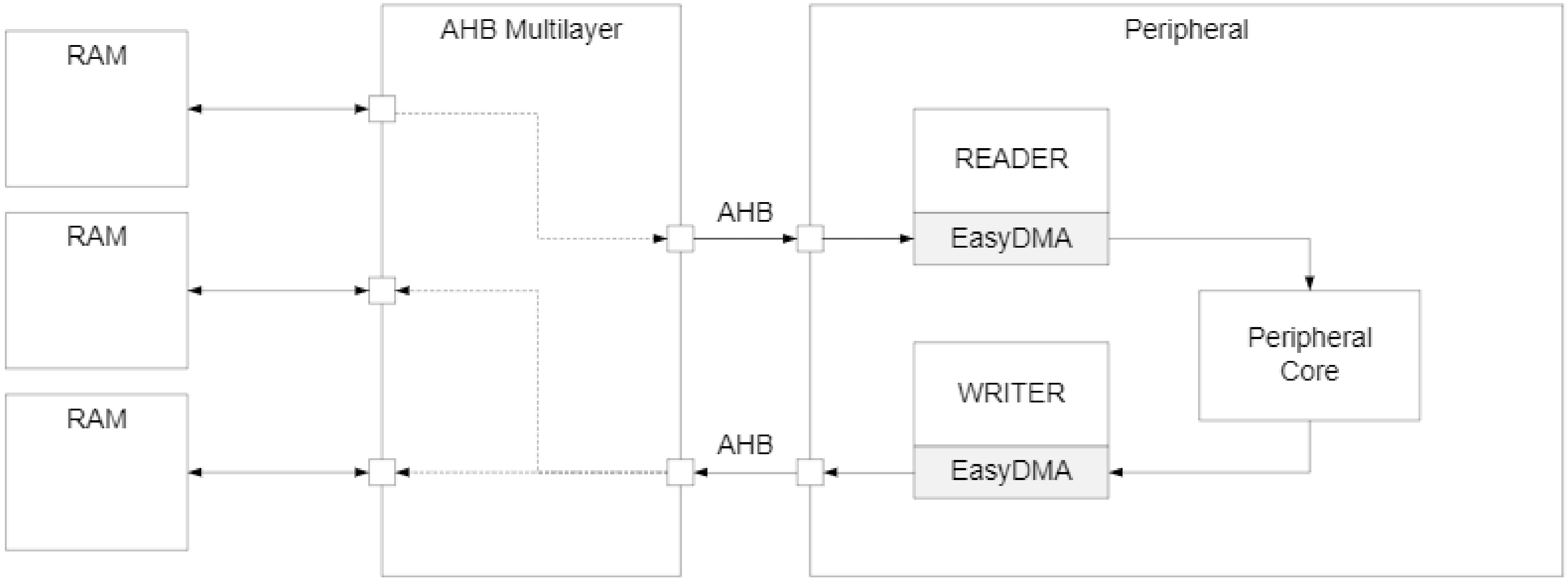
Continue on to other code

5. Interrupt occurs, signaling DMA transfer complete
6. Set up next DMA transfer

Continue on to other code, and repeat

Special-purpose DMA

- nRF52 uses “EasyDMA”, which is built into individual peripherals
  - Only capable of transferring data in/out of that peripheral
  - Easier to set up and use in practice
  - Only available on some peripherals though (no DMA for TEMP)



Warning: addresses for DMA buffer MUST be in RAM!

Break + Open Question

- What kinds of peripherals/devices should you use the DMA for?

Break + Open Question

- What kinds of peripherals/devices should you use the DMA for?

  - Anything where there is a lot of data coming in over a period of time
    - Either a big buffer of lots of data, like a radio message

    - Or a bunch of individual samples, coming in quickly

  - Devices
    - Messages to/from other devices (radios, wired busses)
    - Sensor readings (if read quickly)

    - Canonical example from general computing: disks (HDD/SSD)

Outline

- DMA
- **Prototyping**
- Digital Circuits
- Components

Prototyping goals

- Does this thing work at all?
  - Particular IC
  - Circuit layout
  - Software design
  - etc.

- Sometimes before doing something more serious with it
  - Design a PCB, Make a product, etc.

  - Not uncommon that the prototype is as far as you'll get

Isolating tests

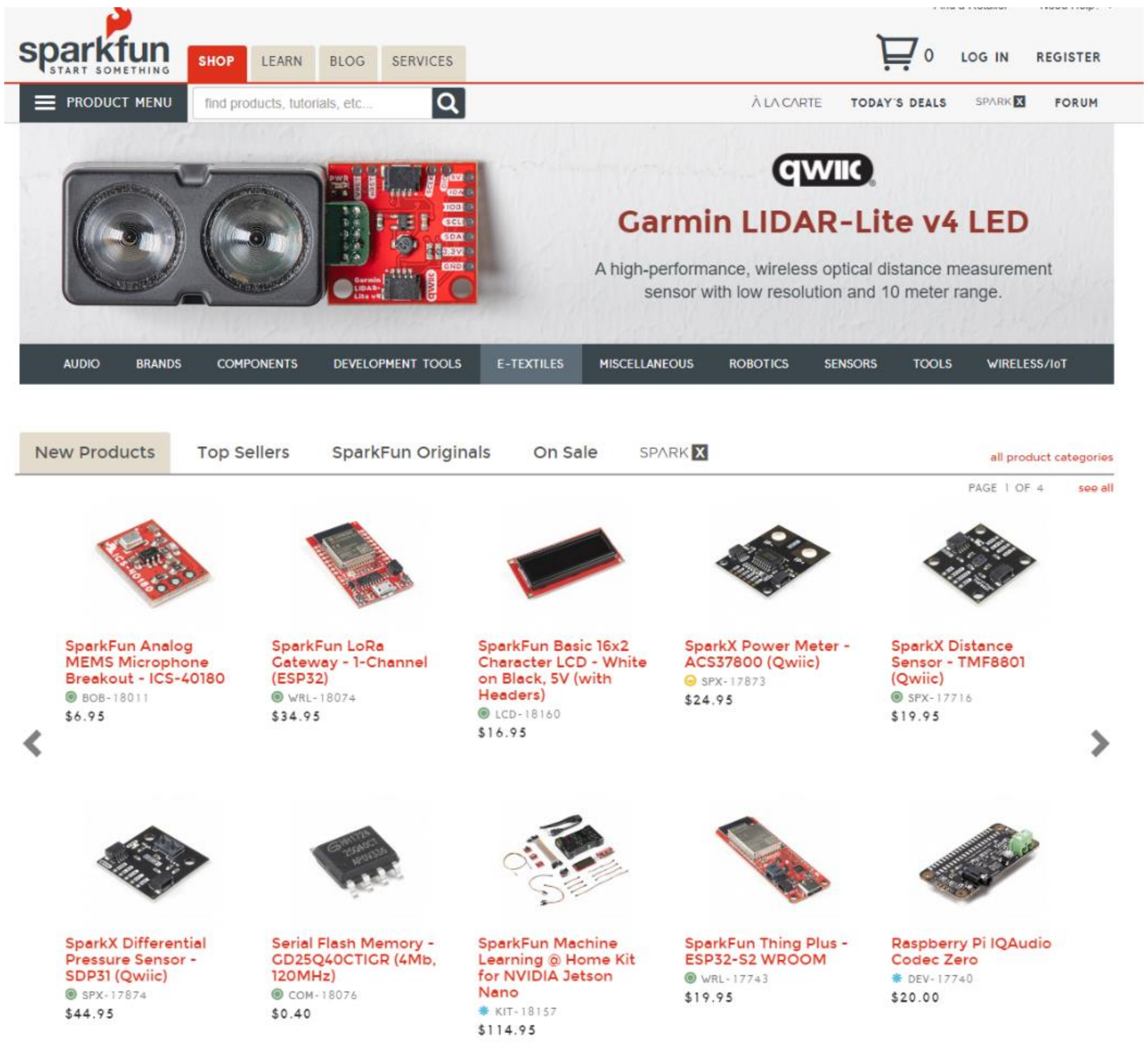
- The goal when prototyping is to isolate the question at hand

- Do consider
  - New sensor/IC/component/whatever

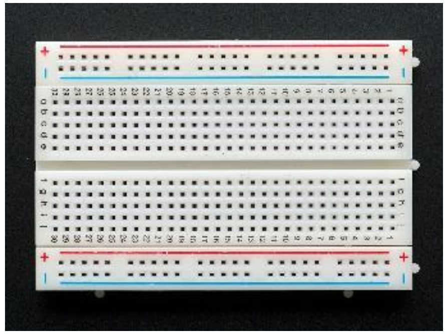
- Do not consider
  - Power
  - Interference
  - Enclosure
  - Stable microcontroller
  - Soldering skills

Buying Parts

- Prototyping vendors
  - Where you look for cool stuff to buy
  - Sparkfun
  - Adafruit
- Electronics vendors
  - Where you buy parts when you know what you need
  - Digikey
  - Mouser

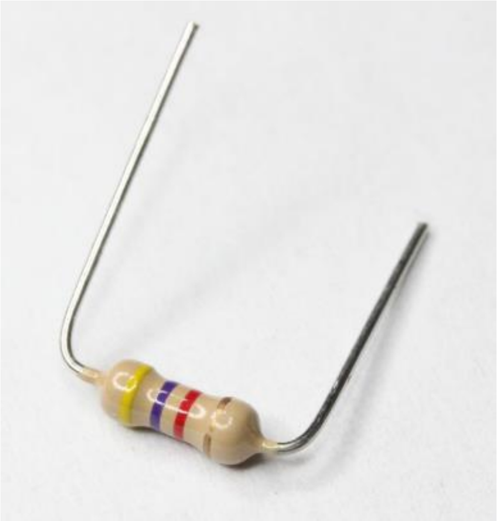


Prototyping methods

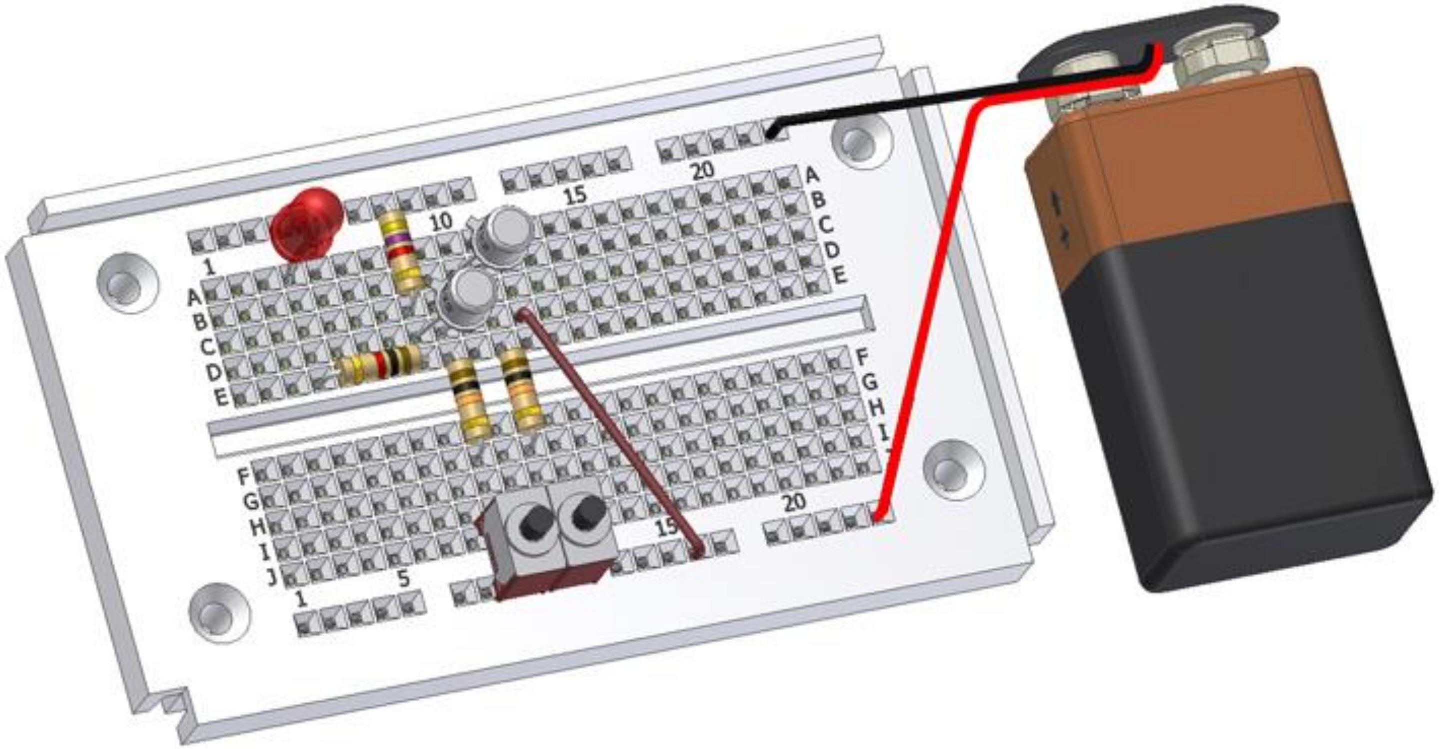


- Breadboarding
  - Plug and connect components as needed
  - Build up arbitrarily complex designs from nothing
- Development kits
  - Pre-fabricated systems design for testing components
- Small-scale test PCBs
  - Design a PCB that demonstrates the thing you’re interested in
    - Making a PCB is less hard than some might think (Eagle, Fritzing, etc.)
    - $20-30 for small, low-speed PCBs from batch services like OSHPark

Breadboards for prototyping

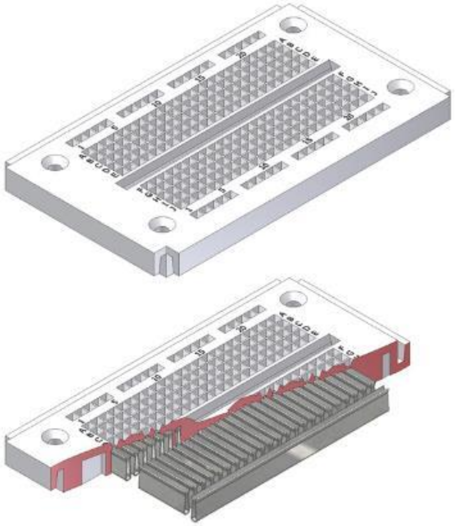


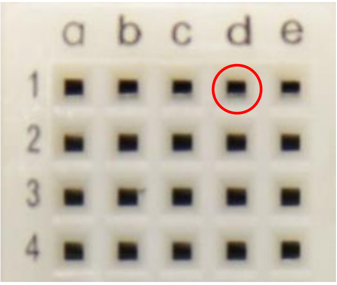
- Reusable platform for temporary circuits
- Plug in jumper wires and through-hold components



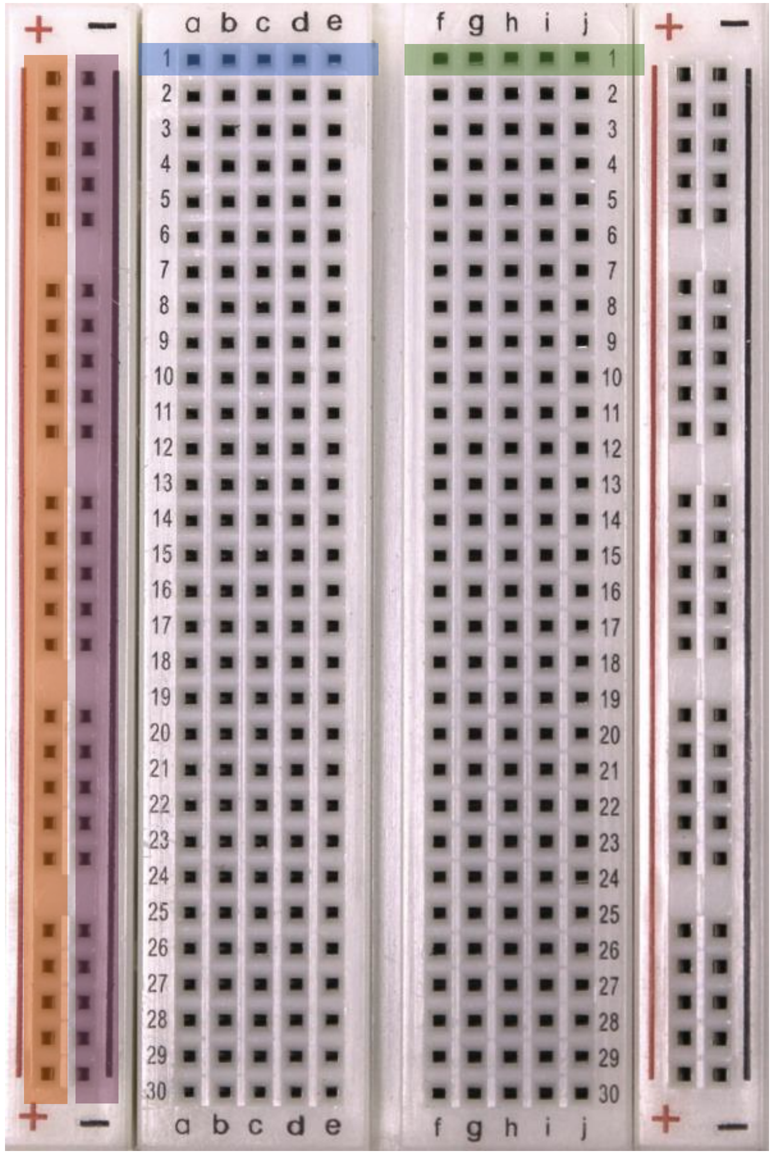
How a breadboard works

- Component leads and wires are inserted into holes in the breadboard
- Half-rows of five holes are connected
- Vertical columns are connected for power/ground



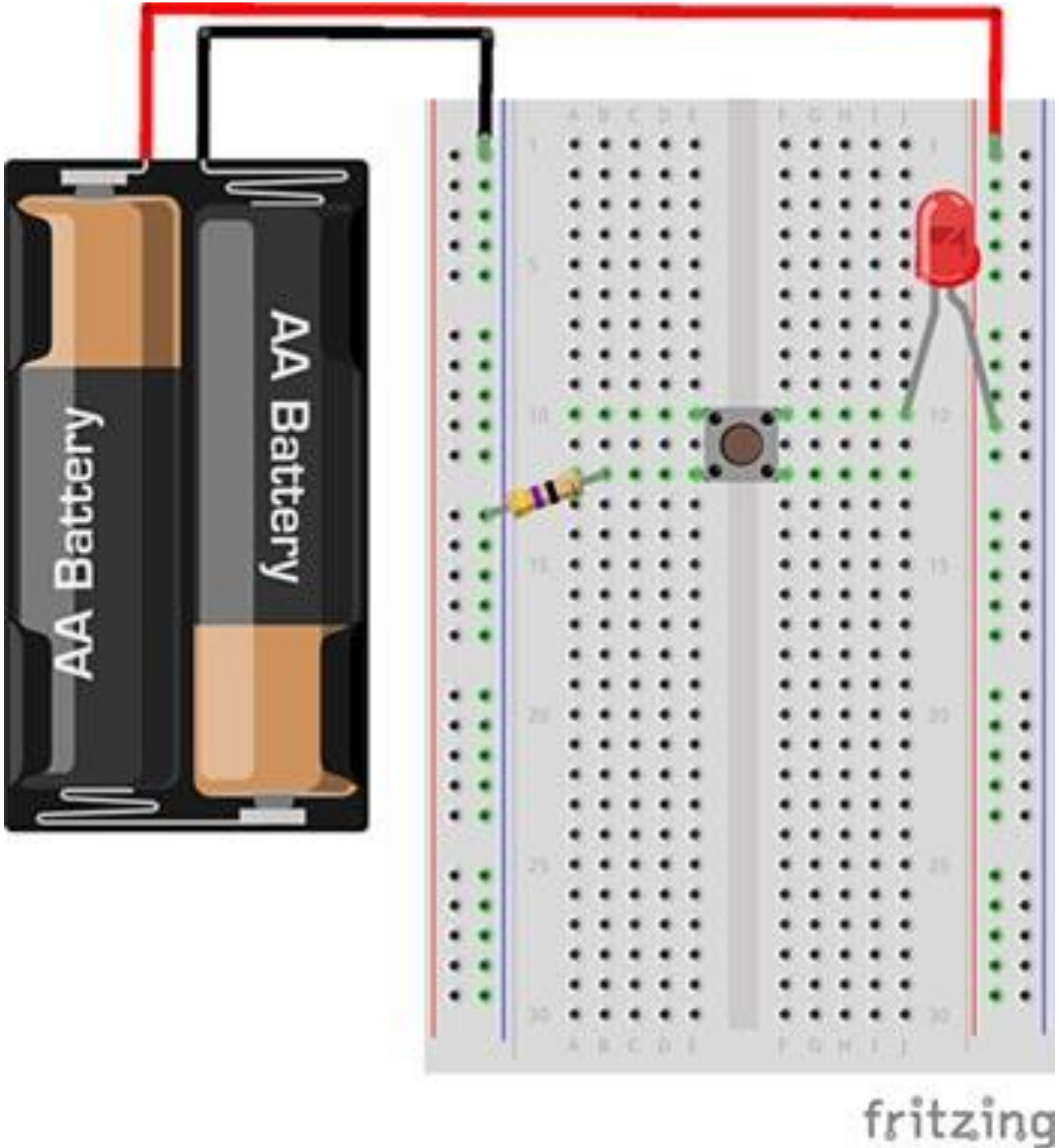


Holes to insert wires



Breadboard LED example

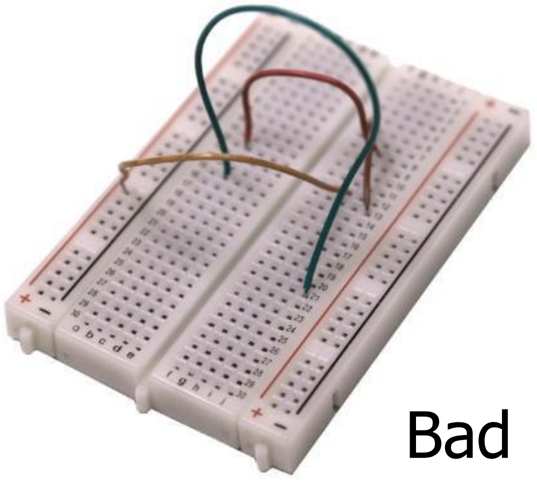
- Uses button to control LED



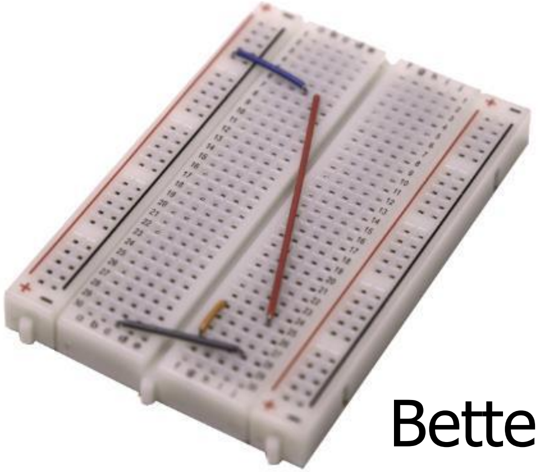
Breadboard guidelines

- Long wires in large bird nests makes debugging very difficult
  - Shorter, constrained wires are easier to understand
  - In this class, we'll only have large jumper wires though…

- Use the minimum jumpers necessary, mostly use breadboard for connections



Bad

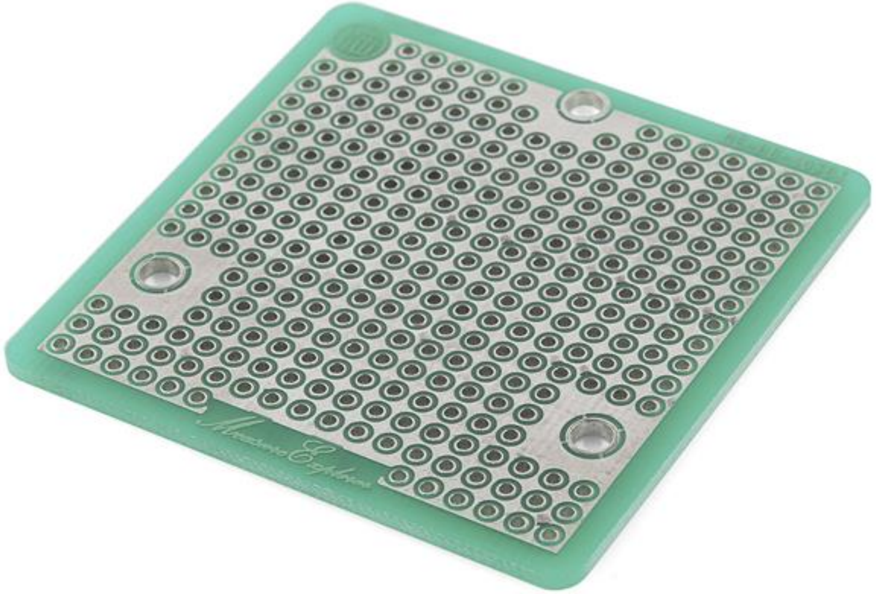


Better

More permanent breadboards

- Breadboards are also known as “Solderless Breadboards”

- Protoboard allows configurable circuits
  - Solder jumper wires between locations
  - Solder adjacent pads to form connection



- Usually not worth it (just make a PCB)
  - Does solve core problem of breadboards: things getting unintentionally unplugged
  - Might be useful for some projects!

When to not use breadboards

- Breadboards work great for digital circuits and simple analog!

- High voltage/current are bad for breadboards
  - Honestly, anything above 12 volts DC shouldn’t be in a breadboard
  - Also avoid high-power applications above a few Watts
  - Never put AC in a breadboard!

- Sensitive analog circuits
  - Particularly anything sensitive to capacitance may not work right
  - Sets of metal holes with strips connecting them function as capacitors

- Anything in long term use

Outline

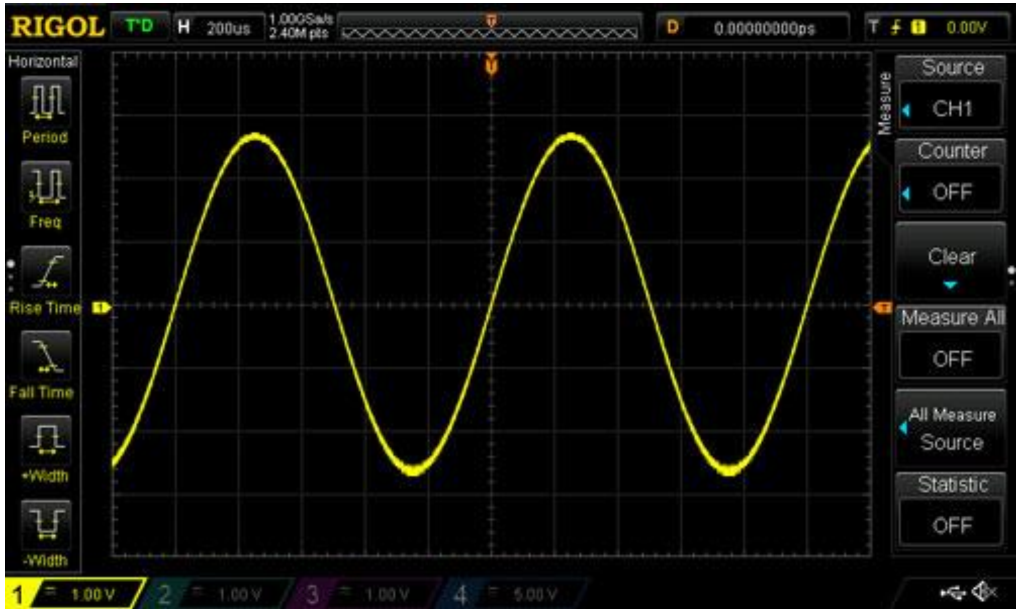
- DMA
- Prototyping
- **Digital Circuits**
- Components

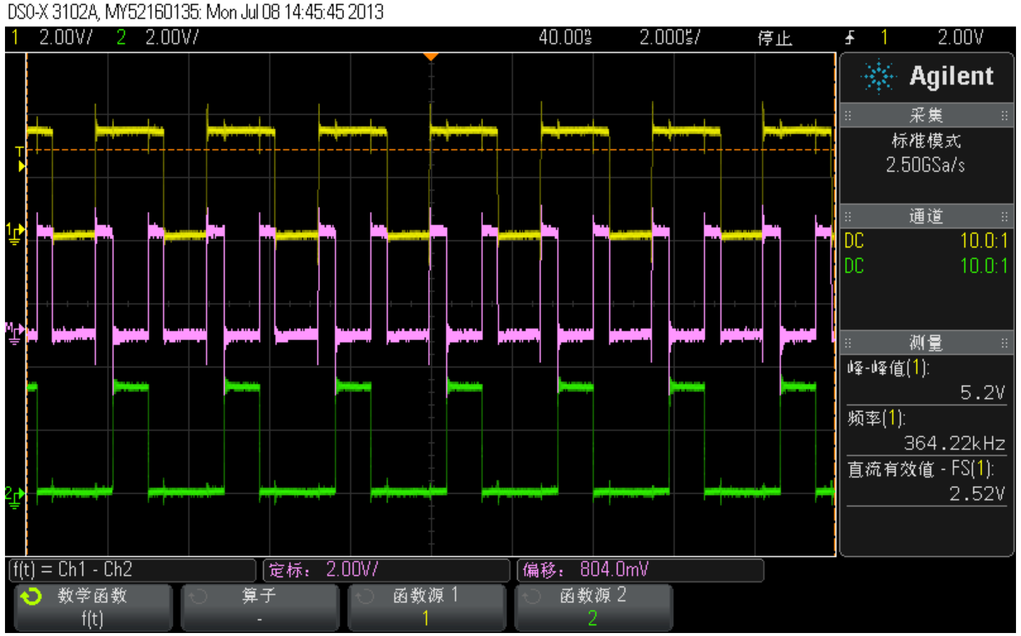
Digital signals

- Exist in two states:
  - High (a.k.a. Set, a.k.a. 1)
  - Low (a.k.a. Clear, a.k.a. 0)

- Simpler to interact with
  - Constrained to two voltages
  - With quick transitions between the two

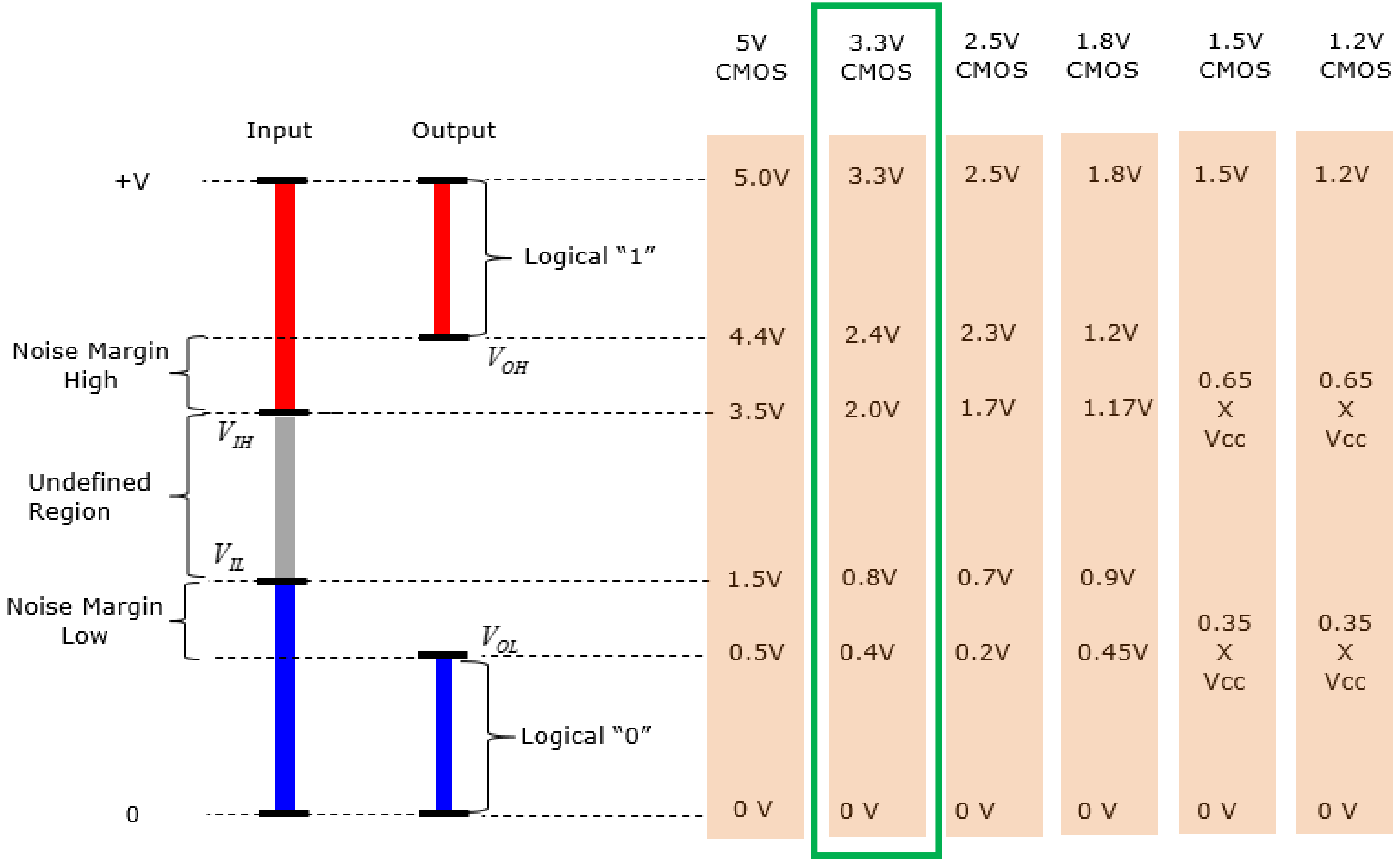
  - No math for voltage level
    - Either high or low





Digital signals map to voltage ranges

- Upper range is high signal
  - ~0.7*VDD
- Bottom range is low signal
  - ~0.3*VDD
- Middle is undefined
  - Only exists during transitions



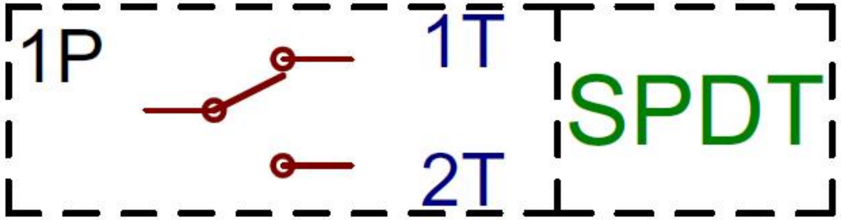
http://www.sharetechnote.com/html/Electronics_CMOS.html

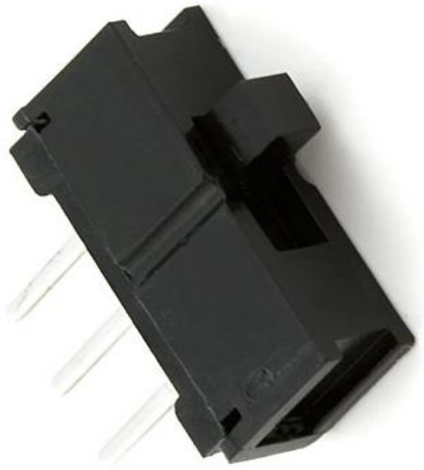
Digital circuits

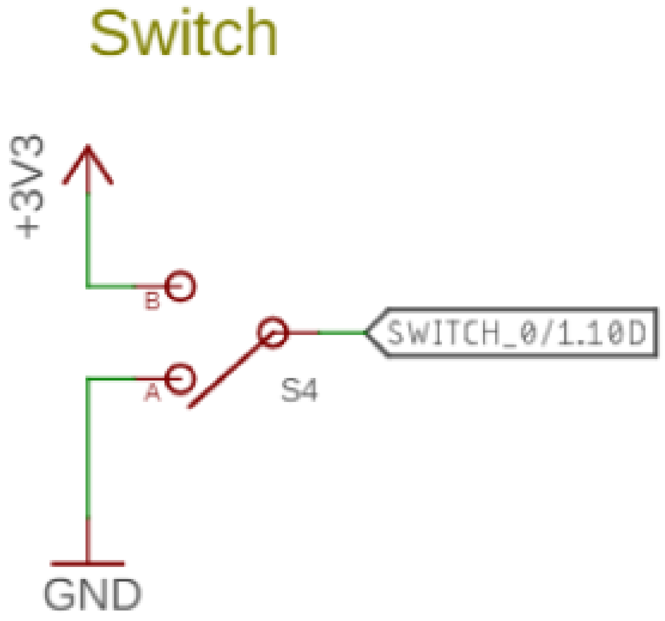
- Connecting components together with digital signals
  - Mostly ICs
  - Also buttons/switches and LEDs
- Way simpler than analog circuits
  - Mostly connecting boxes with wires
  - Plus a few resistors here and there
- An abstraction
  - Not sufficient for fully understanding electronics behavior, but close

Switches

- Single Pole, Double Throw switch
  - Middle pin (Pole) connects to one of two outer pins (Throws)





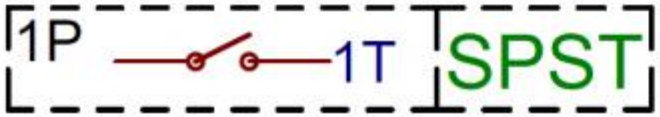


- For controlling microcontrollers
  - Often connect outer pins to VCC and Ground respectively
  - Input then goes High or Low depending on switch state

https://learn.sparkfun.com/tutorials/button-and-switch-basics/

Buttons

- Single Pole, Single Throw switch
  - Pole pin either connects to Throw pin or is disconnected
  - Come in normally-closed (connected) and normally-open (disconnected)



Disconnected circuits

- When button is pushed, input signal is low
- **What is the value of the input when the button is unpressed?**

Disconnected circuits

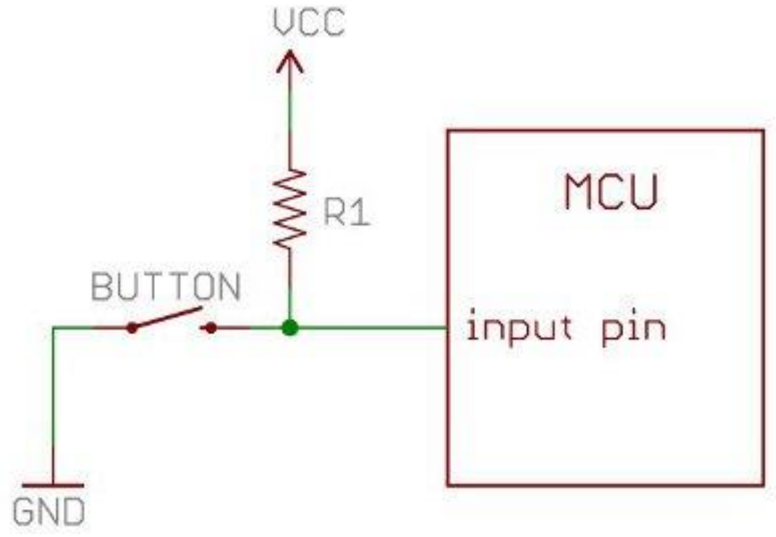
- When button is pushed, input signal is low
- **What is the value of the input when the button is unpressed?**
  - Floating! Could be any voltage

  - Solution: need to connect weakly to either high or low voltage

Current flows through the “path of least resistance”

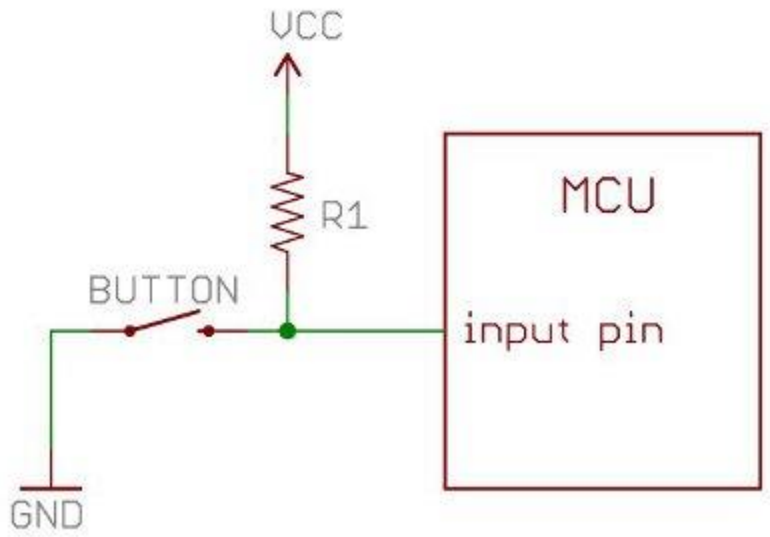
- Simplification
  - Works well for the types of circuits we use

- Pull-up resistor
  - When button is open (disconnected), the only path is through the resistor
  - When button is closed (connected) the least resistance path is through the button to Ground



Pull-up resistors and pull-down resistors

- Resistor sets the "default" value of a wire
  - Pull-up connects to VCC
  - Pull-down connects to Ground
  - Usually 10-100 kΩ

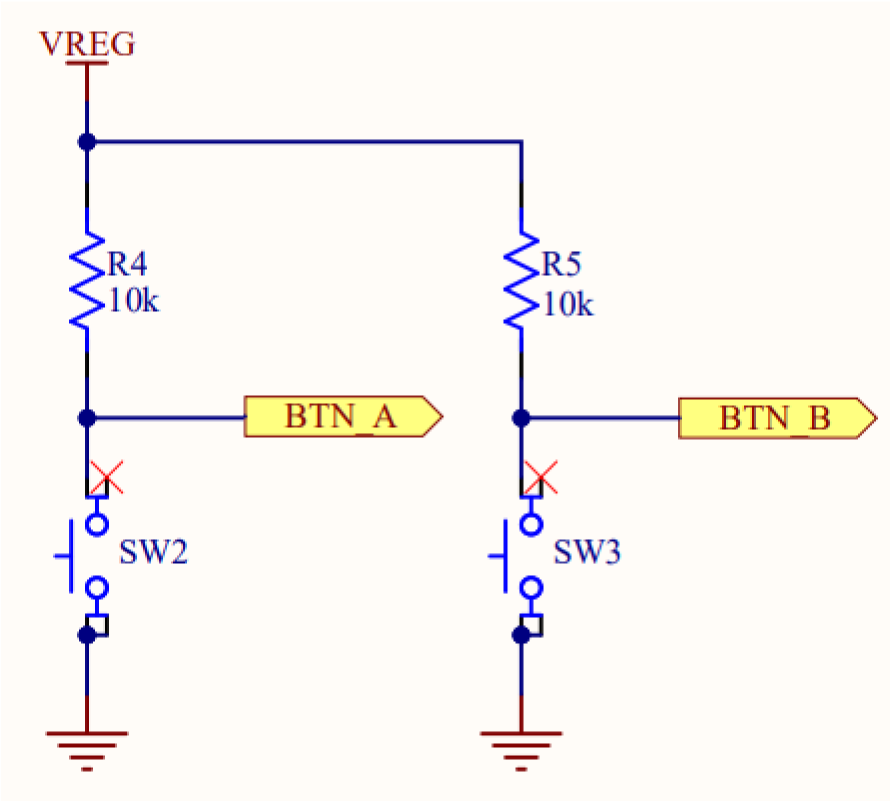


- When button is open (disconnected)
  - Connection through the resistor sets signal

- When button is closed (connected)
  - Signal is directly connected to a voltage source
  - Much lower resistance means that signal dominates

Buttons on the Microbit

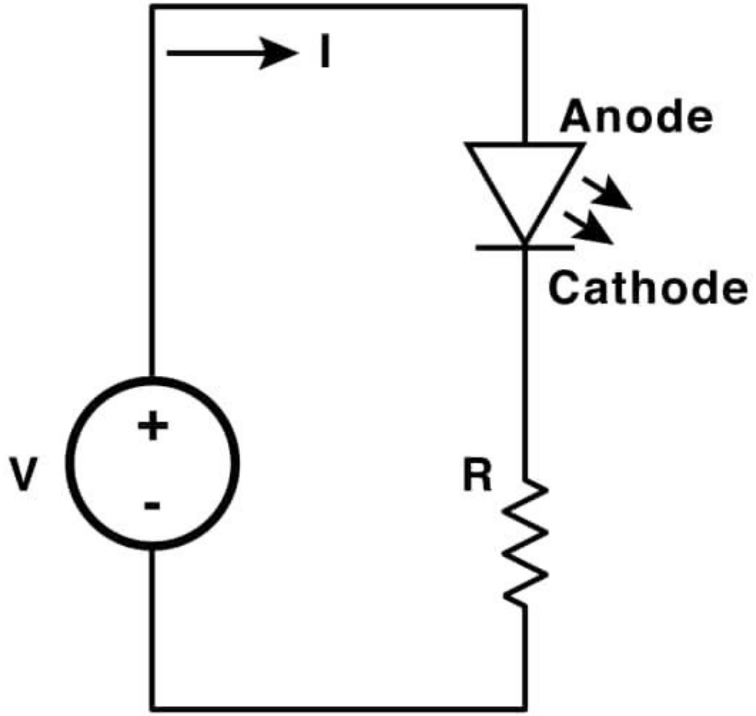
- Normally open buttons
  - Disconnected by default
- Active low signal
  - Activating (pushing) button creates a low signal
- Pull-up resistors
  - Set button signal high by default



LEDs

- Light Emitting Diodes
  - Generate light as current passes through them
  - Various colors available

- Diodes
  - Only allow current to go through one way
  - Not particularly relevant for LEDs
    - Treat as a digital component

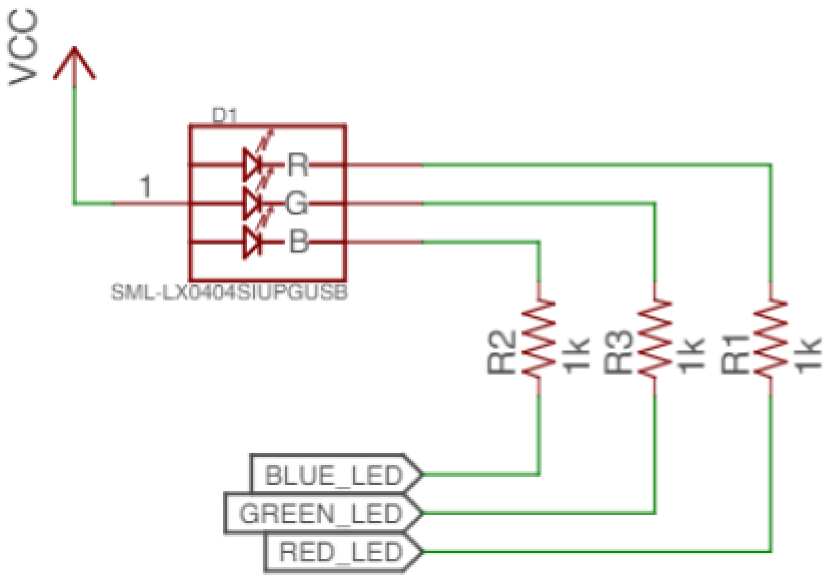


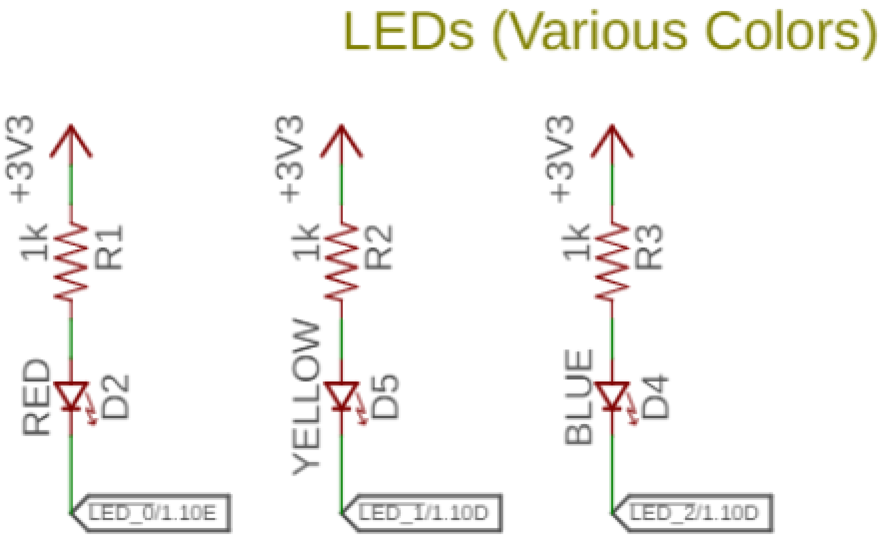
- Connect anode to high voltage and cathode to ground
  - Plus a resistor to limit the total amount of current

https://learn.sparkfun.com/tutorials/light-emitting-diodes-leds

Active state for LEDs

- LEDs can be active high or active low depending on configuration
    - Active high is how people assume they work
    - Active low is often used instead
        - GPIO pins can usually sink more current than they can source

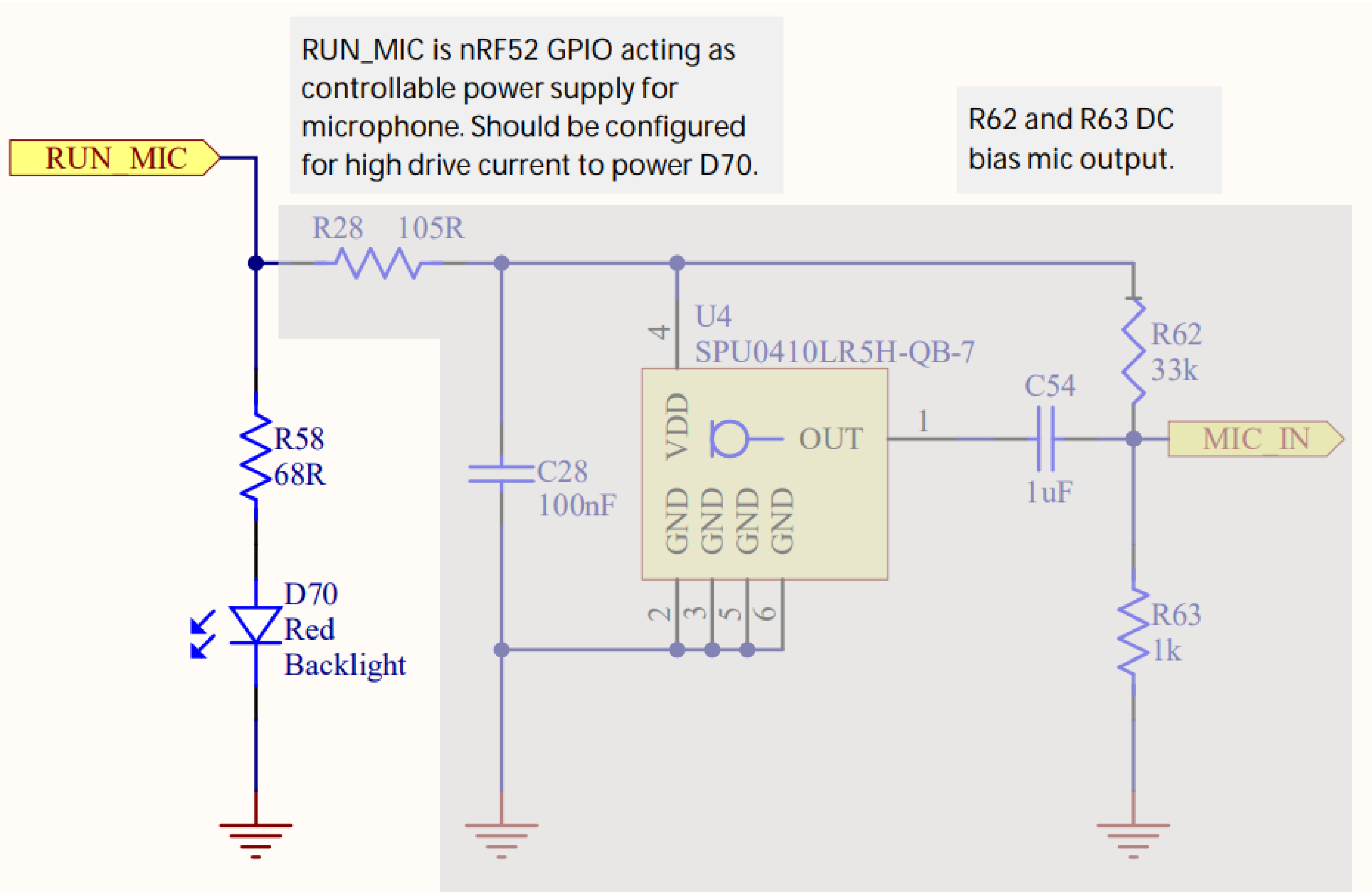




LEDs on the Microbit

- Microphone LED
  - Active high

- Simple to use, just set the GPIO high to enable it

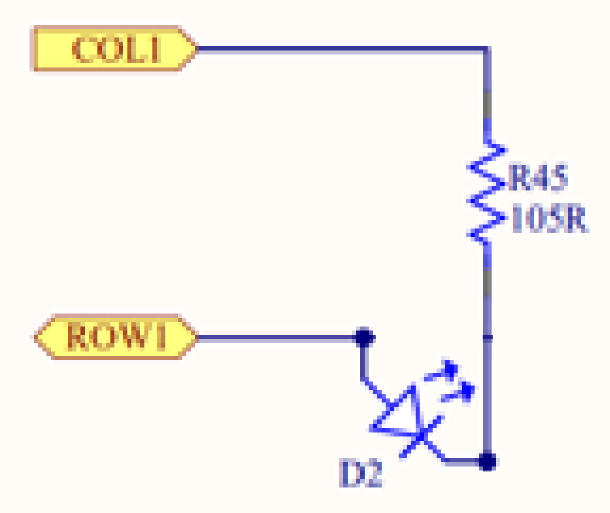


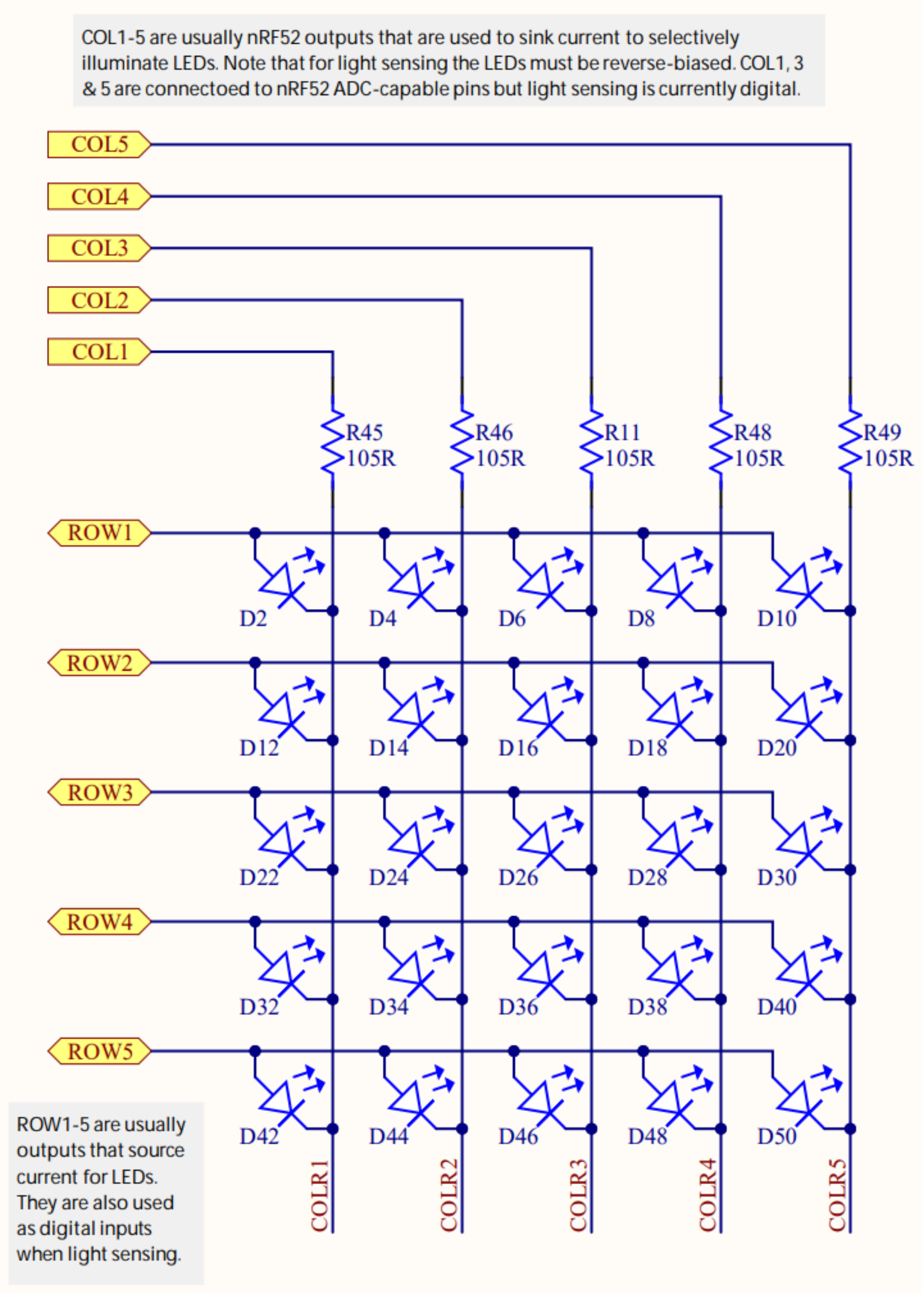
Ignore this other part for now

LEDs on the Microbit

- Use two GPIO pins to control each LED
  - Row high as VDD
  - Column low as Ground

- Remember, connections only exist where there are dots

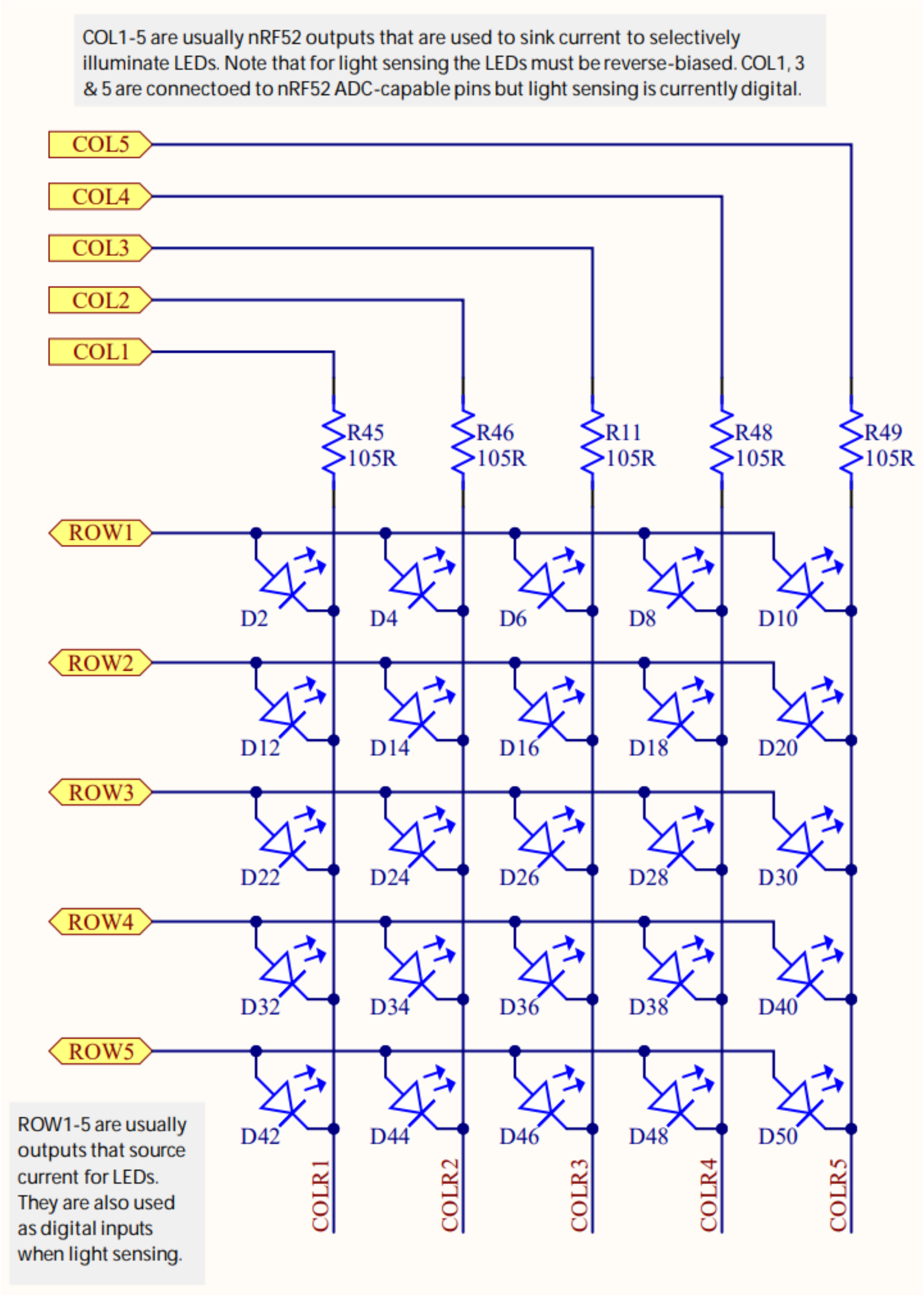




Controlling the LED matrix

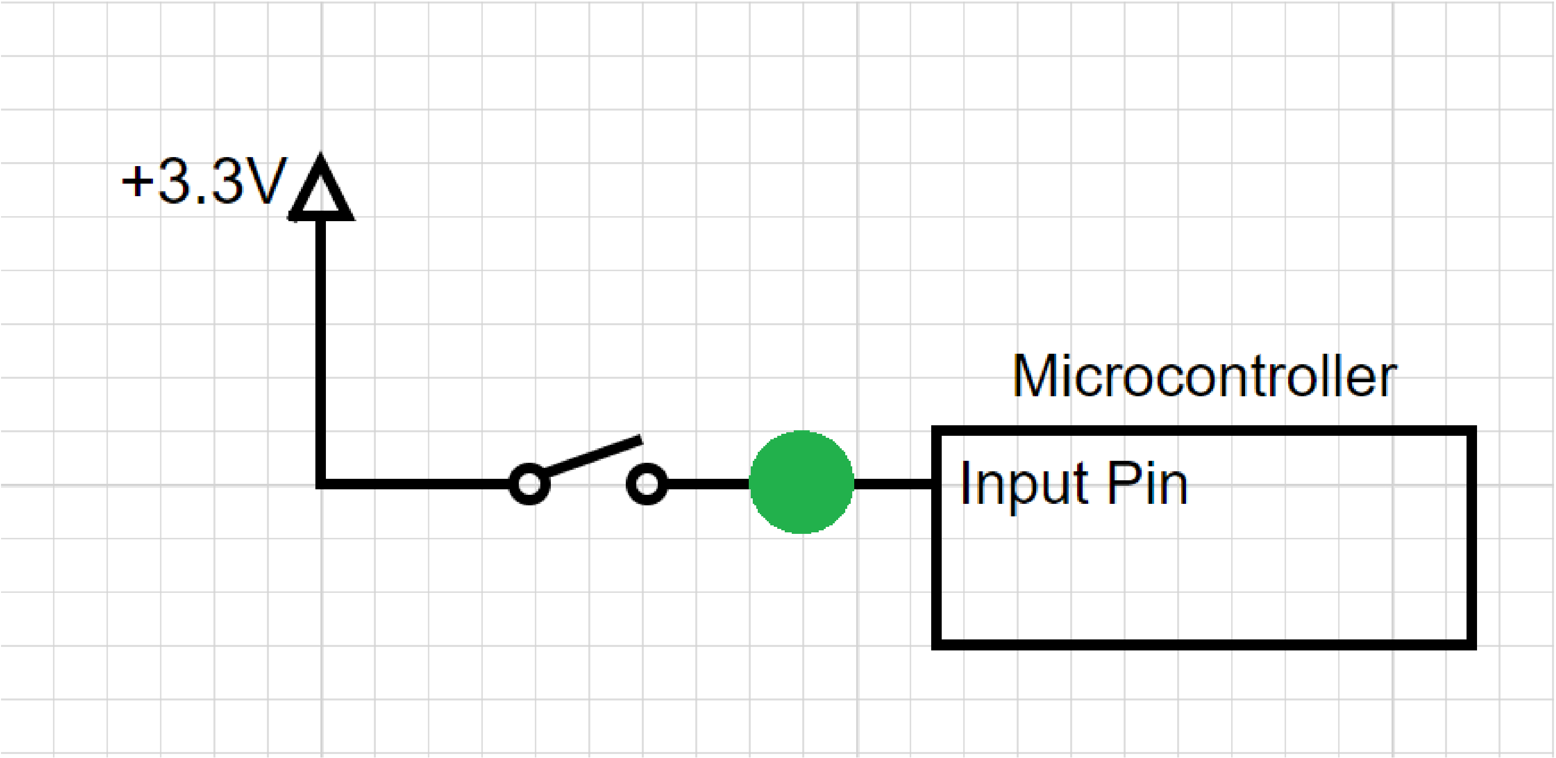
- Cannot individually control all LEDs simultaneously
  - Need to light one row at a time
  - Iterate rows quickly to make them appear on all the time

- We’ll have a lab on these later
  - Combines GPIO and timers



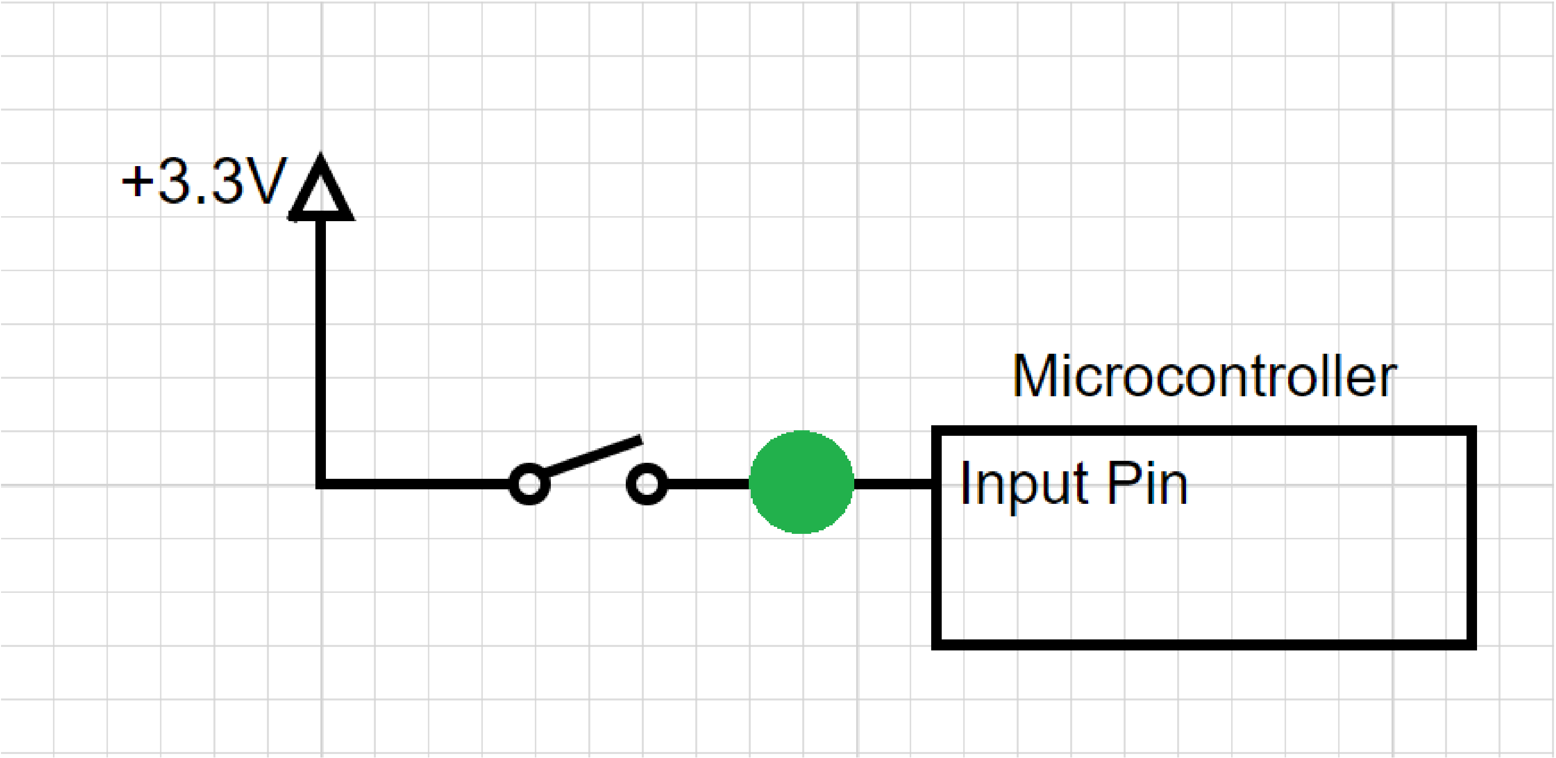
Break + Question

- Should the spot in green have?
  A. Pull-up Resistor
  B. Pull-down Resistor
  C. Either
  D. Neither



Break + Question

- Should the spot in green have?
  A. Pull-up Resistor
  B. Pull-down Resistor (needs to pull input low by default)
  C. Either
  D. Neither



Outline

- DMA
- Prototyping
- Digital Circuits
- **Components**

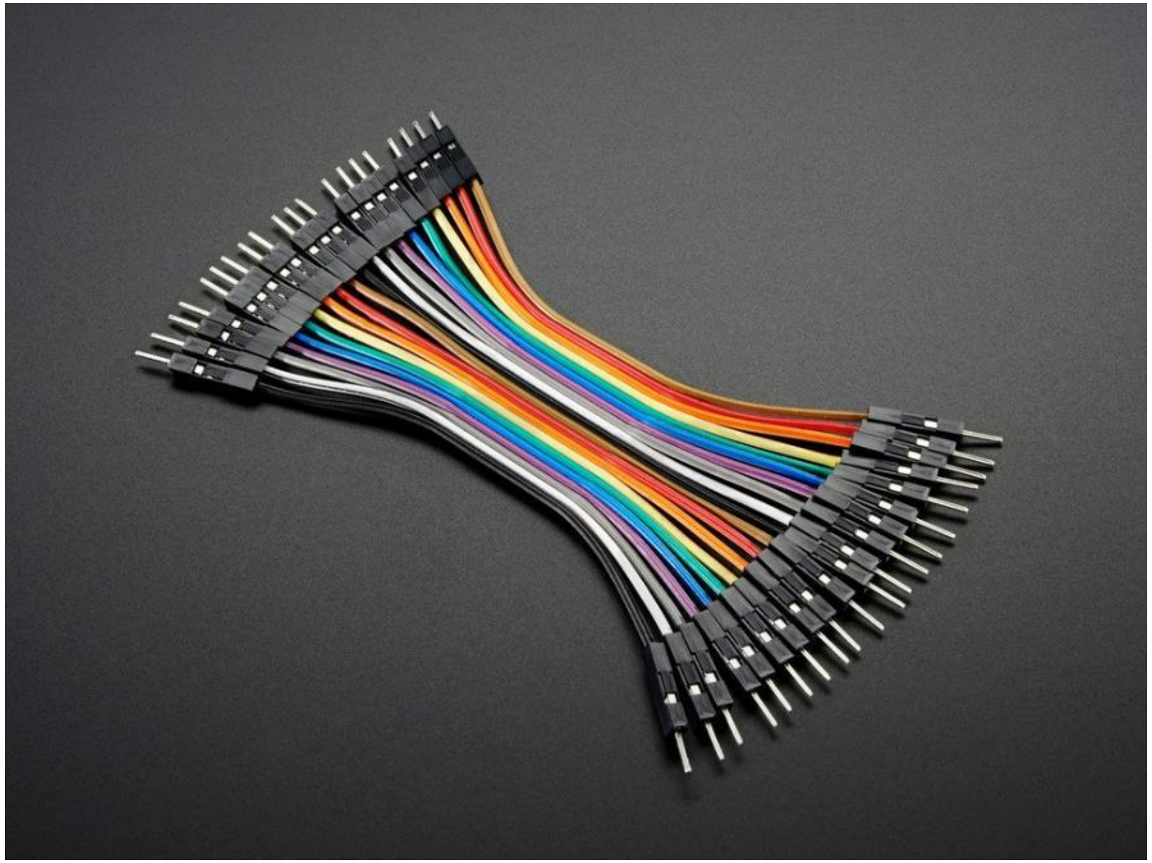
Prototyping with a breadboard

- What kinds of things might you use with a breadboard?
- Jumper wire
- Microbit!
- Resistors/Capacitors
- LEDs
- Buttons/Switches
- Analog Sensors
- Various other through-hole components
  - Transistors, Op-Amps, other ICs

https://www.adafruit.com/product/2975

Jumper wires

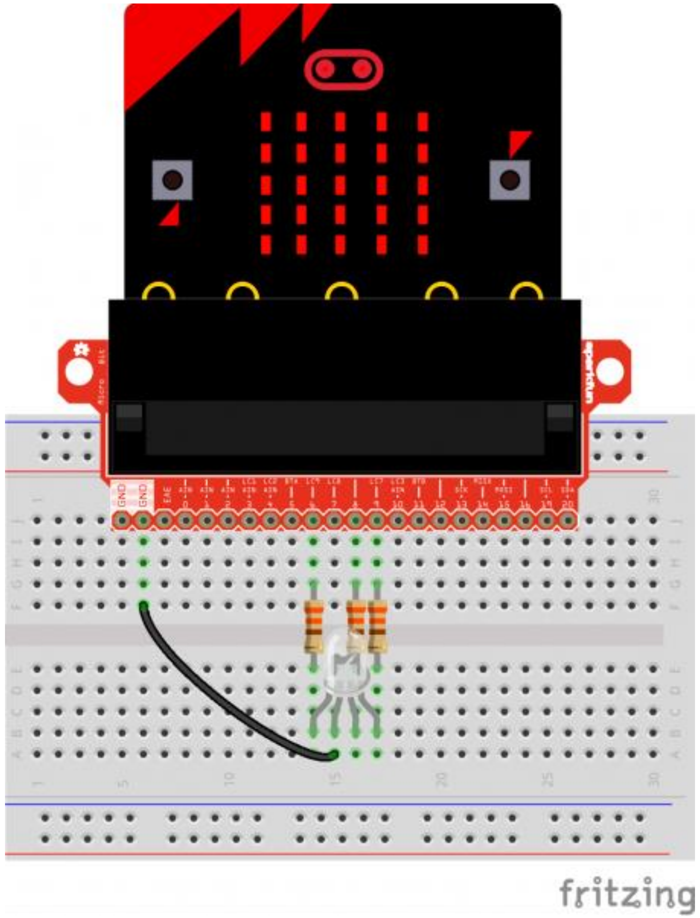
- Connect two rows in the breadboard together
- Recommendation:
  - Peel off sets of 2-4 wires and keep them stuck together
  - Often want to run multiple at once



Microbit

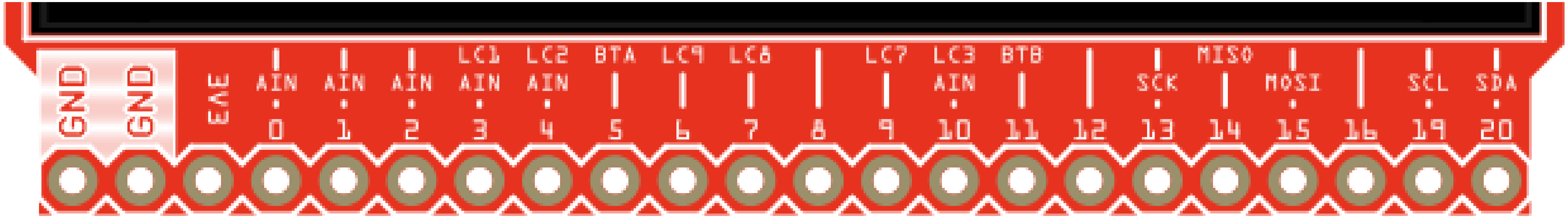
- Always connect LED matrix side up

- Breaks out various pins from board
  - Need to consult table to know which pins
  - https://tech.microbit.org/hardware/schematic/



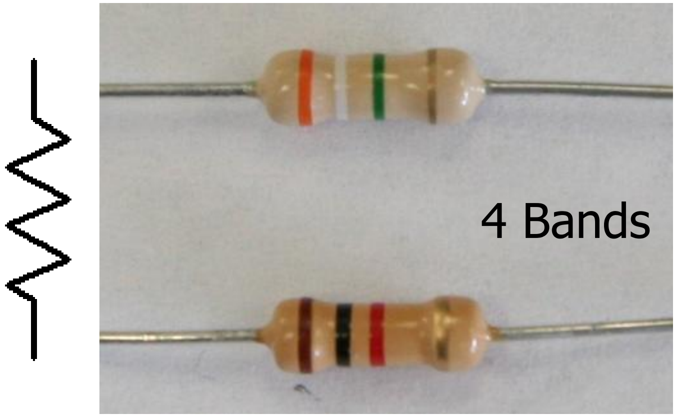
https://www.sparkfun.com/products/13989

https://learn.sparkfun.com/tutorials/microbit-breakout-board-hookup-guide

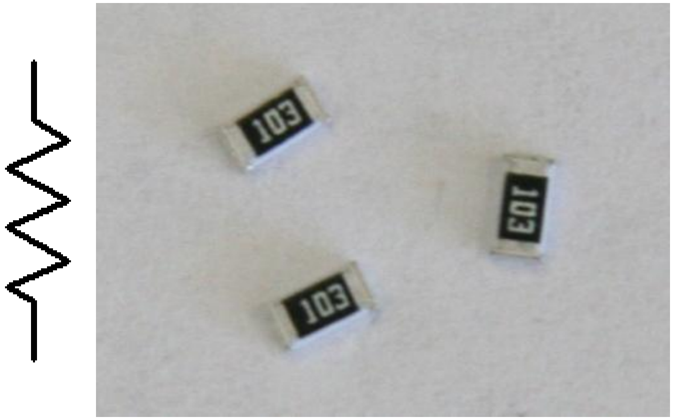


Resistors

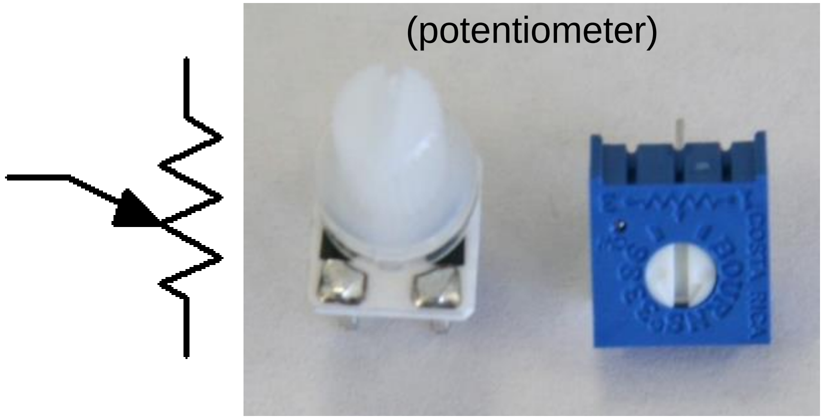
Carbon Film Resistors



Surface Mount Resistors

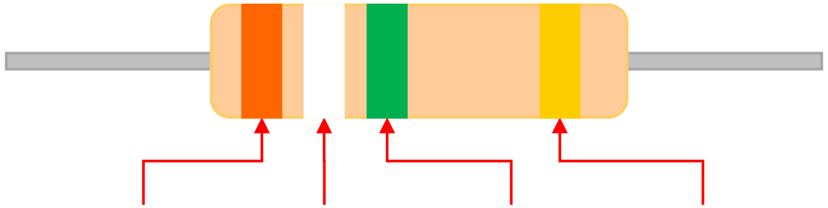


Variable Resistors



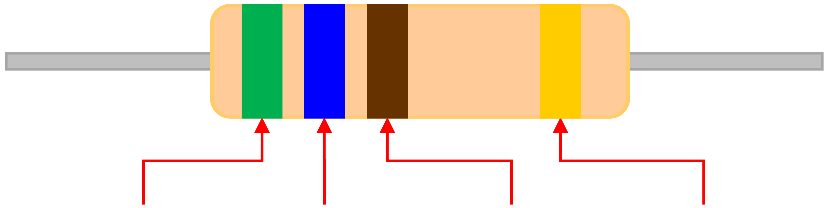
Resistor color codes

- Colored bands on resistors label the resistance value of the part
- First and second bands are the digits
- Third band is multiplier
- Fourth band is tolerance
  - Usually gold: +/- 5%



	1st Band	2nd Band	Multiplier	Tolerance
NONE				20%
Silver			0.01	10%
Gold			0.1	5%
Black	0	0	1	
Brown	1	1	10	
Red	2	2	100	
Orange	3	3	1K	
Yellow	4	4	10K	
Green	5	5	100K	
Blue	6	6	1M	
Violet	7	7	10M	
Gray	8	8	100M	
White	9	9	1000M	

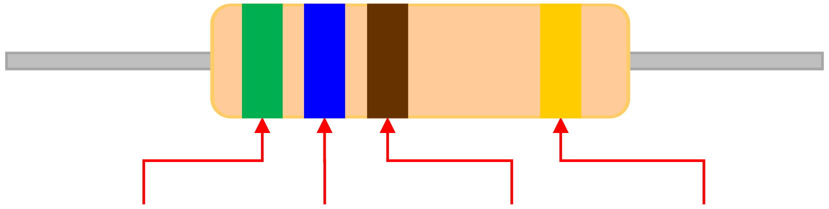
Example: determine the resistor



	1st Band	2nd Band	Multiplier	Tolerance
NONE				20%
Silver			0.01	10%
Gold			0.1	5%
Black	0	0	1	
Brown	1	1	10	
Red	2	2	100	
Orange	3	3	1K	
Yellow	4	4	10K	
Green	5	5	100K	
Blue	6	6	1M	
Violet	7	7	10M	
Gray	8	8	100M	
White	9	9	1000M	

Example: determine the resistor

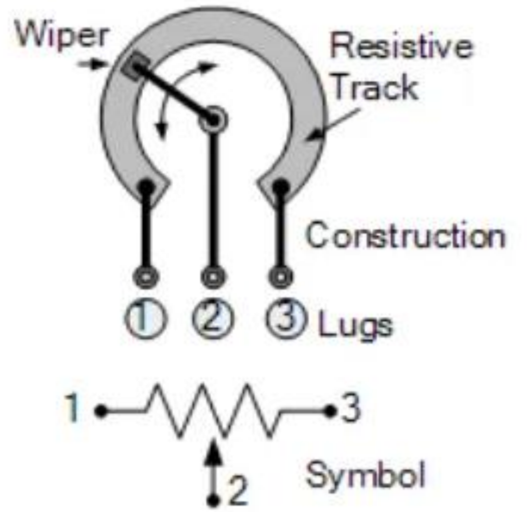
- 56 x 10 Ω = 560 Ω (±5%)



	1st Band	2nd Band	Multiplier	Tolerance
NONE				20%
Silver			0.01	10%
Gold			0.1	5%
Black	0	0	1	
Brown	1	1	10	
Red	2	2	100	
Orange	3	3	1K	
Yellow	4	4	10K	
Green	5	5	100K	
Blue	6	6	1M	
Violet	7	7	10M	
Gray	8	8	100M	
White	9	9	1000M	

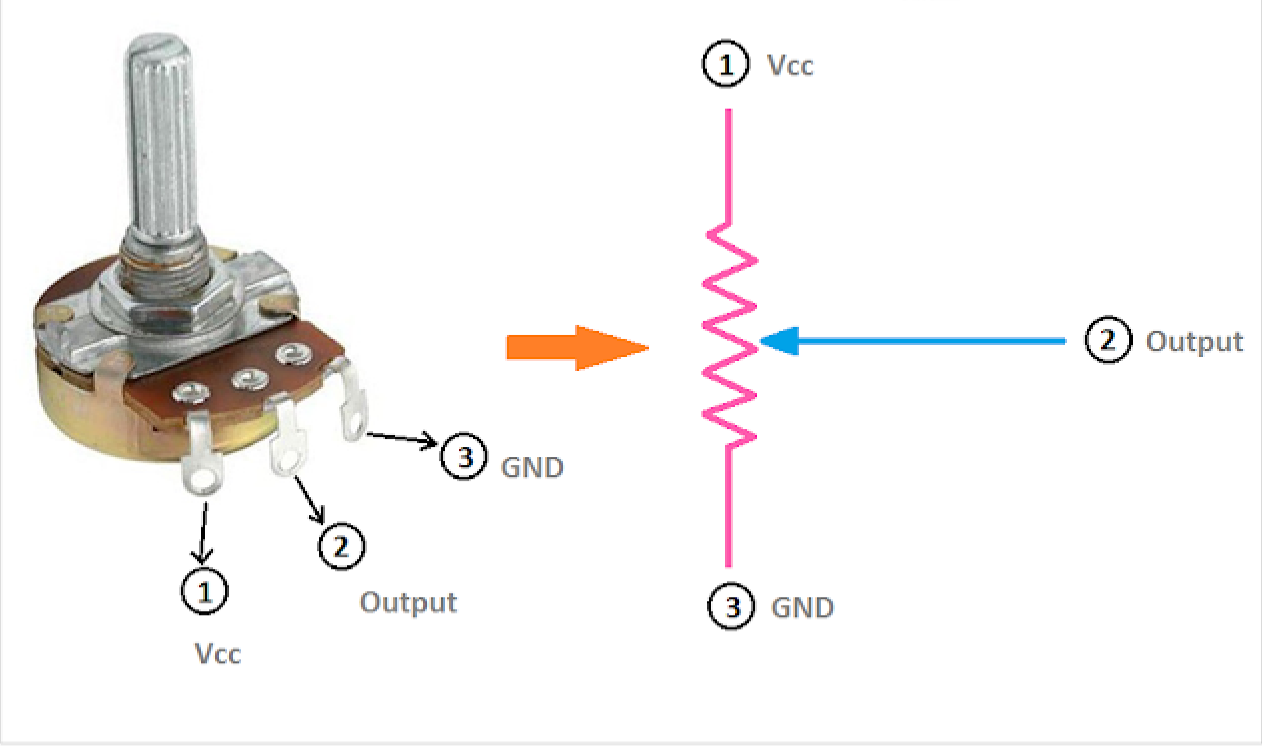
Potentiometers

- Vary resistance between zero and some maximum
  - 1 kΩ, 10 kΩ, 100 kΩ common



- Connect middle and an edge for just a changeable resistor

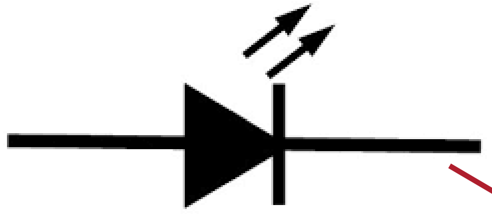
- Middle terminal is a movable resistor divider
  - Knob changes middle output if outer pins are VCC and Ground



LEDs

- Directional component: only allows current to flow one way

- Shorter side is the negative one
  - i.e. where current flows to



Schematic Symbol

Negative (-) lead

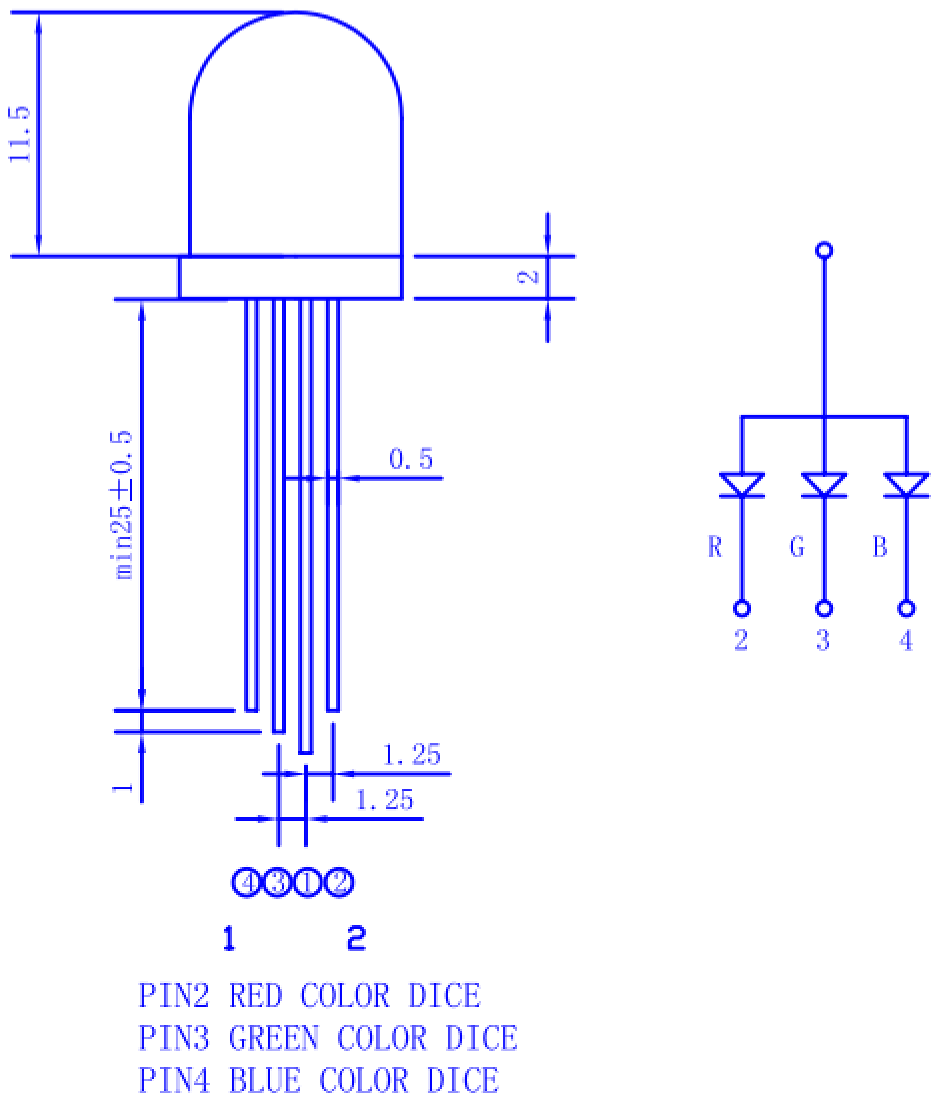
Surface-mount LED

Larger metal component inside of case or case flat spot is cathode or negative (-) lead

Shorter wire is cathode or negative (-) lead

RGB LED

- Three different colors of LED in a single large diffuser
- Short leads are negative ends
  - One for each color
- Long lead is common power
  - Common anode
- Combinations of LEDs give other colors
  - Cyan, Yellow, Violet, White



https://cdn-shop.adafruit.com/datasheets/FLR-100WAS-RGB.pdf

Sensors

- Thermistor

- Photoresistor

We’ll come back to these in a future lecture

Outline

- DMA
- Prototyping
- Digital Circuits
- Components