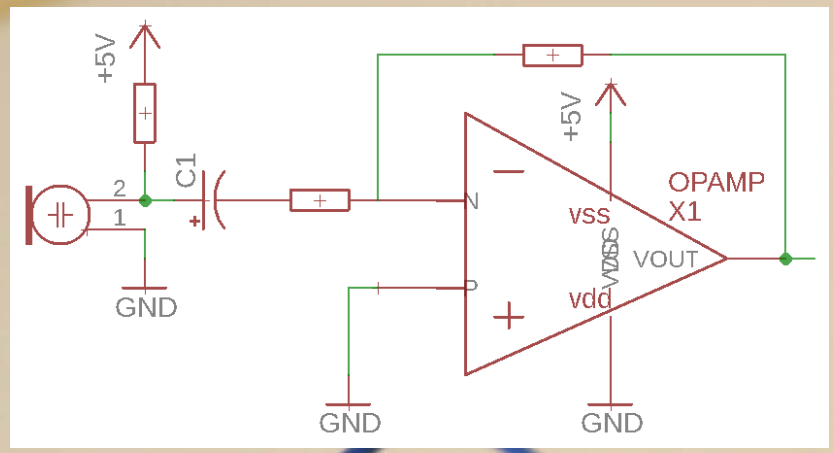
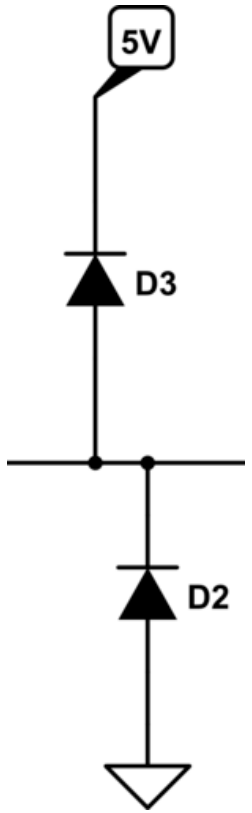
There are *at least* 7 errors in this setup

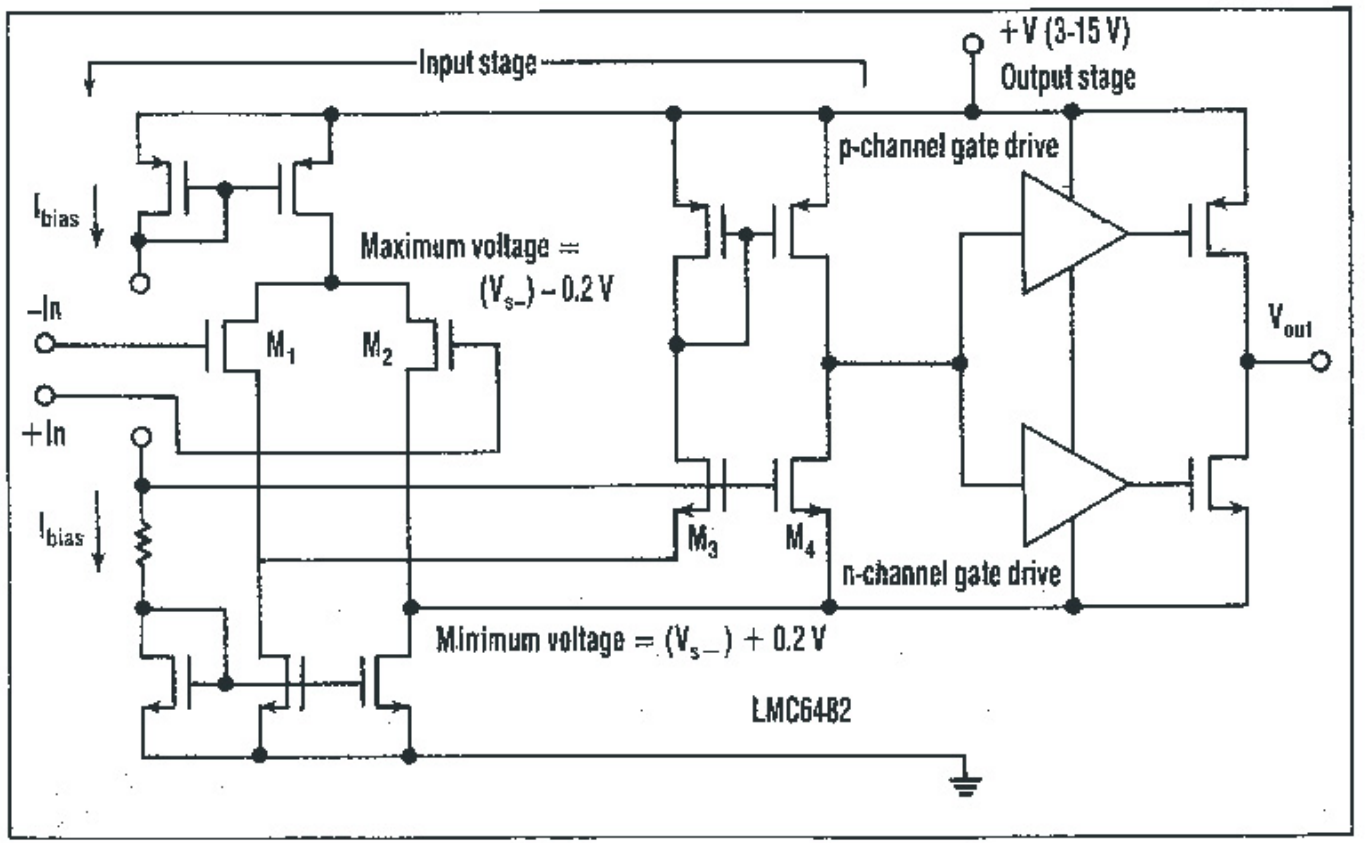


Can anyone see a more fundamental problem with this circuit?

Amplifiers

- *What happens when you send in voltage outside of the supply rail?*
  - Best case: it survives, or completely fails
  - Worst case: it sort of keeps working!





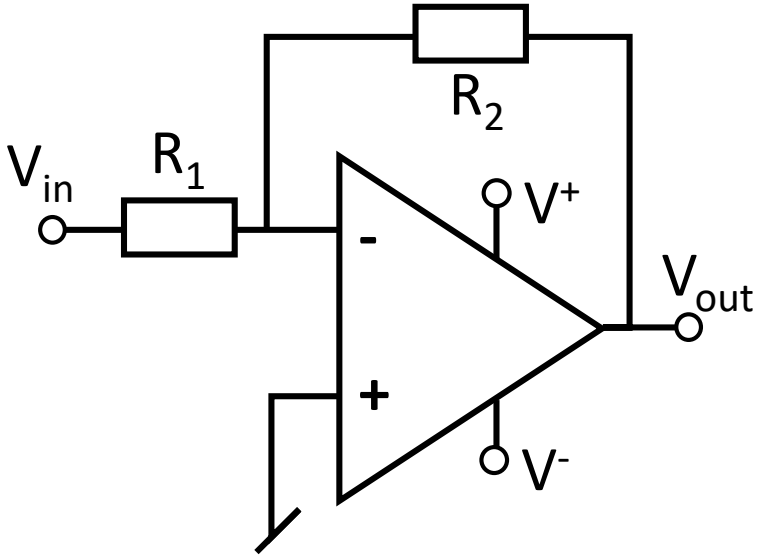
*Min: GND-0.3V

*Max: VCC+0.3V

DON’T GO BEYOND THE SUPPLY RAIL!

Amplifiers

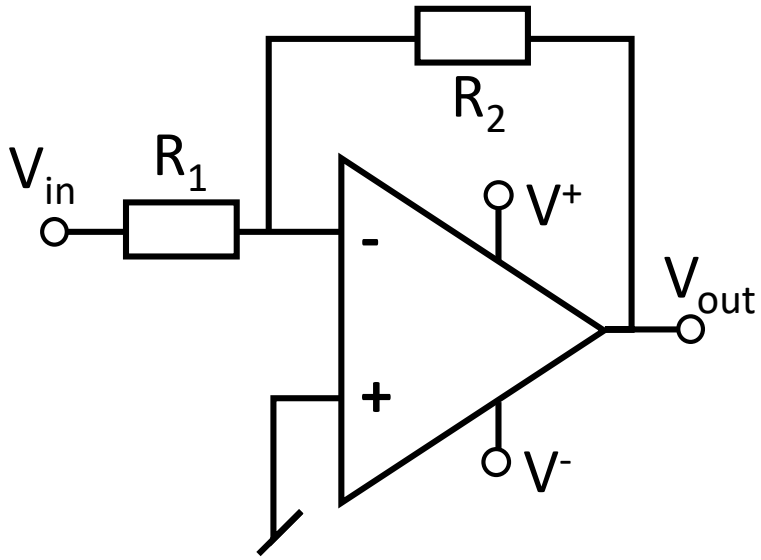
Inverting Amplifier



$$A = -\frac{R_2}{R_1}$$

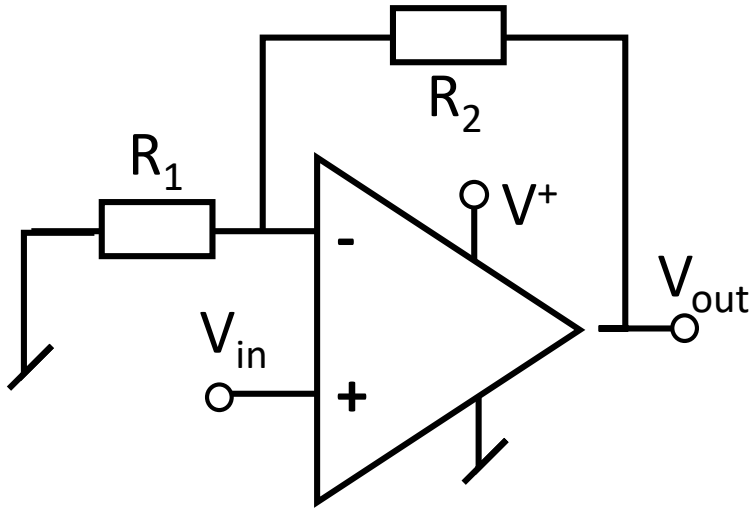
Amplifiers

Inverting Amplifier



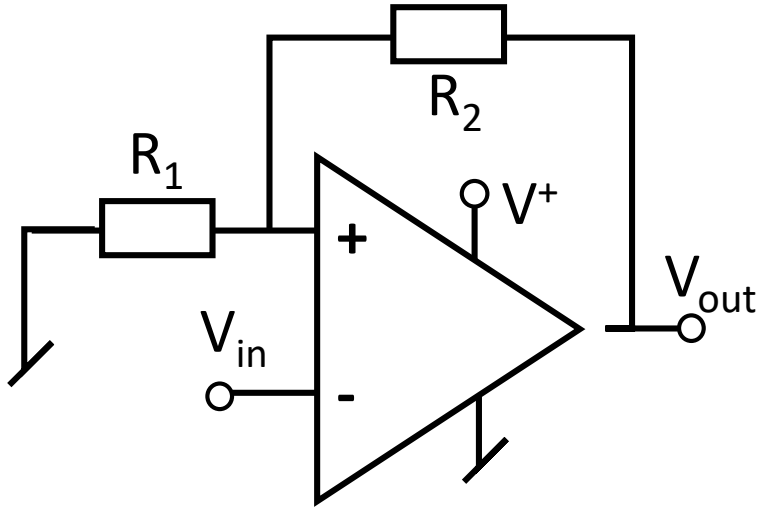
$$A = -\frac{R_2}{R_1}$$

Non-inverting Amplifier



$$A = 1 + \frac{R_2}{R_1}$$

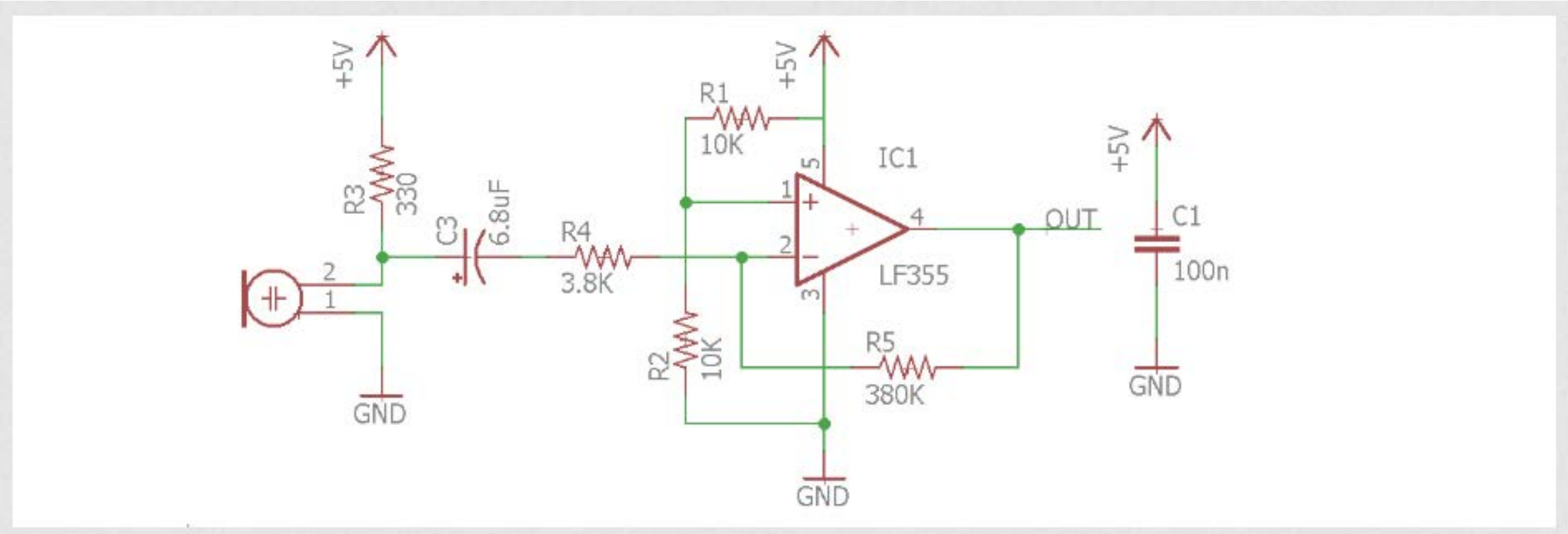
Schmitt Trigger



- What if you're out of analog pins?
- What if you have a signal with a negative component?

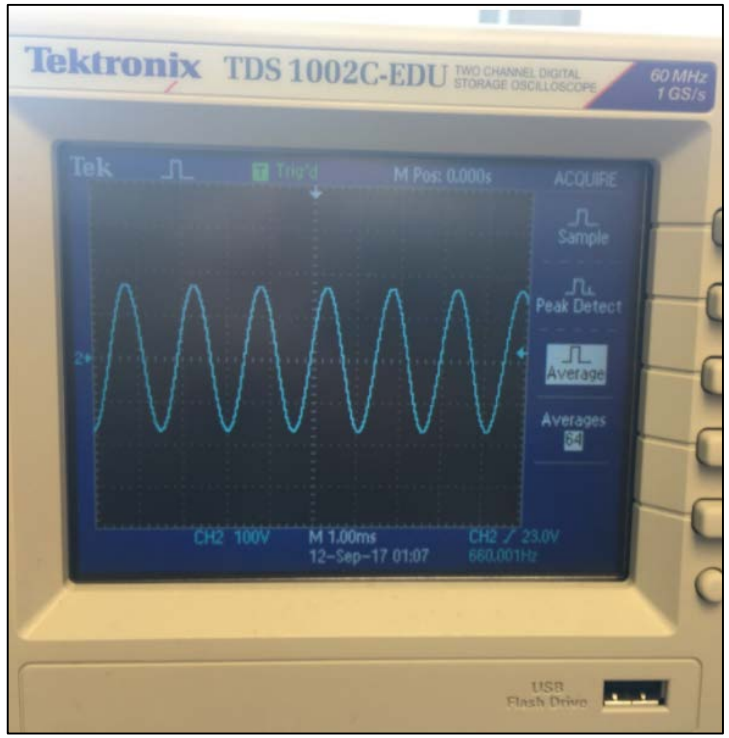
Amplifiers

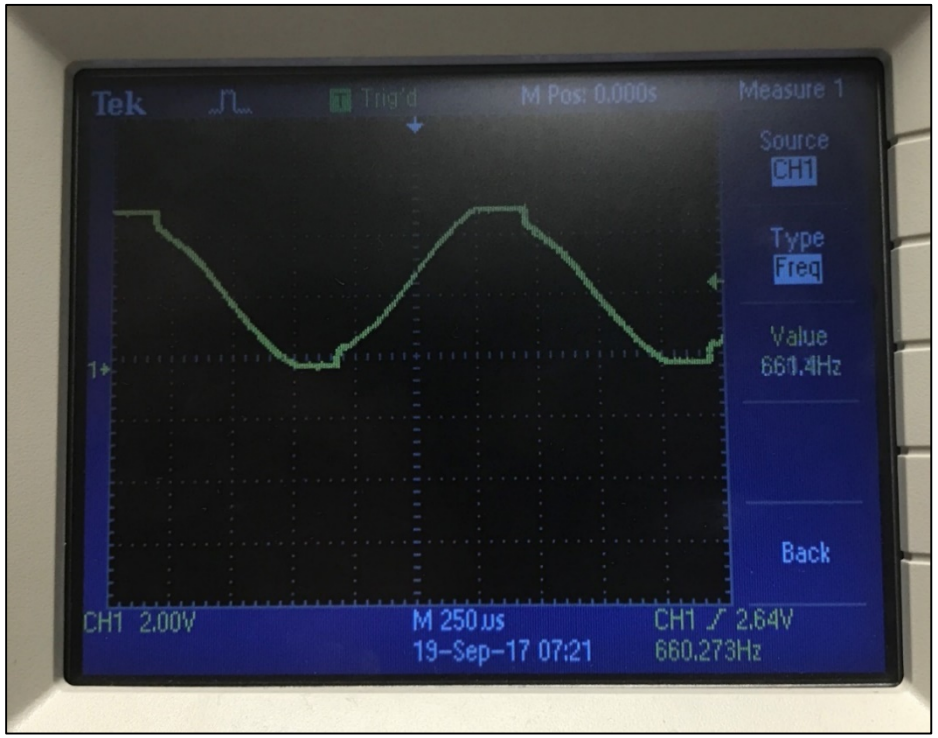
Team Alpha's solution from 2017:

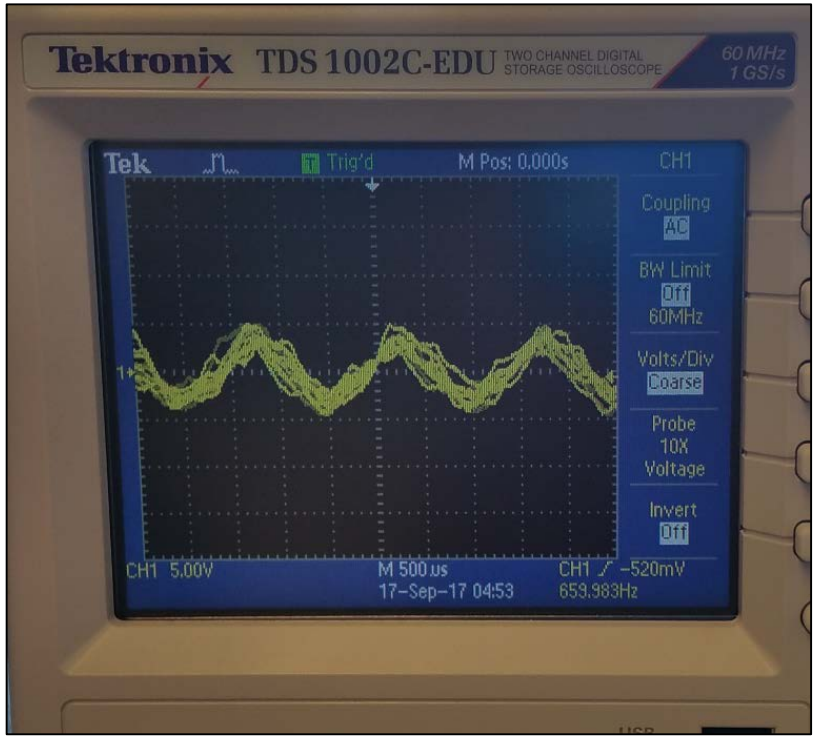


Amplifiers

- *What does a sine wave look like after an amplifier?*



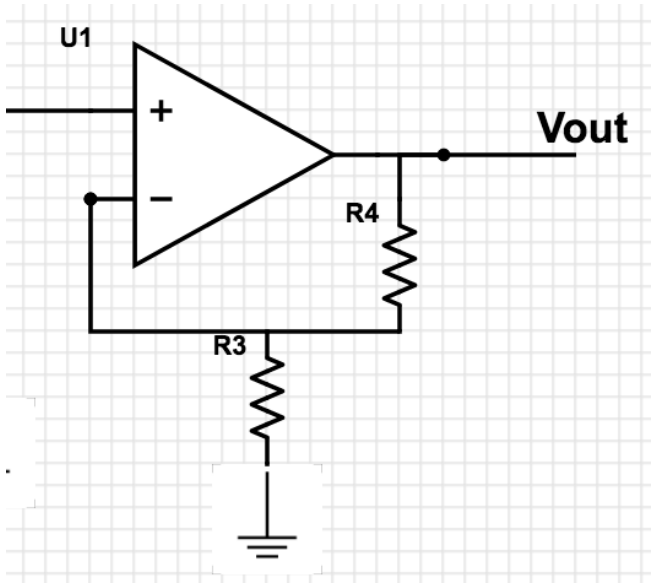




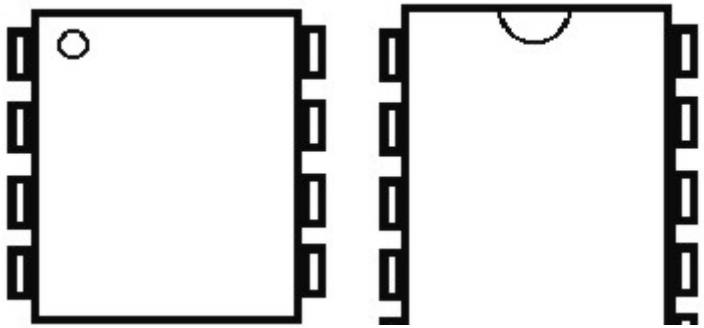
Could you send this signal into the Arduino ADC?

Amplifiers - Sanity Check

- *What if there is no output?*



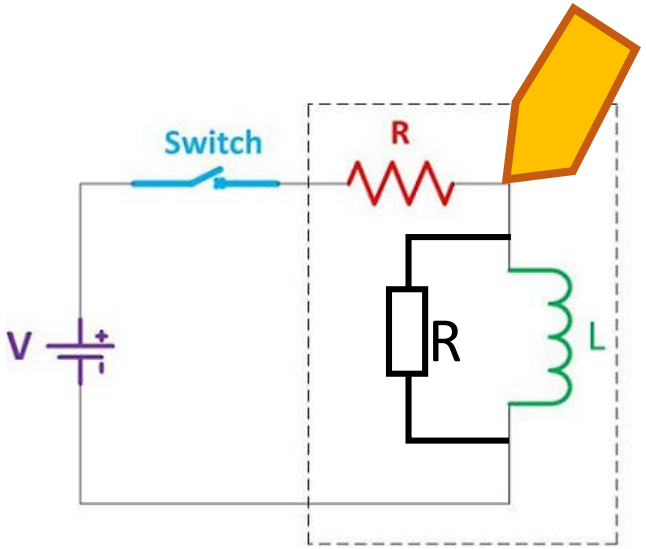
- Is it powered / hooked up right?
- Is it saturated?
  - Check the DC value of your input signal
  - Recalculate the amplification
  - Check if the scope is set to AC

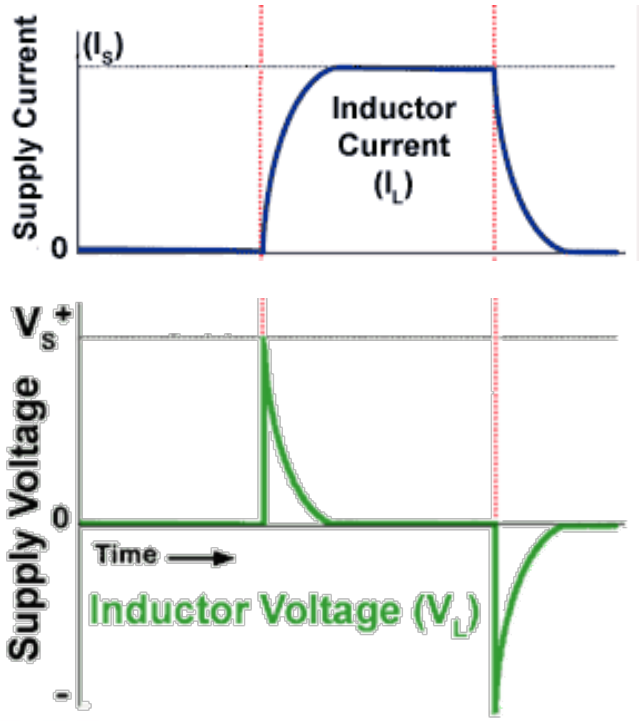


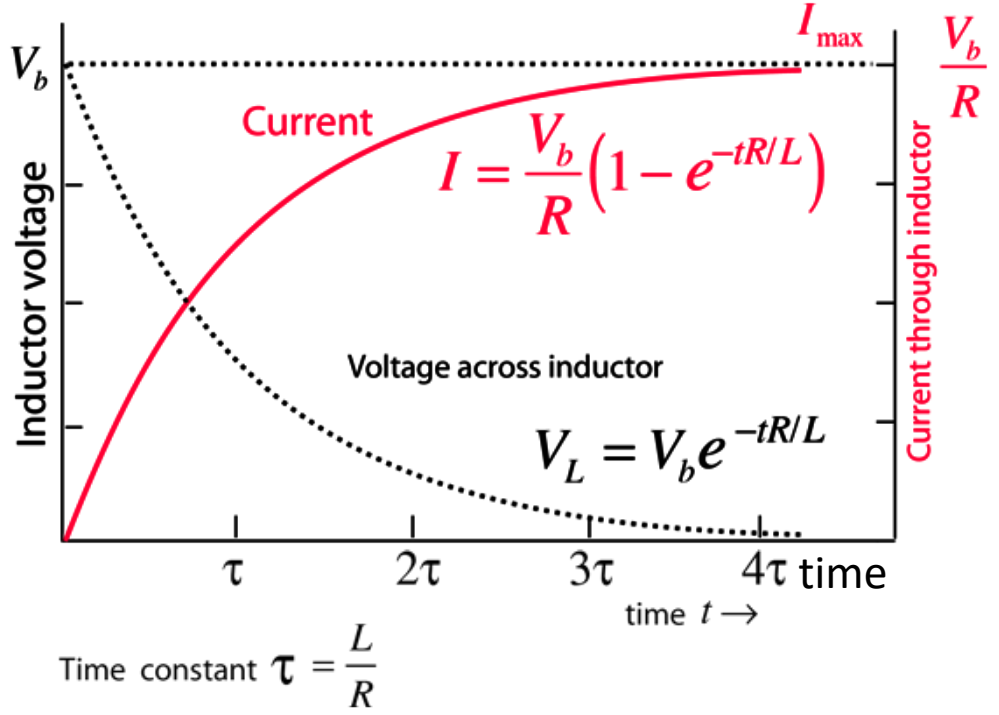
- *Do lots of unit checks!!*

Analog Filters

- *What are the key components in a passive analog filter?*
  - Inductors and capacitors
  - *What happens to the current when the switch closes and opens?*
  - *Is this a high pass or low pass filter?*

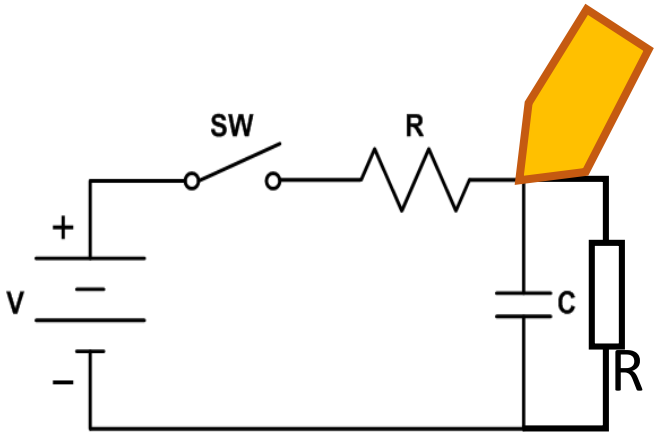


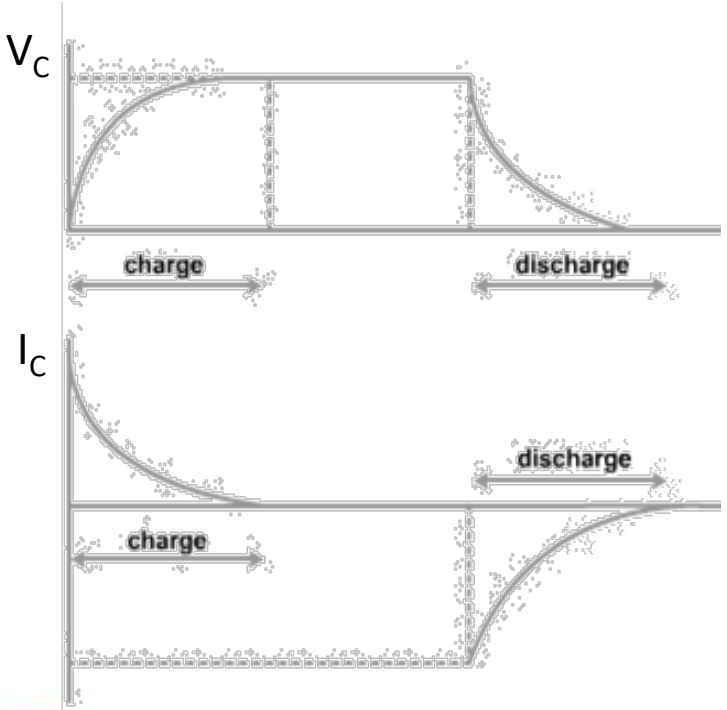


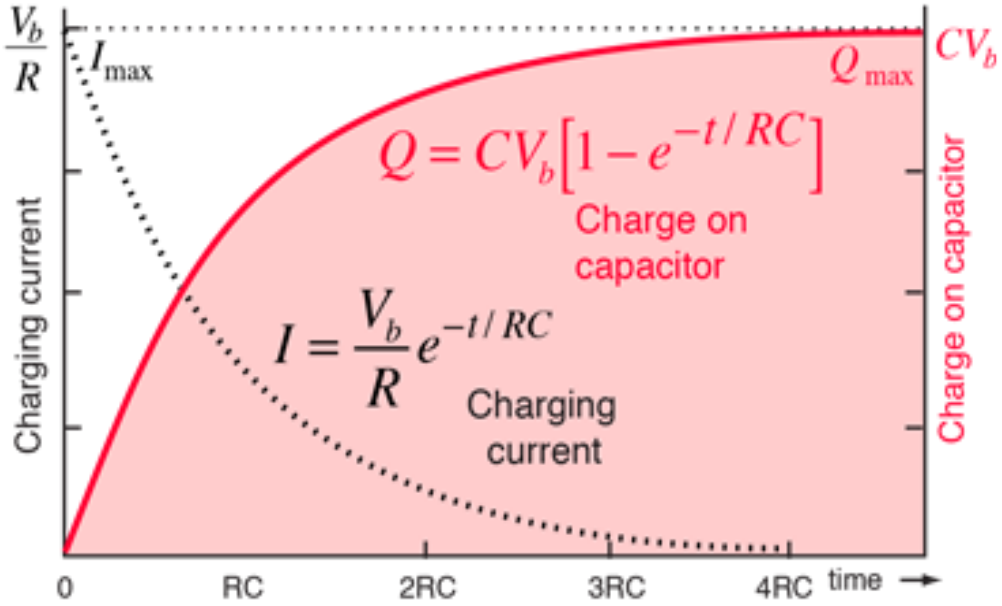


Analog Filters

- *What are the key components in a passive analog filter?*
  - Inductors and capacitors
  - *What happens to the current when the switch closes and opens?*
  - *Is this a high pass or low pass filter?*
  - *What if you wanted a high pass filter?*



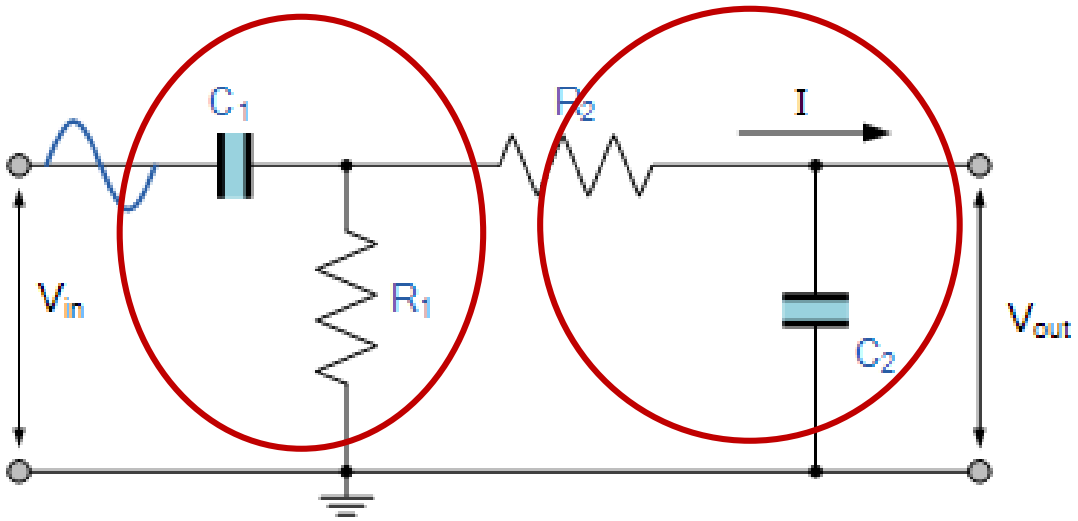




Analog Filters

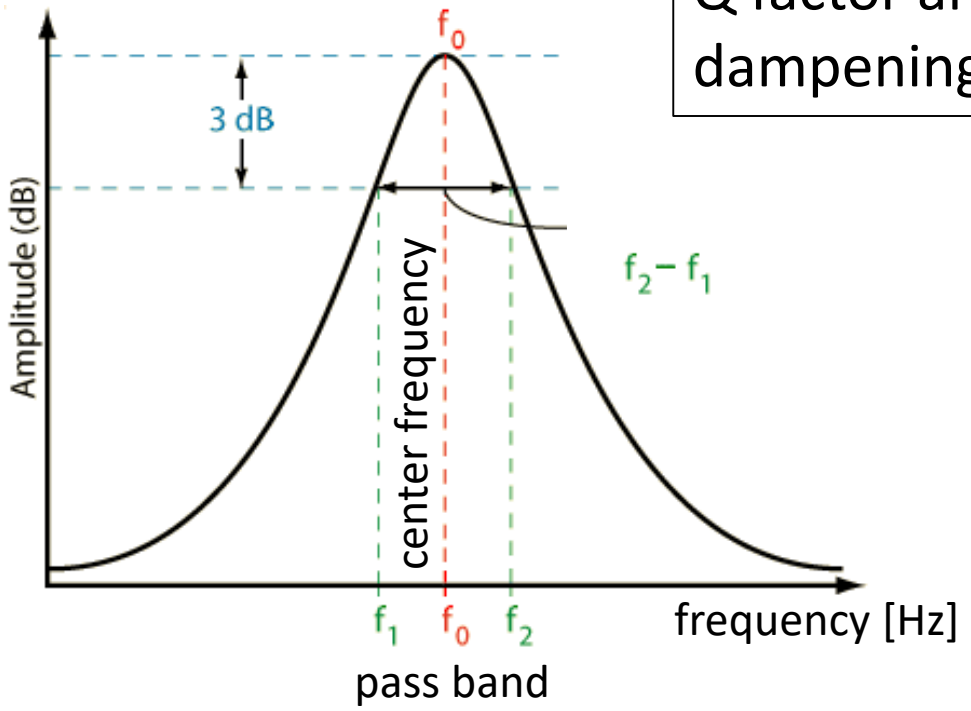
- *What are the key components in a passive analog filter?*
  - Inductors and capacitors
- *What if I want a passive band pass filter?*

Aim:
A filter with a high Q factor and a low dampening



1st stage cuts off low frequencies

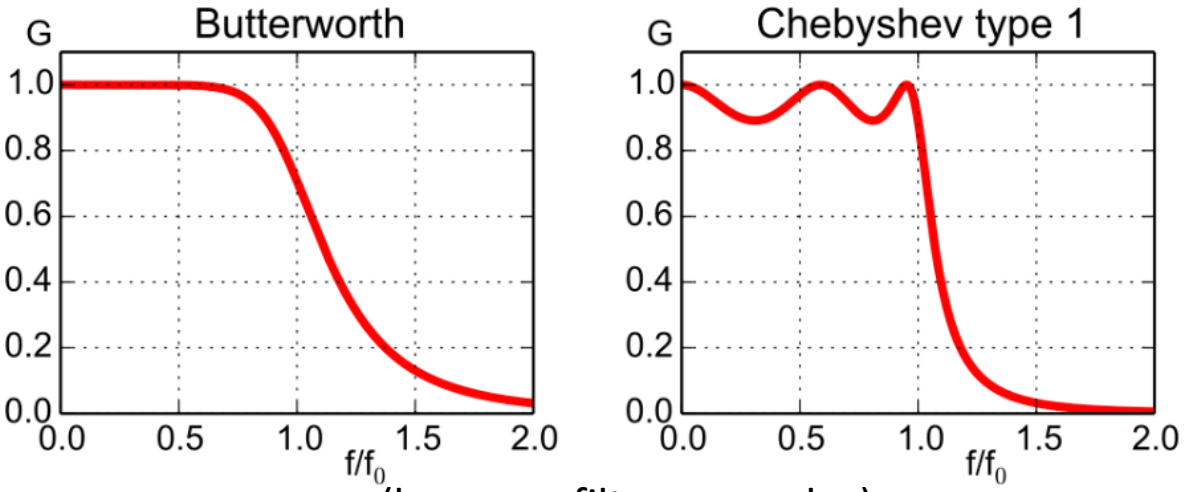
2nd stage cuts off high frequencies



pass band

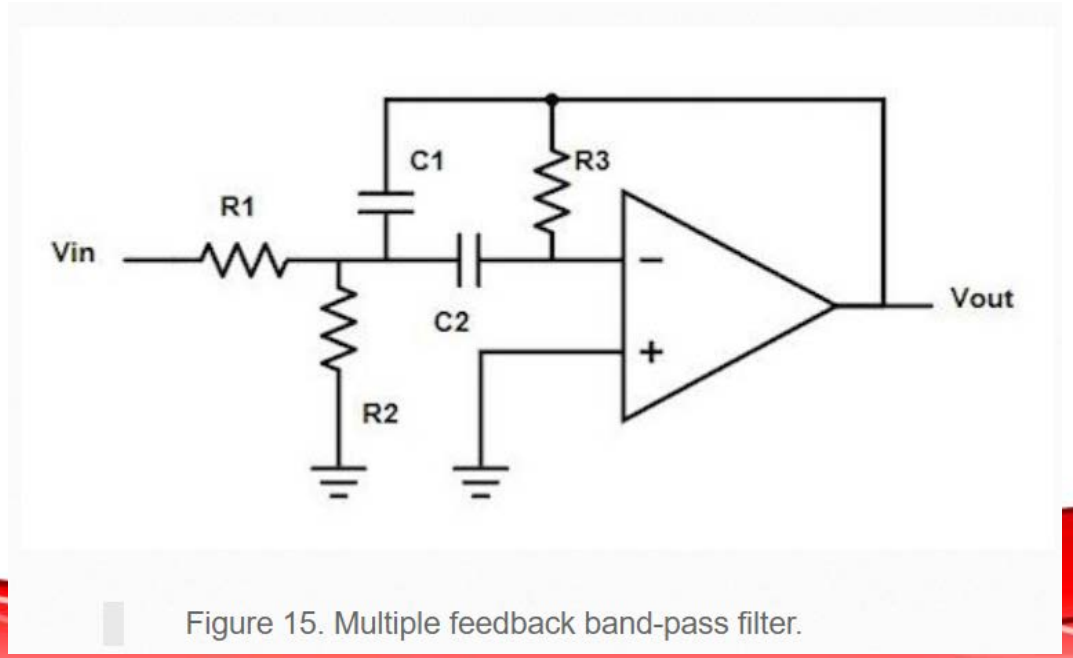
Active Filters

- There are lots of different active filters
  - Chebyshev, Butterworth, Sallen-Key, etc…
- Analog Devices filter wizard
  - http://www.analog.com/designtools/en/filterwizard/
- LT Spice
  - http://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html



(low pass filter examples)

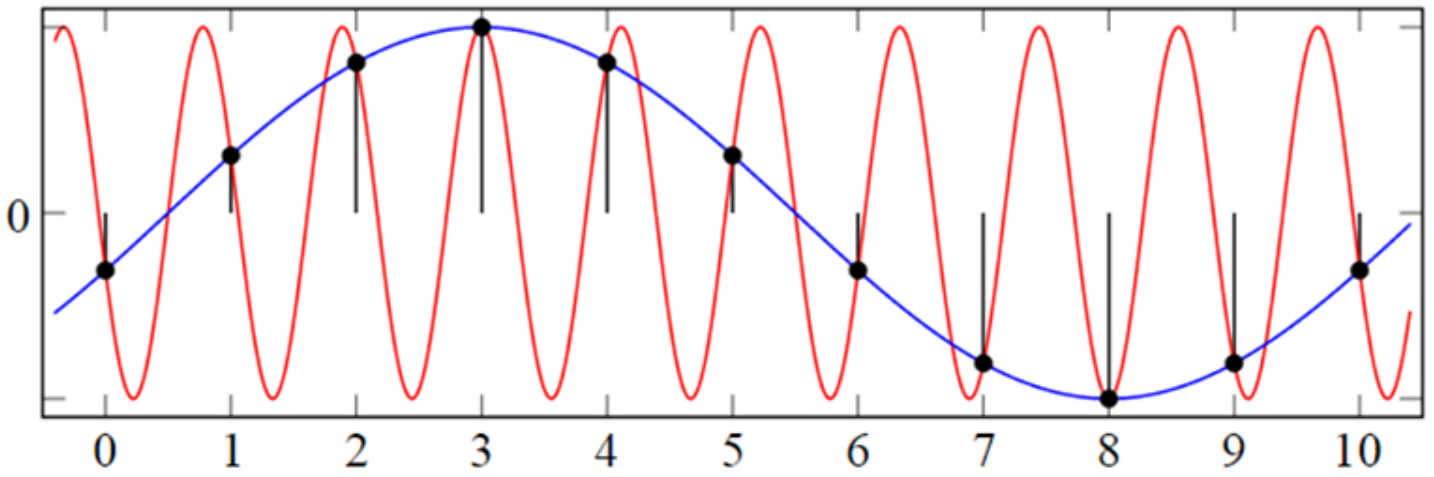
Example from team 15 last year (Chebyshev):



Digital Filters

- *Should you pick a digital over an analog filter?*
  - Analog filters
    - Bulky and require accurate (expensive) components
    - Low relative Q-values
  - Digital filters
    - High relative Q-values and are versatile
    - *BUT they require an ADC (discretization), memory, and processing time*

Discretized Signals and Aliasing



Nyquist Theorem

- The sampling rate must be at least twice the highest frequency component of the signal:

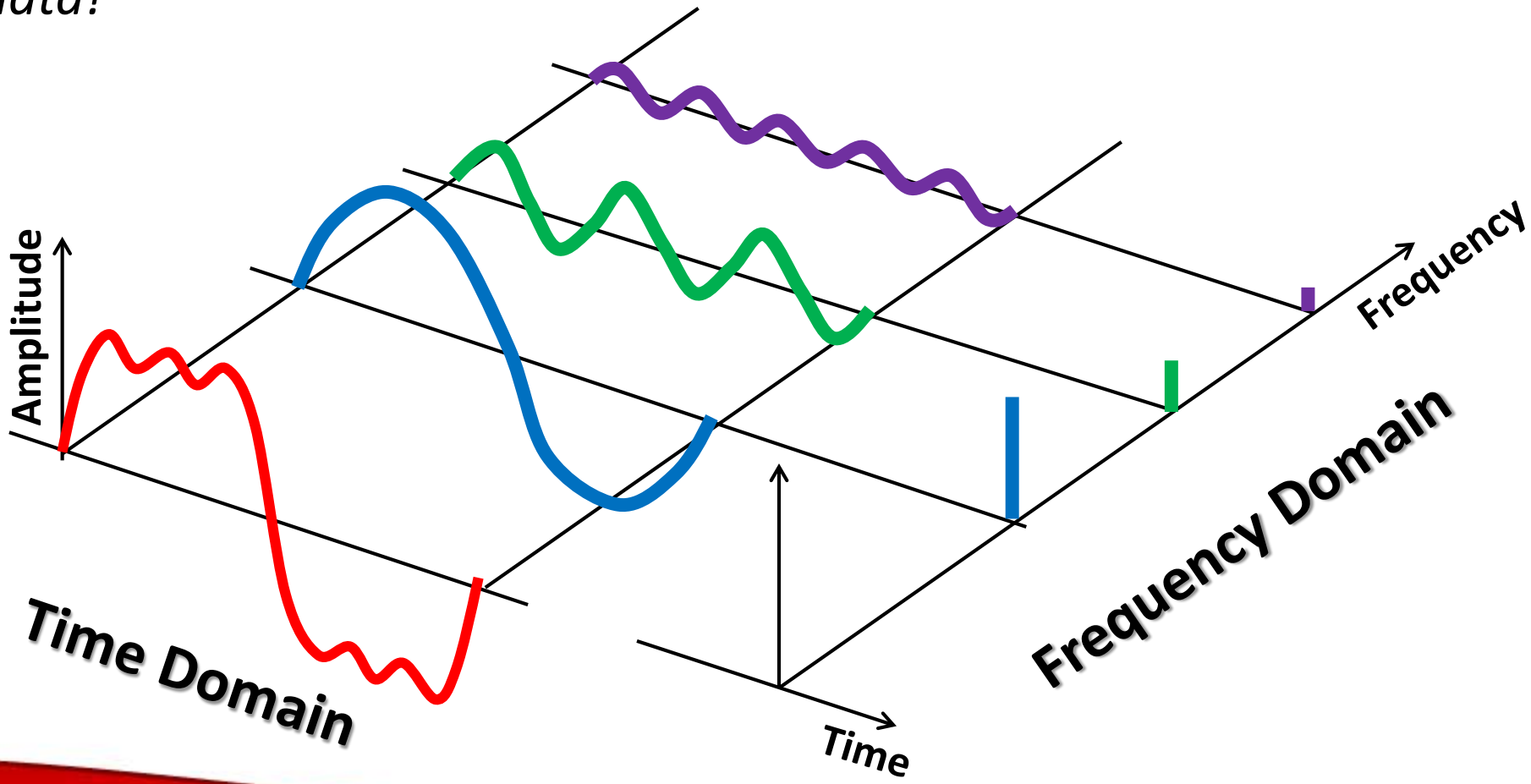
$$f_{sample} = 2 \cdot f_{max}$$

- *How fast do you have to sample you audio signal?*
  - 10kHz music -> 20kHz sampling rate
- *Why not just measure as fast as you can?*
  - Memory becomes an issue
- *It could still make sense to add an analog filter before discretization!*

Fourier Transforms

Modified from: http://hometheaterhifi.com/technical/technical-reviews/up-sampling-aliasing-filtering-ringing-a-clarification-of-terminology/

Arguably the most important set of algorithm for analysis and manipulation of discrete data!

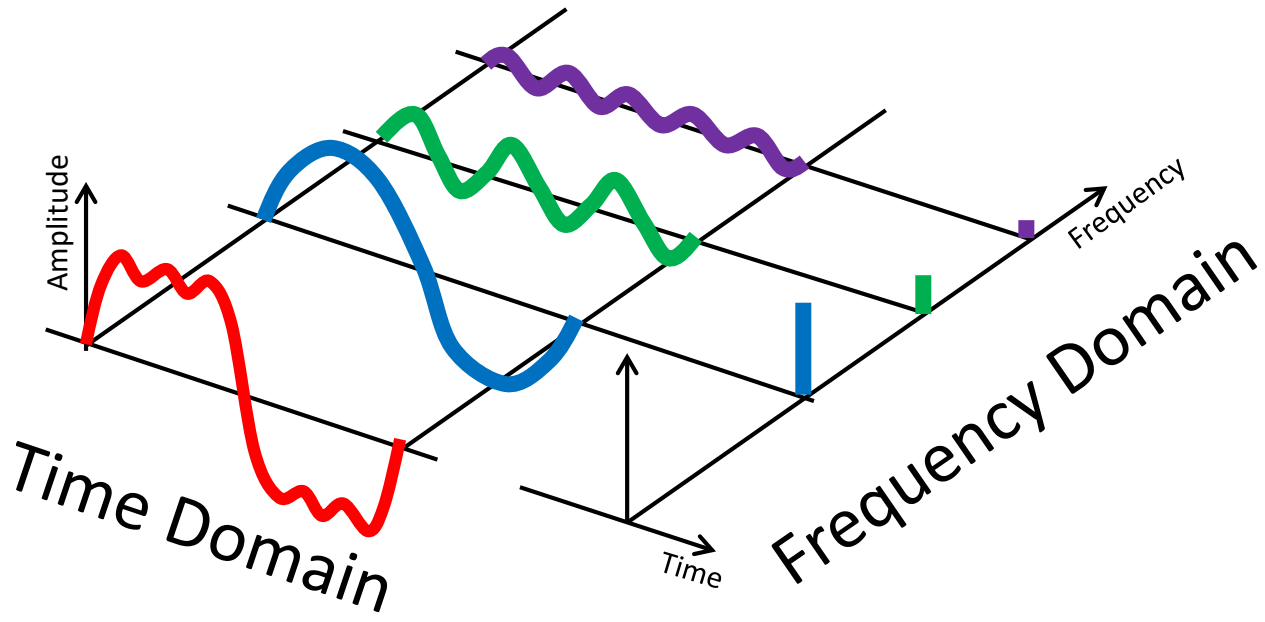


Fast Fourier Transforms

(A version of Discrete Fourier Transforms)

frequency = The sum of contributions to this frequency from each point in time

$$F(x) = \sum_{n=0}^{N-1} f(n) \cdot e^{-i2\pi xn/N}$$



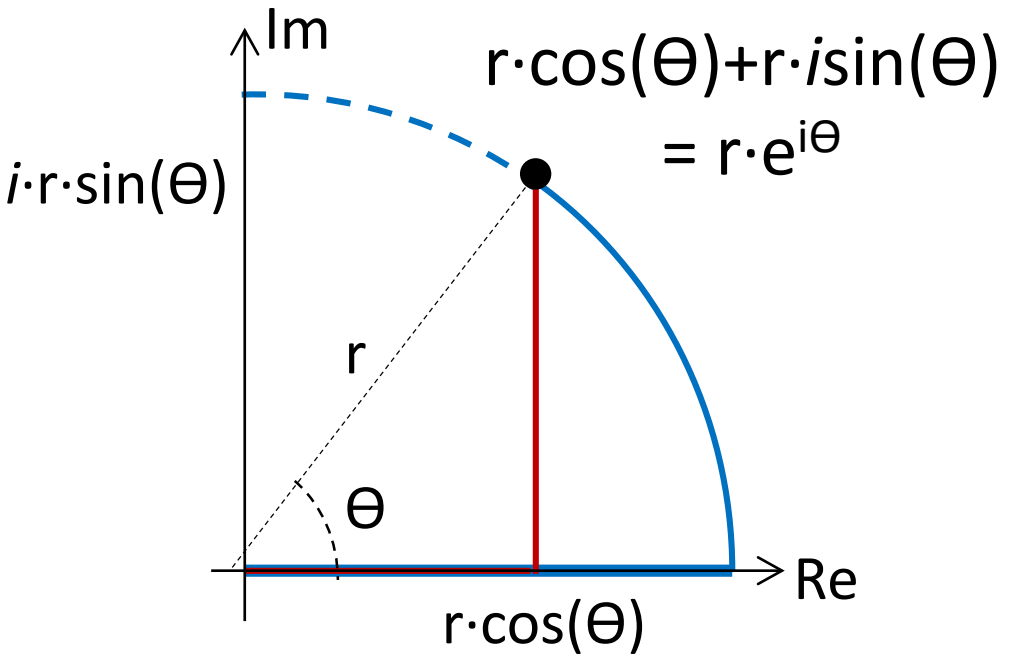
$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) \cdot e^{i2\pi kn/N}$$

Time point = The sum of contributions to this time point from each frequency

Describing a signal in terms of cycles

$$F(x) = \sum_{n=0}^{N-1} f(n) \cdot e^{-i2\pi xn/N}$$

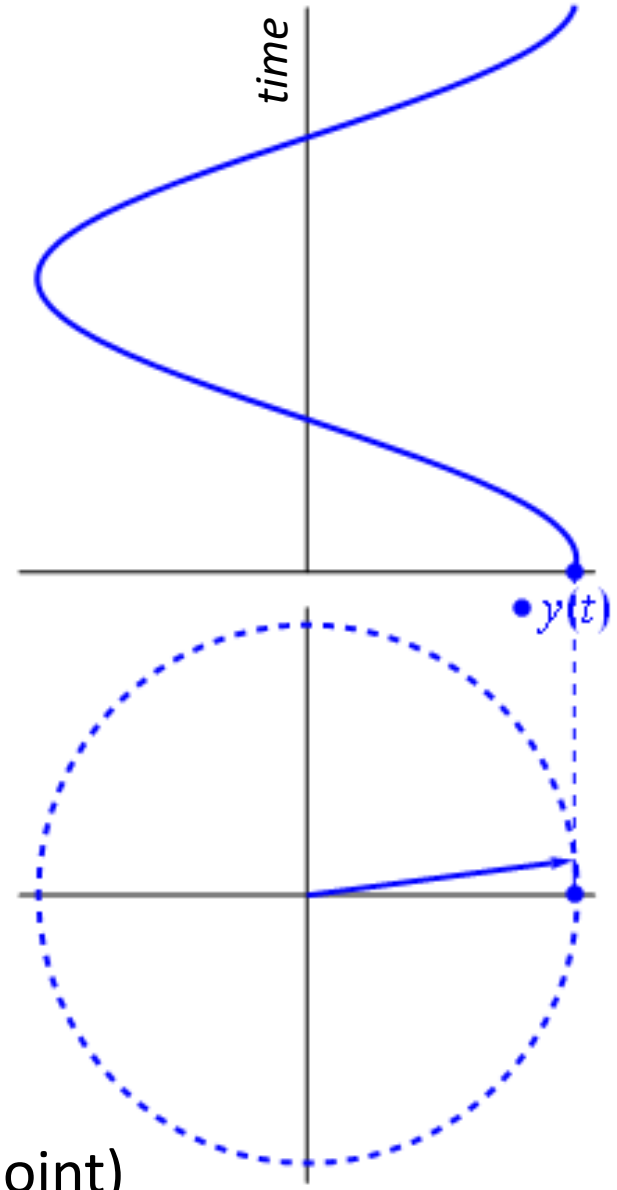
$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) \cdot e^{i2\pi kn/N}$$



Describing a signal in terms of cycles

$$F(x) = \sum_{n=0}^{N-1} f(n) \cdot e^{-i2\pi xn/N}$$

$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x) \cdot e^{i2\pi kn/N}$$



- Frequency (speed)
- Amplitude (radius)
- Phase Angle (starting point)

Describing a signal in terms of cycles

$$F(x) = \sum_{n=0}^{N-1} f(n) \cdot e^{-i2\pi xn/N}$$

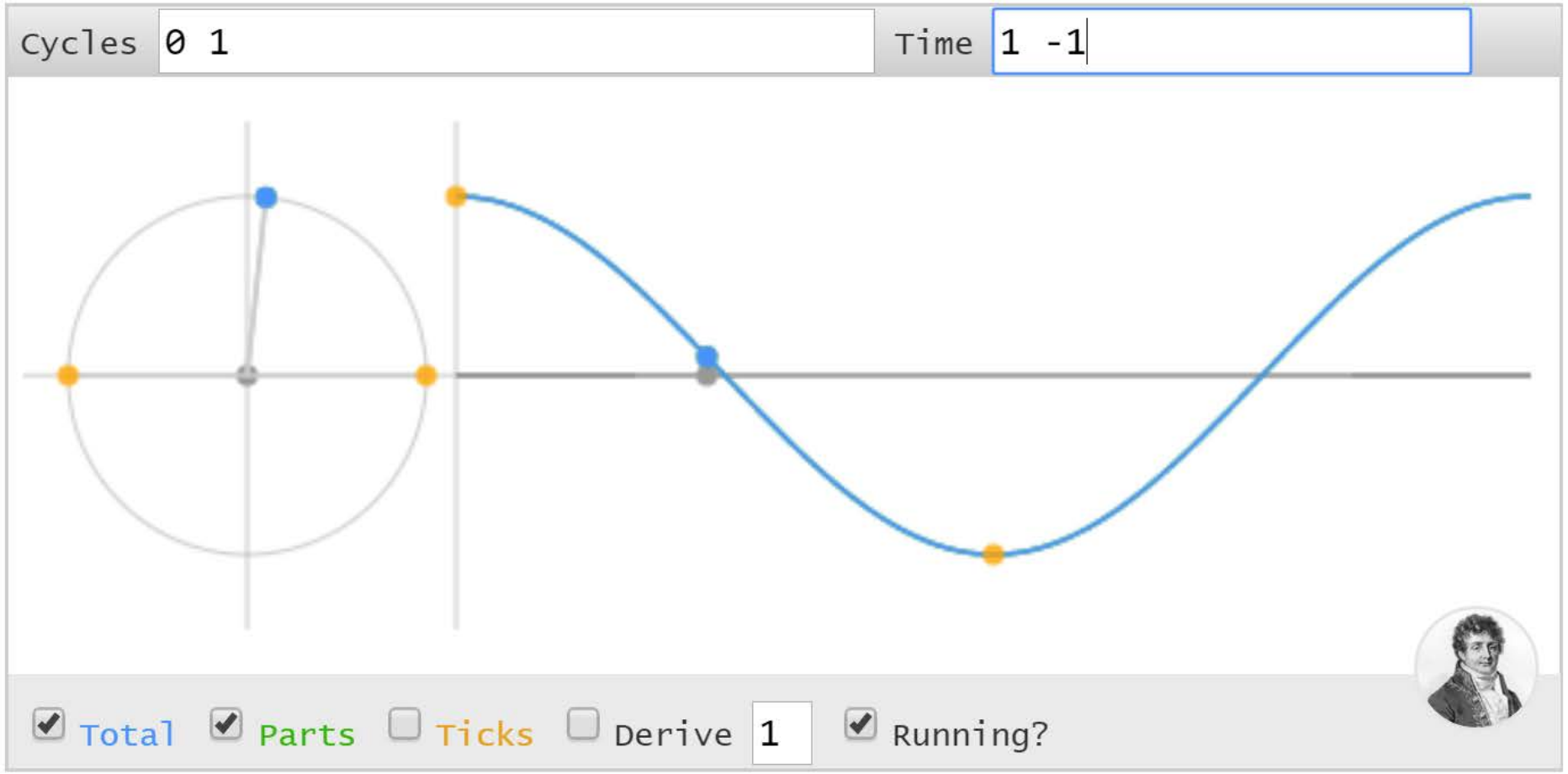
To find the energy at a particular frequency, spin your signal around a circle at that frequency, and average a bunch of points along that path.

quote Stuart Riffle, Blogaday

$$f(n) = \frac{1}{N} \sum_{n=0}^{N-1} F(x_k) \cdot e^{i2\pi kn/N}$$

Describing a signal in terms of cycles

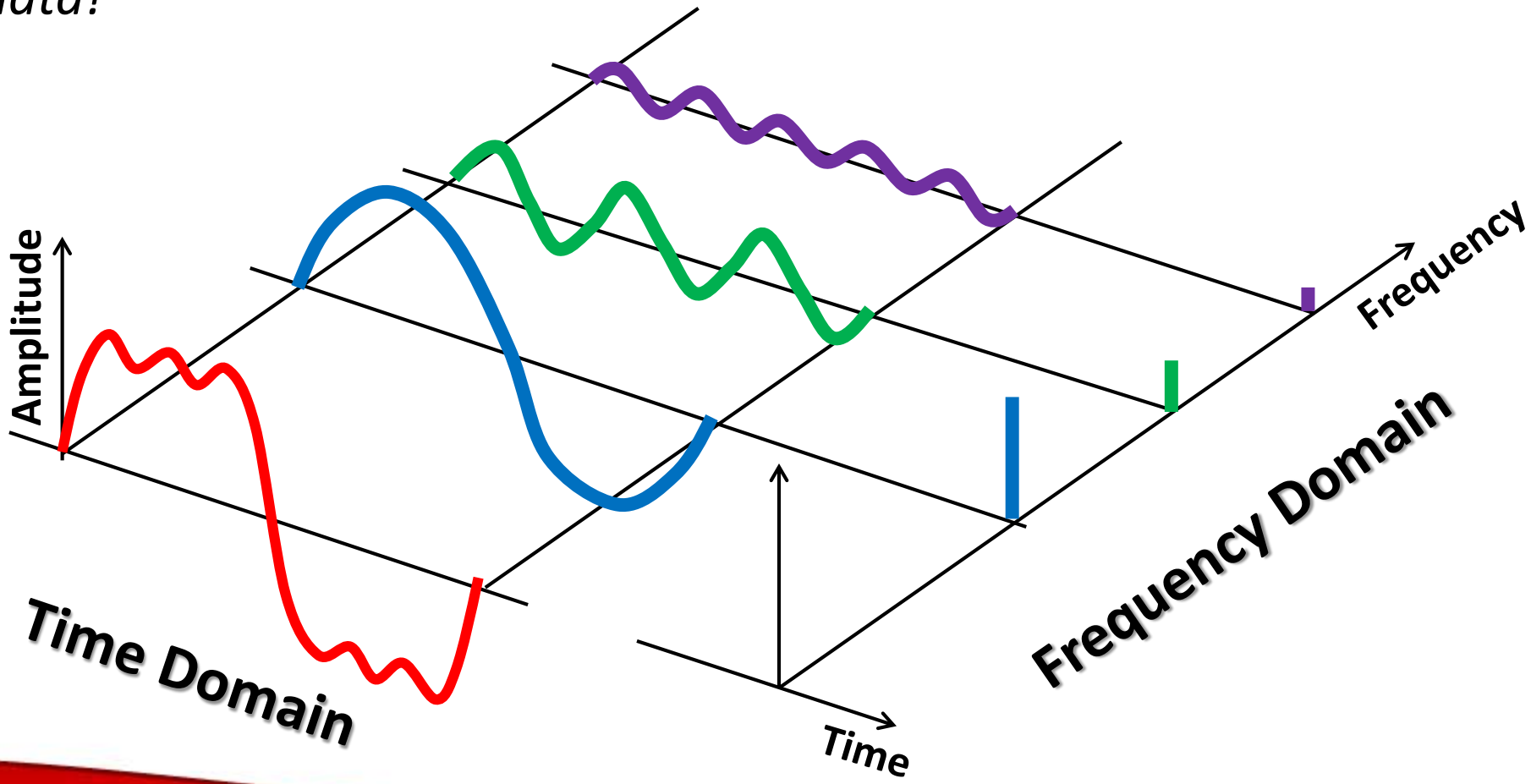
https://betterexplained.com/articles/an-interactive-guide-to-the-fourier-transform/



Fourier Transforms

Modified from: http://hometheaterhifi.com/technical/technical-reviews/up-sampling-aliasing-filtering-ringing-a-clarification-of-terminology/

Arguably the most important set of algorithm for analysis and manipulation of discrete data!



Open Music Labs FFT Library Functions

Courtesy of Vaidehi Garg

`fft_run()`

- Main FFT functional call
- No input variables, no return variables
- Assumes ordered data is available in memory
- Data is stored in array of size N, called fft_input[]
- The array contains 2 16-bit values per FFT datapoint
- Even positions are for real values, odd for imaginary
- Every two positions corresponds to one bin => N/2 FFT bins

Open Music Labs FFT Library Functions

Courtesy of Vaidehi Garg

Before calling fft_run():

- Decide N based on the number of FFT bins you want.
- Fill fft_input[] with datapoints from the ADC. Since the datapoints are real, put them in the even positions, and fill the odd positions with 0's.

Open Music Labs FFT Library Functions

Courtesy of Vaidehi Garg

After calling fft_run():

- Use one of the provided functions to process the FFT output and obtain the output array.
  - fft_mag_log() returns the output in sequential order of FFT frequency bins!
- Check the value of the relevant bin (which bin # would you need to check for a 660Hz or 12kHz frequency?)

(do the prelab!)

Go Build Robots!

http://www.owendavey.com/Build-a-Robot