CSE121: IoT

ADC/DAC

Jose Renau
renau@ucsc.edu

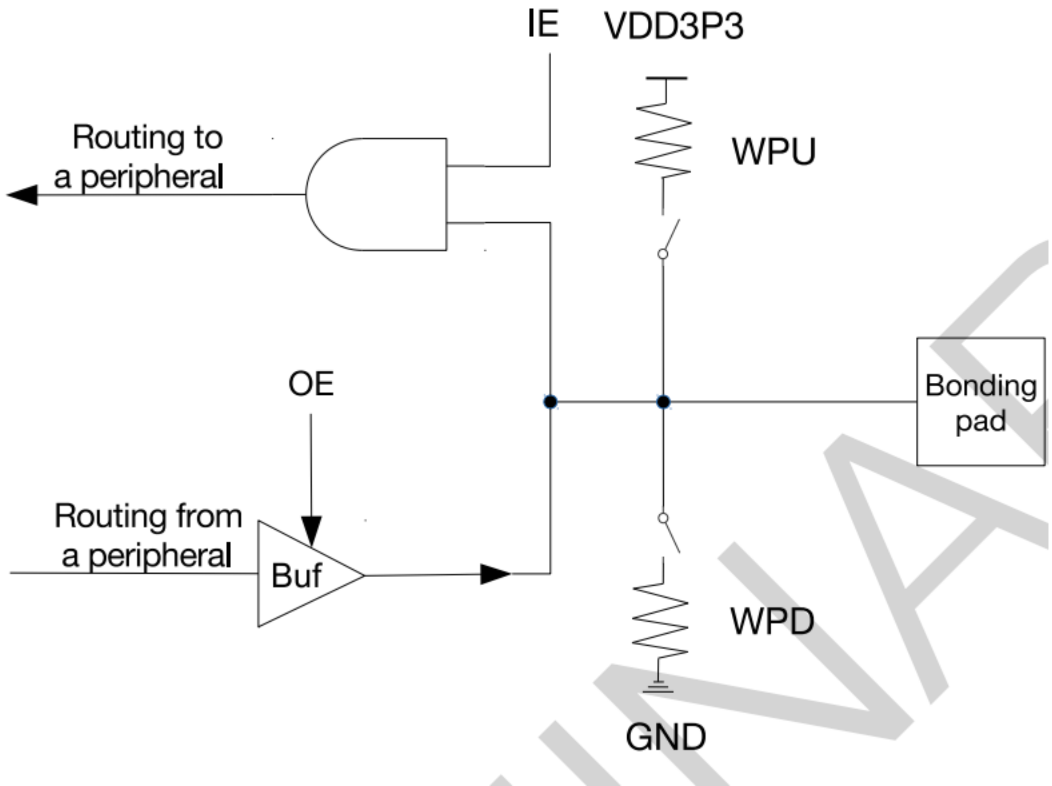
UCSC Baskin School of Engineering
University of California, Santa Cruz

Announcements

- Lab3 due next week
- Quiz next class
- Acknowledgments
  - Heiner Litz
  - Aaron Schulman

Single-Ended GPIO

- GPIO like in ESP32



Single-ended limitations

- Clock frequency limits the amount of information that can be transferred via I/O

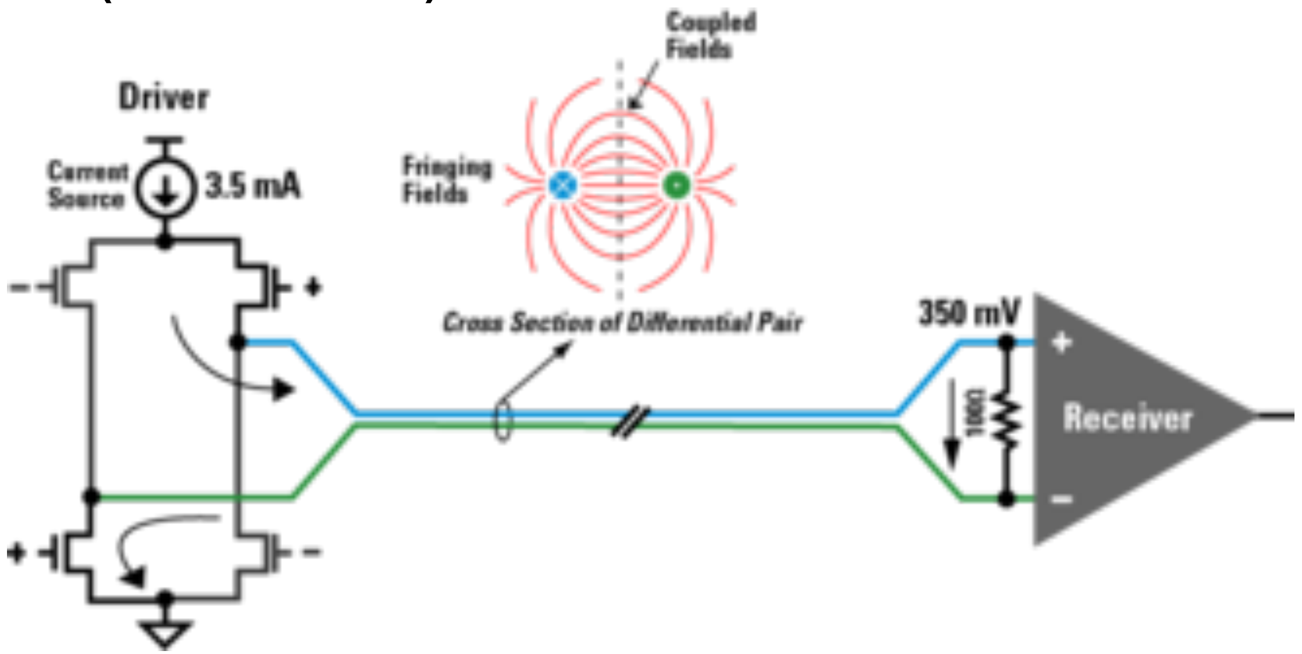
- Maximum frequency is limited (100’s of MHz)
  - Longer PCB traces have higher capacity and impedance
  - Single ended signals operate at 3.3V/5C
  - Need to charge net to full swing

- DC signal generates current flow from one endpoint to the other

- Solution: Differential signaling (LVDS)

LVDS

- Use two pins carrying an inverted voltage level
- No current flow
- Without full swing, just determine if I1 > I2 or I1 < I2
- Much higher frequencies (GHz)
- Longer (impedance-matched) PCB traces
- Supported by many microcontrollers (not PSoC)
- Higher power efficiency
- More Crosstalk immunity



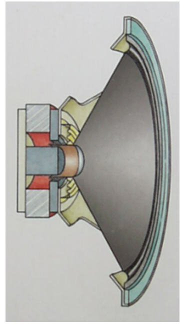
Digital Systems

- Almost all Integrated Circuits are digital systems
  - Robust: easy to distinguish 0 vs. 1 voltage-wise
  - Robust: temperature and variation tolerant
  - Simple: How to you build an electrical circuit to multiply 2 analog voltages? (e.g. 1.7V*0.8V=1.36V)

- BUT: we live in an analog world
  - Audio
  - Video

We live in analog world

- Everything in the physical world is an analog signal
  - Sound, light, temperature, pressure
- Need to convert into electrical signals
  - Transducers: converts one type of energy to another
    - Electro-mechanical, Photonic, Electrical, …
  - Examples
    - Microphone/speaker
    - Thermocouples
    - Accelerometers



ADC/DAC

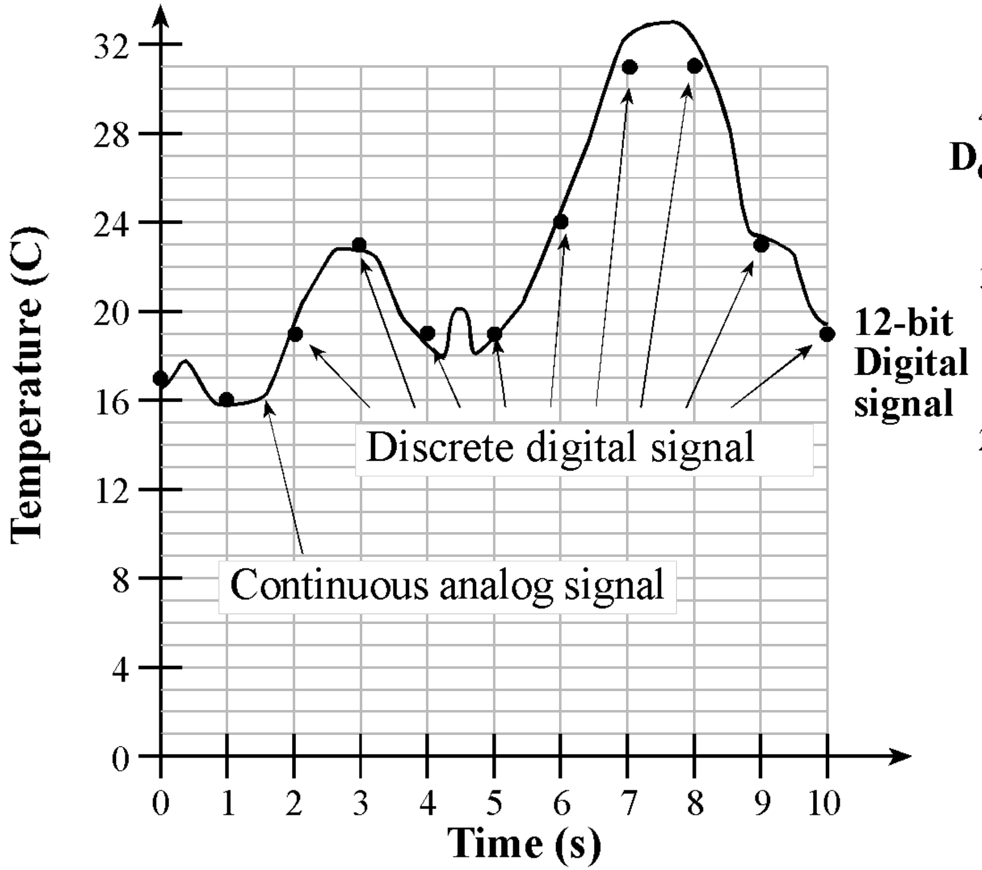
- Analog-to-digital converter (ADC)
  - E.g: Microphone
- Digital-to-analog converter (DAC)
  - E.g: Speaker

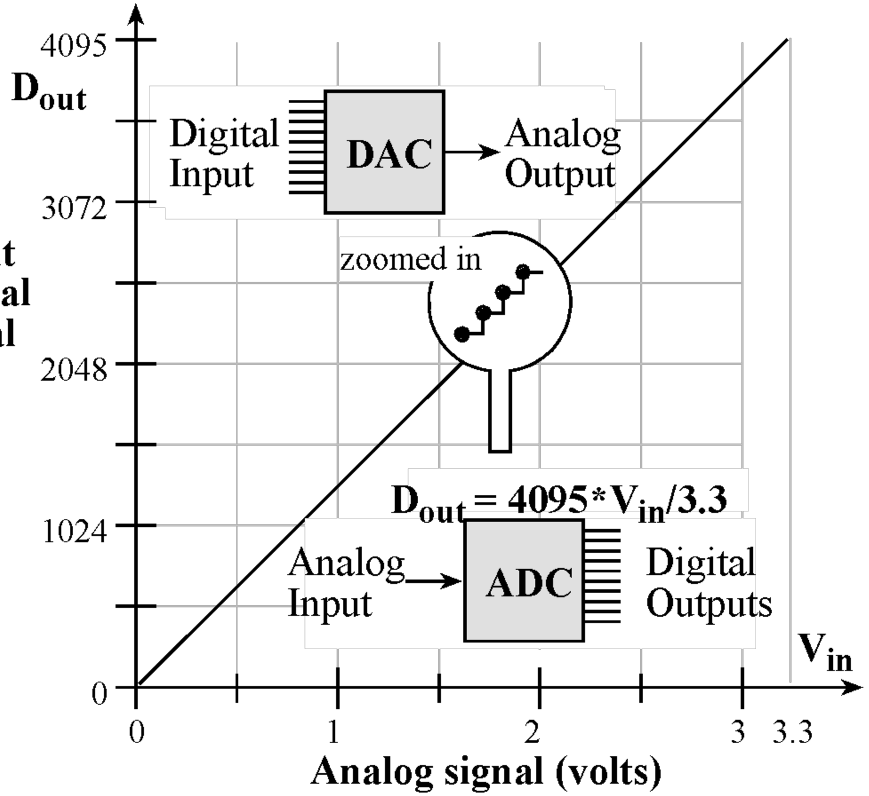
ADC

- Transforms continuous into discrete signal
- Discrete in time (x-axis)
- Described by sampling rate
  - How frequently do we measure?
  - Discrete amplitude (y-axis)
- Sensor produces a continuous range of voltages, e.g. 0V-5V
  - ADC input range needs to match sensor
  - An e.g. 12-bit ADC translates it into 212 discrete values
  - Discrete values uniformly distributed over range

ADC

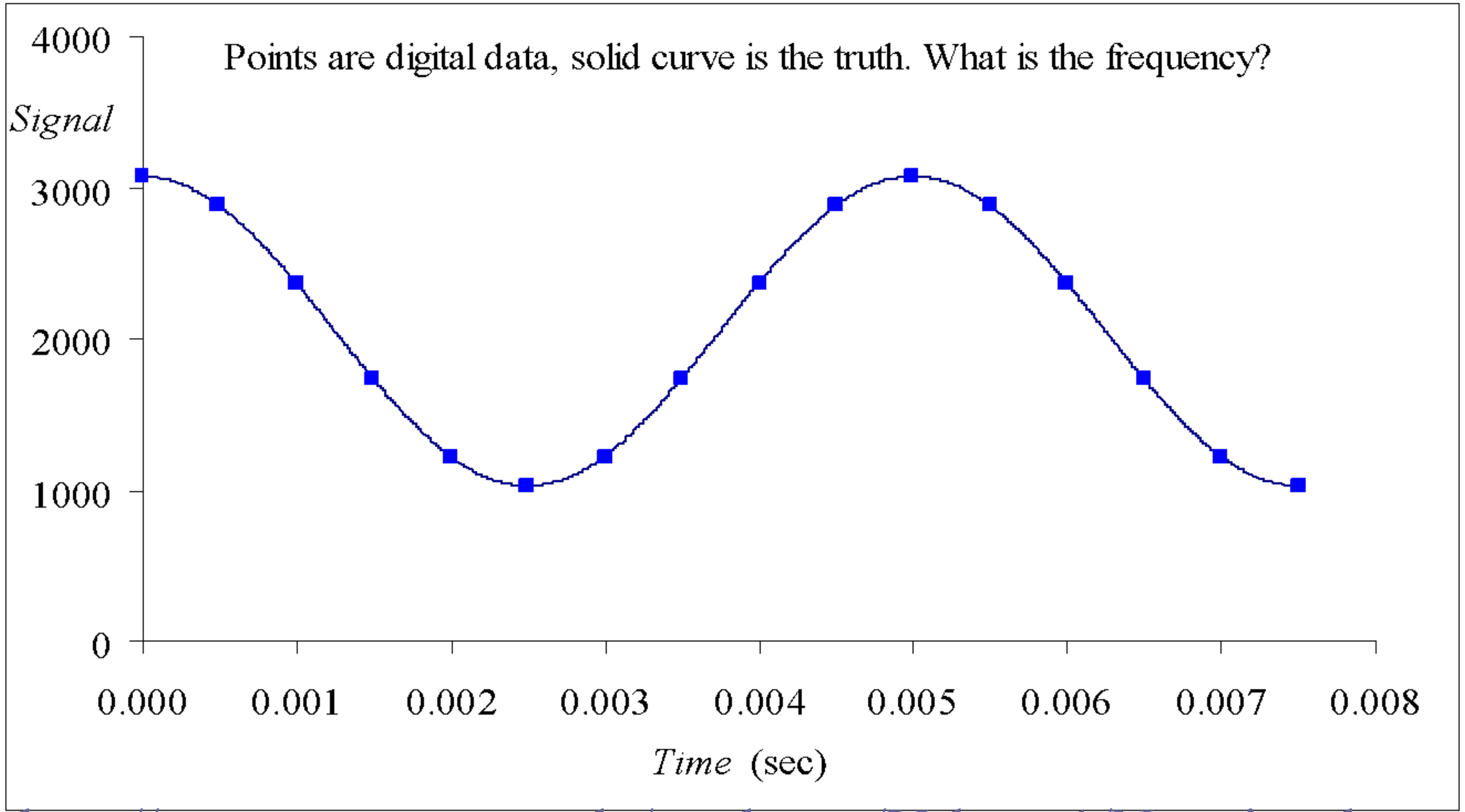
- E.g:
  - 12 levels, 0 to 3.3V





Sampling frequency

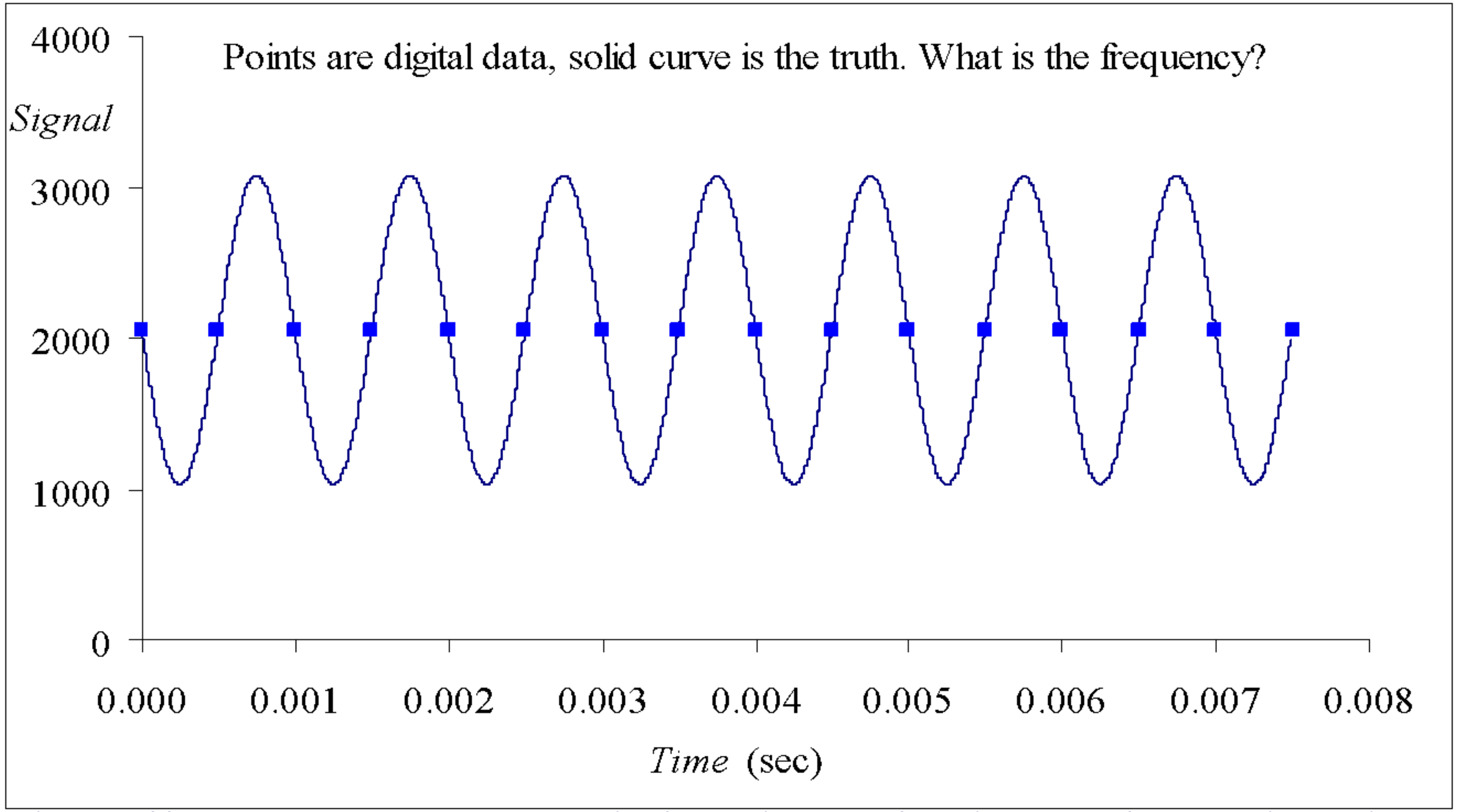
- 200Hz sampled at 2KHz



http://www.ece.utexas.edu/~valvano/Volume1/Nyquist.xls

Sampling frequency

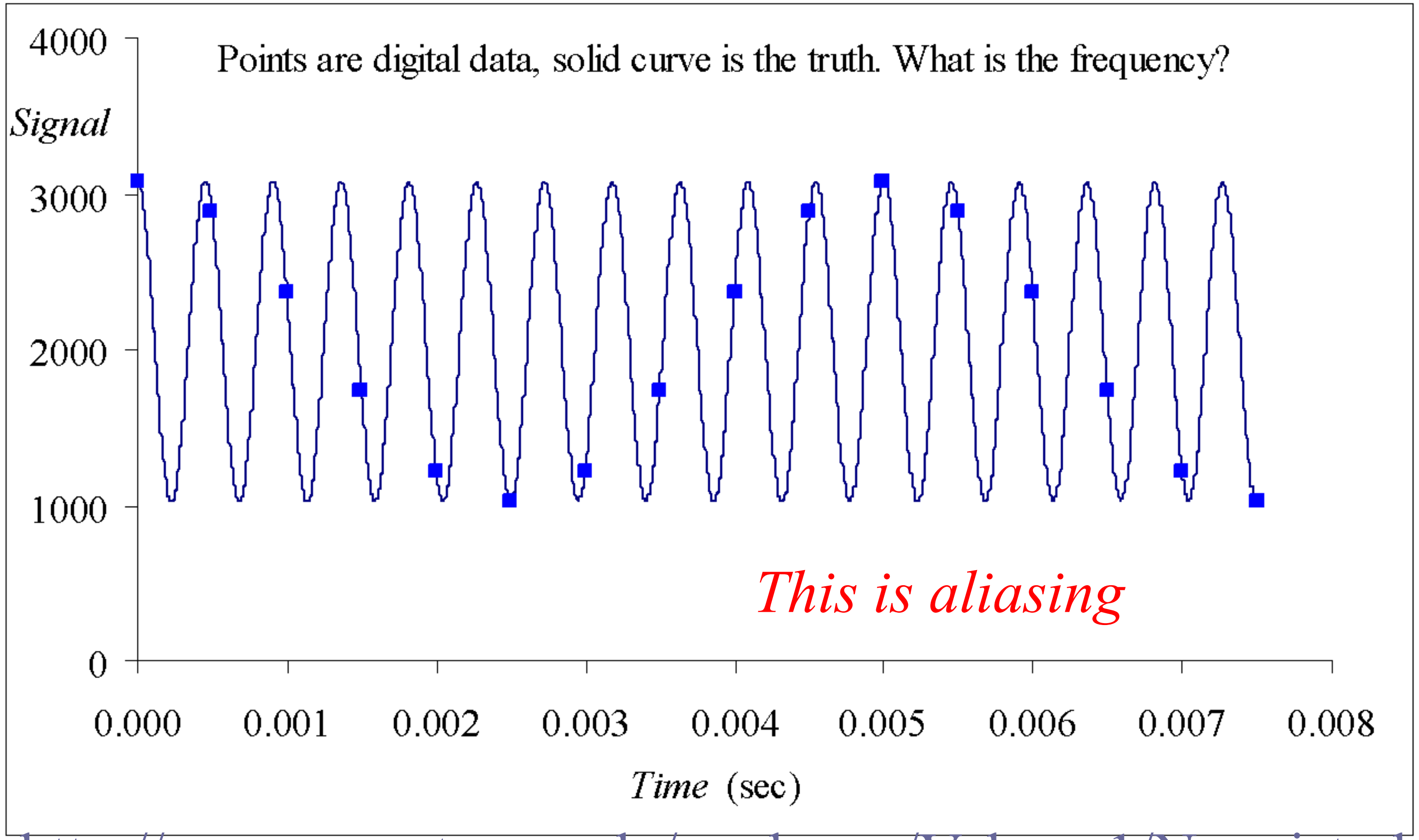
- 1KHz sampled at 2KHz



http://www.ece.utexas.edu/~valvano/Volume1/Nyquist.xls

Sample aliasing “funny effects”

- 2200Hz signal samples at 2000Hz



http://www.ece.utexas.edu/~valvano/Volume1/Nyquist.xls

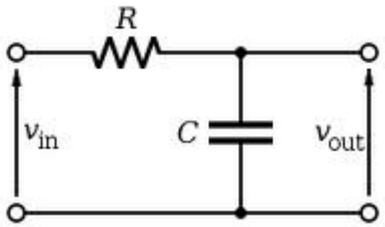
Shannon-Nyquist

- If a function contains no frequencies higher than B hertz, then it can be completely determined from its ordinates at a sequence of points spaced less than 1/(2B) seconds apart.

$$f_{\text{samples}} > 2 f_{\text{max}}$$

What about alias?

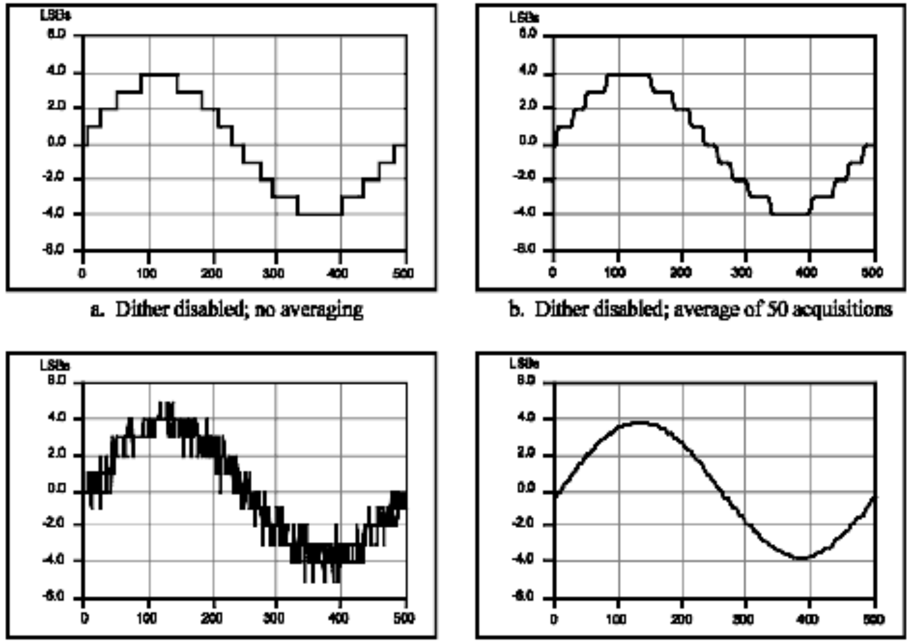
- Use a low-pass filter to remove high frequencies that can alias
  - If you sampling frequency is 1Khz, you should remove frequencies higher than 1KHz to avoid potential alias



  - Most ADC have a built-in filter

ADC Dithering

- Some ADCs have dithering (not ESP32)
  - Oversample with random noise
  - Filter (smooth) data

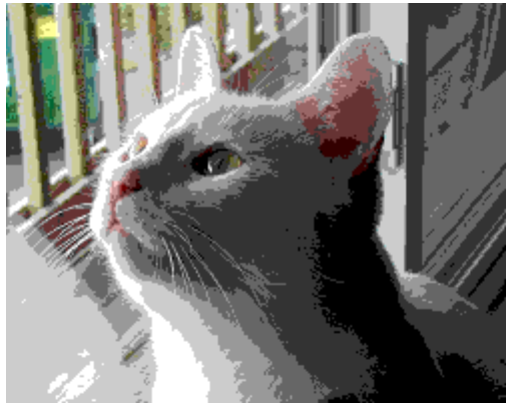


a. Dither disabled; no averaging

b. Dither disabled; average of 50 acquisitions

c. Dither enabled; no averaging

d. Dither enabled; average of 50 acquisitions



Direct Samples

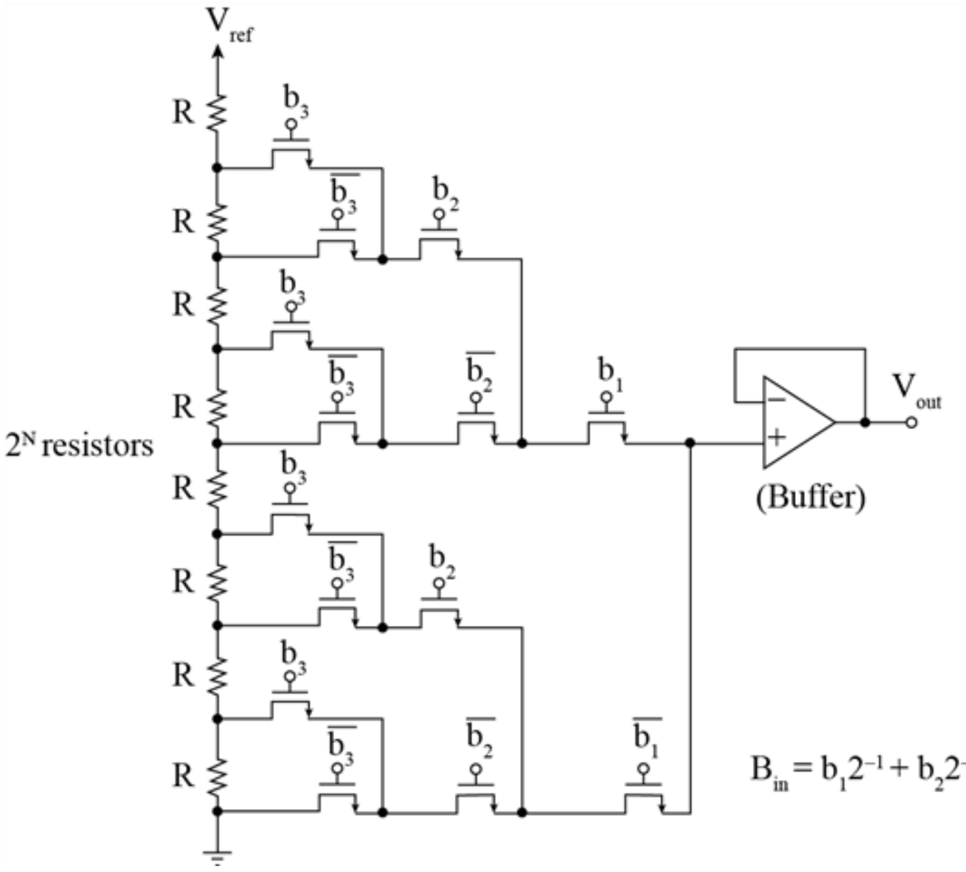
Dithered Samples

Building a DAC/ADC

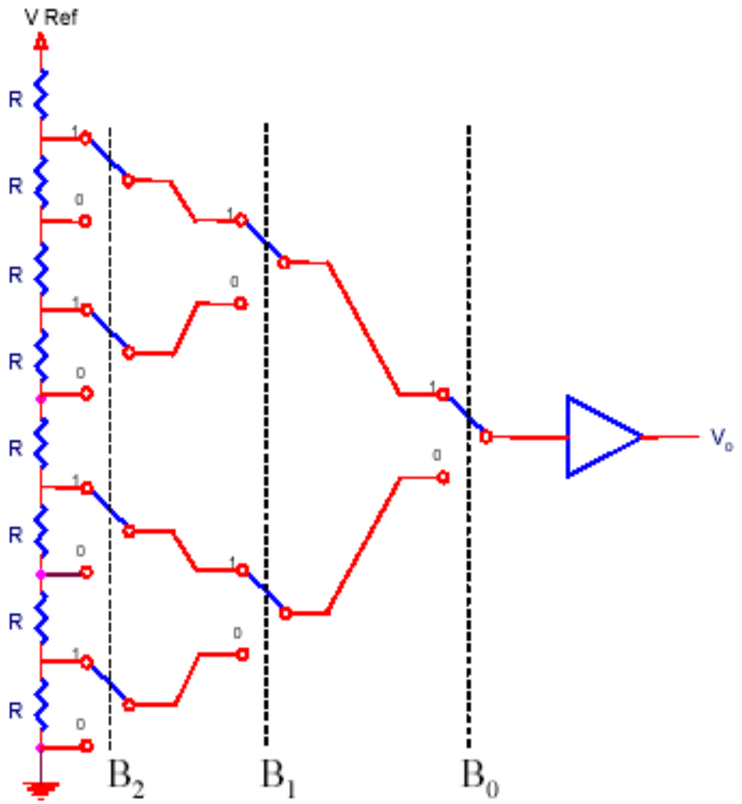
- http://users.ece.utexas.edu/~valvano/Volume1/E-Book/C13_Interactives.htm

Thermometer encoded DAC

- https://www.chegg.com/homework-help/analog-integrated-circuit-design-2nd-edition-chapter-16-solutions-9780470770108
-



$B_{in} = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3}$

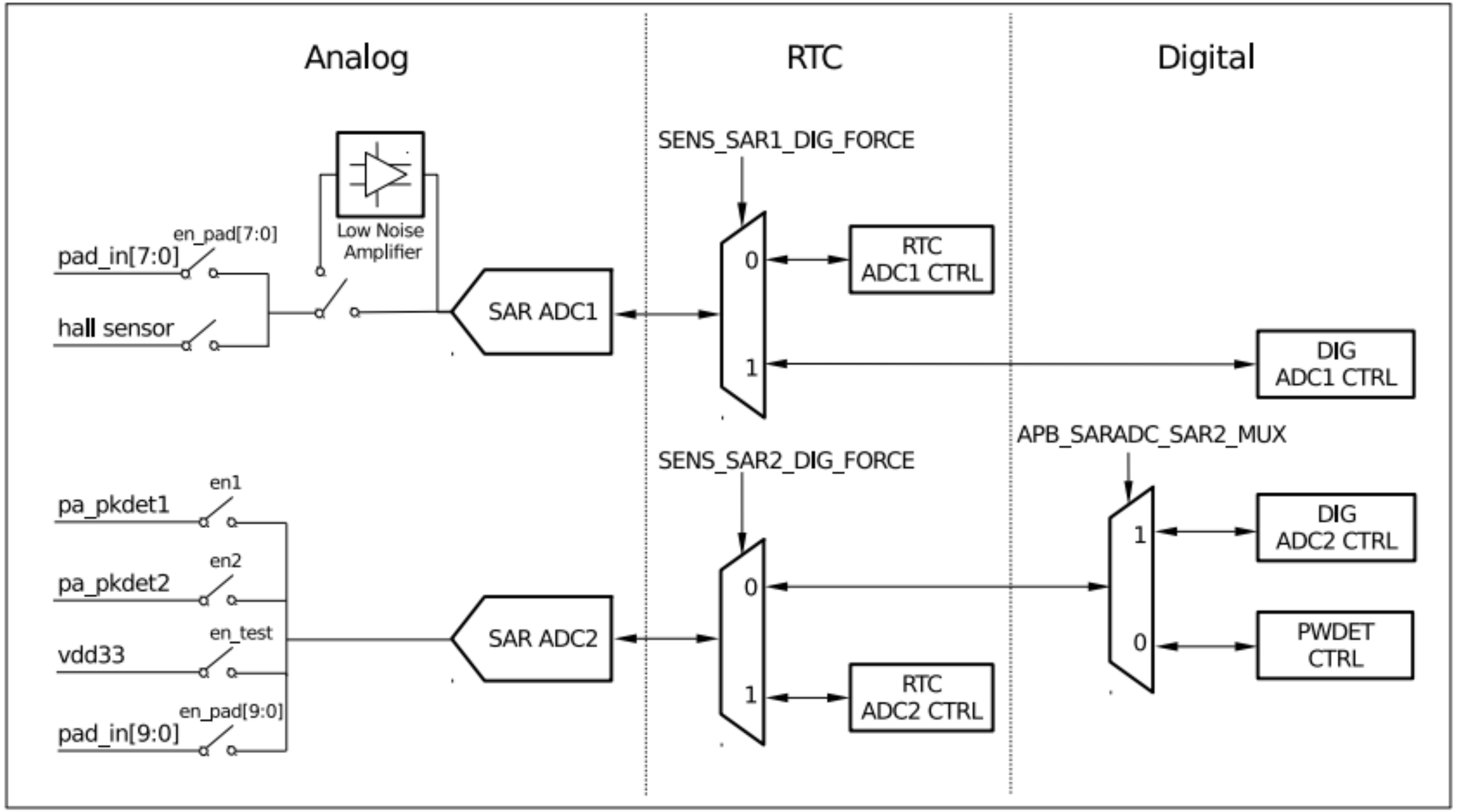


DAC

- Not all the bits may switch at once

![UCSC logo]

ESP32 ADC



Analog
RTC
Digital
SENS_SAR1_DIG_FORCE
Low Noise Amplifier
en_pad[7:0]
pad_in[7:0]
hall sensor
SAR ADC1
0
1
RTC ADC1 CTRL
DIG ADC1 CTRL
APB_SARADC_SAR2_MUX
SENS_SAR2_DIG_FORCE
en1
pa_pkdet1
en2
pa_pkdet2
en_test
vdd33
en_pad[9:0]
pad_in[9:0]
SAR ADC2
0
1
RTC ADC2 CTRL
1
0
DIG ADC2 CTRL
PWDET CTRL

XX

•

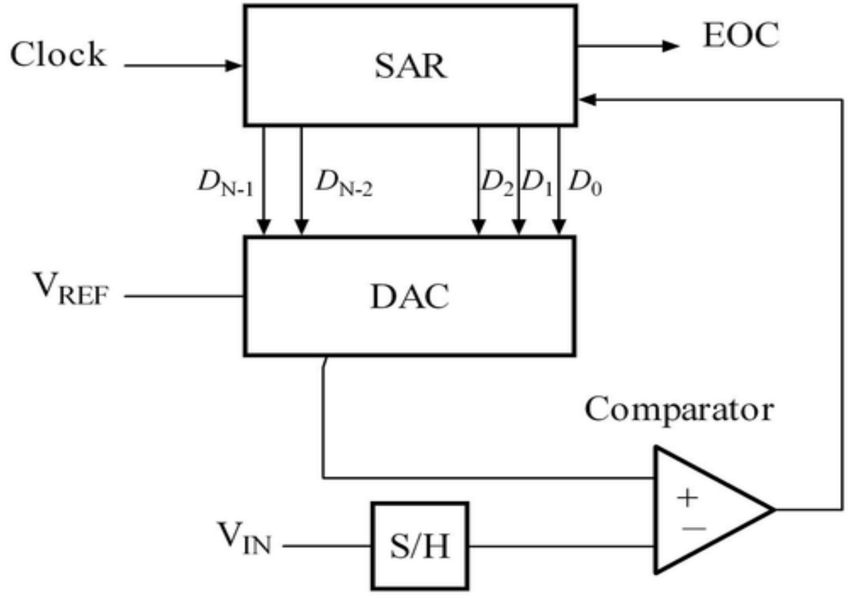
XX

•

- ESP32C3 SAR ADC
  - 2 channels
  - 12-bit, 11-bit, 10-bit, 9-bit configurable resolution
- ESP32C3 DAC
  - 2 8-bit channels

SAR ADC

- Successive approximation ADC
  - Vin approximated as a static value in a sample and hold (S/H) circuit
  - Successive approximation register (SAR) is a counter that increments each clock as long as it is enabled by the comparator
  - output of the SAR is fed to a DAC that generates a voltage to compare with VIN
  - when the output of the DAC = VIN the value of SAR is the digital representation of VIN
  - Brute-force search for VIN



Sampling rate and gyro?

- Check ESP32 doc…

Next Class

- Concurrency