How to Fool an ADC

Or how to hide the destruction of a turbine with the help of DSP

Alexander Bolshev	*Security Consultant*	*@dark_k3y*
Gabriel Gonzalez	*Principal Security Consultant*	*@gabrielgonzalez*

IOActive

Hardware | Software | Wetware
SECURITY SERVICES

Table of Contents

IOActive
Hardware | Software | Wetware
SECURITY SERVICES

Speakers Details

- Alexander Bolshev - Security Consultant @ Madrid HW Lab
  - alexander.bolshev@ioactive.com
  - @dark_k3y

- Gabriel Gonzalez – Principal Security Consultant @ Madrid HW Lab
  - gabriel.gonzalez@ioactive.com
  - @gabrielgonzalez

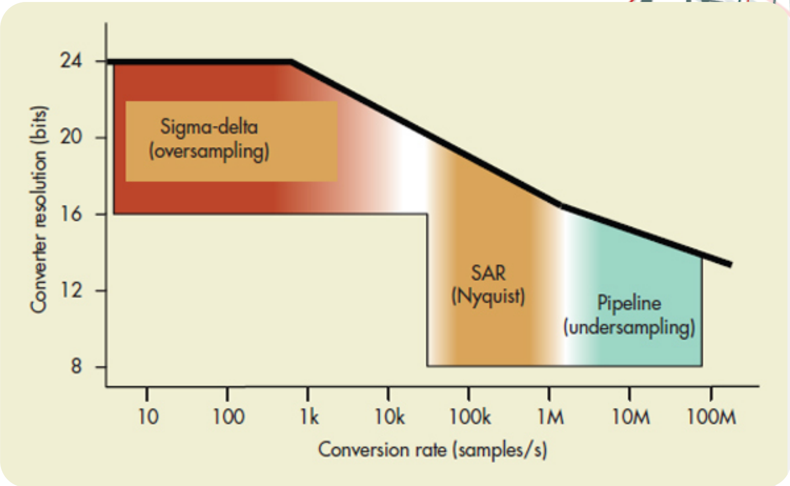
IOActive

Introduction to ADC

- A device that converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity's amplitude
- An ADC is defined by its bandwidth and its signal to noise ratio
- Bandwidth of an ADC is characterized primarily by its sampling rate, and to a lesser extent by how it handles errors such as aliasing

Introduction to ADC

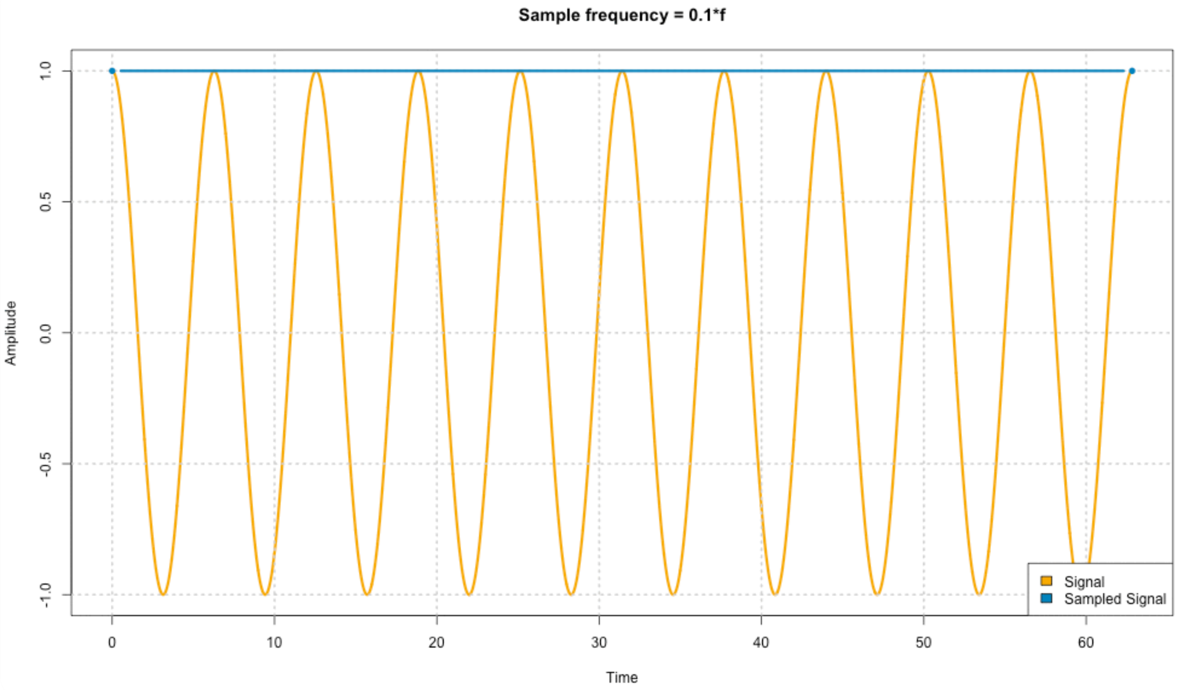
- More Common Types of ADC
  - Successive-approximation ADC (SAR)
  - Sigma-delta ADC
  - Pipeline



Introduction to ADC

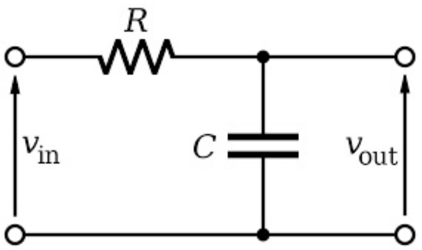
- Aliasing: Signal Distortion due to wrong sampling

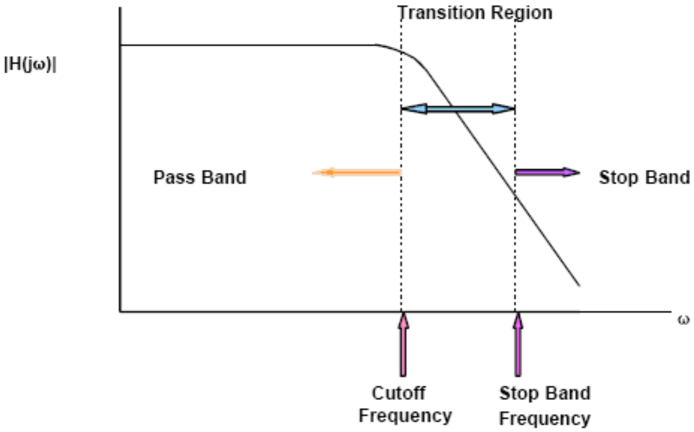
  Nyquist Rule: $f_s > 2f$



Introduction to ADC

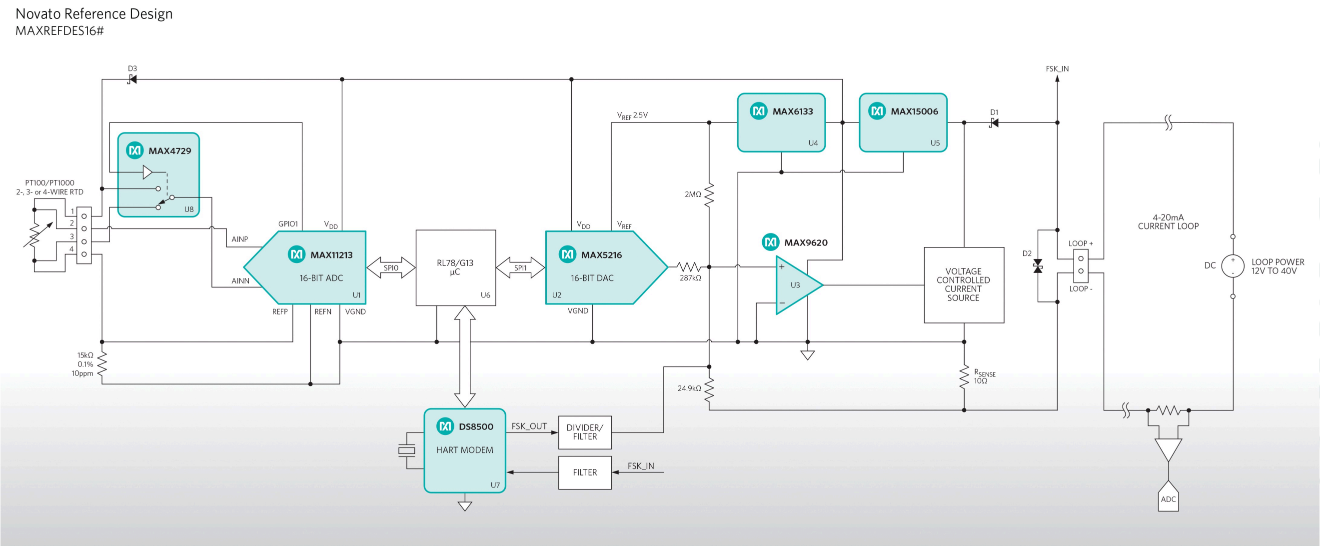
- Antia-Aliasing Filters
  - Cut the signal so it fits into the Nyquist Rule
  - Characterized by cut-off and stop-band frequencies





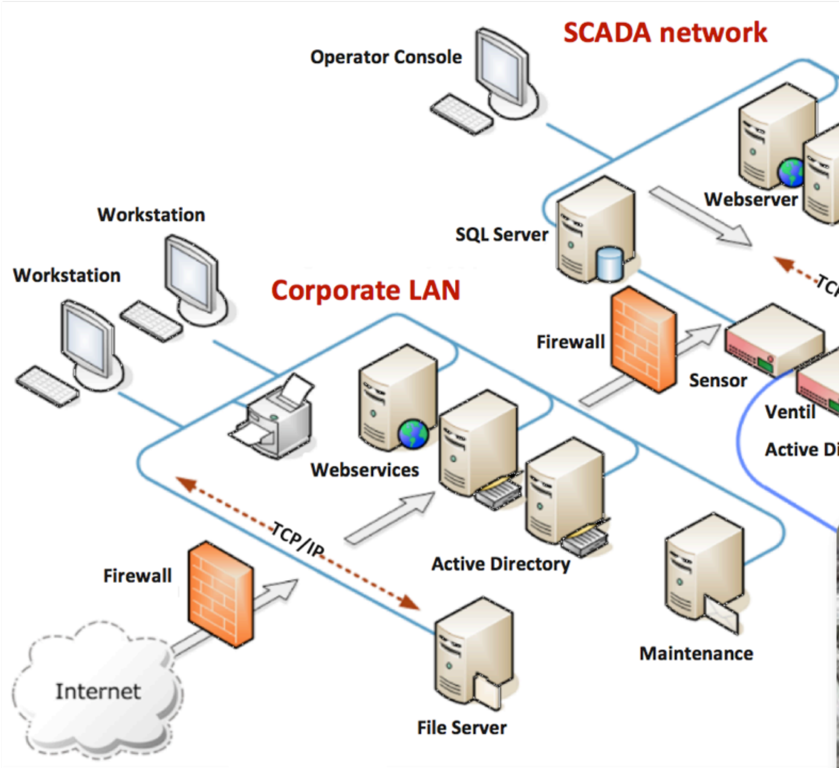
Introduction to ADC

- What is a DAC?



IOActive

ADCs and ICSs

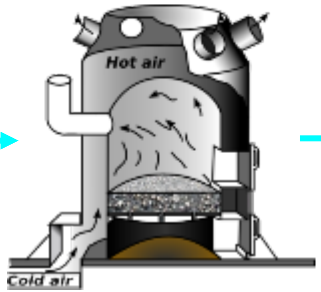


Catastrophic consequences

ADCs and ICSs

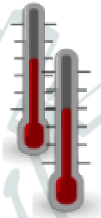
Adjust themselves to influence process behavior

Actuators



Physical Process

Sensors

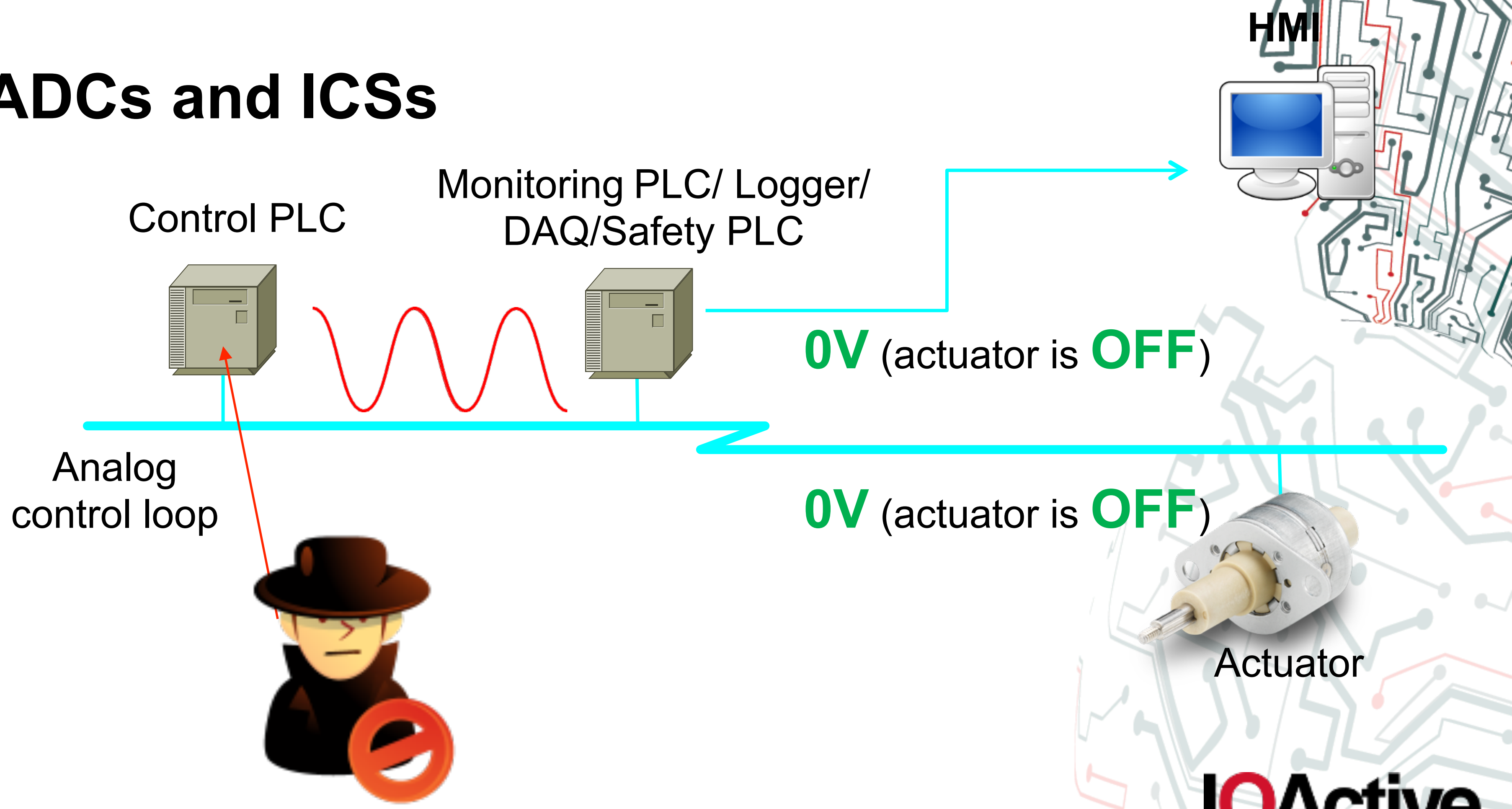


Measure process state

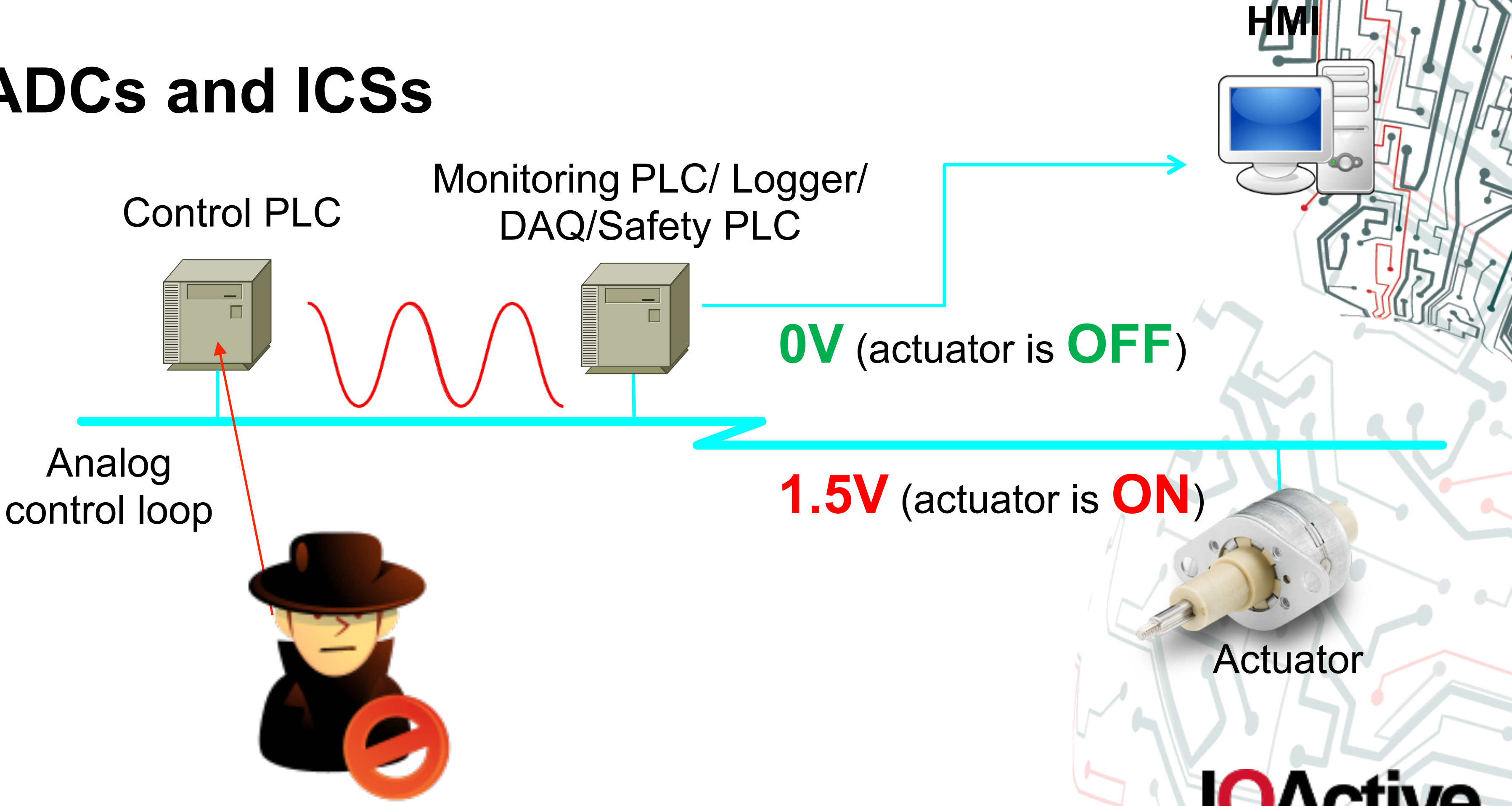
Control system

Computes control commands for actuators

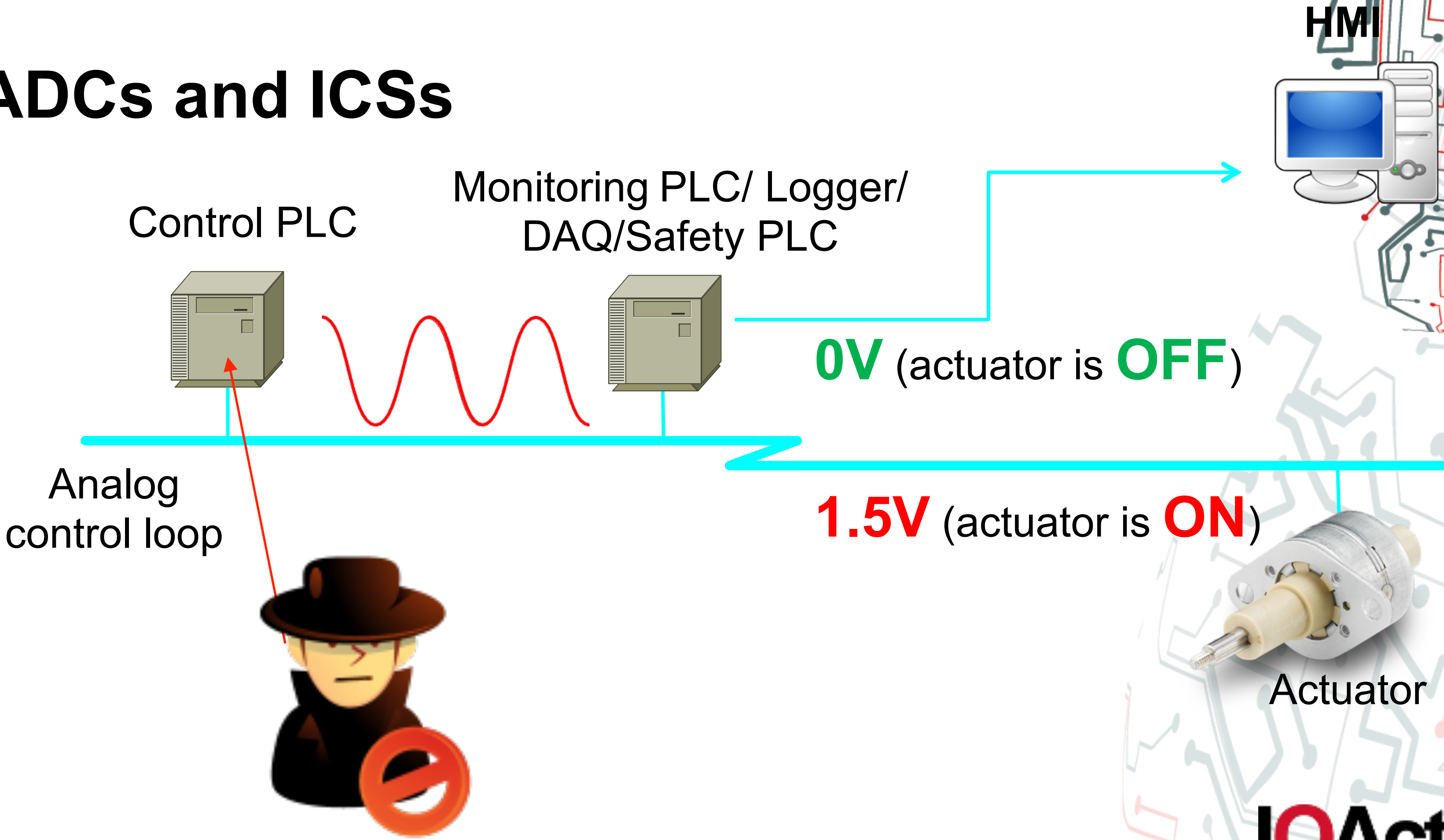
ADCs and ICSs



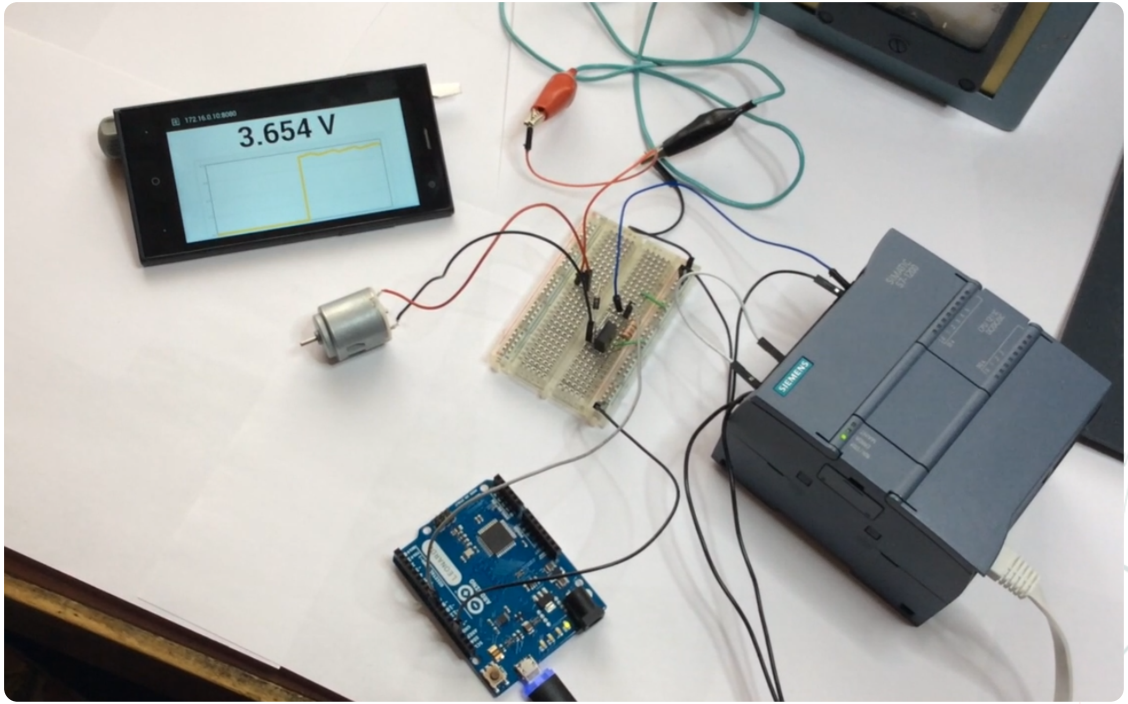
ADCs and ICSs



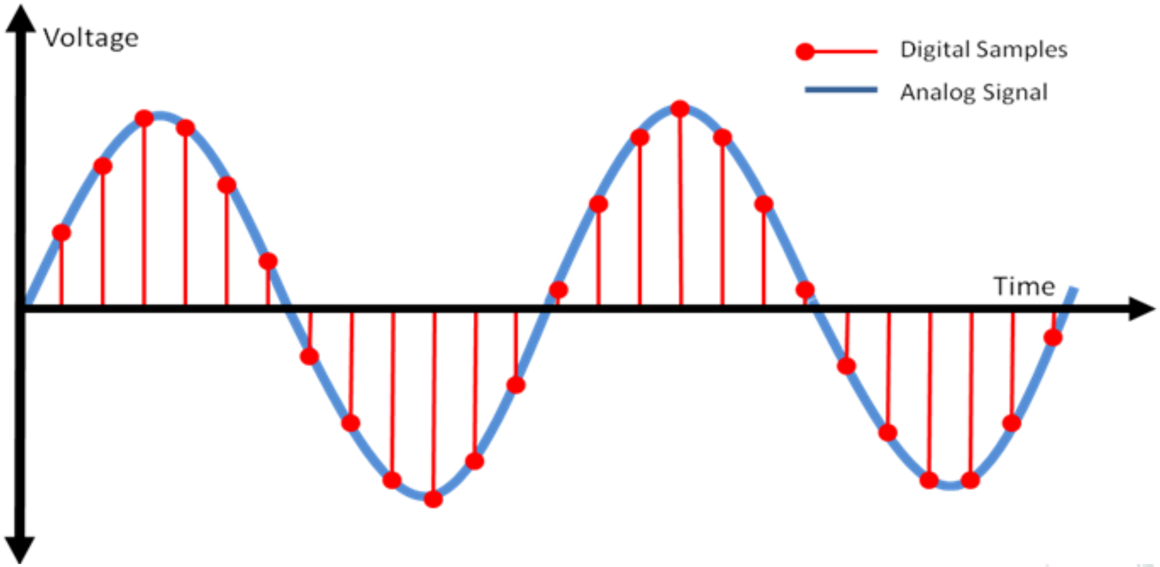
ADCs and ICSs



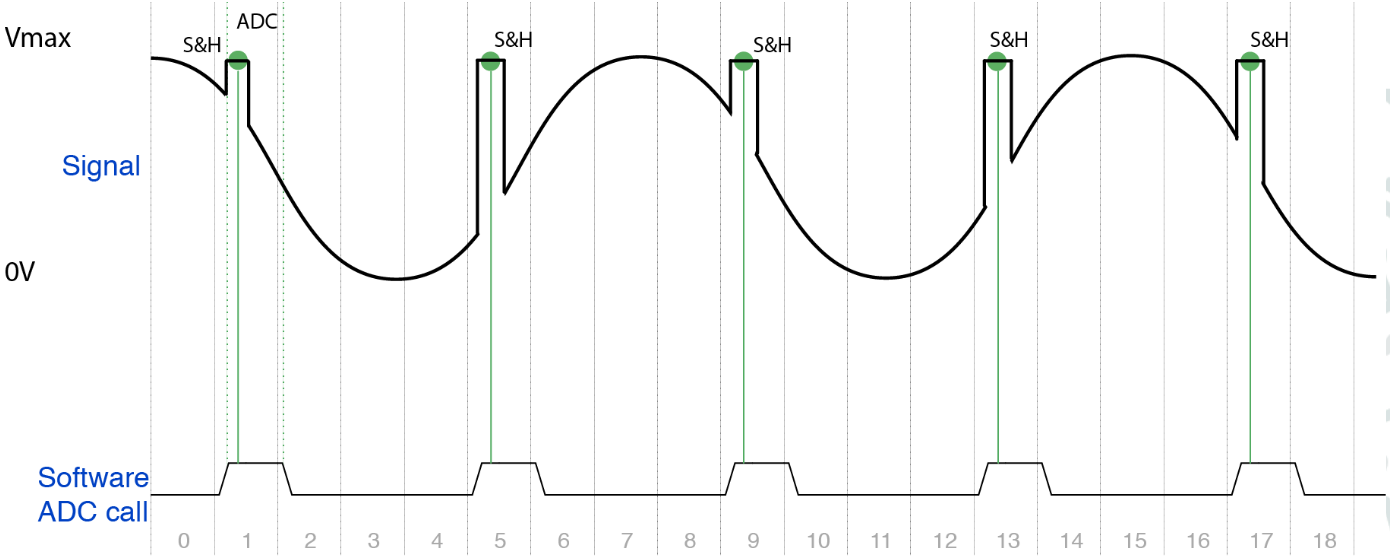
DEMO



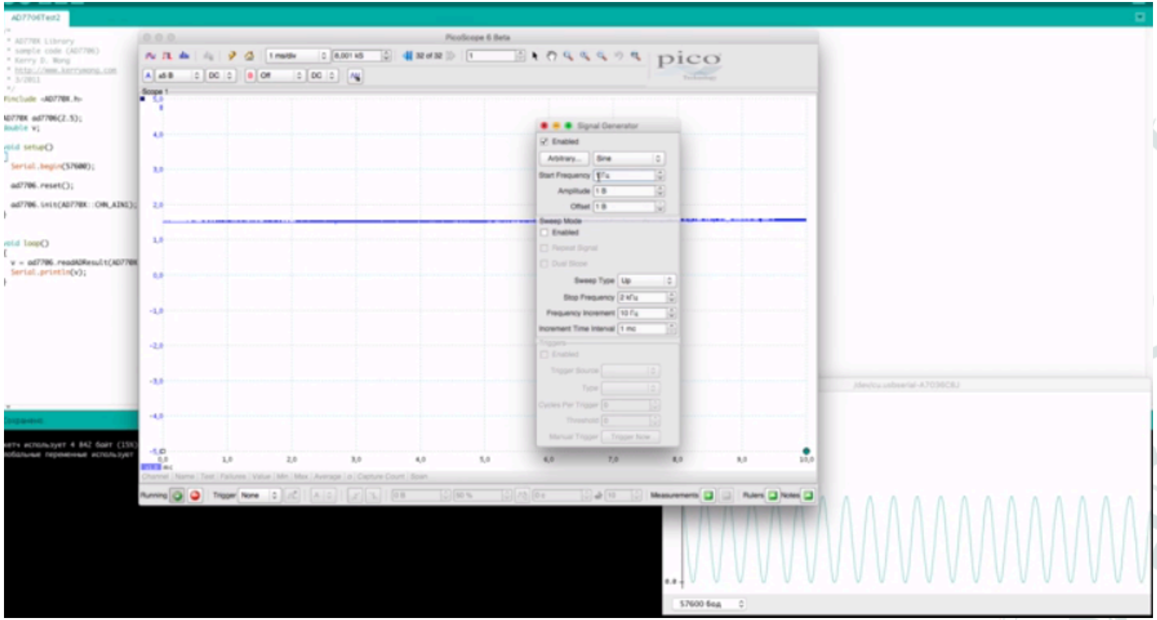
Attacking ADCs: Frequency



Attacking ADCs: Frequency



DEMO

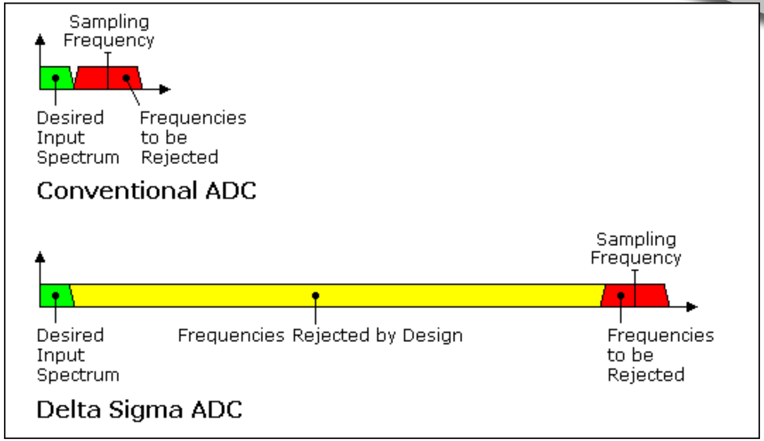


Attacking ADCs: Frequency

- Delta – Sigma ADCs

Oversampling

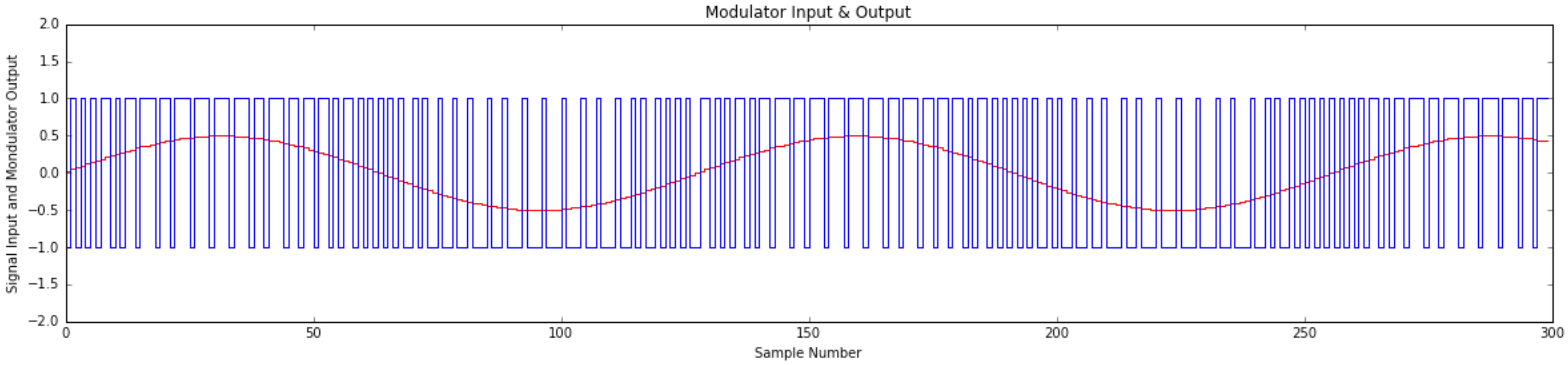
Delta-sigma ADCs use sample rates that are a large multiple, for instance, 128 times the sample rate sufficient for a given signal. For example, to sample a 25 kHz signal, a sample rate greater than the Nyquist rate (that is, > 50 KHz) would be sufficient. However, a delta-sigma ADC using an oversample factor of 128 samples the signal at 6 MHz. This approach has several benefits, such as better antialiasing and higher resolution.



Thus it becomes obvious that conventional ADCs need expensive low pass filters in order to obtain a bandwidth close to the theoretical Nyquist limit. Delta sigma converters require simple RC low pass filters only and with a little more expense for a 2^{nd} order filter one will get a virtually ideal behaviour. On the other hand an output low pass filter preceeding the decimator is required, which again can be realized more precisely, easily and cheap in digital techniques. Note that in practice the proportions are much more extreme than in the graphic above as due to limited space an oversampling rate of approx. 16 only is shown there.

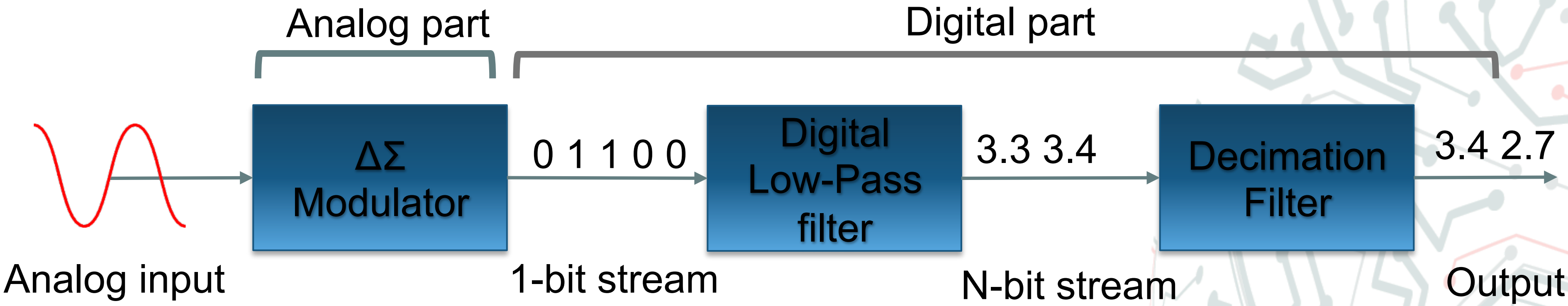
Attacking ADCs: Frequency

- Delta – Sigma ADCs
  - **Δ**-modulation the change in the signal is encoded.
  - The result is a stream of pulses
  - Accuracy improved using 1-bit DAC
  - Adding (**Σ**) the resulting analog signal to the input signal to reduce error



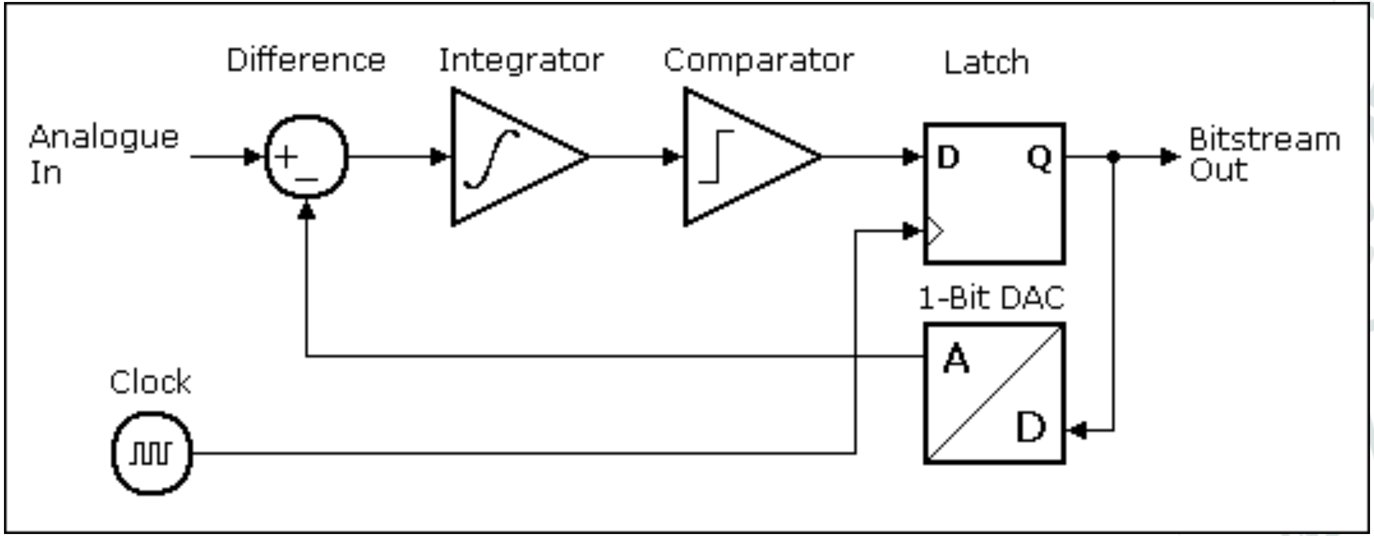
Attacking ADCs: Frequency

- Delta – Sigma ADCs
  - Analog part: High Frequency bitstream
  - Digital part: digital filter and decimation



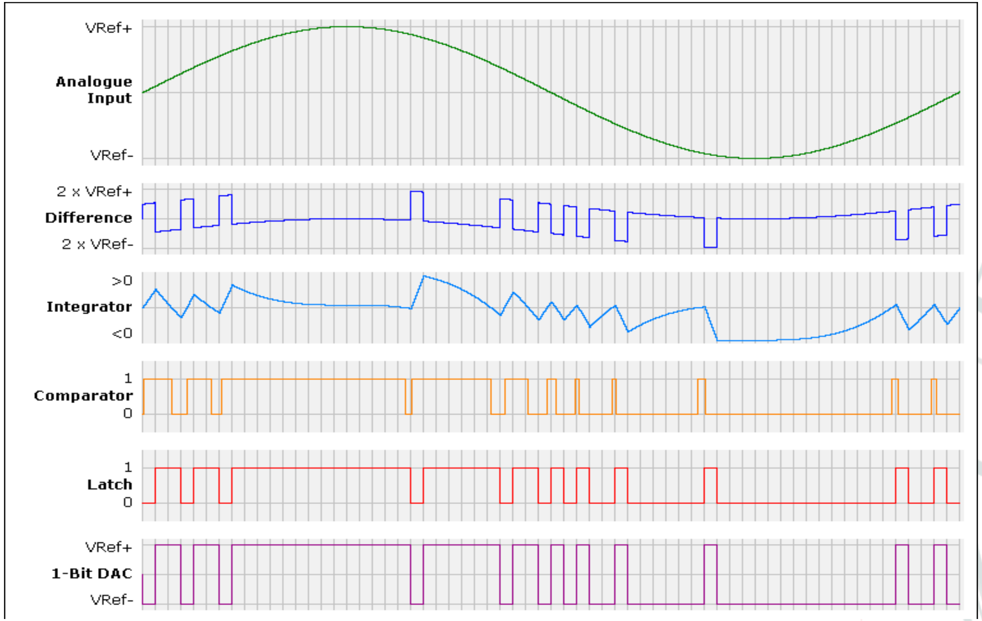
Attacking ADCs: Frequency

- Delta – Sigma ADCs
  - Analog part: High Frequency bitstream



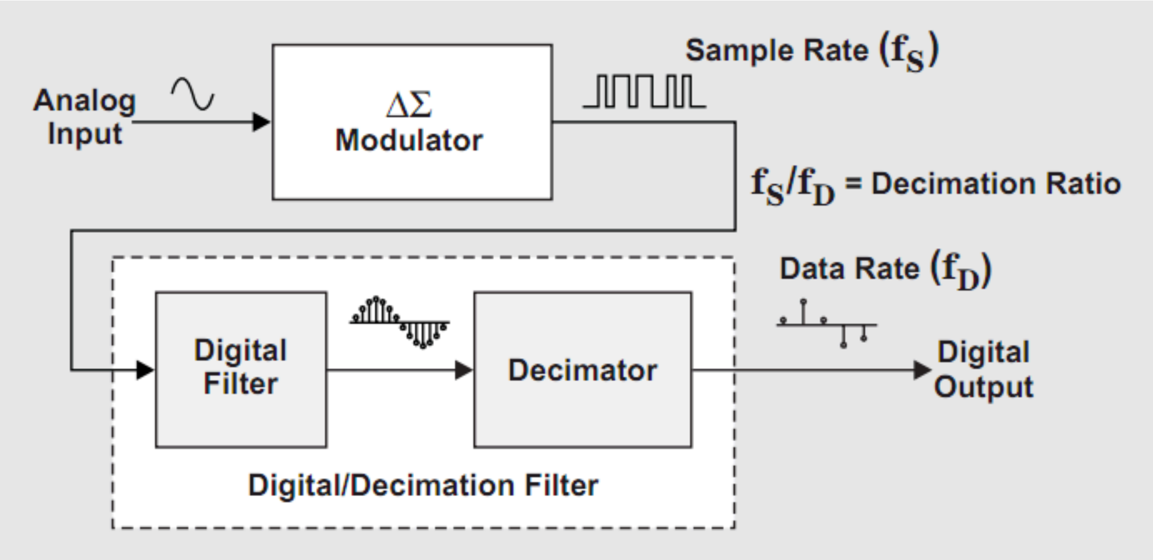
Attacking ADCs: Frequency

- Delta – Sigma ADCs
  - Modulation



Attacking ADCs: Frequency

- Delta – Sigma ADCs
  - Digital Part



Attacking ADCs: Frequency

- Delta – Sigma ADCs: Possible Problems
  - Filter Design / Implementation
  - Decimator Implementation

FUNCTIONAL BLOCK DIAGRAM

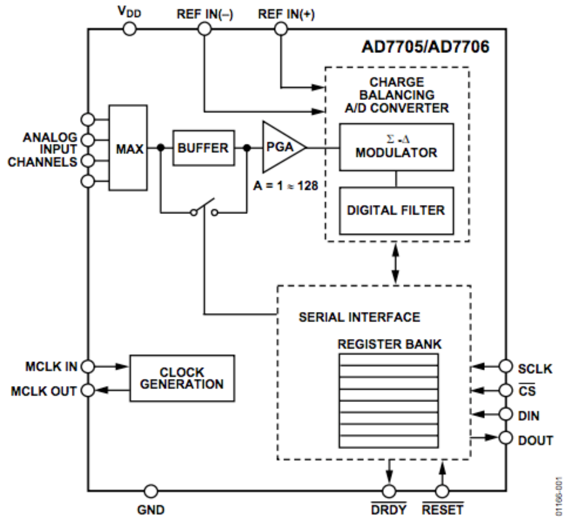
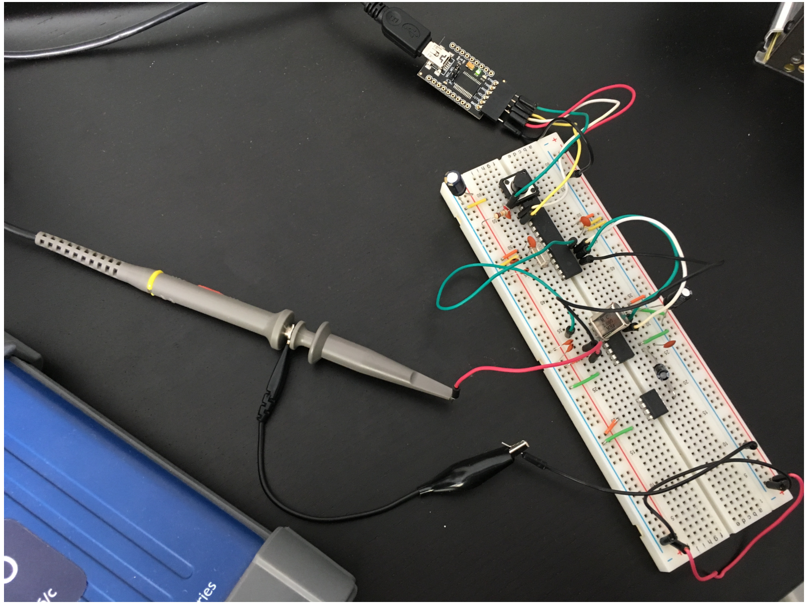


Figure 1.

Attacking ADCs: Frequency

- Delta – Sigma ADCs: Demo Setup



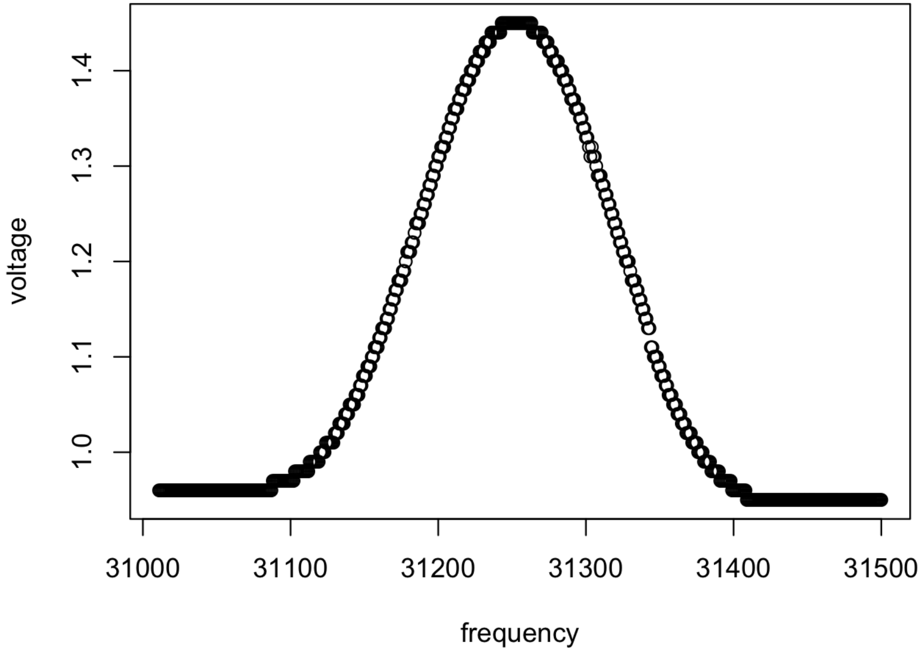
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Demo Steps

  - Supply sinewave with frequency f to the ADC inputs
  - For 1 second acquire f_d voltage samples (equal to ADC samples per second rate) in digital form
  - Calculate and record V_{max}, V_{min}, V_{avg} and stddev(V) for acquired samples
  - Increase frequency by 1 and goto (1)

Attacking ADCs: Frequency

- Delta – Sigma ADCs: Pardon me?



Attacking ADCs: Frequency

- Delta – Sigma ADCs: Results

Table 23. Input Sampling Frequency vs. Gain

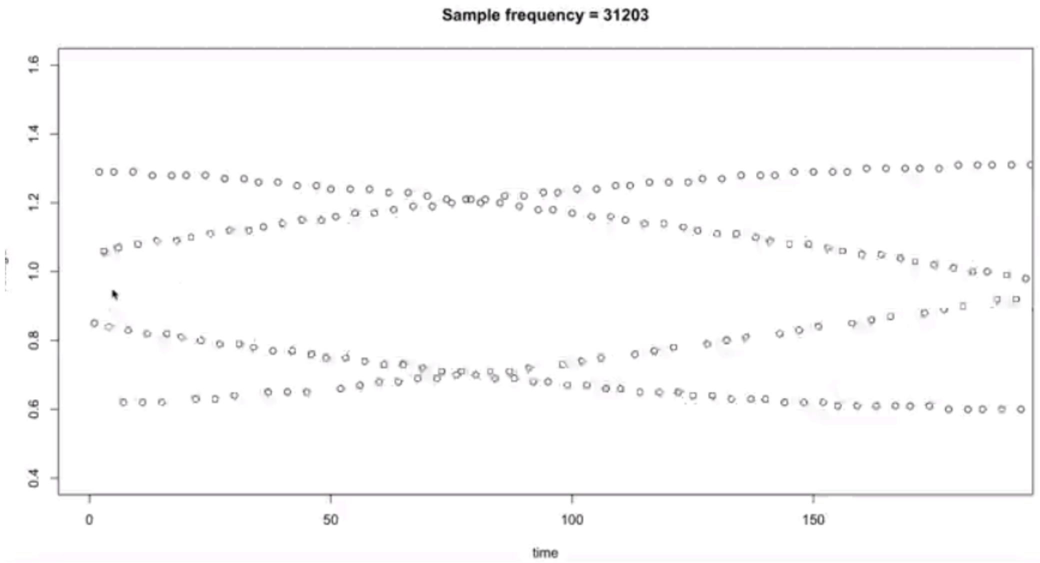
Gain	Input Sampling Frequency (f_S)
1	$f_{CLKIN}/64$ (38.4 kHz @ f_{CLKIN} = 2.4576 MHz)
2	$2 \times f_{CLKIN}/64$ (76.8 kHz @ f_{CLKIN} = 2.457
4	$4 \times f_{CLKIN}/64$ (76.8 kHz @ f_{CLKIN} = 2.457
8 to 128	$8 \times f_{CLKIN}/64$ (307.2 kHz @ f_{CLKIN} = 2.45

In addition, the digital filter does not provide any rejection at integer multiples of the digital filter's sample frequency. However, the input sampling on the part provides attenuation at multiples of the digital filter's sampling frequency so that the unattenuated bands occur around multiples of the sampling frequency, f_S, as defined in Table 23. Thus, the unattenuated bands occur at $n \times f_S$ (where n = 1, 2, 3 . . .). At these frequencies, there are frequency bands $\pm f_{3\text{ dB}}$ wide ($f_{3\text{ dB}}$ is the cutoff frequency of the digital filter) at either side where noise passes unattenuated to the output.

IOActive

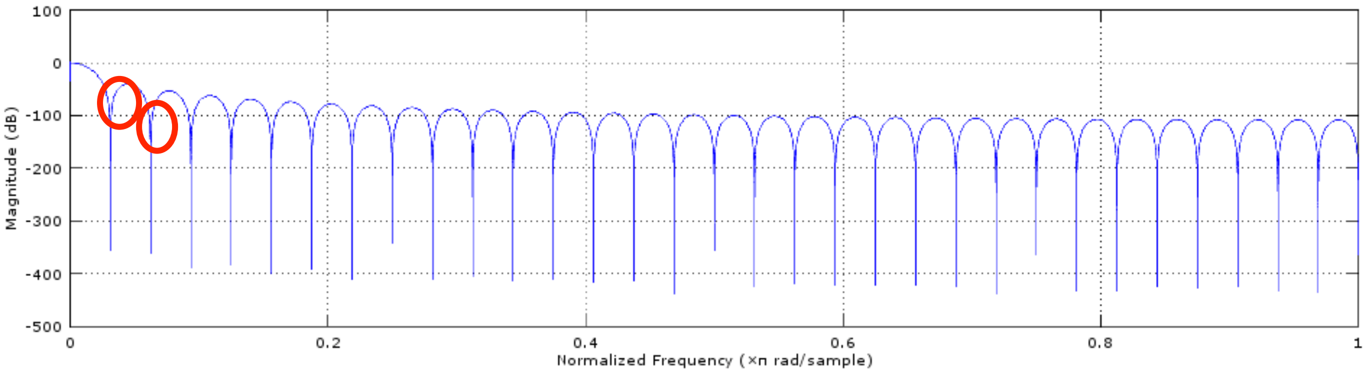
Attacking ADCs: Frequency

- Delta – Sigma ADCs: DEMO



Attacking ADCs: Frequency

- Delta – Sigma ADCs: Possible Explanation
  - This areas could case some noise transition, but not of that size



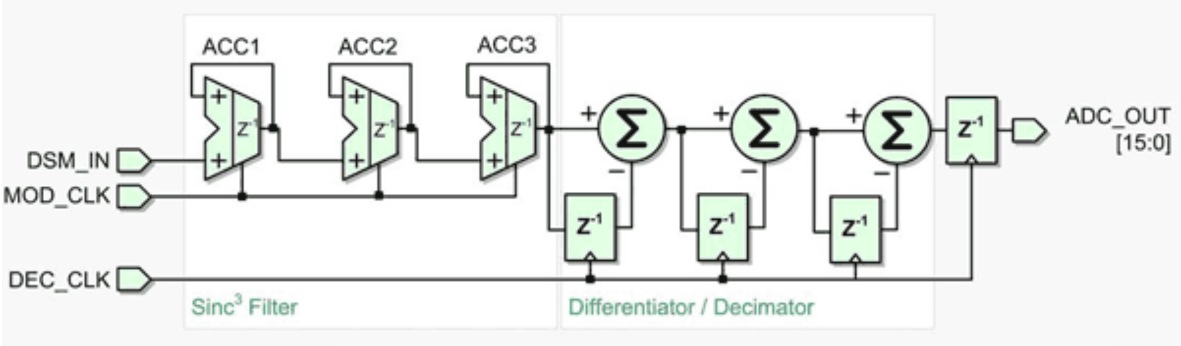
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Possible Explanation
  - Could it be incorrect implementation of digital LPF? But incorrect in what?
  - If (IF!) our filter is implemented badly, we could be in a situation with simple integer overflow.
  - Probability of last statement arises if we have ”MCU” inside ADC

IOActive

Attacking ADCs: Frequency

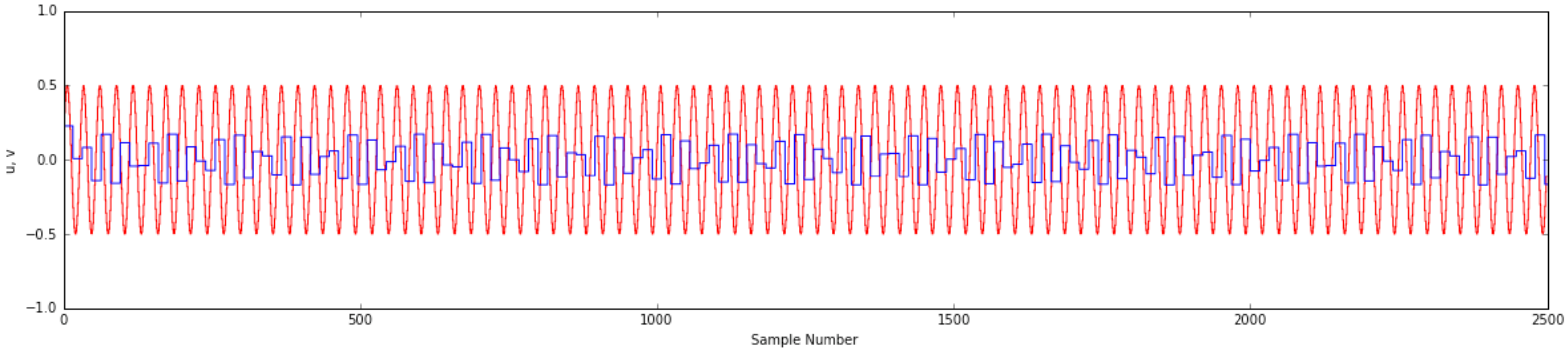
- Delta – Sigma ADCs: Possible Explanation




```
// apply the filter to each input sample
for ( n = 0; n < length; n++ ) {
    // calculate output n
    coeffp = coeffs;
    inputp = &insamp[filterLength - 1 + n];
    acc = 0;
    for ( k = 0; k < filterLength; k++ ) {
        acc += (*coeffp++) * (*inputp--);
    }
    output[n] = acc;
}
```

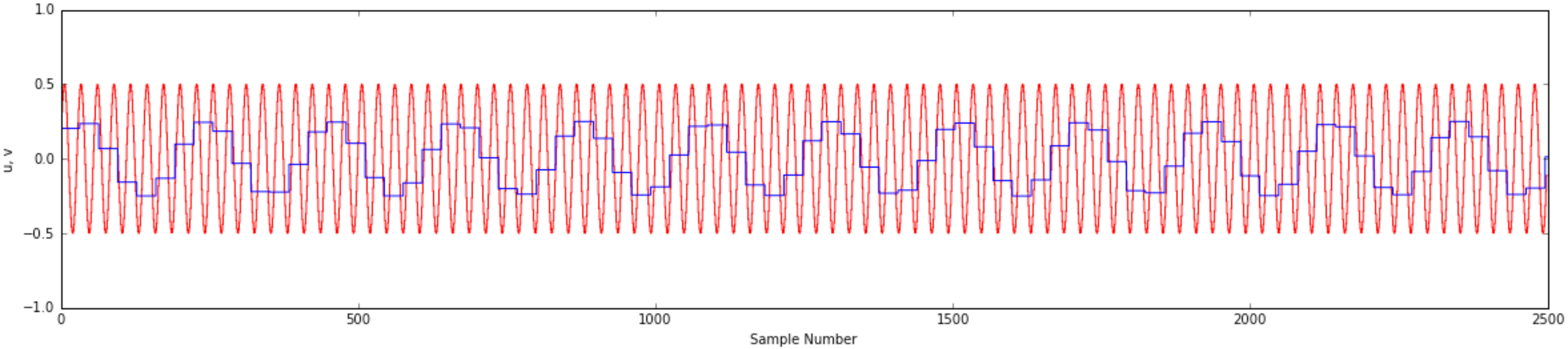
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Possible Explanation



Attacking ADCs: Frequency

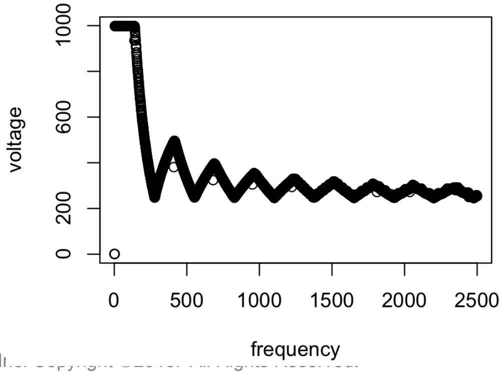
- Delta – Sigma ADCs: Possible Explanation

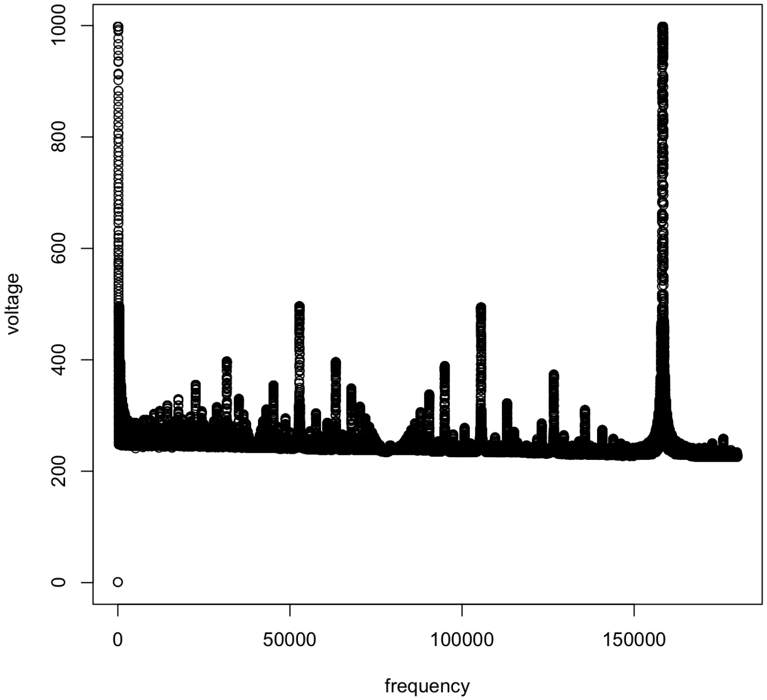


IOActive

Attacking ADCs: Frequency

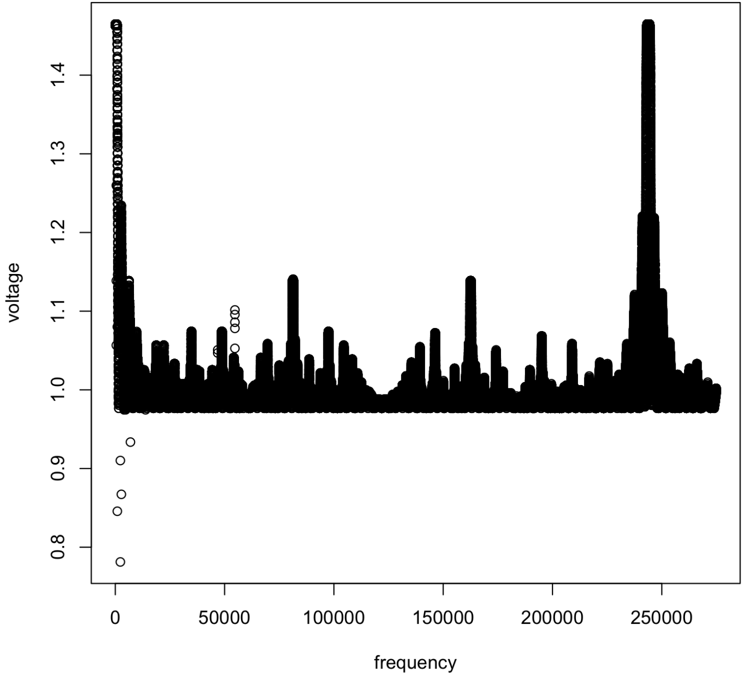
- Delta – Sigma ADCs: MCP3425





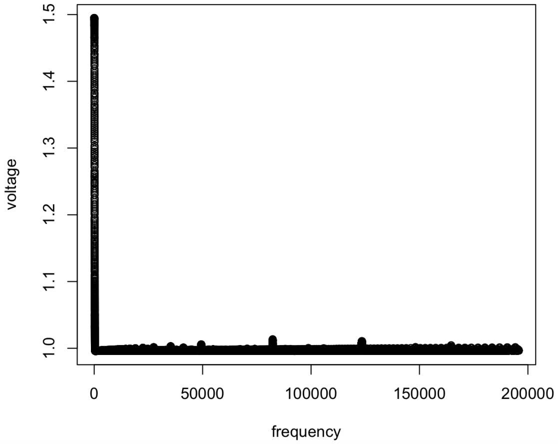
Attacking ADCs: Frequency

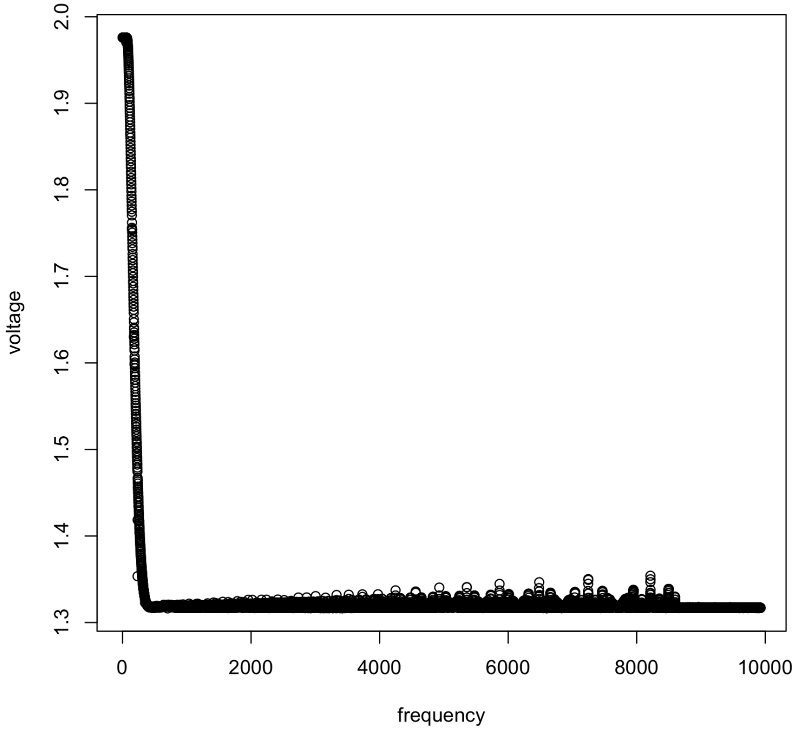
- Delta – Sigma ADCs: ADS1015



Attacking ADCs: Frequency

- Delta – Sigma ADCs: MAX11205

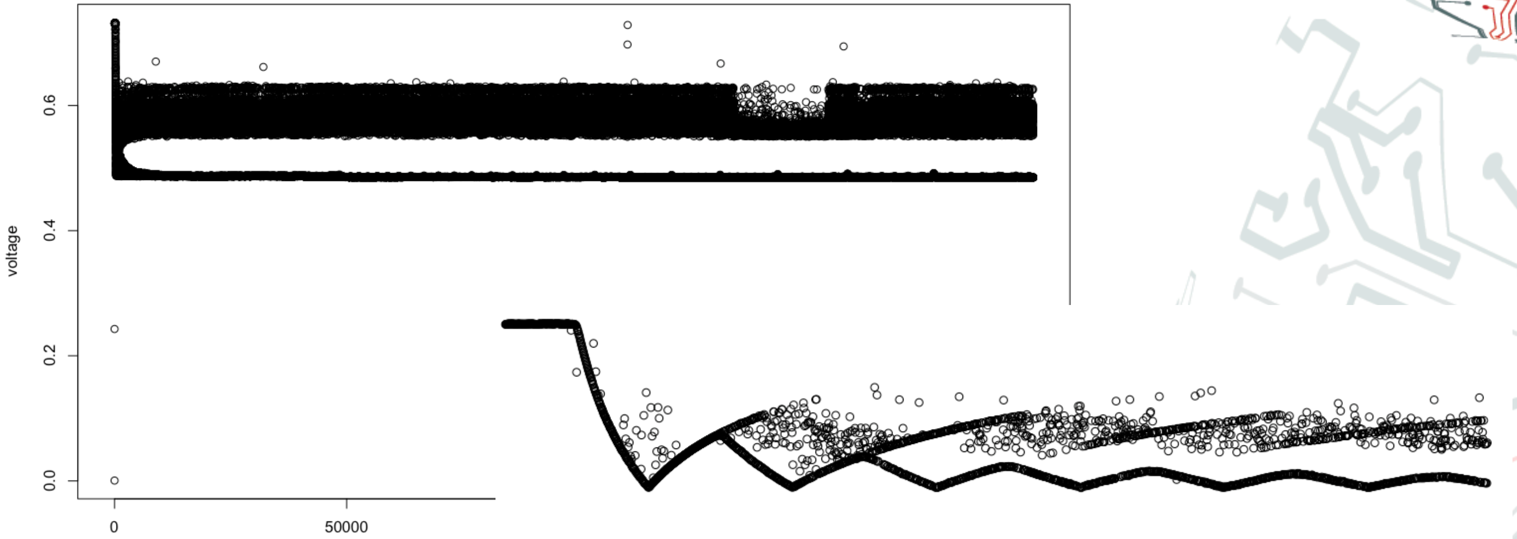




IOActive

Attacking ADCs: Frequency

- Delta – Sigma ADCs: Cypress PSoC5



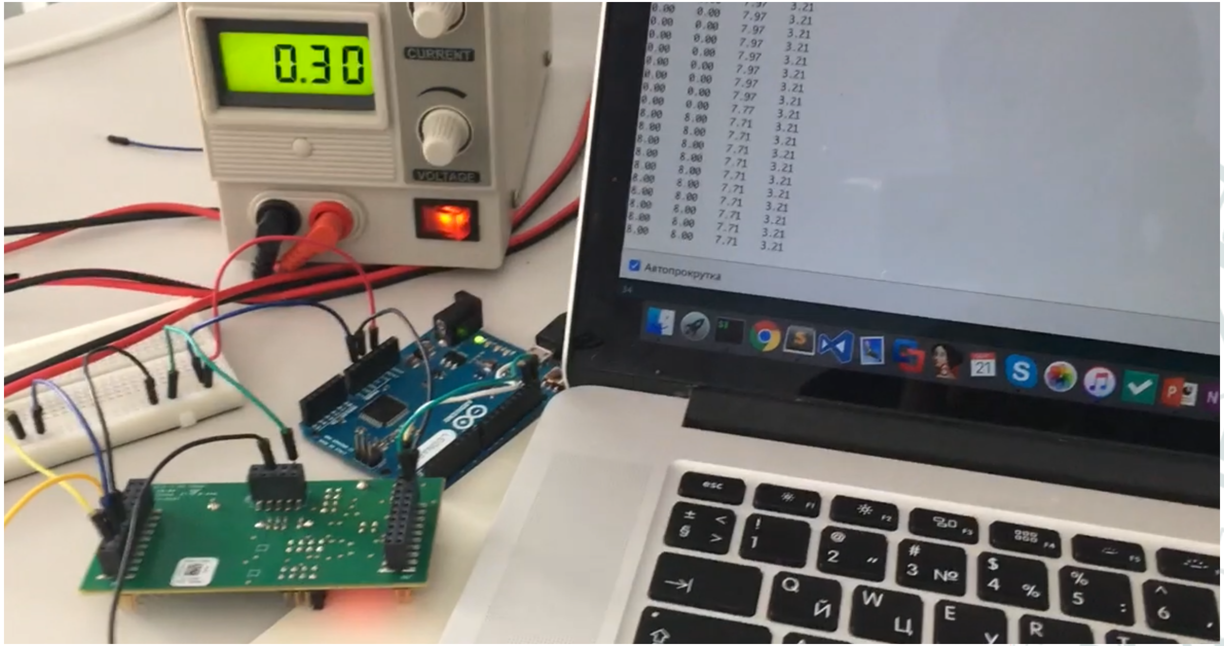
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Summary

ADC	First ”attackable” f	Required AAF f_c	Required AAF f_{sb}	Attack complexity
AD7705/AD7706	31250/38400 Hz	-	30kHz	easy
MCP3425	~51kHz	10-20kHz	30kHz	easy/medium
ADS1015	~86kHz	10-20kHz	50kHz	medium/hard
MAX11205	n/a	any reasonable	any reasonable	~impossible
*PSoC5 LP**	*~1kHz*	*1kHz*	*2kHz*	*n/a (medium)*

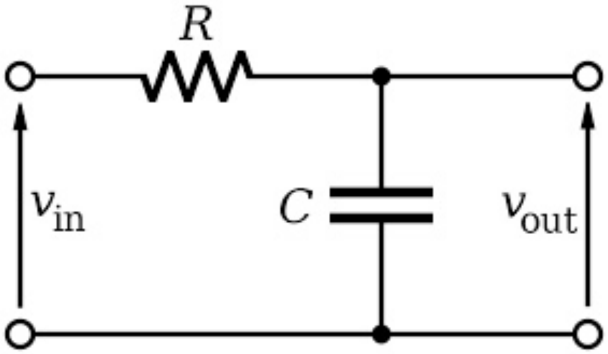
Attacking ADCs: Amplitude

- Demo



IOActive

Conclusion



- Not all **ΔΣ** are the same
- Use Anti Aliasing Filter before the Input

Q & A

IOActive