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
# Table of Contents

1. Speakers  
2. Introduction to ADC world  
3. ADC and ICSs  
4. Attacking ADCs: Frequency  
5. Attacking ADCs: Amplitude  
6. Conclusion
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
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?
ADCs and ICSs

Catastrophic consequences
ADCs and ICSs

Actuators: Adjust themselves to influence process behavior

Control system: Computes control commands for actuators

Sensors: Measure process state

Physical Process
ADCs and ICSs

Control PLC → Monitoring PLC/Logger/DAQ/Safety PLC

- Analog control loop
- 0V (actuator is OFF)
- 0V (actuator is OFF)
ADCs and ICSs

Control PLC

Monitoring PLC/ Logger/ DAQ/Safety PLC

Analog control loop

Actuator

0V (actuator is OFF)

1.5V (actuator is ON)
ADCs and ICSs

Control PLC

Monitoring PLC/ Logger/ DAQ/Safety PLC

Analog control loop

Actuator

0V (actuator is OFF)

1.5V (actuator is ON)
DEMO
Attacking ADCs: Frequency
Attacking ADCs: Frequency
Attacking ADCs: Frequency

- Delta – Sigma ADCs

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 2nd order filter one will get a virtually ideal behaviour. On the other hand an output low pass filter preceding 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
  - $\Delta$-modulation: the change in the signal is encoded.
  - The result is a stream of pulses
  - Accuracy improved using 1-bit DAC
  - Adding ($\Sigma$) 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

![Diagram of analog and digital parts of a Delta-Sigma ADC]
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
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 \( V_{\text{max}} \), \( V_{\text{min}} \), \( V_{\text{avg}} \) 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>
<thead>
<tr>
<th>Gain</th>
<th>Input Sampling Frequency (f_s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>f_{CLKIN}/64 (38.4 kHz @ f_{CLKIN} = 2.4576 MHz)</td>
</tr>
<tr>
<td>2</td>
<td>2 \times f_{CLKIN}/64 (76.8 kHz @ f_{CLKIN} = 2.457)</td>
</tr>
<tr>
<td>4</td>
<td>4 \times f_{CLKIN}/64 (76.8 kHz @ f_{CLKIN} = 2.457)</td>
</tr>
<tr>
<td>8 to 128</td>
<td>8 \times f_{CLKIN}/64 (307.2 kHz @ f_{CLKIN} = 2.457)</td>
</tr>
</tbody>
</table>

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 \ldots). 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.
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
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Possible Explanation

```c
// 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
Attacking ADCs: Frequency

- Delta – Sigma ADCs: MCP3425
Attacking ADCs: Frequency

- Delta – Sigma ADCs: ADS1015
Attacking ADCs: Frequency

- Delta – Sigma ADCs: MAX11205
Attacking ADCs: Frequency

- Delta – Sigma ADCs: Cypress PSoC5
## Attacking ADCs: Frequency

- Delta – Sigma ADCs: Summary

<table>
<thead>
<tr>
<th>ADC</th>
<th>First &quot;attackable&quot; f</th>
<th>Required AAF $f_c$</th>
<th>Required AAF $f_{sb}$</th>
<th>Attack complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD7705/AD7706</td>
<td>31250/38400 Hz</td>
<td>-</td>
<td>30kHz</td>
<td>easy</td>
</tr>
<tr>
<td>MCP3425</td>
<td>~51kHz</td>
<td>10-20kHz</td>
<td>30kHz</td>
<td>easy/medium</td>
</tr>
<tr>
<td>ADS1015</td>
<td>~86kHz</td>
<td>10-20kHz</td>
<td>50kHz</td>
<td>medium/hard</td>
</tr>
<tr>
<td>MAX11205</td>
<td>n/a</td>
<td>any reasonable</td>
<td>any reasonable</td>
<td>~impossible</td>
</tr>
<tr>
<td>PSoC5 LP*</td>
<td>~1kHz</td>
<td>1kHz</td>
<td>2kHz</td>
<td>n/a (medium)</td>
</tr>
</tbody>
</table>
Attacking ADCs: Amplitude

- Demo
Conclusion

- Not all $\Delta\Sigma$ are the same
- Use Anti Aliasing Filter before the Input
Q & A