NEVER TRUST YOUR INPUTS:
CAUSING 'CATASTROPHIC PHYSICAL CONSEQUENCES' FROM THE SENSOR (OR HOW TO FOOL ADC)

Alexander Bolshev & Marina Krotofil

UBM

; CAT /DEV/USER

Alexander @dark_k3y Bolshev, Ph.D.
Security Researcher @ IOActive
Assistant Professor @ SPbETU “LETI”

Marina @marmusha Krotofil
Security Researcher @HoneywellSec

IOActive
Hardware | Software | Wetware
SECURITY SERVICES

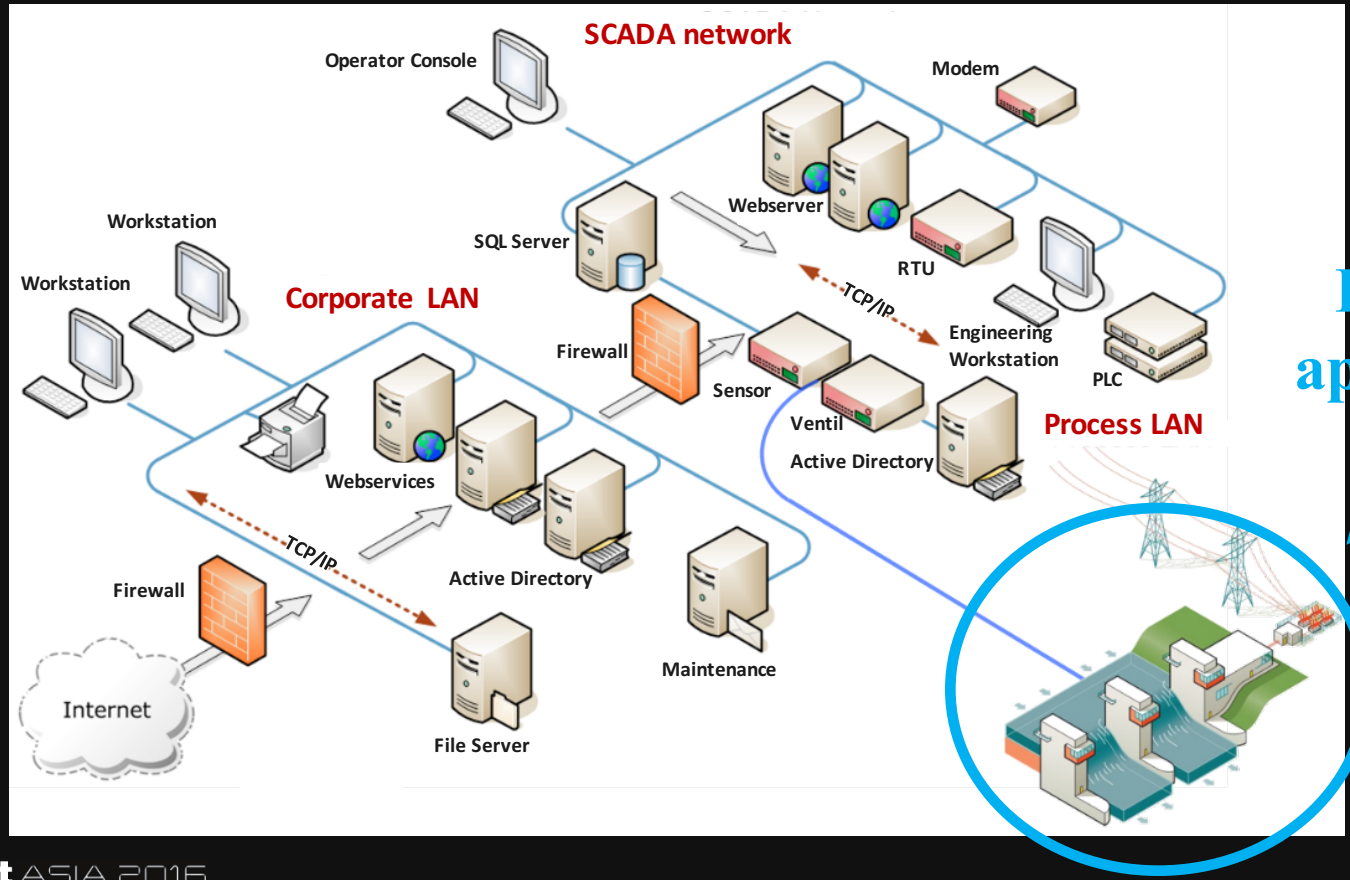
1886

Honeywell

AGENDA

- Problem statement
- Analog-to-Digital Converters (ADC)
- “Racing” with ADC clock
- Invalid amplitude range of signal
- Attack vectors in ICS
- Mitigations

INDUSTRIAL CONTROL SYSTEMS



PROCESS CONTROL IN A NUTSHELL

Actuators

Adjust themselves to influence process behavior

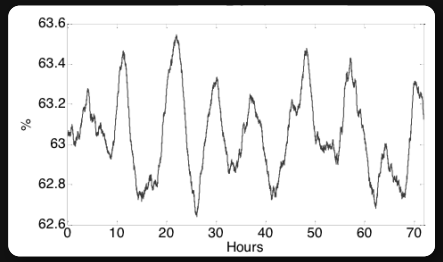
Physical process

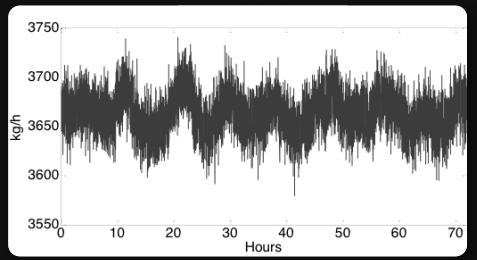
Sensors

Measure process state

Control system

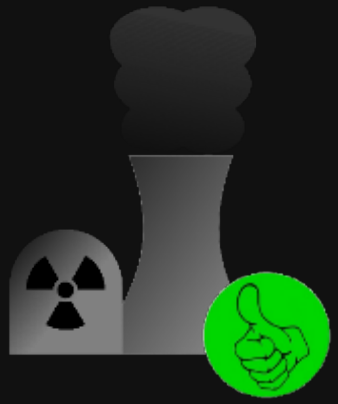
Computes control commands for actuators





IMPACT OF IMPROPER SIGNAL PROCESSING

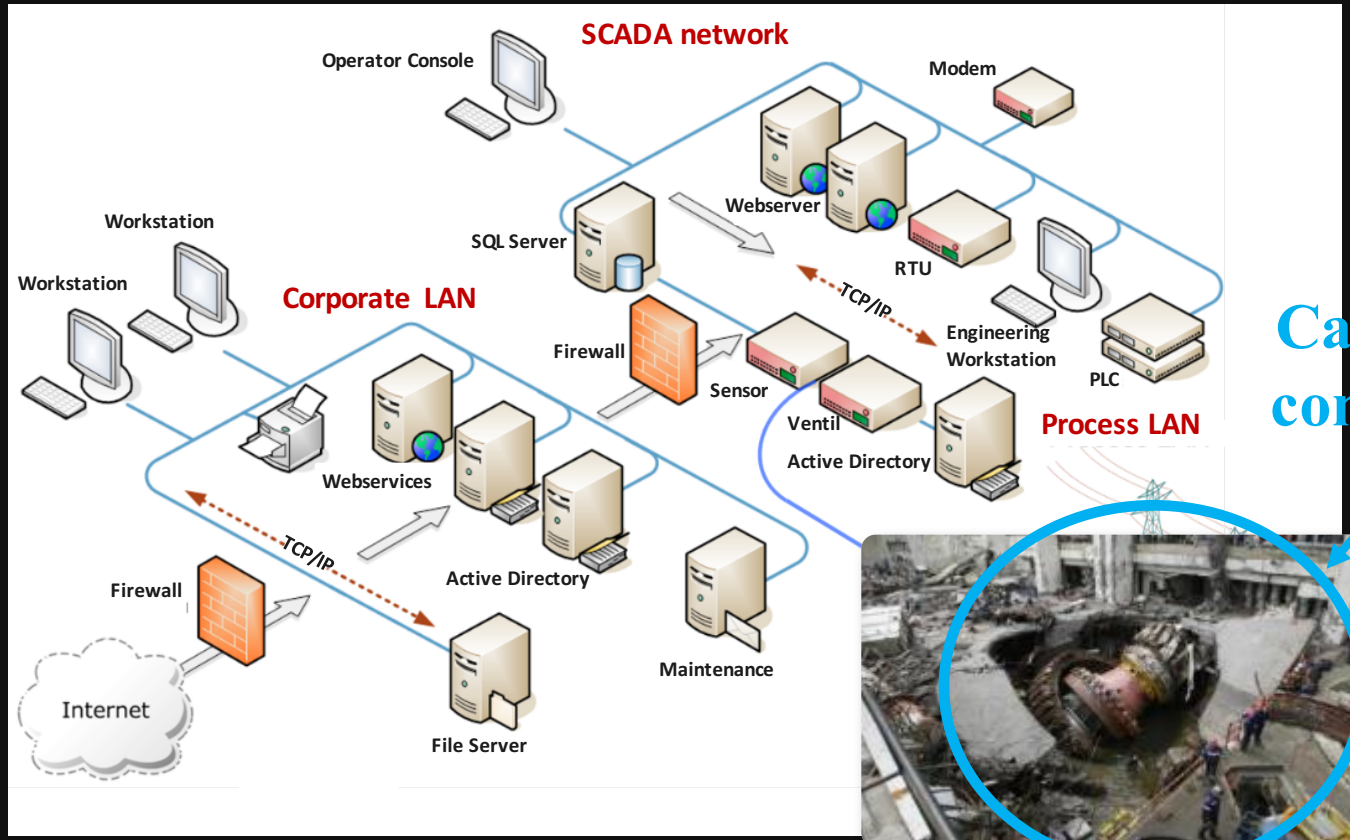
Equipment damage at nuclear plant



- Two identically built nuclear plants. One had flow induced vibration issue. And another did not.
- **The vibrations indication showed itself as hf noise**
  - Field engineer has filtered the signal to get rid of annoying noise
  - Loss of view into vibration issue

http://www.controlglobal.com/blogs/unfettered/marina-krotofils-presentation-on-how-to-hack-a-chemical-plant-and-its-implication-to-actual-issues-at-a-nuclear-plant/

REASON TO SECURE CONTROL SYSTEMS



Catastrophic consequences

PROCESS MONITORING

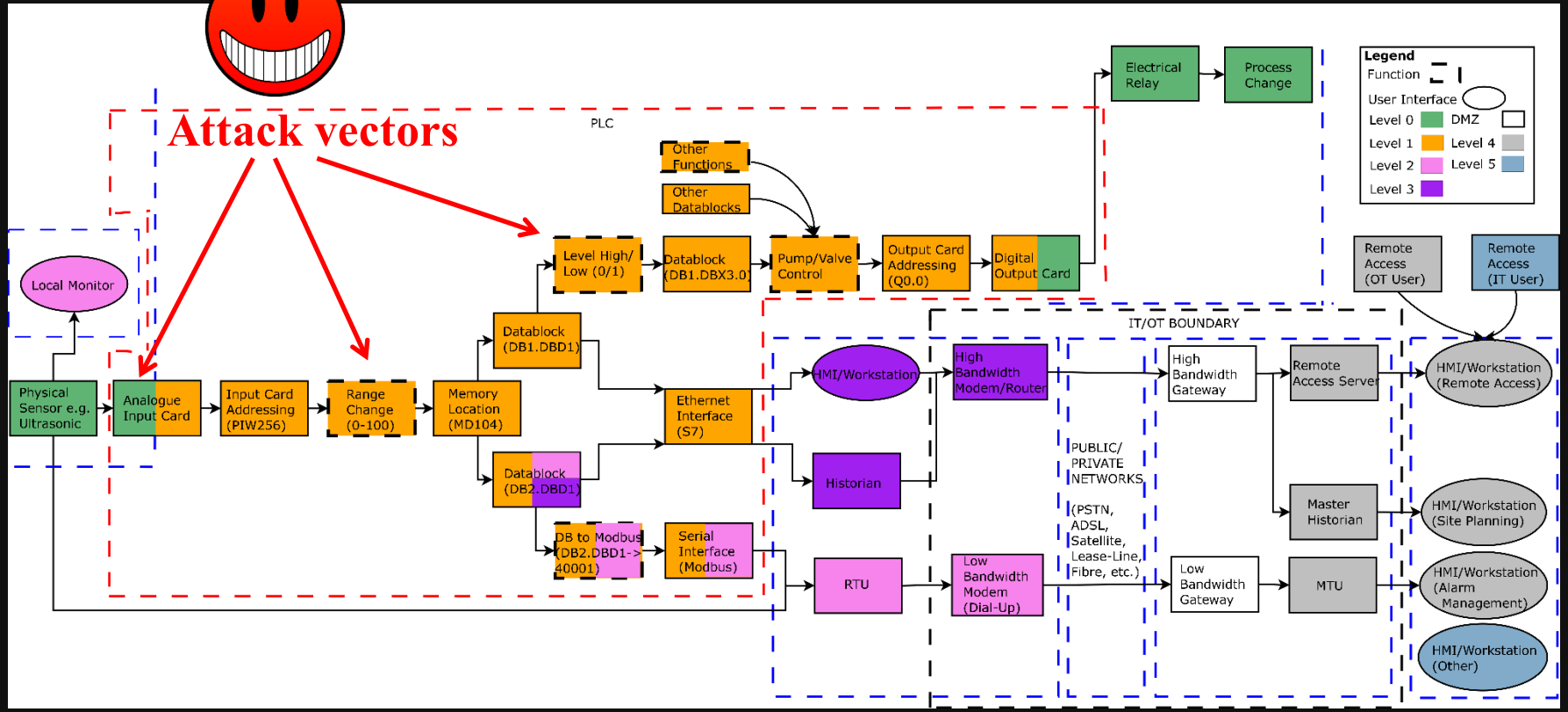
OPERATOR

OPERATOR CONSOLE (HMI)

CONTROL SYSTEM

PROCESS

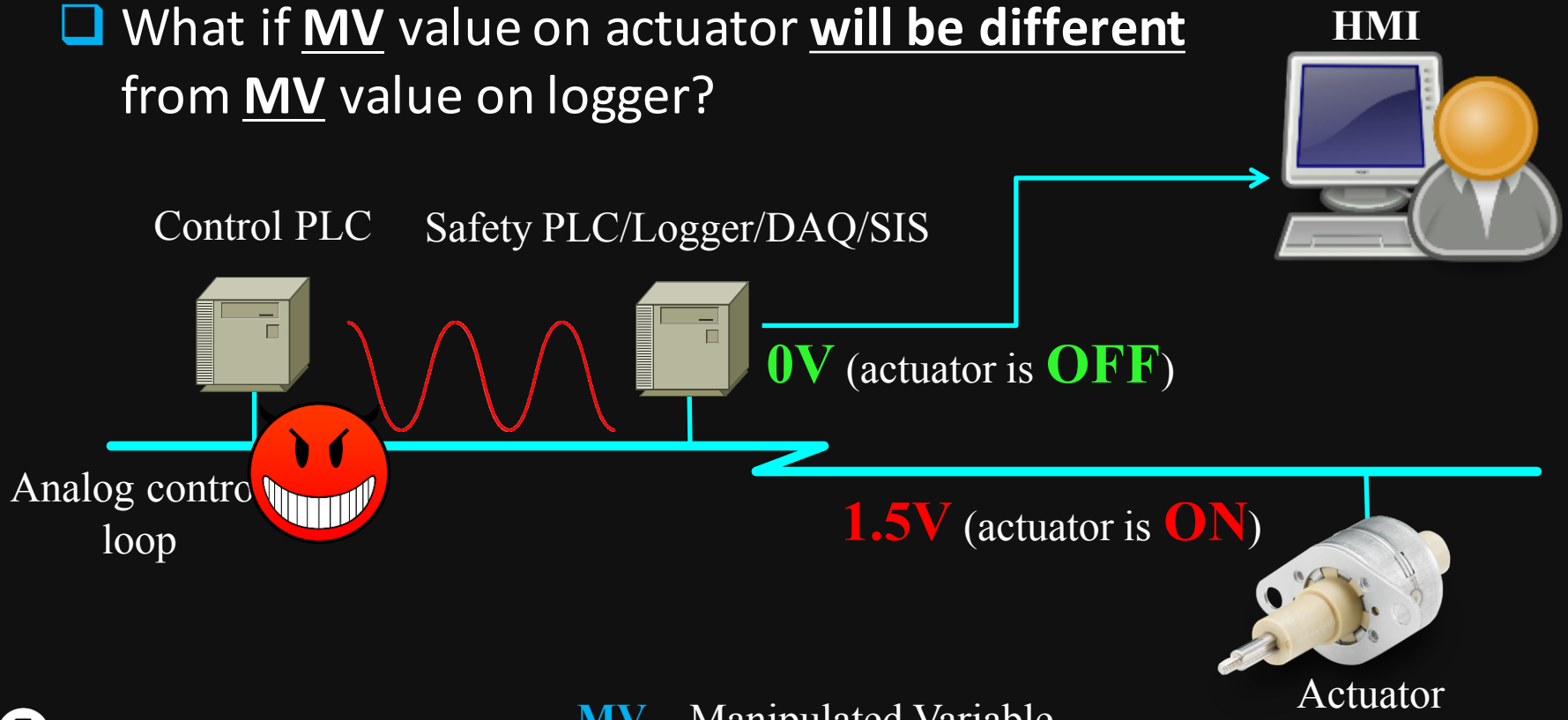
DATA PROCESSING & USE IN ICS (SINGLE SENSOR)



Courtesy B. Green, University of Lancaster

CONSIDER A FIELD ARCHITECTURE

- What if **MV** value on actuator **will be different** from **MV** value on logger?



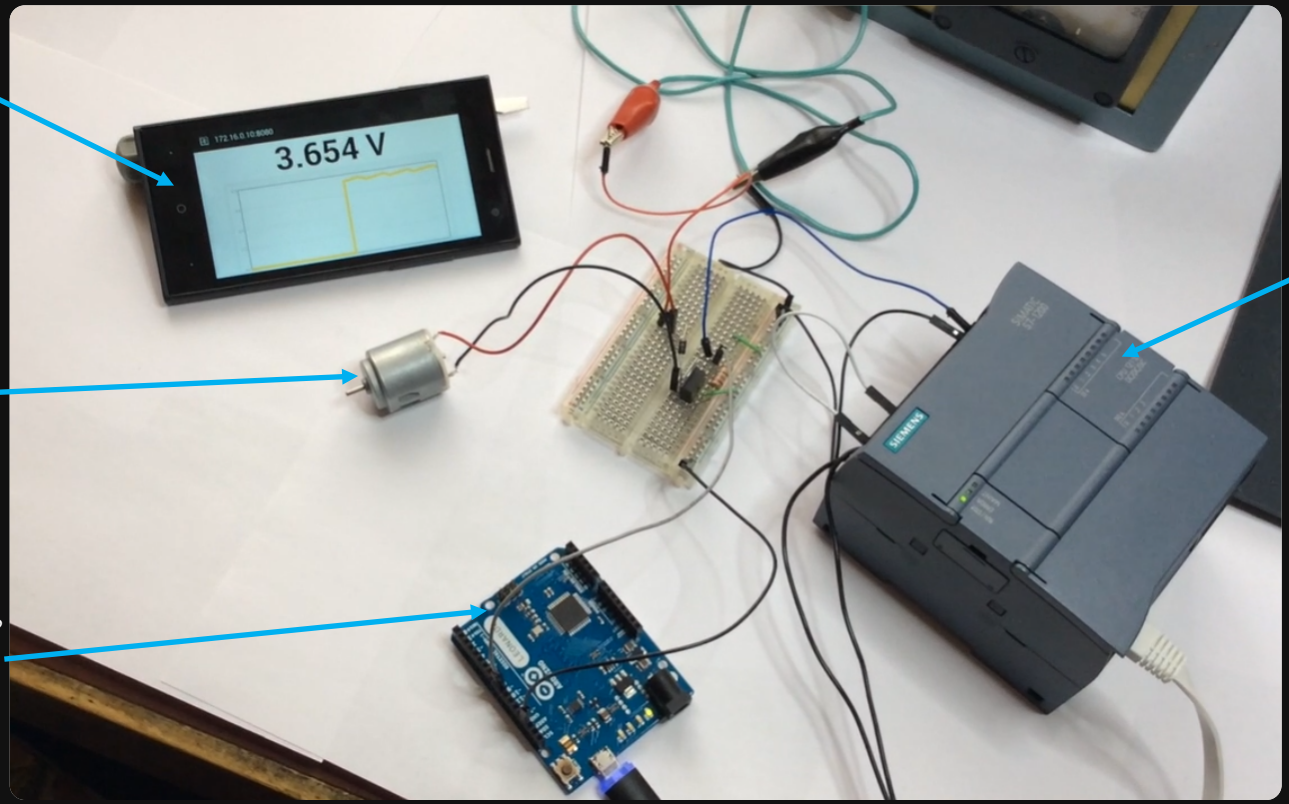
MV – Manipulated Variable

BUT IT'S ANALOG CONTOL LINE!

- It's impossible to have **two different MVs** on the same line at the same time!

Are you sure?

DEMO SETUP


"HMI Panel"
3.654 V
"Actuator"
(motor)
"Control PLC"
(arduino)
"Safety PLC"
(S7 1200)
SIEMENS

DEMO 1

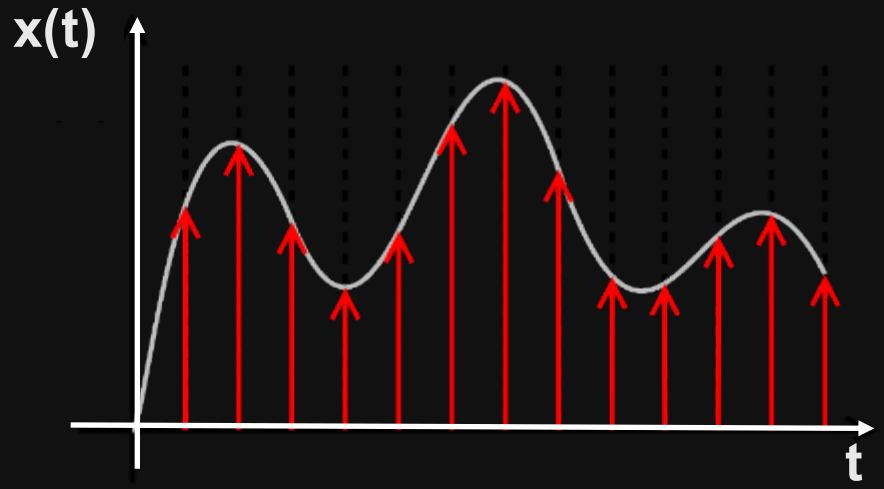
DEMO VIDEO

-- Two devices, two different MVs --

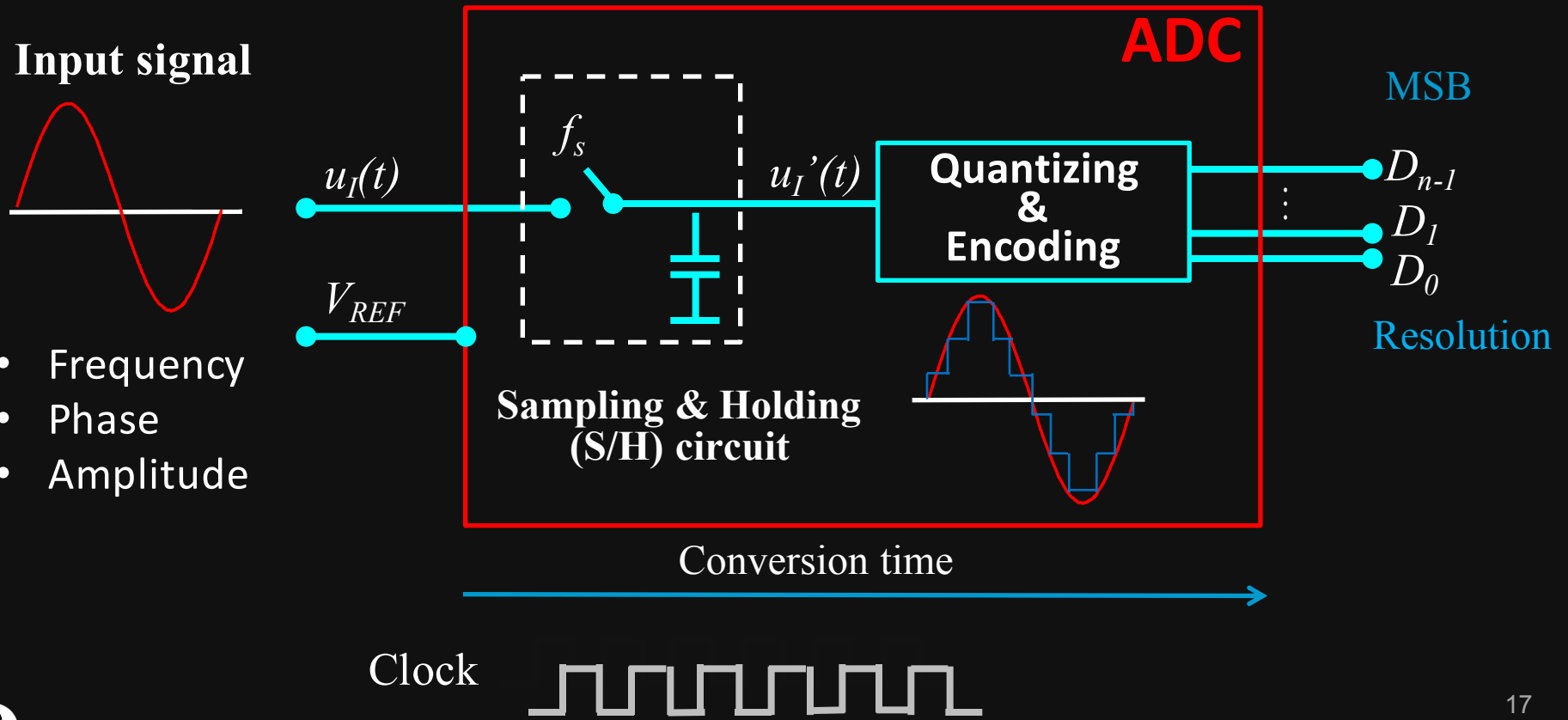
INTRO TO ANALOG-TO-DIGITAL CONVERTERS (ADC)

WHAT IS ADC?

- Converts a continuous analog signal (voltage or amperage) to a digital number that represents signal's amplitude



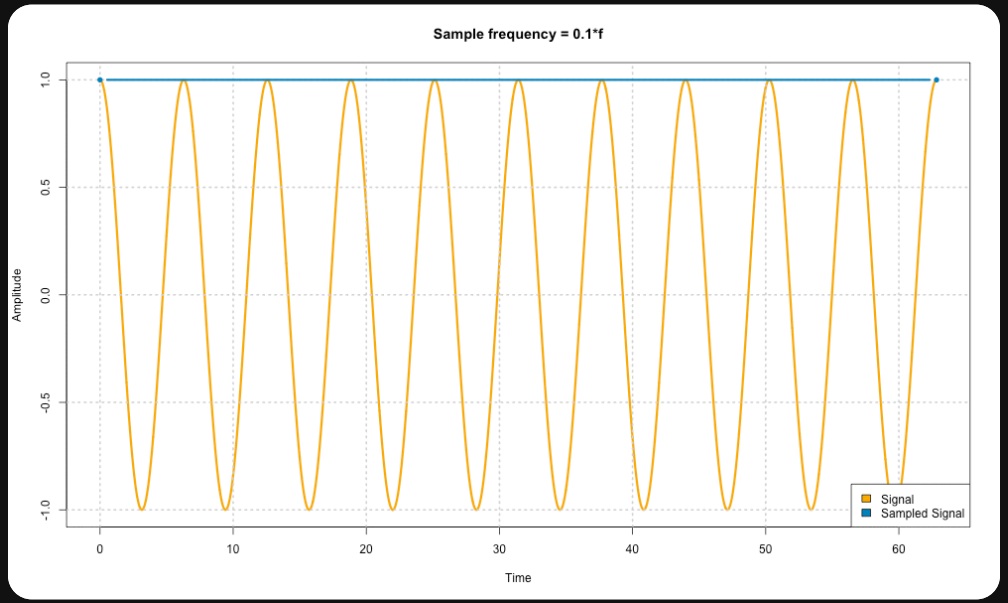
ADC IN A NUTSHELL



- Frequency
- Phase
- Amplitude

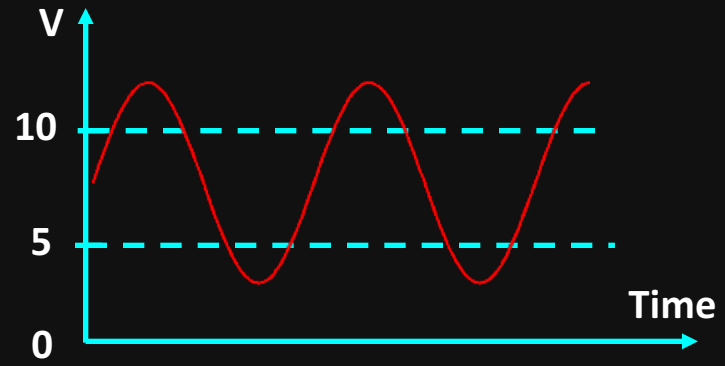
EXPLOITABLE ADC DESIGN CONSTRAINS

- Sampling frequency should follow Nyquist rule ($f_s > 2B$)
  - Otherwise the signal will appear of false (alias) frequency



EXPLOITABLE ADC DESIGN CONSTRAINS

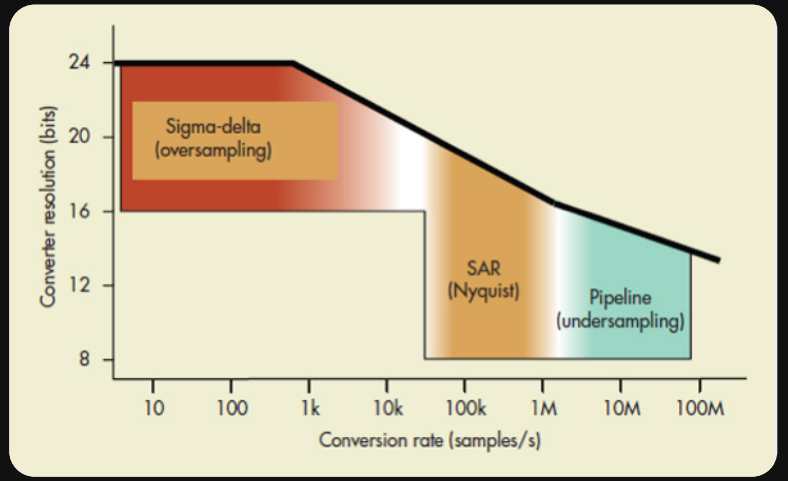
- Amplitude of the input signal should not exceed ADC's dynamic range
  - It is determined by the reference voltage



TYPES OF ADC

There are many ADC types (>10). The most common are:

- Successive-approximation ADC (SAR)
- Sigma-delta ADC
- Pipeline

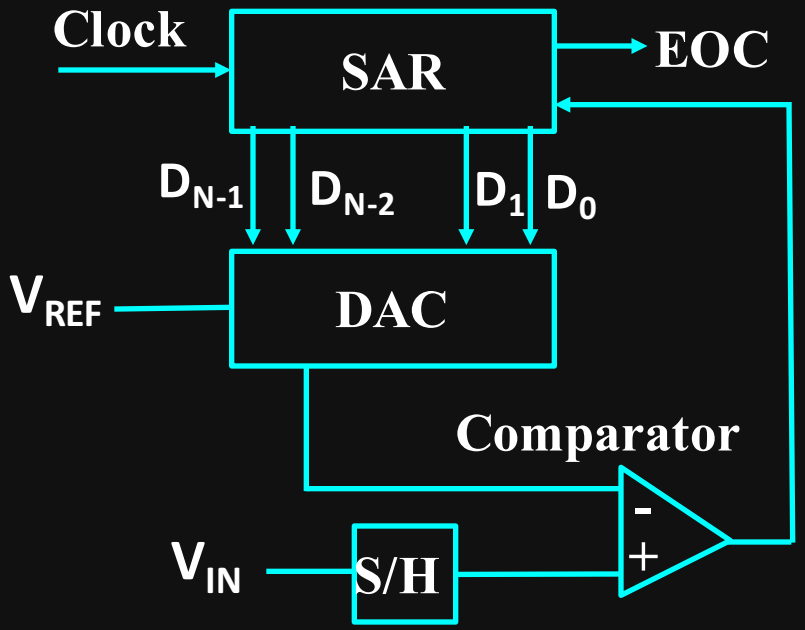


http://electronicdesign.com/analog/real-world-versus-your-adc

http://www.planetanalog.com/author.asp?section_id=3193&doc_id=561627

SUCCESSIVE APPROXIMATION REGISTER (SAR) ADC

BLOCK DIAGRAM



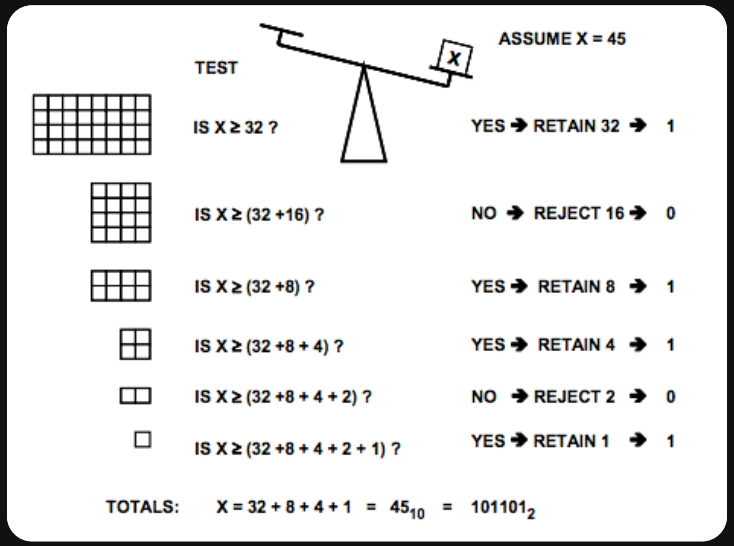
- DAC = Digital-to-Analog converter
- EOC = End of Conversion
- SAR = Successive Approximation Register
- S/H = Sample and Hold circuit
- V_{IN} = Input Voltage
- V_{REF} = Reference Voltage

https://en.wikipedia.org/wiki/Successive_approximation_ADC

SAR: WEIGHING PROBLEM

- SAR algorithm is based on one of the solutions to weighing problem by **Niccolò Fontana Tartaglia**, Italian mathematician and engineer in 1556

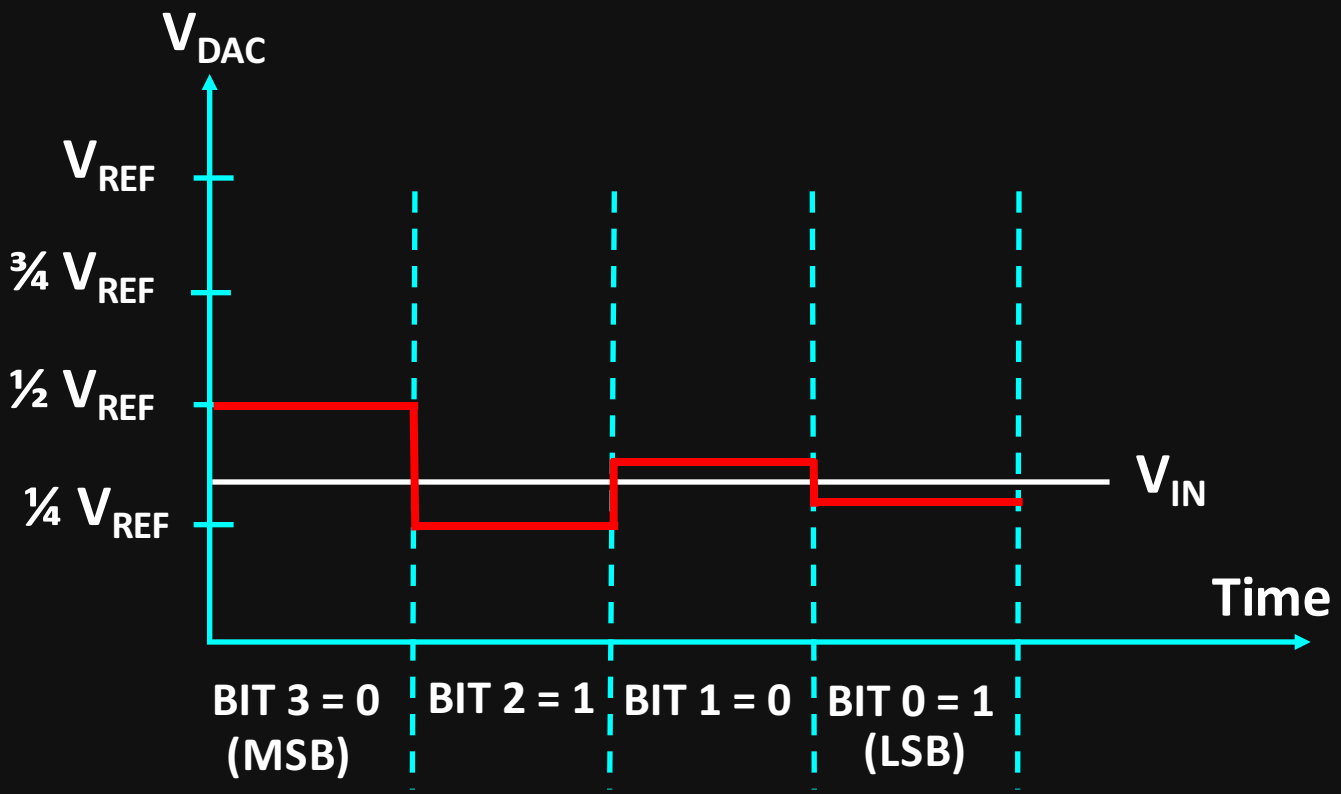
https://en.wikipedia.org/wiki/Niccol%C3%B2_Fontana_Tartaglia



- The objective is to determine the least number of weights which would serve to weigh an integral number of pounds from 1 lb to 40 lb using a balance scale

http://www.analog.com/media/en/training-seminars/tutorials/MT-021.pdf

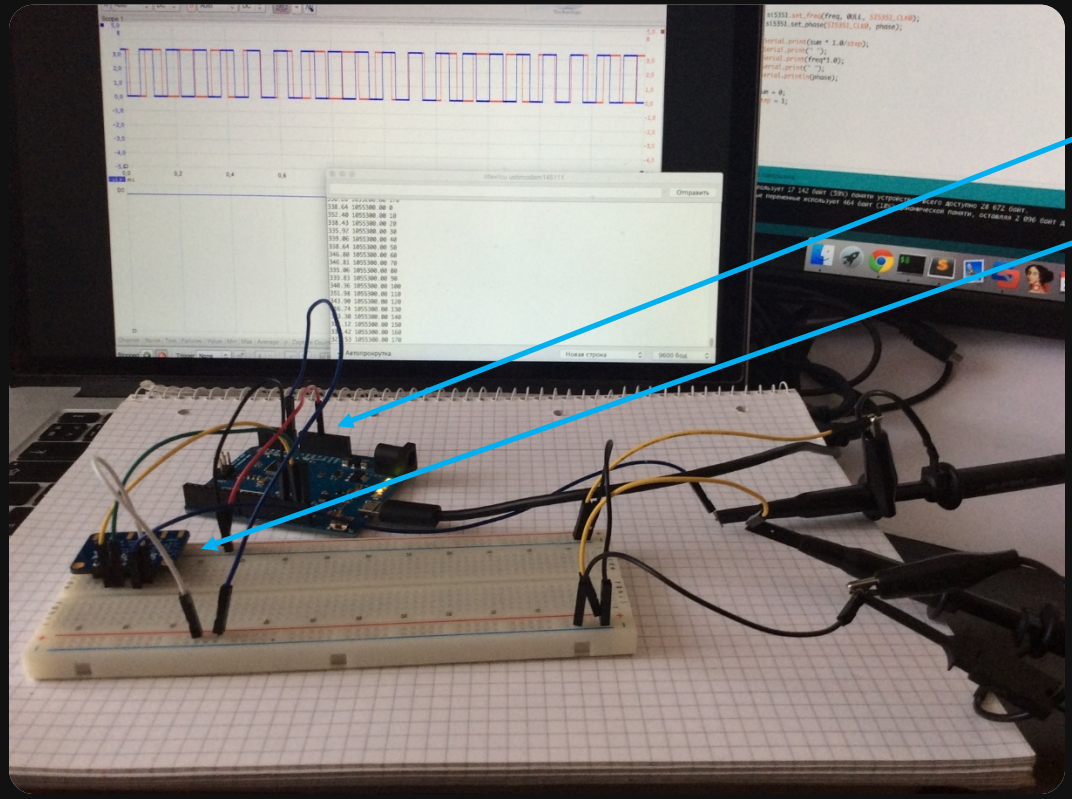
ADC: WEIGHING PROCESS



„Racing“ with ADC CLOCK

-- SAR ADC --

LETS SETUP EXPERIMENT



Experimental setup:

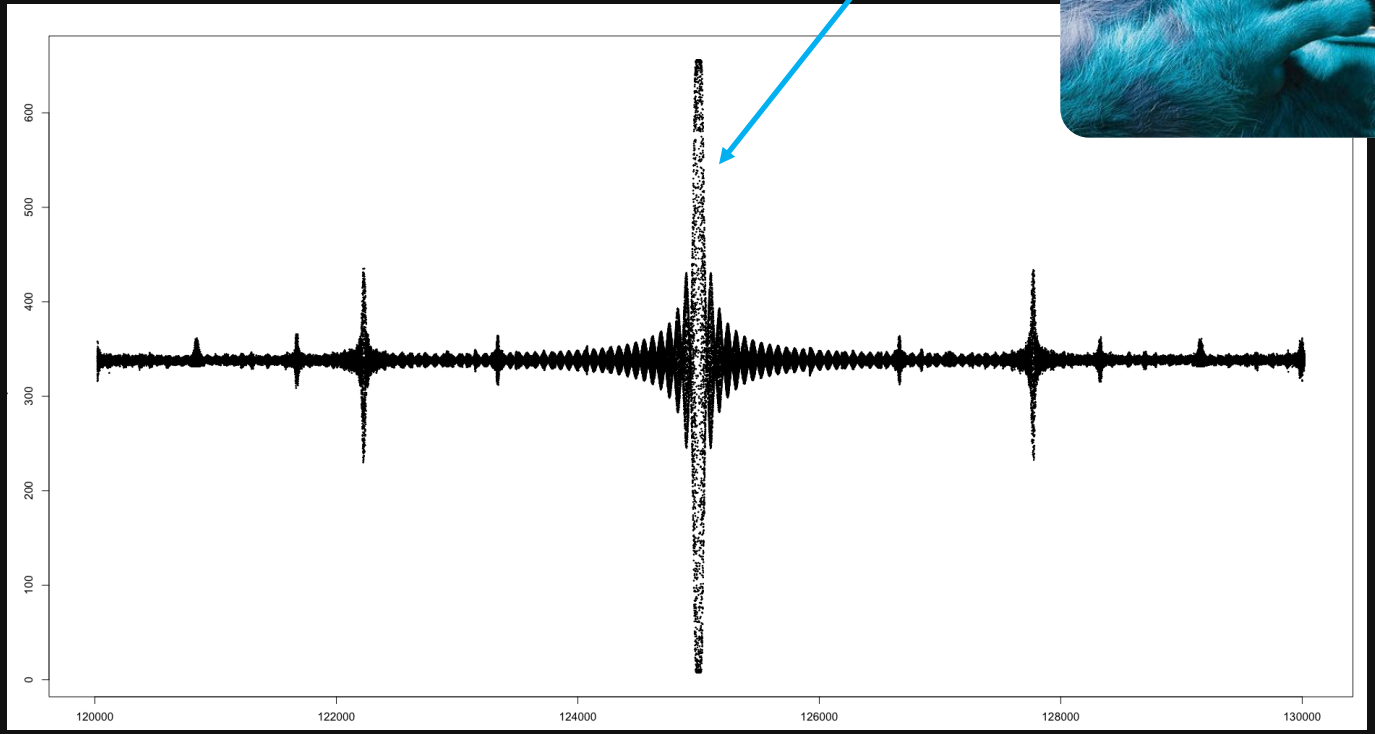
- Arduino Leonardo (Atmega32U4 with build-in ADC, 125kHz int clock)
- Si5351 generator

Algorithm:

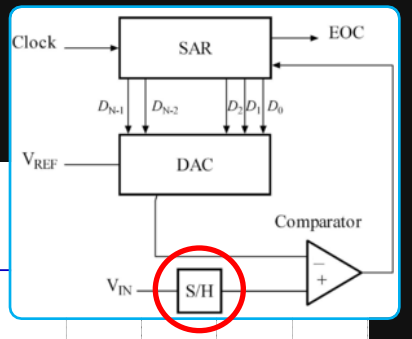
1. Generate square signal with specific frequency and phase,
2. Read 120 ADC values in row and average them,
3. Output to serial port (PC),
4. Increase phase and frequency,
5. GOTO 1.

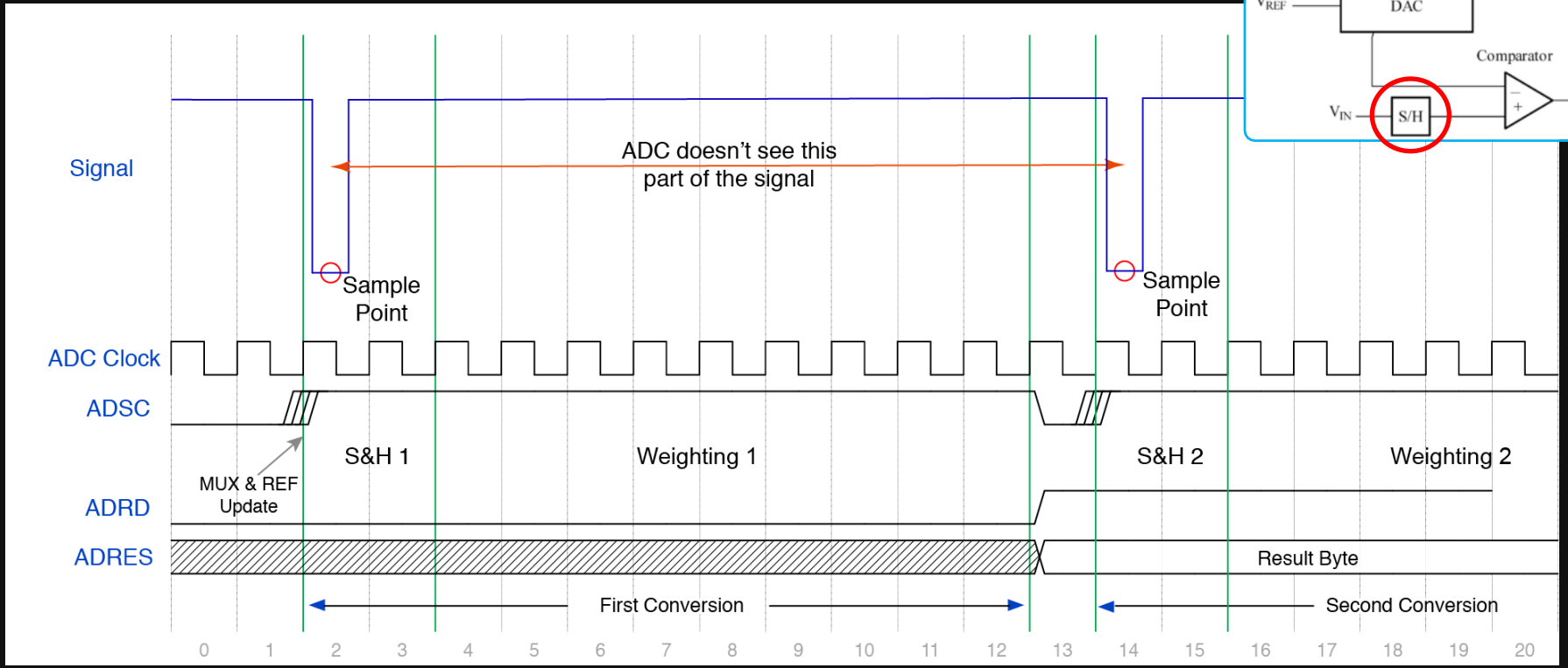
RESULT

What is this?!



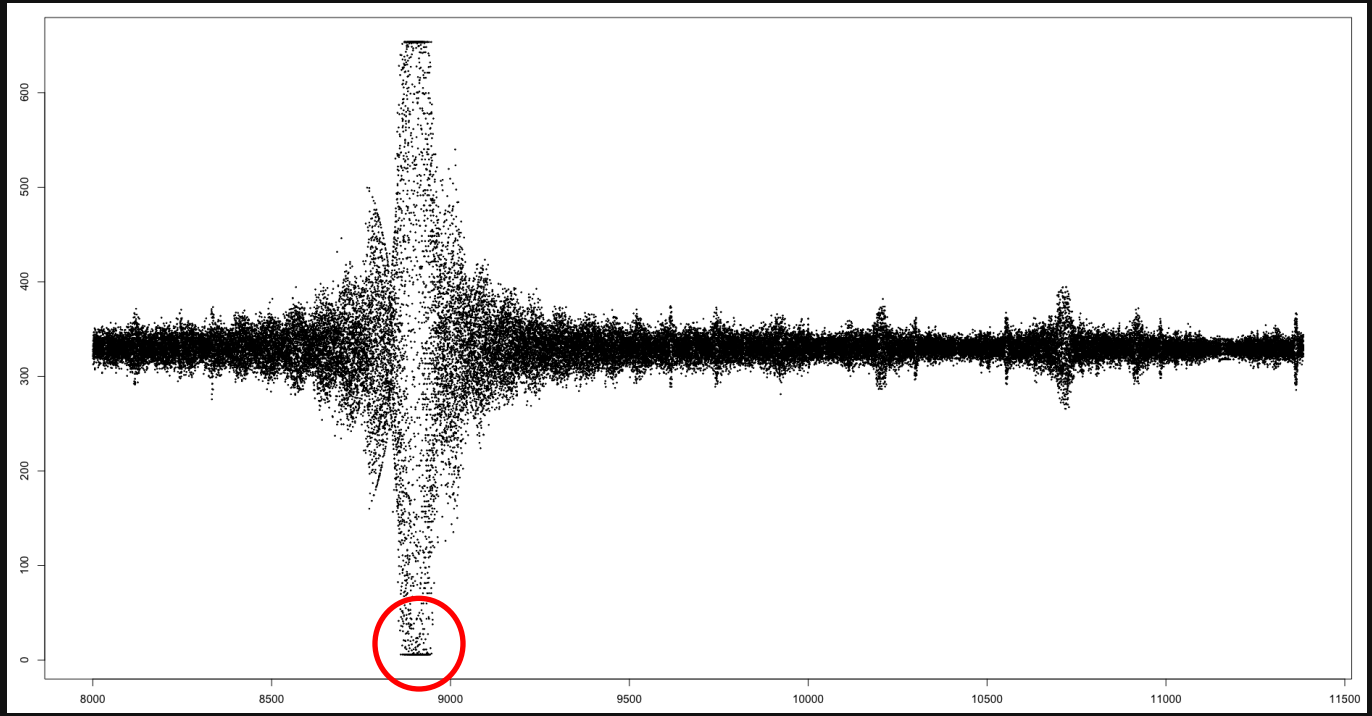
RACING WITH ADC CLOCK





LETS REPEAT OUR EXPERIMENT

Frequency = around 8.9kHz



LETS REPEAT OUR EXPERIMENT

Let's introduce "***counter***" to our code for averaging 120 ADC conversions:

```
for(;;){
  asm("cbi 0x0e, 6");
  val = __fastAnalogRead(A0); //inline function
  asm("sbi 0x0e, 6");
  sum += val;
  step++;

  if(step > 120){
    if(phase >= 170){
      phase = 0;
      freq += 100;
    }else
      phase += 10;

    si5351.set_freq(freq, 0ULL, SI5351_CLK0);
    si5351.set_phase(SI5351_CLK0, phase);

    Serial.print(sum * 1.0/step);
```

We're putting here an outgoing Zero-peak signal to see when ADC do actual work

Fast analog read

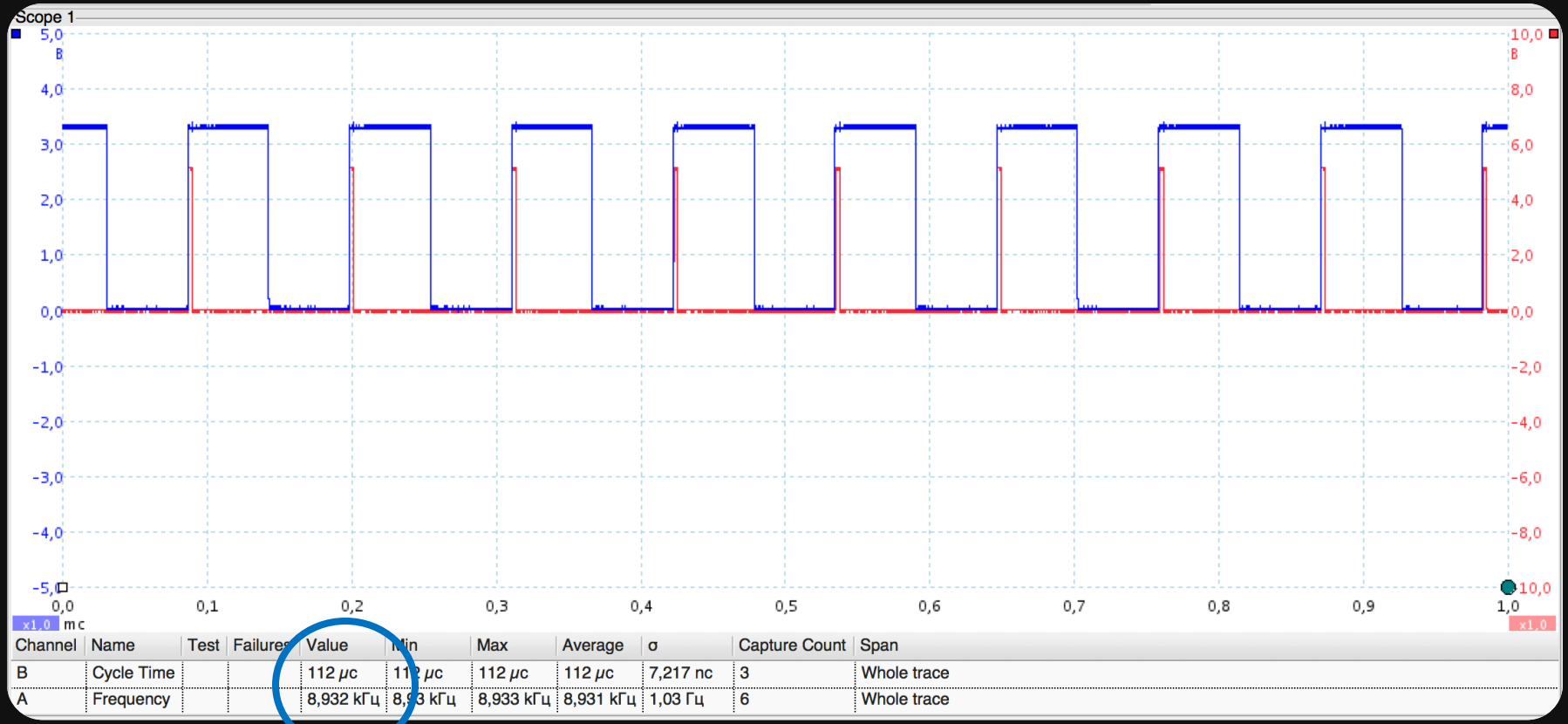
Average, frequency changing and out to serial port goes here

DEMO 2

LIVE DEMO

-- Explanation --

TIMING DIAGRAM EXPLAINS EVERYTHING



Channel	Name	Test	Failures	Value	Min	Max	Average	σ	Capture Count	Span
B	Cycle Time			112 µc	112 µc	112 µc	112 µc	7,217 nc	3	Whole trace
A	Frequency			8,932 кГц	8,933 кГц	8,933 кГц	8,931 кГц	1,03 Гц	6	Whole trace

FROM ATMEGA 34U4 DATASHEET

Chapter 24 on ADC , page 302

Table 24-1. ADC Conversion Time

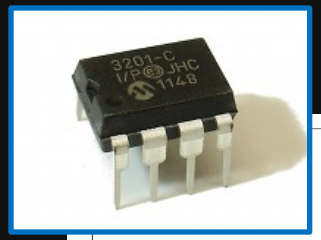
Condition	First Conversion	Normal Conversion, Single Ended	Auto Triggered Conversion
Sample and Hold (Cycles from Start of Convention)	14.5	1.5	2
Conversion Time (Cycles)	25	13	13.5

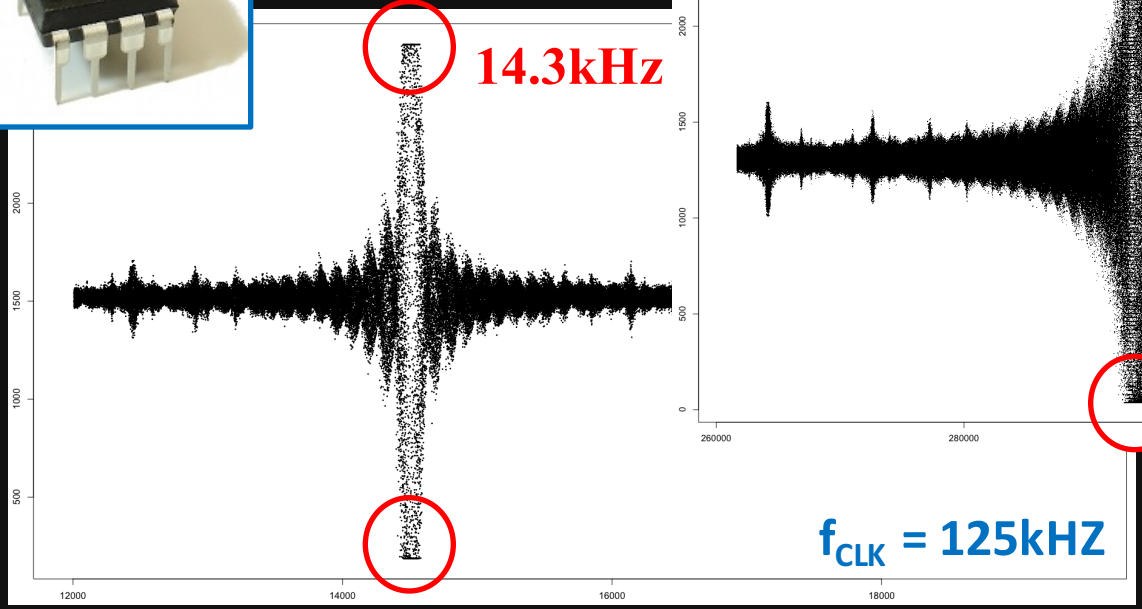
125kHz / 14 ~ 8928Hz (112µs)

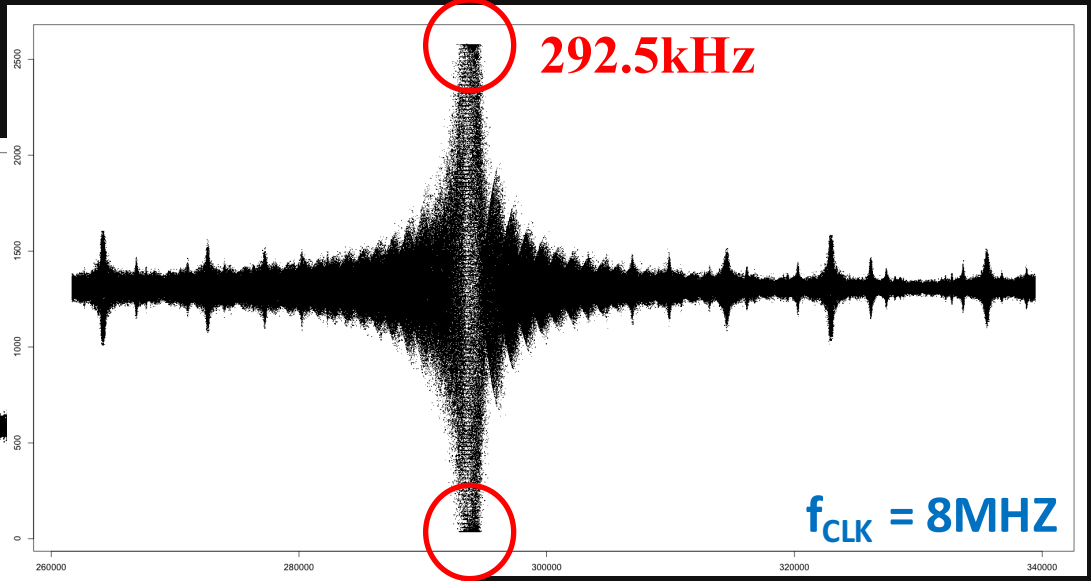
We've just breached through sampling rate precision of the ADC!

NOT ONLY BUILT-IN ADCS

Test results for MCP3201 MCU







SOFTWARE-RELATED PROBLEMS

-- ADC access timing --

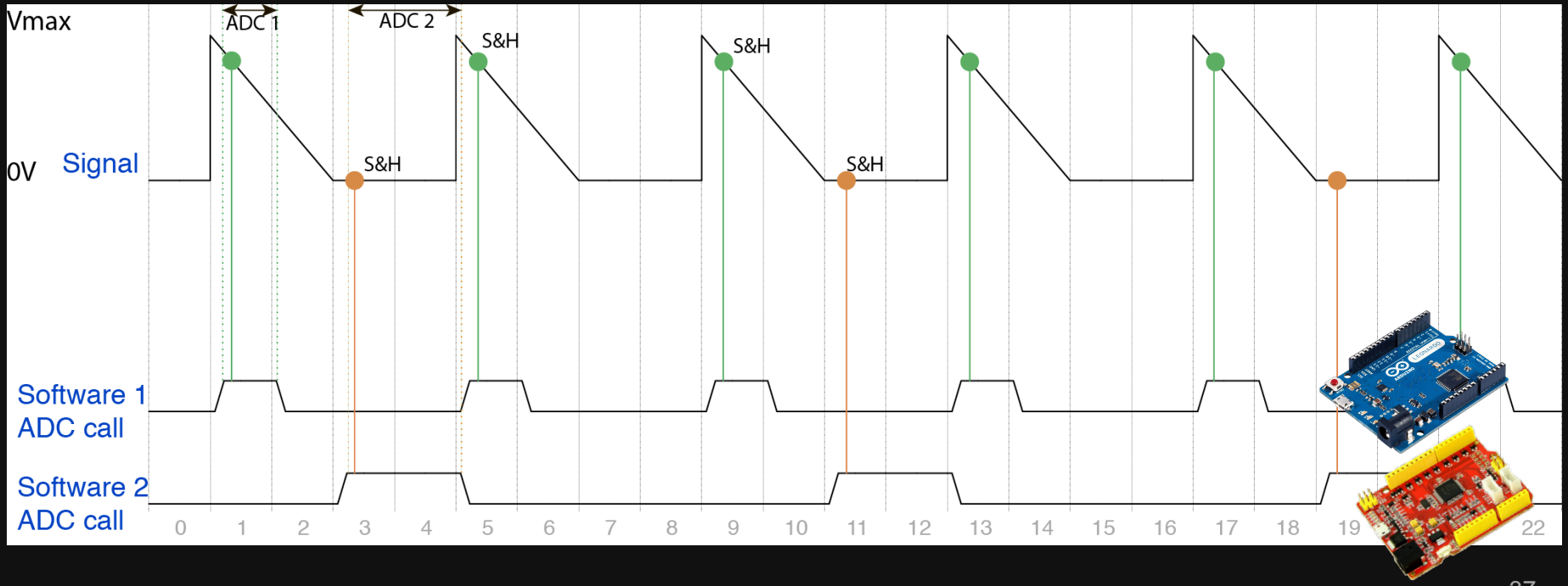
DEMO 3

DEMO VIDEO

-- One signal, two ADCs --

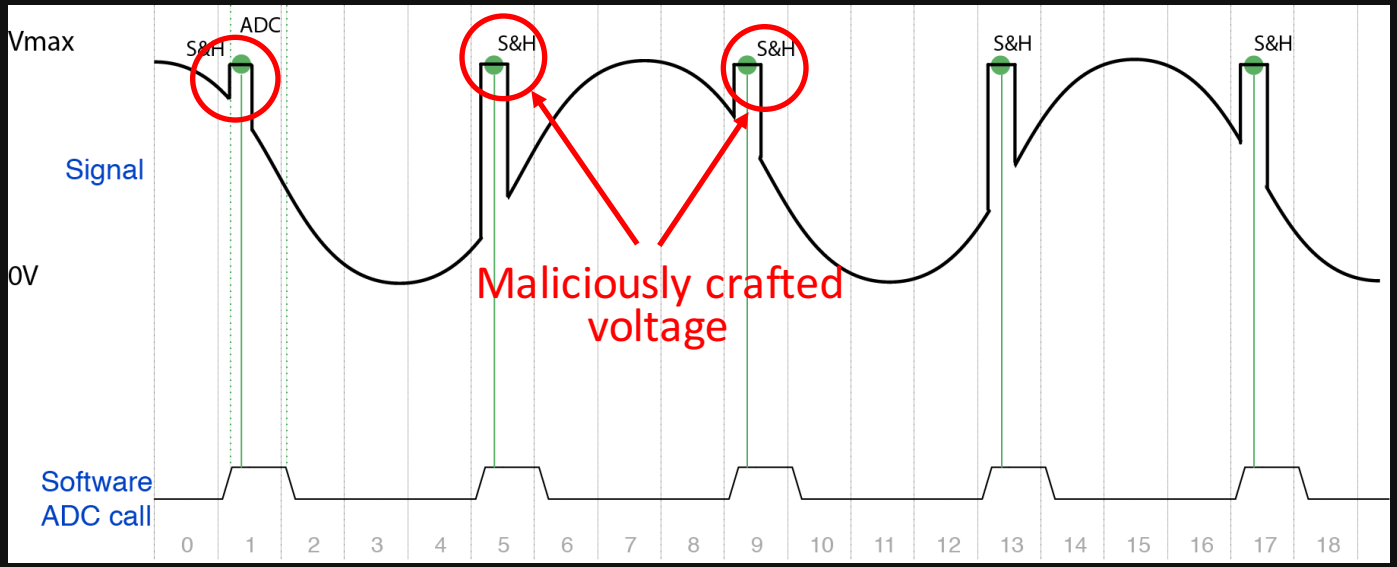
FROM DEMO: TWO DEVICES & TWO DIFF OUTPUTS

Wait, but why? Timing diagrams can explain ;-)



EVERYTHING IS MUCH EASIER IN THE ICS WORLD

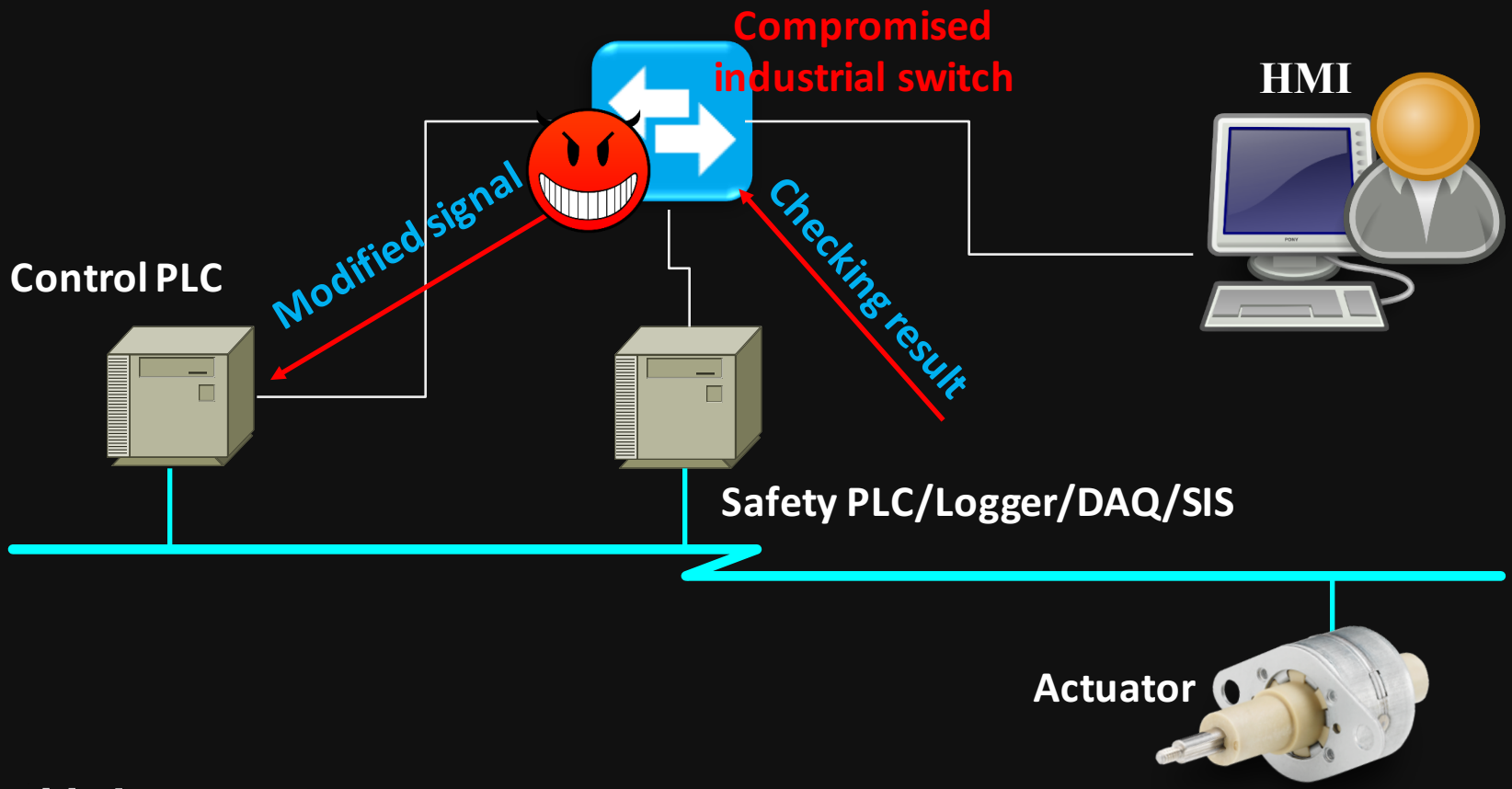
- In many real-world ICS applications ADC doesn't sample input signal with highest possible frequency
  - Typical sampling rate is 1-100 times per second



HURDLES OF THE ATTACKER

- **How to figure out the required phase and frequency to craft needed malicious signal?**
- Send some peak signals and monitor output of the ADC (directly/indirectly)
- E.g. by hacking into switch you can monitor/control both data flow to control PLC **AND** signals from SIS/Safety LC/logger/DAQ/etc

FIGURING OUT SIGNAL PARAMETERS



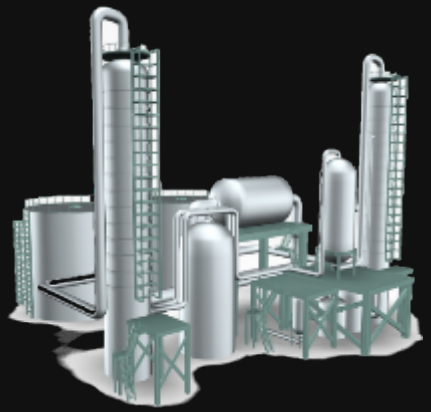
SOFTWARE-RELATED PROBLEMS

-- ADC conversion time --

ADC IN CRITICAL APPLICATIONS

Be careful when using ADC in critical applications

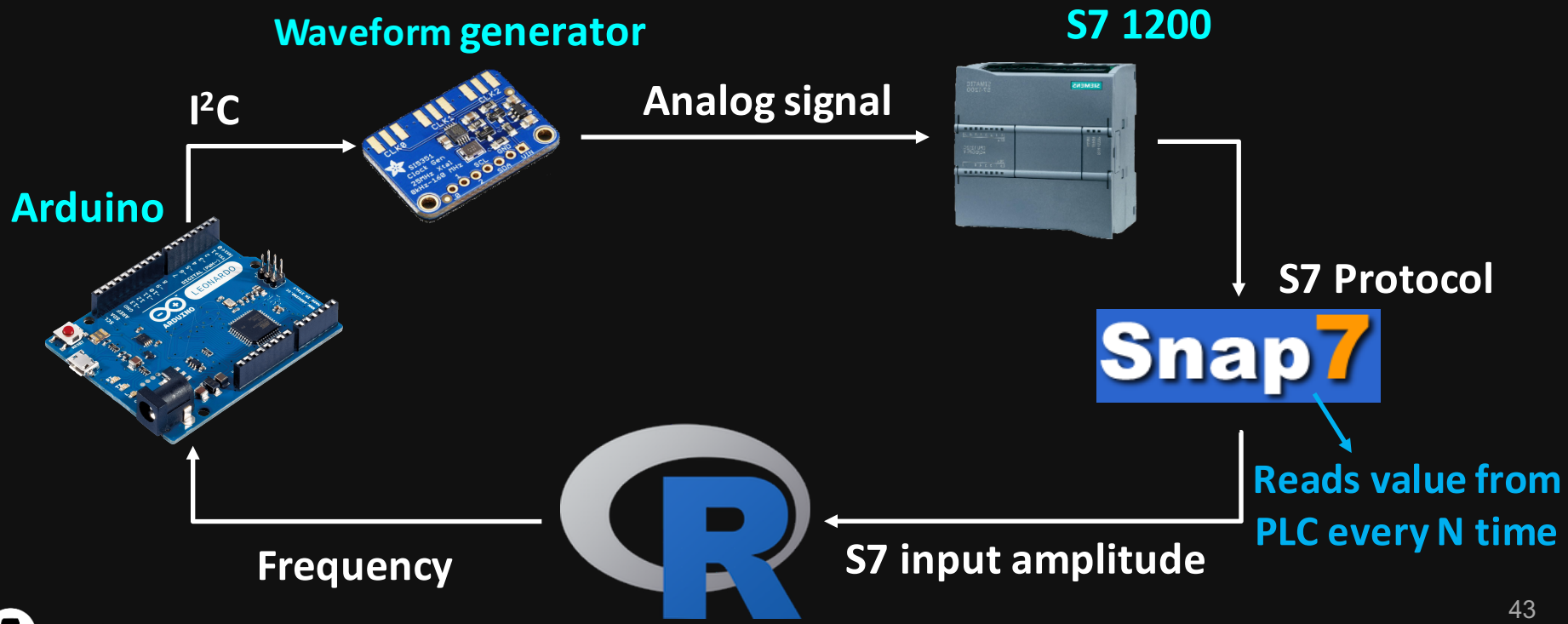
- Industrial PLCs also have analog inputs and built-in ADCs
- Let's test at one of the most popular PLCs S7 1200 µ



Analog value creation	
Integrations and conversion time/resolution per channel	
Resolution with overrange (bit including sign), max.	10 bit
Integration time, parameterizable	Yes
Conversion time (per channel)	625 µs

EXPERIMENT SETUP

Let's check the real conversion time of S7 1200 ADC



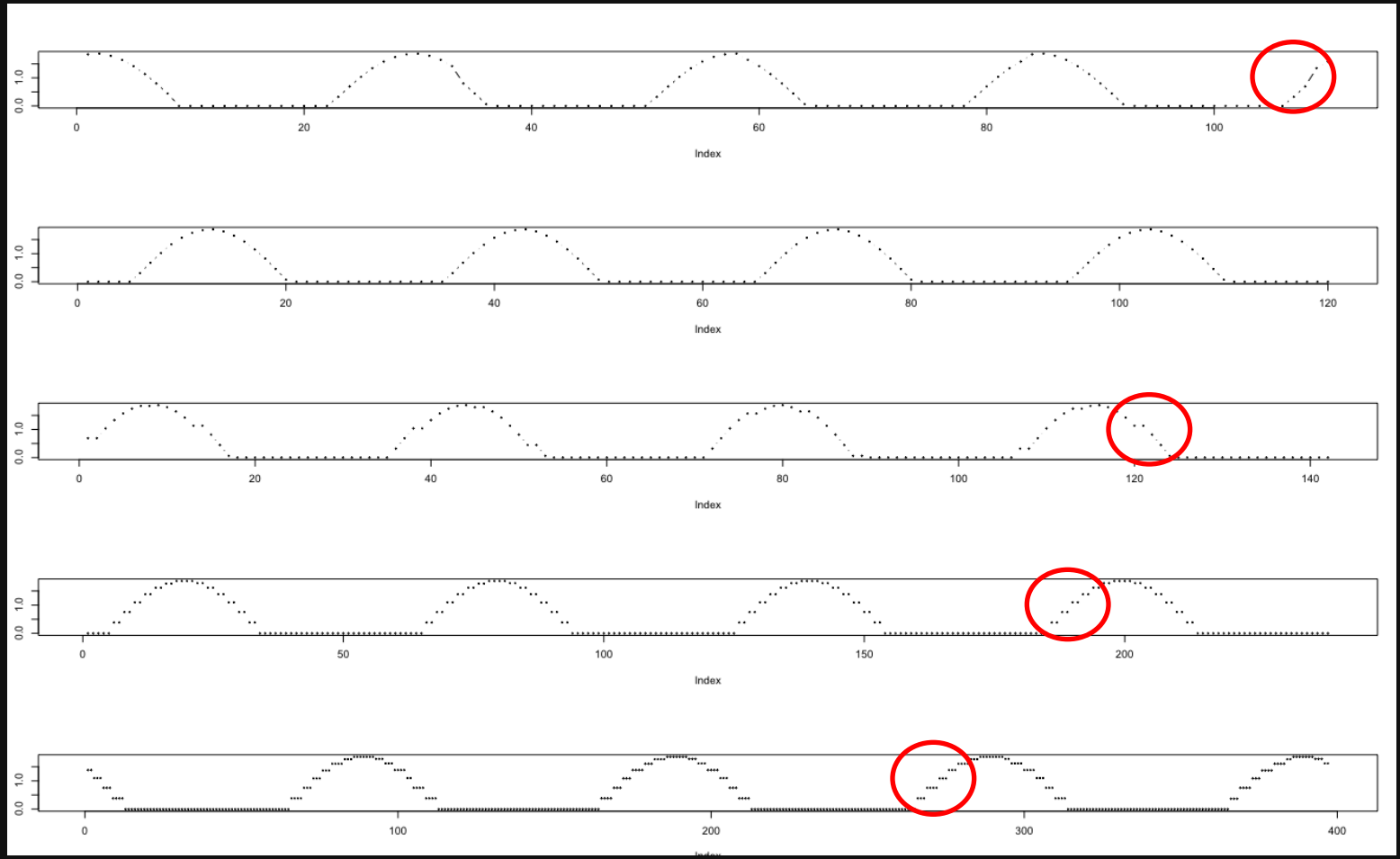
N=9ms

N=8.3ms

N=7ms

N=4.5ms

N=2.5ms

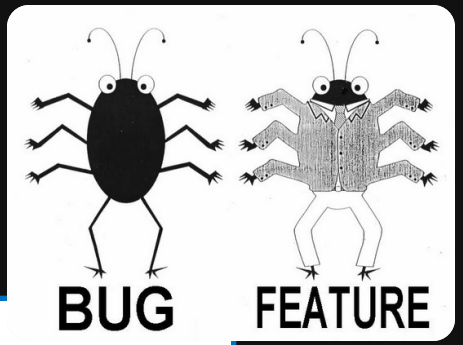


Frequency is fixed


Our ADC Conversion time is just 625uS
I could see non-repeating values only every 8.3mS
BUT...

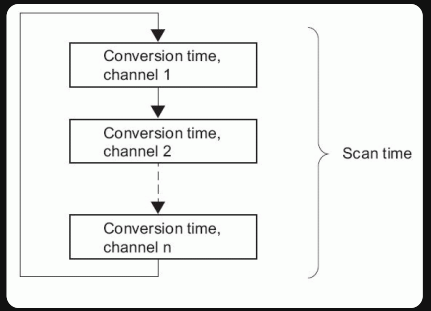
WHAT'S WRONG?

Nothing, really. You just need to read datasheet more thoroughly



Conversion time:
The conversion time is the sum of the time required by the A/D converter (analog-to-digital converter) to record the measured value (basic conversion time), as well as time taken (diagnostics, open-circuit monitoring) to process the measured value in the module (processing time).

Cycle time:
The cycle time, which is also called the sampling time, is the time between two conversions on the same channel.

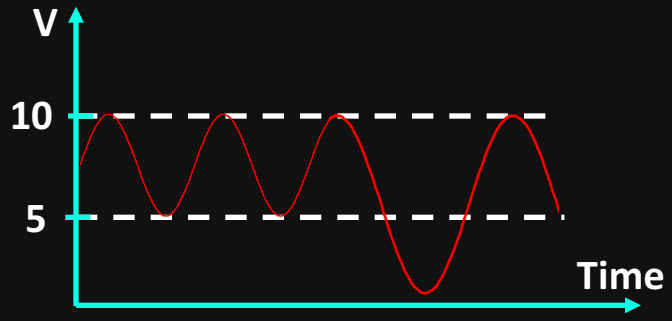


Text in small letters

INVALID RANGE OF SIGNALS

BREAKING SOFTWARE DEFINED RANGES (I)

- Consider a 5-10V signal which is consumed by ADC with ranges 0-15 V
- What will happen if you send signal lower than 5V or higher 10V?



From the real life code:

```
uint8_t val = readADC(0); // reading 8-bit ADC value with ranges 0V -15 V
val = val - 85; // Normalization -> 85 == 5 Volts (255/3)
```

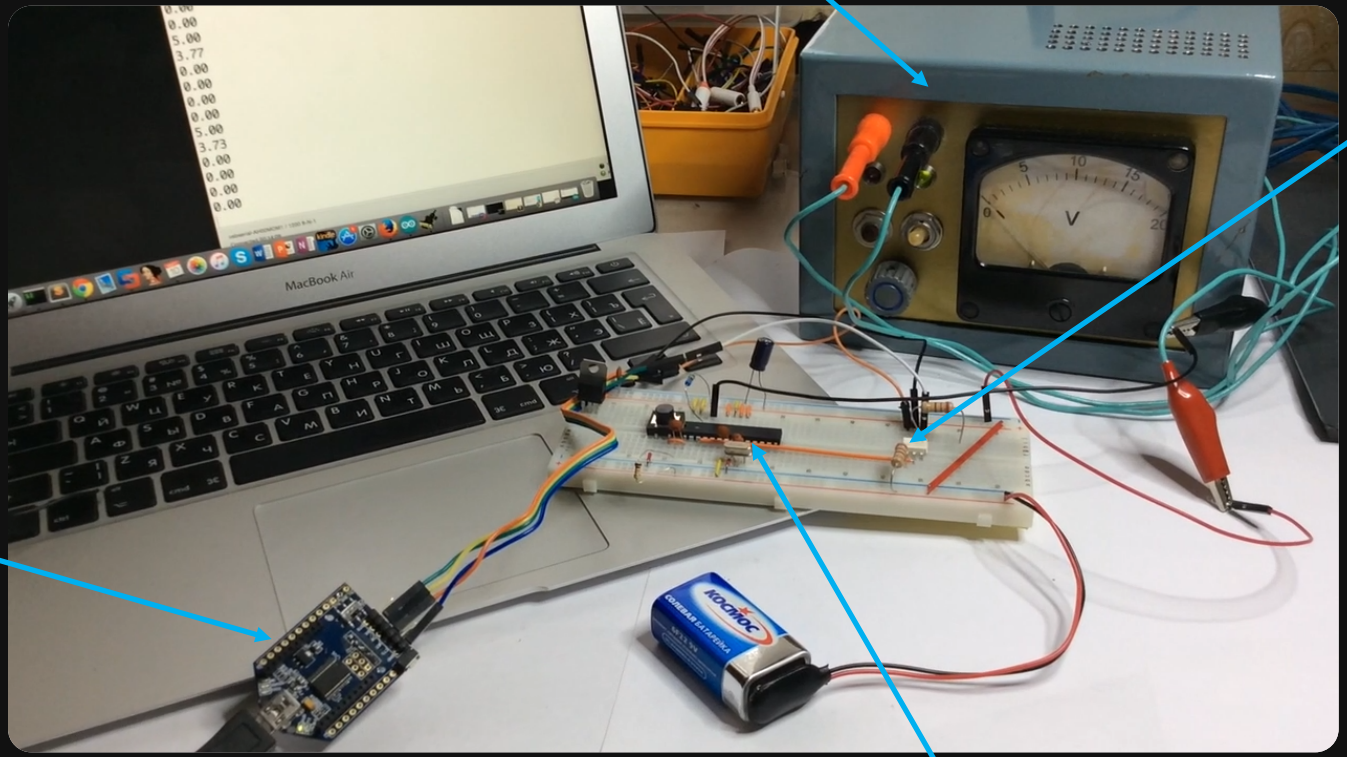
Any signal of less them 5 V `(val < 85)` will cause **integer overflow** in `val`

BREAKING SOFTWARE DEFINED RANGES (II)

What if the attacker sends signal outside of the ADS hardware defined range (>Vref)?

- ADC will output max value (all bit set to 1)
- ADC might be damaged (did not test out of cost factors ☺)
- Values **on other inputs** could be distorted

DEMO SETUP


Negative Power source
Optical Isolator
USB UART
Atmega328p
MacBook Air

DEMO 4

DEMO VIDEO

-- Negative input signal --
(breaking hardware range)

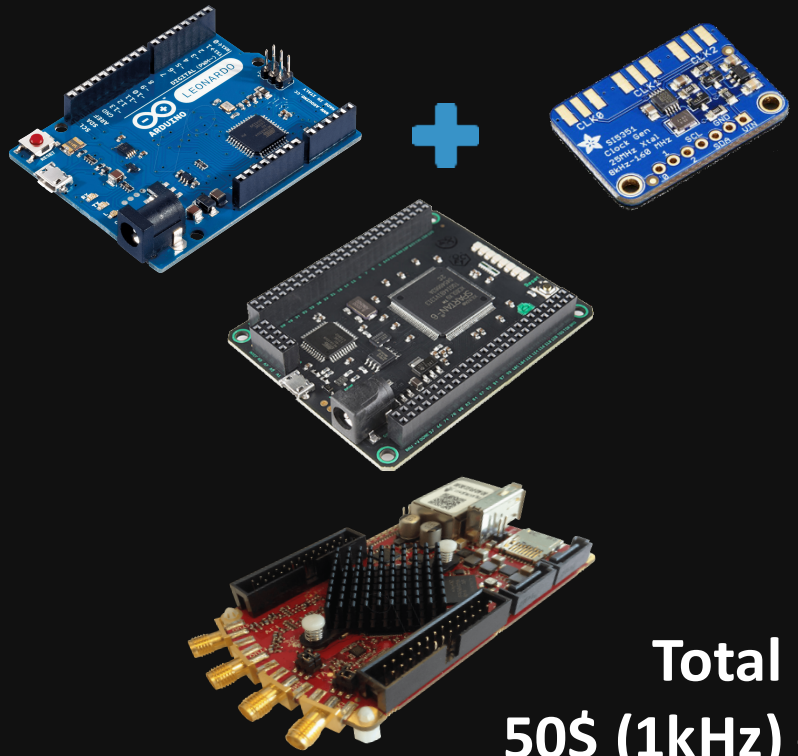
ANOTHER EXAMPLE

Breaking HW RANGES for NXP LPC 11U24F internal ADC (3.3VRef)

ADC/Ref	Volts	A_{-3}	A_{-2}	A_{-1}	A_{-0}	A_{+1}	A_{+2}	A_{+3}
NXP LPC 11U24F (3.3VRef)	0.48			0.0	0.48	1.58	3.3	
	3.39			0.0	3.3	1.59	3.3	
	4.1			0.087	3.3	1.729	3.3	
	4.65			0.17	3.3	1.974	3.3	
	5.1			0.44	3.3	2.212	3.3	
	5.9			0.0	2.035	1.561	3.3	
	6.1-9.8			~	~	~	~	
	-0.48			0.0	0.0	1.58	3.3	
	-1.1			0.0	0.0	1.64	3.20	
	-1.5			0.025	0.0	1.71	3.07	
	-1.7			0.0	0.0	2.5	2.9	
	-2			~	~	~	~	

ATTACK VECTORS IN ICS

DIRECT ACCESS ATTACK TOOL KIT



Line coupling circuit
(usually OpAmp/Transformer)

Total setup cost
50$ (1kHz) -- 400$ (50MHz)

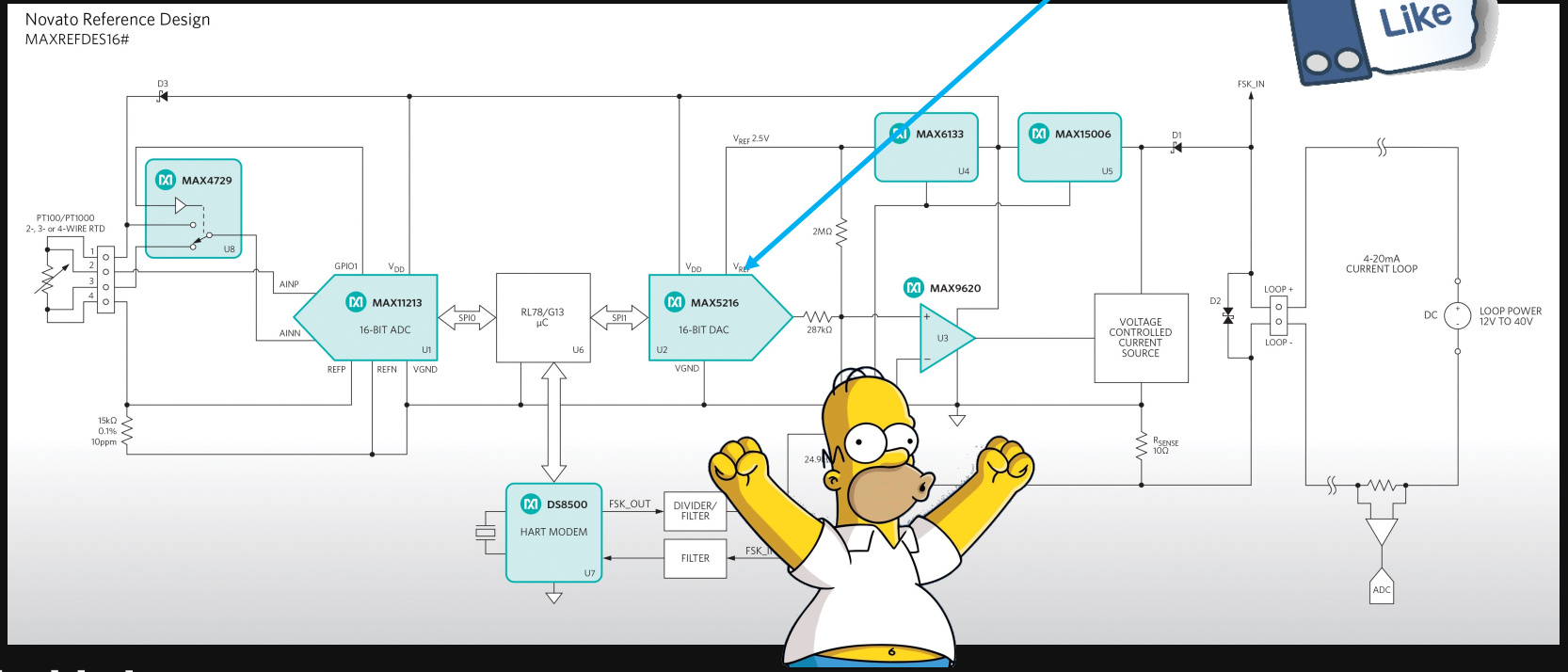
ATTACKING FROM ICS DEVICE

- Compromising one of the field components (PLC, sensor, actuator, DAQ, logger, etc.)
  - Most MCUs inside transmitters/actuators are capable of generating arbitrary signals up to 500-1000Hz
  - Some devices allow to generate signals of 44kHz and above

ATTACK FROM TRANSMITTER

HART transmitter reference design ;-)

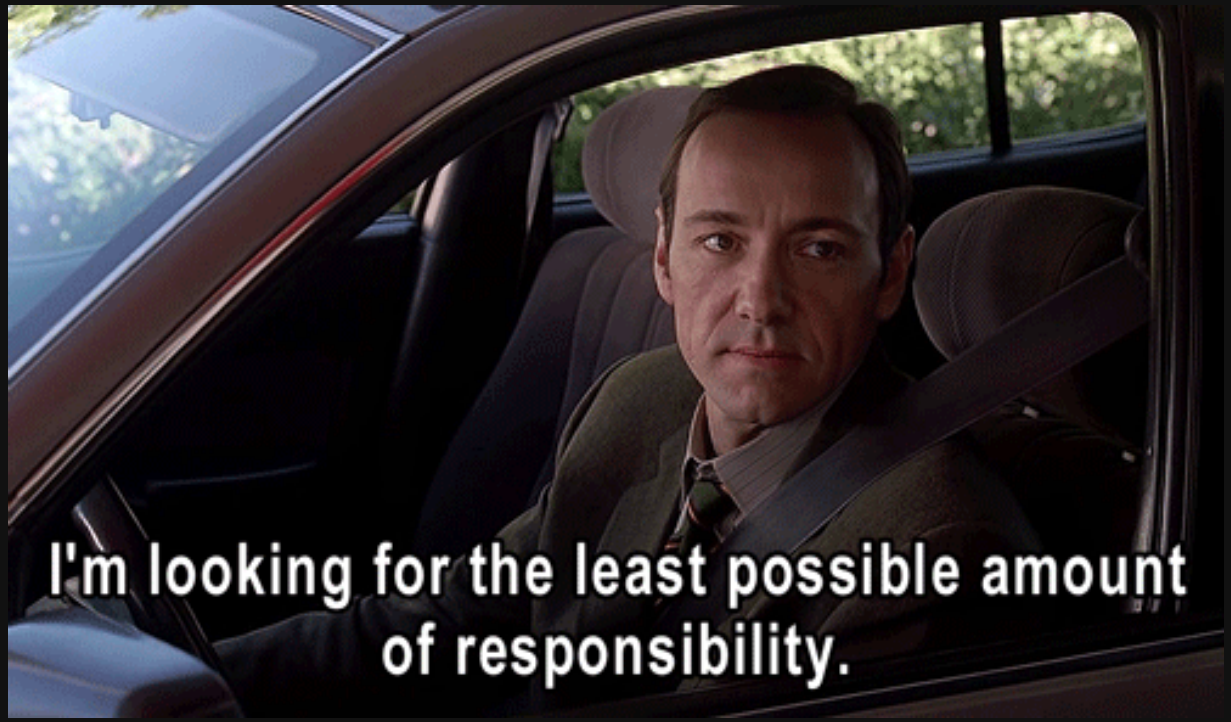
DAC with s/r up to 100kHz
(smooth sine wave at ~ 5kHz)



http://www.tm-eetimes.com/en/accurate-industrial-temperature-measurements-with-loop-powered-transmitter.html?cmp_id=7&news_id=222918850

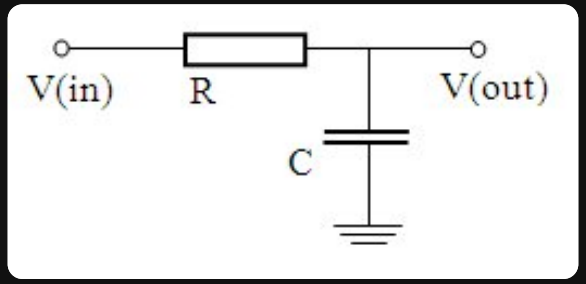
MITIGATIONS

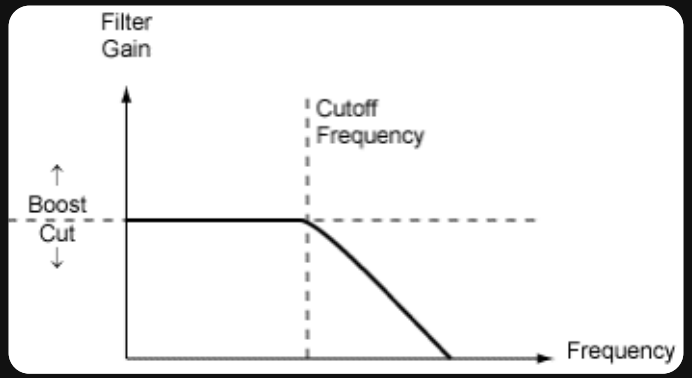
HARDWARE MITIGATIONS



LPF FILTERS IN REFERENCE DESIGN

- Low-pass filter rejects signals with a frequency higher than its cutoff frequency
- Buffer ADC input with LPF
- Good design dictates ADC f_s >= LPF f_c

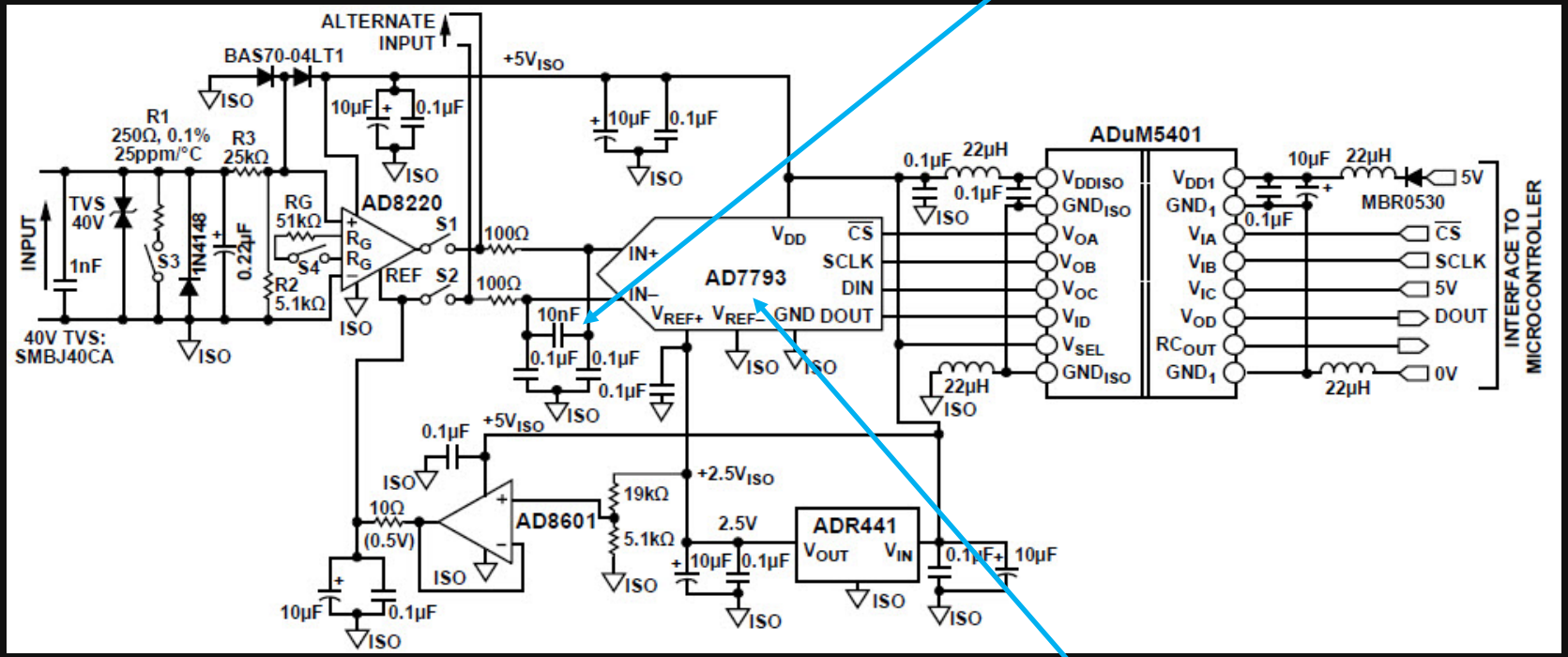




LPF FILTERS IN REFERENCE DESIGN

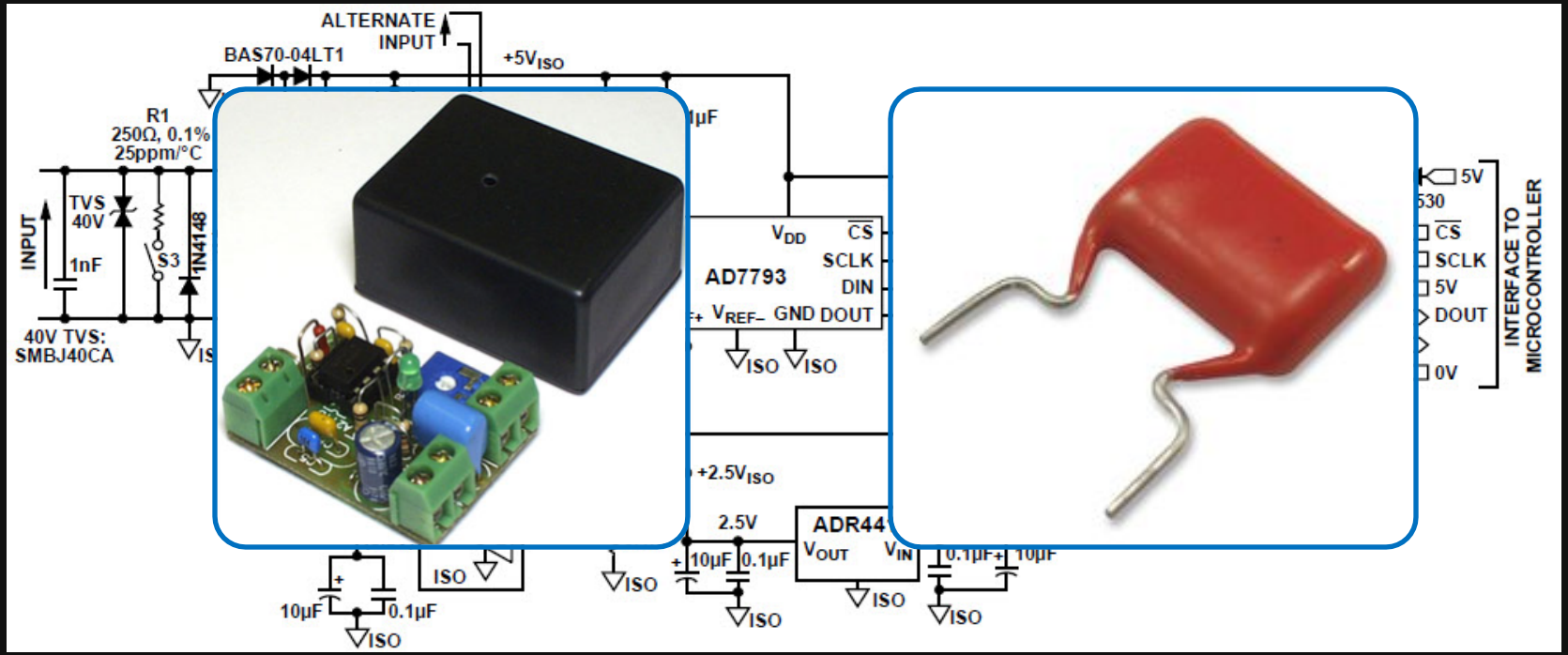
"We included LPF in our design"

LPF with f_c near 15 kHz



ADC with f_s > 470Hz

SOLUTION


ALTERNATE INPUT
BAS70-04LT1
+5V$_{ISO}$
R1
250Ω, 0.1%
25ppm/°C
TVS
40V
1nF
INPUT
S3
1N4148
40V TVS:
SMBJ40CA
AD7793
V_{DD}
CS
SCLK
DIN
V_{REF-}
GND
DOUT
ISO
+2.5V$_{ISO}$
2.5V
ADR44
V_{OUT}
V_{IN}
10µF
0.1µF
5V
CS
SCLK
5V
DOUT
0V
INTERFACE TO MICROCONTROLLER

ACHIEVING ADC ZEN

FLIP SIDE OF USING LPF

”Securing” may lead to more vulnerabilities

- When adding LPF into an individual device, make sure that all related devices have the same cut-off frequencies
- E.g. if PLC input is buffered with LPF $f_c = 1kHz$ and actuator equipped with LPF with $f_c = 5kHz$, the attack not only possible, but the **probability of success increases**!

NOTE: DIGITAL LPF WON'T WORK!

Do not use digital LPF after the ADC!

- ADC will be already compromised by an ill-intended signal and no digital filter will fix the matters

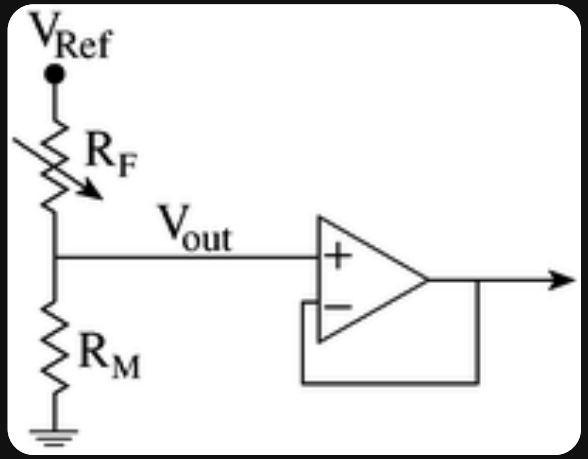
USE ADC WITH HIGHER BANDWIDTH/LOWER CONVERSION TIME

- Using ADC with higher sampling frequency can mitigate "oversampling" attack as the attacker will have to generate signal of much higher frequency
- Generating >1MHz signal and injecting it into analog line is much harder than injecting < 1MHz signal
  - H/f signals subjected to greater attenuation and more affected by noise

SCALE SIGNAL AMPLITUDE BEFORE ADC

- To avoid abuse of ADC ranges, normalize signal amplitude before feeding the signal to ADC
  - Simplest option: voltage divider + OpAmp,
  - Signal conditioning circuits or even dynamic range compression

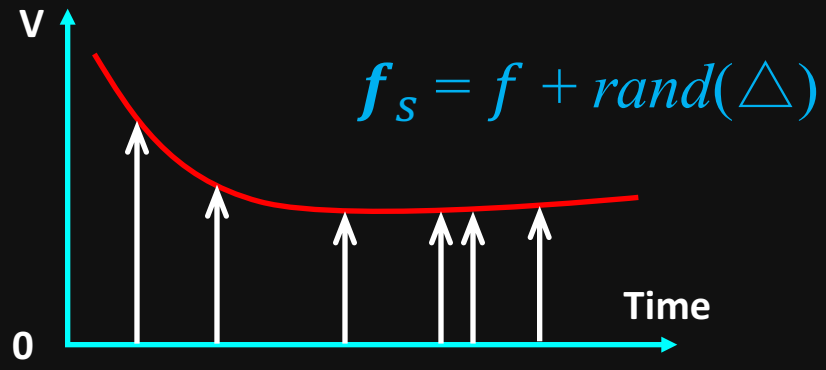
Select what is suitable for your OT process



SOFTWARE MITIGATIONS

SAMPLING FREQUENCY RANDOMIZATION

- Certain randomness in sampling frequency will make attacker's job much harder
  - Many of the discussed attacks will be much more challenging to execute
- Small variation of f_s won't degrade conversion process. On the contrary, it will produce a signal sample of better quality.



APLY SECURE CODING TECHNIQUES

- Scrutinize your ADCs/PLC datasheets to figure out **effective** ranges, conversion time, frequency and other critical parameters
- Even if it is sufficient to control the process with one value per second, sample the signal with higher frequency and average converted values
- When receiving value from ADC, treat it as an absolute value (all bits received from ADC are significant)

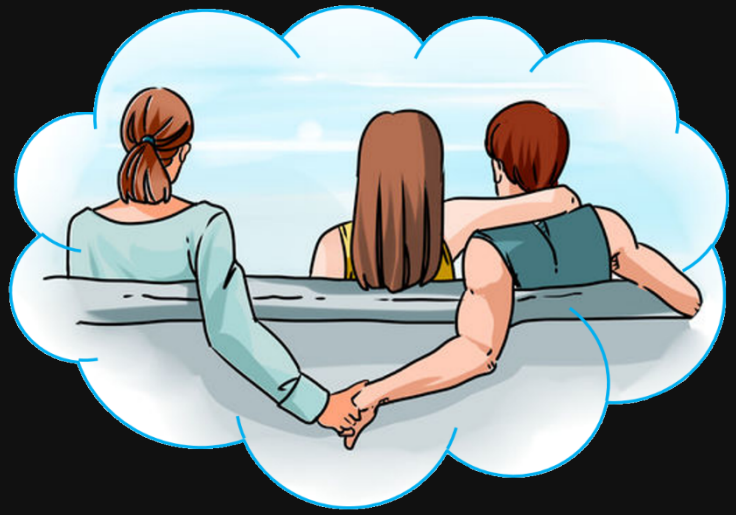
DON'T SLEEP! (WHILE ON DUTY ☺)

Avoid writing/using the following code (if you don't completely understand your process and aren't completely sure about what you are doing)

```
Val = readADC();
Output(Val);
Sleep(Timeout);
```

OT AND IT HAVE COMMON PROBLEMS

NEVER TRUST YOUR INPUTS




IOActive
Hardware | Software | Wetware
SECURITY SERVICES
1886
Honeywell
black hat
ASIA 2016
Any Q?
@dark_k3y
@marmusha
UBM

„OVERSAMPLING“ OF ADC CLOCK

-- Delta-Sigma ADC --

DELTA SIGMA ADC

MODUS OPERANDI

LETS SETUP ANOTHER EXPERIMENT

RESULT

EXPLANATION

ATTACK EFFORTS: SIGMA-DELTA VS. SAR