A Post-Attack Recovery Architecture for Smart Electricity Meters

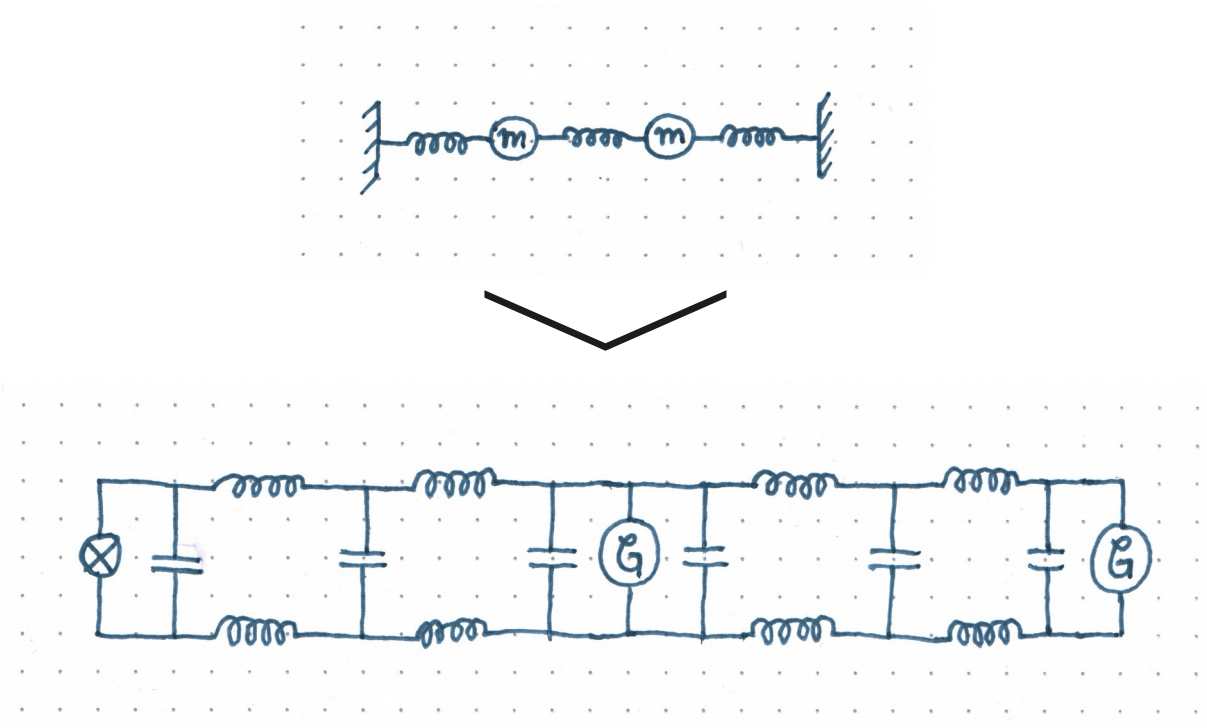
Jan Götte <master@jaseg.de>

Fundamentals

The Structure of the Electrical Grid

- Generators
- Transmission Lines
- Switchgear
- Transformers
- Loads

The Structure of the Electrical Grid



Smart Meter Functionality

- High-resolution
  Load measurement
- Load switching
  → Demand-Side Response
- Disconnecting “Delinquent” customers
- Smart home gateway

Smart Metering Incentives

- **Better load forecasting for a changing energy market**
  - Renewable Energies increase volatility
  - EV charging amplifies load spikes
- **Profit maximization**
  - Variable tariffs pass through costs
- **Selling data**
- **Cronyism**

Smart Metering Regulation

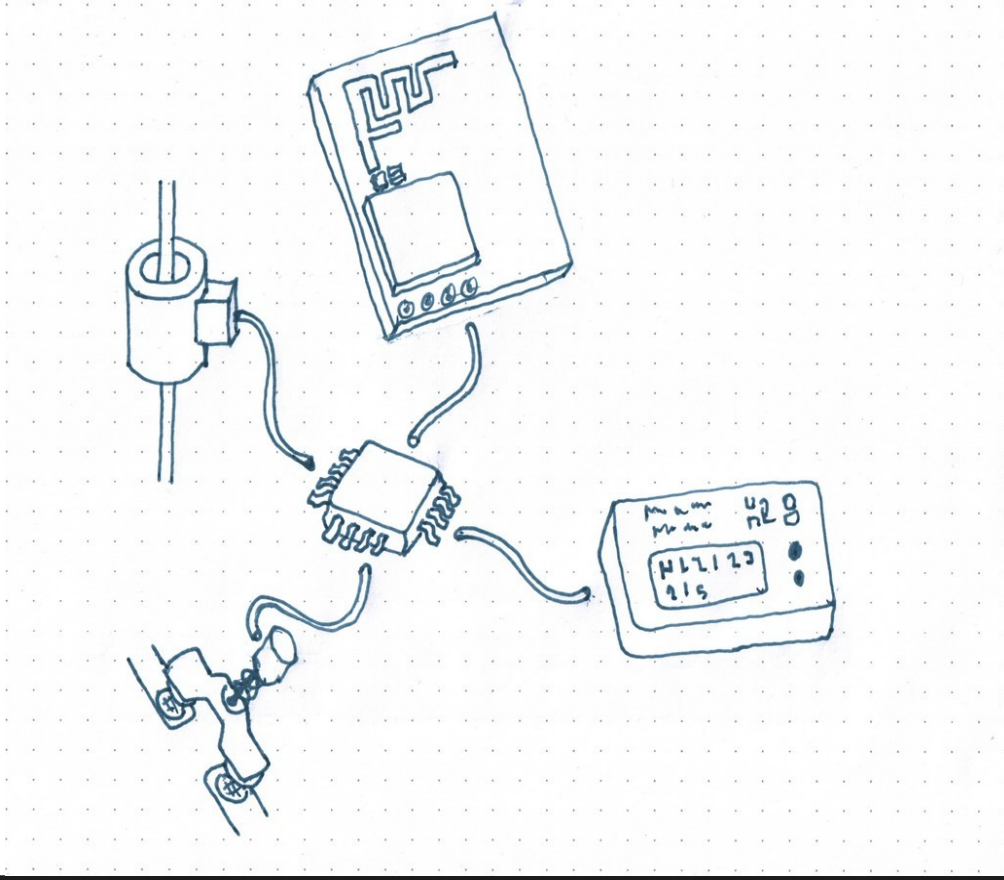
- Multiple competing international standards
- Sometimes no standards at all
- Degree of standardization is variable

➧ IEC 62056 family slowly subsumes national protocol standards

Smart Meter Technology

- Measurement Transformer
- Application Microcontroller
- Modem
- Load switches
- Display

➧ **Meters in DE are radically different from those in rest of the world: In DE Modems are external devices!**



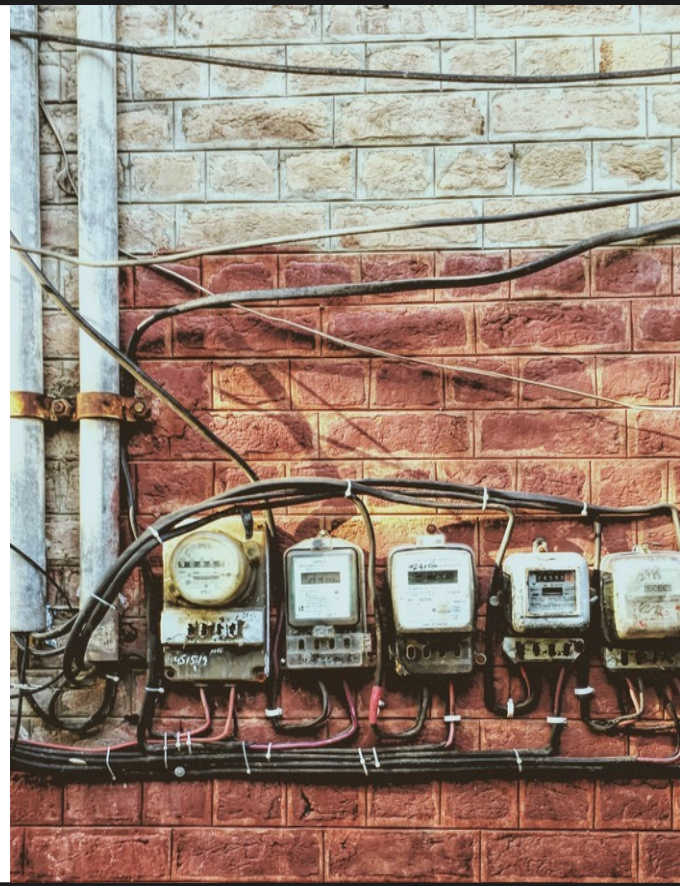
Security in the Distribution Grid

- Large-scale SCADA systems
- Networked
- Physical security is challenging
- Compatibility with decades-old equipment is required!

Endpoint Safety & Security

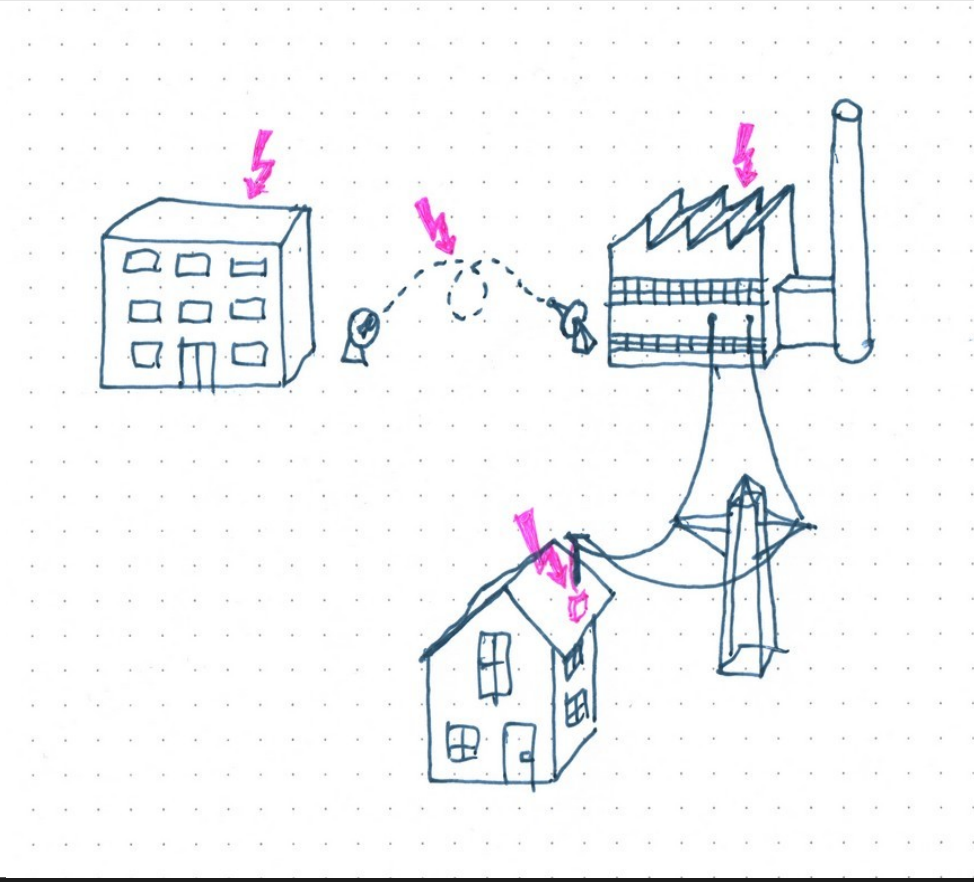
Attacker Prototypes

- **Customers: Electricity theft**
  - Also sold as a service by organized crime
- **Bored teenagers**
- **State actors**



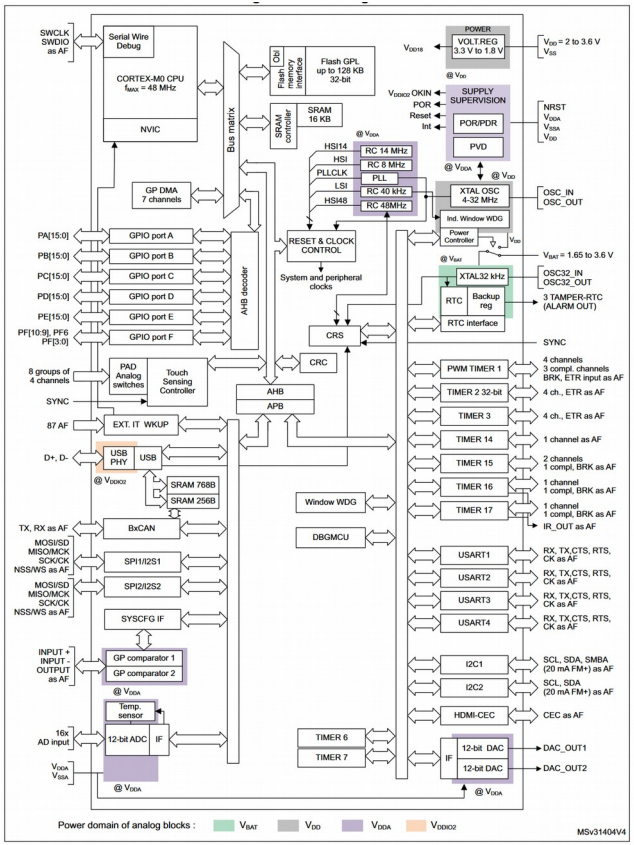
System structure and security

- Exploiting centralized control
- Communication channels exploits
- Control function exploits
- Endpoint exploits



Complex hard- and firmware

- **The line between Microcontroller and System-on-Chip is blurring**
  - DMA is ubiquitous
  - MMUs or MPUs are common
  - Coprocessors and Enclaves can be found in both
- **Complex HW/FW bundles are integrated**
  - Most common: radio modems
  - Also: AI accelerators
  - Also: Complex sensors (e.g. camera/barcode)



The State of Firmware Security

- **Firmware is everywhere**
- **Firmware is *haaard***
  - Meter Vendor Landis+Gyr spend 36% of their R&D budget on code
- **The state of embedded security**
  - Everybody fails: Apple, Samsung, Microsoft, Google
  - µCs lack many modern security features

The Safety Reset

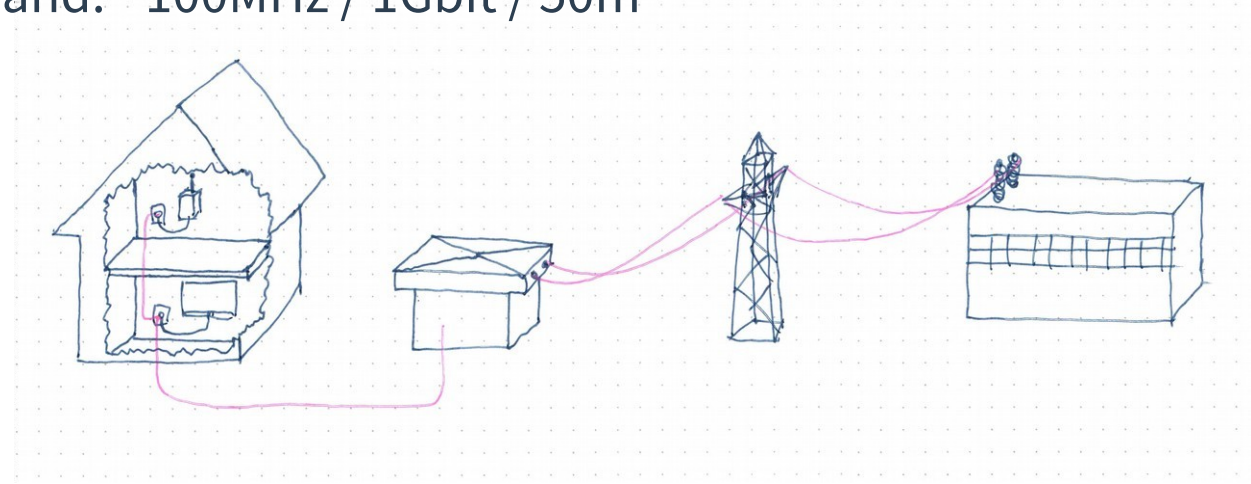
The Safety Reset

- **Triggerable over broadcast channel**
  - avoid Warntag-style issue of 1-to-1 comms service overload in case of emergency
- **Hard firmware reset through JTAG**
  - Do not trust either existing firmware or bootloader
- **Golden image: Known-good, all network comms disabled**
  - → True Fail-Safe

Communication along the Grid

Powerline Communication (PLC)

- **Transmit at higher frequencies through grid wiring**
  - Rundsteuerung: ~300Hz / 10Bd / 50km
  - Narrowband: ~100kHz / 100kBd / 1000m
  - Broadband: ~100MHz / 1Gbit / 50m



Landline IP

- DSL (Copper phone wiring)
- DOCSIS (TV cable)
- Fiber (Ethernet)

➧ All have sub-par reliability and require complex contractual architecture

Wireless IP

- **Cellular 2G/3G/4G/5G**
  - Expensive hardware
- **WiMAX etc.**
- **Satellite communication**
  - Expensive, only viable for very remote locations

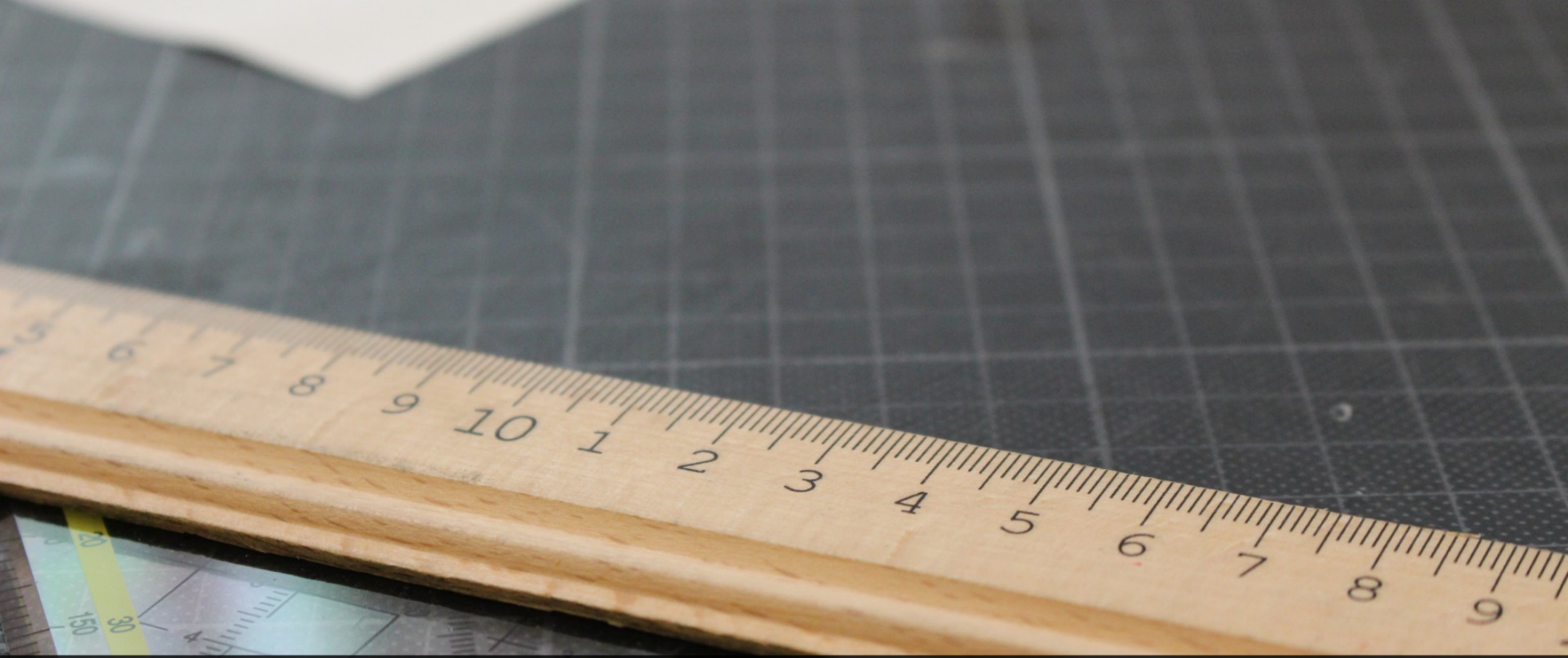
Short-range wireless

- **802.15.4 family**
  - Zigbee, Bluetooth and many proprietary protocols
- **Fully proprietary**
  - Vendor lock-in is possible
- **Frequencies: 2.4GHz, sub-1GHz**
  - Sub-1GHz preferred for range and penetration

The Hack: Grid Frequency Modulation (GFM)

➧ None of these channels work for us: They are too expensive or not reliable under attacks

- Grid frequency can be used for communication
- Grid frequency is load balance dependent
  - Generators/Transmission lines act like spring-coupled oscillators
- Apply a large load, f drops
- Modulate a large load to control Δf

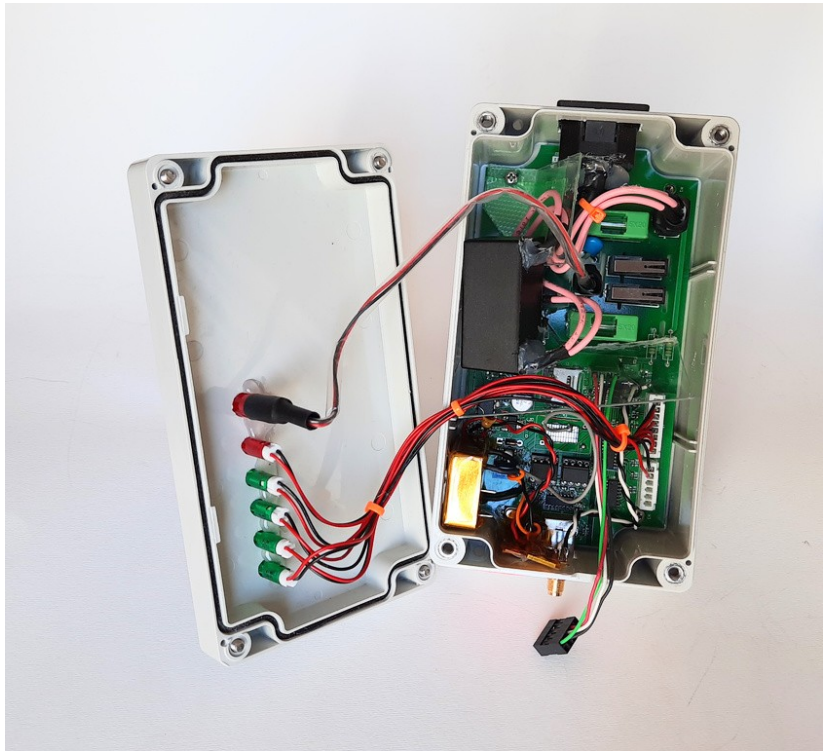


From Grid Frequency to a Reliable Channel

Channel properties

- We know grid frequency is a noisy variable
- Since f=50Hz, any modulation will be *extremely* narrowband
- Grid frequency is euqal in all parts of the grid, but has a phase delay
- Now: Characterize noise characteristics
- Later: Characterize channel transmission characteristics through experiments

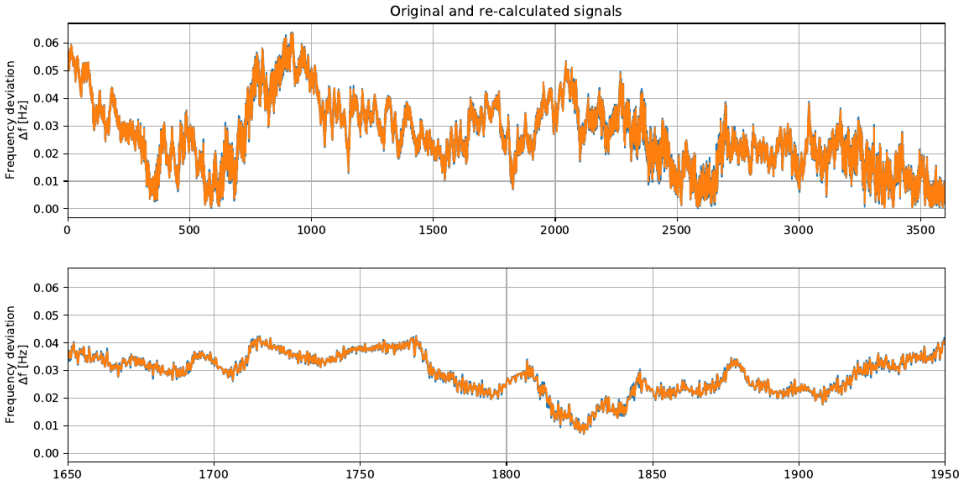
Characterizing Frequency Noise from Local Measurements

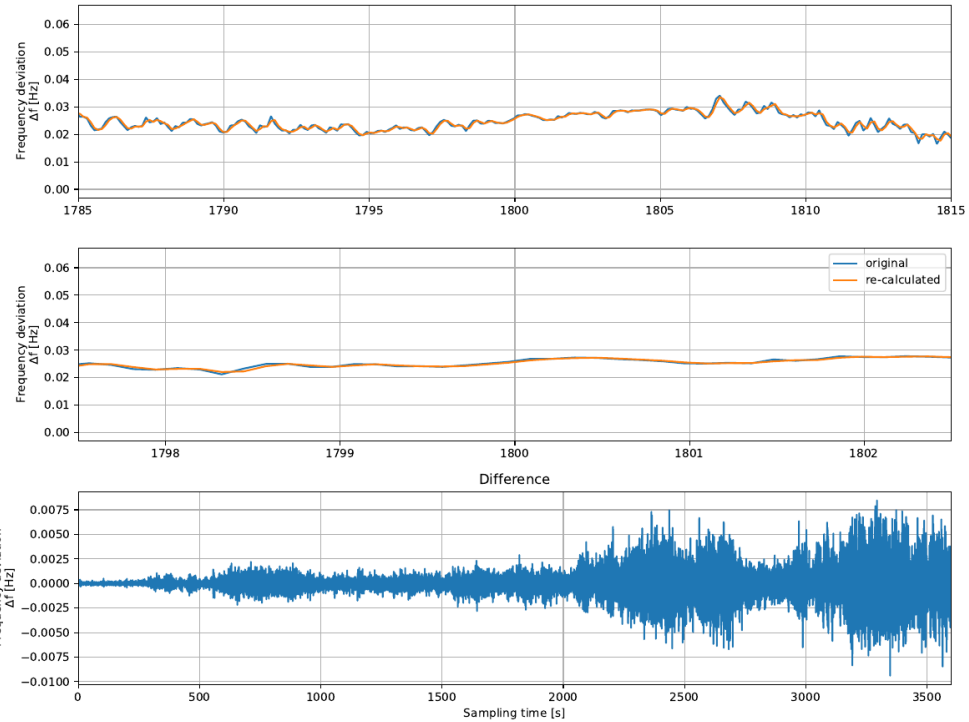


Frequency Measurement Parameters

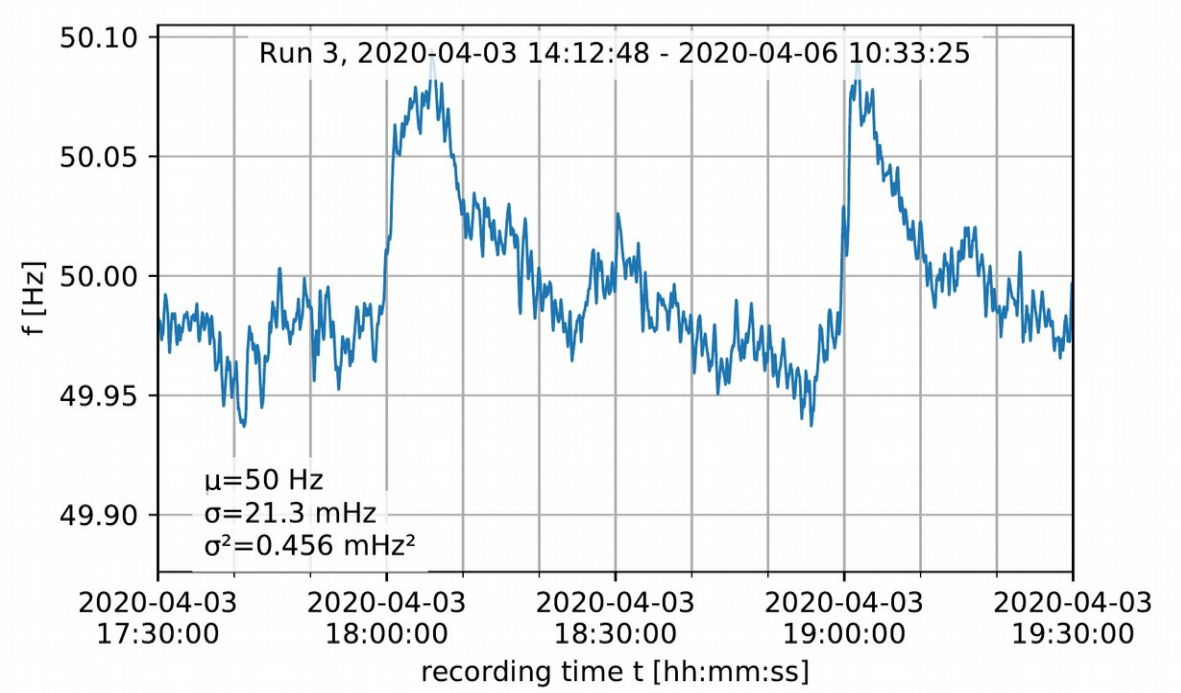
- Simple, FFT-based algorithm: Run STFT on signal, then fit gaussian to output to precisely locate peak
- Input data 1kSps @12bit
- FFT size 256 samples
- Gaussian window, sigma=16.0

Frequency Measurement Accuracy

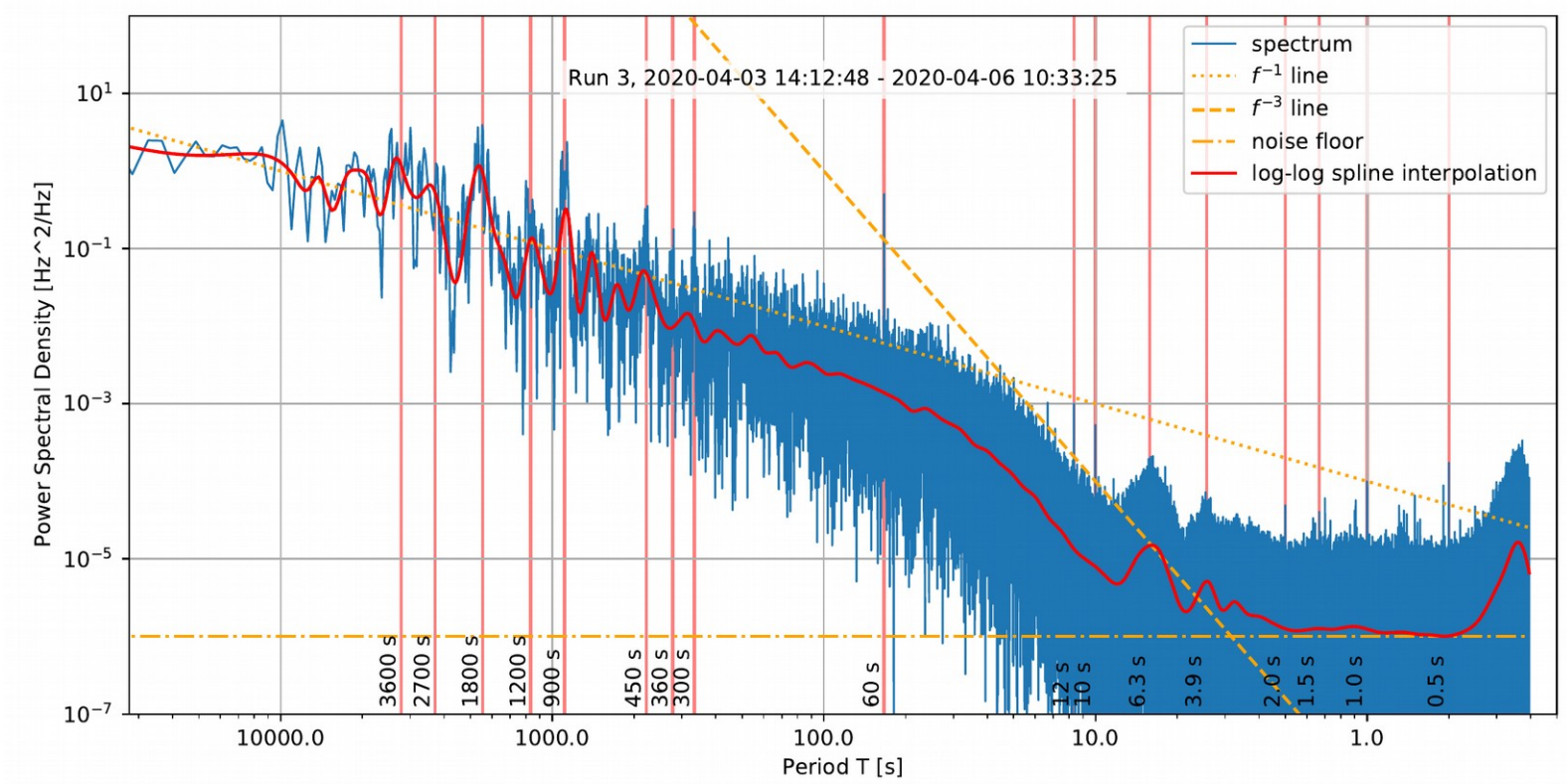




Frequency Noise Measurements



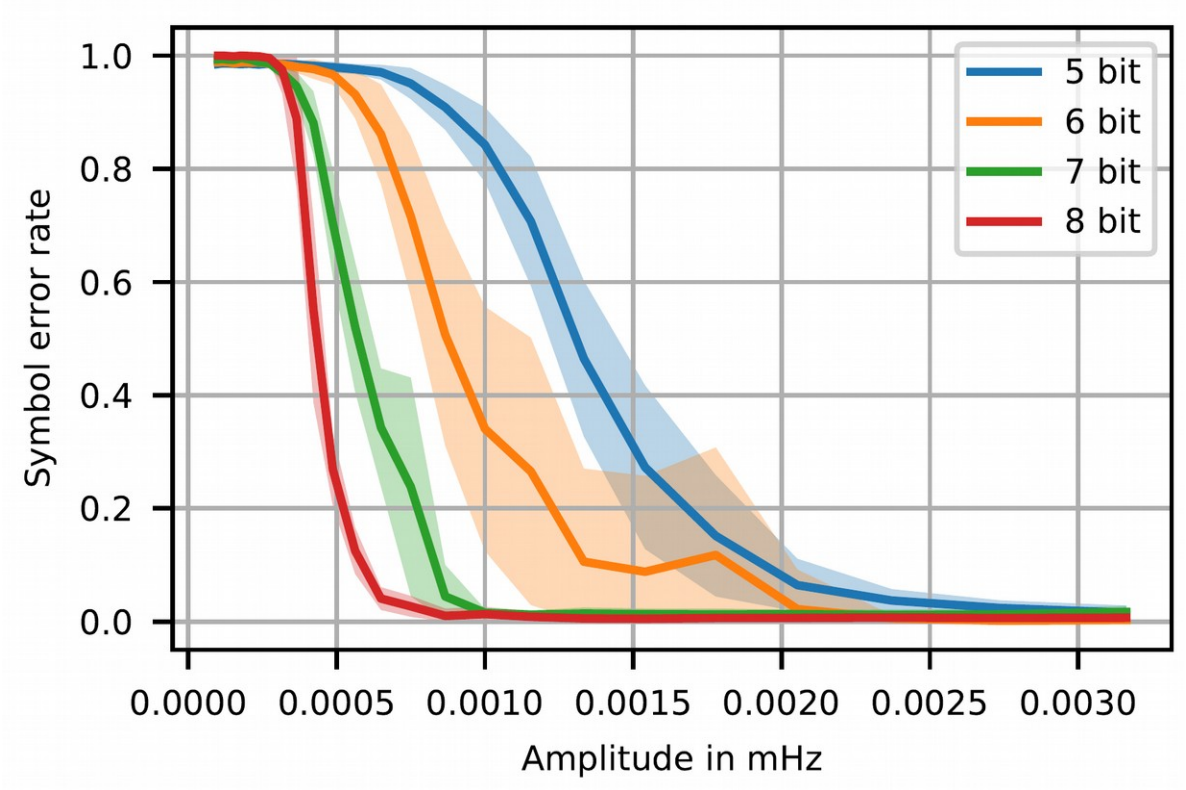
Frequency Noise Measurements



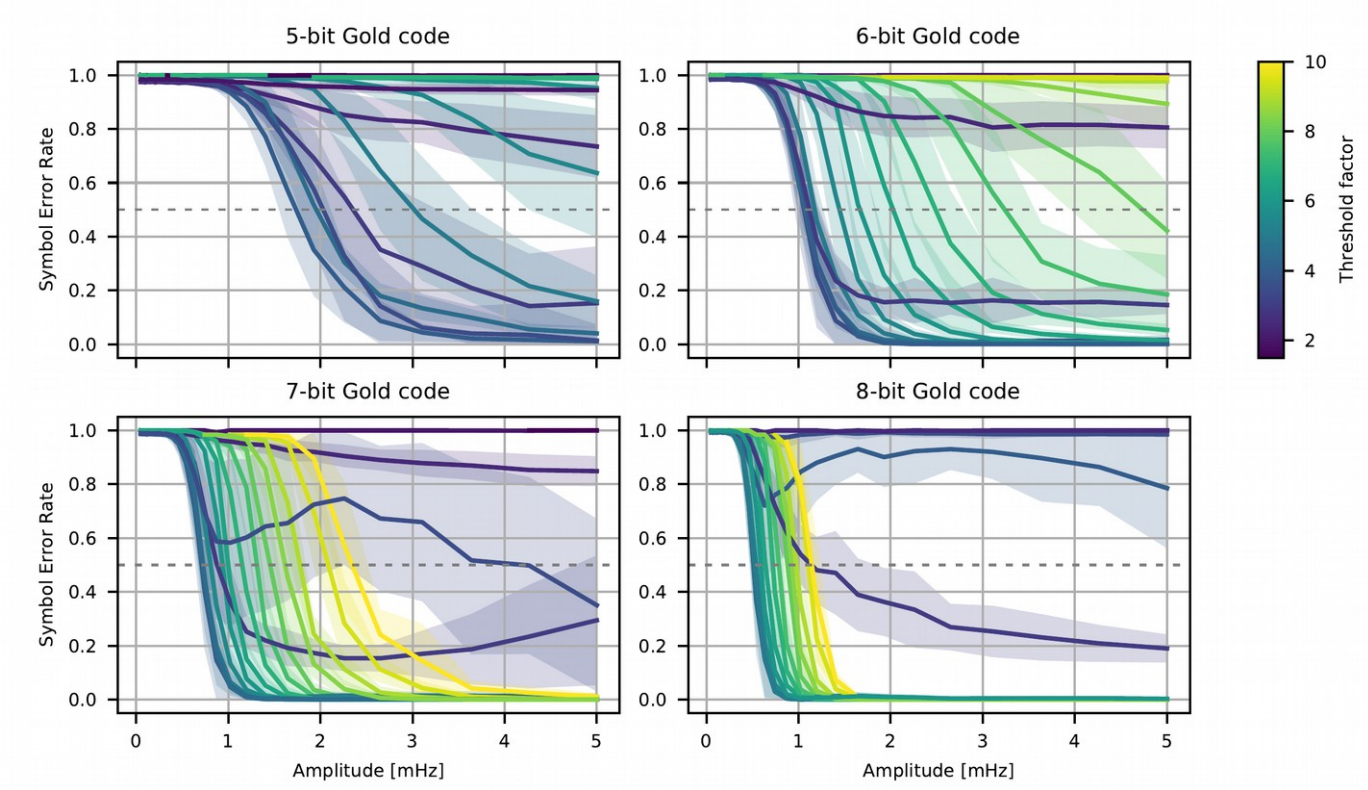
Modulation

- Poor SNR makes UWB necessary
- Limited CPU; Can’t be too complex → DSSS is a good compromise
- Long integration times (minutes) are necessary
- Accurate frequency measurement is a limiting factor

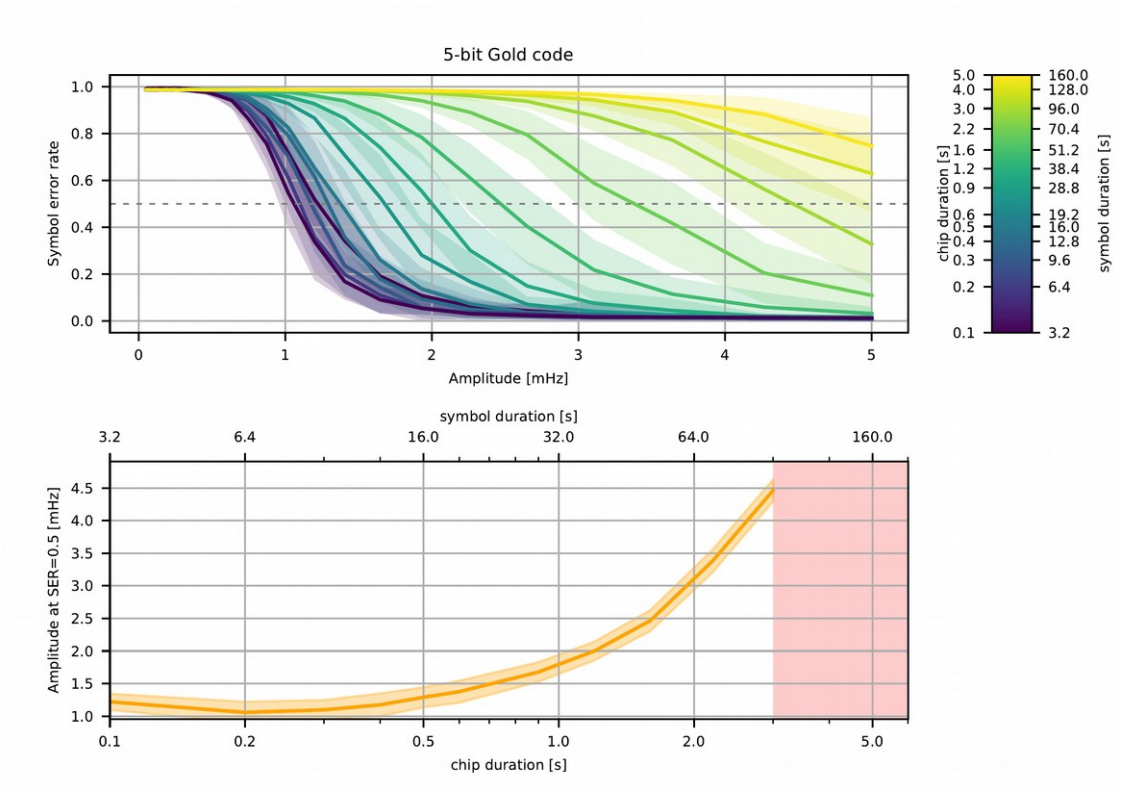
DSSS Modulation Parameters: Bit depth



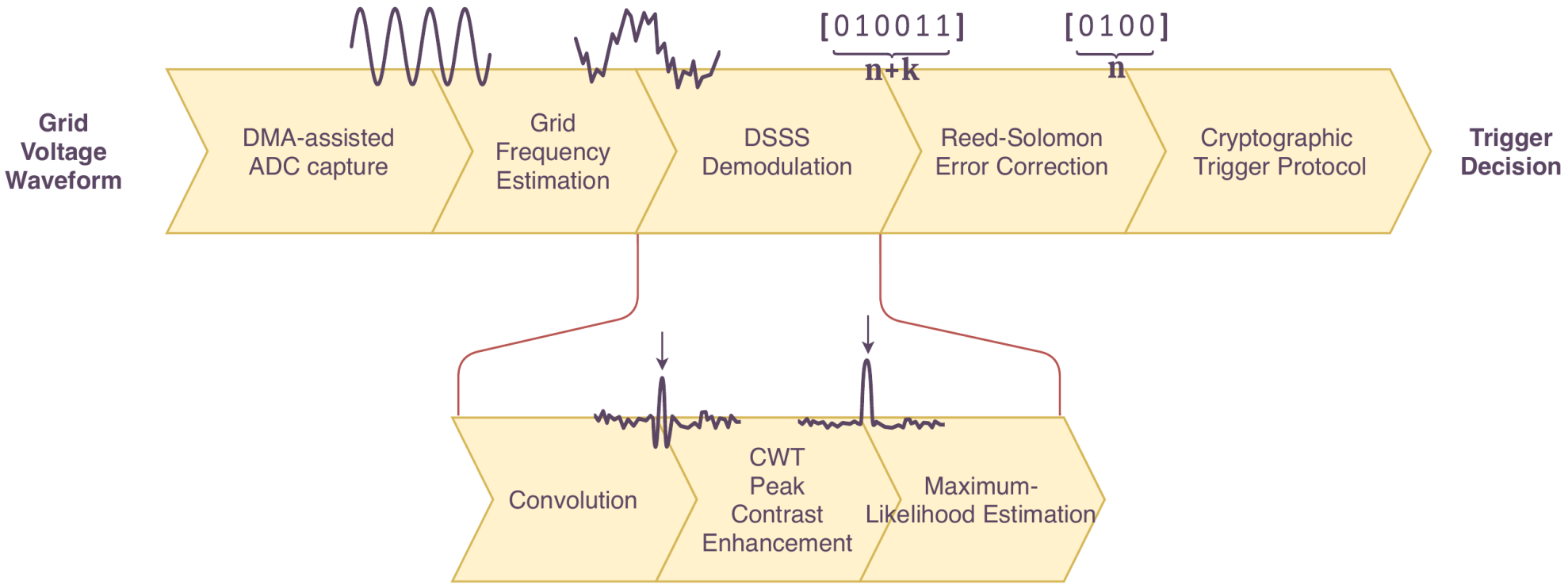
DSSS Modulation Parameters: Detection threshold



DSSS Modulation Parameters: Chip duration



Signal Processing Chain



Chosen Modulation Parameters

- **5 bit** Gold Code
- **1s chip** duration
  → 31s symbol duration
- Threshold factor:
  5.0× background noise level

Error Correction

- Slow transmission requires small block size
- There is a code size limitation
- It can be simple: Efficiency is good, but not critical

Cryptography

- **Non-standard threat model**
- **Simple setup**
- **A trivial custom solution is justifiable to save transmission bandwidth**
- **Simply use pre-computed hash chain**
  - Reset controller knows last hash
  - Reset authority knows first hash
  - RA reveals one previous hash to trigger reset

    → Small transmission size, trivial

Testing & Validation

Extensive simulations in Jupyter

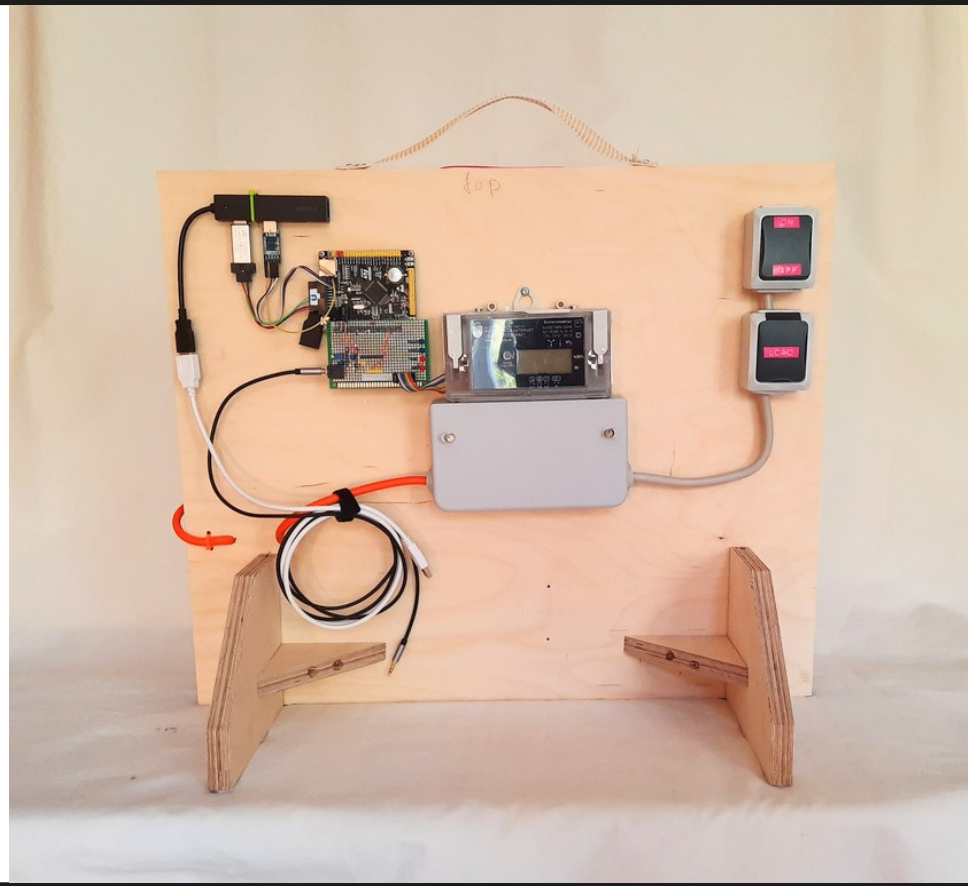
- Jupyter allows real-time tinkering with high-quality, interactive graphical plots
- Python code can easily be extracted for running on remote machinees
- Plots can easily be exported to publication-quality graphics

Host testing of instrumented firmware

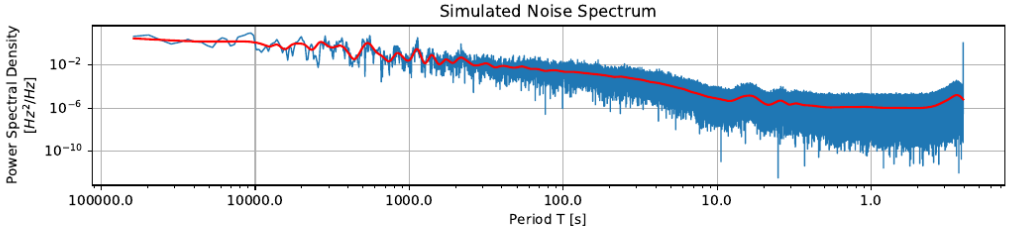
- Firmware implementation of algorithms compiled for host, run from python test fixtures
- Allows for validation of fixed-precision device code against double-precision host prototype

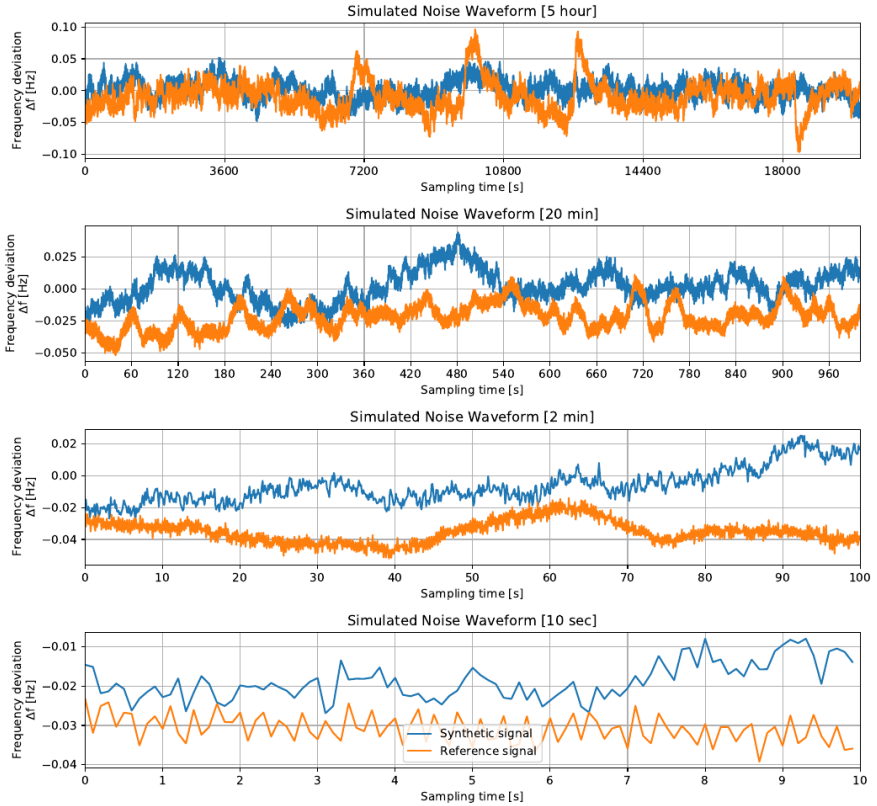
Demonstrator experiments

- Goal: Experimentally verify final optimized set of parameters against synthetic grid voltage trace
- Result: It works :)



Synthetic Signal Quality

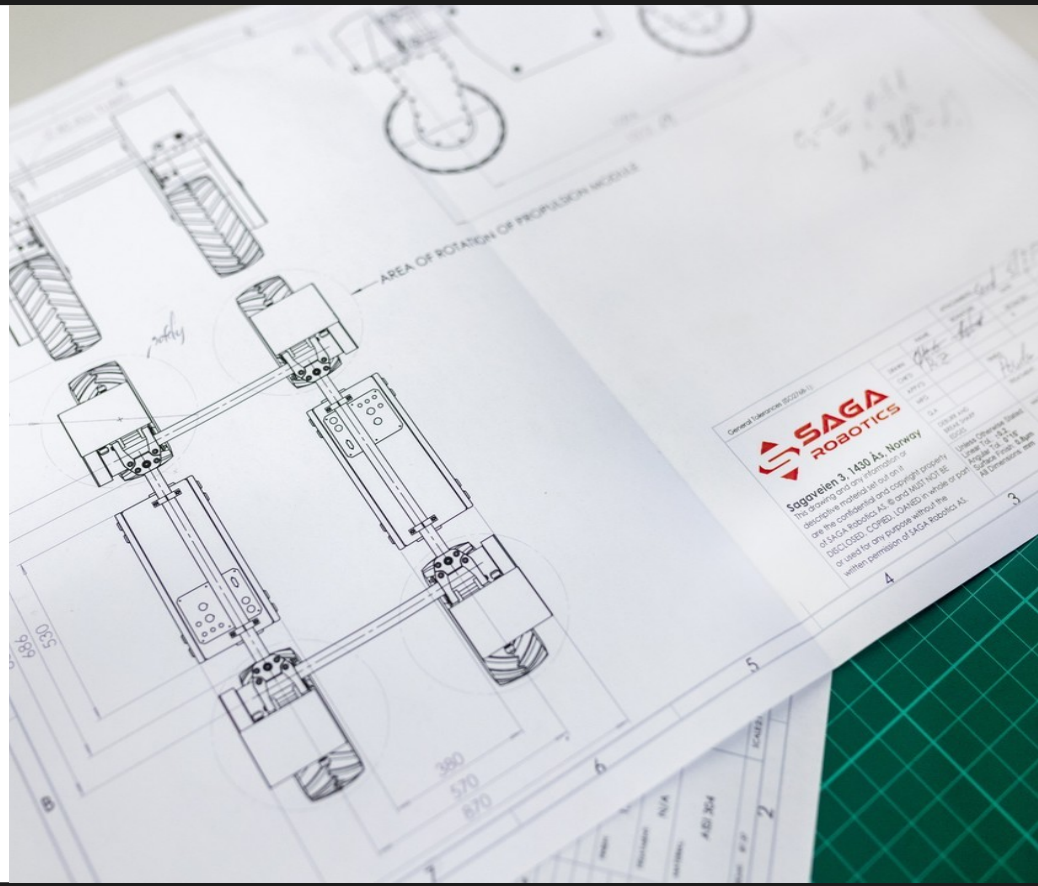




Conclusion

Theoretical analysis results

- FW security is a serious issue
- The attack potential is there
- Safety reset is a viable option
- GFM is viable even during an attack



Experimental results

- **Computer simulations using recorded data**
  - Positive result
- **Practical experiments using emulated data**
  - Positive result
- **Conclusion: 20s/bit after ECC is practical**
  - ~15min for complete trigger

Tangible products [1m]

- The grid frequency sensor
- The demonstrator
- Extensive simulation notebooks
- Prototype firmware
- Firmware code size analysis tool

➧ All Open Source!!


Q&A