

MARC NEWLIN // MATT KNIGHT // BASTILLE NETWORKS

---

# SO YOU WANT TO HACK RADIOS

A PRIMER ON WIRELESS REVERSE ENGINEERING

# WHO ARE THESE GUYS

- ▶ Marc “mou\$e whisperer” Newlin
  - ▶ Security Researcher @ **Bastille**
  - ▶ Discovered **Mousejack** vulnerability in 2016
  - ▶ Finished 2nd in DARPA Spectrum Challenge in 2013
  - ▶ Finished 3nd in DARPA Shredder Challenge in 2011
  
- ▶ Matt Knight
  - ▶ Software Engineer and Security Researcher @ **Bastille**
  - ▶ Reverse engineered the **LoRa** wireless protocol in 2016
  - ▶ BE & BA from Dartmouth

marc@**Bastille**.net  
@marcnewlin

matt@**Bastille**.net  
@embeddedsec

**WHO IS THIS FOR?**

**WHY SHOULD YOU  
CARE?**

WIRELESS SYSTEMS  
ARE EVERYWHERE

# WIRELESS PROLIFERATION

- Cisco IBSG: 50 billion devices by 2020

# WIRELESS PROLIFERATION

- Cisco IBSG: 50 billion devices by 2020

# MOBILE

# WIRELESS PROLIFERATION

- Cisco IBSG: 50 billion devices by 2020

IOT

# WIRELESS PROLIFERATION

- ▶ Cisco IBSG: 50 billion devices by 2020
- ▶ Fewer wires every year

# ABOUT THE INTERNET OF THINGS...

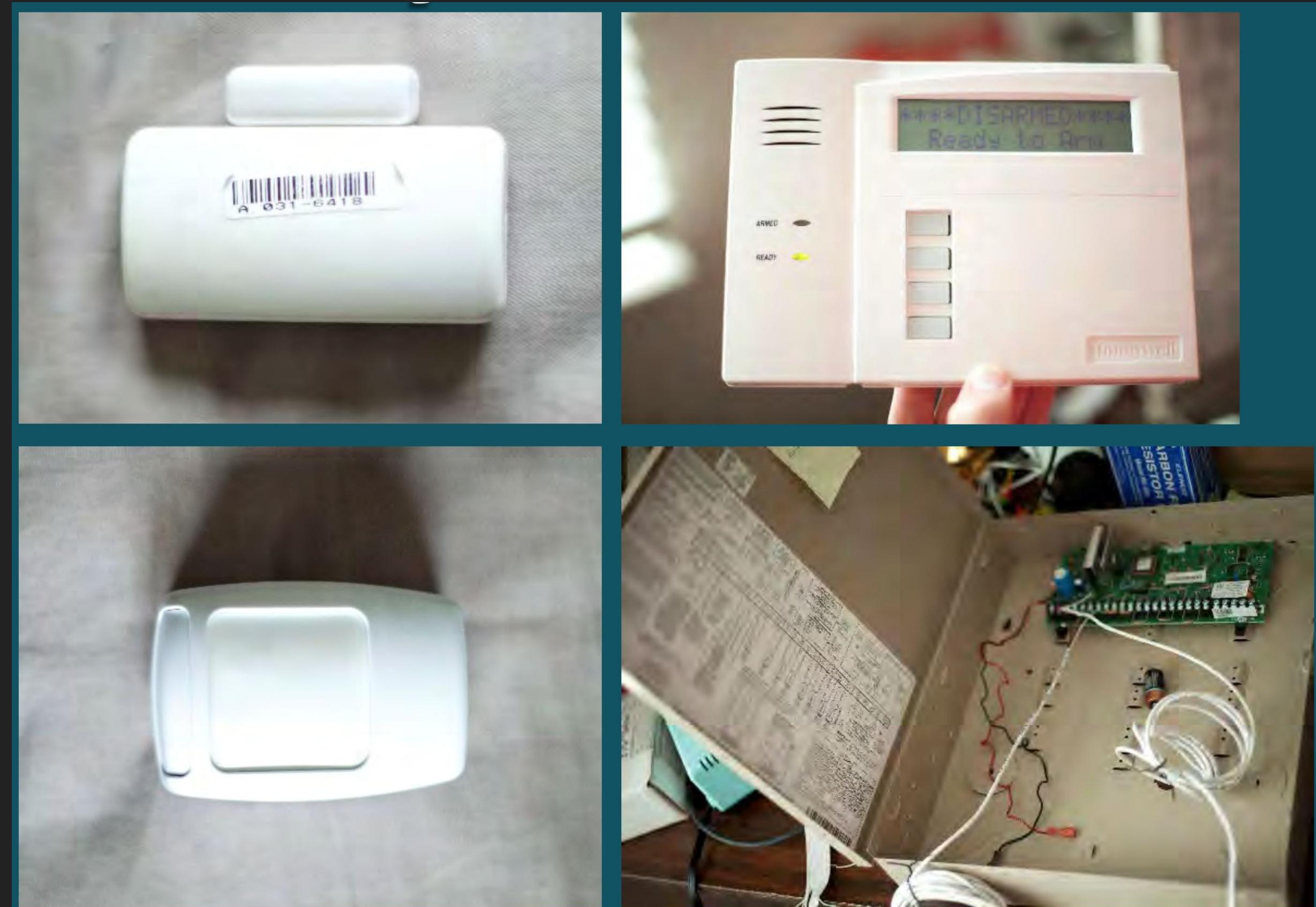
- ▶ America's Favorite Buzzword™
- ▶ What is it, actually?
- ▶ Sales and marketing speak for “connected embedded devices”
- ▶ “Smart” devices are usually pretty stupid

# EMBEDDED REALITIES

- ▶ Embedded systems are built on **compromise**
- ▶ Hardware: Small, inexpensive: Limits connectivity and encryption capabilities
- ▶ Power: Battery powered: Not promiscuous, intensive duty/sleep cycling
- ▶ Deployment: Hard to reach locations: Wireless, easily configurable, legacy compatible
- ▶ Updates: Difficult to update: OTP memory, network limitations, OEM/vendor supply chain

**Vulnerable by Virtue of Being Constrained**

# ALARM SYSTEM VULNERABILITIES

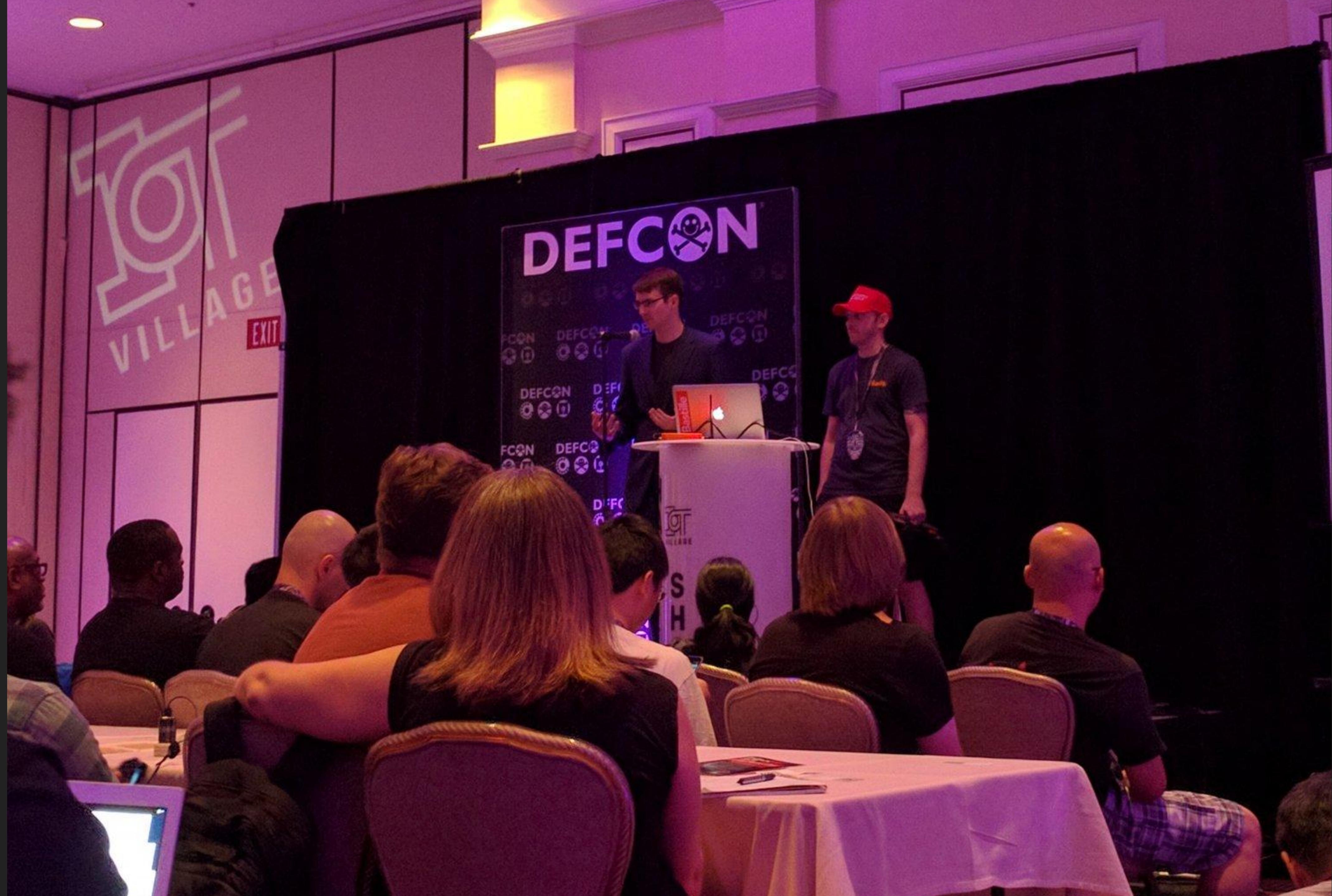


- ▶ Discovered by **Bastille**'s Logan Lamb in 2014
- ▶ Legacy RF link between home alarm system sensors and control panel is vulnerable to:
  - ▶ Jamming (denying alarm reporting)
  - ▶ Command injection (trigger false alarms)
  - ▶ Eavesdropping (detect occupancy, monitor movement)

# MOUSEJACK

- ▶ Discovered by **Bastille**'s Marc Newlin in 2015
- ▶ RF link between non-Bluetooth **wireless keyboards and mice** (100MMs of devices) vulnerable to:
  - ▶ Command injection (running arbitrary commands at current permissions level)
  - ▶ Eavesdropping (sniffing passwords, credit card #s, etc.)





## IOT VILLAGE FEEDBACK

- ▶ Interest in Software Defined Radio and RF systems is high
- ▶ RF is intimidating!
- ▶ Too much EE for software people
- ▶ Too academic!

**NO PHD?**

**NO PROBLEM!**

# AGENDA

1. So you want to hack RF...
2. Introduce **essential** RF concepts
3. Introduce RF reverse engineering **workflow** that applies to **all** systems
4. Do it **live!**
  1. Z-Wave home automation protocol
  2. Wireless doorbell
  3. HP wireless keyboard



This is what it's all about

## WHAT WE WON'T COVER

Digital Signal Processing

SO YOU WANT TO

---

HACK WIRELESS

# BARRIERS TO ENTRY

- ▶ Lower than ever before
- ▶ Commodity hardware is:
  - ▶ Really powerful
  - ▶ Increasingly cheap
- ▶ Free (beer && liberty) software is abundant!

# HARDWARE TOOLS

- ▶ Dedicated Radio Chipset (Hardware Defined Radio)
  - ▶ Does 1 protocol really well
  - ▶ Pros: single-protocol performance, cost, simplicity, low power
  - ▶ Cons: lack of flexibility
- ▶ Examples:
  - ▶ Ubertooth (\$200)
  - ▶ RFCat / Yardstick One (\$100)
  - ▶ nRF24 dongles (\$35)
  - ▶ ApiMote (\$90)

# HARDWARE TOOLS

- ▶ Software Defined Radio (SDR)
  - ▶ Swiss army knife for most-things RF
  - ▶ Pros: flexibility (can implement **any** protocol)
  - ▶ Cons: cost, complexity, power, performance (software and RF)
- ▶ Examples:
  - ▶ Ettus USRP (\$686—>\$\$\$\$\$)
  - ▶ HackRF (\$300)
  - ▶ BladeRF (\$420-\$650)

# FREE SOFTWARE

- ▶ SDR:
  - ▶ GNU Radio: open source digital signal processing suite
  - ▶ GNU Radio OOT Modules: third party plugins
    - ▶ gr-lora, gr-nordic
  - ▶ Baudline, Inspectrum, Fosphor: powerful analysis tools
- ▶ HDR:
  - ▶ Bluez, libubertooth, Killerbee
  - ▶ Marc's nRF24 library

TOOLS ARE  
RIDICULOUS

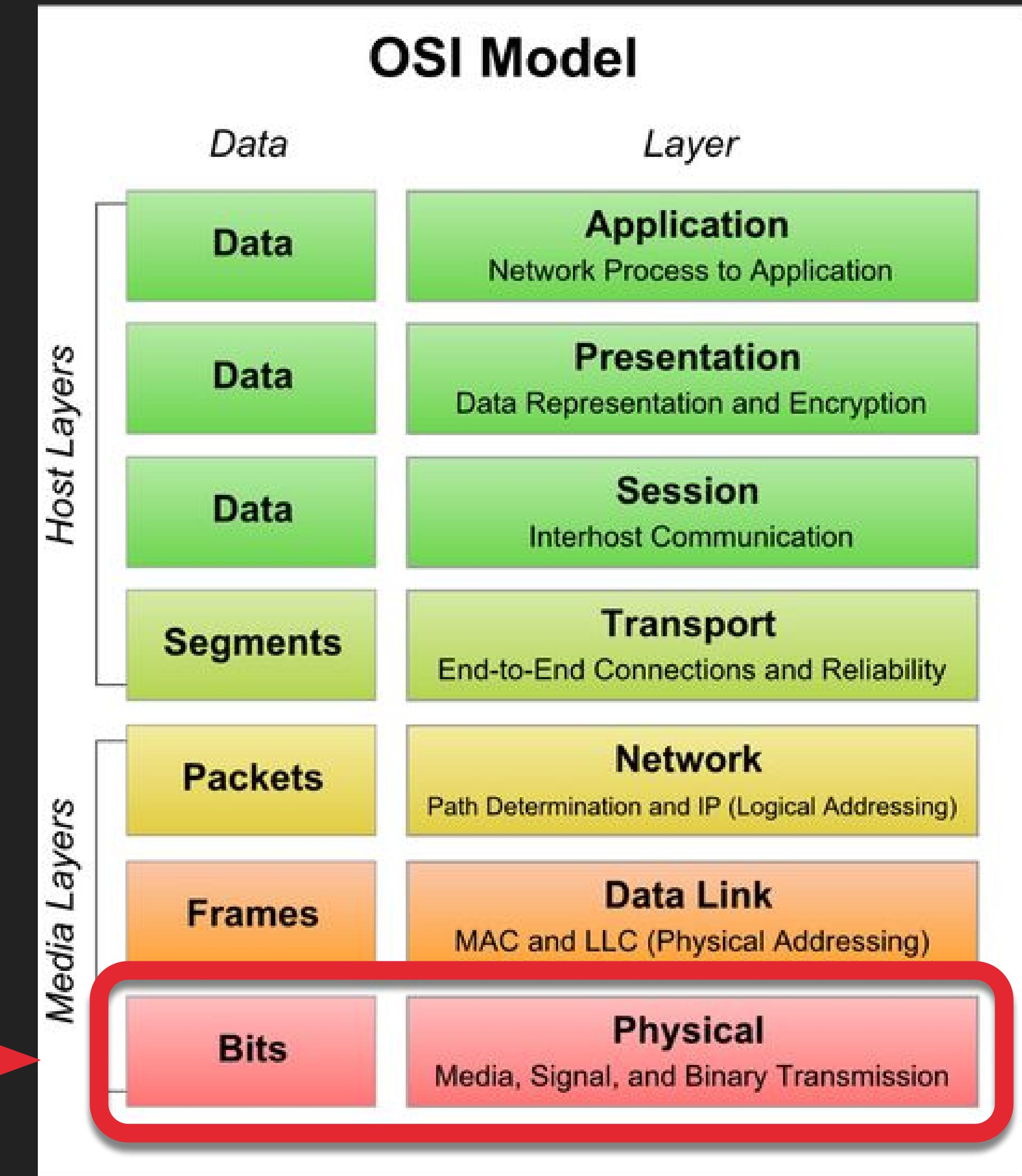
OFFENSIVELY  
~~OBScenely~~ SHORT

---

RADIO CRASH COURSE

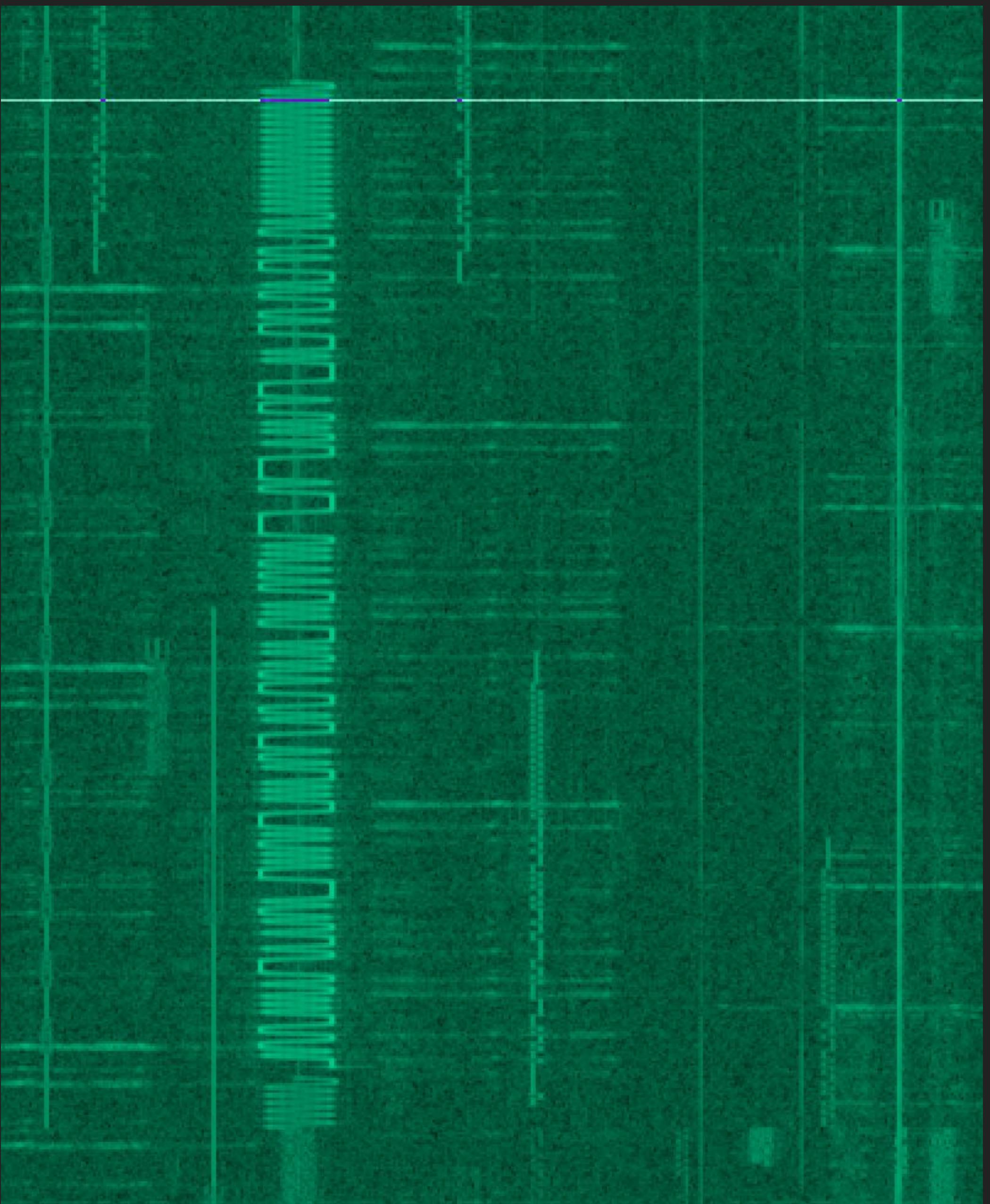
# PHY LAYER

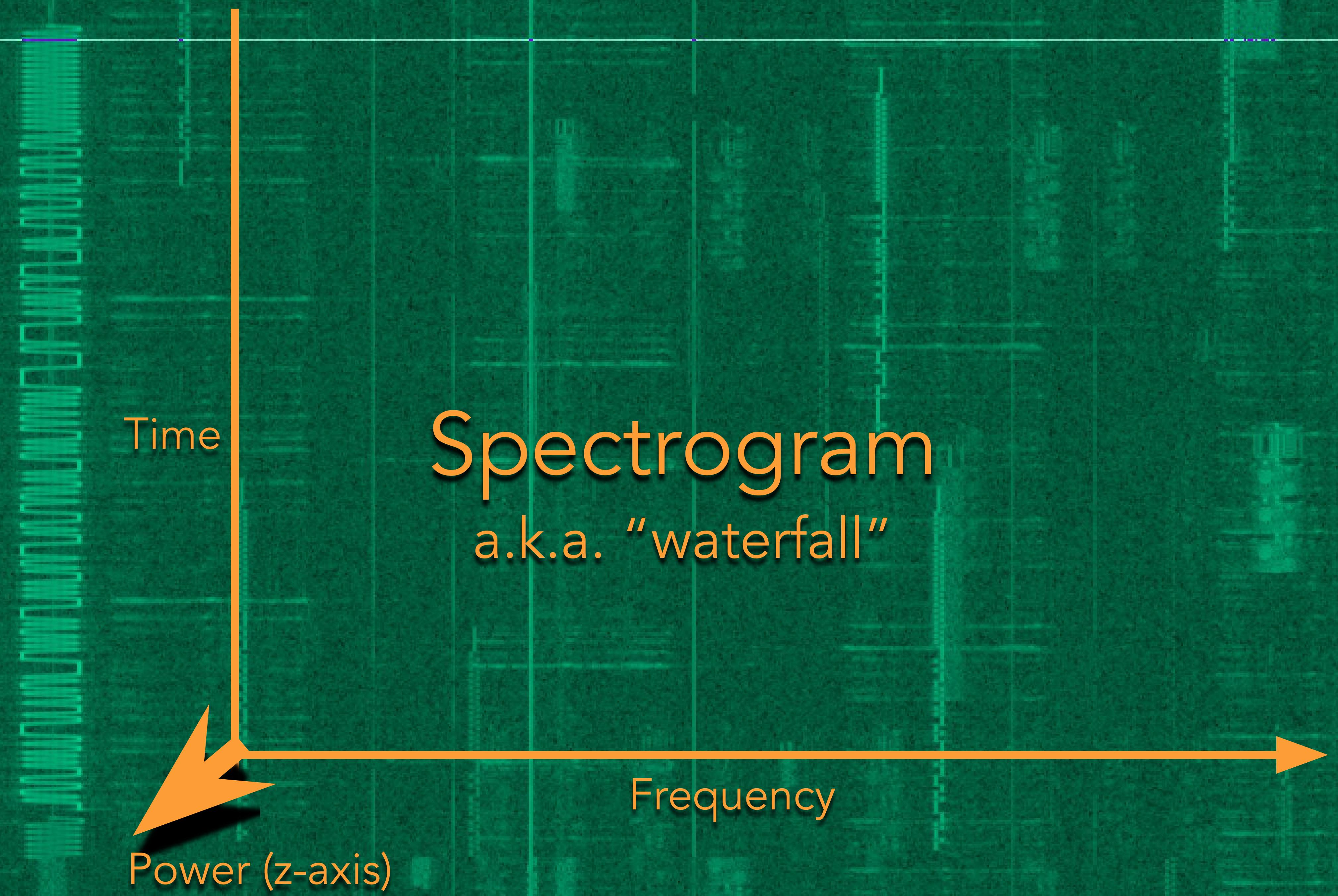
- ▶ Lowest layer in communication stack
- ▶ In wired protocols: voltage, timing, and wiring defining 1s and 0s
- ▶ In wireless: patterns of energy being sent over RF medium



# WHAT IS RF?

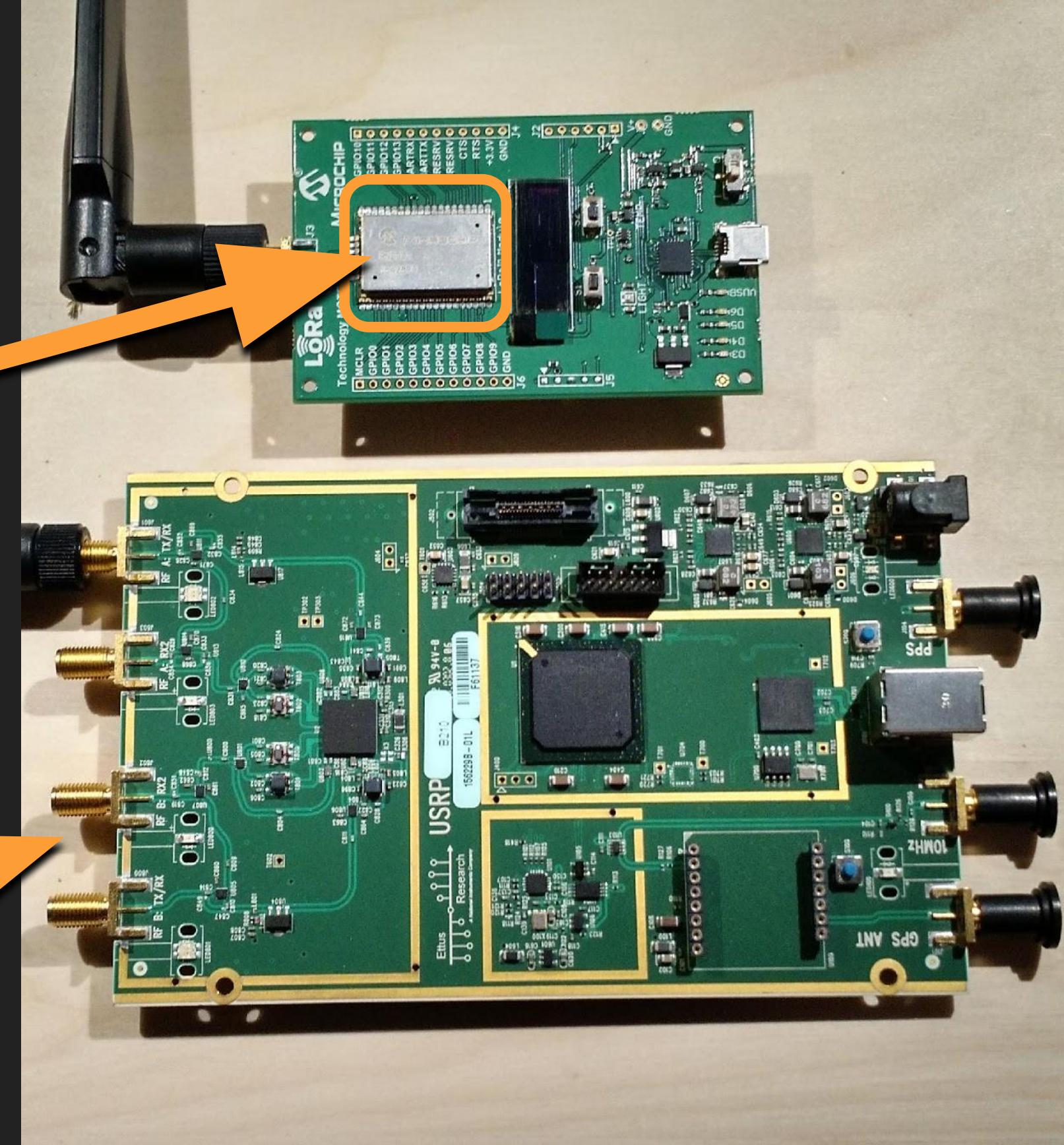
- ▶ “One of the four fundamental forces of the universe” — Tom Rondeau, DARPA Program Manager, former GNU Radio lead
- ▶ “Radio Frequency”
- ▶ Electromagnetic waves
- ▶ Energy





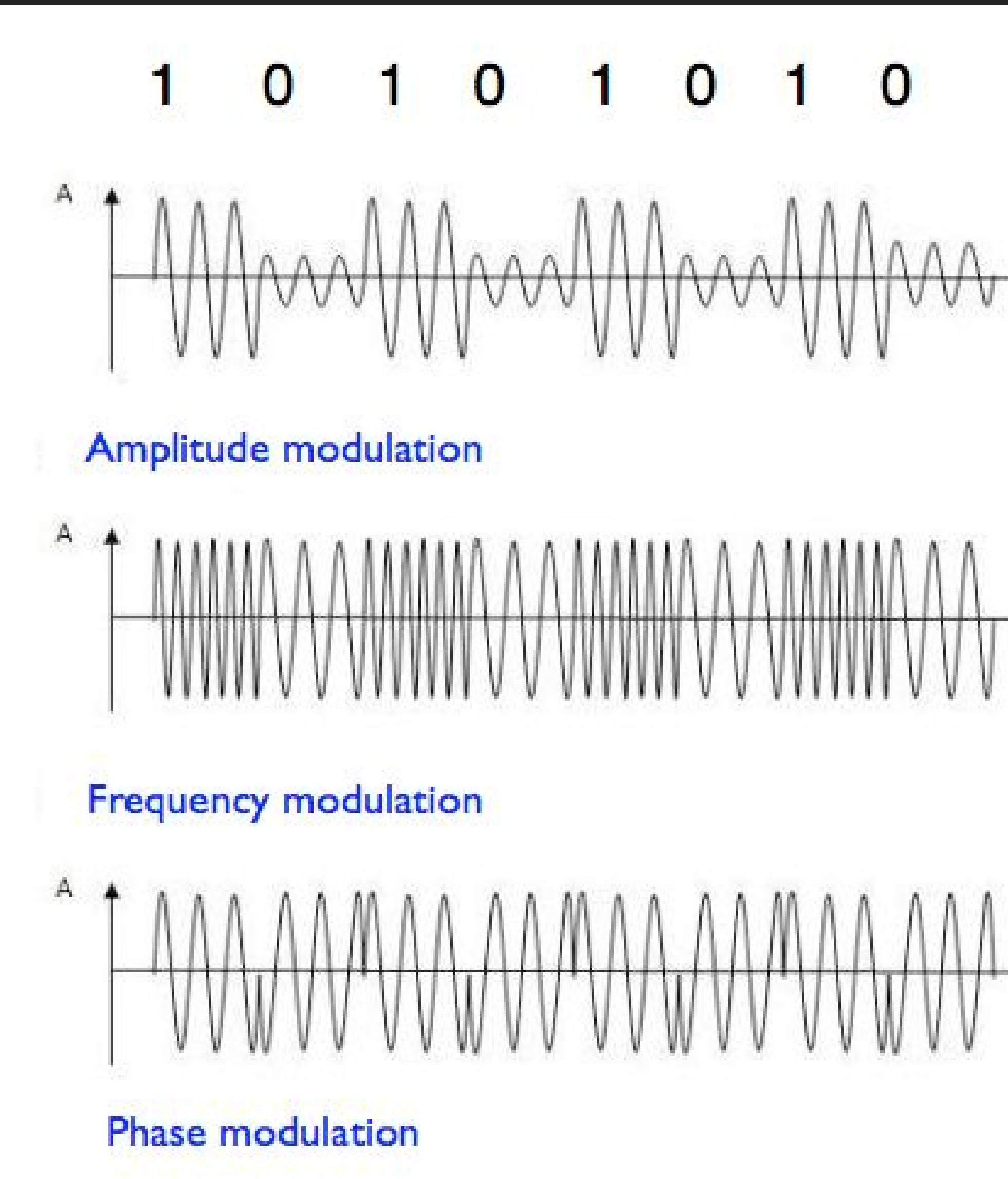
# MANIPULATING RF

- ▶ Done with a radio
- ▶ Hardware defined
- ▶ RF and protocol in silicon
- ▶ Software defined radio (SDR)
- ▶ Flexible silicon handles RF
- ▶ Protocol-specific components implemented in software (CPU or FPGA)



# PHY COMPONENTS

- ▶ Modulation
- ▶ How digital values are mapped to RF energy
- ▶ RF parameters that can be modulated:
  - ▶ Amplitude
  - ▶ Frequency
  - ▶ Phase
  - ▶ some combination of the above

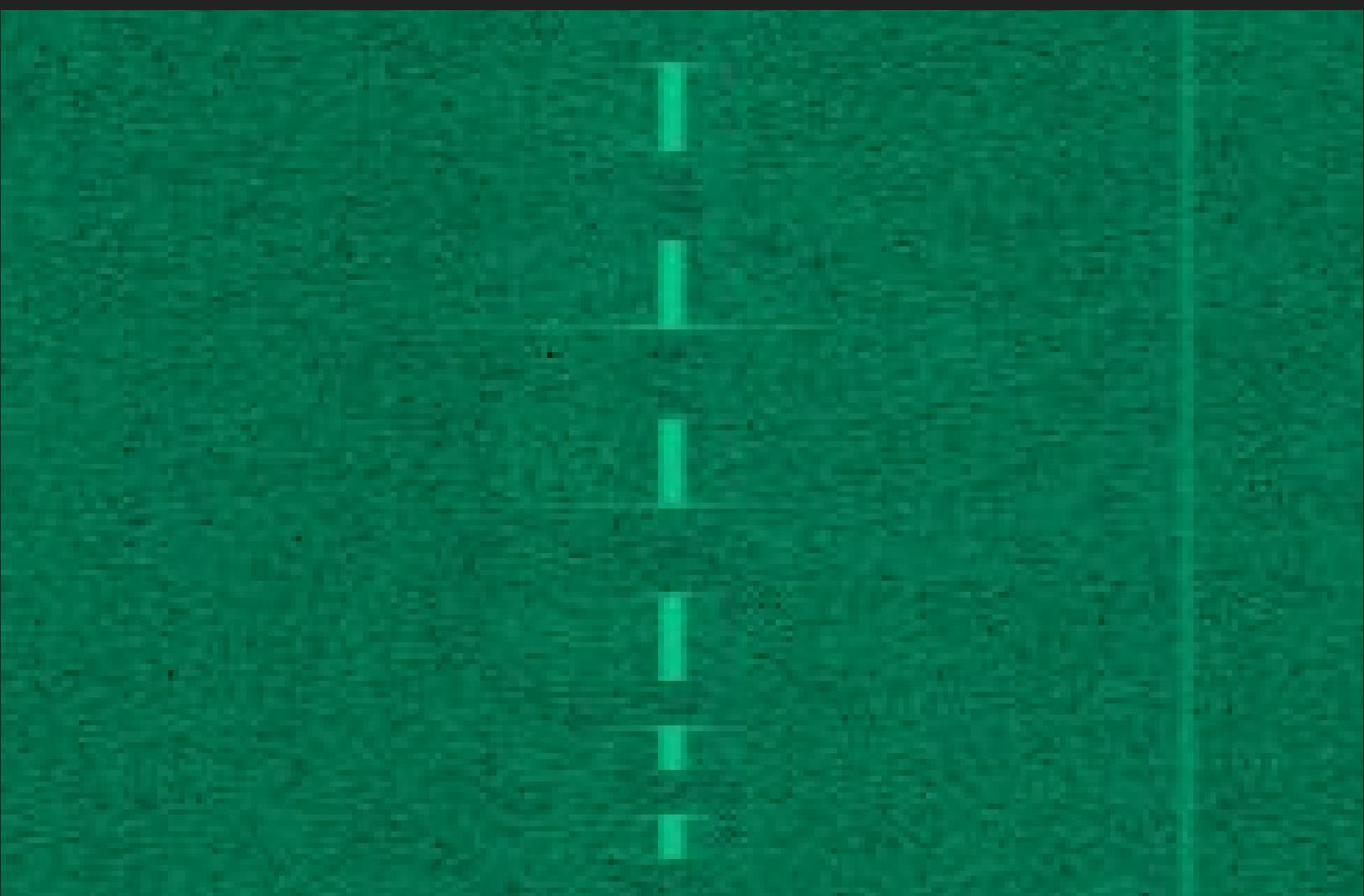
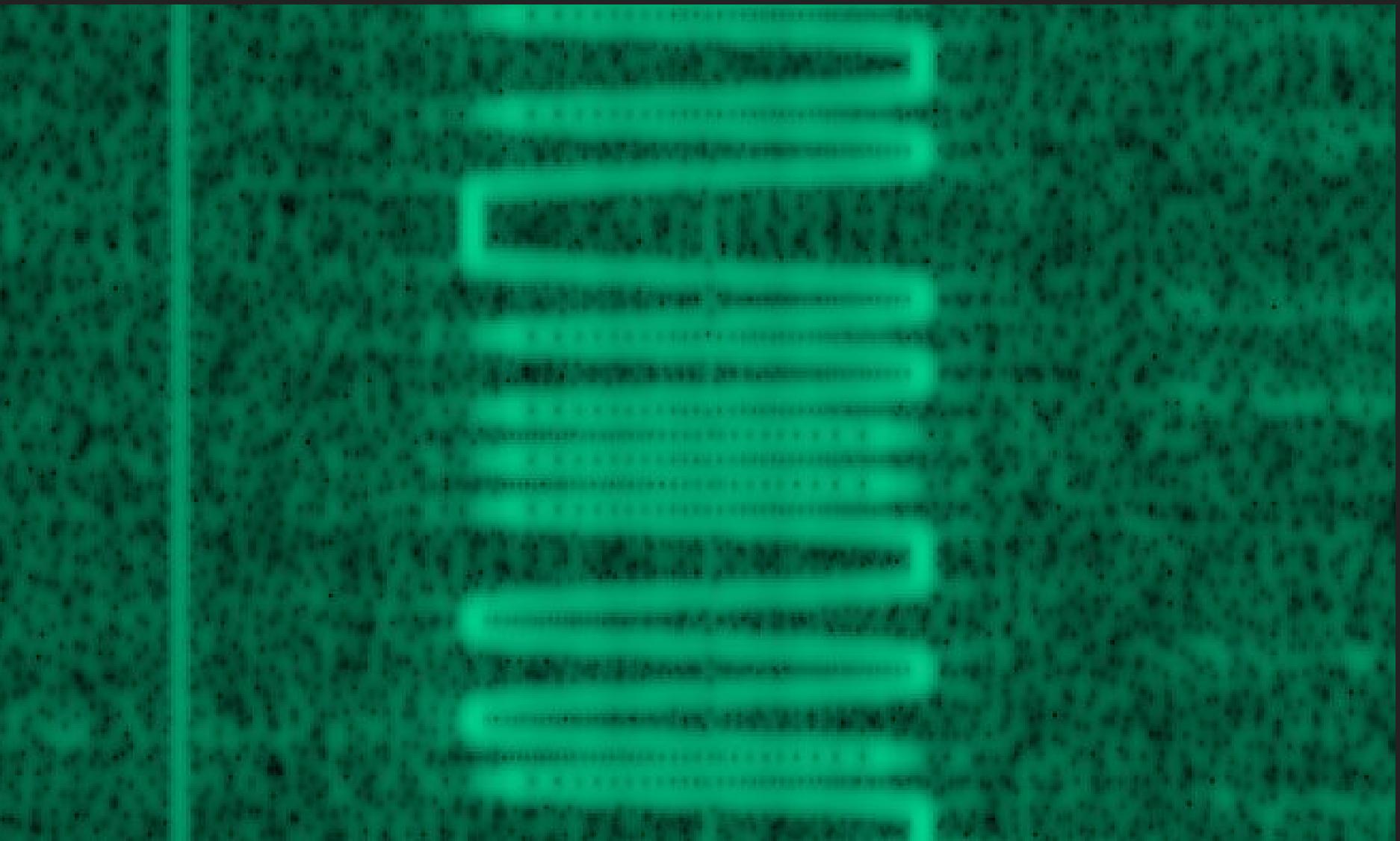


# MODULATION

- ▶ Modulators can modulate **analog or digital** information
- ▶ Digital modulation
- ▶ **Symbols:** discrete RF energy **state** representing some quantity of information

# COMMON IOT PHYS

- ▶ Frequency Shift Keying: FSK, GFSK
  - ▶ RF energy **alternates** between two frequencies to signify digital values
  
- ▶ Amplitude Shift Keying: ASK, OOK
  - ▶ Changes in RF **power** on a certain frequency signify digital values





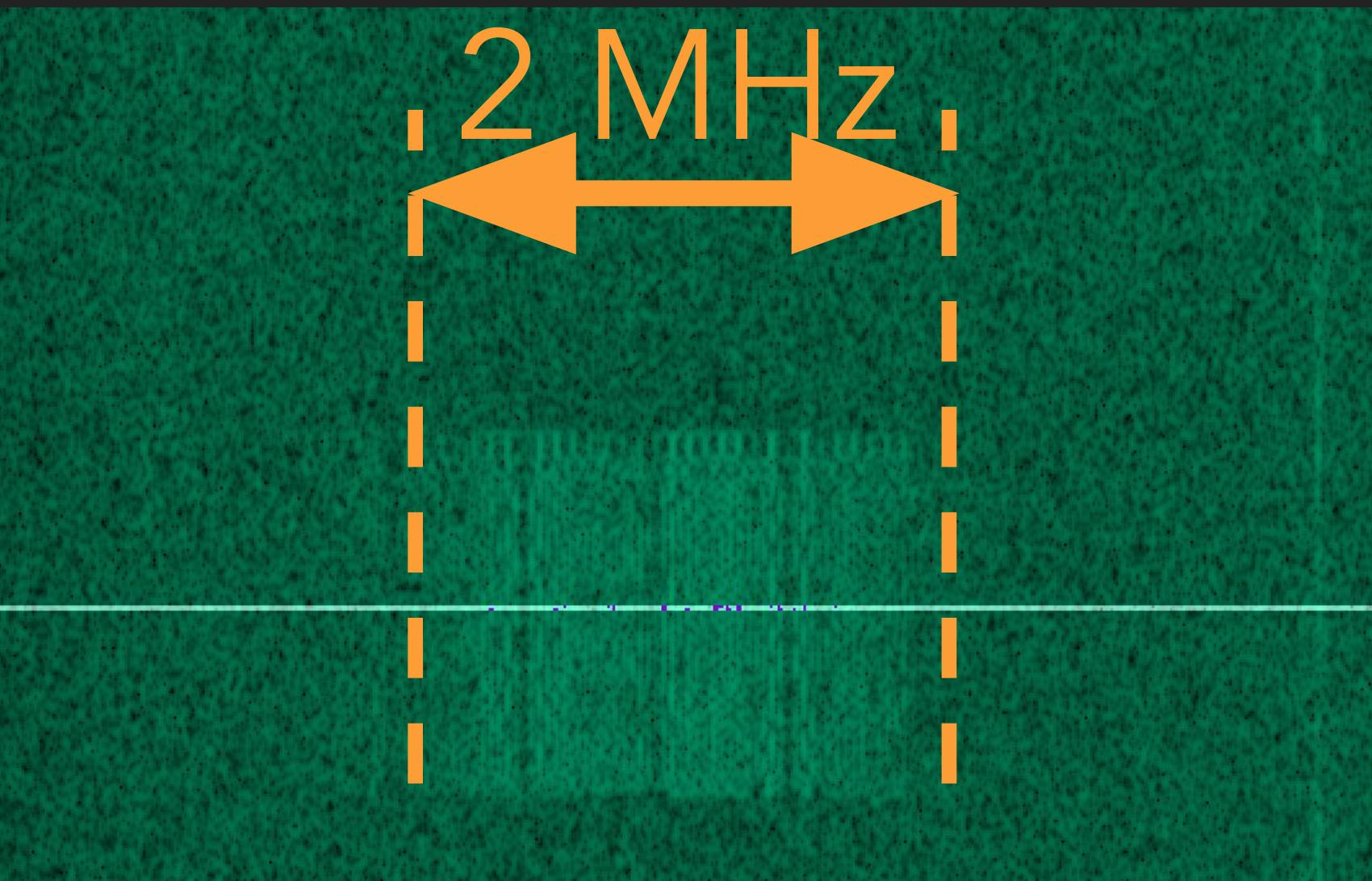
## SYMBOLS ILLUSTRATED

- ▶ Top: FSK
- ▶ Bottom: OOK/ASK
- ▶ Compare with analog modulation
- ▶ Analog = infinite possible symbols
- ▶ Digital = finite number of possible symbols, defined by modulation

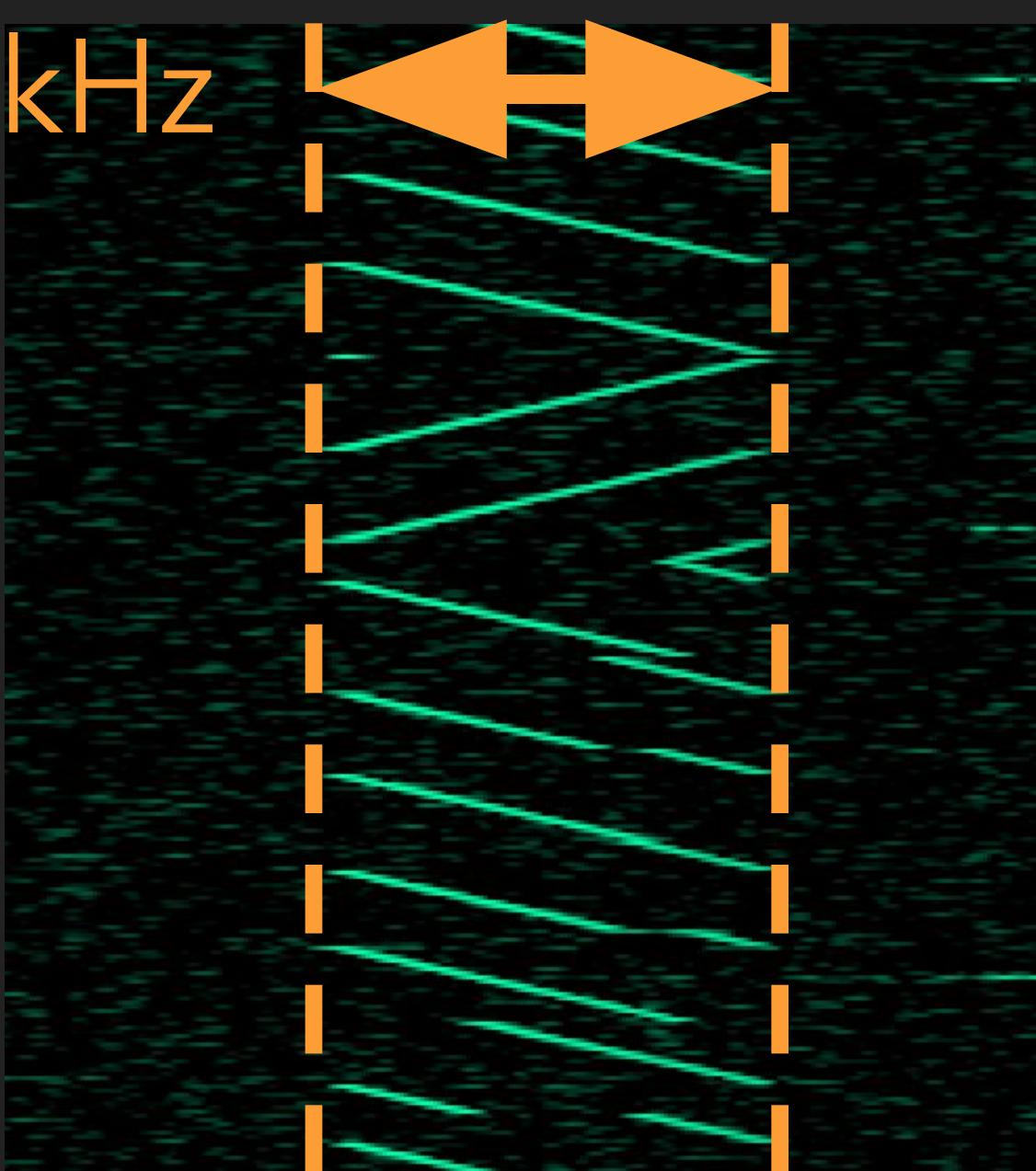


## MORE COMPLICATED IOT PHYS

- ▶ Spread spectrum
- ▶ Data bits are encoded at a higher rate and occupy more spectrum
- ▶ Resilient to RF noise
- ▶ Examples:
  - ▶ 802.15.4 (top)
  - ▶ LoRa (bottom)



125, 250, or 500 kHz



# RADIOS CONTINUED

- ▶ Radios can have two functions:
  - ▶ Transmitting
  - ▶ Receiving
- ▶ If a radio can do both it is dubbed a **transceiver**

# ON REVERSE ENGINEERING

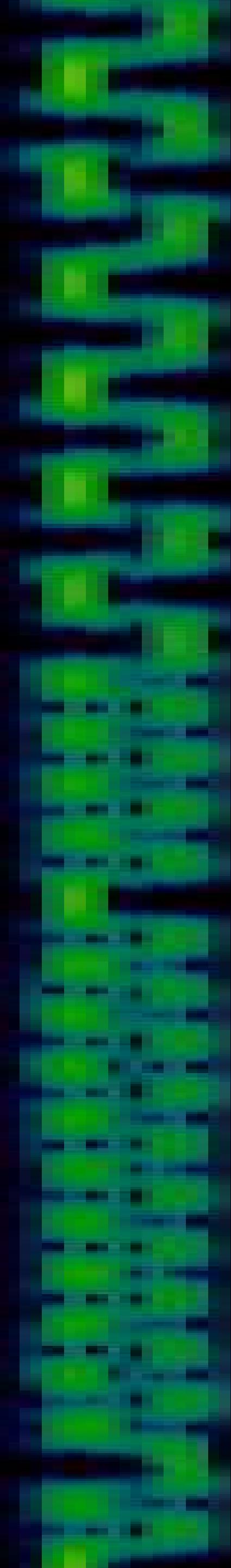
- ▶ How does one reverse engineer an arbitrary wireless system?
- ▶ Main objective: figure out how **data** is mapped to **symbols**
- ▶ Reverse engineering boils down to building **receivers**

WIRELESS REVERSE ENGINEERING

---

METHODOLOGY

[INTERACTIVE]



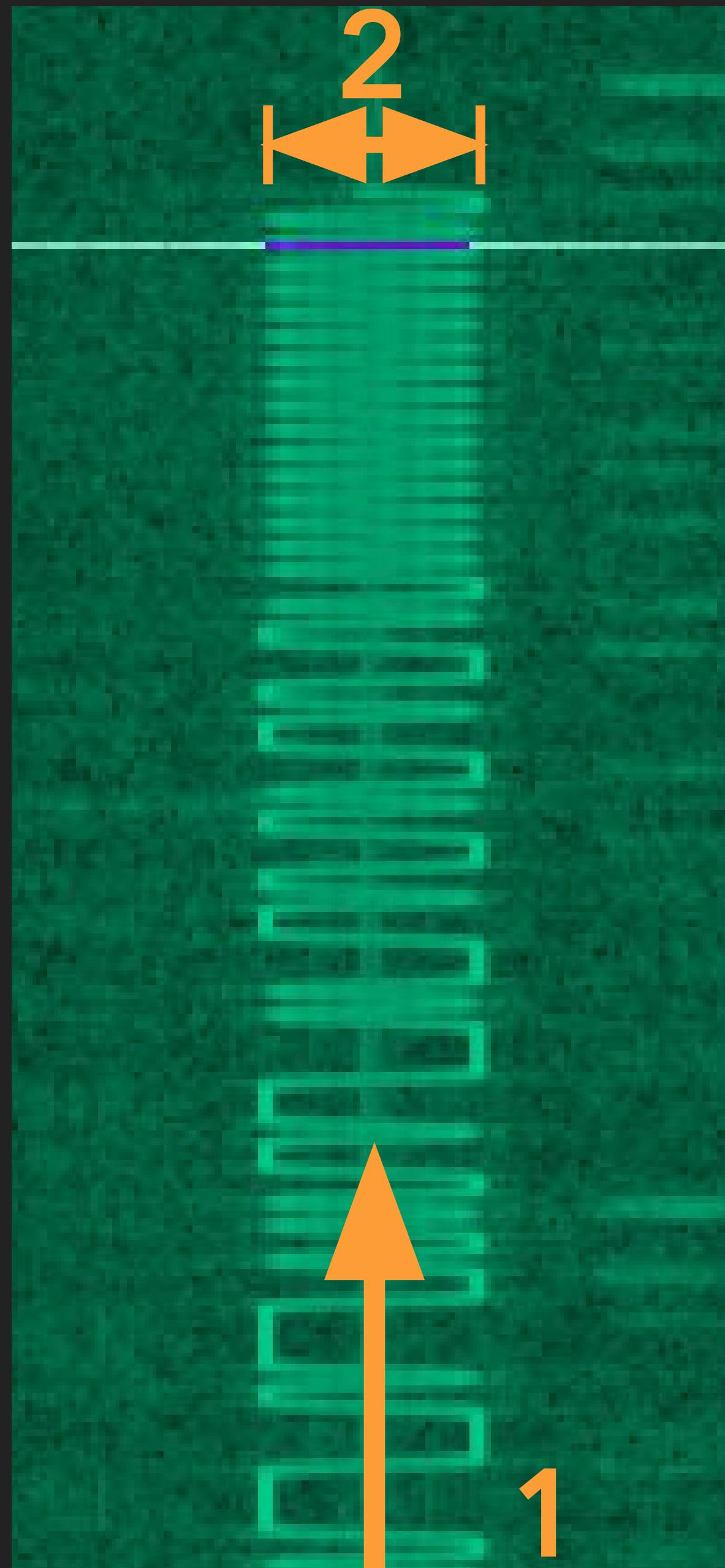
LET'S FORMALIZE  
THIS

# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel

# 1. CHANNEL CHARACTERIZATION

- ▶ Things to identify:
  1. Where on the spectrum is it? i.e. what is its Center Frequency?
  2. How wide is the channel? (kHz or MHz)
  3. Is the channel static or does it hop? If latter, what pattern/timing?

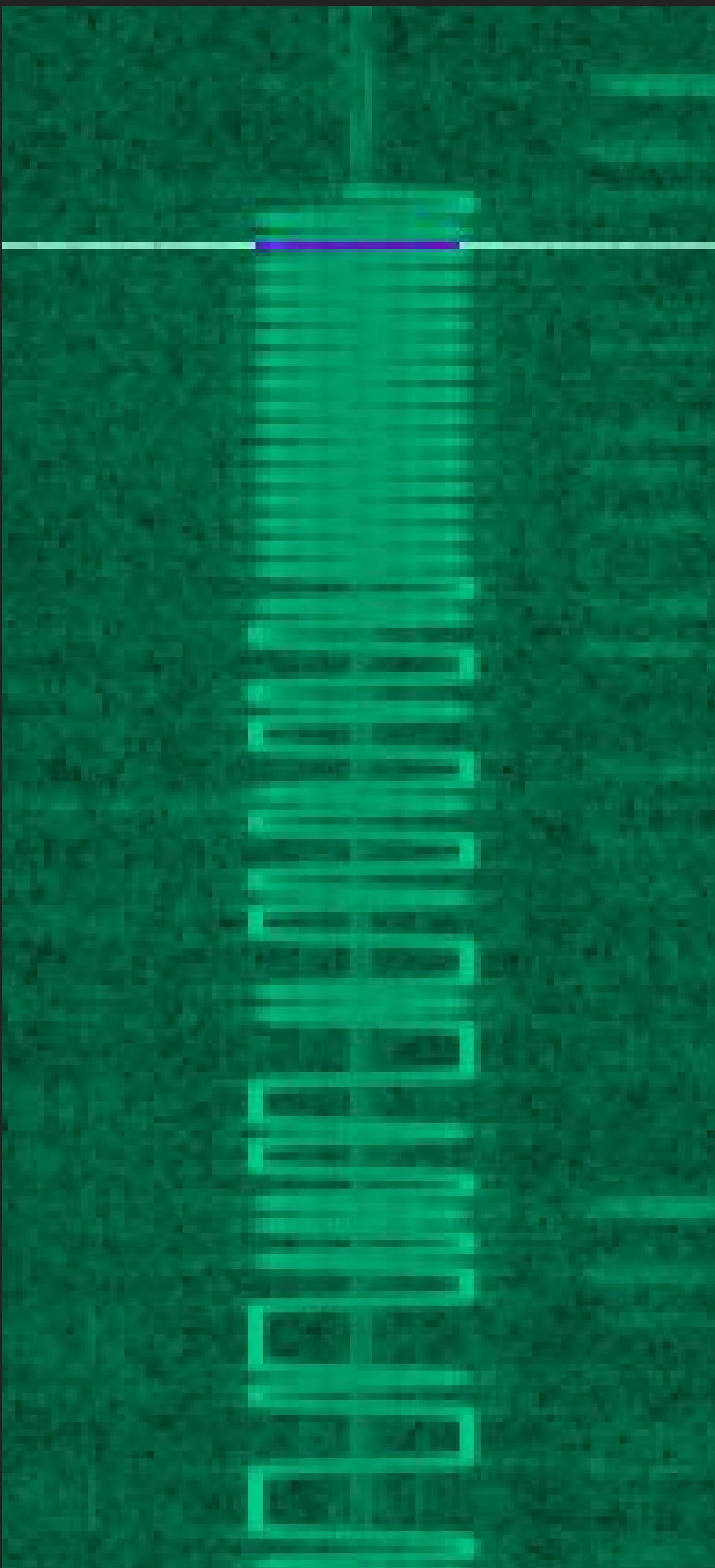


# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel
2. Identify the modulation

## 2. IDENTIFY THE MODULATION

- ▶ Defines how data is mapped to RF energy
- ▶ This is the scariest part!
- ▶ ...until you realize that most modulations are variations on a theme
- ▶ How to identify:
  1. OSINT/Documentation
  2. Intuition!

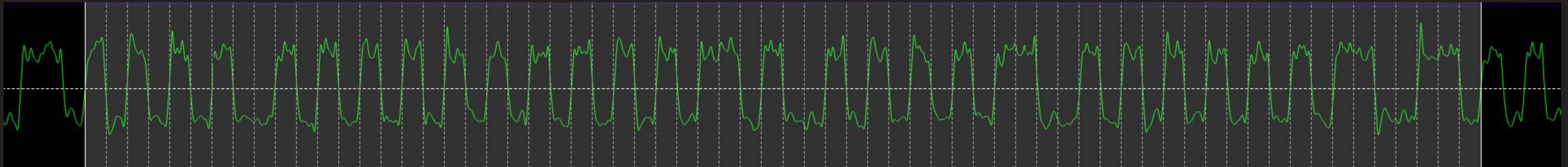


# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate

### 3. DETERMINE SYMBOL RATE

- ▶ How often does the symbol state change?



- ▶ How to identify:
  - ▶ OSINT/Documentation
  - ▶ Measurement (Baudline, Inspectrump)

**Time selection**  
Enable cursors:

Symbols:

Rate: 291.036Hz

Period: 3.436ms

Symbol rate: 19.2084kHz  

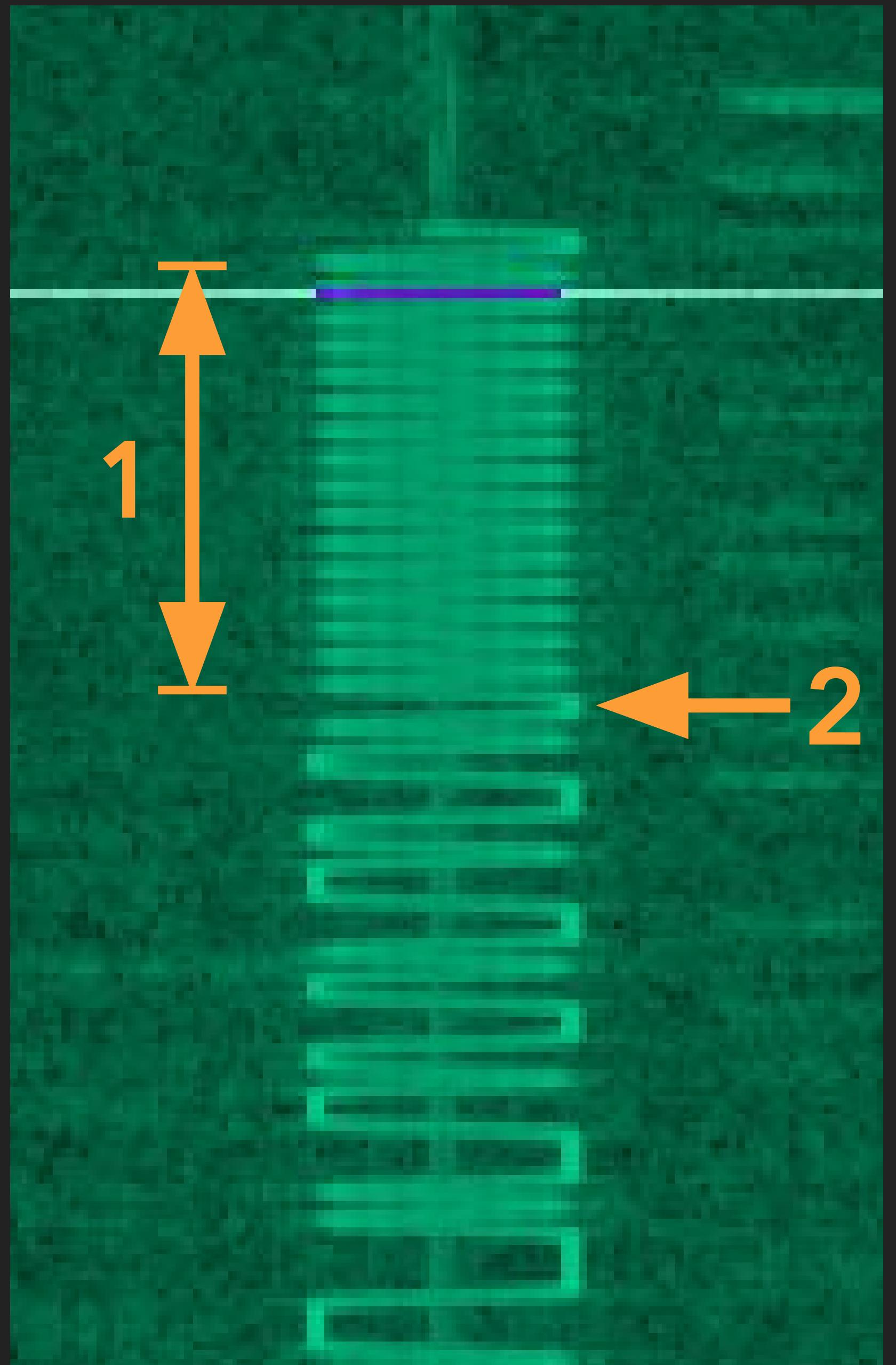
Symbol period: 52.0606μs

# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize

## 4. SYNCHRONIZE

- ▶ Things to identify:
  1. Preamble: pattern that tells receivers “data to follow”, clock recovery
  2. Start of Frame Delimiter (SFD): tells receiver “preamble is over, data follows from here on out”
- ▶ These are present in essentially **ALL** digital communication schemes!



# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

## 5. EXTRACT SYMBOLS

- ▶ De-map symbols into data based on the expected modulation topology
- ▶ Profit! (more on this later)

# RF REVERSE ENGINEERING METHODOLOGY

1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

LET'S SEE IT IN  
ACTION

BUT FIRST

A word on

# OPEN SOURCE INTELLIGENCE

# OPEN SOURCE INTELLIGENCE (OSINT)

- ▶ Information gleaned from public sources:
  - ▶ FCC/regulatory filing documents
  - ▶ Technical documentation (datasheets, application notes)
  - ▶ Patents
  - ▶ etc.
- ▶ See Marc's prior talks on OSINT from FCC filings

# RF REVERSE ENGINEERING METHODOLOGY

0. Open-source intelligence research
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols



Frequency Shift Keying

---

Z-WAVE



# HOME AUTOMATION PROTOCOL

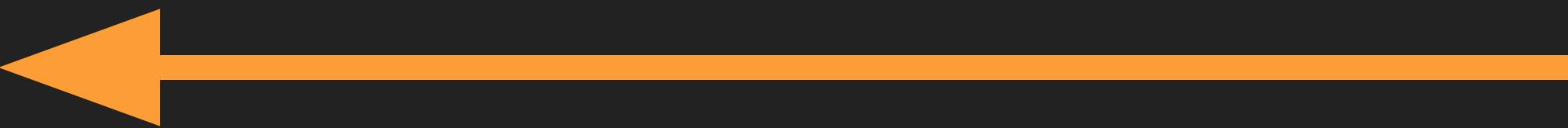
# FULL STACK

# Z-WAVE HOME AUTOMATION SYSTEM

- ▶ Competes with ZigBee Home Automation cluster library
- ▶ Full stack mesh networking protocol, from PHY to application
- ▶ Totally closed source!
- ▶ Let's build a PHY to enable analysis of the upper layers

# Z-WAVE: RF REVERSE ENGINEERING METHODOLOGY

0. Open-source intelligence research
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols



# Z-Wave Device FCCID



# FCC ID U2Z45602-3 Test Photos

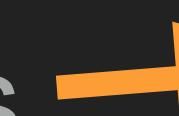


# FCC Test Report EUT Description

## 3.1. EUT Description

Description	: Dimmer lamp module & Relay fluorescent light appliance module
Manufacturer	: SHEENWAY ASIA LTD
Model Number	: ZDP100 45602 ZRP100 45603
Input Power	: Input : AC <u>120V</u> , <u>60</u> Hz, Output: AC <u>300W</u> (Incandescent)/ <u>1500W</u> (resistive)--ZDP100 45602 Input: AC <u>120V</u> , <u>60Hz</u> , Output: AC <u>600W</u> (Incand.)/1800W or 15A(resistive)--ZRP100 45603
Operate Frequency	: 908.42MHz
Modulation	: FSK
Antenna Designation	: integrated

Channel and  
modulation clues



the model No. ZDP100 and the other model No.ZRP100 which are certified are identical in all aspects except for model name, one terminals of outlet, and one is dimmer lamp module, 300W(ZDP100) and other is relay fluorescent light & appliance module,600W(ZRP100).

ZDP100 and 45602, ZRP100 and 45603 are identical in schematic, structure and critical components except for model number, which vary with different customer.

Good start...  
Let's see what  
else we can find

# FCC Reports from Z-Wave IC Manufacturer

**11 results were found that match the search criteria:**

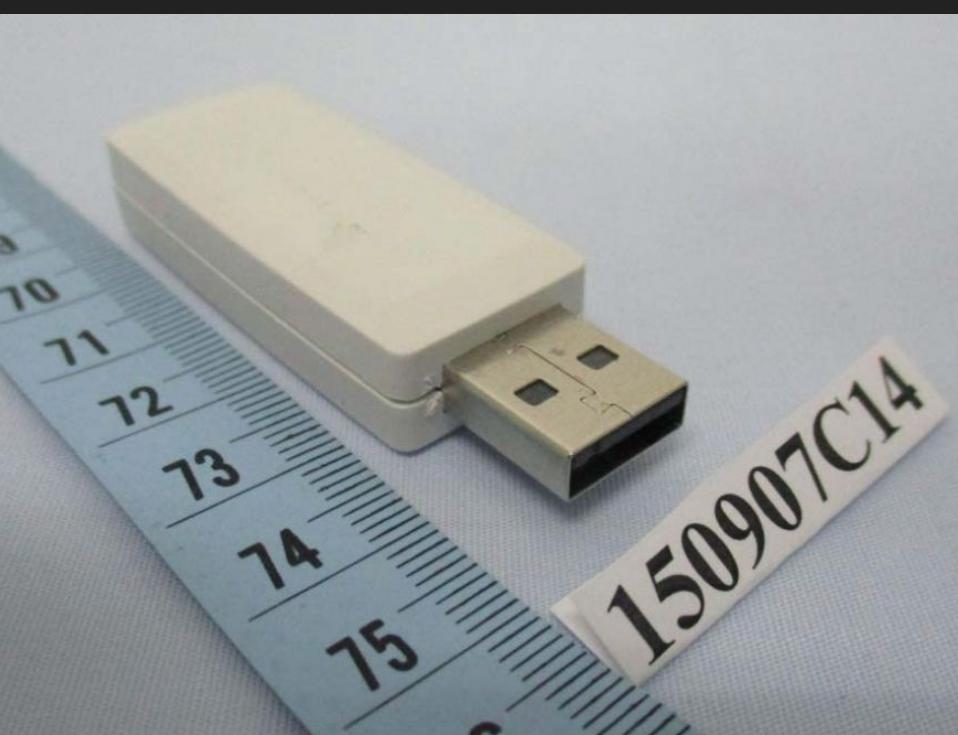
**Applicant Name: sigma designs Lower Frequency: 900 Upper Frequency: 930**

**Displaying records 1 through 10 of 11.**

View Form	Display Exhibits	Display Grant	Display Correspondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-UZB3-HSG	Original Equipment	06/16/2016	920.9	923.1
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-UZB3-U	Original Equipment	10/26/2015	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-ZIRC3502	Original Equipment	08/23/2013	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-UZB3503	Original Equipment	08/23/2013	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-ZIRC	Original Equipment	09/06/2011	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-ZIRC	Original Equipment	09/06/2011	916.0	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-ZM5304-U	Original Equipment	08/23/2013	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-ZIPR3503	Original Equipment	08/23/2013	908.4	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-ZIPR2-U	Original Equipment	02/25/2016	908.42	916.0
<input type="checkbox"/>	<a href="#">Detail Summary</a>	<input checked="" type="checkbox"/>		Sigma Designs Inc	47467 Fremont Blvd.	Fremont	CA	United States	94538	D87-SG-UZB	Original Equipment	03/13/2012	908.4	916.0

[Show Next 10 Rows](#)

# Pick an arbitrary one



## 3 General Information

### 3.1 General Description of EUT

Product	Z-Wave USB Stick
Brand	Sigma Designs
Test Model	UZB3-U
Status of EUT	Engineering sample
Power Supply Rating	5Vdc (Host equipment)
Modulation Type	2FSK (9.6kbps, 40kbps) / 2GFSK (100kbps)
Transfer Rate	9.6kbps, 40kbps, 100kbps
Operating Frequency	908.42MHz, 908.4MHz, 916MHz
Number of Channel	3
Antenna Type	Helical antenna with -1.13dBi gain
Accessory Device	N/A
Data Cable Supplied	N/A

Note:

1. The above EUT information is declared by manufacturer and for more detailed features description, please refer to the manufacturer's specifications or user's manual.

# Z-Wave Channel Mapping

## 3.2 Description of Test Modes

3 channels are provided for EUT:

Channel	Frequency (MHz)	Transfer Rate (kbps)
1	908.42	9.6
2	908.40	40
3	916.00	100

### Radiated Emission Test (Below 1GHz):

- Pre-Scan has been conducted to determine the worst-case mode from all possible combinations between available modulations, data rates and antenna ports (if EUT with antenna diversity architecture).
- Following channel(s) was (were) selected for the final test as listed below.

TESTED CHANNEL	MODULATION TECHNOLOGY	MODULATION TYPE
1	908.42MHz	2FSK
2	908.40MHz	2FSK
3	916.00MHz	2GFSK

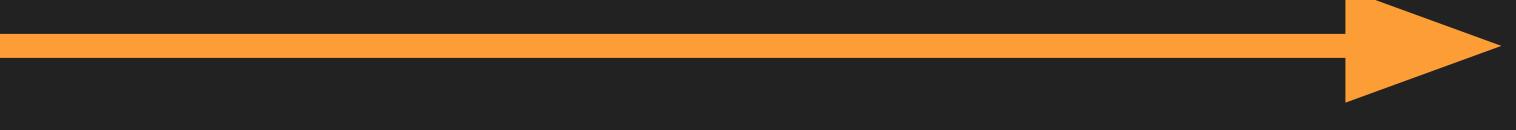
## 0. OSINT

Looking at the 9.6 kbps @ 908.42 MHz channel

- ▶ Frequency: 908.42 MHz
  - ▶ Modulation: FSK
  - ▶ Deviation: +/- 20 kHz
  - ▶ Bit rate: 9600 bits/s
- 
- The diagram consists of four orange arrows originating from the right side of the slide and pointing towards the left. The top arrow points to the text "1. Channel". The second arrow from the top points to the text "2. Modulation". The third arrow from the top points to the text "3. Symbol Rate". The bottom arrow points to the text "OSINT leads to clues for first 3 steps".
1. Channel
  2. Modulation
  3. Symbol Rate

OSINT leads to clues for first 3 steps

# Validating OSINT

- ▶ Frequency: 908.42 MHz  Measure center frequency
- ▶ Modulation: FSK  Visually confirm
- ▶ Deviation: ? kHz  Measure width of channel
- ▶ Bit rate: 9600 bits/s  Measure symbol timing

[INTERACTIVE]

# Validating Symbol Rate

Inspectum



Time selection

Enable cursors:

Symbols:

Rate: 291.036Hz

Period: 3.436ms

Symbol rate: **19.2084kHz**

Symbol period: 52.0606μs

2x expected bit rate (9600 bits/s)

Manchester encoding!

# Manchester Encoding

Data Bits (un-encoded)	Manchester Bits (encoded)
0b0	0b01
0b1	0b10
(illegal state)	0b00
(illegal state)	0b11

Result: encoded bitstream has no more than 2 adjacent symbols with the same value

0b0000 → 0b01010101

0b1111 → 0b10101010

Benefit: lots of symbol changes for receivers to perform **clock recovery/synchronization** against

Cost: restricts bit rate to  $\frac{1}{2}$  baud rate (symbol rate)

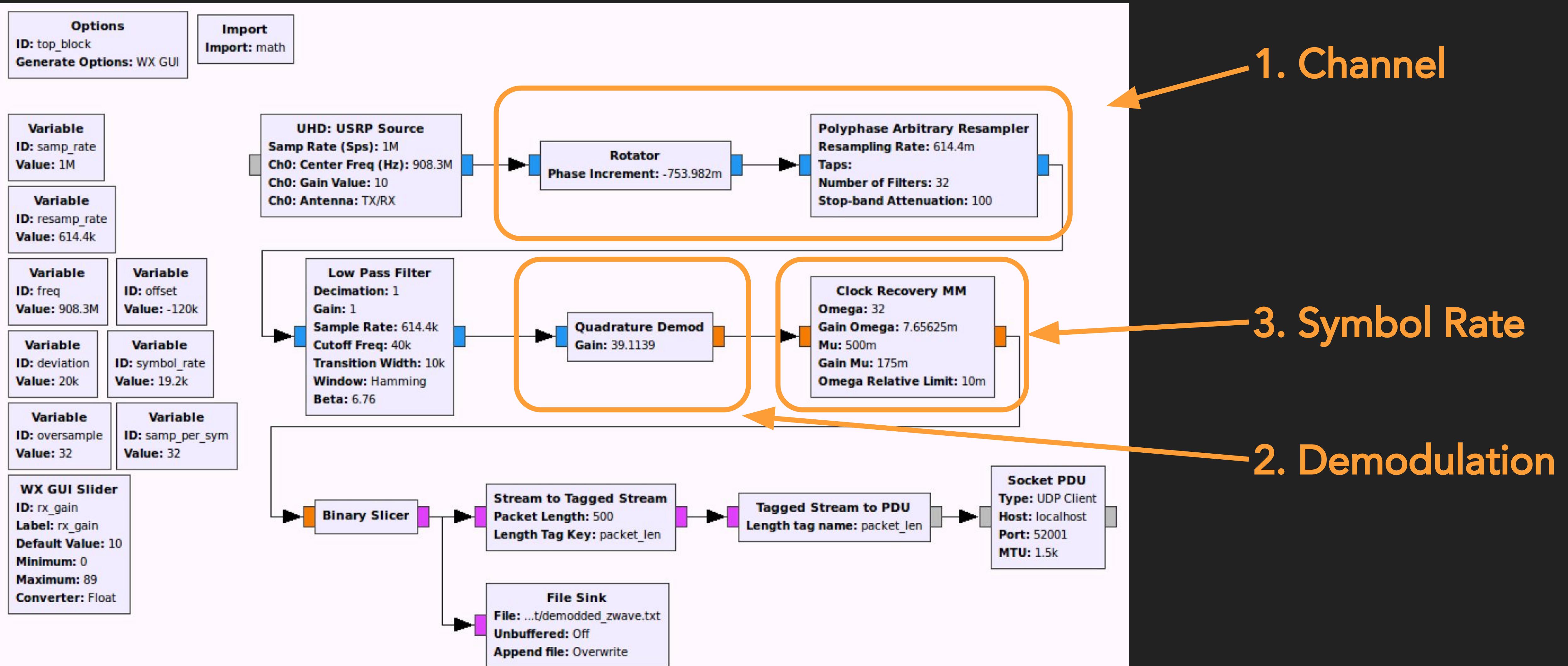
## 0. OSINT

- ▶ Frequency: 908.42 MHz
  - ▶ Modulation: FSK
  - ▶ Deviation: +/- 20 kHz
  - ▶ **Symbol Rate**
  - ▶ ~~Bit rate~~: 9600 bits/s → 19,200 bits/s OTA due to encoding
- 
- The diagram consists of three orange arrows originating from the right side of the slide and pointing towards the left. The top arrow is labeled "1. Channel". The middle arrow is labeled "2. Modulation". The bottom arrow is labeled "3. Symbol Rate".

# Z-WAVE: RF REVERSE ENGINEERING METHODOLOGY

- 0. ✓ Open-source intelligence research
  - 1. ✓ Characterize the channel
  - 2. ✓ Identify the modulation
  - 3. ✓ Determine the symbol rate
  - 4. Synchronize
  - 5. Extract symbols
- 
- GNU Radio Flowgraph to produce a **stream of symbols**
- Python scripting to **parse symbols into data**

# Translate OSINT into GNU Radio Flowgraph



## 4. Synchronization and 5. Symbol Extraction

1. Look for preamble
2. Look for SFD to synchronize
3. Read out frame and de-Manchester. Frame length determined by:
  - a. Preconfigured MTU size
  - b. Power squelch (FSK is constant envelope)
  - c. Decoding failure (i.e. Manchester decoding hits an illegal state)
  - d. Decoded length field
4. Parse frame

# Demo Time!

On-Off Keying / Pulse-Width Modulation

---

**WIRELESS DOORBELL**



## HeathZenith SL-7762

- ▶ Wireless Doorbell
- ▶ Battery operated
- ▶ Two transmitters (buttons)
- ▶ FCC ID BJ4-WLTX201
- ▶ One receiver (chime)
- ▶ Receive-only, no FCC ID

# DOORBELL FCC EXHIBITS

10 Matches found for FCC ID **BJ4-WLTX201**

<u><a href="#">View Attachment</a></u>	<u><a href="#">Exhibit Type</a></u>	<u><a href="#">Date Submitted to FCC</a></u>	<u><a href="#">Display Type</a></u>	<u><a href="#">Date Available</a></u>
<u><a href="#">Letter of Agency</a></u>	Cover Letter(s)	07/17/2014	pdf	07/17/2014
<u><a href="#">Confidentiality Request</a></u>	Cover Letter(s)	07/17/2014	pdf	07/17/2014
<u><a href="#">External Photos</a></u>	External Photos	07/17/2014	pdf	07/17/2014
<u><a href="#">Label Artwork and Location</a></u>	ID Label/Location Info	07/17/2014	pdf	07/17/2014
<u><a href="#">Internal Photos</a></u>	Internal Photos	07/17/2014	pdf	07/17/2014
<u><a href="#">Analysis Report</a></u>	RF Exposure Info	07/17/2014	pdf	07/17/2014
<u><a href="#">Test Report</a></u>	Test Report	07/17/2014	pdf	07/17/2014
<u><a href="#">Timing</a></u>	Test Report	07/17/2014	pdf	07/17/2014
<u><a href="#">Radiated Emission</a></u>	Test Setup Photos	07/17/2014	pdf	07/17/2014
<u><a href="#">User Manual</a></u>	Users Manual	07/17/2014	pdf	07/17/2014

# DOORBELL FCC TEST REPORT

- ▶ 315MHz center frequency

## 1.1 Product Description

The equipment under test (EUT) is a transmitter for Remote door bell operating at 315MHz which is operated by a crystal. The EUT is powered by 1 x 3.0V CR2032 button cell. The EUT has one control key, press the control key on the EUT in order to control the desired door bell receiver. This manually transmitter will automatically deactivate the transmitter within not more than 5 seconds of being released.

# DOORBELL FCC TEST REPORT

- ▶ 315MHz center frequency

## 1.1 Product Description

The equipment under test (EUT) is a transmitter for Remote door bell operating at **315MHz** which is operated by a crystal. The EUT is powered by 1 x 3.0V CR2032 button cell. The EUT has one control key, press the control key on the EUT in order to control the desired door bell receiver. This manually transmitter will automatically deactivate the transmitter within not more than 5 seconds of being released.

# DOORBELL FCC TEST REPORT

- ▶ 320us duration bit 1
- ▶ 13 bits per packet
- ▶ 25.48ms packet spacing
- ▶ ~30% duty cycle

## 8.2 Discussion of Pulse Desensitization

Pulse desensitivity is not applicable for this device. The effective period ( $T_{eff}$ ) is approximately 0.32ms for a digital "1" bit which illustrated on technical specification, with a resolution bandwidth (3dB) of 1MHz, so the pulse desensitivity factor is 0dB.

## 8.3 Calculation of Average Factor

The duty cycle is simply the on-time divided by the period:

The duration of one cycle =  $0.32\text{ms} \times 5 + 0.72\text{ms} \times 8 = 7.36\text{ms}$

Effective period of the cycle = 25.48ms

$$\text{DC} = (7.36\text{ms}) / 25.48\text{ms} = 0.2889$$

Therefore, the averaging factor is found by  $20\log(0.2889) = -10.8\text{dB}$ .

# DOORBELL FCC TEST REPORT

- ▶ 320us duration bit 1
- ▶ 13 bits per packet
- ▶ 25.48ms packet spacing
- ▶ ~30% duty cycle

## 8.2 Discussion of Pulse Desensitization

Pulse desensitivity is not applicable for this device. The effective period ( $T_{eff}$ ) is approximately 0.32ms for a digital "1" bit which illustrated on technical specification, with a resolution bandwidth (3dB) of 1MHz, so the pulse desensitivity factor is 0dB.

## 8.3 Calculation of Average Factor

The duty cycle is simply the on-time divided by the period:

The duration of one cycle =  $0.32\text{ms} \times 5 + 0.72\text{ms} \times 8 = 7.36\text{ms}$

Effective period of the cycle = 25.48ms

$$\text{DC} = (7.36\text{ms}) / 25.48\text{ms} = 0.2889$$

Therefore, the averaging factor is found by  $20\log(0.2889) = -10.8\text{dB}$ .

# DOORBELL FCC TEST REPORT

- ▶ 320us duration bit 1
- ▶ 13 bits per packet
- ▶ 25.48ms packet spacing
- ▶ ~30% duty cycle

## 8.2 Discussion of Pulse Desensitization

Pulse desensitivity is not applicable for this device. The effective period ( $T_{eff}$ ) is approximately 0.32ms for a digital "1" bit which illustrated on technical specification, with a resolution bandwidth (3dB) of 1MHz, so the pulse desensitivity factor is 0dB.

## 8.3 Calculation of Average Factor

The duty cycle is simply the on-time divided by the period:

$$\text{The duration of one cycle} = 0.32\text{ms} \times 5 + 0.72\text{ms} \times 8 = 7.36\text{ms}$$

Effective period of the cycle = 25.48ms

$$DC = (7.36\text{ms}) / 25.48\text{ms} = 0.2889$$

Therefore, the averaging factor is found by  $20\log (0.2889) = -10.8\text{dB}$ .

# DOORBELL FCC TEST REPORT

- ▶ 320us duration bit 1
- ▶ 13 bits per packet
- ▶ 25.48ms packet spacing
- ▶ ~30% duty cycle

## 8.2 Discussion of Pulse Desensitization

Pulse desensitivity is not applicable for this device. The effective period ( $T_{eff}$ ) is approximately 0.32ms for a digital "1" bit which illustrated on technical specification, with a resolution bandwidth (3dB) of 1MHz, so the pulse desensitivity factor is 0dB.

## 8.3 Calculation of Average Factor

The duty cycle is simply the on-time divided by the period:

$$\text{The duration of one cycle} = 0.32\text{ms} \times 5 + 0.72\text{ms} \times 8 = 7.36\text{ms}$$

$$\text{Effective period of the cycle} = 25.48\text{ms}$$

$$DC = (7.36\text{ms}) / 25.48\text{ms} = 0.2889$$

Therefore, the averaging factor is found by  $20\log (0.2889) = -10.8\text{dB}$ .

# DOORBELL FCC TEST REPORT

- ▶ 320us duration bit 1
- ▶ 13 bits per packet
- ▶ 25.48ms packet spacing
- ▶ ~30% duty cycle

## 8.2 Discussion of Pulse Desensitization

Pulse desensitivity is not applicable for this device. The effective period ( $T_{eff}$ ) is approximately 0.32ms for a digital "1" bit which illustrated on technical specification, with a resolution bandwidth (3dB) of 1MHz, so the pulse desensitivity factor is 0dB.

## 8.3 Calculation of Average Factor

The duty cycle is simply the on-time divided by the period:

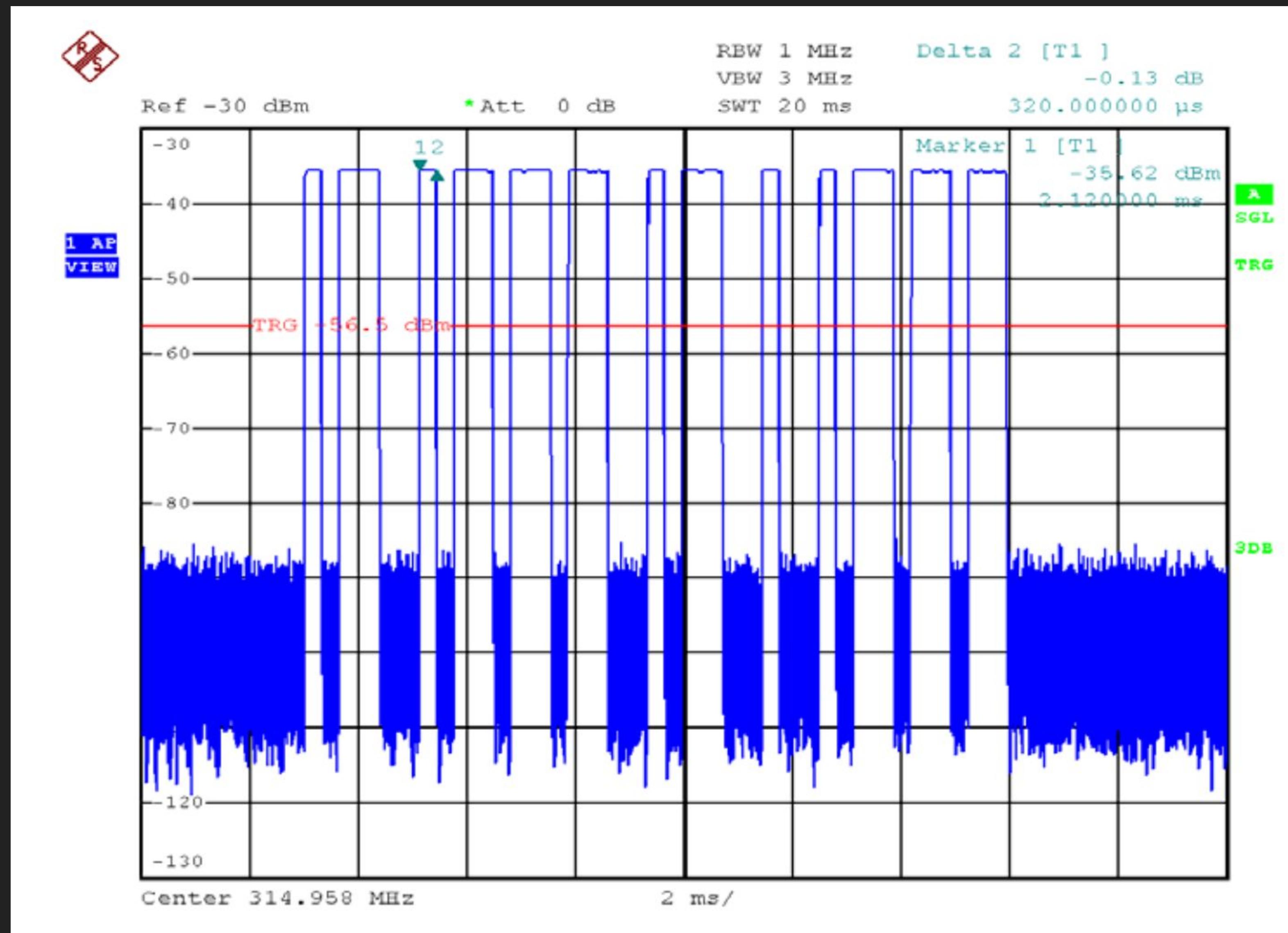
$$\text{The duration of one cycle} = 0.32\text{ms} \times 5 + 0.72\text{ms} \times 8 = 7.36\text{ms}$$

$$\text{Effective period of the cycle} = 25.48\text{ms}$$

$$DC = (7.36\text{ms}) / 25.48\text{ms} = 0.2889$$

Therefore, the averaging factor is found by  $20\log (0.2889) = -10.8\text{dB}$ .

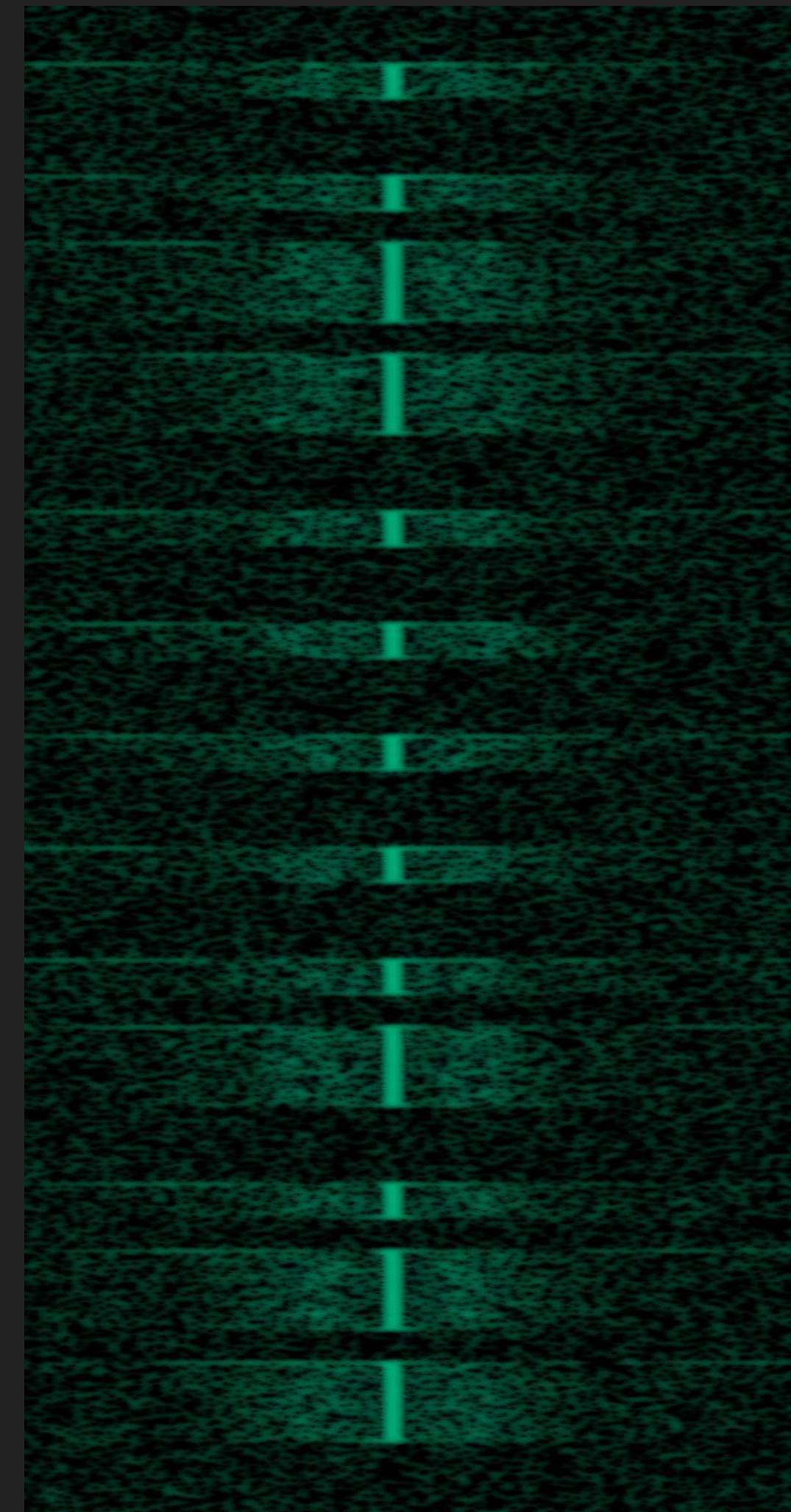
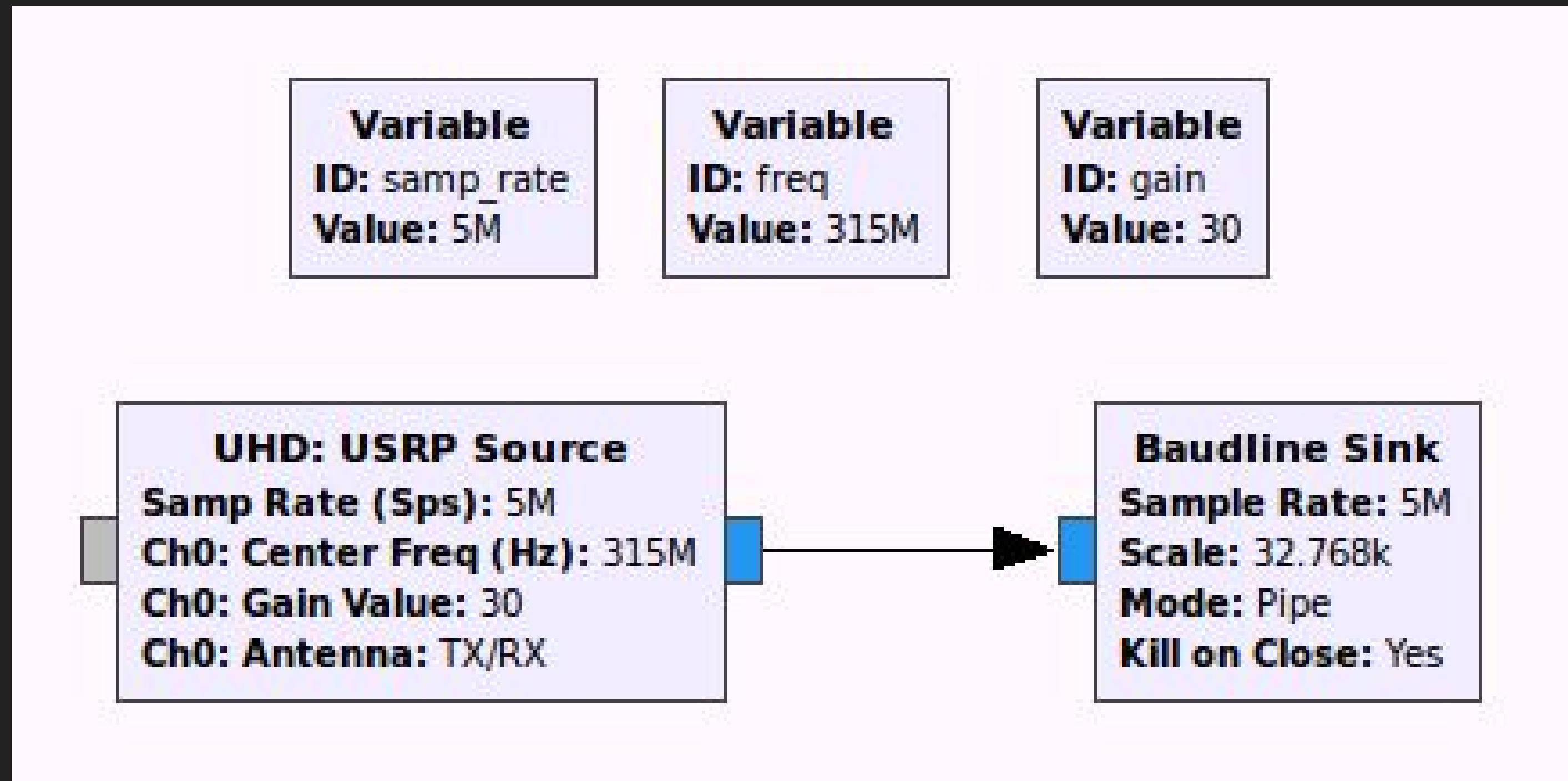
# SO YOU WANT TO HACK RADIOS // BASTILLE NETWORKS



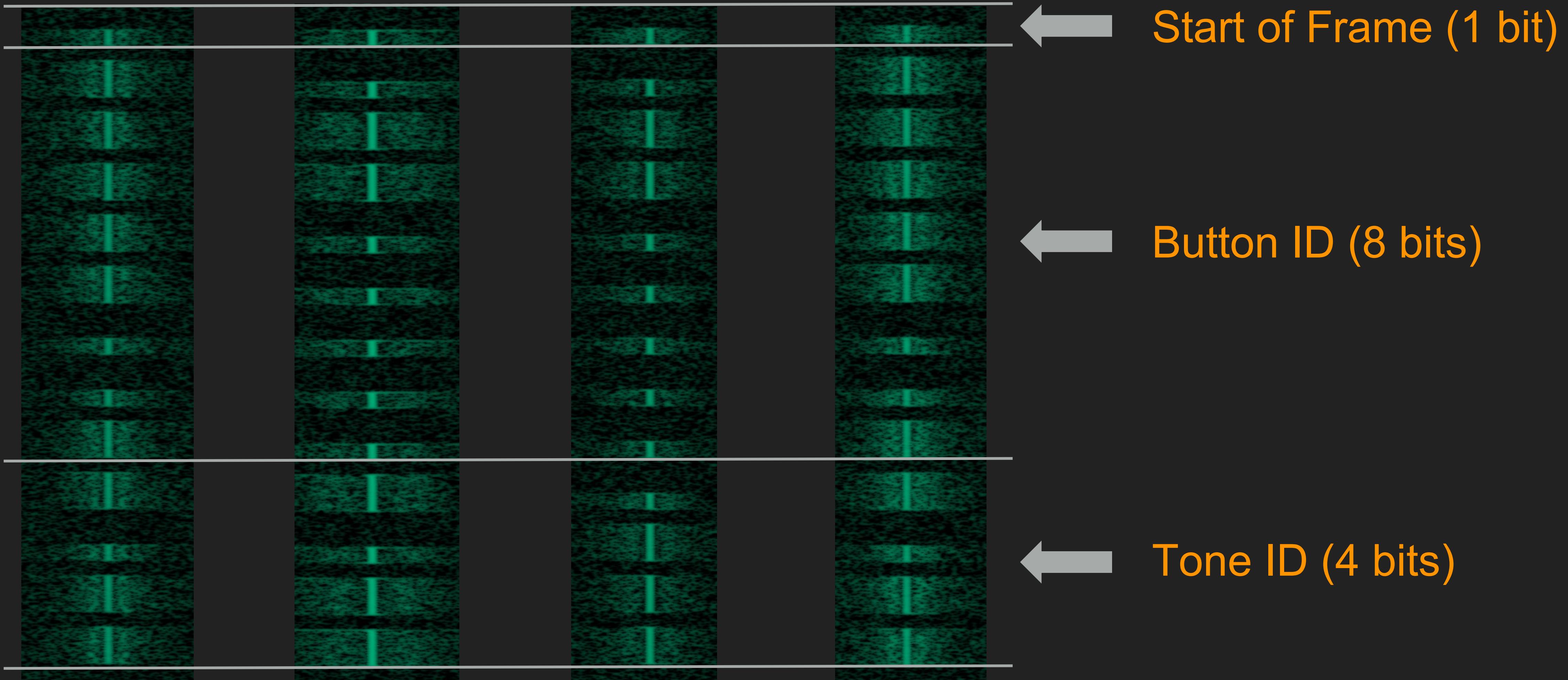
## LOOK AT SIMILAR PRODUCTS

This custom IC is similar to the HT-12E encoder used in the previously certified version (SL-6194-TX) and produces a serial bit stream that corresponds to the state of its address and data control lines. The data rate is approximately 1 kHz and the pattern consists of 8 address bits, 4 data bits and 1 “start” bit (a 13 bit information block). The logic data high bit (one) is represented by a 600 uS pulse-width and a logic low bit (zero) by a 300 uS pulse-width. A minimum of four 13 bit information blocks are sent (transmitted) each time the push button is pressed and will repeat while the switch is held down.

# OSINT SANITY CHECK



# BUTTON WAVEFORMS IN BAUDLINE



# WHAT DID WE LEARN FROM OSINT?

- ▶ 315MHz center frequency [channel]
- ▶ Pulse width modulation [modulation]
- ▶ 1KHz data rate [symbol timing]
- ▶ Bit 1 is ~700us off and ~300us on
- ▶ Bit 0 us ~300us off and ~700us on
- ▶ Packets are 13 bits long [synchronize]
  - ▶ 1 “start bit”
  - ▶ 8 button ID bits
  - ▶ 4 tone ID bits

**DOORBELL  
DEMOS**

this image is  
1000 x 1337  
pixels



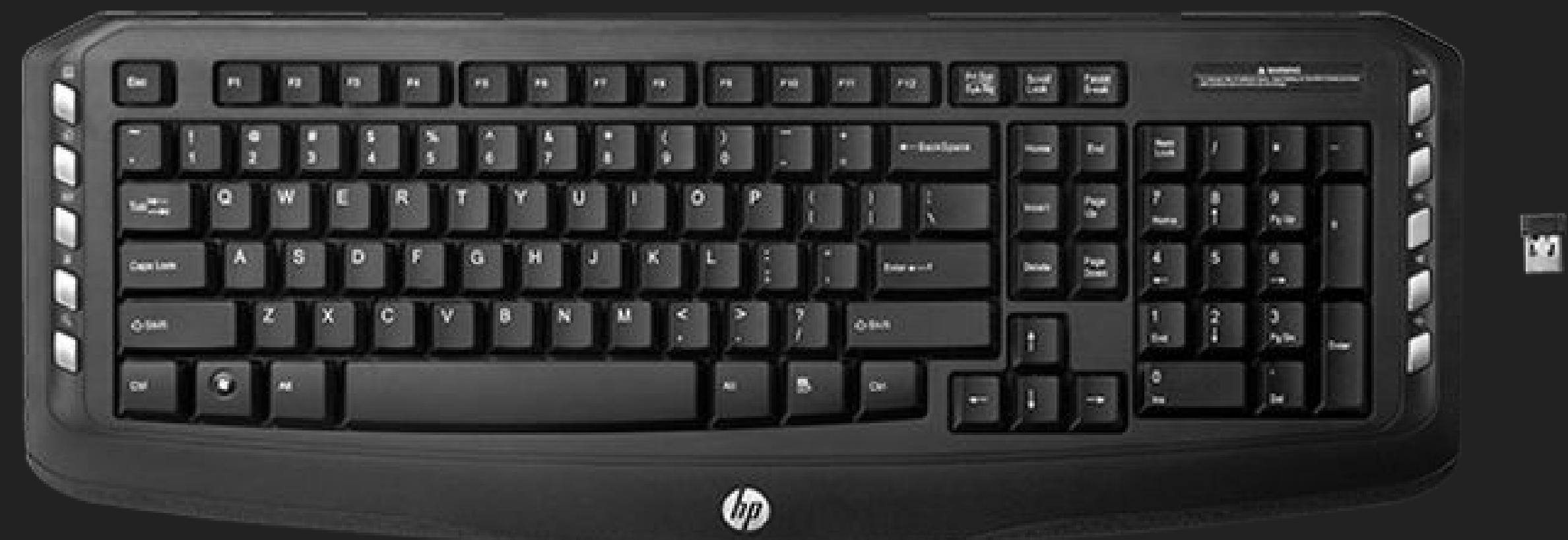
TDMA Frequency Shift Keying

---

HP KEYBOARD

# HP CLASSIC WIRELESS DESKTOP

- ▶ 2.4GHz Wireless Keyboard/Mouse
- ▶ OEM = ACROX
- ▶ Keyboard
  - ▶ FCC ID PRDKB14
- ▶ Mouse
  - ▶ FCC ID PRDMU26
- ▶ Dongle
  - ▶ FCC ID PRDRX02



# HP DONGLE TEST REPORT

EUT	2.4GHz Receiver
MODEL NO.	MRN
FCC ID	PRDRX02
POWER SUPPLY	5Vdc (host equipment)
MODULATION TYPE	GFSK
DATA RATE	1M bit/sec.
OPERATING FREQUENCY	2403MH~2480MHz
NUMBER OF CHANNEL	78
ANTENNA TYPE	Printed antenna
DATA CABLE	NA
I/O PORT	USB
ACCESSORY DEVICES	NA

# HP KEYBOARD TEST REPORT

## 1.1.1 Product Details

The following brands are provided to this EUT.

Brand Name	Model Name	Product Name	Description
ACROX			
HP	KBIM, K2BM	HP Wireless Keyboard K2500	Marketing purpose

## 1.1.2 Specification of the Equipment under Test (EUT)

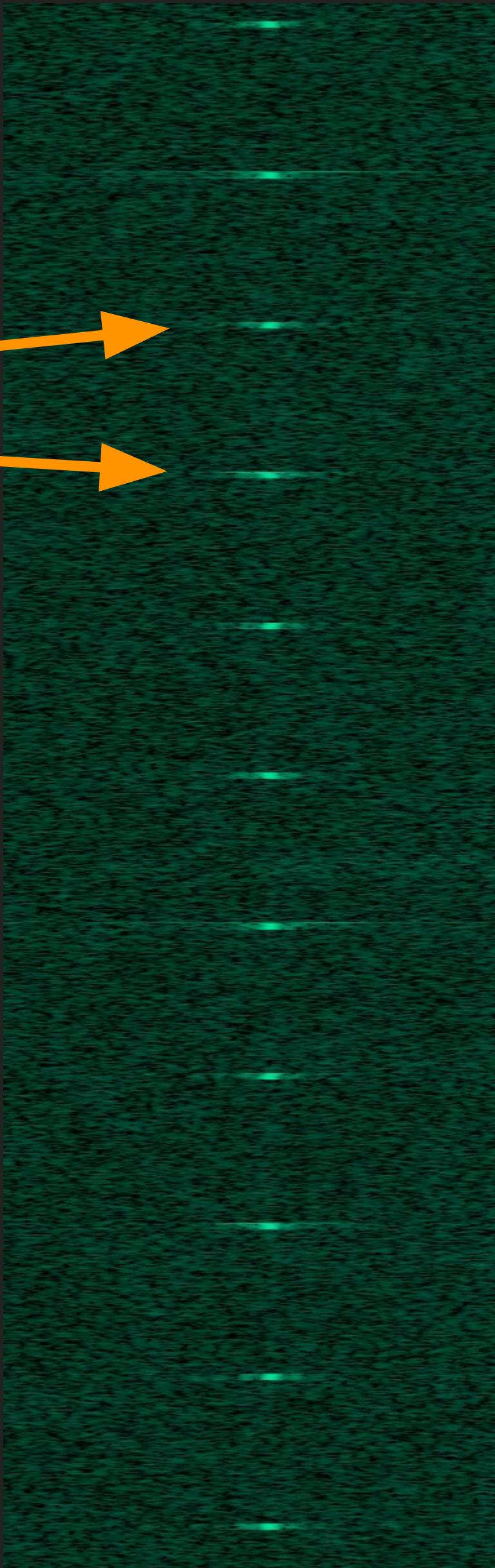
RF General Information				
Frequency Range (MHz)	Modulation	Ch. Freq. (MHz)	Channel Number	Channel Bandwidth (MHz)
2400-2483.5	FSK	2408-2474	1-34 [34]	2

# HP DONGLE DMESG OUTPUT

```
[ +0.276333] usb 1-3.1: new full-speed USB device number 21 using xhci_hcd
[ +0.091959] usb 1-3.1: New USB device found, idVendor=3938, idProduct=1032
[ +0.000012] usb 1-3.1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[ +0.000008] usb 1-3.1: Product: 2.4G RF Keyboard & Mouse
[ +0.000007] usb 1-3.1: Manufacturer: MOSART Semi.
[ +0.000470] usb 1-3.1: ep 0x81 - rounding interval to 64 microframes, ep desc says 80 microframes
[ +0.002402] input: MOSART Semi. 2.4G RF Keyboard & Mouse as /devices/pci0000:00/0000:00:14.0/usb1/:
[ +0.054089] hid-generic 0003:3938:1032.0009: input,hidraw2: USB HID v1.10 Keyboard [MOSART Semi. 2
[ +0.004330] input: MOSART Semi. 2.4G RF Keyboard & Mouse as /devices/pci0000:00/0000:00:14.0/usb1/:
[ +0.055401] hid-generic 0003:3938:1032.000A: input,hiddev0,hidraw3: USB HID v1.10 Mouse [MOSART Ser
```

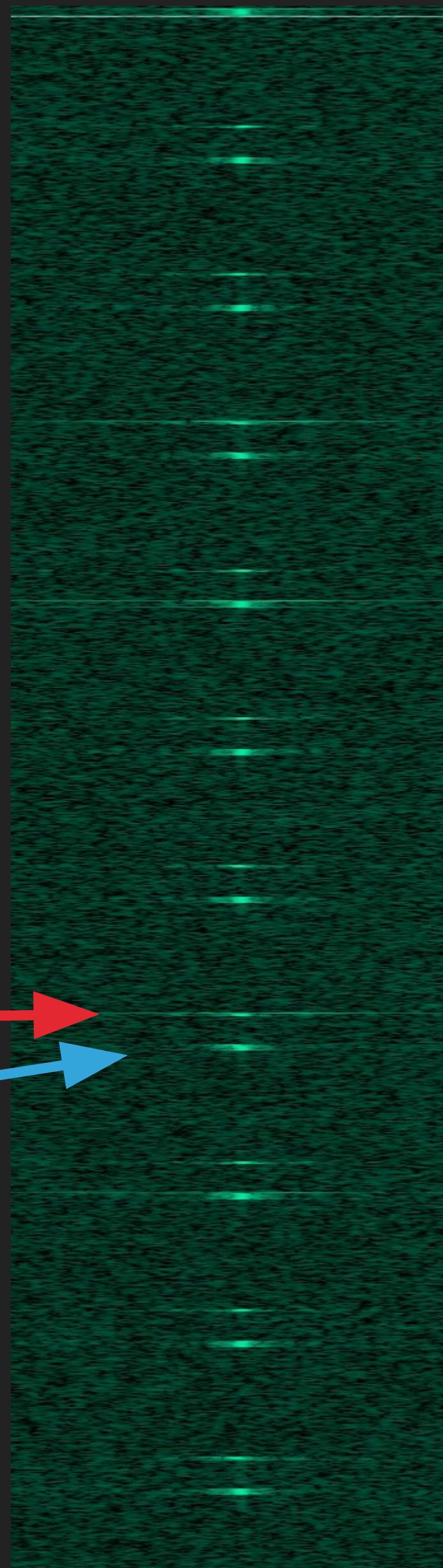
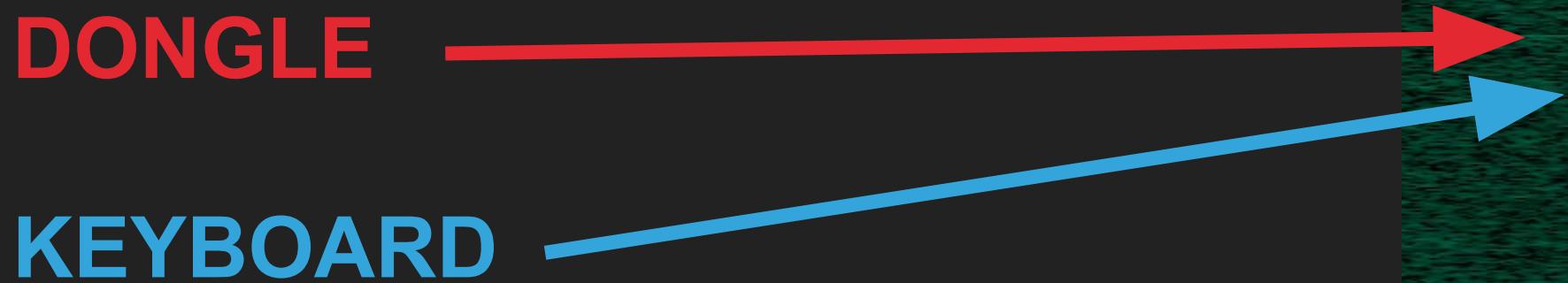
# DONGLE IN BAUDLINE

- ▶ Always transmitting at 8ms intervals
- ▶ No channel hopping
- ▶ TDMA? (Time Division Multiple Access)

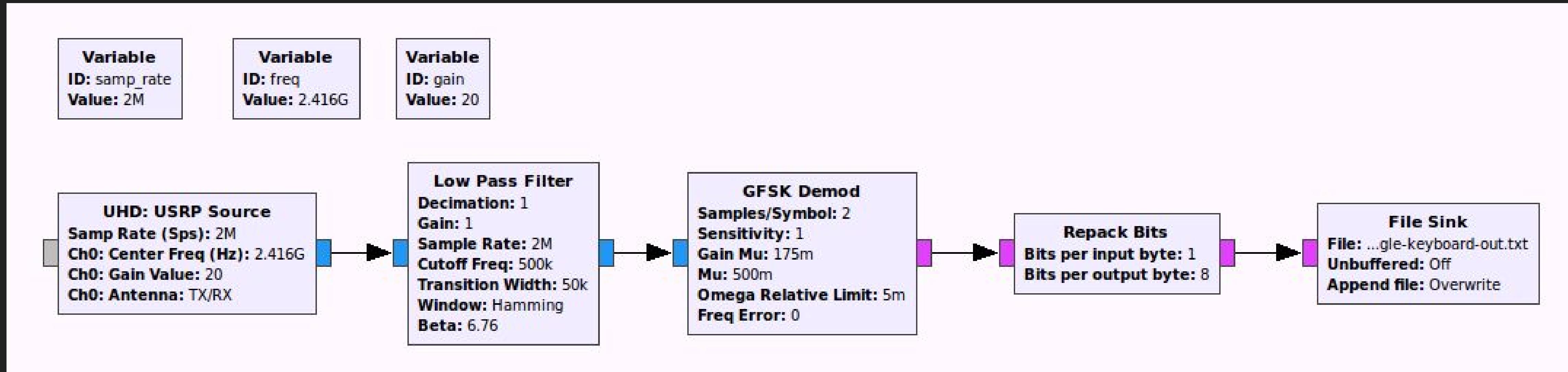


# KEYBOARD IN BAUDLINE

- ▶ Keystrokes follow dongle packets by 2ms
- ▶ Keyboard transmits up to every 8ms
- ▶ Dongle behavior doesn't change



# KEYBOARD DEMOD FLOWGRAPH



# GREP FOR PACKETS

```
xxd -p demod.out |  
tr -d "\n" |  
grep -Po "(00|ff|aa|55)+.{8}" |  
sort |  
uniq -c |  
sort -nr |  
Head -n 10
```

# GREP FOR PACKETS

```
xxd -p demod.out |  
tr -d "\n" |  
grep -Po "(00|ff|aa|55)+.{8}" |  
sort |  
uniq -c |  
sort -nr |  
Head -n 10
```



Bytes to Hex

# GREP FOR PACKETS

```
xxd -p demod.out |  
tr -d "\n" |  
grep -Po "(00|ff|aa|55)+.{8}" |  
sort |  
uniq -c |  
sort -nr |  
Head -n 10
```

Bytes to Hex

Grep for Packets

# GREP FOR PACKETS

```
xxd -p demod.out |  
tr -d "\n" |  
grep -Po "(00|ff|aa|55)+.{8}" |  
sort |  
uniq -c |  
sort -nr |  
Head -n 10
```

Bytes to Hex

Grep for Packets

Sort by Count

# DONGLE PACKET BYTES

fffaaaaaaaaaaaaaaaaeddd4e8

```
sed s/[dongle packets]//g
```

# KEYBOARD PACKET BYTES

aaaaaaaddd4e8

# GREP, GREP, AND GREP SOME MORE!

```
aaaaaaa ddd4e8 2e db 3f 384a
aaaaaaa ddd4e8 2d db 37 6092
aaaaaaa ddd4e8 28 db 3f 98f8
aaaaaaa ddd4e8 25 db 3f c9ba
aaaaaaa ddd4e8 25 db 21 3649
aaaaaaa ddd4e8 21 db 27 30f5
aaaaaaa ddd4e8 20 db 3f 3951
```

# GREP, GREP, AND GREP SOME MORE!

```
aaaaaaa ddd4e8 2e db 3f 384a
aaaaaaa ddd4e8 2d db 37 6092
aaaaaaa ddd4e8 28 db 3f 98f8
aaaaaaa ddd4e8 25 db 3f c9ba
aaaaaaa ddd4e8 25 db 21 3649
aaaaaaa ddd4e8 21 db 27 30f5
aaaaaaa ddd4e8 20 db 3f 3951
```



preamble

# GREP, GREP, AND GREP SOME MORE!

```
aaaaaaa ddd4e8 2e db 3f 384a
aaaaaaa ddd4e8 2d db 37 6092
aaaaaaa ddd4e8 28 db 3f 98f8
aaaaaaa ddd4e8 25 db 3f c9ba
aaaaaaa ddd4e8 25 db 21 3649
aaaaaaa ddd4e8 21 db 27 30f5
aaaaaaa ddd4e8 20 db 3f 3951
```



preamble address

# GREP, GREP, AND GREP SOME MORE!

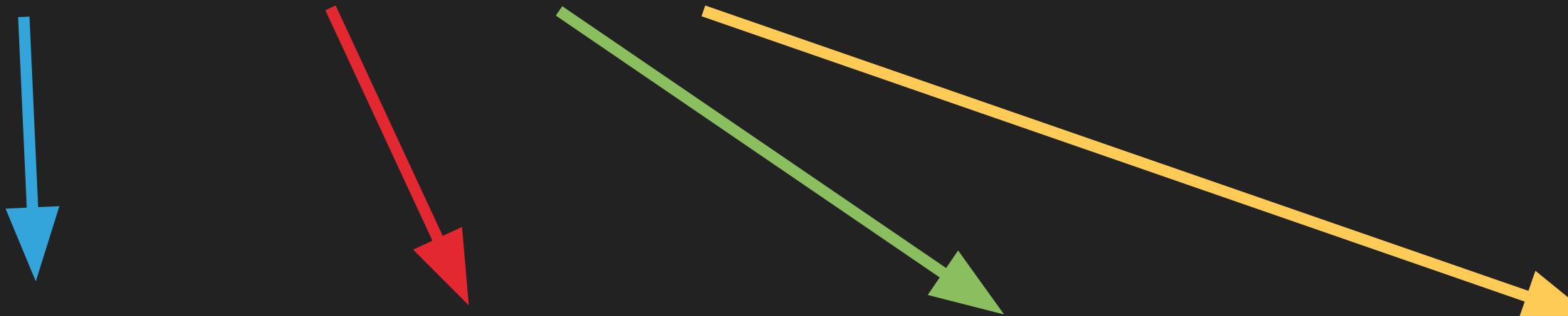
```
aaaaaaa ddd4e8 2e db 3f 384a
aaaaaaa ddd4e8 2d db 37 6092
aaaaaaa ddd4e8 28 db 3f 98f8
aaaaaaa ddd4e8 25 db 3f c9ba
aaaaaaa ddd4e8 25 db 21 3649
aaaaaaa ddd4e8 21 db 27 30f5
aaaaaaa ddd4e8 20 db 3f 3951
```



preamble address sequence

# GREP, GREP, AND GREP SOME MORE!

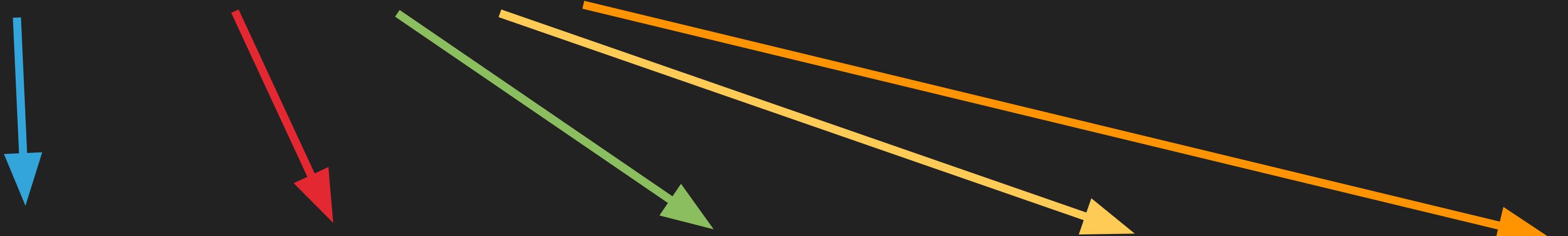
```
aaaaaaa ddd4e8 2e db 3f 384a  
aaaaaaa ddd4e8 2d db 37 6092  
aaaaaaa ddd4e8 28 db 3f 98f8  
aaaaaaa ddd4e8 25 db 3f c9ba  
aaaaaaa ddd4e8 25 db 21 3649  
aaaaaaa ddd4e8 21 db 27 30f5  
aaaaaaa ddd4e8 20 db 3f 3951
```



preamble address sequence frame type

# GREP, GREP, AND GREP SOME MORE!

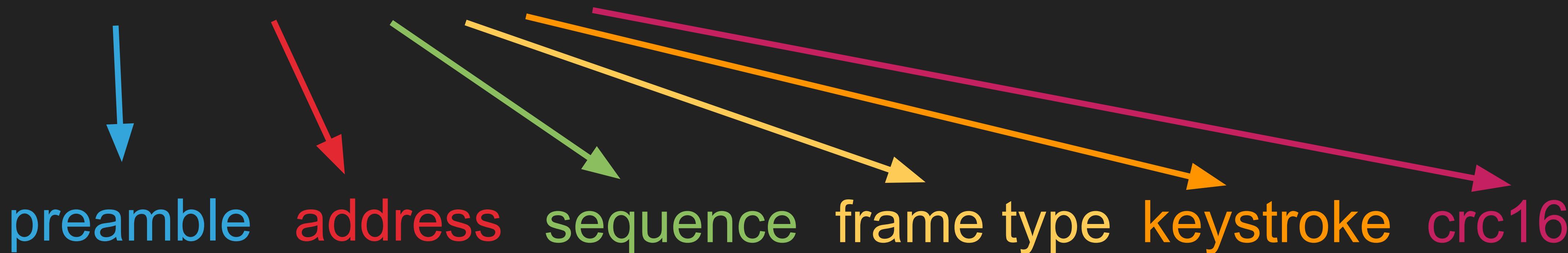
```
aaaaaaa ddd4e8 2e db 3f 384a  
aaaaaaa ddd4e8 2d db 37 6092  
aaaaaaa ddd4e8 28 db 3f 98f8  
aaaaaaa ddd4e8 25 db 3f c9ba  
aaaaaaa ddd4e8 25 db 21 3649  
aaaaaaa ddd4e8 21 db 27 30f5  
aaaaaaa ddd4e8 20 db 3f 3951
```



preamble address sequence frame type keystroke

# GREP, GREP, AND GREP SOME MORE!

```
aaaaaaa ddd4e8 2e db 3f 384a  
aaaaaaa ddd4e8 2d db 37 6092  
aaaaaaa ddd4e8 28 db 3f 98f8  
aaaaaaa ddd4e8 25 db 3f c9ba  
aaaaaaa ddd4e8 25 db 21 3649  
aaaaaaa ddd4e8 21 db 27 30f5  
aaaaaaa ddd4e8 20 db 3f 3951
```



tl;dr  
smarter people than me  
made that easy

Common Threads

---

# Methodology Revisited

# Reverse Engineering Methodology

0. Open-source intelligence research
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

# 1. Channel Characterization

All 3 PHYs share a common notion of a **channel**

Z-Wave	Doorbell	Keyboard
+/- 20 kHz @ 908.42 (plus other channels)	315 MHz	2416 MHz

## 2. Identify Modulation

**Modulation is the biggest variable**  
(but OSINT makes identifying it easy)

Z-Wave	Doorbell	Keyboard
Frequency Shift Keying	Pulse-Width Modulation / On-Off Keying	TDMA Frequency Shift Keying

## 3. Symbol Rate Recovery

All 3 PHYs share a common notion of discrete  
**symbol timing**

Z-Wave	Doorbell	Keyboard
19,200 symbols/s	1000 symbols/s	1,000,000 symbols/s
40,000 symbols/s		
100,000 symbols/s		

## 4. Synchronization

All 3 PHYs contain **synchronization features**  
(preamble and/or Start of Frame delimiter)

Z-Wave	Doorbell	Keyboard
Manchester(0x55..55f0)	Start Bit	Preamble (0xaa..aa) SFD (3 byte address)

## 5. Symbol Extraction

Once you get here it's just **bits on a disk**

Z-Wave	Doorbell	Keyboard

# Reverse Engineering Methodology

0. Open-source intelligence research
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols



Same process for  
3 different PHYs!

# Conclusions

Disparate wireless systems can be rationalized via process

OSINT will help you skip the complex/domain-specific radio parts

Once you demodulate, you have bits on a disk which you can handle any way you  
please

One last thought to leave you with...

# The IoT won't pwn itself

marc@**Bastille**.net  
@marcnewlin

matt@**Bastille**.net  
@embeddedsec

...actually,  
nevermind

marc@**Bastille**.net  
@marcnewlin

matt@**Bastille**.net  
@embeddedsec

# Thanks!

marc@**Bastille**.net  
@marcnewlin

matt@**Bastille**.net  
@embeddedsec

# Questions?

marc@**Bastille**.net  
@marcnewlin

matt@**Bastille**.net  
@embeddedsec

# Preamble and SFD Defines

Preamble = Manchester coded symbols 0110 repeating  
Manchester de-coded: 0b01 repeating

SFD = Manchester coded symbols 1010101001010101  
Manchester de-coded: 0b11110000 == 0xF0