44con 2017

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 2rd in DARPA Spectrum Challenge in 2013
  - Finished 3nd in DARPA Shredder Challenge in 2011

marc@**Bastille**.net
@marcnewlin

- Matt Knight
  - Software Engineer and Security Researcher @ **Bastille**
  - Reverse engineered the LoRa wireless protocol in 2016
  - BE & BA from Dartmouth

matt@**Bastille**.net
@embeddedsec

WIRELESS SYSTEMS ARE EVERYWHERE

MOBILE

WIRELESS SYSTEMS
ARE EVERYWHERE

MOBILE

WIRELESS SYSTEMS
ARE EVERYWHERE

IOT

MOBILE

WIRELESS SYSTEMS ARE EVERYWHERE

IOT

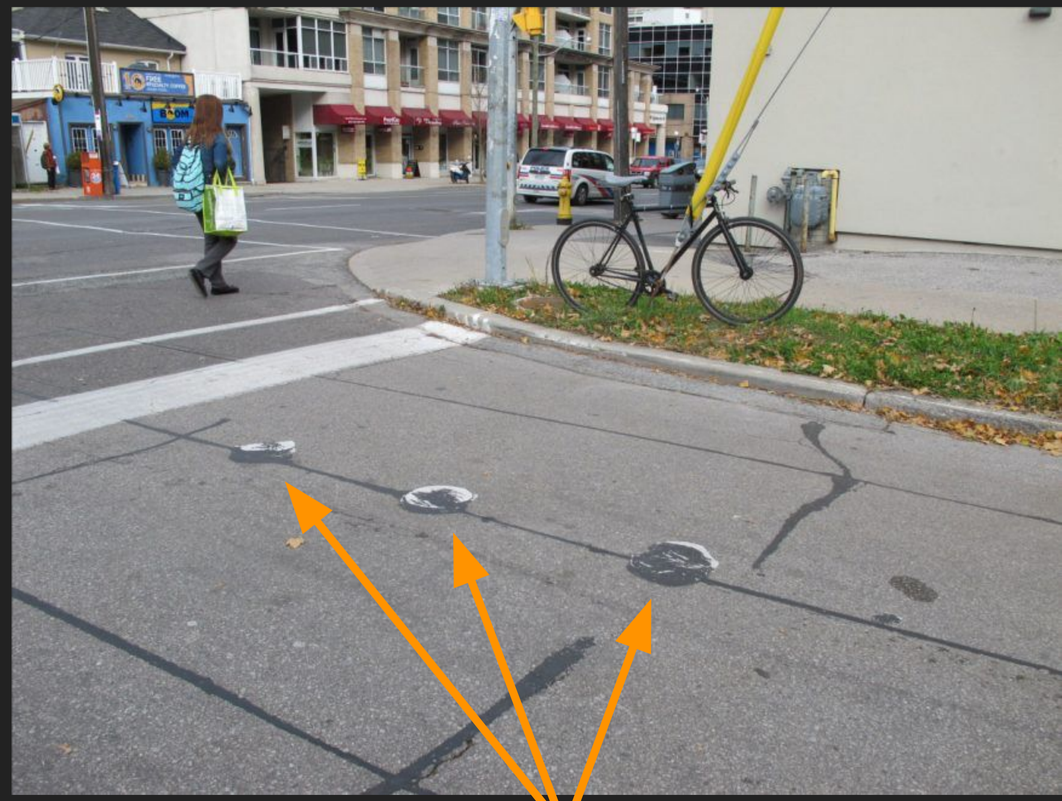
Fewer wires every year!

ABOUT THE INTERNET OF THINGS…

- Everyone's Favorite Four-Letter Word™ **(thanks Don!)**
- What is it, actually?
  - Sales and marketing speak for "connected embedded devices"
  - "Smart" devices are usually pretty stupid

EMBEDDED REALITIES

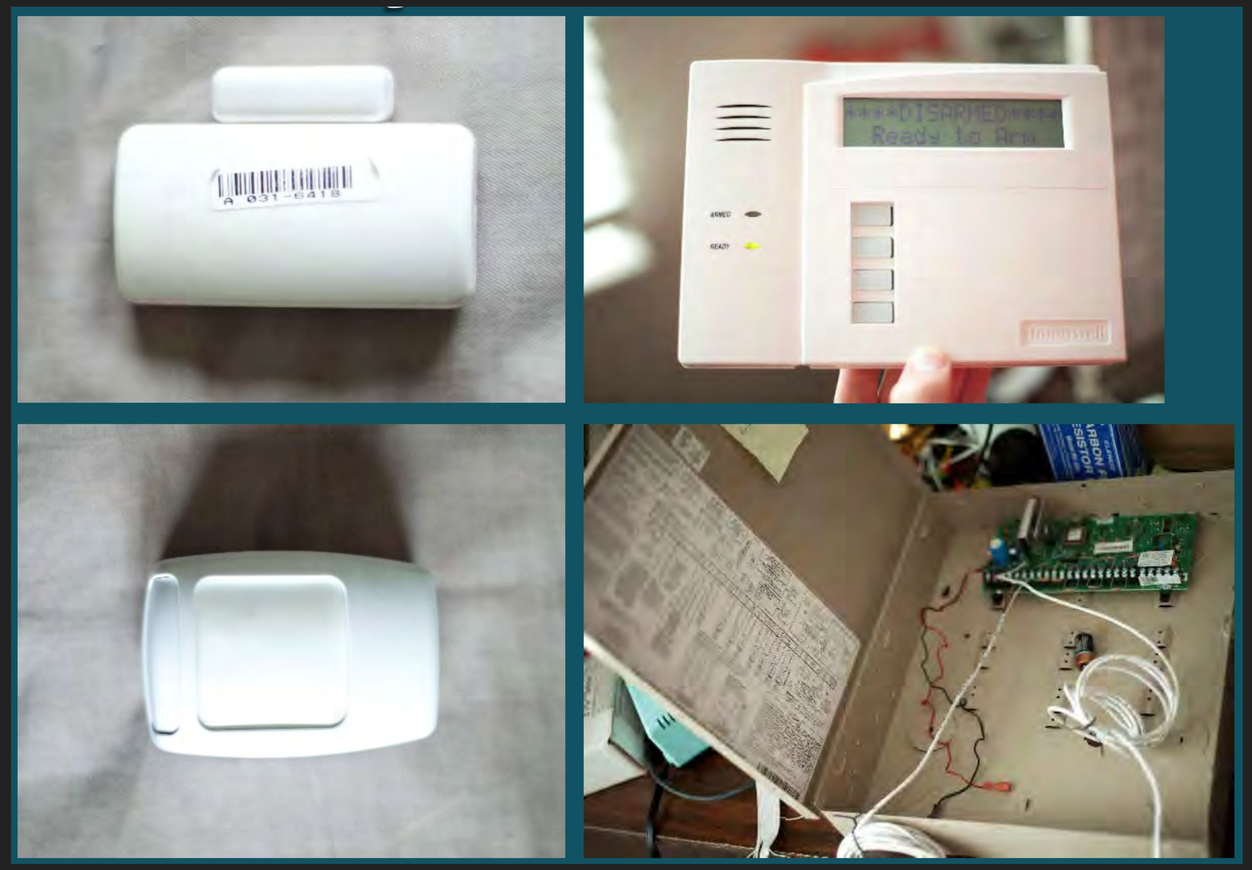
- Embedded systems are built on compromise
  - Size and cost constraints
  - Battery powered
  - Challenging deployment scenarios
  - Difficult to patch



Literal Embedded Systems

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)

Image credit: Logan Lamb

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.)

@krystalmead

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. Wireless home security siren
    2. Wireless barcode scanner
    3. Wireless presentation clicker

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, Universal Radio Hacker: powerful analysis tools
- HDR:
  - Bluez, libubertooth, Killerbee
  - Marc's nRF24 library

TOOLS ARE RIDICULOUS

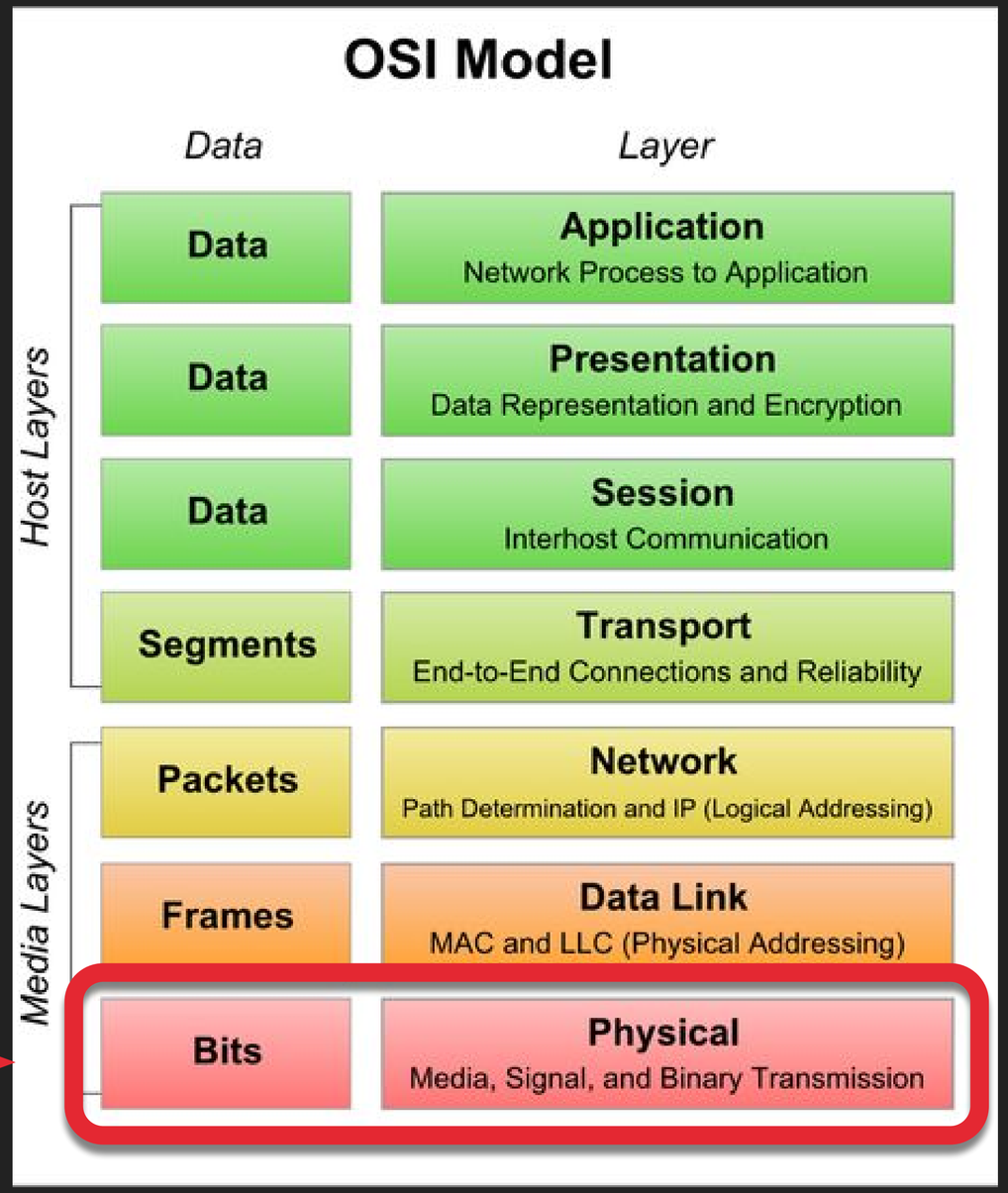
OFFENSIVELY

~~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

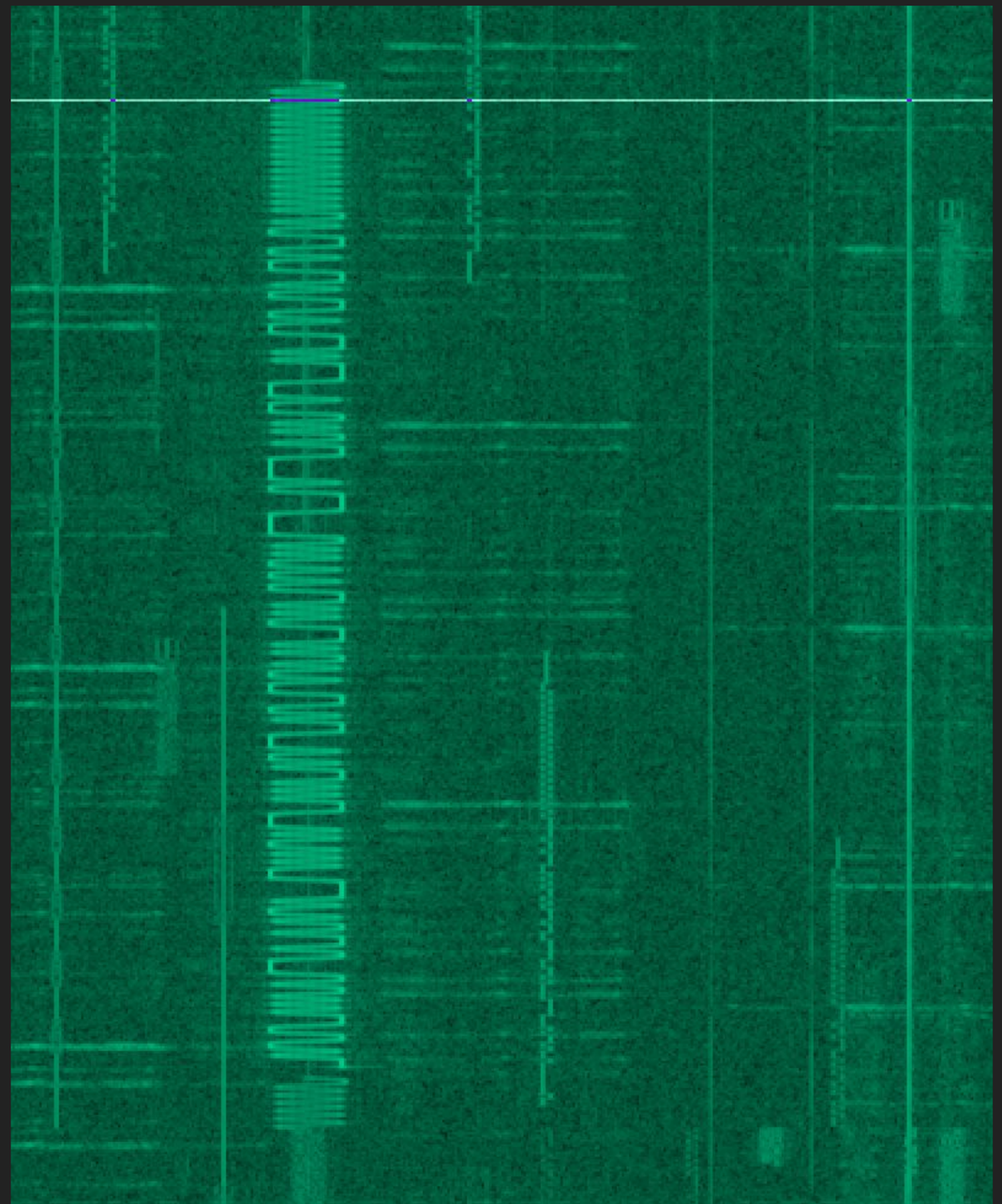


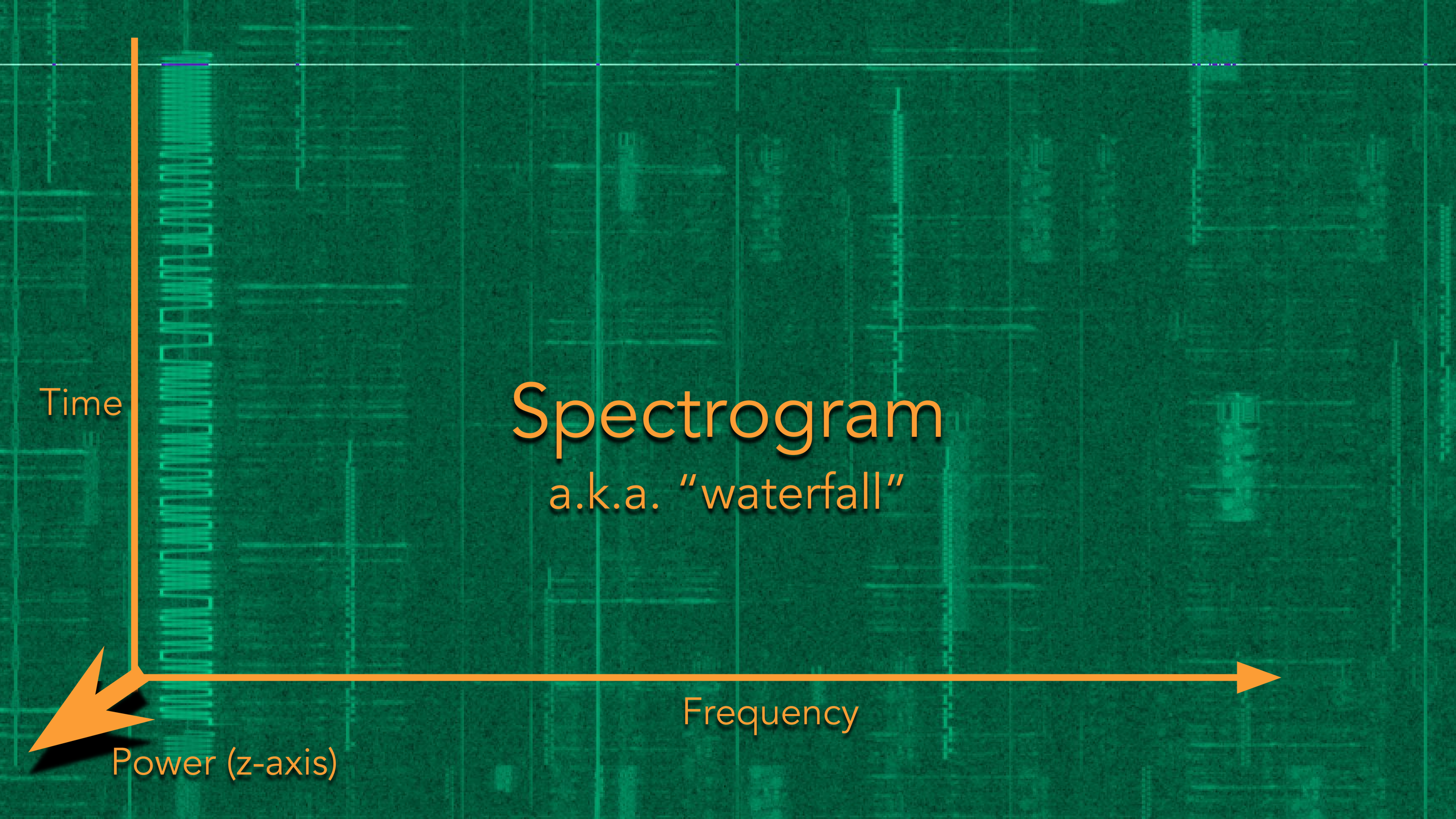
http://www.tech-faq.com/wp-content/uploads/2009/01/osimodel.png

WHAT IS RF?

- "One of the four fundamental forces of the universe" — Tom Rondeau, DARPA Program Manager, former GNU Radio lead

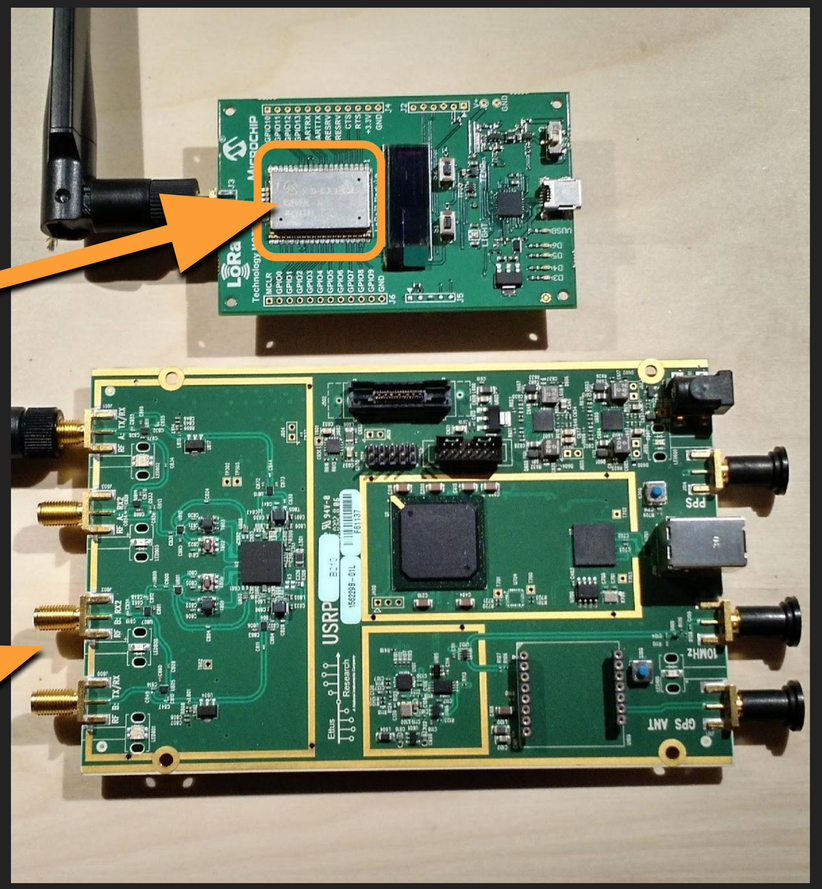
- "Radio Frequency"
- Electromagnetic waves
- Energy




Spectrogram
a.k.a. “waterfall”
Time
Frequency
Power (z-axis)

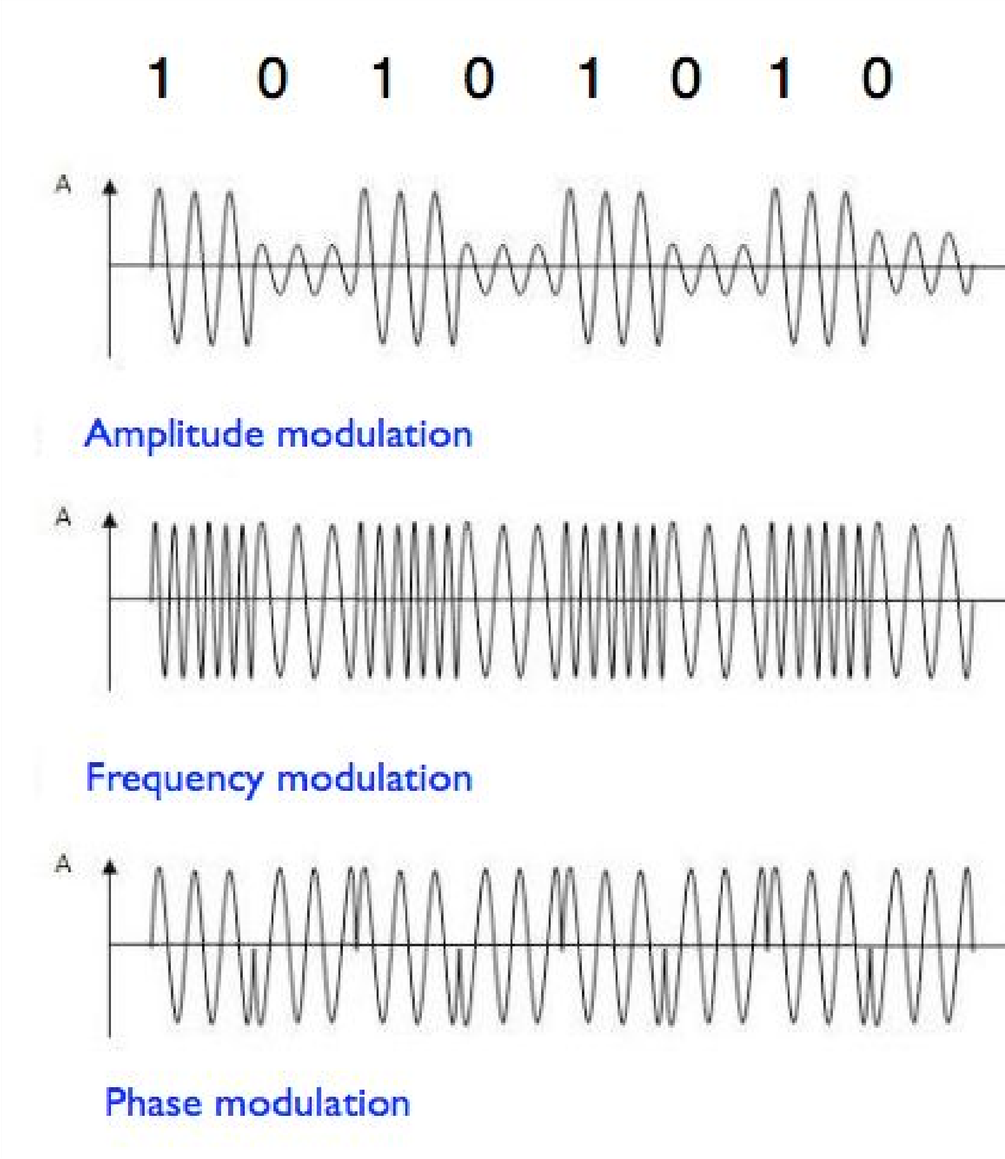
MANIPULATING RF

- 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



http://xenon.colorado.edu/spotlight/kb/gps_basics/modulations.001.png

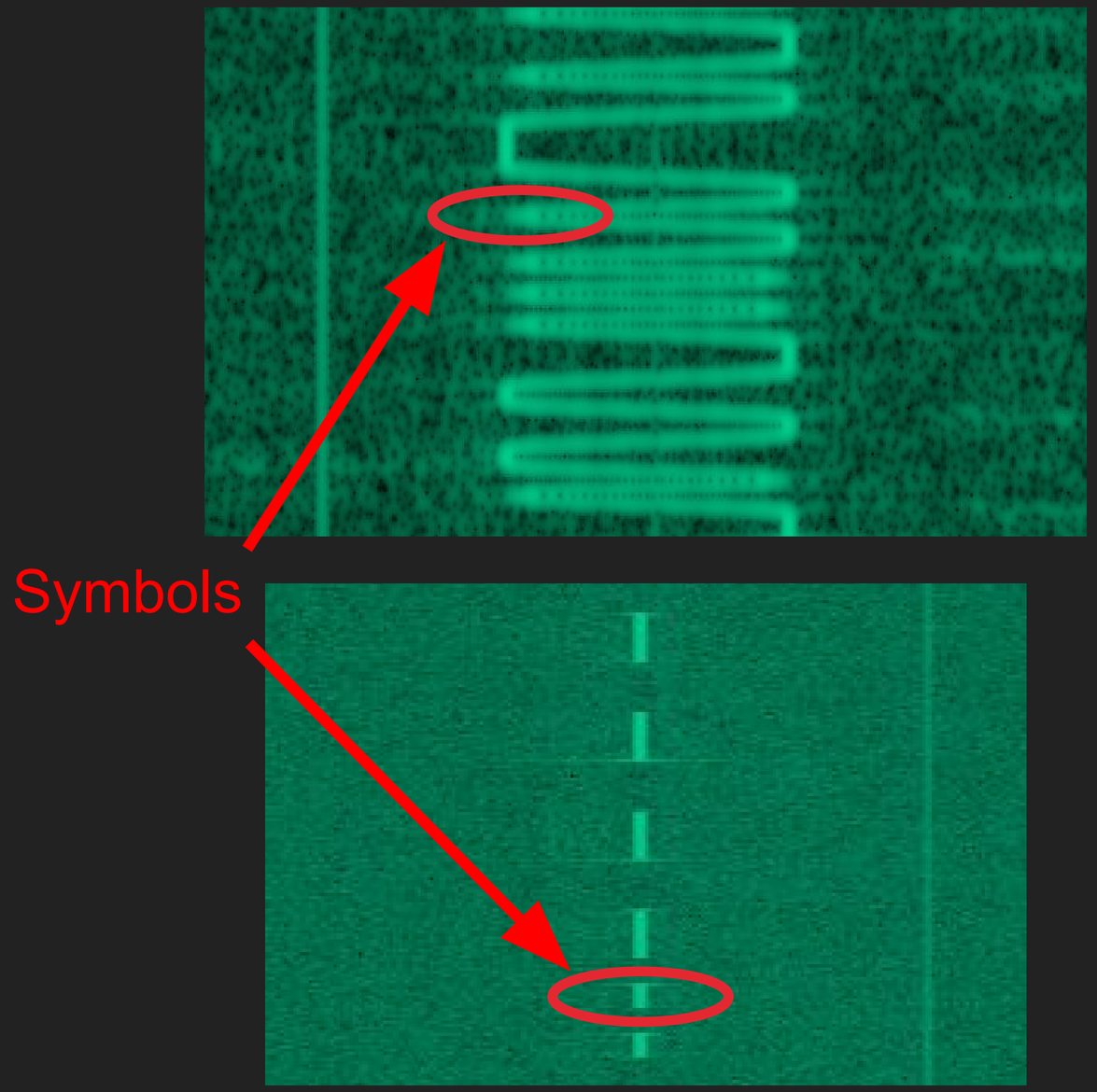
MODULATION

- Modulators can modulate analog or digital information

- Digital modulation
  - Symbols: discrete RF energy state representing some quantity of information

COMMON IOT PHYS

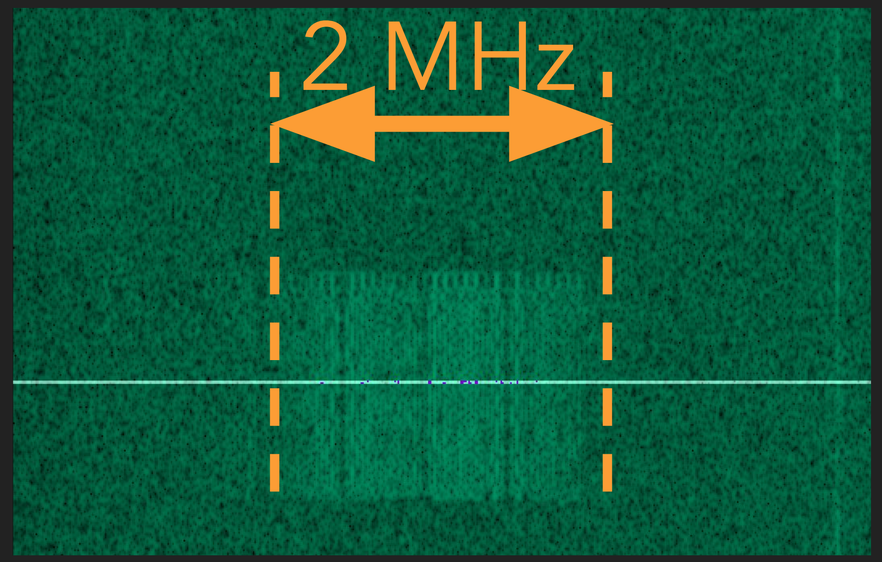
- Frequency Shift Keying: FSK, GFSK
  - RF energy alternates between two frequencies to signify digital values
- Amplitude Shift Keying: ASK, OOK
  - Changes in RF power on a certain frequency signify digital values

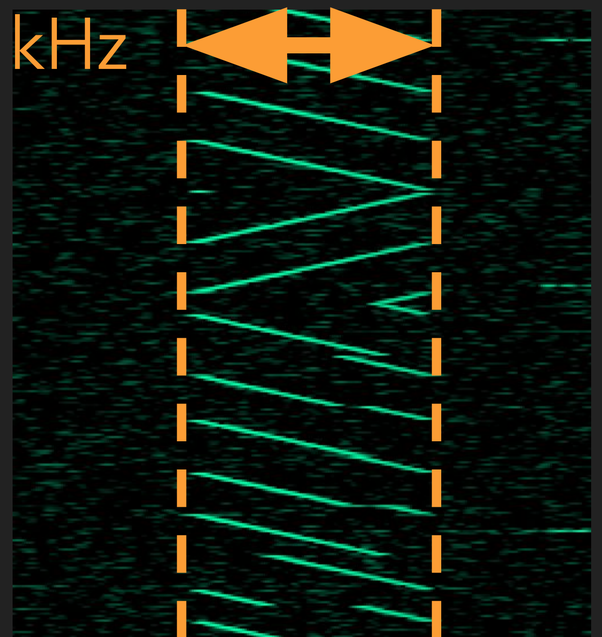


http://3.bp.blogspot.com/-w66qwKucSJI/UgWKKmPUP2I/AAAAAAAADYA/B9NMGYzqJVk/s1600/Screenshot-2013-08-03-04-41-52.png

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)





RADIOS CONTINUED

- Radios can have two functions:
  - Transmitting
  - Receiving

- If a radio can do both it is dubbed a transceiver

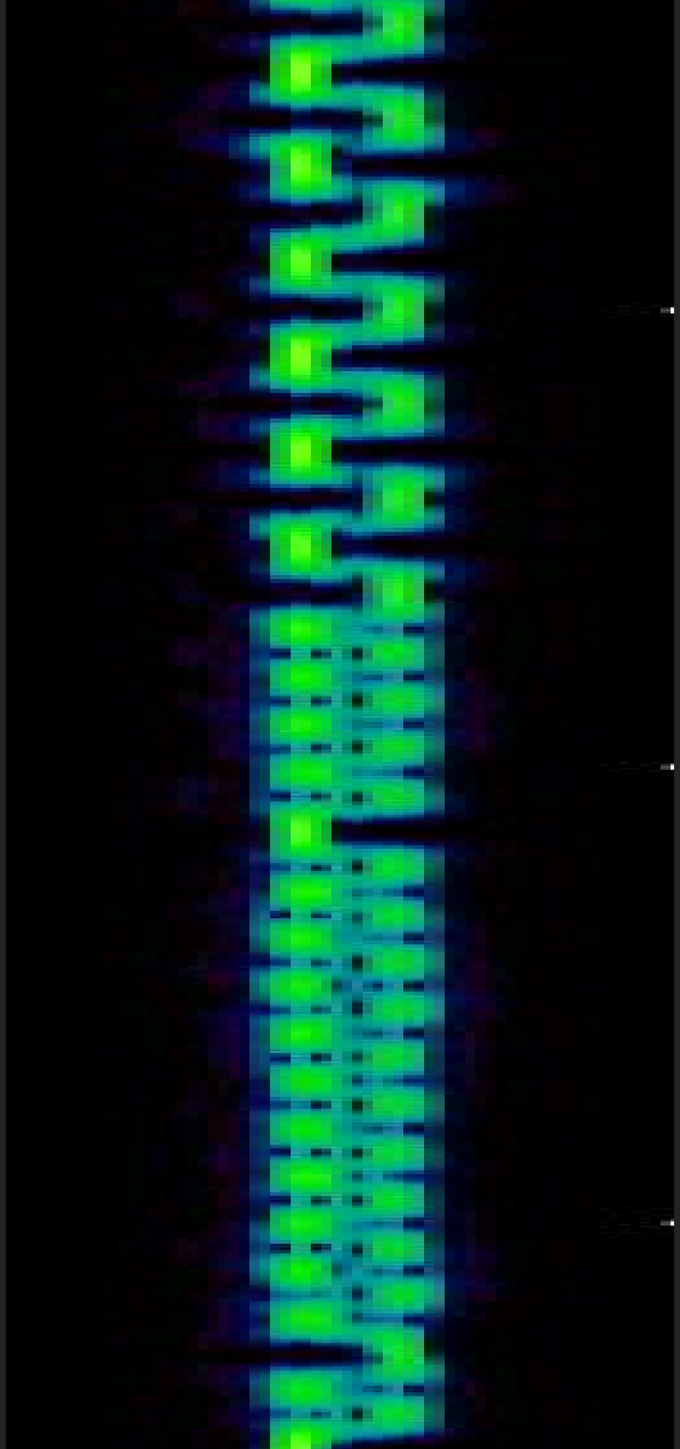
ON REVERSE ENGINEERING

- Main objective: figure out how data is mapped to symbols
- Reverse engineering boils down to building receivers

WIRELESS REVERSE ENGINEERING

METHODOLOGY

[INTERACTIVE]



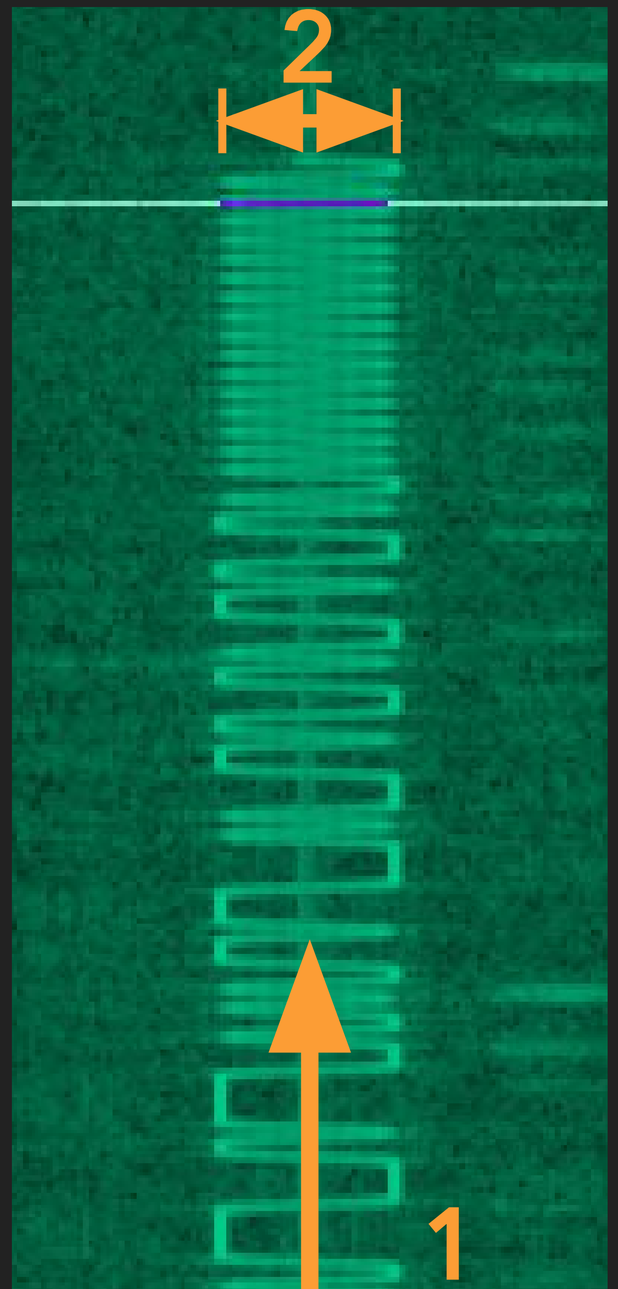
LET’S FORMALIZE THIS

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

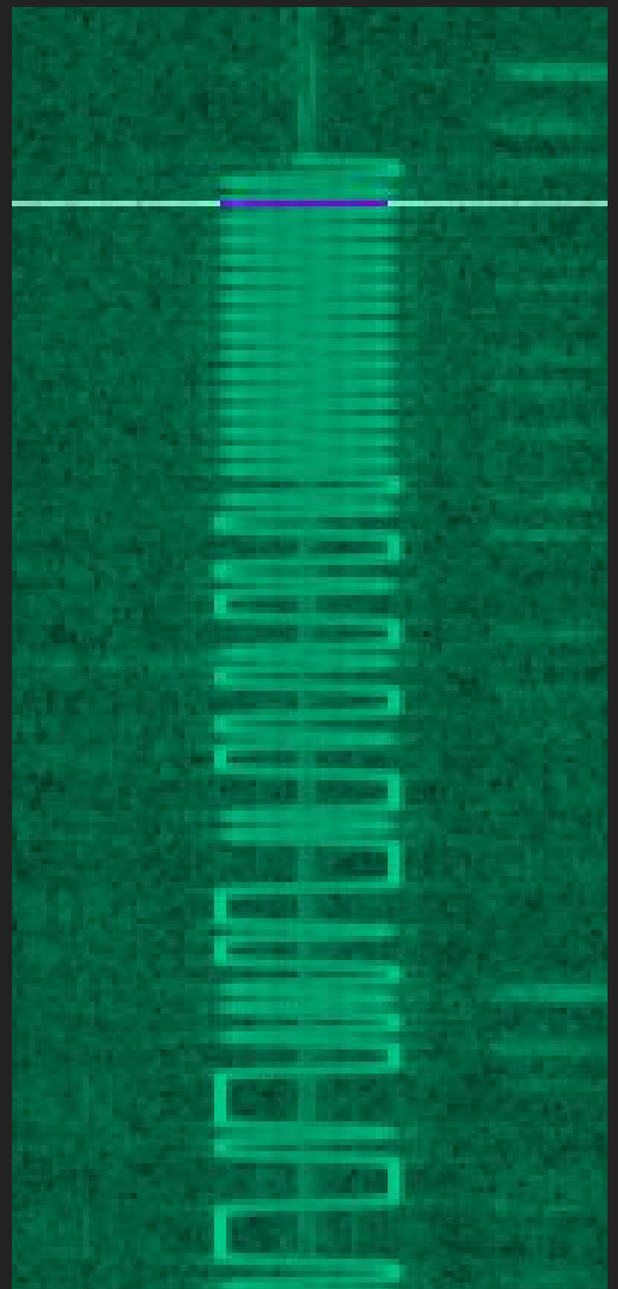
1. Characterize the channel
2. Identify the modulation

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!



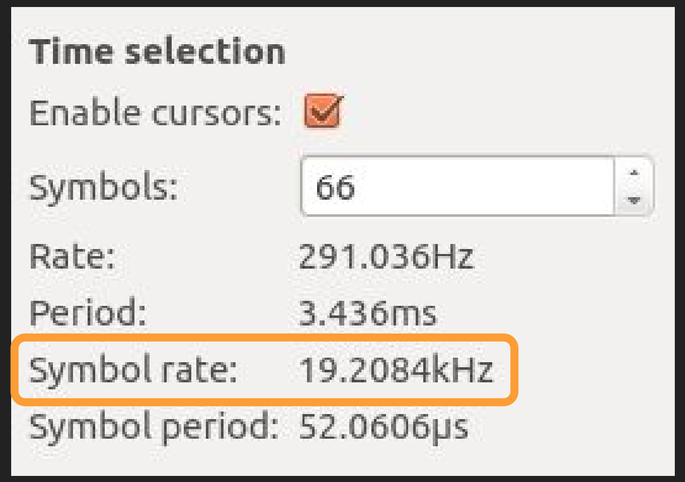
RF REVERSE ENGINEERING METHODOLOGY

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, Inspectrum)

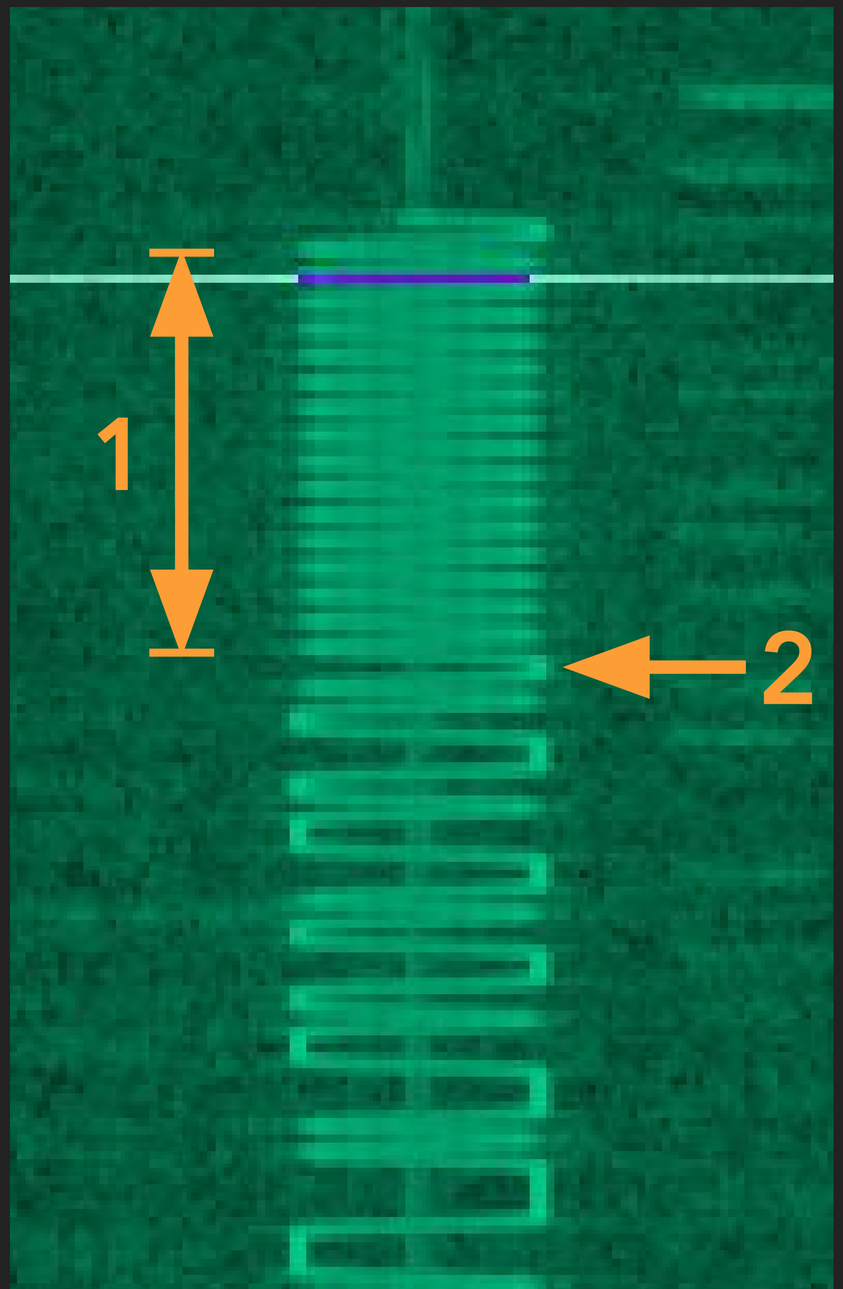


RF REVERSE ENGINEERING METHODOLOGY

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

On-Off Keying

HOME SECURITY SIREN

Fortress Security Store Sentinel Siren

- Combined magnetic door sensor and siren
- Remote control for selecting arming state
- Super loud!

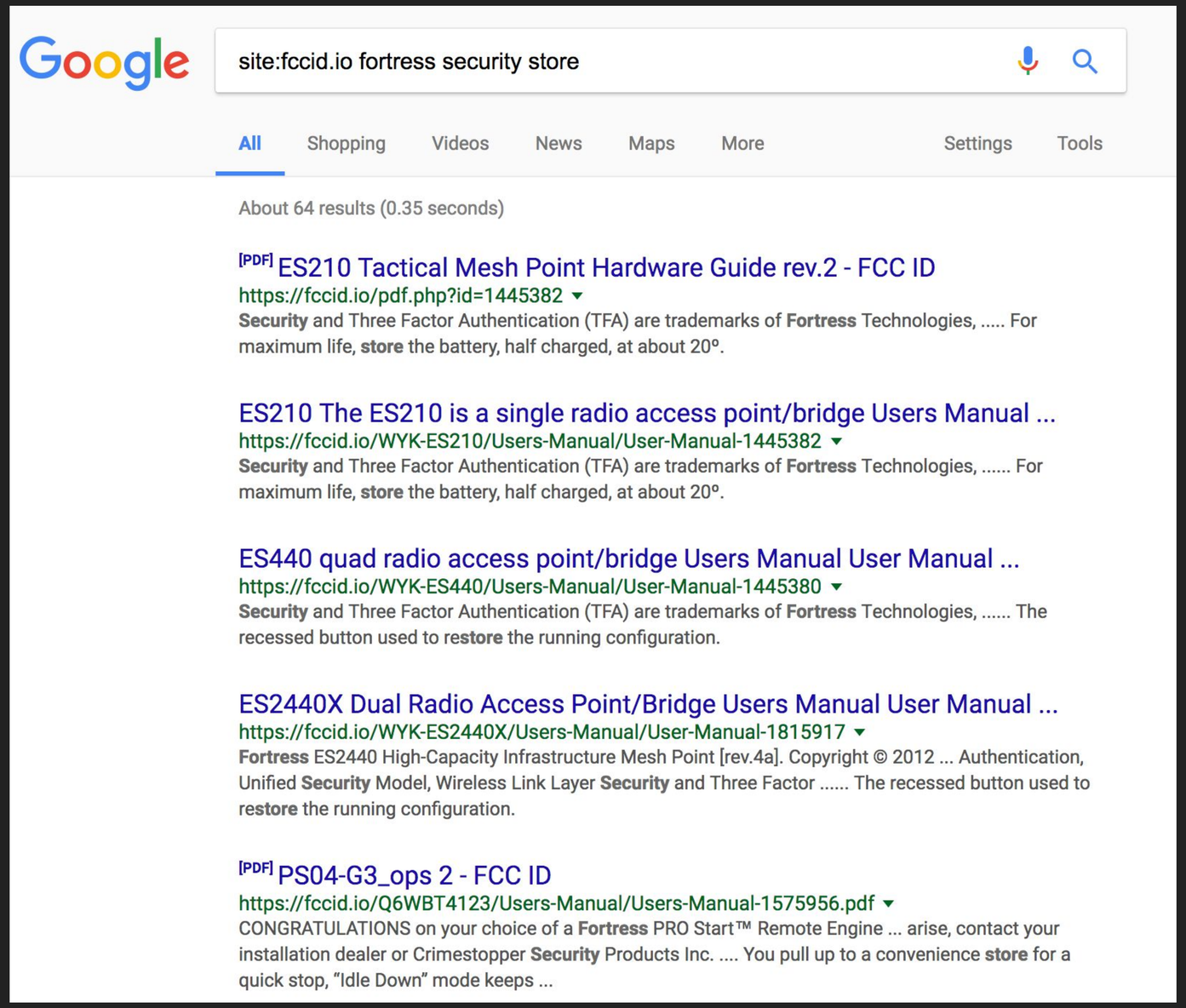
- Let’s reverse it and see how it works!

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

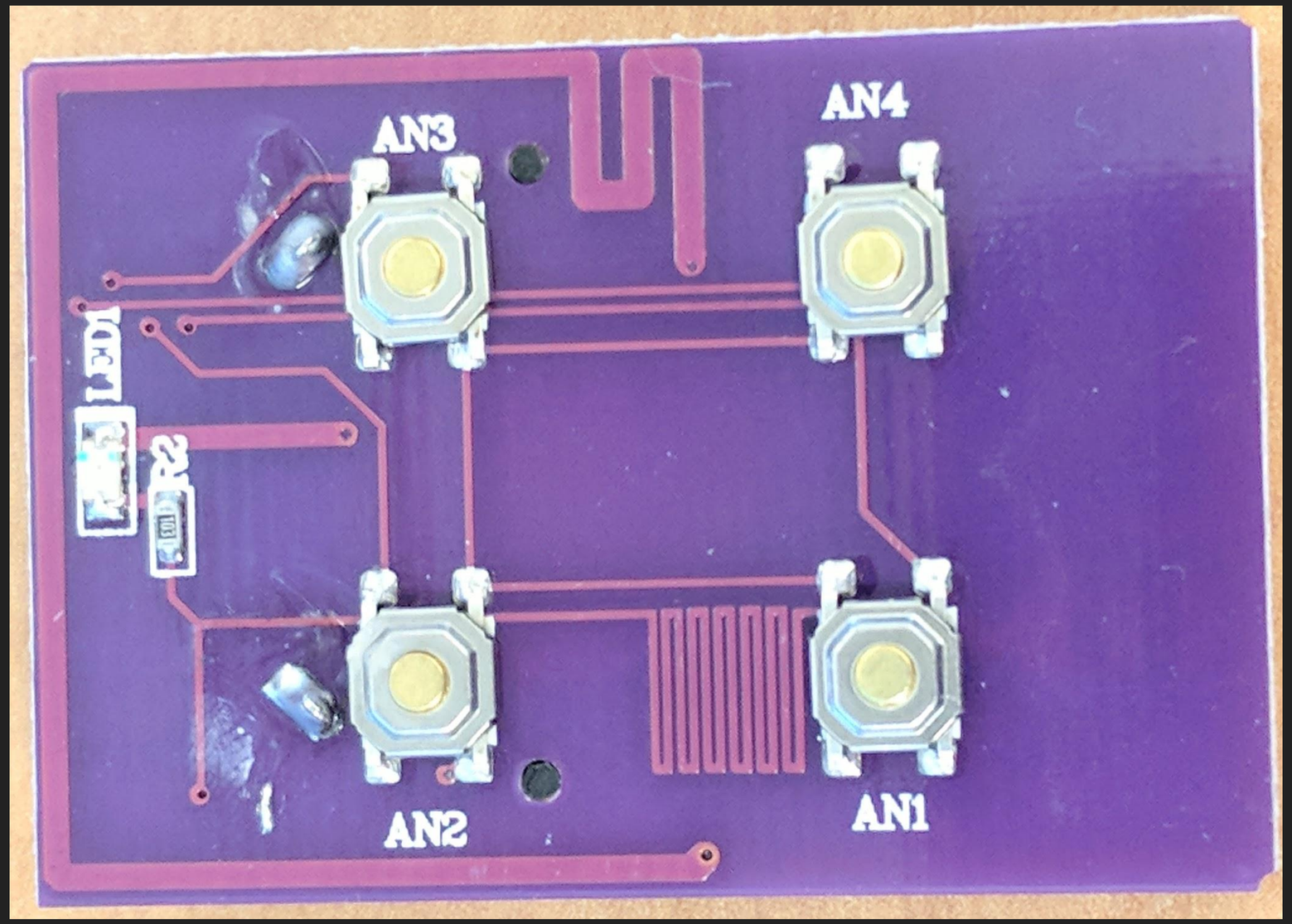
0. OSINT

- No FCC ID silkscreen!
- Google:
  - "site:fccid.io Fortress Security Store"
  - No matches!



0. OSINT

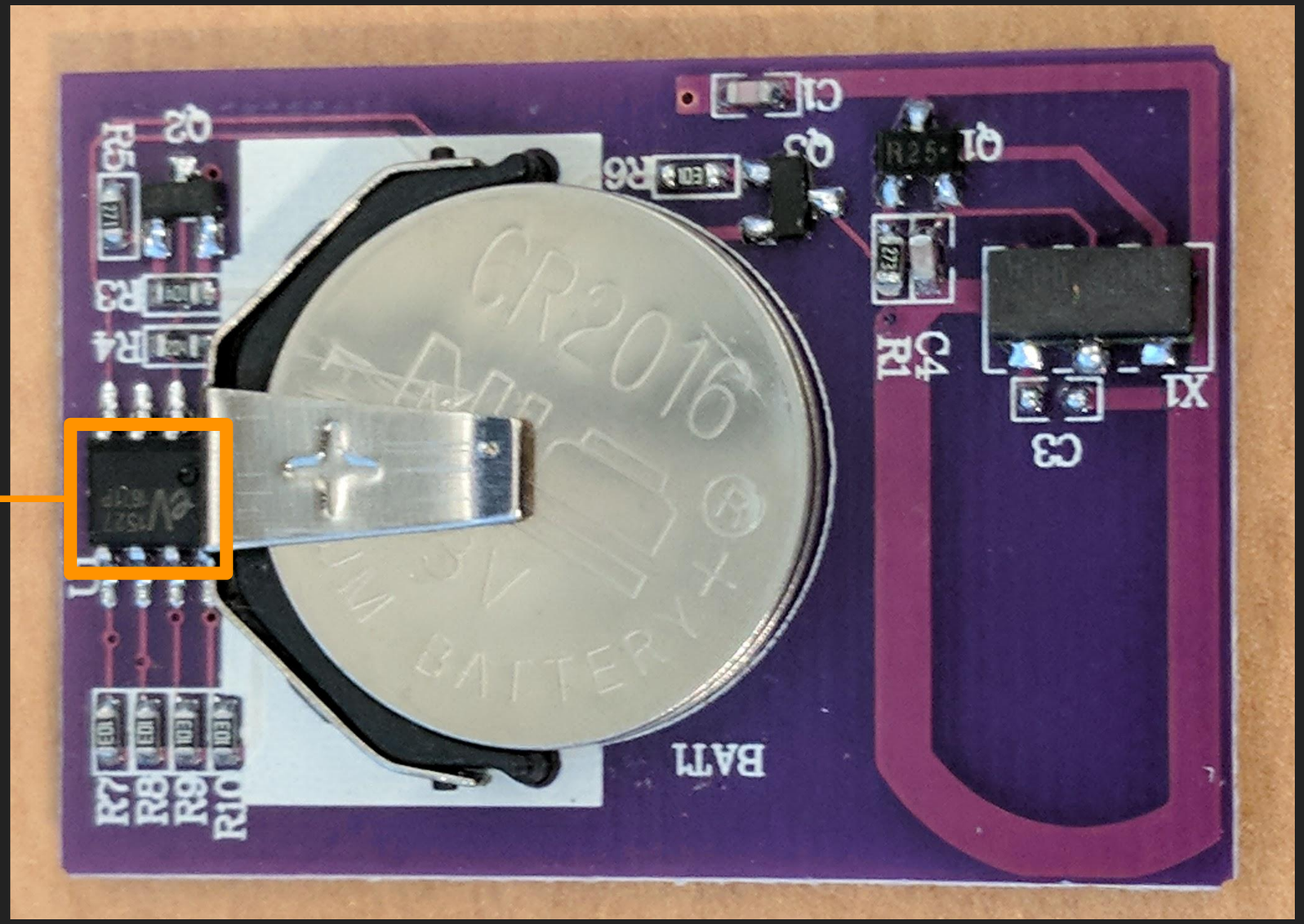
- Crack the case…



0. OSINT

- Crack the case…

eV1527



0. OSINT

- Plug “eV1527” into Google to win

ev1527

All Images Shopping Videos News More Settings Tools

About 333,000 results (0.74 seconds)

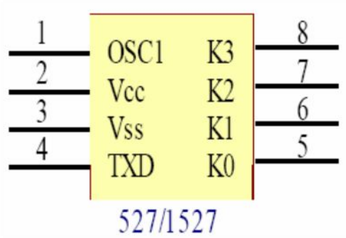
[PDF] SC1527 OTP Encoder - Sunrom
https://www.sunrom.com/get/206000
EV1527 OTP Encoder. **EV1527** Description: (Compatible:**EV1527**,RT1527,FP1527). **EV1527** is OTP Encoder utilizing CMOS technology process. **EV1527**.

EV1527 OTP Encoder

EV1527 Description: (Compatible:EV1527,RT1527,FP1527)
EV1527 is OTP Encoder utilizing CMOS technology process. EV1527 hai a maximum of 20 bits providing up to 1 million codes.It can reduce any code collision and unauthorized code scanning possibilities.

EV1527 Features: (Compatible:EV1527,RT1527,FP1527)
CMOS Technology
Low stand by Current:1.0uA
Wide range of Operating Voltage:Vcc＝3.0～12V
Up to 4 data pins
Total 1048576 address codes
Single Resistor Oscillator
Available in DIP and SOP Package

EV1527 Pin out: (Compatible:EV1527,RT1527,FP1527)



0. OSINT

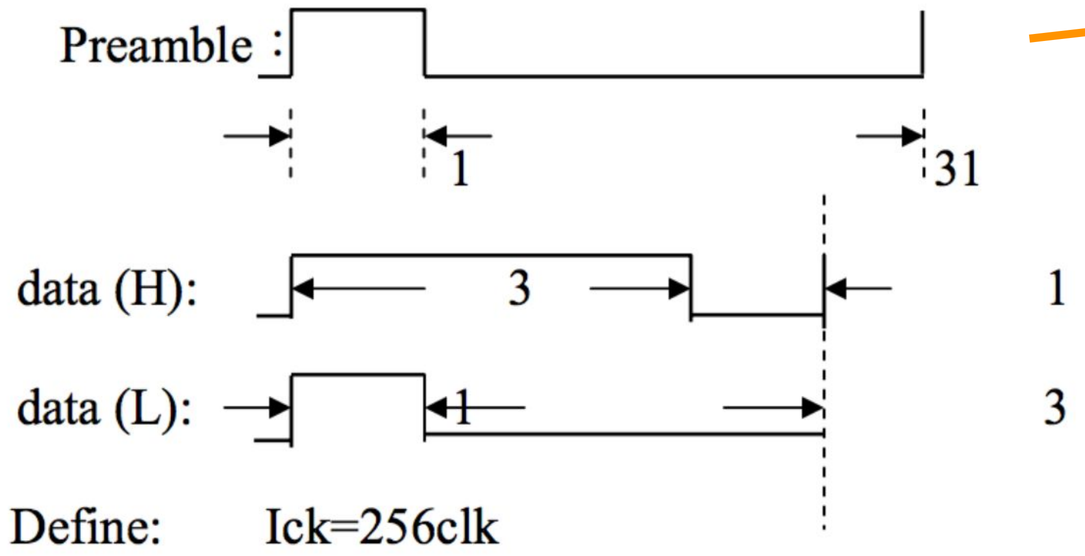
Implies OOK modulation

EV1527 OTP Encoder

EV1527 Output format: (Compatible:EV1527,RT1527,FP1527)

Output data frame:

Preamble	C0 ~ C19 (1 million codes)	D0	D1	D2	D3



Preamble and SFD

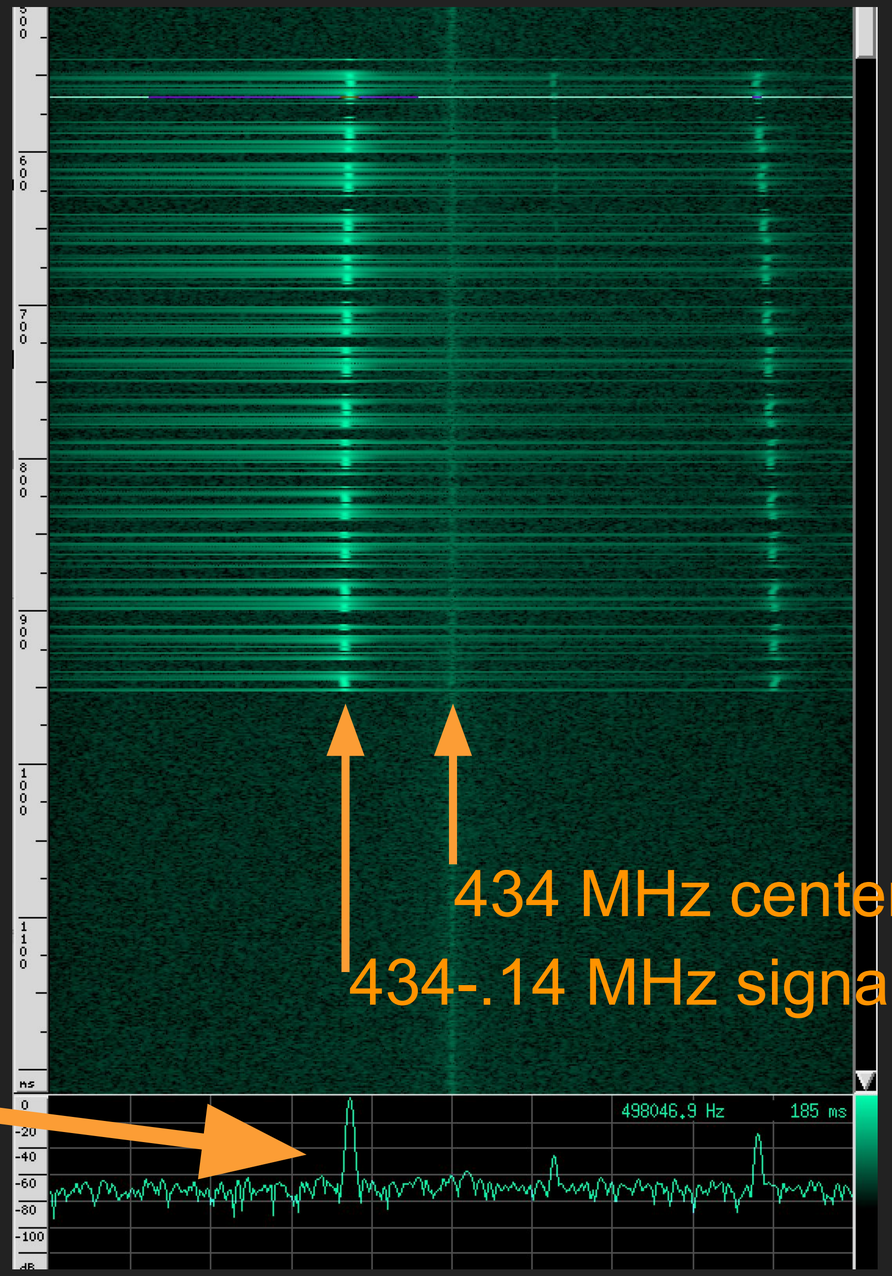
Encoding

RF REVERSE ENGINEERING METHODOLOGY

0. ✓ Open-source intelligence research
1. Characterize the channel
2. Identify the modulation → Likely on-off keying
3. Determine the symbol rate
4. Synchronize → Preamble and SFD
5. Extract symbols → Encoding

1. Channel Characterization

- No frequencies listed in datasheet
- Look at common ISM bands
  - 315 MHz, 345 MHz, 433 MHz, 868 MHz, 915 MHz, 2.4 GHz

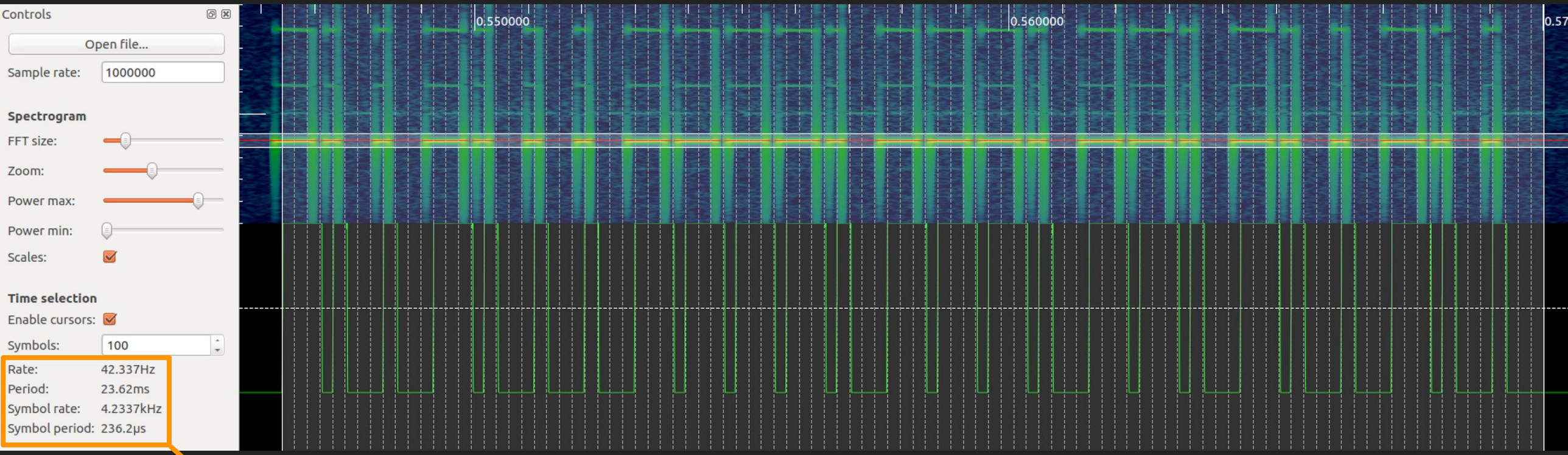


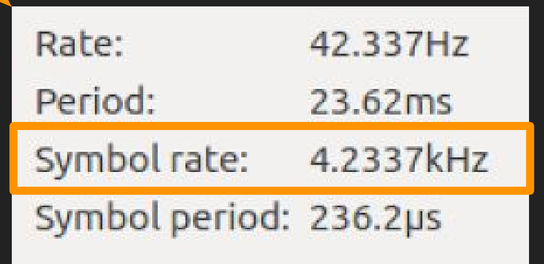
Single carrier confirms OOK

RF REVERSE ENGINEERING METHODOLOGY

0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → 433.86 MHz
2. ✓ Identify the modulation → On-off keying
3. Determine the symbol rate
4. Synchronize → Preamble and SFD
5. Extract symbols → Encoding

3. Determine Symbol Rate





Inspectrum w/ cursors

RF REVERSE ENGINEERING METHODOLOGY

0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → 433.86 MHz
2. ✓ Identify the modulation → On-off keying
3. ✓ Determine the symbol rate → ~4.23 kHz
4. Synchronize → Preamble and SFD
5. Extract symbols → Encoding

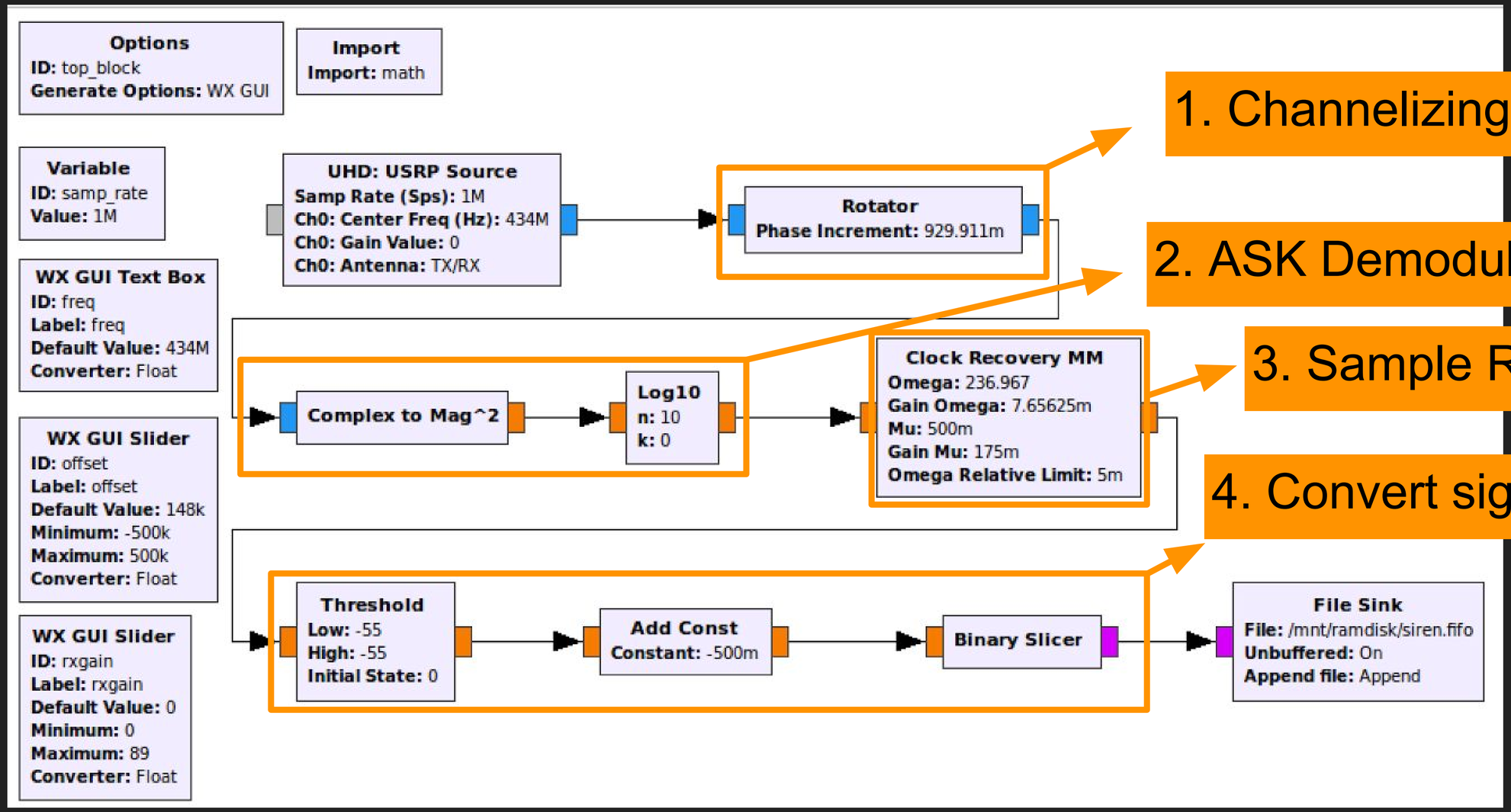
RF REVERSE ENGINEERING METHODOLOGY

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

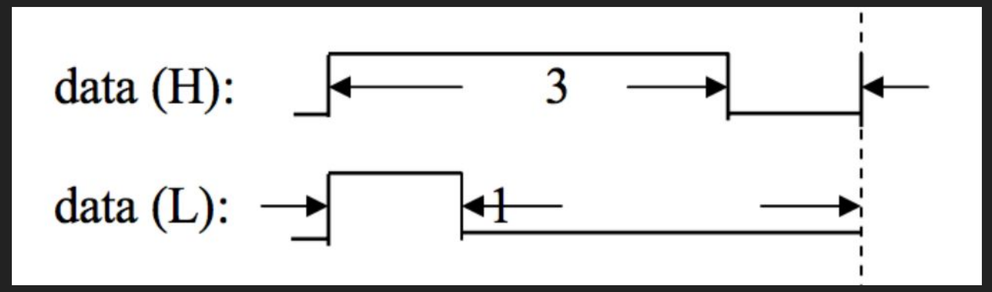
Translate Parameters into a GNU Radio Flowgraph



1. Channelizing
2. ASK Demodulation
3. Sample Rate to Symbol Rate
4. Convert signal power to binary

4. Synchronization and 5. Symbol Extraction

1. Look for preamble/SFD (they're the same for this device)
   a. 0b10000000000000000000000000000000 (1 0b1, 31 0b0)
2. Read out frame using empirically observed MTU size (100 symbols)
3. Decode encoded symbols into bits
   a. Symbols 0b1110 → logical 1
   b. Symbols 0b1000 → logical 0
4. Parse frame



Application Frame Structure

Address

- 20 bit long unique address

State

- 4 bits denoting the button pressed / desired state

Demo Time!

Wireless Barcode Scanner

WoneNice WN6200

WoneNice WN6200

- 433MHz Wireless Barcode Scanner
- Scanner
  - No FCC ID
- Dongle
  - No FCC ID

WoneNice WN6200 Reverse Engineering Workflow

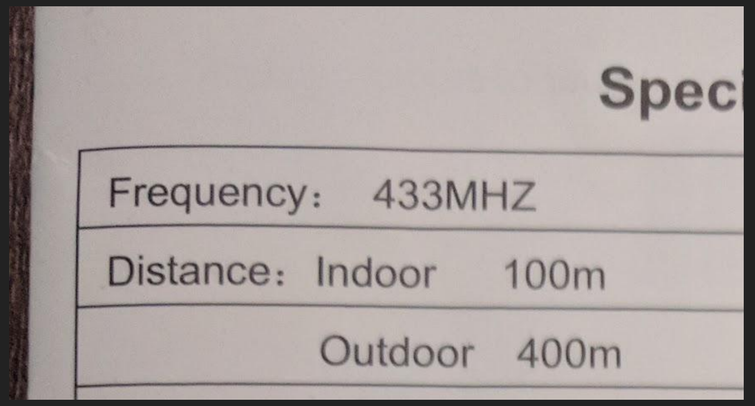
0. Open-source intelligence research
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

WoneNice WN6200 User Manual

Specification

Frequency: 433MHZ			Charging supply: 5V-500mA
Distance: Indoor 100m			Working current: 15-40mA
Outdoor 400m			Standby current : 18uA-5mA-
storage: 100000 barcode			Working time: 7 to 10 days
Scanner parameters:			
Scanning type : bi-directional			Light source: 650nm LD
CPU: ARM 32-bit Cortex			Trigger Mode: Button,Continuous
Printing Contract : >25%			Indication: Buzzer & LED
Depth of Field	3.3mil	2mm-100m	Resolution : 3.3mil
	10mil	2mm-350m	Scanning width : 30cm
	15.6mil	5mm-600m	Decoding speed: 260scans/sec
	35mil	10mm-1000	Bit error rate :1/20million
Scanning angle: rotor angel±30°, inclination± 45°, declination ± 60°			
Anti inference: industrial lighting or sun will not make any difference			
Decoding Capability: UPC/EAN、with complementary UPC/EAN、			
Code39、Code 39Full ASCII、Coda bar、industrial /Interleaved 2 of			
Code93、MSI、Code11、ISBN、ISSN、China post、etc			
(optional)GS1 Databar,Code32			
Button life : 8000,000 times			laser life: 12000 hours
Drop test : 2.0m fall to concrete			
Certificate : CE, FCC, RoHS, IP54			

2



Spec

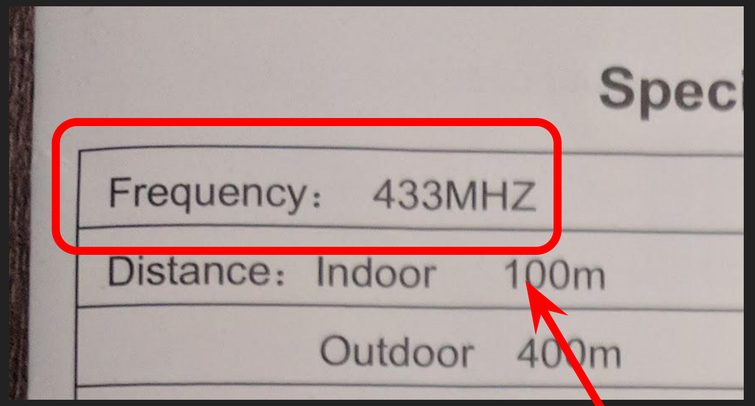
Frequency:	433MHZ
Distance: Indoor	100m
Outdoor	400m

WoneNice WN6200 User Manual

Specification

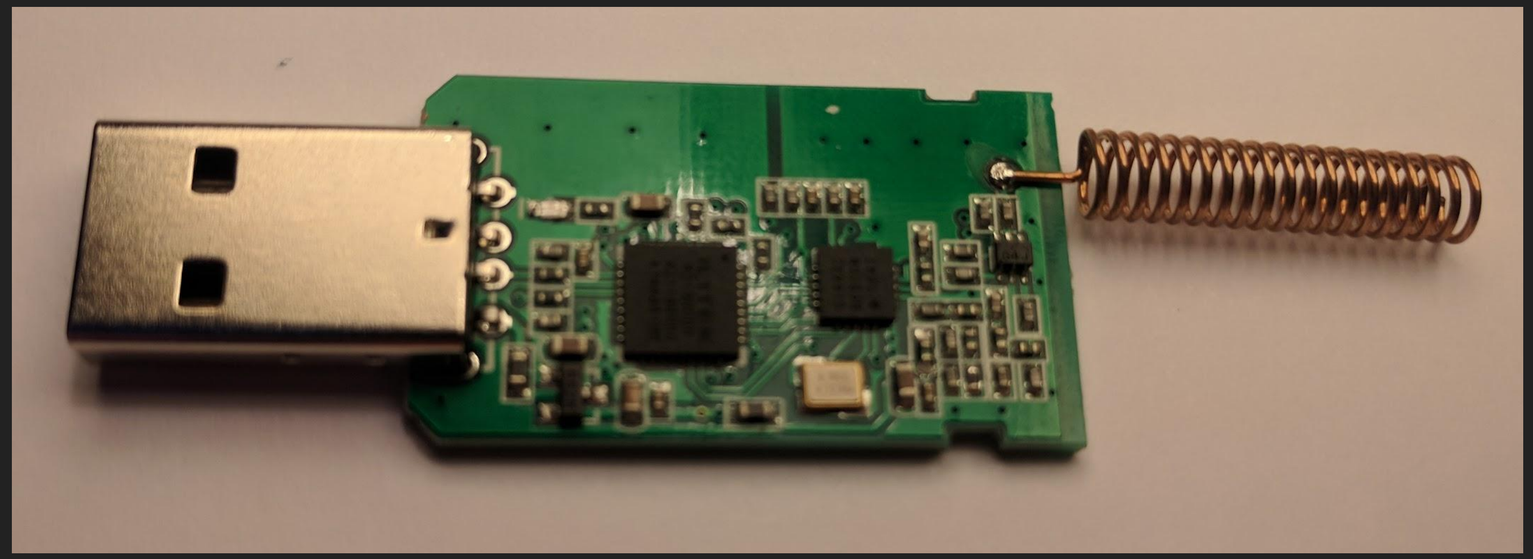
Frequency: 433MHZ			Charging supply: 5V-500mA
Distance: Indoor 100m			Working current: 15-40mA
Outdoor 400m			Standby current : 18uA-5mA-
storage: 100000 barcode			Working time: 7 to 10 days
Scanner parameters:			
Scanning type : bi-directional			Light source: 650nm LD
CPU: ARM 32-bit Cortex			Trigger Mode: Button,Continuous
Printing Contract : >25%			Indication: Buzzer & LED
Depth of Field	3.3mil	2mm-100m	Resolution : 3.3mil
	10mil	2mm-350m	Scanning width : 30cm
	15.6mil	5mm-600m	Decoding speed: 260scans/sec
	35mil	10mm-1000	Bit error rate :1/20million
Scanning angle: rotor angel±30°, inclination± 45°, declination ± 60°			
Anti inference: industrial lighting or sun will not make any difference			
Decoding Capability: UPC/EAN、with complementary UPC/EAN、			
Code39、Code 39Full ASCII、Coda bar、industrial /Interleaved 2 of			
Code93、MSI、Code11、ISBN、ISSN、China post、etc			
(optional)GS1 Databar,Code32			
Button life : 8000,000 times			laser life: 12000 hours
Drop test : 2.0m fall to concrete			
Certificate : CE, FCC, RoHS, IP54			

2

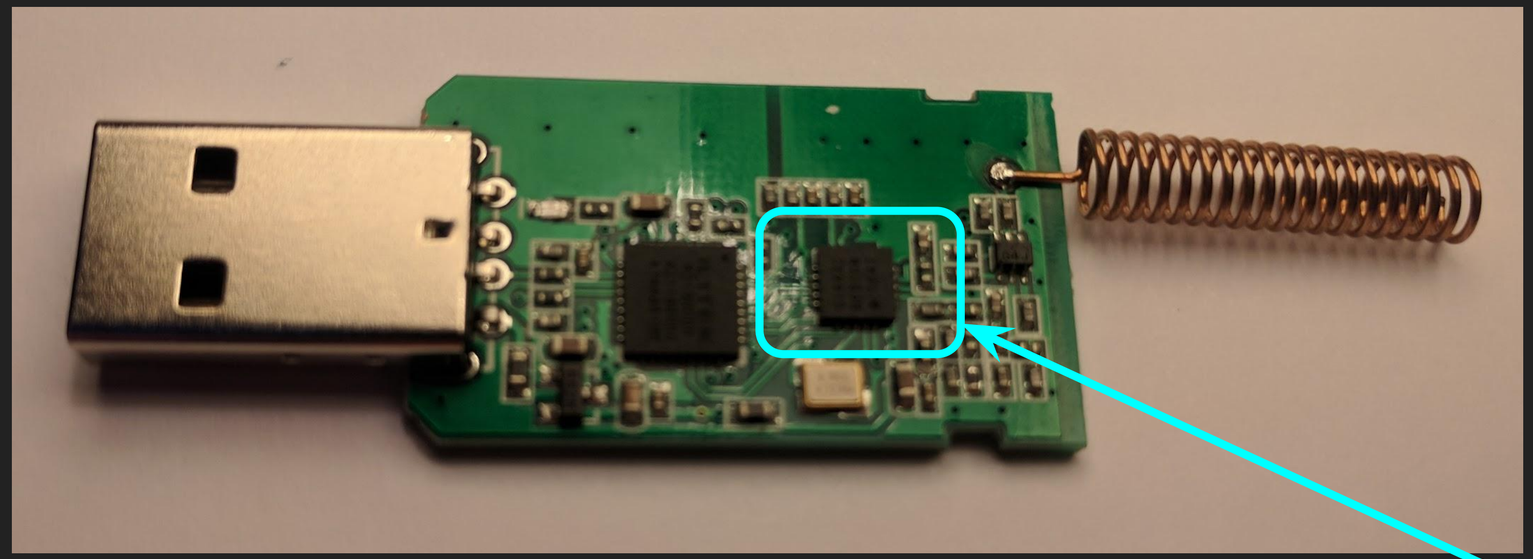


433MHz Center Frequency

Barcode Scanner Dongle



Barcode Scanner Dongle



Silicon Labs Si4432

Silicon Labs Si4432

SILICON LABS

Si4430/31/32-B1

Si4430/31/32 ISM TRANSCEIVER

Features

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

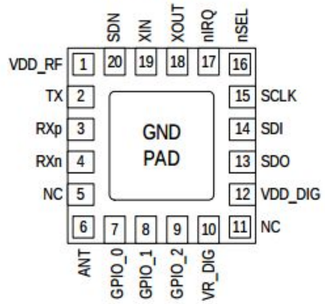
Ordering Information:
See page 67.

Applications

- Remote control
- Home security & alarm
- Telemetry
- Personal data logging
- Toy control
- Tire pressure monitoring
- Wireless PC peripherals
- Remote meter reading
- Remote keyless entry
- Home automation
- Industrial control
- Sensor networks
- Health monitors
- Tag readers

Pin Assignments

Si4430/31/32



Description

Silicon Laboratories' Si4430/31/32 devices are highly integrated, single chip

Silicon Labs Si4432

SILICON LABS

Si4430/31/32-B1

Si4430/31/32 ISM TRANSCEIVER

Features

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

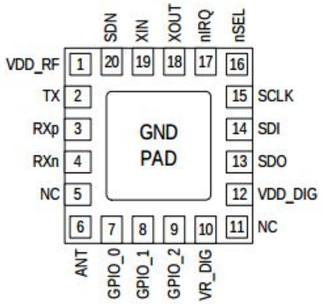
Ordering Information:
See page 67.

Applications

- Remote control
- Home security & alarm
- Telemetry
- Personal data logging
- Toy control
- Tire pressure monitoring
- Wireless PC peripherals
- Remote meter reading
- Remote keyless entry
- Home automation
- Industrial control
- Sensor networks
- Health monitors
- Tag readers

Pin Assignments

Si4430/31/32



Description

Silicon Laboratories' Si4430/31/32 devices are highly integrated, single chip

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

Silicon Labs Si4432

SILICON LABS

Si4430/31/32-B1

Si4430/31/32 ISM TRANSCEIVER

Features

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

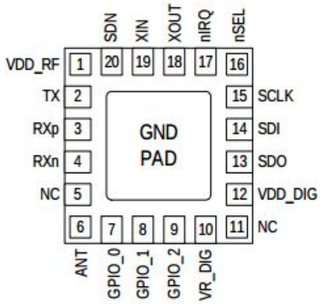
Ordering Information:
See page 67.

Applications

- Remote control
- Home security & alarm
- Telemetry
- Personal data logging
- Toy control
- Tire pressure monitoring
- Wireless PC peripherals
- Remote meter reading
- Remote keyless entry
- Home automation
- Industrial control
- Sensor networks
- Health monitors
- Tag readers

Pin Assignments

Si4430/31/32



Description

Silicon Laboratories' Si4430/31/32 devices are highly integrated, single chip

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

Silicon Labs Si4432

SILICON LABS

Si4430/31/32-B1

Si4430/31/32 ISM TRANSCEIVER

Features

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

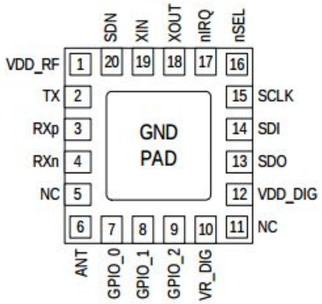
Ordering Information:
See page 67.

Applications

- Remote control
- Home security & alarm
- Telemetry
- Personal data logging
- Toy control
- Tire pressure monitoring
- Wireless PC peripherals
- Remote meter reading
- Remote keyless entry
- Home automation
- Industrial control
- Sensor networks
- Health monitors
- Tag readers

Pin Assignments

Si4430/31/32



Description

Silicon Laboratories' Si4430/31/32 devices are highly integrated, single chip

- Frequency Range
  - 240–930 MHz (Si4431/32)
  - 900–960 MHz (Si4430)
- Sensitivity = –121 dBm
- Output power range
  - +20 dBm Max (Si4432)
  - +13 dBm Max (Si4430/31)
- Low Power Consumption
  - 18.5 mA receive
  - 30 mA @ +13 dBm transmit
  - 85 mA @ +20 dBm transmit
- Data Rate = 0.123 to 256 kbps
- FSK, GFSK, and OOK modulation
- Power Supply = 1.8 to 3.6 V
- Ultra low power shutdown mode
- Digital RSSI
- Wake-up timer
- Auto-frequency calibration (AFC)
- Power-on-reset (POR)
- Antenna diversity and TR switch control
- Configurable packet handler
- Preamble detector
- TX and RX 64 byte FIFOs
- Low battery detector
- Temperature sensor and 8-bit ADC
- –40 to +85 °C temperature range
- Integrated voltage regulators
- Frequency hopping capability
- On-chip crystal tuning
- 20-Pin QFN package
- Low BOM

Silicon Labs Si4432

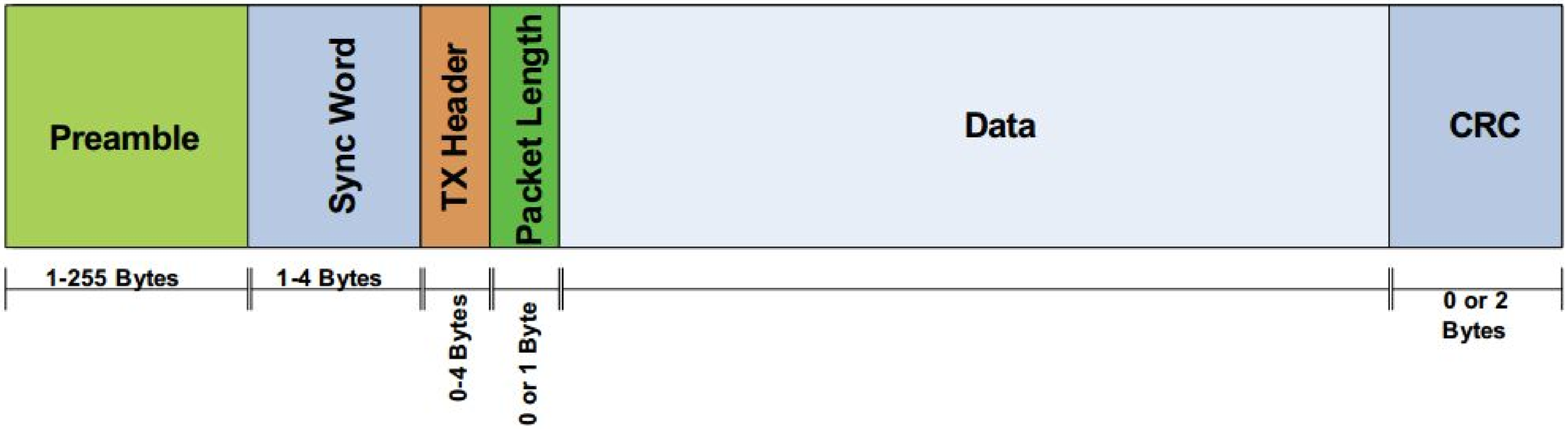


Figure 18. Packet Structure

Silicon Labs Si4432

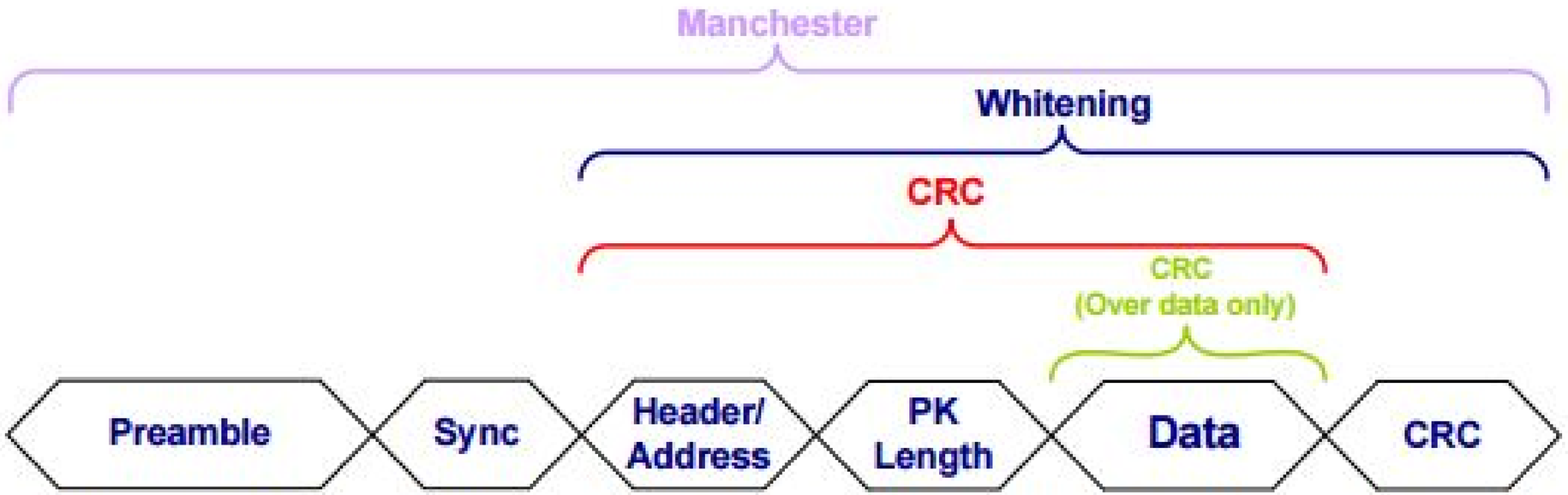
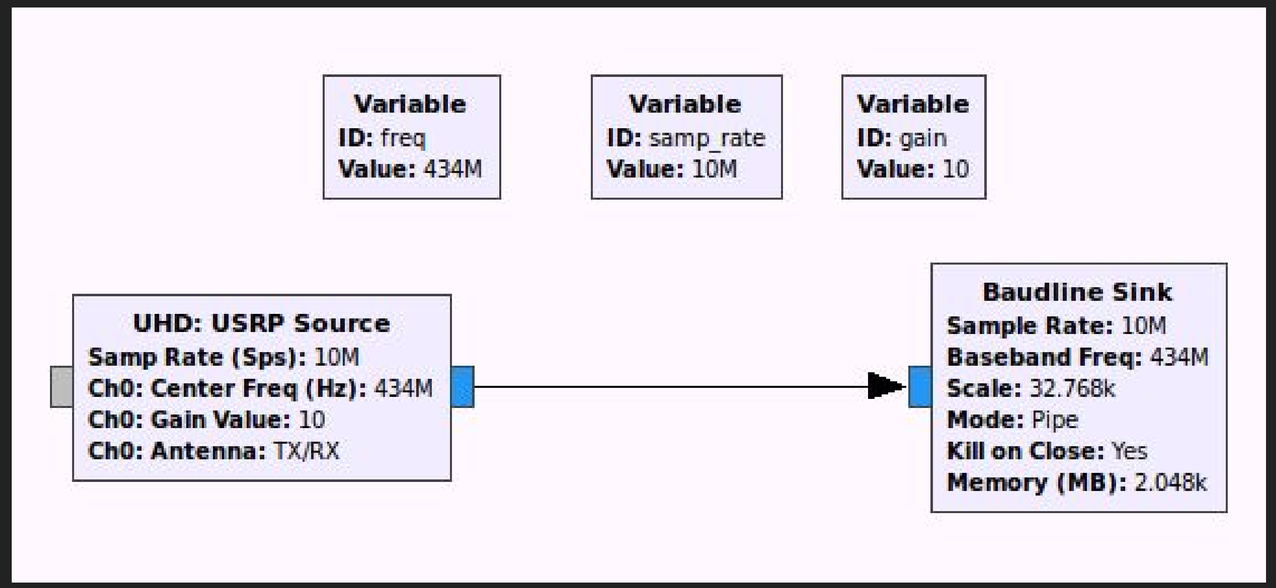


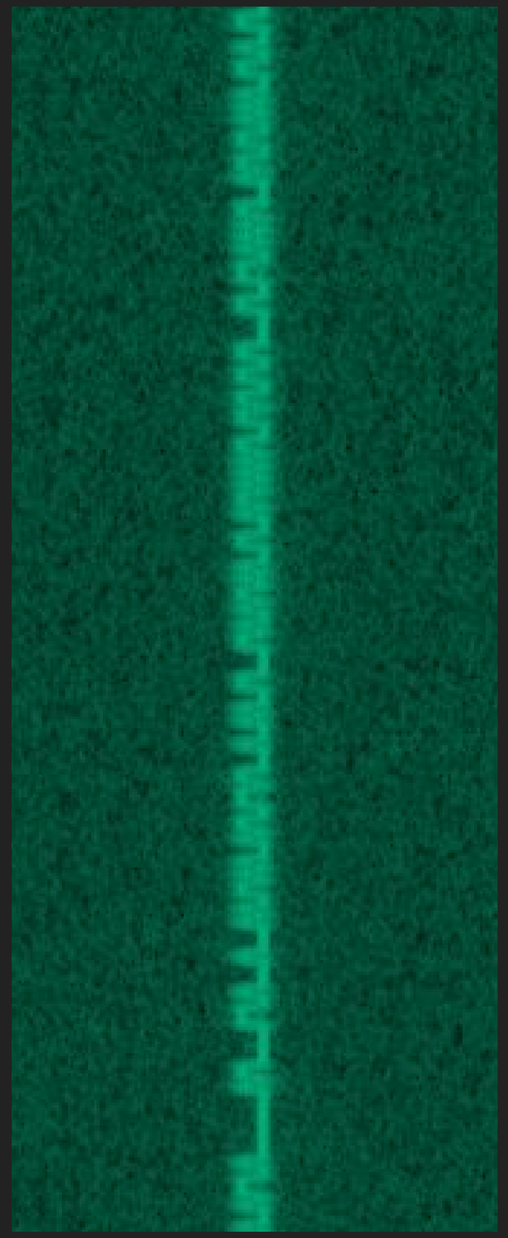
Figure 23. Operation of Data Whitening, Manchester Encoding, and CRC

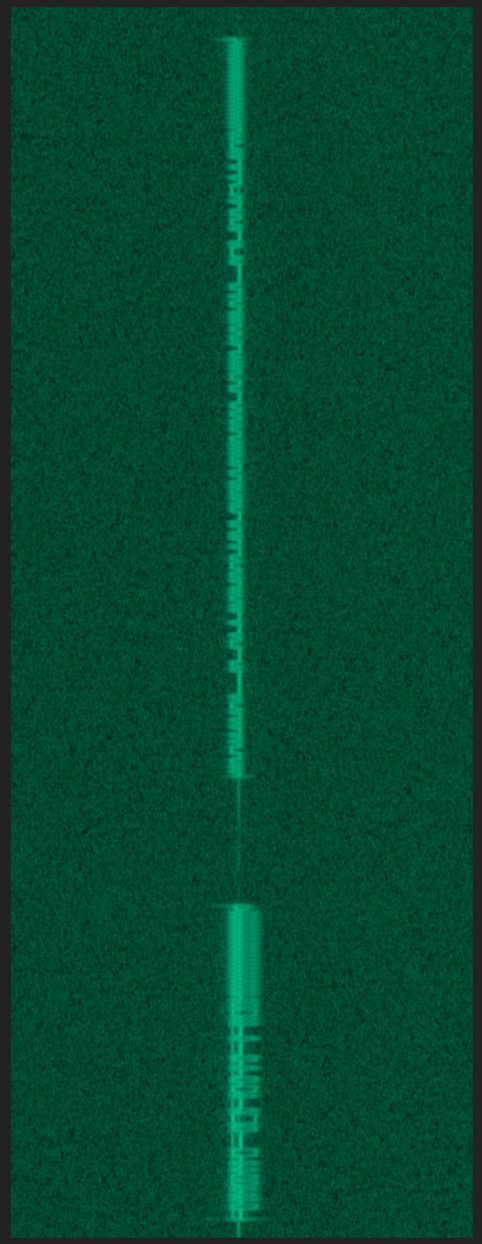
WoneNice WN6200 Reverse Engineering Workflow

0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → 433 MHz
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

Investigate with Baudline







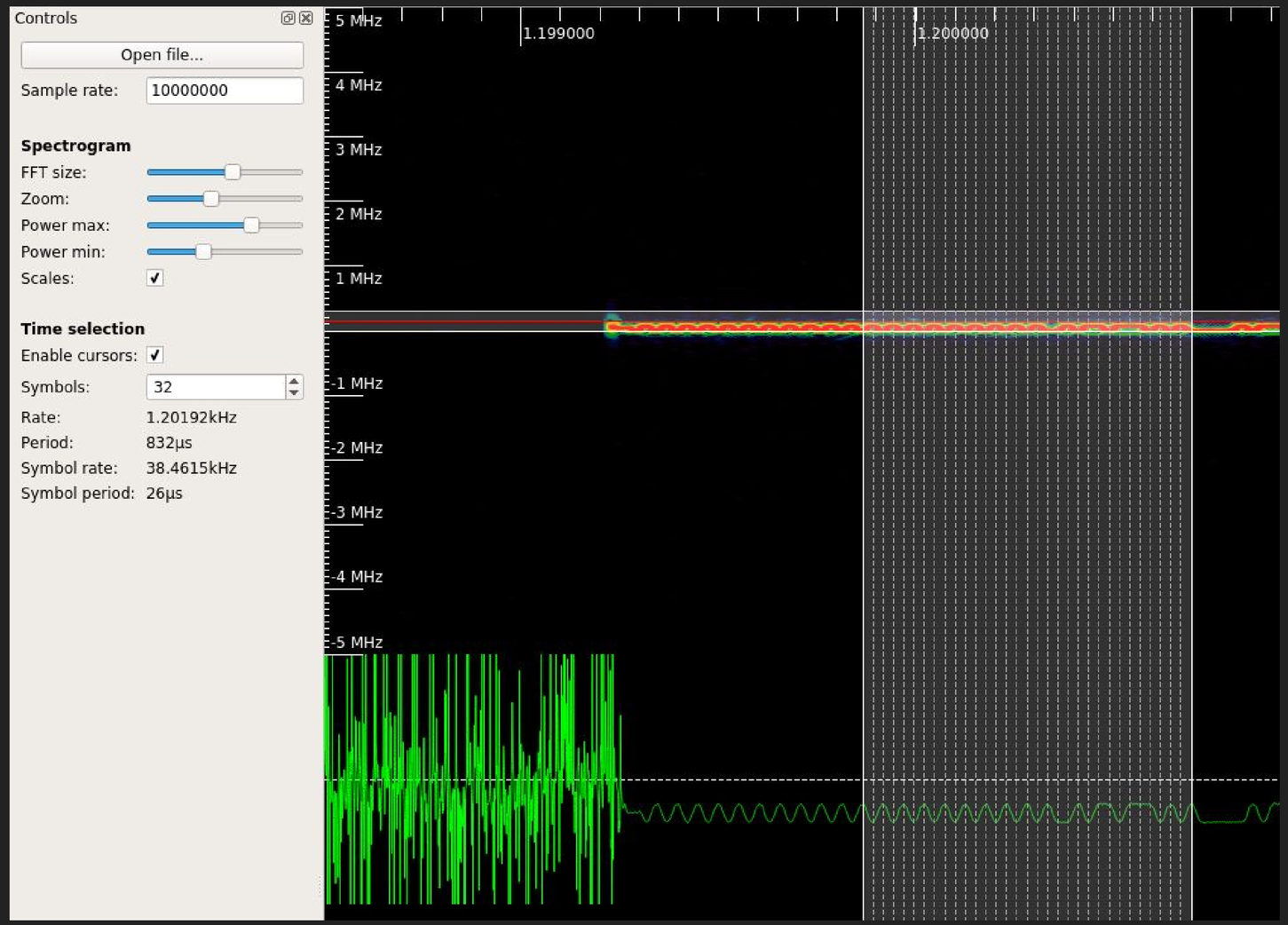
WoneNice WN6200 Reverse Engineering Workflow

0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → ~~433 MHz~~ 434 MHz
2. ✓ Identify the modulation → Frequency-shift Keying
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

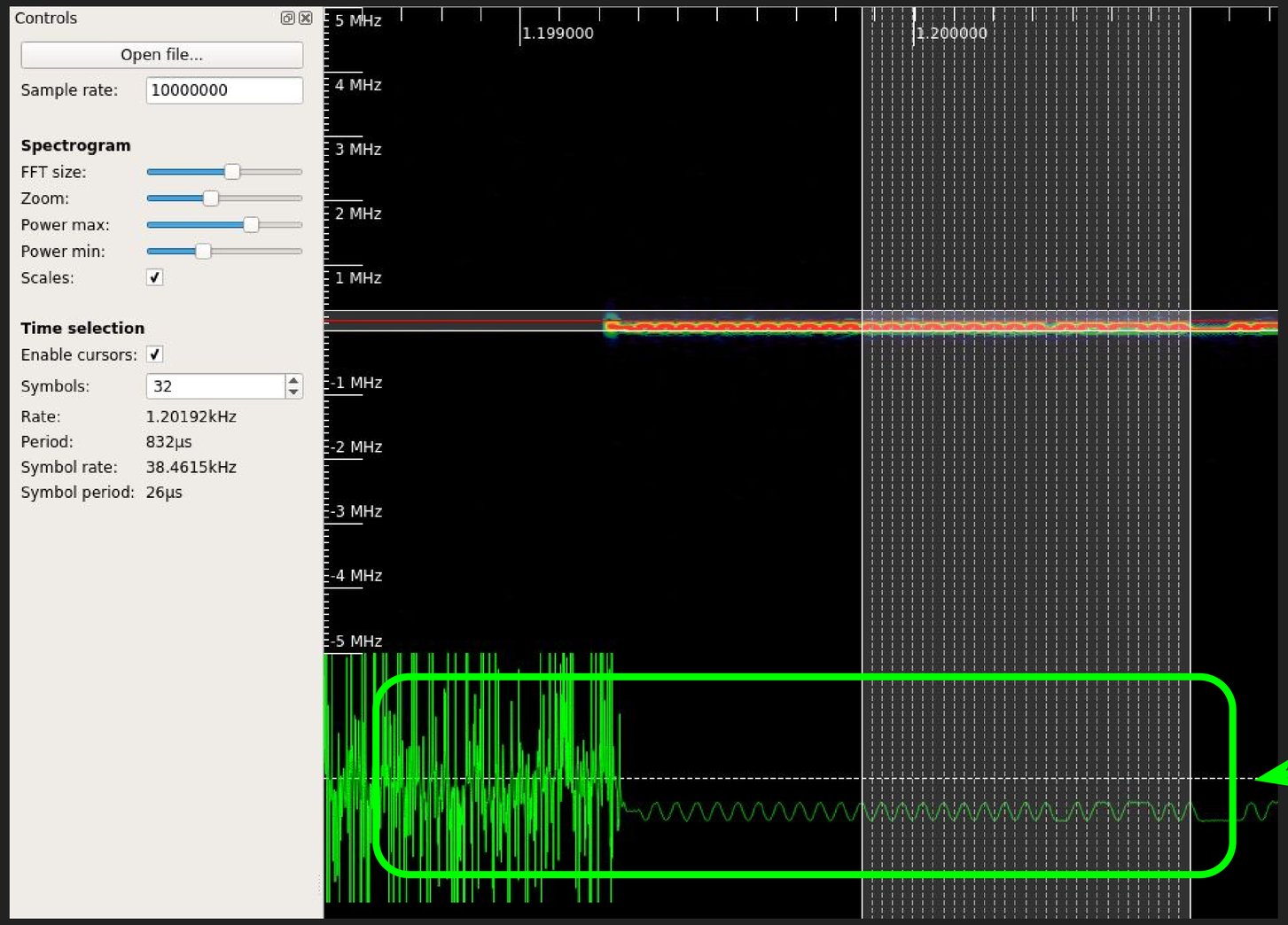
Capture some IQ data

```
$ uhd_rx_cfile \
-A TX/RX \
-r 10e6 \
-f 434e6
-g 20 \
-N 100e6 \
barcode-reader.iq
```

Barcode Reader Packet in Inspectrum

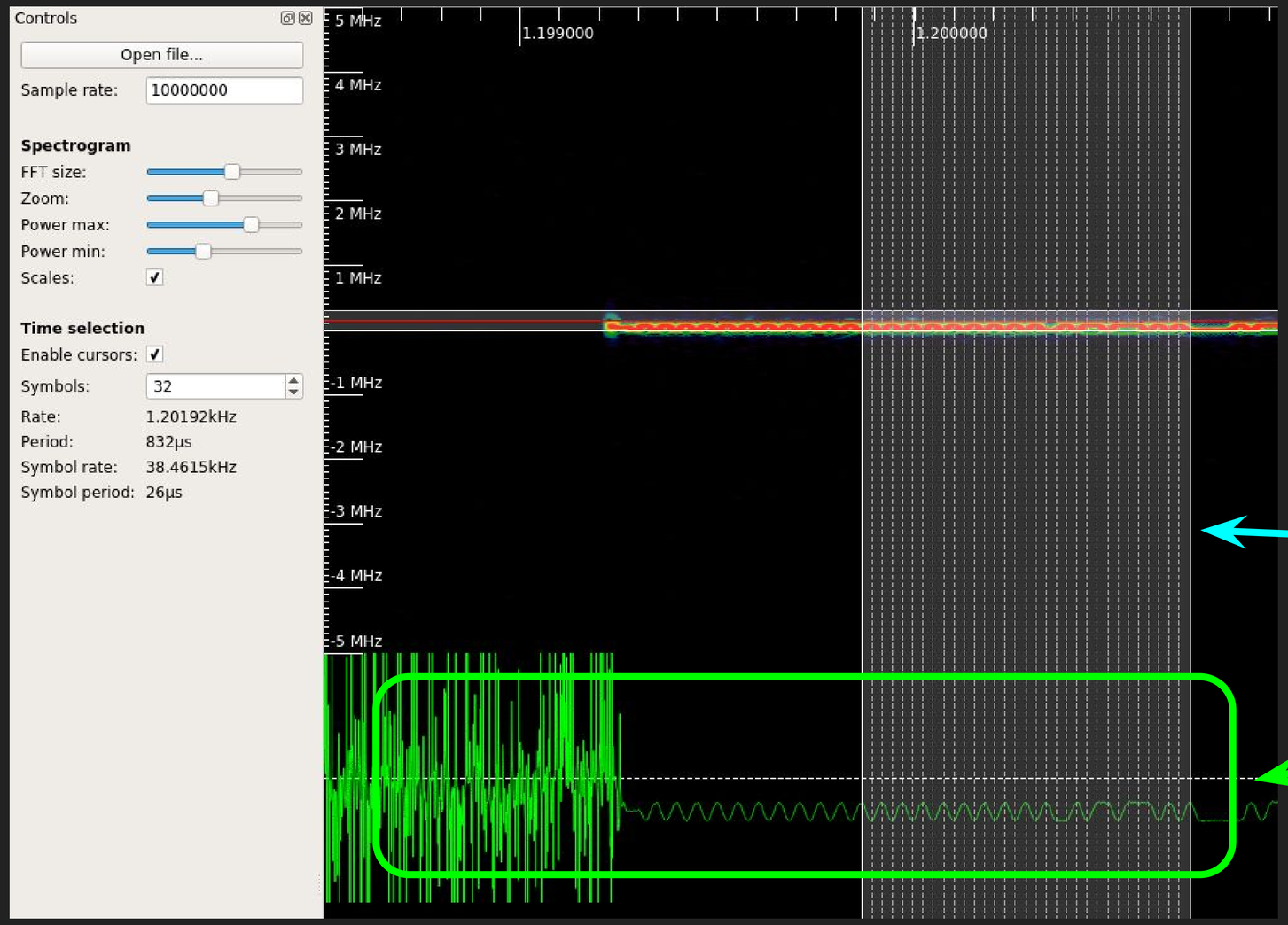


Barcode Reader Packet in Inspectrum



Frequency Plot

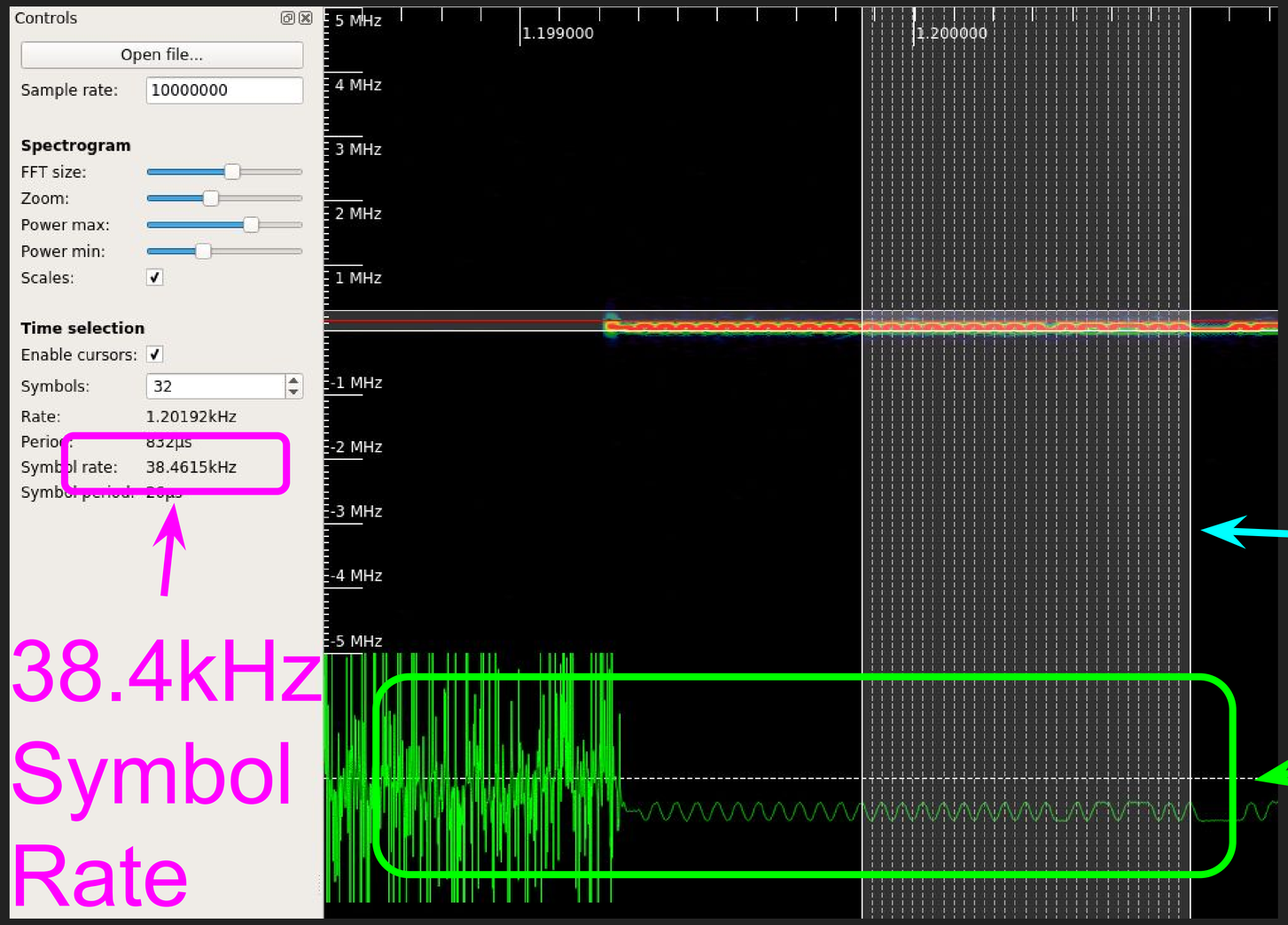
Barcode Reader Packet in Inspectrum



Cursors

Frequency Plot

Barcode Reader Packet in Inspectrum



38.4kHz Symbol Rate

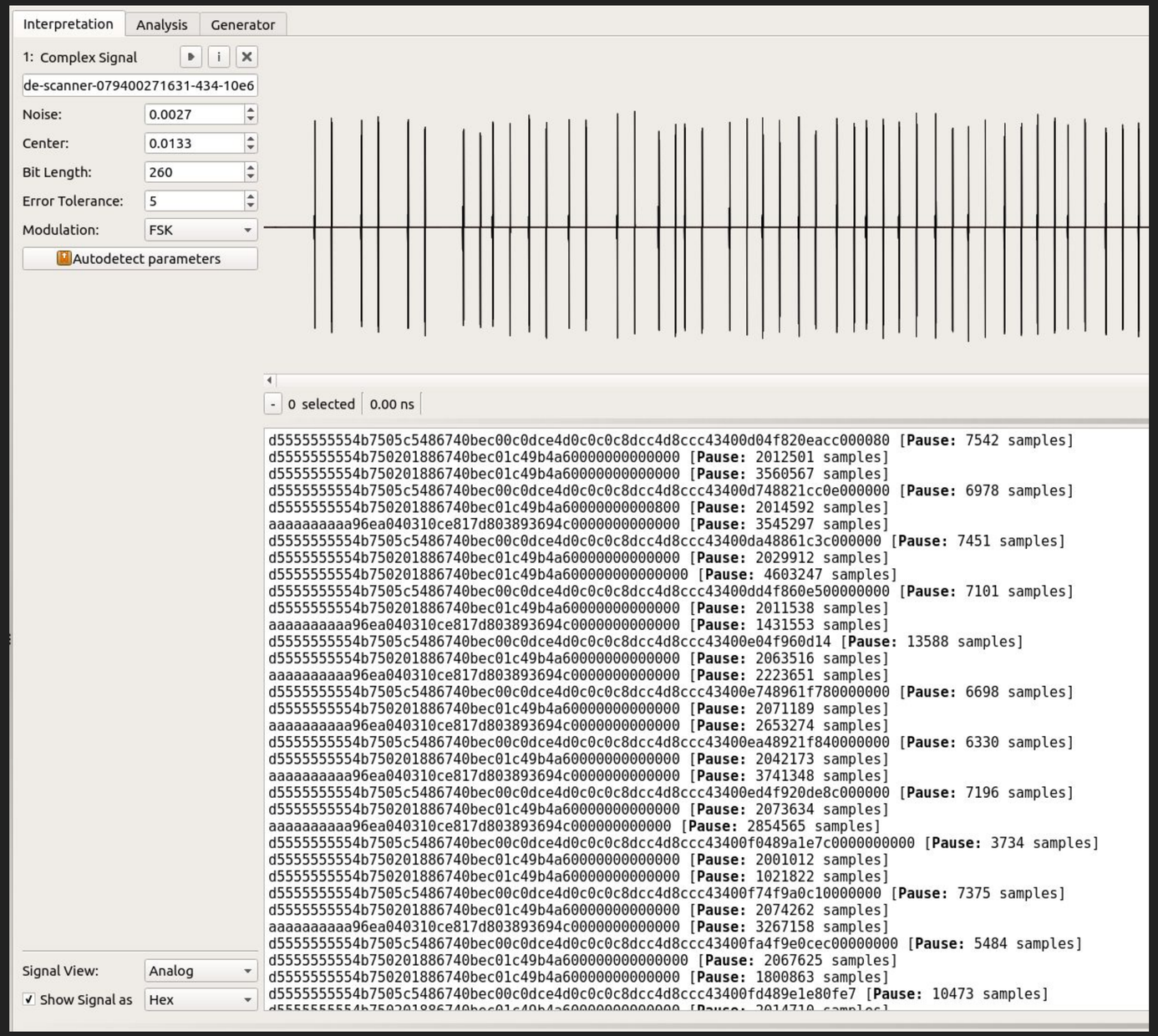
Cursors

Frequency Plot

WoneNice WN6200 Reverse Engineering Workflow

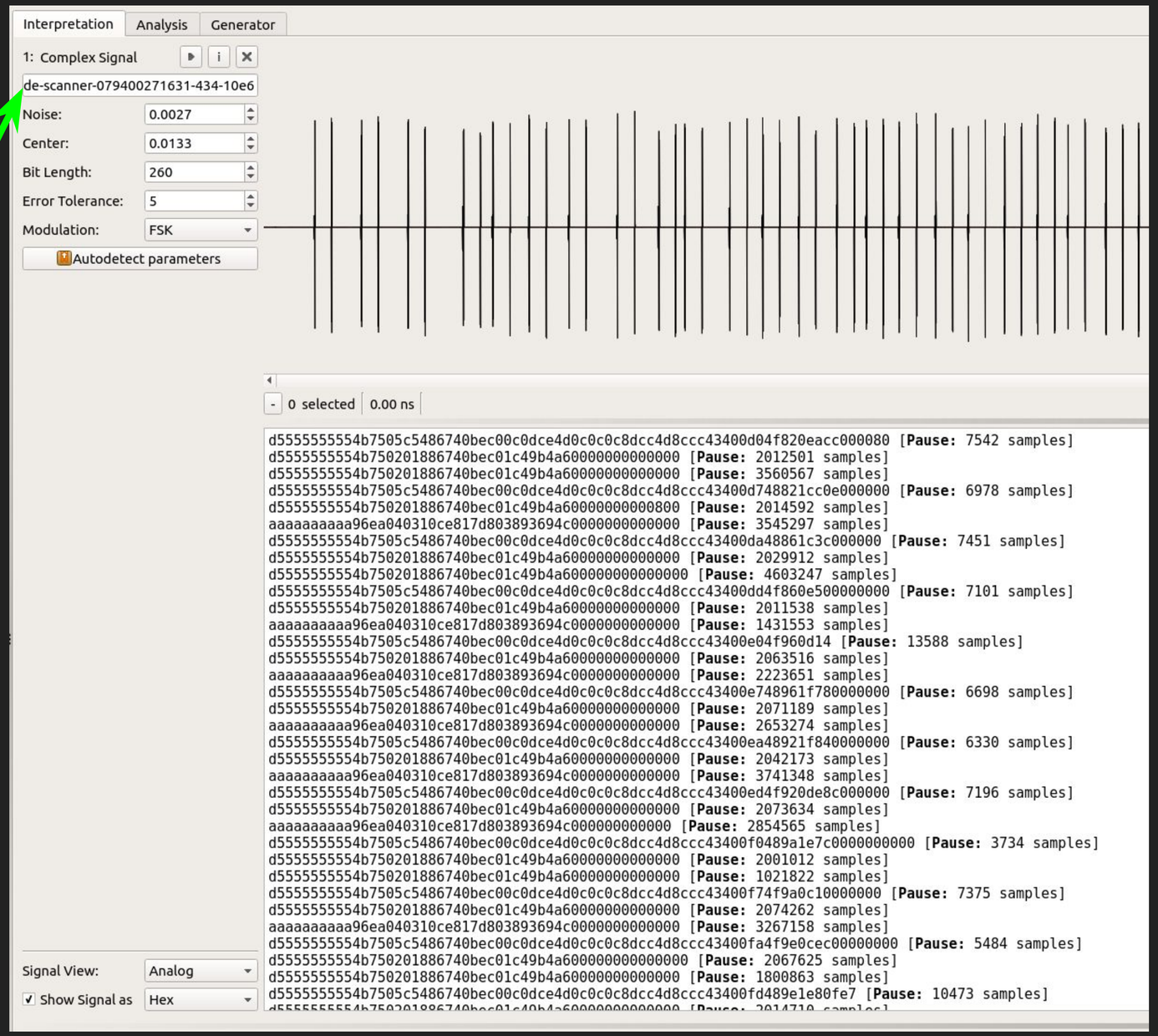
0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → 434 MHz
2. ✓ Identify the modulation → Frequency-shift Keying
3. ✓ Determine the symbol rate → 38.4 kHz
4. Synchronize
5. Extract symbols

Barcode Scanner in Universal Radio Hacker



Barcode Scanner in Universal Radio Hacker

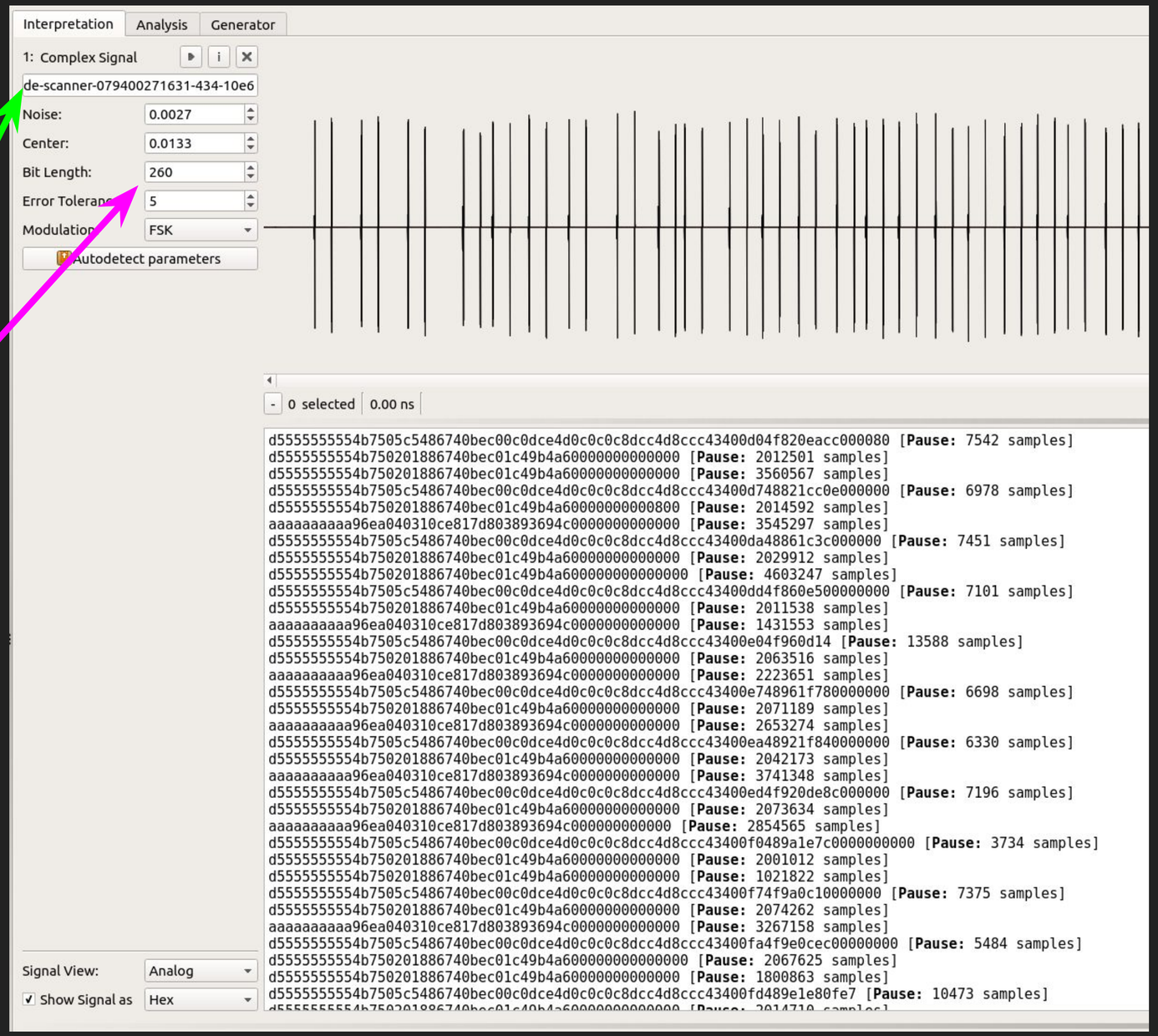
IQ file we recorded earlier



Barcode Scanner in Universal Radio Hacker

IQ file we recorded earlier

Samples per Symbol (10e6 / 34.8e3)

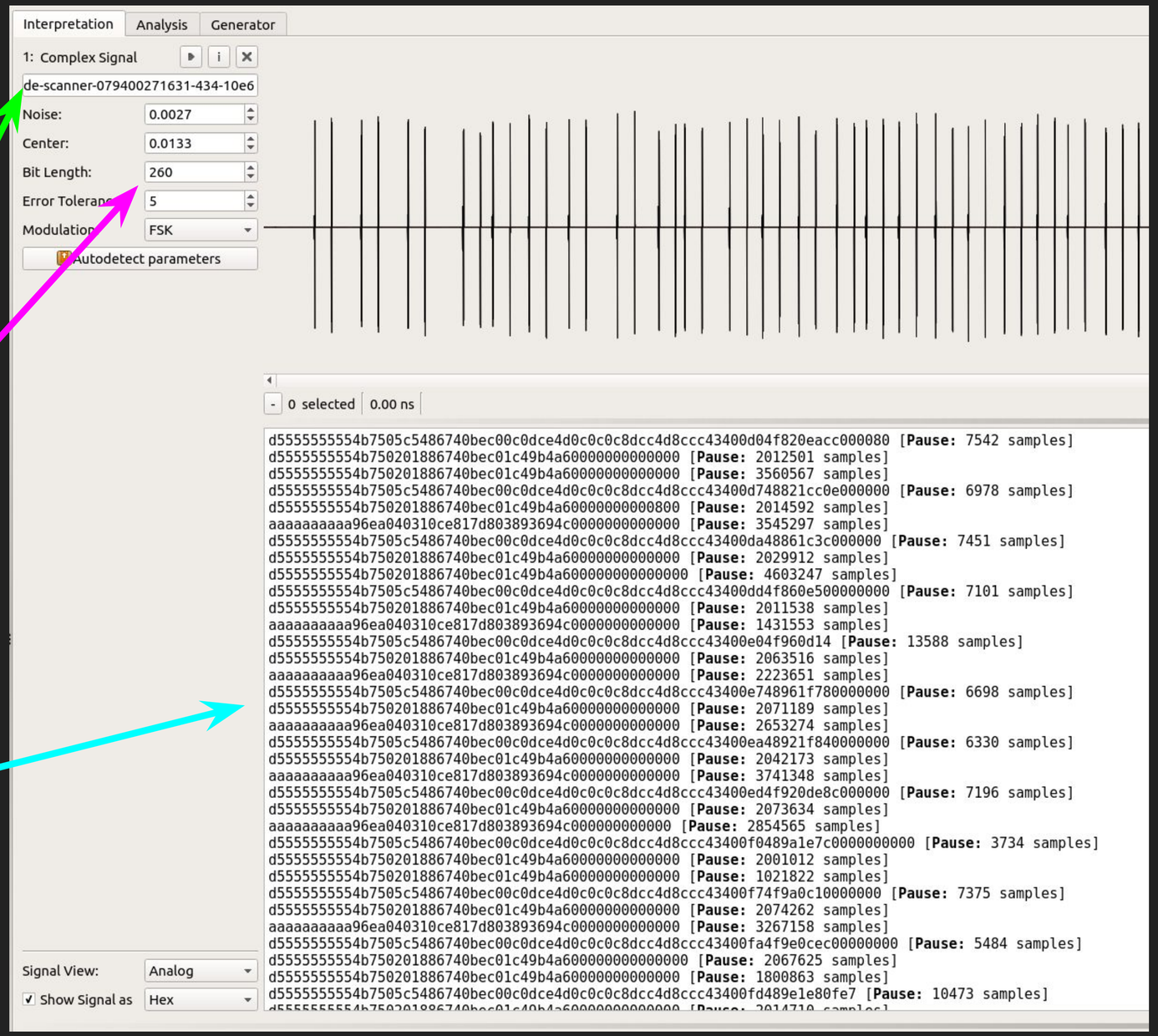


Barcode Scanner in Universal Radio Hacker

IQ file we recorded earlier

Samples per Symbol (10e6 / 34.8e3)

Auto-magically identified packets



Barcode Scanner Packets

5555555554b7505c5486740bec00**c0dce4d0c0c0c8dcc4d8ccc4**3400**d04f820eacc**

5555555554b7505c5486740bec00**e0e0d8c4c0e0e4dce4e4ccc8**3400**0cda766ac80**

5555555554b7505c5486740bec00**e0e0d8c4c0e0e4e0c0c0d0d0**3400**27682f8e848**

Barcode Scanner Packets

5555555554b7505c5486740bec00**c0dce4d0c0c0c8dcc4d8ccc4**3400**d04f820eacc**

5555555554b7505c5486740bec00**e0e0d8c4c0e0e4dce4e4ccc8**3400**0cda766ac80**

5555555554b7505c5486740bec00**e0e0d8c4c0e0e4e0c0c0d0d0**3400**27682f8e848**

12 bytes
(same length as the scanned barcodes)

Barcode Scanner Packet Checksum

```
$ ./reveng -w 16 -s  \
4b7505c5486740bec00c0dce4d0c0c0c8dcc4d8ccc43400d04f820eacc \
4b7505c5486740bec00e0e0d8c4c0e0e4dce4e4ccc834000cda766ac80 \
4b7505c5486740bec00e0e0d8c4c0e0e4e0c0c0d0d0340027682f8e848
```

```
./reveng: no models found
```

Barcode Scanner Packet Format

```
555555555 4b7505c5486740bec00 c0dce4d0c0c0c8dcc4d8ccc4 3400d04f820eacc
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4dce4e4ccc8 34000cda766ac80
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4e0c0c0d0d0 340027682f8e848
```

Barcode Scanner Packet Format

```
555555555 4b7505c5486740bec00 c0dce4d0c0c0c8dcc4d8ccc4 3400d04f820eacc
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4dce4e4ccc8 34000cda766ac80
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4e0c0c0d0d0 340027682f8e848
```

preamble

Barcode Scanner Packet Format

```
555555555 4b7505c5486740bec00 c0dce4d0c0c0c8dcc4d8ccc4 3400d04f820eacc
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4dce4e4ccc8 34000cda766ac80
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4e0c0c0d0d0 340027682f8e848
```

preamble

sync + header?

Barcode Scanner Packet Format

```
555555555 4b7505c5486740bec00 c0dce4d0c0c0c8dcc4d8ccc4 3400d04f820eacc
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4dce4e4ccc8 34000cda766ac80
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4e0c0c0d0d0 340027682f8e848
```

preamble

sync + header?

barcode

Barcode Scanner Packet Format

```
555555555 4b7505c5486740bec00 c0dce4d0c0c0c8dcc4d8ccc4 3400d04f820eacc
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4dce4e4ccc8 34000cda766ac80
555555555 4b7505c5486740bec00 e0e0d8c4c0e0e4e0c0c0d0d0 340027682f8e848
```

preamble

sync + header?

barcode

???

WoneNice WN6200 Reverse Engineering Workflow

0. ✓ Open-source intelligence research
1. ✓ Characterize the channel → 434 MHz
2. ✓ Identify the modulation → Frequency-shift Keying
3. ✓ Determine the symbol rate → 38.4 kHz
4. ✓ Synchronize → Preamble and SFD (assumed)
5. ✓ Extract symbols

Demo Time!

Wireless Presentation Clicker

~~Logitech~~ #China R400

~~Logitech~~ #China R400

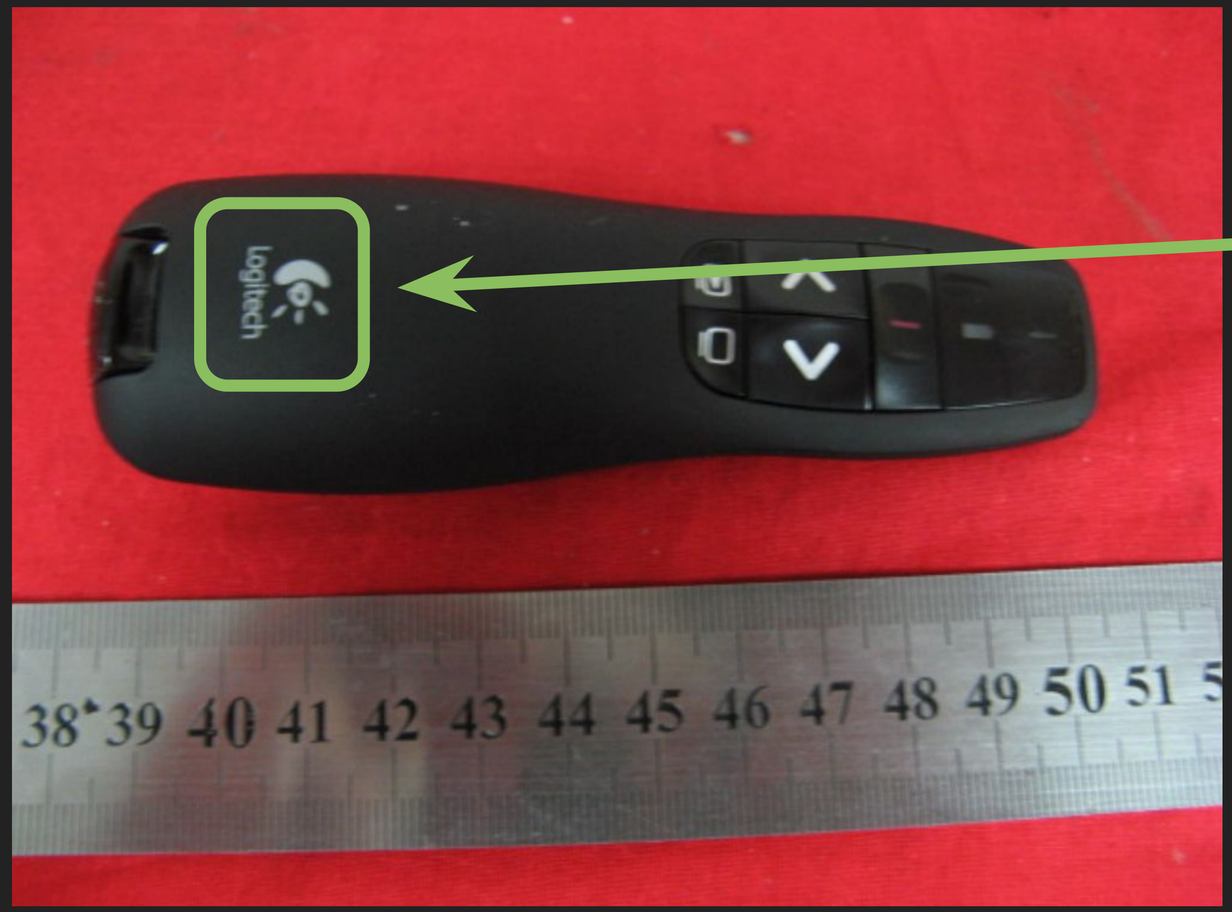
- 2.4GHz Wireless Presentation Clicker
- Remote
  - FCC ID SMFR-R0004
- Dongle
  - FCC ID SMFC-U0005

#China R400 Reverse Engineering Workflow

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

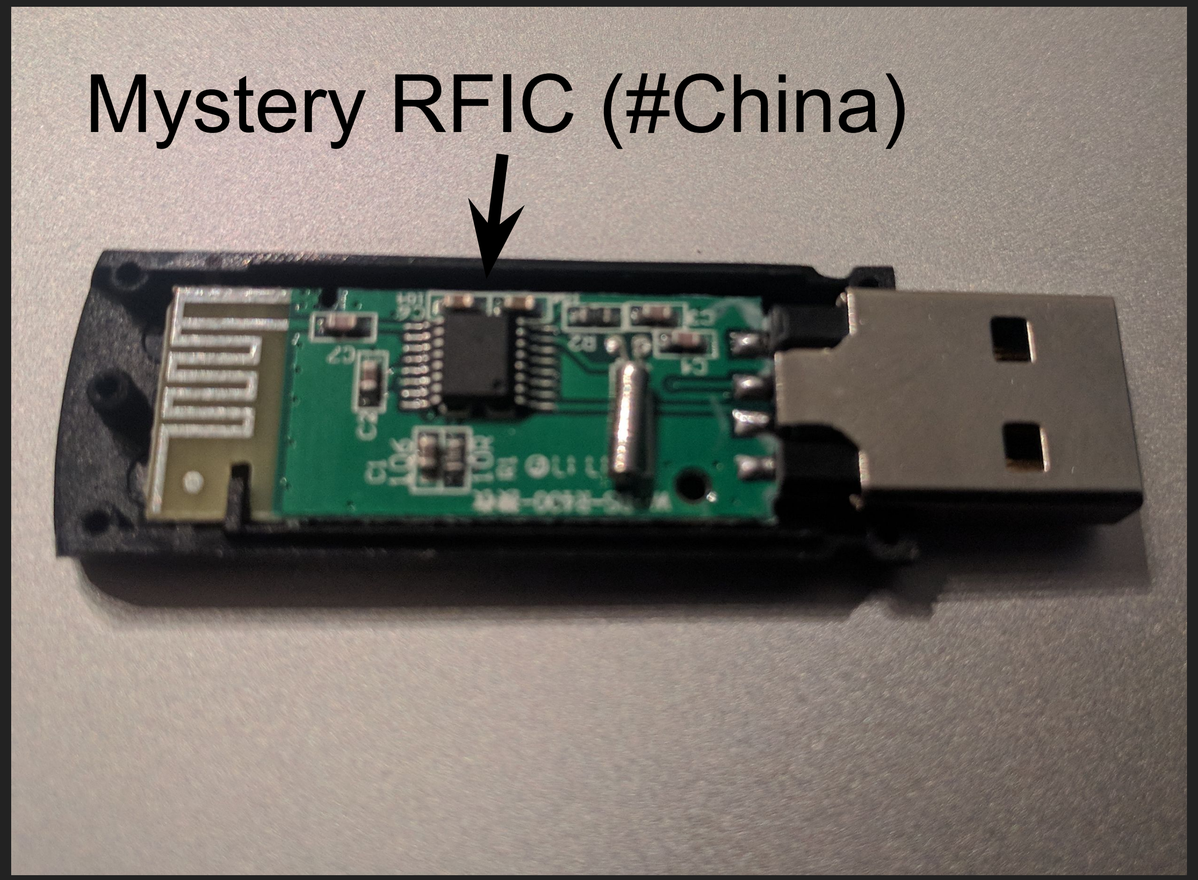
#China R400 FCC Photos

#China R400 FCC Photos



Logitech???

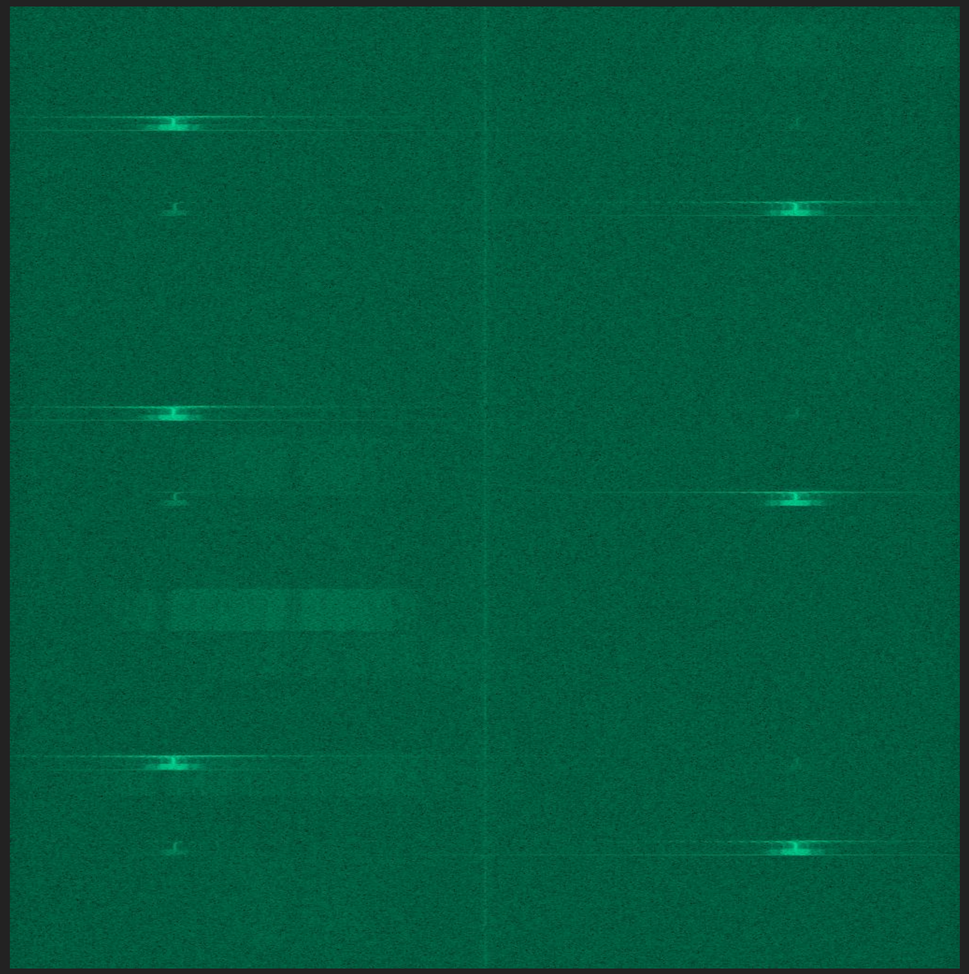
#China R400 FCC Photos vs Reality

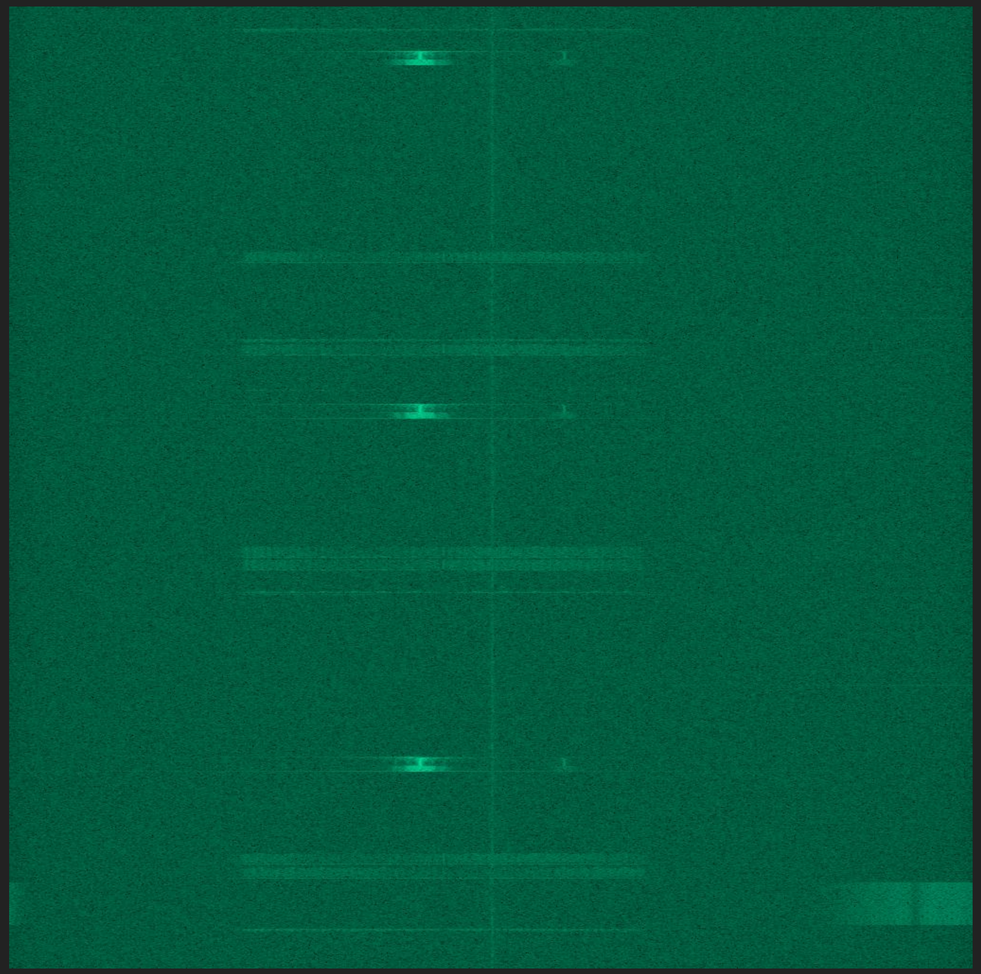


#China R400 Reverse Engineering Workflow

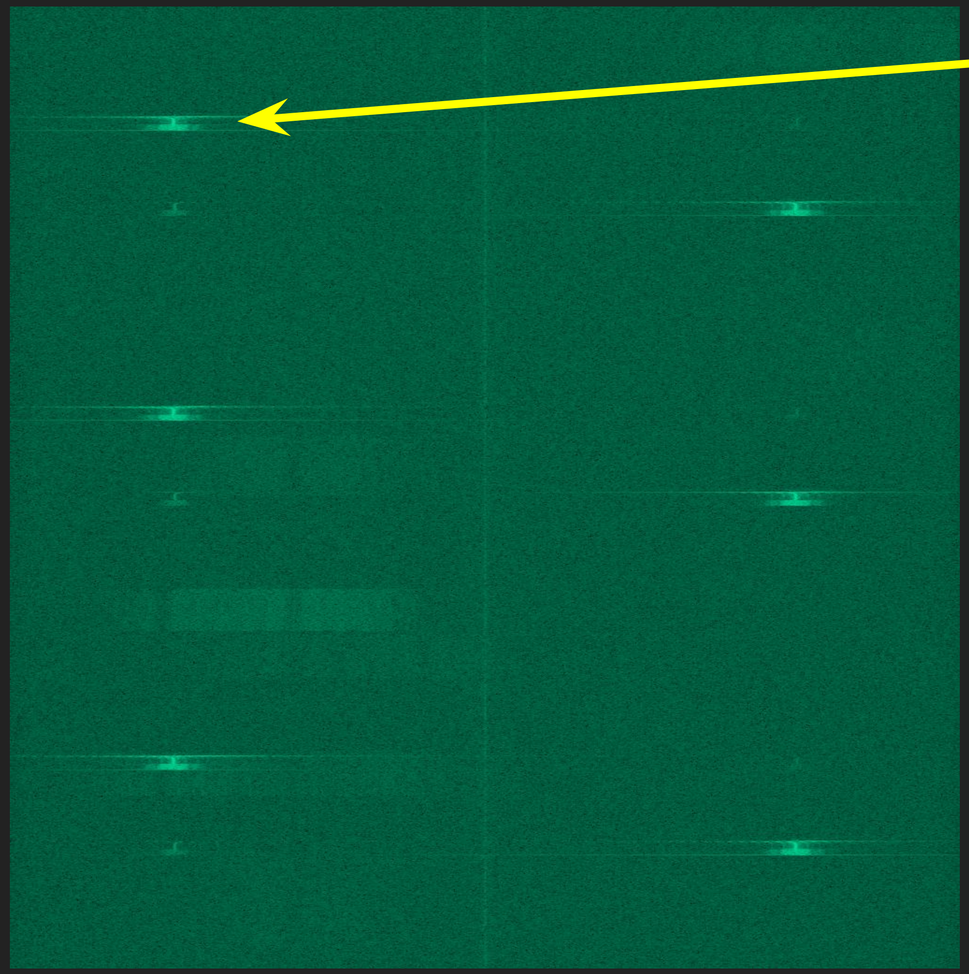
0. ~~Open-source intelligence research~~
1. Characterize the channel
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

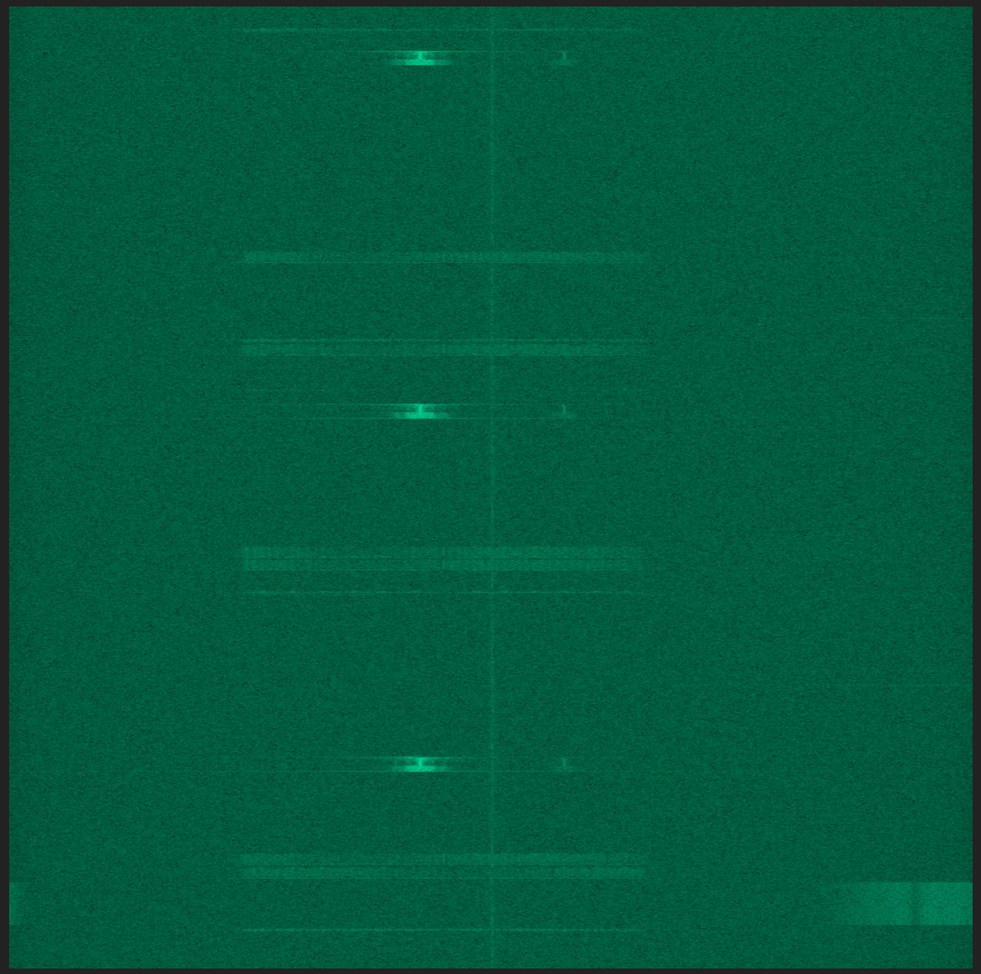
#China R400 in Baudline (holding down ‘left’ button)



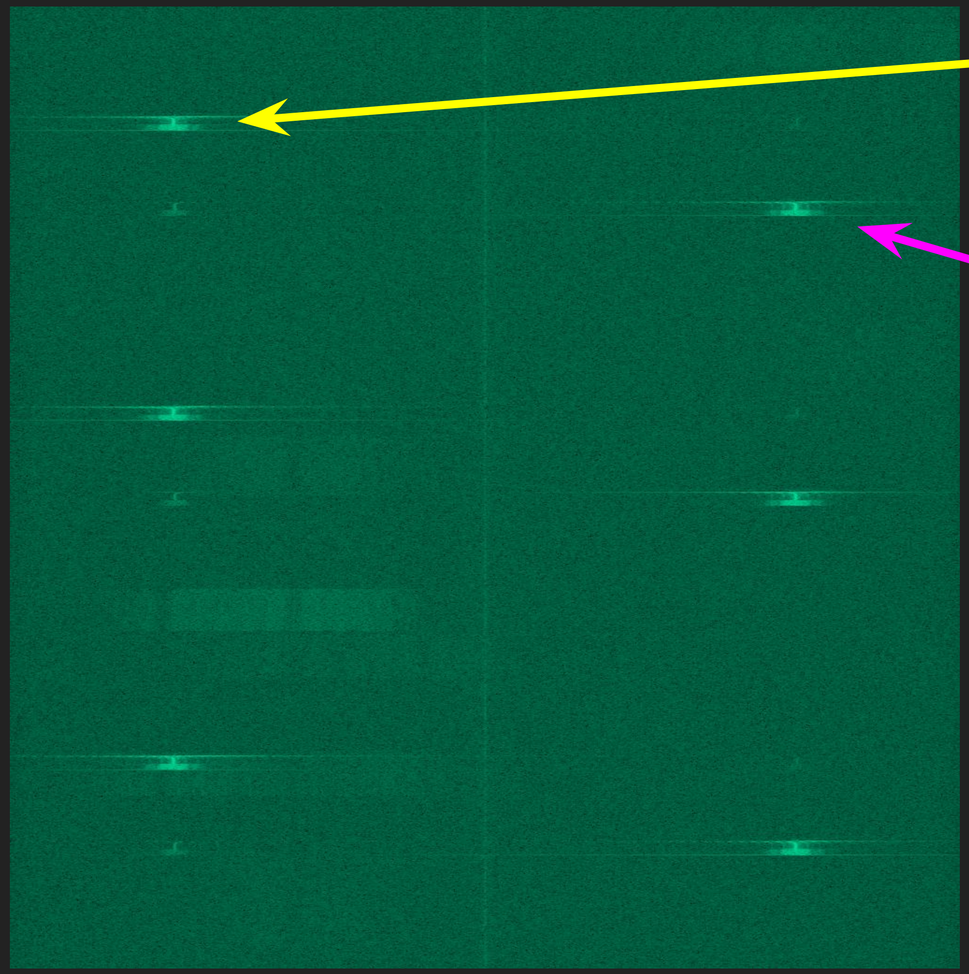


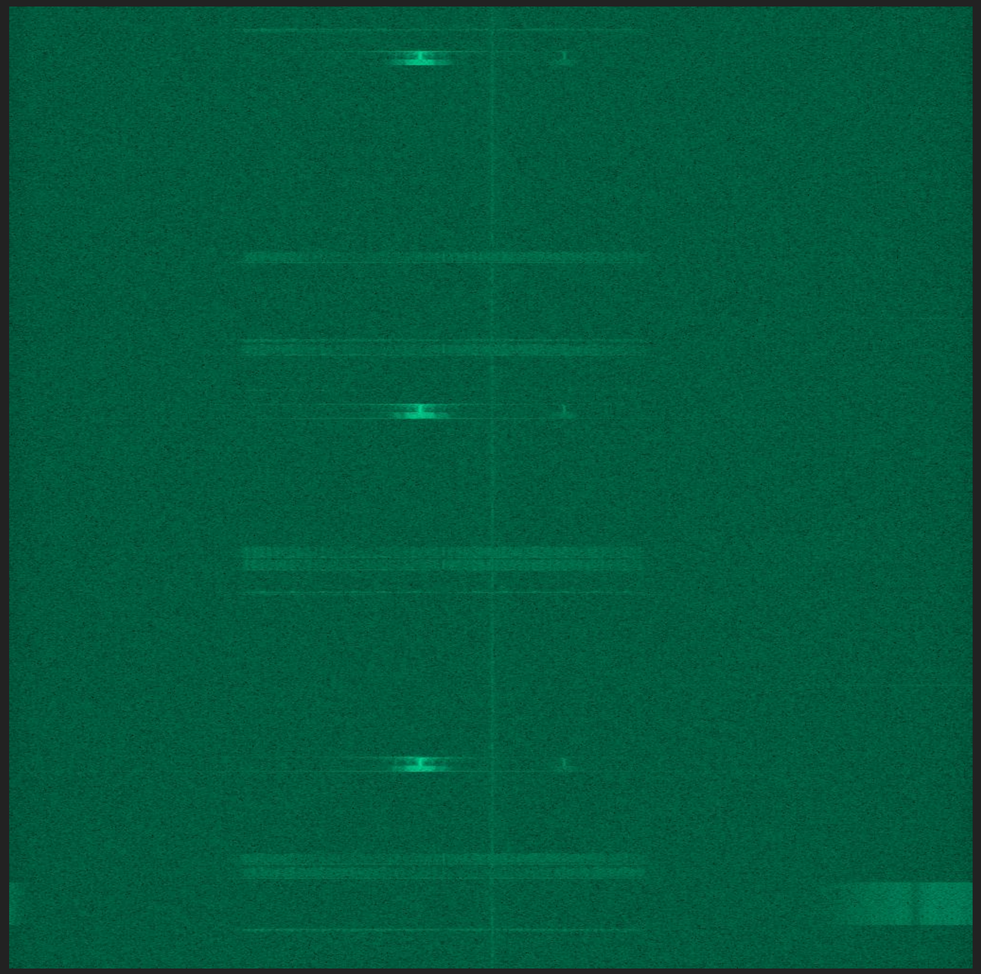
#China R400 in Baudline (holding down ‘left’ button)



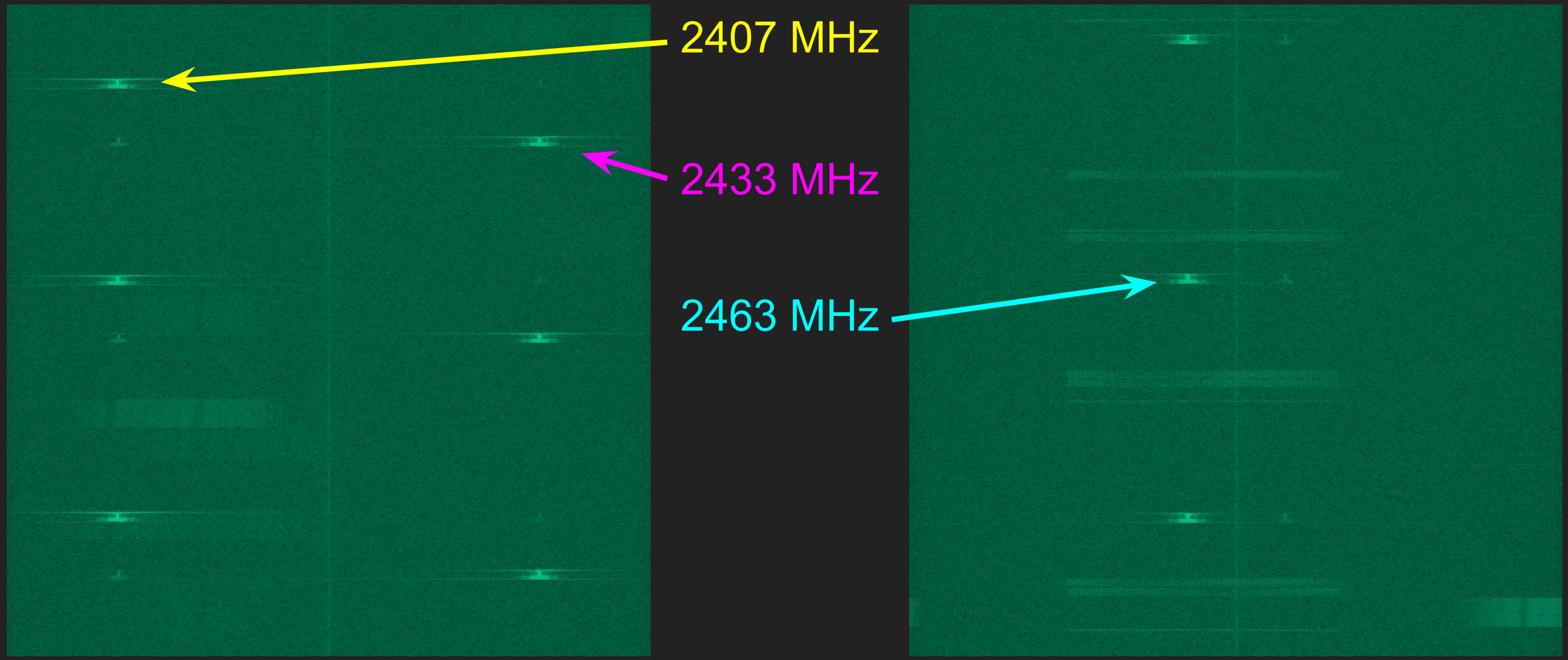


#China R400 in Baudline (holding down ‘left’ button)





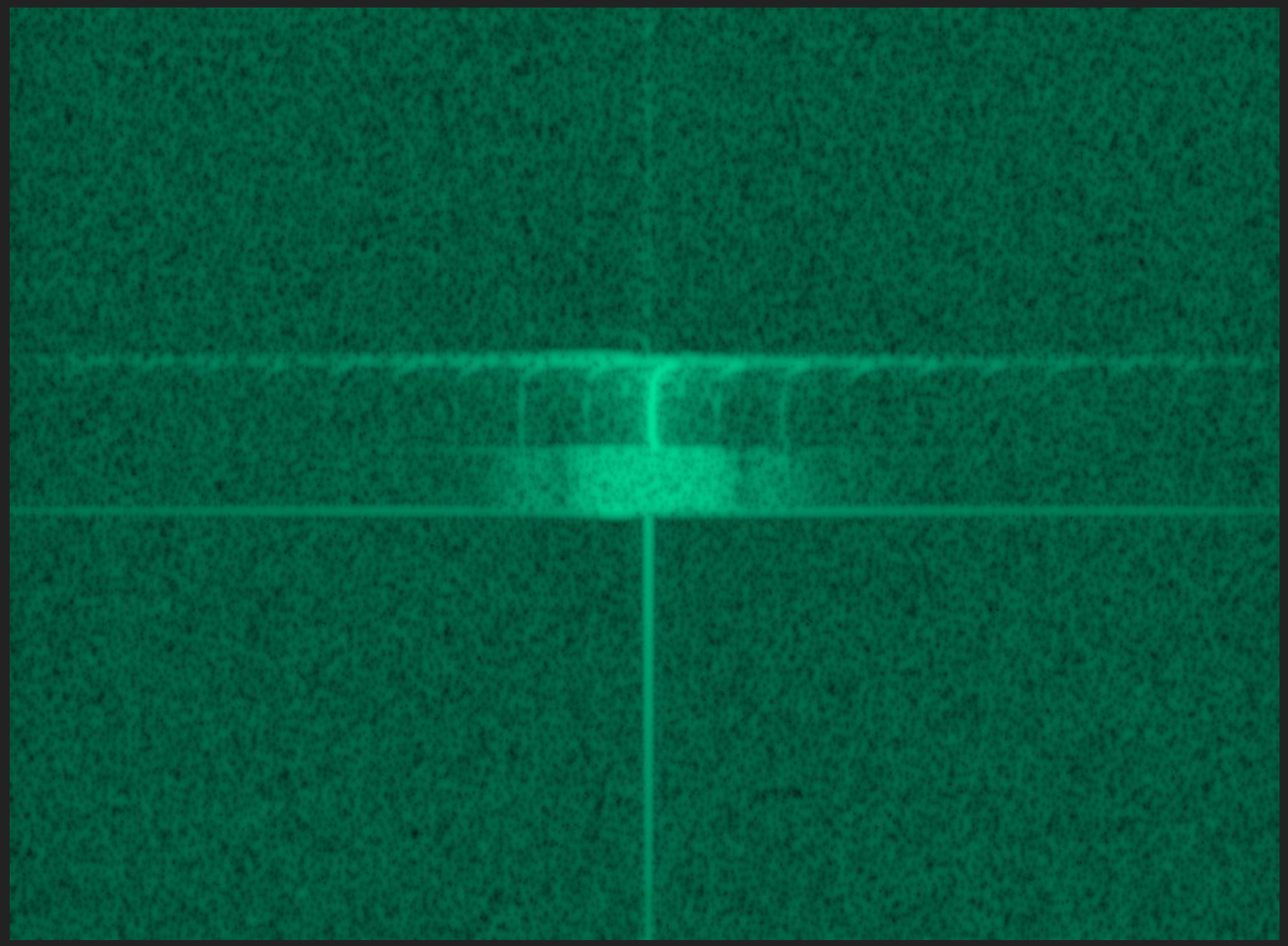
#China R400 in Baudline (holding down ‘left’ button)



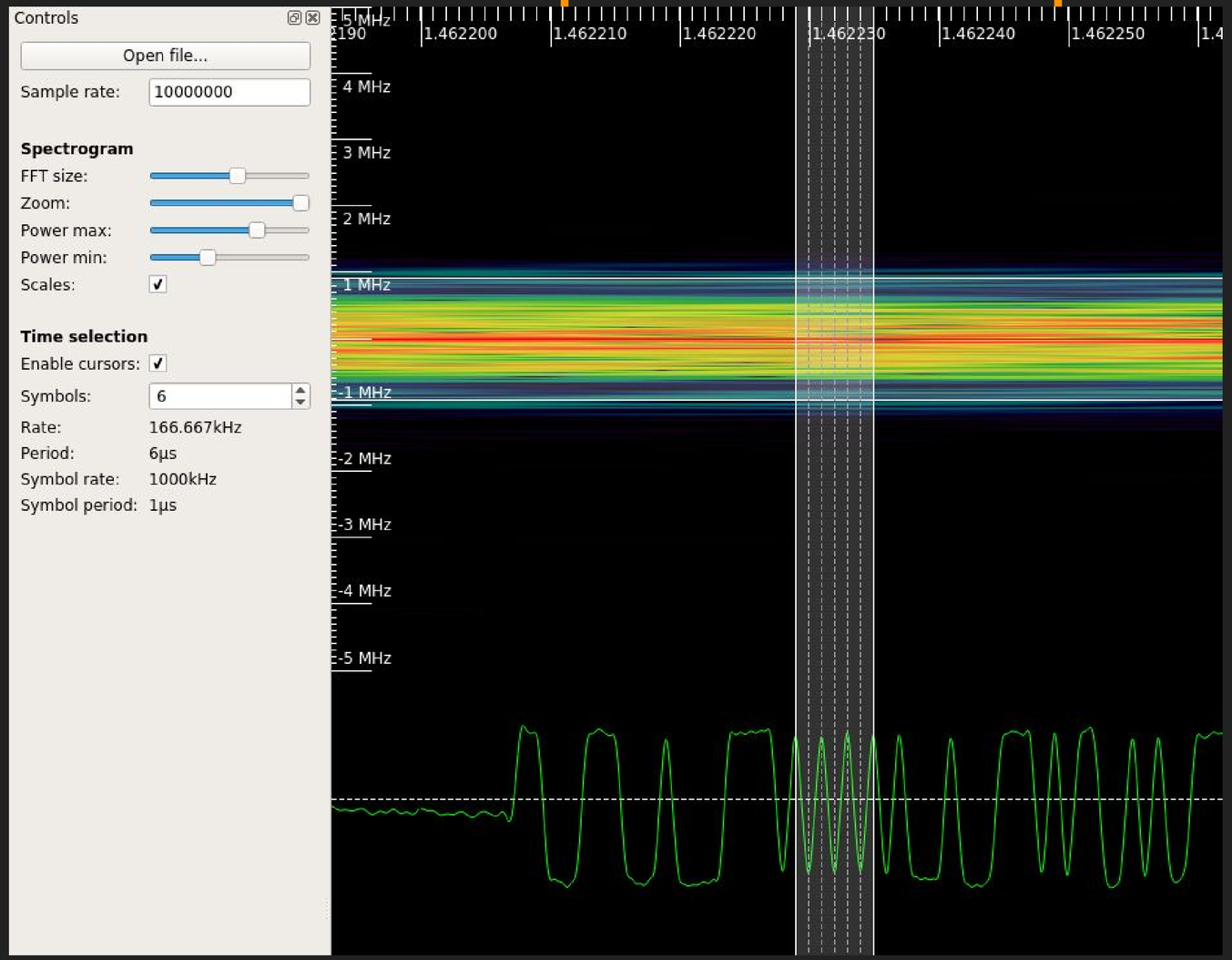
#China R400 Reverse Engineering Workflow

0. ~~Open-source intelligence research~~
1. ✓ Characterize the channel → 2407 MHz, 2433 MHz, 2463 MHz
2. Identify the modulation
3. Determine the symbol rate
4. Synchronize
5. Extract symbols

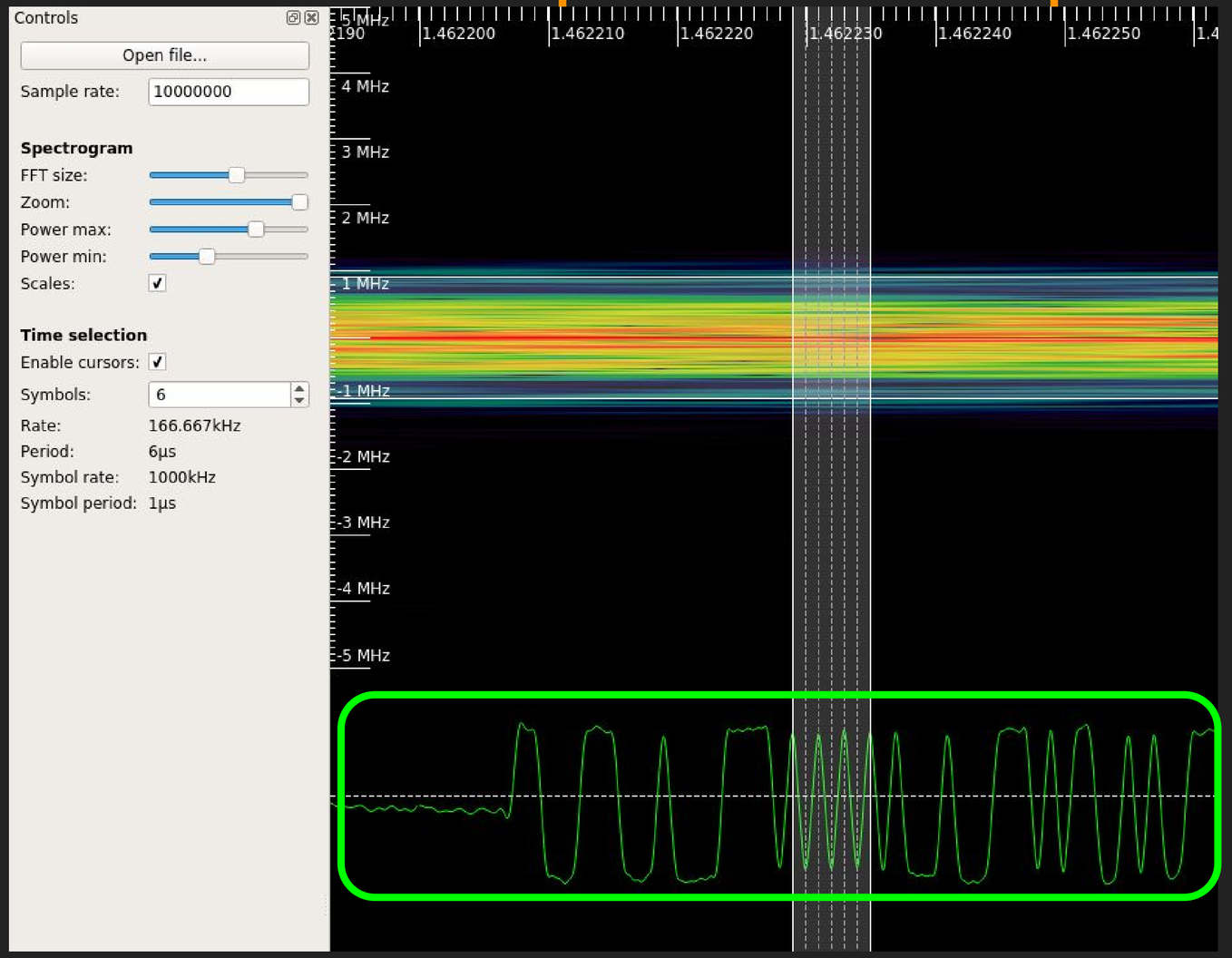
#China R400 packet in Baudline



#China R400 packet in Inspectrum

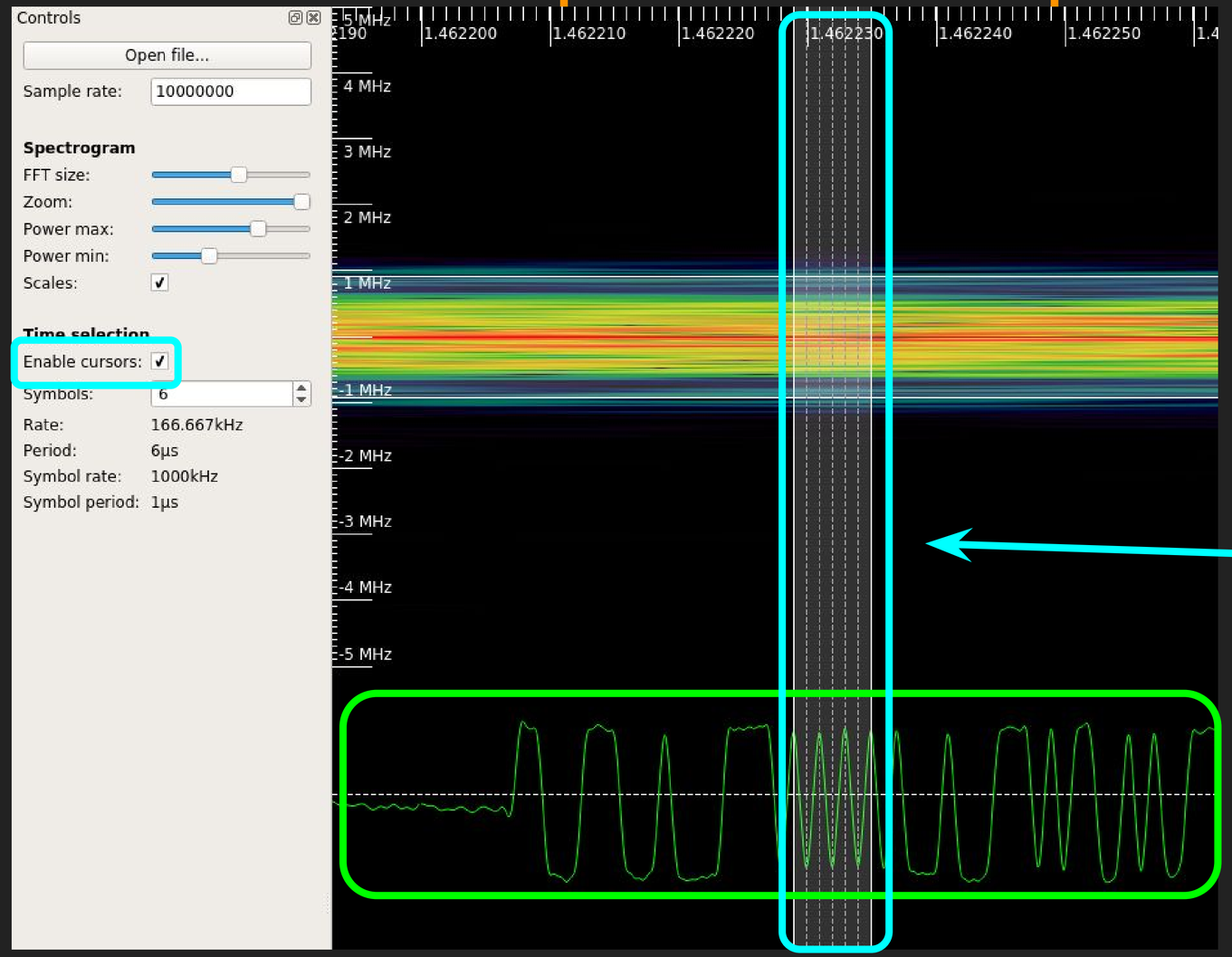


#China R400 packet in Inspectrum



Frequency Plot

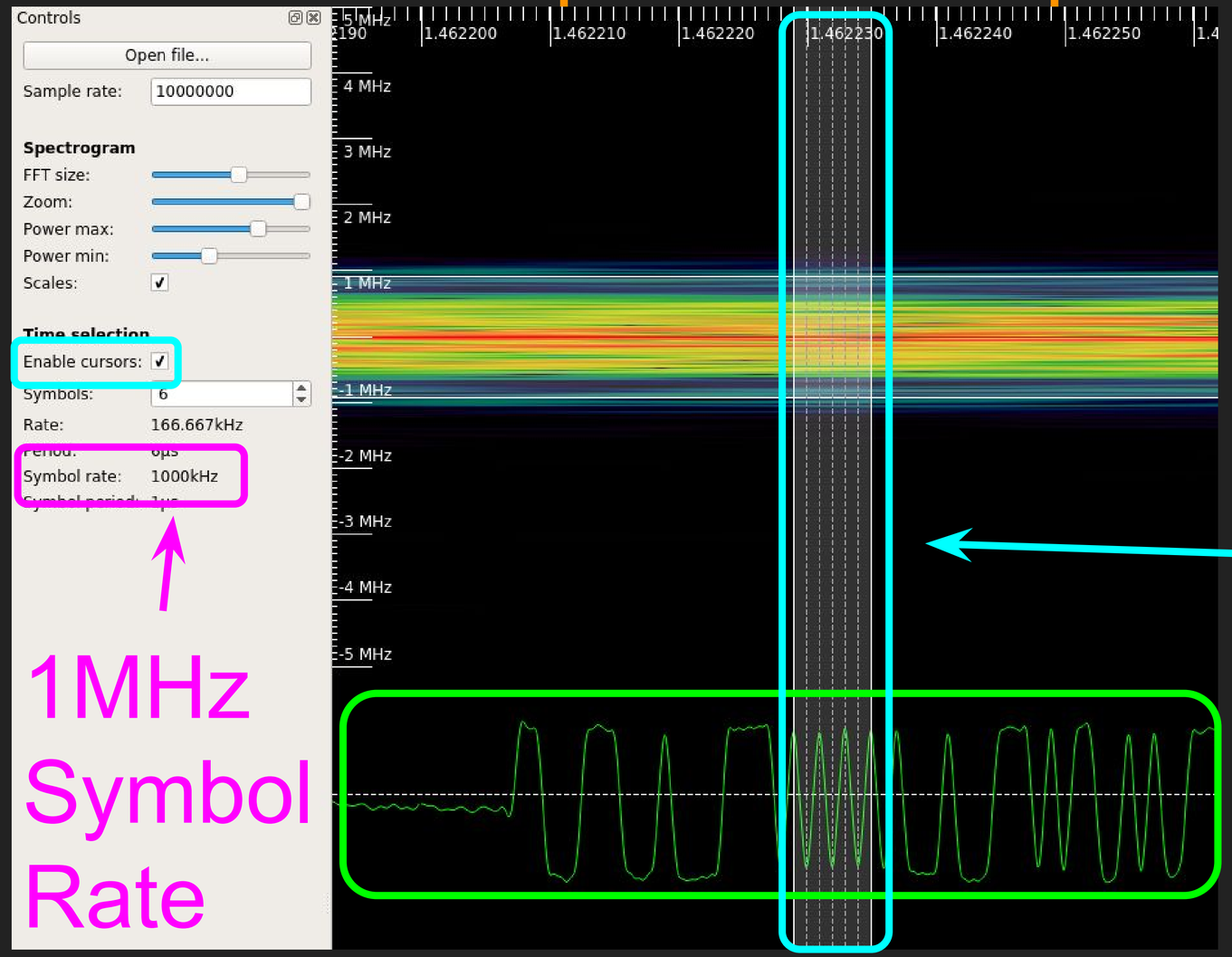
#China R400 packet in Inspectrum



Cursors

Frequency Plot

#China R400 packet in Inspectrum



1MHz Symbol Rate

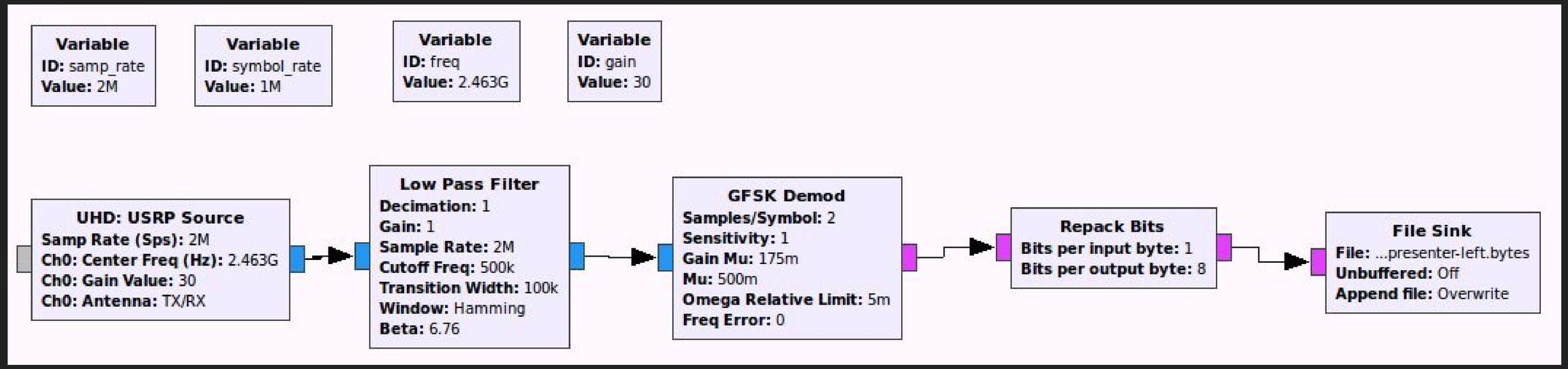
Cursors

Frequency Plot

#China R400 Reverse Engineering Workflow

0. ~~Open-source intelligence research~~
1. ✓ Characterize the channel → 2407 MHz, 2433 MHz, 2463 MHz
2. ✓ Identify the modulation → Frequency-shift Keying
3. ✓ Determine the symbol rate → 1 MHz
4. Synchronize
5. Extract symbols

#China R400 Demodulator 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

Presenter Packet Bytes (left)

```
0071c43d5511d653855d9f299836ba3220aa9abf
```

Presenter Packet Bytes (left, right, play, fs)

001c710f55447594e15767ca660dae8c882aa6af

001c710f55447594e1f767ca660dae8c88ac4faf

001c710f55447594e1f967ca660daeccc87756af

001c710f55447594e12567ca660dae8c88150caf

Presenter Packet Bytes (left, right, play, fs)

001c710f55447594e1**57**67ca660dae**8c882aa6**af

001c710f55447594e1**f7**67ca660dae**8c88ac4f**af

001c710f55447594e1**f9**67ca660dae**ccc87756**af

001c710f55447594e1**25**67ca660dae**8c88150c**af

Presenter Packet Checksum (CRC-16)

```
$ ./reveng -w 16 -s  \
001c710f55447594e15767ca660dae8c882aa6af \
001c710f55447594e1f767ca660dae8c88ac4faf \
001c710f55447594e1f967ca660daeccc87756af \
001c710f55447594e12567ca660dae8c88150caf

width=16  poly=0x1021  init=0xda90  refin=false  refout=false
 xorout=0x0000  check=0xadfb  residue=0x0000  name=(none)
```

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync?

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync? header?

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync? header? button

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync? header? button ???

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync? header? button ??? crc16

Presenter Packet Format

```
00 1c710f55 447594e1 57 67ca660dae8c88 2aa6 af
00 1c710f55 447594e1 f7 67ca660dae8c88 ac4f af
00 1c710f55 447594e1 f9 67ca660daeccc8 7756 af
00 1c710f55 447594e1 25 67ca660dae8c88 150c af
```

preamble sync? header? button ??? crc16 trailer

#China R400 Reverse Engineering Workflow

0. ~~Open-source intelligence research~~
1. ✓ Characterize the channel → 2407 MHz, 2433 MHz, 2463 MHz
2. ✓ Identify the modulation → Frequency-shift Keying
3. ✓ Determine the symbol rate → 1 MHz
4. ✓ Synchronize → Preamble and SFD (assumed)
5. ✓ Extract symbols

Demo Time!

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

Barcode Scanner	Security Siren	Presentation Clicker
434 Mhz	433.86 MHz	3x 2.4 GHz channels

2. Identify Modulation

Modulation is the biggest variable

(but OSINT makes identifying it easy)

Barcode Scanner	**Security Siren**	**Presentation Clicker**
Frequency Shift Keying	Pulse Width Modulation / On-Off Keying	Frequency Shift Keying

3. Symbol Rate Recovery

All 3 PHYs share a common notion of discrete symbol timing

Barcode Scanner	Security Siren	Presentation Clicker
38,400 symbols/s	4,220 symbols/s	1,000,000 symbols/s

4. Synchronization

All 3 PHYs contain synchronization features

(preamble and/or Start of Frame delimiter)

Barcode Scanner	**Security Siren**	**Presentation Clicker**
Preamble + Sync	1 0b1 + 31 0b0s	Preamble + Sync

5. Symbol Extraction

Once you get here it’s just bits on a disk

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

44con Workshop!

Join us for a live hacking session tomorrow from 11:30 to 13:30!

Please bring:

- Laptop w/ GNU Radio (check out the GNU Radio Live USB)
- Install Inspectrum and Baudline
- We'll provide some RTL-SDRs

DEF CON 25 “Radio Exploitation 101”

How to attack and offensively utilize the techniques shown today

Video coming soon!

The IoT is full of holes...

marc@Bastille.net
@marcnewlin

matt@Bastille.net
@embeddedsec

It’s up to you to find them!

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