Lazy-Mode RF OSINT and Reverse Engineering

Marc Newlin | @marcnewlin | TROOPERS18

$(whoami)

- Red Team @ Snap
- Former Wireless Security Researcher @ Bastille Networks
- Wireless CVE’s in products from 21 vendors

MouseJack

KeySniffer

I am lazy and you can too

- Radios
- They aren’t as scary as they might seem
- How to maximize laziness when hacking them
- Making OSINT a little easier

Related talks

So You Want To Hack Radios @ Troopers18

- *Matt Knight and Marc Newlin*
- https://www.youtube.com/watch?v=OFRwqpH9zAQ

Radio Exploitation 101 @ HITB GSEC

- *Matt Knight and Marc Newlin*
- https://www.youtube.com/watch?v=UrVbN23zR9c

What is a radio?

- Magic black-box
- Converts digital data into radio waves (TX)
- Converts radio waves into digital data (RX)
- Radios **can** be analog, but we only really care about digital radios

[Hardware|Software] Defined Radio

Hardware Defined Radio

- Purpose-built to speak a specific protocol
- Usually can’t deviate [much] from the standard
- Logic is baked into silicon
- Easier to use than SDR
- Usually cheaper than SDR

Software Defined Radio

- Flexible radio front-end
- Raw RF samples get sent to the host computer
- Highly reconfigurable
- Protocol logic is implemented in software
- Can get expensive
- More domain knowledge required

How can we use radios?

Hardware Defined Radio

- Talk to devices using standardized protocols (WiFi, BT, etc)
- Talk to devices using proprietary protocols but common RFICs (wireless peripherals, etc)
- Talk to devices using undocumented protocols, after you've reverse engineered the protocol with an SDR, or gathered sufficient OSINT

Software Defined Radio

- Talk to devices using standardized protocols when an HDR isn't available (LoRa, ZigBee, etc)
- Perform PHY-layer attacks (jamming, replay, sniffing, etc)
- Reverse engineer undocumented protocols and devices

Be lazy, find vulns

1. Pick a target
2. Define your goals
3. Gather open-source intelligence
4. Acquire the right hardware/software tools
5. Find some vulns

Pick a target

What are “easier” targets?

- Low power devices designed to work for a long time on a single battery/charge
  - low power == low complexity == [maybe] low security
- Inexpensive devices from lesser-known vendors
  - cheap components means simple RF PHY and [maybe] no encryption
- Devices using COTS RFICs
  - usually means good documentation about the RFICs

What are “harder” targets?

- Devices with no compatible (and accessible) HDR
- Devices that exceed the capabilities of your SDR
  - bandwidth
  - frequency
  - retune time
  - ADC resolution
- Devices with little or no OSINT findings
  - blind reversing requires a significant effort

Devices are built under constraints

- Component cost
- Engineering cost
- Desired features
- Power consumption
- People are more likely to use off the shelf RFICs than roll their own
- Application layer SDKs cut down on software/firmware engineering costs

Target 1: Garage Door Opener

Keyscan TR4

- Garage door opener
- Low power
- Long use on single battery

Target 2: Wireless Barcode Scanner

Netum NT-1698W

- 2.4GHz wireless barcode scanner
- Inexpensive (~$30 USD)
- Lesser-known vendor

Define your goals

Garage Door Opener Goals

- Open the garage door (without the given opener)

Wireless Barcode Scanner Goals

- Determine if the barcode scanner is functionally a keyboard
- Perform a keystroke injection attack

Gather OSINT

What do we actually need to learn about a device?

What do we actually need to learn about a device?

It depends on what your goals are

What do we actually need to learn about a device?

It depends on what your goals are

- For a simple replay attack, you might only need to know the frequency.

What do we actually need to learn about a device?

It depends on what your goals are

- For a simple replay attack, you might only need to know the frequency.
- For a sniffing attack, you might need to to understand the MAC layer.

What do we actually need to learn about a device?

It depends on what your goals are

- For a simple replay attack, you might only need to know the frequency.
- For a sniffing attack, you might need to to understand the MAC layer.
- If it uses an off-the-shelf RFIC, you likely won't need to understand all the details of the PHY (and maybe not the MAC either).

What do we actually need to learn about a device?

It depends on what your goals are

- For a simple replay attack, you might only need to know the frequency.
- For a sniffing attack, you might need to to understand the MAC layer.
- If it uses an off-the-shelf RFIC, you likely won't need to understand all the details of the PHY (and maybe not the MAC either).
- If it uses an unknown RFIC, you'll probably need to reverse engineer the PHY.

What are some good sources for RF OSINT?

- Regulatory filings (FCC)
- RFIC datasheets
- Standards documents
- Prior reverse-engineering work
- Marketing material

Federal Communications Commission (FCC)

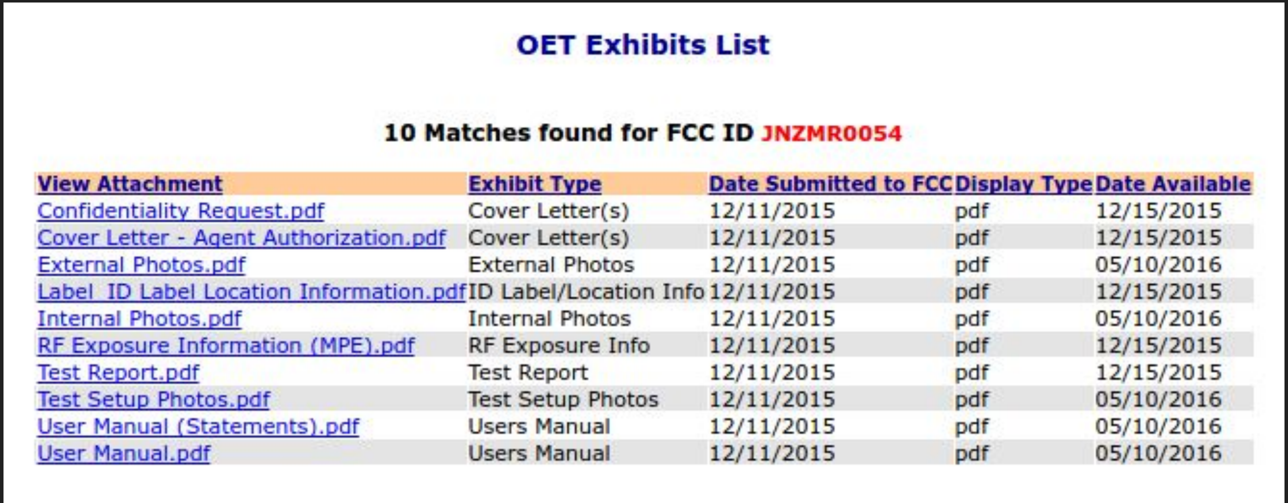
- US regulatory body governing electromagnetic spectrum usage
- Usually relevant to non-US markets and devices
  - Vendors often use a single test lab to certify a device for multiple markets
  - FCC publishes verbose device RF information

FCC Certification Process

1. Device is manufactured
2. Test lab evaluates the device
3. Telecommunications certification body issues a grant of certification
4. Test report, application, and related exhibits published in FCC database
5. Some exhibits are confidential (temporarily or permanently)

Finding FCC Exhibits

- Lookup FCC ID @ https://www.fcc.gov/general/fcc-id-search-page
- Click on the ‘Detail’ link on the results page



OET Exhibits List

10 Matches found for FCC ID JNZMR0054

View Attachment	Exhibit Type	Date Submitted to FCC	Display Type	Date Available
Confidentiality Request.pdf	Cover Letter(s)	12/11/2015	pdf	12/15/2015
Cover Letter - Agent Authorization.pdf	Cover Letter(s)	12/11/2015	pdf	12/15/2015
External Photos.pdf	External Photos	12/11/2015	pdf	05/10/2016
Label ID Label Location Information.pdf	ID Label/Location Info	12/11/2015	pdf	12/15/2015
Internal Photos.pdf	Internal Photos	12/11/2015	pdf	05/10/2016
RF Exposure Information (MPE).pdf	RF Exposure Info	12/11/2015	pdf	12/15/2015
Test Report.pdf	Test Report	12/11/2015	pdf	12/15/2015
Test Setup Photos.pdf	Test Setup Photos	12/11/2015	pdf	05/10/2016
User Manual (Statements).pdf	Users Manual	12/11/2015	pdf	05/10/2016
User Manual.pdf	Users Manual	12/11/2015	pdf	05/10/2016

FCC Documentation

- Applications
- Test Reports
- Internal / External Photos
- User Manuals
- Schematics / Block Diagrams
- Operational Descriptions

FCC Application

- Frequency
- Transmit power
- Type of device (i.e. car key fob)
- Vendor information
- Test lab information

FCC Test Reports

- Does the device meet FCC guidelines?
    - Transmit power
    - Bandwidth
    - Frequencies
    - Duty cycle
- 2498 authorized test labs
- Each lab has one or more report formats
- Each lab provides a varying degree of detail

FCC Internal / External Photos

- Internal / external photos of a device
- Typically taken by the test lab
- No standardization means [potentially] questionable quality
    - Low-resolution images
    - Blurred images
    - Blacked-out chip markings

FCC Schematics

- Most vendors request permanent confidentiality on schematics
- More common with lesser known manufacturers
- When available, extremely useful to learn RFIC specifics

FCC Operational Descriptions and User Manuals

- Describes the device behavior in an undefined format
- Hit or miss, but potentially fruitful
- Some vendors include useful technical details

RFIC Datasheets

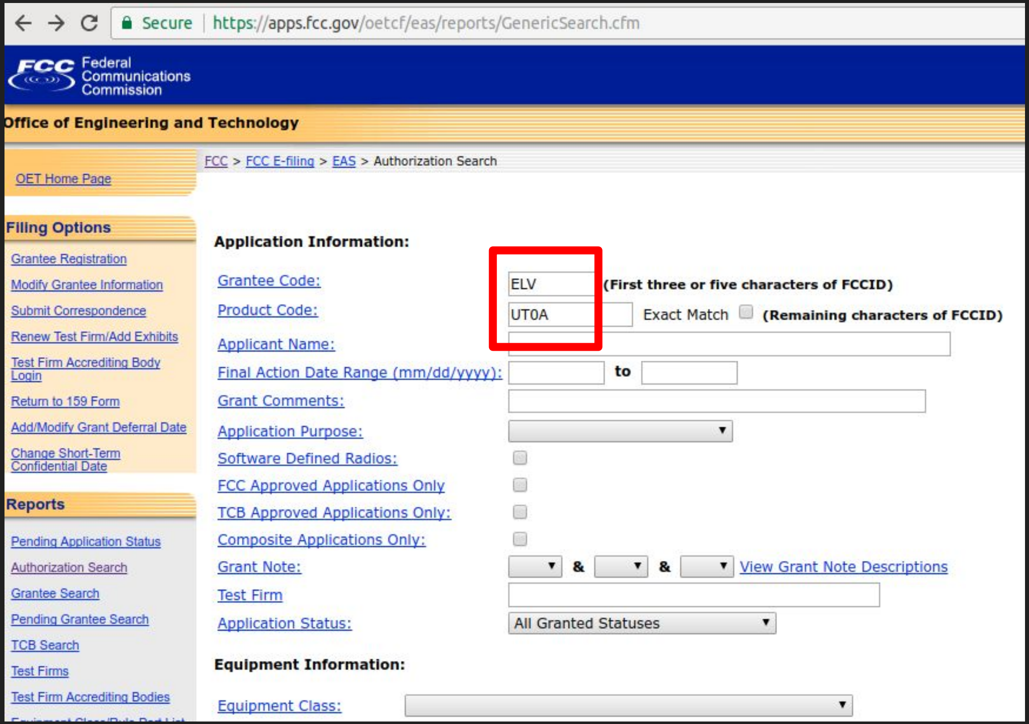
- It’s much easier to use an existing RFIC than to roll your own
- The engineers who build the <wireless device> needed documentation of the RFIC(s) they used
- What documentation did they use?
- Are there existing open-source implementations of the PHY/MAC?
- Is there an available HDR dongle/shield?

Prior reverse-engineering work

- Has somebody already solved this problem?
- Did they release documentation? Code?
- Is it permissively licensed?

Garage Door Opener - FCC Search

FCC ID - ELVUT0A



Secure
https://apps.fcc.gov/oetcf/eas/reports/GenericSearch.cfm
FCC Federal Communications Commission
Office of Engineering and Technology
FCC > FCC E-filing > EAS > Authorization Search
OET Home Page
Filing Options
Grantee Registration
Modify Grantee Information
Submit Correspondence
Renew Test Firm/Add Exhibits
Test Firm Accrediting Body Login
Return to 159 Form
Add/Modify Grant Deferral Date
Change Short-Term Confidential Date
Reports
Pending Application Status
Authorization Search
Grantee Search
Pending Grantee Search
TCB Search
Test Firms
Test Firm Accrediting Bodies

Application Information:

Grantee Code: ELV (First three or five characters of FCCID)

Product Code: UT0A Exact Match □ (Remaining characters of FCCID)

Applicant Name:

Final Action Date Range (mm/dd/yyyy): to

Grant Comments:

Application Purpose:

Software Defined Radios: □

FCC Approved Applications Only □

TCB Approved Applications Only: □

Composite Applications Only: □

Grant Note: & & View Grant Note Descriptions

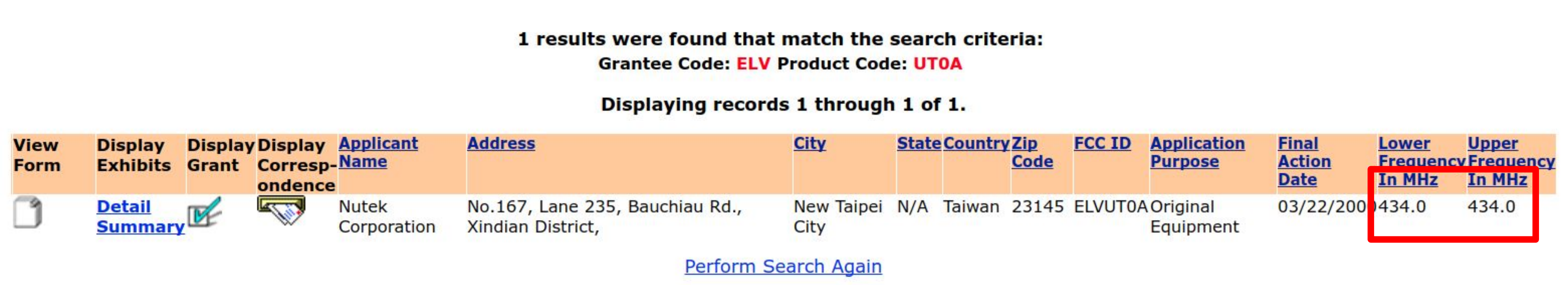
Test Firm

Application Status: All Granted Statuses

Equipment Information:

Equipment Class:

Garage Door Opener - FCC Search Results



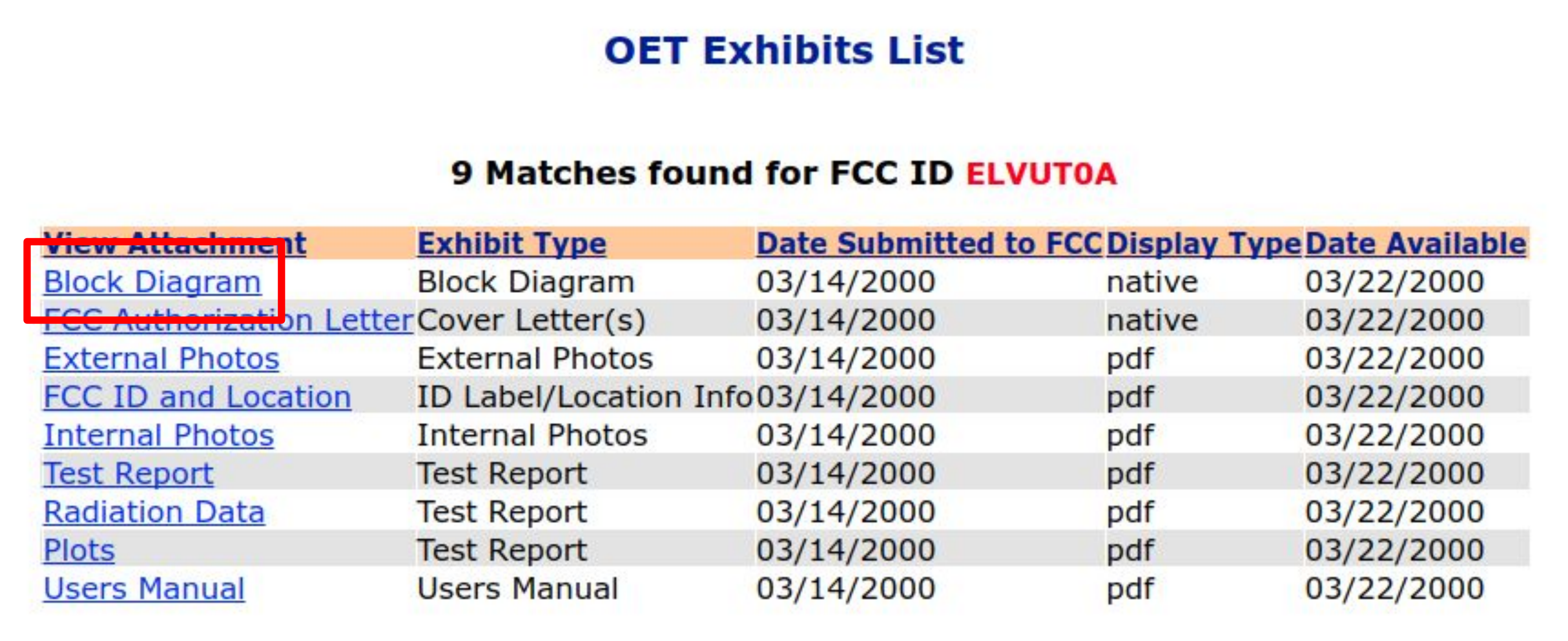
1 results were found that match the search criteria:
Grantee Code: ELV Product Code: UT0A

Displaying records 1 through 1 of 1.

View Form	Display Exhibits	Display Grant	Display Corresp-ondence	Applicant Name	Address	City	State	Country	Zip Code	FCC ID	Application Purpose	Final Action Date	Lower Frequency In MHz	Upper Frequency In MHz
	Detail Summary	✓		Nutek Corporation	No.167, Lane 235, Bauchiau Rd., Xindian District,	New Taipei City	N/A	Taiwan	23145	ELVUT0A	Original Equipment	03/22/200	434.0	434.0

Perform Search Again

Garage Door Opener - FCC Exhibits

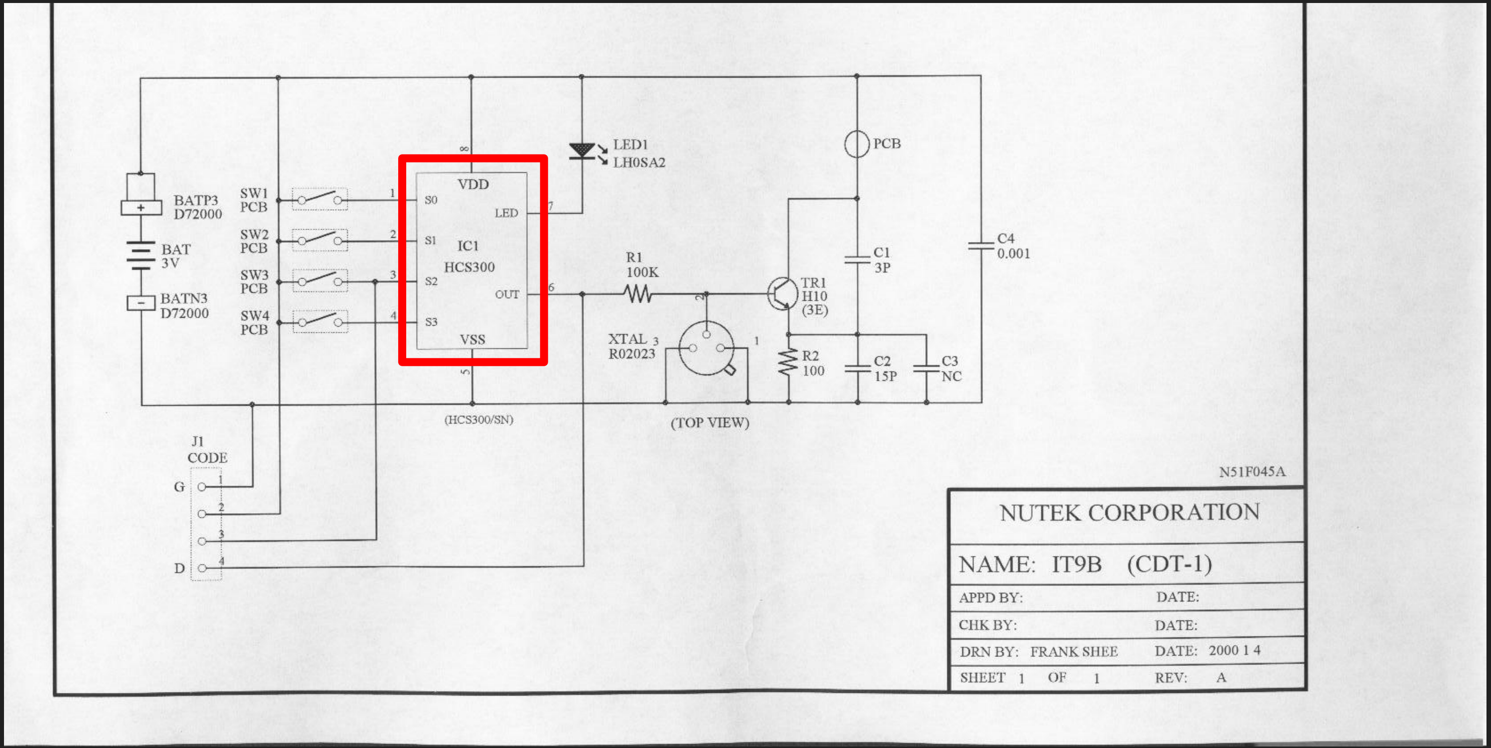


OET Exhibits List

9 Matches found for FCC ID ELVUT0A

View Attachment	Exhibit Type	Date Submitted to FCC	Display Type	Date Available
Block Diagram	Block Diagram	03/14/2000	native	03/22/2000
FCC Authorization Letter	Cover Letter(s)	03/14/2000	native	03/22/2000
External Photos	External Photos	03/14/2000	pdf	03/22/2000
FCC ID and Location	ID Label/Location Info	03/14/2000	pdf	03/22/2000
Internal Photos	Internal Photos	03/14/2000	pdf	03/22/2000
Test Report	Test Report	03/14/2000	pdf	03/22/2000
Radiation Data	Test Report	03/14/2000	pdf	03/22/2000
Plots	Test Report	03/14/2000	pdf	03/22/2000
Users Manual	Users Manual	03/14/2000	pdf	03/22/2000

Garage Door Opener - Block Diagram



NUTEK CORPORATION

NAME: IT9B (CDT-1)

APPD BY: DATE:

CHK BY: DATE:

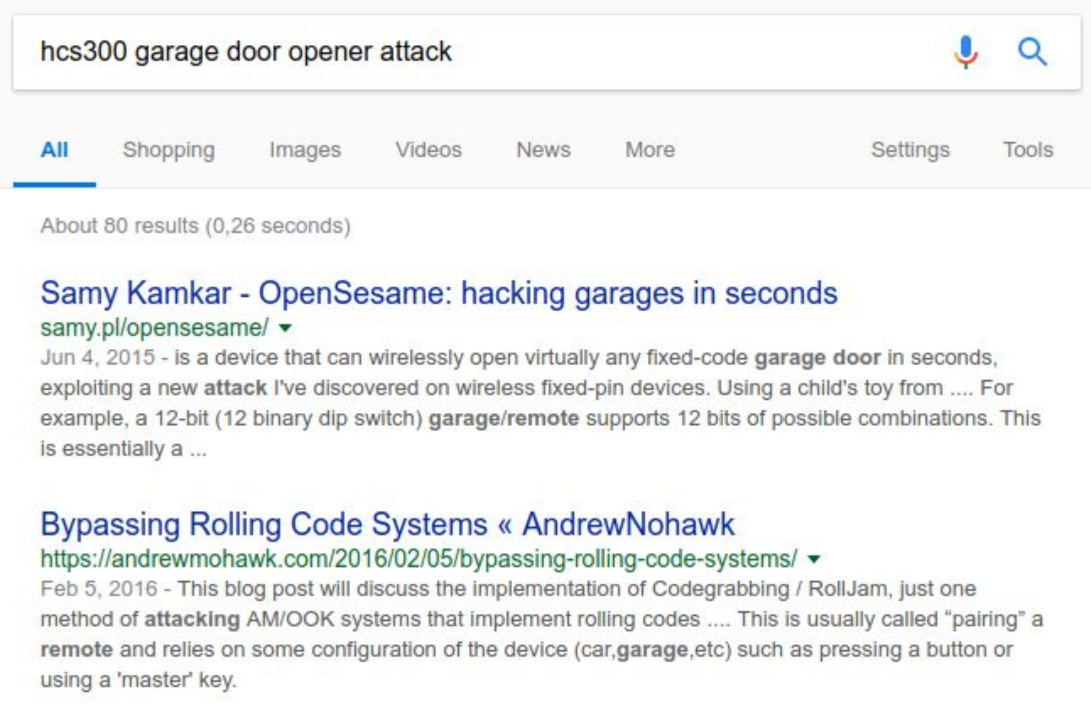
DRN BY: FRANK SHEE DATE: 2000 1 4

SHEET 1 OF 1 REV: A

Garage Door Opener - The Google

Solved problem, thanks to:

- @samykamkar
- @andrewmohawk
- Many others



hcs300 garage door opener attack

About 80 results (0,26 seconds)

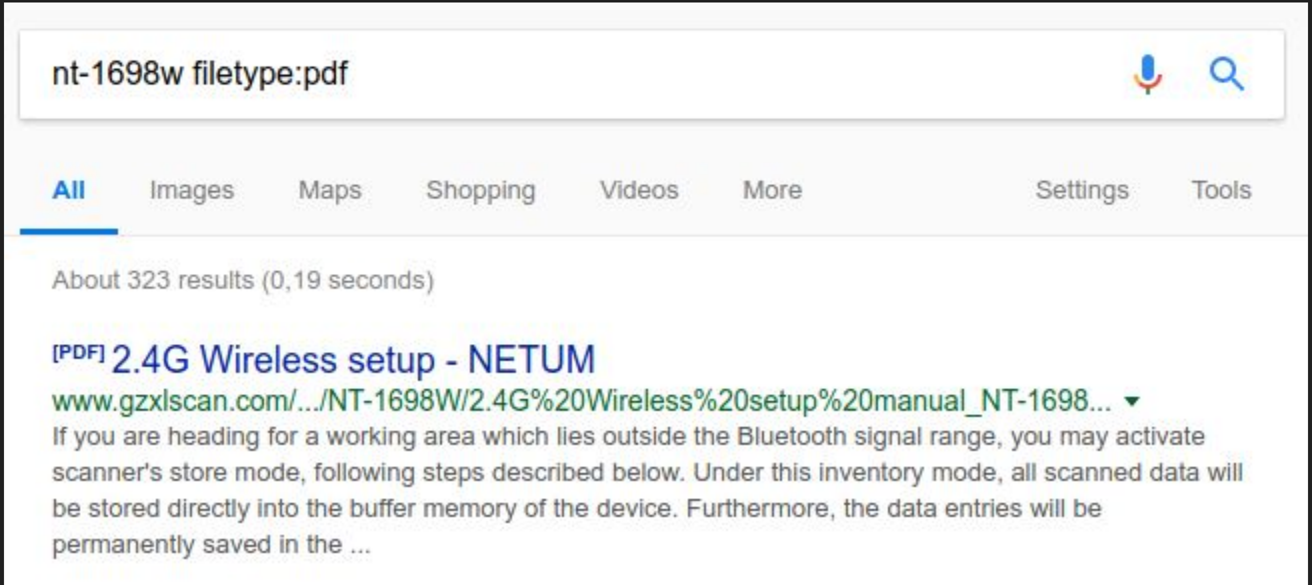
Samy Kamkar - OpenSesame: hacking garages in seconds
samy.pl/opensesame/
Jun 4, 2015 - is a device that can wirelessly open virtually any fixed-code **garage door** in seconds, exploiting a new **attack** I've discovered on wireless fixed-pin devices. Using a child's toy from .... For example, a 12-bit (12 binary dip switch) **garage/remote** supports 12 bits of possible combinations. This is essentially a ...

Bypassing Rolling Code Systems « AndrewNohawk
https://andrewmohawk.com/2016/02/05/bypassing-rolling-code-systems/
Feb 5, 2016 - This blog post will discuss the implementation of Codegrabbing / RollJam, just one method of **attacking** AM/OOK systems that implement rolling codes .... This is usually called "pairing" a **remote** and relies on some configuration of the device (car,**garage**,etc) such as pressing a button or using a 'master' key.

Wireless Barcode Scanner - FCC Search

- No FCC ID :(

Wireless Barcode Scanner - Google



nt-1698w filetype:pdf

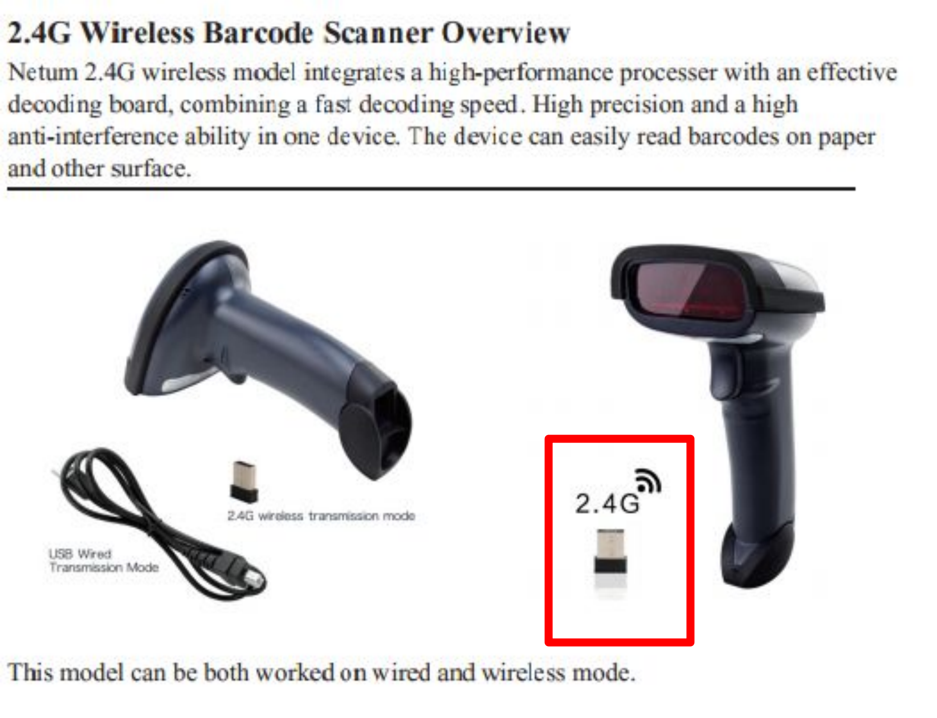
About 323 results (0,19 seconds)

[PDF] 2.4G Wireless setup - NETUM

www.gzxlscan.com/.../NT-1698W/2.4G%20Wireless%20setup%20manual_NT-1698...

If you are heading for a working area which lies outside the Bluetooth signal range, you may activate scanner's store mode, following steps described below. Under this inventory mode, all scanned data will be stored directly into the buffer memory of the device. Furthermore, the data entries will be permanently saved in the ...

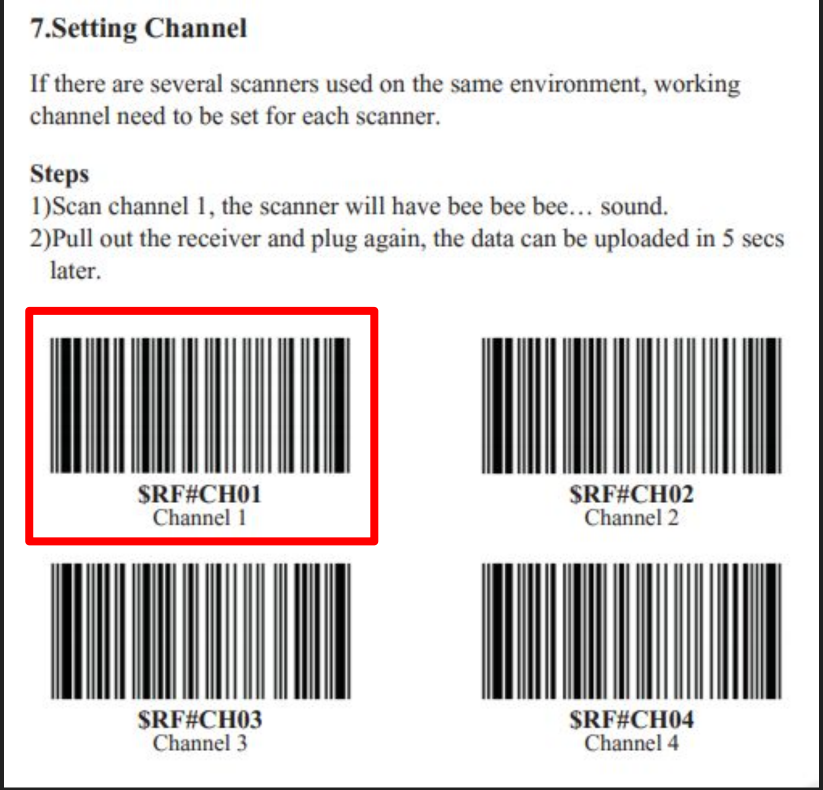
Wireless Barcode Scanner - User Manual



2.4G Wireless Barcode Scanner Overview

Netum 2.4G wireless model integrates a high-performance processer with an effective decoding board, combining a fast decoding speed. High precision and a high anti-interference ability in one device. The device can easily read barcodes on paper and other surface.

This model can be both worked on wired and wireless mode.



7.Setting Channel

If there are several scanners used on the same environment, working channel need to be set for each scanner.

Steps

1)Scan channel 1, the scanner will have bee bee bee… sound.

2)Pull out the receiver and plug again, the data can be uploaded in 5 secs later.

$RF#CH01
Channel 1

$RF#CH02
Channel 2

$RF#CH03
Channel 3

$RF#CH04
Channel 4

Use the right tools

SDR Hardware

(some reasonably-priced devices)

RTL-SDR

- Receive only
- ~20 MHz - 1800 MHz tuning range
- ~2.4 MHz maximum sample rate
- ~$20 USD

HackRF

- Transmit and Receive (half-duplex)
- 1 MHz - 6 GHz tuning range
- 20 MHz maximum sample rate
- ~$300 USD

bladeRF x40

- Transmit and Receive (full-duplex)
- 300 MHz - 3.8 GHz tuning range
- 40 MHz maximum sample rate
- ~$420 USD

PlutoSDR

- Transmit and Receive (full-duplex)
- 325 MHz - 3.8 GHz tuning range
- 20 MHz maximum sample rate
- ~$100 USD

Open-Source SDR Software

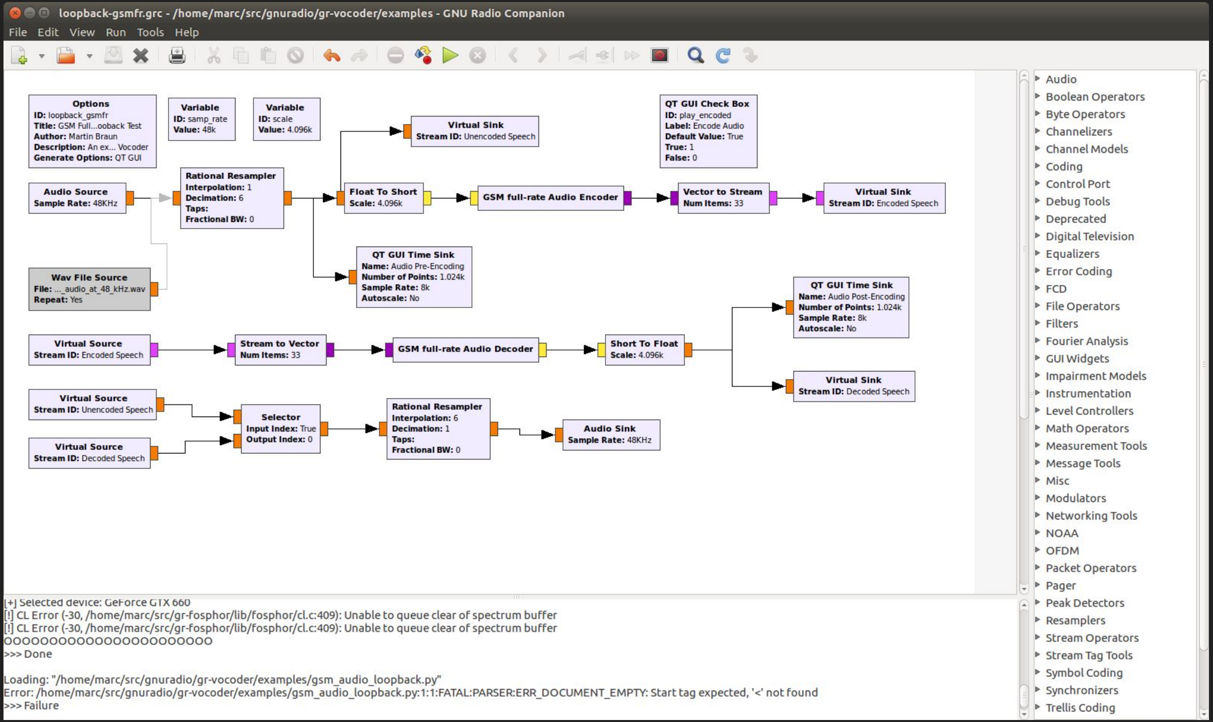
(a small slice of a big ecosystem)

GNU Radio

- Open source SDR toolkit written in C/C++ and Python
- Large selection of signal processing libraries
- Hardware support for common SDR platforms
- Efficient prototyping

GNU Radio Companion

- Drag and drop flow graph creator
- Quick and easy



Options
ID: loopback_gsmfr
Title: GSM Full...ooback Test
Author: Martin Braun
Description: An ex... Vocoder
Generate Options: QT GUI

Variable
ID: samp_rate
Value: 48k

Variable
ID: scale
Value: 4.096k

QT GUI Check Box
ID: play_encoded
Label: Encode Audio
Default Value: True
True: 1
False: 0

Audio Source
Sample Rate: 48KHz

Wav File Source
File: ..._audio_at_48_kHz.wav
Repeat: Yes

Rational Resampler
Interpolation: 1
Decimation: 6
Taps:
Fractional BW: 0

Virtual Sink
Stream ID: Unencoded Speech

Float To Short
Scale: 4.096k

GSM full-rate Audio Encoder

Vector to Stream
Num Items: 33

Virtual Sink
Stream ID: Encoded Speech

QT GUI Time Sink
Name: Audio Pre-Encoding
Number of Points: 1.024k
Sample Rate: 8k
Autoscale: No

Virtual Source
Stream ID: Encoded Speech

Stream to Vector
Num Items: 33

GSM full-rate Audio Decoder

Short To Float
Scale: 4.096k

QT GUI Time Sink
Name: Audio Post-Encoding
Number of Points: 1.024k
Sample Rate: 8k
Autoscale: No

Virtual Sink
Stream ID: Decoded Speech

Virtual Source
Stream ID: Unencoded Speech

Virtual Source
Stream ID: Decoded Speech

Selector
Input Index: True
Output Index: 0

Rational Resampler
Interpolation: 6
Decimation: 1
Taps:
Fractional BW: 0

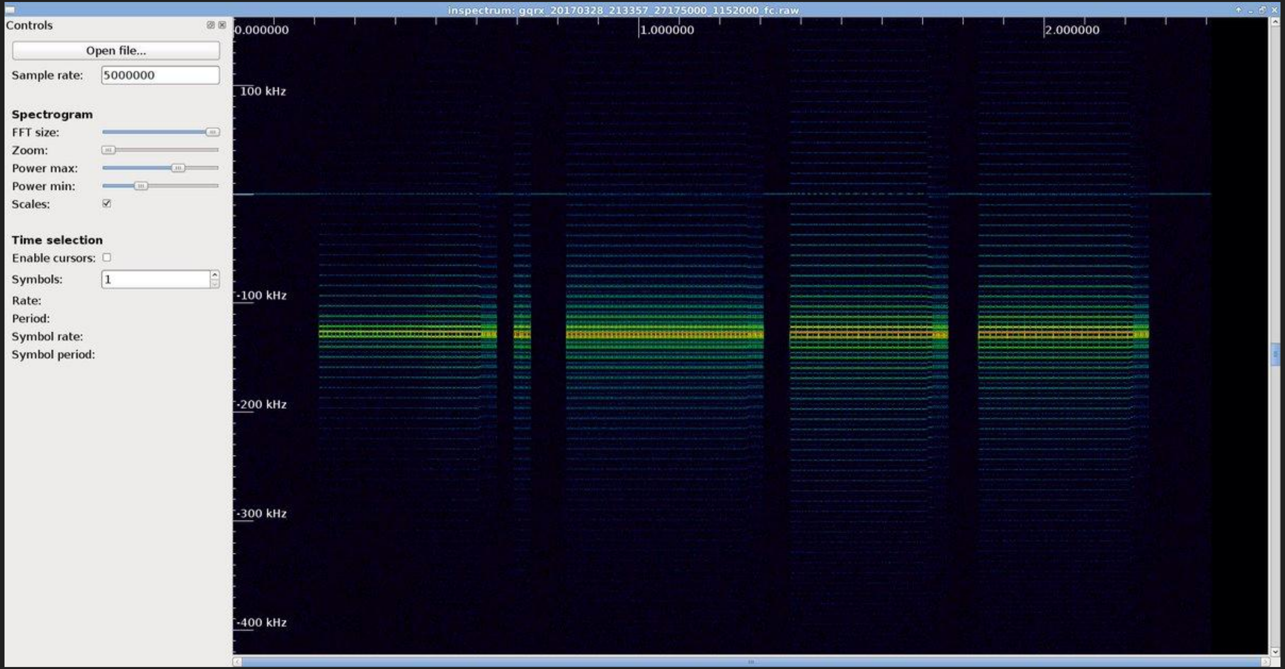
Audio Sink
Sample Rate: 48KHz

```
[+] Selected device: GeForce GTX 660
[!] CL Error (-30, /home/marc/src/gr-fosphor/lib/fosphor/cl.c:409): Unable to queue clear of spectrum buffer
[!] CL Error (-30, /home/marc/src/gr-fosphor/lib/fosphor/cl.c:409): Unable to queue clear of spectrum buffer
OOOOOOOOOOOOOOOOOOOOOOO
>>> Done

Loading: "/home/marc/src/gnuradio/gr-vocoder/examples/gsm_audio_loopback.py"
Error: /home/marc/src/gnuradio/gr-vocoder/examples/gsm_audio_loopback.py:1:1:FATAL:PARSER:ERR_DOCUMENT_EMPTY: Start tag expected, '<' not found
>>> Failure
```

Inspectrum

- Spectrum visualization and analysis tool



inspectrum: gqrx_20170328_213357_27175000_1152000_fc.raw

Controls

Sample rate: 5000000

Spectrogram

FFT size:

Zoom:

Power max:

Power min:

Scales: ☑

Time selection

Enable cursors: □

Symbols: 1

Rate:

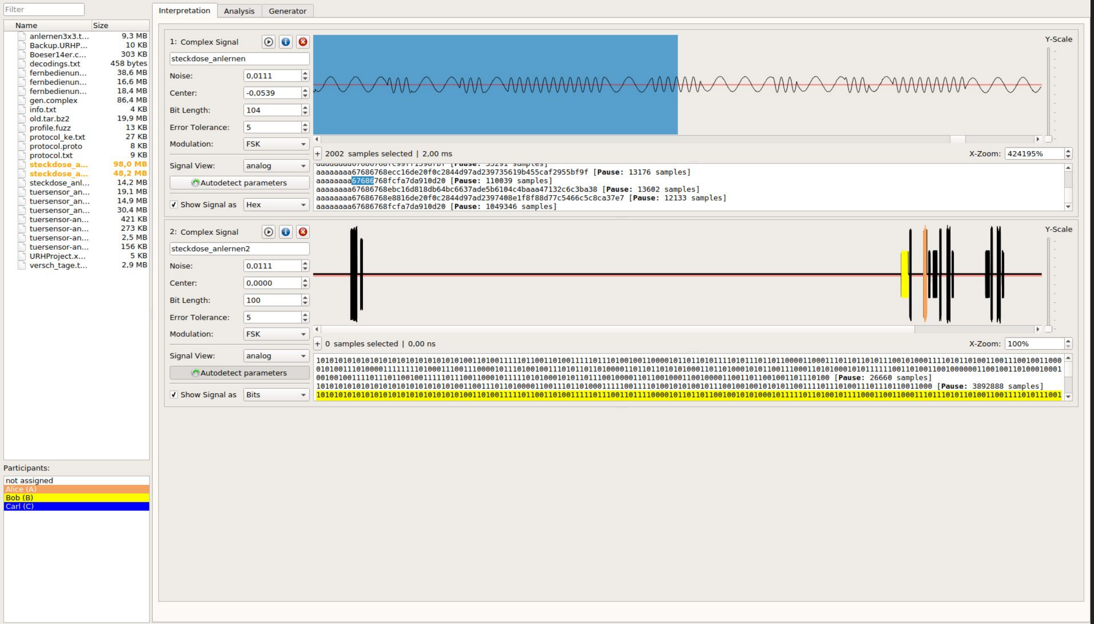
Period:

Symbol rate:

Symbol period:

Universal Radio Hacker

- [Semi] automatic signal / protocol reversing tool



Name	Size
anlernen3x3.t...	9,3 MB
Backup.URHP...	10 KB
Boeser14er.c...	303 KB
decodings.txt	458 bytes
fernbedienun...	38,6 MB
fernbedienun...	16,6 MB
fernbedienun...	18,4 MB
gen.complex	86,4 MB
info.txt	4 KB
old.tar.bz2	19,9 MB
profile.fuzz	13 KB
protocol_ke.txt	27 KB
protocol.proto	8 KB
protocol.txt	9 KB
steckdose_a...	98,0 MB
steckdose_a...	48,2 MB
steckdose_anl...	14,2 MB
tuersensor_an...	19,1 MB
tuersensor_an...	14,9 MB
tuersensor_an...	30,4 MB
tuersensor-an...	421 KB
tuersensor-an...	273 KB
tuersensor-an...	2,5 MB
tuersensor-an...	156 KB
URHProject.x...	5 KB
versch_tage.t...	2,9 MB

not assigned
Alice (A)
Bob (B)
Carl (C)

1: Complex Signal

steckdose_anlernen

Noise: 0,0111
Center: -0,0539
Bit Length: 104
Error Tolerance: 5
Modulation: FSK
Signal View: analog
☑ Show Signal as Hex

2002 samples selected | 2,00 ms
X-Zoom: 424195%

```
aaaaaaaa67686768ecc16de20f0c2844d97ad239735619b455caf2955bf9f [Pause: 13176 samples]
aaaaaaaa67686768fcfa7da910d20 [Pause: 110039 samples]
aaaaaaaa67686768ebc16d818db64bc6637ade5b6104c4baaa47132c6c3ba38 [Pause: 13602 samples]
aaaaaaaa67686768e8816de20f0c2844d97ad2397408e1f8f88d77c5466c5c8ca37e7 [Pause: 12133 samples]
aaaaaaaa67686768fcfa7da910d20 [Pause: 1049346 samples]
```

2: Complex Signal

steckdose_anlernen2

Noise: 0,0111
Center: 0,0000
Bit Length: 100
Error Tolerance: 5
Modulation: FSK
Signal View: analog
☑ Show Signal as Bits

0 samples selected | 0,00 ns
X-Zoom: 100%

```
1010101010101010101010101010101010100110100111110110011010011111011101001001100001011011010111101011101101100001100011101101101011100101000111101011010011001110010011000
01010011101000011111111010001110011100001011101001001110101101101000011011011010101000110110100010101100111000110101000101011111100110100110010000001100100110100010001
0010010011110111011001001111101110011000101111101010001010110111001000011011001000110010000110011011001001101110100 [Pause: 26660 samples]
10101010101010101010101010101010101001100111011010000110011101101000111110011111010010101001011100100100101010110011110111010011101110110011000 [Pause: 3892888 samples]
101010101010101010101010101010101010011010011111011001101001111101110011011110000101101101100100101010001011111011010010111100011001100011101110101101001100111101011 1001
```

Some of my favorite HDR tools

CrazyRadio PA USB Dongle	2.4GHz GFSK
Logitech C-U0007 USB Dongle	2.4GHz GFSK
ADF7242 PMOD/SPI Module	2.4GHz GFSK/OOK, 802.15.4
Ubertooth USB Dongle	Bluetooth
ApiMote	802.15.4
YARD Stick One	Sub-1GHz FSK/OOK

Garage Door Opener - Tools / Next Steps

- YARD Stick One
- @samykamkar’s OpenSesame code
- @andrewmohawk’s RfCat scripts and guide
- Stand on the shoulders of giants, be lazy, and open the garage door

Wireless Barcode Scanner - Tools / Next Steps

- 2.4GHz-capable SDR + Inspectrum
  - Identify the four RF channels used by the barcode scanner
- 2.4GHz-capable SDR + Universal Radio Hacker
  - Auto-magically reverse engineer the packet format
  - Generate and transmit injection packets

The FCC website isn’t perfect

- It’s designed as a document retrieval system, not a search engine
- It can be cumbersome to navigate, especially on mobile
- It’s often bogged down and slow

How can we make this easier?

<copy>

FCC equipment authorization database

<copy>	<paste>
FCC equipment authorization database	Elasticsearch

Introducing kitten.dog

- Yes, kitten.dog, because new TLDs are awesome
- DNS is propagating, so you may need to go to www.kitten.dog or kitten-dog.appspot.com

Questions?

Marc Newlin | @marcnewlin