

# Electronic Access Control Security

Matteo Beccaro || HackInTheBox  
Amsterdam, May 27<sup>th</sup>, 2016



OPPOSING FORCE

- Matteo Beccaro
- Founder & Chief Technology Officer at Opposing Force
  - The first Italian company specialize in offensive physical security
- Twitter: [@\\_bughardy\\_](#) | [@\\_opposingforce](#)

# What do you need? ||

## Extract the zip

### What you will find in the archive:

- VM with all tools and libraries for the hands-on parts
- VirtualBox installer
- VirtualBox guest-addition

username: opposingforce

password: opfor2016

- **Module 1 – Introduction**
  - Historical introduction on access control attacks
- **Module 2 – Attacking NFC**
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies

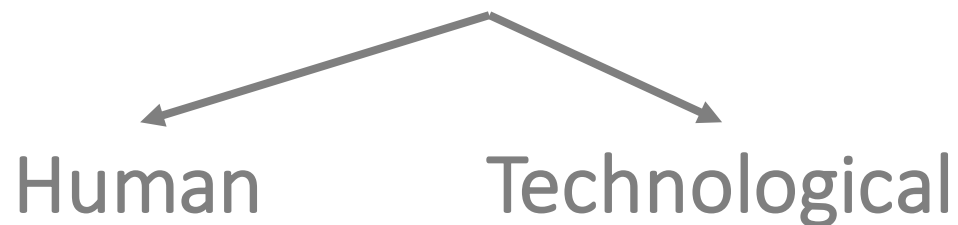
- **Module 3 – Attacking RF communications**
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security
- **Module 4 – The challenge**
  - Introducing the challenge
  - The awards 😊

# Module 1 | | introduction

- Access Control system?

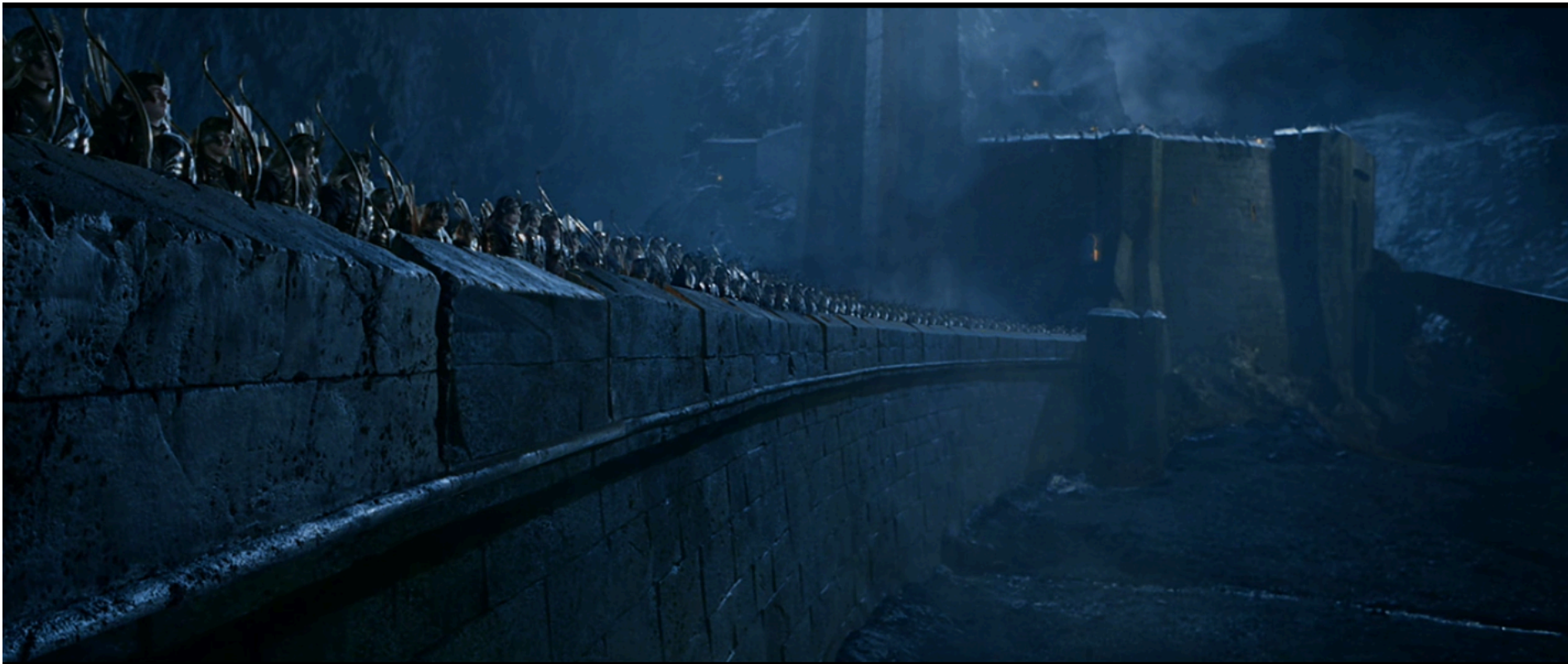
A system composed by several elements which aim is to limit the access to certain resources only to authorized people.

The system is composed by two type of elements:



- What was an Access Control system?

The **technological** elements





- What was an Access Control system?

The **human** elements...

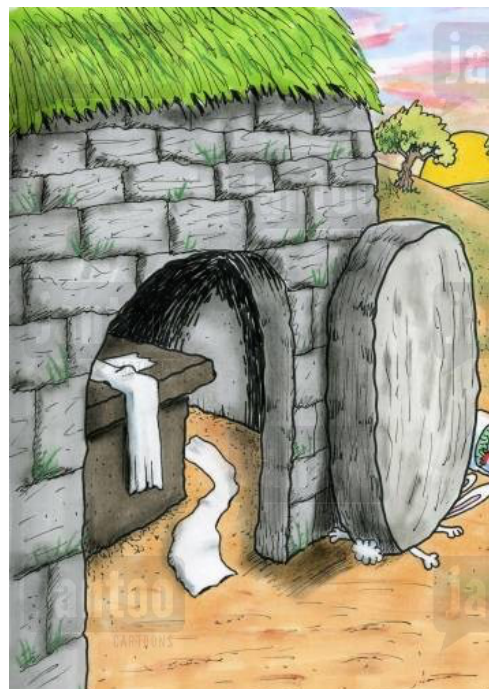


- What was an Access Control system?  
...often fail



- First access control hackers?

Magicians..



- First access control hackers?

## Social Engineers



- What is an Access Control system?



# What is an Electronic Access Control system? | |

- It may employ different technologies
  - NFC
  - RF
  - Biometrics
  - Mag-stripe
  - Mobile phones
  - etc.

# Module 2 || attacking NFC

- Module 2 – Attacking NFC
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies



# What is NFC? ||

- NFC stands for Near Field Communication
- Frequency at 13.56 MHz
- 3-5 cm of range
- Widely used for
  - Access control systems
  - Electronic ticketing systems
  - Mobile phone applications

# Notorious NFC families | |

- MIFARE
  - MIFARE Classic
  - MIFARE Ultralight
  - MIFARE DesFire
- HID iClass
- Calypso
- FeliCa

- 1-4 KB memory storage device
- ~~Strong~~ access control mechanisms
  - A key is required to access data sectors
  - Use of ~~Crypto1~~ **Crapto1** algorithm
  - Sadly broken..
  - ..but still so widely used (!) – RFID door tokens, transport tickets, etc.

- 64 byte memory storage device
- Basic security mechanisms
  - OTP (One-Time-Programmable) sector
  - Lock bytes sector
  - Mostly used for disposable tickets
  - It has some more secure children:
    - ULTRALIGHT C
    - ULTRALIGHT EV

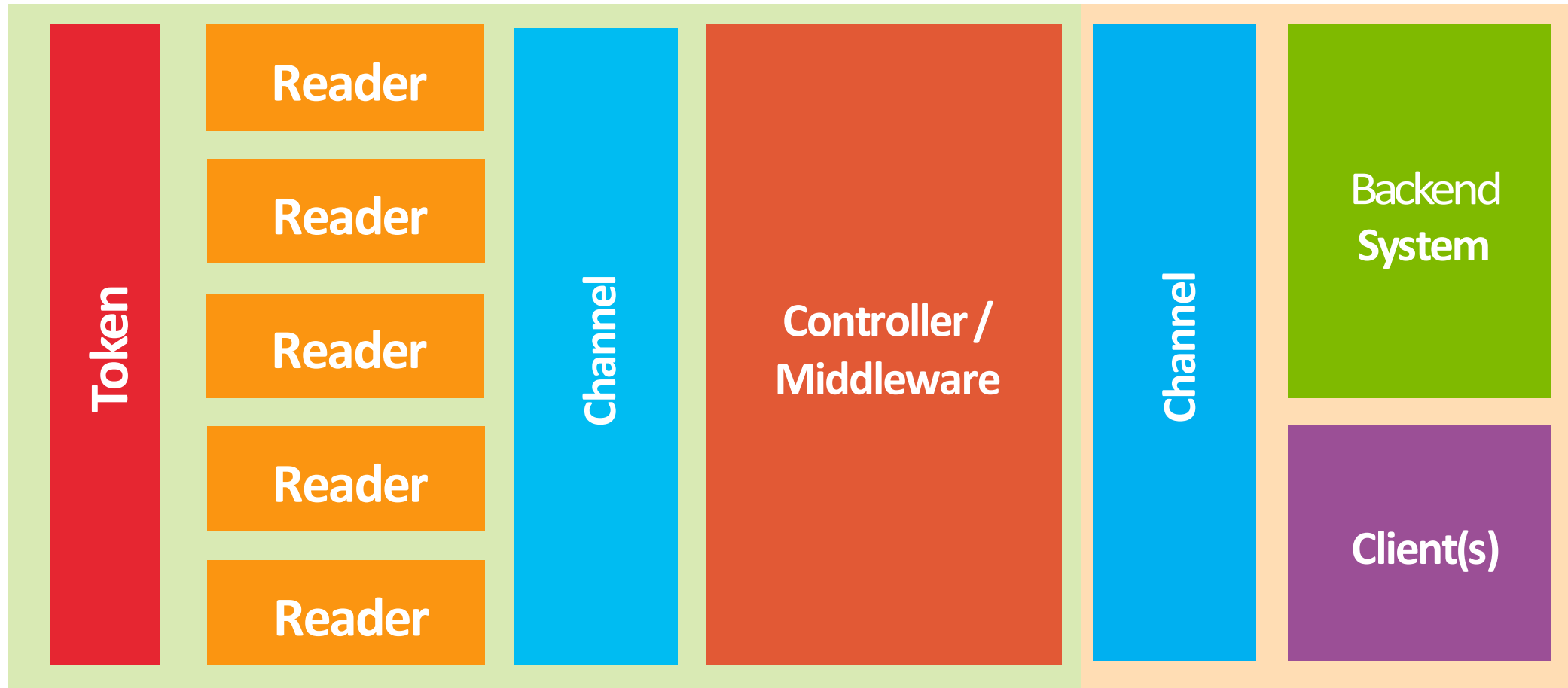
- 2 KB, 4KB or 8 KB memory size
- Advanced security mechanisms (3DES, AES, etc.)
- File system structure is supported
- Several variants are available
  - DESFIRE
  - DESFIRE EV1
  - DESFIRE EV2

- Same encryption and authentication keys are shared across every HID iClass Standard Security installations (!)
- Keys have already been extracted (!!)
- Two variants
  - iClass Standard (very common)
  - iClass High Secure (not that common)
- **Both variants are BROKEN**

# NFC-based Electronic Access Control systems | |

- We need to create a common **methodology**
- We need **tools** to effectively assess these systems
- We need **secure architectures** as references and best practices

# NFC-based Electronic Access Control systems | |





## The token ||

- Usually a NFC card
  - MIFARE Ultralight
  - MIFARE Classic
  - HID
- The card can store
  - Timestamp of the last stamping
  - Details on the location where we used the token
  - Credentials, access level, etc.



## The token ||

- What about MIFACE Classic?
  - It is just BROKEN
- What about MIFARE Ultralight?
  - Well, it's bleeding..
    - Lock attack
    - Time attack
    - Reply attack..
- HID
  - BROKEN, again

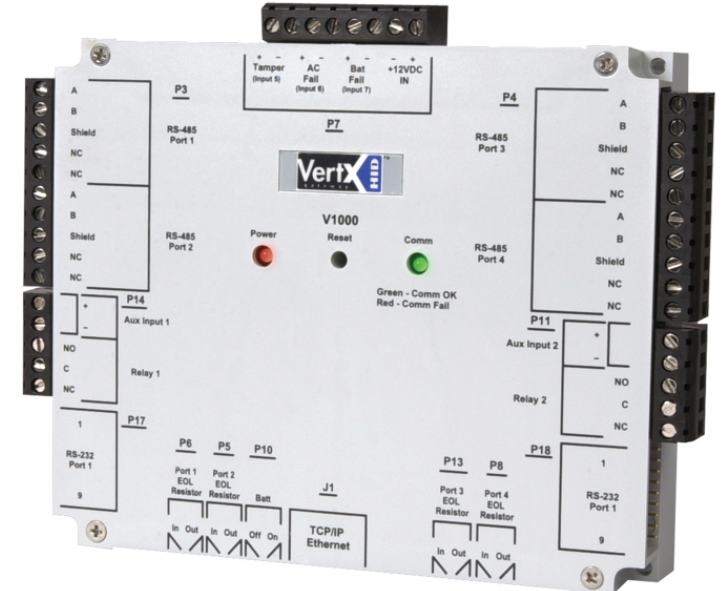


- Can operate offline or online
- Wire or wireless connected to the controller
  - RS232, Ethernet, etc.
- Usually supports multiple standards
- Can store secrets and keys used for authentication
- Usually it can
  - Read token(s) data
  - Send token data to the controller
  - Give a feedback to users on operation's success



# Controller | |

- Connected both to readers and backend
  - Wiegand, Ethernet, rs232
- Receives data from the reader(s)
  - Support multiple readers technologies
- Sends the data to the backend
  - Open the door
  - Deny the access



# The backend ||

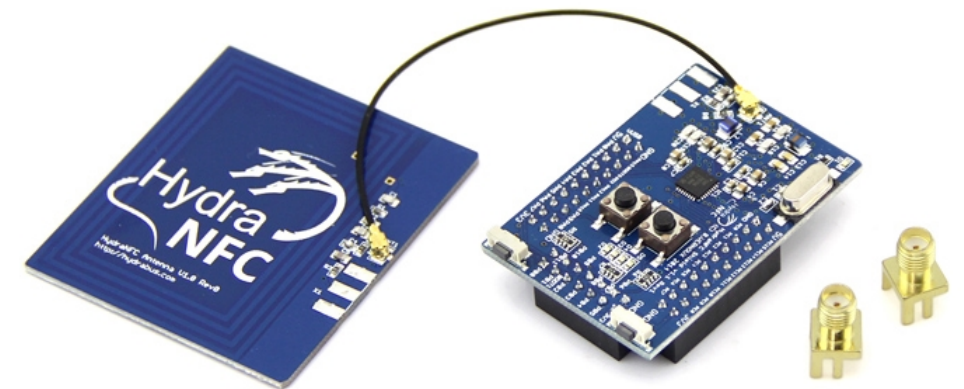
- It can be cloud-based or not
- Usually wired connected
  - RS232, Ethernet, etc.
- Performs multiple operations
  - Provide token validation “logic”
  - Statistics
  - Logging



- Module 2 – attacking NFC
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies

- HydraNFC
- ProxMark3
- ChameleonMini
- NFCuIT

- HydraNFC (~90 €)
  - <http://hydrabus.com/hydranfc-1-0-specifications/>
- Users Texas Instrument TRF7970A NFC chipset (**13.56MHz only**)
- MIFARE 1k and 14443A UID emulation
- ISO 14443A sniffing (also autonomous mode)
- 2 different raw modes





# ProxMark3 ||

- ProxMark3 (~200 €)
- HF and LF capabilities
- Very large community
  - <http://proxmark.org/forum/index.php>
- Supports almost every known RFID tags
- Support sniffing and emulation



## ChameleonMini ||

- ChameleonMini (~100 €)
  - <http://kasper-oswald.de/gb/chameleonmini/>
- HF (13.56MHz) only
- Almost same capabilities as HydraNFC
- Different chipset
- The firmware is only available for old revision



## Opposing Force own weapon ||

- NFCuIT (~0 €)
- Originally designed for ticketing systems, it can be also used for generic EAC system security assessment
- Mobile app for NFC-enabled Android smartphones
  - Implements Lock, Time and Reply attacks
- A “custom edit mode” is available for bit by bit data editing
- The app currently supports the MIFARE Ultralight format only
  - MIFARE Classic support will be released on summer 2016

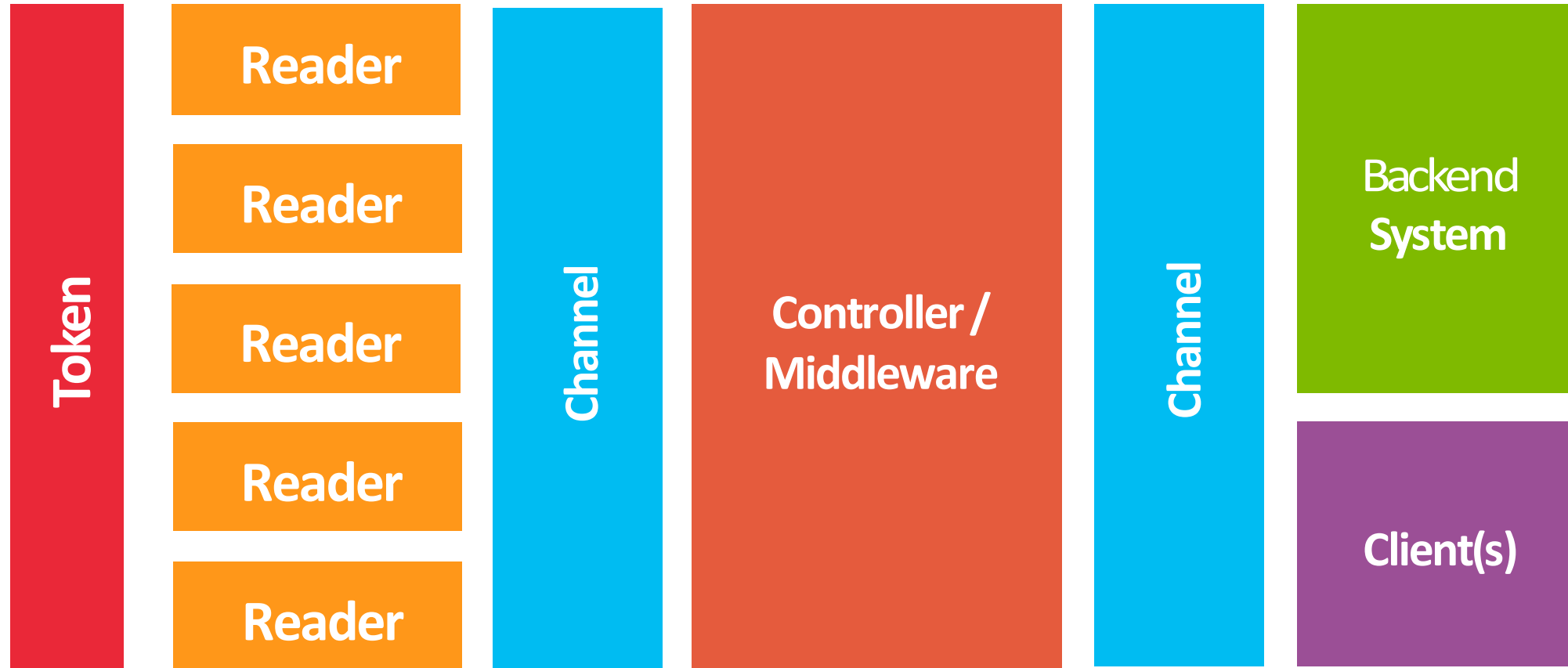
# The custom editing feature | |

- The features is useful to better understand the structure of data stored onto the token
- Quick encoding from hex to bin and back
- The app allows token bit by bit data editing

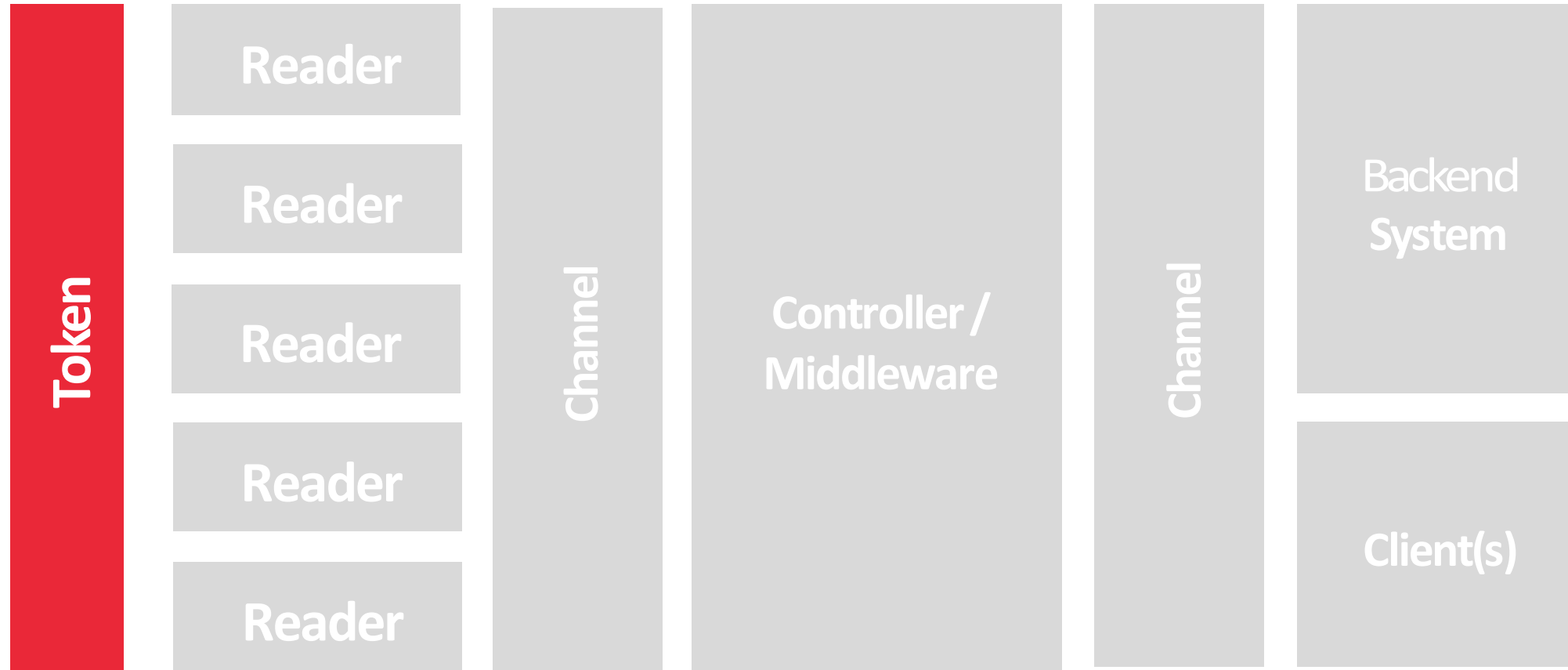


- Module 2 – Attacking NFC
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies

# Access Control system attack surface ||



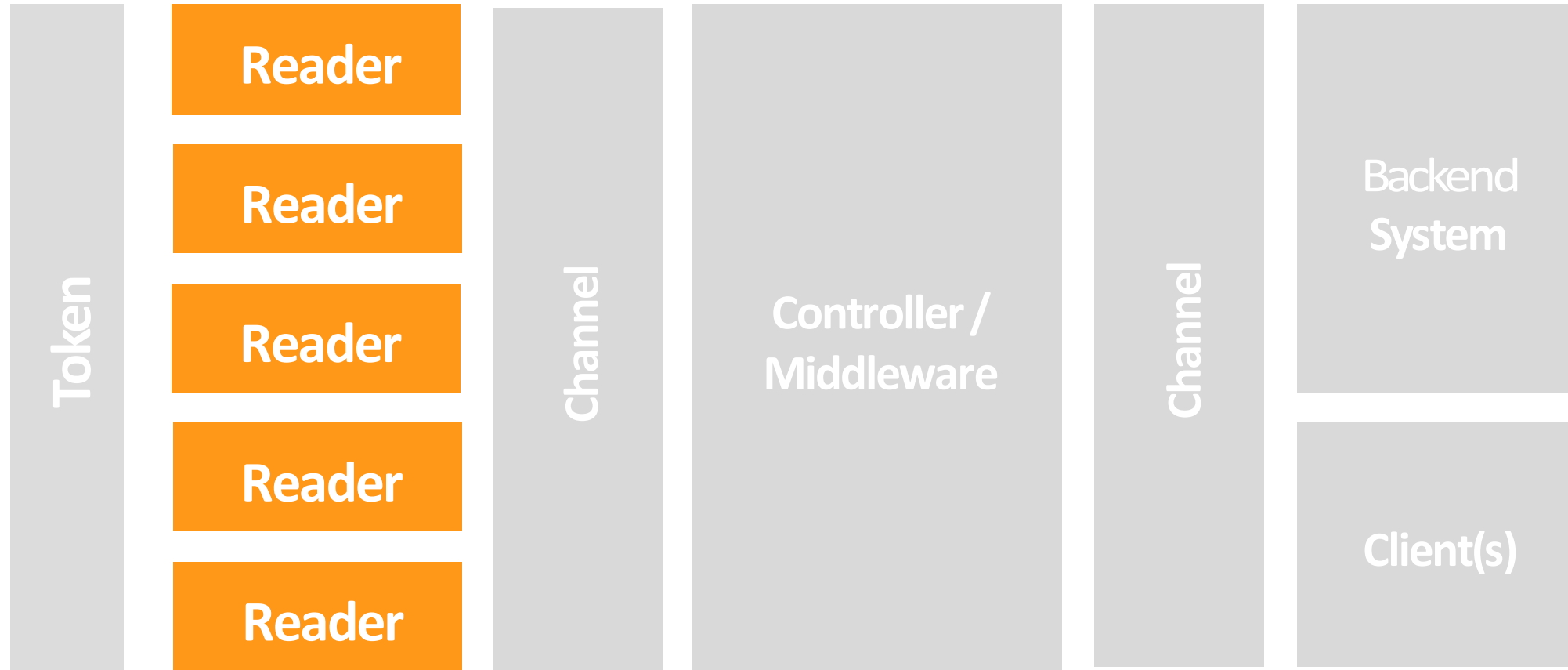
# Access Control system attack surface ||



<b>Attack Surface</b>	<b>Attacks to Perform</b>	<b>Impact</b>
NFC Interface	Analyze the authentication mechanisms	Secrets extraction, MiTM attacks
Hardware board	Side channel attacks	Secrets dumping or guessing
Memory	Assess logic vulnerabilities in the implementation	Bypass security mechanisms

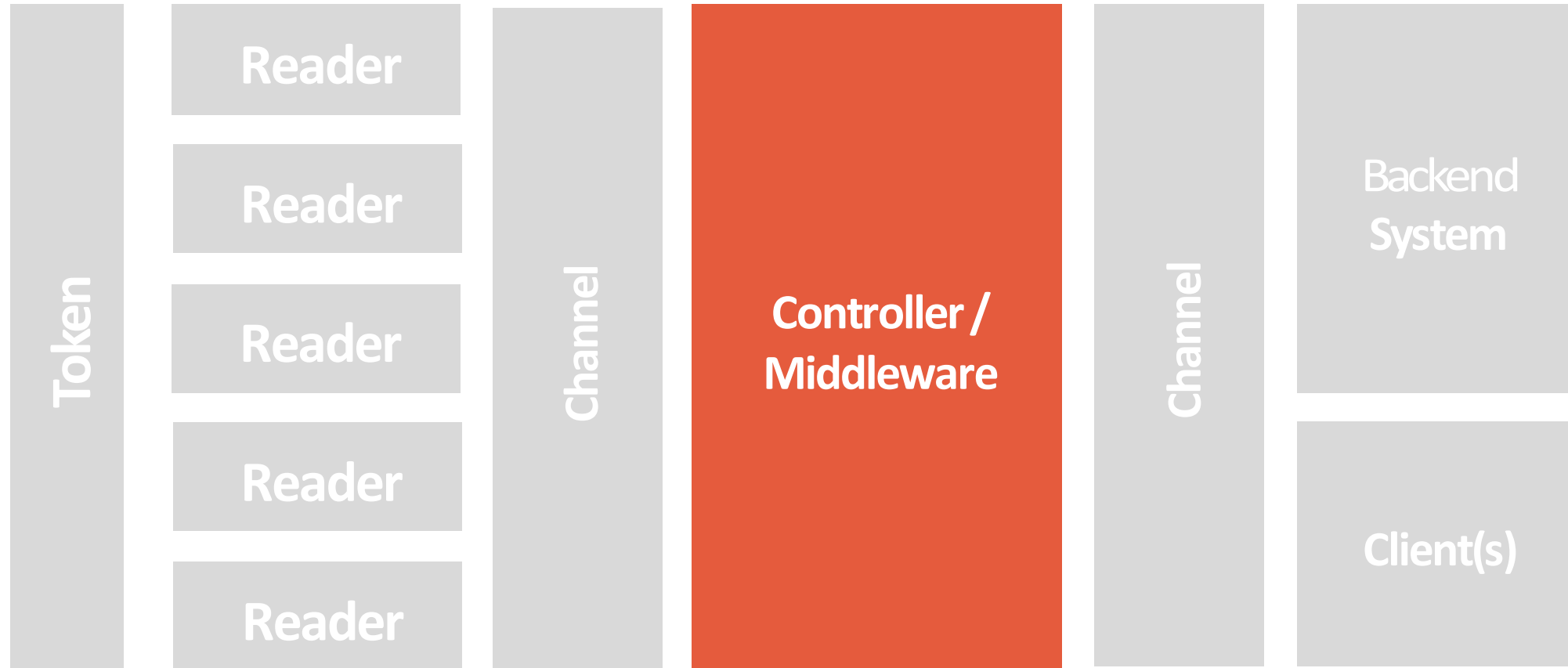


# Access Control system attack surface ||



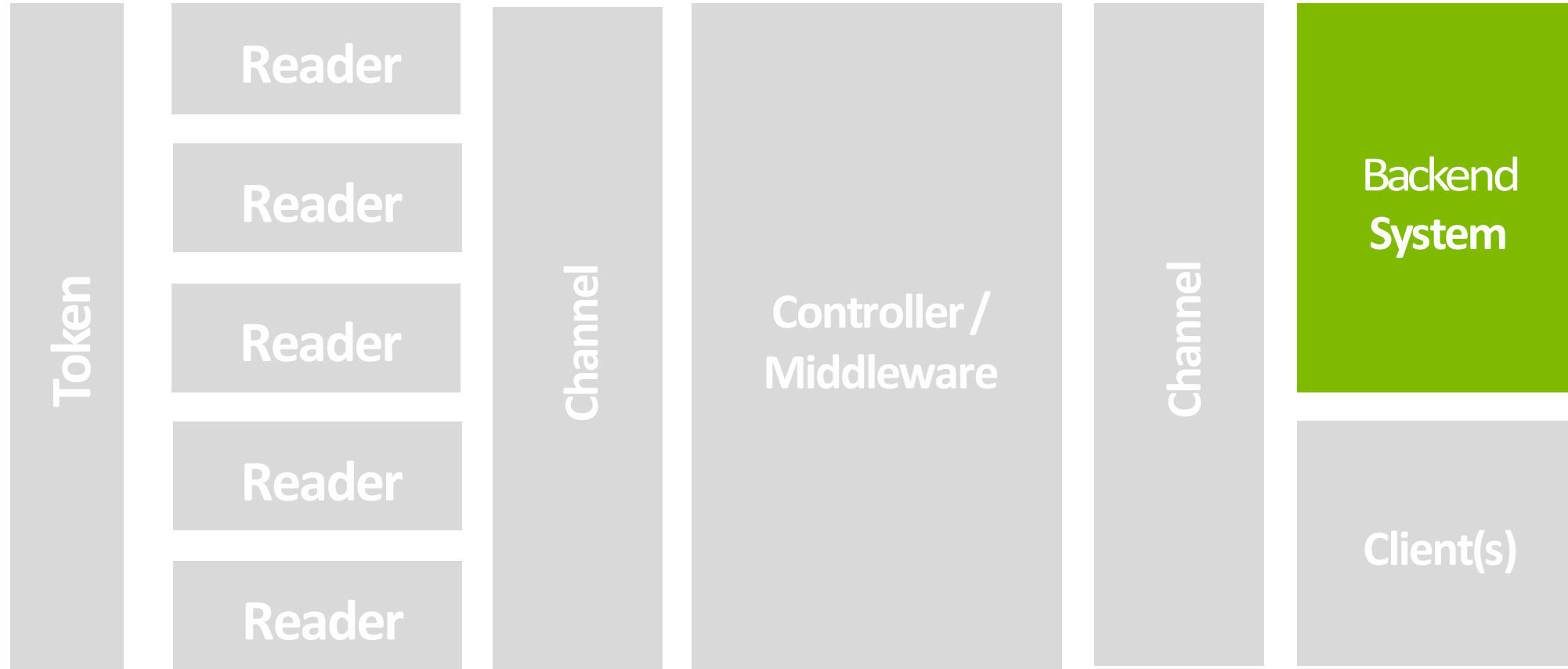
<b>Attack Surface</b>	<b>Attacks to Perform</b>	<b>Impact</b>
NFC Interface	Analyze the authentication mechanisms	Secrets extraction, MiTM attacks
Hardware board	Analyze the exposed interface (JTAG, UART, etc.)	Firmware or secrets dumping
Ethernet, wiegand, etc.	Is MITM possible? Intercepting the exchanged data	Intercepting secrets or sensitive data

# Access Control system attack surface ||



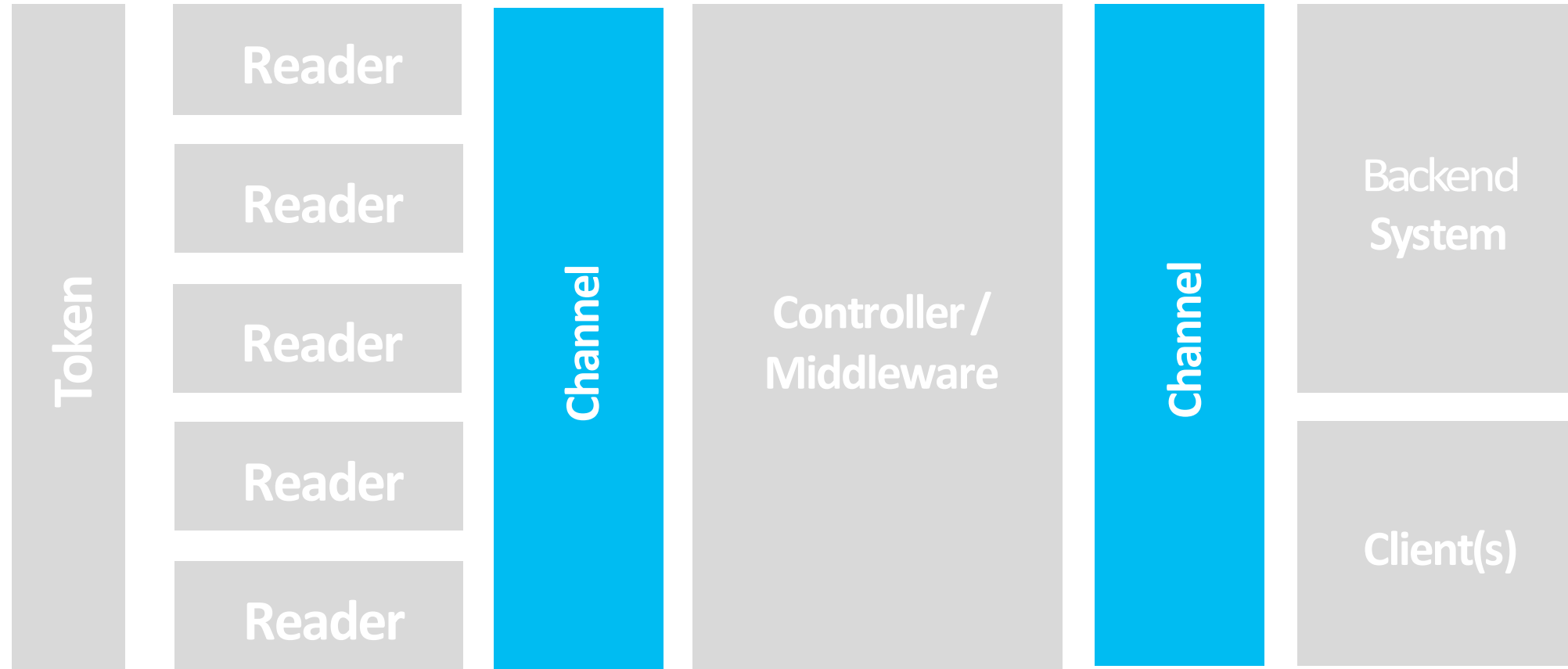
<b>Attack Surface</b>	<b>Attacks to Perform</b>	<b>Impact</b>
Hardware board	Analyze the exposed interface (JTAG, UART, etc.)	Firmware or secrets dumping
Eth, serial Interfaces, etc.	Is MITM possible? Intercepting the data	Intercepting secrets or sensitive data
Computer Application	Analyzing exposed network services	Complete control of the machine (e.g., add new users)

# Access Control system attack surface ||



<b>Attack Surface</b>	<b>Attacks to Perform</b>	<b>Impact</b>
Web application(s)	Classic web app-related attacks	Data exfiltration, service interruption, etc.
Network service(s)	Classic network services-related attacks	Data exfiltration, service interruption, etc.
Physical location	Try to get physical access to the servers	Basically, heavily PWNED

# Access Control system attack surface ||



<b>Attack Surface</b>	<b>Attacks to Perform</b>	<b>Impact</b>
Hardware board	Identify forgotten or backdoor pins	Data exfiltration, firmware dumping
External wires	Try to intercept data passing through those wires	Intercepting sensitive information
Wireless connection	Intercept and inject data	Intercepting sensitive information, send spoofed information



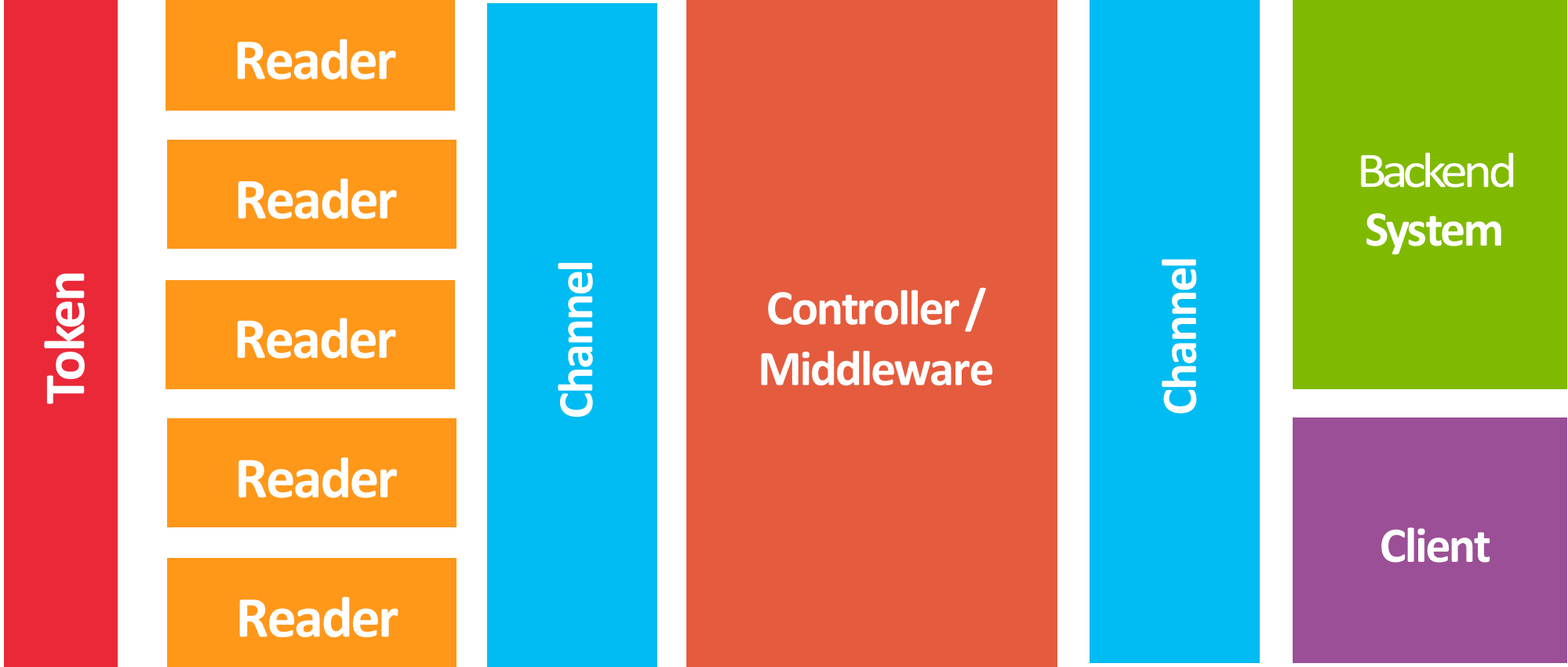
- Module 2 – Attacking NFC
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies

Fire up your

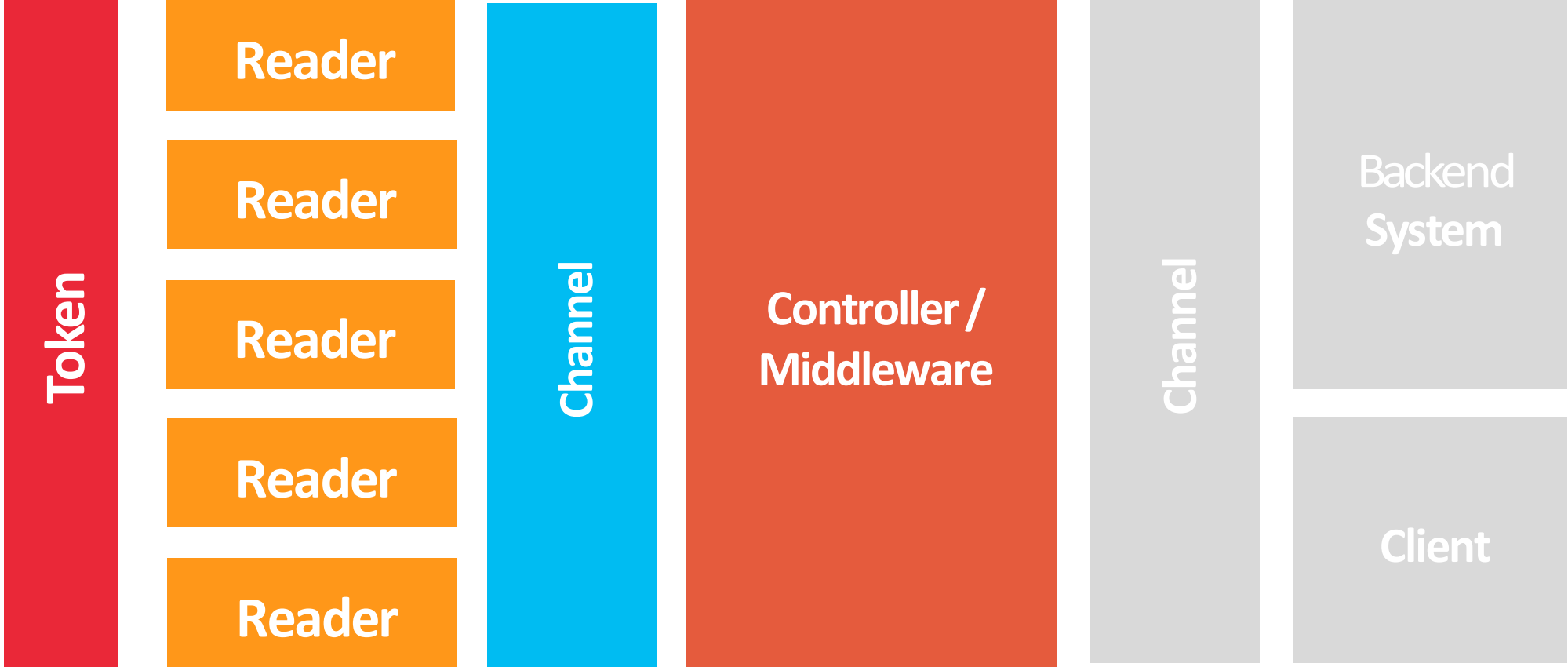


- Module 2 – attacking NFC
  - NFC: what are we talking about?
  - Weapons for NFC-based solutions
  - Penetration test methodology
  - Hands-on
  - Case studies

# MIFARE Ultralight ticketing system | |



# MIFARE Ultralight ticketing system | |



# MIFARE Ultralight ticketing system | |

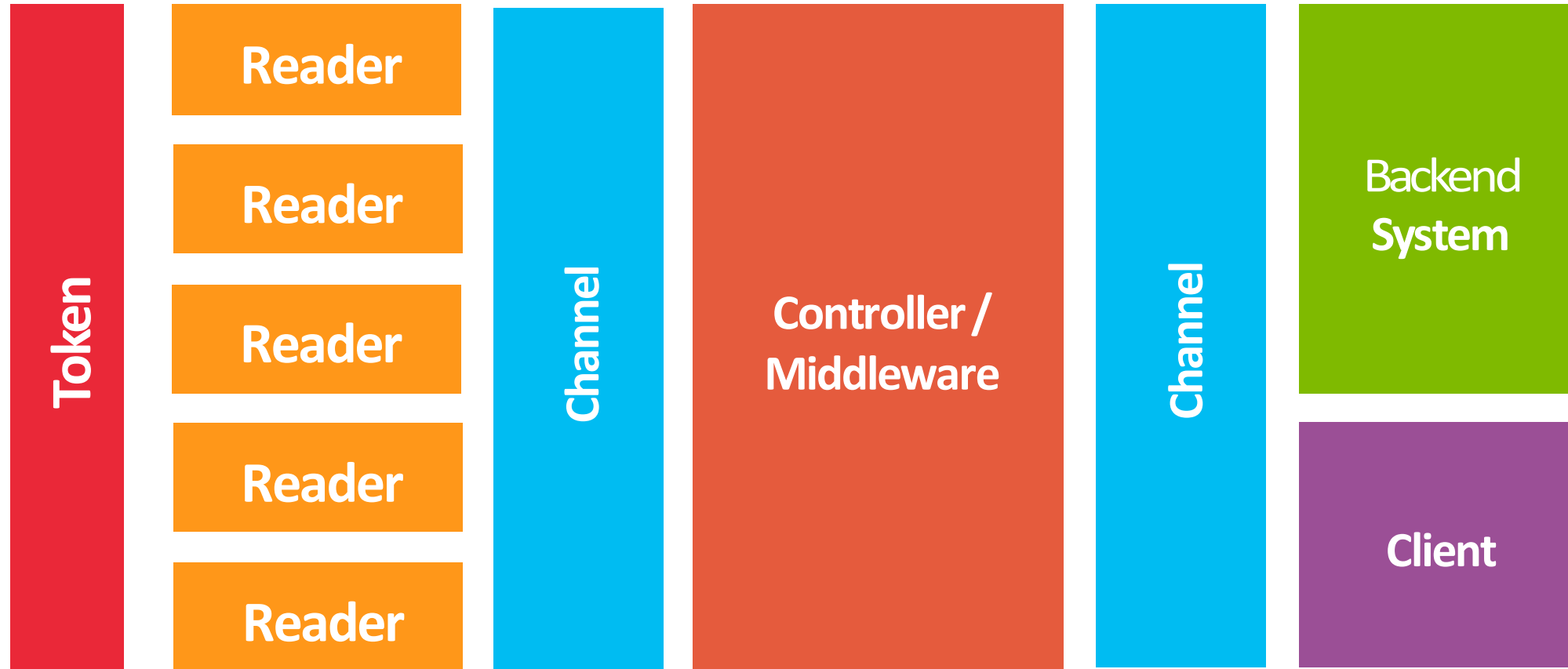
04FBC7B0	2AC13180	° «∞*   1Ä
5A48F203	FFFFFFFF	ZHÚ
01050000	020102BD	Ω
484A4000	00AE10A0	HJ@ Æ †
A0000473	8A84035D	† sãÑ ]
51432E00	04F80000	QC. -
51432E00	001D0004	QC.
F8AE10A0	140249E5	- Æ † IÂ

Absence of a UID blacklist in the backend

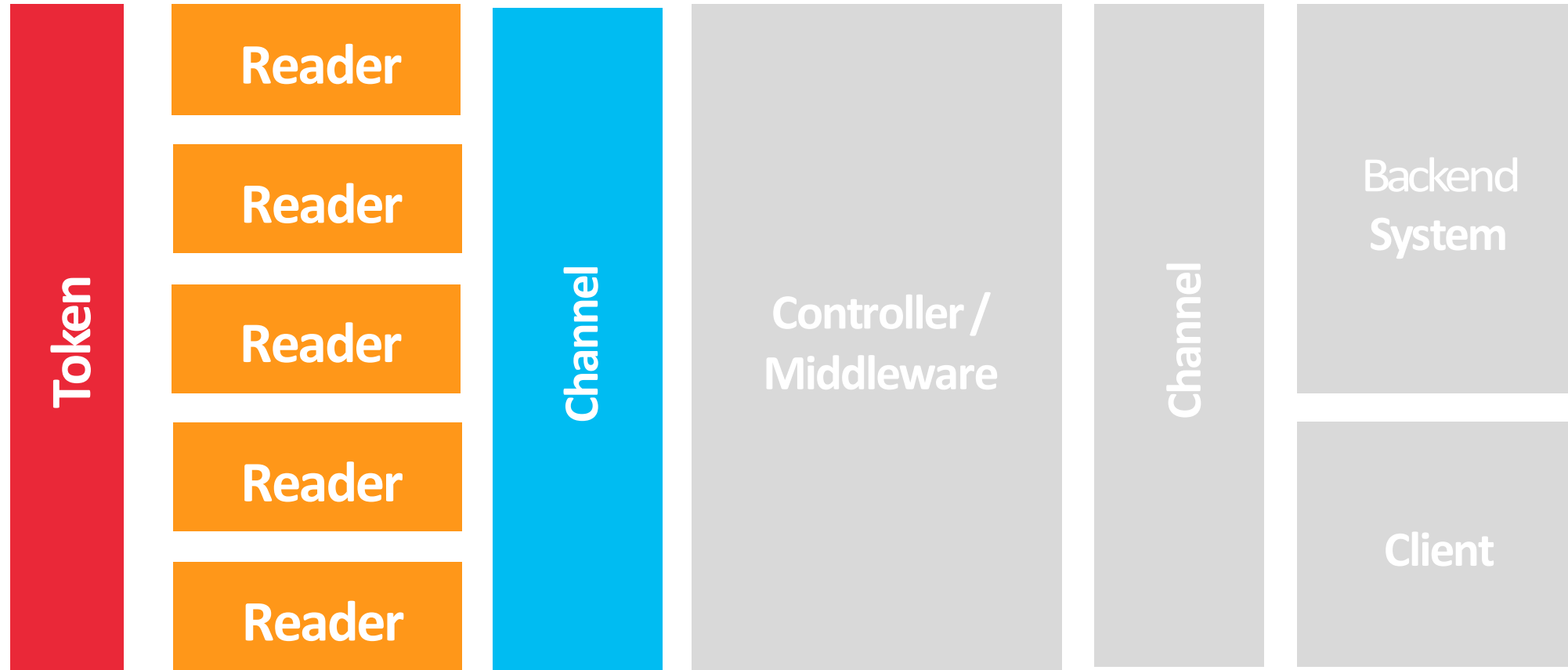
Lock bit for the OTP sector is not checked by the stamping machine

Timestamps are not encrypted nor signed

# MIFARE Classic hotel door lock ||



# MIFARE Classic hotel door lock ||





# MIFARE Classic door lock | |

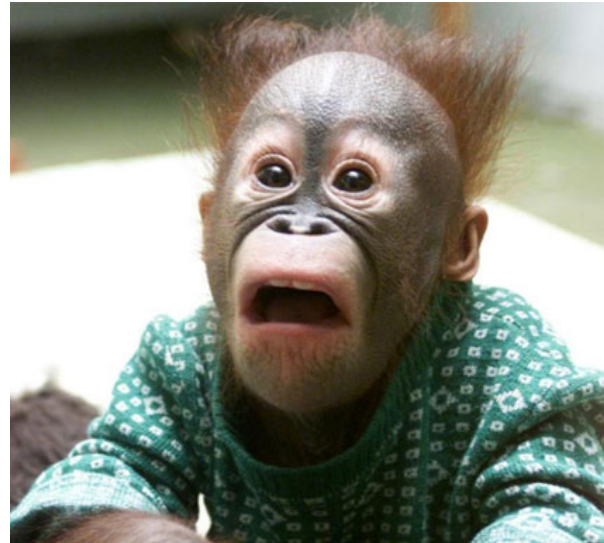


# Module 3 || attacking RF communications

- Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security

# Radio Frequency and EAC Systems ||

- Radio Frequency identification is widely used to control physical accesses
- Advantages
  - Automatic identification
  - High reliability
  - High security



- Different technologies based on operating frequency band
  - Low Frequency (LF) – 125 KHz
  - High Frequency (HF) – 13.56 MHz
  - Ultra High Frequency (UHF) – 433 MHz, 860-960 MHz and 2.4 GHz

# Radio Frequency and EAC Systems | |

## Low Frequency band

- Tags
- Access control token



# Radio Frequency and EAC Systems | |

## High Frequency band

- Door locks
- Ticketing systems



# Radio Frequency and EAC Systems | |

## Ultra High Frequency band

- Automated Gates
- Keyless Entry Systems
- Alarms
- Smart Locks





# Radio Frequency and EAC Systems | |

- Common technologies and protocols
  - Fixed and rolling code
  - NFC
  - Bluetooth
  - ZigBee
  - Z-Wave

- Module 3 –Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security

# Exploring Radio Frequency communication | |

- How to explore wireless communications?
  - Software Defined Radio (SDR) devices with GNU Radio
- Software implementation of most parts of a radio system
  - Cheap hardware
  - High flexible

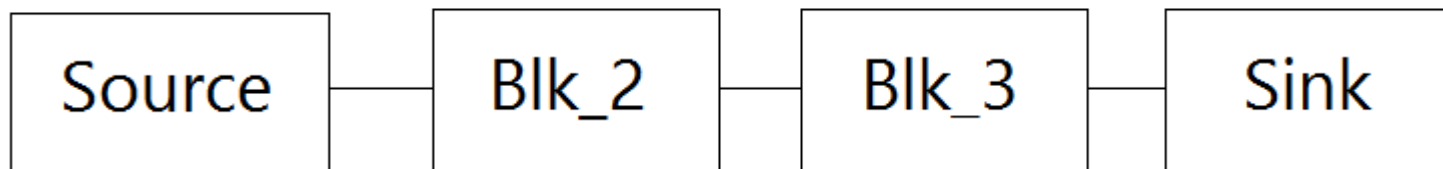
# Exploring Radio Frequency communication | |

## Three SDR-compatible devices

Device	Frequency Range	Bandwidth	Price
RTL-SDR Dongle	24 MHz – 1.76 GHz	2.4 MHz	~ 20 €
HackRF	1 MHz – 6 GHz	20 MHz	~ 300 €
USRP B200	70 MHz – 6 GHz	56 MHz	~ 700 €

# Exploring Radio Frequency communication | |

- GNU Radio
  - Platform to develop radio applications, called **flowgraphs**
    - Series of connected signal processing blocks
  - GNU Radio libraries include blocks to perform signal processing

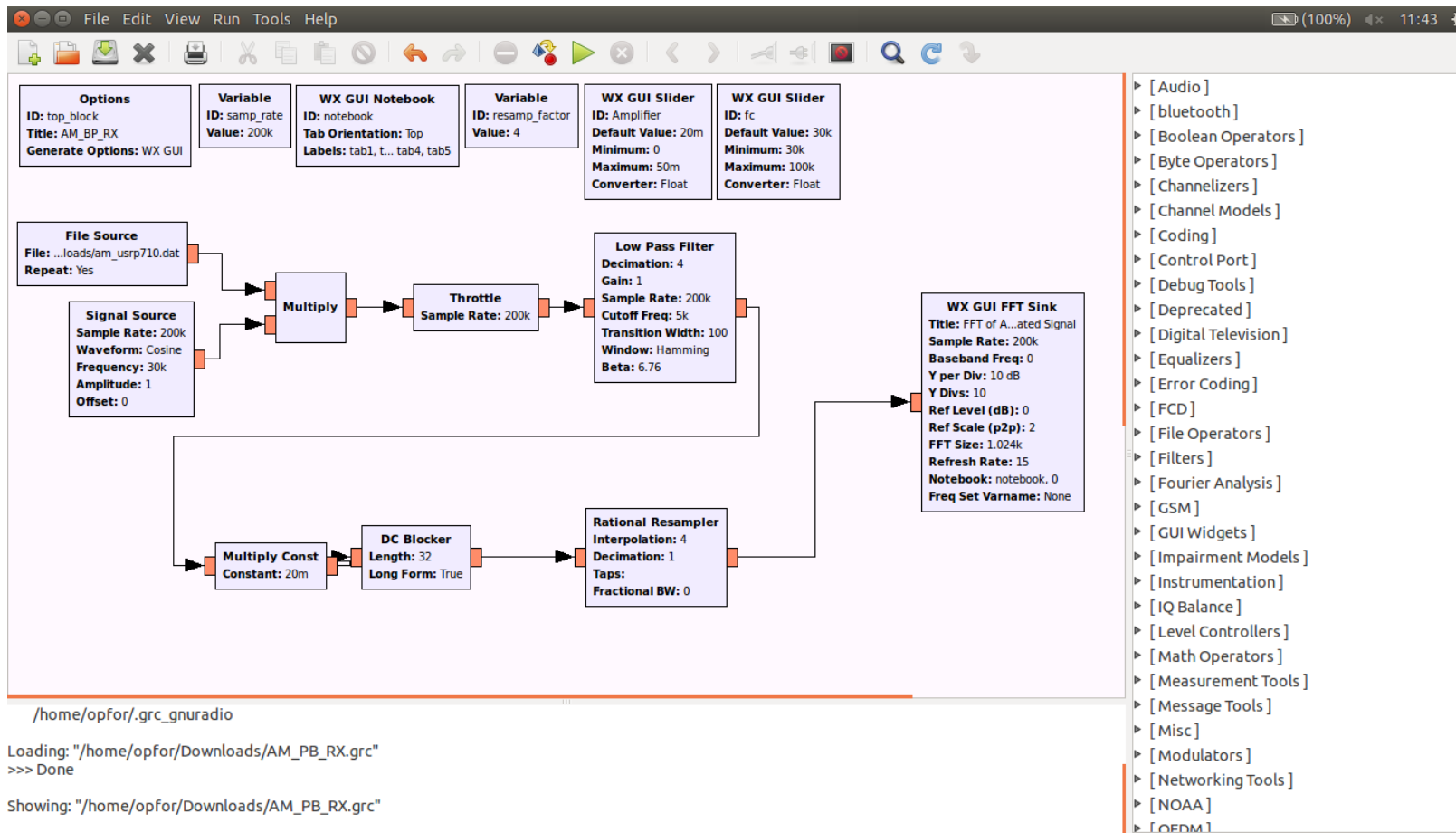


# Exploring Radio Frequency communication | |

- GNU Radio
  - Supports the programming of custom C++ blocks
  - GNU Radio Companion (GRC)
    - Graphical UI to program GNU Radio applications
    - Supports the creation of UI for applications

# Exploring Radio Frequency communication | |

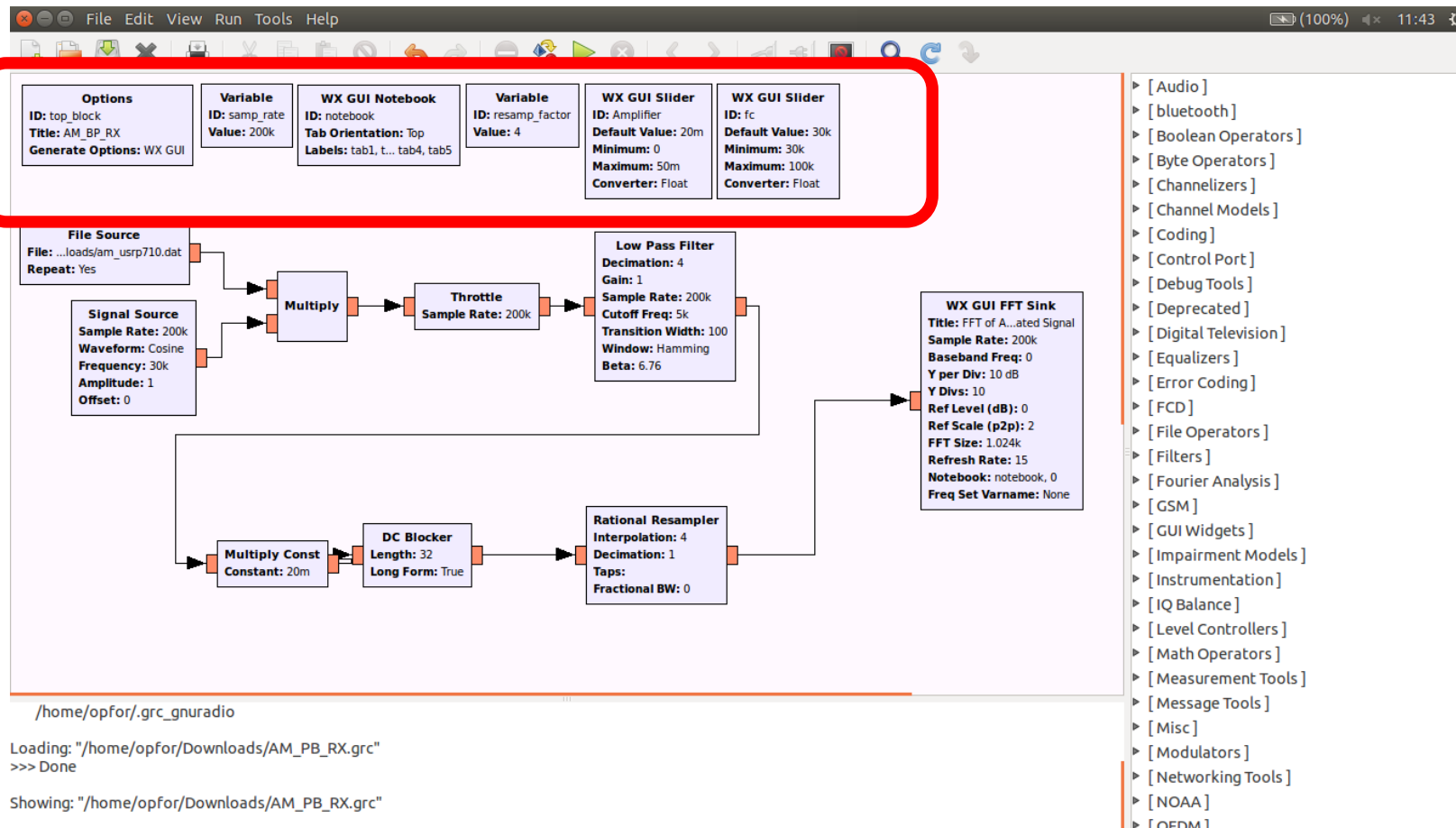
## ■ GRC Interface



# Exploring Radio Frequency communication | |

## ■ GRC Interface

VARIABLES

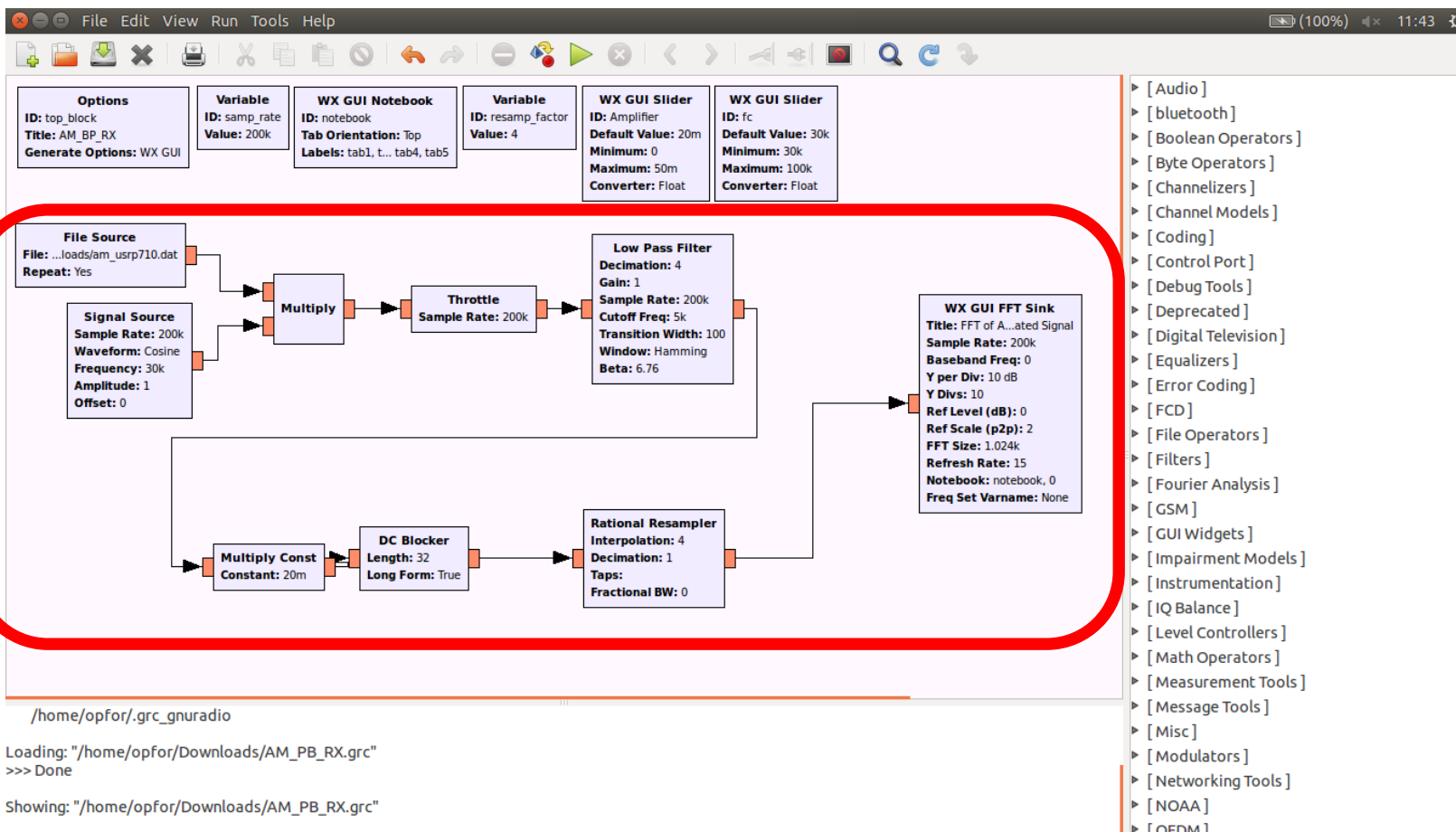




# Exploring Radio Frequency communication | |

- GRC Interface

FLOWGRAPH



# Exploring Radio Frequency communication | |

- GRC Interface

The screenshot displays the GNU Radio Companion (GRC) interface. The main workspace contains a signal flow graph with the following components and connections:

- File Source** (File: ...loads/am\_usrp710.dat, Repeat: Yes) and **Signal Source** (Sample Rate: 200k, Waveform: Cosine, Frequency: 30k, Amplitude: 1, Offset: 0) are connected to a **Multiply** block.
- The output of the **Multiply** block goes to a **Throttle** block (Sample Rate: 200k).
- The output of the **Throttle** block goes to a **Low Pass Filter** (Decimation: 4, Gain: 1, Sample Rate: 200k, Cutoff Freq: 5k, Transition Width: 100, Window: Hamming, Beta: 6.76).
- The output of the **Low Pass Filter** goes to a **WX GUI FFT Sink** (Title: FFT of A...ated Signal, Sample Rate: 200k, Baseband Freq: 0, Y per Div: 10 dB, Y Divs: 10, Ref Level (dB): 0, Ref Scale (p2p): 2, FFT Size: 1.024k, Refresh Rate: 15, Notebook: notebook, 0, Freq Set Varname: None).
- There is also a **Multiply Const** block (Constant: 20m) connected to a **DC Blocker** (Length: 32, Long Form: True), which is connected to a **Rational Resampler** (Interpolation: 4, Decimation: 1, Taps: Fractional BW: 0).

The terminal window at the bottom shows the following output:

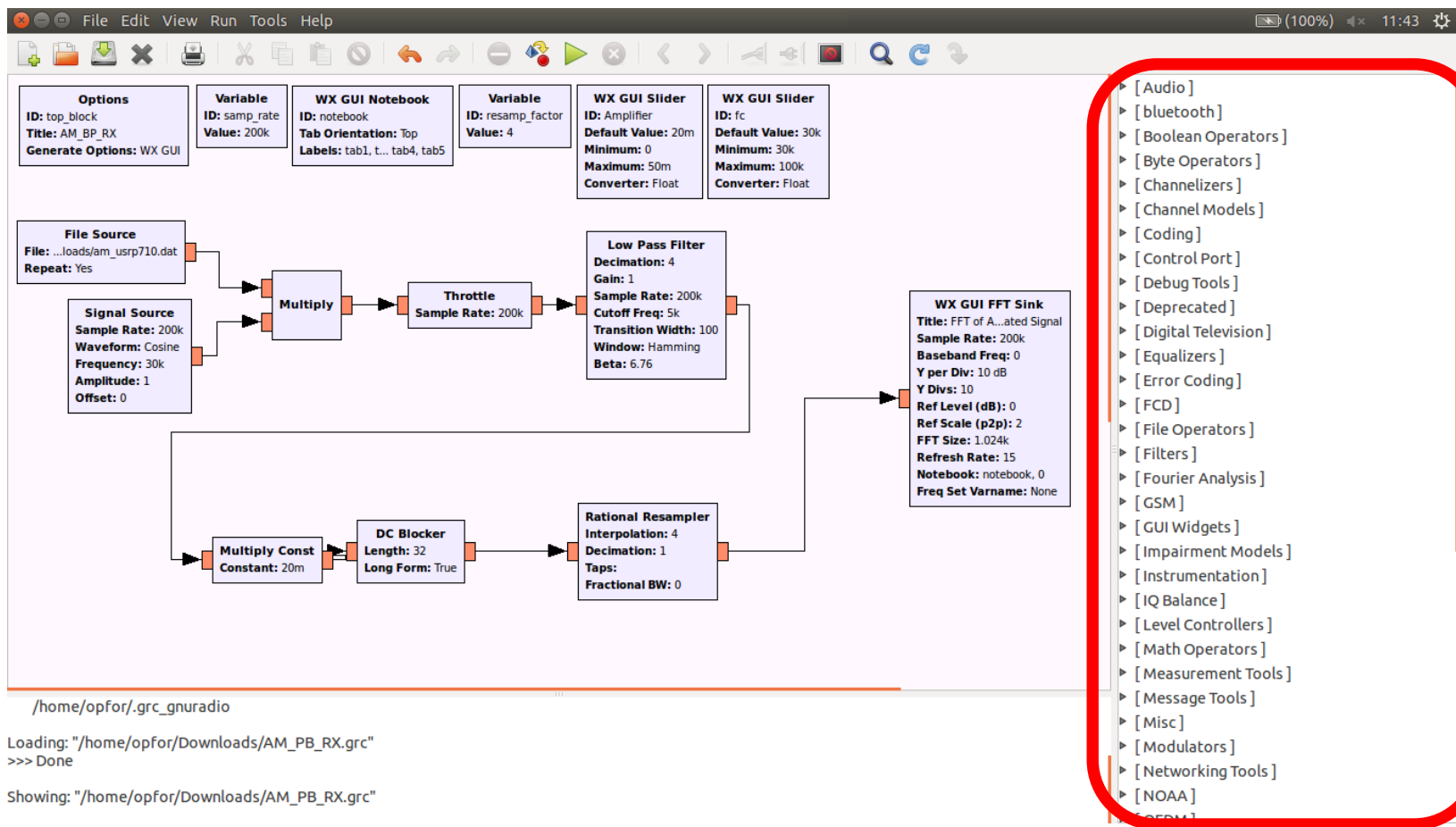
```
/home/opfor/.grc_gnuradio
Loading: "/home/opfor/Downloads/AM_PB_RX.grc"
>>> Done
Showing: "/home/opfor/Downloads/AM_PB_RX.grc"
```

TERMINAL



# Exploring Radio Frequency communication | |

- GRC Interface



BLOCK LIBRARY

# Exploring Radio Frequency communication | |

- “Hello World” in GNU Radio

The screenshot shows the GNU Radio Companion (GRC) interface for a project named 'hello\_world.grc'. The main workspace contains a flow graph with three blocks: 'Options', 'RTL-SDR Source', and 'WX GUI FFT Sink'. The 'Options' block is connected to the 'RTL-SDR Source' block, which is connected to the 'WX GUI FFT Sink' block. A 'Variable' block is also present, connected to the 'RTL-SDR Source' block.

**Options**  
ID: top\_block  
Generate Options: WX GUI

**Variable**  
ID: samp\_rate  
Value: 32k

**RTL-SDR Source**  
Sample Rate (sps): 32k  
Ch0: Frequency (Hz): 434M  
Ch0: Freq. Corr. (ppm): 0  
Ch0: DC Offset Mode: Off  
Ch0: IQ Balance Mode: Off  
Ch0: Gain Mode: Manual  
Ch0: RF Gain (dB): 10  
Ch0: IF Gain (dB): 20  
Ch0: BB Gain (dB): 20

**WX GUI FFT Sink**  
Title: FFT Plot  
Sample Rate: 32k  
Baseband Freq: 0  
Y per Div: 10 dB  
Y Divs: 10  
Ref Level (dB): 0  
Ref Scale (p2p): 2  
FFT Size: 1.024k  
Refresh Rate: 15  
Freq Set Varname: None

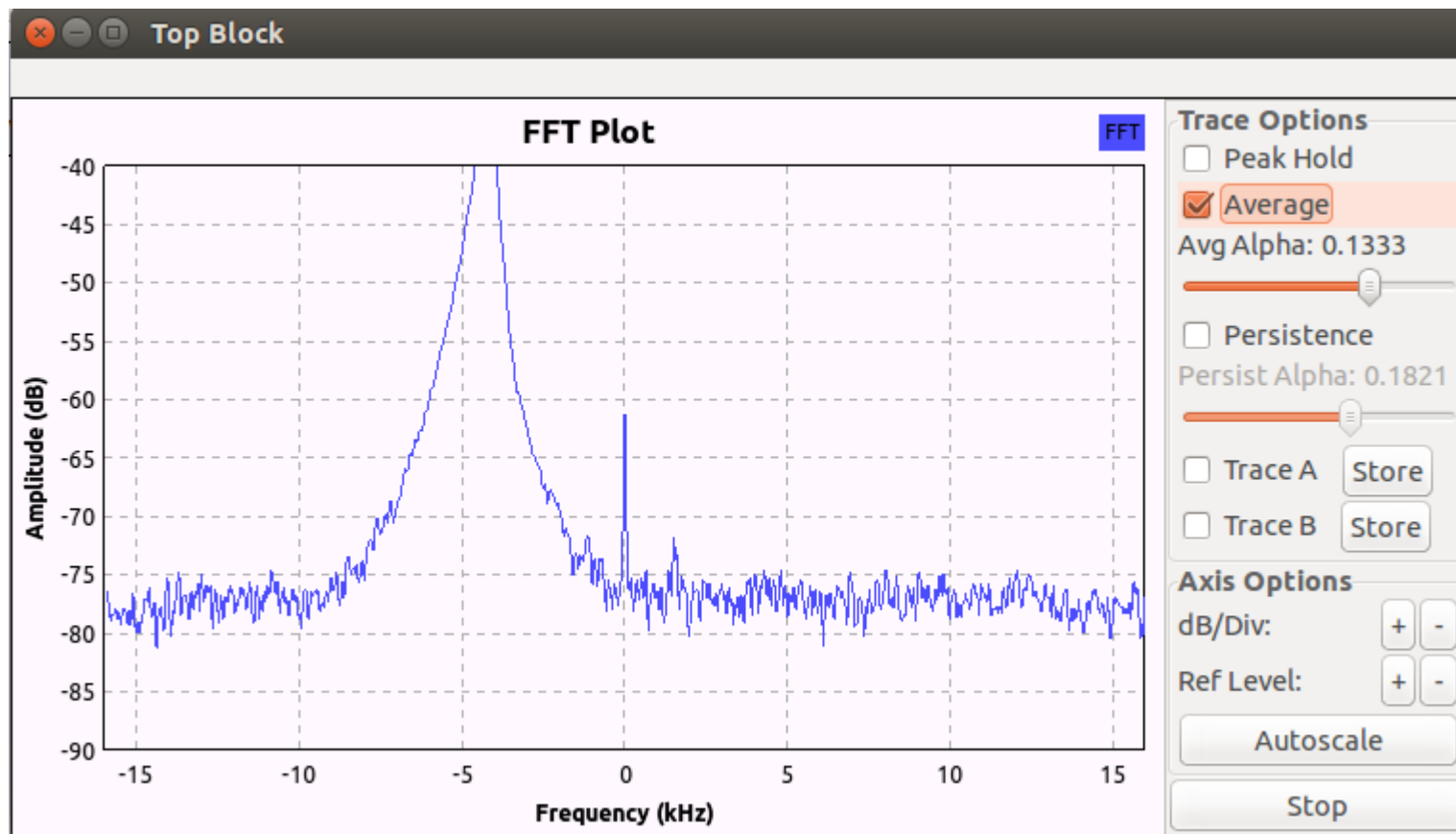
Invalid sample rate: 32000 Hz  
rtlsdr\_read\_async returned with -5

>>> Done

The right sidebar shows a list of categories for the GRC blocks, including [ Audio ], [ bluetooth ], [ Boolean Operators ], [ Byte Operators ], [ Channelizers ], [ Channel Models ], [ Coding ], [ Control Port ], [ Debug Tools ], [ Deprecated ], [ Digital Television ], [ Equalizers ], [ Error Coding ], [ FCD ], [ File Operators ], [ Filters ], [ Fourier Analysis ], [ GSM ], [ GUI Widgets ], [ Impairment Models ], [ Instrumentation ], [ IQ Balance ], and [ Level Controllers ].

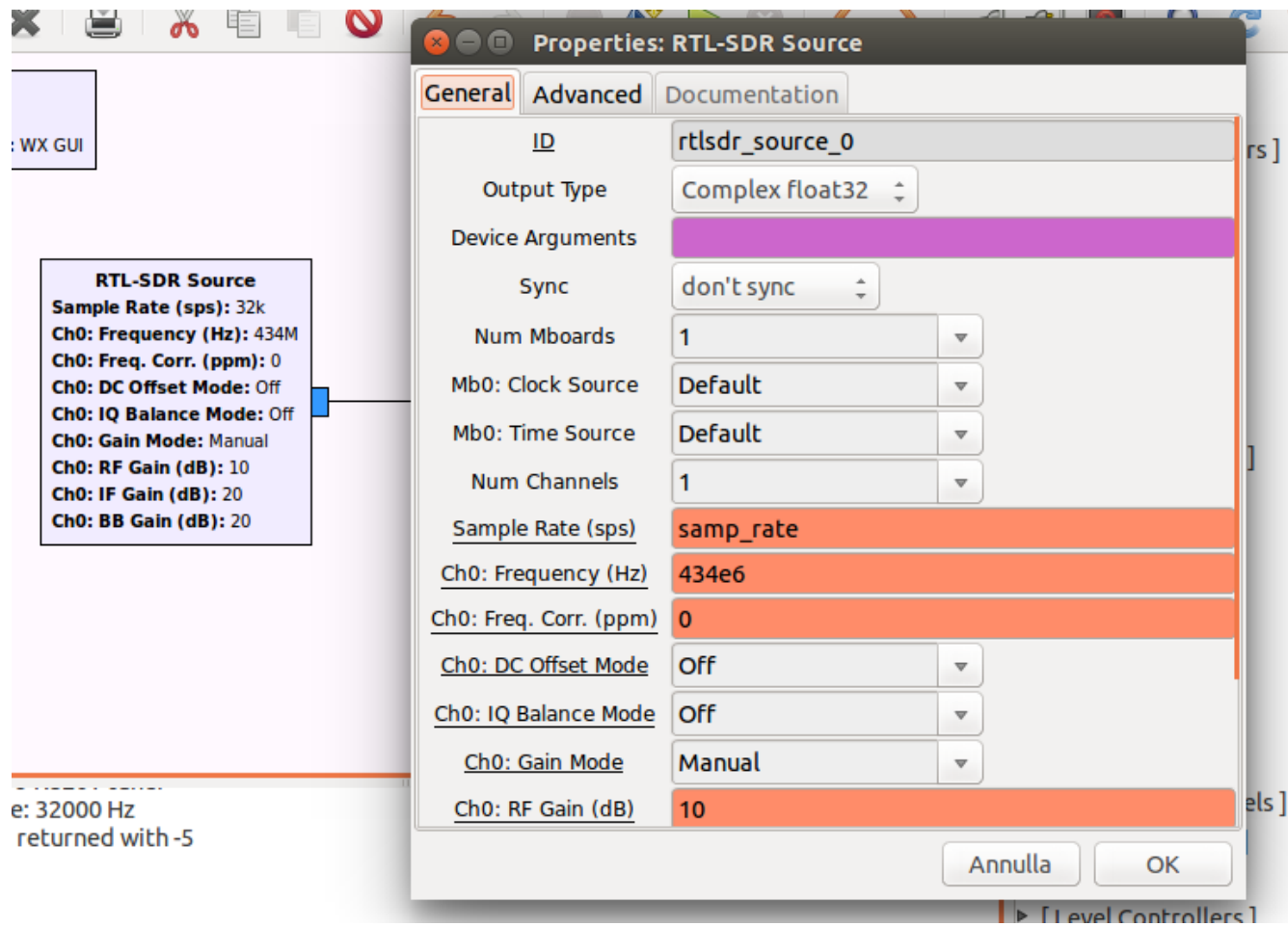
# Exploring Radio Frequency communication | |

- “Hello World” in GNU Radio



# Exploring Radio Frequency communication | |

- RTL-SDR Source Block



The screenshot shows a software interface with a block diagram on the left and a configuration dialog on the right. The block diagram includes a purple block labeled 'WX GUI' and a white block labeled 'RTL-SDR Source'. The 'RTL-SDR Source' block has a tooltip with the following settings:

- RTL-SDR Source
- Sample Rate (sps): 32k
- Ch0: Frequency (Hz): 434M
- Ch0: Freq. Corr. (ppm): 0
- Ch0: DC Offset Mode: Off
- Ch0: IQ Balance Mode: Off
- Ch0: Gain Mode: Manual
- Ch0: RF Gain (dB): 10
- Ch0: IF Gain (dB): 20
- Ch0: BB Gain (dB): 20

The 'Properties: RTL-SDR Source' dialog box is open, showing the 'General' tab. The settings are as follows:

Property	Value
ID	rtlsdr_source_0
Output Type	Complex float32
Device Arguments	
Sync	don't sync
Num Mboards	1
Mb0: Clock Source	Default
Mb0: Time Source	Default
Num Channels	1
Sample Rate (sps)	samp_rate
Ch0: Frequency (Hz)	434e6
Ch0: Freq. Corr. (ppm)	0
Ch0: DC Offset Mode	Off
Ch0: IQ Balance Mode	Off
Ch0: Gain Mode	Manual
Ch0: RF Gain (dB)	10

At the bottom of the dialog box, there are 'Annulla' and 'OK' buttons. Below the dialog box, there is a status bar that reads 'e: 32000 Hz returned with -5'.

# Exploring Radio Frequency communication | |

- WX GUI FFT Sink Block

The image shows a software interface for configuring a 'WX GUI FFT Sink' block. On the left is a 'Properties' dialog box with a 'General' tab selected. It contains various settings for the block, such as ID, Type, Title, Sample Rate, Baseband Freq, Y per Div, Y Divs, Ref Level (dB), Ref Scale (p2p), FFT Size, Refresh Rate, Peak Hold, Average, Window, and Window Size. On the right is a summary box titled 'WX GUI FFT Sink' that lists the current values for these settings. An arrow points from the 'Y Divs' field in the dialog box to the summary box.

Property	Value
ID	wxgui_fftsink2_0
Type	Complex
Title	FFT Plot
Sample Rate	samp_rate
Baseband Freq	0
Y per Div	10 dB
Y Divs	10
Ref Level (dB)	0
Ref Scale (p2p)	2.0
FFT Size	1024
Refresh Rate	15
Peak Hold	Off
Average	Off
Window	Automatic
Window Size	

**WX GUI FFT Sink**  
Title: FFT Plot  
Sample Rate: 32k  
Baseband Freq: 0  
Y per Div: 10 dB  
Y Divs: 10  
Ref Level (dB): 0  
Ref Scale (p2p): 2  
FFT Size: 1.024k  
Refresh Rate: 15  
Freq Set Varname: None

- Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security



Build a FM receiver

Fire up your



- Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security

- Define a methodology to study real world signals
- Three main steps

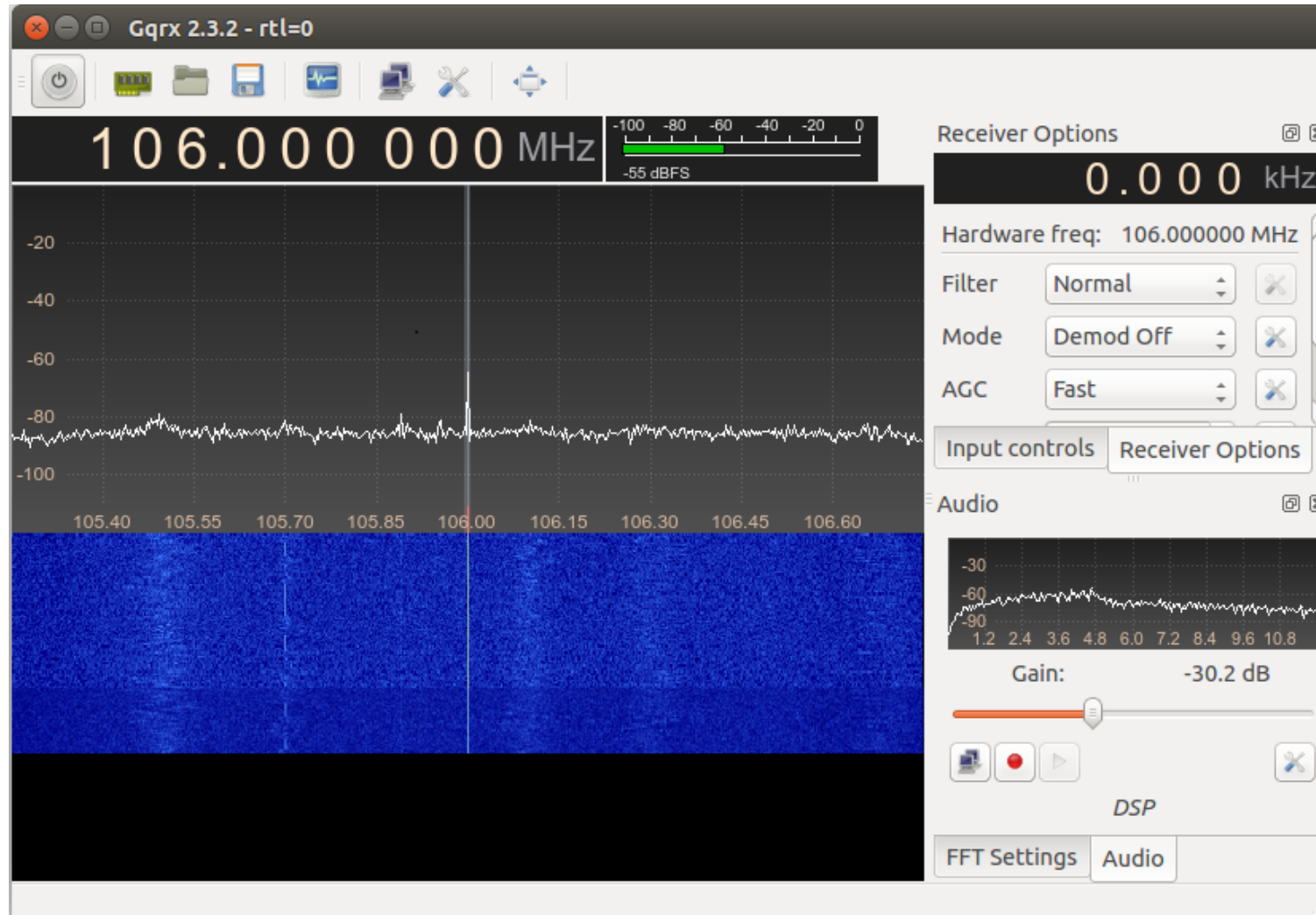


- Define a methodology to study real world signals
- Three main steps



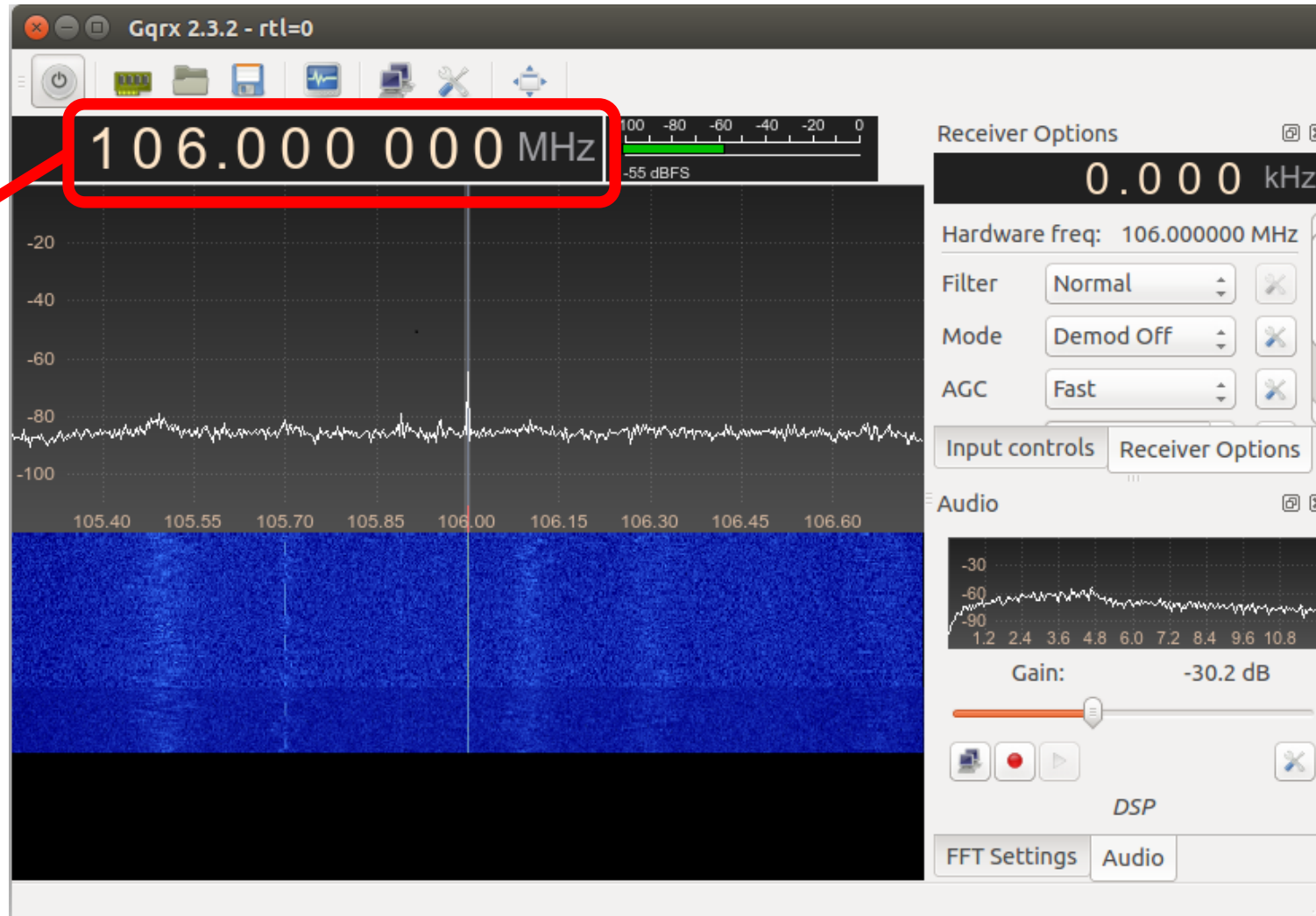
- GQRX
  - SDR receiver and spectrum analyzer based on GNU Radio and QT Graphical toolkit
  - User-friendly interface
  - Supports RTL-SDR, HackRF, USRP and other SDR devices
  - Records signal to WAV file

# SIGINT with GNU Radio ||



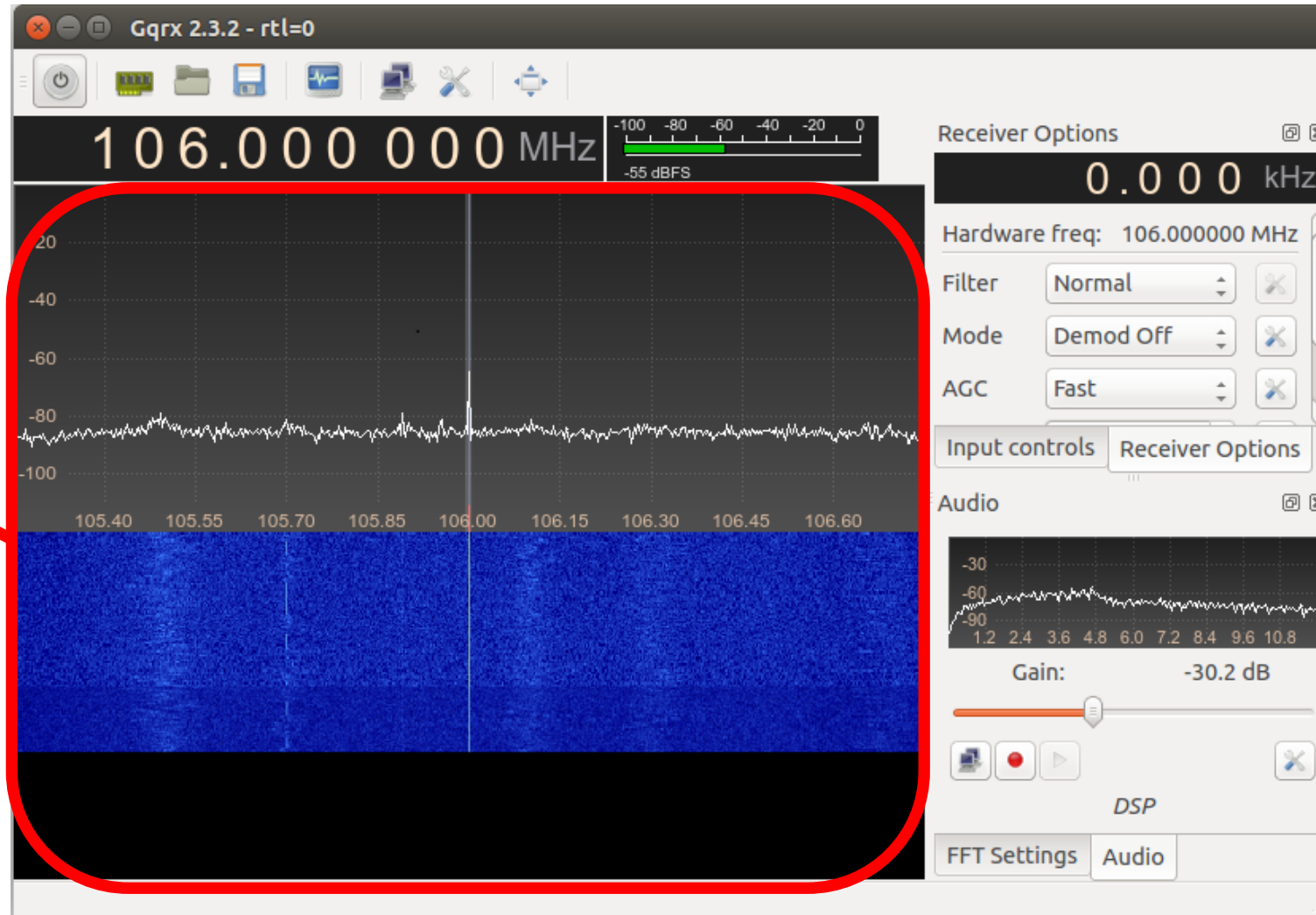
# SIGINT with GNU Radio ||

FREQUENCY  
SELECTOR



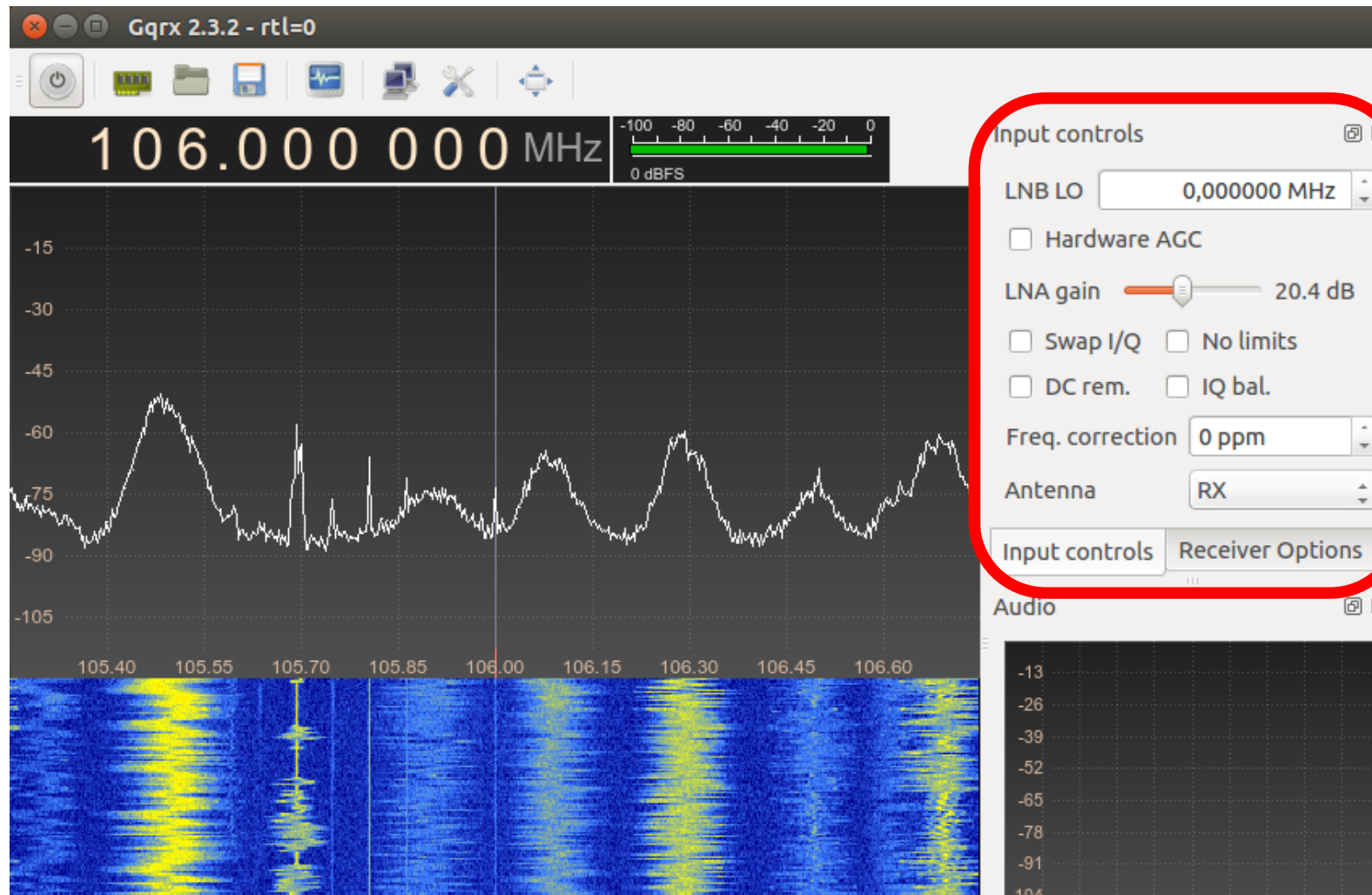
# SIGINT with GNU Radio ||

REAL-TIME  
SPECTRUM



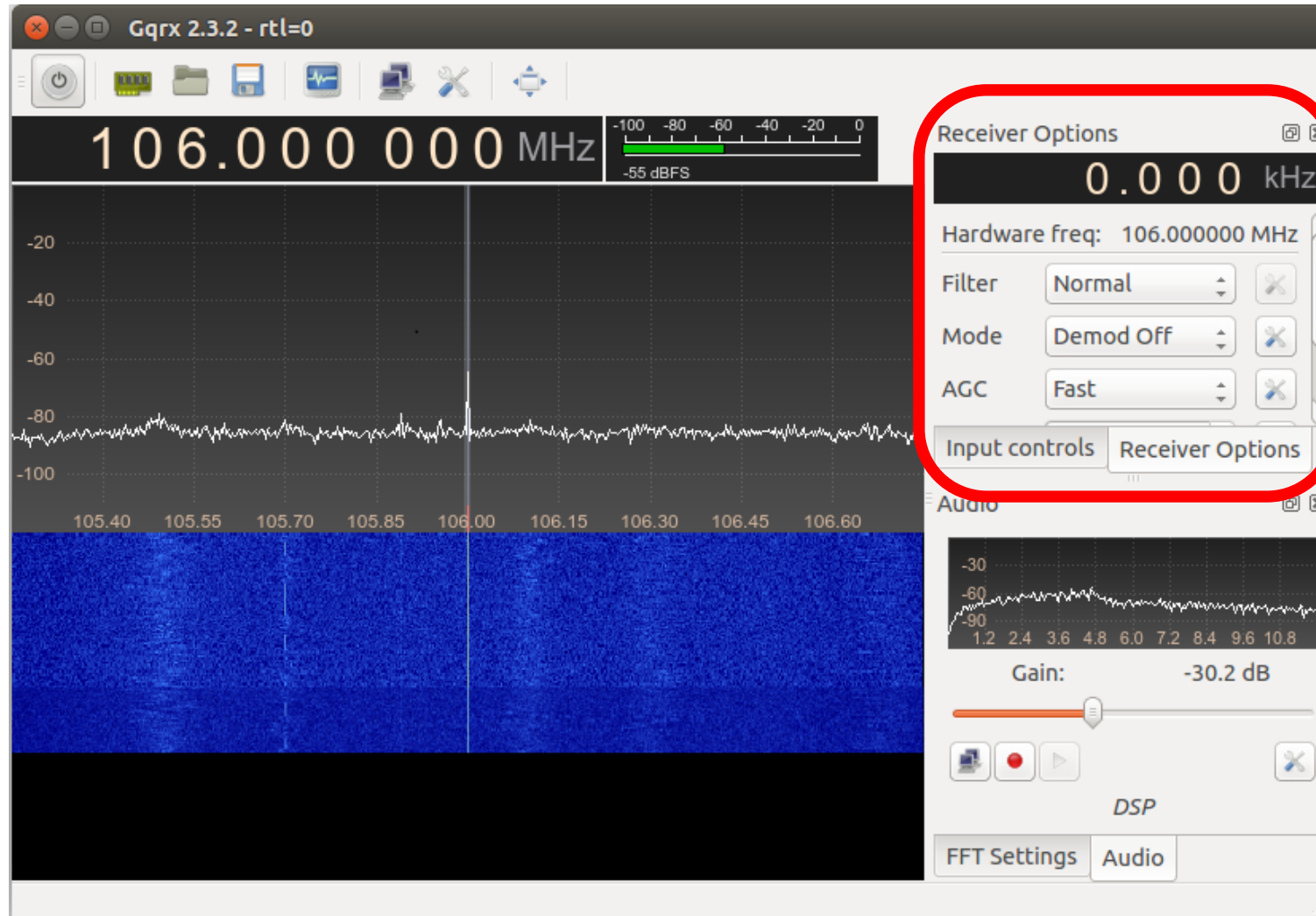


# SIGINT with GNU Radio ||



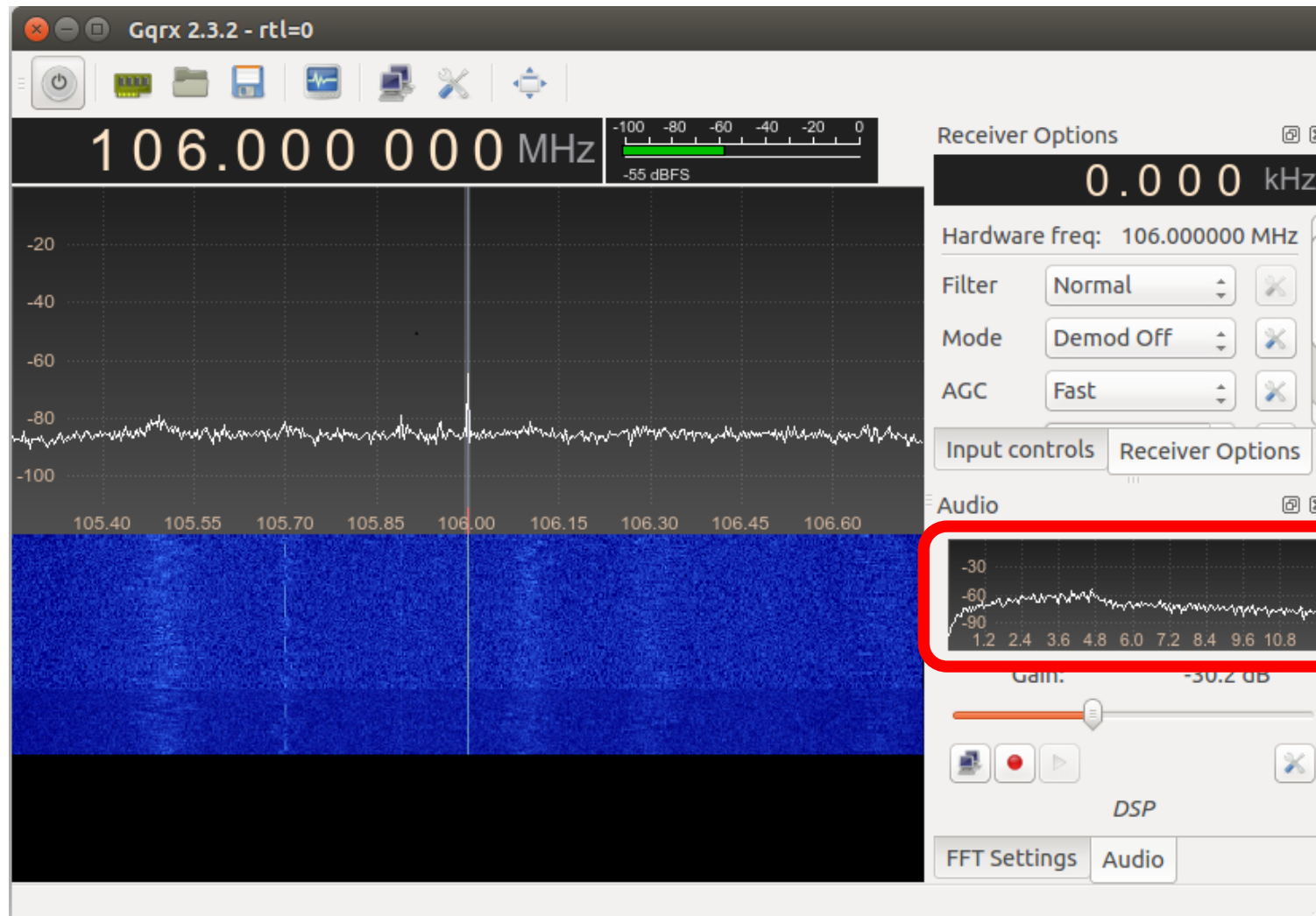
INPUT  
CONTROLS

# SIGINT with GNU Radio ||



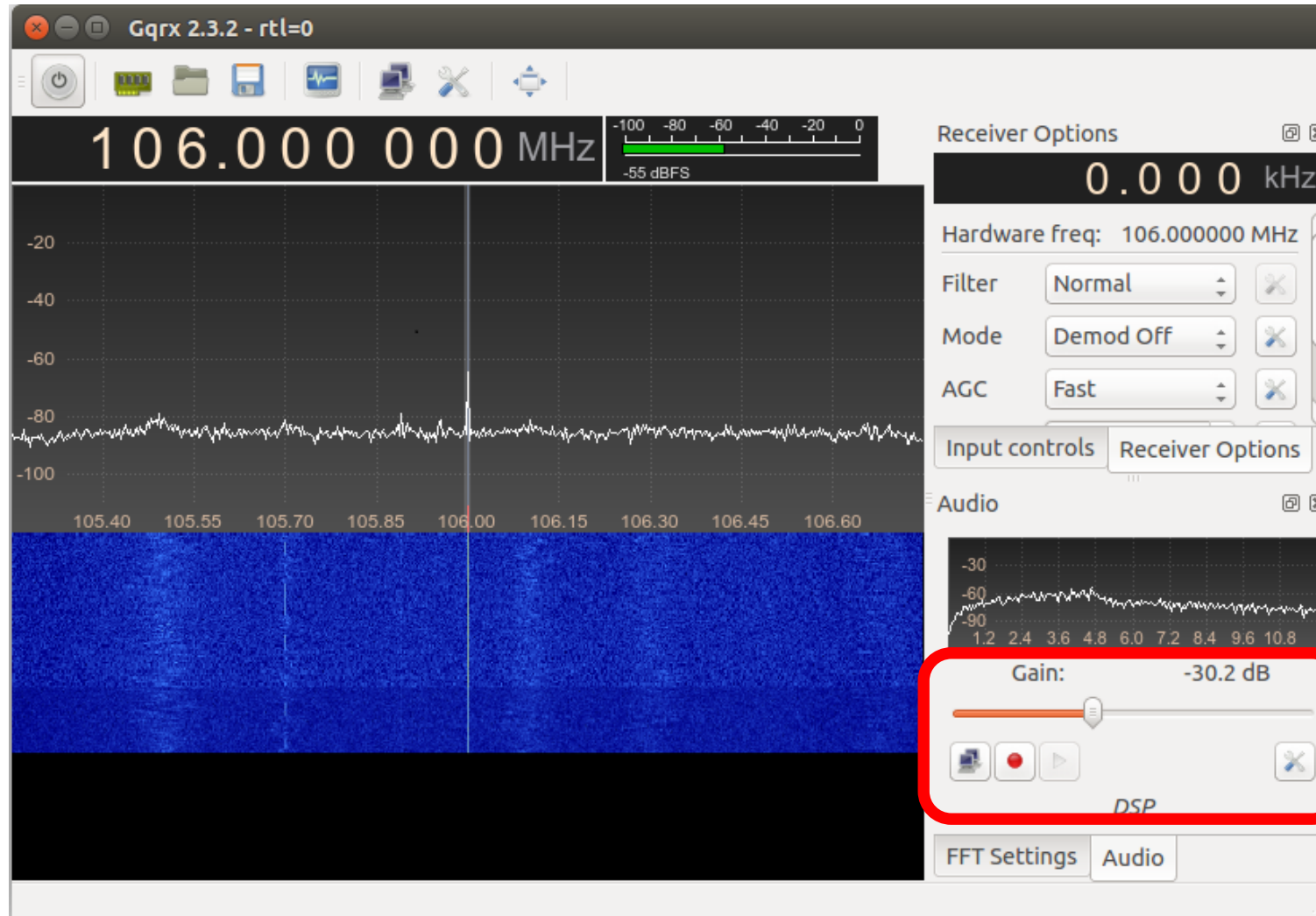
RECEIVER  
OPTIONS

# SIGINT with GNU Radio ||



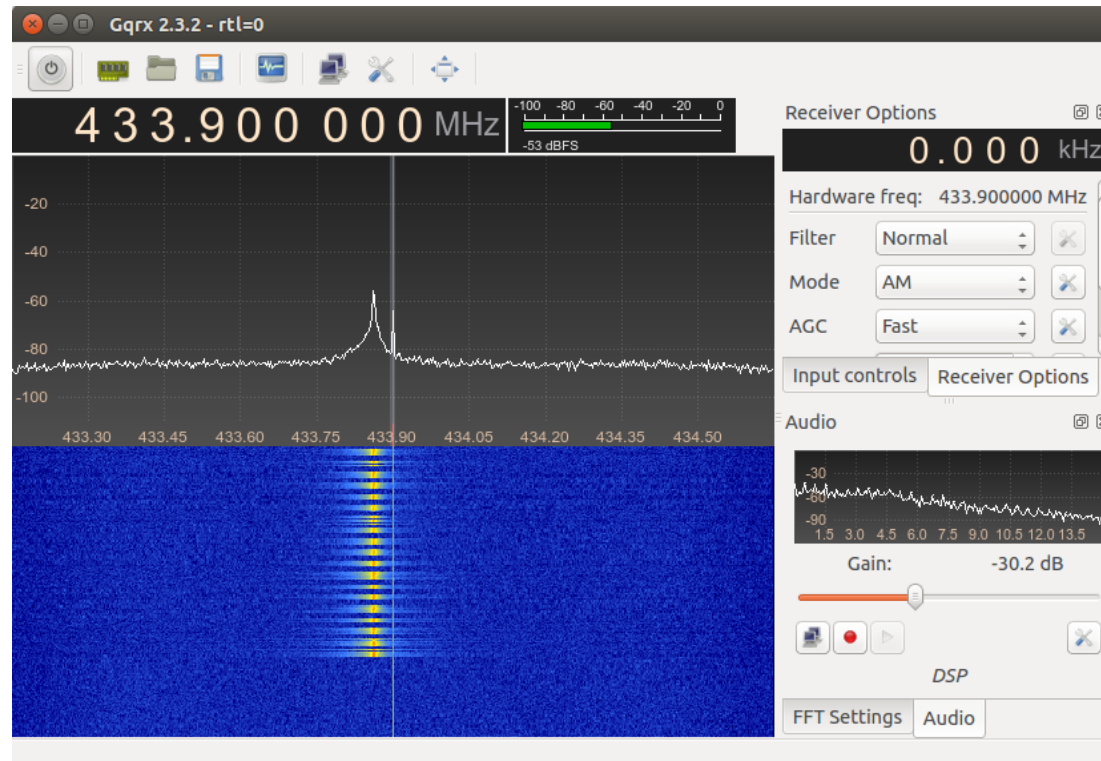
DEMODULATED  
SPECTRUM

# SIGINT with GNU Radio ||



# SIGINT with GNU Radio ||

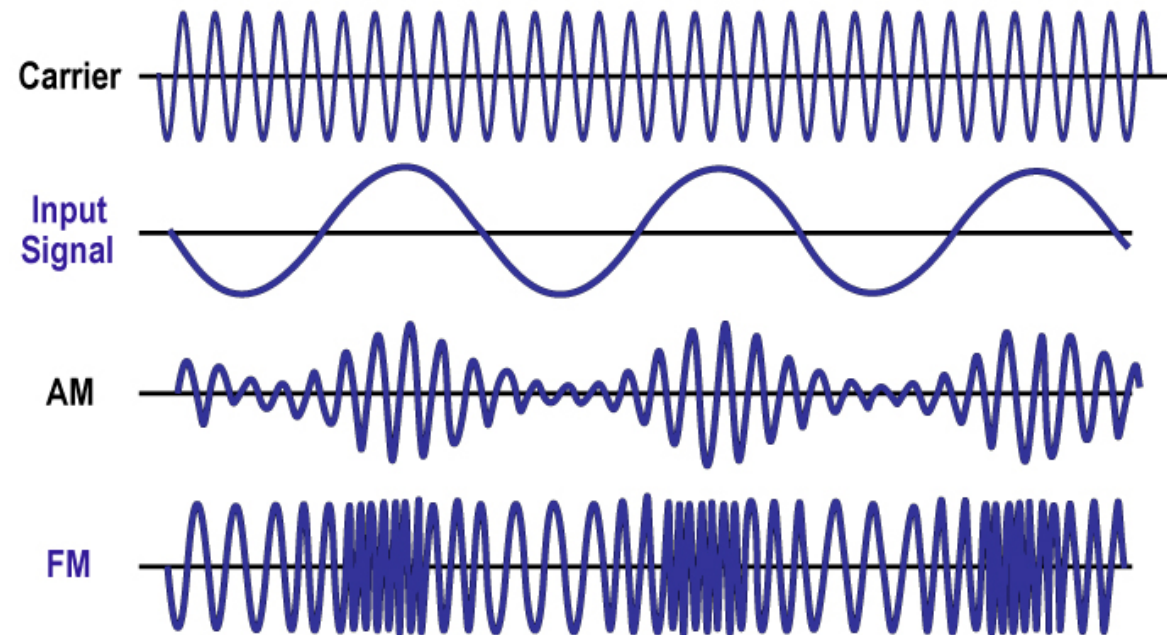
- Black-box interception of a RF signal
  - If the frequency is unknown, search power **peaks** in the spectrum



- Define a methodology to study real world signals
- Three main steps



- Modulation
  - Impresses a waveform, called **carrier**, with another signal that contains data to be transmitted



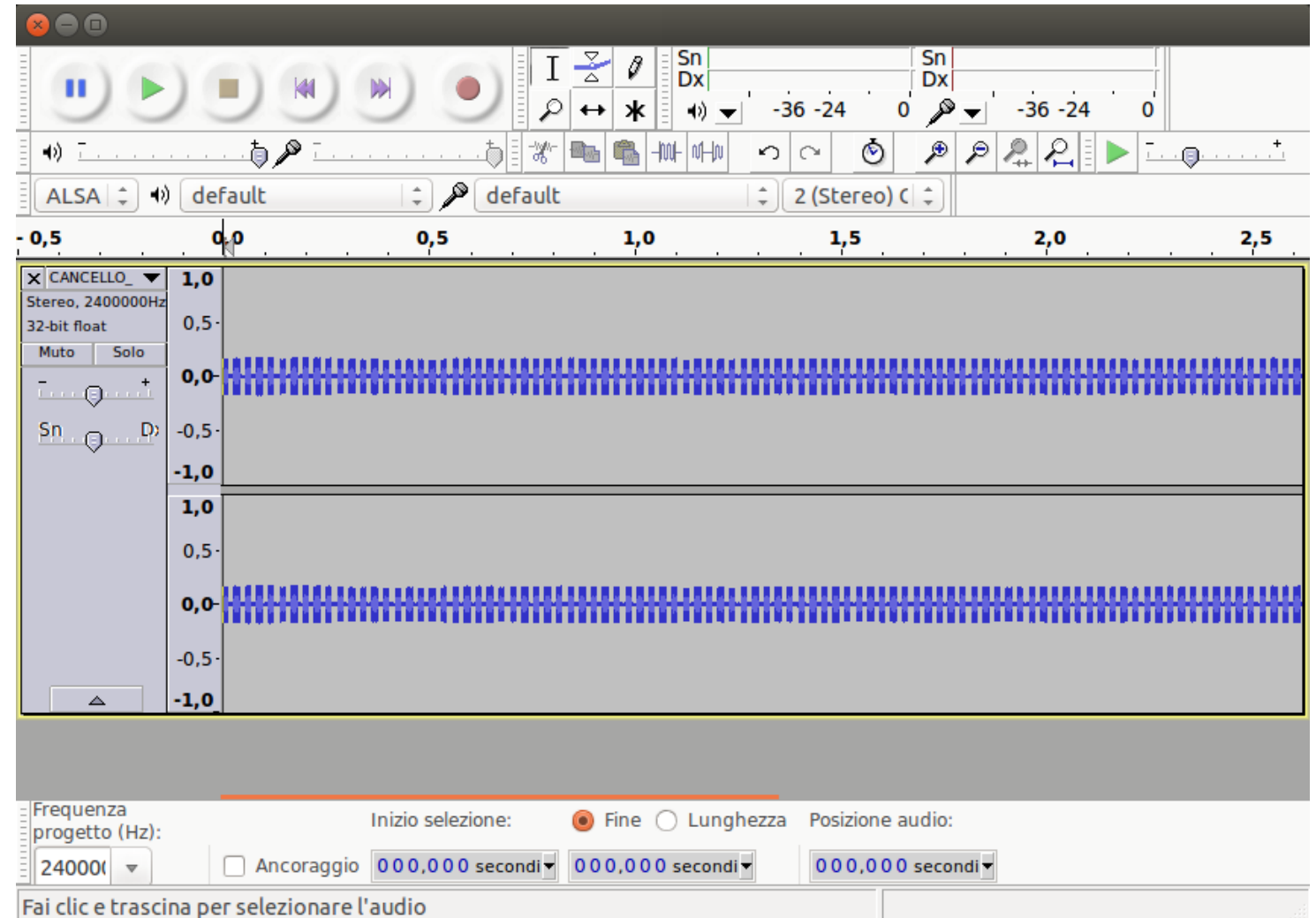
- Signal Identification Guide

Signal Name	Description	Frequency Range	Modulation	Transmission Type	Bandwidth	Location	Audio Player	Spectrogram
<b>The Buzzer (MDZhB UZB76)</b>	famously known by its former call-sign UZB76, is a Russian based military station that occasionally broadcasts <i>Monolit</i> format messages in Russian. Its trademark buzzer is constantly transmitted while there is no message to broadcast.	4.625 MHz — 6.998 MHz	AM	USB	2.8 kHz	Russia	00:00:00	
<b>Tire Pressure Monitoring System (TPMS)</b>	Signal is from a Chrysler TPMS (Tire-Pressure Monitoring System) sensor.	315 MHz — 433 MHz	AM			Worldwide	00:00:00	
<b>Toyota Car Key</b>	Wireless entry rolling code car key.	315 MHz — 433 MHz	AM		40 kHz	Worldwide	00:00:00	



# SIGINT with GNU Radio ||

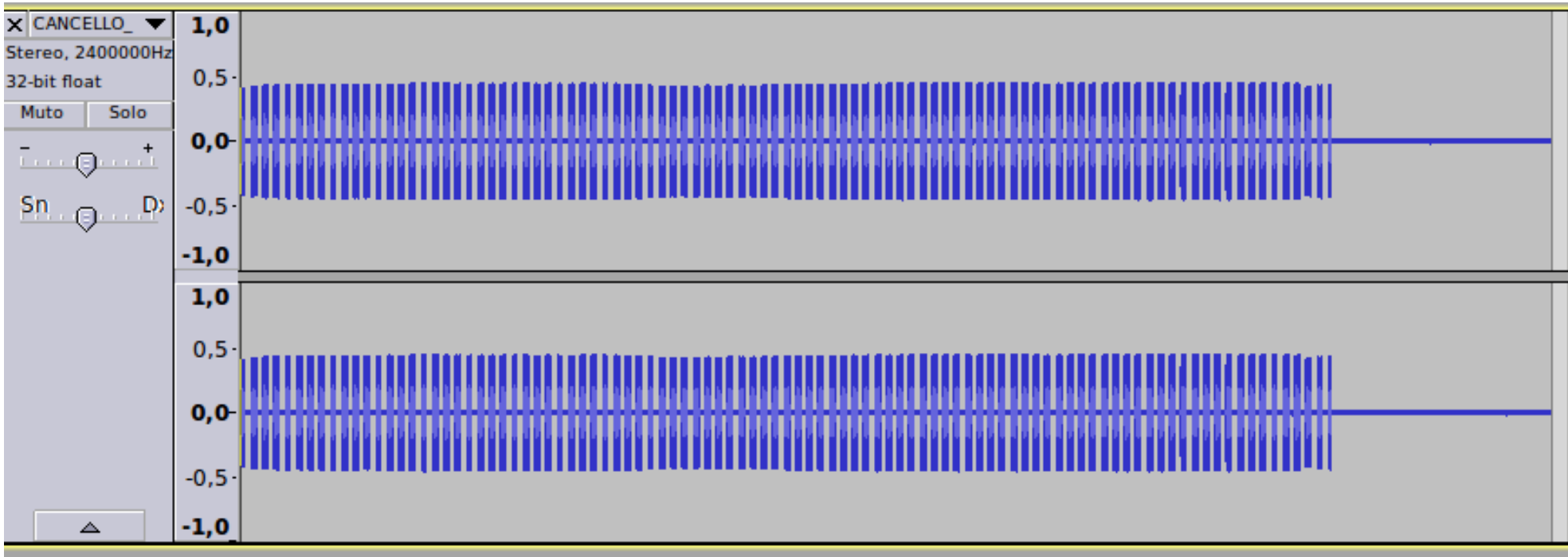
- Audacity
  - Useful to study recorded signals
  - Support RAW data files used with USRP and HackRF utilities



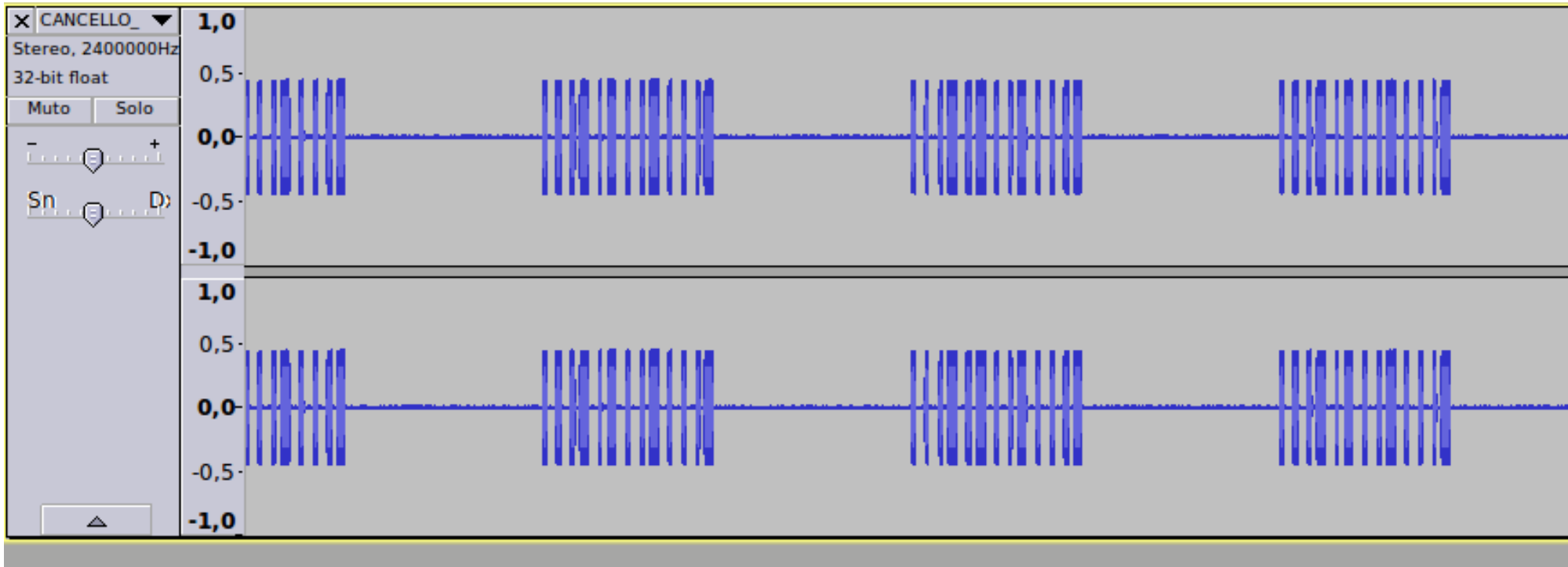
- Case Study: remote control at 433 MHz



- Case Study: remote control at 433 MHz

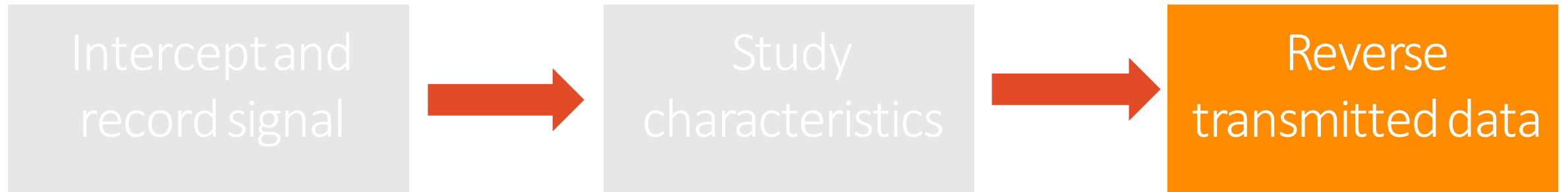


- Case Study: remote control at 433 MHz

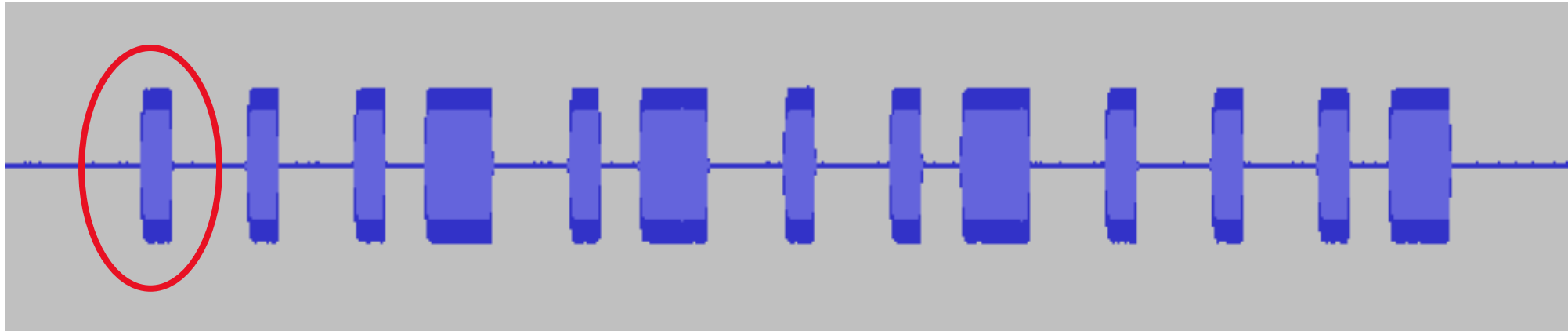


- Let's study the signal
  - Amplitude Modulation (AM)
  - Only two amplitude levels
    - Binary transmission using **On-Off Keying (OOK)** modulation
  - Repeated trains of pulses
    - Different lengths to encode the '0' and '1' bit

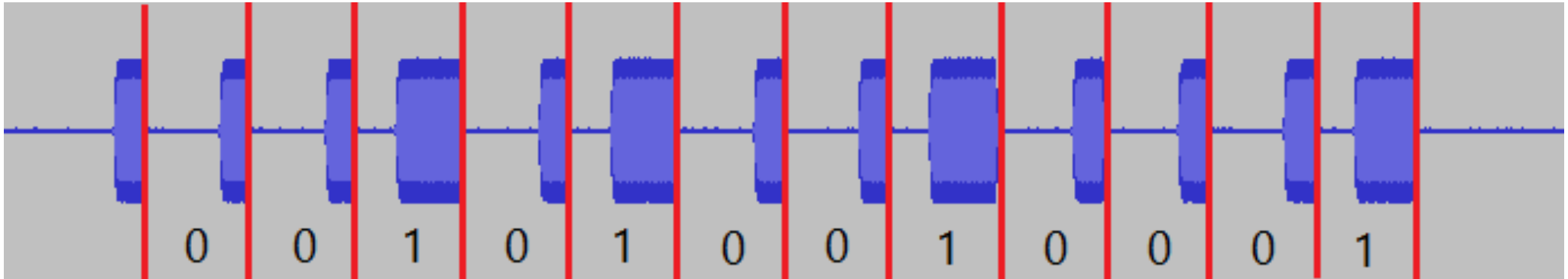
- Define a methodology to study real world signals
- Three main steps



- Focus on a single train
  - The first pulse indicates the beginning of the “message”



- Short pulses represent binary '0' while long pulses binary '1'



- Transmitted message is 001010010001



- Module 3 – Attacking RF communications
  - Radio Frequency and EAC Systems
  - Exploring Radio Frequency communications in practice
  - Hands-on: receiving your first transmission
  - SIGINT with GNU Radio
  - Understanding RF communications security

- Case study's solution security
  - The remote control always sends same **fixed** code (!)
  - Malicious people can record and replay signals thus obtaining an unauthorized access
- Solution
  - Rolling code

- Rolling Code
  - Remote control always sends **different** codes
  - Sender and receiver are synchronized with an internal counter
  - An hardware algorithm calculates the 'next' code on the basis of the internal counter's value
  - A widely used algorithm is **KeeLoq**
  - Rolling code is **NOT** a unbreakable mechanism..

# Module 4 | | the challenge

- Module 4 – The challenge
  - Introducing the challenge
  - The awards 😊

## Challenge introduction | |

You are now part of a Red Team, which has been engaged to breach the physical security of a high security facility controlled by a super secret, and “probably” evil, organization known as **h4k3rZ T34mZ**

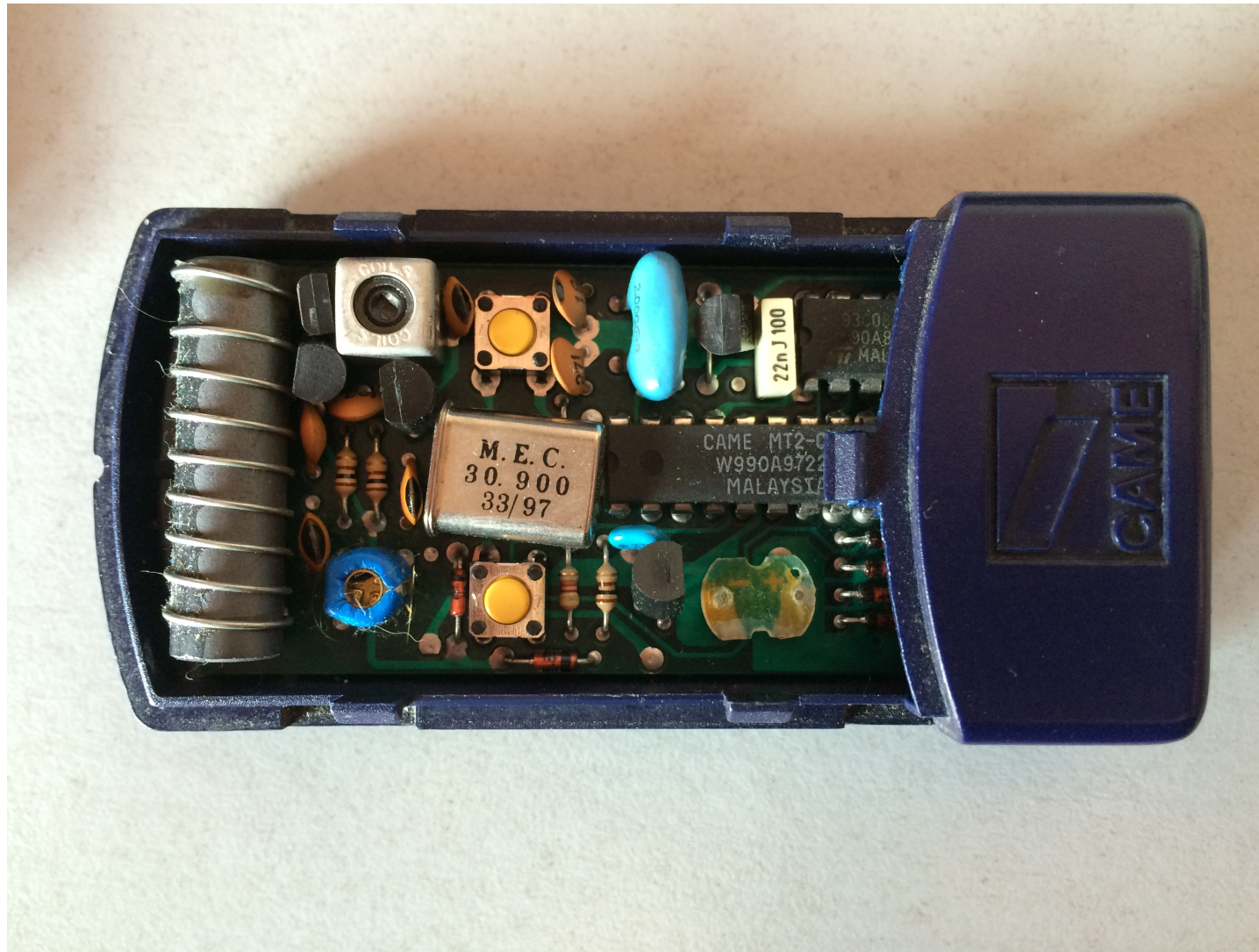
Your task is to open the external facility’s electric gate, thus allow your team to enter the facility and proceed with the intrusion..

Hint? ||

You find one employee's remote controller..

It seems to be broken and you can't use it to open the gate but you decide to open it to see inside....

Hint? ||





- Module 4 – The challenge
  - Introducing the challenge
  - The awards 😊

The first two to complete the challenge will win a:

RTL-SDR Dongle from <http://www.rtl-sdr.com>



Feedback and questions please..  
Don't be shy.. ;-D



# OPPOSING FORCE

Thank you