

Electronic Access Control Security

Matteo Beccaro || HackInTheBox
Amsterdam, May 27th, 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

Workshop's index of contents

- **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

Workshop's index of contents

- **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.

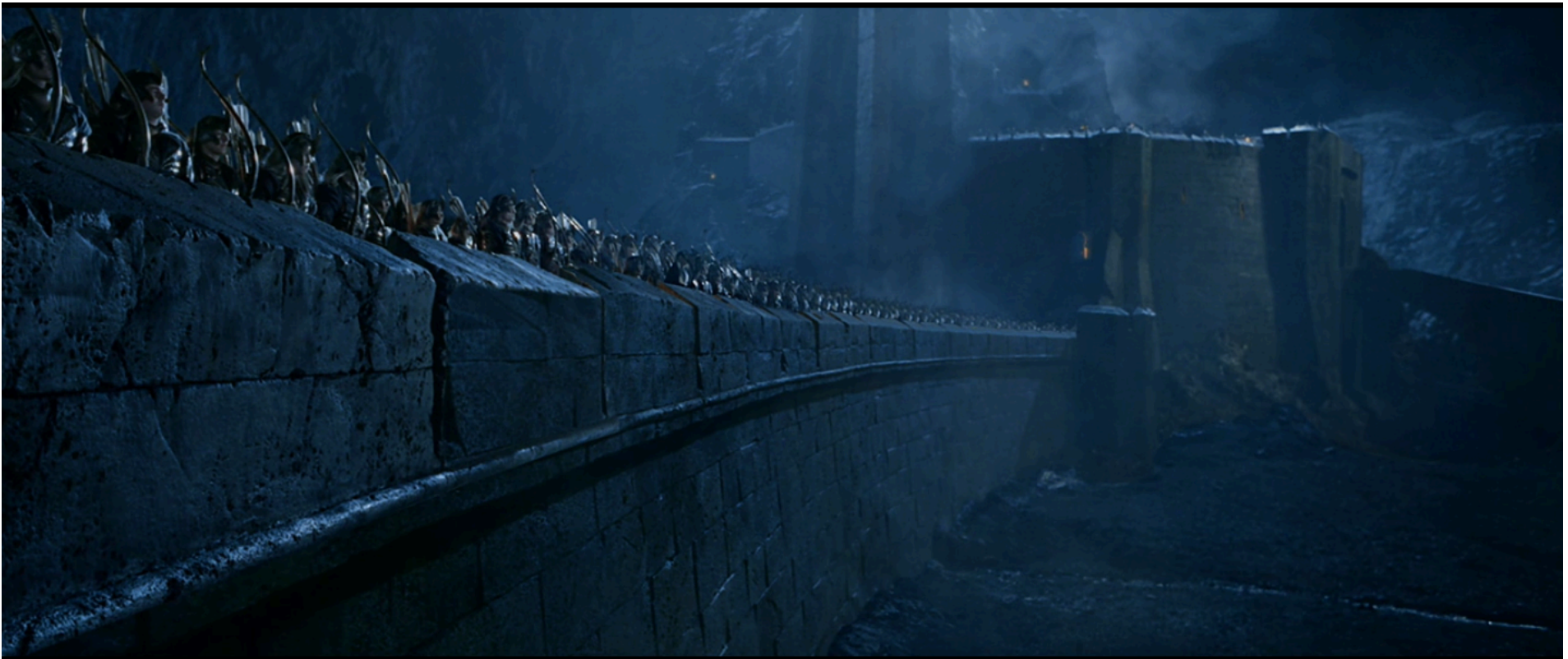
The system is composed by two type of elements:

Human

Technological



- What was an Access Control system?
The technological elements



- What was an Access Control system?

The human elements...

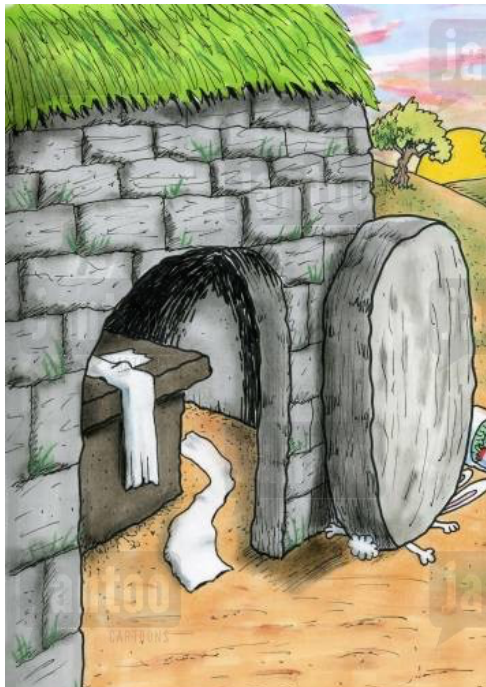


- What was an Access Control system?
...often fail



- 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

MIFARE Classic

- 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.

MIFARE Ultralight

- 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

MIFARE DesFire

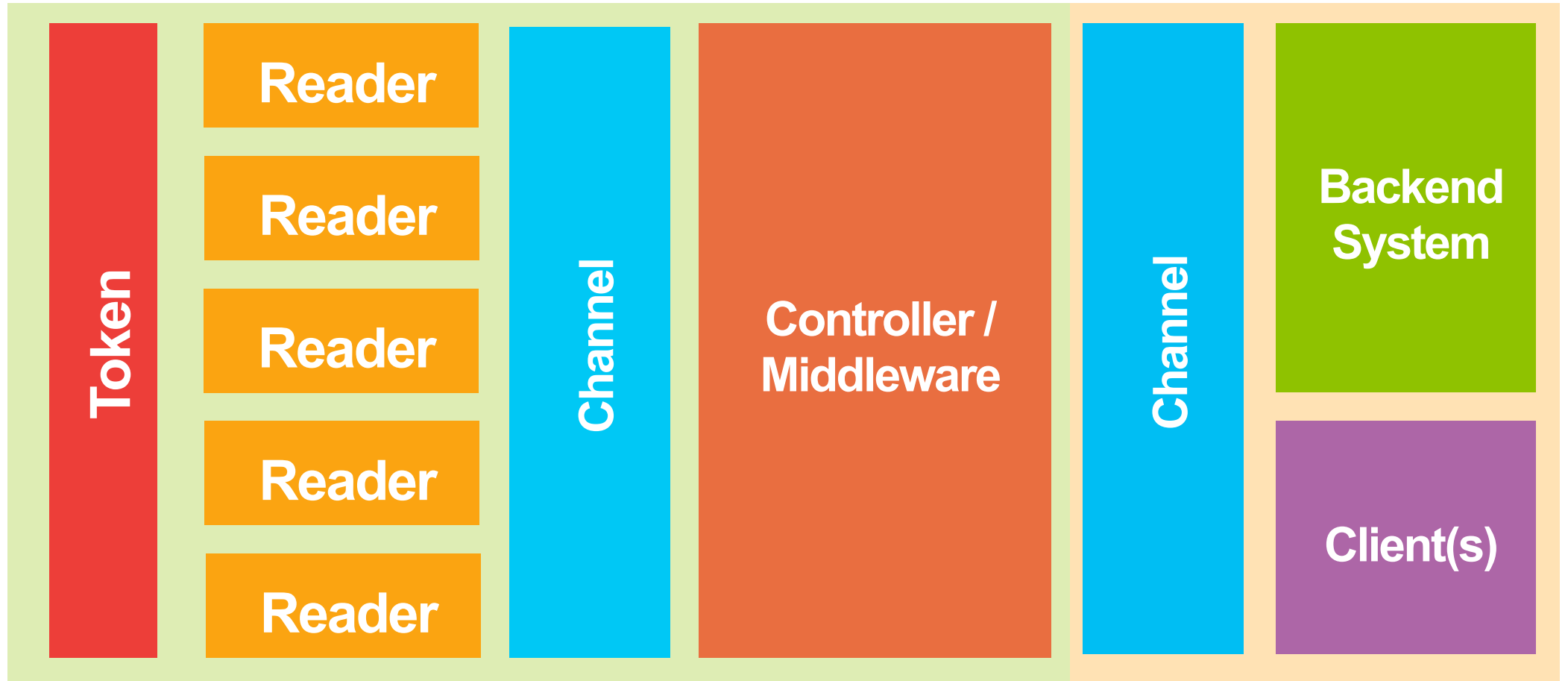
- 2 KB, 4KB or 8 KB memory size
- Advanced security mechanisms (3DES, AES, etc.)
- File system structure is supported
- Several variants are available
 - DESFIRE
 - DESFIRE EV1
 - DESFIRE EV2

- 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

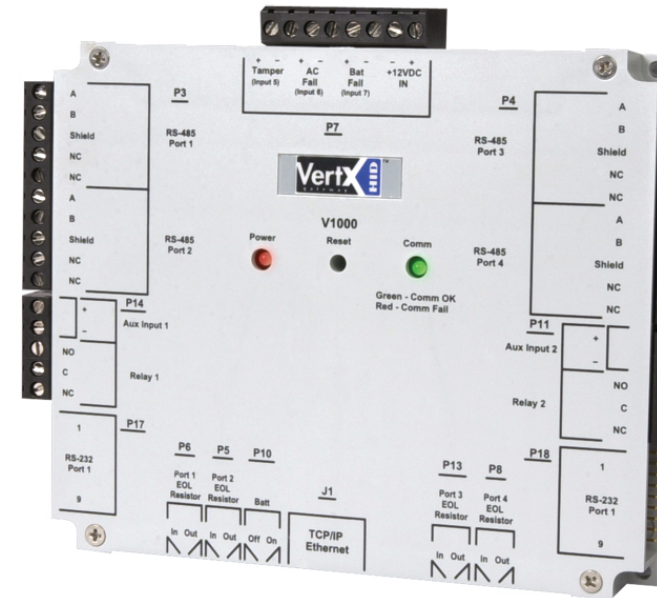
Readers

- 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



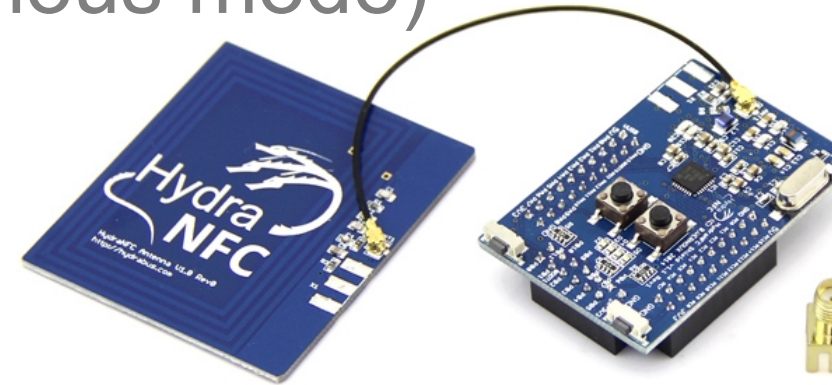
- Module 2 – attacking NFC
 - NFC: what are we talking about?
 - Weapons for NFC-based solutions
 - Penetration test methodology
 - Hands-on
 - Case studies

Tools of the trade

- HydraNFC
- ProxMark3
- ChameleonMini
- NFCuIT

HydraNFC

- 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



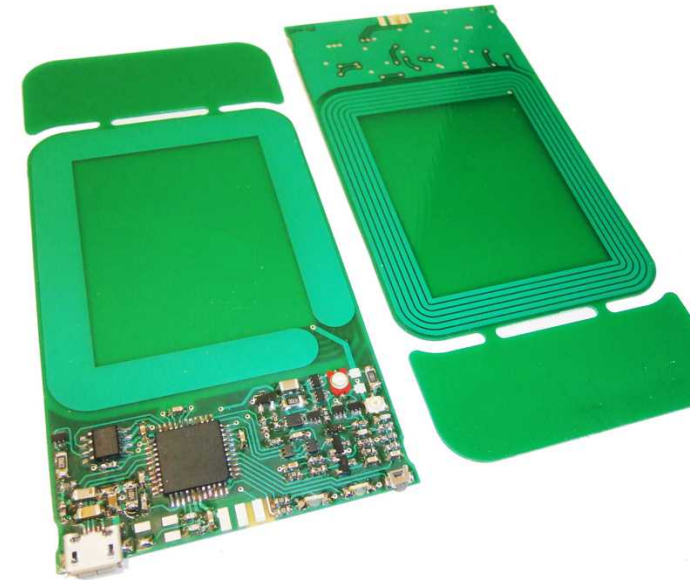
ProxMark3

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



ChameleonMini

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

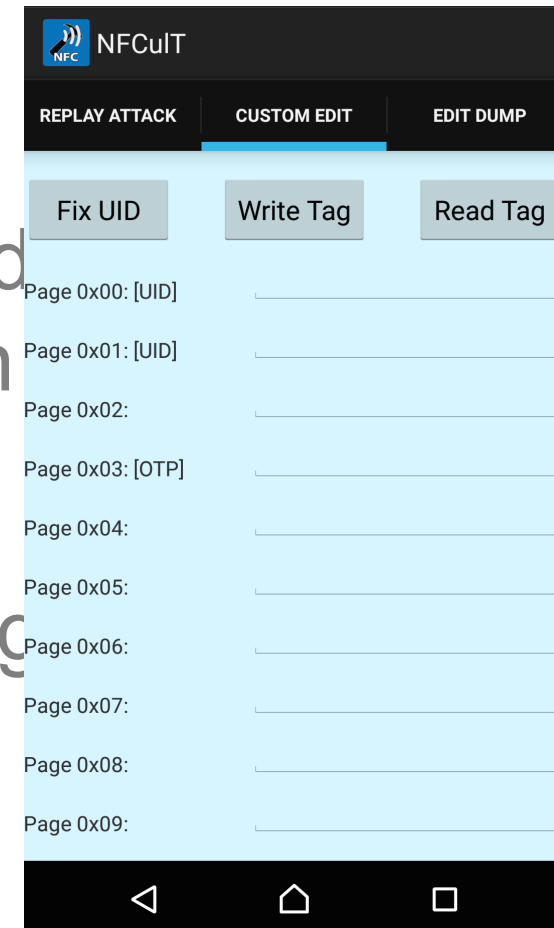


Opposing Force own weapon

- 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**

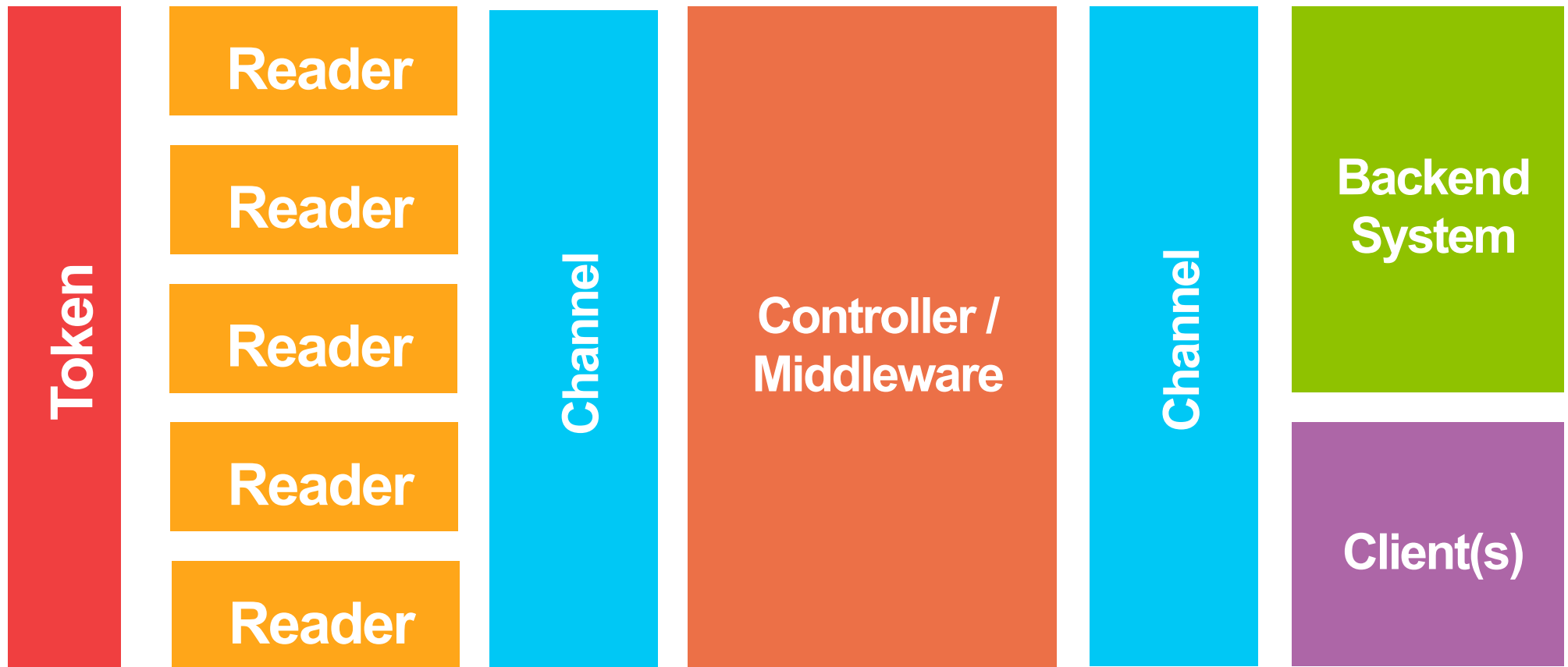
The custom editing feature

- 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

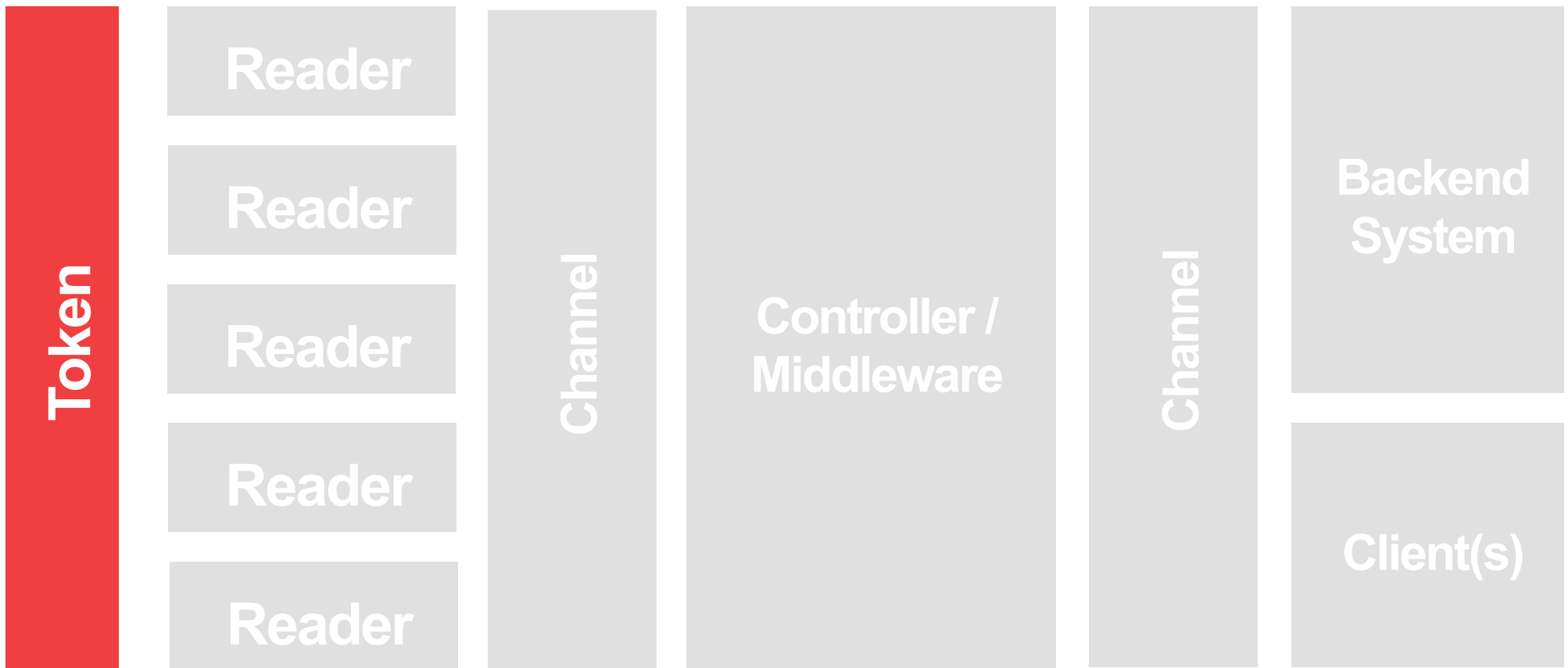


- Module 2 – Attacking NFC
 - NFC: what are we talking about?
 - Weapons for NFC-based solutions
 - Penetration test methodology
 - Hands-on
 - Case studies

Access Control system attack surface

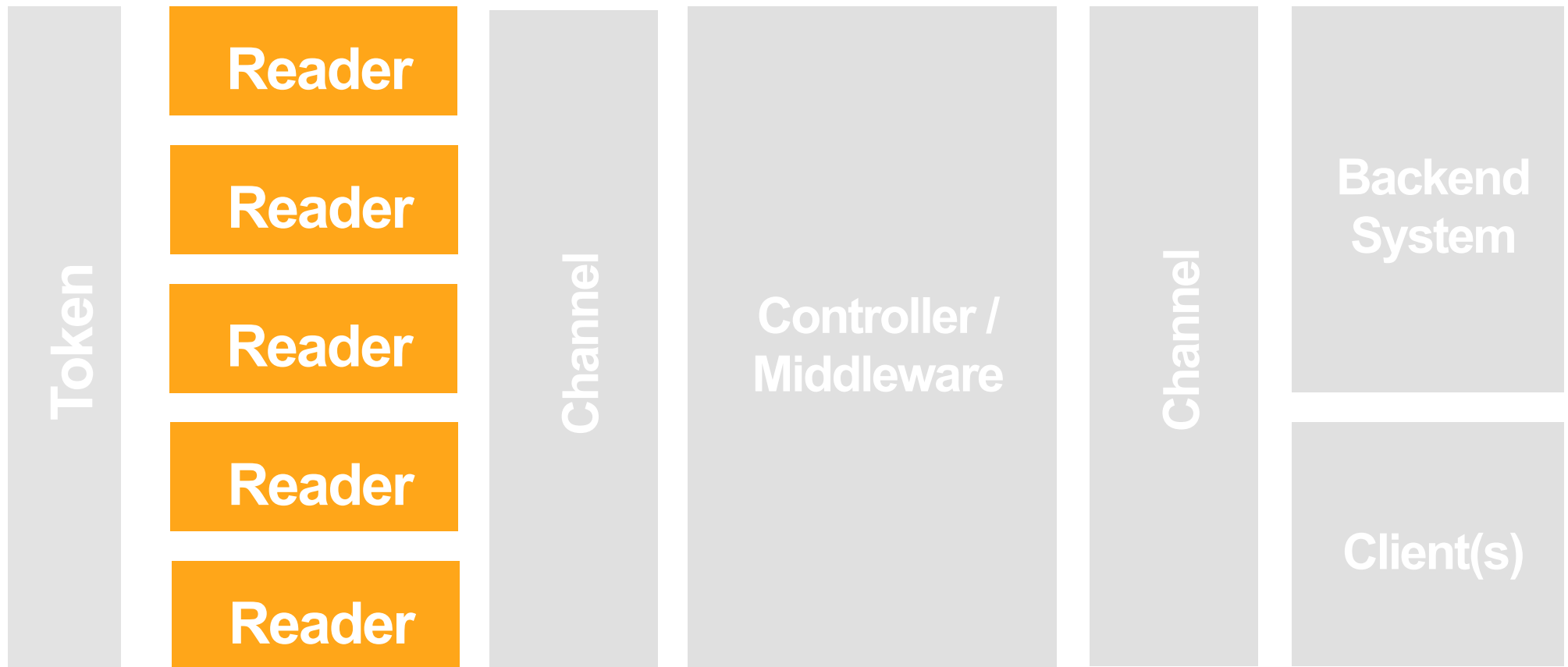


Access Control system attack surface



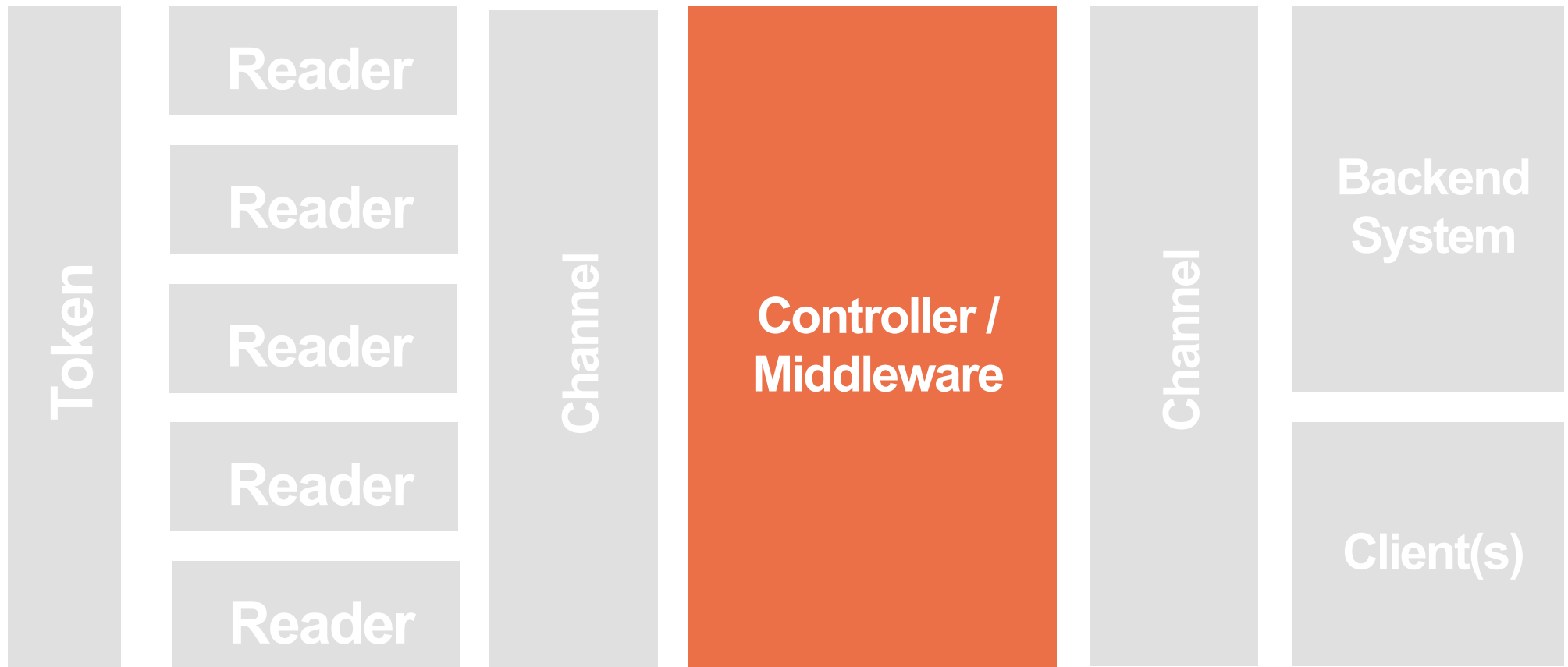
Attack Surface	Attacks to Perform	Impact
NFC Interface	Analyze the authentication mechanisms	Secrets extraction, MiTM attacks
Hardware board	Side channel attacks	Secrets dumping or guessing
Memory	Assess logic vulnerabilities in the implementation	Bypass security mechanisms

Access Control system attack surface



Attack Surface	Attacks to Perform	Impact
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

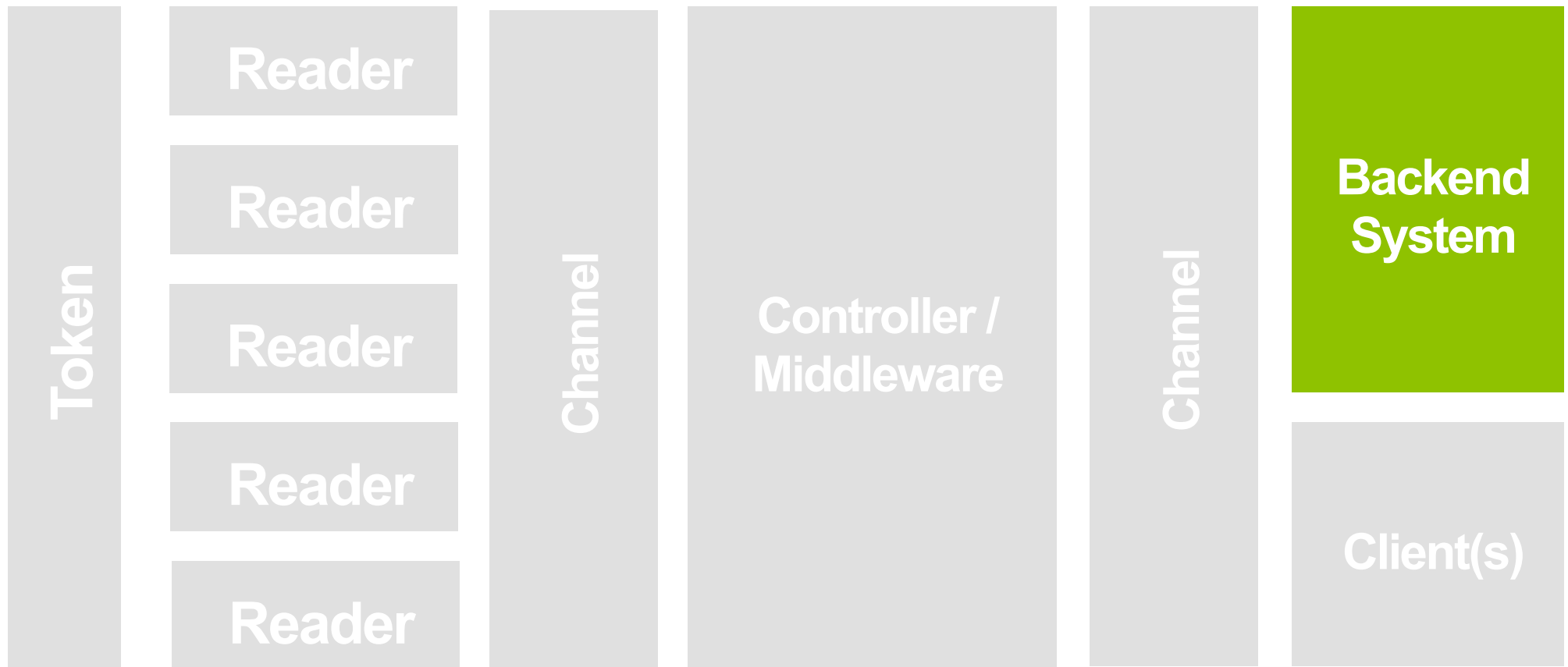
Access Control system attack surface



The controller

Attack Surface	Attacks to Perform	Impact
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

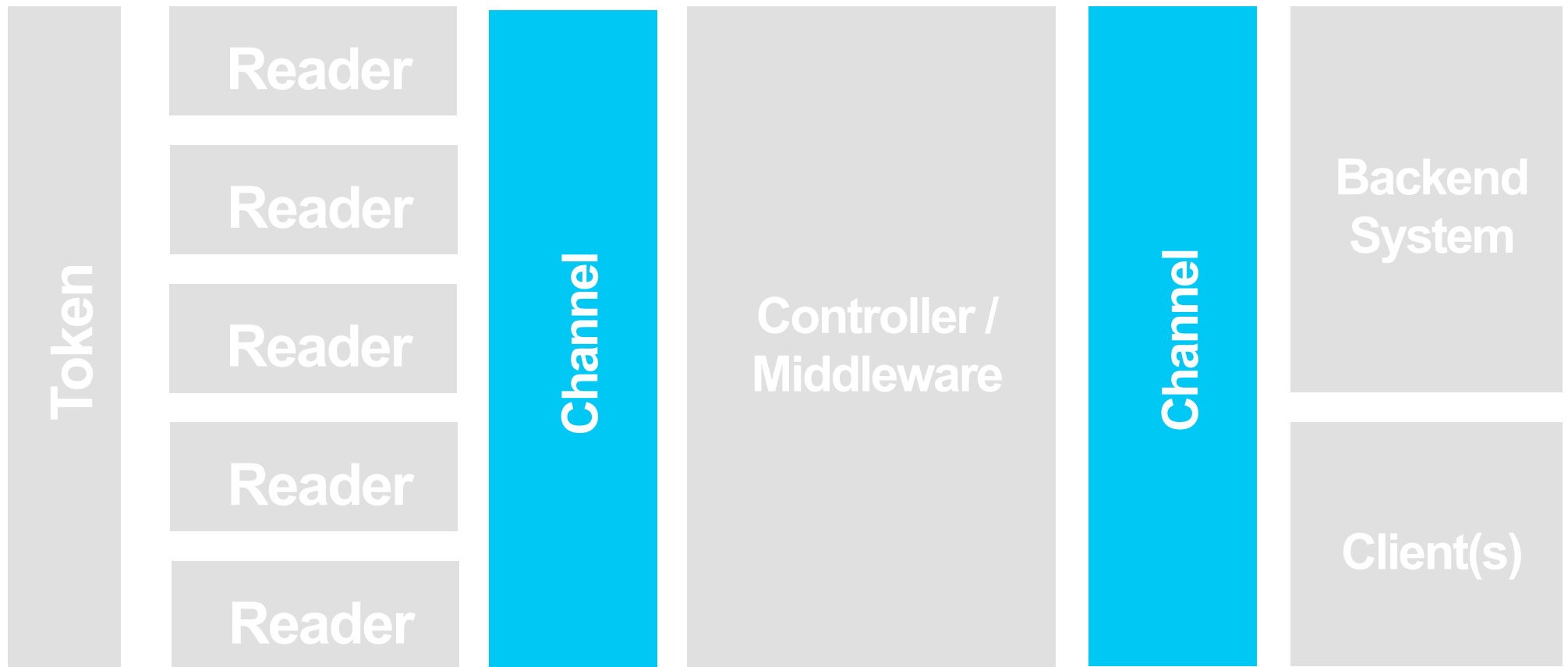


The backend

Attack Surface	Attacks to Perform	Impact
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



The channels

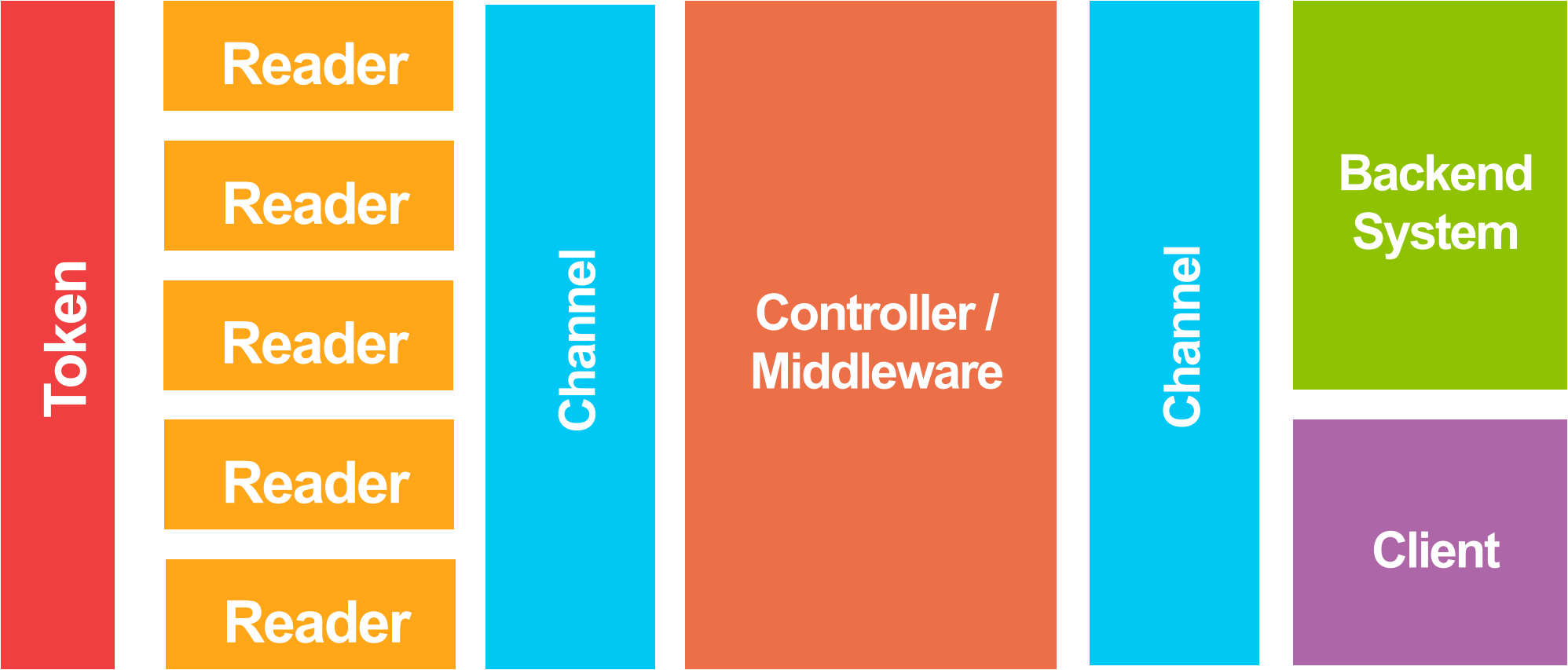
Attack Surface	Attacks to Perform	Impact
Hardware board	Identify forgotten or backdoor pins	Data exfiltration, firmware dumping
External wires	Try to intercept data passing through those wires	Intercepting sensitive information
Wireless connection	Intercept and inject data	Intercepting sensitive information,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

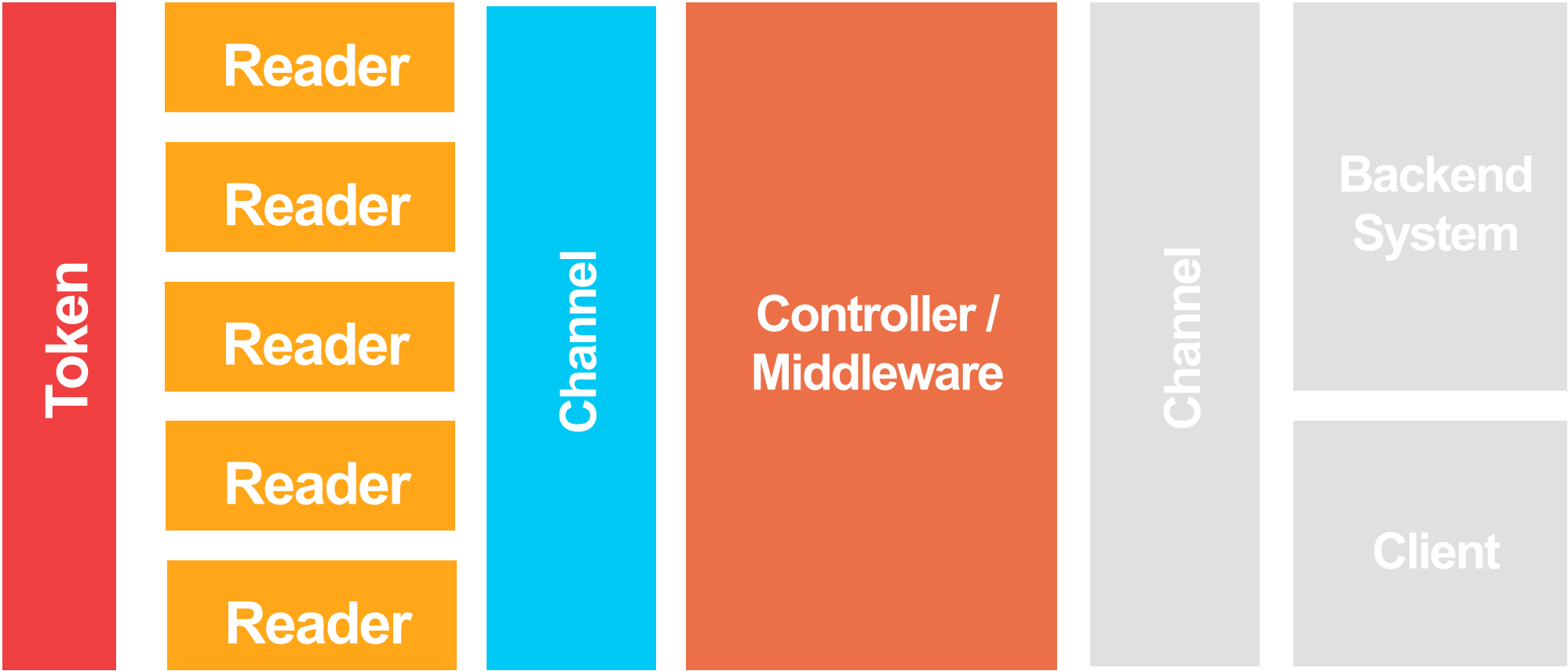
Fire up your 

- 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

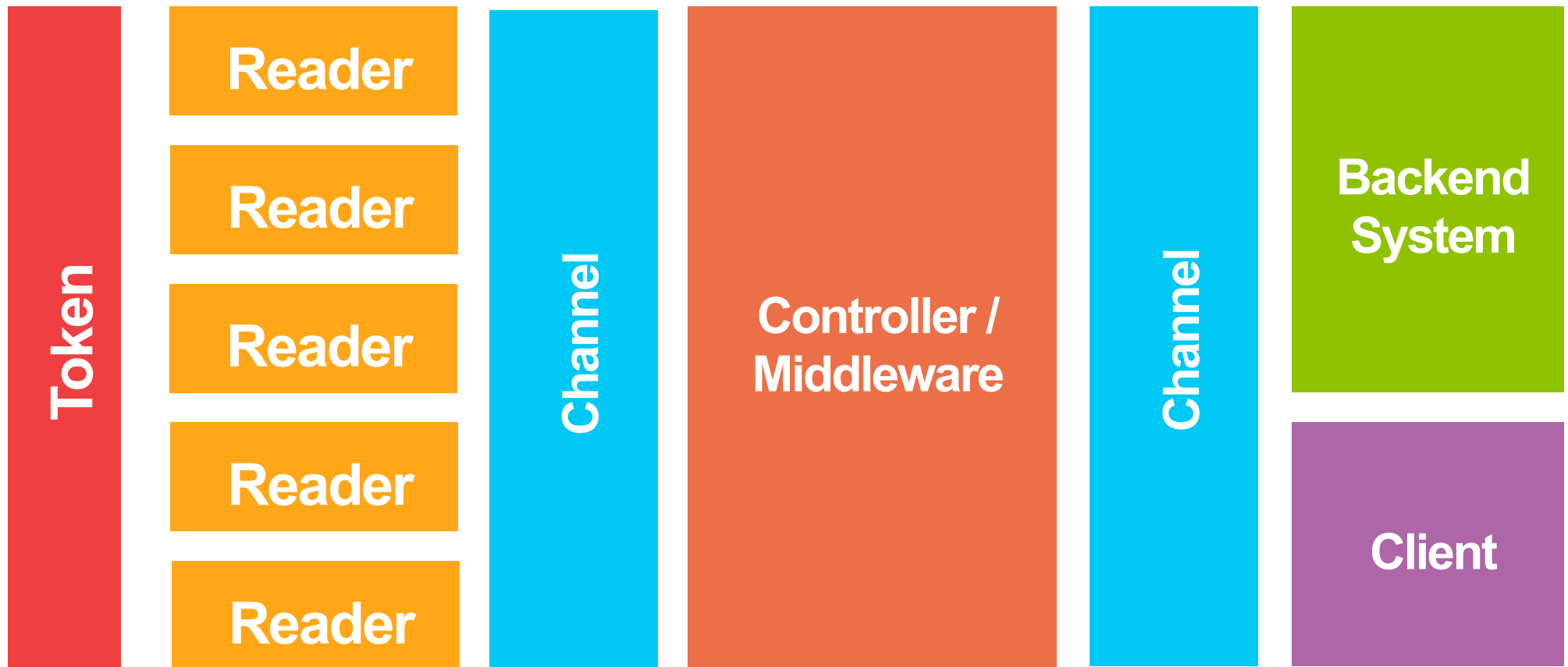
04FBC7B0	2AC13180	° «∞* 1Ä
5A48F203	FFFFFFFF	ZHÚ v v v v
01050000	020102BD	Ω
484A4000	00AE10A0	HJ@ Æ †
A0000473	8A84035D	† sãÑ]
51432E00	04F80000	QC. -
51432E00	001D0004	QC.
F8AE10A0	140249E5	- Æ † IÂ

Absence of a UID blacklist in the backend

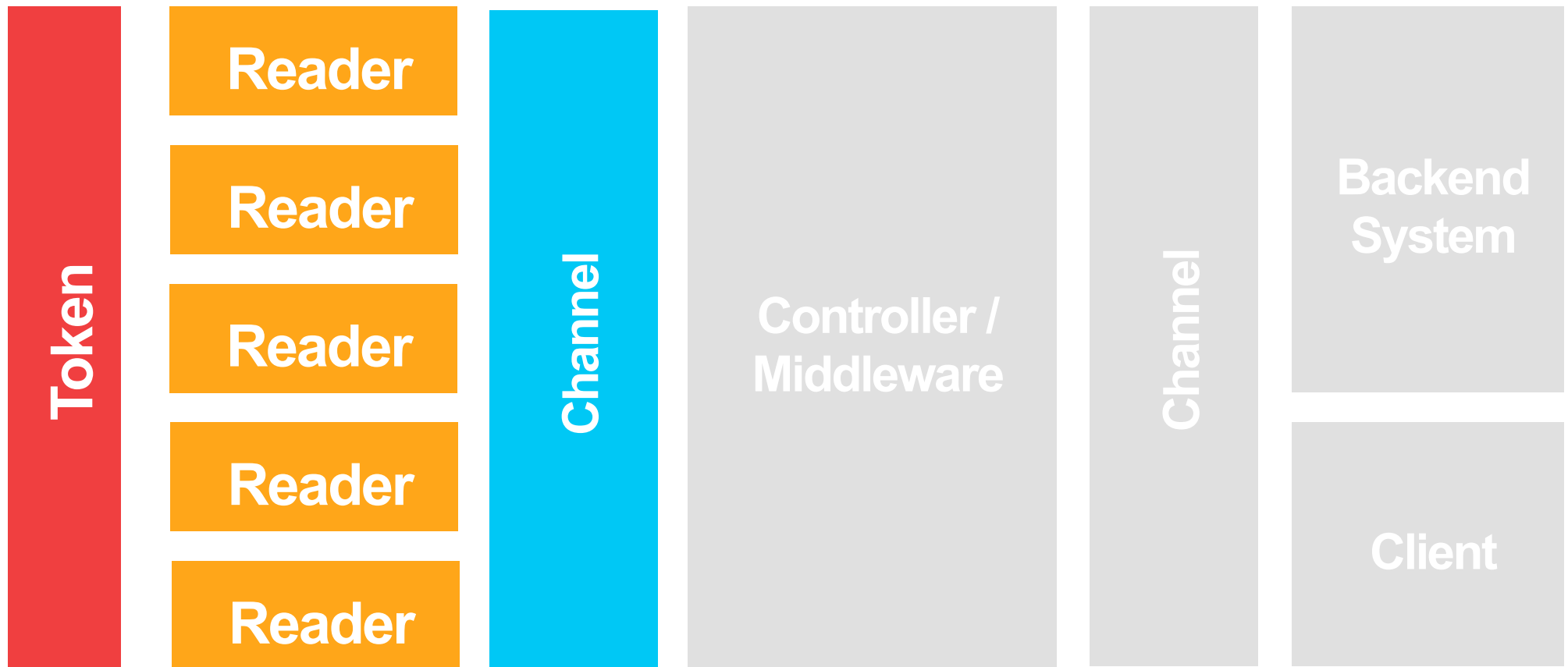
Lock bit for the OTP sector is not checked by the stamping machine

Timestamps are not encrypted nor signed

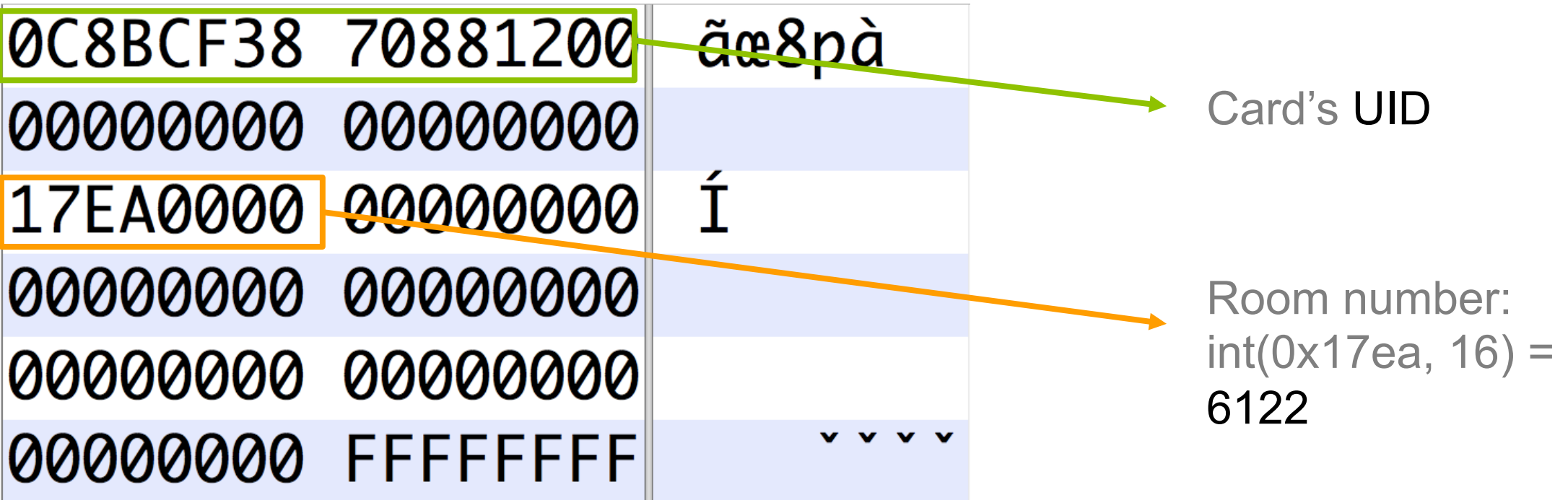
MIFARE Classic hotel door lock



MIFARE Classic hotel door lock



MIFARE Classic door lock

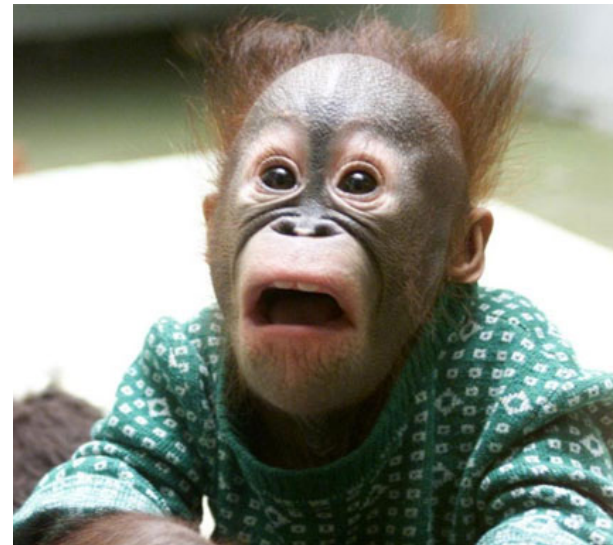


Module 3 || attacking RF communication

- 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~~



Radio Frequency and EAC Systems

- 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

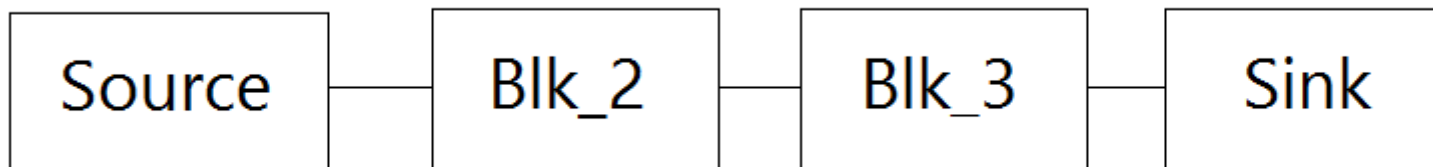
Three SDR-compatible devices

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

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 component library on the right side of the interface includes the following categories:

- [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]
- [Level Controllers]
- [Math Operators]
- [Measurement Tools]
- [Message Tools]
- [Misc]
- [Modulators]
- [Networking Tools]
- [NOAA]
- [OFDM]

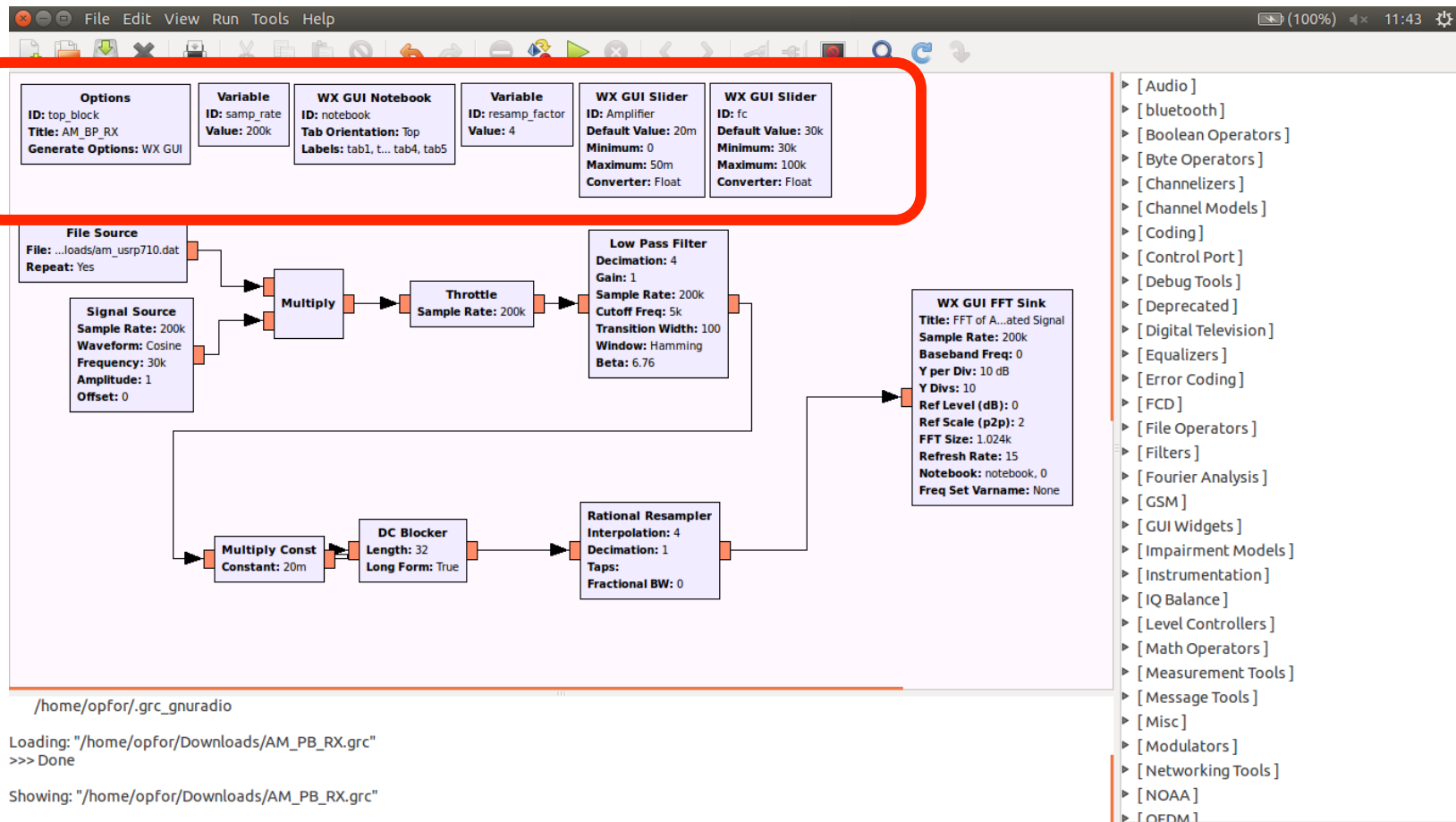
The terminal at the bottom shows the following output:

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

Exploring Radio Frequency communication

■ GRC Interface

VARIABLE



Exploring Radio Frequency communication

■ GRC Interface

The screenshot displays the GNU Radio Companion (GRC) interface. At the top, there is a menu bar (File, Edit, View, Run, Tools, Help) and a toolbar. Below the toolbar, several control panels are visible, including Options, Variable, WX GUI Notebook, and WX GUI Sliders. The main workspace contains a signal flow graph (SFG) with the following components:

- File Source**: File: ...loads/am_usrp710.dat, Repeat: Yes
- Signal Source**: Sample Rate: 200k, Waveform: Cosine, Frequency: 30k, Amplitude: 1, Offset: 0
- Multiply**: Receives input from both File Source and Signal Source.
- Throttle**: Sample Rate: 200k
- Low Pass Filter**: Decimation: 4, Gain: 1, Sample Rate: 200k, Cutoff Freq: 5k, Transition Width: 100, Window: Hamming, Beta: 6.76
- 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
- Multiply Const**: Constant: 20m
- DC Blocker**: Length: 32, Long Form: True
- Rational Resampler**: Interpolation: 4, Decimation: 1, Taps: Fractional BW: 0

A red box highlights the main processing chain from the Multiply block through the Throttle, Low Pass Filter, and Rational Resampler blocks. A red arrow labeled "WGRAP" points to this highlighted area. The bottom of the window shows the terminal output:

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

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:

- File Source**: File: ...loads/am_usrp710.dat, Repeat: Yes
- Signal Source**: Sample Rate: 200k, Waveform: Cosine, Frequency: 30k, Amplitude: 1, Offset: 0
- Multiply**: Receives input from File Source and Signal Source.
- Throttle**: Sample Rate: 200k
- Low Pass Filter**: Decimation: 4, Gain: 1, Sample Rate: 200k, Cutoff Freq: 5k, Transition Width: 100, Window: Hamming, Beta: 6.76
- 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
- Multiply Const**: Constant: 20m
- DC Blocker**: Length: 32, Long Form: True
- Rational Resampler**: Interpolation: 4, Decimation: 1, Taps: Fractional BW: 0

At the bottom, a terminal window 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"
```

A red arrow labeled "MINA" points to the terminal window.

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 blocks 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 **Multiply** block is connected to a **Throttle** block (Sample Rate: 200k).
- The **Throttle** block is connected to a **Low Pass Filter** (Decimation: 4, Gain: 1, Sample Rate: 200k, Cutoff Freq: 5k, Transition Width: 100, Window: Hamming, Beta: 6.76).
- The **Low Pass Filter** is connected 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).
- A **Multiply Const** block (Constant: 20m) is connected to a **DC Blocker** (Length: 32, Long Form: True).
- The **DC Blocker** is connected to a **Rational Resampler** (Interpolation: 4, Decimation: 1, Taps: Fractional BW: 0).
- The **Rational Resampler** is connected to the **WX GUI FFT Sink**.

The top panel shows several control blocks:

- Options** (ID: top_block, Title: AM_BP_RX, Generate Options: WX GUI)
- Variable** (ID: samp_rate, Value: 200k)
- WX GUI Notebook** (ID: notebook, Tab Orientation: Top, Labels: tab1, t..., tab4, tab5)
- Variable** (ID: resamp_factor, Value: 4)
- WX GUI Slider** (ID: Amplifier, Default Value: 20m, Minimum: 0, Maximum: 50m, Converter: Float)
- WX GUI Slider** (ID: fc, Default Value: 30k, Minimum: 30k, Maximum: 100k, Converter: Float)

The bottom panel shows the command line interface:

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

On the right side, a red-bordered box highlights the **Block Library**, which contains a list of blocks categorized by function. An arrow points from the text "BLOCK LIBRARY" to this box.

Exploring Radio Frequency communication

■ “Hello World” in GNU Radio

The screenshot shows the GNU Radio Companion (GRC) interface for a flow graph named 'hello_world.grc'. The window title is 'hello_world.grc - /home/opfor - GNU Radio Companion'. The interface includes a toolbar with various icons for file operations, execution, and navigation.

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

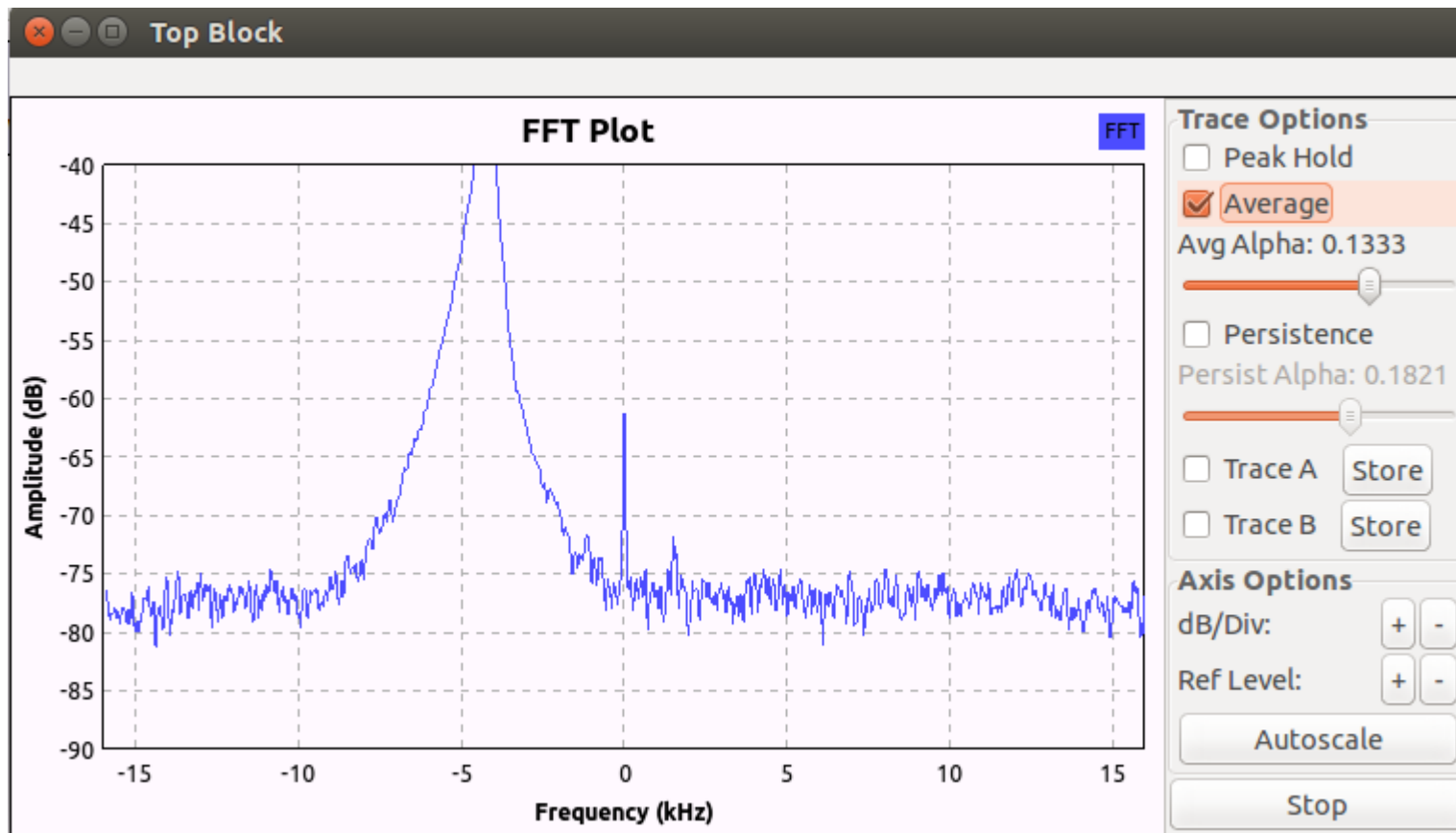
The flow graph shows a signal path from the RTL-SDR Source block to the WX GUI FFT Sink block. A variable block for 'samp_rate' is also present.

Console Output:
Invalid sample rate: 32000 Hz
rtlsdr_read_async returned with -5
>>> Done

Component List:
[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]
[Level Controllers]

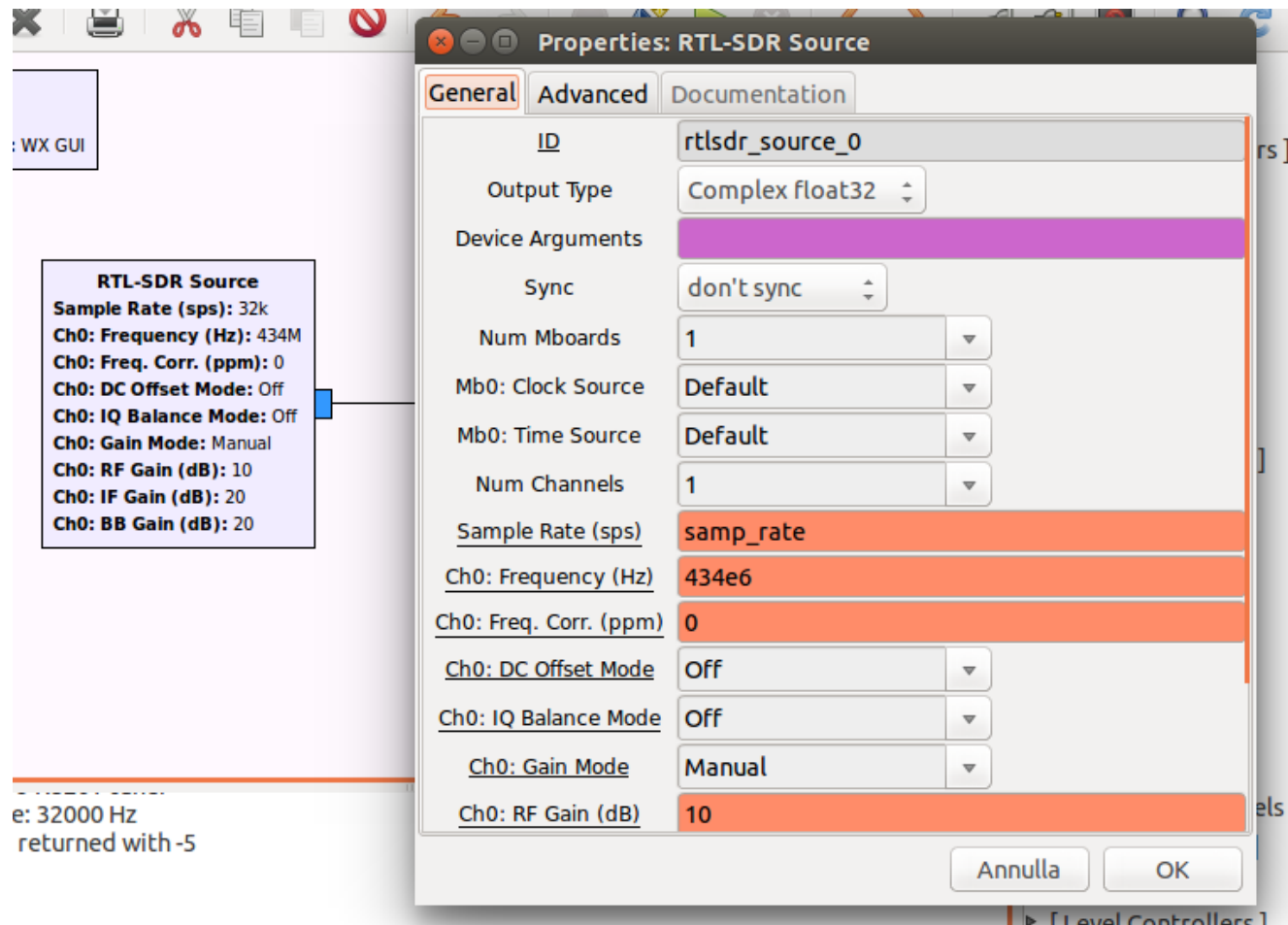
Exploring Radio Frequency communication

- “Hello World” in GNU Radio



Exploring Radio Frequency communication

- RTL-SDR Source Block



Exploring Radio Frequency communication

- WX GUI FFT Sink Block

The image shows the 'Properties: WX GUI FFT Sink' dialog box with the 'General' tab selected. The dialog box contains the following settings:

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	

At the bottom of the dialog box are 'Annulla' and 'OK' buttons.

To the right of the dialog box is a summary box titled 'WX GUI FFT Sink' with the following details:

- 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

SIGINT with GNU Radio

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



SIGINT with GNU Radio

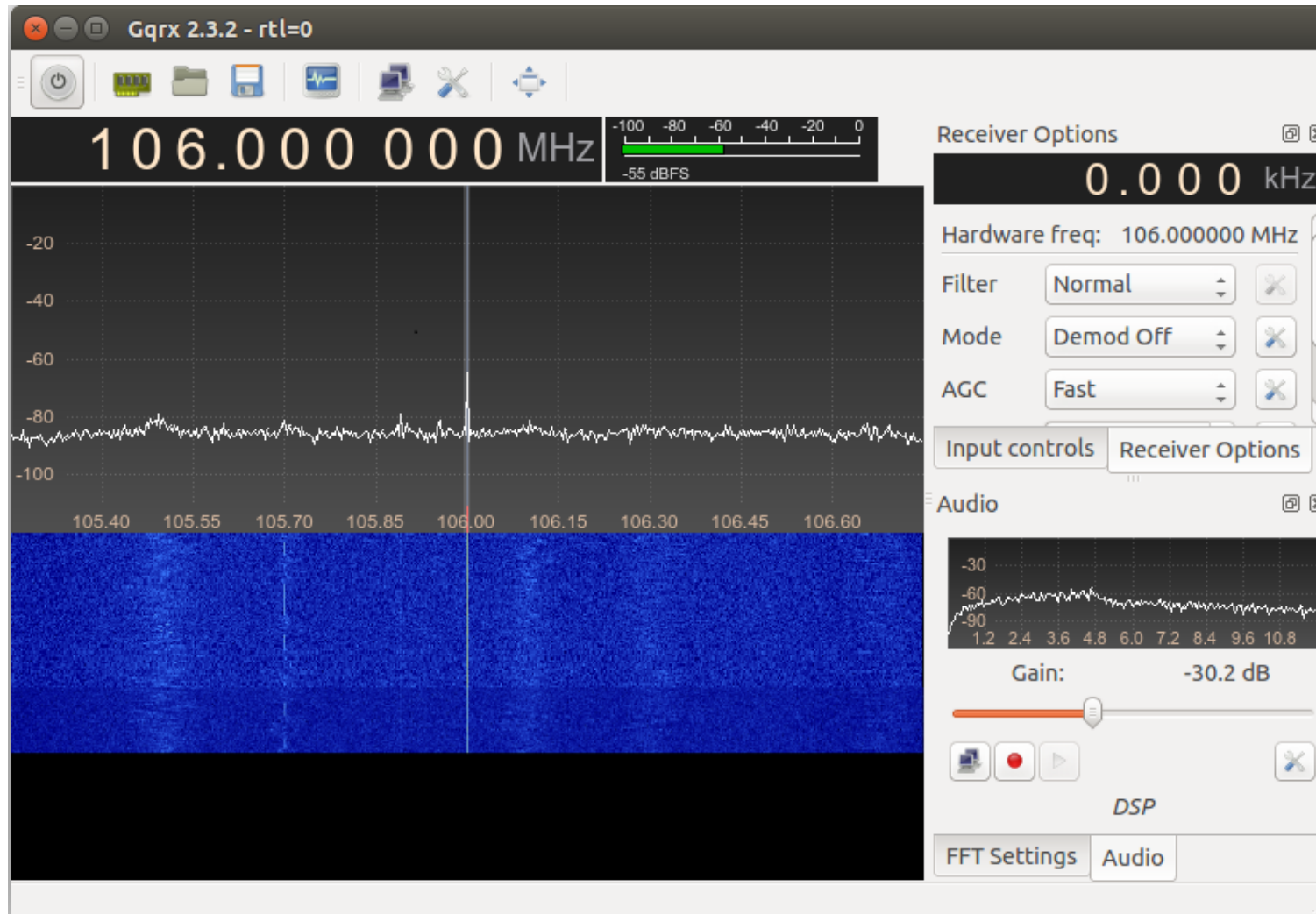
- Define a methodology to study real world signals
- Three main steps



SIGINT with GNU Radio

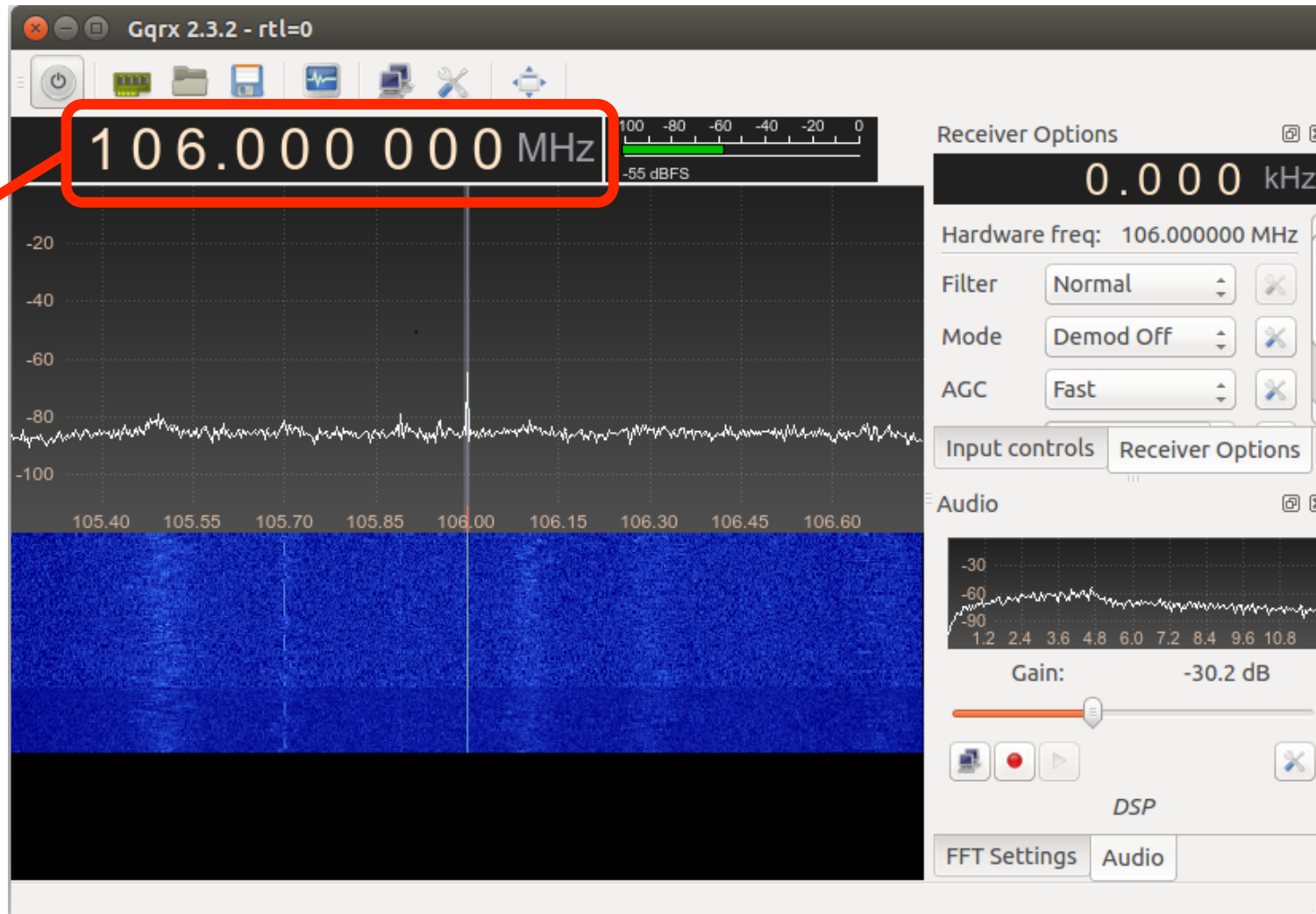
- GQRX
 - SDR receiver and spectrum analyzer based on GNU Radio and QT Graphical toolkit
 - User-friendly interface
 - Supports RTL-SDR, HackRF, USRP and other SDR devices
 - Records signal to WAV file

SIGINT with GNU Radio



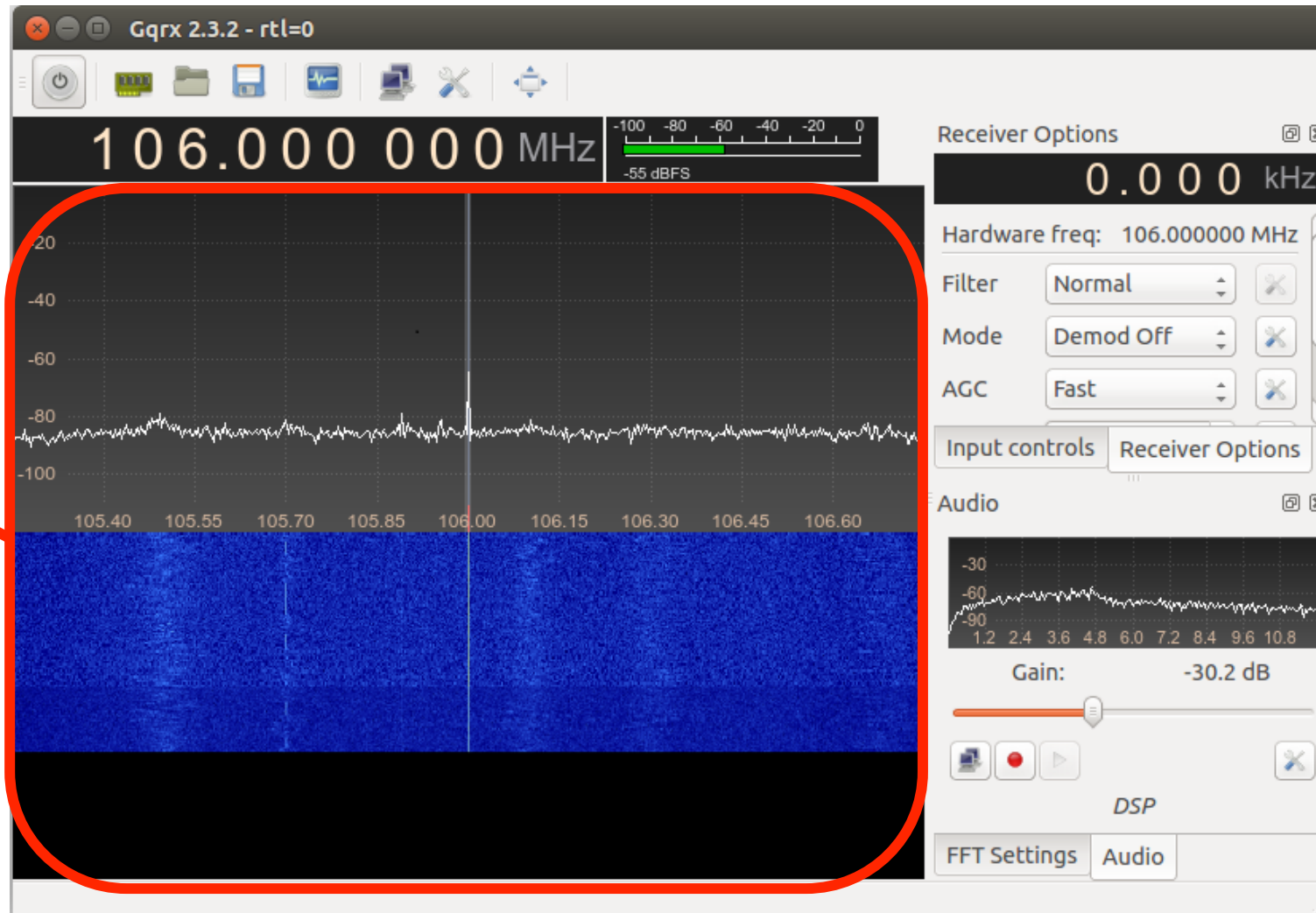
SIGINT with GNU Radio

EQUENC
LECTOR

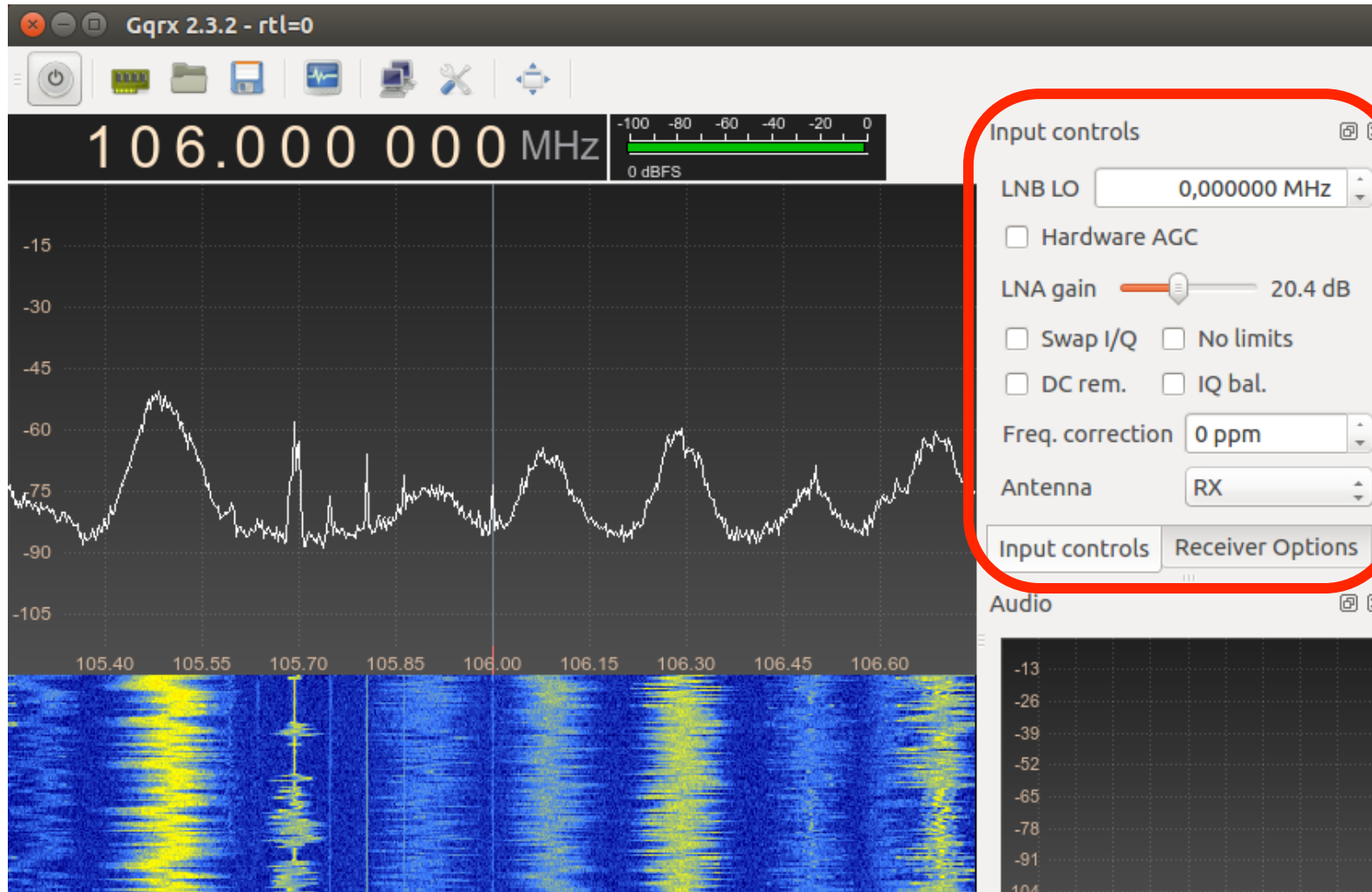


SIGINT with GNU Radio

REAL-TIME
SPECTRUM

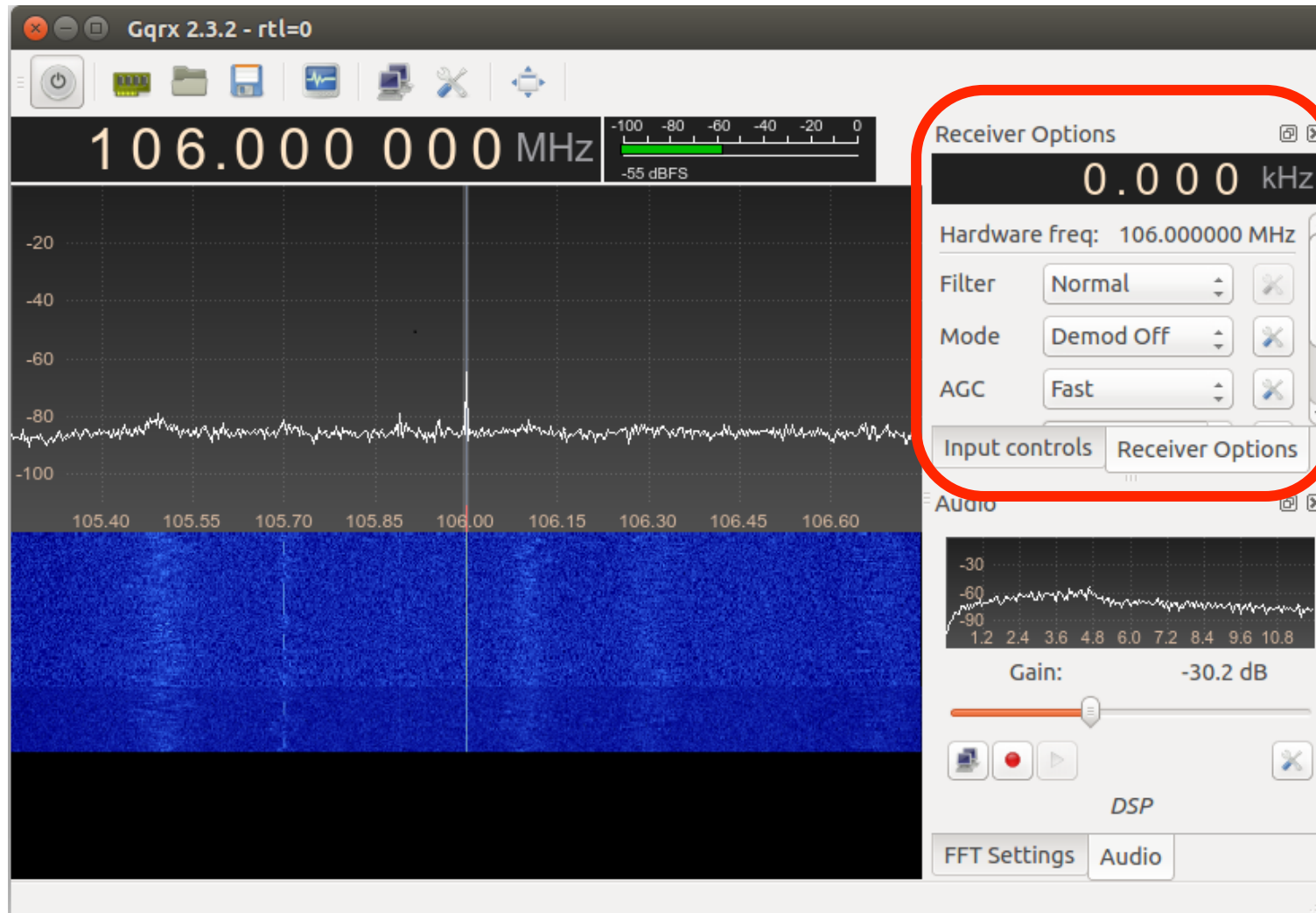


SIGINT with GNU Radio



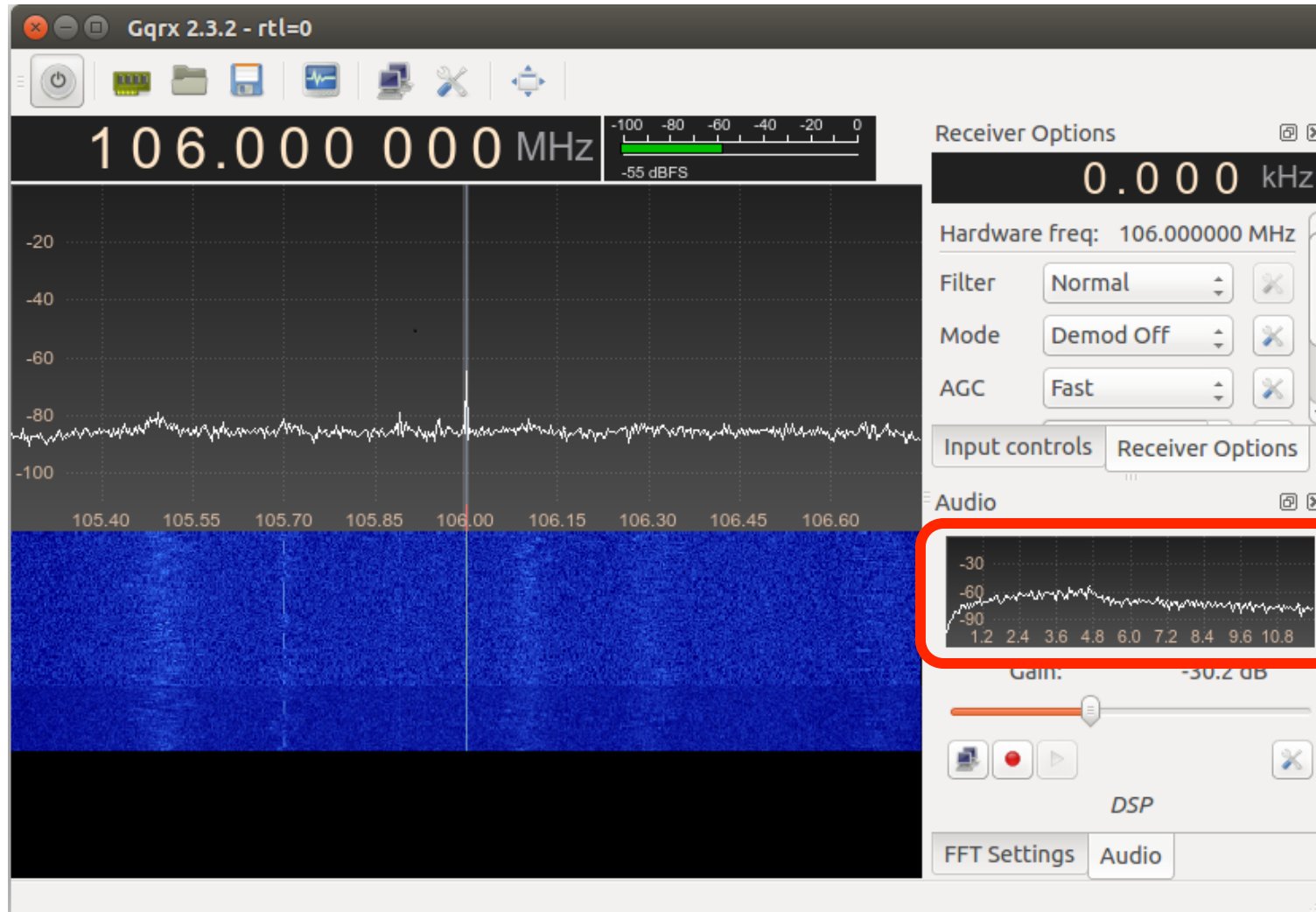
INPUT
CONTROLS

SIGINT with GNU Radio



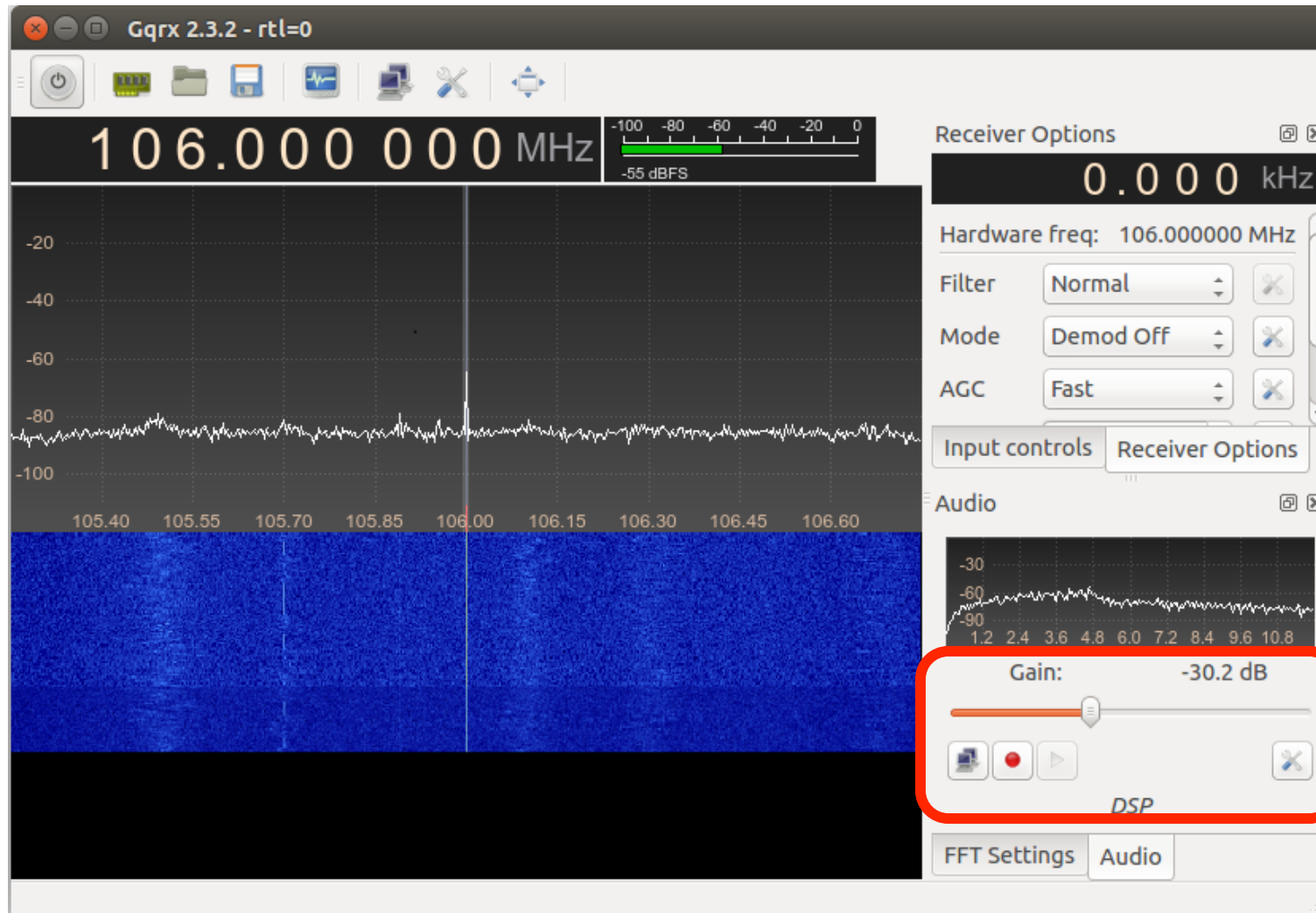
RECEIVER
OPTIONS

SIGINT with GNU Radio



DEMODULATED SPECTRUM

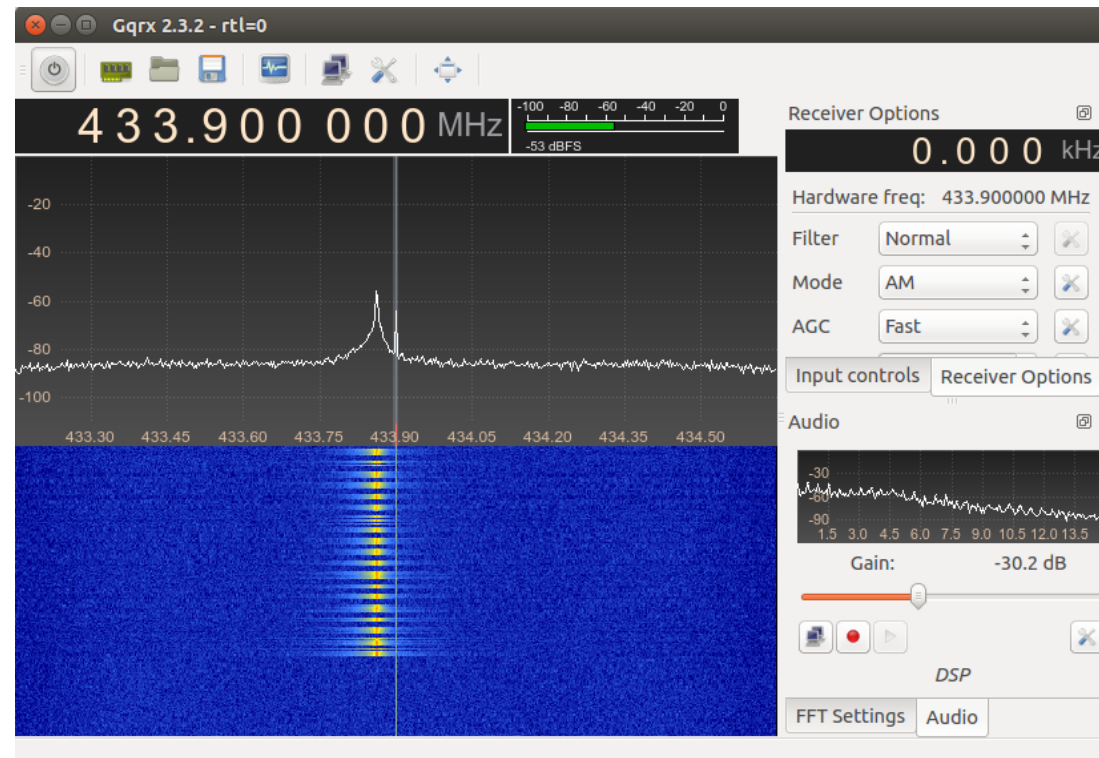
SIGINT with GNU Radio



RECORD
SECTION
N

SIGINT with GNU Radio

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



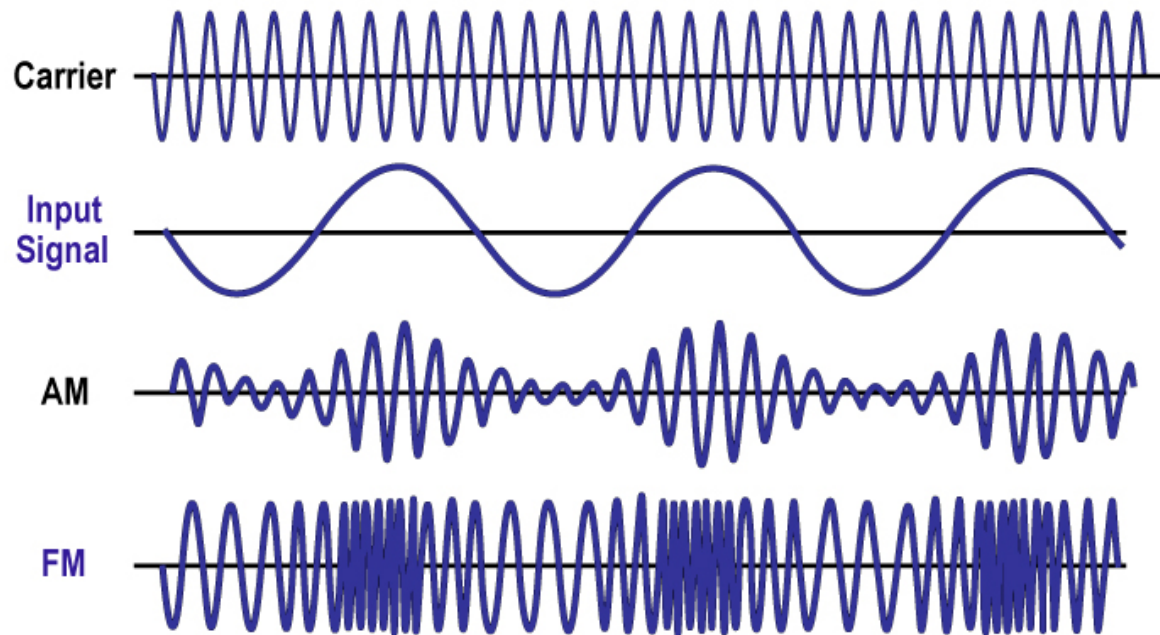
SIGINT with GNU Radio

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



SIGINT with GNU Radio

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



SIGINT with GNU Radio

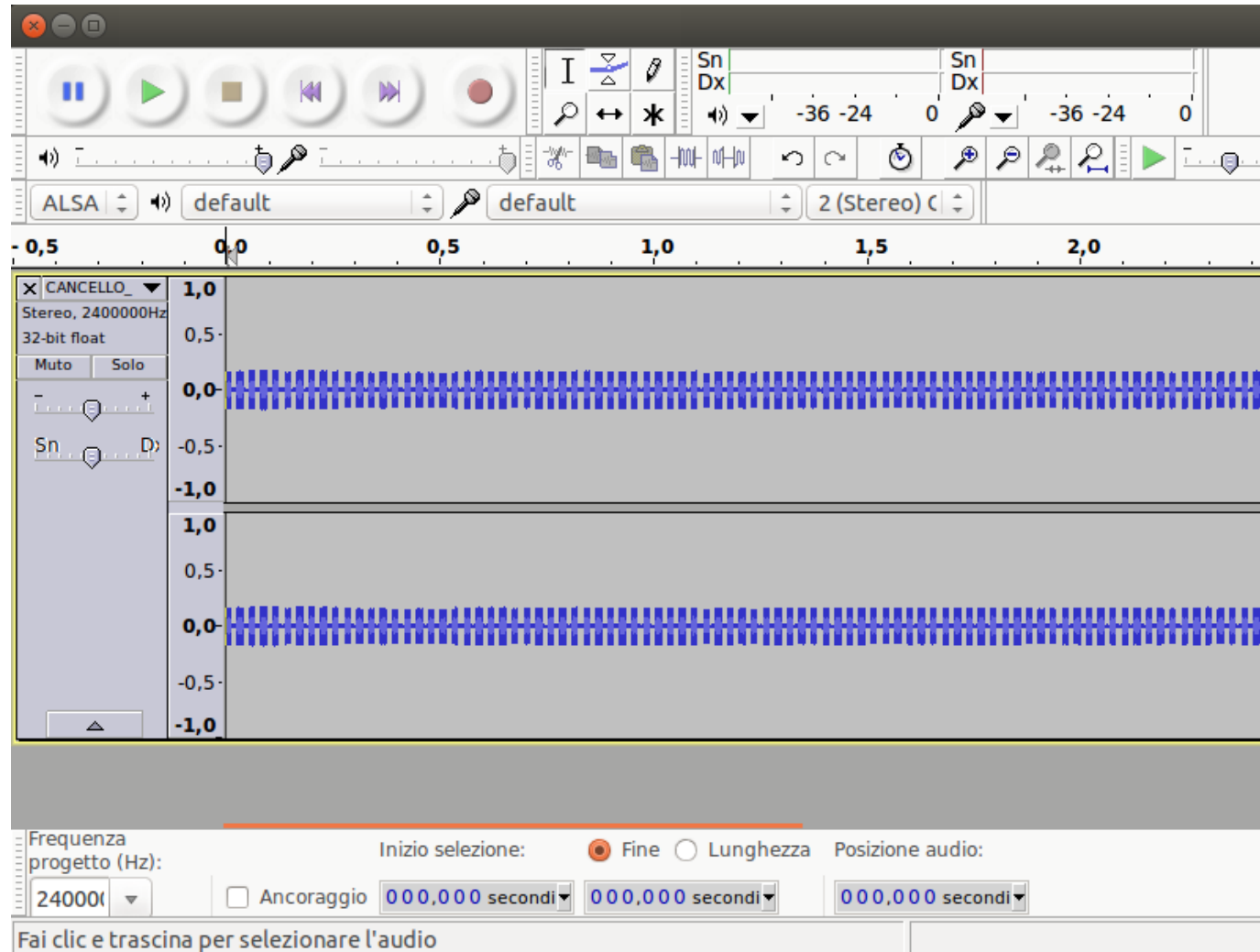
- Signal Identification Guide

Signal Name	Description	Frequency	Modulation	Bandwidth	Location	Waveform	
The Buzzer (MDZhB UZB76)	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	
Tire Pressure Monitoring System (TPMS)	Signal is from a Chrysler TPMS (Tire-Pressure Monitoring System) sensor.	315 MHz — 433 MHz	AM		Worldwide		
Toyota Car Key	Wireless entry rolling code car key.	315 MHz — 433 MHz	AM	40 kHz	Worldwide		

SIGINT with GNU Radio

Audacity

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



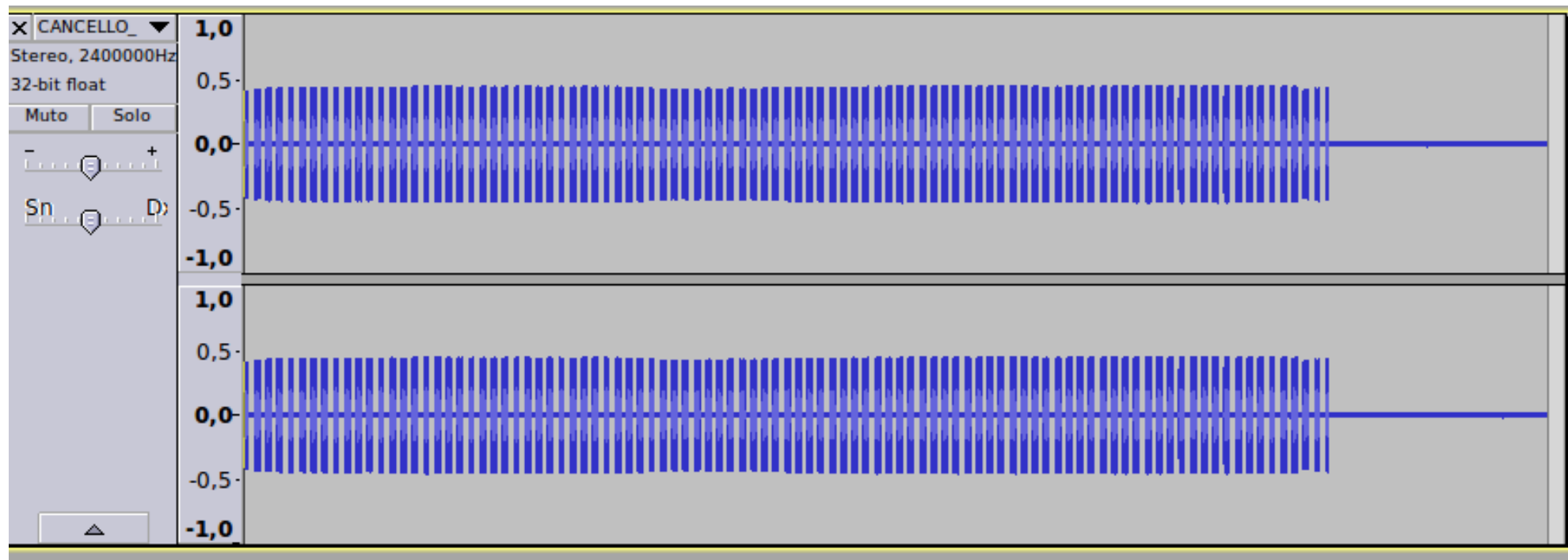
SIGINT with GNU Radio

- Case Study: remote control at 433 MHz



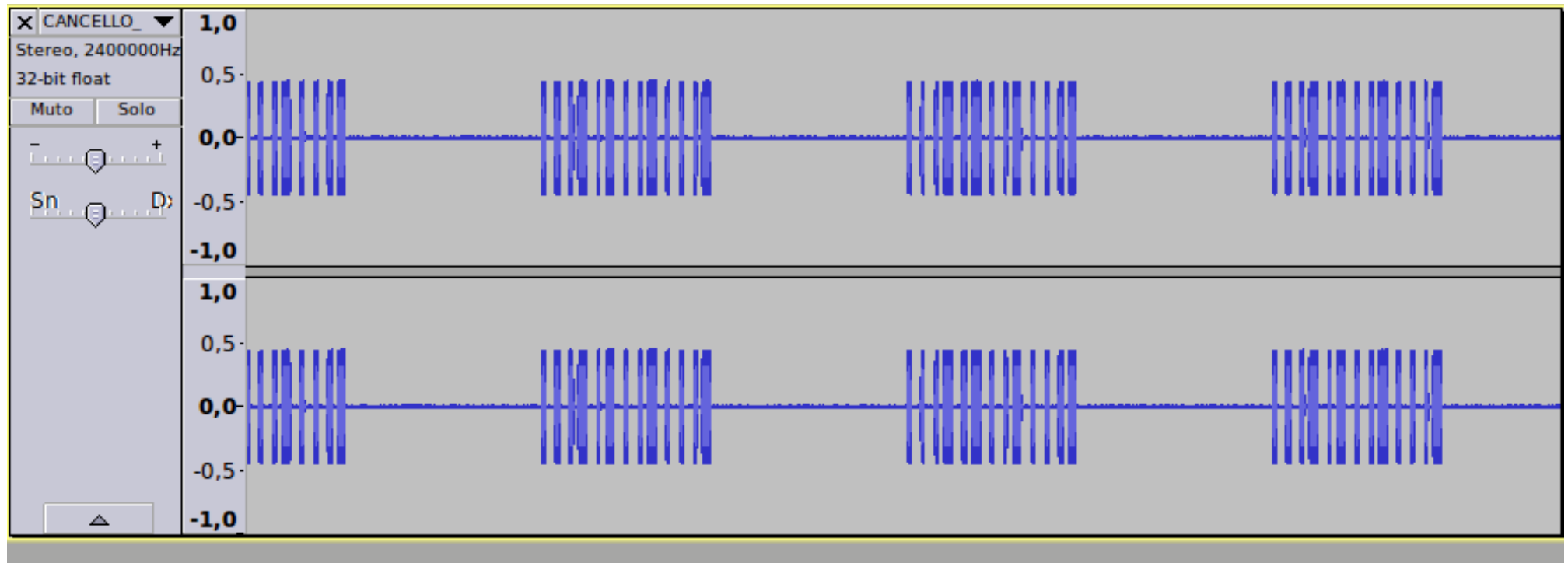
SIGINT with GNU Radio

- Case Study: remote control at 433 MHz



SIGINT with GNU Radio

- Case Study: remote control at 433 MHz



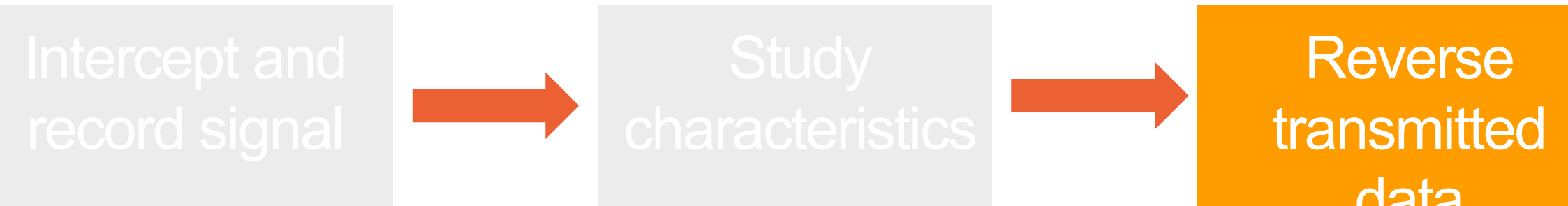
SIGINT with GNU Radio

- 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

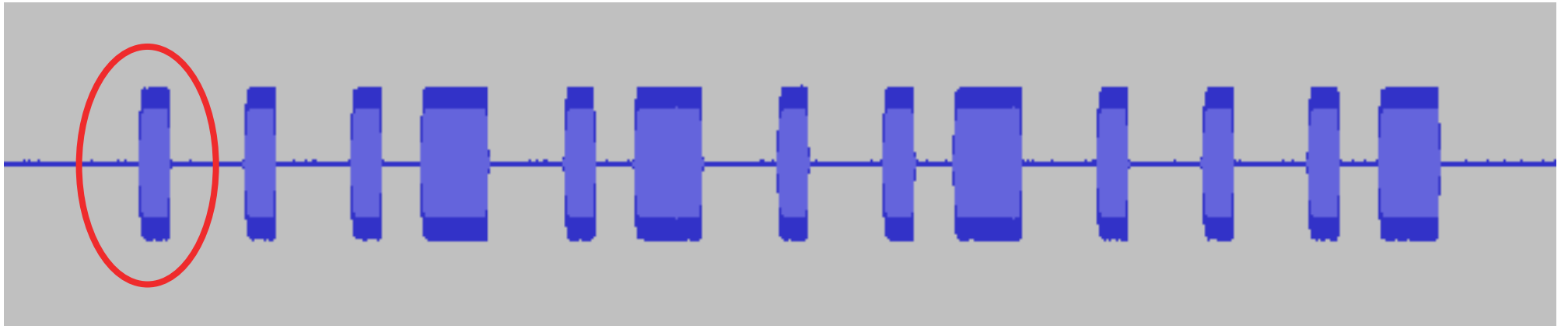
SIGINT with GNU Radio

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



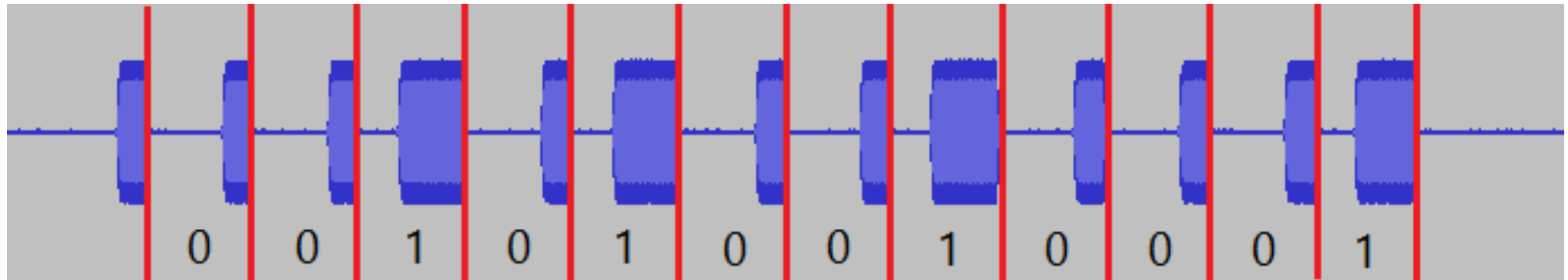
SIGINT with GNU Radio

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



SIGINT with GNU Radio

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

SIGINT with GNU Radio

- 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

SIGINT with GNU Radio

- 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

Agenda

- 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**

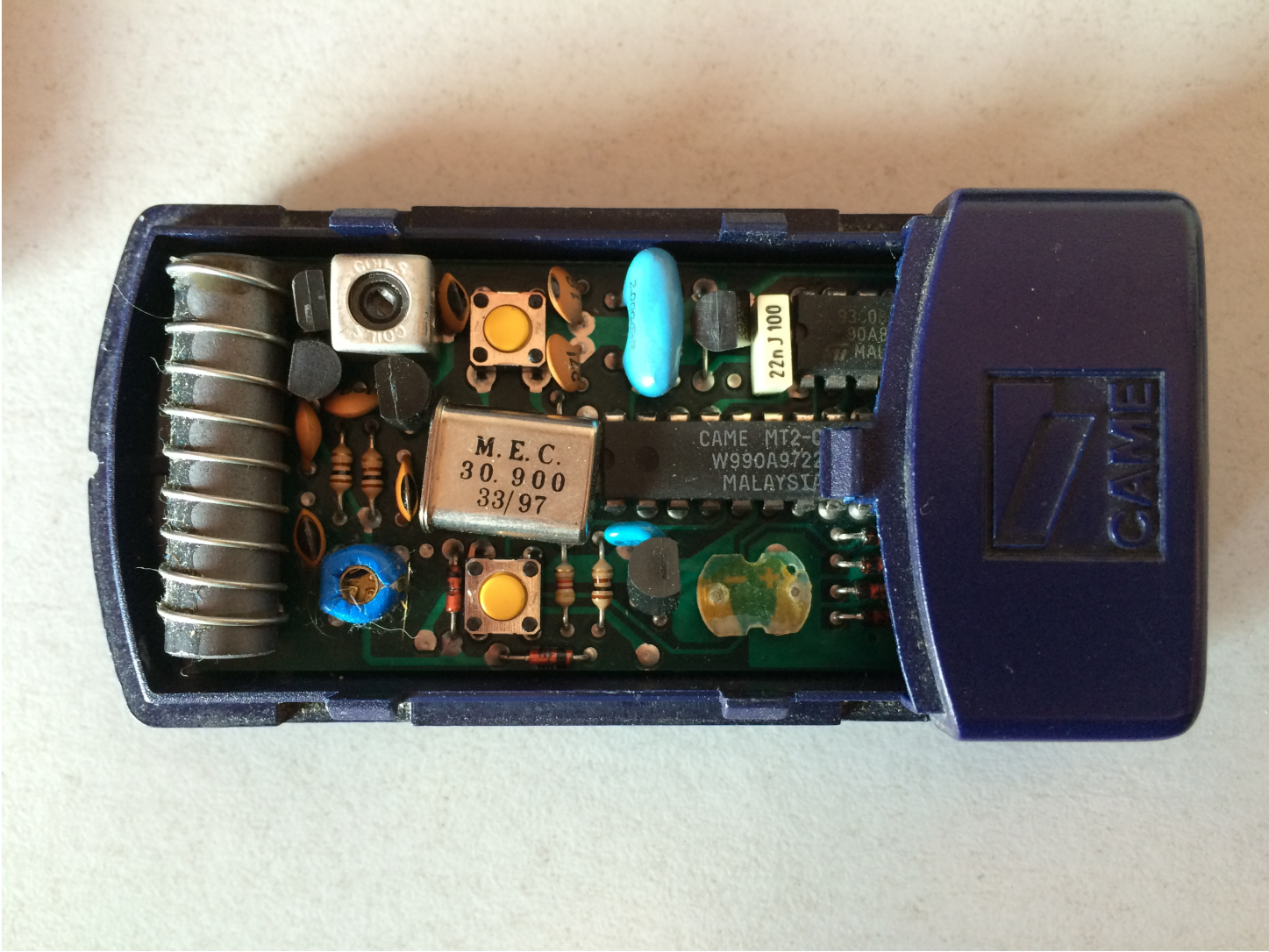
Your task is to open the external facility’s electric gate thus allow your team to enter the facility and proceed with the intrusion..

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?



Agenda

- 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

Contacts – engage@opposingforce.it || www.opposingforce.it ||
[@_opposingforce](https://www.instagram.com/_opposingforce)

Start hack Access Control systems

We have **10 tool box** ~~Now~~ at the special price of **15**

€

- Each toolkit contains
 - Plastic box with Opposing Force sticker 😊
 - 1 HydraBus with its case
 - 1 HydraNFC
 - 1 Mini USB cable
 - 1 SDR-RTL dongle with its antenna
 - 1 Breaboard with some jumpers
 - 1 NFC MIFARE ULTRALIGHT card