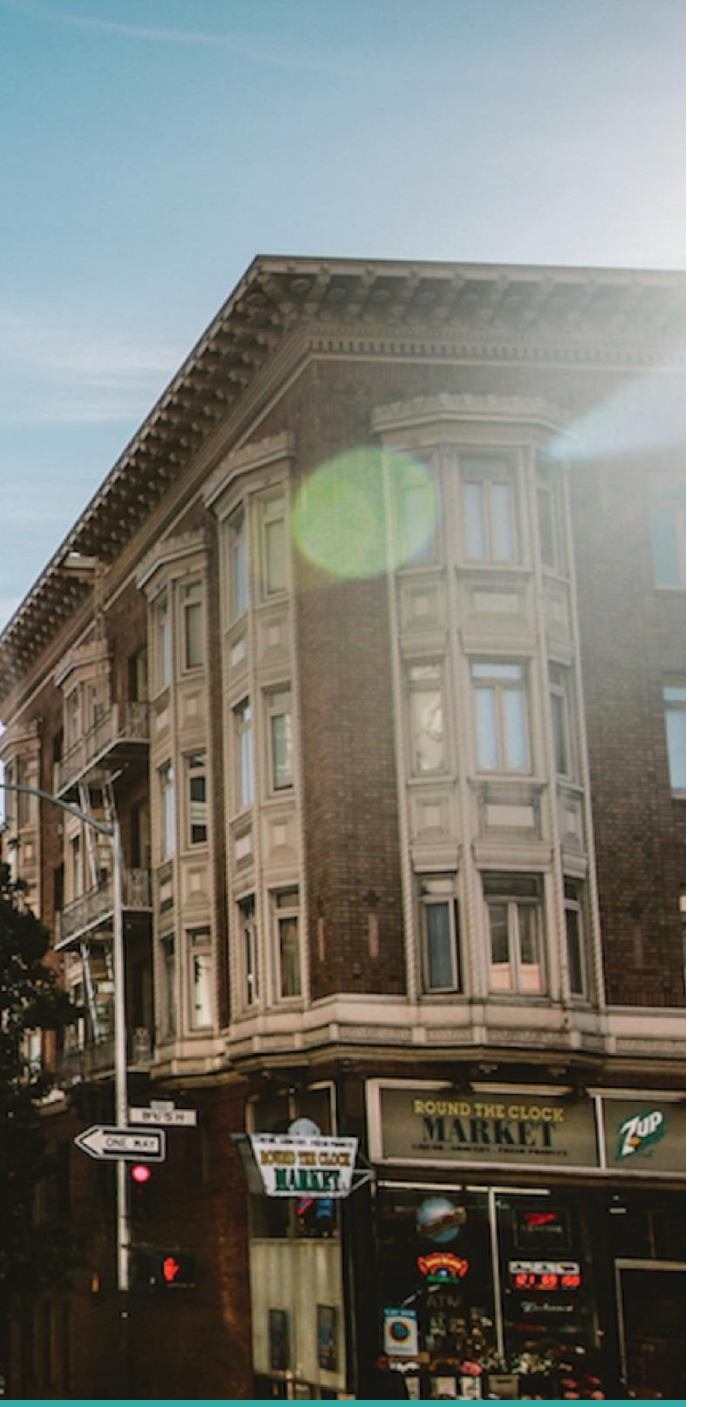
RAPPORT Code radio spatial n°1



23 AOUT

404 CTF 2025

Créé par : JOL

Code radio spatial n°1

Principe de fonctionnement

Résumé succinct

- **Challenge** : décoder une transmission POCSAG depuis un fichier IQ.
- **Méthode** : Gqrx + UDP → pipeline nc | sox | multimon-ng.
- **Flag obtenu** :
- 404CTF{fb31e1acc2e6eae8be01182d3029ffcb958e3368ca991ceb53895b8c97f2f275}

Modalités suivies :

https://github.com/EliasOenal/multimon-ng

https://www.gqrx.dk/

Gqrx est un récepteur radio défini par logiciel (SDR) open source, disponible sur Kali Linux. Il permet d'écouter et d'analyser des signaux radio tels que FM, AM, SSB, CW, et plus encore. Il est compatible avec divers matériels SDR, notamment RTL-SDR, HackRF, Airspy, USRP, et autres périphériques.

L'outil et librairie de **Gqrx** est utilisée pour rejouer le fichier IQ avec les réglages suivants : file=chall.iq, rate=4915200, repeat=true, throttle=true, Modulation : **Narrow FM**, centrée sur **135 kHz**, avec **streaming UDP activé** dans Gqrx (audio → réseau, port 7355).

Le pipeline de décodage est le suivant :

Bash

```
nc -l -u localhost 7355 | \
  sox -t raw -esigned-integer -b 16 -r 48000 - \
     -esigned-integer -b 16 -r 22050 -t raw - | \
  multimon-ng -t raw -a POCSAG1200 -f alpha -
```

Deux chemins reproductibles automatisables suivent:

Chemin A (recommandé) — Sans GUI, tout en local

Objectif : lire le contenu du fichier extrait (chall.iq (Complex64 @ 4 915 200 Hz), **centrer** la porteuse POCSAG (~ +135 kHz), **FM-démoduler** (quadrature), **filtrer/décimer** vers **22 050 Hz** (format attendu par multimon-ng), puis **décoder POCSAG**.— On s'appuie sur la démodulation FM « quadrature demod », standard pour FM/FSK/GMSK, telle que décrite dans GNU Radio et PySDR. GNU RadioPySDRSite 2241

1) Script Python (pocsag_from_iq.py)

Ce script implémente exactement la **démodulation FM/FSK par discriminateur de phase**, recommandée pour FM/FSK (Quadrature Demod). https://wiki.gnuradio.org/index.php/Quadrature_Demod

2) Lancer le décodage

python3 pocsag_from_iq.py --iq chall.iq --fs 4915200 --fshift 135000 \
| multimon-ng -t raw -a POCSAG1200 -f alpha -

- multimon-ng décode POCSAG (512/1200/2400 existent ; ici 1200 suffit), en **entrée raw S16LE @ 22 050 Hz** (c'est son format attendu, d'où le *resample* 22 050 Hz dans le script). https://manpages.debian.org/testing/multimon-ng/multimon-ng.1.en.html

Apparaissent alors dans le terminal des lignes du type « POCSAG1200: … Alpha: … ». Parmi elles, se trouve le message contenant le flag :
404CTF{fb31e1acc2e6eae8be01182d3029ffcb958e3368ca991ceb53895b8c97f2f275}

Chemin B — Via Gqrx + UDP (le script d'origine)

Le script : decode_pocsag.py lance Gqrx en lisant chall.iq, puis attend un flux **UDP audio** que l'on reroute (pipe) vers sox | multimon-ng. Deux points clés à régler :

1. **Configurer Gqrx** pour l'IQ *playback* et l'UDP :

- Lecture IQ avec le bon **rate = 4 915 200 Hz**, **repeat** et **throttle** activés, **Narrow FM**, et **centrage ~ +135 kHz**.
- Activer le **streaming audio UDP** (host 127.0.0.1, **port 7355**) — Gqrx envoie **S16LE 48 kHz**. https://www.site2241.net/june2022.htm

2. **Pipeline terminal** (convertir 48 kHz → 22.05 kHz avant multimon-ng) :

```
nc -l -u 127.0.0.1 7355 | \
  sox -t raw -esigned-integer -b16 -r 48000 - \
    -esigned-integer -b16 -r 22050 -t raw - | \
  multimon-ng -t raw -a POCSAG1200 -f alpha -
```

- Si nc reste « bloqué », c'est **normal tant qu'aucun flux UDP n'arrive**. S'assurer que le **bouton UDP** est bien activé côté Gqrx, host/port corrects (127.0.0.1:7355). Les mainteneurs Gqrx confirment ce test avec nc -l -u 127.0.0.1 7355. https://github.com/gqrx-sdr/gqrx/issues/646

L'outil de diagnostic (script) decode_pocsag.py dépend justement de cette config GUI et du streaming UDP, d'où les « bloquages » si Gqrx n'émet rien. (voir le contenu du script joint).

Pourquoi ces réglages ?

- **POCSAG** = **2-FSK** avec déviation typique ± 4.5 kHz et **débits** 512/1200/2400 bps. Donc FM-démod (quadrature) + bande audio ~quelques kHz, puis horloge à ~1.2 kb/s. https://www.rfcandy.biz/communication/pocsag.html
-
- multimon-ng **attend du S16LE 22 050 Hz** en mode -t raw (classique avec rtl_fm -s 22050 | multimon-ng ...). D'où la **conversion** (sox) dans le chemin B, et le **resample** dans le chemin A. (https://groups.google.com/g/ultra-cheap-sdr/c/8_jaPQGMkAg).

Remarque : **sox » signifie** Sound eXchange**.** C'est un utilitaire audio en ligne de commande — souvent appelé le "couteau suisse" du son — qui sert à convertir, filtrer et ré-échantillonner des flux/fichiers audio. Dans le contexte de Gqrx, on l'emploie fréquemment pour transformer le flux audio (48 kHz, 16 bits LE) envoyé par Gqrx afin qu'il soit accepté par des décodeurs externes (via pipe avec nc, etc.).

Résultat attendu

En appliquant **Chemin A** (sans GUI) ou **Chemin B** (Gqrx+UDP) correctement, dans le terminal le message suivant apparaît contenant :
404CTF{fb31e1acc2e6eae8be01182d3029ffcb958e3368ca991ceb53895b8c97f2f275}

TL;DR

- Le plus robuste : **Chemin A** (script Python → FM demod → 22.05 kHz → multimon-ng).
- Alternative GUI : **Chemin B** (Gqrx IQ playback @ 4.9152 MHz, NFM +135 kHz, **UDP 48 kHz** → sox → multimon-ng).
- Si nc « bloque », c'est que **Gqrx n'envoie pas** (vérifie le bouton UDP / host:port).
- Si le flag s'est affiché, c'est que la chaîne **Gqrx → UDP → sox → multimon-ng** est bien réglée. (sox est l'adaptateur audio au milieu du pipeline (UDP 48 kHz S16LE → raw 22,05 kHz S16LE), indispensable à multimon-ng pour décoder POCSAG proprement.)

Un enregistrement IQ d'une transmission POCSAG est fourni ; il faut le relire avec Gqrx, streamer l'audio en UDP 48 kHz S16LE, le convertir (~22,05 kHz) et le passer à multimon-ng pour décoder le message alphanumérique contenant le flag. Concrètement : Gqrx → UDP:7355 → sox (48 k→22,05 k raw S16LE) → multimon-ng (POCSAG), ce qui révèle le flag.