

Zabawa z radiami zdefiniowanymi programowo

Wprowadzenie

Termin radio zdefiniowane programowo oznacza implementację sprzętowych komponentów radiowych, takich jak modulatory, demodulatory i tunery, przy użyciu oprogramowania. W tym rozdziale omówimy różne przepisy i przyjrzymy się wielu sposobom, w jaki RTLSDR może być używany do zabawy z częstotliwościami i danymi przesyłanymi przez nie.

Skanery częstotliwości radiowych

RTLSDR to bardzo tanie (około 20 USD) radio zdefiniowane programowo, które wykorzystuje klucz sprzętowy tunera telewizji DVB-T. W tym przepisie omówimy podłączanie urządzenia RTLSDR z Kali Linux, aby sprawdzić, czy zostało pomyślnie wykryte.

Przygotowania

Będziemy potrzebować sprzętu do tego przepisu. Można go łatwo kupić na Amazon lub na stronie <https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/>. Kali ma już narzędzia, dzięki którym możemy zacząć z nim pracować.

Jak to zrobić...

Podłączamy nasze urządzenie i powinno zostać wykryte w Kali Linux. Urządzenia często zachowują się nieprawidłowo. Oto przepis na uruchomienie testu:

1. Najpierw uruchomimy test za pomocą polecenia:

```
rtl_test
```

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
root@kali:~# rtl_test
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
 20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
 49.6
[R82XX] PLL not locked!
Sampling at 2048000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.

Reading samples in async mode...
lost at least 16 bytes
lost at least 60 bytes
lost at least 60 bytes
lost at least 60 bytes
lost at least 128 bytes
lost at least 196 bytes
```

2. Możemy zobaczyć kilka upuszczonych pakietów. Dzieje się tak, ponieważ próbowaliśmy tego w konfiguracji VM z samym USB 2.0.

3. W przypadku dużej liczby utraconych pakietów możemy to sprawdzić, ustawiając niższą częstotliwość próbkowania za pomocą `rtl_test -s 1000000`:

```
root@kali:~# rtl_test -s 1000000
Found 1 device(s):
 0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U OEM
Found Rafael Micro R820T tuner
Supported gain values (29): 0.0 0.9 1.4 2.7 3.7 7.7 8.7 12.5 14.4 15.7 16.6 19.7
20.7 22.9 25.4 28.0 29.7 32.8 33.8 36.4 37.2 38.6 40.2 42.1 43.4 43.9 44.5 48.0
49.6
Exact sample rate is: 1000000.026491 Hz
[R82XX] PLL not locked!
Sampling at 1000000 S/s.

Info: This tool will continuously read from the device, and report if
samples get lost. If you observe no further output, everything is fine.
```

4. Teraz jesteśmy gotowi przejść do następnego przepisu i pobawić się naszym urządzeniem.

Praktyczna praca ze skanerem RTLSDR

Skaner RTLSDR to wieloplatformowy interfejs graficzny, którego można używać do analizy widma. Skanuje on dany zakres częstotliwości i wyświetla wynik w postaci spektrogramu.

Jak to zrobić...

Oto przepis na uruchomienie `rtlsdr-scanner`:

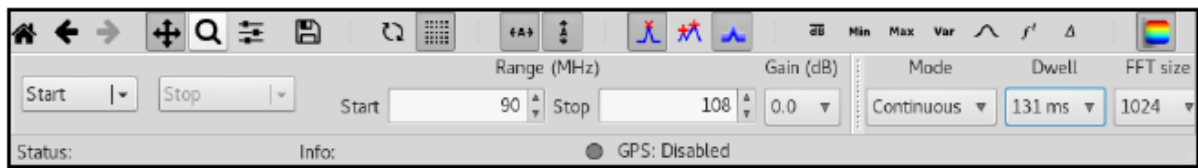
1. Podłączamy RTLSDR do systemu i uruchamiamy skaner za pomocą polecenia:

```
rtlsdr-scanner
```

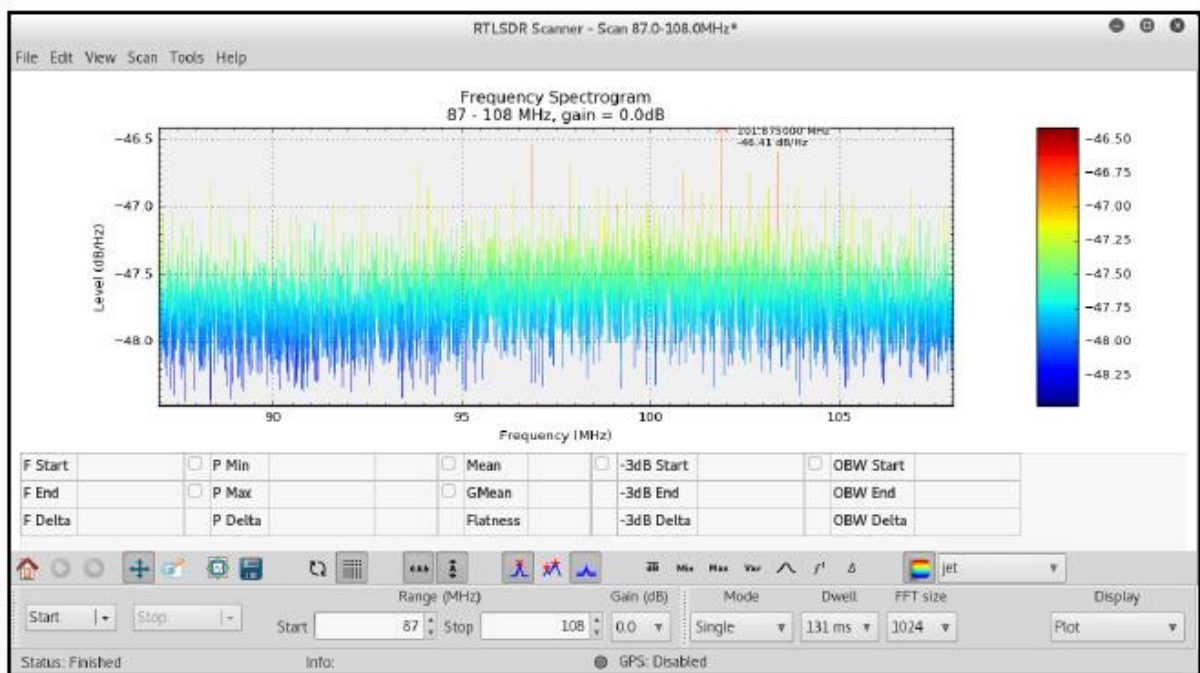
Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
root@kali:~# rtlsdr-scanner
RTLSDR Scanners 105.0 107.5 -50
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
/usr/lib/python2.7/dist-packages/matplotlib/cbook.py:136: MatplotlibDeprecationWarning: The axisbg attribute was deprecated in version 2.0. Use facecolor instead.
  warnings.warn(message, mplDeprecation, stacklevel=1)
/usr/lib/python2.7/dist-packages/matplotlib/cbook.py:136: MatplotlibDeprecationWarning: idle_event is only implemented for the wx backend, and will be removed in matplotlib 2.1. Use the animations module instead.
  warnings.warn(message, mplDeprecation, stacklevel=1)
05:52:24: Debug: ScreenToClient cannot work when toplevel window is not shown
05:52:24: Debug: ScreenToClient cannot work when toplevel window is not shown
05:52:24: Debug: ScreenToClient cannot work when toplevel window is not shown
(rtlsdr_scan.py:6254): Gdk-WARNING **: gdk_window_set_icon_list: icons too large
05:52:24: Debug: ScreenToClient cannot work when toplevel window is not shown
(rtlsdr_scan.py:6254): Gdk-WARNING **: gdk_window_set_icon_list: icons too large
```

2. Powinno otworzyć się nowe okno pokazujące interfejs graficzny narzędzia; w tym miejscu wystarczy wpisać zakres częstotliwości, w którym chcemy przeprowadzić skanowanie, i kliknąć Rozpocznij skanowanie:



3. Zajmie trochę czasu, zanim zobaczymy przegląd częstotliwości, a następnie zobaczymy wynik w formacie graficznym:



Jeśli aplikacja przestanie odpowiadać, zaleca się obniżenie zakresu i wybranie Single jako Mode zamiast continuous.

Zabawa z gqrX

Narzędzie gqrX to odbiornik radiowy (SDR) typu open source, oparty na radiu GNU i graficznym zestawie narzędzi Qt. Posiada wiele funkcji, takich jak:

- * Wykrywanie urządzeń podłączonych do komputera
- * Przetwarzanie danych I/Q
- * Demodulatory AM, SSB, CW, FM-N i FM-W (mono i stereo)
- * Nagrywanie i odtwarzanie dźwięku do/z pliku WAV
- * Nagrywanie i odtwarzanie surowych danych pasma podstawowego
- * Przesyłanie strumieniowe dźwięku przez UDP

W tym przepisie omówimy podstawy gqrX i innego narzędzia, RTLSDR

Jak to zrobić...

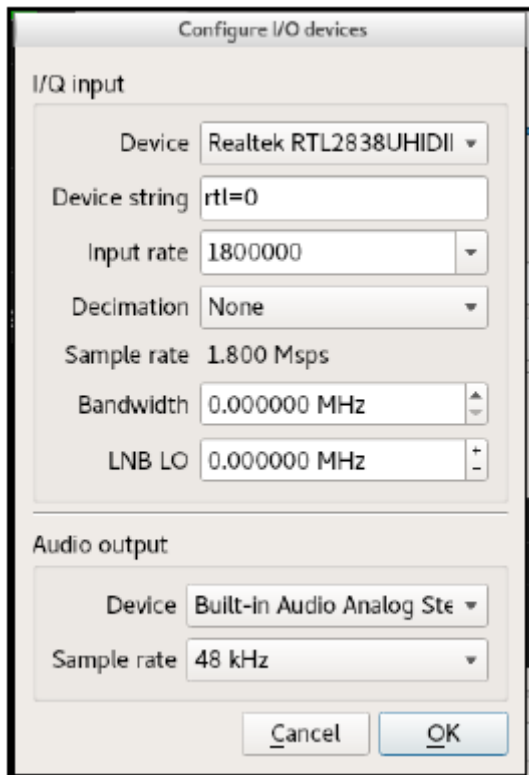
Oto przepis na użycie gqrx:

1. Możemy zainstalować gqrx za pomocą polecenia:

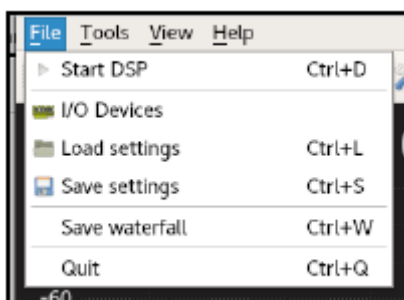
```
apt install gqrx
```

2. Po wykonaniu tej czynności uruchamiamy narzędzie, wpisując gqrx.

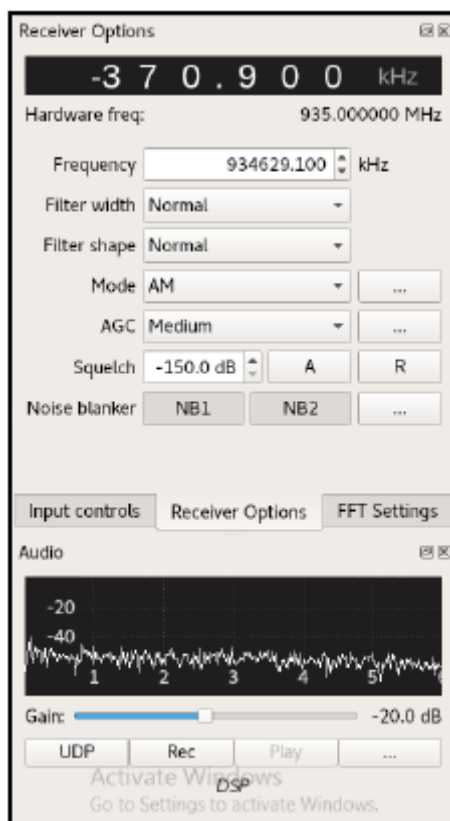
3. Wybieramy nasze urządzenie z menu rozwijanego w oknie, które się otworzy i klikamy OK:.



4. Teraz otwiera się aplikacja GQRX, a po prawej stronie w oknie odbiornika wybieramy częstotliwość, którą chcemy wyświetlić. Następnie przechodzimy do pliku i klikamy na Start DSP:

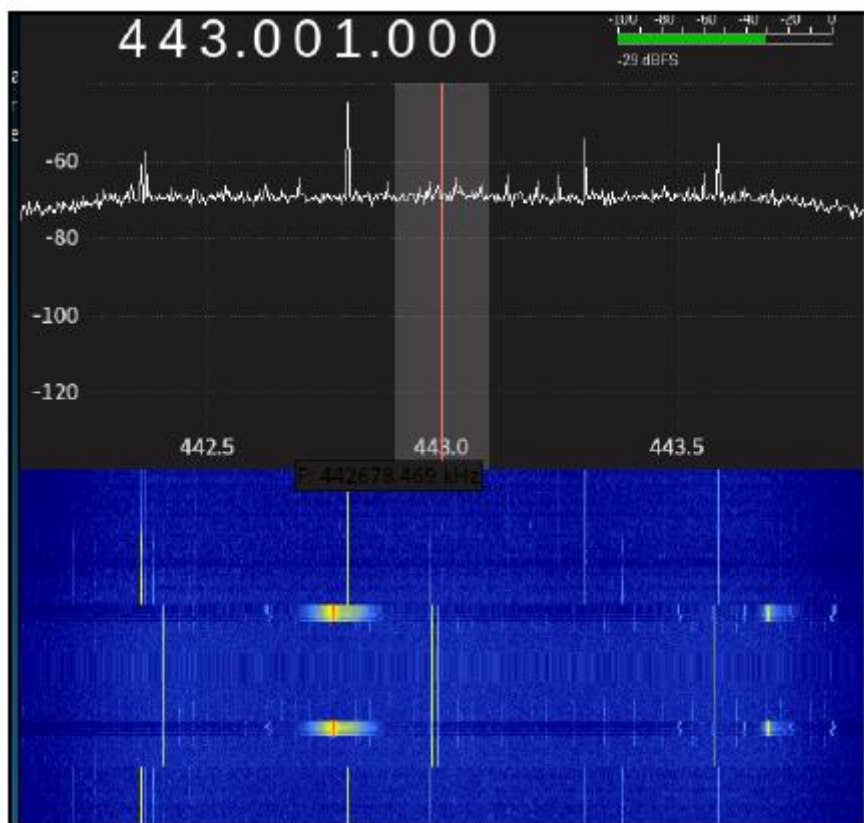


5. Teraz widzimy wodospad i powinniśmy zacząć słyszeć dźwięk w naszym głośniku. Możemy nawet zmienić częstotliwość, której słuchamy, używając przycisków w górę i w dół w oknie Opcje odbiornika:



6. Przyjrzymy się przykładowi pilota samochodowego, który służy do zamykania/otwierania samochodu.

7. Po kilkakrotnym naciśnięciu przycisku zobaczymy zmianę w wodospadzie pokazującą różnicę w sygnale:



8. Możemy nagrać sygnał w oknie nagrywania, a następnie go zapisać. Później można go zdekodować i przesłać z powrotem do samochodu za pomocą transpondera, aby go odblokować.

9. Aby przechwycić dane przy częstotliwości 443 MHz, możemy użyć polecenia:

```
rtl_sdr -f 443M - | xxd
```

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```

root@kali:~# rtl_sdr -f 93.5M - | xxd
Found 1 device(s):
  0: Realtek, RTL2838UHIDIR, SN: 00000001

Using device 0: Generic RTL2832U DEM
Found Rafael Micro R820T tuner
[R82XX] PLL not locked!
Sampling at 2048000 S/s.
Tuned to 93500000 Hz.
Tuner gain set to automatic.
Reading samples in async mode.
0000000: 00c7 00c2 a1ae 40ff 30ff ff97 bab1 15bb @.0.....mm
0000010: da6a b593 ff90 ff19 0ffb2 30de ffa2 ebc0 .j.....0...[]
0000020: 1b8d ff3b 2660 c97e 4aa3 0000 05ff ffff ....&`~J.....
0000030: 5eae 7fff 29c0 6400 64ff 7c79 3ee7 3630 .(....).d.d.|y>.60
0000040: 12f5 8da9 6163 37aa 96ff 3136 c205 2330 ...ac7...16..#0
0000050: ab6a 2ed0 3700 5523 70f7 9c00 6d84 50ff .j..7.U#p...m.P.
0000060: 7201 b239 2e0e 62a3 2bbf 7483 3026 c0ff .n..9..b.+t.0&..
0000070: 0e88 ffff 6eb5 9395 829b 5e7e adff 182c .n.....^~....,
0000080: 0098 7700 a8b4 a4ff ffdc 04ab 205b 41c7 ..w..... [A.
0000090: a9ff 4085 9a00 2964 a9ff 4044 0039 0c53 ..@...)d..@D.9.S
00000a0: 9c21 4b8c de31 2fd4 30b0 9eff 8bff 3332 ..!K..1/.0.....32
00000b0: 4e19 00ff 4f00 4b87 4f49 ef71 0ddb 0087 N...0.K.0I.q....
00000c0: 28ff 0092 e700 4d6d 0099 a304 108e aa07 (. ....Mm.....
00000d0: 7883 4917 cdff 0fff 2872 9940 cffe cb31 x.I.....{r.@...1
00000e0: 6e93 9529 a2a5 5e31 7b47 00c6 d6ff 5ab1 n..) ..^1{G....Z.
00000f0: 0067 ff00 9fb8 d25d 8f92 7947 a0c4 6299 .g.....]..yG..b.
000100: de00 5900 83e3 b164 ff5e 0088 4e63 40af ..Y....d.^..Nc@.

```

Urządzenie kalibrujące do podsłuchu GSM

RTLSDR pozwala nam również przeglądać ruch GSM za pomocą narzędzia o nazwie kal lub kalibrate-rtl. To narzędzie może skanować stacje bazowe GSM w paśmie częstotliwości. W tym przepisie nauczymy się korzystać z kalibrate, a następnie potwierdzimy kanał w gqrx.

Jak to zrobić...

Oto kroki korzystania z kalibrate:

- Większość krajów korzysta z pasma GSM900. W USA jest to 850. Użyjemy następującego polecenia, aby przeskanować stacje bazowe GSM:

```
kal -s GSM900 -g 40
```

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:

```
root@kali:~/config# kal -s GSM900 -g 40
Found 1 device(s):
 0: Generic RTL2832U OEM

Using device 0: Generic RTL2832U OEM
Detached kernel driver
Found Rafael Micro R820T tuner
Exact sample rate is: 270833.002142 Hz
[R82XX] PLL not locked!
Setting gain: 40.0 dB
kal: Scanning for GSM-900 base stations.
```

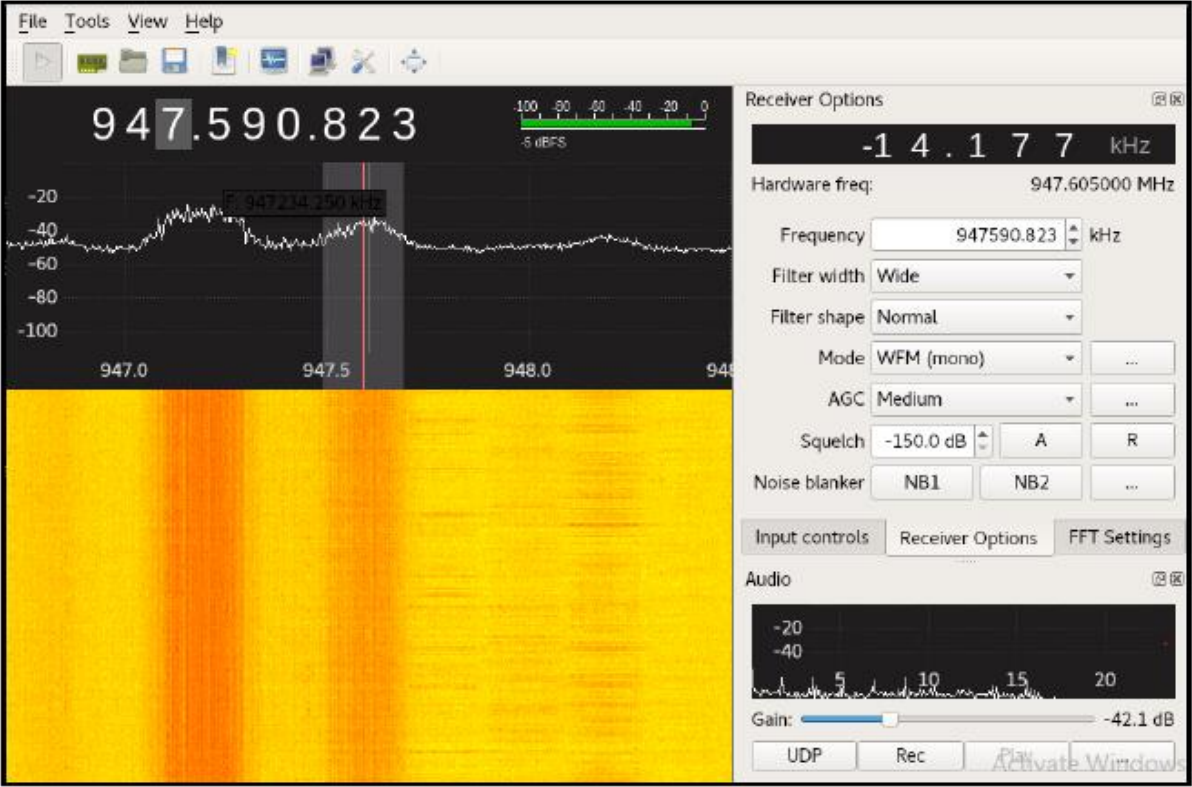
2. Za kilka minut wyświetli nam się lista stacji bazowych:

```
GSM-900:

chan: 32 (941.4MHz - 15.209kHz) power: 991758.24
chan: 34 (941.8MHz - 15.099kHz) power: 835333.49
chan: 51 (945.2MHz - 14.653kHz) power: 2857467.65
chan: 53 (945.6MHz - 14.620kHz) power: 3310824.09
chan: 57 (946.4MHz - 15.736kHz) power: 2261161.19
chan: 61 (947.2MHz - 15.201kHz) power: 4090351.91
chan: 63 (947.6MHz - 14.177kHz) power: 2990914.87
```

3. Zapisujemy częstotliwość; w naszym przypadku użyjemy 947,6 MHz wraz z przesunięciem.

4. Teraz otwieramy GQRX i wprowadzamy go w oknie Opcje odbiornika:



5. Na wykresie wodospadu widać, że urządzenie doskonale wychwytuje sygnały.
6. Teraz przyjrzymy się tym danym na poziomie pakietów. Użyjemy narzędzia znanego jako grgsm.
7. Można je zainstalować za pomocą apt install gr-gsm:

```

root@kali:~#
root@kali:~# apt install gr-gsm
Reading package lists... Done
Building dependency tree
Reading state information... Done
gr-gsm is already the newest version (0.41.2-1).
The following packages were automatically installed and are no longer required:
  apg apt-transport-https aptitude-doc-en augeas-lenses cheese-common commix
  couchdb cups-pk-helper dkms empathy-common erlang-asnl erlang-base
  erlang-crypto erlang-eunit erlang-inets erlang-mnesia erlang-os-mon
  erlang-public-key erlang-runtime-tools erlang-snmp erlang-ssl
  erlang-syntax-tools erlang-tools erlang-xmerl espeak-data exe2hexbat
  firebird2.5-common firebird2.5-common-doc folks-common gdebi-core
  gir1.2-clutter-gst-2.0 gir1.2-javascriptcoregtk-3.0 gir1.2-totem-1.0
  gir1.2-totem-plparser-1.0 gir1.2-webkit-3.0 gnome-control-center-data
  gstreamer1.0-clutter gstreamer1.0-nice gstreamer1.0-plugins-ugly
  guile-2.0-libs ipxe-qemu king-phisher libasn1-8-heimdal libaugeas0
  libbind9-90 libbladerf0 libboost-fs1.55.0
  libboost-program-options1.55.0 libboost-python1.55.0 libboost-regex1.55.0
  libboost-serialization1.55.0 libboost-system1.55.0 libboost-test1.55.0
  libboost-thread1.55.0 libcacard0 libchamplain-0.12-0 libchamplain-gtk-0.12-0
  libclass-accessor-perl libclutter-gst-2.0-0 libcolord-gtk1 libcrypto++6
  libcrypto++9 libdbus-1-dev libdee-1.0-4 libdns100 libebackend-1.2-7
  libedata-cal-1.2-23 libegl1-mesa-drivers libelfg0 libept1.4.12 libespeak1
  libexiv2-13 libfdt1 libfluidsynth1 libfolks-eds25 libfolks-telepathy25
  libfolks25 libfuzzy2 libgdict-1.0-6 libglewl.10 libgphoto2-port10

```

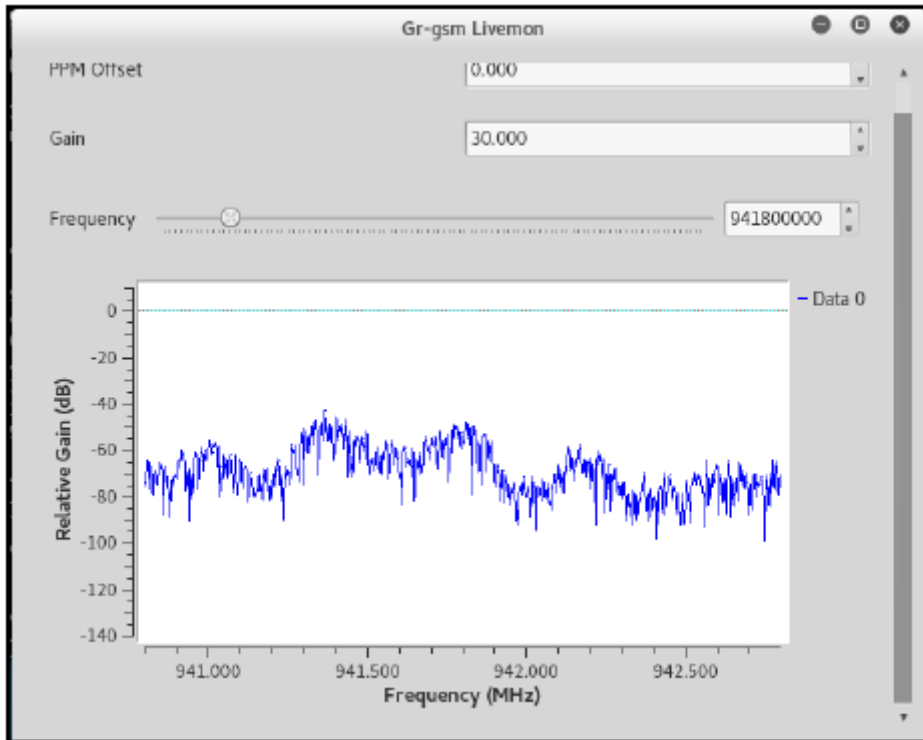
8. Po wykonaniu tej czynności, jeśli wpiszemy grgsm_ i naciśniemy klawisz Tab, wyświetli się nam lista różnych dostępnych narzędzi:

```

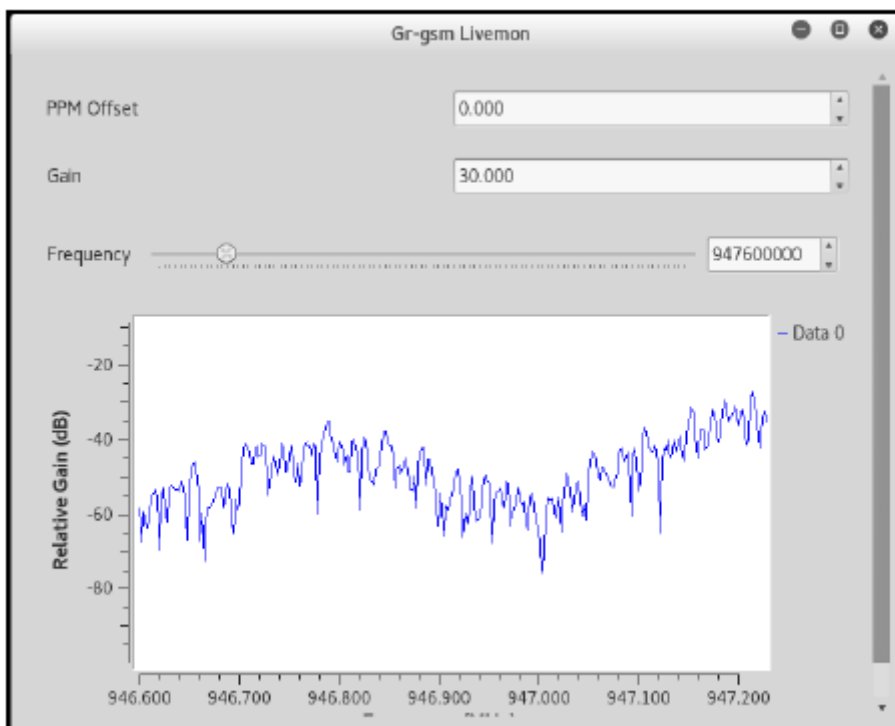
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# grgsm_
grgsm_capture          grgsm_decode          grgsm_livemon_headless
grgsm_channelize      grgsm_livemon         grgsm_scanner
root@kali:~# grgsm_

```

9. Najpierw użyjemy grgsm_livemon do monitorowania pakietów GSM na żywo. Otworzymy terminal i wpiszemy grgsm_livemon:



10. W nowym oknie, które się otworzy, przełączymy się na częstotliwość, którą przechwyciliśmy w poprzednich krokach za pomocą kalibrata:



11. Możemy powiększyć określony zakres, przeciągając i zaznaczając obszar w oknie graficznym.

12. W nowym oknie terminala uruchamiamy Wireshark, wpisując wireshark.

13. Następnie ustawiamy adapter na Loopback: lo i rozpoczynamy przechwytywanie pakietów:



14. Następnie dodajemy filtr gsmmap:

No.	Time	Source	Destination	Protocol	Length	Info
410	6.559896000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
411	6.561027000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
412	6.563428000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
413	6.563698000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
414	6.565694000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
415	6.565874000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
416	6.620051000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
417	6.620165000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
418	6.631228000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
419	6.632487000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
420	6.633805000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
421	6.688895000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
422	6.688854000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown
423	6.692349000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
424	6.692515000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown
425	6.695730000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown
426	6.696818000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
427	6.697082000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown
428	6.754927000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
429	6.760595000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)

15. Powinniśmy zobaczyć pakiety w oknie informacyjnym. Powinniśmy zobaczyć pakiet z etykietą System Information Type 3; otworzymy go:

2121	36.368615000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
2122	36.371373000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown
2123	36.372337000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1
2124	36.374437000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
2125	36.434906000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) System Information Type 3
2126	36.439487000	127.0.0.1	127.0.0.1	LAPDm	81	U, func=Unknown[DTAP] (SS)
2127	36.444452000	127.0.0.1	127.0.0.1	GSMTAP	81	(CCH) (RR) Paging Request Type 1

16. Zobaczymy informacje systemowe, takie jak kod kraju telefonu komórkowego, kod sieci i kod obszarowy lokalizacji:

```
▼ GSM CCCH - System Information Type 3
  ▶ L2 Pseudo Length
  ▶ Protocol Discriminator: Radio Resources Management messages
    Message Type: System Information Type 3
  ▶ Cell Identity - CI (51661)
  ▼ Location Area Identification (LAI)
    ▼ Location Area Identification (LAI) - 404/10/617
      Mobile Country Code (MCC): India (Republic of) (404)
      Mobile Network Code (MNC): Bharti Airtel Ltd., Delhi (10)
      Location Area Code (LAC): 0x0269 (617)
    ▶ Control Channel Description
    ▶ Cell Options (BCCH)
    ▶ Cell Selection Parameters
    ▶ RACH Control Parameters
    ▶ SI 3 Rest Octets
```

17. Dzięki temu przepisowi dowiedzieliśmy się, jak przesyłane są pakiety GSM.

Dekodowanie wiadomości ADS-B za pomocą Dump1090

ADS-B oznacza Automatic Dependent Surveillance-Broadcast (automatyczny zależny nadzór-rozglaszanie). Jest to system, w którym sprzęt elektroniczny na pokładzie samolotu automatycznie nadaje dokładną lokalizację samolotu za pośrednictwem cyfrowego łącza danych. Jak opisano w oficjalnym pliku readme narzędzia, Dump1090 to dekodery Mode S zaprojektowany specjalnie dla urządzeń RTLSDR. Główne cechy to:

- * Solidne dekodowanie słabych wiadomości. Dzięki mode1090 wielu użytkowników zauważyło lepszy zasięg w porównaniu z innymi popularnymi dekodernami.
- * Obsługa sieci — strumień TCP30003 (MSG5), surowe pakiety, HTTP.
- * Wbudowany serwer HTTP, który wyświetla aktualnie wykryte samoloty na Mapach Google.
- * Jednobitowa korekcja błędów przy użyciu 24-bitowego CRC.
- * Możliwość dekodowania wiadomości DF11 i DF17.
- * Możliwość dekodowania formatów DF, takich jak DF0, DF4, DF5, DF16, DF20 i DF21, gdzie suma kontrolna jest XOR-owana z adresem ICAO poprzez brutalne wymuszenie pola sumy kontrolnej przy użyciu adresów ICAO, które omówiliśmy.
- * Dekodowanie surowych próbek IQ z pliku (za pomocą przełącznika wiersza poleceń --ifile).
- * Interaktywny tryb CLI, w którym aktualnie wykryte samoloty są wyświetlane jako lista, odświeżana w miarę napływania nowych danych.
- * Dekodowanie współrzędnych CPR i obliczanie śladu na podstawie prędkości.
- * Serwer TCP przesyła strumieniowo i odbiera surowe dane do/od podłączonych klientów (za pomocą --net).

W tym przepisie użyjemy tego narzędzia do przeglądania ruchu lotniczego za pomocą wizualizacji.

Jak to zrobić...

Oto kroki, aby użyć Dump1090:

1. Możemy pobrać narzędzie z repozytorium Git, używając polecenia `git clone https://github.com/antirez/dump1090.git`:

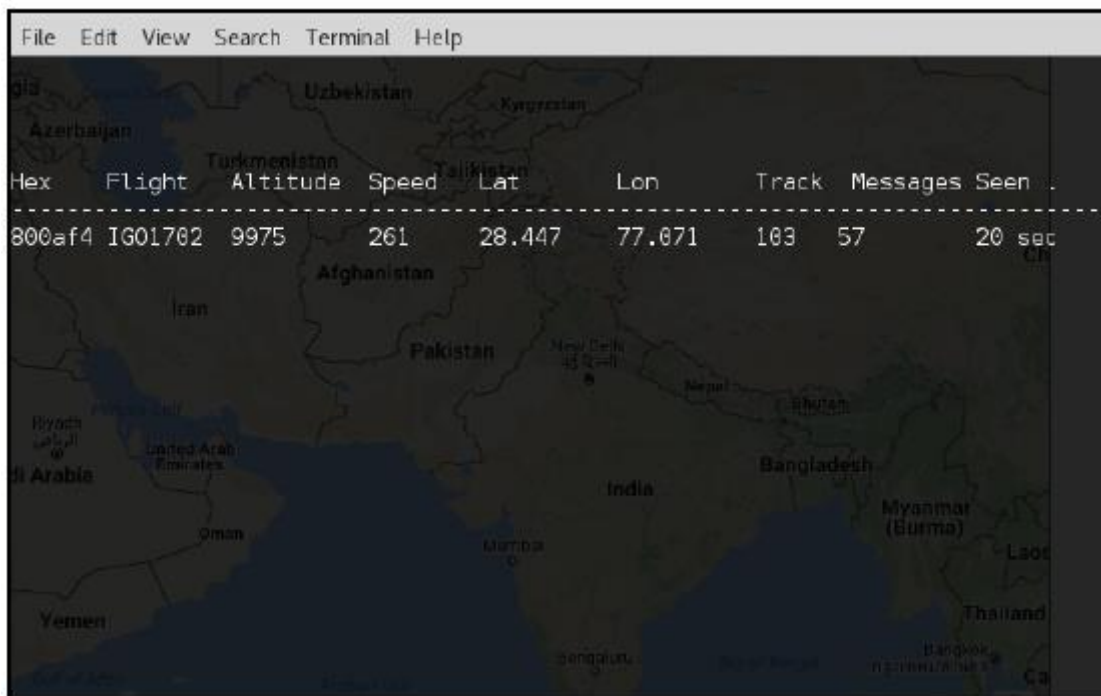
```
root@kali:~# git clone https://github.com/antirez/dump1090.git
Cloning into 'dump1090'...
remote: Counting objects: 265, done.
remote: Total 265 (delta 0), reused 0 (delta 0), pack-reused 265
Receiving objects: 100% (265/265), 536.32 KiB | 266.00 KiB/s, done.
Resolving deltas: 100% (147/147), done.
root@kali:~#
```

2. Po pobraniu przechodzimy do folderu i uruchamiamy `make`.

3. Teraz powinniśmy mieć plik wykonywalny. Możemy uruchomić narzędzie za pomocą następującego polecenia:

```
./dump1090 --interactive -net
```

Poniższy zrzut ekranu pokazuje wynik poprzedniego polecenia:



4. Za kilka minut powinniśmy zobaczyć loty, a po otwarciu przeglądarki na <http://localhost:8080> będziemy mogli zobaczyć loty również na mapie.