Inspectrum

Contents

AN-392

Siddhant Dhawan and Neel Pandeya

Inspectrum is an open-source signal analysis tool used for offline examination of IQ data captured from Software Defined Radios (SDRs) such as USRPs. It provides a spectrogram view (frequency versus time) along with FFT-based spectrum visualization, enabling detailed inspection of recorded RF signals.

Inspectrum supports common USRP capture formats, including CF32 (`.cfile`) and SigMF recordings, and offers interactive zooming and panning for analysis at different time and frequency resolutions. The tool is widely used to measure signal bandwidth, burst duration, frequency offsets, and channel occupancy, making it useful for the investigation of LTE, 5G NR, Wi-Fi, radar, and other wireless communication signals without requiring real-time hardware access.

This guide assumes you have a Linux-based host computer with the required UHD and GNU Radio versions. The procedure described in this document was tested on Ubuntu Linux 24.04.3 LTS with UHD 4.9.0 and GNU Radio 3.10.12.

While other compatible versions may also work, it is recommended to use these versions for consistency with the examples presented in this guide.

Before proceeding, ensure that the following software components are installed and functioning correctly:

- Ubuntu Linux 24.04.3 LTS (or a compatible Linux distribution)
- UHD 4.9.0
- GNU Radio 3.10.12

If these components are not installed, please follow the setup instructions provided in the Ettus Research Knowledge Base article:

https://kb.ettus.com/Instructions_for_System_Setup_and_Configuration

Once the software environment has been configured successfully, you may proceed with the installation and use of Inspectrum for offline analysis of IQ data captured from a USRP device.

Before building Inspectrum, install the required development packages and libraries. These packages provide the build tools, FFT processing libraries, and Qt framework components required for signal visualization and analysis.

Update the package repository and install the required dependencies:

```
sudo apt update
sudo apt install -y git build-essential cmake pkg-config \
libfftw3-dev qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools
```

Inspectrum also depends on the Liquid-DSP library, an open-source digital signal processing library widely used in SDR applications. Liquid-DSP provides optimized implementations of DSP algorithms such as filtering, resampling, modulation/demodulation, synchronization, and spectral analysis.

Inspectrum uses components of this library to efficiently process and analyze recorded IQ data.

Build and install the latest version of Liquid-DSP from source using the following steps:

```
cd ~/git
git clone https://github.com/jgaeddert/liquid-dsp.git
cd liquid-dsp
mkdir build
cd build
cmake ..
make -j$(nproc)
sudo make install
sudo ldconfig
```

The `ldconfig` command updates the system's shared library cache, ensuring that applications can locate the newly installed Liquid-DSP libraries at runtime.

After completing these steps, the system will have all required dependencies installed and will be ready for building Inspectrum from source.

After installing the required dependencies, download the Inspectrum source code and build it from source. Inspectrum is distributed as an open-source project and can be obtained directly from its GitHub repository.

Clone the repository and navigate to the source directory:

```
cd ~/git
git clone https://github.com/miek/inspectrum.git
cd inspectrum
```

Check out the v0.4.0 release tag:

```
git checkout v0.4.0
```

Version 0.4.0 was the latest official Inspectrum release at the time this application note was prepared and was used for validation of the procedures described in this document.

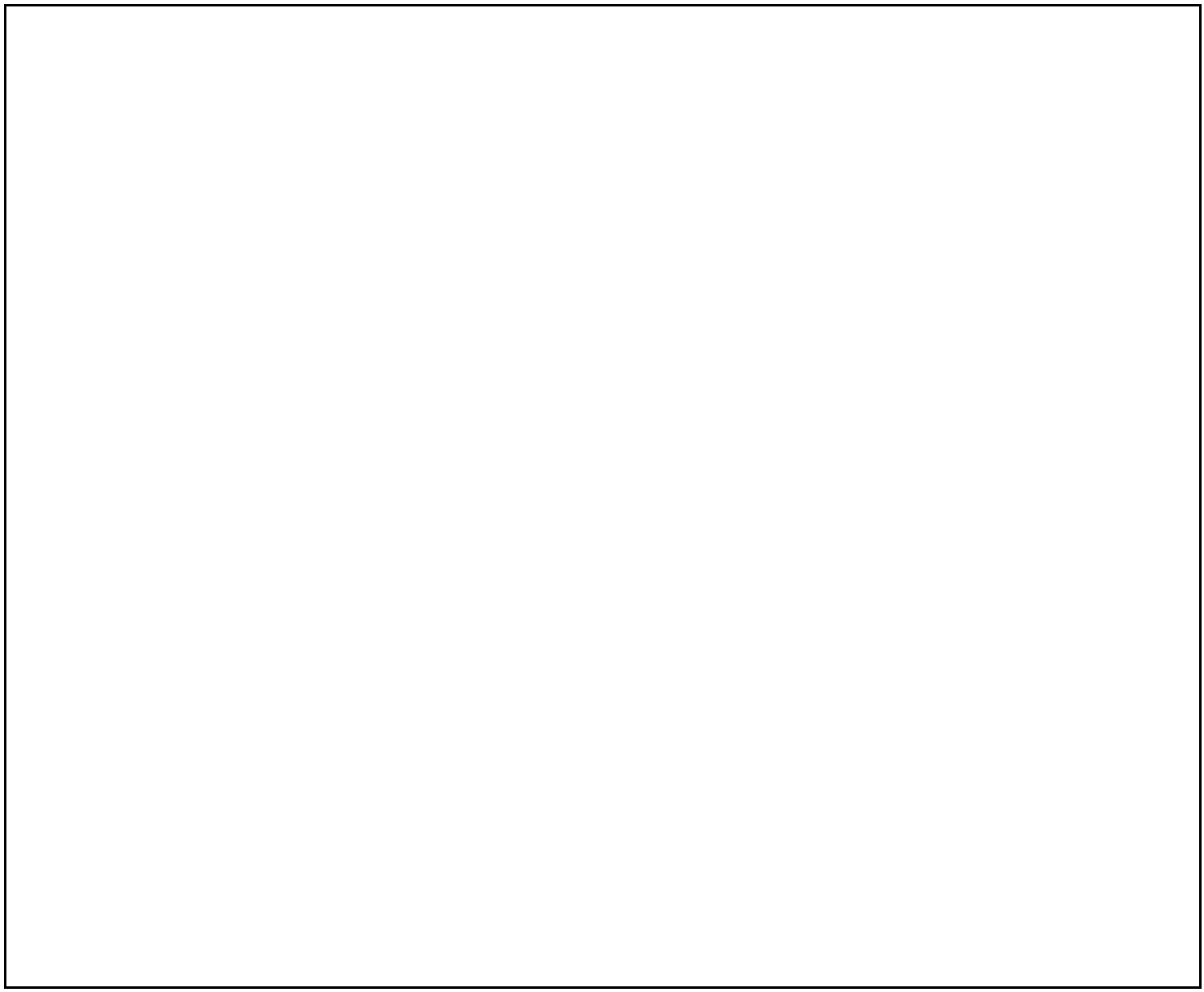
Create a build directory, configure the project using CMake, and compile the source code:

```
mkdir build
cd build
cmake ..
make -j$(nproc)
```

Install the compiled binaries and supporting files:

```
sudo make install
```

Upon successful installation, the `inspectrum` executable will be available on the system and can be launched from a terminal to analyze recorded IQ data files.



Inspectrum supports a wide range of IQ sample formats commonly generated by SDR platforms and signal recording tools. This flexibility allows previously captured RF data to be analyzed without requiring conversion to a specific format.

The supported input formats include:

- SigMF recordings (`*.sigmf-meta`, `*.sigmf-data`)
- Complex float (`*.cf32`, `*.fc32`, `*.cfile`)
- Complex double (`*.cf64`, `*.fc64`)
- Complex integer (`*.cs32`, `*.cs16`, `*.cs8`)
- RTL-SDR / unsigned complex (`*.cu8`, `*.uc8`)
- Real-valued samples (`*.f32`, `*.f64`, `*.s16`, `*.s8`, `*.u8`)

When a file extension is not recognized, Inspectrum assumes the data is stored in Complex Float 32-bit (`.cf32`) format.

In this application note, the example data was captured using GNU Radio and stored in the `.cfile` format.

The recording contains an LTE TDD Configuration 40 signal with a nominal channel bandwidth of 20 MHz and an effective occupied bandwidth of approximately 18 MHz. The captured IQ data is subsequently analyzed in Inspectrum to examine signal bandwidth, frame timing, uplink/downlink activity, and other RF characteristics.

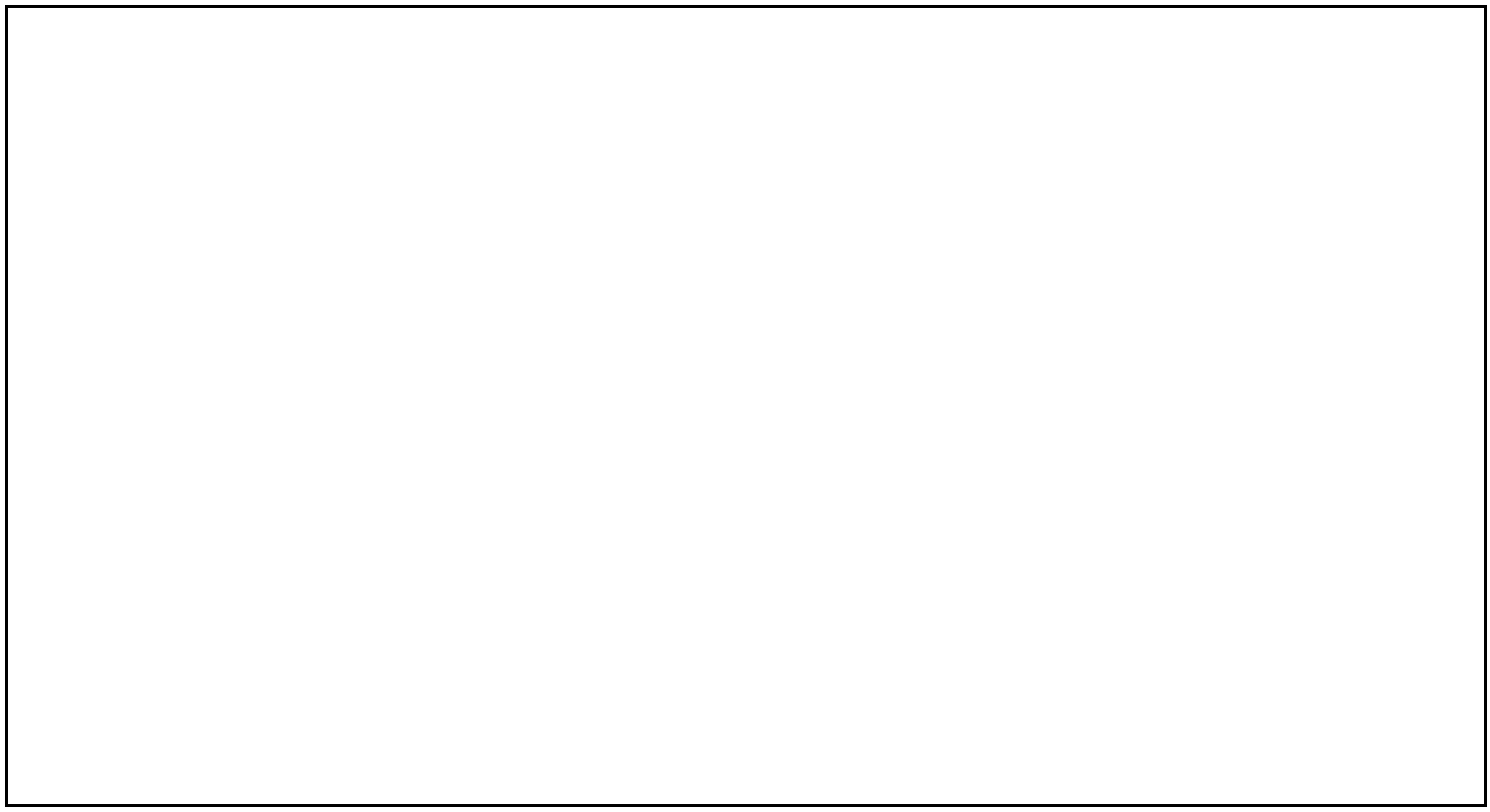


Figure below shows an LTE signal captured with a nominal channel bandwidth of 20 MHz and analyzed using Inspectrum. The spectrogram and FFT views provide both a full-capture overview and a zoomed-in view of the recorded signal.

The occupied spectrum spans approximately 18 MHz, which is consistent with the effective usable bandwidth of a 20 MHz LTE carrier after accounting for guard bands. The signal bandwidth is clearly visible in the spectrum display, demonstrating how Inspectrum can be used to verify channel occupancy and examine LTE signal characteristics from recorded IQ data.