

Generation of Dial Tone using GNU Radio

Assistant Professor D.Diana Josephine, Vijaya Prabha.A, Suba Jeyashree.V, Sangeetha.S

pithodcse@gmail.com, akashgkv152@gmail.com, nlokeshkumar199@gmail.com, sasikukarperiasamy123@gamil.com

Abstract – GNU Radio is widely used to implement cognitive radio designs, enabling both experimentation and research. In recent years, Software defined radio has become a cost efficient and reliable communication paradigm where it's RF front end is simplest as compared to the conventional SCR (software controlled radio). This paper aims to generate a standard dial tone consists of a 350 Hz and a 440 Hz signal using GNU, an open-source SDR platform. The graphical output of the generated dial tone is presented.

Keywords – GNU radio, SDR, Dial tone.

I. INTRODUCTION

Software defined radio (SDR) has recently become an attractive method to implement many RF applications. In general, SDR enables the computer to handle many of the signal processing processes within a software environment; where in the past, this was handled by hardware components. This transition into the software domain leads to several advantages that are not available on hardware based radio. The SDR is able to be more reconfigurable, where it can be reconfigured to adapt to new standards much easily. It can also be much more flexible, where its platform can be universal and be used in a wide range of uses. It is upgradable and more cost efficient when compared to hardware based radio. The SDR can also be used as a cognitive radio, where it automatically changes its parameters so that it can select the best available channel. Reversely, “Cognitive Radio is an SDR that is frequency-agile, fully reconfigurable and able to sense its spectrum surroundings; it knows policies, rules and regulations, and is flexible enough to reconfigure itself to different air interfaces and/or protocols.”

The GNU Radio project was conceptualized by Eric Blossom. It is an open source toolkit, which is made of several signal processing and communication blocks written in C++ and mapped onto Python using a simplified wrapper and interface generator (SWIG). GNU Radio Companion (GRC) is a graphical user interface, which makes dealing with this tool a little easier. It automatically generates source code (in Python) as per the connections between various blocks. Appropriate connection of various signal processing blocks in a signal flow graph creates software radio. It enables users to generate and process waveforms in software with a rich interface.

1. Dial Tone

A dial tone is a telephony signal sent by a telephone exchange or private branch exchange (PBX) to a terminating device, such as a telephone, when an off-hook condition is detected. It indicates that the exchange is working and is ready to initiate a telephone call. The dial

tone sound is simply a combination of 350-hertz tone and a 440-hertz tone.

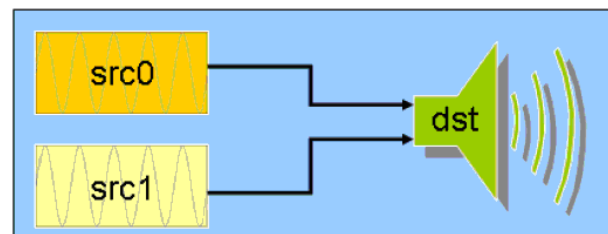


Fig. 1. Dial tone Generation.

In this paper GNU radio is used to produce a tone that is similar to the dial tone one would hear from a landline telephone. Both Python code and the GUI interface of GNU radio is used. Fig 1 shows a simple block diagram and Fig 2 shows a simple flow graph of dial tone generator.

```
sig_source_f (freq = 350) -->  
    audio.sink
```

```
sig_source_f (freq = 440) -->
```

Fig 2 Flow graph of Dial tone Generation

The two sources generate sinusoidal signals of two different frequencies. They are then mixed up and are sent out through a single audio sink.

II. GNU RADIO COMPANION

In addition to writing Python code, the GNU radio companion (GRC) can be used to produce Python code without explicitly writing it. It utilizes block diagrams in a GUI interface to produce the same results of explicitly written Python code. However, it has limited functionality in comparison to Python coding, thus is not used in more advanced and complex projects. Fig 2 shows the block diagram of GNU Radio Companion associated with dial tone generator.

GNU radio companion allows for the user to outline the process needed and the program produces the code. GNU

- USRP. In 2019 19th International Symposium on Communications and Information Technologies (ISCIT) (pp. 442-447). IEEE.
- [5]. Sisbot, E. A., Vega, A., Paidimarri, A., Wellman, J. D., Buyuktosunoglu, A., Bose, P., & Trilla, D. (2019, September). Multi-Vehicle Map Fusion using GNU Radio. In Proceedings of the GNU Radio Conference (Vol. 4, No. 1).
- [6]. Tan, J., Zhong, Z., & Liu, R. (2019, December). Implementation of channel simulation and GMSK modulation signal transmission system based on GNU Radio. In Eleventh International Conference on Signal Processing Systems (Vol. 11384, p. 113840J). International Society for Optics and Photonics.
- [7]. Manco-Vasquez, J., Moy, C., & Bader, F. (2019, June). GNU Radio implementation for Multiuser Multi-Armed Bandit learning algorithms in IoT networks.
- [8]. Kuntal Ray. An Introduction to GNU Radio
- [9]. Ivan Seskar. Hands-on-Tutorial, Future Internet Summer School 2009 Part 2, The State University of New Jersey
- [10]. <https://gnuradio.org/redmine/users/3733>
- [11]. Installing GNU Radio on Ubuntu
<https://gnuradio.org/redmine/projects/gnuradio/wiki/UbuntuInstall>