
Nathan Harter
September 2017
Introduction

What does CyberRadio Solutions do?

CyberRadio’s mission is to deliver cost-effective hardware solutions that combine high-end RF performance, embedded signal processing and standard network data interfaces.

Our product line features:

- Low-cost, high-performance tuners
- Multichannel, phase-coherent receivers
- Variety of general purpose and application-specific firmware loads
- RF switching & distribution
How We Use GNU Radio

- Radio Test Applications
  - Interactive Radio Demonstration GUI
  - Transmit Streaming Testing
  - Transceiver Coherency Testing
Why We Use GNU Radio

● The Problem:
  ● We need to test our various radio models, both for design verification and final product testing
  ● Early test applications:
    ● Single-purpose and often customized for a single radio
    ● Command line applications without a GUI
    ● Used file-based data processing

● GNU Radio is the answer:
  ● We implemented common radio control API and data sources
  ● GUIs are easier and faster to implement
  ● Allows for realtime data processing
  ● GRC is a great tool for rapid development
Demo GUI

- **Purpose:**
  - We needed to provide utilities to accelerate customer evaluation of our radios without requiring software development on their part
  - Provides radio control examples for custom application development
Demo GUI
Demo GUI Implementation

- Radio control implemented in custom block
  - Entire application implemented in GNU Radio Companion

- UDP Source Block
  - Used across our various radio types
  - Flexible Vita49 parsing
  - Stream tagging
**Tx Streaming Control**

- **Purpose:**
  - Verify independent wideband transmit streaming integrity
  - Adjust flow control parameters to optimize throttling of transmit streams
  - Check for packet loss and buffer over/under runs
  - Signal integrity through digital and analog signal paths
Tx Streaming Control
Tx Streaming Implementation

- **Simple Signal Source**
  - Signal Source block could not keep up @ 102.4Msp
  - Replaced with constant source & rotator

- **Allows for tweaking flow control settings in real time**
  - Periodic update rate
  - Full/Empty notifications

- **Control-oriented more than signal processing**
  - UDP Rx blocks feed QT Frequency Sink
Tx/Rx Coherency Test

- **Purpose:**
  - Test receive and/or transmit coherency, i.e. measure constant phase offset between channels at a given frequency across tuning cycles
Tx/Rx Coherency – Application
Tx/Rx Coherency Implementation

- Entire application created using GNU Radio Companion
- Test management and data collection implemented in single block
- Async messaging w/ JSON content for measurement information and radio configuration
- Snapshot Transmit Block
  - Generates PN sequence, uploads to radio, and initiates looped playback
Further Development

- Refine and extend the coherency test approach to implement tests for:
  - Receiver gain
  - Noise figure
  - IIP3
  - Gain & NF optimization

- Production test suite
CyberRadio Software Components

- CyberRadioDriver
  - Unified Python API for all of our radios
- libcyberradio
  - Low-level C++ functions for radio data interfaces
- gr-cyberradio
  - Wrappers for CyberRadioDriver and libcyberradio functions
  - Additional GR blocks
- Available on github: [https://github.com/CyberRadio/gr-cyberradio](https://github.com/CyberRadio/gr-cyberradio)
Questions?

NDR358
8 Channel Wideband Digital Tuner

NDR308
8 Channel Wideband Tuner

NDR551
4 Channel Wideband Digital Tuner

NDR154
Multi-Band RF Distribution Module

NDR155
Wideband RF Distribution Module