High-Performance SDR Leveraging Open Source Frameworks - A Development Retrospective

DRS Signal Solutions
Germantown, MD

March 17, 2016

DRS Technologies

SDR Observations

- Software Defined Radio (SDR) systems, in which signal processing components are defined in software, are proliferating due to improved capabilities of hardware and software platforms
- SDR hardware have seen continued improvement in digital processing and RF front end capabilities
- Modern SDR Frameworks are:
  - Catalyzing drastic changes in signal processing
  - Enabling software engineers and signal processing engineers to work in tandem on core challenges
  - Effectively processing large amounts of data in real-time on limited hardware resources.

SDR Hardware

- Wide frequency coverage: 500 KHz – 6,000+ MHz
- Wide bandwidths: up to 100 MHz
- Multiple channels: 1-12
- High Performance I/O: 1 GigE, 10 GigE, Aurora, PCIe, …
- RF front end performance: Superhet and rapidly improving Zero IF RFICs
- Powerful embedded processing options: Zynq, Kintex, Keystone II, …

SDR Frameworks

- Software Defined Radio (SDR) frameworks, in which signal processing components are defined in software, are proliferating due to improved capabilities of hardware and software platforms.
- Not strictly a signal processing challenge
- Requires substantial expertise in fields such as software engineering and distributed computing
- The dual nature of this problem implies two sets of development goals, leading to two types of software frameworks; processing frameworks and system frameworks
- SDR Frameworks in the SIGINT community include:
  - GNU Radio (processing framework)
  - OpenCPI (processing framework)
  - REDHAWK (processing and system framework)

DRS GNU Radio Projects

- PT-325 - Pico Tuner for USRP
  - High performance Picoceptor tuner module mated to the Ettus Universal Software Radio Peripheral (USRP)
  - GNU Radio compatible
  - With minimal software changes GNU Radio users can experience the benefits of a high performance receiver architecture
  - *Ideal for users who’ve implemented radio applications using low end tuner modules in a laboratory setting and now look to take their application to real world environments.*

- PicoFlexor SDR
  - PicoFlexor has full GNU Radio installation
  - Same waveforms can run on host or embedded
  - Embedded PicoFlexor uses DUBRI interface
  - DRS FLEX DDC IP block in FPGA for co-processing improvement
  - Demonstrated P25 waveform on the GNUradio PicoFlexor w/PicoTransceiver load

PicoTransceiver (PicoFlexor w. transmit)

- Polaris HF/UHF/VHF Tuner
  - Demonstrating Host-based 4 Channel FFT display with SI-9150/D4 Polaris

Polaris

- Harrier HF
  - Demonstrating Host-based 4 Channel FFT display with SI-8746 Harrier HF

Harrier

- Talon
  - Demonstrating Host-based 4 Channel FFT display with SI-8646 Talon UHF/VHF

Talon

Open CPI Overview

- Open Component Portability Infrastructure (OpenCPI) is a processing framework designed to address SWaP constrained radio applications
- An open source software (OSS) framework to simplify complexity and enable code portability of real-time systems
- Creates a hardware abstraction layer for embedded real-time systems
- Real-time middleware for embedded systems
- Designed for heterogeneous component-based applications such as: FPGA, GPP, DSP, GPU, Multicore processing
- Open standards compliant uses: US Government's Software Communications Architecture (SCA) Industry standards

PicoTransceiver
(PicoFlexor w.
transmit)

DRS REDHAWK Project

- **REDHAWK Distributed (Server-based)**
  - Created DUBRI device component
  - Created a sample waveform/application utilizing the DUBRI device component
  - Tested sample waveform on Linux server machine using Picoceptor SDR, PicoFlexor SDR, SI-9138 VPX Tuner, and Polaris Tuner
- **REDHAWK Embedded**
  - Cross-compiled REDHAWK framework, BULKIO Interfaces, and FRONTEND Interfaces
  - Created FRONTEND Interfaces 2.0 compliant Picoflexor device
  - Created the sample waveform/application to run natively using the Picoflexor device
  - Tested the FM Radio sample waveform on the REDHAWK-enabled PicoFlexor/PicoTransceiver
  - DRS FLEX DDC IP block in FPGA for significant processing improvement
  - Outcome is 8649A/PF/SW/RH embedded software/firmware load for the SI-8649A/PF PicoFlexor
    - Delivered on an SD card image that boots the PicoFlexor/PicoTransceiver into a Linux system with a full REDHAWK installation

PicoFlexor

Polaris

SI-9138

Vesper
(planned)

DRS Open Source Lab

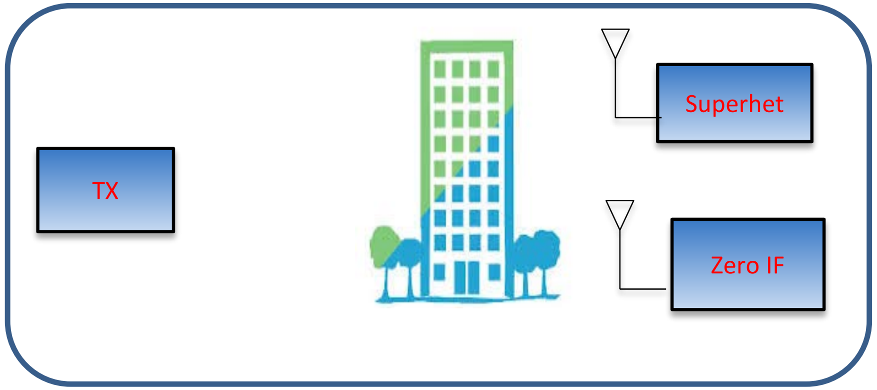
- DRS commitment to Open Standards & Open Source Software development
- Integrate and Demonstrate Open Framework based Apps (GNU Radio, REDHAWK, etc)
  - Demonstrate product functionality
  - Demonstrate flexibility of our products
- Provide an open lab for customers to:
  - Feasibility studies
  - Port and test their Apps
  - Obtain hands on training
- Demonstrate Open Standards Interoperability
  - VITA49 (an analyzer, connected systems, etc)
  - VITA 67 (chassis)
- Quick Reaction Capability (QRC)
  - Rapid prototyping/evaluation of new technology (aka Zynq 7045, Keystone II, Thunderbolt, etc.)
- Evaluate Emerging Embedded Open Source/Open Source Technologies

QRC Projects

- P25 Transceiver
- Point to Point Tactical Communications Demo (Comms Demo)
- Spectral Monitoring
- FM Radio
- DRS Framework Integrations
  - GNURadio (Pico Tuner, PicoFlexor, Polaris, Harrier, Talon, …)
  - REDHAWK (Picoceptor, PicoFlexor, SI-9138, Polaris, Vesper, …)

Comms Demo Project

- Purpose: Evaluate Superhet versus Zero IF radio performance over distance
- Goal: Transfer a 2 MByte file for 3 km in less than 20 seconds in dense urban environment using low SWaP hardware
- Chose GNURadio framework
  - Started with BPSK from repository
  - Ended up using GMSK from repository
- Selected DRS PicoFlexor, DRS Polaris and COTS Radios
- Field performance evaluated at 200 KSPS
- Operated in 400 MHz, and 900 MHz bands



Comms Demo Project Testing

- Lab based integration and test
  - Self Loop
  - Wired
  - Wireless

- NPR
  - What happens when tuners have RF inputs of real world spectrums with hundreds and even thousands of signals? Think 10Log #signals = average power
  - NPR best duplicates the world spectrums with a high level noise spectrum and tunable notch. The unit under test (UUT) is tuned to the notch. The noise spectrum is raised and any intermodulation filling the notch is compared to the Out-of-Notch noise as noise power ratio
  - Our telephone and Comms companies have used NPR tests to verify long hall, satellite and ground communication links for over 75 years

- Outside Testing
  - Parking lot in Germantown, MD
  - Baltimore, MD - Inner Harbor Area

Project Integration Challenges

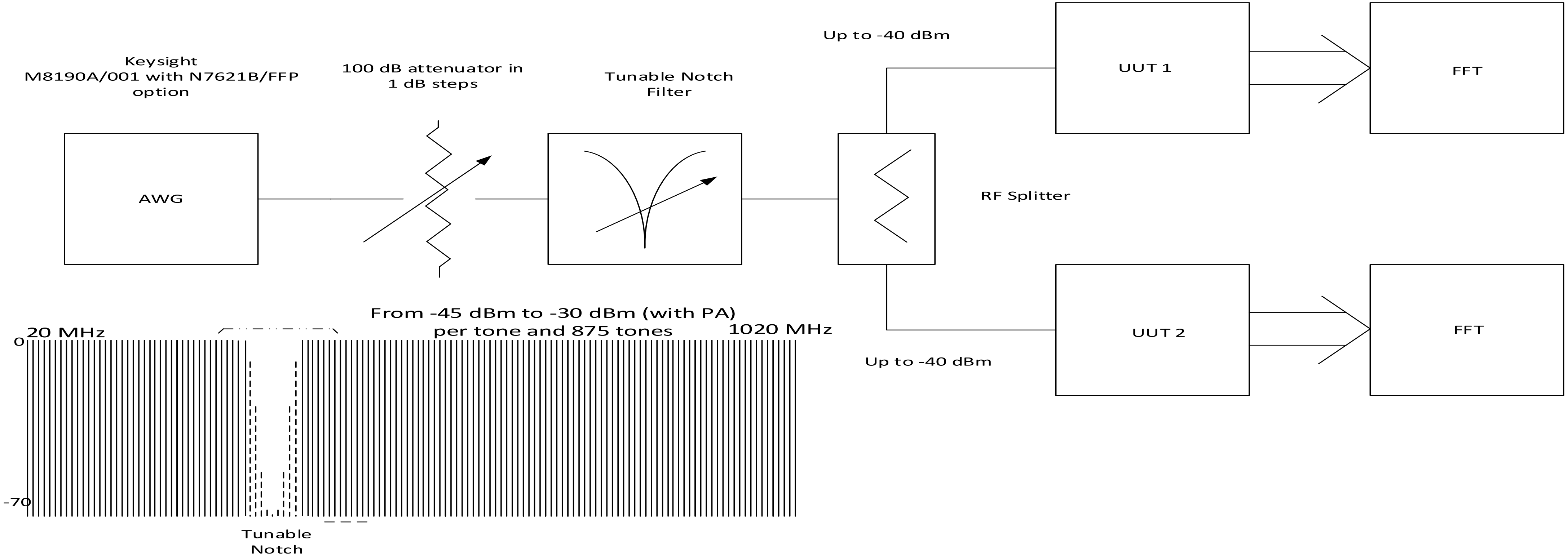
- Demodulators – QPSK, BPSK QAM16, GMSK, …
  - Only GMSK worked successfully over the air
- Forward Error Correction
  - Required significant processing resources
- Processing performance limitations
  - Embedded SDR
  - Laptop computer
- Integrating diverse blocks from public repository required effort

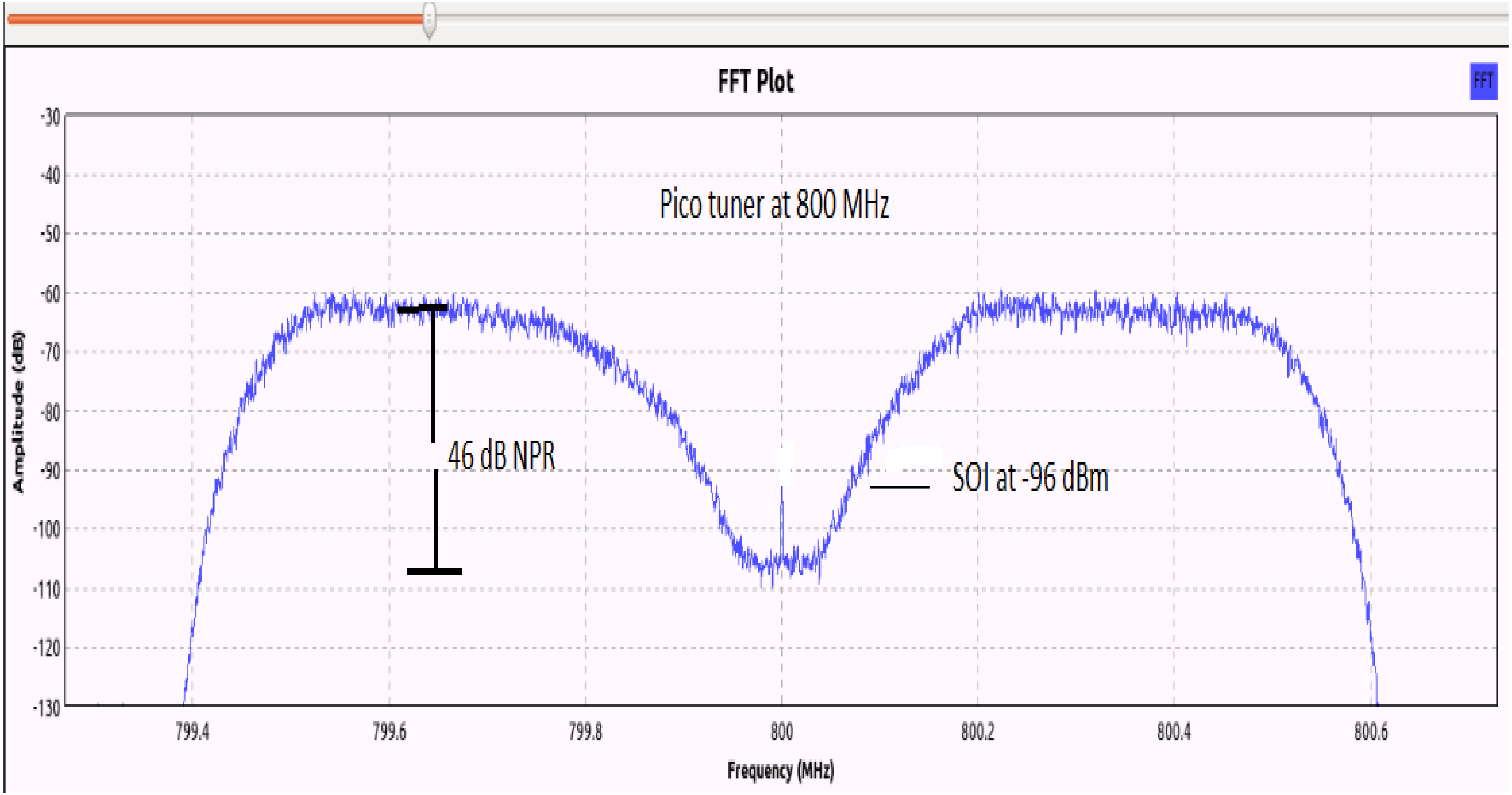
NPR Testing Sequence

- In-band/out-of-band NPR with noise
- In-band/out-of-band NPR with multi-tone signals at low levels
- Out-of-Band NPR with multi-tone signals at high levels

NPR Testing

- **Multi-tones Inside/outside IF/ADC BW**
- Average RF power of 10Log 1000 tones = 30 dBm added to -40 dBm
- Total average power is -10 dBm
- Statistical peak power is another 10 dB or -0.0 dBm

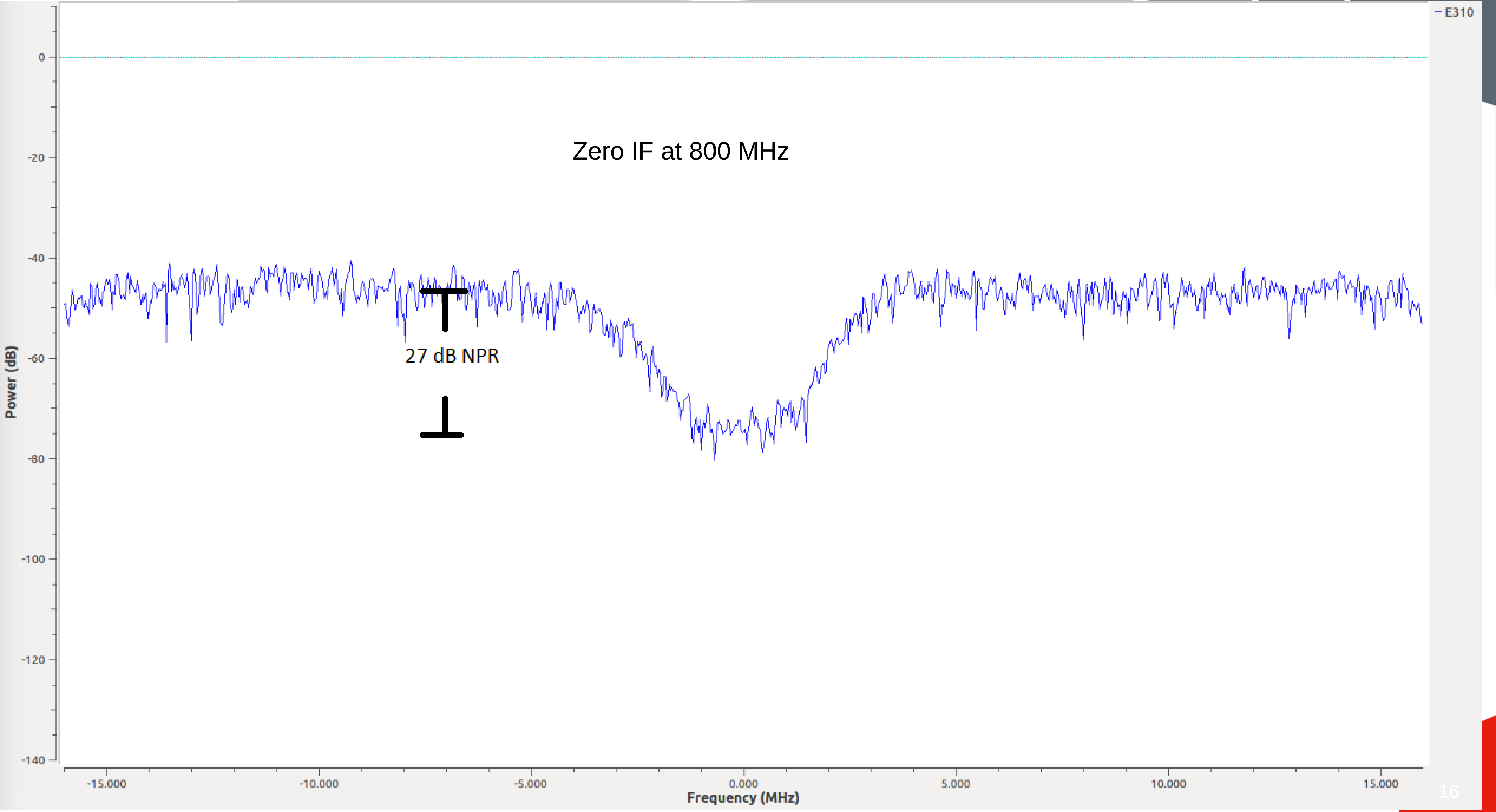

Keysight
M8190A/001 with N7621B/FFP option
AWG
100 dB attenuator in 1 dB steps
Tunable Notch Filter
RF Splitter
Up to -40 dBm
UUT 1
FFT
UUT 2
FFT
Up to -40 dBm
From -45 dBm to -30 dBm (with PA) per tone and 875 tones
20 MHz
1020 MHz
0
-70
Tunable Notch



Pico tuner at 800 MHz

46 dB NPR

SOI at -96 dBm



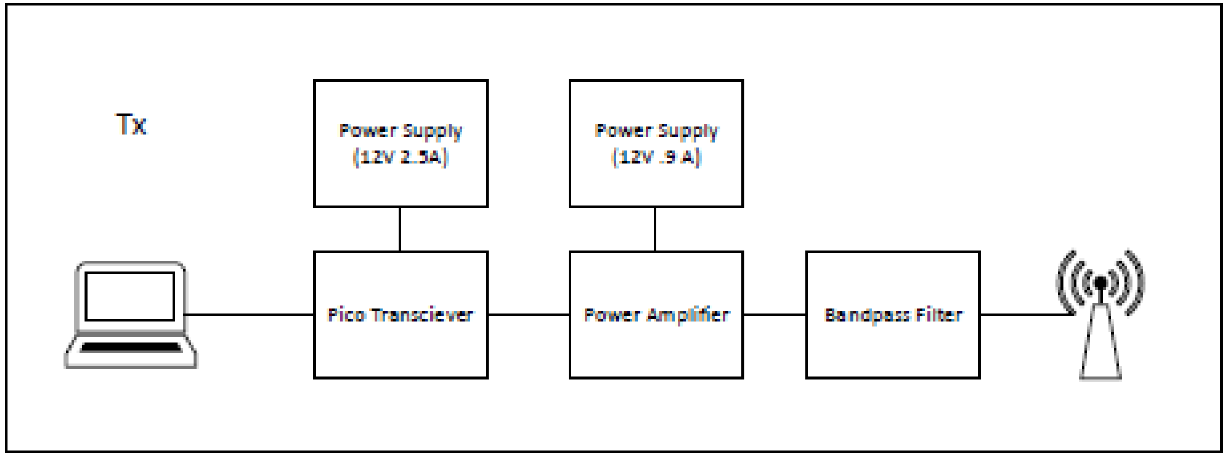
Zero IF at 800 MHz

27 dB NPR

Baltimore City Testing

- Test the performance difference between DRS & Zero IF radio communication in a challenging real world environment.
- Both systems were tested at fixed locations at various ranges to compare the results.
- The receivers were located at a single position and the transmitter moved to various locations.
- Tested in the 400 MHz and the 900 MHz bands

Outside Test Setup

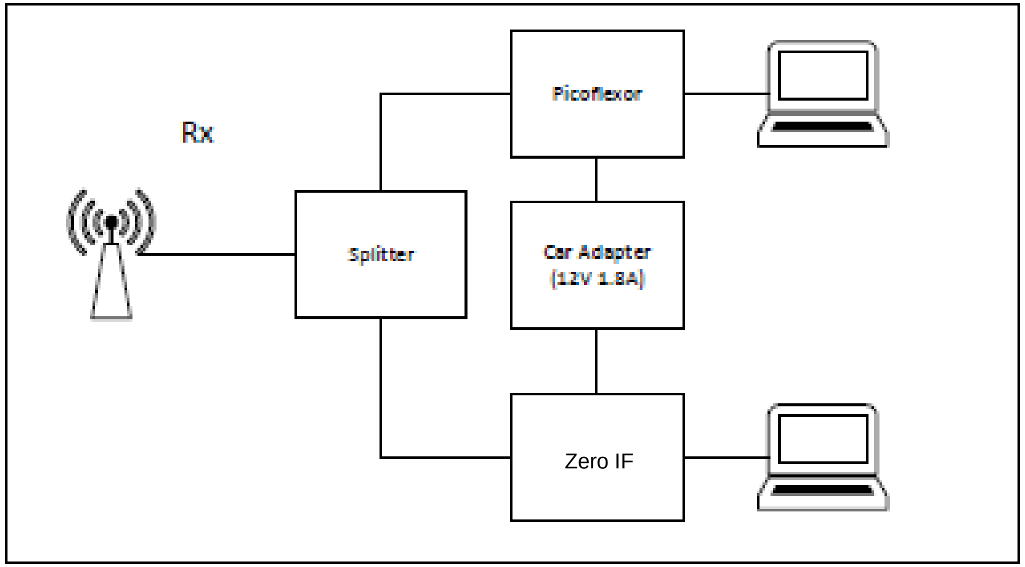


Transmitter

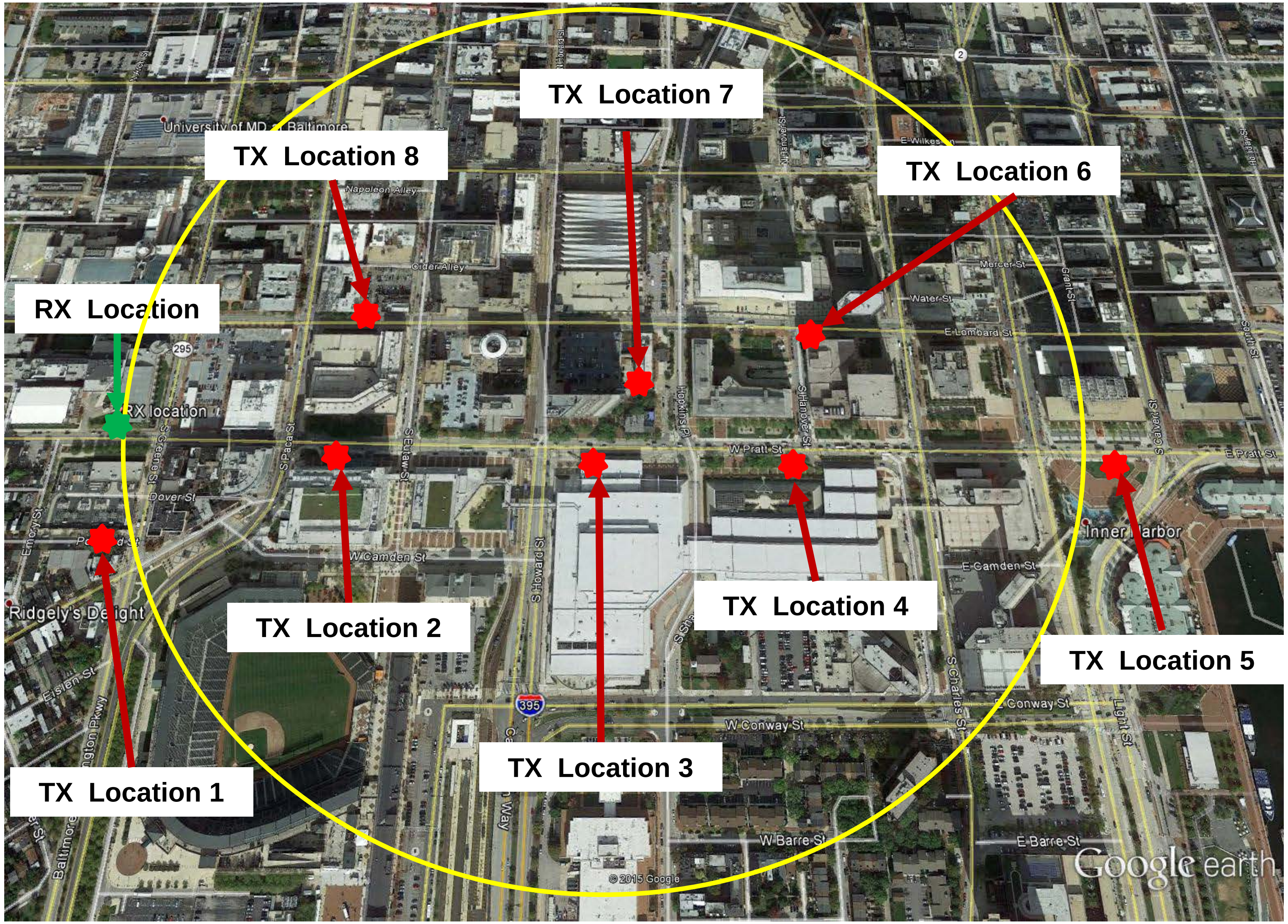
DRS PicoTransceiver

Receivers

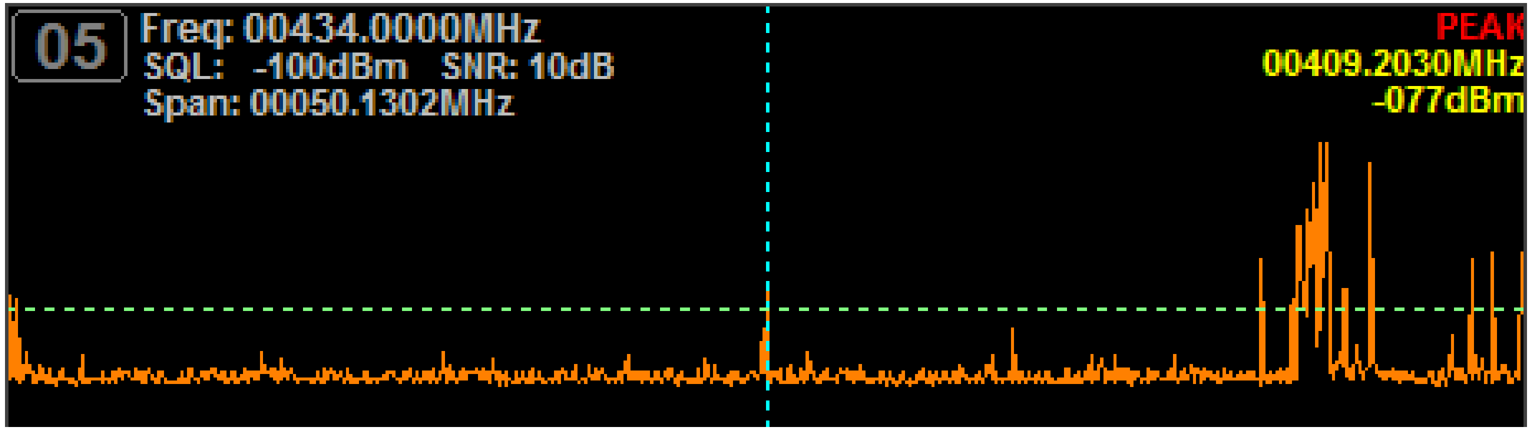
DRS PicoFlexor	Superheterodyne
COTS	Zero-IF



Test area (diameter 1 Km)



400 MHz RF environment (measured at RX location)



05
Freq: 00434.0000MHz
SQL: -100dBm SNR: 10dB
Span: 00050.1302MHz

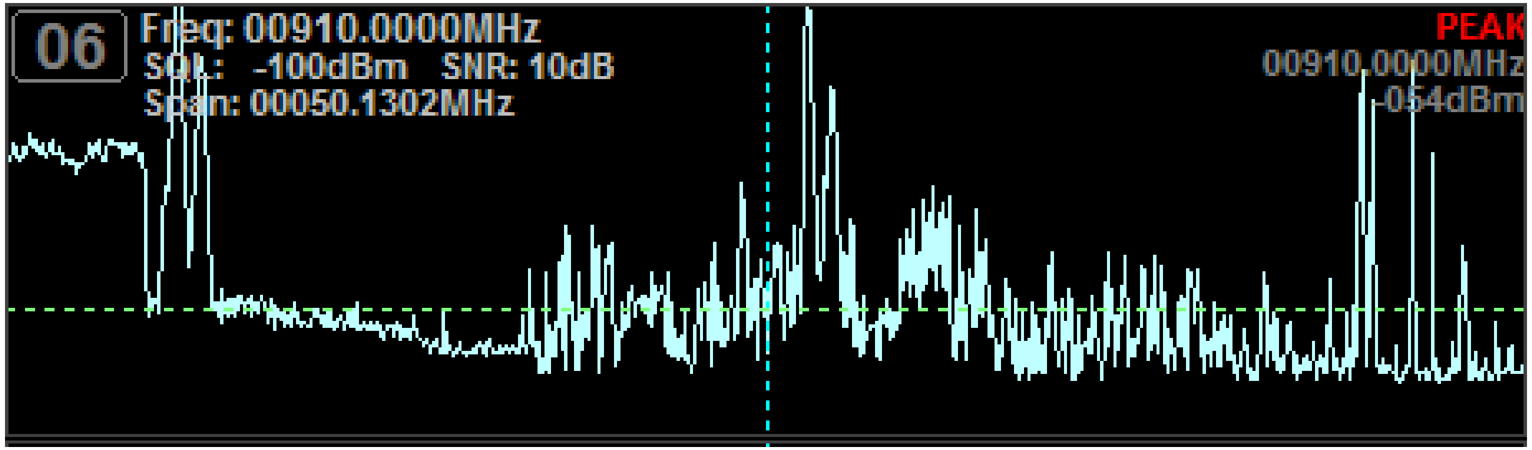
PEAK
00409.2030MHz
-077dBm

Test Results

TX Location	Distance (meters)	LOS (Line of Sight) Path	Power (Watts)	Frequency (MHz)	DRS PICO % success	Zero IF % success
434 MHz						
1	100	unblocked	0.74	433.92	100	100
2	250	unblocked	0.74	433.92	100	100
3	500	unblocked	0.74	433.92	100	100
4	700	unblocked	0.74	433.92	88	49
5	1050	unblocked	0.74	433.92	100	100
6	720	blocked	0.74	433.92	100	100
7	550	blocked	0.74	433.92	100	100
8	300	blocked	0.74	433.92	100	100

The results shown were compiled from multiple data collects.

900 MHz RF environment (measured at RX location)



Freq: 00910.0000MHz
SQL: -100dBm SNR: 10dB
Span: 00050.1302MHz

00910.0000MHz
-054dBm

Test Results

TX Location	Distance (meters)	LOS (Line of Sight) Path	Power (Watts)	Frequency (MHz)	DRS PICO % success	Zero IF % success
910 MHz						
1	100	unblocked	0.74	910	100	0
2	250	unblocked	0.74	910	100	0
3	500	unblocked	0.74	910	97	0
4	700	unblocked	0.74	910	98	0
5	1050	unblocked	0.74	910	83	0
6	720	blocked	0.74	910	0	0
7	550	blocked	0.74	910	12	0
8	300	blocked	0.74	910	96	0

The results shown were compiled from multiple data collects.

Summary of Test Results

- The results show there is a significant performance difference between the two systems.
- The performance of the superhet design is more consistent but there are locations when neither system could establish the communication link.
- The DRS system was able to operate in an extremely dense and harsh RF environment (900 MHz).
- Both the Superhet and Zero IF performed similar in a less crowded part of the RF spectrum (400 MHz).
- The purpose of the test was to put both equipment suites in a harsh congested environment and collect data to determine any performance differences.

Polaris GR Integration Project

- Improved performance of Polaris GNURadio block through threading and buffer optimizations:
- Performance a multi-threaded architecture was implemented focusing on constant data processing running in separate dedicated threads and a transfer of available information on demand.
- The application's performance was further improved by using aligned buffers to prevent cache misses and by tweaking code in several locations to make use of micro-optimizations.
- Currently the application can handle 192 MSPS and we are continuing to push the performance even further. All tests run on the block were run on an Ubuntu 14.04 LTS with a Xeon E5-2609 processor and GNURadio 3.7.
- In order to improve capture performance of packets, drivers for the 10GigE NIC card were compiled with certain settings to optimize

Conclusion

- GNURadio repository was indispensable for reducing development time but blocks in public repository required work to meet real-world conditions
  - Advantages
    - Leverage open source code
    - Rapid prototyping, decreased development time
  - Disadvantages
    - Weak demodulator, no forward error correction
    - Hindrance on performance. But limitation, impacted both receivers equally
- NPR is a valid representation of the real world but field testing is the final proof of system quality

Kerry Rye
Senior Director of Engineering
DRS Signal Solutions
301-944-8359
krye@drs.com

THANK **YOU** FOR YOUR ATTENTION

DRS Technologies