2012 14th International Conference on Ground Penetrating Radar (GPR)
June 4-8, 2012, Shanghai, China @ ISBN 978-1-4673-2663-6

Software Defined Radar: An Open Source Platform for Prototype GPR Development

Jonathon Ralston and Chad Hargrave
Mining Technology Research Group
CSIRO Earth Science and Resource Engineering
Queensland, Australia
Jonathon.Ralston@csiro.au

Abstract— **This discussion paper explores the potential of Software Defined Radio (SDR) technology to provide flexible and low-cost subsurface radar prototypes for the GPR community. Unlike traditional fixed hardware implementations, SDR uses software configurable RF modules which can be used to implement customized signal encoding, decoding and processing. However, the full potential of SDR has not yet been fully understood or exploited for radar-based applications, and so is of special interest for GPR development. This paper introduces the fundamental concepts behind SDRs and describes the underlying hardware and software architectures that are used to implement them. It also provides a simple reference design using open source GNU Radio software and the Universal Software Radio Peripheral (USRP) hardware to indicate how low-cost radar configurations can be designed and evaluated. The benefits and challenges of SDR-based radar architectures are highlighted, and opportunities discussed for new and enhanced subsurface sensing capabilities. Overall, SDR technology is seen to provide new opportunities to boost future radar research and development to provide enhanced GPR system capabilities.**

Keywords- GPR, Software Defined Radio (SDR), GNU Radio, Universal Software Radio Peripheral (USRP), Open Source.

I. INTRODUCTION

Ground penetrating radar (GPR) technology has been successfully deployed as a means to address important sensing problems requiring the detection, imaging, and identification of subsurface structures. As a result, GPR has found special interest in the earth sciences and resource engineering domains, with a broad range of applications in archeology, geophysics, hydrogeology, mineral mining, ordinance detection, civil engineering, and non-destructive testing [1-3].

One key aspect that has contributed to the wide breadth of successful GPR applications has been the ready access to commercially available equipment. The vast majority of GPR systems in use today employ fixed RF electronics to implement impulse-based radar, which is an approach that has provided a useful instrument with simple operating characteristics.

A. Potential Limitations with Fixed Designs

While successful, this impulse radar design is now several decades old and, despite being widely applied, shows specific limitations arising from its fixed hardware implementation. The major limitation is that it offers little or no provision to modify radar the operating characteristics according to the needs of a given application.

Further, most fixed-hardware radar systems also impose a number of important operational constraints which tend to limit the achievable sensing performance. These include:

1. Little or no knowledge of the actual transmitted signal
2. Little or no control over signal encoding or decoding
3. No control over transmitted power or bandwidth utilization, including adaptive signal generation
4. Little or no provision for developing customized RF signal processing stages.

These issues may inadvertently introduce additional signal clutter, obscure target discrimination, and degrade range resolution – all aspects that impact on the degree of inference that can be confidently made using the received GPR signal. Despite the drive and interest for improved capabilities, there are currently few practical options to develop new GPR implementations. This situation is further complicated by the high-level of expertise (and expense) that is typically required when undertaking development of custom radar equipment.

B. An Alternative Flexible Implementation

In an attempt to provide solutions these limitations, this paper explores the use of Software Defined Radio (SDR) technology to develop flexible and low-cost subsurface radar prototypes. SDR is an emergent wireless paradigm in which some or all of the physical radio-frequency (RF) functions are implemented in software rather than fixed hardware. Most interest in using SDRs has focused on implementation of wireless communications protocols rather than radar (ranging) applications, thus the notion of using SDR for GPR is rare.

An outline of the paper follows. Section II introduces the concepts behind SDR and explains the generic hardware and software architectures. Section III identifies a specific SDR implementation based on open source technologies using GNU Radio software and Universal Software Radio Peripheral (USRP) hardware, Section IV provides a simple SDR example implementation, and Section V raises issues and opportunities associated with SDR-based GPRs.

II. SDR CONCEPTS

SDR is an emerging concept that aims to enhance the flexibility of existing wireless systems by transforming "hardware problems into software problems". SDRs are re-programmable devices in which some or all of the physical RF functions are implemented in software rather than fixed hardware [11].

A. SDR History and Applications

The concept of a software-implemented radio first arose in the early 1970's through defense-related research and development [4], and later emerged into the public domain where the term SDR was officially coined [5]. Since this time, much SDR research has focused on software-based radio architectures to implement radio communications functionality. Successful applications of SDR include AM/FM radio receivers [6], GPS receivers [7], RFID readers [8], wireless communications protocols, and cognitive (adaptive) radios [9].

Most SDR research activity has focused on replicating wireless *radio* communications, which are typically more narrowband in operation. Applications of SDR for two-way *radar* ranging are considerably less common [11], and GPR applications are largely unreported. The intention of this paper is to raise awareness of SDR-based technologies and thus benefit the GPR research community.

B. Idealised SDR Architecture

An idealized SDR architecture is shown in Fig. 1. It uses a high-speed digital-to-analogue convertor (DAC) and a wideband power amplifier for the signal transmission path, and a low-noise wideband amplifier and high-speed analog-to-digital converter (ADC) for the receive path. All signal encoding and decoding functions are implemented by a high-speed digital processor. Typically, an external device then interfaces to the higher-data rate digital processor to provide display, signal monitoring, and mode control functionality for the SDR.

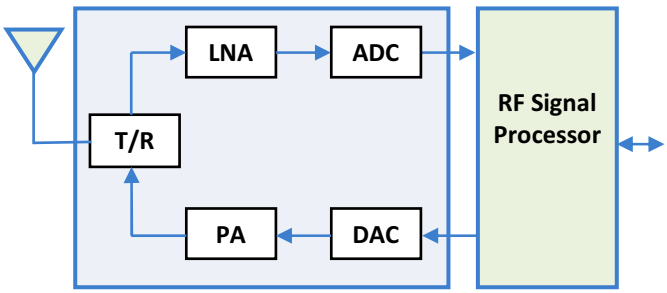

Figure 1. An idealised architecture for implementing SDR-based devices.

C. Practical SDR Implementation Considerations

The idealized SDR implementation as shown in Fig. 1 is conceptually simple, but imposes a very high sampling rate requirement as well as huge signal data throughput on the overall design. As a result, this design tends to be difficult (and expensive) to directly implement in practice, particularly for systems with medium to high frequency requirements or large operational bandwidths. In practice, most SDRs use front-end RF mixers to modulate the baseband transmission signal to the required radio frequency, and to down-convert the received RF signal using the super heterodyne principle. An emerging trend in SDR transceiver design is to integrate both the RF front end and signal processing circuits onto a single chip; such designs are likely to become more commonly available in the future for broadband wireless applications [10]. The antenna itself remains a hardware element and generally requires specialized RF design. As a result, most practical SDR-based designs focus on the RF signal processing chain and retain details of antenna design and implementation as a separate task [13].

D. Requirements for an SDR-based GPR

Conventional GPR designs require a large effective RF bandwidth in order to achieve good target range resolution. For impulse, swept frequency carrier wave (SFCW) and pseudo-random noise GPR designs [1-3], fixed RF hardware is used to implement radar signal encoding and decoding. Here, the data throughput is effectively reduced by using equivalent-time sampling (impulse), heterodyne mixing (SFCW), or direct correlation (pseudo-random noise) implemented directly in silicon, respectively.

In order to develop a radar ranging instrument with useful subsurface imaging performance, an SDR implementation must provide sufficient spectral bandwidth, timing accuracy, and power budget for the given application. These three key requirements are now examined in turn.

Firstly, the bandwidth requirement presents a practical challenge as the majority of SDR applications focus on implementing wireless communications, which tend to be relatively narrowband. Most practical SDR implementations use a hybrid approach that employs a software-tunable bandpass filter at the RF front end. The filter bandwidth is selected to match the bandwidth specifications of the given ADC and DAC, as well as the overall digital communications data rate to and from the host processor data. In this way, larger spectral bandwidths can be effectively synthesized by aggregating the individual spectral sections.

Secondly, the SDR must be able to transmit and receive data with relatively accurate timing in order to achieve reasonable ranging accuracy. The need for accurate signal time-coherence and time-synchronization places specific requirements on the host processor to provide high-speed, high-accuracy timing and control functionality. For GPR applications, the underlying digital data processing and information rate precludes the use of general purpose PC or embedded systems, and customized but programmable FPGA hardware has emerged as a logical choice to fulfill this requirement.

Finally, in terms of signal penetration and ranging performance, a SDR must have sufficiently large transmit power and receiver sensitivity to provide a suitable overall power budget for the given application. In most cases, this aspect can be achieved through the use of readily available wide band, low-noise and power RF amplifiers.

III. An Open Source SDR System

A. Motivation for Open Source Technologies

A fundamental requirement for researching with SDR-based GPR prototypes is the availability of SDR hardware and software development environments. While a number of commercial SDR configurations exist, most tend to be highly specialized, expensive, and generally utilize proprietary software that cannot be modified or distributed.

There are however some very powerful open source technologies that can be used to implement SDR functionality. The benefits gained by adopting an open source approach are significant, particularly during initial research and evaluation phases. These benefits include access to underlying reference designs, the ability to freely use or modify any aspect of hardware or software, opportunities to explore innovative concepts and technologies, and support from the developer community. A specific SDR implementation is now described using the open source Universal Software Radio Peripheral (USRP) hardware and GNU Radio software.

B. The USRP SDR Hardware Device

The USRP device is an open hardware platform that has been specifically designed for developing custom SDR configurations. It was developed by Ettus Research LLC [15] with the intention of providing a flexible and low-cost SDR research platform. It has proven popular for implementing advanced cognitive radio designs [9].

Fig. 2 shows a block diagram of the USRP hardware. The modular design uses pluggable RF daughterboard modules that interface with the main processor motherboard. This motherboard also provides communication link to an external PC for monitoring and control. The motherboard is designed to manage the all the high-speed processing processes such as up/down conversation, decimation, and filtering.

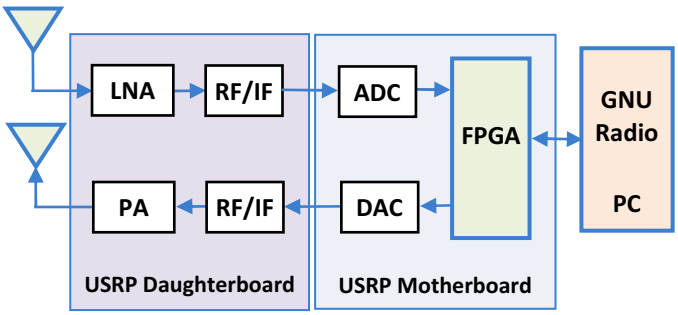

Figure 2. A practical SDR implementation architecture based on USRP hardware and GNU Radio software.

Fig. 3 shows the USRP2 motherboard hardware which comprises a Xilinx Spartan 3-2000 FPGA for main processing, two 100 MHz 14-bit ADCs, two 400 MHz 16-bit DACs, and an embedded Linux operating system with Gigabit Ethernet interface. The USRP can support up to two receive and two transmit daughterboard modules. Each module has access to the high speed ADC and DACs, and can be operated in either real or complex IQ sampling modes.

Figure 3. Internal view of the USRP2 motherboard showing FPGA, DACs, ADC, and daughterboard module interface.

C. The USRP Daughterboards

The USRP motherboard architecture uses interchangeable RF daughterboard modules for the RF front end. Several modules exist to provide RF modulation and demodulation in various transmit, receive and transceiver configurations.

Fig. 4 shows a typical USRP RF daughterboard module (WBX) which provides a software configurable transceiver. This module provides 40 MHz of bandwidth capability but can modulated to operate over a frequency range of 50 MHz–2.2 GHz at 100 mW of output power. The receiver and transmit chains of the WBX operate independently. This particular module is ideal for prototyping GPR applications with the wide operational range.

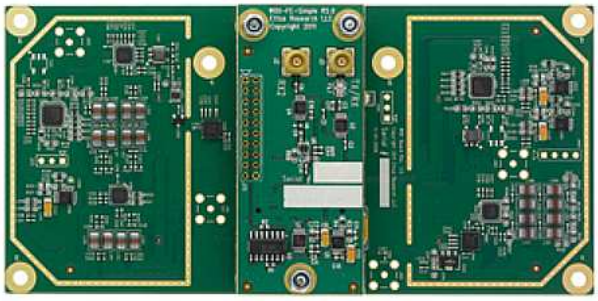

Figure 4. The USRP WBX 50-2200 MHz Rx/Tx daughterboard. The wide operational range of this RF module makes it ideally suited for the development of GPR prototype configurations.

The USRP SDR is open hardware technology design. The schematics and PCB layouts for the USRP motherboard an daughterboard schematics, as well as FPGA software, are freely available and open for access and modification [12,15]. This makes the USRP an attractive platform for low-cost radar research and prototype development.

D. Open Source GNU Radio Software

A growing number of software packages are designed to develop software for USRP devices. Of these, the open source

GNU Radio package [16] has possibly the largest development base and has been widely used in hobbyist, academic and commercial fields to develop wireless communications devices. It can be deployed with a variety of hardware devices, or without hardware in a software simulation environment. GNU Radio is freely available and is licensed under the GNU General Public License (GPL) version 3.

GNU Radio is an open-source software development environment that provides the means to develop real-time RF applications [10]. It is written using C++ and Python, and programs are compiled and run on most general purpose processors supporting Linux, Mac OSX, or Windows.

GNU Radio is also the software of choice recommended by Ettus for implementing SDRs using the USRP [15]. GNU Radio software and USRP hardware are often used together to implement SDR functionality [15], although it should be noted that GNU Radio is far more general in design and can interface with a large number of other hardware devices.

IV. Demonstration SDR Implementation

A. SDR Development Environment

The GNU Radio package includes programming interfaces and runtime environments to implement the desired RF signal processing functionality, applications for creating signal flow graphs, and hardware drivers to access USRP devices [16]. It uses the concept of signal processing "blocks" to achieve the required the signal flow and processing functionality. This functionality includes signal input (sink) and output (source) blocks such as sinusoidal generators, noise generators, modulators, spectral operators, arithmetic processing, and so on. GNU Radio supports well over 100 signal processing blocks which are implemented C++ and custom blocks can also be developed.

These signal processing blocks are coded to create the desired RF signal flow (See Table 1). The signal processing blocks are written in C++, while creating flow graphs and connecting signal blocks is done in Python [15].

Table 1: Example of a typical GNU Radio flow-graph code to implement signal processing blocks for the USRP SDR.

```
#import blocks from the gnu radio package
from gnuradio import gr

#create the top block
tb = gr.top_block()

#create a signal source
src = gr.null_source(1)

#create a signal sink
sink = gr.null_sink(1)

#connect the source to the sink
tb.connect(src, sink)

#run the flow graph
tb.run()
```

B. Graphical Development Enviroment

To facilitate SDR development, an open source application has also been developed that provides a graphical interface to GNU Radio, called GNU Radio Companion [17]. This drag-and-drop graphical development tool is designed to relieve the manual coding effort in by creating signal processing blocks and generating the flow-graph source code. A typical example of this approach is shown in Figure 5.

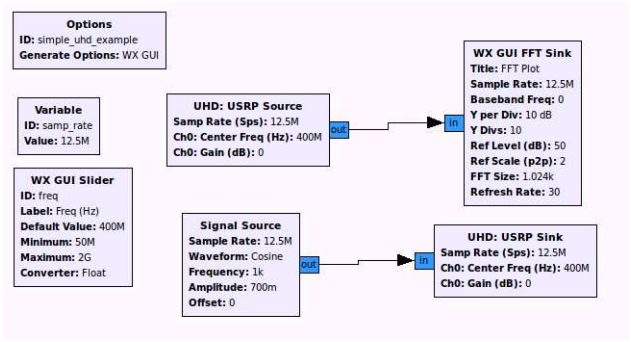

Figure 5. Example of the GUI-based GNU Radio Companion for creating signal processing blocks to implement the desired RF functionality.

C. SDR Example Implementation

As a demonstration of the SDR concept, a prototype configuration was constructed using the USRP2 motherboard and LF 50MHz daughterboard module for the RF front-end. The focus of this reference design is on the transmission and acquisition of RF signal data from a generic PC as shown in Fig. 6.

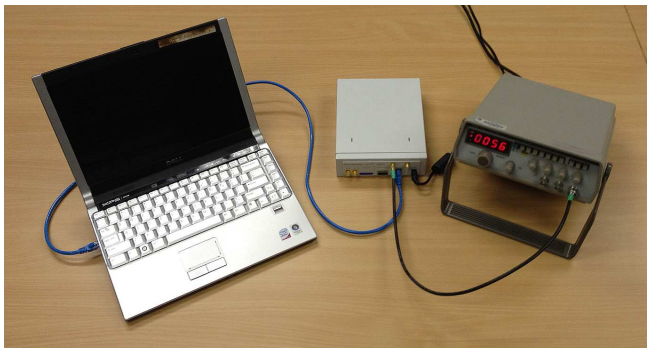

Figure 6. Example of the GUI-based GNU Radio Companion for creating signal processing blocks.

Here a signal generator was used to provide a sinusoidal signal source into the Rx channel of the daughterboard in order to demonstrate the software receiver capability. Fig. 7 shows a signal acquired from the USRP and RF module using GNU radio operating on the PC.

Scope clearly exists for the creation of more sophisticated configurations from within GNU Radio framework such as customized waveforms generation for RF transmission and processing. This simple example provided here based using USRP and GNU Radio software indicates the possibilities rapidly develop of new GPR prototype designs and implementations.

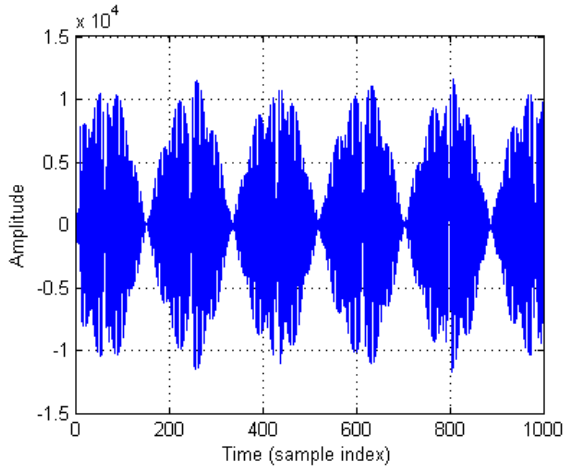

Figure 7: RF waveform acquired using the USRP and GNU Radio software from an external PC.

V. DISCUSSION

A. Advantages of SDR

SDR technology provides a powerful architecture to facilitate the development and evaluation of new GPR configurations. Perhaps the most immediate benefit is the flexible signal generation and processing capabilities, which would allow the software GPR system to be tuned to characteristics of the target environment. As SDR technology continues to develop, more general purpose, high performance modules will become available which could be used to further implement new and advanced radar functionality.

As the hardware and software domains further converge, it also provides new opportunities to better integrate expertise in antenna design, RF implementation, signal processing, EM modeling and subsurface interpretation. Overall these factors will lead to improved GPR performance and capability.

B. Maturity of SDR Development Environments

The notion of translating "hardware functionality into software code" as promoted by the introduction of SDR technology has the effect of imposing new software capability requirements onto radar implementer and researchers. At present, SDR hardware options appear far more mature than the associated software packages required to implement them.

Whilst the core SDR interfacing tools are relatively robust and implemented using modern programming paradigms (e.g., see [12]), the overall environments provided for code development tend to lack integration and robustness. An issue is the high level of software expertise that is required to identify, install, configure and properly utilize SDR technology, which is likely to be beyond the abilities of the non-programmer.

This is effectively an SDR accessibility issue that needs to be addressed before the full potential of SDR can be realized by members of the broader GPR community. Opportunities therefore exist for the development of simpler packages to facilitate rapid development of SDR-based radar designs.

C. A Practical Comment on Throughput Limits

The low-cost USRP-based SDR hardware described in this paper supports a maximum throughput RF streaming bandwidth of 50 MHz to the host controller application. To realize this bandwidth throughput, the connection from the SDR to the controlling PC relies on a gigabit Ethernet communication link, which suggests that some practical limits are being approached with this entry-level SDR. It also means that synthesizing larger bandwidths will introduce some latency into the data processing stages. Whilst not a major concern for this initial research and evaluation stage, it may serve to limit the applicability of the approach in some applications.

Practically speaking, the current bandwidth and throughput capacities suggest that the current lower-cost SDRs would find more immediate application in lower frequency, deeper imaging GPR applications or in situations where some additional signal acquisition latency is tolerable to synthesize-and-process the desired transmit and receive signals.

It is possible to identify higher cost, custom SDRs which provide higher overall system performance. However, it should be noted that rapid improvements in SDR hardware capabilities continue to occur. For example, in 2006, a typical ADC/DAC RF bandwidth specification was around 5 MHz; in 2011, bandwidth specifications of around 50 MHz are common, all in lower cost USRP devices [15]. This trend is likely to continue and so provide enhanced opportunities for GPR research and design.

D. Adaptive Signal Generation

The SDR platform provides many opportunities to improve fundamental radar imaging performance over fixed-hardware designs. For example, common receiver saturation effects could be minimised by intelligently managing gating timing. The transmitted signal could be dynamically adjusted according to the received scattering response signal to reduce clutter suppress strong reflections and null other common nuisance effects. SDR could also provide very useful mechanisms to assist in rapid auto-calibration processes to improve the overall ranging accuracy.

E. Standards and SDR Reference Designs

At present, there are no clear standards or commonly accepted reference designs for implementing SDR, although some possible specifications have been suggested [13,14,16]. However, USRP and GNU Radio have provided a very useful reference design which has gained support from the research community, and so is likely to remain the de-facto standard for the immediate future.

The benefits of open interoperability standards well accepted and will provide important as new SDR implementations are proposed and investigated. There are also important opportunities for standardization which will allow GPR vendors to develop modular generic radar platforms which would boost reliability and reusability.

F. *Anticipated Future Applications of SDR-based GPRs*

There are many interesting future applications of SDR-based GPRs that would positively impact the way in which GPRs are developed and deployed in the future. Immediate applications would replicate current fixed hardware filtering circuits at the RF stage with software controllable filtering.

Future applications of SDR for GPR research and development are likely to include:

1. Enhanced access to simple, low-cost GPR developmental systems for education and training
2. Radars that preview RF spectrum utilization in the immediate vicinity of operation, and then dynamically configure transmitted signal to optimally utilize the available bandwidth or local regulations
3. Tunable radar systems that are able to adaptively scan specific RF bands, exciting particular frequencies of a desired target structure (e.g., [18]) in order to develop a sub-surface target characterization capability
4. Custom software packages that could allow non-specialist users access to low-resolution "sub-surface detection" devices for simple GPR tasks, or pre-survey inspections.

VI. CONCLUSIONS

This paper has explored the potential of SDR technology as a means to provide new, low-cost subsurface GPR prototypes. SDRs implement RF system functionality through software programmable processors rather than through fixed hardware designs. SDR provides an ideal platform for developing novel and improved GPR implementations, and thus are likely to be of special interest to educators, researchers and commercial radar manufacturers. In order for SDR-based GPR to enter mainstream use, the challenge is to provide SDR systems with well developed software environments as well as hardware with sufficient bandwidth, timing accuracy, and power budget in order to deliver an instrument that has a useful subsurface imaging performance. However, the current availability of low-cost, open source software and hardware, together with ever improving performance characteristics, provides a rich area for future GPR system development.

ACKNOWLEDGEMENTS

The authors would like to thank members of the CSIRO Mining Technology Research Group for their assistance in developing this paper. In particular the authors wish to thank Alex Pitt for his invaluable contributions in identifying and configuring the software development environment used for SDR prototyping and validation.

REFERENCES

[1] H. M. Jol, Ground Penetrating Radar: Theory and Applications, Elsevier, Amsterdam, 2009.

[2] D. J. Daniels, Ground Penetrating Radar, 2nd ed., IEE Radar Sonar Navigation and Avionics Series, London, 2004.

[3] A. P. Annan, "The history of ground penetrating radar", *Subsurface Sensing Technologies and Applications*, Vol. 3, No. 4, pp. 303–320, October 2002.

[4] P. Johnson, "New Research Lab Leads to Unique Radio Receiver," E-Systems Team, Vol. 5, No. 4, pp 6-7, 1985.

[5] J. Mitola, "The Software Radio," IEEE National Telesystems Conference, 1992.

[6] D. Iancu, J. Glossner, Y. Hua, Y. Abdelila, S. Stanley, "Reduced complexity software AM radio, "*Joint First Workshop on Mobile Future and Symposium on Trends in Communications, 2003. SympoTIC '03*, pp. 122- 125, 26-28 Oct. 2003.

[7] K. Voet and W. Van Moer, "Design of a software-defined radio for use in the IEEE L- and S-band," *Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE*, pp.1-4, 10-12 May 2011.

[8] D. De Donno, F. Ricciato, L. Catarinucci, and L. Tarricone, "Design and applications of a Software-Defined listener for UHF RFID systems," *Microwave Symposium Digest (MTT), 2011 IEEE MTT-S International*, pp.1-4, 5-10 June 2011.

[9] R. Zhou, Q. Han, R. Cooper, V. Chakravarthy, and W. Zhiqiang, "A Software Defined Radio Based Adaptive Interference Avoidance TDCS Cognitive Radio, "*2010 IEEE International Conference on Communications (ICC),* pp.1-5, 23-27 May 2010.

[10] W. H. Chen, G. Liu, B. Zdravko, and A. M. Niknejad, "A highly linear broadband CMOS LNA employing noise and distortion cancellation," IEEE Journal of Solid-State Circuits, vol. 43, no. 5, pp. 1164–1176, May 2008.

[11] L. K. Patton, *A GNU Radio Based Software-Defined Radar*, Master's Thesis, Wright University, April 2007.

[12] Ettus Research UHD. Website. github.com/EttusResearch/UHD-Mirror [accessed Feb 2012].

[13] P. B. Kenington, RF and Baseband Techniques for Software Defined Radio.Boston, MA: Artech House, 2005.

[14] P. S. Hall, P. Gardner, and A. Faraone, "Antenna Requirements for Software Defined and Cognitive Radios," *Proceedings of the IEEE*, no.99, pp.1-9, 2012.

[15] Ettus Research LLC. Website. www.ettus.com. [Accessed Dec 2011].

[16] GNU Radio Website. gnuradio.org [Accessed Jan 2012].

[17] GNU Radio Companion (GRC) Wiki Guide Version 0.7. Website. gnuradio.org/redmine/projects/gnuradio/wiki/GNURadioCompanion [Accessed Dec 2011].

[18] C. O. Hargrave, N. V. Shuley, M. Bialkowski, and J. C. Ralston, "Radar target identification of mining infrastructure for automated mine machinery navigation," *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, pp.802-806, 11-15 April 2011.