AN EMBEDDED PLATFORM BASED PUBLIC SAFETY COGNITIVE RADIO

Almohanad S. Fayez, Qinqin Chen, Alexander R. Young, Nicholas J. Kaminski, Hediah Alavi , Charles W. Bostian
Wireless@VT
Bradley Department of Electrical and Computer Engineering
Virginia Tech, Blacksburg, VA 24061
(afayez, chenq, alex.young, njkamin, hedieh, bostian}@vt.edu

ABSTRACT

The paper discusses the Virginia Tech (VT) Center for Wireless Telecommunication (CWT) Public Safety Cognitive Radio (PSCR), an all-band all-mode transceiver for public safety applications, and how it was redesigned to run on an embedded platform. The original PSCR was designed to allow first responders to carry it into an area, find existing public safety radios, and configure itself to communicate with those radios. In addition the PSCR is able to operate as a gateway bridging two incompatible radios. The original PSCR was designed to run on a laptop connected to a programmable radio front-end, but a laptop-based implementation makes the transition to a handheld device challenging in terms of power and computational requirements. The embedded PSCR addresses these concerns by using an embedded GPP/DSP processor based platform, a separate user-interface platform, and a programmable RF front-end for over the air communication. The GPP/DSP based processor realizes the radio functionality while the user interface platform is a cellular telephone touch screen and the programmable RF front-end provides the air interface. The paper discusses the selected hardware architecture, low level system implementation, and some of the issues that arise while transitioning a laptop-based system to an embedded platform.

1. INTRODUCTION

Modern public safety communications suffer from interoperability problems because radios used in different jurisdictions by police, firefighters, and emergency medical responders can be mutually incompatible [1]. The problems vary from radios operating on different frequencies to ones using different protocols. Such radio communication issues are especially severe when first responders from geographically distant jurisdictions and departments have to work together. Such scenarios occur during wide spread emergency situations, particularly if infrastructure like repeaters and base stations is inoperable and personnel would have to carry multiple radios to ensure continuous communication.

There are some SDR-based radios in the market today which implement baseband processing in embedded Digital Signal Processors (DSPs). For example the Thales Communication Liberty [2] multi-mode radio covering the following public safety bands (136-174 MHz, 380-520 MHz, 700 MHz, and 800 MHz) it operates in legacy FM and Project 25 (P25), trunked and conventional, modes. P25 is a standard for public safety radios; Phase 1 of P25 is based on Continuous 4 level FM (C4FM) modulation and an Improved Multi-Band Excitation (IMBE) vocoder developed by DVSI [3]. P25 radios also provide back compatibility with analog FM radios. The Harris Corporation RF-1033M is another multiband multimode radio covering 30-50 MHz, 136-174 MHz, and 380-512 MHz and implementing analog FM/AM and P25 conventional [4]. The Harris Unity XG-100 is an upgraded version of the RF-1033M extending the RF coverage to 700 and 800 MHz and also provides full P25 compliance [5]. These radios differ from our prototype in their ability to scan the public safety spectrum, identify waveforms, and configure themselves to interoperate with these waveforms.

2. ORIGINAL PUBLIC SAFETY COGNITIVE RADIO

2.1. Overview

The PSCR is designed to provide interoperability between public safety radios in the field, meaning that it requires minimal a-priori programming because it automatically scans for radios in the spectrum and configuring itself to communicate with those radios. It does this by automatically identifying their physical layer characteristics. The user selects a wanted network from the screen display, and the radio configures itself to interoperate with that network. The PSCR is also able to act as gateway bridging incompatible radios in the field. In Gateway mode, the PSCR is effectively a mobile repeater or audio bridge and is able to receive a signal at one frequency and retransmit it at a different frequency, in "bent pipe" fashion. However, the PSCR is also able to convert the signal from one waveform to another as the signal is retransmitted. It is this feature that really enables communications between otherwise incompatible radios in the field [6].

2.2 Architecture

The original PSCR [6] was based on a Linux based laptop running GNU Radio [7] connected to first generation USRP [8] over USB 2.0. The PSCR was later further improved by creating a PDA interface to the laptop [9], so that the laptop, server-side, would be mounted in vehicle with public safety personnel using a PDA, client-side, to communicate with the vehicular laptop over Wi-Fi as shown in Figure 1.

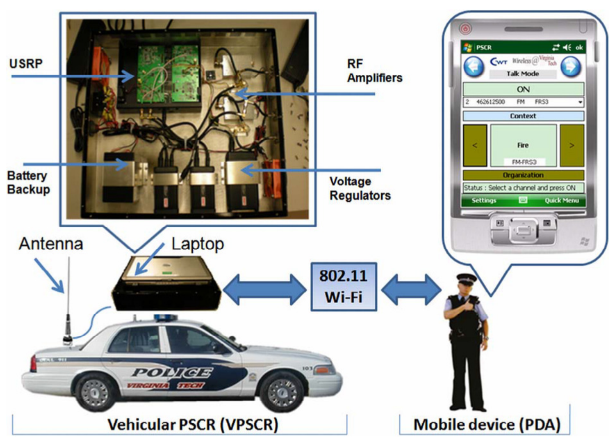

Figure 1. Vehicular PSCR. [9]

The internal architecture of the original PSCR is shown in Figure 2. It is based on the common cognitive radio (CR) design in which a software cognitive engine (CE) controls a software defined radio (or other frequency and mode agile radio platform), sensing the RF environment and the radio's own performance ("reading the meters") and adjusting the radio's physical and network layer parameters ("turning the knobs") to meet the user's requirements. This and other early VT cognitive radios used case-based reasoning and what were termed Wireless System Genetic Algorithms (WSGA) to optimize radio performance efficiently, using past experience to speed up the process. The original PSCR was modular in a way that allowed its components to run as different processes (on different computers if desired) communicating via sockets. While distributed operation and sophisticated optimization techniques are essential to some cognitive radio applications, we felt that they were not needed for public safety, and we de-emphasized or eliminated them in the embedded version of the PSCR, where the principal requirements are low power consumption and rapidly searching the RF environment and rapid radio configuration in response to the user's needs. This was also done because public safety personnel typically prefer their radios to operate in a deterministic well bounded manner.

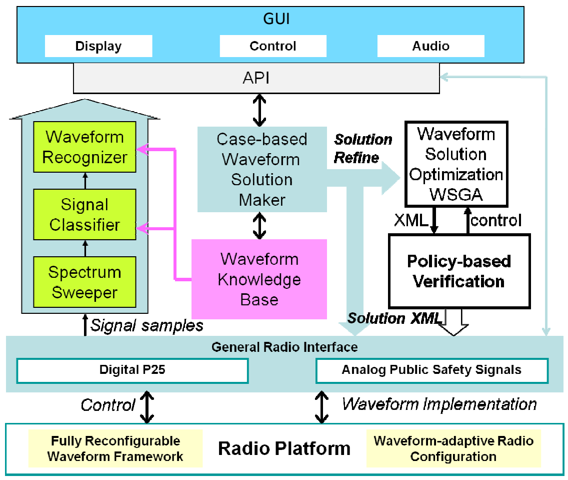

Figure 2. Original PSCR Architecture. [6]

3. EMBEDDED PUBLIC SAFETY COGNITIVE RADIO

3.1. Overview

The next natural milestone for the PSCR project is to port it to a small form factor platform, basically in a form that allows the PSCR to be manufactured as a hand-held radio if commercialized. Moving the system from a laptop to an embedded environment required some system modifications and other design elements which are discussed in this section.

3.2. Hardware Components

The embedded PSCR is made up of three main components, as shown also in Figure 3:

a- Programmable RF front-end:
The Ettus Research USRP E100 [10] was used. The USRP contains 2 64 Msps 12-bit ADCs dual 128 Msps 14-bit DACs. The Ettus WBX RF daughterboard [11] is used to provide coverage from 50-2200MHz for both Rx and Tx.

b- Computational Platform:

The E100 also includes a Gumstix Overo module [12] containing a TI OMAP3530 processor composed of an ARM Cortex A8 general purpose processor (GPP) and a TI C64x+ fixed-point DSP.

c- Graphical User Interface:
A Google Android (Nexus model) phone is used to provide a user-friendly touchscreen and audio interface to the radio. Please note the Nexus is not providing any phone functionalities; it's only an interface.

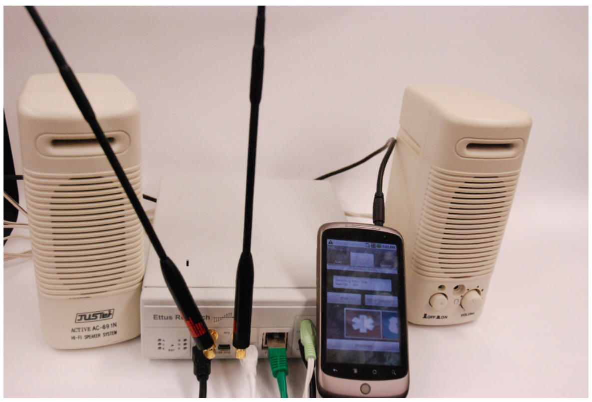

Figure 3. Embedded PSCR

3.3. System Architecture

Figure 4 shows the system architecture used for the embedded version of the PSCR. We revised the earlier architecture to the available resources. Motivated by our attempts to move the PSCR to an embedded platform, we undertook an extensive refactoring of the code base, maintaining functionality while improving performance via the C64x+ DSP as a coprocessor so the quality of audio conversations over the radio is qualitatively acceptable.

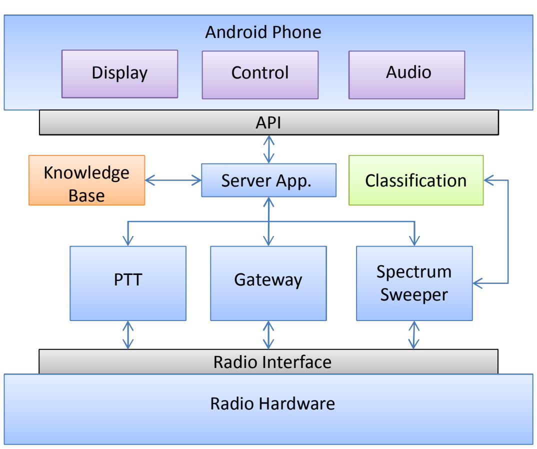

Figure 4. Embedded PSCR System Architecture.

The original PSCR architecture was written using a mix of C++ and Python for control code, using sockets for interprocess communication (IPC). The rewritten PSCR uses pure Python for control, and dispenses with the sockets. Taking inspiration from our work on the Cognitive System Enabling Radio Evolution (CSERE) [13], individual PSCR components are implemented as separate modules. Despite these separate modules, the PSCR operates in a monolithic fashion; all component modules are integrated into a single system at runtime.

The core of the PSCR is the control server, responsible for the radio's primary functionality. It receives commands via a USB-based socket from the PSCR GUI which runs on the Android phone. Upon receiving a command from the GUI, the server runs the appropriate module: Sensor, PTT, or Gateway. Figure 4 shows a system diagram of the server, modules, and GUI. Figure 5 displays some screen captures of the GUI itself.

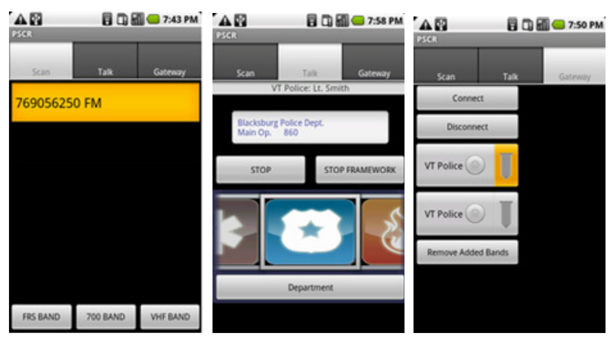

Figure 5. GUI Interface for Scan, PTT, and Gateway

3.4. Platform Support

An important aspect of the project is developing and generating the necessary environment to run and support the PSCR system. OpenEmbedded (OE) [14] is an embedded development framework which is used to compile and generate the Linux kernel, file system, and device drivers. OE provides all the necessary cross-compilation tools necessary to write applications for the USRP E100. Software tools from TI are used to compile code for the DSP and to support inter-processor communication between the GPP and DSP cores in the OMAP processor.

GNU Radio is used to realize the physical communication functionalities in the PSCR and is also used in the embedded version. However GNU Radio is not tuned to work on embedded platforms and does not provide inherent support for DSP processors. Therefore, we added new C64+ DSP components intended to improve radio performance on the E100 [15]. The PSCR is able to run without requiring any modification to account for the C64x+

DSP. This is done by utilizing a control and data link between the GPP and DSP to allow the GNU Radio C64x+ blocks to coordinate DSP configuration and data processing [15] as shown in Figure 6.

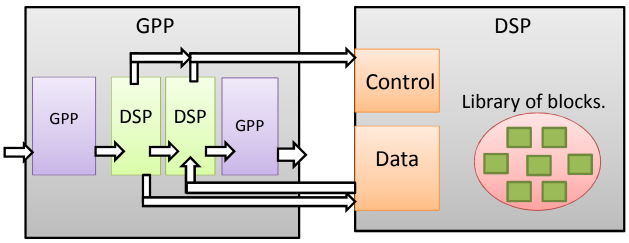

Figure 6. DSP-based GNU Radio Blocks Coordinating DSP Control and Data Processing.

3.5. Radio Flowgraph Implementation

Public safety radios typically include analog FM and digital P25 modes. Since access to the IMBE vocoder is difficult we had to forgo P25 support since the vocoder is a critical piece of the standard. The original PSCR provided FM, BPSK, and QPSK support; however, in the embedded PSCR only FM is currently supported, but the radio can also classify C4FM signals, and support for digital modulations may be added in the future. All the signal processing blocks available in GNU Radio would need to be re-written for the E100's OMAP processor in order to yield acceptable performance.

For the PTT functionality, the receiver chain implements the channel filter and demodulation on the DSP to accelerate receiver performance while de-emphasis is implemented in the GPP. The transmitter chain is implemented purely in the GPP because the GPP/DSP interface is slower in the DSP-to-GPP direction which makes it impossible to maintain a continuous transmission with acceptable audio quality given the TI GPP/DSP communication driver used [16]. However, in running the transmitter on the GPP it was necessary to forgo the channel filter since it was computationally intensive on the GPP core and prevented the transmitter from yielding acceptable sound quality. In future work, we would be looking at alternative drivers for transferring data between the GPP and DSP which can yield better performance measures.

The gateway functionality is implemented as an RF frequency translator since the current embedded PSCR implements one modulation scheme. It is also able to interoperate FM signals with different RF frequencies and bandwidths. In supporting more modulation schemes, the gateway would perform a complete demodulation of the signal to the audio level and then would perform complete modulation of the signal to the desired modulation scheme and RF frequency.

The sensor uses power spectrum density (PSD) energy detection to detect the presence of signals of interest--signals we are interested in communicating with. Radio spectrum is recorded and processed using well known FFT techniques. If the FFT results indicate that a signal is present, additional data are gathered, centering on the frequency of interest. These data are sent to the classifier for processing. The classifier uses a k-nearest neighbor (KNN) algorithm to determine the modulation type [6]. The classifier is C++ code developed for the original PSCR, and runs in a separate process. It determines the modulation of the signal of interest, and returns this information to the sensor for use by the central processes.

4. RELATED WORK

This work presented in this paper has provided he foundation for additional research in related areas, including SDR system profiling, and autonomous vehicle communications.

Some of the research includes the application of models of computation to SDR and CR design and studying how added computational awareness can impact the quality of radio configurations developed by CRs. Current research supported by the Air Force Research Lab in Rome, NY (AFRL) addresses the application of low-cost electronics for enhancing communications and increasing situational awareness in UAVs. We have developed a use case with a UAV flying a nominally cyclic or repeating flight path. As the UAV traverses the path, it experiences varying RF effects, including multipath propagation and terrain shadowing. The goal is to provide the capability for the UAV to learn the flight path with respect to motion and RF characteristics, and modify radio parameters and/or motion behavior proactively to mitigate deleterious effects [13].

5. CONCLUSION

This paper presents the CWT embedded PSCR system. It gives and overview of the original PSCR system and then discusses the new hardware architecture and modified software structure to accommodate its new embedded small form factor.

6. ACKNOWLEDGMENT

This project is supported by Award No. 2009-SQ-B9-K011 K011 awarded by the National Institute of Justice, Office of

Justice Programs, US Department of Justice. The opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the Department of Justice. This material is based on research sponsored by the Air Force Research Laboratory under Agreement #FA8750-09-2-0128. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research Laboratory or the U.S. Government.

This material is based on research sponsored by Air Force Research Laboratory under Agreement # FA8750-11-2-0014. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Research laboratory or the U.S. Government.

Almohanad Fayez was supported by the Saudi Arabian Cultural Mission (SACM).

Nicholas Kaminski was supported by a Bradley Fellowship through the Bradley Department of Electrical and Computer Engineering.

7. REFERENCES

[1] SDR Forum, "Software Defined Radio Technology for Public Safety," 2006.

[2] Thales Communication Inc., "Liberty™ Multiband Land Mobile Radio (LMR)," [Online]. Available: http://www.thalesliberty.com/about_liberty.asp. [Accessed 31 July 2012].

[3] DVSI, [Online]. Available: http://www.dvsinc.com/. [Accessed 31 July 2012].

[4] Harris Corporation, "RF1033M Multiband Multimode Land Mobile Radio (LMR)," [Online]. Available: http://harris.com/view_pressrelease.asp?act=lookup&pr_id=2353. [Accessed 31 July 2012].

[5] Harris Corporation, "Harris Unity™ XG-100 Multiband Radio," [Online]. Available: http://www.brevardcounty.us/docs/emergency-management/unity_xg-100_prod_guide.pdf?sfvrsn=2. [Accessed 31 July 2012].

[6] B. Le, F. Rodriguez, Q. Chen, B. Li , F. Ge, M. ElNainay , T. Rondeau , C. Bostian, "A Public Safety Cognitive Radio Node System," in *Software Defined Radio Technical Conference*, Denver, 2007.

[7] "GNU Radio," [Online]. Available: www.gnuradio.org. [Accessed 31 Juky 2012].

[8] Ettus Research, "Ettus USRP," [Online]. Available: https://www.ettus.com/product/details/USRP-PKG. [Accessed 11 August 2012].

[9] R. Rangnekar , F. Ge, A. Yougn, M. Silvius, A. Fayez, C. Bostian, "A Remote Control and Service Access Scheme for a Vehicular Public Safety Cognitive Radio," in *VTC*, Anchorage, 2009.

[10] Ettus Research, "USRP E100," [Online]. Available: https://www.ettus.com/product/details/UE100-KIT. [Accessed 31 July 2012].

[11] Ettus Research, "WBX Daughterboard," [Online]. Available: https://www.ettus.com/product/details/WBX. [Accessed 31 July 2012].

[12] "Gumstix," [Online]. Available: https://www.gumstix.com. [Accessed 31 July 2012].

[13] A. R. Young, N. J. Kaminski, A. Fayez and C. W. Bostian, "CSERE (Cognitive System Enabling Radio Evolution): A Modular and User-Friendly Cognitive Engine," in *DySpan 2012, in press*, Bellevue, Washington, 2012.

[14] OpenEmbedded, [Online]. Available: www.openembedded.org. [Accessed 31 July 2012].

[15] A. Fayez, N. Kaminski, A. Young and C. Bostian, "Embedded SDR System Design Case Study: An Implementation Perspective," in *IEEE CORAL*, San Francisco, 2012.

[16] A. Fayez, *Designing a Software Defined Radio to Run on a,* M.S. Thesis, Virginia Polytechnic Institute and State University, 2011.