A Software Radio-Empowered Sensor Network

Yafei Li, Zhitao He, Thiemo Voigt
Swedish Institute of Computer Science
{yafei, zhitao, thiemo}@sics.se

Sanna Leidelöf
Saab Systems

***Abstract*— Low power wireless sensors are limited by current radio technologies to short communication range and low throughput. We envision that future radios with advanced software programmable encoding and modulation will bring sensor networks unprecedented flexibility and performance. We have taken a step towards realizing this vision by designing a software-based, narrow-band transceiver using the GNU Radio software and the Universal Software Radio Peripheral hardware. We have verified the compatibility of our implementation with existing wireless sensor platforms. We demonstrate the flexibility of our design with sensing applications running on a sensor network communicating over hybrid radios.**

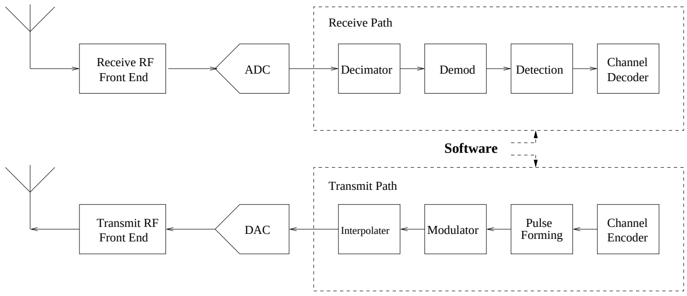

Fig. 1. A typical software radio architecture

I. Introduction

Designing conventional radio transceivers involves hard-wiring of application-specific signal processing blocks. Programmability of encoding and modulation is constrained by the hardware that is optimized for a particular radio standard. Software Defined Radio (SDR) overcomes the constraints imposed by standard-specific hardware. The idea of SDR is to move the hardware-software boundary as close to the antenna as possible [1]. The ideal software defined radio would have an antenna sampled by an ADC and the rest is done in software, turning radio hardware design problems into software problems. The fundamental characteristic of software radio is that software defines the transmitted waveforms, and software demodulates the received waveforms. Implementation of modulation and demodulation is done using software instead of dedicated circuits. Leveraging extra programmability at the physical layer, the system can handle different radio signals without changing hardware. Figure 1 shows a block diagram of a typical software radio architecture, whose digital signal processing is done completely in software.

Wireless sensor networks consist of small, battery-powered, sensor-equipped embedded devices that communicate wirelessly using an on-board low power radio. Sensor networks enable numerous surveillance, monitoring and other applications. Currently, the radios used in sensor network applications are limited in that they can only transmit and receive on one radio frequency at a time. For some radios, the frequency is fixed while other radios, e.g. IEEE 802.15.4-compliant radios, can select between 16 different frequencies in the 2.4 GHz band. SDR enables sensor network researchers to experiment with emerging radio technologies and to test algorithms for future radios in real environments rather than simulated ones.

One of our goals is to integrate SDR nodes into our testbed that uses the WISENET testbed software [2]. Currently, our sensor network testbed is homogeneous in that it consists of only 802.15.4-based Tmote Sky nodes [3]. Using SDR, we want to be able to include MSB nodes that feature a CC1020 radio [4] into the testbed. Without SDR or dual radio nodes, the gateway would be the only point of integration. In this paper, we discuss the design and implementation of a CC1020-compatible transceiver using open source SDR software and hardware.

II. Software and Hardware

A. GNU Radio

GNU Radio is a signal processing package [5] [6]. It provides an open-source library of common signal processing blocks and a software framework to combine blocks into a design. GNU Radio applications are primarily written using the Python programming language, which connects real-time signal processing blocks implemented as C++ routines to form a hierarchical design. The developer is thus empowered with a rapid development environment capable of implementing real-time, high-throughput radio systems.

GNU Radio has been widely used as a research tool for various applications [7], e.g., wireless testbeds [8] [9], distributed sensor networks [10], etc. It is also used by commercial-oriented applications, such as DVB-T modulator [11].

B. Universal Software Radio Peripheral

The most common hardware platform to run GNU Radio on is the Universal Software Radio Peripheral (USRP) [12] [13]. ETTUS' USRP allows general purpose computers to function

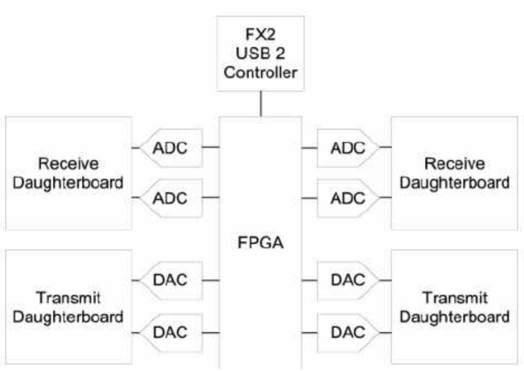

Fig. 2. Block diagram of USRP

as high bandwidth software radios. The USRP mother board contains four 12-bit 64M samples/sec ADCs, four 14-bit 128M samples/sec DACs, an FPGA for IF up/down conversion, and a programmable USB 2.0 controller to transfer control signals and baseband data sequences between the host and the hardware. The mother board can support up to two pairs of TX/RX radio front ends, in the form of daughter boards. There are multiple daughter board options for different frequency bands. We use the RFX2400 and RFX900 front ends in our designs. Figure 2 shows a simple block diagram of USRP.

C. CC1020

CC1020 is a single-chip, sub-GHz UHF transceiver designed for very low power and very low voltage wireless applications [14]. It is especially suited for narrow-band systems such as wireless sensor networks. CC1020 features a low-IF receiver. The received RF signal is amplified by the low-noise amplifier and down-converted in quadrature to the intermediate frequency. The I/Q signal is then complex filtered and amplified, and then digitized by the ADCs and demodulated. When the CC1020 operates in transmit mode, the synthesized RF frequency is fed directly to the power amplifier. The RF output is frequency shift keyed (FSK) by the digital bit stream.

III. Design and Implementation

We have designed a CC1020-compatible FSK transceiver, based on USRP and its RFX900 radio front end as the hardware platform. Within the GNU Radio framework, we design a receiving chain consisting of a data slicer to perform bit decision on the incoming waveform sequence followed by a channel decoder to perform FSK decoding. The transmission chain consists of an FSK encoder followed by an optional Gaussian filter. A proper set-up of the USRP hardware completes the implementation: we configure the IF frequency and the decimation factor in the digital down converter to match the required carrier frequency and channel bandwidth; we set the PGA gain in the digital up converter to a level compatible with CC1020.

Using USRP, we have also verified that a GNU Radio library for the 2.4 GHz IEEE 802.15.4 DSSS PHY [15] [16] communicates with our CC2420-based [17] Tmote Sky nodes. We choose to avoid the constraints of the IEEE 802.15.4 MAC frame format; instead we use a multi-layer stack with a bit-optimized frame format [18].

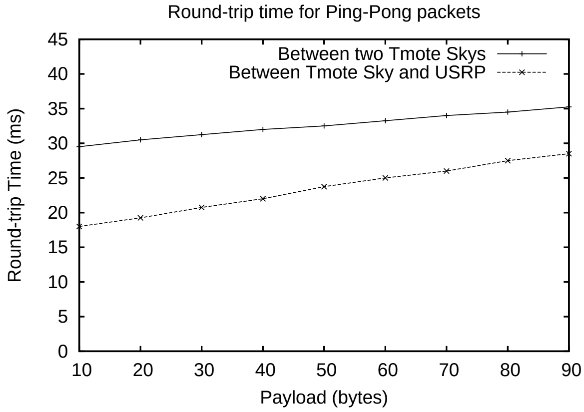

Fig. 3. Round-trip time between Tmote and USRP

IV. Evaluation

Our experiments have shown that USRP can inter-operate with sensor radios. We have measured round-trip packet delays between a Tmote Sky and a USRP, compared with round-trip delays between two Tmote Sky nodes. There is a 5-byte protocol overhead per packet: an 8-bit PHY header, a 16-bit logical channel ID, and a 16-bit CRC field. We vary the packet payload between 10 bytes to 90 bytes by steps of 10 bytes. Figure 3 shows the measured delays in milliseconds. The round-trip delay between Tmote Sky and USRP is shorter as a consequence of higher clock rate on the PC host, otherwise the SDR performance is in line with the sensor nodes.

V. Demo Description

We demonstrate our design using two hardware set-ups[1]. A USRP and two Scatterweb MSB430 sensor boards [4] form a 868 MHz network, which monitors the orientation of each sensor board by measuring their 3-D accelerometers. This can be used for item tracking in transportation applications. A graphical display shows the changes as the user flip the sensor boards in different directions. A second set-up comprises a USRP and two Tmote Sky nodes [3], forming a 2.4 GHz network. We show changes in the nodes' measured lighting and temperature. The demo is designed to highlight the flexibility of software radios by showing working examples of their potential application in sensor networks. Figure 4 shows a photograph of the USRP and the MSB hardware.

Acknowledgments

This work has been performed within the SICS Center for Networked Systems funded by VINNOVA, SSF, KKS, ABB, Ericsson, Saab Systems, TeliaSonera and T2Data. This work

[1]Note that we intend to use demo as presentation form at the event.

Fig. 4. USRP and MSB430

has been partially supported by CONET, the Cooperating Objects Network of Excellence, funded by the European Commission under FP7 with contract number FP7-2007-2-224053.

REFERENCES

[1] D. Saha, D. Grunwald, and D. Sicker, "Wireless innovation through software radios," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 62–67, 2009.

[2] O. Rensfelt, F. Hermans, C. Ferm, P. Gunningberg, and L.-A. Larzon, "An interactive testbed for heterogeneous wireless sensor networks," in *Demo Proceedings of the 4th International Conference on Distributed Computing in Sensor Systems (DCOSS 2008)*, 2008.

[3] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. IPSN/SPOTS'05*, Los Angeles, CA, USA, Apr. 2005.

[4] M. Baar, E. Köppe, A. Liers, and J. Schiller, "The scatterweb msb-430 platform for wireless sensor networks," in *SICS Contiki Hands-On Workshop 2007*, Mar. 2007.

[5] "GNU radio," Web page, visited 2009-03-19. [Online]. Available: http://gnuradio.org

[6] E. Blossom, "Exploring gnu radio." [Online]. Available: http://www.gnu.org/software/gnuradio/doc/exploring-gnuradio.html

[7] D. Valerio, "Open Source Software-Defined Radio: A survey on GNU-radio and its applications," 2008.

[8] S. Valentin, H. von Malm, and H. Karl, "Evaluating the GNU Software Radio platform for wireless testbeds," *University of Paderborn, Tech. Rep*, 2006.

[9] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. New York, NY, USA: ACM, 2008, pp. 401–412.

[10] T. Schmid, T. Dreier, and M. Srivastava, "Software radio implementation of short-range wireless standards for sensor networking," in *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*. ACM New York, NY, USA, 2006, pp. 119–120.

[11] V. Pellegrini, G. Bacci, and M. Luise, "Soft-DVB, a Fully Software, GNURadio Based ETSI DVB-T Modulator," in *5th Karlsruhe Workshop on Software Radios*, 2008.

[12] "Universal software radio peripheral," Web page, visited 2009-03-19. [Online]. Available: http://www.ettus.com

[13] F. A. Hamza, "The USRP under 1.5x magnifying lens," visited 2009-03-19. [Online]. Available: http://gnuradio.org/trac/wiki/UsrpFAQ

[14] Texas Instruments, "CC1020 datasheet," visited 2009-03-19. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/cc1020.html

[15] T. Schmid, "GNU radio 802.15.4 en- and decoding," *UCLA NESL Technical Report*, Sept. 2005.

[16] T. Schmid, L. Choong, and S. Leidelöf, "UCLA ZigBee PHY," visited 2009-03-20. [Online]. Available: https://www.cgran.org/wiki/UCLAZigBee

[17] Texas Instruments, "CC2420 datasheet," visited 2009-03-20. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/cc2420.html

[18] A. Dunkels, F. Österlind, and Z. He, "An adaptive communication architecture for wireless sensor networks," in *Proceedings of the Fifth ACM Conference on Networked Embedded Sensor Systems (SenSys 2007)*, Sydney, Australia, Nov. 2007.