SENDING IMAGES USING AN SDR PLATFORM

Alain C. Houle, Jonathan Daigneault, Jean-Philippe Déry, Jean-François Fortier, Jean-Sébastien Landry, Maxime Lepage, Alexis Ouellet-Patenaude, Alexandre Bisson, Hubert Gagnon-Lamonde (Université de Sherbrooke, Sherbrooke (Qc), Canada; {Alain.Houle, Jonathan.Daigneault, Jean-Philippe.Dery, Jean-Francois.Fortier, Jean-Sebastien.Landry2, Maxime.Lepage, Alexis.Ouellet-Patenaude, Alexandre.Bisson, Hubert.Gagnon-Lamonde}@usherbrooke.ca); Claude Bélisle (Communications Research Centre Canada, Ottawa (On), Canada; claude.belisle@crc.ca)

ABSTRACT

We report a mandatory end of degree project, based on SDR technology, undertaken by a team of eight undergraduate Electrical Engineering students at Université de Sherbrooke. Project goals include the development of a SDR unit capable of being reconfigured as commercial AM and FM receivers as well as an FRS transceiver. In addition, the SDR unit must be capable of emulating a digital waveform (FSK) over the FRS band for data transfer, specifically imagery. The prototype is based on a x86 computer for waveform processing, a Universal Software Radio Peripheral (USRP) on which two daughterboards are connected: a custom-made AM/FM RF front-end and a RFX400 daughterboard for the FRS band. Software architecture of the prototype is explained, concentrating on the management of multiple threads running simultaneously. While the project is scheduled to end by December 2007, it is expected that a live demonstration of the prototype will be made at the 2007 SDR Forum Technical Conference.

1. INTRODUCTION

Undergraduate students enrolled in the Electrical Engineering and Computer Engineering programs at Université de Sherbrooke must undertake a last year design project. Teams of six to eight students work on these projects at a pace of twenty hours per week per student, for a period of eight months (winter and fall semesters). With this workforce, design projects of considerable size and complexity can be undertaken.

In 2006, a team began the development of an SDR radio emulating an FRS transceiver and a commercial AM/FM receiver [1]. In January 2007, a new team of eight Electrical Engineering students took over the task to enhance the capabilities of this first SDR platform by adding improved RF front-end and developing a new communications protocol to support data transfer (file, images, and video).

In this paper, we will first describe the project organization and then give an overview of the hardware and software design of the prototype. Preliminary results and state of the project are reported before concluding on the academic advantages of the project for undergraduate engineering students.

2. PROJECT ORGANIZATION

The typical project steps of the mandatory last year design project at Université de Sherbrooke are:

- Detailed project planning and high-level technology assessment for the winter semester (January to April)
- Project execution, prototype tests and delivery for the fall semester (September to December).

Note that the project is suspended during May to August, allowing students to complete a work term in industry via the coop-program at Université de Sherbrooke.

The SDR unit developed by the 2006 team, on which this year's team was to bring improvements, had the following capabilities:

- FM reception in the commercial band (88–108 MHz) - off-line demodulation only (i.e. non real time)
- AM reception in the commercial band (0.5–1.5 MHz) - theoretical analysis and preliminary coding, not experimentally demonstrated
- Transmission and reception in the FRS band (462–467 MHz) - theoretical analysis and computer simulation

Building on what had been achieved in 2006, the 2007 team established the following objectives for their project:

- Complete commercial FM reception software so that real time continuous FM demodulation can be achieved.
- Design an improved RF front-end that will be adequate for both the FM and AM commercial bands.

- Implement and test the AM reception and FRS transmission and reception.
- Implement an FSK waveform and appropriate protocol to support data transfer in the FRS band.
- Redesign the graphical user interface to support the new capabilities.

With these objectives in mind, a complete and detailed project planning was completed in April 2007. Technical description of the prototype architecture is given in the next section.

3. PROTOTYPE ARCHITECTURE

The SDR platform is based on the Universal Software Radio Peripheral (USRP) from Ettus Research [2]. This platform was selected by the 2006 team for its open source development, the availability of the GNU Radio framework (open source also) [3] and its reasonable price to fit our academic budget.

The RFX400 daughterboard was added to provide transmission/reception capabilities in the FRS band (462 – 467 MHz). An RF front-end unit was developed by this year's team to allow reception on the AM and FM commercial bands (0.5 – 1.5 MHz and 88 – 108 MHz). A USRP/LFRX daughterboard was used as the interface between the USRP platform and the developed RF front-end. Figure 1 shows the general hardware architecture of the prototype. A detailed analysis of the prototype architecture follows.

3.1. AM/FM RF Front-end

Simplicity and modularity are criteria that guided the design of the AM/FM RF front-end. Input levels on the USRP are somewhat flexible and signal quality was deemed more important than design optimization (e.g. cost, number of components, etc.). The target signal to noise ratio (SNR) was set at 20 dB at the USRP's analog-to-digital converters input. The RF front-end must be sensitive enough to receive a commercial FM station with a modest antenna. Figure 2 illustrates the bloc diagram of the AM/FM RF front-end.

For the AM portion, the targeted gain is 50 dB so that a 1 mV input is amplified to 300 mV. The signal received from the antenna goes through a first 2nd order bandpass filter. It is then amplified and filtered again with an 8th order lowpass active filter. No frequency translation is made in the AM path.

The targeted gain for the FM portion is such that a 50μV input is amplified to 300 mV at the input of the USRP's analog-to-digital converters (75 dB gain). The signal received from the antenna first goes through a 2nd order bandpass filter to eliminate frequency components outside the commercial FM band. An amplification stage brings the FM signal to the proper level for analog mixing.

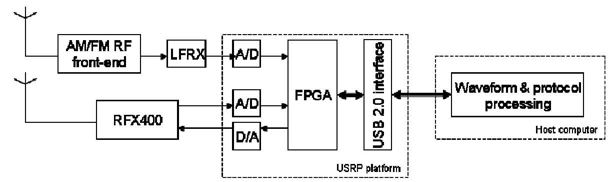

Figure 1. General hardware architecture

Using an 84 MHz oscillator, the mixing stage translates the commercial FM band from 88–108 MHz to an IF band of 4–24 MHz. The IF signal of the FM input goes through an 8th order active bandpass filter.

Both FM and AM sections are combined and go through a 32 MHz lowpass filter to prevent aliasing problems due to the USRP sampling rate of 64 Msamples/s. A final amplification stage allows the AM/FM composite signal to enter the USRP's analog-to-digital converter at a proper level. The two bands being located below 32 MHz, the whole composite signal can be digitized and processed in the digital domain.

For the FRS band, the RFX400 daughterboard allows translation of the 462–467 MHz band to baseband with I and Q components. The analog-to-digital converter used to digitize the I and Q components from the RFX400 daughterboard is not the same as the one used to digitize the commercial AM and FM bands so there is no frequency overlap problems between the FRS band brought to baseband and the composite AM/FM band.

3.2. Digital Signal Processing

The signal processing load for the digital domain is shared by the USRP's FPGA and the host computer. For reception of the AM, FM and FRS bands, the digital signal processing task of the USRP's FPGA follows three main steps:

- Translate the selected band to baseband while producing I and Q component samples.
- Decimate by a factor of 256 the resulting baseband I and Q components since 64 Msamples/s is not required once the signal is in baseband.
- Send the samples to the host computer via a USB 2.0 port for demodulation.

For transmission of the FRS band, the digital signal processing steps of the USRP's FPGA follow:

- Receive I and Q components of the FM modulated baseband signal from the host computer via a USB 2.0 port.
- Upsample I and Q components by a factor of 512 (digital-to-analog converters run at 128 Msamples/s) and send them to the RFX400

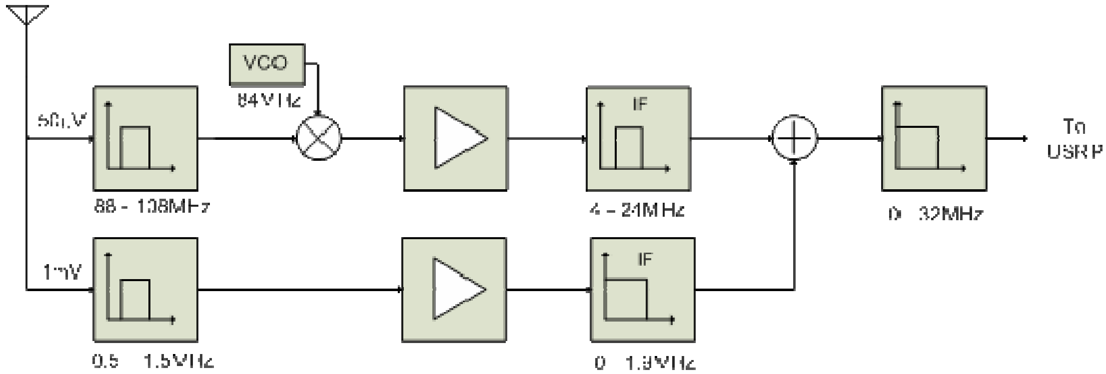

Figure 2. Bloc diagram of the AM/FM RF front-end

daughterboard via a digital-to-analog converter for final processing.

With respect to the host computer, the digital signal processing task mainly concerns the implementation of waveforms. Since the commercial bands are receive only, only demodulation needs to be implemented. For the FRS band, both FM demodulation and modulation are implemented. Moreover, the host computer implements an FSK waveform and communications protocol to support data transfer (file, images, and video) through the FRS band. The communications protocol is an asynchronous protocol. Figure 3 shows the expected BER performance obtained using computer simulation.

3.3. Host Computer Software Architecture

The operating system running on the host computer is a Linux operating system (Fedora Core 6 [4]) and all the programming is made in C++ and uses QT4 by Trolltech [5].

As mentioned in section 2, the first objective of the 2007 team was to complete the FM reception software so that continuous and real time FM demodulation can be processed. The software of the 2006 team could demodulate commercial FM signals but only off-line and by chunks of 8 seconds. The analysis of the first software version made it very clear that the missing part was a multiple threading architecture allowing multiple threads to run simultaneously. Real time management of these multiple threads is required for audio signals to be played back by the computer audio card while demodulating FM simultaneously. To do so, technical elements have to be defined:

- A structured software architecture is required to facilitate code maintenance and favor an eventual transition to an SCA compliant [6] software for use with, as example, the SCARI reference implementation made by the Communications Research Centre Canada [7].
- Data management to avoid conflicts in memory access between threads.
- Communication between threads for starting and stopping threads, share information with the graphical user interface, etc., and for guaranteeing flexible ordering of threads seen from the data path point of view for easy radio configuration. For instance, one might want to switch the data input from USRP to file, or change the output from sound card to FRS transmission or writing to a file, or even two or three at the same time.

The multiple threading framework of QT4 is totally adequate to implement these technical elements. The main thread class is the ThreadFrame class. This abstract class contains all the elements required by the threads, including those for data buffer management. Two data buffers may be used by each thread, one for data input to thread and for data output from thread processing. A same data buffer may be used by multiple threads (e.g. a data buffer may be the output buffer of thread A while being the input buffer of threads B and C). All types of threads require the creation of a class with ThreadFrame as a parent class and implementation of specific functions required by the considered thread.

Some functions allow each thread to manage any types of buffers. Data management functions provide general information like available memory spaces and number of consecutive memory spaces. Data management functions are divided in two groups: public and protected.

Public functions make it possible to assign data buffers to a given thread. It can then dynamically modify the buffer span, get the current buffer index, etc. While functions of the ThreadFrame class are “thread safe”, the public function are only used by the main thread (SDR Manager) to transfer data from one thread to another (e.g.to send data from the USB input thread to the demodulation thread and then from the demodulation thread to the audio playback thread). Protected functions are accessible only by a given thread.

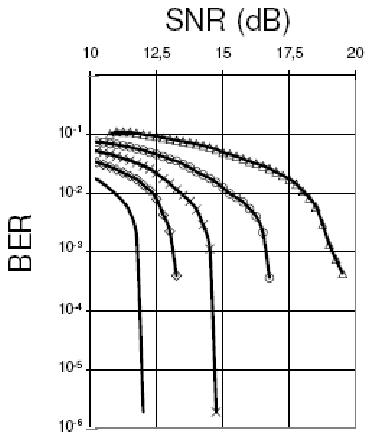

Figure 3. Expected BER for FSK data transfers for bandwidth going from 2.5 Hz/(bit/s) (leftmost curve) to 0.5 Hz/(bit/s) (rightmost curve) by steps of 0.5 Hz/(bit/s)

They are used to provide data to be processed by the thread for a particular implementation.

All data management and communication between threads always goes through the main thread. It is the only one that possesses an event management loop that is used by secondary threads to inform about processed data. The main thread instantiates objects of secondary threads and may call their functions. This establishes a bidirectional communication is established between the secondary threads and the main thread.

4. PERFORMANCE AND STATE OF THE PROJECT

At this point, the 2007 team has made the improvement to allow real time reception of FM signals. Generating an FM signal directly at the input of one of the analog-to-digital converter allowed to demonstrate the validity of the software architecture that efficiently handles multiple threading.

The 2007 team is also active on many aspects of the project simultaneously during their first month of the execution phase. The design of the AM/FM RF front-end is completed and the first front-end prototype is being built. The AM demodulation function and modular FM modulation to accommodate the FRS band are being implemented. The FSK waveform and the required communications protocol to support data transfer are first being developed under Matlab to be subsequently ported to the host computer. Finally, software drivers for the RFX400 daughterboard are being developed in C++ (as they are currently available only in Python) while a new graphical user interface is being designed with the QT4 library.

One goal set by the 2007 team is to present their prototype at the 2007 SDR Forum Technical Conference. To do so, the 2007 team must run many synchronized development activities simultaneously. The 2007 team certainly needs to demonstrate good project management skills, which is part of the academic goals of the activity.

5. CONCLUSION

The multidisciplinary nature of SDR technology makes it an ideal topic for final year design project. Students with various strengths must interact with each other in order to success in SDR. The structured nature of SDR technology also enables the students to apply project management techniques.

This is the second year where a team of undergraduate engineering students develop their final project based on SDR technology at Université de Sherbrooke. It is worth mentioning that at least two members of the 2006 team are now professional engineers active in the SDR technology sector. Hopefully, some members of the 2007 team will either follow the same path or continue at a graduate level by doing research on SDR technology.

The 2007 team needed to understand the achievements of the 2006 team before setting their own set of objectives. In addition to completing the AM/FM/FRS sections undertaken by the 2006 team and to designing a new AM/FM RF front-end, the 2007 team also planned to implement an FSK waveform and associated protocol to that will enable data transfer from one platform to another over the FRS band. The 2007 team also wants to present their prototype for a live demonstration of its prototype at the 2007 SDR Forum Technical Conference. Unfortunately, it is not possible to provide more details on the prototype performance at this time as the execution phase began in early September. The project is set to finish in December. The 2007 team is confident that a stable prototype will be demonstrated at the conference.

Looking forward, the Université de Sherbrooke intends to offer the opportunity to undertake a last year design project based on SDR technology in 2008. An obvious objective would be to continue with the developments completed by the 2007 team and to adapt the project to make an SCA-compliant prototype that would facilitate new developments and waveform implementations. Cognitive radio technology could also be explored.

6. REFERENCES

[1] A. Houle *et al.*, “Tackling a SDR Project as a Last Year Project in an Engineering Curriculum”, 2006 SDR Forum Technical Conference, Orlando, FL, USA, November 13-17, 2006.
[2] http://www.ettus.com
[3] http://gnuradio.org/trac
[4] http://fedoraproject.org
[5] http://trolltech.com
[6] http://sca.jpeojtrs.mil
[7] http://www.crc.ca/scari