A SOFTWARE DEFINED RADIO APPROACH FOR FUTURE ADAPTIVE MODE S TRANSPONDER

Jorel Ngounou, Omar Yeste and René Jr. Landry
Laboratory of Specialized Embedded System, Navigation and Avionics (LASSENA)
École de technologie supérieure, Montréal (Canada)
jngounou@elv.telecom-lille1.eu, omar.yeste@lassena.etsmtl.ca, Rene.Landry@etsmtl.ca
Phone: +1 (514) 396-8506 / Fax: +1 (514) 396-8684

ABSTRACT

For some years now, aerospace and aviation industries have been demanding for a new approach using a single generic transceiver system, which could be reprogrammable and universal, replacing multiple radios and antennas present in aircraft. The new system would allow simultaneous operation of multiple systems (Communication, Navigation, Surveillance – CNS) on a flexible development platform for future avionics applications. This flexible development platform is based on a Software Defined Radio (SDR) system.

This paper reports this new flexible SDR approach for development of a specific avionic system: a mode S transponder application. This task takes part of a more ambitious project currently in development at LASSENA laboratory with the aim of implementing avionics into these types of embedded systems. The project intends to provide avionics with the benefits of SDR technology, i.e. flexibility, reducing total equipment count and weight increased reliability and safety, and reduced time and cost for the design and development, among others.

This paper presents also the results of the mode S technology implementation in an SDR. Several functions of a typical mode S transponder are already implemented in our model, and worked as we might expect.

1. INTRODUCTION

Historically, the surveillance mode S transponder system was developed by MIT Lincoln Laboratory as a modernization of previous Air Traffic Control Radar Beacon System (ATCRBS). Mode S technology has already proved his benefits in comparison to ATCRBS (modes A and C). The most significant ones are the possibility of selective interrogation for aircraft, drastically reducing response garbling, and the increase of data types transmitted so that more information can be acquired from the aircraft. Mode S implements more than 16 million unique ICAO aircraft addresses (on 24 bits) and is capable of common-channel interoperation with previous ATCRBS modes (1030/1090 MHz for emission/ transmission), so that the transition to a full Mode-S system is seamless for airlines.

Industry refers to the RTCA DO-181 E standard [1], which explains the specifications of this technology. Presently, not all aircraft are equipped with mode S technology – transition to the whole globalization of the system is currently moving on – but in several countries and specific air spaces, mode S is mandatory, like in Germany since 2008.

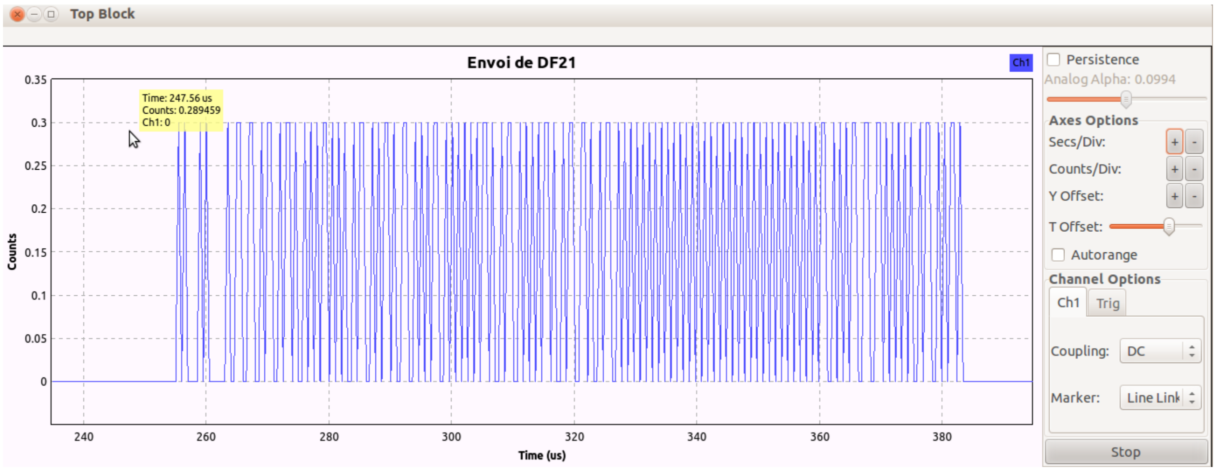
New designs of standards and avionics give several arguments in favor of the adoption of SDR. Thus, the LASSENA laboratory (Montreal, Canada) has started working in mode S protocol through this new SDR approach. One of the final purposes of the project is to develop new efficient and robust methods of digital signal processing for communication, navigation and surveillance applications and their integration into a single airborne SDR platform supporting standalone avionics systems currently in use, such as Distance Measurement Equipment (DME), Transponder Mode S, Automatic Dependent Surveillance – Broadcast (ADS-B) and wideband satellite communication radio.

2. MATERIALS AND METHODS

Transponder Mode S uplink signals (interrogations) use Differential Binary Phase-Shift Keying (DBPSK) modulation; while downlink signals use Pulse-Position Modulation (PPM) (see Figure 1). In our mode S transponder SDR prototype, both kinds of digital modulations will be implemented by software.

2.1. SDR PLATFORM

The SDR platform used in this work is the Universal Software Radio Peripherals (USRP), developed by Ettus Research now a subsidiary of National



Time: 247.56 us
Counts: 0.289459
Ch1: 0

□ Persistence
Analog Alpha: 0.0994
□ Autorange
Coupling: DC
Marker: Line Link

Figure 1: DF21 message, with the PPM signal form

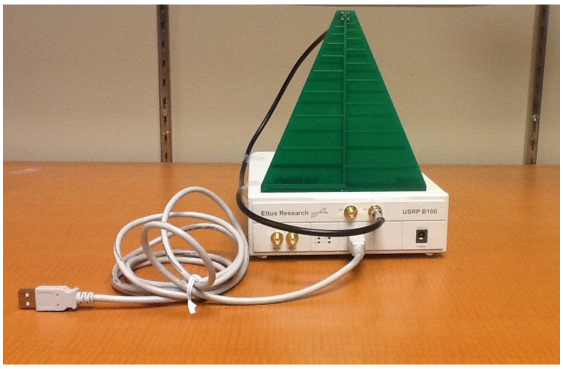


Figure 2: Photo of the SDR hardware (USRP B-100 equipment and antenna used for Mode S transponder prototype

Instruments, with either B-100 series (Figure 2) or N-210 series and WBX daughterboard that allows receiving and transmitting in a pretty wide bandwidth (50MHz – 2.2 GHz) [2].

For the software component, we have worked with the GNU Radio, a free and open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create SDRs. GNU Radio is widely used in academic and commercial environments to support both wireless communications research and real-world radio systems [3].

GNU Radio applications are primarily written using the Python programming language, while the supplied performance-critical signal processing path is implemented in C++ using processor floating-point extensions, where available. It permits the developer to implement real-time, high-throughput radio systems in a simple-to-use, rapid-application-development environment [3].

Thus, our USRP devices have been programmed by means of the GNU-Radio software. Data used for implementing functions of the mode S transponder transmitter section is based on a virtual bank of fields that contains all information needed to generate a reply. A C++ program sticks the concerned fields by some processing operations depending on the specific reply we want to generate. Finally, the messages are transmitted by the USRP following the GNU-Radio architecture shown in Figure 3.

2.2. Implementation of DF messages

DF (Downlink Formats) messages represent all responses of mode S transponders and UF (Uplink Formats) ones make reference to interrogations. To generate these DF messages, we firstly created a program written in C++, in order to concatenate concerned fields of the messages together. Afterwards, we dumped the final message in a binary file, ready to be read by the GNU Radio. The GNU Radio then converts the binary file into a float-valued baseband signal which is sent to the USRP. Finally, the USRP transmits mode S messages by modulating a 1090-MHz carrier with the output of the SDR, as it is shown in Figure 3.

2.3. Transmission/Reception

In order to test the prototype, in particular for the transmission, a mode S receiver [4] processes the replies and plays the role of Secondary Surveillance Radar (SSR) on the ground. If transmission is actually effective, the mode S receiver program shall decode the incoming messages and print its interpretation on terminal. The experimental setup for testing the transponder prototype is shown on Figure 4. Each antenna is connected to an USRP. In this configuration, the carrier frequency is set to 1090 MHz for transmission and reception, as the experiment concerns the downlink.

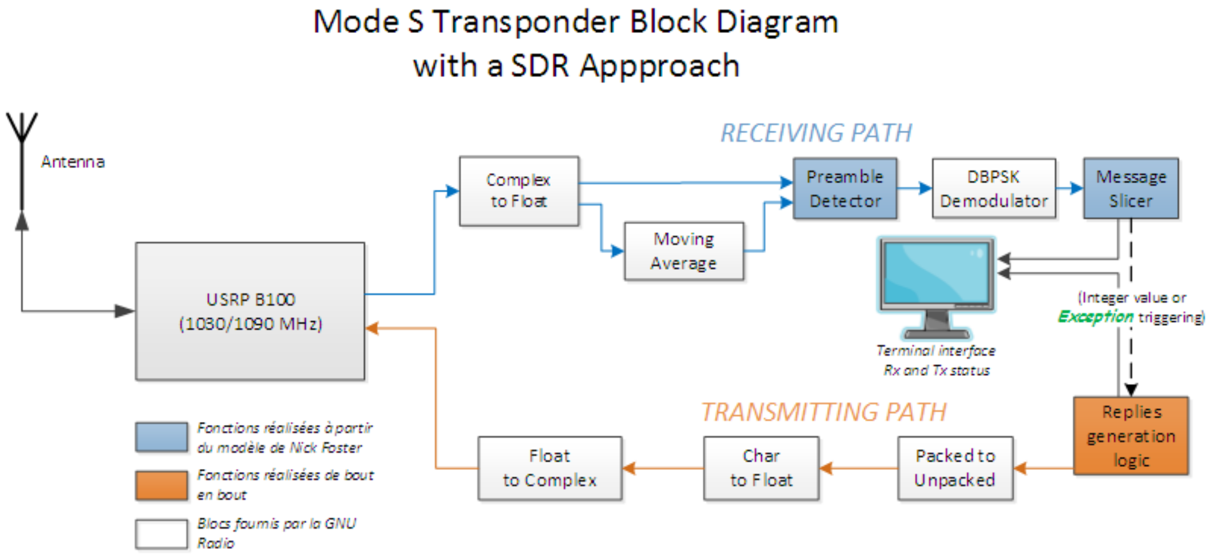

Mode S Transponder Block Diagram
with a SDR Appproach
Antenna
RECEIVING PATH
Complex to Float
Moving Average
Preamble Detector
DBPSK Demodulator
Message Slicer
USRP B100 (1030/1090 MHz)
Terminal interface Rx and Tx status
(Integer value or Exception triggering)
TRANSMITTING PATH
Replies generation logic
Packed to Unpacked
Char to Float
Float to Complex
Fonctions réalisées à partir du modèle de Nick Foster
Fonctions réalisées de bout en bout
Blocs fournis par la GNU Radio

Figure 3: Receiver and transmitter section of the mode S transponder based on GNU Radio blocks

3. RESULTS

3.1. SENDING A RESPONSE

The current prototype is able to transmit all replies corresponding to a Level 2 transponder, as it is defined in the RTCA standard “DO-181-E” [1]. First results show that all generated messages are well decoded by the receiver. However, we saw during tests that some messages could be randomly corrupted. In these cases, messages were also decoded by the receiver, but with one or more mistakes, regarding the original message. We firstly think that these perturbations could be due to a misconfiguration of MTL (Minimum Triggering Level) of the receiver (too low, ≏ -65dB), so it permits to detect signals with a low power level and so a greater chance to contain errors. We are continuing to look for other possible reasons for the occurrence of these errors.

3.2. DETECTING AN INTERROGATION

Critical issue of a transponder is to detect (reception part) an interrogation to specifically send the corresponding response (transmission part). Thus, we have two distinct parts in our model, as we can see in Figure 4. To manage transition from detection to transmission, we have created exceptions (one by UF type) that allow interrupting the detection routine, to swift our mode S transponder program to transmission mode. When these exceptions are raised by the code, they disable detection part of our program, and enable the transmission one (composed of some basic GNU Radio blocks) with the generated message we want to transmit as the source of the flow-graph.

4. DISCUSSION

Basic GNU Radio blocks, which are already supplied by software, are enough to transmit a mode S response (we don’t need to write any code to create the necessary blocks for sending replies). Therefore, as soon as we succeed in generating a message (effective bits containing the information to transmit) with the desired format (PPM pre-modulation), transmission part does not need programming skills. GNU Radio supplies blocks for creating a functional flow-graph.

At this stage, we have demonstrated that it is possible to implement mode S technology in a SDR.

After the implementation of the SDR mode S prototype on USRP B100, our next step is to implement our prototype on an embedded SDR, such as USRB E110. This type of SDR can run standalone operations for embedded applications, thanks to its Real-Time Operating System (RTOS), Embedded Linux.

RTOS are operating systems intended to serve real-

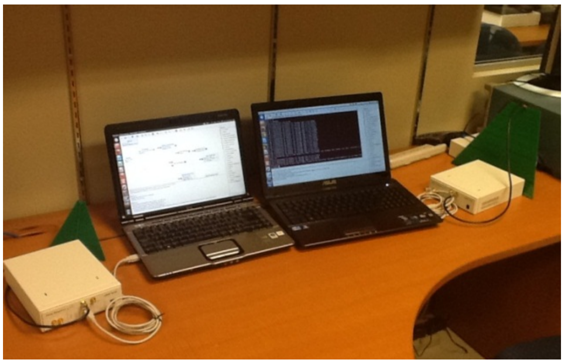


Figure 4: Configuration of Transmitter/Receiver for tests

time application requirements. They must be able to process data as it comes in, typically without buffering delays [5]. This is typically a critical issue to address in the final design of our mode S transponder prototype.

Future purpose is to implement a highly integrated embedded SDR avionics system performing different functions: Mode S, ADS-B, and DME, among others. Functional tests will be managed by the IFR 6000, a dedicated unit for testing mode A/C/S transponder, DME, TCAS, ADS-B and TIS avionic systems.

REFERENCES

[1] "Minimum Operational Performance Standards for Air Traffic Control Radar Beacon System/Mode Select (ATCRBS/Mode S) Airborne Equipment", RTCA, Inc

[2] https://www.ettus.com/product/details/WBX

[3] http://gnuradio.org/redmine/projects/gnuradio

[4] gr-air-modes free software, available on https://github.com/bistromath/gr-air-modes

[5] http://en.wikipedia.org/wiki/Real-time_operating_system

ABOUT THE AUTHORS

Jorel Ngounou is an international student from Telecom Lille 1, a French engineering school. He worked as intern in the Laboratory of Specialized Embedded System, Navigation and Avionics (LASSENA). His project was focused on the Integration of Mode S transponder functions in Software Defined Radio USRP B100.

Omar Yeste received the Telecommunications Engineer degree and the Ph.D. degree from Universidad Politécnica de Madrid, Spain, in 2002 and 2007, respectively. He joined the Signals, Systems, and Radiocommunications Department of Universidad Politécnica de Madrid in 2008 as an Associate Professor. In 2012, he joined the Electrical Engineering Department of École de Technologie Supérieure, Montreal, Canada. Since then, he holds a position as an Institutional Researcher at the Laboratory of Specialized Embedded System, Navigation and Avionics (LASSENA). His research activities cover the areas of GNSS, radionavigation, software defined radio, interference mitigation, digital signal processing, radar, electronic warfare, radiometry, cyclo-stationarity, time-frequency analysis, and time-variant filtering.

René Jr. Landry is a Professor at the Electrical Engineering Department of École de Technologie Supérieure (ÉTS), Montreal, Canada. He is the Director of the Laboratory of Specialized Embedded System, Navigation and Avionics (LASSENA) since 1999 and a member of LACIME and AeroÉTS. He got his Bachelor Engineering degree at the Ecole Polytechnique (Montreal) in 1992, a Master of Science in satellite communication engineering at the University of Surrey (Guildford, England), in 1993, a Master in space electronics and a DEA in microwaves at the ISAE/SupAero (Toulouse, France), in 1994. Professor Landry has obtained his PhD degree at Université Paul-Sabatier (Toulouse), in 1997 and a Post Doc in Space Science at the National French Space Industry (CNES), Toulouse, France, in 1998. His major research activities are related with GNSS and inertial navigation technologies, electronic counter-measures (anti-jamming, robustness), communication and real-time embedded systems. Prof. Landry has numerous patents and publications with several R&D projects with Canadian and International industries. He is also a professional pilot normal aeroplane category with multi-engines class, instrument and seaplane ratings.