Software Defined Radio (SDR) Architecture to Support Multi-Satellite Communications

Mamatha R. Maheshwarappa
Surrey Space Centre
University of Surrey, Guildford
Surrey, United Kingdom – GU27XH
+44 (0) 1483-686029
m.maheshwarappa@surrey.ac.uk

Mark Bowyer
Airbus Defence and Space Ltd
Anchorage Road, Portsmouth
Hampshire, United Kingdom-PO35PU
+44 (0) 2392-705733
mark.bowyer@astrium.eads.net

Christopher P. Bridges
Surrey Space Centre
University of Surrey, Guildford
Surrey, United Kingdom – GU27XH
+44 (0) 1483-689137
c.p.bridges@surrey.ac.uk

ABSTRACT—**Software Defined Radio (SDR) is a key area to realise new software implementations for adaptive and reconfigurable communication systems without changing any hardware device or feature. A review on efficient use of limited bandwidth and increasing distributed satellite missions can lead to the need for a generic yet configurable communication platform that can handle multiple signals from multiple satellites with various modulation techniques, data rates and frequency bands that must be compatible to typical small satellite requirements. SDR is beneficial for space applications as it can provide the flexibility and re-configurability and this is driven by fast development times, new found heritage, reduced cost, and low mass Commercial Off-The-Shelf (COTS) components. The implementation of a combined System-On-Chip (SoC) and SDR communication platform enables additional reduction in cost as well as mass. This paper proposes a SDR architecture in which Field Programmable Gate Array (FPGA) System-on-Chip (SoC) is paired with a Radio Frequency (RF) programmable transceiver SoC to solve back-end and front-end re-configurability challenges respectively. The test-bed is aimed at implementing the signal processing software functions in both the dual-core ARM processors and associated FPGA fabric. The distribution of the functions between the FPGA fabric and dual-processor is based on profiling experiments using signal processing blocks, implemented on the development platform, in order to identify where bottlenecks exist. This paper discusses further the results from the new multi-signal / multi-satellite pipeline architecture and the subsequent bandwidth, data rate and processing requirements. Aspects of implementing and testing signal processing chains needed for CubeSat Telecommand, Telemetry and Control (TT&C) are presented together with initial results. Thus the proposed technology not only contributes for a lightweight and portable ground station but also for an on-board satellite transceiver.**

TABLE OF CONTENTS

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

1. INTRODUCTION

The objectives in multiple small satellite missions such as QB50 [1], Autonomous Assembly of a Reconfigurable Space Telescope (AAReST) [2], Surrey training research Nano-satellite Demonstrator (STRaND-2) [3], CubeSat Proximity Operations Demonstration (CPOD) [4] and Edison Demonstration of Smallsat Networks (EDSN) [5] are very ambitious and are driven by new complexities [6, 7]. There are still challenges to reliably increase the communication window [8, 9], the amount of data back to Earth and supporting multiple signal scenarios as seen in Figure 1 at any given time. These can be in various combinations from multi-satellites to multi-groundstations or even inter-satellite links (ISLs).

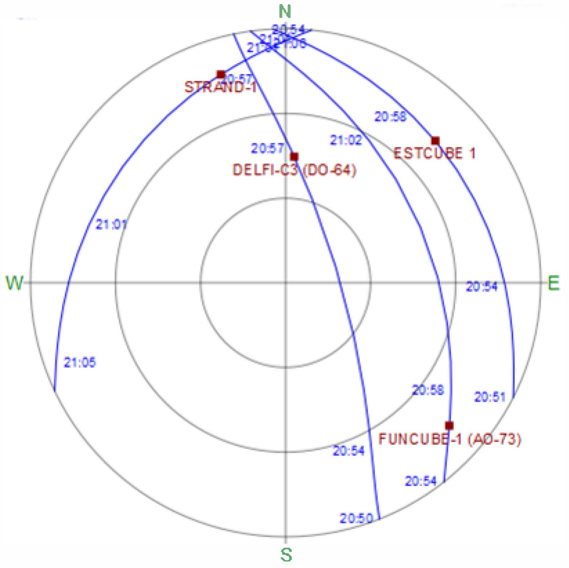


Figure 1. Radar View of the Antenna Showing Different Satellites in Visibility

For over two decades, Software Radio and Software Defined Radio (SDR) [10, 11] technology has promised to revolutionise the communication industry by delivering low-cost, flexible software solutions for communication protocols [12, 13]. In this decade, the introduction of FPGA SoC and, most recently, RF programmable transceiver SoC can fulfill the early promise. Federal Communications Commission (FCC) views Software Defined Radios (SDRs) as the result of an evolutionary process from purely hardware based equipment to fully software based equipment [14]. The paper issued by FCC Technical Advisory Council (TAC) citied that the evolution in SDRs has a potential role in the development of a board array of services and applications in telecommunications and information transfer technologies [15]. Advances in both hardware and software technology are making SDRs a reality [16, 17].

Attributes of space SDR architecture include:

1. Post-launch re-programmability to support control, configuration, re-configurability and new application installation.
2. Flexibility to support multiple signals from multiple satellites.
3. Scalability, extendibility, and modular design to support evolution over time.
4. Ability to allow latest application/waveform development to support new features and services without hardware upgrades.
5. Affordability to promote commercially available computer software and hardware products and standards.

Towards achieving the above attributes on an embedded system, this paper proposes a novel SDR architecture as seen in section 2. Various platforms to implement and evaluate the signals received from real-time satellite and the transmitter built in-house along with the results are discussed in section 3. The pipeline architecture to support multiple-signals from multiple-satellites through SDR and System-on-Chip (SoC) systems is proposed as part of the future work in section 5. Finally the conclusions are summarised in section 6.

2. Multi-Core SDR Architecture

The review of the increasing small satellite missions, the evolution of transceiver and hardware options along with generic problems involved in implementing SDRs in space are discussed in [18]. To better understand the functionalities of each block in the transceiver, the GNURadio open source software tool was chosen to test different combinations. Based on the initial results on performance of existing SDR software chains on Zedboard [19], it was evident that the flexibility of the original SDR concept [20] comes at the price of excessive demand for computational capacity, power and resources. Even compromised approaches, usually summarised under the term of SDR require more Million Instructions Per Second (MIPS) than a mono-processor [21]. Therefore, we focus on the multi-core signal processing system as seen in Figure 2.

This architecture consists of a Base-Band (BB) System-on-Chip (SoC) paired with Radio Frequency (RF) SoC. The BB SoC contains Field Programmable Gate Array (FPGA) fabric and ARM dual-core Cortex A9 processor. For initial development, the Avnet Zedboard containing the Xilinx Zynq 7020 FPGA SoC [22] is chosen, which provides a low-cost and well supported back-end for the signal processing functionalities. On the RF programmable transceiver SoC, initial evaluation took place using the Lime Micro Myriad RF containing the LMS6002D RF SoC [23]. More recent development has taken place using the Analog Devices' AD-FMCOMMS3-EBZ containing the newer AD9361 RF SoC [24]. The overview of the AD-FMCOMMS3 can be seen in Figure 3. It is hoped that future developments will incorporate the latest and most capable Lime Micro SoC, the LMS7002M [25]. The two boards seen in Figure 4 (and constituent SoCs) communicate using conventional parallel I/O for high speed sampled data (up to ~123 complex MSPS) and Serial Peripheral Interface (SPI) for configuration, control and monitoring.

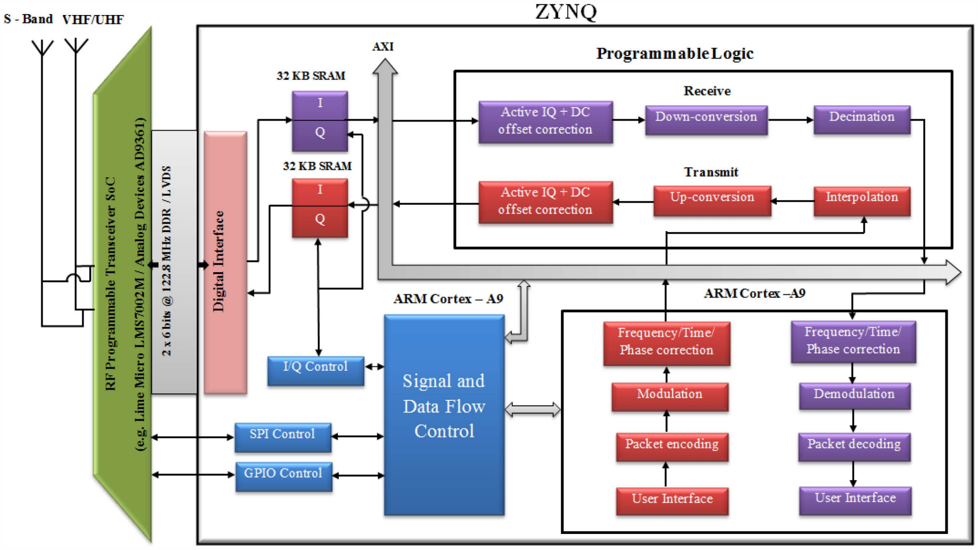


Figure 2. Multi-Core SDR Architecture

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

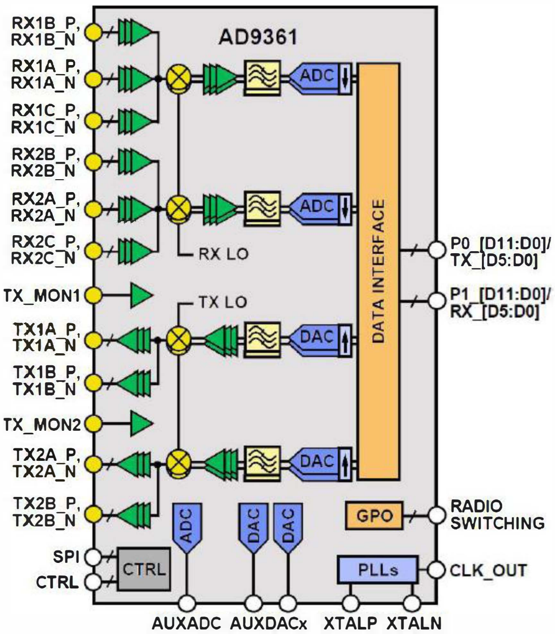


Figure 3. AD-FMCOMMS3 Overview [26]

The partly filtered and partly decimated samples received from the RF SoC are passed to the FPGA fabric (within the BB SoC) for further processing. The samples received over SPI are stored in internal SRAM for further processing. FPGA consists of the blocks that require higher computation capacity such as IQ correction, up/down conversion and decimation/ interpolation. Other blocks such as modulation and packet handling are done on one of the A9 processors. The other processor core performs signal and data flow control, on both transmit and receive paths. Both processor cores have tight connectivity to the FPGA fabric using the high speed SoC Advanced eXtensible Interface (AXI) bus.

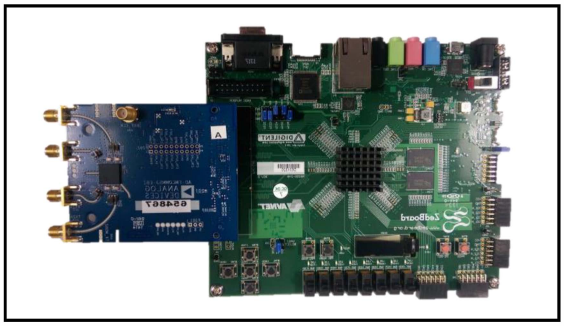


Figure 4. AD-FMCOMMS3 and Zedboard

As a first step towards validating the architecture, a simple receiver demodulator decoder for CubeSat beacon telemetry was implemented. The particular scheme, from AO-40 heritage [27], common among several CubeSats [28], is based on BPSK modulation and a robust concatenated code comprising Viterbi (Rate 1/2) [29] and two Reed Solomon (160,128) [30] blocks as seen in Figure 5.

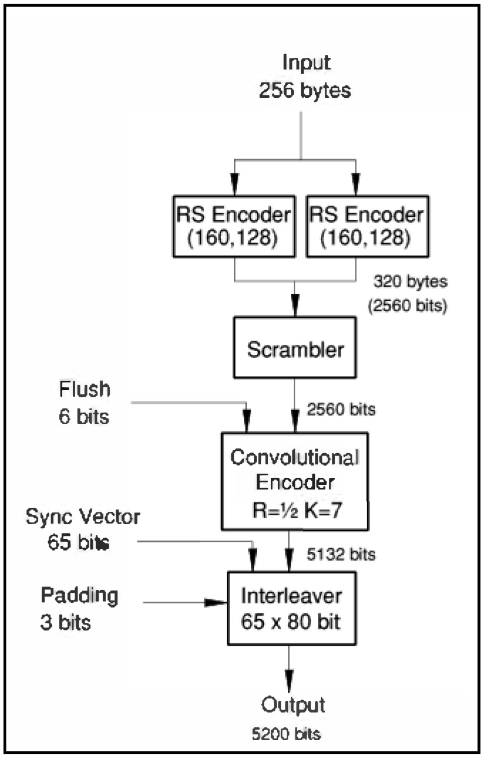


Figure 5. FEC Encoder Steps [31]

A practical problem encountered stems from the lowest filtered decimated sample rate, of order 1.5 Msps that can be output from AD9361 RF SoC. To address this, the AD9361 is configured to produce a multiple of an oversampled symbol rate (e.g. 40x1k2) that is conveniently larger than the 1.5 Msps limit imposed. In this implementation, 1.536 Msps was chosen that derives from 16 x 96 ksps. Now within the BB SoC, the received sample stream is decimated by 16 (initially in software, but to be moved to firmware). The resulting 96 ksps sample stream has sufficient bandwidth to allow residual LO breakthrough and IQ imbalance artifacts to be simply discarded, halving the available bandwidth to ~40 kHz, but the sufficient remaining bandwidth to address spacecraft Doppler and oscillator uncertainties. For greatest flexibility and simple access to floating point arithmetic, the 96 ksps sample stream is processed in software within the ARM Cortex A9. The first signal processing step is coarse carrier acquisition performed using an 8192 point Fast Fourier Transform (FFT). This results in a further 96 ksps sample stream that is approximately band centred on the largest (wanted) carrier. A software based Finite Impulse Response (FIR) filter, 27-taps long, containing a low-pass impulse response, is used to further filter and decimate the signal by factor of 10 to 9.6 ksps and offset by 1.2 kHz from baseband (for heritage reasons). At this stage the underlying signal is down-converted to baseband and matched filtered followed by carrier phase recovery. Finally, from symbol timing recovery a 1k2 symbol stream is produced and passed to the decoder.

3. Transceiver Implementation and Validation

This section focuses on various platforms used to test, debug and validate the tracking, reception, transmission and decoding of signals from FUNcube-1[32].

Setup 1- Real-time satellite signals received on a Dongle connected to a regular PC/laptop

The FUNcube-1 (AO-73) CubeSat provides a good starting point for our work because its telemetry beacon is documented and addressed by number of Open Source Software (OSS) demodulator decoder implementations. Much work here derives from Phil Karn's well-known AO-40 design and implementation [KA9Q] [33]. The chosen OSS starting point to form a "reference implementation" is Alex Csete's FUNcube Decoder (fcdec) available on github [34]. This code base, targeted for linux, is designed to work offline using sample files captured from the FUNcube Pro Dongle [35]. The default sample rate is 96 ksps. It was a reasonably straightforward task to make a modified version of this code to run soft real time on a modem x86 laptop. With some further effort, it was possible to create a soft real implementation based on the higher (and more representative) sample rates available from the RTL-2832 Dongle [36].

This evolved into a reference implementation called "rtl-fcdec". This was tested for interoperability against FUNcube-1 reference waveforms [37] up-sampled, stored and played back on a Rohde & Schwarz SMBV100 Vector Signal Generator (VSG) [38]. The block diagram of setup 1 is shown in Figure 6. Table 1 shows the ground station setup used to track FUNcube-1. The signal from the antenna was split into two; one connected to the dongle and the other to the spectrum analyser.

TABLE I
SSC GROUND STATION SETUP

Equipments	Specification
Antenna	144 and 430 Medium gain circular-polarised yagis
Low Noise Amplifier (LNA)	SP-7000 Preamp (+20dB)
Rotator	G-5500 Controller
Front-end	AD-FMCOMMS3-EBZ
Back-end	Zedboard

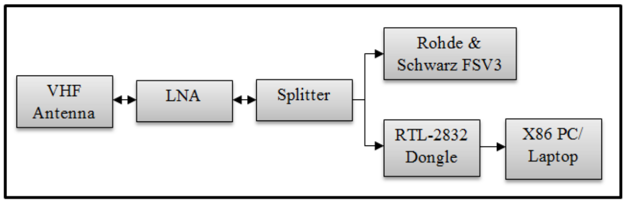
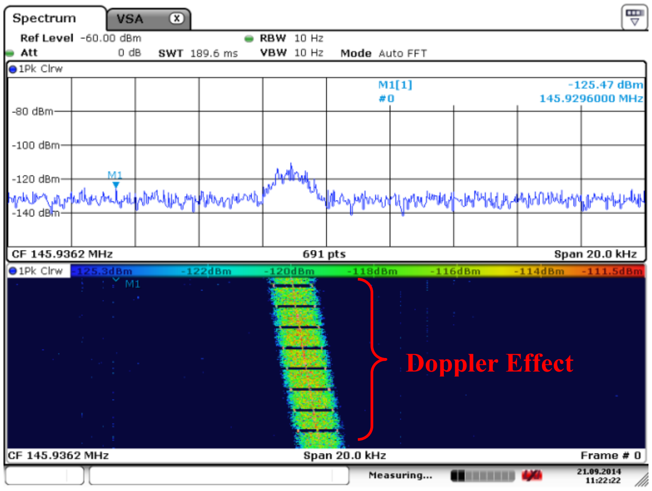


Figure 6. Setup 1 Block Diagram

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

Figure 7 is the signal received from FUNcube-1 centred at 145.935 where the Doppler Effect is evident and Figure 8 shows the constellation plot of the same signal. The signal thus received was decoded on a Personal Computer (PC).



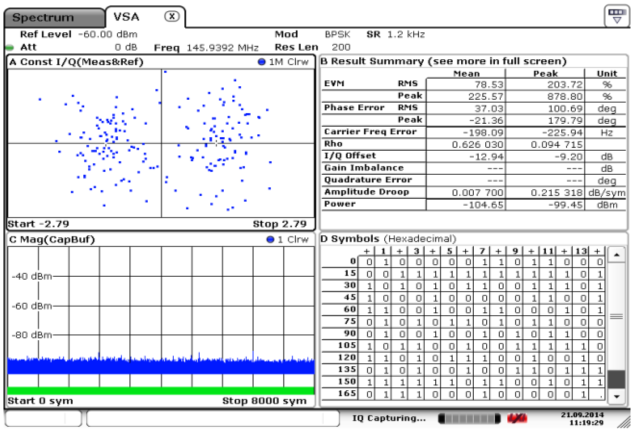
Ref Level -60.00 dBm RBW 10 Hz
Att 0 dB SWT 189.6 ms VBW 10 Hz Mode Auto FFT

CF 145.9362 MHz 691 pts Span 20.0 kHz

CF 145.9362 MHz Span 20.0 kHz Frame # 0

Date: 21.SEP.2014 11:22:22

Figure 7. Signal Received from FUNcube-1



Ref Level -60.00 dBm Mod BPSK SR 1.2 kHz
Att 0 dB Freq 145.9392 MHz Res Len 200

B Result Summary (see more in full screen)

		Mean	Peak	Unit
EVM	RMS	78.53	203.72	%
	Peak	225.57	878.80	%
Phase Error	RMS	37.03	100.69	deg
	Peak	-21.36	179.79	deg
Carrier Freq Error		-198.09	-225.94	Hz
Rho		0.626 030	0.094 715	
I/Q Offset		-12.94	-9.20	dB
Gain Imbalance		---	---	dB
Quadrature Error		---	---	deg
Amplitude Droop		0.007 700	0.215 318	dB/sym
Power		-104.65	-99.45	dBm

D Symbols (Hexadecimal)

	+	1	+	3	+	5	+	7	+	9	+	11	+	13	+
0	0	1	0	0	0	0	0	1	1	0	1	1	0	0	0
15	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1
30	1	0	1	0	1	0	1	1	0	0	1	1	1	0	1
45	1	0	0	0	1	0	0	0	0	0	1	1	0	0	1
60	1	1	0	0	1	0	0	1	1	0	0	0	1	0	1
75	0	1	0	1	0	1	1	0	1	0	1	1	0	0	0
90	0	1	0	0	1	0	0	1	0	1	0	1	1	0	0
105	1	0	1	1	0	0	1	0	0	1	1	1	1	1	0
120	1	1	0	1	0	1	1	1	0	0	1	0	0	1	0
135	0	1	0	1	0	0	1	0	0	1	1	0	1	1	1
150	1	1	1	1	1	1	0	1	1	0	0	1	0	1	1
165	0	1	1	1	0	0	0	1	0	0	0	0	0	1	.

Date: 21.SEP.2014 11:19:29

Figure 8. Constellation Plot of the Signal Received from FUNcube-1

The signals from FUNcube-1 were successfully decoded using this setup and thus the decoder program which is an open-source is marked to be working fine with PC/Laptop. The steps performed in the Decoder are as follows:

(1) Convert 96 KSPS 16-bit real sampled signals into frequency domain using 8192 FFT.

(2) Find the bin with maximum magnitude.

(3) Extract 204 frequency bins around selected the chosen bin (Centrebin).

(4) Convert back to time domain (inverse FFT).

(5) Down sample (and filter) from 96 KSPS to 9600 SPS at this point we have 1200 bps (DBPSK) which is 8 times oversampled.

(6) Complex down-conversion to remove residual 1200 Hz offset.

(7) Carrier phase recovery (just complex multiplication with last symbol when using DPSK)

(8) Correlating for 65 symbol unique word.

(9) 5200 bits of aligned output into FEC Decoder.

As a next step, the Dongle and PC were replaced by AD-FMCOMMS3 and Zedboard with the same decoder program but it failed to decode the signals. However, we could see the doppler shift in the frequency as seen in Figure 9. The equation relating the centrebin and frequency is given by Equation 1:

$$\frac{Centrebin}{8192} 96000 = Frequency \qquad (1)$$

Thus confirmed that we were tracking the right signal from FUNcube-1 on AD-FMCOMMS3. In order to investigate the problem of decoding in detail there was a need of a transmitter that could transmit at any desired time without the need to wait for a good elevation pass. The second setup includes one such transmitter which was built in-house adhering to the FUNcube-1 parameters.

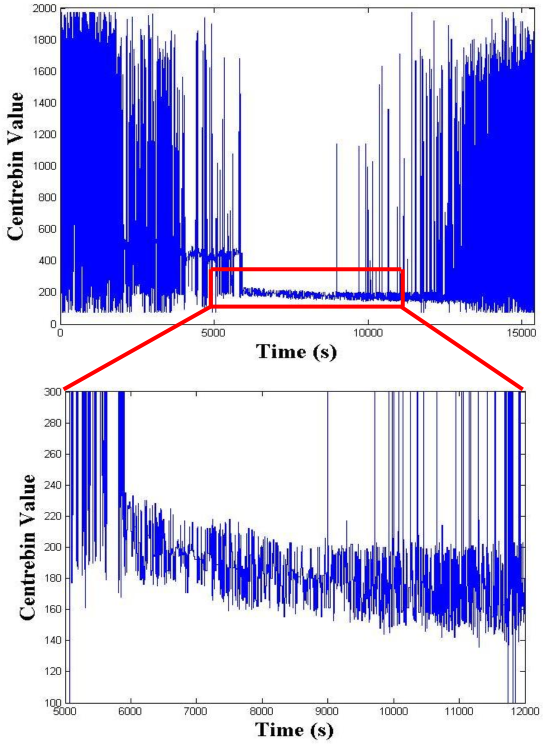


Figure 9. Doppler Curve of the Frequency Detected on the Board

Setup 2 – European Student Earth Orbiter (ESEO) transmitter signals received on dongle and regular PC

Figure 10 shows the block diagram of setup 2. The setup is similar to the previous except for the real satellite being replaced by ESEO transmitter [39], which is connected to the splitter. The rtl-fcdec reference implementation interoperates with a new UoS telemetry beacon design adhering to the FUNcube-1 parameters. The transmitter transmits for ~5 seconds and stays silent for ~3 seconds. Figure 11 shows the signal received on the spectrum analyser centred at 90.014 MHz. We are not seeing the Doppler Effect in the received signal as the signal is from a stationary source and not from a real satellite. And these signals were decoded successfully on a regular PC/Laptop as seen in Figure 12.

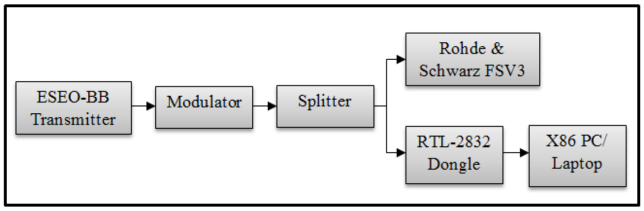
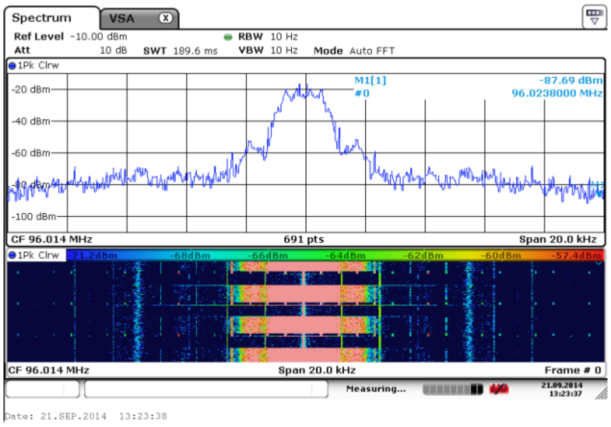


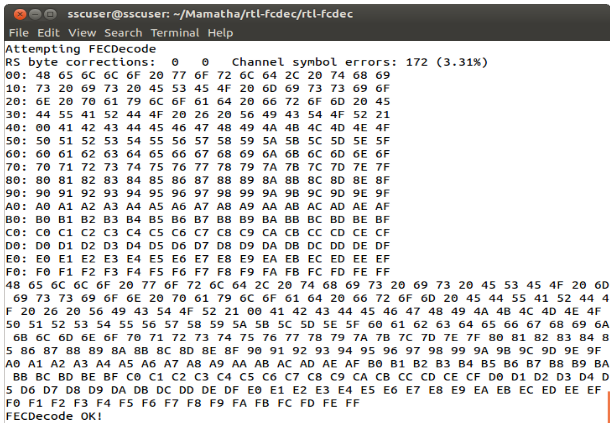
Figure 10. Setup 2 Block Diagram



Ref Level -10.00 dBm
Att 10 dB SWT 189.6 ms
RBW 10 Hz
VBW 10 Hz Mode Auto FFT
CF 96.014 MHz Span 20.0 kHz
CF 96.014 MHz Span 20.0 kHz

Date: 21.SEP.2014 13:23:38

Figure 11. Transmitted Signal from Setup 2


sscuser@sscuser: ~/Mamatha/rtl-fcdec/rtl-fcdec
File Edit View Search Terminal Help

```
Attempting FECDecode
RS byte corrections:  0   0   Channel symbol errors: 172 (3.31%)
00: 48 65 6C 6C 6F 20 77 6F 72 6C 64 2C 20 74 68 69
10: 73 20 69 73 20 45 53 45 4F 20 6D 69 73 73 69 6F
20: 6E 20 70 61 79 6C 6F 61 64 20 66 72 6F 6D 20 45
30: 44 55 41 52 44 4F 20 26 20 56 49 43 54 4F 52 21
40: 00 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50: 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
60: 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
70: 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
80: 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90: 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0: A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0: B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0: C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0: D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0: E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0: F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
48 65 6C 6C 6F 20 77 6F 72 6C 64 2C 20 74 68 69 73 20 69 73 20 45 53 45 4F 20 6D
 69 73 73 69 6F 6E 20 70 61 79 6C 6F 61 64 20 66 72 6F 6D 20 45 44 55 41 52 44 4
F 20 26 20 56 49 43 54 4F 52 21 00 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A
 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 8
5 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA
 BB BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0 D1 D2 D3 D4 D
5 D6 D7 D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
FECDecode OK!
```

Figure 12. Decoded Signals from ESEO Transmitter

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

Setup 3 – Transceiver on Embedded Systems (AD-FMCOMMS3 and Zedboard)

The Analog Devices AD-FMCOMMS3-EBZ has No-OS and Linux OS based device drivers accompanied by some application examples. For this work, we have proceeded with the Zynq Arm Linux OS based approach, assuming at least, the integration and test of application related OSS may be simplified. To this end, Analog Devices provide an extensive AD9361 Linux device driver, dependent on and accessed, using Linux industrial I/O (IIO) framework [40]. Linux IIO allows user space waveform applications to configure/query/sample-stream to and from the AD9361 using familiar UNIX calls (open/close/read/write/ioctl) and perhaps, and more preferred, by a user space library called libiio [41]. The Linux libiio provides a modern high performance abstraction to all IIO devices including the AD9361. Using IIO, it proved straight forward at first to produce an fcdec variant called iio-fcdec reusing much of the proven rtl-fcdec cade base.

Further, it has been possible to create a soft real time reference encoder called iio-fcenc. Successful interoperability testing of iio-fcenc and rtl-fcdec took place and Figure 13 shows the signal being received on a FUNcube Pro+ dongle. This signal was decoded using rtl-fcdec. Figure 14 shows the packets decoded on a Linux machine.

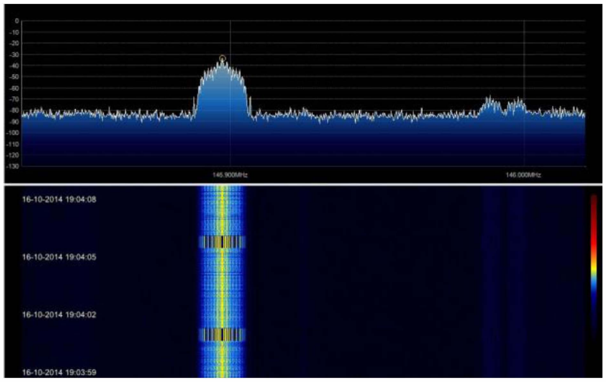
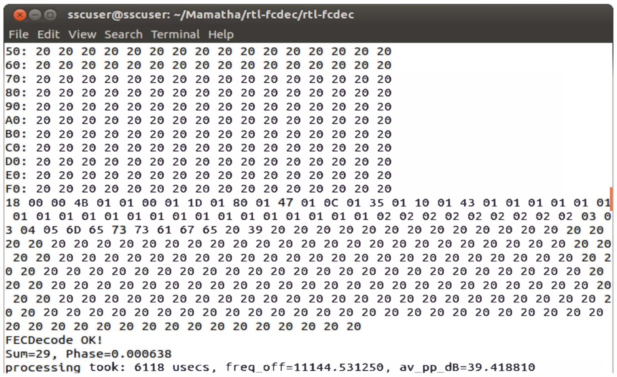


Figure 13. Signal Transmitted from Setup 3


sscuser@sscuser: ~/Mamatha/rtl-fcdec/rtl-fcdec
File Edit View Search Terminal Help

```
50: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
60: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
70: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
80: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
90: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
A0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
B0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
C0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
D0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
E0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
F0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
18 00 00 4B 01 01 00 01 1D 01 80 01 47 01 0C 01 35 01 10 01 43 01 01 01 01 01 01
 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 02 02 02 02 02 02 02 02 02 03 0
3 04 05 6D 65 73 73 61 67 65 20 39 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 2
0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 2
0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
FECDecode OK!
Sum=29, Phase=0.000638
processing took: 6118 usecs, freq_off=11144.531250, av_pp_dB=39.418810
```

Figure 14. Signal Decoded on a Regular PC

The AD-FMCOMMS3 provides the flexibility to transmit at any desired frequency within the range of 70 MHz to 6.0 GHz. This has an advantage over traditional transmitters which involved unique hardware for each frequency and thereby demonstrating the SDR attributes mentioned earlier. The limit here is the front-end bandwidth and the number of samples which is a function of frequency. The freedom to choose the frequency in software helps in compensating for thermal drift, clock timing and Doppler Effects.

The different transmitter platforms (Figure 15) available to test the receiver chain are; real satellite with Doppler frequency shift, constant transmitter (setup 2) at 96.014 MHz and the above transmitter that can be tuned to any desired frequency within the range. The receiver was tested with test setup 2 and the frequency offset was tracked with an average error of 789.75 Hz (calibrated with a reference signal generator). The samples were captured and an FFT was plot to check the spectrum as seen in Figure 16, the received spectrum resembled the spectrum transmitted at the right frequency offset (10kHz from DC) along with other interferences.

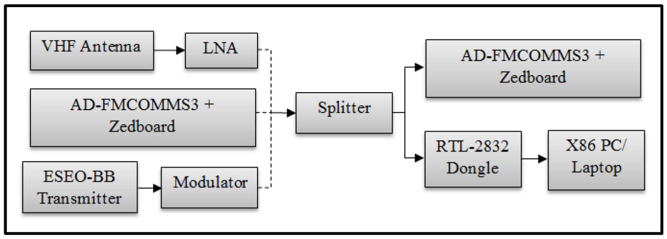


Figure 15. Setup 3 Block Diagram

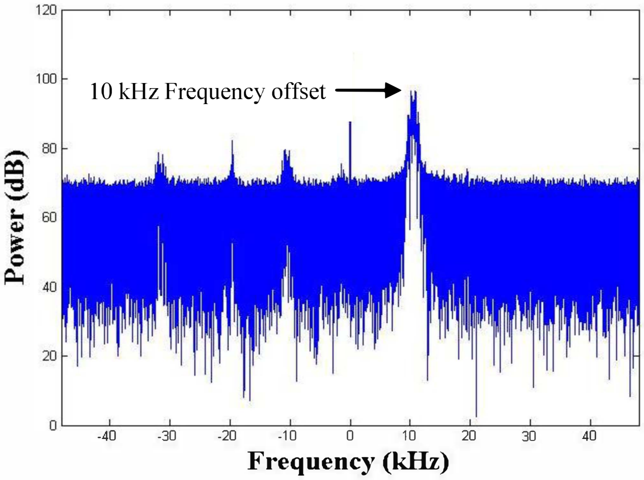
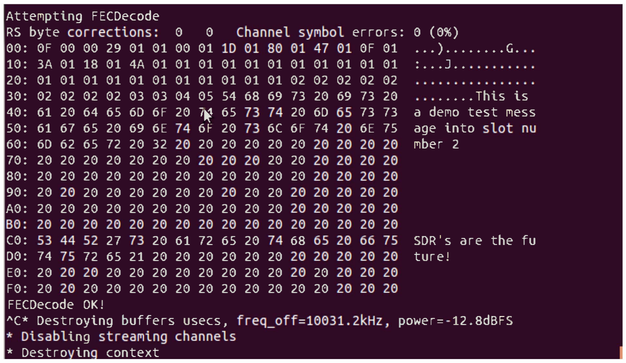


Figure 16. FFT Plot of the Samples Received

The transmitted signals from ESEO transmitter (Setup 2) and the signals that were looped back from Setup 3 were successfully received on the Zedboard and decoded as seen in Figure 17.

978-1-4799-5380-6/15/$31.00 ©2015 IEEE



```
Attempting FECDecode
RS byte corrections:  0   0   Channel symbol errors: 0 (0%)
00: 0F 00 00 29 01 01 00 01 1D 01 80 01 47 01 0F 01  ...)........G...
10: 3A 01 18 01 4A 01 01 01 01 01 01 01 01 01 01 01  :...J...........
20: 01 01 01 01 01 01 01 01 01 01 01 02 02 02 02 02  ................
30: 02 02 02 02 03 03 04 05 54 68 69 73 20 69 73 20  ........This is 
40: 61 20 64 65 6D 6F 20 74 65 73 74 20 6D 65 73 73  a demo test mess
50: 61 67 65 20 69 6E 74 6F 20 73 6C 6F 74 20 6E 75  age into slot nu
60: 6D 62 65 72 20 32 20 20 20 20 20 20 20 20 20 20  mber 2
70: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
80: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
90: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
A0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
B0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
C0: 53 44 52 27 73 20 61 72 65 20 74 68 65 20 66 75  SDR's are the fu
D0: 74 75 72 65 21 20 20 20 20 20 20 20 20 20 20 20  ture!
E0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
F0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
FECDecode OK!
^C* Destroying buffers usecs, freq_off=10031.2kHz, power=-12.8dBFS
* Disabling streaming channels
* Destroying context
```

Figure 17. Decoded signal

The technology not only aims at mobile ground stations but also targets the on-board satellite transceivers. The main constraint in flying this technology on a small satellite is the power consumed by FPGA, this research aims at providing the solution through parallelisation techniques which is part of the future work. Also, this can be extended to other modulation techniques and data rates in order to support multiple-signals from multiple-satellite scenarios. The proposed pipeline architecture is discussed in the next section.

4. Future Pipeline Architecture

In the previous section, the development and implementation of transceiver functionalities have been discussed. The advantage of this implementation is that the RF parameters such as modulation techniques, filters and data rates can be defined and configured by software. Thus, making expansion of its functionality and inclusion of other techniques is relatively challenging. With this architecture as a preliminary design, more signals including different parameters are proposed in the pipeline architecture as seen in Figure 18.

In this architecture, multiple down/up conversion blocks are implemented to provide a parallel system to receive multiple signals at the same time. A Fast Fourier Transform (FFT) will be performed on the incoming samples in order to find the desired signal. The minimum sampling rate that can be achieved on AD9361 is 2Msps, therefore the bin width of 2048 point FFT will be ~1kHz. A carrier tracking algorithm will be implemented, based on spectral estimation of the signal in order to account for different signals and frequency drift due to Doppler effects, temperature changes and component aging. Signal power estimation will be performed by summing the magnitudes of each FFT bin as in [42]. Also, an open-source satellite tracking and orbital prediction program, predict, will be integrated to acquire location and direction information of the satellites.

The algorithm proposed below aims at allowing access to multiple-reception after the signal is converted to complex baseband:

1. Extract the Acquisition of Signal (AOS) information of various satellites communication between the frequency range of 70 MHz to 6.0 GHz from predict at any given time.

2. Receive the complex signal from AD9361.

3. Perform FFT on the signal received to find the FFT peak/peaks.

4. If the difference between the FFT peak and the noise level is greater than or equal to a defined threshold (20dB is chosen in our case)

   a. Perform signal-power estimation.

   b. Record the data (signal power, magnitude of each

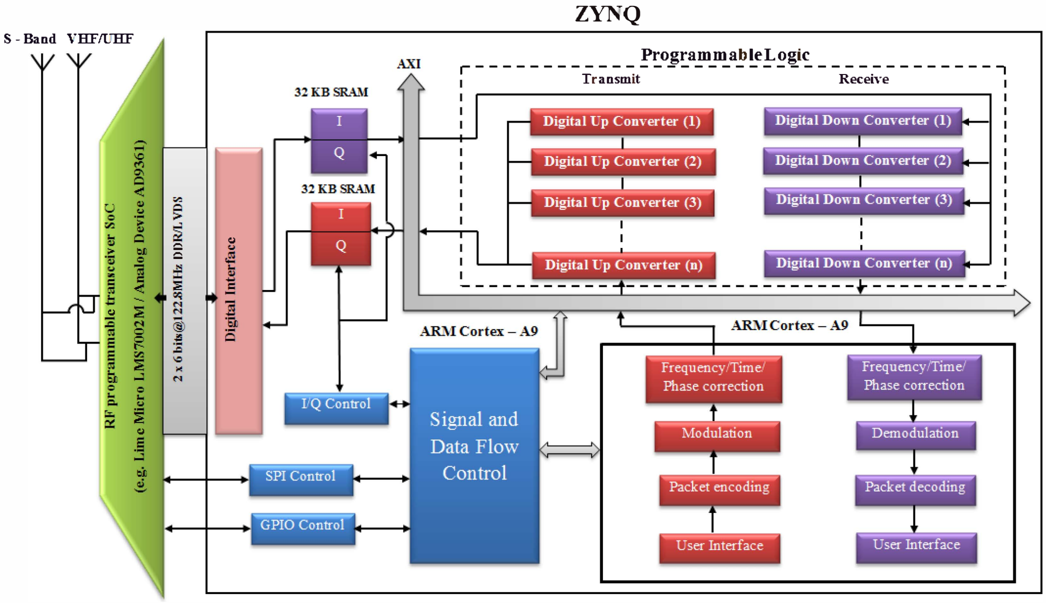


Figure 18. Pipeline Architecture

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

FFT bin used in the power estimation, and the local time)

c. Store it in the memory allotted based on the data rate and the bandwidth.

d. One of the processors will perform low-speed signal processing tasks such as modulation/demodulation and packet handling.

The above algorithm is described for one of the signals in the spectrum. Since the high-speed signal processing blocks are handled by FPGA, the processor can accommodate more than one encoder/decoder at a time (the number of signals will be decided once the performance test is performed). While the above process is carried out the next set of signals will be detected by performing FFT and stored in the memory (FIFO) for further processing.

5. Summary

SDR is a technology in fast evolution and which is receiving enormous acknowledgments and interests in the world of the space industry. SDR technology enhances the implementation of modules of a radio system such as modulation and demodulation, filters and protocols. This helps to develop a reconfigurable radio system, where the parameters are selected in a dynamic mode providing the freedom to change the parameters on ground as well as on-board satellite even after the launch.

In this paper, the design of an adaptive SDR architecture has been presented. Implementation on various platforms to validate the technology provides with the idea of different ways to test the architecture and also the resources required to perform certain task. It was clear that the desktop application can be implemented on an embedded system which would not only aid in upgrading the traditional ground stations but also can be implemented on a small satellite.

Further to this, with the goal of enabling embedded systems for multi-mode communication, the technology is proposed and is under development. The novelty here is to combine the state-of-the-art SDR hardware and open source software tool towards achieving a new generic communication platform for space applications.

Acknowledgement

The authors would like to thank Dr. Brian Yeomans, Research Fellow, Surrey Space Centre for his valuable time, discussions and contributions towards the paper.

References

[1] H. Bedon, C. Negron, J. Llantoy, C. M. Nieto, and C. O. Asma, "Preliminary internetworking simulation of the QB50 cubesat constellation," in Communications (LATINCOM), 2010 IEEE Latin-American Conference on, 2010, pp. 1-6.

[2] C. Underwood, S. Pellegrino, V. Lappas, C. Bridges, B. Taylor, S. Chhaniyara, T. Theodorou, P. Shaw, M. Arya, and J. Breckinridge, "Autonomous Assembly of a Reconfiguarble Space Telescope (AAReST)–A CubeSat/Microsatellite Based Technology Demonstrator," 2013.

[3] C. P. Bridges, B. Taylor, N. Horri, C. I. Underwood, S. Kenyon, J. Barrera-Ars, L. Pryce, and R. Bird, "STRaND-2: Visual inspection, proximity operations &amp; nanosatellite docking," in Aerospace Conference, 2013 IEEE, 2013, pp. 1-8.

[4] J. Westphal, C. Roscoe, S. Lutz, and T. Bennett, "Guidance, Navigation, and Control Algorithms for CubeSat Formation Flying," Rendezvous, vol. 100, p. 50x100x50m, 2014.

[5] H. B. Smith, S. H. K. Hu, and J. J. Cockrell, "NASAs EDSN Aims to Overcome the Operational Challenges of CubeSat Constellations and Demonstrate an Economical Swarm of 8 CubeSats Useful for Space Science Investigations," 2013.

[6] R. Sandau, K. Brieß, and M. D'Errico, "Small satellites for global coverage: Potential and limits," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 65, pp. 492-504, 2010.

[7] D. J. Barnhart, T. Vladimirova, and M. N. Sweeting, "Very-Small-Satellite Design for Distributed Space Missions," Journal of Spacecraft and Rockets, vol. 44, pp. 1294-1306, 2007/11/01 2007.

[8] T. Gergely and A. Clegg, "CubeSat Issues: Where Are We?," presented at the CubeSat High Speed Downlink Communications meeting, held during the 9th Annual CubeSat Developers Workshop, San Luis Obispo, California. , April 2012.

[9] P. Muri, J. McNair, J. Antoon, A. Gordon-Ross, K. Cason, and N. Fitz-Coy, "Topology design and performance analysis for networked earth observing small satellites," in MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011, 2011, pp. 1940-1945.

[10] J. Mitola, "The software radio architecture," Communications Magazine, IEEE, vol. 33, pp. 26-38, 1995.

978-1-4799-5380-6/15/$31.00 ©2015 IEEE

[11] J. Mitola, III, "Software radios: Survey, critical evaluation and future directions," Aerospace and Electronic Systems Magazine, IEEE, vol. 8, pp. 25-36, 1993.

[12] W. Horne, P. Weed, and D. Schaefer, "Adaptive spectrum radio: A feasibility platform on the path to dynamic spectrum access."

[13] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on, 2005, pp. 328-337.

[14] B. Tarver, E. Christensen, and A. Miller, "Software defined radios (SDR) platform and application programming interfaces (API)," in Military Communications Conference, 2001. MILCOM 2001. Communications for Network-Centric Operations: Creating the Information Force. IEEE, 2001, pp. 153-157 vol.1.

[15] F. C. Commission, "In the matter of authorization and use of software defined radio," vol. ET Docket No. 00-47, September 2001.

[16] M. Bosco, P. Tortora, and D. Cinarelli, "Alma Mater Ground Station transceiver: A software defined radio for satellite communications," in Metrology for Aerospace (MetroAeroSpace), 2014 IEEE, 2014, pp. 549-554.

[17] Z. Tong, M. S. Arifianto, and C. F. Liau, "Wireless transmission using universal software radio peripheral," in Space Science and Communication, 2009. IconSpace 2009. International Conference on, 2009, pp. 19-23.

[18] M. R. Maheshwarappa and C. P. Bridges, "Software defined radios for small satellites," in Adaptive Hardware and Systems (AHS), 2014 NASA/ESA Conference on, 2014, pp. 172-179.

[19] FPGA Accelerators in GNURadio with Xilinx's Zynq System on Chip Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/Zynq [Last accessed: October 2014]

[20] T. Ulversoy, "Software Defined Radio: Challenges and Opportunities," Communications Surveys & Tutorials, IEEE, vol. 12, pp. 531-550, 2010.

[21] H. Xiaojing, Y. J. Guo, and J. A. Zhang, "Sample Rate Conversion Using B-Spline Interpolation for OFDM Based Software Defined Radios," Communications, IEEE Transactions on, vol. 60, pp. 2113-2122, 2012.

[22] Zedboard product overview. Available: http://www.zedboard.org/ [Last accessed: October 2014]

[23] Lime Microsystems - LMS6002D. Available: http://www.limemicro.com/download/LMS6002Dr2-DataSheet-1.2r0.pdf [Last accessed: October 2014]

[24] AD-FMCOMMS3-EBZ: AD9361 Software Development Kit. Available: http://www.analog.com/en/evaluation/eval-ad-fmcomms3-ebz/eb.html [Last accessed: October 2014]

[25] LMS7002M Product Brief. Available: http://www.limemicro.com/download/Lime%20LMS7002M%20Product%20Brief.pdf [Last accessed: October 2014]

[26] Analog Devices Software Defined Radio Available: http://www.motiv8forums.com/uploads/3/2/9/8/3298228/analog_rf_june_2014.pdf [Last accessed: October 2014]

[27] AO-40 Telemetry Format. Available: http://www.ka9q.net/papers/ao40tlm.html[Last accessed: October 2014]

[28] C. Bridges, S. Kenyon, C. Underwood, and V. Lappas, "STRaND-1: The world's first smartphone nanosatellite," in Space Technology (ICST), 2011 2nd International Conference on, 2011, pp. 1-3.

[29] S. W. Shaker, S. H. Elramly, and K. A. Shehata, "FPGA implementation of a configurable viterbi decoder for software radio receiver," in AUTOTESTCON, 2009 IEEE, 2009, pp. 140-144.

[30] V. Ponnampalam and B. Vucetic, "Sub-optimum decoding of Reed-Solomon codes," in Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th, 1999, pp. 1625-1628 vol.3.

[31] Oscar-40 FEC Telemetry. Available: http://www.amsat.org/amsat/articles/g3ruh/125.html [Last accessed: October 2014]

[32] FUNcube : UK Amateur Radio Educational Satellite. Available: http://funcube.org.uk/ [Last accessed: October 2014]

[33] P. Karn. FEC-Coded AO-40 Telemetry Link. Available: http://www.ka9q.net/ao40/2002paper/ [Last accessed: October 2014]

[34] A. Csete. FUNcube Decoder. Available: https://github.com/csete/fcdec [Last accessed: October 2014]

[35] FunCube Dongle Pro. Available: http://www.funcubedongle.com/?page_id=1057 [Last accessed: October 2014]

[36] RTL-2382 Dongle Available: http://sdr.osmocom.org/trac/wiki/rtl-sdr [Last accessed: October 2014]

[37] FUNcube-1 Dashboard. Available: http://funcubetest2.files.wordpress.com/2013/11/funcube-dashboard-install-ops-guidance-notes-release-1-2.pdf [Last accessed: October 2014]

[38] R&S SMBV100A Vector Signal Generator. Available: http://www.testequipmenthq.com/datasheets/Rohde-Schwarz-SMBV100-Datasheet.pdf [Last accessed: October 2014]

[39] European Student earth Orbiter (ESEO). Available: http://www.esa.int/Education/European_Student_Earth_Orbiter [Last accessed: October 2014]

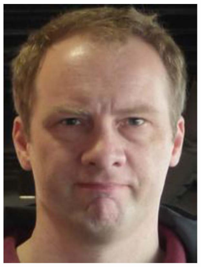
[40] AD-FMCOMMS3-EBZ User Guide. Available: http://wiki.analog.com/resources/eval/user-guides/ad-fmcomms3-ebz [Last accessed: October 2014]

[41] Libiio Github. Available: https://github.com/analogdevicesinc/libiio [Last accessed: October 2014]

[42] F. Machado, E. Villar, P. Marino, F. P. Fontan, G. Blarzino, G. Carrie, L. Castanet, and J. Lemorton, "Beacon Receiver Developments at U. Portsmouth-U. Vigo, and ONERA," Second SatNex Workshop, April 2008, Oberpfaffenhofen, Germany.

BIOGRAPHY

Mamatha R. Maheshwarappa *received her BEng in Electronics and Communication Engineering from Nitte Meenakshi Institute of Technology, Bangalore in 2009. She was the system engineer for STUDSAT-1- India's first pico-satellite, designed and developed by undergraduate students. Later she received MSc in Space Technology and Planetary Exploration from the University of Surrey in 2012 and currently a PhD student at Surrey Space Centre, University of Surrey. Her research is in Software Defined Radios (SDRs) for small satellites.*



Dr Mark Bowyer *(BEng, 1987; PhD, 1993) is Senior Expert in Secure Communications at Airbus defence and Space working in the Communications, Intelligence and Security business unit. He has over 20 years work experience in Satellite Communications (SATCOMS) Systems. He has led activities in SATCOM Software Defined Radio R&D since its inception in 1997. This has included development of three generations of hardware platform and software applications. Today, he consults on all aspects of secure SATCOM including RF, signal processing and waveform design for ground, air, and space-based modem systems. Mark holds a BEng (1st) from Birmingham University and PhD from the University of Kent at Canterbury.*

Dr Christopher Bridges *received his BEng in Electronic Engineering from the University of Greenwich in 2005 and his PhD in Agent Computing for Distributed Satellite Systems in 2009 from Surrey Space Centre, University of Surrey. He is currently leading the On-Board Data Handling (OBDH) Group. He researches software defined radios, real-time embedded systems, agent computing, Java processing, multi-core processing in FPGAs, and astrodynamics software and hardware computing methods towards real spaceflight payloads together with colleagues in SSC & SSTL.*

978-1-4799-5380-6/15/$31.00 ©2015 IEEE