IMPLEMENTATION OF SMART ANTENNA AND TRANSCEIVER API ON OSSIE PLATFORM FOR WIRELESS INNOVATION FORUM STANDARDS

Chiyoung Ahn(HY-SDR Research Center, Hanyang Univ., Seoul, Korea: ahncy@dsplab.hanyang.ac.kr); June Kim (HY-SDR Research Center, Hanyang Univ., Seoul, Korea: nzneer@dsplab.hanyang.ac.kr); and Seungwon Choi*(corresponding author, HY-SDR Research Center, Hanyang Univ., Seoul, Korea: choi@ieee.org)

ABSTRACT

This paper presents an implementation of SA (Smart Antenna) API (Application Programming Interface) and XCVR (Transceiver) API using OSSIE (Open-Source SCA Implementation-Embedded) [7]. Through our implementation, it is verified that the SA API can be utilized in any of SCA (Software Communication Architecture)-based SDR (Software Defined Radio) systems. Also, we verify that the XCVR API can be realized with a commercial RF (Radio Frequency) solution such as USRP2 (Universal Software Radio Peripheral). SA API enables various functions of the array antenna system such as beamforming, MIMO (Multiple Input Multiple Output) of spatial multiplexing, which are core technologies in 4G mobile communication system. In order to support the array antenna structure, XCVR API had to be first extended into multichannel, of which details are shown in our implementation. As OSSIE, on which our implementation is based, provides the SCA framework as well as the tools for developing SCA waveform, the implemented SA API will be applied for a standard design of WINNF.

1. INTRODUCTION

The SA WG (Working Group) of the WINNF has been developing an API for SDR SA systems. Also XCVR WG of the WINNF has been developing a XCVR API specification for the interface between modem and antenna.
Both groups are currently making efforts to standardize SA API and XCVR API. In this paper, to verify the interoperability between SA API and XCVR API, we implemented both APIs.
The implementation of both APIs is based on OSSIE because JPEO (Joint Program Executive Office) adopts an SCA, a de facto standard for SDR systems. OSSIE provides a development environment of SCA waveform which offers the CORBA (Common Object Request Broker Architecture) middleware. Therefore, we believe the implementation and verification of both APIs using OSSIE is a necessary process for the standardization.
Also, the RF transceiver of implemented system is realized with a commercial RF solution, namely, USRP2. The USRP2 is a low-cost hardware device designed for rapid prototyping and research in SDR applications [5]. In our implementation, we developed the waveforms supporting SA API and transceiver API using OSSIE and USRP2.

2. SA API AND XCVR API

2.1. SA API

SA is an array of multiple antennas that are used in conjunction with a signal processing subsystem within a wireless base station, wireless gateway or mobile terminal device to significantly improve wireless system performance.

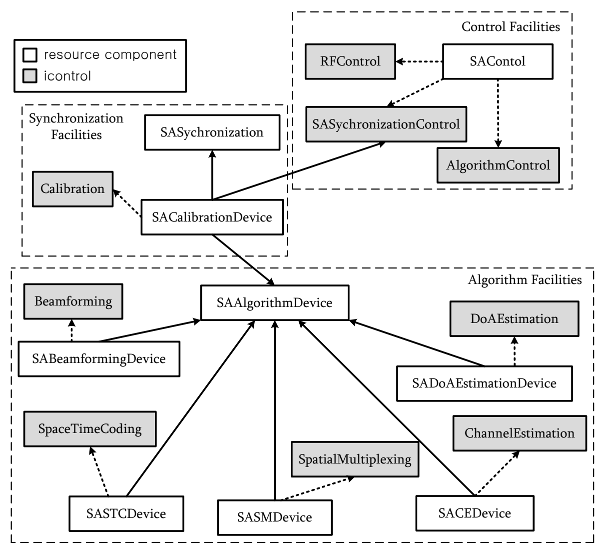

Figure 1 PIM of SA API

Figure 1 illustrates the PIM (Platform Independent Model) of the SA API that has been developed by the WINNF's SAWG. The PIM as shown in Figure 1 consists of three facilities: SAControl facilities, SASynchronization facilities, and SAAlgorithm facilities.
The SAControl facilities, as shown in Figure 1, include the base SAControl component along with the control interfaces for the RF/IF component and the other two groups of facilities.

The synchronization facilities of the SA API are for calibration as well as synchronization of signals. The SASynchronization component is realized by inheriting interfaces from calibration and latency.
The algorithm facilities are used to execute all the algorithms for beamforming, DoA (Direction of Arrival) estimation, channel estimation, SM (Spatial Multiplexing), and STC (Space-Time Coding).
Once the PIM is completed, all the 3 functionalities defined in the model can then be mapped into a proper model for a specific platform. That specific model is referred to as the PSM (Platform Specific Model).

2.2 XCVR API

The term “transceiver” is used here to encapsulate the entire set of hardware and software components within a radio set necessary to convert a low-power RF signal into digital baseband for the receive function, and reciprocally to convert digital baseband signals to low-power RF for the transmit function.

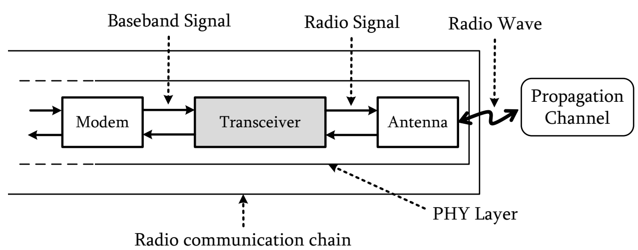

Figure 2 XCVR subsystem

Figure 2 illustrates a block diagram of our implementation of the XCVR subsystem within the PHY (physical) layer of a radio chain. The XCVR subsystem is a part of the radio chain that transposes, for transmission, baseband signal into radio signal, and, for reception, radio signal into baseband signal. Inside the radio chain, the XCVR subsystem is comprised between the modem and the antenna subsystem. It exchanges the baseband signal with modem, and radio signal with the antenna subsystem. Modem, XCVR and antenna are generally considered as parts of the PHY layer.

3. IMPLEMENTATION

3.1 Implementation of SA subsystem

Figure 3(a) is a block diagram of SA subsystem for the SM MIMO waveform. It consists of four blocks: MIMOTxDemo, MIMOChannel, MIMORxDemo, and SASpatialMulti-Plexing.
In the MIMOTxDemo block, random bits are modulated into QPSK symbols, and then the complex-valued symbols are transferred to the MIMOChannel block. Also, pilot signal, provided by external properties, is transmitted to the MIMOChannel block. The MIMOChannel block generates a receive signal from the signal of MIMOTxDemo block. In the MIMORxDemo block, pilot signal is extracted from the receive signal, and then the extracted signal is transmitted to the SASpatialMultiplexing block. At this moment, the pilot signal is provided by external properties in the MIMOTxDemo block. The SASpatialMultiplexing block operates a channel estimation using pilot signal and SM decoding algorithm such as ZF (Zero Forcing).
Figure 3(b) shows a block diagram of SA subsystem for the beamforming waveform. The structure of blocks is similar to the case of SM waveform described abbove. SATxDemo block which generates the transmit signal from binary data, has only one port. The external properties provide the pilot signal to that block. DirectiveChannel block produces a receive signal with a steering vector in accordance with the DoA. The SARxDemo block performs the same function as MIMORxDemo block does, which is described above. Finally, SABeamforming block calculates a weight vector corresponding to the receive signal.

3.2 Implementation of XCVR subsystem

The current version of XCVR API provided by WINNF does not support multi-antenna system. Therefore, it should be extended into a multichannel XCVR to support SA API.

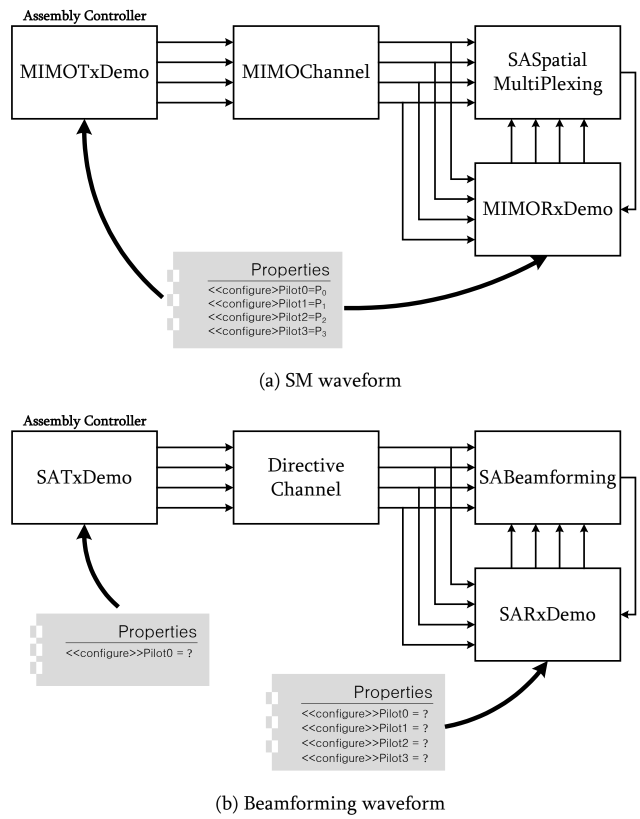

Figure 3 Block diagram of SA subsystem

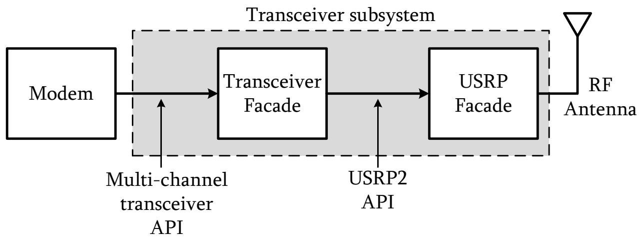

Figure 4 Block diagram of implemented multichannel XCVR subsystem

The implemented multichannel XCVR subsystem using USRP2, as shown in Figure 4, has interfaces such as XCVR API and USRP2 API. The signal from Modem is transmitted to XCVR façade through the XCVR API. In order to support the multi-antenna processing, XCVR API should be extended into multichannel structure. For simplicity, the extension has been simply realized by calling out the XCVR libraries appropriately. The USRP2 API, which controls the USRP2, is functioning between XCVR Façade and RF antenna as shown in Figure 4.

4. VERIFICATION AND EXPERIENCE

In order for the SA API and XCVR API to be recognized as a standard in WINNF, each API should be implemented and verified in such a way that it can adopt the JPEO and ETSI standard. To meet this need, we implemented both APIs in accordance with SCA, and then performed experimental tests for the implemented systems.

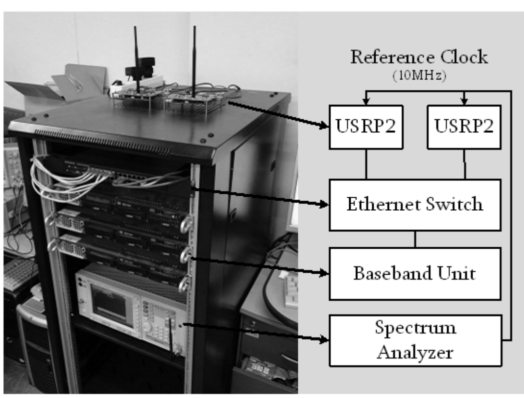

Figure 5 Implemented system of SA and XCVR subsystem

Figure 5 shows our experiment environment of SA and XCVR subsystem. The entire system including both SA and XCVR has multiple transmit and receive antennas together with the USRP2 modules. The receive signal from each antenna is transferred to the baseband unit through the Ethernet switch. In our system, the baseband unit is a personal computer which has an OSSIE platform. In transmit mode, it generates OFDM (Orthogonal Frequency Division Multiplexing) symbols. At the transmitter, a single transmit antenna is used in the case of beamforming waveform and four antennas are used in the case of SM waveform. At the receiver, four receive antennas with a linear array geometry are used for the beamforming waveform. In the case of SM waveform, there are 4 receive antennas placed far, more than ten times wavelength, from each other.

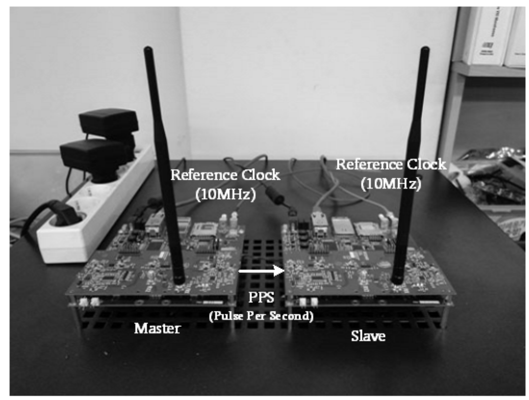

Figure 6 Transmit and receive antennas using USRP2

Figure 6 illustrates the transmit and receive antennas that are connected to USRP2s. To synchronize the timing between two antennas, each USRP2 uses a common reference clock of 10MHz, also the Master USRP2 sends a PPS (Pulse Per Second) signal to the Slave USRP2.

It is necessary to verify if the implemented XCVR subsystem operates properly. For this reason, each of two antennas periodically transmits a preamble signal in order to check the synchronization between the two antennas.

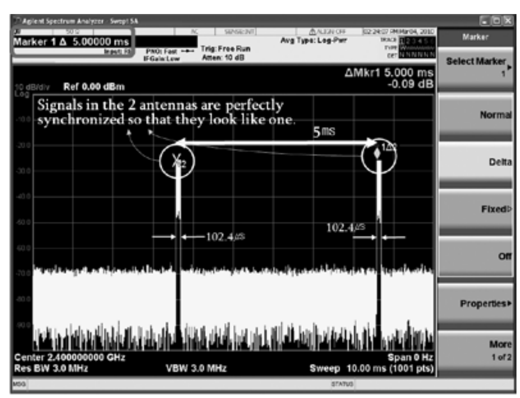

Figure 7 Transmit signal in time domain

Transmit signal in time domain is shown in Figure 7. In the picture, the interval between the preamble signals is 5 ms, and the duration of a preamble signal is 102.4 µs. This means that signals in the two antennas are perfectly synchronized.

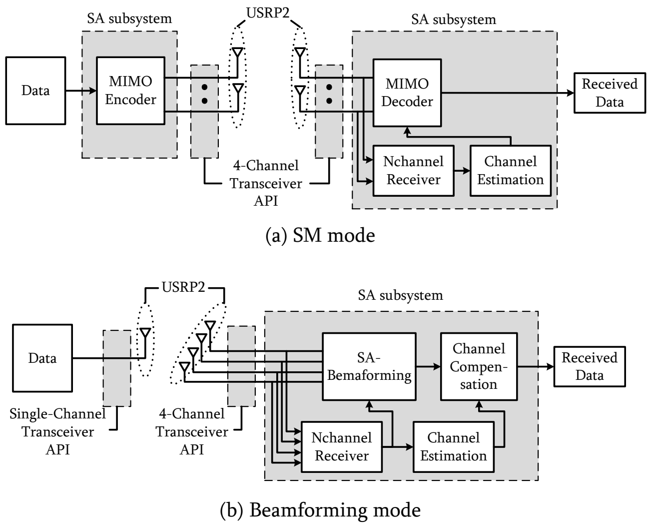

Figure 8 Block diagrams of entire system, combination of SA subsystem and transceiver subsystem

Figure 8 represents block diagrams of the entire system that consists of combination of SA subsystem and XCVR subsystem. As mentioned earlier, all blocks are implemented on OSSIE platform. The system operates in either SM waveform or beamforming waveform. It is noteworthy that NChannelReceiver component and ChannelEstimation component are reusable because the operations of those two components are independent of waveforms.

Figure 8(a) shows a block diagram for SM waveform. In transmitter, SA subsystem operates for the MIMO encoding. The output of SA subsystem is transmitted to each of four transmit antennas through the corresponding port of four USRP2s in accordance with the XCVR API. The receive signals from the four receive antennas are sent to the SA subsystem as indicated in the 4-channel XCVR API. SA subsystem in the receiving mode decodes the receive signal in the case of MIMO signal. First, the NChannelReceiver component transmits the pilot signals to the ChannelEstimation component as shown in Figure 8(a). Then, MIMODecoding component decodes the MIMO signal from the received signal and estimated channel coefficient from the pilot signal using one of the MIMO decoding algorithms, for example, ZF algorithm. Figure 9 shows the constellation of the received signal after MIMO decoding algorithm.

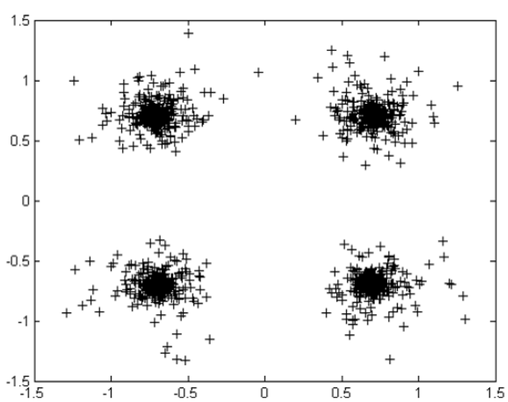

Figure 9 Constellation of received signal after MIMO decoding algorithm

In the case of beamforming waveform, as shown in Figure 8(b), a transmit antenna sends the signal to four receive antennas. The receive signals are forwarded to NChannelReceiver component in SA subsystem using 4-channel XCVR API. The SABeamforming component calculates a weight vector with the pilot signal which is the output of NChannelReceiver component. We have observed that the beam pattern is provided with its main lobe being generated along the direction of desired signal. Figure 10 illustrates the beam pattern provided by the implemented SA sybsystem when its DoA is 20°. Note that, as shown in Figure 10, the main beam has been generated along the direction of 20°.

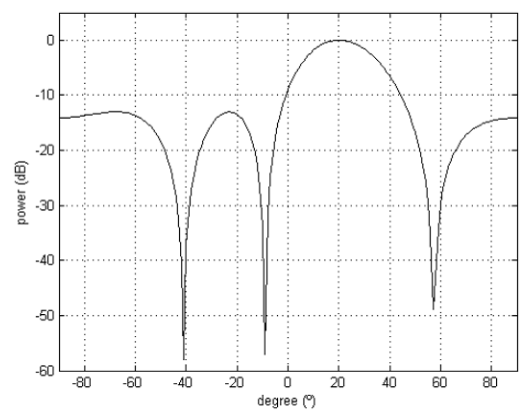

Figure 10 Beam pattern (DoA: 20°)

5. CONCLUSION

In this paper, we have shown an implementation of SA API and XCVR API using OSSIE and USRP2. Through the implementation, it has been verified that the two APIs are compatible to each other, meaning that SDR SA developed by the SA work group of WINNF can be realized using the XCVR developed by XCVR work group of WINNF. In the implementation of XCVR, we have checked the exact synchronizations among the multiple antennas in the SCA-based environment. In the implementation of SA, we have confirmed the functionalities of various kinds of SAs such as beamforming, SM MIMO, STC MIMO, etc on OSSIE platform. In short, interoperability between the two APIs has been confirmed through the implementation shown in this paper. Both APIs have been developed during the procedure of standardization effort for ETSI and JPEO on behalf of WINNF. This paper concludes that the implementation of SA and XCVR API will accelerate the standardization of both APIs.

6. ACKNOWLEDGEMENT

This work was supported by the Seoul R&BD Program (PA090743).

7. REFERENCES

[1] "PIM and PSM for smart antenna specification," *formal/2009-01-01*

[2] S. Hyun, J. Kim, S. Choi, L. Pucker and B. Fette, "Standardizing smart antenna API for SDR networks," *RFDesign*, pp. 34-40, Sep. 2007.

[3] E. Nicollet and L. Pucker, "Standardizing Transceiver APIs for Software Defined and Cognitive Radio," *RFDesign*, pp. 16-20, Feb. 2008.

[4] "Transceiver Facility Specification," *SDRF-08-S-0008-V1.0.0*

[5] E. Paone, "Open-Source SCA Implementation-Embedded and Software Communication Architecture," Feb. 2010

[6] J.H. Reed, "Smart Antennas: A system Level Overview for Software Defined Radios for Creating an API," *SDR Forum Input Document*, SDRF-04-I-0057-V0.00, Jun. 2004.

[7] M. Carrick, D. Cormier, C. Covington, C. B. Dietrich, J. Gaeddert, B. Hilburn, C. I. Phelps, S. Sayed, D. Seeralan, J. Snyder, and H. Volos, "OSSIE 0.8.0 Installation and User Guide," Sep. 2009.