

## 멀티모드 단말기의 라디오 라이브러리를 위한 표준 기능 블록의 선정 및 구현

정 일 도\* · 최 승 원\*\*

### *Selection and implementation of Standard Functional Blocks for Radio Library in multi-mode mobile device*

Jung Ildo · Choi Sengwon

#### 〈Abstract〉

The European Telecommunication Standards Institute (ETSI) Technical Committee (TC) Reconfigurable Radio Systems (RRS) is standardizing the multi-mode Mobile Device (MD). The configuration of multi-mode MD is determined by the downloaded mobile communication standard software. In this paper, we introduce the Radio Library concept for multi-mode MD which is one of the key components of RRS standard. This paper also introduces the Standard Functional Block which is a part of Radio Library. A method for selecting efficiency SFBs for multi-mode MD is presented and a Radio Library is generated based on the selected SFBs. This paper also shows sample Standard Functional Block Set which included in Radio Library. In order to verify the compatibility of the generated Radio Library which was made by C language, we implement the LTE Rel-10 and Wi-Fi(802.11b) to show the efficiency of generating a mobile communication standard software based on the Radio Library. Then using the Prograph Visual Programming Marten™ 1.6.4, we compiled our LTE Rel-10 and Wi-Fi(802.11b) source code.

Key Words : Radio Virtual Machine (RVM), Software Define Radio (SDR), Standard Functional Block (SFB)

## I. 서론

최근 macro, pico, femto cells을 조합한 RANs (Radio Access Networks)를 지원하는 Heterogeneous

Network concept (Het-Net)이 널리 배포됨에 따라, Het-Net을 실현하는 데 중요한 부분인 재구성 가능한 MD(Mobile Device)를 위한 연구와 표준화 활동이 활발하게 진행되고 있다[1-2]. 이에 따라 유럽통신표준기구(European Telecommunications Standards Institute, ETSI) 재구성 가능한 라디오 시스템

\* 한양대학교 전자컴퓨터통신공학과 석사과정

\*\* 한양대학교 전자컴퓨터통신공학과 교수 (교신저자)

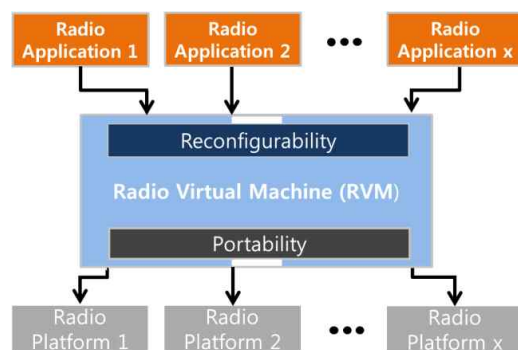
(Reconfigurable Radio Systems, RRS) 기술위원회 (Technical committee, TC)에서는 2008년부터 HetNet 을 지원할 수 있는 재구성 가능한 MD를 위한 소프트웨어 아키텍처와 시스템 인터페이스의 표준을 이끌어왔다 [3-6]. 특히 TC RRS는 SDR(Software Defined Radio) 기술과 CR(Cognitive Radio) 기술을 포괄하는 개념인 RRS에 대한 표준화를 진행하고 있다[7]. RRS에서 소개되는 소프트웨어 모뎀 기반 멀티모드 단말기는 사용자가 LTE, Wi-Fi, WiMAX(World interoperability for microwave access) 등과 같은 라디오 애플리케이션(Radio Application, RA)을 라디오 앱 스토어(Radio App Store)에서 다운로드 받아 설치함으로써, 원하는 통신 표준규격을 사용할 수 있는 단말기이다[1].

본 논문에서는 소프트웨어 모뎀 기반 멀티모드 단말기에서 사용되는 표준 기능 블록 (Standard Functional Block, SFB)들을 제안함으로써 아직 정의 되지 않은 SFB의 선정 방법을 연구해 보고자 한다. 이에 따라 본 논문의 2장에서는 Radio Virtual Machine과 라디오 라이브러리를 소개하고 SFB의 역할과 그 특징들을 살펴본다. 3장에서는 효율성을 보장하는 Radio Library를 만들기 위한 SFB선정 방법을 기술하며, 4장에서는 제안하는 방법으로 선정된 SFB들을 이용하여 라디오 라이브러리를 생성한다. 또한, 생성한 라디오 라이브러리를 사용하여 LTE rel-10, Wi-Fi(802.11b) RA를 만들어 봄으로써 제안하는 방법의 효율성을 검증한다. 마지막 5장에서는 결과 분석 및 결론을 맺는다.

## II. 라디오 라이브러리

TC RRS에서 연구하고 있는 멀티모드 단말기는 사용자가 원하는 RA (Radio Application)를 다운로드

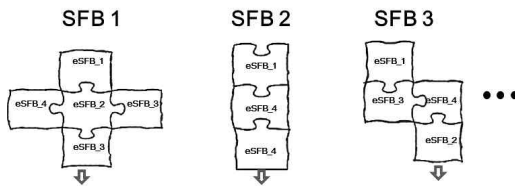
및 설치를 통하여 자신이 원하는 통신 규격을 사용할 수 있는 단말기이다[3]. RA는 LTE, Wi-Fi, WiMAX와 같은 통신 표준규격을 멀티모드 단말기에서 사용할 수 있도록 제작된 Software를 말한다[1]. 멀티모드 단말기는 RA를 다운로드 받아 설치함으로써 해당 단말기의 무선 인터페이스를 재구성할 수 있게 되며, 다양한 이동통신 표준기술을 사용자와 망 관리자에 의해 자유자재로 변경하여 사용할 수 있게 된다[6]. RRS에서 소프트웨어 제공자(RA Provider)는 RA를 만들 때 Signal Processing에 사용되는 기능들을 포함한 라디오 라이브러리를 참조한다. 이 라디오 라이브러리는 RA Provider가 RA를 만들 때 어떤 플랫폼에서 동작할지 고려하지 않아도 되게 만드는 요소이다[3]. 왜냐하면, RA Provider가 참조하는 라디오 라이브러리는 플랫폼과 독립적으로 해당 Functional Block의 정보만을 담고 있기 때문이다. 이 라디오 라이브러리를 참조해서 만들어진 RA code는 해당 플랫폼에 다운로드 되어 RVM(Radio Virtual Machine)이라는 특별한 Virtual Machine을 통해 플랫폼에 최적화되도록 Back-End 컴파일 되고[8], 이를 통해 RA Provider는 RA를 만들 때 플랫폼을 고려하지 않을 수 있다[1-2].



<그림 1> Radio Virtual Machine Concept

<그림1>은 RVM Concept에 대하여 나타내고 있다

[4]. RVM은 abstract machine으로써 RA code들을 라디오 앱 스토어에서 다운로드 받아 분석한 다음 플랫폼에 맞게 RA code들을 컴파일 하는 역할을 한다. 이를 통해서 하나의 RA code는 여러 가지 플랫폼에서 최적화되어 사용될 수 있다.

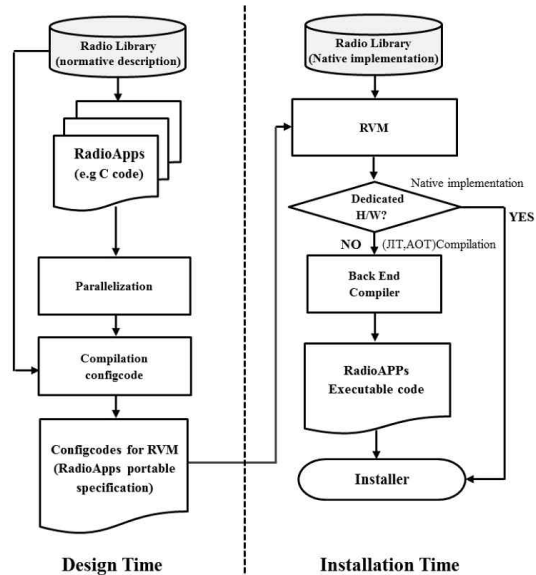


<그림 2> Standard Functional Block 구조의 예시

라디오 라이브러리는 RRS에서 사용하는 라이브러리로서 모든 SFB(Standard Functional Block)를 포함하고 있는 라이브러리를 말한다. 여기서 SFB는 플랫폼 제공자에 의해 Signal Processing에 사용되는 FFT, VITERBI DECODER, TURBO DECODER 등과 같은 기능을 해당 플랫폼에서 동작 가능한 형태로 제공하는 것을 말하는데, <그림 2>에서 보듯이 eSFB(elementary SFB)로 구성되어 있다. eSFB는 나눌 수 없는 기본적인 add, subtract, Logical AND, Logical OR 등과 같은 기능을 제공하는 Functional Block이다[3]. 이와 같은 SFB를 포함하고 있는 라디오 라이브러리는 <그림3>과 같이 RA 코드를 배포하는 과정에서 design time과 Installation Time에서 사용되며, 각각 Reference 라디오 라이브러리와 Native 라디오 라이브러리 형태로 사용된다.

Design time에서 RA Provider는 Reference 라디오 라이브러리를 참조하여 RA를 만들게 된다. 이 라이브러리는 SFB들에 대한 normative description을 포함하고 있으며, normative description은 모든 SFB들의 기능, cost 및 소요시간 등과 같은 정보를 제공한다[3]. RA Provider는 이 description에서 필요한 SFB

들을 사용하여 RA를 만들게 되고, RA는 <그림 3>에서 나타난 바와 같이 RVM이 읽을 수 있는 형태로 컴파일 된다[9].



<그림 3> Radio Application의 배포 및 설치

Installation Time에서 플랫폼 내부의 RVM은 Native 라디오 라이브러리를 참조한다. 여기서 Native 라디오 라이브러리란 Reference 라디오 라이브러리 내부의 SFB들이 어떻게 해당 플랫폼에서 동작하는지에 대한 정보를 제공하는 라이브러리를 말한다. 즉 이 정보에 따라 플랫폼은 SFB를 실행할 때 dedicated hardware accelerators를 사용할지 아니면 programmable devices를 사용할지를 결정하게 된다 [2]. 결과적으로 플랫폼은 Native 라디오 라이브러리를 참조하여 다운로드 받은 RA를 RVM을 통해 해석하고 해당 플랫폼에 맞게 컴파일 하여 RA를 설치할 수 있다[10].

하지만 현재 ETSI TC RRS 표준문서에서는 라디오 라이브러리의 SFB는 정의하고 있지 않고, 오직 기본

적인 연산만을 하는 23개의 eSFB만을 정의하고 있다 [3]. 제공하는 eSFB 만으로 RA를 만들기에 상당한 시간과 노력이 필요하며 결국 RA Provider는 비효율적으로 RA를 만들 수밖에 없는 상황이다. 따라서 본 논문에서는 RA Provider가 RA를 만들 때 효율적으로 RA를 만들 수 있도록 하는 라디오 라이브러리의 SFB의 선정 방법을 제안하려 한다.

### III. 제안하는 효율적인 SFBs

#### 3.1 SFB 선정방법

효율적인 라디오 라이브러리를 만들기 위해서는 2가지 기준을 사용하여 SFB를 선정한다. 그 중 첫 번째 기준인 CRE는 식 (1)에서 나타난 바와 같이 Functional Block이 해당 RA에서 얼마나 많은 Computational Resource를 소모했는지를 알 수 있는 값이다.

$$CRE = \frac{\text{Functional Block Computational Resource}}{\text{Total RA Computational Resource}} \times 100 \quad (1)$$

RA는 실행되면서 Computational Resource를 소모한다[5]. 이 과정에서 특정 Functional Block은 RA 내부의 다른 Functional Block 대비 많은 Computational Resource를 소모한다. 왜냐하면, 비터비, 터보 디코더와 같은 복잡한 알고리즘을 포함하는 Functional Block은 많은 eSFB를 사용해서 구성해야만 하고 이는 곧 많은 Computational Resource를 요구하게 되기 때문이다. 따라서 Computational Resource를 많이 소모하는 Functional Block을 SFB로 선정함으로써 플랫폼 제공자가 dedicated hardware

accelerators로 해당 SFB를 제공할 수 있게 한다. 이를 통해 멀티모드 단말기에서 SFB를 실행할 때 이득을 얻을 수 있다. 또한 CRE가 높은 Functional Block은 높은 Complexity를 갖고 있기 때문에 RA Provider가 eSFB만으로 구현하기에는 비효율적이다. 따라서 CRE가 높은 Functional Block을 SFB로 등록함으로써 RA Provider에게 편의를 제공할 수 있다.

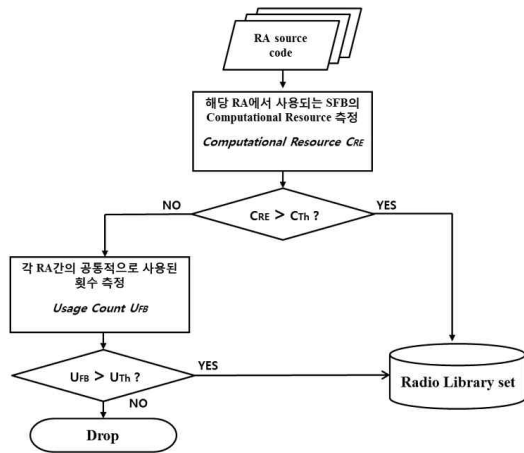
SFB를 선정하는데 사용하는 2번째 기준은 서로 다른 RA들 사이에서 공통으로 사용되는 Functional Block의 사용 횟수를 나타내는 UFB이다. UFB를 구하는 식은 식 (2)와 같다.

$$UFB = \sum_{n=1}^M \text{Usage of } (RAn), n = 1, 2, 3, 4 \dots M \quad (2)$$

식 (2)에서 나타난 바와 같이 UFB는 서로 다른 M개의 RA에서 해당 Functional Block의 사용 여부를 측정하여 그 총합을 나타낸 값이다. 측정한 UFB를 이용하면 RA들 사이에서 재사용 되는 Functional Block을 찾아 SFB로 선정할 수 있다.

식 (1)과 식 (2)에서 측정한 CRE와 UFB를 이용하여 SFB를 선정할 때 본 논문은 Cth(Computational Resource Threshold)와 Uth(Usage Count threshold)를 사용하였다. 이 값들은 전체 RA에서 측정한 Cth와 Uth의 평균값을 적용하도록 하였으며 Cth는 12%, Uth는 2회로 설정하였다.

<그림 4>는 SFB를 선정하기 위한 방법을 나타내고 있다. <그림 4>에서 RA Source code는 LTE, Wi-Fi, WiMAX와 같은 표준 이동통신 RA code들을 나타낸다. RA code가 입력되면 CRE의 값을 측정한 후, 측정한 CRE값이 Cth보다 높으면 라디오 라이브러리의 SFB로 등록되고 Cth보다 낮으면 다음 단계로 진행하게 된다. CRE값을 만족하지 못해 SFB로 등록되지 못한 Functional Block은 다시 서로 다른 RA 사이에서



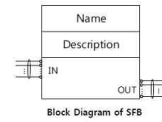
<그림 4> Standard Functional Block의 선정 방법

얼마나 공통적으로 자주 사용되었는지를 나타내는 UFB를 측정하는 단계를 거치게 된다. 만약 측정된 UFB 값이 Uth보다 높으면 SFB로 등록되며, 그렇지 않으면 최종적으로 SFB로 등록될 수 없게 된다.

### 3.2 선정된 SFBs

본 논문에서는 MacOS 기반 Prograph Visual Programming Marten™ 1.6.4를 이용하여 LTE와 Wi-Fi(802.11b) RA를 구현하였다. SFB는 HSPA, HSDPA, LTE Rel-9, Wi-Fi(802.11g), Source Code를 이용하여 3장에서 제안하는 방법으로 선정하였으며, Cth는 해당 RA의 12%이상, Uth는 2회 이상으로 설정하였다. 그 결과 eSFB 들의 조합으로 이루어진 <그림 5>와 같은 38개의 SFB를 선정할 수 있었다.

<그림 5>는 본 논문에서 제안하는 SFB들의 대표적인 38개의 SFB를 나타내고 있다. <그림 5>에서 나타난 바와 같이 SFB들은 CRE값과 UFB 값을 만족하는 SFB들을 포함하고 있으며, 이에 따라 해당 SFB가 포함된 범주를 분류하고 있다. 또한, 각 SFB마다 자신의 입력 값을 나타내는 IN과 출력 값을 나타내는 OUT



Name	Cre	Urs	Category	IN	OUT	Description
FFT	15%	2	Common	IN 1= Data IN 2= FFT_Point	OUT 1=	2-Radix Fast Fourier Transform
IFFT	15%	2	Common	IN 1= Data IN 2= IFFT_Point	OUT 1=	2-Radix Inverse Fast Fourier Transform
Convolution	8%	3	Common	IN 1=x IN 2=h IN 3=N	OUT 1=y	Convolution $y^j = \sum_k h^k * x^{j-k}$

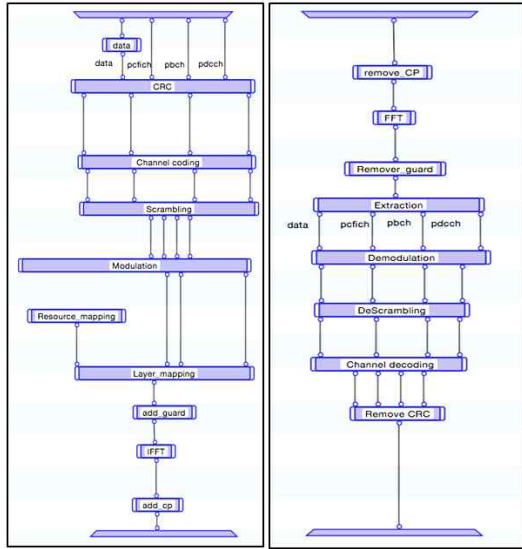
Name	Cre	Urs	Category	IN	OUT	Description
Turbo Encoder	14%	1	LTE	IN 1=x, IN 2=N IN 3=s (s=0 : WCDMA, s=1 : LTE)	OUT 1=y	Turbo Encoder
Turbo Decoder	16%	1	LTE	IN 1=x, IN 2=N IN 3=s (s=0 : WCDMA, s=1 : LTE)	OUT 1=y	Turbo Decoder
Convolution sum	12%	2	Common & Wi-Fi (802.11b)	IN 1=x, IN 2=h	OUT 1=y	$y^j[x] = \sum_{k=-\infty}^{\infty} x[k] * h^k$
Spreading	12%	1	Wi-Fi (802.11b)	IN 1=x IN 2=sequence	OUT 1=y	Spreading Data from input sequence

<그림 5> 제안하는 SFB들의 대표적 기능

항목을 포함하고 있으며, 어떤 기능을 수행하는지 설명하는 Description 항목을 포함하고 있다.

선정된 SFB들은 C language를 통해서 구현하였으며 GCC 컴파일러에 의해 MacOS 기반 애플리케이션 개발 프로그램인 Prograph Visual Programming Marten™ 1.6.4 에서 구동되도록 컴파일 하였다. 다시 말해 C language로 제작된 SFB Source Code를 통해서 RA의 실행 환경에서 구동될 수 있는 라디오 라이브러리를 생성한 것이다.

이처럼 Radio Library를 통하여 <그림 6>과 같이 LTE Rel-10 RA를 구현할 수 있었으며 <그림 7>의 eSFB만을 사용해서 LTE Rel-10 RA 를 만들었을 때와 Functional Block 의사용 횟수를 비교할 수 있었다. eSFB 는 간단하고 기본적인 기능밖에 제공하지 않기 때문에 eSFB만을 사용해서 LTE RA를 만들었을 때는 매우 많은 eSFB를 사용해야 했다. 결과적으로 RA를 만들 때 직접 RA에 사용되는 모든 복잡한 알고리즘을 eSFB의 조합으로 구현해야만 했고, 이는 RA제작



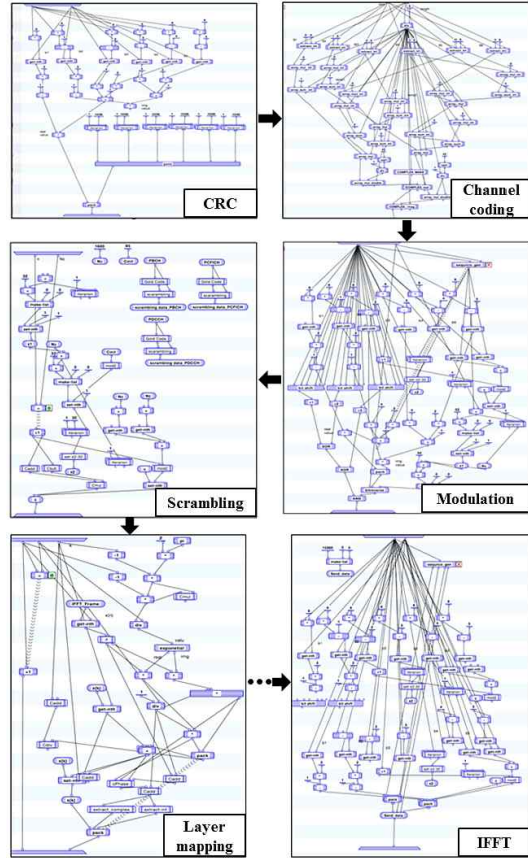
<그림 6> SFB를 사용한 LTE Rel-10 RA 구현

과정의 비효율성을 증대시켰다. 반면 <그림 6> 에서 보다시피 제안한 방법으로 선정된 SFB를 사용해서 LTE RA를 만들었을 때는 RA의 복잡한 알고리즘을 SFB를 이용해서 간단하게 구현할 수 있었다.

<표 1> Functional Block 사용 횟수

Functional Block 사용 횟수			
LTE		Wi-Fi(802.11b)	
SFB 사용	SFB 미사용	SFB사용	SFB 미사용
26회	384 회	12회	164 회

<표 1>은 LTE rel-10과 Wi-Fi(802.11b) RA를 만들 때 eSFB만을 사용할 때와 제안한 SFB를 사용할 때의 Functional Block 사용 횟수를 비교한 TABLE이다. LTE Rel-10 RA는 Functional Block 사용 횟수가 384 회에서 26회로 약 93%의 감소 효과를 얻을 수 있었고, Wi-Fi(802.11b) RA 는 164회에서 12회로 약 91%의 감소효과를 얻을 수 있었다. 이는 전체 Functional Block 사용 횟수를 생각해 봤을 때 큰 감소폭임을 알



<그림 7> eSFB만을 사용한 LTE Rel-10 RA 구현

수 있다. 또한 논문[10]에서 소개하는 방법으로 구현한 LTE rel-10 시스템과 비교 했을 때, 제안하는 SFB 는 중복되는 Functional Block을 호출 하지 않기 때문에 다운로드를 기준으로 디버깅 까지 걸리는 시간이 단축되었음을 알 수 있었다. LTE Rel-10 RA를 eSFB 만을 이용해서 구동 했을 때 15360개의 비트를 디버깅 하는데 까지 4.2ms가 걸렸지만 제안하는 SFB를 사용했을 때 3.6ms의 시간이 걸렸으며, 약 15%정도의 시간이 줄어든 것을 알 수 있었다. 이를 통해서 본 논문에서 제안하는 방법으로 선정된 SFB가 연산시간의 효율성을 보장함을 알 수 있었다.

#### IV. 결론

본 논문에서는 ETSI TC RRS에서 표준화를 진행 중인 멀티모드 단말기를 설명하고 여기서 사용되는 라디오 라이브러리와 SFB의 선정방법을 제안하였다. 또한, 제안하는 방법으로 LTE rel-10와 Wi-Fi(802.11b) 구현에 효율적인 라디오라이브러리를 생성 하였다. 3장에서 소개한 방법을 이용해서 라디오 라이브러리를 구성한 결과 LTE\_rel10 과 Wi-Fi(802.11b)를 위한 38개의 SFB를 선정할 수 있었으며, 선정한 SFB를 바탕으로 LTE\_Rel-9, Wi-Fi(802.11b)의 통신 표준을 구현할 수 있었다.

이를 통해서 SFB를 사용했을 때와 사용하지 않았을 때의 RA 제작과정의 효율성을 비교할 수 있었고, 그 결과 Functional Block의 사용 횟수가 90%가량 감소했음을 알 수 있었다. 이를 통해서 제안하는 SFB들이 효율적임을 알 수 있었으며, RA provider는 본 논문에서 제안하는 라디오 라이브러리를 사용한다면 효율적으로 LTE\_rel10와 Wi-Fi(802.11b)RA를 만들 수 있음을 확인 할 수 있었다. 아직은 SFB선정 Committee가 없어서 효율성이 보장되는 다양한 SFB들이 없지만, 미래에 Committee가 생성되고 멀티모드 단말기 시장이 활성화된다면 다양한 SFB들을 얻을 수 있을 것이라 예상된다.

#### Acknowledge

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No. R0166-16-1031, 멀티모드 단말기용 소프트웨어 모뎀 API를 지원하는 범용 프로토콜 국제표준 개발)

#### 참고문헌

- [1] ETSI EN 303 095: "Reconfigurable radio system (RRS); radio reconfigurable re-lated architecture for mobile devices," ver. 1. 1. 1, 2015.
- [2] ETSI TR 102 839: "Reconfigurable Radio Systems (RRS); Software Defined Radio (SDR) Multiradio Interface for Software Defined Radio (SDR) Mobile Device Architecture and Services," 2011.
- [3] ETSI TS 103 146-4: Reconfigurable Radio Systems(RRS); Mobile Device Information Models and Protocols; Part 4 :Radio, 2013.
- [4] Riadh Ben Abdallah, Tanguy Risset and Antoine Fraboulet, "A Solution for SDR Portability and platform Reconfigurability," IEEE International Symposium, Parallel & Distributed Processing, 2009, pp. 1970-1977.
- [5] Seungwon Choi, and Markus Mueck, "Radio Virtual Machine: Virtual Machine for Reconfigurable Radio," ETSI WORKSHOP, 2016, pp. 63-68.
- [6] Yong Jin, Kyunghoon Kim, Donghyun Kum, Seungwon Choi, and Vladimir Ivanov, "The ETSI Standard Architecture, Related Interfaces, and Reconfiguration Process for Reconfigurable Mobile Devices," IEEE Communications Magazine, 2015, pp. 38-46.
- [7] Yong JIN, Chiyong Ahn, Seungwon Choi, Markus Mueck, Vladimir Ivanov, and Sakar, "Design and Implementation of ETSI-Standard Reconfigurable Mobile Device for Heterogeneous Network," IEICE TRANSCOMMUNICATION, Vol. 99-B, No. 8, 2016, pp.

- 78-86.
- [8] 안치영 · 김 용 · 최승원, “ETSI 표준을 위한 소프트웨어 모뎀 기반 멀티모드 단말기의 표준 개발,” 한국전자과학회 전자과 기술, 제5권, 제24호, 2013, pp. 65-72.
- [9] 김한택 · 안치영 · 김준 · 최승원, “모바일 컴퓨팅 플랫폼을 이용한 SDR 기반 MOBILE WIMAX 수신기 구현,” 디지털산업정보학회 논문지, 제8권, 제1호, 2012, pp. 117-123.
- [10] 금동현 · 최승원, “멀티모드 단말기를 위한 셀 경계 지역에서의 SINR 기반 사용자 선택 방법,” 디지털산업정보학회 논문지, 제11권, 제3호, 2015, pp. 63-67.
- [11] 정성원 · 금동현 · 최승원, “WPAN에서 간섭을 피하기 위한 멀티모드 단말기 채널등급 방법,” 디지털산업정보학회 논문지, 제11권, 제3호, 2015, pp. 91-97.



최 승 원  
Choi Seungwon

2012년~현재  
HY-MC 연구센터 센터장

2002년~2011  
HY-SDR 연구센터 센터장

1992년~현재  
한양대학교 전자전기공학부 교수

1990년~1992년  
일본 우정성 통신연구소  
신입연구원

1989년~1990년  
ETRI 선임연구원

1988년~1989년  
미국 Syracuse대학 전지 및  
전산과 교수

1988년 12월 미국 Syracuse대학 전기공학  
(공학박사)

1985년 12월 미국 Syracuse대학 컴퓨터공학  
(공학석사)

1982년 2월 서울대학교 전자공학 (공학석사)

1980년 2월 한양대학교 전자공학 (공학학사)

관심분야 : SDR, 이동통신, 신호처리  
E-mail : choi@dsplab.hanyang.ac.kr

논문접수일: 2016년 8월 19일  
수 정 일: 2016년 9월 5일  
계재확정일: 2016년 9월 8일

■ 저자소개 ■



정 일 도  
Jung Ildo

2015년 3월~ 현재  
한양대학교 전자컴퓨터통신공학과  
석사과정

2015년 2월 한국산업기술대학교 메카트로닉스  
공학과 (공학학사)

관심분야 : LTE-A, SDR  
E-mail : jung\_ildo@dsplab.hanyang.ac.kr