

## GNU Radio와 TI DSP의 NDK를 이용한 LTE SDR 플랫폼 구현

진 화 종\* · 김 대 진\*\* · 최 승 원\*\*\*

### *Implementation of SDR Platform for LTE using GNU Radio and NDK of TI DSP*

Jin Hwajong · Kim Daejin · Choi Seungwon

#### 〈Abstract〉

This paper presents an implementation method using NDK (Network Developer's Kit) of GNU (GNU is Not Unix) Radio and Multicore DSP (Digital Signal Processor) to implement LTE (Long Term Evolution) SDR (Software Defined Radio) Platform. In order to satisfy 1.4MHz, 3MHz, 5MHz and 10MHz of the bandwidth supported by LTE, USRP (Universal Software Radio Peripheral) X series which is an RF (Radio Frequency) transceiver of Ettus Research was used. To control this, GNU Radio which is an open source software radio toolkit was used. We also used NDK from TI (Texas Instruments) DSP to transfer data between USRP and DSP. Experimental results show throughput results according to each bandwidth, thus confirming the feasibility of implementing LTE SDR Platform using GNU Radio and NDK of TI DSP.

Key Words : GNU Radio, NDK, LTE, SDR, DSP

## I. 서론

SDR (Software Defined Radio) 기술은 무선 이동 통신 시스템에서 기존의 하드웨어로 구성된 부분을 하드웨어의 교체 없이 모듈화 된 소프트웨어의 재구성만으로 진화하는 통신 및 서비스 기능 등을 지원하는 기술이다[1, 2]. SDR 기술을 사용하여 하드웨어 교체 없이 소프트웨어의 재구성만으로 다양한 서비스를 제공함으로써, 통신 사업자는 비용의 효율적 운용 뿐만 아니라 사용자 개개인의 특성에 최적화 된 서비

스를 제공할 수 있는 장점이 있다[1]. SDR 기술은 빠르게 발전하는 통신 기술과 다양화되는 무선통신 방식의 특성에 대응할 수 있는 차세대 이동통신 시스템 구현의 핵심이라 할 수 있다[2].

SDR 플랫폼에서의 모뎀 프로세싱을 위해 GPU (Graphics Processing Unit), FPGA (Field Programmable Gate Array), DSP (Digital Signal Processor) 등의 많은 프로세서가 사용되어 왔다[3, 4, 5]. GPU는 많은 코어를 가지고 있어 병렬 처리에는 용이하나 전력 소모가 많고 순차적인 연산에는 취약한 단점이 있다. 또한 FPGA는 사용자가 원하는 논리 회로를 직접적으로 프로그래밍이 가능하다는 장점이 있으나 설계와 검증 시간이 오래 걸린다는 단점이 있

\* 한양대학교 전자컴퓨터통신공학과 석사과정

\*\* 한양대학교 전자컴퓨터통신공학과 박사과정

\*\*\* 한양대학교 전자컴퓨터통신공학과 교수

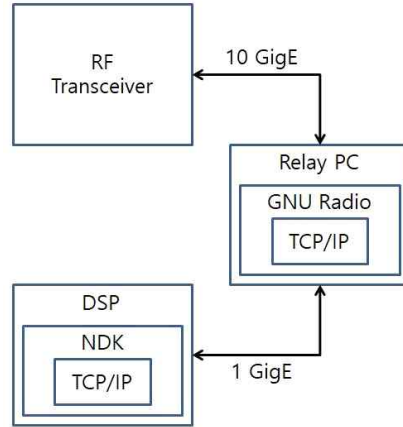
다. 하지만 본 논문에서 SDR 플랫폼의 모뎀 프로세싱을 위해 사용하는 멀티코어 DSP는 여러 코어를 이용한 병렬 처리가 가능하며, 순차적인 연산에 최적화되어 있다. 또한 하드웨어 가속기를 통해 계산량이 많은 알고리즘 연산을 빠른 시간에 처리할 수 있으며, 하이 레벨 (high level) 언어로 프로그래밍이 가능하므로, 설계가 용이하고 비용이 적게 드는 등의 장점을 가진다[6, 7].

본 논문에서는 RF (Radio Frequency) 송/수신을 위한 USRP (Universal Software Radio Peripheral)와 모뎀 프로세싱을 위한 DSP로 구성된 LTE (Long Term Evolution) SDR 플랫폼을 제안 한다. 이 때, DSP와 USRP 간의 데이터 송/수신을 위한 인터페이스는 1 GigE (Gigabit Ethernet) 인터페이스와 TCP/IP (Transmission Control Protocol/Internet Protocol)를 사용하였다. RF 송수신기의 제어를 위해 GNU (GNU is Not Unix) Radio를 사용하였으며, TCP/IP를 사용하기 위해 TI (Texas Instruments) DSP의 NDK (Network Developer's Kit)를 사용하였다. 제안하는 SDR 플랫폼의 실현 가능성을 확인하기 위해 LTE의 대역폭인 1.4MHz, 3MHz, 5MHz 그리고 10MHz에 대해 처리량을 측정하였다. 실험 결과를 통해 GNU Radio와 TI DSP의 NDK를 이용한 LTE SDR 플랫폼 구현 가능성을 확인한다.

본 논문의 구성은 다음과 같다. 2장에서는 LTE SDR 플랫폼 구조와 구현에 사용된 USRP X310, GNU Radio 그리고 NDK를 설명한다. 3장에서는 LTE SDR 플랫폼의 구현 사항과 실험 결과에 대해서 설명하고, 마지막 장에서 결론을 맺는다.

## II. GNU Radio와 DSP의 NDK를 기반으로 하는 LTE SDR 플랫폼 구현

<그림 1>은 GNU Radio와 TI DSP의 NDK를 사용

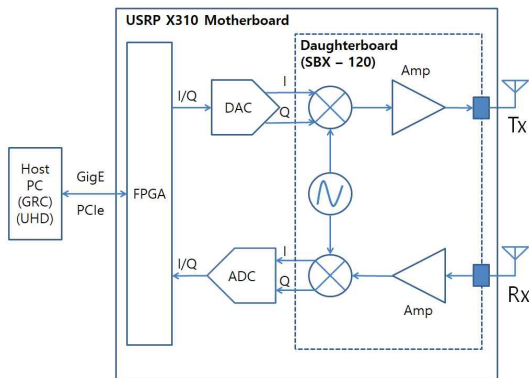


<그림 1> 제안하는 LTE SDR 플랫폼 블록 다이어그램

하여 LTE SDR 플랫폼을 구현한 블록 다이어그램이다. 제안하는 플랫폼에서 RF 송수신기는 Ettus Research 사의 USRP X310을 사용했고, DSP는 TI의 TMS320C6670을 사용하였다. 이 때, 제안하는 SDR 플랫폼 블록 다이어그램에는 Relay PC가 RF 송수신기와 DSP 사이에 존재한다. Relay PC가 존재하는 이유는 USRP에서 제공하는 인터페이스와 DSP에서 제공하는 인터페이스가 상이하기 때문에 상대적으로 다양한 인터페이스를 지원하는 Relay PC를 USRP와 DSP 사이에 둬으로써 인터페이스의 상이함을 해결하기 위함이다[5]. 본 구현에서는 Relay PC와 DSP 간의 데이터 송/수신 프로토콜인 TCP/IP를 소프트웨어로 구현하고 USRP를 제어하기 위해 오픈 소스 소프트웨어 툴킷인 GNU Radio[8]를 사용하였다. GNU Radio는 RF 송수신기 제어를 위한 여러 가지 툴을 제공하는데 그 중 네트워킹 툴 내의 TCP 블록을 사용하여 Relay PC에서 DSP로의 데이터 전송 프로토콜을 구현하였다. 또한 DSP에서의 데이터 수신 프로토콜 구현을 위해 TI DSP에서 네트워킹에 관련된 전반적인 사항을 제공하는 NDK[9]를 사용하여 TCP/IP를 구현하였다. 이번 장에서는 <그림 1>의 제안하는 플랫폼에서 사용된 각 블록에 대해 자세히 설명한다.

## 2.1 USRP X310

USRP X series는 Ettus Research사에서 제작한 RF 송수신기로써 무선 통신 시스템을 설계하기 위한 고성능, SDR 플랫폼이다[10]. 본 구현에서는 Ettus Research사의 USRP X series 중 USRP X310을 사용했다. <그림 2>는 USRP X310의 블록 다이어그램이다. <그림 2>에서 보이는 것과 같이 X310은 마더 보드와 도터 보드로 구성되어 되어있다. X310은 2개의 도터 보드 슬롯 장착을 지원하여, 2X2 MIMO (Multi-Input and Multi-Output) 시스템 구현이 가능하며, 장착하는 도터 보드에 따라 채널 당 최대 160 MHz와 0 ~ 6 GHz 주파수 대역이 지원 가능하다[10, 11]. 본 구현에서는 120 MHz 대역폭과 400 MHz에서 4.4 GHz까지의 주파수 대역을 지원하는 SBX-120 도터 보드를 사용하였다[12]. <그림 2>의 USRP 블록 다이어그램을 보면, 데이터 수신 시 호스트 PC에서 사용자가 정한 인터페이스를 통해 USRP로 데이터가 들어오게 된다.



<그림 2> USRP X310 블록 다이어그램

이 때, GRC (GNU Radio Companion)에서는 USRP를 제어하기 위한 파라미터들을 설정해주고, UHD (USRP Hardware Driver)[13]에서는 GNU

Radio와 같은 소프트웨어가 USRP와 통신을 하기 위한 하드웨어 인터페이스를 제공한다. GRC와 UHD를 지나서 들어온 데이터는 모뎀 프로세싱 기능을 하는 FPGA를 거친 후, 소프트웨어로 설정한 A/D (Analog to Digital), D/A (Digital to Analog) 분해능 (resolution)과 샘플링 레이트에 의해 변환이 된 후, 설정된 중심 주파수와 이득 (gain)값에 따라 도터 보드의 믹서와 증폭기를 통과한다. 본 구현에서 모뎀 프로세싱은 DSP에서 하기 때문에, FPGA에서는 어떠한 연산도 하지 않고 A/D나 D/A 된 신호를 UHD와 GRC로 연결하여 수신하였다. 또한 X310에서는 1 GigE, 10 GigE, PCIe (Peripheral Component Interconnect Express) 등의 인터페이스들을 제공하고 사용하는 인터페이스에 따라 최대 200 MS/s (Sample per second)까지 지원하는데, 본 구현에서는 USRP와 Relay PC 간에는 10 GigE 인터페이스, Relay PC와 DSP간에는 1 GigE 인터페이스를 사용하였다. X310은 A/D, D/A의 클럭 레이트 (clock rate)인 MCR (Master Clock Rate)로 200 MHz와 184.32 MHz를 제공한다. 본 구현에서는 LTE의 샘플링 레이트를 사용해야 하므로 MCR이 샘플링 레이트의 짝수배가 되어야 한다는 하드웨어 사양에 따라 MCR을 184.32 MHz로 설정하였다[10]. MCR을 184.32 MHz로 설정함에 따라 LTE에서 지원하는 모든 샘플링 레이트의 구현이 가능해졌다.

## 2.2 GNU Radio

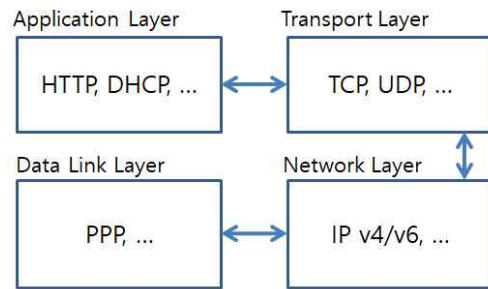
GNU Radio는 SDR 기술을 개발하기 위한 오픈 소스 소프트웨어 개발 툴킷으로 신호 처리와 관련된 애플리케이션을 제공한다. GNU Radio에서 제공하는 애플리케이션을 플로우 그래프 (flowgraph)라고 하는데, 이 플로우 그래프들을 사용하여 외부 RF 하드웨어를 이용하여 SDR시스템을 만들거나, 하드웨어 장비 없이 시뮬레이션을 가능하게 해주기 위해 GRC를

제공한다. GRC는 GNU Radio의 프론트엔드로써 GNU Radio 애플리케이션들을 개발하는데 사용되는 GUI (Graphical User Interface) 기반 소프트웨어이다. GRC를 통해 GNU Radio에서 파이썬이나 C++로 새로운 플로우 그래프를 만들거나 수정하여 컴파일을 하면, 새롭게 생성되거나 수정된 플로우 그래프를 GRC의 GUI 블록으로써 사용자에게 제공할 수 있다 [8]. 본 논문에서는 GUI 기반의 GRC를 사용하여 대역폭, 안테나 이득, 중심 주파수, 샘플링 레이트 등의 USRP를 제어하기 위한 파라미터를 GUI 기반의 GRC를 사용하여 송수신기를 제어했다. 본 구현에서는 GNU Radio v3.7.11 을 사용했다.

### 2.3 NDK

NDK는 TI DSP의 C6000 계열, Cortex-A8, ARM9 계열 등이 내장된 모든 하드웨어에서 동작이 가능하며, RTOS (Real-Time Operating System) 위에서 동작하는 네트워크 개발 키트이다. TI에서 제공하는 이 키트는 네트워크 기반 응용 프로그램의 개발과 시연을 가능하게 해준다[9]. <그림 3>은 NDK 시스템의 오버뷰 다이어그램이다. NDK는 총 4개의 계층에 대한 라이브러리를 제공한다. 애플리케이션 계층에서는 웹 서버 프로그래밍을 위한 HTTP (Hyper Text Transfer Protocol), IP 주소를 동적으로 할당하는 방식인 DHCP (Dynamic Host Configuration Protocol) 등을 제공한다. 전송 계층에서는 단순한 전송 방식에서 사용되는 UDP (User Datagram Protocol)와 연결 지향 전송 방식인 TCP 등을 지원한다. 네트워크 계층에서는 데이터를 보내기 위한 데이터 위주의 프로토콜인 IP v4와 v6를 제공한다. 마지막으로 데이터 링크 계층에서는 두 통신 노드간의 직접적인 연결을 위해 사용되는 PPP (Point to Point Protocol) 등을 제공한다[14]. <그림 3>에서 보이는 것과 같이 NDK는 TCP 뿐만 아니라 UDP, HTTP, DHCP 등 여러 계층에 대해 구현 가

능한 라이브러리를 제공한다. NDK에서 제공하는 각 계층에 대한 라이브러리는 사용자의 선택에 의해 활성화 또는 비활성화를 시켜 구성할 수 있다. 본 구현에서는 제공하는 라이브러리 중에서 TCP와 IP를 사용하였고, NDK는 v2.25를 사용하였다.

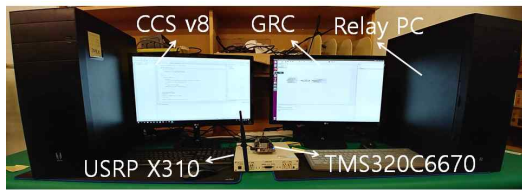


<그림 3> NDK 시스템 오버뷰 다이어그램

### III. 실험 결과

LTE의 구현 가능성을 확인하려면 3GPP (3rd Generation Partnership Project)에서의 MAC (Medium Access Control) 스케줄링 단위인 1ms에 한 개의 서브프레임이 수신 되어야 한다[15]. 3GPP에서의 LTE의 기준이 본 논문에서 사용한 1 Gige를 사용하여 만족하는지를 확인하기 위해 LTE에서 지원하는 대역폭에 따른 한 개의 서브프레임 평균 수신 시간과 처리량을 측정하였다. <그림 4>는 제안하는 SDR 플랫폼을 구현한 사진이다. 사진에서 보이는 것과 같이 RF 송수신기는 Ettus Research 사의 USRP X310을 사용하였고, 도터보드는 SBX-120을 사용하였다. Relay PC는 Intel i7 CPU (Central Processing Unit), 8GB RAM (Random Access Memory) 그리고 OS (Operating System)로는 Ubuntu 16.04 LTS (Long Term Support)를 사용하였다. 이 때, NDK에서 제공하는 데이터 송/수신을 위한 프로토콜이 UDP, TCP

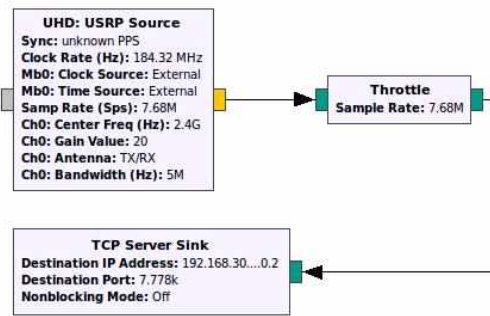
두 가지가 있었는데, Relay PC와 DSP 간의 데이터 송/수신 프로토콜은 데이터 일부 손실이 일어날 수도 있는 UDP 보다는 호스트 간의 신뢰성이 높은 TCP를 사용하였다. USRP의 제어와 DSP와의 TCP 통신을 위해 Relay PC에서 GRC를 사용하였고, NDK를 사용함으로써 DSP에서 TCP 통신을 통한 Relay PC로부터의 데이터 수신을 하였다. 또한 DSP 모델 코드의 모니터링과 디버깅을 위해서 CCS (Code Composer Studio) v8 을 사용했다.



<그림 4> LTE SDR 플랫폼 구현 사진

<그림 5>는 TCP/IP를 사용하여 데이터 송/수신 흐름을 소프트웨어인 GRC로 구성한 블록 다이어그램이다. "UHD:USRP Source" 블록을 통해 대역폭에 따른 파라미터들을 변경하면서 USRP를 제어하였으며, "Throttle" 블록으로 USRP Source 블록에서 설정해준 샘플링 레이트를 한 번 더 설정해줌으로써 안정성을 높였다. 또한 "TCP Server Sink" 블록으로 DSP와 연결될 목적지의 IP 주소와 포트를 설정하여 구현하였다. <표 1>은 LTE SDR 플랫폼에서 사용된 중심 주파수, 안테나 이득, ADC 분해능 (resolution), MCR 파라미터에 대한 설정을 보여주며, <표 2>는 NDK와 GNU Radio를 사용하여 테스트한 대역폭에 따른 처리량을 보여준다. <표 2>의 평균 시간은 하나의 서브프레임을 100번 수신하여 평균을 낸 측정 시간이고, 처리량은 샘플링 레이트에 하나의 샘플 당 32 비트인 ADC 분해능 (resolution)을 곱한 후, 평균 시간으로 나뉘서 계산한 값이다. <표 2>의 결과를 보았을 때, 처리량이 1 GigE의 데이터 레이트인 1 Gbps 내에 들

어오며 LTE의 실시간 처리 제약 시간인 1ms보다 평균 시간이 적은 것을 확인할 수 있었다. 결과적으로, GNU Radio와 NDK를 사용하고 1 GigE 인터페이스를 사용하여 LTE SDR 플랫폼의 구현 가능성을 확인했다.



<그림 5> GRC 블록 다이어그램

<표 1> 테스트를 위한 파라미터 설정

Parameter	Value
Center Frequency (GHz)	2.4
Antenna Gain (dB)	20
ADC resolution	I: 16 bits, Q : 16bits
MCR (MHz)	184.32

<표 2> 테스트를 위한 파라미터 설정

Band Width (MHz)	Sampling rate (MSps)	평균 시간 (ms)	처리량 (Mbps)
1.4	1.92	0.9802	62.681
3	3.84	0.9784	125.592
5	7.68	0.9821	250.239
10	15.36	0.9005	545.830

#### IV. 결론

본 논문은 LTE SDR 플랫폼에서 GNU Radio와 TI DSP의 NDK에서 제공하는 TCP/IP를 사용하여 15MHz와 20MHz를 제외한 나머지 LTE 대역폭에 대

한 처리량을 보여주었다. 실험 결과를 통해 1 GigE 인터페이스에서 LTE SDR 플랫폼 구현 가능성을 확인할 수 있었다. 또한 SDR 플랫폼인 USRP와 이를 제어하는 GNU Radio를 통해 하드웨어의 변경 없이 소프트웨어에서의 변경만을 통해 구현이 가능하다는 것도 보여주었다. 본 플랫폼은 15MHz와 20MHz를 제외한 나머지 대역폭에 대한 구현 결과만을 제시했지만, 데이터 전송 인터페이스만 바뀌준다면 MIMO로 인하여 레이어가 증가하는 상황이나 CA (Carrier Aggregation)에 의해 대역폭이 늘어나는 상황도 구현이 가능할 것이다.

## Acknowledgment

본 연구는 과학기술정보통신부 및 과학기술인재진흥원 2018년 산학연공동연구법인지원사업의 지원을 받아 수행하였음. (2018K000150)

## 참고문헌

- [1] 황경호 · 조동호, "Software Defined Radio 기술," Telecommunications Review, 제10권, 제1호, 2000, pp. 130-143.
- [2] 최진성, "차세대 단말기를 위한 SDR 기반의 통신 모뎀 구조 TREND," TTA저널, 제102호, 2005, pp. 135-141.
- [3] 박종근 · 최승원, "GPU를 이용한 TDD LTE MU-MIMO 시스템에서의 재전송 구현," 디지털산업정보학회 논문지, 제13권, 제2호, 2017, pp. 35-42.
- [4] 광경훈 · 신봉득 · 박동욱 · 어윤성 · 오혁준, "USRP RIO SDR을 이용한 5G 밀리미터파 LTE-TDD HD 비디오 스트리밍 시스템 설계 및 구현," 한국전자과학회논문지, 제27권, 제5호, 2016, pp. 445-453.
- [5] Jaeho Lee, Heungseop Ahn, Daejin Kim, Seungwon Choi, Ildo Jung, Jehoon Oh, "Implementation of an SDR-based LTE Base-station System with a Novel Interface between USRP and Multicore DSP," ICCE-Asia 2018, 2018.06.
- [6] Texas Instruments, <http://www.ti.com/ko-kr/processors/digital-signal-processors/overview.html>
- [7] Texas Instruments, <http://www.ti.com/product/tms320c6670>
- [8] Wikipedia, [https://en.wikipedia.org/wiki/GNU\\_Radio](https://en.wikipedia.org/wiki/GNU_Radio)
- [9] Texas Instruments, "TI Network Developer's Kit (NDK) v2.25 User's Guide", Jan. 2011.
- [10] Ettus Research, <https://kb.ettus.com/X300/X310>
- [11] Ettus Research, [http://files.ettus.com/manual/page\\_usrp\\_x3x0.html](http://files.ettus.com/manual/page_usrp_x3x0.html)
- [12] Ettus Research, <https://www.ettus.com/product/details/SBX120>
- [13] Ettus Research, <https://kb.ettus.com/UHD>
- [14] Wikipedia, [https://en.wikipedia.org/wiki/OSI\\_model](https://en.wikipedia.org/wiki/OSI_model)
- [15] 3GPP TR 36.881, "Study on latency reduction techniques for LTE"
- [16] 이승학 · 김경훈 · 안치영 · 최승원, "GPU를 이용한 SDR 시스템 용 LTE MIMO 기지국 기능 구현," 디지털산업정보학회 논문지, 제8권, 제4호, 2012, pp. 91-98.

■ 저자소개 ■



진 화 중  
(Jin Hwajong)

2018년 3월~현재  
한양대학교 전자컴퓨터통신공학과  
석사과정  
2018년 2월 남서울대학교 정보통신공학과  
(공학학사)  
관심분야 : DSP, vehicular comm, LTE-A,  
etc  
E-mail : hwajong.jin@dsplab.hanyang.  
ac.kr



김 대 진  
(Kim Daejin)

2017년 3월~현재  
한양대학교 전자컴퓨터통신공학과  
박사과정  
2017년 2월 한양대학교 전자컴퓨터통신공학과  
(공학석사)  
2015년 2월 숭실대학교 정보통신전자공학부  
(공학학사)  
관심분야 : vehicular comm, 5G, LTE, Cell  
planning, SDR etc.  
E-mail : kdj317@dsplab.hanyang.ac.kr



최 승 원  
(Choi Seungwon)

2012년 3월~현재  
HY-MC 연구센터 센터장  
2002년~2011년  
HY-SDR 연구센터 센터장  
1992년~현재  
한양대학교 전자전기공학부 교수  
1990년~1992년  
일본 우정성 통신연구소 선임연구원  
1989년~1990년  
ETRI 선임연구원  
1988년~1989년  
미국 Syracuse대학 전지 및  
전산과 교수  
1988년 12월 미국 Syracuse대학 전기공학  
(공학박사)  
1985년 12월 미국 Syracuse대학 전기공학  
(공학석사)  
1982년 2월 서울대학교 전자공학 (공학석사)  
1982년 2월 한양대학교 전자공학 (공학학사)  
관심분야 : SDR, 이동통신, 신호처리  
E-mail : choi@dsplab.hanyang.ac.kr

논문접수일 : 2018년 11월 9일  
수 정 일 : 2018년 12월 3일  
계 재 확정일 : 2018년 12월 4일