

LOM202A 사용자 매뉴얼

V9.2

성지산업

January 04, 2021

목차

1. Document Revision History.....	3
2. Precautions.....	4
3. Join-Request 운용 시 유의 사항.....	5
4. 하드웨어.....	7
5. 디버그 메시지(Log) 구문 분석 예제.....	8
5-1. Debug Message(Log) 사용하기.....	8
5-2. Debug Message(Log) On 의 경우.....	8
5-3. Debug Message(Log) Off 의 경우.....	8
5-4. SKT KR Public 전용의 Debug Message 의 경우.....	9
부록 A. 그 밖의 특징들.....	11
■ Join Retransmission Back-off.....	11
■ LBT AFA.....	11
■ Application 프로세서 진행 시 CLI Command 운용.....	11
부록 B. 디버그 메시지(Log) 상세 구문.....	12
■ Debug Message(Log) On 의 경우.....	12
■ Debug Message(Log) Off 의 경우.....	18
■ SKT KR Public 전용의 Debug Message 의 경우.....	19
부록 C. Other References.....	23
■ UART 사용하기.....	23
■ CLI Command 인터페이스.....	23
■ 펌웨어 업그레이드.....	23
■ 주기보고 데이터의 형식.....	오류! 책갈피가 정의되어 있지 않습니다.

1. Document Revision History

Date	Version	Updated
2017-01-13	1.0	First release
2017-05-02	2.0	[추가] 'Document revision history' 항목 추가 [추가] '적용 모델 및 펌웨어 파일명' 항목 추가 [수정] Set Report time 커맨드에서 최대 설정 값을 30일(2,592,000초)로 변경 [수정] 테스트 프로그램에서 Link Check Request 커맨드에 대한 설명 변경 [추가] 테스트 프로그램에서 Time Sync Request 커맨드에 대한 실행 커맨드 및 설명 추가
2017-08-23	3.0	[수정] 본문 전면 수정
2017-09-09	4.0	[추가] 적용 모델 및 펌웨어 파일명 추가
2017-10-10	5.0	[삭제] 펌웨어 파일명 삭제 [추가] 적용 모델 추가
2018-02-07	6.0	[삭제] Tx Power와 Atten(Antenna gain)관련 목차 및 설명 삭제
2018-02-20	7.0	[수정] 디버그 메시지(Log) 구문 분석 예제에서 Debug Message(Log) On과 OFF 각각의 경우 chapter에서 송신 이후에 서버로부터 ACK 수신을 성공할 때의 parsing 메시지 수정 -. Step 2-1 : Ack received 준비 완료 상태의 parsing 메시지 수정 -. Step 3-1 : READY [추가] READY 디버그메시지 parsing시 주의 사항 추가 [추가] ABP(Activation By Personalization) 모드에서 Join완료를 대체하는 디버그 메시지 추가
2019-02-18	8.0	[삭제] '적용 모델' 챕터 삭제 [삭제] 'Recommended Reading' 챕터 삭제 [수정] 목차 형식 수정 [수정] Document Revision History 형식 수정 [추가] 'Precautions' 챕터 추가 [추가] 'Join-Request 운용 시 유의사항' 챕터 추가 [추가] '부록 C. Other References' 챕터 추가
2019-09-05	9.0	[수정] Document Revision History 형식 수정
2019-12-01	9.1	[수정] 2-1 예외처리 기능 구현 [수정] 2-2 Join 완료 이후 CLI command 실행 시점 준수 [수정] 2-4 LoRa모듈의 "READY" 및 "BUSY"상태인지 [수정] 3 Join request 운용 시 주의사항 [수정] 5-3 Debug Message(log) OFF/None 의 경우
2021-01-04	9.2	[수정] 2-1 예외처리 기능 구현 [수정] 4 모듈 핀 설명 [추가] Debug Message(Log) None

2. Precautions

유저는 반드시 아래의 유의사항을 숙지하고 구현하여야 한다.

2-1. 예외처리 기능 구현

LoRa 모듈이 비정상적인 동작이 발생되어 프로세스를 운용할 수 없는 경우가 발생 될 때, LoRa 모듈을 hardware system reset을 통해 복구할 수 있는 예외처리 기능을 반드시 포함해야 한다.

추가적으로 software system reset 명령은 "LRW 70" 이 있습니다.

(망 사업자의 규정에 따라 이 기능은 제한 될 수 있으며, 이는 망 사업자의 규정을 우선한다.)

비정상 동작 예시

- ① CLI Command 실행 시 'OK', 'ERROR', 'BUSY' 중 하나가 리턴 되지 않는 경우 (CLI Command가 실행되지 않는 경우)
- ② CLI Command 실행 시 'BUSY'가 리턴 되었지만 송신 또는 수신 진행 상태가 아니며 그 시점을 기준으로 일정시간(2분 이상) 경과 한 이후에도 CLI Command 재실행 시 여전히 'BUSY'가 리턴 되는 경우
- ③ 전원 On후 Join을 시도 하지만 Join이 되지 않아 계속 Join을 하는 경우 12시간 이상 Join이 안될 경우
- ④ (Un)confirmed up을 실행 후 1분 10초 이내에 "READY" 가 출력되지 않는 경우

2-2. Join 완료 이후 CLI Command 실행 시점 준수

유저는 LoRa 모듈의 안정적인 프로세스 운용을 위하여 반드시 Join 완료를 나타내는 '**Join is completed**' 디버그 메시지 출력 완료 이후 **2.5초** 이상 경과 후에 CLI Command를 실행해야 한다.
[Ver 1.29 이하 버전]

Ver 2.0 이후 버전은 "Join is completed" 디버그 메시지 출력 완료 이후 0.5초 이상 경과 후에 CLI command를 실행해야 한다.

2-3. LoRa 모듈의 bootin messe상태 인지

LoRa 모듈의 시스템 리셋 직후 UART를 통해 출력되는 디버그 메시지를 인지하여 LoRa 모듈이 시스템 리셋이 된 상태를 유저 LoRa 디바이스의 메인 프로세스에서 알고 있어야 한다. LoRa 모듈의 시스템 리셋 직후에 출력되는 디버그 메시지는 '**Wisol LoRa**' 이다.

2-4. LoRa 모듈의 'READY' 및 'BUSY'상태 인지

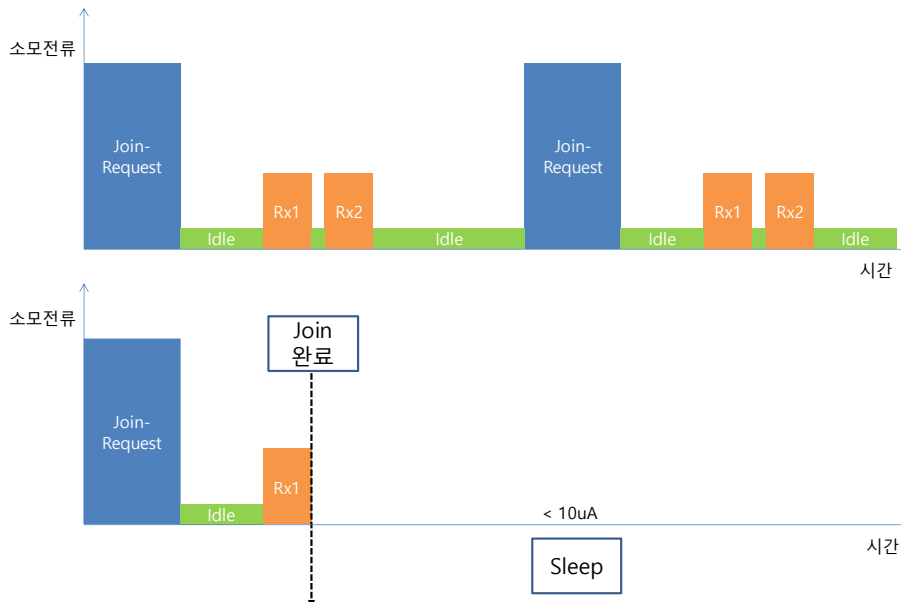
LoRa 모듈은 송신 커맨드 실행 시 처리 완료 디버그메시지로 'READY'를 출력한다. 'READY'가 출력되기 전 송신 또는 수신 진행상태에서 송신 등의 커맨드를 실행하면 디버그메시지로 'BUSY'가 출력되며 그 커맨드는 drop된다. 따라서 'READY'를 확인 한 후에 다음 송신 커맨드를 실행해야 한다. 또한 예외적인 상황이 발생되어 송신커맨드 실행 시 송신 또는 수신 진행 중이 아닌 경우

에도 'BUSY'가 발생 될 경우, 예외처리에 의해 'BUSY' 발생 직후 40초 경과 후에 다시 커맨드를 실행하면 정상 동작 된다. 이와 같이 유의하였는데도 지속적으로 'BUSY'가 발생 할 경우를 대비하여 LoRa 모듈을 hardware system reset을 통해 복구할 수 있는 예외처리 기능을 반드시 포함해야 한다. 또는 Software system reset 명령인 "LRW 70"을 이용하여 Reset 할 수 있다.

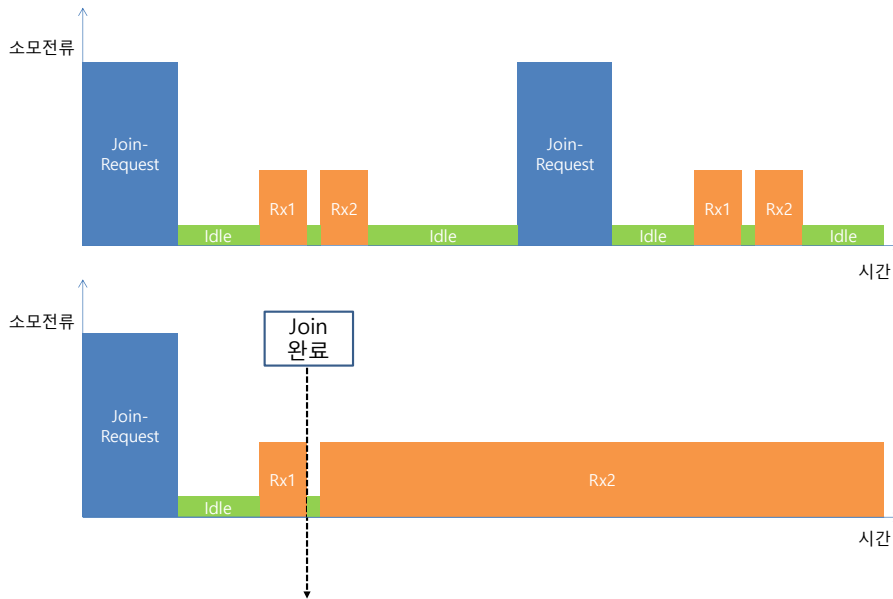
3. Join-Request 운용 시 유의 사항

LoRa Module은 OTAA(Over The Air Activation) 모드에서 전원인가 또는 System reset 시 Join-Request가 시작되며 Class A에서는 Join이 완료되기 전까지는 Sleep mode에 진입하지 않는다. Class C에서는 Join이 완료되더라도 Sleep mode에 진입하지 않고 Rx2 수신 대기 상태가 된다. Class A와 Class C 모두 LoRaWAN 규격에 의해 Join-Request의 재전송은 Join이 완료될 때까지 무한 반복된다. Join-Request의 재전송 간격은 **부록A**에 있는 **Join Retransmission Back-off**를 참고하면 된다.[LOM202A00 Ver 1..29 이하 버전]

-. Class A에서의 Join-Request 동작 방식



-. Class C에서의 Join-Request 동작 방식



- Join-Request 재전송 주기 프로세스

Join-Request 일정 주기로 반복 송신 -> 일정시간 경과 후 재전송 주기시간 증가 -> Join-Request 일정 주기로 반복 송신 -> 일정시간 경과 후 재전송 주기시간 증가 -> Join-Request 일정 주기로 반복 송신 -> 일정시간 경과 후 더 이상 주기시간 증가 안 함 -> Join-Request 일정 주기로 반복 송신(Join이 완료될 때까지 무한반복)

- Class A에서 Join-Request 재전송을 중단하고 Shut down 커Sleep mode로 진입하는 방법

Join-Request 진행 중 -> **LRW 60 CLI 커맨드 입력** -> Join-Request 중단 및 sleep mode 상태 유지 -> Join을 다시 시작 할 경우 System Reset -> Join-Request 시작

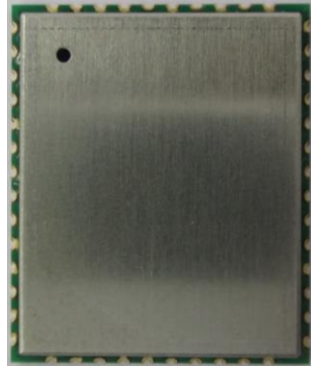
***Class C에서는 LRW 60 CLI 커맨드가 동작되지 않는다. [Ver 1.29이하 버전]**

*** Ver 2.0이후 버전에서는 전원 On 후에 1초 이내에 입력시 Sleep mode에 진입한다.**

- 이 모드는 전류 측정을 위한 Test 모드이므로 Join이후 사용을 하지 마십시오.

4. 하드웨어

LOM202A



< TOP >

Figure 1-1: Picture of LOM202A

모듈 핀 설명

Pin No.	Pin name	MCU pin name		Description
2	PB7	PB7	GPIO IN	Boot Loader Enable(High Active)
3	USART5_TX	PB3	UART5 TX	UART TX
4	USART5_RX	PB4	UART5 RX	UART RX
6	USART1_TX	PA9	UART1 TX	UART TX for Firmware download
7	USART1_RX	PA10	UART1 RX	UART RX for Firmware download
8	SWCLK	PA14	SWCLK	SWCLK
9	SWDIO	PA13	SWDIO	SWDIO
11	SPI2_MOSI	PB15	GPIO OUT	TX/RX : 1(during 300ms), Other : 0
14	SPI2_NSS	PB12	GPIO IN	Payload data bit 3(TBD)
20	PB10	PB10	GPIO IN	Payload data bit 0(TBD)
21	PB11	PB11	GPIO IN	Payload data bit 1(TBD)
23	PA3	PA3	ADC IN	Payload data bit 4 ~ 15(TBD)
26	NRST	NRST	RST	Reset
29	PC2	PC2	ADC IN	Battery Level 12bit (for uplink port 3)
30	WKUP1	PA0_WKUP1	INT IN	Wake Up: Rising Edge, Payload data bit 2(TBD)

5. 디버그 메시지(Log) 구문 분석 예제

5-1. Debug Message(Log) 사용하기

아래 4단계의 흐름을 통해 debug message(Log)를 사용하는 예제를 제시한다.

- Power-up or Reset
- Join-Request / Join-Accept
- Payload data Tx / Rx
- Ready status

5-2. Debug Message(Log) On의 경우

서버접속을 성공할 때의 parsing 메시지.

전원 인가 또는 시스템 리셋 시 서버접속을 위한 Join-Request을 자동으로 송신한다. Join-Accept을 수신하기 전까지 지속적으로 송신한다. 최종 서버접속에 성공한 'Join is completed' 메시지를 parsing 한다.

- Step 1 : Join is completed

송신 이후에 서버로부터 ACK 수신을 성공할 때의 parsing 메시지.

- Step 2-1 : Ack received

송신 이후에 서버로부터 ACK 수신을 실패할 때의 parsing 메시지.

- Step 2-2 : Ack not received

준비 완료 상태의 parsing 메시지

- Step 3-1 : READY

준비 상태 확인을 위한 parsing 메시지

송신 커맨드 실행 시 'BUSY' 메시지가 return되면 준비가 되어있지 않은 상태이다.

- Step 3-2 : BUSY

5-3. Debug Message(Log) Off/None의 경우

서버접속을 성공할 때의 parsing 메시지.

전원 인가 또는 시스템 리셋 시 서버접속을 위한 Join-Request을 자동으로 송신한다. Join-Accept을 수신하기 전까지 지속적으로 송신한다. 최종 서버접속에 성공한 'JOIN_COMPLETE' 메시지를 parsing 한다.

- Step 1 : JOIN_COMPLETE

송신 이후에 서버로부터 ACK 수신을 성공할 때의 parsing 메시지.

- Step 2-1 : **Ack received**

송신 이후에 서버로부터 ACK 수신을 실패할 때의 parsing 메시지.

- Step 2-2 : **Ack not received**

준비 완료 상태의 parsing 메시지

- Step 3-1 : **READY**

준비 상태 확인을 위한 parsing 메시지

송신 커맨드 실행 시 'BUSY' 메시지가 return되면 준비가 되어있지 않은 상태이다.

- Step 3-2 : **BUSY**

※ **READY 디버그메시지 parsing시 주의 사항**

Join완료 메시지인 주기보고 on상태의 'Join is completed' 또는 주기보고 off상태의 'JOIN_COMPLETE' 디버그메시지가 출력된 이후부터 'READY' 디버그 메시지가 유효함. 따라서 Join 이 완료되지 않은 상태에서는 'READY' 디버그 메시지를 parsing하지 말 것

ABP(Activation By Personalization) 모드에서 Join완료를 대체하는 디버그 메시지

ABP 모드에서 송신 이후 Ack수신에 성공 할 경우 Ack수신 시 최초 1회만 'ABP Connection OK' 디버그 메시지가 출력된다.

5-4. SKT KR Public 전용의 Debug Message의 경우

1. Debug Message(Log) On의 경우

1) ThingPlug 제어명령 DevReset 수신 시의 parsing message

DevReset

2) ThingPlug 제어명령 RepPerChange 수신 시의 parsing message

RepPerChange : <interval time>

3) ThingPlug 제어명령 ReplImmediate 수신 시의 parsing message

ReplImmediate

4) ThingPlug 제어명령 extDevMgmt 수신 시의 parsing message

Rx_MSG 222 : 0000170123456789012345678901234567890123456789012345

extDevMgmt

Rx_MSG <Fport> : <LoRaMAC version> <message type> <Payload length> <Payload>

1byte

1byte

1byte

1byte 이상

2. Debug Message(Log) Off의 경우

1) ThingPlug 제어명령 DevReset 수신 시의 parsing message

DEVRESET

2) ThingPlug 제어명령 RepPerChange 수신 시의 parsing message

INTERVAL : <interval time>

3) ThingPlug 제어명령 ReplImmediate 수신 시의 parsing message

REPORT

4) ThingPlug 제어명령 extDevMgmt 수신 시의 parsing message

Rx_MSG_FP222_LEN17 : 012345678901234567890123456789012345

EXTDEVMGMT

RX_MSG_FP<Fport number>_LEN<Length> : <Payload>

부록 A. 그 밖의 특징들

■ Join Retransmission Back-off

Join Request 송신에 대해 서버로부터 Join Accept를 수신 받지 못하면 Join Request 송신에 대한 재전송을 진행한다. 이 때 LoRaWAN 규격에 의해 back-off Time 기능이 적용되어 duty cycle에 의해 아래와 같은 송신점유시간 제한으로 시간 간격으로 재전송을 한다.

Aggregated during the first hour following power-up or reset	$T0 < t < T0+1$	Transmit time < 36 Sec
Aggregated during the next 10 hours	$T0+1 < t < T0+11$	Transmit time < 36 Sec
After the first 11 hours, aggregated over 24h	$T0+11+N < t < T0+35+N$ $N \geq 0$	Transmit time < 8.7 Sec per 24h

Join-Request 송신관련 디버그메시지

10초 간격으로 Join-Request 송신관련 디버그메시지를 출력하지만, 실제 송신이 되는 것이 아니며, 아래의 디버그 메시지가 출력되면서 실제 Join-Request가 송신 된다.

SEND : Freq: <frequency> DR: <dataRate> PWR: <Tx power index>

■ LBT AFA

LBT AFA란 Listen Before Talk Adaptive Frequency Agility의 약자 이다.

송신 전 신호감지(Listen Before Transmission)방식으로 송신 전 5 ms 이상 수신하여 그 수신 신호의 세기가 -65 dBm 이하인 경우에 한하여 전파를 발사하고, 4 초 이내에 송신을 중단하여 50 ms 이상 휴지한다. 일본버전의 경우는 수신 신호의 세기가 -80 dBm 이하로 설정된다.

50ms이상 휴지 후 다시 신호감지를 하여 수신 신호 세기가 -65 dBm을 초과한다면 또 다시 50ms이상 휴지한다. 이를 지속 반복하여 누적 휴지시간이 27,000 ms를 초과하면 현재 채널을 제외한 나머지 사용 가능한 채널 중에서 랜덤으로 선택되어 채널이 변경된다. 채널 변경 직후에도 신호감지를 한다.

■ Application 프로세서 진행 시 CLI Command 운용

Application 프로세서 진행 시 CLI Command 동작 방식

모듈이 프로세서 진행을 완료하지 않은 상태에서 Application에서 커맨드가 오면 커맨드를 실행하지 않고 Application에게 'BUSY' 실패 코드를 전달한다.

수신 완료 후 CLI Command 입력 시간

서버로부터 Unconfirmed-down 등의 데이터 수신 완료 후 CLI Command 입력 시 100ms 이상의 시간 지연 후 CLI Command 실행이 필요하다.

부록 B. 디버그 메시지(Log) 상세 구문

GUI전용 프로그램 Log창에서의 Time stamp Debug message

GUI전용 프로그램 Log창에는 debug message 중간에 '[2016-01-01 12:00:00 000]:'과 같은 형태의 Time stamp가 출력되는데 이는 GUI 프로그램상에서만 출력되며 실제 UART에는 Time stamp가 출력되지 않는다.

CLI Command Debug message

CLI Command debug message는 CLI Command Interface 문서에 상세히 기술되어 있으므로 이를 참고 하면 된다. (CLI Command Interface 문서에서 실행 Message 부분을 참고 할 것)

■ Debug Message(Log) On의 경우

Reset 또는 전원 인가 시 Debug message (Jon Request / Accept)

모듈에 대해 reset 또는 전원 인가 시 pseudo join 또는 real join을 시도한다.

처음에는 항상 pseudo join 부터 시작하며 pseudo join 절차에서 서버로부터 real key를 받아 자동 저장되면, real join으로 변경된다. 따라서 그 다음부터는 시스템 reset을 해도 항상 real join 부터 시작한다.

또한 pseudo join 및 real join 절차를 거쳐 최종적으로 join이 완료되어도 그 다음부터는 시스템 reset을 해도 항상 real join부터 시작한다.

1. CLASS A 모드의 pseudo Join request과 real join request의 debug message

1) pseudo join debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0

SEND : Freq: 922100000 DR: 0 PWR: 1

1st Join

RX1CH Open: freq 922100000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 0

2) real join debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0

SEND : Freq: 922100000 DR: 0 PWR: 1

2nd Join

RX1CH Open: freq 922100000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 0

2. CLASS C 모드의 pseudo Join request과 real join request의 debug message**1) pseudo join debug message**

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0

SEND : Freq: 922300000 DR: 0 PWR: 1

1st Join

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

2) real join debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0

SEND : Freq: 922300000 DR: 0 PWR: 1

2nd Join

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

3. CLASS A 모드 / 주기보고 OFF 모드의 join 완료에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 2 P 0

SEND : Freq: 922300000 DR: 0 PWR: 1

2nd Join

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_JOIN_ACCEPT / Ver: 0

OnRadioRxDone: ADR OFF, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 0 P 32

Join is completed

4. CLASS A 모드 / 주기보고 ON 모드의 join 완료에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0

PrepareFrame: ADR ON, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0

SEND : Freq: 922300000 DR: 0 PWR: 1

2nd Join

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_JOIN_ACCEPT / Ver: 0

OnRadioRxDone: ADR OFF, ADDRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 0 P 32

PrepareFrame: MType: FRAME_TYPE_DATA_CONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 1
 SEND : Freq: 922500000 DR: 0 PWR: 1
 DataReport : 0x3360
 RX1CH Open: freq 922500000, SF 12, BW 0, TO 25, RXC 0
 OnRadioRxDone: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_DOWN** / Ver: 0
 OnRadioRxDone: ADR ON, ADRAckReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 1 P 192

Join is completed

5. CLASS C 모드 / 주기보고 OFF 모드의 join 완료에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0
 PrepareFrame: ADR ON, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0
 SEND : Freq: 922300000 DR: 0 PWR: 1
 2nd Join
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
 RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0
 OnRadioRxDone: MType: FRAME_TYPE_ **JOIN_ACCEPT** / Ver: 0
 OnRadioRxDone: ADR OFF, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 0 P 32

Join is completed

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

6. CLASS C 모드 / 주기보고 ON 모드의 join 완료에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_JOIN_REQ / Ver: 0
 PrepareFrame: ADR ON, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 0
 SEND : Freq: 922500000 DR: 0 PWR: 1
 2nd Join
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
 RX1CH Open: freq 922500000, SF 12, BW 0, TO 25, RXC 0
 OnRadioRxDone: MType: FRAME_TYPE_ **JOIN_ACCEPT** / Ver: 0
 OnRadioRxDone: ADR OFF, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 0 P 32
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
 PrepareFrame: MType: FRAME_TYPE_DATA_ **CONFIRMED_UP** / Ver: 0
 PrepareFrame: ADR ON, ADRAckReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 1
 SEND : Freq: 923100000 DR: 0 PWR: 1
 DataReport : 0x3140
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 923100000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_DOWN** / Ver: 0

OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 1 P 111

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

Join is completed

7. debug message에 대한 설명

- FRAME_TYPE_JOIN_REQ : 송신 메시지 형태로 Join request임을 나타낸다
- FRAME_TYPE_JOIN_ACCEPT : Join request에 대한 수신 ack로 Join accept임을 나타낸다.
- FCnt : 프레임 카운트를 의미하며, 송신할 때마다 1씩 증가한다.
단, 재전송(retransmission) 시에는 프레임 카운트가 증가하지 않는다.
- P : Fport 번호를 나타낸다. Join request의 Fport 번호는 0이다.
할당된 Fport 번호는 아래와 같다.
 - ① Fport 0 : Join request, mac command(서버트리거)에 대한 Request-Answer 송수신에 사용 된다.
 - ② Fport 1 ~ 221 : 유저가 송신(Tx)에 사용하는 Fport번호로 범위내에서 랜덤하게 사용 가능하다. 송신에 대해서 서버로부터 수신 받는 ack 형태의 데이터는 Fport번호가 랜덤하게 부여된다.
 - ③ Fport 222 : ThingPlug 서버에 할당되어 있다. 따라서 ThingPlug 제어명령 수신 데이터의 Fport번호는 222번이다.
 - ④ Fport 223 : 네트워크 서버에 할당되어 있다. 따라서 서버 제어명령 수신 데이터의 Fport번호는 223번이다.
- join request '*1st Join*'은 첫 번째 Join을 의미하며 이는 pseudo join을 시도하는 debug message이다.
- SEND : 데이터 송신을 의미 한다.
- Freq : 송신 Frequency(주파수)를 의미한다. 단위는 Hz이다.
- DR : Data rate을 의미한다.
- PWR : Tx power index를 의미한다.
0 : 14dBm, 1 : 13dBm, 2 : 12dBm, 3 : 11dBm, 4 : 10dBm, 5 : 9dBm
6 : 8dBm, 7 : 7dBm, 8 : 6dBm, 9 : 5dBm, 10 : 4dBm, 11 : 3dBm
12 : 2dBm, 13 : 1dBm, 14 : 0dBm
- *1st Join* : First Join(첫 번째 Join)을 의미하며 pseudo Join request에 해당한다.
- *2nd Join* : Second Join(두 번째 Join)을 의미하며 real Join request에 해당한다.
- RX1CH Open : RX1 Channel로 수신 대기 상태를 의미 한다.
- RX2CH Open : RX2 Channel로 수신 대기 상태를 의미 한다.
- freq : 수신 Frequency(주파수)를 의미한다. 단위는 Hz이다.

- SF : SpreadingFactor를 의미한다. (data rate과 매칭된다)
- Join is completed : 최종적으로 join이 완료되었음을 의미 한다.

송수신(TX/RX) Debug message

송수신 데이터는 Confirmed와 Unconfirmed 두 가지 형태가 있다.

Confirmed는 ack를 요청하는 데이터이며 Unconfirmed는 ack를 요청하지 않는 데이터이다.

송신은 Confirmed-up과 Unconfirmed-up의 Uplink이다.

수신은 Confirmed-down과 Unconfirmed-down의 Downlink이다.

Confirmed-up 형태로 송신을 하면 서버로부터 반드시 ack를 받는다.

ack는 Confirmed-down이나 Unconfirmed-down으로 받을 수 있는데 일반적으로 Confirmed-up에 대해서 Unconfirmed-down으로 ack를 받는다. 이는 ack에 다시 ack를 요청하는 데이터를 요청하게 되면 또 다시 서버로 송신을 해야 하고 이는 무한 루프에 빠지게 된다.

이와 같은 현상을 방지하기 위해 아래와 같은 방식으로 운용된다.

Confirmed-up -> Unconfirmed-down

Confirmed-up -> Confirmed-down -> Unconfirmed-up

Unconfirmed-up -> ack 없음

단, Unconfirmed-up일지라도 송신 후 RX 수신모드로 전환되므로 서버로부터 데이터를 수신 할 수도 있다.

1. CLASS A 모드의 Confirmed-up과 Unconfirmed-down에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_ **CONFIRMED_UP** / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 1

SEND : Freq: 922300000 DR: 0 PWR: 1

DataReport : a1b2c3

OK

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_DOWN** / Ver: 0

OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 1 P 134

2. CLASS A 모드의 Unconfirmed-up에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_UP** / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 2 P 1
 SEND : Freq: 922100000 DR: 0 PWR: 1
 DataReport : a1b2c3

OK

RX1CH Open: freq 922100000, SF 12, BW 0, TO 25, RXC 0
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 0

3. CLASS C 모드의 Confirmed-up과 Unconfirmed-down에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_ **CONFIRMED_UP** / Ver: 0
 PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 1 P 1
 SEND : Freq: 922100000 DR: 0 PWR: 1
 DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
 RX1CH Open: freq 922100000, SF 12, BW 0, TO 25, RXC 0
 OnRadioRxDone: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_DOWN** / Ver: 0
 OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 1 P 210

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

4. CLASS C 모드의 Unconfirmed-up에 대한 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_ **UNCONFIRMED_UP** / Ver: 0
 PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 2 P 1
 SEND : Freq: 923300000 DR: 0 PWR: 1
 DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
 RX1CH Open: freq 923300000, SF 12, BW 0, TO 25, RXC 0
 RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

5. debug message에 대한 설명

- FRAME_TYPE_DATA_CONFIRMED_UP : 송신 메시지 형태로 Confirmed-up임을 나타낸다
- FRAME_TYPE_DATA_UNCONFIRMED_UP: 송신 메시지 형태로 Unconfirmed-up임을 나타낸다
- FRAME_TYPE_DATA_UNCONFIRMED_DOWN: 수신 메시지 형태로 Unconfirmed-down임을 나타낸다

나타낸다

- DataReport : DataReport 콜론(:) 다음에 출력되는 데이터는 송신 메시지이다.

■ Debug Message(Log) Off의 경우

Debug message on/off CLI Command(LRW 64)를 이용하여 Debug message off로 설정 할 경우, 별도의 UART protocol용 debug message가 출력된다. 상황 별 출력 메시지는 아래와 같다.

1st item	2nd item	3rd item	Protocol	Note
boot	boot information	펌웨어 버전	FW_VER : <firmware version>	
		Activation : Over the air activation	OTAA	
		Activation : activation by personalization	ABP	
		Enhanced Provisioning Enable	PROV_ENABLE	
		Enhanced Provisioning Disable	PROV_DISABLE	
		Enhanced Provisioning state None	PROV_STATE_NONE	
		Enhanced Provisioning state Done	PROV_STATE_DONE	
	class mode	Class A	CLASS_A	
		Class C	CLASS_C	
	Uplink Cycle	Uplink Cycle ON	CYCLE_ON	
		Uplink Cycle OFF	CYCLE_OFF	
Join	Before Join	First Join-request alive (pseudo Join)	JOIN_REQ_1ST	
		Second Join-request alive (real Join)	JOIN_REQ_2ND	
		Join-request send	SEND	
		Join-accept 수신	JOIN_ACCEPT	
	After Join	Join 완료	JOIN_COMPLETE	
Tx	SEND	Confirmed	CON_UP	
		Unconfirmed	UNCON_UP	
		message send	SEND	
	Retransmission	Confirmed Retransmission 완료	CON_UP_RETX_OK	
		Unconfirmed Retransmission 완료	UNCON_UP_RETX_OK	
Rx	Receive	RX1 channel open	RX1CH_OPEN	
		RX2 channel open	RX2CH_OPEN	
		Confirmed	CON_DOWN	
		Unconfirmed	UNCON_DOWN	
		수신 메시지(Rx Payload)	RX_MSG_FP<Fport number>_LEN<Length> : <Payload>	
	Mac command	Link check request	LINK_CHECK_REQ	
		Link check answer	LINK_CHECK_ANS: M <value> C <value>	
		Link adr request	LINK_ADR_REQ: DR <value> P <value> M <value>	
		Link adr answer	LINK_ADR_ANS	
		Rx parameter setup request	RX_PARAM_SETUP_REQ: O <value> DR <value> F <value>	
		Rx parameter setup answer	RX_PARAM_SETUP_ANS	
		Device status request	DEV_STATUS_REQ	
		Device status answer	DEV_STATUS_ANS: L <value> SNR <value>	
		New channel request	NEW_CH_REQ: I <value> F <value> DR <value>	
		New channel answer	NEW_CH_ANS	
		Rx timing setup request	RX_TIMING_SETUP_REQ: D 2	
		Rx timing setup answer	RX_TIMING_SETUP_ANS	
	thingPlug Server	디바이스에 메시지 전달 (extDevMgmt)	EXTDEVMGMT	
		디바이스 리셋 (DevReset)	DEVRESET	
		디바이스의 데이터 전송 주기 변경 (RepPerChange)	INTERVAL : <interval time>	
		즉시 보고 (ReplImmediate)	REPORT	
	Network Server	Confirmed Retransmission number change	CON_UP_RETX_NB : <Retransmission number>	
		Unconfirmed Retransmission number change	UNCON_UP_RETX_NB : <Retransmission number>	
Status	Ready	Ready 상태 확인	READY	(1)
Wake-up	Wake-up	Wake-up 상태	WAKE_UP	
sleep mode	sleep mode	Sleep mode 진입	SLEEP	
LBT	LBT	LBT 감지	LBT_DETECT	
UART	UART Status	CLI Command return value	OK	
			ERROR	
			BUSY	

(1) READY 자동 출력

- 1) Second Join 완료 후 (JOIN_COMPLETE 출력 이후)
- 2) Confirmed-up 송신에 대해 서버로 부터 ack 수신완료 후 (단, Confirmed-down의 형태로 ack를 수신 할 경우 Unconfirmed-up ack 전송 완료 후)
- 3) Unconfirmed-up 송신에 대해 RX1과 RX2 Open 완료 후
- 4) Confirmed-up 또는 Unconfirmed-up의 재전송 설정 횟수만큼 재전송 완료 후

■ Debug Message(Log) None의 경우

Debug message를 모두 출력하지 않는다.

기본적인 Command에 대한 응답과 처리 결과만 표시한다.

■ SKT KR Public 전용의 Debug Message의 경우

1. Debug Message(Log) On의 경우

ThingPlug 제어 명령 수신 Debug message

ThingPlug 제어 명령은 DevReset, RepPerChange, ReplImmediate, extDevMgmt 총 4개가 있다.

모듈은 ThingPlug 제어 명령을 수신하면 제어 명령에 대해 Application에 출력하여 UART를 통해 디바이스에 전달만 할 뿐, 그에 따른 동작을 수행하지 않는다. 따라서 디바이스에서는 이를 전달 받아 CLI Command를 통해 모듈을 제어해야 한다.

ThingPlug 제어 명령을 수신하면 모듈은 즉시 ack를 송신하는데, 이때의 ack는 Unconfirmed-up의 형태이며 payload 값은 Null이다.

DevReset : 디바이스 또는 모듈을 리셋한다.

RepPerChange <interval time(sec)> : 주기전송 interval time을 설정한다.

ReplImmediate : 데이터를 즉시 보고(송신) 한다.

extDevMgmt : 사용자 정의 제어 명령을 내린다.

1. ThingPlug 제어명령 DevReset 수신 시의 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_CONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 2 P 1

SEND : Freq: 922300000 DR: 0 PWR: 1

DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 922300000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_DATA_CONFIRMED_DOWN / Ver: 0

OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 2 P 222

Rx_MSG 222 : 008000

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

AppDataProcessWithThingPlug: RCV 128

DevReset

PrepareFrame: MType: FRAME_TYPE_DATA_UNCONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 3 P 4

SEND : Freq: 923300000 DR: 0 PWR: 1

DataReport :

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 923300000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

2. ThingPlug 제어명령 RepPerChange 수신 시의 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_CONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 4 P 1

SEND : Freq: 922900000 DR: 0 PWR: 1

DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 922900000, SF 12, BW 0, TO 25, RXC 0

OnRadioRxDone: MType: FRAME_TYPE_DATA_CONFIRMED_DOWN / Ver: 0

OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 3 P 222

Rx_MSG 222 : 0081010f

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

AppDataProcessWithThingPlug: RCV 129

RepPerChange : 15

PrepareFrame: MType: FRAME_TYPE_DATA_UNCONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 5 P 4

SEND : Freq: 921900000 DR: 0 PWR: 2

DataReport :

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

3. ThingPlug 제어명령 ReplImmediate 수신 시의 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_CONFIRMED_UP / Ver: 0
PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 6 P 1
SEND : Freq: 922100000 DR: 0 PWR: 1
DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
RX1CH Open: freq 922100000, SF 12, BW 0, TO 25, RXC 0
OnRadioRxDone: MType: FRAME_TYPE_DATA_CONFIRMED_DOWN / Ver: 0
OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 4 P 222
Rx_MSG 222 : 008200

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
AppDataProcessWithThingPlug: RCV 130

ReplImmediate

PrepareFrame: MType: FRAME_TYPE_DATA_UNCONFIRMED_UP / Ver: 0
PrepareFrame: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 7 P 4
SEND : Freq: 923300000 DR: 0 PWR: 1
DataReport :
RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
RX1CH Open: freq 923300000, SF 12, BW 0, TO 25, RXC 0
RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

4. ThingPlug 제어명령 extDevMgmt 수신 시의 debug message

PrepareFrame: MType: FRAME_TYPE_DATA_CONFIRMED_UP / Ver: 0
PrepareFrame: ADR ON, ADRACKReq 0, ACK 0 Fpending 0 FOptsLen 0 FCnt 8 P 1
SEND : Freq: 922700000 DR: 0 PWR: 1
DataReport : a1b2c3

OK

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1
RX1CH Open: freq 922700000, SF 12, BW 0, TO 25, RXC 0
OnRadioRxDone: MType: FRAME_TYPE_DATA_CONFIRMED_DOWN / Ver: 0
OnRadioRxDone: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 5 P 222
Rx_MSG 222 : 0000170123456789012345678901234567890123456789012345

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

AppDataProcessWithThingPlug: RCV 0

extDevMgmt

PrepareFrame: MType: FRAME_TYPE_DATA_UNCONFIRMED_UP / Ver: 0

PrepareFrame: ADR ON, ADRACKReq 0, ACK 1 Fpending 0 FOptsLen 0 FCnt 9 P 4

SEND : Freq: 921900000 DR: 0 PWR: 2

DataReport :

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

RX1CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 0

RX2CH Open: freq 921900000, SF 12, BW 0, TO 25, RXC 1

5. debug message에 대한 설명

- FRAME_TYPE_DATA_CONFIRMED_DOWN : 수신 메시지 형태로 Confirmed-down임을 나타낸다
- Rx_MSG : 수신 데이터의 payload 메시지를 의미하며 메시지 형태는 아래와 같다.
Rx_MSG <Fport> : <LoRaMAC version> <message type> <Payload length> <Payload>
1byte 1byte 1byte 1byte이상
- DevReset : ThingPlug 제어명령 중 하나로 디바이스 또는 모듈에 대 리셋 명령어를 나타낸다.
- RepPerChange : ThingPlug 제어명령 중 하나로 주기전송 interval time 설정 명령어를 나타낸다.
RepPerChange <interval time(sec)>의 형태로 출력된다.
- ReplImmediate : ThingPlug 제어명령 중 하나로 데이터 즉시 보고(송신) 요청 명령어를 나타낸다.
- extDevMgmt : ThingPlug 제어명령 중 하나로 사용자 정의 제어 명령어를 나타낸다.

부록 C. Other References

■ UART 사용하기

Application Note : AppNote_LOM20xA_Using_the_UART_[date]_[country].pdf

■ CLI Command 인터페이스

Application Note : AppNote_LOM20xA_CLI_Command_Interface_[date]_[country].pdf

■ 펌웨어 업그레이드

Application Note : AppNote_LOM20xA_Firmware_Upgrade_[date]_[country].pdf