



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

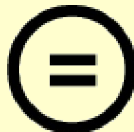
다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위 논문

이기종 리더의 통합 접속 및 관리를
위한 RPManager의 설계와 구현



2009년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

안 종 민

공학석사 학위 논문

이기종 리더의 통합 접속 및 관리를
위한 RPManager의 설계와 구현

지도교수 송 하 주

이 논문을 석사학위 논문으로 제출함

2009년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

안 종 민

안종민의 공학석사 학위논문을
인준함.

2009년 2월 23일



주 심 공학박사 김 중 남 인

위 원 공학박사 신 봉 기 인

위 원 공학박사 송 하 주 인

목차

1. 서론.....	1
1.1 연구의 필요성	2
1.2 연구목적 및 방법	4
1.3 논문 구성	5
2. 관련 연구	6
2.1 RFID 리더	7
2.2 RFID 태그	8
2.3 RFID 미들웨어	10
2.4 EPCglobal Reader Protocol & Reader Management Protocol ...	12
3. 이기종 리더의 통합접속 및 관리를위한 RPManager 의 설계	19
3.1 RPManager 의 구조	19
3.2 RPManager Object 구조	23
3.3 CustomAdapter 부위 구조	28
3.4 RPManager 의 TransportsubSystem 관련 설계	32
3.5 RPManager 의 JMX 지원	35
3.6 RPManager 의 Message 생성 구조	39
3.7 RPManager 의 설정 파일 관련 구조	40

3.8 구현 환경	4 3
4. 구현 및 결과	4 5
4.1 구현 테스트에 사용될 하드웨어 환경	4 5
4.2 구현 결과 및 동작 Sequence	4 5
4.2.1 Alien Reader CustomAdapter 구현	4 7
4.2.2 Intermec Reader CustomAdapter 구현	5 0
4.3 향후 개선 방안	5 3
5. 결론	5 4
6. 참고 문헌	5 5



그림 목차

그림 1 Reader Network 구성 및 제안	3
그림 2 RFID 시스템 구성 요소	6
그림 3 RFID리더의 전체 블럭다이어그램	8
그림 4 RFID미들웨어 구성요소	1 1
그림 5 리더 프로토콜 구성 Layer.....	1 2
그림 6 ReaderProtocol Object Diagram.....	1 3
그림 7 Reader Protocol 구조	1 6
그림 8 RPManager의 구성도	2 0
그림 9 RPManager의 구성도에 따른 기본동작 과정.....	2 2
그림 10 ReaderDevice의 구성	2 4
그림 11 RPManager의 CustomAdapter 구성도.....	2 9
그림 12 RPManager의 TransportsubSysyem Class Diagram	3 3
그림 13 RPManager의 Bean Class	3 6
그림 14 RPManager의 Jconsole 구동	3 8
그림 15 RP RM Message 생성	4 0
그림 16 CustomAdapter 동작 Sequence	4 6
그림 17 AlienReader의 CustomAdapter구현.....	4 8
그림 18 IntermecReader의 CustomAdapter구현	5 1

표 목차

표 1 전원 여부에 따른 태그 분류	9
표 2 ReaderDevice 주요 메소드 설명	2 5
표 3 Source 주요 메소드 설명	2 5
표 4 DataSelector 주요 메소드 설명	2 6
표 5 NotificationChannel 주요 메소드 설명	2 7
표 6 TagSelector 주요 메소드 설명	2 7
표 7 Trigger 주요 메소드 설명	2 8
표 8 CustomReaderDevice 주요 메소드 설명	3 0
표 9 CustomSource 주요 메소드 설명	3 0
표 10 CustomCommandChannel 주요 메소드 설명	3 1
표 11 SourceListener 주요 메소드 설명	3 2
표 12 ReaderControlChannel 주요 메소드 설명	3 3
표 13 ReaderTransportSubSystem 주요 메소드 설명	3 4
표 14 ReaderNotificationChannel 주요 메소드 설명	3 4
표 15 NetworkReaderControlChannel 주요 메소드 설명	3 4
표 16 NetworkReaderNotificationChannel 주요 메소드 설명	3 5
표 17 ReaderDeviceBean 주요 메소드	3 7
표 18 SourceBean 주요 메소드	3 7
표 19 RPManagerBean 주요 메소드	3 7
표 20 NotificationChannelBean 주요 메소드	3 8
표 21 ReaderData.Xml 의 구성	4 1
표 22 Config XML 항목별 설명	4 2

표 23 시스템 구현 환경.....	4 4
표 24 사용되어진 Reader의 하드웨어 환경.....	4 5
표 25 AlienRaederDevice 주요 메소드.....	4 8
표 26 AlienSource 주요 메소드.....	4 9
표 27 AlienNotificationChannel 주요 메소드.....	4 9
표 28 AlienCommandChannel 주요 메소드.....	4 9
표 29 AlienTransactionManager 주요 메소드.....	5 0
표 30 IntermecRaederDevice 주요 메소드.....	5 1
표 31 IntermecSource 주요 메소드.....	5 2
표 32 IntermecCommandChannel 주요 메소드.....	5 2
표 33 IntermecTransactionManager 주요 메소드.....	5 2



이기종 리더의 통합 접속 및 관리를 위한 RPManager의 설계와 구현

안 종 민



RFID시장에서는 Reader Protocol version 1.1 및 Reader Management Protocol version 1.0.1과 같은 RFID 리더 표준 통신 프로토콜이 제안되었음에도 불구하고 RFID 리더를 업체별로 독자적인 프로토콜만을 지원하는 리더만 유통되는 경우가 흔하다. 이는 장기적으로 RFID응용 시스템의 관리, 유지, 보수 측면에서 효율적이지 못하다. 이에 본 논문에서는 여러 회사의 다양한 리더를 표준 프로토콜을 통해 접근 및 관리가 가능한 통합 플랫폼인 RPManager를 제안한다. RPManager를 사용하면 미들웨어 측면에서도 표준 ReaderProtocol만을 사용하여 비표

준의 리더를 접속할 수 있고, 리더들을 통합 관리 할 수 있어 관리 및
유지보수의 효과를 거둘 수 있다.



Design & Implementation of RPManager for the integrated access and management of heterogeneous RFID Readers

Jong-Min Ahn

Department of Computer Engineering, The Graduate School,
Pukyong National University

Abstract

In RFID market, most RFID readers provide only proprietary network protocols, even though global standards such as Reader Protocol version 1.1 (RP v1.1) and Reader Management Protocol version 1.0.1 (RM v1.0.1) have been suggested. It can cause bad effect on long-term maintenance and management of RFID applications. In this paper, we propose RPManager as an integrated

platform that enables access and management of diverse readers from different vendors via standard protocols. Using RPManager, an RFID middleware can interoperate with non-standard readers via RP v1.1, and system administrators can manage readers via RM v1.0.1. Thereby, RPManager enhance the management and maintenance of RFID applications.



1. 서론

RFID(Radio Frequency Identification)는 제품에 붙이는 태그(tag)에 생산, 유통 보관, 소비의 전 과정에 대한 정보를 담고 자체 안테나를 갖추고 있으며, 리더(Reader)로 하여금 이 정보를 읽고, 인공위성이나 이동통신망과 연계하여 정보시스템과 통합하여 사용되는 활동 또는 칩을 말한다.[1] 이런 RFID기술은 우리 생활 주변에서 자주 볼 수 있는데, 주로 주변의 사물에 부착된 태그로부터 전파를 이용하여 사물의 정보 및 주변 환경을 인식하여 각 사물의 정보를 수집, 저장, 가공, 추적함으로써 사물에 대한 측위, 원격 처리, 관리 및 사물간 정보 교환 등 다양한 서비스를 제공하는 기술로 활용을 하고 있다.

이는 기존의 바코드를 대체하여 물품 관리를 네트워크화 및 지능화 함에 따라 유통 및 물품 관리뿐만 아니라 보안, 안전, 환경 관리 등에 혁신을 선도할 것으로 전망된다.[2] 과거에는 RFID장치에서 사용하는 태그의 값이 비싸 여러 분야에 적용하지 못하였는데, 근래의 지속적인 연구개발로 인해 RFID의 태그의 가격이 저렴하게 유통되어, 전세계적으로 여러 기술 방면에 RFID 기술을 사용을 할 수 있게 되었다. 이러한 RFID 기술의 도입과 응용은 IT산업뿐만 아니라 전통산업으로 분류되어온 물류, 유통, 국방, 조달, 건설, 교통, 제조, 서비스 등 전 산업분야에 걸쳐 큰 영향을 미칠 것으로 예상되고 있으며[3], 또한 여러 기관 및 단체에서의 RFID기술에 관한 표준들을 만들어 발표함으로써 좀 더 효율적이고 신뢰성 있는 기술로 발전을 하고 있다.

1.1 연구의 필요성

최근 여러 기관에서 RFID기술에 관한 표준들을 발표하는 가운데, EPCglobal[4]은 개별 물품 식별에서 가장 적합한 주파수 선택을 위한 시험이 저주파, 고주파, 초고주파 대역에서 의류, 의약품, DVD인식 등 모두 7가지 서비스 시나리오에서 진행 중이며, 현재 개별 물품 식별을 위한 새로운 프로토콜이 필요한지에 대한 연구도 진행 중에 있다. 이런 연구들을 통해 EPCglobal은 물리적인 리더와 물리적 리더와의 연동을 통해서 사물의 자동화된 인식과 백 엔드 시스템으로의 정보처리 기능을 수행하는 부분인 Middleware와 통신을 하기 위해 EPCglobal Reader Protocol Standard v1.1[5]과 이를 모니터링 하고 관리해주는 EPCglobal Reader Management Protocol Standard v.1.0.1[6]을 발표하였다. 그러나 현재 출시된 여러 업체의 RFID 리더장치들은 이러한 표준들을 따르지 않고, 독자적인 프로토콜을 사용하여 RFID 미들웨어와 리더 사이의 통신을 하고 있다. 이렇게 EPCglobal의 표준안이 속속 제안됨에 따라 기존 리더의 인터페이스 또한 표준 프로토콜에 의거하여 단일화할 필요가 있다. 이것은 리더와 미들웨어간의 접속뿐만 아니라 Reader의 관리 측면에서도 동일한 기능이 필요하다. 여러 개의 Reader가 하나의 표준화된 프로토콜을 사용하게 되면, 효율적인 측면, 관리적인 측면, 신뢰적인 측면에서 기존의 각 회사마다 고유의 프로토콜을 사용하는 것보다 높은 효과를 가져다 줄 수 있다.

또한 현재 기존에 유통되어 사용되고 있는 리더들 중에는 EPCglobal에서 발표한 표준 중 Low Level Reader Protocol v1.0[8](이하 LLRP)을 지원하는 리더들을 만들어 유통을 하고 있는 업체들도 있다. 하지만

미들웨어와 리더 사이의 통신규정을 제정한 Reader Protocol Standard v1.1(이하 RP v1.1)과 이를 모니터링 하고 관리하는 Reader Management Protocol Standard v1.0.1(이하 RMP)를 지원하는 리더는 시중에 나와있지 않은 실정이다. 이는 현재 시중에 나와있는 리더들은 [그림 1]의 (A)와 같이 모두 독자적인 프로토콜을 사용하여 미들웨어와 통신을 하는 구조로 되어 있다.[9] 국제 표준단체인 EPCglobal에서 표준까지 나온 이 시점에서 독자적인 프로토콜을 사용한다는 건 효율적인 측면, 관리적인 측면, 신뢰적인 측면에서 낮은 기대효과를 가져 올 수 있다.

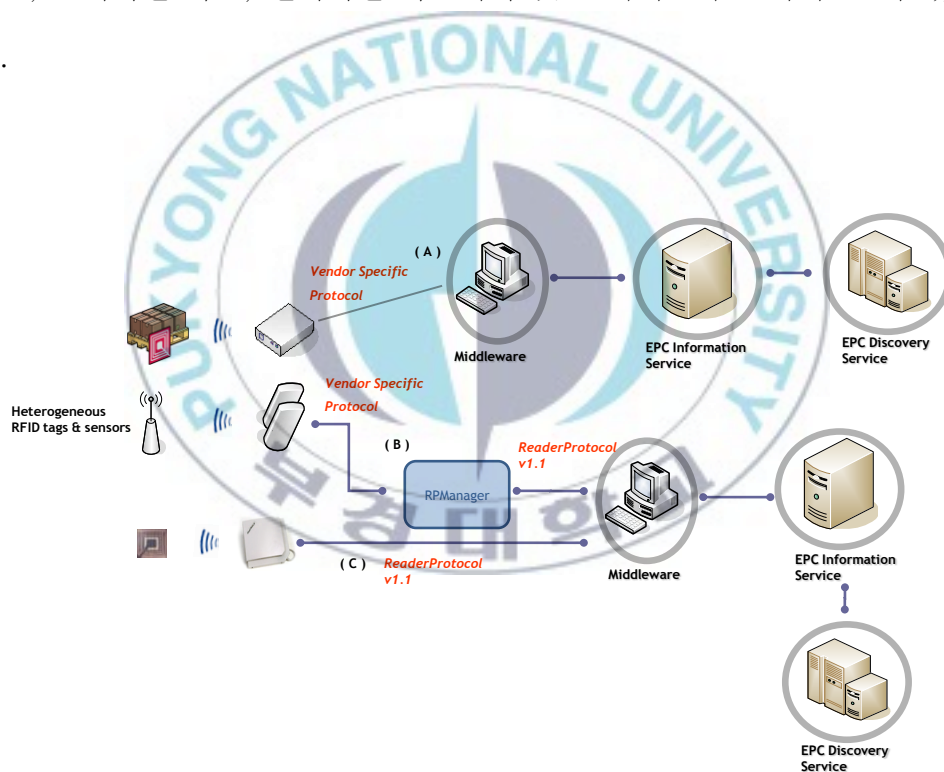


그림 1 Reader Network 구성 및 제안

따라서 기존 표준을 지원하는 리더의 경우 미들웨어와 연동이 될 수 있게 하고, 현재 널리 사용되고 있는 RFID 리더 및 센서 리더들을 일관

된 방식으로 미들웨어와 연동할 수 있게 해주는 프로그램과 이를 모니터하고 관리하는 프로그램이 필요하다.

1.2 연구목적 및 방법

본 논문에서 제시하는 RPManager는 [그림 1]의 (B)와 같이 표준을 지원하지 않는 리더들에 한해서는 리더와 미들웨어 사이에 모듈을 개발하여 여기를 통과하는 모든 비 표준리더들을 표준리더처럼 보여지게 구현하는 것이 주목적이다. 물론 표준을 지원하는 리더에 한해서는 [그림 1]의 (C)처럼 바로 미들웨어에 접근을 하면 된다.

이런 관점에서 볼 때 미들웨어 측면에서는 [그림 1]의 (B)와 (C)처럼 단일의 프로토콜만 사용하여 리더의 종류에 구애 받지 않게 된다면, 미들웨어 본연의 기능에 집중을 할 수 있을 것이고, 또한 그런 미들웨어의 활용성은 크게 늘어나게 된다. 여기에 부가적으로 단순히 통신 프로토콜 지원만으로 그치지 않고 이를 모니터링하고 관리하는 프로그램과 어댑터의 추가로 인한 다양한 비표준리더들을 지원하여 단일 프로토콜로 미들웨어에 접근할 수 있게 하고, 단일 S/W에서 여러 리더들을 표준리더로 지원하는 기술, 더 나아가서는 Web을 통한 리더 관리 등의 기능을 할 수 있도록 하고자 한다.

또한 RFID Network에서 RFID에 관한 전반적인 지식과 태그, 리더에 관해 조사를 하고, EPCglobal에서 제시한 EPC Network[10]에 관한 내용 또한 조사 및 연구를 한다. 그리고 무엇보다 RP v1.1와 RMP v.1.0.1에 대해 조사를 한다. 또한 기본적인 틀은 RP v1.1지원 및 RMP v1.0.1

지원을 바탕으로 설계하며, 다수의 이기종 리더들을 지원/관리 프로그램 개발부분으로 연구를 진행 한다.

1.3 논문 구성

본 논문은 다음과 같은 내용으로 구성된다. 2장에서는 RManager를 바탕으로 관련 연구를 다룬다. 3장에서는 RManager를 설계하고, 프로그램 개발에 대해 알아본다. 4장에서는 구현 및 적용 결과에 대해 알아본다. 마지막 5장에서는 결론과 향후 연구 방향에 대해 기술을 한다.



2. 관련 연구

RFID는 무선주파수를 이용하여 사물에 부착된 태그의 IC칩에 저장되어 있는 고유 정보를 안테나와 리더를 통해서 비 접촉 방식으로 수집하여 대상 물체를 관독 및 인식하는 기술을 말한다. 이런 RFID를 사용하는 기본적인 시스템 구성은 [그림 2] 이다.

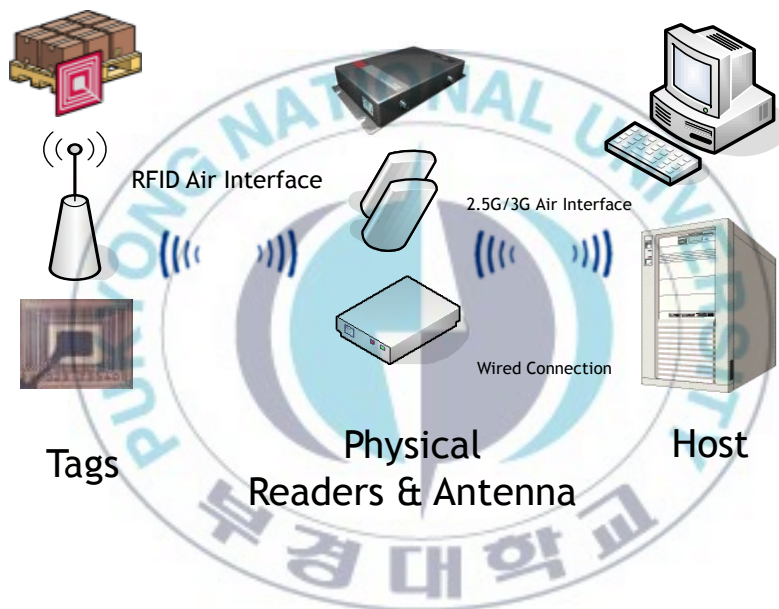


그림 2 RFID 시스템 구성 요소

RFID 시스템은 무선접속 방식에 따라 상호유도 (Inductively coupled) 방식과 전자기파 (Electromagnetic wave) 방식으로 나눌 수 있다. 상호유도 방식은 근거리(1m 이내), 전자기파 방식은 중장거리용 RFID로 사용되며, 상호유도 방식은 코일 안테나를 이용하며 전자기파 방식은 고주파 안테나를 이용해서 서로 무선접속을 한다. 상호유도 방식의 태그는 거의 수동으로 작동된다. 즉 태그의 IC칩이 동작하는데 필요한 모든 에

너지는 리더기에 의해 공급 되어진다. 따라서 리더기기의 안테나 코일은 주변지역에 강한 자기장을 발생한다. 방출된 자기장의 일부분이 리더기와 떨어져 있는 태그의 코일 안테나에 유도성 전압을 발생, 정류된 후 IC를 위한 에너지로 공급된다. 전자기파 방식의 태그는 IC칩을 구동하기 위한 충분한 전력을 리더기로 부터 공급받지 못하므로 장거리 인식을 위한 추가적인 전지를 포함하는 경우(능동형)도 있다.[11]

2.1 RFID 리더

RFID 리더는 전자기장을 이용하여 태그에 전원 공급 및 명령을 전송하고, 태그로부터 그 명령에 대한 응답을 받아 사용자가 저장하고 사용하는 장소로 정보를 송신하는 기능을 수행한다.

RFID 리더의 구성 요소로는 태그에서 사용할 전력을 공급하고, 데이터를 송·수신하기 위해서 무선으로 신호를 주고받을 수 있는 최종단의 안테나부, 안테나를 통해서 무선으로 데이터를 송·수신하기 위한 RF인터페이스부, RF 인터페이스 단을 제어하며 외부 인터페이스로부터 수신된 명령어에 대한 분석 및 태그로부터 수신된 데이터에 대한 전송을 하기 위한 제어부, 외부 인터페이스와 여러가지 통신 방법을 사용하여 태그의 데이터를 전송하기 위한 통신부로 구성되어있다. 다음 [그림 3] 는 RFID리더의 전체 구성요소를 블록다이어그램으로 나타낸 그림이다.

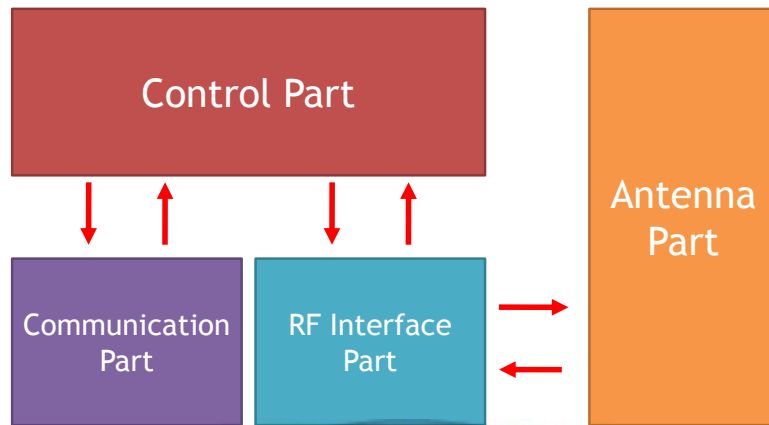


그림 3 RFID리더의 전체 블럭다이어그램

각각 Antenna Part는 자유공간에 신호를 뿌려주고, 또 그 뿌려진 신호를 수신하는 것으로 무선 통신 시스템에서 최종단의 장치이며, RF Interface Part는 디지털의 전기적 신호를 아날로그의 전기장 신호로 변환해 주는 부분으로 송신 선로와 수신 선로로 구분할 수 있다. 그리고 Control Part는 다양한 통신 방식에 의한 Application S/W와의 통신 및 수신된 명령에 대한 실행, 태그와의 통신제어, 신호의 코딩과 디코딩을 담당하고 있다.

2.2 RFID 태그

RFID태그는 IC칩과 안테나로 구성되고 모양과 크기가 다양하다. 동물

을 추적하기 위해 피부 아래에 이식되는 태그는 직경이 연필심만큼 작으며 길이는 1cm밖에 되지 않는다. IC칩의 주요기능은 데이터의 저장으로 메모리의 크기(25비트에서 512KB 이상), 메모리 형태(읽기 전용, 읽고 쓰기가 가능한 형, 한번만 쓰며 여러 번 읽기가 가능한 형태), 메모리 종류(EEPROM, 강유전체RAM(FRAM))에 따라 가격이 다양하다. 내부에 전지 포함 유무에 따라 수동형 태그(Passive Tag)와 능동형 태그(Active Tag)로 구분할 수 있고, 사용하는 주파수 대역에 따라 저주파 시스템과 고주파 시스템으로 나눌 수 있다.[12] 다음 [표 1]는 태그의 전원 여부에 따라 분류를 한 도표이다.

	능동형 RFID	수동형 RFID
특징	*태그에서 자체 RF신호 송신 가능 *배터리에서 전원 공급	*관독기의 신호를 변형 반사 *관독기의 전파 신호로 전원 공급
장점	*장거리(3M) 이상 전송 가능 *센서와 결합 가능	*배터리가 없으므로 저가격 구현 가능 *배터리 교체 비용 없음
단점	*배터리에 의한 가격 상승 *동작시간 제한	*장거리 전송 제한 *센서 류의 모듈 추가 제한
적용 분야	*환경감시, 군수, 의료, 과학 분야 *향후 스마트 디스트에 활용가능하나 현재는 비용 측면에서 불리함	*물류관리, 교통, 보안, 전자상거래 분야 *향후 저가의 스마트 라벨은 기존의 바코드를 대체

표 1 전원 여부에 따른 태그 분류

이 이외에도 상호 통신 접속 방식에 따라 상호유도(Inductively coupled) 방식과 전자기파(Electromagnetic wave) 방식으로 나눌 수 있다. 상호 유도 방식은 1m이내의 근거리, 전자기파 방식은 중장거리용으로 사용되며, 상호유도 방식은 코일 안테나, 전자기파 방식은 고주파 안테나를 이용하여 서로 무선접속을 한다.

2.3 RFID 미들웨어

RFID 미들웨어는 이기종의 RFID환경에서 발생하는 대량의 태그 데이터를 수집, 필터링하여 의미있는 정보로 요약하여 응용 시스템에 전달하는 시스템소프트웨어라고 정의 될 수 있다. [16] 이러한 미들웨어를 사용하는 동기는 크게 세가지로 들 수 있다. 첫 번째는 애플리케이션을 장치 인터페이스로부터 캡슐화하는 것이고, 두 번째는 리더와 센서에서 잡아낸 처리되지 않은 관측 결과를 처리해서 애플리케이션에서는 쓸모가 있는 고수준 이벤트만 볼 수 있도록 하고, 또한 애플리케이션에서 처리할 데이터 분량을 적정 수준으로 낮추는 작업을 한다. 마지막으로 세 번째는 리더를 관리하고 RFID 관측을 질의 하기 위한 애플리케이션 수준의 인터페이스를 제공하는 것이다.[13] 다음 그림은 RFID 미들웨어의 구성요소를 나타낸 그림이다.



그림 4 RFID미들웨어 구성요소

리더 어댑터 에서는 여러 다른 리더 인터페이스를 캡슐화 시켜서 애플리케이션 개발자들이 직접 처리하지 않아도 되도록 만들어주는 역할을 하고 있다. 이벤트 관리 계층에서는 여러 리더로부터 처리 되지 않은 RF 데이터를 받아서 정리하고 미리 설정해 둔 애플리케이션 수준 이벤트 필터를 바탕으로 필터링 역할을 하고 있다. 마지막으로 애플리케이션 수준의 인터페이스는 RFID 미들웨어 스택에서 최상층에 위치하고 있다. 이 인터페이스는 애플리케이션에서 일련의 리더로부터 만들어진 RFID 이벤트에 등록을 하고 필터링된 이벤트를 받아들일 수 있게 해 주는 표준화된 매커니즘을 제공하고 있다.[17] 그리고 미들웨어를 거친 데이터는 상위 EPCIS를 이용하여 상호 거래 당사자들 사이에서 오고 가는 EPC 데이터를 저장하고 질의하여 처리를 할 수 있다.[14] 또한 ONS를 이용하여 생산업체 정보 서비스의 주소를 제공하는 EPC 록업 서비스를 이용할 수 있다.[15]

2.4 EPCglobal Reader Protocol & Reader Management Protocol

EPCglobal Reader Protocol v1.1(이하 RP v1.1) 은 자동인식(AutoID) 기술의 산업 표준을 담당하는 EPCglobal에서 제안하는 리더 프로토콜이다. 900MHz대역의 de-facto 표준으로 자리를 잡아가고 있는 Reader Protocol v1.1은 RFID 리더가 EPCglobal에서 제안하는 미들웨어를 통해 RFID 태그를 읽고, 데이터를 쓸 수 있도록 제어하는 리더와 호스트 간의 프로토콜이다.

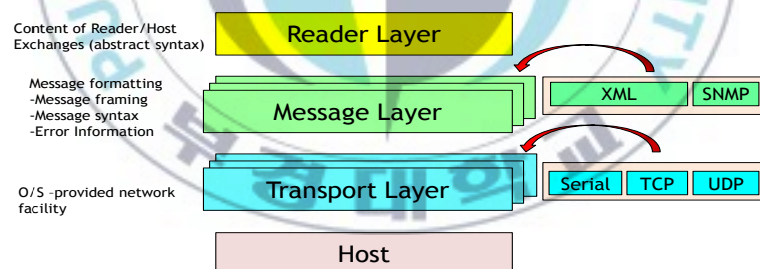


그림 5 리더 프로토콜 구성 Layer

이런 RP v1.1 표준에는 [그림 5]에서 보듯이 크게 Reader Layer, Message Layer, Transport Layer로 이루어져 있는 Protocol Layer와 Message Channel로 구성되어 있다. Message Channel은 리더와 호스트

사이에서 통신을 할 수 있도록 Control 채널, Notification 채널로 구성 되어있다.

Reader Layer에서는 리더와 호스트 사이에서 보내어지는 메시지에 들어갈 수 있는 내용 및 메시지 형식을 정의 하고 있다. 다음으로 Messaging Layer에서는 연결과 보안을 관리하고 호스트 명령 및 리더 응답, 그리고 통지를 패키징해서 전송 계층과 주고 받을 수 있도록 하는 일을 하고 있다. 마지막으로 Transport Layer에서는 OS에서 제공하는 서비스 또는 네트워킹을 지원하기 위해 필요한 하드웨어를 정의하고 있다.

RP v1.1의 Object구조는 다음과 같다.

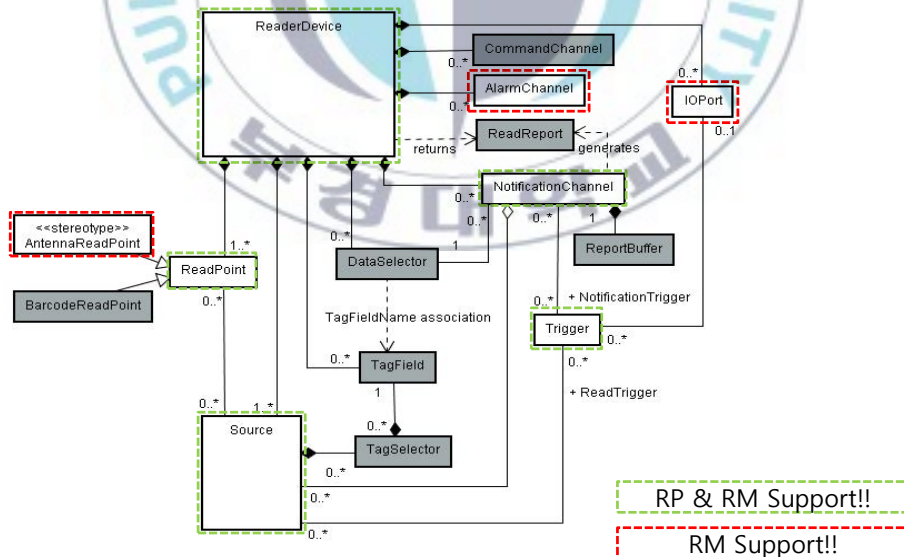


그림 6 ReaderProtocol Object Diagram

ReaderDevice Object : ReaderDevice 객체는 대다수의 객체에 토대가 되고 있는데, 여기에는 적어도 하나의 설정하기 전의 CommandChannel 객체를 리더 객체 모델에 포함되어있다. ReaderDevice 객체는 또한 몇몇의 하위속성들을 가지고 있는데 이는 리더 디바이스를 관리하거나, 리더들의 네트워크를 위해서 리더들의 네트워크 인터페이스가 사용되는 속성들을 가지고 있다.

CommandChannel : 커맨드 채널은 호스트에서 리더측으로 정보를 전송하기 위한 창구이다. 모든 메시지는 요청/응답의 패턴에 따라 커맨드 채널에서 변화되어지고, 대부분의 요소들은 리더와 호스트의 상호작용이 커맨드 채널에서 이루어진다.

Source Object : RF 안테나와 바코드 스캐너로 나뉜다. Source 는 ReaderDevice 로부터 읽어드린다. 읽어드린 Source 는 NotificationChannel 과 결합된다. Reader 객체모델에서 Source 객체는 하나 혹은 그 이상의 센서로부터 읽어 들인 리드 포인트를 요약한다.

ReadPoint Object : ReadPoint 는 자료를 취득하기 가능한 어떤 물리적인 존재를 볼 수 있다. 단순한 RF 태그 Reader 안테나는 ReadPoint 의 간단한 예이다. 현재 버전에서는, ReadPoint 가 유일히 지원하는 것은 AntennaReadPoint 이다.

AntennaReadPoint Object : AntennaReadPoint 는 안테나를 사용하여 물리적으로 실행되는 ReadPoint 이다. 각 AntennaReadPoint 은 Reader 의 상태를 감지하고 통계를 도와주고 유지한다.

Trigger Object : 트리거는 Source 의 합성 일원으로 존재할 수 있으나 NotificationChannel 객체, i.e, 트리거 객체가 Source 와 NotificationChannel 에서 추가되고 제거되지 않는 객체 이다.

IOPort Object : IOPort 는 Reader Device 로 다른 요소와 연결하기 위하여 외부 입출력 선을 제공하는 하드웨어를 의미한다. 하드웨어 업체는 선택적으로 IOPort 를 정의할 수 있다.

AlarmChannel Object : AlarmChannel 은 리더에서 호스트로 부터 비동기화된 메시지를 보내는 역할을 한다. AlarmChannel 에서 발생하는 메시지는 오직 호스트에게 흘러가게 된다. AlarmChannel 은 대부분을 비동기적 전송 상태거나 혹은 호스트에게 가는 메시지가 비상 상태일때 처리모드로 사용되어진다.

NotificationChannel Object : NotificationChannel 객체는 host 에 정보를 비동시성으로 전달하는 것을 책임진다. 한개 이상 Source 는 NotificationChannel 에 조합된다. DataSelector 는 데이터가 호스트로 보고되는 것을 결정한다.

위와 같이 구성되어있는 RP v1.1의 데이터 흐름은 [그림 7]에서 보는 것처럼 크게 네 가지 하위 시스템으로 구성되어 처리된다.

우선 처음 리더에서 데이터를 읽게 되면 여러 개의 안테나로 구성된 Source에서 읽은 태그ID 중 관심 있는 태그만을 필터링(Filtering)하고, 태그 이벤트를 생성 후 상위 Middleware가 요구하는 조건에 맞는 태그 정보를 MTB (Massaging / Transport Bind)를 통하여 전송한다. 이러한 구조를 통해 기본적인 리더의 기능을 정의 하고 있으며, EPC Class 태그 에 대한 기능들을 포함 하고 있다. EPC Class 0, 1을 위한 기본적인

READ, WRITE, KILL, LOCK 명령과 최근 화제가 되고 있는 C1G2 (Class 1 Generation 2)를 위한 Session 기능과 강화된 보안 관련 기능을 포함한다

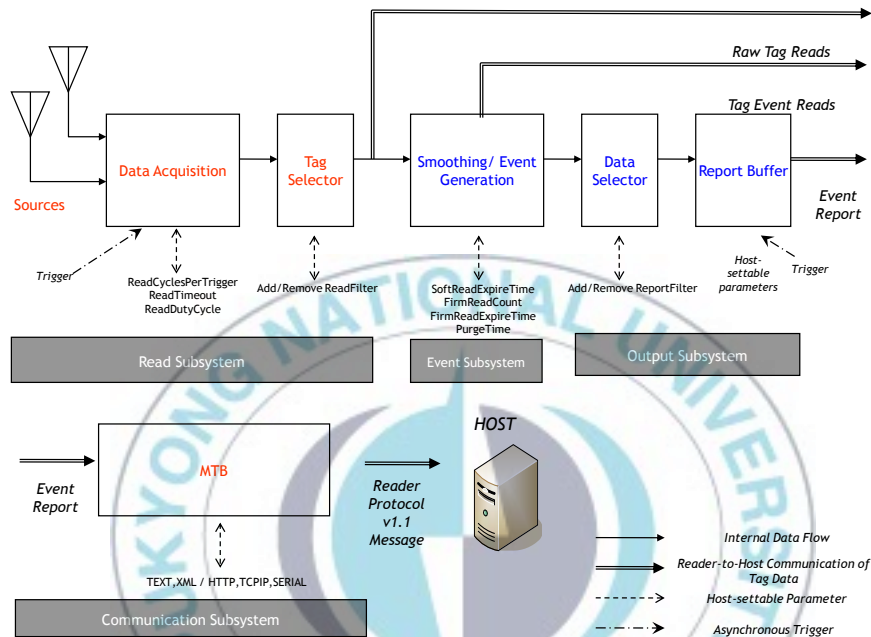


그림 7 Reader Protocol 구조

처음 리더에서 태그를 읽어 들어오는 구간이 Read Subsystem이다. 여기는 RFID 태그 데이터를 획득하는 단계로서 리더로부터 태그 데이터를 득하여 태그 데이터를 Event Subsystem으로 넘겨주는 역할을 한다. Tag Selector는 원하는 패턴의 태그만을 추출하기 위해 Tag Data의 Filtering기능을 수행한다.

다음 단계인 Event Subsystem에서는 Read Subsystem에서 전달 받은 태그 데이터를 Smoothing하고 이벤트 형식으로 이어지는 서브시스템에 전달한다. 태그데이터를 이벤트 형식으로 제공함으로써 호스트와의 통신

이 과다하지 않으면서도 효과적으로 태그 정보의 전달이 가능하게 된다. 그런 다음 Output Subsystem에서는 Event Subsystem에서 받은 이벤트에서 필요한 데이터 영역만을 추출하고 추후 호스트의 요청이 있을 때까지 이벤트를 버퍼에 잠시 유지하는 역할을 수행한다. 마지막으로 Communication Subsystem에서는 원하는 네트워크 프로토콜과 메시지 포맷의 조합(Message Transport Binding) 사용하여 호스트 측과 태그 정보를 교환한다. MTB를 구현하여 Output Subsystem의 Report Buffer에 있는 데이터를 호스트로 넘겨주는 역할을 한다. 그리고 MTB는 다양한 네트워크 환경에서 유연한 메시지 전달이 가능하도록 정의되어있다.

이러한 RP v1.1이 리더와 상위 계층간의 통신을 담당하고 있다면, RP v1.1이 일하고 있는 상태와 Tag의 상태 감시를 하는데 이용하는 EPCglobal Reader Management Protocol(이하 RMP1.0.1) 또한 EPCglobal에서 정의하고 있다. 이는 리더의 기능 정의나 리더의 현재 작동상태, 리더가 태그에 read, write, kill, lock등의 명령을 준 count, 이 명령들이 Failed한 count 등, 각 Tag에 따른 작업 Count, 상태변화등을 Monitoring하는 프로토콜이다.

RMP1.0.1은 Reader Protocol과 마찬가지로 Reader Management Protocol 의해 전달되는 Message는 Reader Protocol 에서도 정의된 MTB방식을 이용하여 Message 전달 및 Transport방법을 정의하고 있는데, 이는 [그림 5]에서 보여주듯 MTB를 2가지의 예를 들어서 소개를 하고 있다.

Reader Protocol Message Format은 input과 output과 같은 XML, transport는 TCP를 사용한다. Messaging formatting은 SNMP에서하고

UDP로 메시지 전송한다. 이렇듯이 똑같은 Reader Management Command Set 은 MTB 둘 다에 적용이 가능하다. 또한 RP v1.1 내의 Object Model을 보면 알 수 있듯이, RMP1.0.1과 한 쌍을 이루어 동작을 하는 것을 볼 수 있다.



3. 이기종 리더의 통합접속 및 관리를위한 RPManager의 설계

RPManger는 앞서 언급을 한대로 EPCglobal의 RP v1.1을 지원하는 리더의 경우에는 표준 프로토콜에 따라 Middleware와 연동이 될 수 있게 하고, 현재 널리 사용되고 있는 RFID 리더 및 센서 리더들에 대해선 일관된 방식으로 Middleware와 연동할 수 있게 하는 기능을 제공하는데 그 목적을 두고 있다. 그리고 또한 EPCglobal의 RMP1.0.1을 지원하는 리더들에 대해서는 표준 프로토콜에 따라 리더를 통합관리가 가능하고, 그렇지 않은 리더들에 대해서는 RMP1.0.1을 이용해서 관리가 가능하도록 S/W 계층을 사용해서 지원 가능하게 하는 모듈이라고 할 수 있다.

3.1 RPManager의 구조

본 논문에서 제시하는 RPManager를 구성하고 있는 요소로는 아래 [오류! 참조 원본을 찾을 수 없습니다.]에서 보듯이, ReaderDevice, Custom Adapter, Jetty, Mbean Server, Admin JSP 로 구성이 되어있다.

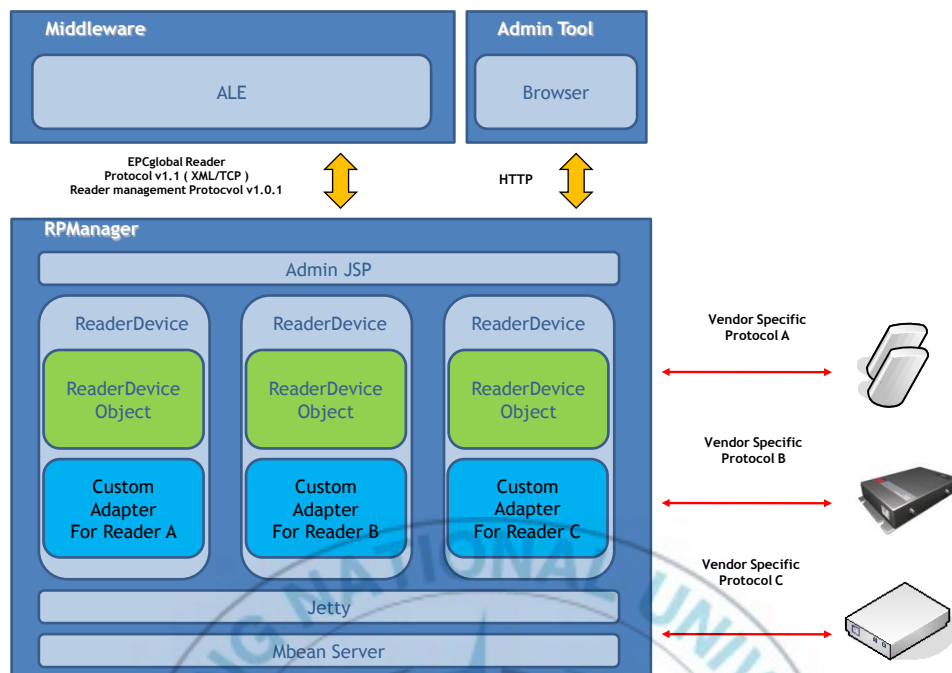


그림 8 RPManager의 구성도

우선적으로 가장 핵심적인 ReaderDevice객체가 있는데, 이는 RP v1.1과 RMP v1.0.1의 기능을 수행하는 객체이다. 그리고 ReaderDevice 객체와 1:1 쌍을 이루는 CustomAdapter가 있는데, 이는 RPManager에 연결되는 리더마다 고유의 CustomAdapter가 존재한다. 이렇게 CustomAdapter를 각각 리더마다 따로 존재하는 이유는 연결되는 리더마다 존재하는 고유의 프로토콜로 접근을 하고, 명령어를 쓰기 때문이다. MBean Server는 RPManager에서 생성되어진 객체를 전부 Bean형식으로 관리하여 좀 더 체계적으로 연결되는 리더와 연관되는 객체의 관리 및 모니터링 하기 위해 사용하고 있다. Jetty와 Admin JSP, Admin tool은 RPManager의 Web을 통해 RPManager의 모니터링과 컨피그 관련

설정들을 할 수 있게 하기 위해 쓰여지고 있다. 여기에서 가장 핵심이 될 수 있는 부분은 바로 ReaderDevice 객체라고 할 수 있다. 이는 앞서 언급한 대로 RP v1.1과 RMP v1.0.1의 기능을 지원하게 해주는 핵심적인 요소라고 할 수 있다. 이 ReaderDevice안에는 ReaderDevice객체와 연결되어지는 리더 전용 CustomAdapter와 한 쌍을 이루어 구성이 되어 있다.

다음은 RManager의 구동에 관련한 글이다.

일단 RManager의 ReaderDevice는 하나의 리더에 한 개의 객체가 생성이 되게 되는데, 리더의 연결 정보는 xml형식으로 되어있는 Config 파일에서 리더의 Data를 읽게 된다. 그런 다음 그에 해당하는 ReaderDevice와 CustomAdater가 생성이 되는 것이다. 이렇게 생성된 CustomAdapter는 연결되는 비표준 어댑터와 송수신을 수행하는 RManager 내의 어댑터로 볼 수 있다. 이처럼 CustomAdapter의 API를 이용해서 여러 회사의 비표준 리더들에 대한 어댑터를 생성하여 RManager를 이용을 한다면 연결되는 비표준 리더가 RP v1.1과 RMP v1.0.1을 지원하는 리더처럼 사용되어 질 수 있다.

아래의 [그림 9]는 위 설명을 바탕으로 나타낸 좀 더 구체적인 RManager의 동작 과정을 나타낸 그림이다.

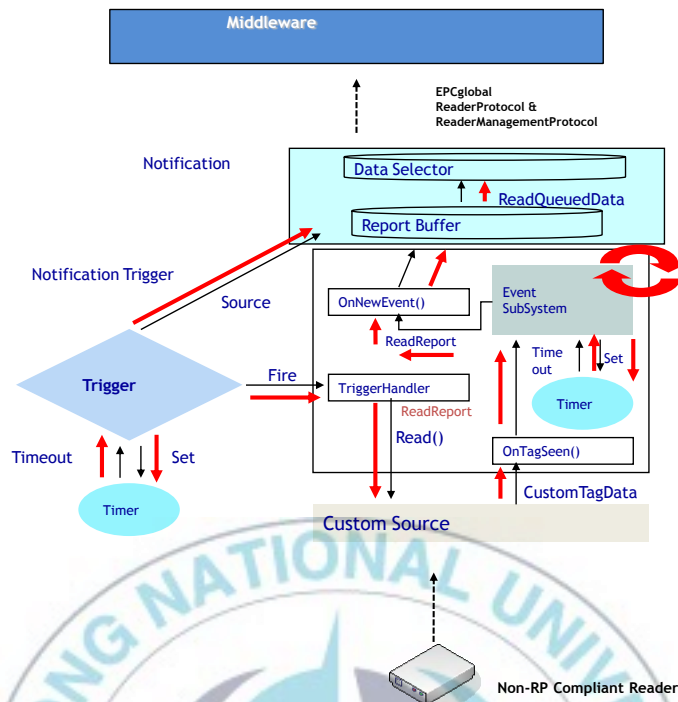


그림 9 RPManger의 구성도에 따른 기본동작 과정

먼저 RPManger내에 태그를 읽게하는 주기를 설정해주는 Trigger를 생성하여 Set을 해주게 되면 Trigger핸들러에 의해 CustomAdapter쪽으로 Read명령이 내려지게된다. 그림 RP v1.1을 지원하지 않는 리더로부터 태그 데이터가 올라오게 되는데, 이때 최초로 CustomChannel단계에서 데이터를 받아 CustomSource의 onTagSeen을 통해 Source로 데이터가 이동하게 된다. 이동하게된 데이터는 내부적으로 태그셀렉터의 여부를 판단하여, 태그셀렉터가 존재한다면 1차 필터링을 하여 데이터가 올라오고 그렇지 않은 경우에는 바로 EventSubSystem으로 데이터가 이동하게된다. EventSubSystem으로 이동한 데이터는 RP v1.1에서 제시한 이벤트 방식으로 이벤트가 태그데이터마다 부여되어 Soomthing / Filtering 되게 되며, 이런 데이터는 ReadReport형식으로 재가공되어

ReportBuffer에 데이터가 담기게된다. 그런다음 ReportBuffer에 담긴 데이터는 NotificationChannel Trigger의 동작에 따라 미리 설정이 되었거나, 혹은 설정을 한 DataSelector에 의해 데이터가 추가적으로 가공되어 상위 애플리케이션으로 완전한 RP v1.1 메시지형태로 이동하게된다.

상위로 이동한 메시지는 XML형식의 메시지이며, 이는 현재 RPManager가 XML메시지 형식만을 지원하여 이런 방식을 지향하는 것이고, 추후 추가 연구를 통해서도 Text또한 지원할 예정에 있다.

3.2 RPManager Object 구조

RPManager에는 기본적으로 상위 미들웨어와 연결 및 통신을 담당하는 `com.lit.rpsys.transportsubsystem`과 하위 물리적인 리더와 연결 및 통신을 담당하는 인터페이스적인 `com.lit.rpsys.abstractreaderdevice` 패키지가 존재한다. 이 인터페이스는 물리적인 리더에 따라 이 클래스를 상속받아서 해당 Adapter를 제작하기도 한다. 다음으로 하위 물리적인 리더에서 올라온 정보들을 상위 미들웨어에게 RP v1.1과 RMP v1.0.1에 상응하는 메시지로 만들어주는 `com.lit.rpsys.msg` 와 `com.lit.rpsys.msg.rpmanager` 패키지가 존재하고 있다. 그리고 가장 중요한 RP v1.1과 RMP v1.0.1의 역할을 수행하는 `com.lit.rpsys.readerprotocol` 패키지가 존재한다. 그외에 RPManager에서 사용할 각종 데이터 타입의 클래스들을 모아둔 `com.lit.rpsys.datatype` 과 `com.lit.rpsys.util` 패키지로 구성되어 있다.

다음은 RPManager의 RP v1.1과 RMP v1.0.1 표준을 지원하기 위해

서 com.lit.rpsys.readerdevice 패키지에 각각 서브 시스템에 관한 클래스를 구성하고 있다. 각각 클래스의 주요 역할은 다음과 같다.

ReaderDevice : 아래의 [그림 10]은 리더 디바이스의 구성을 타낸 그림이다. 리더 디바이스 오브젝트는 모든 리더 오브젝트의 기본 컨테이너가 되는 객체로써 하나의 리더를 대표한다 리더 디바이스는 적어도 하나의 지정된 명령 채널(Command Channel)을 가지고 있으며 추가적으로 리더 디바이스의 속성들과 네트워크에 필요한 객체들을 가지고 있다. 이 객체들은 Sources, DataSelectors, NotificationChannels, Triggers, TagSelectors 들로써 리더 디바이스가 생성될 때 내부적으로 같이 생성된다.

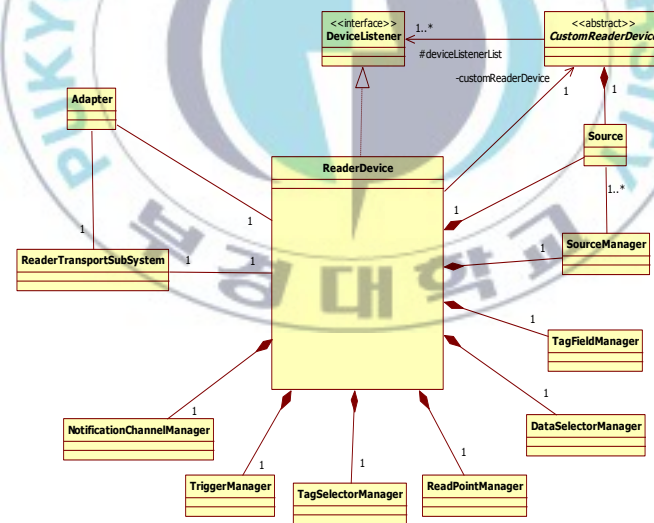


그림 10 ReaderDevice의 구성

주요Method	설명	비고
connectCustomReader()	Config에 설정된 물리적인 리더를 연결하는 메소드	

initialize()	RPManager의 Config에 설정된 Port값을 기본으로 TCP/IP통신을 할 수 있게끔 통신포트를 열어주는 메소드	
initializeDevice()	Source, DataSelector, Trigger등 각 객체별로 생성된 값들을 초기화 해주는 메소드	

표 2 ReaderDevice주요 메소드 설명

Source : 기본적으로 태그 정보를 읽어들이어서 그를 바탕으로 만들어진 이벤트를 상위계층에 전달하는 것이 목적이다.

주요Method	설명	비고
OntagSeen()	태그가 읽혀졌을때 위에서 저장해둔 리스트를 참조해서 그것의 OnTagSeen을 호출함으로써 그 정보를 Source로 전달 즉,Custom Source로부터 Tag 정보가 Source로 전달될때 호출되는 메소드	
NewEvent()	EventSubSystem에서 Event가 발생하였을때 Source에게 알려주기위해 EventSubsystem이 호출하는 메소드	
onNewEvent()	Custom Source로부터 Tag정보가 아닌 Event정보가 바로 올라올때 Custom Source가 호출하는 메소드	
Read()	연결된 물리적 리더에게 태그를 리드하라는 명령을 보내주는 메소드	
Select()	실제 1차 필터링인 TagSelector의 역할을 수행하는 메소드	

표 3 Source 주요 메소드 설명

DataSelector : 데이터 셀렉터 오브젝트는 noti피케이션 채널(Notification Channel)에 의해 유지될 수도 있고 태그를 읽는

명령의 파라미터로써 제공될 수도 있다. 만약 데이터 셀렉터에서 지정할 필드가 유효한 값을 가지고 있지 않을 경우에는 해당 필드는 무시되어서 noti피케이션 정보등에 포함되지 않는다.

주요Method	설명	비고
addFieldNames()	생성된 데이터셀렉터 객체에 사용자가 원하는 필드를 추가하는 메소드	
removeFieldNames()	생성된 데이터셀렉터 객체에 사용자가 원하는 필드를 삭제하는 메소드	
addEventFilters()	생성된 데이터셀렉터 객체에 사용자가 원하는 이벤트필터를 설정하는 메소드	
removeEventFilters()	생성된 데이터셀렉터 객체에 사용자가 원하는 이벤트필터를 해제하는 메소드	
addTagFieldNames()	생성된 데이터셀렉터 객체에 사용자가 원하는 태그필드를 추가하는 메소드	
removeTagFieldNames()	생성된 데이터셀렉터 객체에 사용자가 원하는 태그필드를 삭제하는 메소드	

표 4 DataSelector 주요 메소드 설명

NotificationChannel : noti피케이션 채널은 기본적으로 하나의 호스트에게 noti피케이션 리포트를 제공하는 통로이다. noti피케이션 채널은 일련의 소스들(Sources)의 목록을 유지하고 있으며 이들 소스들로부터 받는 이벤트를 모아서 호스트에 리포트 한다. 이 때 데이터 셀렉터(Data Selector)로부터 리포트하는데 어떤 필드들이 필요한지 정보를 받아서 해당 항목만 리포트 하게 된다.

주요Method	설명	비고
setDataSelector()	현재 생성된 noti피케이션채널 객체에 미리 생성되어진 데이터셀렉터 객체를 셋팅하는 메소드	

putReportBuffer()	상위 애플리케이션쪽으로 보내는 메시지를 생성하기 전에오는 데이터들을 모아주는 메소드	
onFire()	putReportBuffer 메소드를 통해 모인 데이터들을 조립하여 상위 애플리케이션으로 전송해주는 메소드, noti피케이션 채널내의 핸들러 안에 위치.	

표 5 NotificationChannel 주요 메소드 설명

TagSelector : 태그 셀렉터는 소스(Source) 오브젝트가 멤버로써 가질 수도 있고 필요하지 않은 경우 하나도 가지지 않을 수도 있는 오브젝트이다. 기본적으로 소스는 태그 셀렉터를 하나도 가지지 않고 있다. 태그 셀렉터는 리더로부터 읽혀 들어오는 태그들의 정보를 1 차적으로 걸러내는 필터링 역할을 담당하며 단순한 비트단위의 패턴매칭 방법을 사용한다.

주요Method	설명	비고
getTagField	생성된 태그셀렉터 객체에 사용된 태그필드를 가져오는 메소드	
getValue	생성된 태그셀렉터 객체에 사용된 Value를 가져오는 메소드	
getMask	생성된 태그셀렉터 객체에 사용된 Mask를 가져오는 메소드	
getInclusiveFalg	생성된 태그셀렉터 객체에 사용된 InclusiveFlag를 가져오는 메소드	

표 6 TagSelector 주요 메소드 설명

Trigger : 트리거는 source 에 적용될때는 ReaderTrigger 로 NotificationChannel 에 적용될때는 NotificationTrigger 로 동작한다.

주요Method	설명	비고
RegTriggerHanddler()	실제로 트리거의 서비스를 받는 객체의 핸들러를 등록시키는 메소드	
Fire()	Trigger의 타임아웃이 발생하였을시 수행되는 메소드	

표 7 Trigger 주요 메소드 설명

3.3 CustomAdapter 부위 구조

RPManager는 하나의 리더에 대해 한 개의 customAdapter를 생성하여 물리적인 리더와의 통신관계를 정의하고 있다. 이런 물리적인 리더와의 연결 및 해당 명령어 처리는 리더회사마다 전부 다를 수가 있는데, 이를 위해 RPManager는 상위 Abstract Class로 CustomAdapter라는 API를 두어서 회사별로 customAdpater 구현을 통일화 하고 편의 제공을 위해 Framework를 제공하고 있다.

다음 아래 [그림 11]은 CustomAdapter의 구성을 나타낸 Diagram이다.

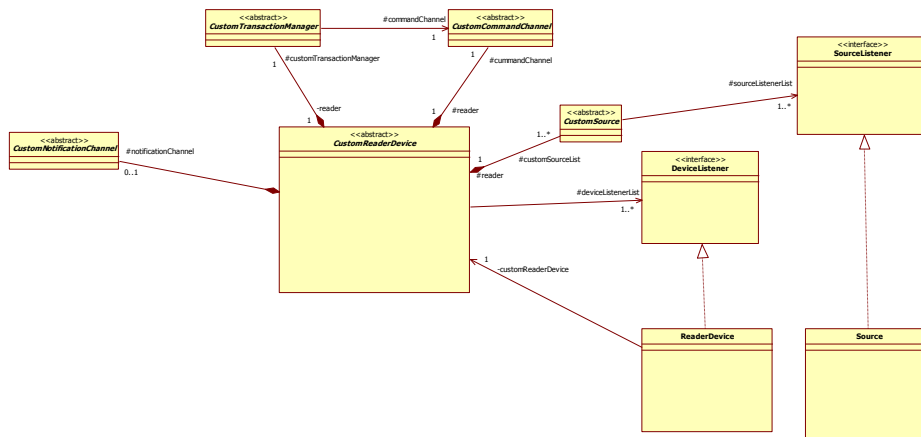


그림 11 RPManager의 CustomAdapter 구성도

CustomAdapter에서 사용되는 주요 클래스들을 살펴보면 아래와 같다.

CustomReaderDevice : 개별 리더의 Custom Reader Protocol을 이용하여 직접 리더를 제어하고, 네트워크 연결을 설정한다. Custom Reader Device 객체는 리더에 Tag 정보를 수집하는 Custom Source를 생성하고 Custom Source 목록을 유지한다. 하나 이상의 Custom Source를 생성할 수 있으며, 반드시 하나 이상의 CustomSource를 유지해야 한다. ReaderDevice 객체로 부터 Connect() Method가 호출되면, 리더에 명령을 전달하는 CustomCommandChannel과 CustomNotificationChannel을 생성하여 리더와 통신할 수 있는 환경을 만든다.

주요Method	설명	비고
isOpen()	현재 CustomReaderDevice와 리더의 연결 상태를 반환하는 메소드	
Connect()	물리적인리더가 통신하기 위한 명령을 수행한다. 리더와의 통신을 위하여 Custom Command	

	Channel과 Custom Notification Channel을 생성하고, 로그인이 필요한 리더는 id, password 파라미터를 이용하여 로그인이 가능하게해주는 메소드	
Close()	물리적인 리더와의 연결 해제를 위하여, Custom Reader Protocol에 정의된 Reader와의 연결 해제 명령을 호출하는 메소드	
sendRequest()	Custom Reader Device가 생성한 Custom Command Channel을 통하여 리더에 명령을 전달해주는 메소드	
onReply()	SendRequest 메소드를 통하여 리더에게 전달된 명령에 대한 응답을 받는 메소드	

표 8 CustomReaderDevice 주요 메소드 설명

CustomSource : 개별 리더에서 사용하는 Custom Reader Protocol을 통하여 Tag 정보를 수집하고, 수집된 Tag 정보를 Source에게 전달한다. Source는 CustomSource에서 제공하는 Method를 이용하여 One-time, AutoMode로 Tag 정보를 수집할 수 있으며, CustomSource에서 읽어들이는 Tag 정보는 SourceListener를 통하여 전달한다.

주요Method	설명	비고
Read()	리더에게 Tag 읽기 명령을 CommandChannel을 통하여 전송하는 메소드	
onReply()	SourceListenerList 목록에 등록된 Source에게 리더로 부터 읽혀진 Tag 정보를 전달해주는 메소드	

표 9 CustomSource 주요 메소드 설명

CustomCommandChannel : TCP/IP를 통하여 리더에게 명령을 전송하기 위한 Command Channel이다. CustomReaderDevice와 리더 사이에서 소켓을 생성하여 리더와 연결한다. SendRequest()를 통해 리더에 명령을 전달하고 SendReply()를 통해 명령을 상위로 전달한다. 일정 시간 리더로부터 응답이 들어오지 않거나, 리더와의 연결상태 유지를 위하여 TimerService를 통한 TimeOut을 설정한다.

주요Method	설명	비고
sendRequest()	Custom Command Channel의 write()를 이용하여 리더에게 명령을 전송하는 메소드	
setConnection()	리더와 연결을 설정하기 위한 메소드	
Disconnect()	리더와의 연결 해제하기 위하여 연결되어 있는 In/Output Stream과 Socket을 닫는 메소드	
Write()	OutputStream을 통하여 리더에게 명령을 전달하는 메소드	
Read()/readLine()	BufferedReader를 통하여 리더로부터 전해오는 응답을 한 문자 또는 한 라인씩 받는 메소드	
sendReply()	리더로부터 받은 TagID와 Tag가 읽혀진 시간을 ReportData 객체에 저장하여 CustomReaderDevice로 올려보내는 메소드	
onReply()	리더에게 보낸 명령에 대한 응답을 Custom Reader Device에게 전달하는 메소드	

표 10 CustomCommandChannel 주요 메소드 설명

SourceListener : CustomSource에서 보내오는 Tag정보를 받기 위한 CustomSource와 Source간의 인터페이스. CustomSource는 CustomSource에 유지하고 있는 Source목록을 참조하여

SourceListener를 통하여 Tag정보를 전달한다

주요Method	설명	비고
onTagSeen()	CustomSource는 리더로 부터 읽혀진 Tag 정보를 Source로 전달하는 메소드	
OnNewEvent()	ventSubSystem을 지원하는 리더에서 보내오는 Event Tag 정보를 Source로 전달하는 메소드	

표 11 SourceListener 주요 메소드 설명

3.4 RPManager의 TransportsubSystem 관련 설계

RPManager의 전송관련 클래스들을 한 패키지에 모아둔 곳이 바로 transportsubsystem 패키지이다. 이 패키지안에 있는 클래스들은 RPManager의 commandchannel 포트와 notificationchannel 포트를 생성해주는 역할을 담당하고 있으며, 최초로 메시지가 생성되어 상위 애플리케이션과 통신 할 수 있도록 연결하여 생성된 메시지가 이동할 수 있는 통로 역할도 한다. 아래의 [그림 12]는 전송관련 클래스들을 모아둔 TransportSubSystem의 클래스 다이어그램을 나타낸 그림이다.

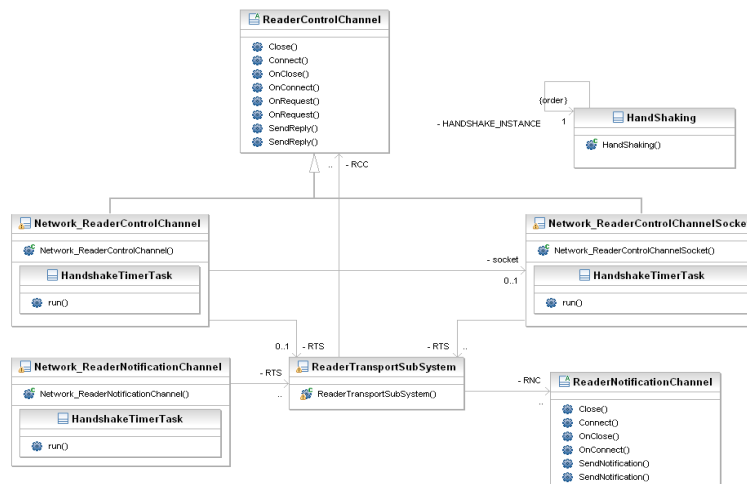


그림 12 RPMManager의 TransportSubSystem Class Diagram

ReaderControlChannel : Adapter에서 ReaderDevice에게 명령을 전송하기 위해 Channel생성에 관한 abstract Class이다.

주요Method	설명	비고
Connect()	Host의 ReaderControlChannel과 연결해주는 메소드	
OnRequest()	Host에서부터 명령이 전송받기 위하여 사용되는 메소드	
SendReply()	Host으로 응답을 전송해주는 메소드	

표 12 ReaderControlChannel 주요 메소드 설명

ReaderTransportSubSystem : ReaderDevice 에서 Adapter 로 명령을 전송하거나 응답을 받기 위한 abstract Class이다.

주요Method	설명	비고
OnRequest()	Adapter에서부터 명령이 전송되었을 경우 호출	

	되는 이벤트 메소드	
SendReply()	Host에게 메시지를 reply해주는 메소드	
SendNotification()	Host에게 Notification 정보를 전송하는 메소드	

표 13 ReaderTransportSubSystem 주요 메소드 설명

ReaderNotificationChannel : ReaderDevice에서 Adapter에게 Notification 정보를 전송 받기 위해 Channel생성에 관한 abstract Class이다.

주요Method	설명	비고
Connect()	Adapter의 HostNotificationChannel과 연결해주는 메소드	
SendNotification()	Adapter에게 Notification 정보를 전송하기 위하여 사용하는 메소드	

표 14 ReaderNotificationChannel 주요 메소드 설명

NetworkReaderControlChannel : Adapter에서 ReaderDevice에게 명령을 전송하기 위한 채널이다.

주요Method	설명	비고
OnConnect()	Connect 명령을 수행한 뒤 Host로부터 전송되는 데이터를 받기 위하여 새로운 쓰레드를 생성한 후 listen 상태로 대기하는 메소드	
OnRequest()	Host에서부터 명령을 전송받기 위하여 사용되는 메소드	

표 15 NetworkReaderControlChannel 주요 메소드 설명

NetworkReaderNotificationChannel : ReaderDevice에서 Adapter에게 Notification 정보를 전송받기 위해 필요한 Channel이다.

주요Method	설명	비고
Connect()	Adapter의 HostNotificationChannel과 연결해 주는 메소드	
SendNotification()	Adapter 에게 Notification 정보를 전송하기 위하여 사용하는 메소드	

표 16 NetworkReaderNotificationChannel 주요 메소드 설명

3.5 PRManager의 JMX 지원

JMX는 Java Management Extension라 하여 프로그래머들에게 자바 어플리케이션의 모니터링과 관리기능을 제공해주고 있다.[18] 실제로 이 API는 웹서버에서 네트워크 디바이스, 웹폰에 이르기까지 자바로 이용가능한 것은 어느 것이든 로컬 혹은 원격으로 처리 할 수 있게 해주고 있다. 이런 JMX 기술을 사용하면 ManageBean으로 알려진 하나 이상의 사용자 JavaBean 오브젝트를 통해 한 기계안의 애플리케이션, 디바이스, 또는 서비스를 원격으로 제어 할 수 있다.

이런 JMX기술을 RPManager에 적용하여, 사용자 및 관리자로 하여금 Jconsole을 통해서 RPManager를 관리할 수 있도록 하는데 목표를 맞추어 개발을 하였다. 또한 추후 RPManager의 추가 보완을 통해 웹 연동 및 리더의 동적 로딩 시스템 구축을 구현 할 때도 유용하게 쓰일 수 있다.

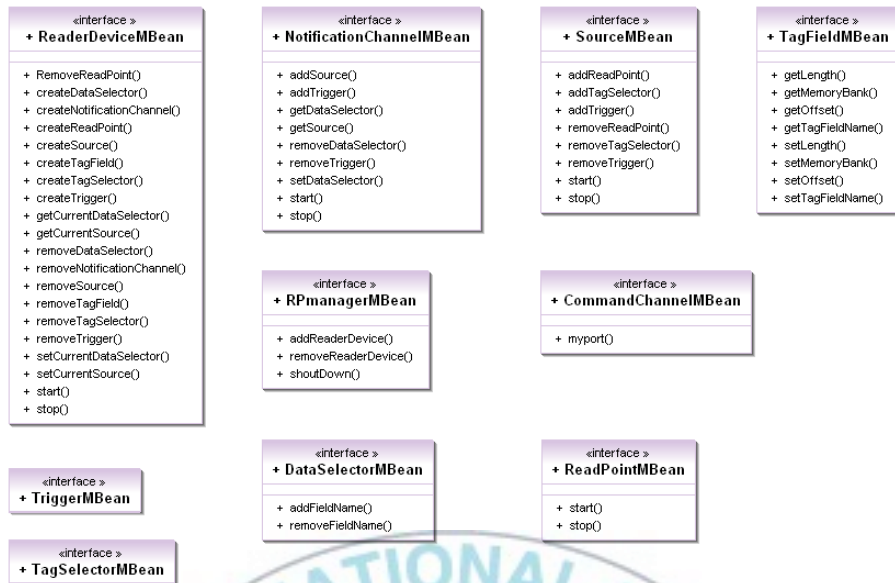


그림 13 RPMManager의 Bean Class

위 [그림 13]은 RPMManager에서 JMX를 적용하기 위해, 자주 사용되는 클래스들을 묶어서 Bean객체로 구현하기 위한 인터페이스들을 클래스 다이어그램으로 표현한 그림이다. 각각 클래스의 설명은 위의 구현 파트에서 설명한 내용과 동일 하므로 생략하고 각 주요 Bean클래스에서 구현하고 있는 메소드에 대한 설명은 아래와 같다.

ReaderDeviceBean : ReaderDevice를 Bean객체로 만들기 위해 생성된 인터페이스

주요Method	설명	비고
Start()	ReaderDevice를 구동하여 CommandChannel에 Accept 하여 Host측의 응답을 대기한다. 핸드쉐이킹은 연결후에 일어난다.	
Stop()	CommandChannel에 Accept를 하지 않고 Start	

	가 불리울때까지 ReaderDevice는 대기한다. Working중일때 이 Method가 호출이 되면 Interrupt를 발생시킨다.	
--	--	--

표 17 ReaderDeviceBean 주요 메소드

SourceBean : Source를 Bean객체로 만들기 위해 생성되어진 인터페이스

주요Method	설명	비고
Start()	생성되어진 source Object를 활성화 시킨다. 이 상태에서는 ReaderDevice에 연결을 할 수 있는 상태를 의미한다.	
Stop()	생성되어진 source Object를 비활성화 시킨다. 만약 활성화 상태에서 Source가 사용중에 있다 면 그 작업에 Interrupt를 걸어 중지 시키고 다 시 start가 호출될때까지 대기한다.	

표 18 SourceBean 주요 메소드

RPManagerBean : RPManager를 Bean객체로 만들기 위해 생성되어진
인터페이스

주요Method	설명	비고
addReaderDevice()	연결되어지는 ReaderDevice의 초기상태는 Physica상의 Reader와는 Start가 오기전 까지 연결을 해지한 상태로 있다.	
removeReaderDevice()	RPManager에 연결된 리더를 제거	

표 19 RPManagerBean 주요 메소드

NotificationChannelBean : NotificationChannel를 Bean객체로 만들기 위해 생성되어진 인터페이스

주요Method	설명	비고
Start()	생성되어진 notificationChannel Object를 활성화 시킨다. 이 상태에서는 ReaderDevice에 연결을 할 수 있는 상태를 의미한다.	
Stop()	생성되어진 notificationChannel Object를 비활성화 시킨다. 만약 활성화 상태에서 notificationChannel 이 사용중에 있다면 그 작업에 Interrupt를 걸어 중지 시키고 다시 start가 호출될때까지 대기한다.	

표 20 NotificationChannelBean 주요 메소드

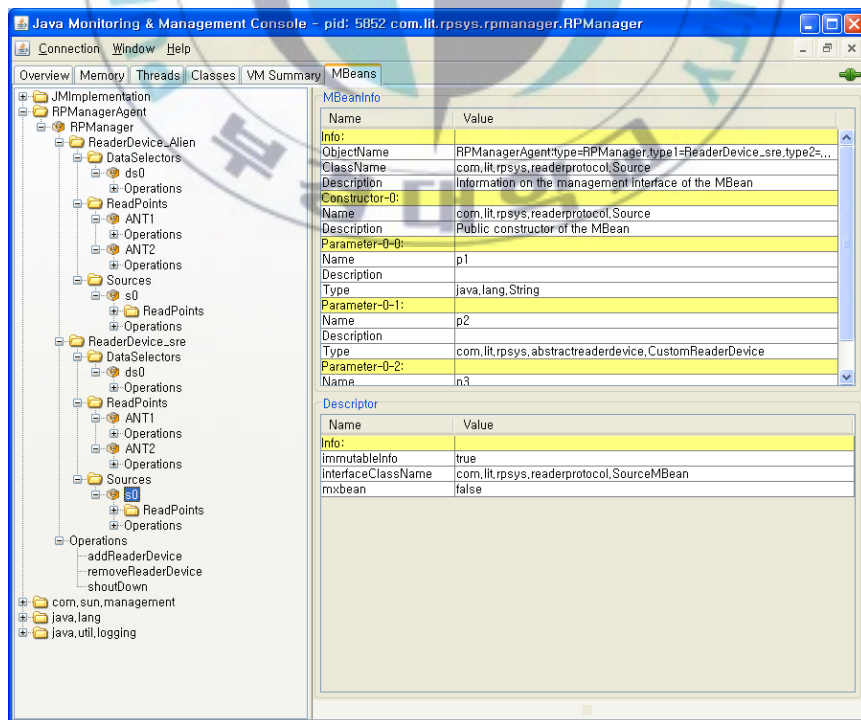


그림 14 RPManager의 Jconsole 구동

위 [그림 14]은 실제로 JMX기술을 RPManager에 접목 시켜 구현하여 Jconsole로 구동한 그림이다. 그림에서 보듯이 RPManager의 각 주요 클래스들을 Bean으로 구성하여 Mbean Server에서 생성된 Bean들을 관리할 수 있는걸 확인할 수 있다. 그리고 RPManager의 각 클래스안의 주요 기능을 하는 함수들을 Bean으로 관리하여 Jconsole 및 Web으로 연동을 가능하게 하여 RPManager를 좀 더 효율적으로 관리하고 사용할 수 있는 환경을 제공해 주고 있다. 그러나 이런 기능을 활용하기 위해서는 JMX기술을 이용하여 그에 맞는 설계를 해야 한다.

3.6 RPManager의 Message 생성 구조

RPManager는 RP v1.1과 RM v1.0.1에서 사용되는 메시지 400여가지를 지원하며, 해당 메시지를 생성하여 통신을 할 수 있는 구조로 이루어져있다. RP 및 RM 메시지를 생성할 때에는 각 객체에 와 쌍이 맞는 MshMaker클래스를 이용하여 메시지를 생성하며, 이 클래스는 XML메이커를 이용하여 해당 XML메시지를 손쉽게 생성하는 방법을 사용하고 있다.

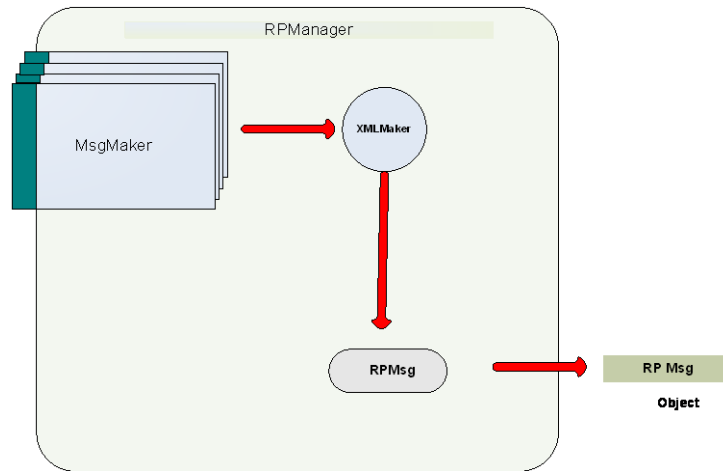


그림 15 RP RM Message 생성

위 [그림 15]는 해당 클래스의 MsgMaker클래스와 XMLMaker클래스를 이용하여 메시지를 생성하며, ReaderDevice에 관련 메시지를 생성할 때에는 ReaderDeviceMsgMaker클래스를 Source관련 메시지는 SourceMsgMaker클래스를 이용하여 생성하는 구조를 나타내고 있다. 물론 방금 명시된 클래스 이외에도 해당 클래스의 MsgMaker클래스를 이용하여 여러 메시지를 생성할 수 있는 구조로 이루어져 있다.

3.7 RPManager의 설정 파일 관련 구조

RPManager는 한 프로세서에서 하나의 비표준 리더를 지원하는 것이 아닌, 1:N의 Architecture로 이루어져 있다 자세히 말하자면 하나의 RPManager가 구동이 되고 있다면 이 RPManager를 이용해서 동시에 여러대의 비표준 리더를 EPCglobal의 표준을 지원하는 리더로 만들 수

있다. 이러한 기능을 수행 하기 위해서 각 리더에 맞는 설정파일이 필요 하며 이러한 설정파일들은 RPManager에서 사용할 수 있게끔 하기 위해 요구하는 양식이 따로 존재한다.

하나의 프로세서로 동작하는 RPManager가 여러대의 비표준 리더를 지원하여 표준 리더로 동작하게 할려면 해당 물리적인 리더의 정보를 받아올 수 있는 방안이 필요하다. 이런 문제를 해결하기 위해 RPManager 는 리더에 관한 정보를 Config파일로 남겨서 이걸 토대로 동작을 하고 있다. 현재 Config파일은 XML형식으로 이루어져 있으며 그 구조는 리더의 IP와 Port부터 시작해서 리더의 고유 정보를 담아 활용 할 수 있는 구조로 이루어져 있다.

```
<ReaderData>
  <CustomReaderClass>CustomAdapterClass</CustomReaderClass>
<CustomReaderAddress>RPmanagerIP</CustomReaderAddress>
  <CustomReaderPort>RPManagerPort</CustomReaderPort>
  <UserID>ReaderID</UserID>
  <Password>Readerpassword</Password>
  <ReaderControlPort>ReaderPort</ReaderControlPort>
  <ReaderEPC>ReaderEPC</ReaderEPC>

<AvailableReaderPointCount>ReaderAntenna</AvailableReaderPointCount>
  <ReaderSerialNumber>ReaderSerial</ReaderSerialNumber>
</ReaderData>
```

표 21 ReaderData.Xml 의 구성

위 [표 21]에서 설정한 각각의 파라메터 값에대한 설명은 다음 표와

같다. 실제 리더가 연결될 때 꼭 필요한 정보들은 아래에 나열된 것만으로도 구동이 되지만, 필요 여부에 따라 혹은 프로그램의 업데이트에 따라 항목이 추가/삭제가 이루어 질 수도 있다.

아래 [표 22]는 위 표에서 구성되어있는 설정 항목들에 대한 설명을 나타낸 표이다.

항목	설명	비고
ReaderData	리더마다 구분짓는 파라메타	한 리더에 한 항목씩 존재
Custom Reader Class	물리적인 리더의 고유 CustomAdpater Class	
Custom Reader Address	물리적인 리더에 실제 할당된 실제 IP주소	
Custom Reader Port	물리적인 리더에 할당된 실제 Port를 의미함.	
UserID	물리적인 리더에 접근할 때 쓰이는 ID	실제 리더에서 쓰이지 않을 경우 공란으로 둠
Password	물리적인 리더에 접근할 때 쓰이는 Password	실제 리더에서 쓰이지 않을 경우 공란으로 둠
Reader Control Port	RPManager에서 쓰이는 Port	
ReaderEPC	리더의 EPC값	
AvailableReader PointCount	물리적인 리더에 연결된 안테나 수	
ReaderSerial Number	물리적인 리더에 할당된 실제 시리얼 번호	

표 22 Config XML 항목별 설명

설정파일의 내용은 기본적으로 RPManager를 처음 구동할시 연결된

물리적인 리더의 내용을 가져오거나 구동 중 필요 정보를 요청할시 설정 파일을 참조하여 리턴하는 형식을 띄고 있는데, 이런 용도 말고도 현재 RManager는 1:N 구동을 위해서도 설정파일을 쓰고있다.

1:N의 구동절차는 사전에 설정파일을 읽어 ReaderData의 Node를 체크하여 RManager에 연결된 2대 이상의 리더의 ReaderDevice와 CustomAdapter를 생성하게 된다. 이렇게 설정파일을 바탕으로 새로 생성 되어진 ReaderDevice와 CustomAdapter는 다수의 비표준 리더들을 표준리더로 동작할 수 있게끔 할 수 있다.

3.8 구현 환경

RManager의 전체적으로 완성된 구성도는 [3.1 RManager의] 장에서 다루었던 [오류! 참조 원본을 찾을 수 없습니다.]과 같이 구성되어 상위 애플리케이션인 미들웨어와 통신을 하며, 더 나아가서는 Web으로 RManager를 접근하여, RManager에 연결된 물리적인 리더의 정보 및 현재 RManager의 모니터링 과 관리를 할 수 있다.

구현물의 전체적인 흐름은 비표준을 지원하는 리더는 RManager의 해당 리더에 맞게 구현 CustomAdapter를 이용하여 RPmanager와 접속을 한다. 그런다음 RManager는 상위 미들웨어와 연결하여 RP v1.1와 RMP v1.0.1에서 정의한 메시지로 통신을 하면서 그에 해당하는 기능을 수행하게 된다. 그리고 사용자에게 제공될 UI는 따로 없으며, RManager개발과 동시에 개발된 추가로 개발될 Web Admin Tool을 이용하여 RManager를 관리 및 모니터링을 하면 된다.

아래[표 23]는 본 논문에서 사용된 컴퓨터의 하드웨어적 시스템 환경과 구현에 사용한 언어에 대해서 나타나있다.

구 분	명 칭	버 전	제공회사	비 고
O S	Window XP	XP SP3	Microsoft	
Language	JDK	6.0	SUN	
Message Language	XML	1.0	W3C	
Program tool	Eclipse	3.3	Eclipse	

표 23 시스템 구현 환경



4. 구현 및 결과

4장에서는 앞서 3장에서 설계한대로 구현한 RPManager를 가지고 구현 결과를 제시하고, 평가하며, 향후 개선해야 하는 사항에 대해 논하고자 한다.

4.1 구현 테스트에 사용될 하드웨어 환경

구현 테스트에 사용된 리더의 하드웨어 사양 및 테스트 환경은 다음 [표 24]와 같다.

명 칭	제조회사	비 고
ARL- 9780	Alien	900Mhz Reader
ARL- 9800	Alien	900Mhz Reader
ARL- 9900	Alien	900Mhz Reader
Aline Gen2 Tag	Alien	Gen2 Tag
IF4 Reader	Intermec	900Mhz Reader

표 24 사용되어진 Reader의 하드웨어 환경

4.2 구현 결과 및 동작 Sequence

RPManager에서 비표준 리더를 사용할 때에는 해당 리더에 맞는 CustomAdapter를 제작해서 사용해야 한다. CustomAdapter는 RPManager내에서 필수 구현 API를 제공해주므로, 최소한의 구현만을

원한다면 API에서 요구하는 클래스만 상속을 받아 구현을 하면 되고, 그 비표준 리더만의 고유 기능을 추가하고 싶을 때 에는 해당 리더의 패키지 안에서 클래스를 따로 구현을 해주면 된다.

그리고 이렇게 구현된 CustomAdapter를 적용하여 RManager를 구동한다.

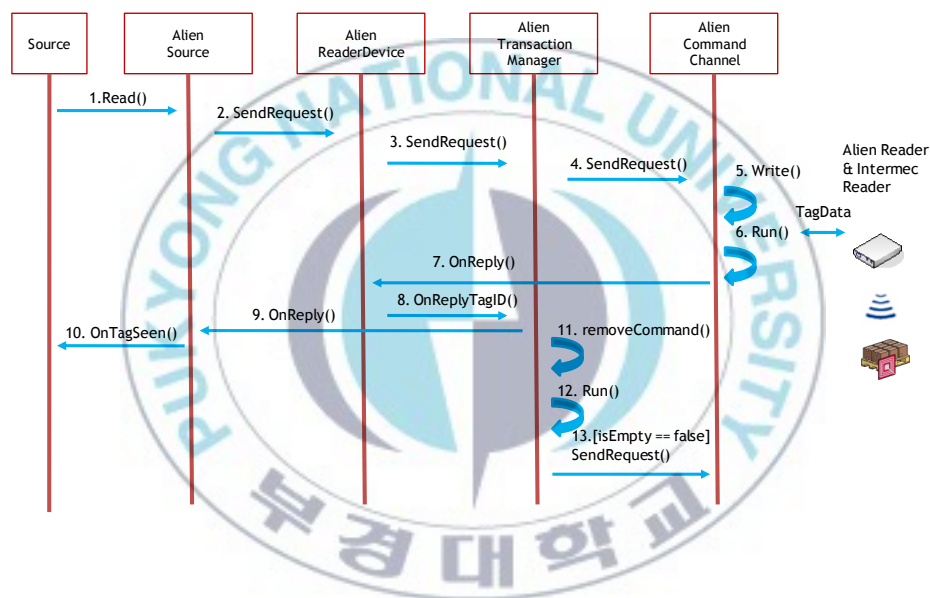


그림 16 CustomAdapter 동작 Sequence

위[그림 16]은 처음 RManager를 구동시켰을 때 콘솔창이 기본적으로 하나 구동이된다. 여기에는 연결된 리더의 초기화를 해주는 일과 리더에서 읽어오는 태그 데이터를 실시간으로 확인할 수 있다. 또한 처음 연결 시에는 이 화면에 해당 리더의 초기화 작업을 하는걸 확인할 수 있

었다. 그리고 연결된 리더에서 읽어 들인 태그의 값을 EPCglobal에서 정의한 RP v1.1 , RMP v1.0.1에서 사용되어지는 메시지로 만들어서 해당 태그의 값에 맞는 이벤트를 할당하고, 설정된 DataSelector의 필드값을 채워서 상위로 올려주는걸 확인 할 수 있었다.

그리고 RPManager에 두대의 비표준 리더를 연결하여 연동 테스트 또한 시도해 보았다. 이때 1:N 테스트에 사용된 리더는 Alien사의 ARL-9800 리더와 ARL-9900을 이용해서 테스트를 진행하였다.

진행 결과 두 대의 리더 각각에서 태그 데이터가 올라오는 것을 확인할 수 있었고, RPManager 또한 문제 없이 두 리더를 지원하는 것을 확인할 수 있었다.

4.2.1 Alien Reader CustomAdapter 구현

아래 [그림 17]는 Alien회사의 리더들을 RPManager에서 사용할 수 있게끔하기 위해 구현한 CustomAdapter의 클래스 다이어그램이다. 각각 해당 클래스들은 Alien이라는 접두어가 붙어서 구현이되고, 상위 Abstract클래스를 상속받아서 구현을 하였다.

주요Method	설명	비고
Read()	물리적인 리더에게 태그를 읽으라고 명령을 전달하는 메소드	
Write()	물리적인 리더에게 태그를 쓰라고 명령을 전달하는 메소드	

표 26 AlienSource 주요 메소드

AlienNotificationChannel : Alien RFID Reader가 Connection Initialize 하여 CustomReaderDevice에게 Notify하는 Tag 정보를 받기위한 채널

주요Method	설명	비고
Run()	ServerSocket을 열고 Notify를 대기하는 메소드	

표 27 AlienNotificationChannel 주요 메소드

AlienCommandChannel : AlienCommandChannel은 CustomCommand Channel을 상속받아 정의된 리더와 직접 네트워크 연결하여 명령을 보내고, 리더를 제어한다.

주요Method	설명	비고
Run()	리더로부터 보내오는 명령의 응답 및 Tag 정보를 받아 처리하는 메소드	

표 28 AlienCommandChannel 주요 메소드

AlienTransactionManager : AlienReader로부터 오는 명령어들을 Transaction관리 하기 위해 생성된 클래스

주요Method	설명	비고
OnReply()	해당 명령어를 트랜잭션리스트 상에서 삭제하고	

	다음 명령어를 수행하도록 해주는 메소드	
--	-----------------------	--

표 29 AlienTransactionManager 주요 메소드

AlienCustomProtocol : CustomReaderProtocol에 정의된 Command Message를 Alien에 맞게 정의해주는 클래스로서, 리더에서 주요사용하는 고유 명령어들을 RPManager에서 손쉽게 사용할 수 있게 재정의할 수 있다.

4.2.2 Intermec Reader CustomAdapter 구현

아래 [그림 17]는 Intermec회사의 리더들을 RPManager에서 사용할 수 있게끔하기 위해 구현한 CustomAdapter의 클래스 다이어그램이다. 각각 해당 클래스들은 Alien이라는 접두어가 붙어서 구현이되고, 상위 Abstract클래스를 상속받아서 구현을 하였다.

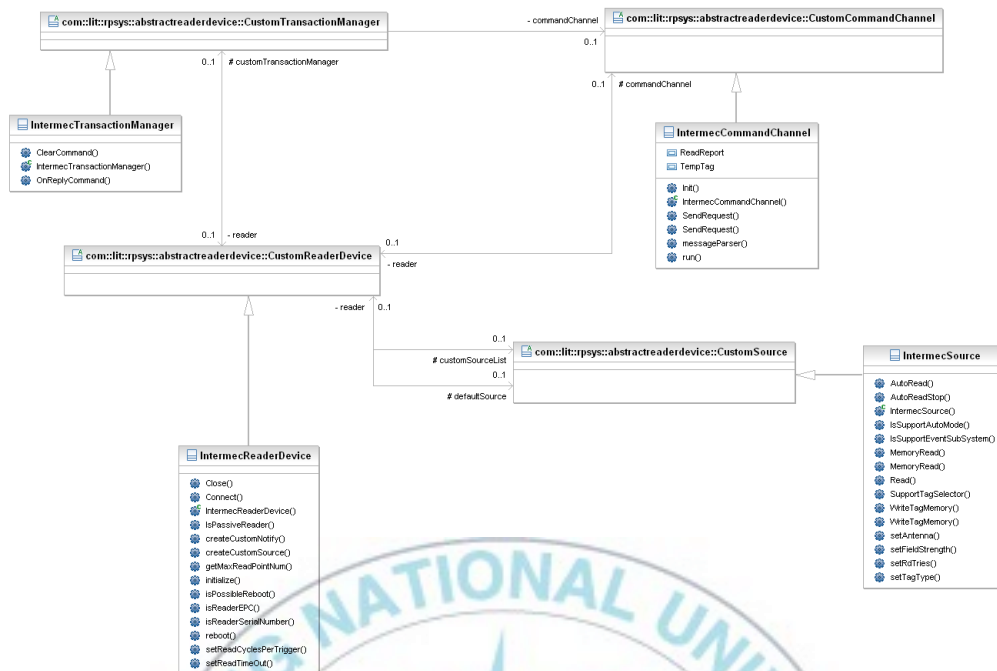


그림 18 IntermecReader의 CustomAdapter구현

IntermecRaederDevice : Intermec Reader가 Connection initialize하여 Notify하는 응답을 받을 수 있는 Intermec Source, Intermec CommandChannel을 생성하고, 관리하는 클래스

주요Method	설명	비고
initialize()	리더를 사용하기 위한 초기설정값을 CustomReaderProtocol을 기반으로하여 수행해 주는 메소드	
Connect()	실제 연결된 물리적인 리더에게 접속을 시켜주는 메소드	

표 30 IntermecRaederDevice 주요 메소드

IntermecSource: Intermec RFID Reader에게 READ, AUTOREAD와 같

은 Tag 수집 관련 명령을 처리하고, EPCglobal ReaderProtocol 1.1 표준 Source에게 읽은 DATA를 전달해주는 클래스

주요Method	설명	비고
Read()	물리적인 리더에게 태그를 읽으라고 명령을 전달하는 메소드	
Write()	물리적인 리더에게 태그를 쓰라고 명령을 전달하는 메소드	

표 31 IntermecSource 주요 메소드

IntermecCommandChannel : IntermecCommandChannel은 Custom Command Channel을 상속받아 정의된 리더와 직접 네트워크 연결하여 명령을 보내고, 리더를 제어한다.

주요Method	설명	비고
Run()	리더로부터 보내오는 명령의 응답 및 Tag 정보를 받아 처리하는 메소드	

표 32 IntermecCommandChannel 주요 메소드

IntermecTransactionManager : IntermecReader로부터 오는 명령어들을 Transaction관리 하기 위해 생성된 클래스

주요Method	설명	비고
OnReply()	해당 명령어를 트랜잭션리스트 상에서 삭제하고 다음 명령어를 수행하도록 해주는 메소드	

표 33 IntermecTransactionManager 주요 메소드

4.3 향후 개선 방안

RPManager 는 현재 Alien 사의 리더와 Intermec 사의 리더만을 지원하고 있다. 또한 Jconsole 을 통한 관리 및 모니터링을 제공해주고 있다. 그러나 추후 추가 개발을 통하여 Web 접근 통하여 RPManager 에 접근하고 해당 객체를 생성하고, 관리 및 모니터링을 Web Admin 을 통해서 확인 할 수 있도록 개발을 진행해 나갈 예정이다. 또한 RPManager 가 구동중에도 설정파일을 수정하여 리더를 추가 및 수정 할 수 있는 Hot-Swapping 기능을 추가 할 예정이다.

그리고 무엇보다 현재 RPManager 가 지원 할 수 있는 리더의 개체 수 를 늘리고, 900Hz 대역 뿐만 아니라 2.4GHz 대역의 리더 또한 지원 할 수 있게끔, CustomAdapter 의 API 확장 및 RPManager 의 기능 개선이 필요하다.

5. 결론

본 논문에서 구현한 RPManager는 비표준화된 리더에 대해 EPCglobal Reader Protocol v1.1과 Reader Management Protocol v1.0.1이라는 표준화된 접근방식을 제공해주고 있다. 기존의 리더들은 각 특정한 프로토콜을 이용하여 Middleware에 접근하게 되는데, 이는 한 회사가 하나의 Middleware를 쓰게 되면 별 문제가 없지만 여러 회사의 리더가 하나의 Middleware를 사용하게 되면 다른 특정한 프로토콜로 그 리더를 접근해야 하기 때문에 관리자, 사용자 측면에서 비효율적이라고 할 수 있다. 그리하여 본 논문에서 제안한대로 RPManager를 사용을 하면, 여러 특정한 프로토콜을 사용하더라도 하나의 표준 프로토콜인 EPCglobal Reader Protocol v1.1과 Reader Management Protocol을 사용하여 Middleware에 접근을 하기 때문에 유지 및 관리 입장에서 효율적이라고 할 수 있다. 또한 나아가서는 리더의 효율적인 관리를 위해 Web을 통한 리더의 추가 및 삭제 관리, 하나의 RPManager 구동으로 여러 대의 리더를 표준화된 접근 방식 제공, LLRP 리더지원 및 Sensor Device의 리더화 등을 제공하고 있어, 이 또한 표준화된 접근방법으로 접근하여 리더 및 Device들을 관리 할 수 있다.

현재의 RFID시장은 계속해서 여러 분야에 활용할 수 있는 리더 및 그에 해당하는 Middleware가 계속 해서 나오고 있다.

향후에는 RPManager의 다양한 타입의 리더 지원을 위한 Adapter연구 및 기능 개선에 대한 연구가 필요로 하다.

6. 참고 문헌

- [1]RFID 확산 전망 및 시사점 - 환경분석, 가격전망 및 정책적 시사점
- ,이은곤, 정보통신정책 제16권 13호 통권 351호 2004-07-16
- [2]최길영, 성낙선, 모희숙, 박찬원, 권성호 “RFID 기술 및 표준화 동향”,
전자통신동향 분석 제22권 제3호, pp. 29~37, 2007.
- [3]최성규, "RFID 산업 동향 및 전망", TTA저널 제 95호, pp. 48~54,
2004
- [4]EPCglobal, <http://www.epcglobalinc.org>
- [5]EPCglobal, Reader Protocol Standard Version 1.1 Ratified
Standard June 21, 2006
- [6]EPCglobal, Reader Management Protocol Standard Version 1.0.1
Ratified Standard May 31, 2007
- [7]안종민, 송하주, "이기종 리더의 통합 접속 및 관리를 위한
RPManager의 설계와 구현", 한국정보과학회 추계학술발표대회 논문
집 제35권 제2호 P92, 2008, 10
- [8]EPCglobal, Low Level Reader Protocol Standard Version 1.0
Ratified Standard April 12, 2007
- [9]EPCglobal, The EPCglobal Architecture Framework Final Version
1.2 Approved 10 September 2007
- [10]안재명, 이종태, 오해석, (주)리테일테크 기술연구소 공저 “EPCglobal
Network기반의 RFID 기술 및 활용”, 2007년 2월
- [11]이근호, 무선식별(RFID)기술, TTA저널, 제89호, 125-126
- [12]RFLink, <http://www.rflink.co.kr>

- [13]EPCglobal, The Application Level Events (ALE) Specification
Version 1.0 Ratified Specification September 15, 2005
- [14]EPCglobal, EPC Information Services (EPCIS) Version 1.0.1
Specification September 21, 2007
- [15]EPCglobal, Object Naming Service (ONS) Version 1.0 Ratified
Specification October 4, 2005
- [16]황재각, 정태수, 김영일, 이용준, "RFID 미들웨어 기술 동향 및 응용", 전자통신동향분석 제20권 제3호, 2005
- [17]빌 글로버, 히만슈 마트, "실무자를 위한 RFID 이해와 활용", 2007
년 2월
- [18]SUN,<http://kr.sun.com/developers>



감사의 글

"Don't think you are know you are" 대학원 석사 과정을 입학한 이래 벌써 2년의 시간이 흘렀습니다. 2년동안 석사과정을 수학하면서 배운 여러 경험 및 지식, 그리고 알게된 사람들은 제 인생에 값진 보물이었던 것 같습니다. 먼저 이런 값진 경험과 부족한 저를 지도해주시고, 옳은길로 인도해주신 송하주 교수님에게 진심으로 감사드립니다. 교수님 아래서 배운것들은 어디서도 배울 수 없는 제 인생에 가장 큰 윤활유였던 것 같습니다. 그리고 부족한 저의 논문을 심사하여 주신 김종남 교수님, 신봉기 교수님께도 진심으로 감사드립니다. 또한 석사학위 과정 동안 저에게 많은 가르침을 주신 여러 교수님들에게도 감사드립니다.

석사과정을 지낸 2년동안 연구실 생활을 하면서 석사 생활에 많이 도움을 준 성진이, 태용이, 승혁이, 영민이, 영준이, 주형이와 같이 동기로서 연구실로 들어와 제게 많은 조언을 해준 화춘이형과 언제나 늘 저에게 목표를 제공해주시고 때로는 따끔한 충고와, 나아갈 방향을 제시해준 명환이형님, 태순이형님, 영규형님에게 감사한 마음을 전합니다. 그리고 저와 같이 동기로서 대학원에 입학해 많은 도움을주신 홍석이형, 원희형, 무경이형, 언제나 항상 응원을 해준 10여년 동안은 같이 알고 지내온 태봉, 승용, 승대 그리고 그 밖의 친구들에게 감사의 마음을 전합니다.

또한 항상 내 곁에서 나의 반쪽이 되어 응원을 해주고 힘이 되어준 은영이에게도 감사의 마음을 전합니다.

마지막으로 제가 이 길로 갈수 있게끔 절 이끌어주시고, 지켜봐주시며, 항상 격려해주신 어머니와 지금 하늘에 계신 아버지에게 이 말을 꼭 전해주고 싶습니다. "어머니 아버지 사랑합니다."

2009년 2월 안중민 드림