LLRP를 지원하는 R1000/R2000 겸용 RFID 리더

정회원 배 성 우*, 준회원 유 원 상*, 곽 호 길**, 정 명 섭**,
종신회원 박 준 석*, 정회원 성 영 락*, 오 하 령*

Design and Implementation of a R1000/R2000 based RFID Reader Which Supports the Low Level Reader Protocol

Sung-Woo Bae* *Regular Member*, Won-Sang Ryu*, Ho-Gil Kwak**,
Sub-Myoung Joung** *Associate Members*, Jun-Seok Park* *Lifelong Member*,
Yeong-Rak Seong*, Ha-Ryoung Oh* *Regular Members*

요 약

RFID 리더 프로토콜은 RFID 리더와 미들웨어, 애플리케이션 등의 상위 호스트 사이의 인터페이스이다. 현재의 리더 프로토콜은 리더 제조업체별로 상이하여 이기종의 리더들을 사용하는 환경에서는 리더들 간의 호환성 문제가 있다. 본 논문에서는 이와 같은 문제를 해결하기 위해 EPCglobal의 LLRP(Low Level Reader Protocol)을 지원하는 리더를 설계하고 구현하였다. 또한, 다양한 응용분야에 적용하기 위해 리더를 두 개의 모듈로 나누어 설계하였고, 다양한 인터페이스를 지원하도록 설계하였다. LLRP는 임베디드 리눅스 환경에서 멀티 쓰레드를 이용해 구현하였으며, LLRP의 대부분의 기능을 지원하고 다양한 요구에 맞출 수 있도록 유연한 하드웨어와 소프트웨어 구조로 설계하였다.

Key Words : RFID, LLRP(Low Level Reader Protocol), reader protocol

ABSTRACT

RFID reader protocol is an interface between RFID readers and higher (host) such as RFID middlewares and applications. At present, reader protocols provided by vendors are different from each other and there are compatibility problems in environment using heterogeneous readers . In this paper, to solve this problem, an RFID reader which supports LLRP(Low Level Reader Protocol), a well-known standard reader protocol presented by EPCglobal is designed and implemented. It is designed with two modules and supports various interfaces for easy adaptation to various applications. The LLRP protocol is implemented over a embedded LINUX multi-thread environment. It not only supports almost all properties of LLRP, and is designed with flexible hardware/software architecture to meet various requirements.

I. 서 론

RFID 시스템은 개별 물체를 식별하기 위한 태그, 태그를 인식하기 위한 리더, 인식된 정보를 처리하는 상위 호스트(애플리케이션 또는 미들웨어) 시스템으로 구성된다. 태그와 리더는 RFID 무선 인터페이스(air interface) 규격에 따라 통신이 이루어지며 리더와 상위 호스트와는 무선 또는 유선으로 통신

※ 본 연구는 지식경제부 IT산업원천기술개발사업(2008-F-050-02, 자기유지 WBAN/USN용 u-Scavenging 기술개발)과 구원장학재단 연구비지원에 의해 수행되었습니다.

* 국민대학교 전자공학과(hroh@kookmin.ac.kr), ** 국민대학교 전자정보 통신공학부

논문번호 : KICS2009-12-620, 접수일자 : 2009년 12월 14일, 최종논문접수일자 : 2010년 2월 11일

이 이루어진다. 그런데 현재까지 개발된 대부분의 RFID 리더들은 각 제조업체 별로 독자적으로 개발된 API(Application Programming Interface)를 사용하여 호스트와 통신하고 있기 때문에, 다수의 이기종 리더들을 사용하는 환경에서는 리더들 간의 상호 호환성이 문제로 대두된다. 이처럼 종류가 서로 다른 RFID 리더들을 같이 운용하기 위해서는 공통 인터페이스 기능을 제공하여 제조사 및 제품별 특성에 상관없이 일관된 방법으로 상호작용이 가능하도록 해야 한다. 이와 같은 문제를 해결하기 위하여 EPCglobal에서 리더 프로토콜(reader protocol)을 제안하고 규격화하였다. 리더 프로토콜은 임의의 RFID 장치에 대해 논리적으로 동일한 인터페이스를 정의하여 RFID 리더의 저수준(low level) 기술 특성에 독립적인 유연한 시스템 구성을 이루어 이기종의 리더간의 API 통합이 어려운 문제점을 해결할 수 있다[1].

본 논문에서는 EPCglobal에서 제정한 표준 리더 프로토콜들 중의 하나인 LLRP(Low Level Reader Protocol)[2]를 지원하는 RFID 리더를 설계하고 구현하였다. LLRP는 리더에게 논리적으로 동일한 인터페이스를 가지며 다른 리더 프로토콜과 다르게 RFID 리더가 사용하는 무선 인터페이스를 직접 제어할 수 있는 특징을 갖고 있다. 리더는 사용 환경, 가격, 시스템 성능 등의 요구사항에 따라 여러 종류의 통신 모듈을 혼용할 수 있도록 설계하였으며 현재 산업현장에서 많이 사용하는 Impinj사의 R1000과 R2000 칩[3]을 혼용하여 사용할 수 있도록 개발하였고, LLRP를 지원하는 RFID 리더 소프트웨어를 임베디드 리눅스 환경에서 설계하고 구현하였다. 임베디드 리눅스는 RFID 리더뿐만이 아니라 많은 임베디드 장치에서 사용하는 플랫폼으로 최근 네트워크 환경에 다양하게 요구되는 RFID 리더의 능력을 수용할 수 있고 기능 확장에 용이한 장점이 있다.

본 논문의 구성은 다음과 같다. 2장 관련연구에서는 리더 칩과 리더 프로토콜에 대해 설명한다. 3장에서는 본 논문에서 설계한 리더와 LLRP에 대해 소개하고 4장에서는 실험에 의해 동작을 검증하고 분석하였으며 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 관련연구

2.1 Impinj R1000과 R2000

Impinj사의 R1000과 R2000은 기존 리더 구성품의 90% 정도를 하나의 칩으로 만들어 리더의 크기를 최소화하여 개발할 수 있게 하였고, 리더를 설계하는 복잡도를 줄였으며 리더 개발에 필요한 비용을 줄일 수 있게 하였다. 그리고 EPCglobal UHF Generation 2 또는 ISO/IEC 18000-6 Type C 의 규격을 지원하며, DRM(Dense Reader Mode)을 지원한다. 또한, 짧은 거리의 임베디드형 모듈에서부터 긴 거리의 고정형 리더기까지 사용자 요구에 따라 다양한 형태의 리더를 개발하는데 사용할 수 있다.

R2000은 기존의 R1000의 성능을 개선한 칩으로 태그의 신호를 분석할 때 리더가 보낸 신호의 캐리어 제거(carrier cancellation) 기술을 통해 인식 거리와 정확도를 개선시켰고 다양한 규격의 요구사항을 지원하기 위해 송신 위상 잡음(transmit phase noise)을 개량시켰으며, 다른 프로토콜을 지원할 수 있는 기능을 추가시켰다. 밑의 표 1은 R1000과 R2000의 특성을 나타낸다.

표 1. Impinj R1000과 R2000 특성

칩		R1000	R2000
지원 규격		ISO 18000-6 Type C DSB, SSB, PRASK DRM 지원	ISO 18000-6 Type C DSB, SSB, PRASK DRM 지원 다른 프로토콜 지원 가능
모뎀		Configurable digital baseband	
동작 주파수		840-960MHz	
센시티비티	LBT	-110dBm	
	DRM	-95dBm	
		-70dBm (10dBm carrier, Rx port)	-82dBm (10dBm carrier, Rx port)
송신 위상잡음 (at 250KHz offset)		-116dBm/Hz	-126dBm/Hz
적용 가능 지역		US FCC, ETSI EN302 208, Korea, Japan, China, India	

2.2 Low Level Reader Protocol

이 절에서는 EPCglobal에서 제안하는 리더 프로토콜인 LLRP 표준에 대해 소개한다. 위의 서론에서 언급했듯이 현재의 리더들은 제조사마다 각기 다른 API를 사용하여 다양한 종류의 리더들을 운용하는 환경에서는 리더들을 제어하는데 많은 어려움이 따른다. 이러한 어려움을 해결하기 위하여 EPCglobal은 개별 물체의 식별자인 EPC(Electronic Product Code)를 기반으로 하여 EPCglobal 아키텍

처 프레임워크(Architecture Framework)를 정의하였다. 아래의 그림 1은 EPCglobal 아키텍처 프레임워크 구조를 나타낸다.

EPCglobal 아키텍처 프레임워크는 구현에 따른 기술요소 분야를 하드웨어, 소프트웨어, 비즈니스 분야로 나누어 기술규격을 정의하고 있다. 이들은 각각 Gen 2 Air Interface, Reader Protocol (LLRP/RP), ALE(Application Level Events), EPCIS(EPC Information Service)으로 구성된다[4]. 이 외에 EPCIS에 접근하여 해당 정보를 얻을 수 있는 ONS(Object Naming Service)에 대해서도 정의하고 있다.

EPGglobal 아키텍처 프레임워크 중 우리가 관심을 가지는 분야는 리더와 상위 호스트와의 인터페이스를 정의하고 있는 LLRP와 RP이다. LLRP와 RP의 가장 큰 차이점은 상위 호스트에서 리더의 설정 지원 유무와 RFID 리더가 수집한 태그의 정제 등의 수집한 태그의 정보 가공 지원 유무에 있다[5]. 따라서 LLRP는 RFID 무선 인터페이스의 이해를 기반으로 RFID 리더의 주파수 특성을 고려하여 RFID 간섭을 최소화하고 리더의 효율을 극대화하는 것을 가능하게 한다.

LLRP는 위의 그림 2와 같이 상위 계층을 클라이언트(client)라고 부르며 리더와 클라이언트 사이의 통신 포맷과 절차를 규정한다. 통신을 위한 기본 데이터 단위는 메시지(message)라고 부르는 단위를 사용하며, 한 가지 메시지를 제외한 모든 메시지는

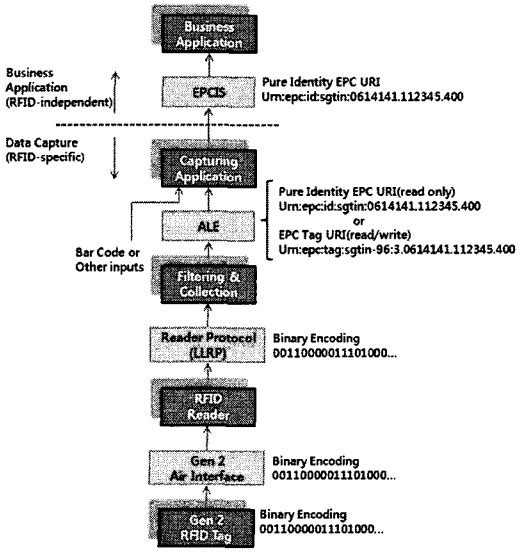

그림 1. EPCglobal 아키텍처 프레임워크

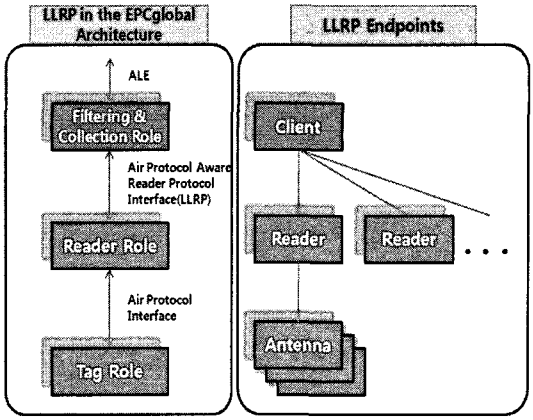

그림 2. EPCglobal 아키텍처 프레임워크내의 LLRP 역할

응답(Response) 메시지를 필요로 한다. 그리고 LLRP에서는 **Spec**이라고 부르는 단위로 세부적인 동작을 구성하고 설명한다. **ROSpec**(Reader Operation Spec), **AISpec**(Antenna Inventory Spec)등 다양한 Spec이 사용되며, 각각 Spec들의 병렬적인 동작으로 LLRP가 동작하게 된다. 본 논문에서 자세하게 소개하지 못하는 스펙들에 대한 설명은 프로토콜 문서를 참고하기 바란다[2]. LLRP는 다음과 같은 특징을 갖는다.

- 리더 제어에 필요한 모든 매개 변수를 설정
- 리더의 주파수 사용에 대한 측량(Survey) 기능을 제공
- 리더의 하드웨어 또는 펌웨어 능력의 정보를 사용자에게 제공
- 무선 인터페이스에 따라 추가적인 확장이 용이
- 사용자에 의한 변수 확장을 지원

Ⅲ. 설계 및 구현

현재의 리더들은 거듭된 연구로 성능은 비슷해졌으며 각각의 응용분야에 따라 특성화되어가고 있다. 이렇게 각 분야에 특성화되어 가고 있는 근래의 리더에 대한 요구는 리더의 기본적인 성능과 더불어 다양한 응용분야에 적용할 수 있는 리더를 원하고 있다. 본 논문에서는 이러한 요구를 수용하기 위하여 EPCglobal의 LLRP 규격을 따르고 다양한 환경의 인터페이스를 지원할 수 있는 리더를 개발하였다. 또한, 기존의 복잡한 리더 구성품들을 기능에 따라 두 개의 모듈로 구성하여 유연성과 확장성을 확보하였다.

3.1 개발 리더의 구조

본 논문에서 개발한 리더는 아래의 그림 3과 같이 통신 모듈과 제어 모듈, 두 개의 모듈로 나누어 개발하였다. 이는 실시간 보장을 요구하는 RFID 무선 인터페이스 프로토콜을 지원하는 부분과 상위 프로토콜을 지원하는 부분을 물리적으로 분리하여 리더의 확장성 및 효율성을 높이고 다양한 인터페이스 환경에 적용할 수 있도록 하였다. 현재 Impinj 사의 R1000과 R2000 리더 칩을 겸용할 수 있도록 각각의 통신 모듈로 개발하였으나, 추후 다양한 성능, 가격 등 사용자 요구에 따라 적절한 모듈을 추가로 개발하여 사용할 수도 있다. 두 모듈 사이의 인터페이스는 일반적으로 많이 사용하는 UART, USB 등으로 구성하여 제어 모듈의 간단한 소프트웨어의 수정만으로 다양한 통신 모듈을 사용할 수 있도록 확장성을 고려하여 설계하였다. 그림 4는 개발된 리더의 외관이다.

통신 모듈은 리더에서 실제 신호를 처리하는 부분을 담당하고 단독 모듈로도 사용할 수 있도록 개발하였으며, 리더 칩, 파워 앰프 칩, Atmel 사의 MCU, 안테나, 그리고 외부 통신을 위한 인터페이스 등으로 구성된다. 리더 칩은 Impinj 사의 R1000과 R2000 칩, 두 가지의 칩을 사용하여 사용자가 모듈을 선택할 수 있도록 개발하였다. 파워 앰프는 리더 칩에서 나오는 신호의 출력을 높이는 역할을 하고, Atmel 사의 MCU는 ARM7 코어가 내장된 칩으로 리더 칩을 제어하는 MAC 부분을 담당하며, 추후의 확장성과 실시간성을 보장하기 위하여 별도의 MCU를 사용하고 있다. 외부 인터페이스는 통신 모듈의 활용도를 높이기 위하여 USB, UART, Bluetooth 등의 다양한 인터페이스를 이용할 수 있도록 설계하였다. 제어 모듈은 통신 모듈을 제어하고 상위 호스트와의 인터페이스를 담당한다. 그리고 본 논문에서 구현한 LLRP가 제어 모듈에 이식된다. 삼성의 S3C6410 ARM MCU를 이용하여 개발하였고 임베디드 리눅스를 이식하여 모듈의 활용도를 높였으며 USB, UART, Bluetooth, 유무선 LAN 등의 다양한 인터페이스를 적용할 수 있도록 개발하였다. 또한, 네트워크를 통해 내부 소프트웨어를 업그레이드 할 수 있도록 하였다. 그리고 모듈들을 제어하고 리더 동작을 검증하기 위하여 별도의 UI(User Interface) 프로그램을 제작하여 테스트하였다. UI프로그램은 마이크로소프트 닷넷 프레임워크 환경에서 C# 언어를 사용하여 구현하였다.

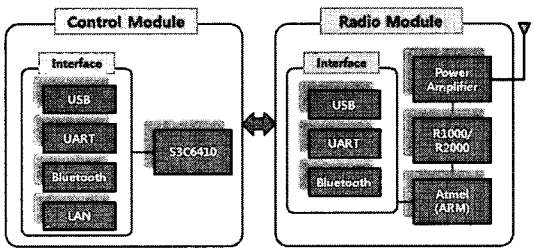

그림 3. 개발 리더 구성도

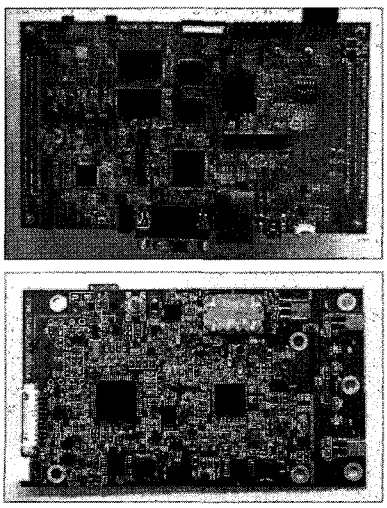

그림 4. 개발 리더 하드웨어

3.2 소프트웨어 개발 환경

제어 모듈은 S3C6410 MCU을 이용하여 임베디드 리눅스 기반으로 설계하였다. 따라서 제어 모듈의 소프트웨어를 개발하기 위하여 크로스 컴파일러 환경이 필요하다. 밑의 표 2는 제어 모듈에서 사용한 소프트웨어 개발환경을 나타낸다.

표 2. 소프트웨어 개발 환경

구 분	세 부 사 항	
프로세서	S3C6410	
메모리	RAM	EDD10323BB(128MB)
	ROM	K9F2G08U0B (256MB, NAND flash)
부트로더	U-Boot 1.3.4	
운영체제	Linux 2.6.29	
통신환경	UART, USB, Bluetooth LAN, WLAN	
플랫폼	CentOS	
툴체인	gcc 4.3.3	

운영체제로 리눅스 커널 2.6.29 버전을 사용하였으며 CentOS 플랫폼에 gcc 4.3.3을 이용하였다. 제어 모듈 내에는 저장할 수 있는 메모리 용량이 부족하여 제어 모듈의 파일 시스템을 호스트 시스템과 TCP/IP로 연결하여 네트워크 파일 시스템(NFS : Network File System)을 구축하여 개발하였다.

3.3 LLRP 구현

본 논문에서 구현한 LLRP를 전부 살펴보기에는 경우의 수가 너무 많기 때문에 LLRP의 대표적인 동작을 나타내는 ROSpec에 대해서 설명한다. LLRP에서 ROSpec은 리더 한 번의 동작과 유사하며 가장 커다란 동작의 단위로 볼 수 있다.

RFID 리더가 LLRP를 제대로 지원하기 위해서는 다수의 작업들을 동시에 병행해서 처리할 수 있어야 한다. 그러므로 리더 소프트웨어는 여러 프로세스 혹은 여러 쓰레드로 나뉘어 구성되는 것이 바람직하다. 본 논문에서는 프로세스 간의 동기화, 자원 공유, 에이징(aging) 문제 등을 감안하여 리눅스의 다중 쓰레드로 리더 소프트웨어를 구현하였다.

그림 5는 본 논문에서 구현한 LLRP의 쓰레드들의 통신 구조를 나타낸다. 총 7개의 쓰레드로 나누어 구현하였으며 각 쓰레드들의 역할은 다음과 같다.

RECV : 상위 호스트와의 수신 처리
SEND : 상위 호스트와의 송신 처리
MANAGER : LLRP의 전반적인 동작 관리
HANDLER/uHANDLER : 무선 인터페이스와 관련한 동작 처리
TRIGGER : LLRP의 동작을 처리하기 위한 시간 관리
REPORT : 동작 결과 처리

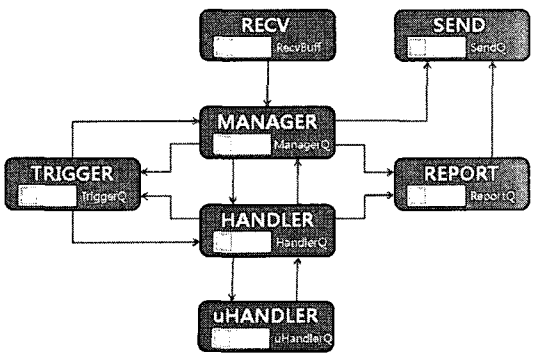

그림 5. LLRP 쓰레드 구조

쓰레드간의 통신은 메시지 큐를 이용하여 구현하였다. RECV를 제외한 모든 쓰레드는 각각 자신만의 메시지 큐를 하나씩 가지고 있으며, 그 메시지 큐를 통해서 전달되는 명령들을 처리하는 방식으로 동작한다. 그러므로 만약 입력 메시지 큐가 비어 있으면 그 프로세스의 동작은 봉쇄(blocking)된다. 이와는 달된다RECV 쓰레드는 통신 인터페이스를 통해 전달되는 메시지에 의해서 동작이 활성화되며 해당 통신 인터페이스에 메시지가 없으면 동작이 봉쇄된다.

본 논문에서 제안하는 쓰레드 구조는 다음과 같은 장점을 가진다. 첫 번째는 RECV 쓰레드와 SEND 쓰레드를 사용한 송수신 계층의 분리를 통하여 상위 계층과의 통신 인터페이스의 변경이 용이하다. 다른 쓰레드와 별개로 RECV 쓰레드와 SEND 쓰레드의 송수신 부분만 변경하면 상위 계층과의 다양한 통신 인터페이스를 적용할 수 있다. 두 번째는 LLRP에서 제안하는 무선 인터페이스 표준 확장이 용이하다는 장점이 있다. LLRP에서는 무선 인터페이스 표준에 관련된 부분을 추가 확장이 가능하도록 정의하고 있다. 본 논문에서 제안하는 쓰레드 구조는 HANDLER와 uHANDLER 쓰레드에서 무선 인터페이스의 처리를 담당하므로, 추가적인 무선 인터페이스에 맞추어 새로운 쓰레드들을 확장하면 기존의 구조와 함께 적용하기 쉬운 장점이 있다. 세 번째는 LLRP를 사용하지 않는 미들웨어를 위한 펌웨어의 수정이 용이하다는 장점이 있다. 실제 RFID 리더의 동작으로 나타내는 HANDLER 쓰레드와 uHANDLER 쓰레드를 이용하면 LLRP와는 다른 프로토콜을 구성하는데도 용이하다는 장점이 있다. 마지막으로 TRIGGER 쓰레드에서 단독적으로 시간에 관련된 처리를 담당하여 이를 이벤트화 해줌으로써 다른 쓰레드들이 사건 처리식(event driven) 방식으로 동작하도록 해주며 시간적인 오차를 줄일 수 있다는 장점이 있다. 문맥 전환(Context Switching) 시간, 혹은 처리 시간 때문에 정밀한 시간 측정이 어려운 단점을 하나의 쓰레드에서 시간에 관련된 처리를 함으로써 보다 정확한 시간 측정이 가능하다. 또한 uHANDLER 쓰레드를 통하여 통신 모듈과의 인터페이스를 지원함으로써 하위(내부) 프로토콜의 변경이 용이하다는 장점이 있으며 현재는 Impinj 사에서 제안한 프로토콜과 실험실 자체의 프로토콜 등을 구현하였다.

본 논문에서는 대표적으로 MANAGER 쓰레드에 대해서 설명한다. MANAGER 쓰레드는 LLRP를

지원하는 RFID 리더의 관리자와 같은 역할을 한다. LLRP의 모든 동작의 처리는 MANAGER 쓰레드에 의해 동작하며 ManagerQ에 의해 봉쇄되어 동작한다.

그림 6은 MANAGER 쓰레드의 의사코드(Pseudo-code)이다. MANAGER 쓰레드는 기본적으로 클라이언트로부터 수신한 메시지의 동작을 처리한다. 수신한 메시지에 대한 데이터 관리 및 실행 등의 동작을 수행하고, 클라이언트로 보낼 메시지가 필요한 경우 응답 메시지를 생성하여 SEND 쓰레드로 메시지를 전송한다. 그리고 리더의 동작을 실행하고자 할 경우 HANDLER 쓰레드에게 관련 정보를 메시지 큐를 통해 전송한다.

```
MANAGER thread()
{
  while(1) {
    msg = RecvWait(managerQ);
    switch (msg->type) {
      case LLRP_REQUEST: // from RECV
        response = ProcessRequest(msg->data);
        if (response != _NULL)
          Send(sendQ, LLRP_RESPONSE, response);
      case CLIENT_RESPONSE: // from RECV
        Send(handlerQ, CLIENT_RESPONSE, msg->data);
      case START_TRIGGER: // from TRIGGER
        StartROSpec(msg->data.RID);
      case STOP_TRIGGER: // from TRIGGER
        StopROSpec(msg->data.RID);
      case KEEPALIVE: // from TRIGGER
        if (NeedKeepAlive())
          Send(sendQ, KEEPALIVE, _NULL);
      case DONE_AIRFSPEC: // from HANDLER
        DoneAIRFSpec(msg->data);
      case CLIENT_REQUEST: // from HANDLER
        Send(sendQ, CLIENT_REQUEST, msg->data);
      default:
        break;
    }
    if (IsIdleHandler()) {
      ros = GetUrgentROSpec();
      if (ros != _STOP) {
        spec = GetNextSpec(ros);
        Send(handlerQ, NEW_AIRFSPEC, spec);
      }
    }
  }
}
```

그림 6. MANAGER 쓰레드의 의사 코드

Ⅳ. 실험 및 분석

이 장에서는 논문에서 설계한 리더를 실험하고 LLRP에 대한 기본적인 동작을 분석하고 검증한다.

리더는 기본적으로 통신 모듈을 단독으로 실험하여 동작을 확인하고 LLRP를 구현한 제어 모듈과 연동하여 동작을 확인하였다. 통신 모듈의 단독 실험은 PC에 임의의 호스트를 구현하여 통신 모듈의 동작을 검증하였다. USB, UART, Bluetooth 등의 인터페이스 환경에서 통신 모듈이 정상적으로 동작하는 것을 확인하였고, 이 후 제어 모듈과 연동하여 제어 모듈의 소프트웨어를 검증하였다. 제어 모듈은 LAN 인터페이스를 통해 NFS 상에서 소프트웨어를 개발하였고 소프트웨어 업그레이드 및 LLRP 실험을 진행하였으며, 테스트용 호스트인 PC와 제어 모듈의 인터페이스들을 연동하여 리더가 정상적으로 동작하는 것을 확인하였다. LLRP가 기본적으로 TCP 소켓(socket)을 지원하도록 되어 있으므로 유무선 LAN 인터페이스를 주로 실험하였으나, RECV 쓰레드를 변경하면 기타의 인터페이스로 쉽게 확장할 수 있음을 확인하였다.

LLRP는 다수의 ROSpec에 대해 생성, 삭제, 활성화, 비활성화, 시작, 종료를 기본적으로 확인한 후, 우선순위에 의한 ROSpec의 동작과 트리거 방식을 사용하는 ROSpec의 동작 등을 확인하였다. LLRP에서는 ROSpec의 실행 중에 종료되는 시점에 대해서는 규정하고 있지 않기 때문에 본 논문에서 구현한 LLRP는 ROSpec 안의 내부 종료 시점에서 하나의 ROSpec이 종료될 수 있다고 정의하였다. 이 시점에 대한 정의는 개발자마다 다를 수 있으며, 본 논문에서 구현한 LLRP도 향후 시스템의 최적화를 위해 변경될 수 있다.

그림 7은 활성화시킨 ROSpec의 내부 스펙 동작을 콘솔 창으로 확인한 모습이다. 큐에서 메시지를 확인하고 메시지의 파라미터에 따라 스펙들이 생성되고 동작한다. 그림 7의 중간 부분의 '#01 : 01 07 02 02 00'이 메시지 파라미터이다. #01은 메시지 넘버이고 그 뒤의 숫자들은 순서대로 ROSpec의 ID, 우선순위, 현재 상태, AISpec의 개수, RFSurveySpec의 개수를 나타낸다. 내부적으로 AISpec이 생성되고 각각의 설정에 따라 안테나와

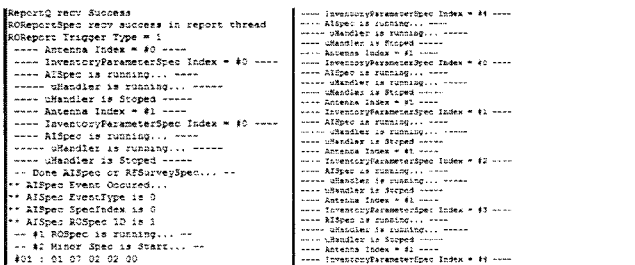

그림 7. ROSpec의 내부 Spec 동작

uHandler가 정상적으로 동작하는 것을 확인할 수 있다.

LLRP에서 트리거를 이용하여 스펙의 시작과 종료를 제어할 수 있다. 이를 사용하는 동작은 내부 스펙들에서 각각 다르게 사용하며, 시간에 관련된 트리거 방식이 여러 스펙에서 중요하게 사용된다. 또한, LLRP에서는 기본적인 동작 외에 다른 이벤트적인 동작들이 있다. 그 중에서도 ROSpec의 우선순위와 관련한 부분이 중요하다. 우선순위가 높은 ROSpec의 실행은 이전에 실행중인 ROSpec에 대한 종료시점이 중요하다. 실행중인 ROSpec이 있다면 종료 후 우선순위가 높은 ROSpec이 선점해야 하기 때문이다. 그러나 앞서 설명한 바와 같이 종료 시점에 대한 부분을 명확히 제시하지 않고 있다. 그림 8은 우선순위가 높은 ROSpec에 의한 선점을 나타낸다.

메시지의 두 번째 파라미터가 우선순위를 나타내며 숫자가 작을수록 높은 우선순위를 나타낸다. ROSpec의 종료 후 ROSpec 03번과 04번의 우선순위 차이에 의해 ROSpec 04번이 선점하여 동작하는 것을 확인 할 수 있다.

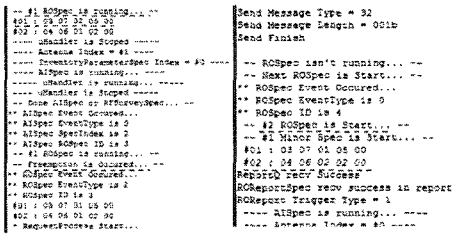

그림 8. 우선순위가 높은 ROSpec의 선점

V. 결 론

본 논문에서는 EPCglobal의 리더 프로토콜인 LLRP를 지원하는 RFID 리더를 설계하고 구현하였다. 리더의 하드웨어는 다양한 리더 시스템의 환경에 적용하고 리더의 유연성과 확장성을 높이기 위해 제어 모듈과 통신 모듈로 나누어 개발하였다. 통신 모듈은 원 칩 솔루션인 Impinj 사의 리더 칩 R1000과 R2000을 이용하여 사용자가 선택하여 구성할 수 있도록 하였으며, 차후 새로운 리더 칩을 이용한 통신 모듈도 쉽게 적용할 수 있도록 하드웨어 및 소프트웨어의 구조를 설계하고 구현하였다. 제어 모듈은 삼성의 S3C6410 MCU를 이용하고 임베디드 리눅스를 이식하여 다양한 인터페이스를 지원하도록 개발하였다. LLRP는 임베디드 리눅스 환경에서 멀티 쓰레드를 이용하여 구현하여 다양한 프로토콜의 수용, 내부 구조의 변경 등 추후의 추가적인 소프트웨어의 업그레이드가 수월하도록 개발하였다. 그리고 향후 실제 RFID 시스템에 적용하여 개발한 리더와 LLRP 클라이언트와의 시스템 연동한 동작 검증이 필요하다.

참 고 문 헌

[1] "RFID SL 기술자격검정", 한국 RFID/USN 협회, 2008

[2] EPCglobal Inc., "EPCglobal Low Level Reader Protocol(LLRP), Version 1.0.1" Aug, 13, 2007

[3] http://www.impinj.com, Indy R1000/R2000 Reader Chip

[4] EPCglobal Inc., "EPCglobal Architecture Framework Version 1.3", Mar, 2009

[5] Christian Floerkemeier, Sanjay Sarma, "An Overview of RFID System Interfaces and Reader Protocols", IEEE International Conference on RFID The Venetian, Las Vegas, Nevada USA, 2008

배 성 우 (Sung-Woo Bae) 정회원

2002년 2월 국민대학교 전자공학과

2004년 2월 국민대학교 전자공학과 석사

2004년 2월~현재 국민대학교 전자공학과 박사과정

<관심분야> RFID/USN, ASIC, 운영체제, Embedded System

유 원 상 (Won-Sang Ryu) 준회원

2007년 2월 국민대학교 전자공학과 학사
2009년 8월 국민대학교 전자공학과 석사
<관심분야> RFID, Wireless LAN

곽 호 길 (Ho-Gil Kwak) 준회원

2000년 2월 순천향대학교 전자공학과 학사
2009년 2월 국민대학교 전자공학과 석사
<관심분야> RFID, 시스템 설계, USN

정 명 섭 (Sub-Myoung Joung) 준회원

1996년 2월 홍익대학교 전자공학과 학사
1999년 2월 국민대학교 전자공학과 석사
2004년 2월 국민대학교 전자공학과 박사
<관심분야> RFID, 시스템 설계, USN

박 준 석 (Jun-Seok Park) 종신회원

1987년 국민대학교 전자 공학과 석사
1993년 국민대학교 전자 공학과 석사
1996년 국민대학교 전자 공학과 박사
1997년~1998년 Dept. of EE, UC-LA(PostDoctoralFellow)
현재 국민대학교 전자공학부 부교수
<관심분야> Mobile RFIC, RFID Active Tag, Wirless LAN

성 영 락 (Yeong-Rak Seong) 정회원

1989년 2월 한양대학교 전자공학과
1991년 2월 한국 과학기술원 전기 및 전자공학과 석사
1995년 2월 한국과학기술원 전기 및 전자공학과 박사
현재 국민대학교 전자공학부 교수
<관심분야> RFID, 실시간 처리, 이산사건 시스템 모델링 및 시뮬레이션

오 하 령 (Ha-Ryoung Oh) 정회원

1983년 서울대학교 전기 공학과
1988년 한국 과학기술원 전기 및 전자공학과 석사
1992년 한국 과학기술원 전기 및 전자공학과 박사
현재 국민대학교 전자공학부 교수
<관심분야> RFID/USN, Embedded System, 실시간 처리, ASIC 설계