

USN 환경에 적합한 임베디드 시스템 상의 RFID 미들웨어의 설계 및 구현

염세준^a, 박승보^a, 조근식^b

^a 인하대학교 컴퓨터 정보 공학과

인천광역시 남구 용현동 253번지 402-751

Tel: +82-32-875-5863, Fax: +82-32-875-5863 E-mail: joshua@eslab.inha.ac.kr, molaal@eslab.inha.ac.kr

^b 인하대학교 컴퓨터 공학부

인천광역시 남구 용현동 253번지 402-751

Tel: +82-32-875-5863, Fax: +82-32-875-5863 E-mail: gsjo@inha.ac.kr

Abstract

RFID 관련 기술은 다른 산업에 상당한 영향을 미치는 기술이다. 또한 짧지 않은 미래에 물류와 계산대를 포함하여 전 문화에 걸쳐서 인류의 생활에 상당한 변화를 불러올 기술로 여겨지고 있다. 하지만 현재 RFID 태그나 리더에 대한 기술개발과 연구는 활발히 이루어지고 있지만 RFID 미들웨어에 대한 연구는 아직도 완전한 표준이 정해져 있지 않으며 아직도 연구가 진행 중에 있다. 더욱이 임베디드 기반의 RFID 미들웨어에 대한 기술연구는 전무한 실정이다. USN 플랫폼에서 RFID 미들웨어가 임베디드 시스템 상에서 이루어져야 하는 이유는 RFID 리더나 RFID 관련 시스템들이 분산된 환경과 열악한 환경에서 실시간으로 업무를 처리하고 작업환경이 특수한 곳에서 쓰여질 것이며 RFID 미들웨어 또한 이와 같은 환경에서 동작 되어 질 것이다. 따라서 범용의 서버나 PC환경으로 미들웨어를 설계하고 구성하는 것보다 임베디드 시스템으로 설계와 구성하여 사용하는 것이 RFID 시스템에서 더 적절하다고 생각된다. 그런데 RFID 미들웨어는 여러 개의 다양한 RFID 리더로부터 수집된 RFID 태그 데이터를 한군데로 모아 오류가 없는지 분석하고 이를 원하는 외부 애플리케이션에 제공 하는 것이 목적이다. 본 연구에서는 RFID 태그 정보를 수집하는 컴포넌트와 RFID 태그에 오류가 없는지를 검사하고 데이터베이스 서버로 전송하는 컴포넌트를 임베디드 시스템으로 구현 하였다. 그리고 RFID 태그 정보를

저장하는 데이터베이스와 외부의 애플리케이션에 필요 하는 RFID 태그 정보를 송신하는 부분은 일반적인 서버나 PC의 OS상에서 구동 될 수 있도록 설계하고 구현하였다. 외부 애플리케이션과 소통은 Web-Service 기술을 이용하도록 구현하여 멀티플랫폼에서 사용 될 수 있도록 하였다. 우리는 임베디드 시스템 상에서 구현 되어 질 수 있는 RFID 미들웨어의 구조에 대해 제안하였으며, 구조에 맞게 RFID 미들웨어 시스템을 구현하여 다양한 RFID 리더로 실험을 진행하였다.

Keyword

RFID-middleware, Embedded system, Ubiquitous, USN

1. 서론

USN(Ubiquitous Sensor Network)은 시간과 장소에 구애 받지 않고 각종 센서에서 감지한 정보를 수집할 수 있도록 구성된 네트워크 환경을 지칭하는 말로 최근에는 물류의 흐름을 파악하기 위하여 RFID(radio frequency identification) 기술을 이용하여 사물에 태그를 부착하여 각종 물류 정보의 흐름을 파악하는 기술이 USN의 대표적인 예로 사용되고 있다. 즉 USN에서의 RFID 기술이란 것은 RFID 리더를 통하여 여러 환경과 위치에 놓여 있는 물건의 RFID 태그 데이터를 실시간으로 수집하여 원하는 애플리케이션에 제공하여 주는 것을 의미한다.

다. 따라서 USN에서 RFID 기술이 적용되기 위해서는 RFID 태그 정보를 한군데로 수집하여 RFID 태그 정보가 정확한지 파악하고 애플리케이션이 원하는 조건이 정보를 필터 하여 애플리케이션에 보내줄 필요가 있다. 이러한 동작을 하는 장치 또는 소프트웨어를 RFID 미들웨어 라고 한다. 본 연구에서 USN 환경에 적합한 임베디드 상의 RFID 미들웨어의 구조를 제안하고 이에 맞는 임베디드 상의 RFID 미들웨어를 설계하고 개발하여 실험 해보았다.

RFID 미들웨어는 열악한 환경이나 일반적인 PC나 서버가 놓여져 있기 힘든 환경에 처해있기 쉬우므로 일반적인 PC나 서버보다는 특정 하드웨어에 임베디드 된 시스템으로 열악한 환경에서도 동작 될 수 있고 쉽게 복구 될 수 있는 시스템으로 개발될 필요가 있다. 또한 RFID 리더들로부터 실시간으로 태그 데이터를 수집하여 필터 해야 하므로 정확한 처리 능력과 최적화된 능력을 보유한 시스템이 필요하다. 따라서 위에서 말한 조건을 충족시킬 수 있는 시스템으로는[RFID Middleware in Embedded System]이 적합하다고 판단된다.

그래서 본 연구에서 다음과 같은 내용을 개발하고자 하였다. [1]

- 임베디드 시스템 기반의 RFID 미들웨어의 설계 및 개발
- Web-Service를 이용한 RFID 미들웨어의 개발
- 다양한 RFID Reader들로부터 태그 데이터를 수신할 수 있는 임베디드 상의 RFID 미들웨어의 설계 및 개발.

본 연구에서 제안한 임베디드 상의 RFID 미들웨어 시스템은 RFID 태그 데이터를 수신(reader interface component) 하고 필터 (event/data manager)하는 부분들로 구현 되어 있다. 그리고 수집된 태그 데이터를 데이터베이스 서버에 보내서 취합하여 저장하게 하였다. 또한 다양한 애플리케이션에서 쉽게 태그 데이터를 수신할 수 있도록 web-service 기술을 사용하여 표준화된 애플리케이션 인터페이스를 제공하였다.

본 논문의 구성은 다음과 같다.

2장에서는 EPCglobal[2] 표준과 ETRI에서 제안한 spec을 표준으로 한 RFID 미들웨어에 대하여 기술 하겠다. 3장에서는 본 연구에서 제안한 방식인 임베디드 시스템 상에서 구현된 RFID 미들웨어의 구조에 대하여 제안하고 기술 하겠다. 4장에서는 본 연구에서 제안한 시스템을 실제로 구현한 RFID 미들웨어와 실험한 내용에 대하여 기술 하겠다. 5장에서 본 연구와 향후 연구 과제에 대하여 기술 하겠다.

2. RFID 미들웨어

RFID 미들웨어는 다음의 조건을 만족시켜주어야 한다. 서로 다른 기종의 RFID 리더 환경에서 RFID 태그 데이터를 수집할 수 있어야 하고, 수 많은 태그로부터 실시간 발생하는 데이터를 효율적으로 걸러내고 요약해야 한다. 그리고 외부 애플리케이션에서 요구하는 RFID 태그 데이터를 쉽게 제공해야 한다. 이러한 조건을 만족시키기 위하여, RFID 기반에서 미들웨어는 RFID 태그 데이터를 정제, 필터, 여과 하는 기능이 필요 하다.

2.1 Architecture of RFID Middleware

RFID 미들웨어는 Fig.1에서처럼 크게 세 개의 컴포넌트로 이루어져있다.[3][4] RFID 리더로부터 데이터를 수신하는 리더 인터페이스 부분과 태그 데이터를 필터 하고 이벤트 발생에 따른 동작을 해주는 Event Management 부분과 외부 애플리케이션에 태그 정보를 제공하는 애플리케이션 부분으로 이루어져 있다.

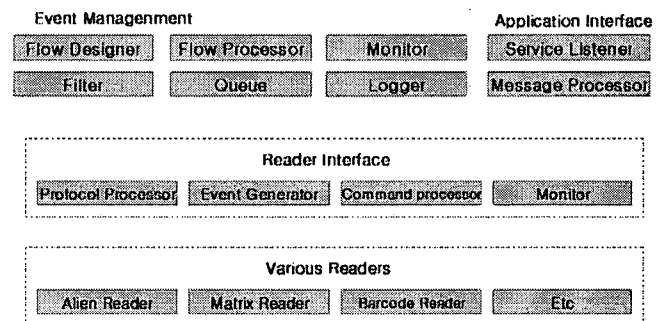


Fig. 1. RFID 미들웨어 프레임워크의 구조 [3]

2.1.1 Reader Interface

RFID 미들웨어는 다양한 형식의 리더들로부터 RFID 태그 데이터를 수신해야 한다. 이렇게 태그 데이터를 수신하는 부분을 리더 인터페이스가 담당하게 된다. 이렇게 서로 다른 리더의 인터페이스를 지원할 수 있는 기능을 미들웨어의 일부로 제공해야 한다. 리더 인터페이스가 담당하는 기능을 요약하면 다음과 같다.

- Reader Interface Protocol 처리
(Command / Response)
- TCP/IP를 통한 장치 연결 관리
- 다양한 Interface의 처리(RS232, LAN, etc)

2.1.3 Event Manager

리더 인터페이스로부터 수신된 RFID 태그 데이터의 양은 초당 수백만 건을 초과 한다. 이러한 태그 데이터를 필요로 하는 애플리케이션에서 활용하기 위해서는 무수하게 발생하는 이벤트를 관리하여 애플리케이션과 연결시켜 주는 관리자가 필요하다. 이 관리자를 Event Manager라 하며 리더로부터 수신된 태그 데이터를 필터하고 집합해주는 역할을 한다.

2.1.3 Application Interface

애플리케이션 인터페이스는 외부 시스템이나 소프트웨어 사이에 통신 역할을 제공한다. 통신은 XML-RPC, SOAP-RPC, Web-service, etc 와 같은 기능을 제공해야 한다.

3. RFID Middleware in Embedded System

본 연구에서는 임베디드 시스템에 적합하도록 RFID 미들웨어의 구조를 설계하였다. 기본적으로 앞에서 설명한 EPCglobal에서 제안한 기준을 따라서 RFID 미들웨어의 구조를 설계하였다.[6] 하지만 다양한 환경에서 실시간으로 태그 데이터를 수집하여 처리해야 하는 USN 환경에서 적합 하도록 RFID 태그 데이터를 수집하고 필터 하는 부분을 임베디드 시스템 상에서 운용되도록 하였고 필터 된 데이터를 데이터베이스에 저장하고 외부 애플리케이션에 제공하

는 기능은 일반적인 서버상에서 운용되도록 설계하고 구현하였다.

다음에 본 연구에서 제안하는 시스템에 대하여 상세히 기술하겠다.

3.1 Architecture of RFID Middleware in Embedded System

제안한 임베디드 상의 RFID 미들웨어는 Fig.2과 같이 크게 RFID 리더로부터 RFID 태그 데이터를 취합하고 필터 역할을 담당하는 Gateway & filtering 컴포넌트와 태그 데이터를 저장하고 외부 애플리케이션에 제공하는 서버 기능으로 구분하여 설계하였다.

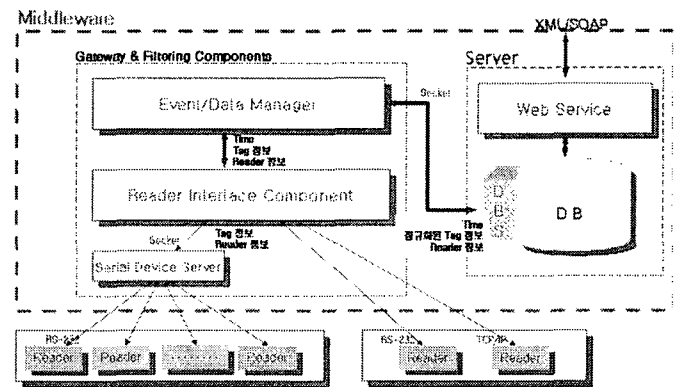


Fig.2. 임베디드 시스템 상의 RFID 미들웨어의 구조
처음의 Gateway & Filtering 컴포넌트는 임베디드 시스템 (임베디드 리눅스)에서 운용되도록 하였으며 이 부분에서 태그 데이터가 취합되고 필요 없는 태그 데이터를 제거하거나 원하는 태그 데이터를 필터 하도록 하였다. 두 번째의 서버는 필터 된 데이터를 수신하여 데이터베이스에 저장하는 데이터베이스 서버 부분과 태그 데이터를 요구하는 애플리케이션에 데이터를 전송할 수 있도록 Web-service 기술이 적용된 애플리케이션 인터페이스 부분으로 구성 하였다. 서버 부분은 리눅스 상에서 운용되도록 구현 하였다.

3.2 Gateway & Filtering Component

RFID 미들웨어는 다양한 리더로부터 RFID 태그 데

이터를 수신한 후 취합하는 부분과 태그 데이터를 필터 하여 데이터베이스서버로 전송하는 부분이 필요하다. 첫 번째의 컴포넌트를 리더 인터페이스 컴포넌트라 하며 두 번째의 필터 컴포넌트는 Event/Data Manager라 한다. 본 논문에서는 위 두 컴포넌트를 임베디드 시스템(임베디드 리눅스) 상에서 설계하였다.

3.2.1 Reader Interface Component

현재 RFID 리더는 여러 가지 프로토콜을 사용하고 있다. 프로토콜은 serial port가 주류를 이루고 있으나 경우에 따라서는 TCP/IP나 USB를 이용하기도 한다. 따라서 RFID 미들웨어의 리더 인터페이스 컴포넌트는 리더의 다양한 프로토콜을 지원해야 하며 다양한 방식(EPC 1, EPC gen2)의 리더로부터 태그 데이터를 수신 할 수 있어야 한다. 본 연구에서는 다양한 프로토콜 중에 serial port(RS-232)와 TCP/IP port를 지원할 수 있는 리더 인터페이스 컴포넌트를 설계하고 구현하였다.

그리고 미들웨어 시스템 자체적으로 serial port를 두어 직접 수신 할 수도 있으나 다수의 리더로부터 태그 데이터를 수신하기 위하여 리더 인터페이스 컴포넌트는 앞에 Serial 기반의 서버를 두었다. Serial 기반의 서버는 다수의 serial port로부터 태그 데이터를 취합하여 TCP/IP로 태그 데이터를 한번에 미들웨어에 송신하는 기능을 담당하고 있다. 이렇게 Serial 기반 서버를 둔 것은 미들웨어에 다수의 serial port를 둔다는 것이 물리적으로 한계가 있기 때문이다.

리더 인터페이스 컴포넌트는 리더로부터 태그 데이터를 수신하는 기능과 리더의 상태 정보를 수신하는 기능을 가지고 있다. 따라서 두 가지 정보를 수신하여 Event/Data Manager에 송신하게 된다.

이때 임베디드 시스템 상에서 리더 인터페이스 컴포넌트는 와 Event/Data Manager라는 독립적인 프로세스가 존재하게 되고 프로세스간에 데이터를 통신을 해야 한다. 그래서 본 시스템에는 공유 메모리 방식을 사용하여 두 개의 프로세스의 데이터 통신을

해결하였다.

그리고 현재의 대다수의 리더들이 단순히 태그 데이터만을 전송하고 있으므로 본 시스템의 리더 인터페이스 컴포넌트에서는 리더의 상태 정보와 태그 데이터에 시간 정보를 추가하여 Event/Data Manager에 송신하도록 하였다.

3.2.2 Event/Data Manager

다음 그림은 Event/Data Manager의 개략적인 구조다. 그림의 Event/Data Manager는 크게 3가지 부분으로 나뉘어 질 수 있는데 첫 번째는 공유 메모리(Share Memory)에서 태그 정보나 리더 상태 정보를 가져오는 부분이고 두 번째는 가져온 정보를 필터 하는 부분이고 세 번째는 데이터베이스 서버에 필터 된 정보를 송신하는 부분이다.

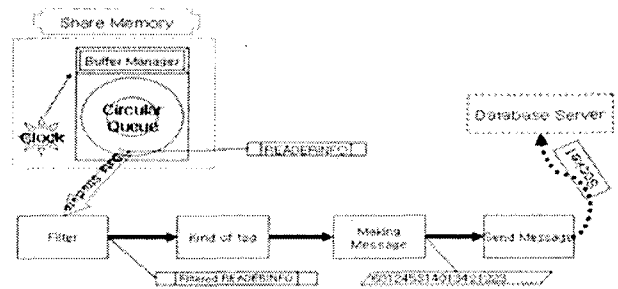


Fig.3. Event Manager 데이터 흐름

-공유 메모리 Access 부분

리더 인터페이스 컴포넌트에서 쓰여진 공유 메모리로부터 RFID 태그 정보나 RFID 리더의 상태 정보를 가져오는 부분으로 공유 메모리에 데이터가 없으면 대기 모드로 있고 데이터가 존재하면 읽어 오는 작업을 반복하여 실행한다. 공유 메모리에서 데이터를 읽어 올 때는 RFID 태그 정보는 READERINFO 구조체 구조로 읽어오게 되고, RFID 리더 상태 정보는 HEAD를 불러와서 리더 상태를 판정하게 된다.

-데이터 필터 부분

공유 메모리에서 불러온 RFID 태그 정보나 리더 상태 정보가 정확한 규격에 해당하는지 판단하는 부분으로 태그의 길이나 특정 비트의 자료 형태 종류나 CRC나 Checksum 같은 방법으로 필터를 실행하게

된다. 필터를 거치게 되면 FILTEREDREADERINFO 구조체가 만들어지며 RFID 태그용 구조체와 RFID 리더 상태 정보용 구조체의 2가지가 있다. RFID 태그용 구조는 ReaderId (Tag가 수신된 RFID 리더의 ID), Tag(RFID 태그의 값), Time(RFID 태그가 수신된 시간), KindTagId(RFID 태그의 종류)로 이루어져 있다. RFID 리더 상태정보용 구조는 ReaderId(상태가 지시하는 RFID 리더의 ID), Time(리더의 상태가 체크된 시간), Status(리더의 상태)로 이루어져 있다.

-데이터 송신 부분

필터 된 구조체 정보는 전송을 위하여 문자열로 변환이 되며, 문자열 생성 규칙 역시 RFID 태그와 리더 상태 정보로 나뉘어 진다. 두 개의 정보의 생성 규칙은 동일하나 생성된 문자열의 패턴이 다르다. 하지만 문자열 생성시 문자열 앞부분에 Header 정보가 포함 되는 것은 동일하다. Header에는 태그 또는 리더 상태 정보의 속성들에 대한 문자 개수들이 할당되며 속성별로 2 바이트씩 할당된다. 생성된 문자열은 메시지가 되어 데이터 그램 소켓을 통해 데이터베이스 서버로 전송된다. 데이터 그램 방식은 TCP/IP 방식에 비해 속도가 빠르다. 데이터가 지속적으로 발생하므로 Event/Data Manager는 정보를 송신한 후 정보가 전달이 되는지 회신을 받지 않고 다음 데이터를 계속 송신한다.

이렇게 세 부분으로 나뉘어지는 Event/Data Manager는 리더 인터페이스 컴포넌트로부터 정보가 발생할 때마다 정보를 필터 한다는 면에서 Event Manager라 명명되었고, 정보를 데이터베이스 서버에 전송한다는 면에서 Data Manager라 명명하였다.

따라서 두 가지 이름을 합하여 Event/Data Manager라 하였다. 또한 리더에서 데이터가 발생할 때 마다 데이터 처리를 해주어야 하므로 리더 한 개 당 한 개의 프로세서가 생성 되도록 하였다.

3.3 서버 (DB & Application Interface)

3.3.1 DB Server & DB

데이터베이스 서버는 Event/Data Manager에서 발생한 정보를 소켓을 통하여 수신하여 데이터베이스에 저장하는 부분이다.

물리적으로 Event/Data Manager와 데이터베이스 서버가 나뉘어져 있기 때문에 둘 간에는 소켓을 통하여 데이터를 송수신하도록 하였으며, 데이터 그램 방식(UDP)을 통하여 데이터를 수신하게 하여 빠르게 정보가 송수신 되도록 하였다.

Event Manager와 달리 데이터베이스 서버는 태그 정보용 프로세스와 리더 상태 정보용 프로세스가 각각 하나씩만 실행되면 된다. 이는 여러 개의 Event Manager 프로세스에서 단일한 IP로 메시지를 보내주게 되면 IP에 해당하는 컴퓨터에서 작동하고 있는 데이터베이스 서버에서 여러 개의 메시지를 일괄적으로 처리해 줄 수 있기 때문이다.

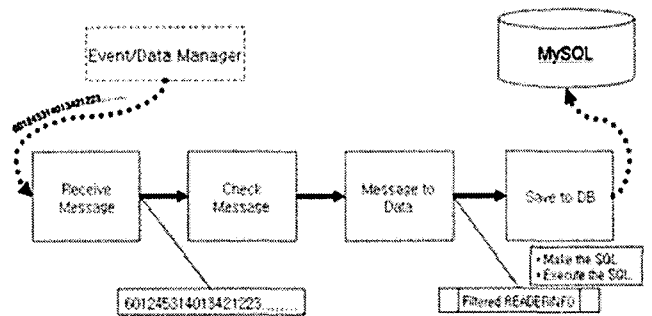


Fig. 4 데이터베이스 서버의 흐름

Event/Data Manager로부터 수신된 문자열은 송수신되면서 오류가 발생되었는지를 Header 정보를 가지고 검사하고 이상이 없을 때에는 구조체(FILTEREDREADERINFO)로 변환한다.

그리고 구조체로 변경된 정보는 SQL 문으로 변환된 후 My-Sql의 데이터베이스의 해당 Table로 저장되게 된다.

3.3.2 Application Interface

Fig.5은 Web Service가 구현되기 위한 전체적인 Framework를 도식화한 것이다.

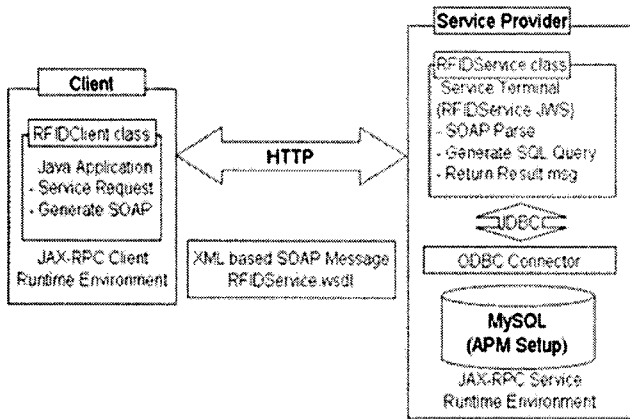


Fig.5. 애플리케이션 인터페이스의 웹 서비스 시스템

외부 애플리케이션에서 RFID와 관련된 정보를 수신하기 위해서 본 연구에서는 Web Service 기술을 사용하였다. Web Service는 인터넷을 통한 네트워크 상에서 기존 애플리케이션 시스템을 표준화한 기술로서 애플리케이션 간의 소통이 가능토록 한 표준화 기술이다. Web-service를 지원함으로써 외부 애플리케이션이 쉽게 데이터베이스의 RFID 정보를 검색할 수 있도록 하였다.

4. 실험 및 실험 장비

본 프로젝트에 사용 되어 진 임베디드 리눅스 시스템은 다음의 조건을 만족하면 될 것으로 판단 되었다.

- RFID 미들웨어 역할을 할 수 있어야 한다.
- 임베디드 리눅스 기반의 시스템을 갖추어 있어야 한다.
- 대량의 RFID 데이터를 처리해야 하므로 속도가 빨라야 한다.
- 네트워크 연결이 편하도록 제공할 필요가 있다.
- 다양한 방식의 프로토콜을 제공해야 한다.

따라서 본 연구에서는 위의 조건을 만족시키면서 안정적인 시스템이 필요하였다.[7] 그래서 임베디드 리눅스 개발 KIT으로도 많이 사용되는 한백전자의 HBE-Empos II(이하 Empos II)를 활용하였다.

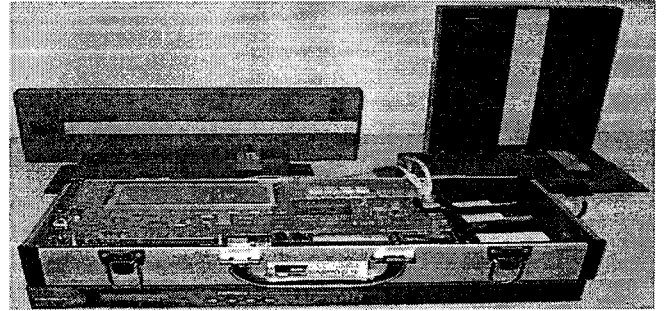


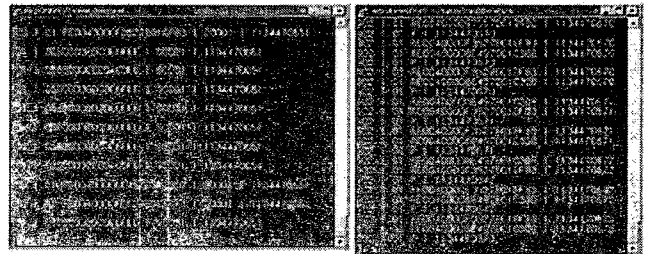
Fig.6. 임베디드 시스템 상의 RFID 미들웨어의 구현 장치

Fig.6은 본 연구에서 구현한 RFID middleware in Embedded System이다.

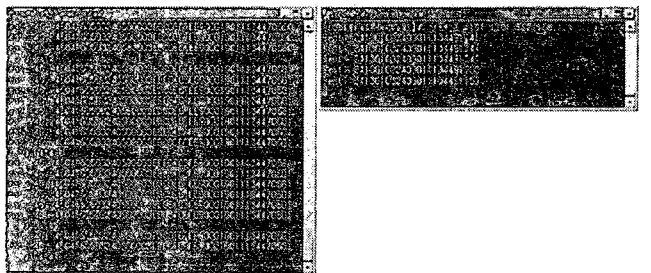
실험은 900MHz RFID reader 2개와 13.56MHz의 RFID 리더 하나를 사용하여 실험하였다.

본 연구에서 제안한 시스템을 구현하여 세 개의 리더를 접속하여 테스트 하였을 때 원활한 데이터 흐름을 보였으며 각 컴포넌트 정확한 동작을 하는 것을 확인하였다.

Fig.7는 미들웨어의 각 컴포넌트에서 발생한 데이터나 string을 Screenshot한 것이다.



a. 리더 인터페이스



b. Event/Data Manager



c. 데이터베이스 서버

Fig.7. RFID 미들웨어 각 컴포넌트의 실행 화면

본 연구에서 Web-service는 Java를 사용하여 구현하였으며 Sun에서 제공하는 Axis를 사용하여 구현하였다.

5. 결론 및 향후 과제

USN 환경에서 RFID 시스템은 다양한 환경에 노출되게 된다. 다양한 환경은 열, 온도, 먼지, 전자파와 같은 해로운 요소들이 되겠다. 이러한 환경에서 RFID 데이터를 수집하여 취합하는 미들웨어는 환경에 강인한 면을 갖추어야 한다. 또한 해로운 환경에서 정상 동작을 해야 하는 것뿐만 아니라 빠르게 실시간으로 대용량의 데이터 처리하는 능력을 갖추어야 한다. 따라서 이러한 조건을 만족하기 위해 일반적인 pc나 서버에 미들웨어를 가지고 있는 것 보다는 임베디드 상에서 미들웨어를 가지고 있는 것이 더 좋다고 판단된다.

본 연구에서는 RFID 데이터를 수집하고 필터 하는 Gateway & Filtering Component를 임베디드 시스템 상에서 설계하여 해로운 환경에서도 빠르게 데이터를 실시간으로 취합하게 하였고 이를 실제로 구현하여 정상 동작하는 것을 확인하였다.

본 연구의 RFID 미들웨어는 향후에 RFID 검색 시스템이 추가 되어질 필요가 있다. 향후 지속적인 연구를 통하여 본 시스템에 Context를 기반으로 한 RFID 검색 시스템이 연동된 미들웨어 시스템으로 발전시킬 계획을 가지고 있다.

참고 문헌

- [1] Dae-Jin Jo, "Theory and Application of RFID" Hongleung pub 2005.
- [2] EPCglobal, <http://www.epcglobalinc.org/>
- [3] Young-II Kim, Joo-Sang Park, Tae-Su Cheong.: Study of RFID Middleware Framework for Ubiquitous Computing Environment. Advanced Communication Technology, 2005, ICACT 2005. The 7th International Conference on Volume 2, 21-23 Feb. 2005 pp. 825-830.
- [4] ISO/IEC JTC1/SC31 WG4, "ISO/IEC WD

15962 Data Syntax", August 2000..

- [5] Auto-ID Center, "Auto-ID Savant Specification 1.0" September 2003.
- [6] ISO/IEC JTC1/SC31 WG4, "ISO/IEC WD 15961 Data Protocol: Application Interface", September 2003.
- [7] Joo-Hee Park, Jin-An Seol, Young-Hwan Oh.: Desing and Implementation of and Effective Mobile Healthcare System Using Mobile and RFID Technology. Enterprise networking and Computing in Healthcare Industry, 2005. HEALTHCOM 2005. Proceedings of 7th International Workshop on 23-25 June 2005 Page(s): 263-266.