임베디드 시스템과 무선 랜을 이용한 이동성이 높은 재고단위의 위치관리 시스템 설계 및 구현

이 재 현† · 권 경 희††

요 약

보관 중인 제품의 정확한 위치를 파악하는 것이 창고관리의 필수적인 요소이다. 그러나 제품이 빈번하게 위치를 변경하며 보관되는 경우에는 그 위치 파악이 어렵게 되며 이로인해 효율적 창고관리가 불가능해진다. 본 논문에서는 임베디드 시스템과 무선 랜을 이용하여, 이러한 이동성이 높은 재고단위(SKU : Stock Keeping Unit)의 창고 내 위치관리에 응용할 수 있는 새로운 시스템을 제안한다. 시스템은 RFID(Radio Frequency IDentification)와 관리 단말기, 모바일 기기인 휴대폰과 PDA 그리고 중앙 관리 시스템으로 구성된다. 시스템의 사례로 이동성이 높은 재고단위의 모델인 대형 중고자동차의 위치관리 시스템을 선택하였으며, 시스템이 운용되는 각 부서의 업무를 분석후 기능별로 구현하였다. 본 연구를 통하여 무선 랜이 접목된 임베디드 시스템이 이동성 높은 재고단위의 입출고 상황과 창고 내에서의 위치를 실시간적으로 정확하게 파악할 수 있어, 창고의 운영 및 관리를 체계적으로 처리할 수 있음을 보인다.

Design and Implementation of Location Management System of Stock Keeping Unit with High Mobility Using Embedded System and Wireless LAN

Jae Hyun Lee† · Kyung Hee Kwon††

ABSTRACT

It is essential to get the exact location information of products for a warehouse management. It is very hard, however, to know the location of products which change their location in warehouse frequently. This causes the effective warehouse management to be almost impossible. In this paper, we suggest a new location management system for such a SKU (Stock Keeping Unit) with high mobility. The system is composed of RFID (Radio Frequency IDentification), a management terminal with wireless LAN, mobile devices (Cellular Phone & PDA), and a central management system. As a model of a SKU with high mobility, we selected a used-car stored in a large-scale warehouse. We designed and implanted used-car location management system. After analyzing the operations of each position in used-car warehouse where the system will run, we implemented each function. This research shows that an embedded system with wireless LAN is able to know the status of coming in and out and location of a SKU with high mobility in warehouse very accurately in real time. Therefore, it makes the warehouse to be controlled systematically.

키워드 : 임베디드 시스템(Embedded System), 무선 랜(Wireless LAN), 위치관리(Location Management)

1. 서 론

물류관리는 신속한 제품 및 서비스 제공을 통해 경쟁적 우위를 확보할 수 있어 기업의 핵심 경쟁무기가 되어 가고 있다. 전통적인 물류관리의 한 분야로서 적시에 비용 효과적인 방법으로 제품을 전달하는 창고관리 또한 기업 목표 달성에 있어 그 중요성은 날로 더해가고 있다. 입고되는 제품의 위치 선정이나 출고될 제품의 위치 파악은 창고관리에서 필수적인 업무이며, 특히 보관되는 제품이 판매될 때까지 자주 창고를 출입해야 하는 이동성이 큰 제품일 경우에는 위치관리가 더욱 중요하나 실제 이러한 제품의 위치관리는 매우 어렵다. 중고 자동차를 판매하는 기업을 예로 들어보자. 판매할 자동차를 창고에 입고되어 최종소비자에게 판매될 때까지 여러번 시운전을 거치는 것이 일반적이다. 자동차를 보관하는 창고의 면적이 시운전을 할만큼 충분히 넓지 않기 때문에 창고 밖으로 나가서 시운전을 하게 되고 다시 창고로 들어왔을 때 원래 있던 장소에 다른 차가 보관되어 있다면 임의의 빈 공간에 보관하게된다. 이렇게되면 한 자동차의 보관장소가 자주 바뀌게 되어 위치 파악이 안되고 소비자가 원하는 제품을 적시에 공급하는데 어려움을 겪게 된다. 심지어 제품이 분실되는 경우에도 이를 파악하는데 많은 시간이 소요되어 실질적으로는 효율적인 창고관리가 거의 불가능해진다.

※ 이 연구는 2001학년도 단국대학교 대학 연구비의 지원으로 연구되었음.
† 준 회 원 : 단국대학교 대학원 전자계산학과
†† 종신회원 : 단국대학교 전자계산학과 교수
논문접수 : 2002년 9월 28일, 심사완료 : 2002년 12월 5일

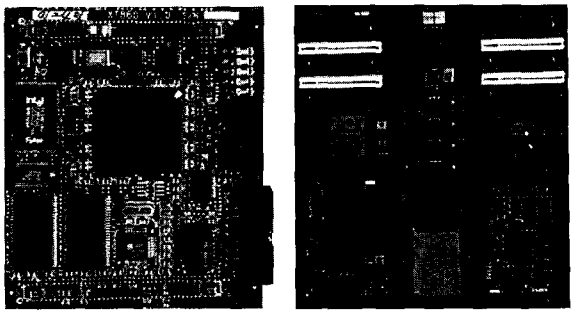
이동성이 높은 재고단위는 어떤 제품의 특성에 의해 정의되기보다는 창고를 사용하는 기업의 업종이나 창고의 용도에 의해 정의된다고 볼 수 있다. 중고자동차뿐만 아니라 새 자동차일지라도 자동차 리스업계의 창고에서는 이동성이 높은 재고단위이다. 임대된 새 자동차가 출고되었다가 다시 돌아오면 창고 내에서 보관되는 위치는 달라지게 될 것이다. 창고의 면적이 협소할수록 그리고 임대되는 횟수가 많아질수록 보관 위치가 달라질 확률은 높아질 것이다. 철판의 경우에도 생산 현장의 자재창고에서는 이동성이 높은 재고단위라고 할 수 있다. 철판은 생산 현장에서 생산될 제품의 용도에 따라 적절한 크기로 잘려진 뒤 생산라인에 공급된 뒤에는 다시 보관장소로 돌아온다. 이때 원래 보관되던 장소에 다른 규격의 철판이 입고되었다면 생산 현장에서 돌아온 철판이 보관되는 위치는 달라지게 될 것이다. 즉 본 논문에서 이동성이 높은 재고단위란 창고에서 보관되는 위치가 자주 변동되는 단위 제품을 말한다.

본 논문에서는 이러한 이동성이 높은 재고단위의 효율적 위치관리 시스템의 모델로써, 대형 중고자동차 위치관리시스템을 설계하고 구현하였다. 물론 본 논문에서 구현한 위치관리시스템은 단지 자동차의 위치만을 관리하는 것이 아니라 자동차의 입고에서 판매 후 출고되는 과정까지를 관리해주는 것이나 본 연구의 핵심이 되는 부분은 위치관리이다. 이는 지금까지 시도되어지지 못하였으며, 최근에 급속히 발전한 임베디드 시스템과 무선 랜 기술로 인해 산업현장에 적용이 가능해진 것이라 볼 수 있다.

본 논문에서는 첫 번째로 시스템에 사용될 하드웨어 제품의 선정에 대해 서술하였다. 두 번째로 본 논문의 시스템에서 쓰인 임베디드 시스템, CMS(Central Management System), 모바일 기기(Mobile Phone, PDA) 그리고 네트워크 설계 및 구축방법 등 전체시스템의 설계에 관해 논의하였다. 세 번째는 차량 및 차량위치정보 그리고 관리자에 대한 데이터 베이스 설계를 소개하였다. 네 번째로 모바일 기기와 차량출입관리 시스템과의 데이터베이스를 통한 차량 위치정보에 대한 실시간 동기화와 CMS와 PDA간에 사용되는 AES (Advanced Encryption System)암호화 알고리즘에 대해서 설명하였다. 다섯 번째로 구현에 대한 코딩 및 실제 작동원리에 대해 설명하였으며 마지막으로 결론 및 시스템의 응용분야에 대해 기술하였다.

2. 하드웨어 제품 구성

본 논문에서 설계한 시스템은 물품정보를 관리하기 위한 임베디드 시스템이 필요하며, CMS로 사용할 리눅스 서버, 관리자 및 영업자가 사용할 모바일 기기(Mobile Phone, PDA) 그리고 무선 랜을 위한 AP(Access Point)가 필요하다. 물품관리를 위한 임베디드 시스템은 물품의 입·출고 정보를 무선랜을 통하여 CMS(Central Management System)와 연동하여 관리하며, 영업 및 관리자는 모바일 기기 및 PC를 통하여 기존의 인터넷을 이용하여 CMS와 연결, 원하는 작업을 수행할 수 있도록 설계하였다.



XN-860Core Module　　XNMB-CR

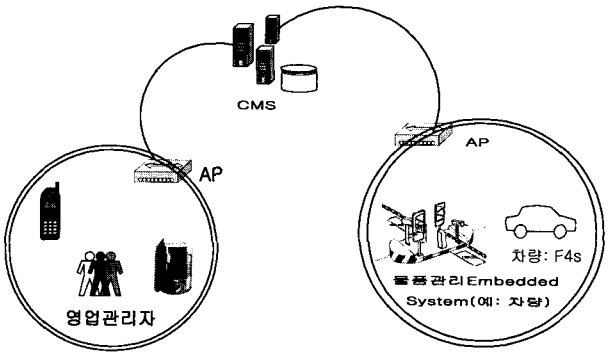
(그림 1) 임베디드 시스템 개발용 보드

(그림 1)에서 임베디드 시스템 기기인 XN-860Core Module은 기본적으로 MPC860SAR 50MHz의 CPU를 사용하고 Board 내부에는 2Mbyte 16Bit Flash Memory, 2Mbyte 32Bit SDRAM 등 메모리가 내장되어 있어 기본적인 프로세서로서의 기능을 가지고있다. 그리고 임베디드 시스템에는 몬타비스타의 임베디드 리눅스인 HardHat Linux2.0JE를 사용하였다. 임베디드 시스템에서 사용할 무선랜을 위하여 Xiontek사의 XNMB-CR board를 사용하는데, XNMB-CR board는 Processor to PCI Bridge인 Tundra 사의 Qspan? 및 PowerSpan을 장착하여 XN860과 XN8260보드에 PCI Interface를 제공하고 있으며, 또한 XN8260의 PCMCIA Interface를 통해 PCMCIA 슬롯을 제공한다[14]. 그리고 임베디드 시스템과 호스트와의 무선통신에 사용하는 AP(Access Point)는 ORINOCO사의 Wireless Access Point-1000을 사용하였다. AP는 속도 11Mbps, RC4 암호화를 통한 128비트 키 보안, IEEE 802.11b (WI-FI) 그리고 스패닝 트리 알고리즘등을 사용한다[15].

영업 및 관리사원이 사용하는 모바일 기기는 휴대폰과 PDA 두 가지로 구현하였다. 휴대폰은 SUN사의 J2ME Wireless Toolkit 에뮬레이터를 사용하였고, PDA는 Compaq iPAQ Pocket PC에 구현하였다.

3. 시스템 설계

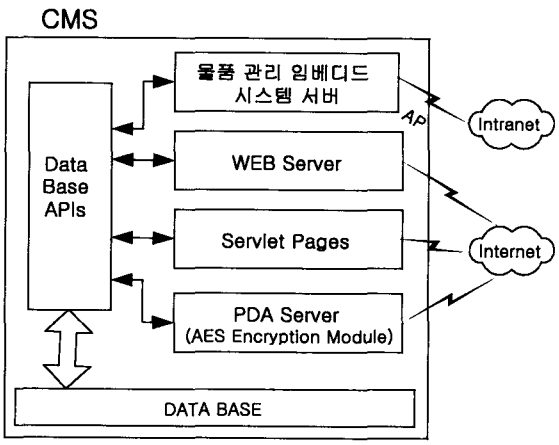
이동하는 물품에 대한 저장위치, 저장상태 그리고 저장시간 등 변화하는 물품의 정보를 임베디드 시스템을 통하여 CMS에게 전송되고, CMS는 그 정보를 데이터 베이스에 저장한 후 영업 및 관리자가 모바일 기기(휴대폰, PDA)를 통하여 요청시 물품정보를 전송하여 준다. 물품에 대한 정보는 RFID를 사용하여 정보의 읽기/쓰기가 가능하며 임베디드 시스템은 안테나와 연결되어 단말기 역할을 하게된다. 모바일 CMS 물품 관리 임베디드시스템 연결은 기존의 케이블링이 불가능한 지역에서도 네트워크를 구축할 수 있게 하기 위해 무선랜을 사용하였다.



(그림 2) 전체 시스템 설계도

3.1 중앙관리 시스템(CMS)

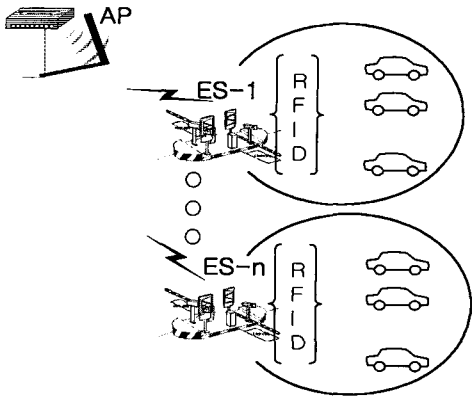
CMS는 임베디드 시스템과 모바일 기기에 TCP/IP을 통해 연결되며, 각 단말기가 요구하는 정보를 전송해 주고 필요한 작업에 대한 API를 제공하는 역할을 한다. (그림 3)은 CMS의 주요작업을 나타낸다. 내부는 각 장치들에 대한 서버군을 형성하며 모든 서버는 데이터베이스와 연결할 수 있는 Data Base API를 이용한다. 각 서버들의 기능은 다음과 같이 요약할 수 있다.



(그림 3) 중앙관리 시스템 구조

- 물품(차량)관리 임베디드 시스템 서버 임베디드 시스템의 가동시 임베디드 시스템 초기화를 위한 데이터 전송과 물품정보(장소, 시간, 상태)의 변화를 실시간으로 관리할 수 있다.
- WEB Server 온라인상으로 기존의 유선망을 사용하여 물품(차량)에 대한 정보관리기능을 제공한다.
- Servlet Pages JVM을 사용하는 휴대폰에서 MIDP를 이용한 프로그램이 호출할 물품(차량)에대한 정보를 XML 문서화하여 전달해 준다.
- PDA Server 각 PDA의 접속당 스레드를 생성하여 통신한다. 통신상의 요청/응답은 AES 암호화 알고리즘을 기반으로 이루어지며, 물품정보에 대한 응답은 XML 문서화를 기본으로 한다.

3.2 물품 관리 임베디드 시스템



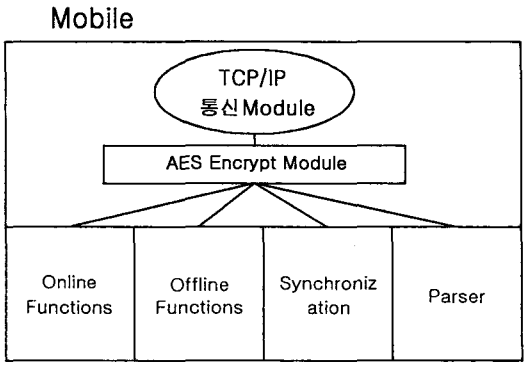
(그림 4) 물품(차량)관리 임베디드 시스템 구조

임베디드 시스템은 물품의 출입을 관리하는 단말기로써 최초 가동시 서버와 무선랜을 통하여 연결되며 TCP/IP를 이용했다. 각 출입단말기는 자신이 관리하는 영역에 대한 차량들의 정보를 단말기 가동시에 서버에서 전달받고, 이 후 각 단말기를 통해 출입하는 차량에 대한 정보를 RFID를 통해 차량에 데이터 읽기/쓰기를 수행하며 또한 차량의 변경 및 추가정보를 서버로 전송하여 CMS가 모바일기기에 실시간 물품정보를 제공할 수 있게하였다.

(그림 4)가 차량관리 임베디드 시스템에 대한 구성도이며, ES-1에서 ES-n까지의 임베디드 시스템 단말기가 자신이 할당받은 영역내의 차량을 관리하며 각 차량의 통과시 서버와 필요한 데이터를 주고받게 되며 AP는 서버와 통신할 수 있는 무선 AP이다.

3.3 영업 및 관리자의 모바일 기기

모바일 기기의 사용자는 회사 Intranet안의 관리자와 외부 영업사원으로 분류할 수 있다. 회사내의 관리자는 모바일 기기를 갖고 회사내의 어디에서든 물품(차량)에 대한 정보를 이용할 수 있는 시스템으로 설계되었다. 회사의 외부에서 근무하는 영업사원은 CMS와 인터넷을 통하여 통신할 수 있는 지역에서 통신이 가능하고, 인터넷을 사용할 수 없



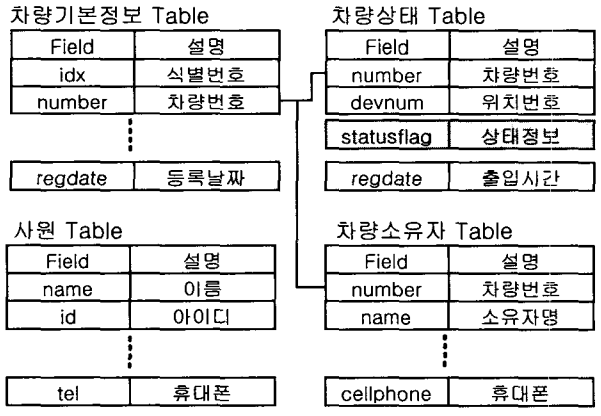
(그림 5) 영업 및 관리자의 Mobile 구조

는 지역에 대해서는 모바일 기기의 Local Application을 가동하여 원하는 차량에 대한 정보를 얻을 수 있도록 하였다. (그림 5)는 모바일 기기에서의 Application 설계 그림이다.

(그림 5)의 Offline Functions은 PDA에만 있는 모듈이고, 나머지는 휴대폰과 PDA의 공통된 모듈이다. 각 기능들을 요약하면 다음과 같다. 첫 번째로 Online 기능은 TCP/IP를 통하여 CMS와 통신이 가능한 상태에서 정보검색 및 수정을 할 수 있는 기능을 말한다. 두 번째로 Offline 기능은 영업사원이 외근 시 PDA에 자신이 원하는 차량의 정보를 복사한 후, 필요에 따라 복사된 정보에 대한 검색 및 수정을 가할 수 있는 기능이다. 세 번째로 동기화는 현장에서 갱신되는 차량에 대한 위치정보를 임베디드 시스템과 PDA간의 실시간으로 갱신된 정보공유를 가능하게 해주는 기능이다. 이로써 차량에 대한 접근 권한 및 이동을 관리자와 일치시켜 차량이동에 대한 통제를 관리자가 실시간을 할 수 있게 하였고, 반대로 차량의 현재 상태를 관리자에게 알릴 수 있어 관리자의 차량에 대한 통제를 제한할 수 있다. 마지막 Parser는 차량의 정보이동 시 XML 형태로 작성된 데이터를 Parsing하는 모듈이다. 위의 기능을 통하여 서버와 통신하기 전에 데이터들은 AES 암호화 알고리즘 모듈상에서 암호화/복호화 과정을 거친 후 통신모듈을 사용하게 된다.

4. 데이터베이스 설계

물품 관리 및 판매에 대한 테이블설계 시 고객관리 및 물품의 자원관리를 위한 상세한 테이블들은 본 논문에서 제외시키고, 전체적인 위치관리시스템 운영상에서 핵심이 되는 테이블만을 선택해 설계하였다. (그림 6)은 4개의 중요한 테이블들을 보여주고 있다. 차량 기본테이블은 차량의 판매에 기본이 되는 정보를 갖고있고, 차량상태 테이블은 차량의 현재 위치 및 상태(출입, 수리, 판매예정)에 대한 정보를 갖고있다. 그리고 판매사원 테이블은 PDA를 이용하는 각 판매사원마다 고유의 아이디와 비밀번호를 이용하여 인증한 후에 정보를 이용할 수 있도록 하였다.



(그림 6) 차량관리 및 판매를 위한 데이터베이스 테이블

차량상태 테이블에서 상태정보에 해당하는 statusflag 필드를 검사하여 임베디드 시스템과 모바일기기 사이에 차량의 출입여부에 대한 잠금기능을 체크할 수 있다. 본 시스템에서 차량은 기본정보 테이블, 차량소유자 테이블, 그리고 차량상태 테이블에 의해서 관리되어지고 그 중에서 차량상태 테이블은 차량출입관리 임베디드 시스템과 사원테이블에 의존적으로 운영되어진다.

5. 무선 랜 보안

무선 랜은 전파를 사용하여 통신하는 특성으로 여러 보안상의 취약점을 가지고 있다. 기존의 유선랜은 네트워크 자원을 사용하기 위해 물리적으로 네트워크에 연결을 해야하는 이유로 권한이 없는 사용자가 네트워크 자원에 접근하기 어려웠다. 그러나 무선랜의 경우에는 사용자의 네트워크 자원에 접근하기 위해 무선 랜 카드에서 나오는 전파가 AP에 다다를 수 있기만 하면 된다. 따라서 다른 사람들이 데이터의 내용을 볼 수 없도록 암호화를 할 수 있는 메커니즘이 필요하다.

AES는 보안기능을 강화하기 위해 802.11 표준의 하위 그룹인 802.11i에서 제안하고 있다. AES 암호화 알고리즘은 입출력이 블록단위로 동작하며 기본적인 단위는 Byte단위로 작동한다. 그리고 키가 64bits로 고정되어있는 DES 암호화 알고리즘과는 달리 AES 알고리즘은 키의 크기를 다양하게 설정할 수 있다. AES에서의 키 크기는 128, 192, 256bits, 블록크기는 64, 128, 256bits로 설계기준이 제안되었다. 각 라운드 변환은 외부에서 주어진 1차원 형태의 128비트 블록을 2차원 4행×N_b(단, N_b=4구성)로 구성되는 State로 변환한 후, State내의 배열에 대해서 연산을 수행한다[13].

AES 암호화 알고리즘은 여러 플랫폼에 빠르고 코드가 간단하며 공격에 강한 암호화 알고리즘으로 Triple DES를 대신하여 업계 표준으로 현재 NIST(National Institute of Standards and Technology)에서 추진 중이다. 본 논문에서 PDA와 CMS사이의 통신에 AES(Advanced Encryption Standard) 암호화 알고리즘을 이용하였으며 128비트 키, 블록과 10라운드를 기준으로 구현하였다.

6. Coding

본 장에서는 임베디드 시스템에서 모바일기기까지의 핵심이 되는 코드를 중심으로 전체 시스템의 기능들에 대해서 설명한다.

6.1 CMS의 통신모듈

CMS에서의 통신모듈은 클라이언트를 임베디드 시스템과 PDA 그리고 웹브라우저와 휴대폰으로 분류할 수 있다. 전자의 경우는 서버 모듈이 별도로 존재하여 각각의 프로세서로 작동하는 반면, 후자의 경우는 웹 서버에 의존적으로 JSP 페이지와 서블릿으로써 작동한다. (알고리즘 1)은 임베

디드 시스템 서버의 구현부분을 요약하여 설명하여 놓은 것이다. 임베디드 시스템은 각 장치당 할당받은 구역에 대해 출입하는 차량을 관리하며 서버에서는 각 장치에 대해 스레드로써 관리한다. 아래의 (알고리즘 1)의 설계는 database manager인 dbq객체, 로그파일을 관리하는 Log객체 그리고 각 장치마다 할당되는 스레드 객체로 이루어져있다. PDA 서버는 이 모듈에 AES 암호화 알고리즘과 XML 문서 파싱 모듈이 추가되면 된다.

```
public static void main ( String [ ] args ) throws IOException
{
init ( ) ; // database manager 초기화
ServerSocket serverSock = new ServerSocket ( PORT ) ;
    // 임베디드에 대한 서버 소켓 생성
while ( true ) {
    Socket sock = serverSock.accept ( ) ;
    new Connection ( sock ) ;
        // 임베디드 시스템을 위한 Thread 생성
}
}
static class Connection extends Thread {
// 임베디드 시스템과 통신할 thread객체
public void run ( ) {
try {
while ( true ){
    if (( count = fromClient.read ( buf )) == -1 ) break ;
    switch ( getKind ( buf ) ) {
// 임베디드 시스템의 요청에 의해 작업분류
case 0 : // init
         temp = new String ( buf ) ;
    devno = Integer.parseInt ( temp.substring (6, 9 )) ;
        // 임베디드 시스템 초기화시 장치번호를 분류
    logfname = "log" + devno ;
         initlog ( ) ;
         // 장치번호로 장치에 대한 로그파일 초기화
      sdata = dbq.getnextparking ( devno ) ;
      // 임베디드 시스템에게 초기 주차자리 지정
    break ;
case 10 : // 현재 차량의 상태정보 요청
    // 생략 …
     sdata = dbq.getStatus ( devno, carno ) ;
        // DB manager한테 상태정보 요청
    break ;
case 20 : // 이동간 차량이 주차를 요청
    if ( dbq.insertCar ( devno, posno, carno ) )
        // 들어온 차량에 대한 차량정보갱신
      sdata = dbq.getnextparking ( devno ) ;
// 다음 주차위치를 임베디드 시스템한테 전송
    else sdata = "nothing" ; // 여분주차위치가 없음
    log.WriteLog ( logmsg ) ;  // 로그파일에 기록
    break ;
case 30 : // 차량이동 시
if ( dbq.deleteCar ( posno, carno ) )
// 차량정보 갱신
        sdata = "ok" ;
     else sdata = "fail" ;
        // 이동할 수 없도록 잠김상태의 차량일경우 실패
      log.WriteLog ( logmsg ) ; // 로그파일에 기록
    break ;
}
toClient.write ( buf, 0, count ) ;
// 임베디드 시스템으로 데이터 전송
}
fromClient.close ( ) ; toClient.close ( ) ; log.close ( ) ;
} catch ( IOException ex ){
}
} }
```

(알고리즘 1) 임베디드 시스템 서버 구현

6.2 임베디드 시스템 기능구현

임베디드 시스템의 주기능은 차량의 출입관리를 통제하는 것이며, 통제상에 필요한 정보를 서버로부터 얻어오기위해 소켓프로그램을 통하여 서버와 통신한다. 시스템을 가동시킴과 동시에 서버와 통신설정을 한 후 서버로부터 초기화에 대한 정보를 입력받는다. 그 후 차량의 이동이 발생하면 car_in_out_process 내에서 RF 안테나를 통해 들어오는 데이터를 파싱하여 차량의 요청에 따라 처리를 하게 되는데 그 때 RFID data 형식은 <표 1>을 따르는 것으로 한다.

```
임베디드 시스템 서버와 연결설정
init_device ( sockfd ) ;  // 시스템 초기화
car_in_out_process ( stdin, sockfd )
    loop
        Parsing RFID data
getcarstatus ( sockfd, carno ) ;
// 서버로부터 상태정보를 얻어온다.
if ( parking in area of this device ) then
    insert_car ( sockfd, carpos )
    // 서버에 처리를 요청 후 결과를 단말기에 출력
    // 주차할 수 없는 경우의 수 처리 ( 미등록, 공간 부족 등 )
else if ( try to move to other place ) then
    del_car ( sockfd )
    // 서버에 처리를 요청 후 결과를 단말기에 출력
    // 차량이동에 대한 허가 처리(잠김상태차량)
end if
    end loop
 end car_in_out_process
```

(알고리즘 2) 임베디드 시스템 구현

<표 1> RFID data format

	Car No	Request	Car Position	Car Status
설 명	차량번호	입·출요청	차량위치	차량상태
크기(bytes)	3	1	3	2

6.3 XML Data Parsing

여기에서 언급하는 XML 데이터 양식은 임베디드 시스템의 특성을 고려하여 본 시스템에서 사용가능하도록 최소한의 규칙을 정의하여 설계하였다. (알고리즘 3)는 XML 데이터 구성의 예를 보여주고 있고 전체적인 규칙은 1)~4)의 규칙을 따른다.

```
< onecar >
        < idx > 3 < /idx >< carnumber > F4s < /carnumber >
                ……
        < regdate > xxxx/xx/xx/xx/xx < /regdate > < /onecar >
```

(알고리즘 3) XML Data Sample

1) DTD를 필요로 하지 않는다.
2) 속성(값)을 필요로 하지 않는다.
3) 시작태그로 시작한 모든 엘리먼트의 끝은 종료태그에 의해 마크되어야한다.
4) 시작과 종료태그의 이름은 반드시 일치한다.

```
public void ParseCarInfo ( String carinfo ){
    while ( !carinfo.equals ( " " )){
 parseValue ( ) 를 반복호출하여 Element의 컨텐츠들을 얻어온다.
 CarInfo 객체에 하나의 차량정보를 구성 후
HashTable로 객체자료를 저장한다.
}
}
public String parseValue ( String starttag, String endtag, String wholestr ) {
// 하나의 엘리먼트에 대한 컨텐트 값을 추출한다.
 int startoffset = 0, endoffset = 0 ;
 int targetstart = 0, targetend = 0 ;
 for ( ; startoffset < wholestr.length ( ) ; startoffset++ ){
   if ( wholestr.regionMatches ( false, startoffset, starttag, 0,
starttag.length ( ) ) ) break ;
 }
       targetstart = startoffset + starttag.length ( ) ;
       endoffset = targetstart ;
       for ( ; endoffset < wholestr.length ( ) ; endoffset++ ){
          if ( wholestr.regionMatches ( false, endoffset, endtag, 0,
endtag.length ( ) ) ) break ;
       }
       targetend = endoffset ;
       return wholestr.substring ( targetstart, targetend ) ;
}
public String deleteTopone ( String tag, String rtag ){
// 상위 엘리먼트 태그를 제거
int startoffset = 0 ;
       int targetstart = 0 ;
       for ( ; startoffset<rtag.length ( ) ; startoffset++ ){
       if ( rtag.regionMatches ( false, startoffset, tag, 0,
tag.length ( ) ) ) break ;
       }
       targetstart = startoffset+tag.length ( ) ;
return rtag.substring ( targetstart ) ;
}
```

(알고리즘 4) XML Data Parsing API

기존에 존재하는 DOM, SAX 파서의 경우 모바일기기에서 사용하기에는 그 크기가 너무 크다. 그리고 J2ME상에서 운영되는 SAX 기반의 TinyXML 또는 NanoXML 같은 파서가 있으나, J2ME로 가면서 자바의 강점인 플랫폼 독립적인 기능이 사라지면서 각 파서는 플랫폼에 맞게 포팅을 해주는 추가 작업이 필요하게 되었다. 본 구현부분에서는 특정한 플랫폼을 기반으로 하지 않기 위하여 퍼슨널 자바와 호환되는 버전의 API를 이용하여 파싱에 필요한 하부 API들만을 구현하였다.

6.4 PDA에서의 암호화 및 기능 구현

모바일 기기는 PDA와 휴대폰으로 나누어 구현되었다. 휴대폰은 J2ME의 MIDP를 이용하여 구현하였고, PDA는 iPAQ PocketPC상에 EVM(Embedded Virtual Machine)인 Joede 플랫폼위에 구현하였으며 (알고리즘 5)는 ECB운영방식으로 구현한 AES 암호화 API를 설명하고 있다.

(알고리즘 5)에서 암호화 객체 ECBAes는 생성시 영업사원 계정의 비밀번호를 파라메타로 128비트 대칭키를 생성한다. 키와 블록 사이즈는 16Bytes(128bits)로하며, 암호화 및 복호화는 Initial Round, Standard Round, Final Round로 이루어지며 각 라운드마다의 방식은 다른 블록 암호화방식과 틀리다.

```
final static int BLOCK_SIZE = 16 ;
final static int KEY_SIZE = 16 ;
.....
public ECBAes ( String keystr ) {
// 비밀번호를 입력으로 대칭(부분)키 생성
byte [ ] kb = new byte [ KEY_SIZE ] ;
while ( keystr.length ( ) < BLOCK_SIZE ) keystr += " " ;
for ( i = 0 ; i < KEY_SIZE ; i++ )
b [ i ] = ( byte ) keystr.charAt ( i ) ;
key = Rijndael_Algorithm.makeKey ( kb, BLOCK_SIZE ) ;
}

public byte [ ] encryptstr ( String cwriting ) {
// 평문을 파라메타로 전달받는다.
int bs = 0, bc = 0 ;
if ( cwriting.length ( ) %BLOCK_SIZE != 0 ) { // 128bits씩 나눈다.
bs = BLOCK_SIZE-( cwriting.length ( ) %BLOCK_SIZE ) ;
for ( i = 0 ; i < bs ; i++ ) cwriting += " " ;
}
byte [ ] text = new byte [ cwriting.length ( ) ] ;
bc = cwriting.length ( )/BLOCK_SIZE ;
for ( i = 0 ; i < bc ; i++ ) {
byte [ ] pt = cwriting.substring ( BLOCK_SIZE×i,
BLOCK_SIZE×( i + 1 ))).getBytes ( ) ;
byte [ ] ct = Rijndael_Algorithm.blockEncrypt ( pt, 0, key,
BLOCK_SIZE ) ;
for ( int ptoc = 0 ; ptoc < BLOCK_SIZE ; ptoc++ ){
text [ i×BLOCK_SIZE + ptoc ] = ct [ ptoc ] ;
}
}
return text ;
}
public String decryptstr ( byte [ ] cstr ){
StringBuffer cipherText = new StringBuffer ( ) ;
 byte [ ] ct = new byte [ 16 ] ;
 int idx = 0 ;
for ( i = 0 ; i < cstr.length/BLOCK_SIZE ; i++ ) {
for ( int i = 0 ; i < BLOCK_SIZE ; i++, idx++ ) ct [ i ] = cstr [ idx ] ;
byte [ ] pt =
Rijndael_Algorithm.blockDecrypt ( ct, 0, key, BLOCK_SIZE ) ;
StringBuffer sbt = new StringBuffer ( " " ) ;
// 16 blanks
for ( int ptoc = 0 ; ptoc < BLOCK_SIZE ; ptoc++ ) {
char ptemp = ( char ) pt [ ptoc ] ;
sbt.setCharAt ( ptoc, ptemp ) ;
}
cipherText.append ( sbt.toString ( )) ;
}
```

(알고리즘 5) ECB방식의 AES 암호화 알고리즘 API

```
public void actionPerformed ( ActionEvent e ) {
if request online mode then
    Connection ( ) ;
      // 서버와 통신연결
    else if user login then
      new ECBAes ( passwd ) ;
      encryptstr ( id ) ;
      // 자신의 계정을 비밀번호로 암호화하여 로그인
    else if copy a car info from server then
      SaveLocal ( ) ;
      // 연결된 서버에서 차량정보를 PDA로 복사
      // 챠량 검색창 생성
    중략 .....
    else if request local data from PDA then
      LoadLocalData ( ) ;
      // 지역 데이터 파일로부터 판매중인 데이터 로드
    else if request sold car list from PDA then
      SoldLocalData ( ) ;
      // 지역 데이터 파일로부터 판매된 데이터 로드
    end if
    }
    else if a car is sold then
```

```
        OnlineSale ( ) ;
        // 판매된 차량에 대한 서버정보 갱신
    else if search cars from server then
    m_sw = new SearchWin ( "Search Window," this, 1 ) ;
        m_sw.show ( ) ;
```

(알고리즘 6) PDA 기능 구현

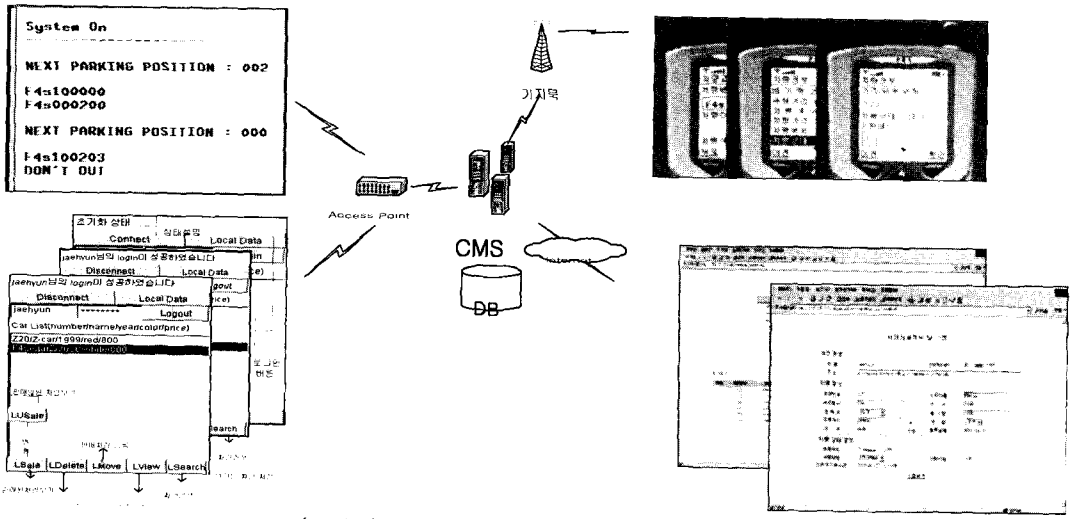
PDA의 기능은 온라인과 오프라인 모드로 운영된다. 온라인모드는 서버와 통신이 연결된 상태에서 차량데이터를 운영하는 접속모드로 연결 및 로그인이 된 상황에서 운영하는 모드이다. 오프라인 모드는 온라인모드 상태에서 자신이 원하는 제품을 PDA로 저장하여 통신 불가지역에서도 차량에 대한 정보를 검색할 수 있도록 하였다. 또한 오프라인 모드에서는 판매된 차량과 미판매된 차량을 구분하여 관리할 수 있도록 하였다. (알고리즘 5)의 actionPerformed 함수내에서 각 event에 대한 처리를 구현하였다. 서버와 통신되는 데이터는 XML 문서형식을 취하며, PDA상에 보관시에는 파일시스템을 이용하고 검색 및 정보수정시에는 해쉬 자료구조를 사용하여 처리하였다.

7. 테스트 및 디버깅

임베디드 시스템의 호스트로 리눅스 서버를 연결하여 RFID로부터 안테나에 읽혀지는 데이터를 전송한 후 그 데이터가 CMS에 정확히 기록되고 모바일 기기(J2ME Wireless Toolkit, PDA)를 통한 정보의 갱신과 검색이 제대로 작동하는지 확인하였다. 또한 기존의 인터넷 라인을 이용하여 통제할 수 있도록 브라우저를 사용하여 데이터를 관리할 수 있는 웹페이지를 구현하여 모바일 기기 및 임베디드 시스템과 연동하는 실험을 하였다. 본 시스템을 구현시에 주요 초점을 맞추었던 부분은 임베디드 시스템과 호환을 위한 데이터 형식부분이었다. 우선 CMS와 임베디드 시스템 사이의 데이터 처리부분은 호스트인 CMS에 무게를 많이 주어 처리후 결과를 임베디드 시스템에 전송함으로써 임베디드 시스템은 출력과 통신부분에만 집중할 수 있게하였다. (그림 8)의 좌측 상단의 그림은 F4s차량의 출입정보를 임베디드 시스템이 읽어와 CMS와 통신하는 과정을 보여준다. 최초 시스템 가동시 CMS로부터 주차할 수 있는 자리의 번호를 출력한다. 다음 두 줄은 F4s차량이 나갔다 들어오는 이동과정의 데이터를 출력하는 것이고, 임베디드 시스템 단말기는 다음 주차할 수 있는 위치를 가리킨다. 이때 다시 F4s차량이 나가려고 시도하지만 이 차량은 관리자에 의해서 잠김(판매대기)상태로써 나가는 것을 허락받지 못한다. 모바일 기기에서의 구현은 임베디드 시스템에 의해 갱신된 정보들을 이용하여 실무에서 사용하는 업무와 연결하는데 집중하였다. 모바일기기를 주로 사용하는 사람은 내/외부 영업 및 사내에서 이동간에 관리를 담당하고 있는 사원으로 정의한다. 위의 영업사원이 사용할 PDA는 다음과 같은 특성을 고려하여 기능을 구성하였다. 첫째, 무선랜 환경에서 데이터베이스에 접근하여 원하는 정보를 검색 및 수정할 수 있어야한다. 둘째, 지역정보의 변화를 기록 갱신할 수 있어야한다. 이러한 기능을 수행하기 위해서 [그림 8]의 좌측 하단에 위치한 PDA 응용프로그램 CMT(Car Management Tool)의 기능은 검색, 수정, 추가, 삭제의 기능들 그리고 기존에 사용하는 데이터형식과 호환할 수 있도록 데이터를 처리하는 부분으로 구현하였다. 위의 테스트에서 PDA와 J2ME Wireless ToolKit에서는 정상적으로 작동을 하였고, 무선랜을 이용할 수 없는 지역에서의 사용을 위한 기능인 지역정보 보관기능에 대해 휴대폰에서는 작은 메모리와 처리속도의 문제를 고려하여 구현을 배제하였으며 그 기능을 PDA에서만 구현하였다. 사무실의 관리자는 현장에서의 상황을 기존의 웹페이지를 통해서 접수 및 통제할 수 있도록 하였다.

8. 결 언

본 연구에서는 임베디드 시스템과 무선 랜을 이용해 이동



(그림 8) 중고 차량관리 시스템 구현

성이 높은 재고단위의 위치관리 시스템을 새롭게 제시하였다. 이동성이 높은 재고단위의 위치관리 모델로서 중고자동차 위치관리 시스템을 설계하고 구현하였는데 이는 다른 이동성이 높은 제품에도 거의 동일하게 적용이 가능하다. 대형 중고자동차 시장을 예로 들면, 현재 이러한 위치관리 시스템은 시도된 적이 없다. 이는 중고자동차 시장의 영세성에도 기인한다고 할 수 있으나, 근본적으로 보관되는 위치가 자주 변하는 중고자동차의 위치를 파악하기가 어렵기 때문이며 RFID에 무선랜을 결합한 임베디드 시스템이 없이는 구현이 거의 불가능한 일이다.

본 논문에서 구현한 시스템을 통하여 기존에 대형 중고자동차 관리에서 다음과 같은 효율성을 발휘할 수 있다.

- 원하는 차량의 위치/상태 정보를 신속히 얻는다.
- 차량의 이동에 대한 정보파악시 인력소모 절감
- 전산화를 통한 전체 차량의 정확한 상태파악
- 원거리의 다중 사무실의 차량정보 공유로 풍부한 물량 보유
- 영업자의 외부영업시 편리한 차량정보관리

시스템의 구현에 있어서 가장 큰 문제점은 임베디드 시스템과 모바일 기기 사이의 효율적인 동기화였다. 본 연구에서는 모바일 기기의 지역 정보가 갱신된 후에 수동으로 동기화를 이루었으나, 본 연구 결과가 원활히 산업화되기 위해서는 장차 자동 동기화에 대한 연구도 진행되어야 할 것이다.

중고자동차 외의 다른 이동성이 높은 재고단위 제품에도 본 시스템의 적용이 가능하지만, 긴 유통 단계를 가지는 일반 상품에의 적용도 연구해 볼 필요가 있다. 생산에서부터 최종소비자에게 판매될 때까지 계속 제품의 보관위치는 변화되는데, 이들 제품의 유통 과정에서 제품의 위치관리가 자동화된다면 소요 인력 감소와 같은 경제적 이득은 물론 생산성 또한 크게 증대될 것이다. TV를 예로 들면, 생산공정이 끝난 TV는 공장의 창고에 저장되고 각 지역의 총판주문에 의해서 공장 창고에서 각 지역의 총판으로 이동된다. 지역이 넓다면 다시 한번 지역내의 업체 관리하의 도매상의 창고로 이동한 후 대리점으로 이동하게 될 것이다. 이 때 본사에서 관리해야 할 영역이 도매점까지라 한다면 이 과정에서 주문 및 물량의 이동량 검사에서 총판 및 도매점에서의 물건 도착을 정확히 조사하기 위해서는 일정한 주기로 사람이 직접 물품에 대한 문서를 감사해야 한다. 또한 창고에서의 문서조작에의한 제품 반출의 위험도 존재한다. 본 시스템을 적용하여 업무를 처리하면, 공장의 생산라인에서 각 제품에 RFID를 부착한 후 공장의 창고로 이동과 동시에 창고에 있는 임베디드 시스템을 통하여 중앙관리시스템의 데이터베이스에 재고의 물량이 기록된다. 각 총판의 주문량은 온라인 및 오프라인을 통하여 주문을 받은 후 물품이동차량 도착시 현장에서 관리자는 PDA를 통하여 임베디드 시스템에게 일정량의 출고를 주문한다. 이 물품들이 총판에 도착하여 총판 창고에 들어갈 때 총판의 임베디드 시스템은 다시 본사의 중앙관리시스템에 물품이 이동했다는 상황을 정확히 기록할 것이다. 이러한 시스템은 생산공정에서 각 대리점의 유통망까지 투명한 작업관리가 가능하게 하며, 본사의 업무도 기존보다 정확하고 신속한 처리가 가능할 것이다.

참 고 문 헌

[1] 이정배, 이두원, "임베디드 시스템 연구동향", 정보처리학회지, 제9권 제1호, pp.13-27, 2002.
[2] 권경희, "Web Engineering", 배움터, 2001.
[3] 박창섭, "암호이론과 보안", 대영사, 1999.
[4] Jennifer Bray, Charles F Sturman, "Bluetooth : Connect Without Cables," Prentice Hall PTR, 2000.
[5] Korea Embedded Linux Project, http://kelp.or.kr.
[6] 디지털 타임즈, http://www.dt.co.kr.
[7] Yu Feng, Dr. Jun Zhu, "Wireless Java Programming with J2ME," Sames, 2001.
[8] A. Menezes, P. vanOorschot, S. Vanstone, "Handbook of Applied Cryptography," CRC Press, pp.250-259, 1996.
[9] Jess Garms and Daniel Somerfield, "Professional Java Security," wrox, pp.51-54, 87-121, 2002.
[10] http://www.jeode.com.
[11] Jan Haagh, <테마특강> 무선LAN의 보안, http://www.etimesi.com/news/detail.html?id=200205130048, 2002.
[12] W. Richard Stevens, "unix network programming," Prentice Hall PTR, 1998.
[13] AES(Advanced Encryption Standard), http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.
[14] 고일한, 황규진, "XN860CM Board Linux Porting Reference Guide," Xiontek, 2001.
[15] (주)FNET(Orinoco국내 총판), http://orinoco.co.kr/product/ap-1000.htm.
[16] 한국 전자통신 연구원, http://www.etri.re.kr.

이 재 현

e-mail : wogusking@dankook.ac.kr
2000년 단국대학교 전자계산학과 졸업 (이학사)
2002년 단국대학교 대학원 전자계산학과 석사과정
2000년~2001년 (주)Asiacontent.com 대리
관심분야 : 컴퓨터 네트워크, XML, 웹 기반 소프트웨어

권 경 희

e-mail : khkwon@dankook.ac.kr
1976년 고려대학교 물리학과(이학사)
1986년 Old Dominion Univ. Dept. of Computer Science(M.S.)
1992년 Louisiana State Univ. Dept. of Computer Science(Ph.D.)
1979년~1984년 산업연구원 연구원
1993년~현재 단국대학교 부교수
관심분야 : 컴퓨터 네트워크, 알고리즘 분석 및 설계, 웹 공학