

工學碩士 學位論文

유비쿼터스 환경의 RFID 인식 단말기의
설계 및 구현

Design and Implementation of RFID Recognition
Terminal Unit in Ubiquitous Environment

指導教授 林 宰 弘

2005年 8月

韓國海洋大學校 大學院

電子通信工學科

催 宰 碩

목 차

Abstract

제 1 장 서론	1
1.1 연구의 배경	1
1.2 연구의 필요성 및 목표	2
제 2 장 RFID 시스템	4
2.1 RFID 시스템의 개요	4
2.2 RFID 표준화	9
제 3 장 RFID 인식단말기의 설계	11
3.1 시스템 설계	11
3.2 데이터베이스 설계	13
제 4 장 RFID 인식 단말기의 구현 및 실험	25
4.1 하드웨어	25
4.2 운영체제	29
4.3 소프트웨어	43
제 5 장 결론	52
참고문헌	53

표 목 차

<표 2-1> RFID의 태그의 능동형과 수동형에 따른 분류	6
<표 2-2> RFID 주파수 대역별 특성 비교	8
<표 4-1> 일반 파워서플라이와 ST-150SL의 비교	27

그림 목 차

<그림 2-1> RFID 활용 사례	4
<그림 2-2> RFID 시스템 구성요소	5
<그림 2-3> RFID 표준화기구 조직표	10
<그림 3-1> RFID 전체 구성도	11
<그림 3-2> 각 부분별 개발범위	12
<그림 3-3> DBMS 서비스의 시작	14
<그림 3-4> 리스너의 시작	15
<그림 3-5> Tmsns.ora 파일의 내용	16
<그림 3-6> Listener.ora 파일의 내용	16
<그림 3-7> 오라클 엔터프라이즈 매니저 콘솔	17
<그림 3-8> 데이터 베이스의 테이블베이스	18
<그림 3-9> 내부스키마 파일정보	19
<그림 3-10> RFID/인식 정보테이블의 ER-다이아그램	20
<그림 3-11> RFID/INFO 테이블	20
<그림 3-12> RFID/LIST 테이블	21
<그림 3-13> RFID/INFO 테이블의 제약조건	22
<그림 3-14> ODBC 데이터 원본관리자	23
<그림 3-15> 오라클 ODBC 드라이버 설정	24
<그림 3-16> RFIDINFO 테이블의 내용	24
<그림 4-1> WEB 6580 타겟보드와 스펙	25
<그림 4-2> 일반 파워서플라이와 ST-150SL	27
<그림 4-3> 하드디스크 타입과 CF 메모리타입	28
<그림 4-4> RFID 연결 케이블과 회로도	28
<그림 4-5> XPE 개발 흐름도	33

<그림 4-6> devices.pmq	33
<그림 4-7> 타겟 디자이너의 개발 흐름도	35
<그림 4-8> 타겟 디자이너	36
<그림 4-9> 컴포넌트 디자이너 개발 흐름도	37
<그림 4-10> 컴포넌트 디자이너	37
<그림 4-11> 컴포넌트 디자이너의 옵션	38
<그림 4-12> 컴포넌트데이터베이스매니저	39
<그림 4-13> Import SLD	39
<그림 4-14> 타겟 디자이너	40
<그림 4-15> 컴포넌트 데이터베이스 중에서 하드웨어 부분	41
<그림 4-16> 컴포넌트 데이터베이스 중에서 소프트웨어 부분	42
<그림 4-17> TagID	43
<그림 4-18> MCom Class 추가	46
<그림 4-19> 쓰레드에서 하는일	46
<그림 4-20> 부모 프로그램에서 하는 일	47
<그림 4-21> Com Port	47
<그림 4-22> 메인 프로그램 디자인	49
<그림 4-23> 최종 디스플레이 및 데이터 흐름	50
<그림 4-24> RFID 인식 단말기 완성품 전면사진	51

Abstract

Across the world today, Ubiquitous technologies are becoming an interesting part of people's lives. Ubiquitous technology is the trend towards increasingly Ubiquitous, connected computing devices in the environment, a trend being brought about by a convergence of advanced electronic and particularly, wireless technologies and the Internet. Role of RFID(Radio Frequency Identification) in Ubiquitous technology environment is perceived as a backbone, in which information and communication flows everywhere, for everyone, at all times. RFID is supported by other similar technologies such as wireless, as hoc and sensor networks, which already play important roles in pervasive computing. Sensing devices, such RFID connected through wireless communication can capture, process and disseminate useful information surrounding human beings.

The sociological impact of Ubiquitous technologies in form of computers may be analogous to two other technologies that have Ubiquitous become technologies. The first is Writing, which is found everywhere from clothes labels to billboards. The second is electricity, which surges invisibly through the walls of every home, office, and car. Writing and electricity become so commonplace and indispensable that we forget their huge impact on everyday life. Similarly, for Computers and RFID, users would start tasking for granted the advantages offered by both in our daily running lives.

In this thesis, I am going to design and to make the RFID recognition terminal unit. RFID recognition terminal unit is used as the windows Embedded XP(XPE) operating system and visual c++ 6.0, target board(web-6580) is implemented for RFID application in Ubiquitous.

제 1 장 서 론

1.1 연구의 배경

유비쿼터스 컴퓨팅은 컴퓨팅을 주관하는 주체가 사람에서 사람과 사물을 포함하는 주변의 모든 것으로 바뀌는 패러다임의 변화이다. 이는 유·무선을 가리지 않는 개념으로, 사용자가 컴퓨터나 네트워크를 인식하지 않은 상태에서 장소와 단말기에 구애받지 않고 네트워크에 접속 가능한 환경을 뜻한다. 이러한 유비쿼터스 환경하의 네트워크 센서장치를 담당하는 기술로서 최근에 정부와 관련업체의 주목을 받고 있는 기술이 바로 무선주파수식별(RFID ; Radio Frequency IDentification)이다. 이 RFID는 정보통신을 포함한 물류·유통·교통·환경 등의 다양한 분야에 적용 가능한 차세대 핵심기술이며, 또한 응용하는 분야별로 주파수 대역을 달리하여 여러 분야에 적용시켜 활용할 수도 있다. 현재 미국, 일본, 유럽 등지에서 RFID의 기술표준화 및 응용사업에 대한 연구가 활발하게 진행되고 있으며, 일부는 상용화되어 인건비 절감, 재고처리의 효율화 등의 효과도 나타나고 있다.

우리나라에서의 RFID 현황은 정보통신부와 산업자원부가 RFID 분야에서 부처간 공조체제를 구축하면서 RFID 사업의 활성화에 나서기로 했다. 전자태그를 기반으로 하여 생활의 모든 분야에 정보화 확산을 목표로 하는 'U-센터 네트워크 계획'을 추진 중이며, 공공기관을 대상으로 하는 'RFID 시범사업'도 추진하고 있다[1][2].

미국의 경우 1991년부터 유비쿼터스 컴퓨팅의 실현을 위한 연구를 추진해왔으며. 그러한 계획의 일환으로 국방부 산하 고등연구계획국과 국가표준기술원(NIST; National Institute of Standards and

Technology)의 정보 기술 응용국(ITAO; Information Technology and Application Office)이 연구자금을 지원하고 있으며, 여러 주요대학과 민간 기업에서 다양한 프로젝트를 수행중이다. 현재 일부 실용화의 단계에 들어선 미국의 예를 들자면, 세계 점유율 1위의 모 대형 창고형 할인마트에서 바코드를 대체하여 재고관리 목적의 독특한 인식번호를 칩안에 내장하여 사용하는 RFID 태그를 도입하여 사용 중이다. 이 예를 보아 알 수 있듯이 미국의 RFID 정책은 기술적 비전 제시와 필요한 부분에서의 조기응용이 주를 이루고 있다.

그와는 다르게 일본은 광, 모바일, 센서, 초소형 기계장치, 가전, 부품, 정밀 가공 등을 연계시켜 국가차원의 정책적 추진에 비중을 두고 사업을 진행하고 있다. 마이크로 센서기술을 이용한 사람과 사물간의 통신, 그리고 그와 연관된 주변기술도 중요하다고 인식하고 있으며, 최종적인 목표는 새로운 산업 및 비즈니스 시장의 창출과 편리하고 풍요로운 라이프 스타일의 실현, 그리고 일본이 직면하고 있는 고령화, 교통 혼잡 등의 문제를 해결하는 것이라고 볼 수 있다[3][4].

이처럼 해외 주요 국가들의 경우 각 국가별 목표에 맞게 표준화와 그에 관련된 연구를 추진하고 있는 실정이며, 응용분야 역시 물류시스템과 같은 특정분야에서는 실용화가 진행되고 있다.

1.2 연구의 필요성 및 목표

RFID는 제 2차 세계대전 중에 영국이 자국 전투기 식별을 위해 개발하여 사용한 것이 최초의 도입사례로 알려져 있으나, 태그 비용이 너무 고가였기 때문에 수십년간 시장에서 실용화되기는 어려웠다. 그 후 70년대에 미국에서 가축관리나 철도차량식별등의 응용분야에 대해 기초기술개발이 시작되었고, 80년대에 들어와 제조현장에서 물

류 관리 자동화 등에 응용되기 시작했다. 현재는 주차관리 시스템, 출입통제 시스템, 대도시의 지능형 교통시스템, 동물인식시스템, 공장 자동화 시스템 등의 다양한 분야에서 연구되고 있으며, 그 응용분야가 확대될 것으로 보인다[5].

그러나 RFID가 바코드처럼 일상생활에서 사용될 정도로 활성화되기 위해서는 기술개발의 선도에 있는 국가 간의 표준화가 우선되어야 하며, RFID의 기술개발과 실용화에 따른 가격경쟁력이 확보되어야 한다. 이런 문제점들이 효과적으로 예방되어 RFID의 이용이 증대된다면 실생활에 매우 유익하게 사용될 것으로 보인다.

본 논문에서는 RFID 기술의 주요 기술중 하나인 RFID 인식 단말기를 설계, 구현하였으며, 인식 단말기의 운영체제는 안정성과 신뢰성이 높은 윈도우즈 임베디드 XP를 이용하였다. 본 논문의 구성은 제 2 장에서 RFID 시스템의 기본구성과 신기술, 표준화 동향에 대해 기술하였고 제 3 장 및 제 4 장에서는 RFID 인식 시스템에 대한 설계와 구현 내용을 기술하고 그 실험 결과에 대하여 고찰하였으며, 제 5 장은 결론과 향후 연구계획에 대하여 기술하였다.

제 2 장 RFID의 시스템

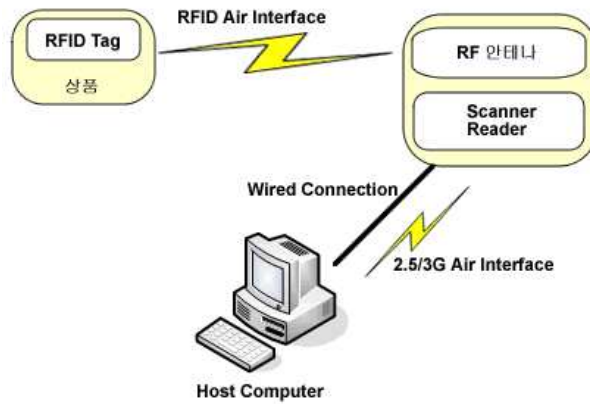
2.1 RFID 시스템의 개요

RFID 기술은 유비쿼터스 환경 하에 인간과 주변기기와의 데이터 통신을 담당하는 핵심기술이라고 할 수 있다. RFID 기술은 태그의 칩에 대용량의 정보를 삽입하여 기존에 사용되고 있는 바코드보다 많은 정보의 전송이 가능하다. <그림 2-1>에서 볼 수 있듯이 보안/출입 제어, 동물 추적, 차량 안전장치, 요금 징수 등 RFID의 활용범위는 매우 넓다. 인식률에 있어서도 자기카드나 IC카드와 마찬가지로 99.9%이상으로 정확도가 높으며, 사용기간 및 데이터 저장능력 또한 여타 매체에 비해 탁월하다[4].



<그림 2-1> RFID 활용 사례

RFID 기술은 Micro Chip을 내장한 태그, 라벨, Card등에 저장된 Data를 무선 주파수를 이용하여 리더기에서 자동 인식하는 무선 인식기술이다. RFID 시스템은 <그림2-2>과 같이 요약하여 표현할 수 있다.



<그림 2-2> RFID 시스템 구성요소

RFID의 시스템은 크게 안테나가 포함된 리더기와 무선자원을 송수신할 수 있는 안테나, 그리고 정보를 저장하고 프로토콜로 데이터를 교환하는 태그, 서버 등으로 구성된다. 각 부분의 기능을 보면 리더기는 RFID 태그에 읽기와 쓰기를 하는 장치이고, 안테나는 정의된 주파수와 프로토콜로 태그에 저장된 데이터를 교환하는 역할을 한다. 태그는 데이터를 저장하는 RFID의 핵심기능을 담당하고 있다. 이 태그는 IC 칩과 안테나로 구성되어 있고 다양한 모양과 크기가 있다 [6].

RFID 시스템은 무선접속 방식에 따라 두 가지로 구분할 수 있다. 1m이내의 근거리 전자기파 방식인 상호유도(Inductively Coupled) 방식과 3m~10m범위의 중·장거리 전자기파(Electromagnetic wave) 방식이 그것이다. 상호유도 방식은 코일 안테나를 사용하며, 태그의

IC칩을 동작시키기 위한 모든 에너지를 리더기에서 공급받는 수동형이다. 전자기파 방식은 고주파 안테나를 이용하여 쌍방 간의 무선접속을 하며, 태그의 IC를 구동시키기 위한 전력을 리더기에서 공급받지 못하는 경우도 있기 때문에 부분적으로 능동형인 경우도 있다 [7][8].

<표 2-1>은 태그에 전원을 어떻게 공급하느냐에 따라서 수동형(Passive)과 능동형(Active)으로 구분되었다. 수동형은 태그의 IC칩이 동작하는데 필요한 모든 에너지는 리더기에 의해 공급되어진다. 따라서 리더기기의 안테나 코일은 주변지역에 강한 자기장을 발생한다. 방출된 자기장의 일부분이 리더기와 떨어져 있는 태그의 코일 안테나에 유도성 전압을 발생, 정류된 후 IC칩을 위한 에너지로 공급된다. 능동형에 비해 가격이 싸고 매우 가벼우며, 반영구적으로 사용이 가능하다. 단점은 인식거리가 짧고 리더기에서 전력의 소모가 더 심하다는 것이다. 이와는 달리 능동형은 자체 전원(Battery)을 필요로 하는 방식으로 리더기의 필요전력을 감소시키고 인식거리를 멀리할 수 있는 장점이 있다. 그 반면에 자체적으로 전원을 공급해야 하기 때문에 제조단가가 비싸고 동작시간의 제한을 받는다[9][10].

<표 2-1> RFID 태그의 능동형과 수동형에 따른 분류

방식	능동형(Active)	수동형(Passive)
특징	태그에서 신호 자체송신 자체 배터리	리더의 신호를 변형반사 리더의 전파신호로 전원공급
장점	장거리(3m~10m) 전송가능 센서와 결합가능	낮은 생산가격 낮은 유지비용
단점	배터리로 인한 가격상승 동작시간 제한	장거리 전송 불가 센서추가 불가
적용분야	과학실험분야, 환경감시, 의료등 연구개발분야	물류관리, 교통, 보안, 재고관리, 전자상거래등 일반분야
주파수 분배		908.hMHz~914MHz(유통,물류)

<표 2-2>에서 확인할 수 있듯이 사용 주파수별로 살펴보면, 장파(125~134KHz), 중파(13.56MHz, 433.92MHz) 및 초고주파(860~930MHz, 2.45GHz) 등 여러 무선 주파수대에서 작동하는 시스템이 개발되어 상용화 되어왔다. 각각의 주파수대 제품들은 나름대로의 특징을 가지고 있는데, 그에 따라서 응용되는 분야가 구분이 된다. 저주파대 제품은 사용거리가 짧고, 데이터 전송속도가 낮지만 양산이 용이하고 가격이 저렴한 특징을 가지고 있기 때문에 출입 통제 및 보안, 동물의 인식 및 추적, 작업의 자동화, 재고관리, 재고자산 추적 등과 같은 분야에 적합하다. 주로 13.56MHz를 사용하는 중파대 제품은 출입 통제 및 보안, 스마트 카드 등에 사용되며 최근에는 물류시스템 관리에도 사용되기 시작하였다. 특히 미국은 433.92MHz의 능동형 시스템을 이용 항만의 컨테이너 관리에 사용하고 있다. 900MHz~수GHz 대역의 주파수를 사용하는 초고주파 제품은 10m까지의 긴 인식거리와 빠른 데이터 전송속도를 제공한다[11][12].

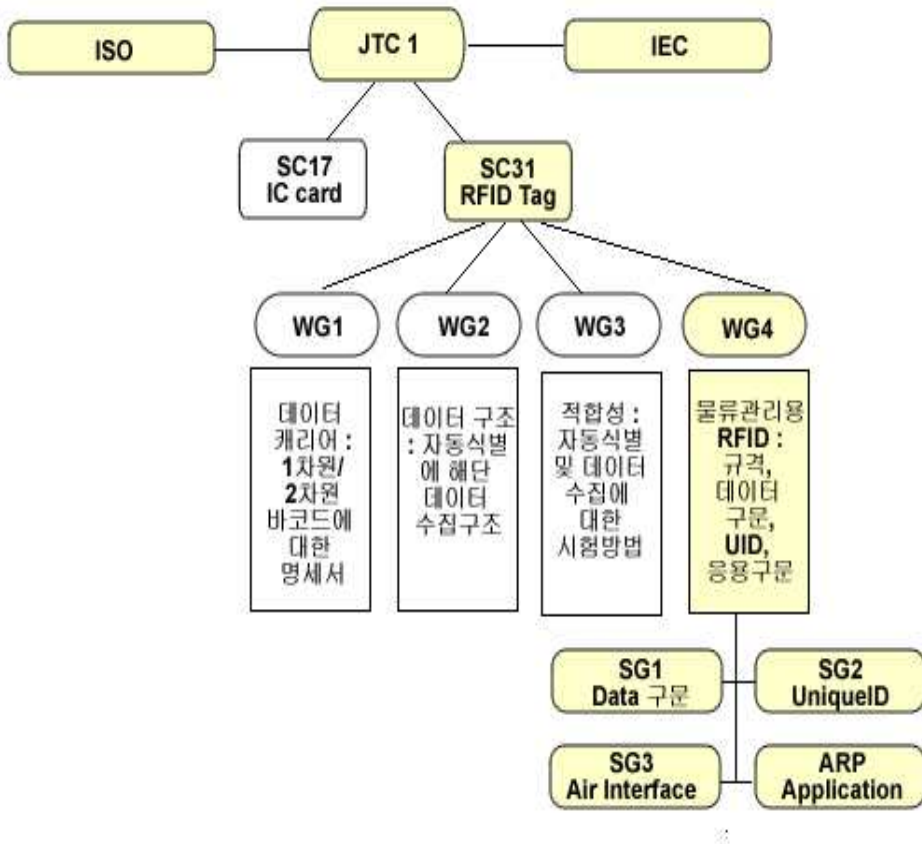
<표 2-2> RFID 주파수 대역별 특성 비교

주파수	저주파(LF)	고주파(HF)	극초단파(UHF)		마이크로파
	125,134kHz	13.56MHz	433.92MHz	860-930MHz	2.45GHz
인식거리	< 60 Cm	약 60 Cm	약 50-100 Cm	약 2.5m-10m(수동)	약 1m(수동)
일반특성	· 비교적 고가 · 환경에 의한 성능저하 거의 없음	· 저주파보다 저가 · 짧은 인식거리와 다중태그 인식이 필요한 응용 분야에 적합	· 긴인식거리 · 실시간 추적 및 컨테이너 내부 습도, 충격 등 환경센싱	· IC 기술 발달로 가장 저가로 생산가능 · 다중태그 인식거리와 성능이 가장 뛰어나	· 900대역태그와 유사한 특성 · 환경에 대한 영향을 가장 많이 받음
동작방식	수동형	수동형	능동형	능/수동형	능/수동형
적용분야	공장자동화 출입통제/보안 동물관리	수화물관리 대여물품관리 교통카드 출입통제/보안	컨테이너 관리 실시간 위치 추적	공급망관리 자동통행료 징수	위조방지
인식속도	저속 <----->고속				
환경영향	강인 <----->민감				
태그크기	대형 <----->소형				

2.2 RFID 표준화

바코드 및 RFID 등의 자동인식 및 데이터 획득 시스템에 대한 AIDC(Automatic Identification and Data Capture) 기술은 상품 공급망 활동에서 사람의 작업이나 판단을 궁극적으로 배제하고 상품이 갖는 정보를 자동적으로 취득하여 온라인으로 관련정보를 처리하는 자동처리시스템 기술이다[13].

국제적으로 AIDC 기술의 표준화는 ISO(International Organization for Standardization)와 IEC(International Electronic Commission)이 주도하고 있다. 양대 기구에서 1996년 3월 JTC(Joint Technical Committee)내에 AIDC 기술표준화를 위한 31번째 분과 위원회를 설립하고 바코드 및 물품관리용 RFID에 대한 국제 표준 활동에 착수하였다. SG31/WG4 안에 다시 4개의 서브그룹이 있어 분야별로 표준화를 진행하고 있다. RFID 시스템의 핵심인 주파수 대역별 Air Interface의 표준화는 SG3에서 진행 중이며, 시스템 간 데이터 프로토콜 표준화는 SG1, RFID 태그의 유일식별을 위한 번호부여 방법 표준화는 SG2에서 진행되고 있다. RFID 활용을 위한 요구사항을 명확히 하기 위해 별도의 ARP(Application Requirement Profile) 그룹이 있어 표준적 응용조건도 논의되고 있다. <그림 2-3>은 ISO와 IEC의표준화기구의 조직표이다. 그 외 물품관리용 RFID 표준화는 ISO를 주축으로 국제표준화 기구인 IEC, ANSI 등의 지역 표준화 단체와 AIM Global Network, 국제 EAN 협회(EAN international), UCC(Uniform Code Council Inc.)등의 민간 표준화 추진협회가 협력하여, 무선시스템에 대한 논의와 사용주파수의 전 세계 통일화에 관한 논의를 진행하고 있다[10][14]-[16].



<그림 2-3> RFID 표준화기구 조직표

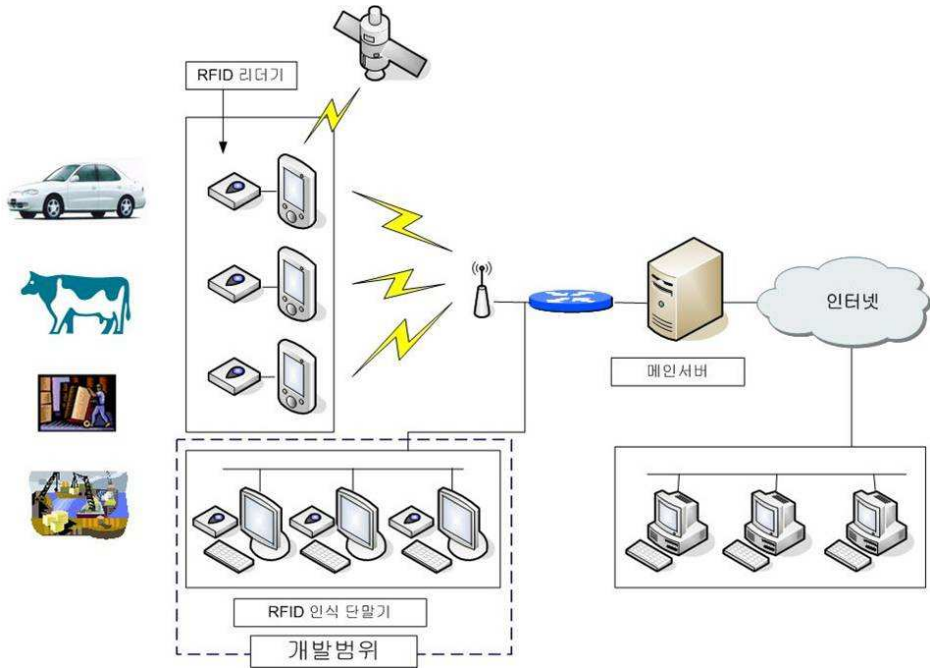
제 3 장 RFID 인식 단말기의 설계

3.1 시스템 설계

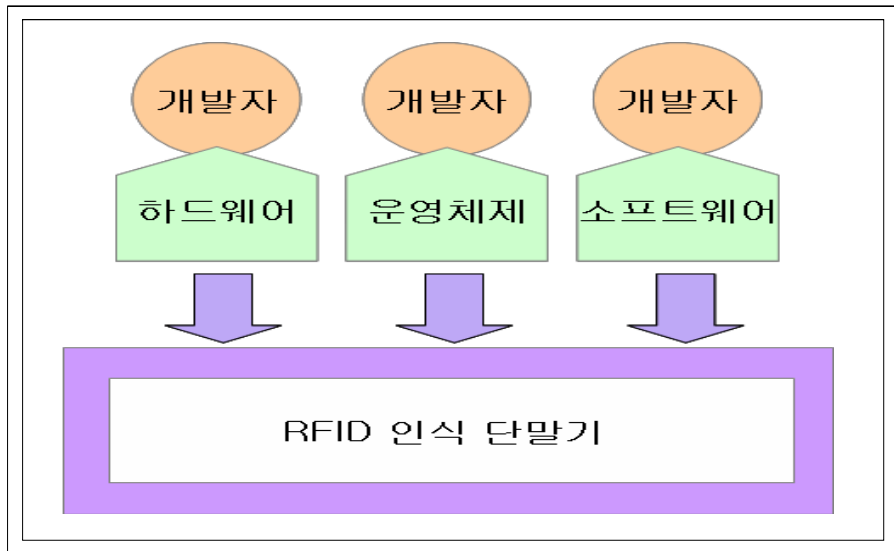
본 논문에서 제작하는 RFID 인식 단말기는 고정된 단말기 형태로 내부에는 산업용 보드와 하드디스크가 내장되어 있다. 운영체제는 임베디드 XP를 사용하였다. 전원은 AC220V에서 사용하는 파워 서플라이를 사용하여 실제 사용현장에서 모니터와 키보드, 마우스를 이용하여 일반 PC와 마찬가지로 사용할 수 있게 설계를 하였다.

본 논문에서 구현하는 범위는 <그림 3-1>의 점선 부분인 RFID 인식 단말기 부분이다. 단말기와 RFID 리더기와는 RS-232C 직렬 통신을 함으로써 RFID 태그 정보를 숫자형태로 단말기로 전송한다. 단말기는 입력된 데이터 값으로 메인 서버에 TCP/IP 통신을 하여 각각 정의된 테이블 값에서 RFID 태그 정보들을 다시 단말기로 받아들여 처리하고 보여준다. 단말기 내부에서 +12V 전원을 RFID 리더기로 제공하며 RFID 리더기는 단말기가 제어를 하게한다.

시제품 구현은 효율적인 시스템의 통합을 위해서 <그림 3-2>에서처럼 하드웨어 개발, 임베디드 XP를 이용한 운영체제 개발, 소프트웨어 개발의 3가지로 구분하였다.



<그림 3-1> RFID 전체 구성도



<그림 3-2> 각 부분별 개발범위

3.2 데이터베이스의 분석 및 설계

3.2.1 데이터베이스의 구성 및 설계

RFID 시스템에서 사용되는 데이터베이스의 설계에 있어서 주요 사항으로 물류시스템과 연동할 수 있는 데이터베이스의 설계에 있다. 따라서 RFID 태그 정보에 대한 테이블과 인식 리스트 테이블을 별도로 설계하고 참조키를 이용하여 관련 정보의 검색을 지원한다.

데이터베이스 관리 시스템인 오라클 서버는 다중 접속자의 지원과 클라이언트로부터의 정보 요청에 따른 태그의 식별 정보의 전송을 담당하고, 클라이언트에서 전송된 태그 인식 리스트 정보를 받아 인식 리스트 테이블에 저장한다. 이 테이블의 정보는 물류 시스템의 응용 프로그램으로부터 요청이 있을 때 지원할 수 있다.

3.2.1.1. 데이터베이스관리 서비스

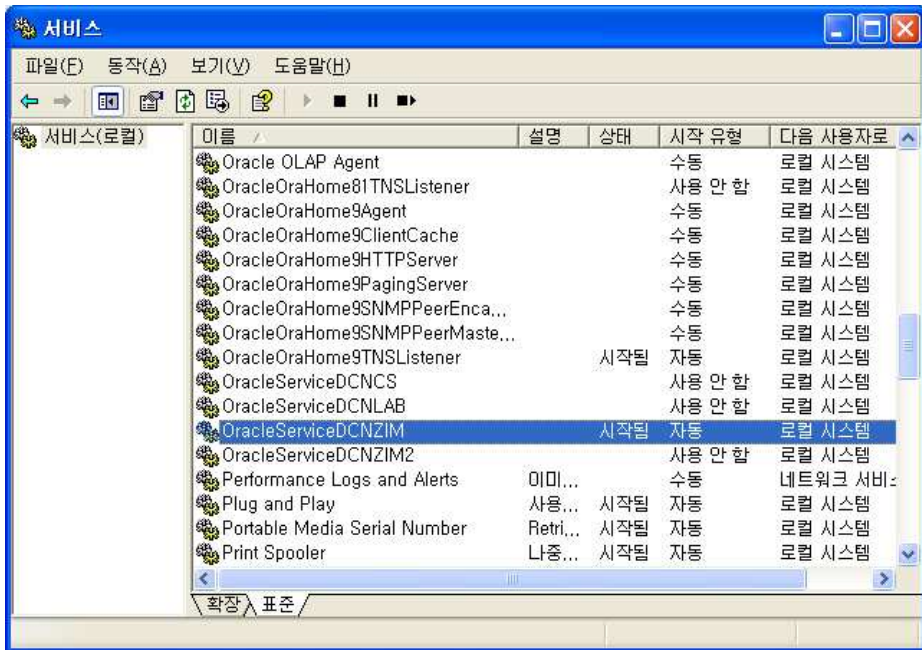
데이터베이스를 관리하는데 필요한 모든 일을 수행하는 시스템을 데이터베이스관리시스템(DBMS)라 하며, DBMS를 제고하는 회사들이 있는데 대표적으로 Oracle, Microsoft, Sybase 등이 있다.

사용자가 DBMS를 제어하기 위해서 DBMS에 명령을 내릴 때 사용하는 언어를 데이터베이스 관리 언어라고 한다. 데이터베이스 관리 언어는 데이터베이스에 저장될 데이터의 구조정의, 데이터의 추가, 변경, 삭제 명령을 비롯하여 여러 가지 다양한 방법을 통한 데이터 검색 등 데이터베이스 관리에 필요한 명령을 포함하고 있다.

데이터베이스의 설계에 앞서 프로그램의 설치 과정이 필요하다. 서버의 설치에 일반 데스크 탑 환경의 운영체제하에 설치가 가능하며,

윈도우 XP 운영체제 환경에서 설치하였다.

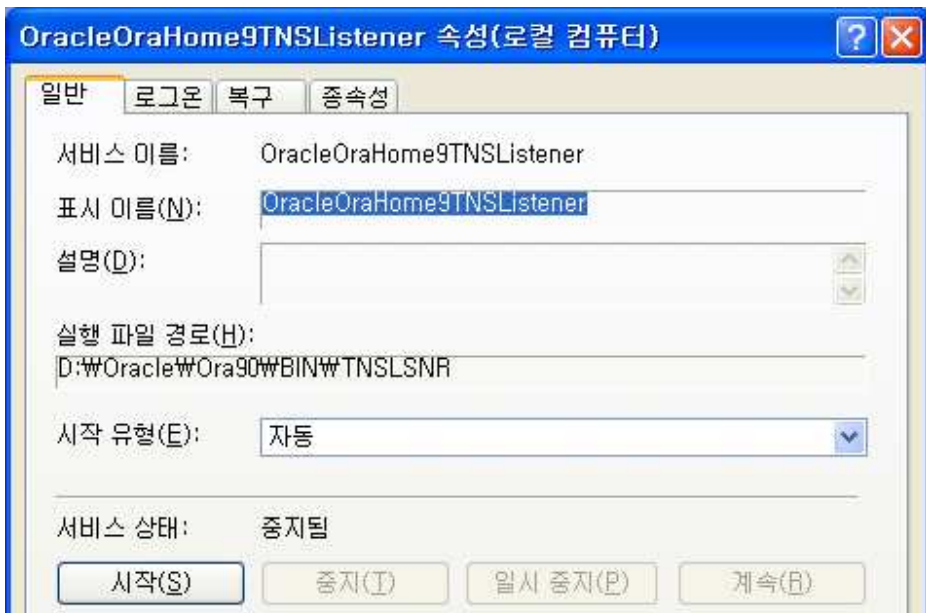
<그림 3-3>은 DBMS의 서비스를 구동하기 위해 설정해야하는 부분으로 설치된 오라클 서비스의 항목과 오라클 리스너를 동작시킨다. 이 과정에서 수행되는 리스너는 외부의 네트워크로부터 접속 요청에 따른 응답과 연결을 담당하며, 오라클 서버의 기능을 이용할 수 있도록 요구되는 아이디와 암호를 인증하고 SQL 질의 검색 등의 서비스를 수행한다.



<그림 3-3> DBMS 서비스의 시작

데이터베이스 애플리케이션 프로그램의 역할은 사용자 인터페이스를 통해 데이터베이스 관리 언어를 생성해 내고 이를 DBMS에 전달함으로써, 데이터베이스를 제어하게 된다. 사용자는 복잡한 데이터베이스의 내부 엔진은 몰라도 데이터베이스를 구성하고 데이터를 입력, 수정, 삭제, 조회하는 명령으로 데이터베이스를 사용할 수 있다.

<그림 3-4>와 같이 리스너를 비롯한 오라클 서비스의 속성 창에는 서비스의 이름을 확인할 수 있으며, 서비스의 상태에 대하여 시작, 중지, 일시중지, 계속의 버튼을 선택할 수 있다. 컴퓨터가 처음 시작부터 자동으로 서비스와 리스너를 구동하려면 시작 유형을 자동으로 설정하여야 한다.



<그림 3-4> 리스너의 시작

<그림 3-5>의 리스너 관련 파일 중에 tmsns.ora의 내용을 살펴보면 오라클 데이터베이스의 인스턴스가 MRKIM이라는 것을 확인할 수 있다. MRKIM의 구성에서 어드레스 항목의 프로토콜이 TCP일 경우 외부의 네트워크로부터의 연결을 설정할 수 있으며, HOST 필드의 203.230.253.89처럼 해당 컴퓨터의 IP 주소를 입력하면 된다. 포트는 기본으로 1521로 되어 있으며, 필요시 변경할 수 있다.

CONNECT_DATA 라인에서 서비스 이름을 mrkim으로 하여 여러

개의 데이터베이스가 설치되었을 경우 특정 인스턴스를 가리키도록 하여야 한다.

```
MRKIM =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 203.230.253.89)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = mrkim)
    )
  )
)
```

<그림 3-5> Tmsns.ora 파일의 내용

<그림 3-6>과 같이 정의된 기본 리스너인 LISTENER와 시스템 식별 번호 (SID; System IDentification)로 정의된 리스너로 구성된 다. SID_LIST_LISTENER의 SID_DESC 항목에서 글로벌 데이터베이스의 이름과 오라클 홈 디렉터리, SID_NAME을 설정한다.

```
LISTENER =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = 203.230.253.89)(PORT = 1521))
  )

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = dcnzim)
      (ORACLE_HOME = D:\Oracle\Ora90)
      (SID_NAME = dcnzim)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = mrkim)
      (ORACLE_HOME = D:\Oracle\Ora90)
      (SID_NAME = mrkim)
    )
  )
)
```

<그림 3-6> Listener.ora 파일의 내용

3.2.1.2. 엔터프라이즈 매니저 접속

설정이 완료된 후 데이터베이스의 접속과 관리는 오라클 엔터프라이즈 매니저 콘솔을 이용하는 방법과 SQL*Plus를 이용하는 방법으로 나눌 수 있다. 오라클 엔터프라이즈 매니저 콘솔 모드는 그래픽 환경의 지원으로 초보자도 쉽게 접근할 수 있으며, 관리 및 운영과 검색 등을 지원한다. SQL*Plus는 텍스트 모드에서 오라클 서버에 접속하여 관리 및 운영과 질의 검색을 수행할 수 있으며, 다양한 명령어의 지원을 통하여 보다 더 전문적으로 제어할 수 있다.

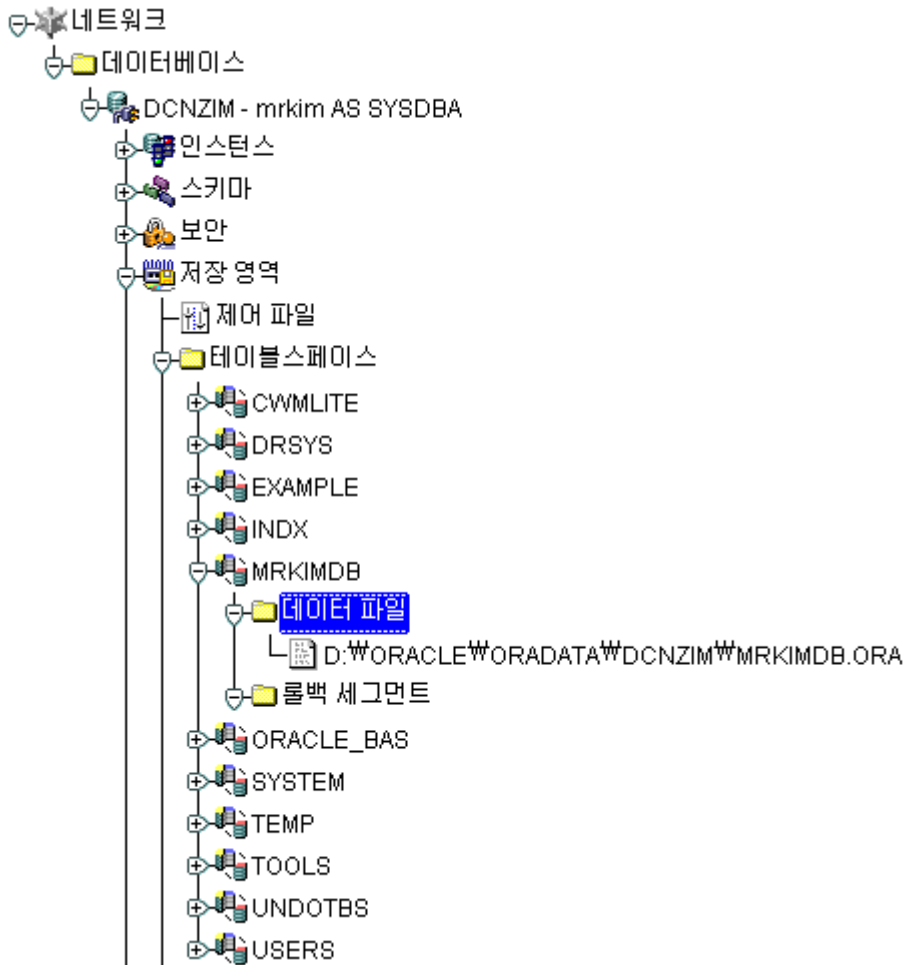
오라클 엔터프라이즈 매니저는 <그림 3-7>와 같이 접속 창으로부터 네트워크, 데이터베이스, DCNZIM 으로 계층화 되어 있으며, 해당 인스턴스인 DCNZIM을 더블클릭 하여 접속할 수 있다. 데이터베이스 접속 정보 대화상자를 통하여 사용자 이름과 암호를 입력하고, 접속 유형으로 SYSDBA라는 DB관리자 권한으로 연결하여야만 데이터베이스의 리소스 자원에 대한 제어가 가능하다.



<그림 3-7> 오라클 엔터프라이즈 매니저 콘솔

새로운 사용자를 등록할 때 아이디와 암호, 접근권한을 명시하여야 하며 생성된 아이디로 재접속하여 데이터베이스를 설계하여야 한다.

데이터베이스를 생성하기 위하여 먼저 수행해야 할 작업은 저장영역에서 물리적 저장 공간의 할당 부분이다. <그림 3-8>과 같이 테이블스페이스 항목에서 다중 사용자 중에 자신이 접속한 아이디 항목 아래에서 할당된 정보를 확인할 수 있다.



<그림 3-8> 데이터베이스의 테이블스페이스

데이터베이스가 설치된 위치가 D:\> 드라이브일 경우 <그림 3-9>과 같이 D:\ORACLE\ORADATA\DCNZIM\MRKIMDB.ORA 로 저장된다. 내부 스키마의 파일 정보는 <그림 3-9>와 같이 설정한 테이블스페이스의 할당 공간이 5 MB이며, 크기 항목을 사용하여 재설정할 수 있다. 처음 공간이 할당된 후 사용되는 용량이 0.063%로 테이블에 자료를 입력할 경우 사용량을 알 수 있다.

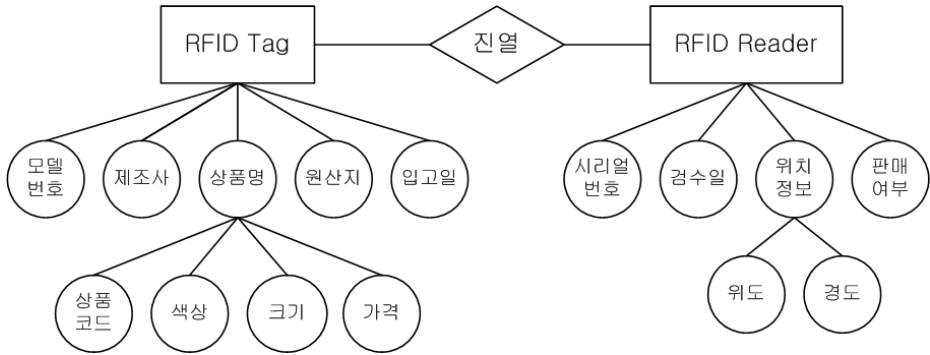
파일 이름	파일 디렉토리	크기	사용됨
MRKIMDB.ORA	D:\ORACLE\ORAD5	5 MB	0.063

<그림 3-9> 내부스키마 파일 정보

3.2.2 데이터베이스 테이블의 설계

RFID 시스템에서 사용되는 데이터베이스 테이블은 RFID 태그에 대한 상세 정보를 저장하는 RFIDINFO 테이블과 RFID 리더기로부터 인식되어진 DPA의 클라이언트 프로그램으로부터 처리되는 RFIDLIST 테이블로 구성된다.

ER(Entity Relationship)-다이아그램은 RFID 태그와 RFID 리더기로부터 인식된 태그 정보에 대한 테이블의 구성이다. 개체의 속성은 기본 정보 테이블인 RFID 태그와 참조 테이블인 RFID 리더 테이블은 <그림 3-10>와 같이 구성된다.



<그림 3-10> RFID 태그/인식 정보 테이블의 ER-다이어그램

<그림 3-11>의 RFIDINFO 테이블은 제약 조건으로 RFID 태그의 시리얼번호인 MODELNO를 Primary Key로 설정하고, 개체 타입의 속성들을 제품번호(MODELNO), 제품명(COMMODITY), 제조사(MAKER), 원산지(SOURCE), 상품코드(PCODE), 색상(COLOR), 크기(SIZE), 입고일(INDATE), 가격(PRICE)으로 정의하였다.

이름	데이터 유형	크기	소수점 이하 자릿수	널입니까?	기본값
MODELNO	CHAR	10			
COMMODITY	VARCHAR2	50		✓	
MAKER	CHAR	30		✓	
SOURCE	CHAR	30		✓	
PCODE	CHAR	20		✓	
COLOR	CHAR	30		✓	
SIZE	VARCHAR2	50		✓	
INDATE	DATE			✓	
PRICE	NUMBER	10	0	✓	

<그림 3-11> RFIDINFO 테이블

<그림 3-12>의 RFIDLIST 테이블의 개체 속성들은 시리얼번호 (SREALNO), 검사일(CHKDATE), 위치정보/위도(POSLAT), 위치정보/경도(POSLONG), 판매여부(SALESTATE)로 정의하였다.

SALESTATE는 기본 값으로 'N'을 설정하여 최초 판매되지 않은 제품으로 저장한다. 이후 판매되었을 경우 'Y'값으로 변경함으로써, 제품의 판매 여부를 확인할 수 있는 필드이다.

The screenshot shows a database management interface with the following details:

- 이름: RFIDLIST
- 스키마: MRKIM
- 테이블스페이스: MRKIMDB
- 테이블: 표준 인덱스를 사용하여 구성됨 (OT) 추상적 데이터 유형

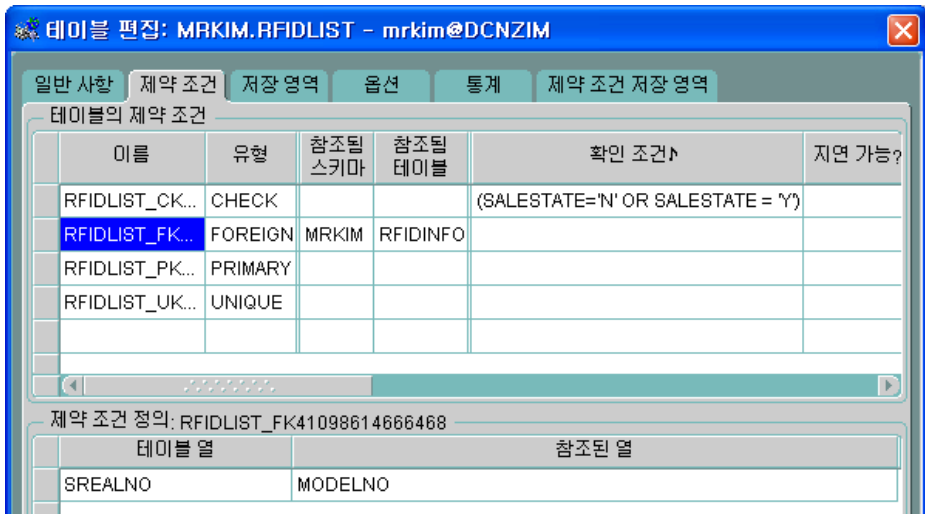
이름	데이터 유형	크기	소수점 이하 자릿수	널입니까?	기본값
SREALNO	CHAR	10			
CHKDATE	DATE			✓	
POSLAT	CHAR	10		✓	
POSLONG	CHAR	11		✓	
SALESTATE	CHAR	1		✓	'N'

<그림 3-12> RFIDLIST 테이블

RFIDLIST 테이블은 RFID 리더기로부터 인식되어진 태그의 고유한 시리얼 번호인 8자리 코드 값을 태그정보의 모델번호 (MODELNO)와 참조한다. 이렇게 인식되어진 시리얼 번호를 데이터 베이스 서버와 TCP/IP 소켓 통신으로 서버 측에 태그 정보를 요청하며, 서버는 응답으로 태그 정보 필드의 내용을 전송한다.

<그림 3-13>에서는 RFIDLIST 테이블에 대한 제약조건의 설정이다. 첫 번째의 테이블 제약 조건은 CHECK 유형으로 SALESTATE

필드에 입력될 수 있는 값이 'N' 또는 'Y'로 판매 여부는 오직 두 가지 경우만 선택하게 한다. 두 번째의 테이블 제약 조건은 외래키의 지정이다. 참조되는 테이블의 MRKIM 스키마의 RFIDINFO 테이블로 테이블의 열은 SREALNO 필드와 MODELNO 필드가 참조된다. 세 번째 테이블 제약 조건은 기본키로 현재 테이블의 SREALNO 필드가 유일하기 때문이다. 네 번째의 테이블 제약 조건은 기본 키의 무결성을 위하여 UNIQUE 유형으로 지정한다.

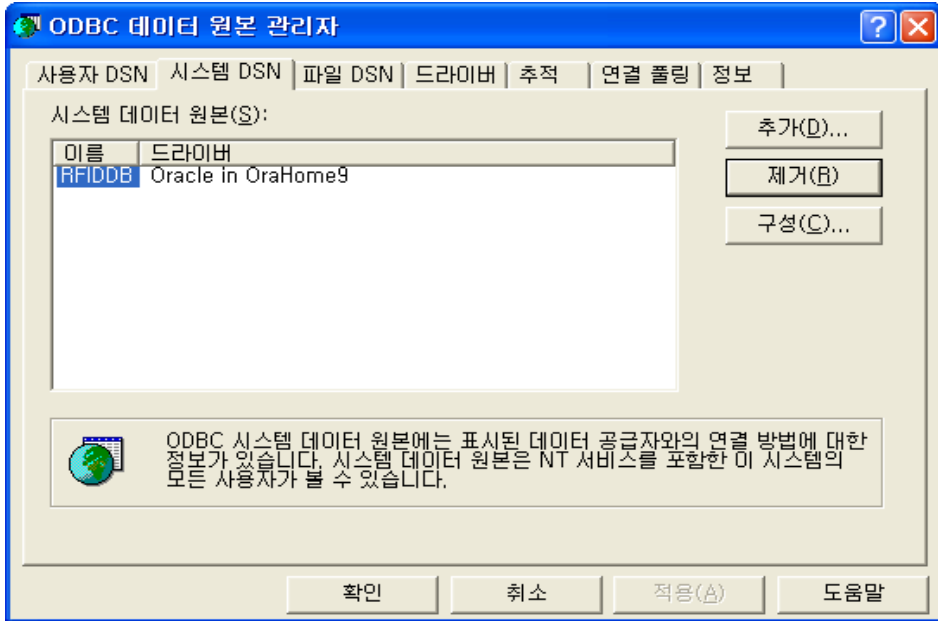


<그림 3-13> RFIDLIST 테이블의 제약조건

3.2.3. ODBC 데이터 원본 관리자

설계된 데이터베이스 테이블을 프로그램과 연결하기 위하여 ODBC 설정을 해야 한다. 제어판의 서비스 항목에 있는 ODBC 데이터 원본 관리자 대화상자에서 시스템 DSN 항목에 시스템 데이터 원본을 추가한다. <그림 3-14>와 같이 ODBC 시스템 데이터 원본 관리자는

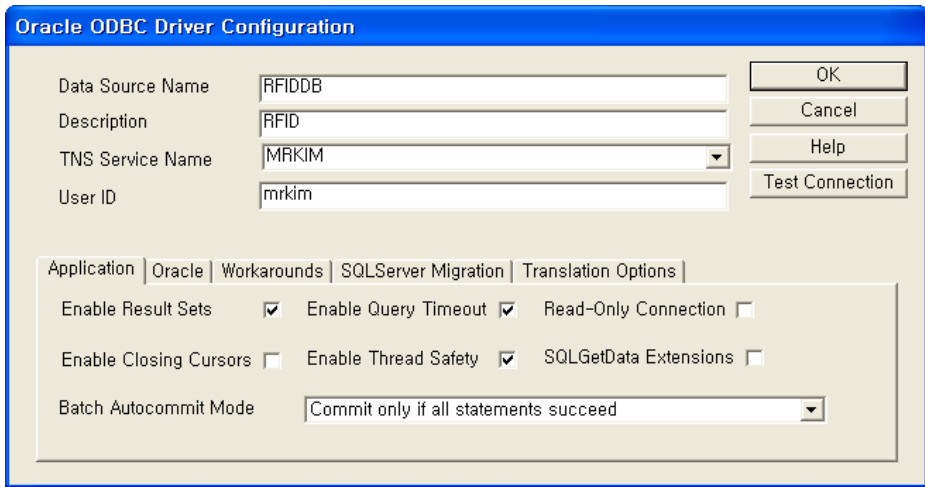
데이터 공급자인 오라클의 드라이버 'Oracle in OraHome9'와 연결 방법에 대한 정보를 제공한다. 이 시스템 데이터의 원본은 NT 서비스를 포함하여 이 시스템의 모든 사용자가 이용할 수 있다.



<그림 3-14> ODBC 데이터 원본 관리자

Visual Studio.NET 2003의 VB.NET에서 다음의 ODBC 데이터 원본과 연결하여 ADODB .Connection으로 접속하며, 생성된 테이블에 대한 접근과 질의어의 검색 등을 수행한다.

<그림 3-15>의 오라클 ODBC 드라이버 설정에서 데이터 소스 이름으로 'RFIDDB'를, TNS 서비스 이름으로 'MRKIM'을 사용자 아이디는 'mrkim'을 설정한다. 설정된 정보가 정확한지의 테스트를 위해 오른쪽의 테스트 커넥션 버튼을 눌러 확인할 수 있다. 정상적으로 연결 접속이 되면 성공했다는 메시지가 나오며, 만약 설정한 값이 틀렸을 경우 다시 설정하여야 한다.



<그림 3-15> 오라클 ODBC 드라이버 설정

<그림 3-16>은 RFIDINFO 테이블의 내용을 엔터프라이즈 매니저 모드의 테이블 편집기를 통하여 확인한 결과이다. 본 연구를 위하여 준비한 RFID 태그는 총 수동형 태그 7개로 각 태그에 부여된 시리얼번호가 MODELNO이다. 나머지 필드의 내용은 제품을 가정하여 생성한 정보로 임의로 설정한 것들이다. 이들의 정보는 필요에 따라 변경할 수 있다.

MODELNO	COMMODITY	MAKER	SOURCE	PCODE	COLOR	SIZE	INDATE	PRICE
02815697	냉장고 SR4811...	LG 전자	한국	01-SR48	크림 화이트	140*82*95	01-6월 -200...	870000
02815698	디지털 캠코더 v...	삼성전자	필리핀	02-VMC5	실버	327*271*31.8	01-4월 -200...	1240000
02355960	노트북 SP30-5...	삼성전자	한국	02-SP30	화이트 실버	1230*1393*5...	08-8월 -200...	2475000
02355961	TV SVP-52Q7H...	LG 전자	중국	01-TV05	하와이이안	574*292*320	10-9월 -200...	3330000
02355962	오디오 MM-B9(...	인텔	말레이시아	03-MMB9	글로시 실버	876*870*675	15-5월 -200...	352000
02355963	김치냉장고 HN...	위니아	한국	04-HNR2	화이트 실버	521*514*440	30-4월 -200...	200000
02355964	살균 건조기 RE...	동양매직	일본	05-RE87	화이트 골드	594*600*1006	07-1월 -200...	222400

<그림 3-16> RFIDINFO 테이블의 내용

제 4 장 RFID 인식 단말기의 구현 및 실험

4.1 하드웨어

4.1.1 메인보드

메인보드 선정은 개발하려 하는 산업용 환경에 맞는 최적의 조건을 고려하여 선정하였다. 첫 번째로 TCP/IP 통신을 위하여 LAN카드가 있어야하며, RFID 리더기와 통신을 위해서 RS-232C 통신포트가 하나이상 있어야 한다. 두 번째로 운영체제를 임베디드 XP를 사용하기 위해서는 중앙처리장치가 300MHz이상 되어야 하며, RAM이 128MB이상 지원하는 보드이어야 한다. 세 번째로는 일반 사무실 이므로 주위 환경의 영향은 크게 받지 않고 유동 물자가 많으므로 시스템이 다운되지 않도록 안정성 있게 설계 되어야 된다. VIA사의 저발열, 저 전력소모의 C3 800 CPU를 하고 128MB의 메모리, 시리얼 통신을 위한 COM Port가 2개, 데이터베이스통신을 위한 100Mbps LAN을 내장한 <그림 4-1>의 타겟보드(WEB 6580)를 선정하였다.



CPU: VIA C3 800MHz
RAM: 128MB
COM Port: 2EA
LAN: 100M
ChipSet: VIA VT8606 +VT82686B
Size: 145mm x 102mm

<그림 4-1> WEB 6580 타겟보드와 스펙

4.1.2 파워 모듈 및 기타 장치

<그림 4-2>의 좌측은 일반 파워서플라이이며, 우측은 ST-150SL이다. 내부 파워 용량은 임베디드 보드, 하드디스크만 사용하므로 많은 용량의 파워를 사용하지 않아도 된다. 여기에서는 사이즈는 작고 용량은 적당한 세븐팀에서 제작한 ST-150SL ATX 파워 서플라이를 사용하였다.



<그림 4-2> 일반파워서플라이와 ST-150SL

<표 4-1>에서 두 파워서플라이를 비교해보면, ST-150SL의 최대 파워 용량은 150W를 지원하며, +3.3V, +5V, +12V, -5V, -12V의 고정 DC 파워를 출력한다. 여기서 사용하는 DC 파워는 +5V, +12V를 사용한다. 파워서플라이의 사이즈는 194mm * 82mm * 43mm 이며, 일반파워 서플라이보다는 적다. 따라서 높이가 낮은 단말기에 사용하기에 적합하여 ST-150SL 제품을 선택하였다.

<표 4-1> 일반 파워서플라이와 ST-150SL의 비교

모델	+3.3V	+5V	+12V	-5V	-12V	+5VSB	MAX.WATT
일반	20A	25A	13A	0.5A	1A	2A	250W
	150W		156W	2.5W	12W	10W	
ST-150SL	10A	15A	5A	0.2A	0.5A	1.5A	150W
	80W		60W	1W	6W	7.5W	

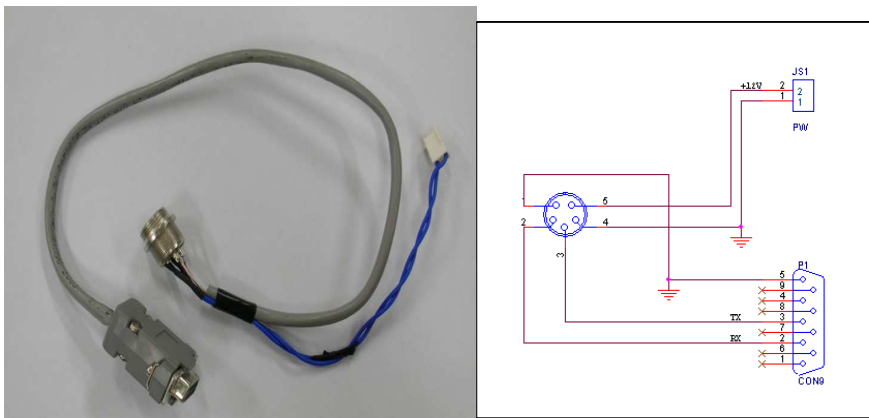
하드디스크는 <그림 4-3>의 일반 디스크 형태의 하드디스크와 CF 타입의 하드디스크 중 디스크 타입의 하드디스크를 사용하였다. 윈도우 임베디드 XP에서 만들어진 이미지는 대략 200MB 정도인데 데이터 용량 50MB 정도 포함하더라도 대략 300MB 정도의 공간만 있으면 된다. 그런데 여기서는 일반 타겟 시스템을 물류시스템에서 RFID 인식 단말기를 제작을 하였으므로, 가격대비 및 일반 보편성을 보더라도 일반 메모리 형태의 하드 보다는 저렴하고 보편화된 디스크 타입의 하드디스크를 선택하였다.

그러나 일반적으로 시스템의 안정화를 위해서는 CF 타입의 메모리를 사용하여야 한다. 왜냐하면 낮은 파워를 사용하여 저전력 임베디드 시스템을 만들 수 있으며, 디스크 타입의 하드 디스크를 사용하면 연결을 위해 사용하는 케이블 때문에 안정성을 떨어뜨리며 부피 또한 메모리 타입의 하드 보다는 커서 단말기 케이스 제작시 많은 불편한 점이 있다. 이 논문에서는 디스크 타입 형태의 일반 하드 디스크를 사용하여 제작하였다.



<그림 4-3> 하드디스크타입과 CF 타입의 메모리

RFID 리더기는 수동형 방식을 사용하였다. 연결케이블은 리더기에 +12V를 제공해야 하며, 메인 보드와 RS-232C 직렬 포트에 연결을 해야 한다. 따라서 RS-232C 통신 포트로서 사용하는 2번, 3번 핀에다가 RFID 리더기에서 출력되는 RX, TX를 연결 하였다. 그리고 파워서플라이에서 공급되는 +12V 전원은 5번 핀에 연결하였다. <그림 4-4>의 좌측은 RFID 리더기와의 연결 케이블을 완성한 그림이며, <그림 4-4>의 우측은 회로도를 나타내고 있다.



<그림 4-4> RFID 연결 케이블과 회로도

4.2 운영체제

4.2.1 윈도우 임베디드 XP 정의 및 특징

윈도우 임베디드 XP(XPE ; Windows Embedded XP)는 개발자들에게 친숙한 API를 갖춘, 안정적이고 기능이 풍부한 운영체제에 대한 필요성에 의해 개발됐다. XPE의 역사는 MS-DOS, 정확하게는 문자기반의 MS-DOS에서 그래픽 사용자 인터페이스가 요구되던 시점으로 거슬러 올라갈 수 있다. 그 당시 임베디드 개발자들은 x86에서 동작하는 운영체제 중 널리 알려진 MS-DOS를 선택했다. 왜냐하면 그래픽 인터페이스를 원하는 개발자는 윈도우즈 3.1도 추가할 수 있었으며, MS-DOS나 윈도우즈를 ROM에 탑재하고 싶은 개발자는 ROM 버전도 사용할 수 있었기 때문이다. 1995년에는 i80386과 같은 32비트 프로세서를 포함하는 임베디드 시장이 형성되기 시작했다. 그러나 몇 년 안에 펜티엄 프로세서가 뒤를 이어 바로 개발됨에 따라 486은 임베디드 시장에서 그대로 건너뛰게 됐다. 개발자들은 뛰어난 처리속도와 그래픽 성능의 향상으로 사람과 기계 사이에 강력한 인터페이스를 탑재한 임베디드 시스템을 개발할 수 있게 됐다. 윈도우즈 95와 NT도 몇몇 플랫폼에서 사용한 적이 있었지만 이미 RTOS와 임베디드 커널에 익숙해져 있는 개발자들을 만족시키기에는 임베디드 관련 기능이 너무 부족했다. 이 시기는 ROM에서 부팅되는 윈도우즈의 축소 버전이 필요한 시기였다[17]. 이때 출시된 것이 윈도우즈 CE이다. 윈도우즈 CE는 애초에 임베디드 시스템과 가전제품에 사용될 목적으로 개발된 운영체제였다. 윈도우즈 CE의 특징으로는 사용자에게 익숙한 Win32 API를 들 수 있는데, 이 Win32 API는 배터리를 사용하고 화면이 작은 모바일 시스템에서도 사용할 수 있도록 개량된 것이었다. 사용자는 이런 개량 외에도 그 당시의 데스크 탑 운영체제인 윈도우즈 95와 윈도

우즈 NT에서나 볼 수 있던 API/MFC 지원 기능을 요구했다. 이런 윈도우 CE에 대한 압도적인 반응, 임베디드 시장에서의 PC 유형 시스템의 지속적인 성장, 그리고 향후 인터넷 기기의 기능성으로 인해 마이크로소프트사는 임베디드 시스템에 주목하게 된다. 윈도우 NT는 통신 시스템, 보안 시스템, 공장 자동화, 콜 센터, 멀티미디어 서버, 네트워크 프린터, 데이터베이스 서버를 포함하는 몇몇 임베디드 어플리케이션에서 널리 사용됐다. 써드 파티(third parties) 쪽에서는 headless operation, 더 적은 메모리, 실시간 처리, 그리고 solid-state 미디어(플래시와 CD-ROM) 지원과 같이 임베디드 어플리케이션에서 필요로 하는 기능을 제공했다. 마이크로소프트사는 이런 기능을 모두 모은 윈도우 NT 임베디드 4.0을 출시하기 위한 몇몇 기술을 개발(혹은 구입)했다. 윈도우 NT 임베디드 4.0은 윈도우 ROM 버전 3.1 이후로는 처음으로 데스크 탑 OS에 기반을 둔 임베디드 제품이었다. 윈도우 NT 임베디드(NTE)는 큰 성공을 거두었다. NTE에서 제공한 보안 및 통신 기술은 임베디드 디바이스의 새 세대를 열었다. NTE 툴을 사용하면 별도로 시스템을 개발할 필요조차 없었다. 시스템 개발에 필요한 것은 적절한 하드웨어를 갖춘 PC와 윈도우 NT, 그리고 어플리케이션을 개발할 비주얼 스튜디오 툴이었다. 어플리케이션을 개발한 다음 NTE 툴을 사용하기만 하면 최종 임베디드 시스템을 구축할 수 있다. XPE는 데스크 탑용, 윈도우 XP 프로페셔널과 동일한 운영체제다. 표준 Win32 API를 지원하며, 윈도우 XP 프로페셔널의 디바이스 드라이버, 어플리케이션, 특성 및 기능은 XPE에서도 모두 찾아 볼 수 있다. XPE는 임베디드 플랫폼에서 사용되는 12,000개 이상의 다양한 컴포넌트를 제공한다. 또한 USB 2.0과 무선 인터페이스를 지원한다. 윈도우 XP SP1 이나 윈도우 2000 Professional와 매우 유사하고, 운영체제 이미지 제작은 윈도우 XPESP1를 사용한다. 이때 XPE용 QFE 업데이트는 설치되어 있어야 한다[18][19][20].

개발시스템의 하드웨어 사양은 펜티엄III500MHz 이상, RAM 256MB 이상이 필요하다. 저장용량은 3GB정도가 필요하다. 타겟 하드웨어의 최소 필요사항은 펜티엄 클래스200MHz이상과 RAM 64MB, 500MB 이상 HDD이다.

4.2.2 임베디드 XP 개발 개요

데이터베이스는 XPE의 심장과도 같다. 수많은 NTE 사용자와 개발자는 데이터베이스 접속을 확장해서 팀 단위로 컴포넌트를 공유할 수 있는 기능이 추가되기를 원했다. 그래서 XPE는 로컬 또는 원격지에서 데이터베이스를 접속할 수 있는 데이터베이스 구조를 필요로 했는데, 이때 가장 중요한 점이 데이터베이스 속도였다. SQL은 원격 데이터베이스 측면에서 보면 가장 손쉬운 선택이었다. 물론, SQL은 로컬 접속도 가능하지만 데이터베이스의 모든 기능이 필요하지 않은 개발자를 위해서는 좀 더 가벼운 데이터베이스 엔진을 제공해야 했다. XPE에서는 이런 개발자를 위해 마이크로소프트에디터 엔진을 제공한다.

컴포넌트 오브젝트 모델(COM; Component Object Model)은 새로운 윈도우 어플리케이션 개발에 있어서 표준으로 자리 잡고 있으며, 이것을 XPE에서도 예외는 아니다. XPE 데이터베이스는 오브젝트로 딱 차있는데, 데이터베이스 내에 있는 모든(파라미터를 포함하는) 컴포넌트는 오브젝트로 취급된다.

컴포넌트 매니지먼트 인터페이스(CMI; Component Management Interface)라는 추상 레이어는 중요한 개발 툴인 타겟 디자이너와 컴포넌트 디자이너가 데이터베이스와 로컬 또는 원격지에서 상호작용할 수 있게 해 준다. 컴포넌트 매니지먼트 인터페이스는 여러 사용자

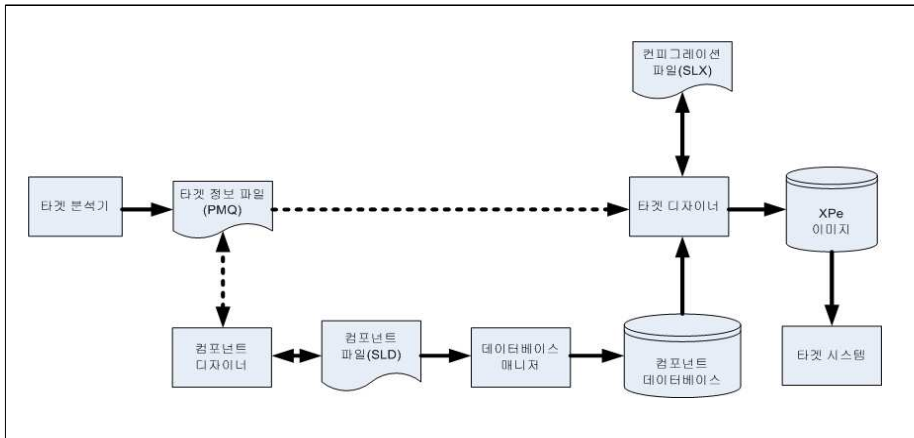
가 SQL과의 원격 접속을 위해 꼭 필요한 비동기 데이터베이스 접속 기능을 제공한다. 데이터베이스는 여러 개발자가 접속 할 수 있으므로 읽기 전용 모드와 배타적 접속모드, 이 두 가지 모드로 작동한다. 예를 들어 컴포넌트 데이터베이스 매니저(CDM; Component Database Manager)가 데이터베이스를 배타적 모드로 접속할 경우를 생각해 보자. CDM은 데이터베이스에 뭔가 변화를 줘야 하므로 배타적 접속 모드를 필요로 한다.

CDM의 배타적 접속 모드는 컴포넌트를 삭제해야 하는데 데이터베이스 매니저에게도 중요한 기능이다. 만약 타겟 디자이너가 CDM이 작업을 시작한 다음에 시작된다면, 타겟 디자이너는 개발자가 선택한 데이터베이스에 접근 할 수 없다는 메시지를 화면에 보여 줄 것이다. 배타적 모드는 여러 사용자가 이미 삭제 했을지도 모르는 정보를 상호 교환하는 오류를 방지할 수 있다.

XPE 틀은 최종 타겟 시스템을 구현할 때 사용된다. XPE 이미지 구축의 전처리와 후처리를 돕는 몇몇 툴과 함께 제공된다. 타겟 디자이너, 컴포넌트 디자이너, 그리고 컴포넌트 데이터베이스 매니저는 개발 프로세스의 서로 다른 부분에서 XPE 데이터베이스와 연동한다. XPE 이미지를 구축하려면 분석(타겟 분석기), 컴포넌트 생성(컴포넌트 디자이너), 컴포넌트의 데이터베이스 추가(컴포넌트 데이터베이스 매니저), 컨피그레이션 구축(타겟 디자이너), 배포의 5단계를 거쳐야 한다.

<그림 4-5>에서도 볼 수 있듯이 초기 분석이 끝나면 그 다음 단계는 두 가지 경로로 갈라진다. 타겟 분석기에서 H/W 정보를 가져오면, 타겟 디자이너로 그 정보를 직접 불러 오는 것이 가능하다. 이때 개발자는 필요한 컴포넌트를 컨피그레이션에 추가하고, XPE 이미지를 생성할 수 있다. 이 논문에서는 PMQ 파일을 가져 와서 컴포넌트 디자이너를 사용, 매크로/플랫폼 컴포넌트를 생성하였다. 여기서

XPE는 PMQ 파일을 생성하는 실행파일, TAP.EXE 을 제공한다.. 매크로 컴포넌트는 여러 컴포넌트를 하나의 컴포넌트에서 관리할 수 있게 해준다. 나중에 설명하겠지만, 매크로 컴포넌트는 메모리 용량을 줄여주기도 한다. 타겟 디자이너는 컴포넌트를 컨피그레이션에 추가하고 이미지를 생성한다. <그림 4-6>는 PMQ 파일내부모습이다.



<그림 4-5> XPE 개발 흐름도

```

devices.pmq - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<HARDWARE IDS>
<DEVICE ID Order="1">ACPIPNP0803</DEVICE ID>
<DEVICE ID Order="2">PNP0803</DEVICE ID>
</HARDWARE IDS>
<COMPATIBLE IDS>
</COMPATIBLE IDS>
</DEVICE>
<DEVICE ConfigFlags="0">
<DEVICEDESC>프린터 포트</DEVICEDESC>
<FRIENDLYNAME>프린터 포트(LPT1)</FRIENDLYNAME>
<HARDWARE IDS>
<DEVICE ID Order="1">ACPIPNP0800</DEVICE ID>
<DEVICE ID Order="2">PNP0800</DEVICE ID>
</HARDWARE IDS>
<COMPATIBLE IDS>
</COMPATIBLE IDS>
</DEVICE>
<DEVICE ConfigFlags="0">
<DEVICEDESC>통신 포트</DEVICEDESC>
<FRIENDLYNAME>통신 포트(COM1)</FRIENDLYNAME>
<HARDWARE IDS>
<DEVICE ID Order="1">ACPIPNP0501</DEVICE ID>
<DEVICE ID Order="2">PNP0501</DEVICE ID>
</HARDWARE IDS>
<COMPATIBLE IDS>
</COMPATIBLE IDS>
</DEVICE>
<DEVICE ConfigFlags="0">
<DEVICEDESC>시스템 스피커</DEVICEDESC>
<HARDWARE IDS>
<DEVICE ID Order="1">ACPIPNP0800</DEVICE ID>
<DEVICE ID Order="2">PNP0800</DEVICE ID>
</HARDWARE IDS>
<COMPATIBLE IDS>
</COMPATIBLE IDS>
</DEVICE>
<DEVICE ConfigFlags="0">
<DEVICEDESC>PCI 버스</DEVICEDESC>
<HARDWARE IDS>

```

<그림 4-6> devices.pmq

XPE 개발 프로세스는 하드웨어 분석, 커스텀 컴포넌트 생성 그리고

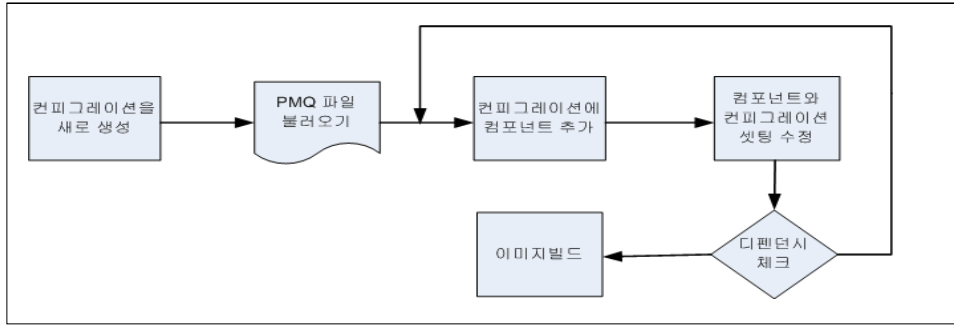
XPE 이미지 구축을 돕는 툴의 결합체로 볼 수 있다. XPE를 설치하고 타겟 하드웨어를 선택했다면, 그 다음에는 플랫폼에 대한 정보를 수집해야 한다.

컨피그레이션은 플랫폼의 기능을 지원하는 여러 개의 컴포넌트로 구성된다. 개발자는 I/O, 칩셋, 칩 리비전, 프로세서 개수, 바이오수 기능 및 용량, 그리고 브리지 칩을 모두 컨피그레이션 파일 내에 기술해서 이미지가 제대로 동작하게 해야 한다. 타겟 분석기 툴이 없으면 개발자는 보드에 있는 모든 IC칩과 장착된 카드에 대해 알아야 하고, 각 IC 칩 내용과 연관된 적절한 컴포넌트를 데이터베이스 내에서 선택해야 한다. 게다가 커스텀 PC 플랫폼을 구축하려면 이런 모든 정보를 개발자가 수집해야 한다.

4.2.3 타겟 디자이너

4.2.3.1 타겟 디자이너의 구성

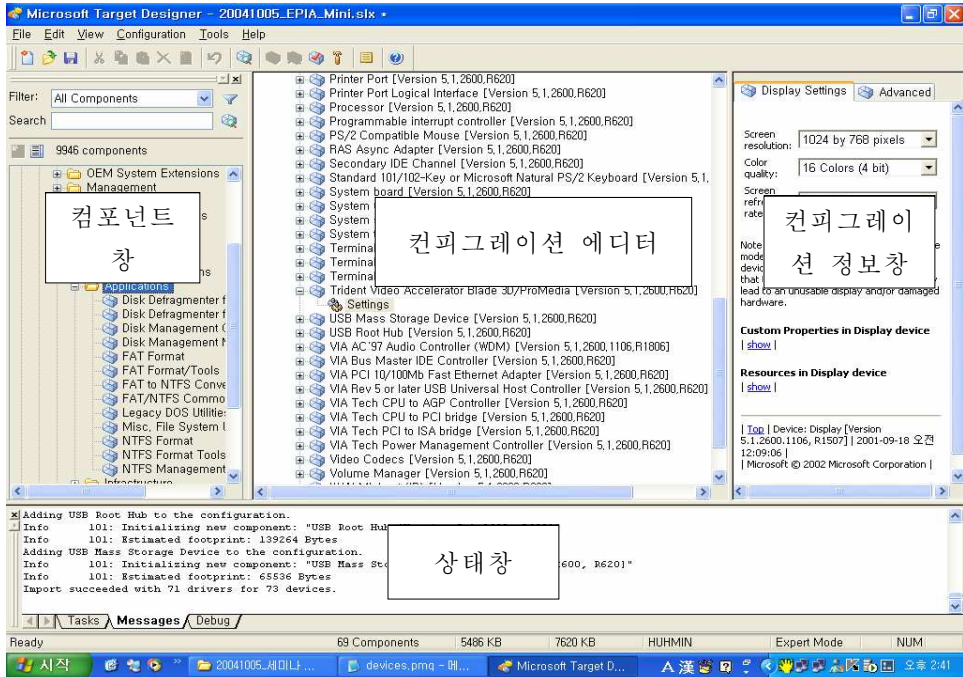
타겟 디자이너는 윈도우즈 임베디드 XP 에서 사용하는 운영체제 컨피그레이션 제작용 툴이다. 따라서 타겟 시스템을 구성하는 운영체제 기능과 특성을 정의 할 수 있게 해 준다. <그림 4-5>의 흐름도에서 컨피그레이션이 정의되면, PMQ파일의 내용으로 타겟 시스템의 하드웨어정보를 받아와 타겟 디자이너는 부팅이 가능한 XPE 이미지를 생성할 수 있다. 커널 소스에서부터 코드를 컴파일 하는 다른 RTOS와는 달리, 타겟 디자이너는 바이너리 파일에서 컴포넌트를 선택하고 XPE 이미지를 생성한다. 따라서 타겟 디자이너를 사용하면 다양한 디바이스에서 XPE 기반 플랫폼을 구축 할 수 있다.



<그림 4-7> 타겟 디자이너의 개발 흐름도

<그림 4-8>은 타겟디자이너를 실행시킨 모습이다.

- ▶ 컴포넌트 창
데이터베이스에 있는 컴포넌트 목록을 보여 준다.
- ▶ 컨피그레이션 에디터
컨피그레이션의 특징 컴포넌트 인스턴스를 나열하고 사용자 - 컨피그레이션 셋팅으로의 접속을 제공한다.
- ▶ 컨피그레이션 정보창
각 컨피그레이션의 속성을 보여준다.
- ▶ 출력창
테스크, 빌드 및 Import 정보를 보여주고 디버그 빌드에 사용되는 OS 디버그 결과를 보여준다.



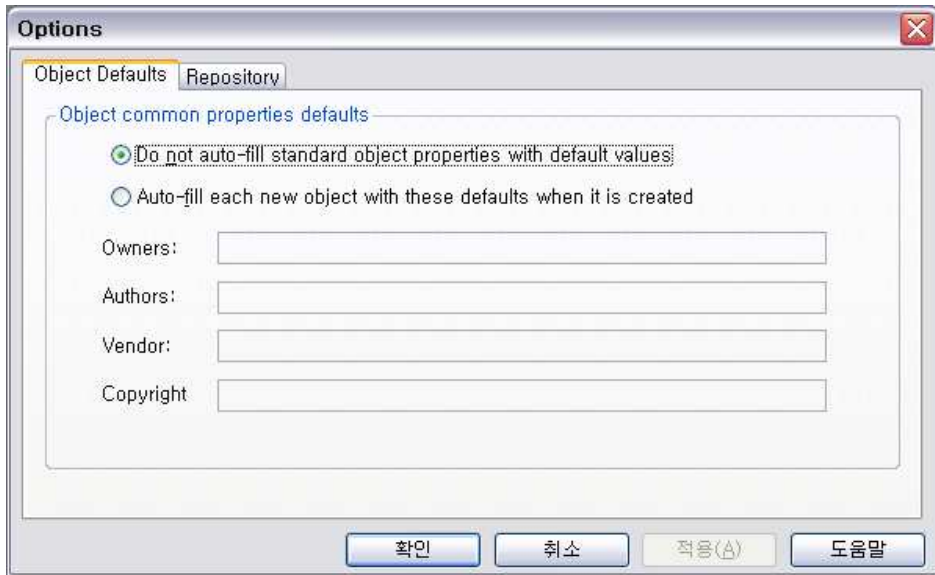
<그림 4-8> 타겟 디자이너

4.2.4 컴포넌트 디자이너

컴포넌트들은 윈도우즈 임베디드 XP에서 기본적으로 제공하는 것만 사용하였다. 그러나 XPE의 실제 장점은 <그림 4-9>의 흐름도에서 볼 수 있듯이 데이터베이스에 포함돼 있지 않은 컴포넌트를 생성하는데 있다. 이런 컴포넌트는 시스템 레벨 정의 파일, 일명 슬라이더(SLD) 파일에 저장된다. 컴포넌트 디자이너는 이런 컴포넌트를 생성하는데 사용하고 컴포넌트 데이터베이스 매니저는 데이터베이스에 새 SLD 컴포넌트를 입력을 하는데 사용한다. <그림 4-10>은 컴포넌트 디자이너를 실행시킨 화면이다.

디자이너에서는 여러 개의 SLD 파일을 동시에 열 수 있으므로, 한 SLD 파일에 있는 정보를 다른 SLD 파일로 쉽게 복사할 수 있다.

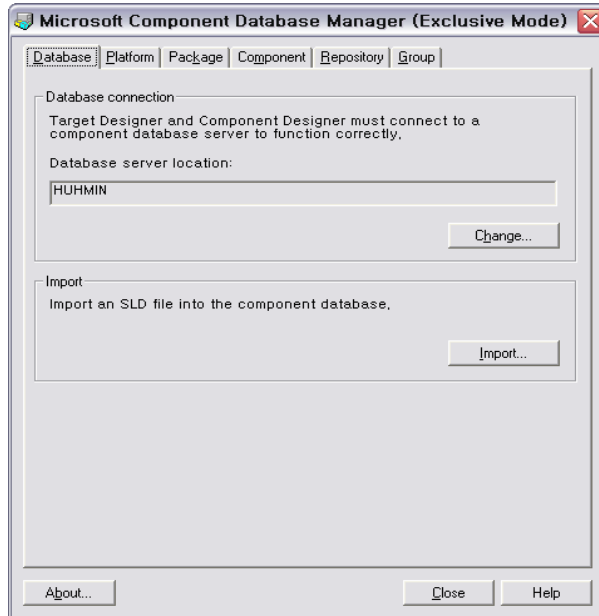
<그림 4-11>은 컴포넌트 디자이너의 옵션으로 오브젝트의 설정값을 보여준다.



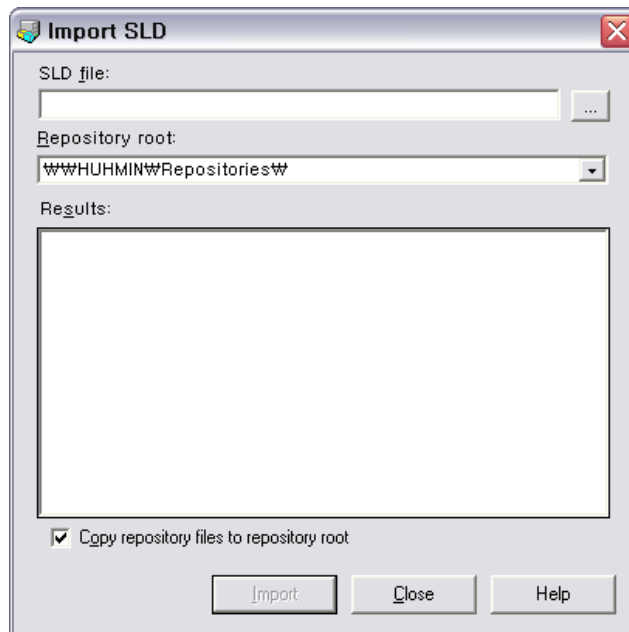
<그림 4-11> 컴포넌트 디자이너의 옵션

4.2.5 컴포넌트 데이터베이스 매니저

<그림 4-12>의 컴포넌트 데이터베이스 매니저(CDM ; Component Database Manager)는 컴포넌트 데이터베이스와 리포지토리 관리 기능을 제공하는데, 이 기능은 컴포넌트 디자이너와 타겟 디자이너 둘에서 사용할 수 있다. CDM은 팀 단위로 공유하는 환경에서 프로젝트를 진행할 때 유용한데, 데이터베이스 관리자가 데이터베이스를 관리할 수 있게 해주기 때문이다. 이 컴포넌트 데이터베이스는 개발 시스템이나 서버에 설치할 수 있고, 다양한 플랫폼을 포함시킬 수 있다



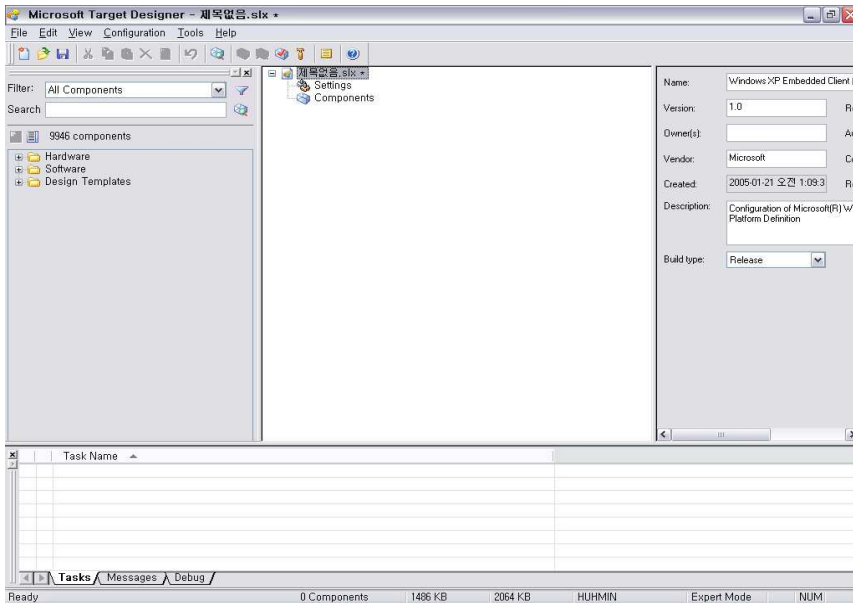
<그림 4-12> 컴포넌트 데이터베이스 매니저



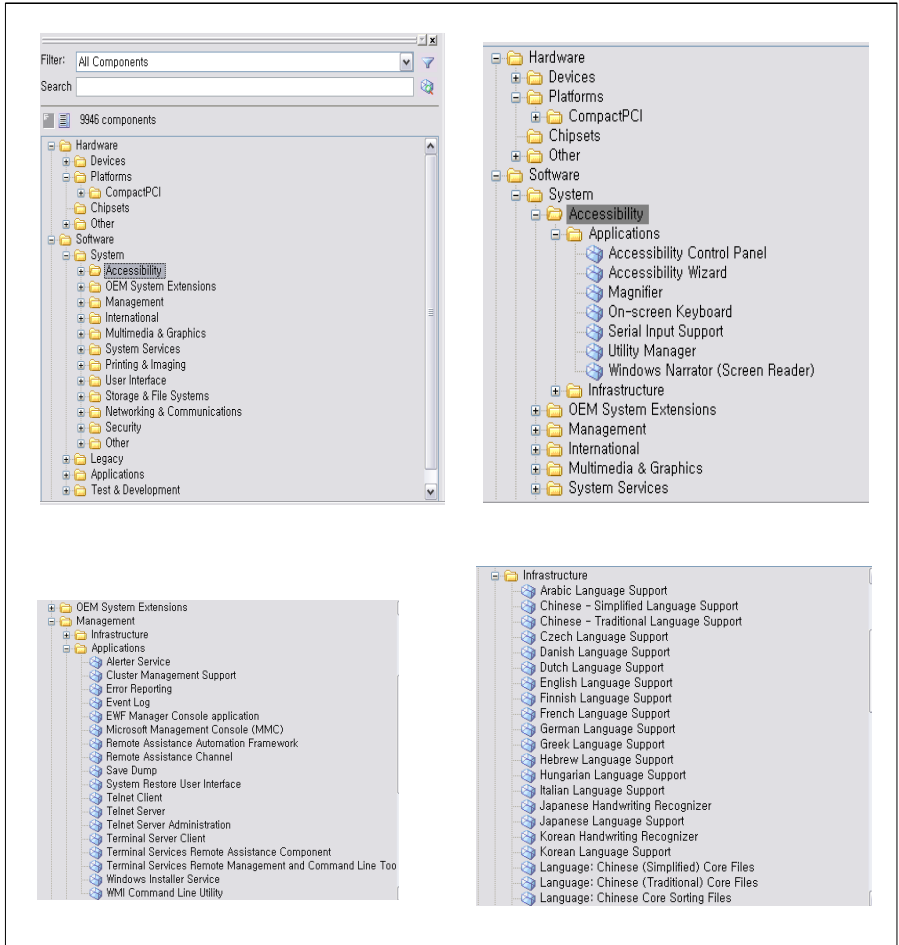
<그림 4-13> Import SLD

<그림 4-13>은 컴포넌트 디자이너에서 새롭게 생성된 컴포넌트가 포함된 SLD파일을 불러오는 모습이다. 불러온 SLD파일은 컴포넌트 데이터베이스 매니저가 관리하게 된다.

컴포넌트 데이터베이스는 의미 그대로 윈도우즈 XP의 요소들을 각각으로 따로 구성하여 컴포넌트화 하였다. 먼저 타겟 디자이너를 실행하고, “File->New”를 선택하며, <그림 4-14>과 같이 실행이 된다. 그림의 좌측에 보이는 것이 컴포넌트 데이터베이스와 연동하여 보여주는 윈도우즈 임베디드 XP 컴포넌트들이다. <그림 4-15>와 <그림 4-16>은 컴포넌트 데이터베이스 매니저가 각각 하드웨어 컴포넌트와 소프트웨어 컴포넌트 창을 열어 관리하는 모습이다.



<그림 4-14> 타겟 디자이너



<그림 4-16> 컴포넌트 데이터베이스 소프트웨어 부분

4.2.6 컴포넌트 데이터 베이스

컴포넌트들은 크게 하드웨어, 소프트웨어, 디자인 템플릿으로 구성 되어있다. 하드웨어는 장치, 플랫폼, 칩셋, 다른 장치의 하위폴더로 다시 나누어진다. 각 폴더에는 윈도우즈 임베디드 XP를 배포하기 전 까지 모든 하드웨어 드라이버들을 부분별로 나누어 정리해 놓았다. 소프트웨어는 윈도우즈 XP의 각 요소들을 부분별로 시스템, 레거시, 어플리케이션, 테스트 및 개발로 나누어 놓았다. 각 하위 폴더에는 더 자세한 요소들로 되어 있는 것을 확인 할 수 있다.

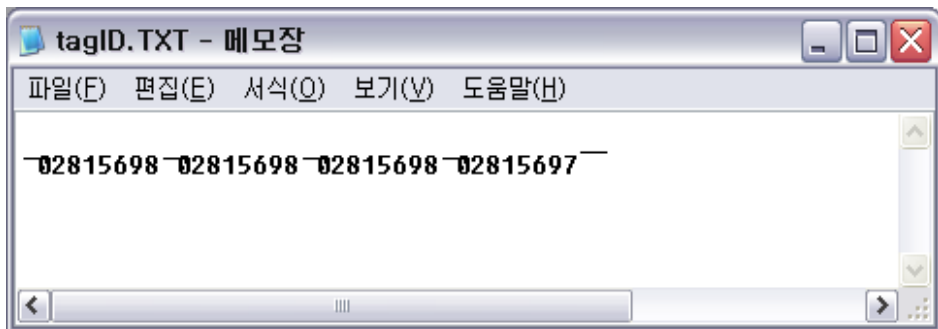
4.3 소프트웨어

4.3.1 RFID 인식 단말기 소프트웨어의 필요 기능

RFID 인식 단말기의 기능면에서 보면 크게 몇 가지로 분류 할 수 있다. 첫 번째로 RFID 리더기와의 RS-232C 직렬통신 부분과 두 번째 RFID 태그 정보를 처리하는 부분이다. 세 번째로는 메인 서버와 TCP/IP 통신하는 부분과 네 번째로 단말기에 맞는 화면 구성을 해야 한다.

4.3.1.1 RFID 태그 정보를 처리하는 부분

RFID 리더기에서는 RFID 태그에서 읽어 들인 값을 RS-232C 통신으로 숫자 값으로 출력한다. 이때 받아온 태그 ID의 값은 <그림 4-17>의 숫자 데이터이다. <그림 4-17>은 PC에서 하이퍼터미널로 RFID 리더기를 통해서 입력받은 값을 텍스트로 캡처한 모습이다. RFID 태그 마다 고유의 숫자 값을 가지며, 숫자 앞뒤로 특수 문자로 시작과 끝을 나타내고 있다.



<그림 4-17> TagID

출력된 데이터는 PC에서 Com Port를 통해서 데이터를 입력 받는다. 따라서 PC에서는 이 부분을 처리하는 루틴이 들어가야 되는데, 문제는 데이터 값이 정확하게 들어왔는가를 판별해야 한다. 그러므로 데이터가 이상 없이 전송되었다는 부분과 만약 잘못 읽어 들였을 때 처리하는 부분도 포함되어야 한다.

4.3.1.2 RFID 리더기와의 RS-232C 직렬통신 하는 부분

RFID 리더기와 PC와의 통신을 위해서 시리얼 통신 부분이 들어가야 된다. 시리얼 통신 부분은 PC를 효율적으로 사용하기 위해서 스레드를 사용해야 하며, Com Port를 항상 체크하고 있다가 데이터가 들어오면 바로 처리해야 하겠다.

4.3.1.3 메인 서버와 TCP/IP 통신하는 부분

메인 서버의 데이터베이스와 통신을 하기 위해서는 TCP/IP 소켓 통신을 하는 부분이 들어가야 된다. 서버는 오라클 데이터베이스를 사용 하였다.

4.3.1.4 단말기에 맞는 화면 구성

단말기의 화면 구성은 사용자가 쉽게 사용할 수 있어야 하며, 화면의 구성도 보기 쉽게 구성되어야 한다. 그리고 정확한 정보전달을 위해서 에러 메시지, 통신상태 등 실시간으로 바로 알려주는 부분이 필요 하겠다. RFID 태그 ID값에 따라 상품명, 태그 종류, 입고일, 출고일, 제품설명, 담당자, 전화번호 등을 보여 주도록 해야 한다.

4.3.2 RFID 인식 단말 소프트웨어 구현

4.3.2.1 RFID 리더기와 RS-232C 데이터 통신

RFID 리더기에서 들어오는 데이터를 검출하기 위해서는 컴퓨터 통신 모듈이 필요하다. 그리고 PC에 부담을 줄이기 위해서 스레드를 사용해야 한다. 소프트웨어 구현은 메인 프로그램과 분리하여 동적 라이브러리 링크(DLL ; Dynamic Link Library)로 하였다.

엠콤(MCom ; Mini Comport)는 Visual C++ 사용자가 RS-232C (NULL MODEM) 통신을 간단하고 편하게 구현할 수 있도록 제작하였다. DLL 라이브러리 형태로 제작하여 RS-232C 통신을 쉽게 제어할 수 있다.

- 엠콤 DLL 파일

- ① MCom.h : MCom class Header file
- ② MCom.lib : lib file for release
- ③ MCom.dll : dll file for release

- 엠콤 DLL 사용방법

- ① MCom Class 추가

통신기능을 구현하려는 객체에 MCom를 멤버변수로 아래와 같이 추가한다. 단, MCom가 추가되는 객체는 MFC의 CWnd 나 CWnd에서 상속받은 객체만 가능하다.(CWnd, CView, CMainFrame, CDialog)

```

#include "MCom.h"                // head file 선언
class CMyWnd : public CWnd
{
    ...
    MCom m_port;                // 멤버변수로 추가
    ...
}

```

<그림 4-18> MCom Class 추가

② CMyCom의 메서드 사용

- ▶ m_port.openPort(1, this) : Serial Port를 연다.
- ▶ m_port.closePort() : Serial Port를 닫는다.

③ 아래의 소스는 DLL 내부에서 핵심 요소인 소스이다.

```

먼저 Serial Port에서 데이터를 읽어옵니다.
dwRead = pComm->ReadComm( buff, 2048);

읽은 데이터를 한 Byte씩 Queue에 집어 넣습니다.
if (BUFF_SIZE - pComm->m_QueueRead.GetSize() > (int)dwRead)
{
    for ( WORD i = 0; i < dwRead; i++)
        pComm->m_QueueRead.PutByte(buff[i]);
}
else AfxMessageBox("m_QueueRead FULL!");

```

<그림 4-19> 쓰레드에서 하는일

현재 Queue에 들어 있는 데이터의 크기를 얻어옵니다.

```
int size= (m_ComuPort.m_QueueRead).GetSize();
```

Queue가 빌때까지 데이터를 읽어옵니다.

```
for( int i=0; i< size; i++ )
```

```
{
```

```
    (m_ComuPort.m_QueueRead).GetByte(&aByte);
```

```
    if( aByte!= NULL ) buff[i]= aByte;
```

```
    else { i--; size--; }
```

```
}
```

<그림 4-20> 부모 프로그램에서 하는 일

```
unsigned char* szReceive;
```

```
int nByteSize;
```

```
if( nPort == 1 )
```

```
{
```

```
    m_portCOM1.receiveByte( bt );
```

```
    nByteSize = m_portCOM1.getReceivedByteSize();
```

```
    szReceive = new unsigned char[nByteSize+1];
```

```
    m_portCOM1.receiveBytes(szReceive, nByteSize);
```

```
    szReceive[nByteSize] = NULL;
```

```
    strReceive = szReceive;
```

```
    delete [] szReceive;
```

```
}
```

```
else if( nPort == 2 )
```

```
{
```

```
    nByteSize = m_portCOM2.getReceivedByteSize();
```

```
    szReceive = new unsigned char[nByteSize+1];
```

```
    m_portCOM2.receiveBytes(szReceive, nByteSize);
```

```
    szReceive[nByteSize] = NULL;
```

```
    strReceive = szReceive;
```

```
    delete [] szReceive;
```

```
}else
```

```
    TRACE("COM port number error.");
```

<그림 4-21> Com port

4.3.2.2 메인 프로그램

메인 프로그램은 마이크로소프트사에서 제공하는 비주얼 C++ 6.0 엠에프씨(MFC; Microsoft Foundation Class Library) 를 사용하여 개발하였다. 세부 모듈별(A, B, C)로 개발하였으며, <그림 4-22>의 RFID 인식 단말기 메인화면을 구성하였다.

① A: RFID TagID 정보를 디스플레이 한다.

제품(상품명, TagID, 태그종류), 날짜(입고일, 출고일, 최종목적지등), 창고(창고번호, 창고위치), 담당자(창고 담당자, 전화번호 등), 제품설명(제고수량, 제조국가, 제조날짜 등)

② B: COM 포트 상태를 디스플레이 한다.

COM1, COM2 번 어디서든 데이터가 들어오던지 처리가능하다.

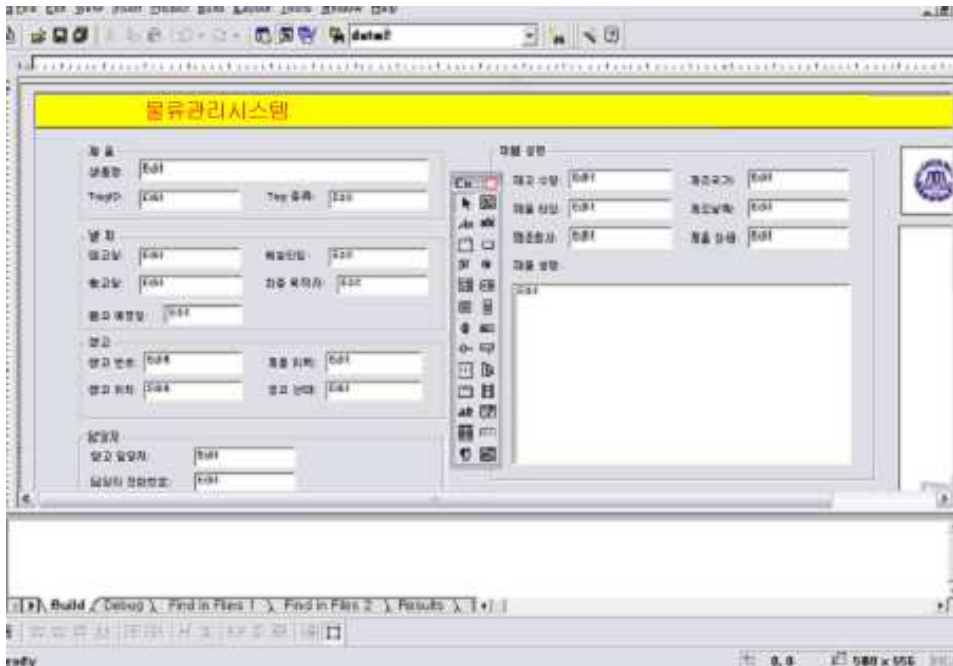
③ C: 데이터베이스와 TCP/IP 통신을 담당한다.

서버통신상태를 나타내며, 데이터베이스에 있는 내용을 추가, 수정, 삭제를 할 수 있으며, 메인화면에 보이는 내용을 인쇄 할 수 있다.

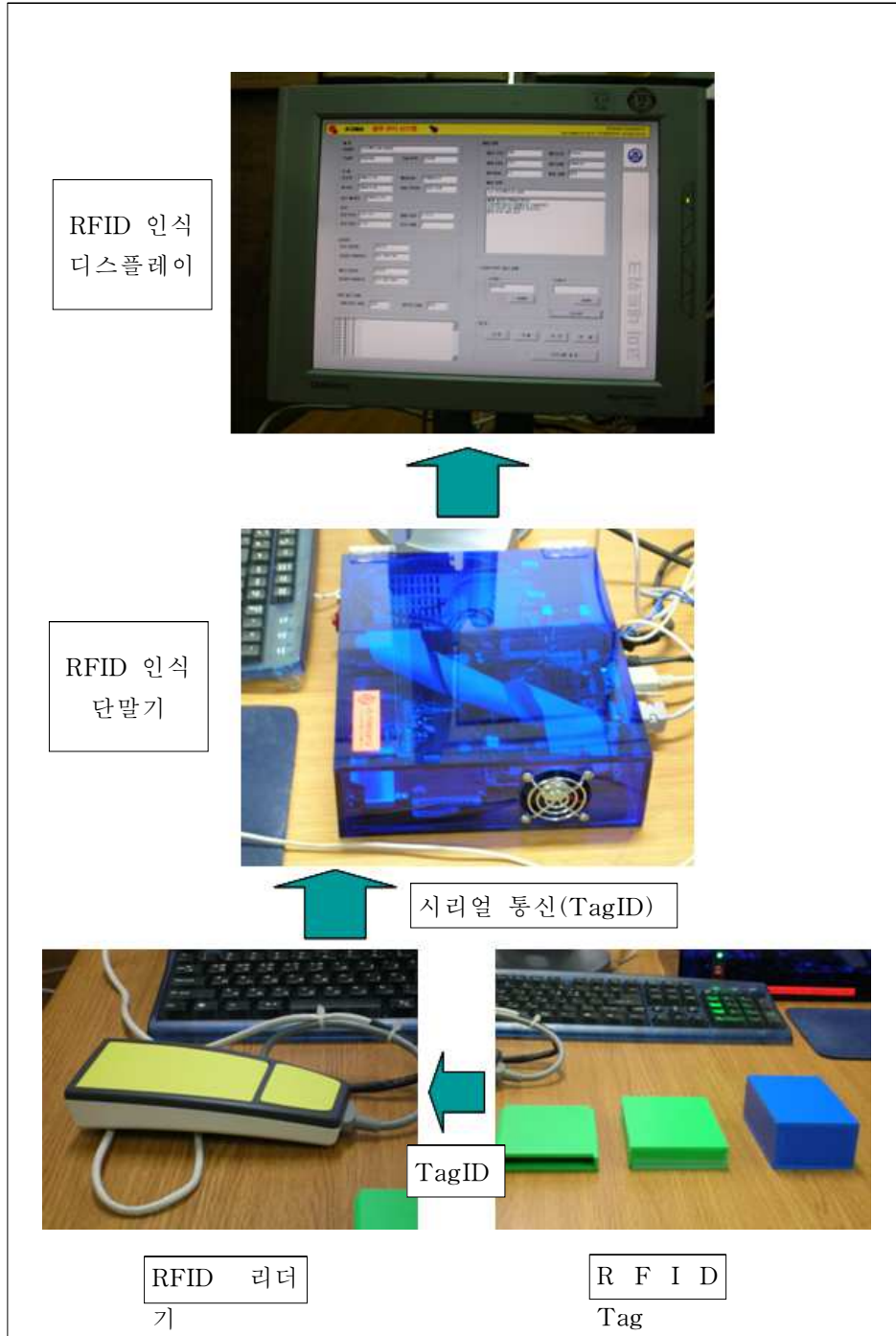
④ 메인 화면 디자인

단말기의 화면은 <그림 4-22>이다. 사용자가 쉽게 사용할 수 있어야 하며, 화면의 구성도 보기 쉽게 구성하였다. 그리고 정확한 정보 전달을 위해서 에러메시지, 통신상태 등 실시간으로 바로 알려주는 부분을 추가하였다. RFID Tag ID값에 따라 상품명, 태그 종류, 입고일, 출고일, 제품설명, 담당자, 전화번호 등을 표시한다. <그림 4-23>의 최종 디스플레이 및 데이터 흐름을 보면, RFID 태그들을 사각상자에 만들어서 안에 붙였다. 각 RFID 태그들은 고유의 숫자 값을 가지고 있는데 이것을 일정한 주파수 대역을 사용한다. RFID 리더기는 RFID 태

그들의 고유한 값을 읽어오는데 사용하며, 이 값들은 PC에서 볼 수 있도록 RS-232C 통신으로 출력 값을 보내게 된다. RFID 인식 단말기는 넘어온 Tag ID 숫자 값으로 메인 데이터베이스에 접속하여 Tag ID값으로 정의된 테이블에 있는 값을 <그림 4-23>의 위쪽에 보는 것과 동일하게 디스플레이 된다.



<그림 4-22> 메인 프로그램디자인



<그림 4-23> 최종 디스플레이 및 데이터 흐름



<그림 4-24> RFID 인식 단말기 완성품 전면사진

제 5 장 결 론

본 연구는 윈도우 임베디드 XP 기반의 RFID 인식 단말기의 개발에서 임베디드 시스템과 RFID를 연동하는 기술은 하드웨어의 설계에서부터 운영체제의 포팅 그리고 응용 애플리케이션의 개발까지 이루어졌으며, 완전한 시스템으로써의 기능을 수행하는데 신뢰성을 보장하고 시스템의 맞춤형 기능으로 최적화할 수 있도록 응용 소프트웨어의 개발까지 진행하였다.

RFID 인식 단말기 시스템은 태그 정보 인식을 위한 태그 인식부와 정보처리를 담당하는 메인 하드웨어부분, 그리고 입력된 정보를 처리, 가공하고 데이터를 저장하는 데이터베이스 부분으로 통합 시스템의 구조를 갖고 있다. 이렇게 구현된 임베디드 환경에서의 응용 소프트웨어를 통하여 처리하고자 하는 정보의 인식과 검색 및 저장이 일반 PC 수준과 동일하게 단말기 상에서 동작됨을 확인할 수 있었다. 또한, 시스템의 개발 비용과 하드웨어 구축 비용이 기존의 유사 시스템에 비하여 저비용으로 설계가 가능하였으며, 기존의 시스템을 대체하여 사용된다면 구축된 데이터베이스의 연동과 관련한 응용 애플리케이션 부분만의 재수정으로 충분히 고려될 수 있다는 장점이 있다. 또한 사용된 운영체제가 안정적이고 기능이 풍부할 뿐만 아니라, 개발자들이 익숙한 API를 갖추고 있는 윈도우 임베디드 XP 환경으므로 개발자 및 엔지니어들에게 도움이 될 것으로 사료된다.

향후 연구 계획으로는 RFID 인식 단말기에 GPS 수신기의 모듈을 장착하여 휴대하기 편리한 구조의 서버와 클라이언트로 구성된 시스템을 개발하고자 한다.

참 고 문 헌

- [1] 정보통신부, “제 2005년도 연두사업보고”, 정보통신부 사업보고 계획서. pp.3-5, .2005.
- [2] 유비쿼터스 컴퓨팅, <http://justit.gigaro.net>
- [3] 김상태, “RFID 기술개요 및 국내외 동향 분석”, 정보 통신 연구진 후원, pp.2-3, 2003
- [4] 김대기, “차세대 성장 산업 RFID 기술 동향 및 시장 현황-해외 시장 현황 및 표준화 동향”, 정보산업 3-4호 pp.32-35 2004.
- [5] 김병찬, “물류 응용을 위한 PDA 기반의 RFID 인식 시스템의 설계 및 구현”, 석사학위논문, 한국해양대학교, pp20-22, 2004.
- [6] RFID system, <http://www.rfid-handbook.com/rfid/index.html>
- [7] 신산업 경영원, “유비쿼터스의 물결 RFID 준비”, 한국 과학 기술 정보 연구원 18권 3호 pp.58-61, 2004.
- [8] 오세진, “RFID 태그를 이용한 감각형 메일 리더”, 한국정보과학회 03 가을학술발표 논문집 2003권 pp.418-420, 2003.
- [9] 김완석, “RFID 표준화 동향”, 한국 전자 통신 연구소 정보통신, 기술 정책 및 산업 주간 기술 동향 pp.1-13, 2003.
- [10] 이근호, 박경훈, “RFID 시스템 표준화 동향”, KT 신사업기획본부 표준화 원고, pp2-5, 2004
- [11] 이은곤, “RFID확산 추진 현황 및 전망”, 정보 통신 정책 제16권 6호 통권 344호 pp.34-39, 2003.
- [12] 조지운, “RFID를 이용한 자동위치 확인 시스템에 관한 연구”, 한국 과학 기술 정보 연구원 17권 4호 pp.31-35 2002.

- [13] 정민화, “RFID 기술의 국제표준화 동향”, 전파진흥 14권 2호 pp.61-73 2004.
- [14] 이근호, “무선 인식 기술”, TTA 저널 제89호 pp.1-7. 2002.
- [15] 권영빈, “RFID 국제 표준화 동향”, 2004년 RFID 국제심포지엄 2004.
- [16] 표철식, “RFID 핵심기술동향”, 정보산업, 3-4호 pp.23-27, 2004.
- [17] Microsoft, “Fast System Startup for PCs Running Windows XP”, January 31, 2002
- [18] Berard, Stephen, “Using the enhanced Writed Filter (EWF) in Windows XP Embedded”, MSDN.Microsoft.com, January 2003
- [19] Bott, Ed, Carl Siechert, and Craig Stinson, “Microsoft Windows XP Inside Out”, Microsoft Press, 2001
- [20] Fincher, Jon, “Windows XP Embedded Architecture Basics”, MSDN.Microsoft.com, December 18, 2001

감사의 글

지난 2년 동안 부족한 저를 배려하고 지도해 주신 임재홍 교수님께 진심으로 감사드리며, 교수님의 지도와 격려에 더욱 열심히 했어야 했다는 아쉬움이 남아 죄송할 따름입니다. 또한 바쁘신 일정에도 논문을 지도해 주시고 심사해 주신 이상배 교수님, 심준환 교수님께도 진심으로 감사드립니다. 그리고 학교 생활을 지도해 주신 학과 교수님께도 감사드립니다.

아울러, 대학원 생활을 하고 논문을 완성하기까지 아낌없이 지원을 해 주신 신송아 선배님, 유선영 선배님, 김창수 선배님, 정성훈 선배님, 모수종 선배님께 감사드립니다. 또한 생활을 같이 하며 지내던 조원희 군, 임창묵 선배님과 저에게 격려해 주었던 DCN_LAB 여러 선배님들에게도 고마움을 전합니다. 그리고 여러 가지 신경을 써주었던 박시형 선배님에게도 감사의 인사를 전합니다.

항상 곁에서 도움을 준 친구들인 김경준, 정기원 군에게 고마움을 전하며, 학과 동기들에게도 감사한 마음을 전합니다.

어려운 여건 속에서도 자식 뒷바라지를 하시며 격려와 용기를 주신 어머니께 감사드리며, 옆에서 큰 도움이 된 동생에게도 고마운 마음을 전합니다.

많은 분들의 도움으로 이루어진 논문인 만큼 더욱 열심히 살라는 가르침으로 알고 마음에 새기고 항상 노력하겠습니다.

본 연구는 산업자원부의 지역혁신 인력양성
사업의 연구결과로 수행되었음