
RFID를 이용한 항만 컨테이너 관리 시스템 설계 및 구현

Design and Implementation of Port Container Management System Using RFID

노철우, 김경민
신라대학교 컴퓨터정보공학부

Cheul-Woo Ro(cwro@silla.ac.kr), Kyung-Min Kim(sillamin@silla.ac.kr)

요약

본 논문은 유비쿼터스의 핵심 기술로 대두되고 있는 RFID를 이용한 항만 컨테이너 관리시스템을 개발 하는데 목적을 둔다. 본 시스템은 컨테이너 자동식별을 위해 RFID 기술을 활용한 항만 게이트 자동화 기술과 게이트와의 통신 및 컨테이너의 입출고 처리를 위한 웹 프로그램 기술, 그리고 항만 네트워크 플랫폼과 크레인 단말을 위한 무선 임베디드 라우터 기술을 기반으로 설계 및 구현 되었다. RFID를 이용하여 개발된 본 시스템은 향후 지능형 항만을 향한 항만 물류 관리 시스템의 주요소로 활용될 것으로 기대된다.

■ 중심어 : | RFID | 컨테이너 관리 시스템 | 게이트 자동화 | 무선 임베디드 라우터 |

Abstract

In this paper, we address to develop a port container management system using RFID, which is one of the main technologies in ubiquitous. Our system is designed and implemented based on the following technologies: a port gate automation using RFID for container identification, web programming for both communication of the gate and deposit/delivery of a container, and wireless embedded router technology for both port container network platform and crain terminals. The developed system using RFID will be expected to use in main part of the port logistic management system aimed to the intelligent port.

■ keyword : | RFID | Container Management System | Gate Automation | Wireless Embedded Router |

I. 서론

미래 IT 시장을 주도할 기술로 유비쿼터스 환경이 주목받고 있으며 유비쿼터스 컴퓨팅의 핵심 기술로 RFID (Radio Frequency IDentification)가 등장하고 있다[1][2].

정보통신부에서는 유비쿼터스 사회의 도래에 대비하기 위해 IT839 전략[3]을 추진 중이며 IT839 전략 중의 첨단 인프라로 RFID/USN[4]이 구축 중에 있다. 해양수산부는 컨테이너의 이동경로를 언제 어디서든 자동으로

추적할 수 있고 통관에 걸리는 시간과 물류비용을 절감시키기 위해 항만을 통하는 모든 수출입 물류에 RFID (전자태그)를 기반으로 실시간 경로 추적이 가능한 ‘유비쿼터스 항만(U-포트)’ 시스템을 구축하려 한다.

본 논문의 구성은 II장에서는 RFID 관련연구에 대해 설명하고 III장에서는 항만 컨테이너 관리 시스템의 구성도와 시스템 구현에 대해 설명하고 IV장에서는 결론을 맺는다.

II. 관련연구

RFID 시스템은 태그, 리더, 서버(미들웨어 및 응용 서비스 플랫폼)로 구성되고 유무선 통신망과 연동되어 사용된다. 기본적인 동작원리는 RFID 태그 안에 내장된 안테나가 리더로부터 전파를 수신하고 내장된 IC칩을 기동하여 칩 안의 정보를 신호화하여 리더의 안테나를 통하여 전파를 통하여 신호를 발신한다. 리더는 발송된 신호를 안테나로 수신하고 수신된 정보를 유무선 통신방식에 의해 서버로 전달한다[5]. RFID 시스템은 저주파(125KHz, 134KHz), 고주파(13.56MHz), 극초단파(433MHz, 860~900MHz) 및 마이크로파(2.45GHz)등의 무선 주파수 대역을 이용하고 있으며 주파수 대역별로 응용 분야가 다르다. 고주파대 제품은 주로 13.56MHz를 사용하여 출입 통제 보안, 스마트카드, 버스카드 등에 사용되며, 433.92MHz 대역은 미국 등에서 일부 컨테이너 관리용으로 사용하고 있으며 860~960MHz 대역은 전 세계적으로 유통, 물류 등의 용도에 적합한 대역으로 전망되고 있다. RFID 미들웨어는 리더에서 계속적으로 발생하는 식별코드 데이터를 수집, 제어, 관리하는 기능을 한다.

RFID의 이용은 칩의 가격, 크기, 성능 등 센서 기술의 발전에 따라 물류/유통분야, 환경, 재해예방, 의료관리 및 식품관리 등 실생활의 활용에 확대될 것으로 전망된다. 국내에서는 산업자원부, 조달청, 국립수의과학검역원, 공항공사, 국방부 등이 RFID 선도 시범사업자로 선정되어 서비스에 나설 계획이다.

본 논문에서는 RFID 시스템을 이용한 항만 게이트 자동화 기술과 임베디드 기술을 활용한 무선 패킷 통신 및 라우터를 포함한 네트워크 플랫폼 기술이 통합된 항만 컨테이너 관리 시스템을 개발한다. RFID 태그 정보를 리눅스 임베디드 시스템에 RFID 리더기를 접목하여 미들웨어의 역할을 수행하고 윈도우용 응용 서비스 플랫폼으로 데이터를 전송하도록 구현하고 응용 서비스 플랫폼으로 데이터를 전송하는 네트워크 플랫폼으로 임베디드 시스템을 이용하여 무선라우터기능을 구현한다.

III. 시스템 설계 및 구현

1. 시스템 구성도

RFID 태그를 컨테이너에 부착하고 부착된 컨테이너의 항만 게이트 출입 시 태그에 의한 컨테이너 번호 자동 식별로 게이트 자동화를 수행한다. 이를 기반으로 한 항만에서의 컨테이너 관리를 수행하는 시스템의 구성도는 [그림 1]과 같다. 이 시스템은 입고 시 컨테이너에 부착되어 있는 RFID 태그 신호를 자동으로 인식하고 이를 관리서버로 전송하는 RFID 시스템, 필터링된 태그 신호와 컨테이너 식별정보를 매핑하여 컨테이너의 입출고 관리 및 컨테이너 야적정보에 관한 웹 서비스를 제공하는 컨테이너 관리 서버, 항만 야적지의 담당 크레인 기사가 컨테이너의 실제 야적 입출고 작업을 처리하는 크레인 단말, 그리고 컨테이너 관리서버 및 크레인 단말 등을 무선으로 연결 시켜주며 항만의 망 관리를 수행할 수 있는 무선 라우터로 구성된다.

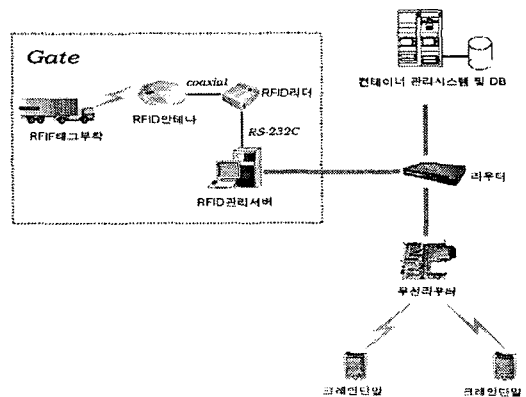


그림 1. RFID를 이용한 항만 컨테이너 관리 시스템 구성도

2. RFID 시스템

2.1 하드웨어 구성

본 논문에서 구현한 RFID 시스템은 KIS900RE장비[6]와 RFID 태그신호를 지속적으로 감지하여 관리서버로 전달해주기 위해 PXA270X CPU를 탑재한 임베디드 보드로 구성된다.

KIS900RE 장비는 910~914MHz의 주파수 대역을 사용

하고 태그의 인식할 수 있는 감지 거리는 30cm~5m 정도이다.

PXA270X보드[7]는 전원 제어를 통한 저전력 설계, 크기 최소화 및 보드 적층형 구조를 채택한 임베디드 보드로 (주)코어벨과 함께 개발하였다. [표 1]은 PXA270X 보드의 사양이다.

표 1. PXA270X보드의 사양

CPU	Pxa270, 520MHz
Memory	Flash(64M), SDRAM(64M) 1.8V VCC & I/O 저전력
Ethernet	LAN91C111, Fully Integrated IEEE 802.3/ 802.3u-100Base-TX/10Base-T Physical Layer
USB	USB 1.1 Host Support
Connector	Main Connector, LCD Connector
LED	+5V, RUN, RST, LNK, ACT
I/O Board	PCMICA, CF, Camera Interface

PXA270X 보드를 통해서 RFID 리더와 시리얼 통신을 하고 RFID 태그 발생 이벤트를 처리하여 결과를 파일로 생성하고 이를 컨테이너 관리서버로 전달한다.

[그림 2]는 RFID 리더와 PXA270보드를 이용하여 구현된 RFID 시스템의 내부이다.

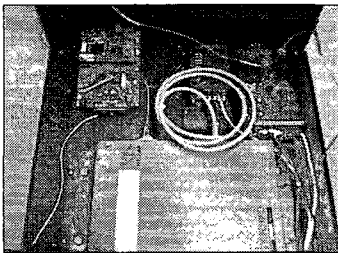


그림 2. RFID 시스템

PXA270 보드와 RFID 리더기는 시리얼 케이블로 연결하고 리더로부터 읽어 들인 태그 신호를 연결된 시리얼 케이블을 통해 PXA270보드로 전송된다. 전송된 태그 신호는 PXA270 보드에 탑재되어 있는 RFID클라이언트 통신 모듈을 통해 관리서버의 윈도우서버통신 모듈과 소켓 통신한다. [그림 3]은 RFID 시스템의 통신 모듈 구조

를 보여준다.

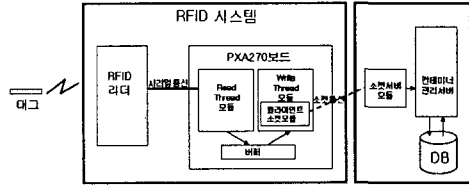


그림 3. RFID 시스템의 통신모듈 구조

2.2 RFID 통신 모듈

RFID 시스템의 통신 모듈은 PXA270 보드에 탑재되어 있는 리눅스용 클라이언트 통신 모듈과 이를 통해 관리서버에서 전송된 정보를 받아들이는 윈도우용 서버통신 모듈로 개발하였다.

리눅스용 클라이언트 통신 모듈[8]에는 RFID리더와의 통신을 위해 RS232C 통신을 위한 모듈을 포함하고 있으며 RS232C를 통해 전달된 신호 중에서 태그 정보에 해당하는 신호만 필터링하여 TCP/IP 소켓 통신을 통해 관리서버의 윈도우용 서버 통신 모듈[9]로 정보를 전달한다. 리눅스용 클라이언트 통신 모듈은 read와 write 쓰레드를 각각 생성하여 RFID로부터 전달되는 태그신호를 지속적으로 감지하여 관리서버의 윈도우용 통신 모듈로 필터링되는 대로 정보를 전달한다.

RFID리더와 시리얼 통신을 위해서는 기본적으로 Baud rate : 38400 , Data bit = 8, Parity = N, Stop bit = 1로 설정을 해주어야 한다. 이를 위해 read 쓰레드에서 initComport() 프로시저를 추가해 주었다.

```

int initComport()
{
    /*시리얼 포트 개방확인*/
    fd = open(RFID_SERIAL_PORT, O_RDWR | O_NOCTTY);
    ...
    tcgetattr(fd, &oldtios);
    memset(&newtios, 0x00, sizeof(newtios));
    /* 포트 설정 */
    newtios.c_cflag = B38400 | CS8 | CLOCAL | CREAD;
    newtios.c_iflag = IGNPAR | ICRNL;
    newtios.c_oflag = 0;
    newtios.c_lflag = 0; // non_canonical
    newtios.c_cc[VTIME] = 50;
    newtios.c_cc[VMIN] = 0;
    tcflush(fd, TCIFLUSH);
    tcsetattr(fd, TCSANOW, &newtios);
}
    
```

read 쓰레드는 시리얼 통신으로 들어오는 태그신호를 계속적 감지하면서 태그 신호의 값이 E로 시작하고 크기가 10이상인 신호를 필터링하여 버퍼에 저장한다. 이때 윈도우용 서버 통신모듈에서 태그정보를 하나씩 추출하기 위하여 종료문자를 추가하여 저장한다.

write 쓰레드는 필터링하여 저장된 정보를 관리서버로 소켓 통신을 통해서 전달한다. 이를 위해서 소켓을 연결하고 read 쓰레드에서 저장된 버퍼의 내용을 소켓 통신으로 전송한다.

서버용 통신모듈은 윈도우용 소켓을 생성하여 리눅스 클라이언트 통신모듈로부터 전달받은 내용을 태그정보 단위로 태그 정보 값과 생성된 날짜와 시간 정보를 파일에 기록한다.

```

while(1)
{
    /*전송메시지 확인 */
    if ( strlen > 0 )
    {
        iOffset = 0;
        for ( iCnt = 0; iCnt < strlen; iCnt++ )
        {
            /*RFID 태그 종료 신호 확인
            ETX = '!'*/
            if ( message[iOffset] == ETX )
            {
                ..
                if ( rflen == 16 ) { /*태그길이 확인*/
                    time(&now); /*시간설정 */
                    t="localtime(&now);
                    /*같은 시간대에 저장된 태그 정보 필터링 */
                    ret = strcmp(buffTime, buffOld, len);
                    /* 파일에 기록 */
                    if ( ret != 0 ) {
                        file = fopen("RFID.txt", "a+");
                        fputs(buffTime,file);
                    }
                    /*같은 시간대에 저장된 태그 정보 필터링-현재정보저장 */
                    strcpy(buffOld, buffTime);
                }
            }
            iOffset++;
        }
    }
}
    
```

3. 컨테이너관리서버 및 터미널 단말

컨테이너 관리서버는 RFID 시스템으로부터 들어오는 컨테이너 태그 정보를 윈도우용 서버 통신 모듈에서 받아 텍스트파일에 기록한 다음 텍스트 파일의 정보를 읽어 컨테이너 식별정보와 매핑하여 데이터베이스에 자동 저장하여 관리하며 컨테이너 관련 작업정보를 무선 라우터

를 경유하여 크레인 단말 시스템에 전송하고 작업 결과를 수신 받아 이에 필요한 작업 관리를 수행하도록 구현하였다.

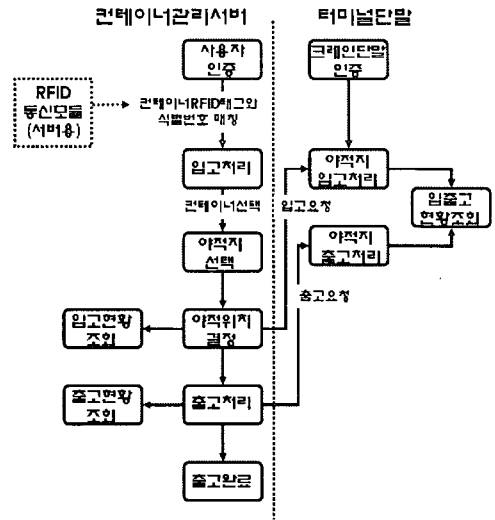


그림 4. 컨테이너관리서버 및 터미널 단말 흐름도

[그림 4]는 컨테이너 관리서버의 관리용 프로그램의 흐름도이다. 컨테이너 관리서버는 윈도우 2000서버의 IIS 웹 서버 환경 하에서 ASP로 웹 프로그램을 작성하였고 데이터베이스 서버로는 MS-SQL서버 2000을 사용하여 구현하였다.

3.1 입고처리

인증 과정을 거친 후 RFID 시스템에 의해 게이트에서 인식된 RFID 태그정보는 RFID 시스템을 통하여 관리서버로 전송되어 파일로 기록되고 기록된 태그정보와 컨테이너 식별번호를 자동으로 매핑하여 데이터베이스에 저장하면 입고처리에서 [그림 5]와 같이 컨테이너 정보가 나타난다. [그림 5]에서 컨테이너 식별 번호를 클릭하면 [그림 6]과 같이 항만의 야적지 별로 입고 가능 정보를 보여주고 입고를 원하는 야적지를 선택한다. 야적지를 선택하면 [그림 7]과 같이 해당 야적지의 입고 가능한 위치를 보여주고 입고 위치를 선택하여 입고 요청을 크레인 단말로 보낸다.

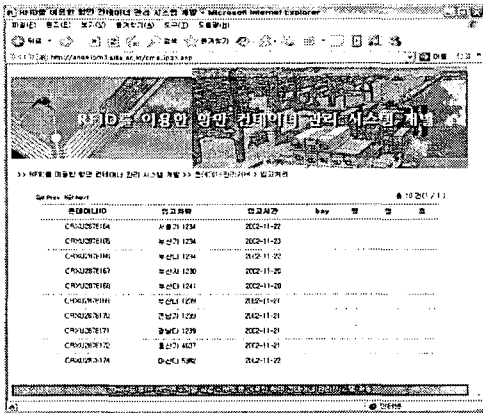


그림 5. 입고처리컨테이너선택

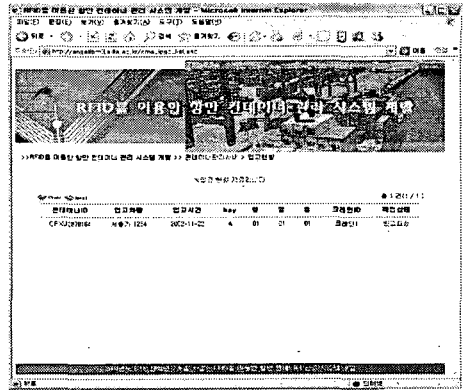


그림 8. 입고 현황 확인

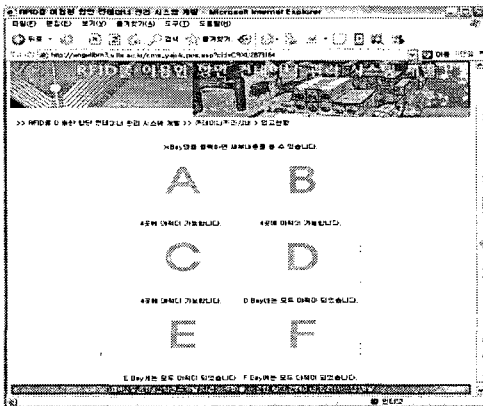


그림 6. 야적지 선택

3.2 터미널 단말 - 입고처리

항만 야적지의 크레인으로 입고 요청이 되면 해당 크레인 기사의 단말기에는 [그림 9]와 같이 요청된 컨테이너 정보가 나타나고 크레인 기사는 컨테이너를 지정된 위치에 놓은 후 [그림 10]과 같이 입고 처리를 하게 된다. 입고 처리가 되면 [그림 8]과 같이 관리서버에서 입고현황을 확인할 수 있다.

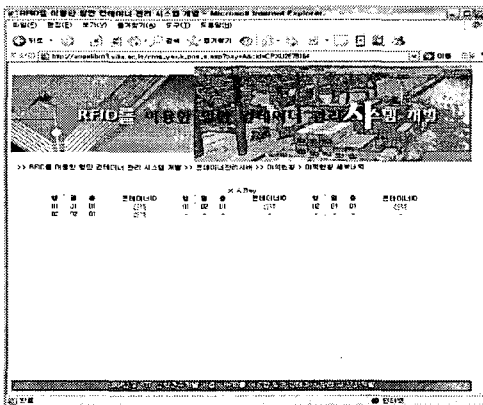


그림 7. 야적 위치 선택

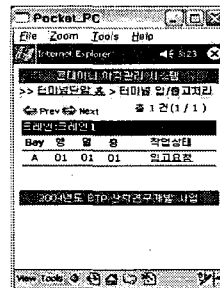


그림 9. 크레인단말 입고요청

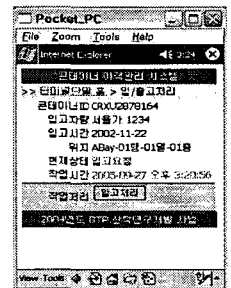


그림 10. 크레인단말 입고처리

3.3 출고 처리

입고 처리가 되어 야적지에 있는 컨테이너를 출고하기 위해서 컨테이너 관리서버에서 해당 야적지의 터미널 단말에 출고 요청을 하게 되면 해당 크레인 기사의 터미널 단말에 출고 요청 정보가 나타나게 되고 출고 작업 후 터미널 단말에서 출고처리를 하면 컨테이너가 터미널 게이트를 빠져나가게 되고 출고 완료가 된다.

4. 임베디드 무선라우터

무선 라우터는 MPC 850 칩을 사용하는 무선 랜 기반 네트워크 보드를 사용하여 구현하였다. 이 보드에 무선 통신을 위한 PCMCIA 기능, 무선 랜 브리지 기능을 담당하는 AP(Access Point), 그리고 Zebra[10]를 기반으로 하는 무선 라우터 프로토콜을 임베디드 리눅스 시스템에 통합하여 구현하였다[11].

Zebra는 공개 소프트웨어로 TCP/IP 기반의 라우팅 프로토콜을 다룬다. RIPv1, RIPv2, OSPFv2와 RFC1771에 기술된 BGP4 프로토콜을 지원한다. Zebra는 각 라우팅 프로토콜별 프로세스를 가지고 있는 방식으로 설계되어 있으며, 멀티스레드 기법을 커널 지원여부에 상관없이 사용할 수 있어 유연성과 신뢰성을 가지고 있다. 또한 프로토콜에 프로그램상 오류가 발생하여도 각각의 모듈은 독립적으로 업그레이드가 가능하게 설계되어 있어 빠른 업그레이드를 통한 시스템 안정화가 가능한 라우팅 프로토콜 소프트웨어이다.

Zebra의 동작 과정은 NIC를 통해 전달된 패킷은 커널을 거쳐 netlink를 통해 Zebra 데몬에 전달된다. Zebra 데몬은 자신에 전달된 패킷을 각 라우팅 프로토콜로 전달하는 방식으로 중계 역할을 하여 전용 라우터로서 동작을 하게 된다.

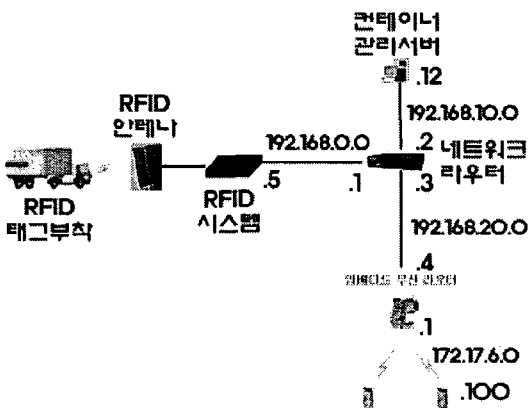


그림 11. 전체 망 구성도

[그림 11]은 임베디드 보드를 이용한 무선 라우터를 비

롯한 시스템에서 실제 구현된 전체 망 구성도를 보여준다.

무선 라우터를 구현하기 위한 임베디드 보드는 (주)코어벨의 LDS3000보드를 이용하였고 Zebra를 다음과 같이 설치하였다. 리눅스 서버에 커널 이미지를 생성하고 리눅스 서버에 Zebra 홈페이지[10]에서 설치파일을 /usr/local/에 다운로드하고 설치한다. Zebra 원격 접속을 위해 리눅스 서버의 램디스크 설치 디렉토리의 /etc/services 파일에 다음 내용을 추가한다.

```

vi /etc/services
#zebra
zebrasrv      2600/tcp
zebra         2601/tcp
ripd          2602/tcp
ripngd        2603/tcp
ospfd         2604/tcp
bgpd          2605/tcp
ospf6d        2606/tcp
    
```

Zebra Makefile을 크로스 컴파일러로 사용할 수 있도록 변경한 후 컴파일한 후 Zebra설치 디렉터리에서 zebra, ripd 2개의 파일을 램디스크 설치 디렉터리 /usr/local/sbin에 복사한다.

램디스크 설치 디렉터리의 /etc 디렉터리의 rc.sysinit 파일에 유선망과 무선망에 대한 정보를 설정하고 Zebra 데몬을 추가하고 무선망을 사용하기 위한 무선 채널을 다음과 같이 추가한다.

```

echo "Ethernet/Wireless LAN Port Roding....."
ifconfig wlan0 172.17.6.1 netmask 255.255.255.0 broadcast 172.17.6.255 up
ifconfig eth0 192.168.20.4 netmask 255.255.255.0 broadcast 192.168.20.255 up

echo "Route Protocol Zebra Loading....."
/usr/local/sbin/zebra -d

echo "WLAN Channel 5 Acting....."
/usr/local/sbin/iwconfig wlan0 channel 5
    
```

망 구성을 설정하기 위하여 Zebra의 configuration file을 아래와 같이 수정한 후 램 디스크 이미지를 생성한다.

```

hostname router_1
password router
enable password router
!
interface wlan0
ip address 172.17.6.1/24
!
interface eth0
ip address 192.168.20.4/24
!
ip route 172.17.6.0/24 wlan0
ip route 192.168.20.0/24 eth0
ip route 192.168.10.0/24 192.168.20.3
ip route 192.168.0.0/24 192.168.20.3

```

[그림 12]은 이렇게 구성된 라우터의 라우팅 테이블 내용을 보여 준다.

Destination	Gateway	Genstat	Flags	Metric	Ref	Use	Face
192.168.20.0	*	255.255.255.0	U	0	0	0	eth0
192.168.0.0	192.168.20.3	255.255.255.0	UG	0	0	0	eth0
172.17.6.0	*	255.255.255.0	U	0	0	0	wlan0
192.168.10.0	192.168.20.3	255.255.255.0	UG	0	0	0	eth0

그림 12. 라우팅 테이블

IV. 결론 및 향후과제

본 논문에서 구현한 RFID를 이용한 항만 관리시스템은 유비쿼터스의 필수분야인 RFID 기술과 IT의 중요 기술 분야인 임베디드 시스템 기술의 통합 기술로 시스템을 개발한데 의의가 있다. RFID 시스템을 이용한 항만 게이트 자동화 기술과 임베디드 기술을 활용한 무선 패킷 통신 및 라우터를 포함한 네트워크 플랫폼 기술이 통합된 본 항만 컨테이너 관리 시스템은 정보 산업의 응용 영역인 항만 컨테이너 정보산업 기술의 파급 효과를 가져 올 수 있을 것이다. 또한 RFID를 이용한 컨테이너 자동식별 기술, 무선 라우터 개발 기술과 항만 컨테이너 관리를 위한 웹 시스템 통합 기술 그리고 크레인에서 사용될 단말 기술들의 통합은 향후 지능형 항만과 유비쿼터스 센스네트워크를 향한 항만 물류 관리 시스템에 주요 소로 크게 활용될 수 있을 것이다.

본 논문에서 구현된 RFID 시스템은 유니 태그를 인식하는 900MHz RFID 리더기를 사용한 것이다. 900MHz

는 거리상에서 효율적이지만 컨테이너 부착용으로는 433MHz를 선호하고 있는 추세이다. 즉 리더기 자체가 멀티채널 및 다양한 태그를 인식할 수 있는 방향으로 발전되어야 한다. 또한 현 유니 태그나 EPC 클래스 0 태그는 read 기능만을 갖고 있어 추후 read/write가 가능한 태그로 발전되어야 하며 write 기능이 가능한 태그를 이용하여 다양한 기능을 수용하는 컨테이너 관리 시스템을 연구하고 크레인 기사가 사용할 크레인 단말기는 현재 PDA에 구현하였으나 웹 패드나 보다 간단한 보드를 개발하여 크레인 전용 단말기를 개발하여 좀 더 효율적으로 작업을 할 수 있도록 연구할 것이다. 또한 RFID 태그를 이용한 컨테이너 식별번호 자동인식에 의한 게이트 자동화이외에 컨테이너 위치추적에 의한 항만 물류관리 시스템 개발을 지속적으로 연구할 예정이다.

참고 문헌

- [1] 이근호, “무선식별(RFID) 기술”, TTA저널, 제89호, pp.124-129, 2003.
- [2] 고현봉, 오영철, 유승화, “RFID 표준화 동향”, Telecommunications Review, 제15권, 제2호, pp.244-256, 2005.
- [3] 정보통신부, 국민소득 2만불로 가는 길 [IT839 전략], 2005.
- [4] 유승화, “RFID/USN 표준화 추진방향”, TTA저널, 제94호, pp.12-18, 2004.
- [5] 유승화, “RFID 기술 현황 및 활용분야”, 정보과학회지, 제23권, 제7호, pp.64-70, 2005.
- [6] <http://www.kiscom.co.kr/>
- [7] <http://www.corebell.com/>
- [8] 웨런 W. 게이, 예제로 배우는 리눅스 소켓 프로그래밍, 인포북, 2004.
- [9] 윤성우, TCP/IP 소켓 프로그래밍, 프리렉, 2003.
- [10] <http://www.zebra.org>
- [11] 노철우, 방석운, “GNU Zebra를 이용한 임베디드 리눅스 라우터 개발”, 신라대학교 자연과학연구소 자연과학논문집, 제12집, pp.255-266, 2003.

저자 소개

노철우(Cheul-Woo Ro)

종신회원



- 1980년 2월 : 서강대학교 물리학
과(학사)
- 1982년 2월 : 동국대학교 전자계
산학과(석사)
- 1995년 8월 : 서강대학교 전자계
산학과(박사)

▪ 1991년~현재 : 신라대학교 컴퓨터정보공학부 정교수
<관심분야> : 이동통신 시스템, 통신 프로토콜 설계/
검증/시험, 페트리 넷 모델링 및 성능, RFID, 임베
디드 시스템

김경민(Kyung-Min Kim)

정회원



- 1993년 2월 : 신라대학교 전자계
산학과(이학사)
- 2000년 8월 : 신라대학교 컴퓨터
교육학과(교육학석사)
- 2005년~현재 : 신라대학교 컴퓨
터정보공학부 박사과정

<관심분야> : 이동통신 시스템, 페트리 넷 모델링
및 성능, RFID, 임베디드 시스템