

논문 2006-43CI-4-2

LACC를 이용한 원격제어 시스템 구현 및 적용

(Implementation and application of remote control system using LACC(Local Area Control Center))

박 태 진*, 장 명 기**, 정 승 렬*

(Tae-Jin Park, Myung-Kee Jang, and Seung-Ryul Chung)

요 약

요즘, 에너지 절감이라는 전 세계의 요구에 따라 많은 연구와 응용개발이 이루어지고 있다. 이러한 응용개발에는 INTERNET, LAN, PLC, ZIGBEE 등과 같은 유무선 네트워크 장치를 이용하고 네트워크 상에서 발생할 수밖에 없는 제어시스템에서의 치명적인 문제를 비교적 저렴하고, 작은 노력을 들여 구축할 수 있는 방법에 대한 연구가 이루어져야 한다. 또한, 네트워크 기능을 가진 임베디드 시스템의 경우 펌웨어 업그레이드가 타 장치에 비해서 장점이 많으므로 네트워크 노드들을 원격으로 관리하고, 신뢰성 있도록 하는 제어방법이 중요할 것이다. 본 논문에서는 센서 노드 및 릴레이를 관리하는 PLC 영역과 ESS의 핵심요소인 LACC 구현과 신뢰성 있는 결과를 얻기 위한 성능분석 및 평가를 행함으로써 본 논문에서 제안된 효과적 원격관리와 제어방법 등을 실험결과를 통해 검증한다.

Abstract

Nowadays, there is called energy saving, that it is accomplished with a lot study and application development as according to demand of all around world . It is using to network device of wire or wireless such as internet, lan, plc, and zigbee and that we should be accomplished to study about method that spending a little cost to fatal error in control system that be expected of frequent occurrence on network, and that have method to do construction a little bit tried out. Also, in that such a embedded system with network function case, there is important thing that it is remote management to network node and control methods to trust because firmware upgrade have the advantage of more than the other device. In this paper, we have verified through a result of experiment for efficient remote management and control method as previously stated in this paper, that it is area of the PLC for sensor node and doing management to relay, and implementation of the LACC which is core part of the ESS, and in order to get a result to trust from doing analysis of capacity and test.

Keywords : PLC(Power Line Controller), Energy saving, Remote management and control, Supervisory control, Embedded system / 전력선제어, 에너지 절감, 원격관리와 제어, 감시제어, 임베디드 시스템

I. 서 론

ESS(Energy Saving System)는 전 세계적인 에너지 절약 시책에 편승, 전력사용이 높은 기업 및 공공기관에 적용될 수 있으며 에너지 절약을 목적으로 한다. 일부 기업의 영업점에서는 기존 설비를 쉽게 활용할 수

있도록 전력선을 사용하는 LonWorks 시스템으로 설비가 되며 LonWorks 시스템은 전용 PLC (Power Line Controller/Communication)^[4]에 의해 센서의 값을 수집하고 액츄에이터를 직접 제어하게 된다. 여기에 PLC와 함께 LACC (Local Area Control Center)가 설치되어 사용되는데, LACC는 최적화된 알고리즘으로서 액츄에이터를 제어하고 최고의 효율을 가지는 에너지 절약 시스템을 목적으로 하고 있다.

임베디드 시스템은 이전 세대의 마이크로컴퓨터를 이용한 시스템과 비교하여 네트워크가 가능하다는 점에서 많은 활용도를 가지고 있고, 포스트PC 시대를 이어

* 정회원, 마산대학 지능형 홈 디자인학과
(Dept. of Intelligent Home & Design, Masan College)

** 학생회원, 한국과학기술원 전자공학과
(Dept. of Electronics Engineering, KAIST)
접수일자: 2006년4월17일, 수정완료일: 2006년7월3일

갈 유비쿼터스 컴퓨팅의 구현 방법으로 학계, 업계를 중심으로 많은 연구와 시도가 이루어지고 있다^{[2][3]}. 기존의 프로그램은 하드웨어를 제어하는 펌웨어 수준에서 이루어진 반면, 현재 네트워크 망과의 복잡한 통신을 고려하고 동시에 기존의 제어 기능을 수행하는 등의 다중처리가 요구되면서, 하드웨어와 응용프로그램 사이에서 안정적인 프로세스 처리를 도와주는 운영체제가 임베디드 시스템에도 도입되기에 이르렀다^{[1][2]}.

본 논문에서는 ESS (Energy Saving System)를 구성하는데 필요한 LACC (Local Area Control Center)의 알고리즘 구현과 신뢰성 있는 결과를 얻기 위한 성능분석 및 평가를 행함으로써 제안된 본 논문의 효율성을 실험결과를 통해 보이고자 한다.

II. ESS의 구성

1. ESS의 구성 개요

ESS(Energy Saving System)는 에너지 절감이라는 국가적인 요구사항에 따라 대규모 편의점에 적용, 목적을 달성할 수 있는 시스템이다. LACC는 PLC 블록과 네트워크망에 접속하여 데이터를 전송, 관리하게 된다.

가. ESS 구성

ESS는 그림 1과 같이 웹 서버(시간 서버), DB 서버, DCS(Data Collection Server) 그룹, 그리고 DCS를 묶는 LA로 스위치로 구성된다. LACC는 여러 인터넷 사업자의 회선을 통해 서버에 접속된다.

(1) 웹 서버

웹 서버의 PLC 정보를 실시간으로 웹 페이지를 통해 보여준다. 또, 웹 서버 프로그램이 DCS IP, 접속 주기, 현재 날짜와 시간을 LACC가 접속할 때 마다 전송해준다.

(2) LACC 그룹

LACC가 접속하여 현재 데이터와 평균데이터를 전송한다. 함께 DB 서버에 접근하여 데이터를 획득한다.

(3) DCS 그룹

DCS에 최대 15,000대의 LACC가 접속할 경우 적체되는 것을 막기 위해 스위치를 사용하여 부하를 적당하게 나눠준다.

(4) DB서버

DCS에 있는 데이터를 데이터베이스 파일로 변환하여 통계화 한다.

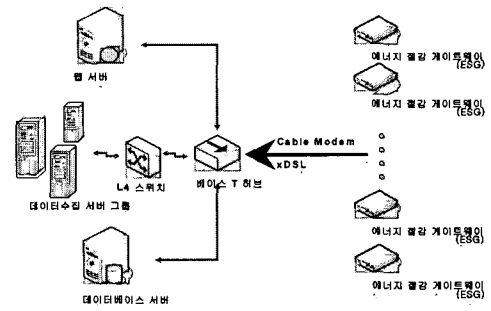


그림 1. 하드웨어 시스템 구성
Fig. 1. Configuration of hardware system.

나. LACC 구성

PLC로부터 얻은 센서 값에 사용자 알고리즘을 적용하여 노드의 액츄에이터를 제어하는 PLC-LACC 인터페이스와 PLC로부터 얻은 센서 및 액츄에이터의 값을 수집, 평균, 저장하여 네트워크를 통해 원격 서버로 전송하는 LACC-Server 인터페이스로 구분한다.

궁극적으로 서버로 전송된 데이터는 데이터베이스화 되어 관리가 된다. PLC의 사용자 알고리즘은 로컬의 엔지니어에 의해 변경되지만, 최종적으로는 데이터베이스의 분석을 통해 이루어진다. 즉, 기업의 각 영업점에서는 시간, 요일, 달, 계절 등의 조건을 분석하여 알고리즘이 적용되고, 여러 영업점을 비교해서 효율이 좋은 영업점의 알고리즘이 적용될 수 있다.

2. 시스템의 구성

하드웨어 시스템의 구성은 센서 노드 및 릴레이를 관리하는 PLC(Power Line Controller/Communication) 블록과 네트워크 망에 접속을 하여 데이터를 전송 관리해주는 LACC(Local Area Control Center) 블록, LACC로부터 전송된 데이터를 관리, 모니터링할 수 있는 통제 서버로 구성될 수 있다. 특히, 부터 로더, 리눅스 커널^[5], 하드웨어 디바이스 드라이버, LACC 애플리케이션 프로그램으로 구성되어 있으며 LACC를 구동하기 위해서

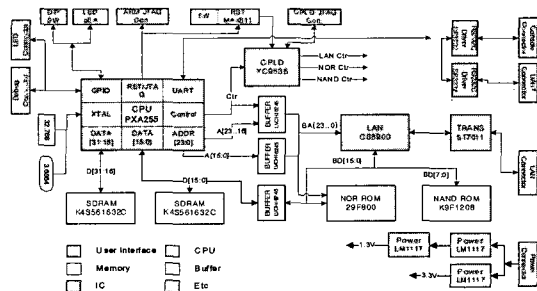


그림 2. LACC의 블록도
Fig. 2. Block diagram of LACC.

SSH, Firewall, Linux 네트워크 패키지가 운영체제에 추가적으로 설치되어야 한다. 그림 2는 LACC의 블록 다이어그램을 나타낸다.

III. 시스템의 설계 및 구현

1. 시스템 설계의 개요

이 시스템은 부터 로더, 리눅스 커널, 하드웨어 디바이스 드라이버, LACC 애플리케이션 프로그램으로 구성되며 네트워크 노드에서의 신뢰성 있는 원격관리와 제어를 목적으로 하고 산업현장에서 직접 활용될 수 있도록 설계되었다.

2. 주요 설계 요소

원격제어에 있어서 제어의 신뢰성, 고장의 유무를 판단할 수 있는 자가 진단기능 및 도움 요청 기능, 네트워크 망에서의 해킹에 의한 오용을 방지 할 수 있는 보안 기술, 원격지에서 긴급 제어권 할당 및 원격 펌웨어 업그레이드를 통한 복구 용이성을 가져야 한다. 이와 같은 성능문제를 해결하기 위해서 몇 가지 설계를 고려하였다^[6-8].

가. Watchdog Reboot

하드웨어적인 문제 발생으로 동작이 멈출 경우, 외부에 독립적으로 장착되어 있는 실시간 클럭(Clock)과 프로세싱 플래그(Processing flag)에 의해 LACC 전원을 재 부팅 시켜 주며, 그때의 상태 레지스터와 시간 정보를 외부에 설치되어 있는 EEPROM에 기록을 한다. 또한 재 부팅 카운터를 증가 시키면서 오류의 상태와 재 부팅된 상태를 EEPROM하나만으로 검출 할 수 있게 하였다.

나. Control and Feedback Protocol

LACC 블록과 실제 노드를 제어하는 PLC 노드사이에는 양방향의 제어 및 피드백을 수행할 수 있는 통신 규약을 정의함으로써 노드의 상태를 LACC에서 관찰할 수 있고, 노드를 직접 제어 할 수 있으며, 위험 상황 발생 시, 경고 메시지를 제공할 수 있는 기능을 포함하도록 하였다.

다. Network Control Protocol

1000개 이상의 다수의 노드가 동시에 하나의 서버에 접근함으로써 데이터 접속 폭주가 이루어 질 경우 서버

가 오동작 할 수 있으며, 처리과정이 느려져 실시간 제어 감시가 어려울 수 있다. LACC에서는 한국표준과학 연구소에서 제공하는 실시간 시간 정보를 이용하여, LACC 보드의 시간을 동기화 시키며, 각각의 ID에 따라 동시 접속 시간을 서버에서 조정가능 하게 하였다. 그리고 랜덤 접근 기법을 이용하여 제한된 시간에 확률적이면서 유니폼하게 접근 시간을 분산함으로써 서버에 오는 부하를 줄이는 방법을 택하였다.

라. Dure Core Observation System

임베디드 시스템에서는 동작을 제대로 하고 있는 지를 판단할 수 있는 주 프로세서 외의 감시 프로세서가 필요하다. 특히 엄격한 제어가 필요한 상황에서는 여러 상황 발생시에 가장 안정한 모드로 제어권을 반환하여 제어할 필요가 있으며, 제어함으로써 하나의 시스템 이상으로 발생 할 수 있는 위험한 상황에서 컨트롤 장치와 연결된 여러 노드들이 폭주하지 않고 안정적으로 동작할 수 있게 해 준다^[6]. 제안된 시스템에서는 PLC블록에서 노드들과 직접적으로 연결되어 LACC의 오동작 발생시에 수동제어로 동작하도록 설계 하였으며, 오류가 수정 될 경우 다시 제어권이 LACC로 반환 되어 정상상태로 복귀 가능하게 하였다.

마. FTP Firmware Updating

LACC는 네트워크기능 중 ftp를 설치하고 펌웨어가 이더넷에 연결된 상태라면 어디서든지 업그레이드를 할 수 있게 하였다. 1000개의 지역에 설치된 임베디드 시스템을 한곳에서 동시에 업그레이드를 한다는 것은 시간적, 비용적 측면에서 관리자에게 큰 효용성이 있다. Ftp에 접근하기 위해서는 각각의 임베디드 시스템에 고유 ID가 부여 되어야 하며, 서버에서는 그 고유 ID와 함께 고유 IP를 저장하고 있어야 한다. 고정IP 기반에서는 이러한 접근이 가능하지만, 유동IP 기반에서는 IP가 바뀔 때 마다 ID 정보에 따른 IP를 서버에 알려 줌으로서 서버에서 LACC의 제어권을 놓치는 일이 없도록 했다.

바. Error Managing Server

PLC 노드들의 에러상태는 PLC-LACC 프로토콜에 의해서 수집되어 지며, 각각의 데이터 상태들은 1분 단위로 파일에 저장된다. 이 값들은 에러 데이터와 상태 데이터로 구분되어 지며, 서버에서는 등록된 LACC보드의 상태 값들을 매 분마다 읽어서 모니터 창에 나타

나도록 한다. 상태 값들의 분석과 통계를 구하여 기존의 예측된 값보다 차이가 날 경우 사용자에게 알려준다. LACC자체가 고장 날 경우에는 WatchDog(재 부팅 감시자)에 의해서 부팅이 이루어지며, 그때 저장된 상태 레지스터 값이 재 부팅 될 때에 서버로 보내어 저서 관리자가 상태를 점검가능하게 하였다. LACC가 완전 동작을 하지 않을 경우, 서버에서는 1분 단위로 스캔되지 않는 LACC를 찾게 된다.

3. LACC의 구동원리 및 알고리즘

LACC는 최적화된 알고리즘으로서 구동기를 제어함과 동시에 최고의 효율을 가지는 에너지 절약 시스템을 목적으로 하고 있으며 본 시스템을 기업의 각 영업점에서 수집한 데이터를 가공하여 원격 서버에 전송하는 기능도 가진다.

본 절에서는 앞에서 언급했던 시스템의 주요 설계요소를 고려하고 시스템을 구성하는데 핵심이 되는 LACC의 구동원리와 알고리즘을 제시한다.

가. 구동원리

[단계1]부팅과 동시에 시스템 환경파일을 읽어 들인 후 유동 또는 고정IP인지 확인하고 LACC 제어권을 유지하도록 한다.

[단계2]디바이스 장치에 접근하여 쓰레드를 생성, 실행시키도록 하였다.

[단계3]현재 시간의 설정과 쓰레드가 브로드캐스트 되도록 반복동작을 수행하고 PLC로부터 패킷을 검사, 메시지를 판독한 후 데이터를 설정, 상태를 통지하게 된다.

[단계4]신호 대기 및 상태를 검사하고 웹 서버에 접속하여 PLC의 정보를 실시간으로 제공하고 시간 설정, DCS(Data Collection Server) IP, 접속 주기, 날짜와 시간 등의 정보를 LACC가 접속할 때 마다 전송하게 된다.

[단계5]소켓을 생성, DCS에 접속하여 패킷 또는 패킷파일을 생성(CRC 생성), 현재 데이터를 전송하거나 패킷 파일을 읽어 들임으로서 평균데이터를 전송한다. 데이터 전송이 성공적으로 완료되면 파일로부터 패킷을 삭제하게 되는데, 결과적으로 DCS에 존재하는 데이터를 데이터베이스 파일로 변환하여 통계 값을 획득하게 되는 것이다.

나. 알고리즘

그림 3은 LACC 동작원리를 구현한 알고리즘이다.

```

Algorithm: movement principles of the LACC

[Step 1]LACC를 구동시키기 위하여 초기화를 수행
FILE *fp;
g_mode = ACTIVE_MODE;
if( fp > 0 && initConf(fp) == 1 ){
    // every search for Domain, IP, CVS ID,
    // and avrDataCycle, curDataCycle }
useDHCP = USE_DHCP_DEFAULT;
fp = fopen("/LACC/localip.conf", "r");
if( fp > 0 ){ .....; }
if( useDHCP ){.....; }
//Initialize Hardware Device
//Create to threads

[Step 2]소켓설정 및 디바이스 장치에 접근하기 위한 통신 설정
struct termios tio;
// { Serial 통신에 대한 정보 설정 .... }
memset( &tio, 0, sizeof(struct termios) );
// { Serial 통신 설정.... }
Serial_fd = open(SERIAL_DEVICE, O_RDWR);
if( Serial_fd <= 0 ){ return 0; }
tio.c_cflag = B9600 | CS8 | CLOCAL | CREAD;
tio.c_lflag = 0; tio.c_cc[VTIME] = 0; tio.c_cc[VMIN] = 1;
tcflush( Serial_fd, TCIFLUSH );
tcsetattr( Serial_fd, TCSANOW, &tio );
return 1;

[Step 4]웹 서버로부터 시간과 데이터 서버의 IP를 가져 오기
randNum = getBootIndex() % 60;
while( !g_bEnd ) { sleep(1);
// Implementation of thread for lock, wait, and unlock

if( g_mode != ACTIVE_MODE ) continue;
if( (g_threadRun&=RUN_WEB_SYNC) == 0 ) continue;
if( g_WebSync == 0 ){
    if( count < 3 ) getInfoFromWebServer();
    else {
        if( g_bTimeSync < 3 ) getTimeFromTimeServer();
        else count = 0;
        if( count == 5 ) count = -1; }
count++;
} else if( g_bTimeSync == 4 ){
    g_bTimeSync = 3; getInfoFromWebServer(); }
if( g_bTimeSync == 3 && tiCurTime.day !=
beforeDay&&tiCurTime.sec+tiCurTime.min*60 >=
randNum ){
    if( beforeDay != 0 ) getInfoFromWebServer();
    beforeDay = tiCurTime.day; deleteOldFile(); }
}

[Step 5] Receive Packet From PLC Lonwork
// Implementation of thread for lock, wait, and unlock
while( !g_bEnd ){
    n=read(Serial_fd,&recvPacket[recvSize],PLC_PACKET_SIZE);
    recvSize += n;
    while( recvSize >= PLC_PACKET_SIZE )
        recvSize = decodePacketFromPlc(recvPacket, recvSize);
}
/* Collect PLC Data & Send Current Data */
while( !g_bEnd ){
// Implementation of thread for lock, wait, and unlock

if( g_bTimeSync == 3 && g_bDcsIPSync != 0 &&
g_bPlcSync == 1 ) // check state
{ if((tiCurTime.totalSec-lastSendTime)> curDataCycle) {
    sendCurDataToDataServer();
    lastSendTime = tiCurTime.totalSec; } }
} // make packet (generate CRC)

/* Send Average Data & Send Data to DCS */
while( !g_bEnd ){
// Implementation of thread for lock, wait, and unlock
if( g_bDcsIPSync != 0 && g_numAddAvrDataToFile +
g_numRemainAvrData > 0 ){
    sendAvrDataToDataServer();
    lastSendTime = tiCurTime.totalSec; }
sleep(10); }
// make packet file (generate CRC)

```

그림 3. LACC 알고리즘
Fig. 3. LACC algorithm.

IV. 실험방법 및 성능평가

1. 실험방법

LACC의 모든 기능을 실험하기 위해서는 로컬 환경에서의 ESS 환경이 필수적이다. 본 논문이 제안한 방법을 검증하기 위하여 리눅스 환경의 PC 또는 Notebook, Windows에서 구현되었으며 추가적인 장비로는 Multi-ICE(JTAG), Power supply, Spectrum Analyzer, Frequency counter, Network analyzer 등이 사용되었다. 또한, PLC 상태와 PLC 노드의 입.출력을 실험환경에 맞도록 PLC 대신 시뮬레이션 프로그램이 사용되었다.

LACC 시스템의 완전한 구성과 성능분석을 위해서 여러 가지 부품들이 필요하며 각 부품들의 정상적인 조립과 기능을 확인하기 위한 검증 평가가 필요하다.

2. 분석 및 성능 평가

본 논문에서 사용될 LACC 성능 평가는 개발 내용의 적합성(LACC functional test), 기계적인 내성(LACC mechanical test), 시스템 동작 환경(LACC environment test) 등 다양한 각도에서 평가되도록 하였으며 일반 실내로 제한하였다. PCB에는 컴포넌트 면과 솔더 면에 부품이 실장되어 있으며 제작된 LACC 보드에서 성능평가를 수행한다. 또한, LACC는 PLC와 함께 사용되면서 액추에이터를 제어할 수 있는 기능을 가지고 있다. 따라서 LACC 명령어, PLC 명령어, 서버, DCS 등의 각 요소별 기능을 평가, 검증하고 그 결과를 보임으로써 효율 높은 에너지 절감 시스템의 구현이 가능함을 나타내고자한다.

가. LACC mechanical test

기구적인 평가를 위한 표준 조건은 주변의 온도가 25±5℃ 이며 상대적인 습도는 10~85%이다. 또한, 주파수의 범위는 8~200헤르츠(Hz)로 한다.

나. LACC environment test

(1)Zero temperature expose and operation test

샘플은 0±4℃ 실내에서 유지되고 최소 2시간 동안 축적되었다.

(2)High temperature test

샘플은 +70±4℃ 실내에서 유지되고 최소 2시간 동안 축적되었다.

(3)High temperature and high humid expose test

샘플은 +60±4℃, 85~90%RH 실내에서 유지되고 최소 2시간 동안 축적되었다.

(4)High temperature and high humid operation test

샘플은 +60±4℃, 90~95%RH 실내에서 유지되고 최소 2시간 동안 축적되었다.

(5)Temperature cycling test

샘플은 그림 4와 같이 5사이클로 이루어지고 최소 2시간 동안 실내조건에서 축적되었다.

(6)Humidity test (composite humidity)

샘플은 그림 5와 같이 10사이클로 이루어지고 최소 2시간 동안 일반 실내조건에서 축적되었다.

다. Typical Test

LACC의 부품별 정상적인 조립과 기능을 확인하기 위해 테스트 폼에 몇 가지 시험이 요구되었으며 그 결과는 그림 6과 같다.

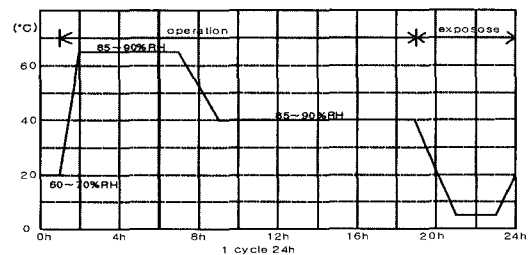


그림 4. 온도 주기 평가

Fig. 4. Temperature cycling test.

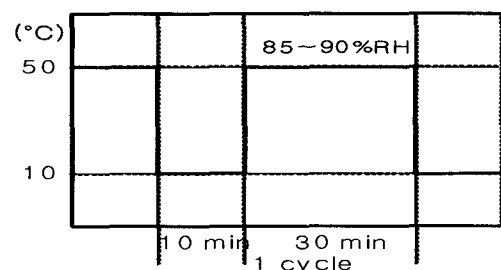


그림 5. 습도 평가

Fig. 5. Humidity test.

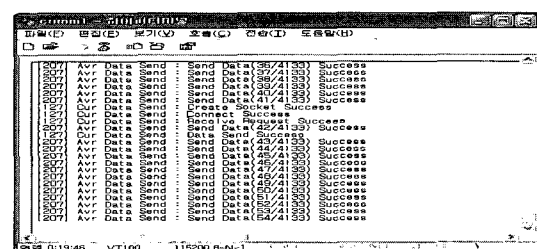


그림 6. 전형적 평가

Fig. 6. Typical Test.

(1)LED는 전원인가에서 LACC 실행까지 모든 LED 상태를 확인하고 덤스위치와 직렬 EEPROM은 부트 로더 메시지를 확인한다.

(2)전원공급기의 전류는 1A 이하이면 정상이다.

(3)RS-232C는 인스톨 이후 PLC 제어가 이루어지면 정상, 이더넷 포터는 인스톨이 되면 정상이다.

(4)LACC의 순차적인 동작 확인은 전원인가에서 실행까지 시간을 기록하고 1분 미만이면 정상이다.

(5)RTC는 웹(시간)서버에 접속하면 시간과 동기화가 되고 전원을 끄고 1분후 전원을 넣어서 RTC시간 및 배터리 전압을 기록, 서버와 같은 시간이면 정상이다.

라. LACC command checking on Console mode
콘솔모드의 LACC 명령어를 검사한다. 설정을 변경할 필요가 있거나 수동으로 PLC를 제어할 필요가 있을 경우, 콘솔을 통해 동작모드에서 구성 모드로 동작시킬 수가 있다.

(1)동작모드(Active mode)

그림 7은 동작모드에서 명령어를 실행한 화면이다. 입력 메시지의 반향 없이 LACC의 상태를 출력하는 것을 확인할 수 있었다.

(2)구성모드(Configuration mode)

그림 8은 구성 모드에서 명령어1("config")로 현재 사이클 값을 1분 10초로 확인하고 명령어2("cycle")로 2분 20초로 수정하여 실행한 결과이다.

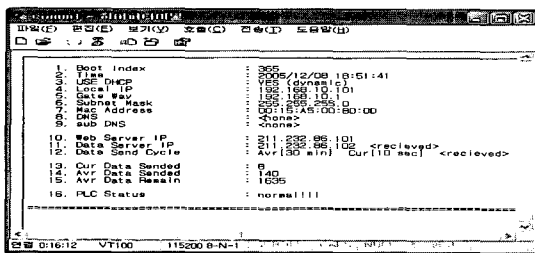


그림 7. LACC_status 명령어 실행결과 상태
Fig. 7. execution result status of LACC_status command.

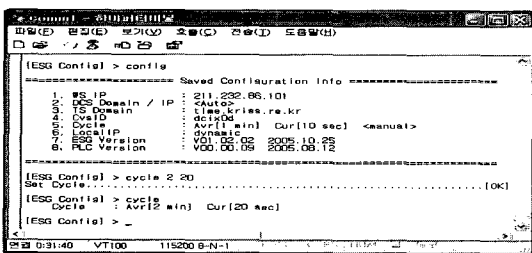


그림 8. config/cycle 명령어의 실행결과 상태
Fig. 8. execution result status of config/cycle command.

마. PLC command checking on Console mode
그림 9는 구성 모드에서 PLC 시뮬레이션 프로그램의 상태를 확인한 것이다.

바. Server Test on Active mode

(1)IP(DHCP) discover checking

LACC의 로컬 IP를 동적인 환경으로 설정 했을 경우 IP를 정상적으로 받을 수 있는지 확인한다. 또, 자동적으로 허용시간을 갱신하는지도 함께 확인해야 한다. 부팅 시에 콘솔에서 확인할 수 있으며 LACC의 응용 프로그램이 실행됨과 동시에 IP가 자동으로 수신되고 있음을 그림 10과 같이 확인할 수 있었다.

(2)Time, Date, Cycle, IP synchronous

LACC는 웹(시간) 서버 프로그램에 접속하면 시간, 사이클, DCS 서버의 IP를 수신 받는다. LACC가 접속 후 서버와 동기화 되었는지 확인한다. 그림 11은 LACC를 실행한 후의 웹 서버 접속결과 상태를 타나낸다.

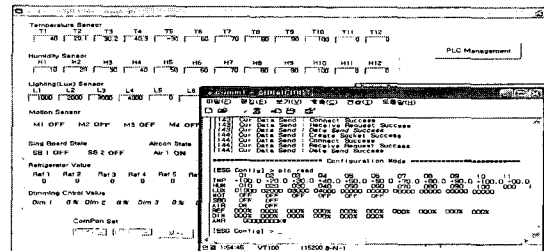


그림 9. PLC시뮬레이션 프로그램 상태
Fig. 9. Status of PLC simulation program.

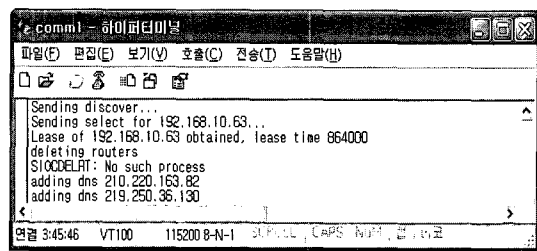


그림 10. DHCP 실행 및 IP자동 수신결과 상태
Fig. 10. execution of DHCP and status of IP automatic receiving result.

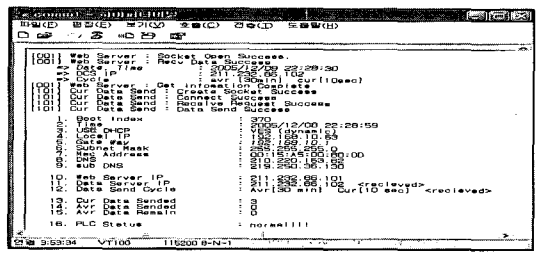


그림 11. 웹 서버 프로그램 접속 결과 상태
Fig. 11. Connection result status of web server program.

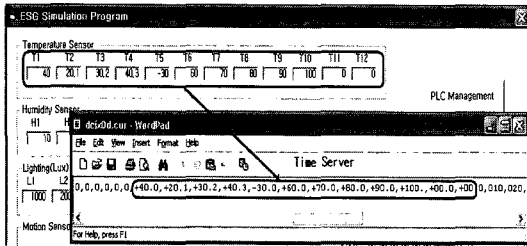


그림 12. DCS에 전송된 현재데이터 정보
Fig. 12. Information of current data sending to DCS.

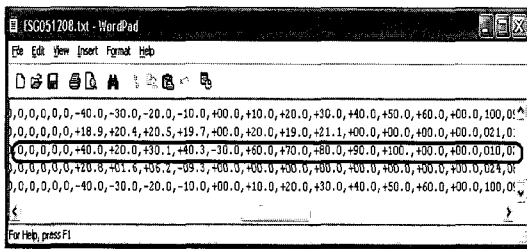


그림 13. DCS에 전송된 평균데이터 정보
Fig. 13. Information of average data sending to DCS.

(3)Current data sending to DCS

PLC의 현재 데이터가 서버 쪽으로 사이클에 맞도록 정확히 전송되고 있는지 확인한다. 그림 12에서 보이는 것처럼 PLC 시뮬레이션 데이터와 DCS의 현재데이터가 일치하였다.

(4)Average data sending to DCS

PLC의 평균데이터가 서버 쪽으로 사이클에 맞도록 전송되고 있는지 확인한다. 평균데이터의 검증은 LACC의 기능 중에서 변수적인 요소가 강하기 때문에 별도로 검증을 수행해야 한다. 그림 13과 같이 PLC 시뮬레이션 데이터와 DCS의 평균데이터가 일치하였다.

사. PLC Test on Active mode

(1)Fault time out checking

PLC가 이상이 생겼을 경우 오류가 발생하게 되며, 이 신호가 LACC에서 검출하는데 소요되는 시간을 확

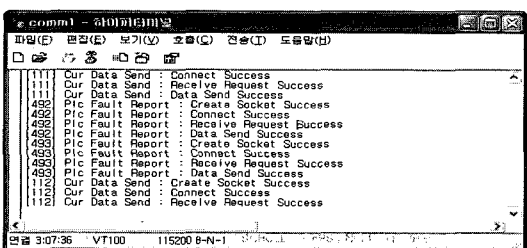


그림 14. PLC 오류 발생 및 LACC의 검출상태
Fig. 14. Occurrence of PLC error and detection status of LACC.

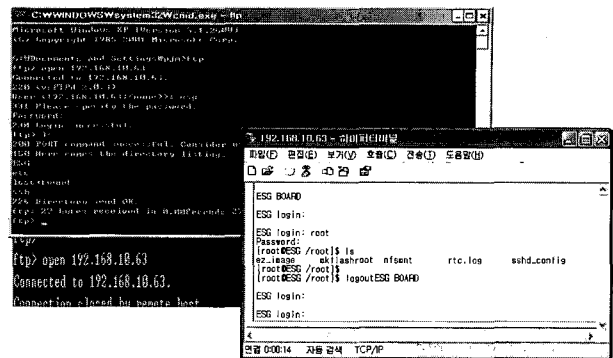


그림 15. Ftp/Telnet/FireWall 서비스
Fig. 15. Ftp/Telnet/FireWall service.

인한다. 현재 설정은 5분이다. PLC의 동작을 정지 시킨 후 약 5 분 후에 그림 14와 같이 오류가 나타났다.

아. Network Service

LACC는 그림 15와 같이 웹 브라우저에서 PLC의 상태를 표시하는 웹, 원격으로 LACC로 접속하기 위한 Ftp 클라이언트 프로그램, 터미널에 접속하여 LACC를 제어하기 위한 텔넷 또는 보안셸(SSH), 그리고 방화벽 기능을 확인하는 방화벽(Fire Wall) 서비스 등을 제공한다.

V. 결 론

이번에 제안한 ESS의 설계 및 개발 사례는 네트워크 상에서 제어를 하는 시스템에서 일어 날 수밖에 없는 치명적인 문제를 비교적 저렴하고, 작은 노력을 들여 구축할 수 있는 가장 효율적인 방법이라 생각한다. 따라서 네트워크 망이 확대됨에 따라서 보안 문제와 기기 식별 문제가 이슈로 등장할 것으로 보이며, 이 연구를 바탕으로 하여 앞으로 생겨 날 문제에 대한 새로운 접근 방법이 계속 연구되어야 한다. 추가적으로 터널, 공장 같은 제어하기 힘든 곳이나 위험 지역에 설치됨으로서 원격관리에 의한 인력 절감효과, 수리 보수비용 절감 등 산업기반의 시설물에서 그 활용도가 매우 뛰어나다고 할 수 있겠다.

참 고 문 헌

[1] Alex Lennon, "Embedding Linux," IEE Review, pp. 33-37, May 2001.
[2] Victor Yodaiken, "The RT-Linux approach to hard realtime," <http://www.rtlinux.org>
[3] Rajesh Gupta, "Guest editorial: Special issue on

networked embedded systems," ACM Trans, Vol. 3, No. 1, pp. 1-2, 2004.

[4] H. C. Ferreira, H. M. Grove, O. Hooijen, and A. J. Han Vinck, "Power Line Communication: Overview," IEEE 4th AFRICON'96, Stellenbosch, Vol. 2, pp. 558-563, Sept. 1996.

[5] Linux Kernel Archive : <http://www.kernel.org/pub/linux/kernel/v2.4>

[6] Sheridan, Thomas B., "supervisory Control," Gavriel Salvendy(Ed.), John Wiley&Sons, New York : Handbook of Human Factors and Ergonomics(Third Edition), 2006.

[7] William R.Hamel, "Fundamental Issues in Telerobotics," proceedings of IEEE-RAS HUMANOIDS, pp. 66, 2003.

[8] Cha, dong-Hyuk, "Shared control of telerobot systems using artificial intelligence algorithms," 한국 과학기술원 박사 학위 논문, 1995.

[9] J. Wang, Binoy Ravindran, "Time-utility switched Ethernet: packet scheduling algorithm, implementation, and feasibility analysis," IEEE Trans on PDS, Vol. 15, No. 2, pp. 119-133, 2004.

저 자 소 개



박 태 진(정회원)-교신저자
 1988년 동의대학교 물리학과 학사 졸업.
 1995년 부경대학교 전자계산학과 석사 졸업.
 2002년 부경대학교 전자계산학과 박사 수료.
 2000년 3월~2003년 2월 거제대학 전자계산학과 초빙전임교수
 2005년 9월~현재 마산대학 지능형 홈 디자인 학과 교수
 <주관심분야 : 영상압축, 영상처리, 멀티미디어 공학, 임베디드 시스템>



정 승 렬(정회원)
 1981년 경북대학교 전자공학과 학사 졸업.
 1988년 경상대학교 물리학과 석사 졸업.
 2003년 경남대학교 응용물리학과 박사 졸업.
 1982년 3월~1991년 2월 경상대학교 실험실습 조교
 1991년 3월~현재 마산대학 지능형 홈 디자인 학과 교수
 <주관심분야 : 정밀계측, 신호처리>

장 명 기(학생회원)
 2006년 한국과학기술원 전자공학과 학사 졸업.
 2005년 대덕 하이테크 클럽 총무
 2006년~현재 ENTUS 대표
 <주관심분야 : 유비쿼터스, 임베디드 시스템>