

디젤 발전기를 위한 원격 임베디드 컨트롤러 개발

김용환^o 김영백 최명실 안광선

경북대학교 컴퓨터공학과

hypnus@korea.com^o realtech@hanafos.com choisil@hotmail.com gsahn@knu.ac.kr

Remote Embedded Controller Development for Diesel Generator

Yonghwan Kim^o, Youngback Kim, Myoungsil Choi, Gwangseon Ahn
Department of computer Engineering, KyungPook University

요 약

임베디드 시스템은 마이크로프로세서와 소프트웨어 기술의 빠른 성장에 따른 소형화, 경량화에 힘입어 각종 분야에서 널리 사용되고 있다. 본 연구는 임베디드 시스템을 이용하여 원격 제어, 실시간 모니터링을 지원하는 디지털 제어 방식의 원격 임베디드 컨트롤러를 개발한다. 개발된 컨트롤러는 기기 운용 과정에서 발생할 수 있는 각종 에러를 단문 메시지 서비스(SMS)를 이용해 전송하는 장점이 있다. 이는 기초 산업 분야에서 임베디드 시스템 운용의 가능성을 보임으로써 산업 전반에 걸친 활용 모델이 될 것이다.

1. 서 론

디젤 발전기는 소규모 건물이나 지역에 전원이 차단되더라도 지속적으로 전원을 공급하기 위한 전력 발전기이다. 특히 태풍, 수해 등의 천재지변, 예기치 않은 정전사고 등이 발생할 경우 전력을 공급해주기 위하여 사용되고 있다. 병원과 같은 전력 공급의 차단이 치명적인 결과를 초래하는 건물이나 지역의 경우 이러한 디젤 발전기는 필수적이라 할 것이다. 디젤 발전기는 최근까지도 아날로그 방식을 채택하고 있다. 그러나 아날로그는 데이터의 저장과 전달 시에 왜곡과 잡음의 발생으로 정확한 데이터를 전달할 수 없다. 현재 정확한 데이터의 전달과 비용의 절감을 위해 디지털 방식의 컨트롤러를 사용하고 있다.

디지털 컨트롤러는 마이크로프로세서만을 사용해 왔다. 그로 인해 특정 데이터의 전달이나 자체적인 문제의 해결성을 갖지 못했다. 이러한 문제점은 디젤 발전기에서 어떤 문제가 발생하는 경우 관리자가 방문하지 않는다면 이를 알지 못하거나 해결할 수 없다는 것이다. 본 논문에서는 이러한 문제점을 개선하기 위해 고성능 마이크로프로세서(32bit Microprocessor)와 임베디드 운영체제를 탑재한 임베디드 컨트롤러를 제안한다.

임베디드 컨트롤러는 인터넷과 연결되어 원격지에서 인터넷이 가능하다면 언제나 제어와 모니터링이 가능하다. 또한 인터넷을 사용하지 못하더라도 발전기에서 발생한 알람 메시지를 휴대폰을 통해 전달 받는다. 그러므로 본 논문에서 제안하는 임베디드 컨트롤러는 보다 편리하고 향상된 기능을 통해 디젤 발전기를 운용하는 것이 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 전체 시스템의 구성에 대해 설명하고 3장에서 원격 웹 서버 구축이나 통신, 긴급 메시지를 전달하기 위한 단문 메시지 서비스 등의 시스템 운용에 필요한 모듈에 대해 개괄적 설명한다. 4장에서 결론을 제시함으로써 본 논문을 맺는다.

2. 전체 시스템 구성

제안하는 시스템은 로컬 임베디드 컨트롤러와 중앙 서버 2가지로 구성된다. 로컬 임베디드 컨트롤러는 디젤 발전기를 제어하고, 중앙 서버로 데이터를 전달하기 위해 인터넷 망을 사용한다. 중앙 통제 서버는 로컬 임베디드 컨트롤러에서 들어오는 자료를 저장하고 분석한다. 그림 1은 임베디드 컨트롤러가 데이터를 중앙 서버로 전송하고, 중앙 서버가 전송된 데이터를 MySQL DB에 저장하는 것을 그림으로 나타내었다. 이후 관리자는 웹 서버에 접속하여 MySQL DB에 저장된 데이터를 웹 브라우저를 통해 모니터링할 수 있다. 만일 로컬 임베디드 컨트롤러가 긴급 상황을 알리는 메시지를 중앙 서버에 전달하면 TCP/IP 서버 데몬(Server daemon)은 이를 판단하고 관리자에게 여러 메시지를 전송한다[1].

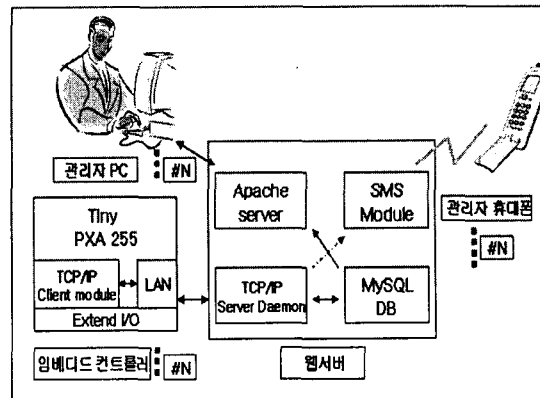


그림 1. 전체 시스템 구성도

※ 본 연구는 대구광역시와 (재)대구테크노파크에서 시행한 차세대선도산업기술 연구개발사업의 수행결과임

3. 시스템 동작

3.1 임베디드 컨트롤러 설계

개발된 임베디드 컨트롤러는 Intel Xscale PXA255 (400Mhz) 마이크로프로세서를 사용하고, 운영체제는 임베디드 리눅스를 포함하였다[2]. 그림 2의 임베디드 컨트롤러에서 설계한 확장 I/O의 제어를 위해 컨트롤러의 어드레스 디코더에서 남는 핀을 확장 I/O와 연결하였다. 설계한 확장 I/O 보드는 디젤 발전기의 아날로그 신호를 8채널의 ADC 0809(A/D 컨버터)를 이용하여 디지털 데이터로 변환하고, 이 변환된 데이터를 임베디드 컨트롤러에서 처리한다.

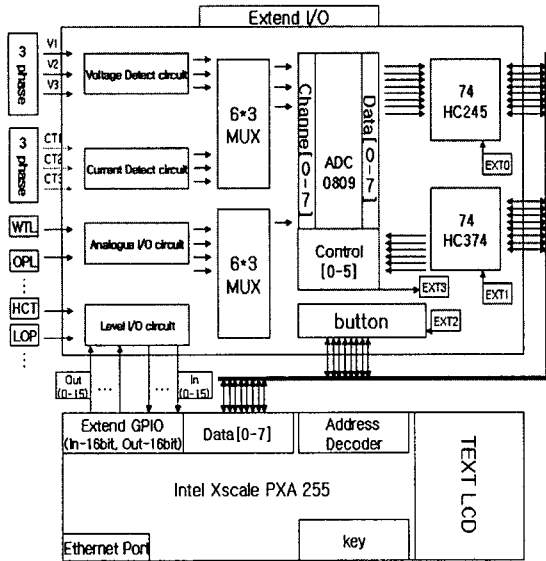


그림 2. 디젤 발전기 임베디드 컨트롤러

표 1은 응용 프로그램에서 확장 I/O에 접근하기 위한 메모리 맵을 나타낸다.

표 1. Extend I/O 메모리 맵

GPIO(9)	주파수
0xf17000000(Virtual)	Text LCD
0xf16000000(Virtual)	내부 LED
0xf15000000(Virtual)	Primary key
0xf14000000(Virtual)	Extend key
0xf13000000(Virtual)	Extend GPIO Out
0xf12000000(Virtual)	Extend GPIO In
0xf11000000(Virtual)	Adc Control Out
0xf10000000(Virtual)	Adc Data In

3.2 응용 프로그램 구현

소프트웨어 모듈은 긴급하게 처리해야 할 부분을 인터럽트나 폴링을 이용해 리얼타임으로 컨트롤러를 제어한다. 응용 프로그램은 C 언어로 작성되었으며 이 프로그램은 함수 초기화, 각 장치 디바이스 드라이버 open, 시그널 등록 및 I/O 포트 초기화, 확장 I/O에서 수집된 데이터 계산, Text LCD 디스플레이, 에러 처리, 입력 장치 처리 등의 작업을 한다.

```
int main()
{
    /* 함수 초기화 */
    /* 각 장치 디바이스 드라이버 open */
    /* 시그널 등록 */
    /* I/O 포트 초기화 */

    do{
        gotgauge();
        if(mode==disp) display(sbmd);
        if(mode!=errp) errread();
        if(err.total) errproc();
        if(pkey) keyproc(pkey);
        if(ekey) button(ekey);
        usleep(100000);
    }while(1);
    /* 각 장치 디바이스 드라이버 close */
    return 0;
}
```

3.3 통신 모듈 구현

TCP/IP 프로토콜은 응용 계층에서의 E-Mail 전송, FTP 프로토콜을 이용한 파일 전송, HTTP 프로토콜을 이용한 웹 페이지 표현 등의 응용 계층의 프로토콜들을 사용하는데 적합하다[3]. 따라서 본 논문에서는 임베디드 컨트롤러와 웹 서버 간에 TCP/IP 프로토콜을 사용한다.

응용 프로그램에서는 디젤 발전기 운용과정에서 수집한 각종 데이터와 정상 범위를 벗어난 에러를 로그 파일로 저장한다. 동시에 TCP/IP 클라이언트 모듈은 소켓을 생성하고, 웹 서버와 연결을 시도한다. 그리고 로그 파일을 읽어서 소켓을 통해 웹 서버의 TCP/IP 서버 데몬으로 전송한다.

아래의 TCP/IP 서버 데몬은 select() 함수를 사용하여 다중 TCP/IP 클라이언트들과 통신을 할 수 있도록 한다[4]. TCP/IP 클라이언트 모듈의 연결 요청이 있을 경우 요청을 처리하고, TCP/IP 클라이언트가 데이터를 전송한 경우 read() 함수에서 데이터를 읽어서, message라는 변수에 저장하고 파싱한 후 MySQL DB에 저장한다.

```
if(--fd_max < fd_max+1, &temps, 0, 0, &timeout))
    exit(1);
for(fd=0; fd < fd_max+1; fd++){
    if(FD_ISSET(fd, &temps)){
        if(fd==serv_sock){ /* 연결 요청인 경우 */
            clnt_len = sizeof(clnt_addr);
            clnt_sock = accept(serv_sock,
                (struct sockaddr*)&clnt_addr,
                &clnt_len);
            FD_SET(clnt_sock, &reads);
            if(fd_max < clnt_sock)
                fd_max = clnt_sock;
        }else{
            str_len = read(fd, message, sizeof(message));
            message[str_len] = '\0';
            if(str_len == 0) /* 연결 종료 요청인 경우 */
                FD_CLR(fd, &reads);
                close(fd);
            }else /* 데이터를 받은 경우 */
                write(fd, message, str_len);
                strcpy(strque, "insert into woorig values (0,0,");
                if(lisalNum(message[str_len-1]))
                    message[str_len-1] = '\0';
                strcpy(strque+strlen(strque), message);
                strcpy(strque+strlen(strque), ",0)");
                mysql_query(&mysql, strque);
        }
    }
}
```

3.4 원격 웹 서버 구축

웹 서버의 운영체제는 리눅스이며 웹 서버를 구축하기 위해 APM(Apache+PHP+MySQL) 패키지를 설치하였다. 표 2는 TCP/IP 클라이언트에서 전송된 데이터를 저장하기 위한 MySQL DB 에러 테이블이다. 이 데이터들은 위에서 언급한 바와 같이 각

각의 데이터 종류 별로 파싱되어 해당되는 테이블에 저장된다. 테이블에서 SQ는 각 TCP/IP 클라이언트 모듈들에서 데이터가 전송될 때마다 자동 증가되는 카운터 넘버이며 중복되지 않는다.

표 2. MySQL DB 에러 테이블

Field	Type	Null	Key	비고
SQ	int		PRI	카운터 넘버
ID	int			각 Controller 고유 ID
ER	text			에러 내용
ET	int	YES		에러 레벨

그리고 아래의 PHP 코드는 MySQL DB 테이블에 TCP/IP 클라이언트 모듈로부터 전송 받은 새로운 SQ가 있는지를 검사하고, 새로운 SQ가 있다면 에러의 내용을 그림 3의 웹 페이지에 보여준다[5]. 그림 3의 각종 수치에 대한 데이터 값들은 수치 테이블에 저장된다.

```
try{
    $mysql = new safe_mysql('localhost', 'root', '', 'test');
    $stmt = $mysql->prepare(
        "SELECT * from woorig where sq > ?");
    $sequence = 0;
    while(true){
        $stmt->bind_param('i',$sequence);
        $stmt->execute();
        $stmt->bind_result($sq,$id,$er,$et);
        while($stmt->fetch()){
            $sequence = (int)$sq;
            echo "$sq. $er<br>";
        }
        sleep(1);
    }
    $stmt->free_result();
} catch(SQLException $e){
    die("SQL Error : ". $e->getSQLError().
        " in <hr><PRE>". $e->getSQL(). "</PRE><hr>");
} catch(DBException $e){
    die("Database Error : ". $e->GetMessage());
} catch(Exception $e){
    die(exception_dump($e));
}
```

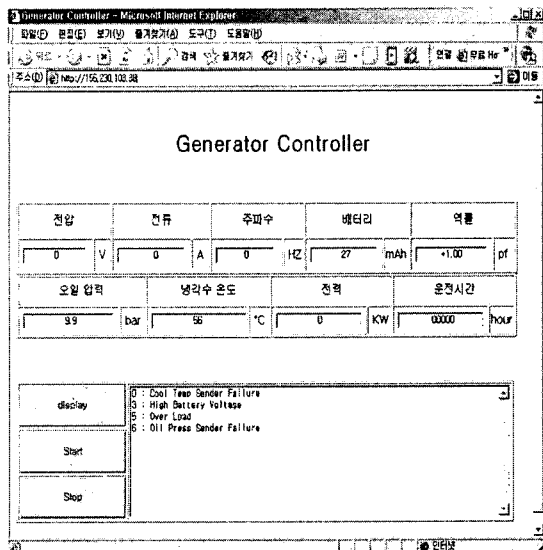


그림 3. 디젤 발전기 원격 제어 및 모니터링 웹 페이지

3.5 단문 메시지 서비스

단문 메시지 서비스는 어떤 클라이언트에서 다른 모바일 클라이언트로 문자 메시지를 전달할 수 있는 서비스이다. 본 논문에서는 임베디드 컨트롤러에서 긴급한 오류가 발생된 경우 이를 중앙 서버에 전달하도록 하였다. 긴급 오류 메시지를 받은 중앙 서버의 TCP/IP

서버 대문은 이를 슈어엠에서 제공하는 문자 메시지 서비스 모듈을 통해 외부 관리자에게 전달한다[6].

슈어엠에서 제공하는 단문 메시지 서비스의 동작 방식은 일반 소켓 프로그램 방식과 유사하다. 우선 모듈은 단문 메시지 서비스 서버에 접속하고, sendto() 함수로 에러 데이터를 서버에 전송하고, 단문 메시지 서비스 서버에서는 관리자에게 단문 메시지를 전송한다.

```
/* Connect the socket to the specified server */
if(connect(sd, (struct sockaddr *)&sad, sizeof(sad)) < 0){
    fprintf(stderr, "connect failed\n");
    exit(1);
}
if(sendto(sd, (char*)&AcsPkt, PACKET_SIZE, 0,
    (struct sockaddr *)&sad, sizeof(sad)) < 0)
    perror("Sending datagram message");
if(read(sd, (char*)&AcsPkt, (PACKET_SIZE)) < 0){
    perror("Error receiving data from server");
    exit(1);
}
if(AcsPkt.szStatus == 'O') {
    /* sending success */
} else {
    /* sending fail */
}
```

4. 결론

본 논문에서는 효율적으로 디젤 발전기를 제어할 수 있는 방법을 제안하였다. 제안하는 시스템은 기존의 아날로그형 디젤 발전기에 장착 되어 실시간 모니터링과 갑작스런 상황에 대한 대처 능력을 갖는다. 제안하는 원격 임베디드 컨트롤러의 장점은 다음과 같다.

첫째, 웹 브라우저의 사용으로 별도의 새로운 관리 프로그램을 설치하지 않아도 된다. 둘째, 디지털 방식의 채택으로 데이터의 저장과 분석이 용이하다. 셋째, 긴급한 데이터의 전달이 가능하다. 마지막으로 제안된 시스템은 특정 기기에 의존하지 않기 때문에 다양한 분야에 적용이 가능하다.

그러므로 본 논문에서 제안하는 임베디드 컨트롤러는 기존의 제안된 방법이 갖는 장점을 수용하면서, 유용성이 추가된 방법이다.

5. 참고 문헌

- [1] Jen-Hao. T. Chin-Yuan. T. Yu-Hung. C, "Integration of networked embedded systems into power equipment remote control and monitoring", 2004 IEEE Region 10 Conference Volume C, vol 3, pp. 566-569, Nov. 2004
- [2] (주)한백전자, "HBE-EMPOS Tiny User's Manual ver 1.0"
- [3] M. T. Jones, "Embedded with the mailman", Embedded Systems Programming, pp. 45-52, Oct. 2001.
- [4] 윤성우, "tcp/ip 소켓 프로그래밍", 프리렉, pp. 284-292, 2004
- [5] 민신현, "Advanced PHP for windows", 대림, pp. 166-171, 2003
- [6] 슈어엠, <http://www.surem.co.kr>