

2007년 2월
석사학위 논문

로드 셀 계측용 원격 모니터링 시스템 개발

조선대학교 대학원
제어계측공학과
김 경 석

로드 셀 계측용 원격 모니터링 시스템 개발

Remote monitoring system development for load cell
measuring

2007년 2월 23일

조선대학교대학원

제 어 계 측 공 학 과

김 경 석

로드 셀 계측용 원격 모니터링 시스템 개발

지도교수 장 순 석

이 논문을 공학 석사학위신청 논문으로 제출함

2006년 10월

조선대학교대학원

제 어 계 측 공 학 과

김 경 석

김경석의 석사학위 논문을 인준함

위원장 조 선 대학교 교수 강 성 권

위원 조 선 대학교 교수 김 준 섭

위원 조 선 대학교 교수 장 순 석

2006년 11월

조선대학교 대학원

목차

ABSTRACT

I. 서론

II. 방법

A. 센서와 센서 디바이스

1. 센서
2. 센서 디바이스
 - a. Power supply
 - b. Sensor
 - c. Amplifier
 - d. A/D convertor
 - e. micro controller

B. 통신 모듈

1. Bluetooth wireless transceiver module
2. Ethernet module

C. 인터넷 호스트 PC 및 관리 프로그램

1. controller programming
2. 인터넷 호스트 PC 및 관리 프로그램

III. 결론

참고문헌

부록

ABSTRACT

Remote monitoring system development for load cell measuring

Kim Kyung-suk

Advisor : Prof. Jang Soon-suck Ph.D.

Department of Control and Instrumentation Eng,

Graduate School of Chosun University

In the present work a weight measuring instrumentation system in which data conversion and a series of signal processing were totally equipped were designed. The 16 load cell were incoming sensors and each output of the strain gauge was amplified and filtered for proper analog signal processing. Several measuring instrumentation OP amps and general purposed OP amps were used. The 12 bits A/D converters converted analog signals to digital bits, and a PIC microprocessor controlled the 16 channels of strain gauges. RF RS232 modules were used for wireless communication between the PIC microprocessor and an ethernet host for a remote sensor monitoring system development.

I . 서론

유비쿼터스를 구체적으로 실현하는 연구는 이미 매우 진보된 상태로 진행 중이다. 산업 설비의 원격 계측 및 자동화뿐만 아니라 가정이나 사무실, 백화점 등에서 사용하는 정보가전기기에도 이제는 인터넷과 RF에 의한 원격 제어를 위한 센서 디바이스 및 임베디드 모듈이 옵션이 아닌 필수 품목으로 장착되고 있다. 이 같은 유비쿼터스를 향한 기술적 시도가 더욱 성공하기 위해선 부가되는 모듈들이 더욱 경량화되고 소형화되어야 할 필요성을 가지며 이를 위해 SoC(System-on-Chip)이라는 반도체 설계 기술이 센서 및 RF 기술과 더불어 융합되며 발전하고 있다. 특정 목적 혹은 다중 목적의 다양한 SoC 칩이 개발되면 이를 응용하는 유비쿼터스 지향 모듈이나 극소형 시스템 개발이 더욱 활발해질 것이다.

유비쿼터스 실현을 위해서는 센서 디바이스 및 임베디드 시스템, RF 통신, 인터넷 네트워크, 인터넷 서비스들을 모두 하나로 묶기 위한 각 개별 모듈 간의 상호 연결이 불가피해 진다. 특히 독립된 임의의 마이크로프로세서 임베디드 시스템과 인터넷 네트워크와의 교신이 가장 중요한 기술적 관건이 될 수 있는데, 최근에 더욱 발전되고 있는 무선통신의 방법들은 이러한 요구에 따른 움직임으로 볼 수 있다. 아울러 시리얼 통신 규격과 인터넷 통신 규격 사이의 신호 변환 전송을 위한 변환 스위치에 IP 주소가 할당되면서 그 자체가 호스트로서의 역할을 하게 된다. 이로써 각종 계측기기 등을 포함하여 직렬 통신 포트를 가지고 있는 디바이스는 인터넷을 통한 원격 제어 및 모니터링을 가능하게 한다. 즉, 유선과 무선 등의 여러 데이터 통신의 발달과 이를 제어할 수 있는 컨트롤러 또는 프로세서의 발전 등은 유비쿼터스 지향의 대중화를 더욱 가속화할 수 있게 한다.

본 논문은 유비쿼터스 기술 분야 중 센서 임베디드 디바이스와 RF 통신 그리고 인터넷 네트워크(Ethernet)에 의한 원격 계측(Telemetry) 시스템 개발에 관한 것이다. 센서로는 16 채널의 로드 셀(load cell)을 적용하였고, 각 채널 별로 아날로그 신호를 증폭, 오프셋, 필터링하여 12 비트 A/D 변환시켜 주는 센서 디바이스를 제작하고, 16 채널로부터 동기화된 다중 신호를 입력 시키고 센서 디바이스를 제어하는 PIC 마이크로컨트롤러를 중심으로 한 임베디드 시스템을 제작하였다. 약 50~100m의 거리 간격을 가지는 센서 디바이스 임베디드 시스템과 인터넷 네트워크 스위치와의 교신을 위해 RF 무선 통신 시스템을 개입시켰다. RS232 시리얼 통신과 인터넷 네트워크 TCP 통신 간의 규격 맞춤을 위해 변환 장치를 사용하였는데, 이 변환 장치 내부에 독립된 IP 주소를 갖추도록 하여 인터넷에 의한 원격 접속이 가능하도록 하였다. 인터넷 호스트에서는 C# 언어로 원격 계측 시스템 구동에 필요한 접속 프로그램을 제작하였으며, 이로써 데이터 베이스 및 인터넷 서비스와도 연동이 되는 장점을 가지도록 하였다.

II 장 방법 부분에서는 위에 언급한 각각의 개별 시스템에 대해 상세히 기술하고 III 장 결과 부분에서는 인터넷을 경유한 원격 계측의 정확도와 여러 단계의 매개체를 통한 통신 과정에서 발생할 수 있는 전송 에러와 이를 극복하는 소프트웨어 알고리즘을 논하고자 한다.

II. 방법

그림 1 은 본 논문에서 개발된 시스템의 전체적인 아웃라인을 보여준다. 전체 시스템은 크게 센서디바이스 모듈, 마이크로프로세서 모듈, 인터넷 호스트 및 관리 프로그램 이렇게 세 부분으로 나눌 수 있다.

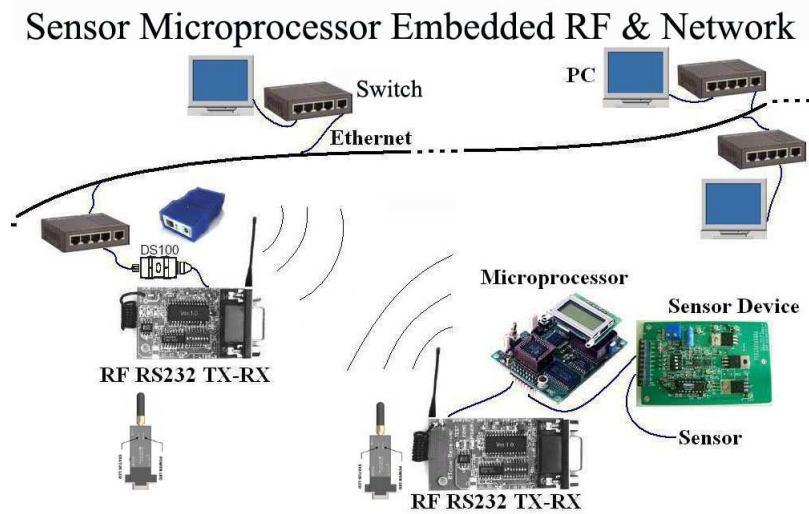


그림 1. 센서디바이스-Bluetooth 통신모듈-인터넷 네트워크 시스템의 전체 윤곽도.

A. 센서와 센서 디바이스

본 시스템은 16 채널의 로드 셀(load cell)로부터 하중의 변화량을 정량적으로 수집하는 인터넷 경유 원격 계측 시스템 개발이다. 이는 앞으로 유비쿼터스 시대에 요구되는 센서네트워크 시스템의 구현에 앞서 산업현장의 특수한 환경에 대한 적용이라고 할 수 있다.

1. 센서

본 시스템에서 쓰이는 센서는 로드 셀(load cell)이다. 로드 셀은 하중을 전기 신호로 변환하는 방법에서 가장 많이 사용하고 있는 스트레인 게이지를 이용한 것이다. 스트레인 게이지와 피측정물을 일체화하여 로드 셀이란 하중 변환기가 상품화 되었다.



그림 2. Single point load cell(왼쪽), S-type load cell(오른쪽).

로드셀의 출력은 일반적으로 브리지 회로에 가해지는 전압 1V당 출력 전압 (mV/V)으로 표시한다. 이 출력 전압은 브리지에 인가되는 전압을 V_i , 브리지의 출력 전압을 V_o , 브리지가 받는 등가 변형을 ε , 스트레인 게이지의 변형을 모두 동일하게 k 로 하면 다음과 같이 된다.

$$\text{전압 출력} \quad \frac{V_o}{V_i} = \frac{1}{4} k \cdot \varepsilon$$

일반적으로 로드 셀의 사양을 살펴보면 정밀급은 정격 출력 $3\text{mV/V} \pm 0.1\%$ 이며, 보통급은 $2\text{mV/V} \pm 0.1\%$ 즉, 실제 계측된 데이터를 수집하는 데에는 데이터의 적절한 조절이 필요하다.

2. 센서 디바이스

센서 디바이스 부분은 센서로부터의 입력 즉, 미미한 아날로그 신호를 적절히 증폭하고 실제 우리가 인지하고 가공할 수 있도록 데이터를 디지털로 변환하는 역할을 한다.

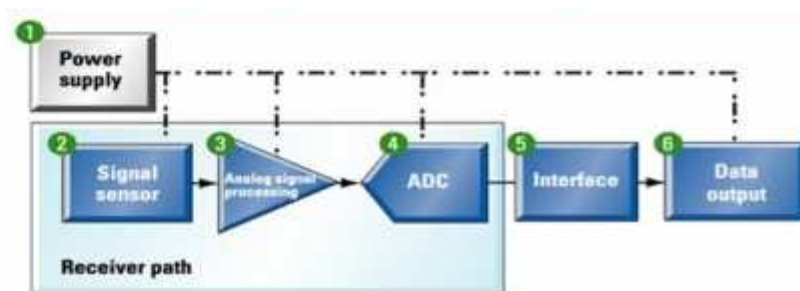


그림 3. 일반적인 신호 패스 시스템 블록 다이어그램

센서디바이스의 구성을 블록 다이어그램(그림 3)을 통해 살펴보면 다음과 같다.

a. Power supply (①)

전원을 공급하는 역할을 하는 부분으로서 LDO(Low Dropout) 레귤레이터를 사용하여 각 부품들의 동작에 필요한 전원을 공급하고 있다.

b. Sensor (②)

로드 셀(load cell)을 사용하여 하중의 변화에 따른 출력을 낸다.

c. Amplifier (③)

센서로부터의 데이터를 적절히 조절하는 부분이다. 이곳에서는 계측용 증폭기를 사용하고 있으며, 오프셋 조절기, 버퍼용 증폭기 등을 차례로 구성하여 신호를 증폭, 필터링하고 A/D 변환기로 전달한다.

d. A/D convertor (④)

아날로그 신호 데이터를 수집 및 가공이 용이한 디지털 신호로 변환한다. 본 논문에서는 12bit의 분해능을 가지며 100kSPS의 처리능력을 가진 ADC를 사용하였다.

e. micro controller

본 계측기기의 제어는 PIC16F877 마이크로컨트롤러를 내장한 PBM-R5 모듈을 사용하고 있다. 마이크로컨트롤러 모듈은 AD 변환기의 동작 제어와 데이터의 관리 전달의 역할을 한다. 즉, AD변환기로부터 전달받은 데이터를 RS232 직렬 통신 포트를 이용하여 데이터를 컴퓨터 또는 무선 통신장비를 통해 다른 호스트로 전달한다.

참고로 센서디바이스 도면 (그림 4), 센서디바이스 제어 모듈 도면 (그림 5), 센서디바이스 모듈 PCB, 제어 모듈 PCB (그림 6), 센서디바이스 모듈, 제어 모듈 (그림 7)을 각각 아래에 나타내었다.

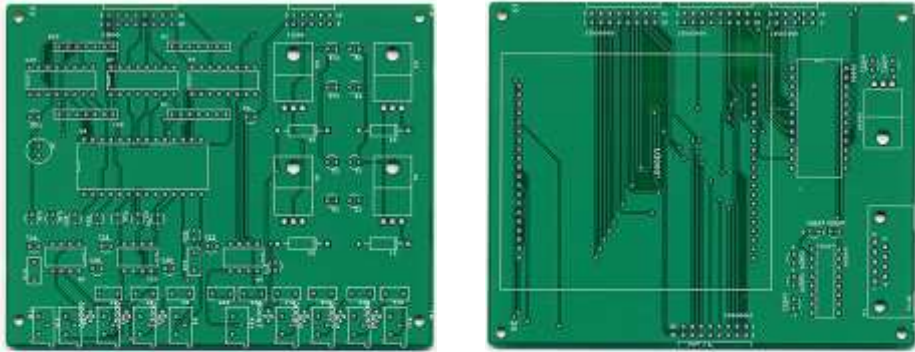


그림 6. 센서디바이스 모듈 PCB(왼쪽), 제어 모듈 PCB(오른쪽)

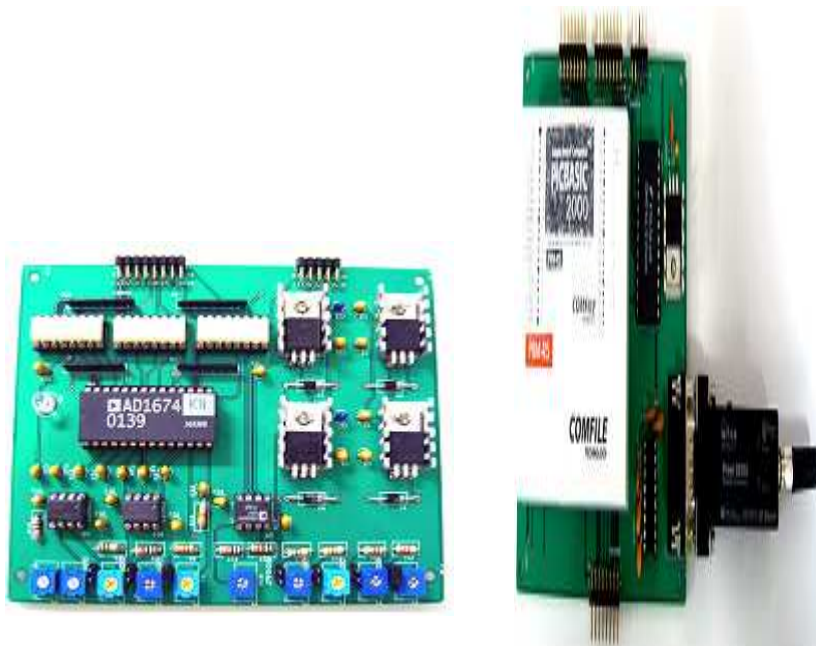


그림 7. 센서디바이스 모듈(왼쪽), 제어 모듈(오른쪽)

B. 통신 모듈

유비쿼터스 센서 네트워크는 각기 독립된 센서와 컨트롤러가 서로 정보를 수집하고 이를 전달하여 사람들이 변화를 인지하고 대처하도록 하는 시스템이라고 할 수 있겠다. 이러한 네트워크 시스템의 중심은 바로 통신의 방법이라고 할 수 있다. 본 연구에서는 그 대상을 공장 등과 같은 산업현장으로 정하고 보다 정확하고 안정화된 데이터 전달 방법을 필요로 통신 방법을 선택하게 되었다.

1. Bluetooth wireless transceiver module

블루투스(Bluetooth)는 IEEE 802.15.1 PANs(Personal Area Networks)의 산업 표준이다. 블루투스의 큰 특징은 다양한 기기들이 안전하고 저렴한 비용으로 전세계적으로 이용이 가능한 라디오 주파수를 이용해 서로 통신할 수 있게 한다는 것이다 (그림 8 참조). ISM 대역인 2.45GHz를 사용하며 버전 1.1과 1.2의 경우 속도가 723.1kbit/s에 달한다. 버전 2.0의 경우 EDR을 특징으로 하는데, 이를 통해 2.1 Mbit/s의 속도가 가능하다. 블루투스는 유선의 USB를 대체하는 개념이며, 암호화에는 SAFER(Secure And Fast Encryption Routine)을 사용한다. 즉, 무선이면서 고속이고 뛰어난 보안성을 지니고 있기 때문에 더욱 주목 받고 있는 기술이다.

컨트롤러를 통해 전달되는 데이터는 RS-232 직렬 포트 또는 USB를 통하여 블루투스 무선 통신 모듈로 전달 원거리의 다른 통신 모듈까지 약 500m 거리 내에서 전달 가능하다.

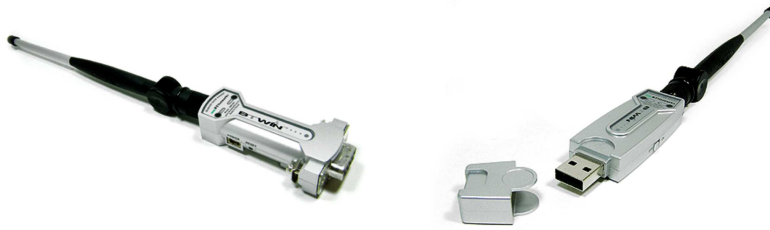


그림 8. 블루투스 무선 통신 모듈

2. Ethernet module

블루투스 무선 통신을 이용해 전달된 데이터는 인터넷을 통해 호스트 PC로 전달 저장된다.

Ethernet을 이용한 인터넷 사용 방법에는 여러 가지가 있을 수 있으나 간편하게 앞서 사용되었던 블루투스와 맞물려 동작할 수 있는 serial to ethernet module을 사용하였다.

이 serial to ethernet module은 이름 그대로 RS-232 포트를 사용하고 있는 모든 장치 디바이스에 별도의 IP를 부여하여 하나의 서버로서 동작할 수 있게 한다(그림 9 참조).

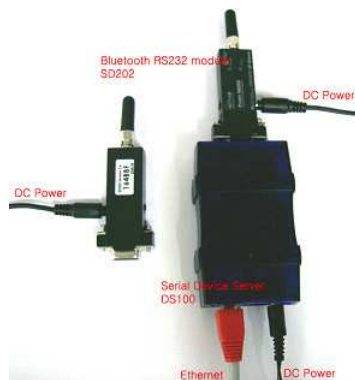


그림 9. Bluetooth RS-232 module과 Serial device server

C. 인터넷 호스트 PC 및 관리 프로그램

고유의 IP를 가지고 서버로서의 역할을 수행 가능하게 된 시스템으로부터 이제는 데이터를 요구하고 저장 관리하는 명령 수행 및 데이터 전달 방법에 관한 프로그램이 필요하다.

본 연구에서 요구되는 프로그램은 크게 두 개로 나눌 수 있다. 첫째는 직렬 디바이스로부터 전달 받은 데이터를 관리하고 다시 통신 모듈 또는 PC로 전달하는 역할을 하는 컨트롤러의 프로그램이다. 둘째는 직렬장치서버로의 데이터를 요구하고 전달받아 저장하고 보여주는 프로그램이 그것이다.

1. Controller programming

컨트롤러가 수행하는 역할은 센서로부터 계속해서 들어오는 신호를 AD컨버터가 변환 작업을 수행하도록 하는 명령과 변환된 데이터를 손실 없이 받아 다시 PC또는 통신모듈을 통해 별도의 서버 등으로 전달하는 역할을 수행해야 한다.

컨트롤러는 4개의 I/O 출력 핀을 이용하여 4*16 디코더로 A/D컨버터를 선택 순차적으로 데이터를 전달 받게 된다. 아울러 A/D변환기가 데이터를 변화할 수 있는 동기 신호를 주기적인 펄스를 통해 내보내야 한다.

A/D변환기의 데이터 출력은 byte 단위로 PIC 마이크로컨트롤러의 2개의 포트를 통해 들어오며 이는 다시 RS232 송수신 핀을 통해 Bluetooth module로 데이터를 전송한다.

2. 인터넷 호스트 PC 및 관리 프로그램

사용자는 언제 어디서나 데이터 관리 프로그램을 설치하고 로그인 하여 데이터를 확인하고 저장할 수 있어야 한다.

인터넷 호스트에서는 C# 언어로 원격 계측 시스템 구동에 필요한 접속 프로그램을 제작하였으며, 이로써 데이터베이스 및 인터넷 서비스와도 연동이 되는 장점을 가지도록 하였다.

블루투스 모듈과 인터넷망을 거치는 동안에 센서 임베디드 시스템과 원격 소프트웨어 PC와는 시간 동기화가 불가능하다. 따라서 호스트 PC에서 센서 임베디드 시스템에 매번 접속하고 접속을 종료할 때까지 상호 시스템간의 Handshaking 교신이 절대적으로 필요해 진다(그림 10 참조). 특히 RS-232 통신은 1 바이트를 기준으로 하므로 12비트의 디지털 데이터 전송을 위해서 2 바이트의 시리얼 전송이 불가피해지는데 각 1 바이트 전송 시간 사이에서 비동기화에 따른 데이터 손실이 발생할 수 있으며 이는 실제로 매우 심각한 문제가 된다, 따라서 Handshaking 교신은 매우 중요한 소프트웨어적 해결 알고리즘이라 할 수 있겠다.

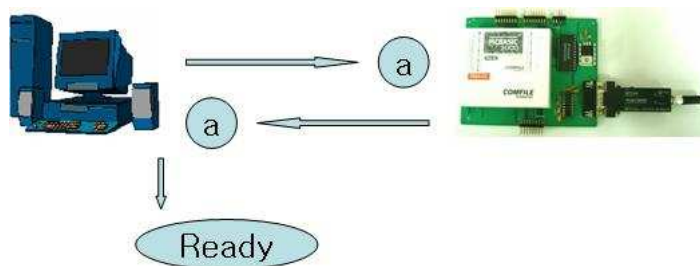


그림 10. Handshaking 알고리즘

Handshaking은 먼저 데이터를 요구하기 전에 컨트롤러의 응답과 AD컨버

터의 동작에 대한 응답을 요청하는 것이다.

그림 11 은 호스트 PC 화면에 보여지는 원격 계측 프로그램의 구동 상태이다. 인터넷을 경유하여 로드 셀의 하중 변화량을 수집하고 있으며 이들 자료들은 데이터베이스에 연동되어 계속 실시간으로 저장되고 외부의 사용자들이 인터넷으로 자료를 볼 수 있도록 프로그램을 고안 작성하였다.

이 프로그램은 채널을 입력하는 곳에다 1, 2, ..., 16번 중 원하는 채널의 번호를 입력하고 입력이 끝난 상태에서 선택을 누르면 그 결과 값이 Output : 에디터박스에 출력이 나타난다 (단, 999는 전체 채널의 결과 값들을 읽어와 오른쪽 All Channel에 1번부터 16번 포트 까지 연속으로 출력(그림 12)).

ID	VALUE
----	-------

Output : 1953

채널 8 선택

DB reading DB writing Clear

All Channel

그림 11. 한 개 채널의 데이터 요구 출력

ID	VALUE
8번 채널	1959
9번 채널	4095

Output :

채널

그림 13. DB를 통한 데이터 저장

III. 결론

본 연구에서는 로드 셀 원격 모니터링 시스템 개발이라는 목표로 로드 셀 계측에 적합한 시그널 컨디셔닝을 하는 센서디바이스와 이를 제어하는 컨트롤러를 개발하고 블루투스 무선 통신 그리고 인터넷 네트워크를 적용하여 원격 계측 시스템을 개발하였다.

아울러 무선 통신에 따른 신호 전송상의 문제점을 발견하고 이를 극복하고자, 소프트웨어적인 알고리즘을 추구하여 신호 전송 에러의 문제점을 해소하였다.

또한 본 연구에서 얻은 결과를 계속해서 가전정보기기에 응용하여 로드 셀이 아닌 가정정보기기의 디스플레이부나 제어부와 원격 계측 제어에 활용하는 시스템 개발에 적용하고자 한다.

참고 문헌

- [1] Sang-Chul Park, “Level up OrCAD,” Sungandang, Jan, 2002.
- [2] Chi-Ho Lee, “OrCAD 9.0 theory and circuit drawing method,” Ilji nsa, Jan, 2001.
- [3] Yong-Bum Cho, “OrCAD ver10.0,” Bukdu Press, Aug. 2004.
- [4] Sung-Kwang Ha, “ASP.NET 2003,” Darim Press, Aug, 2003.
- [5] Sang-Jo Sung, “C# Web Programing actual fighting guide,” Devpia, June, 2003.
- [6] Tae-Il Kim, “Visual C#.NET Programing,” Devpia, Oct, 2001.
- [7] http://www.analog.com/UploadedFiles/Data_Sheets/8976538-54AD620_g.pdf
- [8] http://www.analog.com/UploadedFiles/Data_Sheets/3466691-45AD1674_c.pdf
- [9] <http://icc.skku.ac.kr/%7Erfic/image/lecture/laboratory2/data/-HA17741.pdf>
- [10] <http://www.comfile.co.kr/download/pb/PBMAN10.pdf>
- [11] http://www.initium.co.kr/bizdata/userguide/Promi-SD사용설명서_V2.0.pdf
- [12] <http://docs.tibbo.com/index.html?ds100.htm>

부록

A. Controller code

```
CONST DEVICE = R5  
SET RS232 38400
```

```
DIM KH AS BYTE           'High bit of I  
DIM KH_BUF AS BYTE      'High bit of I for save  
DIM KL AS BYTE           'Low bit of I  
DIM KK AS BYTE
```

```
KH = 0
```

```
KL = 0
```

```
KH_BUF = 0
```

```
KK = 0
```

```
ON RECV GOSUB 100
```

```
10          OUT 29, 0  
           GOTO 10
```

```
100         GET KK  
           IF KK = 97 THEN      'About a value for select
```

```

                                KH = 0
                                KL = 0
                                KH_BUF = 0

                                OUT 29, 0
                                OUT 12, 1
                                OUT 13, 1
                                OUT 31, 1
                                OUT 30, 0

                                DELAY 1

                                OUT 29, 1
                                DELAY 1

                                KL = BYTEIN(2)
                                KH = BYTEIN(3)

                                OUT 29, 0

                                DELAY 1

                                ' 상위 4비트 백업
                                KH_BUF      =      KH      AND
&B00001111

                                PUT KL

                                END IF

                                .
                                .

```



```

IF          KK          =          90          THEN
          'About Z value of High bit
PUT KH_BUF
END IF

```

```

RETURN

```

B. C# code

```

using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

using System.Data.Odbc;

using System.Net;
using System.Net.Sockets;
using System.Text;

namespace WindowsApplication6
{

```

```

/// <summary>
/// Form1에 대한 요약 설명입니다.
/// </summary>

public class Form1 : System.Windows.Forms.Form
{
    static void Main()
    {
        Application.Run(new Form1());
    }

    private void button1_Click(object sender, System.EventArgs e)
    {
        listView1.Items.Clear();

        string source = "DRIVER={MySQL ODBC 3.51 Driver};" +

        "SERVER=220.67.220.150;" +

        "DATABASE=test;" +
        "UID=root;" +
        "PASSWORD=;" +
        "OPTION=3";

        // 데이터베이스 연결
        OdbcConnection MyConnection = new OdbcConnection(source);
        MyConnection.Open();
    }
}

```

```

OdbcDataAdapter adapter = new OdbcDataAdapter();
DataSet ds = new DataSet();

string sql = "SELECT * FROM embedded";
adapter.SelectCommand = new OdbcCommand( sql, MyConnection );
adapter.Fill( ds );

DataTable dt = ds.Tables[0];

foreach (DataRow row in dt.Rows)
{
    ListViewItem item = new ListViewItem( row[0].ToString() );

    item.SubItems.Add( row[1].ToString() );

    listView1.Items.Add( item );
}

MyConnection.Close();
}

private void button3_Click(object sender, System.EventArgs e)
{
    byte [] data = new byte[1024];

    IPEndPoint ipep = new

```

```

IPEndPoint(IPAddress.Parse("220.67.220.143"), 1001);
    Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram, ProtocolType.Udp);

    try
    {
        if(textBox1.Text=="1")
        {
            data = Encoding.Default.GetBytes("a");

            server.SendTo(data, data.Length, SocketFlags.None, ipep);

            IPEndPoint sender1 = new IPEndPoint(IPAddress.Any, 0);
            EndPoint remote = (EndPoint)(sender1);

            data = new byte[1024];

            int recv_size = server.ReceiveFrom(data, ref remote);
            int rst = data[0];

            data = Encoding.Default.GetBytes("Z");
            server.SendTo(data, data.Length, SocketFlags.None, ipep);

            recv_size = server.ReceiveFrom(data, ref remote);

            rst = rst + data[0] * 256;

```

```
label2.Text="";
label2.Text=Convert.ToString(rst);
}

else

//label2.Text=Convert.ToString(rst_sum[1]);
}
else
{
label2.Text="Selection of Invalid Port";
textBox1.Text="";
}

server.Close();
}

catch
{
server.Close();
}
finally
{
server.Close();
}
}
```

```

private void button2_Click(object sender, System.EventArgs e)
{
    string source = "DRIVER={MySQL ODBC 3.51 Driver};" +
        "SERVER=220.67.220.150;" +
        "DATABASE=test;" +
        "UID=root;" +
        "PASSWORD=;" +
        "OPTION=3";

    OdbcConnection MyConnection = new OdbcConnection(source);
    MyConnection.Open();

    OdbcDataAdapter adapter = new OdbcDataAdapter();
    DataSet ds = new DataSet();

    //string sql = "SELECT * FROM embedded";
    string sql1 = "insert into embedded(name, input)
values('"+textBox1.Text+"번 채널"+", '"+label2.Text+"')";
    adapter.SelectCommand = new OdbcCommand( sql1, MyConnection
);

    adapter.Fill( ds );

    MyConnection.Close();
}

private void button4_Click(object sender, System.EventArgs e)

```

```
{  
label5.Text="";  
label6.Text="";  
label7.Text="";  
label8.Text="";  
label9.Text="";  
label10.Text="";  
label11.Text="";  
label12.Text="";  
label13.Text="";  
label14.Text="";  
label15.Text="";  
label16.Text="";  
label17.Text="";  
label18.Text="";  
label19.Text="";  
label20.Text="";  
  
}
```

```
private void Form1_Load(object sender, System.EventArgs e)  
{  
  
}  
}  
}
```

저작물 이용 허락서

학 과	제어계측공	학 번	20057128	과 정	석사
성 명	한글: 김 경 석 한문 : 金 庚 奭 영문 : Kim Kyung-Suk				
주 소	광주광역시 남구 방림2동 65-1				
연락처	E-MAIL : kyungsukkim@paran.com				
논문제목	로드 셀 계측용 원격 모니터링 시스템 개발				
	Remote monitoring system development for load cell measuring				

본인이 저작한 위의 저작물에 대하여 다음과 같은 조건아래 조선대학교가 저작물을 이용할 수 있도록 허락하고 동의합니다.

- 다 음 -

1. 저작물의 DB구축 및 인터넷을 포함한 정보통신망에의 공개를 위한 저작물의 복제, 기억장치에의 저장, 전송 등을 허락함
2. 위의 목적을 위하여 필요한 범위 내에서의 편집·형식상의 변경을 허락함.
다만, 저작물의 내용변경은 금지함.
3. 배포·전송된 저작물의 영리적 목적을 위한 복제, 저장, 전송 등은 금지함.
4. 저작물에 대한 이용기간은 5년으로 하고, 기간종료 3개월 이내에 별도의 의사 표시가 없을 경우에는 저작물의 이용기간을 계속 연장함.
5. 해당 저작물의 저작권을 타인에게 양도하거나 또는 출판을 허락을 하였을 경우에는 1개월 이내에 대학에 이를 통보함.
6. 조선대학교는 저작물의 이용허락 이후 해당 저작물로 인하여 발생하는 타인에 의한 권리 침해에 대하여 일체의 법적 책임을 지지 않음
7. 소속대학의 협정기관에 저작물의 제공 및 인터넷 등 정보통신망을 이용한 저작물의 전송·출력을 허락함.

동의여부 : 동의() 반대(○)

2007년 2월 23 일

저작자: 김 경 석

(서명 또는 인)

조선대학교 총장 귀하