

# 유비쿼터스 지향의 16 채널 스트레인 게이지 계측 시스템 개발 16 Channel Strain Gauge Measuring Ubiquitous System Development

장순석\*, 김경석, 원용일, 김대곤  
(Soon-Suck Jarnng, Kyung-Suk Kim, Yong-ill Won, and Dae-Gon Kim)

**Abstract :** A strain gauge weight measuring instrumentation system was designed with RF sensor network facilities. In the sensor module system, data conversion and a series of signal processing were totally equipped. 16 strain gauges are 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. 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.

**Keywords :** strain gauge, instrumentation amplifier, opAmp, sample and holder, analog to digital converter, micro controller, RS-232, RF

## I. 서론

유비쿼터스를 구체적으로 실현하는 연구는 이미 매우 진보된 상태로 진행 중이다. 산업 설비의 원격 계측 및 자동화 뿐만 아니라, 가정이나 사무실, 백화점 등에서 사용하는 정보 가전기기도 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.

센서 디바이스 및 임베디드 시스템, RF 통신, 인터넷 네트워크, 인터넷 서비스들을 모두 하나로 묶기 위해서는 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 개발된 시리얼 통신 규격과 인터넷 통신 규격 사이의 신호 변환 전송을 위한 변환 스위치에 IP 주소가 할당되면서 그 자체로써 호스트로서의 역할을 하게 되었다. 이로써 각종 전자제품의 유비쿼터스 지향의 대중화는 더욱 가속화될 수 있게 되었다.

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 스트레인 게이지를 적용하였고, 각 채널 별로 아날로그 신호를 증폭, 오프셋, 필터링하여 12 비트 A/D 변환시켜주는 센서 디바이스를 제작하고, 16 채널로부터 동기화된 다중 신

호를 입력시키고 센서 디바이스를 제어하는 PIC 마이크로컨트롤러를 중심으로 한 임베디드 시스템을 제작하였다. 약 50m~100m의 거리 간격을 가지는 센서 디바이스 임베디드 시스템과 인터넷 네트워크 스위치와의 교신을 위해 RF 무선 통신 시스템을 개입시켰다. RS232 시리얼 통신과 인터넷 네트워크 UDP 통신간의 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.

다음으로 방법에서는 위에 언급한 각각의 개별 시스템에 대해 상세히 기술하고 결과에서는 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 1은 본 논문에서 개발된 센서 디바이스 임베디드 RF 인터넷 네트워크 시스템의 전체적인 윤곽도를 보여준다. 전체 시스템은 크게 세 부분으로 나눌 수 있다.

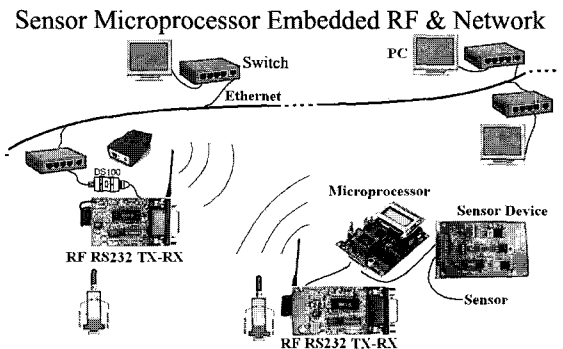


그림 1. 센서-임베디드-Bluetooth module-인터넷 네트워크 시스템의 전체 윤곽도.

Fig. 1. Sensor-Embedded-Bluetooth module-Ethernet total system outline.

\* 책임저자(Corresponding Author)

논문접수 : 2005. 10. 28., 채택확정 : 2006. 6. 8.

장순석, 김경석, 원용일, 김대곤 : 조선대학교 정보통신공학부 (ssjarnng@chosun.ac.kr/kyungsukkim@paran.com/ywon@nambu.ac.kr/lovee@nate.com)

※ 본 연구는 산업자원부 2005년 RIS 사업비 지원에 의해 연구되었음.

1. 센서 디바이스-마이크로프로세서 임베디드 시스템
2. RF RS232 통신 모듈 및 시리얼 Ethernet 변환 모듈
3. 인터넷 호스트 PC 및 C# 소프트웨어 프로그래밍

1. 센서 디바이스-마이크로프로세서 임베디드 시스템

본 시스템의 목표는 16 채널의 스트레인 게이지로부터 하중의 변화량을 정량적으로 수집하는 인터넷 경유 원격 계측 시스템 개발이다. 데이터 수집 부분은 크게 두 개로 나뉘어진다. 각 채널에 스트레인 게이지 센서가 연결되어 있고 (그림 2(a))에 묘사된 회로도 와 같이 센서의 미소한 아날로그 신호를 증폭하는 instrumentation 전치 증폭기, 오프셋 조절기, 버퍼, OP 증폭기들이 차례로 신호를 증폭, 필터링한 후 A/D 변환기를 통해 디지털 데이터를 출력하는 부분(그림 4(a))과 각 채널을 제어 출력된 데이터의 수집, 통신 모듈로의 데이터를 전송하는 컨트롤러 부분(그림 2(b), 그림 4(b))이다.

본 계측기기의 제어는 PIC16F877 마이크로컨트롤러를 내장한 PBM-R5 모듈을 사용하고 있다. PIC 마이크로컨트롤러 모듈과 AD 변환기를 통해 각 채널의 데이터 입력을 제어한다. AD사의 12 비트 A/D 변환기(부품 번호 AD1674)는 three-state 버퍼가 자체 내장되어 있다. 16 채널 중 데이터를 획득하고자 하는 채널은 A/D 변환기의 칩 선택 핀에 OV 디지털 신호를 주어 선택하도록 하였고, 각 채널 별 A/D 변환기마다 1개의 디지털 신호 회선만으로도 A/D 변환과 three-state 버퍼로의 12 비트 디지털 신호가 전송하게 된다. 따라서 채널 선택용과 A/D 변환용 2개의 디지털 신호 회선이 마이크로프로세서로부터 공급되어야 한다.

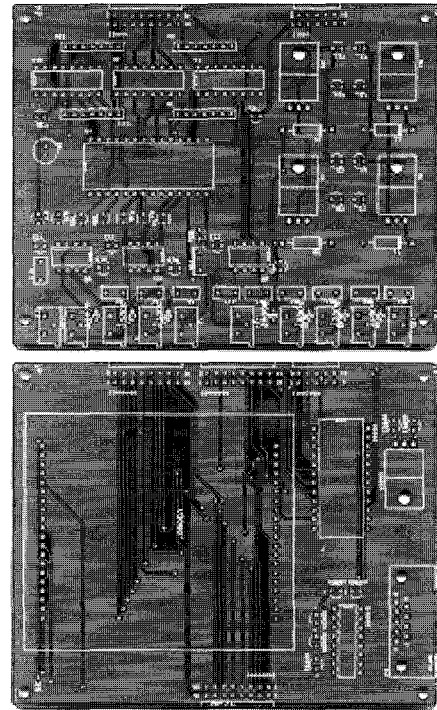
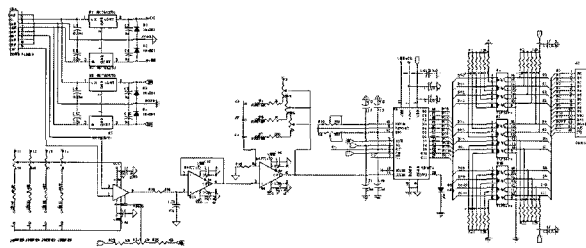
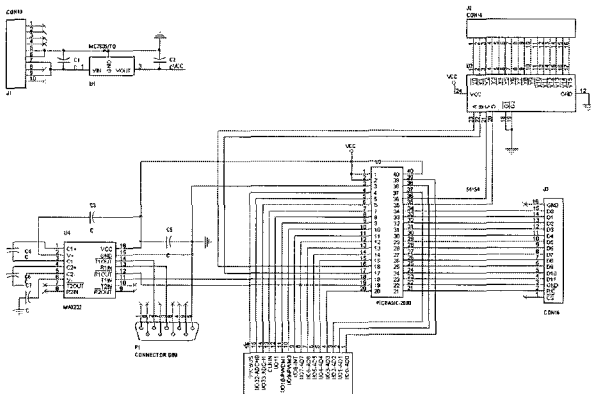


그림 3. (a) 스트레인 게이지 센서 신호 증폭, 필터링, A/D 변환 회로도, PCB 기판 (b) 16 채널 전체 제어 회로도, PCB 기판.

Fig. 3. (a) Strain gauge sensor signal amplification, filtering, A/D conversion circuit PCB (b) 16 channel total control circuit PCB.



(a)



(b)

그림 2. (a) 스트레인 게이지 센서 신호 증폭, 필터링, A/D 변환 회로도 (b) 16 채널 전체 제어 회로도.

Fig. 2. (a) Strain gauge sensor signal amplification, filtering, A/D conversion circuit (b) 16 channel total control circuit.

```
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=KHAND &B00001111

  PUT KL
END IF
.
.

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

RETURN
    
```

위의 PIC BASIC을 통해 A/D 변환을 위한 서브루틴과 인터넷 호스트와의 교신을 위한 서브루틴을 확인할 수 있다. A/D변환기의 선택은 PIC 마이크로컨트롤러의 4개의 출력 핀 으로부터 4\*16 디코더를 통해 입력을 받고자 하는 A/D변환기를 선택하게 되는 것이다. 그리고, 또 하나의 출력 핀을 통

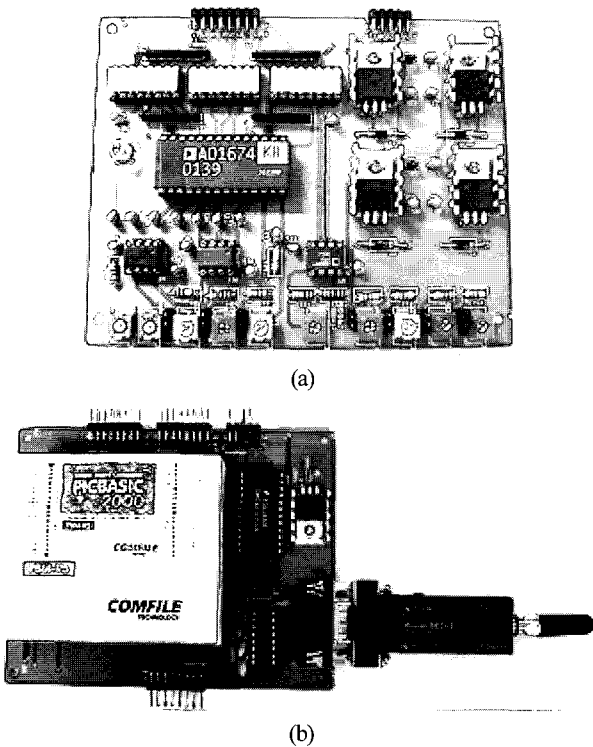


그림 4. (a) 1채널 센서 신호 증폭기, 필터링, A/D 변환기 (b) PIC16F877 마이크로컨트롤러를 내장한 PBM-R5 모듈, 디코더, bluetooth 송수신 모듈.

Fig. 4. (a) 1 channel sensor signal amplifier, filtering, A/D converter (b) PBM-R5 module with PIC16F877 microcontroller, decoder, bluetooth transceiver module.

해 계속해서 A/D변환기가 데이터를 읽고 변환할 수 있도록 주기적인 펄스를 내보낸다.

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

2. RF RS232 통신 모듈 및 시리얼-Ethernet 변환 모듈

센서 디바이스 임베디드 시스템과 인터넷망과의 거리가 약 50m~100m의 간격을 가진다고 하고 그 거리 사이의 통신을 위해 bluetooth RS232 무선 통신 모듈(SD202)을 개입시켰다(그림 5) 다음으로 RF RS232 시리얼 통신 출력은 계속해서 Ethernet과 연결되어야 하는데 이를 위해 시리얼-Ethernet 변환 모듈(모델 번호 DS100)을 사용하였다(그림 5). 시리얼 통신과 인터넷 네트워크 UDP 통신간의 규격 맞춤을 위해 이 같은 변환 모듈을 사용하였는데 이 변환 장치 내부에는 독립된 IP 주소를 갖추고 있어 그 자체로써 서버가 되는 것이며 이로써 인터넷에 의한 원격 접속이 가능하게 되었다.

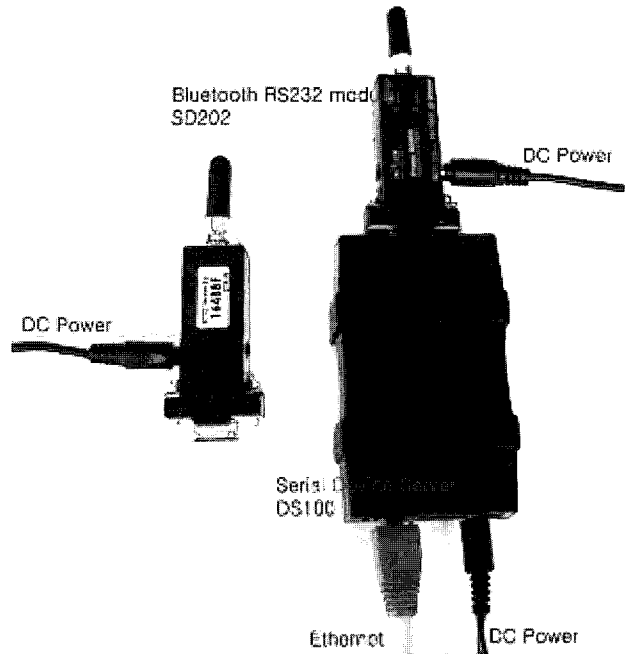


그림 5. 직렬통신 기기 서버 DS100.

Fig. 5. Serial device server DS100.

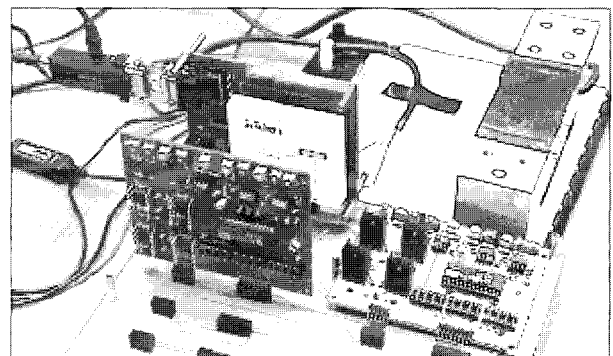


그림 6. 전체 스트레인 게이지 측정 시스템 사진도.

Fig. 6. Total strain gauge measurement system photograph.

## 3. 인터넷 호스트 PC 및 C# 소프트웨어 프로그래밍

인터넷 호스트에서는 C# 언어로 원격 계측 시스템 구동에 필요한 접속 프로그램을 제작하였으며 이로써 데이터 베이스 및 인터넷 서비스와도 연동이 되는 장점을 가지도록 하였다. 다음은 C#로 작성한 센서 임베디드 시스템 구동 프로그램의 주요부분을 보여준다.

```
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]; // 문자 한 개씩 전송

        IPEndPoint ipep = new IPEndPoint(IPAddress.Parse("220.67.220.143"), 1001); // 220.67.220.143는 PICBASIC 서버측 IP 주소임.
        Socket server = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp); // UDP 방식의 교신

        try
        {
            if(textBox1.Text=="1")
            {
                data = Encoding.Default.GetBytes("a"); // "a" 문자를 PICBASIC으로 전송
                server.SendTo(data, data.Length, SocketFlags.None, ipep);

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

                data = new byte[1024];

                int rcv_size = server.ReceiveFrom(data, ref remote); //PICBASIC에서 Low Byte 8 bits 전송받는 명령
                int rst = data[0];

                data= Encoding.Default.GetBytes("Z"); // "Z" 문자를 PICBASIC으로 전송
                server.SendTo(data, data.Length, SocketFlags.None, ipep); //PICBASIC에서 High Byte 4 bits 전송받는 명령
                rcv_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)
{
}
}

```

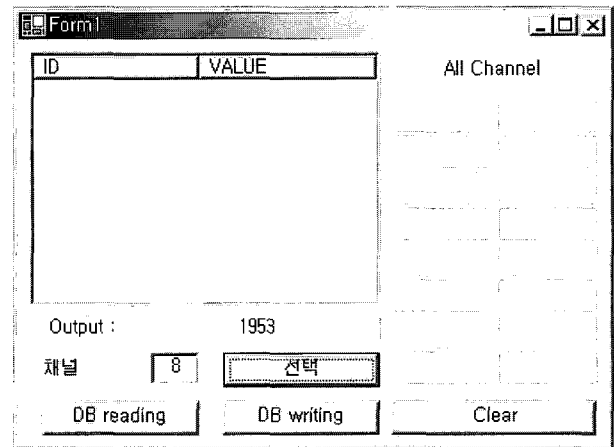
RF RS232와 인터넷망을 거치는 동안에 센서 임베디드 시스템과 원격 소프트웨어 PC와는 시간 동기화가 불가능하다. 따라서 호스트 PC에서 센서 임베디드 시스템에 매번 접속하고

접속을 종료할 때까지 상호 시스템간의 handshaking 교신이 절대적으로 필요해진다. 위의 프로그램은 C#로 작성한 handshaking 교신 프로그램의 일부를 보여준다. 특히 RS232 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 알고리즘이랄 수 있겠다.

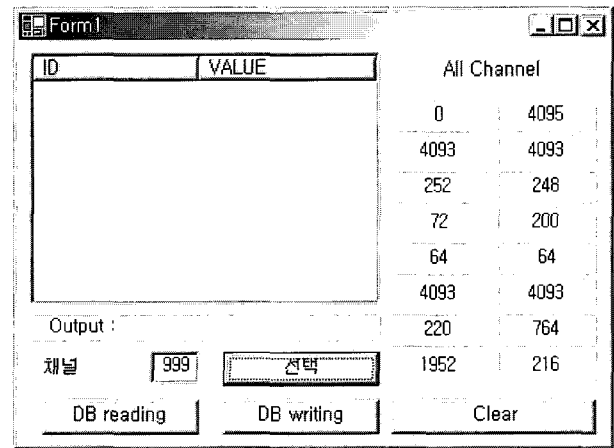
III. 결과

(그림 7)은 호스트 PC 화면에 보여지는 원격 계측 프로그램의 구동 상태이다. 인터넷을 경유하여 스트레인 게이지 센서의 하중 변화량을 수집하고 있으며 이들 자료들은 데이터 베이스에 연동되어 계속 실시간으로 저장되고 외부의 사용자들이 인터넷으로 자료를 볼 수 있도록 프로그램을 고안, 작성하였다.

이 프로그램의 구동 원리는 채널을 입력하는 곳에다 1, 2, ..., 16번 중 원하는 채널의 번호를 입력하고 입력이 끝난 상태에서 선택을 누르면 그 결과값이 output: 에디터박스에



(a)



(b)

그림 7. (a) 한 개 채널에서의 출력 (b) 전체 채널의 출력.

Fig. 7. (a) 1 channel output (b) All channel output.

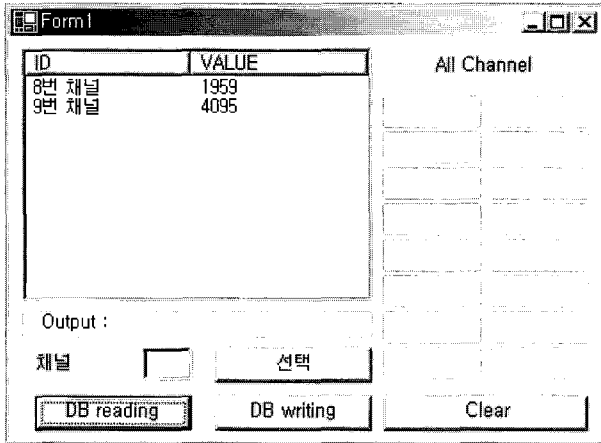


그림 8. DB를 통한 데이터 저장.  
Fig. 8. Save data through the database.

출력이 나타난다(그림 7(a)) (단, 999는 전체 채널의 결과값들을 읽어와서 오른쪽 All Channel에 1번부터 16번 포트 까지 연속으로 출력(그림 7(b))).

1부터 16번 중 각각의 선택된 채널의 정보는 PBM R5 PIC 마이크로컨트롤러 모듈에 정보를 요청을 하고, 전달 방법은 1, 2, ..., 16이라는 값이 PIC 마이크로컨트롤러에 a, b, ..., p라는 값으로 각각 입력이 되어 있기 때문에 1, 2 ... 16을 입력하는 것은 즉 PICBASIC에 a, b, ..., p라는 값으로 입력 받는 것이라고 말할 수 있다.

각각의 채널의 정보를 C#이라는 언어로 만들어진 아래그림과 같은 프로그램에 정보를 출력이 되며 그 출력된 값들은 DB에 데이터를 저장함으로써 사용자가 필요할 때 데이터를 확인할 수 있다. 앞서 기술한 바와 같이 직렬기 서버를 DB와 함께 사용함으로써 거리가 먼 사용자에게도 그 결과값들을 바로 확인할 수 있게 되는 것이다.

출력된 데이터는 (그림 8)에서 보여지는 바와 같이 DB writing을 통해 각 채널 별로 데이터를 저장하거나 전체 채널을 저장할 수 있다. 이와 함께 DB writing을 통해 앞서 저장한 데이터를 확인할 수 있게 된다.

IV. 결론

본 논문은 센서 임베디드 시스템과 RF 통신 그리고 인터넷 네트워크에 의한 원격 계측 시스템 개발에 관한 것으로 이를 실제로 구현하는 시스템을 개발하여 신호 전송상의 문제점을 발견하고 이를 극복하고자 소프트웨어적인 알고리즘을 추구하여 신호 전송 에러의 문제점을 해소하였다. 하중 변화량의 정밀도는 12비트 A/D 변환기를 사용하였을 경우에 5g의 오차를 가지며 하중 측정은 1초 간격으로 원격 측정하도록 하였으며, PICBASIC 마이크로프로세서에서는 12비트의 데이터를 각각 8비트와 4비트로 나누어 host에 전송하도록 하였다. 본 연구는 계속해서 가정정보기기에 응용하여 스트레인 게이지가 아닌 가정정보기기의 디스플레이부나 제어부와 원격 계측 제어에 활용하는 시스템 개발에 적용하고자 한다.

참고문헌

- [1] S. Arimoto, "Linear controllable systems," *Nature*, vol. 135, pp. 18-27, July, 1990.
- [2] S.-C. Park, "Level up OrCAD," *Sungandang*, Jan. 2002.
- [3] C.-H. Lee, "OrCAD 9.0 theory and circuit drawing method," *Ijinsa*, Jan. 2001.
- [4] Y.-B. Cho, "OrCAD ver10.0," *Bukdu Press*, Aug. 2004.
- [5] S.-K. Ha, "ASP.NET 2003," *Darim Press*, Aug. 2003.
- [6] S.-J. Sung, "C# Web Programing actual fighting guide," *Devpia*, June, 2003.
- [7] T.-I. Kim, "Visual C#.NET Programing," *Devpia*, Oct. 2001.
- [8] [http://www.analog.com/UploadedFiles/Data\\_Sheets/8976538-54AD620\\_g.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/8976538-54AD620_g.pdf)
- [9] [http://www.analog.com/UploadedFiles/Data\\_Sheets/3466691-45AD1674\\_c.pdf](http://www.analog.com/UploadedFiles/Data_Sheets/3466691-45AD1674_c.pdf)
- [10] <http://icc.skku.ac.kr/%7Erfic/image/lecture/laboratory2/data-HA17741.pdf>
- [11] <http://www.comfile.co.kr/download/pb/PBMAN10.pdf>
- [12] [http://www.initium.co.kr/bizdata/userguide/Promi-SD사용설명서\\_V2.0.pdf](http://www.initium.co.kr/bizdata/userguide/Promi-SD사용설명서_V2.0.pdf)
- [13] <http://docs.tibbo.com/index.html?ds100.htm>



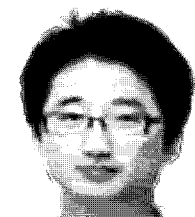
**장순석**  
1984년 한양대학교 전자공학과 학사 졸업. 1991년 영국 버밍엄대학교 전기전자공학과 박사 졸업. 1992년~현재 조선대학교 정보통신공학부 정교수. 연구 분야는 보청기, 의공학.



**김경석**  
2005년 조선대학교 정보제어계측공학과 졸업. 현재 조선대학교 일반대학원 제어계측공학과 석사과정.



**원용일**  
1987년 인하대학교 산업공학과 졸업. 2000년 휴스턴대학교 산업공학과 석사 졸업. 2005년 조선대학교 일반대학원 제어계측공학과 박사수료. 현재 남부대학교 디지털공학과 전임강사.



**김대곤**  
2001년 조선대학교 정보제어계측공학과 입학. 현재 조선대학교 정보제어계측공학과 3학년.