

```

using System;
using System.Text;
using System.Windows.Forms;

namespace InputTool
{
    public class Form1 : Form
    {
        private System.ComponentModel.IContainer components = null;
        private System.IO.Ports.SerialPort com1 = null;
        private System.Windows.Forms.ListBox listBox1;

        private byte[] byteBuffers = null;
        private string MeasurementValue = "";
        private bool receiveState = true;

        // フォームのコンストラクタ
        public Form1()
        {
            InitializeComponent();

            // シリアルポートの利用準備
            InitializeIT007R("COM1"); // 接続するシリアルポート番号を指定

            this.Load += new EventHandler(Form1_Load);
            this.FormClosed += new FormClosedEventHandler(Form1_FormClosed);
        }

        // フォームの最終処理
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        // フォームの初期設定
        private void InitializeComponent()
        {
            this.listBox1 = new System.Windows.Forms.ListBox();
            this.SuspendLayout();
            //
            // listBox1
            //
            this.listBox1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.listBox1.FormattingEnabled = true;
            this.listBox1.ItemHeight = 12;
            this.listBox1.Location = new System.Drawing.Point(0, 0);
            this.listBox1.Name = "listBox1";
            this.listBox1.Size = new System.Drawing.Size(284, 256);
            this.listBox1.TabIndex = 0;
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 12F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(284, 263);
            this.Controls.Add(this.listBox1);
            this.Name = "Form1";
            this.Text = "Form1";
            this.ResumeLayout(false);
        }
    }
}

```

```

// シリアルポートコントロールの初期設定
private void InitializeIT007R(string tgtPortName)
{
    if (com1 == null)
    {
        // ここで tgtPortName が有効なシリアルポート名であることを検証すること
        // このサンプルでは省略
        com1 = new System.IO.Ports.SerialPort(tgtPortName);
        com1.DataReceived += new System.IO.Ports.SerialDataReceivedEventHandler(OnCom1DataReceived);

        com1.BaudRate = 2400; // ボーレートは 2400bps
        com1.DataBits = 8; // データビットは 8 ビット
        com1.StopBits = System.IO.Ports.StopBits.One; // ストップビットは 1 ビット
        com1.Parity = System.IO.Ports.Parity.None; // パリティなし
        com1.DiscardNull = false;
        com1.Encoding = System.Text.Encoding.ASCII;

        byteBuffers = null;
    }
}

// フォームのロードイベント
void Form1_Load(object sender, EventArgs e)
{
    if (com1 == null) return;
    if (com1.IsOpen) return;

    try
    {
        com1.Open();
    }
    catch (Exception ex)
    {
        Console.Write(ex.Message);
        return;
    }
    com1.DtrEnable = true;
    com1.RtsEnable = true;
    com1.DiscardInBuffer();
    com1.DiscardOutBuffer();

    byteBuffers = null;
    receiveState = true;
}

// フォームのクローズイベント
void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    com1.DtrEnable = false;
    com1.RtsEnable = false;
    com1.Close();

    com1.Dispose();
    byteBuffers = null;
}

```

```

// シリアルポートの受信イベント
private void OnCom1DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    byte workByte;

    while (com1.BytesToRead > 0)
    {
        workByte = (byte)com1.ReadByte();
        if (workByte == 13)    // キャリッジリターンを受信したら
        {
            if (byteBuffers.Length > 0)
            {
                System.Text.Encoding workEncoding = System.Text.Encoding.Default;
                MeasurementValue = workEncoding.GetString(byteBuffers, 0, byteBuffers.Length);
                byteBuffers = null;
                this.Invoke(new MethodInvoker(listBox1ItemAddDelegator));
            }
            receiveState = true;
        }
        else if (receiveState)
        {
            receiveState = false;
            byte[] workBytes = new byte[1];
            workBytes[0] = workByte;
            byteBuffers = workBytes;
        }
        else
        {
            byte[] workBytes = new byte[byteBuffers.Length + 1];
            byteBuffers.CopyTo(workBytes, 0);
            workBytes[byteBuffers.Length] = workByte;
            byteBuffers = workBytes;
        }
    }
}

private void listBox1ItemAddDelegator()
{
    listBox1.Items.Add(MeasurementValue);
}

// プログラムスタートアップポイント
static class Program
{
    [STAThread]
    static void Main()
    {
        Application.Run(new Form1());
    }
}
}

```