

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
using System.Net;
using System.Net.Sockets;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void sendCommand()
            {
                byte[] data = new byte[1024];
                IPEndPoint ipep =
                    new IPEndPoint(IPAddress.Parse("127.0.0.1"), 9050);
                Socket server = new Socket(AddressFamily.InterNetwork,
SocketType.Dgram,
                ProtocolType.Udp);
                string sendString = textBox1.Text.ToString() + "\n";
                data = Encoding.ASCII.GetBytes(sendString);
                server.SendTo(data, data.Length, SocketFlags.None, ipep);

                csvStringChanged();
            }

            #####
            /// Reciving UDP data:
            #####

            private Thread workerThread;
            private void Form1_Load(object sender, EventArgs e)
            {
                workerThread = new Thread(this.DoReciveUDP);
                workerThread.IsBackground = true; //End
                // Start the worker thread.
                workerThread.Start();
                //checkBoxChanged();
            }
            public string strRecievedUDPMessage;

            public void DoReciveUDP()
            {
                UdpClient sock = new UdpClient(9050);
                IPEndPoint iep = new IPEndPoint(IPAddress.Any, 0);
                while (true)
                {
                    try
                    {
                        //Recieve message as UDP
                        byte[] data = sock.Receive(ref iep);
                        //Convert Bytes to a ASCII String
                        strRecievedUDPMessage =

```

```

        Encoding.ASCII.GetString(data, 0, data.Length);
        //Cal the function UdpDataRecieved
        this.Invoke(new EventHandler(this.UdpDataRecieved));
    }
    catch (Exception e) { }
}
sock.Close();
}
//This is only called by DoReciveUDP
public void UdpDataRecieved(object sender, EventArgs e)
{
    //Show the message in the form
    textBox2.Text = strRecievedUDPMessage;
    csvStringChanged();
}

private void csvStringChanged()
{
    string[] comands = textBox2.Text.Split(','); // Make a string table of
comands
    if (comands[0] == "killRead")
    {
        trackBar1.Value = Int32.Parse(comands[1]);
        trackBar2.Value = Int32.Parse(comands[2]);
    }
    if (comands[0] == "kill")
    {
        trackBar1.Value = Int32.Parse(comands[1]);
        trackBar2.Value = Int32.Parse(comands[2]);
    }
    if (comands[0] == "temp")
    {
        label6.Text = comands[1];
    }

    if (comands[0] == "killCount")
    {
        label8.Text = comands[1];
    }
    if (comands[0] == "dogngrader")
    {
        progressBar1.Value = Int32.Parse(comands[1]);
    }

}

#####
//# buttons and other actions:
#####

private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = "kill,100,4";
    sendCommand();
}

private void button2_Click(object sender, EventArgs e)
{
    textBox1.Text = "kill,50,2";
    sendCommand();
}

```

```
private void button3_Click(object sender, EventArgs e)
{
    textBox1.Text = "kill," + trackBar1.Value+ "," + trackBar2.Value;
    sendCommand();
}

private void button4_Click(object sender, EventArgs e)
{
    textBox1.Text = "temp,100,4";
    sendCommand();
}

private void button5_Click(object sender, EventArgs e)
{
    textBox1.Text = "killCount,100,4";
    sendCommand();
}

private void button6_Click(object sender, EventArgs e)
{
    pictureBox1.ImageLocation = "http://pi09.soc3001.net/capture.jpg";
}

private void button7_Click(object sender, EventArgs e)
{
    textBox1.Text = "reset";
    sendCommand();
    progressBar1.Value = 0;
}
}
```