GNSS信号中频数据采集与预处理软件源码

1. **Program.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace usb20in

{

static class Program

{

/// <summary>

/// 应用程序的主入口点。

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

1. **Form.cs**

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.Windows.Forms;

using CyUSB;

using System.Threading;

using System.IO;

namespace usb20in

{

public partial class Form1 : Form

{

USBDeviceList usbDevices;

CyUSBDevice myDevice;

CyControlEndPoint CtrlEndPt=null;

CyBulkEndPoint BulkIn=null;

CyBulkEndPoint BulkOut = null;

public Form1()

{

//InitializeComponent();

InitializeComponent();

usbDevices = new USBDeviceList(CyConst.DEVICES\_CYUSB);

usbDevices.DeviceAttached += new EventHandler(usbDevices\_DeviceAttached);

usbDevices.DeviceRemoved += new EventHandler(usbDevices\_DeviceRemoved);

// Get the first device having VendorID == 0x2222 and ProductID == 0x5555

myDevice = usbDevices[0x04B4, 0x00F1] as CyUSBDevice;

if (myDevice != null)

{

toolStripStatusLabel1.Text = myDevice.FriendlyName + " connected.";

CtrlEndPt = myDevice.ControlEndPt;

foreach (CyUSBEndPoint ept in myDevice.EndPoints)

{

if (ept.bIn && (ept.Attributes == 2))

BulkIn = ept as CyBulkEndPoint;

if ((!ept.bIn) && (ept.Attributes == 2))

BulkOut = ept as CyBulkEndPoint;

}

}

}

void usbDevices\_DeviceRemoved(object sender, EventArgs e)

{

USBEventArgs usbEvent = e as USBEventArgs;

toolStripStatusLabel1.Text = usbEvent.FriendlyName + " removed.";

CtrlEndPt = null;

BulkIn = null;

BulkOut = null;

}

void usbDevices\_DeviceAttached(object sender, EventArgs e)

{

USBEventArgs usbEvent = e as USBEventArgs;

toolStripStatusLabel1.Text = usbEvent.Device.FriendlyName + " connected.";

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

CtrlEndPt.Target = CyConst.TGT\_ENDPT;

CtrlEndPt.ReqType = CyConst.REQ\_VENDOR;

CtrlEndPt.Direction = CyConst.DIR\_TO\_DEVICE;

CtrlEndPt.ReqCode = 0xd0;

CtrlEndPt.Value = 0;

CtrlEndPt.Index = 0;

int len =512;

byte[] buf = new byte[512];

bool rfsetok = true;

if (rfsetok)

textBox1.AppendText("rfset ok!\r\n");

else

textBox1.AppendText("rfset error!\r\n");

}

private void button2\_Click(object sender, EventArgs e)

{

Control.CheckForIllegalCrossThreadCalls = false;

Thread thread = new Thread(ThreadFuntion);

thread.IsBackground = true;

thread.Start();

}

private void ThreadFuntion()

{

runflag = true;

textBox1.Text = "";

FileStream fw;

if (comboBox3.Text.Contains("INT8") || comboBox3.Text == "RAWDATA")

fw = new FileStream("E:\\FastData\\IF\\usbdata\_n.bin", FileMode.Create);//C:\\MATLAB7\\work\\usbdata.bin//I:\\prom\\work\\

else

fw = new FileStream("E:\\data\\usbdataB3I\_16M369X4\_1.dat", FileMode.Create);

BinaryWriter bw;

bw = new BinaryWriter(fw);

int len = 0x40000; //总字节数

byte[] buf = new byte[0x80000];

byte[] bufin = new byte[0x80000];

//len=1/4MB,对于80MB/s,大概为320

//对于64MB/s,大概为大概为256

//对于32MB/s，大概为128

//对于40MB/s，大概为160

int speed = 64;

if (comboBox6.Text == "16.369MHz")

speed = 64;

else if (comboBox6.Text == "10MHz")

speed = 40;

int hour=0,min=0,s=0;

hour = Convert.ToInt32(textBox2.Text);

min=Convert.ToInt32(textBox3.Text);

s = Convert.ToInt32(textBox4.Text);

int kmax=(hour \* 3600 + min \* 60 + s) \* speed \* datascale;

int k = 0;

bool savefileok=false;

progressBar1.Minimum = 0;

progressBar1.Maximum = kmax;

//定义数组用于查表

byte[] IF2BIT = { 0x1, 0x3, 0xff, 0xfd };

//byte[] IF3BIT = { 0x1,0x3,0x5,0x7,0xf9,0xfb,0xfd,0xff};//TWO’S COMPLEMENT BINARY

byte[] IF3BIT = { 0x1, 0x3, 0x5, 0x7, 0xff, 0xfd, 0xfb, 0xf9 };//SIGN/MAGNITUDE

//用一个bulkout复位清空DDR3缓存

BulkOut.XferData(ref buf, ref len);

Thread.Sleep(100);

int[] flags = new int[] { 1, 1, 1, 1, 1, 1, 1, 1 };

if (comboBox3.Text == "RAWDATA")

while (BulkIn != null && k++ < kmax && runflag == true)

{

BulkIn.XferData(ref buf, ref len);

bw.Write(buf, 0, len);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

if (comboBox3.Text == "RAWDATA\_INT8")

{

//List<BinaryWriter> bwt = new List<BinaryWriter>();//创建了一个空列表

//List<FileStream> fwt = new List<FileStream>();//创建了一个空列表

//List<string> fullpatht = new List<string>();//创建了一个空列表

BinaryWriter[] bwt = new BinaryWriter[8];//创建了一个空数组

FileStream[] fwt = new FileStream[8];//创建了一个空数组

string[] fullpatht = new string[8];//创建了一个空数组

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

string path = "E:\\FastData\\IF\_Test\\";

string name = "usbdata" + Convert.ToString(i) + ".bin";

fullpatht[i - 1] = path + name; //保存文件完整路径

fwt[i - 1] = new FileStream(fullpatht[i - 1], FileMode.Create);

bwt[i - 1] = new BinaryWriter(fwt[i - 1]);

}

}

while (BulkIn != null && k++ < kmax && runflag == true)

{

BulkIn.XferData(ref buf, ref len);

bw.Write(buf, 0, len);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

int length = len / 16;

//BulkIn.XferData(ref buf, ref len);

switch (i)

{

case 1:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16] >> 2) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 4] >> 2) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 8] >> 2) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 12] >> 2) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 0) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 0) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 0) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 0) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 2:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16] >> 2) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 4] >> 2) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 8] >> 2) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 12] >> 2) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 0) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 0) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 0) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 0) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 3:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 1] >> 2) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 5] >> 2) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 9] >> 2) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 13] >> 2) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 1] >> 0) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 5] >> 0) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 9] >> 0) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 13] >> 0) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 4:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 1] >> 6) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 5] >> 6) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 9] >> 6) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 13] >> 6) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 1] >> 4) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 5] >> 4) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 9] >> 4) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 13] >> 4) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 5:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 2] >> 2) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 6] >> 2) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 10] >> 2) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 14] >> 2) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 2] >> 0) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 6] >> 0) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 10] >> 0) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 14] >> 0) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 6:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 2] >> 6) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 6] >> 6) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 10] >> 6) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 14] >> 6) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 4) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 4) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 4) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 4) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 7:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 3] >> 2) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 7] >> 2) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 11] >> 2) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 15] >> 2) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 3] >> 0) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 7] >> 0) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 11] >> 0) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 15] >> 0) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

case 8:

for (int j = 0; j < length; j++)

{

bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 3] >> 6) & 0x03];

bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 7] >> 6) & 0x03];

bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 11] >> 6) & 0x03];

bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 15] >> 6) & 0x03];

bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 3] >> 4) & 0x03];

bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 7] >> 4) & 0x03];

bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 11] >> 4) & 0x03];

bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 15] >> 4) & 0x03];

}

bwt[i-1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

break;

default:

break;

}

}

}

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

bwt[i - 1].Close();

fwt[i - 1].Close();

}

}

}

else if (comboBox3.Text == "RF2IRF1I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 8;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 8 + 4] >> 0) & 0xC0) | ((buf[i \* 8 + 4] >> 2) & 0x30) | ((buf[i \* 8] >> 4) & 0xC) | ((buf[i \* 8] >> 2) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2IQRF1IQ\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 4;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = buf[i \* 4];

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF1I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 12] << 4) & 0xC0) | ((buf[i \* 16 + 8] << 2) & 0x30) | ((buf[i \* 16 + 4] >> 0) & 0xC) | ((buf[i \* 16] >> 2) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF1Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 12] << 6) & 0xC0) | ((buf[i \* 16 + 8] << 4) & 0x30) | ((buf[i \* 16 + 4] << 2) & 0xC) | ((buf[i \* 16] >> 0) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 12] >> 0) & 0xC0) | ((buf[i \* 16 + 8] >> 2) & 0x30) | ((buf[i \* 16 + 4] >> 4) & 0xC) | ((buf[i \* 16] >> 6) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 12] << 2) & 0xC0) | ((buf[i \* 16 + 8] >> 0) & 0x30) | ((buf[i \* 16 + 4] >> 2) & 0xC) | ((buf[i \* 16] >> 4) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF3I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 13] << 4) & 0xC0) | ((buf[i \* 16 + 9] << 2) & 0x30) | ((buf[i \* 16 + 5] >> 0) & 0xC) | ((buf[i \* 16 + 1] >> 2) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF3Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 13] << 6) & 0xC0) | ((buf[i \* 16 + 9] << 4) & 0x30) | ((buf[i \* 16 + 5] << 2) & 0xC) | ((buf[i \* 16 + 1] >> 0) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF4I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 13] >> 0) & 0xC0) | ((buf[i \* 16 + 9] >> 2) & 0x30) | ((buf[i \* 16 + 5] >> 4) & 0xC) | ((buf[i \* 16 + 1] >> 6) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF4Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 13] << 2) & 0xC0) | ((buf[i \* 16 + 9] >> 0) & 0x30) | ((buf[i \* 16 + 5] >> 2) & 0xC) | ((buf[i \* 16 + 1] >> 4) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 14] << 4) & 0xC0) | ((buf[i \* 16 + 10] << 2) & 0x30) | ((buf[i \* 16 + 6] >> 0) & 0xC) | ((buf[i \* 16 + 2] >> 2) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 14] << 6) & 0xC0) | ((buf[i \* 16 + 10] << 4) & 0x30) | ((buf[i \* 16 + 6] << 2) & 0xC) | ((buf[i \* 16 + 2] >> 0) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF6I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 14] >> 0) & 0xC0) | ((buf[i \* 16 + 10] >> 2) & 0x30) | ((buf[i \* 16 + 6] >> 4) & 0xC) | ((buf[i \* 16 + 2] >> 6) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF6Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 14] << 2) & 0xC0) | ((buf[i \* 16 + 10] >> 0) & 0x30) | ((buf[i \* 16 + 6] >> 2) & 0xC) | ((buf[i \* 16 + 2] >> 4) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 15] << 4) & 0xC0) | ((buf[i \* 16 + 11] << 2) & 0x30) | ((buf[i \* 16 + 7] >> 0) & 0xC) | ((buf[i \* 16 + 3] >> 2) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 15] << 6) & 0xC0) | ((buf[i \* 16 + 11] << 4) & 0x30) | ((buf[i \* 16 + 7] << 2) & 0xC) | ((buf[i \* 16 + 3] >> 0) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF8I\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 15] >> 0) & 0xC0) | ((buf[i \* 16 + 11] >> 2) & 0x30) | ((buf[i \* 16 + 7] >> 4) & 0xC) | ((buf[i \* 16 + 3] >> 6) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF8Q\_BIT2")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i] = (byte)(((buf[i \* 16 + 15] << 2) & 0xC0) | ((buf[i \* 16 + 11] >> 0) & 0x30) | ((buf[i \* 16 + 7] >> 2) & 0xC) | ((buf[i \* 16 + 3] >> 4) & 0x3));

}

bw.Write(bufin, 0, length);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF1I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16] >> 2) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 4] >> 2) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 8] >> 2) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 12] >> 2) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF1Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16] >> 0) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 4] >> 0) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 8] >> 0) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 12] >> 0) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF1I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16] >> 1) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 4] >> 1) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 8] >> 1) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 12] >> 1) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16] >> 6) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 4] >> 6) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 8] >> 6) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 12] >> 6) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16] >> 4) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 4] >> 4) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 8] >> 4) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 12] >> 4) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF2I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16] >> 5) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 4] >> 5) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 8] >> 5) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 12] >> 5) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF3I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 1] >> 2) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 5] >> 2) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 9] >> 2) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 13] >> 2) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF3Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 1] >> 0) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 5] >> 0) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 9] >> 0) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 13] >> 0) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF3I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 1] >> 1) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 5] >> 1) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 9] >> 1) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 13] >> 1) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF4I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 1] >> 6) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 5] >> 6) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 9] >> 6) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 13] >> 6) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF4Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 1] >> 4) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 5] >> 4) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 9] >> 4) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 13] >> 4) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF4I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 1] >> 5) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 5] >> 5) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 9] >> 5) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 13] >> 5) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 2] >> 2) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 6] >> 2) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 10] >> 2) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 14] >> 2) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 2] >> 0) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 6] >> 0) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 10] >> 0) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 14] >> 0) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF5I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 2] >> 1) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 6] >> 1) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 10] >> 1) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 14] >> 1) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF6I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 2] >> 6) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 6] >> 6) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 10] >> 6) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 14] >> 6) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF6Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 2] >> 4) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 6] >> 4) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 10] >> 4) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 14] >> 4) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF6I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 2] >> 5) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 6] >> 5) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 10] >> 5) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 14] >> 5) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF7I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 3] >> 2) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 7] >> 2) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 11] >> 2) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 15] >> 2) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF7Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 3] >> 0) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 7] >> 0) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 11] >> 0) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 15] >> 0) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF7I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 3] >> 1) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 7] >> 1) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 11] >> 1) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 15] >> 1) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF8I\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 3] >> 6) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 7] >> 6) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 11] >> 6) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 15] >> 6) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF8Q\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF2BIT[(buf[i \* 16 + 3] >> 4) & 0x03];

bufin[i \* 4 + 1] = IF2BIT[(buf[i \* 16 + 7] >> 4) & 0x03];

bufin[i \* 4 + 2] = IF2BIT[(buf[i \* 16 + 11] >> 4) & 0x03];

bufin[i \* 4 + 3] = IF2BIT[(buf[i \* 16 + 15] >> 4) & 0x03];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

else if (comboBox3.Text == "RF8I\_BIT3\_INT8")

while (BulkIn != null && k++ < kmax && runflag == true)

{

int length = len / 16;

BulkIn.XferData(ref buf, ref len);

for (int i = 0; i < length; i++)

{

bufin[i \* 4] = IF3BIT[(buf[i \* 16 + 3] >> 5) & 0x07];

bufin[i \* 4 + 1] = IF3BIT[(buf[i \* 16 + 7] >> 5) & 0x07];

bufin[i \* 4 + 2] = IF3BIT[(buf[i \* 16 + 11] >> 5) & 0x07];

bufin[i \* 4 + 3] = IF3BIT[(buf[i \* 16 + 15] >> 5) & 0x07];

}

bw.Write(bufin, 0, length \* 4);

if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

}

progressBar1.Value = kmax;

savefileok = true;

bw.Close();

fw.Close();

if (savefileok)

textBox1.AppendText("save file ok!\r\n");

else

textBox1.AppendText("save file error!\r\n");

}

private void button3\_Click(object sender, EventArgs e)

{

textBox1.Text = "";

textBox1.Text += "1、本采集器使用MAX2771射频,详细设置可参考MAX2771 datasheet。\r\n";

textBox1.Text += "2、本软件默认提供8个射频IQ支路，共32bit原始数据同时输出。\r\n";

textBox1.Text += "3、原始数据每32bit，RF8I在最高位，RF8Q在次高字节，RF1I在次低字位，RF1Q在最低位\r\n";

textBox1.Text += "4、TCXO=16.369MHz,采样率：16.369MHz，GPS中频中心频点：3.996MHz，BD中频中心频点：3.996875MHz。\r\n";

textBox1.Text += "5、Matlab软件接收机数据格式为int8,无需额外转换。\r\n";

}

bool runflag = true;

private void button4\_Click(object sender, EventArgs e)

{

runflag = false;

}

int datawidth=16;

int clkmul = 1;

int datascale = 1;

private void button5\_Click(object sender, EventArgs e)

{

//myDevice.ReConnect();

//myDevice.Reset();

//textBox1.AppendText("Reset OK!\r\n");

CtrlEndPt.Target = CyConst.TGT\_ENDPT;

CtrlEndPt.ReqType = CyConst.REQ\_VENDOR;

CtrlEndPt.Direction = CyConst.DIR\_TO\_DEVICE;

CtrlEndPt.ReqCode = 0xC0;

CtrlEndPt.Value = 0;

CtrlEndPt.Index = 0;

int len = 4;

byte[] buf = new byte[4] { 0, 0, 0, 0 };

if (comboBox5.Text == "16bit") datawidth = 16;

else if (comboBox5.Text == "8bit") datawidth = 8;

else if (comboBox5.Text == "32bit") datawidth = 32;

else

datawidth = 16;

buf[0] = 3;

if (comboBox4.Text == "1倍频") { clkmul = 1; buf[0] = 0; datascale = datawidth \* clkmul / 8; }

else if (comboBox4.Text == "2倍频") { clkmul = 2; buf[0] = 1; datascale = datawidth \* clkmul / 8; }

else if (comboBox4.Text == "4倍频") { clkmul = 4; buf[0] = 3; datascale = datawidth \* clkmul / 8; }

else { clkmul = 1; buf[0] = 3; datascale = datawidth \* clkmul / 8; }

bool modesetok = true;

if (CtrlEndPt.XferData(ref buf, ref len) == false && modesetok == true) modesetok = false;

textBox1.AppendText(String.Format("mode={0:d},datascale={1:d}\r\n", buf[0], datascale));

if (modesetok)

textBox1.AppendText("modeset ok!\r\n");

else

textBox1.AppendText("modeset error!\r\n");

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void comboBox3\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void groupBox5\_Enter(object sender, EventArgs e)

{

}

private void button7\_Click(object sender, EventArgs e)

{

Control.CheckForIllegalCrossThreadCalls = false;

Thread thread = new Thread(ThreadFuntion\_Convert);

thread.IsBackground = true;

thread.Start();

}

private void ThreadFuntion\_Convert()

{

runflag = true;

textBox1.Text = "";

string filePath = "E:\\FastData\\IF\_Test\\";

string fileName = "usbdata\_D2";

string filePathName = filePath + fileName + ".bin"; //保存文件完整路径

FileStream fr;

fr = new FileStream(filePathName, FileMode.Open);

BinaryReader br;

br = new BinaryReader(fr);

FileInfo fileInfo = new FileInfo(filePathName);

int kmax = Convert.ToInt32(fileInfo.Length /4);

int k = 0;

bool savefileok = false;

progressBar2.Minimum = 0;

progressBar2.Maximum = kmax;

//int len = 0x40000; //总字节数

//byte[] buf = new byte[0x80000];

//byte[] bufin = new byte[0x80000];

////len=1/4MB,对于80MB/s,大概为320

////对于64MB/s,大概为大概为256

////对于32MB/s，大概为128

////对于40MB/s，大概为160

//定义数组用于查表

byte[] IF2BIT = { 0x1, 0x3, 0xff, 0xfd };

//byte[] IF3BIT = { 0x1,0x3,0x5,0x7,0xf9,0xfb,0xfd,0xff};//TWO’S COMPLEMENT BINARY

byte[] IF3BIT = { 0x1, 0x3, 0x5, 0x7, 0xff, 0xfd, 0xfb, 0xf9 };//SIGN/MAGNITUDE

//用一个bulkout复位清空DDR3缓存

//BulkOut.XferData(ref buf, ref len);

//Thread.Sleep(100);

int[] flags = new int[] { 1, 1, 1, 1, 1, 1, 1, 1 };

int rawdata = 0;

byte binI, binQ;

BinaryWriter[] bwt = new BinaryWriter[8];//创建了一个空数组

FileStream[] fwt = new FileStream[8];//创建了一个空数组

string[] fullpatht = new string[8];//创建了一个空数组

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

string path = "G:\\FastData\\IF\_Test\\";// filePath;

string name = fileName +"\_" + Convert.ToString(i);

fullpatht[i - 1] = path + name + ".bin"; //保存文件完整路径

fwt[i - 1] = new FileStream(fullpatht[i - 1], FileMode.Create);

bwt[i - 1] = new BinaryWriter(fwt[i - 1]);

}

}

try

{

while (k++ < kmax && runflag == true) //while (BulkIn != null && k++ < kmax && runflag == true)

{

rawdata = br.ReadInt32();

//bw.Write(buf, 0, len);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

//int length = len / 16;

//BulkIn.XferData(ref buf, ref len);

switch (i)

{

case 1:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16] >> 2) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 4] >> 2) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 8] >> 2) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 12] >> 2) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 0) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 0) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 0) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 0) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 2) & 0x03];

binQ = IF2BIT[(rawdata >> 0) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 2:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16] >> 2) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 4] >> 2) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 8] >> 2) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 12] >> 2) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 0) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 0) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 0) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 0) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 6) & 0x03];

binQ = IF2BIT[(rawdata >> 4) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 3:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 1] >> 2) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 5] >> 2) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 9] >> 2) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 13] >> 2) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 1] >> 0) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 5] >> 0) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 9] >> 0) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 13] >> 0) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 10) & 0x03];

binQ = IF2BIT[(rawdata >> 8) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 4:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 1] >> 6) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 5] >> 6) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 9] >> 6) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 13] >> 6) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 1] >> 4) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 5] >> 4) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 9] >> 4) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 13] >> 4) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 14) & 0x03];

binQ = IF2BIT[(rawdata >> 12) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 5:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 2] >> 2) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 6] >> 2) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 10] >> 2) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 14] >> 2) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 2] >> 0) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 6] >> 0) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 10] >> 0) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 14] >> 0) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 18) & 0x03];

binQ = IF2BIT[(rawdata >> 16) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 6:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 2] >> 6) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 6] >> 6) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 10] >> 6) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 14] >> 6) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16] >> 4) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 4] >> 4) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 8] >> 4) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 12] >> 4) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 22) & 0x03];

binQ = IF2BIT[(rawdata >> 20) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 7:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 3] >> 2) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 7] >> 2) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 11] >> 2) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 15] >> 2) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 3] >> 0) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 7] >> 0) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 11] >> 0) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 15] >> 0) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 26) & 0x03];

binQ = IF2BIT[(rawdata >> 24) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

case 8:

//for (int j = 0; j < length; j++)

//{

// bufin[j \* 8] = IF2BIT[(buf[j \* 16 + 3] >> 6) & 0x03];

// bufin[j \* 8 + 2] = IF2BIT[(buf[j \* 16 + 7] >> 6) & 0x03];

// bufin[j \* 8 + 4] = IF2BIT[(buf[j \* 16 + 11] >> 6) & 0x03];

// bufin[j \* 8 + 6] = IF2BIT[(buf[j \* 16 + 15] >> 6) & 0x03];

// bufin[j \* 8 + 1] = IF2BIT[(buf[j \* 16 + 3] >> 4) & 0x03];

// bufin[j \* 8 + 3] = IF2BIT[(buf[j \* 16 + 7] >> 4) & 0x03];

// bufin[j \* 8 + 5] = IF2BIT[(buf[j \* 16 + 11] >> 4) & 0x03];

// bufin[j \* 8 + 7] = IF2BIT[(buf[j \* 16 + 15] >> 4) & 0x03];

//}

//bwt[i - 1].Write(bufin, 0, length \* 8);

//if ((k % (kmax / 100)) == 0) progressBar1.Value = k;

//bw.Close();

//fw.Close();

binI = IF2BIT[(rawdata >> 30) & 0x03];

binQ = IF2BIT[(rawdata >> 28) & 0x03];

bwt[i - 1].Write(binI);

bwt[i - 1].Write(binQ);

break;

default:

break;

}

}

}

if ((k % (kmax / 100)) == 0) progressBar2.Value = k;

//Console.WriteLine("{0},{1},{2},{2}", cha, num, doub, str);

}

}

catch (EndOfStreamException e)

{

//Console.WriteLine(e.Message);

//Console.WriteLine("已经读到末尾");

textBox1.AppendText("已经读到末尾!\r\n");

}

finally

{

for (int i = 1; i <= 8; i++)

{

if (flags[i - 1] == 1)

{

bwt[i - 1].Close();

fwt[i - 1].Close();

}

}

//Console.ReadKey();

}

progressBar2.Value = kmax;

savefileok = true;

br.Close();

fr.Close();

if (savefileok)

textBox1.AppendText("save file ok!\r\n");

else

textBox1.AppendText("save file error!\r\n");

}

private void groupBox7\_Enter(object sender, EventArgs e)

{

}

}

}