DUO C# Samples

Overview

This article is a review of the `C#` Microsoft .NET samples that ship with the DUO SDK. This will be updated as new features are added or specifications updated to the DUO API. In the `Developers/Samples` folder within the SDK download you can find the latest version of these samples with a Visual Studio project to compile.

Prerequisites

Before reading this guide it is recommended to review our API and SDK Docs to get an understanding of the design, common practices and usage. With DUO devices you will always receive a "frame" which contains all relevant sensor data. The following examples showcase the usage of the DUOFrame structure while using different device functions. Also make sure you have the latest SDK download.

- DUOLib.lib & DUOLib dynamic library that match your compiler and architecture (x86/x64)
- DUODeviceLib provided DUOLib interop library and bindings (Included)
- Microsoft Windows with Visual Studio 2012 or newer
- Microsoft .NET Framework 2-4 or newer

Methods/Structures

Here are some of the common methods used through-out the examples:

- `EnumerateResolutions` - Lists available resolutions.
    - `DUOResolutionInfo` - Structure used for resolution information.
- `DUOFrameCallback` - Callback on frame update.
- `OpenDUO` - Opens a new connection to the device.
- `StartDUO` - Initializes capture on the device.
- `StopDUO` - Un-Initializes capture on the device.
- `CloseDUO` - Closes the connection to the device.

And variables/structures:

- `DUOFrame` - Structure containing device sensor data.
    - `pFrameData->timeStamp` - Frame time stamp in 100μs increments.

DUODeviceLib

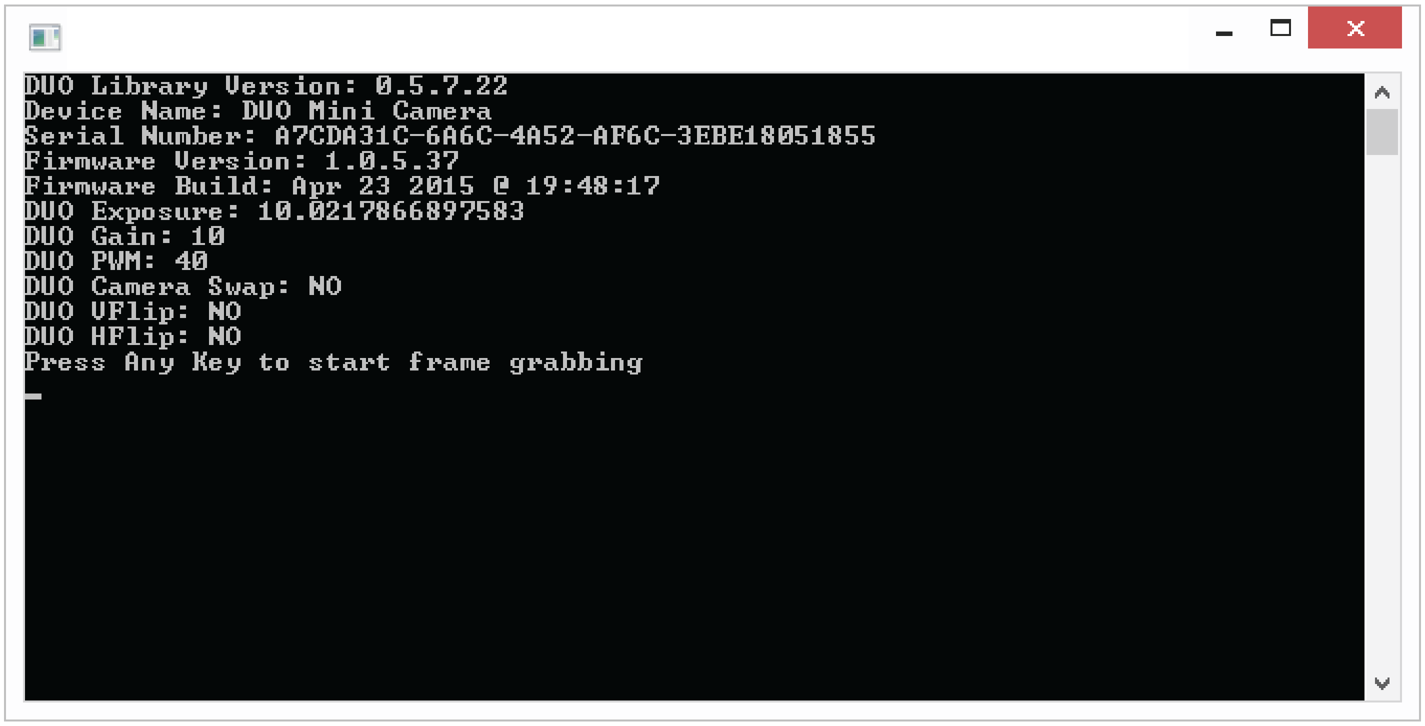
To access the DUOLib in C# we use provide a class to help facilitate interaction with the native DLL. Here is a quick overview of the class and the key methods exposed to C#.

```
using System;
using System.Runtime.InteropServices;
namespace DUODeviceLib
{
    public delegate void DUOFrameCallback(ref DUOFrame pFrameData, IntPtr pUserData);
    public static class DUOLib
    {
        public static string GetLibVersion();
        public static int EnumerateResolutions(DUOResolutionInfo[] resolutions, int resListSize, int
 width, int height , DUOBinning binning, float fps);
        public static extern bool OpenDUO(ref IntPtr duo);
        public static extern bool CloseDUO(IntPtr duo);
        public static extern bool StartDUO(IntPtr duo, DUOFrameCallback frameCallback, IntPtr pUserD
ata, bool masterMode = true);
        public static extern bool StopDUO(IntPtr duo);
        public static bool GetDUODeviceName(IntPtr duo, ref string val);
        public static bool GetDUOFirmwareVersion(IntPtr duo, ref string val);
        public static bool GetDUOFirmwareBuild(IntPtr duo, ref string val);
        public static bool GetDUOResolutionInfo(IntPtr duo, ref DUOResolutionInfo val);
        public static bool GetDUOExposure(IntPtr duo, ref double val);
        public static bool GetDUOExposureMS(IntPtr duo, ref double val);
        public static bool GetDUOGain(IntPtr duo, ref double val);
        public static bool GetDUOHFlip(IntPtr duo, ref bool val);
        public static bool GetDUOVFlip(IntPtr duo, ref bool val);
        public static bool GetDUOCameraSwap(IntPtr duo, ref bool val);
        public static bool GetDUOLedPWM(IntPtr duo, ref double val);
        public static bool GetDUOFrameDimension(IntPtr duo, ref uint w, ref uint h);
        public static bool SetDUOResolutionInfo(IntPtr duo, DUOResolutionInfo val);
        public static bool SetDUOExposure(IntPtr duo, double val);
        public static bool SetDUOExposureMS(IntPtr duo, double val);
        public static bool SetDUOGain(IntPtr duo, double val);
        public static bool SetDUOHFlip(IntPtr duo, bool val);
        public static bool SetDUOVFlip(IntPtr duo, bool val);
        public static bool SetDUOCameraSwap(IntPtr duo, bool val);
        public static bool SetDUOLedPWM(IntPtr duo, double val);
        public static bool SetDUOLedPWMSeq(IntPtr duo, DUOLEDSeq[] seq, uint count);
    }
}
```

Sample 01

Console Application

Example using the DUOLib in C# via the DUODeviceLib bindings, in this sample we showcase setting up a device and outputting results to a standard console window. Accessing the DUOLib class directly compared to using the Device helper class as we showcase in the next sample.



```
DUO Library Version: 0.5.7.22
Device Name: DUO Mini Camera
Serial Number: A7CDA31C-6A6C-4A52-AF6C-3EBE18051855
Firmware Version: 1.0.5.37
Firmware Build: Apr 23 2015 @ 19:48:17
DUO Exposure: 10.0217866897583
DUO Gain: 10
DUO PWM: 40
DUO Camera Swap: NO
DUO VFlip: NO
DUO HFlip: NO
Press Any Key to start frame grabbing
_
```

Program.cs

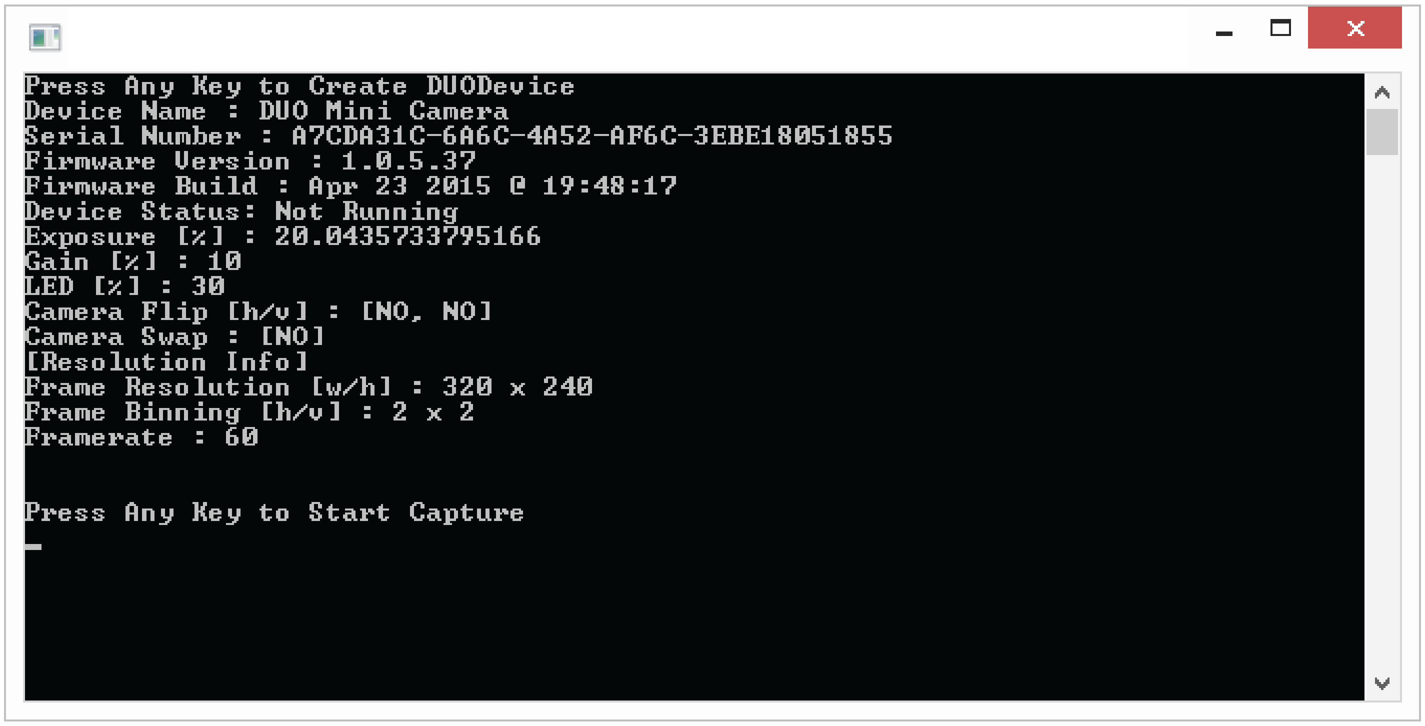
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DUODeviceLib;

namespace DUOLibConsoleDemo
{
    class Program
    {
        static DUOFrameCallback frameCallback = null;
        static int frameCount = 0;
        static void FrameCallback(ref DUOFrame pFrameData, IntPtr pUserData)
        {
            frameCount++;
            Console.WriteLine("Frame ID: {0}, Timestamp: {1}", frameCount, pFrameData.timeStamp);
            if (pFrameData.accelerometerPresent)
            {
                Console.WriteLine("Acceleration : ({0}:{1}:{2})", pFrameData.accelData[0], pFrameDat
a.accelData[1], pFrameData.accelData[2]);
                Console.WriteLine("Gyro : ({0}:{1}:{2})", pFrameData.gyroData[0], pFrameData.gyroDat
a[1], pFrameData.gyroData[2]);
                Console.WriteLine("Temperature : {0} C", pFrameData.tempData);
            }
        }
        static void Main(string[] args)
        {
            string libName = DUOLib.GetLibVersion();
            Console.WriteLine("DUO Library Version: {0}", libName);
            IntPtr duo = IntPtr.Zero;
            frameCallback = new DUOFrameCallback(FrameCallback);
            bool isOpened = DUOLib.OpenDUO(ref duo);
            if (isOpened)
            {
                string infoString = "";
                if (DUOLib.GetDUODeviceName(duo, ref infoString))
                    Console.WriteLine("Device Name: {0}", infoString);
                if (DUOLib.GetDUOSerialNumber(duo, ref infoString))
                    Console.WriteLine("Serial Number: {0}", infoString);
                if (DUOLib.GetDUOFirmwareVersion(duo, ref infoString))
                    Console.WriteLine("Firmware Version: {0}", infoString);
                if (DUOLib.GetDUOFirmwareBuild(duo, ref infoString))
                    Console.WriteLine("Firmware Build: {0}", infoString);
                DUOResolutionInfo[] resInfos = new DUOResolutionInfo[1];
                int resolutionsCount =
                DUOLib.EnumerateResolutions(resInfos, 1, 320, 240,
                (DUOBinning)((int)DUOBinning.DUO_BIN_HORIZONTAL2 + DUOBinning.DUO_BIN_VERTICAL2), 60
);
                if (resolutionsCount == 0x00)
                {
                    DUOLib.CloseDUO(duo);
                    return;
                }
                DUOLib.SetDUOResolutionInfo(duo, resInfos[0]);
                DUOLib.SetDUOCameraSwap(duo, false);
                DUOLib.SetDUOHFlip(duo, false);
                DUOLib.SetDUOVFlip(duo, false);
                DUOLib.SetDUOExposure(duo, 10.0);
                DUOLib.SetDUOGain(duo, 10.0);
                DUOLib.SetDUOLedPWM(duo, 40.0);
                double paramValue = 0.0;
                if (DUOLib.GetDUOExposure(duo, ref paramValue))
                    Console.WriteLine("DUO Exposure: {0}", paramValue);
                if (DUOLib.GetDUOGain(duo, ref paramValue))
                    Console.WriteLine("DUO Gain: {0}", paramValue);
                if (DUOLib.GetDUOLedPWM(duo, ref paramValue))
```

```
                    Console.WriteLine("DUO PWM: {0}", paramValue);
                bool bValue = false;
                if (DUOLib.GetDUOCameraSwap(duo, ref bValue))
                    Console.WriteLine("DUO Camera Swap: {0}", bValue ? "YES" : "NO");
                if (DUOLib.GetDUOVFlip(duo, ref bValue))
                    Console.WriteLine("DUO VFlip: {0}", bValue ? "YES" : "NO");
                if (DUOLib.GetDUOHFlip(duo, ref bValue))
                    Console.WriteLine("DUO HFlip: {0}", bValue ? "YES" : "NO");
                Console.WriteLine("Press Any Key to start frame grabbing");
                Console.ReadKey();
                if (DUOLib.StartDUO(duo, frameCallback, IntPtr.Zero))
                {
                    Console.WriteLine("Press Any Key to stop frame grabbing");
                    Console.ReadKey();
                    DUOLib.StopDUO(duo);
                }
                DUOLib.CloseDUO(duo);
            }
            Console.WriteLine("Press Any Key to exit application");
            Console.ReadKey();
        }
    }
}
```

Sample 02

Console Application (using the Device Class)

Example using the DUOLib in C# via the DUODeviceLib and DUODevice helper class. This class helps simplify using the DUO in your application by providing common features such as device creation as well as using traditional event system to update/notify upon parameter changes.



```
Press Any Key to Create DUODevice
Device Name : DUO Mini Camera
Serial Number : A7CDA31C-6A6C-4A52-AF6C-3EBE18051855
Firmware Version : 1.0.5.37
Firmware Build : Apr 23 2015 @ 19:48:17
Device Status: Not Running
Exposure [%] : 20.0435733795166
Gain [%] : 10
LED [%] : 30
Camera Flip [h/v] : [NO, NO]
Camera Swap : [NO]
[Resolution Info]
Frame Resolution [w/h] : 320 x 240
Frame Binning [h/v] : 2 x 2
Framerate : 60


Press Any Key to Start Capture
_
```

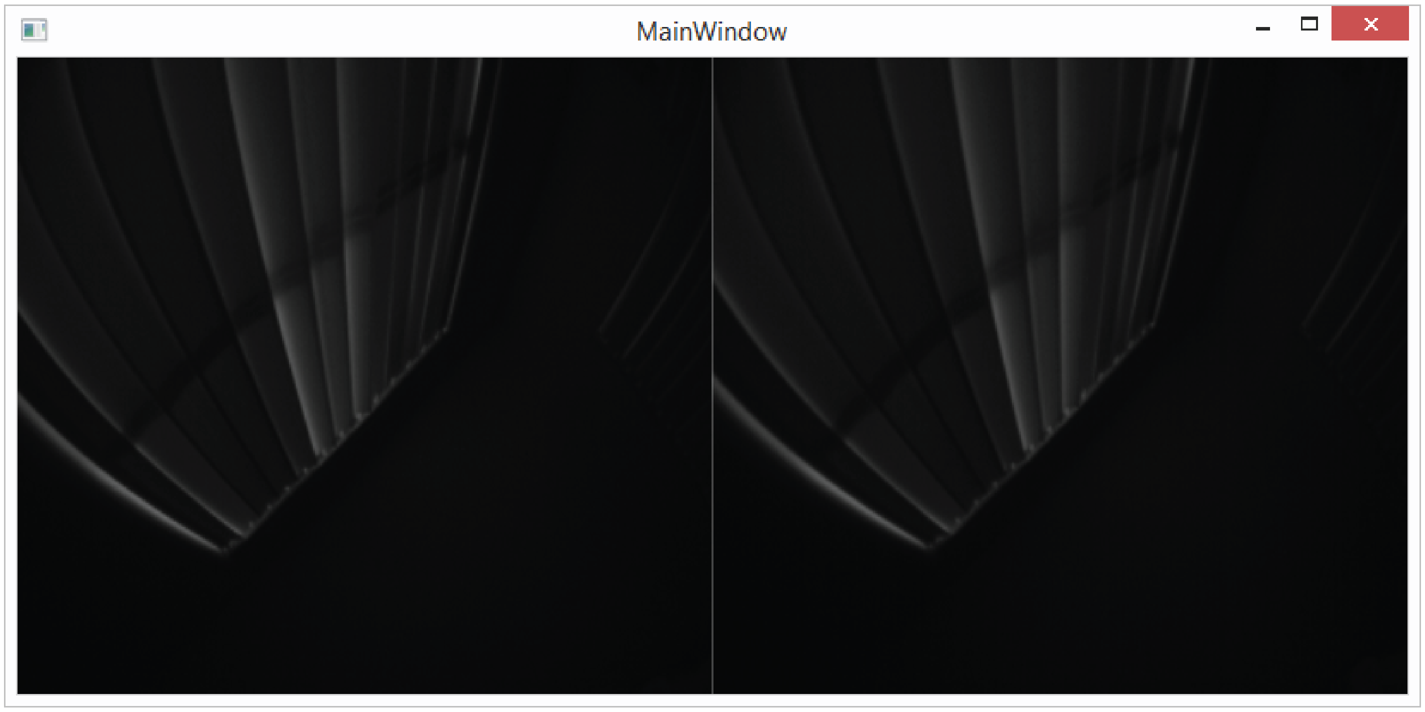
Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using DUODeviceLib;

namespace DUODeviceConsoleDemo
{
    class Program
    {
        static void DUOParameterChanged(DUODevice sender, DUOParameter parameter, object value)
        {
            switch (parameter)
            {
                case DUOParameter.DUO_EXPOSURE:
                    Console.WriteLine("[EXPOSURE] : {0}", value);
                    break;
                case DUOParameter.DUO_GAIN:
                    Console.WriteLine("[GAIN] : {0}", value);
                    break;
                case DUOParameter.DUO_HFLIP:
                    Console.WriteLine("[HFLIP] : {0}", value);
                    break;
                case DUOParameter.DUO_VFLIP:
                    Console.WriteLine("[VFLIP] : {0}", value);
                    break;
                case DUOParameter.DUO_SWAP_CAMERAS:
                    Console.WriteLine("[SWAP CAMERAS] : {0}", value);
                    break;
                case DUOParameter.DUO_LED_PWM:
                    Console.WriteLine("[LED] : {0}", value);
                    break;
                default:
                    break;
            }
        }

        static void DUODeviceStatusChanged(DUODevice sender, bool isRunning)
        {
            if (isRunning)
                Console.WriteLine("[START DUO DEVICE]");
            else
                Console.WriteLine("[STOP DUO DEVICE]");
        }
        static int frameCount = 0;
        static void DUOFrameReceived(DUODevice sender, ref DUOFrame pFrameData)
        {
            frameCount++;
            Console.WriteLine("Frame ID: {0}, Timestamp: {1}", frameCount, pFrameData.timeStamp);
            if (pFrameData.accelerometerPresent)
            {
                Console.WriteLine("Acceleration : ({0}:{1}:{2})", pFrameData.accelData[0], pFrameDat
a.accelData[1], pFrameData.accelData[2]);
                Console.WriteLine("Gyro : ({0}:{1}:{2})", pFrameData.gyroData[0], pFrameData.gyroDat
a[1], pFrameData.gyroData[2]);
                Console.WriteLine("Temperature : {0} C", pFrameData.tempData);
            }
        }
        static void Main(string[] args)
        {
            Console.WriteLine("Press Any Key to Create DUODevice");
            Console.ReadKey();
            DUODevice device = new DUODevice();
            Console.WriteLine(device);
            Console.WriteLine("\nPress Any Key to Start Capture");
            Console.ReadKey();
            device.DUODeviceStatusChanged += DUODeviceStatusChanged;
            device.DUOFrameReceived += DUOFrameReceived;
            device.DUOParameterChanged += DUOParameterChanged;
            device.Start();
            Console.WriteLine("\nPress Any Key to Stop Capture");
            Console.ReadKey();
            device.Stop();
            device.Dispose();
        }
    }
}
```

Sample 03

Using with WPF

In this sample we showcase using the DUO within the WPF (Windows Presentation Foundation) application. We use the DUODeviceLib and DUOFrameViewer to display frames using standard XAML.



MainWindow.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.ComponentModel;
using DUODeviceLib;

namespace DUODeviceWPFDemo
{
    ///

    /// Interaction logic for MainWindow.xaml
    ///

    public partial class MainWindow : Window
    {
        DUODevice duoDevice = null;
        public MainWindow()
        {
            InitializeComponent();
            this.Loaded += new RoutedEventHandler(MainWindow_Loaded);
            this.Closing += new System.ComponentModel.CancelEventHandler(MainWindow_Closing);
        }
        void MainWindow_Closing(object sender, CancelEventArgs e)
        {
            duoDevice.Stop();
        }
        void MainWindow_Loaded(object sender, RoutedEventArgs e)
        {
            duoDevice = new DUODevice();
            frameViewer.Device = duoDevice;
            duoDevice.Start();
        }
    }
}
```

MainWindow.xaml

```
<Window x:Class="DUODeviceWPFDemo.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:duo="clr-namespace:DUODeviceLib.WPF;assembly=DUODeviceLib"
        Title="MainWindow" Height="350" Width="525">
    <Grid>
        <duo:DUOFrameViewer Framerate="60.0"  x:Name="frameViewer"></duo:DUOFrameViewer>
    </Grid>
</Window>
```

Resources

Related

- DUO API
- DUO SDK
- DUO Developers
- DUO Devices
- DUO Downloads