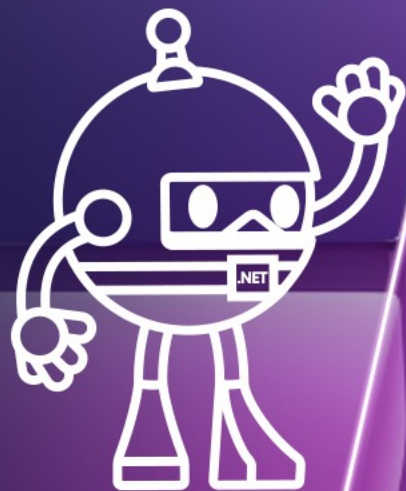


# .NET Conf China 2023

2023/12/16  
09:30 - 18:00

中国 · 北京



中国·北京

# .NET Conf China 2023

## Avalonia

C#在Linux化工业中的利刃

演讲人：李来福

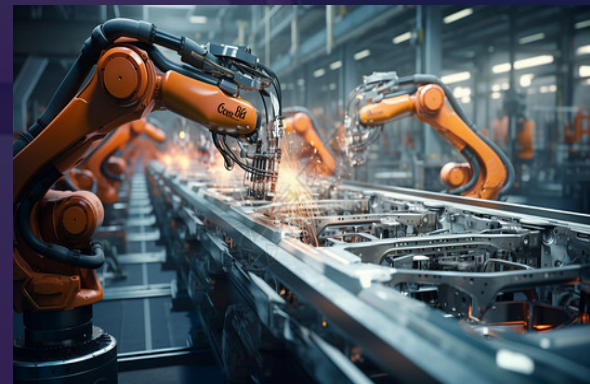
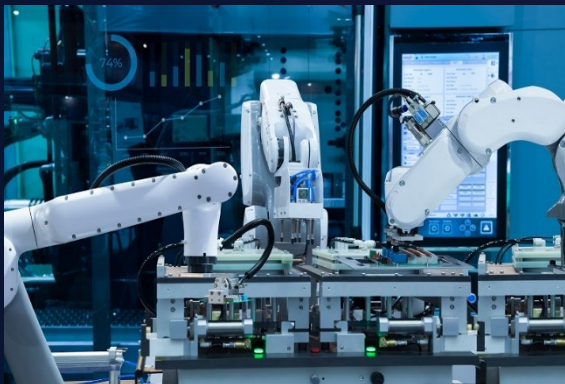
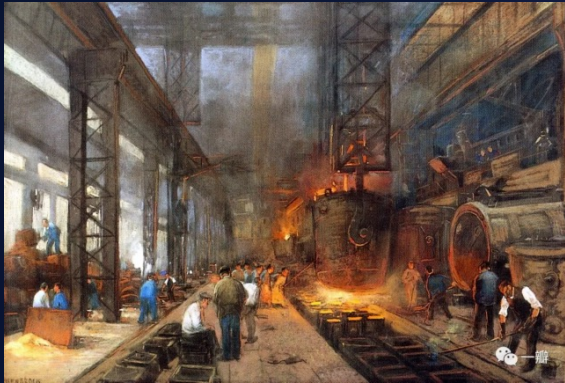




# 01 C#与工业的前生今世

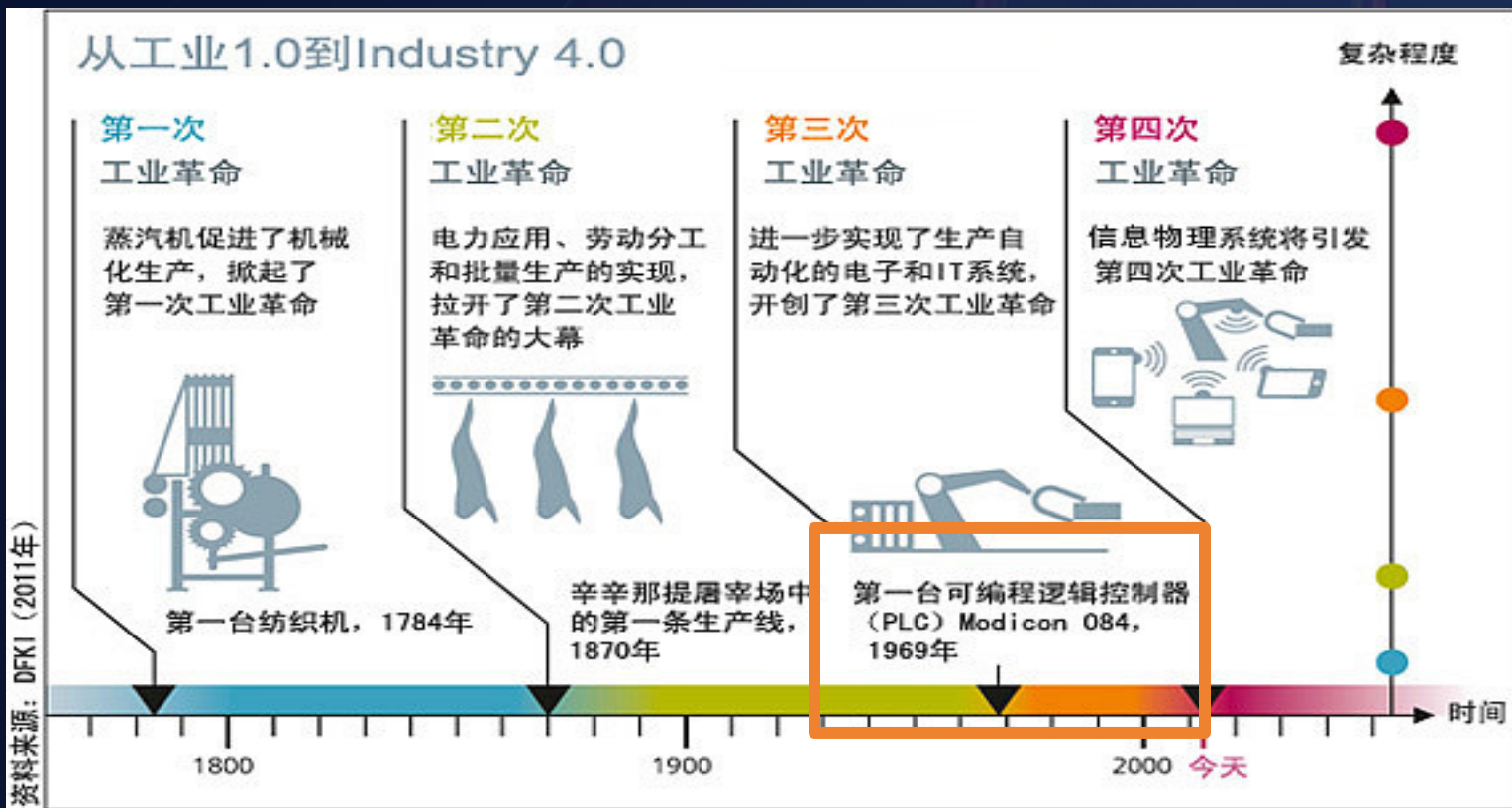


- **工业：**工业是在设备、劳动力、机器、工具以及化学或生物加工或配方的基础上进行的商品生产。





工业的发展历程



• C# + 工业能发生什么呢？



A screenshot of an industrial HMI (Human-Machine Interface) software interface. The interface is dark-themed with blue and white text. At the top, it shows the date '2023年11月14日 16:15:28' and the text 'C# 工业 HMI'.

The interface is divided into several sections:

- 堆垛编号:** Z000000001234
- 识别信息:** 品级码 14012345, 批次码 20231010 TD02 B 10:00:00, 单次数值 20, 产线 TD03, 班次 B.
- 相机故障:** 等级: 严重, 解决办法: ...
- 侧面融合图:** A large black rectangular area, likely a placeholder for a camera feed or image.
- 顶部3D深度图:** Another large black rectangular area, likely a placeholder for a 3D depth map.
- 识别信息表:** A table with columns: 序号, 流水号, 批次码, 品级码, 异常状态, 产线编码.

序号	流水号	批次码	品级码	异常状态	产线编码
1	Z0000010112	20121117	14016	系统置信度内部含	T13C
1	233320107	12111TD03	140178	系统置信度内部含	T13C
1	2333201012	12111TD03	140178	系统置信度内部含	T13C
1	2333201012	2012111TD03	140178	系统置信度内部含	T13C
1	2333201012	2012111TD03	140178	系统置信度内部含	T13C
1	2333201012	2012111TD03	140178	系统置信度内部含	T13C
1	2333201012	2012111TD03	140178	系统置信度内部含	T13C





- 在工业领域的上位机界面开发中，C#以绝对的实力占据第1位，究其原因，我个人认为有2大因素：
  - ✓ Windows桌面操作系统
  - ✓ C#桌面开发的两大利器：WinForm和WPF





## • 目前C#在工业面临的挑战

- ✓ 互联网及云服务大行其道
- ✓ 工业自动化生产的竞争日益激烈
- ✓ 受环境影响，工业急需降本增效，不断压缩成本

利好因素	不利因素
.net core开始支持跨平台（包括Linux）	Winform不支持跨平台
对Linux设备具有较好的支持（串口、CAN口）	WPF不支持跨平台
不断完善的支持跨平台的库	MAUI支持跨平台，但不支持Linux





# 02 新的利刃：Avalonia



- Avalonia: 一款跨平台的UI框架, 使用.NET进行开发, 官网地址: <https://www.avaloniaui.net/>, 它具有以下优点:

- ✓ 完全开源, 遵守MIT协议
- ✓ 平台适配性强, 多系统平台适配
- ✓ 社区活跃, 且具有活跃中文社区
- ✓ 丰富的开源库



- 丰富的成熟案例 ( <https://www.avaloniaui.net/Showcase> ) :





- Avalonia对C#上位软件开发者是友好的。这种友好体现在两方面：
  - ✓ 编辑器友好
    - 跨平台IDE支持
    - UI的实时预览
    - 智能代码提示
    - 良好的调试体验
  - ✓ 编码方式友好



• 编辑器友好

三大编辑器支持



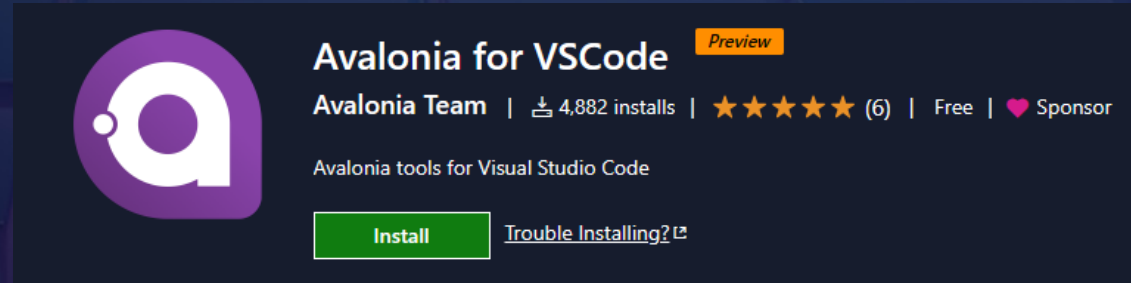
Visual Studio Code



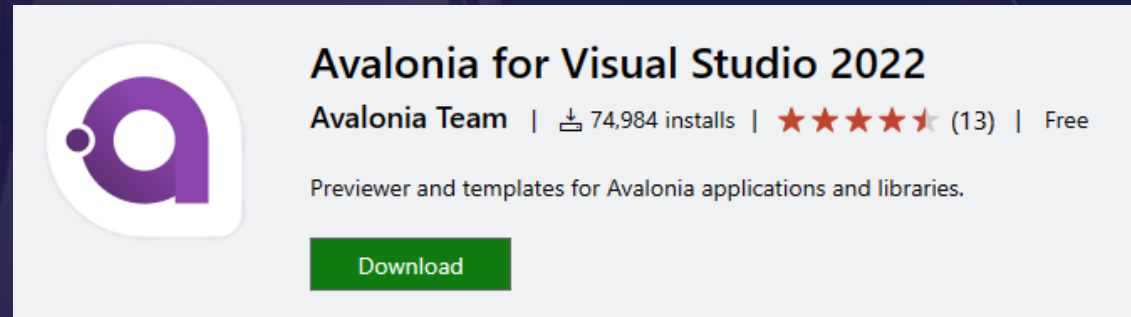
Visual Studio



JetBrains Rider



**Avalonia for VSCode** Preview  
Avalonia Team | 4,882 installs | ★★★★★ (6) | Free | Sponsor  
Avalonia tools for Visual Studio Code  
[Install](#) [Trouble Installing?](#)



**Avalonia for Visual Studio 2022**  
Avalonia Team | 74,984 installs | ★★★★★ (13) | Free  
Previewer and templates for Avalonia applications and libraries.  
[Download](#)



Framework  
**AvaloniaRider**  
★★★★★  
Friedrich von Never  
[Get](#)  
Compatible with Rider



• 编码方式友好



WPF开发者

上手难度：几乎为0



1天即可进行相应开发

- ✓ XAML语法绘制界面
- ✓ C#代码编写逻辑
- ✓ 完整的MVVM支持
- ✓ 绑定、命令、样式的支持
- ✓ 第三方框架的支持 (Prism、CommunityToolkit.Mvvm...)



WinForm开发者

上手难度：较低



很快可进行相应开发

- ✓ XAML语法绘制界面 (需要学习\*)
- ✓ C#代码编写逻辑
- ✓ Code-behind代码支持





# 03 Avalonia工业适用性



- Avalonia工业适用前提：.NET的跨平台支持以及其优越的性能，对各种工业协议的支持



- 工业案例1：使用OPC UA通讯，读取PLC变量，并绘制曲线
  - 跨平台运行：Windows、Linux
  - 使用C#完成OPC UA通讯，与PLC进行交互
  - 枚举PLC侧变量以树形结构展示在UI中，并实现懒加载功能
  - 可定制化选取变量，进行变量曲线绘制







OpcUAClient

服务器地址:   使用密码认证 用户名:  密码:

- DeviceSet
  - DeviceFeatures
  - Sinsegye x86 64-Linux-SM-CNC
    - Resources
      - Application
        - SerialNumber
        - RevisionCounter
        - Manufacturer
        - Model
        - DeviceManual
        - DeviceRevision
        - SoftwareRevision
        - HardwareRevision
        - Programs
        - Tasks
      - GlobalVars
        - GVL
          - t1
          - t2
          - t3
          - t4**
      - NetworkSet
      - DeviceTopology
      - Server

开始监听 结束监听

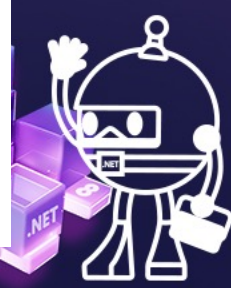
重点值监视区:

曲线监听值列表:

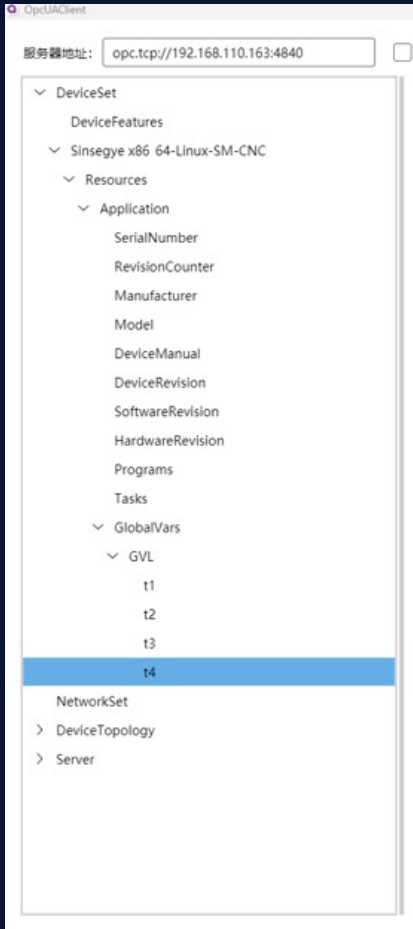
```
t1(ns=4;s=|var|Sinsegye x86  
t2(ns=4;s=|var|Sinsegye x86  
t3(ns=4;s=|var|Sinsegye x86  
t4(ns=4;s=|var|Sinsegye x86
```

10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

15:54:15 15:54:20 15:54:25 15:54:30 15:54:35 15:54:40 15:54:45 15:54:50 15:54:55



## • 树形图及右键菜单的制作



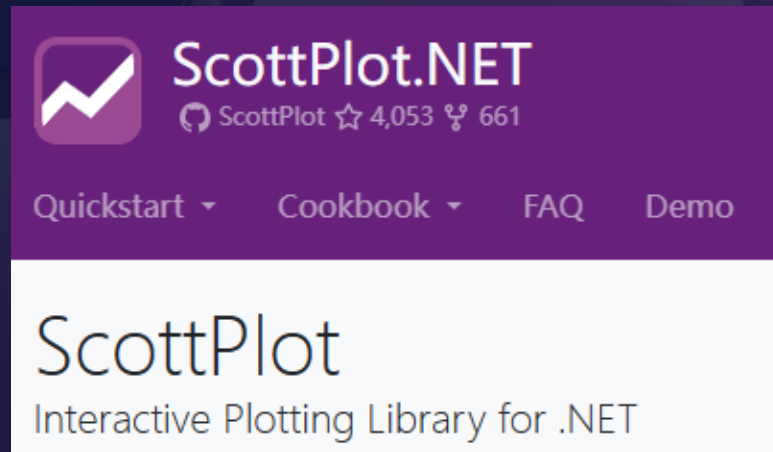
```
<TreeView Grid.Row="1" Grid.Column="0" BorderThickness=" [all] 2" BorderBrush="#CCCCCC" Items="{Binding Path= (LineListenViewModel).NodeTree}">
  <TreeView.Styles>
    <Style Selector="TreeViewItem">
      <Setter Property="ContextMenu">
        <ContextMenu>
          <MenuItem Header="添加订阅"
            Command="{ReflectionBinding Path= $parent[TreeView].DataContext.AddSubscribe}"
            CommandParameter="{Binding DataContext=LineListenViewModel }" />
          <MenuItem Header="移除订阅"
            Command="{ReflectionBinding Path= $parent[TreeView].DataContext.RemoveSubscribe}"
            CommandParameter="{Binding DataContext=LineListenViewModel }" />
          <MenuItem Header="添加到重点监听"
            Command="{ReflectionBinding Path= $parent[TreeView].DataContext.AddToImportantListen}"
            CommandParameter="{Binding DataContext=LineListenViewModel }" />
          <MenuItem Header="移除重点监听"
            Command="{ReflectionBinding Path= $parent[TreeView].DataContext.RemoveImportantListen}"
            CommandParameter="{Binding DataContext=LineListenViewModel }" />
        </ContextMenu>
      </Setter>
    </Style>
  </TreeView.Styles>
  <TreeView.ItemTemplate>
    <TreeDataTemplate ItemsSource="{Binding Path= (NodeModel).Child}">
      <StackPanel Orientation="Horizontal">
        <Panel Width="12" Margin=" [lr] 12, [tb] 0">
          <ToggleButton Classes="ExpandCollapseChevron" Focusable="False"
            IsVisible="{Binding Path= (NodeModel).HasChild}"
            IsChecked="{Binding Path= $parent[TreeViewItem].IsExpanded}"
            Command="{ReflectionBinding Path= $parent[TreeView].DataContext.Expand}"
            CommandParameter="{Binding DataContext=NodeModel }" />
        </Panel>
        <TextBlock Text="{Binding Path= (NodeModel).Name}" />
      </StackPanel>
    </TreeDataTemplate>
  </TreeView.ItemTemplate>
</TreeView>
```



- 曲线图的绘制

### Charts & Plots & Diagrams

- [GoDiagram](#) - A .NET library for rapidly building interactive diagrams.
- [LiveCharts2](#) - Simple, flexible, interactive & powerful charts, maps and gauges for .Net.
- [Microcharts](#) - A simple Avalonia only port of the Microcharts library.
- [OxyPlot](#) - A cross-platform plotting library for .NET.
- [ScottPlot](#) - Interactive Plotting Library for .NET
- [StockPlot](#) - A Stcok Markets Technical Analysis library for AvaloniaUI, based on ScottPlot, supporting live data stream.





- 曲线图的绘制

```
<liv:CartesianChart Grid.Row="1" Grid.Column="2" Margin=" [all:] 50" Series="{Binding Path= (LineListenViewModel). CartesianChartLineSeries}"
    YAxes="{Binding Path= (LineListenViewModel). CartesianChartYAxis}"
    TooltipTextPaint="{Binding Path= (LineListenViewModel). TooltipTextPaint}" AnimationsSpeed="00:00:00.500"
    XAxes="{Binding Path= (LineListenViewModel). CartesianLineChartXAxis}" />
```

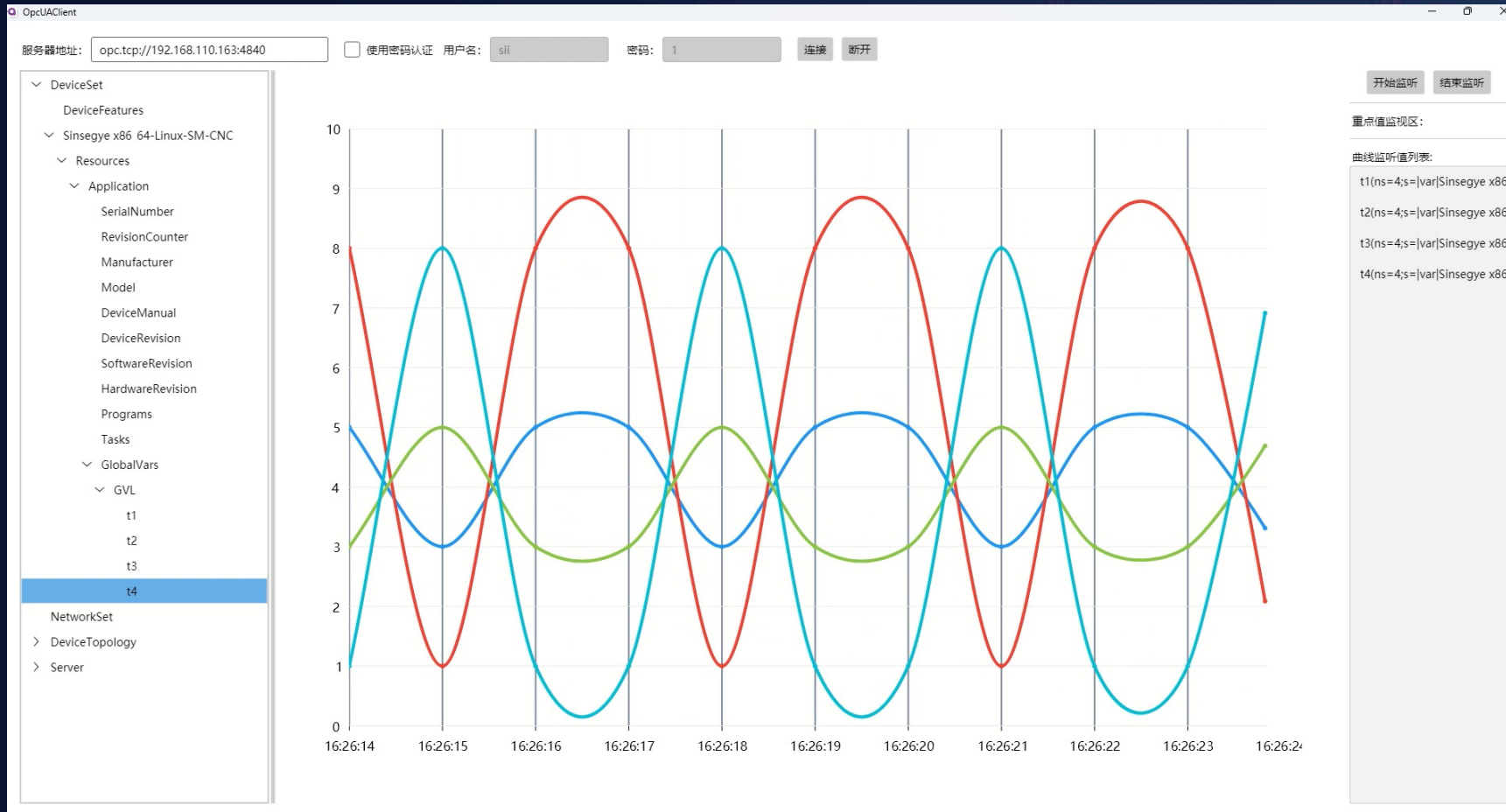
```
CartesianLineChartXAxis = new[]
{
    new Axis
    {
        MinLimit = DateTime.Now.Ticks,
        MinStep = TimeSpan.FromSeconds(1).Ticks,
        UnitWidth = TimeSpan.FromSeconds(1).Ticks,
        SeparatorsPaint = new SolidColorPaint(SKColors.LightSlateGray) { StrokeThickness = 2 },
        Labeler = value:double => new DateTime((long)value).ToString(format: "HH:mm:ss"),
        TicksPaint = new SolidColorPaint(SKColors.Black),
        SeparatorsAtCenter = true,
        LabelsPaint = new SolidColorPaint()
        {
            Color = SKColors.Black,
            SKTypeface = SKFontManager.Default.MatchCharacter('汉'),
        }
    }
};
```

```
CartesianChartYAxis = new[]
{
    new Axis
    {
        LabelsPaint = new SolidColorPaint()
        {
            Color = SKColors.Black,
            SKTypeface = SKFontManager.Default.MatchCharacter('汉'),
        },
        Labeler = value:double => Convert.ToInt32(value).ToString(CultureInfo.CurrentCulture),
        MinStep = 1,
        MaxLimit = 10,
        MinLimit = 0
    }
};
```

◆ 中文字体的设置: SKTypeface, 解决汉字乱码问题



• 成果演示



- 工业案例2：接入工业实时监控画面
  - 跨平台运行：Windows、Linux
  - 可接入网络视频流（Rtsp、Rtmp等）
  - 可连接工业相机，进行取图、视频展示等功能





## • 使用Avalonia接入网络视频流

Mono, .NET Framework and .NET Core runtimes are supported.

- Xamarin.Android
- Xamarin.iOS
- Xamarin.tvOS
- Xamarin.Mac (Cocoa)
- Windows Classic (WPF, WinForms, GTK)
- Windows Universal (UWP for Desktop, mobile and Xbox)
- Linux including desktop, server and Raspberry Pi (GTK)
- Xamarin.Forms
- Uno Platform (UWP, Android, iOS)
- **Avalonia (Windows, macOS, Linux)**
- .NET Standard 1.1 and 2.0
- .NET Framework 4.0 and later
- .NET Core (including ASP.NET Core)
- [Unity3D](#)
  - Windows Classic
  - Android



◆ Windows使用：安装Nuget包：VideoLAN.LibVLC.Windows

◆ Linux使用：安装系统依赖libvlc-dev

```
sudo apt install libvlc-dev
```



- 使用Avalonia接入网络视频流

```
MainWindow.axaml | MainWindow.axaml.cs | MainWindowViewModel.cs

<Window xmlns="https://github.com/avaloniaui"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:vm="using:VideoStreamDemo.ViewModels"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"
  x:Class="VideoStreamDemo.Views.MainWindow"
  x:DataType="vm:MainWindowViewModel"
  xmlns:vlc="clr-namespace:LibVLCSharp.Avalonia;assembly=LibVLCSharp.Avalonia"
  Icon="/Assets/avalonia-logo.ico"
  Title="VideoStreamDemo">

  <Design.DataContext>
    <vm:MainWindowViewModel />
  </Design.DataContext>

  <Grid RowDefinitions="10,*,10,Auto">
    <vlc:VideoView Grid.Row="1"
      MediaPlayer="{Binding Path=(MainWindowViewModel).MediaPlayer}"
      HorizontalAlignment="Stretch"
      VerticalAlignment="Stretch" />
    <Button Grid.Row="3"
      HorizontalAlignment="Center"
      HorizontalContentAlignment="Center"
      Width="70"
      Content="播放"
      Command="{Binding Path=(MainWindowViewModel).Play}" />
  </Grid>
</Window>
```

```
MainWindow.axaml | MainWindow.axaml.cs | MainWindowViewModel.cs

using System;
using LibVLCSharp.Shared;

namespace VideoStreamDemo.ViewModels;

public class MainWindowViewModel : ViewModelBase
{
    private readonly LibVLC _libVlc = new();

    private const string RtmpUrl = "rtmp://ns8.indexforce.com/home/mystream";

    public MediaPlayer MediaPlayer { get; }

    public MainWindowViewModel()
    {
        MediaPlayer = new MediaPlayer(_libVlc);
    }

    public void Play()
    {
        using var media = new MediaC_libVlc, new Uri(RtmpUrl));
        MediaPlayer.Play(media);
    }
}
```

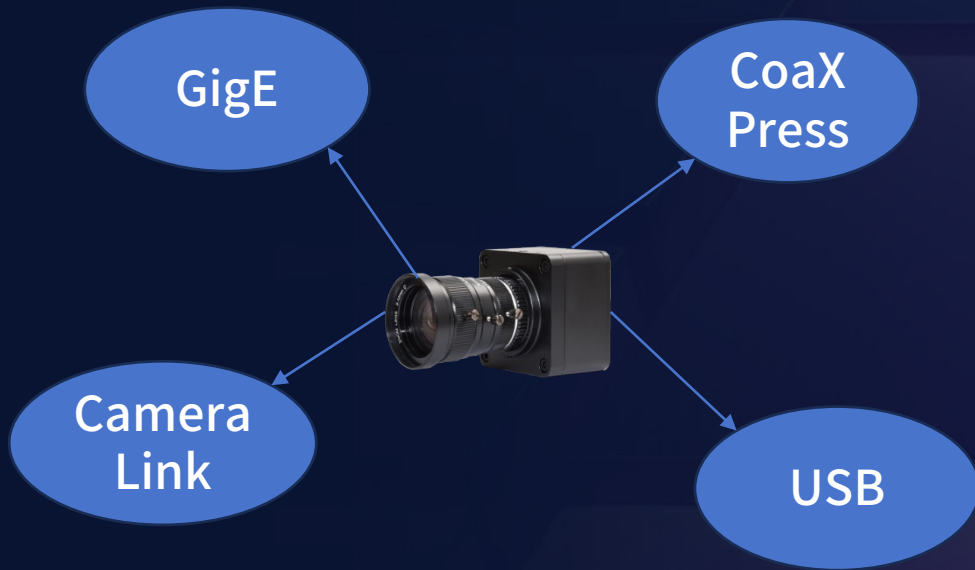


- 使用Avalonia接入网络视频流





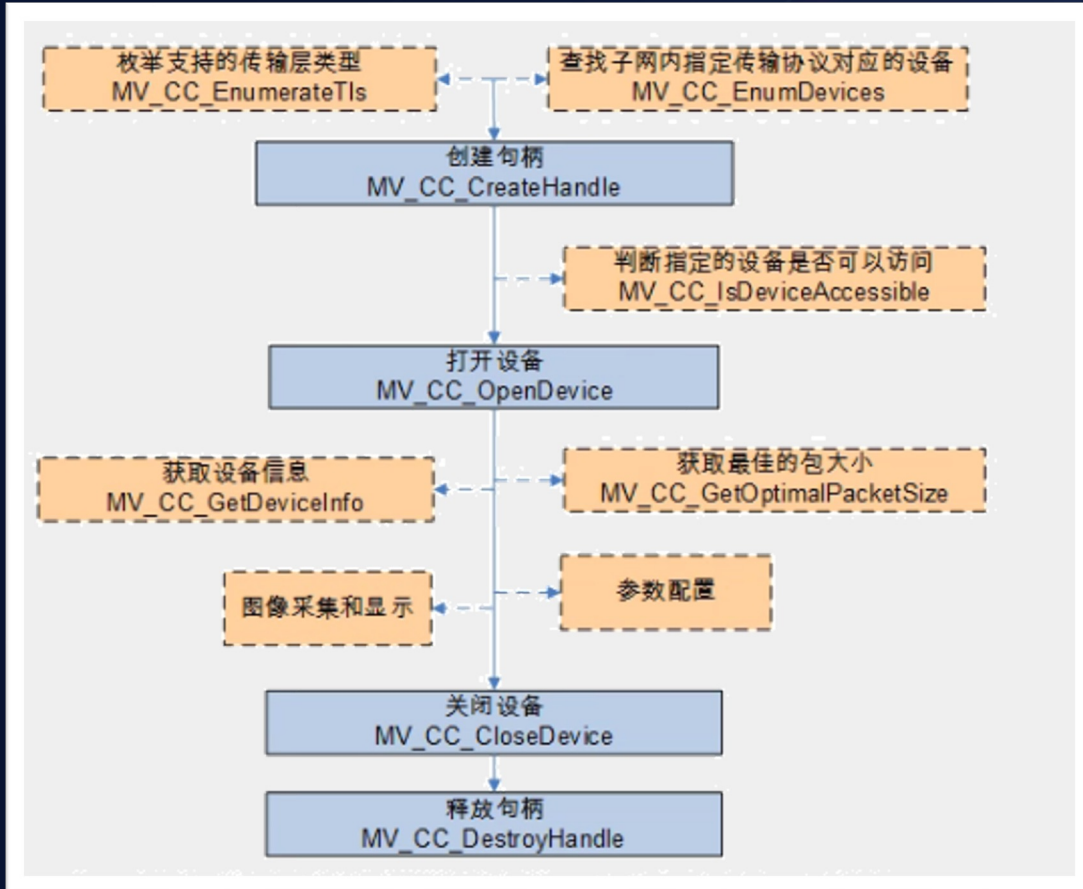
- 使用Avalonia连接工业相机



类型	Camera Link	CoaXPress 2.0	GigE	10Gige	100Gige	USB4 Gen3*2	USB4 2.0
单线速率 (Gbps)	2	1.25-12.5	1	10	100	40	80
最大速率 (Gbps)	6	50(4路 CXP-12)	1	10	100	40	80
最大速率(GB/s)	0.75	6.25(4路 CXP-12)	0.125	1.25	12.5	5	10



• 使用Avalonia连接工业相机



工业相机Linux SDK开发指南(C)

说明 首页 环境配置 更新记录 编程引导 接口函数 结构体定义 相机参数节点表 示例程序 错误码 常见问题

### 概述

本手册适用于 Linux 系统的SDK(C)。工业相机SDK提供了一套通用的编程接口，兼容 GigE Vision 和 USB3 Vision 两种标准协议，隐藏了不同接口之间的大部分差异，允许从一个相机移植到另一个相机而无须修改大量的代码。工业相机SDK的作用是实现对相机的控制，包括工业相机SDK接口、示例程序、SDK驱动。工业相机SDK接口主要有图像采集、相机参数设置和图像处理等接口。

### 开始使用

更新记录	更新记录显示工业相机SDK不同版本的更新摘要。
环境配置	SDK编程环境配置，包括网络配置、驱动安装等。
编程引导	相机连接流程和取图方式介绍。
工业相机SDK接口	包含工业相机SDK接口参考文档，您可以和示例程序、编程引导一起使用以入门。工业相机SDK接口文档可以直接参见接口函数一栏。
相机参数节点表	包含了相机的常用节点。列表中参数节点表仅供参考，不是每个相机都支持列表中所有参数。您可以将相机连接到PC并在MVS软件中打开它，相机支持的参数以MVS参数列表中的为准。
示例程序	常见用例和接口基本的使用方法。
错误码	所有工业相机SDK接口返回的错误码，有助于程序问题排查。
常见问题	介绍了SDK问题的排查思路和其他常见问题。

### 支持系统

Linux 32位/64位:

- Ubuntu 14.04 (32/64)、Ubuntu 16.04 (32/64)、Ubuntu 18.04 (32/64)、Ubuntu 20.04 (32/64)、Redhat7 (64)、Centos7 (32/64)、gcc/g++版本要求4.6.3及以上
- Linux ARM可支持的板卡类型: NVIDIA TX2(Ubuntu16.04)、RaspberryPi3.0+(NOOBS\_2.8.2)、ODROID-XU4(Ubuntu16.04/Ubuntu18.04)
- X86架构对于Gige相机支持驱动

### 注意事项

- 本手册中提供的代码段仅作为示例代码提供。不正确的代码可能会导致您的相机功能与预期不同，并可能危及您的应用程序。为确保代码段在您的应用程序中正常运行，您必须对其进行调整以满足您的特定需求，并且在使用前对其进行彻底测试。
- 不是所有的接口所有的相机都支持，以实际相机支持的功能为准。



- 使用Avalonia连接工业相机中的难点
  - 一般的工业相机SDK，如海康，在Linux中只有C的SDK，没有C#的
  - 工业相机的使用，一般需要使用句柄





- 难点1: Linux中C SDK的使用

P/Invoke  
(平台调用)

DllImportAttribute

DllImportAttribute  
(.net 7.0引入)

```
using System;
using System.Runtime.InteropServices;

public class Program
{
    // Import user32.dll (containing the function we need) and define
    // the method corresponding to the native function.
    [DllImport("user32.dll", CharSet = CharSet.Unicode, SetLastError = true)]
    private static extern int MessageBox(IntPtr hWnd, string lpText, string lpCaption, uint uType);

    public static void Main(string[] args)
    {
        // Invoke the function as a regular managed method.
        MessageBox(IntPtr.Zero, "Command-line message box", "Attention!", 0);
    }
}
```

```
[DllImport(
    "nativelib",
    EntryPoint = "to_lower",
    StringMarshalling = StringMarshalling.Utf16)]
internal static partial string ToLower(string str);
```



```
/// 创建句柄 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_CreateHandle(out IntPtr handle, IntPtr pstDevInfo);

/// 销毁句柄 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_DestroyHandle(IntPtr handle);

/// 打开设备 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_OpenDevice(IntPtr handle, uint nAccessMode, ushort nSwitchoverKey);

/// 关闭设备 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_CloseDevice(IntPtr handle);

/// 开始采集 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_StartGrabbing(IntPtr handle);

/// 停止采集 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_StopGrabbing(IntPtr handle);

/// 显示图像 ...
[DllImport(dllName: "Lib/libMvCameraControl")]
public static extern int MV_CC_Display(IntPtr handle, IntPtr windowHandle);
```

- DllImport是跨平台的，支持搜索.dll、.so和.dylib
- 引入动态链接库时，可以不添加后缀名，可以自动添加搜索
- 库也可以添加到环境变量中，程序也会搜索环境变量中的动态链接库
- C/C++中声明的类型，与C#不同，需要注意类型的对应
- 如需传输指针，可使用IntPtr代表



- 难点2: Avalonia中Handle的使用

◆ MV\_CC\_Display()

```
MV_CAMCTRL_API int __stdcall MV_CC_Display (IN void * handle,  
                                             void * hWnd  
                                             )
```

显示图像，注册显示窗口，内部自动显示（与 MV\_CC\_GetImageBuffer() 不能同时使用）

**参数**


- handle [IN] 句柄
- hWnd [IN] 显示窗口句柄

**返回**

成功，返回MV\_OK；失败，返回错误码

```
/// 显示图像 ...
```

```
[DllImport("Lib/libMvCameraControl")]
```

1 用法  lilaifu

```
public static extern int MV_CC_Display(IntPtr handle, IntPtr windowHandle);
```





- 难点2: Avalonia中Handle的使用

```
public class HkCameraControl : NativeControlHost
{
    public static readonly StyledProperty<IntPtr> HandleProperty = AvaloniaProperty.Register<HkCameraControl, IntPtr>(
        name: "Handle", defaultValue: default, inherits: false, BindingMode.OneWayToSource);

    /// <summary>
    /// 窗口句柄
    /// </summary>
    public IntPtr Handle
    {
        get => GetValue(HandleProperty);
        set => SetValue(HandleProperty, value);
    }

    protected override IPlatformHandle CreateNativeControlCore(IPlatformHandle parent)
    {
        var handle = base.CreateNativeControlCore(parent);
        this.Handle = handle.Handle;
        return handle;
    }
}
```





```
mc:Ignorable="d" d:DesignWidth="800" d:DesignHeight="450"
xmlns:controls="clr-namespace:CameraDemo.Controls"
x:Class="CameraDemo.Views.MainWindow"
x:DataType="vm:MainWindowViewModel"
FontFamily="微软雅黑"
Icon="/Assets/avalonia-logo.ico" WindowState="Maximized"
Title="CameraDemo" Name="window">

<Design.DataContext ...>
<Grid ColumnDefinitions="Auto,*" Margin=" [all] 20">
  <Border ...>

  <DockPanel Grid.Column="1" Margin=" [r] 20, [tb] 0">
    <StackPanel DockPanel.Dock="Top" Orientation="Horizontal" Margin=" [l] 0, [t] 0, [r] 0, [b] 10">
      <Button Content="开始采集" Command="{Binding Path= (MainWindowViewModel).StartPlay}" IsVisible="{Binding Path= (MainV
      <Button Content="停止采集" Command="{Binding Path= (MainWindowViewModel).Stop}"
        IsVisible="{Binding Path= (MainWindowViewModel).IsPlaying}" />
      <Button Content="抓取图片" Margin=" [l] 10, [t] 0, [r] 0, [b] 0" Command="{Binding Path= (MainWindowViewModel).GrabPicture}"
        IsVisible="{Binding Path= (MainWindowViewModel).IsPlaying}" />
    </StackPanel>
    <controls:HkCameraControl HorizontalAlignment="Stretch" VerticalAlignment="Stretch"
      Handle="{Binding Path= (MainWindowViewModel).Handle}" />
  </DockPanel>

public void StartPlay()
{
  var isOk:int = HkCameraControl.MV_CC_CreateHandle(out _cameraHandel, pstDevInfo:_cameraIntPtrDictionary[SelectedCamera]);
  Console.WriteLine($"创建句柄{isOk:x8}");

  isOk = HkCameraControl.MV_CC_OpenDevice(_cameraHandel, nAccessMode:1, nSwitchoverKey:0);
  Console.WriteLine($"打开设备{isOk:x8}");

  isOk = HkCameraControl.MV_CC_RegisterImageCallBackEx(handle:_cameraHandel, _receiveImageData, userCustom:new IntPtr());
  Console.WriteLine($"注册取图回调{isOk:x8}");

  isOk = HkCameraControl.MV_CC_StartGrabbing(_cameraHandel);
  Console.WriteLine($"开始采集{isOk:x8}");

  isOk = HkCameraControl.MV_CC_Display(handle:_cameraHandel, windowHandle:Handle);
  Console.WriteLine($"开始播放{isOk:x8}");

  this.IsPlaying = true;

  GetFpsValue();
}
```

```
public void StartPlay()
{
  var isOk:int = HkCameraControl.MV_CC_CreateHandle(out _cameraHandel, pstDevInfo:_cameraIntPtrDictionary[SelectedCamera]);
  Console.WriteLine($"创建句柄{isOk:x8}");

  isOk = HkCameraControl.MV_CC_OpenDevice(_cameraHandel, nAccessMode:1, nSwitchoverKey:0);
  Console.WriteLine($"打开设备{isOk:x8}");

  isOk = HkCameraControl.MV_CC_RegisterImageCallBackEx(handle:_cameraHandel, _receiveImageData, userCustom:new IntPtr());
  Console.WriteLine($"注册取图回调{isOk:x8}");

  isOk = HkCameraControl.MV_CC_StartGrabbing(_cameraHandel);
  Console.WriteLine($"开始采集{isOk:x8}");

  isOk = HkCameraControl.MV_CC_Display(handle:_cameraHandel, windowHandle:Handle);
  Console.WriteLine($"开始播放{isOk:x8}");

  this.IsPlaying = true;

  GetFpsValue();
}
```



# 感谢聆听!



项目开源地址: <https://github.com/Sinsegye-CSharp/AvaloniaSamples>

