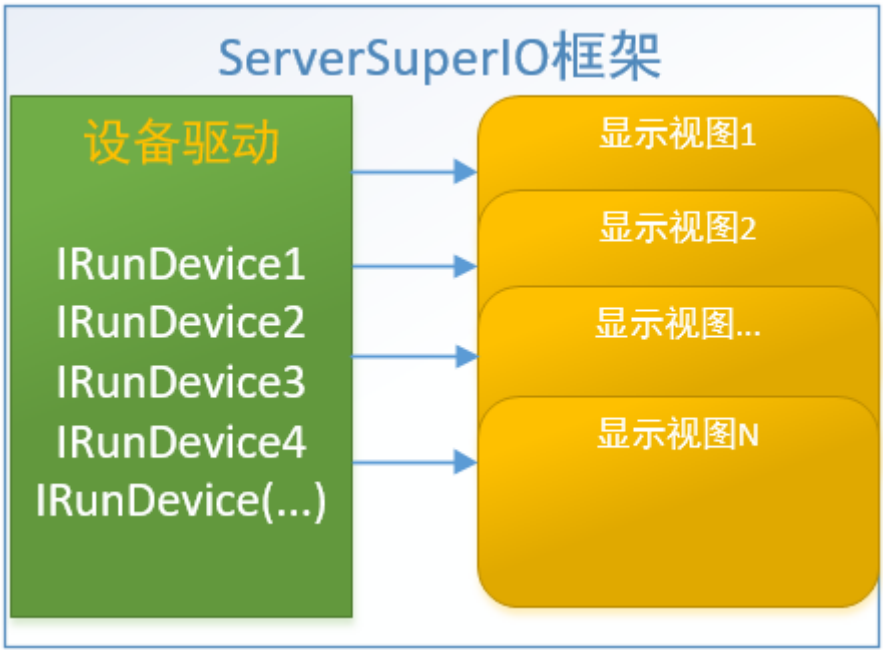
目　　录

官方网站：http://www.bmpj.net

13．自定义视图显示接口开发，满足不同的显示需求

13.1 概述

如果 ServerSuperIO 部署在服务端，那么完全没有必要在 SSIO 视图接口上进行视图显示开发，可以用 WEB 端来做。如果 ServerSuperIO 部署在 PC 机端或者嵌入式主机，并且带现场显示屏幕，那么就需要有实时显示的界面，以及满足现场用户的不同需求。针对这种情况，ServerSuperIO 内部提供了视图显示接口，设备驱动提取完成数据后，可以把数据实时传输给视图接口，可以立即显示，也可以先缓存数据，定时进行显示；可以几个设备驱动的数据显示在一个视图，也可以全部设备驱动的数据显示在一个视图。示意如下图：



13.2 视图显示接口

视图抽象类 GraphicsShow 继承自 IGraphicsShow 接口，进行二次开发可以继承 GraphicsShow 抽象类。在 ServerSuperIO 上开发好视图接口后，完全可以在二次开发套件下挂载和运行（二次开发套件下载）。在进行二次开时，有几点特别需要注意：

1. ShowGraphics(IWin32Window windows)接口，windows 参数是当前显示

视图的父窗体，如果父窗体设置了 IsMdiContainer=true,那么可以设置当前显示视图 showForm.MdiParent = (Form)windows。

2. UpdateDevice(string devid, object obj)实时数据更新接口，如果二次开发的视图的实例增加到当前服务实例中，那么设备驱动会通过 OnDeviceObjectChanged 数据改变事件通知视图的 UpdateDevice 接口进行数据更新。OnDeviceObjectChanged 何时触发，完全由二次开发者自己把控。
3. OnGraphicsShowClosed 视图关闭事件接口，当前视图进行关闭时，一定要触发这个事件，ServerSuperIO 会自动释放资源，并把当前视图实例从服务实例中销毁，以便下次能够正常显示。
4. MouseRightContextMenu 上下文菜单显示接口，如果右键单击当前视图显示的某个设备视图时，可以调用这个上下文菜单接口，会调用设备驱动 IRunDevice 接口的 ShowContextMenu 函数，可以在这个函数中展示上下文菜单。这个功能特别有用，因为针对不同类型的设备驱动功能不一样，可以通过上下文菜单自定义该类型设备的功能特性。

GraphicsShow 接口代码定义如下：

```
public interface IGraphicsShow : IPlugin
    {
        /// <summary>
        ///     服务Key,要求唯一
        /// </summary>
        string ShowKey { get;}

        /// <summary>
        ///     服务名称
        /// </summary>
        string ShowName { get; }

        /// <summary>
        /// 显示窗体
        /// </summary>
        /// <param name="windows"></param>
        void ShowGraphics(IWin32Window windows);

        /// <summary>
        ///
```

```
        /// </summary>
        void CloseGraphics();
        /// <summary>
        ///     更新设备
        /// </summary>
        /// <param name="devCode">设备ID</param>
        /// <param name="obj">设备对象</param>
        void UpdateDevice(string devCode, object obj);

        /// <summary>
        ///     移除设备
        /// </summary>
        /// <param name="devCode">设备ID</param>
        void RemoveDevice(string devCode);

        /// <summary>
        ///     关闭窗体事件时发生
        /// </summary>
        event GraphicsShowClosedHandler GraphicsShowClosed;

        /// <summary>
        ///     单击右键
        /// </summary>
        event MouseRightContextMenuHandler MouseRightContextMenu;

        /// <summary>
        /// 是否被释放
        /// </summary>
        bool IsDisposed { get; }
    }
```

13.3 设备驱动开发及注意事项

1. 给视图实例传递数据信息，通过 OnDeviceObjectChanged 事件。代码如下：

```
List<string> list = new List<string>();
list.Add(_devicePara.DeviceCode);
list.Add(_devicePara.DeviceName);
list.Add(_deviceDyn.Dyn.Flow.ToString());
list.Add(_deviceDyn.Dyn.Signal.ToString());
OnDeviceObjectChanged(list.ToArray());
```

2. 显示上下文菜单，视图接口的 MouseRightContextMenu 会调用设备驱动的

ShowContextMenu 接口函数。代码如下：

```
public override void ShowContextMenu()
        {

this._contextMenuComponent.ContextMenuStrip.Show(Cursor.Position);
        }
```

13.4 运行效果