目　　录

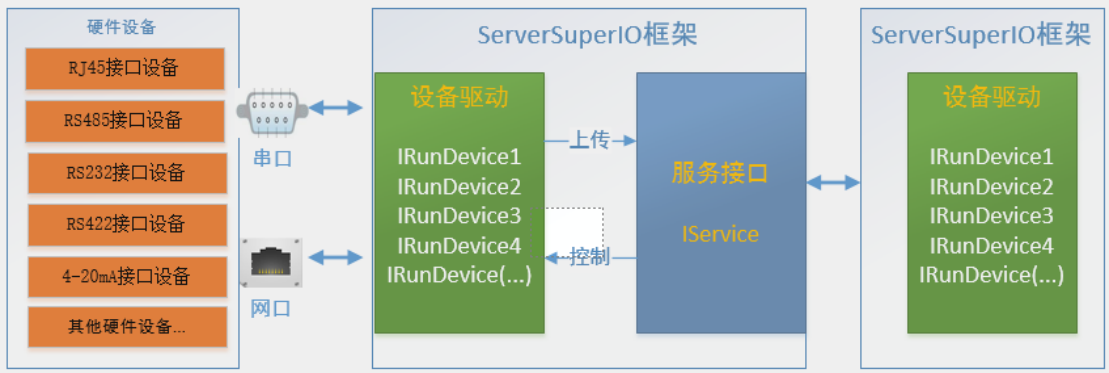
官方网站：http://www.bmpj.net

12.服务接口的开发，以及与云端双向交互

12.1 概述

服务接口 IService 是除设备驱动接口 IRunDevice 外的特殊应用场景，例如：短信报警、LED 输出、模拟量的采集、OPC 服务端/客户端、数据的级联转发、以及其他的特殊应用。是基于设备驱动接口 IRunDevice 的有力扩展，让框架的功能更丰富和强大。

这篇文章主要介绍数据的转发和控制服务，其他的服务开发与此类似。如果基于 ServerSuperIO 服务接口的开发具备数据转发和控制能力，那么在物联网建设中将具备数据的无限级联传递与控制能力，使用一套框架形成一套解决方案。示意图如下：



12.2 设备链接器

IRunDevice 设备驱动接口继承了服务的 IServiceConnectorDevice 接口，设备驱动会实现 RunServiceConnector 接口函数，代表可以接收或运行来自服务实例传递过来的数据信息。代码定义如下：

```
public interface IServiceConnectorDevice
    {
        /// <summary>
        /// 支行设备连接器
        /// </summary>
        /// <param name="fromService"></param>
        /// <param name="toDevice"></param>
        /// <returns></returns>
```

```
        object RunServiceConnector(IFromService fromService, IServiceToDevice toDevice);
    }
```

12.3 服务链接器

IService 接口继承自 IServiceConnector 服务链接器，代表服务具备两大职能：1、向设备驱动发送命令或信息。2、接收设备驱动处理命令或信息的结果。代码如下：

```
public interface IServiceConnector
    {
        /// <summary>
        /// 设备连接器回调函数，在这里写回调的处理结果
        /// </summary>
        /// <param name="obj"></param>
        void ServiceConnectorCallback(object obj);

        /// <summary>
        /// 如果执行方出现异常，则返回给这个函数结果
        /// </summary>
        /// <param name="ex"></param>
        void ServiceConnectorCallbackError(Exception ex);

        /// <summary>
        /// 连接器事件，发起端
        /// </summary>
        event ServiceConnectorHandler ServiceConnector;

        /// <summary>
        /// 确发事件接口
        /// </summary>
        /// <param name="fromService"></param>
        /// <param name="toDevice"></param>
        void OnServiceConnector(IFromService fromService, IServiceToDevice toDevice);
    }
```

12.4 场景假设

设备驱动实时采集传感器的数据信息，然后通过 OnDeviceObjectChanged 事件接口把数据信息传递给服务实例 UpdateDevice 接口函数，然后进行数据的缓存。开启服务线程，连接云端服务并且定时上传传感器的数据信息。云端服务下

发控制指令 command:control:1（注：命令控制 1 号编码设备），然后设备驱动把命令下发给传感器，最终把结果返回给服务实例。

12.5 设备驱动开发及注意事项

1. 给服务实例传递数据信息，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.接收服务实例传递的命令和信息。代码如下：

```
public override object RunServiceConnector(IFromService fromService, IServiceToDevice toDevice)
        {
            Console.WriteLine(this.DeviceParameter.DeviceName+",接收到云端指令:"+toDevice.Text);
            return this.DeviceParameter.DeviceName+",执行完成";
        }
```

12.6 服务实例开发及注意事项

服务实例主要的职能是：缓存数据、连接云端服务并定时发送数据、接收云端命令信息并传递给设备驱动、接收驱动执行命令信息的结果。

1. 缓存数据，代码如下：

```
public override void UpdateDevice(string devCode, object obj)
        {
            lock (_SyncObject)
            {
                if (obj != null)
                {
                    if (obj is string[])
                    {
                        string[] arr = (string[])obj;
```

```
                                    //OnServiceLog(String.Format("服务：{0} 接收到
'{1}'的数据>>{2},{3}", ServiceName, arr[1], arr[2], arr[3]));
                                    if (arr.Length >= 2)
                                    {
                                        if  (this._Cache.ContainsKey(devCode))  // 判
断 ID
                                        {
                                            this._Cache[devCode] = arr;
                                        }
                                        else
                                        {
                                            this._Cache.Add(devCode, arr);
                                        }
                                    }
                                }
                            }
                        }
```

2. 连接云端服务并定时发送数据，代码如下：

```
    private void Target_Service()
            {
                while (_IsRun)
                {
                    try
                    {
                        if (_tcpClient != null)
                        {
                            lock (_SyncObject)
                            {
                                string content = String.Empty;
                                foreach (KeyValuePair<string, string[]> kv in
_Cache)
                                {
```

```
                                content += String.Join(",", kv.Value) +
Environment.NewLine;
                            }

                            if (!String.IsNullOrEmpty(content))
                            {
                                byte[] data =
System.Text.Encoding.ASCII.GetBytes(content);
                                this.OnSend(data);
                            }
                        }
                    }
                    else
                    {
                        this.ConnectServer();
                    }
                }
                catch (SocketException ex)
                {
                    this.CloseSocket();
                    OnServiceLog(ex.Message);
                }
                catch (Exception ex)
                {
                    OnServiceLog(ex.Message);
                }
                finally
                {
                    System.Threading.Thread.Sleep(2000);
                }
            }
        }
```

3. 接收云端命令信息并传递给设备驱动

```
private void ReceiveCallback(IAsyncResult ar)
        {
            TcpClient socket = (TcpClient)ar.AsyncState;
            try
            {
                if (socket != null)
                {
                    int read = socket.Client.EndReceive(ar);
                    if (read > 0)
                    {
                        //处理数据.....通知设备
                        string text =
System.Text.Encoding.ASCII.GetString(_Buffer, 0, read);
                        OnServiceConnector(new
FromService(this.ServiceName,this.ServiceKey),new
ServiceToDevice("1",text,null,null) );

                        OnReceive();
                    }
                    else
                    {
                        this.CloseSocket();
                    }
                }
            }
            catch (SocketException ex)
            {
                this.CloseSocket();
                this.OnServiceLog(ex.Message);
            }
            catch (Exception ex)
            {
                this.OnServiceLog(ex.Message);
```

```
            }
        }
```

4. 接收驱动执行命令信息的结果

```
public override void ServiceConnectorCallback(object obj)
        {
            OnServiceLog(obj.ToString());
            OnServiceLog("设备已经处理完成指令");
        }
```

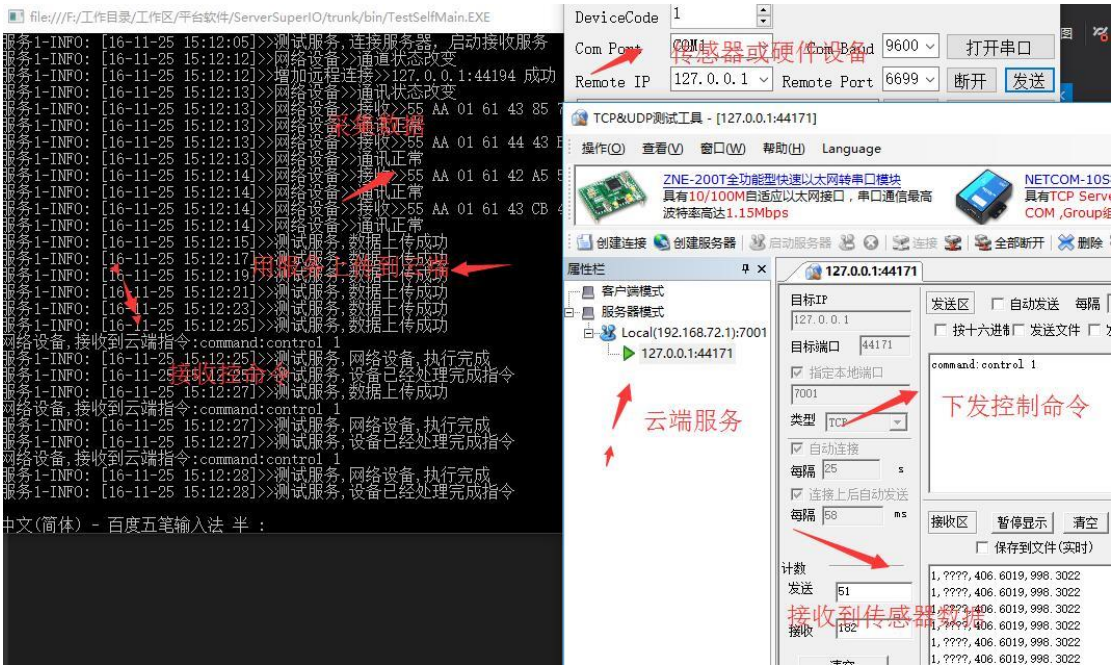
12.7 宿主程序服务实例配置注意事项

```
static void Main(string[] args)
        {
            DeviceSelfDriver dev2 = new DeviceSelfDriver();
            dev2.DeviceParameter.DeviceName = "网络设备";
            dev2.DeviceParameter.DeviceAddr = 1;
            dev2.DeviceParameter.DeviceID = "1";
            dev2.DeviceDynamic.DeviceID = "1";
            dev2.DeviceParameter.DeviceCode = "1";
            dev2.DeviceParameter.NET.RemoteIP = "127.0.0.1";
            dev2.DeviceParameter.NET.RemotePort = 9600;
            dev2.CommunicateType = CommunicateType.NET;
            dev2.Initialize("1");

            IServer server = new ServerManager().CreateServer(new ServerConfig()
            {
                ServerName = "服务1",
                ComReadTimeout = 1000,
                ComWriteTimeout = 1000,
                NetReceiveTimeout = 1000,
                NetSendTimeout = 1000,
                ControlMode = ControlMode.Self,
                SocketMode = SocketMode.Tcp,
                StartReceiveDataFliter = true,
                ClearSocketSession = false,
                StartCheckPackageLength = true,
                CheckSameSocketSession = false,
                DeliveryMode = DeliveryMode.DeviceCode,
            });

            server.AddDeviceCompleted += server_AddDeviceCompleted;
```

```
        server.DeleteDeviceCompleted+=server_DeleteDeviceCompleted;
        server.Start();

        server.AddDevice(dev2);

        TestService.Service service=new TestService.Service();
        service.IsAutoStart = true;
        server.AddService(service);

        while ("exit" == Console.ReadLine())
        {
            server.Stop();
        }
    }
```

12.8 运行效果

1. 图片



2. 视频

http://imgcache.qq.com/tencentvideo_v1/playerv3/TPout.swf?max_age=86400&v=20161117&vid=b0349qe9nlh&auto=0