

## 目 录

11. 实现设备（驱动）与设备（驱动）交互和级联控制.....	2
11.1 概述 .....	2
11.2 设备链接器.....	2
11.3 场景假设 .....	3
11.4 设备驱动开发及注意事项.....	3
11.5 宿主程序服务实例配置注意事项 .....	5
11.6 运行效果 .....	6

官方网站: <http://www.bmpj.net>

## 11. 实现设备（驱动）与设备（驱动）交互和级联控 制

### 11.1 概述

物联网建设中数据采集是基础, 进行控制是目的, 这是两个根本要素。在采集设备数据的时候, 如果该设备的实时数据出现异常, 那么是否存在其他设备要进行联动? 也就是说一个设备出现异常之后, 要对其他某个设备进行联动控制, 以达到避免出现更大危险的情况。

那么不仅要对某个设备进行联动控制, 还要对控制的结果进行反馈给出现异常的设备。形成异常、联动、控制、反馈的闭环, 以达到监测与控制共同作用的目的。

### 11.2 设备链接器

IRunDevice 设备驱动接口继承了 IDeviceConnector 设备链接器接口, 以便使设备驱动与设备驱动之间可以双向交互, 实现数据传递与命令控制的目的。接口的定义如下:

```
public interface IDeviceConnector
{
    /// <summary>
    /// 支行设备连接器
    /// </summary>
    /// <param name="fromDevice">信息传递的发送端</param>
    /// <param name="toDevice">信息传递的目的端, 以及包含的数据信息</param>
    /// <returns></returns>
    object RunDeviceConnector(IFromDevice fromDevice, IDeviceToDevice toDevice);

    /// <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 DeviceConnectorCallback(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 DeviceConnectorCallbackError(Exception ex);

/// <summary>
/// 连接器事件, 发起端
/// </summary>
event DeviceConnectorHandler DeviceConnector;

/// <summary>
/// 确发事件接口
/// </summary>
/// <param name="fromDevice"></param>
/// <param name="toDevice"></param>
void OnDeviceConnector(IFromDevice fromDevice, IDeviceToDevice toDevice);
}
```

通过这个接口定义, 可以得知通过 `OnDeviceConnector` 事件触发信息传递的起始端, 会把信息传递给接收信息端设备驱动的 `RunDeviceConnector` 接口函数, 执行完 `RunDeviceConnector` 函数后, 会把结果数据返回给发送端设备驱动的 `DeviceConnectorCallback` 接口函数, 如果接收数据端设备驱动执行 `RunDeviceConnector` 过程中出现异常, 则会调用发起端设备驱动的 `DeviceConnectorCallbackError` 接口函数, 至此整个流程结束。

## 11.3 场景假设

0 号设备驱动的名称叫金三, 1 号设备驱动的名称叫普京。金三得知韩国朴大妈事件后, 问普京: 大哥, 朴大妈为什么还不下课? 。普京回答: 你不觉得这才是真正的韩剧吗? 傻小子。金三说: 奥黑也真够坑爹的! 完成整个流程的信息传递。

## 11.4 设备驱动开发及注意事项

1. 金三接到信息后, 发起信息传递。触发 `OnDeviceConnector` 事件, 并把 `IDeviceToDevice` 接口 `deviceCode` 参数设置为 1, 指向传递给普京。代码如下:

```
public override void Communicate(ServerSuperIO.Communicate.IRequestInfo
```

```
info)
{
    byte[] cmds = this.Protocol.GetCommand(info.Data);
    CommandArray cr = (CommandArray)cmds[0];

    dynamic obj = this.Protocol.DriverAnalysis<byte[]>(cr.ToString(),
info.Data, info.BigData);
    if (obj != null)
    {
        if (cr == CommandArray.RealTimeData)
        {
            _deviceDyn.Dyn = (Dyn)obj;
            OnDeviceRuningLog("通讯正常");

            Console.WriteLine(">>>>模拟控制命令开始");
            this.OnDeviceConnector(new
FromDevice(this.DeviceParameter.DeviceID,this.DeviceParameter.DeviceCode,t
his.DeviceParameter.DeviceAddr,this.DeviceParameter.DeviceName),new
DeviceToDevice("1",this.DeviceParameter.DeviceName+"问：大哥，朴大妈为
什么还不下课？",null,null) );
        }
    }
}
```

2. 普京接收到信息后，进行信息解析后，并返回他的意思（结果）。代码如下：

```
public override object RunDeviceConnector(IFromDevice fromDevice, IDeviceToDevice
toDevice)
{
    Console.WriteLine(toDevice.Text); //输出其他设备传来的数据。
    return this.DeviceParameter.DeviceName + "答：你不觉得这才是真正的韩剧吗？傻
小子";
}
```

3. 金三接收到普京的返回信息后，把矛头指向了奥黑。代码如下：

```
public override void DeviceConnectorCallback(object obj)
{

```

```
        Console.WriteLine(obj.ToString()); //输出返回结果

        Console.WriteLine(this.DeviceParameter.DeviceName+ "说：奥
黑也真够坑爹的！");

        Console.WriteLine(">>>>模拟控制命令结束");
    }
}
```

注：这里并没有用到 DeviceConnectorCallbackError 执行异常返回接口函数。

## 11.5 宿主程序服务实例配置注意事项

这是用自控模式实现的，设备驱动并没有主动发送数据。代码如下：

```
static void Main(string[] args)
{
    DeviceSelfDriver dev1 = new DeviceSelfDriver();
    dev1.DeviceParameter.DeviceName = "金三";
    dev1.DeviceParameter.DeviceAddr = 0;
    dev1.DeviceParameter.DeviceID = "0";
    dev1.DeviceDynamic.DeviceID = "0";
    dev1.DeviceParameter.DeviceCode = "0";
    dev1.DeviceParameter.COM.Port = 1;
    dev1.DeviceParameter.COM.Baud = 9600;
    dev1.DeviceParameter.NET.RemoteIP = "127.0.0.1";
    dev1.DeviceParameter.NET.RemotePort = 9600;
    dev1.CommunicateType = CommunicateType.NET;
    dev1.Initialize("0");

    DeviceSelfDriver dev2 = new DeviceSelfDriver();
    dev2.DeviceParameter.DeviceName = "普京";
    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(dev1);
server.AddDevice(dev2);

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

## 11.6 运行效果

### 1. 图片



### 2. 视频

[https://imgcache.qq.com/tencentvideo\\_v1/player/v3/TPout.swf?max\\_age=86400&v=20161117&vid=j0348pmc84a&auto=0](https://imgcache.qq.com/tencentvideo_v1/player/v3/TPout.swf?max_age=86400&v=20161117&vid=j0348pmc84a&auto=0)