

目 录

20. 网络通讯控制器分组, 提高交互的负载平衡能力.....	2
20.1 概述	2
20.2 网络控制器示意图	2
20.3 串口控制器示意图	3
20.4 设备驱动网络控制器分组示例代码	3

官方网站: <http://www.bmpj.net>

20. 网络通讯控制器分组, 提高交互的负载平衡能力

20.1 概述

ServerSuperIO 原来在网络通讯模式下, 只有一个网络控制器, 在自控模式、并发模式和单例模式下时都是异步处理返回的数据, 并不会出现性能问题。但是在轮询模式下, 一个网络控制器要按顺序逐一操作设备驱动的发送和接收, 这样就会出现轮询周期过长, 达不到读取数据频率的情况。

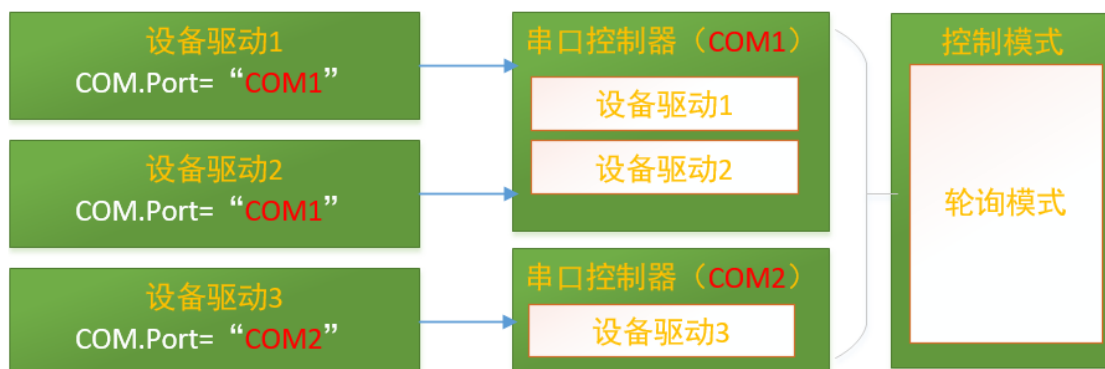
为了解决上述问题, 现在对设备驱动的参数增加了网络控制器分组功能, 网络控制器会根据设备参数设置的分组名称控制设备驱动。例如轮询模式下, 有 1000 个设备驱动, 可以每 10 个设备驱动设置相同的分组名称, 这 10 个设备驱动受控于同一个网络控制器, 如果每间隔 1s 读取一次数据, 那么每个设备驱动的轮询周期是 10s 钟, 其他网络控制器类似。

20.2 网络控制器示意图



根据设置设备驱动网络参数的 ControllerGroup, 可以把设备驱动分配到不同的网络控制中运行, 并且适用于轮询、自控、并发和单例控制模式。

20.3 串口控制器示意图



顺便提一下串口控制器, 通过设置设备驱动的串口号决定被分配到不同的串口控制器, 该控制器只能适用于轮询控制模式。

20.4 设备驱动网络控制器分组示例代码

```
static void Main(string[] args)
{
    string deviceID = "2";
    DeviceDriver dev3 = new DeviceDriver();
    dev3.DeviceParameter.DeviceName = "设备2";
    dev3.DeviceParameter.DeviceAddr = 0;
    dev3.DeviceParameter.DeviceID = deviceID;
    dev3.DeviceParameter.DeviceCode = deviceID;
    dev3.DeviceDynamic.DeviceID = deviceID;
    dev3.DeviceParameter.NET.RemoteIP = "127.0.0.1";
    dev3.DeviceParameter.NET.RemotePort = 9600;
    dev3.DeviceParameter.NET.ControllerGroup = "G2";
    dev3.CommunicateType = CommunicateType.NET;
    dev3.DeviceParameter.NET.WorkMode = WorkMode.TcpServer;
    dev3.Initialize(deviceID);

    deviceID = "3";
    DeviceDriver dev4 = new DeviceDriver();
    dev4.DeviceParameter.DeviceName = "设备3";
    dev4.DeviceParameter.DeviceAddr = 0;
    dev4.DeviceParameter.DeviceID = deviceID;
    dev4.DeviceParameter.DeviceCode = deviceID;
    dev4.DeviceDynamic.DeviceID = deviceID;
    dev4.DeviceParameter.NET.RemoteIP = "127.0.0.1";
```

```
dev4.DeviceParameter.NET.RemotePort = 9600;
dev4.DeviceParameter.NET.ControllerGroup = "G3";
dev4.CommunicateType = CommunicateType.NET;
dev4.Initialize(deviceID);

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

server.AddDeviceCompleted += server_AddDeviceCompleted;
server.DeleteDeviceCompleted += server_DeleteDeviceCompleted;
server.SocketConnected+=server_SocketConnected;
server.SocketClosed+=server_SocketClosed;
server.Start();

server.AddDevice(dev3);
server.AddDevice(dev4);

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