

## 目 录

5. 轮询通讯模式开发及注意事项 .....	2
5.1    概述 .....	2
5.2    通讯机制说明.....	2
5.3    设备驱动开发注意事项 .....	3
5.3.1 实时发送数据 .....	3
5.3.2 优先发送其他数据 .....	3
5.4    宿主程序服务实例配置注意事项 .....	4
5.5    轮询运行效果.....	4

官方网站: <http://www.bmpj.net>

## 5. 轮询通讯模式开发及注意事项

### 5.1 概述

轮询通讯模式是 SuperIO 最早且唯一的通讯模式,在此基础上增加了自控模式、并发模式。ServerSuperIO 框架除了具备这 3 种通讯模式外,还具备单例模式,后边会对这些通讯模式进行详细介绍。

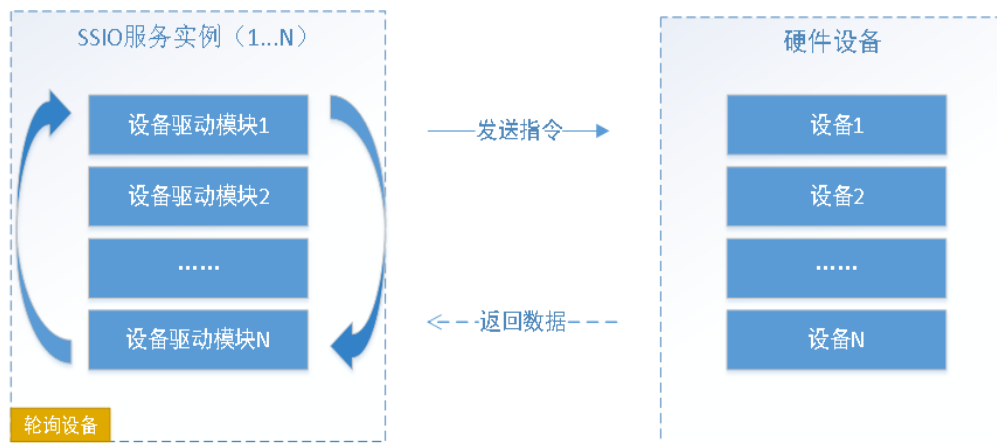
刚参加工作的时候,也不知道什么通讯模式,但是当时工业领域的通讯环境非常复杂,调制解调的通讯链路,300 波特率的传输速率,受地质、电缆等强电磁干扰,经常会出现数据被干扰的现象。当时也没有所谓的互联网技术,为了保障通讯的稳定性,只能进行轮询的通讯模式。

### 5.2 通讯机制说明

这是框架最早的运行模式,串口和网络通讯时都可以使用这种控制模式。当有多个设备 连接到通讯平台时,通讯平台会轮询调度设备进行通讯任务。某一时刻只能有一个设备发送请求命令、等待接收返回数据,这个设备完成发送、接收(如果遇到超时 情况,则自动返回)后,下一个设备才进行通讯任务,依次轮询设备。

应用场景是这样的,服务端与设备进行通讯遵循呼叫应答的方式,也就是 IO 可用的情况下,服务端先发起通讯命令请求,设备根据命令信息,检验通过后返回数据给服务端。这种通讯模式很好理解,每个设备的通讯都遵循排队原则。但是如果某个设备的命令需要及时发送,怎么办? ServerSuperIO 框架是支持设备优先级别调度的,例如:对某个设备要进行实时的检测,需要连续发送命令,那么就需要对设备进行高级别设置,发送请求数据命令。

通讯结构如下图:



## 5.3 设备驱动开发注意事项

### 5.3.1 实时发送数据

ServerSuperIO 框架会轮询调度所有设备, 以呼叫应答的方式向设备发送请求实时数据命令, 对于同一个设备的请求实时数据命令一般相对固定。在调度某一具体设备驱动的时候, 会调用固定的调用 `IRunDevice` 驱动接口的 `GetConstantCommand` 函数, 以获得请求实时数据的命令。代码如下:

```
public override byte[] GetConstantCommand()
{
    byte[] data = this.Protocol.DriverPackage<String>("0", "61", null);
    string hexs = BinaryUtil.ByteToHex(data);
    OnDeviceRuningLog("发送>>" + hexs);
    return data;
}
```

`this.Protocol.DriverPackage` 驱动调用 61 命令获得要发送的命令, 并返回 `byte[]` 数组, ServerSuperIO 获得数据后会自动通过 IO 接口下发命令数据。如果返回 `null` 类型, 系统不进行下发操作。

### 5.3.2 优先发送其他数据

对于一个设备不可能只有一个读实时数据的命令, 可能还存在其他命令进行交互, 例如: 读参数、实时校准等, 这时就需要进行优先级调度发送数据信息。可以通过两种方式让 ServerSuperIO 框架优先调度该设备驱动。

1. 把命令增加发送数据缓存中, 框架从缓存中获得数据后会自动删除, 代码如下:

```
this.Protocol.SendCache.Add("读参数", readParaBytes);
```

2. 设置设备的优先级别属性, 代码如下:

```
this.DevicePriority=DevicePriority.Priority;
```

## 5.4 宿主程序服务实例配置注意事项

在宿主程序中创建服务实例的时候, 需要把服务实例的配置参数设置为轮询通讯模式, 并启动服务实例, 把实例化的设备驱动增加到该服务实例中。代码如下:

```
DeviceDriver dev1 = new DeviceDriver();
dev1.DeviceParameter.DeviceName = "串口设备1";
dev1.DeviceParameter.DeviceAddr = 0;
dev1.DeviceParameter.DeviceID = "0";
dev1.DeviceDynamic.DeviceID = "0";
dev1.DeviceParameter.COM.Port = 1;
dev1.DeviceParameter.COM.Baud = 9600;
dev1.CommunicateType = CommunicateType.COM;
dev1.Initialize("0");

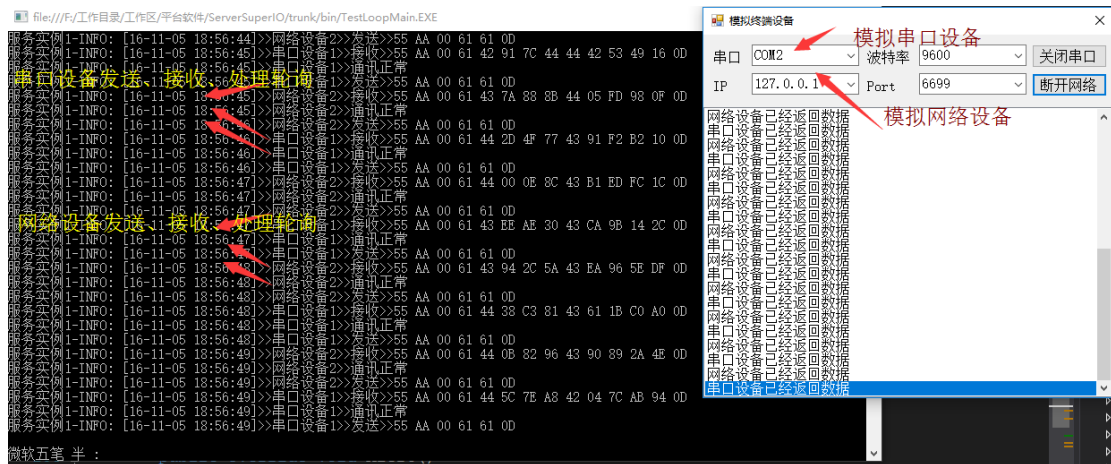
IServer server = new ServerFactory().CreateServer(new ServerConfig()
{
    ServerName = "服务实例1",
    SocketMode = SocketMode.Tcp,
    ControlMode = ControlMode.Loop,
    CheckSameSocketSession = false,
    StartCheckPackageLength = false,
});

server.Start();
server.AddDevice(dev1);
```

ControlMode = ControlMode.Loop 代码是设置服务实例调度设备为轮询控制模式。

## 5.5 轮询运行效果

1. 图片



## 2. 视频

<http://v.qq.com/x/page/o03436yrmu3.html>