# 编写的目的

为了达到对农村站点集中管控的目的，现需对监控一体机实现远程设备控制,设备运行状态数据采集，工况配置的功能。

# 数据库设计

**Xml配置中检测点的字段**

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 类型 | 备注 |  |
| StationUnique |  | 主键 | 原有字段 |
| StationId |  | 站点编号 | 原有字段 |
| RegisterNo |  | 寄存器编号 | 原有字段 |
| TestId |  | 检测点Id | 原有字段 |
| Multiple |  | 倍数 最终获取的数据得乘以这个倍数 | 原有字段 |
| FunctionCode | 字符 | 功能码(1代表线圈,3代表寄存器) | 新增字段 |
| ReceiveTimeout | 数字 | 接收超时(单位:毫秒) | 新增字段 |
| DataType | 字符 | 通道数据类型  (1:16位无符号二进制 2:32位无符号二进制) | 新增字段 |
|  |  |  |  |

**Sqlserver数据库gong\_kuang\_config表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 长度 | 备注 |  |
| Id（不自增） | Int |  | 唯一约束，通过程序控制累加 | 原有字段 |
| Name | nvarhcar | 50 | 配置项名称 | 原有字段 |
| Testid | Int |  | 检测点id | 原有字段 |
| Value\_type | Varchar | 20 | 值类型(与syscode表关联) | 原有字段 |
| address | int |  | 地址 | 原有字段 |
| write\_value | Varchar | 50 | 设置的值 | 原有字段 |
| Read\_value | Varchar | 50 | 读取的值 | 原有字段 |
| Read\_Register | Varchar | 50 | 读寄存器 | 原有字段 |
| Write\_Register | Varchar | 50 | 写寄存器 | 原有字段 |
| Station\_id | Int |  | 站点Id | 原有字段 |
| Remark | nvarchar | 300 | 备注 | 原有字段 |
| Config\_type | Varhcar | 20 | 跟syscode表关联。例如01代表时间配置，02代表策略配置，03代表设备运行状态，04设备故障状态 | 原有字段 |
| FunctionCode | varchar | 20 | 功能码(1代表线圈,3代表寄存器) | 新增字段 |
| ReceiveTimeout | int |  | 接收超时(单位:毫秒) | 新增字段 |
| DataType | varchar | 20 | 通道数据类型  (1:16位无符号二进制 2:32位无符号二进制) | 新增字段 |

**Sqlserver数据库control\_command表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 类型 | 默认值 | 备注 |  |
| Count | Int | 0 | 命令执行的次数  如果执行次数达到指定的次数后,则不再执行此命令(前提:从没执行成功过) | 新增字段 |

在服务端sysconfig.xml中添加一个属性:ExecuteFailureCount,值为3 。 （程序中某条指令执行失败达到ExecuteFailureCount指定的次数后则不再执行）

**执行指令**

**指令读取条件**：control\_command表中state=0 or state is null,并根据gong\_kuang\_id, read\_or\_write分组,取各组日期最大的一行指令。

读取指令的sql:

|  |
| --- |
| SELECT \* FROM dbo.control\_command WHERE add\_datetime IN(  SELECT MAX(c.add\_datetime) FROM dbo.control\_command c ,dbo.gong\_kuang\_config g WHERE c.gong\_kuang\_id=g.id AND g.station\_id=145 AND (c.state=0 OR c.state IS NULL) GROUP BY c.gong\_kuang\_id,read\_or\_write) |

**指令执行成功**:设置

control\_command.state=1 //执行完毕

Control\_command.execute\_result=1; //表示执行成功

control\_command.execute\_comment=” 执行成功”;

control\_command.Complete\_datetime=DateTime.Now;

Gong\_kuang\_config.read\_value=value //这里的value是获取的到值

**指令执行失败**:设置

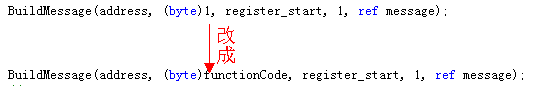
Control\_command.execute\_result=0; //表示执行失败

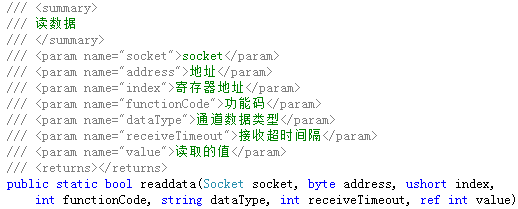
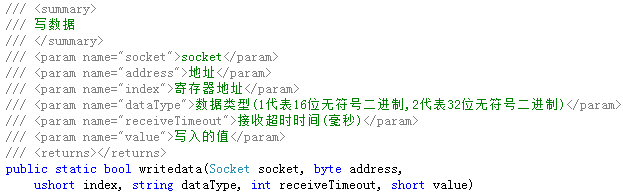
control\_command.execute\_comment=” 执行失败”;

control\_command.Count+=1;

当指令执行失败时，将重复执行指令,直到执行成功或者count达到ExecuteFailureCount指定的次数为止。然后再将control\_command.state=1

**C#服务程序中的修改部分**

1. modbus.sendFc()方法多添加一个参数（参数名称：functionCode 值有1和3两个值)。  
   然后再在BuildMessage改成如下图所示。
2. 给modbus.readdata()和modbus.writedata()方法都添加一个receiveTimeout参数，最终传递到DTU.rdata()方法中做为socket接收的超时时间。
3. 当数据类型为“16位无符号二进制”时  
   当数据类型为“32位无符号二进制”时  
   以上两句代码均在Modbus.sendFc方法中。

Modbus类的读和写方法  


**WEB页面修改部分：**

1. 工况配置页面增加这三个字段（functionCode，receiveTimeout，dataType）的配置。  
   如下图:在原因字段的基础上增加以下三个字段.

保存

取消

下拉框

数字框

下拉框

功能码:

接收超时时间:

通道数据类型:

1. 在syscode表中添加type=98代表功能码(code=1代表线圈，code=3代表寄存器)
2. 在syscode表中添加type=99代表通道数据类型 （code=1代表16位无符号二进制，code=2代表32位无符号二进制）

以下是添加功能码和通道数据类型的脚本

|  |
| --- |
| INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '00','98', N'功能码',N'modbus指令功能码',0.0,0.0 )  INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '98','1', N'线圈',N'',0.0,0.0 )  INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '98','3', N'寄存器',N'',0.0,0.0 )  INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '00','99', N'通道数据类型',N'寄存器的数据类型或者可以说是线圈的数据类型',0.0,0.0 )  INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '99','1', N'16位无符号二进制',N'16位无符号二进制',0.0,0.0 )  INSERT INTO dbo.syscode( type, code, name, note, value1, value2 ) VALUES ( '99','2', N'32位无符号二进制',N'32位无符号二进制',0.0,0.0 ) |