本说明文档适用于modbus\_tcpcex 和 modbuscomm\_ex 规约

# modbus\_tcpc\_ex 与 modbus\_comm\_ex 协议配置

1 环境配置：

增加环境变量 FEPROOT 路径和 SEROOT ⼀样； path 不用添加；

TB2011\_PROTOCOL 规约参数表中添加 modbus\_tcpc\_ex 或 modbus\_comm\_ex这个规约；

检查/cfg 目录中是否有 modbus\_tcpex\_rtu#.csv 或者对应的 modbus\_commex\_rtu#.csv

\*注：'#' 符号填写使用该协议的RTU参数表中序号；

如图所示：



有几个RTU使用配置几个 .csv 文件 如下图显示RTU1使用该规约：



检查 TB2001\_RTUPARA RTU参数表和 TB2002\_CHANPARA 通道参数表配置是否正确，其中注意: RTU表中 遥信个数 和 遥测个数。

2 /cfg 目录下 modbus\_tcpex\_rtu#.csv 文件配置：

表格示例如下：



表说明如下:

|  |  |
| --- | --- |
| **表头** | **描述** |
| 地址 | 对应往下采集modbus\_slave的寄存器或线圈地址 |
| 采集序号 | 对应大平台的采集序序号 可以自行预留和定义 |
| 开始 | 指一个点从寄存器第几位开始（第一位为0） |
| 结束 | 结束是指一个点到寄存器哪一位结束 |
| 类型 | DI，AI，AO，DO，PI，VA |
| 值类型 | 指定点的数据类型 bit int16 uint16 等 |
| 组号 | 一帧或几帧读取的寄存器，包括其中多余不需要的寄存器, 组与组之间的寄存器或线圈则不会被读取 |
| 功能号 | 功能号只用写在相同组号的第一行 功能号分为 1、2、3、4、5、6、15、16 |
| 时间间隔 | 时间间隔只用写在相同组号的第一行 时间间隔是帧和帧之间的间隔 min:100ms |

# 详细说明 ：

1. **地址：**

地址列里面数据为寄存器或线圈地址，相同组号地址需要从小到大按顺序输入，不同组号间的地址可以不按照从小到大的顺序，如图所示：



注：填写顺序必须根据地址按顺序填写，相同地址如果分成AI和DI点，则先配置DI，后配置AI。

1. **开始和结束：**

配置文件的一行是一个AI或DI点

开始是指⼀个点从寄存器第几位开始（第一位为0），如果是线圈，则输入0，代表本线圈。

结束是指⼀个点到寄存器哪⼀位结束，如果是⼀个线圈，则输入0，代表本线圈。

如果两个寄存器合并为⼀个点，结束位写为31号，如图所示：



注：本规约默认第一个寄存器或线圈地址为0，寄存器第一位为0。

**3.类型：**

本规约支持的类型有以下分类：DI，AI，PI，AO，DO，VA。

注：VA类型

VA类型点是指由寄存器转为多个DI点，其值代表某一个DI点为1。具体转化为多少个DI点由结束列决定。DI点的采集序号为本行采集序号加上第几个DI点。

例如：





地址为50的寄存器0号位，1号位，2号位，3号位，4号位，形成5个DI点，采集序号分别为：6，7，8，9，10。

**4.值类型：**

线圈组成的DI点统⼀填写 bit 即可，寄存器则区分以下几种情况：

bit: 寄存器内部字节前高后低;

rv: 前缀代表寄存器内部字节前低后**⾼**;

sw: 前缀代表前寄存器为**⾼**位，后寄存器为低位;

例如：

rv\_bit表示⼀个寄存器前字节为低位，后字节为高位，组成的⼀个DI点; rv\_int16 寄存器内部字节前低后⾼;

sw\_int32 前寄存器为⾼，后为低，寄存器内部字节前⾼后低;

sw\_rv\_int32 前寄存器为⾼，后为低，寄存器内部字节前低后⾼;

须从int16、uint16、rv\_int16、rv\_uint16、float、int32、uint32、rv\_int32、rv\_uint32、rv\_float、sw\_int32、sw\_uint32、sw\_float、sw\_rv\_int32、sw\_rv\_uint32、sw\_rv\_float、bit、rv\_bit这几种类型中选择一种合适的类型填入配置文件中相应的值类型处。

注：

在PI点中只有int16，int32，uint16，uint32，float这5种类型。

DI点统一使用bit类型或者rv\_bit类型。

AI点类型不可为bit类型或者rv\_bit类型。

VA类型统一为uint16。

 **5. 组号和时间间隔:**

1. 组号自行配置，连续点读取可以只设置同⼀组
2. 组与组之间的寄存器或线圈则不会被读取 如果空⼀包后读取必须设置新的组号（本规约中一包为250个字节即125个寄存器，所以同一组中设置地址增加时可以有空寄存器如1、2、4、50、51；但相邻地址的差不能大于125个即一包的长度如1、2、3、5、131、132，需在地址131时再设一个新组） 如连续采集可以设置⼀个组号
3. 同组号的采集地址按照升序设置
4. 采集点⼀个新的组第⼀行必须包含 功能号 和 时间间隔 控制点可以不用填时间间隔

**6.功能号**

功能号和时间间隔只用写在相同组号的第一行，本规约支持的功能号分为1、2、3、4、5、6、15、16。

示例：

**（1）01、02功能码线圈地址读取相应字节数按位读取DI点状态（可取其中连续几位）**

配置图如下图：



说明：读取线圈地址为1-26 的DI点状态 通过对应采集序号为11-28

**（2）03、04功能码读取⼀个或两个寄存器 按位转成 DI，AI，PI，VA点（支持从寄存器的某位开始与结束，两个寄存器数据组合）**

**读取一个寄存器，按位转成DI点**

配置如下图：

说明: 将地址900寄存器转成16个DI点，⼤平台DI采集序号从1-16



说明：将地址1的寄存器中前三位转成DI使用采集序号1，第四位转成另一个DI使用采集序号2

 DI长度限制为8位

**读取一个寄存器，转成AI点**

配置如下图：

说明：将地址1102-1107的6寄存器转为6个AI点，⼤平台AI采集序号为203-208

![EDIHMRQ`Y7KD3]X72Y}_6US]()

![3$X9XOK9FIHY](6ZX3`]1VD]()



 说明：将地址1的寄存器中4-15位转成一个AI点使用采集序号1；

地址19的寄存器中0-7位转成一个AI点使用采集序号9。

 注：如果同一个寄存器里同时放入DI和AI，那他们的值类型应该同时有rv前缀或者都没有，不能DI为rv-bit

AI为无前缀类型。

**读取两个寄存器，转成一个AI点**

配置如下图：

说明：将地址1122和1123的2寄存器 <前寄存器为⾼位，后寄存器为低位> 转为1个AI点，大平台AI采集序号为223

支持不同位置的组合

配置如下：

![EDIHMRQ`Y7KD3]X72Y}_6US]()

 

注：由地址为305,306两个寄存器组成的AI点开始位为3，结束位是29，则取值规则是第一个寄存器的值计算出来后取3-15位，第二个寄存器值计算出来后取0-13（即16-29位），根据值类型进行组合计算出最后AI点的值 使用采集序号156。

**与上述类似 读取寄存器转化为PI，VA点**

配置如下： ![EDIHMRQ`Y7KD3]X72Y}_6US]()









注：PI点只能使用满寄存器即单个寄存器（0-15）或两个寄存器（0-31）

**（3）05功能码控制一个线圈DO点**

配置如下：

 说明：将DO控制点号1-6的点控制对应地址为1-6的线圈 可以不用填开始和结束列

**（4）06功能码控制一个寄存器AO点**

配置如下：



说明：AO控制点号1,2,3 控制地址是5001,5011,5021 可以不用填开始和结束列

**06功能码控制DO点，一个寄存器内的每一位(或其中几位)都是DO点**

注：此为特殊功能

首先确保要控制的点在采集部分已被读取为AI点 值类型必须为 uint16 类型 AI

每次下控只控制一个DO点，即寄存器的某一位

配置如下图：



说明：寄存器地址为8004和8005先被采集为AI点，DO控制点33为8004寄存器的第0 位，DO控制点34为8004寄存器的第1位，以此类推 。

必须填开始和结束列，与DI相似

**（5）本规约中15功能码控制一个寄存器DO点，16功能码控制一个寄存器AO点**

配置如下





说明：AO控制点1-3为寄存器10-12的值，DO控制点1-3为寄存器23-25的值

可以不设置开始结束列。

⬇Tips

DI DO AI AO 类型采集序号点按照升序排列，同⼀个类型的采集序号是唯一的

组号必须设置，控制点相同、同功能码的设置组号可以设置同⼀组

建议以地址列为基准 升序排列需要的转换的DI和AI点

配置文件表格整体结构的类型顺序为先反馈 (DI AI) 后控制 (DO AO)