

2020-07 QT-ISCS3.0 BATA3 更新说明

- 运行script文件夹内 `oracle_upgrade.bat` 文件

► 设备信息标签增加 **图纸代码** 和 **位置信息**

- `TB1004_DEVICE` 设备表中新增两列 **图纸代码** 和 **位置信息** 需要配置后保存 关联设备信息标签增加显示；限制为 `char(80)`

寿命	图纸代码	位置信息
0	EAF/RAF-A01	B端环控机房

- 设备左键标签 **设备信息** 如下图显示

图纸代码: EAF/RAF-A01

位置信息: B端环控机房

► AO控制配置说明

一个设备最大只能显示 5个 **AO** 控制条目

- 单个设备配置 一个 AO 必须配置对应一个 AI 点代码一致 注:AI点的采集序号可以填-1
- AI配置注意 **量测类型** & **单位** 列对应图中标注位置



- 排序按照AI表中 **点代码** 升序显示
- |TB4008_AO 表中 **转换斜率** 和 **转换截距** 与|TB4007_AI 表中工程置换 **转换斜率** 和 **转换截距** 保持一致 注: 如 AO 斜率配置0 则默认就不会对下发的值做处理
- 设备如果有 **就地状态** 则无法控制 !!!

◎ TIPS:

- 如果要不显示 AO , 修改对应 AI 点的设备名称;
- 如同一量测类型有多个点 导致控制界面显示同一量测类型区分不出来 需要 增加 量测类型进行区分

► modbus_tcp_c_ex 协议配置

1 配置:

- 增加环境变量 **FEPROOT** 路径和 **SERROOT** 一样; path里不用添加
- |TB2001_protocol 规约参数表中添加 **modbus_tcp_c_ex** 这个规约
- 检查/cfg 目录中是否有 **modbus_tcpex_rtu#.csv** # 符号填写使用该协议的RTU参数表中序号
- 有几个RTU使用配置几个 .csv 文件 如下图显示RTU1使用该规约:

modbus_tcpex_rtu1.csv	2020/6/11 17:17	XLS 工作表	1 KB
-----------------------	-----------------	---------	------

- 检查 |TB2001_RTUPARARTU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#文件配置:

• 表格示例如下:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔				
2	900	1	0	0	DI	bit	1	3	100				
3	900	2	1	1	DI	bit	1						
4	900	3	2	2	DI	bit	1						
5	900	4	3	3	DI	bit	1						
6	900	5	4	4	DI	bit	1						
7	900	6	5	5	DI	bit	1						
8	900	7	6	6	DI	bit	1						
9	900	8	7	7	DI	bit	1						
10	900	9	8	8	DI	bit	1						
11	900	10	9	9	DI	bit	1						
12	900	11	10	10	DI	bit	1						
13	900	12	11	11	DI	bit	1						
14	900	13	12	12	DI	bit	1						
15	900	14	13	13	DI	bit	1						
16	900	15	14	14	DI	bit	1						
17	900	16	15	15	DI	bit	1						
18	901	17	0	0	DI	bit	1						
19	901	18	1	1	DI	bit	1						
20	901	19	2	2	DI	bit	1						
21	901	20	3	3	DI	bit	1						
22	901	21	4	4	DI	bit	1						
23	901	22	5	5	DI	bit	1						
24	901	23	6	6	DI	bit	1						
25	901	24	7	7	DI	bit	1						
26	901	25	8	8	DI	bit	1						
27	901	26	9	9	DI	bit	1						
28	901	27	10	10	DI	bit	1						
29	901	28	11	11	DI	bit	1						
30	901	29	12	12	DI	bit	1						
31	901	30	13	13	DI	bit	1						
32	901	31	14	14	DI	bit	1						
33	901	32	15	15	DI	bit	1						
34	902	33	0	0	DI	bit	1						
35	902	34	1	1	DI	bit	1						
36	902	35	2	2	DI	bit	1						
37	902	36	3	3	DI	bit	1						
38	902	37	4	4	DI	bit	1						
39	902	38	5	5	DI	bit	1						
40	902	39	6	6	DI	bit	1						
41	902	40	7	7	DI	bit	1						
42	902	41	8	8	DI	bit	1						
43	902	42	9	9	DI	bit	1						
44	902	43	10	10	DI	bit	1						
45	902	44	11	11	DI	bit	1						

• 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
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：

- 同组号地址需要从小到大按升序输入

2. 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：

	A	B	C	D	E	F	G	H	I
1	地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
598	1122	223	0	31	AI	sw_int32	3		

3. 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: 寄存器内部字节前低后高；
 - 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高位，后字节为低位，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；

4. 组号和时间间隔：

- 组号自行配置，连续点读取可以只设置同一组
- 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** 如连续采集可以设置一个组号
- 同组号的采集地址按照升序设置
- 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

示例：

1. 03读取一个寄存器 按位转成 DI点

- 配置如下图：

地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
900	1	0	0	DI	bit	1	3	100
900	2	1	1	DI	bit	1		
900	3	2	2	DI	bit	1		
900	4	3	3	DI	bit	1		
900	5	4	4	DI	bit	1		
900	6	5	5	DI	bit	1		
900	7	6	6	DI	bit	1		
900	8	7	7	DI	bit	1		
900	9	8	8	DI	bit	1		
900	10	9	9	DI	bit	1		
900	11	10	10	DI	bit	1		
900	12	11	11	DI	bit	1		
900	13	12	12	DI	bit	1		
900	14	13	13	DI	bit	1		
900	15	14	14	DI	bit	1		
900	16	15	15	DI	bit	1		

- 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

2. 03读取一个寄存器 转成 AI点

- 配置如下图：

A	B	C	D	E	F	G	H	I
地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
1101	574	13	13	DI	bit	3		
1101	575	14	14	DI	bit	3		
1101	576	15	15	DI	bit	3		
1102	203	0	15	AI	int16	3		
1103	204	0	15	AI	int16	3		
1104	205	0	15	AI	int16	3		
1105	206	0	15	AI	int16	3		
1106	207	0	15	AI	int16	3		
1107	208	0	15	AI	int16	3		
1108	209	0	15	AI	int16	3		
1109	210	0	15	AI	int16	3		

- 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

3. 03读取两个寄存器 转成 一个AI点

- 配置如下图：

A	B	C	D	E	F	G	H	I
地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
1121	222	0	15	AI	int16	3		
1122	223	0	31	AI	sw_int32	3		

- 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 默认1122+1

4. 05命令控制一个线圈 DO点

- 配置如下图：

地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
1	1			DO	bit	55	5	
2	2			DO	bit	55		
3	3			DO	bit	55		
4	4			DO	bit	55		
5	5			DO	bit	55		
6	6			DO	bit	55		

- 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

5.06命令控制一个寄存器的某一位 DO点

- 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** 类型为 **AI** 如果此点已经被处理，则在处理下面加上一行，按照规定设置相应的数据
- 配置如下图：

A	B	C	D	E	F	G	H	I
地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
8004	31000	0	15	AI	uint16	4	3	100
8005	31002	0	15	AI	uint16	4		
8004	33	0	0	DO	bit	4	6	
8004	34	1	1	DO	bit	4		
8004	35	2	2	DO	bit	4		
8004	36	3	3	DO	bit	4		
8004	37	4	4	DO	bit	4		
8004	38	5	5	DO	bit	4		
8004	39	6	6	DO	bit	4		
8004	40	7	7	DO	bit	4		
8004	41	8	8	DO	bit	4		
8004	42	9	9	DO	bit	4		
8004	43	10	10	DO	bit	4		
8004	44	11	11	DO	bit	4		
8004	45	12	12	DO	bit	4		
8004	46	13	13	DO	bit	4		
8004	47	14	14	DO	bit	4		
8004	48	15	15	DO	bit	4		
8005	49	0	0	DO	bit	4		
8005	50	1	1	DO	bit	4		
8005	51	2	2	DO	bit	4		
8005	52	3	3	DO	bit	4		
8005	53	4	4	DO	bit	4		
8005	54	5	5	DO	bit	4		
8005	55	6	6	DO	bit	4		
8005	56	7	7	DO	bit	4		
8005	57	8	8	DO	bit	4		
8005	58	9	9	DO	bit	4		
8005	59	10	10	DO	bit	4		
8005	60	11	11	DO	bit	4		
8005	61	12	12	DO	bit	4		
8005	62	13	13	DO	bit	4		
8005	63	14	14	DO	bit	4		
8005	64	15	15	DO	bit	4		

- 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

6.06命令控制一个寄存器 AO点

- 配置如下图：

A	B	C	D	E	F	G	H	I
地址	采集序号	开始	结束	类型	值类型	组号	功能号	时间间隔
5001	1			AO	int16	89	6	
5011	2			AO	int16	89		
5021	3			AO	int16	89		

- 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

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 切记!!!
- 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
- 按寄存器转换位的16位都转 不然增删减会增加工作量
- 表格整体结构的类型顺序为先反馈 (DI AI) 后控制 (DO AO)