



北京宏达信诺科技有限公司
Beijing Hodacigna Technology Co.Ltd

HXGE 系列网关采集

PLC_S7_200_SMART 快速

北京宏达信诺科技有限公司

<http://www.hodacigna.com>



目录

目录	1
第 1 章 插件概述.....	2
第 2 章 系统配置.....	3
第 3 章 插件使用详解.....	4
第 4 章 配置数据标签.....	6
第 5 章 本地运行操作（一般做测试用）	10
1. 本地运行.....	10
2. 远程监视	10
3. 查看报文	11
第 6 章 网关运行操作.....	11
1. 上传下载->更新工程至网关.....	11
2. 远程监视	12
3. 查看报文	13

第1章 插件概述

S7-200 SMART CPU 集成 1 个以太网接口和 1 个 RS485 接口，通过扩展 CM01 信号板，其通信端口数量最多可增至 3 个。可满足小型自动化设备连接触摸屏、变频器等第三方设备的众多需求。

以太网通信

所有 CPU 模块标配以太网接口，支持西门子 S7 协议、TCP/IP 协议、有效支持多种终端连接：

- 可作为程序下载端口（使用普通网线即可）
- 与 SMART LINE HMI 进行通信
- 通过交换机与多台以太网设备进行通信，实现数据的快速交互
- 最多支持 4 个设备通信

串口通信

S7-200 SMART CPU 均集成 1 个 RS485 接口，可以与变频器、触摸屏等第三方设备通信。如果需要额外的串口，可通过扩展 CM01 信号板来实现，信号板支持 RS232/RS485 自由转换，最多支持 4 个设备。

串口支持下列协议：

- Modbus-RTU
- PPI
- USS
- 自由口通信

与上位机的通信

通过 PC Access，操作人员可以轻松通过上位机读取 S7-200 SMART 的数据，从而实现设备监控或者进行数据存档管理。

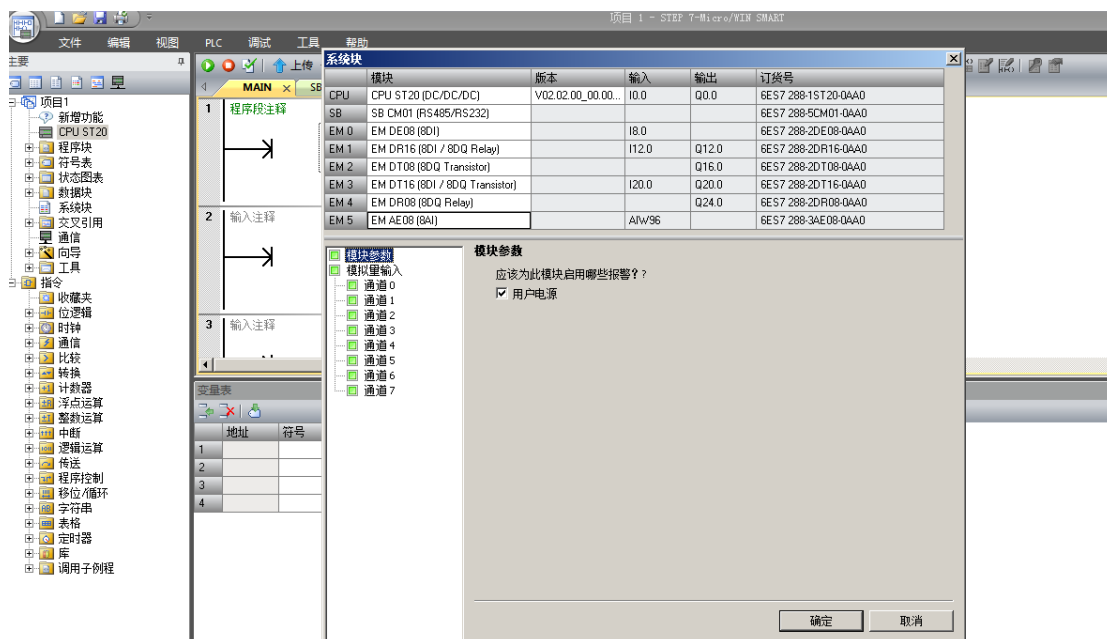
（PC Access 是专门为 S7-200 系列 PLC 开发的 OPC 服务器协议，专门用于小型 PLC 与上位机交互的 OPC 软件）

本插件用于采集服务，通过 PROFINET 接口读写 Siemens S7-200 SMART 系列 PLC 设备的各种寄存器的数据。

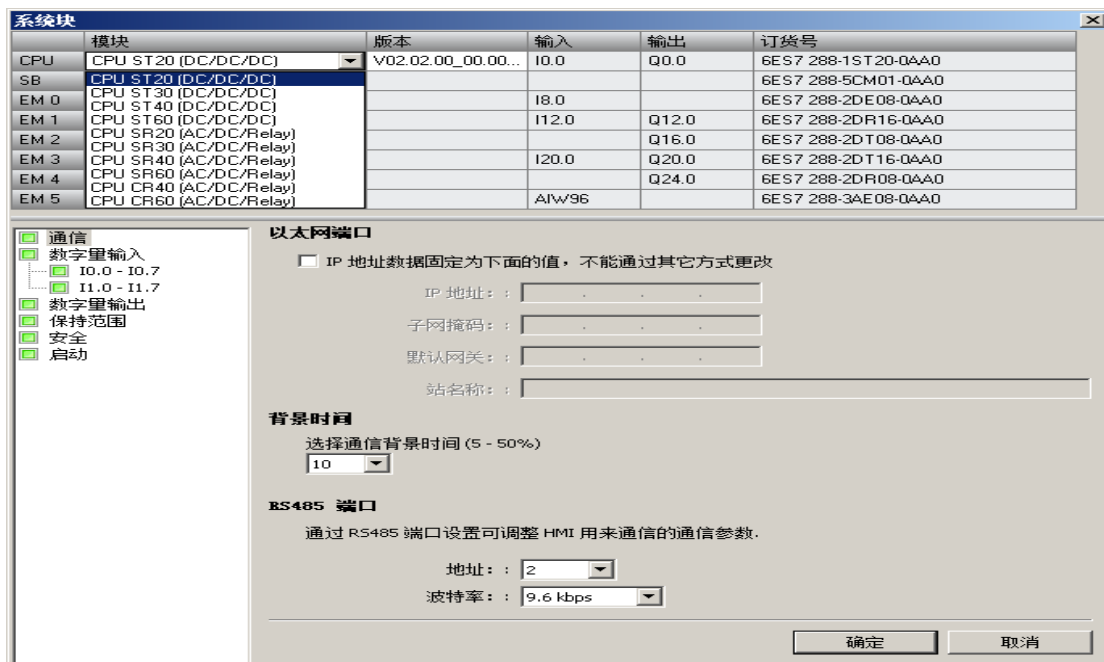
第2章 系统配置

SIMATIC S7-200 SMART CPU 本机上集成了一个以太网端口和一个 RS485 接口，必要时可以通过信号板扩展一个 RS232 端口，其中编程设备只能通过以太网端口连接到 CPU。

在项目中，双击 CPU 类型打开，根据实际硬件配置，CPU 选择 SR20，依次选择自动配置。

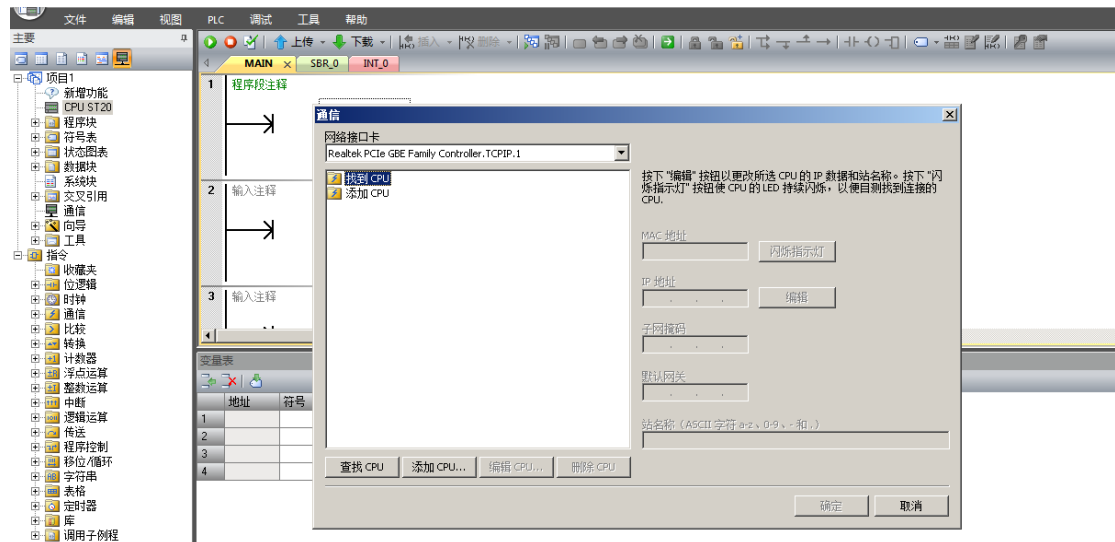


此处可以不配置静态 IP，如果配置 IP，那么也必须在此处修改。



因为 CPU 没有预组态的 IP 地址，可以通过 STEP 7 S7 200 SMART V2.2 的编程软件确定 CPU 的 IP 地址。

打开编程软件，单击导航栏的通信按钮，打开通信对话框，选择网络接口卡的 TCP/IP，双击更新可访问的设备，获取已连接的 CPU 列表，选中所需的 CPU，单击闪烁指示灯按钮，观察指示灯 状态，识别 CPU，如果没有设定固定 IP，则可以编辑设置动态 IP，必须与 CPU 同类型，同子网段。



另外，本插件和 PLC 通讯，需要设置 PLC 的链接机制（只要使用 Siemens S7 以太网协议，都必须设置此项）。选中 PLC 节点，右击，选择“属性” - “常规” - “保护”，勾选链接机制下的“允许从远程伙伴（PLC、HMI、OPC、...）使用 PUT/GET 通讯访问”。

第3章 插件使用详解

1. 创建通道 选择插件，如下图



插件选择-采集

配置插件-问答式

MODBUS ASCII

MODBUS RTU

MODBUS TCP

OPC LINUX

串口服务器程序(串口转TCP SERVER)

SNMP 客户端

仿真驱动(数据模拟器)

电力行业标准

PLC

AB DF1串口协议

AB ETHERNET/IP CIP协议(SLC500系列)

AB LOGIX 5000 TCP

GE-ETHERNET(以太网)

mitsubishi FX系列

mitsubishi FX3u系列(以太网)

mitsubishi Q系列串口

mitsubishi Q系列

mitsubishi Q系列(3E)

PLC_NAIS_NEWTOCOL

OMRON FINS NET

OMRON HOST LINK

S7-1200

S7-1500

S7_200 PPI

S7_200 SMART

S7_200 TCP

S7_300 TCP

S7_400 TCP

大数据

继电保护

穿网闸驱动

行业标准

数据库

名称: PLC_S7_200_SMART

描述: S7_200 SMART

分类: PLC

厂家: SIEMENS

确定

取消

路径

历史

升级

序号	插件名称	平台	版本	最后修改日期
1	PLC_S7_200_SMART.335x	335x		2019-02-22 08:36:15
2	PLC_S7_200_SMART.dll	dll	5.0.0.1	2020-02-19 08:28:46
3	PLC_S7_200_SMART.x86	x86		2019-06-13 06:22:54
4	PLC_S7_200_SMART.x86_64	x86_64		2020-02-19 06:32:05
5	PLC_S7_200_SMART.xt	xt		2019-02-22 08:32:51
6	PLC_S7_200_SMART.zq	zq		2020-02-22 08:35:10

无插件描述信息

2. 通道参数

名称(必须为英文) C1

描述: 通道1

其他参数

厂家: SIEMENS

描述: S7_200 SMART

选择插件

帮助

路径: E:\网关配置软件\网关配置软件HXGEBuild V6\PluginIo\PLC_S7_200_SMART

通道类型: TCP客户端

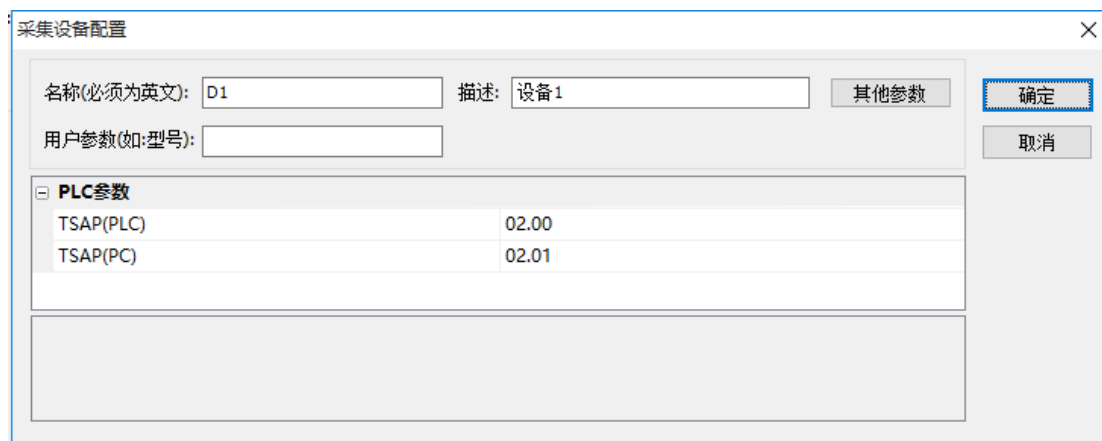
远程IP 192.168.0.88

远程端口: 102

需要绑定本地IP

序号	名称	说明
1	通道类型	Siemens S7以太网协议为TCP连接，默认为TCP客户端
2	远程IP	填写PLC的IP地址
3	远程端口	默认值: 102

3. 创建设备



采集设备配置

名称(必须为英文): 描述:

用户参数(如:型号):

☒ PLC参数

TSAP(PLC)	02.00
TSAP(PC)	02.01

4. 设备参数

PLC参数

PLC: 连接资源的地址, 默认为02.00

PC: 连接资源的地址, 默认为 02.01

第4章 配置数据标签

根据用户提供的点表信息, 配置采集标签。

A. 配置对话框



IO数据标签-新建

名称: 描述:

寄存器类型:

偏移地址:

解析方式:

位偏移:

配置帮助

①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址可以保存一个量值, 地址为0, 4, 8等。

③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

序号	名称	说明
1	寄存器类型	选择PLC内各寄存区，目前支持M、I、Q、T、C、VS、AI区
2	偏移地址	填写对应寄存器在PLC中定义的地址或偏移量
3	解析方式	选择对寄存器内数据的解析方式，支持的格式有：BIT位、8位无符号、8位有符号、16位无符号、16位有符号、32位无符号、32位有符号、32位IEEE格式浮点数
4	位偏移	解析寄存器内某个位的值，范围：0~7

B. 示例1:

假设PLC中有Tag_1~Tag_5共5个点需要获取数据:

配置参数 数据标签									
全部	浮点	整型	布尔	字符串	二进制				
序号	名称(英文)	描述(中文)	数据类型	读写方向	采集周期(秒)	寄存器类型	寄存器地址	解析方式	位偏移量
1	Tag1	标签1	浮点	只读	1000	0	0	0	0
2	Tag2	标签1	浮点	只读	1000	1	1	1	0
3	Tag3	标签1	浮点	只读	1000	2	2	2	0
4	Tag4	标签1	浮点	只读	1000	3	3	3	0
5	Tag5	标签1	浮点	只读	1000	5	4	7	0

获取 Tag_1 的值, Tag_1 在 PLC 中地址为 0, 该点属于 M 区, 数据类型 BIT(数据长度 1 位), 寄存器地址填 “0”, 数据位填 “0”, 配置好的标签对话框如下:

IO数据标签-修改

名称: Tag1

描述: 标签1

其他参数

确定

取消

寄存器类型: M寄存器

偏移地址: 0

解析方式: BIT位

位偏移: 0

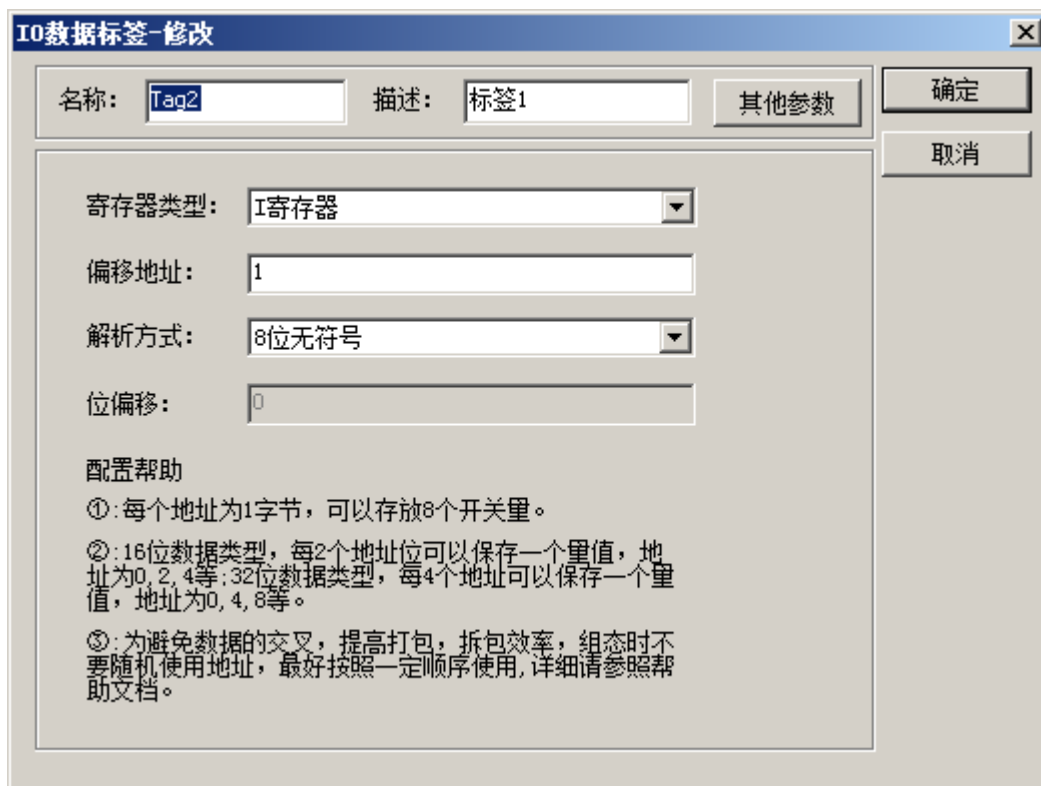
配置帮助

①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址位可以保存一个量值, 地址为0, 4, 8等。

③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

获取 Tag_2 的值, Tag_2 在 PLC 中地址为 1%Q0.1, 该点属于 I 区, 数据类型 BYTE(数据长度 1 字节), 寄存器地址填“1”, 数据位填“0”, 配置好的标签对话框如下:



I/O 数据标签-修改

名称: 描述: 其他参数

寄存器类型:

偏移地址:

解析方式:

位偏移:

配置帮助

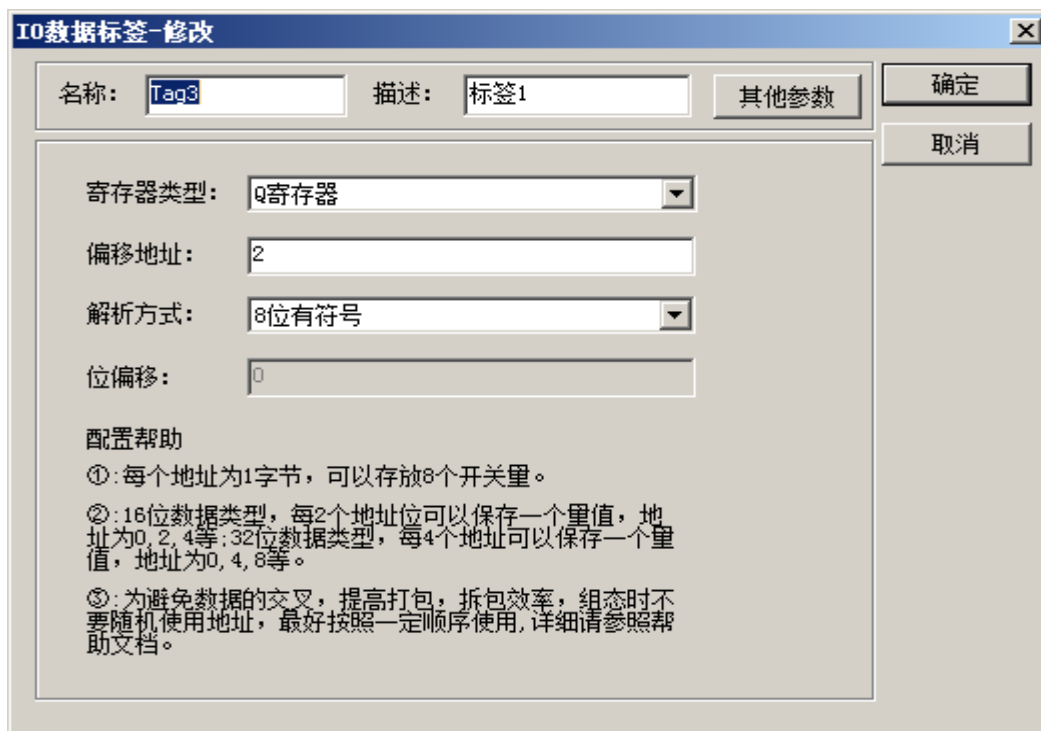
①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址可以保存一个量值, 地址为0, 4, 8等。

③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

确定 取消

获取 Tag_3 的值, Tag_3 在 PLC 中地址为 2, 该点属于 M 区, 数据类型 BYTE(数据长度 1 字节), 寄存器地址填“2”, 数据位填“0”, 配置好的标签对话框如下:



I/O 数据标签-修改

名称: 描述: 其他参数

寄存器类型:

偏移地址:

解析方式:

位偏移:

配置帮助

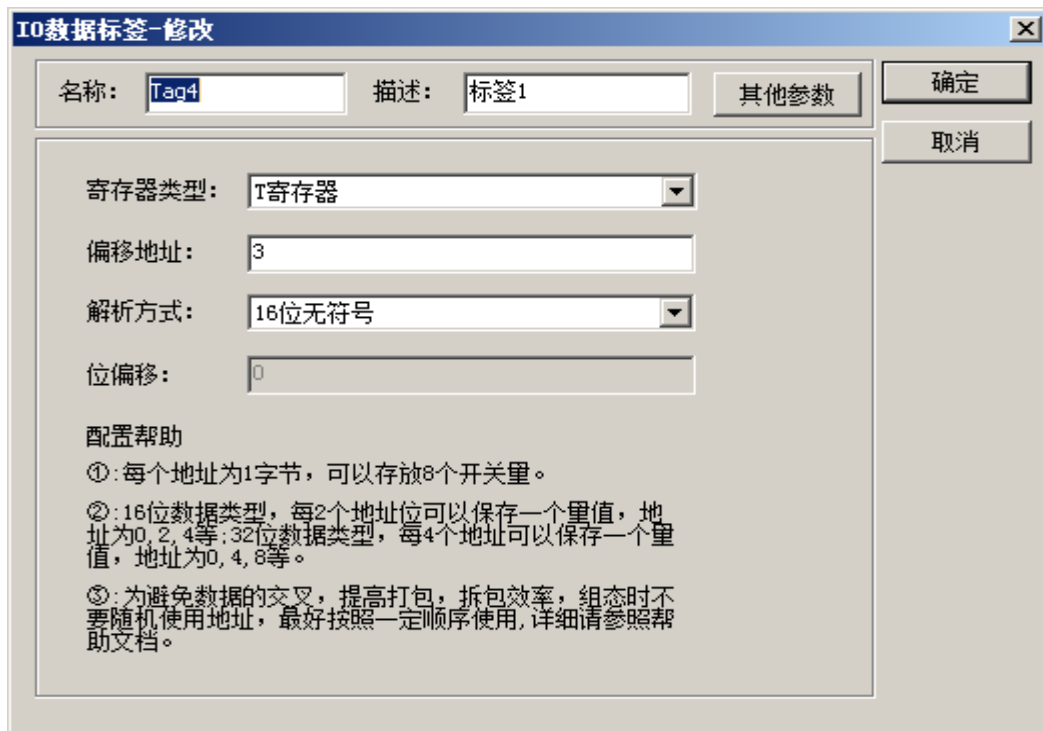
①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址可以保存一个量值, 地址为0, 4, 8等。

③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

确定 取消

获取 Tag_4 的值, Tag_4 在 PLC 中地址为 3, 该点属于 M 区, 数据类型 Word(数据长度 2 字节), 寄存器地址填“3”, 配置好的标签对话框如下:



IO数据标签-修改

名称: Tag4 描述: 标签1 其他参数 确定 取消

寄存器类型: T寄存器

偏移地址: 3

解析方式: 16位无符号

位偏移: 0

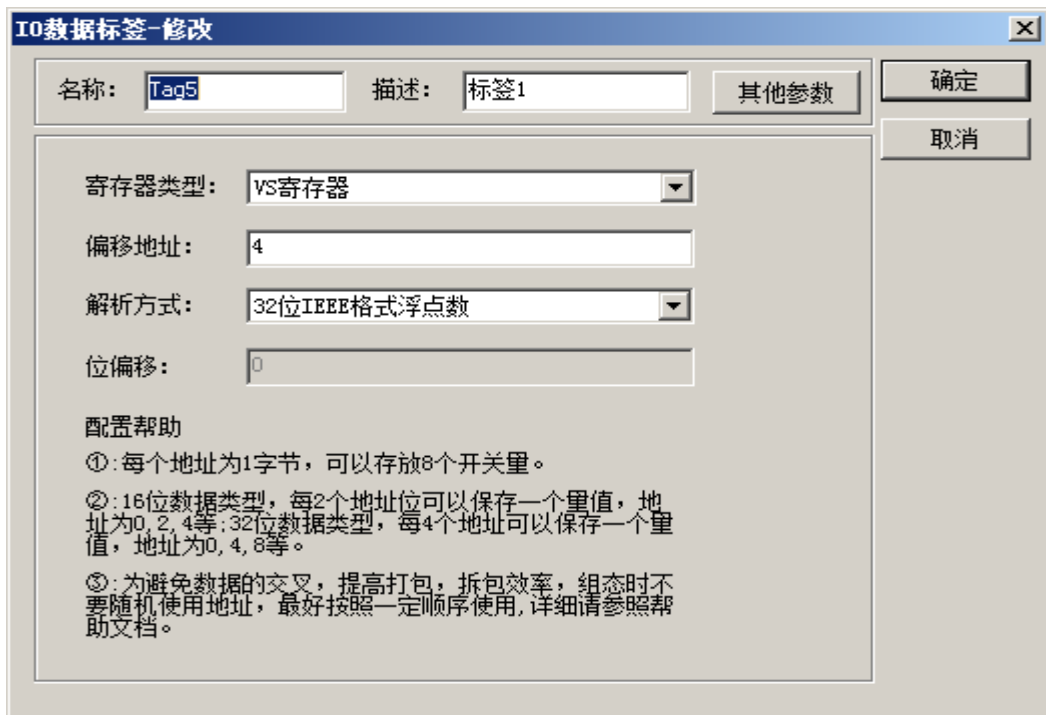
配置帮助

①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址位可以保存一个量值, 地址为0, 4, 8等。

③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

获取 Tag_5 的值, Tag_5 在 PLC 中地址为 4, 该点属于 VS 区, 数据类型 DWord(数据长度 4 字节), 寄存器地址填“4”, 配置好的标签对话框如下:



IO数据标签-修改

名称: Tag5 描述: 标签1 其他参数 确定 取消

寄存器类型: VS寄存器

偏移地址: 4

解析方式: 32位IEEE格式浮点数

位偏移: 0

配置帮助

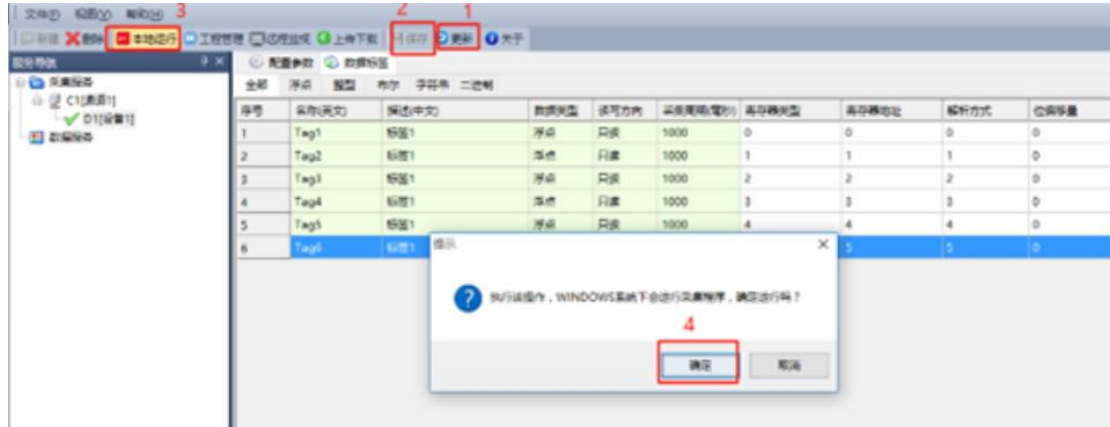
①: 每个地址为1字节, 可以存放8个开关量。

②: 16位数据类型, 每2个地址位可以保存一个量值, 地址为0, 2, 4等; 32位数据类型, 每4个地址位可以保存一个量值, 地址为0, 4, 8等。

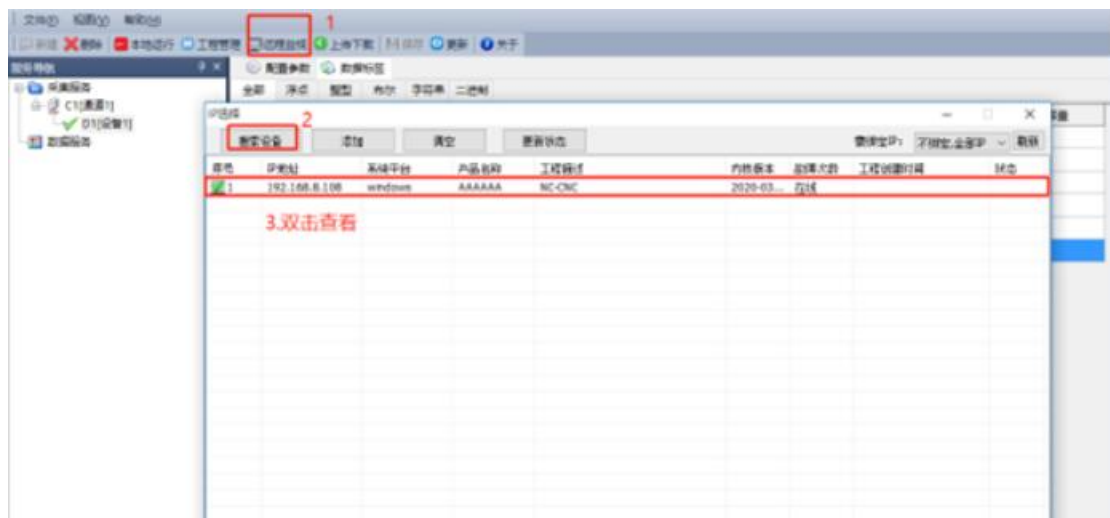
③: 为避免数据的交叉, 提高打包, 拆包效率, 组态时不要随机使用地址, 最好按照一定顺序使用, 详细请参照帮助文档。

第5章 本地运行操作（一般做测试用）

1. 本地运行

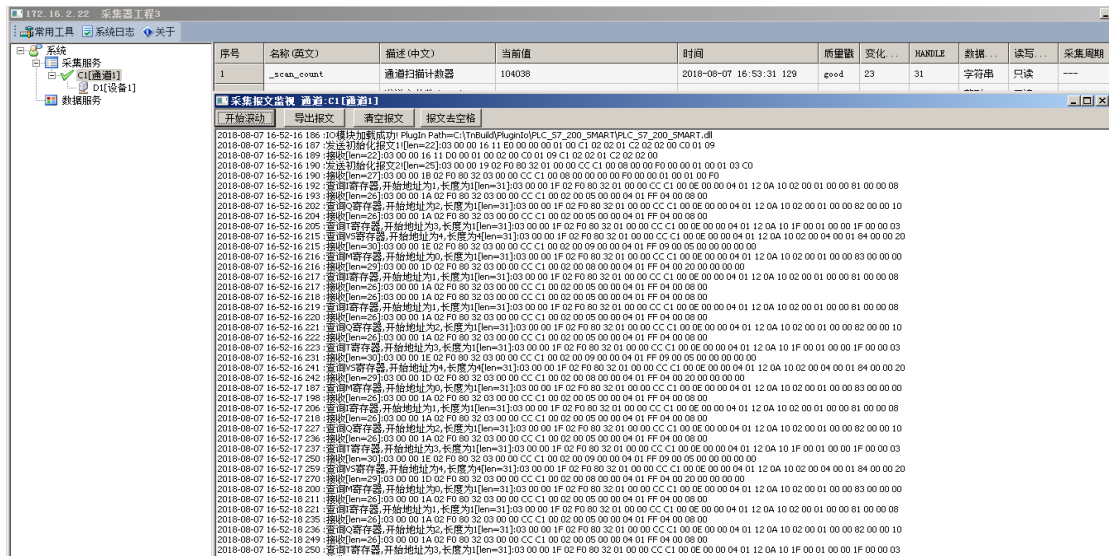


2. 远程监视



序号	名称(英文)	描述(中文)	数据类型	读写方向	采集周期/单位	寄存器地址	寄存器地址	寄存器方式	采集结果
1	Tag1	标签1	浮点	只读	1000	0	0	0	0
2	Tag2	标签1	浮点	只读	1000	1	1	1	0
3	Tag3	标签1	浮点	只读	1000	2	2	2	0
4	Tag4	标签1	浮点	只读	1000	3	3	3	0
5	Tag5	标签1	浮点	只读	1000	4	4	4	0
6	Tag6	标签1	浮点	只读	1000	5	5	5	0

3. 查看报文

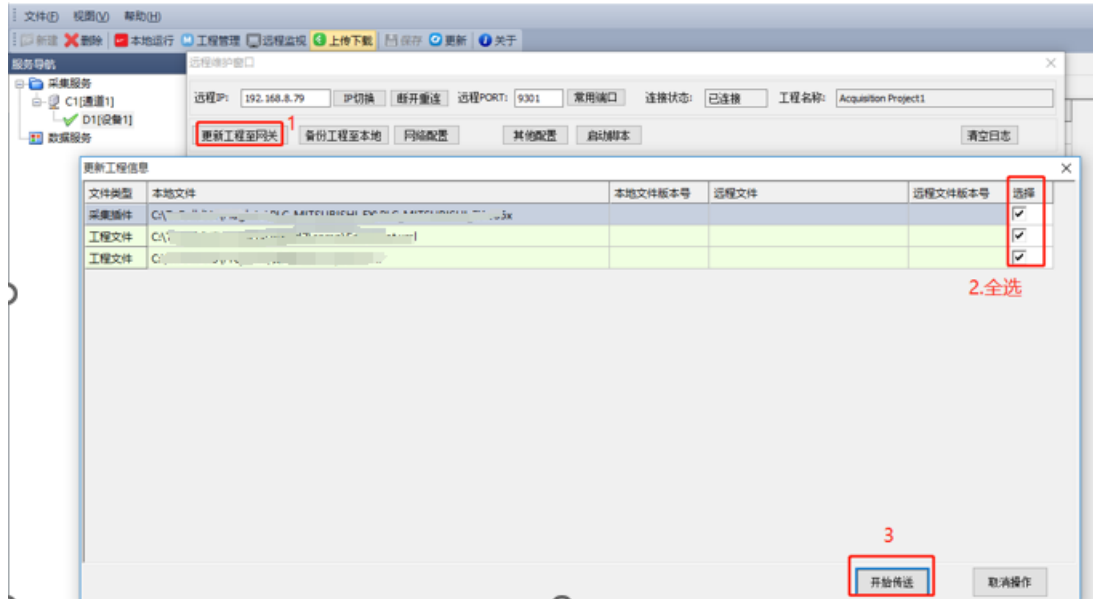


第6章 网关运行操作

1. 上传下载->更新工程至网关

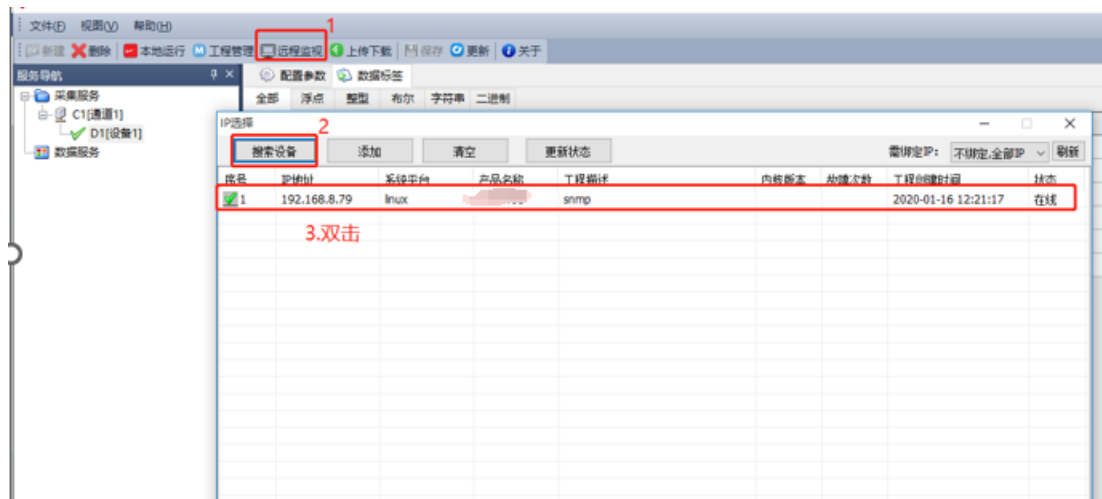


开始传送



所有文件传送完毕，然后退出上传下载界面。

2. 远程监视





3. 查看报文

172.16.2.22 采集器工程3
常用工具 系统日志 关于

系统
采集器
C1[设备1]
D1[设备1]
数据服务

序号	名称(英文)	描述(中文)	当前值	时间	质量戳	变化...	HANDLE	数据...	读写...	采集周期
1	_scan_count	通道扫描计数器	104038	2018-08-07 16:53:31.129	good	23	31	字符串	只读	---

采集报文监视 通道: C1 [通道1]
开始滚动 导出报文 清空报文 报文去空格

2018-08-07 16:52:16 186: I/O模块加载成功! Plugin Path=C:\TrBuild\Plugins\PLC_S7_200_SMART\PLC_S7_200_SMART.dll
2018-08-07 16:52:16 187: 发送初始化报文1[Len=25]:03 00 00 16 11 F0 00 00 01 00 C1 02 02 01 C2 02 02 00 C0 01 09
2018-08-07 16:52:16 189: 接收[Len=23]:03 00 00 16 11 00 00 01 00 02 00 C0 01 09 C1 02 02 01 C2 02 02 00
2018-08-07 16:52:16 190: 发送初始化报文2[Len=25]:03 00 00 19 02 F0 80 32 01 00 00 CC C1 00 08 00 00 F0 00 00 01 00 01 03 00
2018-08-07 16:52:16 190: 接收[Len=27]:03 00 00 18 02 F0 80 32 03 00 00 CC C1 00 08 00 00 00 F0 00 00 01 00 01 00 F0
2018-08-07 16:52:16 193: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 81 00 00 08
2018-08-07 16:52:16 193: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 202: 查询寄存器, 开始地址为2, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 82 00 00 10
2018-08-07 16:52:16 204: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 205: 查询寄存器, 开始地址为3, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 1F 00 01 00 00 1F 00 00 03
2018-08-07 16:52:16 215: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 84 00 00 20
2018-08-07 16:52:16 215: 接收[Len=30]:03 00 00 1E 02 F0 80 32 03 00 00 CC C1 00 02 00 09 00 00 04 01 FF 09 00 05 00 00 00 00 00
2018-08-07 16:52:16 216: 查询寄存器, 开始地址为0, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 83 00 00 00
2018-08-07 16:52:16 216: 接收[Len=29]:03 00 00 1D 02 F0 80 32 03 00 00 CC C1 00 02 00 08 00 00 04 01 FF 04 00 20 00 00 00 00
2018-08-07 16:52:16 217: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 81 00 00 08
2018-08-07 16:52:16 217: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 218: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 219: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 81 00 00 08
2018-08-07 16:52:16 219: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 221: 查询寄存器, 开始地址为2, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 82 00 00 10
2018-08-07 16:52:16 222: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 223: 查询寄存器, 开始地址为3, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 1F 00 01 00 00 1F 00 00 03
2018-08-07 16:52:16 223: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:16 231: 接收[Len=30]:03 00 00 1E 02 F0 80 32 03 00 00 CC C1 00 02 00 09 00 00 04 01 FF 09 00 05 00 00 00 00 00
2018-08-07 16:52:16 241: 查询寄存器, 开始地址为4, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 04 00 01 84 00 00 20
2018-08-07 16:52:16 243: 接收[Len=29]:03 00 00 1D 02 F0 80 32 03 00 00 CC C1 00 02 00 08 00 00 04 01 FF 04 00 20 00 00 00 00
2018-08-07 16:52:17 187: 查询寄存器, 开始地址为0, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 83 00 00 00
2018-08-07 16:52:17 196: 接收[Len=29]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:17 206: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 81 00 00 08
2018-08-07 16:52:17 218: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:17 227: 查询寄存器, 开始地址为2, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 82 00 00 10
2018-08-07 16:52:17 236: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:17 237: 查询寄存器, 开始地址为3, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 1F 00 01 00 00 1F 00 00 03
2018-08-07 16:52:17 250: 接收[Len=30]:03 00 00 1E 02 F0 80 32 03 00 00 CC C1 00 02 00 09 00 00 04 01 FF 09 00 05 00 00 00 00 00
2018-08-07 16:52:17 259: 查询寄存器, 开始地址为4, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 04 00 01 84 00 00 20
2018-08-07 16:52:17 270: 接收[Len=29]:03 00 00 1D 02 F0 80 32 03 00 00 CC C1 00 02 00 08 00 00 04 01 FF 04 00 20 00 00 00 00
2018-08-07 16:52:18 200: 查询寄存器, 开始地址为0, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 83 00 00 00
2018-08-07 16:52:18 211: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:18 221: 查询寄存器, 开始地址为1, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 81 00 00 08
2018-08-07 16:52:18 235: 接收[Len=30]:03 00 00 1E 02 F0 80 32 03 00 00 CC C1 00 02 00 09 00 00 04 01 FF 09 00 05 00 00 00 00 00
2018-08-07 16:52:18 236: 查询寄存器, 开始地址为2, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 02 00 01 00 00 82 00 00 10
2018-08-07 16:52:18 249: 接收[Len=26]:03 00 00 1A 02 F0 80 32 03 00 00 CC C1 00 02 00 05 00 00 04 01 FF 04 00 08 00
2018-08-07 16:52:18 250: 查询寄存器, 开始地址为3, 长度为[Len=31]:03 00 00 1F 02 F0 80 32 01 00 00 CC C1 00 0E 00 00 04 01 12 0A 10 1F 00 01 00 00 1F 00 00 03