



智流形机器人

从 制 造 到 智 造

INSTRUCTION

智流形机器人控制系统

Modbus 通讯使用文档

版权声明

深圳市智流形机器人技术有限公司版权所有。

深圳市智流形机器人技术有限公司保留所有版权以及相关的知识产权。

声明

深圳市智流形机器人技术有限公司保留在没有预先通知的情况下修改产品或其特性的权利。

深圳市智流形机器人技术有限公司并不承担由于使用产品不当而产生的直接或是间接的伤害或损坏的责任。

联系我们

深圳市智流形机器人技术有限公司

Shenzhen iManifold Robot Technology Ltd.

地址：深圳市宝安区华丰国际机器人产业园二期 B 栋 410

智云流形科技（江阴）有限公司

iManifold Technology (Jiangyin) Ltd.

地址：江苏省无锡市江阴市长山大道 55 号天安数码城 60 幢 102 室

目 录

目录

版权声明	2
声明.....	2
联系我们	2
目 录.....	3
1 Modbus 通讯协议介绍.....	5
2 寄存器地址表	6
2.1 Input_Register	7
2.1.1 系统状态.....	7
2.1.2 伺服状态.....	7
2.1.3 工作模式.....	8
2.1.4 当前姿态号	8
2.1.5 当前外部轴号	8
2.1.6 回放速度.....	8
2.1.7 示教速度.....	8
2.1.8 当前关节位置	8
2.1.9 当前笛卡尔位置.....	9
2.1.10 当前工件坐标系 1 位置	9
2.1.11 当前工件坐标系 2 位置	10
2.1.12 当前世界坐标系位置.....	10
2.1.13 当前法兰坐标系位置.....	10
2.1.14 报警信息	11
2.1.15 当前坐标系	11
2.1.16 当前程序	11
2.1.17 全局数值变量监控.....	13
2.1.18 全局字符串变量监控.....	13
2.1.19 模拟量输入	14
2.1.20 模拟量输出	15
2.2 Holding_Register.....	15

2.2.1	伺服使能.....	15
2.2.2	清除报警.....	15
2.2.3	设定回放速度	15
2.2.4	设定示教速度	16
2.2.5	切换工作模式.....	16
2.2.6	执行程序.....	16
2.2.7	运动到点.....	17
2.2.8	设定坐标系	17
2.2.9	点动机器人	18
2.2.10	数字量输出	19
2.2.11	模拟量输出	19
2.2.12	数值变量写入.....	19
2.2.13	字符串变量吸入操作.....	23
2.2.14	全局数值变量监控.....	23
2.2.15	全局字符变量监控.....	23
2.3	Discrete_Input.....	24
2.3.1	数字量输入.....	24
2.3.2	数字量输出.....	24
3	通讯示例	25
3.1	操作示例.....	25
4	常见问题	35
4.1	数据类型的转换	35
4.2	数据格式.....	35
4.3	数值的输入.....	36
4.4	读/写定义	36
4.5	写入多个线圈/多个寄存器.....	37
4.6	Modbus poll 错误码.....	39

1 Modbus 通讯协议介绍

本功能为工业机器人提供 modbus_tcp 通讯方式,提供工业机器人的系统配置参数,变量资源,运动控制指令,机器人工作状态等接口。

通过本功能可对机器人系统进行modbus_tcp 通讯,从而实现机器人实时运行状态的监控和收集,多设备互联互通和协同配合运行,远程操作控制和运维等需求。

部分特性:

1. 远程模式下生效
2. 更新周期:
 1. input 变量每 100ms 更新一次
 2. hold 中变量部分每 100ms 更新 100 个,循环更新 (全部变量相当于 1000ms 更新一次)
 3. coil 变量每 100ms 更新一次
 4. discrete 变量每 100ms 更新一次
 5. 用户触发命令通过信号槽立即更新
3. 详细地址请使用地址计算工具

2 寄存器地址表

Modbus 支持四种寄存器类型

寄存器种类	对象类型	访问类型	内容
Input_Register 状态寄存器	16-Bit	只读	I/O 系统提供这种类型数据
Coli 线圈 数字量输入	1-Bit	读写	通过应用程序改变这种类型数据
Holding_Registe 命令寄存器	16-Bit	读写	通过应用程序改变这种类型数据
Discrete_Input 数字量输入	1-Bit	只读	I/O 系统提供这种类型数据

2.1 Input_Register

状态寄存器：只读

2.1.1 系统状态

名称	数据类型	地址	备注
系统状态	16-bit integer	0x0000	0: 待机; 1: 使能; 2: 执行中; 3: 暂停; 2560: 警告; 2561: 伺服警告 2562: 执行中警告 2563: 暂停中警告 2564: 报警;

2.1.2 伺服状态

名称	数据类型	地址	备注
伺服状态	16-bit integer	0x0001	0: 伺服关闭; 1: 伺服打开; 2: 上伺服报警; 3: 驱动器错误;

2.1.3 工作模式

名称	数据类型	地址	备注
工作模式	16-bit integer	0x0002	0: 远程示教模式; 1: 远程回放模式;

2.1.4 当前姿态号

名称	数据类型	地址	备注
姿态号	16-bit integer	0x0003	当前姿态号

2.1.5 当前外部轴号

名称	数据类型	地址	备注
外部轴号	16-bit integer	0x0004	

2.1.6 回放速度

名称	数据类型	地址	备注
回放速度	16-bit integer	0x0005	百分比

2.1.7 示教速度

名称	数据类型	地址	备注
示教速度	16-bit integer	0x0006	百分比

2.1.8 当前关节位置

名称	数据类型	地址	备注
轴1	64-bit double	0x1000~0x1003	
轴2	64-bit double	0x1004~0x1007	
轴3	64-bit double	0x1008~0x100B	
轴4	64-bit double	0x100C~0x100F	
轴5	64-bit double	0x1010~0x1013	
轴6	64-bit double	0x1014~0x1017	
轴7	64-bit double	0x1018~0x101B	
轴8	64-bit double	0x101C~0x101F	
轴9	64-bit double	0x1020~0x1023	
轴10	64-bit double	0x1024~0x1027	
轴11	64-bit double	0x1028~0x102B	
轴12	64-bit double	0x102C~0x102F	
轴13	64-bit double	0x1030~0x1033	
轴14	64-bit double	0x1034~0x1037	
轴15	64-bit double	0x1038~0x103B	
轴16	64-bit double	0x103C~0x103F	

2.1.9 当前笛卡尔位置

名称	数据类型	地址	备注
x	64-bit double	0x1050~0x1053	
y	64-bit double	0x1054~0x1057	
z	64-bit double	0x1058~0x105B	
r	64-bit double	0x105C~0x105F	
p	64-bit double	0x1060~0x1063	
y	64-bit double	0x1064~0x1067	

2.1.10 当前工件坐标系 1 位置

名称	数据类型	地址	备注
x	64-bit double	0x1070~0x1073	
y	64-bit double	0x1074~0x1077	
z	64-bit double	0x1078~0x107B	
r	64-bit double	0x107C~0x107F	
p	64-bit double	0x1080~0x1083	
y	64-bit double	0x1084~0x1087	

2.1.11 当前工件坐标系 2 位置

名称	数据类型	地址	备注
x	64-bit double	0x1090~0x1093	
y	64-bit double	0x1094~0x1097	
z	64-bit double	0x1098~0x109B	
r	64-bit double	0x109C~0x109F	
p	64-bit double	0x10A0~0x10A3	
y	64-bit double	0x10A4~0x10A7	

2.1.12 当前世界坐标系位置

名称	数据类型	地址	备注
x	64-bit double	0x10B0~0x10B3	
y	64-bit double	0x10B4~0x10B7	
z	64-bit double	0x10B8~0x10BB	
r	64-bit double	0x10BC~0x10BF	
p	64-bit double	0x10C0~0x10C3	
y	64-bit double	0x10C4~0x10C7	

2.1.13 当前法兰坐标系位置

名称	数据类型	地址	备注
x	64-bit double	0x10D0~0x10D3	
y	64-bit double	0x10D4~0x10D7	
z	64-bit double	0x10D8~0x10DB	
r	64-bit double	0x10DC~0x10DF	
p	64-bit double	0x10E0~0x10E3	
y	64-bit double	0x10E4~0x10E7	

2.1.14 报警信息

名称	数据类型	地址	备注
报警信息	128-byte string	0x1100~0x113F	以字节流保存报警信息

2.1.15 当前坐标系

名称	数据类型	地址	备注
tcs_id	16-bit integer	0x1140	
wcs_id	16-bit integer	0x1141	
pcs1_id	16-bit integer	0x1142	
pcs2_id	16-bit integer	0x1143	
当前坐标系	16-bit integer	0x1144	0: 关节坐标系; 1: 笛卡尔坐标系; 2: 工件坐标系1; 3: 工件坐标系2; 4: 世界坐标系; 5: 工具坐标系; 6: 扩展关节坐标系; 7: 未定义

2.1.16 当前程序

--	--	--	--

名称	数据类型	地址	备注
当前工程名	128-byte string	0x1150~0x118F	以字节流保存报警信息
当前程序名	128-byte string	0x1190~0x11CF	以字节流保存报警信息
当前执行程序行数	32-bit integer	0x11D0~0x11D1	
当前执行程序模式	16-bit integer	0x11D2	0: 一次; 1: 步进; 2: 循环;

2.1.17 全局数值变量监控

名称	数据类型	地址	备注
变量01	64-bit double	0x11E0~0x11E3	
变量02	64-bit double	0x11E4~0x11E7	
变量03	64-bit double	0x11E8~0x11EB	
变量04	64-bit double	0x11EC~0x11EF	
变量05	64-bit double	0x11F0~0x11F3	
变量06	64-bit double	0x11F4~0x11F7	
变量07	64-bit double	0x11F8~0x11FB	
变量08	64-bit double	0x11FC~0x11FF	
变量09	64-bit double	0x1200~0x1203	
变量10	64-bit double	0x1204~0x1207	

2.1.18 全局字符串变量监控

名称	数据类型	地址	备注
变量01	128byte string	0x1210~0x124F	
变量02	128byte string	0x1250~0x128F	
变量03	128byte string	0x1290~0x12CF	
变量04	128byte string	0x12D0~0x130F	
变量05	128byte string	0x1310~0x134F	
变量06	128byte string	0x1350~0x138F	
变量07	128byte string	0x1390~0x13CF	
变量08	128byte string	0x13D0~0x140F	
变量09	128byte string	0x1410~0x144F	
变量10	128byte string	0x1450~0x148F	

2.1.19 模拟量输入

名称	数据类型	地址	备注
AI0	64-bit double	0x2000~0x2003	
AI1	64-bit double	0x2004~0x2007	
.....			
AI49	64-bit double	0x20C4~0x20C7	

2.1.20 模拟量输出

名称	数据类型	地址	备注
AO0	64-bit double	0x2100~0x2103	
AO1	64-bit double	0x2104~0x2107	
.....			
AO49	64-bit double	0x21C4~0x21C7	

2.2 Holding_Register

命令寄存器（加粗表示命令字，其余为参数）

2.2.1 伺服使能

名称	数据类型	地址	备注
伺服使能	16-bit integer	0x0000	上升沿有效，生效后自动复位； 如当前未使能，则该信号触发后伺服使能，如当前已经使能，则该信号触发后伺服去使能

2.2.2 清除报警

名称	数据类型	地址	备注
清除报警	16-bit integer	0x0001	上升沿有效，生效后自动复位；该信号触发后尝试清除报警信息

2.2.3 设定回放速度

名称	数据类型	地址	备注
回放速度	16-bit integer	0x0002	百分比，值改变立即生效

2.2.4 设定示教速度

名称	数据类型	地址	备注
示教速度	16-bit integer	0x0003	百分比，值改变立即生效

2.2.5 切换工作模式

名称	数据类型	地址	备注
工作模式	16-bit integer	0x0004	0: 远程点动; 1: 远程回放 值改变立即生效

2.2.6 执行程序

名称	数据类型	地址	备注
执行工程名	128-byte string	0x1000~0x103F	以字节流保存报警信息
执行程序名	128-byte string	0x1040~0x107F	以字节流保存报警信息
执行程序行数	32-bit integer	0x1080~0x1081	
执行程序模式	16-bit integer	0x1082	0: 一次; 1: 步进; 2: 循环;
执行程序	16-bit integer	0x1083	沿有效，生效后自动复位；该信号触发后执行程序
暂停程序	16-bit integer	0x1084	上升沿有效，生效后自动复位；该信号触发后暂停程序
继续执行	16-bit integer	0x1085	上升沿有效，生效后自动复位；该信号触发后停止程序
停止程序	16-bit integer	0x1086	上升沿有效，生效后自动复位；该信号触发后停止程序
恢复执行	16-bit integer	0x1087	上升沿有效，生效后自动复位；该信号触发后停止程序

2.2.7 运动到点

名称	数据类型	地址	备注
坐标系类型	16-bit integer	0x10A0	0: 关节坐标系; 1: 笛卡尔坐标系; 2: 工件坐标系1; 3: 工件坐标系2; 4: 世界坐标系; 5: 工具坐标系; 6: 扩展关节坐标系; 7: 未定义
pcs_id	16-bit integer	0x10A1	
wcs_id	16-bit integer	0x10A2	
tcs_id	16-bit integer	0x10A3	
关节位置(16-double)	64-bit double*16	0x10A4~0x10E3	存储结构参考2.1.1
笛卡尔位置(6-double)	64-bit double*6	0x10E4~0x10FB	存储结构参考2.1.2
姿态号	16-bit integer	0x10FC	
外部轴号	16-bit integer	0x10FD	
运动坐标系	16-bit integer	0x10FE	0: 关节; 1: 笛卡尔
执行运动到点	16-bit integer	0x10FF	0: 停止运动; 1: 开始运动 500ms后自动停止运动

2.2.8 设定坐标系

名称	数据类型	地址	备注
tcs_id	16-bit integer	0x1110	值改变立即生效
wcs_id	16-bit integer	0x1111	值改变立即生效

pcs1_id	16-bit integer	0x1112	值改变立即生效
pcs2_id	16-bit integer	0x1113	值改变立即生效
当前坐标系	16-bit integer	0x1114	0: 关节坐标系; 1: 笛卡尔坐标系; 2: 工件坐标系1; 3: 工件坐标系2; 4: 世界坐标系; 5: 工具坐标系; 6: 扩展关节坐标系; 7: 未定义 值改变立即生效

2.2.9 点动机器人

名称	数据类型	地址	备注
运动模式	16-bit integer	0x1120	0: 连续运动; 1: 寸动
轴1点动	16-bit integer	0x1121	0: 停止运动; 1: 正向运动; 2: 反向运动 500ms后自动停止运动
轴2点动	16-bit integer	0x1122	0: 停止运动; 1: 正向运动; 2: 反向运动 500ms后自动停止运动
.....			
轴16点动	16-bit integer	0x1130	0: 停止运动; 1: 正向运动; 2: 反向运动 500ms后自动停止运动

2.2.10 数字量输出

名称	数据类型	地址	备注
组号	16-bit integer	0x1140	共计50组, 1..50
端口号	16-bit integer	0x1141	每组16个, 1..16
目标值	16-bit integer	0x1142	值改变立即生效

2.2.11 模拟量输出

名称	数据类型	地址	备注
编号	16-bit integer	0x1150	共计50组, 1..50, 改变立即生效
目标值	64-bit double	0x1151-0x1154	值

2.2.12 数值变量写入

名称	数据类型	地址	备注
编号	16-bit integer	0x1160	共计1000组, 1..1000, 改变立即生效
目标值	64-bit double	0x1161-0x1164	值

2.2.13 字符串变量吸入操作

名称	数据类型	地址	备注
编号	16-bit integer	0x1170	共计1000组, 1..1000, 改变立即生效
目标值	128 byte array	0x1171-0x11B0	值改

2.2.14 全局数值变量监控

名称	数据类型	地址	备注
编号01	16-bit integer	0x11C0	
编号02	16-bit integer	0x11C1	
编号03	16-bit integer	0x11C2	
编号04	16-bit integer	0x11C3	
编号05	16-bit integer	0x11C4	
编号06	16-bit integer	0x11C5	
编号07	16-bit integer	0x11C6	
编号08	16-bit integer	0x11C7	
编号09	16-bit integer	0x11C8	
编号10	16-bit integer	0x11C9	

2.2.15 全局字符变量监控

名称	数据类型	地址	备注
编号01	16-bit integer	0x11D0	
编号02	16-bit integer	0x11D1	
编号03	16-bit integer	0x11D2	

编号04	16-bit integer	0x11D3	
编号05	16-bit integer	0x11D4	
编号06	16-bit integer	0x11D5	
编号07	16-bit integer	0x11D6	
编号08	16-bit integer	0x11D7	
编号09	16-bit integer	0x11D8	
编号10	16-bit integer	0x11D9	

2.3 Discrete_Input

2.3.1 数字量输入

名称	数据类型	地址	备注
DI0	1-bit	0x0000	
DI1	1-bit	0x0001	
.....			
DI799	1-bit	0x031F	

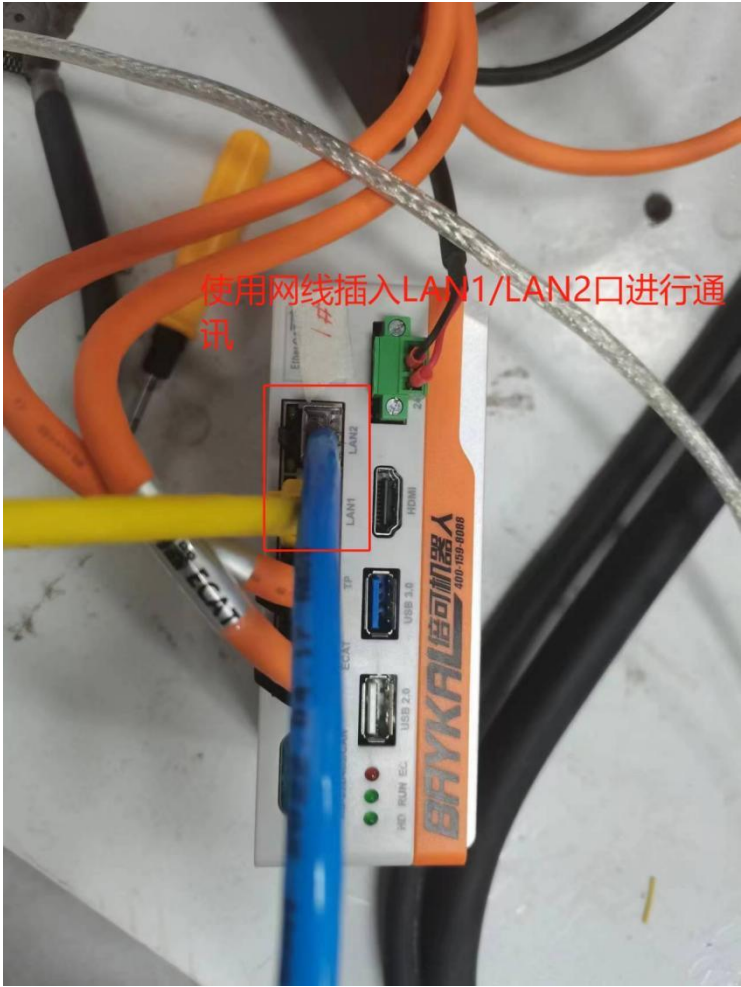
2.3.2 数字量输出

名称	数据类型	地址	备注
DO0	1-bit	0x1000	
DO1	1-bit	0x1001	
.....			
DO799	1-bit	0x131F	

3 通讯示例

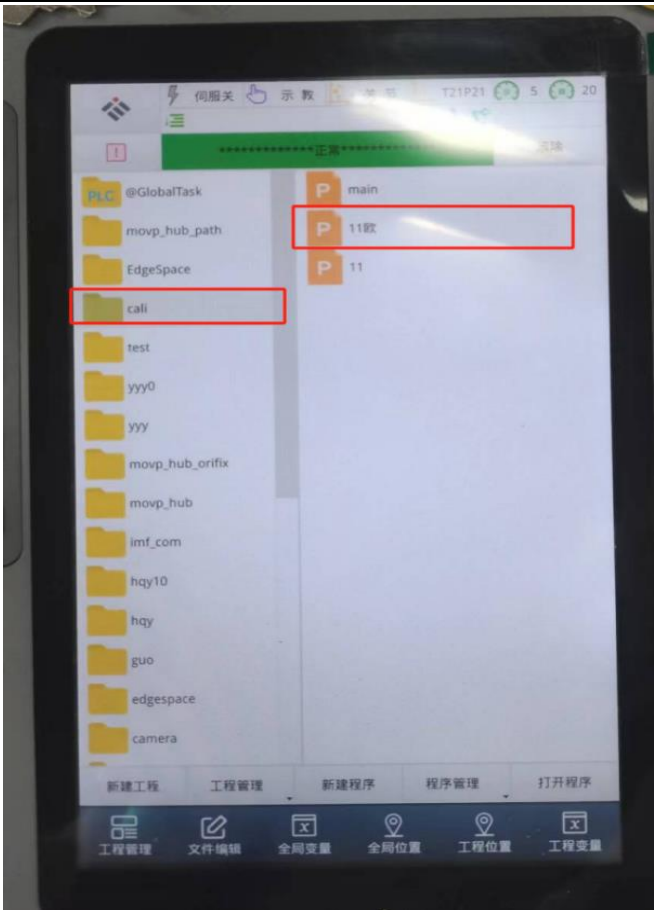
Modbus 有众多客户端，本文档中采用 Modbus poll 工具演示如何通过 Moubus 与机器人系统进行通讯。Modbus 通讯仅在机器人处于远程模式下进行。

3.1 操作示例

项目	操作步骤
连接控制器	 网口 Lan1 或 Lan2 与电脑通过网线连接

设置网络	需要电脑 IP 与控制器 IP 保持一致
连接设置	打开 Modbus Poll 后打开连接界面，连接协议为 Modbus TCP/IP； IP 地址要与机器人 IP 地址相同； 服务器端口为 9501；协议版本为 Ipv4；其余默认。

项目示例



演示：用 Modbus poll 启动一个工程名为 cali，程序名为 11 欧，执行程序模式为多次循环,从程序第 11 行开始执行

设置读/写定义

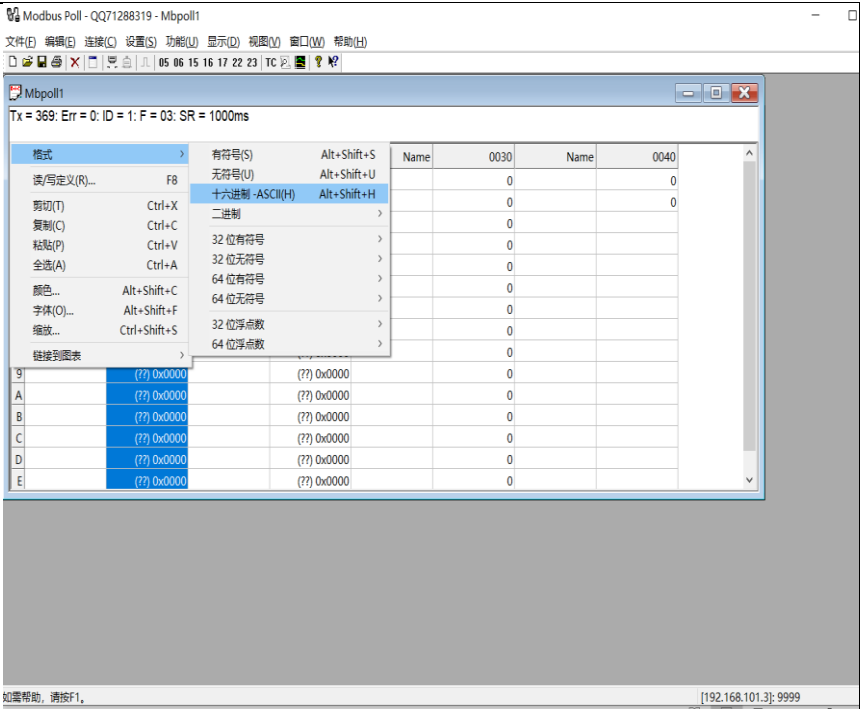
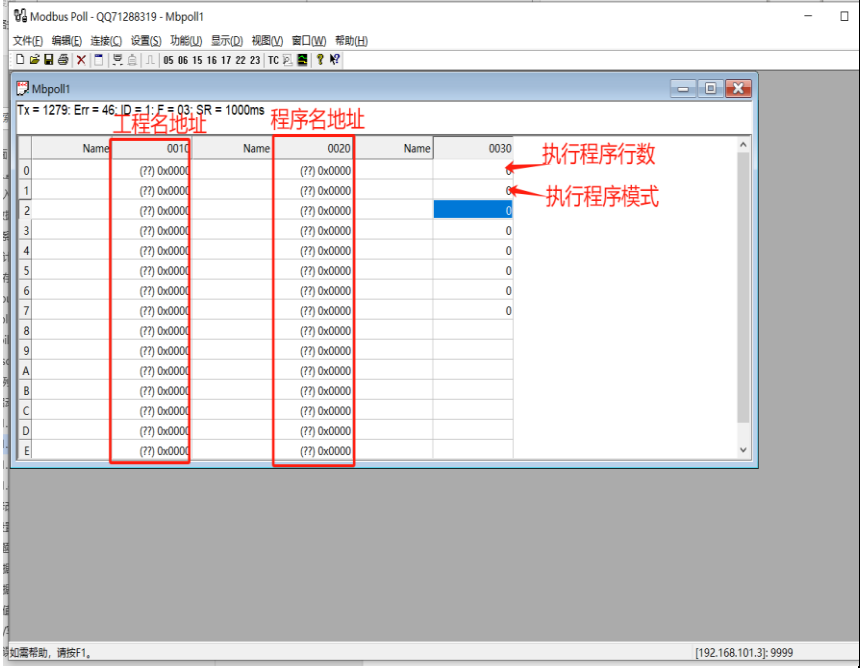
根据使用的功能码不同，需要找到对应的功能地址表进行对应；地址格式默认为 10 进制，必需改为 16 进制，否则数据地址将会错误；

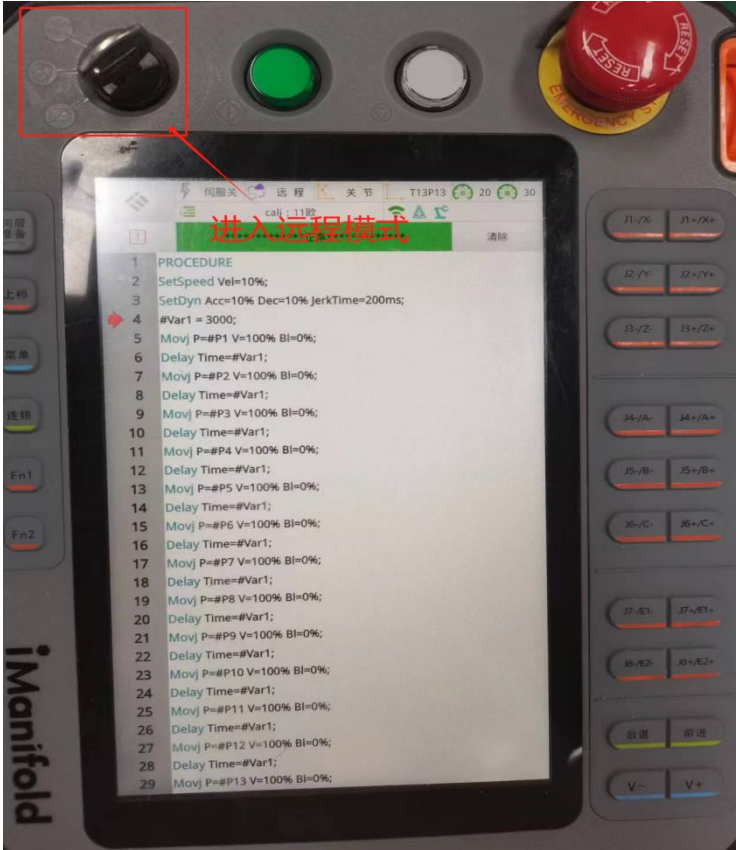
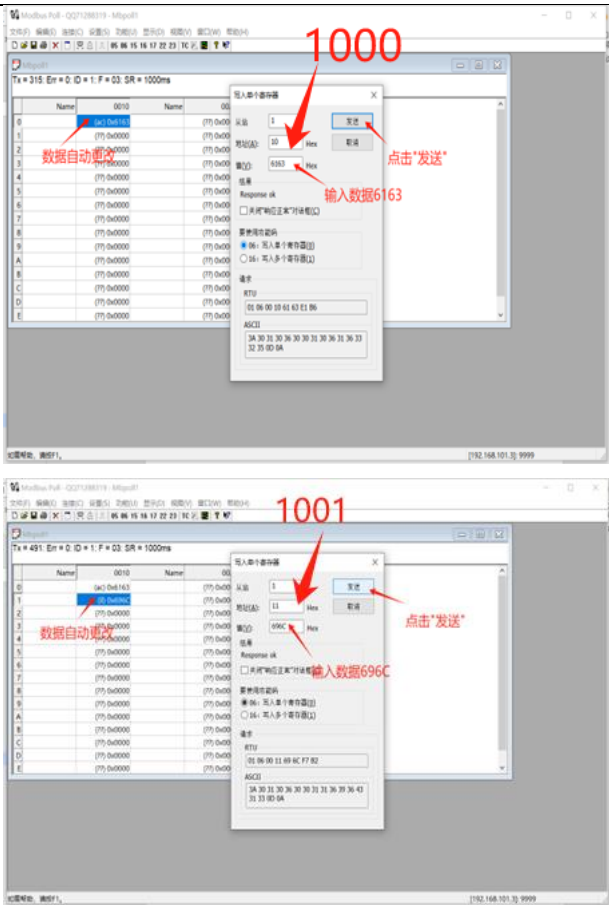
功能表地址切换

执行工程名↵	128-byte string↵	0x1000~0x103F↵	以字节流保存报警信息↵
执行程序名↵	128-byte string↵	0x1040~0x107F↵	以字节流保存报警信息↵
执行程序行数↵	32-bit integer↵	0x1080~0x1081↵	↵
执行程序模式↵	16-bit integer↵	0x1082↵	0：一次；↵ 1：步进；↵ 2：循环；↵

根据功能地址表找到执行工程名，执行程序名，执行程序行数，执行模式地址不同地址需要输入的数据类型不同，需要转换数据类型。需要使用转换工具转换为 16 进制。

转换工具 (<http://www.kjson.com/transform/ox2str/>)

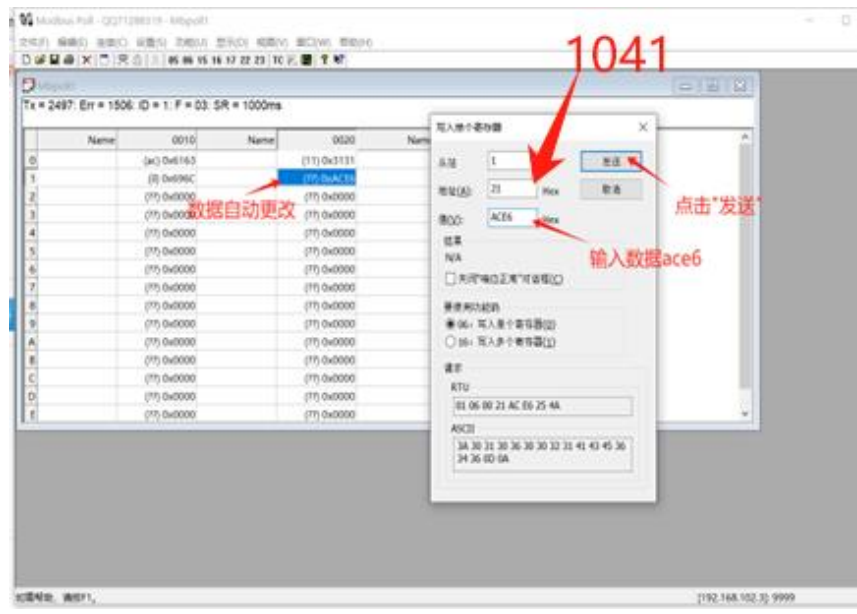
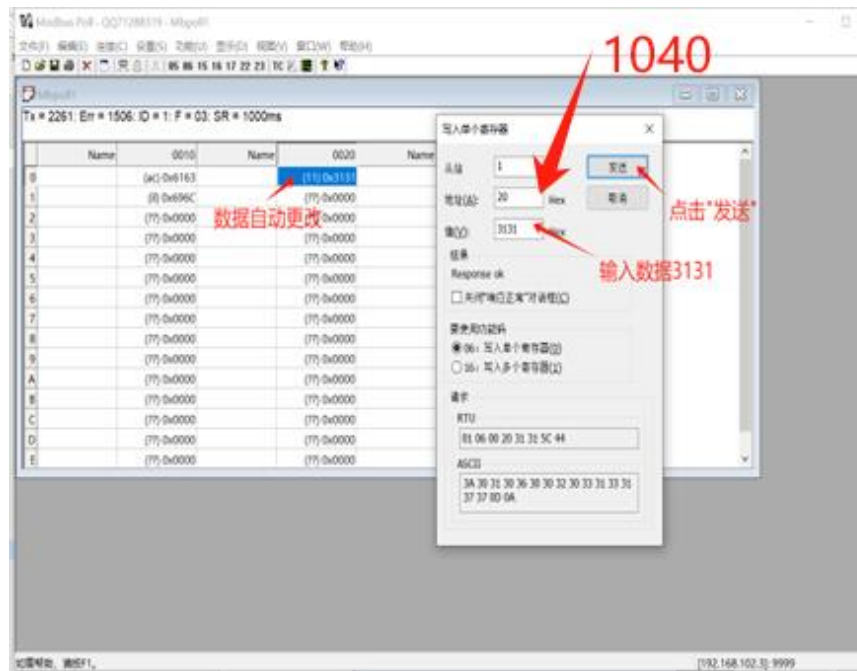
格式选择	 选中地址 1000、1040，更改地址格式为 “16 进制-ASCII”
数据地址	 根据地址表，找到 0x1080 执行程序行数、0x1082 执行程序模式地址进行设置

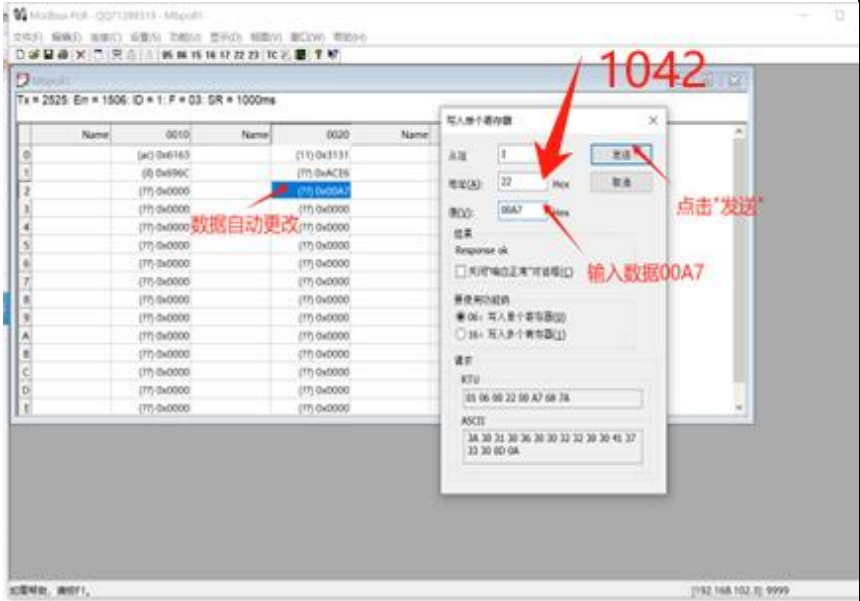
切换模式	
写入工程名	

本功能数据传输格式使用 little_endian_swap, 需要注意字节序。将工程名“cali”用转换工具转换为 16 进制码。

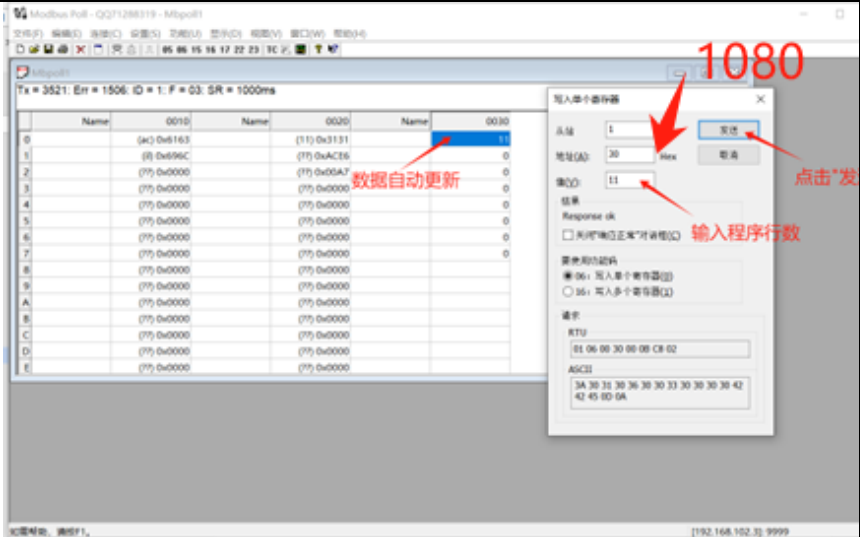
转换工具 (<http://www.kjson.com/transform/ox2str/>)

写入程序名

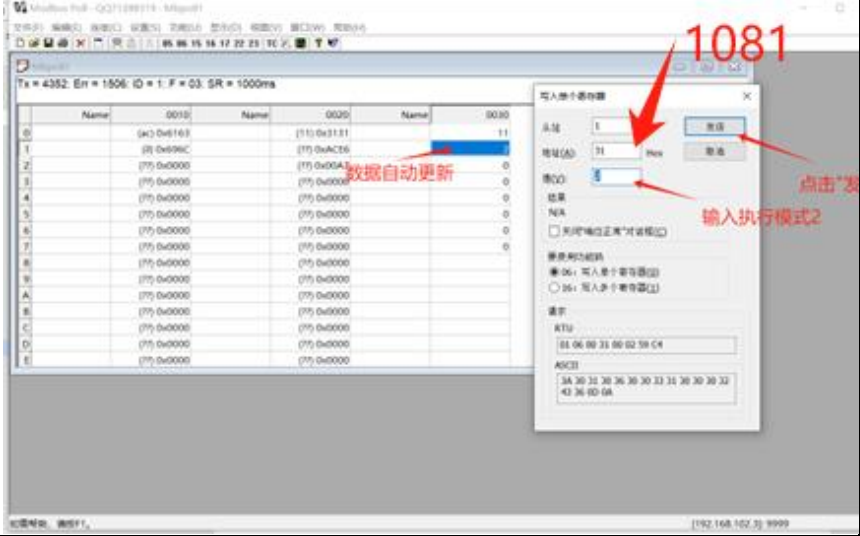


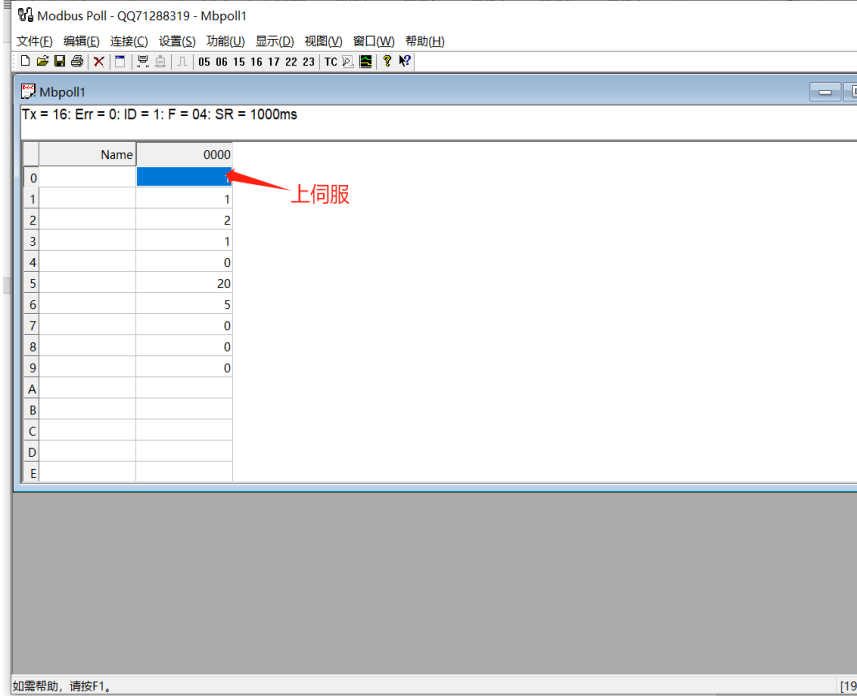


写入执行程序
行数



写入执行程序
模式



	<table border="1"> <thead> <tr> <th>名称</th><th>数据类型</th><th>地址</th><th>备注</th></tr> </thead> <tbody> <tr> <td>伺服使能</td><td>16-bit integer</td><td>0x0000</td><td> 上升沿有效，生效后自动复位； 如当前未使能，则该信号触发后伺服使能，如当前已经使能，则该信号触发后停去使能 </td></tr> </tbody> </table>  根据 Holding_Register 命令寄存器功能地址表，找到上伺服地址进行远程上伺服	名称	数据类型	地址	备注	伺服使能	16-bit integer	0x0000	上升沿有效，生效后自动复位； 如当前未使能，则该信号触发后伺服使能，如当前已经使能，则该信号触发后停去使能												
名称	数据类型	地址	备注																		
伺服使能	16-bit integer	0x0000	上升沿有效，生效后自动复位； 如当前未使能，则该信号触发后伺服使能，如当前已经使能，则该信号触发后停去使能																		
执行程序	<table border="1"> <tbody> <tr> <td>执行程序模式</td><td>16-bit integer</td><td>0x1082</td><td> 0: 一次； 1: 步进； 2: 循环； </td></tr> <tr> <td>执行程序</td><td>16-bit integer</td><td>0x1083</td><td>沿有效，生效后自动复位；该信号触发后执行程序</td></tr> <tr> <td>暂停程序</td><td>16-bit integer</td><td>0x1084</td><td>上升沿有效，生效后自动复位；该信号触发后暂停程序</td></tr> <tr> <td>继续执行</td><td>16-bit integer</td><td>0x1085</td><td>上升沿有效，生效后自动复位；该信号触发后停止程序</td></tr> <tr> <td>停止程序</td><td>16-bit integer</td><td>0x1086</td><td>上升沿有效，生效后自动复位；该信号触发后停止程序</td></tr> </tbody> </table>	执行程序模式	16-bit integer	0x1082	0: 一次； 1: 步进； 2: 循环；	执行程序	16-bit integer	0x1083	沿有效，生效后自动复位；该信号触发后执行程序	暂停程序	16-bit integer	0x1084	上升沿有效，生效后自动复位；该信号触发后暂停程序	继续执行	16-bit integer	0x1085	上升沿有效，生效后自动复位；该信号触发后停止程序	停止程序	16-bit integer	0x1086	上升沿有效，生效后自动复位；该信号触发后停止程序
执行程序模式	16-bit integer	0x1082	0: 一次； 1: 步进； 2: 循环；																		
执行程序	16-bit integer	0x1083	沿有效，生效后自动复位；该信号触发后执行程序																		
暂停程序	16-bit integer	0x1084	上升沿有效，生效后自动复位；该信号触发后暂停程序																		
继续执行	16-bit integer	0x1085	上升沿有效，生效后自动复位；该信号触发后停止程序																		
停止程序	16-bit integer	0x1086	上升沿有效，生效后自动复位；该信号触发后停止程序																		

Modbus-64 - 0271288119 - All Rights Reserved

主地址: 0000 | 从地址: 0000 | 波特率: 9600 | 数据位: 8 | 停止位: 1 | 校验位: 无

Tx = 546; Err = 0; ID = 1; F = 03; SR = 1000ms

Name	0000	Name	00
0	(R) 0x0163	(11) 0x11	
1	(R) 0x096C	(77) 0xA0	
2	(77) 0x0000	(77) 0x00	
3	(77) 0x0000	(77) 0x00	
4	(77) 0x0000	(77) 0x00	
5	(77) 0x0000	(77) 0x00	
6	(77) 0x0000	(77) 0x00	
7	(77) 0x0000	(77) 0x00	
8	(77) 0x0000	(77) 0x00	
9	(77) 0x0000	(77) 0x00	
A	(77) 0x0000	(77) 0x00	
B	(77) 0x0000	(77) 0x00	
C	(77) 0x0000	(77) 0x00	
D	(77) 0x0000	(77) 0x00	
E	(77) 0x0000	(77) 0x00	

写入单个寄存器

从站: 1

地址(A): 1083

值(V): 1

结果: N/A

☐ 关闭"响应正常"对话框(C)

要使用寄存器

☒ 写入单个寄存器(W)

☐ 写入多个寄存器(L)

地址

RTU: 00 06 00 32 00 01 09 C5

ASCII: 3A 30 31 30 36 30 30 33 32 30 30 30 31 42 34 00 0A

1083

写入1

地址表

根据地址表，找到 0x1083 地址，写入数值 1，程序执行

4 常见问题

4.1 数据类型的转换

Modbus poll 不支持字符串、ASCII 码需要转换为 16 进制。

中文、中文符号需要使用字符串转 16 进制工具。

例：“圈” → “e59c88”

转换工具 (<http://www.kjson.com/transform/ox2str/>)

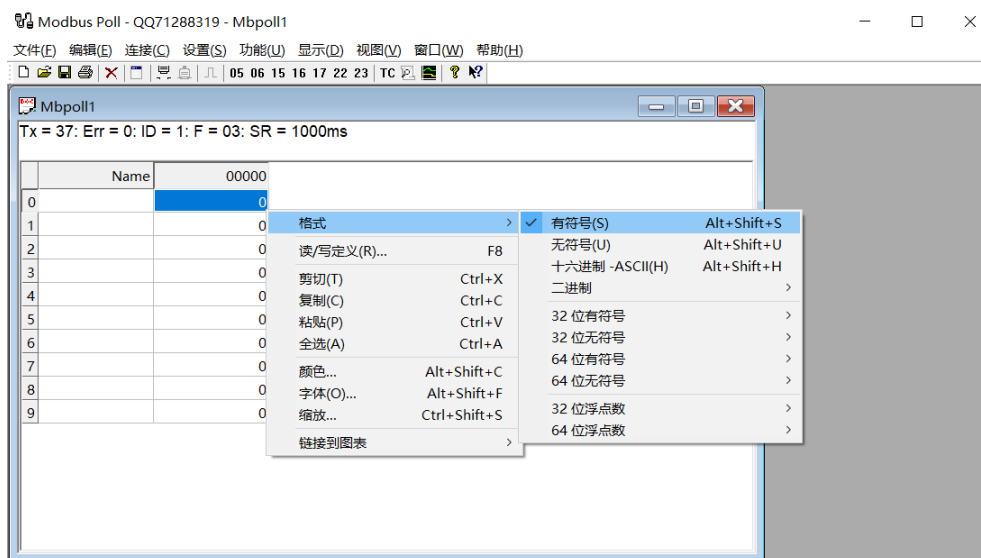
ASCII 码需要使用 ASCII 码转 16 进制工具。

例：“cali” → “63616C69”

4.2 数据格式

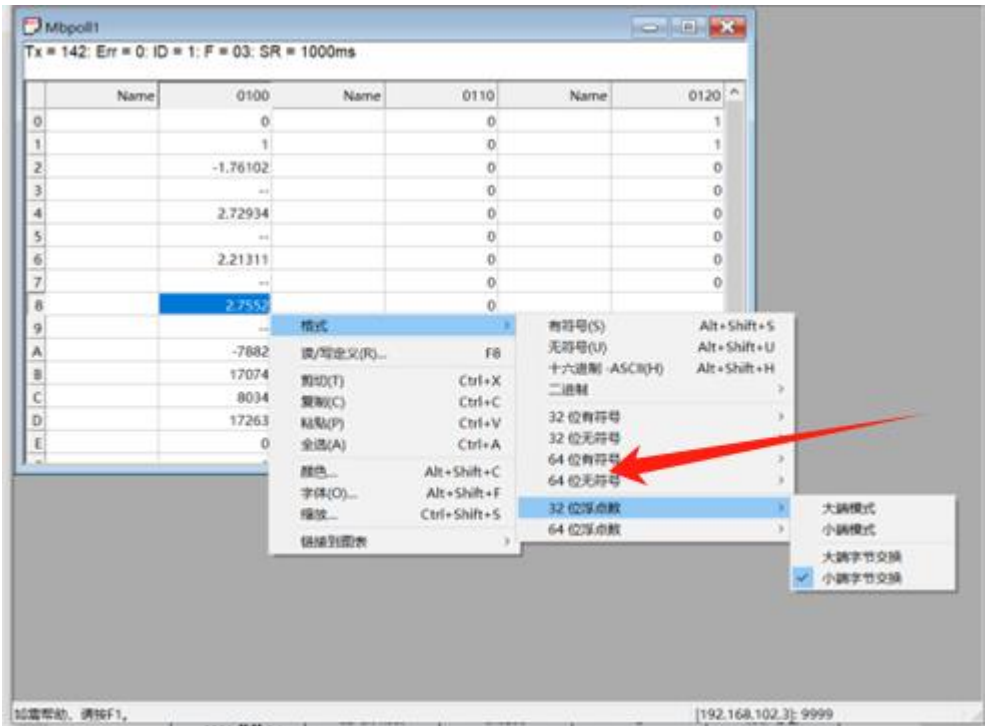
本功能数据传输格式使用 little_endian_swap。功能表数据类型分为 16-bit integer、64-bit double、32-byte string、128-byte string 查看数据时需要注意数据类型。

Modbus Poll 默认为 “16-bit integer (有符号)”。



如果数据类型为 “64-bit float”，则需要将格式设置为 “64 位浮点数” → “小端字节”

交换”才可查看数据。



4.3 数值的输入

地址数值输入需要看功能码的备注栏进行输入，上升沿有效命令字只能输入 1。

字符串数据输入到地址需要更改格式为“16 进制-ASCII”，再双击需要更改的数值即可修改（使用“写入单个寄存器”功能默认是 10 进制输入，无法写入 16 进制数据）。

输入 16 进制码时，注意 4 个字为 1 组且需要注意字节序，数据超过 4 位时，需要输入多个地址。

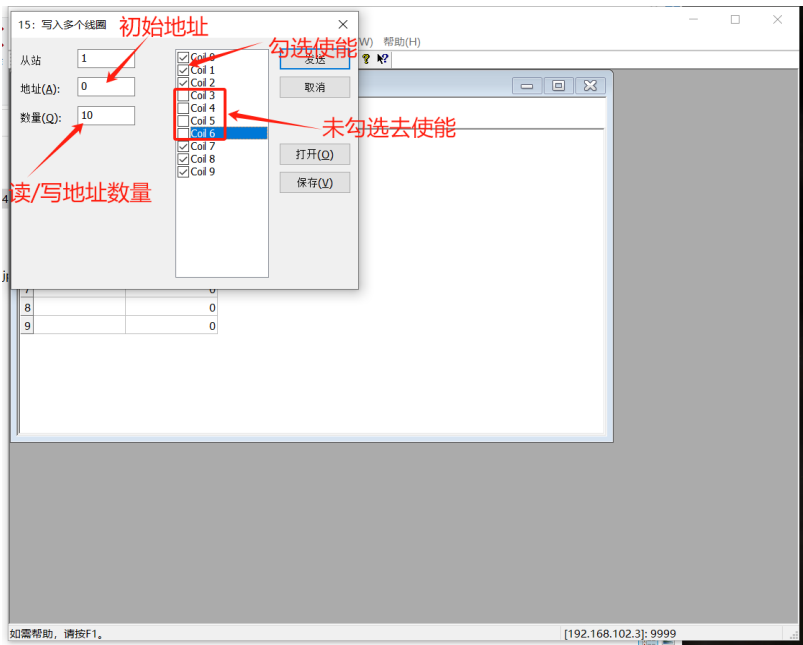
例：“圈” → “。e59c88” 输入地址时写成 “0x9ce5” “0x0088”

4.4 读/写定义

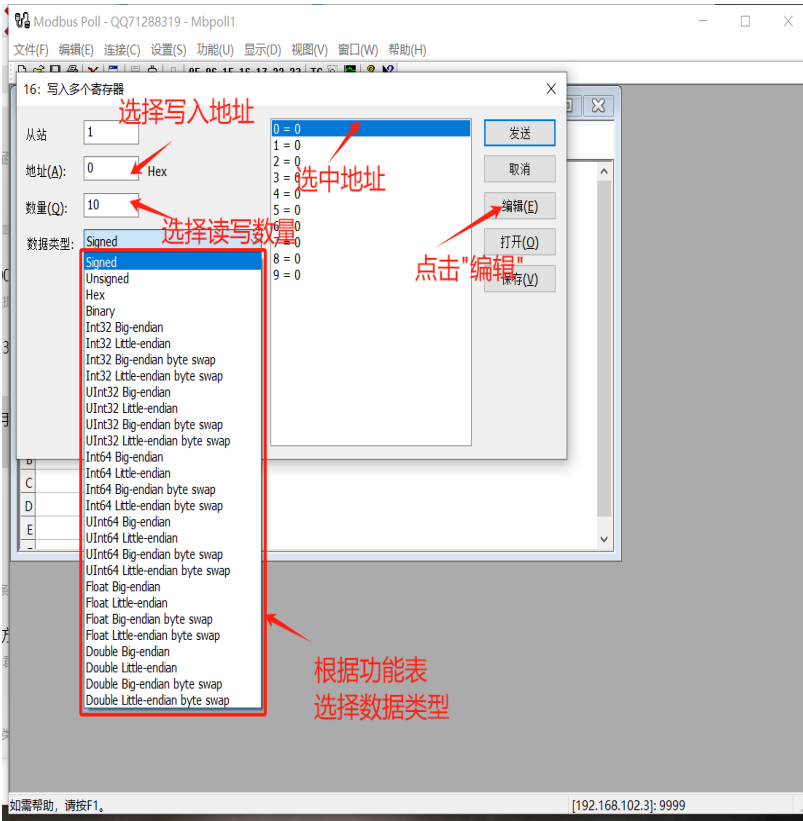
读/写定义中，从站固定为 1；功能码共 4 个：01、02、03、04；地址格式默认为 10 进制，需要改为 16 进制，否则数据地址将会错误；扫描速率默认 1000ms

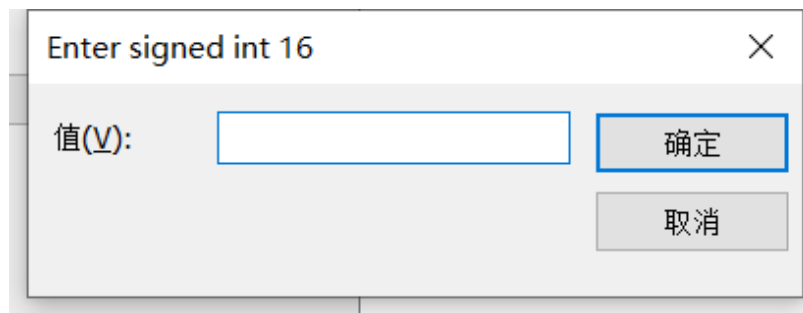
4.5 写入多个线圈/多个寄存器

Modbus 支持多个线圈/寄存器输入，例：多个线圈使能



例：多个寄存器写入





4.6 Modbus poll 错误码

No connection	正常未连接的状态, 正常 COM 口未被占用的情况下连接即可消除。
Timeout	所有由软件发出指令, 从机设备未回复的情况, 都会显示 Timeout。而从机设备不回复, 可能性有很多, 比如: 1、连接配置错误, 主机的波特率、Slave ID 等信息跟从机设备对应不上, 从机就不会回。 2、线路异常, 电脑跟从机设备之前的通信线存在异常, 也是无法正常收到回复。 3、从机设备解析异常不回复, 这个具体可以查看 Modbus 协议详解。
Write error/Read error	使用的是 USB 转 485 的工具, 调换 485 的 A、B 线, 可能会出现这种情况。 使用的是 USB 转 232 或 TTL 的工具, 则短接 Tx 和 Rx 就会出现这种情况。 使用网线连接时, 断开网线会出现这种情况。 在发送的过程中, 收到数据, 即总线上数据冲突, 也会出现类似的错误。
Checksum Error	从机设备返回的 CRC 校验不正确。

	通信总线上存在干扰。 连接配置里的校验码、数据位、停止位配置错误。
Insufficient bytes received	接收的字节数不完整, 可能由于线路或某种原因, 导致返回的指令长度与理论上返回的指令长度对不上, 则会报些错误。
Illegal Function	功能码异常, 一般是当访问的从机设备不存在可操作的功能码时, 会返回不存在此功能码不存在的 01 异常码, 软件接收到此指令时就会报出这个错误。
Illegal Data Address	地址异常, 一般是当访问的从机设备不存在要读取的寄存器/线圈地址时, 会返回不存在此地址的 02 异常码, 软件接收到此指令时就会报出这个错误。
Illegal Data Value	数据异常, 一般是当前所要读/写的数 据, 从机设备不允许此地址数据的操作, 会 返回数据不可操作的 03 异常码, 软件接收 到此指令时就会报出这个错误。
Slave Device Busy	从机繁忙, 一般是从机设备回复了 06 的异常码, 软件接收到此指令时就会报出 这个错误。
Response Error	响应异常, 一般是从机设备回复了一些

	无法识别的指令，就会报这个异常，另外一个不知道算不算是这个软件的 Bug，一般广播指令（0 地址）是不可以读数据，只能广播写，但这个软件可以设置用广播指令读取，当这样设置时，就会出现这个 "Response Error" 的异常。
--	--