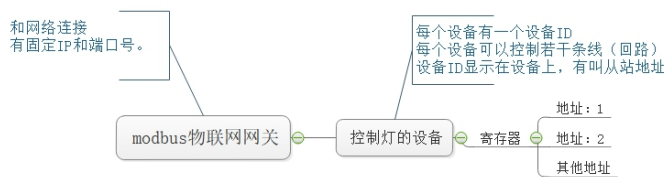


Modbus 协议简单讲解和测试

扶程星云 20210313

前言

首先，以一张图来说明 Modbus 在和灯具交互的说明



寄存器地址概念：

一个地址即所谓的控制回路，可以理解为一个地址控制一条线。

但是1和2分别是该寄存器所有控制回路全关和全开，给寄存器1写入0，表示全关，给寄存器2写入1表示全开，该寄存器地址内的内容属于无效数据。剩下的地址是每条回路，读取的时候值为1表示开，0表示关，写入1控制该回路的灯全开，写入0控制该回路的灯全关。

刚开始接触 Modbus 协议有个思维误区，以为从站是接收的和被动的，实际上在某种角度看是有误会的，从站上的寄存器地址存放的是设备传过来的值，我们读取的也是从站寄存器地址上的值，特别是单向设备的值：即仅发送自身的值，比如温湿度，电表的电压、电流等。从站上的寄存器是和最终设备互动的，比如灯开了，从站寄存器对应的值就是 1，灯灭了，从站寄存器地址的值就是 0。

我们可以通过寄存器地址的值来控制这个设备开关，比如设置为 1，灯就会执行开灯动作，设置为 0，灯就会执行关灯动作。

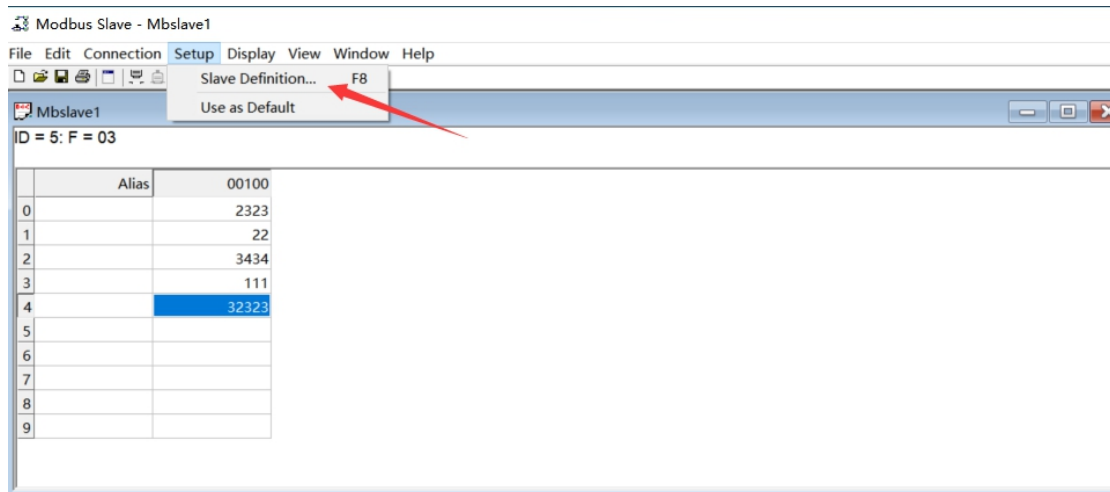
简易测试

ModbusSlave 是一个从站设备仿真软件，它用于接收主设备的命令包，并回送数据包；可用于测试和调试 Modbus 主站设备，便于观察 Modbus 通信过程中的各种报文。

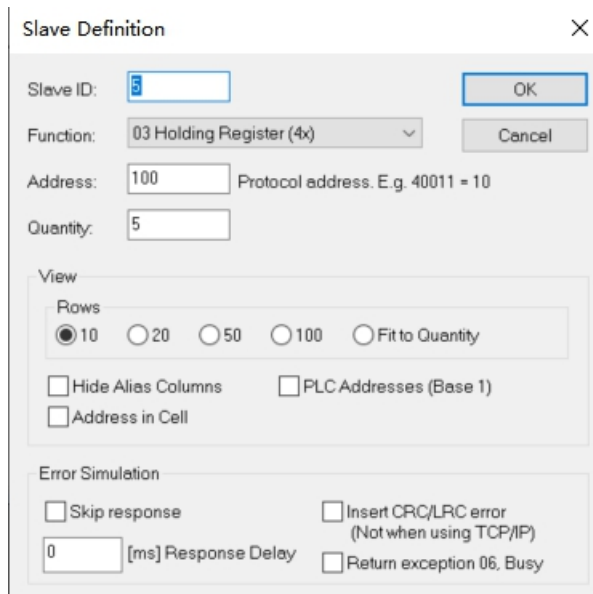
Modbus Poll：Modbus 主机仿真器，用于测试和调试 Modbus 从设备。

1、我们用 Modbus Slave 做一个设备仿真器。

1.1 我们设置从站信息：

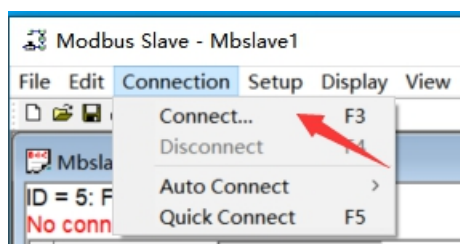


1.2 设置从站信息如下：

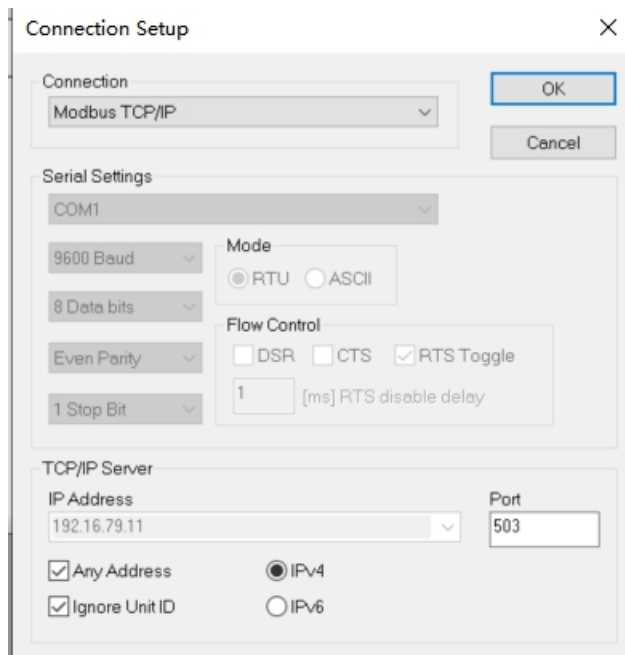


1.3 我们需要设置从站地址->Slave ID,存放模式->Function, 存放的起始地址->Address, 数量->Quantity, 其他的先用默认的设置。然后点击 OK。

1.4 点击 Connection 设置连接。



1.5 主要设置的是连接方式->Connection, 端口号->Port, 然后点击 OK。



1.6 在列表中修改对应的值。

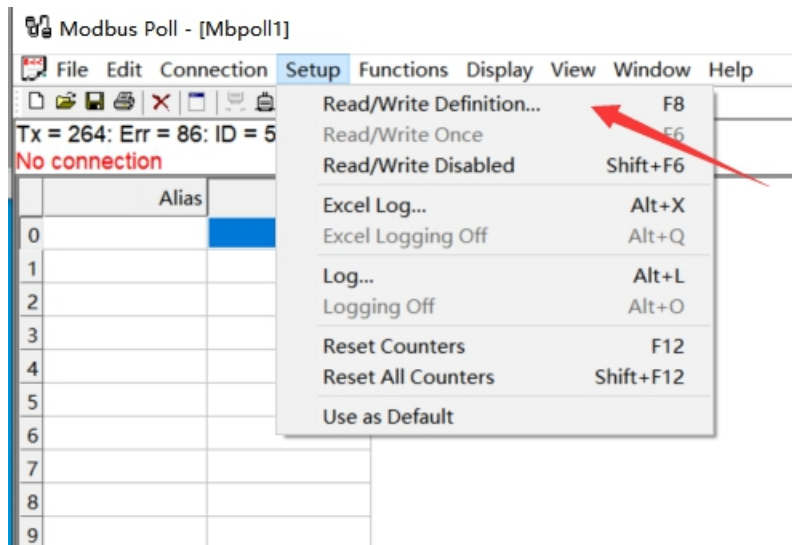
Mbslave1

ID = 5: F = 03

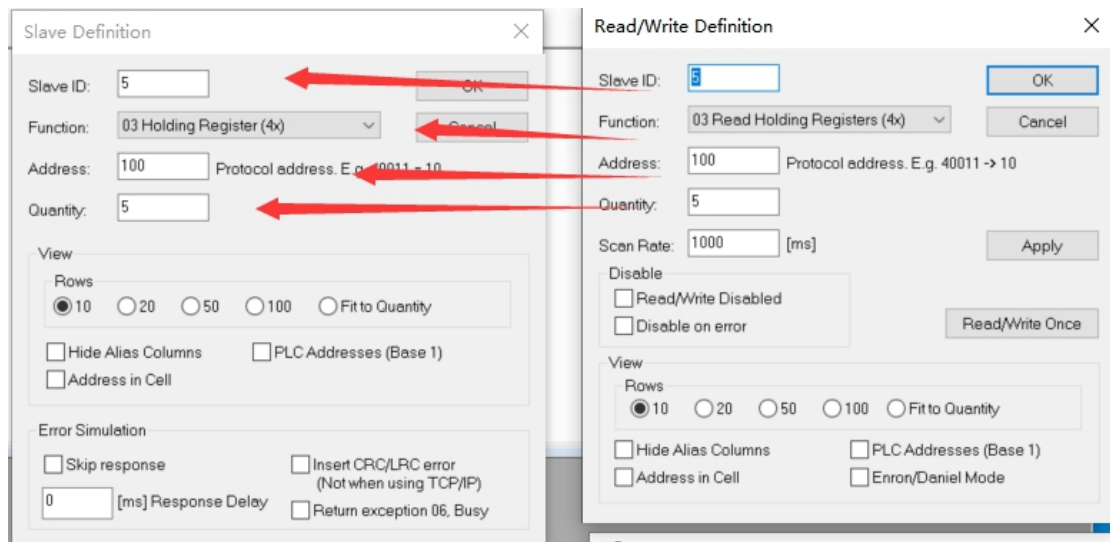
	Alias	
		00100
0		2323
1		22
2		3434
3		111
4		32323
5		
6		
7		
8		
9		

2、接着，我们设置 MObus Poll 来读取数据。

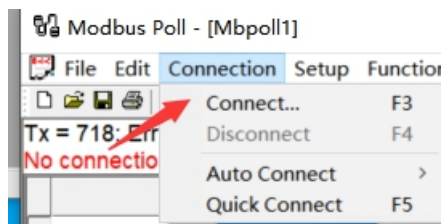
2.1 首先设置读取模式。



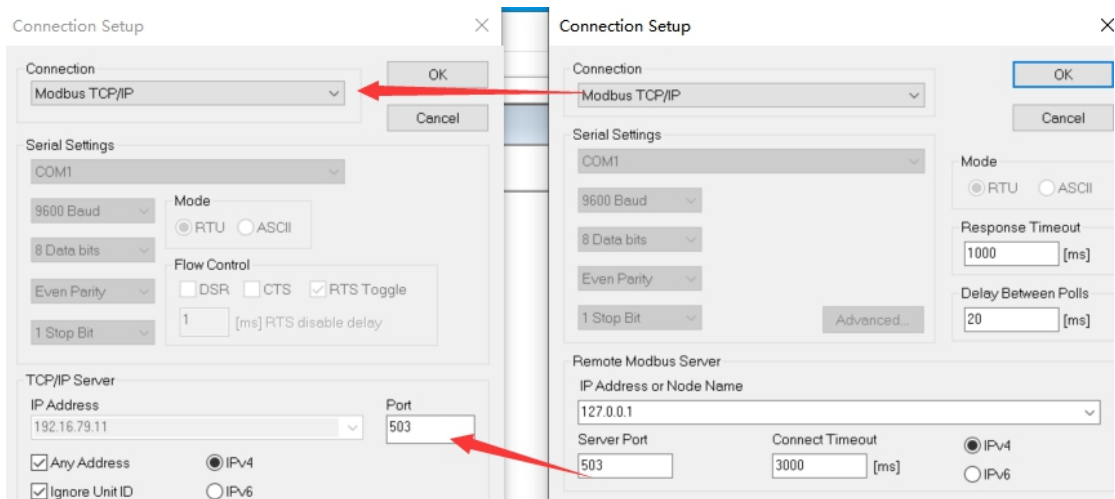
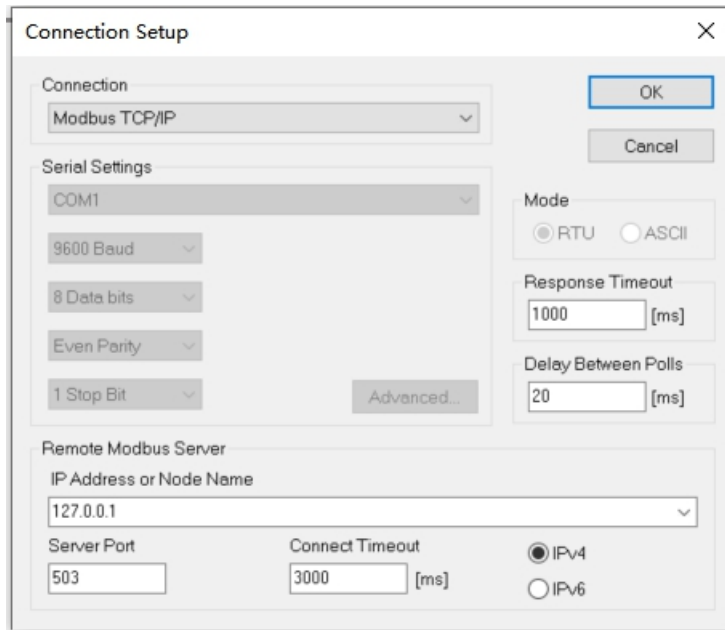
2.2 参考之前从站信息来设置。



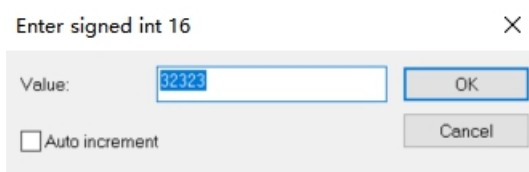
2.3 打开连接。



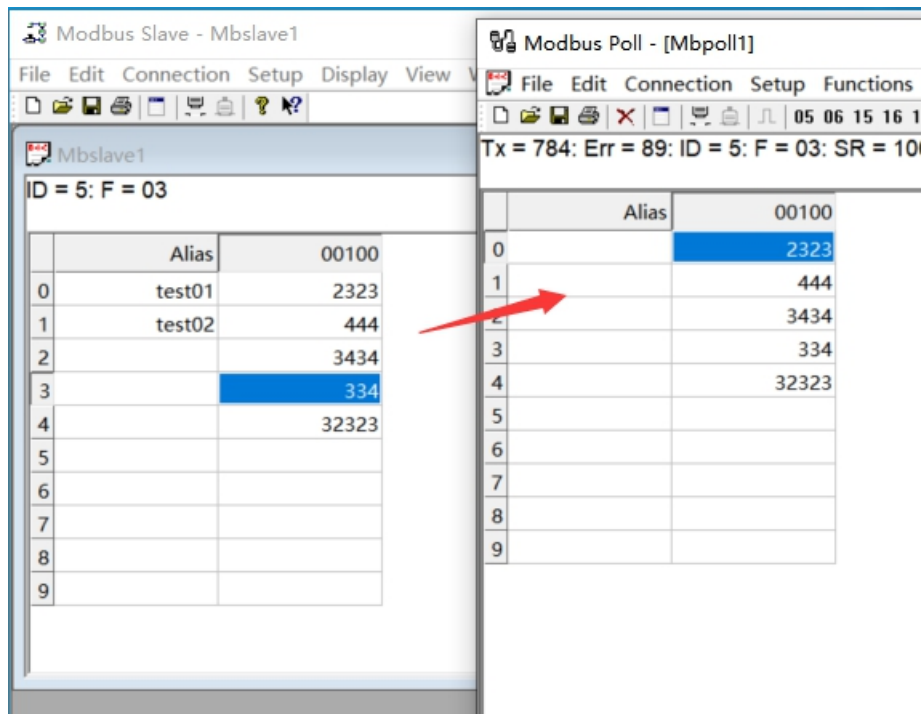
2.4 设置连接信息，因为是本机所以 IP 地址是 127.0.0.1，服务器端口号也是之前设置的 503，其余默认，点击 OK 键。



2.5 我们双击左边的值，在弹出的窗口填写值。



2.6 右边的 Modbus Poll 会实时更新值。



3、我们以 MThings 为例，开始做测试。

3.1 首先移除所有链接和设备



3.2 新建网络连接（服务器）



3.2.1 先设置一个 TCP 服务器，采用 MODBUS-TCP 传输协议，本地端口是 502，然后点击确认按钮。

网络参数配置

参数名	数值
链接名称	NET001
链接模式	TCP服务器
链接空闲保持时间 (秒)	6000
传输模式	MODBUS-TCP(同步)
本地端口	502
目标IP	127.0.0.1
目标端口	--

确认 取消

3.3 新增设备

3.3.1 在链路上点击设备操作的添加按钮。

全部链接 刷新 新增网络链接 移除网络链接

链接	链接状态	链接参数	链接操作	设备起始地址	设备结束地址	设备操作	备注
<input type="checkbox"/> NET001	已关闭	配置	启动	1	1	添加	

3.3.2 模拟一个从机(设备仿真)

请选择添加的设... ? X

模拟主机 模拟从机

确定 取消

3.3.3 设置一些值，点击数据，点击配置。

MThings 高效工作，快乐生活

链接 数据 自定义 统计 辅助 关于

恢复 禁止 移除 >> 输入查找名称 配置 筛选列 筛选行 筛选区块 解析报文 曲线

序号	名称	ID	名称	数值	单位	区块	地址	数量
001	[S]NET001-001							

3.3.4 点击新增按钮



3.3.5 配置 5 条数据，起始地址是 100，点击确定。



3.3.6 出来以下列表，继续点击配置按钮后保存当前配置。


ID	名称	数值	单位	区块	地址	数量	位偏移	位数	系数	范围	曲线	传输类型	呈现类型	字节序	字序	间隔时间(ms)
1	--	0	--	保持寄存器(RW)	100	1	0	16	1	--	<input type="checkbox"/> 未选择	UINT	FLOAT	大端	大端	0
2	--	0	--	保持寄存器(RW)	101	1	0	16	1	--	<input type="checkbox"/> 未选择	UINT	FLOAT	大端	大端	0
3	--	0	--	保持寄存器(RW)	102	1	0	16	1	--	<input type="checkbox"/> 未选择	UINT	FLOAT	大端	大端	0
4	--	0	--	保持寄存器(RW)	103	1	0	16	1	--	<input type="checkbox"/> 未选择	UINT	FLOAT	大端	大端	0
5	--	0	--	保持寄存器(RW)	104	1	0	16	1	--	<input type="checkbox"/> 未选择	UINT	FLOAT	大端	大端	0

3.3.7 要修改哪个值就双击数值列中的值。

3.3.7 关于

输入查找名称  配置 筛选列 筛选行 筛选区块 解析报文 曲线

ID	名称	数值	单位	区块	地址	数量	位偏
1	--	0	--	保持寄存器(RW)	100	1	0
2	--	0	--	保持寄存器(RW)	101	1	0
3	--						
4	--						
5	--						

 选择模拟数据方法

参数名	数值	说明	示例
策略	固定值		
固定值	0		

应用 返回

3.3.8 选择值的策略和值，我们分别设置下如下图。

输入查找名称  配置 筛选列 筛选行 筛选区块 解析报文 曲线

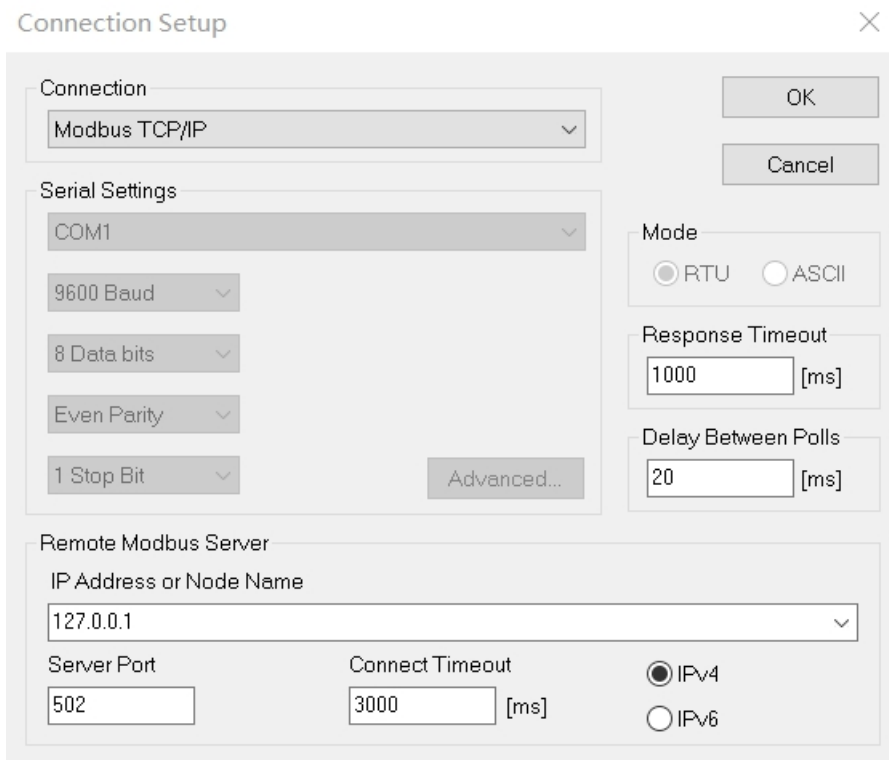
ID	名称	数值	单位	区块	地址	数量	位偏移	位数	系数
1	--	--	--	保持寄存器(RW)	100	1	0	16	1
2	--	555	--	保持寄存器(RW)	101	1	0	16	1
3	--	6767	--	保持寄存器(RW)	102	1	0	16	1
4	--	6761	--	保持寄存器(RW)	103	1	0	16	1
5	--	9889	--	保持寄存器(RW)	104	1	0	16	1

3.3.9 我们也可以设置设备的从地址，双击[S]NET001-001，表示从站的设备，这里可以修改设备地址，也就是从站地址。

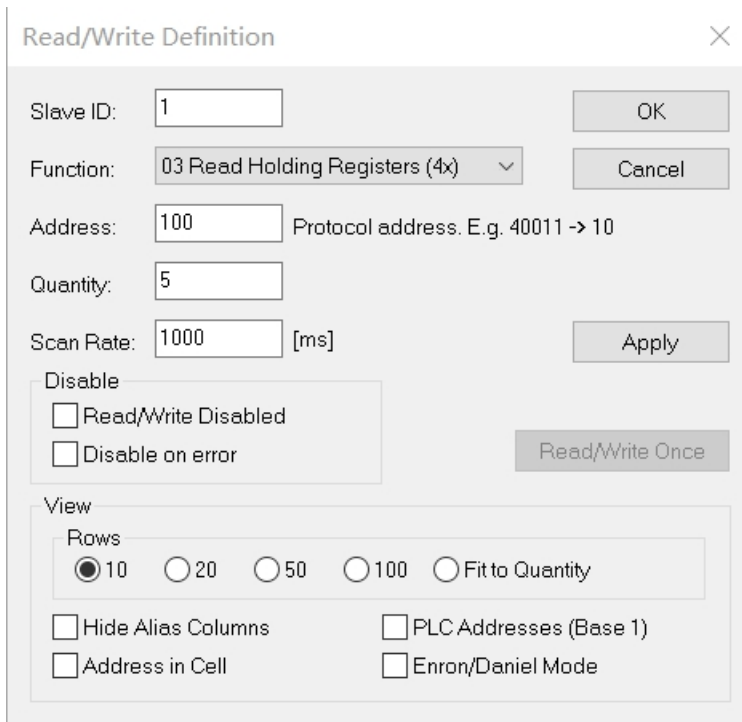


4、我们用 Modbus Poll 来测试

4.1 设置 IP 地址和 Port

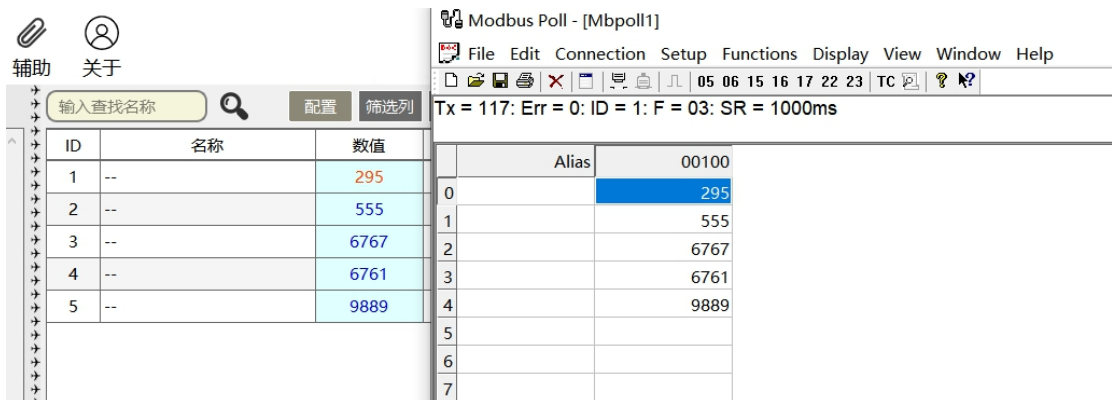


4.2 设置从站地址和起始地址。



4.3 记得点击连接。

4.4 然后，就可以看到 Modbus Poll 会随着从站里面寄存器的值的变化而变化了。



5、我们来看 MThings 的主站功能。

5.1 点击链接->新增网络链接，设置链接模式是 TCP 客户端，传输模式是 MODBUS-TCP（同步），目标地址是从站的地址，目前是本机，目标端口是 502，这些设置都要根据从站设备来对应设置。

网络参数配置

参数名	数值
链接名称	NET002
链接模式	TCP客户端
重新建链周期 (秒)	0
链接空闲保持时间 (秒)	6000
传输模式	MODBUS-TCP(同步)
本地端口	--
目标IP	127.0.0.1
目标端口	502

确认 取消

5.2 确认后，同样点击设备操作的添加按钮来添加设备，这里选择的是模拟主机。

请选择添加的设... ? ×

模拟主机 模拟从机

确定 取消

恢复 禁止 移除 >>

序号	名称
<input type="checkbox"/> 001	 [S] NET001-001
<input type="checkbox"/> 002	 [M] NET002-001

变更设备关键信息

参数名	数值
设备名称	[M]NET002-001
链接	NET002
设备地址	1

同步链接至其它设备 确认 取消

注意从站地址要正确。

5.3 数据读取，点击数据，点击主站[M]NET002-001，点击配置和新增，在新增数据配置输入和从站相同的数据配置，然后再点击配置保存。

新增数据配置

参数	数值
配置条数	5
起始数据地址	100
插入模板行下方	<input type="checkbox"/> 选择

确定 取消

ID	名称	数值	单位	读	指令	写	区块	地址	数量	位偏移	位数	系数	范围	批量读	批量写	曲线	传输类型	呈现类
1	--	--	--	读	--	写	保持寄存器(RW)	100	1	0	16	1	--	<input checked="" type="checkbox"/> 已选择	<input type="checkbox"/> 未选择	<input type="checkbox"/> 未选择	UINT	FLOA
2	--	--	--	读	--	写	保持寄存器(RW)	101	1	0	16	1	--	<input checked="" type="checkbox"/> 已选择	<input type="checkbox"/> 未选择	<input type="checkbox"/> 未选择	UINT	FLOA
3	--	--	--	读	--	写	保持寄存器(RW)	102	1	0	16	1	--	<input checked="" type="checkbox"/> 已选择	<input type="checkbox"/> 未选择	<input type="checkbox"/> 未选择	UINT	FLOA
4	--	--	--	读	--	写	保持寄存器(RW)	103	1	0	16	1	--	<input checked="" type="checkbox"/> 已选择	<input type="checkbox"/> 未选择	<input type="checkbox"/> 未选择	UINT	FLOA
5	--	--	--	读	--	写	保持寄存器(RW)	104	1	0	16	1	--	<input checked="" type="checkbox"/> 已选择	<input type="checkbox"/> 未选择	<input type="checkbox"/> 未选择	UINT	FLOA

5.4 我们可以看到左侧这个设备是没有连接的。



5.5 继续点击链路菜单，点击该路链接的启动按钮。



这里需要注意的是，如果之前 MODBUS Poll 在连接中，请先中断连接。显示已连接。



5.6 点击主站设备，点击数据，然后点击读的按钮，就会读取数据出来。



5.7 我们也可以点批量读，然后循环，这个设备就会不停的读数。我们可以通过修改从站寄存器地址的值来看是否变化。

筛选区块		解析报文		曲线		批量读		批量写	
读	指令	写	区块	地址	数				
读	--	写	保持寄存器(RW)	100					
读	--	写	保持寄存器(RW)	101					
读	--	写	保持寄存器(RW)	102					
读	--	写	保持寄存器(RW)	103					
读	--	写	保持寄存器(RW)	104					



关于

输入查找名称		配置	筛选列	筛选行	筛选区块	解析报文	曲线	批量读	批量写					
ID	名称	数值	单位	读	指令	写	区块	地址	数量	位偏移	位数	系数		
1	--	369	--	读	--	写	保持寄存器(RW)	100	1	0	16	1		
2	--	555	--	读	--	写	保持寄存器(RW)	101	1	0	16	1		
3	--	666	--	读	--	写	保持寄存器(RW)	102	1	0	16	1		
4	--	6761	--	读	--	写	保持寄存器(RW)	103	1	0	16	1		
5	--	9889	--	读	--	写	保持寄存器(RW)	104	1	0	16	1		

5.8 我们可以在指令里面填数，然后点击写的按钮，回写从站寄存器中对应的值，同时读回来该值。

关于

输入查找名称		配置	筛选列	筛选行	筛选区块	解析报文	曲线	批量读	批量写					
ID	名称	数值	单位	读	指令	写	区块	地址	数量	位偏移	位数	系数		
1	--	255	--	读	--	写	保持寄存器(RW)	100	1	0	16	1		
2	--	22	--	读	22	写	保持寄存器(RW)	101	1	0	16	1		
3	--	666	--	读	--	写	保持寄存器(RW)	102	1	0	16	1		
4	--	6761	--	读	--	写	保持寄存器(RW)	103	1	0	16	1		
5	--	9889	--	读	--	写	保持寄存器(RW)	104	1	0	16	1		

附录:

以下信息来自网络搜索结果。

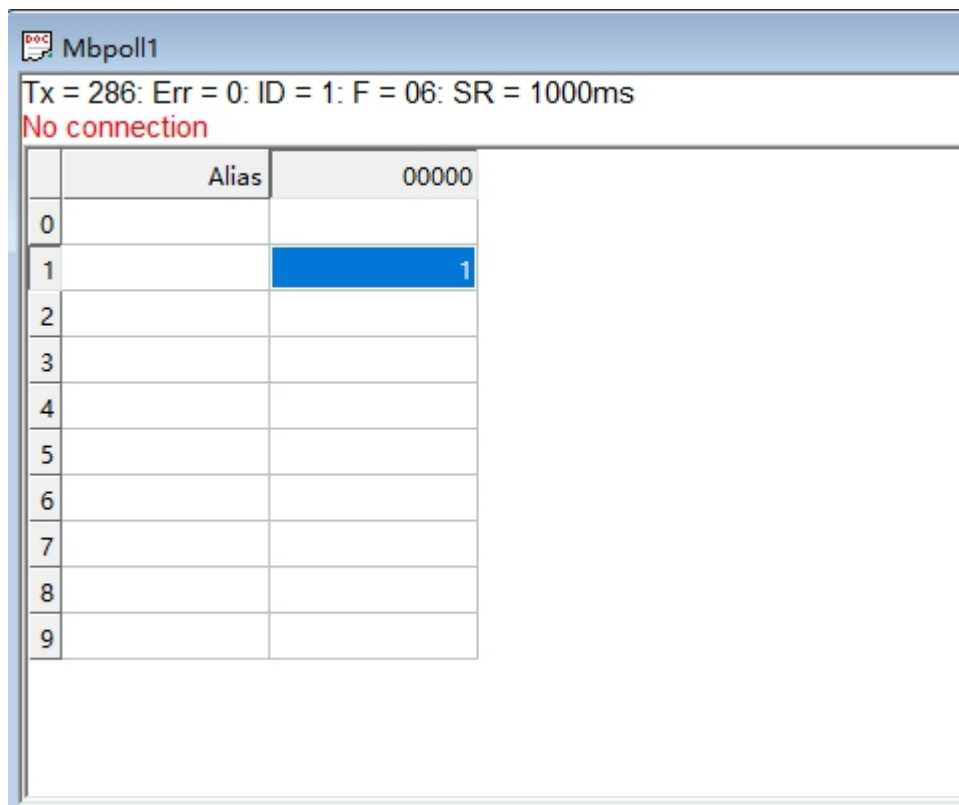
Modbus 调试工具之 ModbusPoll 的使用方法

1: 打开软件进入主页面后点击该软件页面第二行“Connection”并选择“Connect.. F3”

2: (1) 执行上一步操作后会弹出“Connection Setup”界面，在“Connection”下拉框根据对接文档选择对应的协议(一般为 Modbus TCP/IP)

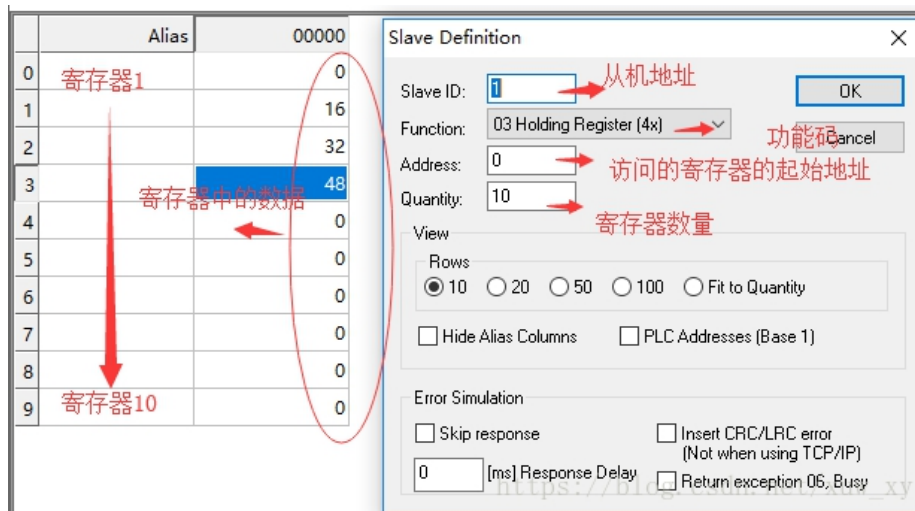
(2) 在“IP Address or Node Name”下的输入框中输入网关的 IP 地址

(3) 在“Server Port” 根据对接文档输入对应端口(默认为 502)，点击“OK”



如果连接效果如上图，显示红色“**No connection**”，可能是地址输入错误或者端口有误，如果对照文档确认无误仍是这样，那基本上就是设备出了问题，如电线未连接好等原因，可以联系设备厂家或者安装方询问原因并协调解决

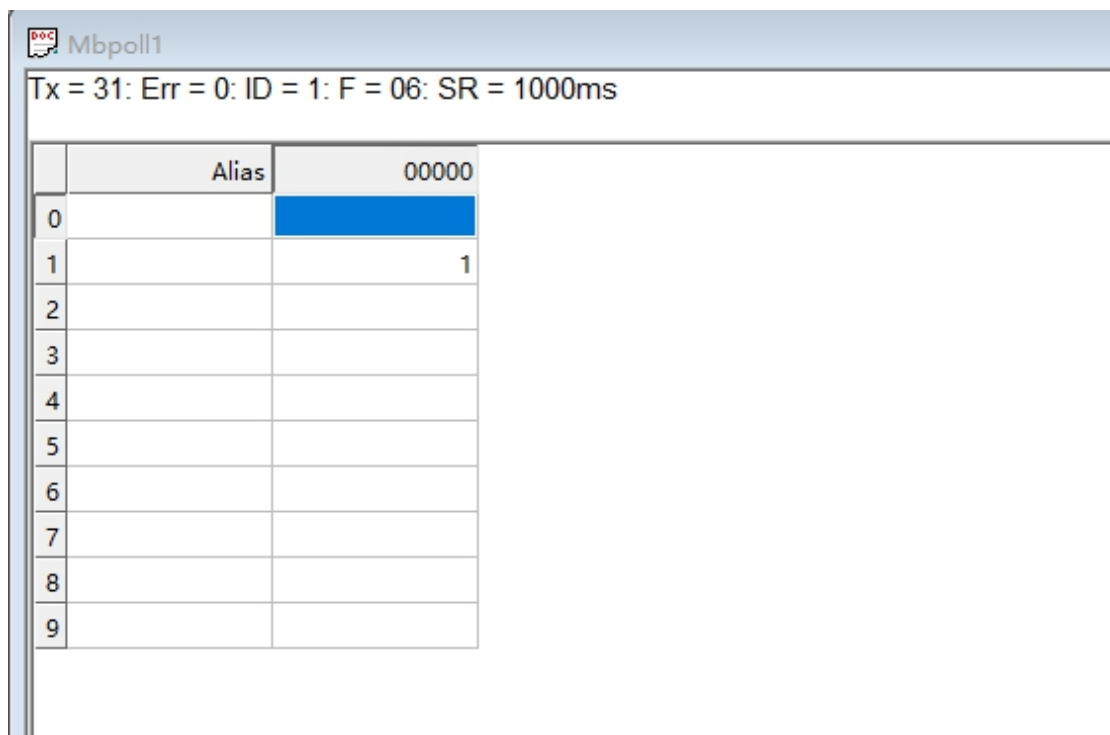
3: 在主页面“Setup”下选择第一个“Read/Write Definition...”



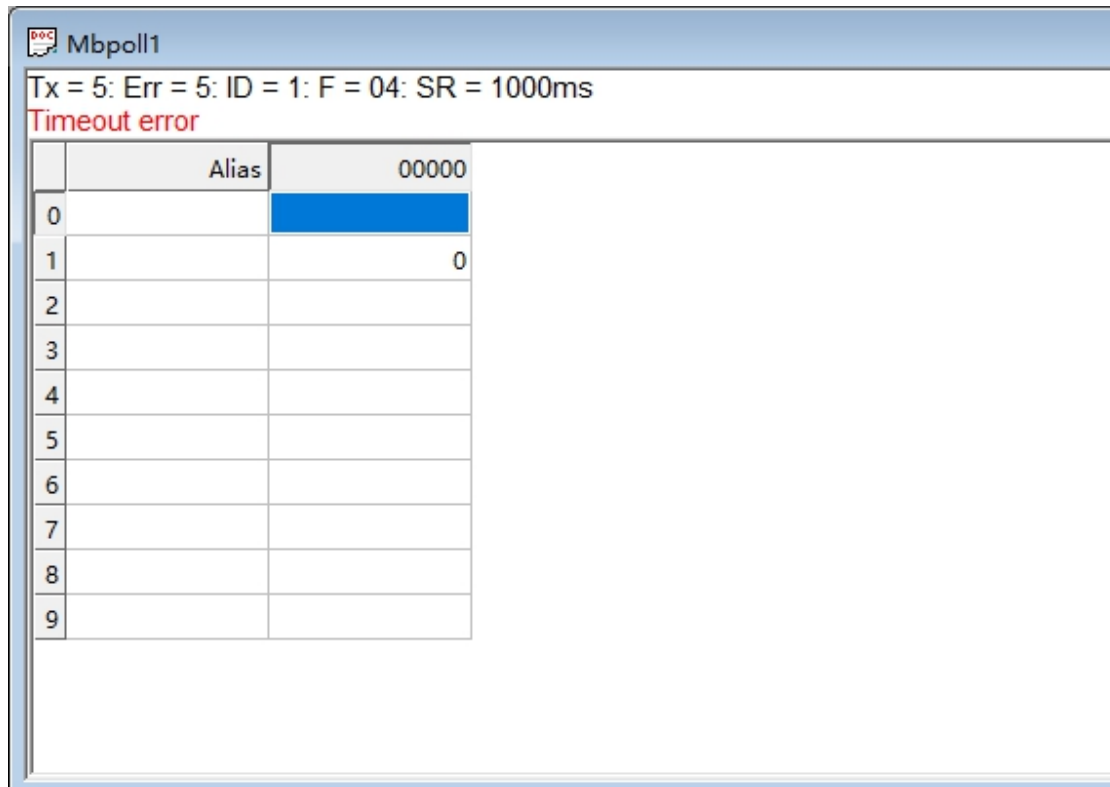
- (1) “Slave ID” 为从站地址，按照对接文档输入相应的值
- (2) “Function” 为寄存器功能码的选择

代码	中文名称	寄存器PLC地址	位操作/字操作	操作数量
01	读线圈状态	00001-09999	位操作	单个或多个
02	读离散输入状态	10001-19999	位操作	单个或多个
03	读保持寄存器	40001-49999	字操作	单个或多个
04	读输入寄存器	30001-39999	字操作	单个或多个
05	写单个线圈	00001-09999	位操作	单个
06	写单个保持寄存器	40001-49999	字操作	单个
15	写多个线圈	00001-09999	位操作	多个
16	写多个保持寄存器	40001-49999	字操作	多个

目前只做通过 Modbus 实现对寄存器的开关操作，所以下拉选择“06 Write Sxxx”
 (3) “Address” 为寄存器起始地址(默认从 1 开始)，同样按照对接文档进行输入
 (4) “Quantity” 操作数量，因为“06 Write xxx”为操作单个，所以值只能为 1，其他都会提示错误，输入完后点击“OK”

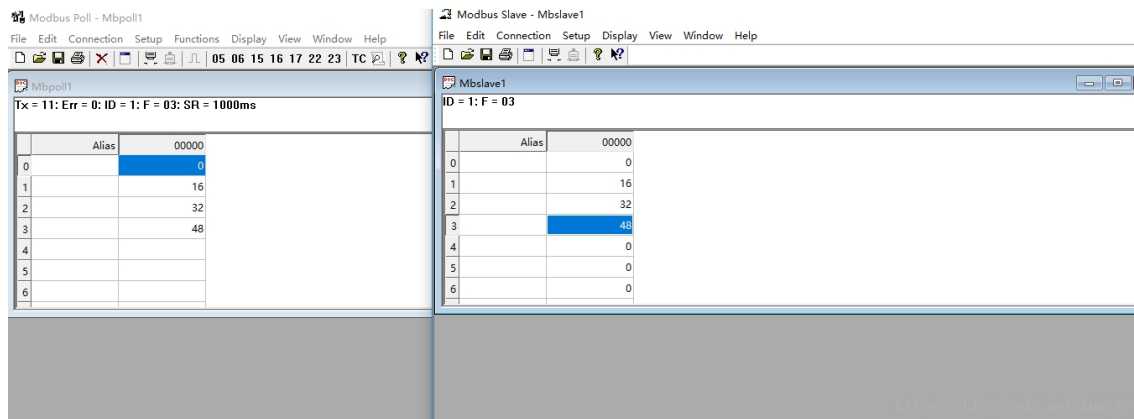


正确连接后的效果如上图，可以点击“00000”下方的“1”双击改 Value 为“0”实现设备的开和关，当然，具体问题要具体分析，这里只能作为示例进行参考

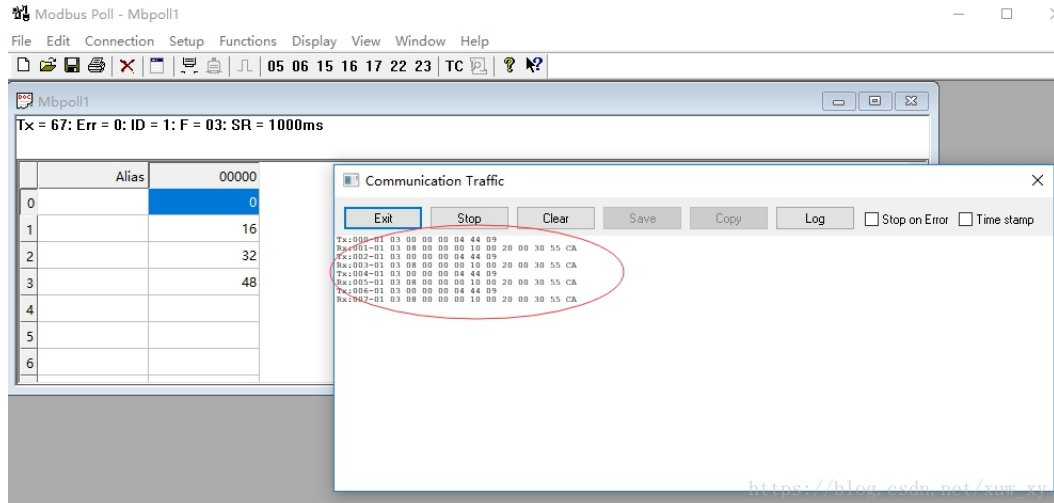


如果效果如上图，可能是从站地址、寄存器功能码或者寄存器起始地址输入错误，可对照文档核实从站地址、寄存器起始地址等，做到具体问题具体分析。

通讯开始，主机端显示：TX=11, Err=0, ID=1, F=03, SR=1000ms。意思是，发送 11 次命令，错误次数 0，从机 ID，功能号 03，轮询间隔 1000ms。



使用工具栏的“Communication Traffic”按钮，可以显示出当前发送命令和接受的数据，如下图：



再加一条数据解析：

```

TX发送数据
01: 从机地址
03: 功能码
00 00: 寄存器地址
00 06: 读取数据个数 6个
C5 C8: CRC校验
000066-Tx:01 03 00 00 00 06 C5 C8
000067-Rx:01 03 0C 00 0A 00 02 00 04 00 06 00 08 00 00 E7 CA
          10 2 4 6 8 0

RX接收数据
01: 从机地址
03: 功能码
0C: 返回字节数 (16进制数) 12字节
00 0A: 10
00 02: 2
00 04: 4
00 06: 6
00 08: 8
00 00: 0
E7 CA: CRC校验

```

<https://blog.csdn.net/u013184970>

ModbusSlave 是一个从站设备仿真软件，它用于接收主设备的命令包，并回送数据包；可用于测试和调试 Modbus 主站设备，便于观察 Modbus 通信过程中的各种报文。

Modbus Poll : Modbus 主机仿真器, 用于测试和调试 Modbus 从设备。该软件支持 ModbusRTU、ASCII、TCP/IP。用来帮助开发人员测试 Modbus 从设备，或者其它 Modbus 协议的测试和仿真。它支持多文档接口，即，可以同时监视多个从设备/数据域。每个窗口简单地设定从设备 ID，功能，地址，大小和轮询间隔。你可以从任意一个窗口读写寄存器和线圈。如果你想改变一个单独的寄存器，简单地双击这个值即可。或者你可以改变多个寄存器/线圈值。提供数据的多种格式方式，比如浮点、双精度、长整型（可以字节序列交换）。

Modbus Slave: Modbus从设备仿真器, 可以仿真32个从设备/地址域。每个接口都提供了对EXCEL

报表的 OLE 自动化支持。主要用来模拟 Modbus 从站设备,接收主站的命令包,回送数据包。帮助 Modbus 通讯设备开发人员进行 Modbus 通讯协议的模拟和测试,用于模拟、测试、调试 Modbus 通讯设备。可以 32 个窗口中模拟多达 32 个 Modbus 子设备。与 Modbus Poll 的用户界面相同,支持功能 01, 02, 03, 04, 05, 06, 15, 16, 22 和 23, 监视串口数据。