

# Modbus 配置手册

网测科技

2021/1/20

## 目录

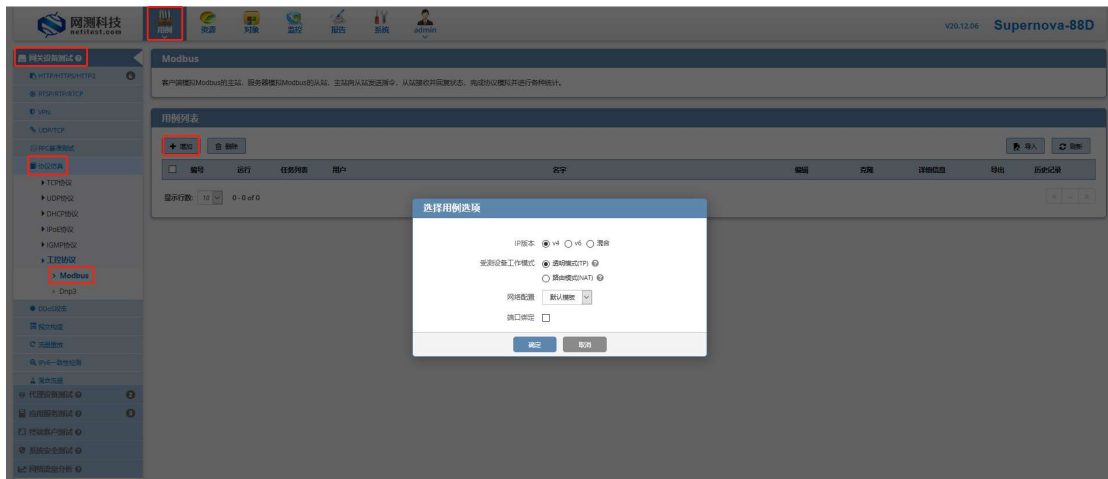
1. Modbus 用例说明.....	3
2. 创建 Modbus 测试用例.....	3
3. 运行 Modbus 测试用例.....	5
4. 详解客户端读或写的指令.....	7
4.1 状态码 01 读线圈.....	7
4.2 状态码 02 读离散输入（触点）状态.....	7
4.3 状态码 03 读保持寄存器数据.....	8
4.4 状态码 04 读输入寄存器数据.....	8
4.5 状态码 05 写单个线圈状态.....	8
4.6 状态码 06 写单个寄存器数据.....	9
4.7 状态码 15 写多个线圈状态.....	9
4.8 状态码 16 写多个寄存器数据.....	10
5. 服务器从站的配置.....	12
5.1 添加从站.....	12
5.2 下载 Pcap 报文验证.....	13

# 1. Modbus 用例说明

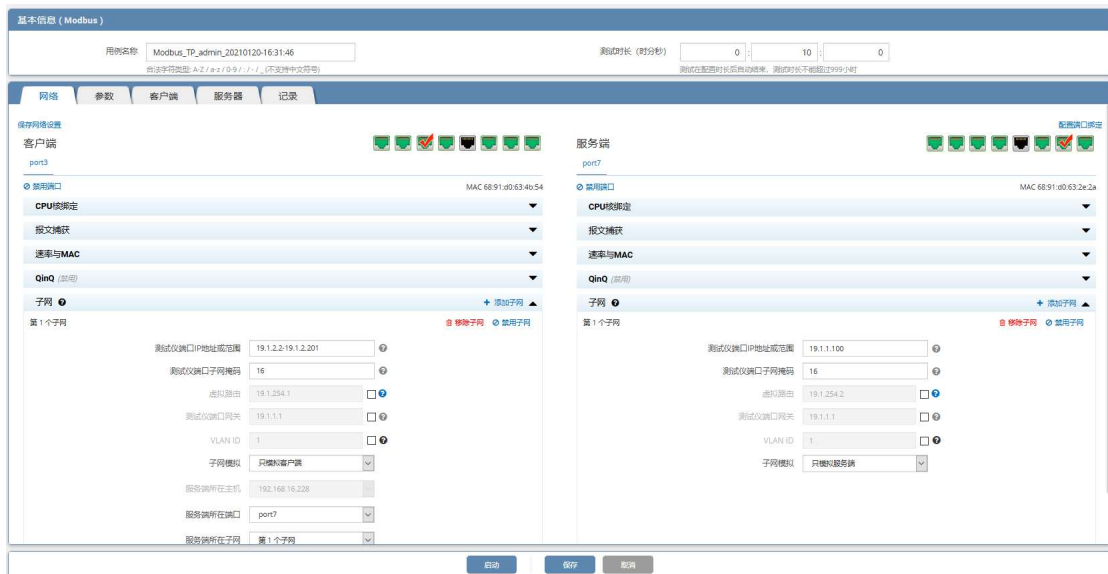
客户端模拟 Modbus 的主站，服务器模拟 Modbus 的从站，主站向从站发送指令，从站接收并回复状态，完成协议模拟并进行各种统计。

# 2. 创建 Modbus 测试用例

依次点击，用例 -> 网关设备测试 -> 协议仿真-> 工控协议 -> Modbus-> 增加。单击增加，在弹出的选择用例选项中，编辑用例网络选项，根据需要修改配置参数，然后点击确定，进入用例配置页面。



输入用例的名称，设置测试时长，选择测试端口，配置网络。



设置主要参数，Modbus 主要配置参数如下：

基本信息 (Modbus)

用户名: Modbus\_TP\_admin\_20210120-163146      测试时长 (时:分:秒): 0 : 10 : 0

网络 | 参数 | 客户端 | 服务器 | 记录

用列参数 | 通用参数

用户Admin占用内存: 60 GB  
用户Admin申请占用内存数量, 可以理解为->内存管理页面申请

用例运行占用内存: 60 GB  
运行此用例占用的内存空间, 包括DPDK的大页内存, 以及SSL会话占用的堆叠系统内存, 范围: 2-6GB

DPDK大页内存占比: 50 %  
DPDK大页内存占比指运行内存的百分比, 当中HTTP并发调用时, 建议大页内存占比50%, 剩余内存用于建立SSL会话, 范围: 30%-95%

从站数量: 247      客户端即为从站  
范围: 1-247, 每个客户端端口IDCPU序, 至少需要一个从站

Modbus模式: TCP模式      工作模式  
Modbus模式包括: TCP模式, 和UDP/TCP的模式

发送Modbus命令速率: 1      发送命令速率  
每秒发送多少个从站及DPDK Modbus命令数据, 0表示不限制, 范围0-100000

启动 | 保存 | 取消

基本信息 (Modbus)

用户名: Modbus\_TP\_admin\_20210120-163146      测试时长 (时:分:秒): 0 : 10 : 0

网络 | 参数 | 客户端 | 服务器 | 记录

选择端口范围: 10000-65535  
范围: 1,000-65,535

执行方法: 选择所有    取消所有

- 01读线圈状态
- 02读离散输入 (线圈) 状态
- 03读保持寄存器数据
- 04读输入寄存器数据
- 05写单个线圈状态
- 06写单个寄存器数据
- 15写多个线圈状态
- 16写多个寄存器数据

配置客户端发送或获取指令

启动 | 保存 | 取消

基本信息 (Modbus)

用户名: Modbus\_TP\_admin\_20210120-163146      测试时长 (时:分:秒): 0 : 10 : 0

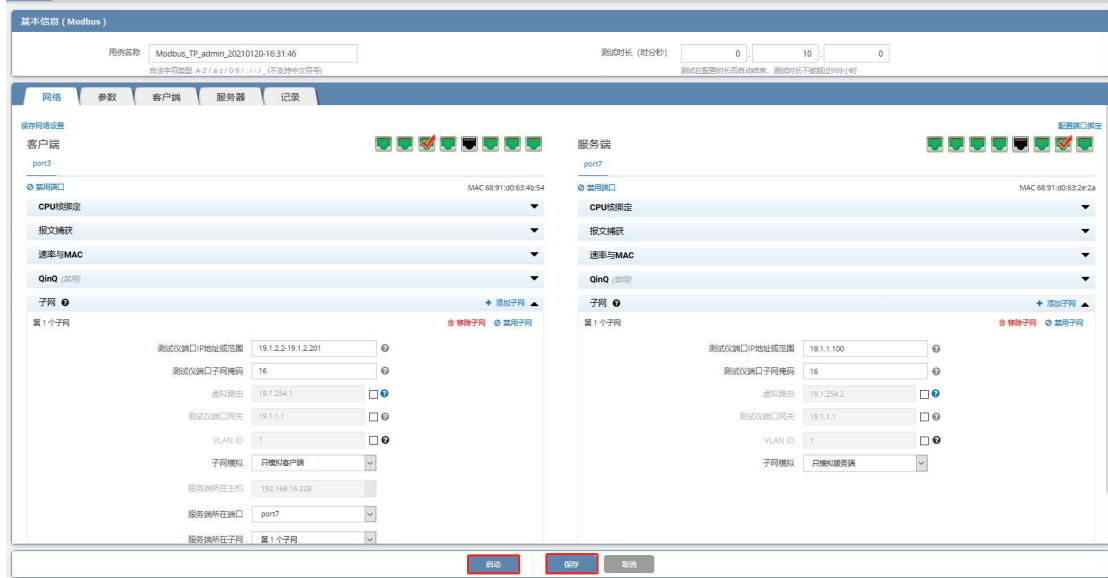
网络 | 参数 | 客户端 | 服务器 | 记录

从站对象类型: 默认Modbus地址

服务器端口: 502      从站读取指令编辑

启动 | 保存 | 取消

用例参数配置完成后，可以点击保存，保存编辑完成的用例配置，同样也可以点击启动，直接启动运行测试用例。

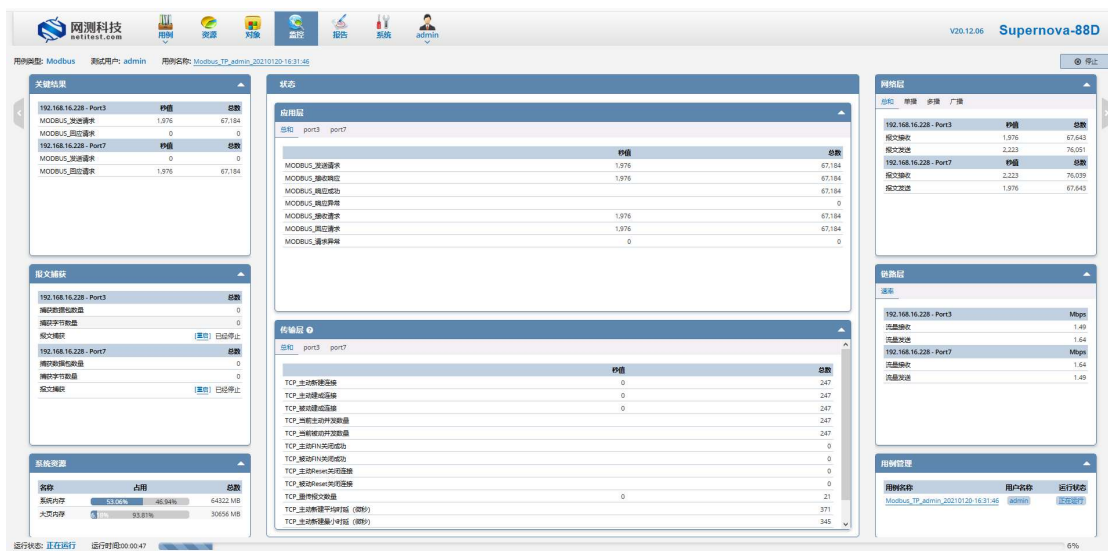


### 3. 运行 Modbus 测试用例

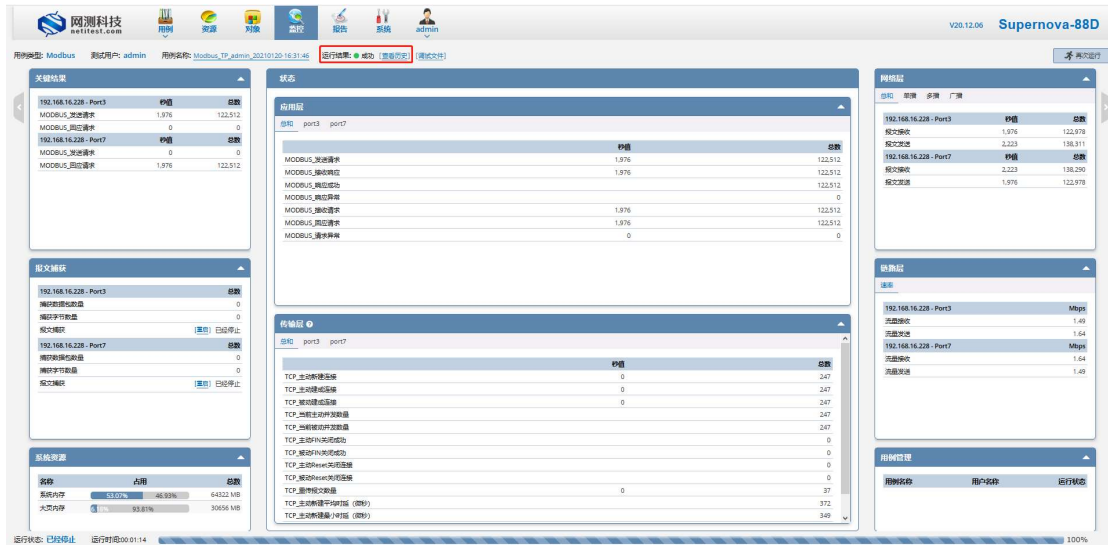
1. 点击 Modbus ， 选择编辑完成的测试用例， 点击启动按钮， 运行启动测试用例。



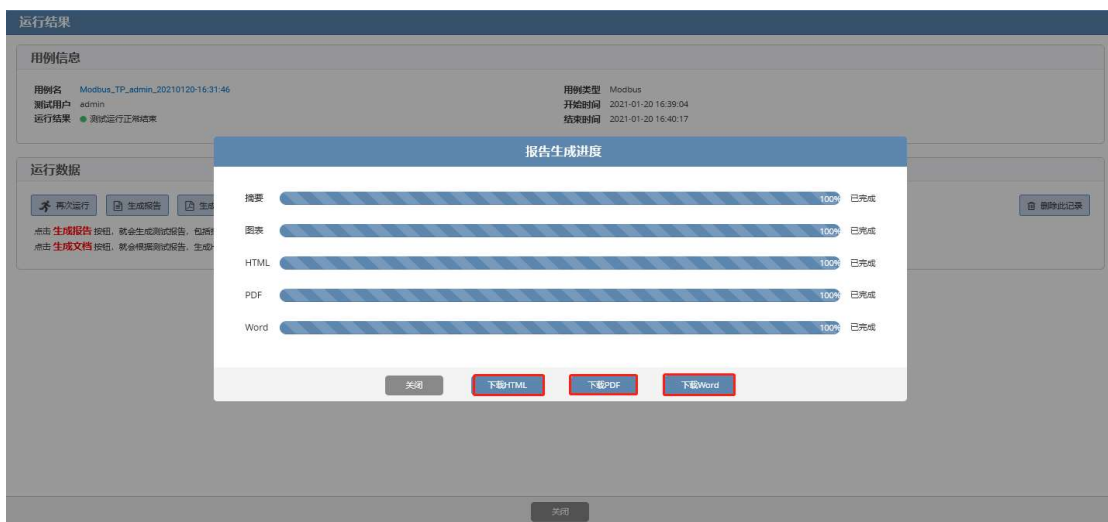
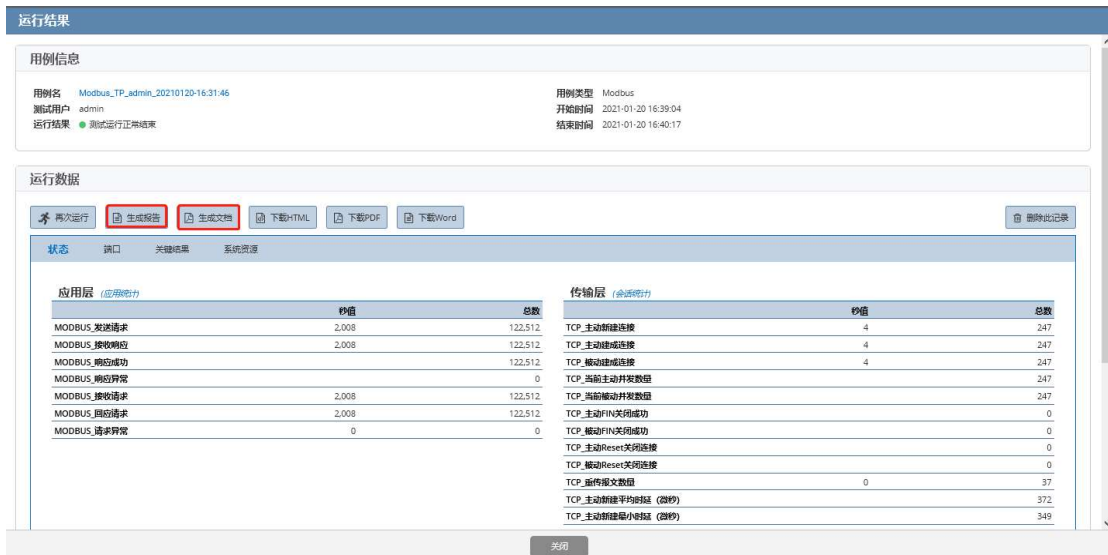
2. 用例启动后，可以在监控页面查看用例运行数据信息



3. 用例运行结束，点击查看历史，可以查看测试报告。



4. 生成报告数据及 HTML/PDF/Word 报告，报告生成后，可以下载 HTML/PDF/Word 格式测试报告



## 4. 详解客户端读或写的指令

执行方法 **选择所有** | **取消所有**

- 01读线圈状态 
- 02读离散输入 (触点) 状态 
- 03读保持寄存器数据 
- 04读输入寄存器数据 
- 05写单个线圈状态 
- 06写单个寄存器数据 
- 15写多个线圈状态 
- 16写多个寄存器数据 

01-06, 15, 16为状态码, 是modbus协议中最常用的八种指令

### 4.1 状态码 01 读线圈

#### 01读线圈状态

起始地址  **在什么位置开始读**

可用地址范围: 0 - 65,535

数量  **连续读多少**

范围: 1 - 2,000, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

确定

取消

### 4.2 状态码 02 读离散输入 (触点) 状态

#### 02读离散输入 (触点) 状态

起始地址  **在哪里开始读**

可用地址范围: 0 - 65,535

数量  **连续读多少**

范围: 1 - 2,000, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

确定

取消

### 4.3 状态码 03 读保持寄存器数据

#### 03读保持寄存器数据

起始地址  **在哪里开始读**  
可用地址范围: 0 - 65,535

数量  **连续读多少**  
范围: 1 - 125, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

### 4.4 状态码 04 读输入寄存器数据

#### 04读输入寄存器数据

起始地址  **在哪里开始读**  
可用地址范围: 0 - 65,535

数量  **连续读多少**  
范围: 1 - 125, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

### 4.5 状态码 05 写单个线圈状态

#### 05写单个线圈状态

起始地址  **在哪里开始写**  
可用地址范围: 0 - 65,535

下一步

取消

#### 05写单个线圈状态

起始地址: 0, 数量: 1

	00000
0	<input type="text" value="0"/> <input type="button" value="v"/>
1	0
2	1
3	
4	
5	
6	
7	
8	
9	

**写0或1**

在 wireshark 中 16 进制 0 显示为 0000, 1 显示为 ff00;



## 4.6 状态码 06 写单个寄存器数据

**06写单个寄存器数据**

起始地址:   
可用地址范围: 0 - 65,535

数值类型:  无符号的取值可以为0到65535  
 有符号的取值可以为-32768到32767

### 06写单个寄存器数据

起始地址: 0, 数量: 1, 数值类型: 16位无符号整型

	00000
0	1
1	
2	
3	
4	
5	
6	
7	
8	
9	

在写单个寄存器数据中，无符号整型 0 的 16 进制表示法为 0000，65535 表示为 ffff；有符号整型的 0 为 0000，1 为 0001，-1 为 ffff，-2 为 fffe；

## 4.7 状态码 15 写多个线圈状态

**15写多个线圈状态**

起始地址:   
可用地址范围: 0 - 65,535

数量:   
范围: 1 - 2,000, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

## 15 写多个线圈状态

起始地址: 0, 数量: 10

	00000
0	1
1	0
2	1
3	0
4	1
5	1
6	1
7	1
8	0
9	1

写多个线圈, 每个线圈写0或1

在写多个线圈中, 16 进制的表示方法是需要计算得到的, 如上图写 1010111101 这 10 个数字, 每 8 位是 1 个字节, 超出 8 位不足 16 位的在后面添加 0 到 16 位, 也就变成 1010111101000000, 由于计算机写入时会把每 8 位都拿出来从后往前写即第一个 8 位 11110101, 第二个 8 位 00000010, 最后转换成 16 进制即 f502;

## 4.8 状态码 16 写多个寄存器数据

**16 写多个寄存器数据**

起始地址:  可用地址范围: 0 - 65,535

数量:  范围: 1 - 123, 且不大于可用地址数量 (可用地址数量 = 65536 - 起始地址)

数值类型: 

- 16位无符号整型
- 16位有符号整型
- 32位无符号整型
- 32位有符号整型
- 64位无符号整型
- 64位有符号整型
- 32位浮点类型
- 64位双精度类型

16位32位64位分别代表在填入数据时占用几个小格子, 16占1个, 32占2个, 64占4个

寄存器数据

线圈状态

寄存器数据

起始地址: 0, 数量: 10, 数值类型: 32位有符号整型

	00000
0	1
1	--
2	1
3	--
4	1
5	--
6	1
7	--
8	1
9	--

起始地址: 0, 数量: 10, 数值类型: 64位无符号整型

	00000
0	1
1	--
2	--
3	--
4	1
5	--
6	--
7	--
8	
9	

16 位无符号所填值的范围 0 到 65535

32 位无符号所填值的范围 0 到 6294967295

64 位无符号所填值的范围 0 到 18446744073709553000

16 位有符号所填值的范围-32768 到 32767

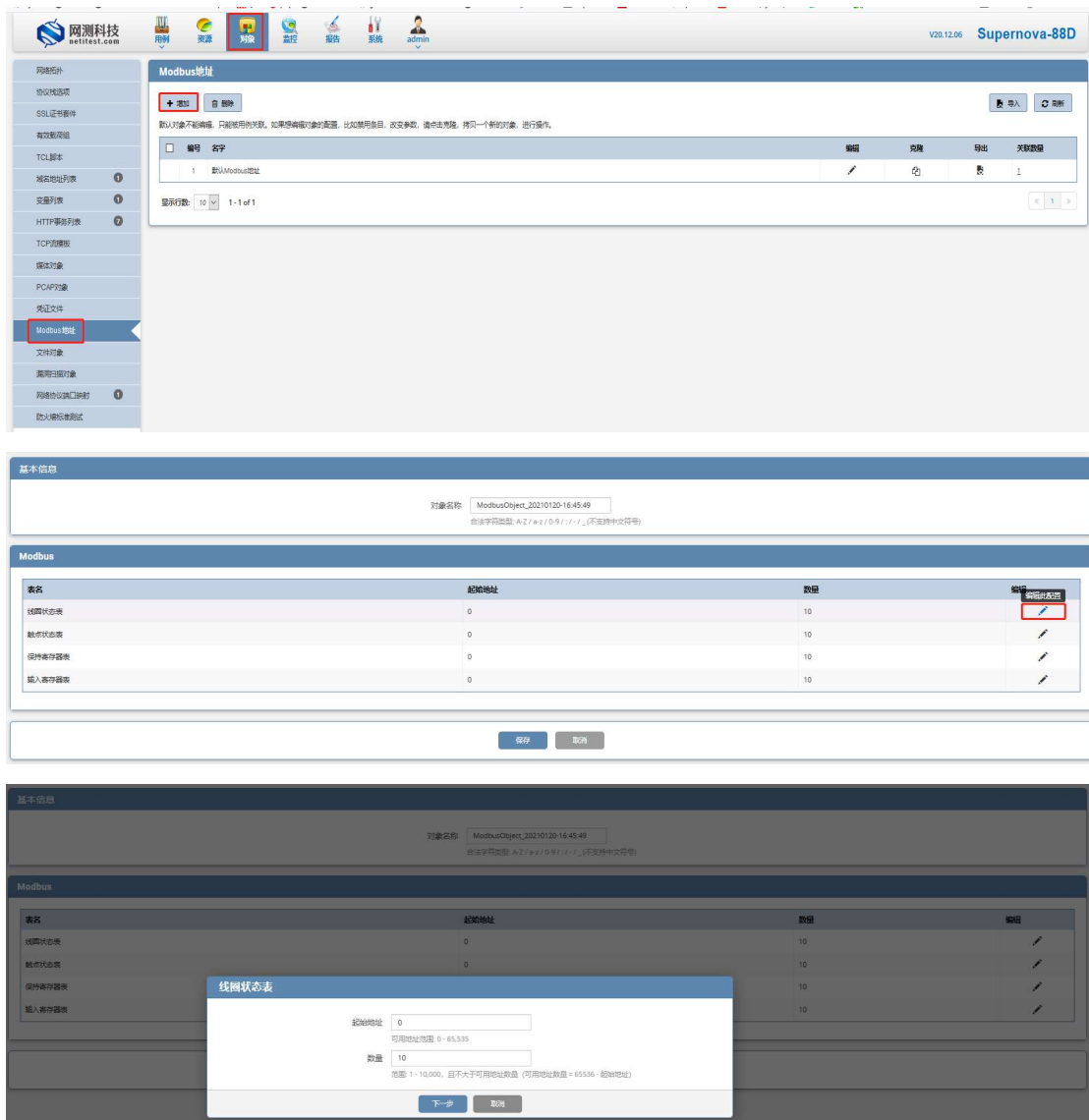
32 位有符号所填值的范围-2147483648 到 2147483647

64 位有符号所填值的范围-9223372036854776000 到 9223372036854776000

## 5. 服务器从站的配置

### 5.1 添加从站

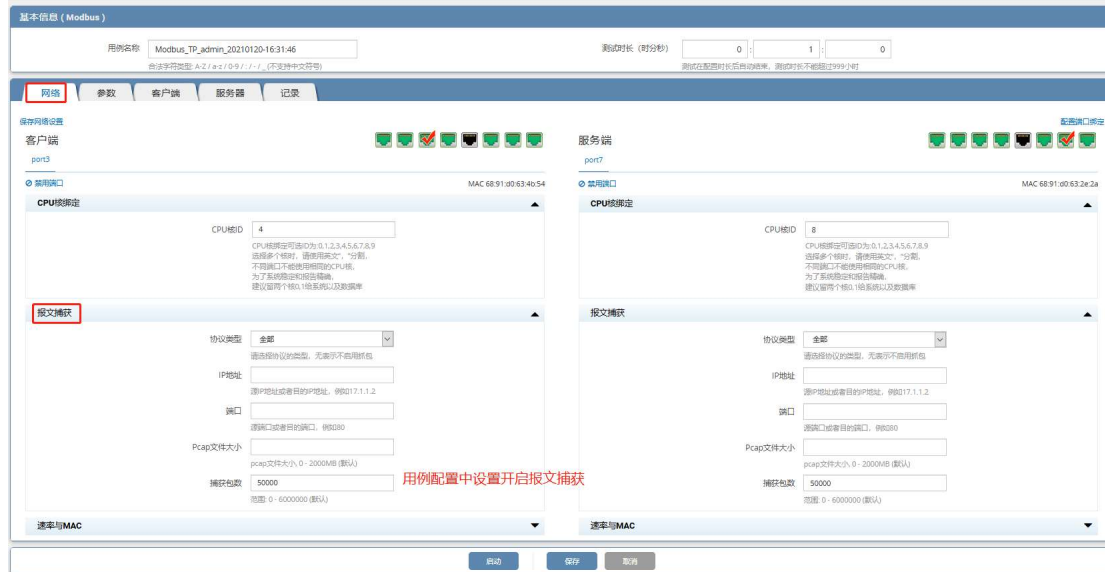
依次点击，对象 -> Modbus 地址 -> 增加。单击增加，在弹出的配置选项中，编辑 Modbus 配置选项，根据需要修改配置参数，然后点击保存，保存 Modbus 地址对象。服务器端从站的设置与功能和主站的设置与功能类似，主站向从站写入和获取指令。



The screenshot displays the Nettest software interface for configuring Modbus addresses. The top section shows the 'Modbus地址' (Modbus Address) configuration window, which includes a table with one entry: '1 数据Modbus地址'. Below this, the 'Modbus' configuration page is shown, featuring a table with columns for '表名' (Table Name), '起始地址' (Start Address), '数量' (Quantity), and '编辑' (Edit). The table lists four objects: '线圈状态表' (Coil Status Table), '触点状态表' (Contact Status Table), '保持寄存器表' (Retentive Register Table), and '输入寄存器表' (Input Register Table). A '保存' (Save) button is visible at the bottom of this page. The bottom section shows the '线网状态表' (Line Network Status Table) configuration dialog box, which includes fields for '起始地址' (Start Address) and '数量' (Quantity), along with a '下一步' (Next Step) button.

## 5.2 下载 Pcap 报文验证

用例配置中，设置开启报文捕获



The screenshot shows the 'Modbus' configuration page with two client configurations. The 'PCAP Capture' (报文捕获) section is highlighted with a red box. A red annotation '用例配置中设置开启报文捕获' (Set PCAP capture in the case configuration) points to the 'PCAP Capture' checkbox, which is checked. Other settings include protocol type (全部), IP address (17.1.1.2), port (80), PCAP file size (200MB), and capture rate (50000).

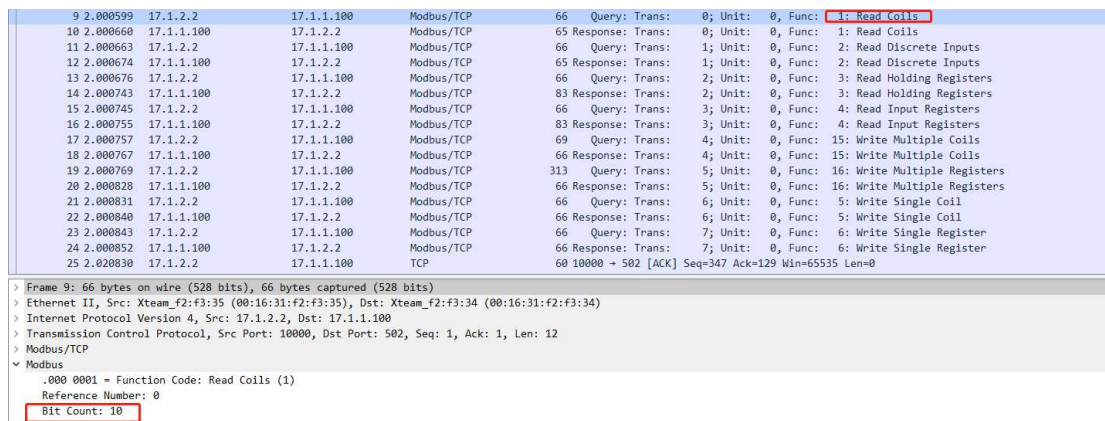
用例运行结束，下载 Pcap 报文



The screenshot shows the 'Supernova-88D' interface. A message states '目前没有正在运行的测试' (No tests are currently running). Below this, a list of test cases is shown with a '下载' (Download) button next to the selected case.

通过 Wireshark 打开，查看验证运行结果

Time	Source	Destination	Protocol	Length	Info	State
9.2.000599	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 0; Unit: 0; Func: 1; Read Coils	1; Read Coils
10.2.000660	17.1.1.100	17.1.1.2	Modbus/TCP	65	Response: Trans: 0; Unit: 0; Func: 1; Read Coils	1; Read Coils
11.2.000663	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 1; Unit: 0; Func: 2; Read Discrete Inputs	2; Read Discrete Inputs
12.2.000674	17.1.1.100	17.1.2.2	Modbus/TCP	65	Response: Trans: 1; Unit: 0; Func: 2; Read Discrete Inputs	2; Read Discrete Inputs
13.2.000676	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 2; Unit: 0; Func: 3; Read Holding Registers	3; Read Holding Registers
14.2.000743	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans: 2; Unit: 0; Func: 3; Read Holding Registers	3; Read Holding Registers
15.2.000745	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 3; Unit: 0; Func: 4; Read Input Registers	4; Read Input Registers
16.2.000755	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans: 3; Unit: 0; Func: 4; Read Input Registers	4; Read Input Registers
17.2.000757	17.1.1.2	17.1.1.100	Modbus/TCP	69	Query: Trans: 4; Unit: 0; Func: 15; Write Multiple Coils	15; Write Multiple Coils
18.2.000767	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans: 4; Unit: 0; Func: 15; Write Multiple Coils	15; Write Multiple Coils
19.2.000769	17.1.1.2	17.1.1.100	Modbus/TCP	313	Query: Trans: 5; Unit: 0; Func: 16; Write Multiple Registers	16; Write Multiple Registers
20.2.000828	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans: 5; Unit: 0; Func: 16; Write Multiple Registers	16; Write Multiple Registers
21.2.000831	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 6; Unit: 0; Func: 5; Write Single Coil	5; Write Single Coil
22.2.000840	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans: 6; Unit: 0; Func: 5; Write Single Coil	5; Write Single Coil
23.2.000843	17.1.1.2	17.1.1.100	Modbus/TCP	66	Query: Trans: 7; Unit: 0; Func: 6; Write Single Register	6; Write Single Register
24.2.000852	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans: 7; Unit: 0; Func: 6; Write Single Register	6; Write Single Register
25.2.020830	17.1.2.2	17.1.1.100	TCP	60	10000 → 502 [ACK] Seq=347 Ack=129 Win=65535 Len=0	



The screenshot shows a detailed view of a Modbus packet in Wireshark. The packet list pane shows a 'Read Coils' packet (ID 1) with a state of '1; Read Coils'. The packet details pane shows the 'Modbus' section with 'Function Code: Read Coils (1)' and 'Reference Number: 0'. The packet bytes pane shows the raw data and its hex representation.

主站向从站发起获取指令，在 0 的位置开始连续读 10 个

9	2.000599	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	0; Unit: 0, Func:	1: Read Coils
10	2.000660	17.1.1.100	17.1.2.2	Modbus/TCP	65	Response: Trans:	0; Unit: 0, Func:	1: Read Coils
11	2.000663	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	1; Unit: 0, Func:	2: Read Discrete Inputs
12	2.000674	17.1.1.100	17.1.2.2	Modbus/TCP	65	Response: Trans:	1; Unit: 0, Func:	2: Read Discrete Inputs
13	2.000676	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	2; Unit: 0, Func:	3: Read Holding Registers
14	2.000743	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	2; Unit: 0, Func:	3: Read Holding Registers
15	2.000745	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	3; Unit: 0, Func:	4: Read Input Registers
16	2.000755	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	3; Unit: 0, Func:	4: Read Input Registers
17	2.000757	17.1.2.2	17.1.1.100	Modbus/TCP	69	Query: Trans:	4; Unit: 0, Func:	15: Write Multiple Coils
18	2.000767	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	4; Unit: 0, Func:	15: Write Multiple Coils
19	2.000769	17.1.2.2	17.1.1.100	Modbus/TCP	313	Query: Trans:	5; Unit: 0, Func:	16: Write Multiple Registers
20	2.000828	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	5; Unit: 0, Func:	16: Write Multiple Registers
21	2.000831	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	6; Unit: 0, Func:	5: Write Single Coil
22	2.000840	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	6; Unit: 0, Func:	5: Write Single Coil
23	2.000843	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	7; Unit: 0, Func:	6: Write Single Register
24	2.000852	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	7; Unit: 0, Func:	6: Write Single Register
25	2.020830	17.1.2.2	17.1.1.100	TCP	60	10000 → 502 [ACK]	Seq=347 Ack=129 Win=65535 Len=0	

> Frame 10: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)  
 > Ethernet II, Src: Xteam\_f2:f3:34 (00:16:31:f2:f3:34), Dst: Xteam\_f2:f3:35 (00:16:31:f2:f3:35)  
 > Internet Protocol Version 4, Src: 17.1.1.100, Dst: 17.1.2.2  
 > Transmission Control Protocol, Src Port: 502, Dst Port: 10000, Seq: 1, Ack: 13, Len: 11  
 > Modbus/TCP  
 > Modbus  
 .000 0001 = Function Code: Read Coils (1)  
 [Request Frame: 9]  
 Byte Count: 2  
 Data: da03

从站回复指令，下图是从站相对应的设置

### 线圈状态表

起始地址：0，数量：10

	00000
0	0
1	1
2	0
3	1
4	1
5	0
6	1
7	1
8	1
9	1

11	2.000663	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	1; Unit: 0, Func:	2: Read Discrete Inputs
12	2.000674	17.1.1.100	17.1.2.2	Modbus/TCP	65	Response: Trans:	1; Unit: 0, Func:	2: Read Discrete Inputs
13	2.000676	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	2; Unit: 0, Func:	3: Read Holding Registers
14	2.000743	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	2; Unit: 0, Func:	3: Read Holding Registers
15	2.000745	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	3; Unit: 0, Func:	4: Read Input Registers
16	2.000755	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	3; Unit: 0, Func:	4: Read Input Registers
17	2.000757	17.1.2.2	17.1.1.100	Modbus/TCP	69	Query: Trans:	4; Unit: 0, Func:	15: Write Multiple Coils
18	2.000767	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	4; Unit: 0, Func:	15: Write Multiple Coils
19	2.000769	17.1.2.2	17.1.1.100	Modbus/TCP	313	Query: Trans:	5; Unit: 0, Func:	16: Write Multiple Registers
20	2.000828	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	5; Unit: 0, Func:	16: Write Multiple Registers
21	2.000831	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	6; Unit: 0, Func:	5: Write Single Coil
22	2.000840	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	6; Unit: 0, Func:	5: Write Single Coil
23	2.000843	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	7; Unit: 0, Func:	6: Write Single Register
24	2.000852	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	7; Unit: 0, Func:	6: Write Single Register
25	2.020830	17.1.2.2	17.1.1.100	TCP	60	10000 → 502 [ACK]	Seq=347 Ack=129 Win=65535 Len=0	

> Frame 12: 65 bytes on wire (520 bits), 65 bytes captured (520 bits)  
 > Ethernet II, Src: Xteam\_f2:f3:34 (00:16:31:f2:f3:34), Dst: Xteam\_f2:f3:35 (00:16:31:f2:f3:35)  
 > Internet Protocol Version 4, Src: 17.1.1.100, Dst: 17.1.2.2  
 > Transmission Control Protocol, Src Port: 502, Dst Port: 10000, Seq: 12, Ack: 25, Len: 11  
 > Modbus/TCP  
 > Modbus  
 .000 0010 = Function Code: Read Discrete Inputs (2)  
 [Request Frame: 11]  
 Byte Count: 2  
 Data: ed03

从站回复指令，下图是从站相对应的设置

### 触点状态表

起始地址：0，数量：10

	00000
0	1
1	0
2	1
3	1
4	0
5	1
6	1
7	1
8	1
9	1

13	2.001359	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	2;	Unit: 0,	Func: 3: Read Holding Registers
14	2.001369	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	2;	Unit: 0,	Func: 3: Read Holding Registers
15	2.001371	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	3;	Unit: 0,	Func: 4: Read Input Registers
16	2.001428	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	3;	Unit: 0,	Func: 4: Read Input Registers
17	2.001431	17.1.2.2	17.1.1.100	Modbus/TCP	69	Query: Trans:	4;	Unit: 0,	Func: 15: Write Multiple Coils
18	2.001440	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	4;	Unit: 0,	Func: 15: Write Multiple Coils
19	2.001443	17.1.2.2	17.1.1.100	Modbus/TCP	313	Query: Trans:	5;	Unit: 0,	Func: 16: Write Multiple Registers
20	2.001454	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	5;	Unit: 0,	Func: 16: Write Multiple Registers
21	2.001456	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	6;	Unit: 0,	Func: 5: Write Single Coil
22	2.001513	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	6;	Unit: 0,	Func: 5: Write Single Coil
23	2.001516	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	7;	Unit: 0,	Func: 6: Write Single Register
24	2.001525	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	7;	Unit: 0,	Func: 6: Write Single Register
25	2.021481	17.1.2.2	17.1.1.100	TCP	60	10000 -> 502 [ACK]	Seq=347	Ack=129	Win=65535 Len=0

[Request Frame: 13]

Byte Count: 20

Register 0 (UINT16): 1238

Register 1 (UINT16): 12

Register 2 (UINT16): 6295

Register 3 (UINT16): 12

Register 4 (UINT16): 144

Register 5 (UINT16): 0

Register 6 (UINT16): 0

Register 7 (UINT16): 0

Register 8 (UINT16): 0

Register 9 (UINT16): 0

由于是32位所以会有多一位的占位

从站回复指令，下图是从站相对应的设置

### 保持寄存器表

起始地址：0，数量：10，数值类型：32位无符号整型

	00000
0	1238
1	--
2	12
3	--
4	6295
5	--
6	12
7	--
8	144
9	--

15	2.001371	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	3;	Unit: 0,	Func: 4: Read Input Registers
16	2.001428	17.1.1.100	17.1.2.2	Modbus/TCP	83	Response: Trans:	3;	Unit: 0,	Func: 4: Read Input Registers
17	2.001431	17.1.2.2	17.1.1.100	Modbus/TCP	69	Query: Trans:	4;	Unit: 0,	Func: 15: Write Multiple Coils
18	2.001440	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	4;	Unit: 0,	Func: 15: Write Multiple Coils
19	2.001443	17.1.2.2	17.1.1.100	Modbus/TCP	313	Query: Trans:	5;	Unit: 0,	Func: 16: Write Multiple Registers
20	2.001454	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	5;	Unit: 0,	Func: 16: Write Multiple Registers
21	2.001456	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	6;	Unit: 0,	Func: 5: Write Single Coil
22	2.001513	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	6;	Unit: 0,	Func: 5: Write Single Coil
23	2.001516	17.1.2.2	17.1.1.100	Modbus/TCP	66	Query: Trans:	7;	Unit: 0,	Func: 6: Write Single Register
24	2.001525	17.1.1.100	17.1.2.2	Modbus/TCP	66	Response: Trans:	7;	Unit: 0,	Func: 6: Write Single Register
25	2.021481	17.1.2.2	17.1.1.100	TCP	60	10000 -> 502 [ACK]	Seq=347	Ack=129	Win=65535 Len=0

[Request Frame: 15]

Byte Count: 20

Register 0 (UINT16): 65535

Register 1 (UINT16): 0

Register 2 (UINT16): 999

Register 3 (UINT16): 123

Register 4 (UINT16): 1

Register 5 (UINT16): 227

Register 6 (UINT16): 1

Register 7 (UINT16): 1

Register 8 (UINT16): 1

Register 9 (UINT16): 1