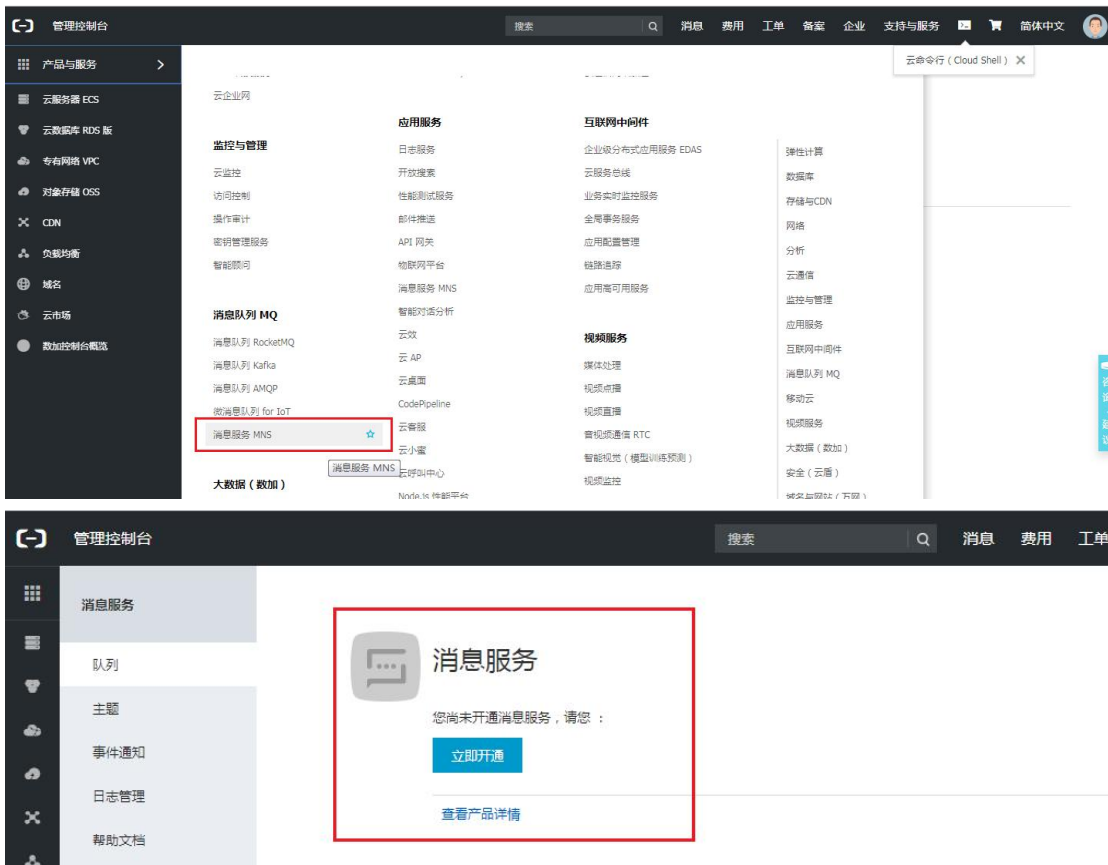


MQT-805 连接阿里云 IoT 向导

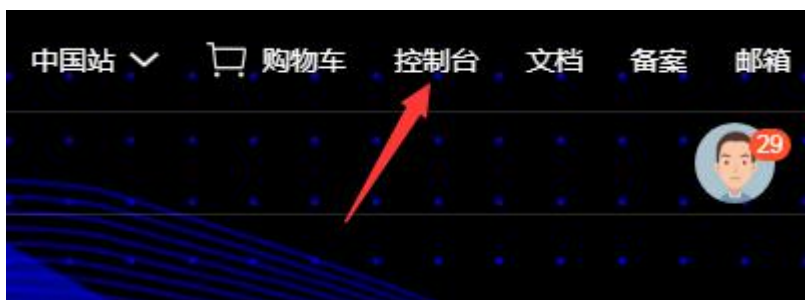
1、登陆阿里云平台（可以用淘宝账号直接登陆），链接如下：

https://account.aliyun.com/login/login.htm?oauth_callback=https%3A%2F%2Fiot.console.aliyun.com%2Fproduct

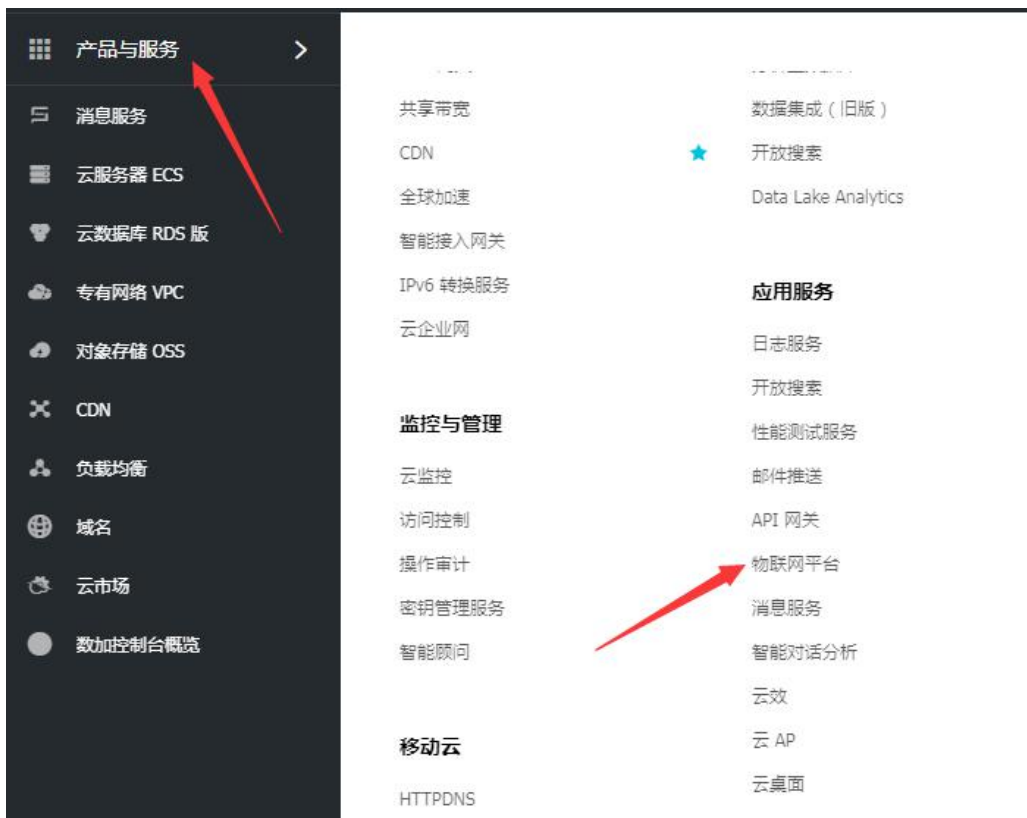
2、开通“消息服务 MNS”：



3、从“控制台”进入：



4、从左上角的“产品与服务”中找到“物联网平台”，进入物联网平台：



5、创建产品，依次点击：设备管理-产品-创建产品-基础版

The image shows a web form titled '新建产品 / 第二步：填写产品信息 (共二步)'. The form contains the following fields and options:

- * 产品名称：** A text input field with the placeholder text '请输入您的产品名称'.
- * 节点类型：** Radio buttons for '设备' (selected), '网关', and '其他'.
- * 使用 ID² 认证：** Radio buttons for '是' and '否' (selected).
- 产品描述：** A text area with the placeholder text '请输入产品描述' and a character count '0/100'.

At the bottom of the form, there is a link '使用文档' and two buttons: '上一步' and '完成'.

其中，“节点类型”选择“设备”。

6、添加设备，依次点击：设备管理-设备-设备列表-添加设备

添加设备

特别说明：deviceName可以为空,当为空时,阿里云会颁发全局唯一标识符作为deviceName。

* 产品：
mqt805

DeviceName：
支持英文字母、数字和特殊字符_@.，长度限制4~32

确认 取消

添加完显示如下：（设备未连接显示为“未激活”）

设备列表

DeviceName	设备所属产品	节点类型	状态/启用状态	最后上线时间	操作
12345678	mqt805	设备	未激活 <input type="checkbox"/>	-	查看 删除

点击“设备列表”中的对应设备，可以看到以下设备信息，其中 ProductKey、DeviceName、DeviceSecret 需要保存，且需要在配置软件 MQT-123 中进行相应设置（在配置软件的“无线互联网-4G”中）

设备信息

产品名称	mqt805	ProductKey	a19oPEbcmAl 复制	区域	华东2（上海）
节点类型	设备	DeviceName	12345678 复制	DeviceSecret	IEA55n8Hujtl... 复制 隐藏
当前状态	未激活	IP地址	-	固件版本	-
添加时间	2019/02/11 14:45:36	激活时间		最后上线时间	
实时延迟	测试				

在设备的“Topic 列表”中可以看到以下信息：（该列表中的 Topic 是这台设备可以发布或者订阅的 Topic）

设备信息 **Topic列表** 设备影子 日志服务

设备的Topic列表 ●

设备的Topic	设备具有的权限	发布消息数	操作
/a19oPEbcmAI/12345678/update	发布	0	发布消息
/a19oPEbcmAI/12345678/update/error	发布	0	发布消息
/a19oPEbcmAI/12345678/get	订阅	0	发布消息

7、添加/删除/修改 “Topic”

依次点击设备管理-产品-查看-Topic 类列表-定义/编辑/删除 Topic。

在配置软件 MQT-123 中设置的 Topic（Topic 名称）需和这里设置的 Topic 一致（也可以选择其中的一部分，未必需要全部用上）

产品信息 **Topic类列表** 服务端订阅 日志服务

产品Topic类列表 ● 定义Topic类

Topic类	操作权限	描述	操作
/a19oPEbcmAI/{deviceName}/update/error	发布		编辑 删除
/a19oPEbcmAI/{deviceName}/update/error	发布		编辑 删除
/a19oPEbcmAI/{deviceName}/get	订阅		编辑 删除

8、设置 “服务端订阅”

产品信息 Topic类列表 功能定义 **服务端订阅** 日志服务 在线调试

服务端订阅 ● 使用教程 设置

您还没有选择推送给服务端订阅的消息类型 [立即设置](#)

服务端订阅 (推送MNS) ● 设置

您还没有选择推送给服务端订阅的消息类型 [立即设置](#)

配置服务端订阅

选择推送的消息类型：

设备上报消息 ● 设备状态变化通知

设备生命周期变更 ●

保存 取消



9、将配置软件 MQT-123 中的参数按照配置软件内的指示以及自己的需求，配置完成，即可将 MQT-805 连接至阿里云 IoT 平台。

10、进入“消费服务 MNS”，找到创建的服务端订阅，设备连接后可以通过“发送消息”、“接收消息”来进行数据通信。



11、同文件夹中的《阿里云 IoT 测试 Demo》，是从阿里云 IoT 通过 SDK 调用接收到的数据的例程，供参考设计。

附：数据通讯格式

数据通讯格式示例如下：（其中，红字为注释）

➤ 发布（MQT-805 发送至阿里云 IoT，JSON 格式）

① 属性名称模式：

GPS:

```
{
  "version": "1.1.1.0",
  "GPS": {
    "Latitude": 31.057020, //纬度
    "Longitude": 121.189590 //经度
  }
}
```

DI:

```
{
  "version": "1.1.1.0",
  "DI":{
    "DI1": false,
    "DI2": true
  }
}
```

Modbus 主站/ Modbus 从站/Modbus TCP 从站:

读线圈状态/读输入状态/线圈状态:

```
{
  "version": "1.1.1.0", //版本，以便不同版本的格式数据管理
  "time": "19/03/07,17:04:07", //时间戳
  "properties": {
    "BI3": false, //属性名称模式发送主站或从站数据时，若
    "BI4": true, 一次性要传输超过 8 个属性的数据时，那么
    "BI5": false, MQT-805 网关会将数据每 8 个数据为一包，
  }
}
```

```
"BI6": false    分多包发送，直至将所有数据发送完毕。
}
}
```

读保持寄存器/读输入寄存器/保持寄存器:

```
{
  "version": "1.1.1.0",
  "time": "19/03/07,17:10:03",
  "properties": {
    "AI2": 999999.500000, //数据格式与配置软件 MQT-123 中的数据类型一致
    "AI3": 888888.500000,
    "AI4": 777777.500000,
    "AI5": 666666.500000
  }
}
```

透明传输: 使用专有的 Topic 发送，不能与其它数据的 Topic 共用

② 寄存器模式:

GPS:

```
{
  "version": "1.1.1.0",
  "type": "GPS",
  "Latitude": 31.057243,
  "Longitude": 121.188759
}
```

DI:

同时配了两路 DI:...

```
{
  "version": "1.1.1.0",
  "type": "DI1,DI2", //type 是数据的类型，这里为 DI
  "npoint": 2,      //npoint 是 data 中数据的个数
  "data": "00,ff"
```

```

}

只配置了一路 DI:

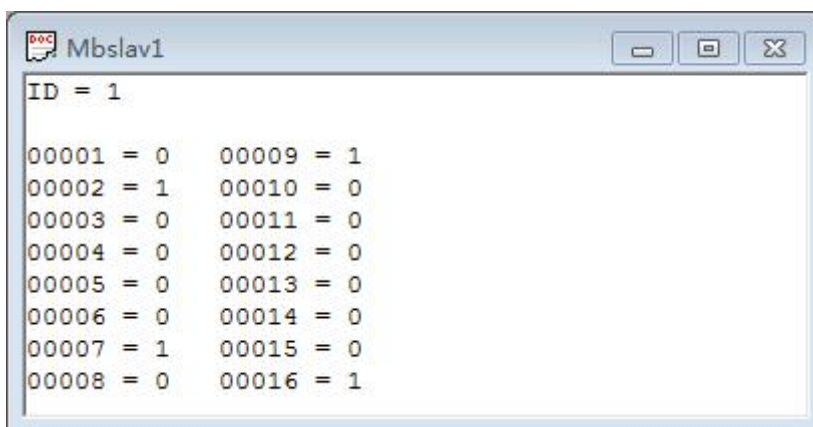
{
  "version": "1.1.1.0",
  "type": "DI1",
  "npoint": 1,
  "data": "ff"
}

```

Modbus 主站:

读线圈状态/读输入状态:...

实例（读线圈状态）:



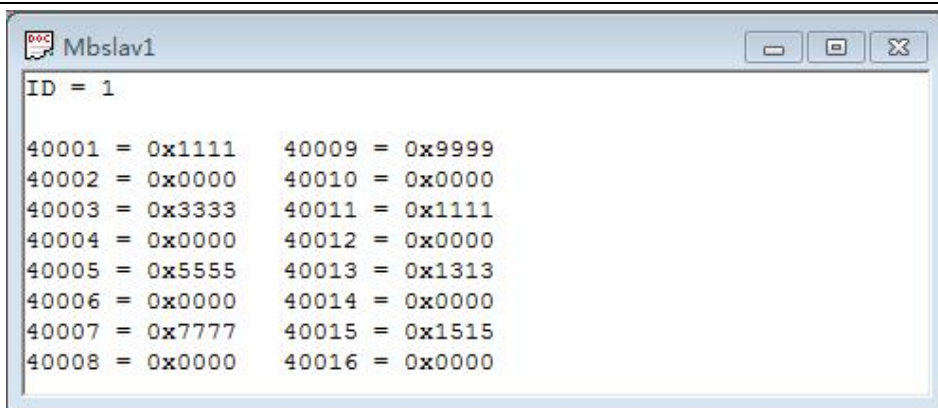
```

{
  "version": "1.1.1.0",
  "type": "MODBUS_MASTER",
  "slave id": 1, //slave id 是从站的地址
  "fc": 1, //fc 是 Modbus 的功能码
  "addr": 0, //addr 是 Modbus 寄存器起始地址，从 0 开始
  "npoint": 2, //npoint 是 data 中数据的个数
  "data": "42,81", //data 中的数据与 Modbus 协议一致
  "time": "19/02/27,16:05:14"//时间戳，如不配置则没有
}

```

读输入寄存器/读保持寄存器:...

实例（读保持寄存器）:



```

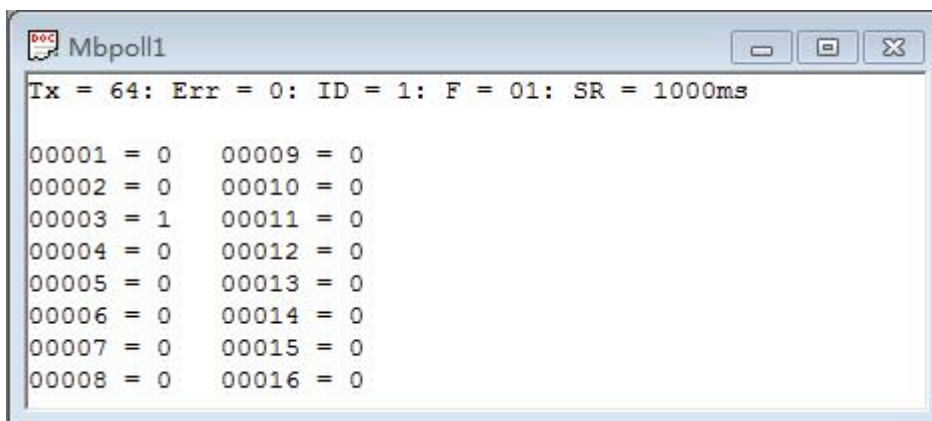
{
    "version": "1.1.1.0",
    "type": "MODBUS_MASTER",
    "slave id": 1,
    "fc": 3,
    "addr": 0,
    "npoint": 32, //寄存器模式最大单次发送数据个数为 100 个，大于 100 个则分包
    发送
    "data":
    "11,11,00,00,33,33,00,00,55,55,00,00,77,77,00,00,99,99,00,00,11,11,00,00,13,13,00,
    00,15,15,00,00"
}

```

Modbus 从站/Modbus TCP 从站:

线圈状态:...

只改变一个线圈的值:



```

{

```

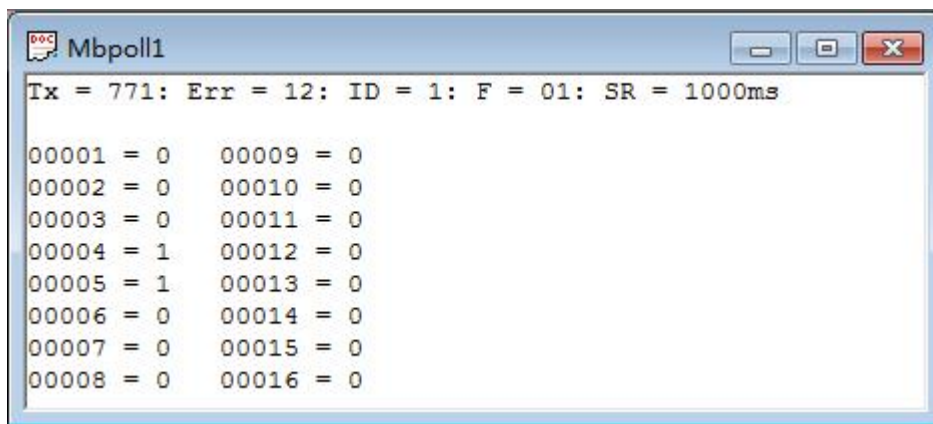
```

"version": "1.1.1.0",
"type": "MODBUS_TCP_SLAVE",
"block": 0, //block 代表区
" addr ": 2, //addr 是 Modbus 寄存器起始地址，从 0 开始
"npoint": 1, //npoint 为 1 时，说明只改变了一个寄存器，
"data": "FF", 那么 data 中为 00 或 FF，分别对应 0 或 1
"time": "19/02/27,16:13:40"
}

```

改变连续多个线圈的值：

实例 1：

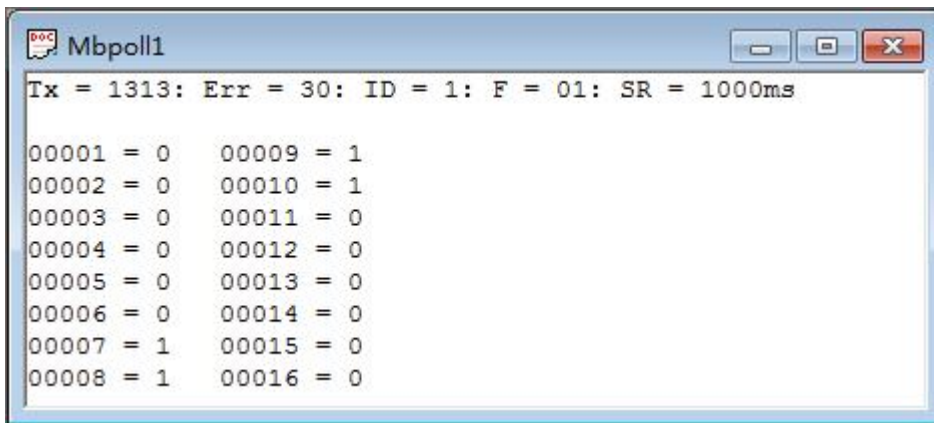


```

{
"version": "1.1.1.0",
"type": "MODBUS_TCP_SLAVE",
"block": 1,
" addr ": 3, //addr 是 Modbus 寄存器起始地址，从 0 开始
"npoint": 2, //npoint 不为 1，说明连续改变多个线圈的值，
"data": "18", 此例子需要去读取 data 中的第 1 个字节的
"time": "19/02/27,16:23:35" 第 4 位和第 5 位
}

```

实例 2：



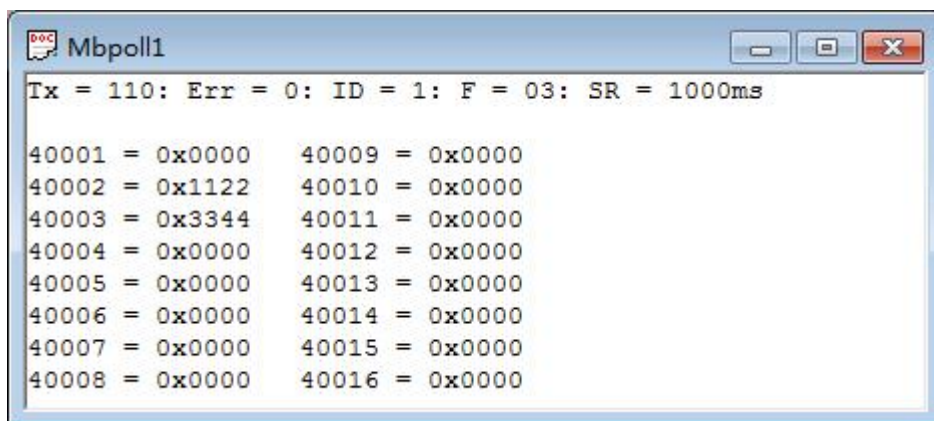
```

{
    "version": "1.1.1.0",
    "type": "MODBUS_TCP_SLAVE",
    "block": 0,
    "addr": 6, //addr 是 Modbus 寄存器起始地址, 从 0 开始
    "npoint": 4, //npoint 不为 1, 说明连续改变多个线圈的值,
    "data": "c0,03", //此例子需要去读取 data 中第 1 个字节的第
    "time": "19/02/27,16:35:01" // 1 位和第 2 位以及第 2 个字节的
} // 第 7 位和第 8 位

```

保持寄存器:..

实例:



```

{
    "version": "1.1.1.0",
    "type": "MODBUS_TCP_SLAVE",
    "block": 4,
}

```

```
" addr ": 1,  
"npoint": 4,  
"data": "11,22,33,44",  
"time": "19/02/27,16:48:39"  
}
```

➤ 订阅（阿里云 IoT 发送至网关，数据均使用 JSON 格式）

① 属性名称模式：（一次最多只能改变 8 个属性）

DO:

```
{  
  "DO": {  
    "DO1": false,  
    "DO2": true  
  }  
}
```

Modbus 主站/Modbus 从站/Modbus TCP 从站:

写线圈/输入状态:...

```
{  
  "properties": {  
    "BO3": false,  
    "BO4": true,  
    "BO5": false  
  }  
}
```

预置寄存器/输入寄存器:...

```
{  
  "properties": {  
    "AO3": 123.5,  
    "AO4": 321.5,  
    "AO5": 555.6  
  }  
}
```

}

② 寄存器模式:

DO:

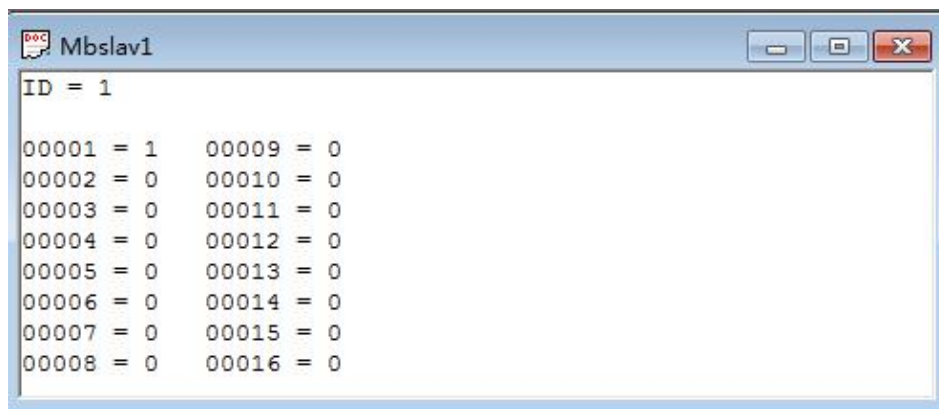
```
{
  "type": "DO1",
  "npoint": 1,
  "data": "00"
}
```

Modbus 主站:

写线圈:..

写单个线圈:

```
{
  "type": "MODBUS_MASTER",
  "slave id": 1,
  "fc": 5, //fc 是 Modbus 的功能码
  "addr": 0,
  "npoint": 1, //npoint 为 1 时, 说明只改变了一个寄存器,
  "data": "FF" //那么 data 中为 00 或 FF, 分别对应 0 或 1
}
```



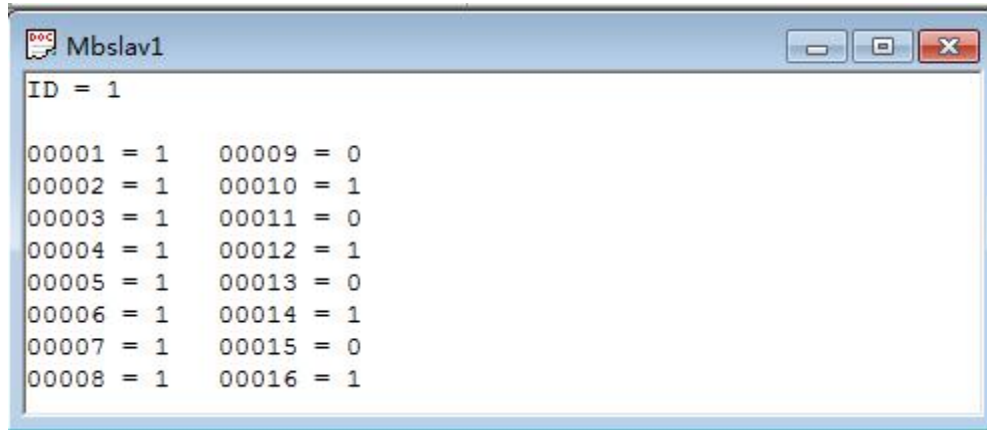
写多个线圈:

```
{
  "type": "MODBUS_MASTER",
```

```

"slave id": 1,
"fc": 15, //fc 是 Modbus 的功能码
" addr ": 0,
"npoint": 2, //npoint 为 2 时，与发布时的位对应关系一致
"data": "ff,aa"
}

```



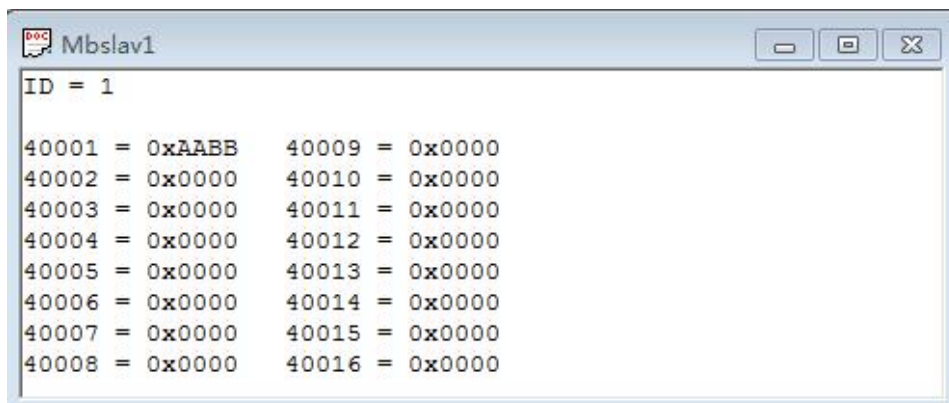
预置寄存器:..

预置单个寄存器:

```

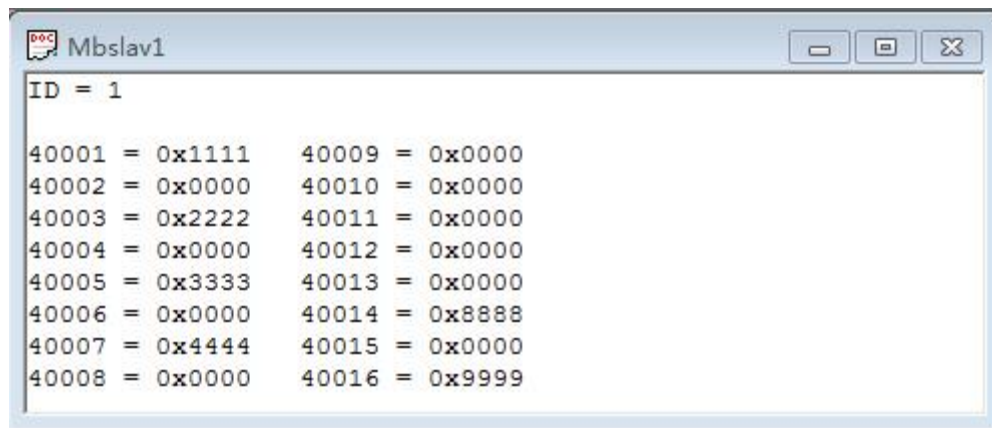
{
  "type": "MODBUS_MASTER",
  "slave id": 1,
  "fc": 6,
  " addr ": 0,
  "npoint": 2,
  "data": "aa,bb"
}

```



预置多个寄存器:

```
{
  "type": "MODBUS_MASTER",
  "slave id": 1,
  "fc": 16,
  "addr": 0,
  "npoint": 32,
  "data":
    "11,11,00,00,22,22,00,00,33,33,00,00,44,44,00,00,00,00,00,00,00,00,00,00,88,
    88,00,00,99,99"
}
```

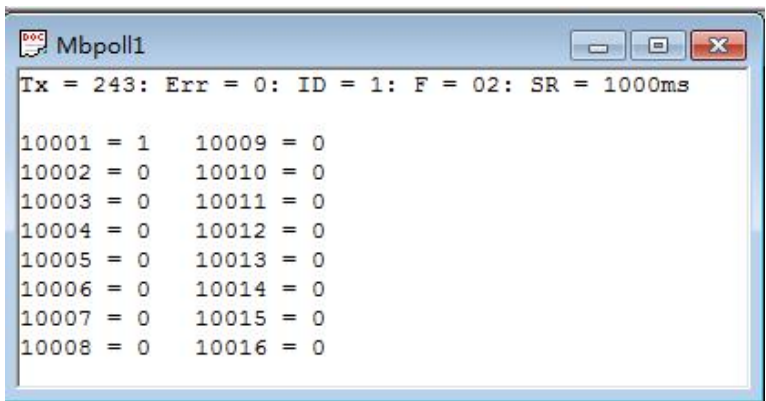


Modbus 从站/Modbus TCP 从站:

输入状态:...

改变单个输入状态位:

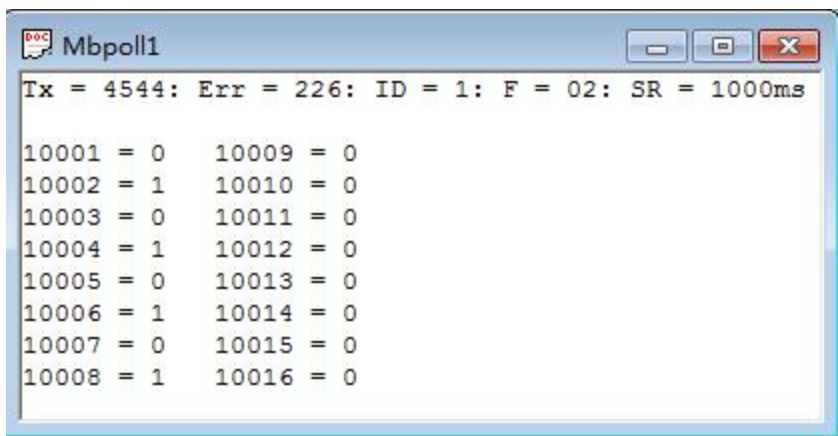
```
{
  "type": "MODBUS_SLAVE",
  "block": 1, //block 代表区
  "addr": 0,
  "npoint": 1,
  "data": "FF"
}
```



改变多个输入状态位:

```

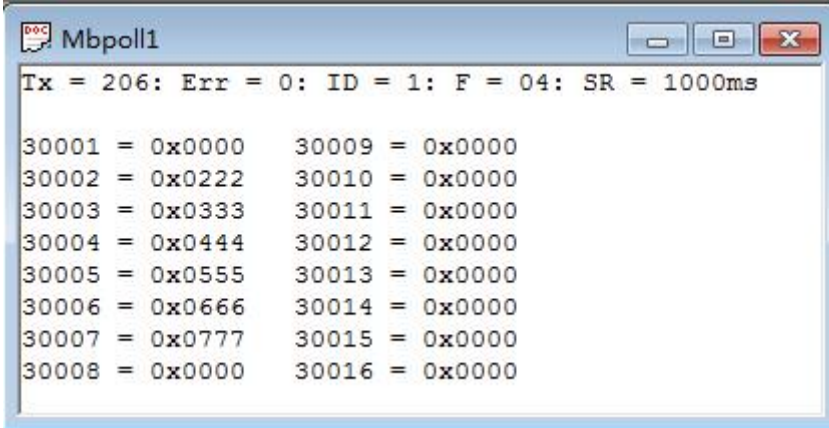
{
    "type": "MODBUS_SLAVE",
    "block": 1,
    "addr": 0,
    "npoint": 8, //npoint 不为 0, 说明改变多个, 这个例子是
    "data": "aa" 改变了 8 位
}
    
```



输入寄存器:..

```

{
    "type": "MODBUS_SLAVE",
    "block": 3,
    "addr": 1,
    "npoint": 12,
    "data": "02,22,03,33,04,44,05,55,06,66,07,77"
}
    
```

```
Tx = 206: Err = 0: ID = 1: F = 04: SR = 1000ms  
  
30001 = 0x0000    30009 = 0x0000  
30002 = 0x0222    30010 = 0x0000  
30003 = 0x0333    30011 = 0x0000  
30004 = 0x0444    30012 = 0x0000  
30005 = 0x0555    30013 = 0x0000  
30006 = 0x0666    30014 = 0x0000  
30007 = 0x0777    30015 = 0x0000  
30008 = 0x0000    30016 = 0x0000
```