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":

(-)	言理控制台		18.5	te Q 消息	费用 工	单 备案 企业	支持与服务	: DE 14	简体中文	0
III 7*8	品与服务 >						云命令	行(Cloud Shell)	×	
11 云瓶	膨脹 ECS	云企业网					- 1			
₩ 云8	数据库 RDS 版		应用服务	互联网中间件			- 1			
a. +#	词 dist VPC	监控与管理	日志服务	企业级分布式应用服务 EDAS		弹性计算	- 1			
		云监控	开放搜索	云服务总线		数据库				
• 对象	教存儲 OSS	访问控制	性能测试服务	业务实时监控服务		存储与CDN	- 1			
× CDM	N	操作审计	邮件推送	全局事务服务		网络	- 1			
A (05	助衛	密钥管理服务	API 例天	应用配置管理		分析	- 1			
(∰) ±\$2	ŝ	暂能队引	初联网千哲 注意服务 MMC	1993年1月17日 1995年1月11日 1995 1995年11 1995 1995 1995 1995 1995 1		云通信	- 1			
		法内别利 100	智能対抗分析	7277914 - J 727829		监控与管理	- 1			
	D-400	HERAYIMQ	云效	海缅服务		应用服务	- 1			
200	在控制台構造	清息队列 KOCKETMQ	Z AP	MERCHAN ZE		互联网中间件	- 1			
			云桌面	· 编辑:2019年		消息队列 MQ	- 1			咨
		微满思队列 for IoT	CodePipeline	视频直播		移动云	- 1			询
		消息服务 MNS	☆云春服	會视频通信 RTC		视频服务	- 1			建
		246	云小蜜	智能视觉(模型训练预测)		大数据(数加)				
		大数据(数加)	eage MINS 长呼叫中心	视频监控		安全(云盾)				
			Node.1s 体翻半台			1 域文与図社(万図)				
(-)	管理控制台				搜索		Q	消息	费用	工単
100 100	14									
	消息服务			1						
	队列		1	5						
	主题		您尚未开通消息朋	B务,请您:						
-										
~	争作进入		立即开通							
	日志管理									
×			查看产品详情							
	帮助文档									
•										

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

* 产品名称:	
请输入您的产品名称	
*节点类型:	
 ● 设备 ○ 网关 	
* 使用 ID ² 认证:	
○是 ●否	
产品描述:	
请输入产品描述	
	0/100

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

特别说明:device 为deviceName。	eName可以为空,当为空明	时,阿里云会颁发全)	哥唯一标识符作	
*产品:				
mqt805		\sim		
DeviceName :				
支持英文字母、数字	阳特殊字符@:,长度刚	長制4~32		

添加完显示如下:(设备未连接显示为"未激活")

设备列表						批量添加	添加设备
请输入DeviceName	请选择设备标签 >	搜索					
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 🚺	2011	查看 删除	余

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 	a19oPEbcmAI 复制	区域	华东2(上海)
节点类型	상음	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 🕘	測试				

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)

上海泗博自动化技术有限公司

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
/a19oPEbcmAl/12345678/update	发布	0	发布消息
/a19oPEbcmAl/12345678/update/error	发布	0	发布消息
/a19oPEbcmAl/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美列表 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	摄作
/a19oPEbcmAl/\$(deviceName}/update/error \times	发布		編輯 删除
/a19oPEbcmAl/\${deviceName}/update/error	发布		编辑 删除
/a19oPEbcmAl/\${deviceName}/get	订阅		編編 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 在线测	司武
服务端订阅 • 3. 忽还没有选择推送给服务端订	使用数程 设置
服务端订阅 (推送MNS) 您还没有选择推送给服务端订	1间的消息关键 立即设置
配置服务端订阅 选择推送的消息类型: ② 设备上报消息 ③ ② ③ ② ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ③ ④ ④ ④ ④ ④	★ 恋変化通知



目列化力	技术有限公司	021-3126 5138	SiboTech 解决方案
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 🌑			使用数程
	() 憲还	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) 💿		
服务端订阅 (推 订阅消息类型	送MNS) 💿 设备上报消息		设置 資 資 論
服务端订阅 (推 订阅消息类型 区域	送 MNS) ② 设备上报消息 cn-shanghai		送置 済 済 減 建 议
服务端订阅 (推 订阅消息类型 区域 队列	送MNS) ② ②备上报消息 cn-shanghai aliyun-iot-a1RRixXXBA1v		送置 咨询: 建议

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。

(-)	管理控制台			投索	Q 消息 ⁸ 费F	目 工单 备案 企业	支持与服务 🔼 🏹 简体中文 🧔
	消息服务	队列列表 华北2(北京) 4	a东1(杭州) 香港 华北5(呼和浩特) 华北1(青岛) 华东2(上海) 华南1(深圳)	云命令行(Cloud Shell) 🗙 😽 🕹
	队列	华北3(张家口) 印度尼西亚(雅加波	日本(东京) 印度(孟买) 大) 徳国(法学京福) 英国	新加坡 澳大利亚(悉尼 (伦敦) 阿联酋(油種)	 3来西亚(吉隆坡) 美国(東吉尼亚) 美国(建谷) 	3	
	主题		-/ ////////////////////////////////////	(1007) (1000 (2017)		X	
a	事件通知	温馨提示: MNS计费进行调整,1	将于2018年12月06号生效新计费制	豐型 ,详情请查看公告。			
×	日志管理	队列查询: 仅支持前缀搜索,不支	持模糊搜索	援索			
^	**切又相 官方论坛	队列名称	消息生命周期(秒) 消息延时(秒	非) 活跃消息数 • •	活跃消息数 延迟消息数 。	创建/最后修改时间 开启logging	操作
@	▼ 短信		1209600 0	1 0	0	2019-03-27 20:29:35 true	修改设置 📗 🖩 🗎
•	短信概览					2019-03-27 20:29:35	发送消息 接收消息
	短信签名	aliyun-iot-a1RRkXXBA1v	1209600 0	0 0	0	2019-04-18 16:53:03 true	修改设置 删除
	短信模版		ananateur (813		50	2019-04-18 16:53:03	发送消息 接收消息 11
	32183/cs1						建议

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: { "1.1.1.0", //版本,以便不同版本的格式数据管理 "version": "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, 6 www.sibotech.net

SiboTech[®]

```
"BI6":
                                分多包发送,直至将所有数据发送完毕。
                         false
              }
          }
           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:
          {
                         "1.1.1.0",
              "version":
              "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"
```

```
www.sibotech.net
```



```
}
<u>只配置了一路_DI;</u>
{
"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):

💬 Mbslav1		
ID = 1		
00001 = 0	00009 = 1	
00002 = 1	00010 = 0	
00003 = 0	00011 = 0	
00004 = 0	00012 = 0	
00005 = 0	00013 = 0	
00006 = 0	00014 = 0	
00007 = 1	00015 = 0	
0 = 80000	00016 = 1	

{

```
      "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):

www.sibotech.net



💬 Mbsla	iv1		
ID = 1			
40001 =	• 0x1111	40009 = 0x9999	
40002 =	0x0000	$40010 = 0 \times 0000$	
40003 =	• 0x3333	$40011 = 0 \times 1111$	
40004 =	0x0000	$40012 = 0 \times 0000$	
40005 =	0x5555	$40013 = 0 \times 1313$	
40006 =	0x0000	$40014 = 0 \times 0000$	
40007 =	• 0x7777	$40015 = 0 \times 1515$	
40008 =	0x0000	40016 = 0x0000	

{

```
"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":

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:

1	🥰 Mbpoll1 👘 🔲 🖾													
Тx	=	64:		Err = 0:	II) =	1:	F	=	01:	SR	=	1000ms	
000	01	=	0	00009	=	0								
000	02	=	0	00010	=	0								
000	03	=	1	00011	=	0								
000	04	=	0	00012	=	0								
000	05	=	0	00013	=	0								
000	06	=	0	00014	=	0								
000	07	=	0	00015	=	0								
000	08	=	0	00016	=	0								

{



"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:

Tx = 771			🗒 Mbpoll1 💿 💌													
	: Err = 1	2:	ID	=	1:	F	=	01:	SR	=	1000ms					
00001 =	0 00009	=	0													
00002 =	0 00010	=	0													
00003 =	0 00011	=	0													
00004 =	1 00012	=	0													
00005 =	1 00013	=	0													
00006 =	0 00014	=	0													
00007 =	0 00015	=	0													
00008 =	0 00016	; =	0													

{

```
"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:



1	🥦 Mbpoll1 👘 📼 💌															
Tx	=	131	13:	Eri	: =	30	: II	D =	1:	F	=	01:	SR	=	1000ms	
000	001	=	0	00	000	9 =	1									
000	002	=	0	00	01	0 =	1									
000	003	=	0	00	01	1 =	0									
000	004	=	0	00	01	2 =	0									
000	005	=	0	00	01	3 =	0									
000	006	=	0	00	01	4 =	0									
000	007	=	1	00	01	5 =	0									
000	008	=	1	00	01	6 =	0									

{

"version": "1.1.1.0",

"type": "MODBUS_TCP_SLAVE",

"block": 0,

" addr ":6,	//addr 是 Mod	lbus	寄存器起始地址,从0开始
"npoint": 4,	//npoint 不为	J 1,	说明连续改变多个线圈的值,
"data": "c0,03",	此例子需要	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:

💬 MI	opol	11		
Tx =	11(0: Err =	0: ID = 1: F = 03: SR	= 1000ms
4000	1 =	0x0000	$40009 = 0 \times 0000$	
4000	2 =	0x1122	$40010 = 0 \times 0000$	
4000:	3 =	0x3344	$40011 = 0 \times 0000$	
4000	4 =	0x0000	$40012 = 0 \times 0000$	
4000	5 =	0x0000	$40013 = 0 \times 0000$	
4000	6 =	0x0000	$40014 = 0 \times 0000$	
4000	7 =	0x0000	$40015 = 0 \times 0000$	
4000	8 =	0x0000	$40016 = 0 \times 0000$	

{

"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
}
```

```
www.sibotech.net
```



} (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 为1时,说明只改变了一个寄存器, "npoint": 1, "data": "FF" 那么 data 中为 00 或 FF, 分别对应 0 或 1 } Mbslav1 ID = 1

00001 = 1 00009 = 000010 = 0 00002 = 000003 = 0 00011 = 000004 = 000012 = 000005 = 0 00013 = 000006 = 000014 = 000007 = 000015 = 000008 = 000016 = 0

写多个线圈:

{

"type": "MODBUS_MASTER",



```
      "slave id": 1,

      "fc": 15,
      //fc 是 Modbus 的功能码

      " addr ": 0,
      //npoint 为 2 时,与发布时的位对应关系一致

      "npoint": 2,
      //npoint 为 2 时,与发布时的位对应关系一致
```

}

📴 Mbslav1		
ID = 1		
00001 = 1	00009 = 0	
00002 = 1	00010 = 1	
00003 = 1	00011 = 0	
00004 = 1	00012 = 1	
00005 = 1	00013 = 0	
00006 = 1	00014 = 1	
00007 = 1	00015 = 0	
00008 = 1	00016 = 1	

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"

}

40009 = 0x0000	
$40010 = 0 \times 0000$	
$40011 = 0 \times 0000$	
$40012 = 0 \times 0000$	
$40013 = 0 \times 0000$	
$40014 = 0 \times 0000$	
40015 = 0x0000	
$40016 = 0 \times 0000$	
	40009 = 0x0000 40010 = 0x0000 40011 = 0x0000 40012 = 0x0000 40013 = 0x0000 40014 = 0x0000 40015 = 0x0000 40016 = 0x0000

www.sibotech.net



上海泗博自动化技术有限公司

021-3126 5138

预置多个寄存器:

{

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

"slave id": 1,

"fc": 16,

"addr": 0,

"npoint": 32,

"data":

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"



💬 Mb	poll	1									-	- • •
Tx =	243	3:	Err = 0	: II) =	1:	F	=	02:	SR	=	1000ms
10001	=	1	10009	= ()							
10002	2 =	0	10010	= ()							
10003	3 =	0	10011	= ()							
10004	=	0	10012	= ()							
10005	; =	0	10013	= ()							
10006	5 =	0	10014	= ()							
10007	=	0	10015	= ()							
10008	=	0	10016	= ()							

```
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
```

}

{

🗒 M	Mbpoll1																
Tx =		454	14:	Eri	=	14	226	5:	ID	=	1:	F	=	02:	SR	=	1000ms
1000	1	=	0	10	000	9	=	0									
1000	2	=	1	10	01	0	=	0									
1000	3	=	0	10	01	1	=	0									
1000	4	=	1	10	01	2	=	0									
1000	5	=	0	10	01	3	=	0									
1000	6	=	1	10	01	4	=	0									
1000	7	=	0	10	01	5	=	0									
1000	8	=	1	10	01	6	=	0									

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"
```

```
}
```



SiboTech 解决方案

💬 Mbpoll1		
Tx = 206: Err =	0: ID = 1: $F = 04$:	SR = 1000ms
$30001 = 0 \times 0000$	30009 = 0x0000	
$30002 = 0 \times 0222$	$30010 = 0 \times 0000$	
$30003 = 0 \times 0333$	$30011 = 0 \times 0000$	
30004 = 0x0444	$30012 = 0 \times 0000$	
$30005 = 0 \times 0555$	$30013 = 0 \times 0000$	
$30006 = 0 \times 0666$	$30014 = 0 \times 0000$	
$30007 = 0 \times 0777$	30015 = 0x0000	
$30008 = 0 \times 0000$	$30016 = 0 \times 0000$	

