

MODBUS 智能网关用户手册

http://www.brimesh.com

Copyright 2012-2021 @上海光因科技有限公司



目录

1	产品简	简介	5
	1.1	概述	5
	1.2	应用场景	5
	1.3	产品优势	9
	1.4	产品外观	. 11
	1.5	规格参数	. 13
2	入门打	皆南	. 16
	2.1	连接网络	. 16
	2.2	连接串口	. 17
	2.3	连接电源	. 17
	2.4	RESET 键	. 17
3	网关的	的基本概念	. 18
	3.1	网关的 Modbus 功能码定义	. 18
	3.2	Modbus 数据地址	. 19
	3.3	Modbus 的字节序	. 19
	3.4	网关虚拟地址空间	. 20
	3.5	网关位地址和字地址的关系	. 22
	3.6	网关故障标志的地址空间	. 23
	3.7	网关内变量的表达方式	. 24
	3.8	组态软件如何读取网关数据	. 24
	3.9	网关中主站模式和从站模式的区别	. 25
	3.10	Modbus TCP 和 Modbus RTU Over TCP 的区别	. 25
4	主站桥	莫式	. 27
	4.1	采集 Modbus RTU 从站	. 27
	4.2	采集 Modbus TCP 从站	. 29
	4.3	采集 Modbus RTU Over TCP 从站	. 30
	4.4	批量导出导入配置	. 30
5	从站槽	莫式	. 32
	5.1	创建 RS-232/RS-485 从站	. 32
	5.2	创建 Modbus TCP 从站	. 33
	5.3	创建 Modbus RTU Over TCP 从站	. 33
6	数据证	运算处理	. 34
	6.1	运算例子 1:高低字节序调换	. 34
	6.2	运算例子 2: 由不同字节组成一个整数	. 34

6.3 7	上传	运算例 数据到无	列子 3: 云端	浮点转整型	. 35 . 36
7.1		通过N	Modbus	s 连接云端	. 36
7.2		诵讨 N	MOTT 냍	车接云端	. 37
		7.2.1	MQTT	配置	. 37
		7.2.2	Publisł	n Topic 数据格式	. 39
		7.2.3	Comm	and Topic 数据格式	. 39
		7.2.4	Respoi	nse Topic 数据格式	. 40
		7.2.5	MQTT	调试	. 40
8	数据	显示			. 42
81		 守时迷	な足		42
0.1					42
8.2		山(四) (二)	X店		. 43
9	系统	日志解读	卖		. 45
9.1		回复赶	舀时		. 45
9.2		采集人	人站异常		. 45
9.3		ト位相	几主站异	常	. 45
10	安뎺		·—· [//		47
10.1	رم جر انہ 1	 网关利	 D变频器	与继电器模块通讯	. 47
		10.1.1	模拟变	频器和继电器模块	. 47
		1012	「主立」	計1下配署亦物哭和继由哭柑也	49
		10.1.2	[4]/≯		J
		10.1.3	[从站模]王]下创建本地从站	. 52
		10.1.4	上位机	读取网关数据	. 53
10.	2	通过N	MQTT	E接阿里云物联网平台	. 57
		10.2.1	阿里云	配置	. 57
		10.2.2	网关配的	置	. 64

10	.3 通过 KEPServerMqtt 远程监控网关	74
	10.3.1 网关配置	74
	10.3.2 KEPServer 配置	76
10	.4 通过 thingsboard 远程监控网关	88
	10.4.1 Thingsboard 配置	88
	10.4.2 网关配置	90
	10.4.3 查看变频器当前频率的历史曲线	91
	10.4.4 远程控制继电器,设置频率并查看频率曲线	94
	10.4.5 Restful API	108
11 常见	见问题	111
11.	.1 如何判断从站数据是否已经采集上来	111
11.	.2 上位机如何通过网关把数据写入从站	111
11.	.3 如何对从站只写不读	111
11.	.4 如何实现 2 个从站间的数据搬运(通讯)	112
11.	.5 如何升级固件	112



1 产品简介

1.1 概述

Modbus 智能网关是我们新推出的产品,专门针对 PLC,HMI 或者组态软件的采 集。传统的 Modbus 网关只是 Modbus TCP 和 Modbus RTU 协议做个转换,网关本 身并不主动采集数据,而我们的采集网关是可以主动采集从站数据并缓存到网关内部, 从而上位机可以通过 TCP 或者 RS-485 总线直接从我们采集网关取数据,这样就减少 了上位机主站与从站的交互次数,极大地提高了采集速度,实时性,可靠性以及稳定性; 对于多 RS-485 主站想要访问相同的从站数据,传统的 Modbus 网关是无法解决的, 而由于我们的网关是自动采集并存储数据,因此完全支持多主站同时通过 RS-485 总线 读取数据;另外,针对一些没有以太网口的 PLC 或者 HMI,也可以通过我们的 Modbus 智能网关的 RS-232/RS-485 端口,采集到 Modbus TCP 从站的数据。

1.2 应用场景



● 通过 MQTT/Modbus 连接到云端



● 多 Modbus TCP 主站同时访问, 毫秒级响应速度:





● 多 RS485 主站同时访问多 RS485 从站:



(Modbus RTU Master)



● Modbus RTU 主站访问 Modbus TCP 从站:



1.3 产品优势

Modbus 协议在工业中得到广泛的应用, 而随着工业 4.0 的发展趋势, 越来越多的 支持 Modbus 协议的仪器仪表要求联网监控, 传统的 Modbus 网关就是把原来支持串 口 Modbus 的设备适配成支持网口, 这样就能方便联网。而我们的 Modbus 智能网关 相比传统 Modbus 网关做了很多的改进, 如下:

- 更高的采集速度。在采集多个 Modbus 从站数据时比传统 Modbus 网关采集速度 快很多,传统的 Modbus 网关只是 Modbus RTU 与 Modbus TCP 的协议转换, 本身并不处理数据,而我们的智能网关是自动采集数据并缓存到本地。因此,如果 主站要采集 100 台从站数据,原本要 100 次读命令,而通过 Modbus 智能网关, 最优的情况是只需要一次读命令就可以。
- 更高的可靠性。将多个读命令中的数据集中到几个命令来完成,减少数据采集过程 中来回交互的次数,也就减少了出错概率。
- 支持 MQTT/Modbus 连接云端。 用户可自定义上传数据的格式,然后网关通过 支 MQTT 协议发布到云端。同时也支持 Modbus TCP 协议连接云端,网关作为 TCP Client,云端主动发送 Modbus 读写请求。
- **支持数学运算处理**。支持高低字节交换,浮点运算。
- 编程更简单。通过 PLC 或组态软件采集数十个设备的数据,编程过程比较繁琐。而
 通过智能网关,编程过程变得简单很多,只需要集中处理所采集的数据。
- 支持多 RS485 上位机。传统 Modbus 网关是不能支持 485 总线上多个主站的, 原因是多主站会产生数据冲突。而使用 Modbus 智能网关就不存在这个问题。多 主站是直接与智能网关通讯,读写智能网关缓冲中的数据。
- 支持多 Modbus TCP 上位机。传统 Modbus TCP 网关在多个主站同时访问从站时,采用排队的机制,这受制于 485 总线同时只能有一个主站。因此同时访问的主站越多,回复会越慢。而我们的智能网关是把数据缓冲在设备内部,实时读取,没有延时。
- 支持 64 位浮点运算。网关可以在本地处理整型,单精度浮点,双精度浮点运算;
 处理字节调换等复杂的数学运算,简化上位机 (PLC/SCADA/HMI)的工作量,特别是 PLC,在处理字节序转换,浮点运算比较麻烦。
- **支持透传模式。**网关也可以配置成透传模式,并且支持多 TCP 上位机访问。
- **支持串口集线器功能**。多个RS232/RS485主站同时访问一个RS232/RS485从站, 支持任何主从串口通讯的协议。
- 完善的错误跟踪机制。传统 Modbus 网关只数据转换,不做数据交互的逻辑判断, 所以不会记录错误信息,这些错误分析都需要在上位机处理。在采集很多从站时, 若有一台从站出问题,很难在很短时间内排查出来。而我们的智能网关内部集成了



详细的日志功能,可以快速找到出问题的从站,前期调试非常方便和迅速。同时, 上位机也可以读取所有从站的工作状态,及时发现问题并通知工作人员。

● 降低工程成本。原先需要多台 PLC 采集数据,现在一台数据采集器配合一台 PLC 就可以完成采集。



1.4 产品外观







1.5 规格参数

		GY-G300	GY-G600	GY-G6A	GY-G6H-60K									
以太网	端口数量		1											
	速率	10/100 Mbps,	10/100 Mbps, 自适应 MDI/MDIX 8 针 RJ45											
	接口													
	电磁隔离	内建 1.5 KV												
串口	串口数量	3	6	6										
	电磁隔离	采用优于光耦隔离	的芯片级隔离方案,降	隔离电压 2.5KV										
	串口类型	RS-232/RS-485	RS-232 x 3	RS-485 x 6										
		二选一	RS-485 x 3											
	接口	RS-232: DB9		18P 3.81mm 接线端子										
		RS-485:端子												
	串口信号	RS-232: RX,TX, C	SND, RTS, CTS	RS-485: A(+), B(-), GND										
		RS-485: A(+), B(-	-), GND											
串口	波特率	1200 ~ 230400												
通信	数据位	7,8												
	停止位	0.5, 1, 1.5, 2												
	校验位	None, Even, O	dd											
Modbus	主站采集	除了可采集 Modl	bus RTU/TCP 从站,	支持写操作;每一路 RS-485 最多挂 32 个从站	5;最多可采集 490 条从站纪									
		录												
	本地从站	PLC/HMI 等主站	可通过 Modbus RTU/	TCP 读取采集的数据										
		每一个 Modbus ⁻	TCP 服务最多同时支持	8个 TCP 主站连接访问,最多可创建 16个本均	也主站									

	存储空间	1万个字		3万个字							
机械特性	外壳	铝合金	铁								
	重量	200g	450g								
	尺寸	无挂耳:134 x 85 x 25mm	160 x 120 x 35mm								
		有挂耳:156 x 85 x 25 mm									
工作环境	工作温度	-40 ~ 75 °C									
	存储温度	-40 ~ 85 °C									
	工作湿度	5% ~ 95% (无凝露)									
电源要求	输入电压	7 - 24 VDC									
	功耗	135 mA @ 12V,69 mA @ 24V									
保修	保修期	3年									



2 入门指南

2.1 连接网络

出厂 IP 地址统一是 **192.168.1.222**。先将电脑网络配置成 192.168.1.XXX 网段, 对于即有网口又有 WIFI 的笔记本电脑,如果通过网口对设备配置,建议关闭 WIFI (当 两个网络接口都是 192.168.1 网段,默认路由走 WIFI,可能导致无法通过网口配置设 备)。 然后通过 <u>http://192.168.1.222</u> 访问设备配置,出厂的用户名是 admin,密码 是 admin,打开的 WEB 配置页面如下:



建议使用 Chrome 浏览器对网关进行配置,并且配置网关后要重启后才能 生效。



2.2 连接串口

RS-232 接口是 DB9 (针式) 的, 支持的信号脚如下。

DCD RxD TxD DTR GND



DSR RTS CTS RI

DB9 - Male(RS-232)

2.3 连接电源

每个型号提供圆孔,内正外负的 DC 电源座子,电压在"规格参数"中有详细说明, 在产品外观上也有指示。同时,可以接 3.81mm 间距的绿色端子的正,负供电。

2.4 RESET 键

如果需要恢复到出厂值,例如忘记之前配置的 IP 地址或者密码,就可以使用复位 键。具体操作是,先断电,然后一直按住复位键,上电超过 5 秒钟,松开复位键,系统 就自动恢复出厂值。



3 网关的基本概念

3.1 网关的 MODBUS 功能码定义

Modbus 标准中的常用功能码是 1,2,3,4,5,6,15,16。网关定义的功能码含义稍微有 点差别,这里特别说明一下,网关定义的写功能码 5,6,15,16 是循环写,只用于数据搬 运,一般很少用。用户需要写网关的时候,在网关定义的功能码 1,3,130,131 已经包 含了。网关在接收到上位机写数据时,会自动选择用功能码并把数据写到下位机。这种 写只有在收到上位机写命令的时候才会被触发,在写完之后会恢复到循环读的状态。如 果扫描周期设置成 0,那么网关就不会做读操作,只有在上位机写数据时才会把数据写 向下位机。具体请看下面的定义:

[1] Read Coils & Write Multiple Coils: 网关周期性(周期长度可以设置,如果周期为0就是不做读操作)的使用功能码1读取数据,当上位机往网关写入位时,网关先存储数据,同时使用功能码15将虚拟地址中被更新的多个位写入从站。 网关的功能码1包含了读(1)也包含了写(15),这个写只有在上位机往网关写数据的时候才会被触发。

[2] Read Discrete Inputs: 使用功能码 2 读取数据位。

[3] Read Holding Registers & Write Multiple Registers:使用功能码 3 读取数据,当上位机有数据写入时,使用功能码 16 将虚拟地址中的数据写入从站。

[4] Read Input Registers: 使用功能码 4 读取数据。

[5] Write Single Coil: 使用功能码 5 将虚拟地址中的数据循环写入从站。

[6] Write Single Register: 使用功能码 6 将虚拟地址中的数据循环写入从站。

[15] Write Multiple Coils:用功能码 15 将虚拟地址中的数据循环写入从站。

[16] Write Multiple Registers:使用功能码 16 将虚拟地址中的数据循环写入从站。

[130] Read Coils & Write Single Coil: 跟[1] Read Coils & Write Multiple Coils 的唯一区别是,在上位机往网关写入数据时,网关使用功能码 5 一个一个分多次的往下 位机写入线圈,有些下位机只能支持写单个线圈功能码,比如继电器。

[131] Read Holding Registers & Write Single Register: 用功能码 3 读取数据, 当上位机有数据写入时,使用功能码 6 将单个 Register 写入从站。



3.2 MODBUS 数据地址

当网关采集下位机数据的时候,采用的数据地址是从0开始的标准地址,区别于组态或者 PLC 的地址 (0x, 1x, 3x, 4x)。有些下位机手册在描述数据地址是采用的是 PLC 的编写方法,对应关系如下:

若设备手册地址是 40001, 等同于 Modbus 标准地址的 0

若设备手册地址是 30001, 等同于 Modbus 标准地址的 0

若设备手册地址是 10001, 等同于 Modbus 标准地址的 0

若设备手册地址是 00001, 等同于 Modbus 标准地址的 0

也有些设备手册采用的是0开始的标准地址,若是这样就不需要转换。

3.3 MODBUS 的字节序

数据在 Modbus 协议里面通常是以大端 (Big Endian,也就是 MSB,Most Significant Byte,高字节在前)或者小端 (Little Endian, LSB, Least Significant Byte,低字节在前)存储。但也有一些其它的字节序,这里用 A 表示最高 (左边)的字节,字的低 (最右边)字节用 B 表示,双字的低 (最右边)字节用 D 表示,64 位双精度浮点的低 (最右边)字节用 H 表示。这里举个例子,比如十进制的无符号字4386,换成十六进制就是 0x1122,这里 0x11 就是高字节,0x22 就是低字节。如果依次存储在内存中是 0x11,0x22,这种存储字节序就是 Big Endian,可用 AB 来表示。如果依次存储在内存中是 0x22,0x11,那么就是 Little Endian,就用 BA 来表示。

字节序	描述										
AB											
BA											
ABCD	双字/整数(32 位)的字节序是 MSB										
DCBA	双字/整数(32 位)的字节序是 LSB										
CDAB	字的 Byte 顺序是 MSB,双字/整数 (32 位)的 Word 顺序										
	是LSB										
BADC	字的 Byte 顺序是 LSB, 双字/整数 (32 位)的 Word 顺序										
	是 MSB										
ABCDEFGH	长整型/双精度浮点(64 位)的字节序是 MSB										



HGFEDCBA	长整型/双精度浮点(64 位)的字节序是 LSB
GHEFCDAB	字的 Byte 顺序是 MSB,长整型/双精度浮点(64 位)的
	Word 顺序是 LSB
BADCFEHG	字的 Byte 顺序是 LSB,长整型/双精度浮点(64 位)Word
	顺序是 MSB

举个例子:

比如有个 32 位的整数(双字),数值是 287454020,换成十六进制就是 0x11223344。 如果用 ABCD 放置在内存就是 11 22 33 44(十六进制),如果按照 CDAB 放置在内 存就是 33 44 11 22。

反之,如果内存的 4 个字节分别是 11 22 33 44 (十六进制),按照 ABCD 的字节 序取值就是 0x11223344(十进制 287454020);如果按照 CDAB 取值就是 0x33441122, 这个稍微有点绕,其实简单,先画个表,如下

С	D	А	В
11	22	33	44

CDAB 分别对应 11 22 33 44, 然后把 CDAB 顺序排列成 ABCD, 对应的数值就是 0x33441122 (十进制 860098850)。

3.4 网关虚拟地址空间

网关的虚拟地址是一个连续的内存块,一共有 20000 个字节,最后 100 个字 (Word9900~Word9999) 系统保留。系统上电后自动分配好内存,所以用户在设置 采集下位机并把下位机数据映射到虚拟地址的时候,虚拟地址可以任意分配,前提是不 超出这 20000 个字节。

因此, 网关最多可存储 9900 个字 (点, 模拟量)。

Modbus 标准中的数据地址的单位可分为两类,一个是从0开始的**位地址**,功能码 1,2,5,15 使用的地址就是位地址。另外一个是从0开始的**字地址**,功能码 3,4,6,16 使用 的地址就是字地址。 位地址的单位就是比特位,而字地址的单位就是字(占用2个字 节)。

网关的虚拟地址空间是统一并且重合的,使用不同的 Modbus 功能码访问的都是同一个地址空间。比如要访问网关空间的 Bit 16-31 一共 16 个位,可以用访问位的功能码 (1,2) 也可以用访问寄存器 (字)的功能码 (3,4)。

用功能码 1 访问时, Modbus 标准数据地址是 16, 组态或者 PLC 设置地址是 00016 (0x),数量是 16。

用功能码 2 访问, Modbus 标准数据地址也是 16, 组态或者 PLC 设置地址是 10016 (1x),数量是 16。

用功能码 3 访问, Modbus 标准数据地址是 1 (字地址), 组态或者 PLC 设置地 址是 40002 (4x), 数量是 1 (1 个字, 也就是 16 个位)。

用功能码 4 访问, Modbus 标准数据地址是 1 (字地址), 组态或者 PLC 设置地 址是 30002 (3x), 数量是 1 (1 个字,也就是 16 个位)。

3.5 网关位地址和字地址的关系

Modbus 协议标准中的寻址只有两种, 位寻址和字寻址。通过网关可以变换寻址方式, 比如下位机只能通过位寻址(功能码 1,2,5,15)进行操作, 而上位机只能进行字寻址(功能码 3,4,6,16), 这时就要涉及位地址和字地址相互转换。

这里举个例子,例如位地址 50 (从 0 开始,对应 Modbus 功能码是 1,2,5,15), 如何通过字地址 (功能码 3,4,6,16)去操作?

首先,位在字节中的排列是从低(右)往高(左)依次递增(总共 8 个 bit),例 如,字节数据 0x0A 的二进制是 00001010,从右往左依次是 bit0 ~ bit7。

字节在字中的排序是高字节在前(下图中的左边),低字节在后(下图中的右边, 这里默认字节序是 Big endian,也是 modbus 协议的默认字节序,同时也是网络传输 的默认字节序,例如 TCP/IP 协议)。

地址			寄石	字器	高.	字节	ל (M	SB)		寄存器低字节(LSB)								数值	1
字节 (BYTE) 地址	字 (WORD) 地址	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	十六进 制	十进制
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0x0401	1025
8	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

在清楚上面的基本概念之后,我们再来分析位地址 50 (从 0 开始的第 50 个位)是 映射到哪个字地址的哪个位。50 除以 16 等于 3.125,也就是从 0 开始的的字地址 3

(上表中的黄色部分)。然后 50 % 16 = 2 (百分号%是余运算符号)。然后左移 2 位 (上表中的红色部分),就是 0x04,然后又是高字节,转成字就是 0x0400。因此往字 地址 3 写入 16 位数据 0x0400 (十进制是 1024),就可以对应位地址 50。假如位地 址是 56,那么 56 % 16 = 8,然后 8 % 8 = 0 (低字节的 bit0),因此直接往字地址 3 写如 1,就可以对应到位地址 56。

3.6 网关故障标志的地址空间

网关的故障位存储在系统空间的 Word9900~Word9931 (*型号 GY-G6H 内存是 60K Bytes, 故其故障位所在的地址是 Word29900 ~ Word29931*), 一共 32 个字 (64 个字节, 512 个位)。在主站模式下,每一条记录有个 ID, ID 从 1 到 500, 类似数据 库中的索引号。每一个 ID 对应一个比特位,顺序依次从第一个字节的低位开始到高位, 然后第二个字节的地位到高位,以此类推。

举个例子, 比如 Word9900 = 0x010B, 换成二进制就是 0000 0001 0000 1011, 其中 0000 0001 是高字节 0x01, Bit0~Bit7 对应 ID1~ID8, 最低比特是 ID1,高比特是 ID8。这里 0x01 中的 Bit0 是 1, 表示 ID1 出问题了,通常是掉线了。0000 1011 是低 字节,Bit0~Bit7 对应 ID8~ID16,其中 Bit0,Bit1,Bit3 都是 1,表示 ID8, ID9,ID11 出 问题了。简单说,Word9900.0 ~ Word9900.7 对应 ID8 ~ ID16,Word9900.8 ~ Word9900.15 对应 ID1 ~ ID8。

通常组态或者 PLC 用 4 号功能码(只读),对应的起始地址是 39900,这样就可 以访问网关的故障位区域。



3.7 网关内变量的表达方式

本小节的概念只有【数学运算】和 MQTT 通讯中才会用到,可直接跳过。 变量定格式定义:

V [Data Type] [Byte Address] # [Byte Order]

V	变量开始的	变量开始的标识符号							
Data Type	由BWDRLCSI表示								
	В	8位无符号字节							
	W	无符号字							
	D	无符号双字							
	R	32 位浮点							
	L	64 位双精度浮点							
	С	8位有符号字节							
	S	16 位有符号字							
	I	32 位有符号双字(整数)							
	N/A	V 后面直接跟字节地址,表示访问位							
Byte Address									
	成字节地址就要乘以 2, 也就是 200。								
#	井号表示后	面跟的是字节序表达式,可以省略掉,表示默认的 MSB。							
Byte Order	参考 Modbus 的字节序章节								

3.8 组态软件如何读取网关数据

组态设置的地址通常用 0x,1x,3x,4x 表示,这跟网关的地址有所不同。网关的地址 是 Modbus 标准协议的地址(都从 0 开始),并且是统一的地址空间。

组态软件读线圈通常用 0x 地址,比如 DO。组态配置地址时只需要把网关的地址 加 1。假设网关的地址是 0,那么组态就配置成 00001 地址 (加 1), **注意有些组态配 置的线圈地址前面必须用 0 补全成为 5 位数的地址,比如 32,那么就用 00032 表示**。 其实组态软件读 (功能码 1)或者写线圈 (功能码 5,15)的时候,会自动把用户配置的 地址减 1,变成标准 Modbus 数据地址 (Modbus Data Address),然后发送给网关。

组态软件读只读的离散位通常用 1x 地址, 比如 DI 类型。配置的时候, 组态软件需要把目标网关的地址加 10001。组态软件用功能码 2 读取数据。

组态软件读可读可写寄存器时,通常用 4x 地址,比如 AO 类型。配置的时候,组 态软件需要把目标网关的地址加 40001。组态软件用功能码 3 读取数据。

组态软件读只读寄存器时,通常用 3x 地址,比如 AI 类型。配置的时候,组态软件 需要把目标网关的地址加 30001。组态软件用功能码 4 读取数据。

总结: 无论 0x,1x,3x,4x,这些都会被转换成标准的从 0 开始的 Modbus 数据地址,跟网关通讯。而网关本身就是采用的就是标准 Modbus 数据地址(绝对地址),内部配置(在网页上配置)不需要做+1,+10001,+30001 或者+40001 的偏移。

3.9 网关中主站模式和从站模式的区别

主站模式只跟下位机(从站)通讯,而从站模式只跟上位机(主站)通讯。

在主站模式下配置时,可以添加基于串口的 Modbus RTU 从站,或者添加基于 Modbus TCP 的从站。网关会对用户配置的从站不停地轮训采集数据并储存在网关内 存中。

在从站模式下,网关也可以创建基于某个串口的 Modbus RTU 从站或者基于网口的 Modbus TCP 从站,这样上位机就可以读走网关所采集的数据。

主站模式下的从站不要跟从站模式下的从站混淆,前者是下位机,后者可以理解为 网关内建的从站。

3.10 MODBUS TCP 和 MODBUS RTU OVER TCP 的区别

在 RS-232/RS-485 上通讯的 Modbus 称为 Modbus RTU 协议,如下图,绿色的 是数据帧部分,第一个字节是从站地址,最后 2 个字节是 CRC 校验。



而 Modbus RTU 协议除了串口也可以在 TCP 上运行,整个帧格式不变,这种通讯 方式称为 Modbus RTU Over TCP。而 Modbus TCP 协议是有改动的,去掉最后 2 个 字节的 CRC 校验 (底层 TCP 协议已经包含了 CRC 校验,并且网口不像串口通讯那样



干扰大,因此不需要 CRC),并在数据帧前面加了一个协议头,称为 MBAP Header,如下图:



MBAP协议头包含了事务标识(Transaction Identifier, 2个字节, 主站每发送一个请求, 该标识都加1, 用于回复与请求的匹配验证), 协议标识(始终是0, 占用2个字节), 数据长度(2个字节)以及单元标识(Unit Identifier 对应 Modbus RTU协议中第一个字节的从站地址, 1个字节), 总共7个字节。



4 主站模式

主站模式和从站模式一起,对 Modbus 从站的数据进行重新组态。网关可以通过 RS-232/RS-485,也可以通过以太网采集从站数据。 打开网页,以 GY-G600 为例, 左侧的菜单可以看到如下图的子菜单。



点击子菜单,将显示所有的基于对应的串口或网络所要采集的从站信息

主站采集												
	ID	接口	从站 功能 地址	韬 数据 地址	数量	虚拟地址	重试 次数	扫描 周期	回复 超时	命令 延时	配置	
		[A]	[B] [C) [D]	[E]	[F]	[G]	[H]	[1]	[J]		
添加从站 全部删除 下载从站 帮助												

4.1 采集 MODBUS RTU 从站

点击"添加从站"按钮, 网页会弹出"从站配置"窗口, 如下图,



从站配置		
ID:	1	☑自动增加
接口:	RS232-1	-
从站地址:	1	☑自动增加
功能码:	[3] Read Holding Registers & Write Multiple Re	egisters 🗸
数据地址:	0	□目动増加
数量:	10]
虚拟地址:	1000	】☑自动增加
重试次数:	3]
扫描周期:	200	ms
回复超时:	500	ms
命令延时:	0	ms
创建数量:	1]
保存帮助	返回	

在点击保存之后,从站列表中会显示增加的这条记录,如下图:

主站系	¥集													
	ID	接口	从站 地址	功能码	数据 地址	数量	虚拟地址	重试 次数	扫描 周期	回复 超时	命令 延时		配置	
		[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[1]	[J]			
	1	RS232-1	1	3	0	10	1000	3	200	500	0	Edit Del		
添加	加从这	沾 全部	删除] 下载	认站	帮助]							

注意:对从站进行配置的时候, Modbus 采集会停止, 新配置在重启后才会生效。

字段名称	说明
ID	范围 1~500, 最大可以存储 500 条记录, 这个只是表中的索引号, 不要
	跟 Modbus 从站地址混淆。网关会根据指定的 ID 存放在表中对应的位置。
自动增加	【自动增加】选项用于连续创建多条记录。【ID】和【从站地址】每次增
	加的值是1,【数据地址】和【虚拟地址】增加的值就是【数量】个单位。
	再创建完一条或多条记录后,对应的字段值会自动增加,方便创建新记录。
从站地址	要采集的 Modbus 从站地址,设置范围是 1~255。
功能码	参考 网关的 Modbus 功能码定义章节的描述。
数据地址	参考 Modbus 数据地址章节的描述。
数量	如果功能码是对寄存器操作,这的单位就是寄存器个数,一个寄存器占用
	2 个字节宽度。如果功能码是对线圈操作,那么单位就是 Bit。
虚拟地址	参考网关虚拟地址空间章节的描述,注意,每条新添加的记录的虚拟地址
	不能跟之前的重合。
重试次数	通讯失败后连续重试的次数,设置范围1~255。重试次数超过设定值,该



	记录对应的故障位被置位,网关继续以【扫描周期】的时间间隔不间断重
	试,直到恢复通讯并清除故障位。
扫描周期	执行本操作所间隔的时间,单位是ms,设置范围0~65535。注意:特殊
	值0将不做扫描动作,这样的设置通常用于只写不读的场景。
回复超时	ModBus 命令发出后,等待回复的时间,单位毫秒。
命令延时	网关收到 ModBus 回复后, 在发送下一条 ModBus 命令前的等待时间 (毫
	秒),特殊值0表示由系统自动设置。
创建数量	一次创建采集记录的数量,如果记录数量大于1,请参考'自动增加'的规则
	进行设置。

4.2 采集 MODBUS TCP 从站

从站配置		
ID:	2	🕑 自动增加
接口:	Modbus TCP 🔹	
从站IP:	192.168.1.200]
从站端口:	502]
从站地址:	1	🕑 自动增加
功能码:	[3] Read Holding Registers & Write Multiple Register	s 🔻
数据地址:	0	□ 自动增加
数量:	10]
虚拟地址:	0	🕑 自动增加
重试次数:	3]
扫描周期:	1000	ms
回复超时:	1000	ms
命令延时:	0	ms
创建数量:	1]
保存帮助	助返回	



4.3 采集 MODBUS RTU OVER TCP 从站

从站配置		
ID:	3	☑ 自动增加
接口:	Modbus RTU Over TCP V	-
从站IP:	192.168.1.200]
从站端口:	502]
从站地址:	1	🕑 自动增加
功能码:	[3] Read Holding Registers & Write Multiple Register	s 🔻
数据地址:	0	□ 自动增加
数量:	10]
虚拟地址:	0	✓ 自动增加
重试次数:	3	ĺ
扫描周期:	1000	ms
回复超时:	1000	ms
命令延时:	0	ms
创建数量:	1]
保存帮助	助返回	

参考: Modbus TCP 和 Modbus RTU Over TCP 的区别章节。

4.4 批量导出导入配置

当网关内的采集的从站比较多的时候,配置导出导入功能就很有用,方便测试或者 现场更换设备。比如用户配置了3个从站,如下图

站采集											
	接口	从站 地址	功能码	数据	数量	虚拟地址	重试 次数	扫描 周期	回复 超时	命令 延时	配罟
	[A]	[B]	[C]	[D]	[E]	[F]	[G]	[H]	[1]	[J]	
1	RS232-1	1	3	0	10	0	3	1000	1000	0	Edit <u>Del</u>
2	Modbus TCP 192.168.1.200:502	1	3	0	10	10	3	1000	1000	0	<u>Edit</u> <u>Del</u>
3	Modbus RTU Over TCP 192.168.1.201:502	2	3	0	10	20	3	1000	1000	0	<u>Edit</u> <u>Del</u>
添加从這	站 全部删除 下载从站 帮	助									

点击【下载从站】按钮,默认文件名是 Slaves.csv,用 Excel 打开,A-J 列对应上图的字段。

	A	В	С	D	E	F	G	Н	I	J
1	1	1	3	0	10	0	3	1000	1000	0
2	100:192.168.1.200:502	1	3	0	10	10	3	1000	1000	0
3	101:192.168.1.201:502	2	3	0	10	20	3	1000	1000	0



 字		台行	而	근된	罜	•
1女	ш	נה	H	υĒ		•

接口	描述
RS-232/RS-485	数值 1-6 分别标识串口 1-6
Modbus TCP	格式: 100:192.168.1.200:502, 中间用冒号分割
	接口: 100 表示从站是 Modbus TCP Server
	IP: 192.168.1.200 表示服务器 (从站) 的 IP 地址
	端口:502 表示 TCP 端口
Modbus RTU Over TCP	格式: 100:192.168.1.201:502, 中间用冒号分割
	接口: 101 表示从站是 Modbus RTU Over TCP Server
	IP: 192.168.1.201 表示服务器 (从站) 的 IP 地址
	端口:502 表示 TCP 端口

当需要批量导入配置的时候,选择要导入的配置文件,点击【上传从站】按钮,配置就可以导入到网关,重启网关就可以生效。

上传从站采集表

C:\Users\Administrator\De 浏览... 上传从站



5 从站模式

从站模式是跟上位机通讯,可以创建基于 RS232/RS485 的从站,也可以创建基于 Modbus TCP 从站。

5.1 创建 RS-232/RS-485 从站

本地从站西	置							
ID	接口	从站地址	0x位偏移	1x位偏移	3x字偏移	4x字偏移	配置]
添加本地	以此站 全部	删除 帮助						

在【从站模式】下选择【添加从站】按钮

编辑本地从站	
ID:	1
接口:	RS485-1 V
从站地址:	1
0x位偏移:	0
1x位偏移:	0
3x字偏移:	0
4x字偏移:	0
保存 帮助 過	

名称	说明
ID	本地从站在表中的索引号,范围1~16。
接口	RS-232/RS-485指设备选择对应的串口作为从站接口,或者 Modbus TCP
	(Server),Modbus RTU Over TCP(Server)。
服务端口	TCP 服务器端口,只有接口是 Modbus TCP 或者 Modbus RTU Over TCP
	オ需要配置。
从站地址	本从站的 Modbus 设备地址,范围是 1~255。
0x 位偏移	Modbus 功能码 1, 5, 15 在虚拟地址中的起始位偏移,设置范围 0 ~
	65535。
1x 位偏移	Modbus 功能码 2 在虚拟地址中的起始位偏移,设置范围 0 ~ 65535。
3x 字偏移	Modbus 功能码 4 在虚拟地址中的起始字偏移,设置范围 0 ~ 65535。
4x字偏移	Modbus功能码 3,6,16 在虚拟地址中的起始字偏移,设置范围 0~65535。

注意: 偏移设置只在上位机跟网关通讯时有效,而网关跟下位机通讯时无效。并且偏移都是正偏移,也就是基于上位机发给网关的数据地址上再增加用户设置的偏移值。

比如上位机的程序的数据地址是固定的,比如 40001 (对应 3 号功能码,4x), 而下位机采集来的数据存储在寄存器地址 100,这里就可以在创建的从站里面的 4x 设 置 100,这样上位机通过 40001 就可以访问寄存器地址 100 开始的数据,网关会把上 位机给的地址自动加上 100。

5.2 创建 MODBUS TCP 从站

编辑本地从站			
ID:	1		
接口:	Modbus TCP		
服务端口:	502		
从站地址:	1		
0x位偏移:	0		
1x位偏移:	0		
3x字偏移:	0		
4x字偏移:	0		
保存帮助返回			

5.3 创建 MODBUS RTU OVER TCP 从站

编辑本地从站			
ID:	1		
接口:	Modbus RTU Over TCP V		
服务端口:	502		
从站地址:	1		
0x位偏移:	0		
1x位偏移:	0		
3x字偏移:	0		
4x字偏移:	0		
保存帮助返回			



6 数据运算处理

数学运算的功能可对网关采集的数据进行运算,比如简单的高低字节调换,这个在 PLC 里面处理比较麻烦。

数学运算表达式由运算脚本组成,一行代表一个运算表达式,行的长度不限,可以 有多行,整个运算脚本不超过4096个字符。表达式由变量和加减乘数,括号构成,类 似 C 语言。下图是运算脚本输入框,保存重启后运算脚本会一直循环执行。

L28=(VR0 * 100) D4000#DCBA=VD2000#ABCD	

关于变量的定义请参考 网关内变量的表达方式章节。

6.1 运算例子 1: 高低字节序调换

VD4000#DCBA=VD2000#ABCD

把字节地址 2000 开始的双字由 MSB 转换成 LSB,并且存储在字节地址 4000 开始的 内存中。运算结果不建议覆盖,这样方便调试数据。

6.2 运算例子 2: 由不同字节组成一个整数

VD2010=VB2001*16777216+VB2003*65536+VB2005*256+VB2007



把 VB2001,VB2003,VB2005,VB2007 由高到低字节组成一个 32 位整数,并存储在地址 2010。16777216 就是 16 进制的 0x1000000,乘以该值表示左移 24 位,乘以 65536 就是左移 16 位,乘以 256 就是左移 8 位。

6.3 运算例子 3: 浮点转整型

VD20=(VR0 * 10)

把地址 0 的浮点乘以 10 并取整存储在地址 20 开始的双字中。



7 上传数据到云端

网关可以通过 Modbus 协议或者 MQTT 协议连接到云端,以此实现向云端上传数 据或者接收云端的命令。



7.1 通过 MODBUS 连接云端

Modbus to Cloud			
Protocol: 主机地址: 端口:	Modbus TCP mdbus.brimesh.com 502		
注册包: 心跳包:	AA0B000001 HEX Enable AA0C000001 HEX Enable		
 小跳间隔: 10s ▼ 保存 返回 			

Protocol: 用户可以选择 Modbus TCP 或者 Modbus RTU Over TCP 协议。网 关作为 TCP Client 连接到服务器。在 Modbus TCP 通讯中,一般作 TCP Client 的都 是 Modbus 主站,也就是说主动发送 Modbus 命令请求。但在连接云端的时候正好反 过来,作为 TCP Client 的网关,同时也是 Modbus 从站,Modbus 命令请求由云端发 送,网关只返回结果。

主机地址: 云端的域名地址或者 IP 地址。

端口: 云端的服务器 TCP 端口。

注册包: 用户自定义注册包数据,可以 ASCII 文本,也可以十六进制数据。当网 关上电后,网关在与服务器建立 TCP 连接之后,如果 Enable 被勾选,网关就会主动发 送注册包。如果 TCP 断开重连的时候,注册包也会被发送一次。平常通讯时是不会发 送注册包的。 注册包主要是服务器用来识别网关。


心跳包: 用户自定义的心跳包数据,可以 ASCII 文本,也可以十六进制数据。每间隔一段时间,网关就会主动发送心跳包给服务器。

云端 (Modbus 主站)

7.2 通过 MQTT 连接云端

MQTT 的配置比较简单,但用户需要对 MQTT 有一定的了解。关于 MQTT 协议的 说明请参考 <u>http://mqtt.org</u>。

7.2.1 MQTT 配置



MQTT to Cloud	
Enable MQTT	
MQTT Broker:	www.youdomain.com
Port:	1883
ClientID:	sdaf231
Keep Alive(s):	10
Publish Topic:	/test/pub
Command Topic:	/test/cmd
Response Topic:	/test/feedback
Response Delay(s):	2
Publish Interval(s):	10
AUTH Enable	
User Name:	test
Password:	test
保存返回	

MQTT数据格式定义	
上传	浏览 未选择文件。
查看下载	

MQTT 调试工具 MQTT.fx 下载链接: <u>http://www.jensd.de/apps/mqttfx</u>

MQTT 配置参数	说明
MQTT Broker	MQTT 的服务器域名或者 IP 地址
Port	MQTT 服务器端口, 一般是 1883
ClientID	用户自定义的唯一标识网关的识别码,不同网关不能重合,否则
	会被服务器踢掉。
KeepAlive	MQTT 心跳包(也就是 PING 包)的发送间隔时间,单位是秒
Publish Topic	网关主动发布数据的 Topic,具体格式参考下一节的定义
Command Topic	网关订阅的命令 Topic, 控制端可通过该 Topic 发布写命令
Response Topic	网关返回命令执行结果,后面小节有详细介绍
Publish Interval	网关发布数据的时间间隔, 单位秒
AUTH Enable	使能密码设置



User name	MQTT 用户名
Password	MQTT 密码

7.2.2 PUBLISH TOPIC 数据格式

MQTT Publish 文本最大支持 48K 个字符, 支持多行, 每一行以回车换行符号结束, 并且行的最大长度不超过 48K 个字符。

MQTT Publish 格式定义的核心是变量定义,请参考<u>网关内变量表达式</u>章节。而 MQTT 上传时若引用某个变量,需要在引用符号\${},比如\${VW0},表示取 16 位无符 号数。下面是一个简单的 JSON 定义格式,

{"Temperature":\${VW0},"Humidity":\${VW2}}

MQTT数据格式定义	
上传	Choose File No file chosen
查看下载	
然后洗择刚定义的文	木文件 并占击【上传】按钮 成功后重启网关 网关在发布数据

然后选择刚定义的文本文件,并点击【上传】按钮,成功后重启网关。网关在发布数据的时候就会把变量名替换成实际值进行 Publish,如下:

{"Temperature":370,"Humidity":980}

注意: 网关通过 MQTT 发布数据的时候,一个消息只对应一行,若想把所有数据放在 一个消息里面,请在一行内定义完。

7.2.3 COMMAND TOPIC 数据格式

网关同时支持两种下发指令,一种是 JSON 格式,另外一种是数学运算格式。该主题最大支持 2048 个字节大小的消息。

7.2.3.1 JSON 格式

JSON 格式以中括号开头并结尾。下发的任意 KEY 数值在用户定义的 MQTT 模板 中能找到,网关就会自动把 VALUE 写入模板配置的寄存器中。例如,模板是 { "Temperature": \${VW2000}}

如果网关在订阅的 topic 上收到消息,



{ "Temperature" : 270}

网关就会把 270 写道寄存器 VW2000 上,如果 VW2000 对应到下位机某个寄存器, 网关就会把温度值写给下位机。

7.2.3.2 数学运算格式

MQTTCommand Topic 格式跟数学运算格式一样,最大支持 2048 个字符,可以 多行,具体可参考<u>数据运算</u>章节的介绍。所不同的是,数学运算里面的目标变量是不能 被写到下位机的,而 MQTT 中的对目标变量赋值,则网关会将数据写入对应的下位机。

例子:

VW10=1234		
V0.1=1		

第一行是对第五个字(字节地址 10)赋值 1,如果第五个字有下位机对应,那么网关就会将数据写入下位机,效果等同于上位机通过 modbustcp/rtu 协议,功能码 6 或者 16 往网关地址 5 写入一个字数据 1234。

第二行是对第0个字节的 Bit 1设置1,如果有继电器对应,那么网关就会将对应的继电器打开。效果等同于上位机通过 modbustcp/rtu 协议,功能码5或者15往网关地址1写入1 (ON)。

7.2.4 RESPONSE TOPIC 数据格式

Response Topic 是在网关通过 Command topic 接收到命令后返回执行结果,这 样订阅者可以很快的查看命令执行结果,而不必等待下一次的 Publish 数据。如果 Response Topic 是空的,网关则不会返回执行结果。返回的数据格式跟 Command Topic 一样。

7.2.5 MQTT 调试

在配置好 MQTT 并重启后,可以在【系统日志】里面看到如下信息,表示 MQTT 与服务器成功连接上。



系统	志			
	ID	Time	Туре	Content
	1	0.002	INFO	MQTT: Start connecting to www.youdomain.com:1883
	2	1.014	INFO	MQTT: Connected to www.youdomain.com:1883
	3	1.035	INFO	MQTT: Connected to host www.youdomain.com
	4	1.057	INFO	MQTT: Successfully subscribed the topic
刷新	i 🦷	静除 保存		

订阅者通过 Command Topic 发布命令时,用户可以在【系统日志】里面看到接收到 消息的信息,如下图。这样方便用户调试查问题。

5 5190.024 INFO MQTT RECV: message len = 10



8 数据显示

从站数据分为实时数据和故障数据,如下图



8.1 实时数据

"实时数据"页面显示智能网关内存空间所有的数据。

Quantity:	100 •		• Ux • U	• Byte •						
Data Forma	t: Signed		•							
Address	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0	0
120	0	0	0	0	0	0	0	0	0	0
140	0	0	0	0	0	0	0	0	0	0
160	0	0	0	0	0	0	0	0	0	0
180	0	0	0	0	0	0	0	0	0	0

Address 后面的 0xxxx 表示高 5 位数, 0xxx 表示高 4 位数, 0xx 表示高 3 位数, 0x 表示高 2 位数, 最后一个表示个位数。最后面单位可以选择 Bit, Byte 或者 Word。*用户* 可以通过上述数值的选择组合成任意的位地址,字节地址,或者字地址。

Quantity 表示显示多少个数据,数据长度和格式在 Data Format 里面定义。

Data Format 有如下的选项,



Binary Signed Unsigned Hex Long AB CD Long CD AB Long BA DC Long DC BA Float AB CD Float CD AB Float BA DC Float DC BA Double AB CD EF GH Double GH EF CD AB Double BA DC FE HG Double HG FE DC BA

Binary, Signed, Unsigned, Hex 都是显示一个字(2个字节)的数据,其它的数据定 义请参考 Modbus 的字节序章节。

那么这个表怎么看?很简单,行号加列号就是单元格所对应的内存地址。例如,如果单 位选 Byte,行号 60,列号 2 的单元格对应的字节地址就是 62。如果单位选 Word, 行号 60,列号 2 的单元格对应的<mark>字地址</mark>就是 62。

8.2 故障数据

"**故障数据**"页面显示所有出故障的从站,并详细标明出故障的代码,而正常运行的从站在该页面不显示,如下图:

故障位地址	故障码
1	101
2	101
3	101

"故障位地址"的范围是 1~500, 等同于从站表中的 ID 号。关于如何访问故障位的方法,请参考网关的故障标志的地址空间章节。

"故障码"表示具体的出错原因。

异常代码	说明
1	无效的功能码



2	无效的地址空间,访问了下位机不存在的地址
3	无效的数据
101	超过最大重试次数,通常是下位机掉线导致



9 系统日志解读

系统日志在前期调试的时候非常有帮助,里面记录采集从站时的出错信息,以及主站读 取智能网关时出现的问题。

9.1 回复超时

系统日	志				
	ID	时间	类型	日志内容	
	1	1.007	WARN	Slave Index[0], Time out to get Modbus response!	
刷新	í ľ	訴 保存			

出现此告警是由于从站在规定时间内没有回复,这里 Slave Index[0]表示从站表中的索引为 0 的记录,请检查物理连接是否正常,比如 RS485 接线是否正确 (A 接 A, B 接 B),从站是否运行正常,参数是否设置正确 (从站地址是否设置正确),波特率是否匹配。

9.2 采集从站异常

系统E	志				
	ID	时间	类型	日志内容	
	1	0.056	ERROR	Slave Index[0], Exception F=0x81, code=2!	
刷新 清除 保存					

这条错误信息表示从站表中的索引为0的记录在采集数据时遇到异常,F=0x81, 表示出现异常的功能码, code=2表示异常码,这里的2表示无效的数据地址,出现 该错误请检查"数据地址"和"数量"是否超出从站的数据范围。具体代码的含义请参 考上一章的"故障数据"中的"异常代码"。

9.3 上位机主站异常

上位机在采集数据网关中数据的时候,如果地址不正确,数据网关会报错,如下图



系统日	系统日志					
	ID	时间	类型	日志内容		
	1	22.695	INFO	TCP Client connected from 192.168.1.123 : 10610		
	2	22.721	ERROR	Illegal data address F=3 A=1000, Q=3		
	3	23.729	ERROR	Illegal data address F=3 A=1000, Q=3		
	4	24.737	ERROR	Illegal data address F=3 A=1000, Q=3		
	5	25.767	ERROR	Illegal data address F=3 A=1000, Q=3		
	6	26.797	ERROR	Illegal data address F=3 A=1000, Q=3		
刷新	刷新 清除 保存					

F=3,表示功能码是03, A=1000,表示地址是1000,Q=3,表示读取的数量是3。 出现此错误的时候,请检查上位机读取的数据范围是否在"实时数据"页面范围内。



10 案例

10.1 网关和变频器与继电器模块通讯

- **目的:** 通过 Modbus 智能网关,采集变频器和继电器模块,然后上位机通过 RS232/RS485 或者 Modbus TCP 读取设备数据,并设定频率和开关继电器。
- **工具:** 一台安装 Windows 系统的电脑。

一台型号为 GY-G300 的 Modbus 智能网关 (GY-G6A/G600 都可以)

一个 USB 转 RS-232 的转换器加一个 RS-232 转 RS-485 的转接头(或者直接 USB 转 485)

Modbus Slave, Modbus Poll (<u>www.modbustools.com</u> 官网可以下载试用版,网络上也有很多下载的试用版本)

接线:将电脑串口接到 GY-G300 的 Port-1 (第一个 485 口)

10.1.1 模拟变频器和继电器模块

基于电脑 485 口(通常配一个 USB 转 485 接头),通过 Modbus Slave 调试软件, 模拟变频器和继电器。

Connection Setup	\times
Connection OK Serial Port ~ Cancel	
USB Serial Port (COM7)	
9600 Baud Mode 9600 Baud Image: State of the state	
TCP/IP Server Port IP Address Port 127.0.0.1 ✓ ✓ Any Address ● IPv4 Ignore Unit ID ○ IPv6	



模拟变频器,从站号 1

Slave Definition X
Slave ID: 1 OK Function: 03 Holding Register (4x) Cancel Address: 0 Quantity: 2 View Rows @ 10, 0, 20, 0, 50, 0, 100, 0, 55 to Supplier
Hide Alias Columns PLC Addresses (Base 1)
Error Simulation Skip response Insert CRC/LRC error (Not when using TCP/IP) [ms] Response Delay Return exception 06, Busy

模拟 8 路继电器模块, 从站号 2

Slave Definition	×
Slave ID: 2 Function: 01 Coil Status (0x) Address: 0 Quantity: 8 View Rows 10 0 20 0 50 0 11 Hide Alias Columns	OK Cancel
Error Simulation Skip response 0 [ms] Response Delay	 Insert CRC/LRC error (Not when using TCP/IP) Return exception 06, Busy

对当前频率,设定频率,以及继电器进行标注并设置初始值,如下:



Eile	Image: Slave - Mbslave2 - - × File Edit Connection Setup Display View Window Help Image:								
ID =	Mbslave1 = 1: F = 03		23		ID =	Mbslave2 : 2: F = 01			
	Alias	00000	^			Alias	00000	^	
0	current_frequency	998			0	relay_1	0		
1	target_frequency	1000			1		0		
2					2		0		
3					3		0		
4					4		0		
5					5		0		
6			~		6		0	~	
						1			
For H	For Help, press F1. Port 7: 9600-8-N-1								

10.1.2 [主站模式]下配置变频器和继电器模块

配置好 485 通讯参数。

串口设置						
串口	接口	波特率	数据位	奇偶校验	停止位	流控
Port-1	RS-485 ×	9600 ~	8 ~	None ~	1 ~	None ~
Port-2	RS-485 ~	9600 ~	8 ~	None 🖂	1 ~	None 🖂
Port-3	RS-485 ×	9600 ~	8 ~	None 🖂	1 ~	None 🖂
保存						



BRIMESH Link Everything to Internet 主站采集 系统信息 网络设置 从站 功能码 地址 数据 地址 数量 虚拟地址 重试 扫描 回复 命令 次数 周期 超时 延时 [E] [F] [G] [H] [I] [J] 接口 ID 配置 串口设置 [A] [B] [C] [D] 主站模式 🗆 所有从站 Port-1 Port-2 Port-3 以太网 添加从站 全部删除 下载从站 帮助 上传从站采集表 浏览... 未选择文件。 上传从站 从站模式 云端设置 🗹 高级配置 数学运算 □从站掉线数据置零 数据显示 ☑

添加变频器

Brimesh	Link Everything to Internet
系统信息 网络设置 串口设置	从站配置 ID: <u>1</u>
主站模式 ■ 所有从站 Port-1 Port-2 Port-3 以太网	人站地址: 1 □ 自动增加 功能码: [3] Read Holding Registers & Write Multiple Registers ▼ 数据地址: 0 □ 自动增加 数量: 2 □ 虚拟地址: 10 □ 自动增加 重试次数: 3 □ 扫描周期: 1000 ms
从站模式 云端设置 ☑ 数学运算 数据显示 ☑ 透传模式	回复超时: 1000 ms 命令延时: 0 ms 创建数量: 1 保存 帮助 返回

添加继电器模块



Brimesh	Link Everything to	Internet
系统信息	从站配置	
网络设置		一口动地加
串口设置	接口: Port-1 ×	
主站模式 ■	从站地址: 2	□自动增加
	功能码: [1] Read Coils & Write Multiple Coils	
所有从站	数据地址: 0	□自动增加
Port-1 Port-2	数量: 8	
Port-3	虚拟地址: 0	□自动增加
以太网	重试次数: 3	
	扫描周期: 1000	ms
	回复超时: 1000 会公研时: 0	ms
云端设直 ⊻	前续运动: 0 创建数量: 1	
数学运算		
数据显示 ☑	保存 帮助 返回	
透传模式		

配置完的列表如下

Brimesh	Link Everything to Internet
系统信息	主站采集
网络设置 串口设置	接口 从站功能码 数据 数量 虚拟地址 重试 扫描 回复 命令 ID 地址 地址 次数 周期 超时 延时 配置 IA1 IB1 IC1 ID1 IF1 IG1 IU1 IU1
主站模式 ■	1 Port-1 1 3 0 2 10 3 1000 1000 0 Edit Del
所有从站 Port-1 Port-2	2 Port-1 1/2 1/1 1/2 1/2 1/2 1/2 添加从站 全部删除 下载从站 帮助
Port-3 以太网	上传从站采集表
从站模式 云端设置 ☑	浏览… 未选择文件。 上传从站

重启网关



Brimesh	Link Everything to Internet	
系统信息 网络设置 串口设置 主站模式 ■	语言设置(Language Settings) 语言(Language): 中文 保存(Save)	,
所有从站 Port-1 Port-2 Port-3 以太网	重启设备 单击此按钮将重新启动网关。 重启网关	
从站模式 云端设置 ☑ 数学运算 数据显示 ☑	恢复出厂设置 单击此按钮将使串口服务器的所有设置恢复到出厂时的默认状态。 恢复出厂设置	
透传模式 系统设置 系统日志	WEB管理 WEB端口: 80 保存	•

10.1.3 [从站模式]下创建本地从站

本地从站可以给予 RS485/RS232 串口或者网口创建,方便上位机的主站读取数据。

创建一个基于 Port-2 (RS485-2) 的从站

Brimesh	Link Everything to Internet
系统信息 岡路口 小 が 留 置 置 二 人 以 環 置 二 二 人 以 環 置 二 二 、 以 環 置 二 二 、 し 環 置 二 二 、 し 環 置 二 二 、 し 、 し 環 置 二 、 、 し 、 し で し に 、 つ い ら い い い い い い い い い い い い い	本地从站配置 ID 接口 从站地址 0x位偏移 1x位偏移 3x字偏移 4x字偏移 配置 添加从站 全部删除 帮助



编辑本地从站	
ID:	1
接口:	Port-2
从站地址:	1
0x位偏移:	0
1x位偏移:	0
3x字偏移:	0
4x字偏移:	0
【保仔 】 帮助 返	

创建一个基于 Modbus TCP 的从站

编辑本地从站		
ID:	2	
接口:	Modbus TCP -	
服务端口:	502	
从站地址:	1	
Ox位偏移:	0	
1x位偏移:	0	
3x字偏移:	0	
4X字偏移:	0	
保存帮助逃		

创建完之后,在本地从站表中会显示两个从站

本地	本地从站配置								
	ID	接口	从站地址	0x位偏移	1x位偏移	3x字偏移	4x字偏移	配置	
	1	Port-2	1	0	0	0	0	<u>Edit</u> <u>Del</u>	
	2	Modbus TCP : 502	1	0	0	0	0	<u>Edit</u> <u>Del</u>	
添力	添加从站 全部删除 帮助								

重启智能网关,然后上位机就可以通过 Port-2 (中间的 485 口)或者 Modbus TCP,使用 Modbus 功能码 1,2,3,4 来读取网关虚拟地址中的数据。

10.1.4 上位机读取网关数据

BRIMESH http://www.brimesh.com

在上位机读取网关数据之前,请检查"系统日志"页面,看看是否有报错,系统日志在调试时很有帮助,详细的信息请看后面的"系统日志"介绍。

然后打开"实时数据"页面,就可以在网页上显示被采集到的数据。

Brimesh

Link Everything to Internet

系统信息 网络设置 串口设置 主站模式 ☑	实时数据 Address: Quantity: Data Format:	0xxxx × 0x 100 × Signed	xx ~ 0xx ~	0x ~ 0 ~	Word ~						
从站惧式	Address	0	1	2	3	4	5	6	7	8	9
云端设置 ☑	0	256	0	0	0	0	0	0	0	0	0
<u> </u>	10	998	2222	0	0	0	0	0	0	0	0
奴子巴异	20	0	0	0	0	0	0	0	0	0	0
数据显示 ■	30	0	0	0	0	0	0	0	0	0	0
	40	0	0	0	0	0	0	0	0	0	0
实时数据	50	0	0	0	0	0	0	0	0	0	0
故障数据	60	0	0	0	0	0	0	0	0	0	0
	70	0	0	0	0	0	0	0	0	0	0
透传模式	80	0	0	0	0	0	0	0	0	0	0
系统设置	90	0	0	0	0	0	0	0	0	0	0
系统日志	刷新										

您可以选择 Port-2 来读取数据,也可以选择 Modbus TCP 来读取数据。打开 Modbus Poll 程序,点击"Setup"菜单中的"Read/Write Definition",弹出如下对话 框

Read/Write Definition	×
Slave ID:	OK
Function: 03 Read Holding Registers (4x)	✓ Cancel
Address: 0 Protocol address. E.	g. 40011 -> 10
Quantity: 30	
Scan Rate: 1000 [ms]	Apply
Disable Read/Write Disabled Disable on error	Read/Write Once
View Rows ● 10 ○ 20 ○ 50 ○ 100 ○ Fit	to Quantity
Hide Alias Columns PLC Ad Address in Cell Enron/E	dresses (Base 1))aniel Mode

然后点击 Connection 菜单中的 Connect,



onnection Setup			- X
Connection			OK
Serial Port	•		
Serial Settings			Cancel
USB Serial Port (COM5)	•	Mode	
9600 Baud 🔻		RTU () Ascii
8 Data bite 💌		Response T	imeout
		1000	[ms]
None Parity 🔻		Delay Betwe	en Polls
1 Stop Bit 🔹	Advanced	20	[ms]
Remote Modbus Server			
IP Address or Node Name			
192.168.1.222			-
Server Port	Connect Timeout	IPv4	
500	2000 [mol		

如果要通过计算机串口通过网关的 RS485-2 来采集数据,那么在 Connection 里面选择"Serial Port",选择对应的与网关 RS485-2 相连接的计算机串口,并配置好波特率。如果要通过 Modbus TCP,那么在 Connection 里面选择 Modbus TCP/IP,配置如下图:

-	
	Cancel
-	Mode
	RTU O ASCI
	Response Timeout
	Dalau Bahuran Balli
Advanced	20 [ms]
	•
Connect Timeout	IPv4
	Advanced

这里的 Server Port 设置成 502 是由于我们在"本地从站"中设置好的,建立连接后就 会看到如下的数据,跟网关采集到的数据一致(数据可以设置十进制或者十六进制显示)。



谢립 M	월 Modbus Poll - Mbpoll1								×		
	1 is Edit Connection Setup Functions Display View Window Help 2 is ■ ● X □ 2 is ■ ● X □ 3 is 0 3 is 0 4 is 0 5 is 0 5 is 0 6 is 0 7 is 0 6 is 0 7 is 0 7 is 0 7 is 0 8 is 0 8 is 0 9 is 0										
	Mbpoll1					- • •					
Tx =	= 60: Err = 0: ID =	= 1: F = 03: SR = 1000ms									
	Alias	00000	Alias	00010	Alias	00020					
0	relay_1	0000 0001 0000 0000	current_frequency	998		0					
1		0	target_frequency	2222		0					
2		0		0		0					
3		0		0		0					
4		0		0		0					
5		0		0		0					
6		0		0		0					
7		0		0		0					
8		0		0		0					
9		0		0		0					
or H	elp, press F1.				[192.1	68.1.222]: 502					



10.2 通过 MQTT 连接阿里云物联网平台

本案例通过网关把变频器的数据和灯光控制器的状态传给阿里云,并通过阿里云远程控制继电器或者设定变频器频率。阿里云物联网平台的配置地址: https://iot.console.aliyun.com

10.2.1 阿里云配置

阿里云的帮助信息

https://help.aliyun.com/document_detail/131611.html

10.2.1.1 添加产品

[设备管理]->[产品] -> [创建产品]

* 产品名称	
GY-G300	
*所属品类 🕢	
○ 标准品类 ● 自定义品类	
* 节点类型	
● 直连设备	₩ 网关设备
连网与数据	
* 连网方式	
以太网	\sim
* 数据格式 ②	
ICA 标准数据格式(Alink JSON)	~
* 认证方式 🔞	
设备密钥	\sim



产品 (1)

创建产品快速入门] 请输入产品名称查询	Q 请送	选择产品标签 🖌	
产品名称	ProductKey	节点类型	创建时间	操作
GY-G300	a1iNf0nbsNq	网关	2020/09/30 09:07:06	查看 管理设备 删除

10.2.1.2 添加设备

[设备管理]->[设备] -> [添加设备]

添加设备 💿	\times
针别说明: DeviceName 可以为空,当为空时,阿里云会颁发全局标识符作为 DeviceName。	3唯—
产品	
GY-G300	\sim
DeviceName 🕜	
Device	
备注名称 🕜	
请输入备注名称	
備认	取消



设备									
全部产品	~		设备总数 ② 1		● 激活设备 ^② 0	● 当前在 0	线 🕜		
设备列表	批次管理								
添加设备	批量添加	DeviceName	✔ 请输入 Devi	ceName	Q	请选择设备标签	\sim		
DeviceN	ame/备注名称	设备	所属产品	节点类型	状态/启用状态 ♀		最后上线时间	操作	
Device		GY-G	5300	网关	• 未激活 🌑		-	查看	删除 子设备(0)

10.2.1.3 功能定义

← **GY-G300**

ProductKey 设备数	a1iNf0nbsNq 复制 1 前往管理				ProductSecret	******** 查看
产品信息	Topic 类列表 切前	能定义 数据解析	服务端订阅	设备开发		
编辑草稿	物模型 TSL 生成设	全备端代码				
功能类型	功能名;	称 (全部) 🖓	标识符		数据类型	数据定义

← 编辑草稿

产品名称 GY-G	300			ProductKey	a1iNf0nbsNq	复制
添加标准功能	添加自定义功能 快速导入	物模型 TSL 历	史版本 >			
 您正在编辑的是草稿 	, 需点击发布后, 物模型才会正式生实	攵。				
功能类型	功能名称(全部) 🔽	标识符		数据类型	数排	居定义

BRIMESH http://w	vww.brimesh.c	om
添加自定义功能		×
* 功能类型 👔		
属性 服务 事件		
* 功能名称 💿		
当前频率		
* 标识符 🕐		
current_frequency		
* 数据类型		
int32		\sim
* 取值范围		
0	~ 5000	
* 步长		
1		
单位		
赫兹 / Hz		\sim
*读写类型		
🔾 读写 💿 只读		



添加自定义功能	×
* 功能类型 😨	
属性 服务 事件	
* 功能名称 🕐	
设定频率	
*标识符 🕐	
target_frequency	
* 数据类型	
int32	\checkmark
* 取值范围	
0 ~ 5000	
* 步长	
1	
单位	
赫兹 / Hz	\sim
* 读写类型	
● 读写 ○ 只读	



添加自定义功能	
* 功能类型 😮	
属性 服务 事件	
* 功能名称 ②	
继电器	
*标识符 ?	
relay_1	
* 数据类型	
bool	
* 布尔值	
0- ¥	
1 - 开	
* 读写类型	
● 读写 ○ 只读	
描述	
请输入描述	

← 编辑草稿

添加協定义功能 快速导入 物模型 TSL 历史版本 ~ ① 您正在编辑的是草稿,需点击发布后,物模型 TSL 历史版本 ~ 功能差型 功能名称(全部) ▽ 标识符 数据类型 数据定义 操作 属性 当前频率 自定义 current_frequency int32 (整数型) 取值范围: 0~5000 编辑 属性 设定频率 自定义 target_frequency int32 (整数型) 取值范围: 0~5000 编辑 属性 現車備 自定义 relay_1 bool (布尔型) 0. 关 编辑	产品名称	GY-G300		ProductKey	a1iNf0nbsNq 复制	
● 您正在编辑的是草幕,素点击发布后,物模型才会正式生效. 功能类型 功能名称(全部) ♀ 标识符 数据类型 数据定义 操作 庫性 当前频率 自定义 current_frequency int32 (整数型) 取值范围: 0 ~ 5000 编辑 属性 设定频率 自定义 target_frequency int32 (整数型) 取值范围: 0 ~ 5000 编辑 属性 變电器 自定义 relay_1 bool (佈尔型) 0 · ズ 编辑	添加标准功能	添加自定义功能快速导入	物模型 TSL 历史版本 ~	,		
功能类型 功能名称(全部) □ 标识符 数据类型 数据定义 操作 属性 当前频率(自定义) current_frequency int32 (整数型) 取值范围: 0 ~ 5000 编辑 属性 设定频率(自定义) target_frequency int32 (整数型) 取值范围: 0 ~ 5000 编辑 属性 缆电器(自定义) relay_1 bool (布尔型) 0 · ズ 编辑	⑤ 您正在编辑的	的是草稿,需点击发布后,物模型才会正式生	效。			
属性 当前频率 自定义 current_frequency int32 (整数型) 取值范围: 0~5000 编辑 属性 设定频率 自定义 target_frequency int32 (整数型) 取值范围: 0~5000 编辑 属性 理电器 自定义 relay_1 bool (布尔型) 0~关 编辑	功能类型	功能名称(全部) 🖓	标识符	数据类型	数据定义	操作
属性 设定频率 target_frequency int32 (整数型) 取值范围: 0 ~ 5000 编辑 属性 運电器 自定义 relay_1 bool (布尔型) 0 - 关 编辑	属性	当前频率(自定义)	current_frequency	int32 (整数型)	取值范围: 0~5000	编辑 删除
属性 继电器 ieign 1 bool (布尔型) 0 - 关 编辑 1 - 开	属性	设定频率(自定义)	target_frequency	int32 (整数型)	取值范围: 0 ~ 5000	编辑 删除
	属性	继电器(自定义)	relay_1	bool (布尔型)	布尔值: 0-关 1-开	编辑 删除
		~~~				



彻歇两十百 / 这番目理 / 厂的	<b>ú</b> / 广山山干18				
← GY-G300					发布
ProductKey a1iNf0nbs! 设备数 1 前往管理	Ng 复制 里		ProductSecret	******** 查看	
产品信息 Topic 类列表	表 功能定义 数据解析	服务端订阅 设	备开发		
当前展示的是已发布到线」	=的功能定义,如需修改,请点击 9	扁辑草稿			
物模型 TSL 生成设备端付	代码				
功能类型	功能名称(全部) 🖓	标识符	数据类型	数据定义	操作
属性	当前频率(自定义)	current_frequency	int32 (整数型)	取值范围: 0~5000	查看
属性	设定频率(自定义)	target_frequency	int32 (整数型)	取值范围: 0~5000	查看
属性	继电器(自定义)	relay_1	bool (布尔型)	布尔值: 0 - 关 1 - 开	章看 章看

确认发布产品

 $\times$ 

您即将发布的产品为: GY-G300 产品发布后将从开发阶段进入到正式投产或使用阶段。

请勾选并确认该产品的信息和各项功能已具备发布条件:

第1步	请确认产品的各项基本信息准确无误,产品发布后将无法再做修改和删 除。	已确认 ✔
第2步	请确认设备的各项功能已经完成开发和调试,发布后产品的功能改动请通 过 OTA 升级。	已确认 ✔
第3步	请确认产品已经具备上线发布条件,开始进入规模化接入和部署。	已确认 🖌
		发布取消

BRIMESH http://www.brimesh.com

10.2.2 网关配置

10.2.2.1 模拟变频器和继电器模块

参考案例1的《模拟变频器和继电器模块》

10.2.2.2 [主站模式]下配置变频器和继电器模块

参考案例1《[主站模式]下配置变频器和继电器模块》

10.2.2.3 在阿里云物联网平台查阅账号

下面的链接是详细介绍阿里账号配置的文档,供参考。

https://help.aliyun.com/document_detail/86706.html

#### 配置步骤如下:

~	设备总数 <b>1</b>	数 🕗	<ul> <li>激活设备^②</li> <li>〇</li> </ul>	● 当前在 0	线 🕗	
批次管理						
批量添加	DeviceName 🗸	请输入 DeviceName	Q	请选择设备标签	$\checkmark$	
ne/备注名称	设备所属产品	日 节点类型	状态/启用状态 🖓	<u>5</u>	最后上线时间	操作
	GY-G300	网关	● 未激活 ●			查看 删除 子设备(0)
						$\smile$
e 未激活						
GY-G300 査	石		DeviceSecret	******** 查看	>	
a1iNf0nbsNc	复制					
Topic 列表	物模型数据 设备影	· 子 文件管理 日志	志服务 在线调试	子设备管理	分组	
GY-G3	00	ProductKey	a1iNf0nbsNq 复制	1	区域	华东2(上海)
网关		DeviceName	Device 复制		认证方式	沿东南印
	北次管理 北星添加 に/备注名称 第 用 GY-G300 査 a1iNf0nbsNg Topic 列表 GY-G30 の 素	・     役名总1       批次管理     1       批量添加     DeviceName     ・       tkala     0     ・       te/名註名称     役名所属产品       GY-G300     音看 a tiNf0nbsNq 复制       Copic 列表     物模型数据     役名最       GY-G300     音看 a tiNf0nbsNq 复制			② 祭总数 ●       ● 激活设备 ●       ● 当時在         北公管理       ●       ●       ●         北公管理       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       ●       <	○       ○       ● 当前在线 ●         1       ○       ● 当前在线 ●         北公管理        ●         北空福田       ○       ● 四市保留         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○       ●         1       ○     <



<b>设备</b>	<b>证书</b> 证书 — _{键复制}						×
Pro	ductKey	a1iNf	OnbsNq 复制	>			
Dev	viceName	Devic	e 复制	>			
Dev	viceSecret	3e7f2	02f0fcf935a51	f0028e737671	1e 复制		
烧录	方式介绍						
<b>∼</b> >	机一密、一型—	密介绍					
							关闭
密	码	生	成	I	具	链	接
https:/	/files.alicd	n.com/tr	osservice/	88413c66	e471bec8	26257781	969d1bc7.zip

https://files.alicdn.com/tpsservice/88413c66e471bec826257781969d1bc7.zip?s pm=a2c4g.11186623.2.25.3bb07908PepkPW&file=88413c66e471bec8262577 81969d1bc7.zip

## 填入设备信息:

productKey: a1iNf0nbsNq
deviceName: Device
deviceSecret: 3e7f202f0fcf935a51f0028e7376711e
timestamp:
clientId: 123456
method: hmacsha1 ~
点击这里: Generate
签名结果:

password: 5DA797D80AEDD2FCC41DD5AEC74F31C910636877



### 查看物联网云平台中 Topic 定义

← → ♂ ☆		🛈 🔒 https://iot.conso	e.aliyun.com/product	88. 🖸	90% … ☆	± II\ © ⊖ ¤ ⁵ ● ≡
📄 火狐官方站点 🥌 新	新手上路	📄 常用网址 💮 京东商城				移动设备上的书签
☰ (-) 阿里云	华东2(	(上海) 🔻	Q 搜索文档、		愛用 工単 备案 企	2业 支持 官网 🖸 🖞 🍹 🕐 简体 🍚
物联网平台公共	^	物联网平台 / 设备管理 /	产品			
概览		产品 (1)				
实例管理		创建产品 快速入门	请输入产品名称查询	Q	请选择产品标签 💙	
设备管理	^	产品名称	ProductKey	节点类型	创建时间	操作
产品		GY-G300	a1iNf0nbsNq	网关	2020/09/30 09:07:06	( 直看 )管理设备
设备						$\bigcirc$
分组						
CA 证书						
规则引擎	~					
监控运维	^					
아마ikk						Ĩ

物联网平台 / 设备管理 / 产品 / 产品详情

### ← **GY-G300**

	ProductKey a1i	Nf0nbsNq 复制		ProductSecret *******	查看			
	设备数 1 1	前往管理						
<	产品信息 Topic	类列表 功能定义 数据	设备开发					
	产品信息							
	产品名称	GY-G300	节点类型	网关设备	创建时间			
	所属品类	自定义品类	数据格式	ICA 标准数据格式 (Alink JSON)	认证方式			
	状态	● 已发布	连网协议	以太网	产品描述			



	产品信息 Topic 类列表 功能定义 数据解析 服务端订阅 设备开发											
	基础通信 Topic 物模型通信 Topic 自定义 Topic											
!	物模型通信 Topic 列表											
	功能	Topic类	操作权限	描述								
		/sys/a1iNf0nbsNq/\${deviceName}/thing/event/property/post	发布	设备属性上报								
<	周任工权	/sys/a1iNf0nbsNq/\${deviceName}/thing/event/property/post_reply	订阅	云端响应属性上报								
	属性设置	/sys/a1iNf0nbsNq/\${deviceName}/thing/service/property/set	订阅	设备属性设置								
	<b>声供</b> 」招	/sys/a1iNf0nbsNq/\${deviceName}/thing/event/\${tsl.event.identifier}/post	发布	设备事件上报								
	事件工权	/sys/a1iNf0nbsNq/\${deviceName}/thing/event/\${tsl.event.identifier}/post_reply	订阅	云端响应事件上报								
		/sys/a1iNf0nbsNq/\${deviceName}/thing/service/\${tsl.service.identifier}	订阅	设备服务调用								
	服务调用											

其中,属性上报 Topic 就是网关发布数据的主题,属性设置 Topic 就是网关订阅并接收 云端下发的配置命令主题。

#### 10.2.2.4 在网关内配置阿里云物联网平台账号以及相关 TOPIC

Brimesh	Link Ev	verything to Internet
▲ 系统信息 网络设置 串口设置 主站模式 人站模式 云端设置 ▲ Modbus MQTT 数学运算 数据显示 ■	MQTT to Cloud Enable MQTT MQTT Broker: Port: ClientID: Keep Alive(s): Publish Topic: Command Topic: Response Topic: Publish Interval(s): AUTH Enable User Name: Password:	✓         a1iNf0nbsNq.iot-as-mqtt.cn-shanghai.aliyuncs.com         1883         123456 securemode=3,signmethod=hmacsha1          60         /sys/a1iNf0nbsNq/Device/thing/event/property/post         /sys/a1iNf0nbsNq/Device/thing/service/property/set         10         ✓         Device&a1iNf0nbsNq         SDA797D80AEDD2FCC41DD5AEC74F31C910636877
^{头时数据} 故障数据 故障数据 透传模式	保存 MQTT数据格式定义	

以上的参数的来源请参考上一节。注意,这里的 Keep Alive 的时间如果小于 30 秒,阿里云会报错"Invalid ClientID",这个并不是账号配置错误。



#### 10.2.2.5 定义上报数据模板

Alink 协议是阿里云基于 json 的协议规范,详细介绍如下:

https://help.aliyun.com/document_detail/90459.html

将如下数据保存到一个文本文件中(一定要在同一行),例如 mqttdef.txt

{"params":{"current_frequency": \${VW20}, "target_frequency": \${VW22}, "relay_1": \${V0.0}}}

然后在网关的 MQTT 数据格式定义页面中选择刚保存的文件,点击上传

BRIMESH	Link Ev	erything to Internet	
系统信息 网络设置 串口设置 主站模式 ☑ 从站模式 云端设置 ■	MQTT Broker: Port: ClientID: Keep Alive(s): Publish Topic: Command Topic: Response Topic: Publish Interval(s): AUTH Enable User Name: Password:	aliNf0nbsNq.iot-as-mqtt.cn-shanghai.aliyuncs.com 1883 123456 securemode=3,signmethod=hmacsha1  60 /sys/a1iNf0nbsNq/Device/thing/event/property/post /sys/a1iNf0nbsNq/Device/thing/service/property/set 10 © Device&a1iNf0nbsNq 5DA797D80AEDD2FCC41DD5AEC74F31C910636877	~
数学运算 数据显示 ■ ^{实时数据} 故障数据 透传模式	保存 MQTT数据格式定义 上传 查看 下载	浏览 mqtt_def.txt	*

# **BRIMESH** Link Everything to Internet

系统信息 网络设置 串口设置 主站模式 ☑ 从站模式 云端设置 ■ Modbus MQTT MQTT definition file 90 bytes are successfully uploaded, please reboot the gateway.



然后在 MQTT 数据格式定义页面点击【查看】按钮, 就可以显示刚配置的数据模板。

Brimesh	Link Everything to Internet
系统信息 网络设置 串口设置 主站模式 ☑ 从站模式 云端设置	{"params":{"current_frequency": \${VW20}, "target_frequency": \${VW22}, "relay_1": \${V0.0}}}
Modbus	

#### 重启网关之后,在系统日志可以看到连接成功的消息。

## **BRIMESH** Link Everything to Internet

网络议直				
串口设置	系统日志			
主站模式 ☑	ID	Time	Туре	Content
从站模式	1	0.002	INFO	MQTT: Start connecting to a1iNf0nbsNq.iot-as-mqtt.cn-sh
	2	1.085	INFO	MQTT: Connected to a1iNf0nbsNq.iot-as-mqtt.cn-shanghai.
云 场 设 直 □	3	1.163	INFO	MQTT: Connected to host a1iNf0nbsNq.iot-as-mqtt.cn-shan
	4	1.171	INFO	MQTT: Successfully subscribed the topic
Modbus				
MQTT	刷新	清除保存		

### 这时在云端的【日志服务】里面可以看到网关发布的数据日志

☰ (-) 阿里云	华东2	(上海)	•		Q 搜索文		东和资源 费用 工单	备案 企业 支持	自网 🖸 🖞	<b>引 ⑦ 简</b> 体
物联网平台公共	-	Ì	1 小时 搜索 重置	$\sim$						
概览		ι.	时间	TraceID	消息内容	DeviceName	业务类型(全部) 🔽	操作 🕝	内容	状态 😰
实例管理			2020/10/01 12:46:37.539	0a302647160152759753 01762d14ba	查看	Device	设备到云湍息	/sys/a1iNf0nbsNq /Device/thing	{"Content":"Publish message to topic:/sys	200
设备管理 规则引擎	×		2020/10/01 12:46:37.544	0a302647160152759753 01762d14ba	直右	Device	物模型消息	/sys/a1iNf0nbsNq /Device/thing		200
监控运维	^		2020/10/01 12:46:37.543	0a302647160152759753 01762d14ba		Device	物模型	Check	-	200
实时监控		<	2020/10/01 12:46:17.536	0a302647160152757752 87608d14ba	查看	Device	设备到云涧息	/sys/a1iNf0nbsNq /Device/thing	{"Content":"Publish message to topic:/sys	200
在线调试		L 1	2020/10/01 12:45:57.537	0a302647160152755752 45521d14ba	责者	Device	物模型消息	/sys/a1iNf0nbsNq /Device/thing	-	200
设备模拟器			2020/10/01 12:45:57.537	0a302647160152755752 45521d14ba	-	Device	物模型	Check	-	200
日志服务 固件升级			2020/10/01 12:45:57.529	0a302647160152755752 45521d14ba	查看	Device	设备到云海息	/sys/a1iNf0nbsNq /Device/thing	{"Content":"Publish message to topic:/sys	200
<	>	·	2020/10/01 12:45:37.534	0a302647160152753752 23589d14ba	查看	Device	物模型消息	/sys/a1iNf0nbsNq /Device/thing		200

同时,可以在查看设备中的【物模型数据】中实时地看到当前数值



		物联网平台 / 设备管理 / 设备 / 设备详情					
概览		← Device œ					
实例管理		产品 GY-G300 音看			DeviceSecret	********	音看
设备管理	^	ProductKey a1iNf0nbsNq 复制			benebeeree		
产品		设备信息 Topic 列表 物模型数据	设备影子 文件管理	日志服务	在线调试	子设备管理	分组
<b>设备</b> 分组		运行状态 事件管理 服务调用					
CA 证书		请输入属性名称或标识符 Q					
规则引擎	~	当前频率	继电器	查看数据	设定频	率	查看数据
监控运维	^	998 Hz	0 (关) •		100	0 Hz 💿	
实时监控		2020/10/01 12:48:57.566	2020/10/01 12:48:57.566		2020/1	0/01 12:48:57.56	6
运维大盘	~						

### 10.2.2.6 下发控制协议

CA 证书		在线调试
规则引擎	$\sim$	
监控运维	^	请选择设备: GY-G300 V Device V
实时监控	- 1	调试真实设备  调试虚拟设备
运维大盘		属性调试 服务调用
在线调试		
设备模拟器		调试功能: 设定频率 (target ∨ 方法: 设置 ∨
日志服务		<pre>2 "target_frequency": 2222 3 }</pre>
固件升级		
远程配置		
告警中心		

### 通过日志服务查看刚下发的指令

### 日志服务

产品: GY-G300	$\sim$						
云端运行日志	设备本地日志 日志转	专储					
请输入 DeviceName	Q	请输入 Traceld		Q 请输入内容关键字、	Messageld	Q 全部状态	$\sim$
1 小时	$\sim$						
搜索 重置							
时间	TraceID	消息内容	DeviceName	业务类型(全部) 🖓	操作 🕜	内容	状态 🕜
2020/10/01 09:37:40.968	0be3e0be160151626090 42902e43cd		Device	云到设备消息	/sys/a1iNf0nbsNq /Device/thing	{"Content":"Publish message to topic:/sys	200





上图就是刚下发给网关的原始数据。网关收到云端发送的配置后,就会更改设定频率, 如下图



同时,我们的模拟器中的目标频率也被更改为 2222。如果是接实际的变频器,我 们就可以看到变频器中的频率改变。



23 N	Modbus Slave - Mbslave1 —											
<u>F</u> ile	ile <u>E</u> dit <u>Connection</u> <u>S</u> etup <u>D</u> isplay <u>V</u> iew <u>W</u> indow <u>H</u> elp											
D												
🗒 Mbslave1												
ID =	= 1: F = 03			ID =	2: F = 01							
	Alias	00000 ^			Alias		00000	^				
0	current_frequency	998		0	relay_1		0					
1	target_frequency	2222		1			0					
2				2			0					
3				3			0					
4				4			0					
5				5			0					
6				6			0					
		¥					1	*				
For H	elp, press F1.		F	Port	7: 9600-8-N-1							

类似,如果要控制继电器


在线调试

请选择设备:	GY-G300	$\sim$	Device	$\sim$	
调试真实试	设备 调试虚排	以设备			
属性调试	服务调用				
调试功能:	继电器 (relay_1)	~ 方	法: 设置	~	
1 * { 2 "re 3 }	lay_1": 1				
发送指令	重置				

# 点击发送指令,可以看到模拟器中的继电器状态变成1。

<u>.</u>	📓 Modbus Slave - Mbslave2 — 🗆 🖂										
<u>F</u> ile	ile <u>E</u> dit <u>C</u> onnection <u>S</u> etup <u>D</u> isplay <u>V</u> iew <u>W</u> indow <u>H</u> elp										
D											
9	Mbslave1										
ID :	ID = 1: F = 03 ID = 2: F = 01										
										_	
	Alias	00000	^				Alias		00000	^	
0	current_frequency	998				0	relay_1		1		
1	target_frequency	2222				1			0		
2						2			0		
3						3			0		
4						4			0	- 1	
5						5			0		
6			~			6			0	~	
		1			Ľ					-	-
For H	eln press F1				P	ort	7: 9600-8-N-1				



## 10.3 通过 KEPSERVERMQTT 远程监控网关

KEPServer 的 mqtt 功能需要 mqtt broker 配合,因此先安装 mqtt broker 软件。

10.3.1 网关配置

10.3.1.1 模拟变频器和继电器模块

参考案例1的《模拟变频器和继电器模块》

10.3.1.2 [主站模式]下配置变频器和继电器模块

### 参考案例1《[主站模式]下配置变频器和继电器模块》

Link Everything to Internet
主站采集
接口         从站功能码         数据         数量/虚拟地址         重试         扫描         回复         命令           ID         地址         地址         次数         周期         超时         延时         配置           [A]         [B]         [C]         [D]         [E]         [F]         [G]         [H]         [J]
1         Port-1         1         3         0         2         10         3         1000         100         0         Edit Del           2         Port-1         2         1         0         8         0         3         1000         1000         0         Edit Del
添加从站 全部删除 下载从站 帮助
上传从站采集表 》

## 10.3.1.3 MQTT 配置



MQTT to Cloud	
Enable MQTT	
MQTT Broker:	www.brimesh.com
Port:	1883
ClientID:	2134safd
Keep Alive(s):	10
Publish Topic:	/mqtt_rx_data_channel/Device1
Command Topic:	/mqtt_rx_cmd_channel/Device1
Response Topic:	
Publish Interval(s):	10
AUTH Enable	$\checkmark$
User Name:	test
Password:	test
保存	



## 10.3.1.4 定义上报数据模板

将如下数据保存到一个文本文件中(一定要在同一行),例如 mqttdef.txt

{"current_frequency": \${VW20}, "target_frequency": \${VW22}, "relay_1": \${V0.0}, "mqtt_rx_cmd_channel.Device1.target_frequency" :\${VW22}, "mqtt_rx _cmd_channel.Device1.relay_1" :\${V0.0}}

然后在网关的 MQTT 数据格式定义页面中选择刚保存的文件,点击上传。文本中的两个变量。mqtt_rx_cmd_channel.Device1.target_frequency 和 mqtt_rx_cmd_channel.Device1.relay_1 是 KEPServer 的下发控制客户配置变量。

10.3.2 KEPSERVER 配置

10.3.2.1 远程接收网关数据的配置

10.3.2.1.1创建 MQTT_RX_DATA_CHANNEL

创建基于 mqtt client driver 的 channel, 该通道用于接收网关发送上来的数据。



🙉 KEPServerEX 6 Cor	figuration [Con	nected to Run	time]						
File Edit View Too	ols Runtime	Help							
🗋 💕 🗟 🛃 🍀	🖬 🖾 🐿 🛛		🗈 🗈 🗙 🛄						
Project		Device Name	٢	Model		ID		Descriptio	on
-("") Connectivity 	d_channel 1 <b>lata_channel</b> 1	Device 1		MQTT					
Advanced 1	Property Ed	itor - mqtt_rx_	data_channel						
Add Area	Property Groups								
🗄 🐸 Data Logger	General		Host			www.bnmesh.com			
EFM Exporte	Write Optimization	ns				Disable			
E-S IDE for Solur	Advanced					Disable			
Add Splu	MQTT Server		Client ID			IF3KKpUT6xMLITe1	NXHrYN7		
🖨 🔆 loT Gateway	Timing		Usemame			test			
mqtt_tx_			Password						
E Local Historia			Client Certificate			Disable			
Add Data			Optional Configure	ation					
E Scheduler			MQTT Server Versio	n		Auto			
SNMP Agent			Subscription QoS			0 (At most once)			
Add Age									
Date 🗸 Tim									*
(1) 2020/11/14 21:			Defaults		ОК	Cancel	Apply	Help	
1 2020/11/14 21:									
(1) 2020/11/14 21:33	:21 KEPServe	erEX\R Local	Historian Plug-in V6.5.829	0.0					
2020/11/14 21:33	:21 KEPServe	erEX\R IDF fo	r Splunk V6.5.829.0						_
0 2020/11/14 21:33	21 KEPServe	Sched	Juler Plug-in V6.5.829.0						
2020/11/14 21:33	21 KEPServe	BREANR IOT G	ateway V6.5.829.0						· ·
Part								Defect User Class	
кеаду								Derauit Üser Clien	ts: U Active tags: 0 of 0
			<b>B</b> :N_				Сн 📻 🥊	) 🖣 🔺 😼 📴 🕯	22:14 2020/11/14



# 10.3.2.1.2创建 DEVICE1

KEPServerEX 6 Cor	nfiguration [Con	nected to Run	time]				
File Edit View Too	File Edit View Tools Runtime Help						
D 📂 🗃 🛃 🕄	e 💷 🖾 🐑 e	- 	🤊 👗 🗈 🛝 🗙 🔛				
E B Project	roject Tan Name / Address Data Tyne Scan Bate						
🖃 📳 Connectivity		aureat frea	want fearurant (anti-abased/DeviceTearurant fearurant fearur				
· · · · mqtt_rx_cm	nd_channel	relay 1	/mgtt_rx_data_channel/Device1+relay_1	Boolean	100	None	
Device	el In alternat	anget frequ	ency /matt rx data channel/Device1+target freque	n Word	100	None	
	ta_cnannei						
Aliases						-	
Advanced T	📴 Property Ed	itor - mgtt_rx_d	data_channel.Device1		×		
🗎 😩 Alams & Eve						1	
Add Area	Property Groups		Name	Device1			
EFM Exporte	General		Description				
Add Poll	Tag Gaparation		Driver	MQTT Client			
DF for Splur	Tag Generation		Model	MQTT			
- 📎 Add Splu			Channel Assignment	mqtt_rx_data_channel			
□ ╬╬ IoT Gateway			Operating Mode				
I local Historia			Data Collection	Enable			
Add Data			Simulated	NO			
Scheduler							
Add Sch							
E-@ SNMP Agent							
Add Age							
						•	
Date 🗸 Tim							
(1) 2020/11/14 21:					Analy Units		
1 2020/11/14 21:			Derduits	Cancer	Арру		
1 2020/11/14 21:33	21 KEPServe	erEX\R Local	Historian Plug-in V6.5.829.0			-	
1 2020/11/14 21:33	:21 KEPServe	erEX\R IDF fo	r Splunk V6.5.829.0				
(i) 2020/11/14 21:33	:21 KEPServe	erEX\R Sched	Juler Plug-in V6.5.829.0				
(1) 2020/11/14 21:33	21 KEPServe	erEX\R loTGa	ateway V6.5.829.0				
			m				
Ready	-				Default User (	Jients: U Active tags: 0 of 0	
(2)			C+1		сн 🧰 😨 🍷 🔺 😼 🕻	22:15 2020/11/14	



## 10.3.2.1.3创建变量

分别创建 current_frequency, target_frequency, relay_1, 注意这几个 tag 的名字必须跟网关中的 mqtt 上报格式中的名称要一致。另外, 图中的 Address 是通 过 topic+tag 的格式。

KEPServerEX 6 Configuration [Configuration]	nnected to Runtime]				
File Edit View Tools Runtime	Help				
🗅 💕 🗟 🛃 🏶 🛅 🖏 🐿	🤬 🔂 🚰 🖌 🥱 🕻	🗅 🛝 🗙   🔛			
Project	Tag Name /	Address	Data Type	Scan Rate	Scaling
⊡ () Connectivity	current_frequency	/mgtt_rx_data_channel/Device1+current_frequency	Word	100	None
Device1	🐨 relay_1	/mqtt_rx_data_channel/Device1+relay_1	Boolean	100	None
mqtt_rx_data_channel     Device1	Carget_frequency	/mqtt_rx_data_channel/Device1+target_frequency	Word	100	None
Advanced Tags					
Add Area 					
Group EFM Exporter     Add Poll Group					
Add Splunk Connection					
mqtt_tx_cmd_channel					
Add Datastore					
Add Schedule					
	•	m			۱.
Date V Time Source	Event				*
i) 2020/11/14 21:33:21 KEPServ	erEX\R Connection Shari	ng Plug-in V6.5.829.0			
i 2020/11/14 21:33:21 KEPServ	erEX\R Security Policies	Plug-in V6.5.829.0			
1 2020/11/14 21:33:21 KEPServ	erEX\R Local Historian Plug-in V6.5.829.0				
(i) 2020/11/14 21:33:21 KEPServ	erEX\R IDF for Splunk V	6.5.829.0			
U 2020/11/14 21:33:21 KEPServ	erEX\R Scheduler Plug-ir	V6.5.829.0			
1 2020/11/14 21:33:21 REPServ	erex\R Io I Gateway V6.	5.623.0			· ·
Ready				Default User Cliente	s: 0 Active tags: 0 of 0
					22:16
				= 🐨 🗸 🍐 😽 🗒	2020/11/14

10.3.2.1.4运行 OPC QUICK CLIENT 查看接收到的数据

在网关运行并上传数据后,可以看到三个变量都有数据上来。



KEDSenverEX 6 Configuration (Con	nnected to Runtimel			
File Edit View Tools Buntime	Help			
- a mar war u u u	Tag Name / Address	Data Type	Scan Rate	Scaling
一 ()) Connectivity	durrent_frequency /mqtt_rx_data_channel/Device	e1+current_frequency Word	100	None
Device1	☑ relay_1 /mqtt_rx_data_channel/Device	e1+relay_1 Boolean	100	None
- • mqtt_rx_data_channel	Maget_frequency /mqtt_rx_data_channel/Device	e1+target_frequency Word	100	None
Device I				
💽 OPC Quick Client - 无标题 *				
File Edit View Tools Help				
🗋 🗅 🚅 🔛 📸 💣 💕 👗 🖻	a 🖻 🗙			
loT_Gateway.mqtt_tx_cmd_ch	nannel 🔺 Item ID	Data Type 🛆 Value	Timestamp	Quality
System	<pre>@mqtt_rx_data_channel.Device1.relay_1</pre>	Boolean 1	22:28:39.929	Good
	mqtt_rx_data_channel.Device1.current_freq.	Word 3	22:24:49.883	Good
mqtt_rx_cmd_channelSystem	Image: Comparison of the second se	Word 4	22:24:59.898	Good
mqtt_rx_cmd_channel.Device1				
mqtt_rx_cmd_channel.Device1	System ⊨			
mgtt nx data channel. System				
mqtt_rx_data_channel.Dev	vice1			
mqtt_rx_data_channel.Device1	System			
۰ III.	• •	III		F.
Date Time	Event			*
0 2020/11/14 22:24:17	Added group 'mqtt_rx_data_channel.Device1' to 'Kepy	ware.KEPServerEX.V6'.		
0 2020/11/14 22:24:17	Added 11 items to group 'mqtt_rx_data_channel.Devic	e1System'.		
2020/11/14 22:24:17	Added group 'mqtt_rx_data_channelStatistics' to 'Ke	pware.KEPServerEX.V6'.		
2020/11/14 22:24:17 2020/11/14 22:24:17	Added 3 items to group matt_rx_data_channel.Device Added group imatt_px_data_channel_System to 'Keny	: I . ware KEPServerEX V6'		- P
0 2020/11/14 22:24:17	Added 12 items to group 'mqtt_rx_data_channelStati	stics'.		E
1 2020/11/14 22:24:17	Added 4 items to group 'mqtt_rx_data_channelSyste	m'.		*
Ready			Ī	tem Count: 113 //
0 2020/11/14 21:48:10 KEPServ	PREATR Moddus ICP/IP Ethemet device anvertoaded suc	cessfully.		
2020/11/14 21:49:08 Modbus	TCP/IP Ethemet Manager started.	'hannel - watt av and abannel' Sa	himseb.com/1002 MOTTV-	minn = 211
(i) 2020/11/14 22:24:16 MQTTCI	ient MQTT version automatically selected by server. [C lient MQTT version automatically selected by server.] C	hannel = 'mgtt_rx_cmo_channel', server = 'www 'hannel = 'mgtt_rx_data_channel' Server = 'www	brimesh.com:1663, MQTT Ve	rsion = 3.1.1.
<				•
Ready			Default User Clients: 1	Active tags: 113 of 113
👌 🏉 🚞 💿	3 📴 🔛			22:30 2020/11/14

## 10.3.2.2 控制网关的配置

KEPServer 往网关下发控制数据是通过 IoT Gateway 来实现的,官方有说明文档,可以参考:

https://www.kepware.com/getattachment/96bdb7bb-4f9a-4cfe-be30-ef048d1 6dd83/iot-gateway-manual.pdf

参考步骤如下:

# 10.3.2.2.1创建 MQTT_RX_CMD_CHANNEL

创建用于接收控制命令的通道,用户也可以选择 KEPServer 其它的任何通道创建数据,这里仅仅是为了演示,然后在 KEPServer 选择周期性的或者数据有变化时发送控制指令。

# 10.3.2.2.2创建 MQTT_TX_CMD_CHANNEL

在 IoT Gateway 下面创建发送指令的通道,特别留意发布的 Topic,网关会订阅这个 Topic,用于接收控制命令 (在 cmd topic 填入):

KEPServerEX 6 Configuration [Configuration]	nnected to Runtime]					
File Edit View Tools Runtime	Help					
🗋 💕 🗟 📓 📚 🛍 🗟 🕥	🖲 🚰   🤊 👗 🖬 🛝 🗡					
Project	Server Tag	L	State	Data Type	Scan Rate (ms)	Send Every Sc
□-(iii) Connectivity	mgtt rx cmd channel.Device	1.relay 1	Enabled	Boolean	1000	False
	mqtt_rx_cmd_channel.Device	1.target_frequency	Enabled	Word	1000	False
Aliases						
Advanced Tags	Dranarty Editor JaT Gata	way matt to and channel			_	
Alams & Events	Property Editor - 101 Gate	way.mqtt_tx_cmd_cnanner				
Add Area	Property Groups	MQTT Broker				_
	General	URL		tcp://www.brimesh.com:18	383	_
Add Poll Group	Client			iotgateway		_
DF for Splunk	Message			1 (At least ence)		_
Add Splunk Connection	Security	Bate (me)		10000		
🖶 🚟 loT Gateway	Last Will	Format		Narrow Format		
mqtt_tx_cmd_channel	Subscriptions	Max Events per Publish		1000		_
Local Historian	Licensing	Transaction timeout in (s)		5		
Add Schedule						
SNMP Agent						
Add Agent						
						_
		Topic				+
Data Timo Source		Specify the name used to filte	r or organize data published (	on the broker.		
() 2020 (11 (14 21 40 10 KEDC)						
(1) 2020/11/14 21:48:10 KEPSe (1) 2020/11/14 21:49:09 Modbu		Defaults	ОК	Cancel	Apply Help	
1 2020/11/14 22:24:18 MOTT						
(i) 2020/11/14 22:24:18 MQTT CI	ient MQTT version automatic	cally selected by server.   Chann	el = 'matt nx data channel',	Server = 'www.brimesh.com	:1883', MQTT Version = 3.1.1	
1 2020/11/14 22:32:39 MQTT CI	ient MQTT version automatic	cally selected by server.   Chann	el = 'mqtt_rx_cmd_channel',	Server = 'www.brimesh.com	1883', MQTT Version = 3.1.1	
i) 2020/11/14 22:32:45 KEPServ	erEX\R MQTT agent 'mqtt_tx_cr	md_channel' is connected to br	oker tcp://www.brimesh.com	n:1883'		-
•		III				•
Ready				D	efault User Clients: 0 Activ	re tags: 2 of 2
🕘 🏉 📜 🚳				сн 🚎 🕐 🛱	- 😼 🛱 🖬 🕪 2	22:36 020/11/14

修改 message 的模板格式, 记得 Message Format 要选 Advanced Template



KEPServerEX 6 Configuration	n [Connected to Runtime]					
File Edit View Tools Runti	ime Help					
- 🗋 🐸 🗟 🛃 📚 🙆 🚭	) 🖲 🖀 💆 🎽 🗈 🖎	× 🗠				
Project	Server Tag	/	State	Data Type	Scan Rate (ms)	Send Every Sc
□-(a) Connectivity	<pre>e_mqtt_rx_cmd_channel.Devic</pre>	e1.relay_1	Enabled	Boolean	1000	False
→ mqt_rx_data_channel → mqt_rx_data_channel → ≫ Aliases	Communication and the second s	e1.target_frequency	Enabled	Word	1000	False
Advanced Tags	💷 Property Editor - IoT Gat	eway.mqtt_tx_cmd_chann	el			×
Add Area	Property Groups	Message				
Data Logger	General	Message Format		Advanced Template		
Add Poll Group	Client	Template		{ "values": [  #eacl	NALUES { "ITAGNAMEI":	<u>IVA</u>
DF for Splunk	Message					
Add Splunk Connection	n Security					
는 괮북 loT Gateway	Last Will					
Incal Historian	Licensing					
Add Datastore	Donnarg					
Scheduler						
Add Schedule						
Add Agent						
		Template				
		Specify a template for pushi	ng data. This template	e engine uses a subset of Hand	lebars.	
Date / Time So	urce					
U 2020/11/14 21:48:10 KE	PSe	Defaults OK Cancel Apply Help				
1 2020/11/14 21:49:08 Mo						
(i) 2020/11/14 22:24:18 MC	T Client most i version automatically selected by server. [Chamiler - impaindinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderinderin					
1 2020/11/14 22:32:39 MC	MQTT Client MQTT version automatically selected by server. [Channel = 'mqtt.rx.cmd_channel', Server = 'www.brimesh.com: 1883; MQTT Version = 3.1.1.					
(i) 2020/11/14 22:32:45 KE	22:32:45 KEPServerEX\R MQTT agent 'mgtt_tx_cmd_channel' is connected to broker tcp://www.brimesh.com:1883'					
•		m				
Ready					Default User Clients: 0	扬声器: 67% 2
	ex 🔚 🔤			сн 🚎 (	) 🖡 🔺 😽 🗑 🖿 🔶	22:37

Template 的模板如下:

```
{
|#each VALUES|
"|TAGNAME|": |VALUE|,
|/each|
```

}

该模板会把所有下发控制的变量遍历一遍生成 json 并 publish 给 iotgateway topic。 网关通过订阅 iotgateway 接收控制指令。关于模板的具体介绍,请看 KEPServer 官方 文档 iot-gateway-manual.pdf 中《Advanced Template Format》介绍,摘取如下



#### **Advanced Template Data Format**

The IoT Gateway pushes data in a standard JSON format via the REST and MQTT Clients. This format may then be consumed by the third party endpoint and broken down in an appropriate way. The REST Client agent and MQTT agent have the ability to use an advanced template for pushing data. This template engine uses a subset of Handlebars. The Advanced Template allows more complete control over the payload format. In addition to JSON, formats like XML and CSV can be generated using this template. The Advanced Template is a drop down selection in the Message Format field.

#### Variables

Like the Standard Template, variables can be inserted into the template representing publish time and data changes. Top-level variables like SERVERDATE and SERVERTIMESTAMP can be inserted anywhere in the tem plate.

SERVERTIMESTAMP	Fime of publish, represented as the number of milliseconds since January 1st,		
	1970, midnight		
SERVERDATE	Date and time of publish as human-readable string		

The VALUES variable represents a list of every data change in the publish. Each data change contains variables for the tag name, value, quality, and timestamp.

TAGNAME	The name of the tag ('Channel.Device.Tag')
VALUE	The value of the tag
QUALITY	"true" if the tag was read successfully, "false" if the tag could not be read (e.g. a con- nection issue)
TIMESTAMP	The time at which the tag was read, represented as the number of milliseconds since January 1st, 1970, midnight.

#### Syntax

The 'each' keyword allows text to be generated for each item update. The template inside the 'each' block is evaluated once for every item update in the publish. Depending on the format, this can be used to make each update a JSON object or a CSV row for example.

```
|#each VALUES|
|TAGNAME|, |VALUE|, |QUALITY|, |TIMESTAMP|,
|/each|
```

#### **Additional Syntax**

The default advanced template contains the following syntax:

|#unless @last|,|/unless|

This can be read as: "Unless this is the last item in the list, insert a comma." This effectively eliminates the trailing comma for the last item in a list. While this can be omitted in most cases, it is necessary if the consumer of the payload treats trailing commas as a syntax error.



# 10.3.2.2.3创建 SERVER TAG

Server tag 是在已经创建的 tag 里面选择,然后 KEPServer 的 loT Gateway 模块 会根据设置选择周期性的或者数据有变化时把控制数据发布给网关。



在 mqtt_rx_cmd_channel 通道里面选择需要控制的变量,如下图:



	6 Configure	ation (Conno	acted to Puntimal				
Ella Edit View	Teels	auon (conne Puntimo - Ha	ala				
File Edit View							
	2   X> 🛀		) 🐨   🤊 🔉 🖬 🖾 🗙   📟				
Project		9	Server Tag	/ State	Data Type	Scan Rate (ms)	Send Every Scan
😑 (🎁 Connec	tivity	. 6	mott rx cmd channel Device1 relay 1	Enabled	Boolean	1000	False
	t_rx_cmd_cha	nnel	mgtt_rx_cmd_channel.Device1.target_freque	ncy Enabled	Word	1000	False
	Device i t rx data cha	mal					
	Device1						
Advance Advance	ed Tags						
Alams	& Events						
Add	d Area						
	ogger morter						
Add	Poll Group						
B-B IDF for	Splunk						
- 🔁 Ado	Splunk Conne	ection					
😑 👬 🔤 Gat	eway						
mq	tt_tx_cmd_c	hannel					
	listonan I Datastera						
	ler						
Add	d Schedule						
	Agent						
Add	d Agent						
			< [				4
-	-						
Date \	lime	Source	Event				^
2020/11/14	23:45:48	MQTT Client	t Unexpected character found while pars	ng MQTT payload.   Char	nnel = 'mqtt_rx_data_channel', Topi	c = '/mqtt_rx_data_channel/De	evice1', Line = 1, Col = :
2020/11/14	23:46:20	KEPServerE	X\R MQTT agent 'mqtt_tx_cmd_channel' dis	connected - reason 'Rea	d timed out	10001	
2020/11/14	23:46:45	MQTT Client	t Connection with MQ I I server lost.   Cha	annel = 'mqtt_rx_cmd_cha	annel', Server = 'www.brimesh.com:'	1883'.	
2020/11/14	23:46:40	KERSeeverE	t Connection with MQTT server lost. [Chi	annel = mqtt_rx_data_cna	annel, Server = www.bnmesn.com:	1883.	
2020/11/14	23.47.03	MOTT Client	t Eailed to connect to MOTT server LCh:	ed to connect - reason. v	www.bhinesri.com . appel' Server - 'www.brimesh.com :	1993	
2020/11/14	23.49.29	MOTT Client	t Failed to connect to MQTT server. [Chi	annel = 'matt_rx_cinta_cha	annel' Server = www.brimesh.com:	1883'	
1 2020/11/15	8:45:48	MQTT Client	t Failed to connect to MQTT server. I Cha	annel = 'mgtt nx data cha	annel', Server = 'www.brimesh.com:	1883', Reason = 'TCP connect	timeout'.
1 2020/11/15	8:45:50	MQTT Client	t Failed to connect to MQTT server.   Cha	annel = 'mqtt_rx_cmd_cha	annel', Server = 'www.brimesh.com:	1883', Reason = 'TCP connect	timeout'.
4							
Beady						Default User Cliente: 1	Active tags: 113 of 113
						Conduit Coor Circlitts. 1	0.00
		ex				🏛 🕐 🏹 🔺 🍡 🕼 🕻 🕯	2020/11/15

# 10.3.2.2.4发送控制指令

# 使用 mqttfx 往 mqtt_rx_cmd_channel 主题发送控制命令,如下图

🌚 MQTT.fx - 1.7.1							
File Extras Help							
www.brimesh.com			Connect	Disconnect			
Publish Subscribe	Scripts	Broker Status	Log				
Switch Fountain ON		» /mqtt_rx_cmd	_channel/Device1		•	Publish	
Switch Fountain OFF	C 🖬	{"relay_1" : false, "t	arget_frequency": 88}				

同时订阅 iotgateway 主题,就可以看到 KEPServer 发送下来的控制指令。



🌚 MQTT.fx - 1.7.1	- C	$\times$
File Extras Help		
www.brimesh.com	Connect Disconnect	
Publish Subscribe	Scripts Broker Status Log	
iotgateway	Subscribe QoS 0 QoS 1 QoS 2 Autoscroll	0;▼
iotgateway Dump Messages	iotgateway	26 QoS 0
	iotgateway	36 QoS 0
	iotgateway	24
	15-11-2020 10:54:10.39250571	QoS 0
	<pre>{     "mqtt_rx_cmd_channel.Device1.target_frequency": 88,     "mqtt_rx_cmd_channel.Device1.relay_1": false, }</pre>	

# 在网关的系统日志里面可以看到下发的数据,如下图:

ID	Time	Туре	Content	
1	0.002	INFO	MQTT: Start connecting to www.brimesh.com:1883	
2	4.004	ERROR	MQTT: Failed to connected to www.brimesh.com:1883	
3	9.005	INFO	MQTT: Start connecting to www.brimesh.com:1883	
4	9.146	INFO	MQTT: Connected to www.brimesh.com:1883	
5	9.169	INFO	MQTT: Connected to host www.brimesh.com	
6	9.191	INFO	MQTT: Successfully subscribed the topic	
7	474.977	INFO	MQTT RECV: message len = 245	
8	474.977	INFO	MQTT: VW22=600	
9	474.977	INFO	MQTT: V0.0=1	
10	474.978	INFO	MQTT: VW22=600	
11	474.978	INFO	MQTT: V0.0=1	
12	604.839	INFO	MQTT RECV: message len = 127	
13	604.839	INFO	MQTT: VW22=88	
14	604.839	INFO	MQTT: V0.0=0	

# 同时,在【实时数据】页面可以查看到写下来的数值



Address: Quantity: Data Format:	0xxxx ~ 0x 100 ~ Signed	xx ~ 0xx ~	0x ~ 0 ~	Word \vee						
Address	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
10	0	88	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0
70	0	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0	0



## 10.4 通过 THINGSBOARD 远程监控网关

ThingsBoard 是一个开源的物联网平台,用于数据收集、处理、可视化展示以及 设备管理。ThingsBoard 使用行业标准物联网协议 (MQTT, CoAP 和 HTTP) 实现设 备连接,并支持云和本地部署。

ThingsBoard 结合网关可以做什么? Thingsboard 配合网关可以实现将现场采集的的设备数据储存到云端数据库中,并且提供多种部件库,用于展现当前以及历史数据,以及远程控制设备的界面。同时对于高级用户,可以调用 ThingsBoard 提供的 restful 接口来自己开发前端界面。

本例子使用 ThingsBoard,通过 mqtt 协议收集网关上报的数据,并存储到 pgsql 数据库中,同时通过调试工具 curl 远程调用 ThingsBoard 的 restful 接口,以此达到 远程控制网关下的 Modbus RTU 设备。

## 10.4.1 THINGSBOARD 配置

#### 10.4.1.1 THINGSBOARD 安装

Thingsboard 官方有很详细的安装说明文档,这里以 Windows 平台作为测试,具体文档请参考如下链接,一步步跟着操作就可以完成安装

https://thingsboard.io/docs/user-guide/install/windows/

需要注意,安装 pgsql 的时候,语言选中文可能会有安装问题,如果是这样请选择 English语言。

Thinsboard 的 github 库链接:

https://github.com/thingsboard/thingsboard

Thinsboard 的正式发布版本链接 (注意, v3.1.1 接收 mqtt 有问题, 请使用 v2.5.4):

https://github.com/thingsboard/thingsboard/releases

在安装完之后,登录 http://localhost:8080

BRIMESH http://www.brimesh.com

以默认管理员账户 sysadmin@thingsboard.org / sysadmin 登录并创建一个租户账 号, 用于创建设备。

## 10.4.1.2 创建设备

← → C û 0 <u>%</u> 192	2.168.124.202:8080/de	vices ﷺ ···· ☆ 보 II\ 🗉 Θ ≒ ↔ ● Ξ
	□ 设备	
▲ 首页		G300
<ul> <li>・・・&gt; 規则链库</li> <li>・・・&gt; 客户</li> </ul>	G300 MODBUS GATE	
■ 资产 □□ 设备		<     详细信息     属性     最新遥测     警告     事件     关联     >       公开     分配给客户     管理凭握     删除设备     ^
₩ 实体视图		€ 复制设备ID € 复制访问令牌
● 部件库 ● 《表板库		名称* G300
<b>⑥</b> 审计日志		设备类型* Modbus Gateway X
		Label

# 10.4.1.3 THINGSBOARD 的 MQTT 通讯

ThingsBoard 的 Mqtt 通讯在官网中有很详细的说明, 链接如下:

https://thingsboard.io/docs/reference/mqtt-api/

网关通过 v1/devices/me/telemetry 这个 Topic,以 json 的格式来上报数据,通 过 v1/devices/me/rpc/request/+这个 Topic 来下发控制指令。也就是说,网关订阅 v1/devices/me/rpc/request/+来接收控制指令,这里符号+是一个通配符,网关实际 收到的数据来自于由 v1/devices/me/rpc/request/和一个 id 号组成的 Topic,例如, 第一个指令来自 Topic v1/devices/me/rpc/request/1,第二个指令就是来自 v1/devices/me/rpc/request/2,以此类推。而指令回复是通过 v1/devices/me/rpc/response/\$request_id 这个 Topic 发送的,也就是这个 Topic 由 v1/devices/me/rpc/response/和指令 Topic 后面的 id 组成,成双成对的,例如:



指令 Topic: v1/devices/me/rpc/request/888

回复Topic: v1/devices/me/rpc/response/888

10.4.2 网关配置

10.4.2.1 模拟变频器和继电器模块

参考案例1的《模拟变频器和继电器模块》

10.4.2.2 [主站模式]下配置变频器和继电器模块

参考案例1《[主站模式]下配置变频器和继电器模块》

Brimesh	Link Everything to Internet
系统信息	主站采集
网络设置 串口设置	接口         从站功能码         数据         数量         虚拟地址         面試         扫描         回复         命令           ID         地址         地址         次数         周期         超时         延时         配置           IA)         IB)         ICI         ID         IF1         IG1         IH1         IJ1
主站模式 ■ ^{所有从站}	1         Port-1         1         3         0         2         10         3         1000         100         Edit         Del           2         Port-1         2         1         0         8         0         3         1000         1000         0         Edit         Del           2         Port-1         2         1         0         8         0         3         1000         0         Edit         Del
Port-1 Port-2 Port-3	添加从站 全部删除 下载从站 帮助
从站模式	上传从站采集表 

10.4.2.3 MQTT 配置



MQTT to Cloud	
Enable MQTT	
MQTT Broker:	192.168.1.3
Port:	1883
ClientID:	2134safd
Keep Alive(s):	10
Publish Topic:	v1/devices/me/telemetry
Command Topic:	v1/devices/me/rpc/request/+
Response Topic:	v1/devices/me/rpc/response/
Publish Interval(s):	10
AUTH Enable	
User Name:	GvbTKDtWj1yeHJEvp699
Password:	
保存	

图中的用户名是通过 ThingsBoard 页面中设备的【复制访问令牌】得到,唯一对应于创建的每个设备,密码为空。ClientID 用户自己设置,可以用序列号,这样保证不重复。

10.4.2.4 定义上报数据模板

将如下数据保存到一个文本文件中(一定要在同一行),例如 mqttdef.txt

{"current_frequency": \${VW20}, "target_frequency": \${VW22}, "relay_1": \${V0.0}}

然后在网关的 MQTT 数据格式定义页面中选择刚保存的文件,点击上传。

10.4.3 查看变频器当前频率的历史曲线

在设备的【最新遥测】页面可以看到当前设备的最新数据,如下图:

BRIMESH http://www.brimesh.com
--------------------------------

G300 设备详细信息			0 ×
详细信息 属性 最新遥测	警告 事件 🗦	关联 审计日志	
2020-10-29 07:35:23	ааа	111	
2020-10-29 07:35:23	bbb	222	
2020-10-29 07:35:23	ccc	333	
2020-11-15 16:17:57	current_frequency	6	
2020-11-15 16:17:57	relay_1	1	
2020-11-15 16:17:57	target_frequency	8	
2020-11-02 15:25:29	Тетр	0	
	Page: 1 🔻 R	Rows per page: 10 👻 1 - 7 of 7	~ < >

# 同时,也可以对当前数据进行组态,查看历史曲线,例如选 current_frequency,

详细信息	属性	最新遥测	截生言口	事件	关联	审计日志	
1 遥测 着	波选中						★ 在部件上显示
L 最后	更新时间		键个			值	
202	0-10-29 07:35:	23	aaa			111	
202	0-10-29 07:35::	23	bbb			222	
202	0-10-29 07:35:	23	ccc			333	
<b>~</b> 202	0-11-15 16:20:	17	current_free	quency		6	
202	0-11-15 16:20:	17	relay_1			1	

点击右上角【在部件上显示】,显示 Charts 部件,

Br	iME	<b>SH</b> http	://www.b	rimesh.c	om			
详细信息	属性	最新遥测	<u>較</u> 生 言口	事件	关联	审计日志		
当前包	Char	ts				系统 →	添加到仪表板	×
<	Timeser 6 6	ries - Flot	•					>
将部件	牛添加	到仪表板	Ī			×		
0	选择现	<b>月仪表板</b> 仪表板						
۲	<b>创建新</b> 新仪表板 currer	的仪表板 ^{页标题*} nt_fequenc	у					
l	✔ 打开	刊仪表板	添	加	取消			

然后在仪表盘库里面就可以看到当前频率的历史曲线。

BRIMES	http://www.brimesh	i.com		
👸 ThingsBoard	■ 仪表板库 > 📲 curr	ent_fequency		
<b>俞</b> 首页	current_fequency		current_fequency	G300
<b>◇·&gt;</b> 规则链库				
24 客户	Timeseries - Flot	0		
1 资产	6			
<b>同</b> 设备	6 16:27:10 16:27:20 16:27:30 16:27:40 16:27:50 1	5:28:00		
₩ 实体视图		平均值		
部件库	- current_trequency	6		
■ 仪表板库				
<b>③</b> 审计日志				

通过手机也可以查看历史曲线。如果提供给用户使用,需要由租户创建 custom 账户,然后分享给 custom,客户就可以登录查看。

10.4.4 远程控制继电器,设置频率并查看频率曲线

在仪表盘库页面选择【添加新的仪表盘】按钮(右下角红色圆圈),



添加仪表板	?	×
标题*		
my dashboard		-
描述		_
添加	取消	





# 进入编辑模式

🎉 ThingsBoard	■■ 仪表板库	🕂 my dashboard			::	:
♠ 首页	my dashboard		60	S	Ŧ	:3
<b>〈··〉</b> 规则链库						
22 客户						
<b>王</b> 资产						
□□ 设备						
➡ 实体视图						
計 部件库		没有配置部件				
➡ 仪表板库						
🕲 审计日志						
					进入	、编辑模式
			Pow	ered by Th	ningsboar	d v.2.5.4

# 设置实体别名



添加实体别名





添加别名				×
别名* device1			解决为多实体	^
过滤类型* <b>设备类型</b>			<b>~</b>	
设备类型* Modbus Ga	teway		×	
名称前缀		添加	取消	~

# 添加完并保存





下面开始添加每个部件 (Widget)

# 10.4.4.1 添加开关按钮





₽ 仪表板库	> 📑 my	dashboard			::	:
♦ ■			\$ []	$\bigcirc$	Ŧ	::
^{标题*} my dash	选择部	件				×
	当前包	Cards		系	统	
		Charts		系	统	
		Control widgets		系	统	
		Date		系	统	
		Digital gauges		系	统	
			 		~	

## 选择 Switch Control





汊	添加部件							?	×
	数据	设置	高级	动作					^
	目标设备	·							
	device	1						×	
									~
							添加	取消	

在高级设置里面,这两个参数 RPC get value method 和 RPC set value method 很重要,涉及到跟网关通讯。其中, RPC get value method 是 ThingsBoard 用来获取网关某个参数数值的通讯方式,该通讯方式基于 mqtt json,网关收到问询后 会回复当前数值。同样, RPC set value method 是 ThingsBoard 设置网关某个参数 的通讯方式。在跟网关通讯时,这两个参数必须遵循如下格式:

RPC get value method: get_\$TAGNAME

RPC set value method: set_\$TAGNAME

其中,\$TAGNAME就是网关上报数据模板中的某个参数,这里继电器用的是 relay_1, 所以这两个参数用的是 set_relay_1 和 get_relay_1。如果用其它控制部件,比如设 置目标频率,那么对应这两个参数就是 set_target_frequency 和 get_target_frequency。我们再看一下网关的 mqtt 上传模板:

{"current_frequency": \${VW20}, "target_frequency": \${VW22}, "relay_1": \${V0.0}}



添	加部件	:						?	×
	数据	设置	高级	动作					^
	RPC ge	et value method elay_1						^	
	RPC se	t value method elay_1							
	Parse v	ralue function, for return da	(data), returns ata ? <b>tr</b> u	boolean 1e : false;				ł.	
					JAVASCRIPT	TIDY	FULLSCREEN		~
							添加	取消	

## 添加完注意应用更改,





这时按下按钮就可以看到网关下的继电器会有打开

■■ 仪表板库 > 🚽 my dashboard
my dashboard
Switch control

另外,在网关的【系统日志】里面也可以看到 ThingsBoard 下发的信息,并打开继电器的记录,方便调试查问题。

系统[	系统日志						
	ID	Time	Туре	Content			
	1	10284.258	INFO	MQTT RECV: message len = 38			
	2	10284.259	INFO	MQTT: V0.0=1			
			-				
刷新	Ť	清除  保存					

10.4.4.2 添加频率设置按钮





添	加部件						?	×
_	数据	设置	高级	动作				,
	目标设备							
	device1						×	
						添加	取消	



添加部件		?	×
Knob title			^
设置目标频率			
Get value method *			
get_target_frequency			
Set value method *			
set_target_frequency			
RPC request timeout *			
500			
		~	
			~
	添加	取消	

# 设置浮点位数,若是整数就是0

<mark>设置频率</mark> Knob Control					? >
数据 设置	高级	动作		Ŭ	
<u>◆</u> 背景颜色 #e6e7e8		<b>∢.</b> 文字颜色 rgba(0, 0, 0, 0, 0.87)	填充 Opx	边缘	
特殊符号展示值			浮点数后的位数 0		٢
移动端设置					
顺序			● 高度		•

## 添加完就可以看到下图的频率设置旋钮,可随意设置频率





# 10.4.4.3 添加当前频率和设置频率的曲线



# BRIMESH http://www.brimesh.com

溕	家加部化	4							?	×
	数据	设置	高级	动作						^
	<ul> <li>✓ 使用仪表板的时间窗口</li> <li>✓ Display timewindow</li> <li>数据源</li> </ul>					时间窗口	<b>()</b> 实时-最后分		_	
		类型	参数							
	1.	实体	✓ devic	ce1	×		current_freq : curi :arget_frequ : tarç	rent_frequ / ×	×	~
								添加	取消	



## 10.4.4.4 手机查看和控制

tenant 账户 (可以编辑仪表盘) 可以创建 custom 账户 (只有查看仪表盘权限), 然后通过手机浏览器就可以远程查看控制仪表盘。





## 10.4.5 RESTFUL API

ThingsBoard 提供丰富的 Restful 接口, 方便客户做二次开发, 通过如下链接,

就可以查看具体的 api 含义 <u>http://YOUR_HOST:PORT/swagger-ui.html</u>,例如

#### http://192.168.124.202:8080/swagger-ui.html

rpc-controller : Rpc Controller	Show/Hide L	st Operations Expand Operations									
POST /api/plugins/rpc/oneway/{deviceId}			handleOneWayDeviceRPCRequest								
Post /api/plugins/rpc/twoway/{deviceId}			handleTwoWayDeviceRPCRequest								
Response Class (Status 200) OK Model Example Value											
{     "result": {},     "setOrExpired": true }											
Response Content Type */* >											
Parameters											
Parameter Value	Description	Parameter Type	Data Type								
deviceId (required)	deviceId	path	string								
requestBody (required)	requestBody	body	string								

#### 详细资料,请查看官网资料。

https://thingsboard.io/docs/api/

10.4.5.1 通过调试工具 CURL 实现远程控制

将下面的文本复制到一个文本文件,并以 bat 结尾,例如 get_token.bat,其中 替代里面的用户名和密码,修改 IP 地址

```
curl -X POST --header "Content-Type: application/json" --header "Accept:
application/json" -d "{\"username\":\"user@email\",
\"password\":\"12345678\"}"
http://192.168.124.202:8080/api/auth/login
```

运行后收到如下回复,
BRIMESH http://www.brimesh.com

{"token":"eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ3dWp1bkBicmltZXNoLmNvbS IsInNjb3BlcyI6WyJURU5BTIRfQURNSU4iXSwidXNlcklkIjoiYTNiYzhmNDAtMT k2Zi0xMWViLWE2NWItNTFIMDk5YTA4ZmNhIiwiZW5hYmxlZCI6dHJ1ZSwia XNQdWJsaWMiOmZhbHNILCJ0ZW5hbnRJZCI6IjZjYWFjZmQwLTE5NmYtMT FIYi1hNjViLTUxZTA5OWEwOGZjYSIsImN1c3RvbWVySWQiOiIxMzqxNDAwM C0xZGQyLTExYjItODA4MC04MDgwODA4MDgwODAiLCJpc3MiOiJ0aGluZ3Ni b2FyZC5pbyIsImlhdCI6MTYwNTQ3OTU0NywiZXhwIjoxNjA1NDq4NTQ3fQ.X HW8yFkoodCTrY4Lh-A-K7GZhflOOFB2L_mHqXzaEh4qBbXYv2iPvqlgmZrA2 sg1BY8 u5p-T98ht AUvCWx9w", "refreshToken": "eyJhbGciOiJIUzUxMiJ9.e yJzdWIiOiJ3dWp1bkBicmltZXNoLmNvbSIsInNjb3BlcyI6WyJSRUZSRVNIX1R PS0VOII0sInVzZXJJZCI6ImEzYmM4ZjQwLTE5NmYtMTFIYi1hNjViLTUxZTA5 OWEwOGZjYSIsImlzUHVibGljIjpmYWxzZSwiaXNzIjoidGhpbmdzYm9hcmQu aW8iLCJqdGkiOiI0MzE0ZjM4NS1hYjU2LTQ4YzQtYTA0ZS04YzVhNjFlMDlmM GEiLCJpYXQiOjE2MDU0Nzk1NDcsImV4cCI6MTYwNjA4NDM0N30.uKepaiBB egohZLvbRI99tG2sQrCo7o7-fUO455PmnxhTE2y1jbN0ISLMjO38GSb7NOgS zFG4gzmVvRsGolgYDg"}

把上面下划线部分替代掉下图框中的 token,下图中的粗体部分需要在 thingsboard 网页中【设备】页面选择对应的控制的设备,并选择【复制设备 id】, 并替代掉下框中粗体部分,并以文件 twoway.bat 保存,并运行

curl -X POST --header "Content-Type: application/json" --header "Accept: application/json" --header

"X-Authorization:Bearer<u>eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJ3dWp1bkBicmlt</u> ZXNoLmNvbSIsInNjb3BlcyI6WyJURU5BTIRfQURNSU4iXSwidXNlcklkIjoiYTN iYzhmNDAtMTk2Zi0xMWViLWE2NWItNTFIMDk5YTA4ZmNhIiwiZW5hYmxlZ CI6dHJ1ZSwiaXNQdWJsaWMiOmZhbHNILCJ0ZW5hbnRJZCI6IjZjYWFjZmQ wLTE5NmYtMTFIYi1hNjViLTUxZTA5OWEwOGZjYSIsImN1c3RvbWVySWQiOi IxMzgxNDAwMC0xZGQyLTExYjItODA4MC04MDgwODA4MDgwODAiLCJpc3 MiOiJ0aGluZ3Nib2FyZC5pbyIsImlhdCI6MTYwNTQ3OTU0NywiZXhwIjoxNjA 1NDg4NTQ3fQ.XHW8yFkoodCTrY4Lh-A-K7GZhflOOFB2L_mHqXzaEh4qBbX Yv2iPvqlgmZrA2sg1BY8_u5p-T98ht_AUvCWx9w" -d

"{\"method\":\"rpc\",\"params\":{\"relay_1\":1}}" "http://192.168.124.202:8080/api/plugins/rpc/twoway/**c9eae1d0-196f-**

## 11eb-a65b-51e099a08fca"



运行结果如下:

C:\software\thingsboard_tools>curl -X POST --header "Content-Type: application/j son" --header "Accept: application/json" --header "X-Authorization:Bearer eyJhbG ci0iJIUzUxMiJ9.eyJzdWIi0iJ3dWp1bkBicmltZXNoLmNvbSIsInNjb3BlcyI6WyJURU5BT1RfQURNS J4iXSwidXNlcklkIjoiYTNiYzhmNDAtMTk2Zi0xMWViLWE2NWItNTF1MDk5YTA4ZmNhIiwiZW5hYmx1Z CI6dHJ1ZSwiaXNQdWJsaWMiOmZhbHN1LCJ0ZW5hbnRJZCI6IjZjYWFjZmQwLTE5NmYtMTF1Yi1hNjViL FUxZTA50WEwOGZjYSIsImN1c3RvbWvySWQi0iIxMzgxNDAwMC0xZGQyLTExYjIt0DA4MC04MDgwODA4M DgwODAiLCJpc3Mi0iJ0aG1uZ3Nib2FyZC5pbyIsImlhdCI6MTYwNTQ3OTUONywiZXhwIjoxNjA1NDg4N FQ3fQ.XHW8yFkoodCTrY4Lh-A-K7GZhf100FB2L_mHqXzaEh4qBbXYv2iPvq1gmZrA2sg1BY8_u5p-T9 Sht_AUvCWx9w" -d "{\"method\":\"rpc\", \"params\":{\"relay_1\":1}}" "http://192.1 68.124.202:8080/api/plugins/rpc/twoway/c9eae1d0-196f-11eb-a65b-51e099a08fca" {"method":"rpc", "params":{\"relay_1":1} C:\software\thingsboard_tools>

这是可以看到继电器被打开,同时在网关的实时数据里面也可以看到对应的 Bit 被设置

成1,如下图:

时数据										
Address: Quantity:	s: $0xxxx \vee 0xxx \vee 0xx \vee 0x \vee 0x \vee 0x \vee 0x$									
Data Format:	Binary	~	]							
Address	0	1	2	3	4	5	6	7	8	9
0	0000000 <mark>1</mark>	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
U	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
10	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
10	00000110	00001000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
澵										



11 常见问题

## 11.1 如何判断从站数据是否已经采集上来

通讯成功后,可以在【实时数据】页面显示被采集上来的数据,如果有通讯失败的,在 【故障数据】页面里显示出故障的记录,并且可在【系统日志】页面查看具体的原因。

11.2上位机如何通过网关把数据写入从站

[0] NOP
<ol> <li>Read Coils &amp; Write Multiple Coils</li> </ol>
[2] Read Discrete Inputs
[3] Read Holding Registers & Write Multiple Registers
[4] Read Input Registers
[5] Write Single Coil
[6] Write Single Register
[15] Write Multiple Coils
[16] Write Multiple Registers
[128] Copy Single Register
[129] Copy Multiple Registers
[130] Read Coils & Write Single Coil
[131] Read Holding Registers & Write Single Register

在上图这些功能码中,5,6,15,16是以设定的扫描周期把源虚拟地址中的数据写入 目标从站,这里,源虚拟地址中的数据来源于某个从站,也可以来源于上位机主站。网 关相当于不停地把虚拟地址中的数据搬运从站地址空间。

而功能码 1, 3, 130, 131 是以设定的扫描周期间隔把从站数据读取到虚拟地址空间。 当上位机往虚拟地址写数据的时候,功能码 1 就会通过写多线圈的命令 15 把改动的线 圈写入到从站,功能码 3 通过写多个寄存器命令 16 把改动的数据写入从站,功能码 130 通过写单个线圈的命令 5 把改动的线圈写入到从站,功能码 131 通过写单个寄存器的 命令 6 把改动的单个寄存器写入到从站。这些写从站的动作只有在收到上位机写数据请 求时才会触发,区别于 5, 6, 15, 16 功能码 (不停地循环写),功能码 1, 3, 130, 131 在往从站写完数据后,下个周期会继续读数据。

## 11.3 如何对从站只写不读

在功能码 1, 2, 3, 4, 130, 131 中选择满足要求的选项,并且把扫描周期设置成 0, 这样网关就不会去读,但在收到上位机写数据请求的时候,网关会把数据写入到从站。

BRIMESH http://www.brimesh.com

11.4 如何实现 2 个从站间的数据搬运 (通讯)

首先,在网关设置读命令,把从站数据读取到指定虚拟地址,然后设置写命令,网关会把虚拟地址内的数据写入设定的从站。注意,读命令要在前,写命令在后。

## 11.5 如何升级固件

通过网络升级固件的功能极大地提高了设备在现场的可维护性。在升级之前,请先 下载要升级的新的固件,并保存到某个目录,假设名字是 R6.20.07.bin,保存于 E 盘。 同时请下载安装 tftp Server。推荐使用 tftpd32 软件,非常易用和方便。详细下载地 址是:<u>http://tftpd32.jounin.net/</u>,(在我们官网上链接 <u>http://www.brimesh.com/download/TFTPD.rar</u>)打开显示如下界面,在"Current Directory"选择固件所保存固件的文件夹,这里选择 E 盘。这样 TFTP 服务器就安装好 了。

🏘 Tftpd32 by Ph. Jounin							
Current Directory E:\ Browse							
Server interfaces 192.168.1.123  Show D							
Tftp Server Tftp Client Syslog server Log viewer							
peer	file	start time pr	progress				
•	III		٩				
About	Settings		Help				

之后打开浏览器访问网关,在【系统设置】菜单内可看到【升级固件】对话框,输入tftp 服务器IP地址(也就是计算机的IP,Windows系统通过ipconfig命令查看)和固件 全称(注意,这里一定要全称,包括文件后缀.bin),如下图:



升级固件	
TFTP服务器:	192.168.1.123
固件文件名:	R6.20.07.bin
升级	

点击【升级】。整个升级过程约 20 秒左右,然后手动刷新网页。升级成功后可以通过 系统信息的页面查看新的版本号。

开始升级的提示:

• Start upgrading firmware, please wait 20s and manually refresh the page.

看到网页显示这条消息后, 网关会开始升级并且灯会闪烁, 整个过程大约 20 秒, 然后 请手动刷新网页。

升级失败的提示:

• ERROR: Firmware R6.20.07.bin is not found at tftp server!!!

错误原因:固件没有找到,请确认固件是否放置到 tftp 服务软件的当前目录(Current Directory)下,并且名称填写正确。

• ERROR: TFTP service at 192.168.1.123 is unavailable!!!

错误原因:无法访问服务器,或者 tftp 端口(UDP 端口 69)没有打开。确认 IP 是可访问的,并且 tftp 服务端口没有被防火墙屏蔽掉。