

# WSTools 4G 网关

## 使用手册



赋能万物互联

驱动智能未来

佛山卫软科技有限公司

## 目录



	1
1. 概述 .....	3
1.1. 产品概述 .....	3
1.2. 数据采集系统架构: .....	3
1.3. 网关设备数据通信原理: .....	5
1.4. 产品规格型号 .....	6
2. WSTools 软件启动 .....	7
2.1. 设备接入 .....	7
2.2. WSTools.exe 应用程序 .....	8
2.3. 执行应用程序 .....	9
2.4. 查看设备日志 .....	10
3. 功能使用说明 .....	11
3.1. 通信连接 .....	11
3.2. 基础配置 .....	12
3.3. 系统 ID 和输入源（重要）: .....	14
3.4. 采集协议 .....	15
3.5. 网络传输 .....	25
3.6. 应用层协议 .....	36
3.7. IOT 云平台配置 .....	51
3.8. 系统配置 .....	57
4. 附录: .....	60
4.1. 本地下载固件 .....	60
4.2. 附录 1: Cron 表达式 详细说明 .....	60

# 1. 概述

## 1.1. 产品概述

FY-XXX 系列产品是我公司研发的物联数据采集网关，是一种基于 4G Cat. 1 模组的 DTU，主要用于物联网终端设备的数据采集和边缘计算及远程控制，支持 TDD-LTE/FDD-LTE 的无线传输通信。可以通过串口进行各种串口协议的数据采集，有 RS485 串口和 RS232 可选，对接各类设备传感器，并完全实现了 modubs-rtu 标准协议，同时支持模拟量的采集和数字量的专门接口采集，对所采集的数据进行边缘计算，按照定制协议进行数据传输，支持 TCP/UDP/HTTP/MQTT 等方式的数据传输，支持 TDD-LTE、FDD-LTE 网络数据连接，同时可为客户提供数据传输、协议解析及场景联动与二次计算等功能。

FY-DTU 网关可广泛应用于各个物联网领域，如工业数据采集、智慧农业、气象要素采集、电力监控、水文水利、环境污染监测、智能家居、安全管理等多种应用场景。

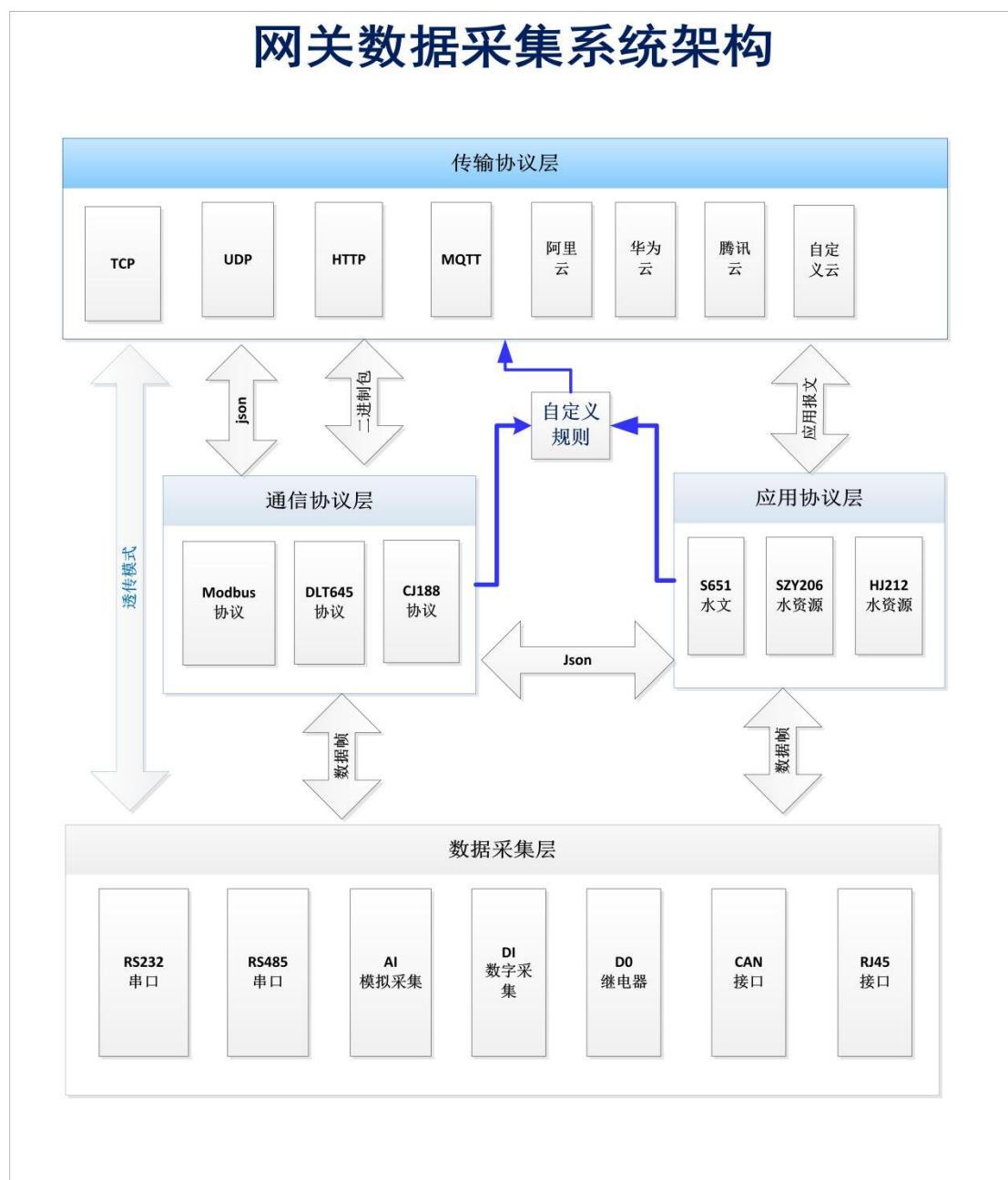
FY-DTU 网关采用先进的高度集成的硬件和软件平台，对众多常用的物联网平台协议进行了优化，并实现了物理层、采集层、传输层、应用层等各层应用协议的灵活配置，并对各层进行对象化管理，按照流式方式进行各层数据的输入与输出。

## 1.2. 数据采集系统架构：

采用分层的网络系统架构：

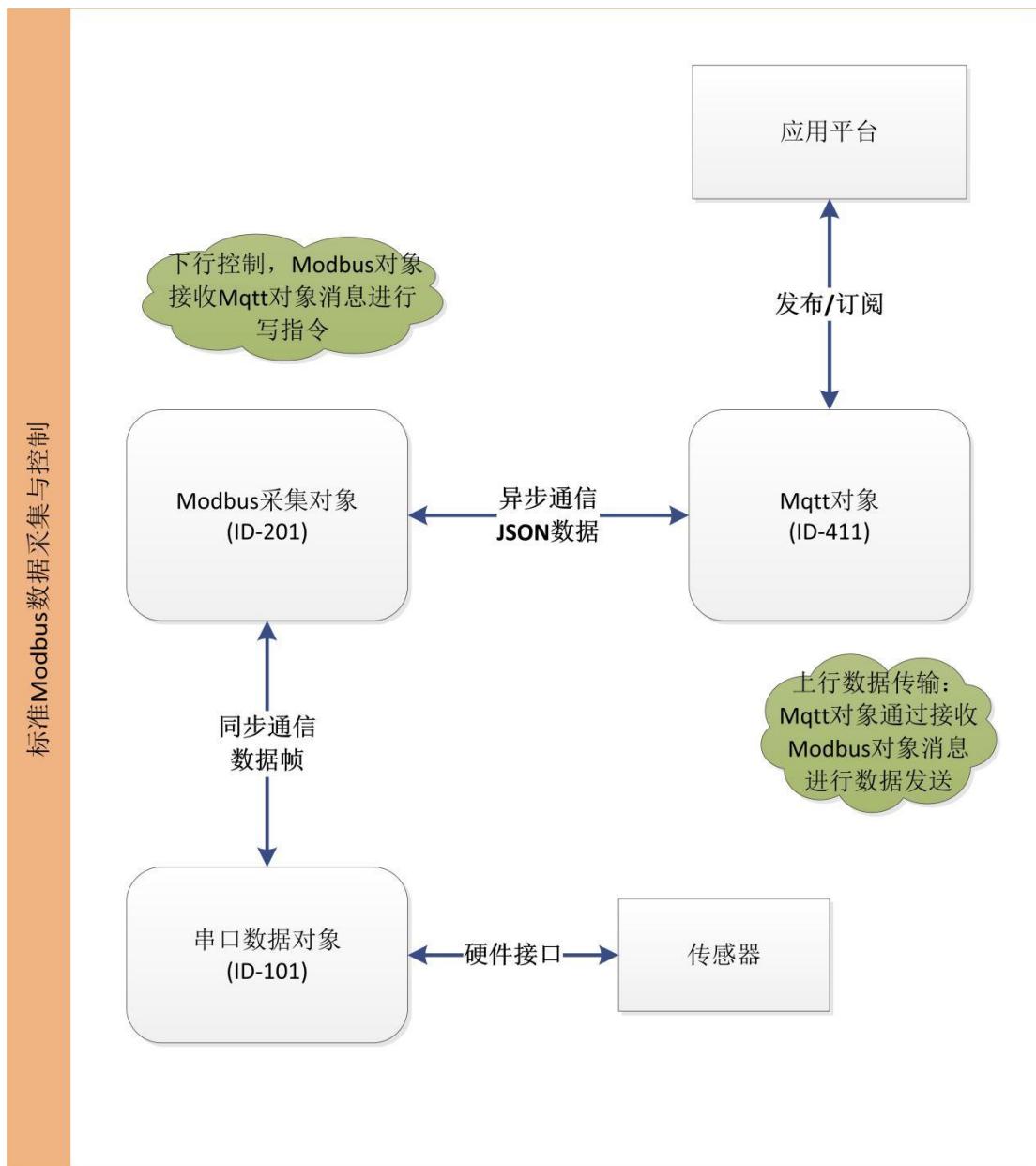
网络分层	详情	备注
硬件接口层	分串口 RS232、RS485、AI、DI/DO、CAN	AI 模拟量电压输入（0~5V）开关量输入；DI（开关量输入，高低电平）；DO（继电器输出）
串口协议层	MODBUS、DLT645、CJ188	DLT645 电力抄表协议；

		CJ188 水表、气表抄表；
网络传输层	TCP/UDP/MQTT/HTTP	支持证书加密，支持动态密码验证
应用协议层	S651/HJ212/SZY206/自定义规则/二次计算	S651: 水文采集协议； HJ211: 环保采集协议； SZY206: 水利采集协议；
云平台协议	阿里云、腾讯云、百度云、ONENET 云、华为云 TLINK 云、涂鸦云、私有云	云平台连接协议可能随云平台的升级有所变化。



### 1.3. 网关设备数据通信原理:

网关设备应用软件采用数据对象化、节点化的数据结构，并采用流水线式作业方式对各层数据进行输入输出处理。所有对象节点统一输入输出接口，各个对象节点可以无缝对接。各个对象节点既可以异步通信，也可以同步通信。



## 1.4. 产品规格型号

型号	硬件接口	功能说明
FY-101	单路 485	串口协议, Modbus 协议, 抄表协议, TCP/UDP/MQTT/HTTP, 自定义规则与二次计算, 水文 S651/SZY206/HJ212 , 云平台接入
FY-102	1 路 485+1 路 232	同上
FY-103	双路 485	同上
FY-103B	双路 485	导轨式
FY-104	双路 485+1 路 232	同上
FY-105	单路 485+2 路 AI+2 路 DI+2 路 DO	多模拟输入, 数字输入和继电器输出, 其他功能同上
FY-201	单路 485+RJ45+2 路 DI+2 路 DO	加上本地 LAN 网络功能, 数字输入和继电器输出, 没有模拟输入
FY-104M	双路 485+1 路 232	8M 内存+8M 存储, 用于高性能场景, 支持上千个点的数据采集
FY-103M	双路 485	8M 内存+8M 存储, 用于高性能场景, 支持上千个点的数据采集
FY-104U	双路 485+1 路 232	海外版, 8M 内存+8M 存储, 用于高性能场景, 支持上千个点的数据采集

## 2. WSTools 软件启动

### 2.1. 设备接入

#### 2.1.1. USB 驱动

直接用 Type-C 数据线连接网关设备，windows 系统会自动产生 3 个虚拟串口。一般不需要装 usb 驱动，串口号或许因电脑环境不同而不同的端口号。如图所示：



#### 2.1.2. 设备运行状态

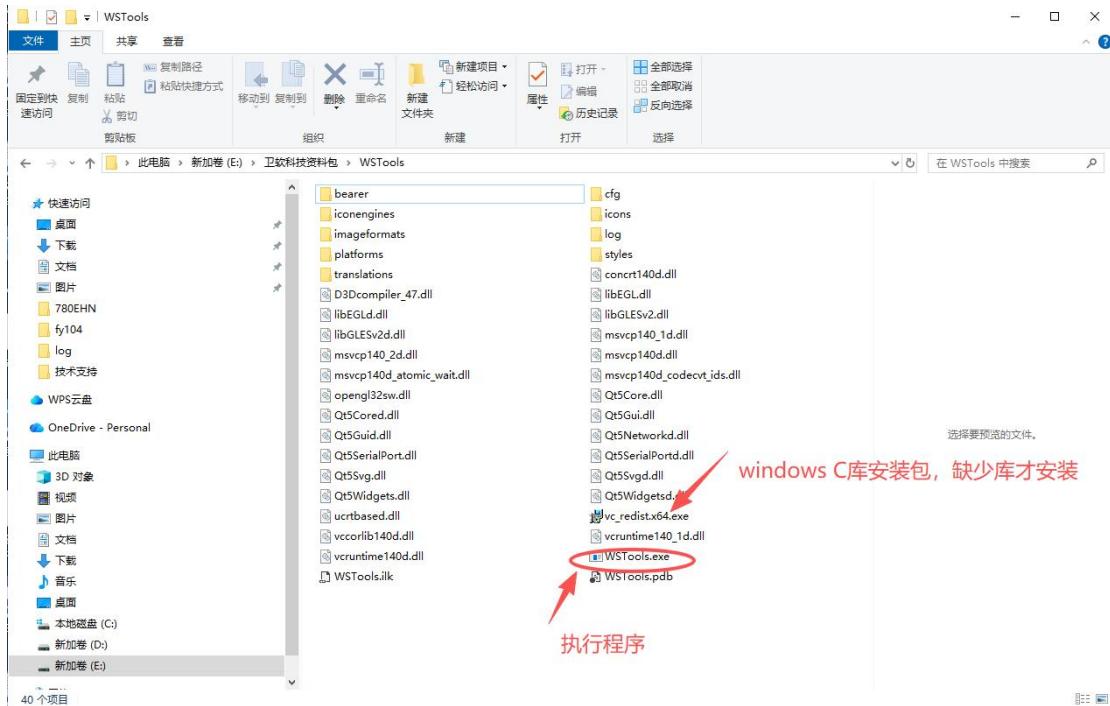
LED: 红灯亮表示电源开启，绿灯亮表示 4G 网络已经正常联网，绿灯灭表示外网不通，黄灯闪烁表示应用程序正常运行中。如图所示：



## 2.2. WSTools.exe 应用程序

直接解压我公司提供的软件压缩包 WSTools.zip 文件。将会看到下面文件，由于 windows 环境各不相同，有的需要安装 C 库，就安装 vc\_redist.x64.exe .wstools.exe 是配置工具执行程序，采用 c++ 开发，

因此需要依赖 windows 环境 C 库，注意本软件是 64 位软件，如需 32 位环境，请联系技术支持，重新提供 32 位程序。

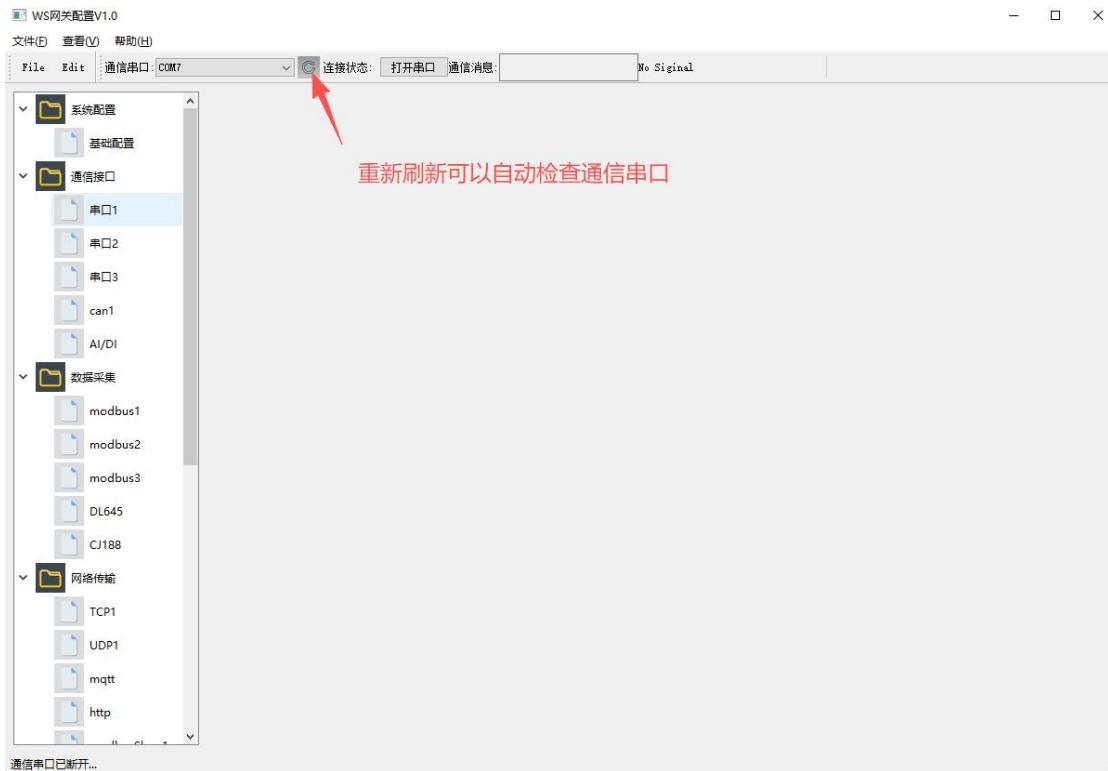


## 2.3. 执行应用程序

直接双击 wstools.exe，正常会出现下面界面：



如果看不到右边工作区，请点击刷新图标，如下图所示：



## 2.4. 查看设备日志

点击查看菜单->查看日志，打开网关设备日志窗口。日志查看是设备连接后的三个虚拟串口的第一个串口，这里是串口 5。可以暂停日志显示，删除日志等常用操作，还可以将日志保存，日志保存在运行目录下的 log 子目录下，并以时间命名。



### 3. 功能使用说明

#### 3.1. 通信连接

WSTools 配置程序和设备终端的通信采用串口通信, 使用 Type-c usb 串口通信连接, 其通信速率和其他参数自动设置, 用户无需设置。

Wstools 终端程序和设备通过 usb 虚拟串口进行通信, 点击通信串口会列出 windows 所有串口, 选择正确的通信串口, 点击打开, 正常通信会收到通信消息, 和 4G 信号, 设备 cpu 温度等数据。如果你不知道是那个虚拟串口, 可以直接点击刷新按钮, 系统会自动找到正确的通信串口并自动打开。如图所示:

WSTools 串口通信连接图





## 3.2. 基础配置

基础配置模块可以查看设备的 ID(imei)，设备应用程序版本以及本地 IP、运行模式，日志调试等级等重要信息。

基础配置可以设置配置版本（整数数字），运行模式：在线运行和离线运行；调试等级，以及定时重启时间，断网重启时间等参数。

确保通信正常后，点击读取配置可以读取本地设备的基础配置信息，点击保存配置即将修改后的配置保存到设备终端。

### 基础配置界面



### 3.2.1. 调试等级

调试模式一般用于初次试用调试设备用，便于查找配置错误等问题。强烈建议系统调试完毕后设置无日志模式。

### 3.2.2. 设置运行模式

在线运行	终端连接卫软科技 IOT 平台，可在线配置管理	
离线运行	终端不开在线配置	
自定义通信	终端连接私有配置平台，可在线配置管理	

### 3.2.3. 设置通信服务器

自定义通信通信模式下需配置服务的 IP 地址，端口，用户和密码信息。

### 3.2.4. 初始化设备

初始化设备将清空所有已有配置。

### 3.2.5. 强制升级

一般终端联网开机会自动检查当前应用升级版本。如果需要其他应用，可以强制切换升级。

### 3.2.6. 自动重启机制

网关重启机制需要自己事先设定，有定时重启和断网重启两种机制。如果设置定时重启时间或断网重启时间大于 0，将按设定分钟启动重启机制。

### 3.2.7. 一键配置

如果多个网关配置一样功能，可以点击一键配置，将会把主机上的有效配置全部发送到终端设备。

## 3.3. 系统 ID 和输入源（**重要**）：

本终端系统全部采用对象输入输出流的方式进行数据采集与传输，使用前必须理解我们的设计原理和通信机制。从数据采集-数据解析-边缘计算-数据传输的整个数据处理流程，我们都设计相应的数据处理对象，各对象处理数据过程就好比流水，通常称之为数据流，各对象有系统统一分配唯一的 ID，每个对象都可以引用其他对象作为输入源，在输入源编辑框输入对象的 ID，多个对象就用逗号分隔，即标识输入源的对象数据处理后的输出将作为本对象的输入进行二次处理。每个对象启用后将独立工作。处理完数据后将发送消息给下游对象。



**输入源界面**

**是否启用：** 对该模块进行启动或停止操作。

**输入源：** 输入源是接收上游对象的输出，并作为本对象的输入，其数据格式多为 JSON 对象或透传二进制数据包，可以接收多个对象数据进行处理，多个

对象用逗号分隔。

自动升级：勾选自动升级后设备启动会检查是否有更新，如有更新就会立即更新后重启动，**每次固件升级过程中都有可能异常断电风险，如果应用稳定可以选择不自动升级。**

运行模式：在线运行，设备和卫软科技内部的 iot 平台进行通信，并进入在线设备管理，便于远程配置。在线运行模式只有配置时候设置或读取数据使用流量，平时连接占有流量极小。如果设置离线运行，就不能进行在线远程配置。

## 3.4. 采集协议

### 3.4.1. 串口配置

串口配置是配置设备终端和其他设备通信的硬件通信配置。主要是 RS232 配置和 RS485 通信配置：注意设备端口号的对应：

通信接口	对应端口号	备注
RS232	串口 1	FY104、FY102 才有 RS232
RS485-1	串口 2	
RS485-2	串口 3	

串口设置其他参数：

波特率、校验位、数据位、比特顺序、测试输出

测试输出是用于配置时候测试串口通信是否正常，你可以用串口助手来接收测试消息，正常使用请禁用测试输出，以免占用设备资源。

比特顺序一般都是低位在前，如有特殊需求请选择高位在前，要根据对端设备的通信规约进行设置。



### 3.4.2. AI/DI/DO

AI: 模拟输入, 主要用于采集电压, 采集周期为间隔秒, 或输入时间 Cron 表达式按计划触发采集。

DI: 开关量输入, 其触发模式分上升沿触发, 下降沿触发, 和均触发三种模式。

DO: 控制继电器开关, 初始电平为低电平或高电平, 持续时间如果大于 0, 标识 DO 输出秩序指定时间后就恢复到原来状态, 比如开灯, 持续时间 1 秒, 1 秒后又恢复到关灯状态。



### 3.4.3. Modbus 采集配置

Modbus 完全按照 modbus 标准协议实现，支持 RTU 和 TCP 两种协议方式，可以读取传感器设备的数据，也可以写入参数，读取和写入的数据都是用 JSON 对象进行数据交换，所有采集的数据都返回在 JSON 对象中，写入数据也是，上层系统只需要把要设置的数据放到 JSON 对象中，网关将会根据参数映射表进行设置。

#### 1. 基础配置

属性名称	输入说明	解释
发送通道		可以是串口，TCP, MQTT
通信协议	ModbusRTU ModbusTCP	ModbusRTU 协议是基于串口的，因此必须首先选择 modbus 对象操作的串口号，注意串口号对应。
指令发送间隔	100 毫秒	有的传感器因通信速率可能会粘包（几条指令粘在一起），就需要增大发送间隔，比如 200 毫秒
合并模式	合并发送/单独发送	合并发送表示几条读指令采集的数据一起发送，单指令发送则是分开发送
采集设备状态	从站设备通信状态	如果从站设备不在线或其他故障，将返回该设备的状态 {"_devState": {"s1": 1, "s2": 0}} s1 表示从站 1，1 表示在线，0 不在

属性名称	输入说明	解释
		线
设备上报周期		上报设备状态的间隔周期, 秒

## 2. 串口对应

UART 端口	对应串口	备注
串口 1	RS232	232 串口只有 FY104,FY102 才有
串口 2	RS485-1	
串口 3	RS485-2	

Modubs TCP 协议是基于网络的, 只需要配置网络连接对象就可以。

ModubsRTU 还可以通过网络透传方式来实现, 即可实现 RTU OVER TCP。

## 3. 指令配置:

点击新增指令按钮, 在指令配置表的新增行中输入:

从站 ID, 指令, 起始地址, 数量, 采集周期, 是否单独采, CRON 表达式

属性名称	输入说明	解释
从站 ID	从站地址编号 (1~255)	
指令	选择 modus 指令	有读寄存器, 写寄存器, 读线圈, 写线圈等指令。
起始地址	10进制数字输入	操作寄存器的起始地址
数量	数字输入	操作寄存器的个数
采集周期	数字输入, 单位秒	循环采集间隔周期
上报模式	选择输入	立即上报, 规则上报, 时间周期且规则, 时间周期或规则
上报周期 (CRON 表达式)	*****	上报时间可按 cron 表达式进行计划时间表进行周期上报,cron 格式为 6 字段的配置格式, 分别是 [秒 分 时 日 月 周] 的格式进行配置

## 4. 变量映射表配置:

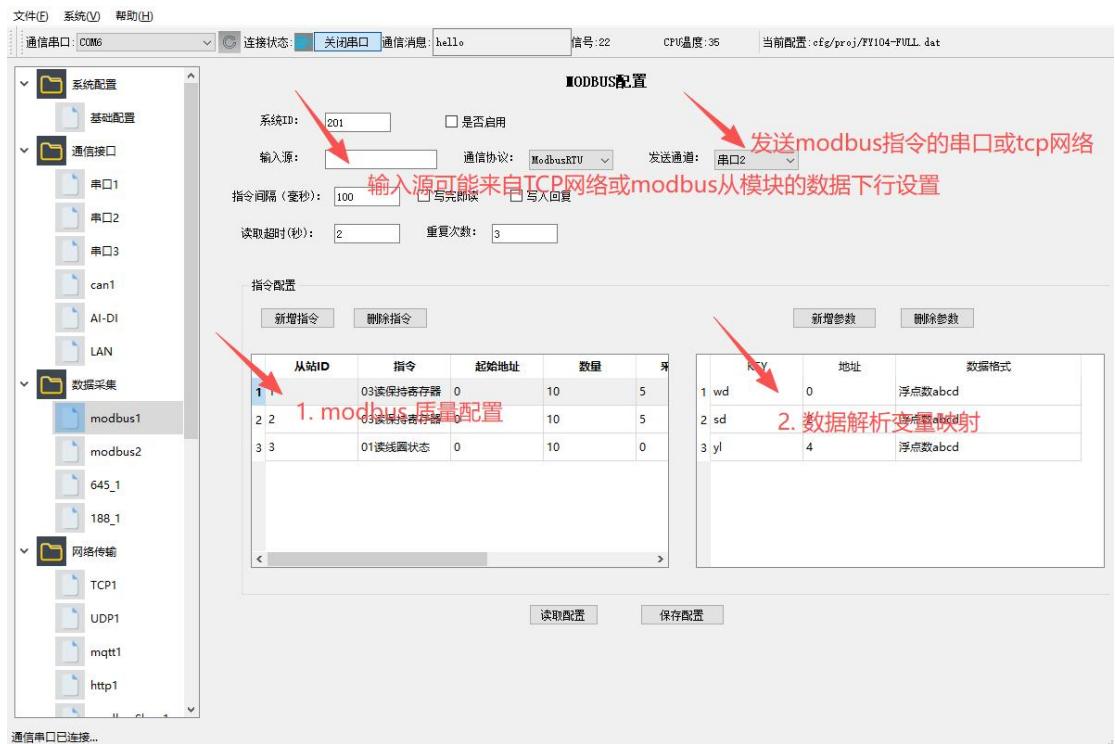
每条 modbus 指令对应一个参数表, 参数表主要配置采集的数据解析及地址映射。

属性名称	输入	解释
KEY	字符串	数据解析变量, 对应解析后的 Json 数据对象的某个属性,

属性名称	输入	解释
		比如温度, {"wd":10}
地址	变量对应的数据地址	根据地址和数据格式所占位数进行读取数据或写入数据
数据格式	选择输入	各种类型的数据格式, 分大端小端字节顺序, 需要根据连接的终端配置进行。 线圈操作: bit 单字节读写: 8 位 整数 双字节读写: 16 位 整数 四字节读写: 32 位 整数或浮点数 八字节读写: 64 位 双精度数
比例	数字	上报数据=采集数据*比例
偏移		上报数据=采集数据*比例+偏移
过滤条件	"x"表示当前变量 单条件和逻辑组合条件。 逻辑操作符不能有空格: >,<,==,>=,<=	取上下限之间: "x >= 1 and x <= 10" 取上下限制之外: lian"x <= 1 or x >= 10" 单独上限: "x >= 10" 单独下限: "x <= 10"

Modbus 操作流程指引, 如图所示:

1. 配置 modbus 指令, 按照标准 modbus 协议进行指令配置, 并设置相关的寄存器的地址和操作寄存器的个数、采集周期、上报时间周期策略 (cron)、上报规则。
2. 配置变量映射表, 不管读取指令和写入指令都需要配置映射表, JSON 数据如何到 modbus 指令包或指令包的数据如何变成 json 属性都是根据该映射表进行解析的。
3. 如果 modbus 下行数据有系统平台或本地其他设备发来数据, 就需要设置输入源, 比如 mqtt 对象 id 或 modbusSlave 对象 id 等等。
4. 设置发送通道就是 modbus 指令发送的硬件接口, 通常是串口, 也有通过 TCP 或 MQTT 网络进行指令发送。



## 5. 数据格式：

Modbus 数据格式可以是单字节整数，2 个字节的整数，4 个字节的浮点数，还有 bit 位的线圈数据，不同传感器设备的数据大小端格式不同，一定要根据对端规定的数据格式进行配置。注意单字节数据格式，由于 Modbus 协议按寄存器（2 个字节）操作数据，单字节数据需要指定字节顺序（ab），a 表示高位字节，b 表示低位字节。这两个单字节数据的寄存器地址是同一地址。



## 6. 数据采集及上报:

立即上报	按采集周期, 采集到数据即时上报	
规则上报	按采集周期采集, 采集数据根据变量过滤规则 条件判断是否满足才上报	
时间周期且规则	按采集周期采集, 按时间周期上报, 并且必须 满足过滤规则	
时间周期或规则	按采集周期采集, 如果规则条件满足就上报, 不满足, 如果时间周期条件满足也上报	

## 7. 批量配置

MODBUS配置

基础配置

系统ID: 201  是否启用 发送通道: 串口1  
 输入源: 通信协议: ModbusRTU  
 指令发送间隔(毫秒): 2 合并模式: 合并发送  采集设备状态 状态上报周期(秒): 20

可周期性采集设备状态

指令配置

新增指令  删除指令 新增参数  删除参数  删除所有  一键新增  一键同步

从站ID	指令	起始地址	数量	采集周期(秒)
1 1	03读保持寄存器	0x0000	10	10
2 2	03读保持寄存器	0x0000	10	10
3 1	16写多个寄存器	0x0000	4	0
4 2	06写单个寄存器	0x0000	1	0

KEY	地址	数据格式	比例
1 wd	0x0000	浮点数abcd	1 0
2 sd	0x0002	浮点数abcd	0 0
3 yl	0x0004	浮点数abcd	0 0

读取配置 保存配置

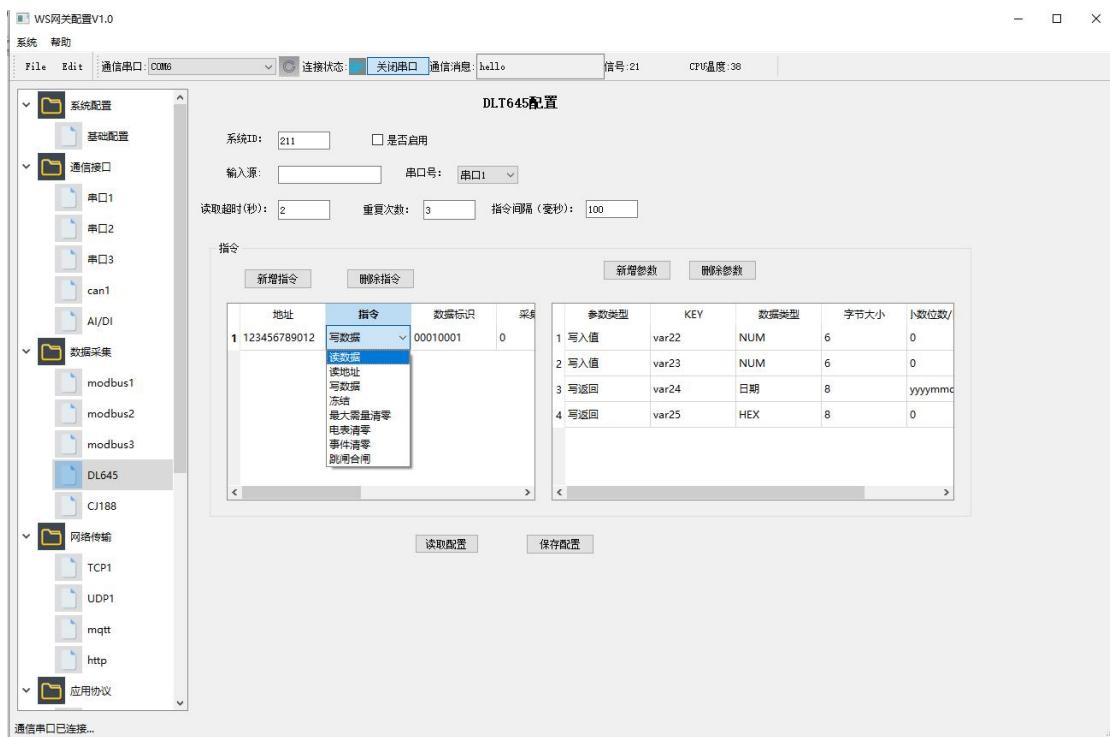
### 3.4.4. DLT645

属性	输入说明	解释
地址	表地址 (6字节, 低位在前)	电表地址
指令		有读数据, 读地址, 写数据, 跳闸等指令
数据标识	数据标识由4字节组	根据协议规定数据标识配置所需采集的数据

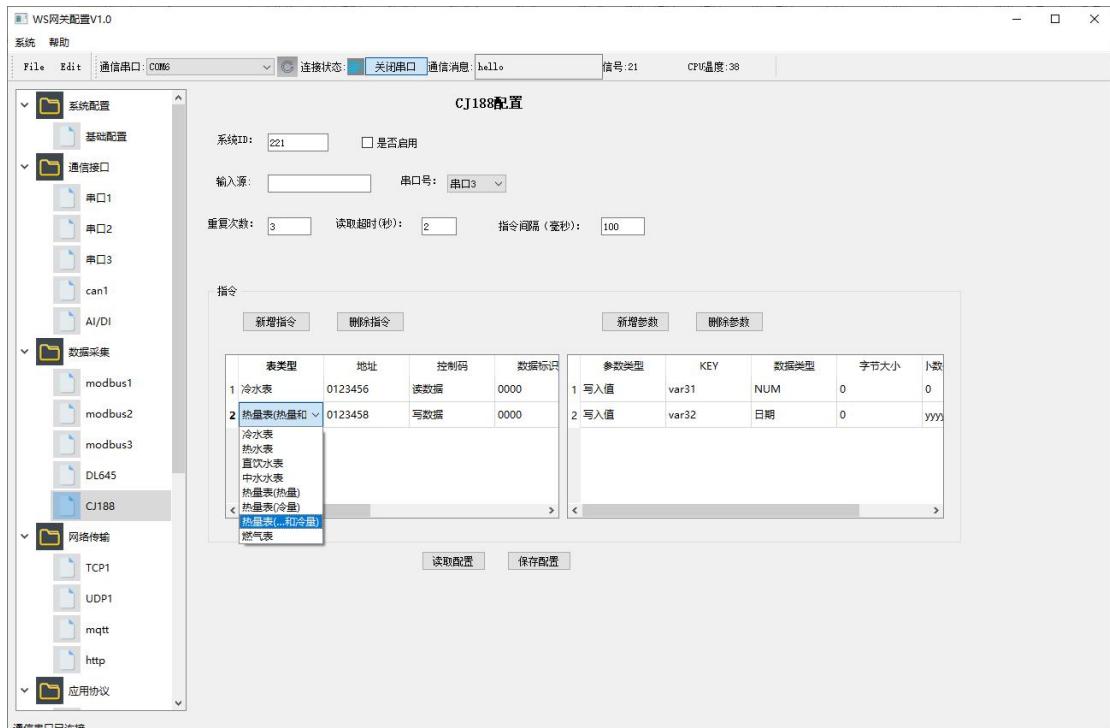
属性	输入说明	解释
地址	表地址 (6字节, 低位在前)	电表地址
	成, 表示电能表的具体数据项(如电压、电流、电量等)	
采集周期	数字输入, 单位秒	循环采集间隔周期
是否单独采	选择输入	单独采集就是该指令单独按照自己设定的周期进行采集, 否则就是按照第一条指令的周期进行统一采集
CRON 表达式	*****	如果采集周期=0, cron 不为空, 就按 cron 表达式进行计划时间表进行采集,cron 格式为 6 字段的配置格式, 分别是 [秒 分 时 日 月 周] 的格式进行配置
返回记录数	数字 N	N 条记录说明, 645 读取费率等数据会有多条记录, 配置需要取的记录个数, 如果不确定可以设置大一点, 数据有多少个就取多少个, 不设置或设置 1 就取第一条记录 如果 N 条记录, 变量名从第二条记录开始, 加 n, 比如变量名 v,v1,v2....v(n-1), 数据字节就是前面字段配置 总数*n

#### 参数表说明:

属性	输入说明	解释
参数类型	选择输入	读返回: 读取指令执行后的返回参数 写入值: 下行设置参数 写返回: 写入指令后返回的数据参数
KEY	参数变量	
数据类型	选择输入	数据格式: 0) 数值型, XXX.XXX ; 1) BCD 数字, NNNN ; 2) 日期, ssmmhhDDMMYY smhDMY 各种双组合 3) ascii ,AAAA ; 4) 十六进制数据 HEX 5) 空数据, NIL
字节大小		
小数位数或日期格式		配置中的地址, 数据标识项目都按正序输入字符串, 日期格式按数据规定设置, 多数是要反序 比如某时间格式 ssmmhhDDMMYY



### 3.4.5. CJ188 协议配置



新增指令：

属性名称	输入说明	解释
表类型		可选择水表, 热量表, 燃气表等不同类型
地址		A0~A6 地址域由 7 位字节组成, BCD 码 14 位, 低位在前, 高位在后
控制码		控制码: 可选读数据, 读地址, 写数据, 写地址, 写电机同步等选向。
数据标识		根据需要查表配置
采集周期	选择输入	单独采集就是该指令单独按照自己设定的周期进行采集, 否则就是按照第一条指令的周期进行统一采集
是否单独采	*****	如果采集周期=0, cron 不为空, 就按 cron 表达式进行计划时间表进行采集,cron 格式为 6 字段的配置格式, 分别是 [秒 分 时 日 月 周] 的格式进行配置
CRON 表达式	选择输入	单独采集就是该指令单独按照自己设定的周期进行采集, 否则就是按照第一条指令的周期进行统一采集

### 选择表类型

地址: 地址长度为 14 位十进制数, 低地址在前, 高地址在后

指令

新增指令
删除指令
新增参数
删除参数

表类型	地址	控制码	数据标识
1 冷水表	0123456	读数据	0000
2 冷水表	0123458	写数据	0000
热水表 直饮水表 中水表 热量表(热量) 热量表(冷量) 热量表(...和冷量) < 燃气表			

参数类型	KEY	数据类型	字节大小	小数
1 读返回	var1	NUM	8	1
2 读返回	var2	NUM	8	2
3 读返回	var3	日期	0	hhm

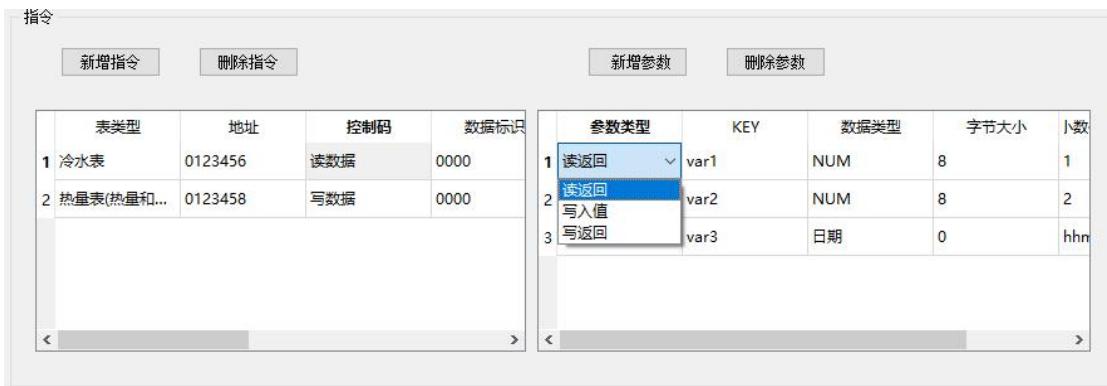
读取配置
保存配置

指令

新增指令
删除指令
新增参数
删除参数

表类型	地址	控制码	数据标识
1 冷水表	0123456	读数据	0000
2 热量表(热量和...	0123458	读数据	0000
读地址 写数据 写地址 写电机同步			

参数类型	KEY	数据类型	字节大小	小数
1 读返回	var1	NUM	8	1
2 读返回	var2	NUM	8	2
3 读返回	var3	日期	0	hhm



## 3.5. 网络传输

### 3.5.1. TCP/UDP 配置

属性	输入说明	解释
SSL 加密	分不加密, 无证书加密, 证书加密三种方式	一般 SSL 加密都是无证书加密, 不需要证书
传输协议	分 TCP 和 UDP	
服务器地址	服务器 IP 或域名	
端口	连接端口	
心跳时间	秒	
心跳包	有需要才填	可以是变量或函数
注册包	有需要才填	可以是变量或函数
前缀	有需要才填	可以是变量或函数
后缀	有需要才填	可以是变量或函数
服务器证书	可以使用现有对象的证书 \${对象 ID}	复制证书文件有效内容即可
客户端证书	可以使用现有对象的证书 \${对象 ID}	复制证书文件有效内容即可
客户端 KEY	可以使用现有对象的 KEY \${对象 ID}	复制 KEY 文件有效内容即可
私钥口令		

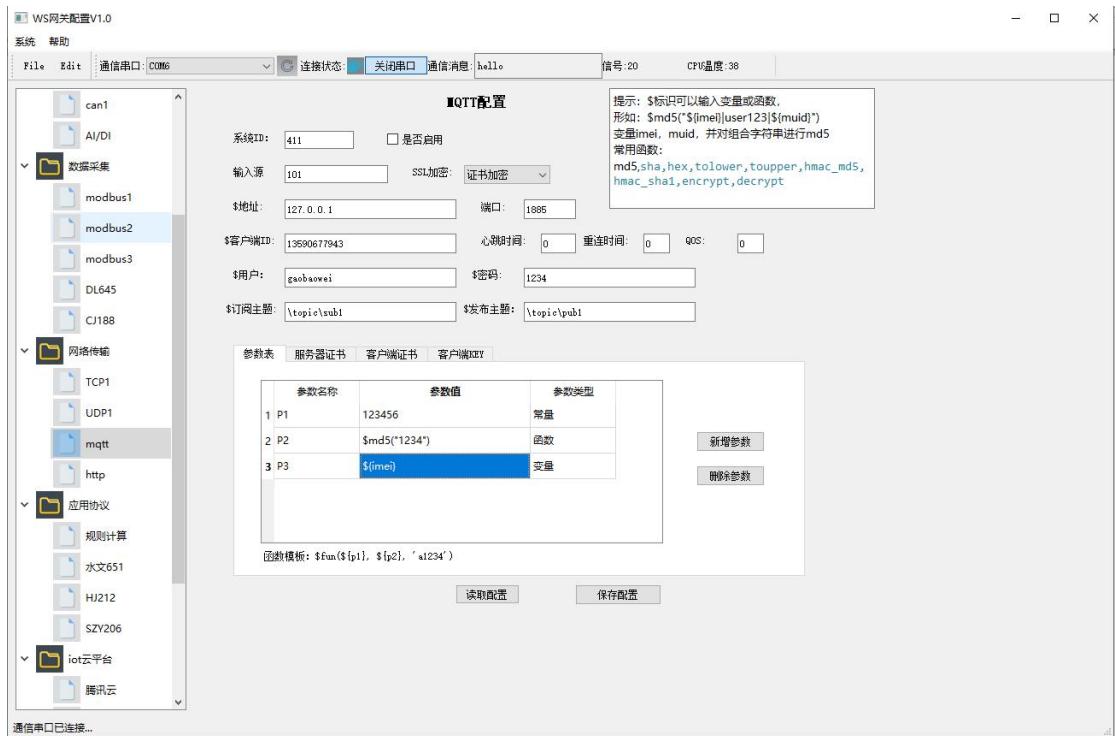


### 3.5.2. MQTT 配置:

#### 1. 基本功能说明:

属性名称	输入说明	解释
加密方式	分不加密, 无证书加密, 证书加密三种方式	一般 SSL 加密都是无证书加密, 不需要证书
地址	连接服务器地址	
端口		
客户端 ID	可以动态输入变量	
用户名	可以动态输入变量	
密码	可以动态输入变量	
订阅主题	网关接收平台数据的通道	
发布主题	网关发送数据到平台的通道	
心跳时间		
重连时间		
Qos	QoS 等级: 0, 1 和 2。	
参数表	根据动态密码验证的需要, 可以自定义常量, 变量和函数	
证书	参考 TCP 证书说明	

#### 2. 基本参数配置:



### 3. 动态密码验证:

客户端 ID, 用户, 密码, 上报主题, 订阅主题等都可以使用变量参数, 需要的参数可以在参数表里进行专门的定义, 参数本身支持变量表达式和函数, 以实现动态密码验证。比如 Jelink 平台的动态密码验证

**MQTT配置**

系统ID:	411	<input checked="" type="checkbox"/> 是否启用
输入源:	201	SSL加密: <input type="button" value="不加密"/>
\$地址:	8.134.75.214	端口: 1883
\$客户端ID:	<input type="text" value="\${imei}"/>	心跳时间: 60 重连时间: 10 QoS: 0
\$用户名:	<input type="text" value="\${user1}"/>	\$订阅主题: <input type="text"/>
\$密码:	<input type="text" value="\${passwd}"/>	\$发布主题: <input type="text" value="/FY104/\${deviceId}/properties/report"/>

参数表

参数名称	参数值	参数类型
1 deviceId	\$(imei)	变量
2 ts13	\$(timestamp13)	变量
3 user1	gbw \$(ts13)	变量
4 passwd	\$(md5(gbw \$(ts13) 123456))	函数

新增参数

删除参数

函数模板: \$fun(\${p1}, \${p2}, 'a1234')

#### 4. 输出 JSON 配置:

Mqtt 输出数据的 json 格式可以自定义: 比如增加时间戳属性字段, 或是把数据放到指定的属性。

表格字段	说明	
属性	上报 JSON 属性, 包含数据采集属性和自定义属性	
数据	可以是变量表达式, 如当前时间戳 \${timestamp} 表示当前秒级时间戳 \$num_str(\${wd},1) 表示把温度 wd 数据格式化 1 位小数	
类型	常量: 固定数字或字符串 变量: \${xxx} 魔法输入 函数: 魔法输入 \$fun_name(p1,p2) 采集: 指上游对象比如 modbus 采集到的数据 公式: 支持公式计算 比如: sd*10+5 表示把 sd 值乘以 10 再加 5	函数参数可以是魔法变量, 比如: \${wd} 表示读取采集的温度变量。 公式计算一般用于采集数据的比例和偏移
路径	指该属性在 json 中的结构位置	

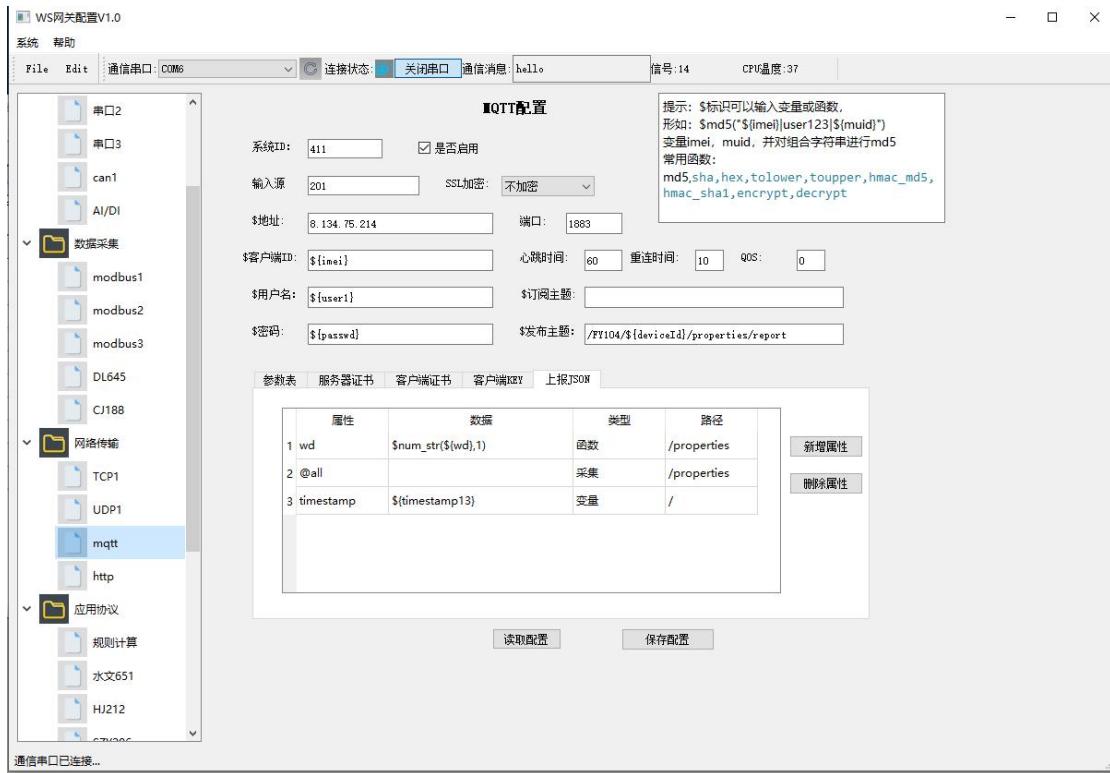
改变属性名或路径: 如果只改变变量在 json 的路径, 属性名为原来的变量

名，数据为空，类型选择采集，输入路径就可以。如果还要使用新的属性名，就在属性名输入新的名称，数据输入原来的变量名。如下图所示：

3 var6		采集	/bw
4 V5	var5	采集	/bw

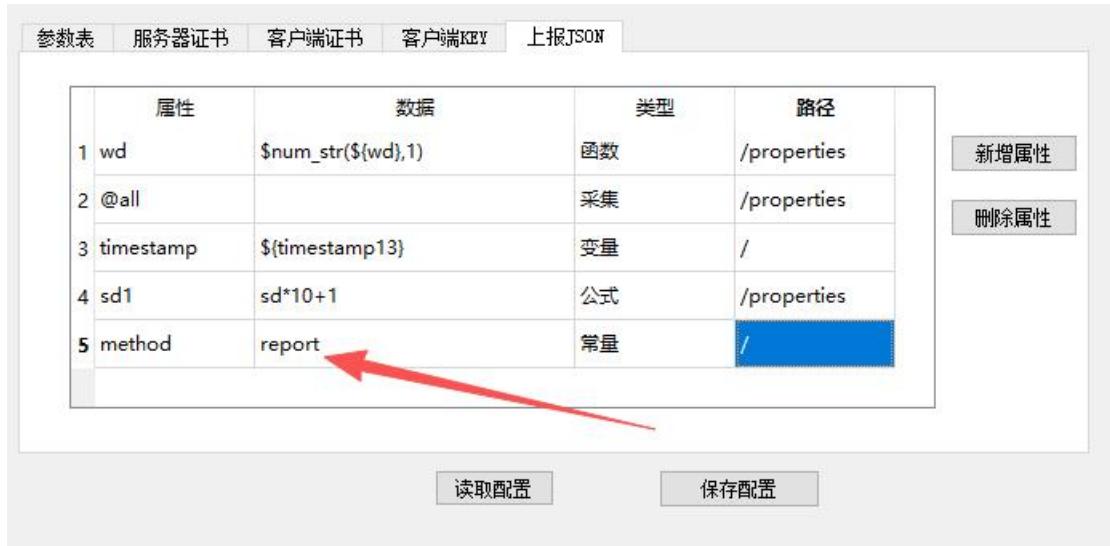
如图所示。



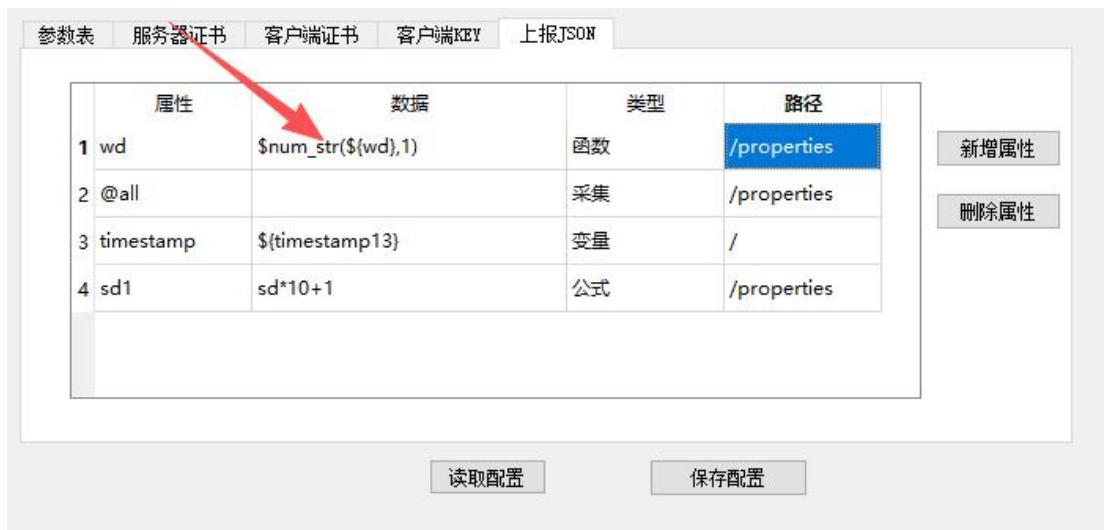


## 5. 常量属性:

有的 IOT 平台需要在上报 JSON 写入固定属性值, 比如腾讯 IOT, 需要写入固定属性 method , 值是 report 字符串



函数属性: 可以使用系统函数对数据进行二次处理, 比如将温度值转换成指定小数位数的 json 属性值后在上传。



属性	数据	类型	路径
1 wd	\$num_str(\${wd},1)	函数	/properties
2 @all		采集	/properties
3 timestamp	\${timestamp13}	变量	/
4 sd1	sd*10+1	公式	/properties

新增属性      删除属性

读取配置      保存配置

计算属性：类型为公式就可以进行二次计算，比如新增湿度属性 sd1，其值将用采集的湿度  $sd*10+1$  进行计算后再上报，一般用于需要比例倍乘和偏移的变量。



属性	数据	类型	路径
1 wd	\$num_str(\${wd},1)	函数	/properties
2 @all		采集	/properties
3 timestamp	\${timestamp13}	变量	/
4 sd1	sd*10+1	公式	/properties

新增属性      删除属性

@all 属性，设置@all 属性表示把输入数据（比如 modbus 采集后的表数据）的所有属性放到指定路径，用这个属性就不用定义每个属性在相同路径的情况。如果某个属性需要单独重新计算，需要把该属性放到@all 属性前面定义，设计规则是后面的属性值不会覆盖前面的属性值。

属性	数据	类型	路径
1 wd	\$num_str(\${wd},1)	函数	/properties
2 @all		采集	/properties
3 timestamp	\${timestamp13}	变量	/

MQTT 动态密码支持配置案例:

Jetlink IOT 平台接入:

Jetlink 协议规定:

在 IOT 平台创建设备时候需要设置 mqtt 认证 的 secureId, 和 secureKey

clientId: 设备 ID

username: secureId+"|"+timestamp

password: md5(secureId+"|"+timestamp+"|"+secureKey)

5d41402abc4b2a76b9719d911017c592

说明: secureId 以及 secureKey 在创建设备产品或设备实例时进行配置.

timestamp 为当前时间戳(毫秒),

与服务器时间不能相差 5 分钟.

md5 为 32 位, 不区分大小写.

按照这个规定我们如下步骤进行 mqtt 配置:

1. 设置 mqtt 对象的基本属性, 服务器地址, 端口, 输入源 201 表示 modbus 采集的串口数据将输入到 mqtt 对象进行处理。
2. 在参数表中, 增加设备 ID 参数 deviceId, 用户参数 user1, 密码参数 passwd, 时间戳参数 ts13, 其中 deviceId 直接去设备的 imei, 时间戳参数计算当前时间戳毫秒, 用户参数字符串拼接, 密码参数用 md5 函数计算用户+密钥 ID
3. 在客户端 ID, 用户名, 密码, 主题等输入参数变量, 格式 \${变量名}
4. 配置输出 JSON

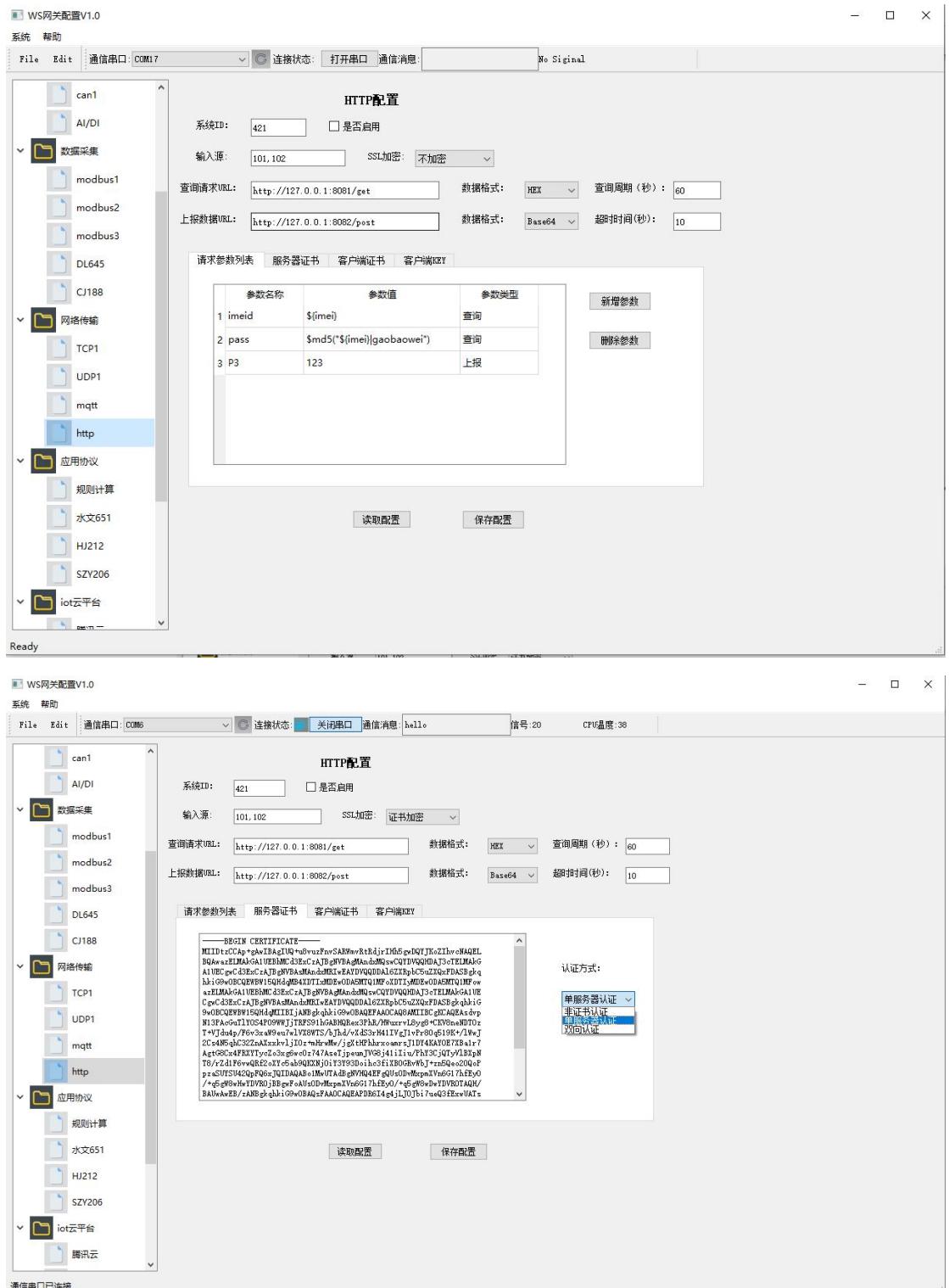
需要配置一个 timestamp 在对象根目录下, 然后将所有采集属性配置到 properties 节点下

data-json:

```
{
  "timestamp": 1739793171517,
  "properties": {
    "wd": 25.5,
    "sd": 60.3
  }
}
```

### 3.5.3. HTTP 对象说明

属性名称	输入说明	解释
加密方式	分不加密, 无证书加密, 证书加密三种方式	一般 SSL 加密都是无证书加密, 不需要证书
查询请求 url:		使用 http GET 方法进行数据查询
上报数据 url		使用 POST 方法进行数据上报,
数据格式	TEXT (文本格式), HEX(16 进制数据格式), Base64 编码格式。	一般选择 TEXT 文本格式。
查询周期		以固定间隔时间 (多少秒) 进行一次查询
请求参数列表		这里的参数配置不管查询请求还是上报请求, 都是 url 地址参数配置。终端会根据这些参数拼接完整的 url 地址。
数据格式		规定使用 http 数据通信格式必须是 json, 把数据放到 tdata 属性, 其他属性用户自定义, 不管上行, 还是下行, 所有数据必须在 tdata 属性中 请求参数列表里不能占用 tdata 属性, tdata 里的内容为多包数组, 每包可能是纯文本, 或 json 对象, 由用户去定义, 比如串口串口采集的二进制数据可能是 hex 或 base64 数据
证书配置		: 如服务器需要证书认证才需设置, 一般不用。证书内容可以直接复制证书文本到编辑框, 注意证书文件内容比较多, 除非必要, 非必要请勿设置。在 socket , http,mqtt 都有证书配置, 如果是同一证书, 只需配置一个即可, 其他有需要用证书的只需要在证书内容如输入 \${对象 ID} , 这里的对象 ID 就是配置好证书的那个对象的 ID, 比如 \${401} 就是共用 TCP 对象的证书。



### 3.5.4. ModbusSlave 从站配置

Modbus slave 做从站可以对各接口采集的数据进行汇聚，然后以 modbus 从站协议方式提供给其他主站调用。同时主站可以集中下发数据到次从站，从站可

可以把数据转发到 Modbus 对象实现个传感设备的写入数据功能。Modbusslave 作为从站可以用串口和 TCP/UDP/MQTT 进行数据通信，数据协议包 ModbusRTU 和 ModbusTCP 都支持。

操作流程指引：

### 1. 输入源设置

如果要把 Modbus 通过串口采集的各数据汇聚到 modbus 从站对象，就必须设置 modbusSlave 对象的数据输入源，比如 201 表示 modbus1，还可以是 can 等其他接口采集的数据输入，多个对象用逗号分割就可以。汇聚原理是根据输入源的各数据属性，按照本对象的映射表直接映射到虚拟寄存器中。

### 2. 配置寄存器

根据对端 modbus 主站要求设置对应的寄存器，把输入对象的数据源都存在定义的虚拟寄存器中。

### 3. 配置变量映射

变量映射和 modbus 对象的变量映射是一样的原理。



## 3.6. 应用层协议

### 3.6.1. 规则配置：

规则配置可以对数据进行过滤，二次计算，自定义数据解析，再次封装应用层协议包等强大功能。下面是功能表说明：

触发规则	输入说明	解释
1. 时间触发	按时间计划 CRON 表达式触发,可以定时，指定时间点计划，按秒，分，时，日，月，周 六个规则组合。 CRON 表达式详见专门说明。	

2. 条件触发	<p>可以根据需要配置自己的触发条件和逻辑表达式, 支持大于、小于、等于等比较符, 也支持并且、或者等逻辑运算符。比如 变量 <code>a &gt; 9 or b &gt; 10</code>, 如果变量值介于 两个数之间, 用 <code>between</code>, 比如变量 <code>c between 10 40</code></p> <ol style="list-style-type: none"> <li>1. 单数比较 <code>var1 &gt; a</code></li> <li>2. 双操作数 <code>var1 between a b</code></li> <li>3. 多数 in <code>var1 in a,b,c</code></li> </ol> <p><code>var0 &gt; 10 and var1 &gt; 20 and var2 in (10,20,30) and var3 between (1,10)</code></p>	
3. 自定义函数触发	<p>输入: Lua table 类型 <code>tb</code>, <code>tb</code> 属引用参数, 修改 <code>tb</code> 的熟悉将直接改变主程序的值。</p> <p>输出: bool, 举例:</p> <pre>function(tb)     if tb.key1 == 1 then         return true     else         return false     end end</pre>	
变量	<p>规则定义中的变量就是上游节点对象输入数据的 json 属性变量, 比如上游节点是 <code>modbus</code> 对象。</p>	
二次计算	<p>如果需要对变量进行二次计算, 则可自定义计算公式进行二次计算, 二次计算的新变量可以输出或做逻辑运算比较。</p>	
二次解析	<p>自定义执行函数可以对输入源数据进行协议解析, 或再次封装其他协议包, 然后通过消息发送给其他对象进行处理。</p> <p>这是一个非常灵活有用的功能, 由于二次开发, 适合 <code>lua</code> 开发人员应用。</p>	

## 条件触发

注意：

二次计算：

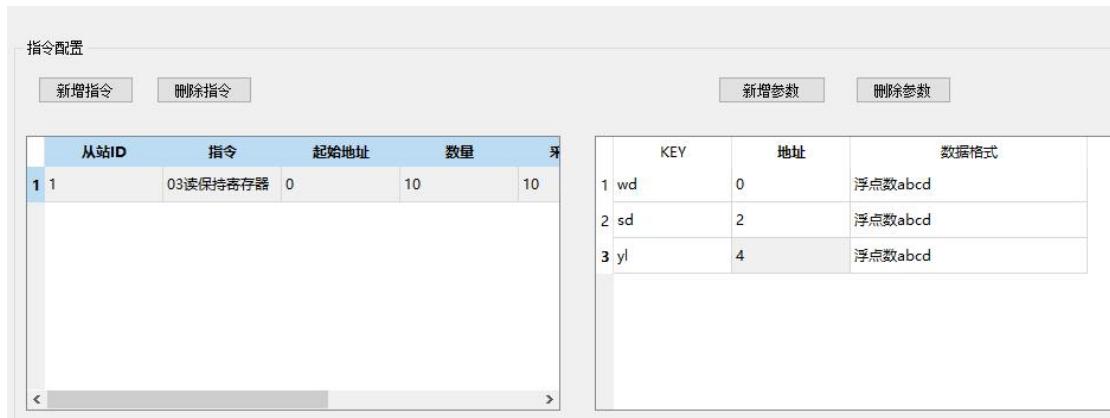
属性	输入说明	备注
新增变量	新变量标识, 比如 var_new1	
计算公式	支持常用计算表达式, 包括常用运算符和括号例如: "(a-b)*c + 1000 + #name * 10" #name 表示 name 变量字符串的长度	
计算方式	触发前：触发前计算的变量可以写入规则表达式条件中 触发后：一般指满足某种条件规则后才进行的新的计算表达式。	

自定义执行函数

属性	详情	备注
自定义函数参数	function(self, tb) self 传当前对象参数, tb 数据表 自定义规则函数和规则触发后的函数都可以修改 tb , 如果规则触发自定义函数处理, 需要自己发送消息 self: send_msg({cmd="tb", data=tb})	自定义执行函数是满足某种条件的规则触发时候, 终端再次执行的动作, 比如发送消息给其他模块对象。

规则配置案例：

1. 配置 modbus 数据, 分别是温度, 湿度, 压力三个数据对应的 key 为 wd,sd,yl



2. 输入源，这里选择 modbus 对象 201 作为规则对象的输入源，表示 modbus 采集的数据要输出到规则对象进行处理。



3. 规则触发：有三种触发规则，分别时间触发、条件触发、函数触发  
 时间触发  
 条件触发：

### 规则计算配置

系统ID:   是否启用

输入源:  CRON表达式:

条件触发  函数触发  二次计算  自定义执行函数

(wd>30 and sd >60)



多种与、或、in、between条件逻辑组合

函数触发:

### 规则计算配置

系统ID:   是否启用

输入源:  CRON表达式:

规则触发  函数触发  二次计算  自定义执行函数

```
function(self, tb)
tb.V100=100
if tb.wd>30 and tb.sd>60 then
    return true
else
    return false
end
```

函数里面的变量需要加参数引用tb.wd

读取配置

## 二次计算:

可以对 modbus 采集数据进行比例和偏移计算。

公式可以是 lua 支持的各种数学计算公式。

规则触发  函数触发  二次计算  自定义执行函数

新增变量  删除变量

变量	计算公式	计算方式
1 wd1	$(wd+1)*10$	触发后
2 sd	$sd+10$	触发前

公式可以是 LUA 支持的各种计算公式

读取配置  保存配置



### 自定义函数：

自定义函数不但可以定义规则触发后执行逻辑，还可以对数据进行二次解析或重新封装。



buf 参数就是输入数据的原始二进制数据，比如串口采集的二进制数据，可再次进行自定义协议解析，解析后的数据放到 tb 对象属性中。

**规则计算配置**

系统ID:   是否启用

输入源:  CRON表达式:

[条件触发](#) [函数触发](#) [二次计算](#) [自定义执行函数](#)

```
function(self, tb, buf)
tb.V101=101
log.info("99999999999999999999",buf:toHex())
tb.wd=33.1
tb.sd=66.2
tb.yl=66.1
local msg1 = {}
msg1.cmd = "tb"
msg1.data = tb
self:send_to(msg1,411)
return tb
end
```

执行函数

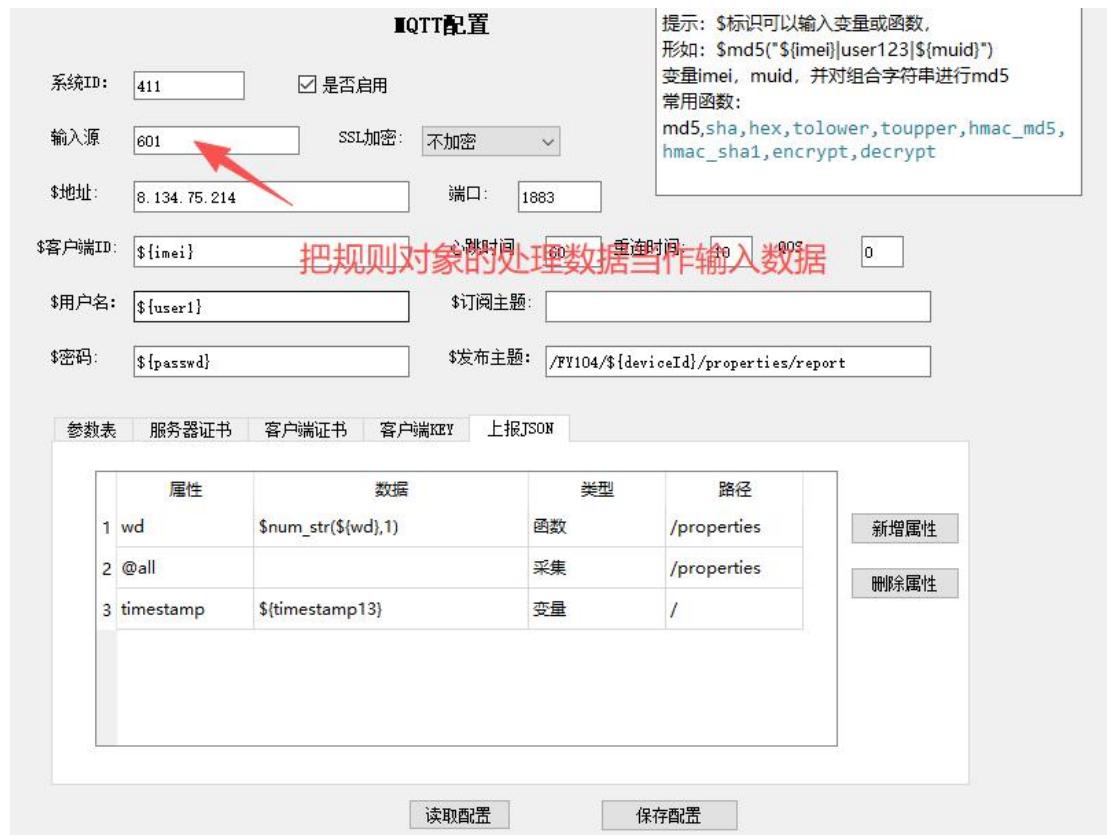
自定义函数里面可以发送数据消息给指定其他对象

[读取配置](#) [保存配置](#)

发送消息：

在自定义执行函数中发送消息，比如发送到 mqtt 对象，id 为 411，先定义好消息数据包，然后调用 `send_to` 函数，注意必须加上 `self:` 引用。

```
local msg1 = {}
msg1.cmd = "tb"
msg1.data = tb
self:send_to(msg1,411)
```



### 自定义协议解析实现：

用户可以自定义协议解析串口数据，比如解析摄像头控制数据。

参数说明：self 对象引用参数，tb 输出表参数，buf 输入二进制数据

说明：该功能适合简单协议解析，如果协议特别复杂，需要升级固件来实现。

```
function(self,tb,buf)
tb.V101=101
log.info("99999999999999999999",buf:toHex())
tb.wd=33.1
tb.sd=66.2
tb.yl=66.1
local msg1 = {}
msg1.cmd = "tb"
msg1.data = tb
self:send_to(msg1,411)
return tb
end
```

### 自定义协议解析：



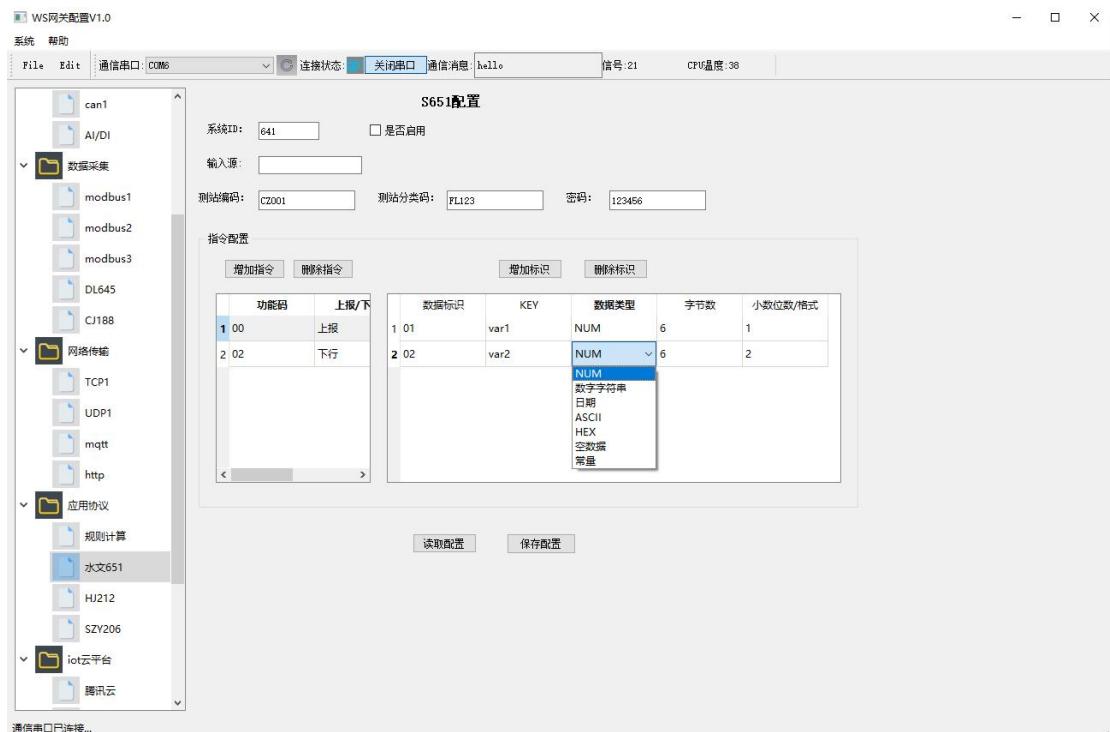
### 3.6.2. S651 协议配置

“水文 S651 协议”中的 S651 指的是中华人民共和国水利行业标准《SL 651-2014 水文监测数据通信规约》。用于水利水文、环境监测等领域。

这是一个在中国水文领域至关重要的技术标准，用于规范水文监测站点与数据中心之间数据传输的格式、内容和流程。SL 651 是一个“说好的规则”，它确保了全国各地、不同厂家生产的水文监测设备（如雨量计、水位计、流量计等）都能用同一种“语言”向数据中心“汇报”数据。

配置属性	输入说明	备注
测站编码	应用平台统一编码 (HEX) 遥测站地址编码由 5 字节构成 (A5~A1)，其中 A5 为高位字节，A1 为低位字节。	
测站分类码		
密码	密码为 2 字节 HEX 码，由中心站生成	

配置属性	输入说明	备注
功能码	参照 S651 协议查表	
上报、下行	上报是只数据采集上报； 下行是指下行控制命令，比如阀门控制	



### 数据配置说明：

变量属性	输入说明	备注
数据标识	参照 S651 协议查表	
KEY	变量标识	
数据类型	<p>数据类型：</p> <p>0) NUM BCD 数值型 N(D, d) ; D 是数据位数, d 是小数位数</p> <p>1) BCD 数字, NNNN ; 大数据当字符串去取, 下面就不计算成数值型, 比如 “99999999.99”</p> <p>2) DATE 日期 , YYMMDDhhmmSS</p> <p>3) ASCII ,AAAA</p> <p>4) HEX 十六进制数据</p> <p>5) 空数据, NIL</p> <p>6) 常量数据 此时 opt 为常量值(HEX) 用于定义特别</p>	

变量属性	输入说明	备注
	标识等数据	
字节数	变量数据在 S651 协议包中所占数据字节数	
小数位数 /格式	小数位数, 如果是日期类型, 就是日期类型格式	

### 3.6.3. HJ212 协议配置

所有的通讯包都是由 ASCII 码（汉字除外，采用 UTF-8 码，8 位，1 字节）字符组成。

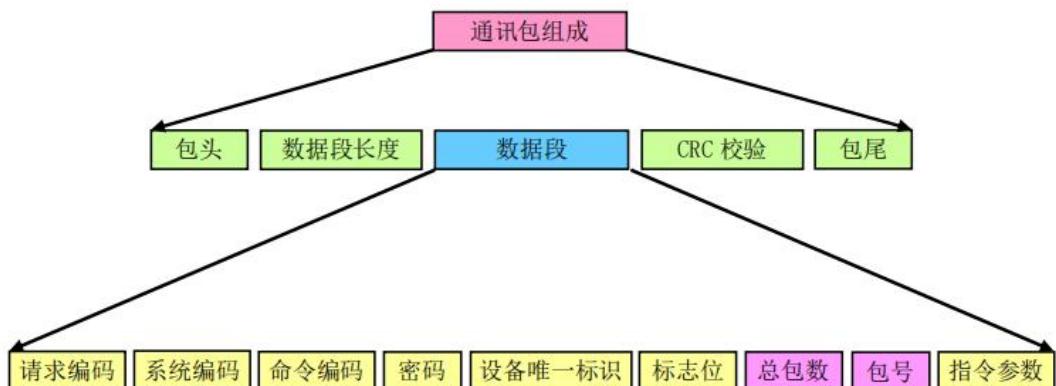


图 4 通讯协议数据结构

#### 1) 基础配置参数:

分类	参数名称	输入说明	备注
基础配置	系统编码	HJ212	
	设备标识	可以用 \${imei} 自动获取 ID	
	密码		
	标识位	根据上报要求填写	
	缓存记录	根据时间需要设置数据缓存, 最多 3 天。	
命令配置	命令编码	自动选择上报实时, 分钟, 小时, 日数据命令	
	统计类型	边缘计算统计分钟, 小时, 日 统计指标: 最小值, 最大值, 平均值, 累计值, 记录总数	
	上报周期	用 cron 表达式: 规定上报时间计划	
上报因子	变量因子	对应 212 协议要求的上报参数	
	KEY	对应 modbus 的采集变量	

#### 2) 系统编码表:



表 5 系统编码表 (引用 GB/T16706-1996 )

系统名称	系统编码	描述
地表水质量监测	21	
空气质量监测	22	
声环境质量监测	23	
地下水质量监测	24	
土壤质量监测	25	
海水质量监测	26	
挥发性有机物监测	27	
大气环境污染源	31	
地表水体环境污染源	32	
地下水体环境污染源	33	
海洋环境污染源	34	
土壤环境污染源	35	
声环境污染源	36	
振动环境污染源	37	
放射性环境污染源	38	
工地扬尘污染源	39	
电磁环境污染源	41	
烟气排放过程监控	51	
污水排放过程监控	52	
系统交互	91	用于现场机和上位机的交互

### 3) 因子表:

参考 HJ212 协议文档。

#### 4) 缓存注意事项:

如果不需要缓存, 尽量不要设置缓存, 缓存需要存储空间, 我公司的产品带 M 的才支持缓存, 比如 FY104M. 设置 0 缓存, 还是可以统计分钟数据, 小时小数据并上传。只是不保存历史数据。

### 3.6.4. SZY206 协议配置

206 协议主要是用来向平台发送采集的数据, 通过 MODBUS-RTU 采集各传感器数据, 按照 206 协议通过 TCP 上报网关终端作为测站 (从动站), 由平台统一分配一个测站地址平台作为主动站, 发起请求, 控制命令网关多数为自报帧, 向平台发送定制的数据

该模块传感器->串口输入->Modbus 对象: joon 数据->206 对象: 协议包->到 TCP 网络对象->应用平台

数据类型及格式:

- 0) NUM BCD 数值型 XXX.XXX ;
- 1) BCD 数字, NNNN ; 大数据当字符串去取, 下面就不计算成数值型, 比如“99999999.99”
- 2) DATE 日期, ssmmhhDDMMYY smhDMY 各种双组合
- 3) ASCII ,AAAA ;
- 4) HEX 十六进制数据
- 5) 空数据, NIL

功能码	自报参数:	字节数	小数位数	单位
1	雨量	3	1	mm
2	水位	4	3	m 4*N
3	流量 (或水量)	5	3 或 0	m^3/s 5*N 瞬时流量, 有小数
4	流速	3	3	3*N 高 4 位表示正负 0H 正 FH 负
5	闸位	3	3	m/s 3*N
6	功率	3	0	kw 3*N
7	气压	3	0	
8	风速 (含风向)	3	2	高 4 位为方向 0~8
9	水温	2	1	
10	水质			5+4*N+1
11	土壤含水率	2	1	2*N
12	蒸发量	3	1	3*N
14	统计雨量	1+3	1	
15	水压	4	2	4*N



### 3.7. IOT 云平台配置

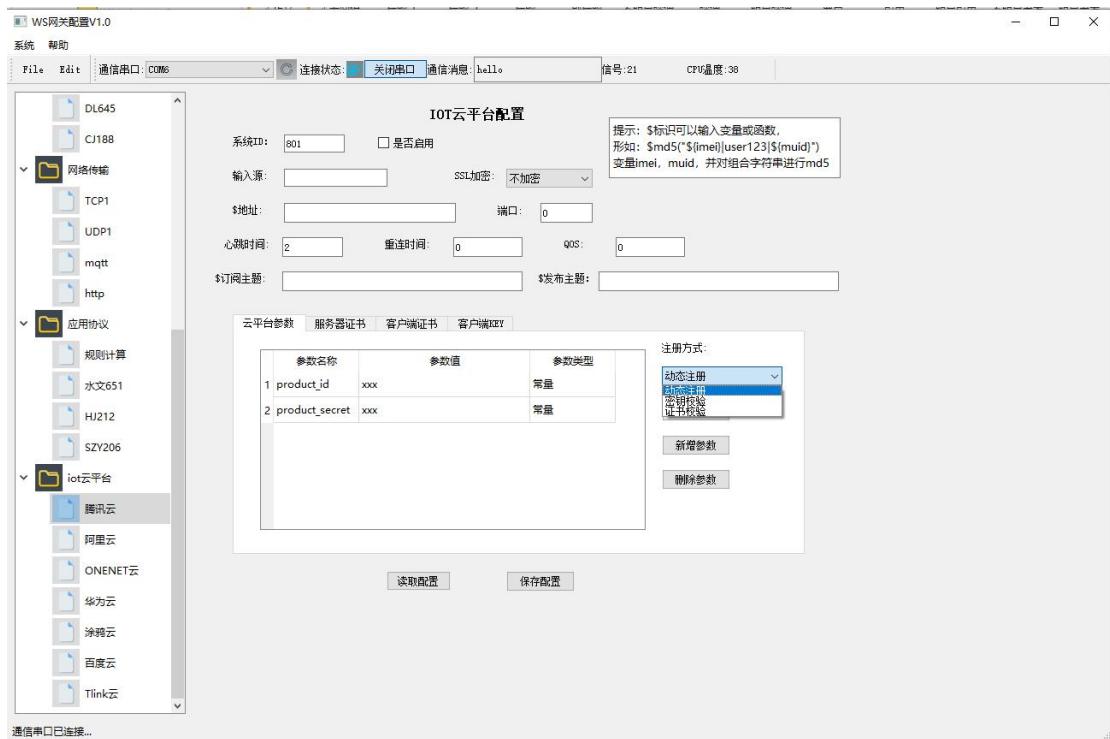
IOT 云平台，各云平台以各种注册类型进行校验，但都是基于 mqtt 协议进行通信，各云平台进行注册认证的参数不同，系统提供了各云平台的默认设置参数，点击重设参数按钮即是设置各种注册类型的默认配置参数，注意可能因云平台版本升级更新，参数设置会有所变化。

云平台支持及注册类型

编号	云平台:	注册类型:
1	腾讯云	{"动态注册", "密钥校验", "证书校验"}
2	阿里云	{"一型一密(免预注册)", "一型一密(预注册)", "一机一密(预注册)"}
3	ONENET 云	{"动态注册", "一型一密", "一机一密"}
4	华为云	{"动态注册(免预注册)", "密钥校验(预注册)"}
5	涂鸦云	{"密钥校验"}
6	百度云	{"密钥校验", "证书校验"}

编号	云平台:	注册类型:
7	Tlink 云	{"密钥校验", "证书校验"}

### 3.7.1. 腾讯云:





### 3.7.2. 阿里云:



### 3.7.3. ONENET 云:



### 3.7.4. 华为云:



### 3.7.5. 涂鸦云:



### 3.7.6. 百度云:



### 3.7.7. TLink 云:



## 3.8. 系统配置

### 3.8.1. 菜单配置

进入程序主菜单，点击系统菜单，点击功能配置子菜单，如下图所示：

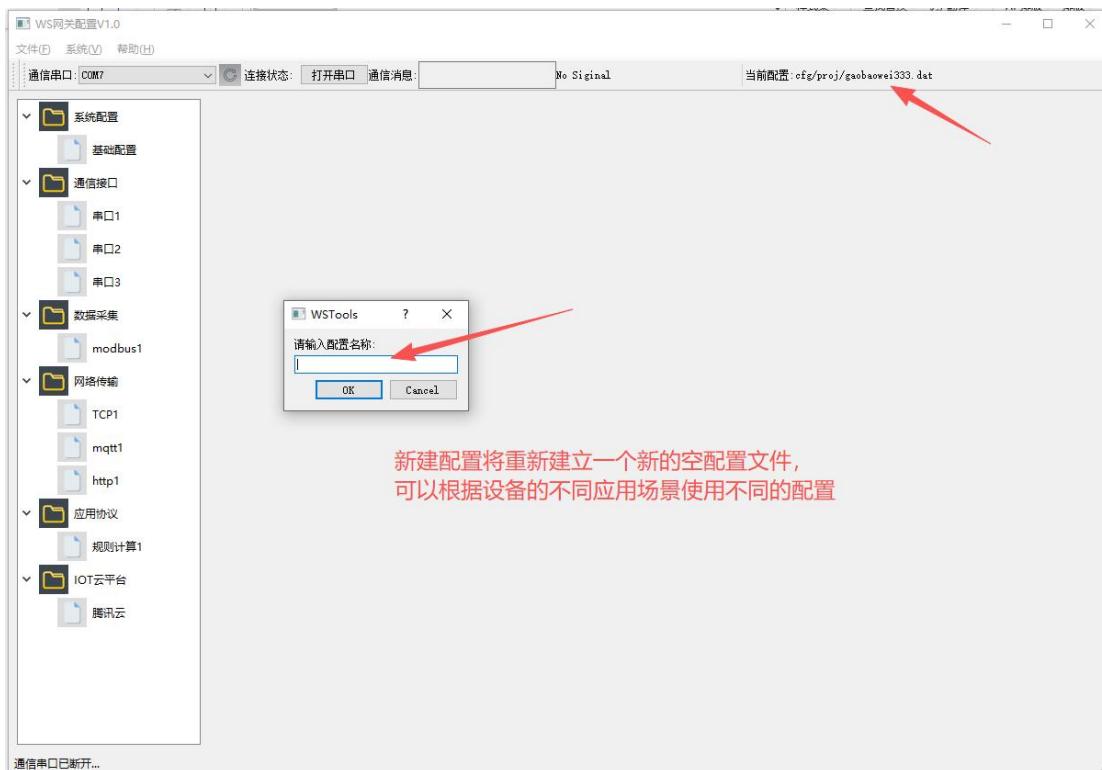


卫软科技网关将数据采集到计算解析到应用分层处理，各层有对应的模块对象实现相应功能，根据实际需要，勾选对应的功能模块，直接启用或不启用即可以。



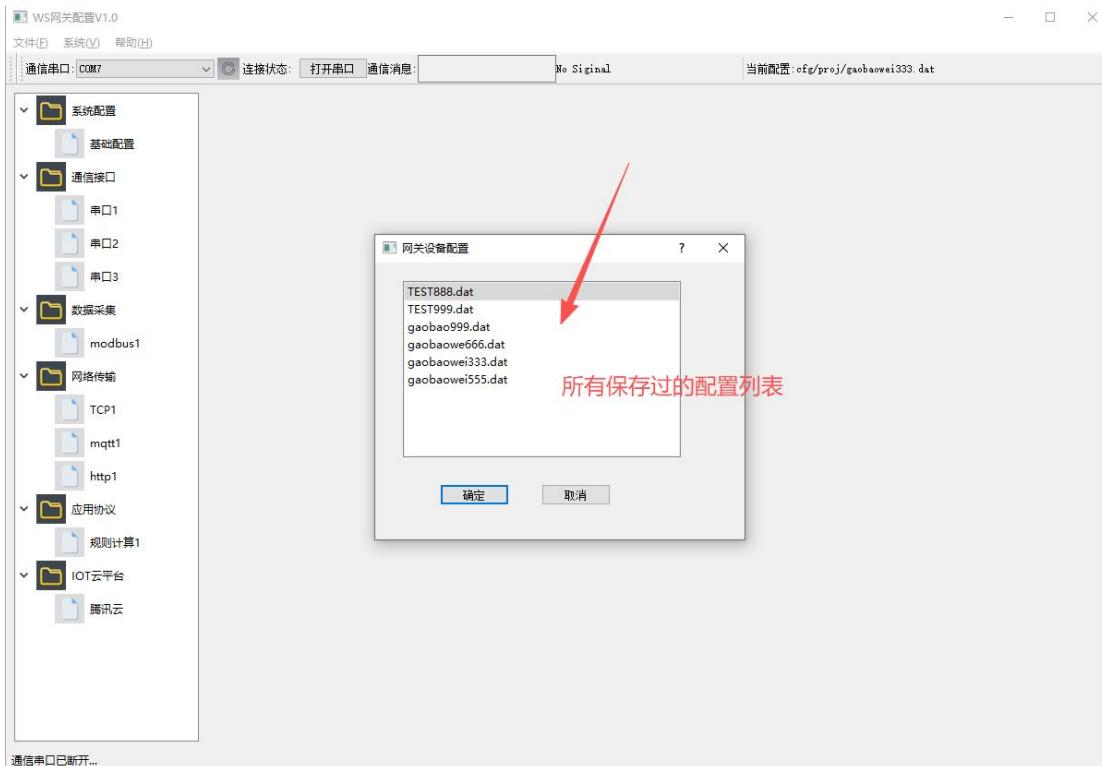
### 3.8.2. 新建配置

在功能模块配置完毕后，就可以根据那些选定的功能模块来创建新的配置。



### 3.8.3. 打开配置

打开之前配置过的配置文件，对多个设备分类配置用此功能。



---

下载固件：

### 3.8.4. 另存配置

网关配置保存后会自动保存到当前配置文件，如果需要把当前配置另存，点击系统菜单，另存配置，并自动切换到新的配置。该功能主要方便配置的历史保存和后期重新加载。

## 4. 附录：

### 4.1. 本地下载固件

卫软科技产品一般都不需要本地下载固件，建议直接在线升级。

### 4.2. 附录 1：Cron 表达式 详细说明

Cron 表达式是一个字符串，由 6 个或 7 个字段组成（取决于 Cron 系统的实现），字段之间用空格分隔。它用来配置定时任务在何时执行。

标准的 Unix/Linux Cron 表达式有 5 个字段，而许多现代库（如 Spring Quartz）扩展了它，增加了 秒 和 年 字段，成为 6 个或 7 个字段。“秒，分，时，日，月，周”是 6 字段表达式，这也是目前非常常见的一种格式（例如在 Spring @Scheduled 注解或 Quartz 调度器中）。

---

### 4.2.1. 6 字段 Cron 表达式格式

字段	含义	允许的值	允许的特殊字符
1	秒	0-59	, - * /
2	分	0-59	, - * /
3	时	0-23	, - * /
4	日	1-31	, - * ? / L W
5	月	1-12 或 JAN-DEC	, - * /
6	周 (0 和 7 都代表星期日)	0-7 或 SUN-SAT	, - * ? / L #

注意“日”和“周”字段的互斥性：由于通常不需要同时指定具体的“几号”和“星期几”，为了避免冲突和歧义，通常会对这两个字段设置一个规则：其中一个字段必须被设置为 ? (表示不指定)。这是很多 Cron 实现（如 Quartz）的常见约定。

### 4.2.2. 特殊字符详解

字符	名称	含义与示例
*	星号	表示每一个值。例如，在“分”字段使用 *，表示“每分钟”。
?	问号	表示不指定值。仅用于“日”和“周”字段，以解决这两个字段的互斥性。

字符名称	含义与示例
<b>-</b> 连字符	表示一个范围。例如，在“时”字段使用 <code>10-12</code> ，表示“10 点， 11 点， 12 点”。
<b>,</b> 逗号	表示列出多个值。例如，在“周”字段使用 <code>MON,WED,FRI</code> ，表示“周一，周三，周五”。
<b>/</b> 斜杠	表示步长。例如，在“秒”字段使用 <code>0/15</code> ，表示“从第 0 秒开始，每 15 秒一次”（即 0, 15, 30, 45 秒）。在“分”字段使用 <code>*/10</code> ，表示“每 10 分钟一次”。

### 4.2.3. 常用示例

表达式	说明
<code>0 0 12 * * ?</code>	每天中午 12:00:00 执行
<code>0 15 10 ? * *</code>	每天上午 10:15:00 执行
<code>0 0/5 14 * * ?</code>	每天下午 2 点开始，到 2:55 结束，每 5 分钟执行一次(14:00, 14:05, ..., 14:55)
<code>0 0 18 ? * MON-FRI</code>	每周一到周五的下午 6:00:00 执行
<code>0 0 10,14,16 * * ?</code>	每天上午 10 点、下午 2 点、下午 4 点各执行一次
<code>0 0 12 1 * ?</code>	每月 1 号的中午 12:00:00 执行
<code>0 15 10 L * ?</code>	每月最后一天的上午 10:15:00 执行
<code>0 0 8 1W * ?</code>	每月 1 号，如果 1 号是周末，则延至最近的工作日上午 8:00:00 执行

---

表达式	说明
<code>0 0 10 ? * 6#3</code>	每月的第三个星期五上午 10:00:00 执行
<code>*/10 * * * * ?</code>	每 10 秒执行一次 (注意: 这是 6 字段格式, 5 字段格式不支持秒)

---

#### 4.2.4. 5 字段 Cron 表达式 (传统 Unix Cron)

如果你看到的是 5 字段的表达式, 它省略了“秒”字段, 并且“年”字段通常也不存在。格式为: 分 时 日 月 周。

字段	含义	允许的值
<b>1</b>	分	0-59
<b>2</b>	时	0-23
<b>3</b>	日	1-31
<b>4</b>	月	1-12
<b>5</b>	周	0-7 (0 和 7 是星期日)

**5 字段示例:**

`0 0 * * *` 每天 0 点 0 分执行

`*/15 * * * *` 每 15 分钟执行一次

