IOTGateway 使用手册(v2.6.16)

v1.7 版本更新（2023.8.7）

序号	章节	说明
1	1.3	增加了软件升级说明
2	3.2	增加了 htmlview.html 说明
3	3.9	增加了 SSL 通讯说明
4	3.10	增加了 WebSocket 通讯说明
5	4.5	增加了 JS 操作关系库说明
6	附件 1	增加了 JS 操作数据库函数

v1.8 版本更新（2023.8.19）

序号	章节	说明
1	2.4.4	增加了电子邮件发送
2	2.4.5	增加了微信发送
3	7.1-7.6	日志查询界面更新，增加了筛选查询

v1.8.1 版本更新（2023.9.2）

序号	章节	说明
1	附件 6	增加了海康威视视频演示（支持摄像机和硬盘录像机）
2	2.6 4.4	更新变量写入权限

v1.8.5 版本更新（2023.12.1）

序号	章节	说明
1	5.1	增加 InfluxDB2 历史归档，增加智方实时数据历史归档
2	5.3	增加 InfluxDB2 报表归档归档
3	8.13	增加智方实时数据库驱动

v1.8.6 版本更新（2023.12.7）

序号	章节	说明
1	8.14	Windows 增加 OPDA 驱动
2	2.1	数据库连接字符增加加密功能
3	8.15	DLT645 驱动
4	8.16	DTU 驱动

v1.8.7 版本更新（2024.1.3）

序号	章节	说明
1		报警更新，在线修改报警变量为不报警，立即删除报警控件内容，停止驱动后相关变量立即从报警控件删除
2		定时调度和报警事件更新
3	7	日志查询更新，增加 url 变量筛选
4		行表列表归档查询界面更新
5		历史归档查询更新
6		实时数据库界面更新，增加布尔量切换菜单
7	2.1	系统配置页面更新
8		删除大部分系统配置表格的 ID 列

9		修复扩展读取没有判断 Enable 为 True 的问题

V2.0 版本更新（2024.2.4）

序号	章节	说明
1	8.17	增加唐码实时库驱动
2	7	日志查询权限调整为 10
3	2.4.7	增加 Modbus 从站扩展
4	2.4.6	增加爱普生票据打印机扩展
5	附件 7	增加 InfluxDB 安装说明
6	1.2 （5）	增加欧拉系统安装说明
7	2.1	后端增加语言设置，支持中文和英文
8		前端增加语言支持，中文和英文
9		修复驱动采集重试次数不起作用的问题
10		修改多变量调用历史趋势显示无效的问题
11	5.1	增加唐码实时库历史归档

V2.1 版本更新（2024.2.14）

序号	章节	说明
1	2.1	增加 NTPServer
2	2.2	增加 UDPService
3	2.4.8	增加 OPCUA 服务器扩展
4	2.4.9	增加 TCP 服务器扩展
5	8.18	增加 TCP 驱动
6	2.1	增加整个项目 zip 格式导入和导出
7		openGauss（华为高斯）归档数据库适配完成
8		openGauss（华为高斯）项目数据库和日志数据库适配完成

V2.1.2 版本更新（2024.3.7）

序号	章节	说明
1	5.1	历史库增加 SQLServer 和 MySQL 支持
2	附件 1	更新函数
3	附件 2	更新函数

V2.1.3 版本更新（2024.3.10）

序号	章节	说明
1		增加批量写入函数 增加全局 js 函数文件设置和启动、停止 js 脚本支持
2		调整模拟驱动到最后一个启动的驱动
3	附件 10	增加欧拉系统 MySQL 安装文档
4		

V2.1.4 版本更新（2024.3.30）

序号	章节	说明
1	8	增加系统变量功能说明
2	5.5	调整关系库定时删除的执行时间
3	8.7	更新网络接收驱动说明
4		更新 database.html，变量读取增加等待图标
5		更新报表归档查询界面，查询过程禁用按钮

6		更新历史归档查询界面，查询过程禁用按钮，增加变量选择功能
7	2.1	增加全局 JS 脚本设置，用于组态启动和停止时执行
8	8.11	更新 Http 驱动文档
9		修复 modbus 串行驱动无法启动的问题
10	5.2	关系库同步名称包括@时 从表中直接读取变量进行更新，不再使用配置的变量
11	5.3，5.4	列表归档和行表归档增加自定义系统设置用于控制归档规则

V2.3 版本更新（2024.4）

序号	章节	说明
1		调整多租户画面组态功能 使用 HTML5 设置的组态权限控制进入 designer.html 权限 保存权限控制用户保存组态页面 默认设置组态和保存权限为 10 更新实时系统日志显示速度
2		增加 JYC311（静远电子）网口和串口短信模块支持
3	5.3	增加多设备存储到 1 个列表归档的配置说明
4	2.9 8.1	增加 C# RunTime dll 支持 加载、启动、停止函数支持，模拟驱动函数支持
5		历史归档，报表归档更新周期写入定时器精度 SQLServer，MySQL 历史归档定时写入设置启用 处理部分归档停止再运行无法查询的问题
6	2.1	增加复杂密码设置 增加密码有效期设置
7	1.3	项目文件结构升级，多项目支持，每个目录 1 个存储 1 个项目

V2.5 版本更新（2024.10）

序号	章节	说明
1	8.21	增加 TCPClient 驱动,支持 JS 交互
2	8.22	增加用户驱动,支持 JS 解析和反向写入
3	附录 12	增加 rest，udpClient,tcpClinet,webSocketClient,mqttClient js 访问功能
4	8.11	http 接收驱动增加 JS 支持，支持反向写入
5	5.1 附件 10	MySQL 连接字符串增加说明

V2.6 版本更新（2024.12）

序号	章节	说明
1	2.4.7	Modbus 服务器增加本地配置设置
2	2.4.10	MQTT 转发增加本地配置设置
3	2.4.1	TCPIO 转发增加本地配置设置
4		实时数据库界面浮点数小数个数按变量配置显示
5		更新驱动和扩展配置界面，自动增加新增的配置项
6	附件 1	增加通过 JS 控制设备的启停演示代码
7	附件 13	增加扩展 WebApi 接口

V2.6.3 版本更新（2025.2）

序号	章节	说明
1		修复后台管理变量查找不正常的问题，修复变量检查错误
2		更新部分内置组态控件问题，增加 displayNew 函数 新窗口打开页面
3		Web 组态增加预定义文本、线条和填充颜色设置
4		增加 Web 页面在线多语言切换功能 相关文本显示控件增加 8 种语言设置，0 为默认语言文字 setLang 函数 或者 url 增加 langid=1（0-7） 参数
5		更新 OPCUA 驱动
6		日志显示和查询页面增加导出功能（仅导出当前显示页）

V2.6.7 版本更新（2025.5）

序号	章节	说明
1		Web 增加图形控件删除功能
2		InfluxDB 查询增强（最大 最小特征加入）
3		Web 标签替换功能增强，标签统计功能增加保存按钮
4		Web 文本和对象查找功能增强
5		更新用户导入和导出功能，用户权限配置功能增强
6		Web 修复页面过小时中键平移问题
7		Web 页面复制黏贴对象命名规则更新
8		SQLite 报表归档 Time 自动转换为 Time 格式
9		变量驱动地址长度从 64 调整到 256
10		报警级别大于 2 播放语音报警（循环）

V2.6.9 版本更新（2025.8）

序号	章节	说明
1		后台画面设置按更新时间倒序排列
2		报警画面增加语音报警测试功能
3	附件 14	scripts 目录加入 qrcode.min.js
4	2.1	系统设置增加 UpdateUrl 和 UpdateToken 设置 多个系统之间远程画面同步和发布
5	附件 14	动态二维码生成和显示
6		网络接收驱动更新 （1）首次统一设置为公用组，驱动配置可以修改 （2）除首次后不更新变量描述信息，驱动配置可修改 增加接收数据变量单位 Kb
7		圆形指示灯控件更新
8		按钮用于弹窗窗口时增加 窗口尺寸和位置属性设置 画面导航向导按钮外观更新
9	附件 2	scada.js 增加从远程同步画面到本地函数 updateJsonFileFromRemote()

V2.6.10 版本更新（2025.8）

序号	章节	说明
1		变量 Excel 导入不再判断驱动名和设备 ID 必须相同
2	2.4.3	定时调度设置变量功能扩展，支持变量引用和表达式计算

3	8.23	Logix5000 驱动
4	8.24	ConnectedCIP 驱动
5	8.25	和利时 PLC 驱动

V2.6.11 版本更新（2025.9）

序号	章节	说明
1	8.26	AB SLC500&Micro 驱动
2	3.1.1	HTML 控件支持变量绑定数据更新脚本，element 属性是原生对象 Script 控件支持变量绑定数据更新脚本
3	3.1.2	HTML 控件支持 HTML5 input 控件 text，number，datetime，time，range，color，select，button
4	3.1.3	变量数据控件增加 html 属性,通过 html 控件绑定点击显示内置输入控件，支持回车写入，element 属性是原生对象
5		变量浏览器重新加载功能修复 内置一些 HTML5 控件 css 配置，用于按钮和 Input 控件 增加输入控件演示画面
6	3.2	跨域登录方法

V2.6.12 版本更新（2025.11）

序号	章节	说明
1		日志数据库发生改变，改名为 logv2.db 版本升级运行将自动把 Default 目录下的 logv2.db 复制到当前项目中，确保 logv2.db 存在于默认项目目录中
2	7.2	报警历史查询界面增加变量分组查询功能 可以通过 url 增加参数方式初始界面
3		Rest 接口增加历史数据查询接口 历史报警分组查询接口
4		OPCUA 服务器增加会话数量日志输出，会话发生变化时输出到系统日志
5		组态环境更新界面，图形对话框增加排序和筛选功能
6	附件 2	增加内置变量写入时执行用户验证功能（电子签名） 增加 js 函数
7		OPCUA 服务器经过 5 万点测试，CPU 使用率提高 5%-10%

V2.6.13 版本更新（2025.11）

序号	章节	说明
1	3.7	实时报警界面 url 支持增加 taggroup 参数 例如：alarmview.html?taggroup=1 alarmview.html?taggroup=1,3,2
2	8.26	SLC500 驱动增加 N7:1/0 读写支持 L9:1/0 读写支持
3	8.2，8.25，8.16	Modbus 以太网和串口驱动、DTU 驱动、利时增加 BigUnicode 字符编码
4	8.4	S7 驱动增加字符编码设置
5	附件 3	表达式增加取位函数 bit 和 bit64 bit([second],1) 16 和 32 位数据取位，返回 true 或者 false

		bit64([second],1) 64 位数据取位，返回 true 或者 false

V2.6.15 版本更新（2025.12）

序号	章节	说明
1	1.4	项目导出压缩包增加 images 目录下的背景图片、用户控件、用户图片目录下的文件，导入时覆盖操作
2	8.6	更新 MQTT 驱动配置文档，增加 MQTT 驱动配置地址信息说明
3		更新 ABLink 驱动断线重连异常问题

V2.6.16 版本更新（2026.2）

序号	章节	说明
1		修复运行时登录失效跳转到英文登录界面的问题
2		修复定时调度和 JS 事件扩展导入后不执行的问题
3		修复画面参数文件导入问题
4	2.11	增加配方功能 从老版本项目升级需在系统配置页面点击一次**“创建组态表”**按钮
5	2.10	MQTT 转发增加 JS 代码发送支持
6	附件 1 附件 2	增加配方函数 app.DownloadRecipe(string recipeName) downloadRecipe，recipeView
7		修复 username 变量登录后不更新的问题 修复操作系统设置 UTC 时间后用户配置页面打不开的问题

国产系统支持列表（x86 和 arm 构架）：

1）麒麟操作系统 UOS 操作系统 深度操作系统 欧拉操作系统

2）达梦数据库 openGauss(高斯)数据库

3）唐码实时数据库 智方实时数据库

提醒：版本升级请清理浏览器缓存。

目录

1.软件安装 1
1.1 软件下载和解压 1
1.2 启动运行 2
1.3 软件升级 8
1.4 项目备份和导入 9
2.功能配置 10
2.1 系统配置 10
2.2 组态配置 16
2.3 驱动配置 17
2.4 扩展配置 22
2.4.1 IO 转发 23
2.4.2 报警事件 24
2.4.3 定时调度 24
2.4.4 电子邮件发送 26
2.4.5 微信发送 27
2.4.6 爱普生票据打印机 30
2.4.7 Modubs 从站扩展 34
2.4.8 OPCUA 服务器 35
2.4.9 TCP 服务器扩展 36
2.4.10 MQTT 发送扩展 36
2.4.11 JYC311 短信发送扩展 38
2.5 变量配置 40
2.6 用户配置 42
2.7 画面设置 44
2.8 变量替换 44
2.9 RunTime 函数 45
2.10 语音报警 45
2.11 配方设置 47
3. Web 组态和运行 48
3.1 组态环境 48
3.1.1 HTML 和 Script 控件数据更新事件 49
3.1.2 HTML5 原生控件 49
3.1.3 变量写入控件 50
3.1.4 设备控制和变量写入 50
3.1.5 用户图形的变量绑定 51
3.2 运行环境 51
3.3 实时数据库 54
3.4 系统状态 56
3.5 实时趋势 57
3.6 历史趋势 58
3.7 实时报警 58
3.8 实时日志 58

3.9 SSL 加密通讯 59
3.10 WebSocket 通讯 61
4.辅助配置 62
4.2 画面分组 62
4.3 设备分组 62
4.4 变量分组 63
4.5 JS 脚本 63
4.6 批量配置 66
4.7 趋势配置 66
5.归档配置 66
5.1 历史归档库 66
5.2 关系库同步 69
5.3 列表归档 70
5.4 行表归档 74
5.5 自动删除数据 75
6.归档查询 76
6.1 历史归档查询 76
6.2 列表归档查询 76
6.3 行表归档查询 77
7.日志查询 78
7.1 系统日志查询 78
7.2 报警日志查询 78
7.3 变量日志查询 79
7.4 操作日志查询 79
7.5 访问日志查询 80
7.6 登录日志查询 80
8.驱动配置 82
8.1 模拟驱动 82
8.2 Modbus 以太网驱动 83
8.3 Modbus 串行驱动 86
8.4 S7 驱动 86
8.5 OPCUA 驱动 90
8.6 MQTT 驱动 92
8.7 网络接收驱动 94
8.8 Ping 驱动 95
8.9 IEC104 驱动 96
8.10 关系库驱动 99
8.11 HTTP 接收驱动 102
8.12 BACNet 驱动 105
8.13 智方实时数据库驱动 107
8.14 OPCDA 驱动 108
8.15 DLT645 驱动 110
8.16 DTU 驱动 112
8.17 唐码实时数据库驱动 114

8.18 NETTCP 驱动 115
8.19 三菱 PLC 驱动（A1E） 116
8.19 三菱 PLC 驱动（Qna3E） 118
8.20 欧姆龙 PLC 驱动 120
8.21 TCPClient 驱动 122
8.22 用户驱动 124
8.23 Logix5000 驱动 126
8.24 ConnectedCIP 驱动 127
8.25 和利时 PLC 驱动 128
8.26 AB PLC 驱动 130
9. REST 接口 132
10. 操作系统支持和授权方式 132
附件 1：用于 JS 脚本文件的内置 JavaScript 函数（后端使用） 133
附件 2：用于 HTML5 Web 页面的内置 JavaScript 函数（前端使用） 141
附件 3：表达式计算 147
附件 4：数据库连接字符串格式 150
附件 5：Docker 使用说明 151
附件 7：InfluxDB 安装 152
附件 8：华为 openGauss 数据库 156
附件 9：UDPService 服务 159
附件 10：欧拉系统 MySQL 离线安装 160
附件 11：扩展 C# DLL 164
附件 12：用于 JS 的网络通讯类 166
附件 13：WebAPI 扩展 170
附件 14：二维码 QRCode 动态生成 171

1.软件安装

1.1 软件下载和解压

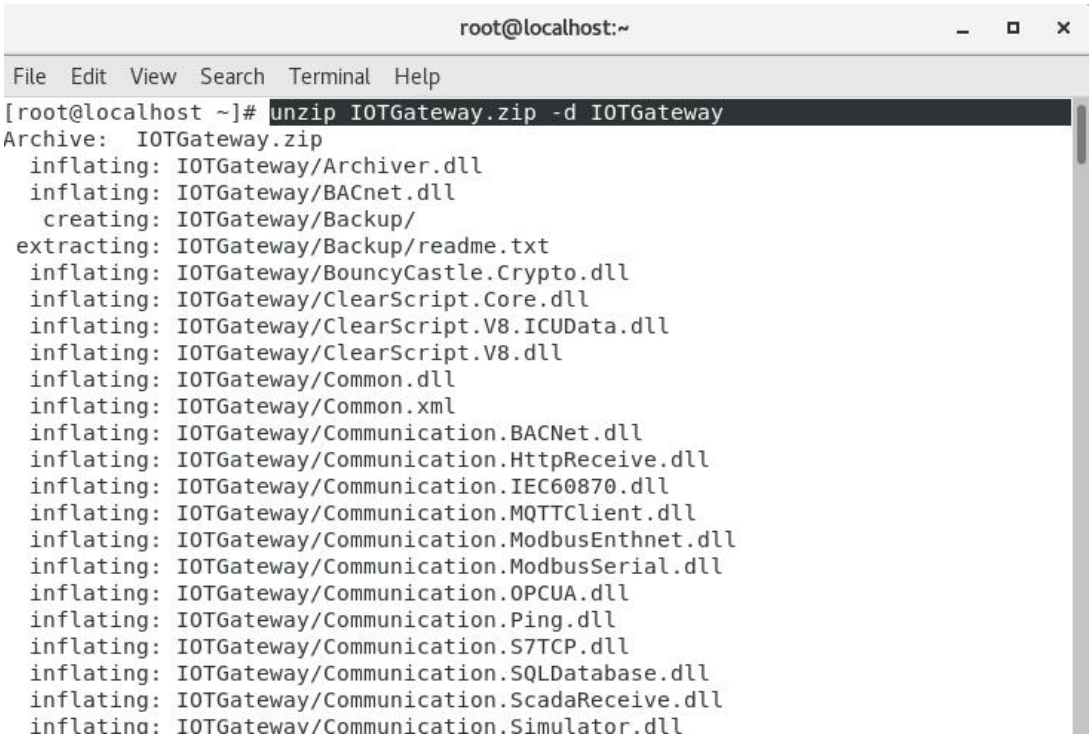
1）拷贝软件压缩包到 64 位 Linux 系统，解压到某个目录下。

可以在 Windows 下通过 PowerShell 上载软件到 Linux 系统，使用 scp 命令：

scp d:\IOTGateway.zip root@192.168.10.13:/home/Downloads

或者安装 MobaXterm 等支持 SSH 上传的工具软件，直接上传

解压缩命令：unzip IOTGateway.zip -d IOTGateway



2）进入解压后的目录下对 IOTGateway 文件赋予可执行权限，执行下列命令。

chmod +x IOTGateway

3）修改系统时区为北京时间

sudo timedatectl set-timezone UTC

sudo timedatectl set-timezone Asia/Shanghai

检查是否成功

timedatectl

1.2 启动运行

1）在终端中输入./IOTGateway 启动程序

从上图的显示可以看到 Web 服务运行在 8080 端口，按 Ctrl+C 停止运行，WebAPI 路径：swagger/index.html（一般建议关闭）

如果启动错误信息为 Could not find a valid ICU package installed on the System to run with no globalization support.

表示系统缺少多语言支持库，需要手动安装 libicu 库

对于 CentOS 执行 sudo yum install libicu

2）修改 Web 服务端口的方法

打开 appsettings.json 文件

urls 是 Web 服务绑定端口设置，多个地址使用分号分隔

修改 ProjectReadOnly 为 true，可以禁止项目编辑

修改 Html5ReadOnly 为 true，可以禁止 H5 画面保存

MaxTags,设置非 0 时,授权变量数大于设置时 status.html 显示的授权点数会以无限点版本显示

CorsPolicy 是跨域访问设置，AllowedHosts 是允许访问的地址设置

启用 https 注释掉第一行 urls，取消注释 Kestrel 下面的设置。

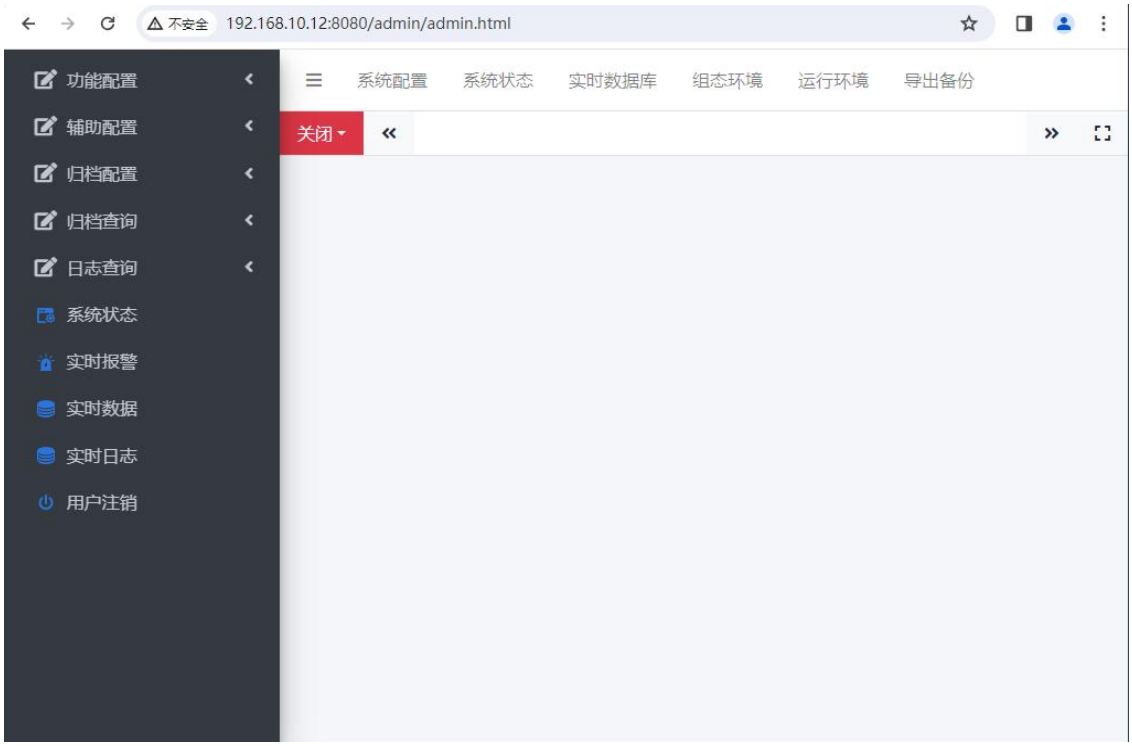
3）进入管理后台系统

http://ip:8080/admin

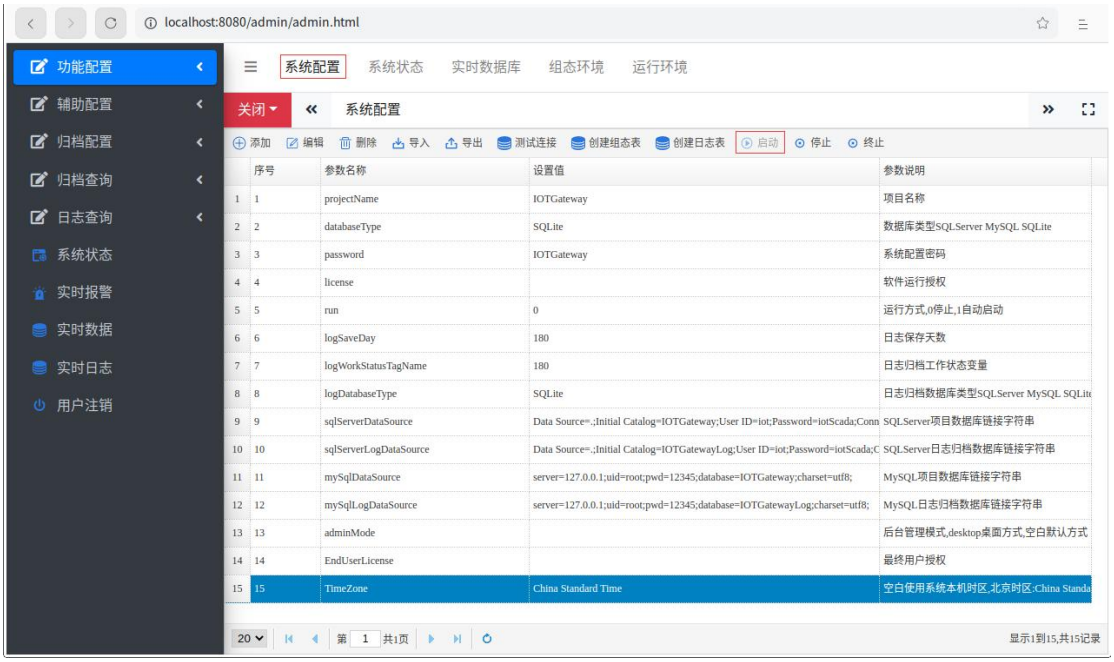
默认管理员：admin admin@admin

用于系统配置的超级管理员：sysadmin IOTGateway

强烈建议发布项目修改此 2 个密码



点击系统配置，进入配置界面



根据需要进行参数设置，点击启动按钮，启动系统运行。

修改 run 为 1 则可以运行软件后自动进入运行，停止状态下修改 run 为 1 也会立即进入运行方式。

默认使用的数据库为内置的SQLite项目数据库,数据库文件存在Data目录下(已经包括默认配置可以直接运行）。

根据需要可以修改为 SQLServer、MySQL、PostgreSQL、达梦数据库（需要先创建空白数据库），在系统配置中修改数据类型和数据库连接字符串后，先点测试连接，测试成功后点击创建组态表和创建日志表按钮，必须保证 2 个数据库（项目数据库和日志数据库）是空白的（无任何表，否则无法创建成功）。如果要使用 1 个数据库，可以手动合并表。

各种数据库的连接字符模版如下：

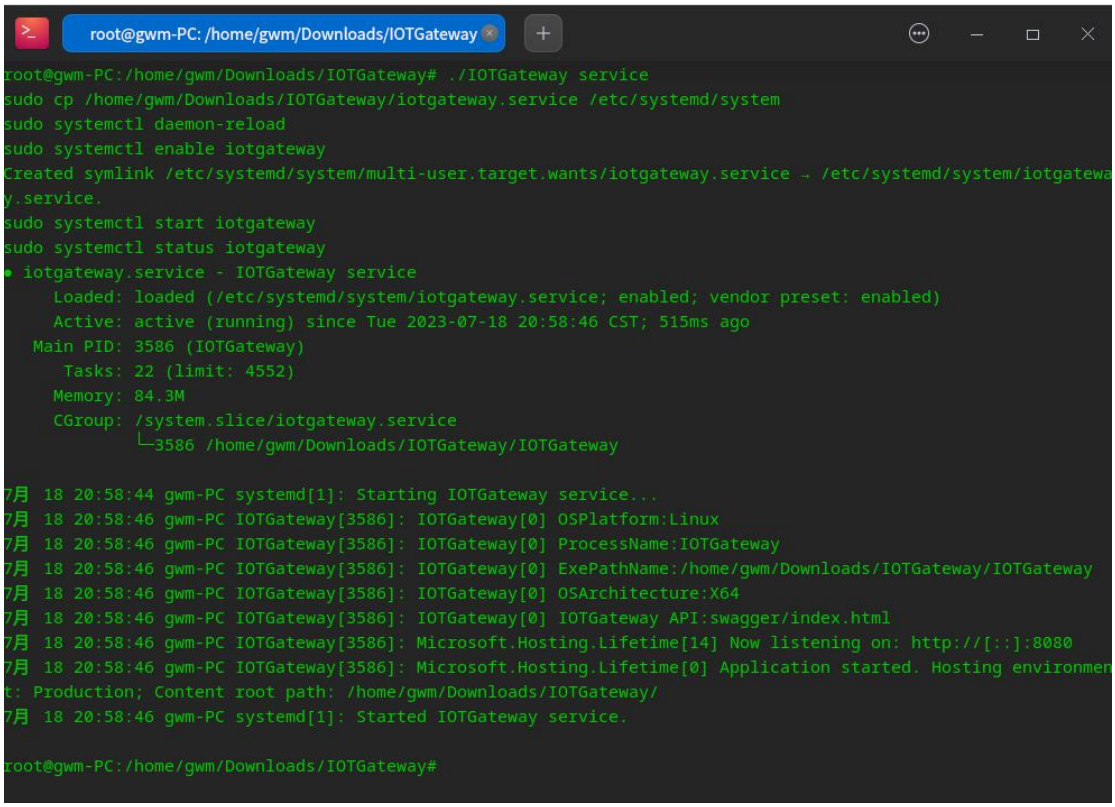
序号	数据库类型	连接字符串格式
1	SQLServer	Data Source=.;Initial Catalog=iotLog;User ID=iot;Password=iot;Connect Timeout=30;
2	MySQL	server=192.168.10.33;uid=root;pwd=123456;database=iotLog;Connect Timeout=30;charset=utf8;
3	PostgreSQL	Host=192.168.10.33;Port=5432;Username=postgres;Password=postgres;Database=iotLog;encoding=UTF8;
4	达梦 8	Server=192.168.10.33;UserId=SYSDBA;PWD=SYSDBA;DATABASE=IOTM

		LOG;

4）安装为系统服务

对于主流发行版的 Linux，执行 systemd 的系统可以安装为系统服务，使用管理员权限执行

sudo ./IOTGateway service



相关服务操作命令如下：

systemctl status iotgateway 查看服务状态

systemctl stop iotgateway 停止运行

systemctl start iotgateway 启动服务

systemctl disable iotgateway 删除服务

服务模式必须设置 run 为 1，否则虽然系统能自动启动但不会进入运行状态，通常配置调试完成安装为服务。安装成功之后系统就会自动运行，不再需要通过命令行启动。

journalctl -u iotgateway 查看服务完整日志，用于异常分析

5）欧拉系统

（1）服务安装后无法启动 203/EXEC 错误

getenforce　　命令查看 SELinux 显示 Enforcing 则需要修改

setenforce 0　临时关闭 SELinux

restorecon -Rv　恢复安全上下文

再次安装服务，如果成功了，就按下面的设置进行操作

修改 /etc/selinux 目录下 config 文件

enforcing - SELinux security policy is enforced.

permissive - SELinux prints warnings instead of enforcing.

disabled - No SELinux policy is loaded.

SELINUX=enforcing => SELINUX=disabled

修改为 **permissive** 或者 **disabled** ，重启系统

（2）软件启动后 Web 无法访问

检查防火墙是否运行 systemctl status firewalld

关闭防火墙命令 systemctl stop firewalld

启动防火墙命令 systemctl start firewalld

禁用防火墙服务命令 systemctl disable firewalld

下列命令添加 1 个 8080 端口到防火墙

firewall-cmd --zone=public --add-port=8080/tcp --permanent

firewall-cmd --reload

查看已经打开的防火墙端口：

firewall-cmd --zone=public --list-ports

（3）安装 influxdb 1x 版本软件

拷贝 rpm 安装文件到系统

执行 rpm -ivh influxdb.rpm

cp /usr/lib/influxdb/scripts/influxdb.service /usr/lib/systemd/system/

systemctl enable influxdb

systemctl start influxdb

测试 8086 端口是否访问正常，后续服务自动运行

系统配置文件路径：/etc/influxdb/influxdb.conf

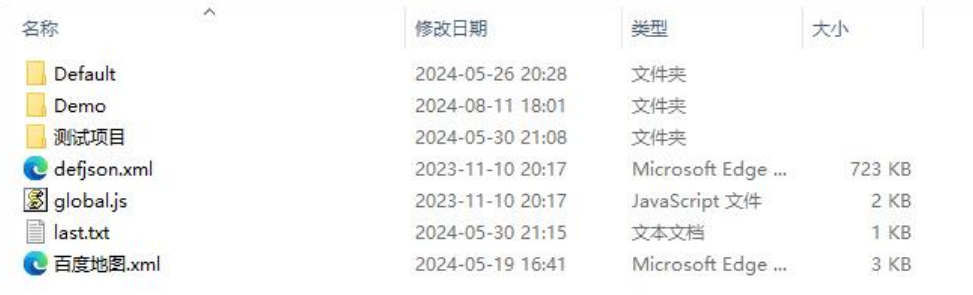
默认存储文件路径：/var/lib/influxdb/

1.3 软件升级

名称	修改日期	类型	大小
defjson.xml	2023-06-13 21:48	XML 文档	723 KB
log.db	2023-07-30 19:18	Data Base File	80 KB
project.db	2023-07-30 19:18	Data Base File	24 KB
scada.db	2023-07-30 19:15	Data Base File	1,176 KB

Data 目录下的文件是项目配置，project.db 是项目参数设置数据库，scada.db 是内置 SQLite 组态数据库，软件版本更新时需要进行备份。

从 2.3.5 版本开始项目采用独立目录存储，Data 目录结构发生变化



名称	修改日期	类型	大小
Default	2024-05-26 20:28	文件夹	
Demo	2024-08-11 18:01	文件夹	
测试项目	2024-05-30 21:08	文件夹	
defjson.xml	2023-11-10 20:17	Microsoft Edge ...	723 KB
global.js	2023-11-10 20:17	JavaScript 文件	2 KB
last.txt	2024-05-30 21:15	文本文档	1 KB
百度地图.xml	2024-05-19 16:41	Microsoft Edge ...	3 KB

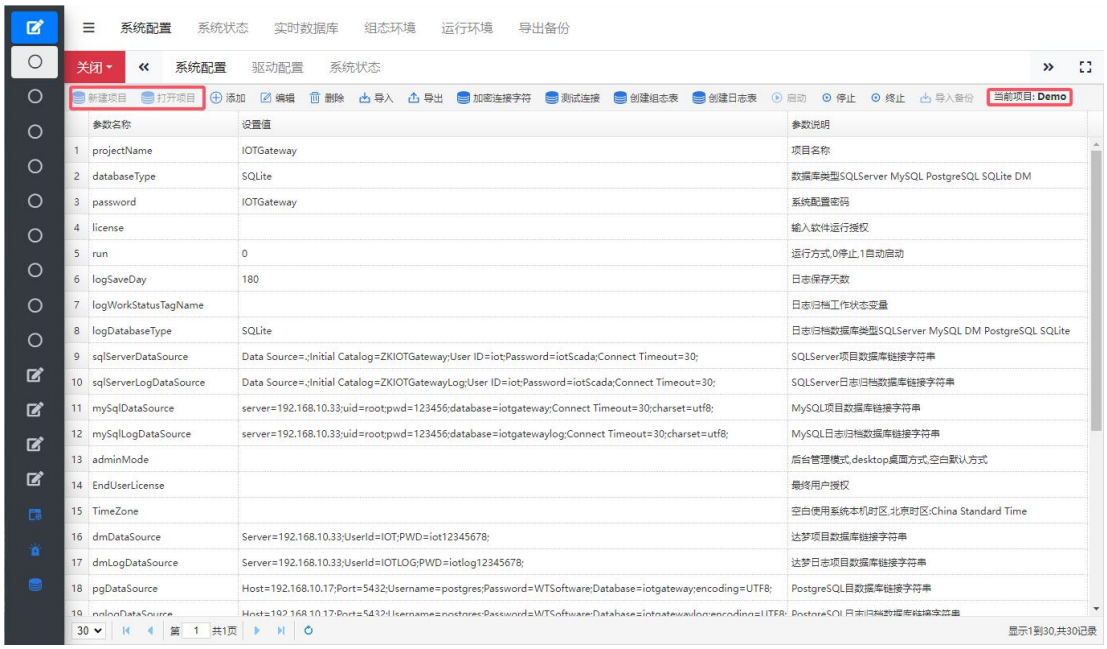
Last.txt 文件存储当前项目名称

Default 目录是模板项目文件，新建项目时从该目录进行项目复制。

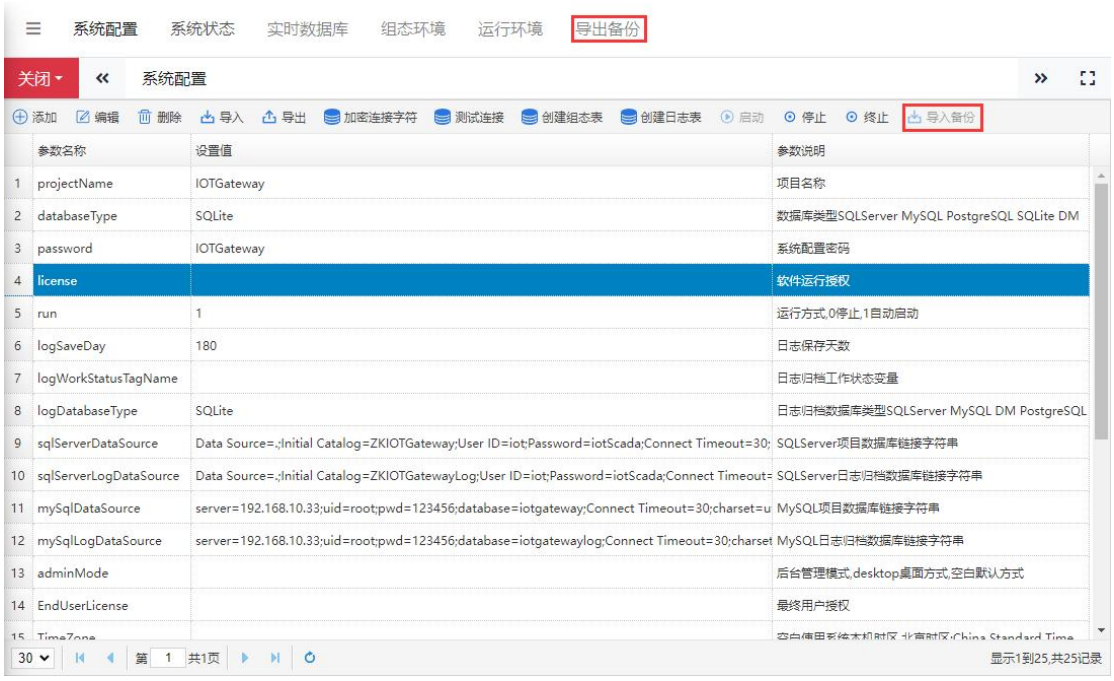
打开项目文件时该文件自动更新。

旧版本升级到 2.3.5 和后续版本时把 3 个 db 文件拷贝到 Default 目录或者新建 1 个目录拷贝进去，启动软件后进入系统设置打开新建的项目，运行即可。

项目迁移只需要拷贝对应的项目目录。



1.4 项目备份和导入



导出备份为 1 个 zip 压缩包，需要在停止状态导入备份，zip 压缩包内有 3 个文件和 1 个 imags 目录，可以根据需要删除某文件，系统配置在导出文件中，但不会导入。

压缩包中的 excel 文件是各种配置，xml 是画面文件和用户控件。

images 目录下包括 background，user，control/upload 目录，存储用户上载的图片控件。

2.功能配置

本章节仅在管理员权限可以进入。

2.1 系统配置

序号	参数名称	设置值	参数说明
1	projectName	IOTGateway	项目名称
2	databaseType	SQLite	数据库类型 SQLServer MySQL SQLite
3	password	IOTGateway	系统配置密码
4	license		软件运行授权
5	run	0	运行方式,0 停止,1 自动启动 配置为系统服务后必须设置为 1
6	logSaveDay	180	日志保存天数
7	logWorkStatusTagName	180	日志归档工作状态变量
8	logDatabaseType	SQLite	日志归档数据库类型 SQLServer MySQL SQLite
9	sqlServerDataSource	Data Source=.;Initial Catalog=IOTGateway;User ID=iot;Password=iotScada;Connect Timeout=30;	SQLServer 项目数据库链接字符串
10	sqlServerLogDataSource	Data Source=.;Initial Catalog=IOTGatewayLog;User ID=iot;Password=iotScada;Connect Timeout=30;	SQLServer 日志归档数据库链接字符串
11	mySqlDataSource	server=127.0.0.1;uid=root;pwd=12345;database=IOTGateway;charset=utf8;	MySQL 项目数据库链接字符串
12	mySqlLogDataSource	server=127.0.0.1;uid=root;pwd=12345;database=IOTGatewayLog;charset=utf8;	MySQL 日志归档数据库链接字符串
13	dmDataSource	Server=192.168.10.33;UserId=SYSDBA;PWD=SYSDBA;DATABASE=SCADA;	达梦数据库项目数据库链接字符串
14	dmLogDataSource	Server=192.168.10.33;UserId=SYSDBA;PWD=SYSDBA;DATABASE=SCADA;	达梦数据库日志归档数据库链接字符串
15	pgDataSource	Host=192.168.10.17;Port=5432;Username=postgres;Password=postgres;Database=IOTGateway;encoding=UTF8;	PostgreSQL 项目数据库链接字符串

16	pglogDataSource	Host=192.168.10.17;Port=5432;Username=postgres;Password=postgres;Database=IOTGatewayLog;encoding=UTF8;	PostgreSQL 日志归档数据库链接字符串
17	adminMode		后台管理模式,desktop 桌面方式,空白默认方式
18	EndUserLicense		最终用户授权
19	TimeZone		空白使用系统本机时区,北京时区:China Standard Time,非空白系统采用 UTC 时间
20	Language		zh-CN 或者 en-US 空白使用默认值
21	NtpServer	fasle	是否启用 NTP 时钟服务器 true false
22	UDPService	0	UDP 数据服务端口,非 0 启用
23	MaxHistCacheCount	3600	历史归档缓存区数量，最小 60 当历史归档无法写入完成时会在内存堆积，如果内存不够则可能引起应用程序因内存不够崩溃
24	MaxReportCacheCount	2000	报表归档缓存区数量 当归档无法写入完成时会在内存堆积，如果内存不够则可能引起应用程序因内存不够崩溃，最小 60
25	GlobalScript		全局 js 脚本文件名称
26	StartFunction	start()	启动时执行全局 js 函数
27	StopFunction	stop()	停止时执行的全局 js 函数
28	ComplexPassword	1	复杂密码设置(长度大于 8，包括大小写数字和特殊字符) 设置 0 关闭
29	PasswordValidityPeriod	90	密码有效期(天) 设置 0 关闭
30	UpdateUrl	http://localhost:8080	远程更新路径
31	UpdateToken	X1234567890X	远程更新 Token

注:1)通常应该修改项目名称和默认系统配置密码

2)考虑项目移值方便建议采用 SQLite 数据库类型,特别大的系统使用 SQLServer、MySQL、PostgreSQL 数据库。

3)使用软件授权方式时在 license 值输入授权信息。

4)TimeZone 一般不建议修改，非空白系统采用 UTC 时间，归档中的时间都会使用 UTC 时间。

5）UDPService 设置非 0 则启动 UDP 服务。

6）配置为系统服务模式后，run 参数必须设置为 1。

时区设置表如下（设置时区 ID 到 TimeZone）

时区 ID	时区名称	时间差
Dateline Standard Time	国际日期变更线标准时间	-12:00:00
UTC-11	UTC-11	-11:00:00
Hawaiian Standard Time	夏威夷标准时间	-10:00:00
Aleutian Standard Time	阿留申群岛标准时间	-10:00:00
Marquesas Standard Time	马克萨斯群岛标准时间	-09:30:00
UTC-09	UTC-09	-09:00:00
Alaskan Standard Time	阿拉斯加标准时间	-09:00:00
Pacific Standard Time (Mexico)	太平洋标准时间(墨西哥)	-08:00:00
UTC-08	UTC-08	-08:00:00
Pacific Standard Time	太平洋标准时间	-08:00:00
US Mountain Standard Time	美国山地标准时间	-07:00:00
Mountain Standard Time	山地标准时间	-07:00:00
Mountain Standard Time (Mexico)	山地标准时间(墨西哥)	-07:00:00
Yukon Standard Time	育空标准时间	-07:00:00
Central America Standard Time	中美洲标准时间	-06:00:00
Central Standard Time	中部标准时间	-06:00:00
Easter Island Standard Time	复活节岛标准时间	-06:00:00
Central Standard Time (Mexico)	中部标准时间(墨西哥)	-06:00:00
Canada Central Standard Time	加拿大中部标准时间	-06:00:00
Eastern Standard Time	东部标准时间	-05:00:00
Eastern Standard Time (Mexico)	东部标准时间(墨西哥)	-05:00:00
US Eastern Standard Time	美国东部标准时间	-05:00:00
Cuba Standard Time	古巴标准时间	-05:00:00
SA Pacific Standard Time	南美洲太平洋标准时间	-05:00:00
Haiti Standard Time	海地标准时间	-05:00:00
Turks And Caicos Standard Time	特克斯和凯科斯群岛标准时间	-05:00:00
SA Western Standard Time	南美洲西部标准时间	-04:00:00
Paraguay Standard Time	巴拉圭标准时间	-04:00:00
Venezuela Standard Time	委内瑞拉标准时间	-04:00:00
Pacific SA Standard Time	太平洋南美洲标准时间	-04:00:00
Atlantic Standard Time	大西洋标准时间	-04:00:00
Central Brazilian Standard Time	巴西中部标准时间	-04:00:00
Newfoundland Standard Time	纽芬兰标准时间	-03:30:00
SA Eastern Standard Time	南美洲东部标准时间	-03:00:00
Saint Pierre Standard Time	圣皮埃尔标准时间	-03:00:00
E. South America Standard Time	东部南美洲标准时间	-03:00:00
Argentina Standard Time	阿根廷标准时间	-03:00:00

Bahia Standard Time	巴伊亚标准时间	-03:00:00
Montevideo Standard Time	蒙得维的亚标准时间	-03:00:00
Magallanes Standard Time	麦哲伦标准时间	-03:00:00
Tocantins Standard Time	托坎廷斯标准时间	-03:00:00
Mid-Atlantic Standard Time	中大西洋标准时间	-02:00:00
UTC-02	UTC-02	-02:00:00
Greenland Standard Time	格陵兰标准时间	-02:00:00
Azores Standard Time	亚速尔群岛标准时间	-01:00:00
Cape Verde Standard Time	佛得角标准时间	-01:00:00
UTC	协调世界时	0:00:00
Sao Tome Standard Time	圣多美标准时	0:00:00
Greenwich Standard Time	格林威治标准时间	0:00:00
GMT Standard Time	GMT 标准时间	0:00:00
Morocco Standard Time	摩洛哥标准时间	0:00:00
W. Central Africa Standard Time	中非西部标准时间	1:00:00
Romance Standard Time	罗马标准时间	1:00:00
Central European Standard Time	中欧的标准时间	1:00:00
Central Europe Standard Time	中欧标准时间	1:00:00
W. Europe Standard Time	西欧标准时间	1:00:00
West Bank Standard Time	西岸加沙标准时间	2:00:00
Kaliningrad Standard Time	俄罗斯 TZ 1 标准时间	2:00:00
South Africa Standard Time	南非标准时间	2:00:00
Sudan Standard Time	苏丹标准时	2:00:00
E. Europe Standard Time	东欧标准时间	2:00:00
Egypt Standard Time	埃及标准时间	2:00:00
South Sudan Standard Time	南苏丹标准时间	2:00:00
Namibia Standard Time	纳米比亚标准时间	2:00:00
Libya Standard Time	利比亚标准时间	2:00:00
Israel Standard Time	耶路撒冷标准时间	2:00:00
Middle East Standard Time	中东标准时间	2:00:00
FLE Standard Time	FLE 标准时间	2:00:00
GTB Standard Time	GTB 标准时间	2:00:00
Turkey Standard Time	土耳其标准时间	3:00:00
Volgograd Standard Time	伏尔加格勒标准时间	3:00:00
E. Africa Standard Time	东非标准时间	3:00:00
Syria Standard Time	叙利亚标准时间	3:00:00
Jordan Standard Time	约旦标准时间	3:00:00
Arabic Standard Time	阿拉伯 (Arabic) 标准时间	3:00:00
Belarus Standard Time	白俄罗斯标准时间	3:00:00
Arab Standard Time	阿拉伯 (Arab) 标准时间	3:00:00
Russian Standard Time	俄罗斯 TZ 2 标准时间	3:00:00
Iran Standard Time	伊朗标准时间	3:30:00
Russia Time Zone 3	俄罗斯 TZ 3 标准时间	4:00:00

Caucasus Standard Time	高加索标准时间	4:00:00
Azerbaijan Standard Time	阿塞拜疆标准时间	4:00:00
Georgian Standard Time	格鲁吉亚标准时间	4:00:00
Saratov Standard Time	萨拉托夫标准时间	4:00:00
Mauritius Standard Time	毛里求斯标准时间	4:00:00
Arabian Standard Time	阿拉伯半岛标准时间	4:00:00
Astrakhan Standard Time	阿斯特拉罕标准时间	4:00:00
Afghanistan Standard Time	阿富汗标准时间	4:30:00
Pakistan Standard Time	巴基斯坦标准时间	5:00:00
Qyzylorda Standard Time	克孜洛尔达标准时间	5:00:00
Ekaterinburg Standard Time	俄罗斯 TZ 4 标准时间	5:00:00
West Asia Standard Time	西亚标准时间	5:00:00
Sri Lanka Standard Time	斯里兰卡标准时间	5:30:00
India Standard Time	印度标准时间	5:30:00
Nepal Standard Time	尼泊尔标准时间	5:45:00
Bangladesh Standard Time	孟加拉国标准时间	6:00:00
Omsk Standard Time	鄂木斯克标准时间	6:00:00
Central Asia Standard Time	中亚标准时间	6:00:00
Myanmar Standard Time	缅甸标准时间	6:30:00
North Asia Standard Time	俄罗斯 TZ 6 标准时间	7:00:00
Altai Standard Time	阿尔泰标准时间	7:00:00
Tomsk Standard Time	托木斯克标准时间	7:00:00
N. Central Asia Standard Time	新西伯利亚标准时间	7:00:00
SE Asia Standard Time	东南亚标准时间	7:00:00
W. Mongolia Standard Time	西蒙古标准时间	7:00:00
Ulaanbaatar Standard Time	乌兰巴托标准时间	8:00:00
North Asia East Standard Time	俄罗斯 TZ 7 标准时间	8:00:00
China Standard Time	中国标准时间	8:00:00
Taipei Standard Time	台北标准时间	8:00:00
Singapore Standard Time	马来西亚半岛标准时间	8:00:00
W. Australia Standard Time	澳大利亚西部标准时间	8:00:00
Aus Central W. Standard Time	澳大利亚中西部标准时间	8:45:00
Tokyo Standard Time	东京标准时间	9:00:00
North Korea Standard Time	朝鲜标准时间	9:00:00
Transbaikal Standard Time	外贝加尔标准时间	9:00:00
Yakutsk Standard Time	俄罗斯 TZ 8 标准时间	9:00:00
Korea Standard Time	韩国标准时间	9:00:00
AUS Central Standard Time	澳大利亚中部标准时间	9:30:00
Cen. Australia Standard Time	中部澳大利亚标准时间	9:30:00
West Pacific Standard Time	太平洋西部标准时间	10:00:00
AUS Eastern Standard Time	澳大利亚东部标准时间	10:00:00
E. Australia Standard Time	东部澳大利亚标准时间	10:00:00
Vladivostok Standard Time	俄罗斯 TZ 9 标准时间	10:00:00

Tasmania Standard Time	塔斯马尼亚岛标准时间	10:00:00
Lord Howe Standard Time	豪勋爵岛标准时间	10:30:00
Russia Time Zone 10	俄罗斯 TZ 10 标准时间	11:00:00
Bougainville Standard Time	布干维尔岛标准时间	11:00:00
Central Pacific Standard Time	太平洋中部标准时间	11:00:00
Sakhalin Standard Time	萨哈林标准时间	11:00:00
Norfolk Standard Time	诺福克岛标准时间	11:00:00
Magadan Standard Time	马加丹标准时间	11:00:00
UTC+12	UTC+12	12:00:00
New Zealand Standard Time	新西兰标准时间	12:00:00
Kamchatka Standard Time	堪察加标准时间	12:00:00
Fiji Standard Time	斐济标准时间	12:00:00
Russia Time Zone 11	俄罗斯 TZ 11 标准时间	12:00:00
Chatham Islands Standard Time	查塔姆群岛标准时间	12:45:00
Tonga Standard Time	汤加标准时间	13:00:00
UTC+13	UTC+13	13:00:00
Samoa Standard Time	萨摩亚群岛标准时间	13:00:00
Line Islands Standard Time	莱恩群岛标准时间	14:00:00

6)系统配置参数的名称不可修改，也不能删除。

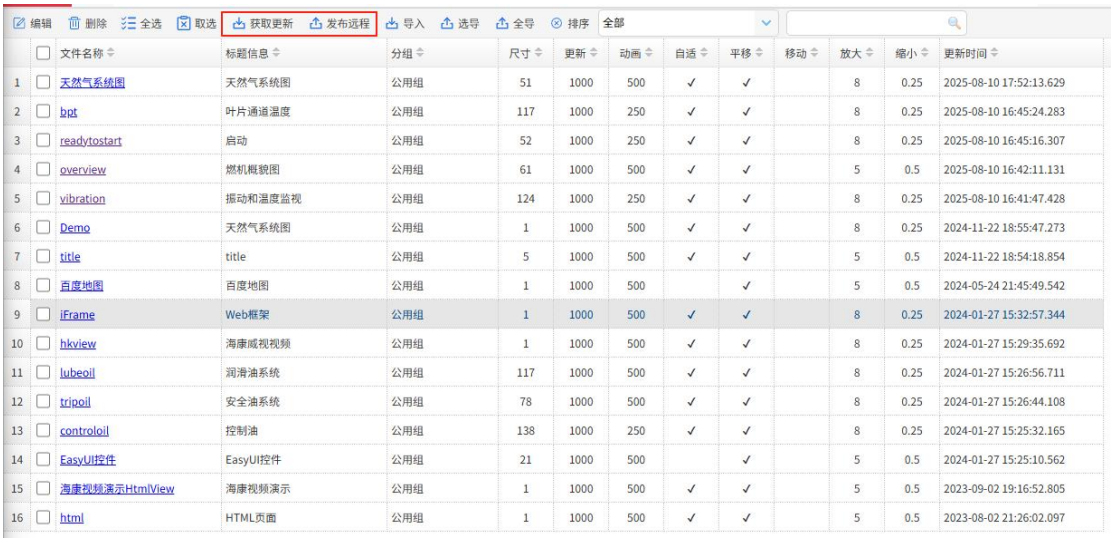
7）系统配置中的数据库连接字符串可以使用工具栏的加密按钮进行加密，加密后连接字符串可以改为密文显示（系统不提供解密显示手段）。

8）下列设置是默认值，根据需要添加以改变设置

序号	参数名称	设置值	参数说明
1	urls	http://*:8080	Web 服务端口，*代表本机全部 IP 地址，可以修改为 http://192.168.10.12:8080 指定 IP 地址（IP 地址必须存在）多个绑定地址使用分号分隔
2	Swagger	true	启动或者关闭 WebAPI 文档接口
3	RecRequestSaveDay	3	访问记录保存天数
4	RecRequest	false	开启或者关闭访问记录，记录耗时大于 10ms 的访问
5	https	false	http 自动切换到 https 当 http 和 https 均启用后有效
6	Middleware	IOTGateway.Middleware.dll,IOTGateway.Middleware.Middleware	中间件（DLL,类名）多个中间间使用分号分隔

9）openGauss 数据库使用 PostgreSQL 设置，数据库连接字符串后增加 No Reset On Close=true;

10）远程更新功能用于多套系统之家的画面同步，UpdateToken 是服务器端的同步令牌设置，UpdateUrl 是服务器端的访问地址。后台画面设置界面上有从远程获取更新画面和发布画面到远程按钮可以同步画面。



	文件名称	标题信息	分组	尺寸	更新	动画	自适	平移	移动	放大	缩小	更新时间
1	天然气系统图	天然气系统图	公用组	51	1000	500	✓	✓		8	0.25	2025-08-10 17:52:13.629
2	bpt	叶片通道温度	公用组	117	1000	250	✓	✓		8	0.25	2025-08-10 16:45:24.283
3	readytostart	启动	公用组	52	1000	250	✓	✓		8	0.25	2025-08-10 16:45:16.307
4	overview	燃机概貌图	公用组	61	1000	500	✓	✓		5	0.5	2025-08-10 16:42:11.131
5	vibration	振动和温度监视	公用组	124	1000	250	✓	✓		8	0.25	2025-08-10 16:41:47.428
6	Demo	天然气系统图	公用组	1	1000	500	✓	✓		8	0.25	2024-11-22 18:55:47.273
7	title	title	公用组	5	1000	500	✓	✓		5	0.5	2024-11-22 18:54:18.854
8	百度地图	百度地图	公用组	1	1000	500		✓		5	0.5	2024-05-24 21:45:49.542
9	iFrame	Web框架	公用组	1	1000	500	✓	✓		8	0.25	2024-01-27 15:32:57.344
10	hkview	海康威视视频	公用组	1	1000	500	✓	✓		8	0.25	2024-01-27 15:29:35.692
11	lubeoil	润滑油系统	公用组	117	1000	500	✓	✓		8	0.25	2024-01-27 15:26:56.711
12	tripoil	安全油系统	公用组	78	1000	500	✓	✓		8	0.25	2024-01-27 15:26:44.108
13	controloil	控制油	公用组	138	1000	250	✓	✓		8	0.25	2024-01-27 15:25:32.165
14	EasyUI控件	EasyUI控件	公用组	21	1000	500		✓		5	0.5	2024-01-27 15:25:10.562
15	海康视频演示HtmlView	海康视频演示	公用组	1	1000	500	✓	✓		5	0.5	2023-09-02 19:16:52.805
16	html	HTML页面	公用组	1	1000	500	✓	✓		5	0.5	2023-08-02 21:26:02.097

2.2 组态配置

序号	参数名称	设置值	参数说明
1	start	overview	默认启动画面
2	startmode	editor	可设置三种状态 run editor status
3	editor_level	0	画面编辑权限设置
4	pagesize_tag	50000	标签浏览分页尺寸
5	defbackground	white	默认图形文件背景颜色
6	viewpiclib	false	是否在组态界面实现图片控件(默认显示用户图片)
7	viewjsoncontrol	false	是否在组态界面显示 JSON 控件 true false
8	save_level	0	保存权限
9	whitelist		白名单，使用逗号分隔多个 IP4 地址，支持*通配符，如 192.168.10.*
10	autologinuser	user	白名单登录的用户名
11	login		设置 0 或者 false,启用匿名进入运行环境 true false，权限限制为公用权限组

注:1)start 启动画面指的是进入运行模式不带画面名称时显示的画面，根据实际进行修改。

2)startmode 运行状态设置为 run。

3)whitelist 白名单是 IP 地址列表，使用逗号分隔，白名单的 IP 地址自动登录为 autologinuser 设置的用户。

4)editor_level 和 save_level 用于控制进入组态界面的权限和修改保存组态画面的权限，通常设置大于 0。组态画面仅管理员权限可以正常使用。

5）login 为 false 时，匿名登录的用户名为 autologin，权限限定为公用权限组。

2.3 驱动配置

序号	名称	显示名称	使能	DLL 文件
1	sim	模拟驱动		Communication.Simulator.dll
2	modbusenthnet	Modbus 以太网驱动		Communication.ModbusEnthnet.dll
3	modbusserial	Modbus 串行驱动		Communication.ModbusSerial.dll
4	scadarev	网络接收驱动		Communication.ScadaReceive.dll
5	ping	Ping 驱动		Communication.Ping.dll
6	httprev	Http 接收驱动		Communication.HttpReceive.dll
7	bacnet	BACNet 驱动		Communication.BACNet.dll
8	iec104	IEC104 驱动		Communication.IEC60870.dll
9	s7tcp	S7TCP 驱动		Communication.S7TCP.dll
10	opcua	OPCUA 驱动		Communication.OPCUA.dll
11	sql	关系数据库驱动		Communication.SQLDatabase.dll
12	mqtt	MQTT 驱动		Communication.MQTTClient.dll
13	dtu	ModbusDTU 驱动		Communication.ModbusDTU.dll
14	zfrtdb	智方实时数据库		Communication.ZFRtdb.dll
15	tmrtdb	唐码实时数据库		Communication.TMRtdb.dll
16	opcda	OPCDA 驱动		Communication.OPCDA.dll
17	dlt645	DLT645 驱动		Communication.DLT645.dll
18	melale	三菱 A1E 驱动		Communication.MelsecA1E.dll
19	qna3e	三菱 Qna3E 驱动		Communication.MelsecQna3E.dll
20	finsnet	欧姆龙 Fins 驱动		Communication.OmronFinsNet.dll
21	tcp	TCP 驱动		Communication.TCP.dll
22	user	用户驱动		Communication.User.dll
23	tcpclient	TCP 客户端驱动		Communication.TCPClient.dll
24				
25				

驱动用于数据采集，要使用某个驱动首先需要使能该驱动。在驱动内配置设备，在设备中配置变量，每个驱动是 1 个独立的 dll 文件，驱动可以复用，名称必须唯一。

1）模拟驱动是内部驱动，用于内存变量的创建（模拟驱动通常是最后一个启动的驱动）。

2）Modbus 以太网驱动，支持 Modbus TCP 和 Modbus UDP 以及 TCP 方式的 RTU。

3）Modbus 串行驱动，支持 RTU 和 ASCII 格式。

4）网络接收驱动，用于 scada 之间的数据转发，支持 IOTGateway 之间的数据转发，FScada 和 WTGateway 的数据转发。

5）Ping 驱动，用于以太网设备的网络状态检测（类似 ping 命令）。

6）Http 接收驱动，支持通过 Http Post 更新变量。

7）BACNet 驱动，楼宇通讯协议，支持 BACNet IP 和 MSTP。

8）IEC104 驱动，电力规约，支持 104 TCP 规约。

9）S7TCP 驱动，西门子 S7 PLC 驱动，支持全系列 S7400，S7300，S7200，S71500，S71200，Smart200 以太网通讯（支持 DTU 模式）。

10）OPCUA 驱动，支持 OPCUA opc.tcp 通讯模式，支持用户认证和匿名认证模式。

11）关系数据库驱动，支持 SQLServer、MySQL 关系库数据读取和写入。

12）MQTT 驱动，MQTT 客户端驱动，支持 keyvalue 格式，可以通过 javascript 脚本自解析和自定义发送。

13）ModbusDTU 驱动，支持带自定义注册包，透传 ModbusRTU 或者 ModbusTCP 协议。

14）智方实时数据库（商业实时数据库）。

15）唐码实时数据库（商业实时数据库）。

16）OPCDA 驱动，仅用于 Windows（需要安装 64 位 OPCDA 发行包）。

17）DLT645 电表驱动，支持串口和网络。

18）三菱 FX 网络驱动。

19）三菱 QnA 网络驱动。

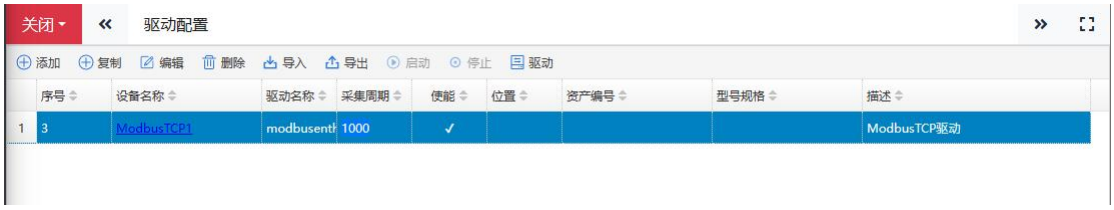
20）欧姆龙 Fins 网络驱动。

21）TCP 驱动用于连接 FScada 和 Gateway，OPCLink 的服务器。

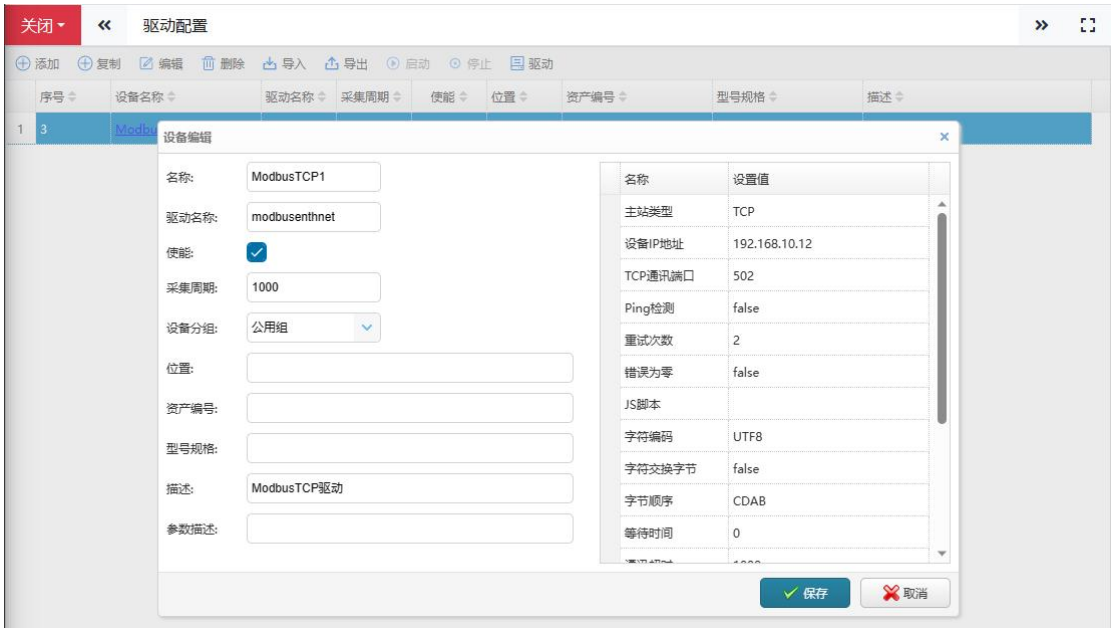
名称是唯一的，不可以重名。

双击或者点击编辑按钮编辑驱动。

点击名称链接进入设备配置界面。

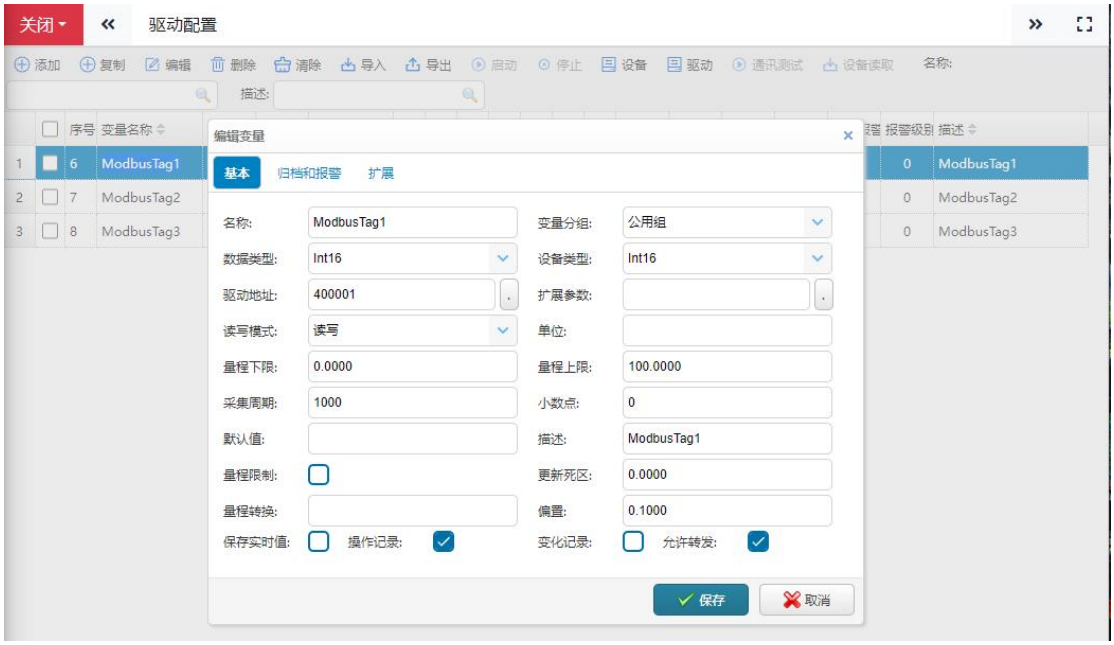


双击或者点击编辑按钮编辑设备。



设备名称是唯一的，不可以重复。

单击设备名称链接进入变量编辑。



变量名称是唯一的，不可以重复。

变量分组：变量分组设置，通过配置分组权限可以控制访问

驱动地址和扩展驱动地址：根据驱动配置（***当驱动地址是 eval 时，扩展驱动地址支持表达式计算***）

模拟驱动时当驱动地址是 js 时，扩展参数后面的按钮可以选择 js 函数名称

采集周期：单位毫秒，默认 1000ms

小数点：用于格式显示

默认值：变量的初始值

更新死区：变量更新事件的死区值设置

量程限制：启用后超过量程范围的值无法被写入

量程转换：表达式支持 + - * /,比如/100

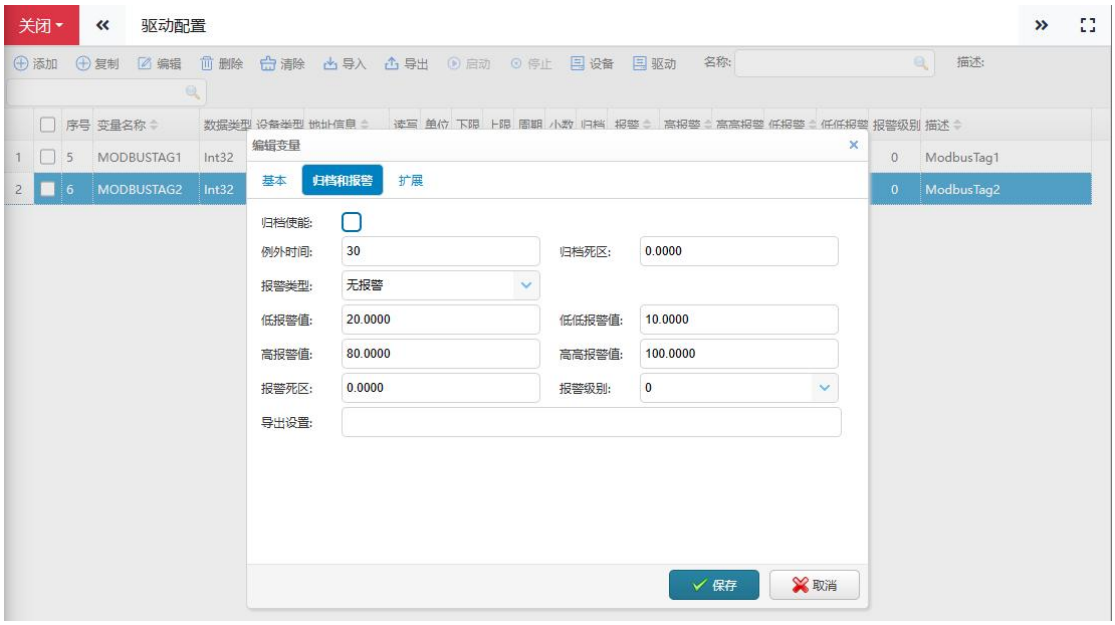
偏置：变量值等于驱动值+偏置

保存实时值：驱动停止时保存当前值

操作记录：记录变量值设置到操作记录中

变化记录：记录变量值变化到变化日志中

允许转发：变量转发扩展使用



归档使能：历史归档使能配置

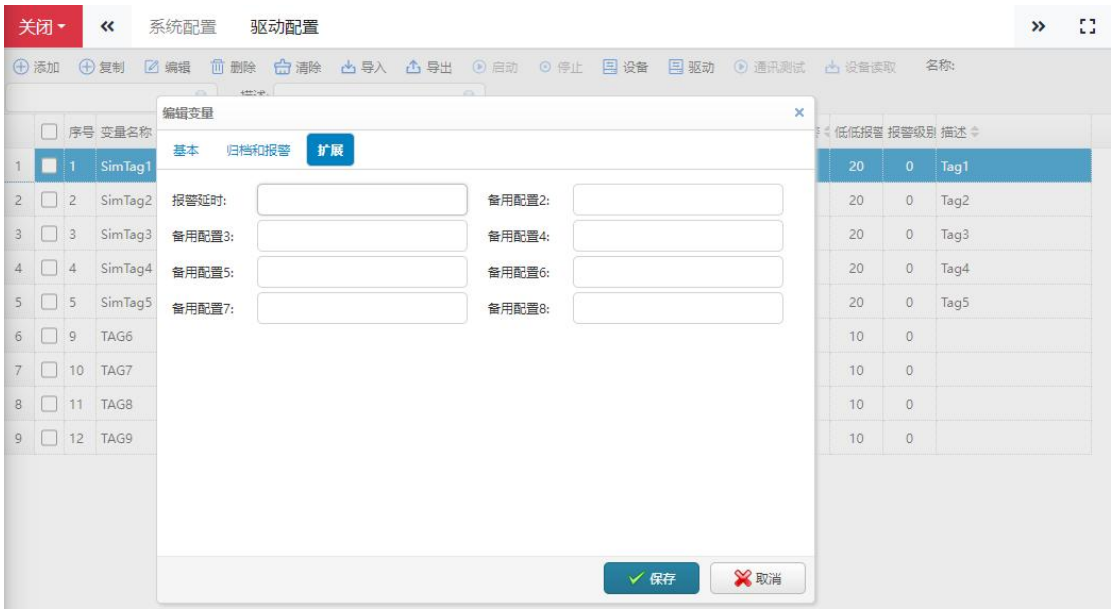
归档死区：当变量值改变超过死区范围时记录历史值

例外时间：如果变量的值不变，达到例外时间记录历史值

报警类型：无报警、模拟量报警、ON 报警、OFF 报警、数字量变化报警、模拟量变化报警

序号	报警类型	变量类型	描述
1	无报警	全部类型	不报警
2	模拟量报警	模拟量类型	变量通过高低报警值定义触发报警
3	ON 报警	数字量类型	变量的值为 True 触发报警
4	OFF 报警	数字量类型	变量的值为 False 触发报警
5	数字量变化报警	数字量类型	变量的值发生改变触发报警
6	模拟量变化报警	模拟量类型	变量的值发生改变触发报警

导出设置：部分扩展使用，根据扩展说明进行配置



报警延时：单位秒，设置非 0 数字用于消除突变报警

备用配置 2：智方实时数据库变量名称

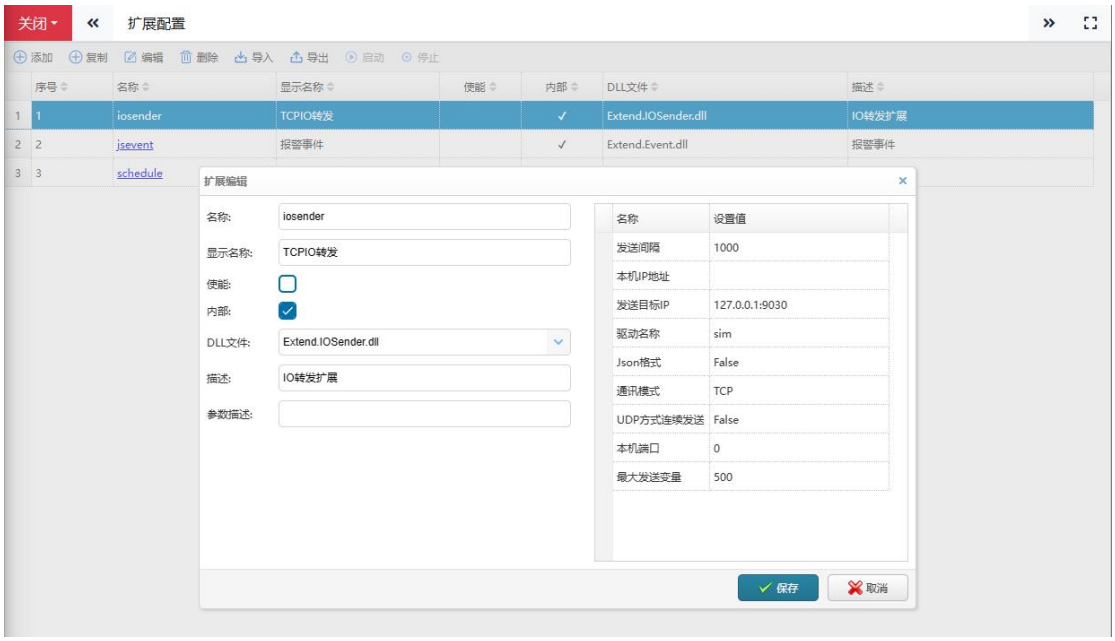
备用配置 3：DTU 驱动的注册包字符串

每个驱动的地址配置说明见第 8 章。

2.4 扩展配置

上图是系统内置的扩展，使能后有效

点击编辑或者双击配置扩展，点击名称超级链接进入详细配置页面



2.4.1 IO 转发

用于 IOTGateway 之间的数据转发，使用网络接收驱动接收

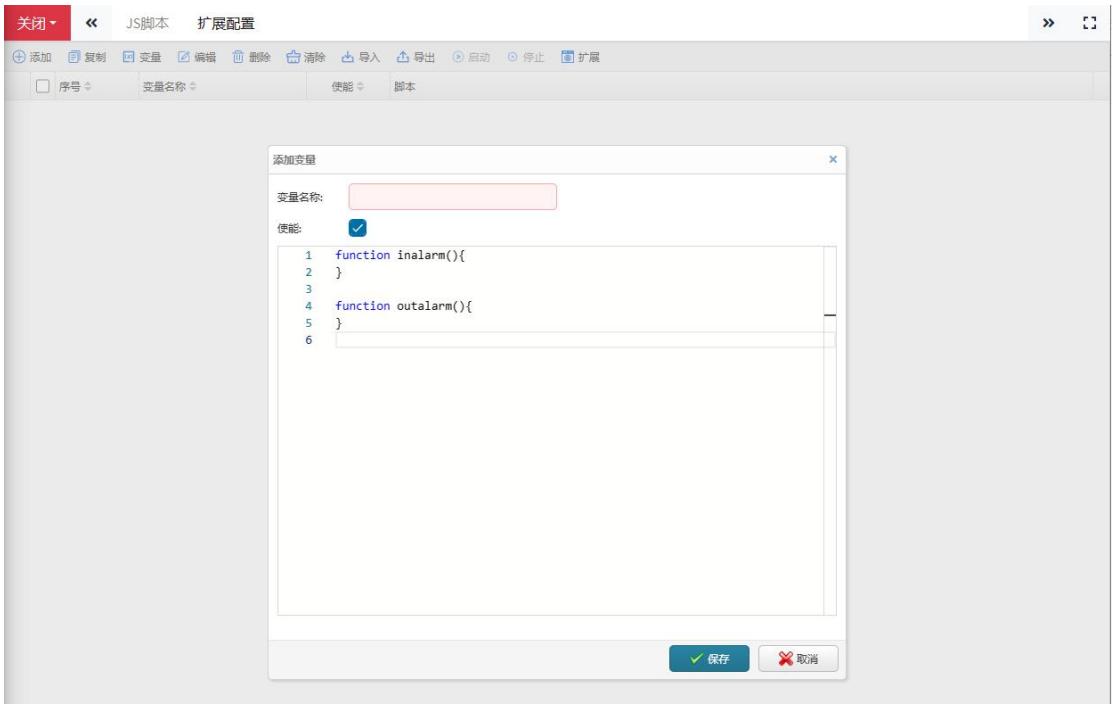
IO 转发配置说明：

名称	设置值	说明
发送间隔	1000	转发频率，单位 ms，默认 1000ms，最小 500ms
本机 IP 地址		绑定本机的 IP 地址，可以空白，指定网卡 IP 地址时使用
发送目标 IP	127.0.0.1:9030	转发目标地址和端口，使用逗号分割可以设置多个转发目标
驱动名称		转发驱动名，空白转发全部驱动，使用逗号分隔多个驱动（本地配置为 False 时有效）
Json 格式	False	以标准 JSON 格式转发，默认 False，使用压缩格式传输，Scada 内部转发必须设置 False，给第三方转发时可以使用 True，以明文方式转发（仅对 UDP 方式有效）
通讯模式	TCP	TCP 和 UDP 两种方式，TCP 方式支持反向写入，UDP 方式只读，TCP 方式采用变化发送方式，UDP 方式除了变化发送外，每个变量每分钟至少发送一次 UDP 方式一般用于穿透正向隔离装置使用
UDP 方式连续发送	False	默认 False，设置为 True 时每次发送全部变量，只能适合小数据发送
本机端口	0	绑定的本机 TCP 端口，默认 0，使用随机端口，必须指定端口时才需要设置
最大发送变量	500	每次发送的最大变量数，超过设置分包发送（仅对 UPD 方式有效）
本地配置	False	设置为 True 时使用扩展配置，每行一个变量
变量前缀		自动在转发变量的名称前面增加自定义前缀

2.6 版本开始支持单独配置转发变量。

2.4.2 报警事件

使用 JavaScript 脚本执行报警事件或数字量变化事件

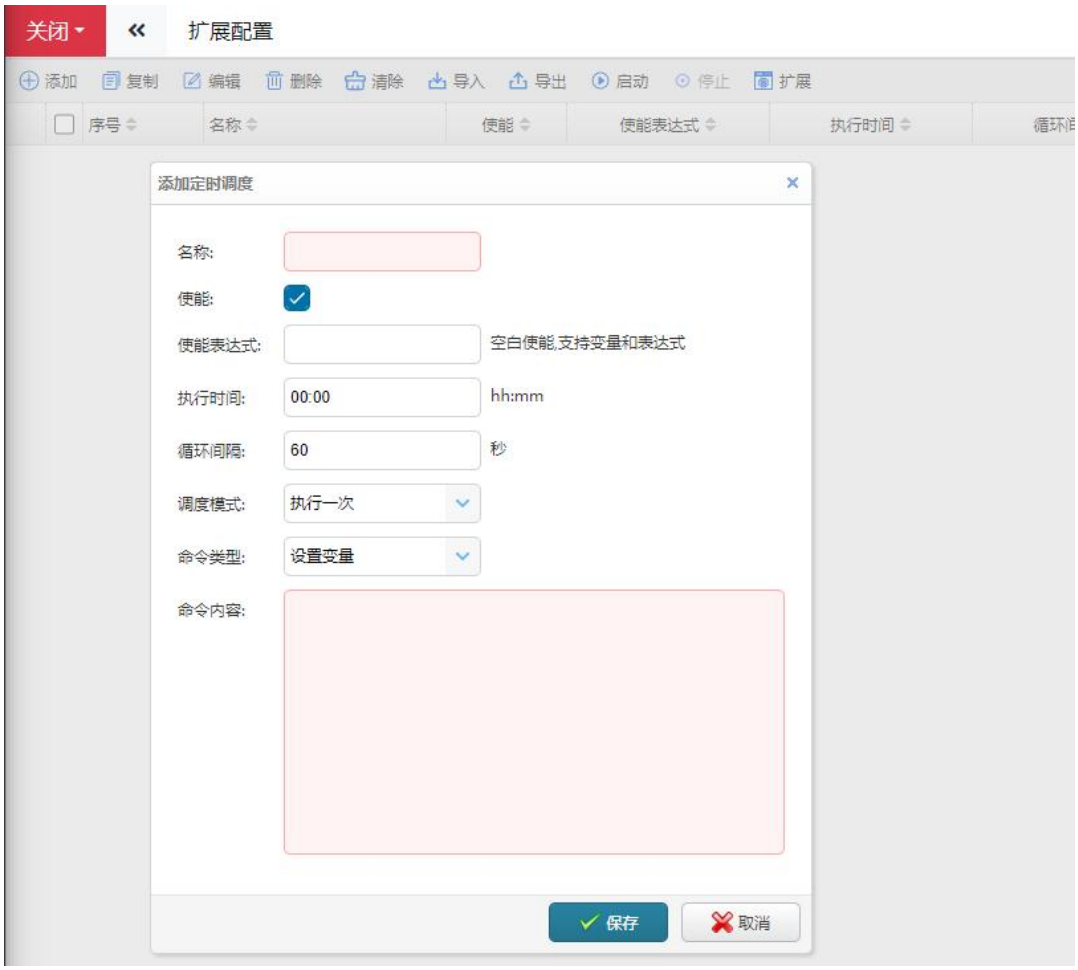


当变量配置了报警，进入报警和离开报警分别执行指定的 js 代码，inalarm 和 outalarm 函数不可以修改；如果变量没有配置报警，那变量必须是数字量类型，变量的值为 true 时执行 inalarm 代码，false 时执行 outalarm 代码。

该扩展工作于 500ms 的线程轮询模式，并非实时执行，无法捕捉到很短的变化过程。

2.4.3 定时调度

用于配置定时任务的执行



使用表达式：空白使能，支持变量名称和表达式，使能才会执行调度

调度模式：执行一次、每天、工作日、每月、连续执行

执行时间：小时:分钟，定时执行时的执行时间

循环间隔：连续执行的间隔时间，单位秒

命令类型：

命令类型	命令内容

设置变量	使用逗号分隔多个设置，例如
	tag1=1,tag2=3,tag3=4
	Tag1,tag2=5,tag3
	没有等于号时写入值 1
	扩展功能（v2.6.10 开始）：
	Tag1=@second //引用变量值
	Tag2=[Second] //引用单个变量值

	Tag3=([Second]+[Minute]) //使用表达式
复位变量	使用逗号分隔多个设置，例如 Tag1,Tag2,Tag3
执行 JS 脚本	JavaScript 脚本，执行 exec()函数，支持使用全局 JS 代码 =codename =codename?@param=xxx **使用线程池异步执行脚本，系统具备脚本未执行完禁止再次执行的功能**
切换变量	使用逗号分隔多个设置，例如 Tag1,Tag2,Tag3 切换变量的值，True 到 False，False 到 True
批量命令	使用逗号分隔多个设置，例如 Batch1=1,Batch2=3 Batch1:批量名称

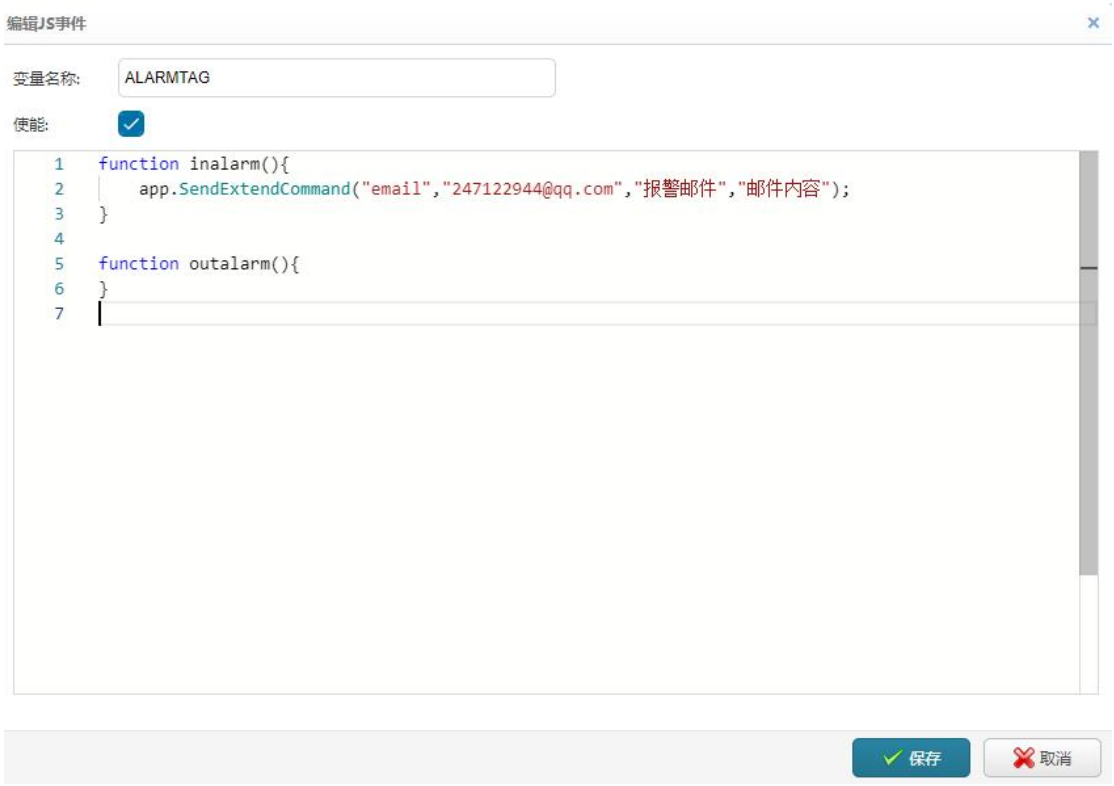
2.4.4 电子邮件发送

提供通过 Smtp 协议发送 Email

参数设置说明

名称	设置值	说明
Smtp 服务器	smtp.qq.com	邮箱 Smtp 服务器地址
邮箱地址	xxx@qq.com	邮箱账号
邮箱密码		登录邮箱的密码
邮箱端口	25	邮箱端口，标准邮箱端口是 25，使用 SSL 时根据邮箱服务器设置
SSL	false	加密连接选项，QQ 邮箱使用 SSL 时端口为 587

启用扩展后，使用 js 脚本发送电子邮件，例如可以配置报警事件发送邮件



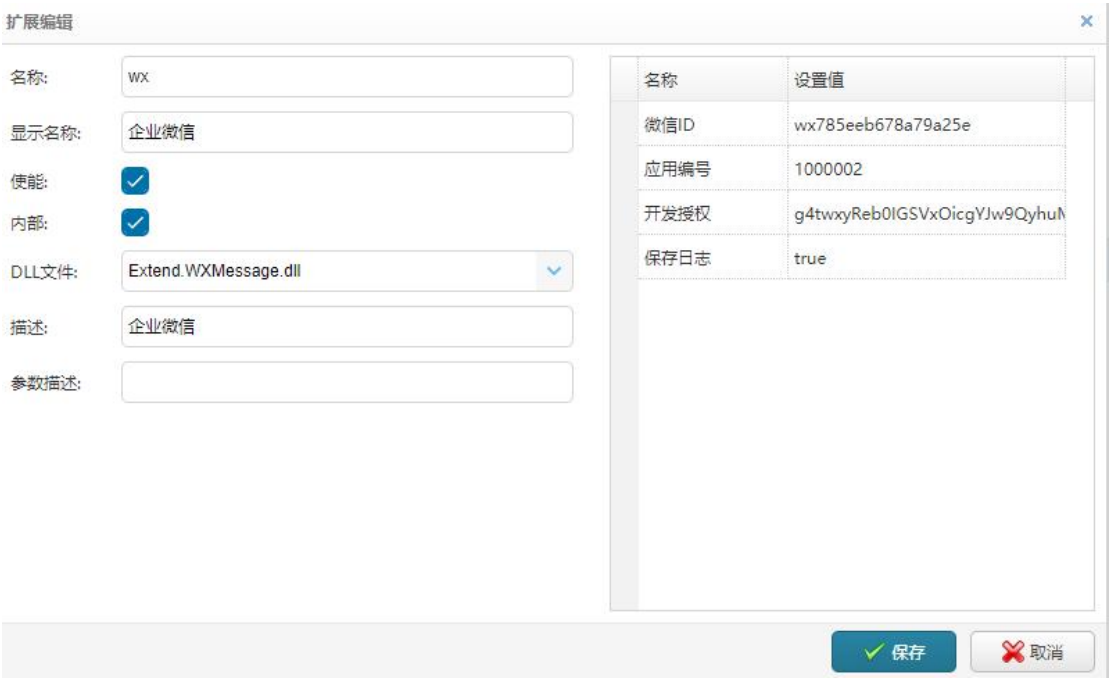
js 的发送命令如下：

```
app.SendExtendCommand("email","xxx@qq.com","报警邮件标题","邮件内容");
```

第 1 个参数是电子邮件扩展的名称，第二个参数是接收者邮箱，多个地址使用分号进行分割。

2.4.5 微信发送

通过企业微信发送信息，企业微信有独立的 App，需要自行注册，只有加入到企业微信的用户才能接收信息。



参数设置说明

名称	设置值	说明
微信 ID	wx...	企业微信账号
应用编号		微信企业号对应应用的编号
开发授权	10001	微信企业号对应应用的开发授权
保存日志	false	是否记录发送日志

js 的发送命令如下：

```
app.SendExtendCommand("wx","admin","微信内容","");
```

第 1 个参数是微信扩展的名称，第二个参数是接收者名称或者部门名称或者标签名称，使用|分隔多个名称。

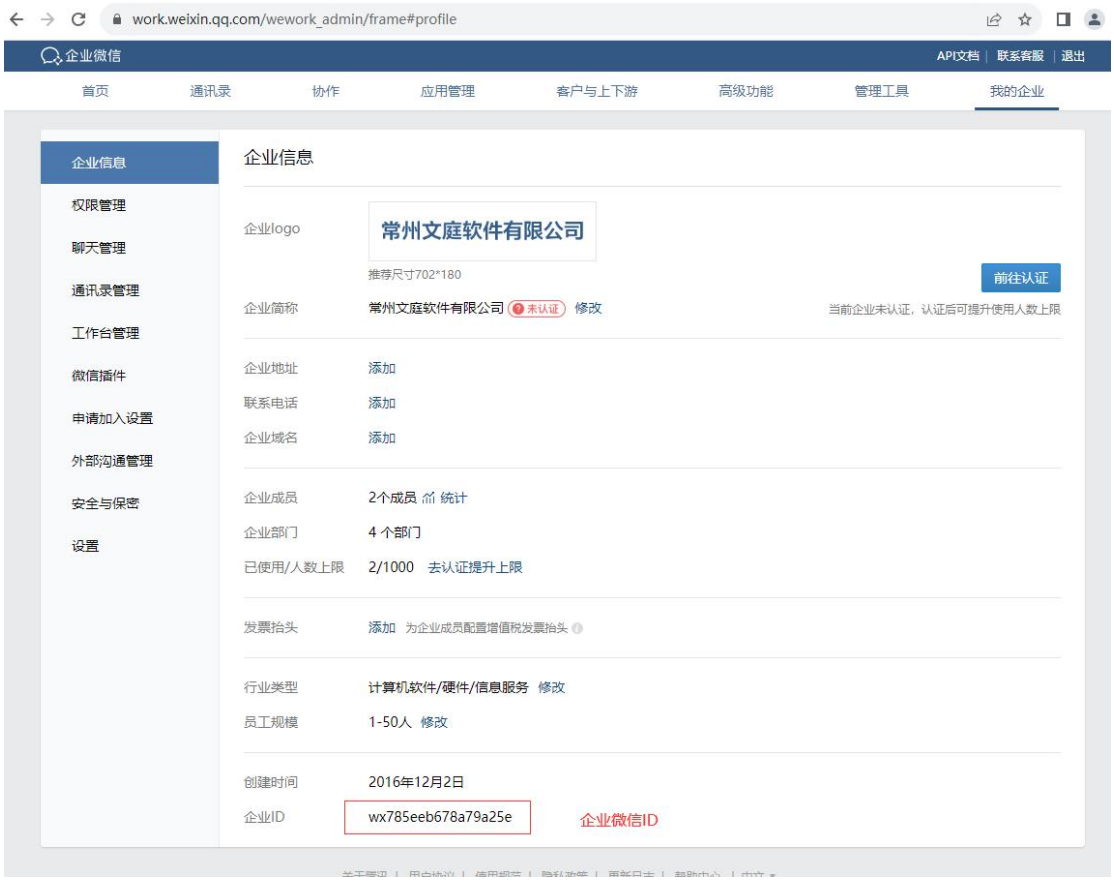
例如：party:xxx|tag:yyy|user1|user2

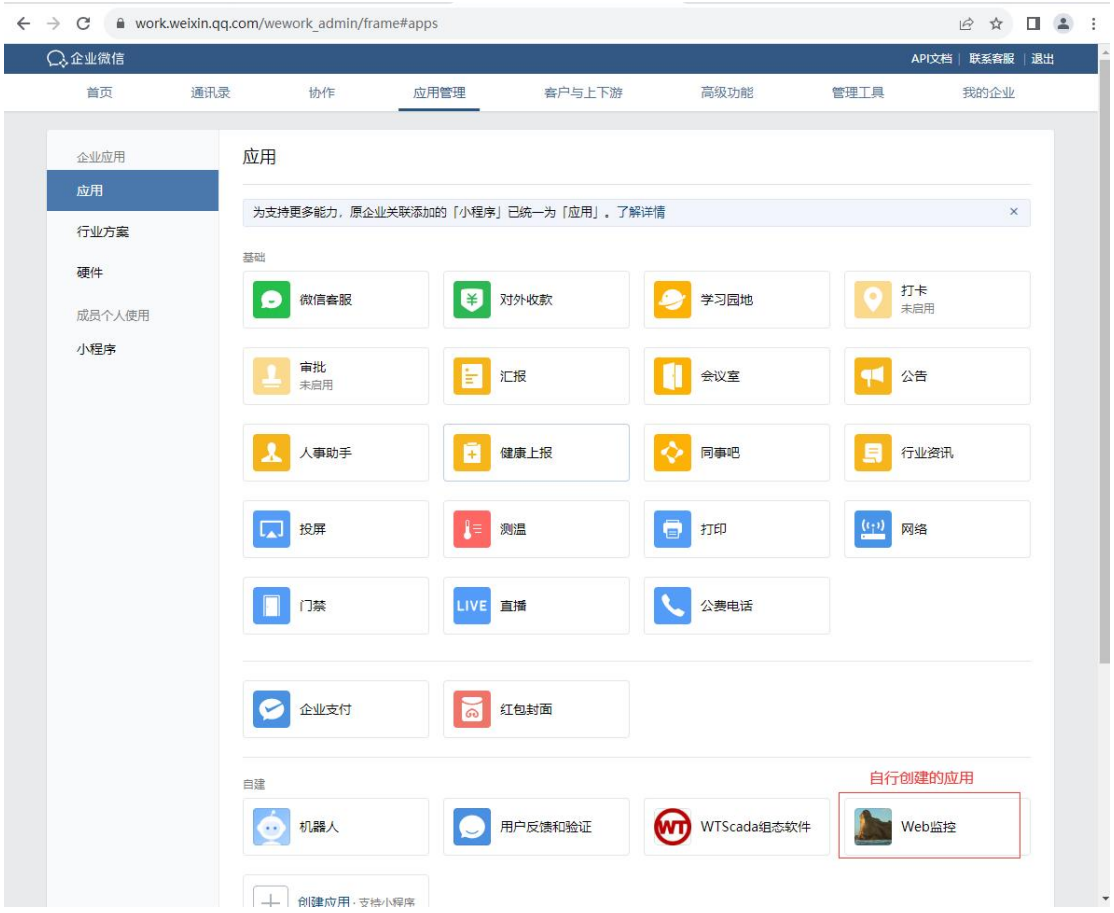
发送到部门为 xxx 和标签为 yyy 和用户名为 user1 和 user2 的用户(用户名为企业微信账号里面的名称，不是微信账号或手机号码)

@all 表示发送全部用户

发送限制：每应用不可超过账号上限数*200 人次/天，每成员不可超过 30 次/分钟。

企业微信网址：https://work.weixin.qq.com/

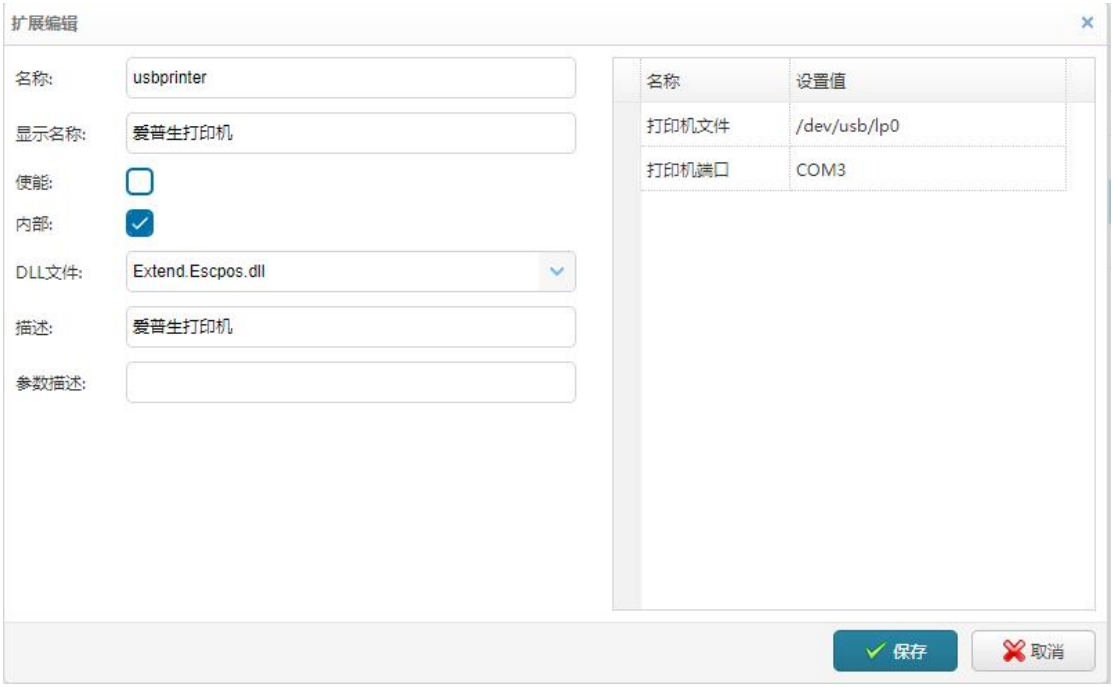

work.weixin.qq.com/wework_admin/frame#profile
企业微信
API文档 | 联系客服 | 退出
首页
通讯录
协作
应用管理
客户与上下游
高级功能
管理工具
我的企业
企业信息
权限管理
聊天管理
通讯录管理
工作台管理
微信插件
申请加入设置
外部沟通管理
安全与保密
设置
企业信息
企业logo
常州文庭软件有限公司
推荐尺寸702*180
前往认证
企业简称
常州文庭软件有限公司 未认证 修改
当前企业未认证，认证后可提升使用人数上限
企业地址
添加
联系电话
添加
企业域名
添加
企业成员
2个成员 统计
企业部门
4 个部门
已使用/人数上限
2/1000 去认证提升上限
发票抬头
添加 为企业成员配置增值税发票抬头
行业类型
计算机软件/硬件/信息服务 修改
员工规模
1-50人 修改
创建时间
2016年12月2日
企业ID
wx785eeb678a79a25e
企业微信ID


work.weixin.qq.com/wework_admin/frame#apps
企业微信
API文档 | 联系客服 | 退出
首页
通讯录
协作
应用管理
客户与上下游
高级功能
管理工具
我的企业
企业应用
应用
行业方案
硬件
成员个人使用
小程序
应用
为支持更多能力，原企业关联添加的「小程序」已统一为「应用」。了解详情
基础
微信客服
对外收款
学习园地
打卡
未启用
审批
未启用
汇报
会议室
公告
人事助手
健康上报
同事吧
行业资讯
投屏
测温
打印
网络
门禁
直播
公费电话
企业支付
红包封面
自建
自行创建的应用
机器人
用户反馈和验证
WTScada组态软件
Web监控
创建应用 支持小程序

只有具备应用访问权限的用户才能收到信息。

2.4.6 爱普生票据打印机

爱普生票据打印机是 1 个扩展，通过 JS 脚本进行控制，可以根据需要输出打印内容。



参数设置说明

名称	设置值	说明
打印机文件	/dev/usb/lp0	Linux 下打印机文件名称
打印机端口	COM3	Windows 下打印机端口

举例：

```
var extPrint = app.GetExtend("usbprinter");//扩展名称
var enableTag = $Tag("PrintEnable");//打印使能变量
var count=0;//打印次数
function update(){
  if (extPrint){
  if (enableTag && enableTag.BooleanValue){
      enableTag.DoUpdate(false);
      extPrint.DoComamnd("CenterAlign","",""); //居中
      extPrint.DoComamnd("Styles","Bold",""); //粗体
      extPrint.DoComamnd("printline","title",""); //打印
      extPrint.DoComamnd("LeftAlign","",""); //居左
      extPrint.DoComamnd("Styles","None",""); //取消粗体
      extPrint.DoComamnd("printline","line1","");//打印
      extPrint.DoComamnd("printline","line2","");//打印
      extPrint.DoComamnd("cut","","");//分隔
      extPrint.DoComamnd("end","","");//发送到打印机
      count ++;
  }
  return count;
}
else
```

```
    return -1;
}
```

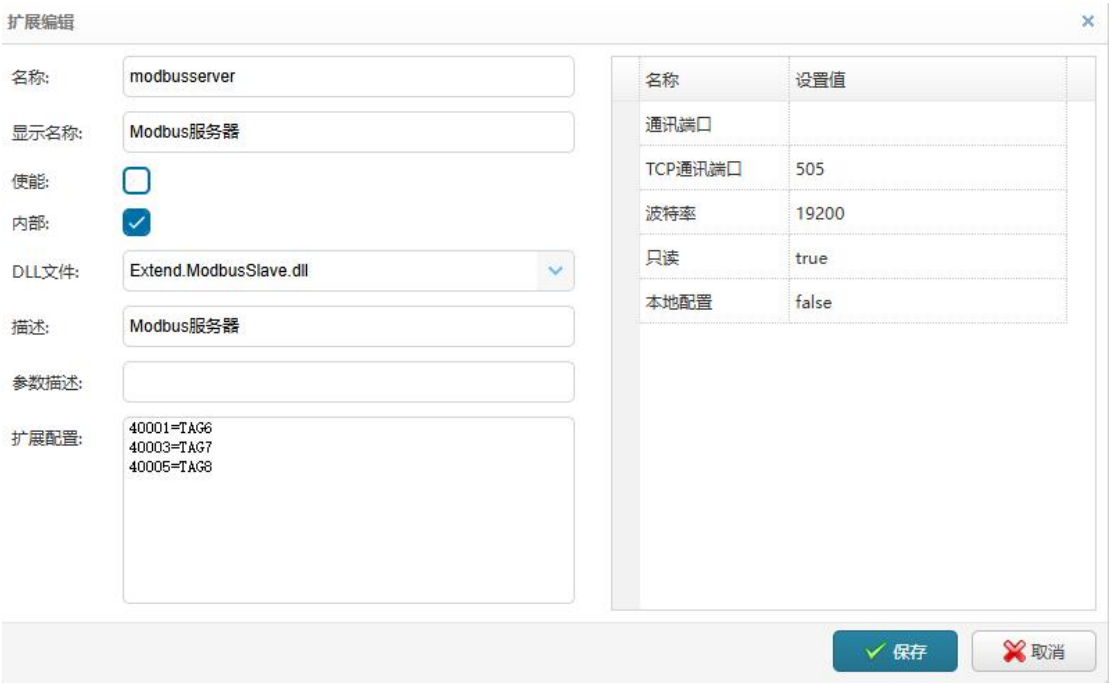
支持的命令列表

命令	参数	说明
LeftAlign		左对齐
CenterAlign		中心对齐
RightAlign		右对齐
LeftAlignAlt		左对齐
CenterAlignAlt		中心对齐
RightAlignAlt		右对齐
Styles	None FontB Proportional Condensed Bold DoubleHeight DoubleWidth Italic Underline	设置打印格式
print	打印内容	打印
printline	打印内容	打印并换行
cut		切纸
feedline		空行
fullcut		完整切纸
codepage	PC437_USA_STANDARD_EUROPE_DEFAULT KATAKANA PC850_MULTILINGUAL PC860_PORTUGUESE PC863_CANADIAN_FRENCH PC865_NORDIC HIRAGANA ONE_PASS_KANJI ONE_PASS_KANJI2 PC851_GREEK PC853_TURKISH PC857_TURKISH PC737_GREEK ISO8859_7_GREEK WPC1252 PC866_CYRILLIC2 PC852_LATIN2 PC858_EURO	语言代码

	KU42_THAI TIS11_THAI TIS13_THAI TIS14_THAI TIS16_THAI TIS17_THAI TIS18_THAI TCVN3_VIETNAMESE_L TCVN3_VIETNAMESE_U PC720_ARABIC WPC775_BALTIC_RIM PC855_CYRILLIC PC861_ICELANDIC PC862_HEBREW PC864_ARABIC PC869_GREEK ISO8859_2_LATIN2 ISO8859_15_LATIN9 PC1098_FARSI PC1118_LITHUANIAN PC1119_LITHUANIAN PC1125_UKRANIAN WPC1250_LATIN2 WPC1251_CYRILLIC WPC1253_GREEK WPC1254_TURKISH WPC1255_HEBREW WPC1256_ARABIC WPC1257_BALTIC_RIM WPC1258_VIETNAMESE KZ1048_KAZAKHSTAN DEVANAGARI BENGALI TAMIL TELUGU ASSAMESE ORIYA KANNADA MALAYALAM GUJARATI PUNJABI MARATHI	
end		输出到打印机
clear		清除打印内容

2.4.7 Modubs 从站扩展

Modbus 从站扩展提供 ModbusTCP 和 ModbusRTU 输出功能，通过在变量配置上启用寄存器地址。



当本地配置启用后，使用扩展配置，每行一个变量配置。

支持标准寄存器 0x，1x，3x，4x，站号固定为 1

0x，1x 用于布尔量变量，内存地址相同

3x，4x 用于模拟量，内存地址相同，根据变量数据类型自动设置寄存器长度

Int32，Single 使用 2 个寄存器地址

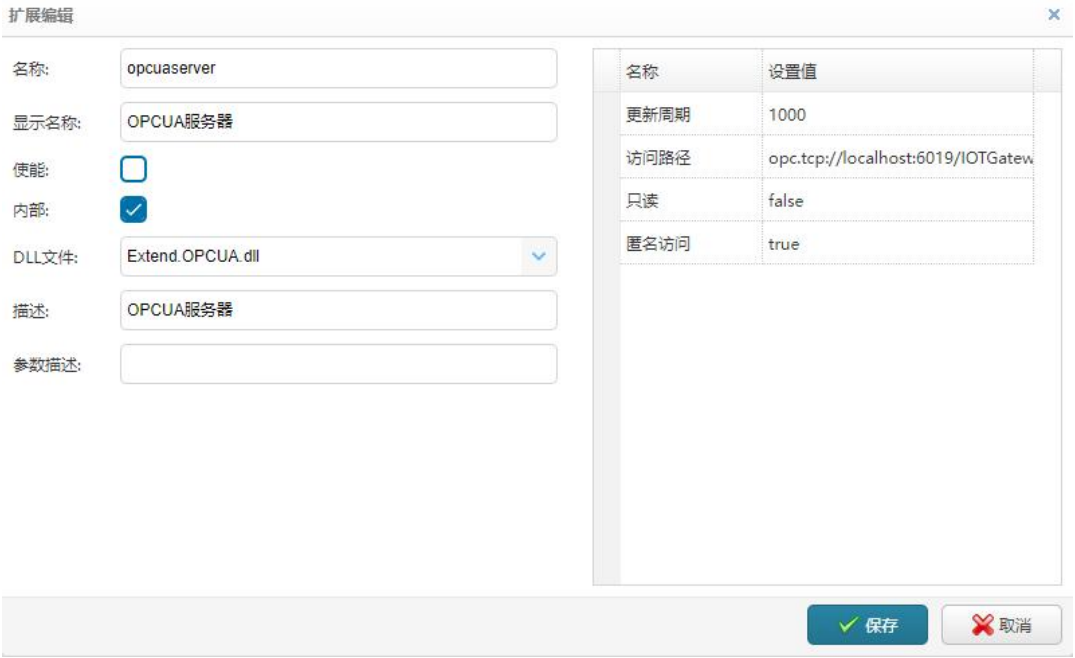
Int64，Double 使用 4 个寄存器地址

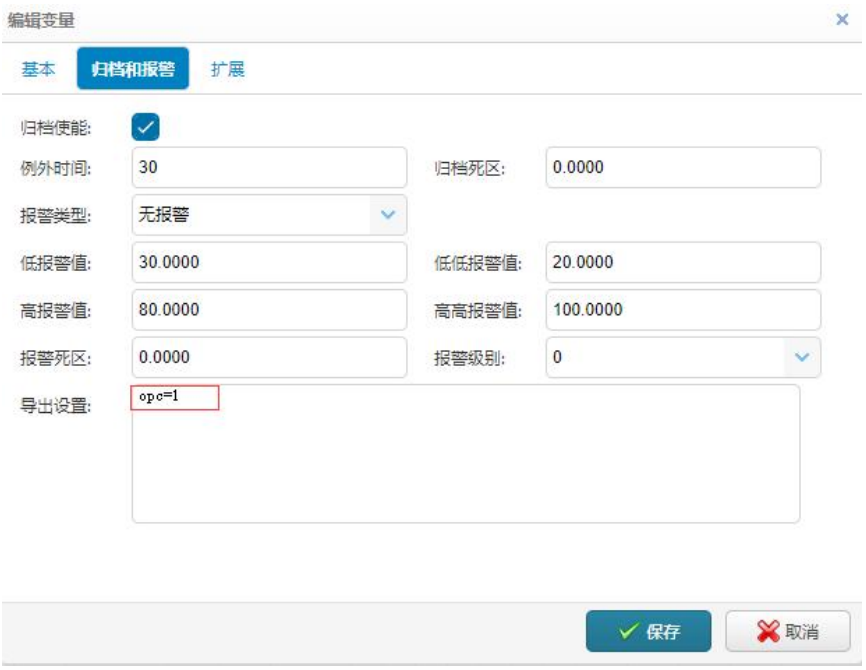
目前不支持字符变量

对于 Linux 系统使用 1024 以下的端口需要 root 权限执行，如果使用 502 端口没有使用 root 权限执行，服务可能无法启动。解决的方法是使用系统服务方式运行或者使用 root 权限运行，或者使用大于 1024 的端口。

2.4.8 OPCUA 服务器

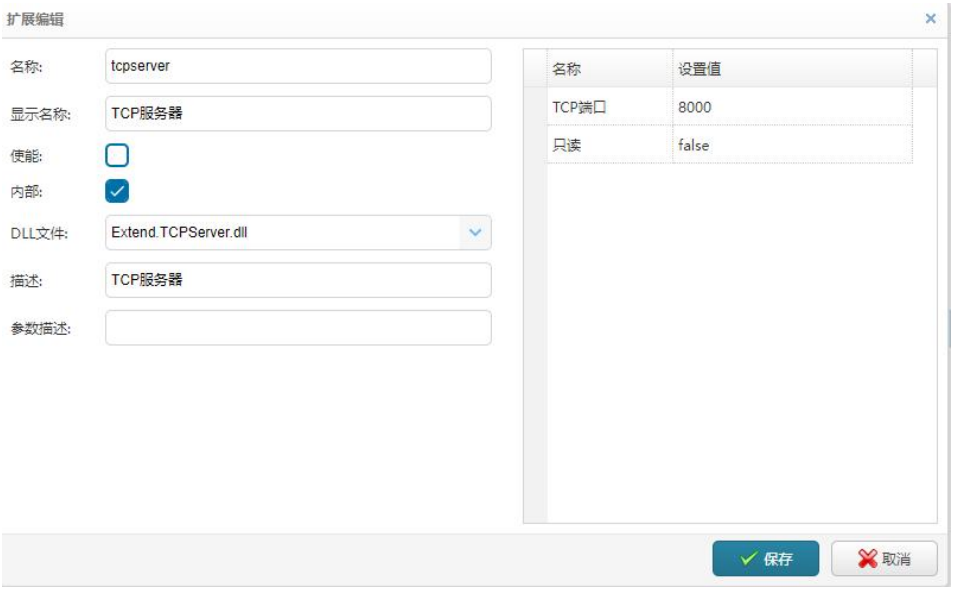
提供 OPCUA 服务，可以配置访问路径和是否只读，用户认证模式使用内置用户名和密码验证，匿名模式变量只读。





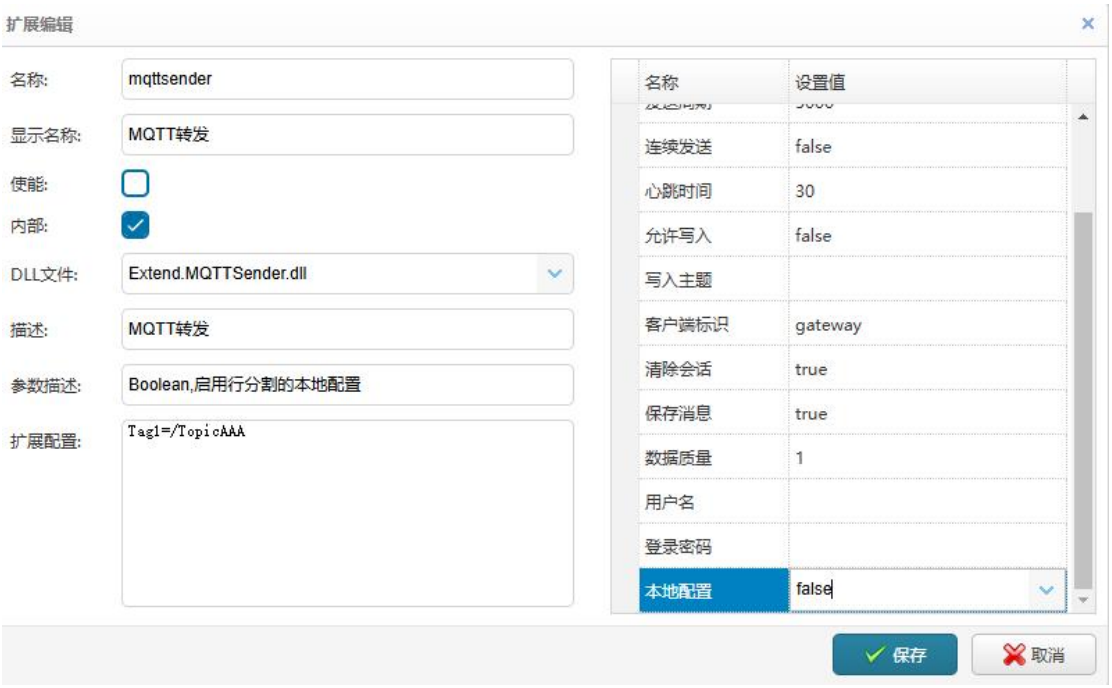
变量配置的导致设置增加一行 opc=1

2.4.9 TCP 服务器扩展



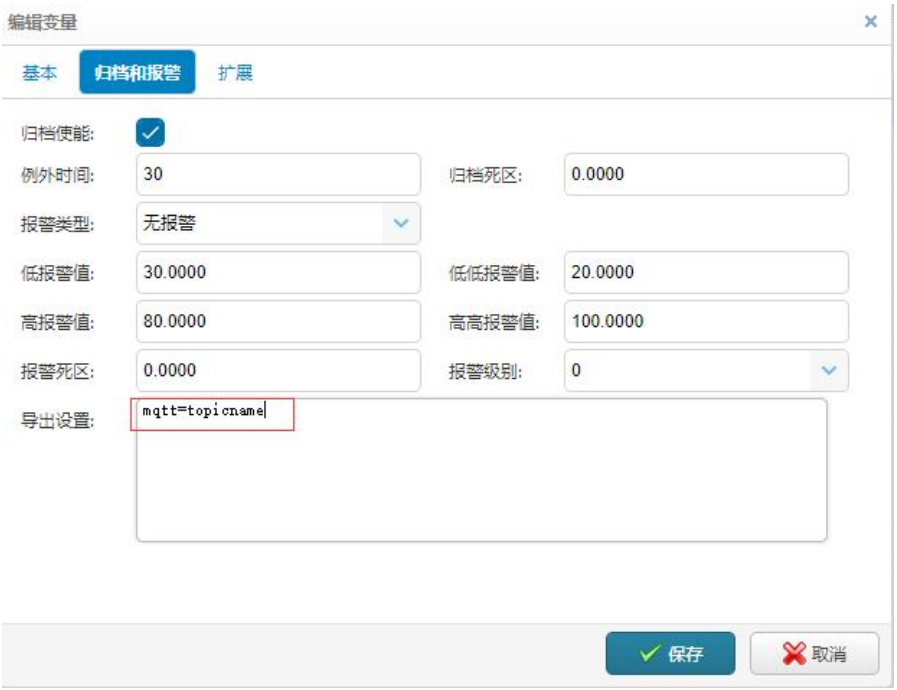
TCP 服务器扩展用于 TCP 驱动采集数据，支持 FScada、WTGateway 的 NetTCP 驱动，支持 WTOPCServer。

2.4.10 MQTT 发送扩展



提供 MQTT 转发服务

当本地配置启用后使用扩展配置，每行 1 个变量，否则使用变量上的导出设置。默认情况每次仅当值发生变化才发送数据，设置连续发送为 True 则定时发送全部数据。



在变量配置上启用

转发数据格式兼容 Kepware 的精简格式

[{“id”:“tagname”,“v”: 1, “q”:true,“t”:1452222}]

配置写入主题后支持反向写入，格式为

[{“id”:“tagname”,“v”:1}]

当 JS 脚本文件设置后，MQTT 发送内容使用 JS 生成，JS 函数名称为 update，返回内容为 String 类型。

参考 JS 代码如下：

```
function update(tags){
    var length = tags.Length;
    //var strTime = new Date().getTime();
    var result=[];
    for (i=0;i<length;i++){
        result.push({"tag":tags[i].id,"value":tags[i].v,"time":tags[i].t});
    }
    return JSON.stringify(result);
}
```

2.4.11 JYC311 短信发送扩展

支持石家庄市静远电子科技有限公司（www.jycomm.cn）JYC311-4G-RJ45 产品。

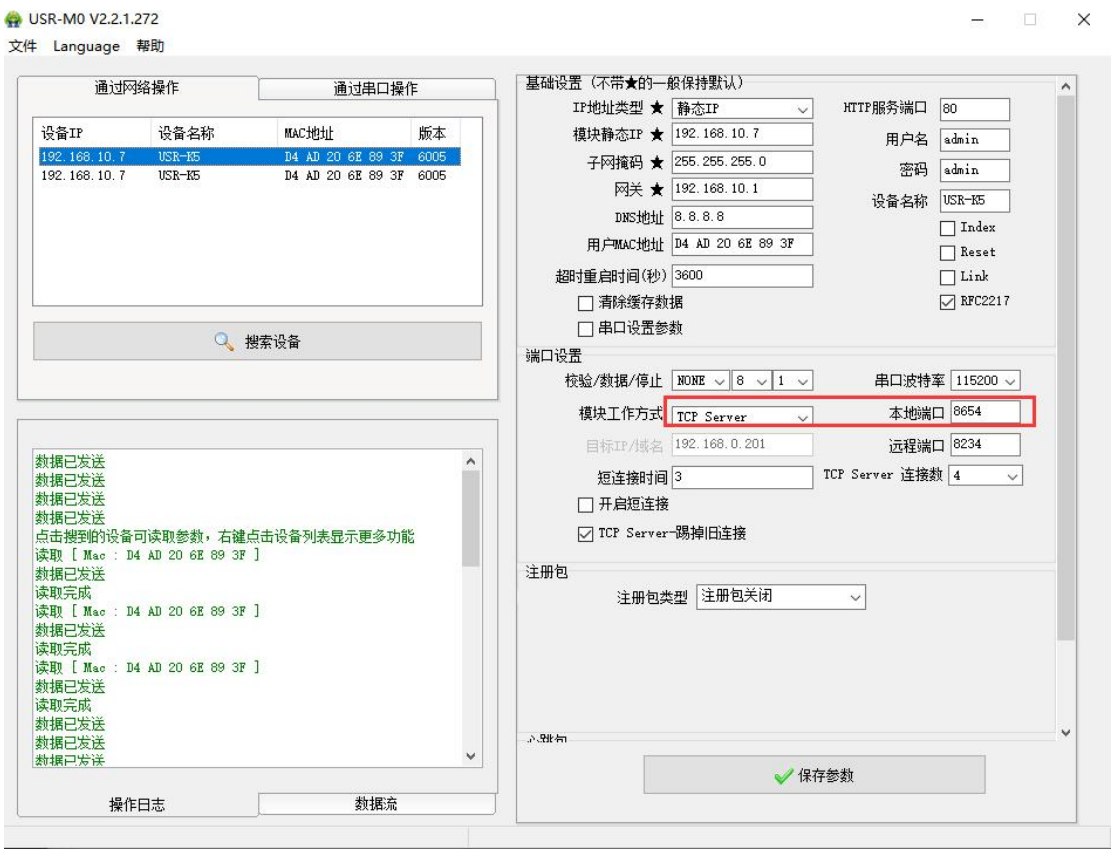
配置网口模块为 TCPServer 方式。

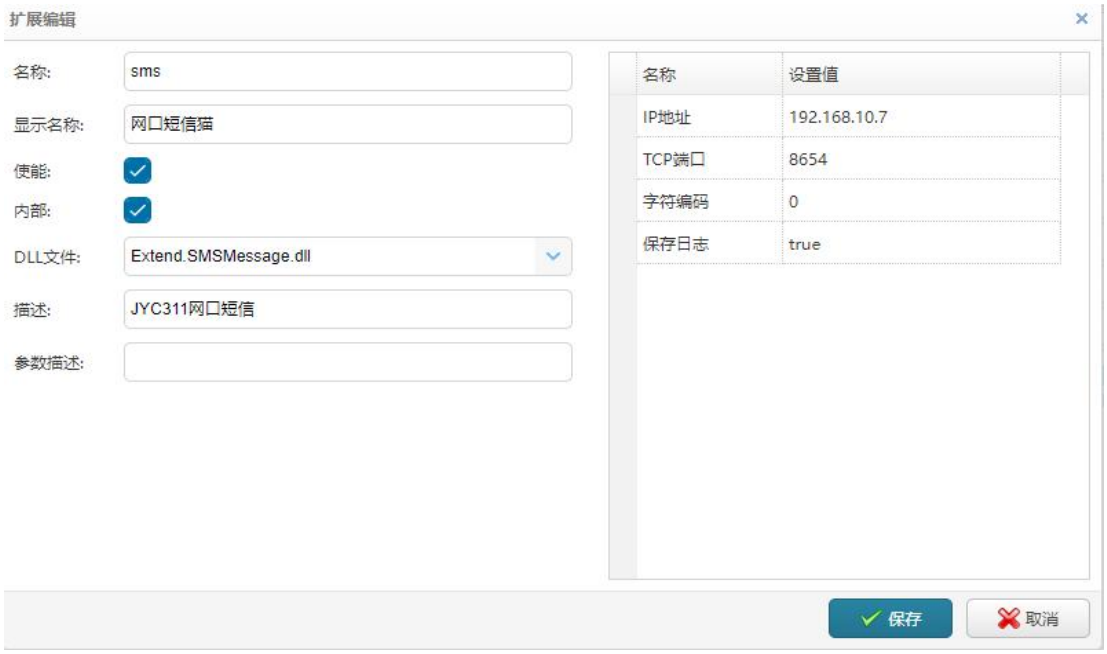
设备默认 IP 地址为 192.168.0.7/192.168.0.1/255.255.255.0，支持 Web 浏览器访问(admin,admin)。

注意：如果运行中拔出了 Sim 卡，重新插入后需要重启模块才能再次发送短信。

短信的发送速度大概 6 秒每条，短信模块发送失败重试 2 次，扩展发送失败重试 3 次，发送失败时每条短信用时大于 20 秒。

建议使用定时调度每天定时发送一条短信，验证短信模块工作正常。





参数设置说明

名称	设置值	说明
串口名称		非空白使用串口，如 COM1
波特率	19200	通讯波特率 8N1 格式
IP 地址	192.168.0.7	短信模块 IP 地址
TCP 端口	8654	TCP 服务端口
字符编码	0	GBK 编码

保存日志	false	是否记录发送日志

js 的发送命令如下：

```
app.SendExtendCommand("sms","18661151798","报警信息");
app.SendExtendCommand("sms","18661151798,13001111111","报警信息");
```

发送失败会记录到日志中，如果保存日志设置为 true，发送信息会记录到系统日志中。

短信模块指示灯状态：

Power 指示灯：加电后常亮。

Run 指示灯：　设备 1 秒亮，1 秒灭

Online 指示灯：指示设备入网情况，状态参见下表。

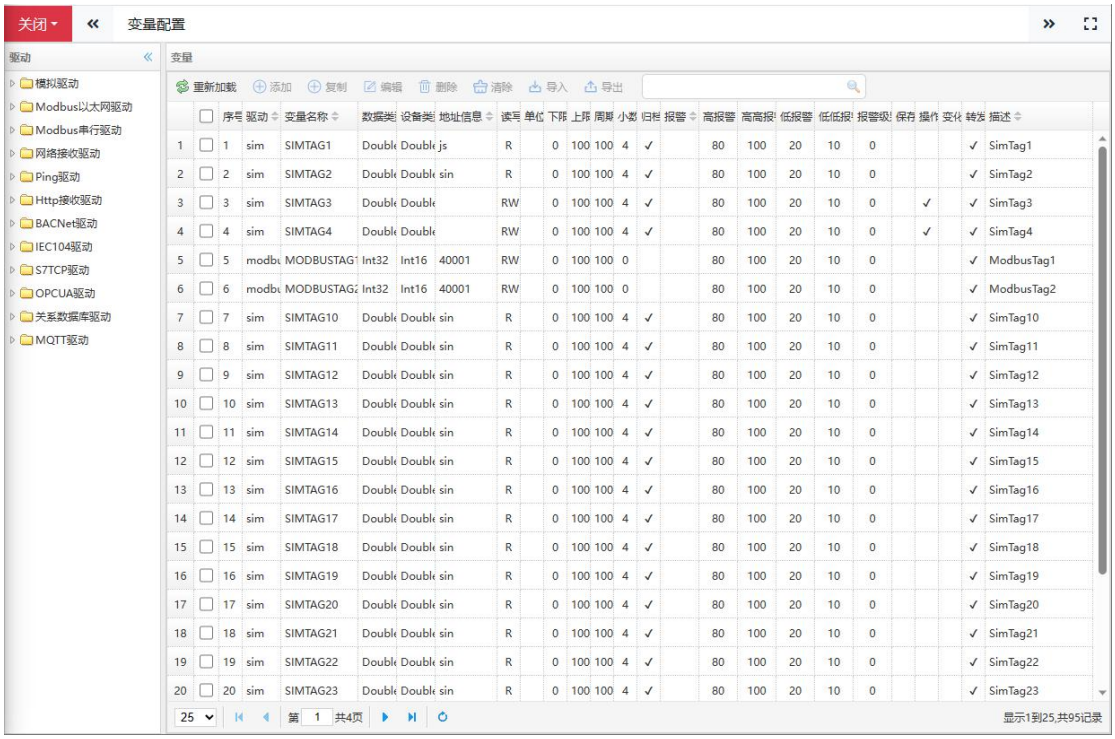
现 象	说 明
不亮	送厂家维修
200ms　亮/1800ms 灭	表示没有 sim 卡，或者正在查找网络
1800ms 亮/200ms 灭	表示注册到移动网络上了，待机状态
125ms　亮/125ms 灭	有数据传输
常亮	通话中

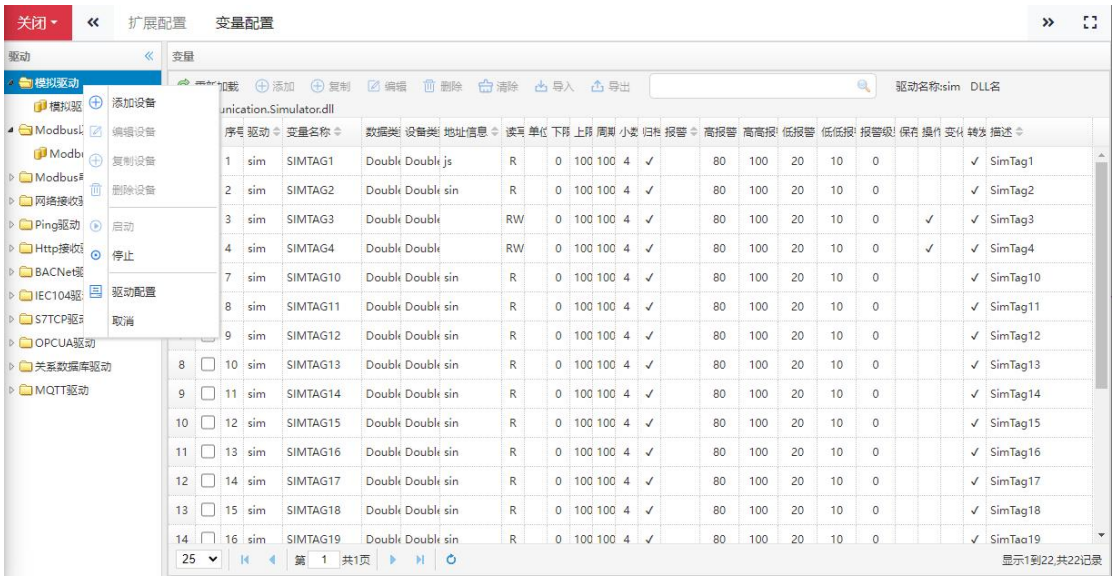
设备加电后 10 秒左右，设备会注册到 GSM 通信网络上。在注册到网络之后，网络指示灯会点亮并有规律闪烁。Net 指示灯状态在 4G 模式下和短信模式下有所不同，注意区分。

网口指示灯	功能	说明
绿灯	连接状态指示	正确连接到网络时绿灯亮。
黄灯	数据指示	模块有数据接收或发送时闪烁，包括模块收到网络广播包。

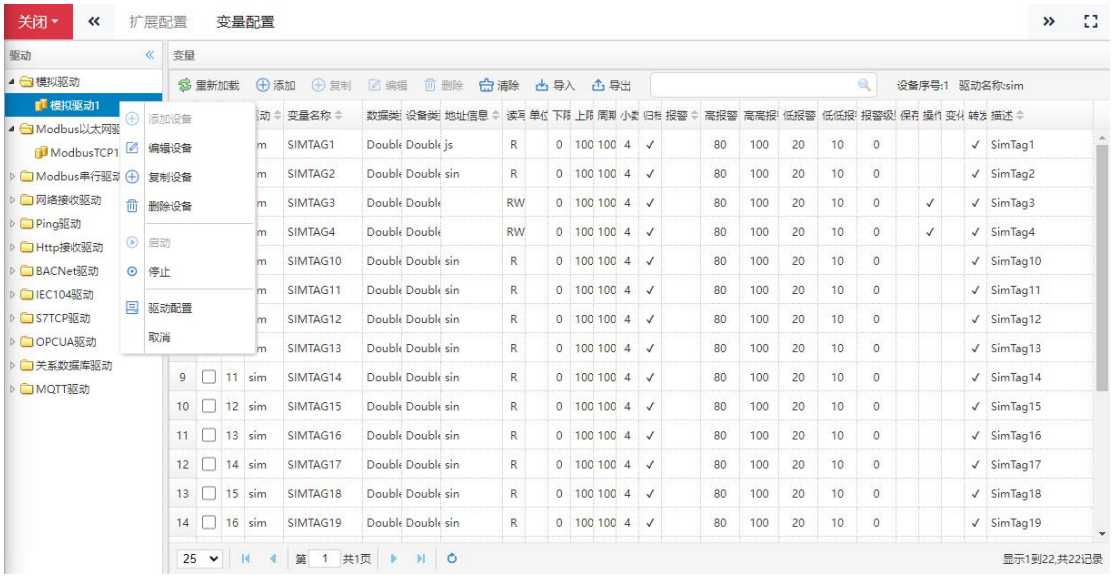
2.5 变量配置

变量配置提供了统一的界面进行设备和变量信息配置。

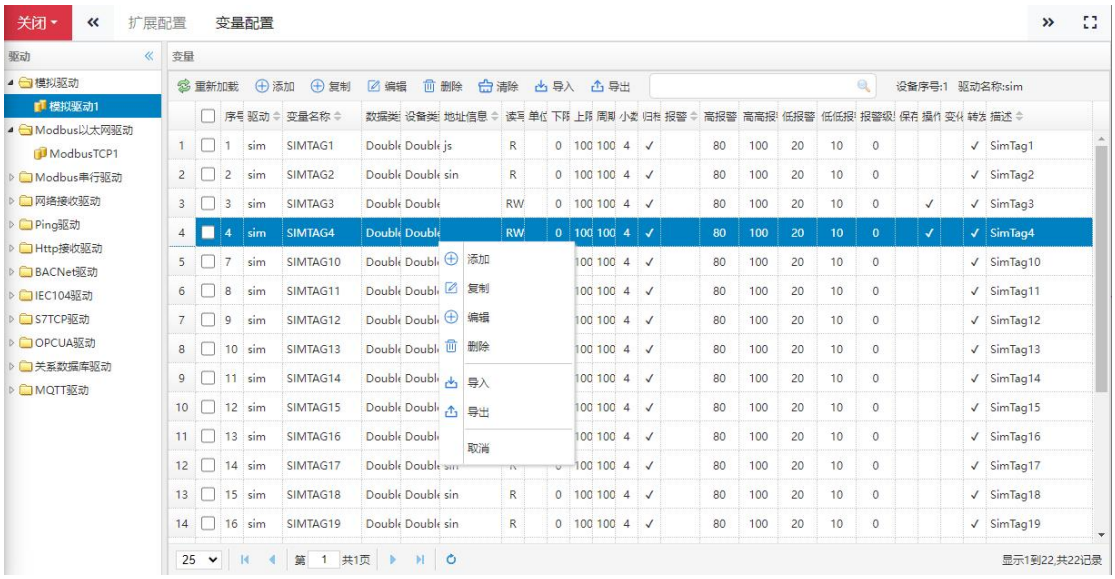




在驱动上鼠标右键可以添加设备



在设备上鼠标右键可以编辑设备、复制和删除设备

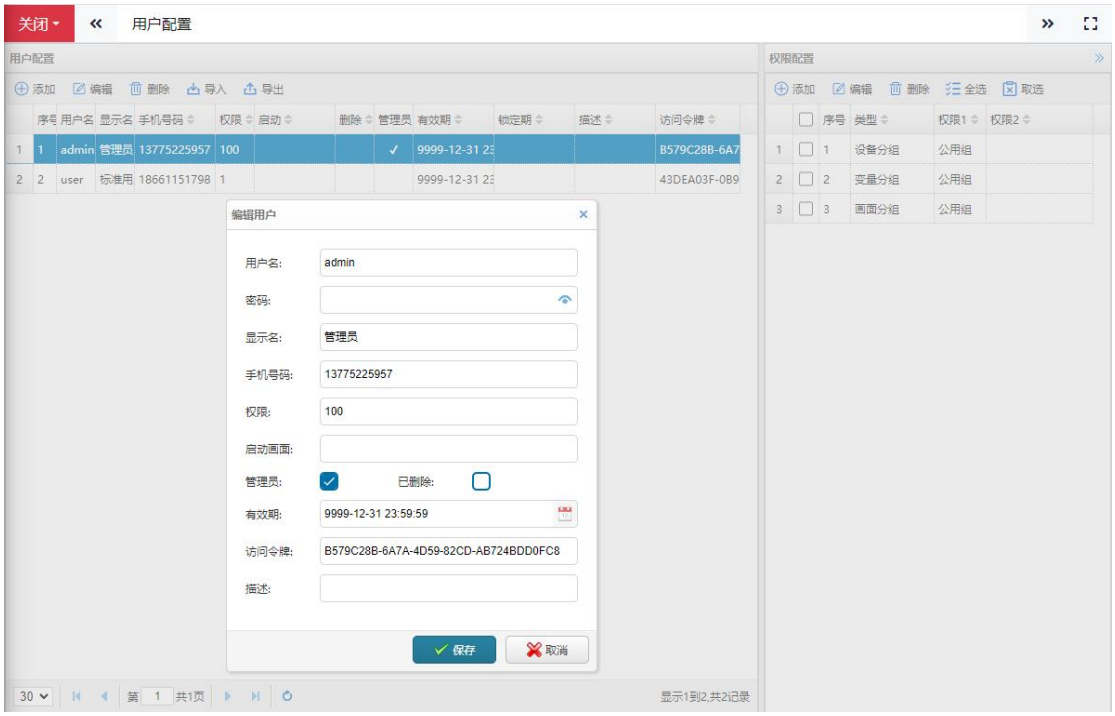


在变量列表界面支持鼠标右键操作

注:变量配置是直接在项目数据库上的配置,所有操作会立即更新到项目数据库,无法撤退,导出功能可以进行备份,导入功能可以从 Excel 格式快速导入变量配置。

导入时 id 值相同的行执行更新,id 为 0 时执行增加,通常导入 Excel 格式均使用这个规则,导入时判定设备 ID 和驱动名称,不相符的数据无法导入。

2.6 用户配置



用户权限值大于等于 100 或者勾选了管理员选项的都是系统管理员，管理员具备访问所有资源的权限。

用户权限值小于等于 0 时不可以执行变量写入操作。

访问令牌用于 token 验证

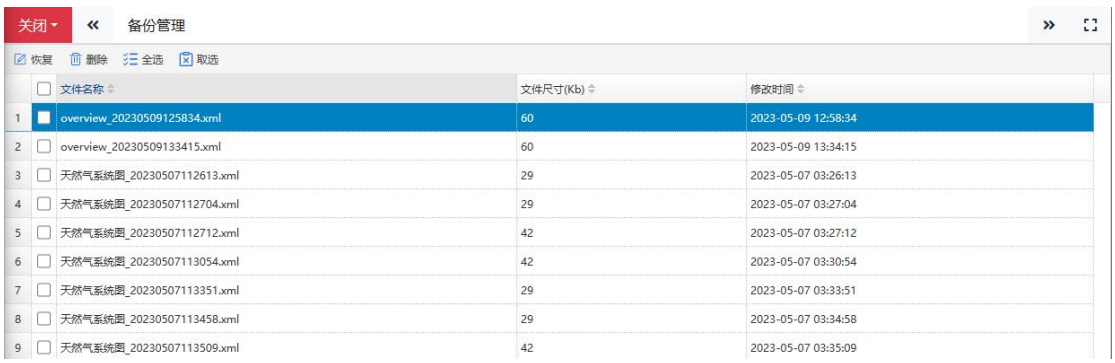
有效期用于控制登录时间范围

启动画面用于配置用户默认 html5 画面

每个用户可以配置多个设备组、变量组和画面组，用于控制用户的对变量、设备和画面的访问范围。

注意：项目发布必须把内置用户的初始密码和访问令牌修改掉。

2.7 备份管理

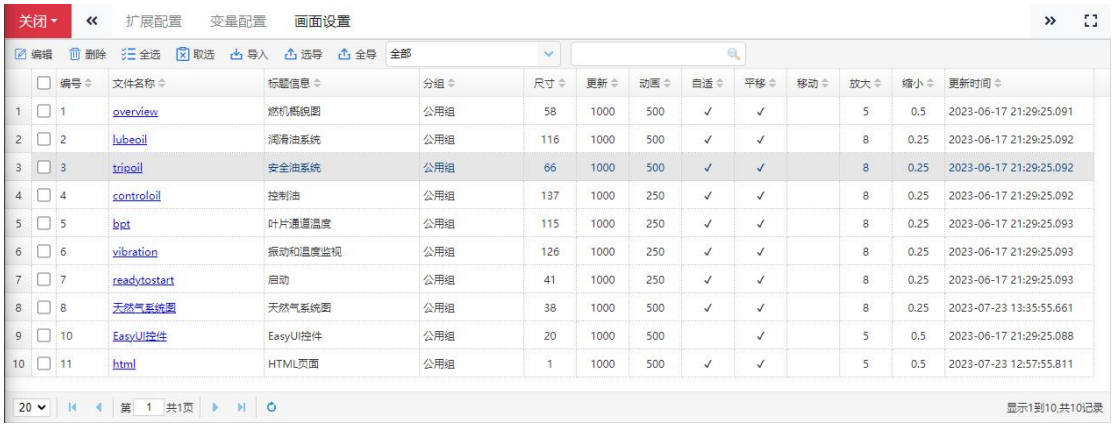


Html5 画面组态每次保存都会在硬盘上产生 1 个备份文件，可以选择从备份文件

中恢复画面，根据需要定期删除旧的备份文件，也可在硬盘上直接删除文件。

备份文件存储位置：软件目录下的 Backup 目录

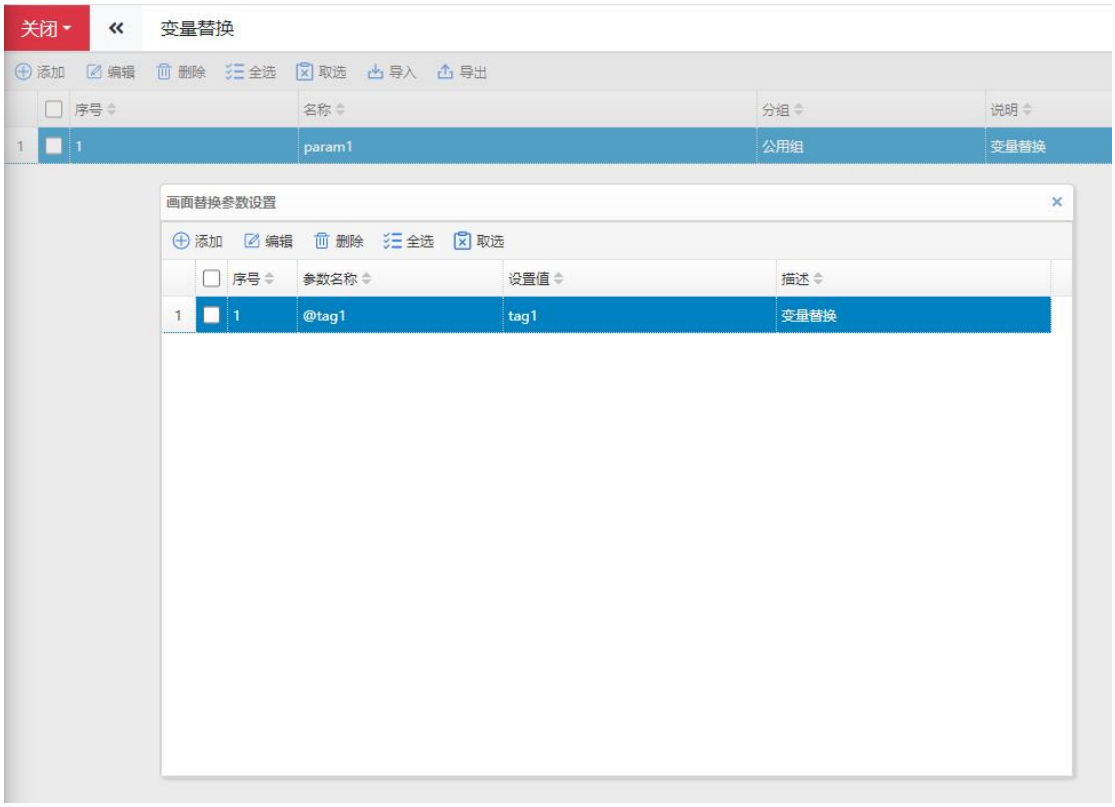
2.7 画面设置



Html5 设置提供了对画面信息的统一设置和删除功能，该界面无法修改画面的内容，点击链接可以打开画面编辑器。

2.8 变量替换

变量替换提供了画面复用的功能，通过配置变量替换文件和替换内容可以实现画面复用，由于画面文件是 1 个 json 格式的文本，替换是基于文本进行的，可以替换的内容包括文字，变量名称和脚本代码。



替换参数必须以@开头，1 个替换文件可以包含多个替换内容，建议尽量减少替换行以减少替换的执行时间。

2.9 RunTime 函数

目录下的 RunTime.dll 提供了用户 C#函数，可以使用 VS2022 编辑后更新该 dll，Global 类定义了 3 个函数：

OnProjectLoad ： 项目运行前执行

OnProjectStart ：项目运行后执行

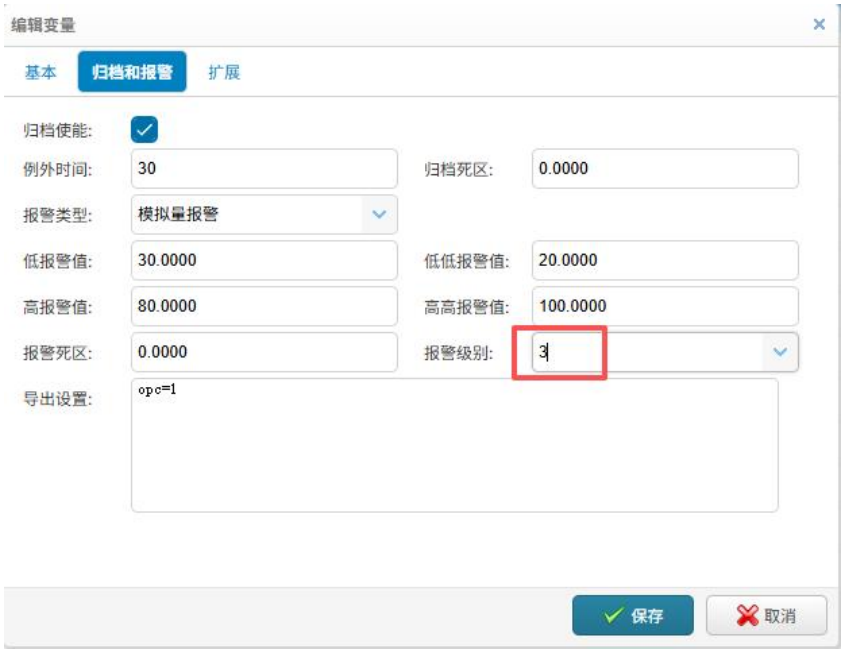
OnProjectStop ： 项目停止前执行

Function 类中的函数可用于模拟驱动。

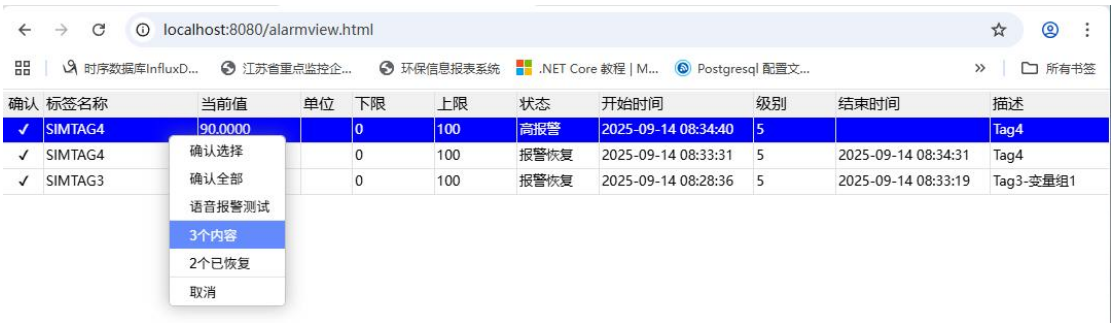
出于安全考虑，这些函数都都是静态函数，如果使用了内存分配，需要自行在停止运行函数中完成内存释放。

2.10 语音报警

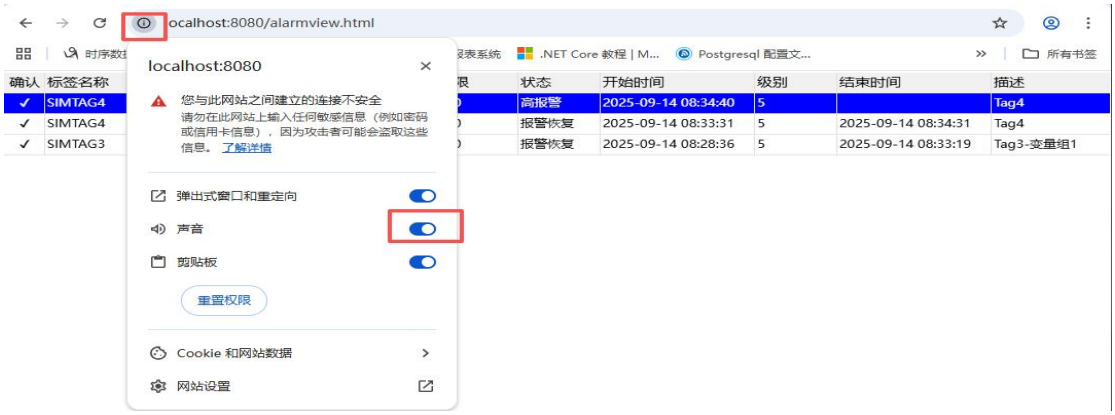
当变量当报警级别大于等于 3 时报警浏览器将会启用语音报警。



alarmview.html 页面是报警浏览器主页面，报警控件就是嵌入了该页面。报警浏览器文字可以直接打开 alarmview.html 文件进行修改。

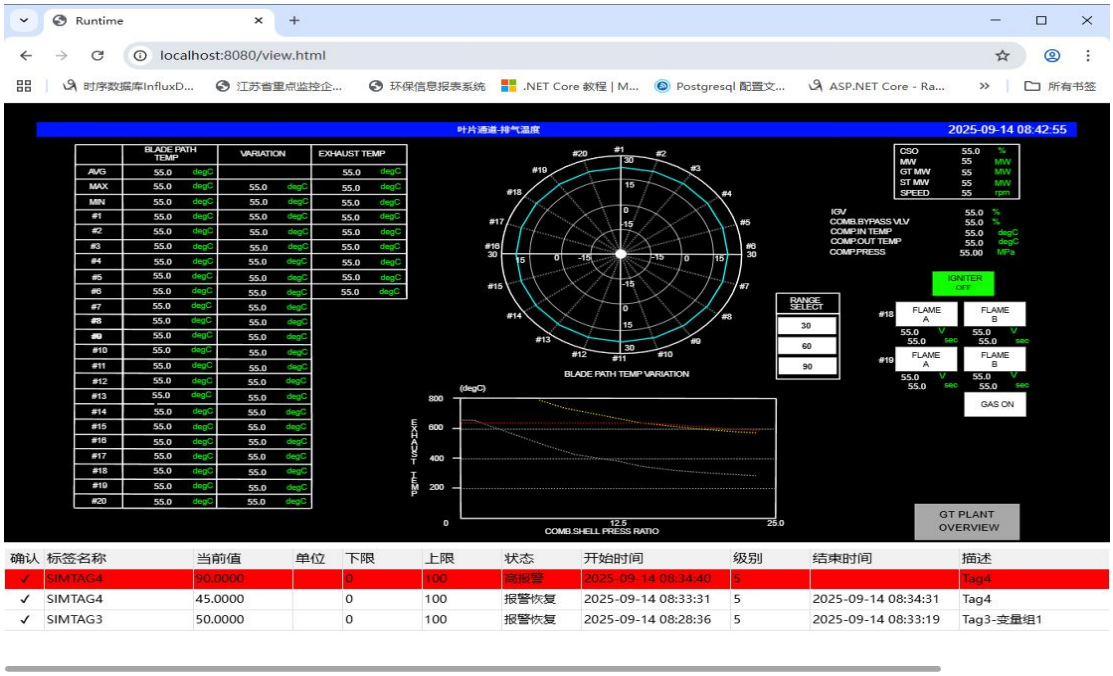


可以通过鼠标右键菜单测试语音报警功能是否正常，如果没有声音则需要检查计算机声音配置和浏览器设置。

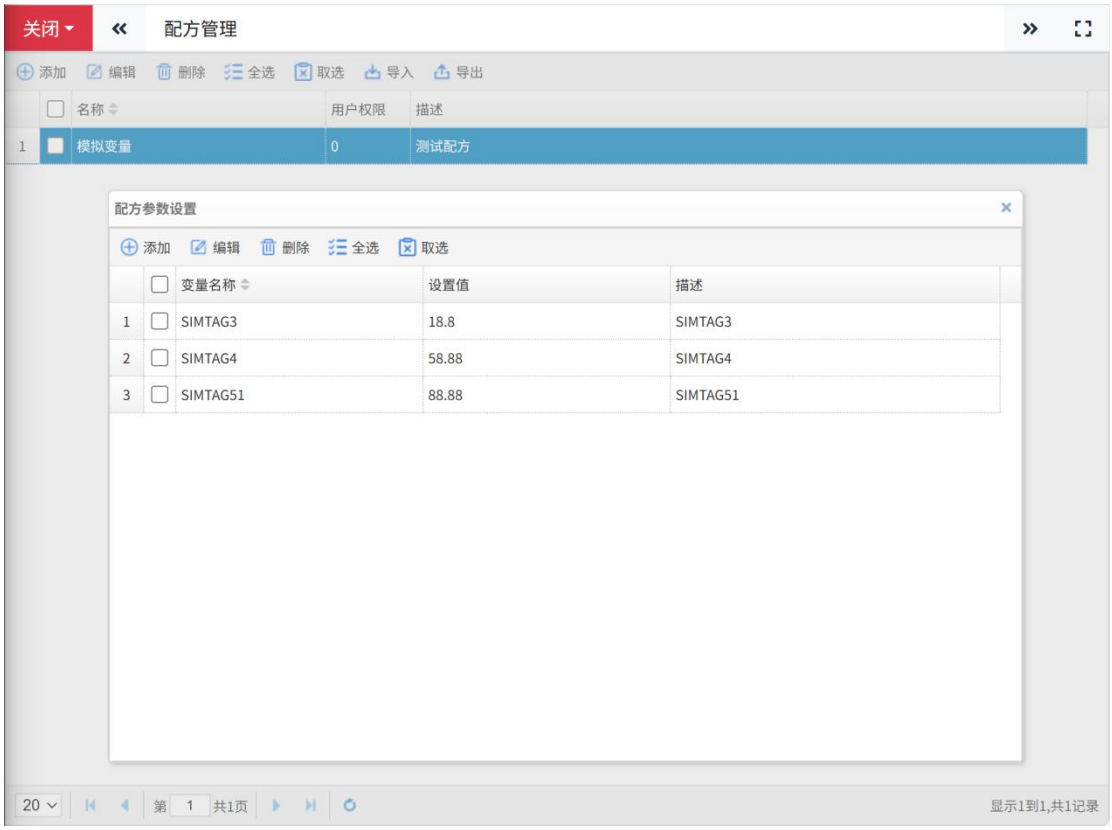


Google 浏览器默认是关闭的，Edge 默认是开启的，修改为开启状态。

对于需要使用报警浏览器的项目，通常建议使用 view.html 页面作为项目主页，该页面采用分割模式报警浏览器显示在底部，切换画面不会影响报警控件。



2.11 配方设置



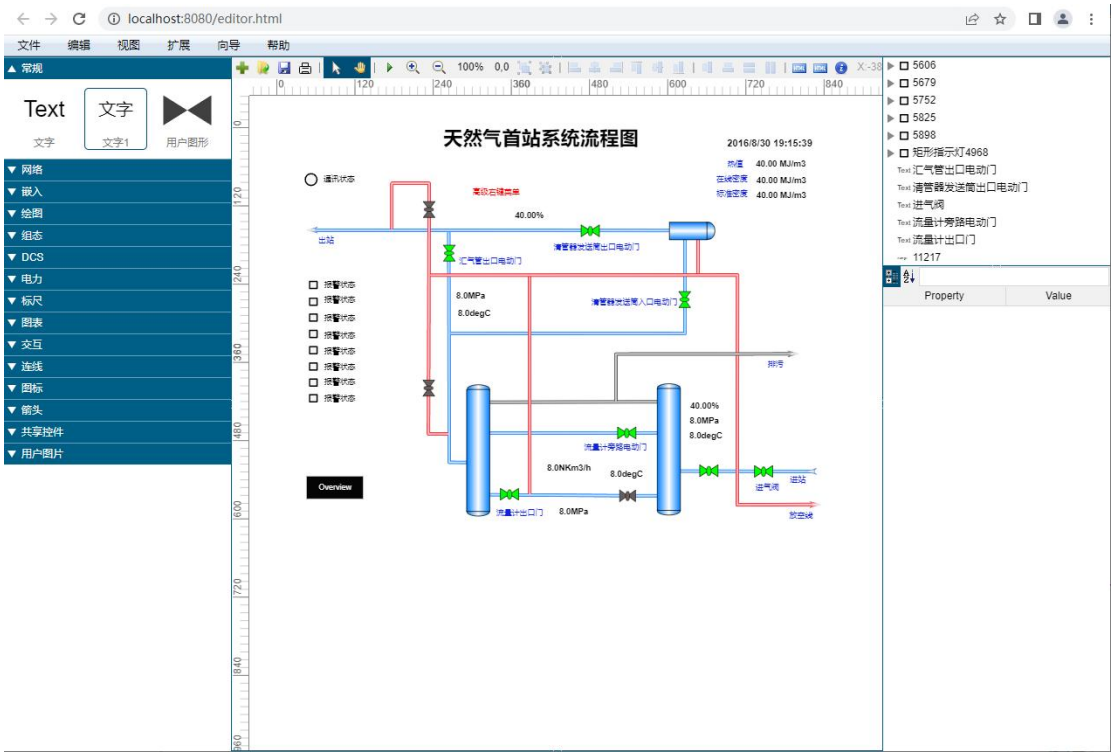
配方功能提供了批量变量设置功能，客户端可以对配方数据进行添加、删除和修改。

3. Web 组态和运行

3.1 组态环境

URL 路径：editor.html(editor-en.html)

注意：系统运行后才能浏览到变量



多语言支持：文本和按钮对象支持多语言设置，需要设置 lan0 默认语言

控件的鼠标提示支持多语言设置，需要设置 lan0 默认语言

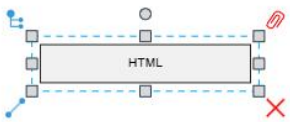
可以 setLan(0-7)函数在线切换画面显示语言，也可以通过浏览器 url 参数 langid 进行切换。

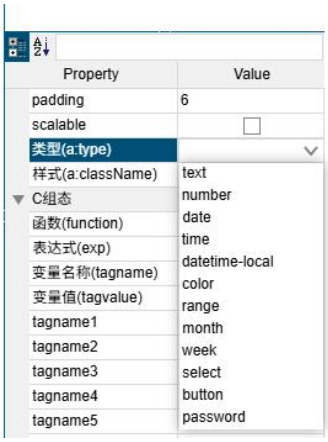
直线和多线段不支持旋转。

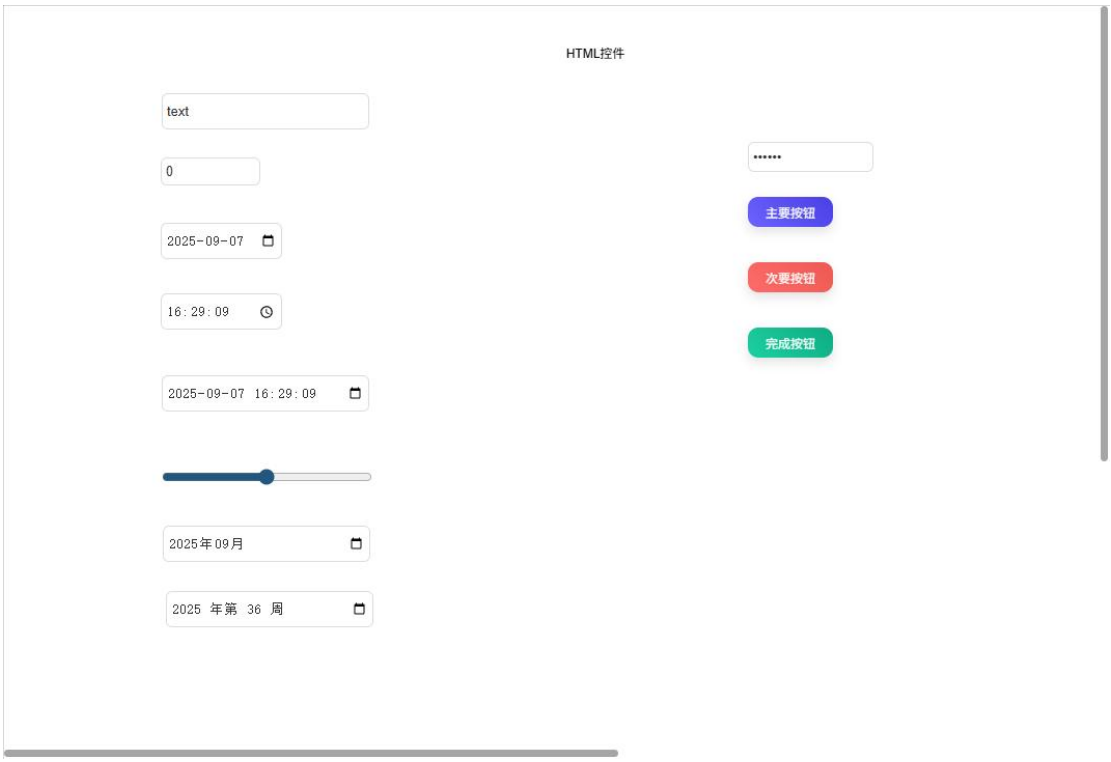
3.1.1 HTML 和 Script 控件数据更新事件

HTML 和 Script 控制从 2.6.11 开始支持绑定变量和数据更新事件,HTML 控件运行时会被替换为 HtmlNode 对象，Script 控件运行时直接创建 js 脚本，自身被隐藏。

3.1.2 HTML5 原生控件







可以通过控件的 element 属性获取原生对象进行操作。

3.1.3 变量写入控件

这样设置后点击控件就会现实输入控件，在输入控件内回车就可以写入数据，再次点击控件可以隐藏输入控件。

3.1.4 设备控制和变量写入

原则上设备操作应该使用单命令方式，组态只对设备下发一次命令，复位命令应该由下位机处理。特殊情况需要发送脉冲建议的操作方式为按下或者弹出鼠标时执行：

```
writeTag(...1);
setTimeout(()=>{writeTag(...0)},500);
```

使用定时器延时 100-500ms 发送复位命令，避免使用按下鼠标写入 1，弹出鼠标写入 0 的方式，因为可能出现弹出鼠标按钮时鼠标离开了控件位置导致弹出事件没有执行。

对于大多数 PLC 直接两条写条命令（具体需要自行测试）

writeTag(...1);

writeTag(...0);

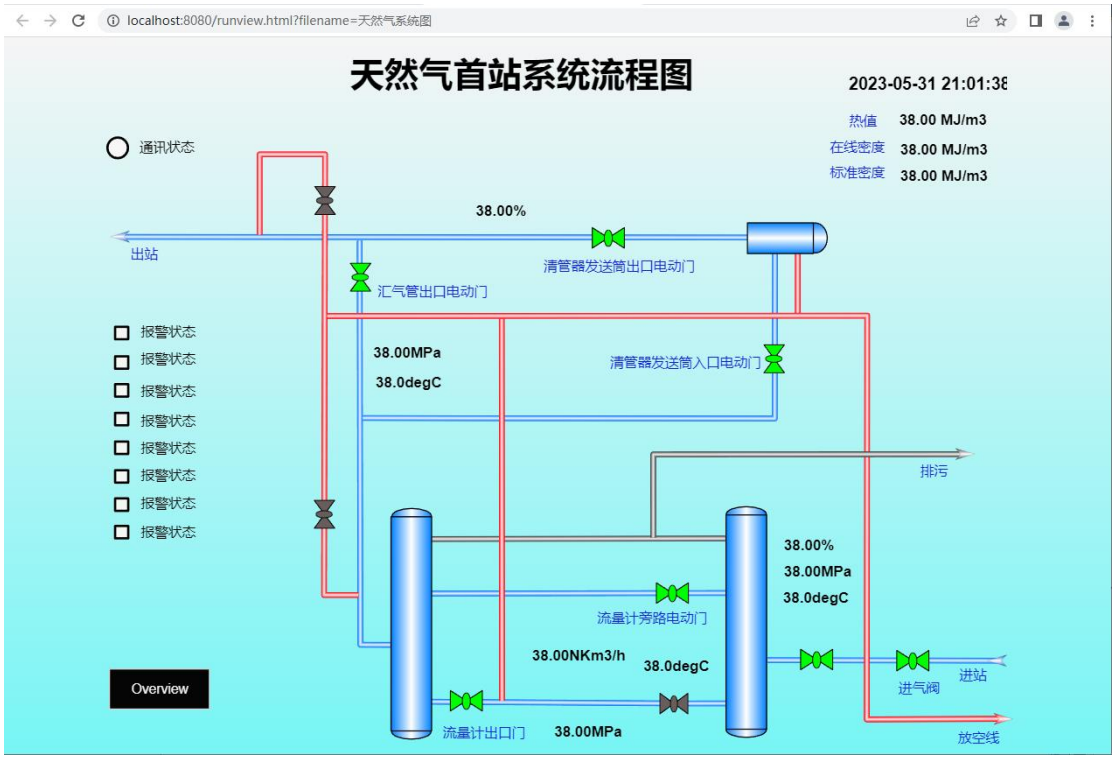
也能满足要求，因为写入指令一般需要 5ms 或者更长时间，plc 的扫描周期一般只要 1ms，因此 2 条指令写入 plc 已不在同一个扫描周期。

3.1.5 用户图形的变量绑定

用户图形绑定变量时如果是单图形控件，执行显示和隐藏操作，双图形控件则执行切换图形操作。

3.2 运行环境

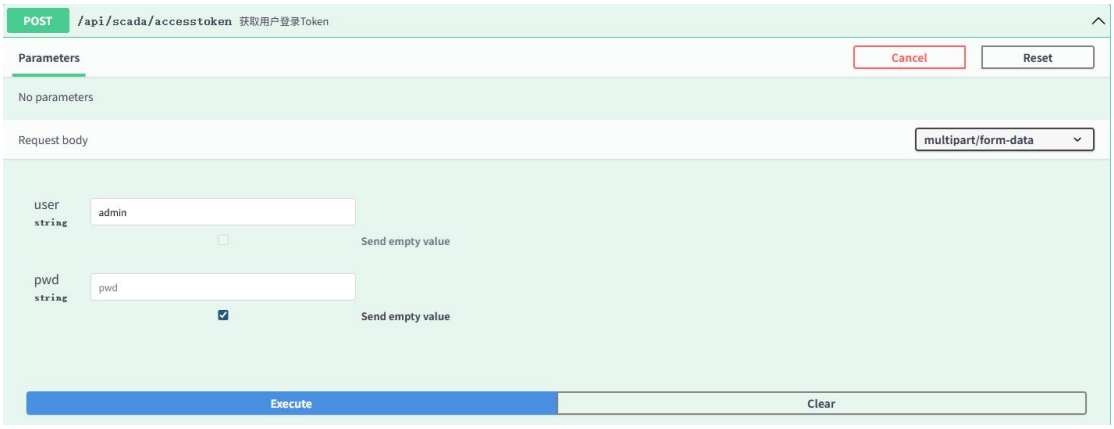
1）HTML5 组态画面 URL 路径：runview.html



（1）runview.html 支持携带 token 访问自动登录和 IP 范围的白名单自动登录例如 runview.html?filename=pic1&token=xxx-xx-xx-xxx，token 方式下一但完成登录，后续不再检查 token 参数。为了安全考虑，管理员不支持 token 方式登录（使用权限值 10 代替，取消管理员标志）。但是这样的方式 token 由于是固定的，因此很容易泄漏出去，安全性不高。

（2）从 V2.6.11 版本开始支持通过 /api/scada/accesstoken （post form 方式传递 user 和 pwd）获取 token，调用 tokenview.html?accesstoken=xxx 进行

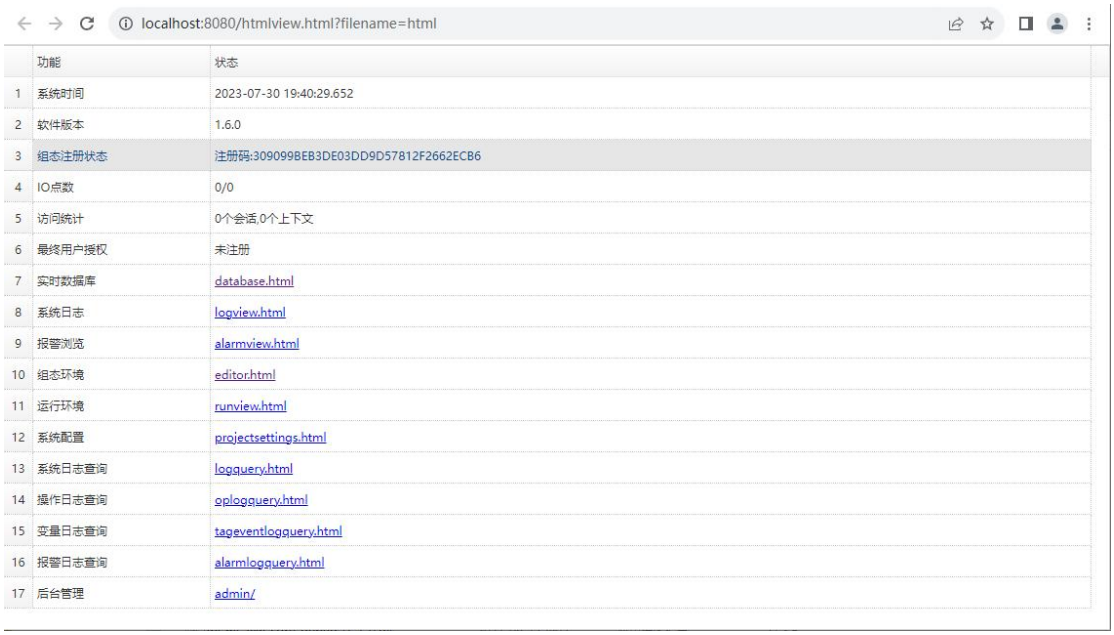
登录和页面跳转，该 token 有效期为 1 小时，登录后即刻失效。



（3）支持 runview.html?lang=en 强制使用英文，默认情况根据浏览器的语音设置自动适配中文和英文。

（4）登录时，用户密码连续输错 3 次，账户会锁定 5 分钟。

2）HTML 组态画面 URL 路径：htmlview.html



HTML 画面运行时 H5 组态不会显示，仅使用 Body 和 Head 设置。支持 token 自动登录和 IP 白名称。

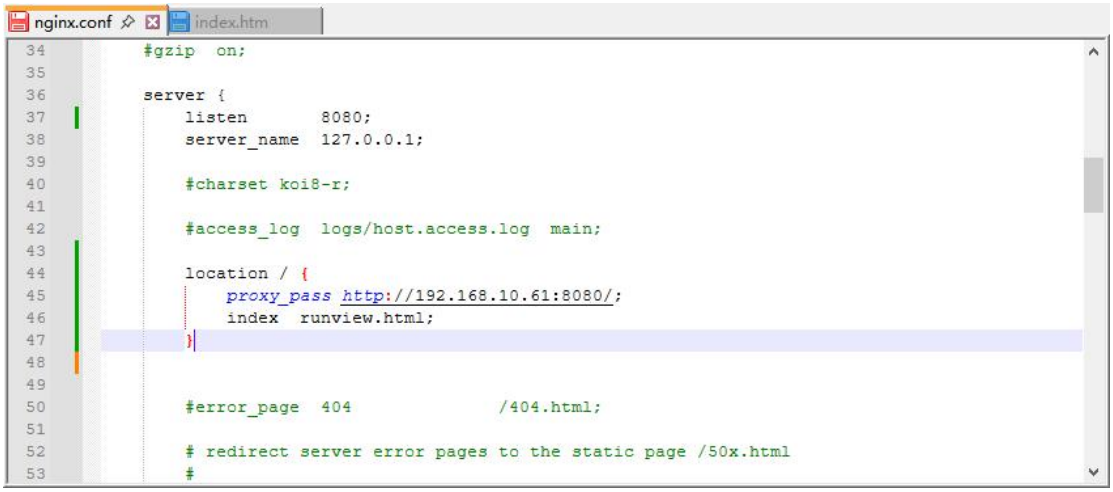
3）跨域访问

当使用 iframe 跨域嵌入组态运行环境时，如果 2 个站点的 IP 地址不一样，url 添加 token 和使用白名单都无法工作，原因是浏览器不允许跨域保存 cookie，这时就需要通过代理方式把 2 个站点统一 IP 地址，通常使用 nginx。

（1）使用 nginx 把多个站点配置为子域

上面的配置把 iotgateway 通过 /html5/路径进行了映射，与主站点同域，此时 runview.html 就可以支持 token 登录和 IP 白名单登录。

（2）在主站点服务器安装 nginx 把 WebScada 站点改为同站

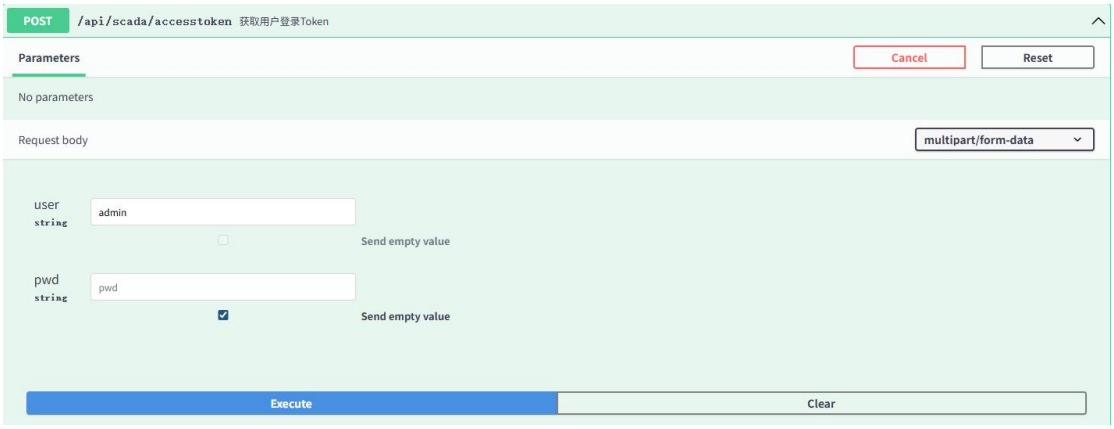


这个配置把本机 8080 端口映射到了 192.168.10.61（WebScada）的 8080 端口，浏览器访问本机 IP 地址的 8080 就能访问到 WebScada。

假定主站地址是 192.168.10.59，nginx 安装在这台机器上，WebScada 安装在 192.168.10.61 上，配置后浏览器输入 http://192.168.10.61:8080 和 http://192.168.10.59:8080 显示的页面完全一致，可以支持 iframe 嵌入访问。

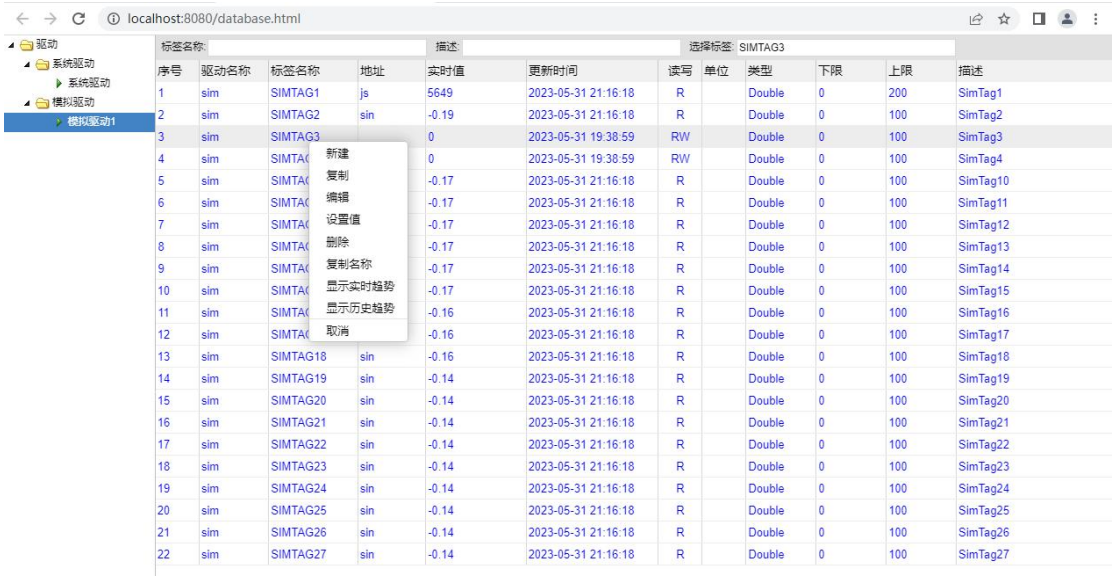
3）Token 登录

通过 /api/scada/accesstoken （post form 方式传递 user 和 pwd）获取 token，调用 tokenview.html?accesstoken=xxx 进行登录和页面跳转，该 token 有效期为 1 小时，登录后即刻失效。



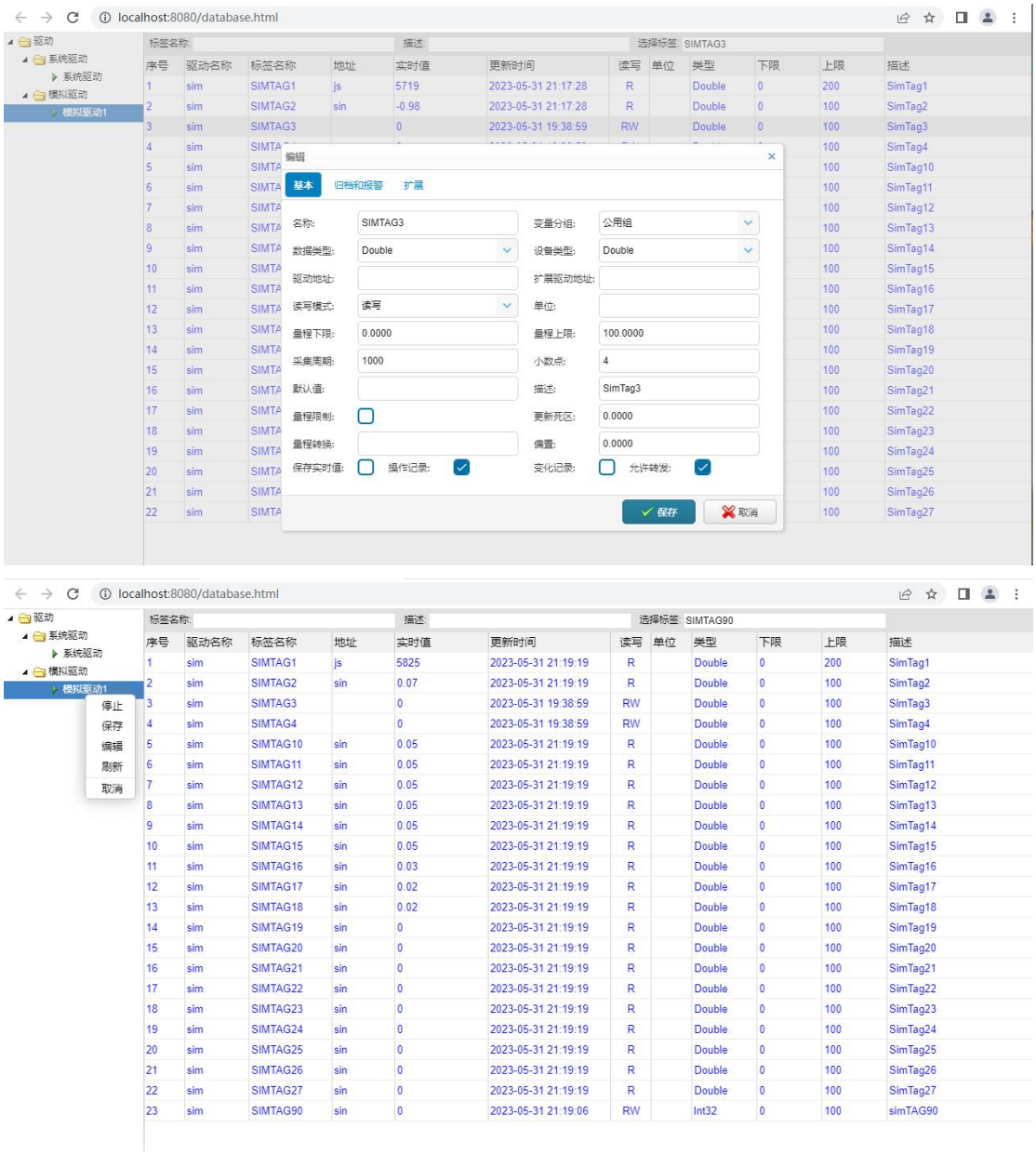
3.3 实时数据库

URL 路径：database.html

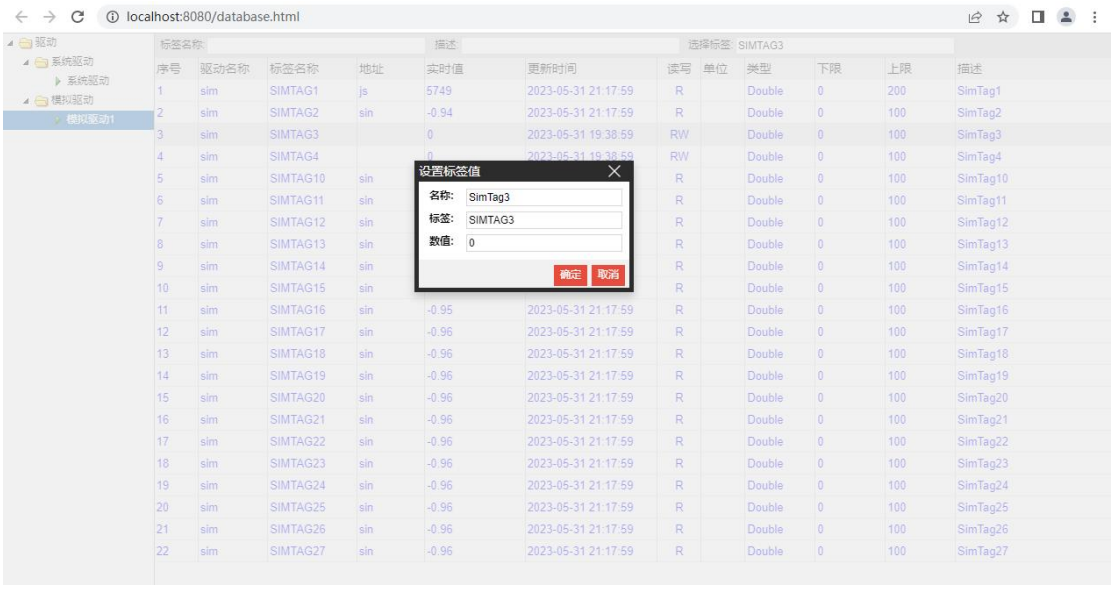


序号	驱动名称	标签名称	地址	实时值	更新时间	读写	单位	类型	下限	上限	描述
1	sim	SIMTAG1	js	5649	2023-05-31 21:16:18	R		Double	0	200	SimTag1
2	sim	SIMTAG2	sin	-0.19	2023-05-31 21:16:18	R		Double	0	100	SimTag2
3	sim	SIMTAG3		0	2023-05-31 19:38:59	RW		Double	0	100	SimTag3
4	sim	SIMTA(		0	2023-05-31 19:38:59	RW		Double	0	100	SimTag4
5	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag10
6	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag11
7	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag12
8	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag13
9	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag14
10	sim	SIMTA(		-0.17	2023-05-31 21:16:18	R		Double	0	100	SimTag15
11	sim	SIMTA(		-0.16	2023-05-31 21:16:18	R		Double	0	100	SimTag16
12	sim	SIMTA(		-0.16	2023-05-31 21:16:18	R		Double	0	100	SimTag17
13	sim	SIMTAG18	sin	-0.16	2023-05-31 21:16:18	R		Double	0	100	SimTag18
14	sim	SIMTAG19	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag19
15	sim	SIMTAG20	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag20
16	sim	SIMTAG21	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag21
17	sim	SIMTAG22	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag22
18	sim	SIMTAG23	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag23
19	sim	SIMTAG24	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag24
20	sim	SIMTAG25	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag25
21	sim	SIMTAG26	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag26
22	sim	SIMTAG27	sin	-0.14	2023-05-31 21:16:18	R		Double	0	100	SimTag27

实时数据库在管理员方式下可以对变量进行编辑，可以设置变量值，也可启动和停止驱动。



对设备或者变量进行在线修改后，需要保存才会更新到项目数据库，否则重启后会丢失修改。



3.4 系统状态

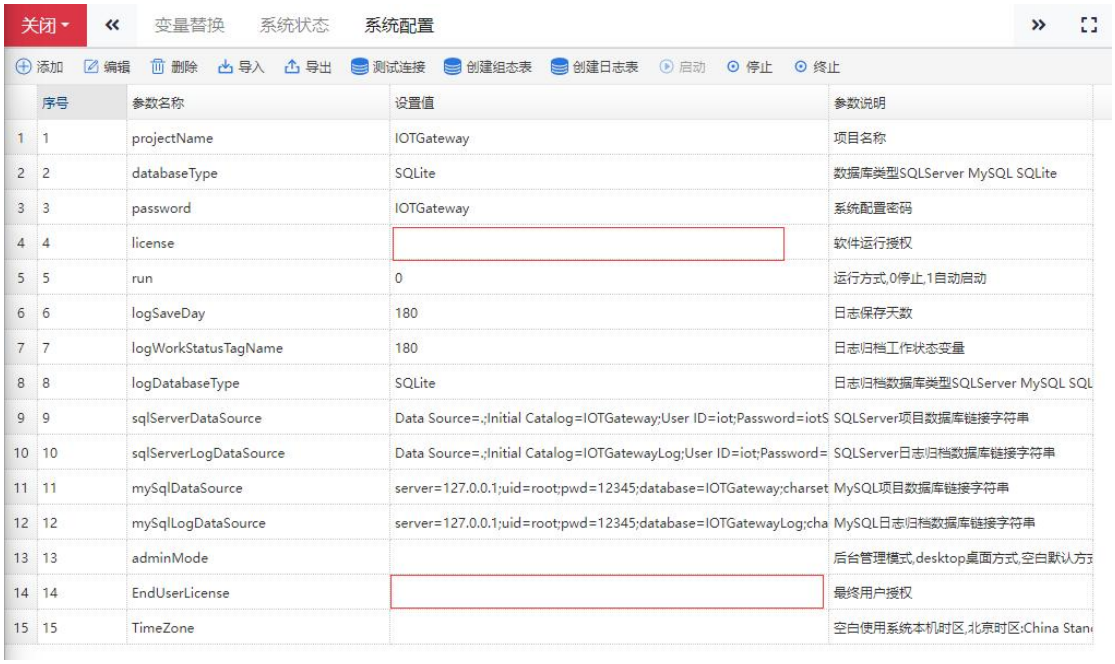
URL 路径：status.html

关闭 | 扩展配置 | 变量配置 | 画面设置 | 系统状态

	功能	状态
1	系统时间	2023-07-30 19:43:34.506
2	软件版本	1.6.0
3	组态注册状态	注册码:309099BEB3DE03DD9D57812F2662ECB6
4	IO点数	0/0
5	访问统计	0个会话,0个上下文
6	最终用户授权	未注册
7	实时数据库	database.html
8	系统日志	logview.html
9	报警浏览	alarmview.html
10	组态环境	editor.html
11	运行环境	runview.html
12	运行环境	htmlview.html
13	系统配置	projectsettings.html
14	系统日志查询	logquery.html
15	操作日志查询	oplogquery.html
16	变量日志查询	tageventlogquery.html
17	报警日志查询	alarmlogquery.html
18	后台管理	admin/

系统状态显示了一些系统参数，使用软件授权方式时提供注册码信息给供货商。

注：docker 运行时不支持软件授权。

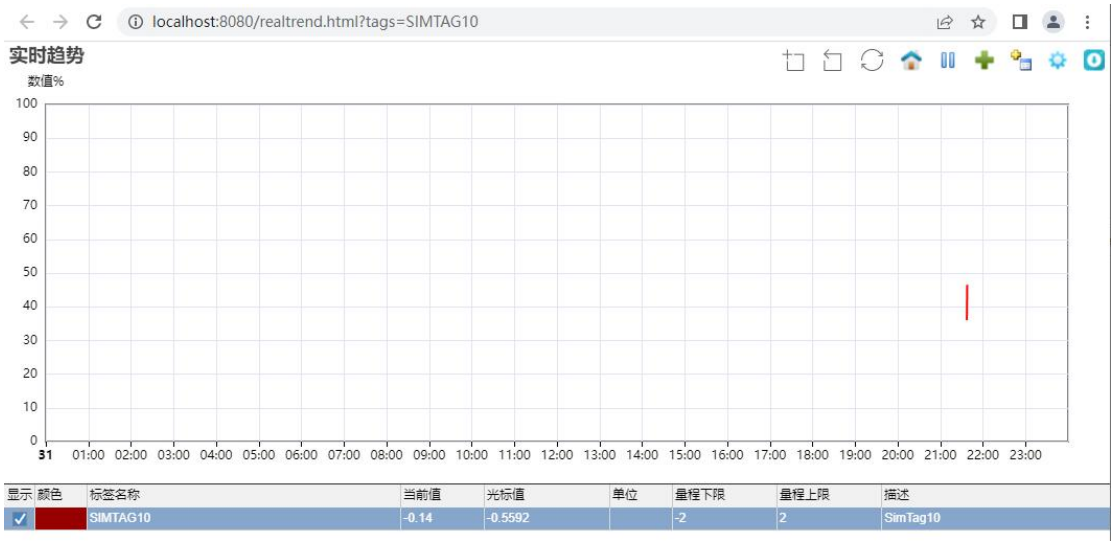


	序号	参数名称	设置值	参数说明
1	1	projectName	IOTGateway	项目名称
2	2	databaseType	SQLite	数据库类型SQLServer MySQL SQLite
3	3	password	IOTGateway	系统配置密码
4	4	license		软件运行授权
5	5	run	0	运行方式,0停止,1自动启动
6	6	logSaveDay	180	日志保存天数
7	7	logWorkStatusTagName	180	日志归档工作状态变量
8	8	logDatabaseType	SQLite	日志归档数据库类型SQLServer MySQL SQL
9	9	sqlServerDataSource	Data Source=.;Initial Catalog=IOTGateway;User ID=iot;Password=iotS	SQLServer项目数据库链接字符串
10	10	sqlServerLogDataSource	Data Source=.;Initial Catalog=IOTGatewayLog;User ID=iot;Password=	SQLServer日志归档数据库链接字符串
11	11	mySqlDataSource	server=127.0.0.1;uid=root;pwd=12345;database=IOTGateway;charset	MySQL项目数据库链接字符串
12	12	mySqlLogDataSource	server=127.0.0.1;uid=root;pwd=12345;database=IOTGatewayLog;cha	MySQL日志归档数据库链接字符串
13	13	adminMode		后台管理模式,desktop桌面方式,空白默认方
14	14	EndUserLicense		最终用户授权
15	15	TimeZone		空白使用系统本机时区,北京时区:China Stan

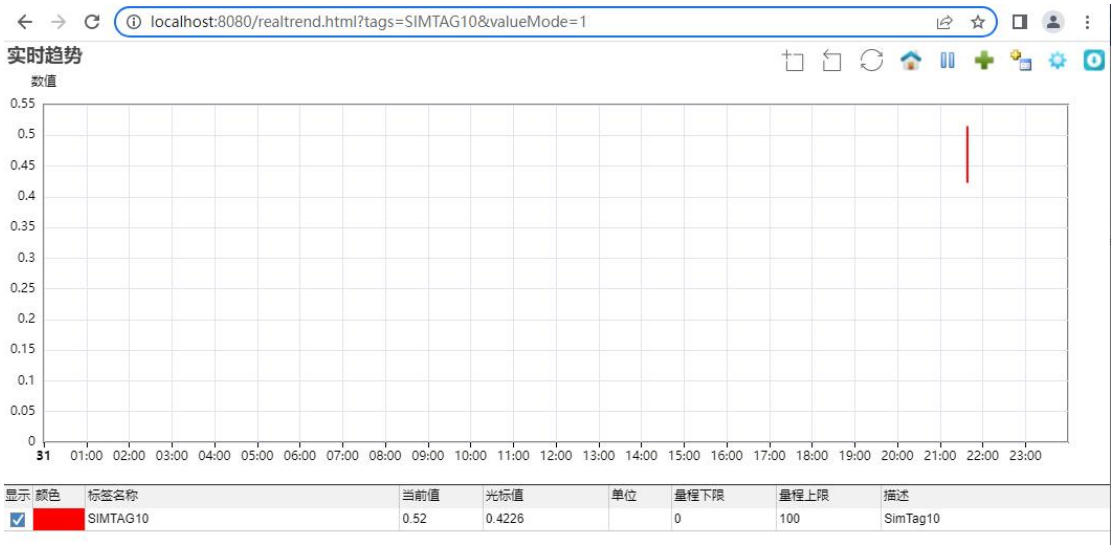
软件授权注册打开系统配置界面，在 licnese 内输入授权信息，在 EndUserLicense 输入最终用户授权信息。

3.5 实时趋势

URL 路径：realtrend.html（realtrend-en.html），通过 url 参数确定变量，也可以指定趋势组



默认 Y 量程采用百分比量程。



URL 参数中包括 valueMode 时使用自动 Y 量程

当变量存在历史配置时会自动显示一段历史数据

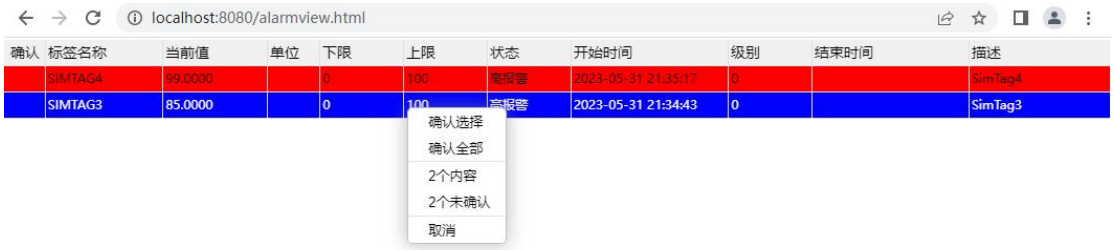
3.6 历史趋势

URL 路径：histrend.html(histrend-en.html)

变量配置了历史归档属性时历史趋势可用，url 参数和实时趋势一致。

3.7 实时报警

URL 路径：alarmview.html(alarmview.html?lang=en)



实时报警采用 WebSocket 推送，仅显示用户权限范围内的变量报警。

WebSocket 路径为 ws://ip/wsalarm

外部调用可以携带 token 进行登录，否则无法访问

指定报警分组：alarmview.html?taggroup=1

指定多个报警分组：alarmview.html?taggroup=1,3

3.8 实时日志

URL 路径：logview.html(logview-en.html)

	时间	来源	内容
1	2023-05-31 19:39:04.112	历史归档	InfluxDB连接测试失败,等待连接中...
2	2023-05-31 19:38:59.758	历史归档	InfluxDB1,线程开始运行
3	2023-05-31 19:38:59.736	列表归档	归档配置不完整,任务未启动
4	2023-05-31 19:38:59.729	行表归档	归档配置不完整,任务未启动
5	2023-05-31 19:38:59.720	模拟驱动	模拟驱动1任务启动运行
6	2023-05-31 19:38:59.719	变量服务	启动变量订阅服务
7	2023-05-31 19:38:59.678	模拟驱动	模拟驱动1开始启动运行
8	2023-05-31 19:38:59.678	模拟驱动	开始启动运行
9	2023-05-31 19:38:59.625	驱动管理器	Communication.Simulator.dll开始初始化
10	2023-05-31 19:38:59.625	驱动管理器	Communication.Simulator.dll成功加载到内存
11	2023-05-31 19:38:59.578	报警服务	报警任务运行
12	2023-05-31 19:38:59.577	报警服务	启动报警服务
13	2023-05-31 19:38:59.576	System	未检测到授权,8小时后停止运行
14	2023-05-31 19:38:59.436	System	开始启动
15	2023-05-31 19:37:50.565	System	系统激活成功

实时系统日志存储在内存中，保存最近的 1000 条记录。

3.9 SSL 加密通讯

1）使用 nginx 代理软件配置 ssl 证书的方法

```
http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
        listen       880 ssl;
        server_name  wtscada.com;

        ssl_certificate      wtscada.com_bundle.pem;
        ssl_certificate_key  wtscada.com.key;
        ssl_session_cache    shared:SSL:1m;
        ssl_session_timeout  5m;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {
            #root   html;
            proxy_pass   http://127.0.0.1:8080;
            index  index.html index.htm;
        }
```

修改证书的域名和证书文件，证书文件存放在 conf 目录下

2）使用内置 Kestrel 配置 pfx 证书

appsettings.json

```
{
  "urls": "http://*:8080;",
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "CorsPolicy": "*",
  "ProjectReadOnly": false,
  "Html5ReadOnly": false,
  "MaxTags": 0,
  "AppName": "IOTGateway",
  "Kestrel": {
    "Endpoints": {
      "Https": {
        "Url": "https://*:443",
        "Certificate": {
          "Path": "./iotgateway.pfx",
          "Password": "IOTGateway"
        }
      }
    }
  }
}
```

该配置覆盖 urls 配置的端口，使用 https 443 端口提供加密通讯

软件目录下的 iotgateway.pfx 是 1 个 20 年期的自签名证书，可以更换为自己的证书，也可以直接使用。

可以使用 windows 的 powershell 创建 1 个自签名证书，把下面的代码复制到 powershell 中

```
# setup certificate properties including the commonName (DNSName) property for Chrome 58+
$certificate = New-SelfSignedCertificate `
    -Subject "IOTGateway" `
    -DnsName "localhost"  `
    -KeyAlgorithm RSA `
    -KeyLength 2048 `
    -NotBefore (Get-Date) `
    -NotAfter (Get-Date).AddYears(20) `
    -CertStoreLocation "cert:CurrentUser\My" `
    -FriendlyName "IOTGateway" `
    -HashAlgorithm SHA256 `
    -KeyUsage DigitalSignature, KeyEncipherment, DataEncipherment `
    -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.1")
$certificatePath = 'Cert:\CurrentUser\My\' + ($certificate.ThumbPrint)
```

```
# create temporary certificate path
$tmpPath = "C:\tmp"
If(!(test-path $tmpPath))
{
New-Item -ItemType Directory -Force -Path $tmpPath
}
# set certificate password here
$pfxPassword = ConvertTo-SecureString -String "IOTGateway" -Force -AsPlainText
$pfxFilePath = "c:\tmp\iotgateway.pfx"
$cerFilePath = "c:\tmp\iotgateway.cer"
# create pfx certificate
Export-PfxCertificate -Cert $certificatePath -FilePath $pfxFilePath -Password $pfxPassword
Export-Certificate -Cert $certificatePath -FilePath $cerFilePath
# import the pfx certificate
#Import-PfxCertificate -FilePath $pfxFilePath Cert:\LocalMachine\My -Password $pfxPassword
-Exportable
# trust the certificate by importing the pfx certificate into your trusted root
#Import-Certificate -FilePath $cerFilePath -CertStoreLocation Cert:\CurrentUser\Root
# optionally delete the physical certificates (don’t delete the pfx file as you need to copy this to
your app directory)
# Remove-Item $pfxFilePath
#Remove-Item $cerFilePath
```

3.10 WebSocket 通讯

URL：ws://ip/wstag

外部调用可以携带 token 进行登录，否则无法访问

WebSocket 接口命令 JSON 格式如下：

```
{
    cmdName: "命令",
    tags:[变量名称数值];
```

}

WebSocket 接口支持的命令如下：

序号	命令	说明
1	subscribe	启动变量订阅，tags 为变量名称数组
2	unsubscribe	取消变量订阅
3	readvalues	读取变量值 tags 为变量名数值

4.辅助配置

本章节仅在管理员权限可以进入。

4.1 用户控件

组态界面的共享控件内容，提供导入、导出和名称修改功能。

4.2 画面分组

画面分组用于画面的分组，通过设置用户的画面访问组实现画面的访问权限控制。公用组无法删除，当组被使用时也无法删除。

4.3 设备分组

设备分组用于设备的分组设置，通过设置用户的设备分组实现变量访问权限控制。公用组无法删除，当组被使用时也无法删除。

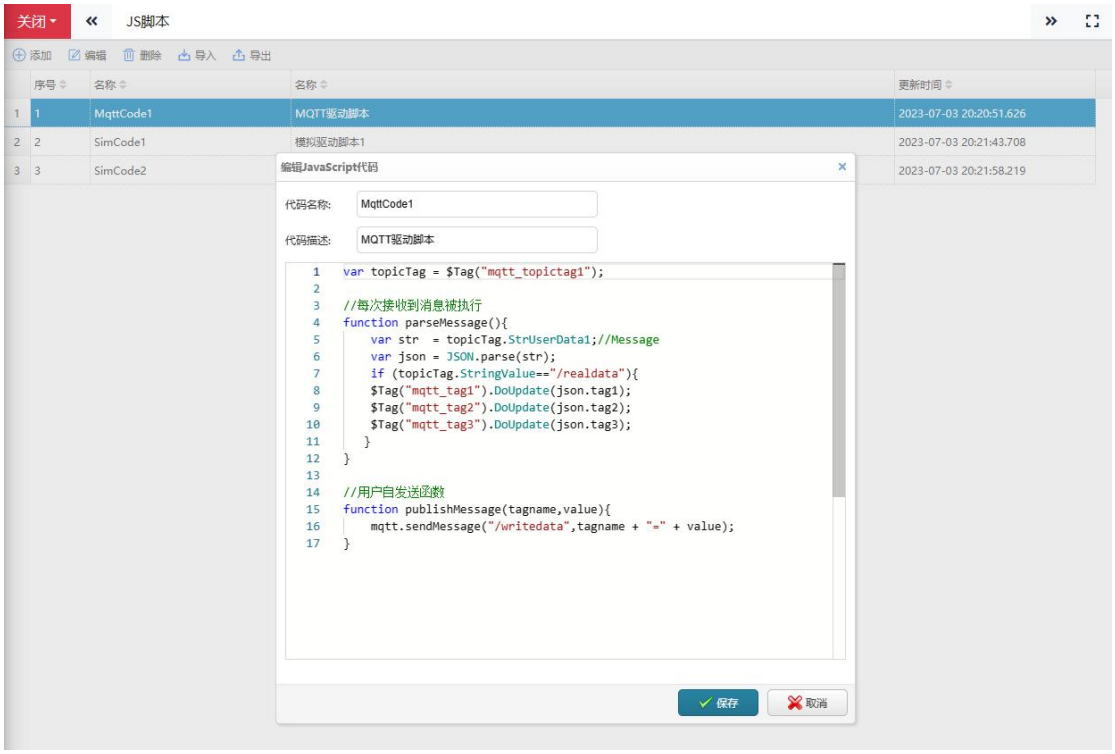
4.4 变量分组

变量分组用于变量的分组设置，通过设置用户的变量分组实现变量访问权限控制。公用组无法删除，当组被使用时也无法删除。

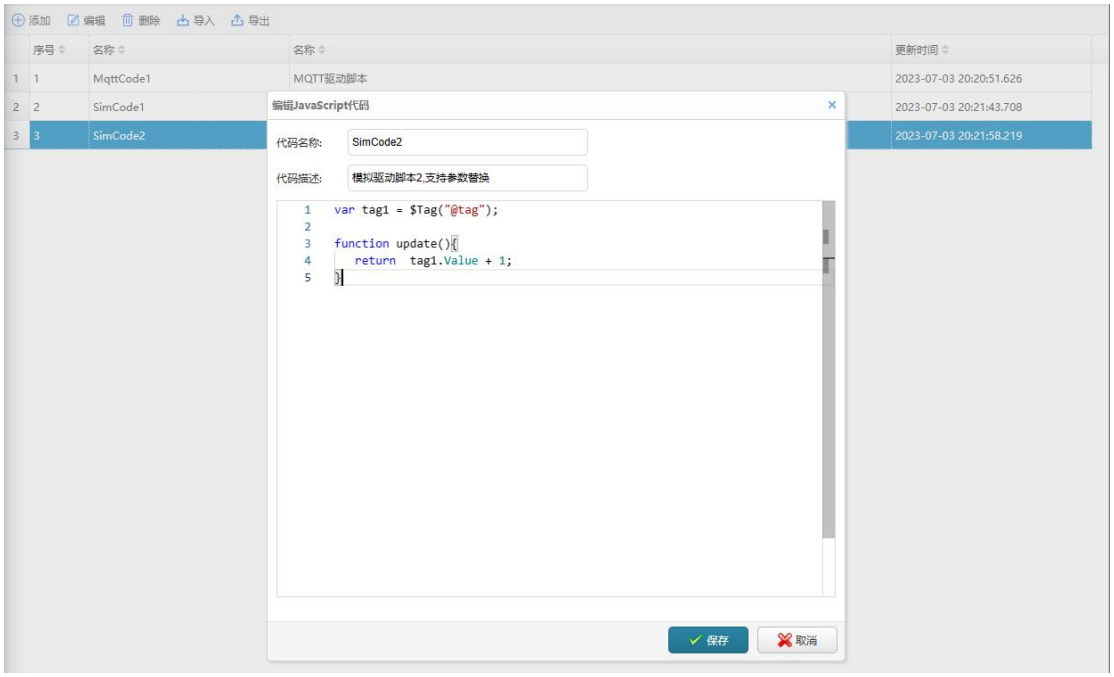
用户配置变量权限组时通过设置权限字符串 R，可以对该变量分组配置为只读。

用户配置设备权限组时通过设置权限字符串 R，可以对该设备分组配置为只读。

4.5 JS 脚本



提供统一 JavaScript 代码存储，在需要使用 JS 代码的位置输入 JS 脚本文件名称就可，从而实现了 JS 文件的复用。内置 JS 函数见附件。JS 编辑器支持代码感知技术。JS 脚本支持 URL 替换，例如：SimCode2 支持参数替换



使用时传递格式为：SimCode2?@tag=xxx,替换@tag 的为 xxx

JS 数据库访问方法如下：

1）SQL 查询 1

```
var sql = "select * from report_1";
function update(){
  var ret="";
   var json = app.SQLQuery("SqlServer","sqlserverDataSource",sql);
   //sqlserverDataSource 系统配置变量
   //var count = app.SQLExec("SqlServer","sqlserverDataSource","delete from ....");
  if (json){
      var datas = JSON.parse(json);
      for (var i=0;i<datas.length;i++){
      if (ret)
        ret += "," + datas[i].id;
      else
        ret = datas[i].id;
      }
  }
  return ret;
}
```

2）SQL 查询 2

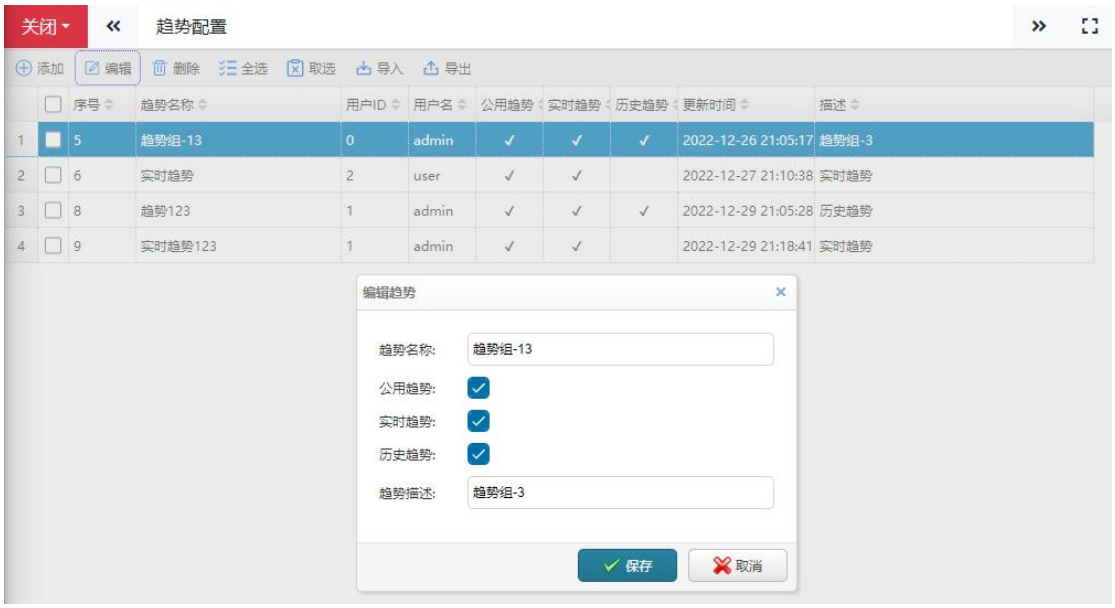
```
var strconn = "Data Source=.;Initial Catalog=Scada;User
ID=iot;Password=iotScada;Connect Timeout=30;";
var sql = "select * from report_1";
var conn = app.GetDbConnection("SqlServer",strconn);
function update(){
  var ret="";
  if (conn){
      var reader = app.QueryData(conn,sql);
      //var count = app.ExecuteNonQuery(conn,"delete from ...");
      if (reader){
        while (reader.Read()){
          if (ret)
              ret += "," + reader.GetInt32(0).toString();
          else
              ret = reader.GetInt32(0).toString();
```

```
            }
            reader.Close();
        }
        conn.Close();
    }
    return ret;
}
```

4.6 批量配置

批量配置用于表示一组变量，使用批量函数可以简单的实现对一组变量同时赋值操作。

4.7 趋势配置

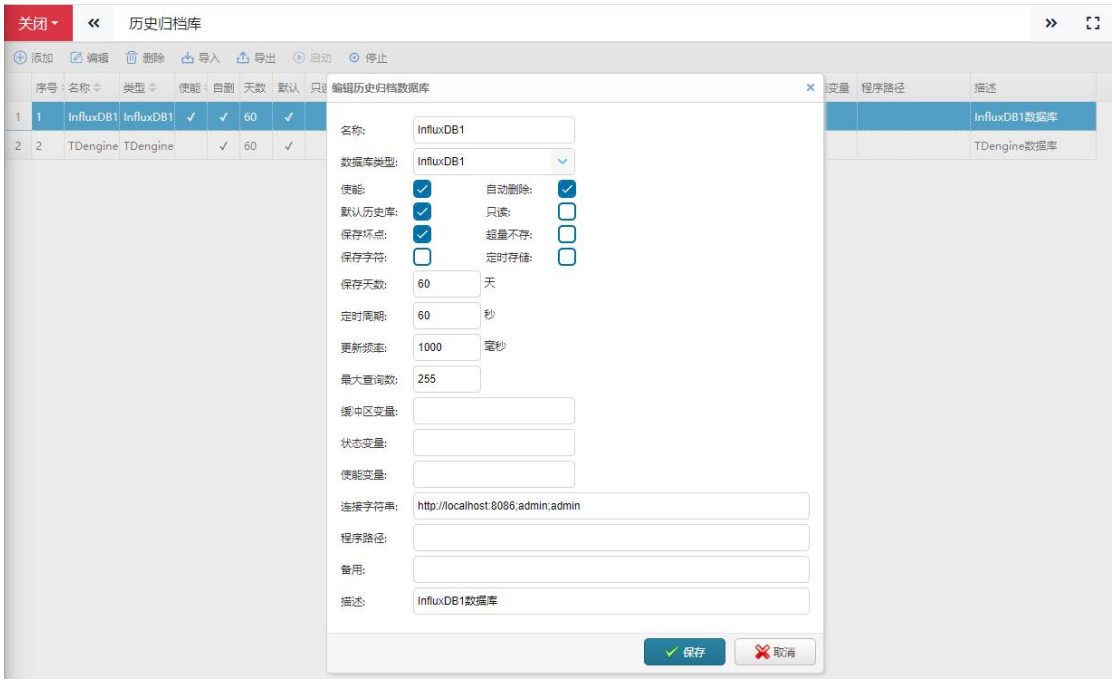


实时历史趋势管理，非公用趋势归属某个用户，公用趋势组所有用户都可以访问。趋势组可以由管理员预定义，也可以由用户在趋势界面创建和更新。

5. 归档配置

本章节仅在管理员权限可以进入。

5.1 历史归档库



变量的历史数据在配置历史归档数据库后被存储，支持配置多个历史归档数据库，默认的历史归档数据库被用于历史数据查询，当前支持 InfluxDB1（1.7~1.8.6）、InfluxDB2 和 TDEngine3，智方实时库，唐码实时库,SQLServer,MySQL（8.0 及后续版本）。

定时存储：用于 SQLServer 和 MySQL 关系数据库设置，InfluxDB1，InfluxDB2 和 TDEngine 和实时数据库不支持。

InfluxDB1 数据库连接字符格式：

（1）地址;用户;密码

http://localhost:8086;admin;admin

（2）地址

http://localhost:8086

（3）地址;用户;密码;最大归档例外时间

http://localhost:8086;admin;admin;60

InfluxDB2 数据库连接字符格式：

（1）地址;数据库;Token;组织名称

http://localhost:8086;histdata;Rud4w-bMcUsjvozt9Nc-zG2MNGGcThHwqIq3SiSgzbhDsJbylcK5JbuoQ1s6bH9QkgFixKL5cIEcF1Q4pI-hrg==;scada

（2）地址;数据库;Token;组织名称;最大归档例外时间;分表设置

http://localhost:8086;histdata;Rud4w-bMcUsjvozt9Nc-zG2MNGGcThHwqIq3SiSgzbhDsJbylcK5JbuoQ1s6bH9QkgFixKL5cIEcF1Q4pI-hrg==;scada;300;false

TDEngine3 数据库连接字符格式：地址;端口;数据库;用户名;密码

127.0.0.1;6030;iotgateway;root;taosdata

唐码实时库连接字符格式：ws://127.0.0.1:921;admin;admin

智方实时库连接字符格式：

server=192.168.10.17;port=5130;username=sa;password=zfrtdb

当变量配置历史归档后要写入到智方实时库时需要配置 1 个扩展参数指定写入到实时库的变量名称，该名称能自动创建(*其他历史库直接使用变量名写入*)。

SQLServer 数据库连接字符串：Data Source=.;Initial Catalog=iotGatewayArchiver;User ID=iot;Password=iotScada;Connect Timeout=30;

SQLServer 使用块插入模式，插入速度非常快，不写入数据库日志中，也不会执行数据库触发器。

MySQL 数据库连接字符串：

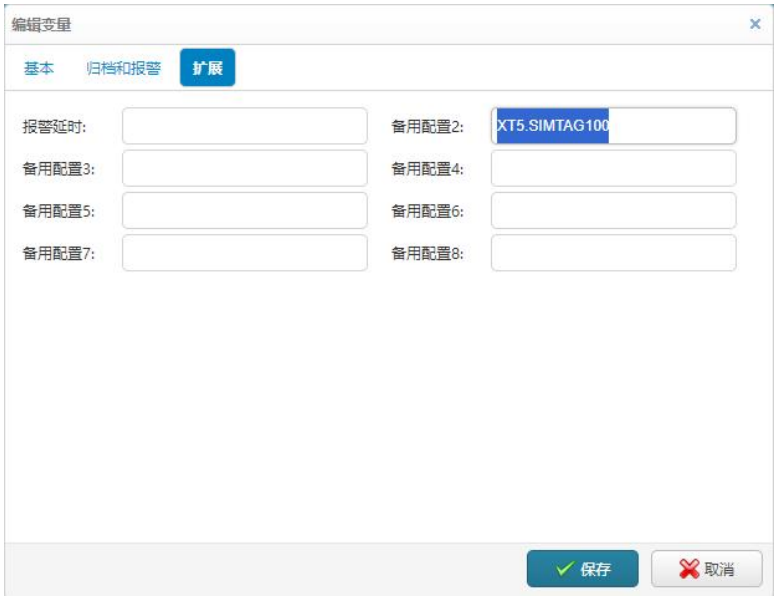
server=192.168.10.53;uid=root;pwd=MySQL@123;database=scada;Connect Timeout=30;AllowLoadLocalInfile=true;

MySQL 使用文件插入模式，因此需要数据库的支持，MySQL8 开始支持,注意必须包括 AllowLoadLocalInfile（也可以在 my.cnf 配置文件开启设置）

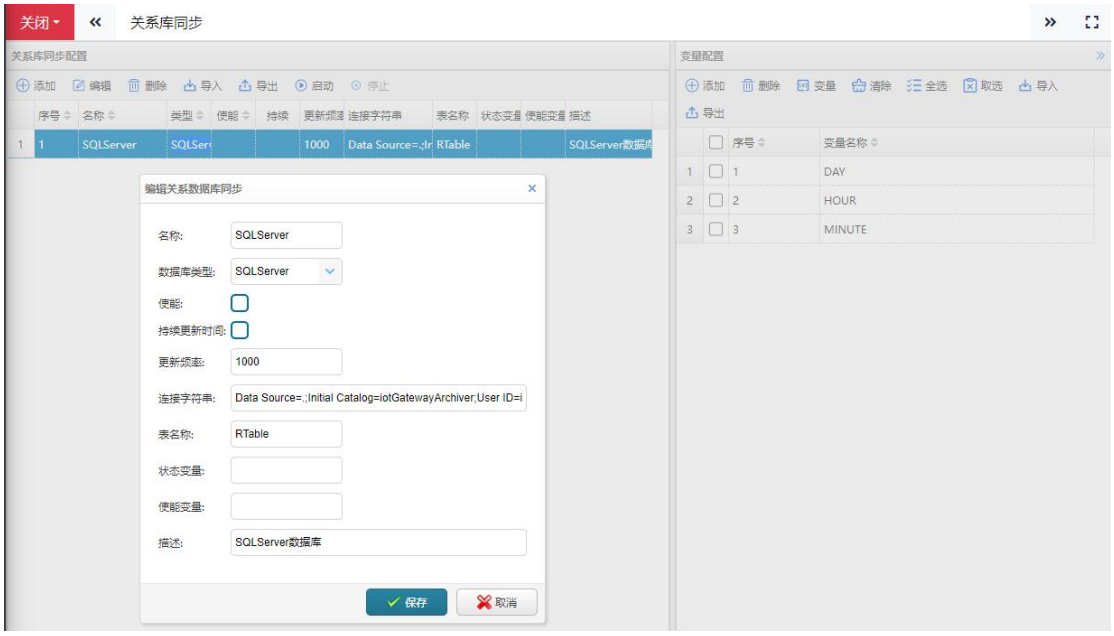
密码方式不对的情况下尝试加入下面的段：

SslMode=none;AllowPublicKeyRetrieval=True

智方实时库写入变量名称设置如下：



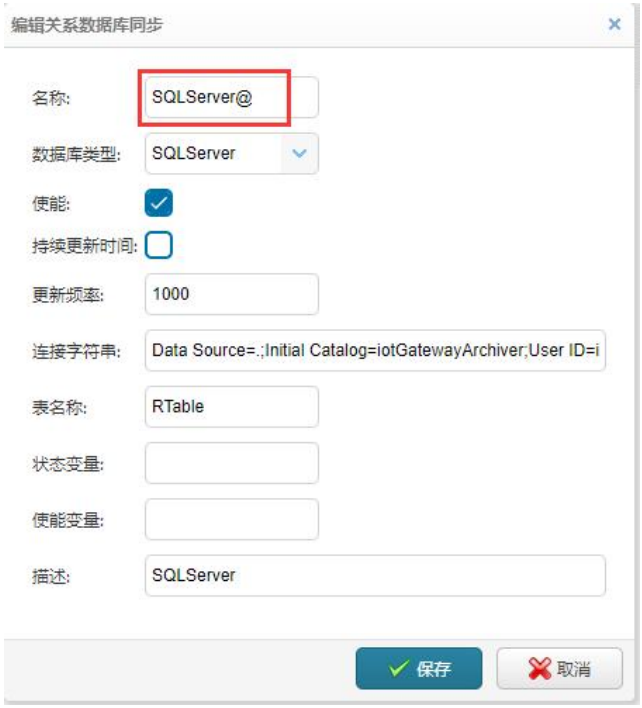
5.2 关系库同步



关系数据库同步支持 SQLServer、MySQL、PostgreSQL、达梦，openGauss，把配置的变量按设定周期更新到关系库中，数据库中的表会自动创建，支持多个数据库配置同时运行，使用 Update SQL 更新行，更新频率单位是 ms，默认采用变化更新，勾选持续更新后变量在 60 秒内至少会更新一次时间到数据库。

正常情况下关系库同步运行时先清除数据库表的全部数据，然后导入配置的变量后开始执行更新，配置多个数据库的情况每个库中的数据是一样的，特殊情况下可能需要每个数据库使用不一样的变量，此时可按下述步骤操作。

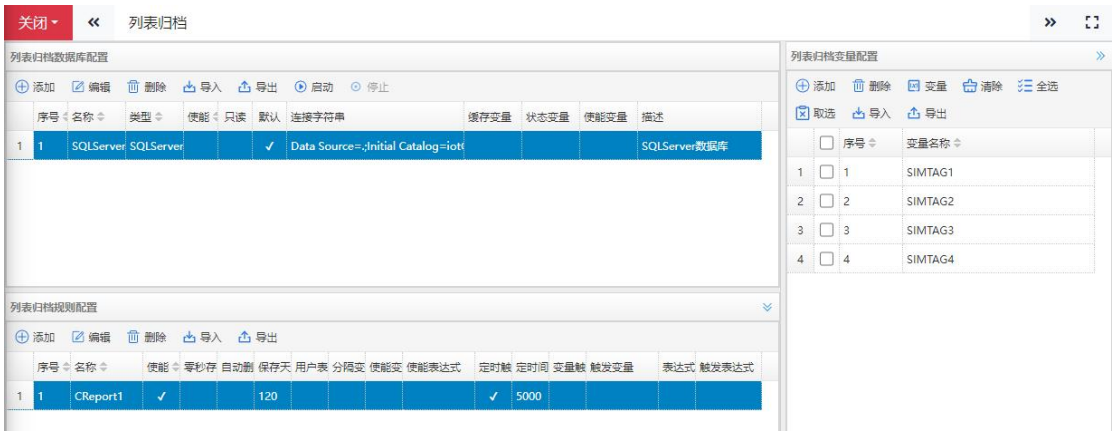
1）修改配置名称中包括@符号



2）运行归档，等待自动创建表

3）手动在表中添加需要同步的变量名称

5.3 列表归档



列表归档支持 SQLServer、MySQL、PostgreSQL、达梦，InfluxDB2，openGauss 和 SQLite 多数据库配置，列表归档的规则就是数据库的表名，每个规则支持多种触发设置，变量是表中的列，time 时间列自动产生，一行存储所有变量，每个规则配置不能超过 1000 个变量。

列表归档的线程执行周期是 100ms。

当使用 SQLite 内置数据库是连接字符串格式如下：

DataSource= /root/user/xxx.db　　指定存储目录

DataSource={project}/report.db　　软件 Data 目录下

列表归档支持定时触发执行，按定时间隔插入数据到表中。

列表归档支持变量触发执行，触发变量的值发生变量插入数据到表中。

列表归档支持布尔表达式触发执行，表达式的结果为 True 时插入数据到表中。

所有执行操作前都会判断使能变量或者使能表达式的状态，使能变量空白使能。

举例：变量触发配置 boolTag，那么 boolTag 变化时执行写入，如果使能变量也配置 boolTag，那么 booTag 变为 True 时执行写入。

例：要实现 5 分钟定时写入有 2 种设置方法

（1）变量触发

设置触发变量名称为 5MINUTEPULSE

（2）表达式触发

设置表达式为 ([Minute]%5)==0

实现 15 分钟定时写入，方法同理可以使用 15MINUTEPULSE 变量触发

实现每分钟写入一次，使用变量触发设置变量名称为 Minute

实现每小时写入一次，使用变量触发设置变量名称为 Hour

实现每天写入一次，使用变量触发设置变量名称为 Day

实现每月写入一次，使用变量触发设置变量名称为 Month

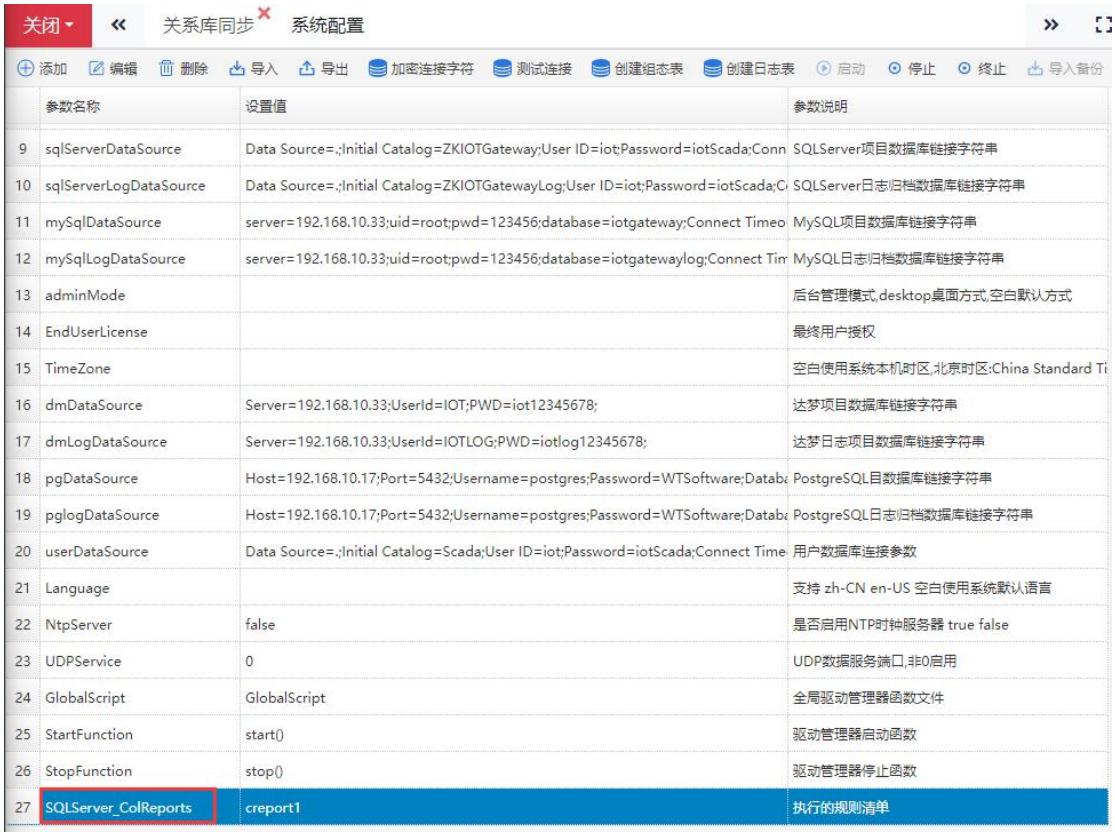
列表归档的时间支持零秒设置，支持定时删除。

当用户表名非空白时不创建表，用户表名非空白时需要用户自己创建符合格式的表格，列可以比组态实际变量多，可以添加额外的列。

当分隔变量勾选后，对于 dev1#tag1 这样的变量格式，按#分隔得到 tag1 作为变量的名称，配合用户表实现多个列表归档使用同一个表存储的目的。

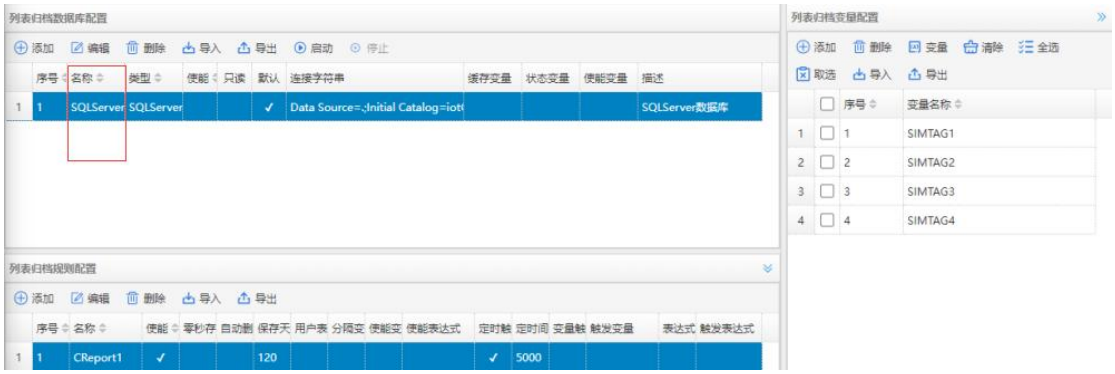
默认情况下所有的数据库存储的归档是相同的，如果要指定某个归档到某个数据库，则需要在系统配置增加自定义配置。

name_ColReports ： creport1,creport2 这样就指定了 2 个表到对应的数据库。



	参数名称	设置值	参数说明
9	sqlServerDataSource	Data Source=.;Initial Catalog=ZKIOTGateway;User ID=iot;Password=iotScada;Conn	SQLServer项目数据库链接字符串
10	sqlServerLogDataSource	Data Source=.;Initial Catalog=ZKIOTGatewayLog;User ID=iot;Password=iotScada;C	SQLServer日志归档数据库链接字符串
11	mySqlDataSource	server=192.168.10.33;uid=root;pwd=123456;database=iotgateway;Connect Timeo	MySQL项目数据库链接字符串
12	mySqlLogDataSource	server=192.168.10.33;uid=root;pwd=123456;database=iotgatewaylog;Connect Tim	MySQL日志归档数据库链接字符串
13	adminMode		后台管理模式,desktop桌面方式,空白默认方式
14	EndUserLicense		最终用户授权
15	TimeZone		空白使用系统本机时区,北京时区:China Standard Ti
16	dmDataSource	Server=192.168.10.33;UserId=IOT;PWD=iot12345678;	达梦项目数据库链接字符串
17	dmLogDataSource	Server=192.168.10.33;UserId=IOTLOG;PWD=iotlog12345678;	达梦日志项目数据库链接字符串
18	pgDataSource	Host=192.168.10.17;Port=5432;Username=postgres;Password=WTSoftware;Databa	PostgreSQL目数据库链接字符串
19	pglogDataSource	Host=192.168.10.17;Port=5432;Username=postgres;Password=WTSoftware;Databa	PostgreSQL日志归档数据库链接字符串
20	userDataSource	Data Source=.;Initial Catalog=Scada;User ID=iot;Password=iotScada;Connect Time	用户数据库连接参数
21	Language		支持 zh-CN en-US 空白使用系统默认语言
22	NtpServer	false	是否启用NTP时钟服务器 true false
23	UDPService	0	UDP数据服务端口,非0启用
24	GlobalScript	GlobalScript	全局驱动管理器函数文件
25	StartFunction	start()	驱动管理器启动函数
26	StopFunction	stop()	驱动管理器停止函数
27	SQLServer_ColReports	creport1	执行的规则清单

name:归档数据库配置名称



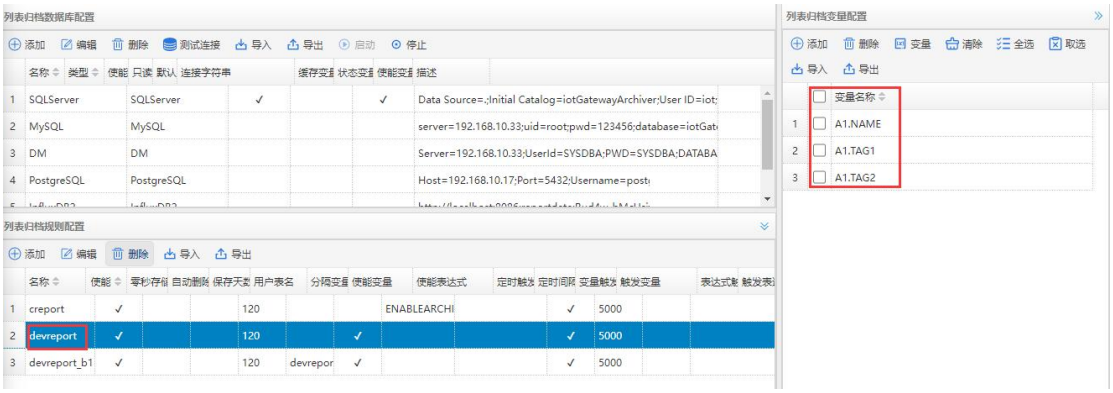
_ColReports:固定

多个设备相同变量名存储到同一个表的配置方法如下：

2	A1.TAG1	Int32	Int32
3	A1.TAG2	Int32	Int32
4	A1.NAME	String	String
5	B1.TAG1	Int32	Int32
6	B1.TAG2	Int32	Int32
7	B1.NAME	String	String

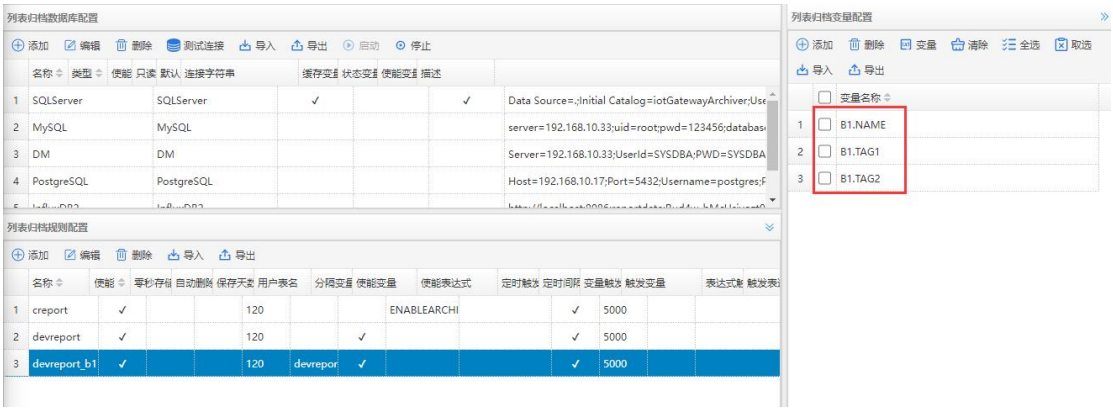
上例 2 个设备 A1 和 B1，分别有 3 个变量

配置 A1 设备的列表归档，加入 A1 设备的 3 个变量



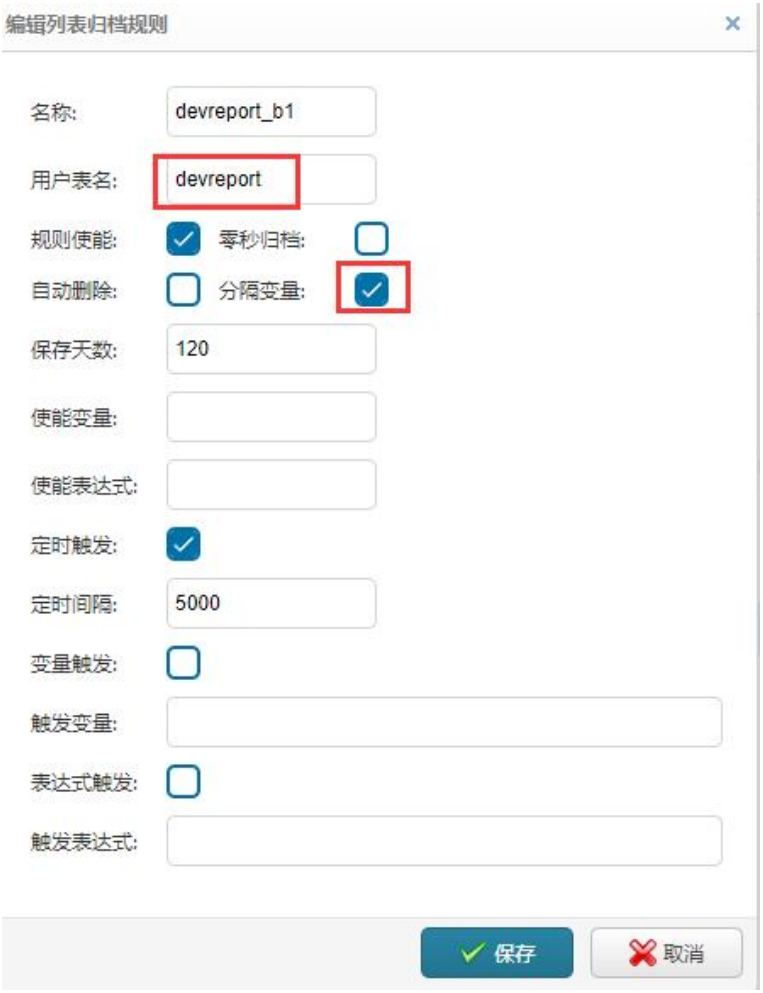
在规则中勾选分隔变量，勾选后数据库表中就使用 NAME，TAG1，TAG2 作为列名，这样 B1 设备就可以使用该表存储。

配置 B1 设备的列表归档，加入 B1 变量名称



在规则中勾选分隔变量，在用户表名中设置 A1 的表名称，这样 B1 存储时就使用 A1 的表，按此方法可以增加多个设备。

需要注意的是 B1 和后续配置使用 A1 表的变量不能多于 A1，并且变量名必须在 A1 的定义中。



5.4 行表归档

行表归档和列表归档配置类似，区别在于表格式固定，每个变量存储一行。

列表归档的线程执行周期是 100ms。

默认情况下所有的数据库存储的归档是相同的，如果要指定某个归档到某个数据库，则需要在系统配置增加自定义配置。

name_RowReports　report1,report2 这样就指定了 2 个表到对应的数据库。

5.5 自动删除数据

列表归档配置的自动删除在每天凌晨 0:15 分执行。

行表归档配置的自动删除在每天凌晨 0:20 分执行。

系统日志归档配置的自动删除在每天凌晨 0:30 分执行。

SQLServer 和 MySQL 数据库历史归档配置自动删除数据的执行时间每天凌晨 1 点进行。

这样设置的目的是防止数据库在 0 点有备份任务执行，影响操作，配置数据库定时任务时请注意。另外项目设置中可以增加 **name**_PauseTimeRange （name 是归档数据库设置的名称，例如：**MySQL**_PauseTimeRange）设置格式为 00:00:00-00:05:00，这样可以控制此时间范围内不执行行表、列表归档、SQL 历史归档的数据写入。

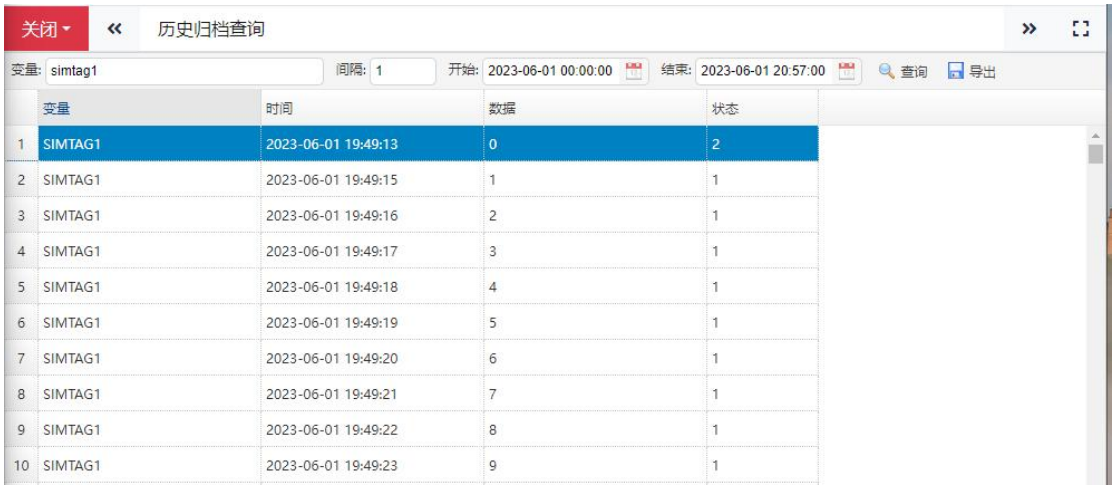
5.6 MySQL 显示毫秒数

默认情况下 MySQL 列表归档 datetime 不显示毫秒，如果需要毫秒可自行修改数据库的 datetime 列格式。

#	名称	数据类型	长度/集合	无符号的	允许 NULL	填零	默认	注释
1	id	INT	10	☑	☐	☐	AUTO_INCRE...	
2	Time	DATETIME	3	☐	☑	☐	NULL	
3	SIMTAG5	DOUBLE		☐	☑	☐	NULL	
4	SIMTAG4	DOUBLE		☐	☑	☐	NULL	
5	SIMTAG3	DOUBLE		☐	☑	☐	NULL	
6	SIMTAG2	DOUBLE		☐	☑	☐	NULL	
7	SIMTAG1	DOUBLE		☐	☑	☐	NULL	
8	TAG6	DOUBLE		☐	☑	☐	NULL	
9	TAG7	DOUBLE		☐	☑	☐	NULL	
10	TAG8	DOUBLE		☐	☑	☐	NULL	
11	TAG9	DOUBLE		☐	☑	☐	NULL	
12	Flag	SMALLINT	5	☐	☑	☐	NULL	

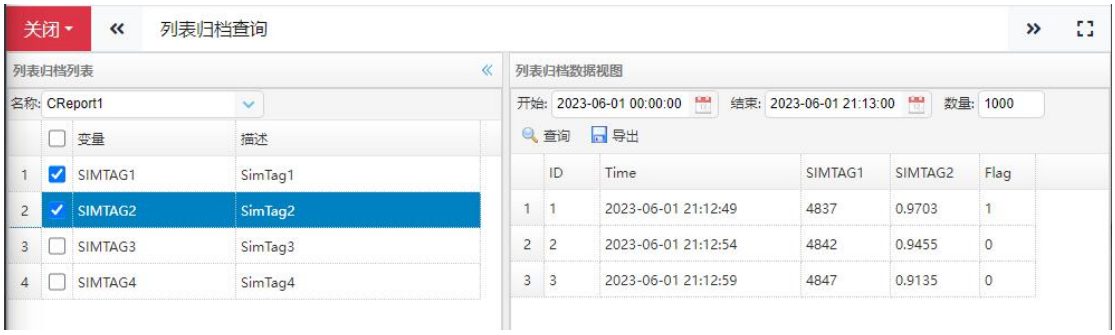
6. 归档查询

6.1 历史归档查询



历史归档支持 1 个或多个变量（逗号分隔）根据时间范围查询历史数据，时间范围较长时应该设置间隔，避免查询数据过多，间隔为 1 秒，时间范围限制为 2 小时。

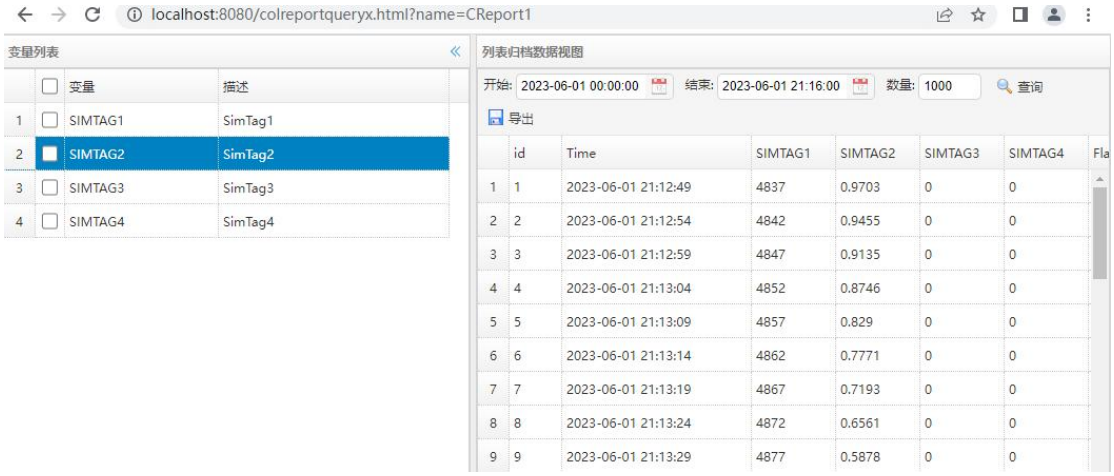
6.2 列表归档查询



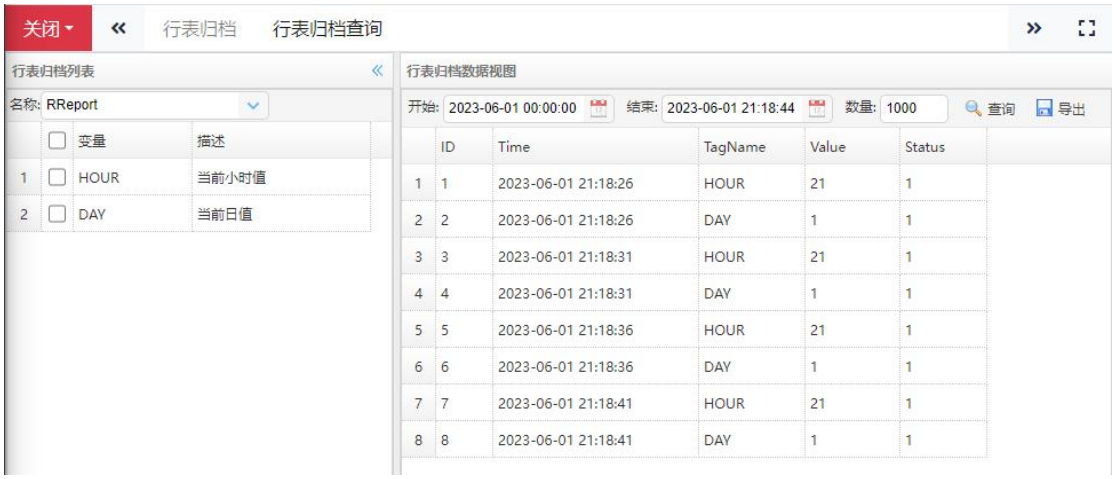
列表归档数据查询界面，支持运行界面调用，每次查询最多返回 1 万行数据。

Url 路径：colreportquery.html

可以使用 colreportqueryx.html?name=CReport1 查询某个归档的数据



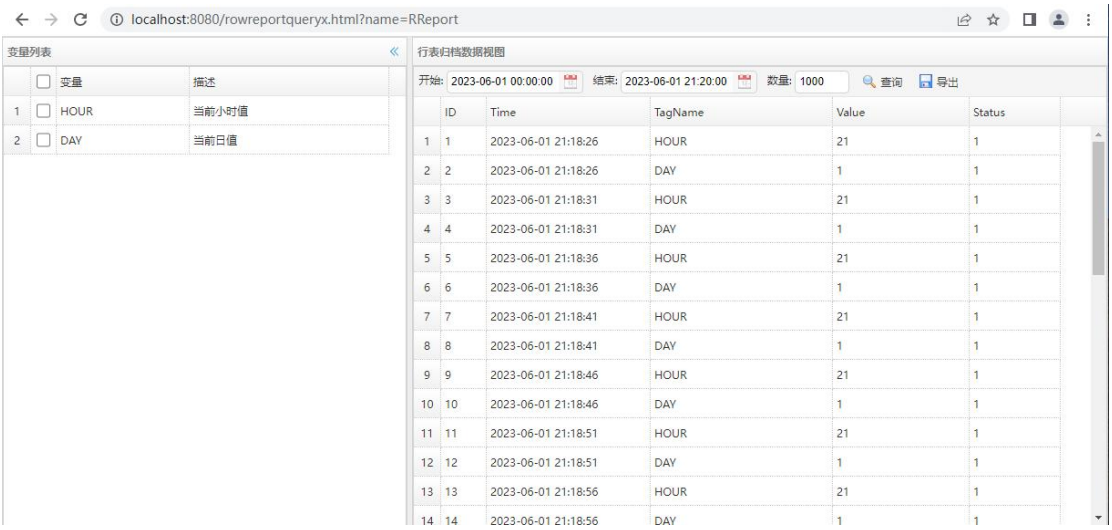
6.3 行表归档查询



行表归档数据查询界面，支持运行界面调用，每次查询最多返回 1 万行数据。

Url 路径：rowreportquery.html

可以使用 rowreportqueryx.html?name=RReport1 查询某个归档的数据



7. 日志查询

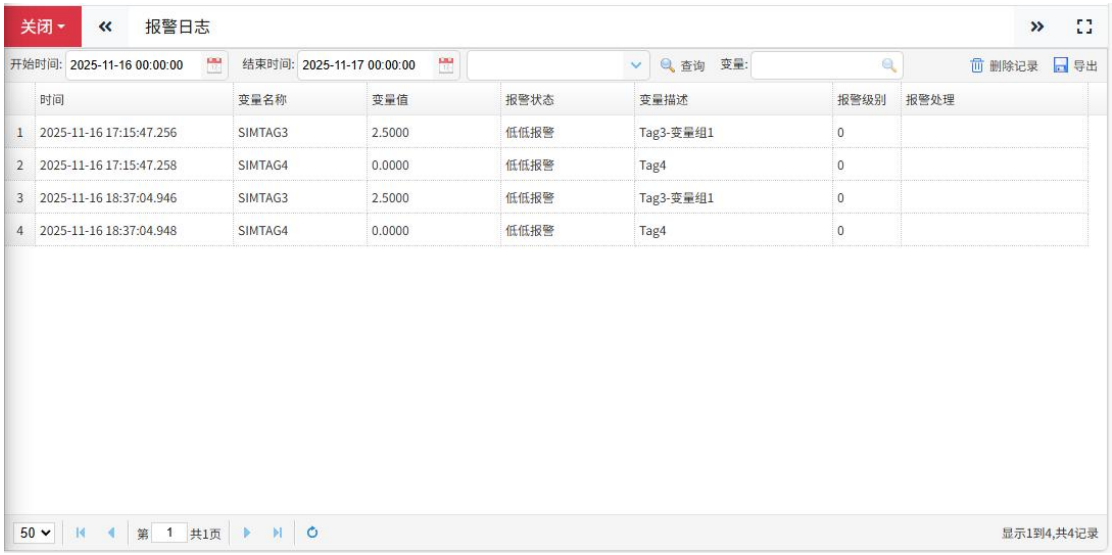
本章节需要用户权限大于 10 才可以访问，删除操作需要系统管理员权限。

7.1 系统日志查询

可以通过来源进行筛选查询，支持 url 增加 ?source=xx 传递来源

导出功能仅导出当前页面内容

7.2 报警日志查询



可以通过变量名称进行筛选查询，支持 url 增加?tagName=xx 传递变量名

可以通过变量分组进行筛选查询，支持 url 增加?tagGroupID=x 传递变量分组 ID 值

导出功能仅导出当前页面内容

7.3 变量日志查询

记录了变量值的变化日志，当变量配置勾选变化记录后才记录（切勿对大量变量勾选该记录，一般用于数字量变量记录，模拟量谨慎勾选）。

可以通过变量名称进行筛选查询，支持 url 增加 ?tagName=xx 传递变量名

导出功能仅导出当前页面内容

7.4 操作日志查询

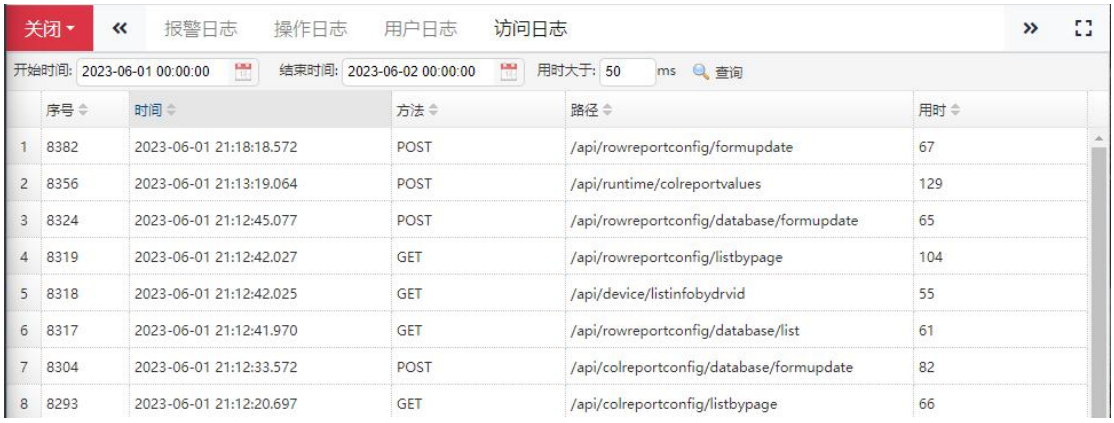
	序号	时间	变量名称	变量值	变量描述	用户名	操作结果
1	60	2023-08-19 17:02:08.712	SIMTAG3	5.5	Tag3	admin	
2	61	2023-08-19 17:02:32.912	SIMTAG4	0.15	Tag4	admin	
3	62	2023-08-19 17:02:39.856	SIMTAG5	0.16	Tag5	admin	
4	63	2023-08-19 17:06:16.507	SIMTAG3	8	Tag3	admin	
5	64	2023-08-19 17:06:38.521	SIMTAG4	88	Tag4	admin	

记录了变量设置值的日志，当变量配置勾选操作记录后才记录。

可以通过变量名和用户名称进行筛选查询，支持 url 增加 ?tagName=xx 传递变量名

导出功能仅导出当前页面内容

7.5 访问日志查询

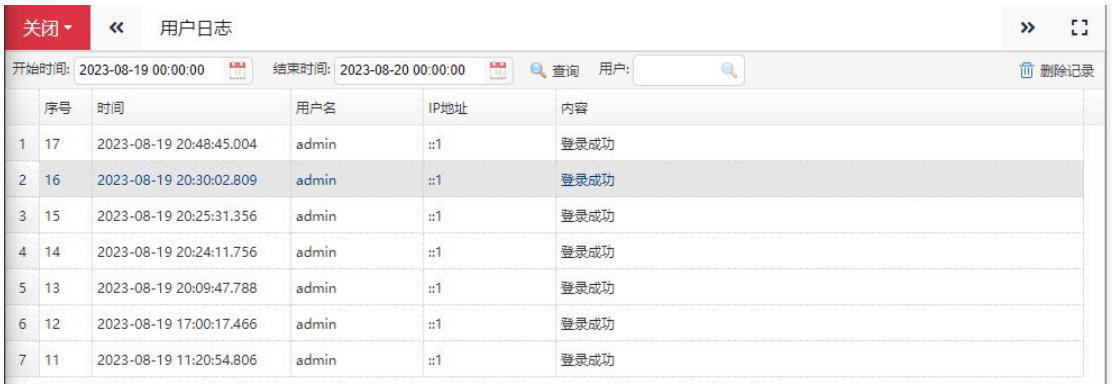


	序号	时间	方法	路径	用时
1	8382	2023-06-01 21:18:18.572	POST	/api/rowreportconfig/formupdate	67
2	8356	2023-06-01 21:13:19.064	POST	/api/runtime/colreportvalues	129
3	8324	2023-06-01 21:12:45.077	POST	/api/rowreportconfig/database/formupdate	65
4	8319	2023-06-01 21:12:42.027	GET	/api/rowreportconfig/listbypage	104
5	8318	2023-06-01 21:12:42.025	GET	/api/device/listinfobydrvid	55
6	8317	2023-06-01 21:12:41.970	GET	/api/rowreportconfig/database/list	61
7	8304	2023-06-01 21:12:33.572	POST	/api/colreportconfig/database/formupdate	82
8	8293	2023-06-01 21:12:20.697	GET	/api/colreportconfig/listbypage	66

记录了 URL 访问时间大于 10ms 的记录，用于分析访问时间过长的用途，可以通个配置关闭记录。

从 1.7 版本开始默认关闭访问日志记录，可从系统配置设置开启（修改配置后需要重启系统）。

7.6 登录日志查询



	序号	时间	用户名	IP地址	内容
1	17	2023-08-19 20:48:45.004	admin	::1	登录成功
2	16	2023-08-19 20:30:02.809	admin	::1	登录成功
3	15	2023-08-19 20:25:31.356	admin	::1	登录成功
4	14	2023-08-19 20:24:11.756	admin	::1	登录成功
5	13	2023-08-19 20:09:47.788	admin	::1	登录成功
6	12	2023-08-19 17:00:17.466	admin	::1	登录成功
7	11	2023-08-19 11:20:54.806	admin	::1	登录成功

记录了用户登录信息，登录日志包括通过正常登录的、通过 Token 登录的以及

WebSocket 登录的信息。

可以通过用户名称进行筛选查询。

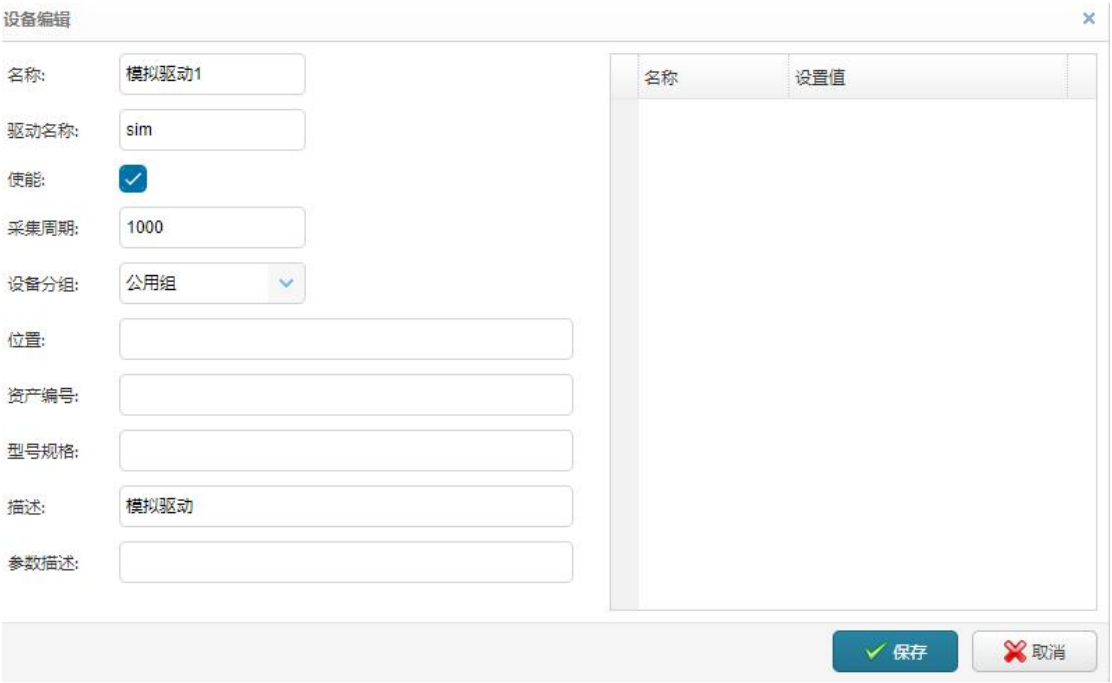
8.驱动配置

系统驱动不支持配置和修改，相关变量功能说明如下

序号	名称	描述
1	RUNTIME	软件启动后连续运行的天数
2	STARTTIME	软件启动时间
3	CPU	软件使用的总 CPU 百分比 Linux 下跟 top 命令显示的不一样，top 命令最大 CPU 用量是 N*100%，如果是 8 核心 CPU，最大会到 800%
4	MEMORY	软件使用的物理内存
5	5MINUTEPULSE	时间分钟为 5 时为 True，维持 1 分钟变为 False

驱动状态变量要配置在对应设备下才有效。

8.1 模拟驱动



模拟驱动的设备配置比较简单，没有扩展参数需要设置，勾选使能，设置名称和采集周期就可以。

扩展配置 2 非空白时 Js 和 C#变量将使用线程池异步执行，通常在脚本执行时间较长的情况下设置，由于驱动扫描时逐个变量扫描，这样可以避免脚本执行时间过长引起驱动扫描周期加长，系统具备脚本未执行完成禁止再次执行的功能。

模拟驱动的变量驱动地址和扩展驱动地址说明：

驱动地址	扩展驱动地址	说明
空白	空白	普通内存变量
eval	表达式	计算机表达式变量
rnd		随机整数变量
sin		正弦函数变量
js	JS 文件名称（支持 url 参数）如 xxx?@tag=aaa	JS 脚本变量 必须包括 update 函数
javascript	JS 文件名称（支持 url 参数）如 xxx?@tag=aaa	JS 脚本变量 必须包括 update 函数
c#	命名空间.类名.函数名称	RunTime.dll 中的静态函数 RunTime.Function.xxx

8.2 Modbus 以太网驱动

设备配置参数

名称	设置值	说明
主站类型	TCP	通讯方式，支持 TCP UDP 和 RTUOverTCP
设备 IP 地址	192.168.10.12	设备的 IP 地址

TCP 通讯端口	502	设备的通讯端口，默认 502
Ping 检测	false	是否使用 ping 检测设备正常才开始通讯采集
重试次数	2	通讯故障重试次数
错误为零	false	通讯故障时是否把变量值设置为 0
JS 脚本		JS 文件名称，每个采集周期执行的 js 脚本
字符编码	Ascii	文本的字符编码方式，支持 UTF8 Ascii Unicode GB2312 BigUnicode
字符交换字节	false	Modubs 数据是字，1 个字有 2 个字节，这个设置判断字符的顺序是否要交货位置
字节顺序	CDAB	32 位数据的字节顺序，默认 CDAB 标志模式 支持 ABCD BADC CDAB DCBA 西门子 PLC 使用 ABCD 格式
等待时间	0	每次通讯完成后是否需要等待 n 毫秒，wifi 通讯方式可能需要设置等待几十毫秒，防止出现粘包导致设备无法响应查询
0x10 功能码	true	使用 0x10 功能码写入单个寄存器，false 使用 0x06 功能码
通讯超时	1000	通讯的超时判断，单位 ms
最大读取字	120	Modbus 规定一次通讯最大读取 120 个字，某些设备不支持这么多的数据读取，根据需要调整设置
写入更新	false	设置变量值后是否立即更新变量为写入值，通常应该等待采集后更新
掩码写入	false	Bit 写入是否使用 0x22 功能码(例如:40001.0)，大多设备不支持 0x22 功能，这时使用当前值进行计算后写入到设备
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型

		0：通讯正常 10000：网络不通

Modbus 驱动变量驱动地址说明：

驱动地址	扩展驱动地址	说明
400001		400001~465000 保持寄存器读取 数据类型支持 Int16 Int32 Int64 Float Double 根据数据类型自动判断数据长度 可读写
300001		300001~365000 输入寄存器读取 数据类型支持 Int16 Int32 Int64 Float Double 根据数据类型自动判断数据长度 只读
000001		继电器读取，可读写
100001		开关量输入读取，只读
400001.x		保持寄存器 bit 读取(0~15)，可读写
300001.x		输入寄存器 bit 读取(0~15)，只读

说明：4001 和 40001、400001 意义相同，4 表示保存寄存器，后面的数字是地址偏移。如果启用了 Ping 操作，设备 Ping 成功才开始通讯采集，ping 失败时不采集。

驱动运行方式说明：

（1）驱动地址存储 Modbus 地址参数，如 40001，2:40001(站号 2），根据设备数据类型设置确定数据的字节长度。

（2）驱动首先对变量按站号和 Modbus 地址顺序进行排序，根据通讯最大字数设置得到采集数据包，相邻或连续的地址会尽可能在 1 个数据包中采集（相邻的地址跨度小于 16【位地址】、32【字地址】会在 1 个数据包中），通讯包的采集周期按采集包中变量的最小扫描周期参数设定。

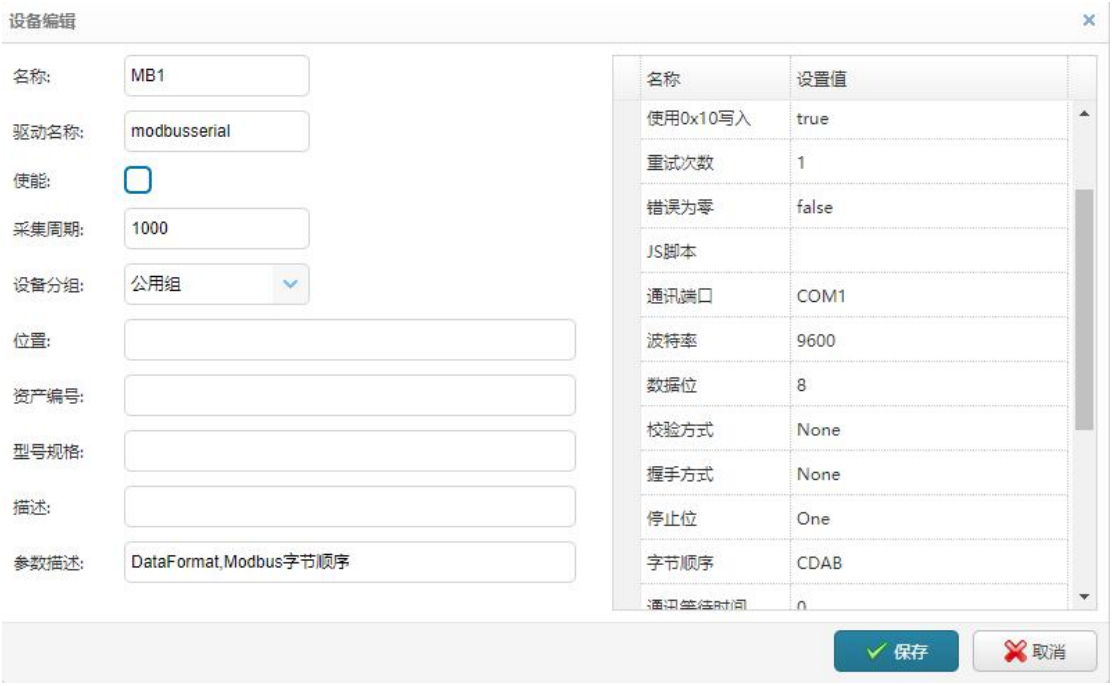
（3）1 个通讯包采集异常时多个站的情况下会暂停 10 个采集周期，单站情况暂

停 2 个采集周期。

（4）启用 Ping 后，Ping 失败不执行采集过程，Ping 失败后 2 秒 Ping 一次，成功后继续采集。

（5）可以通过通讯使能变量值暂停驱动采集。

8.3 Modbus 串行驱动



Modbus 串行通讯和 ModbusTCP 通讯类似，支持 RTU 和 ASCII 两种方式。

注意 Linux 系统串口的名称一般为 /dev/ttyS0 /dev/ttyS1

USB 串口的名称一般为 /dev/ttyUSB0 /dev/ttyUSB1

8.4 S7 驱动

S7 驱动支持西门子 S7200，1200，1500，Smart200，300，400 PLC

设备配置参数

名称	设置值	说明
TCP 通讯端口	102	S7 PLC 默认通讯端口是 102
IP 地址	192.168.10.100	PLC 的 IP 通讯地址
S7200 本地 TSAP	4D57	S7200 多站使用
S7200 目标 TSAP	4D57	S7200 多站使用
S7300TSAP	258	S7300 使用，通常默认
S7300 连接类型	1	S7300 使用，通常默认
通讯超时	1000	通讯超时时间，单位 ms
PLC 类型	S1500	PLC 类型选择,200,1200,1500,Smart200,300，400
机架号	0	用于 S7 300 和 400，其他类型设置 0
采集周期	1000	不使用
重试次数	1	通讯错误重试次数
槽位	0	用于 S7 300 和 400，其他类型设置 0
使用 Ping	true	Ping 网络检测使能
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型

通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通
DTU 模式	false	启动 DTU 通讯模式
DTU 端口	5010	DTU 服务使用的 TCP 端口
DTU 编号	0001	DTU 注册包内容（ASCII）
字符编码	Ascii	文本的字符编码方式，支持 UTF8 Ascii Unicode GB2312 BigUnicode

S7 驱动变量驱动地址说明：

驱动地址	数据类型	说明
Ix.x	Boolean	I0.0 DI 读取
Mx.x	Boolean	M0.0 M 位变量读写
Qx.x	Boolean	Q0.0 Q 位变量读写
IBx	Byte	IB0 字节方式读取 DI 1 字节
IWx	Int16	IW0 字方式读取 DI 2 字节
IDx	Int32	ID0 双字方式读取 DI 4 字节
QBx	Byte	QB0 字节方式读写 Q 1 字节
QWx	Int16	QW0 字方式读写 Q 2 字节
QDx	Int32	QD0 双字方式读写 Q 4 字节
AIx	Int16	AI0 模拟量输入 2 字节
AQx	Int16	AQ0 模拟量输出 2 字节
MBx	Byte	MB0 字节方式读写 M 地址 1 字节
MWx	Int16	MW0 字方式读写 M 地址 2 字节
MDx	Int32, Float	MD0 双节方式读写 M 地址 4 字节
Vx.y	Boolean	V0.0 S7200,Smart200
VBx	Byte	VB0 S7200,Smart200 字节方式读写 1 字节
VWx	Int16	VW0 S7200,Smart200 字方式读写 2 字节

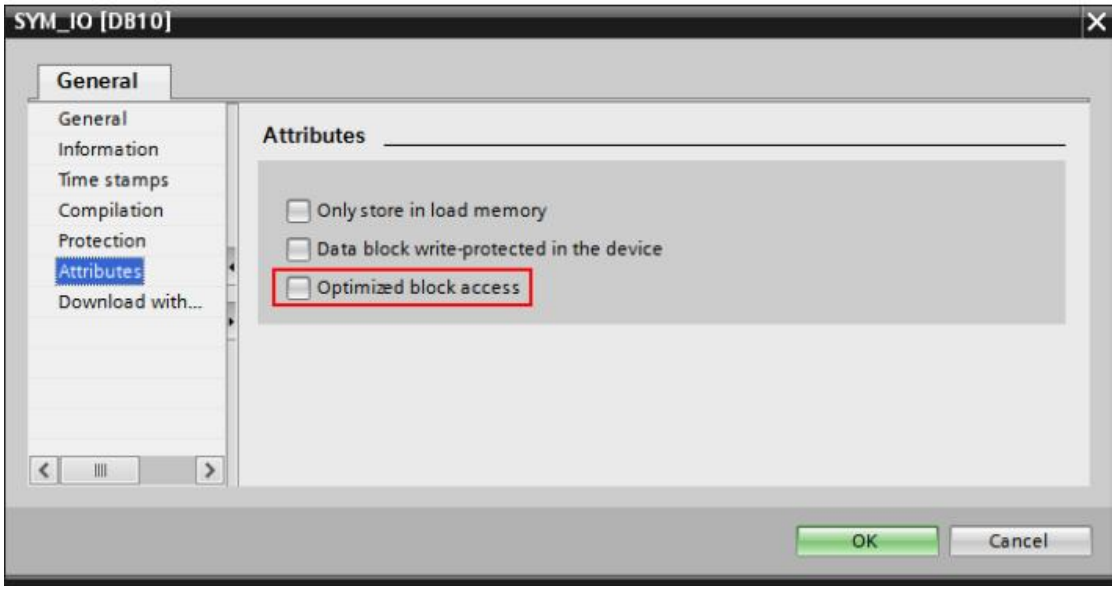
VDx	Int32，Float	VD0 S7200,Smart200 双字方式读写 4 字节
VBx.y	String	VB20.30 S7200,Smart200,长度为 30 的字符串 Smart200 使用 S7String 类型，开头是一个字节的长度数据 如果不是 S7String 需要把偏移加 1
DBXx.y.z	Boolean	DBX100.10.0 读写 DB100，偏移地址为 10，bit 0
DBBx.y	Byte	DBB100.10 读写 DB100，偏移地址 10 的字节
DBBx.y.z	String	DBB100.10.20 读写 DB100,偏移地址为 10,长度为 20 个字符串 使用 S7String 类型，开头是 2 个字节的长度数据 如果不是 S7String 需要把偏移减 2
DBWx.y	Int16	DBW100.10 读写 DB100，偏移地址为 10 的 16 位整数
DBDx.y	Int32，Float	DBD100.10 读写 DB100，偏移地址为 10 的 32 位数据
DBL.y	Int64,Double	DBD100.10 读写 DB100，偏移地址为 10 的 64 位数据

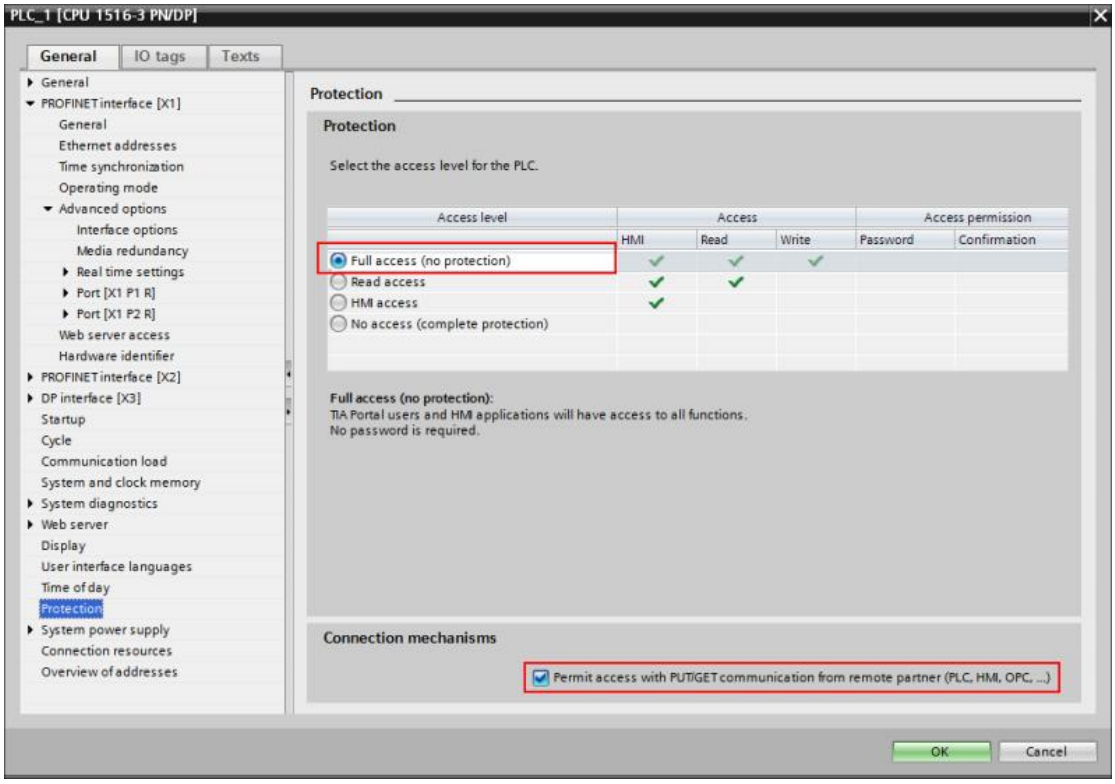
说明：如果启用了 Ping 操作，设备 Ping 成功才开始通讯采集，ping 失败时不采集。V 地址等于 DB1，DB 块的偏移不能超过 DB 块的长度，否则会影响相关连续的地址变量。

对于 S7-1200，S7-1500 需要确认：

（1）DB 块优化访问已经关闭。

（2）访问权限已经关闭，允许 PUT GET 已经开启。





8.5 OPCUA 驱动

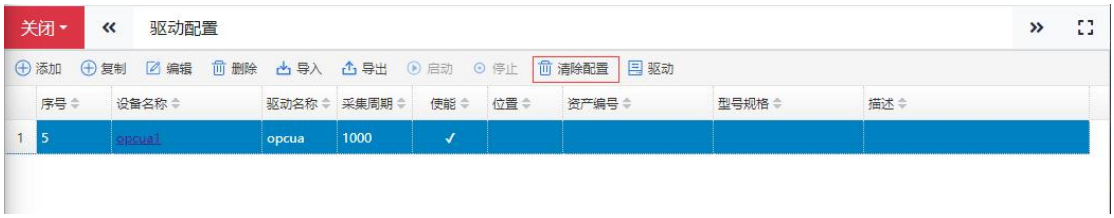
设备配置参数

名称	设置值	说明
组最大变量	5000	一个通讯组的最大变量数，通讯自动分组参数
同步读取	false	默认订阅方式，true 使用同步查询方式

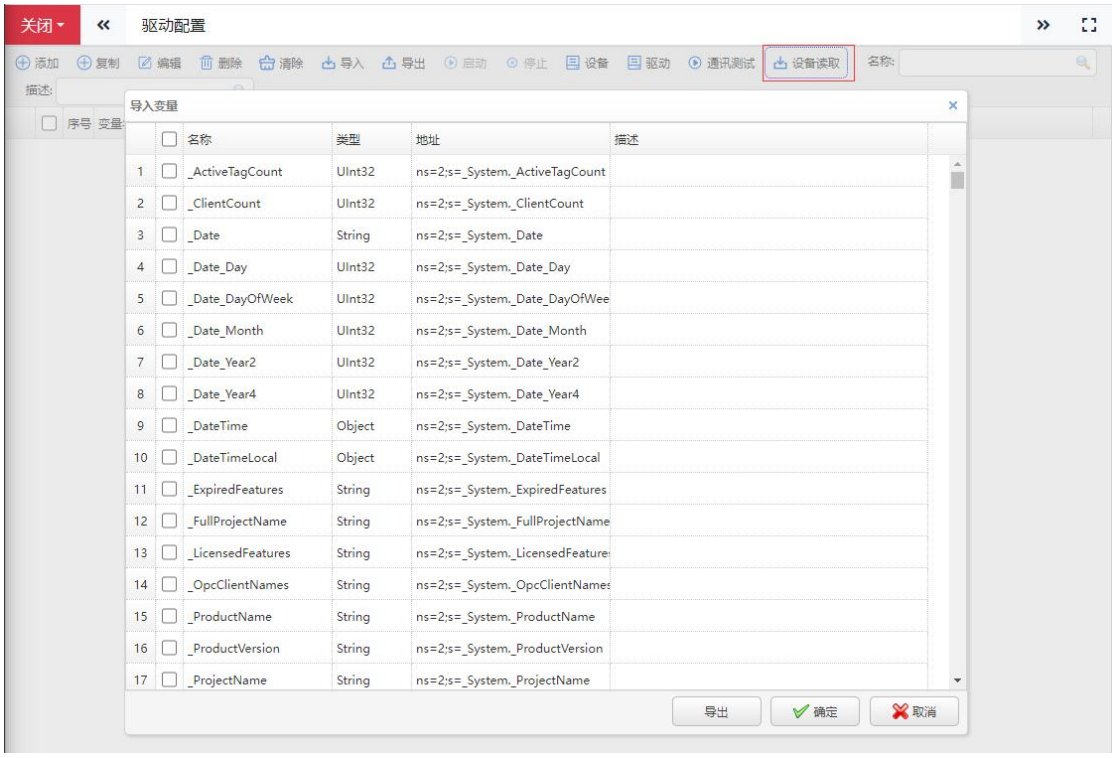
心跳时间	5	单位秒
会话超时	30	通讯会话的超时时间，单位秒
服务器时间	false	使用变量的时间更新
服务器地址	opc.tcp://192.168.10.100:49320	OPCUA TCP 连接地址
用户认证	true	用户认证模式
安全连接	false	加密连接方式
证书模式	false	证书连接模式
用户名	admin	用户认证的用户名
密码	admin	用户认证的密码
证书文件		加密通讯证书文件路径
证书密码		证书密码
采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

OPCUA 驱动变量驱动地址说明：

驱动地址	数据类型	说明
OpcUA 变量地址	设备数据类型必须和原始数据类型一致，否则无法写入成功	例：ns=2;s=模拟器示例.函数.R6



当 OPCUA 服务器发生信息改变后驱动可能无法连接，尝试清除配置。



OPCUA 驱动支持读取地址列表和通讯测试，每次最多选择 200 个变量，建议点击导出后使用 Excel 编辑后导入。

8.6 MQTT 驱动

设备配置参数

名称	设置值	说明
IP 地址	127.0.0.1	MQTT 服务器地址

TCP 端口	1883	MQTT 服务器端口
接收主题名变量		内部变量名称 String 类型
写入主题名称		内部变量 String 类型
用户名		MQTT 服务器登录用户
密码		MQTT 服务器登录密码
客户端 ID	iotgateway	客户端唯一 ID
订阅主题列表		订阅主题列表,多个主题分号或者逗号分隔
心跳时间	30	单位秒
连接超时	5000	单位 ms
Kepware 兼容格式	false	兼容 Kepware Server 格式

使用 JS 解析	false	使用 JS 脚本解析信息
		function parseMessage() {}
		从主题变量获取信息
		var topicTag = $Tag("mqtt_topictag1");
		var str = topicTag.StrUserData1;

使用 JS 发布	false	使用 JS 脚本发布信息
		function publishMessage(tagname,value){
		mqtt.sendMessage("/writedata",tagname +
		"=" + value);
		}

数据质量	1	0, 1, 2
清除会话	true	注销连接时服务器清除会话内容
保存消息	true	保存最后的消息在服务器上
采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

MQTT 驱动变量地址说明：

驱动地址	扩展驱动地址	说明
参数信息	订阅主题	通过订阅主题和驱动信息获取值 标准的数据格式是 KeyValue JSON 信息

默认支持 KeyValue JSON 数据，格式如下：

{“tag1”:1,“tag2”:false}

假如订阅主题为：/readdata

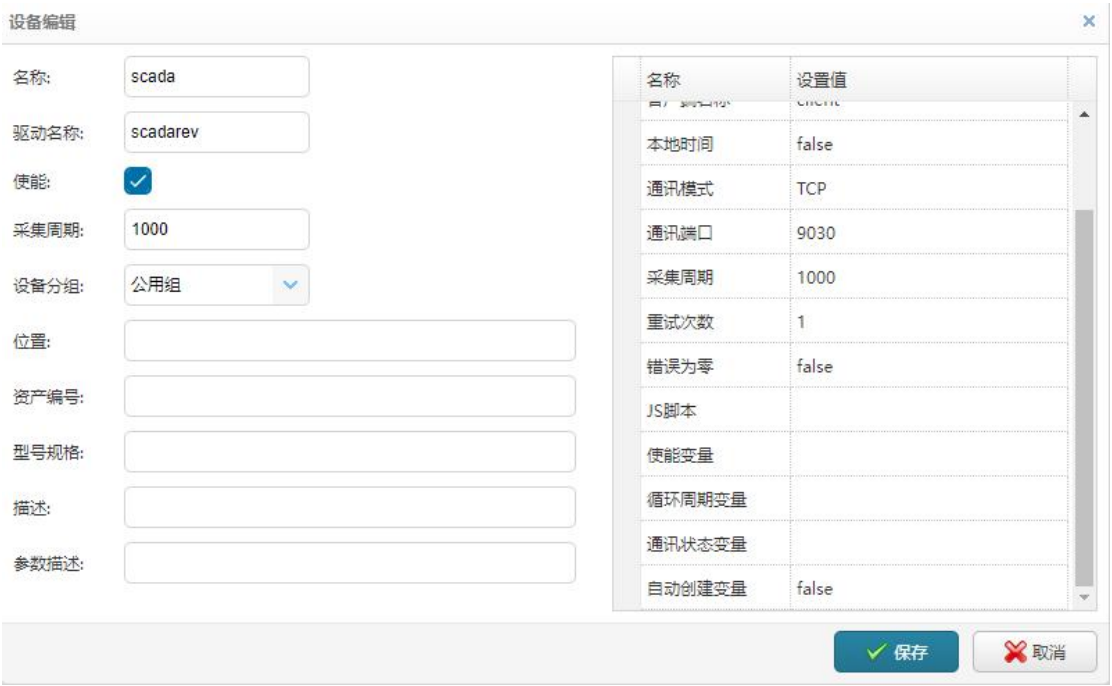
则 MQTTTag1 驱动地址信息为：tag1　扩展驱动地址为：/readdata

写入时写入的数据也是 JSON 格式：{“tag1”:value}

非标格式可以使用 JavaScript 代码自行解析，内置参考 JS 代码

8.7 网络接收驱动

网络接收驱动接收来自 IOTGatway、WTGateway、FScada 的 IO 转发数据，TCP 方式是双向可写入，UDP 方式是单向只读。



网络接收驱动的变量名称跟接收到的名称一致，驱动必须包括一个变量才能启动，当自动创建变量设置为 true 后运行后驱动会自动创建接收的变量。

“客户端名称”用于校验，不一致会连接方会出现登录失败，FScada 和 OPCLink 连接时 IOTGateway 网络接收驱动侧设置“客户端名称”为空白就可以避免。

老版本 FScada、WTGateway、OPCLink TCP 转发无法连接 IOTGateway 网络接收驱动，遇到这个情况请更新 FScada、WTGateway、OPCLink 软件版本或者改用 UDP

方式传输。

设备配置参数

名称	设置值	说明
通讯模式	TCP	TCP 或者 UDP
通讯端口	9030	TCP 或 UDP 通讯端口
自动创建变量	true	false 时不自动创建变量
客户端名称	client	认证信息，空白不判断认证
本地时间	false	使用本机时间
数据包数量		内部变量名称，整数类型
连接数变量		内部变量名称，整数类型
采集周期	1000	不使用
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

网络接收驱动变量驱动地址说明：

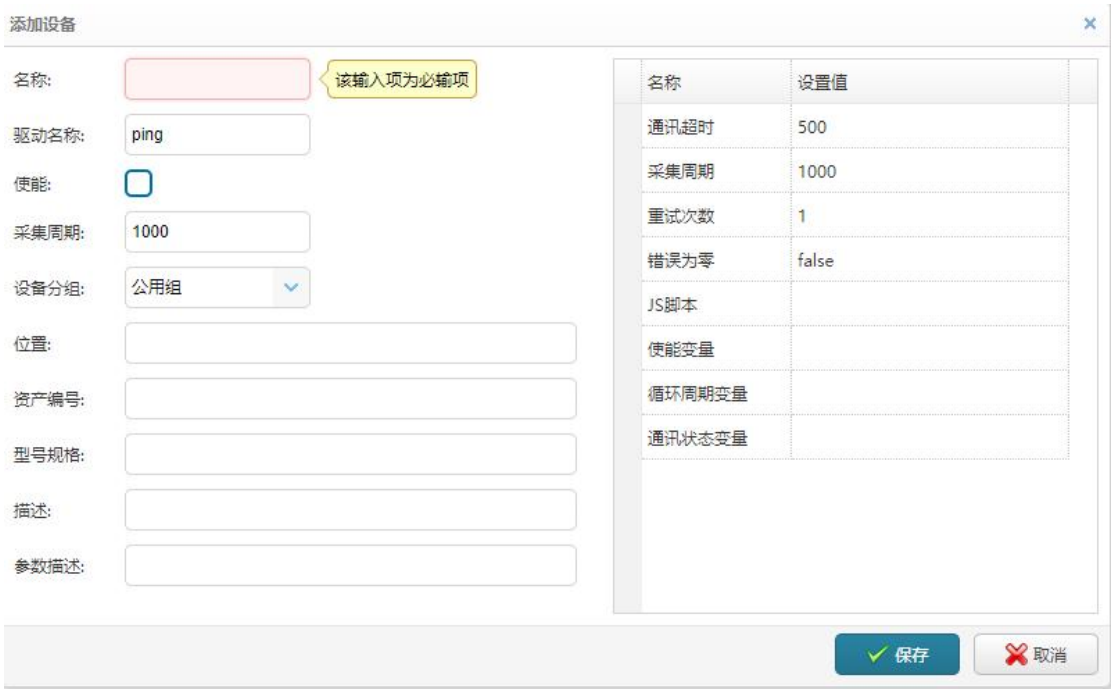
驱动地址	扩展驱动地址	说明
远程变量名		远程变量名

说明:网络接收驱动运行自动创建的变量不会立即保存到项目数据库中，需要在实时数据库界面上在设备上鼠标右键菜单中点击保存，保存后刷新一次就可在线编辑。

网络接收驱动变量创建后就不再更新变量分组和描述,变量创建时统一设置为公用组。

8.8 Ping 驱动

Ping 驱动工作原理是使用 ping 设备的 IP 地址来给变量赋值，ping 操作是异步的，大多以太网驱动均有 Ping 功能，通过设置通讯状态变量可以获取网络状态。



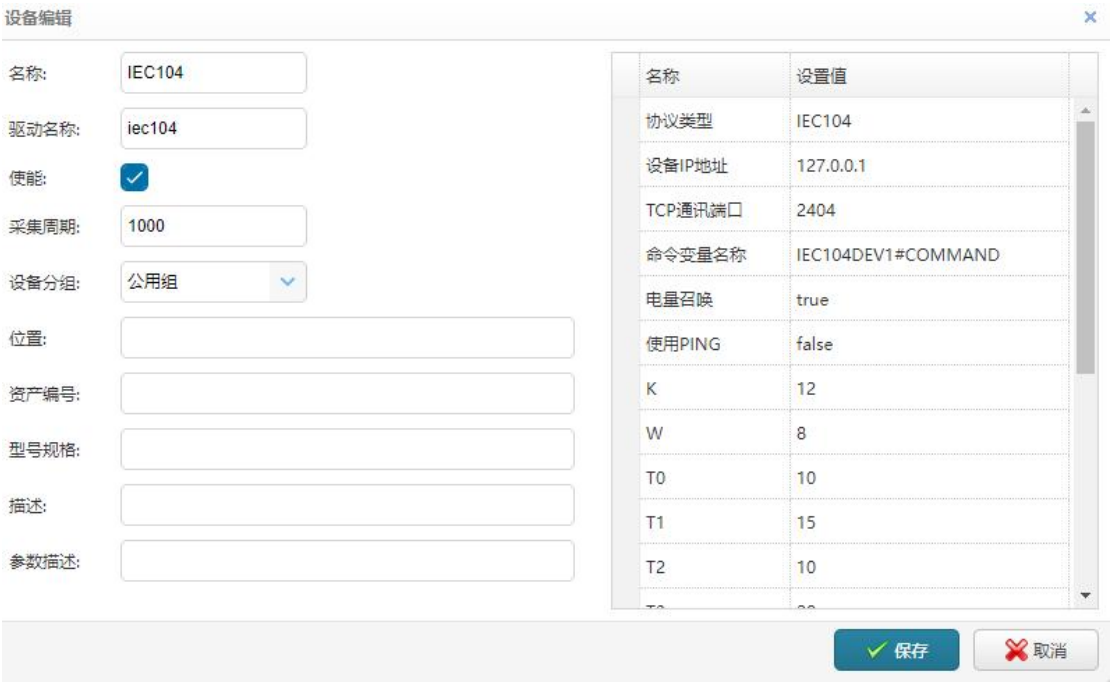
设备配置参数

名称	设置值	说明
通讯超时	500	Ping 超时
采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

网络接收驱动变量驱动地址说明：

驱动地址	扩展驱动地址	说明
IP 地址		IP4 格式的地址 192.168.1.100 当变量数据类型为 Boolean 时 Ping 成功为 True 当变量数据类型为整数时 Ping 成功为 0,断线为 10000

8.9 IEC104 驱动



设备配置参数

名称	设置值	说明
协议类型	IEC104	104TCP 方式
设备 IP 地址	127.0.0.1	设备的 IP4 地址
TCP 通讯端口	2404	TCP 通讯端口 默认 2404
命令变量名称	IEC104DEV1#COMMAND	总招命令变量 设置值为 1 执行总招，设置为 2 执行累计量总招
电量召唤	true	15 分钟一次总招，是否招呼电量
使用 PING	false	使用 Ping 设备
K	12	104 通讯参数
W	8	104 通讯参数
T0	10	104 通讯参数
T1	15	104 通讯参数
T2	10	104 通讯参数
T3	20	104 通讯参数
OA	0	104 通讯参数
CA	1	104 通讯参数

采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

IEC104 驱动变量驱动地址说明：

驱动地址	扩展驱动地址	说明
M_SP_NA_1.x		单点遥信 YX
M_SP_TA_1.x		单点遥信带时标 YX
M_DP_NA_1.x		双点遥信 YX
M_DP_TA_1.x		双点遥信带时标 YX
M_ST_NA_1.x		步位置信息
M_ME_NA_1.x		归一化值 YC
M_ME_TA_1.x		归一化值带时标 YC
M_ME_NB_1.x		标度化遥测值 YC
M_ME_TB_1.x		标度化测量值带时标 YC
M_ME_NC_1.x		短浮点遥测值 YC
M_ME_TC_1.x		短浮点遥测值 带 CP24 YC
M_IT_NA_1.x		累计量 YM
M_ME_ND_1.x		归一化遥测值 YC
M_SP_TB_1.x		单点遥信带时标 YX
M_DP_TB_1.x		双点遥信 YX
M_ME_TD_1.x		归一化遥测值带时标 YC
M_ME_TE_1.x		标度化遥测值 YC
M_ME_TF_1.x		短浮点遥测值 YC
M_IT_TB_1.x		累计量 YM
C_SC_NA_1.x		单点遥控 YK

C_DC_NA_1.x		双点遥控 YK
C_RC_NA_1.x		升降遥控 YK
C_SE_NA_1.x		归一化设定值 YT
C_SE_NB_1.x		标度化设定值 YT
C_SE_NC_1.x		短浮点设定值 YT
C_SC_TA_1.x		单点遥控 带时标 YK
C_DC_TA_1.x		双点遥控带时标 YK
C_RC_TA_1.x		双点遥控带时标 YK
C_SE_TA_1.x		归一化设定值带时标 YT
C_SE_TB_1.x		标度化设定值带时标 YT
C_SE_TC_1.x		短浮点设定值带时标 YT

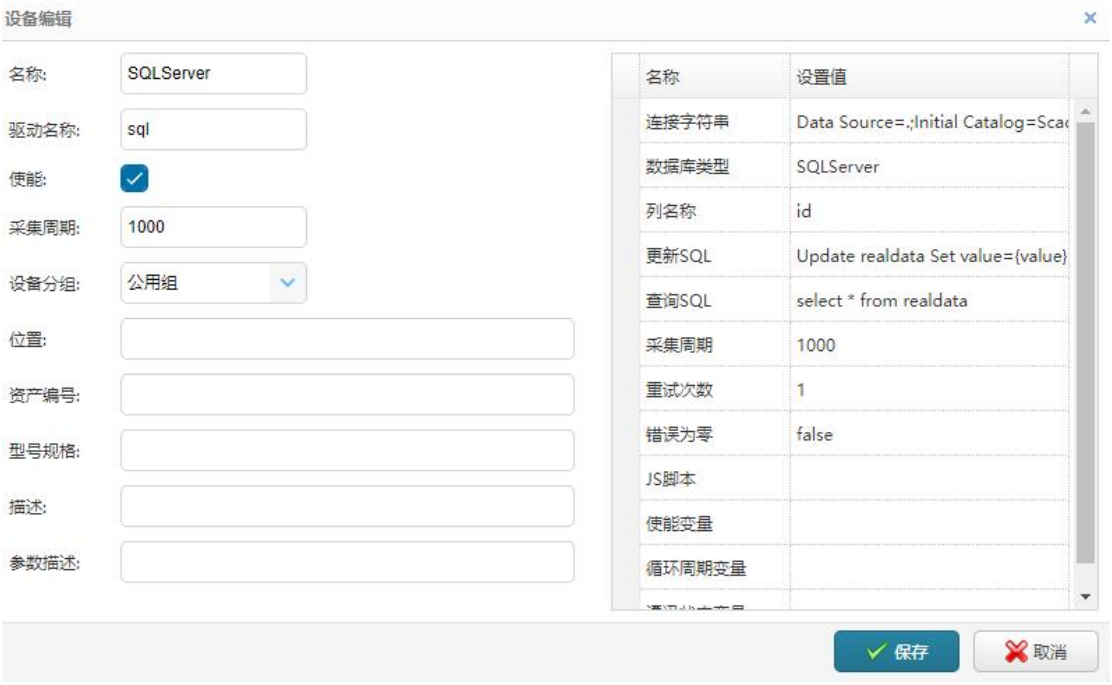
下表是标准 IEC104 地址数据

	1997 版	2002 版
YX	1H--------400H	1H---------4000H
YC	701H------900H	4001H------5000H
YK	b01H-------b80H	6001H------6100H
YT	B81H-------c00H	6201H------6400H
YM	C01H-------c80H	6401H------6600H

104 协议工作流程：连接成功发送总招命令，之后开始接收设备的推送信息，每 15 分钟发送一次总招。

8.10 关系库驱动

关系数据库驱动的工作原理是通过指定的 SQL 查询获取数据,根据标识列获取某行，然后根据列获取数据。



设备配置参数

名称	设置值	说明
连接字符串	Data Source=.;Initial Catalog=Scada;User ID=iot;Password=iotScada;Connect Timeout=30;	数据库连接字符串，不同数据连接字符串不一样
数据库类型	SQLServer	支持 SQLServer 和 MySQL
列名称	id	标识列名称
更新 SQL	Update realdata Set value={value} where id={id}	更新数据的 SQL
查询 SQL	select * from realdata	查询数据的 SQL
采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

关系数据库驱动变量驱动地址说明：

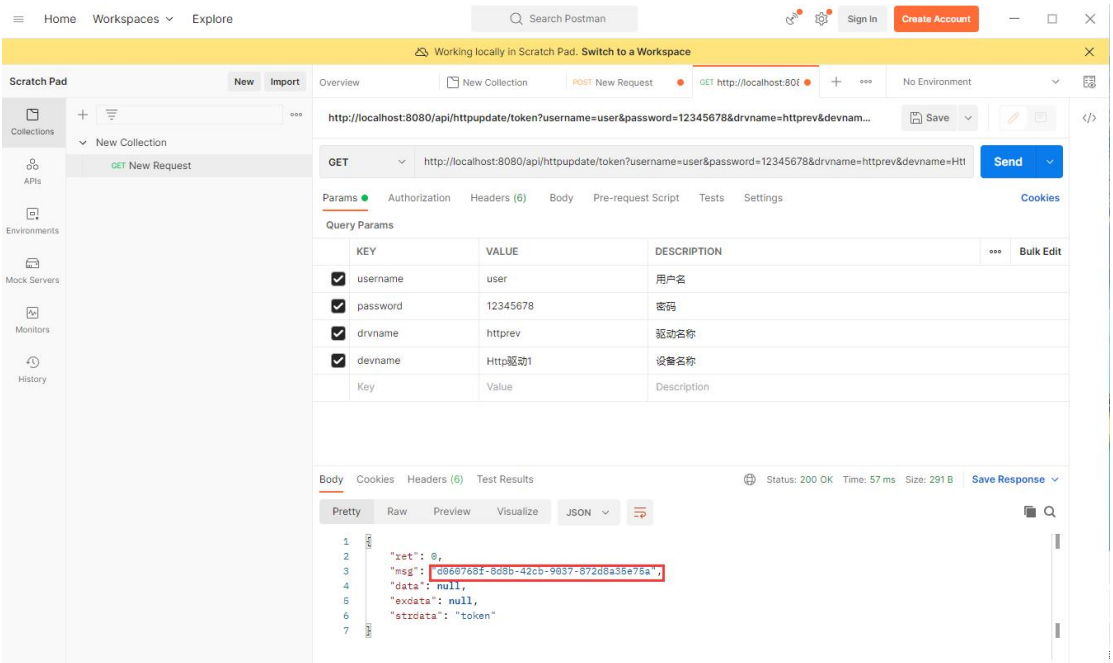
驱动地址	扩展驱动地址	说明
1@value		标识@列名 更加行标识和列名获取值更到变量

上述配置行标识为 1，列名为 value 的值

8.11 HTTP 接收驱动

HTTP 驱动接收来自外部的 Http 数据更新到组态变量中，Http 接收 POST 数据，一般需要获取 token。

获取 token：GET URL:/api/httpupdate/token



1 个设备只允许 1 个 token 更新数据，每次获取 token 后之前的 token 会失效

POST URL：/api/httpupdate/updatetags?token=xxxx ，Post 数据格式如下：

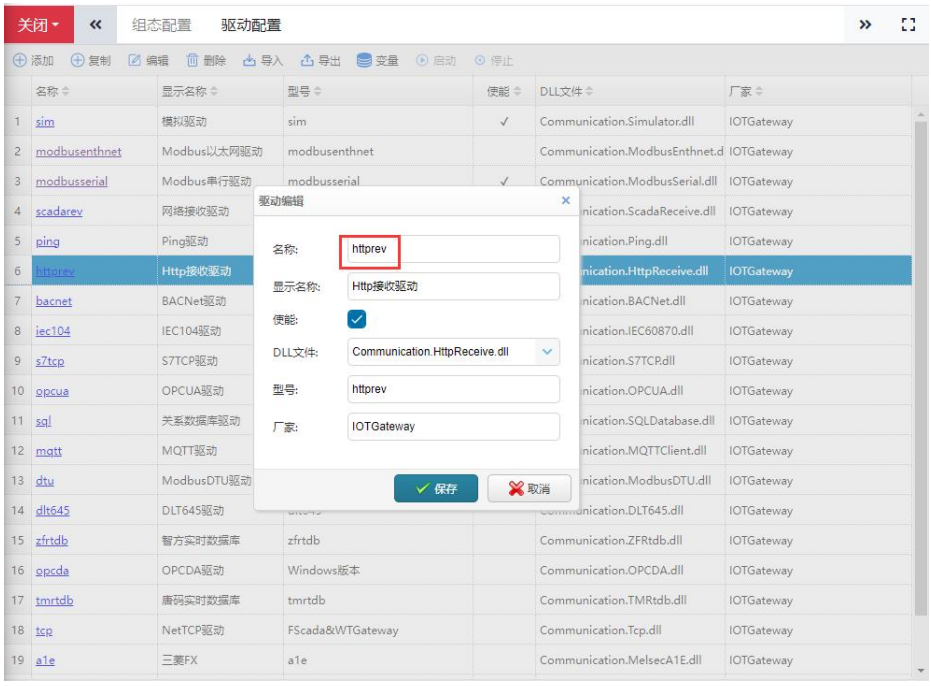
```
{
    "drvName": "httprev",
```

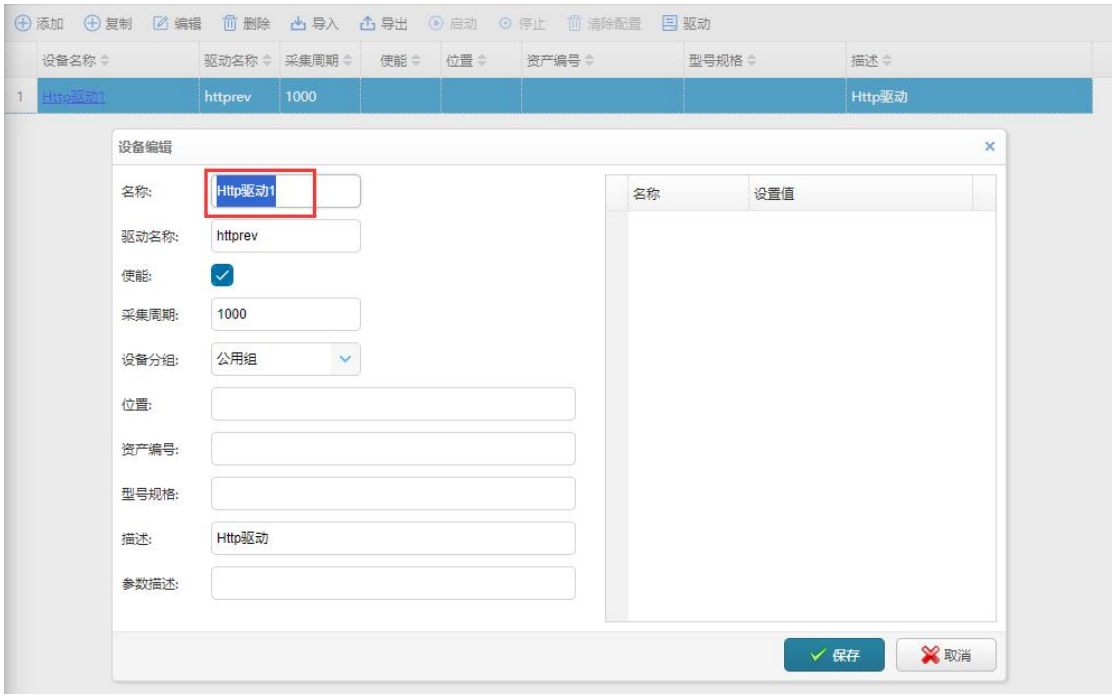
```
    "devName": "Http 驱动 1",
    "datas": [
        {
            "id": "HttpTag1",
            "v": 1,
            "s": 1
        },
        {
            "id": "HttpTag2",
            "v": 12,
            "s": 1
        }
    ]
}
```

id:变量名称

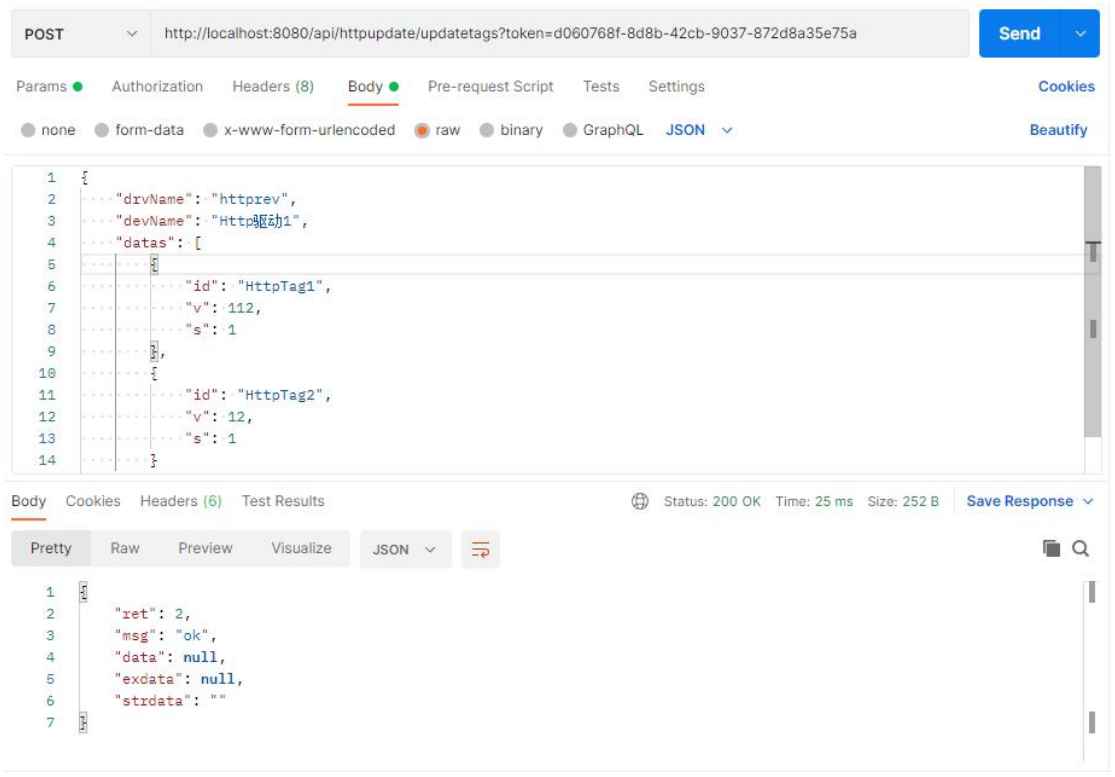
v：变量值

s：变量状态 0：未知 1：好点 2：坏点




添加
复制
编辑
删除
导入
导出
启动
停止
清除配置
驱动
设备名称
驱动名称
采集周期
使能
位置
资产编号
型号规格
描述
1
Http驱动1
httprev
1000
Http驱动
设备编辑
名称:
Http驱动1
驱动名称:
httprev
使能:
采集周期:
1000
设备分组:
公用组
位置:
资产编号:
型号规格:
描述:
Http驱动
参数描述:
名称
设置值
保存
取消


系统配置
系统状态
实时数据库
组态环境
运行环境
导出备份
关闭
驱动配置
添加
复制
编辑
删除
清除
导入
导出
启动
停止
设备
驱动
通讯测试
设备读取
名称:
描述:
变量名称
数据类型
设备类型
地址信息
读写
单位
下限
上限
周期
小数
归档
报警
高报警
高高报警
低报警
低低报警
报警级别
保存
操作
变化
转发
描述
1
HTTPTAG1
Int32
Int32
R
0
100
1000
0
80
100
20
10
0
✓
2
HTTPTAG2
Int32
Int32
R
0
100
1000
0
80
100
20
10
0
✓


POST
http://localhost:8080/api/httpupdate/updatetags?token=d060768f-8d8b-42cb-9037-872d8a35e75a
Send
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
Cookies
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
Beautify
{
"drvName": "httprev",
"devName": "Http驱动1",
"datas": [
{
"id": "HttpTag1",
"v": 112,
"s": 1
},
{
"id": "HttpTag2",
"v": 12,
"s": 1
}
Body
Cookies
Headers (6)
Test Results
Status: 200 OK
Time: 25 ms
Size: 252 B
Save Response
Pretty
Raw
Preview
Visualize
JSON
{
"ret": 2,
"msg": "ok",
"data": null,
"exdata": null,
"strdata": ""
}

当用户名为空白或者“null”时不使用 Token 认证。

通过 JS 脚本,驱动也能支持反向写入(onWrite),比如可以使用 rest 发送变量设置数据到注定的服务接口。

RestHttp 接口

```
var restHttp = new rest("http://fscada.net:8080");
var res = "";
function onWrite(adr,value){ //模拟驱动更新函数
    restHttp.addParam("tagName",adr);
    restHttp.addParam("value",value);
    res = restHttp.postForm("/rest/writetagvalue");//Form 表单发送
    restHttp.clearParams();

    //var obj = {"tagName":adr,"value":value};
    //res = restHttp.postRequest("/json/xx",JSON.stringify(obj));//Post Json
}
```

8.12 BACNet 驱动

设备配置参数

名称	设置值	说明
通讯模式	IP	IP 或者 Mstp
本机 IP 地址		指定本机用于通讯的网卡 IP 地址
网关 IP 地址		透过网关通讯时指定网关 IP 地址

通讯端口	47808	BacNet 默认通讯端口
独占端口	true	默认 true，非共享端口
默认写入级别	8	默认 8，写入级别
写入级别变量		写入级别变量，变量为整数类型，当值指定变量后写入值时使用该变量的值作为写入级别
Mstp 串行端口	COM1	Mstp 通讯的串行端口
Mstp 波特率	38400	Mstp 通讯的波特率
Mstp 源地址	127	Mstp 通讯的源地址
通讯超时	5000	单位 ms
最大读取数量	20	批量读取参数，支持批量读取的设备可以设置
采集周期	1000	不使用
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

BACNet 驱动变量驱动地址说明：

驱动地址格式：类型.地址.值域

例：AI.1（支持 AI AO BI BO AV BV MSI MSO MSV）

例：ANALOG_INPUT.1.PRESENT_VALUE

驱动地址格式如下：

驱动地址	扩展驱动地址	说明
ANALOG_INPUT.x.y	设备节点号	模拟量输入，浮点数 AI
ANALOG_OUTPUT.x.y	设备节点号	模拟量输出，浮点数 AO
BINARY_INPUT.x.y	设备节点号	数字量输入，开关量 BI
BINARY_OUTPUT.x.y	设备节点号	数字量输出，开关量 BO
ANALOG_VALUE.x.y	设备节点号	模拟量值，浮点数 AV
BINARY_VALUE.x.y	设备节点号	数字量值，开关量 BV

MULTI_STATE_INPUT.x.y	设备节点号	多态值输入，整数 MSI
MULTI_STATE_OUTPUT.x.y	设备节点号	多态值输出，整数 MSO
MULTI_STATE_VALUE.x.y	设备节点号	多态值，整数 MSV

值域可以使用数字，例如：ANALOG_INPUT.1.85

（1）扩展驱动地址： NodeID,SNet,SAdr

例：200,0,0 或 200

当 SNet 或者 SAdr 大于 0 时 并且网关 IP 地址配置不为空白,则使用路由地址。

（2）本机 IP 地址说明

本机 IP 地址应该配置为本机某 1 个网卡的地址如 192.168.1.201，当子网掩码为 255.255.255.0 时 UDP 广播地址为 192.168.1.255，当子网掩码为 255.255.0.0 时 UDP 广播地址为 192.168.255.255，当 IPAdress 为 0.0.0.0 时使用 255.255.255.255 广播地址。

支持的值域参数如下：

名称	值	说明
PRESENT_VALUE	85	实时值
ALARM_VALUE	6	报警值
DESCRIPTION	28	描述
FEEDBACK_VALUE	40	反馈值
OBJECT_NAME	77	对象名称
OUT_OF_SERVICE	81	服务状态
POLARITY	84	极性
PRIORITY	86	级别
PRIORITY_FOR_WRITING	88	写入级别
SETPOINT	108	设定值
STATE_TEXT	110	状态文本
STATUS_FLAGS	111	状态标志
ENABLE	133	使能

8.13 智方实时数据库驱动

设备配置参数

名称	设置值	说明
连接字符串	server=192.168.10.17;port=5130;username=sa;password=zfrtdb	数据库连接字符串
采集周期	1000	
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动支持实时快照值写入，变量的驱动地址配置为数据库的变量全名称，变量的采集周期（ms）控制采集频率。不允许写入的变量配置为只读。一个设备中不允许有相同驱动地址的变量，如果有则重复的变量不会更新。驱动支持在线添加变量，修改变量和删除变量。驱动支持通讯测试和在线导入数据库的变量。

8.14 OPCDA 驱动

设备配置参数

名称	设置值	说明
服务器地址	localhost	OPC 服务器地址
服务器名称	WT.OPCServer	OPCServer 名称
OPCServer 类名		注册表类型
心跳检测	false	是否使用心跳检测
心跳检测时间	300	单位秒 心跳检测的用途是检测 OPCServer 是否发生了异常，比如突然崩溃了，检测到异常后 10 秒后自动重启启动 OPCServer
使用 OPC 时间	false	变量时间是否使用 OPCServer 的变量时间
读取后订阅	false	连接成功后是否先读取一次再订阅
采集周期	1000	默认采集周期 ms
JS 脚本		JS 脚本文件名称
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

OPCDA 驱动仅支持 Windows 系统，需要安装 OPC 64 位发行组件包。

8.15 DLT645 驱动

设备配置参数

名称	设置值	说明
通讯类型	TCP2007	支持 1997、2007 串口和 TCP 方式
IP 地址	192.168.1.101	TCP 方式的 IP 地址
TCP 通讯端口	5020	TCP 方式的通讯端口
使用 Ping	true	Ping 网络检测使能
通讯端口	COM1	串行口
波特率	9600	
数据位	8	
校验方式	None	
握手方式	None	
停止位	One	
通讯等待时间	0	每次采集等待 ms 数
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型

		0：通讯正常 10000：网络不通

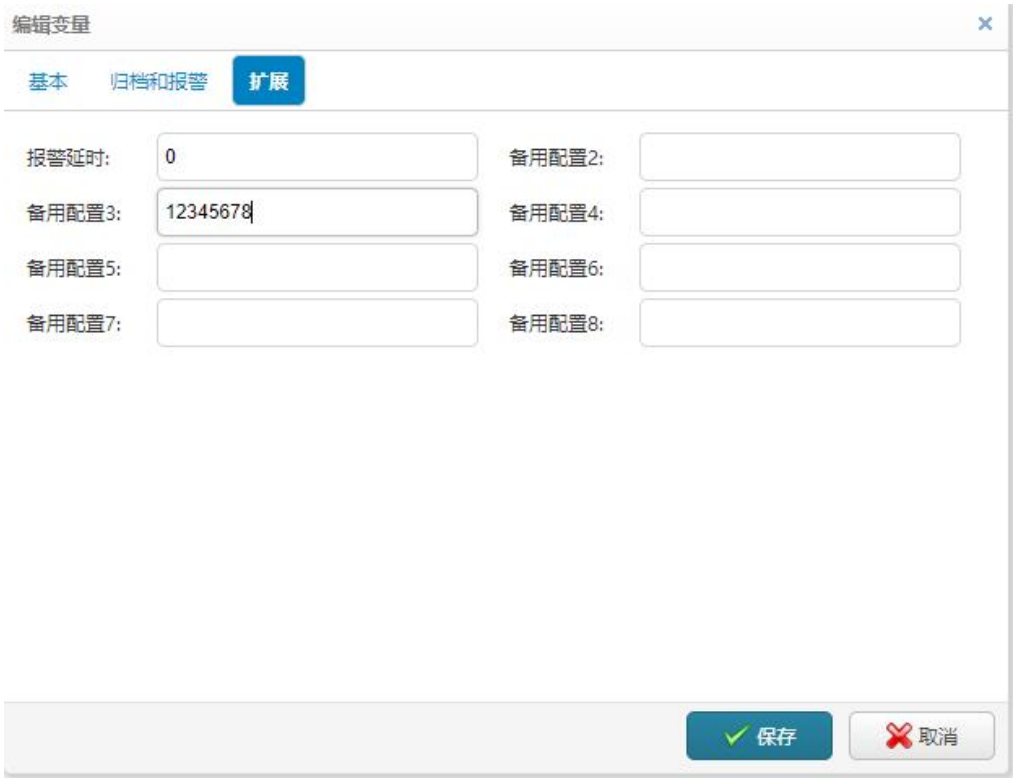
DLT645 是国标电度表通讯协议。

2007 版本协议使用 4 位驱动地址：00-00-00-01

1997 版本协议使用 2 位驱动地址：00-00

扩展参数：设置数据格式

备用配置 3：设置仪表地址



8.16 DTU 驱动

设备配置参数

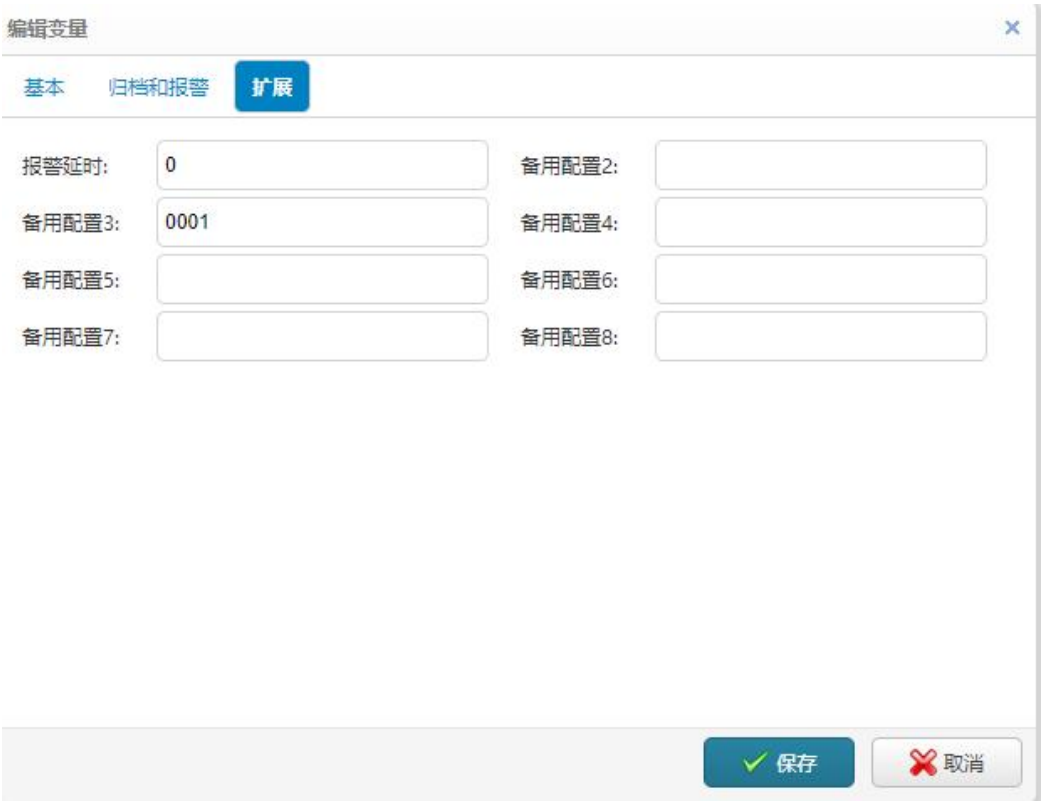
名称	设置值	说明
通讯协议	TCP	支持 Modbus TCP 和 Modbus RTU 协议
服务器 IP 地址		服务器绑定的 IP 地址

TCP 服务端口	5010	绑定的通讯端口
16 进制注册包	false	注册包格式是 16 进制
IP 地址注册	false	使用设备的 IP 地址作为注册包,否则使用第一个通讯包作为注册包
字节顺序	CDAB	
字符编码	Ascii	字符变量使用
字符交换字节	false	字符变量使用
等待时间	0	每次通讯等待时间 ms
离线时间	60	多长时间没有收到数据判断为离线 s
读取超时	10	通讯超时时间单位秒
最大读取字	120	最大读取字个数
0x10 写入	true	使用 10 功能码写入 16 位数据 否则使用 0x06
掩码写入	false	使用掩码方式写入 bit
采集周期	1000	单位秒
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通
字符编码	Ascii	文本的字符编码方式,支持 UTF8 Ascii Unicode GB2312 BigUnicode

DTU 通讯仅支持 Modbus TCP 和 RTU 协议。

备用配置 3：设备注册包

地址格式：Modbus



8.17 唐码实时数据库驱动

设备配置参数

名称	设置值	说明
连接字符串	ws://127.0.0.1:921;admin;admin	websocket 地址；用户名；密码
采集周期	1000	单位 ms，最小 1000ms

重试次数	1	不使用
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

根据变量的采集周期循环从实时数据库读取快照数据，支持值写入到快照。

变量不会写入历史数据到唐码实时数据库中，变量的历史查询自动使用变量地址执行。

8.18 NETTCP 驱动

NETTCP 驱动用于连接 FScada、WTGateway TCP 服务，采集实时数据，也可以连接 IOTGateway TCP 扩展服务和 OPCLink 的 TCP 服务器。

设备配置参数

名称	设置值	说明
用户名	admin	用户名
密码	admin	密码
主服务器地址	127.0.0.1	服务器 IP 地址
备用服务器地址		备用服务器 IP 地址

主 TCP 端口	8000	TCP 通讯端口
备用 TCP 端口	0	
老版本	false	FScada 使用 true
只读	false	变量只读
采集周期	1000	单位 ms，最小 1000ms
重试次数	3	通讯重试次数
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

8.19 三菱 PLC 驱动（A1E）

A1E 驱动用于连接 FX0 2 3 PLC，使用 TCP 通讯，FX 一般使用第三方网络转换器转换成以太网通讯。

设备配置参数

名称	设置值	说明

IP 地址	192.168.0.100	PLC IP 地址
TCP 端口	5001	PLC 配置的 TCP 通讯端口
使用 Ping	true	是否使用 ping 检测设备网络状态
Ascii 协议模式	false	Ascii 或者二进制协议，根据 plc 设置配置
最大读取字	64	FX 最大 64
重连等待时间	5000	单位 ms，重连等待时间
网络地址	255	设备网络地址
通讯超时	1000	单位 ms，通讯超时时间
故障次数	5	单位次，判断通讯故障的次数
故障时间	60	单位秒，通讯故障后恢复检测时间
采集周期	1000	单位 ms，最小 1000ms
重试次数	3	通讯重试次数
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动变量驱动地址说明：

驱动地址	数据类型	说明
TVx TNx CVx CNx	Int16	TV0 读取定时器值 TN0 读取计数器值
TSx CSx CCx	Boolean	TS0 定时器输出 CS0 计数器输出
Xx	Boolean	X01 读取输入，X01 X07 X010

Yx		Y01 读取输出 以 0 开头表示 8 进制地址格式，否则为 16 进制格式
Mx Sx	Boolean	M0　读取 M 地址 10 进制格式 S0　读取系统地址 10 进制格式
Dx Wx Rx Fx Bx	Int16 Int32 Single	16 位数据区 D0，R0，F0 读取数据区　10 进制格式 W0，B0，16 进制格式
Dx.y	Boolean	D0.1　读取数据位
Dx.y	String	Dx.8　读取字符串 16 个字节 字符类型 Ascii

说明：如果启用了 Ping 操作，设备 Ping 成功才开始通讯采集，ping 失败时不采集。

8.19 三菱 PLC 驱动（Qna3E）

Qna3E 驱动用于连接 FX5 和 Q 系列 PLC，需要配置 PLC 配置一条通讯线路，每个线路仅支持 1 个通讯连接。

设备配置参数

名称	设置值	说明

IP 地址	192.168.0.100	PLC IP 地址
通讯方式	UDP	TCP 或者 UDP 模式
TCP 端口	5002	PLC 配置的通讯端口
协议类型	Binary	二进制或者 Ascii 协议
使用 Ping	true	是否使用 ping 检测设备网络状态
Ascii 协议模式	false	Ascii 或者二进制协议，根据 plc 设置配置
最大读取字	250	最大支持 250
重连等待时间	5000	单位 ms，重连等待时间
网络地址	255	设备网络地址
通讯超时	1000	单位 ms，通讯超时时间
故障次数	5	单位次，判断通讯故障的次数
故障时间	60	单位秒，通讯故障后恢复检测时间
使用 Ping	true	使用 Ping 检测 PLC 网络状态
采集周期	1000	单位 ms，最小 1000ms
重试次数	3	通讯重试次数
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动变量驱动地址说明：

驱动地址	数据类型	说明
TVx TNx CVx CNx	Int16	TV0 读取定时器值 TN0 读取计数器值
TSx	Boolean	TS0 定时器输出

TCx CSx CCx SSx SCx		CS0 计数器输出
Xx Yx	Boolean	X01 读取输入 16 进制格式 Y01 读取输出 16 进制格式
Mx Lx Sx Fx Vx	Boolean	M0 读取 M 地址 10 进制格式 S0 读取系统地址 10 进制格式
Dx Wx ZRx Zx Bx Rx	Int16 Int32 Single	16 位数据区 D0，R0，F0 读取数据区 10 进制格式 W0，B0，ZR0 16 进制格式
Dx.y	Boolean	D0.1 读取数据位
Dx.y	String	Dx.8 读取字符串 16 个字节 字符类型 Ascii

说明：如果启用了 Ping 操作，设备 Ping 成功才开始通讯采集，ping 失败时不采集。

8.20 欧姆龙 PLC 驱动

欧姆龙 PLC 使用 FinsNet 协议，支持 TCP 和 UDP 方式。

设备配置参数

名称	设置值	说明
IP 地址	192.168.0.100	PLC 的 IP 地址
TCP 端口	5001	PLC 的 TCP 端口
协议类型	TCP	TCP 或者 UDP
使用 Ping	true	使用 Ping 检测 PLC 网络状态
上位机节点地址	1	上位机的节点地址，假如你的电脑的 Ip 地址为 192.168.0.13，那么这个值就是 13，多个电脑连接这个值不能相同
上位机单元号	0	上位机的单元号地址
PLC 节点地址	0	PLC 的节点地址，这个值在配置了 ip 地址之后是默认赋值的，默认为 Ip 地址的最后一位
PLC 单元地址	0	PLC 的单元号地址
通讯包尺寸	256	单位:字节，通讯数据包尺寸，最大 512
通讯超时	1000	单位：ms
故障次数	5	判断通讯故障的次数
故障时间	60	单位：秒,判断通讯故障的时间

采集周期	1000	单位 ms，最小 1000ms
重试次数	3	通讯重试次数
JS 脚本		JS 脚本文件名称
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

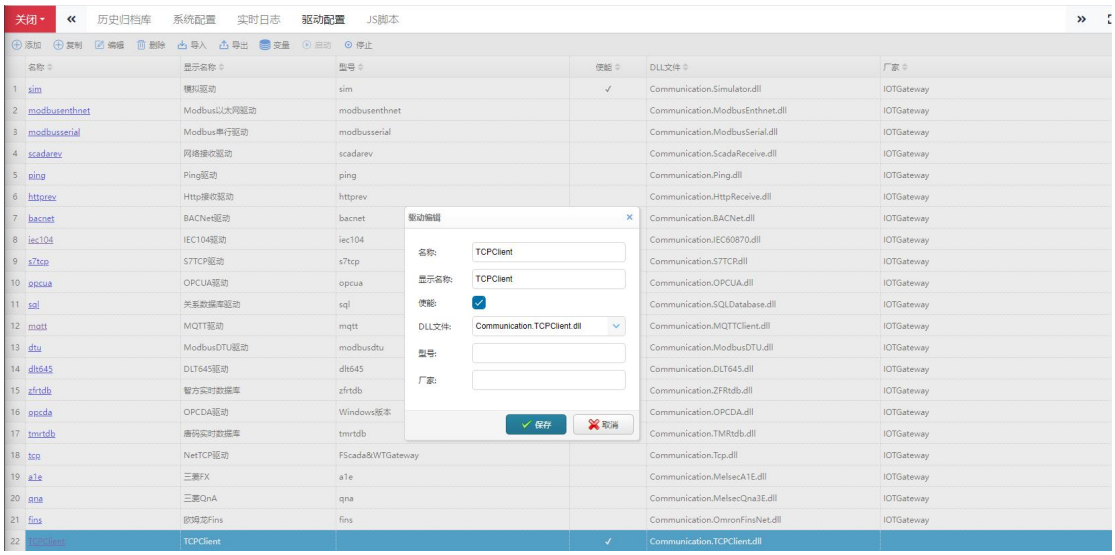
驱动变量驱动地址说明：

驱动地址	数据类型	说明
CIOx.y	Boolean	CIO0.0 输入输出读写
CIOx	Int16 Int32	CIO 输入输出寄存器读写 10 进制地址
HRx.y ARx.y WRx.y Dx.y	Boolean	AR0.0 输入输出读写 10 进制地址
HRx ARx WRx Dx	Int16 Single Int32	HR0 寄存器读写 10 进制地址 AR0 寄存器读写 10 进制地址 WR0 寄存器读写 10 进制地址 D0 寄存器读写 10 进制地址

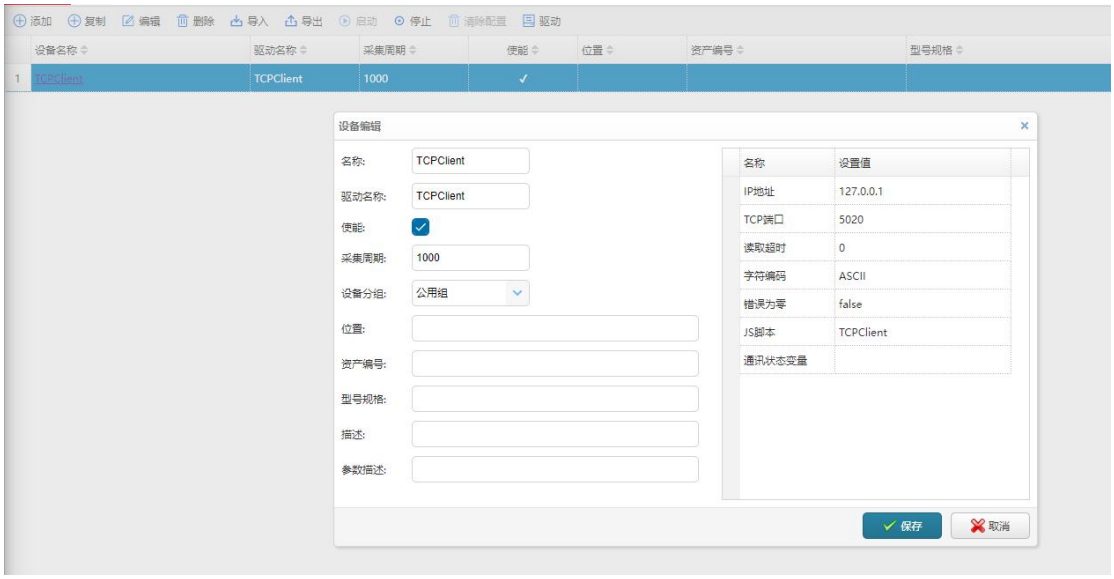
说明：如果启用了 Ping 操作，设备 Ping 成功才开始通讯采集，ping 失败时不采集。

8.21 TCPClient 驱动

（1）添加驱动



（2）添加设备



JS 脚本：用于解析的 javascript 代码名称
读取超时：单位毫秒，设置为 0 不超时，否则指定时间收不到数据会断开连接重新连接
字符编码：字节转换为字符的编码类型，默认 ASCII
IP 地址和端口：TCP 服务器的 IP 地址和 TCP 端口

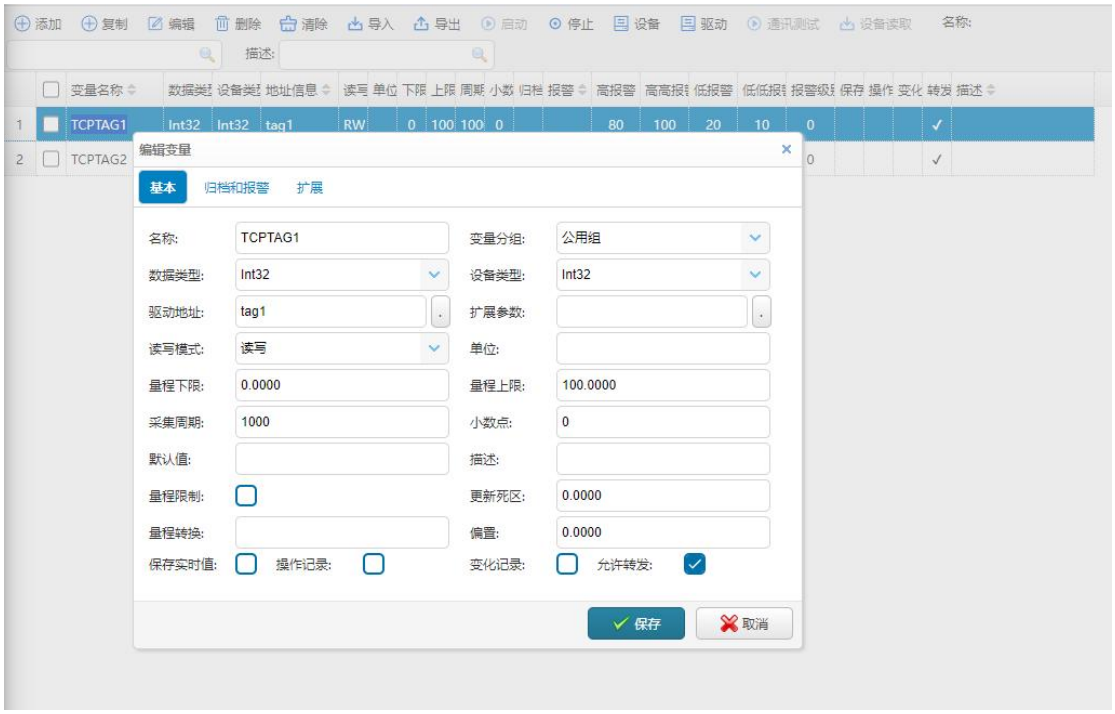
（3）JS 代码格式

必须包含 onReceive 函数，内置对象 tcp 是 TCP 连接对象，可用于发送文本或者二进制数据

```
function onReceive(str,hexstr){
    $Tag("tcpTag1").DoUpdate(k);//更新变量值
    //app.LogInfo("String",str);
    //app.LogInfo("HexString",hexstr);
}
```

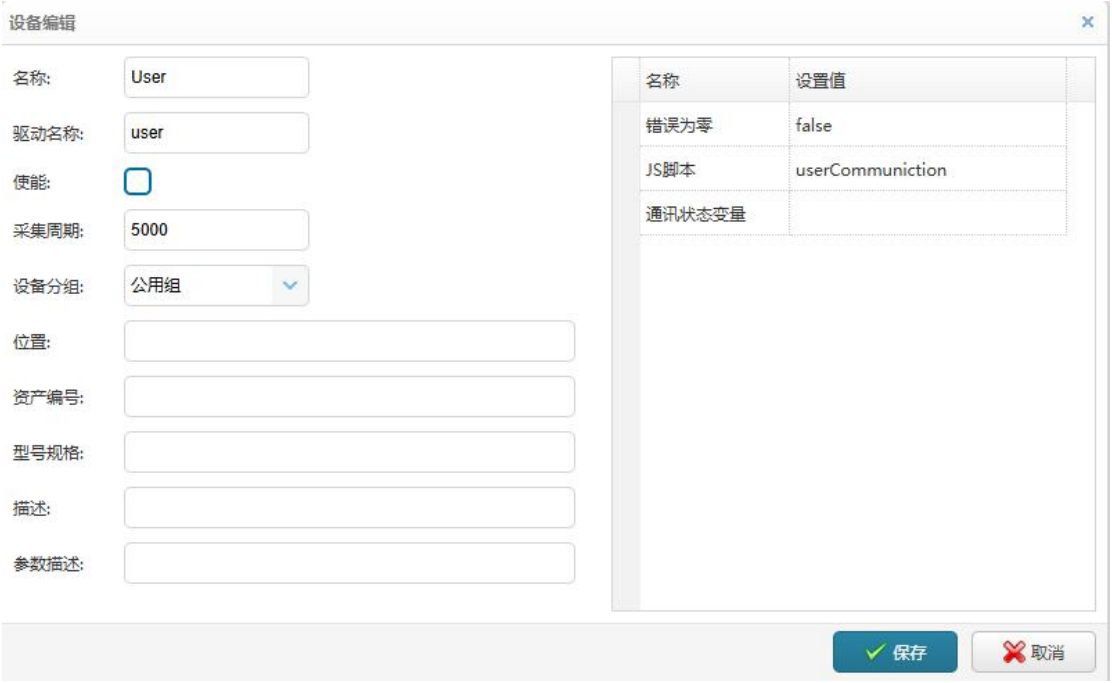
```
str：接收到的字符串，如 123
hexstr：接收到的 16 进制字符串，如 31 32 33
//变量写入回调函数 drv 是驱动信息 data 是值 连接后会执行一次 drv 和 data 均为空白
function onWrite(drv,data){
if (drv && data)
  tcp.sendMessage( drv + " = " + data);
else
    app.LogInfo("TCPClient","连接成功");
}
//循环执行 可以发送文本或者二进制文本
function onRequest(){
//tcp.sendBytes("30 31 32 33 34 35 36 37");
//tcp.sendMessage("1234567");
}
```

注意：通讯变量驱动地址不要空白，否则变量会被当做计算变量使用



8.22 用户驱动

用户驱动使用 JavaScript 脚本实现交互

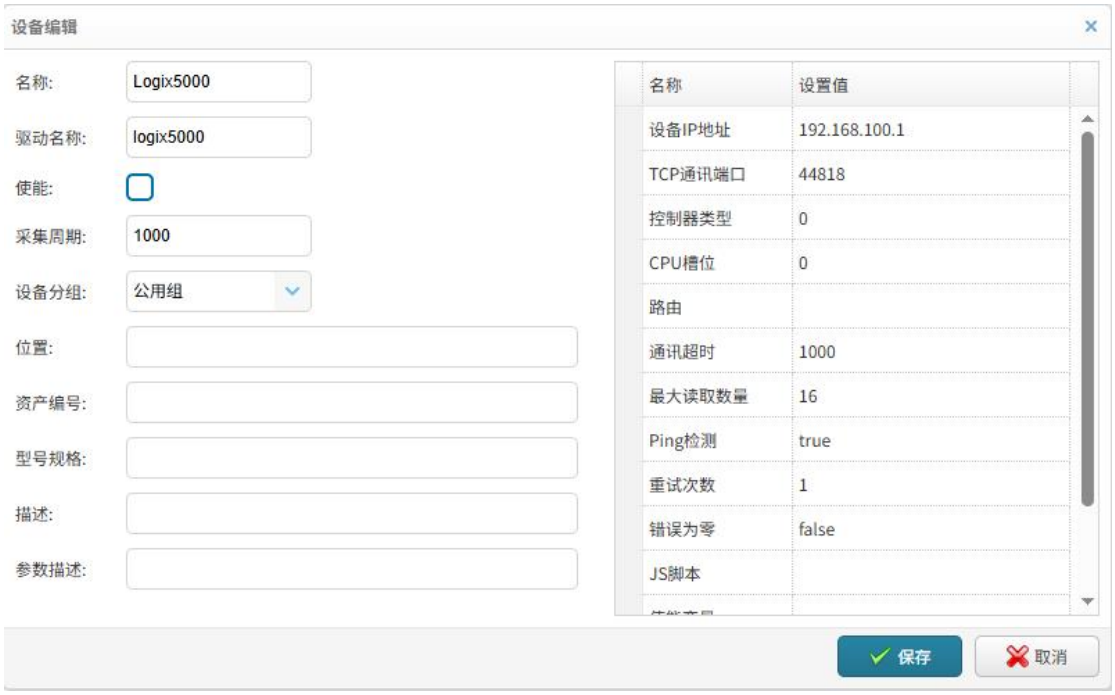



```
var restHttp = new rest("http://fscada.net:8080");
var res = "";
var tag1 = $Tag("userTag1");
function onStart(device){
  app.LogInfo("用户驱动",device.Name + " Started","",false);
}
function onStop(){
  restHttp = null;
  tag1 = null;
}
function onRequest(){
  restHttp.addQueryParam("tagNames","SIMTAG4,second,minute");
  res = restHttp.getRequest("/rest/gettagvalues");
  if (res){
    var json = JSON.parse(res);
    if (json){
      tag1.DoUpdate(json.rows[0].value);
    }
  }
}
function onWrite(adr,value){
  restHttp.addParam("tagName", adr);
  restHttp.addParam("value",value);
  res = restHttp.postForm("/rest/writetagvalue?token=B579C28B0FC8");
}
```

用户驱动的 js 代码基本格式如下，必须包括 onStop、onRequest、onWrite 函数，

onRequest 被驱动循环调用，onWrite 用于变量写入执行。

8.23 Logix5000 驱动



设备配置参数

名称	设置值	说明
设备 IP 地址	192.168.1.100	PLC 通讯 IP 地址
TCP 端口	44818	TCP 通讯端口
控制器类型	0	0:Logix5000，1:Micro800
CPU 槽位	0	CPU 在背板的位置
路由		EIP 通讯路由，如 1.2.2
通讯超时	1000	单位 ms
最大读取数量	16	批量读取数，该值跟变量名称长度有关，根据实际情况可调整 8-20
Ping 检测	true	网络连接测试
重试次数	1	通讯故障重试次数
错误为零	false	通讯故障时是否把变量值设置为 0
JS 脚本		JS 文件名称，每个采集周期执行的 js 脚本
采集周期	1000	单位秒

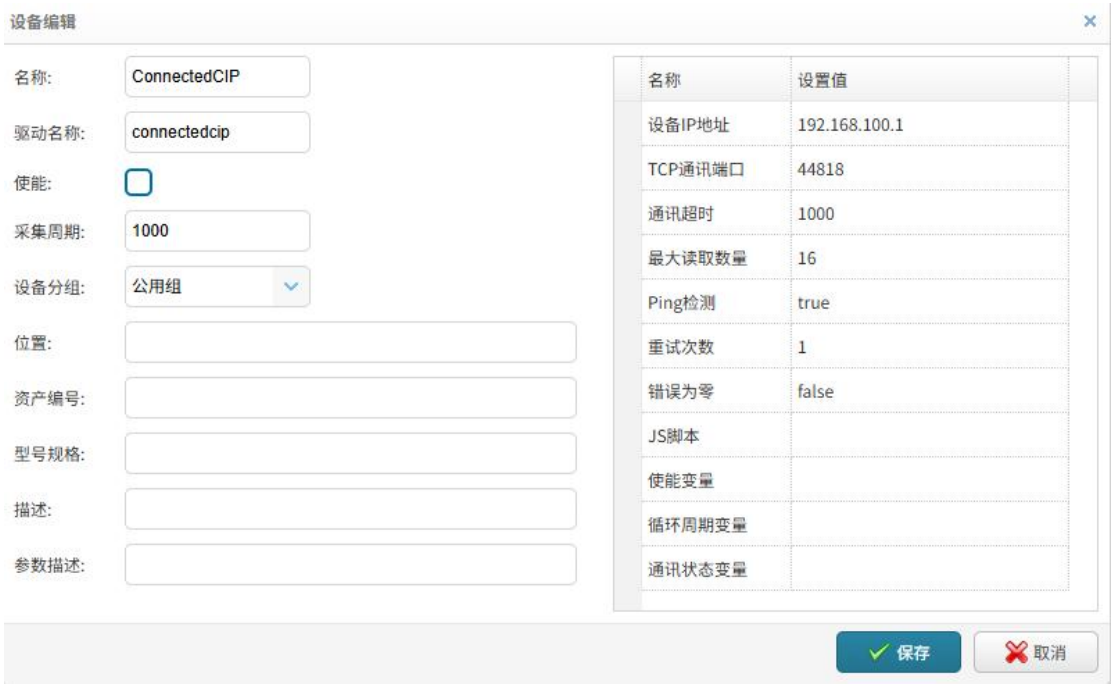
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动变量驱动地址说明：

驱动地址	扩展驱动地址	说明
TagName		控制器变量名称
A1[0]		数组
A3[0].DI	数据偏移可选	结构数组变量

说明：Logix5000 控制器批量读取变量数比较小，相同数据类型的变量进行打包批量采集，1000 个变量采集或许需要数秒，但是 1000 个 Int 数组几十毫秒就能采集到。

8.24 ConnectedCIP 驱动



设备配置参数

名称	设置值	说明
设备 IP 地址	192.168.1.100	PLC 通讯 IP 地址

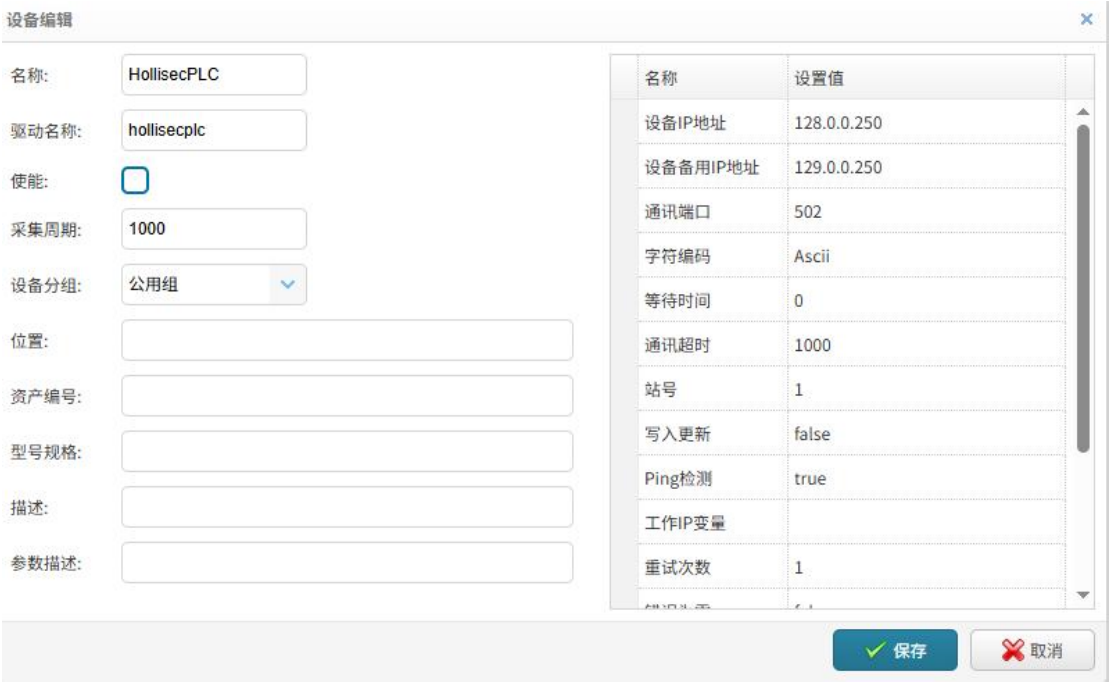
TCP 端口	44818	TCP 通讯端口
通讯超时	1000	单位 ms
最大读取数量	16	批量读取数，该值跟变量名称长度有关，根据实际情况可调整 8-64
Ping 检测	true	网络连接测试
重试次数	1	通讯故障重试次数
错误为零	false	通讯故障时是否把变量值设置为 0
JS 脚本		JS 文件名称，每个采集周期执行的 js 脚本
采集周期	1000	单位秒
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动变量驱动地址说明：

驱动地址	扩展驱动地址	说明
TagName		控制器变量名称
A1[0]		数组
A3[0].DI	数据偏移可选	结构数组变量

说明：Logix5000 控制器批量读取变量数比较小，相同数据类型的变量进行打包批量采集，1000 个变量采集或许需要数秒，但是 1000 个 Int 数组几十毫秒就能采集到。该驱动不支持 Bool 数组的写入。

8.25 和利时 PLC 驱动



设备配置参数

名称	设置值	说明
设备 IP 地址	128.0.0.250	设备的 IP 地址
设备备用地址	129.0.0.250	设备的备用 IP 地址
TCP 通讯端口	502	设备的通讯端口，默认 502
字符编码	UTF8	文本的字符编码方式，支持 UTF8 Ascii Unicode
等待时间	0	每次通讯完成后是否需要等待 n 毫秒，wifi 通讯方式可能需要设置等待几十毫秒，防止出现粘包导致设备无法响应查询
通讯超时	1000	通讯的超时判断，单位 ms
站号	1	Modbus 站号
写入更新	false	设置变量值后是否立即更新变量为写入值，通常应该等待采集后更新
Ping 检测	false	是否使用 ping 检测设备正常才开始通讯采集
重试次数	2	通讯故障重试次数
错误为零	false	通讯故障时是否把变量值设置为 0
JS 脚本		JS 文件名称，每个采集周期执行的 js 脚本
工作 IP 变量名称		指示工作的 IP 地址

使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通
字符编码	Ascii	文本的字符编码方式，支持 UTF8 Ascii Unicode GB2312 BigUnicode

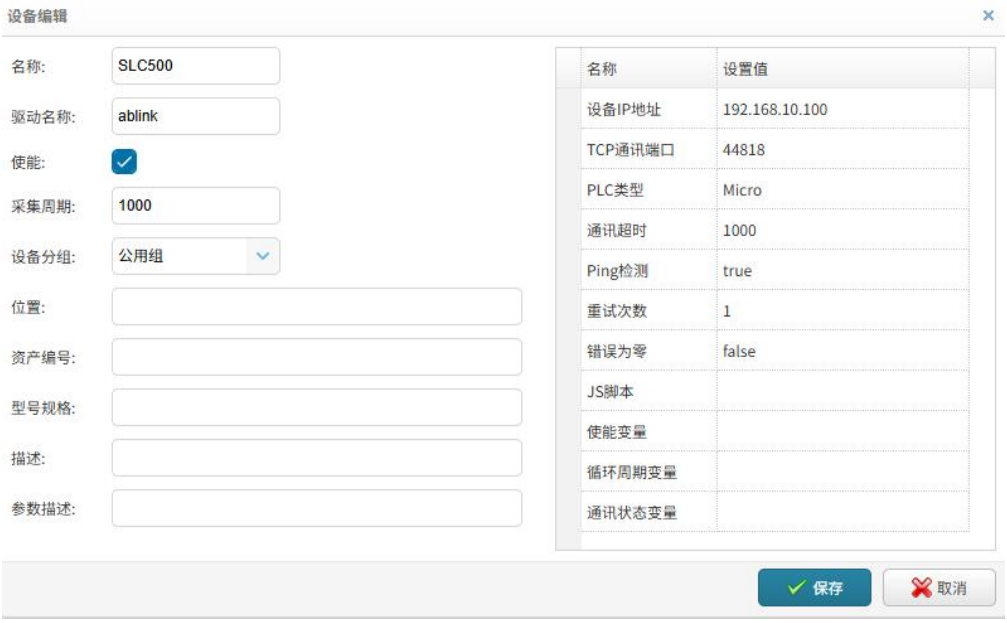
驱动变量驱动地址说明：

驱动地址	类型	说明
%IX0.1	Boolean	输入信号
%QX0.1	Boolean	输出信号
%IW0	INT,Single	模拟量输入
%IW0.0	Boolean	
%QW0	INT,Single	模拟量输入
%QW0.0	Boolean	模拟量输入
%MX0.0	Boolean	
%MW0	INT,Single	模拟量
%MW0.0	Boolean	

和利时 PLC 采用 Modbus TCP 协议通讯，支持双网口冗余。

8.26 AB PLC 驱动

AB SLC500 和 Micro 以太网驱动。



设备配置参数

名称	设置值	说明
设备 IP 地址	192.168.10.100	设备的 IP 地址
TCP 通讯端口	44848	设备的通讯端口，默认 44818
PLC 类型	SLC	SLC500 Micro
通讯超时	1000	通讯的超时判断，单位 ms
Ping 检测	false	是否使用 ping 检测设备正常才开始通讯采集
重试次数	2	通讯故障重试次数
错误为零	false	通讯故障时是否把变量值设置为 0
JS 脚本		JS 文件名称，每个采集周期执行的 js 脚本
使能变量		通讯使能变量 布尔类型
循环周期变量		通讯周期显示变量 整数类型
通讯状态变量		通讯状态显示变量 整数类型 0：通讯正常 10000：网络不通

驱动变量驱动地址说明：

驱动地址	类型	说明
B3:0/0	Boolean	布尔量文件
N7:0	Int16	整数文件

N7:0/1	Boolean	位读写
F8:0	Single	浮点数文件
L9:0	Int32	32 位整数文件
L9:0/3	Boolean	位读写

9. REST 接口

使用 API 文档工具找到 REST 接口部分，除了写入需要 token 授权，查询接口都可以直接访问，token 是用户设置每个用户配置的 token 值。

10. 操作系统支持和授权方式

1）Windows 64 位操作系统（Windows7 及后续版本），软件授权或者 USB 授权

2）主流 Linux 64（x86 平台）位操作系统，软件授权或者 USB 授权

3）主流 Liunx 64（ARM 平台）位操作系统，软件授权或者 USB 授权

4）Linux Docker，USB 授权

附件 1：用于 JS 脚本文件的内置 JavaScript 函数（后端使用）

JavaScript 语言区分大小写，所有英文字母都是半角

一）内置对象

1）app：应用程序对象

函数	说明
app.BasePath	返回软件根目录名称字符串
app.ProjectPath	返回项目路径名称字符串
void app.WriteColReport(string ruleName,bool direct=false)	写入报表归档
bool app.WriteColReport(string ruleName,string datetime, directWrite = false)	指定时间写入报表归档
bool app.WriteRowReport(string ruleName, directWrite = false)	写入行表归档
bool app.WriteRowReport(string ruleName, string datetime, directWrite = false)	指定时间写入行表归档
bool app.WriteRowReportTag(string ruleName,string tagName)	写入单个变量到行表归档中
bool app.WriteRowReportTag(string ruleName, string datetime, string tagName)	指定时间写入单个变量到行表归档中
bool app.PluseTag(string tagName, object inteval)	变量脉冲操作,例如：app.PulseTag("tag1",5000) 设置变量 tag1 的值为 1（True）,5 秒后再设置为 0（False）
bool app.SetTagValue(string tagName, object value）	设置变量值，建议使用$简化函数
bool app.AddTagValue(string tagName, object value）	对变量值执行加减操作
bool app.ToogleTagValue(string tagName）	切换变量值 True 到 False、False 到 True、 0 到 1、 1 到 0
BaseChannel app.GetChannel(string name)	返回变量对象

bool app.WriteFile(string filename, string data)	写文件，文件路径是 Data 目录
string app.ReadFile(string filename)	读 Data 目录下文件
bool app.Wait(string tagname, int intelval)	等待变量的值变为 True，超过等待时间返回 False 时间单位 ms
void app.LogInfo(string source, string message)	记录日志信息
void app.LogWarning(string source, string message)	记录警告日志
void app.LogError(string source, string message)	记录错误日志
DBConnection app.GetDbConnection(string dbType,string strconn)	获取数据库连接
int app.ExecuteNonQuery(DBConnection conn,string sql)	执行 SQL 命令
DBReader app.QueryData(DBConnection conn,string sql)	执行 SQL 查询
string app.GetProjectData(string name)	读取项目配置
string app.GetHtmlData(string name)	读取 HTML 项目配置
string app.SQLQuery(string conntype,string strconnname,string sql)	执行 SQL 查询
int app.SQLExec(string conntype,string strconnname,string sql)	执行 SQL 命令
bool app.StartDevice(string devname)	启动设备
bool app.StopDevice(string devname)	停止设备
BaseChannel [] app.GetDeviceChannels(string devname)	获取设备变量
BaseChannel [] app.GetAllChannels()	获取变量
bool app.ReStartCommuncation(string drvname)	重启驱动
app.SaveTagValues()	保存变量值到文件
app.BatchWrite(string batchName,object value)	批量写入

app.UpdateTagAlarm(string tagName,int alarmType)	更新报警设置 0:不报警,1:模拟量报警,2:On 报警,3:Off 报警,4:数字量变化报警,5:模拟量变量报警
app.SendExtendCommand(string exName, string cmd, string param, string data="")	发送扩展命令
app.Now	返回.Net DateTime 对象
app.DateTimeString	返回当前日期时间字符串
app.DateString	返回当前日期字符串
app.TimeString	返回当前时间字符串
app.ParseDateTime(string t1)	返回.Net DateTime 对象
app.SubDateTime(string t1,string t2)	返回.Net TimeSpan 对象
app.SubDateTimeToSecond(string t1,string t2)	返回时间差的累计秒值
app.SubDateTimeToString(string t1,string t2)	返回 0.00:00:00 格式时间差字符串
app.SubDateTimeToString1(string t1,string t2)	返回 0 天 0 小时 0 分钟 0 秒时间差字符串
app.HistValueQuery(string tags, string startTime, string endTime, int second)	历史归档查询，返回 Json 字符串
app.HistTimerValues(string startTime, string tagNames,string type="")	多变量历史时刻值查询，返回 Json 字符串
app.HistTimerValue(string startTime, string tagName,string type = "")	单变量历史时刻值查询，返回 Json 字符串
app.RowReportQuery(string ruleName, string tags,string startTime,string endTime,int maxCount, int ms = 0, int second = 1)	行表归档查询，返回 Json 字符串
app.ColReportQuery(string ruleName,string tags,string startTime,string endTime,int maxCount,int ms = 0,int second = 1)	列表归档查询，返回 Json 字符串
int app.DownloadRecipe(string recipeName)	下载配方，返回设置的变量个数

t1,t2：日期时间格式如 2023-6-1 12:00:00

二）变量快捷读写函数

$("tagname",value) ：变量读写

例如：$(“tag1”)返回变量值，$(“tag1”,10)设置 tag1 的值为 10

$("tagname")：变量读取

例如：$(“tag1”)返回变量值

$F("tagname")：读取变量的 32 位浮点数值

例如：$F(“tag1”)返回变量的 32 位浮点数值

$I("tagname")：读取变量的 32 位整数值

例如：$I(“tag1”)返回变量的 32 位整数值

$D("tagname")：读取变量的 64 位浮点数值

例如：$D(“tag1”)返回变量的 32 位整数值

$B("tagname")：读取变量的布尔值

例如：$B(“tag1”)返回变量的布尔量值

$S("tagname")：读取变量的字符串值

例如：$S(“tag1”)返回变量的字符串值

add(“tagname”,value)：加减变量值

例如：add(“tag1”,10) 对变量 tag1 的值加 10

$Tag(“tagname”) :得到变量对象

变量操作举例：

```
var tag1 = $Tag("Net1#tag3");//获取变量对象
var k    = 0;
function exec(){
    k++;
    tag1.DoUpdate(k);//更新变量值
}
```

变量对象有很多属性，例如：

tag1.Value 变量值，读写

tag1.Description 变量描述，读写

tag1.Name 变量名称，只读

tag1.Unit 变量值，读写

tag1.RangeMin 量程下限，读写

tag1.RangeMax 量程上限，读写

tag1.Status 变量状态，只读字符串 Good Bad Unknown

用于网络通信的 js 类见附件 12。

JS 演示代码：

MQTT 驱动 JS 代码	var topicTag = $Tag("mqtt_topictag1");
	//每次接收到消息被执行
	function parseMessage(){
	var str = topicTag.StrUserData1;//Message
	var json = JSON.parse(str);
	if (topicTag.StringValue=="/realdata"){
	$Tag("mqtt_tag1").DoUpdate(json.tag1);
	$Tag("mqtt_tag2").DoUpdate(json.tag2);
	$Tag("mqtt_tag3").DoUpdate(json.tag3);
	}
	}
	//用户自发送函数
	function publishMessage(tagname,value){
	mqtt.sendMessage("/writedata",tagname + "=" + value);
	}

模拟驱动 JS 脚本变量	var tag1 = $Tag("simtag1");
	function update(){
	    return tag1.Value + 1;
	}

带参数替换的模拟驱动 JS 脚本变量	var tag1 = $Tag("@tag");
	function update(){
	return tag1.Value + 1;
	}

模拟驱动变量，保存变量实时值	var c = 0;
	function update(){
	app.SaveTagValues();
	c++;
	return c;
	}

模拟驱动变量，设备检查和启停控制	var dev1 = comms.GetDevice("模拟驱动 1");//获取设备
	var count =0;
	function update(){
	if (dev1){

	if (dev1.CheckLastUpdate(60))
	app.LogInfo("System","最后更新:" +
	dev1.LastUpdateTime.ToString(),"",false);
	else
	app.LogInfo("System","没有更新","",false);
	//if (dev1.IsRuning)
	// dev1.Stop();
	// else
	// dev1.Start();
	//Name,Enable
	//UserTag
	//dev1.Stop();
	//dev1.Start();
	//dev1.CheckLastUpdate(120) //120 秒没有数据更新返回 false，否则返回 true
	}
	return count++;
	}

GlobalScript
全局驱动脚本
	//app.LogInfo("Golbal","Start up");
	//驱动管理器启动函数
	function start(){
	//app.LogInfo("Golbal","start function");
	//$Tag("SIMTAG3").DoUpdate(50);
	//$Tag("SIMTAG4").DoUpdate(50);
	}
	//驱动管理器停止函数
	function stop(){
	//app.LogInfo("Golbal","stop function");
	}

模拟驱动变量
SQLServer 数据库查询
	var sql = "select * from report_1";
	/**
	* id,Time
	*/
	function update(){
	var ret="";
	var json =
	app.SQLQuery("SqlServer","userDataSource",sql);//userDataSource 系统配置变量
	//var count = app.SQLExec("SqlServer","userDataSource","delete from ....");
	if (json){
	var datas = JSON.parse(json);
	for (var i=0;i<datas.length;i++){
	if (ret)

	ret += "," + datas[i].id;
	else
	ret = datas[i].id;
	}
	}
	return ret;
	}

模拟驱动变量SQLServer数据库查询	var strconn = "Data Source=.;Initial Catalog=Scada;User ID=iot;Password=iotScada;Connect Timeout=30;";
	var sql = "select * from report_1";
	var conn = app.GetDbConnection("SqlServer",strconn);
	/**
	* id,Time
	*/
	function update(){
	var ret="";
	if (conn){
	var reader = app.QueryData(conn,sql);
	//var count = app.ExecuteNonQuery(conn,"delete from ...");
	if (reader){
	while (reader.Read()){
	if (ret)
	ret += "," + reader.GetInt32(0).toString();
	else
	ret = reader.GetInt32(0).toString();
	}
	reader.Close();
	}
	conn.Close();
	}
	return ret;
	}

模拟驱动变量，MySQL数据库查询	var hasTable = false;
	var tableSQL = "CREATE TABLE `testTable` (`id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,`Time` DATETIME,`TagName` VARCHAR(32) , `Value` FLOAT, PRIMARY KEY (`id`))";
	var checkTableSQL = "SELECT 1=1 FROM testTable LIMIT 1";
	function update(){
	var ret="";
	if (!hasTable){
	var rt= app.SQLQuery("MySQL","mySQL",checkTableSQL);//mySQL 系统配置变量
	if (rt == ""){
	app.SQLExec("MySQL","mySQL",tableSQL); //创建表

```
hasTable = true;
app.LogInfo("Script","MySQL Create Table");
}
else
hasTable = true;
}
else{
var value = $F("SIMTAG1");
var time = app.DateTimeString;
var sql = "Insert into `testTable`(`Time`,`TagName`,`Value`)
VALUES('" + time + "','SIMTAG1',"+value+")";
if (app.SQLExec("MySQL","mySQL",sql)>0)//插入数据
{
app.LogInfo("Script","Inset Data OK","",false);
}
}
return ret;
}
```

附件 2：用于 HTML5 Web 页面的内置 JavaScript 函数（前端使用）

全局属性列表：

名称	描述	说明
isRuning	布尔量，指示 Scada 已经运行	只读
ie	布尔量，指示 IE 浏览器	只读
filename	String 对象，当前文件名称	只读
paramname	String 对象，当前参数文件名称	只读
sys_blink	布尔量，根据设定的动画周期循环变化	只读
dm	数据模型对象和 dataModel 一样	dm.getDataByTag(“tag”)
g2d	2d 图形对象	GraphView 对象
view	2d 图形 DIV 对象	view = g2d.getView();
tagsList	列表对象，指示当前画面需要的标量列表	可以在启动脚本加入变量
valuechangedcallback	全局回调函数	如果指定了该函数 valuechangedcallback=function（tag） 每次接收到数据都会被调用，可用于数据计算或者转换

变量对象：

名称	数据类型	说明
name	字符	变量名称大写
type	字符	数据类型 Int32 Single Double String Boolean 等
value	对象	变量值
status	数值	变量状态 0 UNKNOW 1 GOOD 2 BAD
alarm	数值	报警状态 0 没有报警
desc	字符	变量描述
digcount	数值	小数点个数
unit	字符	单位
time	字符	变量时间
min	数值	量程最小值
max	数值	量程最大值
atype	数值	0-5 报警类型 NotAlarm TimeoutAlarm OnAlarm OffAlarm ChangeAlarm
all	数值	低低报警设置
al	数值	低报警设置
ah	数值	高报警设置
ahh	数值	高高报警设置

hs	数值	是否历史点 1:历史点

本地系统变量:

名称	数据类型	说明
IOERROR	布尔类型	HTTP 通讯故障
PLAYBACKMODE	布尔类型	历史回放模式
NORMALMODE	布尔类型	正常运行模式

本地 js 全局变量:

名称	数据类型	说明
isRuning	布尔类型	运行状态指示
sys_blink	布尔类型	系统变化 ON OFF
username	字符类型	用户名
userlevel	数值类型	用户权限

全局函数列表:

名称	说明
getinputvalue(display, value)	显示输入窗口返回输入值
$T(tagid)	dm.getDataByTag(“tag”)的简写
getTag(tag)	返回当前画面数据库的变量对象
getTagValue(tag)	返回当前画面数据库的变量对象值
addTags(tags)	添加变量数组到页面数据库 举例:addTags([“tag1”,”tag2”])
createTag(tag)	创建页面变量点
addTag(tag)	添加变量到页面数据库 举例: addTag(“tag2”)
questionbox(msg)	询问对话框
display(name, param)	切换画面 举例: display(“file1”); display(“file2”,”p1”);
displayhtml(name, param)	从 HtmlView.html 显示画面 举例: displayhtml(“file1”); displayhtml(“file2”,”p1”);
displayex(name, urlparam)	替换参数:@param1=A&@param2=1 举例: displayex(“file2”,“@p1=a&@p2=sx”);
display_child(child,name, param)	改变子画面显示 举例: Display_child(“c1”,”smallpic”) c1:Tag 属性
display_child_url(child,url)	改变子画面显示 举例: Display_child(“c1”,”status.html”) c1:Tag 属性

editor(filename)	进入组态模式 举例：editor();编辑默认画面或当前画面 editor(filename)；编辑当前运行画面 editor(“file2”)；编辑指定画面
login()	显示登陆页面
logout()	注销用户
viewlog()	显示系统日志
viewstate()	显示系统状态
viewrealtrend(param) openrealtrend(param)	显示实时趋势 param：趋势名称 当参数中带’#’符号时必须使用 encodeURIComponent 函数 viewrealtrend(’ tags=tag1,tags’); viewrealtrend(’ trend=realtrend1&title=历史趋势 2’);
viewhistrend(param) openhistrend(param)	显示历史趋势 param：趋势名称 当参数中带’#’符号时必须使用 encodeURIComponent 函数 openhistrend(’ tags=ioclient.tag1,ioclient.tags’); openhistrend(’ trend=histrend1&title=历史趋势 2’);

writetag(tag, v,callback)	写变量值
	举例：
	writetag(“tag1”,2);
	callback=function(data)
	writetag(“tag2”,2,function(data){
	alert(data.msg);//data.status=0 成功
	});

toogletag(tag,callback)	切换变量值 举例： toggletag (“tag1”); callback=function(data) //data.status=0 成功
addtagvalue(tag, v,callback)	变量值加减操作 举例： addtagvalue(“t1”,1); callback=function(data)//data.status=0 成功
setTagValue(data)	当对象有 tagname 属性时,调用设置数据显示窗口
setTagForm(disp, tagname, v)	显示变量设置窗口
writetagvalues(vs, callback)	vs: key:value 值对 {“tag1”:1,”tag2”:”name”} 批量写值
reload(callback)	设计时,重新加载数据库变量 callback=function(data) var id = data.id;// : 0 OK else error

	var m = data.msg;
confirm(msg)	询问对话框 msg:信息内容 确定返回 true
showMessage(title, msg)	显示信息
getInput(title, txt, action)	获取输入值对话框 action=function(data)
showTagInfo(tagname)	显示变量信息窗口
showPanelInfo(title, msg, w, h, style)	显示浮动 jquery 信息框 title:标题 msg:html 信息 w:宽度 h:高度 style:样式 primary,default,info,success,warning,danger
popupView(title, w, h, view)	弹出子画面 title:标题 w, h: 宽度和高度 view: 图形画面名称 举例: popupView(“#1 泵”,200,250,”pump1”)
viewCurrentTags	运行时显示当前画面变量数据列表
stopScada	停止运行当前画面
startScada	启动运行当前画面
stopAll	停止当前画面全部窗口
startAll	启动当前画面全部窗口
startplayback	从历史数据启动当前画面的回放
stopplayback	停止历史回放
getTagsValue(tags)	同步读取变量值 返回变量数组对象 tags:变量名称 d1.zt.a1,d1.zt.a2
getAllFileNames()	同步读取画面列表 返回值字符串数组
jsonfileExist(filename)	同步判断画面名称是否存在 返回值 true false
linkbutton_Click(data)	当对象有 filename 属性时，该函数执行画面跳转操作
getHistValues(tags, starttime, endtime, second, callback, type)	tags:逗号分割的变量名全名称 starttime 开始时间 endtime 结束时间 second 间隔秒 callback 回调函数 type:用户参数 callback 回调函数 function(data)
updatevalues(vs, callback)	vs: key:value 值对 {“tag1”:1,”tag2”:”name”}
changePassword(oldpwd,newpwd)	修改当前用户的密码 成功返回 json {“msg”:”ok”}

changePasswordDlg()	调用修改当前用户密码对话框
addLog(msg,type, severity)	添加日志记录 msg:记录信息 type:log oplog severity:0 information 1 warning 2 error
dologin(user)	调用登陆对话框 user:显示的用户名 成功返回:true
dologout	用户注销
reloadview()	重新加载页面
restart()	重启 scada.js
layerView(title,filename,param,width,height,offsetx,offsety)	浮动弹出画面
layerAlert(message)	浮动消息
layerConfirm(message,okcallback,cancelcallback)	浮动询问
writeValue(title, tagName,desc)	写变量对话框
writeUserValue(title, tagValue, desc,callback)	用户写入对话框
updateSingleAlarmSetting(tag, type,value, callback)	type:h hh l ll 模拟量报警定值设置
saveTags()	保存驱动修改
sqlQuery(dbtype,strconn,sql,callback)	dbtype:sqlserver mysql postgresql strconn:项目配置的参数名称，数据库连接字符 sql:查询 SQL 内容 返回 JSON 对象
sqlExec(dbtype,strconn,sql,callback)	dbtype:sqlserver mysql postgresql strconn:项目配置的参数名称，数据库连接字符 sql:执行 SQL 内容 返回 JSON 对象
batchWriteName(batchName,value)	写入批量 batchName:批量名称 value:值
updateJsonFileFromRemote()	从远程更新画面文件
checkUser(userName,password,callback)	验证用户名和密码
doCheckUser(user, callback)	用户验证对话框
writeValueEx(title, tagName, desc)	带用户验证的写入变量对话框
recipeView(recipeName)	显示配方编辑对话框
downloadRecipe(recipeName, callback)	下载配方
htmlview(name, param)	显示 Html 页面 举例：htmlview(“file1”);

	htmlview(“file2”,”p1”);
getQueryString(name)	读取 url 查询参数

附件 3：表达式计算

IOTGateway 提供了在线计算功能（网络接收驱动和 HTTP 接收驱动不支持），支持驱动采集变量的计算，也支持内部变量的计算，驱动内的内部变量在扫描采集成功完成后进行计算，支持数学，布尔，函数，字符串计算。

三角函数：Abs,Acos,Asin,Atan,Cos,Sign,Sin,Tan

数学函数：

Ceiling,Exp,Floor,IEEERemainder,Log,Log10,Pow,Round,Sqrt,Truncate

三目函数：(布尔表达式) ？ x ： y 如果条件为真返回 x，否则返回 y

条件判断：

IF((布尔表达式),x,y) 如果条件满足返回 x，否则返回 y

in(x,x1,x2,x3...) 如果 x 在后面的列表中，返回 true，否则返回 false

运算符号

数学运算符：+,-,*,/, % ,^

逻辑运算符：AND, OR, NOT, &&, ||, ~

比较运算符：!=,<>,==,>,>=,<,<=

位运算符：&,|,~,>>,<<

字符串使用单引号包括，如‘year’

组态变量使用综括号包括，如 [second]

变量当前值：val

字符串相加时第 1 个对象必须是字符类型：

‘111’ + ‘2’

str([simtag1]) + ‘AAA’

组态函数

settag(变量名称，变量值，成功返回值，失败返回值)

settag(‘tag1’,100,0,-1) ---- 设置 Tag1 的值为 100，成功返回 0，失败返回-1

tagstatus(变量名称,失败返回值)

例：tagstatus(‘tag2’,-1) ---读取 tag2 的状态,成功返回变量状态(0:UnKnow，1：Good，2：Bad），失败返回-1

tagvalue(变量名称,失败返回值)

例：tagvalue(‘tag2’,0) ---读取 tag2 的状态，成功返回变量值，失败返回 0

tagdesc，参数同上，读取变量描述

tagunit，参数同上，读取变量单位

tagalarmstatus，参数同上，读取变量报警状态

str(value),值转换为字符

int2bcd(value) 值转换为 BCD

bit(tagvalue,bitvalue) 32 位变量或者值取位，返回 true 或 false

bit64(tagvalue,bitvalue) 64 位变量或者值取位，返回 true 或 false

例：bit([second],2)

bit(8,2)

在变量配置界面的表达式运行时可以在线修改，如果变量的扫描周期为 0，驱动每次成功采集后都会对相关变量进行表达式计算,如果不为 0 则大于扫描时间才会执行计算,不管扫描周期为多少,采集失败的情况下不会执行表达式计算。

举例：

1）把字变量拆分位变量

R1，为 1 个 16 位的 Int 类型

创建 1 个布尔类型变量，设置计算表达式为 [R1] & 1 得到 R1 的第 1 位的值

第 2 位 ([R1] & 2)=2 返回 true 或者 false

第 3 位 ([R1] & 4)=4 返回 true 或者 false

第 4 位 ([R1] & 8)=8 返回 true 或者 false

仅能用于 16 位变量，32 位和 64 位该方法会失败

2）简单计算

R1，R2 为驱动采集变量

求和：新建 1 个 R3 内部变量，输入表达式 [R1]+[R2]

运算: [R1] * 10 + 0.5 - 1

[R1] > 5 ? true : false

val > 6 ? true : false

求模：[R1] % 3

判断和比较：([R1] > 1) && ([R2]>3)

当前值开平方：Sqrt(val)

3）可变定值报警

使用 1 个内部变量设定位报警定值，如 alHi

创建 1 个数字变量，使用表达式判断输出，在该变量上设置报警,变量上设置如下表达式

[aTag] > [alHi]

aTag：模拟量采集变量

alHi：报警定值

该表达式计算在 aTag 的值大于 alHi 是输出 True，否则输出 False，这样就实现了可变定值报警功能。

4）变量状态判断

创建 1 个内部整数类型变量，设置表达式 tagstatus(‘TagName’，0)

TagName：变量名称

返回值：1：Good

附件 4：数据库连接字符串格式

1）SQLServer

Data Source=192.168.10.33;Initial Catalog=IOTGateway;User ID=iot;Password=iotScada;Connect Timeout=30;

2）MySQL

server=192.168.10.33;uid=root;pwd=iotScada;database=iotgateway;Connect Timeout=30;charset=utf8;

3）达梦数据库（DM）

Server=192.168.10.33;UserId=SYSDBA;PWD=SYSDBA;DATABASE=SCADA;

4）PostgreSQL 数据库

Host=192.168.10.17;Port=5432;Username=postgres;Password=postgres;Database=IOTGateway;encoding=UTF8;

5）openGauss 数据库

Host=192.168.10.17;Port=7654;Username=postgres;Password=postgres;Database=IOTGateway;encoding=UTF8;No Reset On Close=true;

6）SQLite

DataSource= /root/user/xxx.db　　指定存储目录

DataSource={project}/report.db　　软件 Data 目录下

附件 5：Docker 使用说明

1）Linux 版本带 docker 构建文件，运行配置正常后，可以构建 docker

构建：sudo docker build -t iotgateway:v1.0 .

2）从 tar 文件恢复到 Docker 中执行如下命令

sudo docker load<iotgateway.tar

3）运行系统执行如下命令

测试运行模式：sudo docker run -it -d -p 8080:8080 --restart always --name iotgateway iotgateway:v1.0

加密狗模式：sudo docker run -it --privileged -d -p 8080:8080 --restart always --name iotgateway --device=/dev/usb/hiddev0 iotgateway:v1.0

本机 IP 模式：sudo docker run -it -d --network host --restart always --name iotgateway iotgateway:v1.0

--network host 使用本机 IP 地址，不需要使用-p 8080:8080 进行端口映射,当容器无法访问外部网络的情况，使用本机 IP 地址配置。

4）Docker 中运行不支持软件授权，仅支持 USB 加密狗授权，因此需要使用特权命令映像 usb 设备到容器，运行前需要把 USB 加密狗插入计算机上，然后查看/dev/usb 下是否存在 hiddev0 文件。

--device=/dev/usb/hiddev0

附件 7：InfluxDB 安装

Linux 软件包下载地址：

rpm 格式 x86：

http://repos.influxdata.com/stable/

rpm 格式 Arm64：

http://repos.influxdata.com/stable/aarch64/main/

deb 格式 x86：

https://repos.influxdata.com/debian/packages/

deb 格式 Arm64：

https://download.docker.com/linux/ubuntu/dists/xenial/pool/stable/arm64/

下载软件包后手动安装命令如下：

1）rpm 包　sudo rpm -ivh influxdb_2.7.5.rpm

2）deb 包　sudo dpkg -i influxdb_2.7.5.deb　或

　　sudo apt install ./influxdb_2.7.5.deb

卸载命令：卸载前先停止运行（sudo systemctl stop influxdb）

1）rpm 包　sudo rpm -e influxdb

2）deb 包　sudo apt-get remove --purge influxdb 或

　　sudo dpkg -r -P influxdb

安装完成根据需要修改配置文件：/etc/influxdb/config.toml，可以设置数据存储路径，根据需要修改配置参数；

安装完成后 influxdb 是 1 个服务，手动启动方式为：systemctl start influxdb

启动后就可以使用浏览器访问 http://localhost:8086；

JSON 格式的标准配置文件格式如下（config.json）：

```
{
        "assets-path": "",
        "bolt-path": "influxd.bolt",
        "e2e-testing": false,
        "engine-path": "engine",
```

"feature-flags": null,
"flux-log-enabled": false,
"hardening-enabled": false,
"http-bind-address": ":8086",
"http-idle-timeout": 180000000000,
"http-read-header-timeout": 10000000000,
"http-read-timeout": 0,
"http-write-timeout": 0,
"influxql-max-select-buckets": 0,
"influxql-max-select-point": 0,
"influxql-max-select-series": 0,
"instance-id": "",
"log-level": "info",
"metrics-disabled": false,
"nats-max-payload-bytes": 0,
"nats-port": 0,
"no-tasks": false,
"pprof-disabled": false,
"query-concurrency": 1024,
"query-initial-memory-bytes": 0,
"query-max-memory-bytes": 0,
"query-memory-bytes": 0,
"query-queue-size": 1024,
"reporting-disabled": false,
"secret-store": "bolt",
"session-length": 525600,
"session-renew-disabled": false,
"sqlite-path": "influxd.sqlite",
"storage-cache-max-memory-size": 1073741824,

"storage-cache-snapshot-memory-size": 26214400,
"storage-cache-snapshot-write-cold-duration": "10m0s",
"storage-compact-full-write-cold-duration": "4h0m0s",
"storage-compact-throughput-burst": 50331648,
"storage-max-concurrent-compactions": 0,
"storage-max-index-log-file-size": 1048576,
"storage-no-validate-field-size": false,
"storage-retention-check-interval": "30m0s",
"storage-series-file-max-concurrent-snapshot-compactions": 0,
"storage-series-id-set-cache-size": 0,
"storage-shard-precreator-advance-period": "30m0s",
"storage-shard-precreator-check-interval": "10m0s",
"storage-tsm-use-madv-willneed": false,
"storage-validate-keys": false,
"storage-wal-fsync-delay": "0s",
"storage-wal-max-concurrent-writes": 0,
"storage-wal-max-write-delay": 600000000000,
"storage-write-timeout": 10000000000,
"store": "disk",
"testing-always-allow-setup": false,
"tls-cert": "",
"tls-key": "",
"tls-min-version": "1.2",
"tls-strict-ciphers": false,
"tracing-type": "",
"ui-disabled": false,
"vault-addr": "",
"vault-cacert": "",
"vault-capath": "",

```
        "vault-client-cert": "",
        "vault-client-key": "",
        "vault-client-timeout": 0,
        "vault-max-retries": 0,
        "vault-skip-verify": false,
        "vault-tls-server-name": "",
        "vault-token": ""
}
```

查看授权信息：http://localhost:8086/api/v2/authorizations

API 文档：https://docs.influxdata.com/influxdb/v2/api/

Windows 版本把这个配置文件放到 influxdb 软件目录下就可以，启动后会在软件目录下建立 engine 目录，否则归档目录会设置到用户目录下。

engine-path：数据存储目录

http-bind-address：http 服务端口

session-length：Web 管理界面会话超时设置（秒）

附件 8：华为 openGauss 数据库

openGauss 数据库是一个国产开源免费数据库系统，兼容 PostgreSQL 数据库，openEuler22.03 安装时选择服务器模式，包括 openGuass 数据库

sudo firewall-cmd --list-ports　列出防火墙开放端口

sudo firewall-cmd --permanent --add-port=7654/tcp　添加防火墙端口

sudo firewall-cmd --reload

检查 OpenGauss 是否安装

```
ps ux | grep gauss
```

设置数据库

配置 `postgresql.conf` 和 `pg_hba.conf`,2 个文件,都在 `opengauss` 用户的 home 目录中

首先登入 `opengauss` 用户

```
su - opengausscd ~/data

vi postgresql.conf
```

修改内容：

```
listen_address= '*'

local_bind_address = '0.0.0.0'

password_encryption_type = 1
```

注意取消掉前面的注释符号#

然后配置

```
vi pg_hba.conf
```

在最后加一条：

```
host all all 0.0.0.0/0 sha256
```

重启系统

reboot

进入 gsql 交互式环境

```
gsql -d postgres -p 7654 -r
```

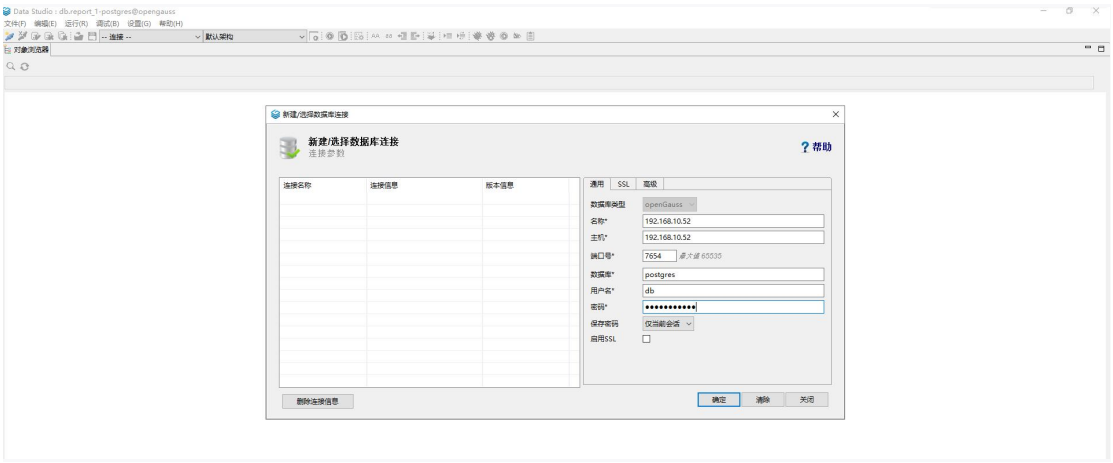
重置超级管理员用户 opengauss 的密码

```
ALTER ROLE opengauss PASSWORD 'Test@123';
```

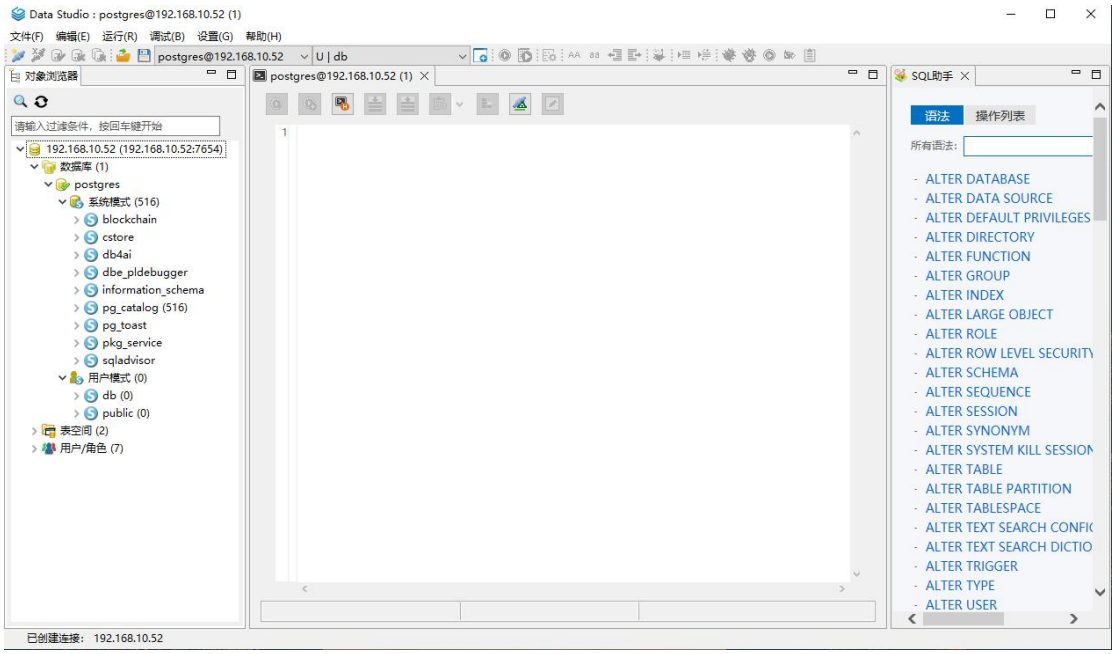
创建用于连接的用户

```
CREATE USER db WITH PASSWORD 'Test@123';

ALTER USER db sysadmin;
```

之后就可以使用创建的用户通过管理工具（Data Studio）连接数据了。



连接的数据库为 postgres，端口为 7654，不勾选 SSL



之后 openGauss 数据库就可以当做 PostgreSQL 数据库使用

Host=192.168.10.52;Port=7654;Username=db;Password=xxx;Database=scada;encoding=UTF8;No Reset On Close=true;

红色内容在连接字符串必须加入，这个区别于 PostgreSQL 数据库。

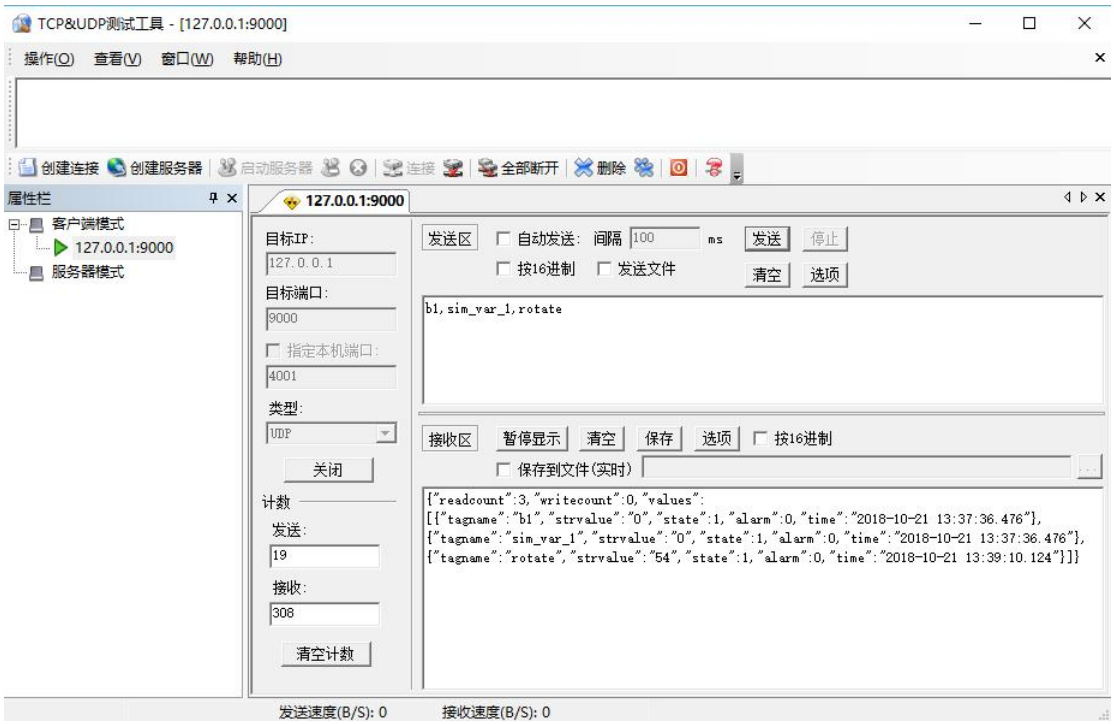
备份数据库命令：gs_dump -f back.tar -p 7654 dbName -F -t

恢复数据库命令：gs_restore back.tar -p 7654 -d dbName

\q 退出 psql 命令行

附件 9：UDPService 服务

该服务提供了简单高效的数据查询和设置功能，该服务不具备权限认证，建议本机使用。通讯数据全部为文本格式,UTF-8 编码，系统运行后通过往该端口发送 UDP 数据包可以查询和设置变量。



直接发送变量名称,多个变量名称直接使用逗号分割

设置变量方式变量名称=值，查询和读取可以同时发送，例如：发送 tag1=2,tag1,tag2

由于 UDP 数据包尺寸有限制为 64K,因此需要限制发送和查询的数量。

附件 10：欧拉系统 MySQL 离线安装

安装包下载地址 https://zhuanlan.zhihu.com/p/649349766

1、关闭 selinux 设置，关闭防火墙

vim /etc/sysconfig/selinux

重启 reboot

2、添加用户

```
groupadd mysql
useradd -g mysql mysql
passwd mysql  #设置mysql用户密码
```

编辑 /etc/sudoers 文件

修改 vi /etc/sudoers 为可写入

vi /etc/sudoers

```
# env_reset is disabled or HOME is present in the env_keep list.
#
Defaults    always_set_home
Defaults    match_group_by_gid

# Prior to version 1.8.15, groups listed in sudoers that were not
# found in the system group database were passed to the group
# plugin, if any. Starting with 1.8.15, only groups of the form
# %:group are resolved via the group plugin by default.
# We enable always_query_group_plugin to restore old behavior.
# Disable this option for new behavior.
Defaults    always_query_group_plugin

Defaults    env_reset
Defaults    env_keep =  "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults   env_keep += "HOME"

Defaults    secure_path = /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
mysql   ALL=(ALL)       ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL
```

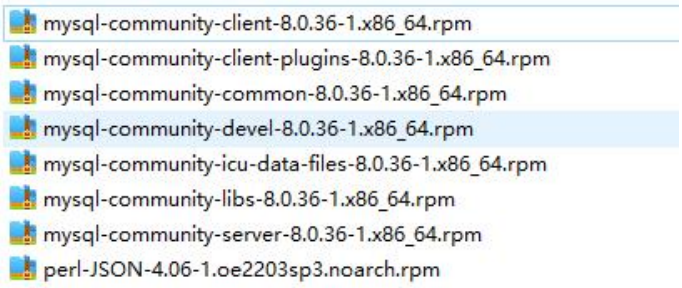
加入

mysql ALL=(ALL)   ALL

保存后修改 vi /etc/sudoers 为只读

3、拷贝文件开始安装

建立 1 个目录，拷贝 rpm 文件包到目录下



su - mysql 切换到 mysql 用户

使用 mysql 用户安装 rmp 包

sudo rpm -ivh perl-JSON-4.06-1.oe2203sp3.noarch.rpm

sudo rpm -ivh mysql-community-client-plugins-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-common-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-libs-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-client-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-devel-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-icu-data-files-8.0.36-1.x86_64.rpm

sudo rpm -ivh mysql-community-server-8.0.36-1.x86_64.rpm

完成安装后，切换到 root 用户，修改启动文件

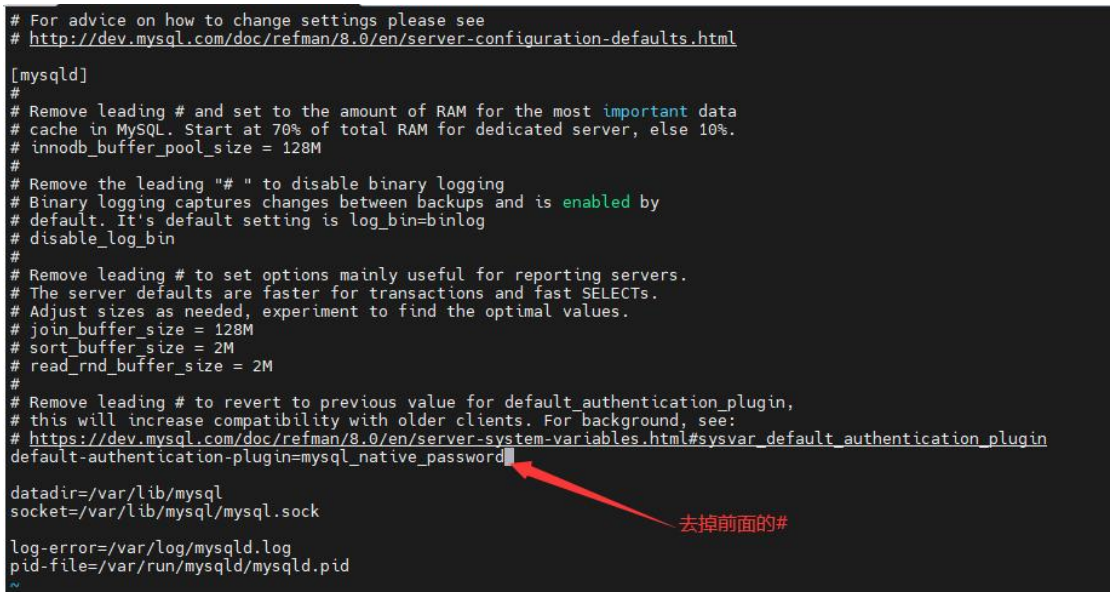
vi /etc/rc.d/init.d/mysqld (添加 2 行)

```
    install -d -m0751 -omysql -gmysql "$datadir" || exit 1
      fi
      if [ ! -h "$datadir" -a "x$(dirname "$datadir")" = "x/var/lib" ]; then
    chown mysql:mysql "$datadir"
    chmod 0751 "$datadir"
      fi
      if [ -x /sbin/restorecon ]; then
    /sbin/restorecon "$datadir"
    for dir in /var/lib/mysql-files /var/lib/mysql-keyring ; do
        if [ -x /usr/sbin/semanage -a -d /var/lib/mysql -a -d $dir ] ; then
      /usr/sbin/semanage fcontext -a -e /var/lib/mysql $dir >/dev/null 2>&1
      /sbin/restorecon -r $dir
        fi
    done
      fi
      # Now create the database
      initfile="$(install_validate_password_sql_file)"
      action $"Initializing MySQL database: " /usr/sbin/mysqld --initialize --datadir="$datadir" --user=mysql
      ret=$?
      rm -f "$initfile"
      [ $ret -ne 0 ] && return $ret
      # Generate certs if needed
      if [ -x /usr/bin/mysql_ssl_rsa_setup -a ! -e "${datadir}/server-key.pem" ] ; then
    /usr/bin/mysql_ssl_rsa_setup --datadir="$datadir" --uid=mysql >/dev/null 2>&1
      fi
  fi
  if [ ! -h "$datadir" -a "x$(dirname "$datadir")" = "x/var/lib" ]; then
      chown mysql:mysql "$datadir"
      chmod 0751 "$datadir"
  fi

  mkdir -p /run/mysqld
  chown -R mysql:mysql /run/mysqld

  # Pass all the options determined above, to ensure consistent behavior.
  # In many cases mysqld_safe would arrive at the same conclusions anyway
  # but we need to be sure.  (An exception is that we don't force the
  # log-error setting, since this script doesn't really depend on that,
  # and some users might prefer to configure logging to syslog.)
  # Note: set --basedir to prevent probes that might trigger SELinux
  # alarms, per bug #547485
  $exec $MYSQLD_OPTS --datadir="$datadir" --socket="$socketfile" \
    --pid-file="$mypidfile" \
    --basedir=/usr --user=mysql $extra_opts >/dev/null &
  safe_pid=$!
```

更新系统配置

systemctl daemon-reload

修改 /etc/my.cnf 配置文件




```
# For advice on how to change settings please see
# http://dev.mysql.com/doc/refman/8.0/en/server-configuration-defaults.html

[mysqld]
#
# Remove leading # and set to the amount of RAM for the most important data
# cache in MySQL. Start at 70% of total RAM for dedicated server, else 10%.
# innodb_buffer_pool_size = 128M
#
# Remove the leading "# " to disable binary logging
# Binary logging captures changes between backups and is enabled by
# default. It's default setting is log_bin=binlog
# disable_log_bin
#
# Remove leading # to set options mainly useful for reporting servers.
# The server defaults are faster for transactions and fast SELECTs.
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
#
# Remove leading # to revert to previous value for default_authentication_plugin,
# this will increase compatibility with older clients. For background, see:
# https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_default_authentication_plugin
default-authentication-plugin=mysql_native_password

datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

启用防火墙，添加 3306 端口

```
firewall-cmd --zone=public --add-port=3306/tcp --permanent
firewall-cmd --reload
```

启动 mysql

```
systemctl start mysqld
```

查看默认密码：

grep 'temporary password' /var/log/mysqld.log

记录下密码 zoa>q/7uafgI（每次不一样）

登录 mysql

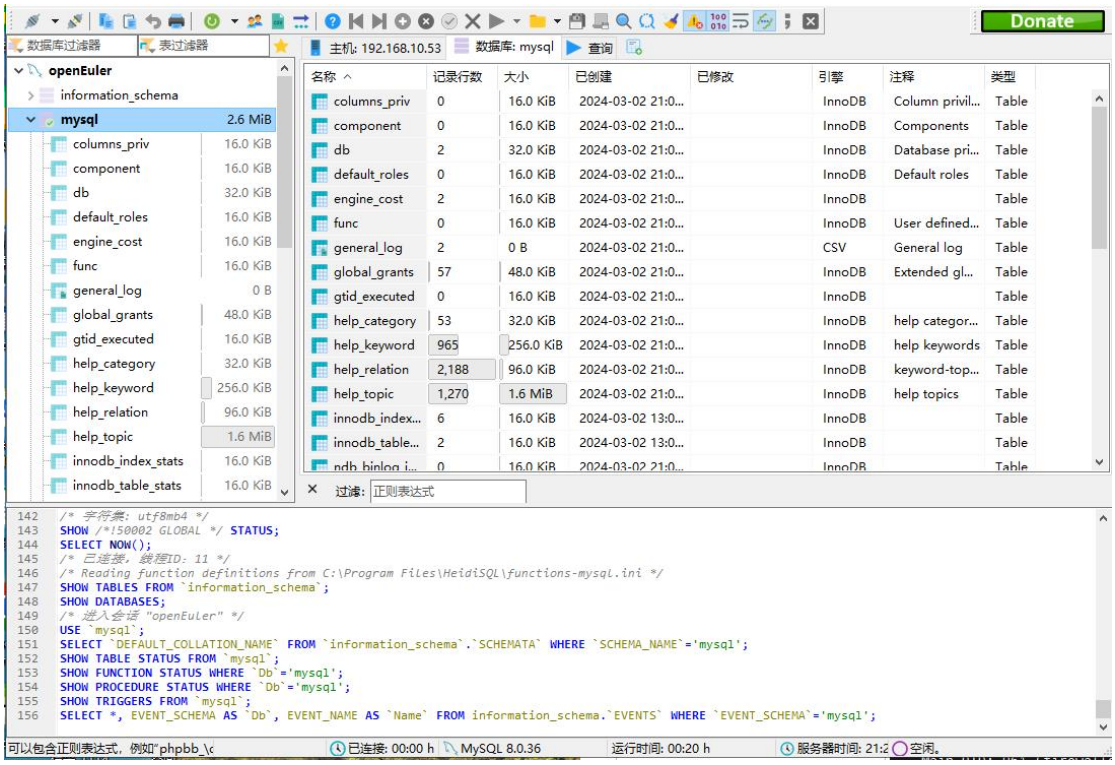
mysql -u root -p

输入密码，成功后执行下列语句（MySQL@123 密码必须包含大小写和数字以及特殊字符）

mysql>

```
alter user 'root'@'localhost' identified WITH mysql_native_password by 'MySQL@123'
password expire never;
flush privileges;
use mysql;
update user set host = '%' where user = 'root' and host='localhost';
flush privileges;
alter user 'root'@'%' identified WITH mysql_native_password by 'MySQL@123' password
expire never;
flush privileges;
```

现在可以远程登录到数据库了



MySQL 8 版本开始用户密码默认加密方式已经改变，如果密码不是“`mysql_native_password`”方式，没有管理员权限可以修改的话可以尝试在连接字符串加入

SslMode=none;AllowPublicKeyRetrieval=True

附件 11：扩展 C# DLL

目录下 RunTime.dll 是 1 个扩展 DLL 文件，可以使用 VS 修改编译。

名称	修改日期	类型	大小
bin	2024-04-16 18:31	文件夹	
obj	2024-04-16 18:34	文件夹	
Functions.cs	2024-04-15 19:17	C# Source File	1 KB
Global.cs	2024-04-15 19:20	C# Source File	3 KB
RunTime.csproj	2024-04-13 19:39	C# Project File	1 KB

项目是 1 个.net core 6 类库，如果需要引用第三方类库，先到 IOTGateway 目录下看下是否有，有的话直接添加引用，需要防止出现版本冲突。

Global.cs 文件的 3 个预定义函数分别用于项目启动前执行，启动后执行，停止前执行。

Global.cs

```
using IOTGateway.Common;
using IOTGateway.Common.Helper;
using IOTGateway.Common.Interfaces;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Common.Helper;

namespace RunTime
{
    public class Global
    {
        /// <summary>
        /// 静态函数
        /// </summary>
        static Global()
        {
        }

        /// <summary>
        /// 项目加载函数
        /// 启动运行前执行
        /// </summary>
        public static void OnProjectLoad()
        {
            LogInfo("RunTime", "OnProjectLoad");
        }

        /// <summary>
        /// 项目启动函数
        /// 已经启动运行
        /// </summary>
        public static void OnProjectStart()
        {
            LogInfo("RunTime", "OnProjectStart");
        }

        /// <summary>
        /// 项目停止函数
        /// 即将停止运行前执行
        /// </summary>
        public static void OnProjectStop()
        {
            LogInfo("RunTime", "OnProjectStop");
        }
```

Functions.cs 定义了 2 个参考模拟驱动执行函数，被模拟驱动执行的函数必须是静态函数。

如果在 RunTime 库中创建了对象，分配了内存，建议在项目停止函数中释放，否则可能导致内存泄漏。

Functions.cs

```
using IOTGateway.Common;
using IOTGateway.Common.Interfaces;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace RunTime
{
    public class Functions
    {
        /// <summary>
        /// RunTime.Functions.SelfAdd
        /// </summary>
        /// <param name="ch"></param>
        /// <returns></returns>
        public static double SelfAdd(BaseChannel ch)
        {
            return ch.DoubleValue+1;
        }

        /// <summary>
        /// RunTime.Functions.IntAdd
        /// </summary>
        /// <param name="ch"></param>
        /// <returns></returns>
        public static Int32 IntAdd(BaseChannel ch)
        {
            return ch.Int32Value + 1;
        }
    }
}
```

通过 C#添加 JS 访问类和对象的方法

1）创建 1 个 C#类

```
/// <summary>
/// js调用C#演示类
/// </summary>
2 个引用
public class JsClass
{
    private int count = 0;

    /// <summary>
    ///
    /// </summary>
    /// <returns></returns>
    0 个引用
    public string Now()
    {
        return DateTime.Now.ToString("yyyy-MM-dd HH:mm:ss.fff");
    }

    /// <summary>
    ///
    /// </summary>
    0 个引用
    public int Count
    {
        get
        {
            count++;
            return count;
        }
    }
}
```

2）在加载函数添加代码

myObject 对象可以直接在 js 中使用，通过 myObject 访问类的功能

myClass 类可以在 js 中被创建对象

```
/// <summary>
/// 项目加载函数
/// 启动运行前执行
/// </summary>
0 个引用
public static void OnProjectLoad()
{
    LogInfo("RunTime", "OnProjectLoad");

    //添加对象给JS访问
    if (Env.Current!=null)
        Env.Current.AddHostObject("myObject", new JsClass());

    //添加类给JS访问
    if (Env.Current != null)
        Env.Current.AddHostType("myClass",typeof(JsClass));

}
```

3）js 访问

myObject.Now(),通过对象访问方式

//创建实例后访问的方式

var obj = new myClass();

obj.Count;

附件 12：用于 JS 的网络通讯类

js 类可用于模拟驱动的 js 函数,定时调度的 js 函数以及驱动更新的 Js 函数中。

TCP 客户端　模拟驱动 JS 变量演示

```
var c= 0;
var tcp = new tcpClient(jsengine);
function onConnected(){
  app.LogInfo("tcpClient", "onConnected","",false);
}
function onDisConnected(){
  app.LogInfo("tcpClient", "onDisconnected","",false);
}

function onError(msg){
```

```
  app.LogInfo("tcpClient", "onError:" + msg,"",false);
}
function onRead(len,strData){
  app.LogInfo("tcpClient", "onRead:" + len + "," + strData,"",false);
}
tcp.encoding=0;
tcp.binaryMode = false;
tcp.connect("127.0.0.1",5020);
function update(){ //模拟驱动更新函数
    if (tcp.connected){
      tcp.writeString("1234567890");
    }
    else {
      tcp.connect("127.0.0.1",5020);
    }
    return c++;
}
function onDispose(){//用于自动释放对象
    tcp.close();
}
```

UDP 客户端 模拟驱动 JS 变量演示

```
var c = 0;
var udp = new udpClient(jsengine);
function onError(msg){
  app.LogInfo("udpClient", "onError:" + msg,"",false);
}
function onRead(len,strData){
  app.LogInfo("udpClient", "onRead:" + len + "," + strData,"",false);
}
udp.encoding=0;
udp.binaryMode = false;
function update(){ //模拟驱动更新函数
  udp.writeString("1234567890","127.0.0.1",5021);
  return c++;
}
function onDispose(){//用于自动释放对象
  udp.close();
}
```

WebSocket 客户端 模拟驱动 JS 变量演示

```
var c= 0;
var websocket = new websocketClient(jsengine);
function onConnected(){
  app.LogInfo("websocket", "onConnected","",false);
```

```
  var obj = {
  "cmdName":"subscribe",
  "tags":["Second","Minute"]
  };
  websocket.writeString(JSON.stringify(obj));
}
function onDisConnected(){
  app.LogInfo("websocket", "onDisconnected","",false);
}
function onError(msg){
  app.LogInfo("websocket", "onError:" + msg,"",false);
}
function onRead(len,strData){
  app.LogInfo("websocket", "onRead:" + strData,"",false);
}
//tcp.encoding = 1;
//tcp.binaryMode = false;
websocket.connect("ws://fscada.net:8080/wstag?token=B579C28B-6A7A0FC8",5000);
function update(){ //模拟驱动更新函数
  if (websocket.connected) {
  var obj = {
  "cmdName":"readvalues",
  "tags":["Second","Minute"]
  };
  websocket.writeString(JSON.stringify(obj));
  }
  else {
    websocket.connect("ws://fscada.net:8080/wstag?token=B579AB724BDD0FC8",5000);
  }
  return c++;
}
function onDispose(){//用于自动释放对象
  if (websocket.connected)
    websocket.close();
}
```

RestHttp 接口

```
var restHttp = new rest("http://fscada.net:8080");
var res = "";
var c= 0;
//restHttp.addHeader("Content-Type","text/plain;charset=utf-8");
//restHttp.addHeader("Content-Type","application/json");
//restHttp.clearHeaders();
//restHttp.LastError;
function update(){ //模拟驱动更新函数
```

```
    res = restHttp.getRequest("/rest/alltags");
    restHttp.addParam("tagName", "SIMTAG3");
    restHttp.addParam("value",restCount);
    res = restHttp.postForm("/rest/writetagvalue?token=B579C28BD0FC8");
    restHttp.clearParams();
    //res = restHttp.postRequest("/json/xx","jsonString");
    return c++;
}
```

MQTT 客户端 模拟驱动 JS 变量演示

```
var c= 0;
var mqtt = new
mqttClient(jsengine,"127.0.0.1","iot","admin","admin",1883,30,true,5000);
//iot mqtt 客户端 id  true 保存消息  1883 端口   30 秒心跳时间    5000 超时 5 秒
mqtt.retain = true;
mqtt.qos = 1;//0:AtMostOnce 1:AtLeastOnce 2:ExactlyOnce 3:OnlyTransfer
mqtt.encoding = 1;//0 : Ascii 1 : UTF8 2 : Default 3 : Unicode
function onError(){
  app.LogWarning("mqttClient", "onError","",false);
}
function onRead(topic,message){
  app.LogInfo("mqttClient", "onRead:" + topic + "," + message,"",false);
}
function update(){ //模拟驱动更新函数
  if (mqtt.connected){
    mqtt.publicMessage("topic","1234567890");
  }
  else {
    if (mqtt.connect()){
      mqtt.subscribeMessage("Topic1,Topic2");
      app.LogInfo("mqttClient", "连接成功","",false);
    }
    else{
      app.LogWarning("mqttClient", "连接失败","",false);
    }
  }
  return c++;
}
function onDispose(){//用于自动释放对象
  mqtt.close();
}
```

附件 13：WebAPI 扩展

IOTGateway 支持 dotnet core 6 编写 WebAPI 扩展函数。

输出 ExtendWebApi_xxx.dll 格式的包括 WebAPI 的 dll 可以被正常加载。

```
using Microsoft.AspNetCore.Mvc;
using System.Text;
using Newtonsoft.Json;
using IOTGateway.Common;
using IOTGateway.Common.Helper;
using Microsoft.Extensions.Logging;
using IOTGateway.Json;

namespace ExtendWebApi
{
    /// <summary>
    /// 测试扩展API接口
    /// </summary>
    [Route("/api/extend")]
    [ApiController()]
    3 个引用
    public class DatabaseController : ControllerBase
    {
        #region

        /// <summary>
        ///
        /// </summary>
        /// <param name="logger"></param>
        0 个引用
        public DatabaseController(ILogger<DatabaseController> logger) ...

        /// <summary>
        ///
        /// </summary>
        /// <param name="page"></param>
        /// <param name="rows"></param>
        /// <returns></returns>
        [Route("/api/extend/read")]
        [HttpPost]
        0 个引用
        public ResultRows ReadData([FromForm] int page=1, [FromForm] int rows=30)...
```

建议使用/api/extend/xxx 格式路由。

附件 14：二维码 QRCode 动态生成


```
var qrnode = dm.getDataByTag("qrcode");//qrcode 为页面对象的名称
if (qrnode){
     $.post("api/scada/makeqrcode",{"code":"1234567890" ,"width":40},function(ret){
            if (ret.ret==0 && ret.msg){//获取二维码 base64 编码信息
                qrnode.setImage(ret.msg);//更新对象显示
            }
       });
}
```

按钮的点击事件 Js 代码如上，生成的二维码是 SVG 矢量图形格式。

scrpits 目录下也包含了 qrcode.min.js 文件，也可以通过编写 JS 脚本在前端生成二维码。

前端 JavaScript 代码生成二维码的方法：

```
var qrcode = new QRCode(document.getElementById("qrcode"), {
    text: "https://www.example.com/", // 要编码的文本
    width: 128, // 二维码宽度
    height: 128, // 二维码高度
    colorDark : "#000000", // 前景色
    colorLight : "#ffffff", // 背景色
    correctLevel : QRCode.CorrectLevel.H // 容错级别
});
```