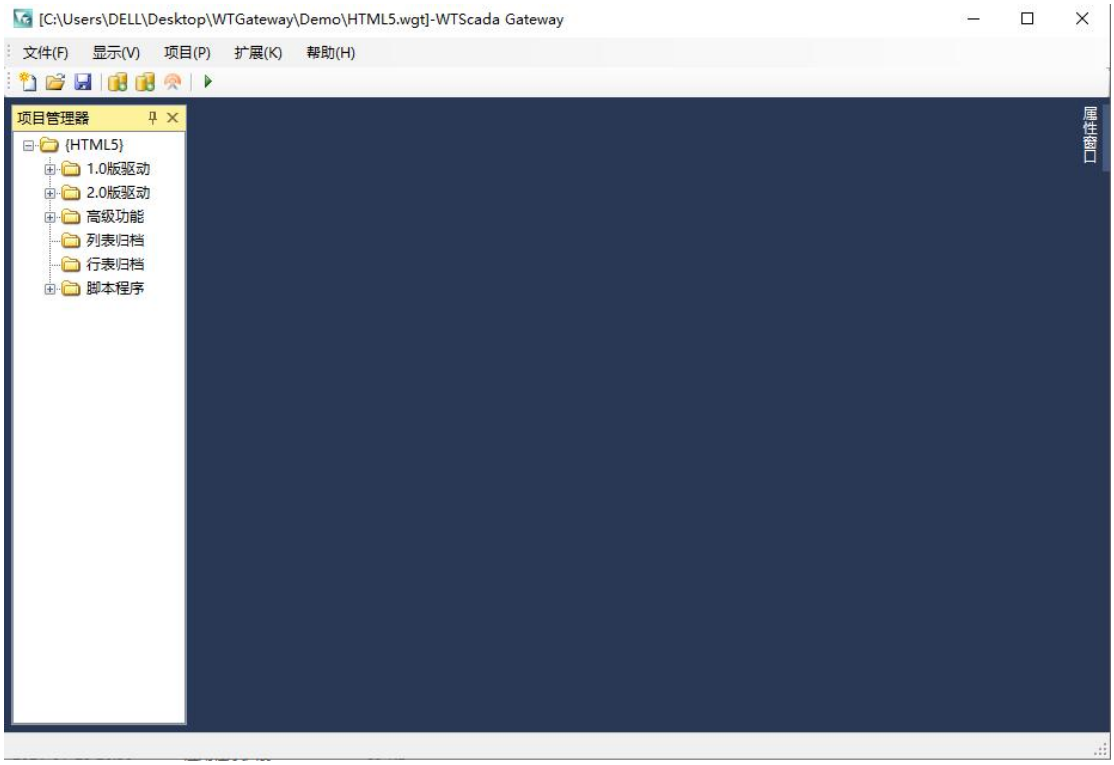
WTScada Gateway 使用手册 v2.13

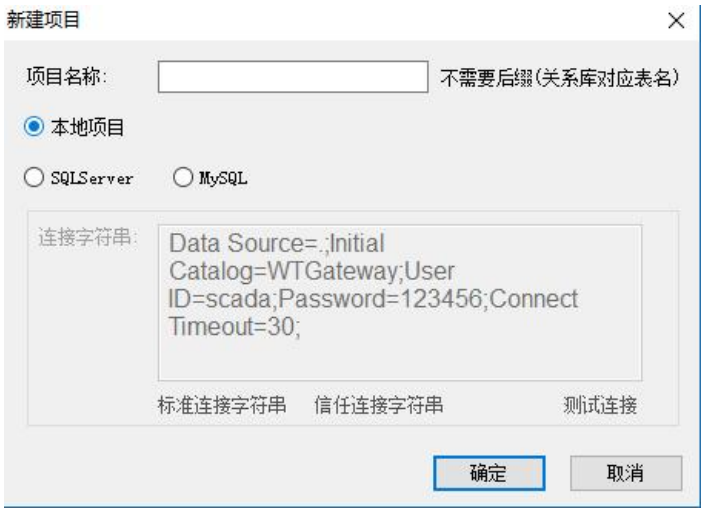
WTScada Gateway 软件是 FScada 软件的升级版本，包括数据采集和网页呈现功能，项目配置存储在关系数据库中，采用 JSON 格式存储，驱动配置支持在线修改和删除，支持多数据库历史、报表归档，操作便捷性和易用性比 FScada 有了较大提升，特别适用于 MES 数据采集、信息化项目。WTScada Gateway 大部分功能和驱动已经可以在 Linux 系统下运行（使用 Windows 版本组态环境进行项目配置）。WTScada Gateway 包含 32 位和 64 位 Windows 版本。大于 5 万点 IO 的项目建议使用 64 位版本，32 位版本有可能会出现内存不够的情况（尤其是组态时的数据导入）。

1. 配置环境



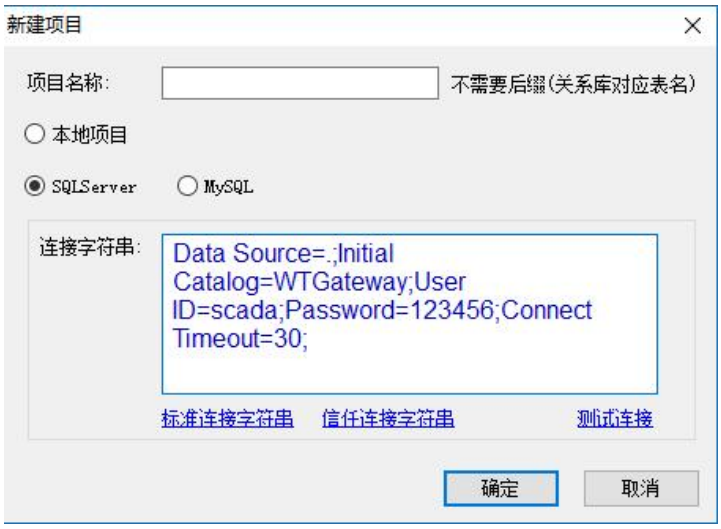
WTGateway 软件支持本机保存项目组态文件和使用 SQLServer、MySQL 存储项目文件，支持项目相互转换。

默认情况下运行 Designer.exe 会尝试获取版本更新，如果电脑无法连接网络时可以通过“项目”菜单下的“自动更新”菜单关闭该功能。



新建项目对话框中可以进行项目存储方式的选择，本地项目使用 SQLite 存储，项目文件后缀为 wgt，该文件是 1 个 SQLite3 标准数据库。

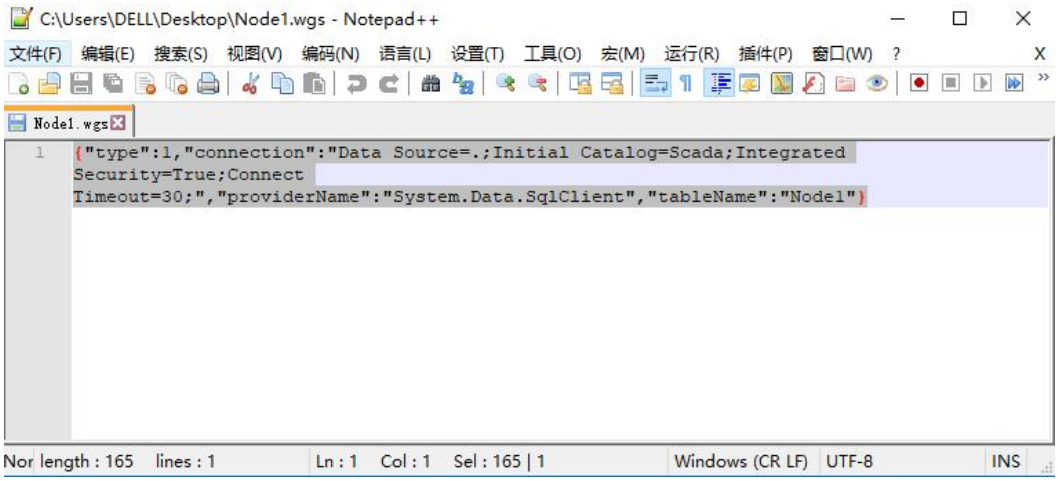
SQLServer 和 MySQL 项目的项目文件后缀为 wts，该文件是 1 个 json 文本，只记录数据库信息，项目信息全部存储在关系数据库中。



（密码可以加密存储，详见附录）

数据库连接字符串提供了模板，只需要修改数据库地址，名称，账号和密码就可以，1 个数据库可以存储多个项目，每个项目使用独立的表名为项目文件名。当创建 SQLServer、MySQL 关系库存储项目时不存在指定的表，则创建新表，添加默认配置。

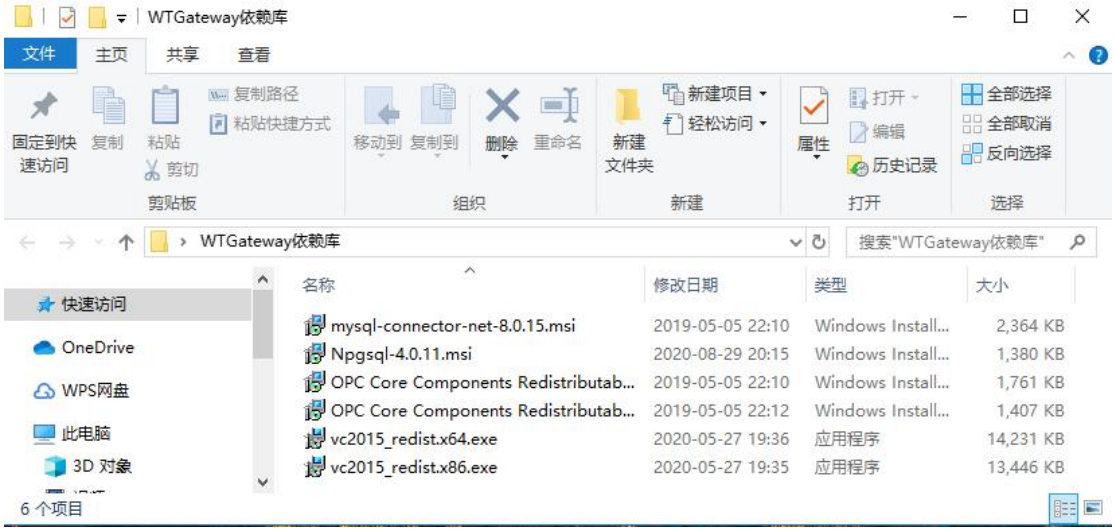
wts 格式的数据库项目文件也可以手动创建和修改数据库信息，用记事本软件打开项目文件进行修改内容。



上图是 SQLServer 数据库项目文件格式，tableName 就是数据库中表的名称

使用 MySQL 存储项目数据时要注意由于 MySQL 默认数据包尺寸为 1M，某个驱动如果数据量大可能会出现保存失败，这时候需要修改 MySQL 默认数据包尺寸。

依赖包说明：



1）如果 js 脚本引擎无法使用请安装 vc2015_redist 运行库。

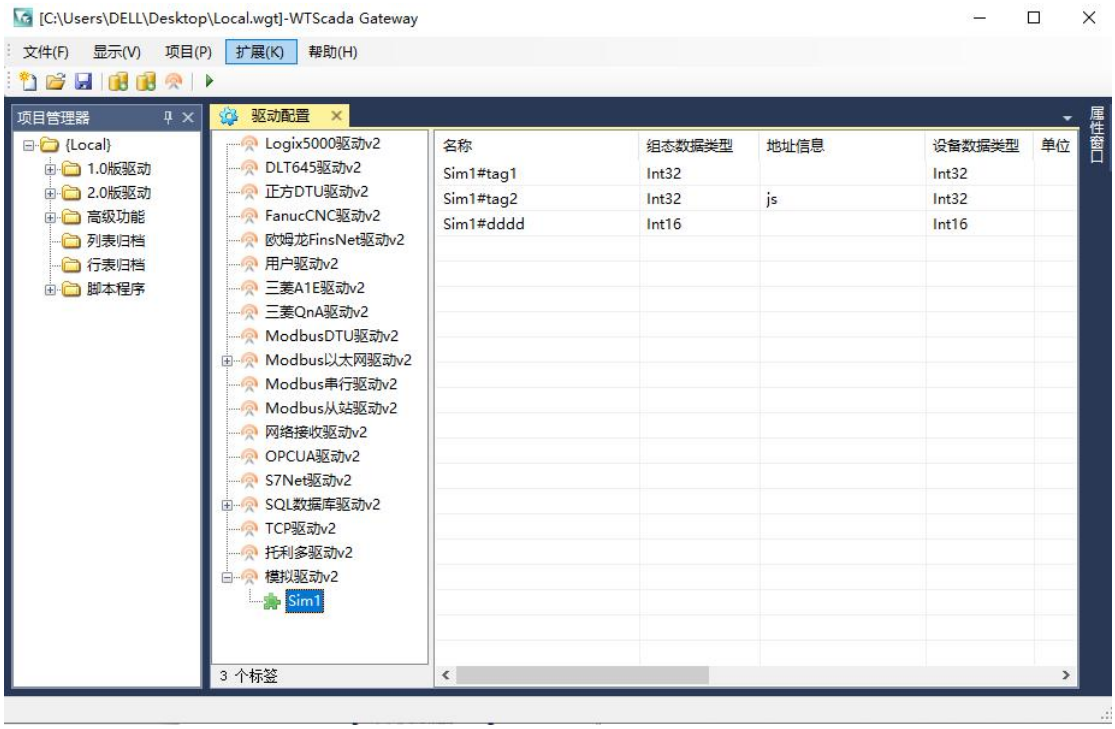
2）MySql-connector-net-8.015.msi MySQL 驱动，必须安装才可以连接 MySQL 数据库

3）Npgsql-4.0.11.msi　PostgreSQL 驱动，必须安装才可以使用 PostgreSQL 数据库

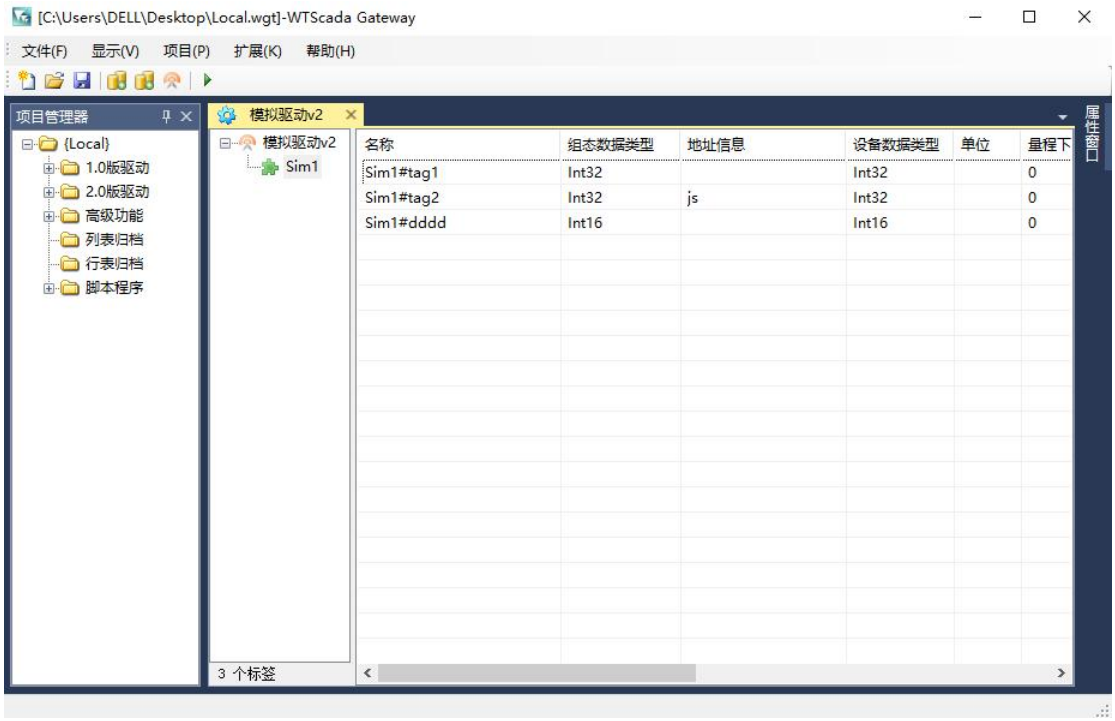
4）Opc Core Components 是.Net OPCDA 发行包（OPCNet DA 驱动无法使用可以安装下该库）

1.1 驱动配置

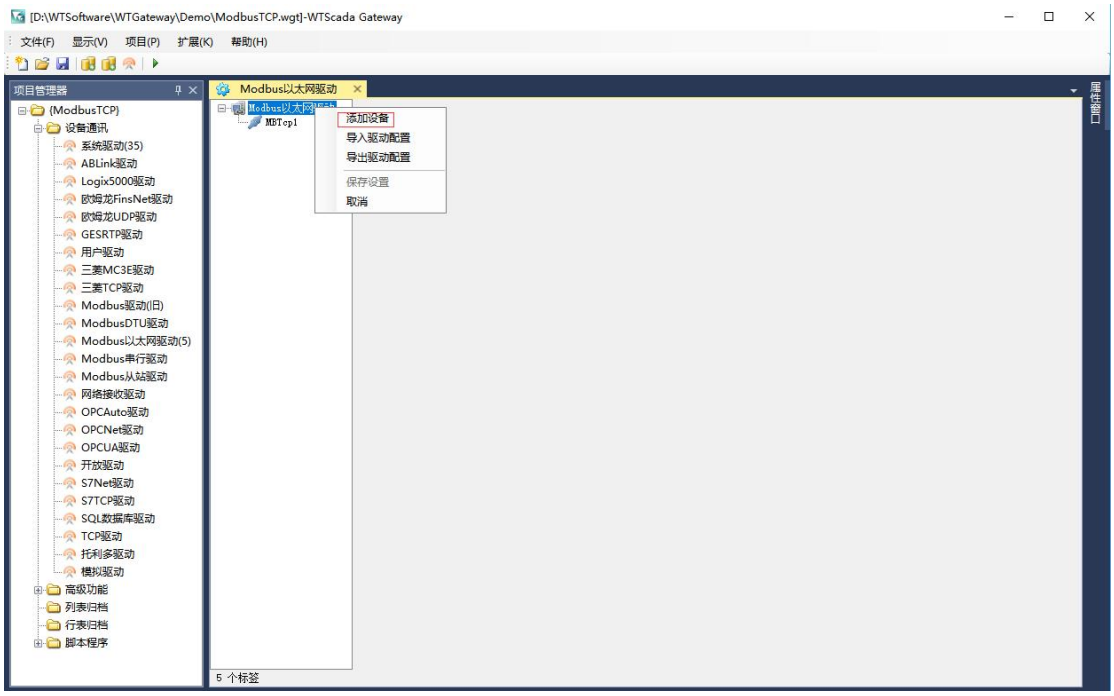
项目管理器目录树设备通讯上鼠标右键可以打开全局驱动配置界面，驱动分为1.0版本和2.0版本，2.0版本按跨平台模式设计，可以在Linux系统运行。



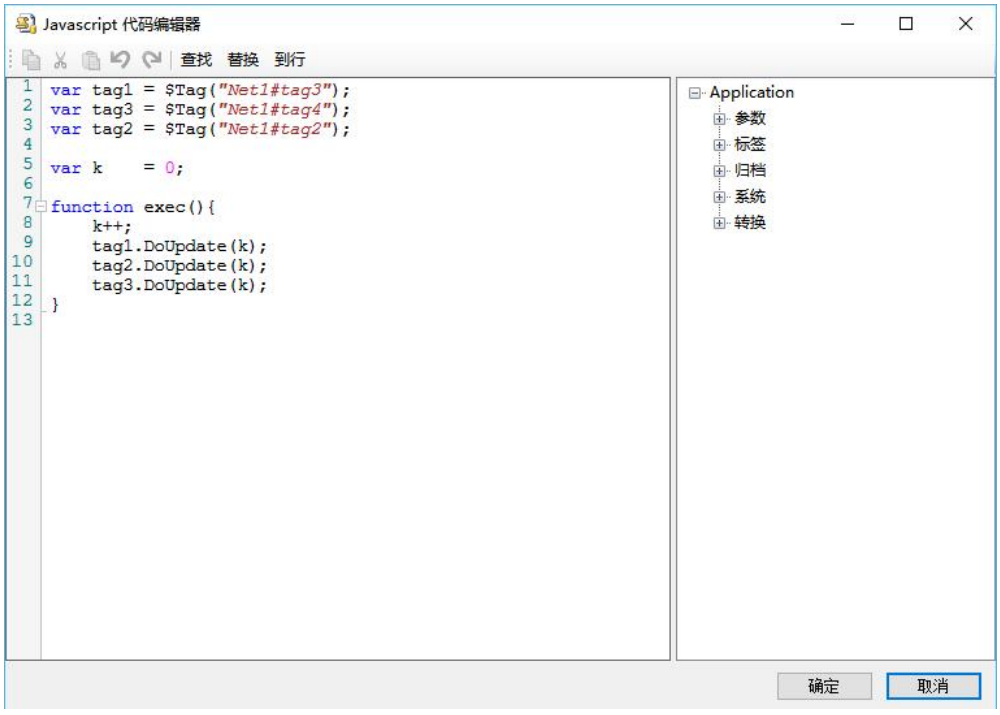
单个驱动节点鼠标双击或者右键菜单可以打开单个驱动的配置界面



所有的驱动配置方式都是在驱动节点上鼠标右键添加设备，选中设备有添加变量

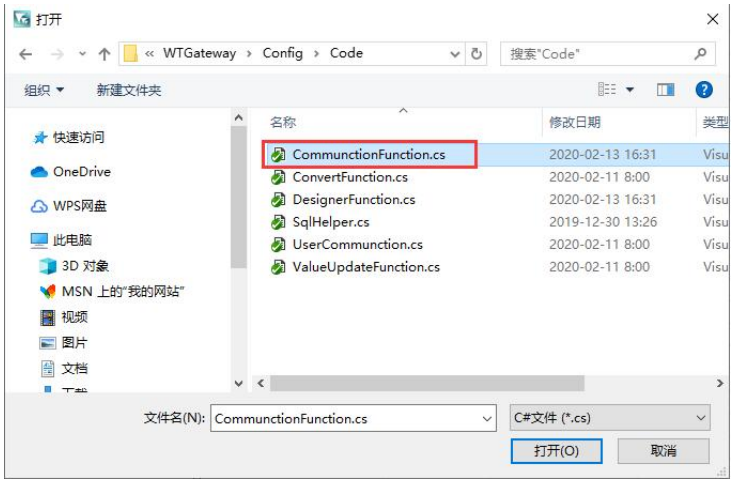


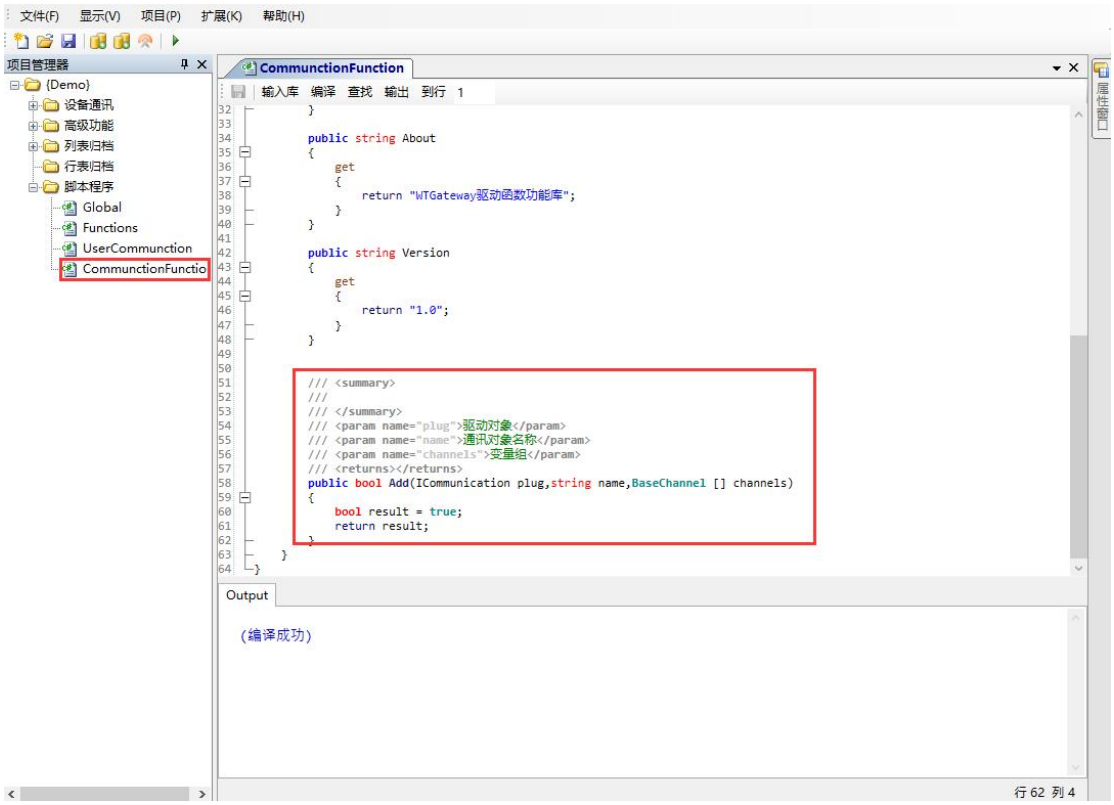
大多驱动的设备上都提供了 C#脚本执行函数和 JavaScript 函数（JavaScript 函数说明见附录），该函数在驱动的每个采集周期正常完成后被调用，可以进行复杂的计算处理。



注意：exec 是循环执行函数，不可以删除。

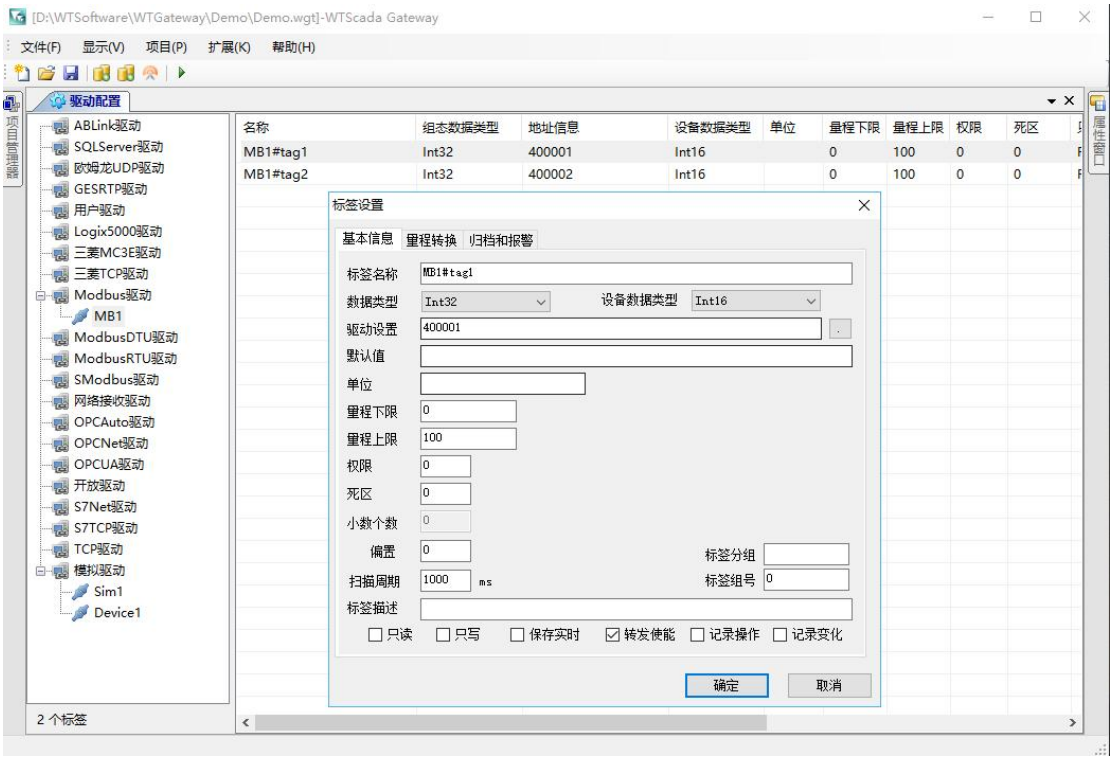
Config 目录下提供了执行函数的模板文件。





编译成功后周期执行函数的下拉框可以选择到函数

函数中参数中的 name 是设备名称，channels 数组是设备的变量数组，通过该函数可以用来更新内存变量的值或者执行归档操作。



标签分组：BaseChannel 的 ChannelArea 属性，字符串

标签组号：BaseChannel 的 Area 属性，整数

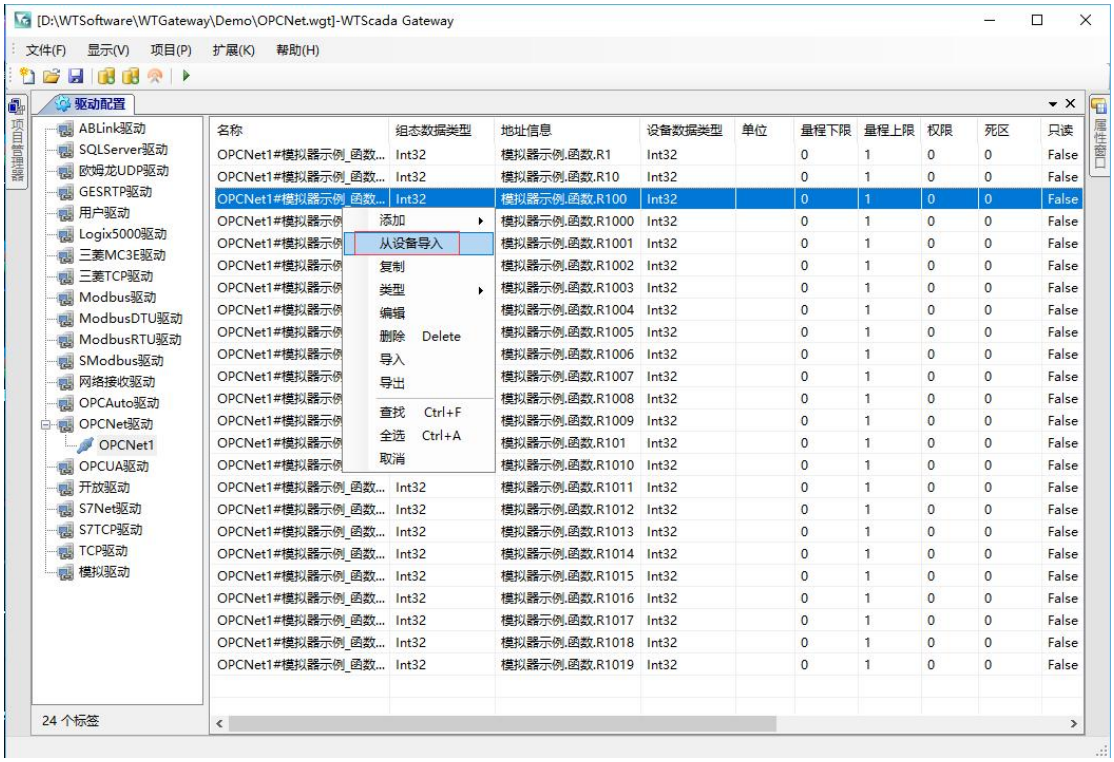
通过该设置可以把某些变量进行归类，方便 c#代码处理

特殊作用：

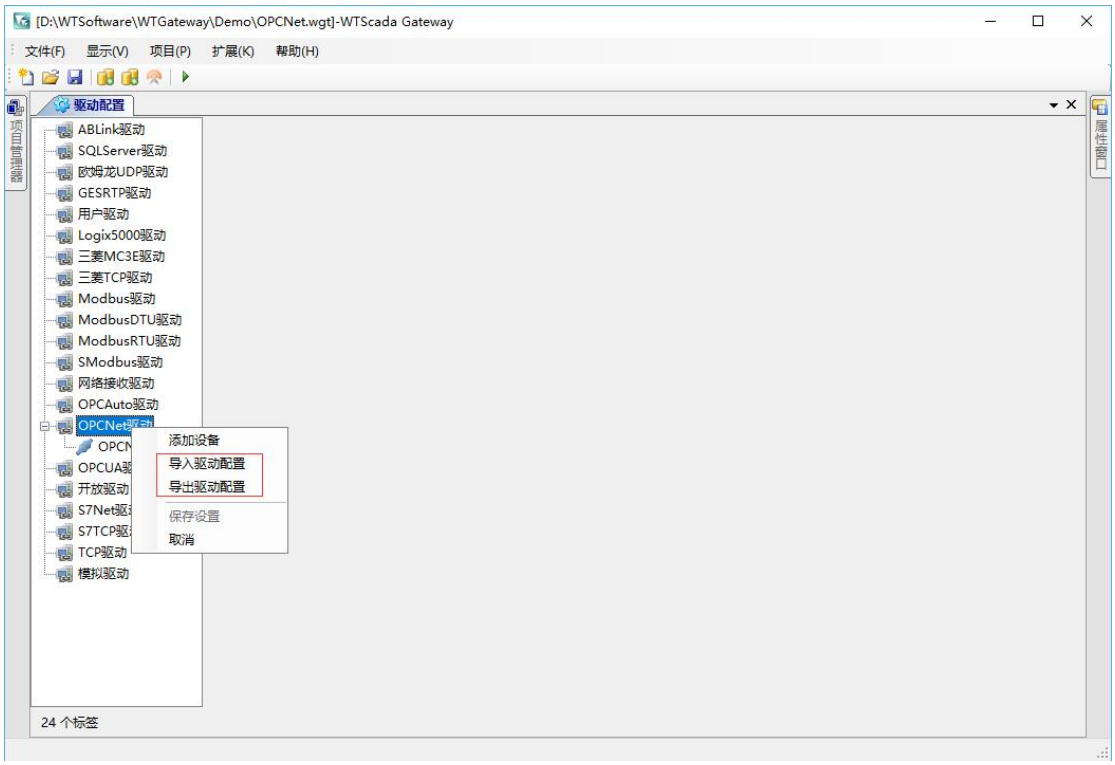
1）网络接收驱动中如果收到的变量数据属性中 ChannelArea（标签分组）为 9999，则强制启用历史归档。

2）如果变量上设置的 Area 属性为 9999 时，历史归档对该变量使用变量时间进行存储，历史归档中的变量时间总是和实际时间相同，通常用于实时数据库中某些特定变量的存储。

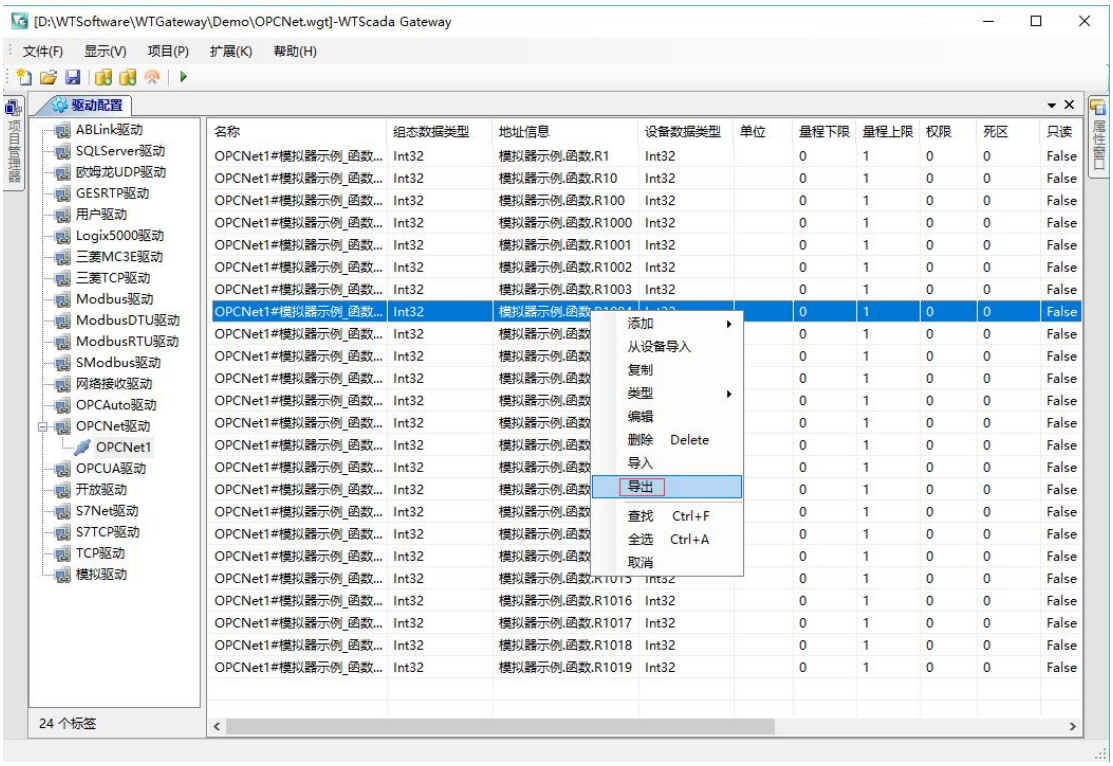
某些驱动支持从设备导入变量（OPC 驱动、TCP 驱动、PI 驱动）



所有驱动均支持JSON格式的导入和导出功能,导出的数据格式和存储在项目文件中的格式一致。

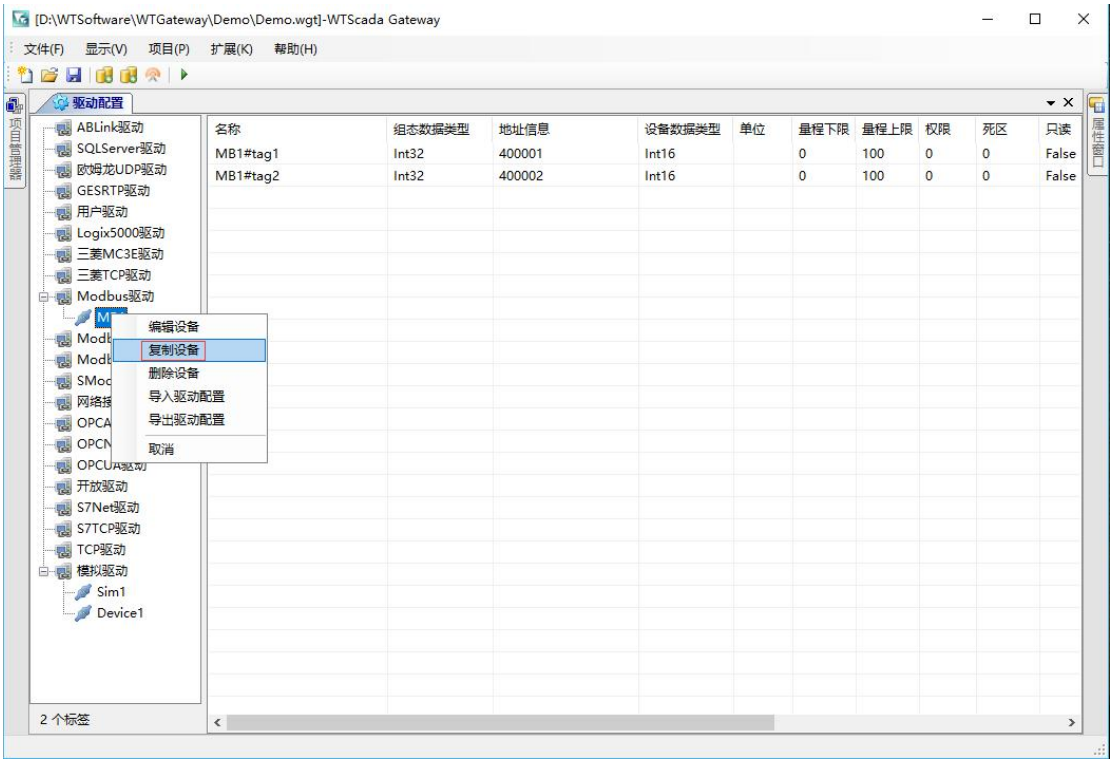


所有的设备均支持 CSV 格式的导入和导出

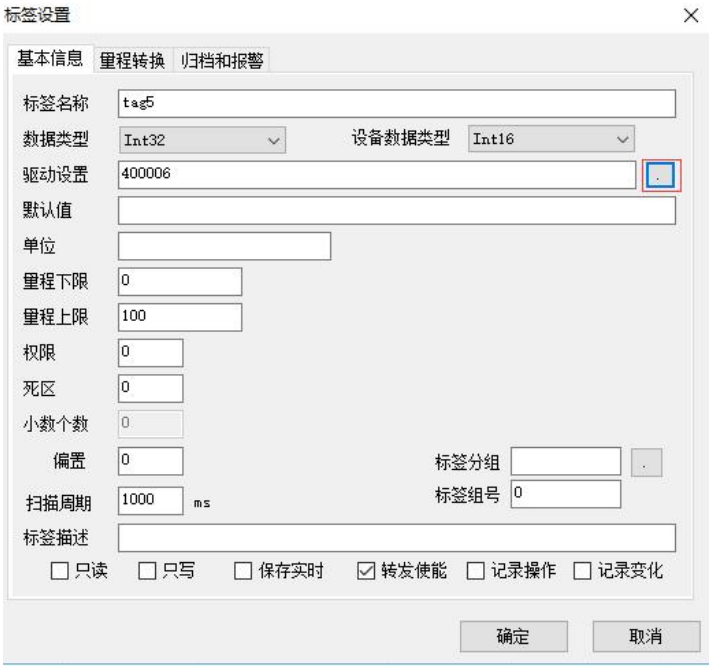


导出 CSV 格式说明：
导出时用于替换换行和回车符号

所有的设备均支持驱动内的复制操作，用于快速创建相同类型的设备



1.2 变量配置



标签设置界面对所有驱动都是相同的，部分驱动没有设备数据类型的选择，驱动设置按钮有的驱动没有实现，直接在驱动设置文本框内输入变量地址信息。

转发使能：TCP、UDP、IOServer 转发服务使用（***默认勾选，如果不勾选则该变量不会转发，IOSever 也不会处理该变量***）

记录变化：变量变化后记录到日志数据库

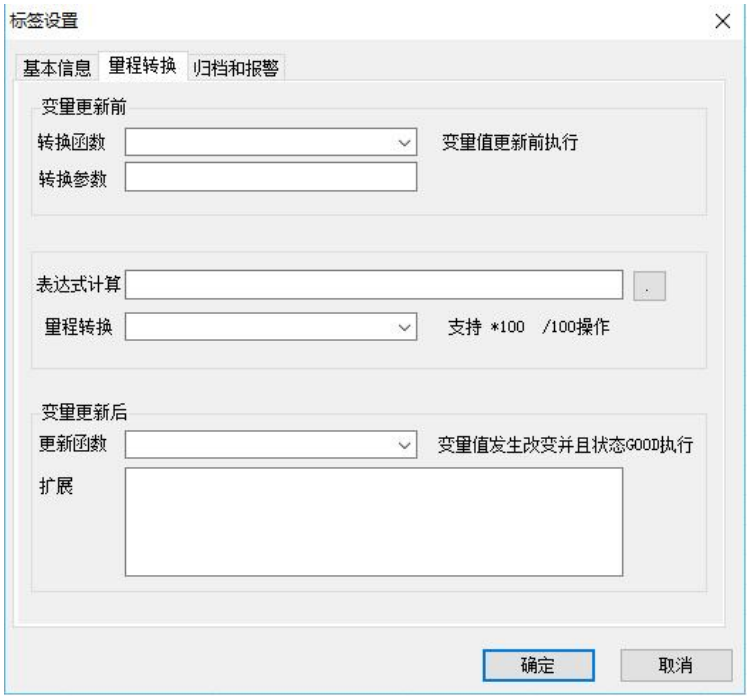
操作记录：变量值设置记录到日志数据库

量程上下限：不会限制变量值的显示和设置，主要用于趋势

保存实时值：系统停止前把变量的当前值存储，再次运行显示上一刻的值

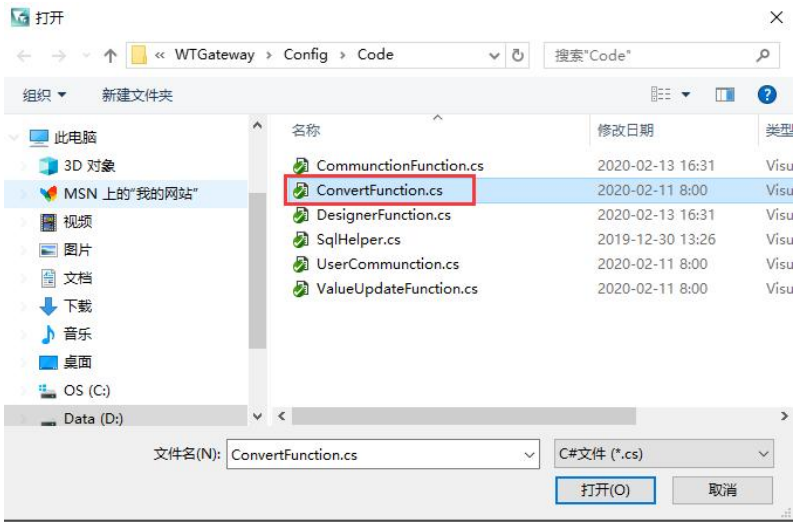
偏置：对输入的值进行加减操作，相当于零位调整，该设置被应用于反向输出。

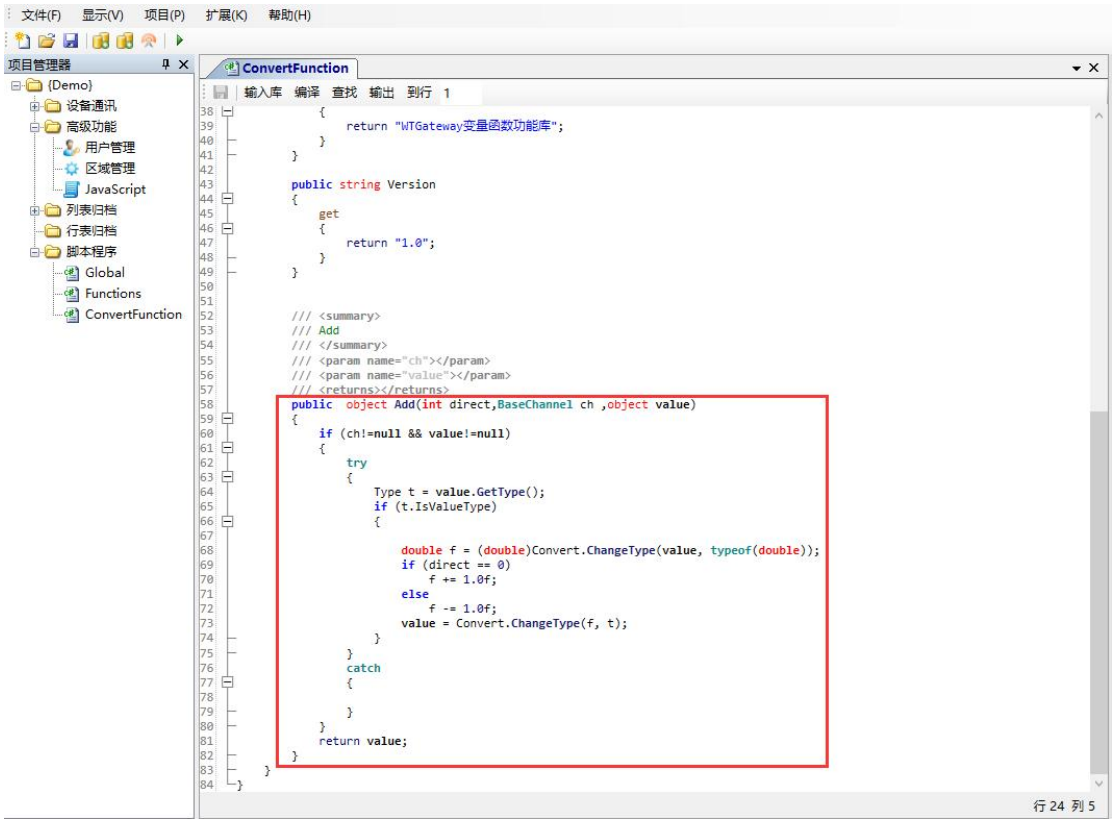
扫描周期：某些驱动不支持变量设置单独的扫描周期时使能禁止。



变量更新前：

变量被驱动采集到后可以通过设置变量更新前转换函数进行数据变换处理，提供了 1 个转换参数，该函数是 C#函数，Config 目录下有函数模板文件，导入到项目修改编译后就可以被选择。





该函数有 3 个参数

direct： 0 表示输入方向，1 表示输出方向

ch：变量对象

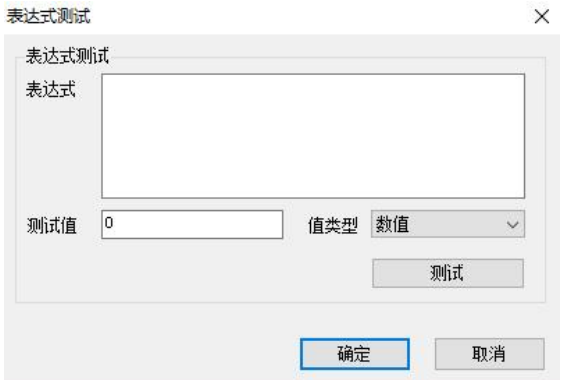
value：驱动采集的值，如果设备具有独立的数据类型，则 value 的数据类型就是设备数据类型，否则就是组态数据类型

该函数的返回值在 direct=0 时更新到变量值，direct=1 时输出到设备。

direct=0 时函数返回值必须能被转换到组态数据类型，direct=1 是输出类型必须能被转换到设备数据类型。

注意：不要在函数内执行耗时操作，会阻塞采集线程。

表达式计算：用于变量显示前的计算，val 表示当前变量值



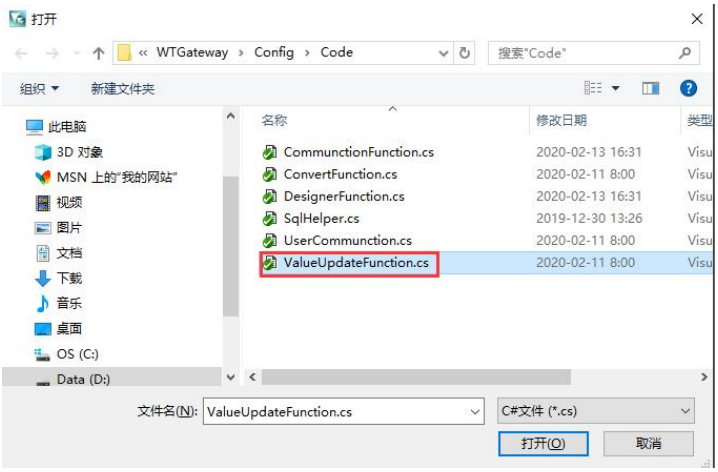
提供了 1 个测试界面用于测试计算表达式是否正确，测试值就是 val 的值

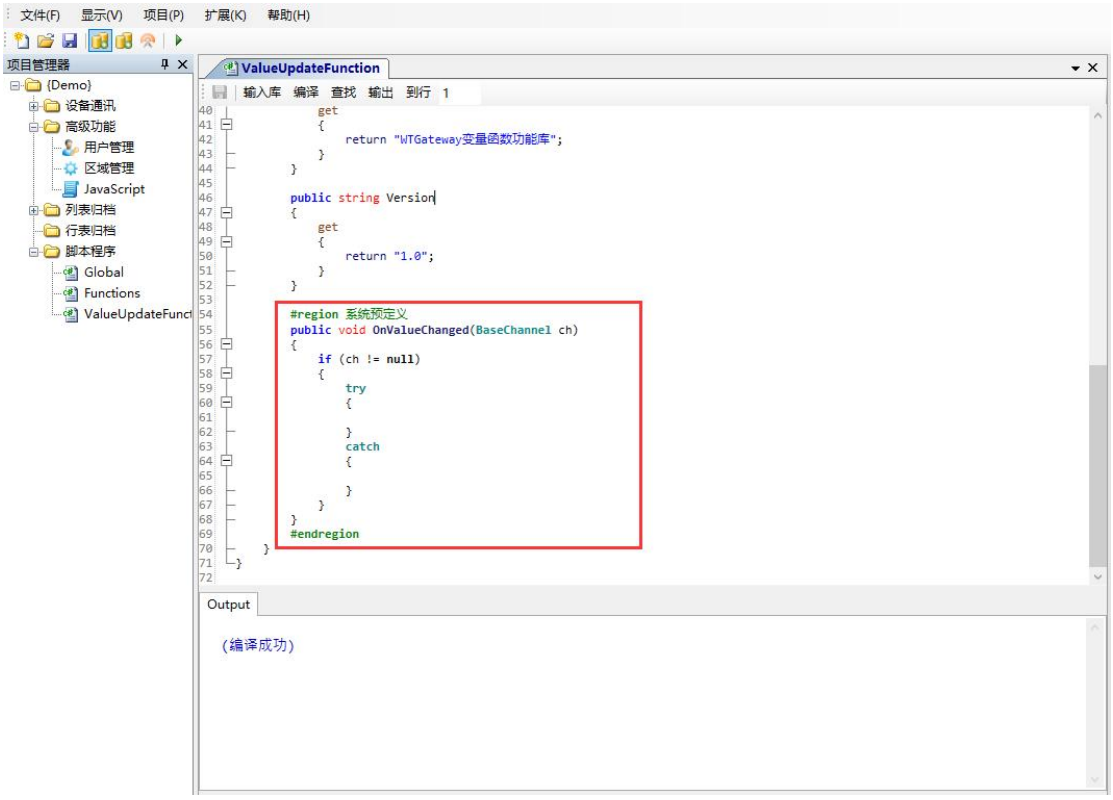
量程转换：提供简单的加、减、乘、除转换，该转换可以被反向输出，量程转换和表达式计算只能选择 1 个。反向输出的意思就是输入假如输入转换是乘 100，那么输出是就自动除 100。

如果同时设置了量程转换和表达式计算，则量程转换有限，表达式转换无效。

变量更新后：

也可以配置 1 个 C#函数用于复杂操作处理，Config 目录下有函数模板文件，导入到项目修改编译后就可以被选择。





注意：不要在函数内执行耗时操作，会阻塞采集线程。

提示：在驱动中没有配置地址的变量均为内存变量，内存变量总是在驱动采集周期成功完成后执行扫描，因此内存变量的计算结果总是和驱动采集周期同步的，如果内存计算变量的扫描周期时间设置为 0 则每个驱动采集周期都执行计算，如果非 0 则计算间隔大于设置值才扫描计算。

1.2.1 量程转换举例

如果要对某个变量输入值是 4000~32000，需要转换为 0~100

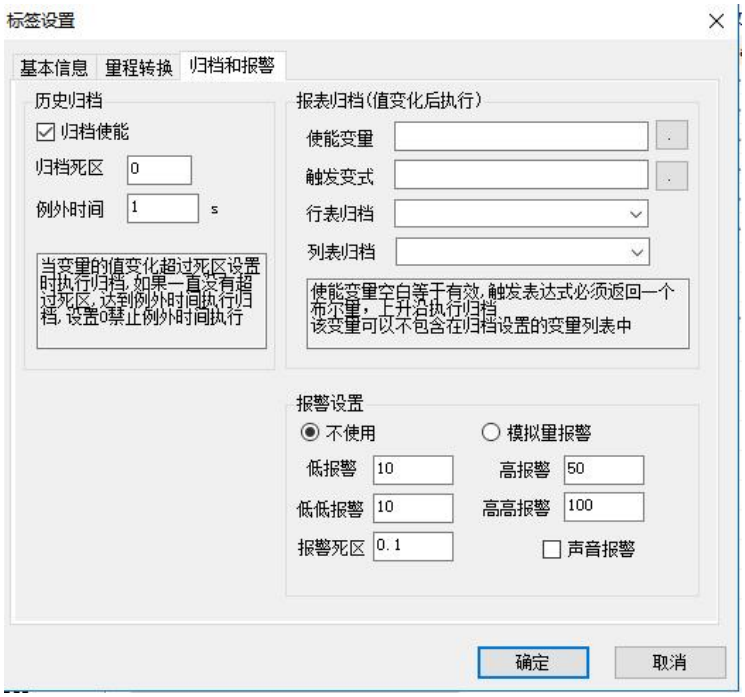
1)变量的数据类型应该设置为浮点数

2)设置偏置为-4000，量程转换为/28

计算过程为：(x-4000)/28

或者设置计算表达式为：(val-4000)/28

第一种方式当发生数据写入时能够自动被转换为正确的值写入设备，第二种方式不会反向转换。



1.2.2 历史归档

归档使能：启用变量的历史归档，支持运行时修改

归档死区：变量的值和上一次归档时的值绝对值大于死区设置就进行归档（必须设置为正数），支持运行时修改

例外时间：如果变量的值一直不变化，达到例外的设置时间也会进行一次归档，设置为 0 不使用例外时间（必须设置为正整数），支持运行时修改

1.2.3 报表归档

使能变量：控制归档的变量设置，使用变量的布尔值判断（支持运行时修改）

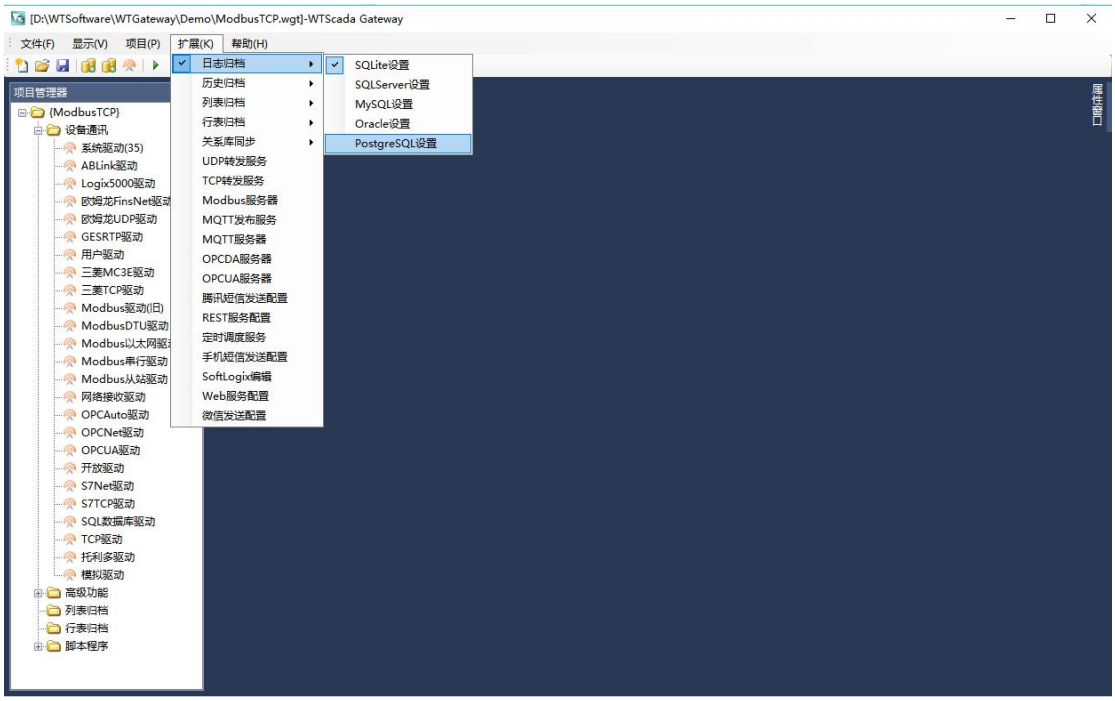
触发变式：表达式计算，输出必须是布尔类型（支持运行时修改），触发表达式输入 True 的上升延执行归档

行表归档：选择行表归档名称，触发动作后执行行表归档的写入，如果在行表归档设置前面加@符号，则归档时仅对当前变量存储

列表归档：选择列表归档名称，触发动作后执行行表归档的写入

该功能当变量的值发生改变后判断是否需要执行归档写入。

1.3 日志归档



扩展菜单的菜单项都有功能启用显示，菜单项前有√符号表示该配置已经启用。

项目创建后默认自动启用 SQLite 归档，当前支持 SQLite、SQLServer、MySQL、PostgeSQL、Oracle 归档。

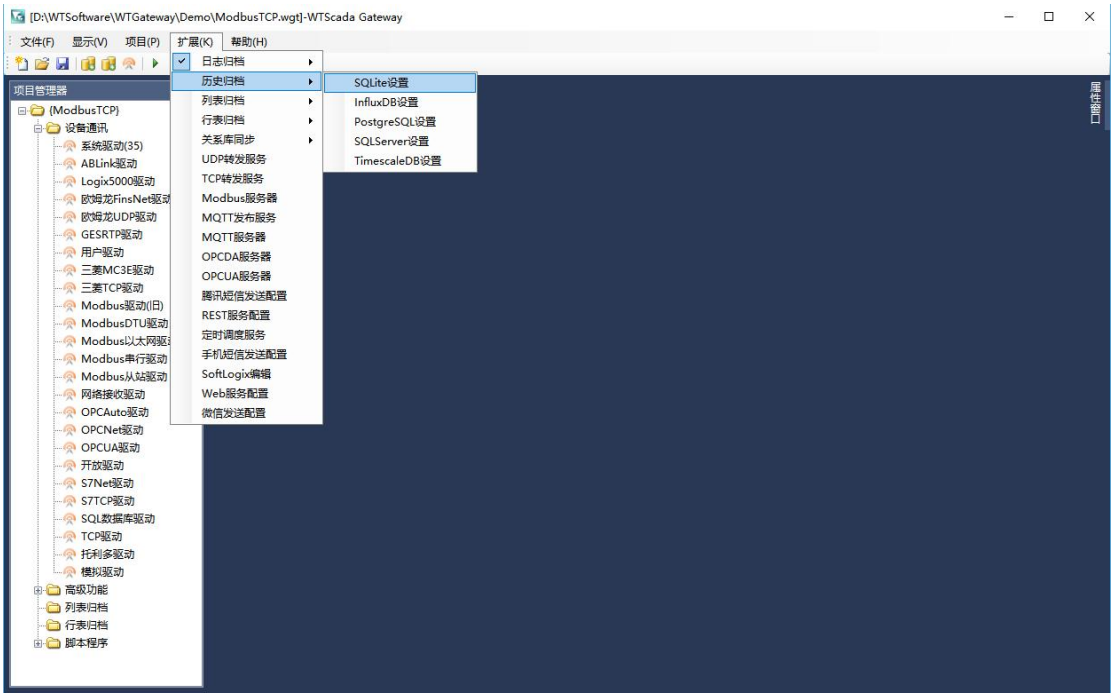
可以同时配置多个归档数据库，运行时同时记录

默认归档：用于运行环境日志查询选择的查询来源

日志归档都支持定时删除功能，每天零点删除保存时间之前的数据。

1.4 历史归档

历史归档数据库配置



当前支持内置 SQLite 数据库，InfluxDB 时序数据库，TimescaleDB、PostgreSQL、SQLServer、MySQL、Oracle、唐码实时库和 PI 实时库进行历史归档。

1000 点以下的历史归档可以使用内置 SQLite 数据库，历史数据存储在项目文件路径的同名目录下（如项目文件为 d:\scada\demo.wtg，则历史归档文件存储在 d:\scada\demo 目录下）。

1000 点以上的变量推荐使用 InfluxDB 开源时序数据库，该数据库可以轻松实现每秒 5 万变量的历史存储，很快的查询效率和压缩比，3 万变量每月存储容量约 30G。

历史归档线程运行周期 100ms。

1.4.1 InfluxDB 历史库配置

把 InfluxDB 库安装到设置的路径，配置好相关设置，点击启动，启动后点击测试连接提示正常就可以使用。

如果要对字符变量进行归档勾选“启动字符变量存储”，字符变量使用 1 个独立的列存储，这意味着数据存储容量会增加。

地址设置支持如下格式：

1）不需要认证的格式：http://localhost:8086 (IP 地址和端口可以根据需要修改，可以不是本机)

2）认证格式: http://localhost:8086;admin;admin(地址、用户、密码)

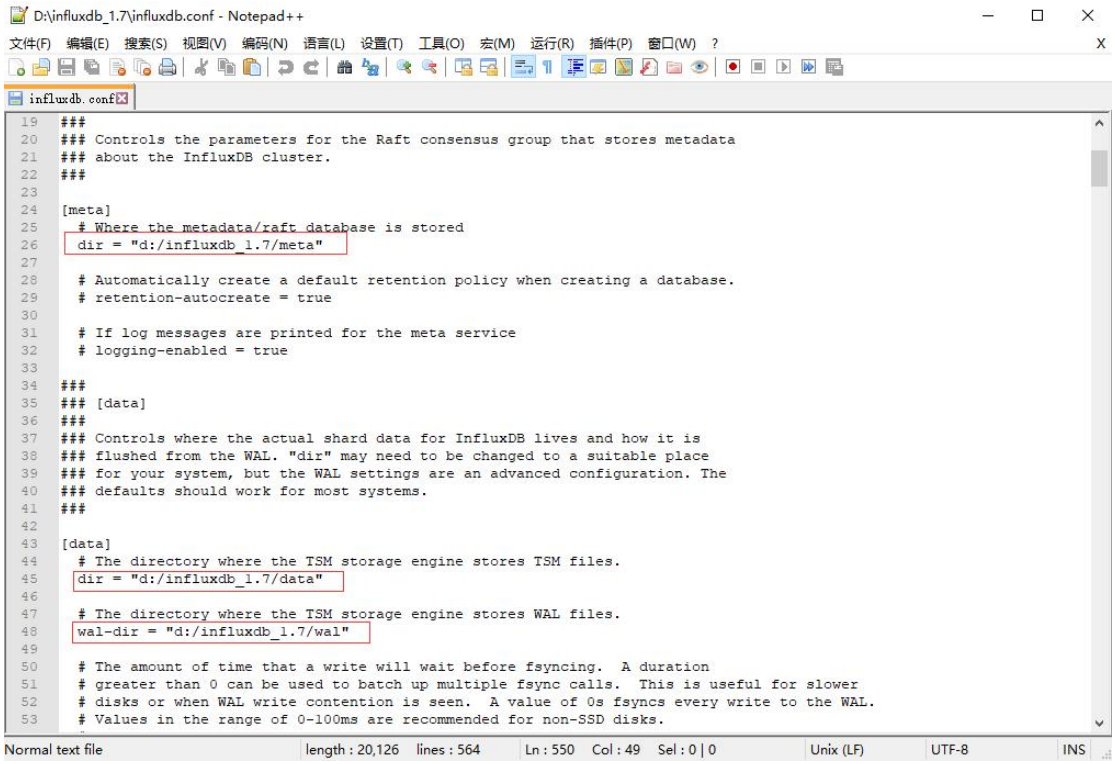
3）完整格式: http://localhost:8086;admin;admin;5(地址、用户、密码、最小归档间隔时间)

最小归档间隔时间单位是秒，设置后会替代变量上的归档例外时间设置，如果最小归档间隔时间是 5 秒，变量上的归档例外时间是 30 秒，则会使用 5 秒，如果变量上的归档例外时间是 3 秒，则不变。

InfluxDB 配置方法如下：

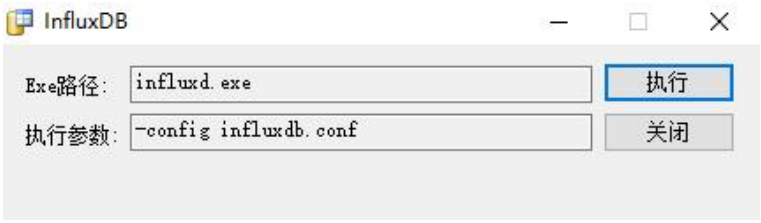
1）解压下载的压缩包到 D:\influxdb_1.7 目录

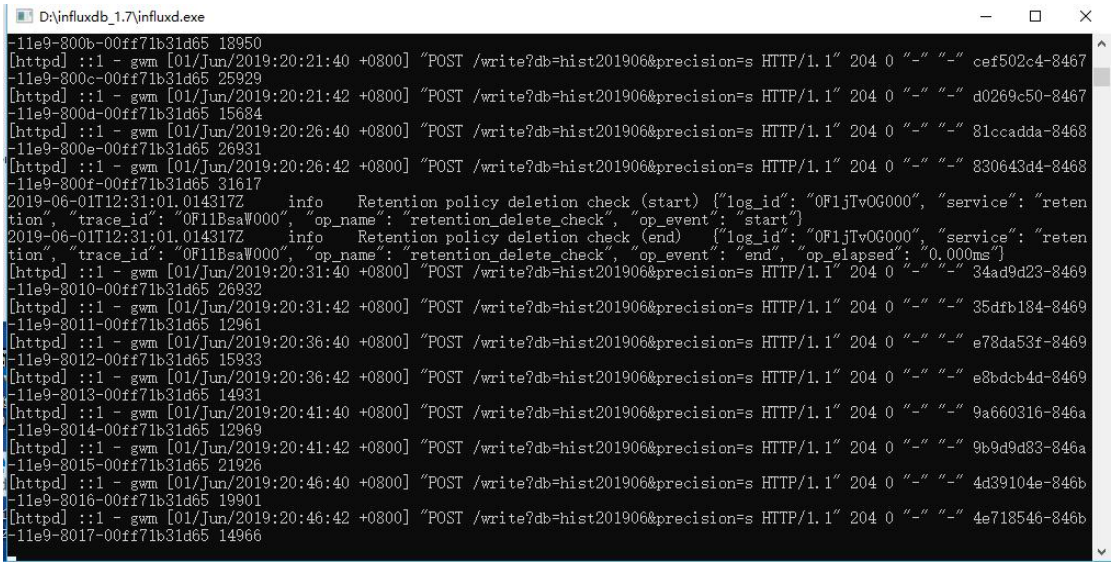
2）如果目录不是 D:\influxdb_1.7，修改 influxdb.conf 配置文件如下段内容的目录。



3）运行 InfluxDB.exe 测试软件是否工作正常

2 个程序最小化显示在任务栏上



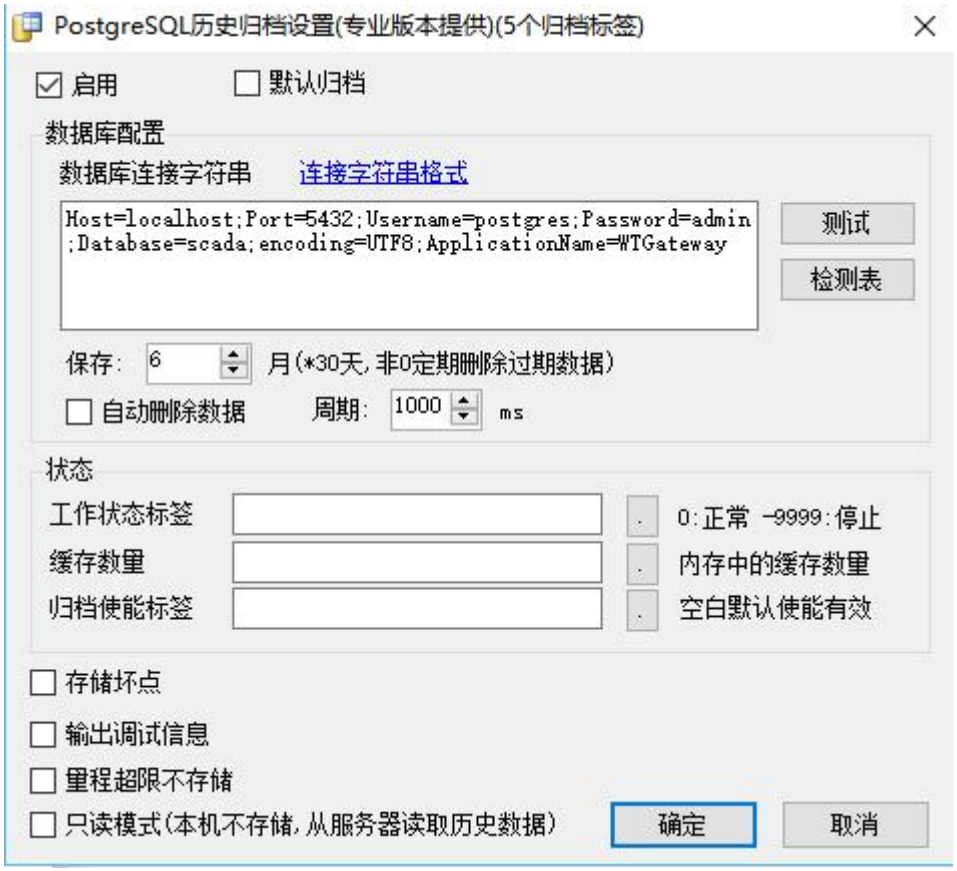


正常后可以关闭 InfluxDB，组态运行启动时会自动启动为后台进程，InfluxDB 相同配置不能同时运行多个进程。

InfluxDB 历史归档每月创建 1 个新的数据库，自动删除数据在每月 1 日 0 点执行，删除保存时长之外的数据库。

1.4.3 PostgreSQL 配置

安装好 PostgreSQL，创建一个数据库就可以使用



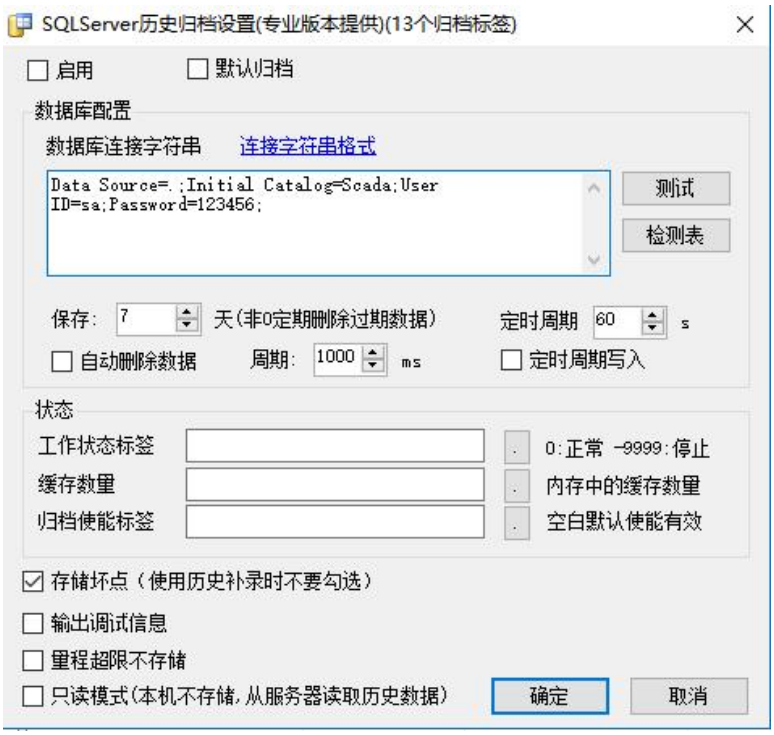
点击“测试”按钮成功后，点击“检测表”安装就完成了表的创建（包括了 1 个变量名称表和历史表）。

使用 PostgreSQL 提供的分区表功能，每天创建一个新表，以 hist_日期命名,由于是数据库的分区表因此可以在主表上进行查询。

从测试情况下 PostgreSQL 的性能和 Timescale 差不多，但是 PostgreSQL 没有 Timescale 那样的聚合函数可以按时间间隔查询。

PostgreSQL 历史归档在每天 0 点删除保存时长之外的时间。

1.4.4 SQLServer 配置



SQLServer 历史归档使用 1 个变量名称表和 1 个历史数据表进行存储，没有分区表功能，因此如果要存储大容量和长时间的历史数据需要 DBA 进行优化，配置分区表，不熟悉 SQLSrever 的用户不建议使用。

如果勾选了定时周期写入，则变量的历史归档死区和例外就不再起作用，按定时周期设置存储。

SQLServer 历史归档在每天 0 点删除保存时长之外的时间。

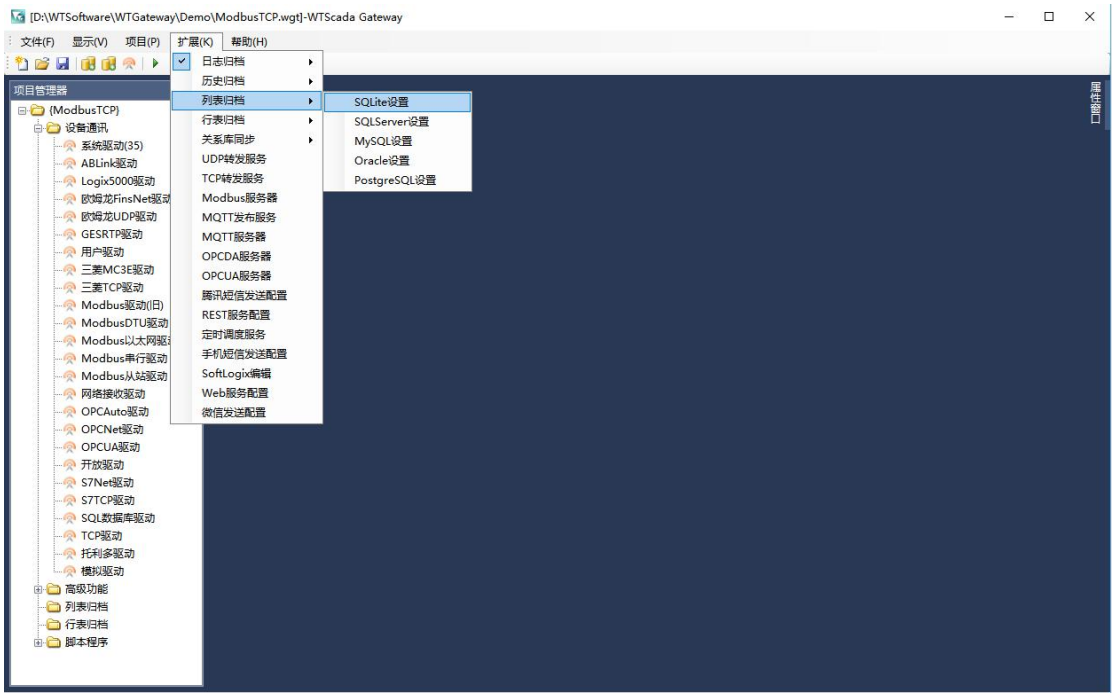
MySQL、Oracle 历史库配置和 SQLServer 方式类似。

1.4.5 实时数据库配置

唐码历史库和 PI 历史库都是第三方商业数据库，需要企业版授权才能使用，参见独立使用手册。

1.5 列表归档

1.5.1 数据库配置

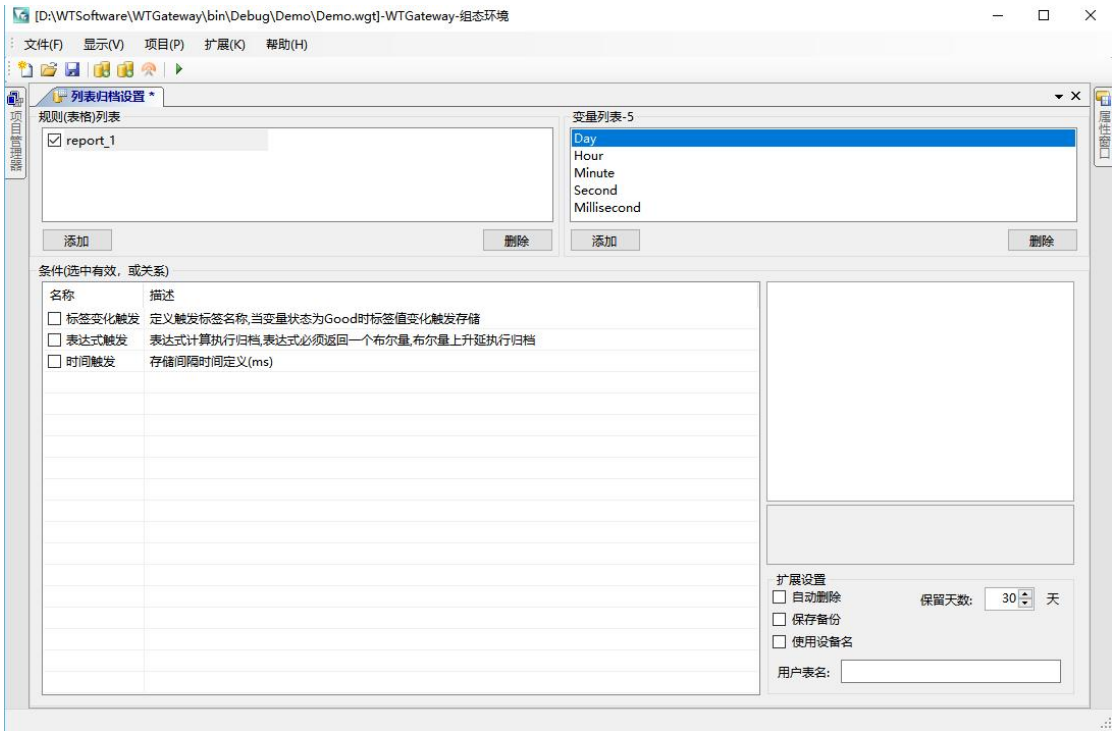


当前支持 SQLite、SQLServer、MySQL、PostgeSQL、Oracle，可以同时配置多个归档数据同时存储，默认归档的功能用于组态运行环境界面的数据查询。

列表归档线程运行周期为 100ms。

1.5.2 存储配置

通过工具栏按钮或者目录树右键菜单打开存储配置

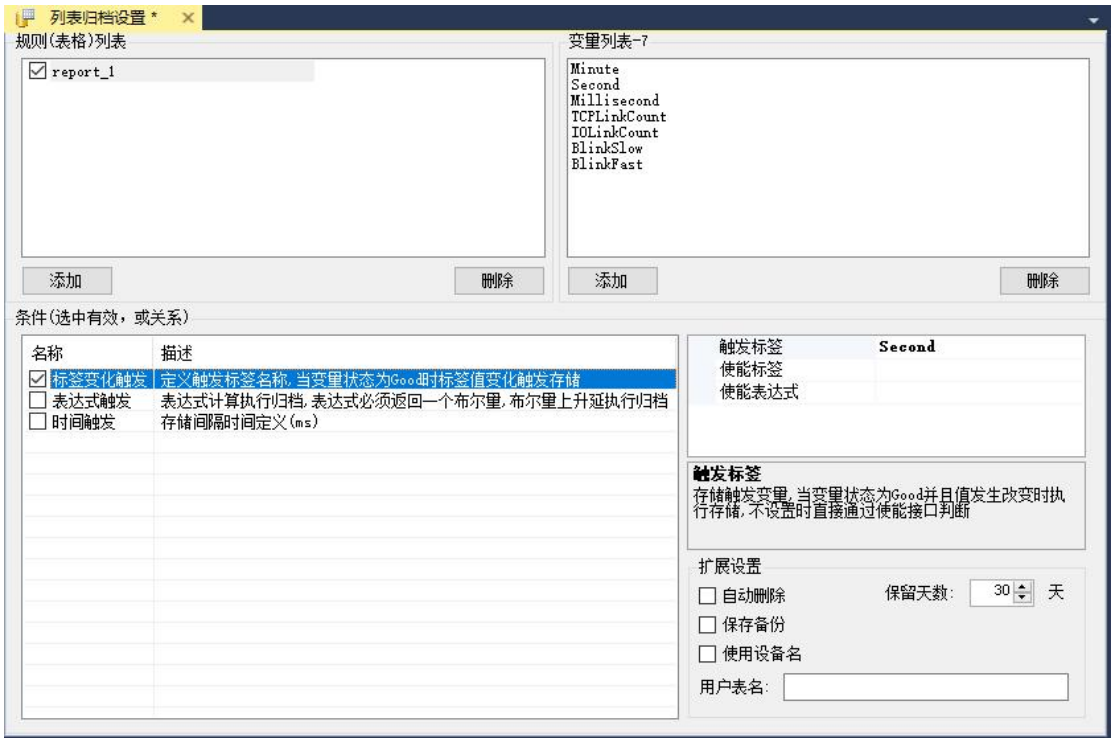


每一个规则表格设置对应数据库中 1 个表，如果配置了用户表名则运行时使用用户表，没有设置用户表的规则运行时自动创建表。

每个规则（对应 1 个表）可以配置 1000 个变量，列表归档按列方式存储数据，所有的变量存储为一行。

归档执行模式有 4 种：

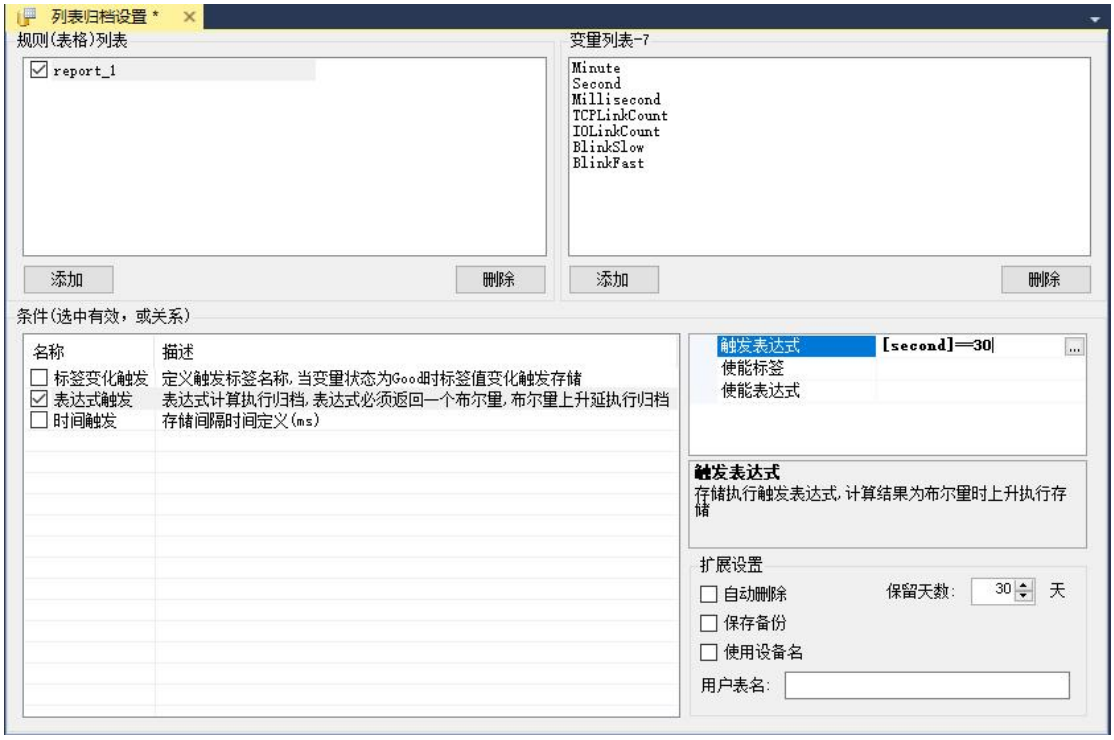
1）标签变化触发



执行流程：当触发标签的状态为 Good，值发生变化后，判断使能状态是否为 True，如果是执行归档。

使能状态判断：如果有使能表达式，则执行表达式，表达式必须返回一个布尔量如果没有表达式，判断是否有使能变量，如果有判断使能变量的状态为 Good，值为 True 则返回 True，如果没有表达式和使能变量直接返回 True。

2）表达式触发



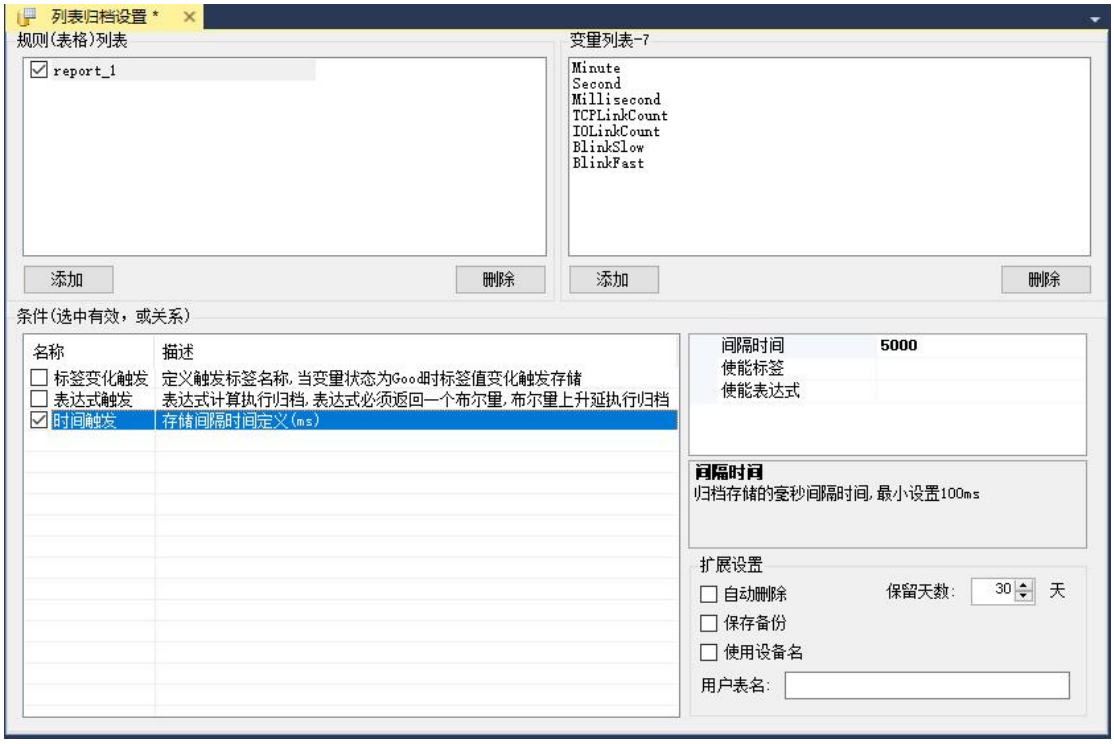
执行流程：判断表达式计算结果是否为 True，如果是 True，再判断使能状态是否为 True，如果是执行归档。

表达式计算必须返回 1 个布尔量

例如：[second]==1 && [tag1]==8

如果秒变量等于 1 并且 tag1 变量等于 8 则返回 true

3）时间触发



执行流程：当归档间隔时间大于设定值后，判断使能状态是否为 True，如果是执行归档。

间隔时间单位为 ms。

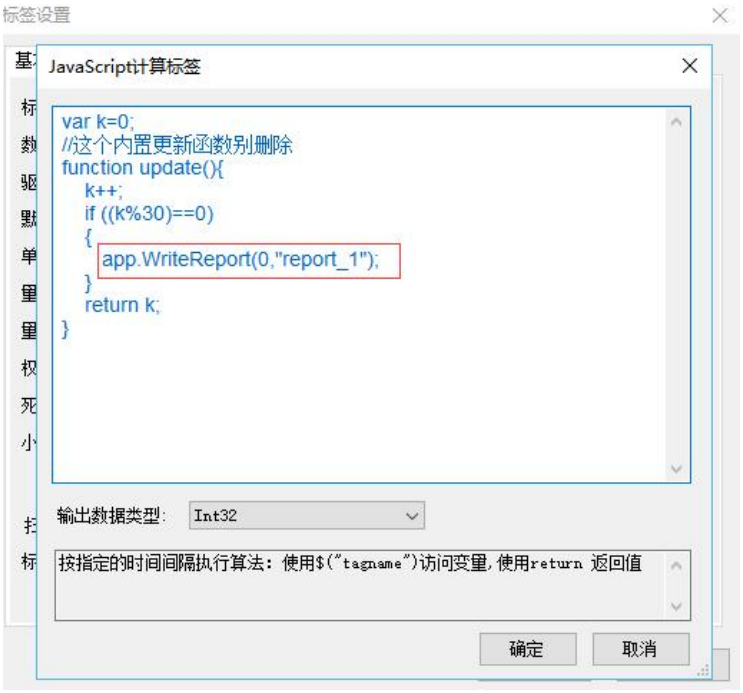
4）自定义触发（C#或者 Javascript 或者变量配置）

变量上的配置参见 1.2 章节

C#执行代码如下：

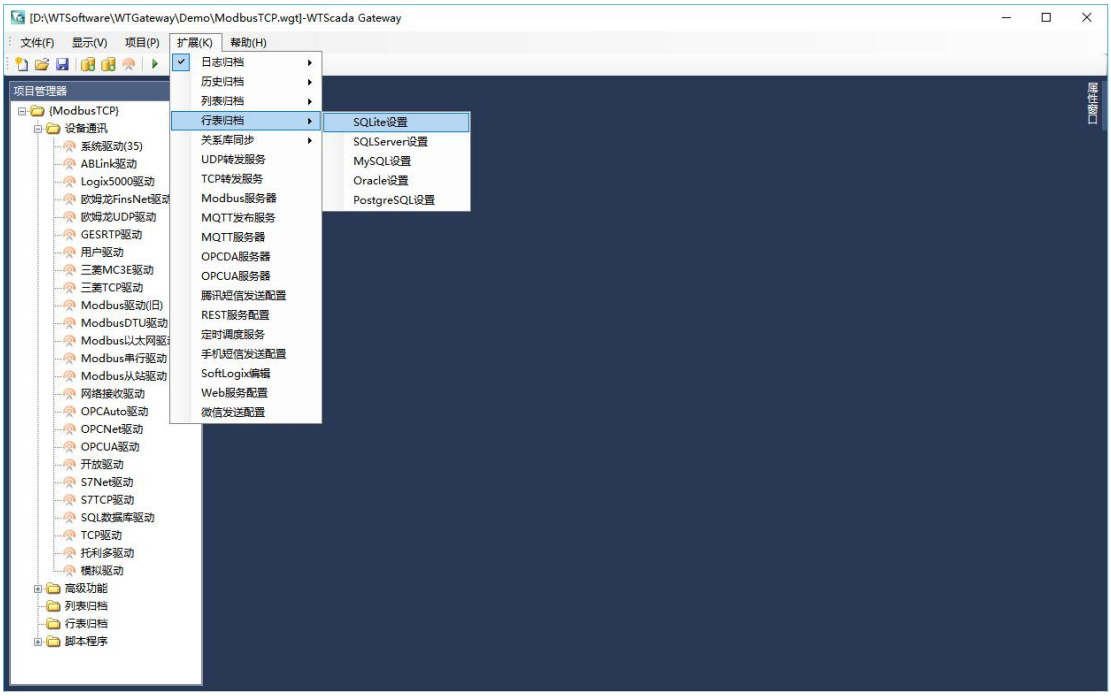
```
if (Env.Current.ColReportArchiver!=null)
{
    //1) 规则名，是否立即写入，否则加入写入队列
    Env.Current.ColReportArchiver.Write("report1",false);

    //2) 规则名，时间，是否立即写入
    Env.Current.ColReportArchiver.WriteArchiver("report1",DateTime.Now,false);
}
```

Javascript 执行代码如下：



1.6 行表归档

1.6.1 数据库配置

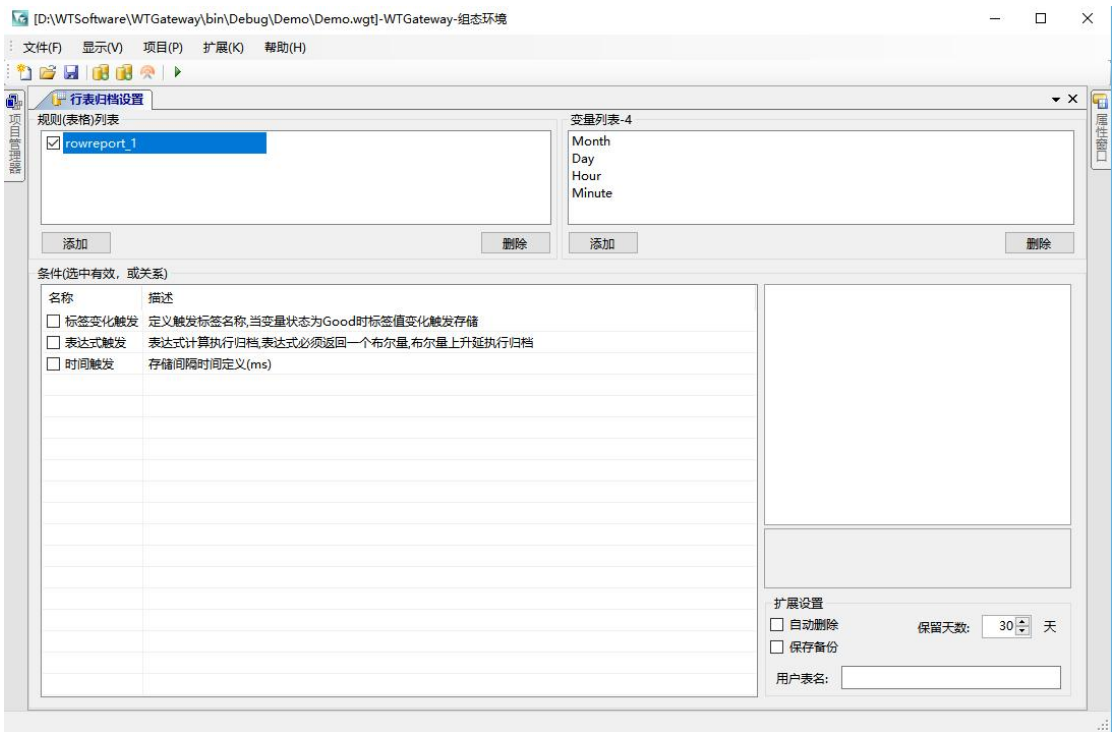


当前支持 SQLite、SQLServer、MySQL、PostgeSQL、Oracle，可以同时配置多个归档数据同时存储，默认归档的功能用于组态运行环境界面的数据查询。

行表归档线程运行周期为 100ms。

1.6.2 存储配置

通过工具栏按钮或者目录树右键菜单打开存储配置



每一个规则表格设置对应数据库中 1 个表，如果配置了用户表名则运行时使用用户表，没有设置用户表的规则运行时自动创建表。

每个规则不限制配置的变量数，行表归档按行方式存储数据，每个变量存储为一行。

归档执行模式有 4 种

1）标签变化触发

和列表归档一致

2）表达式触发

和列表归档一致

3）时间触发

和列表归档一致

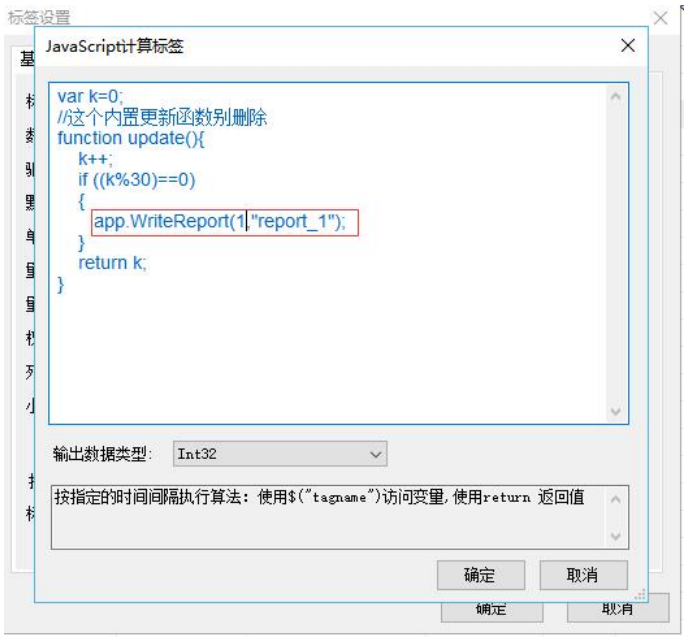
4）自定义触发（C#或者 Javascript 或者变量配置）

变量上的配置参见 1.2 章节

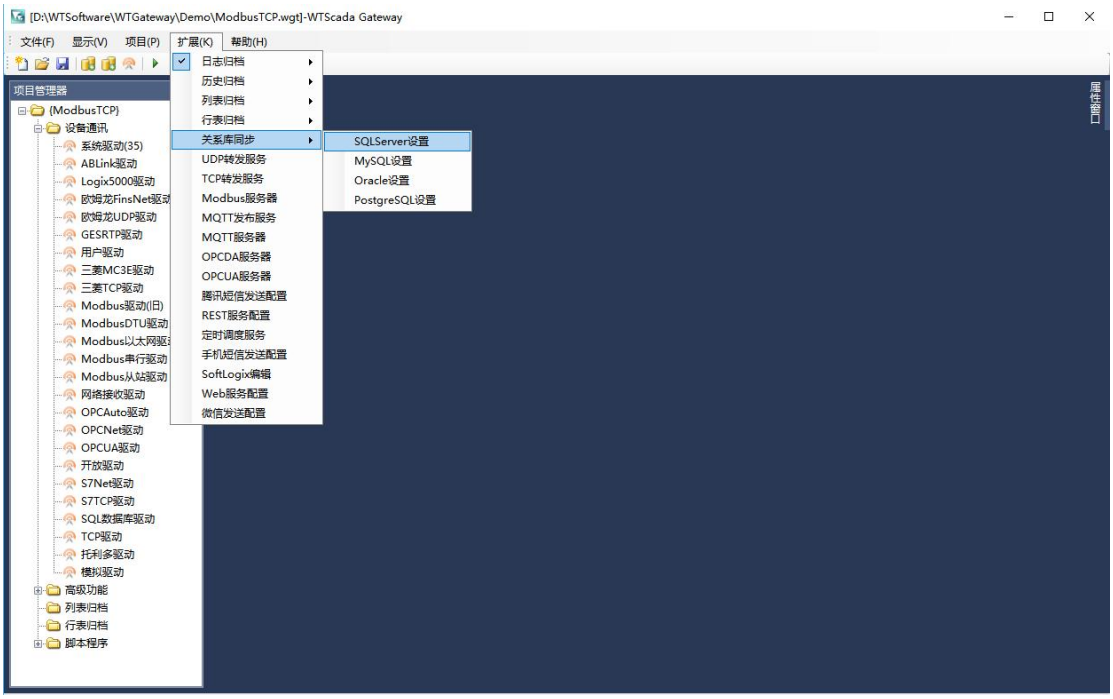
C#执行代码如下：

```
if (Env.Current.RowReportArchiver!=null)
{
    //1)规则名称，是否立即希尔
    Env.Current.RowReportArchiver.Write("report1",false);
    //2)规则名称，时间，是否立即希尔
    Env.Current.RowReportArchiver.WriteArchiver("report1",DateTime.Now,true);
    //3)规则名称，单个变量对象，时间
    Env.Current.RowReportArchiver.WriteChannel("report1",channel,DateTime.Now);
}
```

Javascript 执行代码如下：



1.7 关系库同步



当前支持 SQLServer、MySQL、PostgeSQL、Oracle，可以同时配置多个归档数据同时存储。

扩展模式和普通模式的区别：

1）普通模式需要先创建表，导入选择的变量，运行时不能自动同步新加变量

2）扩展模式运行时自动创建表，自动导入全部变量，运行时添加和删除的变量也会同步添加和删除

3）表格的格式一样，扩展模式多了一些信息

4）扩展模式运行前会先进行清库操作，再重新导入变量，标准模式按已有的变量进行同步（MySQL 不执行清库操作，启动时执行判断更新和删除的变量）

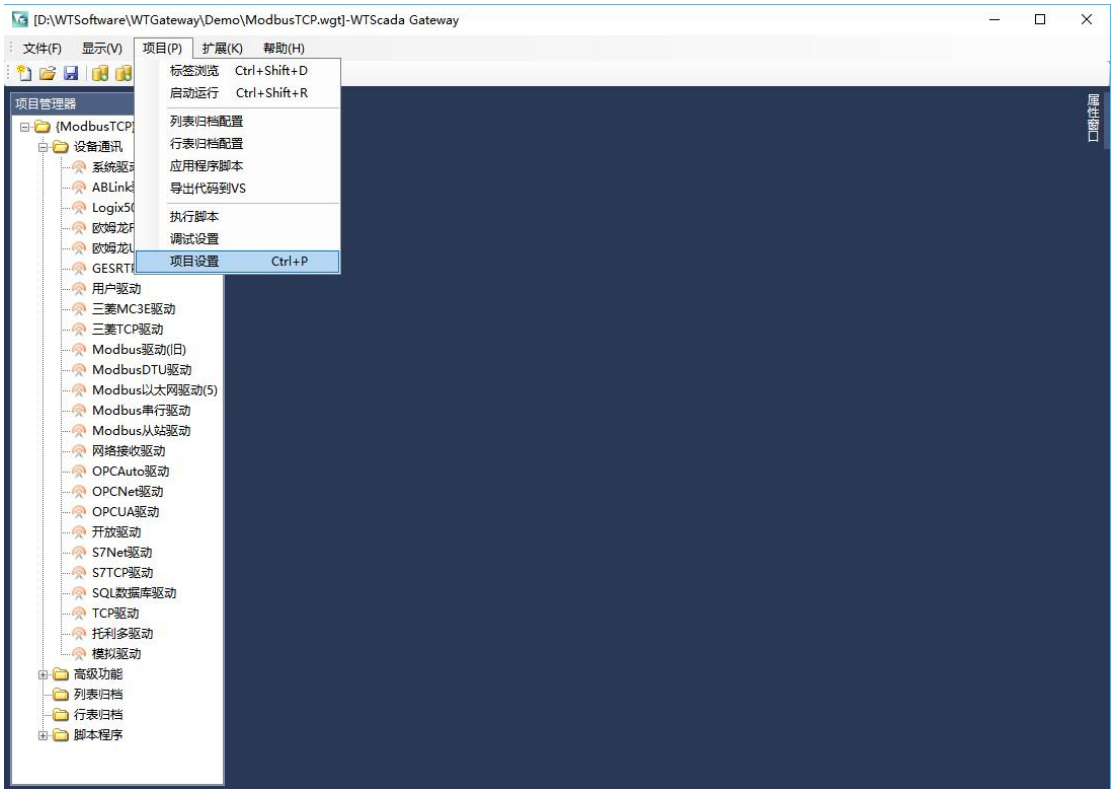
周期：归档线程运行周期

例外：变量不变化时间超过设置值也会更新一次，设置 0 时不启用

关系库同步使用 Update SQL 根据表的 ID 值更新关系库行表内容

超过 5000 个变量同步的情况下，建议使用数据库的内存表。

1.8 项目设置



项目名称：运行后显示在软件标题上的名称

加载脚本：C#函数，项目读取完成运行前执行，可以设置变量状态，创建变量等操作

驱动脚本：C#函数，项目运行后执行

停止脚本：C#函数，项目停止时执行

用户脚本：编译产生的 DLL 文件，如果选择了用户脚本 DLL，则项目运行时不再编译 C#代码，项目的 C#代码就没用了，可以删除，用于产品代码保护。

项目安全：设置打开项目的用户认证

UDP 数据服务端口：提供 UPD 数据服务的端口

时钟同步端口：提供时钟同步的 UDP 端口

对时模式：服务器、客户端、全功能模式

时钟服务器地址：时钟服务器的地址（WTGateway 或者 FScada Server）

小时偏差：设置时间偏差

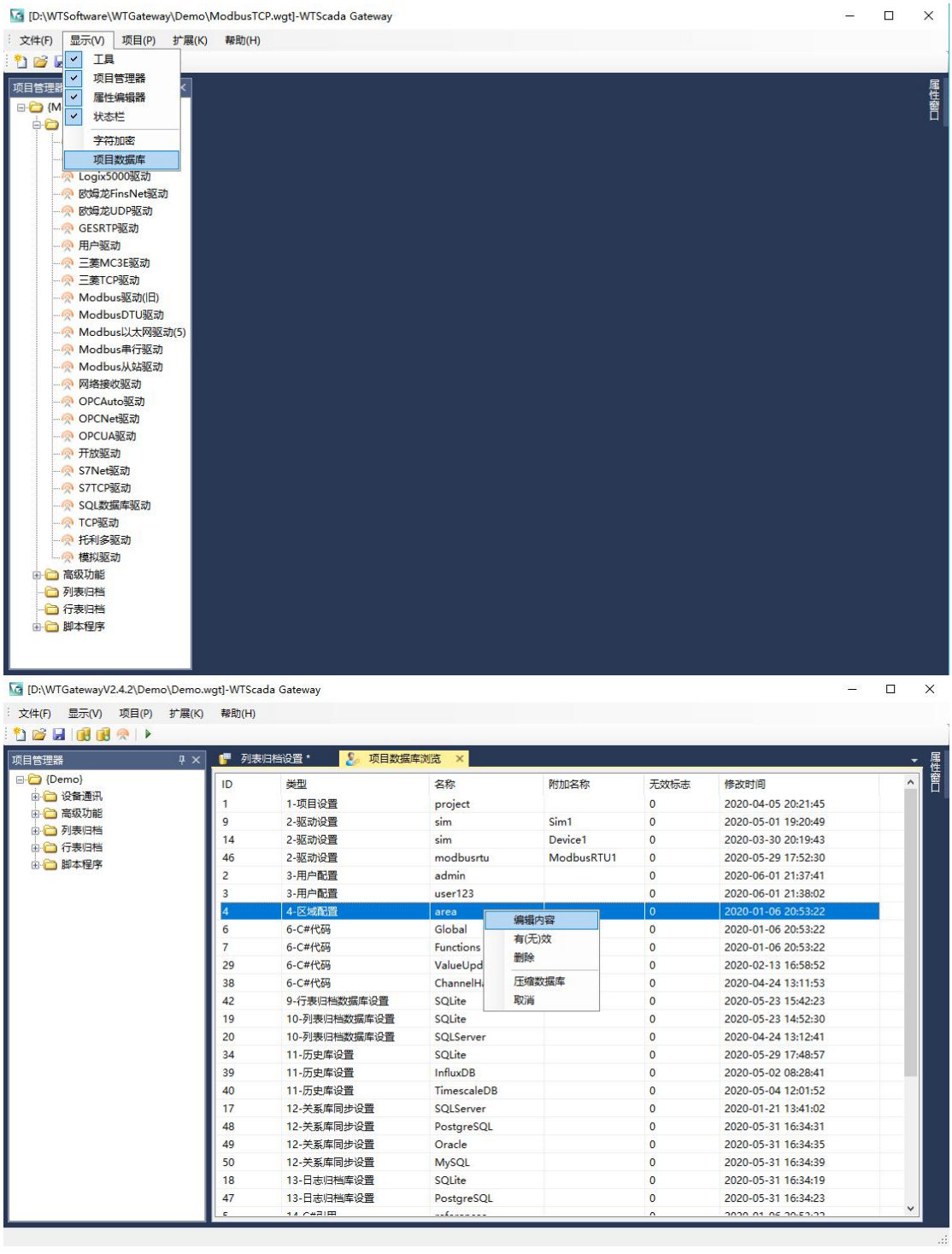
网络 TCP：提供 TCP 数据服务，通过 TCP 驱动采集或者 WTOPCServer 采集提供数据服务。

数据服务：历史数据服务 TCP 端口。

报警服务：远程 TCP 报警服务 TCP 端口。

IOServer 服务：IOServer 服务，提供给 WTScada HTML5、SDK、FScada 使用，提供同步实时数据功能。

1.9 项目数据库浏览



提供直接在数据库上管理项目的功能

通过设置无效标志可以禁止软件加载配置

提供了直接修复存储内容编辑的功能（***慎重修改，修复后需要重新加载项目***）

提供了压缩数据库的功能（因为数据库删除内容后空间并不会回收，如果加入了几万个变量，项目文件会很大，删除变量后项目文件不会自动减少，压缩后才能

恢复）。

1.10 TCP、UDP 转发扩展

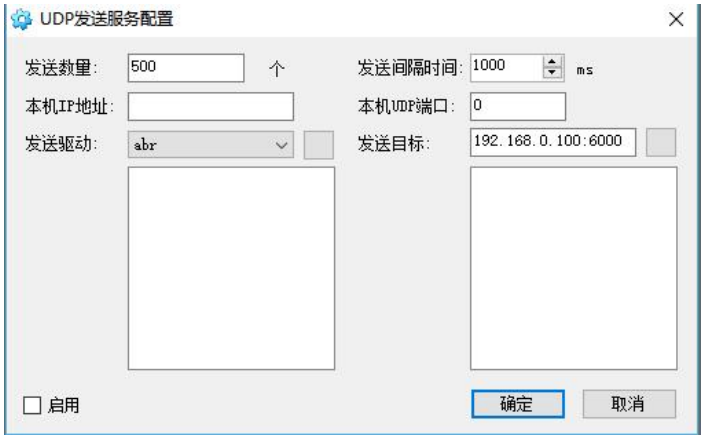
本机地址：一般空白就可以，如果填了就表示要指定通讯网卡，该 IP 地址必须存在

发送驱动：选择某些驱动，不选表示全部驱动

发送目标：接收方的 IP 地址和端口（接收方为 FScada Server 的网络接收驱动、WTGatewy 的网络接收驱动、HTML5 的 IOServer 接口）

TCP 发送连接建立后会先发送一次变量信息表，运行中添加的变量也会同步发送。

历史补录功能：启用配置后当 TCP 发送失败，会在本地存储发送的数据，网络恢复后发送把存储的数据作为历史数据发送出去，该功能仅能配合网络接收驱动使用。



本机地址：一般空白就可以，如果填了就表示要指定通讯网卡，该 IP 地址必须存在

本机 UDP 端口：0 表示随机端口

发送驱动：选择某些驱动，不选表示全部驱动

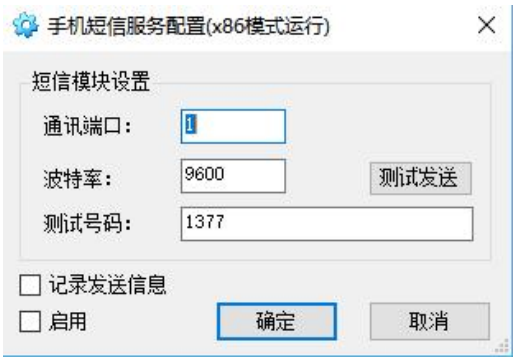
发送目标：接收方的 IP 地址和端口（接收方为 FScada Server 的网络接收驱动、WTGatewy 的网络接收驱动、HTML5 的 IOServer 接口）

发送数量：由于 UDP 包被限制为 64K，因此每次通讯包的尺寸就收到了限制，另外有效防火墙也会限制 UDP 包尺寸，一般建议的设置是 500-2000，根据项目点数确定。

UDP 发送时没有变化的数据 1 分钟发送一次，变化的数据按设定的发送间隔发送。

UDP 发送运行时会发送变量信息表，运行中添加的变量也会同步发送，但是对方接收程序如果没有运行就会收不到变量信息表，因此要同步变量运行前保证对侧软件已经运行。

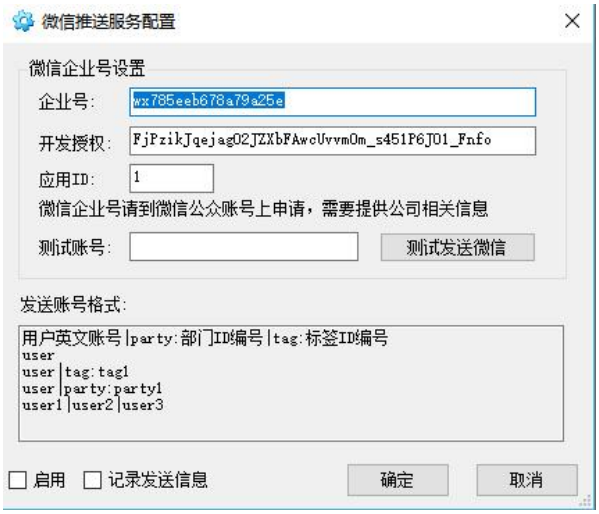
1.11 短信发送扩展接口



该接口仅提供信息发送服务，可以通过 C#脚本或者 js 脚本发送信息。

C#代码：Env.Current.Application.ExtendSendCommand("SMS","13813666181","sms 短信");

1.12 微信发送扩展接口

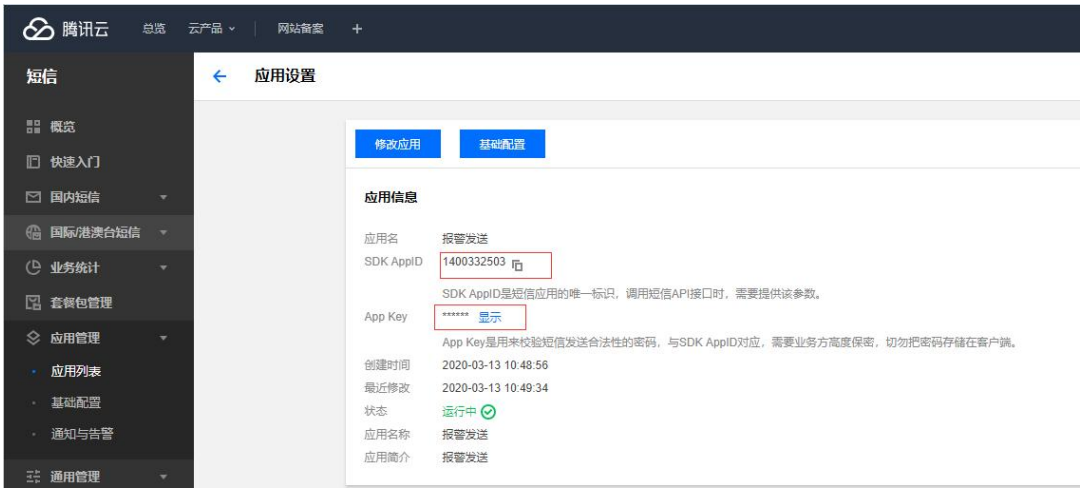


该接口仅提供信息发送服务，可以通过 C#脚本或者 js 脚本发送信息。。

C#代码：Env.Current.Application.ExtendSendCommand("WxMessage","user","微信消息");

1.13 腾讯云短信服务器扩展

配置信息从腾讯云购买短信服务，创建应用，签名，模板之后可以获取到设置。



短信模板必须包括 4 个分段

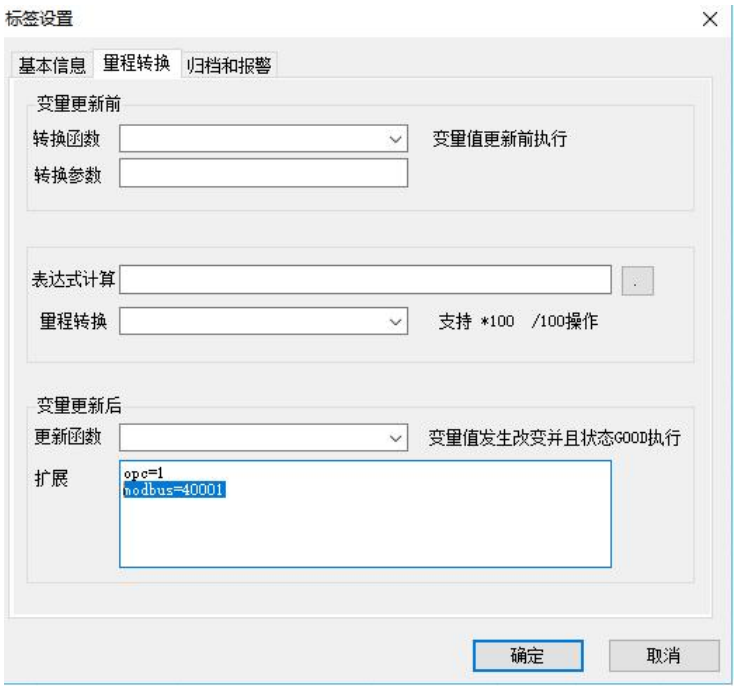
第 1 个是时间，后面三个可以自定义，例如如下的格式

ID	模板名称	内容
553536	报警信息	时间:{1} 变量:{2} 类型:{3} 当前值:{4}

1.14 Modbus 服务器扩展

Modbus 服务器同时支持 RTU 和 TCP 服务

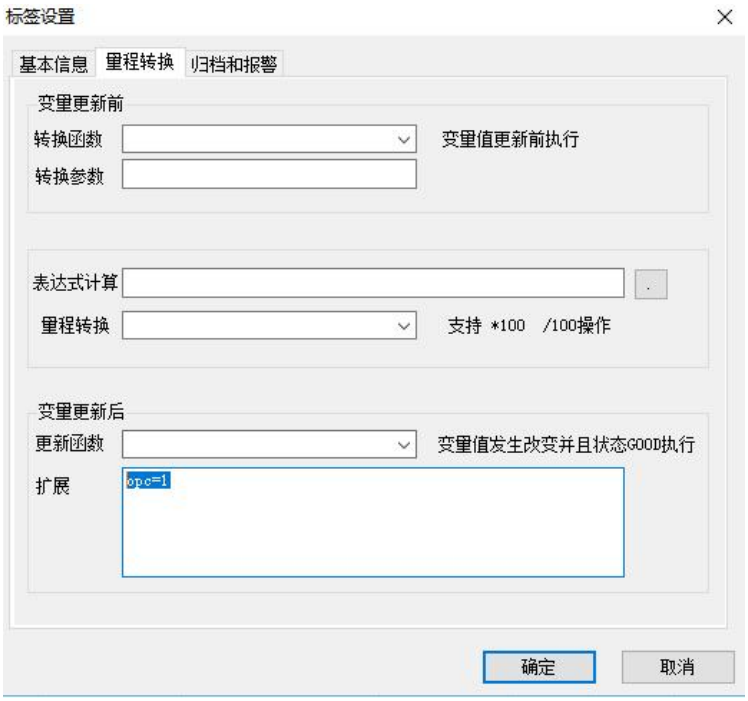
Modbus 服务器只加入已经配置了 modbus 输出的变量，不支持运行中加入变量，配置方式如下：



扩展设置里面有 1 行的内容为 modbus=寄存器地址（40001、00001）

1.15 OPCUA 服务器扩展

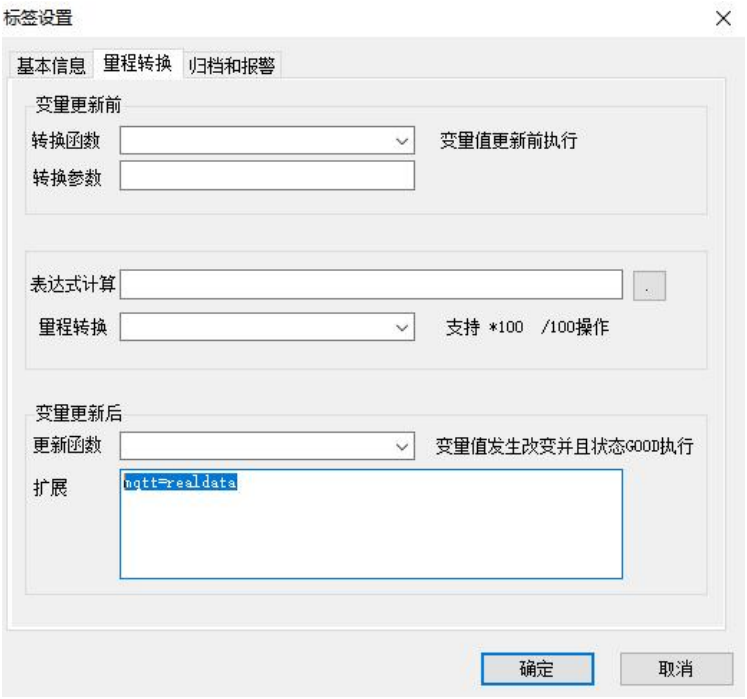
OPUA 服务器只加入已经配置了 opc 输出的变量，不支持运行中加入变量配置方式如下：



扩展设置里面有 1 行的内容为 opc=1

1.16 MQTT 发送扩展

在变量上的扩展配置内写入 mqtt=xxx，xxx 为发布的 Topic 名称



JSON 数组数据格式，和 Kepware 数据格式兼容,JSON 数据格式如下：

{“id”:tagname,”v”:value,”q”:true,”t”=1234567}

id:变量名，v:变量值，q:变量状态,t:unix 时间

支持订阅写入，JSON 数组格式，JSON 数据格式如下：

{“id”:tagname,”v”:value}

id:变量名，v:变量值

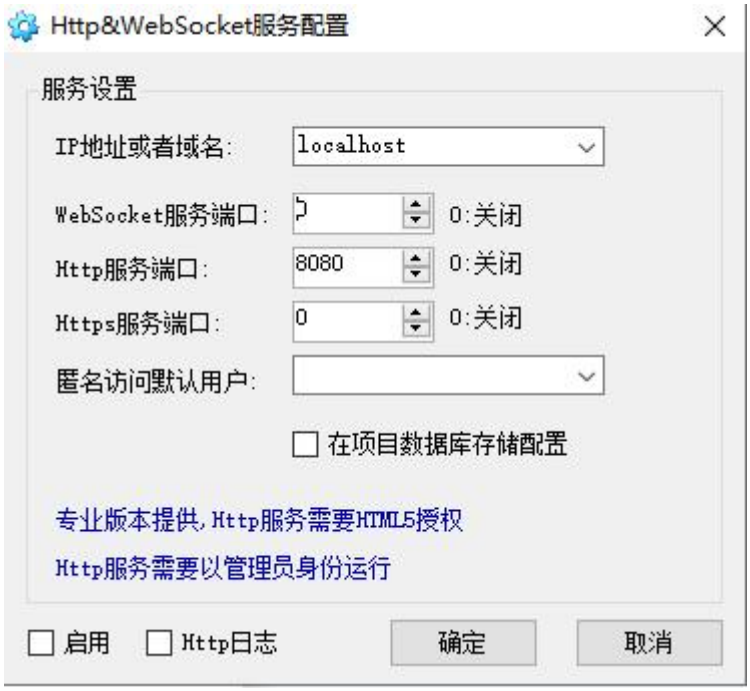
该扩展提供了 C#代码发送功能，参考 DEMO 目录下的“MQTT 脚本发布.wgt”

调用的函数是

Env.Current.Application.ExtendSendCommand("MQTTSender","realdata","{\"tag\":1}
");

第一个参数是扩展名称，第二个参数是发布的主题名称，第三个参数是发送的文本内容，可以在 js 脚本中使用。

1.17 HTML5 Web 服务



提供 WebSocket 服务和 HTTP 服务，HTTP 服务用于提供 HTML5 网页组态功能

使用 HTTP 服务时程序需要以管理员方式启动。

websocket.html 提供了 WebSocket 功能演示，如 WebSocket 端口设置为 6012，这 WebSocket 的访问方式为:ws://localhost:6012。

在 Windows 环境下 WebSocket 和 HTTP 中可以调用 C#脚本，参考 WebSocket.wtg 项目。

IP 地址：如果要被本机之外的机器访问到，IP 地址或者域名必须填写非 127.0.0.1 和 localhost，WebSocket 服务端口和 Http 服务端口根据需要自行修改，不能相同，修改好需要开启网络防火墙。

匿名访问用户：如果设置了用户就相当于不需要登录就可以访问。

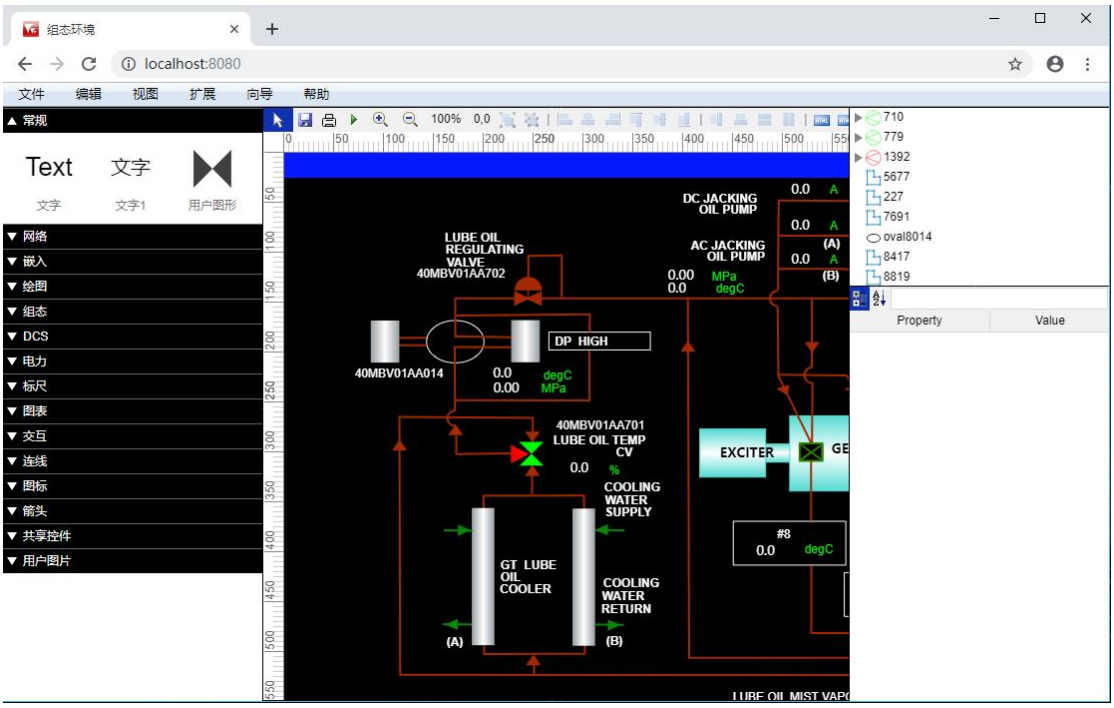
Http 日志：在 Logs\http 下按天记录 http 访问日志

默认情况下 HTML5 网页组态系统的项目配置文件存储在 Config 目录下的 html5.db 中，如果勾选了“在项目数据库存储配置”则 HTML5 网页组态配置存储在项目文件中。

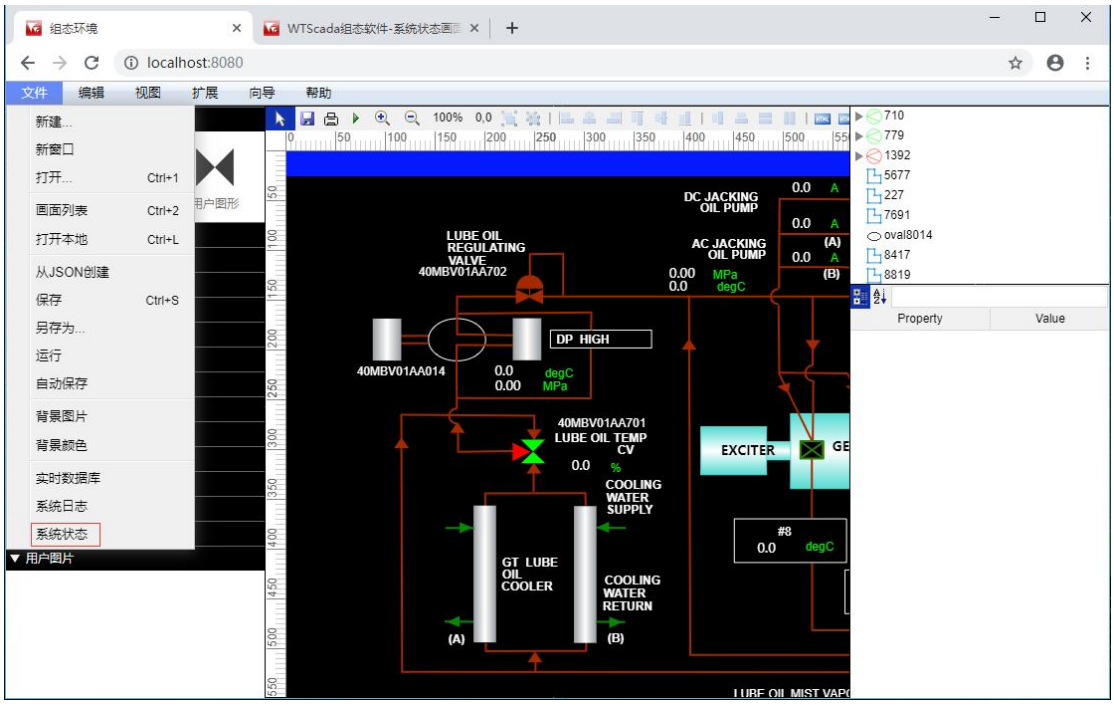
WebSocket 交互数据格式见附录。

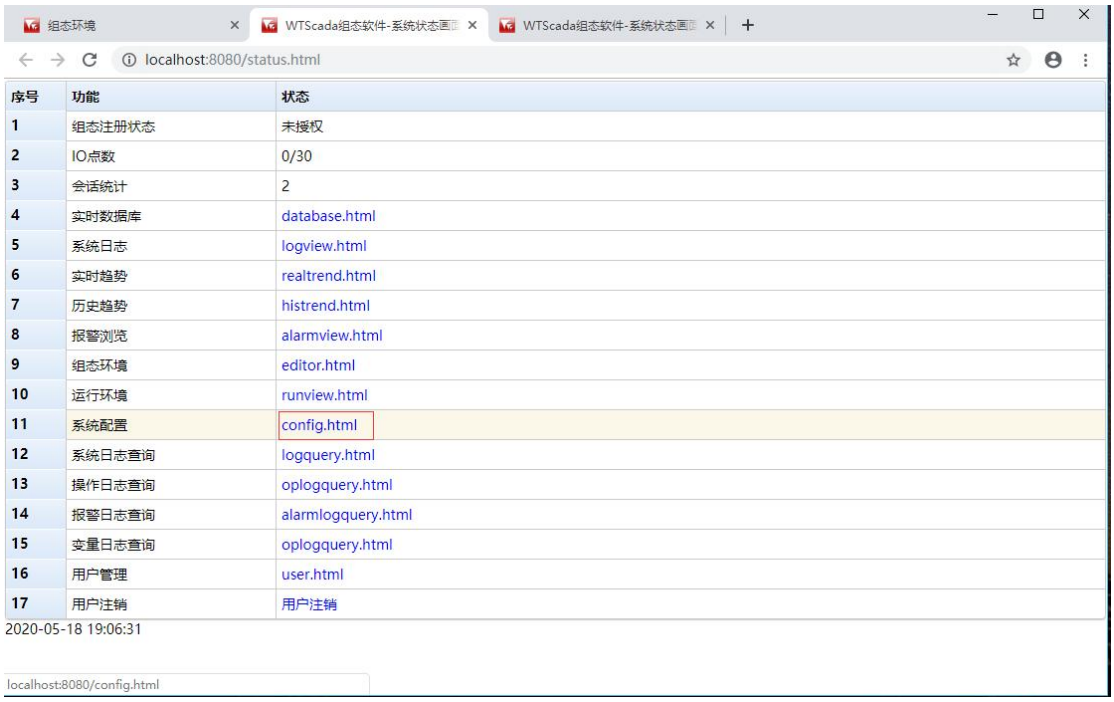
运行后如果视图菜单下有“HTML5 网页”菜单，表示网页组态功能已经启动，点击菜单会打开浏览器。

进入 HTML5 画面组态环境

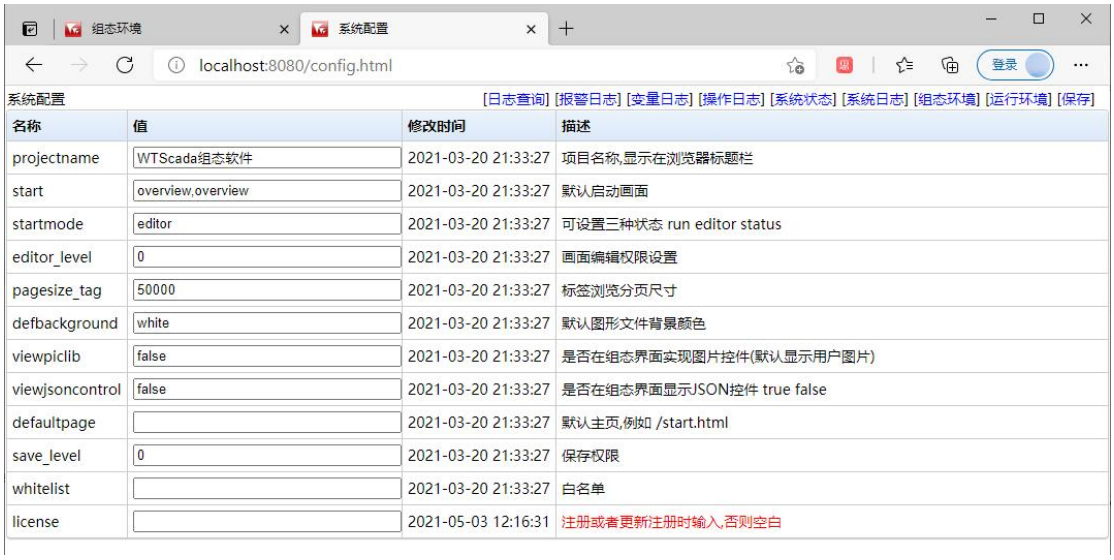


进入文件菜单下的系统配置，修改 HTML5 系统的启动状态和系统设置





序号	功能	状态
1	组态注册状态	未授权
2	IO点数	0/30
3	会话统计	2
4	实时数据库	database.html
5	系统日志	logview.html
6	实时趋势	realtrend.html
7	历史趋势	histrend.html
8	报警浏览	alarmview.html
9	组态环境	editor.html
10	运行环境	runview.html
11	系统配置	config.html
12	系统日志查询	logquery.html
13	操作日志查询	oplogquery.html
14	报警日志查询	alarmlogquery.html
15	变量日志查询	oplogquery.html
16	用户管理	user.html
17	用户注销	用户注销



这里提供了软件显示名称修改，启动画面编辑，运行模式编辑等配置，修改完成后点击保存。

projectname:项目名称，根据需要进行修改

start:默认启动画面名称，逗号分隔是电脑和手机版本显示的画面，访问 runview.html 不附带画面文件名称时会读取这个设置显示画面。

startmode:访问网站根目录进入的模式，默认是编辑，通常应该设置为 run，进去运行状态。

whitelevel:免登录的 IP 地址列表，用逗号进行分隔，免登录用户使用 user 用户权限（必须保证有 user 用户存在）

license：通过 web 端进行软件授权注册时可以输入授权码。

注意：

1）http 目录下 websocket.html 文件是一个 websocket 演示文件，正式运行前为了安全请删除该文件，或修改为自己想要的页面内容。

2）系统发布后如果修改了 WebSocket 端口，已经访问过系统的 web 客户端需要清除浏览器缓存（Ctrl+F5），否则某些功能无法正常使用。

3）白名单功能可以设置某些 IP 地址访问使用 user 用户自动登陆。

以 IP 地址 192.168.1.100 的计算机设置端口为 8080 为例：

http://192.168.1.100:8080 根据 start 设置进入组态环境或者运行环境

http://192.168.1.100:8080/status.html 进入系统状态页面

http://192.168.1.100:8080/editor.html 进入组态编辑界面

http://192.168.1.100:8080/editor.html?filename=xxx 进入组态编辑界面打开 xxx

http://192.168.1.100:8080/runview.html 进入运行界面打开默认画面

http://192.168.1.100:8080/runview.html?filename=xxx 进入运行界面显示 xxx

http://192.168.1.100:8080/config.html 进入系统配置界面（管理员才可以修改）

http://192.168.1.100:8080/histrend.html 进入历史趋势界面

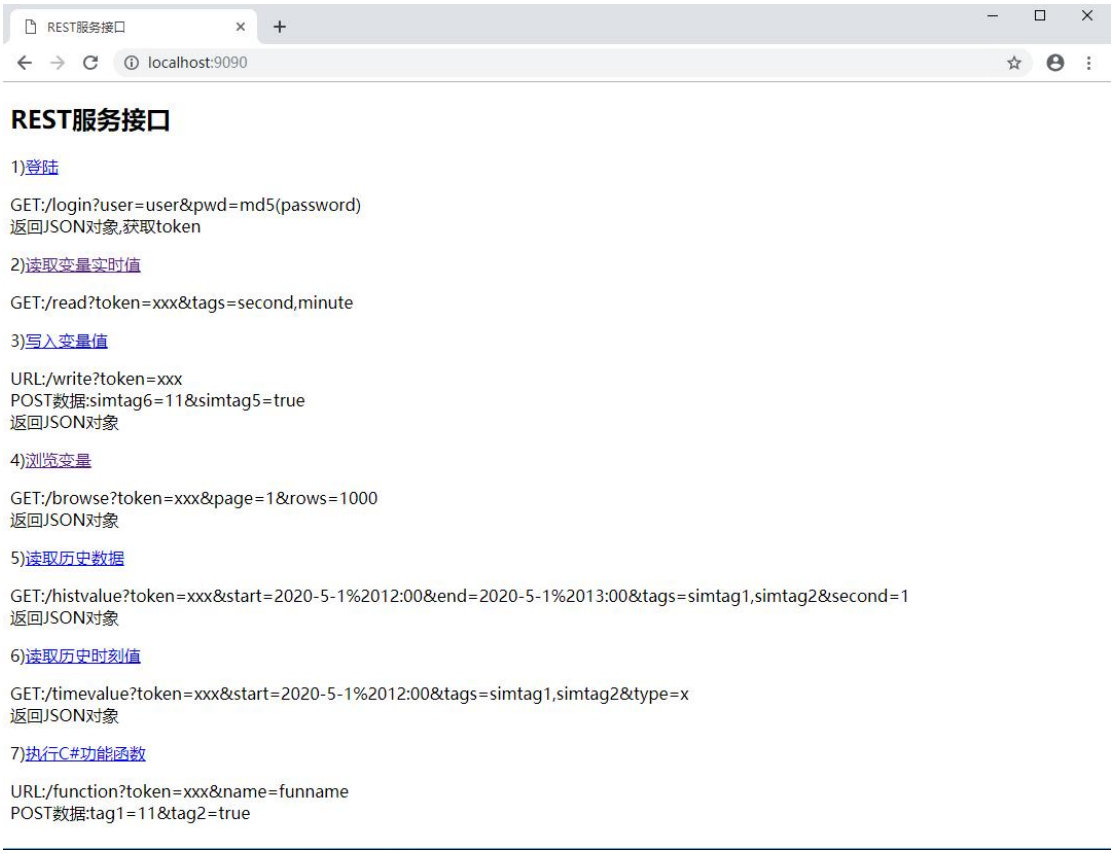
http://192.168.1.100:8080/realtrend.html 进入实时趋势界面

Web 函数手册见附件。

1.18 REST WebAPI 服务

REST 服务提供了 1 个简单的 web api 服务，用于给第三方提供在浏览器方式的数据访问，提供了变量实时数据读取、变量枚举、历史数据读取、变量写入、自定义 c#函数访问等功能。

上述配置下：浏览器输入 http://localhost:9090 可以访问系统默认页面，该页面提供了 api 使用方法



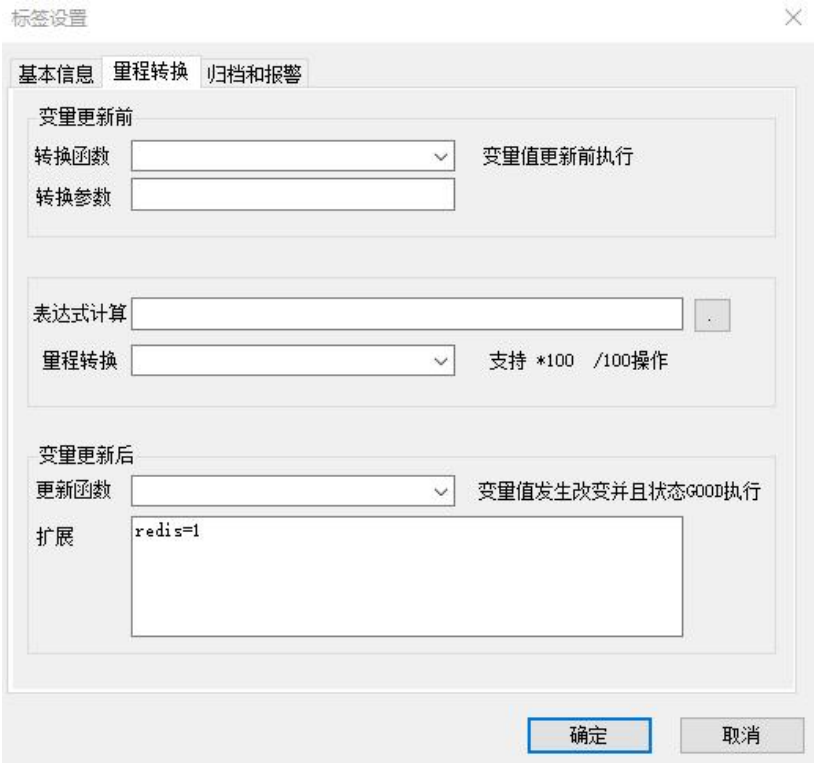
如果要允许匿名访问，匿名访问用户名选择框内选择一个默认用户，否则所有 api 访问都需要验证 token。

1.19 Redis 扩展

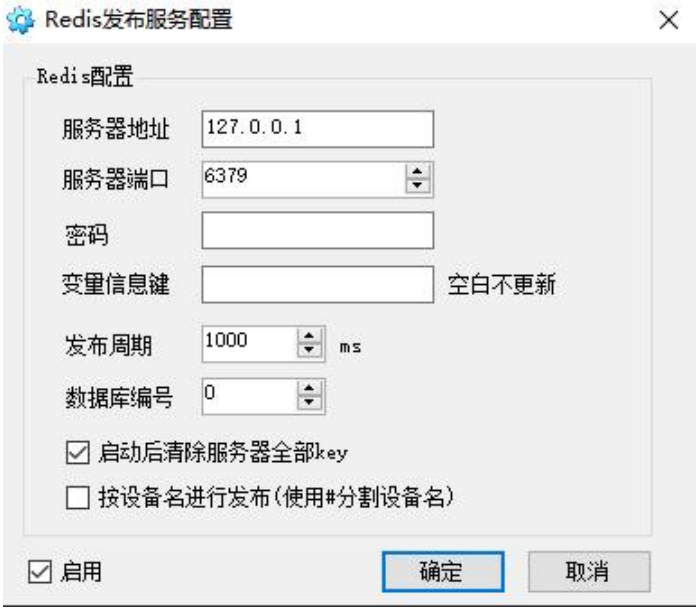
Redis 扩展的功能是把组态变量的信息实时同步到 Redis 内存数据库中，根据设置可以把变量信息同步到一个 Hash key 中，也可以把一个设备的信息同步到 1 个 Key 中。

使用 Redis 服务器首先要有 1 个 Redis 服务器，如果没有可以下载 Redis for Windows 安装包进行安装。

在标签设置界面上对需要的变量启用 redis 转发，扩展中包含一行 redis=1



Redis 发布服务配置：



变量信息 key：如果非空白则把变量信息写入设置的 Hash 键中，内容是 Json 文本，键值运行中不更新。

数据库编号：使用的数据库 ID 号

启动后清除服务器全部 key:如果设置了启动运行和停止运行时会清除服务器指定数据库的全部 Key。

默认情况下每个变量都会在 Redis 中存储个 Key 值，内容是 Json 格式文本，使用

set 操作。

按设备名进行发布：勾选后将会把发布的变量按设备名称进行分类更新到对应的 Key 中，每个设备 1 个 Key 值，实时更新，使用 set 操作和 publish 操作（可以被订阅到）。

支持的脚本命令：

参考 Demo 目录下 Redis.wgt 项目

在 C#代码中的执行方式如下：

IExtend ext = Env.Current.ExtendDlls[extName];

if (ext != null)　　ext.DoComamnd(command, message);

command 参数：setkey 键名称　　设置键值

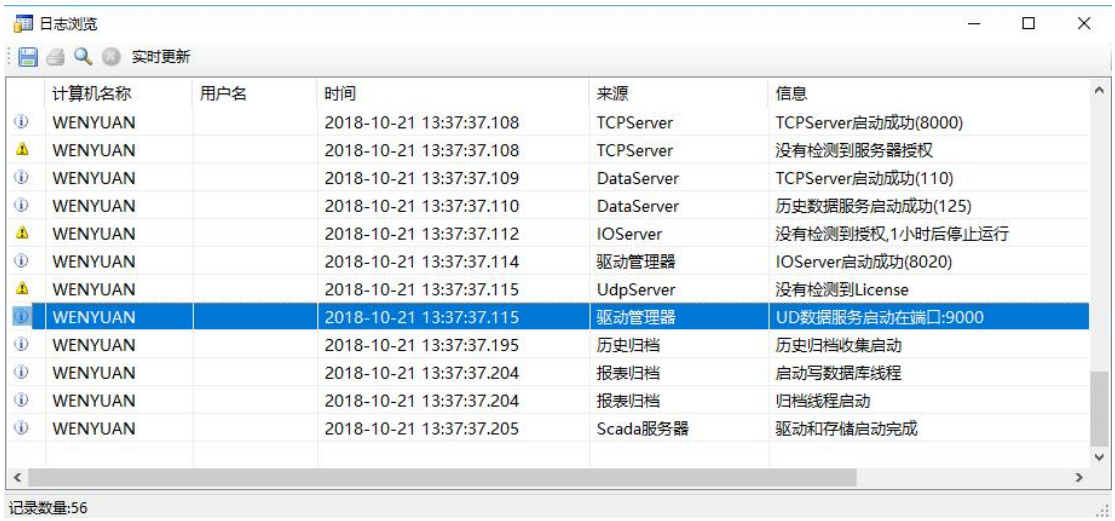
　　publish 键名称　设置并发布订阅的键名称

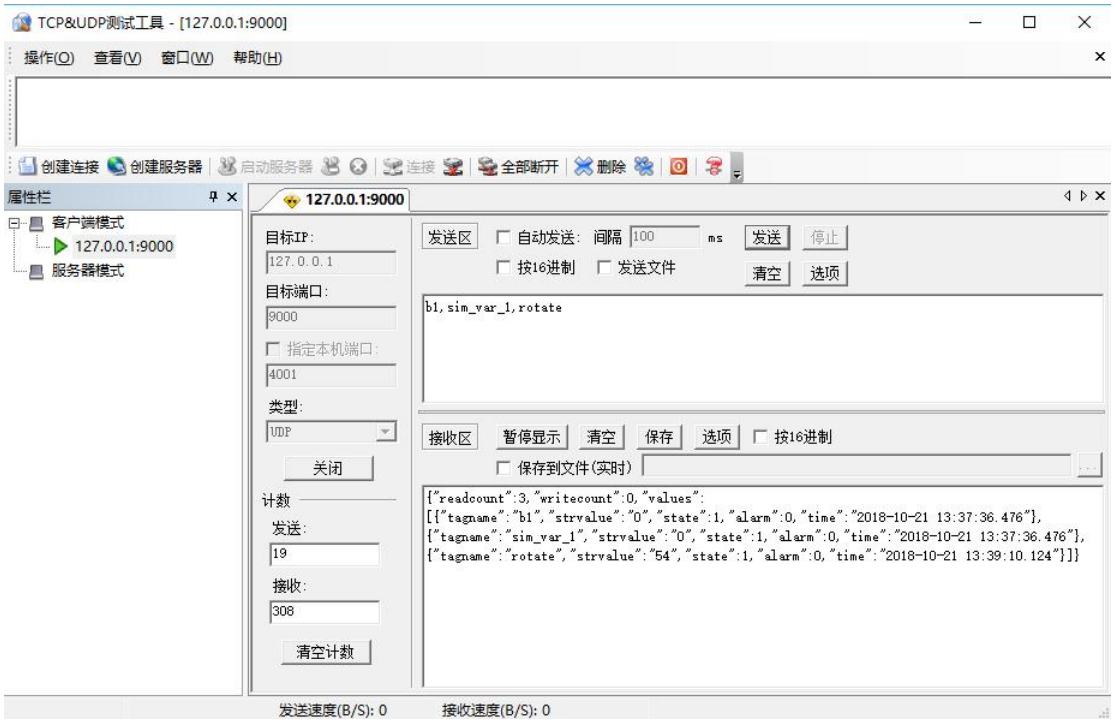
message 参数：发送的文本信息

1.20 UDP 数据服务

在项目配置界面可以配置 UDP 数据服务端口，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。





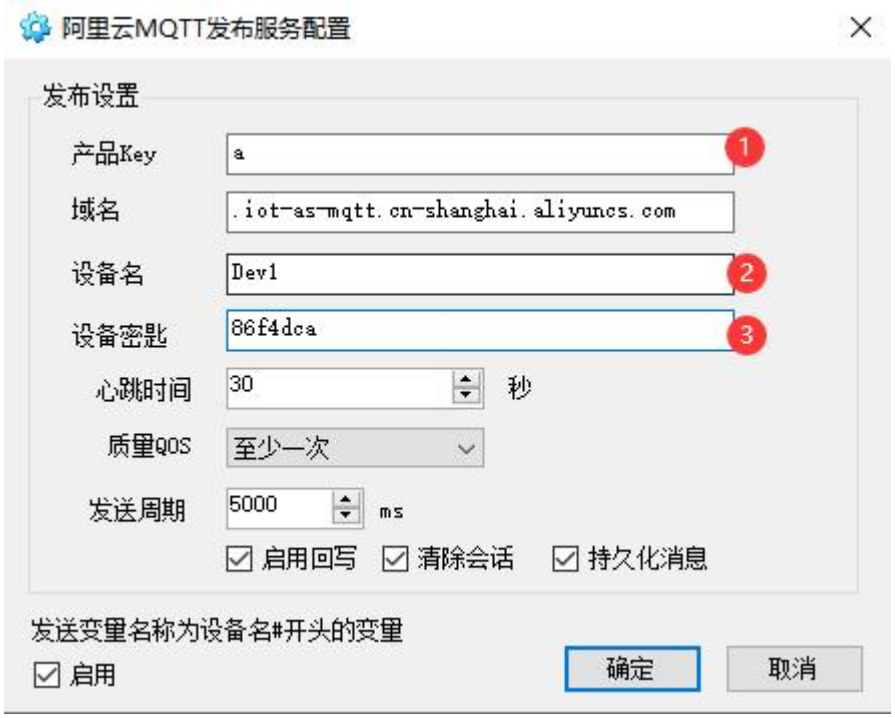
直接发送变量名称,多个变量名称直接使用逗号分割

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,因此可能会有多个数据包返回(以 20K string 长度,实际字节长度在 20K-60K 之间,UTF-8 字符编码)

1.21 阿里云物联转发



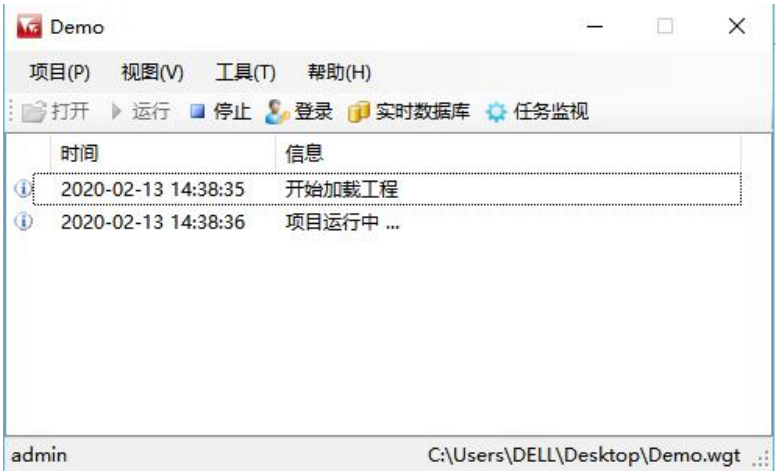
输入阿里云平台设备三要素，如果需要写入点击“启用回写”选项。

发送的变量名称为设备名#开头的变量，上列将发送 Dev1#开头的全部变量到云平台，发送时变量名只使用#后面的内容。

如有 2 个变量 Dev1#Current 和 Dev1#Voltage，发送的变量是 Current 和 Voltage。

1 个扩展只能发送 1 个设备，如果有多个设备需要发送，则需要开启多设备扩展配置，方法是复制 1 个 Extend.AliyunSender.dll，改名 Extend.AliyunSender1.dll，再启动配置程序就能看到第 2 个阿里云扩展配置。

2. 运行环境



2.1 实时数据库

增强的查询筛选功能

默认只显示有变量的驱动

点击第一个“显示全部”按钮，则会显示全部驱动

更新时间和更新次数的意义：

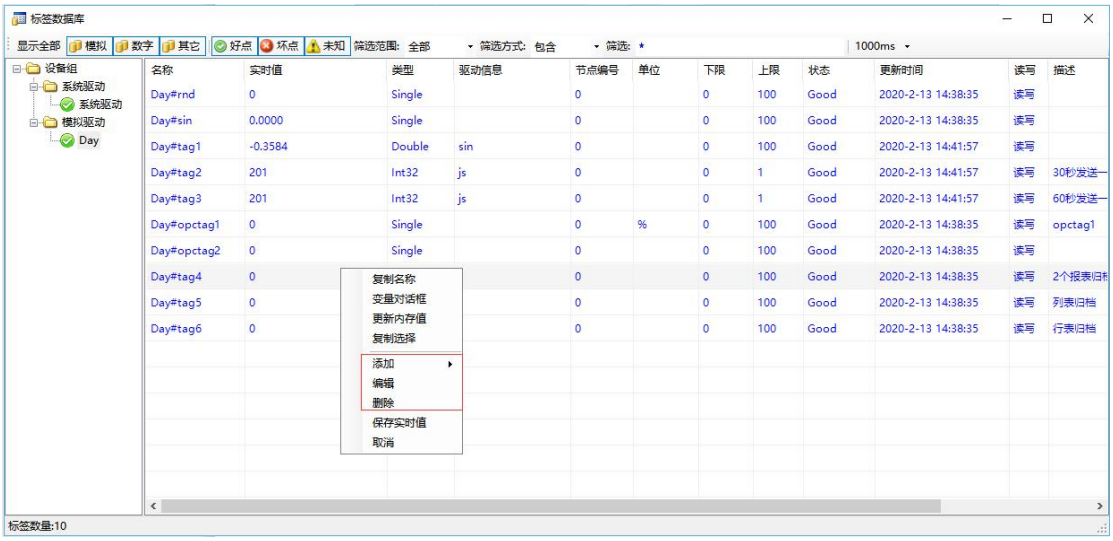
更新时间：当变量的值发生改变时时间发生改变，当变量启用了“变量时间”后每次值更新函数被调用都会更新时间。

更新次数：每次值更新函数被调用，更新该值，更新的频率取决于驱动的采集周期。

当具备管理员权限后，就可以对驱动进行重新加载，启停，在线添加、删除、修改变量、修改设备 JavaScript 脚本。

单个设备的启停和重新加载。

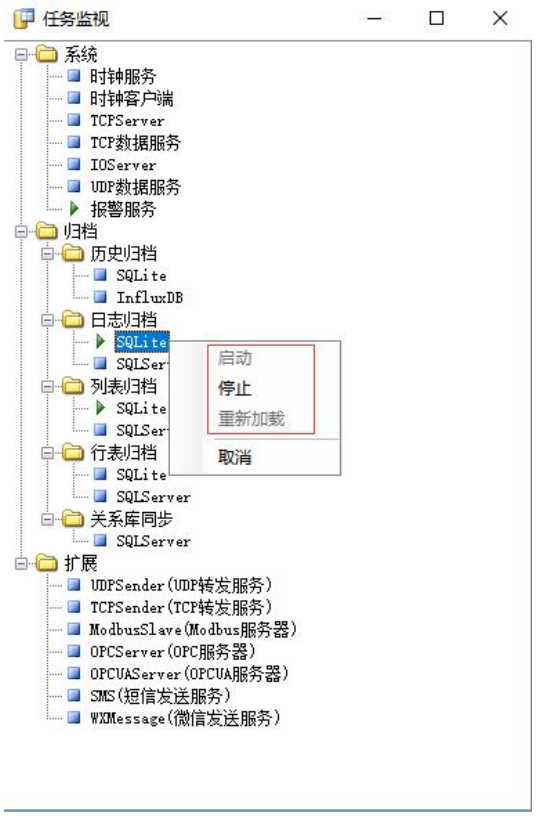
运行时变量添加、删除和修改。



运行时修改循环执行脚本：

在线修改驱动变量后要使用文件菜单下的“保存”菜单保存到项目中，否则下次启动修改就没有了。

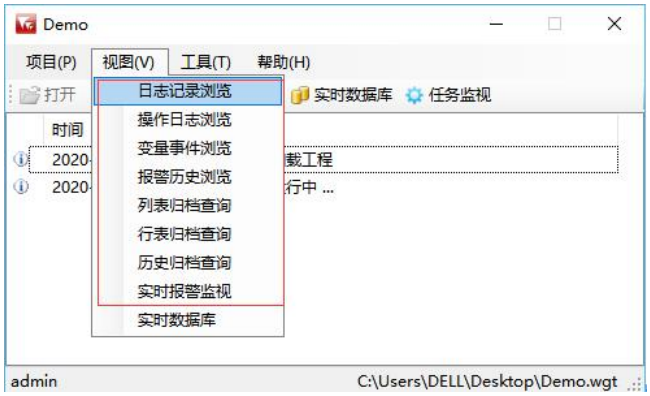
2.2 任务监视

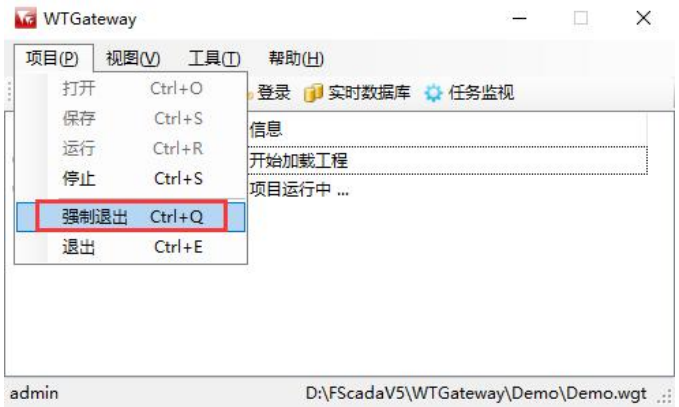


可以对各任务重新加载、启停控制，绿色三角表示运行状态，鼠标右键“启动”菜单为灰表示未启动，在组态配置环境启用后，点击重新加载就可以启用。

列表归档和行表归档归档规则改变后需要再目录上点击停止，重新加载，再点运行。

2.3 数据查询

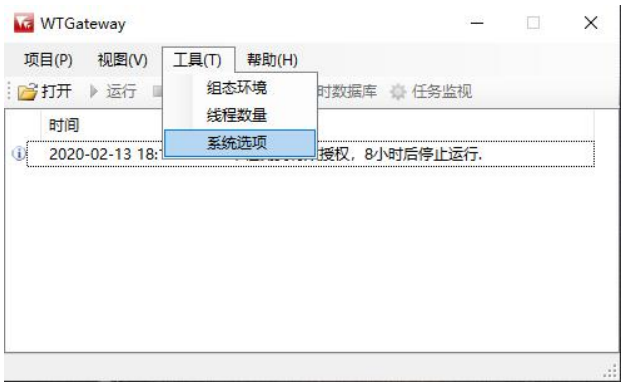




当系统由于发生异常无法正常终止时使用“强制退出”可以立即终止进程。

软件退出运行必须使用项目菜单的退出或者强制退出，窗口的关闭按钮只会让软件最小化到工具栏。

2.4 系统选项

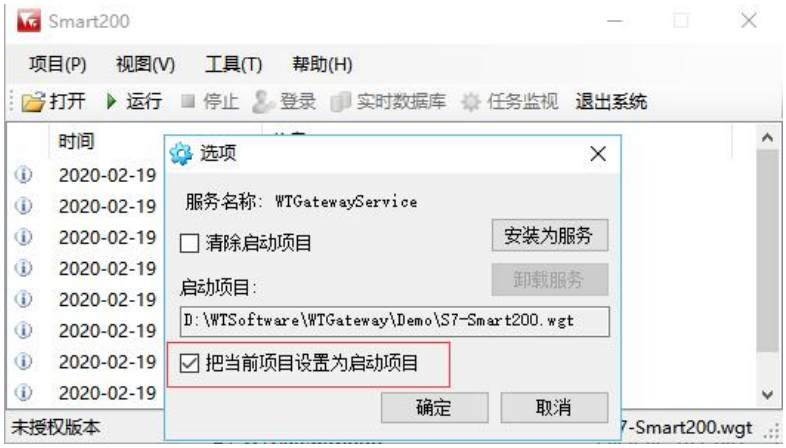


设置自启动项目的方法：

1）运行 Runtime.exe

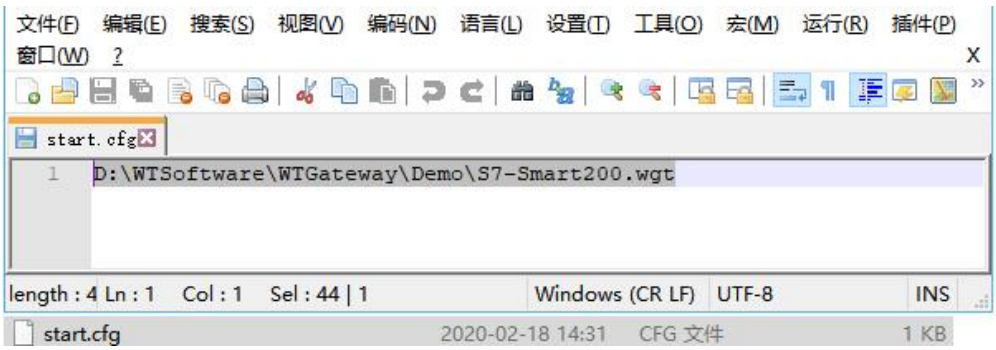
2）打开 1 个项目配置

3）点击工具菜单下的“系统选项”

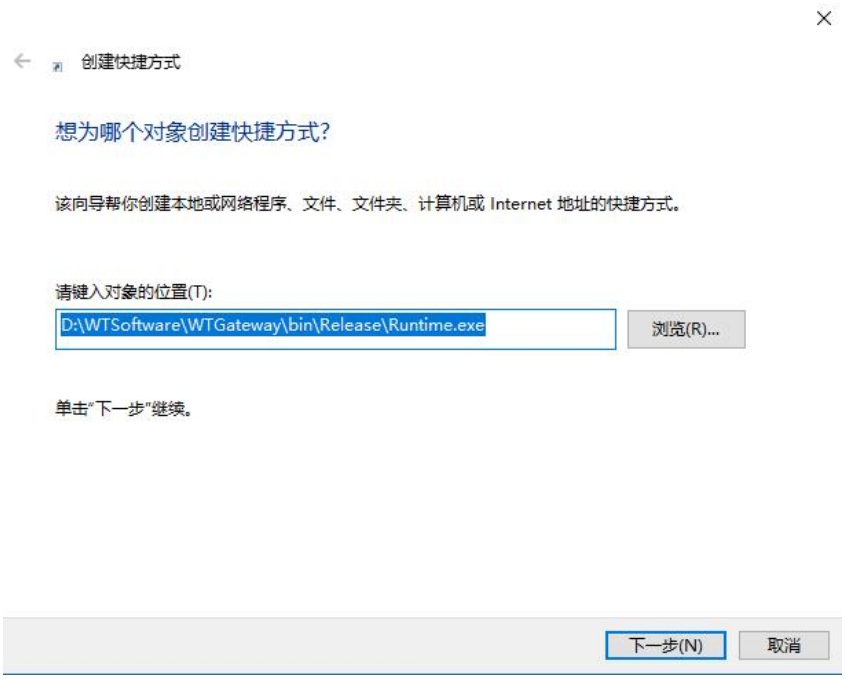


选中“把当前项目设置为启动项目”，点击确定，再次运行软件时就自动进入运行模式。

其原理是在软件目录下建立了 1 个 start.cfg 文件，该文件是 1 个文本文件，里面包含了一行项目文件名称。



4）也可以通过命令行参数启动项目，或创建 1 个快捷方式



D:\WTGateway\Runtime.exe D:\WTGateway\Demo\Demo.wgt

在 exe 路径后面加 1 个空格然后输入项目文件命令

命令行方式的优先级别高于启动项目“start.cfg”

5）也可以把 wgt 项目文件改名为 start.wgt 拷贝到软件根目录下，也能自动启动

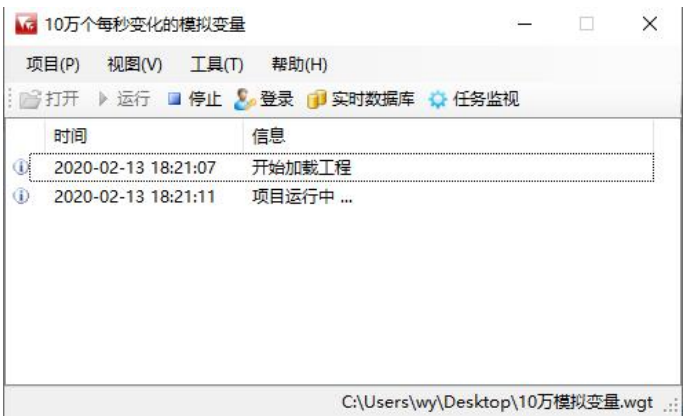
6）也可以把 wgs 项目文件改名为 start.wgs 拷贝到软件根目录下，也能自动启动

7）自启动优先级如下：命令行参数、start.wgt、start.cfg

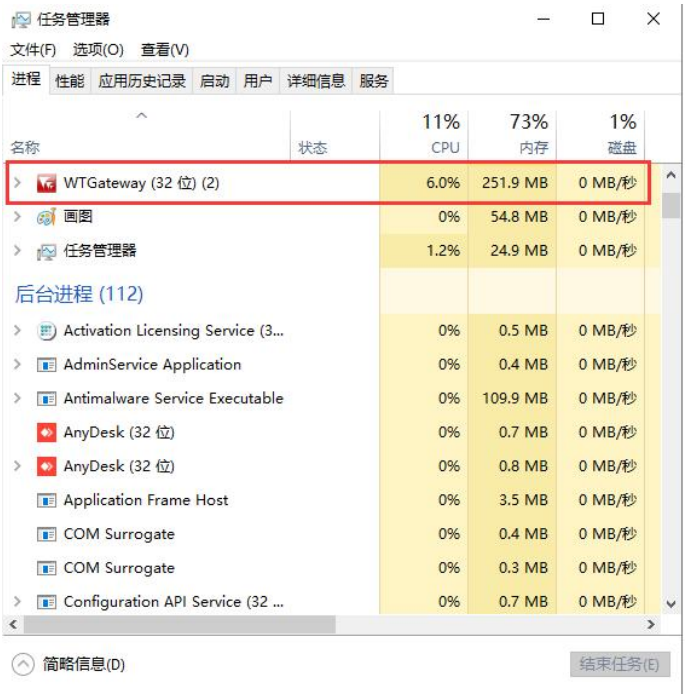
如果安装为服务，必须配置好启动项目。

服务模式为 windows 系统后台服务，运行后没有界面，用于无人值守的系统，跟随操作系统启动而启动。

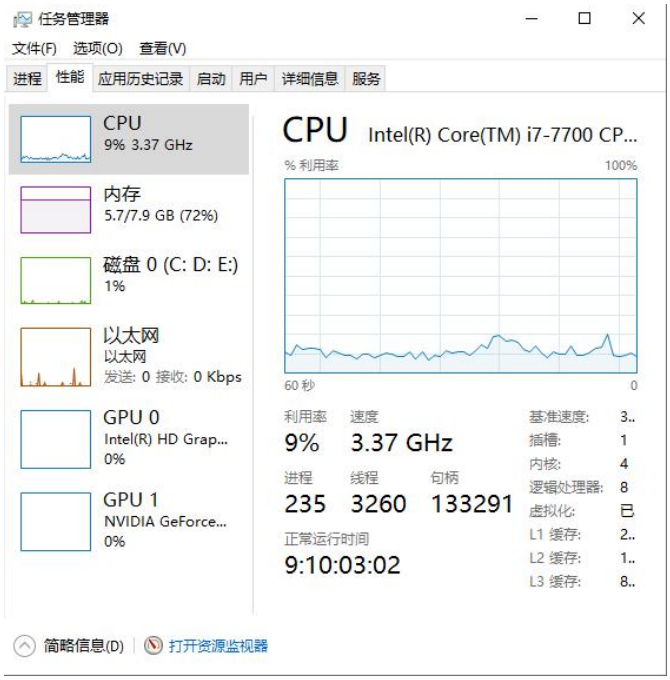
2.5 系统性能



10 万个每秒变化的模拟变量，每个设备 1 万个变量，每秒更新一次值（实际的项目中即使 10 万变量每秒变化的一般不会超过 20%）



CPU 使用率 5%-6%左右，内存使用 250M（测试电脑 i7-7700 CPU 8 核心，8G 内存，Windows10 家庭版本）。


任务管理器
文件(F) 选项(O) 查看(V)
进程 性能 应用历史记录 启动 用户 详细信息 服务
CPU
9% 3.37 GHz
内存
5.7/7.9 GB (72%)
磁盘 0 (C: D: E:)
1%
以太网
以太网
发送: 0 接收: 0 Kbps
GPU 0
Intel(R) HD Grap...
0%
GPU 1
NVIDIA GeForce...
0%
CPU Intel(R) Core(TM) i7-7700 CP...
% 利用率 100%
60 秒 0
利用率 速度
9% 3.37 GHz
进程 线程 句柄
235 3260 133291
正常运行时间
9:10:03:02
基准速度: 3..
插槽: 1
内核: 4
逻辑处理器: 8
虚拟化: 已
L1 缓存: 2..
L2 缓存: 1..
L3 缓存: 8..
简略信息(D)
打开资源监视器

3. 授权模式

WTGateway 授权分标准版本，专业版本，企业版本三种模式

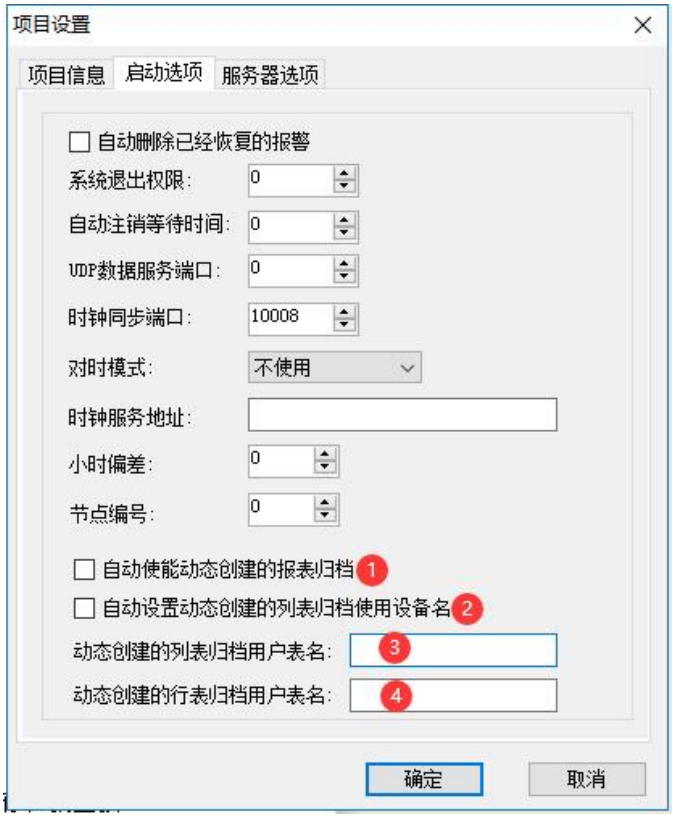
1）标准版本通常应用于 WTScada HTML5，提供驱动采集、历史归档、报表归档、转发接口、短信发送、MQTT 转发。

2）专业版本通常应用于 MES 系统，比标准版本增加了扩展和开放驱动、数据库驱动、关系库同步、SDK 通讯、IOServer、TCP Server、WebSocket、时钟服务、HTML5 网页组态。

3）企业版本通常应用于 SIS 等实时信息系统，比专业版本增加了唐码实时库、PI 实时库接口。

4. 附录

4.1 行表归档和列表归档配置说明



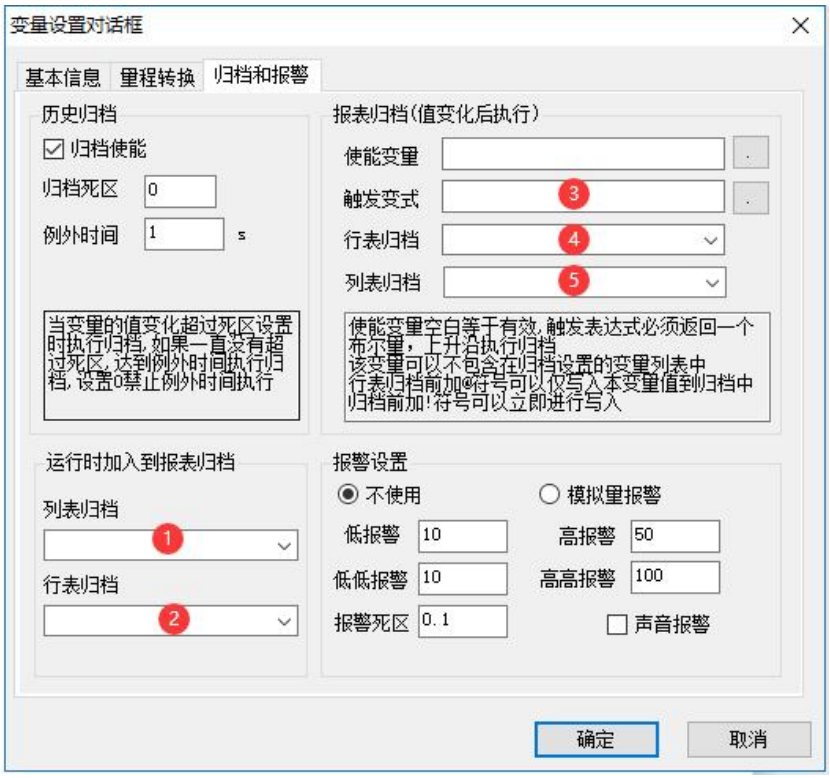
① 运行时自动创建的列表或者行表归档设置自动设置为启用状态

② 运行时自动创建的列表归档配置为使用设备名

③ 运行时自动创建的列表归档设置用户表名

④ 运行时自动创建的行表归档设置用户表名

自动创建的行表或者列表归档设置是指下图的①和②的设置



① 运行时自动把本变量加入到指定的列表归档配置中，如果列表归档配置已经存在就直接加入，如果不存在就创建 1 个（使用项目配置中的设置配置是否自动启用，是否使用设备名分割，是否设置用户表名）

② 运行时自动把本变量加入到指定的行表归档配置中，如果行表归档配置已经存在就直接加入，如果不存在就创建 1 个（使用项目配置中的设置配置是否自动启用，是否设置用户表名）

③ 当变量的值发生改变后执行触发表达式，如果表达式输出为一个布尔量，则表达式输出从 False 到 True 过程执行后续操作（单次执行）；如果表达式输出为其他值类型，如果表达式计算结果和上一次计算结果不一样就执行后续操作。

后续操作指 执行④ ⑤定义的归档操作。

④ 对设定的行表归档执行写入操作，如果以@符号开头，则对@符号后面的数据库表执行写入本变量的操作。

⑤ 对设定的列表归档执行写入操作，如果以!符号，则立即写入!符号后面的列表归档设置。

触发表达式说明：

1）触发表达式可以是 1 个值变量，如 [Second] ： 和上次的值不一样就执行

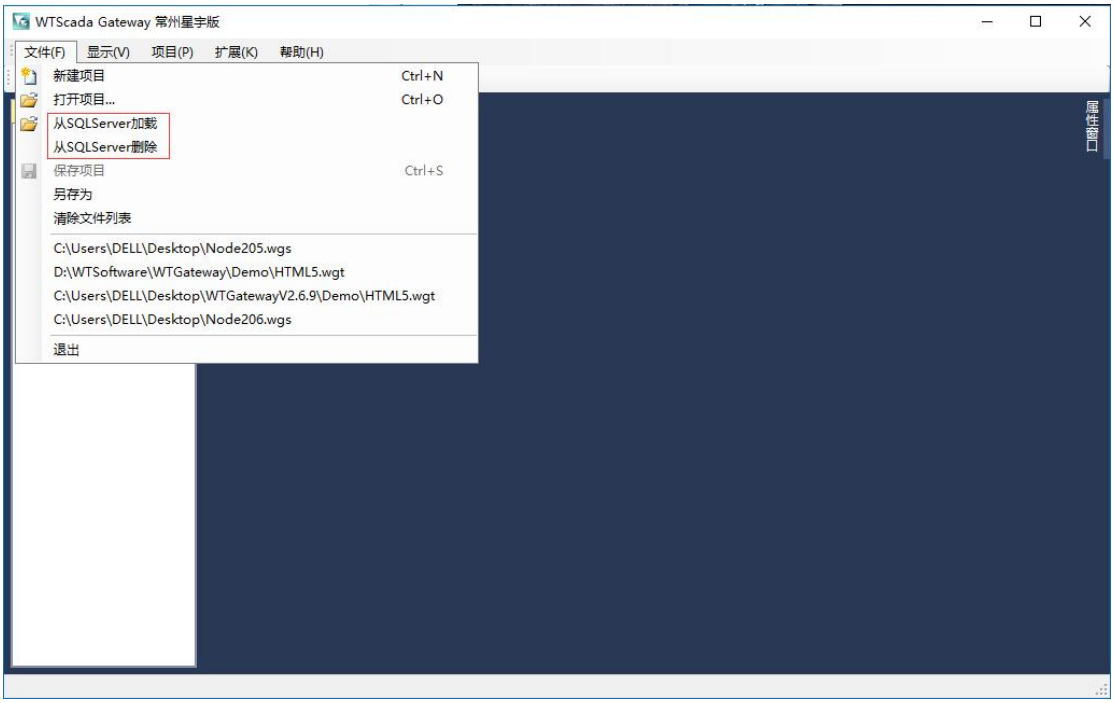
④ ⑤

2）触发表达式可以是 1 个布尔值变量,如 [Blink] ： False 到 True 就执行④ ⑤

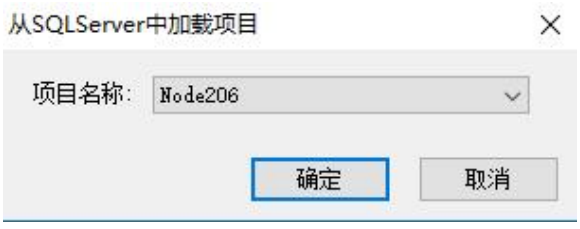
3）触发表达式可以是 1 个 1 个布尔表达式，如 [Minute] % 5 ==0 ： Minute 变量的值是 5 的倍数就执行④ ⑤

4.2 使用 SQLServer 存储项目配置的补充说明

当组态软件目录下有 sqlserver.txt 文件时就会开启 SQLServer 存储项目配置的菜单，在文件菜单下会显示“从 SQLServer 加载”和“从 SQLServer 删除”两个菜单。



点击“从 SQLServer 加载”会显示



可以在网络可达的电脑打开组态项目，项目保存在 SQLServer 的最大优势是可以在其他地方对项目进行组态修改，统一备份（必须对数据库进行定期备份防止数据丢失，一般建议存储 1 个本地项目备份）。

点击“从 SQLServer 删除”会显示

可以删除项目配置。

sqlserver.txt 文件内容是 SQLServer 数据的连接字符串信息：

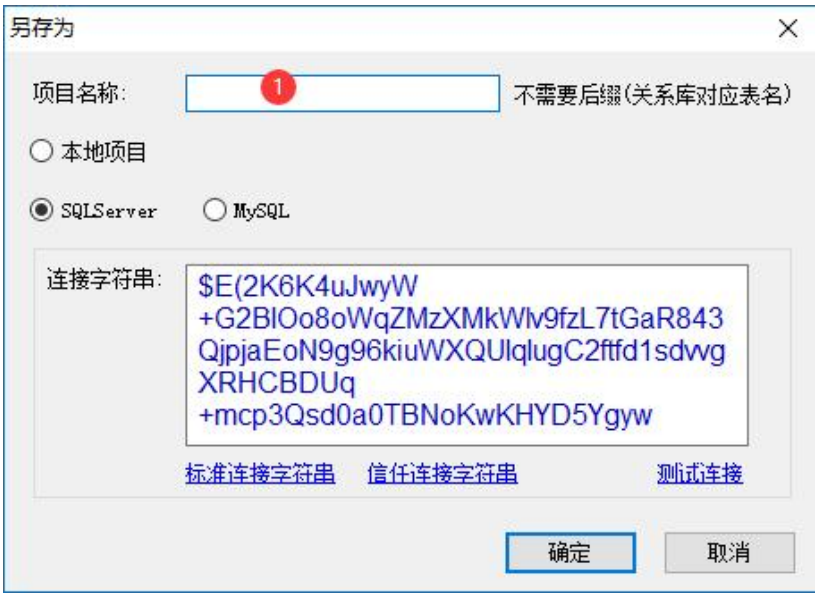
Data Source=192.20.100.212;Initial Catalog=ScadaProject;User ID=sa;Password=xxx.;Connect Timeout=30;

该连接字符串可以利用组态工具进行加密，当前目录下的是加密信息，软件可以自动判断连接信息是明文还是加密信息。

本地项目保存到 SQLServer 的操作步骤如下：

1）打开本地组态项目

2）点击文件菜单的另存为

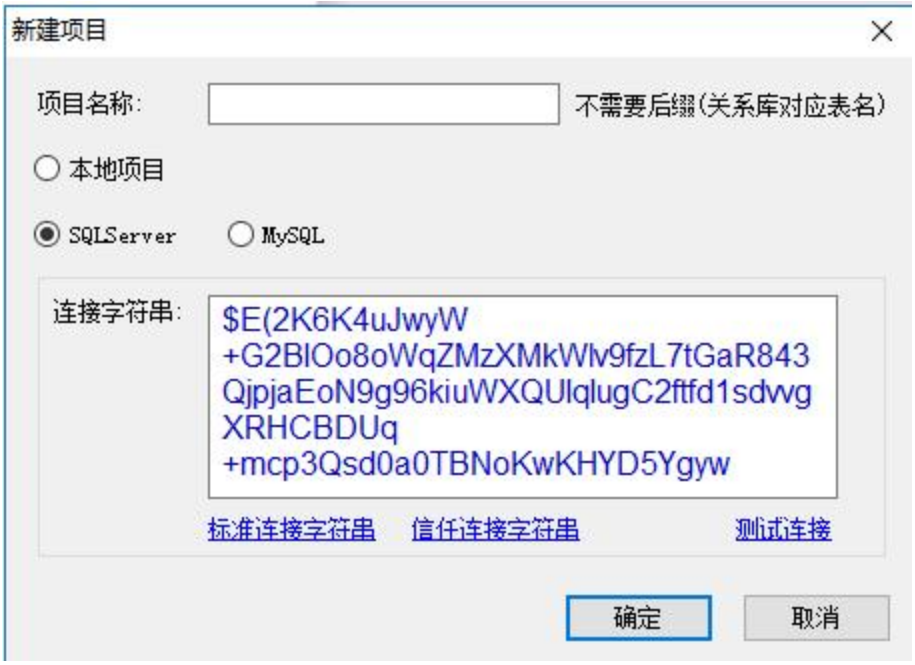


在项目名称中填写数据库中表的名称如 Node250，如果表存在无法保存。

3）点击确定，选择本地存储文件名称后就存储到了 SQLServer 中，数据库中会增加 1 个项目名称的表。

创建 SQLServer 中的项目操作步骤如下：

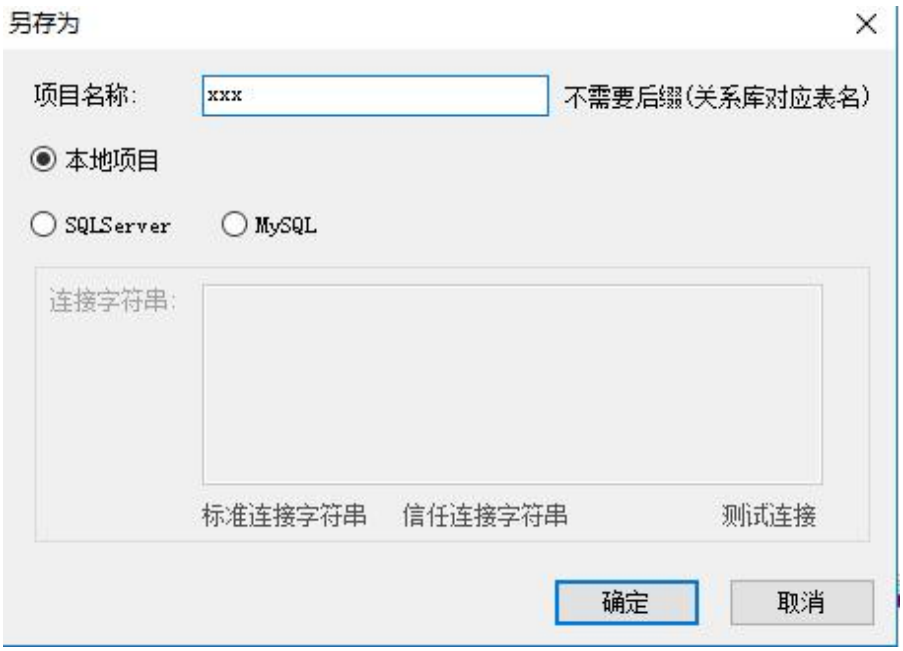
1）文件菜单点击新建



在项目名称中填写数据库中表的名称如 Node240，如果表存在无法保存。

4）点击确定，选择本地存储文件名称后就完成了项目创建，数据库中会增加 1 个项目名称的表。

组态项目存储在 SQLServer 中时本地的项目文件只是 1 个数据库连接信息，打开项目可以另存为本地项目备份。



4.3 表达计算

WTGateway 提供了在线计算功能（网络接收驱动不支持），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。

数学函数

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 如果条件忙返回 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’

组态函数符号

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)

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)

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

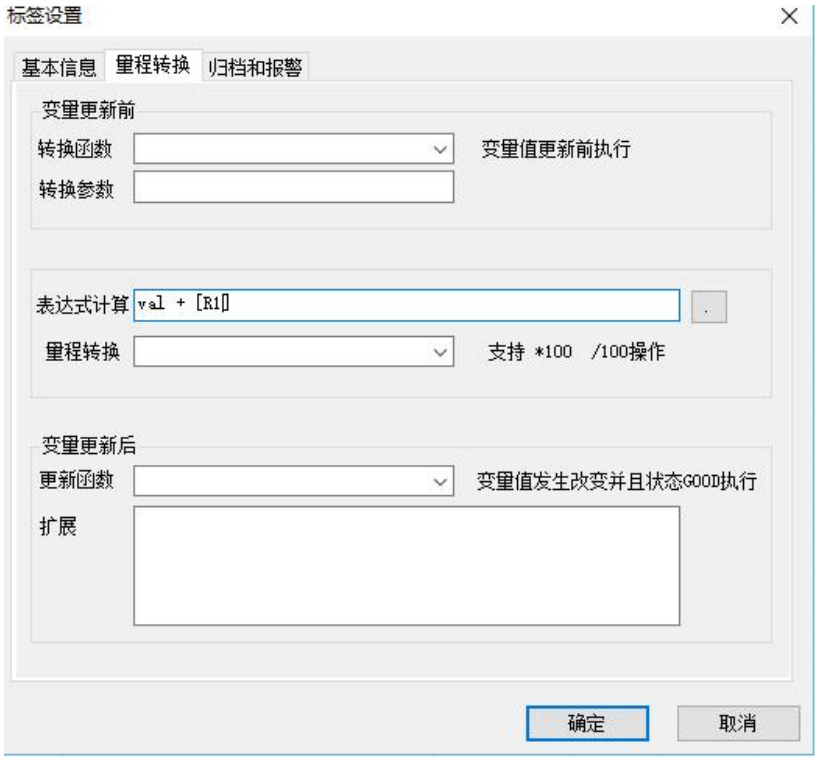
tagunit，参数同上，读取变量单位

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

配置界面

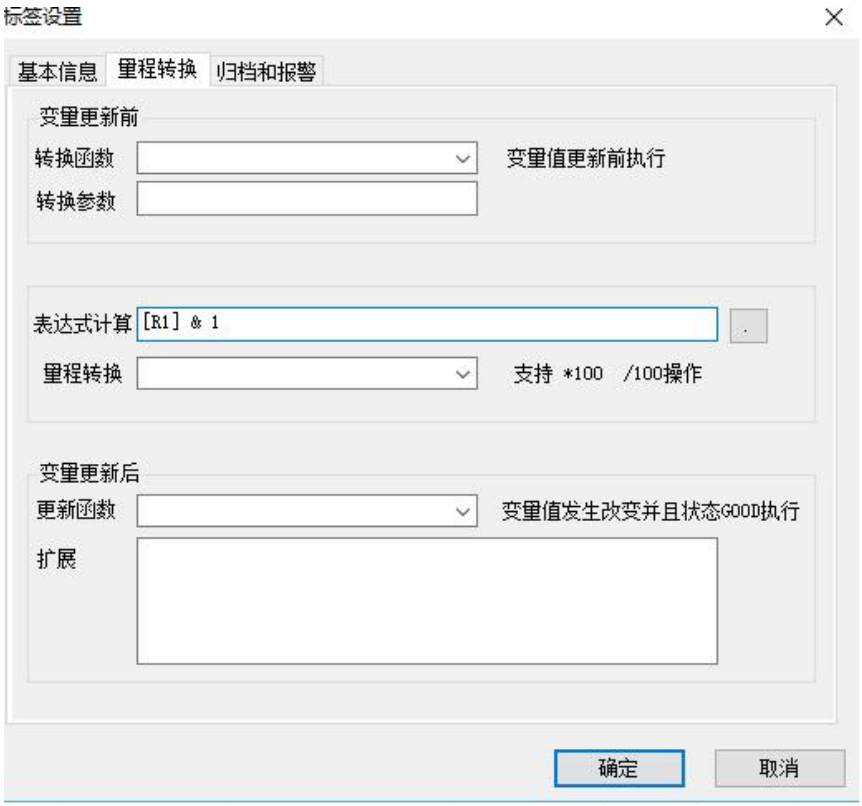


在驱动变量配置界面的表达式运行时可以在线修改（管理登录），如果变量的扫描周期为 0，驱动每次成功采集后都会对相关变量进行表达式计算，如果不为 0 则大于扫描时间才会执行计算，不管扫描周期为多少，采集失败的情况下不会执行表达式计算（模拟驱动除外，模拟驱动没有采集失败的情况）。

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，第 3 位 [R1] & 4，第 4 位 [R1] & 8

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。

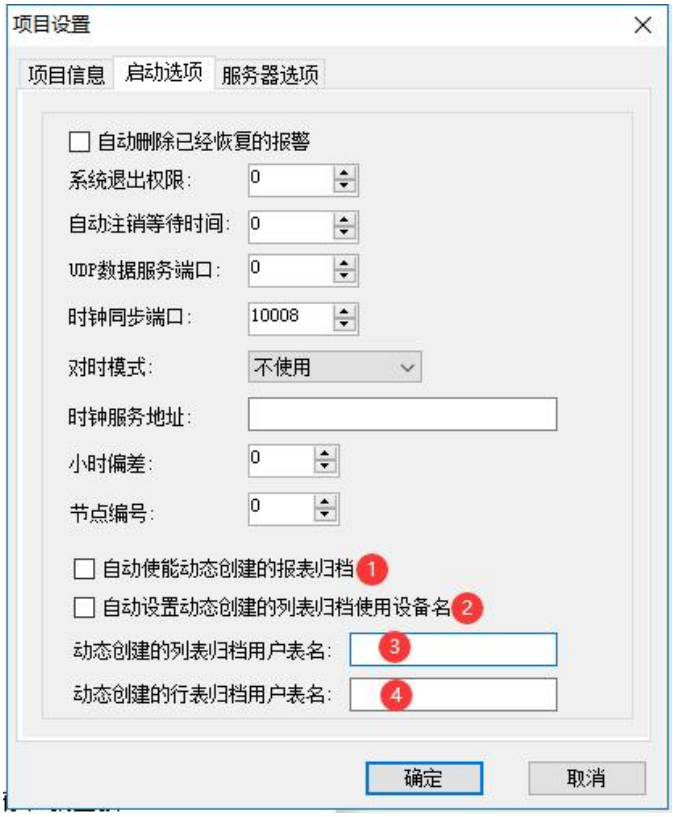
4）变量状态判断

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

4.4 归档配置



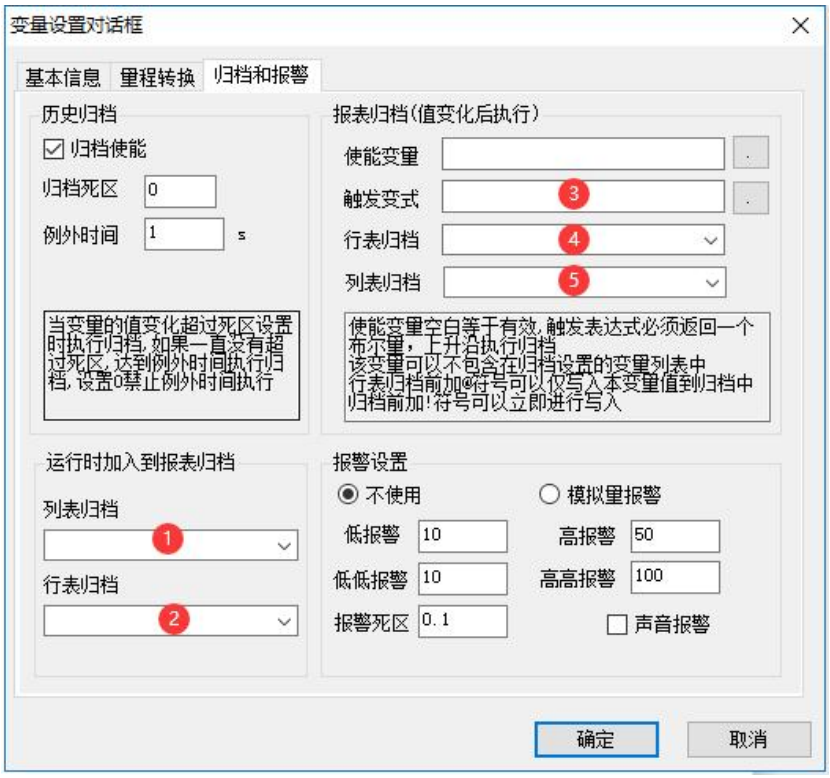
① 运行时自动创建的列表或者行表归档设置自动设置为启用状态

② 运行时自动创建的列表归档配置为使用设备名

③ 运行时自动创建的列表归档设置用户表名

④ 运行时自动创建的行表归档设置用户表名

自动创建的行表或者列表归档设置是指下图的①和②的设置



① 运行时自动把本变量加入到指定的列表归档配置中，如果列表归档配置已经存在就直接加入，如果不存在就创建 1 个（使用项目配置中的设置配置是否自动启用，是否使用设备名分割，是否设置用户表名）

② 运行时自动把本变量加入到指定的行表归档配置中，如果行表归档配置已经存在就直接加入，如果不存在就创建 1 个（使用项目配置中的设置配置是否自动启用，是否设置用户表名）

③ 当变量的值发生改变后执行触发表达式，如果表达式输出为一个布尔量，则表达式输出从 False 到 True 过程执行后续操作（单次执行）；如果表达式输出为其他值类型，如果表达式计算结果和上一次计算结果不一样就执行后续操作。

后续操作指 执行④ ⑤定义的归档操作。

④ 对设定的行表归档执行写入操作，如果以@符号开头，则对@符号后面的数据库表执行写入本变量的操作。

⑤ 对设定的列表归档执行写入操作，如果以!符号，则立即写入!符号后面的列表归档设置。

触发表达式说明：

4）触发表达式可以是 1 个值变量，如 [Second] ： 和上次的值不一样就执行

④ ⑤

5）触发表达式可以是 1 个布尔值变量,如 [Blink] ： False 到 True 就执行④ ⑤

6）触发表达式可以是 1 个 1 个布尔表达式，如 [Minute] % 5 ==0 ： Minute 变量的值是 5 的倍数就执行④ ⑤

4.5 JavaScript 函数

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

string BasePath：返回软件根目录名称

string ProjectPath：返回项目路径名称

void WriteReport(int type,string ruleName,bool direct=false) 可以在模拟驱动使用 js 变量*进行报表归档*

写入报表归档，type=0 列表归档，1 行表归档

例如：app.WriteReport(0,“report1”) 执行列表归档 report1 写入

bool PlusTag(string tagName, object inteval)

变量脉冲操作

例如：app.PlusTag("tag1",5000) 设置变量 tag1 的值为 1（True）,5 秒后再设置为 0（False）

bool SetTagValue(string tagName, object value）

设置变量值，建议使用$简化函数

bool AddTagValue(string tagName, object value）

对变量值执行加减操作

bool ToogleTagValue(string tagName）

切换变量值 True 到 False、False 到 True、 0 到 1、 1 到 0

BaseChannel GetChannel(string name)

返回变量对象

void WriteOperatorLog(string tagName, string value, string desc,string memo="")

写操作日志

void Execute(string app, string param)

运行外部程序

bool WriteFile(string filename, string data)

写文件，文件路径是 Config 目录

string ReadFile(string filename)

读文件

int GetChannelState(string tagName)

读取变量状态 0 Unknow 1 Good 2 Bad

void PostMessage(string appTitle, int msgid, int wparam, int lparam = 0)

调用 Windows API PostMesage 发消息

void ActiveAndPostMessage(string appTitle, int state, int msgid, int wparam, int lparam = 0)

Windows API 激活窗口并发送消息

void SaveTagValues()

保存变量值

bool 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 LogInfo(string source, string message)　　记录日志信息

void LogWarning(string source, string message) 记录警告日志

void LogError(string source, string message)　记录错误日志

2）comms：驱动管理器

object[] Channels　得到全部变量数组

string [] PluginIds 得到驱动 id 列表

object [] Plugs

object [id] 获取某个驱动

3）logger：日志管理器

void Log(string msg)：记录日志

void LogInfo(string source,string message)：记录日志

void LogWarning(string source, string message)：记录警告日志

void LogError(string source, string message)：记录错误日志

记录操作日志

例如：logger.log(“一条日志记录”);

ext：扩展管理器

bool DoCommand(string extName,string cmd,object data)

发送扩展命令

例如：ext.DoCommand(“WxMessage”，“wxUser”，“报警消息”）;

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可以从 C#中获取，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

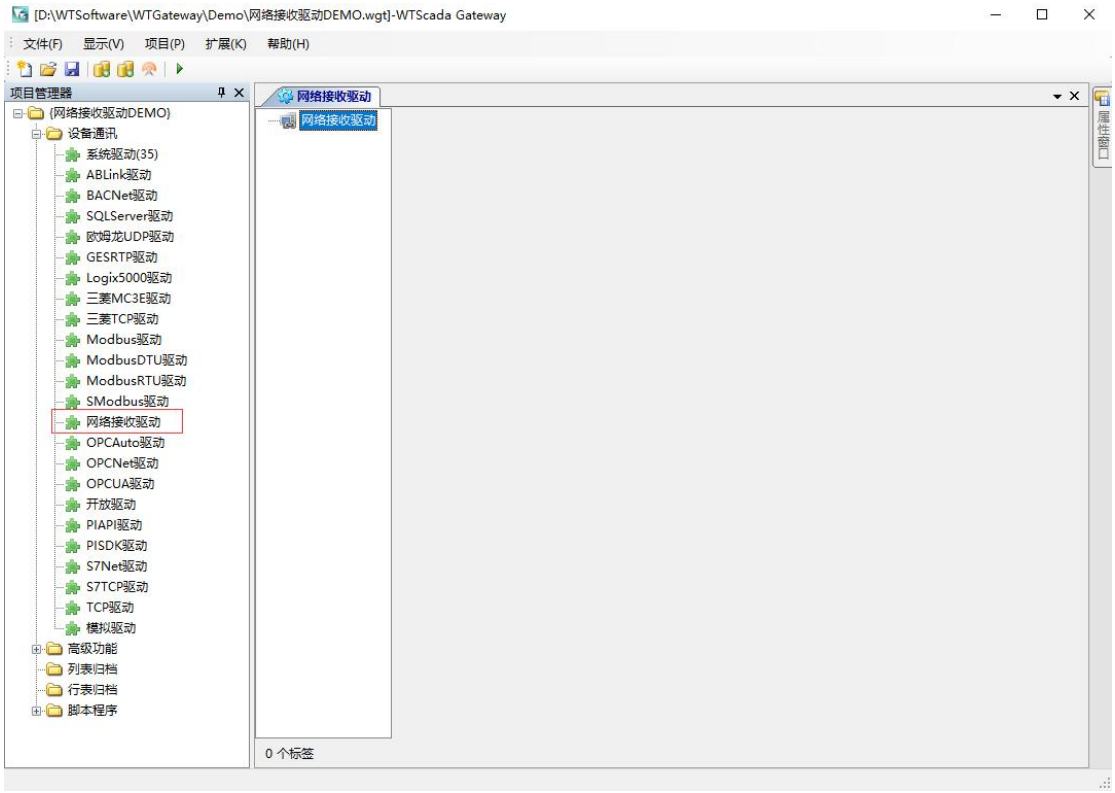
tag1.RangeMin 量程下限，读写

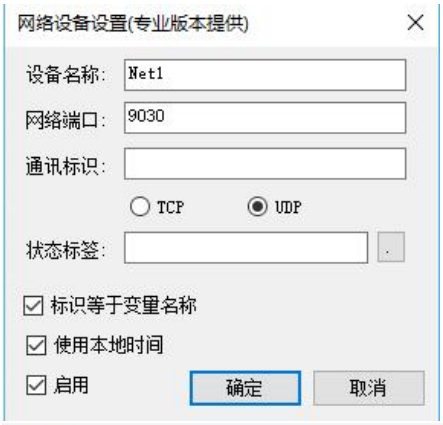
tag1.RangeMax 量程上限，读写

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

Config 目录下的 global.js 是预定义全局函数,可以修改该文件增加自己的函数，全局函数在任何 js 代码中都有效。

4.6 网络接收驱动





上面创建了 1 个 UDP 接收的设备，UDP 端口为 9030

使用本地变量时间

用户通过 UDP 把数据发到 9030 端口上

数据格式为经过 GZIP 压缩后的 XML 文档，格式如下

<tagvalues>

<tagvalue id=”Net1#tag1” v=”5” t=”2020-4-20 12:00:00” s=”1” />

<tagvalue id=”Net1#tag2” v=”6” t=”2020-4-20 12:00:00” s=”1” />

<tagvalue id=”Net1#tag3” v=”7” t=”2020-4-20 12:00:00” s=”1” />

<tagvalue id=”Net1#tag4” v=”8” t=”2020-4-20 12:00:00” s=”1” />

</tagvalues>

id:变量名称（如果变量不在本驱动中，在其他驱动中中也会更新）

v：变量值

t：变量时间

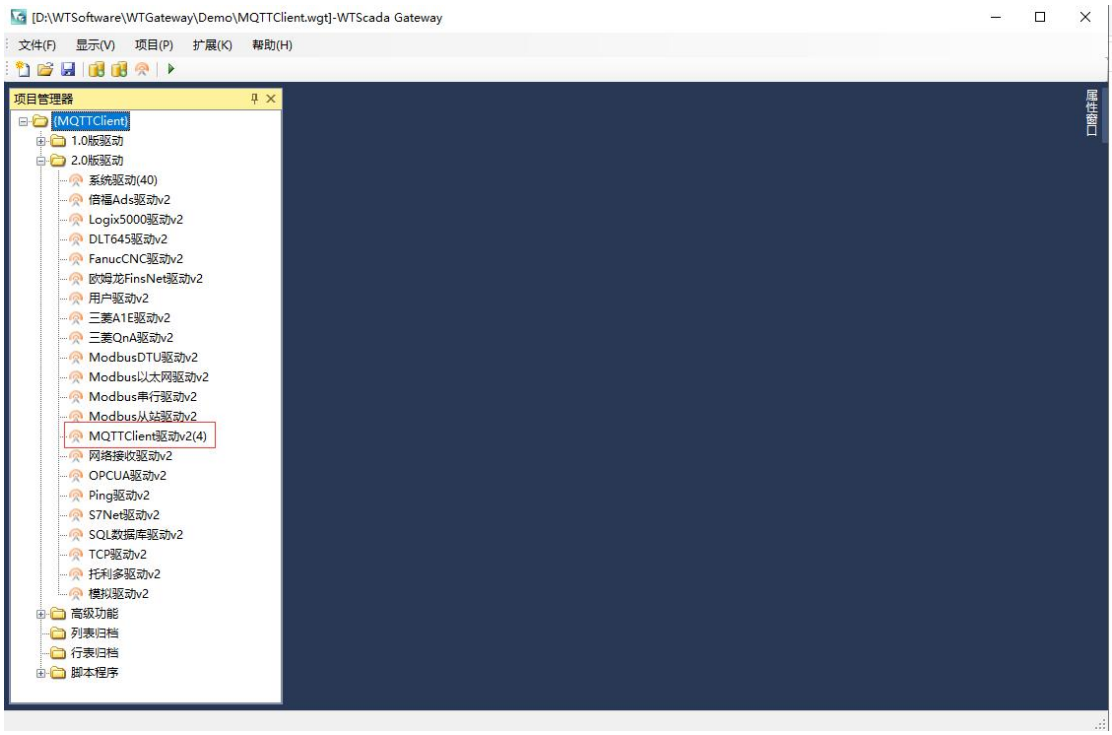
s：变量状态 0：未知 1：Good 2：Bad

UDP 模式下，变量是只读的，无法进行变量设置，是最简单的数据通讯方式，由于 UDP 协议限制，每个数据包必须限制在 64K 以下，由于经过了压缩，一般每

个数据包可以传输的变量数大于 5000 个。

网络接收驱动的 TCP 端口需要使用偶数端口如（9030，9032）

4.7 MQTT 驱动

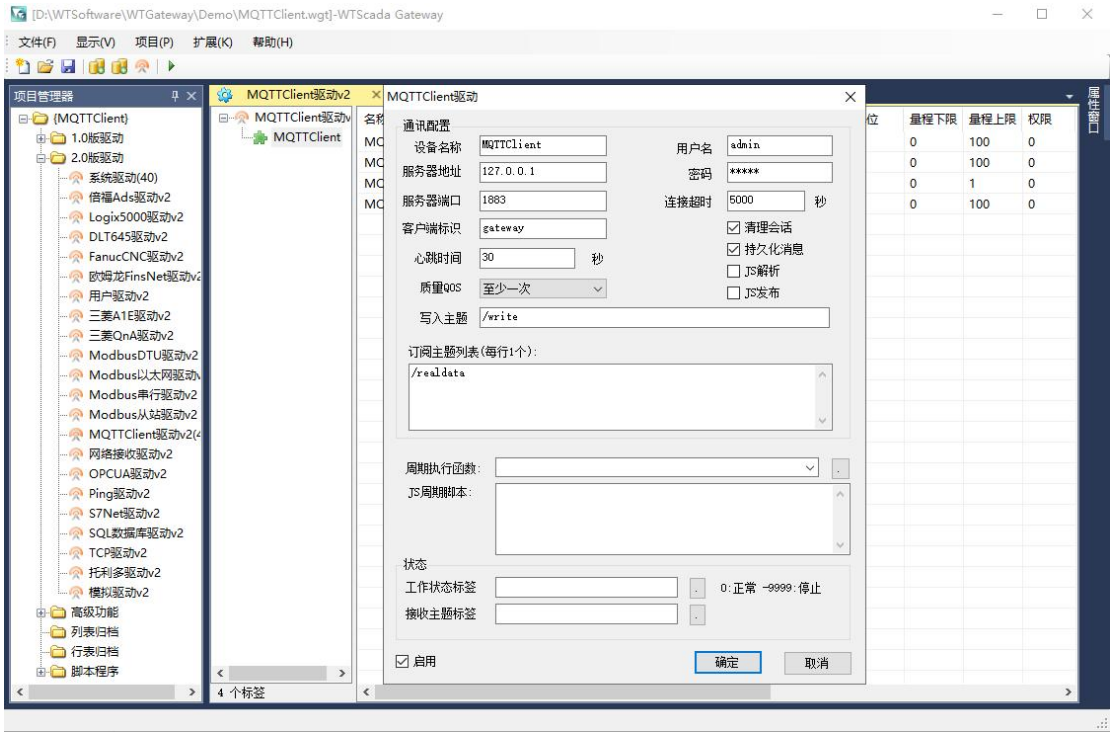


Demo 目录下提供了 2 个演示项目（MQTTClient.wgt 和 MQTTClient_自解析.wgt）

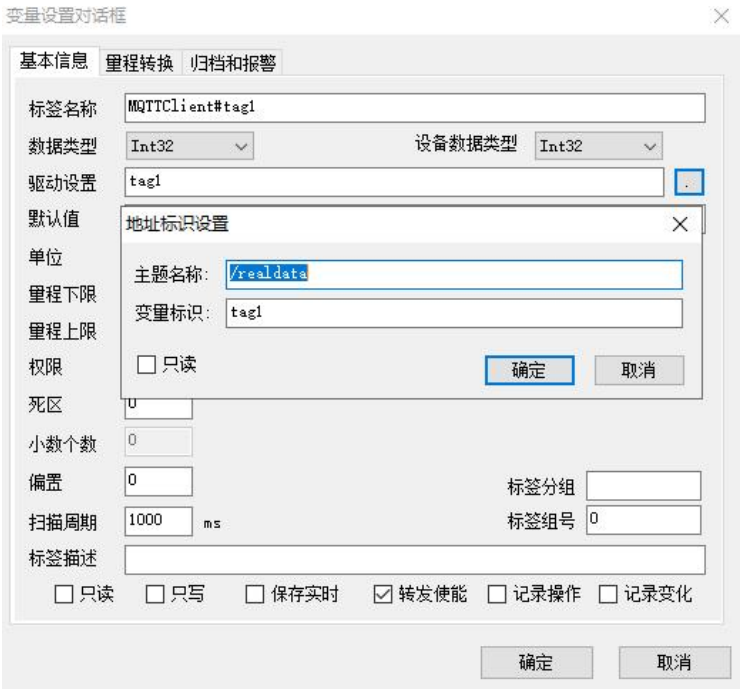
MQTTClient 驱动支持对标准 JSON 格式的键值对数据进行解析

格式如：{“tag1”:”12”,”tag2”:true}

其中 tag1，tag2 作为变量的地址值



订阅主题：设置需要订阅的主题列表，每行 1 个订阅，支持多个订阅
写入主题：对变量写值时发布的主题名称，默认写入数据为 Json 数据
变量设置：

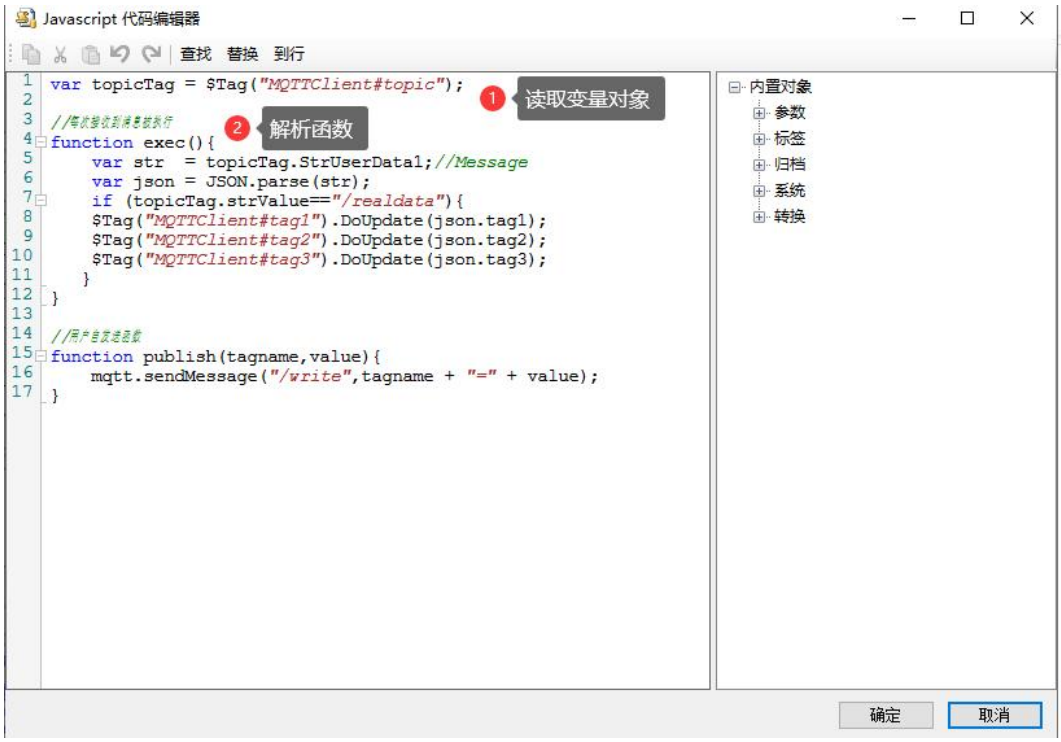


每个变量对应 1 个已经订阅的主题名称和 JSON 键值对的 1 个数值
当变量执行写入时对写入主题发布 1 个消息，数据格式为 JSON，内容为{“tag1”:数据}，JSON 的键是变量标识内容。

MQTTClient 驱动还可以使用 JS 脚本实现自定义数据的解析（MQTTClient_自解析.wgt）

当设备勾选了“JS 解析”后，就可使用 JS 脚本进行解析然后更新到变量中

首先需要配置 1 个接收主题标签变量，驱动在接收到订阅后会把主题名称和数据写入到该变量中，然后就可以在 JS 中进行解析

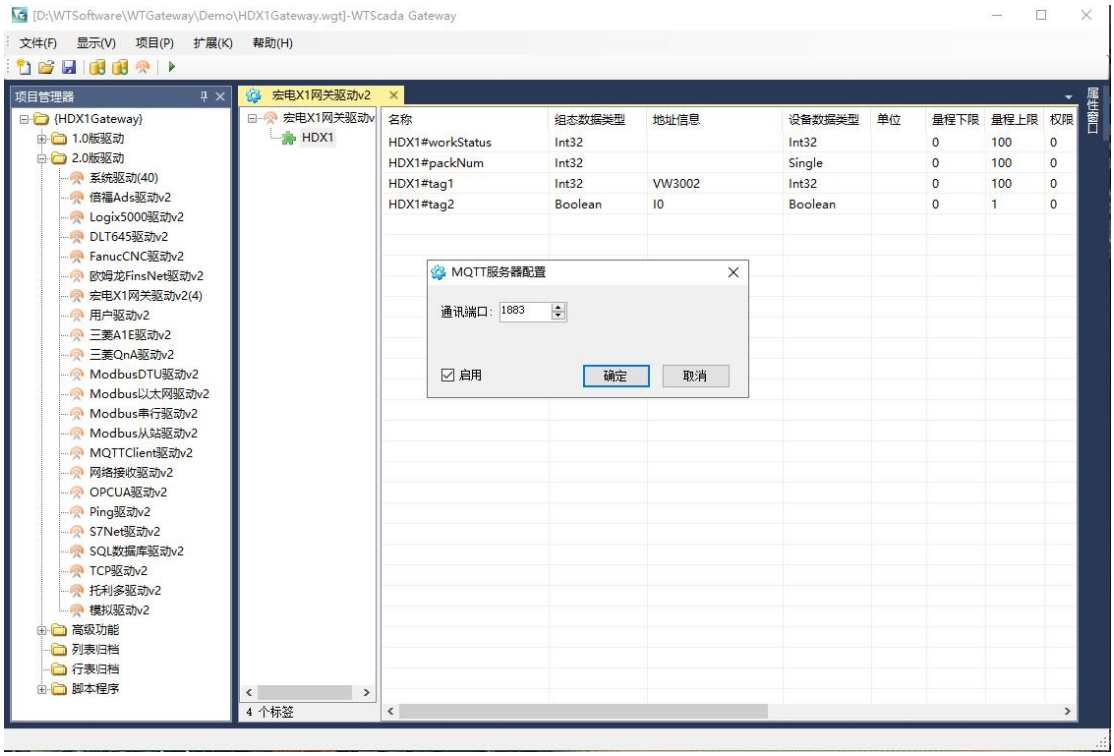


MQTTClient 驱动还支持自定义数据写入的发送（MQTTClient_自解析.wgt）

当设备勾选了“JS 发送”后，当变量进行写值时就会调用 publish 函数，该函数中可以发布信息到 MQTT 服务器

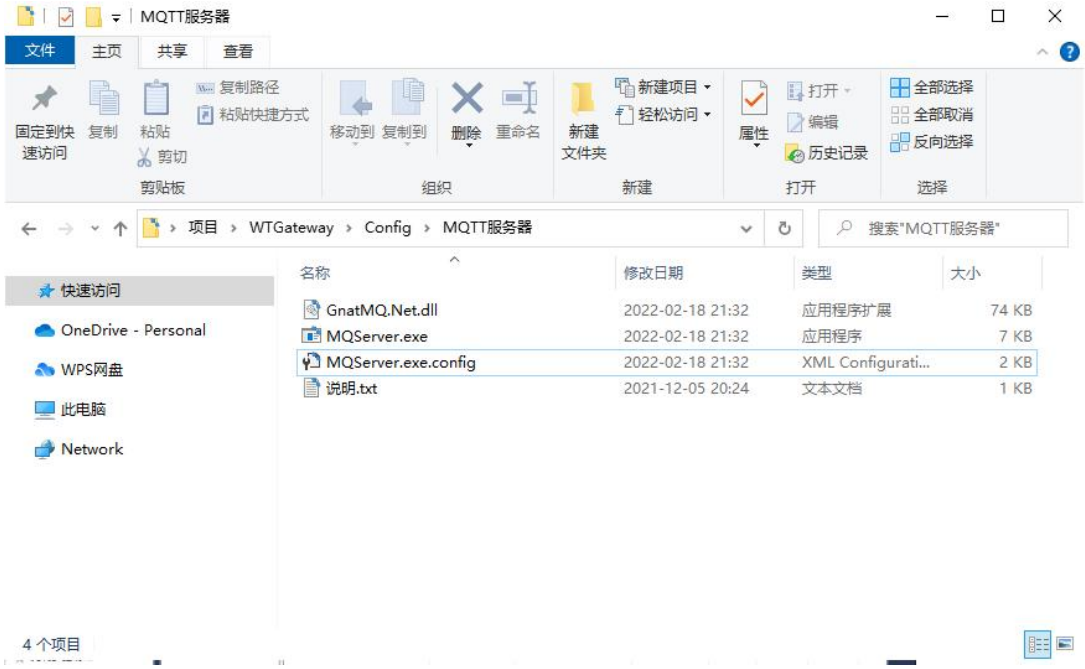
4.8 宏电 X1 网关驱动

本驱动支持宏电 X1 网关通过 MQTT 协议采集数据，驱动支持读取和写入操作，Gateway 从 V2.13.1 版本开始提供驱动。宏电 X1 网关的 MQTT 是 1 个客户端，在 MQTT 中必须存在一个 MQTT 服务器，客户端连接上后向某个主题名发送数据，其他订阅了该主题的客户端就能收到数据，Gateway 也是一个 MQTT 客户端，通过订阅特定的主题获取 X1 网关发布的的数据，X1 网关同时也订阅了 1 个主题用来获取 Gateway 的数据写入操作。Gateway 扩展接口有 1 个可以使用的 MQTT 服务器。

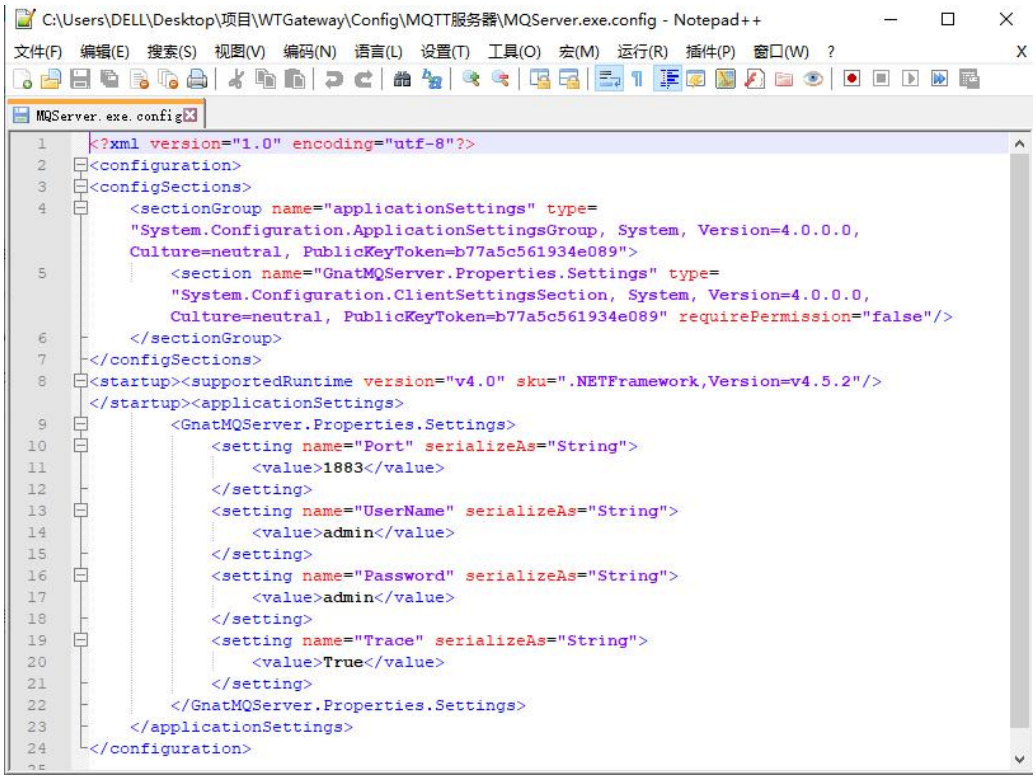


服务器的登陆密码来自 Gateway 的用户配置。

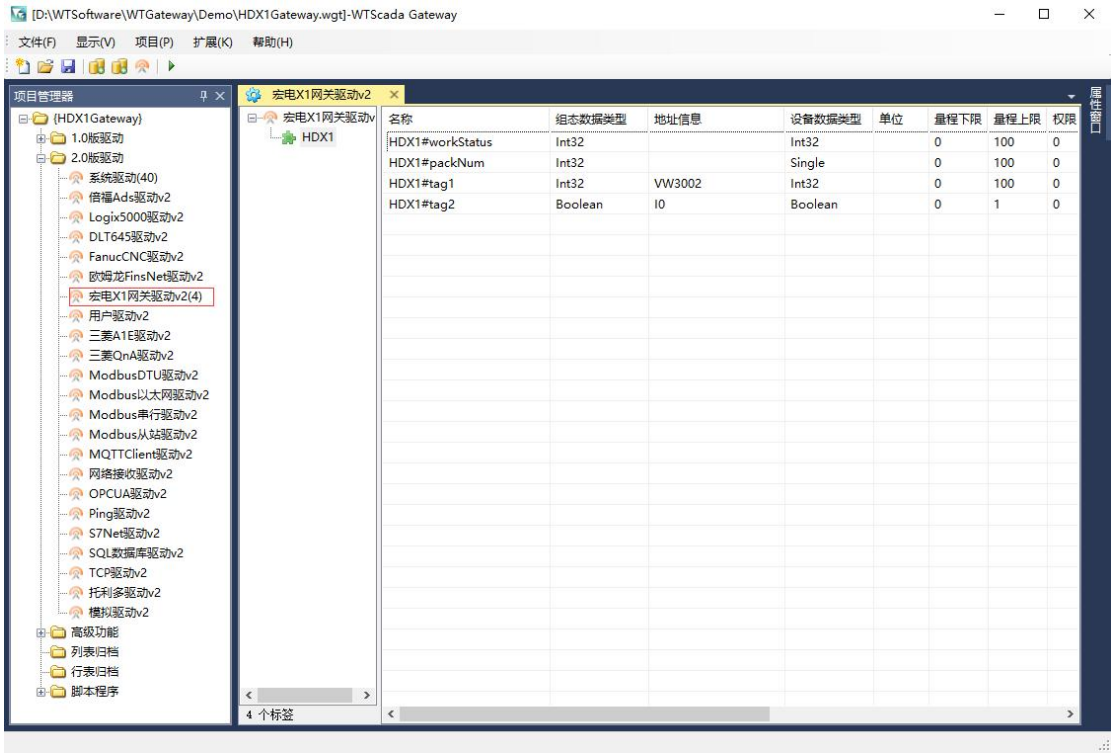
Gateway 软件的 Config 目录下还有 1 个 MQTT 服务器软件可以使用



可以修改 MQTTServer.exe.config 文件进行端口和密码配置，这个程序可以显示客户端的数据信息，因此调试状态比较有用，建议扩展服务用于正式项目。


C:\Users\DELL\Desktop\项目\WTGateway\Config\MQTT服务器\MQServer.exe.config - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
MQServer.exe.config
<?xml version="1.0" encoding="utf-8"?>
<configuration>
<configSections>
<sectionGroup name="applicationSettings" type=
"System.Configuration.ApplicationSettingsGroup, System, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089">
<section name="GnatMQServer.Properties.Settings" type=
"System.Configuration.ClientSettingsSection, System, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false"/>
</sectionGroup>
</configSections>
<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5.2"/>
</startup><applicationSettings>
<GnatMQServer.Properties.Settings>
<setting name="Port" serializeAs="String">
<value>1883</value>
</setting>
<setting name="UserName" serializeAs="String">
<value>admin</value>
</setting>
<setting name="Password" serializeAs="String">
<value>admin</value>
</setting>
<setting name="Trace" serializeAs="String">
<value>True</value>
</setting>
</GnatMQServer.Properties.Settings>
</applicationSettings>
</configuration>

宏电 X1 网关驱动配置


[D:\WTSoftware\WTGateway\Demo\HDX1Gateway.wgt]-WTScada Gateway
文件(F) 显示(V) 项目(P) 扩展(K) 帮助(H)
项目管理器
(HDX1Gateway)
1.0版驱动
2.0版驱动
系统驱动(40)
倍福Ads驱动v2
Logix5000驱动v2
DLT645驱动v2
FanucCNC驱动v2
欧姆龙FinsNet驱动v2
宏电X1网关驱动v2(4)
用户驱动v2
三菱A1E驱动v2
三菱QnA驱动v2
ModbusDTU驱动v2
Modbus以太网驱动v2
Modbus串行驱动v2
Modbus从站驱动v2
MQTTClient驱动v2
网络接收驱动v2
OPCUA驱动v2
Ping驱动v2
S7Net驱动v2
SQL数据库驱动v2
TCP驱动v2
托利多驱动v2
模拟驱动v2
高级功能
列表归档
行表归档
脚本程序
宏电X1网关驱动v2
宏电X1网关驱动v
HDX1
4 个标签
名称 组态数据类型 地址信息 设备数据类型 单位 量程下限 量程上限 权限
HDX1#workStatus Int32 Int32 0 100 0
HDX1#packNum Int32 Single 0 100 0
HDX1#tag1 Int32 VW3002 Int32 0 100 0
HDX1#tag2 Boolean I0 Boolean 0 1 0
属性窗口

使用 Gateway 的 MQTT 时服务器 IP 地址总是设置为 127.0.0.1

离线时间：如果超过这个时间网关没有发送数据到驱动则变量显示坏点

一般情况下，每个 X1 网关在 Gateway 配置 1 个设备，每个网关的发送主题和订阅主题不能相同，每个网关的设备名称也不能相同。

提醒：为了减少 4G 流量，建议网关的设备名称尽量短，网关中配置的变量名称

尽量短，数据发送频率根据需要设置。

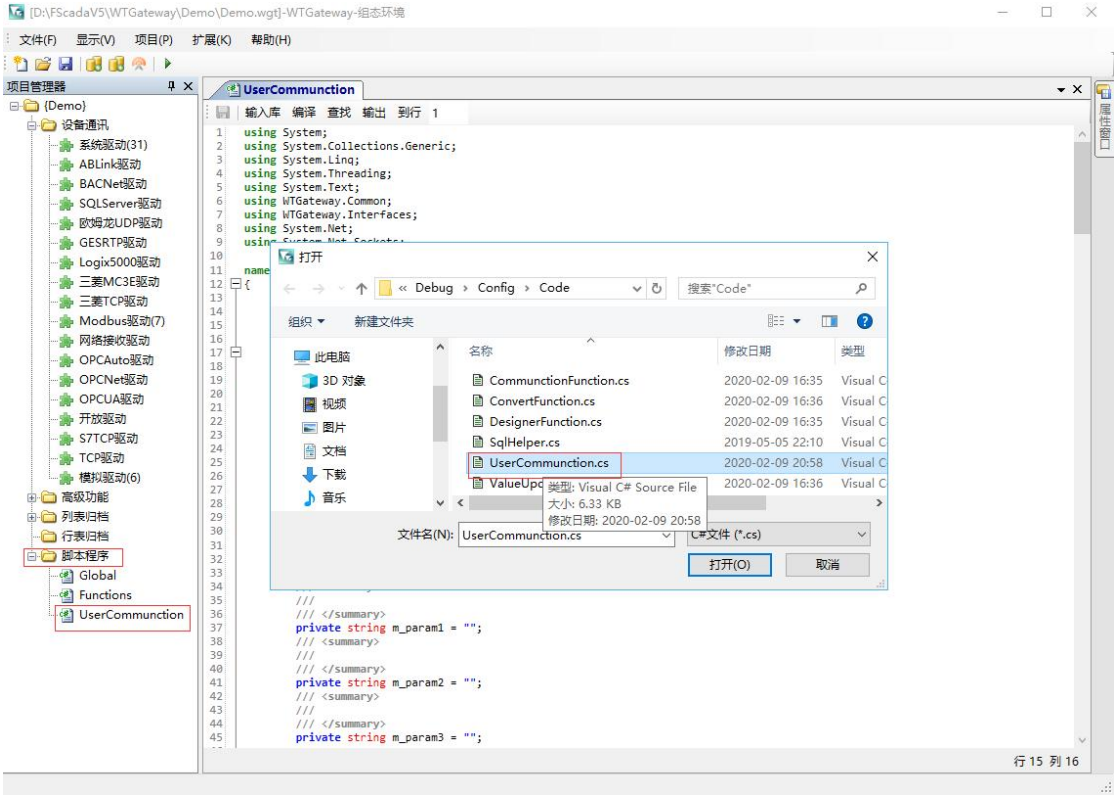
如果没有写入操作，可以多个网关配置使用 1 个设备驱动接收，设备配置界面的订阅主题可以输入多个订阅主题，但是请注意区分设备名称，变量寻址的原理是**订阅主题.设备名称.变量名称**。

驱动还提供了 MQTT 服务器连接状态变量和接收数据包变量的配置，用于监视通讯过程。

提醒：建议修改 MQTT 服务器默认端口 1883 为其他端口，并强化密码，防止被恶意攻击，另外请注意开始服务器防火墙端口。

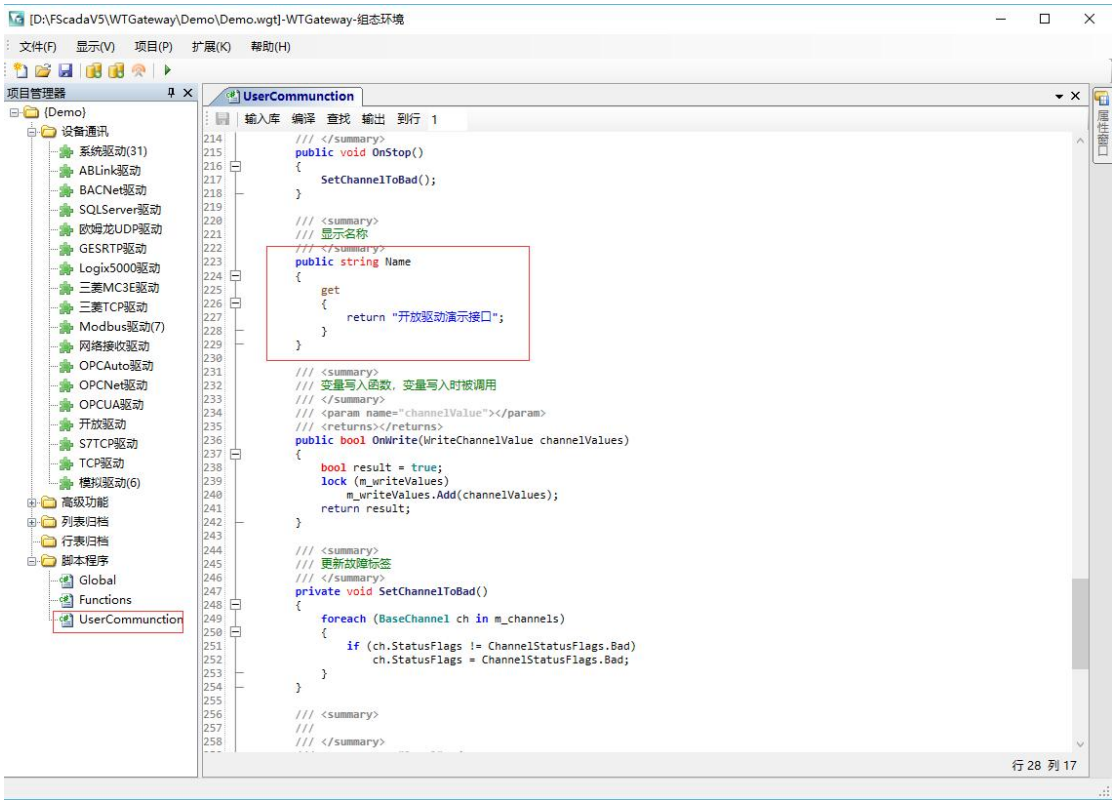
4.9 开放驱动

开放驱动为用户自定义驱动提供了一种非常简单的方式，开放驱动通过 C# 脚本实现驱动开发，系统提供了底层封装和 1 个驱动开发模板文件。



首先在脚本程序上鼠标右键导入 Config\Code 目录下的 UserCommunction.cs 文件到项目中。

类名和脚本文件名称和根据需要进行修改。



Name 属性：用于代码编译后的显示名称，驱动配置界面可以找到这个名称。

该类主要函数如下：

Init 函数：驱动初始化被调用

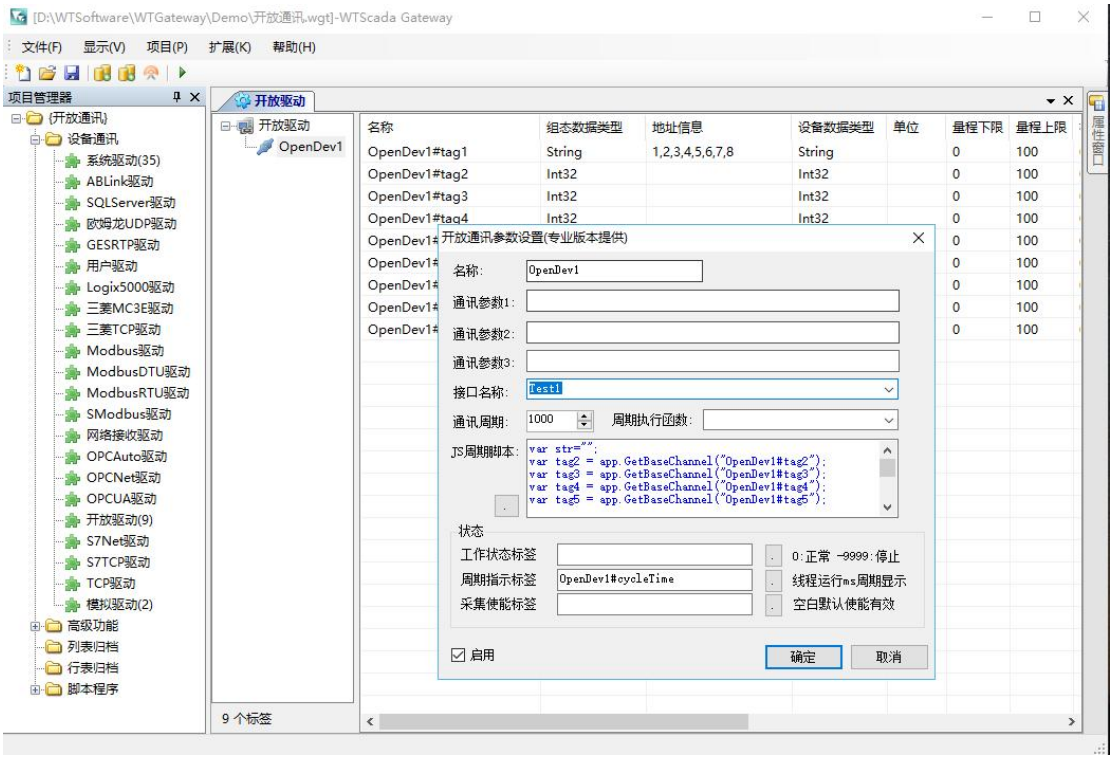
OnStart 函数：驱动启动前被调用,返回 True 驱动开始启用

OnStop 函数：驱动停止时被调用

Dispose 函数：释放内存

OnWrite 函数：变量值被写入

OnCycle 函数：采集线程循环函数



驱动配置界面接口中选择对应的类名称

配置界面提供了 3 个字符串类型的参数配置

1 个标准驱动的开放驱动执行过程如下：

1）系统调用初始化化函数，该函数传递了设备名称，通讯参数，变量信息和状态变量定义，在该函数中保存需要的信息，对变量根据通讯地址定义进行排序，生成和设备的通讯列表，返回 true 表示初始化成功。

2）系统调用 OnStart 函数，通知驱动即将启动，返回成功。

系统开始按采集频率设置循环调用 OnCycle 函数，在函数中实现数据采集。

OnCycle 函数中可以更新设备状态，用于界面显示。

3）系统调用 OnStop 函数，通讯驱动准备停止。

4）系统调用 Dispose 函数，用于释放资源。

5）当变量需要写入值时，值写入函数被调用。

4.10 SQL 关系库驱动

SQL 关系库驱动支持主流关系数据库，使用行标识读取指定列的数据，支持反向写入。

Oracle 和 SQLServer 驱动已经集成到组态软件中，PostgreSQL、MySQL 需要

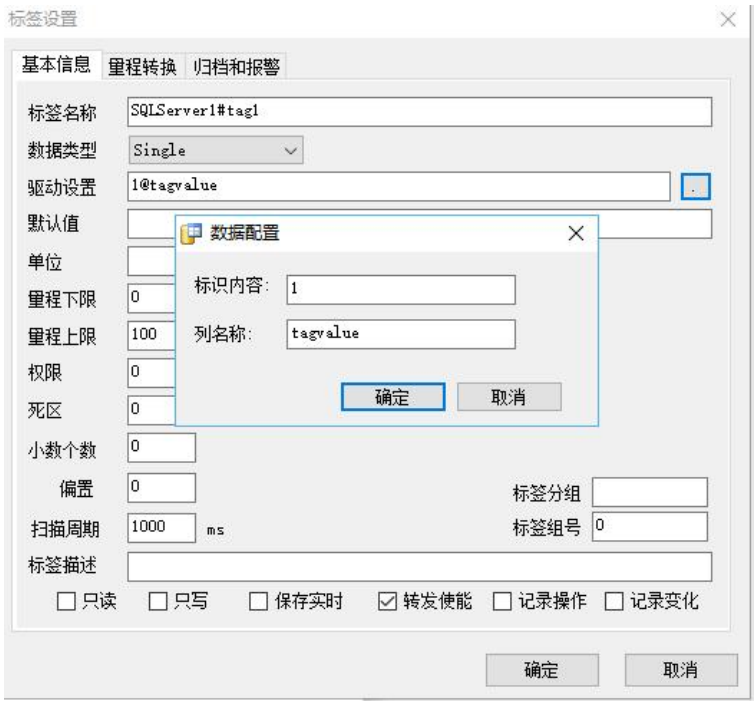
自行安装 Config 目录下的 mysql-connector-net-8.0.15 安装包。

标识列：用于变量确定对应行的标识，通常为 id

刷新间隔：SQL 查询周期

查询 SQL：循环查询执行的 SQL，查询可以返回多个行集

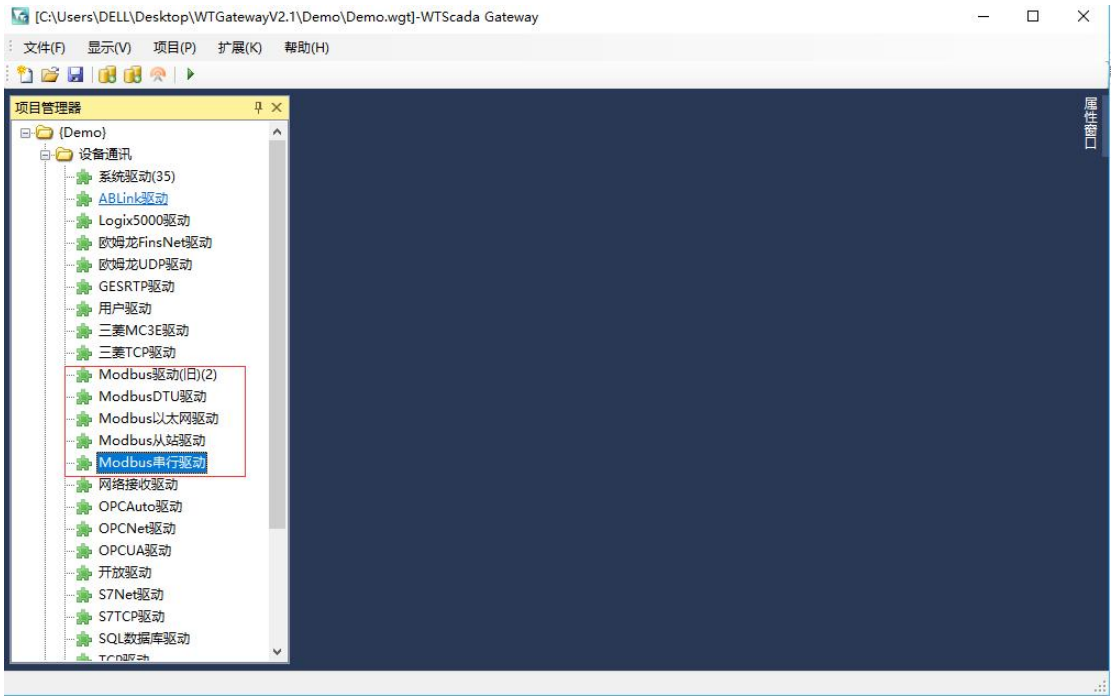
更新 SQL：变量写入时执行的 SQL 代码模板，支持{tagname} {value} {id} {col}替换符。



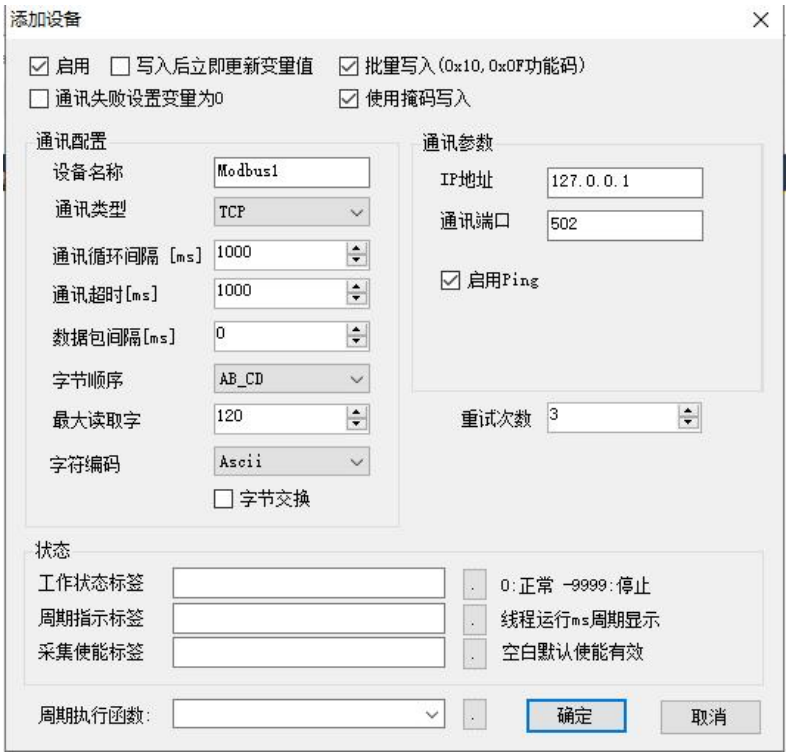
上面变量的驱动设置表示，该变量取值来着于行标识为 1 中的 tagvalue 列值。

4.11 Modbus 驱动说明

WTGateway 软件包括了 5 个 Modbus 驱动，分别应用于不同的场合，本文档对各驱动的应用进行详细说明。



1）Modbus 驱动（旧）



标准 Modbus 驱动，支持全部类型的 Modbus 协议（TCP、UDP、RTU、ASCII），不适用于虚拟串口，该驱动串口通讯在发生通讯异常时会重新连接，串口通讯遵循标准 Modbus 协议严格判断字符间隔。

该驱动已经淘汰，多设备下存成 CPU 使用过高的问题，不再建议使用。

2）ModbusDTU 驱动

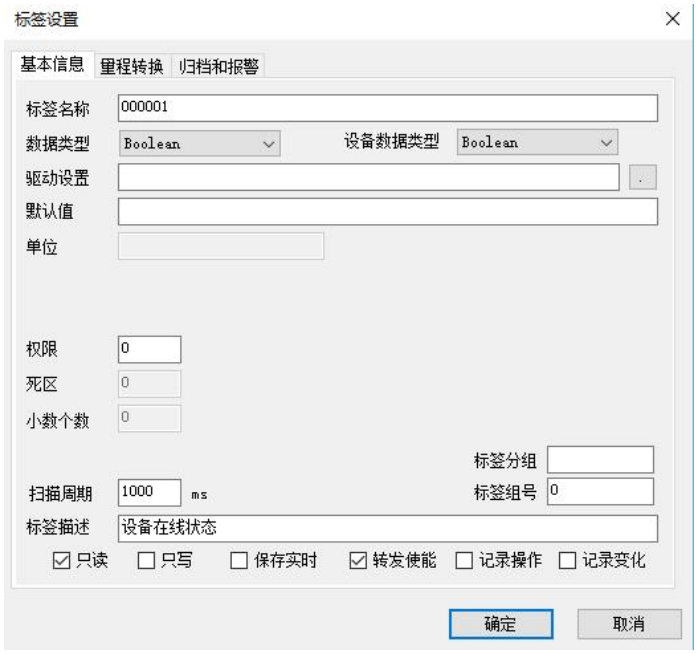
用于在支持透明传输的 Gprs、4G DTU 上进行 Modbus 通讯，支持 RTU 和 TCP 协议，该驱动多个 DTU 设备共用一个通讯线程，可用于大量的 DTU 通讯场合。

DTU 标识在变量上进行配置

透明串的 DTU 通讯过程如下：DTU 上线后连接到组态配置的 TCP 通讯端口，发送注册包用于标识设备名称（对应组态的 DTU 编号），之后 DTU 进入透明传输模式。支持 1 个字节的心跳包。

建议控制 1 个设备配置不超过 100 个 DTU。

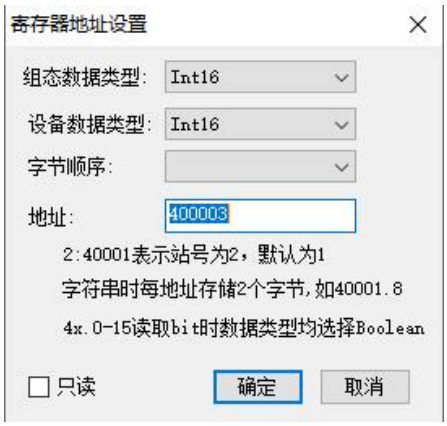
可以创建一个以 DTU 编号命名的 Boolean 只读变量用于指示 DTU 在线状态。



这样就可以获取 DTU 在线状态。

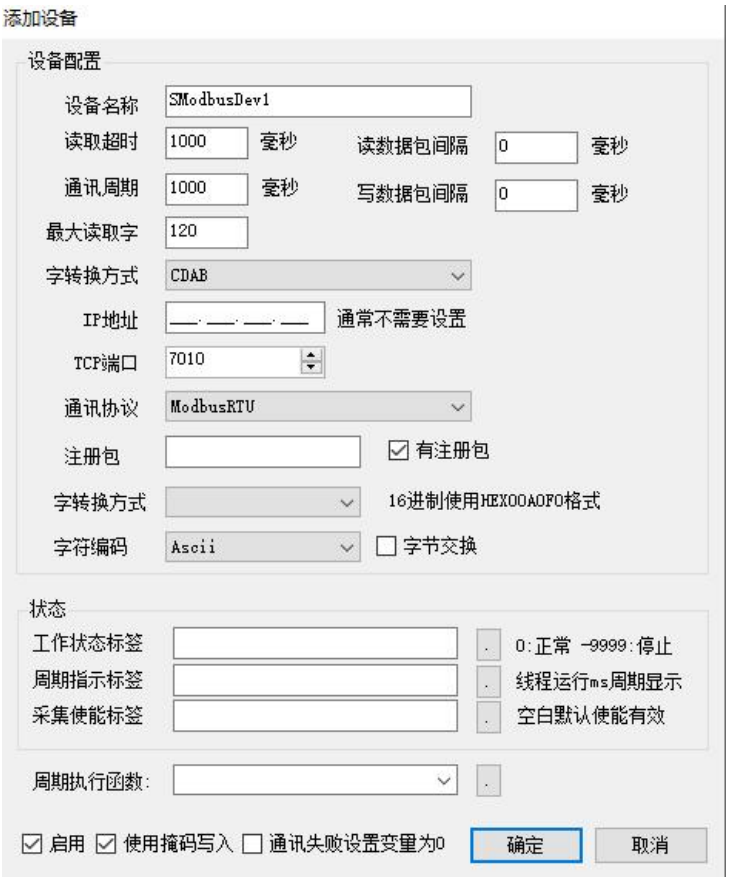
3）Modbus 串行驱动

新的 Modbus 串口驱动，支持 RTU、ASCII，该驱动可用于虚拟串口上的 Modbus RTU 协议和使用串口服务器进行串口转以太网的 ModbusRTU，TCP 模式下，驱动使用 TCP 连接到设置的串口服务器后使用 ModbusRTU 协议进行数据通讯。



该驱动提供了每个变量提供了独立的字节顺序设置，空白情况下使用驱动全局设置。

4）Modbus 从站驱动



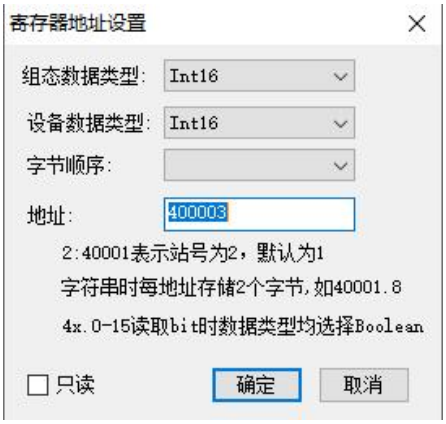
该驱动和 ModbusDTU 类似，接收设备发起的 TCP 连接后开始进行 ModbusRTU 或者 ModbusTCP 通讯，每个设备对应 1 个通讯端口，每个设备使用 1 个线程采集，在实时性方面高于 ModbusDTU 驱动，设备数量支持上少于 ModbusDTU 驱动。

WiFi、4G 等通讯场合建议使用该驱动。

在 Windows 下 1 个进程最大的设备支持量为 300 个。

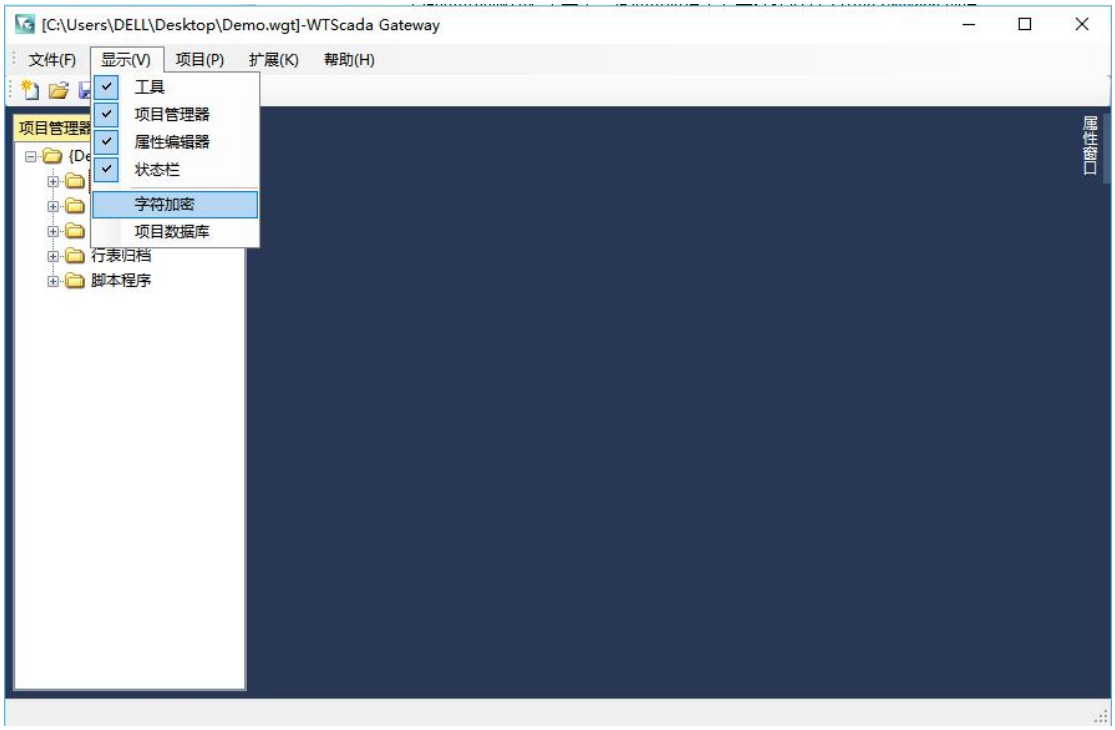
5）Modbus 以太网驱动

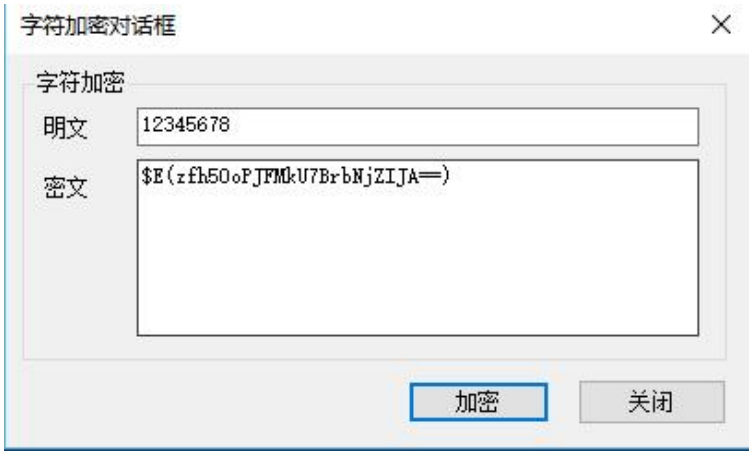
新的 Modubs 以太网驱动，提供 ModbusTCP 和 UDP 通讯，“配置通讯站”按钮可以对每个站提供独立的通讯状态变量和使能变量设置。



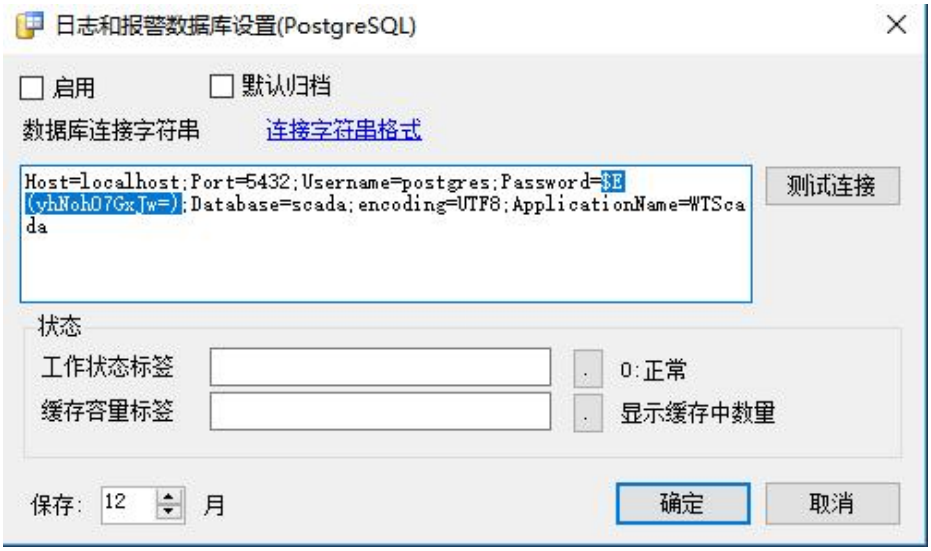
该驱动提供了每个变量提供了独立的字节顺序设置，空白情况下使用驱动全局设置。

4.12 数据库连接字符串加密





该功能用于加密数据库链接字符串的部分内容，常用的方法是加密连接密码，在连接字符串中使用密文替换就可以。

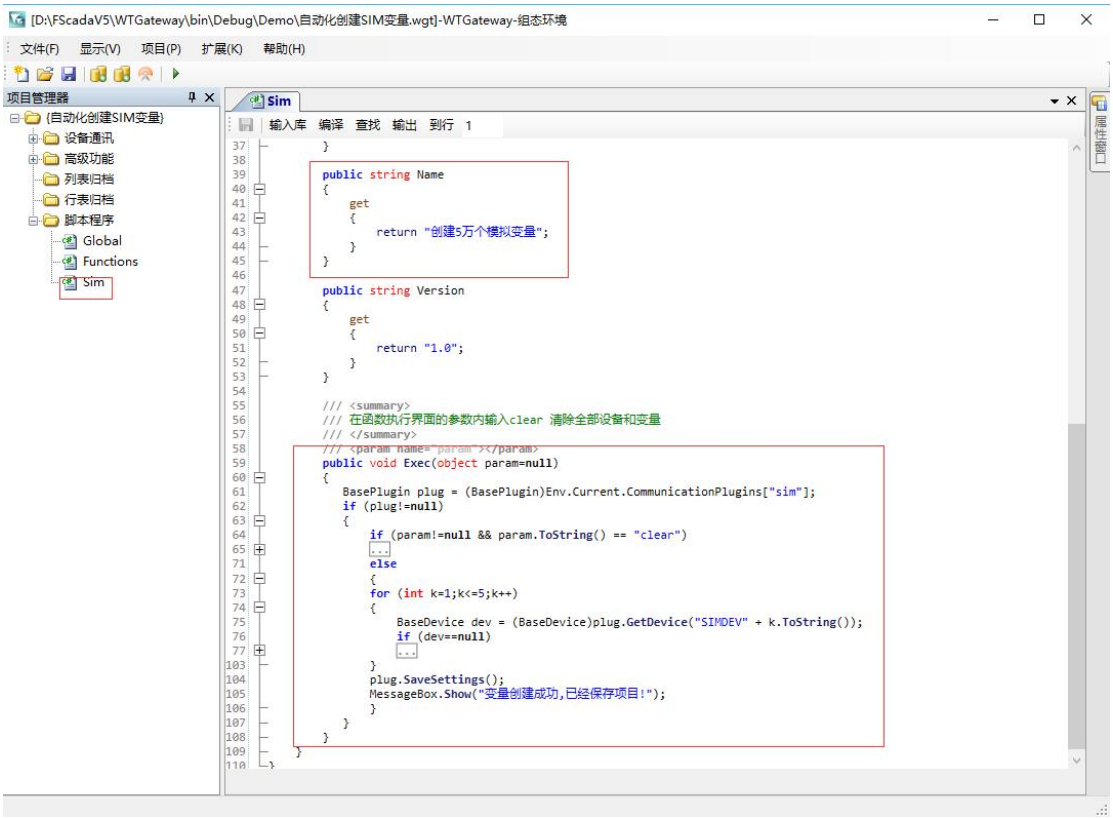


4.13 设计时自动化代码批量创建和复制设备

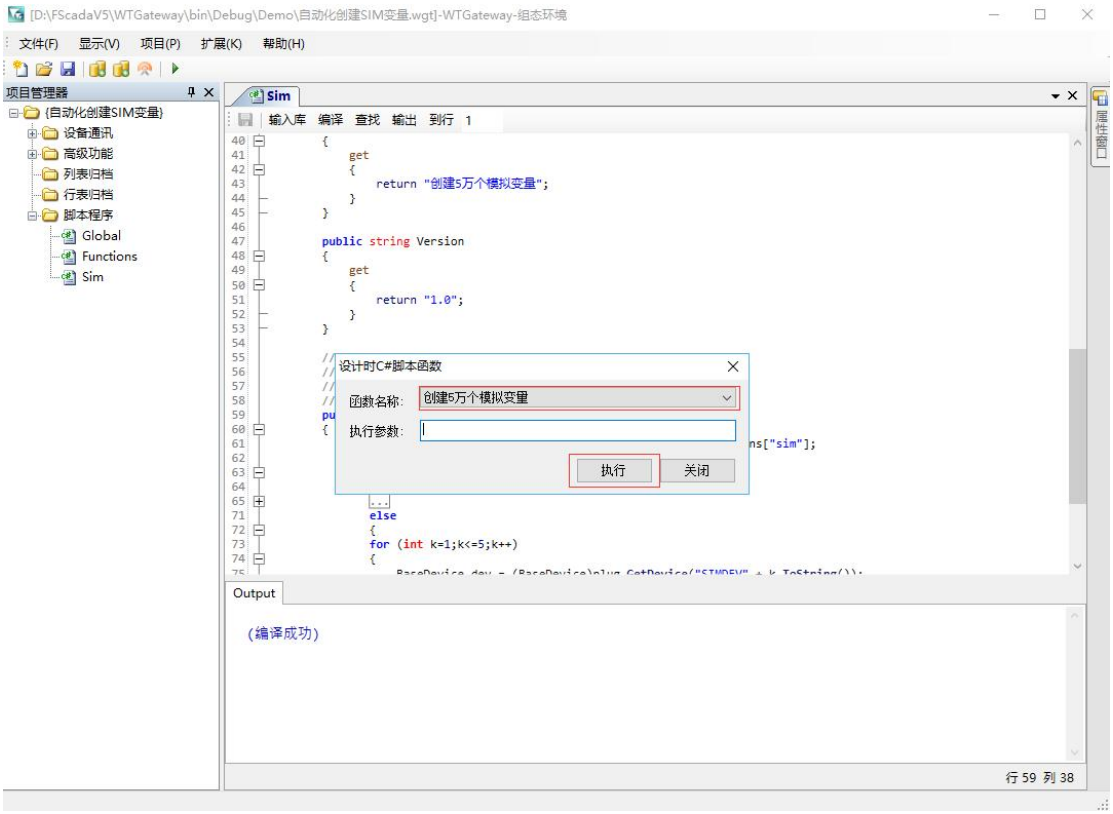
当有大量相同类型的设备需要组态变量时，通过导入导出虽然可以快速完成，但是仍然需要大量的时间，特别是批量修改时还需要再次导入导出，设计时自动化脚本通过 C#的强大功能可以实现在数秒内复制上千个设备，批量的修改变量信息，驱动配置信息。

DEMO 目录下提供了自动创建变量和批量复制设备代码的演示项目。

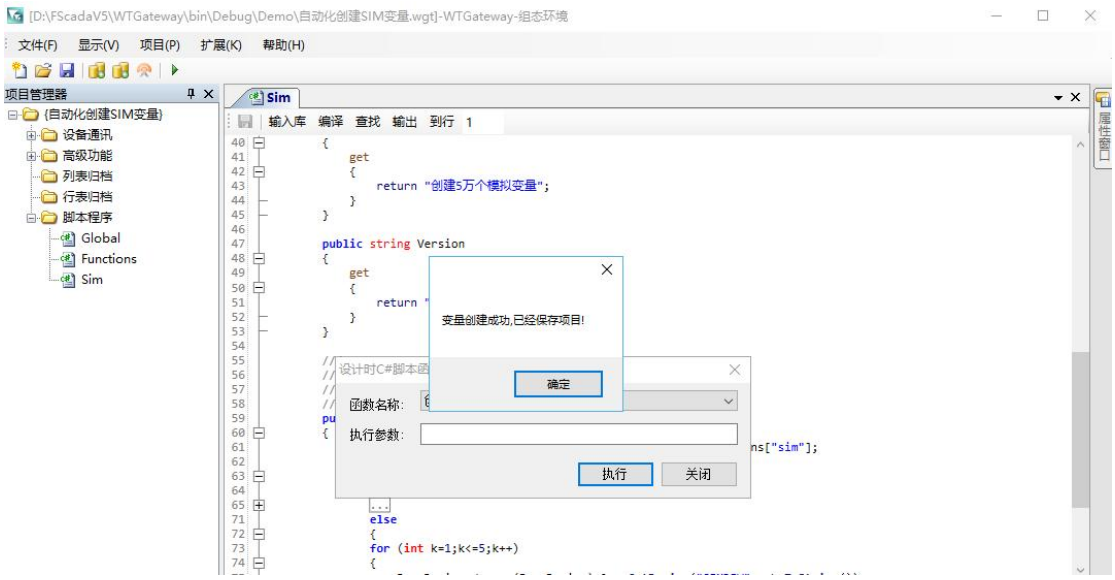
1）打开 DEMO 目录下的“自动化创建 SIM 变量”项目文件



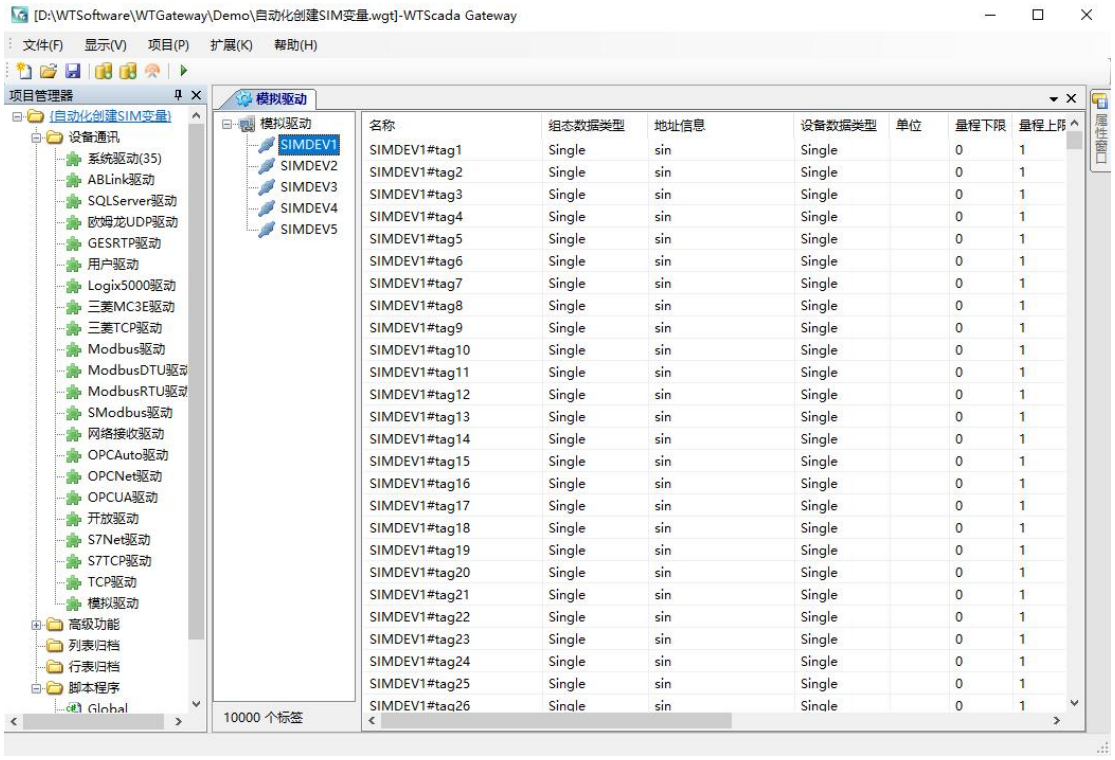
点击工具栏“编译”按钮编译脚本，提示成功后，点击“项目”菜单下的“执行设计时脚本”



点击执行

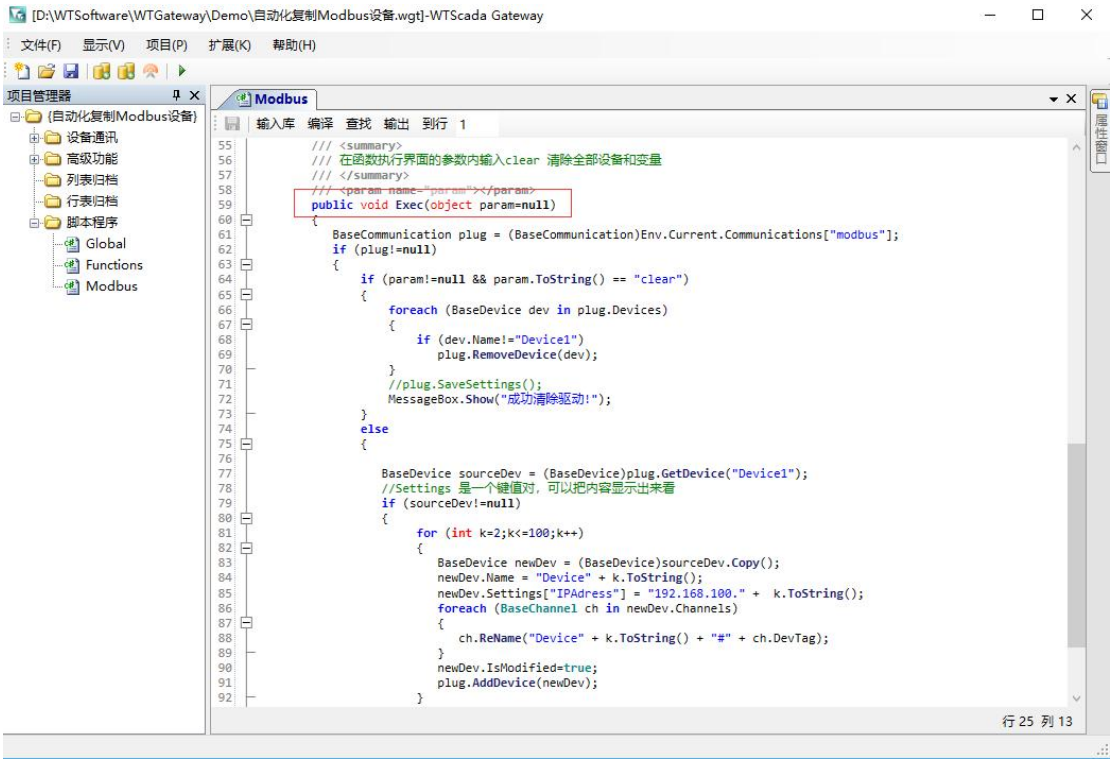


打开模拟驱动

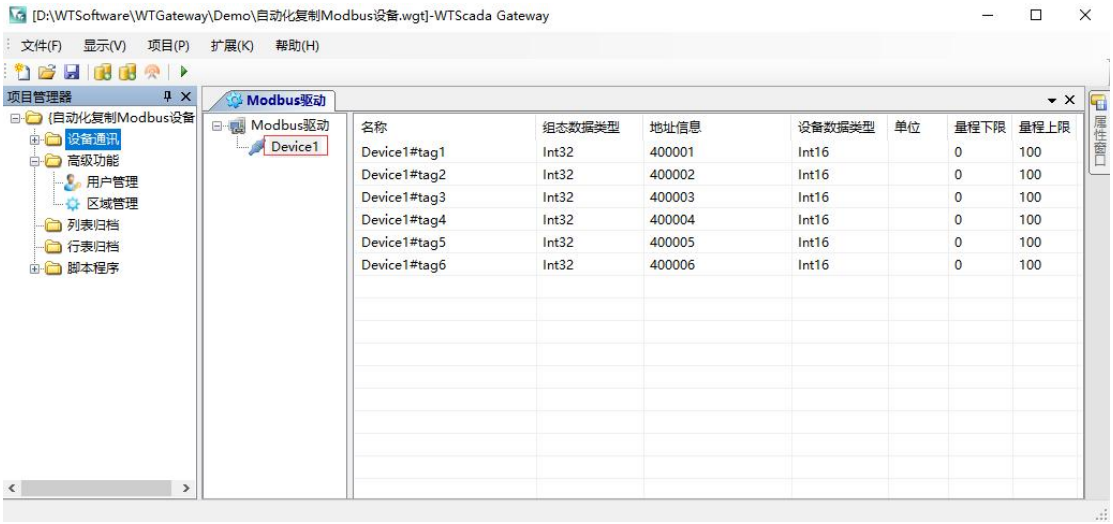


看到创建了 5 个设备，每个设备 10000 个变量

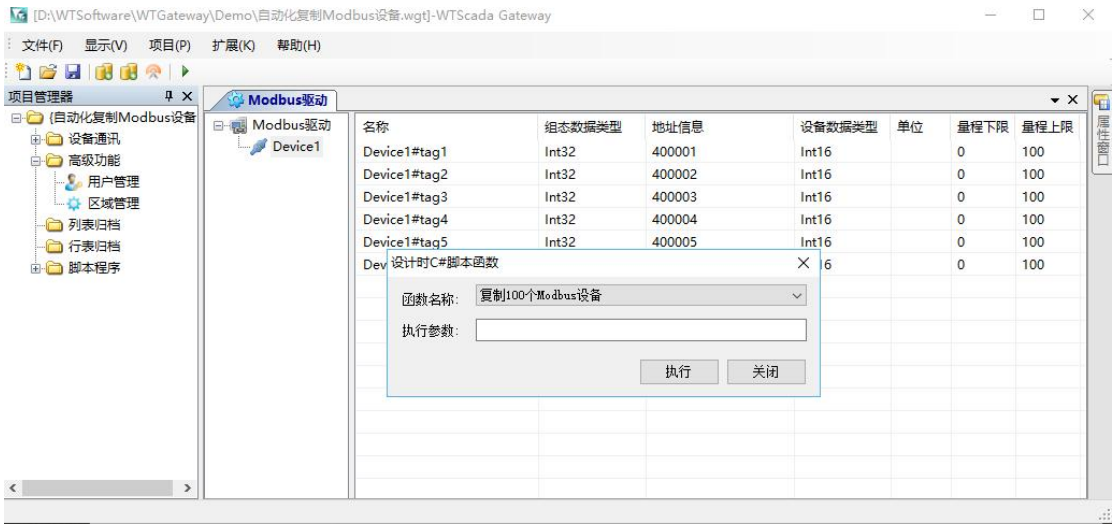
2）打开 DEMO 目录下的“自动化复制 Modbus 设备”项目



该演示从已经创建的 Modbus 驱动设备名称为 Device1 进行复制



编译成功，点击“项目”菜单下的“执行设计时脚本”

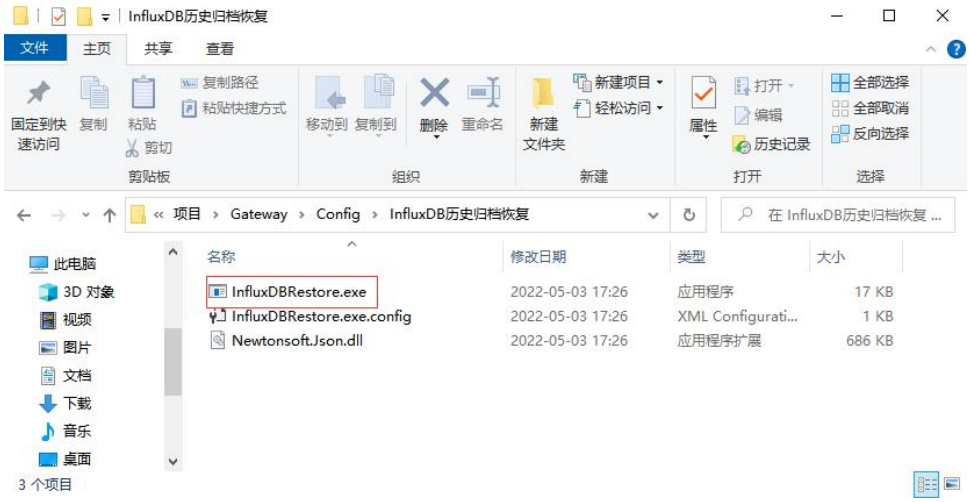


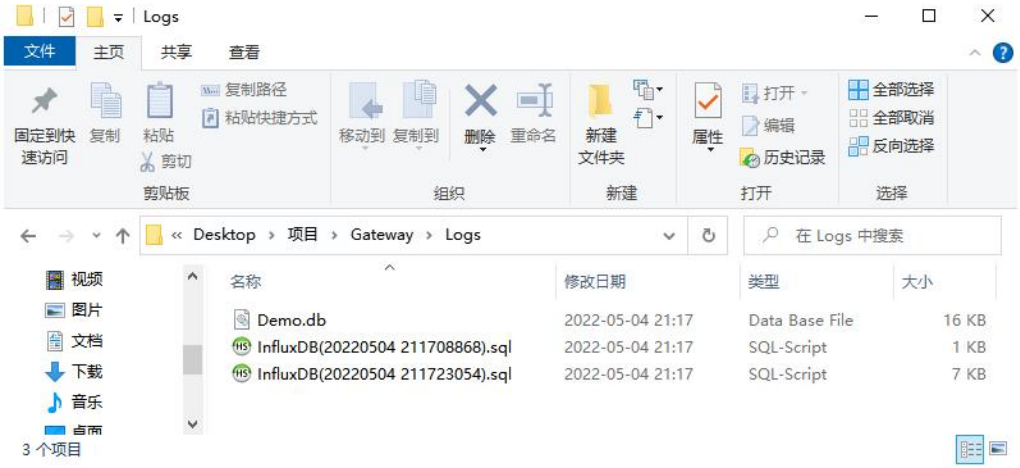
打开驱动配置可以看到 100 个 Modubs 设备已经复制完成

演示项目代码中包括了删除设备的代码，执行对话框的参数中输入”clear”执行就可以删除设备。

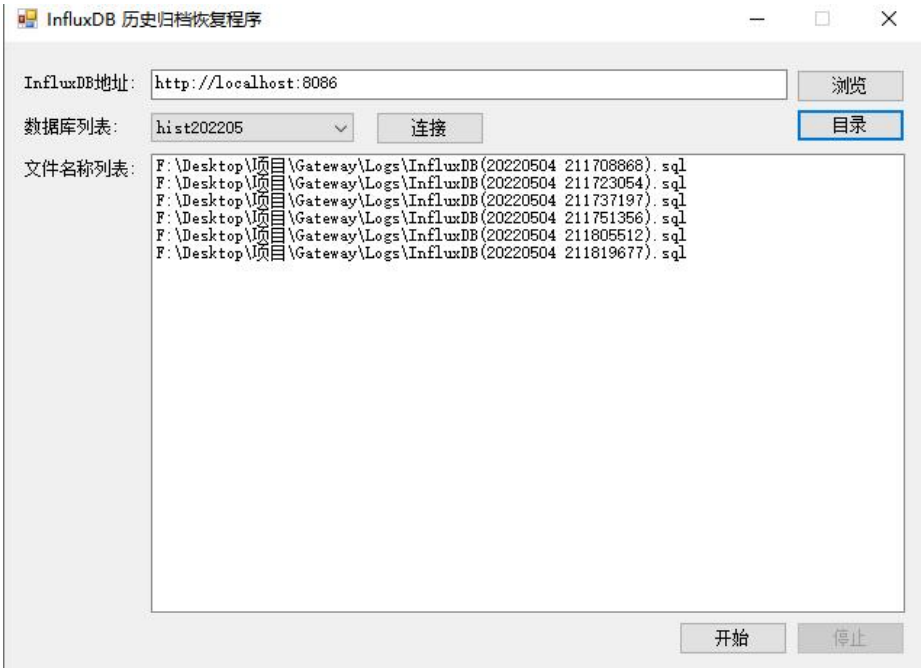
4.14 InfluxDB 历史恢复功能

InfluxDB 历史归档在写入失败后会把日志文件记录到 Gateway 软件的 Logs 目录下，使用 Config 目录下 InfluxDB 恢复程序可以从日志文件中把历史数据恢复到数据库中。




Logs
文件
主页
共享
查看
固定到快速访问
复制
粘贴
复制路径
粘贴快捷方式
剪切
剪贴板
移动到
复制到
删除
重命名
组织
新建文件夹
新建
属性
打开
编辑
历史记录
打开
全部选择
全部取消
反向选择
选择
« Desktop › 项目 › Gateway › Logs
在 Logs 中搜索
视频
图片
文档
下载
音乐
名称
修改日期
类型
大小
Demo.db
2022-05-04 21:17
Data Base File
16 KB
InfluxDB(20220504 211708868).sql
2022-05-04 21:17
SQL-Script
1 KB
InfluxDB(20220504 211723054).sql
2022-05-04 21:17
SQL-Script
7 KB
3 个项目


InfluxDB 历史归档恢复程序
InfluxDB地址:
http://localhost:8086
浏览
数据库列表:
hist202205
连接
目录
文件名称列表:
2 点目录或者浏览加入历史日志文件
1 连接成功后会显示数据库列表
3 点开始执行
开始
停止



恢复完成后 Logs 下的文件会自动删除，如果 InfluxDB 配置了用户名密码修改连接地址格式如:http://localhost:8086;admin;admin

为防止批量恢复失败，建议先选几个文件恢复，然后使用查询工具查询到数据后再批量执行。

注意：如果需要恢复的数据库不存在，则不会恢复，文件也不会删除。

4.15 OPCLink

1、软件用途

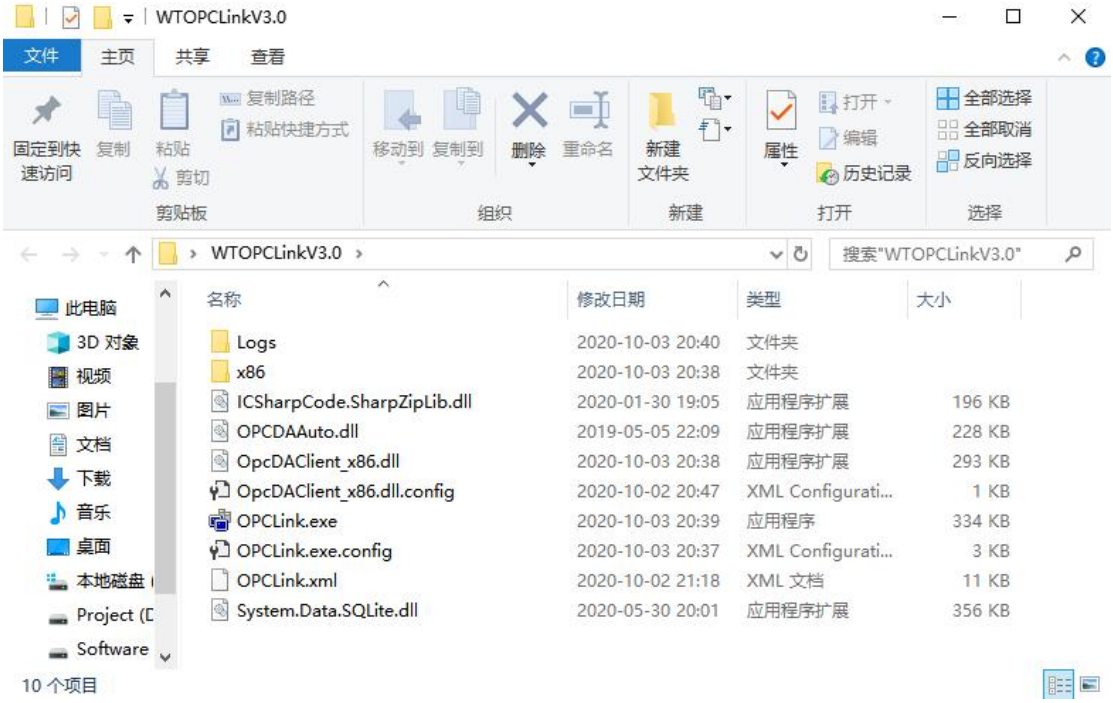
OPCLink 软件是 1 个 OPCDA 客户端数据采集程序，除了标准的 OPCDA 客户端功能外还提供了 TCP 服务和网络转发功能，设计该软件的目的是为了解决远程 OPC 难以配置或者穿透防火墙或者网络隔离装置，通过把 OPCLink 软件安装在 OPC 服务器上进行本地采集，然后通过可控制端口的 TCP 或者 UDP 模式把数据发送出去，另外 TCP 服务功能和 WTOPCServer 软件配合可以在远程电脑上提供 OPC 服务器功能。

OPCLink 支持断点续传，以保障网络故障后历史数据的完整性。

OPCLink 转发功能是免费的，转发功能支持 Gateway 和 FScada4.18 及后续版本，服务器功能和断点续传功能需要授权才能持续运行。

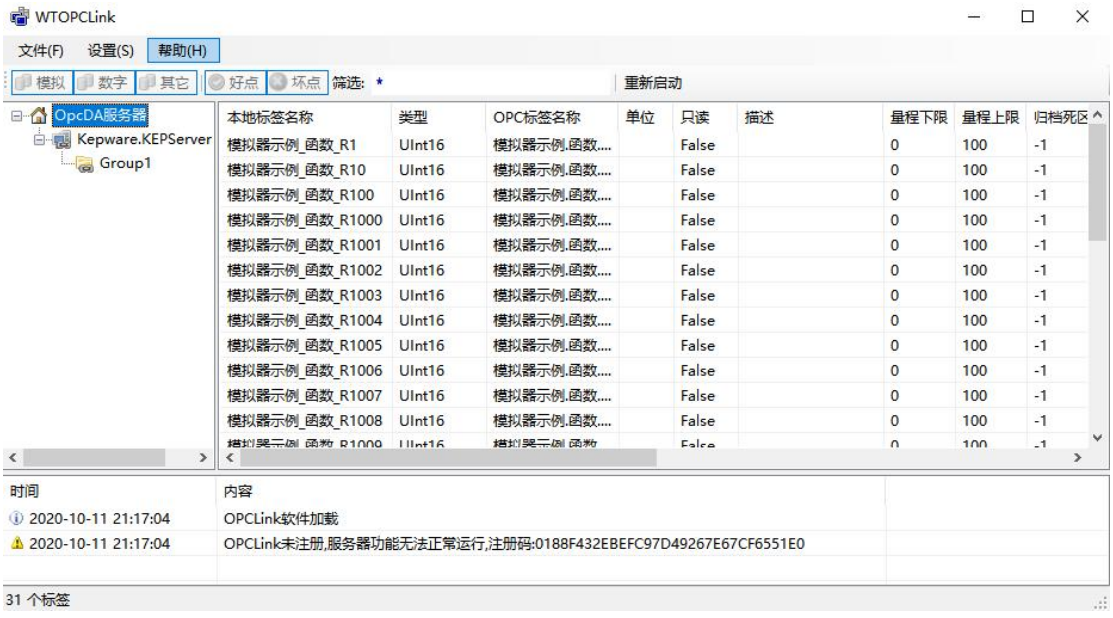
运行环境:WindowXp SP3 及后续版本，安装 dotNet4.0 或更高版本，安装 opcda（x86）组件包。

2、软件目录



OPCNetLink.exe 是执行程序，opc 目录下提供了 opc 组件发行包和第三方的标准 opc 客户端工具软件，Logs 目录存储系统日志，配置文件 opclink.xml 存储在根目录下。

3、软件配置



1）添加 OPC 服务器

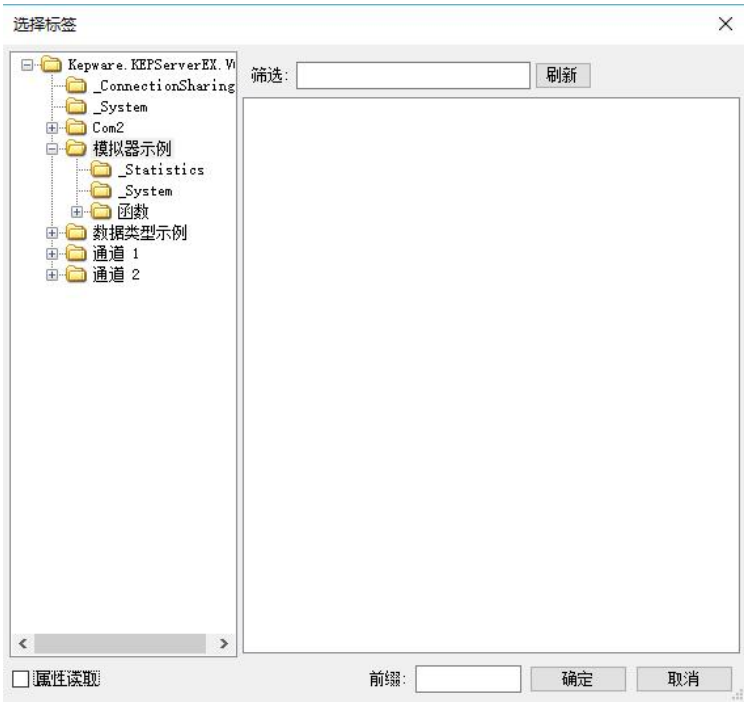
心跳检测（默认启用）：该功能定时检测 OPC 服务器状态，启用后如何 OPC 服务器异常奔溃才能被检测到，进行重新连接。

重启检测（默认关闭）：该功能检测 OPC 数据更新，如果超过设置的时间数据没有更新，就会自动重新启动软件，用于特殊的不稳定的 OPC 服务器。

使用服务器时间：选择后变量的时间戳使用 OPC 服务器的变量时间，不选择使用本机时间。

2）添加 OPC 通讯组

3）添加通讯变量

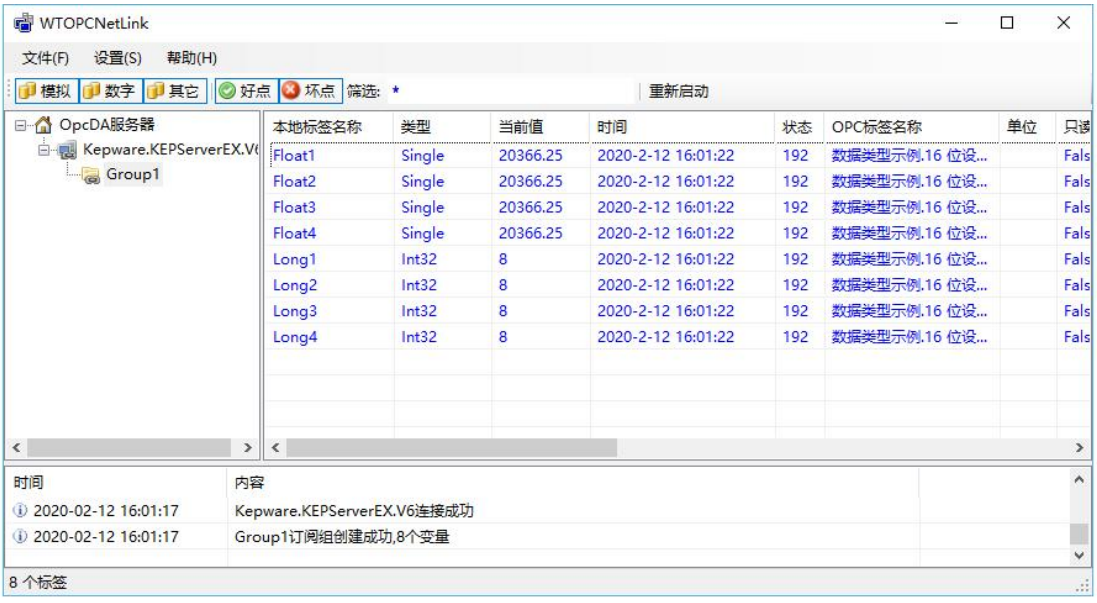


属性读取：并非所有的 OPC 服务器支持属性读取，一般通讯正常的能支持，不正常的会卡涩，属性读取功能主要用于变量信息的读取，如果变量很多启动后浏览变量会很慢。如果没有使用属性读取添加后的变量数据类型为 Object，保存运行一次读取到值后变量的数据类型就会自动更新为 OPC 服务器变量类型。

在软件启用转发获取服务功能前必须保证变量类型正确，不是 Object 类型。

配置界面的数据鼠标右键复制后可以粘贴到 Excel 进行编辑和修改，然后再粘贴回到配置界面，也可以使用导入导出 CSV 功能进行编辑。

4）保存，运行



设置菜单下有“自动启动”的选项，设置自动启动后，运行软件就会处于运行模式，默认是配置模式。

设置菜单下的“转发设置”，运行时可配置更新

通过鼠标右键菜单可以添加 TCP 或者 UDP 发送配置，发送的目标可以是 FScada 的网络接收驱动、WTGateway 的网络接收驱动、WTScada HTML5 的 IOServer 接口，变量信息自动推送，动态添加的变量也会实时同步发送。

TCP 模式和 UDP 模式的区别：TCP 模式是数据变化就发送，UDP 除了数据变化就发送，没有变化的数据每分钟还会发送一次。

TCP 模式工作过程：运行后连接到设置的服务器端口上，使用名称进行登录验证，发送变量列表，之后开始检测数据变化，变化就发送。

TCP 模式下断点续传文件名称不为空白就表示启用断点续传功能，使用断点续传功能时变量的归档死区必须大于等于 0，设置合适的例外时间和死区属性，断点续传本质上是在本地存储配置了历史归档属性的历史数据。

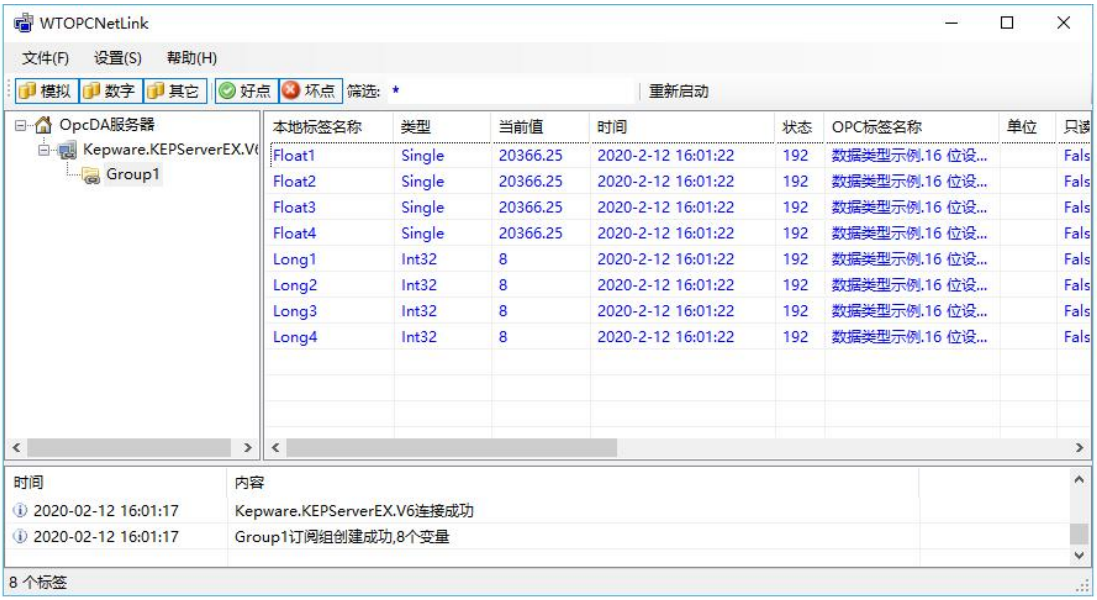
UDP 模式工作过程：运行后发送变量列表，之后开始检测数据变化，变化就发送，如果变量 1 分钟没有数据变化则发送一次。

设置菜单下的“服务设置”，运行时可配置更新

该服务提供 TCP 服务功能，该功能启用后可以使用 FScada 或者 WTGateway 的

NetTCP 驱动连接采集数据，该功能需要授权，没有授权不要启用该功能。

4、在线添加和删除 OPC 变量



运行模式下选中 OPC 组后，右侧的界面上鼠标右键添加和删除功能就可以使用。该表格界面在运行模式下鼠标双击可以修改除名称和实时值外的其他列，修改好后选中一些行可以使用鼠标右键“发送更新”菜单同步变量信息。

在线添加的变量会自动发送出去，删除的变量信息不会同步。

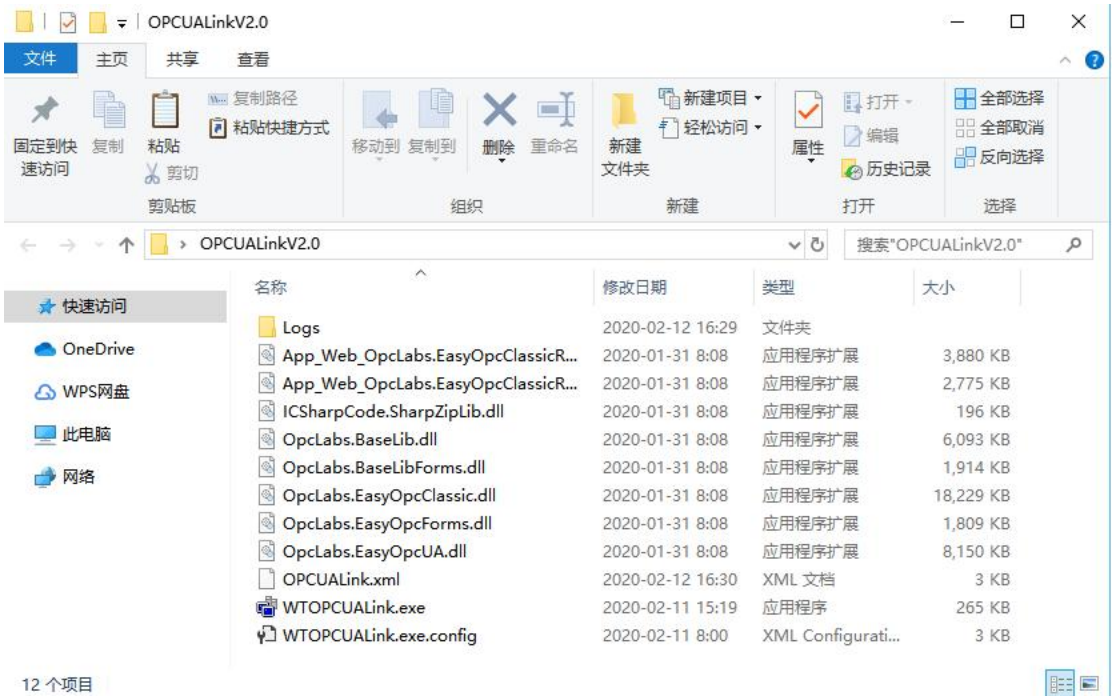
4.16 OPCUALink

1、软件用途

OPCUALink 软件是 1 个 OPCUA 客户端数据采集程序，除了标准的 OPCUA 客户端功能外还提供了 TCP 服务和网络转发功能，设计该软件的目的是为了实现分布式采集和转发，OPCLink 软件采集数据后，通过可控制端口的 TCP 或者 UDP 模式把数据发送出去，另外 TCP 服务功能和 WTOPCServer 软件配合可以在远程电脑上提供 OPCDA 服务器功能。

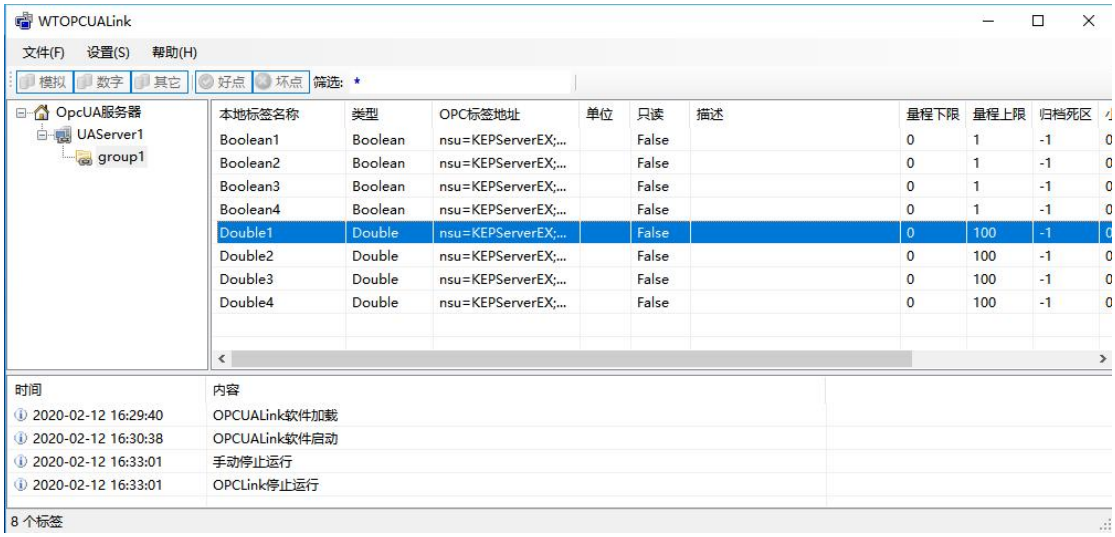
运行环境:Window7 SP1 及后续版本，安装 dotNet4.5.2 或更高版本

2、软件目录

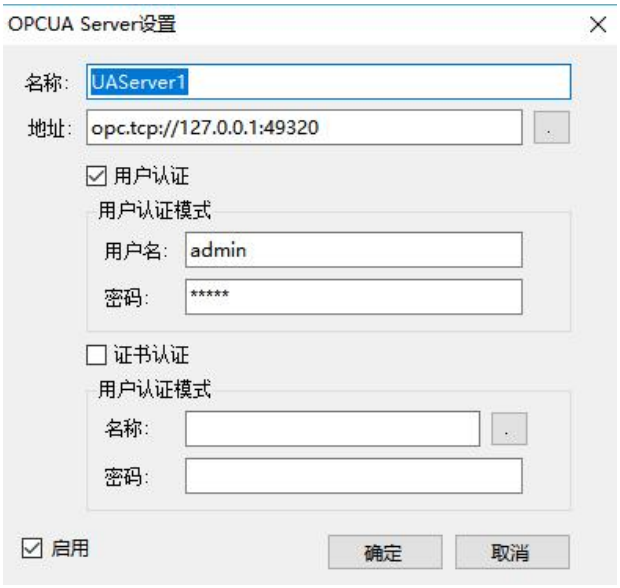


WTOPCUALink.exe 是执行程序,Logs 目录存储系统日志,配置文件 opcualink.xml 也存储根目录下。

3、软件配置

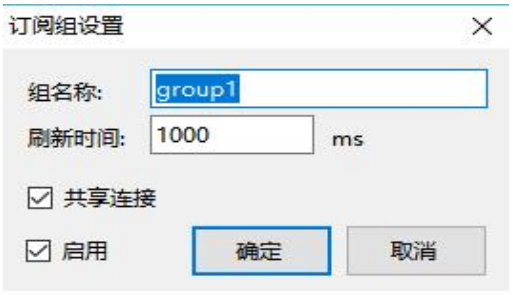


1）添加 OPCUA 服务器

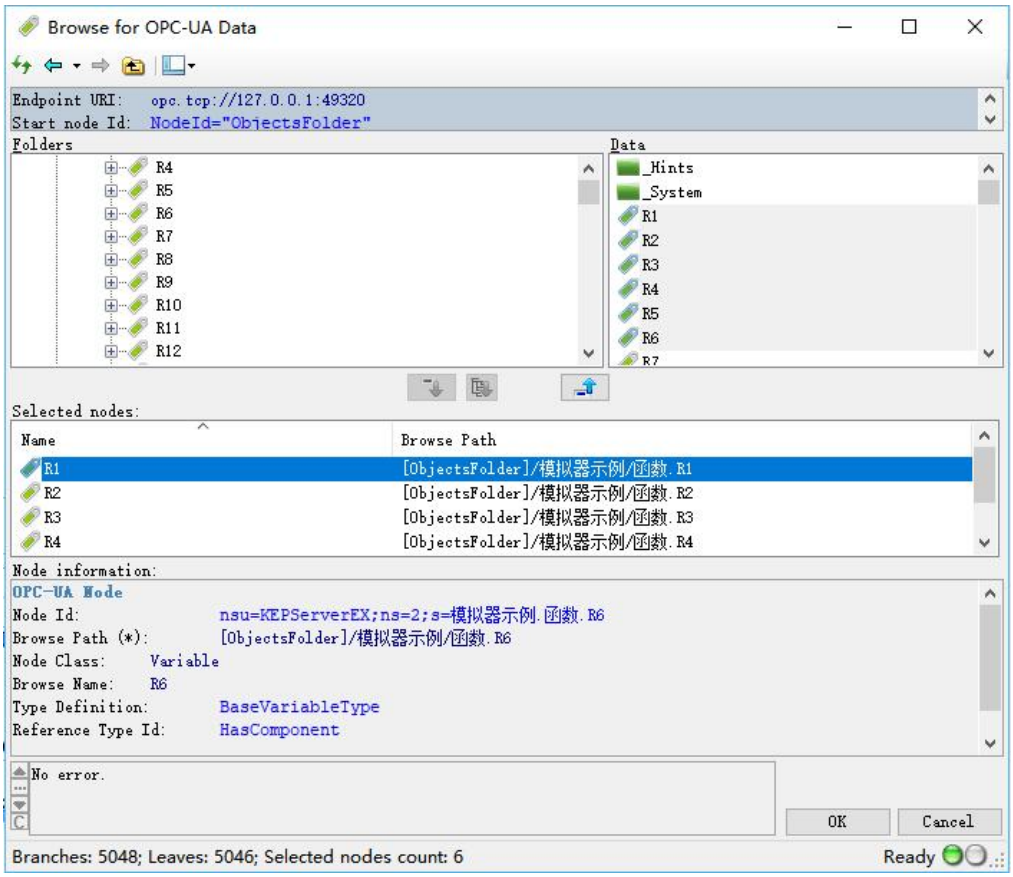


变量的时间戳使用本机时间。

2）添加 OPC 通讯组

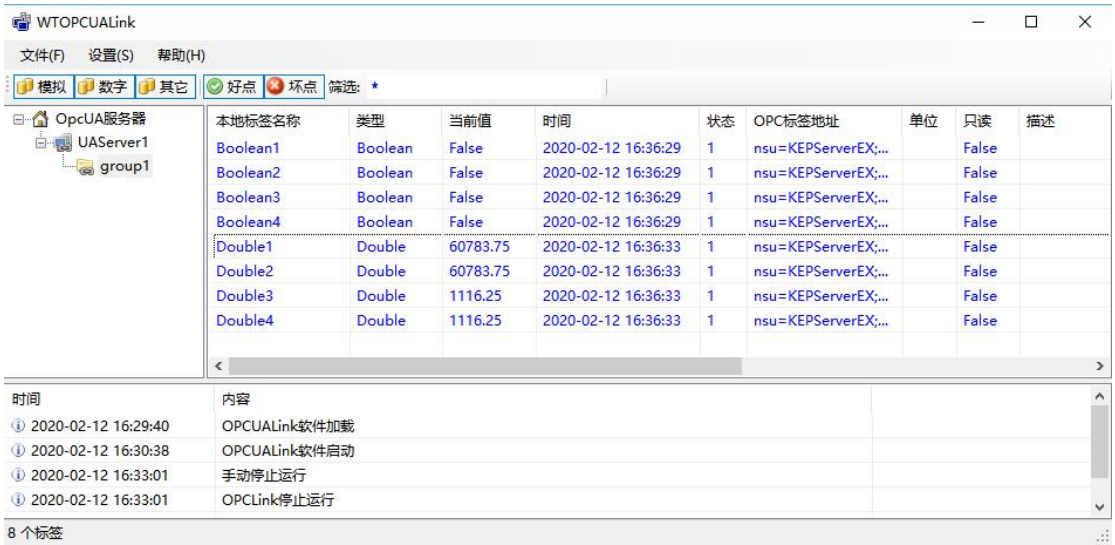


3）添加通讯变量



配置界面的数据鼠标右键复制后可以粘贴到 Excel 进行编辑和修改，然后再粘贴回到配置界面，也可以使用导入导出 CSV 功能进行编辑。

4）保存，运行



设置菜单下有“自动启动”的选项，设置自动启动后，运行软件就会处于运行模式，默认是配置模式。

设置菜单下的“转发设置”，运行时可配置更新

通过鼠标右键菜单可以添加 TCP 或者 UDP 发送配置，发送的目标可以是 FScada 的网络接收驱动、WTGateway 的网络接收驱动、WTScada HTML5 的 IOServer 接口，变量信息自动推送，动态添加的变量也会实时同步发送。

TCP 模式和 UDP 模式的区别：TCP 模式是数据变化就发送，UDP 除了数据变化就发送，没有变化的数据每分钟还会发送一次。

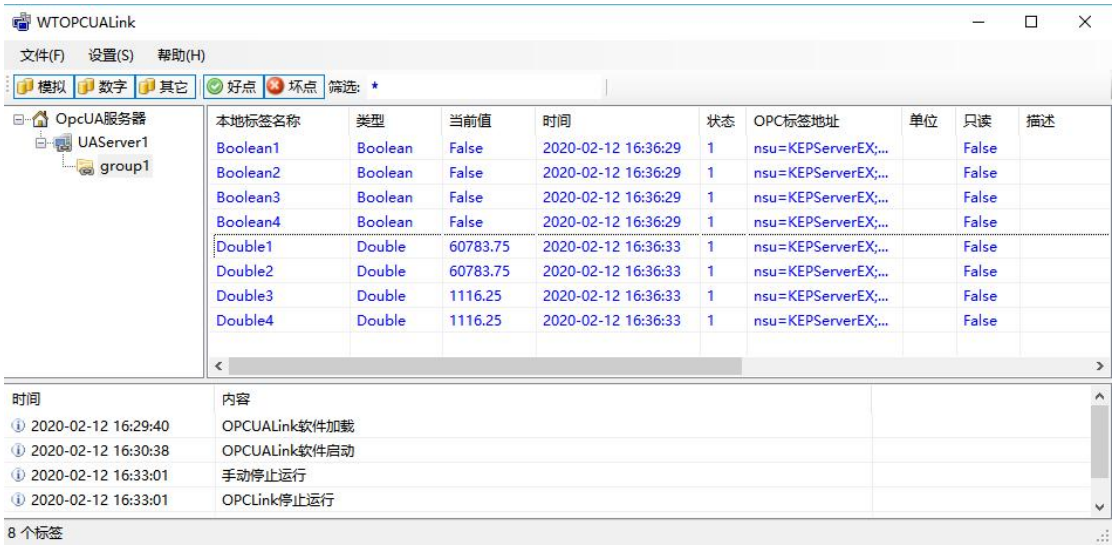
TCP 模式工作过程：运行后连接到设置的服务器端口上，使用名称进行登录验证，发送变量列表，之后开始检测数据变化，变化就发送。

UDP 模式工作过程：运行后发送变量列表，之后开始检测数据变化，变化就发送，如果变量 1 分钟没有数据变化则发送一次。

设置菜单下的“服务设置”，运行时可配置更新

该服务提供 TCP 服务功能，该功能启用后可以使用 FScada 或者 WTGateway 的 NetTCP 驱动连接采集数据，该功能需要授权，没有授权不要启用该功能。

4、在线添加和删除 OPC 变量



运行模式下选中 OPC 组后，右侧的界面上鼠标右键添加和删除功能就可以使用。该表格界面在运行模式下鼠标双击可以修改除名称和实时值外的其他列，修改好后选中一些行可以使用鼠标右键“发送更新”菜单同步变量信息。

在线添加的变量会自动发送出去，删除的变量信息不会同步。

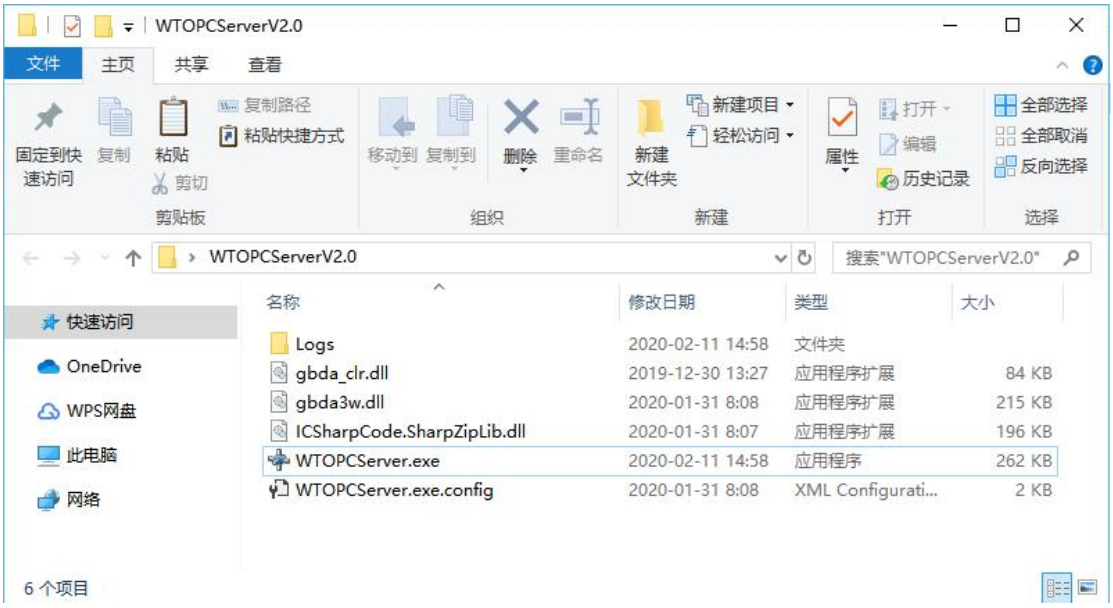
4.17 WTOPCServer

1. 软件用途

WTOPCServer 软件是 1 个 OPCDA 服务器程序，可以通过 TCP 通讯从 OPCLink 软件或者 FScada、WTGateway 的 TCP 服务接口获取实时数据，提供本地的 OPC 服务器功能。

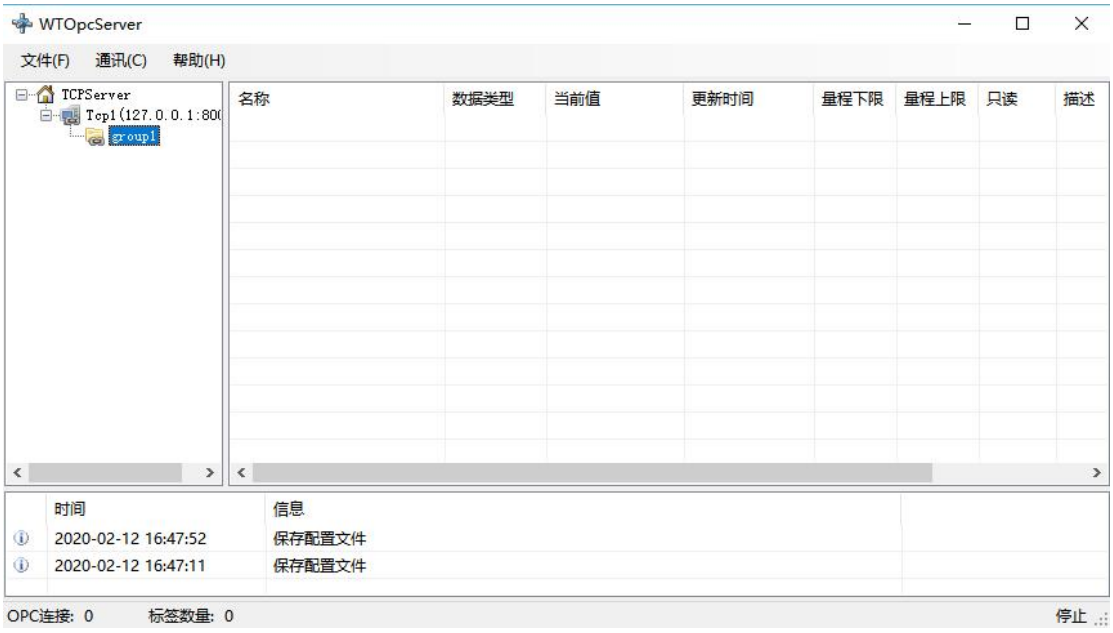
运行环境:WindowXp SP3 及后续版本，安装 dotNet4.0 或更高版本，安装 opcda（x86）组件包，安装 vc2012 x86 版本。

2. 软件目录

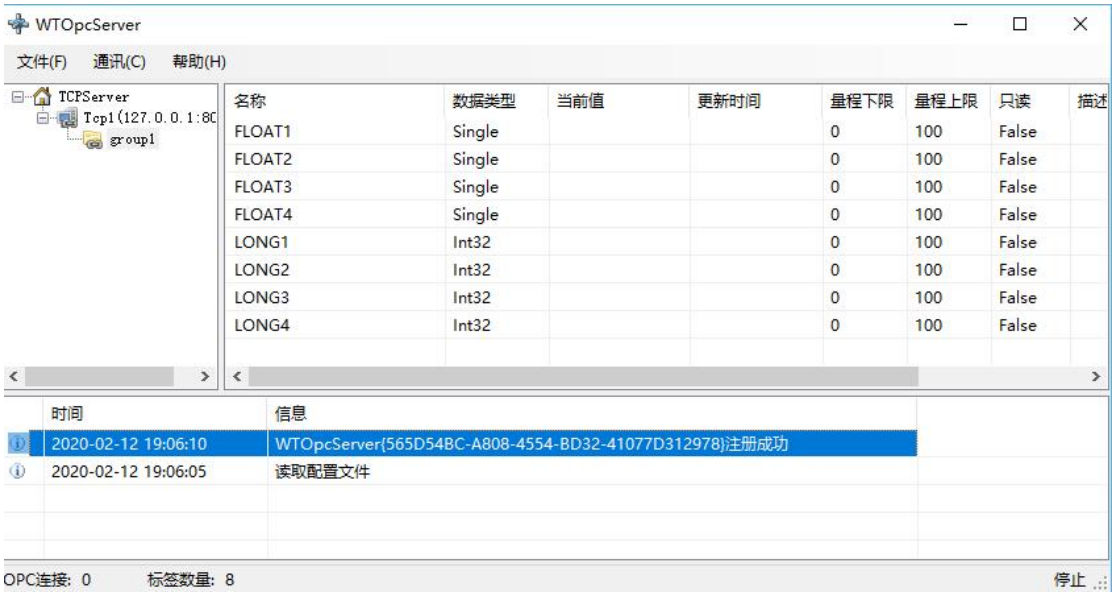


WTOPCServer.exe 是执行程序，Logs 目录存储系统日志，配置文件 config.xml 存储在根目录下。

3. 软件配置



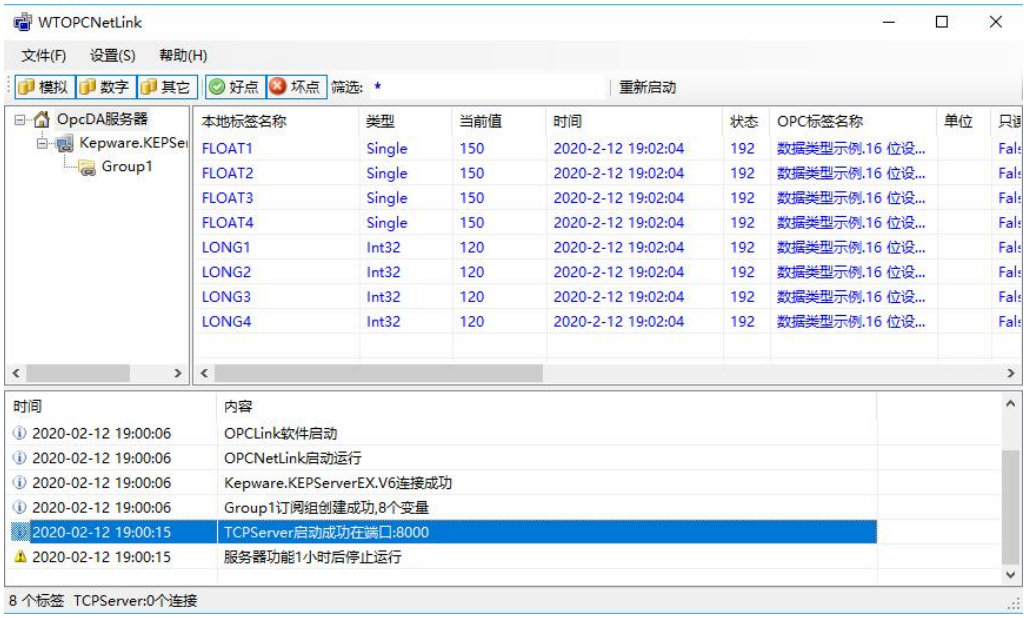
以管理方式启用后，点击通讯菜单下注册 OPC Server



成功注册 OPCServer

1）添加服务器

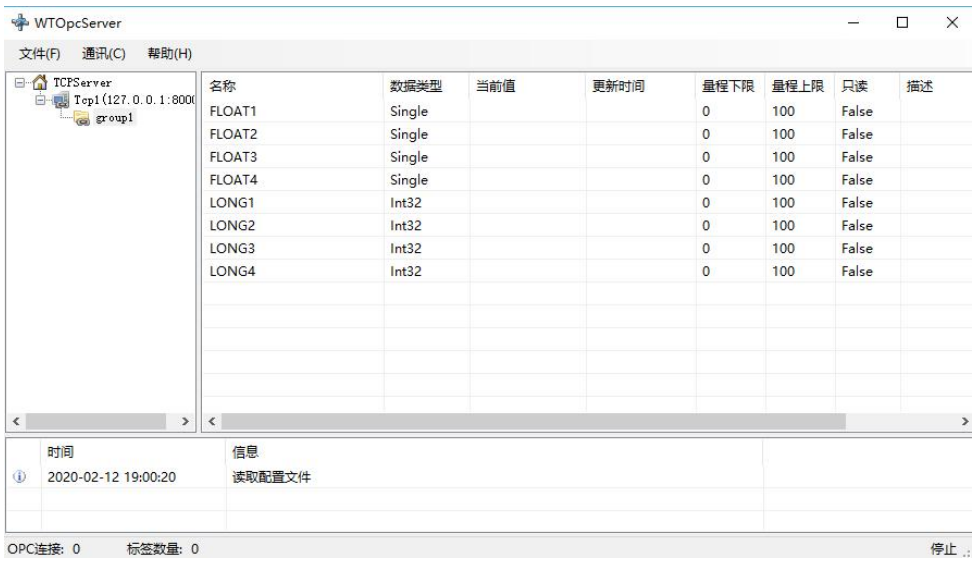
OPCLink 服务设置



OPCLink 已经在 8000 端口启用了 TCP 服务

2）添加通讯组

3）从服务器导入选择的变量

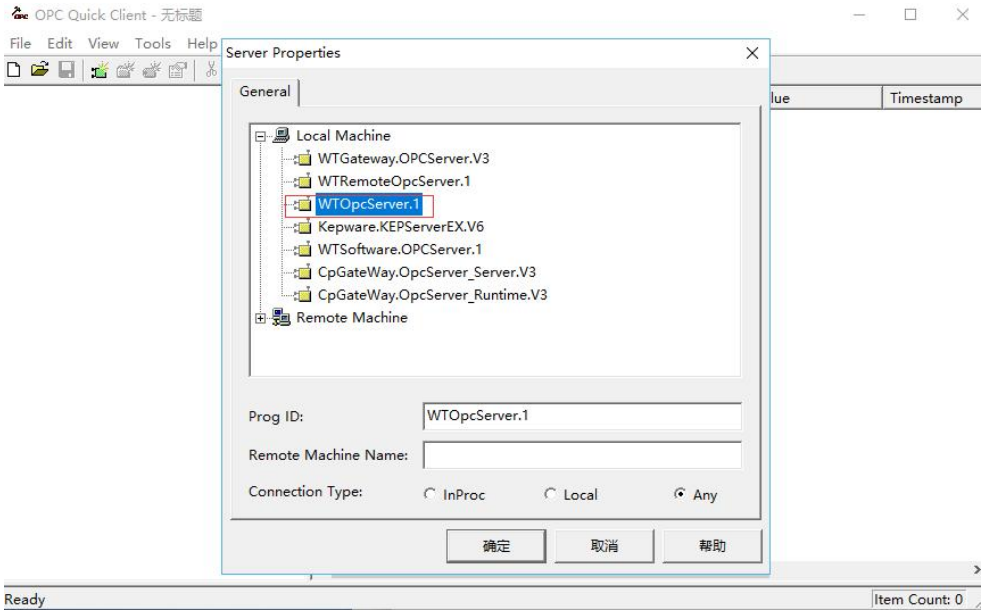


4）保存，运行

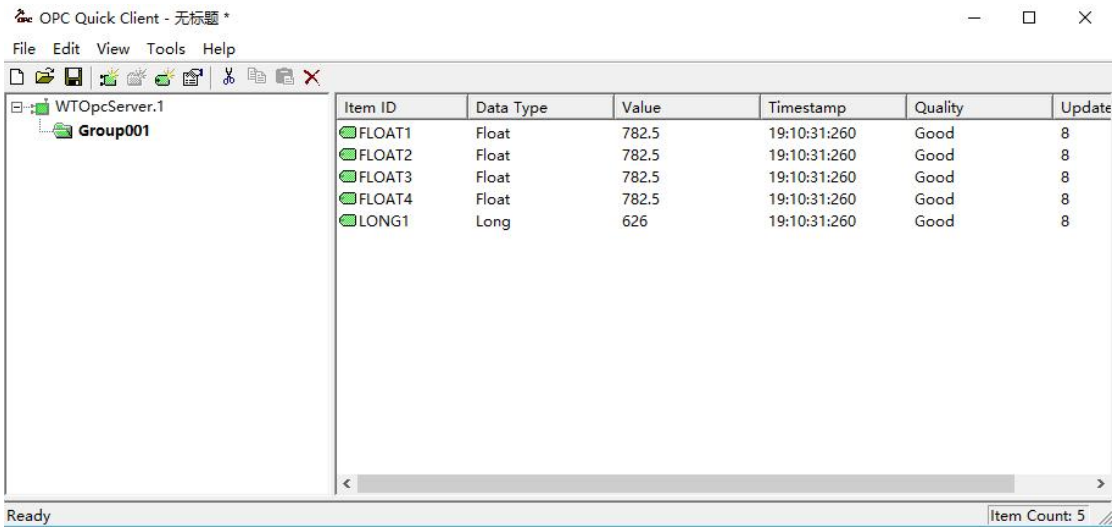
可以在线添加变量，软件启动运行后在线添加的变量可以被 OPCClient 直接找到，不需要重新启动 OPC 服务器或者客户端软件。

4. 保存关闭 WTOPCServer

运行 OPC 客户端工具软件，连接 WTOPCServer



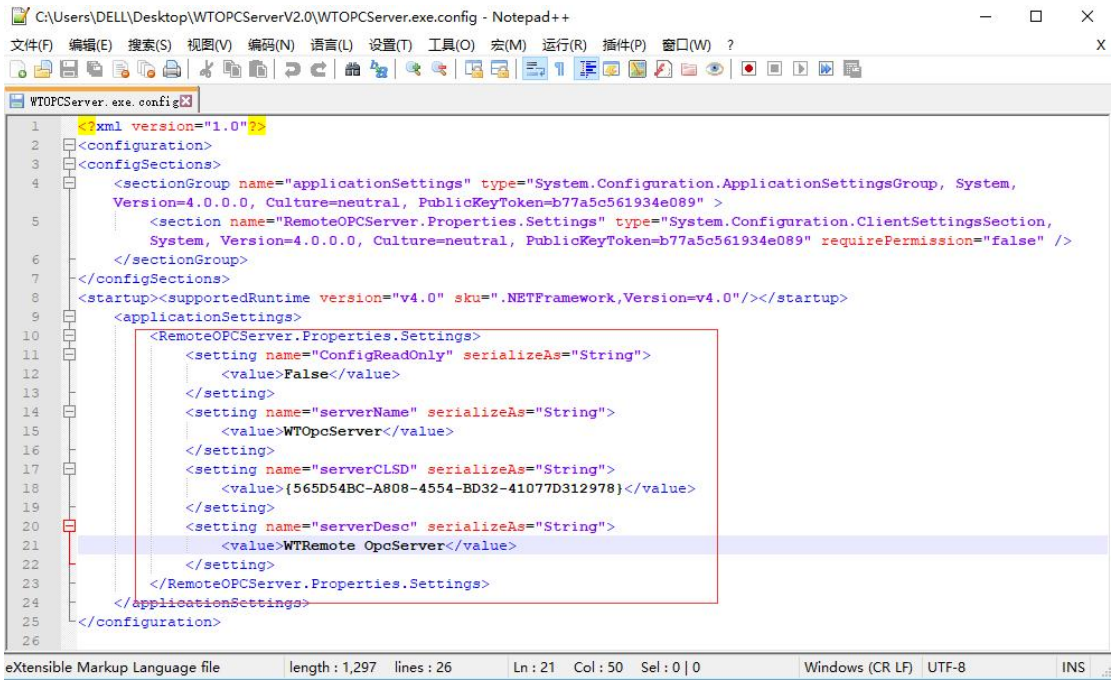
可以看到 WTOPCServer 自动启动，最小化在任务图标上



注意：手动运行 WTOPCServer 直接进入配置模式，需要点击菜单的“启动”才会启动 OPCServer 服务，被 OPCClient 启动时自动运行。

5. OPCServer 的注册信息修改

OPC 服务器的注册信息可以通过 WTOPCServer.exe.config 文件进行修改



4.18 WTScada Gateway SDK

SDK 目录下包含了 C#版本的 SDK 库和演示项目

SDK.dll 是 SDK 的库文件

DataServiceDemo 是数据服务端口使用演示（TCP 方式，实时数据获取采用查询方式）

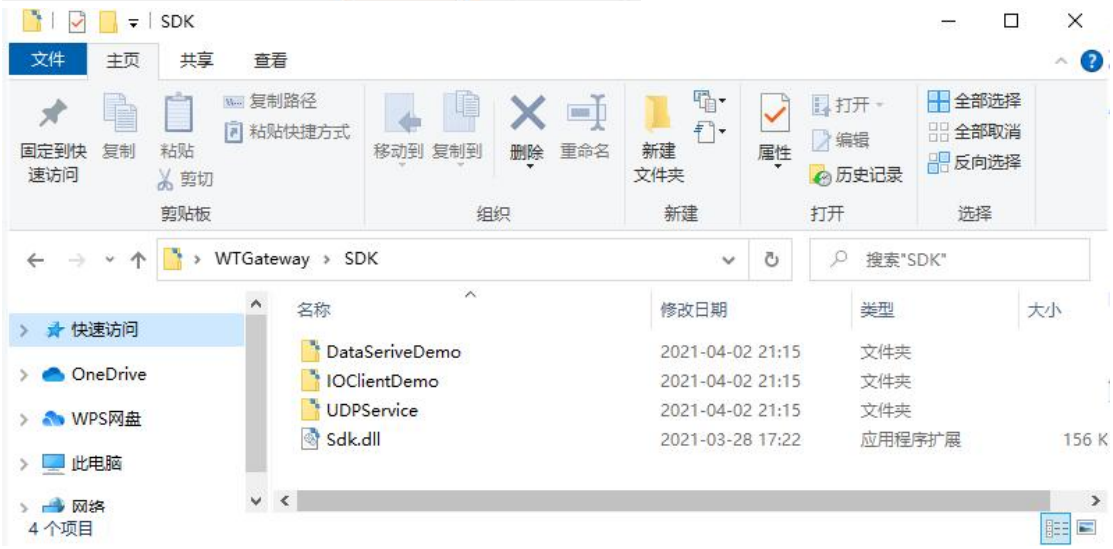
服务配置界面如下：

IOClientDemo 是 IO 服务端口使用演示（TCP 方式，实时数据获取由组态软件自动推送）

服务配置界面如下：

UDPService 是 UDP 数据服务使用演示（UDP 方式，实时数据获取采用查询方式）

服务配置界面如下：



4.19 WebSocket 交互数据格式

```
{ "msgtype",//string
  "p1", //string
  "p2",//string
  "p3",//string
  "p4",//string
  "p5",//string
  "ps":[]//string[]
}
```

查询实时数据：msgtype 设置为 tagvalues，ps 为变量数组

写入：msgtype 设置为 write,p1,p2 分别设置变量名和变量值

批量写入：msgtype 设置为 writes,ps 设置为多变量值

登录：msgtype 设置为 login,p1,p2 分别设置为用户名和 md5 计算的密码

变量列表：msgtype 设置为 tags,p1,p2 分别设置为开始序号和结束序号（从 1 开始）

订阅变量：msgtype 设置为 startsubscribe，p1 为发送周期，ps 为变量数组

取消订阅：msgtype 设置为 stopsubscribe，ps 为变量数组

读取历史数据：msgtype 设置为 histquery_trend，p1 为开始时间，p2 为时间长度，ps 为变量数组

返回数据包括 ret，非 0 异常

举例：查询{"msgtype":"tagvalues","ps":["tag1","tag2"]}

设置{"msgtype":"write","p1":"tag1","p2":"100"}

批量设置{"msgtype":"writes","ps":["tag1=1","tag2=4"]}

变量列表{"msgtype":"tags","p1":"1","p2":"100"}

登录{"msgtype":"login","p1":"admin","p2":"xxxxxxxxxxxxx"}

订阅{"msgtype":"startsubscribe","p1":"1000","ps":["tag1","tag2"]}

取消订阅:{"msgtype":"stopsubscribe"}

读取历史趋势:{"msgtype":"histquery_trend","p1":"2021-1-1 12:00:00","p2":"3600","ps":["tag1","tag2"]}

如果不需要设置变量值，可以不进行登陆，

Http 目录下的 websocket.html 提供了完整的功能演示。

4.20 对外提供数据的方式汇总

1）通过 UDP 服务对外提供实时数据访问

2）通过 WTOPCServer 提供 OPCDA 访问服务

3）通过 OPCUA 扩展提供 OPC UA 访问

4）通过关系库同步扩展提供实时数据访问

5）通过 SDK 获取实时、历史数据访问

6）通过 MQTT 发送到 MQTT 服务器

7）通过 REDIS 扩展同步实时数据到 REDIS 服务器

8）通过 Modbus 从站扩展提供 Modbus 访问

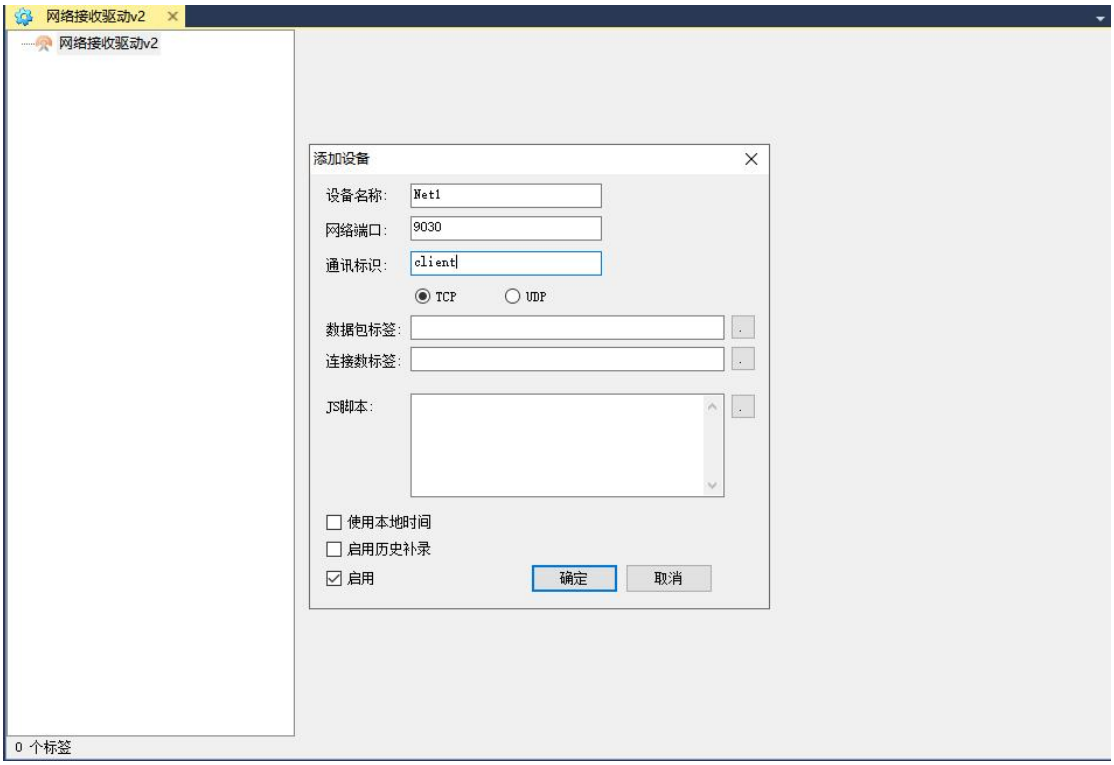
9）通过 REST web api 提供数据访问

4.21 网络接收驱动用途

网络接收驱动的功能如下：

1）接收 OPCLink 软件的实时数据（转发配置）

2）接收 FScada、Gateway 软件的转发数据（转发扩展）



TCP 模式支持反向控制，是首选方式，1 个 TCP 接收需要使用 2 个连续的 TCP 端口。

通讯标识仅用于 TCP 模式，相当于验证密码，发送方和接收方必须设置相同的字符。

UDP 模式是单向的，接收方无法对数据进行写值操作。

该驱动还可用于云端数据接收。

数据包标签：Int32 类型，用于指示网络接收数据包的数量

连接数标签：Int32 类型，TCP 方式下当前连接数量，1 个 TCP 服务可以支持多个网络连接。

4.22 HTML5 Web 函数手册

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
isReadonly	布尔类型	当前用户只读模式
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
userarea	数值类型	用户区域
userid	数值类型	用户 ID

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”);
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
	});

toggletag(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
queryLog(startTime, endTime, logtype,callback)	logtype: syslog eventlog operatorlog alarmlog 查询日志 返回 json 表格 callback 空白情况下输出会打印到调试器
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) scada.js 中改函数可以打开关闭压缩, 打开关闭调试信息 分析查询性能	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”} LocalDB 限制的 IP 可以设置
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 scadascadav58.js
layerView(title,filename,param,width,height,offsetx,offsety)	浮动弹出画面 scadav58.js
layerAlert(message)	浮动消息 scadav58.js
layerConfirm(message,okcallback,cancelcallback)	浮动询问 scadav58.js

4.23 WTScada Gateway 版本区别

WTScada Gateway 软件商用版本有 4 种授权，分别是标准版本、专业版本和企业版本、跨平台版本。

除跨平台版本外 3 种版本都可以提供软件授权或者 USB key 授权。

下表列出了 4 种版本的功能差别。

序号	功能	测试版	标准版	专业版	企业版
1	驱动				
1.1	系统驱动	☑	☑	☑	☑
1.2	模拟驱动	☑	☑	☑	☑
1.3	S7 驱动 Net	☑	☑	☑	☑
1.4	Modbus 驱动	☑	☑	☑	☑
1.5	三菱驱动	☑	☑	☑	☑
1.6	欧姆龙驱动	☑	☑	☑	☑
1.7	OPCNet（Atuo）驱动	☑	☑	☑	☑

1.8	OPCUA 驱动	☑	☑	☑	☑
1.9	托利多电子秤驱动	☑	☑	☑	☑
1.10	Modbus 驱动（旧）	☑	☑	☑	☑
1.11	Modbus 以太网驱动	☑	☑	☑	☑
1.12	Modbus 串行驱动	☑	☑	☑	☑
1.13	ModbusDTU 驱动	☑	☑	☑	☑
1.14	Modbus 从站驱动	☑	☑	☑	☑
1.15	网络接收驱动	☑	☑	☑	☑
1.16	TCP 驱动	☑	☑	☑	☑
1.17	ABLink 驱动	☑	☑	☑	☑
1.18	Logix5000 驱动	☑	☑	☑	☑
1.19	GE 驱动	☑	☑	☑	☑
1.20	关系数据库驱动	☑	☑	☑	☑
1.21	开放驱动	☑	☑	☑	☑
1.22	JSON 驱动	☑	☑	☑	☑
1.23	PI API 驱动	☑			☑
1.24	PI SDK 驱动	☑			☑
1.25	Ping 驱动	☑	☑	☑	☑
1.26	倍福 AdsNet 驱动	☑	☑	☑	☑
2	**归档**				
2.1	SQLite 日志归档	☑	☑	☑	☑
2.2	SQLServer 日志归档	☑		☑	☑
2.3	MySQL 日志归档	☑		☑	☑
2.4	PostgeSQ 日志归档	☑		☑	☑
2.5	Oracle 日志归档	☑		☑	☑
2.6	SQLite 历史归档	☑	☑	☑	☑
2.7	InfluxDB 历史归档	☑	☑	☑	☑
2.8	PostgeSQL 历史归档	☑		☑	☑
2.9	SQLServer 历史归档	☑		☑	☑
2.10	TimescaleDB 历史归档	☑		☑	☑
2.11	MySQL 历史归档	☑		☑	☑
2.12	Oracle 历史归档	☑		☑	☑
2.13	唐码历史归档	☑			☑
2.14	PI 历史归档	☑			☑
2.15	SQLite 列表归档	☑	☑	☑	☑
2.16	SQLServer 列表归档	☑		☑	☑
2.17	MySQL 列表归档	☑		☑	☑
2.18	PostgeSQL 列表归档	☑		☑	☑
2.19	Oracle 列表归档	☑		☑	☑
2.20	SQLite 行表归档	☑	☑	☑	☑
2.21	SQLServer 行表归档	☑		☑	☑
2.22	MySQL 行表归档	☑		☑	☑
2.23	PostgeSQL 行表归档	☑		☑	☑

2.24	Oracle 行表归档	☑		☑	☑
2.25	SQLServer 关系库同步	☑		☑	☑
2.26	MySQL 关系库同步	☑		☑	☑
2.27	PostgeSQL 关系库同步	☑		☑	☑
2.28	Oracle 关系库同步	☑		☑	☑
3	**扩展**				
3.1	TCP 发送	☑	☑	☑	☑
3.2	UDP 发送	☑	☑	☑	☑
3.3	Redis 发布	☑	☑	☑	☑
3.4	Modbus 服务器	☑		☑	☑
3.5	OPCUA 服务器	☑		☑	☑
3.6	微信发送	☑	☑	☑	☑
3.7	腾讯云短信发送	☑	☑	☑	☑
3.8	MQTT 发送	☑	☑	☑	☑
3.9	WebSocket & HttpServer	☑		☑	☑
3.10	定时调度	☑	☑	☑	☑
3.11	REST 服务（web api）	☑		☑	☑
3.12	阿里云转发	☑	☑	☑	☑
4	**系统服务**				
4.1	报警服务	☑	☑	☑	☑
4.2	对时服务	☑		☑	☑
4.3	UDP 数据服务	☑	☑	☑	☑
4.4	TCP 服务	☑	☑	☑	☑
4.5	IOServer 服务	☑	☑	☑	☑
4.6	Javascript 脚本	☑	☑	☑	☑
4.7	C#脚本	☑	☑	☑	☑

说明：CNC 驱动需要独立授权，按机器数量进行授权。