

采集服务 ◆ 插件帮助

脚本执行器 JAVASCRIPT

北京宏达信诺有限公司

一、脚本采集插件使用详解

1. 创建通道。选择插件，如下图

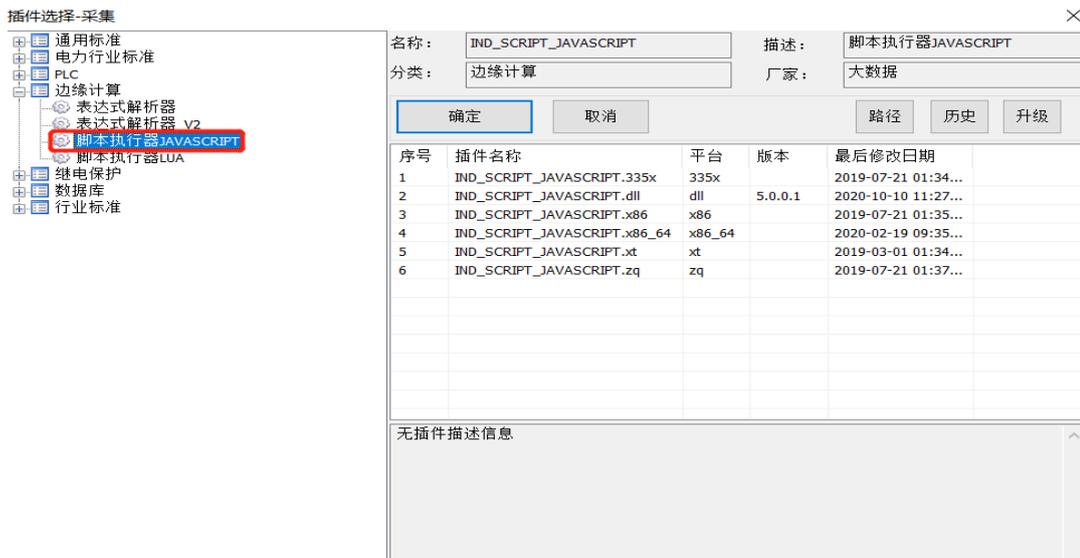


图1

2. 通道参数

是否开启调试终端： 主要用来在WINDOWS版本下运行，测试一下脚本驱动。打印函数输出的信息均能在终端上显示出来，方便观察调试，效果如下



名称(必须为英文) 描述:

厂家: 描述:

路径:

通道类型:

基本参数

脚本执行周期(毫秒)	1000
值设定:相同值不执行持续周期(毫秒)	6000

高级参数

是否开启调试终端?	NO
-----------	----

图2

脚本执行周期： 指调用脚本的周期； **值设定：** 值相同时不执行的持续周期

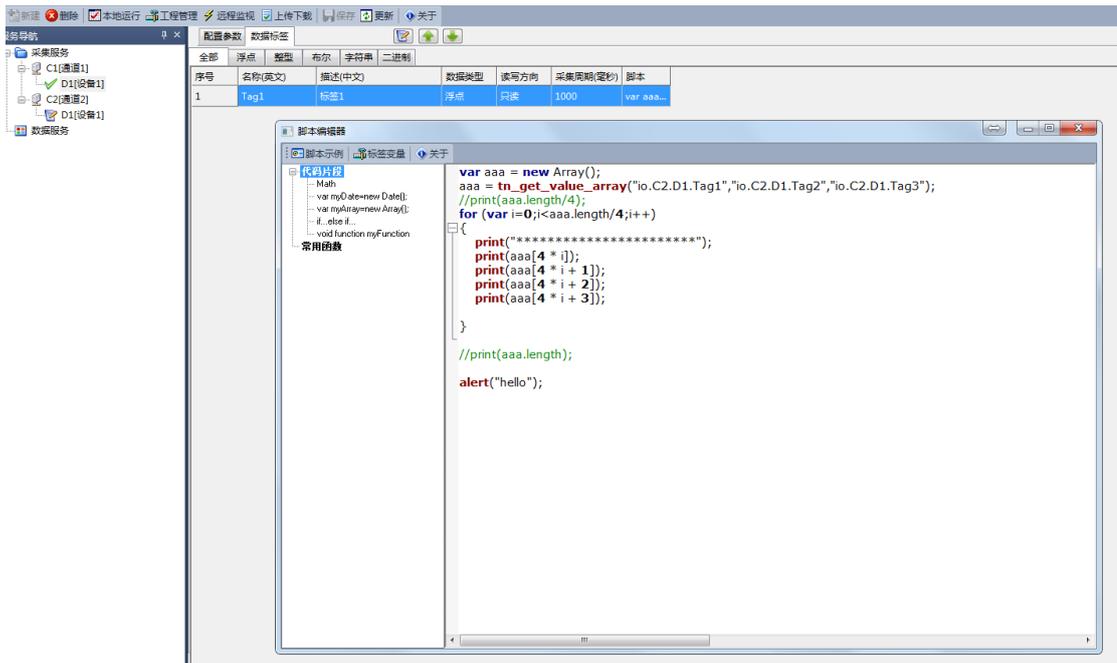


图3

是否开启调试终端：YES状态

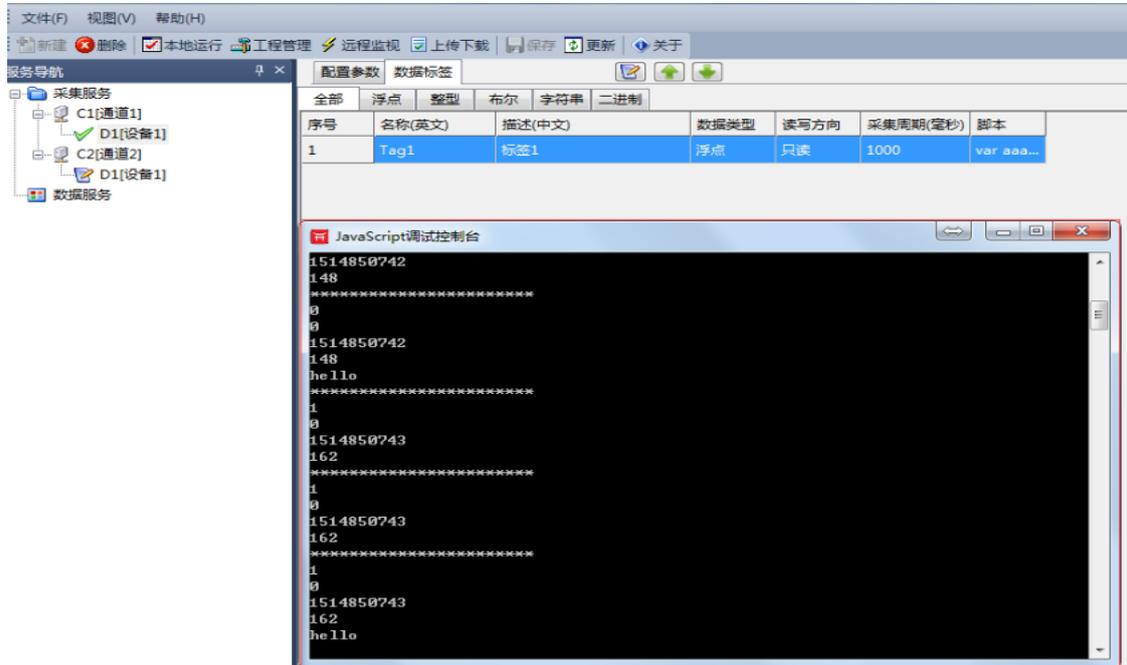


图4

3. 设备参数

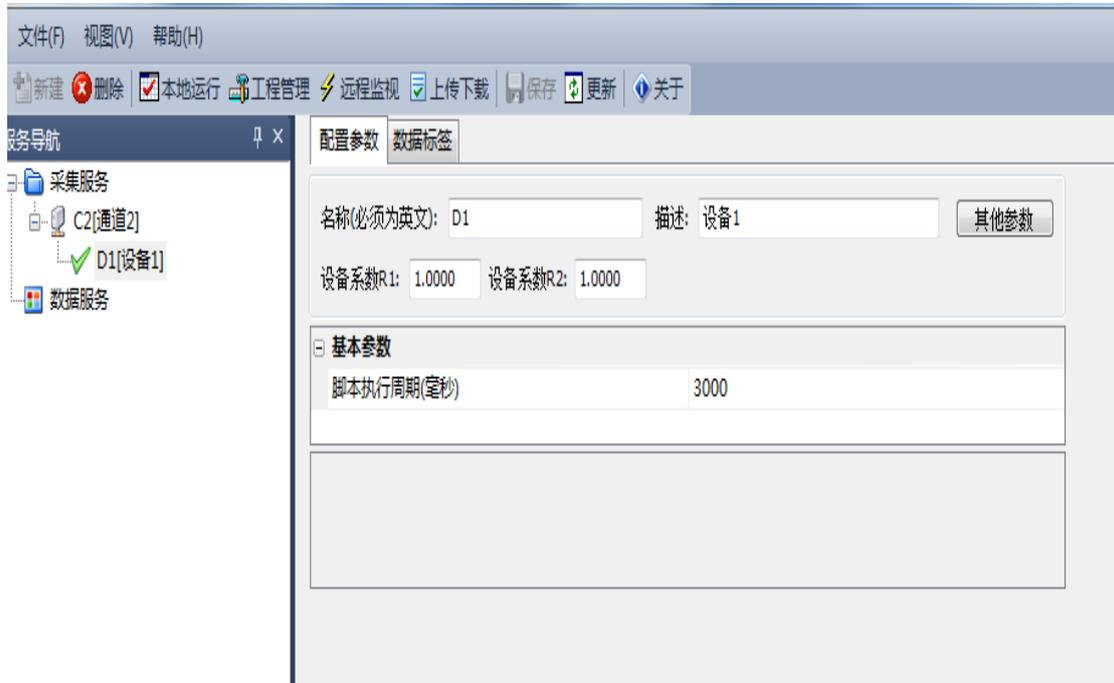
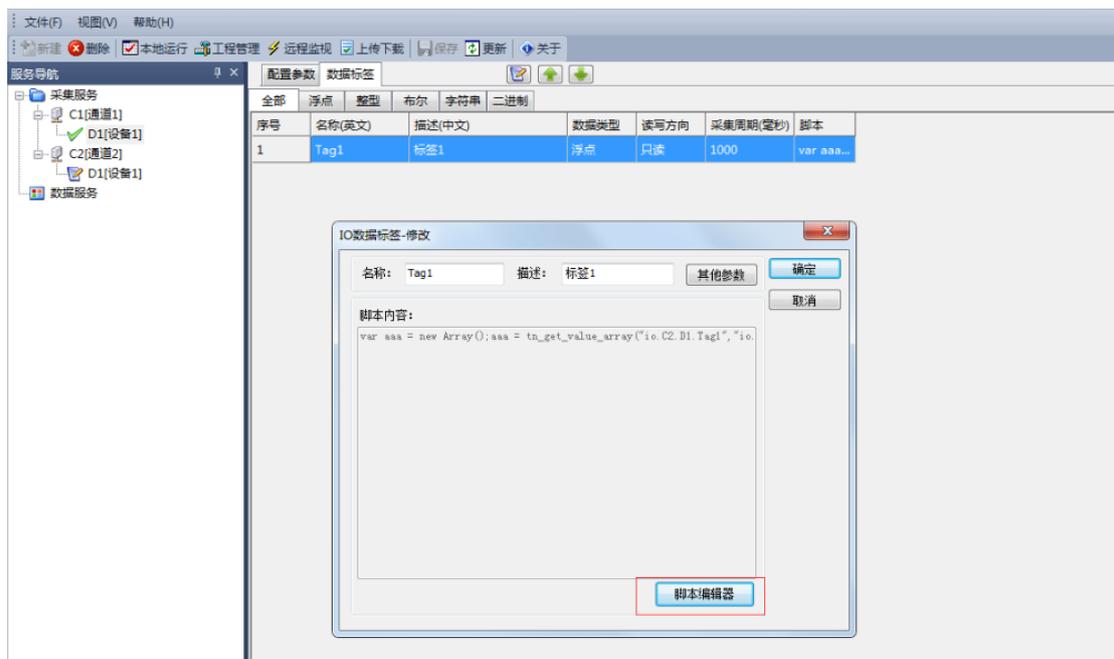


图5

二、 配置数据标签

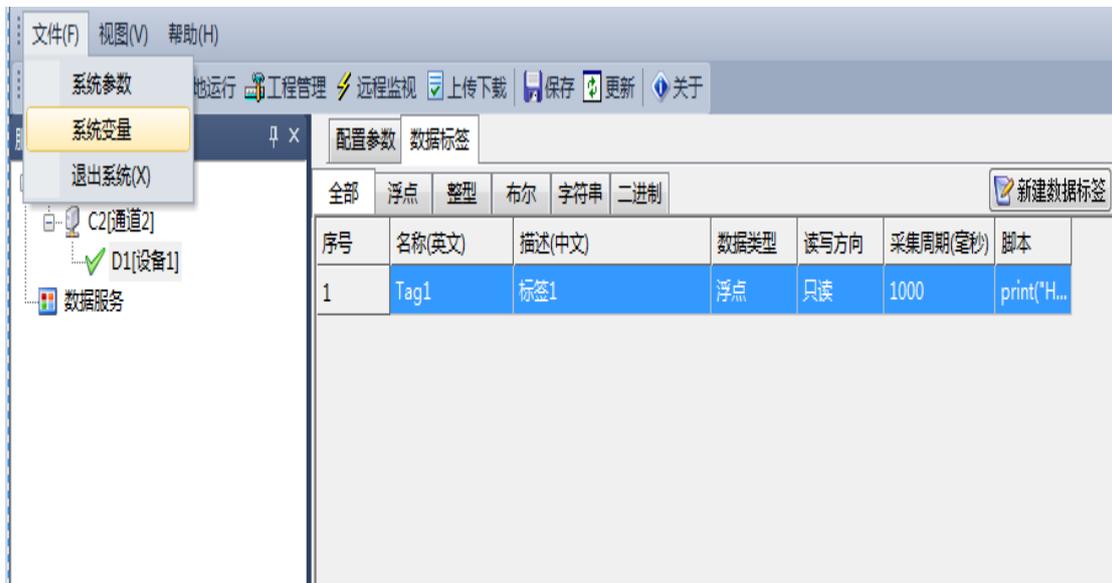


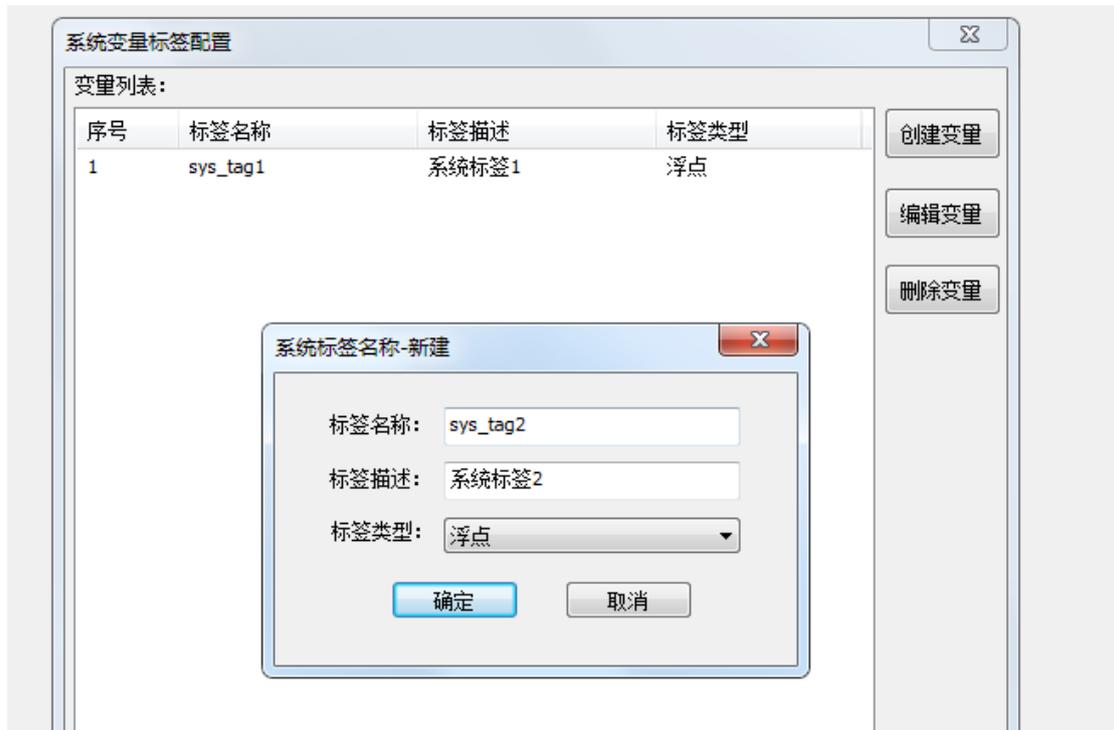
一个标签，执行一个 javascript 逻辑。

数据标签上，点击脚本编辑器，即可对脚本进行编辑，脚本编辑器见上图 3。

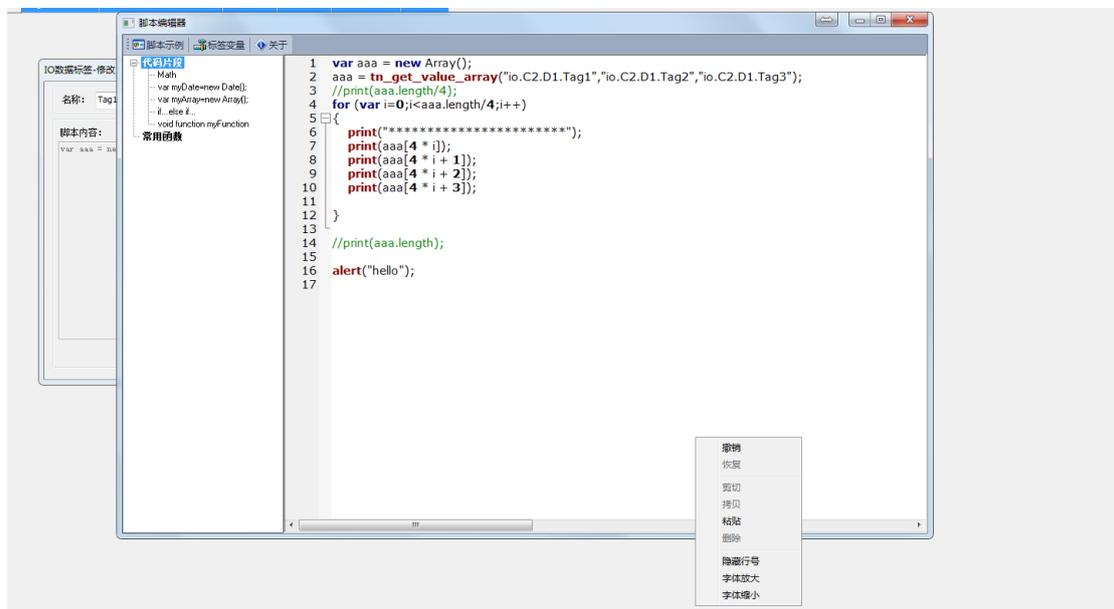
三、 系统变量

系统变量有别于从通道、设备中创建的数据标签，前者是程序员使用，或者脚本使用，临时创建的中间变量；后者是设备中真实存在的一个数据标签。如下图所示，点击文件，点击系统变量，则可进入系统变量配置框。**系统变量为非实时采集的数据标签，在使用脚本时需要创建系统变量作为中间变量。**





四、脚本编辑器使用



脚本编辑器特色

- ◇ 语法高亮，高亮部分包括：关键字，常用函数，数字，注释
- ◇ 自动提示，自动提示部分包括：关键字，常用函数

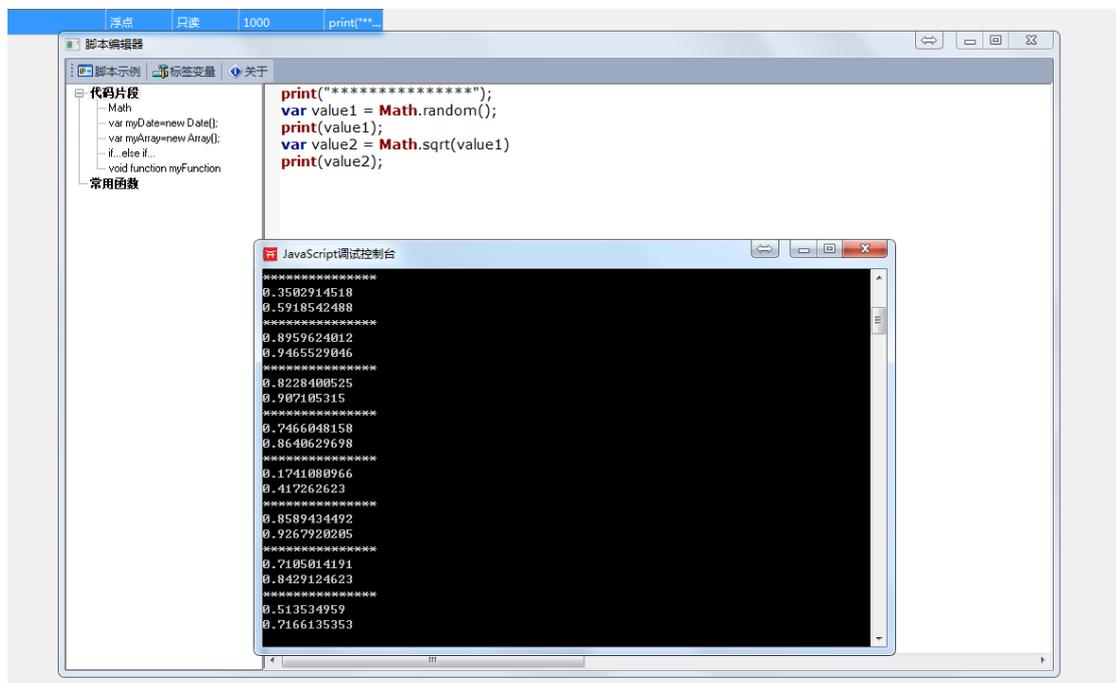
◇ 其他辅助功能，字体放大，缩小，显示行号等

五、常用函数介绍及举例

函数包含系统提供的函数主要有两种，第一 javascript 标准提供的 Math 以及下属成员，第二是我们自定义的，同数据库进行交互的以及常用的功能函数。

1. Math函数，这个为标准函数，常用方法可以参考相关文档，或者我们提供的 javascript参考手册。

下面为一个例子，产生随机数，并计算开方值，并打印出来



2. 自定义的函数

◇ `tn_get_value(csTagLink)`: 获取单个点的值,不含质量戳,只有1个链接项,1个值

例:

```
var value = tn_get_value("io.C1.D1.Tag1");
print(value);
```

◇ `tn_set_value (csTagLink, csValue)`: 仅对数据标签设定值，对IO不进行写操作。



例:

```
tn_set_value("io.C1.D1.Tag1","1234");
```

- ✧ `tn_get_values (csTagLink1, csTagLink2,...)`: 获取多个点的值,不含质量戳, 1个链接项1个值, N个值,使用的时候需要创建数组

```
var array = new Array()  
  
var value = tn_get_values("io.C2.D1.Tag1","io.C2.D1.Tag2","io.C2.D1.Tag3");  
  
print(value[0]);  
  
print(value[1]);  
  
print(value[2]);
```

- ✧ `tn_get_value_array (csTagLink1, csTagLink2,...)`: 获取多个点的值,每个标签顺序为值-质量戳-秒-毫秒, 1个链接项4个值, N * 4个值

例:

```
var array = new Array()  
  
var value = tn_get_value_array("io.C2.D1.Tag1","io.C2.D1.Tag2","io.C2.D1.Tag3");  
  
print("*****");  
  
for(var i=0;i<3;i++)  
{  
  
    print(value[4*i+0]);  
  
    print(value[4*i+1]);  
  
    print(value[4*i+2]);  
  
    print(value[4*i+2]);  
  
}
```

```
*****  
2  
0  
1514852849  
237  
2  
0  
1514852849  
237  
2  
0  
1514852849  
237
```

上面打印的结果共 3 组，每组 4 个值，这四个值分别为：当前值，质量戳，秒，毫秒

◇ `tn_yk(csTagLink, csValue)`: 对某个点进行遥控设定，该点读写属性需要为读写，或者只读。

例：

```
tn_yk("io.C2.D1.Tag1","123");
```

◇ `print(csValue)`: 打印函数，用于调试

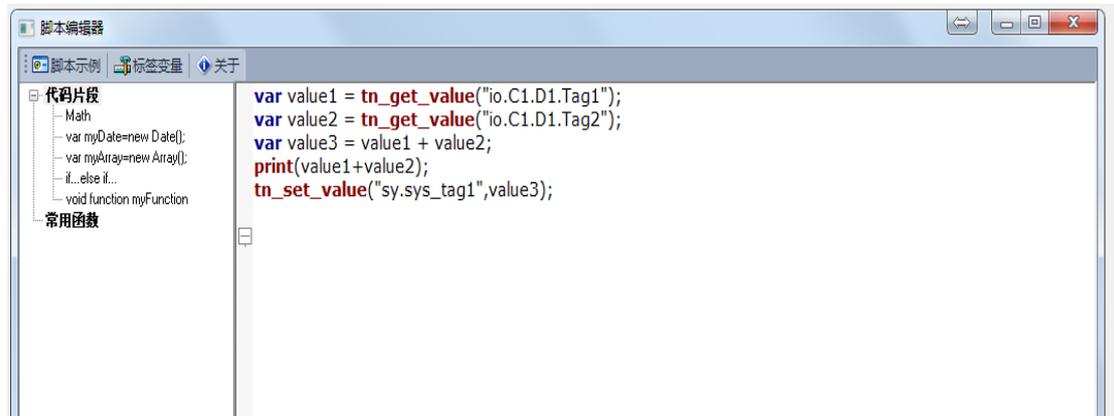
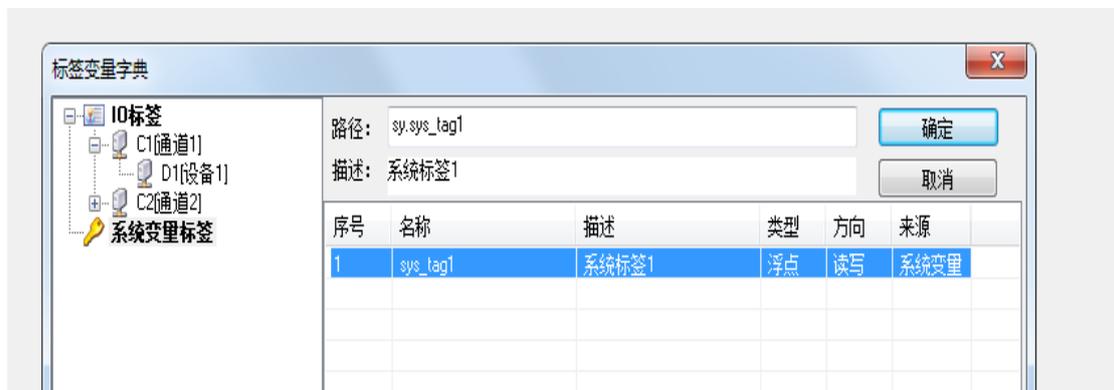
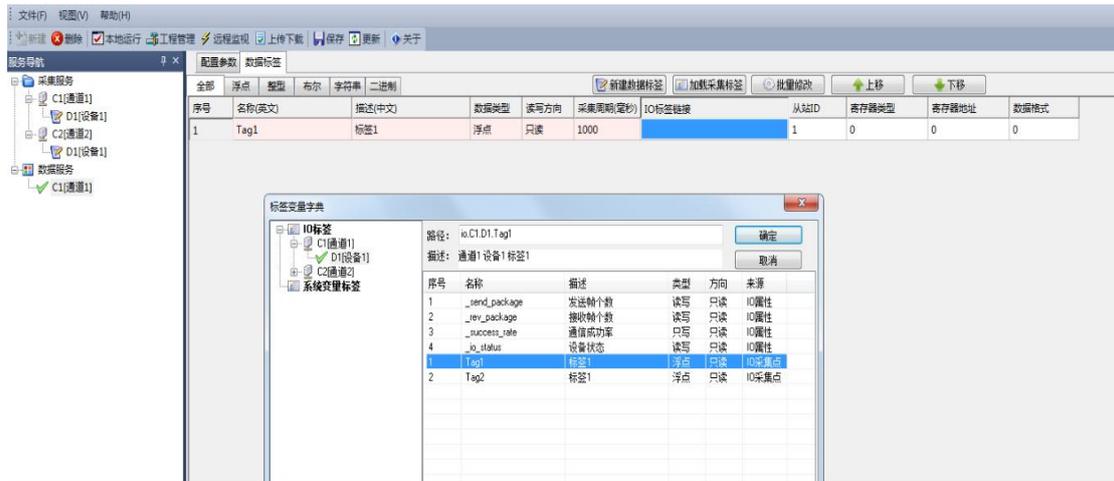
◇ `printf(csValue)`: 打印函数，用于调试

◇ `alert(csValue)`: 打印函数，用于调试

◇ `tn_sleep(nMisecond)`: 休眠函数，单位为毫秒

六、 脚本使用(例子)

采集有两个点 `io.C1.D1.Tag1` 和 `io.C1.D1.Tag2` ，两个值相加送给系统变量 `sy.sys_tag1`



var value1 = tn_get_value("io.C1.D1.Tag1");

var value2 = tn_get_value("io.C1.D1.Tag2");

var value3 = value1 + value2;

print(value1+value2);

tn_set_value("sy.sys_tag1",value3);



192.168.0.89 仿真驱动测试

常用工具 系统日志 关于

系统
采集服务
C1[通道1]
D1[设备1]
C2[通道2]
数据服务
C1[any][通道1]

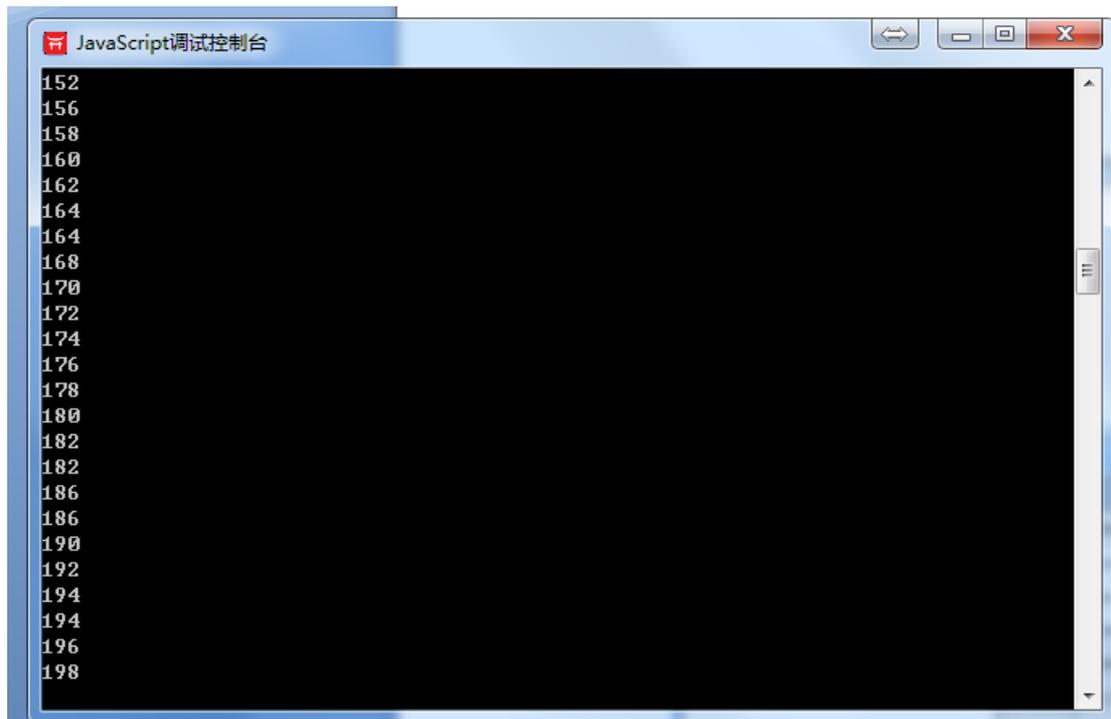
序号	名称(英文)	描述(中文)	当前值	时间
1	_send_package	发送帧个数	64	2017-11-27 13:4
2	_rev_package	接收帧个数	64	2017-11-27 13:4
3	_success_rate	通信成功率	100.000000	2017-11-27 13:4
4	_io_status	设备状态	1	2017-11-27 13:4
5	Tag1	标签1	66.000000	2017-11-27 13:4
6	Tag2	标签1	66.000000	2017-11-27 13:4

192.168.0.89 仿真驱动测试

常用工具 系统日志 关于

系统
采集服务
C1[通道1]
D1[设备1]
C2[通道2]
数据服务
C1[any][通道1]

序号	名称(英文)	描述(中文)	当前值	时间
1	_kernel_version	应用程序内核版本	3.1.0.6	2017-11-27 13:48:00 441
2	_cpu	CPU负荷	0.000000	2017-11-27 13:48:00 441
3	_mem_total	内存总数(K)	0	2017-11-27 13:48:00 441
4	_mem_used	已经使用内存(K)	0	2017-11-27 13:49:05 521
5	_m_run_time	机器运行时间(小时)	0.000000	2017-11-27 13:49:05 521
6	_p_run_time	采集程序运行时间(小时)	0.000000	2017-11-27 13:49:05 521
7	_m_sn	本机编号		2017-11-27 13:48:00 441
8	_time_zone	时区	0	2017-11-27 13:48:00 441
9	_local_time	网关本地时间	2017-11-27 13:49:25 552	2017-11-27 13:49:25 552
10	_product_name	产品名称	HXGE-6204	2017-11-27 13:48:00 441
11	_project_name	工程名称	仿真驱动测试	2017-11-27 13:48:00 441
12	_project_create_time	工程创建时间	2017-11-27 10:37:21	2017-11-27 13:48:00 441
13	_machine_code	授权机器码	18D1CBCEC3F9	2017-11-27 13:48:00 441
14	_license_info	授权信息	已经授权	2017-11-27 13:48:00 441
15	_left_time	运行剩余时间(小时)	99999999.999000	2017-11-27 13:49:05 521
16	sys_tag1	系统标签1	172.000000	2017-11-27 13:49:28 657



(JavaScript 调试控制台需本地运行中调用)