

## 数据服务 ◆ 插件帮助

# CIP 协议 (数据主动上传)

\*该插件的使用，要求人员有一定的 PLC 编程基础，能区分程序中定义的变量类型。

插件文件拷入配置软件根目录的 PluginDs 目录下。

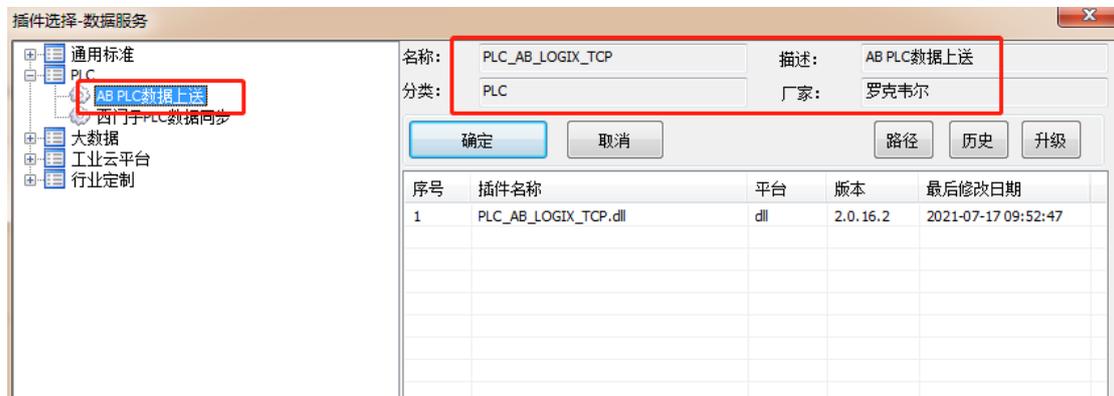
## 插件使用场景

\*采集底层设备数据，将采集到的数据（源变量）写入 AB PLC 定义好的变量点（目标变量）中。

\*本插件不支持反向写入操作，即不能通过 PLC 向底层设备写入数据。

\*数组点可批量写入，为提高写入效率，建议 PLC 中对于各种数据类型都创建对应的数组点。

## 插件的选择



## 通道的配置



## 插件支持写入的点类型及配置

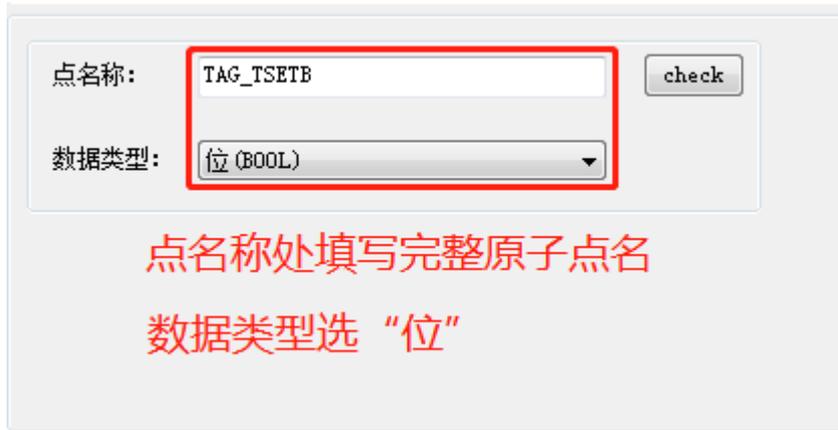
本插件支持数组和原子点的写入，支持的数据类型见下：

**数组：**PLC 中定义的同一种数据类型的变量点集合。我们配置时的点名规范为：AAA[XX]；其中“AAA”为点名称，“[]”中的“XX”为数组中偏移。

**原子点：**PLC 中定义的单点变量。我们配置时的点名规范为：AAA，“AAA”为点名称。

### 1. BOOL 量

- a. 原子点 BOOL 量写入。假设源变量为一个指示开关变化的单点，我们需要将该点写入 PLC 中，PLC 中定义了“TAG\_TESTB”的原子型 BOOL 变量，配置工具中的标签配置窗口如下：

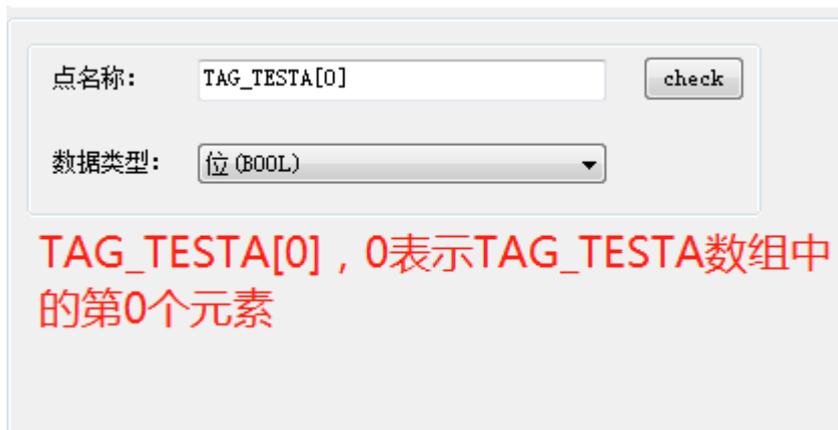


点名称: TAG\_TSETB check

数据类型: 位 (BOOL)

点名称处填写完整原子点名  
数据类型选“位”

- b. 数组 BOOL 量写入。假设底层设备有 **N** 个指示状态的源变量需要写入 PLC，由于原子点写入效率低，我们可以在 PLC 中创建一个尺寸为 32 位的 BOOL 量数组点（注意，BOOL 量数组点只支持 32 位，其他尺寸大小的 BOOL 数值暂不支持），点名“TAG\_TESTA”，配置如下：



点名称: TAG\_TESTA[0] check

数据类型: 位 (BOOL)

TAG\_TESTA[0], 0表示TAG\_TESTA数组中的第0个元素

。。。

点名称: TAG\_TESTA[x] check

数据类型: 位 (BOOL)

**注意，点名中的x表示数组中的第x个元素，为10进制值，取值范围0~31**

如果状态值的个数 N 大于 32：当 N 为 32 的整倍数时，PLC 中至少要定义 N/32 个 BOOL 数组点；当 N 不为 32 的整倍数时，PLC 中至少要定义 N/32 + 1 个 BOOL 数组点。

## 2. SINT

单字节的整形源变量需要写入 PLC 中，则可以在 PLC 中定义一个 SINT 的原子点。如果有多个单字节整型源变量，则可以在 PLC 中定义一个 SINT 的数组。

### a. 单字节源变量 单点配置

点名称: TAG\_TESTF check

数据类型: 8位 (SINT)

### b. 单字节源变量 数组配置

点名称: TAG\_TESTS[x] check

数据类型: 8位 (SINT)

**TAG\_TESTS[x]中的x为TAG\_TESTS数组中的第x个元素，x为10进制的值**

## 3. INT

16 位整形变量，配置类似 SINT 类型数据

点名称: TAG\_TESTI

数据类型: 16位 (INT) ▼

4. DINT  
32 位整形变量，配置类似 SINT 类型数据

点名称: TAG\_TESTD

数据类型: 32位 (DINT) ▼

5. FLOAT  
浮点型源变量

点名称: TAG\_TESTF

数据类型: 32位 (FLDAT) ▼