**TwinCAT3 Json数据处理**

|  |  |
| --- | --- |
|  | 作者：史震宇  职务：技术工程师  邮箱：zy.shi@beckhoff.com.cn  日期：2021年 2月20日 |
| **摘 要：**  TwinCAT3通过Tc3\_JsonXml库实现对Json数据的支持，本文将提供通过该库中的指令实现Json数据交互的方法。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | | 1 | PyJson.py | Python脚本 | | 2 | testjson1.json | Json文件 | | 3 | testjson2.json | Json文件 | | 4 | TwinCAT Json.tnzip | TwinCAT程序 | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |  |  |  | | |
| **免责声明：**  我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，如有改动，恕不事先通知，也欢迎您提出改进建议。 | |
| **参考信息：** | |

目 录

[1. 测试准备 3](#_Toc64646432)

[1.1. 软硬件版本 3](#_Toc64646433)

[1.2. Json基础知识 3](#_Toc64646434)

[2. 指令介绍 3](#_Toc64646435)

[2.1. 手动添加键-值对的方法 4](#_Toc64646436)

[2.2. 直接转化结构体数据的方法 4](#_Toc64646437)

[3. 样例使用方法 5](#_Toc64646438)

[3.1. 将结构体直接转化为Json数据格式 5](#_Toc64646439)

[3.2. 将指定Json字符串转化为结构体 6](#_Toc64646440)

[3.3. 通过键-值对的方式手动建立和添加Json数据 6](#_Toc64646441)

[3.3.1. 将缓存中的数据输出为Json字符串 7](#_Toc64646442)

[3.4. JSON文件读写 8](#_Toc64646443)

[3.4.1. 写入Json文件 8](#_Toc64646444)

[3.4.2. 读取Json文件 8](#_Toc64646445)

[3.5. 使用Python解析TwinCAT生成的Json文件 10](#_Toc64646446)

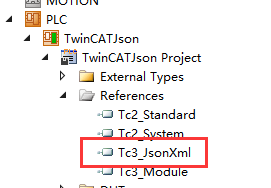
# 测试准备

## 软硬件版本

测试平台：普通PC；

TwinCAT3版本：4024.11；

Tc3\_JsonXml库版本：3.3.14.0。



该库参考链接：

https://infosys.beckhoff.com/content/1033/tcplclib\_tc3\_jsonxml/index.html?id=1462033306839781294

## Json基础知识

JSON(JavaScript Object Notation) 指的是 JavaScript 对象表示法，是一种轻量级的数据交换格式。它可以很轻松地被进行阅读和编写，同时也很方便机器进行解析和生成。JSON采用完全独立于程序语言的文本格式，也使用了类C语言的习惯（包括C, C++, C#, Java, JavaScript, Perl, Python等）。这些特性使JSON成为理想的数据交换语言。

JSON基于两种结构：

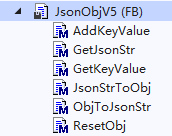
“键-值”对的集合（A collection of name/value pairs）。不同的编程语言中，它被理解为对象（object），纪录（record），结构（struct），字典（dictionary），哈希表（hash table），有键列表（keyed list），或者关联数组 （associative array）。

值的有序列表（An ordered list of values）。在大部分语言中，它被实现为数组（array），矢量（vector），列表（list），序列（sequence）。

这些都是常见的数据结构。目前，绝大部分编程语言都以某种形式支持它们。这使得在各种编程语言之间交换同样格式的数据成为可能。

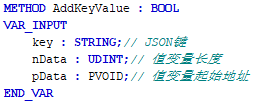
# 指令介绍

本文封装了JsonObjV5这个功能块，通过这个功能块中的method，可以通过手动添加键值对生成Json字符串，也可以实现结构体数据与Json字符串之间的转换。



## 手动添加键-值对的方法

1.AddKeyValue



该方法以键-值对的形式向缓存中添加Json数据，输入变量为键（字符串），值变量的起始地址以及长度，使用方法如下：



2.ResetObj



该方法可以重置当前的缓存，即清空已添加的数据，使用方法如下：



3.GetJsonStr

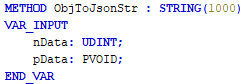


该方法可以将当前缓存中的Json数据进行输出，得到Json字符串，使用方法如下：



## 直接转化结构体数据的方法

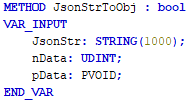
1.ObjToJsonStr



该方法可以直接将数据转化为Json字符串并输出，输入变量为需要转化的变量的长度以及起始地址，返回Json字符串，使用方法如下：



2.JsonStrToObj



该方法可以将Json字符串转化为特定结构体，输入变量为需要转化的Json字符串，用于接收返回值的变量长度以及起始地址。

# 样例程序的使用

本文提供的样例程序可以实现如下功能：

1.将指定结构体对象转化为Json字符串，或者将指定Json字符串转化为结构体对象；

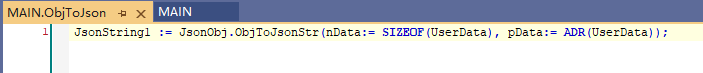
2.通过键-值对的方式手动建立和添加Json数据，添加完成后将缓存中的Json对象输出成Json字符串；

3.读写Json文件；

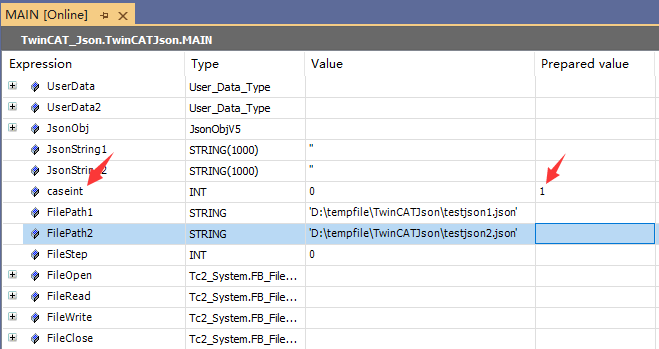
4.Python（也可以是其他语言）读写Json文件，或者使用ADS通讯的方式直接获取Twincat3生成的Json字符串，即可实现Twincat3与其他平台的数据交互。

首先打开样例程序，激活配置并login启动程序。

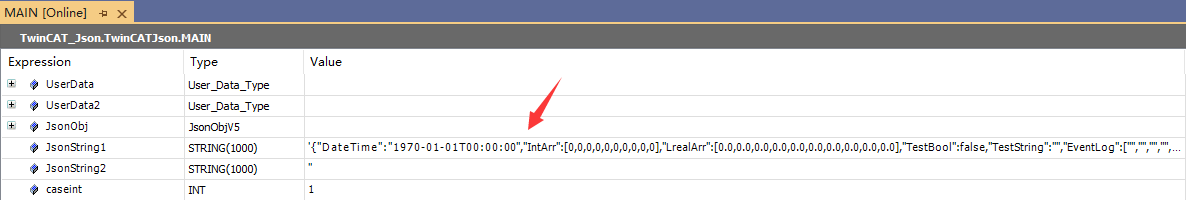
## 将结构体直接转化为Json数据格式



调用JsonObjV5的ObjToJsonStr方法即可直接将指定的数据转化为Json字符串，对于数据类型没有限制，支持用户自定义的结构体数据。

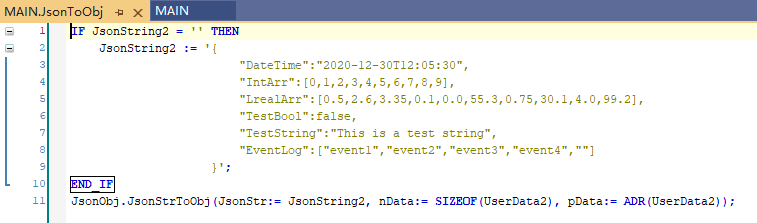


将caseinit置为1即可调用该方法，调用后效果如下：



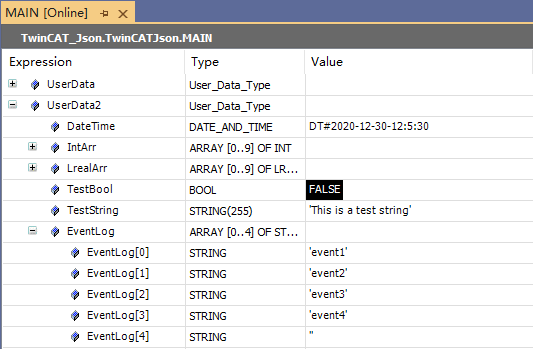
已经将结构体转化为Json字符串并写入指定变量。

## 将指定Json字符串转化为结构体



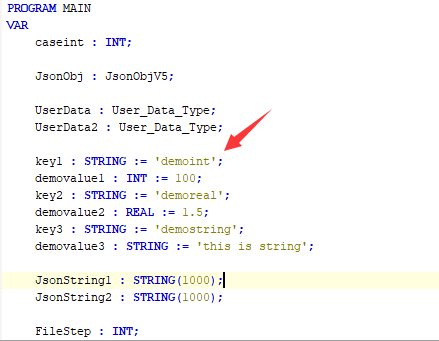
调用JsonObjV5的JsonStrToObj方法即可直接将指定Json数据转换为结构体。

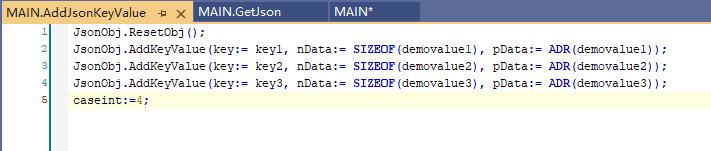
将caseinit置为2即可调用，效果如下：



## 通过键-值对的方式手动建立和添加Json数据

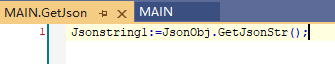
JsonObjV5的AddKeyValue方法支持以键-值对的方式手动添加Json数据。





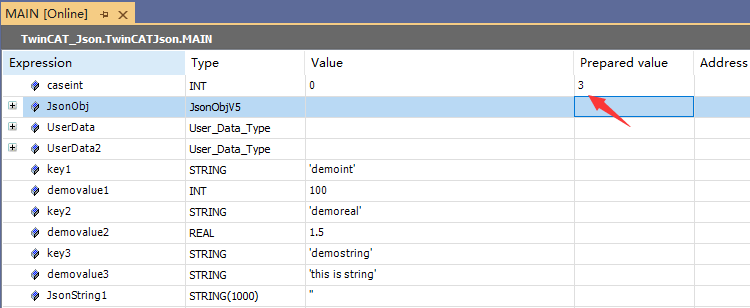
该方法也不限制数据类型，使用该方法之后，会将键-值对加入到缓存中，需要使用GetJsonStr的方法对缓存进行输出。

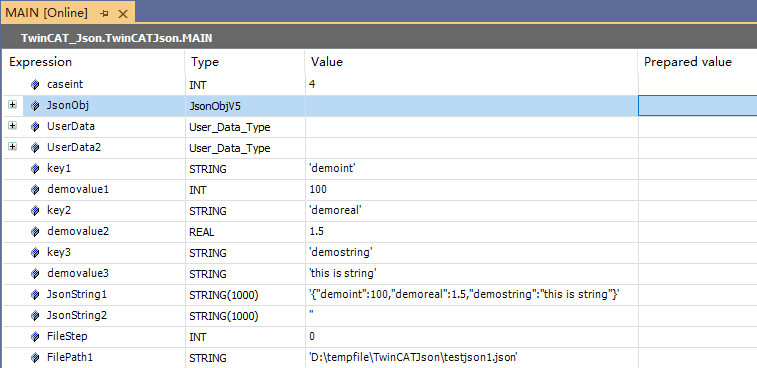
### 将缓存中的数据输出为Json字符串



JsonObjV5的GetJsonStr方法可以输出缓存中的Json数据。

将caseint置为3，运行效果如图：



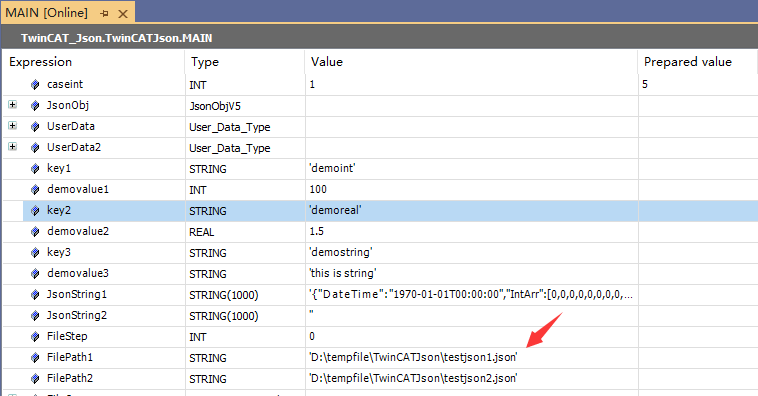


## JSON文件读写

Json文件后缀为.json可以使用记事本打开以及编辑，TwinCAT3可以通过FB\_FileOpen，FB\_FileWrite，FB\_FileRead，FB\_FileClose对Json文件进行操作。

### 写入Json文件

使用文件写入之前，先将caseint置为1，将结构体转化为Json字符串，然后将caseint置为5，进行文件写入，文件写入的默认路径是FilePath1，测试的时候也可以替换为其他文件路径：

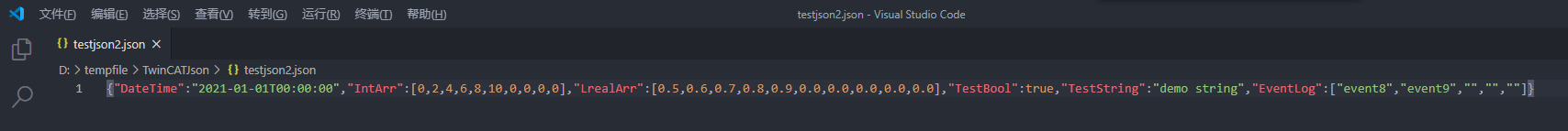


到对应路径下，使用文本编辑器如notepad++，notepads，VScode等工具打开.json文件，可以直接看到写入的内容，本文以VScode来操作：

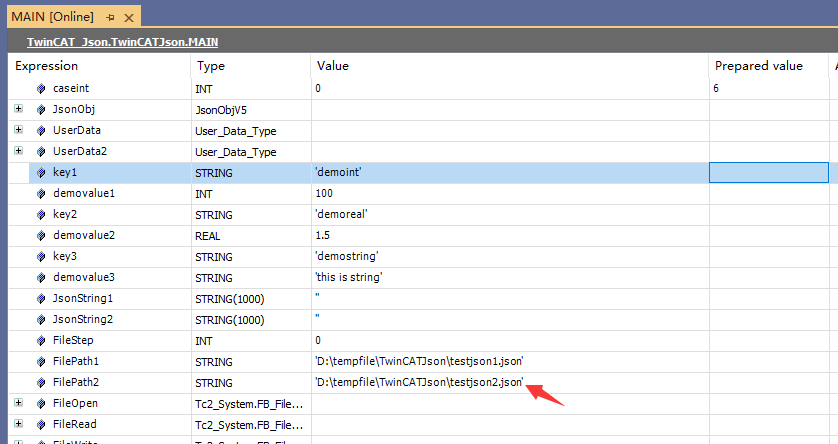


### 读取Json文件

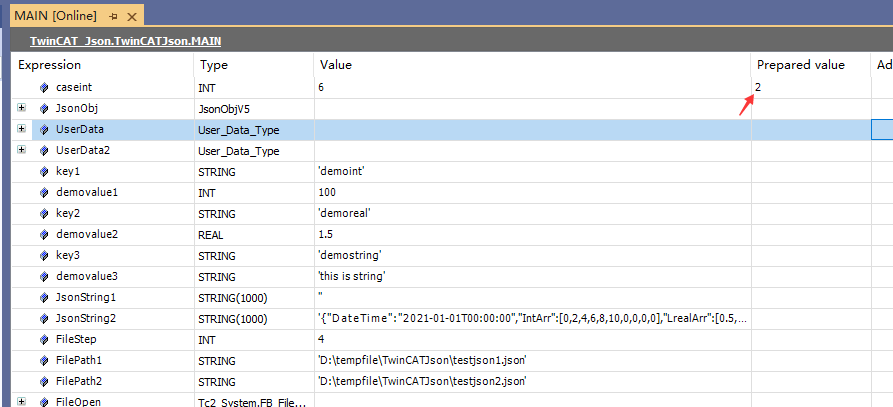
通过FB\_FileRead也可以读取Json文件，使用VScode打开已经准备好的testjson2文件，可以看到里面的数据：



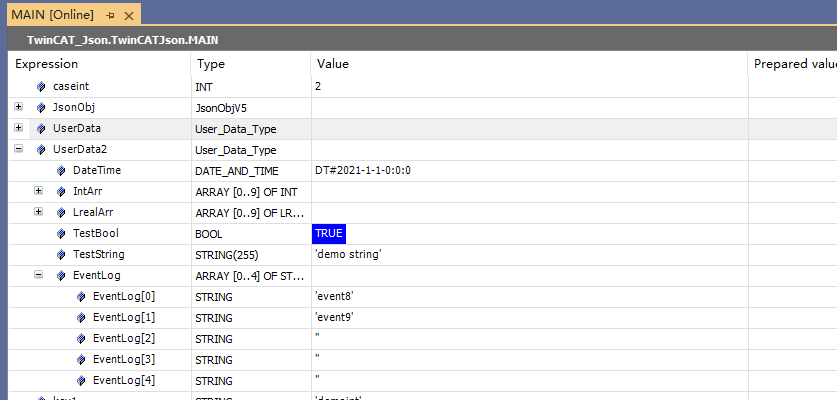
将该文件的路径填入FilePath2：



读取成功之后字符会被存入JsonString2，再调用JsonToObj这个action即可向UserData2这个结构体写入数据：



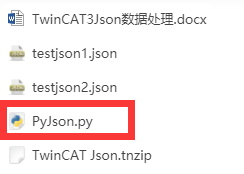
运行效果如图：



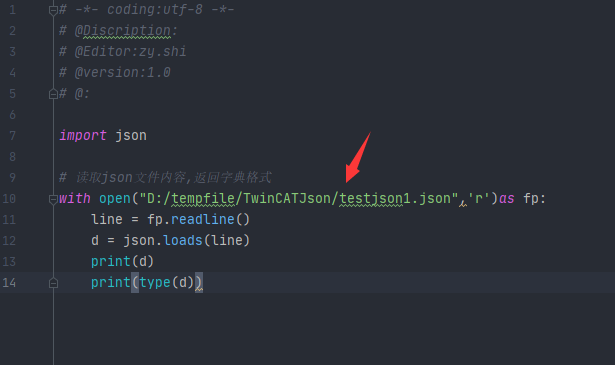
## 使用Python解析TwinCAT生成的Json文件

支持JSON的编程语言非常多，如C++中通过JsonCpp等其他几种Json库来解析Json数据；Javascript可以直接使用JSON.prase()以及JSON.stringify()直接转换；PHP也配备有json\_encode，json\_decode等相应的解析函数，本文使用Python来读取TwinCAT生成的Json文件。

Python通过Json这个包来实现对Json数据的支持，并且可以将Json数据解析为字典类型，操作方法如下：

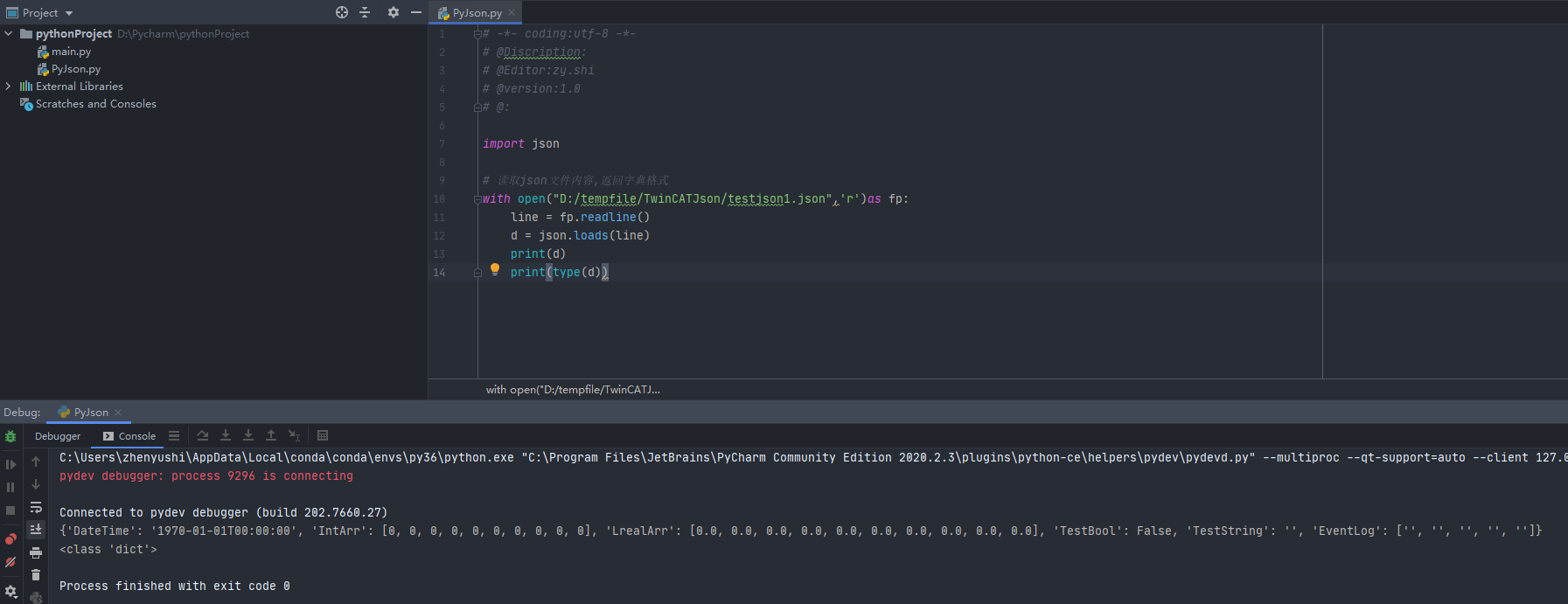


打开附件中的PyJson脚本，脚本中已经导入了Json包，将TwinCAT生成的Json文件路径填入：



脚本中使用readline()方法进行文件读取，再使用json.loads方法，将读取到的字符串载入为字典，最后通过终端打印。

使用效果如图：



**上海（ 中国区总部）**

中国上海市静安区汶水路 299 弄 9号（市北智汇园）

电话: 021-66312666

**北京分公司**

北京市西城区新街口北大街 3 号新街高和大厦 407 室

电话: 010-82200036 邮箱: beijing@beckhoff.com.cn

**广州分公司**

广州市天河区珠江新城珠江东路16号高德置地G2603 室

电话: 020-38010300/1/2 邮箱: guangzhou@beckhoff.com.cn

**成都分公司**

成都市锦江区东御街18号 百扬大厦2305 房

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 | 倍福官方网站：  https://www.beckhoff.com.cn  在线帮助系统：  https://infosys.beckhoff.com/index\_en.htm |
| 倍福虚拟学院：  https://tr.beckhoff.com.cn/ |
| 招贤纳士：job@beckhoff.com.cn  技术支持：support@beckhoff.com.cn  产品维修：service@beckhoff.com.cn  方案咨询：sales@beckhoff.com.cn |
|  |