**TwinCAT HMI Framework Control创建和使用**

|  |  |  |
| --- | --- | --- |
|  | 作者：马欣欣  职务：数据科学与工程部 软件工程师  公司：BECKHOFF中国  邮箱：xx.ma@beckhoff.com.cn  日期：2023-11-23 | |
| **摘 要：**  结合附件代码，了解创建、使用TwinCAT HMI Framework Control的整体流程。   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | | 1 | FrameworkPrjEmpty1 | 示例程序 | |  |  |  | |  |  |  | |  |  |  | | | **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. 软硬件版本 3](#_Toc153380251)

[1.1. 控制器硬件 3](#_Toc153380252)

[1.2. 控制软件 3](#_Toc153380253)

[2. 操作步骤 3](#_Toc153380254)

[2.1. Framework Project的创建和结构 3](#_Toc153380255)

[2.1.1. Framework Project创建 3](#_Toc153380256)

[2.1.2. Framework Project结构 4](#_Toc153380257)

[2.2. 具体步骤 8](#_Toc153380258)

[2.2.1. 在Template.html加入元素 8](#_Toc153380259)

[2.2.2. 在Description.json定义参数 8](#_Toc153380260)

[2.2.3. 在Control.js编写逻辑 8](#_Toc153380261)

[2.2.4. 添加CSS 9](#_Toc153380262)

[2.3. 导入和发布Framework Control 9](#_Toc153380263)

[2.3.1. 添加HMI Project 9](#_Toc153380264)

[2.3.2. 发布Framework Control 11](#_Toc153380265)

[3. 运行效果 13](#_Toc153380266)

# 软硬件版本

## 控制器硬件

本地电脑调试

## 控制软件

笔记本是基于TwinCAT 3.1 Build 4024.50版本

TE2000是1.12.760.42版本

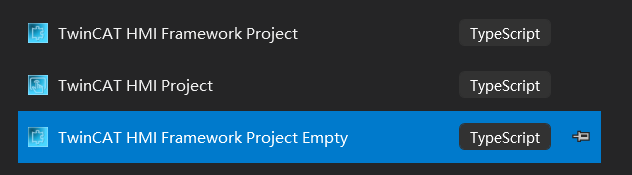
TF2000是1.12.760.42版本

# 操作步骤

## Framework Project的创建和结构

### Framework Project创建

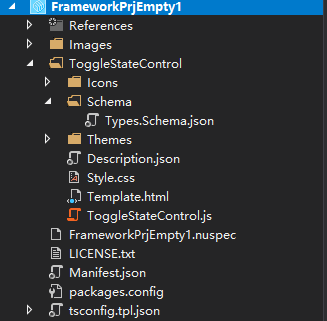
选择Framework Project可直接创建一个包含 Framework Control(ts)的工程，如果想使用js可以先选择Project Empty再手动添加Framework Control (js)。



电脑萤幕的截图

描述已自动生成

### Framework Project结构



首先需要了解项目结构，其中几个重要的文件如下：

1. **Template.html**

用来编写HMTL代码，可以使用标准的HTML元素或TwinCAT HMI控件

1. **Style.css**

给html文件中的元素添加样式

1. **Control.js**

每个framework control会有一个同名的js、ts文件。其中已经有部分代码，例如生命周期init、attach、detach,需要在这个文件中完成控件的逻辑功能

1. **Description.json**

这是一个重要的配置文件，创建控件一般需要设置的Attributes、Functions、Events、Access、Data types都需要在这里定义。打开该文件可以看到许多属性，主要会修改以下几个属性：

* 1. dependencyFile 这里包含了所有需要的依赖文件，默认包含了一个control.js和一个css文件。
  2. attributes 设置属性，显示在HMI属性配置页面。字段如下，**红色字段必填**：

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 属性名称的标准格式以：data-命名 |
| propertyName | 内部变量省去前缀 |
| propertySetterName | Setter函数：设置属性值 |
| propertyGetterName | Getter函数：获取属性值 |
| displayName | 在配置窗口，该属性的显示名称 |
| visible | 属性是否显示在配置窗口，默认true |
| themeable | 默认Standard |
| displayPriority | 属性在窗口展示的优先级 |
| type | 数据类型，也可以自行定义新的datatype |
| category | 代表属性属于哪个分类，可以使用标准的分类名称，如果是定义新的分类需要在attributeCategories定义 |
| description | 属性描述，在配置窗口中鼠标移上去进行提示 |
| readOnly | 是否只读 |
| bindable | 是否可绑定变量 |
| heritable | 是否可继承，默认true |
| defaultValue | 默认值，如果为null则在窗口中不显示 |
| defaultValueInternal | 内部默认值，setter设置的默认值 |

* 1. attributesCategories 设置新属性分类，标准的属性分类例如Value 、Common等等。例在定义时将分类设置为Common，该属性则展示在Common分类下。

如果想使用新的属性分类，该分类需要先在此定义，字段如下，**红色字段必填**：

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 分类名称 |
| displayPriority | eg.Common=100,Layout=200,Border=300  优先级大于500，则低于这些标准分类，展示在属性窗口最下面 |
| defaultCollapsed | 是否折叠 |
| description | 描述 |

* 1. functions 自定义函数，可以选择是否带参数。字段如下，**红色字段必填**：

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 函数名称 |
| displayName | 显示名称 |
| visible | 是否可见 |
| description | 描述 |
| category | 所属分类 |
| params | 参数 |
| type | 返回值类型 |
| heritable | 是否可继承 |
| searchTerms | 函数的搜索词 |

params

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 参数名称 |
| displayName | 显示名称 |
| description | 描述 |
| type | 数据类型 |
| visible | 是否可见 |
| bindable | 是否允许绑定变量 |

* 1. events 设置事件，在HMI show events窗口展示。字段如下，**红色字段必填**：

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 事件名称 |
| displayName | 显示名称 |
| visible | 是否可见 |
| displayPriority | 在配置窗口中展示的优先级次序 |
| Category | 事件分类 |
| description | 描述 |
| heritable | 是否可继承, 默认值true |
| allowsPreventDefault | 是否允许阻止默认行为，默认值false |

* 1. datatype

可以使用现有的DataType，也可以自定义DataType。自定义数据类型可分为Object、Arrays、Enum。步骤如下：

* 新建datatype文件

文本

描述已自动生成

* 定义CustomDataType.json内容如下

{

"$schema": "http://json-schema.org/schema#",

"definitions": {

"**MachineStates**": {

"type": "string",

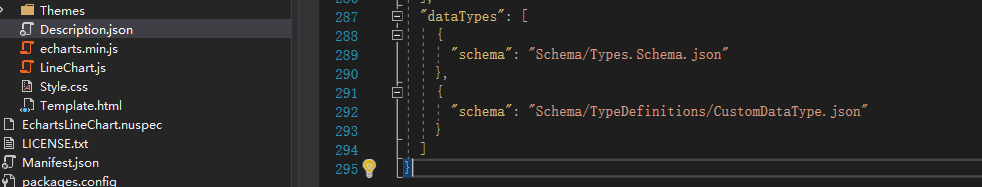
"enum": [ "Start", "Stop", "Reset" ]

}

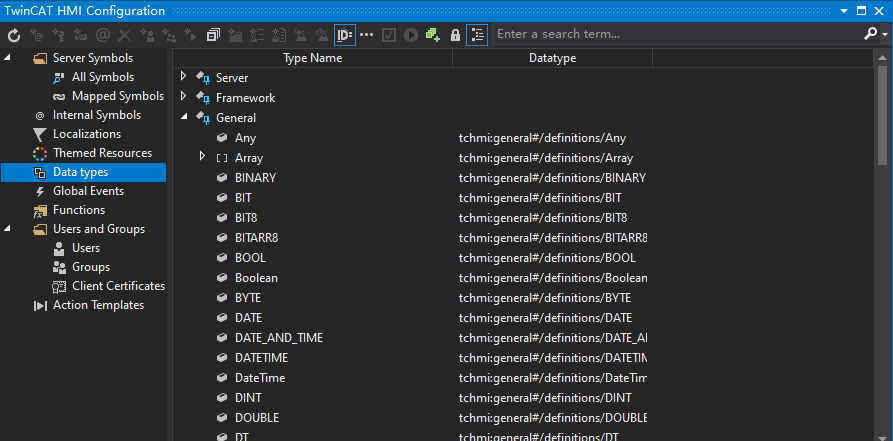
}

}

* 新建datatype文件后，该文件也需要在Description.json中引用



* HMI Configuration窗口可以查看已有的Data types，参考下图



* 新建CustomDataType有可复用的优点。当dataType比较复杂时，也可以将其中的子项拆分出来。例如定义一个object，其中的每一项为MachineStates类型，使用$ref引用

"TEST": {

"type": "**object**",

"propertiesMeta": [

{

"name": "State",

"category": "State",

"displayName": "State",

"displayPriority": 10,

"description": "",

"defaultValue": "Start",

"defaultValueInternal": "Start"

}

],

"properties": {

"State": {

"$ref": "tchmi:framework#/definitions/MachineStates"

}

},

"required": [ "State" ]

}

* 1. access 权限字段如下：

|  |  |
| --- | --- |
| **字段** | **描述** |
| name | 权限名称 |
| displayName | 显示名称 |
| description | 描述 |
| visible | 是否可见 |
| defaultValueInternal | 默认值 |

关于以上属性，详细参考information system: TE2000-Framework-Extensions

[Beckhoff Information System - English](https://infosys.beckhoff.com/english.php?content=../content/1033/te2000_tc3_hmi_engineering/8972372619.html&id=8085440533816053318)

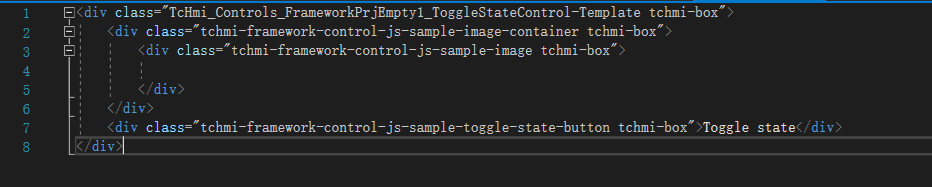
## 具体步骤

完成以上新建工作，后续的操作步骤如下。

### 在Template.html加入元素

在html中添加image、button元素，实现点击button切换不同状态。

Class作为类名，用于CSS样式的显示和切换。



### 在Description.json定义参数

1. 设置三个属性，在attributes中进行添加。分别为：data-tchmi-type（只读）、data-tchmi-machine-state、data-tchmi-show-toggle-state-button
2. 设置两个函数, toggleState是不带参数的，toggleToState是带参数的
3. 设置一个事件onStateChanged
4. 设置一个权限 allowStateChange

### 在Control.js编写逻辑

该文件中已有部分代码，包括constructor和各个生命周期。其中constructor中用于定义内部变量。生命周期函数如下：

|  |  |
| --- | --- |
| \_\_previnit() | 在此阶段，可以判断DOM是否已存在 |
| \_\_init() | 在此阶段，已完成Setter calls |
| \_\_attach() | 在此阶段，DOM已加载完成 |
| \_\_detach() | 在此阶段，控件已从DOM移除 |
| destroy() | 在此阶段，控件已销毁 |

* **其他函数介绍：**

每个属性都必须有自己的setter(除了只读属性外)、getter，在setter执行后属性值被修改，因此还需要定义一个用于处理这个变化的方法。

如：setMachineState是【data-tchmi-machine-state】属性的setter，getMachineState是其getter。

在属性值被修改后，\_\_processMachineState方法根据\_\_machineState给\_\_elementImage赋予不同的class。在css中进行样式定义，使用js调用RemoveClass、addClass方法进行class切换。

* **用到的TwinCAT HMI API：**

1. TcHmi.Access.checkAccess(control,requestAccessright)判断用户是否有该control的权限。本程序中定义的权限是allowStateChange，分为disabled、activate两种。同样在css中定义这两种类的样式。
2. TcHmi.EventProvider.raise 唤起某事件
3. TcHmi.ValueConverter 数据类型转换

参考Information System -> TE2000->Framework->API Reference->TcHmi

[Beckhoff Information System - English](https://infosys.beckhoff.com/english.php?content=../content/1033/te2000_tc3_hmi_engineering/3728925963.html&id=7832936402195120991)

### 添加CSS

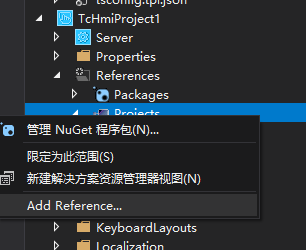
本程序在Themes->Style.css进行了样式定义。例如：

* *tchmi-framework-control-js-sample-state-start*
* *tchmi-framework-control-js-sample-state-stop*
* *tchmi-framework-control-js-sample-state-reset*
* .tchmi-framework-control-js-sample-toggle-state-button.hide
* .tchmi-framework-control-js-sample-toggle-state-button.visible

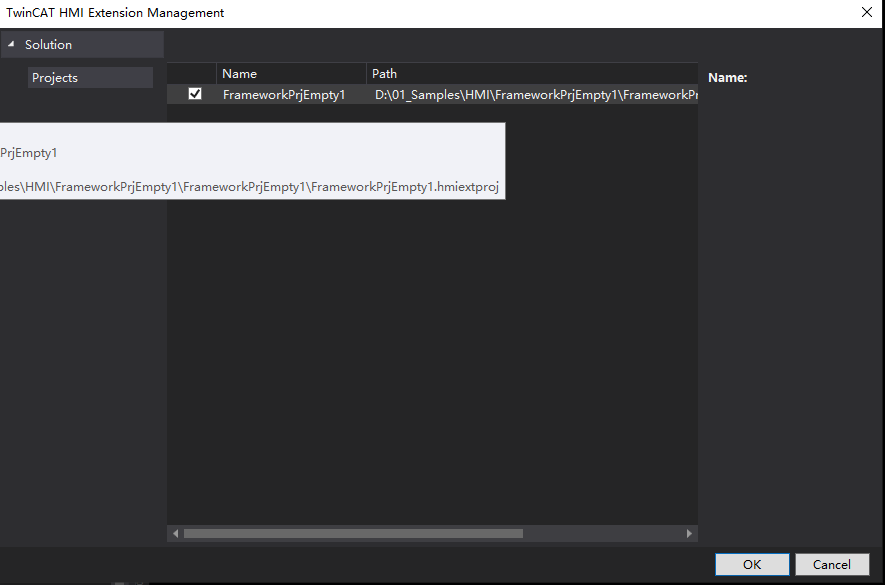
## 导入和发布Framework Control

### 添加HMI Project

1. 在该解决方案下添加一个HMI Project。
2. 点击References->Projects 右击选择【Add Reference】。



1. 将前面的选择框勾上。



1. 现在可以在HMI Project中使用该控件，进入Desktop.view，打开toolbox可以看到该控件。

图形用户界面, 文本, 应用程序

描述已自动生成

1. 将该控件移动到画布中，在属性窗口可以看到该控件上的两个属性【MachineState、ShowToggleStateButton】,设置完成后预览项目,在控件上点击按钮切换状态。
2. 在设计该控件时创建了两个函数，所以调用function也可以实现状态的切换。如何在窗口中找到function见下图：

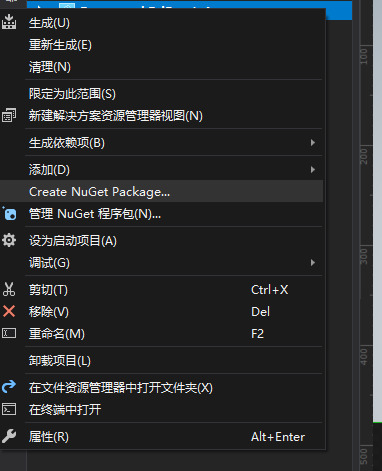
图形用户界面, 文本, 应用程序

描述已自动生成

### 发布Framework Control

通过【Add Reference】的方法可用于测试，测试完成后可以进行发布。

1. 右击Framework Project，选择【Create NuGet Package】。



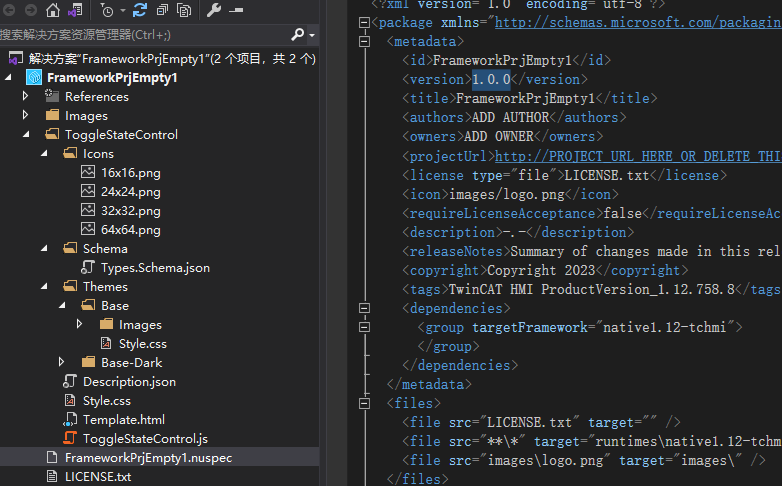
图形用户界面, 应用程序

描述已自动生成

文本

描述已自动生成

1. 点击Explore可以看到生成好的Package存在【C:\TwinCAT\Functions\TE2000-HMI-Engineering\References】路径下。因此，如果需要将该控件给其他人使用，对方拿到Package也要放在这个路径。
2. 如果需要修改版本号，可以在.nupesc文件->version进行修改，参考下图：

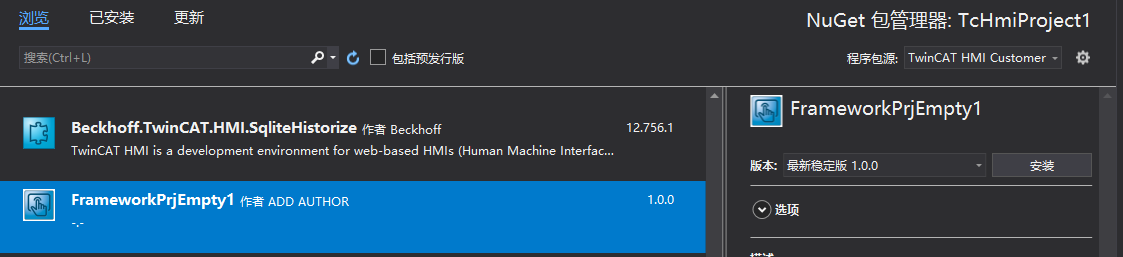


1. 下面可以创建一个新的HMI Project，测试通过NuGet安装的方式使用控件。
2. 创建完成后，右击References文件夹，选择管理NuGet程序包。

文本

描述已自动生成

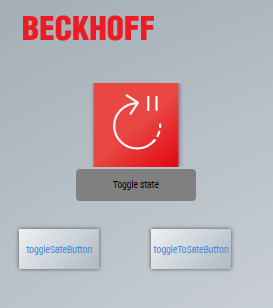
1. 注意右上角程序包源切换到【TwinCAT HMI Customer】,然后在浏览中可看到Framework Project，点击安装。



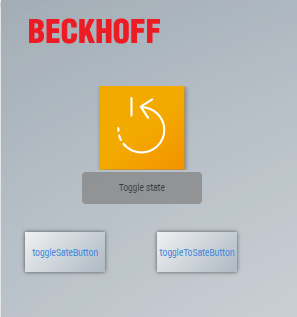
1. 安装完成后，同样在toolbox中可以使用该控件，使用方式与之前一致。

# 运行效果

1. 预览画面



1. 点击控件中的Toggle State按钮切换状态，控件下方的两个按钮也可进行状态切换

图形用户界面, 应用程序, 网站, Teams

描述已自动生成

1. 如果当前用户没有allowStateChange权限，效果如下。使用该功能需要提前创建用户分组及用户。

电脑萤幕的截图

描述已自动生成



图形用户界面, 应用程序

描述已自动生成

**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32号利通广场1303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 室

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 |
|  |