**基于C#实现TwinCAT3 Ads Notifications方式通讯**

|  |  |
| --- | --- |
|  | 作者：孙烨职务：华东区 技术工程师公司：BECKHOFF中国邮箱：ye.sun@beckhoff.com.cn日期：2024-12-12 |
| **摘 要：**在基于C#进行开发上位机时，可以直接通过Nuget中添加官方Beckhoff的Packages来做Ads通讯，这点非常棒，通常读写TwinCAT PLC变量我们会新建一个task进行周期性访问，那么读取一个重要变量变化肯定有延迟，那么下面就是通过Ads Notifications方式，去实现只需要TwinCAT PLC变量发生改变时，就能通知到上位机这个功能。 |
| **附 件：**

|  |  |  |
| --- | --- | --- |
| 序 号 | 文件名 | 备注 |
| 1、 | TwinCAT  | Rev：v3.1.4024.62 |
| 2、 | Visual Studio | Rev: 17.12.2 |
|  |  |  |
|  |  |  |

 |
| **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：**1. [Beckhoff Information System - German](https://infosys.beckhoff.com/) ：ADS.NET V6章节
 |

目 录

[1. 软硬件版本 3](#_Toc184897770)

[1.1. 倍福Beckhoff 3](#_Toc184897771)

[1.1.1. 控制器硬件 3](#_Toc184897772)

[1.1.2. 控制软件 3](#_Toc184897773)

[2. TwinCAT3编程 3](#_Toc184897774)

[2.1. TwinCAT在GVLs添加 Global Variable List并添加相关变量 3](#_Toc184897775)

[2.2. 激活并运行程序 4](#_Toc184897776)

[3. C#编程 4](#_Toc184897777)

[3.1. 编写如下代码： 4](#_Toc184897778)

[3.2. 运行调试如下： 6](#_Toc184897779)

[4. 总结 7](#_Toc184897780)

# 软硬件版本

## 倍福Beckhoff

### 控制器硬件

TwinCAT控制制器，PC或者EPC，包括：

嵌入式控制器：CX5020-0125（IMG版本：CX1800-0411-0007 v3.92）

工控机：C6xxx、CP2xxx、CP6xxx等

### 控制软件

笔记本和控制器都是基于TwinCAT 3.1 Build 4024.62版本

# TwinCAT3编程

## TwinCAT在GVLs添加 Global Variable List并添加相关变量

新建TwinCAT 3工程项目添加PLC项目工程，并新建gSubwayScreenDoorData：



同样要注意在PLC setting中勾选Symbloic Mapping



## 激活并运行程序



到此TwnCAT编程已经完成。

# C#编程

## 编写如下代码：

MAIN函数中如下代码：

新建 .NET 6.0的工程

//建立Ads通讯

using (AdsClient adsClient = new AdsClient())

{

 // Add the Notification event 'Ex' handler

 adsClient.AdsNotificationEx += Client\_AdsNotification;

 // Connect to target

 adsClient.Connect(AmsNetId.Local, 851);

 // 创建符号加载器

 var symbolLoader = SymbolLoaderFactory.Create(adsClient, SymbolLoaderSettings.Default);

 // 创建一个Byte类型数据

 ISymbol myVar1Symbol = symbolLoader.Symbols["gSubwayScreenDoorData.nMode"];

 // 创建一个bool类型数据

 ISymbol myVar2Symbol = symbolLoader.Symbols["gSubwayScreenDoorData.bClosed"];

 // 创建一个string类型数

 ISymbol myVar3Symbol = symbolLoader.Symbols["gSubwayScreenDoorData.sBarCode"];

 uint notificationHandle1 = 0;

 uint notificationHandle2 = 0;

 uint notificationHandle3 = 0;

在编写回调函数：

承接上文

 try

 {

 // 设置通知

 var notificationAttributes = new NotificationSettings(AdsTransMode.OnChange, 200, 0);

 notificationHandle1 = adsClient.AddDeviceNotificationEx(myVar1Symbol.InstancePath, notificationAttributes, myVar1Symbol, typeof(byte));

 notificationHandle2 = adsClient.AddDeviceNotificationEx(myVar2Symbol.InstancePath, notificationAttributes, myVar2Symbol, typeof(bool));

 notificationHandle3 = adsClient.AddDeviceNotificationEx(myVar3Symbol.InstancePath, notificationAttributes, myVar3Symbol, typeof(string),new int[] { myVar3Symbol.Size });

 Console.WriteLine("Notifications set. Press any key to exit...");

 Console.ReadKey();

 }

 finally

 {

 // Unregister the Event / Handle

 adsClient.DeleteDeviceNotification(notificationHandle1);

 adsClient.DeleteDeviceNotification(notificationHandle2);

 adsClient.DeleteDeviceNotification(notificationHandle3);

 adsClient.AdsNotificationEx -= Client\_AdsNotification;

 }

}

//编写回调函数

static void Client\_AdsNotification(object sender, AdsNotificationExEventArgs e)

{

 ISymbol symbol = (ISymbol)e.UserData;

 // 根据数据类型处理不同的通知数据

 if (symbol.TypeName == "BYTE")

 {

 byte byteValue = (byte)e.Value;

 Console.WriteLine($"Notification received: {symbol.InstancePath} = {byteValue}");

 }

 else if(symbol.TypeName == "BOOL")

 {

 bool boolValue = (bool)e.Value;

 Console.WriteLine($"Notification received: {symbol.InstancePath} = {boolValue}");

 }

 else if (symbol.TypeName == "STRING(8)")

 {

 string stringValue =(string)e.Value;

 Console.WriteLine($"Notification received: {symbol.InstancePath} = {stringValue}");

 }

}

## 运行调试如下：

C#后台输出和TwinCAT中值一致：





当我改变红框变量中值效果如下：

A: BOOL 变量发生改变



B: Byte 变量发生改变



C:String变量发生改变



# 总结

在TwinCAT 3 ADS (Automation Device Specification) 中，AddDeviceNotificationEx 是一个用于订阅设备变量变化通知的函数。通过这个函数，你可以注册一个回调函数，当指定的变量发生变化时，系统会调用你的回调函数，从而实现对变量变化的监控。

具体来说，AddDeviceNotificationEx 可以用于以下场景：

* 当设备变量（如PLC中的变量）变化时，自动触发特定的操作。
* 实现基于变量变化的逻辑控制。
* 监控和记录设备变量的变化情况。

ADS Notifications 这种响应式通讯的作用大致为减少网络负载和消息数量。

**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 |
|  |