**TcCOM基本概念与说明**

|  |  |
| --- | --- |
|  | 作者：余洋职务：技术支持工程师邮箱：yang.yu@beckhoff.com.cn日期：2023-08-04 |
| **摘 要：**TcCOM基于COM（Microsoft Windows世界的组件对象模型），旨在模块化不同的组件。一个驱动程序可以由一个或者多个TcCOM模块组成。 |
| **附 件：**

|  |  |  |
| --- | --- | --- |
| 序 号 | 文件名 | 备注 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

 |
| **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. TcCom中的交互流程 3](#_Toc142293140)

[1.1. 模块ID 3](#_Toc142293141)

[2. TcCom接口说明 3](#_Toc142293142)

[3. TcCom中模块的使用寿命（周期） 4](#_Toc142293143)

[4. TcCom中模块的属性 4](#_Toc142293144)

[5. TcCom模块状态机（TwinCAT Module State Machine） 5](#_Toc142293145)

[6. TcCom中模块的通讯 7](#_Toc142293146)

[6.1. IO映射 7](#_Toc142293147)

[6.2. IO数据指针 7](#_Toc142293148)

[6.3. 接口关联 7](#_Toc142293149)

[6.4. ADS通讯 7](#_Toc142293150)

# TcCom中的交互流程



首先是不同的TcCOM类，首先这些需要被创建实例。实例之间可以通过方法调用交互，实例需要通过objectserver实现。（objectserver是TwinCAT系统中的一个核心组件，它充当了一个中央服务器的角色，用于管理和协调系统中的对象）类的实例与ObjServer访问是通过TcQueryInterface交互（TcQueryInterface是TwinCAT中的一个函数，用于查询对象是否实现了指定的接口，并获取该接口的引用）。

## 模块ID

模块自身会带有ID，TcCOM使用的ID有两种：

1. GUIDs：ModulIDs, ClassIDs and InterfaceIDs.
2. 32 bit long integers: ParameterIDs, ObjectIDs, ContextIDs, CategoryID.

# TcCom接口说明

接口是TcCOM非常很重要的组成部分。基于TwinCAT规范，所有接口都派生自基本接口ItcUnknown。（ItcUnknown提供了TcQueryInterface和控制模块生命周期的服务）因此，每个模块都有一些固定的接口：

1. ITcCyclic ：由模块实现，这些模块称为循环（“CycleUpdate”）。该模块可以通过 TwinCAT 任务的 ITcCyclicCaller 接口注册以获取循环调用。
2. ITcADI：可用于访问模块的数据区域。
3. ITcWatchSource：用于ADS设备通知和其他功能。
4. ITcTask：由实时系统任务实现，提供有关周期时间，优先级和其他任务信息的信息。
5. ITComObjectServer ：接口由 ObjectServer 实现，并由所有模块引用。

# TcCom中模块的使用寿命（周期）



模块的使用寿命通过参考计数器（RefCounter）确定。每当查询模块接口时，参考计数器都会递增。释放接口时，计数器递减。当模块登录到 ObjectServer（ITComObject 接口）时，还会查询接口，此时计数器至少为 1。注销时计数器递减。当计数器达到 0 时，模块会自动删除释放内存资源。如果有别的模块通过接口指针调用此模块，那么该模块将继续存在，并且接口指针保持有效，直到释放此指针。

# TcCom中模块的属性

TcCOM模块中有很多属性，有些是可选（浅蓝色），有些是必带属性（深蓝色）：



1. Module Desctiption：每个 TcCOM 模块都有一些通用描述参数，比如ClassID，Class描述文件（.tmc）和实例描述文件（.tmi）。
2. State Machine（状态机）：每个Module都包含一个状态机，它描述了模块的初始化状态以及可以从外部修改此状态的方法。状态机描述在模块启动和停止期间发生的状态。这涉及模块的创建、参数化等等。TcCOM 模块的状态机定义了状态 INIT、PREOP、SAFEOP 和 OP。尽管状态名称与 EtherCAT 现场总线下的名称相同，但实际上是完全不同的两个概念。
3. ITComObject Interface/ interface：两种接口一种是模块通用的接口，每个模块都必须有，另外一种是开发用户可以自定义的接口。
4. Parameters：参数是用户自定义的一些数据，参数可以在初始化期间读取或写入，也可以在运行时读取或写入（OP 状态）。每个参数都由对应的参数 ID 指定。参数 ID可以是全局的、有限的全局或特定于模块的。
5. Interface pointers：接口指针是接口的对应项，如果一个模块想要使用另一个模块的数据接口，必须具有接口类型的指针才能调用，确保它能够指向另一个模块。
6. Data Areas：模块里包含了数据区域，用于存放数据。数据区域里可以包含任何数据，比如输入输出，或者中间变量等等。
7. Data Area Pointer：同样，如果一个模块下要访问其他模块的数据区域，它必须包含了数据区域的指针，访问直接到内存区域。
8. Context：在模块处理任务时，即对应的是实时任务。需要考虑上下文信息。这个内容对于配置模块十分重要。如EtherCAT主站可以支持多个独立的实时任务，也就是多个与时间相关的上下文（代码）。这些内容必须在模块描述中进行配置和指定。
9. Categories：模块可以通过实现接口 ITComObjectCategory 来提供类别。类别由 ObjectServer 枚举。
10. ADS：每个模块都可以通过ADS访问，每个模块都有对应的ADS端口，通过该端口可以接受专用的ADS指令（非实时）。

模块可以用C++和ST语言创建。TwinCAT PLC的面向对象扩展就是为创建模块量身定制。两个不同类型的模块可以通过接口进行交互，方式与纯C++模块相同。面向对象的扩展使接口与 C++ 中相同。PLC模块也通过ObjectServer进行注册，并且通过它访问。PLC模块在复杂性方面有所不同。无论是只生成一个小的FB模块，还是将完整的PLC程序打包到一个模块中，都没有区别，因为每个PLC程序都是TwinCAT模块意义上的模块。PLC程序会自动打包到一个模块中，并在ObjectServer和一个或多个任务模块中注册自身。对模块过程数据的访问（例如，与现场总线驱动器有关的映射）也通过定义的数据区域和ITcADI进行控制。

# TcCom模块状态机（TwinCAT Module State Machine）

每个TwinCAT模块都有对应的状态机（INIT、PREOP、SAFEOP、OP），这个状态机和EtherCAT状态机的概念类似，但是内容缺大相径庭。除了某一个特定的状态之外TwinCAT模块还有对应的转换过程状态（IP、PI、PS、SP、SO、OS）。

**（1）INIT：**

初始化状态只是一个虚拟状态。创建模块后，模块就会立即从 INIT 转化为 PREOP，即执行 IP 状态转换。实例化和 IP 同时发生，因此模块不会保持 INIT 状态。只有当模块被移除时，它才会在短时间内保持INIT状态。

**（2）IP（INIT TO PREOP）：**

在 IP 状态转换期间，模块使用其唯一的 ObjectID 向对象服务器（ObjectServer）注册。初始化参数（也在对象创建期间分配）将传输到模块。在此转换期间，模块无法建立与其他模块的连接，因为不清楚其他模块是否已存在并已注册到 ObjectServer。当模块需要系统资源（例如内存）时，可以在状态转换期间分配这些资源。同理，在从PREOP反过渡到INIT（PI）期间，必须再次释放所有分配的资源。

**（3）PREOP：**

在 PREOP 状态下，模块创建完成，模块通常已完全参数化（不排除在从 PREOP 过渡到 SAFEOP 期间可能会添加更多参数）。该模块已在 ObjectServer 中注册，但尚未创建与其他模块的连接。

**（4）PS（PREOP TO SAFEOP）：**

在此状态转换中，模块可以与其他模块建立连接。为此，它通常会接收其他模块的 ObjectID 以及初始化数据，这些 ObjectID 现在通过 ObjectServer 转换为与这些模块的实际连接。在此状态转换期间，可以传输更多参数。例如，父模块可以将自己的参数传输到子模块。

**（5）SAFEOP：**

模块在此阶段仍处于非实时上下文中，正在等待系统或其他模块切换到OP状态。

**（6）SO（SAFEOP TO OP）：**

从 SAFEOP到OP状态的转换在实时上下文中进行，此时不再分配系统资源。该模块的资源现在可以由其他模块请求，同时该模块也可以向其他模块注册，例如为了在任务期间获得循环调用。（此转换不应用于长时间运行的任务：如，文件操作应在PS期间执行。）

**（7）OP：**

在OP状态下，模块开始工作，并在TwinCAT中完全处于活动状态。

**（8）OS（OP TO SAFEOP）：**

在此状态转换在实时上下文中发生。SO转换中的所有操作都将被撤消，SO转换期间请求的所有资源将再次释放。

**（9）SP（SAFEOP TO PREOP）：**

PS 转换中的所有操作都将撤消，并且PS转换期间请求的所有资源将再次释放。

**（10）PI：（PREOP TO INIT）**

IP 转换中的所有操作都将被撤消，并且IP 转换期间请求的所有资源都将再次释放。模块从 ObjectServer 注销并通常会自行删除（见使用寿命）。

# TcCom中模块的通讯

## IO映射

TcCOM中模块的输入和输出可以通过IO映射进行链接，就像PLC链接到实际物理模块的输入输出通道一样。对于C++模块，可以[在 TMC 编辑器中创建描述](https://infosys.beckhoff.com/content/1033/tc3_c/721886859.html)相应输入/输出的数据区域。然后将这些链接在 TwinCAT 解决方案中。C++、PLC、MATLAB都可以使用该方法。



## IO数据指针

可以通过 TMC 编辑器中创建的数据区域指针在任务中直接去访问内存。但是如果出现一个任务的多个调用方或其他任务的调用方，用户必须通过适当的机制确保数据一致性，可用于C++和 MATLAB。

## 接口关联

TcCOM 模块可以提供也在 TMC 编辑器中定义的接口。在TMC编辑器当中建立完接口之后，它将提供适当的方法。然后，调用模块将具有指向该模块的“接口指针”，以便调用方法。使用接口的方式可能会造成调用阻塞（返回值产生之后才会释放）。如果出现一个任务的多个调用方或其他任务的调用方，用户必须通过适当的机制确保数据一致性。一般使用在C++里。

## ADS通讯

作为TwinCAT系统的内部通信方式，ADS可用于模块之间的通信（C#中是非实时的）。ADS还可用于从UserMode收集或提供数据，并与其他控制器通信（即通过网络）。此外ADS还可用于确保数据一致性通信，例如任务/内核/CPU之间的通信。在这种情况下，TcCOM模块既可以是客户端，也可以是服务器。PLC，C++，Matlab都可以使用（具体的使用方法见对应语言的ADS库文件以及Automation Interface）

**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 |
|  |