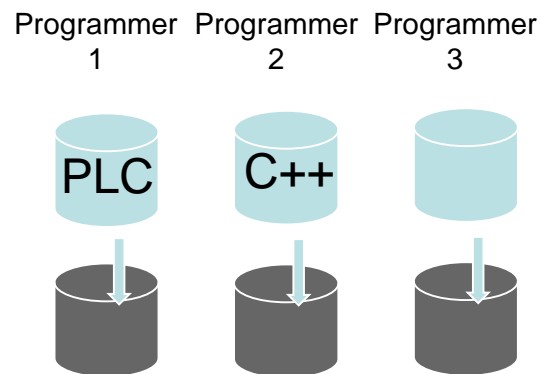


## TwinCAT 3: eXtended Automation



# TwinCAT 3 概览

- Engineering
  - 集成到Visual Studio中
  - 支持不同编程语言 IEC61131-3 / C++ / Matlab & Simulink
  - 模块化概念: 模块可以用不同语言实现
- Run time
  - 二进制模块无需编译就可以任意组合使用
  - 支持多核
  - TC3模块在runtime中可以互相映射



不同的已编译的二进制文件

- 便于测试
- 只需要配置无限再次编译

下载

- 通过不同的任务在不同的核中被触发执行

# 编程电脑要求

- 编程电脑要求



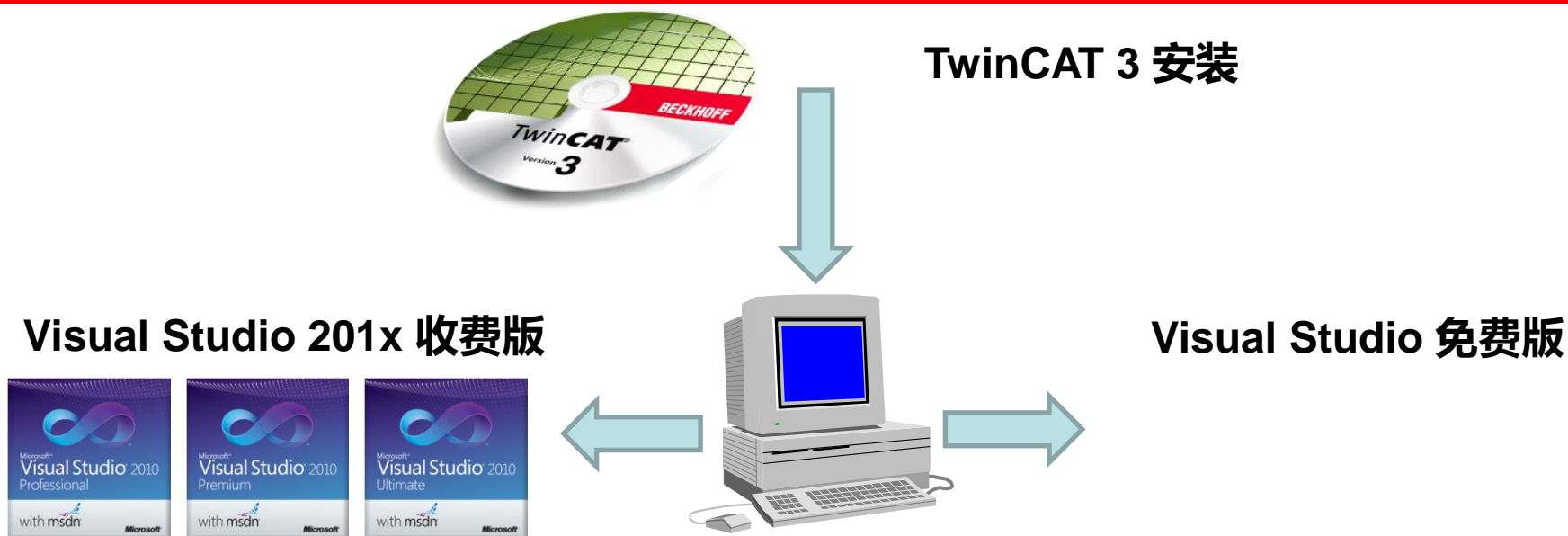
- Windows XP 或 Windows 7 / 8  
x86 和 x64 平台都支持
- Microsoft Visual Studio 2010 / 2012 / 2013 的一个完整版本  
专业版, 高级版, 旗舰版等

Beckhoff 只是 Visual Studio 合作伙伴 – 但不是VS的代理商

- Microsoft Windows Driver Kit (WDK) v7.1.0  
可以从网上下载 (大约700MB)  
具体可以看TC3的帮助文档系统
- TwinCAT 3 XAE Engineering (500MB 安装包)  
基本工程界面是免费的



# 可交付产品: 工程界面



## TwinCAT 3 安装

### Visual Studio 201x 收费版

### Visual Studio 免费版

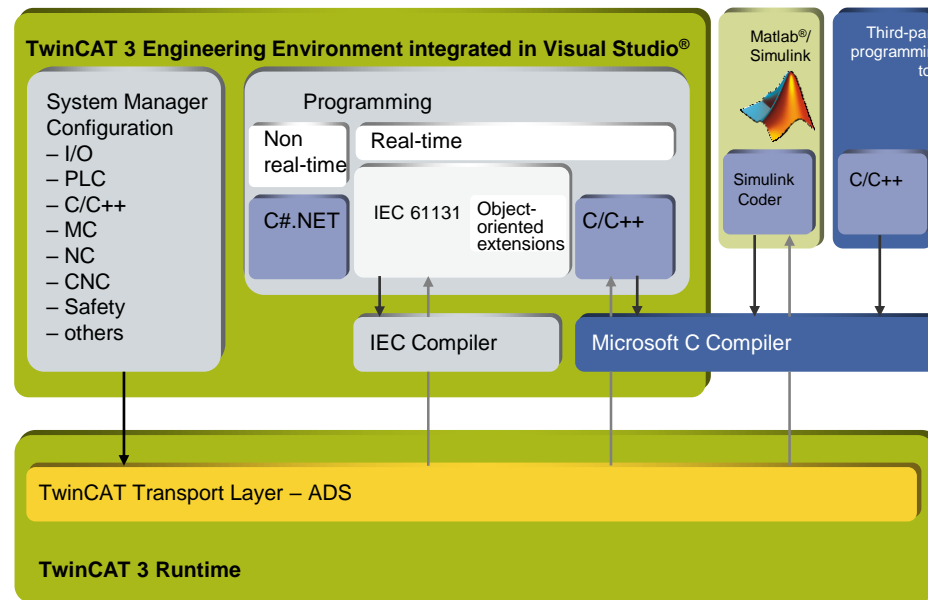
- **TwinCAT 3 集成到VS201X收费版**
- 集成到 Microsoft Visual Studio中  
TwinCAT 作为一个全新的模板, 就像C#/C++一样
- 集成了系统配置 / IEC 61131-3编程
- 集成了安全PLC
- C 和 C++ 编程/调试 (实时)
- 支持与Matlab Simulink实现交互
- C# / .NET 编程 (非实时)  
可以应用在同一个环境中(VS)

- **TwinCAT 3 安装包中包含免费VS版本 VS2010-Shell**
- 提供 Microsoft Visual Studio Shell  
也可以继承到最新版本的shell中
- 集成了系统配置 / IEC 61131-3编程
- 集成了安全PLC
- 执行 C/C++ 和 Matlab 发布的模块  
(不能编程/调试)

# 扩展自动化工程(XAE)

## TwinCAT 3

- 可供PLC 和 C/C++ 程序员使用
- 组态, 配置 和 诊断
- 模块生成 (C/C++ 或 Matlab/Simulink)
- 对PLC, C/C++, Matlab/Simulink进行调试



# Runtime PC的要求: C++

## 在Run-Time 设备上的要求

- TwinCAT 3 XAR Runtime (安装包 100MB)
- Windows XP, Windows 7/8, Windows Embedded Standard  
注意: WindowsCE 只支持PLC的执行 – 不支持 C++ RT)
- 不需要 Microsoft Visual Studio 和 Microsoft Windows Driver Kit required (!)  
调试 和 监控 C++代码可以通过远程编程PC上完成
- 许可证

C++ runtime 许可证可以反复用7天试用版

商业版C++ runtime许可证包括以下几种:

TC 1300-00xx : TwinCAT 3 C++

TC 1320-00xx : TwinCAT 3 C++/Matlab

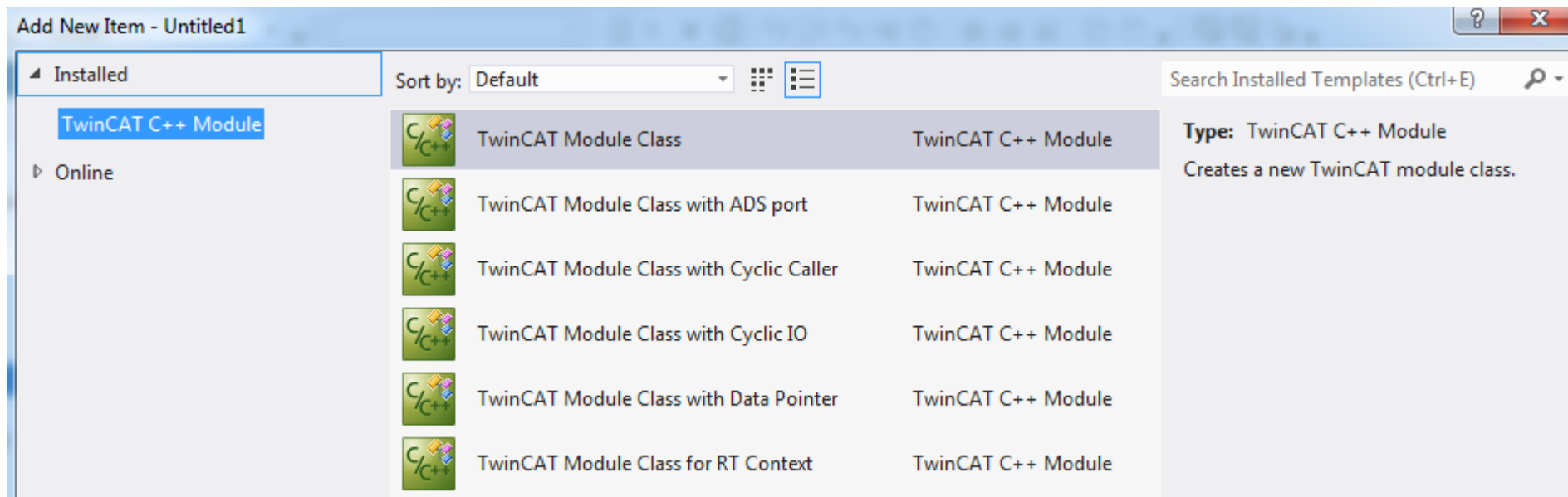
TC 1210-00xx : TwinCAT 3 PLC/C++

TC 1220-00xx : TwinCAT 3 PLC/C++/Matlab



# TwinCAT 3 – C++: 向导工具使得工程创建更简单

- TwinCAT3 C++ 提供3个向导用来快速创建工程
  - 项目向导
  - 类向导提供了6种模板



**例如：“TwinCAT Module Class with Cyclic IO“**

创建一个新的 TwinCAT 类模块，实现了循环调用的接口，以及包含了输入和输出的数据区

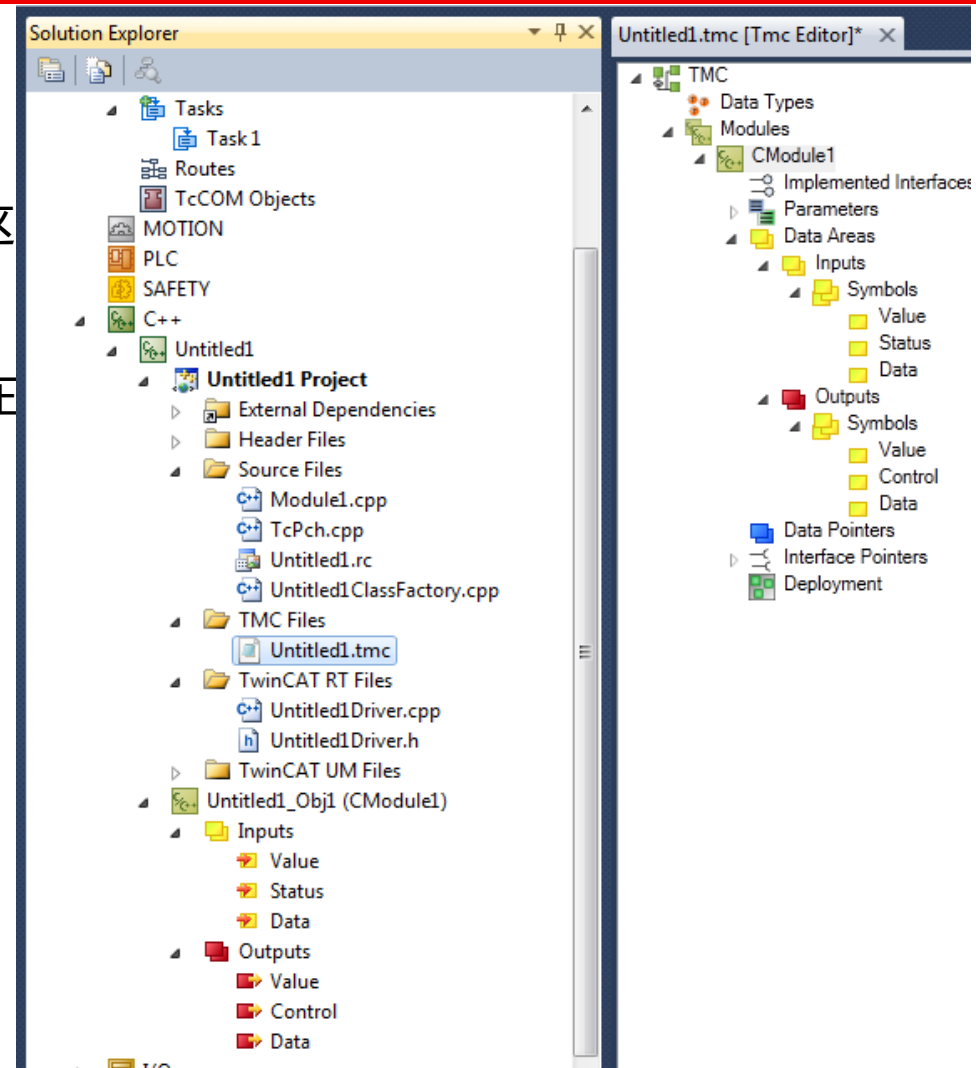
# TwinCAT 3 – C++: 向导创建了输入/输出/方法

## TwinCAT3 – C++ 模块支持输入, 输出, 参数, 指针等等

- 在PLC语言中定义一个输入或者输出可以这样: `>bln AT%I* : bool;<`
- 但在C++中没有这种语法定于输入和输出
- C++: 都把模块输入/输出/参数等信息定义在XML文件中: .tmc
- 利用symbol向导来生成和修改变量

### 注意:

- 每一个 TwinCAT 3 类模块只有一个 TMC 文件
- 每一个类实例都有一个TMI文件





# TwinCAT 3 – C++ : Symbol向导

- TwinCAT3 TMC 文件编辑器

The screenshot displays the TwinCAT 3 TMC Symbol Wizard interface. On the left, a project tree shows the following structure:

- Data Types
  - Modules
    - CMyModule
      - Data Areas
        - Inputs
          - Symbols
            - Value** (selected)
            - Status
            - Data
          - Outputs
            - Symbols
              - Value
              - Control
              - Data
- Parameters
- Interface Pointers
- Data Pointers

The main area of the wizard is titled "Edit the properties of the Symbol." and is divided into three sections:

- General properties**
  - Name: Value
  - Specification: Alias
- Define the data type**
  - Select: UDINT
  - Type Information
    - Name: UDINT
    - Namespace:
    - Guid:
    - Resolve Type button
- Optional symbol settings**
  - Offset [Bits]:
  - Size [Bits]:
  - Unit:
  - Comment:
  - Create symbol

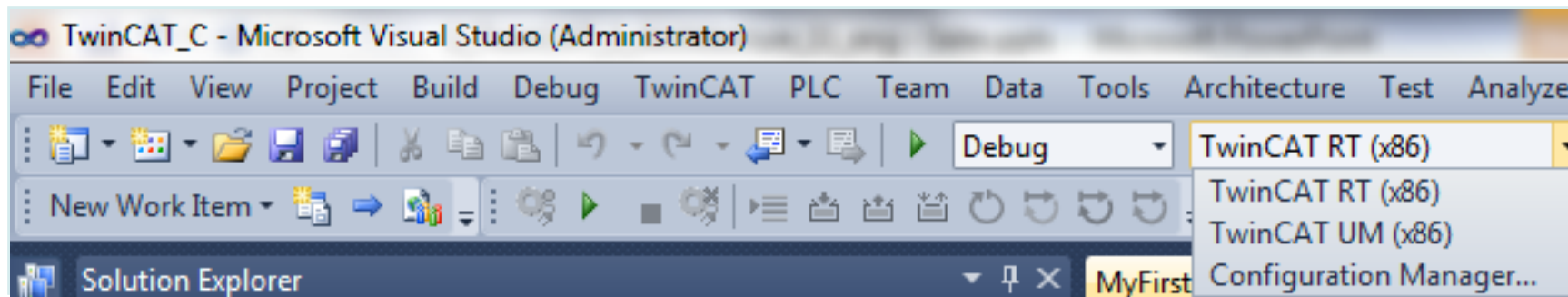
# TwinCAT 3 – C++: CycleUpdate 程序

方法 CycleUpdate: – 从一个Task中周期调用

```
#####  
// ITcCyclic  
HRESULT CMyModule::CycleUpdate(ITcTask* ipTask, ITcUnknown* ipCaller, ULONG context)  
{  
    HRESULT hr = S_OK;  
  
    // TODO: Replace the sample with your cyclic code  
    m_counter+=m_Inputs.Value;  
    m_Outputs.Value=m_counter;  
  
    return hr;  
}
```

# TwinCAT 3 – C++ : runtimes

- TwinCAT3 C++ 支持不同目标平台:
  - TwinCAT RT Runtime for x86/32bit 和 x64 OS
  - TwinCAT UM User Mode-x86/32bit 和 x64 OS
  - TwinCAT CE (x86/ARM) Runtime (仅支持 PLC – 不支持 C++)



- 注意:
  - TC3 PLC 模块是本地代码 – 但PLC提供了一个托管 runtime 环境: 当设备正在运行的时候可以“Online Change”
  - TC3 C++ 模块是本地内核模式驱动程序 – 非托管 runtime  
Focus: Speed! (非托管效率高)

# TwinCAT 3 – C++: 调试器

VS201x 标准调试器:

监控 / 修改变量只能通过添加断点方式来完成

```
TC3_IEC_CPP_Sample (Running) - Microsoft Visual Studio
File Edit View Project Build Debug TwinCAT
Process: Thread:
CustomCode.cpp FairCppDemoServices.h
(Unknown Scope)
ULONG nCounter;
HRESULT CustomCodeCyclic(PFairCppDempIn
{
    //pOut->value = pIn->value;
    //pOut->value = pOut->value++;
    nCounter++;
    pOut->value = nCounter;
    return S_OK;
}
```

Name	Value
this	Invalid Expression
pOut->value	2890
nCounter	2891

```
TC3_IEC_CPP_Sample (Debugging) - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug TwinCAT Team Data Tools Architecture Test Analyze Wi
Process: [0] TwinCat Runtime Thread: [256] TwinCat Engine Thread
CustomCode.cpp FairCppDemoServices.h CustomCode.h
(Unknown Scope)
ULONG nCounter;
HRESULT CustomCodeCyclic(PFairCppDempInputs pIn, PFairCppDempOutputs pOut)
{
    //pOut->value = pIn->value;
    //pOut->value = pOut->value++;
    nCounter++;
    pOut->value = nCounter;
    return S_OK;
}
```

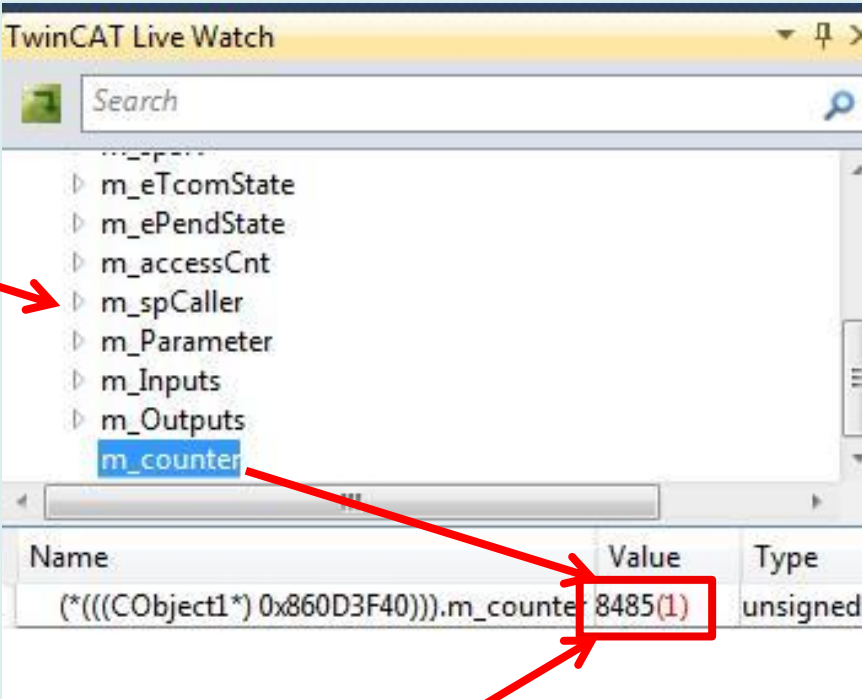
Name	Value	Type
this	Invalid Expression	
pOut->value	2890	unsigned
nCounter	2891	unsigned

Name
FairCppDe

# TwinCAT 3 – C++: 调试器

TwinCAT 3 支持不需要添加断点就可以在线监控到变量:

变量列表  
(Symbolic) →



The screenshot shows the 'TwinCAT Live Watch' window. It has a search bar at the top. Below it is a tree view of variables: m\_eTcomState, m\_ePendState, m\_accessCnt, m\_spCaller, m\_Parameter, m\_Inputs, m\_Outputs, and m\_counter. The 'm\_counter' variable is selected and highlighted in blue. Below the tree view is a table with three columns: Name, Value, and Type. The table contains one row: Name: (\*((CObject1\*) 0x860D3F40)).m\_counter, Value: 8485(1), Type: unsigned. A red box highlights the value '8485(1)'. Red arrows point from the Chinese text annotations to the tree view and the value in the table.

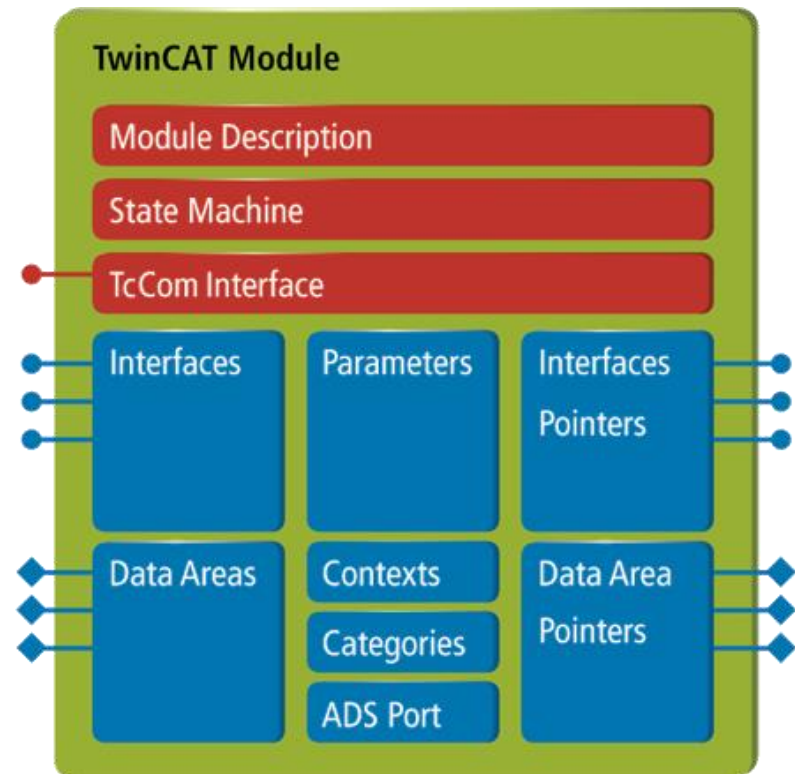
Name	Value	Type
((*(CObject1*) 0x860D3F40)).m_counter	8485(1)	unsigned

实际值和括号中需要强制的值

# 扩展自动化实时 (XAR)

## 模型实时接口

- 封装的功能划分到不同的模型中
- 可扩展性: 模型可以包含简单的功能, 复杂的算法, 一个实时任务, 甚至完整项目
- 目标: 模块化设计方便反复利用
- 标准化: 定义接口和定义行为
- 合作: 可以用不同语言创建模型
  - IEC 61131-3
  - C/C++
  - Matlab/simulink



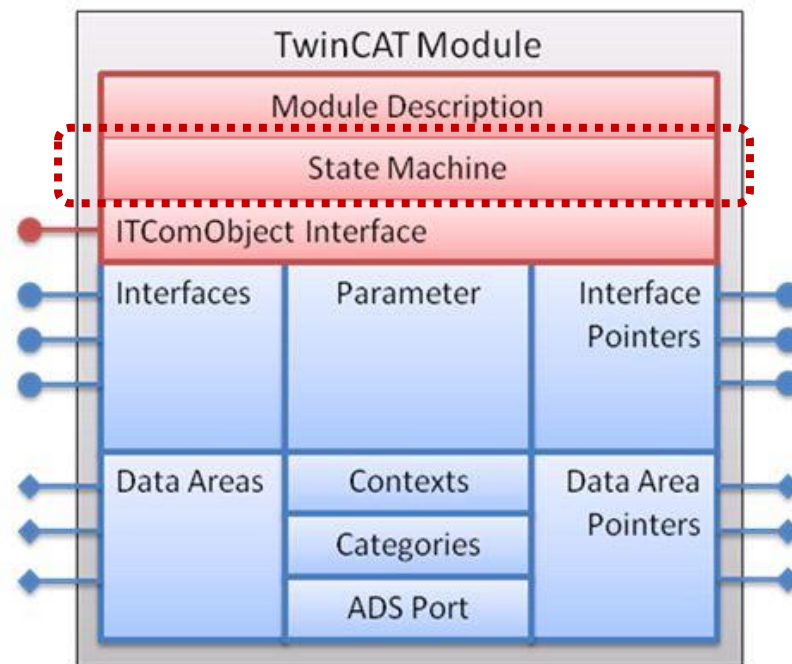
# 模块: 状态机

## 必须的

- 模块的描述
- 状态机
  - 描述模块初始化状态

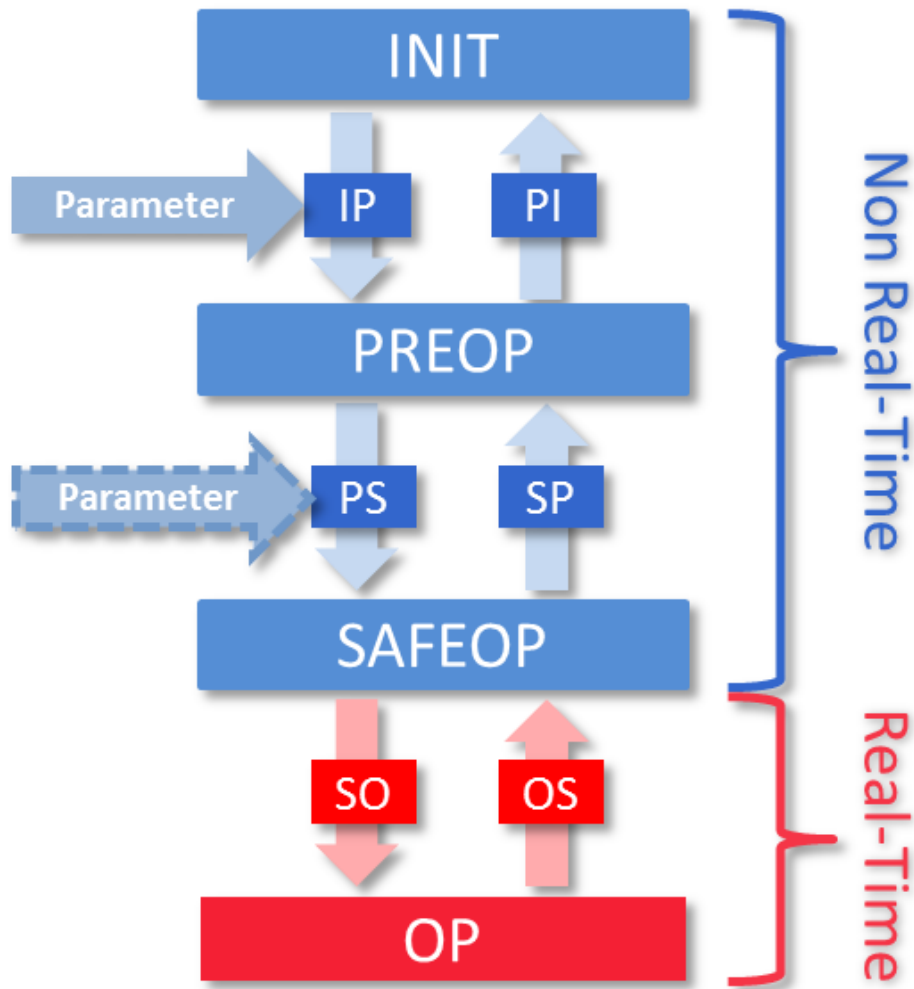
```

on
// State Transitions
// State transition from PREOP to SAFEOP
IMPLEMENT_ITCOMOBJECT_SETOBJSTATE_IP_PI(CModule1)
// State transition from PREOP to SAFEOP
// Initialize input parameters
// Allocate memory
HRESULT CModule1::SetObjStatePS(PTComInitDataHdr
Declarations
HRESULT CModule1::SetObjStateSO() { ... }
Declarations
HRESULT CModule1::SetObjStateOS() { ... }
Declarations
HRESULT CModule1::SetObjStateSP() { ... }
    
```



# 模块: 状态机文档

文档: <http://infosys.beckhoff.com>



必须的

状态机

- 描述模块初始化状态
- 定义了模型生成, 参数化, 以及创建与其他模块连接的顺序

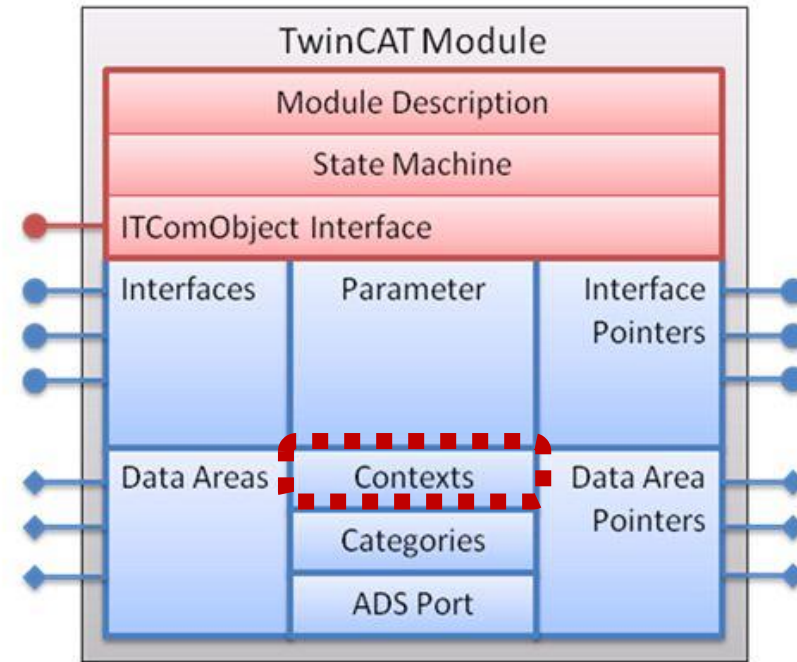


# 模块: 环境

## 可选的

- 环境
  - 实时配置

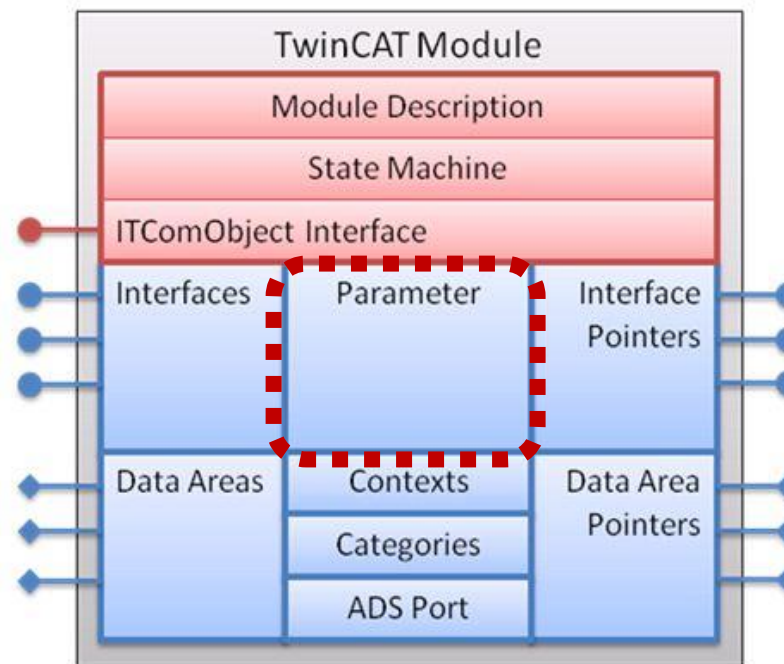
ID	Task	Name	Priority	Cycle ..
1	02000101	Task 1	1	10000



# 模块: 参数

## 可选的

- 参数
  - 通过INIT/OP进行读写操作
  - 可以写保护
  - 包含实参
  - 可通过ADS获取

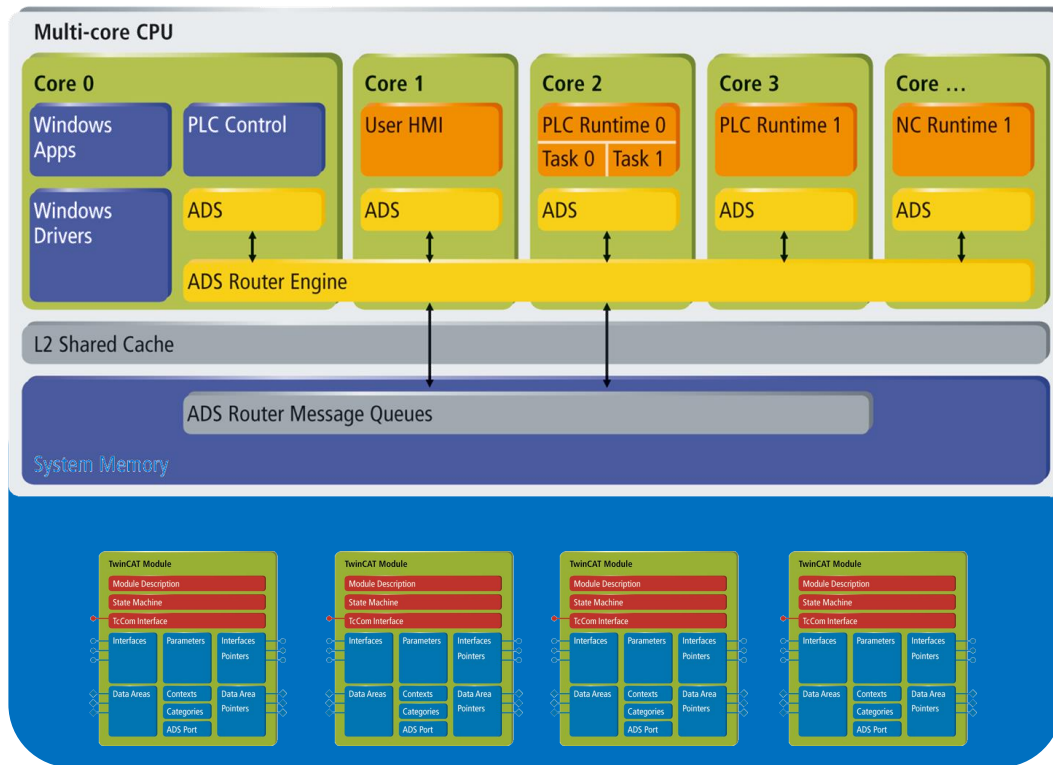


Object	Context	Parameter (Init)	Data Area	Interfaces	Interface Pointer	
PTCID	Name	Value	CS	Unit	Type	Com...
- 0x00000001	Parameter		<input type="checkbox"/>			
	.data1	0			UDINT	
	.data2	0			UDINT	
	.data3	0.0			LREAL	

# 模块与模块之间通信

## 目标

- TwinCAT3 支持多核系统
  - 任务在内核中执行
  - 模块充分利用内存



## 挑战:

- 模块之间的交互

## 如何...

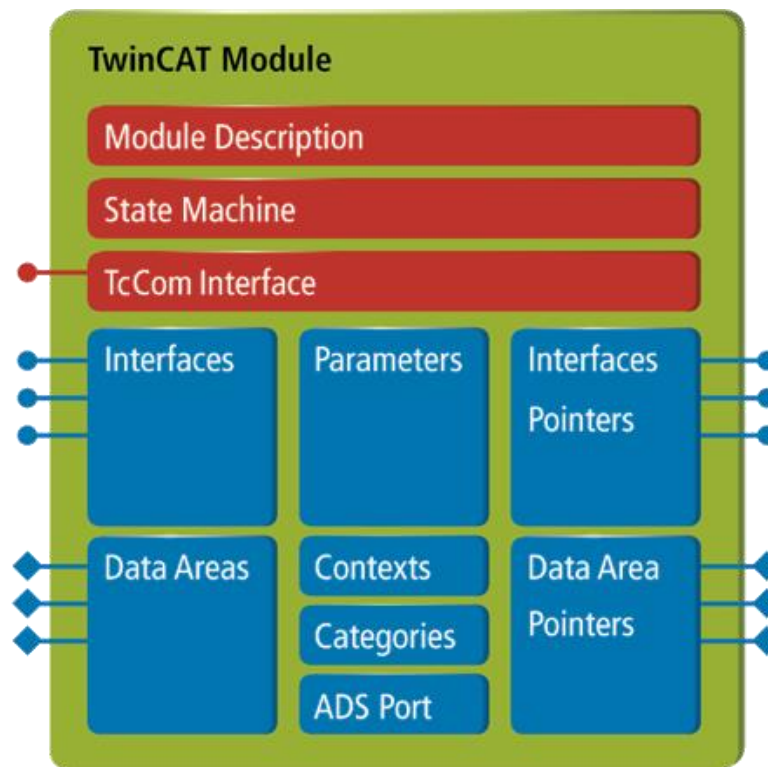
- 保证数据一致性?
- 对重入的保护?

# 模块与模块之间通信

## 方式

4 种不同的方法可以实现

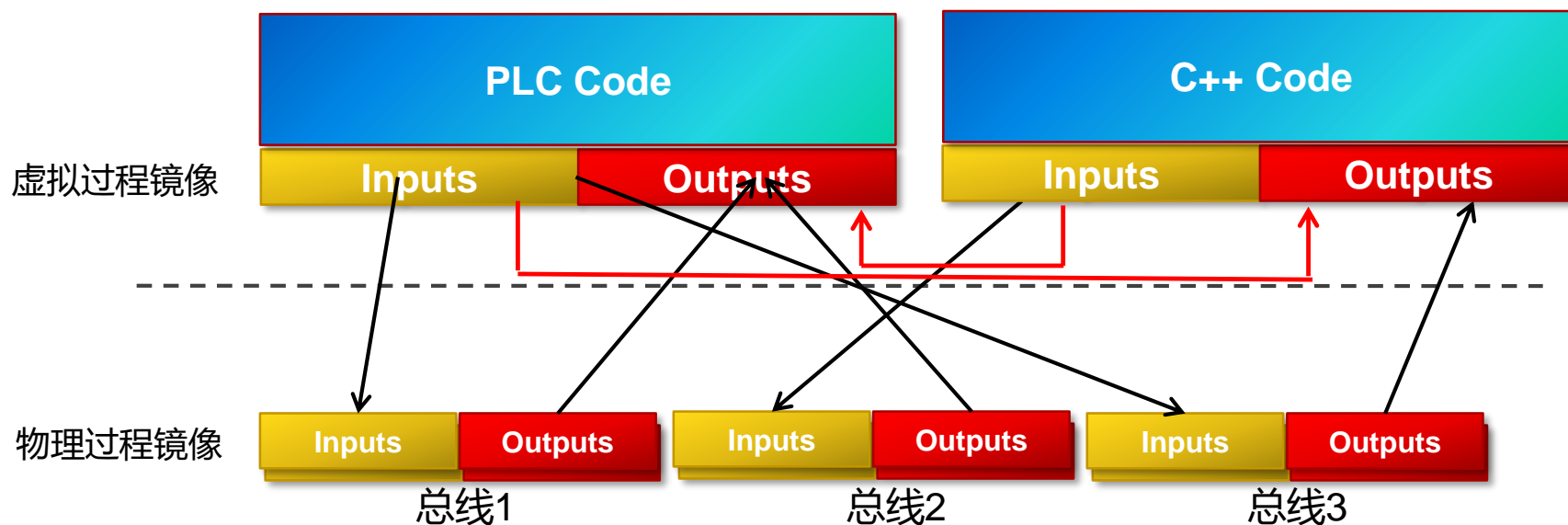
1. 数据区数值映射
2. 数据指针映射
3. 访问接口
4. ADS 通信
  - ADS 服务器
  - ADS 客户端



# 模块与模块之间通信

## 数据区数值映射

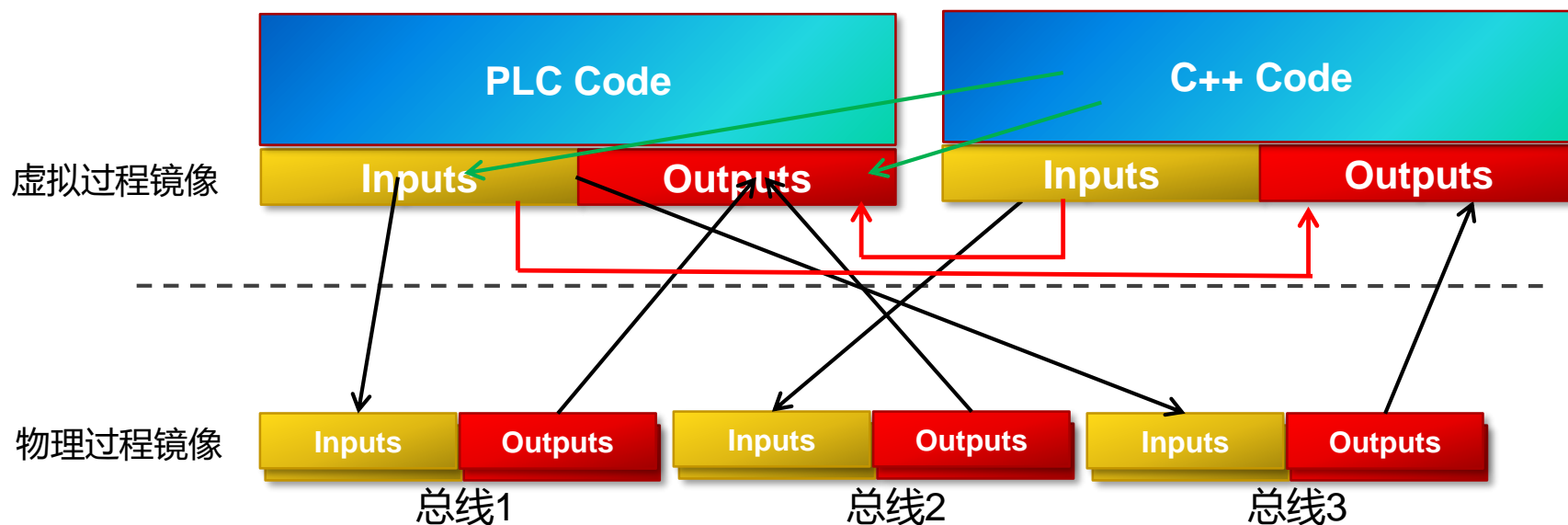
- 数值映射
  - 物理IO和逻辑数据区之间
  - 数据区与数据区之间
- 所有方向都支持 PLC/C++/Matlab
- 数据一致性可以保证
- 在任务周期中完成映射



# 模块与模块之间通信

## 数据指针映射

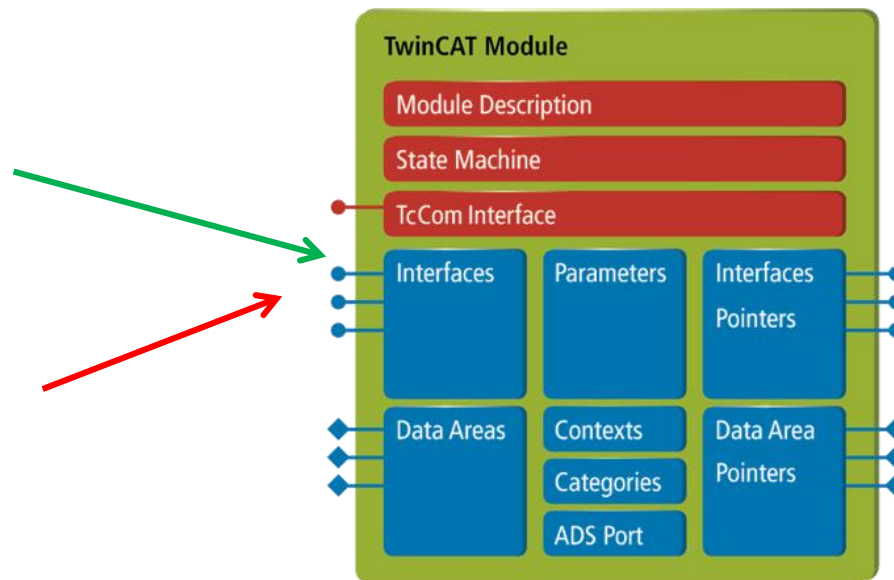
- 指针映射
  - 物理IO与逻辑数据区之间
- 所有方向都支持 PLC/C++/Matlab
- 用户自己考虑数据的一致性
- 立刻访问代码



# 模块与模块之间通信

## 访问接口

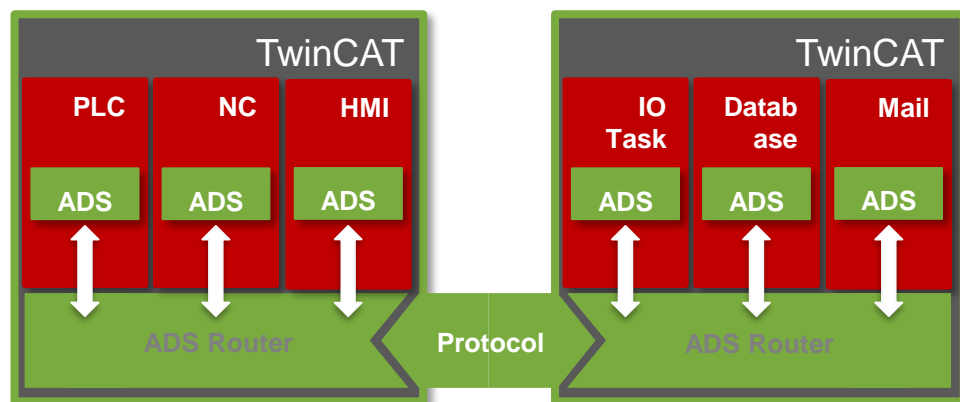
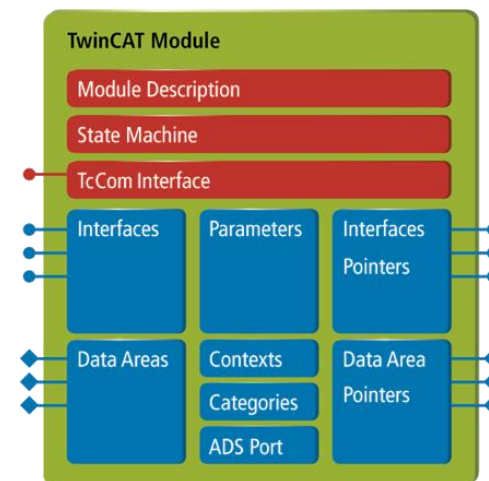
- PLC 可以访问 C++ 或 Matlab Simulink 中的接口  
目前: C++ 不能访问 PLC 中的接口
- 立刻访问代码, 访问者被阻止直到方法完全执行并且返回给访问者
- 用户自己考虑数据一致性
  - 在一个周期中一个访问者访问  
另一个模块没有问题 – 无需保护
  - 在不同周期中多个访问者  
访问同样接口就必须实现同步机制  
例如. 加入临界区



# 模块与模块之间通信

## ADS 通信

- 一种循环/事件驱动的通信
- ADS 客户端发送ADS报文
- ADS 服务器接受并且分析每一个ADS端口中ADS报文
- 支持TC3模型与用户应用程序通信
- 支持TC3模型直接通信
- 支持在多核多CPU
- 所有方向都支持 PLC/C++/Matlab
- 数据一致性可以保证





# 联系我们

## 德国倍福自动化有限公司

### 上海总部

上海市汶水路299弄9号 (市北智汇园)



电话: 021-6631 2666

传真: 021-6631 5696

E-Mail: [support@beckhoff.com.cn](mailto:support@beckhoff.com.cn)

Web: [www.beckhoff.com.cn](http://www.beckhoff.com.cn)

虚拟学院: <http://tr.beckhoff.com.cn>

FTP: <ftp://ftp.beckhoff.com.cn>

技术热线: 4008207388

扫一扫, 关注倍福官方微信!

© 德国倍福自动化有限公司

本 PowerPoint 演示文稿中的所有照片及图片均受版权保护。未经许可, 任何用户不得擅自复制、使用、转载或将其提供给任何第三方。

Beckhoff®、TwinCAT®、EtherCAT®、Safety over EtherCAT®、TwinSAFE®、XFC® 和 XTS®是德国倍福自动化有限公司的注册商标。本 PowerPoint 演示文稿中所使用的其它名称可能是商标名称, 任何第三方为其自身目的而引用, 都可能触犯商标所有者的权利。

本PowerPoint 演示文稿中所包含的信息仅是一般描述或性能特征简介, 在实际应用中并不总是与所述完全一致或者可能由于产品的进一步开发而不完全适用。仅在书面认同情况下, 才提供相关特性信息。

