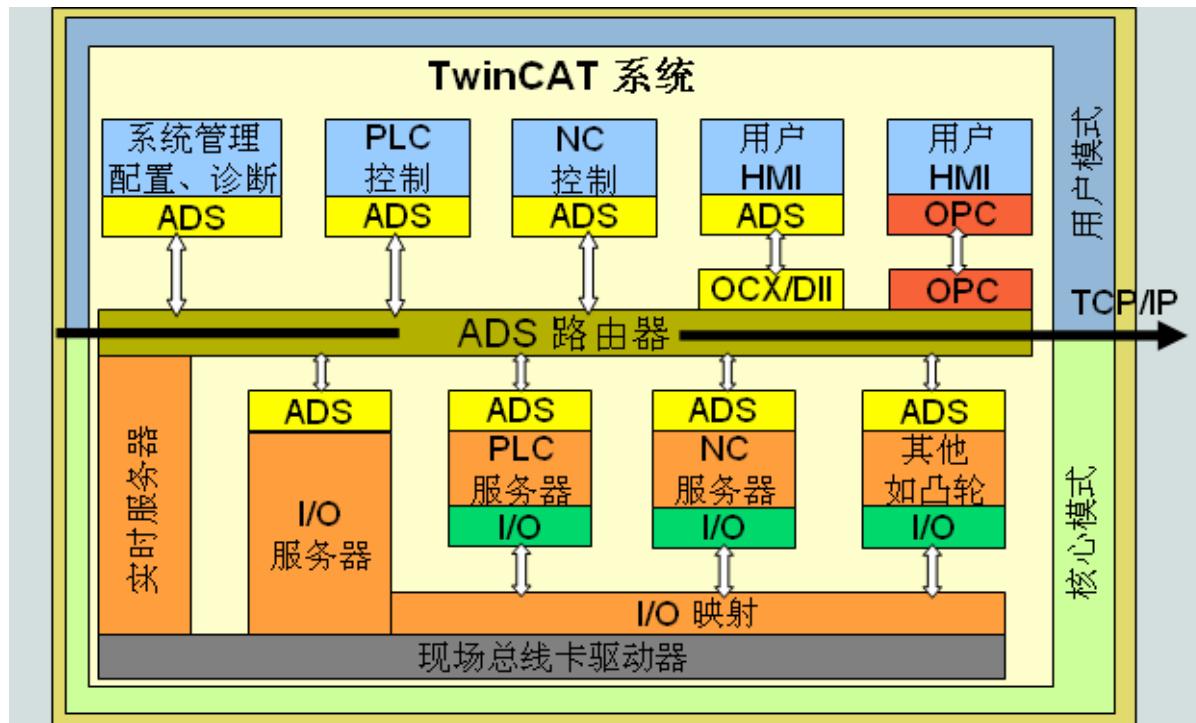


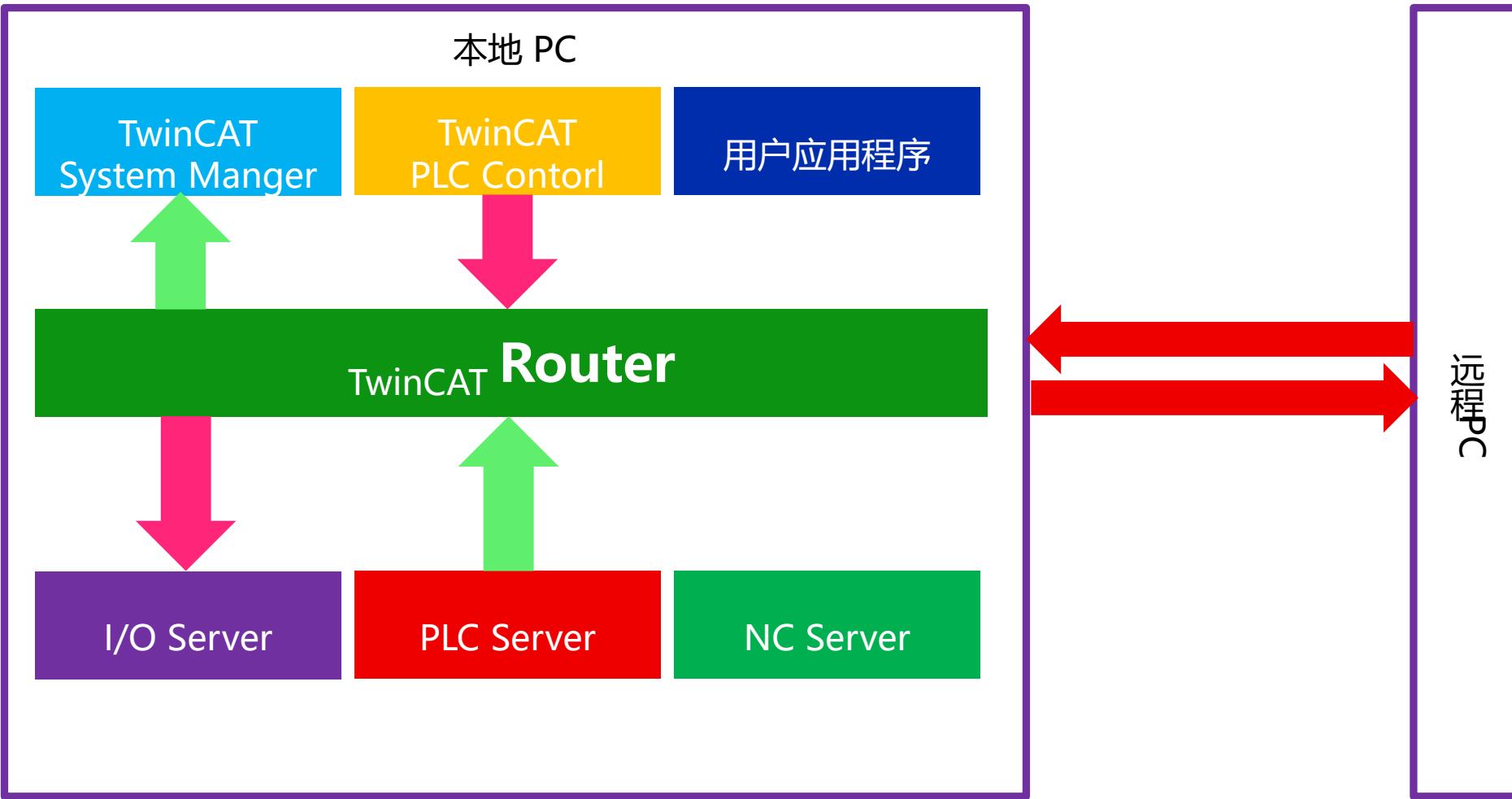


Ads基本原理

- ADS即 (Automation Device Specification) 自动化设备规范。
- TwinCAT系统各模块均作为独立的设备
- 每个任务均存在一个服务模块，服务端或客户端
- 由Message Router统一交换数据

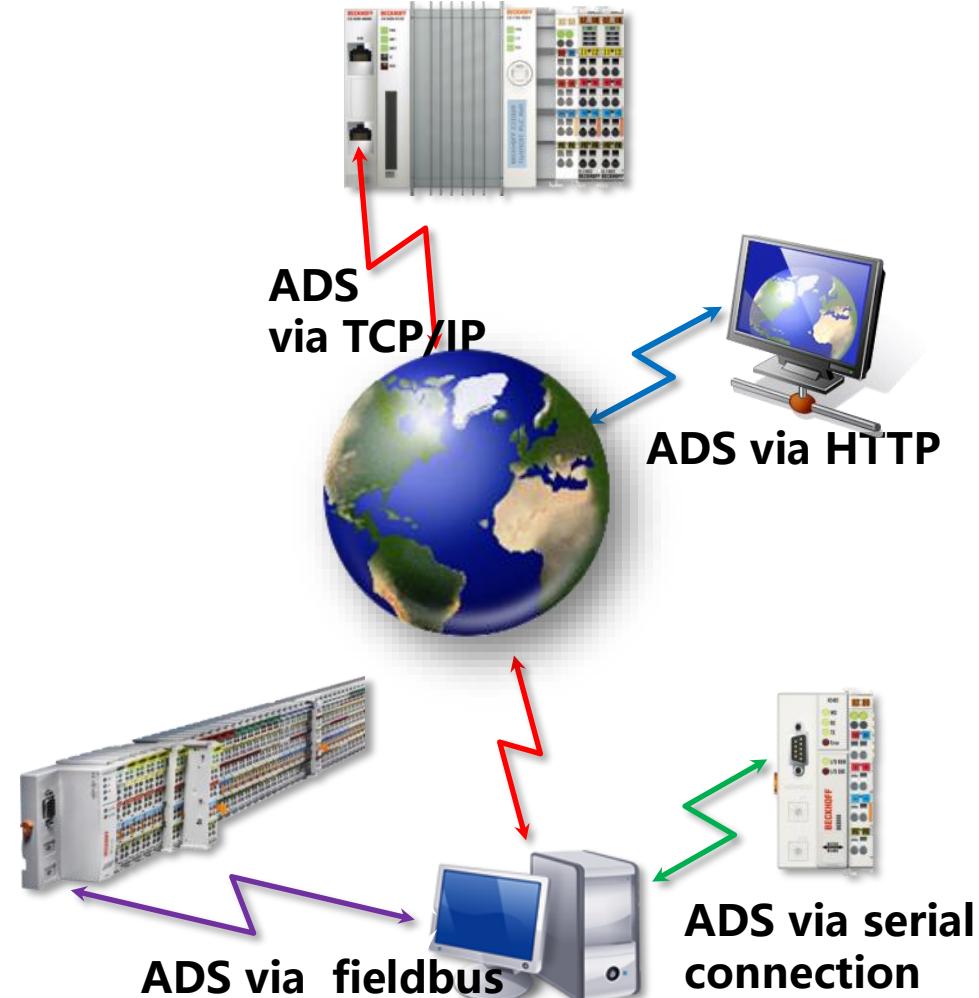


- ADS通过ADS Router进行数据交互

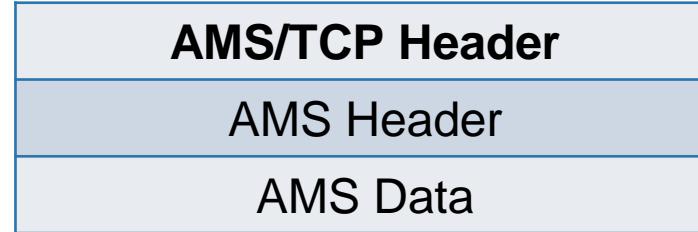


ADS支持多种协议

- 应用程序间TCP/IP通讯
- 基于Web的HTTP通讯
- 通过其他第三方协议（串口等）



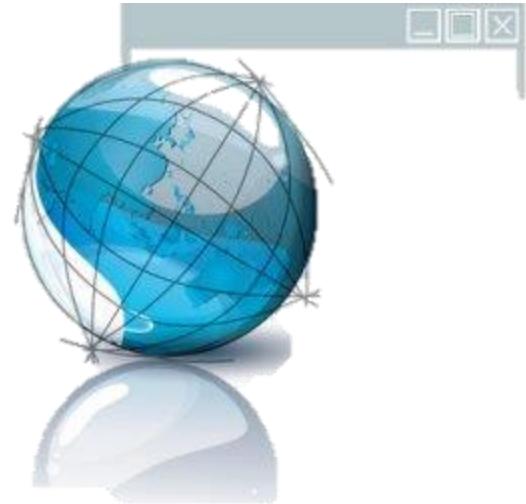
- 通讯端口48898
 - 基于TCP/IP协议
 - TwinCAT ADS Router (ADS Server) 监听端口48898用于等待新的客户端
- 数据包格式



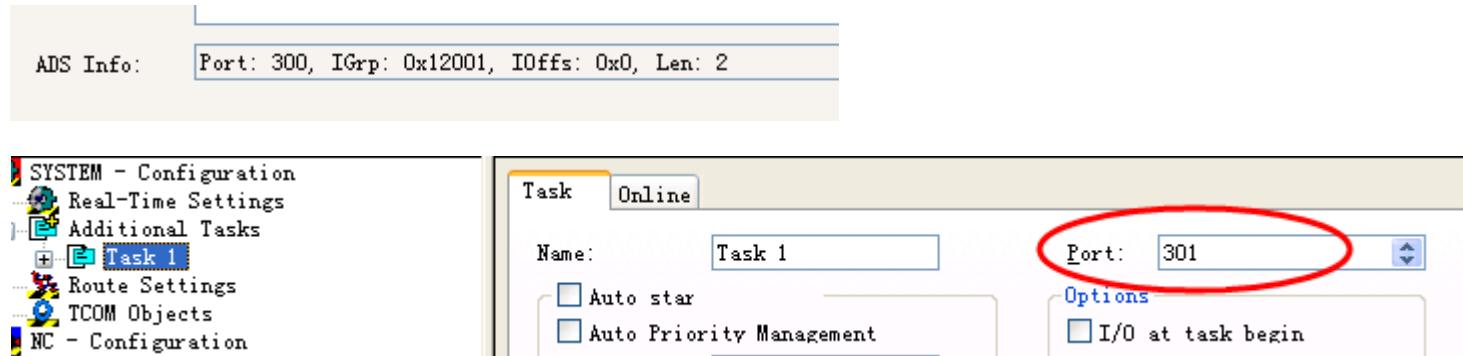
数据包	大小	描述
AMS/TCP Header	6 bytes	包含了ADS Data的长度。
AMS Header	32 bytes	此处包含了通讯的发送方和接收方地址，以及ADS错误代码、ADS命令代码和其他一些信息。
ADS Data	n bytes	此处包含了一个ADS命令的参数。参数的数据结构由ADS命令所决定，一些ADS命令也可以没有附加的数据。

- 通讯端口48899

- 基于UDP协议
- 主要用于广播的方式查找路由



- ADS设备唯一标识
 - AdsAmsNetId (NetId) : 用于确定设备硬件
 - AdsPortNr (AdsPort) : 用于确定软件服务



- 格式：xxx.xxx.xxx.xxx.xxx.xxx
- 例如：192.168.131.67.1.1
- AdsNetId作为TCP/IP地址的扩展

**注意： AdsNetId不是在IP
地址后加 .1.1**

- AdsNetId构成：
 1. 安装完TwinCAT后，第一次启动时的IP地址加.1.1
NetId在任何时候都不会自动改变。
 2. 部分BC9000， BCXXXX控制器在当前的IP地址后加.1.1

标识一个设备上的不同软件模块

- 通过端口号，在不同的ADS模块中传递数据

端口号固定，不可更改

- 各个ADS模块具有固定的端口号
- PLC：801，NC：500等

端口号	ADS设备描述	端口号	ADS设备描述
100	日志记录	801,811 821,831	PLC RuntimeSystem 1,2,3,4(TC2)
110	Eventlogger	851	PLC RuntimeSystem (TC3)
300	IO	900	Camshaft controller
301,302	additional Task 1, 2, ... (TC2)	10000	System Service
351,352	additional Task 1, 2, ... (TC3)	14000	Scope
500	NC		

- TwinCAT系统通过路由表记录远程设备信息

Current Routes			
Route	AmsNetId	Address	Type
BAC-KAICHEN	10.41.32.25.1.1	169.254.222.1	TCP/IP
CEM-800485	255.128.4.133.1.1	169.254.222.7	TCP/IP
CEM-810485	255.190.64.252.1.1	169.254.181.201	TCP/IP
CX-OC8C76	5.12.140.118.1.1	10.41.32.27	TCP/IP



ADS客户端
IP:192.168.1

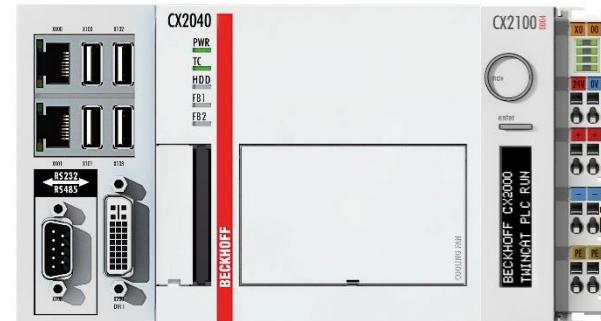
读取5.2.5.5.1.1:801
变量A

1.查表获取5.2.5.5.1.1对应通讯

2.向192.168.1.1发送读取ADS命令

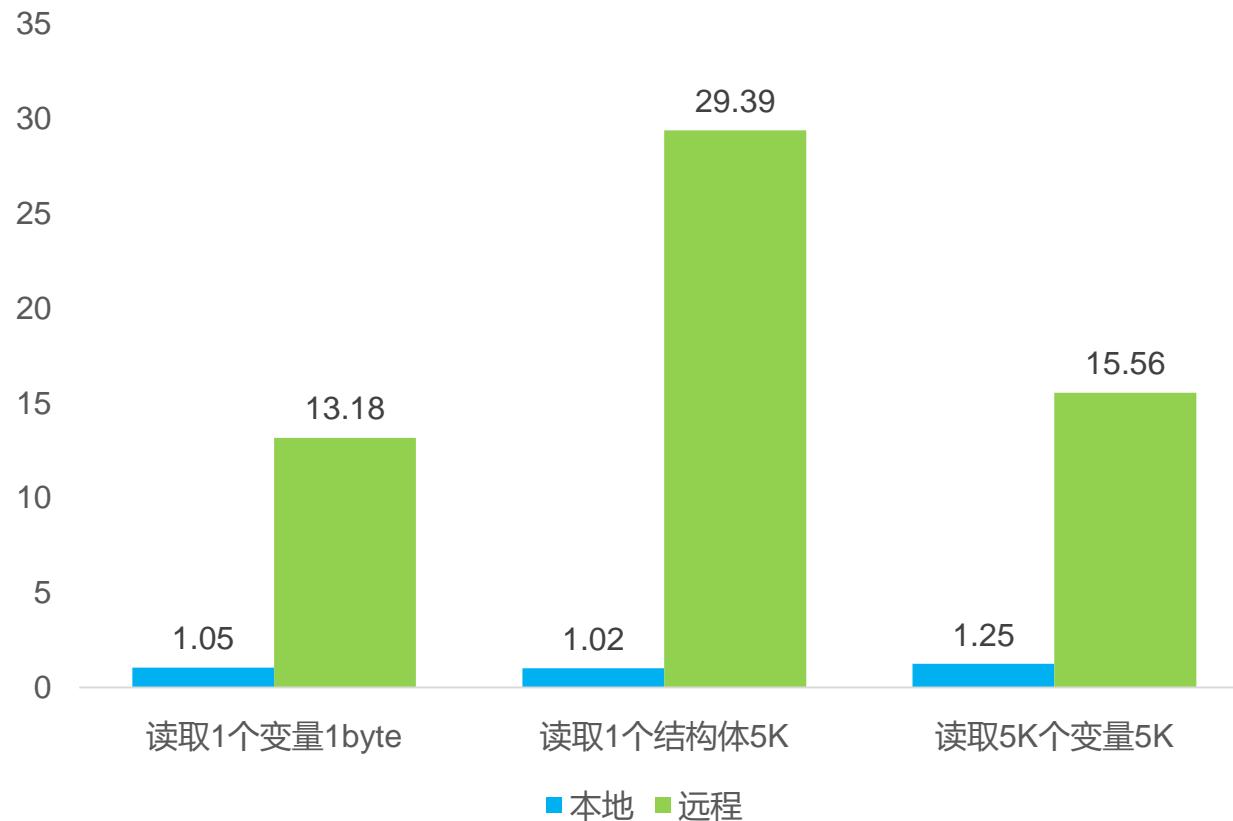
3.检查5.2.5.10.1.1及对应
IP192.168.1.2是否在路由表中

4.向192.168.1.2发送读取结果



ADS服务端
IP:192.168.1.1
NetId:5.2.5.5.
1.1

- Ads通讯不是一个实时的通讯协议，其通讯过程受到系统和网络状况的多重影响，无法保证一个稳定的通讯时间。

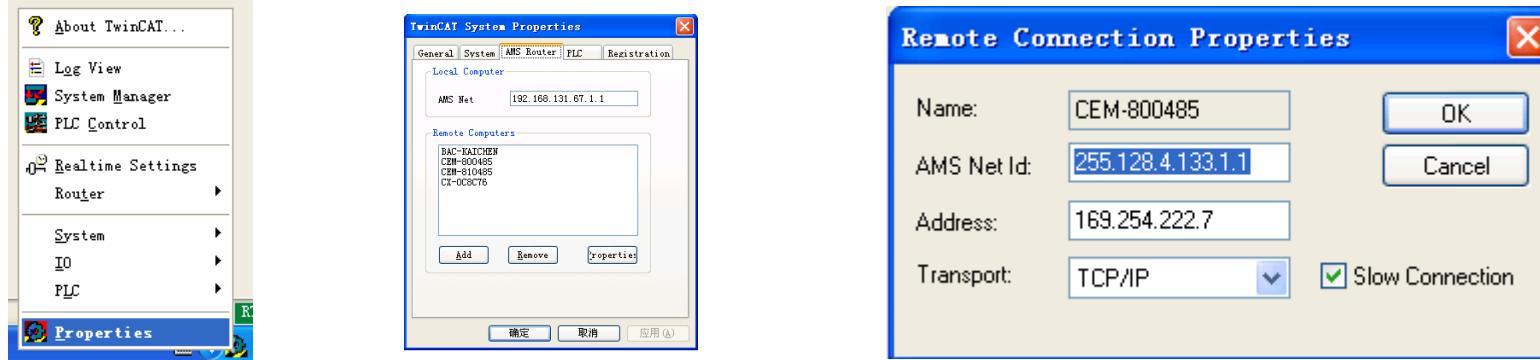


- 默认是5s
- 可以通过代码更改：
- C++
- C#

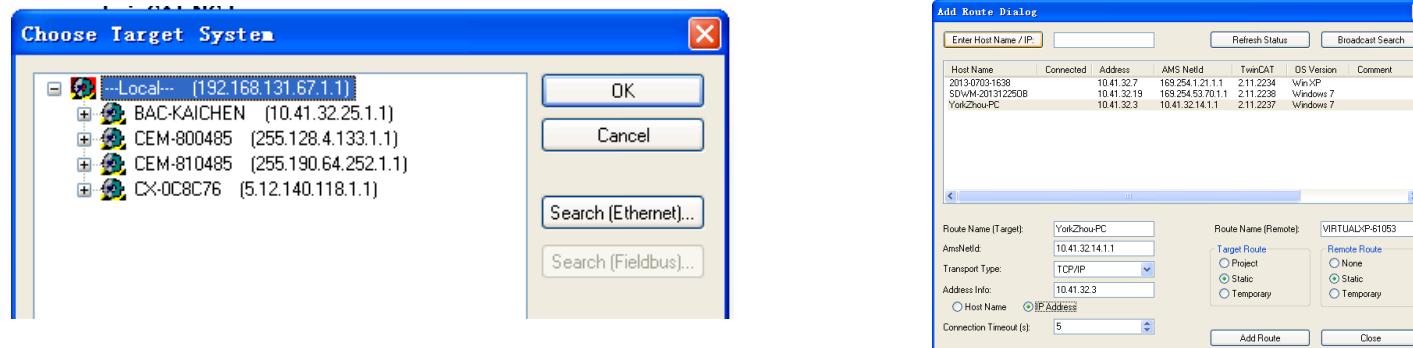
```
__declspec( dllexport )
long __stdcall AdsSyncSetTimeout( long nMs ); // Set timeout in ms
```

```
/// 摘要:
///     Sets the timeout for the ads communication. Unit is in ms.
public int Timeout { get; set; }
```

通过属性添加AmsNetId



通过搜索远程设备自动添加



进行ADS通讯

命令	描述
ADS Read Device Info	读取ADS设备的名称和版本号
ADS Read	可以从ADS设备中读取数据
ADS Write	可以对ADS设备写入数据
ADS Read State	读取ADS设备的ADS状态和设备状态
ADS Write Control	改变ADS设备的ADS状态和设备状态
ADS Add Device Notification	在ADS设备中创建一个通知
ADS Delete Device Notification	在ADS设备中删除一个通知
ADS Device Notification	数据将独立的从ADS设备发送到ADS客户端
ADS Read Write	可以对ADS设备写入数据，同时也能送ADS设备中读取数据

同步方式

- 同步读写
 - ADS 客户端向ADS 服务器发送ADS 请求，在通讯过程中客户端程序停止执行，直到获得 ADS 服务器返回的响应。
 - 能即时返回结果。

异步方式

- 异步读写
 - ADS 客户端向ADS 服务器发送ADS 请求，同时客户端继续自己的工作。ADS 服务器处理请求后，把响应以Call-back 函数方式发给客户端。
- 通知方式
 - ADS 客户端向ADS 服务器发送ADS 请求，ADS 服务器以Call-back 函数的方式不断向客户端发送响应，直到客户端取消该请求

同步方式(server)

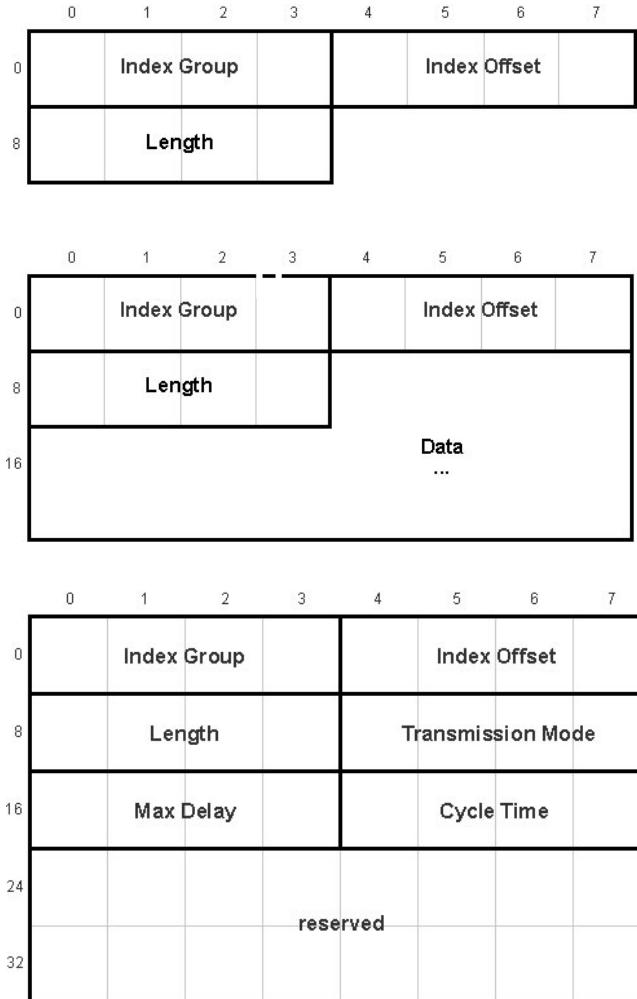
- 高级语言做server
 - 高级语言在twincat上注册一个port，然后由twincat调用ads来往这个port发数据
 - 由plc主动触发，而不需要进行轮训。
 - 很多倍福的功能块都是基于这种方式

- 异步通讯变量上限: **550**
- 批量读取变量上限: **500**
- 异步和批量读取需要通道支持

- 读写PLC变量
 - 按地址方式
 - 按变量名方式

GroupIndex值	描述
0x00004020	M寄存器（用%M声明）
0x00004021	MX寄存器（用MX声明）
0x0000F020	I寄存器（用%I声明）
0x0000F021	IX寄存器（用IX声明）
0x0000F030	Q寄存器（用%Q声明）
0x0000F031	QX寄存器（用QX声明）
0x0000F003	取变量的句柄
0x0000F005	通过变量句柄访问变量值
0x0000F006	释放句柄

- 执行其他命令
 - 获取系统时间、读写磁盘文件等



读写不同存储区

 M寄存器: 0x00004020 I寄存器: 0x0000F020 Q寄存器: 0x0000F030

查找PLC变量地址

 在PLC程序中显式指定

CloseBoard_Left_Place AT%IX0:BOOL;

查找其他变量地址

 通过System Manager查看ADS Info

Variable	Flags	Online
Name:	DevState	
Type:	UINT	
Group:	Inputs	Size: 2.0
Address:	0 (0x0)	User ID: 0
Linked to..		
Comment:	0x0001 = Link error 0x0002 = I/O locked after link error (I/O reset) 0x0010 = Out of send resources (I/O reset required) 0x0020 = Watchdog triggered 0x0040 = Ethernet driver (miniport) not found 0x0080 = I/O reset active	
ADS Info:	Port: 300, IGrp: 0x12001, IOffs: 0x0, Len: 2	

- 格式为% + 范围前缀 + 大小范围前缀 + 自然数组合（1到多个，空格分隔）

前缀	说明
I	输入
Q	输出
M	内存定位

- 大小范围前缀

前缀	说明
X	单个位
无	单个位
B	字节(8位)
W	字(16位)
D	双字(32位)

示例：

%QX75.1 %Q75 输出位 75

%IW215 输入字 215

%QB7 输出字节 7

%MD48 内存定位中的第 48 个双字的位置

创建句柄

- 通过变量名创建句柄
- 句柄为4Byte整形
- 作为索引值提高效率

通过句柄读写

- 通讯过程中使用句柄，不是用变量名

删除句柄

- 删除服务端句柄，释放资源

何时使用

- 大数据个数读写
- 周期性更新
- 延迟要求尽可能少

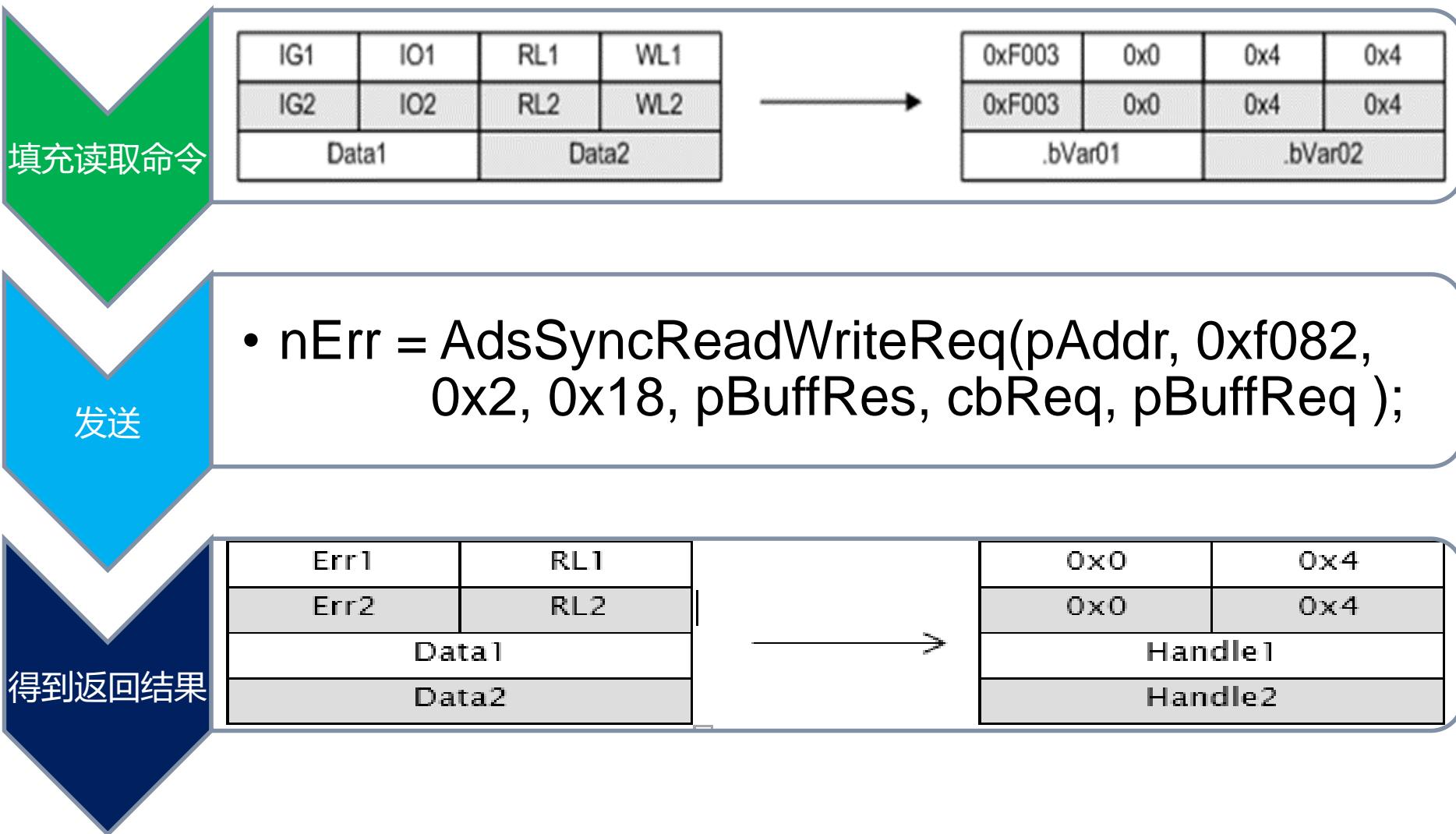
如何使用

- 通过ADS_READWRITE命令执行
 - 0xF080: ADSIGRP_SUMUP_READ 批量读取
 - 0xF081: ADSIGRP_SUMUP_WRITE 批量写入
 - 0xF082: ADSIGRP_SUMUP_READWRITE 批量读写

本质

- 通过特定的Index Group, 指定Data区的数据为ADS命令格式

ADS执行批量命令——例子：批量读取句柄



变量类型	C++	C#	长度
INT	short	short, INT16	2
UINT	unsigned short	unsigned short UINT16	2
DINT	int	int, INT32	4
UDINT	Unsigned int	UINT32	4
real	float	float	4
LREAL	double	double	8
string	char[81]	char[81]	81
bool	bool(小写)	bool	1

Microsoft Help Viewer 2.2 - Visual Studio Documentation

Contents Filter Contents

Help Viewer Home

- Beckhoff Information System
 - TwinCAT 3
 - Notes on the documentation
 - Overview
 - Licensing
 - TE1000 XAE
 - eXtended Automation Engineering
 - TC3 Workbench
 - Remote Manager
 - Source Control
 - System
 - Security Management
 - PLC
 - C/C++
 - Matlab/Simulink
 - Safety
 - I/O
 - Connectivity
 - EAP
 - ADS
 - Introduction
 - Reference
 - Samples
 - Appendix
 - ADS Return Codes
 - Diagnostic aid
 - Automation Interface
 - Remote Access
- TExxxx | TC3 Engineering
 - Engineering Functions
 - TE1111 | TC3 EtherCAT Simulation
 - TE1120 | TC3 XCAD Interface
 - TE13xx | TC3 ScopeView
 - TE132x | TC3 Bode Plot
 - TE1610 | EAP Configurator
- TCxxxx | TC3 Base
 - Overview
 - TC1000 | TC3 ADS
 - TC1100 | TC3 I/O
 - TC1200 | TC3 PLC
 - TC1210 | TC3 PLC/C++
 - TC1220 | TC3 PLC/C++/Matlab®/Simulink®
 - TC1250 | TC3 PLC/NC PTP 10

ADS Return Codes

Error codes: 0x000..., 0x500..., 0x700..., 0x1000..., 0x274C...

Global Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x0	0	no error		
0x1	1	Internal error		
0x2	2	No Rtime		
0x3	3	Allocation locked memory error		
0x4	4	Insert mailbox error	No ADS mailbox was available to process this message.	Reduce the number of ADS calls (e.g. ADS-Sum commands or Max Delay Parameter)
0x5	5	Wrong receive HMSG		
0x6	6	target port not found	ADS Server not started	
0x7	7	target machine not found	Missing ADS routes	
0x8	8	Unknown command ID		
0x9	9	Bad task ID		
0xA	10	No IO		
0xB	11	Unknown ADS command		
0xC	12	Win 32 error		
0xD	13	Port not connected		
0xE	14	Invalid ADS length		
0xF	15	Invalid ADS Net ID		
0x10	16	Low Installation level		
0x11	17	No debug available		
0x12	18	Port disabled		
0x13	19	Port already connected		
0x14	20	ADS Sync Win32 error		
0x15	21	ADS Sync Timeout		
0x16	22	ADS Sync AMS error		
0x17	23	ADS Sync no index map		
0x18	24	Invalid ADS port		
0x19	25	No memory		
0x1A	26	TCP send error		
0x1B	27	Host unreachable		
0x1C	28	Invalid AMS fragment		

Router Error Codes

Hex	Dec	Description	Possible Causes	Solution
0x1C	28	Invalid AMS fragment		

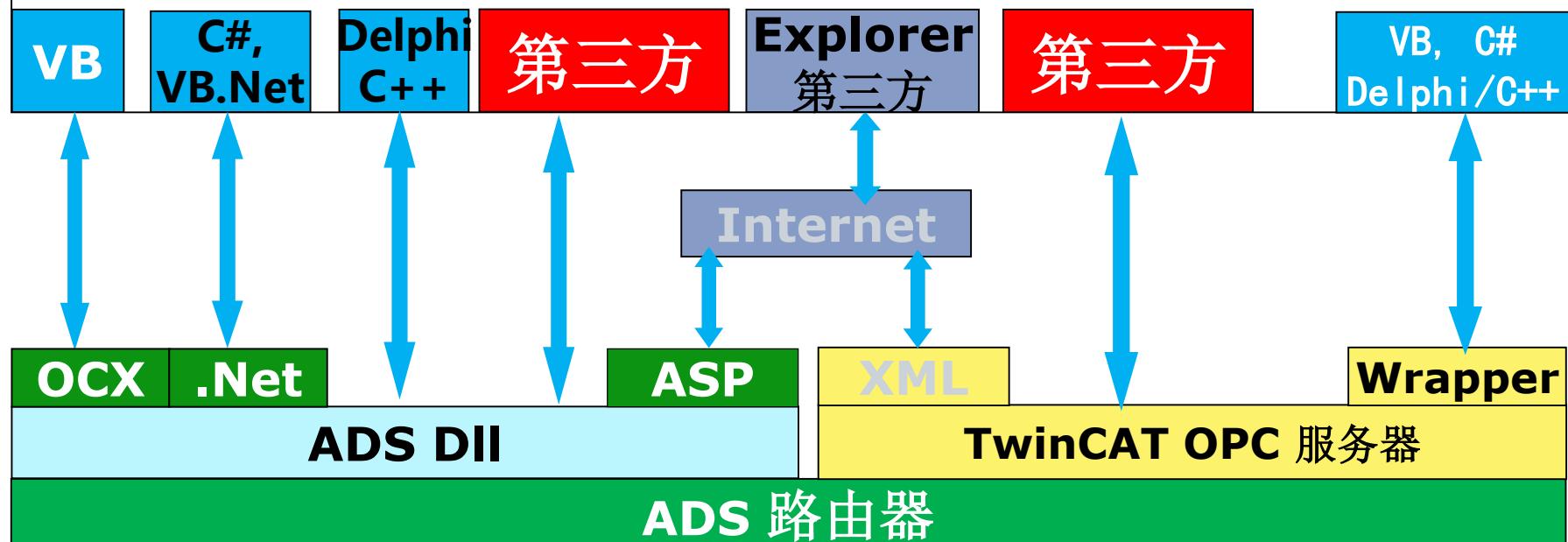
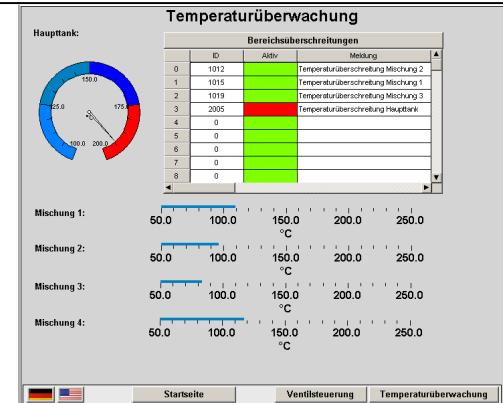
ADS客户端编程

Beckhoff ADS 接口

- 开放、简单、灵活、文档齐全
- 支持多家 **Scada** 软件

标准**OPC**接口

支持的规范: **DA/AE/XML**



为方便各种计算机高级语言及其环境使用ADS通讯， Beckhoff TwinCAT提供相如下组件：

- ADS .NET Component

 适用.NET平台， VB.NET,C#等

- ADS-OCX (ActiveX COM控件)

 适用于Visual Basic, Visual C++, Delphi等.

- ADS-DLL

 适用于Visual C++等.

- ADS-Script-DLL

 适用于 VBScript, Jscript等脚本， 创建B/S架构的应用.

- JAVA DLL

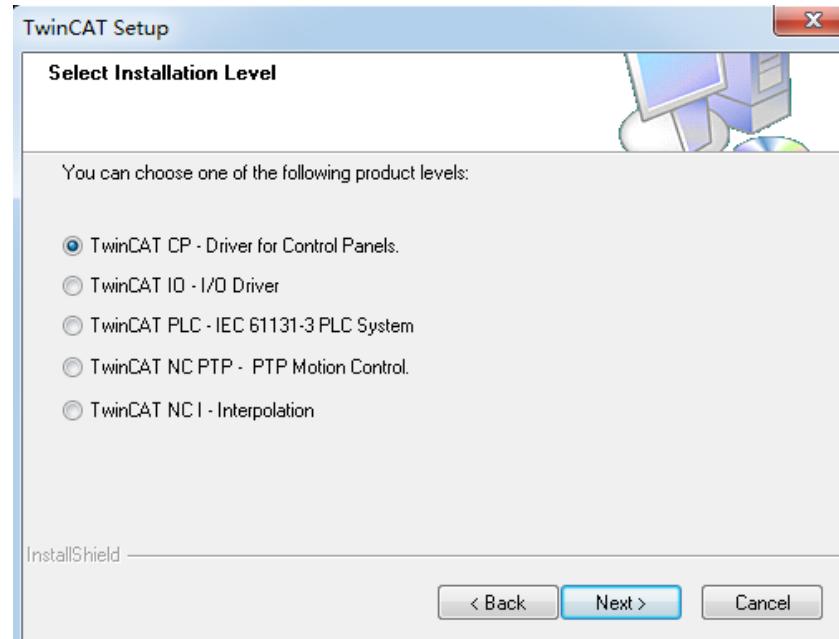
 适用于JAVA语言。

- “PlcSystem.lib” PLC 库

ADS设备可以通过据此和其他ADS设备以及Windows进行通讯。

ADS通讯组件使用要求

- 需要twincat平台的支持
- 可选择安装twincat2的cp版



- Twincat3的 通讯组件

 TC31-ADS-Setup.3.1.4022.16.exe 2018-02-28 22:25 应用程序 129,844 KB

- 经测试， 两者可以通用

- C++
- 有支持多线程的一组函数

◀ Extended Functions

Extended functions (for multithreaded applications)

AdsPortOpenEx
AdsPortCloseEx
AdsGetLocalAddressEx
AdsSyncWriteReqEx
AdsSyncReadReqEx2
AdsSyncReadWriteReqEx2
AdsSyncReadDeviceInfoReqEx
AdsSyncWriteControlReqEx
AdsSyncReadStateReqEx
AdsSyncAddDeviceNotificationReqEx
AdsSyncDelDeviceNotificationReqEx
AdsSyncSetTimeoutEx
AdsSyncGetTimeoutEx
AdsAmsPortEnabledEx

- C#
- Public static (Shared in Visual Basic) members of this type are safe for multithreaded operations. Instance members are not guaranteed to be thread-safe.
- 建议加锁

TcAdsDII.dll

在 Visual C++ 中进行链接和程序开发时所需要的文件：

TcAdsDII.dll – 动态链接库

TcAdsDII.lib – 使用 TcAdsDII.dll 的函数库

TcAdsApi.h – 声明 ADS 功能的头文件

TcAdsDef.h – 声明结构和常数的头文件

主要接口：

AdsPortOpen: 建立 TwinCAT 信息路由器连接 (通讯口)。

AdsPortClose: 关闭 TwinCAT 信息路由器的连接(通讯口)。

AdsGetLocalAddress: 返回本地 NetId 和端口号

AdsSyncWriteReq: 同步写数据到一个 ADS 设备

AdsSyncReadReq: 从 ADS 服务器中同步读数据。

AdsSyncReadWriteReq: 同步写入数据到 ADS 服务器中并从 ADS 设备接收返回的数据。

AdsSyncReadStateReq: 从 ADS 服务器中读 ADS 状态和设备状态。

AdsSyncAddDeviceNotificationReq: 通知在 ADS 服务器中进行定义 (如 PLC)。当某个事件发生时，该功能 (回调功能) 在ADS 客户端进行调用 (C 程序)。

AdsSyncDelDeviceNotificationReq: 从 ADS 服务器中删除以前定义的通知。

函数名	描述	函数名	描述
AddDeviceNotification	连接一个变量到ADS客户端	ReadDeviceInfo	读取ADS服务器的版本号
AddDeviceNotificationEx	连接一个变量到ADS客户端	ReadState	读取ADS服务器的ADS状态和设备状态
Connect	建立一个至ADS服务的连接	ReadSymbol	读取一个变量的值，并返回object类型
CreateSymbolInfoLoader	创建一个新的SymbolInfoLoader类	ReadSymbolInfo	获取一个变量的信息
CreateVariableHandle	生成一个ADS变量的唯一句柄	ReadWrite	将数据写入ADS服务器并读取数据
DeleteDeviceNotification	删除设备通知	Write	写入数据到ADS服务器
DeleteVariableHandle	释放一个ADS变量句柄	WriteAny	写入数据到ADS服务器
Read	从ADS服务器读取数据	WriteControl	改变ADS服务器的ADS状态和设备状态
ReadAny	从ADS服务器读取数据	WriteSymbol	写入一个变量的值

- 地址
- <https://github.com/Beckhoff/ADS>

支持多个平台:

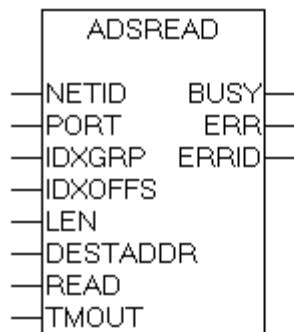
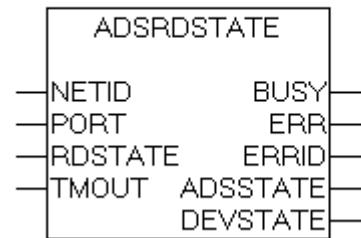
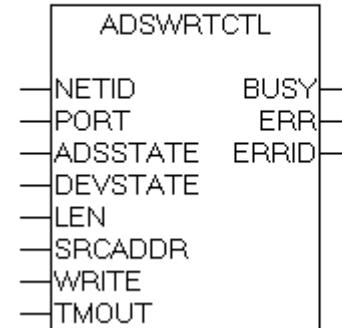
FreeBSD 11.1 amd64	clang 4.0.0
macOS 10.13.3	clang 900.0.39.2
Ubuntu 16.04 LTS x86_64	gcc 6.3.0
Win7 64-bit	gcc 6.4.0 + cygwin64
Win7 64-bit	gcc 5.4.0 + cygwin32
Win7 64-bit	MS Visual C++ 2015 (untested)

基于C++11编写

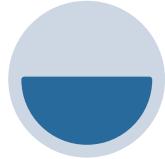
不需要twincat底层支持

由德国总部开放源码

功能块名称	描述
ADSREAD	通过ADS读取数据
ADSWRITE	通过ADS写入数据
ADSRDWRT	通过ADS读写数据
ADSRDSTATE	通过ADS读取ADS服务器的ADS状态和设备状态
ADSWRTCTL	通过ADS改变ADS服务器的ADS状态和设备状态
ADSRDDEVINFO	通过ADS读取ADS服务器的版本号
ADSREADEX	通过ADS读取数据并返回读取的字节数
ADSRDWRTEX	通过ADS写入数据并返回读取的字节数

FUNCTION_BLOCK ADSREAD**FUNCTION_BLOCK ADSRDSTATE****FUNCTION_BLOCK ADSWRTCTL**

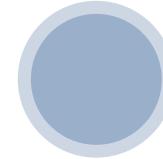
ADS组件库集成在TwinCAT软件中，安装任何版本的TwinCAT软件都包含ADS通讯组件



TwinCAT2

TS6000-0000

TwinCAT_ADS_Communication_Library



TwinCAT3

TC31-ADS-Setup

文件	描述
C:\TwinCAT\ADSApi\TcAdsDII\	ADS DLL
C:\TwinCAT\ADSApi\.NET	.NET组件 (支持三个版本的Framework)
C:\TwinCAT\ADSApi\AdsToJava	ADS JAVA DLL
C:\TwinCAT\ADSApi\CompactFramework	.NET Compact Framework组件
C:\TwinCAT\ADSApi\TcAdsWebService	WebService组件
C:\WINDOWS\system32\AdsOcx.ocx	ADS OCX
C:\TwinCAT\TcScript.dll	ADS-Script-DLL

ADS故障诊断

- TwinCAT ADS Monitor

通讯数据帧 记录

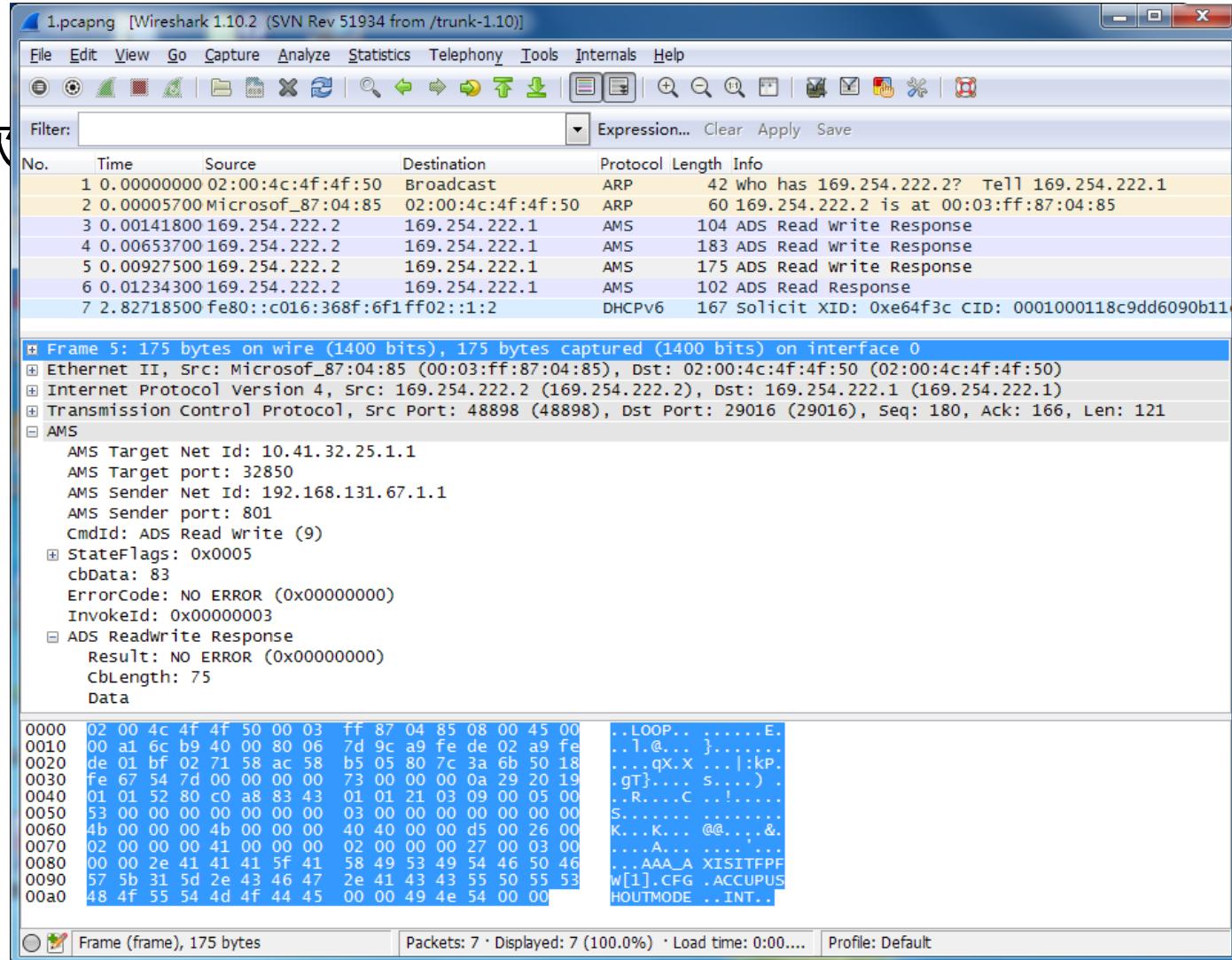
No.	TimeStamp	CmdId	Sender	Target	Error	hUser	cbData	Data
1	2008-6-19 17:07:52 147 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0x9	12	IG 0x1 IO 0x6 Len 4
2	2008-6-19 17:07:52 147 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0x9	12	Res 0x0, Len 4 + 00 00 00 00
3	2008-6-19 17:07:52 149 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xA	12	IG 0x1 IO 0x7 Len 4
4	2008-6-19 17:07:52 149 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xA	12	Res 0x0, Len 4 + 50 00 00 00
5	2008-6-19 17:07:52 156 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xB	12	IG 0x1 IO 0x6 Len 4
6	2008-6-19 17:07:52 156 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xB	12	Res 0x0, Len 4 + 00 00 00 00
7	2008-6-19 17:07:52 158 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xC	12	IG 0x1 IO 0x7 Len 4
8	2008-6-19 17:07:52 158 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xC	12	Res 0x0, Len 4 + 50 00 00 00
9	2008-6-19 17:07:52 164 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xD	12	IG 0x1 IO 0x6 Len 4
10	2008-6-19 17:07:52 164 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xD	12	Res 0x0, Len 4 + 00 00 00 00
11	2008-6-19 17:07:52 167 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xE	12	IG 0x1 IO 0x7 Len 4
12	2008-6-19 17:07:52 167 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xE	12	Res 0x0, Len 4 + 50 00 00 00
13	2008-6-19 17:07:52 208 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0xF	12	IG 0x1 IO 0x6 Len 4
14	2008-6-19 17:07:52 208 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0xF	12	Res 0x0, Len 4 + 00 00 00 00
15	2008-6-19 17:07:52 211 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0x10	12	IG 0x1 IO 0x7 Len 4
16	2008-6-19 17:07:52 211 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0x10	12	Res 0x0, Len 4 + 50 00 00 00
17	2008-6-19 17:07:52 273 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0x11	12	IG 0x1 IO 0x6 Len 4
18	2008-6-19 17:07:52 273 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0x11	12	Res 0x0, Len 4 + 00 00 00 00
19	2008-6-19 17:07:52 275 ms	R_Req	192.168.0.37.1.1 (32801)	192.168.0.37.1.1 (200)	0	0x12	12	IG 0x1 IO 0x7 Len 4
20	2008-6-19 17:07:52 275 ms	R_Res	192.168.0.37.1.1 (200)	192.168.0.37.1.1 (32801)	0	0x12	12	Res 0x0, Len 4 + 50 00 00 00

ADS数据结构

帧数据

Ready VTheme

- Wireshark
- 网络封包分析软

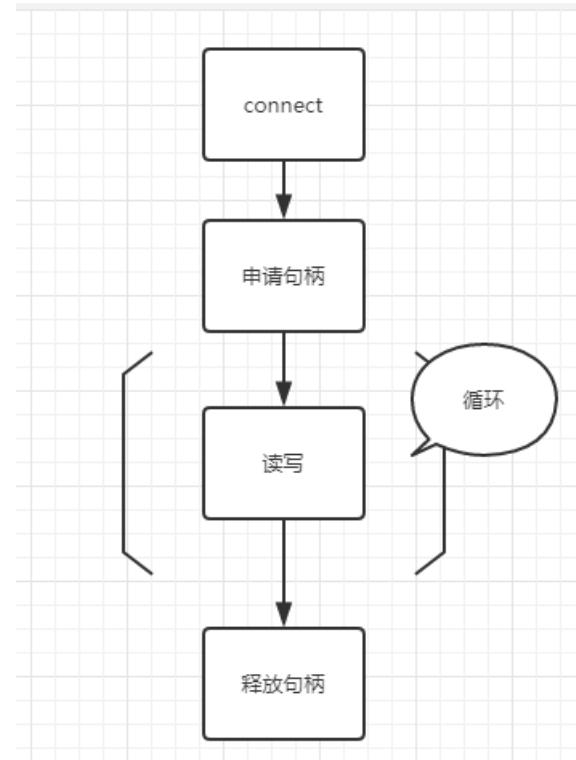


ADS错误代码	描述	建议
0x745(1861)	通讯超时	常见故障，检测连线、路由设置、防火墙设置、服务端TwinCAT是否已经启动
0x710(1808)	对象没有找到	检测变量名称是否正确，注意全局变量为： .变量名
0x705(1797)	参数长度错误	常见于批处理，检查计算命令长度
0x707(1799)	设备未就绪	可能为tsm配置不正确，重新激活配置

- ip地址是否在一个网段
- Netid是否一样
- 是否ping的通
- 是否添加路由
- 确认下Tc3和tc2，确认下port号
- 确认下变量名(tc3和tc2有区别)
- 看error code

故障诊断流程(通讯变慢)

- 确实是通讯变慢，而不是客户代码问题
- 查看网络ping的时间
- 确认下客户的代码逻辑



- 这个问题一般是疑难杂症
- 代码关键部分加log
- 使用工具长时间测试网络的状态
- Ping工具的技巧 -t `ping www.baidu.com -w 1000`

- 问题一般是：ads支持xxx语言么？（包括但不限于python, node.js, php等）
 - 理论上：如果这门语言可以实现tcp，就能支持ads
 - 但是，我们一般没有对这些语言提供专门的ads库，需要自己基于tcp来实现
-
- 网上有一些第三方的ads库，可以作为参考。
 - 比如pyhon和node.js都有第三方的ads库

- 是否可以不安装twincat走ads
- 建议安装twincat通讯组件

- 如果坚持不安装
- 编译开源版本ads库并调用，理论上可以不安装twincat
- 客户水平较高时可以使用此方法

如何判断ads函数执行成功

1. C++

看返回值

2. C#

用try...catch...

返回值 (tryread, trywrite)

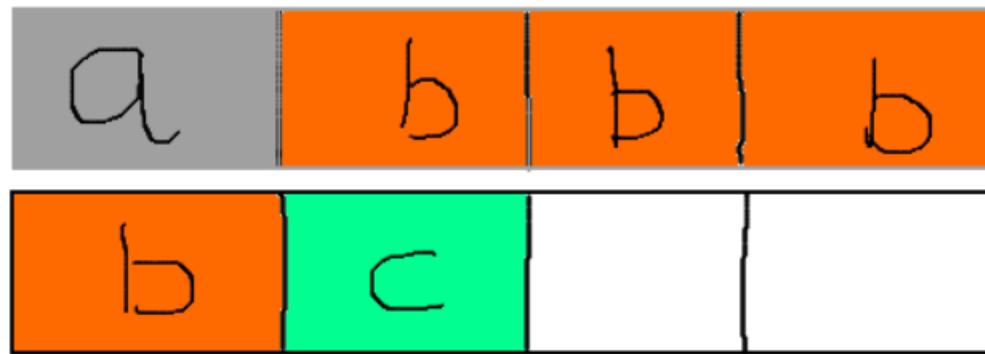
Twincat2和Twincat3 ads的区别

- 1, 端口不同
- 2, 默认数据的对齐方式不同
- 3, 默认全局变量的变量名格式不同

对齐方式计算方法：

- 是提高内存访问速度的策略，cpu在访问未对齐的内存需要经过两次内存访问，而经过内存对齐一次就可以了

```
typedef struct test{  
    char a;  
    int b;  
    char c;  
};
```



- 结构体或union联合的数据成员，第一个数据成员是要放在offset == 0的地方，以后每个数据成员存储的起始位置要从该成员大小或者成员的子成员大小的**整数倍**开始。
- 如果结构体作为成员，则要找到这个结构体中的最大元素，然后从这个最大成员的**整数倍**地址开始存储（struc a中有一个struct b， b里面有char, int, double...那b应该从8的整数倍开始存储）
- 结构体的总大小，必须要是其内部最大成员的**整数倍**，不足的要补齐

```
1 typedef struct bb
2 {
3     int id;
4     double weight;
5     float height;
6 }BB;
7
8 typedef struct aa
9 {
10    char name[2];
11    int id;
12
13    double score;
14    short grade;
15    BB b;
16 }AA;
```

- 如果设置的对齐数小于取整数倍的值，则以对齐数为标准计算整数倍

```
1  typedef struct bb
2  {
3      int id;          // [0]....[3]
4      double weight;   // [8]....[15]           原则 1
5      float height;    // [16]...[19],总长要为8的整数倍,补齐[20]...[23]       原则 3
6  }BB;
7
8  typedef struct aa
9  {
10     char name[2];    // [0],[1]
11     int id;          // [4]...[7]           原则 1
12
13     double score;    // [8]....[15]
14     short grade;    // [16],[17]
15     BB b;           // [24].....[47]        原则 2
16 }AA;
```