

# 基于C#的上位机与台达运动控制器通信实现

李小泉 杨科

中达电通股份有限公司

DOI:10.12238/acair.v3i1.11876

**[摘要]** 本文探讨了如何使用C#编程语言实现上位机与台达运动控制器之间的高效通信。详细阐述了通信协议的选择、通信接口的设计以及数据处理和错误处理机制。通过实际案例和测试结果,证明了所提出方法的有效性和可靠性,为工业自动化领域中的运动控制提供了有价值的技术参考。

**[关键词]** C#; 上位机; 运动控制器; 通信

**中图分类号:** TM571 **文献标识码:** A

## Communication between upper computer and Delta motion controller based on C #

Xiaoquan Li Ke Yang

Zhongda Dentsu Co., LTD.

**[Abstract]** This paper discusses how to use C # programming language to achieve efficient communication between upper computer and Delta motion controller. The selection of communication protocol, design of communication interface and data processing and error processing mechanism are detailed. The effectiveness and reliability of the proposed method are proved through practical case and test results, which provide a valuable technical reference for motion control in the field of industrial automation.

**[Key words]** C #; upper computer; motion controller; communication

### 引言

在现代工业自动化系统中,人机交互方式除了触摸屏,按钮等方式外,上位机是利用其强大的运算,数据处理功能,广泛应用于数据采集,人机交互,报警通信等。上位机与运动控制器之间的稳定通信对于实现精确的运动控制至关重要。C#作为一种强大的编程语言,具有丰富的类库和良好的跨平台特性,为实现这种通信提供了便捷的手段。除了用组态软件之外,本文介绍通过C#语言实现数据通信与可视化的方法。

台达电子机电事业近年推出了很多优秀的运动控制器,知名的自动化领域企业,其AX系列motion广泛应用在能源,食品,制药,包装等领域。本文介绍AX8系列MOTION与上位机通信为背景,实现数据采集与管理的方法。

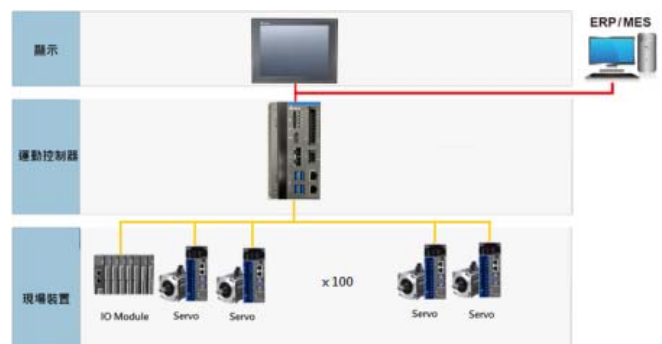
### 1 通信协议选择

常见的通信协议如串行通信(RS232/RS485)、以太网通信等各有优缺点。通信方式包括了API函数,OPCUA, ModbusTCP等多种方式。考虑到数据传输速率和稳定性要求,本文选择以太网通信协议作为上位机与运动控制器之间的通信方式,并采用TCP/IP协议进行数据传输。

### 2 通信接口设计

台达运动控制器AX8H15系列运动控制器,使用Intel X86 i3系列CPU,本体集成2个EtherCAT总线接口,2个独立IP的以太网

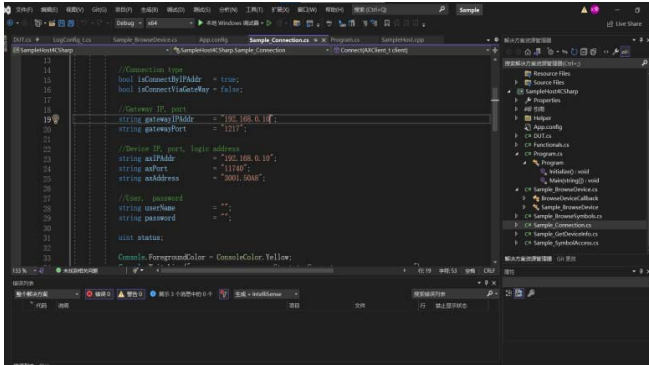
口,数据存贮容量达1Gb,在高速运控指令处理达0.4ns,从站数量可以512站,支援128轴同步运动控制,最快控制周期0.25ms,内建EtherCAT、EtherNet/IP、Modbus TCP/IP、OPCUA、Profinet (Device)、Socket 32/16通讯协定提供8点的高速数位输入,4点高速数位输出,1个RS485串口。适合中大等规模产线,高端制造的核心控制器。



IA Communication SDK是台达推出的上位机实现与motion数据交换的数据连接方式,在C#中,可以方便使用,来实现网络通信。实现客户的类似MES,组态画面,数据监控与报警等。

SDK支持与AX系列处理器建立连接和断开连接,通过符号名发起异步写入请求,并通过回调函数获取写入操作的执行结果

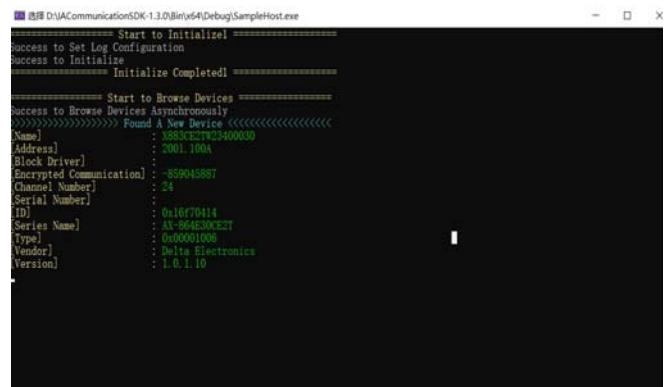
等功能, 支援多种数据格式的访问。x64 SDK的动态链接库 AXDriver.dll, IACommService.dll和IACommService4CSharp.dll 复制到可执行文件的同级目录即可。



建立一个通信链接的代码:

```
public static uint Connect(AXClient_t client)
{
    if (null == client)
        return ErrorCode.InvalidArgument;
    //Connection type
    bool isConnectByIPAddr = true;
    bool isConnectViaGateWay = false;
    //Gateway IP, port
    string gatewayIPAddr = "192.168.0.10"; //AXmotion的默认port1 IP地址
    string gatewayPort = "1217";
    //Device IP, port, logic address
    string axIPAddr = "192.168.0.10";
    string axPort = "11740";
    string axAddress = "3001.50A8";
    //User, password
    string userName = "";
    string password = "";
    uint status;
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Start to Connect =====");
    if (isConnectByIPAddr)
    {
        if (isConnectViaGateWay)
            status = client.ConnectViaGateway(gatewayIPAddr, gatewayPort, axIPAddr, axPort, userName, password);
        else
            status = client.Connect(axIPAddr, axPort, userName, password);
    }
    else
    {
        status = client.ConnectViaGateway(gatewayIPAddr, gatewayPort, axIPAddr, axPort, userName, password);
    }
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Connect Completed =====\n");
    return status;
}
//以下为断开链接:
public static void Disconnect(AXClient_t client)
{
    if (null == client)
        return;
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Start to Disconnect =====");
    var status = client.Disconnect();
    MessageLog.PrintOperationResult("Disconnect", status);
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Disconnect Completed =====\n");
}
运行测试:
```

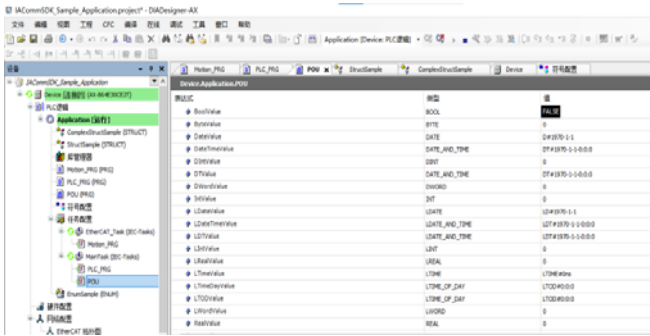
```
{
    if (isConnectViaGateWay)
        status = client.ConnectViaGatewayByAddress(gatewayIPAddr, gatewayPort, axAddress, userName, password);
    else
        status = client.ConnectByAddress(axAddress, userName, password);
}
MessageLog.PrintOperationResult("Connect", status);
Console.ForegroundColor = ConsoleColor.Yellow;
Console.WriteLine("===== Connect Completed =====\n");
return status;
}
//以下为断开链接:
public static void Disconnect(AXClient_t client)
{
    if (null == client)
        return;
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Start to Disconnect =====");
    var status = client.Disconnect();
    MessageLog.PrintOperationResult("Disconnect", status);
    Console.ForegroundColor = ConsoleColor.Yellow;
    Console.WriteLine("===== Disconnect Completed =====\n");
}
运行测试:
```



### 3 数据处理

接收到的数据需要进行解析和处理, 以提取出有用的运动控制信息。同时, 上位机发送给运动控制器的数据也需要按照特定的格式进行封装。

Codesys平台软件查看写入的数据情况:



举例枚举数组的访问:

```
//Create a Enum Array value
EnumSample[] value = new EnumSample[]
{EnumSample.SecondMember, EnumSample.FirstMember };
// Create a DataItem_t instance to write data by the
Enum Array value
DataItem_t dataItem = new DataItem_t((object)value);
//Set the Enum Array value to the DataValue_t instance
in DataItem_t dataItem.Value.Set(value);
//Get the Enum Array value from the DataValue_t
instance in DataItem_t
value = dataItem.Value.Get(typeof(EnumSample[])) as
EnumSample[];
//Create a DataValue_t instance by the Enum Array
value
DataValue_t dataValue = new DataValue_t(value);
//Set the Enum Array value to the DataValue_t instance
dataValue.Set(value);
```

```
// Get the Enum Array value from the DataValue_t
instance
```

```
value=dataValue.Get(typeof(EnumSample))asEnumSampl
e[]
```

#### 4 错误处理机制

在通信过程中,可能会出现网络连接中断、数据传输错误等异常情况。因此,需要建立完善的错误处理机制,及时捕获并处理这些异常,保证系统的稳定性。

#### 5 实验结果与分析

通过实际的运动控制实验,对上位机与运动控制器之间的通信性能进行测试。结果表明,通信延迟低、数据准确性高,能够满足运动控制的实时性要求。

#### 6 结论

本文提出了一种基于C#的上位机与运动控制器通信的实现方法,通过合理选择通信协议、设计通信接口、进行数据处理和建立错误处理机制,实现了稳定高效的通信。该方法在工业自动化领域,尤其数据通信层MES, SCADA等具有广泛的应用前景。

#### [参考文献]

- [1]王小科.C#开发宝典[M].北京:机械工业出版社,2012.
- [2]马立新,康存锋.CoDeSys V3基础编程入门[G].德国3S软件有限公司,2022.
- [3]翁荣城.运动控制技术在港口无人集卡上的应用[J].自动化应用,2025,66(01):31-33.
- [4]台达电子,AX-8系列操作手册,文件编号:AX-8-0389400-3.

#### 作者简介:

李小泉(1974--),男,汉族,浙江杭州人,本科,高级工程师,研究方向:智能制造与自动控制技术。