|  |  |  |
| --- | --- | --- |
| 作者： | 王子鹏 | 上海市江场三路市北工业园区163号5楼（200436）TEL: 021-66312666FAX: 021-66315696 |
| 日期： | 2018-7-10 |
| 版本： | V1.1 |
| E\_mail: | Support@beckhoff.com.cn |

|  |
| --- |
| **使用Microsoft Visual C++实现对TwinCAT3 PLC状态的读取和控制** |

#

|  |
| --- |
| 概 述 |

|  |  |
| --- | --- |
| 使用Microsoft Visual C++ 软件，实现对TwinCAT3 PLC状态的查询，并控制PLC的启动/停止本次测试所用的TwinCAT3版本为4022.22，用到的VisualStudio版本为2017 Community。 |  |

|  |
| --- |
| 文档中包含的文件 |

文件名称 文件说明

|  |  |
| --- | --- |
| TcAdsDll.lib | 为了使用 TcAdsDll.dll所要联接的库 |
| TcAdsDef.h | 结构体和常量的声明 |
| TcAdsApi.h | ADS 函数的声明 |
| C++ | C++程序源文件 |
| PLC | PLC源文件 |

|  |
| --- |
| 备 注 |

|  |
| --- |
| 关键字：高级语言程序对PLC程序的读和写 |

|  |
| --- |
| 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，也欢迎您提出改进建议。*文档内容可能随时更新**如有改动，恕不事先通知* |

**通过C++程序读取/控制PLC的状态**

**（PLC状态读取）编写PLC程序**

1. 打开TwinCAT3，新建一个TwinCat Project，完成后鼠标右键点击左菜单栏中的PLC选项选择Add New Item，选择Standard PLC Project建立标准PLC项目

1. 展开左菜单栏中Project->POUs选项，双击选择编辑MAIN(PRG)，完成后效果如下

3.在变量声明区定义一个名称为BOOL1的BOOL型变量，并赋予其地址“AT%MX0.0”，在程序代码区中输入一个“；”

4.编写完程序后，我们BUILD程序并激活，LOG IN建立851端口通讯，并启动PLC程序。

**（PLC状态读取）编写C++程序**

5.首先我们启动Visual Studio 2017，新建一个项目。点击窗口左上方菜单栏的文件->新建->项目。出现窗口如下图所示。我们选择Visual C++中的Windows控制台应用程序选项。输入项目名称（我们这里起名为：Samples）并选择为解决方案创建目录。点击确定。

6.此时我们需要添加链接库，以便实现通讯中用到的各种函数的直接调用。我们在左侧找到我们所建立的项目处，点击右键->属性。效果如下图。

7.选择配置属性->链接器->常规->附加库目录->编辑，进入附加包含目录的编辑界面。如下图所示，添加我们的程序的所在目录（C:\TwinCAT\AdsApi\TcAdsDll\Lib），点击确定。（本路径是以安装TwinCAT3时的默认路径配置的，若用户在安装时选择自定义路径，则请用户自行修改）

8.再选择链接器－>输入－>附加依赖项，点击右边的按钮选择编辑进入附加依赖项界面。如下图所示添加我们的.lib文件的文件名（TcAdsDll.lib），点击确定。再次点击确定，链接库添加完成。

9.此时我们还需要将头文件加入，点击解决方案下的Samples－> 头文件－> stdafx.h，双击打开。向其中添加头件，如下图所示（tcAdsDef.h 和TcAdsApi.h ）。因为头文件没有复制到程序目录，所以在前面加上了头文件所在地址。接着我们再次向程序添加头文件（iostream、windows.h、conio.h）。效果如图：

代码：

//添加头文件

#include <iostream> //标准的输入输出流头文件

#include <conio.h> //控制台输入输出头文件

#include <windows.h> //windowsAPI的函数声明和宏

#include "C:\TwinCAT\ADSApi\TcAdsDll\Include\TcAdsDef.h" //结构体和常量的声明

#include "C:\TwinCAT\ADSApi\TcAdsDll\Include\TcAdsAPI.h" // ADS函数的声明

10.接下来我们开始编写主程序

代码：

#include "stdafx.h"

using namespace std;

void main()

{

 USHORT nAdsState; //包含PLC的状态信息

 USHORT nDeviceState;

 long nErr, nPort;

 AmsAddr Addr;

 PAmsAddr pAddr = &Addr;

 //建立端口通讯

 nPort = AdsPortOpen();

 nErr = AdsGetLocalAddress(pAddr);

 if (nErr) cerr << "Error: AdsGetLocalAddress: " << nErr << '\n';

 pAddr->port = 851;

 do

 {

 //向PLC读取PLC的状态信息

 nErr = AdsSyncReadStateReq(pAddr, &nAdsState, &nDeviceState);

 if (nErr)

 cerr << "Error: AdsSyncReadStateReq: " << nErr << '\n';

 else

 {

 cout << "PLCState: " << nAdsState << '\n'; // 输出PLC状态信息

 }

 cout.flush();

 } while (getchar() == '\n'); // 在“回车”时实现循环读取

 //关闭通讯端口

 nErr = AdsPortClose();

 if (nErr) cerr << "Error: AdsPortClose: " << nErr << '\n';

}

11.编写程序并调试，程序成功读取PLC的状态（注意：返回数字为5时PLC在启动状态，返回数字为6时PLC在停止状态）。如下图。

12.此时PLC为启动状态，在控制台上显示为“5”。当我们将TC3中的PLC停止，在控制台上按下“回车”，值就变换为“6”了，图示：

这样我们向TC3读取PLC状态就完成了。

下面我们介绍如何通过C++控制PLC的状态。

**（PLC状态控制）编写PLC程序**

13.打开TwinCAT3

14.编写程序并运行（与读取部分相同）

**（PLC状态控制）编写C++程序**

15.打开Visual Studio 2017，创建一个控制台项目，并且为项目配置好所需链接库。

16.编写程序

代码：

#include "stdafx.h"

using namespace std;

void main()

{

 USHORT nAdsState; //包含PLC的信息

 USHORT nDeviceState = 0;

 long nErr, nPort;

 char ch;

 void \*pData = NULL;

 AmsAddr Addr;

 PAmsAddr pAddr = &Addr;

 //建立接口通讯连接

 nPort = AdsPortOpen();

 nErr = AdsGetLocalAddress(pAddr);

 if (nErr) cerr << "Error: AdsGetLocalAddress: " << nErr << '\n';

 pAddr->port = 851;

 //输入控制命令

 cout << "(R) -> PLC Run\n"; //输入'R'启动PLC

 cout << "(S) -> PLC Stop\n"; //输入'S'停止PLC

 cout.flush();

 ch = getchar();

 ch = toupper(ch);

 //根据输入指令改变PLC的状态

 while ((ch == 'R') || (ch == 'S')) //实现循环控制

 {

 switch (ch)

 {

 case 'R':

 nAdsState = ADSSTATE\_RUN;

 break;

 case 'S':

 nAdsState = ADSSTATE\_STOP;

 break;

 }

 nErr = AdsSyncWriteControlReq(pAddr, nAdsState, nDeviceState, 0, pData);

 if (nErr) cerr << "Error: AdsSyncWriteControlReq: " << nErr << '\n';

 cin.ignore();

 ch = getchar();

 ch = toupper(ch);

 }

 //关闭端口通讯

 nErr = AdsPortClose();

 if (nErr) cerr << "Error: AdsPortClose: " << nErr << '\n';

}

17.编写完成后，编译，运行，如图：

18.我们在控制台输入“R”并“回车”，TC3中的PLC即开始运行。效果如图：

19.在控制台中再次输入“S”并“回车”，TC3中的PLC运行终止。效果如图：



|  |
| --- |
| 如果您在阅读过程中对于文档有任何疑问，或者发现文档中有差错，欢迎您联系倍福微信公众号。扫描下方二维码即可关注。 |