



固高科技
GOOGOL TECHNOLOGY



CPAC入门使用手册

目录

1. 硬件电路连接.....	1
2. 控制器系统配置.....	1
3. OtoStudio 编程软件操作.....	2
3.1 OtoStudio 软件版本说明.....	2
3.2 新建一个项目.....	3
3.3 编程.....	5
4. 联机调试.....	10
4.1 控制器通讯准备.....	10
4.2 OtoStudio 软件通讯参数设置.....	12
4.3 联机运行.....	14
5. 点位运动例程.....	14

注:产品型号中带有CPAC字样的控制器可以使用0toStudio软件进行程序开发。



1. 硬件电路连接

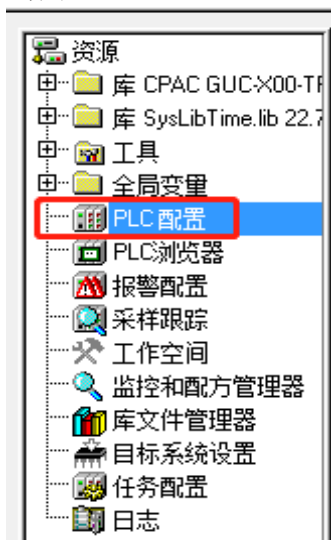
如图,按照图示连接好24V电源线、调试网口线、显示器。USB口可以用于外接键盘、鼠标和U盘。



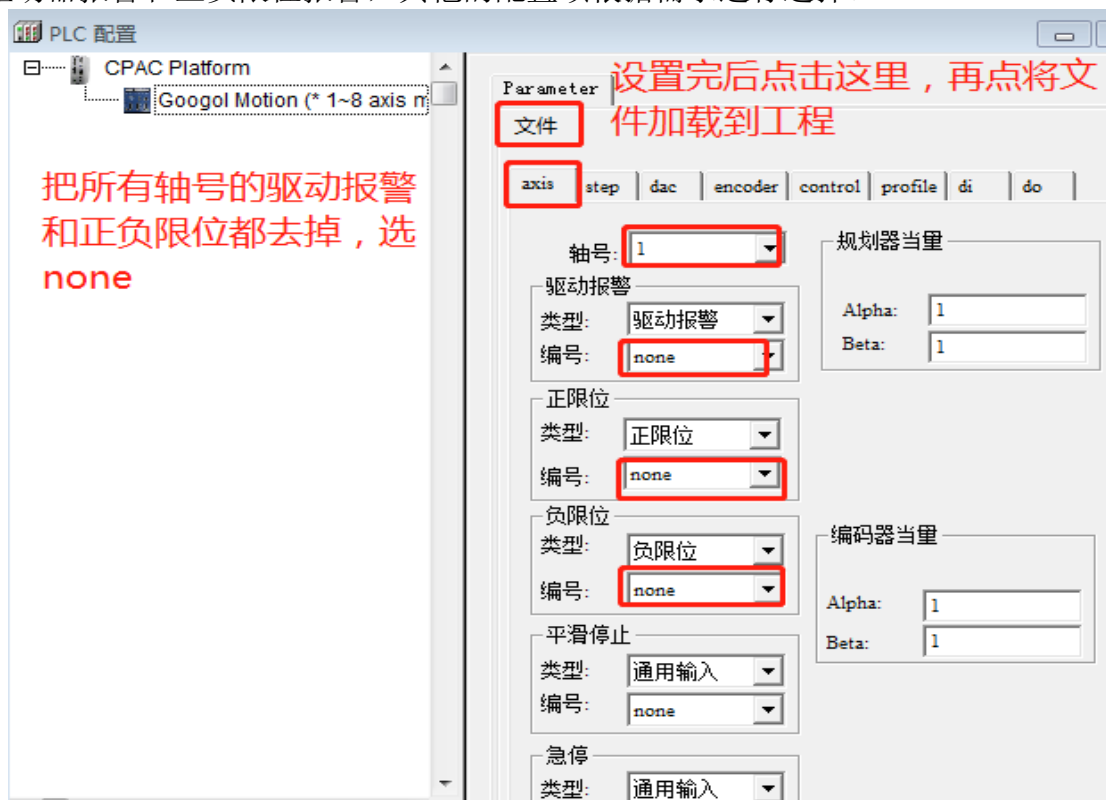
2. 控制器系统配置

在使用CPAC控制器进行运动控制操作之前,需要对控制器进行配置,使控制器的状态和各种工作模式能够满足现场需求。

打开资源管理器中的 PLC 配置。



如果实际控制器没有接驱动器和电机，则要进行如下配置，将每个轴的驱动报警信号和正负限位信号选为 none；如果接了驱动器和正负限位就不需要取消驱动器报警和正负限位报警，其他的配置项根据需求进行选择。



在上图“文件”菜单下点击了“将 MC 配置加载到当前工程”后，当下载程序时，系统会自动将“加载到当前工程”的配置信息加载到下位机中（控制器）。

3. OtoStudio 编程软件操作

3.1 OtoStudio 软件版本说明

OtoStudio 软件分为体检版和通用版。

体验版软件是仿真版的，不能下载程序到 CPAC 控制器中，在固高官网上可以下载到体验版本的软件。

CPAC-OtoStudio-体验版本V2.2

通用版本为正式版编程软件，该软件为收费软件，如果已经购买会随控制器附带，如未附带请联系固高科技进行购买。

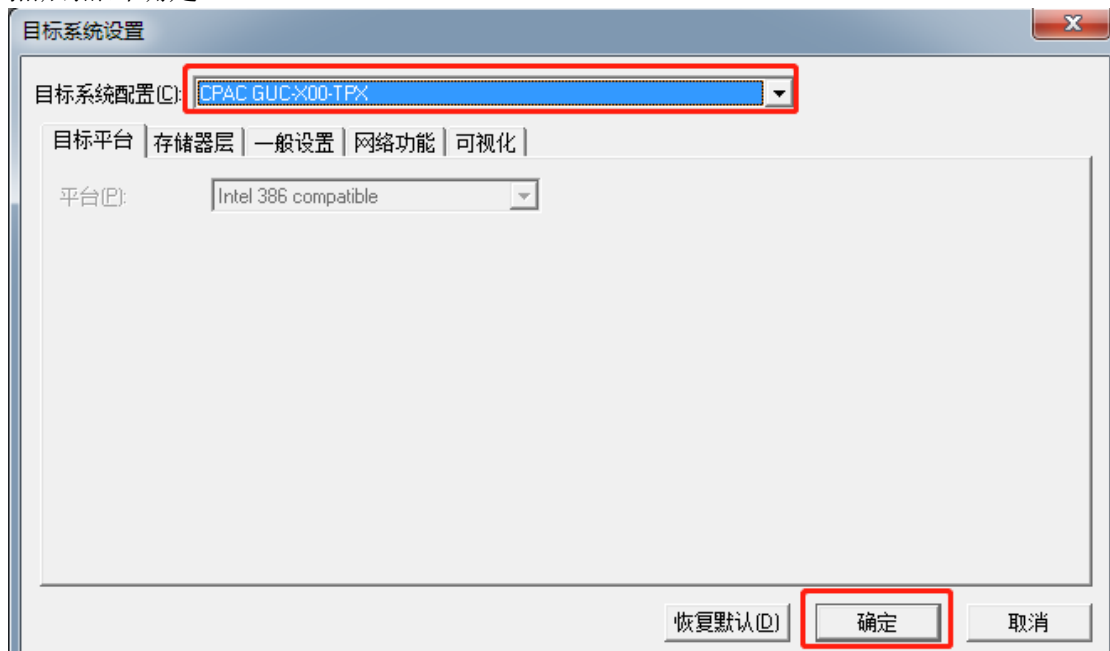
CPAC-OtoStudio通用版本 V2.2

3.2 新建一个项目

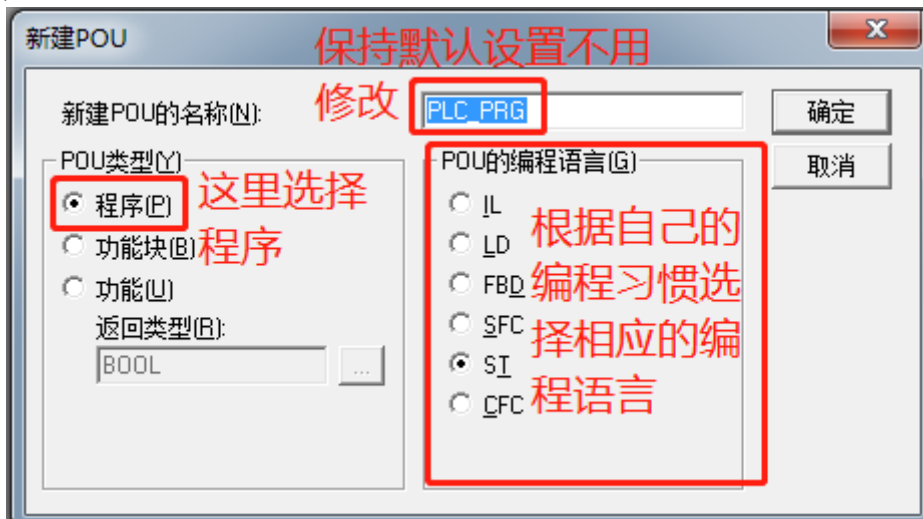
首先打开软件。



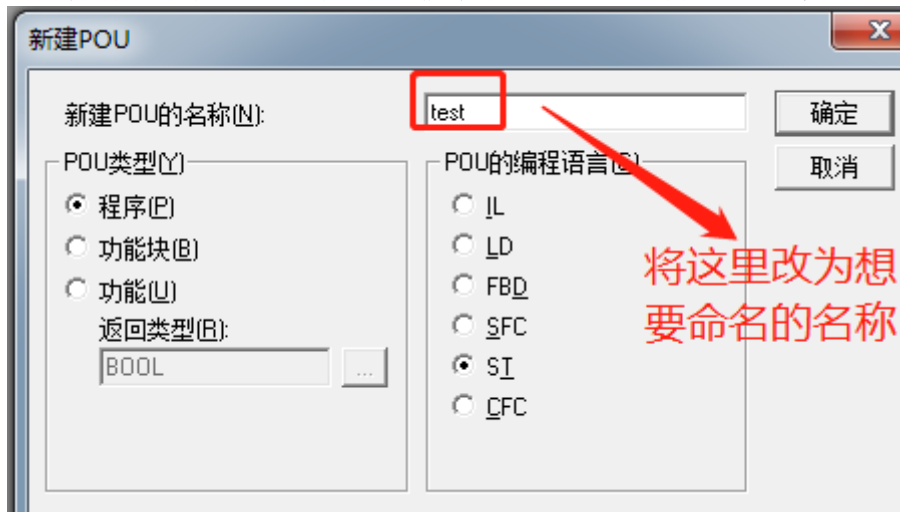
新建一个项目，目标系统一般选为 CPAC GUC-X00-TPX，其它的保持默认设置，然后点击确定。



新建一个 POU 主程序，这个主程序相当于 C 语言中的 main 主程序，是所有程序的入口。



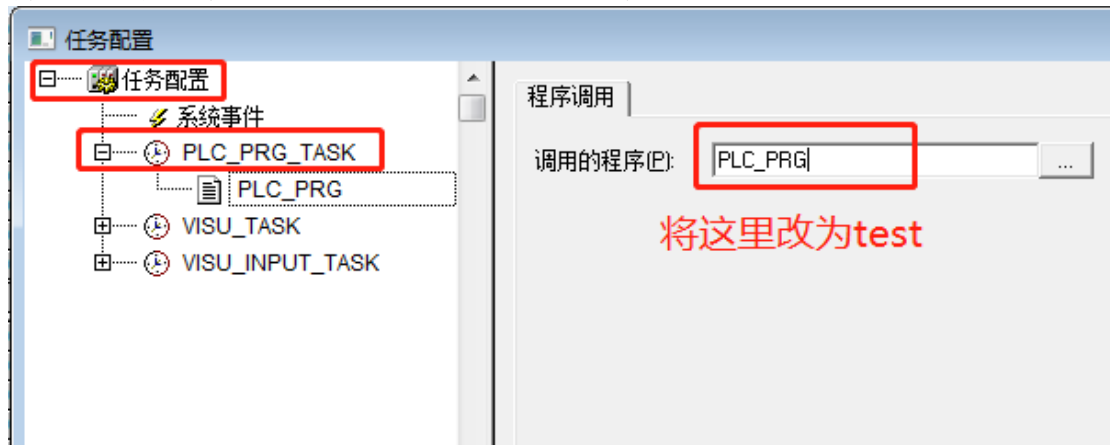
在这一步当中，如果新建的 POU 程序的名称想设为其他的名称，比如设为 test，则要在“资源”中“系统设置”里的“任务配置”//“调用的程序”设为 test。具体操作过程如下图。如果不修改程序名称，则跳过这一步。



切换到资源管理界面，打开任务配置。



将调用的程序改为前面所命名的名称，这里改为 test。



到这一步为止，一个新的项目已经建立完毕，接下来就可以进行程序编写了。

3.3 编程

这里以软件帮助手册里的 JOG 运动模式中的例程进行说明。

a. 第一步，对程序中使用到的变量类型进行定义，这一步也可以在编写程序的过程中再进行定义，因为当使用到未定义的变量系统会弹出变量类型定义提示框。

```

PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003   Right:BOOL;
0004   Left:BOOL;
0005   rtn:INT;
0006   AXIS_X:INT:=1;
0007   Sts:DWORD;
0008   JogPrm:TJogPRM;
0009   PrfVel:LREAL;
0010   PrfPos:LREAL;
0011   Encpos:LREAL;
0012   EncVel:LREAL;
0013 END_VAR
0014

```

定义全局变量类型及
初始值

b. 第二步，JOG 运动模式程序编程，例子中使用的编程语言为 ST 编程语言。该程序实现的功能为：当按下 Right JOG 键，控制器规划电机往正向运动；当按键松开，则电机停止；当按下 Left JOG 键，控制器规划电机往负向运动；当按键松开，则电机停止。

```

PLC_PRG (PRG-ST)
0001 IF RIGHT OR LEFT THEN
0002 (*上伺服*)
0003   rtn:=GT_ClrSts(AXIS_X,1);
0004   rtn:=GT_GetSts(AXIS_X,ADR(Sts),1,0);
0005   rtn:=GT_AxisOn(AXIS_X);
0006 (*将X轴设置为JOG模式*)
0007   rtn:=GT_PrJog(AXIS_X);
0008   JogPrm.acc:=10;
0009   JogPrm.dec:=10;
0010   JogPrm.smooth:=0.8;
0011   rtn:=GT_SetJogPrm(AXIS_X,ADR(JogPrm));
0012 (*启动Jog运动*)
0013   IF RIGHT THEN
0014     rtn:=GT_SetVel(AXIS_X,50);
0015   END_IF
0016   IF LEFT THEN
0017     rtn:=GT_SetVel(AXIS_X,-50);
0018   END_IF
0019   rtn:=GT_Update(SHL(DWORD#1,AXIS_X-1));
0020 ELSE
0021   rtn:=GT_ClrSts(AXIS_X,1);
0022   rtn:=GT_GetSts(AXIS_X,ADR(Sts),1,0);
0023   rtn:=GT_AxisOff(AXIS_X);
0024 END_IF
0025 rtn:=GT_GetSts(AXIS_X,ADR(Sts),1,0);
0026 rtn:=GT_GetPrfVel(AXIS_X,ADR(PrfVel),1,0);
0027 rtn:=GT_GetPrfPos(AXIS_X,ADR(PrfPos),1,0);
0028 rtn:=GT_GetEncPos(AXIS_X,ADR(Encpos),1,0);
0029 rtn:=GT_GetEncVel(AXIS_X,ADR(EncVel),1,0);

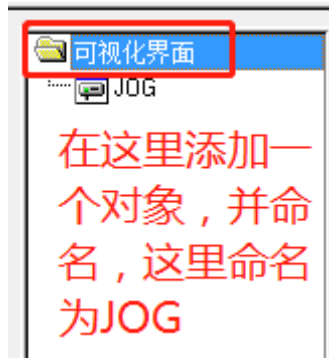
```

JOG运动程序

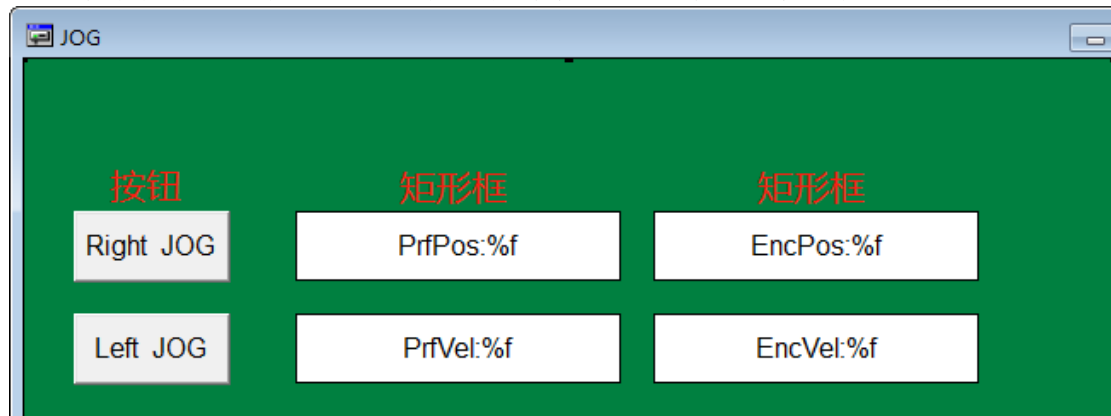
c. 可视化界面编辑，如果此处没有可视化界面对象，则当程序编辑好后在 CPAC 控制器中运行时，将不会有显示界面。



添加可视化界面对象。



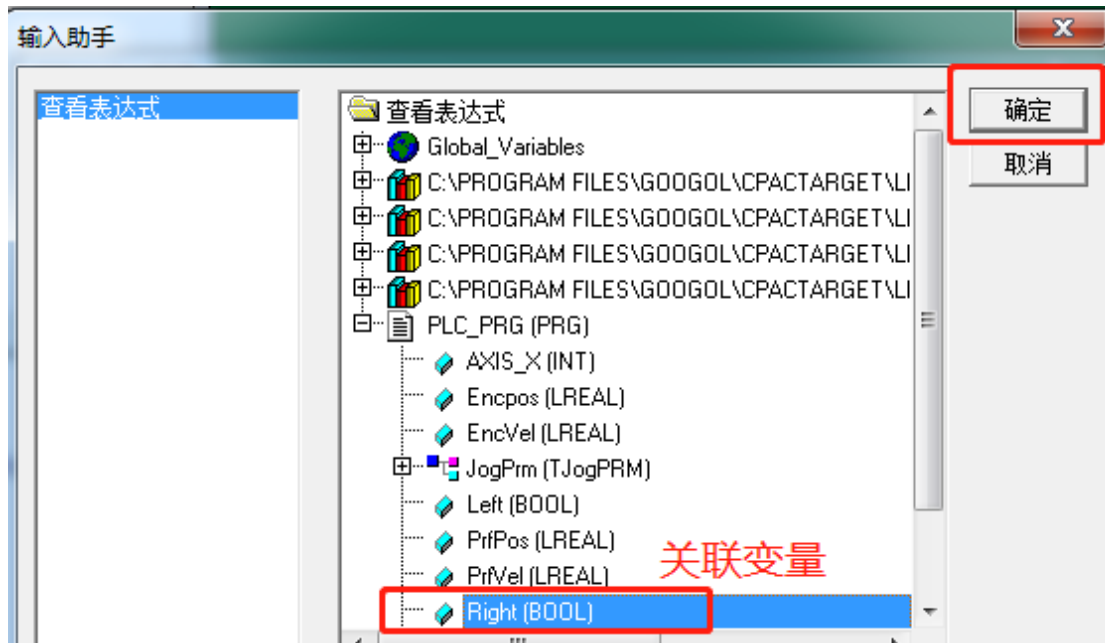
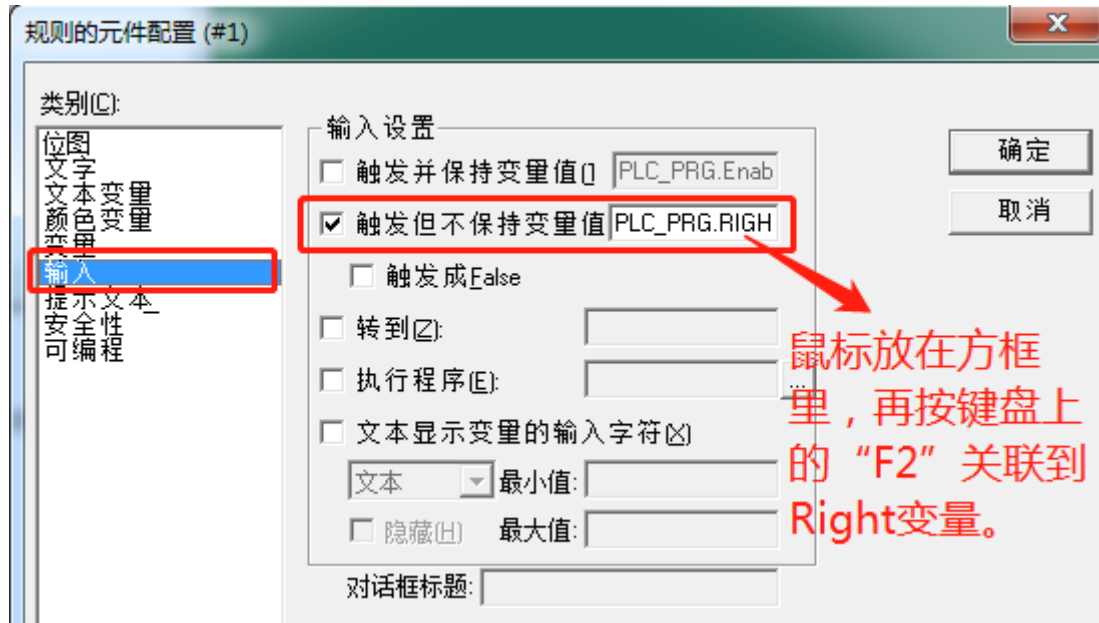
点击菜单栏中的“插入 (I)”插入两个按钮和四个矩形框，如下图所示。



将按钮和程序中的变量关联，方法为鼠标左键双击按钮图标，弹出配置框。

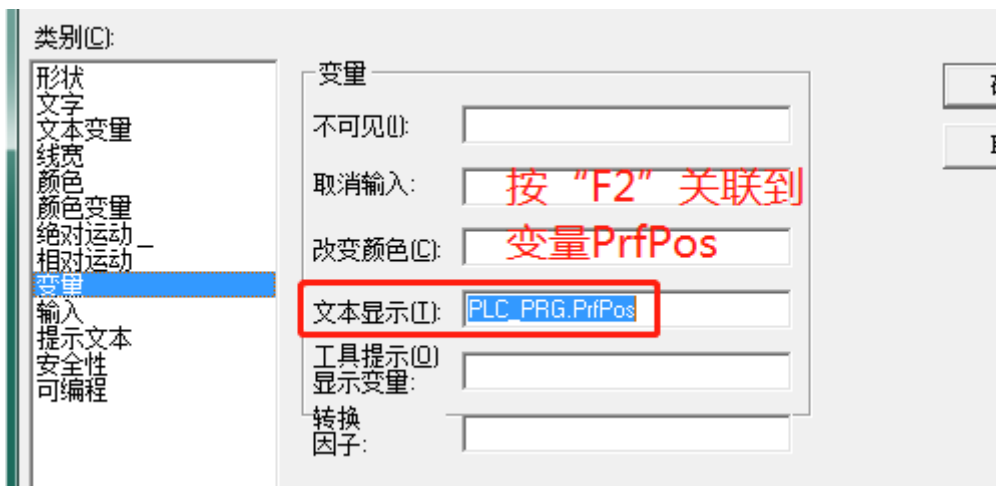


关联变量。

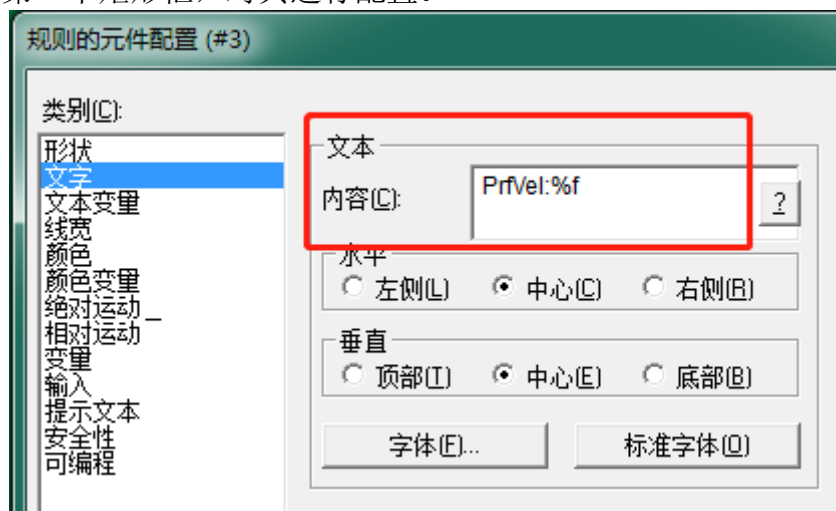


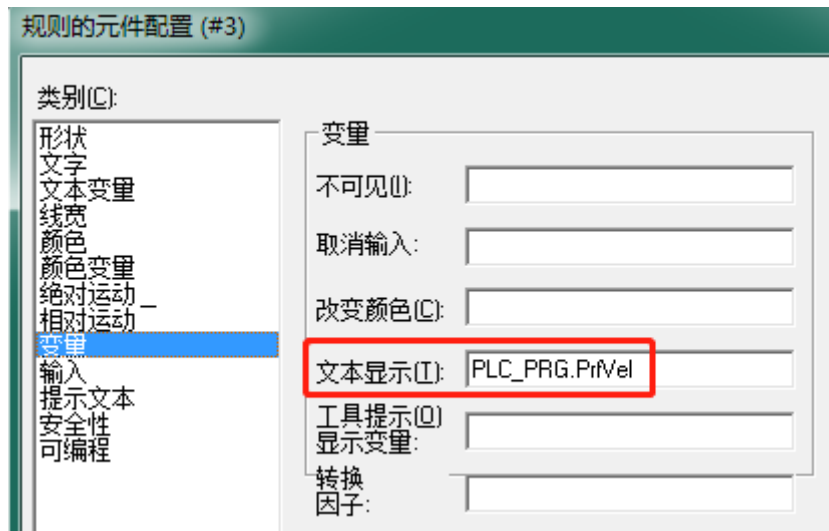
Left JOG 按钮的配置方法和上面一样，只要把关联的变量改为 Left 即可。

左键双击第一个矩形框，对矩形进行配置。



左键双击第二个矩形框，对其进行配置。





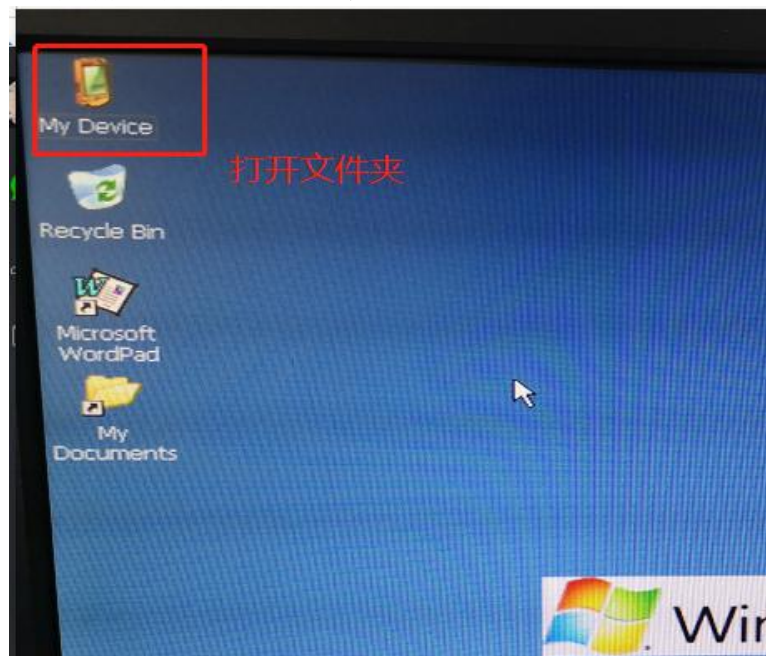
EncPos 和 EncVel 矩形显示框的配置和上面一样，只要把关联变量分别关联到 Encpos 和 EncVel 即可。

4. 联机调试

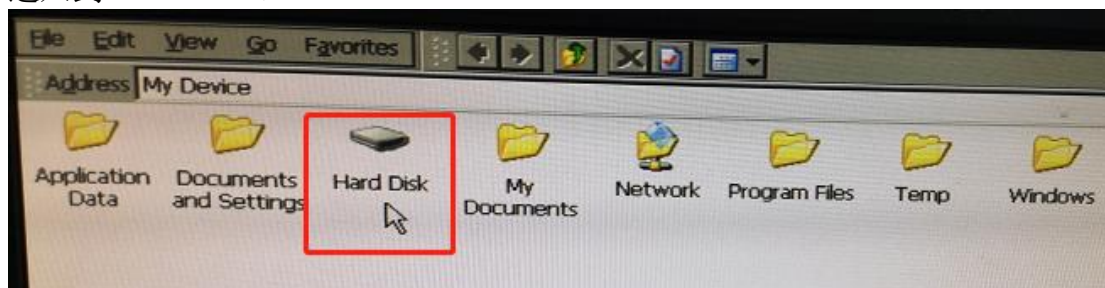
4.1 控制器通讯准备

用网线将电脑和控制器连接好，一般控制器的调试口为 LAN1，IP 地址为 192.168.0.2 或 192.168.1.2。

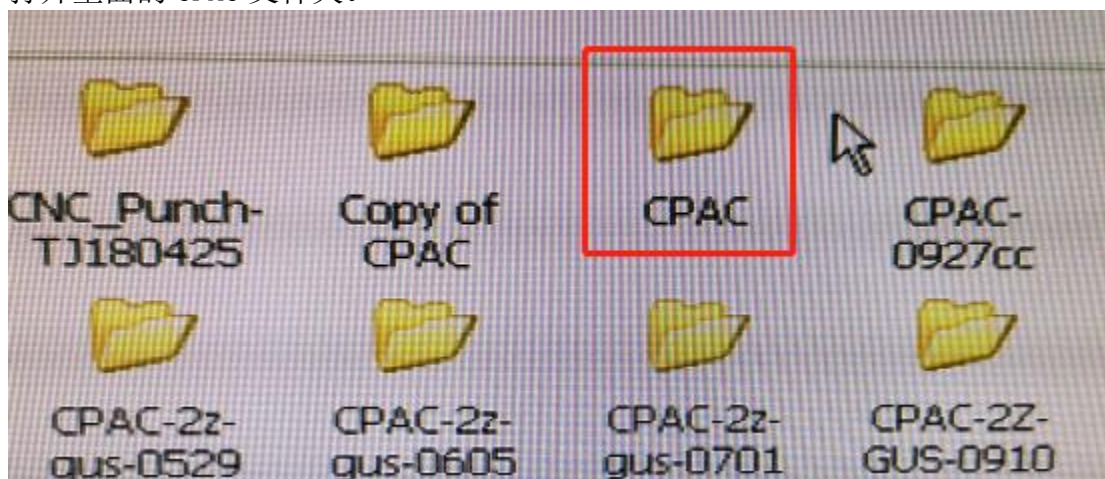
将鼠标插在控制器中，打开资源管理器。



进入到 Hard Disk。



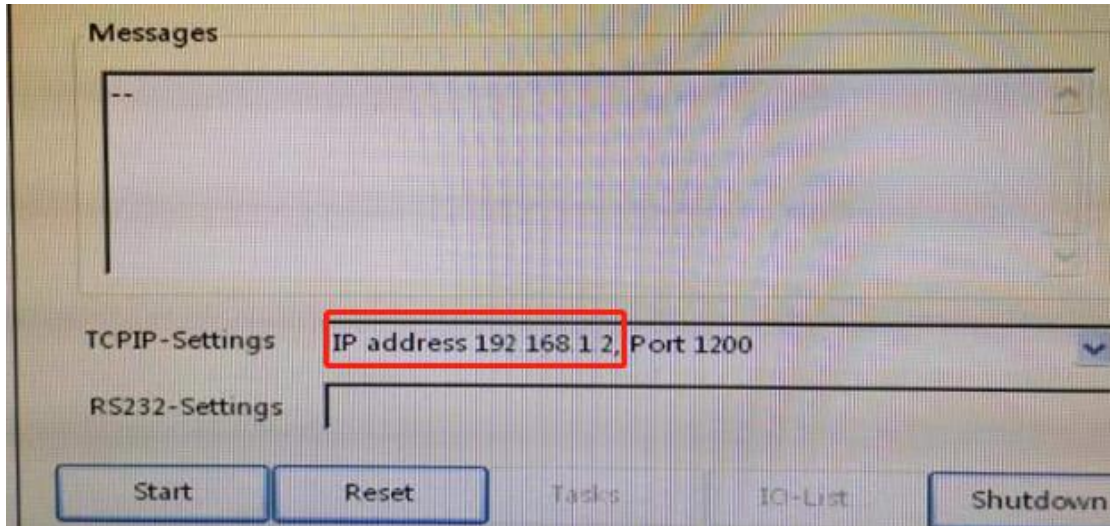
打开里面的 CPAC 文件夹。



运行里面的 GRT 程序，运动控制器的应用软件都是通过 GRT 打开的。

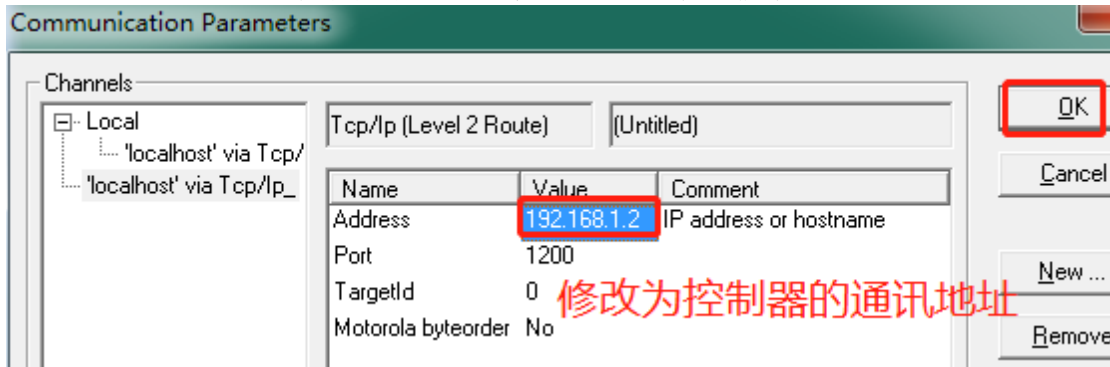


运行 GRT 以后会进入到下图所示的界面，在这个界面我们也可以看到 IP 通讯地址和通讯端口号，一定要双击 GRT 进入这个界面，否则 OtoStudio 软件和控制器的通讯将无法建立。如果一打开控制器就已经进入了下图所示界面，则略过以上几个配置步骤。



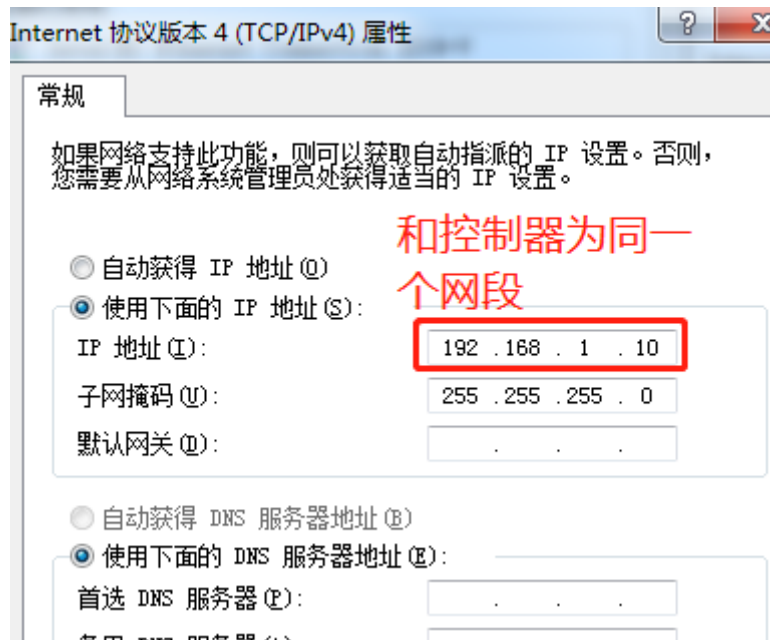
4.2 OtoStudio 软件通讯参数设置

点击 OtoStudio 菜单栏中的“联机”//“参数”修改 IP 地址。



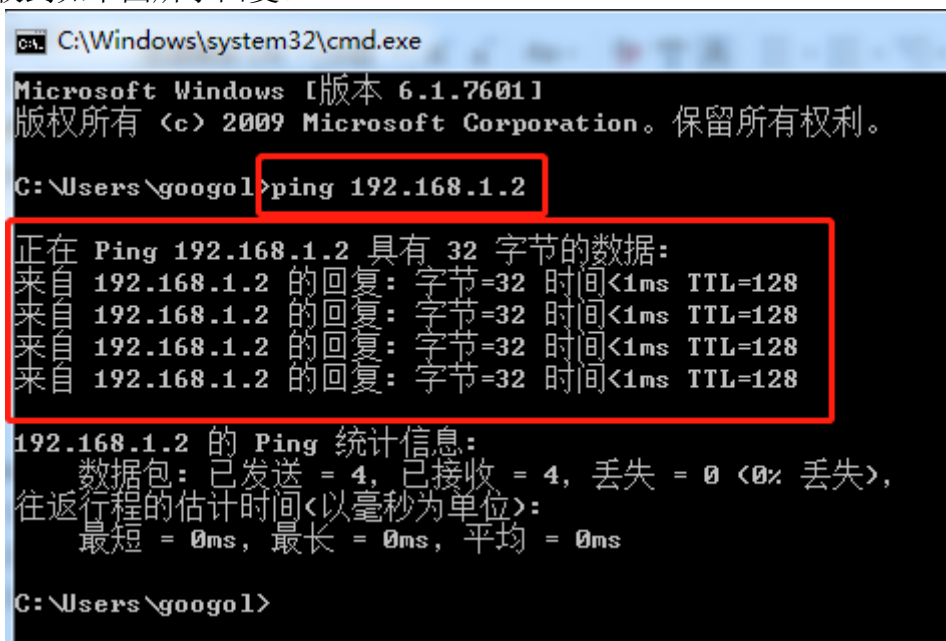
打开调试电脑的“网络和共享中心”//“本地连接”//“属性”把 IP 地址设置为和控制器在同一个网段的地址。





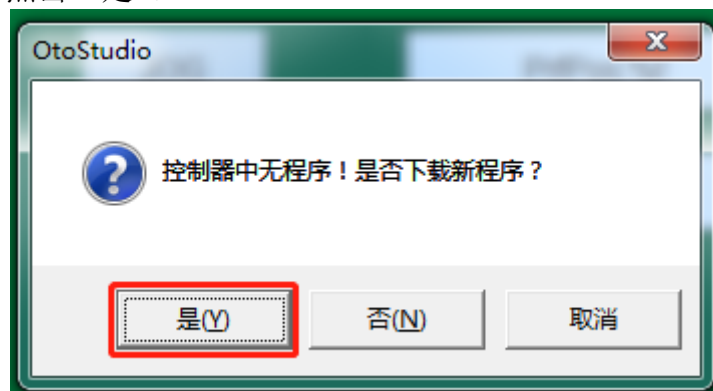
确认电脑是否和控制器通讯上的方法:

在电脑运行框中输入“cmd”点击回车调出电脑的命令提示框，输入命名：ping 192.168.0.2 或 ping 192.168.1.2, 如果控制器和电脑的 IP 地址都是对的，将会收到如下图所示回复。



4.3 联机运行

OtoStudio 软件中的通讯参数和电脑的 IP 地址都设置好，电脑可以 ping 通控制器的 IP 地址后，点击 OtoStudio 软件中的“联机”//“登录”，会出现下载程序的提示，点击“是”。



程序下载完成后再点击“联机”//“运行”就可以运行程序了。如下图所示是以上例程在运动控制器中运行的效果。



5. 点位运动例程

定义全局变量：

```
PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003   rtn: INT;
0004   sts: DWORD;
0005   Enable: BOOL := FALSE;
0006   AXIS_X: INT := 1;
0007   Prm: TTrapPrm;
0008   EncVel: LREAL;
0009   PrfVel: LREAL;
0010   PrfPos: LREAL;
0011   Encpos: LREAL;
0012   star: BOOL := FALSE;
0013   clear: BOOL;
0014 END_VAR
```


点位运动程序:

```
PLC_PRG (PRG-ST)
0004 BBOCCPAM PLC_PRC
0001 IF Enable THEN
0002   rtn:= GT_ClrSts(AXIS_X, 1);
0003   rtn:= GT_GetSts(AXIS_X, ADR(sts), 1, 0);
0004   rtn:= GT_AxisOn(AXIS_X);
0005   rtn:= GT_PrTrp(AXIS_X);
0006   Prm.acc:= 1;
0007   Prm.dec:= 0.5;
0008   Prm.smoothTime:=10;
0009   Prm.velStart:=5;
0010   rtn:= GT_SetTrapPrm(AXIS_X, ADR(Prm));
0011   rtn:= GT_SetVel(AXIS_X, 20);
0012   rtn:= GT_SetPos(AXIS_X, 100000);
0013   IF star THEN
0014     rtn:= GT_Update(1);
0015   END_IF
0016 ELSE
0017   rtn:= GT_Stop(16#1,0);
0018   rtn := GT_ClrSts(AXIS_X,1);
0019   rtn := GT_AxisOff(AXIS_X);
0020 END_IF
0021 IF clear THEN
0022   rtn:=GT_SetPrfPos(AXIS_X,0);
0023 END_IF
0024 rtn:=GT_GetSts(AXIS_X,ADR(sts),1,0);
0025 rtn:=GT_GetPrfVel(AXIS_X,ADR(PrfVel),1,0);
0026 rtn:=GT_GetPrfPos(AXIS_X,ADR(PrfPos),1,0);
0027 rtn:=GT_GetEncPos(AXIS_X,ADR(Encpos),1,0);
0028 rtn:=GT_GetEncVel(AXIS_X,ADR(EncVel),1,0);
```

可视化界面:

