

## 西门子模块6ES7355-2CH00-0AE0参数详细

|      |                             |
|------|-----------------------------|
| 产品名称 | 西门子模块6ES7355-2CH00-0AE0参数详细 |
| 公司名称 | 浔之漫智控技术（上海）有限公司-西门子总部       |
| 价格   | 900.00/件                    |
| 规格参数 | 西门子:模块<br>1000:件<br>保内:原装   |
| 公司地址 | 上海市松江区广富林路4855弄88号3楼        |
| 联系电话 | 18602118379 18602118379     |

## 产品详情

### 西门子模块6ES7355-2CH00-0AE0参数详细

一、S7-300/400之间互连1、采用PLC与PLC通过以太网访问，需要增加以太网模块，如CP 343-1，CP443-1的模块，或者采用带有PN接口的PLC，如CPU315-2DP/PN的PLC。2、采用两块CP343-1互连，先在一个项目里做好两套PLC的硬件组态，分配好IP地址，打开NetPro，选择其中一个PLC，双击连接列表上的空白行，添加一个S7 connection。选择连接的另一个PLC，记住连接的ID号。通讯双方的其中一个站为Client端，激活“Establish an active connection”的选项(默认是激活的)。这样NetPro的配置就完成了，编译、下载完成3、打开其中两套PLC的OB1，调用FB12(BSEND)块发送数据，FB13(BRCV)块接收数据，1#PLC的发送对应2#PLC的接收，1#PLC的接收对应2#PLC的发送。4、调用FB12的时候几个参数的含意，a、REQ，发送开始，上升沿触发发送工作；b、ID，连接ID号，这个ID号是在NetPro组态时生成的ID号；c、R\_ID，连接号，相同连接号的FB块互相对应发送/接收数据，这个是由用户编程时定义的，例如1#PLC的FB12的R\_ID是DW#16#2，则2#PLC的FB13的R\_ID也是DW#16#2，这样才能保证正常接收发送；d、DONE，发送完成，如果REQ端是一个不停的脉冲，正常情况下DONE端也应该也是一个不停的脉冲反馈；e、SD\_1，发送的开始地址，以P# x.x byte xxx来表示，f、LEN，发送数据的长度。5、采用FB12、FB13来进行发送/接收是比较麻烦的，一方面需要两端的以太网模块都支持双边发送，如Cp343-1-Lean就不支持了，另一方面，两边都要写程序，对于增加的系统不太方便，所以这时可以在NetPro里把One-way(单边通讯，激活后FB12、FB13不能使用)激活，采用FB14(GET)取数据、FB15(PUT)发送数据来做，FB14的参数定义和FB12差不多，只是少了R\_ID和LEN，这样更方便一些，而且还有一个好处，不需要对原有的系统进行什么更改，只需要把单边的PLC组态和程序下载就可以了6、需要注意的另一个问题是FB和SFB的区别，在S7-400

上采用SFB，在S7-300上采用FB，在S7-300集成的PN口上，采用SFB，这个问题曾经折磨过我一两天才解决。二、WinLC与PLC互连WinLC是在PC机上的软件PLC，可以把一台PC机当成PLC使用。这样对于不能采用模拟器连接的软件是比较方便的，如采用Intouch、Ifix这些SCADA软件做为上位机软件时，测试是个麻烦的问题，需要与实际机器连接起来才能测试，但如果有了WinLC，可以把程序拷到WinLC上运行，上位机软件与WinLC联机测试，省去不少工夫，所以WinLC对于不方便进行现场测试的环境，是个不错的替代方案。WinLC既然是一套软件PLC，就能够实现PLC的功能，那么与PLC通讯也是可以实现的，下面说一说WinLC与S7-300互连

- 1、新建一个项目，建立一个S7-300的站点，硬件组态；建立一个PC站点，在PC站点上增加一个WinLC和一个IE General，IE General的IP地址与PC机的IP地址\*，其实IE General指的就是PC机的以太网卡。
- 2、打开NetPro，在WinLC上新建一个S7 connection，指向S7-300，编译、存盘
- 3、在PCStation上增加WinLC和IE General，需要注意槽号要与硬件配置时候一样
- 3、打开WinLC的OB块，调用SFB14、SFB15来读写S7-300的数据
- 4、启动WinLC软件，下载程序，这样就可以实现软件PLC与PLC互连了

三、两种方式的结合如果有多台PLC需要与PC机上的SCADA软件通过OPC进行通讯，但是又发现SCADA软件读取的速度太慢，不能满足刷新要求，或者数据要做些预处理才能显示，SCADA软件上运行太多的脚本又影响速度，那么通过软PLC把多台PLC的数据先读过来进行预处理，SCADA软件只与一台PLC进行通讯，这样SCADA软件上的工作量会少一些。这个时候把上面说的两种方式结合一下，就能够实现这样的功能了。

- 1、通讯连接参照上面的方法设置
- 2、在WinLC的PC站点上添加一个OPC服务器，在NetPro里面打开OPC的连接列表，添加一个S7 connection，指向WinLC，编译、下载
- 3、通过OPC客户端可以直接访问OPC服务器的内容，常用的SCADA软件如Intouch、Citect、IFix这些软件都支持OPC，设置访问连接就可以在软件上直接读写了，如果更简单的界面，可以自己用VB编写一个OPC客户端，读写数据都在VB程序上实现，不使用SCADA软件

西门子PLC具有很完善的自诊断功能，如出现故障，借助自诊断程序可以方便的找到出现故障的部件，更换后就可以恢复正常工作。故障处理的方法可参看西门子S7-200PLC系统手册的故障处理指南。实践证明，外部设备的故障率远\*\*PLC，而这些设备故障时，PLC不会自动停机，可使故障范围扩大。为了及时发现故障，可用梯形图程序实现故障的自诊断和自处理。

1. 超时检测机械设备在各工步的所需的时间基本不变，因此可以用时间为参考，在可编程控制器发出信号，相应的外部执行机构开始动作时起动一个定时器开始定时，定时器的设定值比正常情况下该动作的持续时间长20%左右。如某执行机构在正常情况下运行10s后，使限位开关动作，发出动作结束的信号。在该执行机构开始动作时起动设定值为12s的定时器定时，若12s后还没有收到动作结束的信号，由定时器的常开触点发出故障信号，该信号停止正常的程序，起动报警和故障显示程序，使操作人员和维修人员能迅速判别故障的种类，及时采取排除故障的措施。通过项目管理器下载

1. 在项目窗口中，选择想要下载的用户程序或块。
2. 通过选择菜单命令PLC > 下载将所选对象下载到可编程逻辑控制器。其它方法(拖放)

1. 打开项目的离线窗口和在线窗口。
2. 在离线窗口中选择想要下载的对象，并将它们拖到在线窗口中。不带项目管理的下载

1. 使用菜单命令PLC > 显示可访问节点或通过单击工具栏中的相应按钮，打开"可访问节点"窗口。
2. 在"可访问节点"窗口中双击所需节点("MPI=...")以显示"块"文件夹。
3. 打开想要将其用

户程序或块下载到可编程逻辑控制器的库或项目。为此，使用菜单命令文件 > 打开。4. 在为项目或库打开的窗口中，选择想要下载的对象。5. 通过在"可访问节点"窗口中使用拖放将对象复制到"块"文件夹，可将对象下载到可编程逻辑控制器

在可编程控制器中重新装载块可以用新版本的块覆盖已存在于S7可编程逻辑控制器CPU中的装入存储器(RAM)或工作存储器中的块(重载它们)。覆盖已存在的版本。重载S7块的步骤与下载相同。将显示提示信息，询问是否希望覆盖已存在的块。不能删除存储在EPROM中的块，但是一旦重载将会声明无效。装入替换的块到RAM中。这在装入存储器或工作存储器中造成间隙。如果这些间隙zui后意味着没有新的块可以下载，则应该压缩存储器。注意如果电源先关闭然后再打开，并且RAM没有电池装备，或随后CPU的存储器复位，"旧"的块再次有效。在集成的EPROM上保存下载块对于带一个集成EPROM的CPU (例如CPU

312)，可将RAM中的块保存到集成EPROM中，从而在断电或存储器复位后不会丢失数据。1.

使用菜单命令视图 >

在线，显示含已打开项目在线视图的窗口，或，在工具栏中单击"可访问节点"按钮或选择菜单命令PLC > 显示可访问节点，打开"可访问节点"窗口。2.

在项目的在线窗口中选择S7或M7程序，或在"可访问节点"窗口中选择节点。3.

使用下列方法之一，在CPU上选择要保存的"块"文件夹：§

如果正在使用项目管理，那么在项目的在线窗口中 §

如果没有使用项目管理，那么在"可访问节点"窗口中4. 选择菜单命令PLC > 将RAM保存到ROM。

通过EPROM存储卡下载要求要访问专为S7可编程逻辑控制器而设计的编程设备中的EPROM存储卡，需要合适的EPROM驱动程序。要访问为M7可编程控制系统设计的编程设备中的EPROM存储卡，必须安装FLASH文件系统(仅适用于PG 720、PG 740和PG

760)。当安装STEP 7标准软件包时EPROM驱动程序和Flash文件系统作为选件提供。如果使用PC，则需要外部编程器来保存到EPROM存储卡。也可以在以后安装驱动程序。为此，通过开始 > Simatic > STEP 7 >

存储卡参数分配或通过控制面板(双击"存储卡参数分配"图标)调用相应的对话框