

无锡西门子PLC代理商

产品名称	无锡西门子PLC代理商
公司名称	浔之漫智控技术-西门子PLC代理商
价格	666.00/件
规格参数	品牌:西门子 产品规格:模块式 产地:德国
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15221406036

产品详情

无锡西门子PLC代理商

面向对象编程是计

算机语言的一种先进的编程模式，在

工业控制系统的plc

程序中也可以采用这种设计思想，虽然我们无法实现面向对象的很多特点如“继承”，甚至于它根本就不具备面向对象编程语言的特点，但面向对象编程的基本概念就是类和类的实例（即对象），我们只需要使用这种概念就可以了。在计算机编程中我们需要把一些事物抽象和归纳，才能编写类，而在工业控制系统中，控制对象如：电机，阀等等是很明显的控制类别，不需要抽象就可以很明显的针对它们编写类，以下将会用到西门子的Step7编程语言和施奈德的Unity 编程语言来讲解PLC的面向对象编程。

一、实现方式

面向对象编程在Step7中使用功能块（即FB）编程，一谈到此大家就会想到西门子提出的模块化编程，不错，就是这个模块化编程，但西门子提出的模块化、背景数据块、多重背景等名词并不能让大家很明白的理解和使用这种的设计理念。如果大家从面向对象编程的角度去理解，则可以很好的理解这种设计模式。“FB块”被看成“类”，它可以被看成是对相似的控制对象的代码归纳，如对MM440的变频器可以编写FB块:MtrMM440,这在面向对象编程中称为“类”，当需要编程控制具体的电机时，可以给它分配一个背景DB块，在面向对象编程中称为类的实现（即创建类的实例：对象），当需要控制多个电机时，可以分配不同的背景DB到这个FB块，即创建类的多个实例。Step7中有另外一种程序块，即FC块，以FC块为主的编程在西门子中称为结构化编程，这也可以类比于计算机编程中的面向过程编程，即纯粹以函数为主体的编程。

施奈德的Unity软件编程可以更好的理解面向对象编程。它的DFB定义中包含输入/输出参数，私有/共有变量，以及代码实现，而这正是计算机的面向对象编程中“类”的基本元素，而创建类的实例（对象）就像创建普通的“布尔”变量一样，只需在“Function Blocks”中定义这种“类”的变量即可。

Step7和Unity都可以采用面向过程和面向对象编程方式，这两种编程方式的区别类似于计算机语言中的C语言和C++语言编程的区别。

以下的讲解将会把Step7中的FB和Unity中的DFB称为“类”，Step7中的FB+背景DB以及Unity中DFB的实例称为“对象”。

二、面向对象编程架构

以上讲解的是实现细节，而编程思想是建立在程序架构上的，不是某个局部使用了面向对象方式，则可以称之为这种编程就是面向对象编程。这种编程需要从以下方面着

手：

1、 电路设计的结构化。

这里主要以自动线为主介绍，对于单机机床可以是它的简化结构，

<1>、自动线层：这是高层次，它拥有一个主PLC，对属于它下面的各区域控制

<2>、工程层：拥有独立的配送电系统，但没有PLC，只有分布式模块，由自动线控制。顾名思义，它有着较大的独立性，可以作为一个单独的工程项目设计和制造，当自动线比较小时，可以省略该层次。

<3>、功能组层：根据工艺划分，将实现某一个工艺功能的区段设备划分为一个功能组，它隶属于工程层，当工程层被省略时，隶属于自动线层。

面向对象编程并不一定要求使用以上的结构，但好的电气结构更利于面向对象编程。

2、 任何控制对象逻辑都在“类”中实现。

为了做到这点，必须分析与控制对象相关的信息，譬如，对于一个电机，有以下相关的信息需要考虑：

输入信息：

<1>、电路保护信息，如电机的 空气开关，热继电器等。

<2>、功能保护信息，如运动电机的限位开关，风机的风压开关，油泵的油位开关等。

<3>、启动和终止条件，以上的电路保护和功能保护都可能导致电机运转终止，复位也可能导致重新启动，但这里的条件指的是正常运行的启动和终止条件，譬如顺序控制的流程步。

<4>、控制模式：如手动和自动等。

<5>、故障复位：通过复位信息，重新启动。

输出信息：

<1>、控制输出，如控制电机的主 接触器。

<2>、状态信息输出

<3>、故障输出

.....

状态储存信息：

用于代码实现的中间变量以及可以被 人机界面读出的状态变量等

把以上信息都整合到一个类中，并尽量使类的参数标准化。不过，同编程语言还是曾在一些差别，针对Step7,应该遵循的标准是：程序结构由FC实现，对象控制由FB实现

，如下的一种结构体系（其电气结构来自上面的介绍）：这只不过是一个粗略的PLC程序架构体系，好的架构应该更完善和科学。

3、 规划好数据结构

数据结构的定义相当重要，并尽量统一这些结构，不要顾虑存储空间，当今的PLC内存足以容纳大量的数据。说明一点的是在Step7中尽量不要在类的外部定义数据结构(UDT),而是在类里面定义，虽然会造成不同类中同一结构的重复性定义，但却提高了类的独立性。

三、 优越性

1、 标准化

使用这种设计模式，可以将程序设计分为两个阶段，即标准库、基本架构开发，以及实际应用层面设计。其中标准库、基本架构是制定程序标准化的基础，而应用层设计是针对具体的控制工程编程，这样可以把程序设计人员分成两类，一类是标准开发，由程序员负责，一类是应用设计（其中程序调试规划到应用设计），由经过标准化培训的一般程序员完成，通过这种分配就可以解决中国工业自动化中面临的尴尬局面。传统的中国控制工业，一个程序设计由一个人完成，这样他还必须负责现场调试，而拥有丰富经验的程序员一般是三十岁后，这时他已经成家，而显然长期出差对家庭不利，很多的程序员为了家庭考虑不得不改行，要么转到管理岗位，要么去制造工厂搞设备维护，这是资源的严重流失。毫无疑问，使用以上的设计流程，我么可以让经验丰富的程序员搞标准库和架构的设计，而让刚踏入这个行业的年轻人搞应用设计和调试，这不仅可以让老程序员继续他自己的工作，而不影响家庭，也可以让年轻的程序员参入现场调试，培养自己的经验，提高自己的收入。

这可能让某些人士担心，认为年轻的程序员可以参加现场的调试吗？可以肯定的是没有标准化支撑的程序不仅年轻的程序员编不出来，而且现场调试会问题多多。但有了好的标准化后，一年半以上工作经验的程序员就应该能够独立面对自动线。

PLC中的面向对象编程的核心就是黑匣子编程，针对Step7,我们使用FB去实现每一个对象的控制，控制逻辑、报警处理、信号交换全在FB中，对于应用设计人员，不需要明白里面的代码实现，只需要了解该FB的功能以及如何使用好它就行，这样对于应用程序人员的编程能力要求大大降低，对于编程只不过是遵循架构，拷贝代码，改变输入输出条件而已。

那么调试呢？很多人认为使用FB编程的烦就是FB的多次调用后，根本无法诊断这些代码，从技术层面上讲确实如此，我们除了从背景DB上查看信息外，是无法在它多次被调用后监控代码的，但我已说过，这是黑匣子编程，我们不需要诊断这些代码，只需要知道什么样的输入、什么样的参数设定导致什么样的输出就行，代码的逻辑与功能好坏是由标准库开发人员负责的，这就要求标准开发人员需要对他设计的功能块在不同条件下进行不同的测试，保证无误，还需要编写完整、详尽的功能说明文档，以便于应用设计人员了解这些块，标准架构并不是制定出来就一劳永逸的，针对千变万化的工程，它是需要不断完善和修订的，这也是一个工程公司可以实实在在进行知识积累的地方。

程序不仅需要给调试人员使用，而且用户（设备维护人员）也需要了解，如果把完整的标准库文档给用户，可能曾在技术外泄的可能，若不给，对他们诊断设备可能曾在困难，这就需要标准制定人员制作另外一分文档，即设备维护文档，其知识的透漏以用户能够使用程序进行诊断为限。

2、重用性和易管理型

计算机面向对象编程的优点也有重用性和易管理型，在PLC中也曾在，以Step7为

例，需要讨论FC和FB的差异。观察数据类型，FB比FC只不过多一个“STAT”类型，在使用上FB需要背景DB，FC不需要，但就这个差别导致FB拥有自己独立的数据储存空间，而FC的数据储存却必须借助公有变量（如中间变量M或者共享DB），有这样一种准则，程序块的独立性越强，其重用性也越好，产生数据访问冲突的可能性也越少，则更易于管理。有些公司生产的PLC，其程序语言没有类似FB的这种特性，这时可以采用类似“FC+共享DB”的替代方案解决，但它的独立性已经大大降低。

同样的代码的独立性是标准制定的一个重要环节，很难想象一个与其他功能块之间有着千丝万缕联系的功能块能够被作为标准块在不同工程中有效的重复使用。

纵观计算机语言的发展，开始的编程都是令人恐怖的，而当今的编程让人们得到很大的解脱，有很多现成的标准类库实用，人们可以把更多的编程精力放在实现功能本身上，plc编程也应该朝这种方向发展，应该让更多的人从事应用层面的设计，那些标准功能块不应该重复的被不同人员开发，虽然各大PLC厂开发了大量的程序库，但工业控制对象各式各样，不同行业都应该拥有自己的程序库，而代码的可重用性是评价这些功能块好坏的关键。

3、设计思想的先进性

在电路图

设计中我们早已在使用针对控制对象的绘图方式，即把基本的主配送电路和PLC配置完成后，我们会针对每一个现场控制对象如：电机、阀、气缸等控制对象绘制电路图，他们的电源

来自主配送电路，控制和反馈与PLC建立连接，硬件连锁根据实际情况调整，一个个控制对象就象搭建积木一样有组织的堆积起来，同样的，编程也是针对一个个控制对象使用相应的标准控制块实现就可以，把程序控制细节实现了有效的封装，使程序看起来简洁和易

于维护，而好的设计可以把原理图和程序进行很好的关联，甚至于做到一对一的关系，如原理图中的一个控制对象可以在程序中找到相应的FB调用与之对应，真正做到面向控制对象编程。

可能有人疑虑，PLC编程大部分是步进编程，这一个个标准块都是针对控制对象的，那控制顺序如何实现呢？这就要求编写专门的顺序控制FB块，或者使用Siemens现成的Graph7来实现，这点与一般编程没什么差别。

结束语：现在的工业控制领域有很多程序高手，他们很精通算法，也有着自己的编程理念，当我和一些人探讨标准化时，他们认识到标准化的高效性，但认为这样无法体现自己的编程水准，是的，如上所述作为应用层面的程序设计是不要很高的编程水平，但要想一想，一个人难道能一辈子去搞现场调试吗？若想体现自己的价值，可以从事标准编程。我更希望他们能花一点时间研究程序架构，各行各业，真正的大师是系统架构设计者，编程小技巧只不过是好的架构锦上添花

实验条件：电脑已安装的软件 Wincc 7.3、TIA博途 V13、STEP 7 Professional SP1 for V13.0、S7-plcSIM V13.0 SP1 Upd1
实验步骤：1、TIA新建项目，新建设备，控制器CPU选1500系列任一型号，我用的1515-2PN，添加一个输入模块，并设置好

2、添加几个实验变量

3、在控制面板里设置PG/PC接口

4、点击工具栏红色圆圈位置的图标打开仿真软件，下载组态好的PLC和程序，空程序也不影响实验

点击RUN按钮运行仿真

5、新建wincc项目，打开变量管理器，右键单击变量管理，添加新的驱动程序通道

6、在OMS下新建连接，右键设置连接参数

- 7、 然后读取AS符号，这里需要注意，此操作必须是wincc的运行系统已经激活，否则会提示错误
- 8、 变量加载成功
- 9、 选择需要访问的变量，会发现被选择的变量自动添加到了变量标签页内