

T/JQZN

团 体 标 准

T/JQZN 002—2025

西门子 PLC 程序编写规范

2025-7-10 发布

2025-7-15 实施

嘉兴市机器人与智能装备协会 发布

前 言

本标准按 GB/T 1.1-2009《标准化工作导则 第1部分：标准的结构和编写》规定的进行起草。作为组织生产和检验产品的依据，其中的各项技术要求将随技术进步及产品的改进而修改。

本标准由嘉兴市机器人与智能装备协会提出

本标准由麒盛科技股份有限公司、嘉兴一路帮机电技术服务有限公司、嘉兴大学、嘉兴视联智能科技股份有限公司、上海市人工智能技术协会、昆山市人工智能应用创新协会、海宁红狮宝盛科技有限公司、嘉兴子言科技有限公司负责起草。

本标准主要起草人：蔡孝挺、杨国、陈斌、徐晓华、周毅、周振峰、汪未雅、朱伟民、马维、沈振国、沈力、韩圣贤、潘嘉怡、严欢欢。

本标准批准人：周振峰

本标准为首次发布。

1、目的

规范公司自动化设备在招标、设计、采购、制造、安装、调试验收等阶段的电气程序编写标的，有利于后续产线设备的管理与维护。

2、范围

本规范适用于设备制造商在进行定制项目设计、制造阶段应当遵守的必要环节。

3、定义与缩写

3.1 定义

3.1.1 总体要求

针对工厂产线自动化设备在 PLC 控制器程序设计时所涉及的程序结构、变量定义、控制流程等相关内容进行表述

——程序结构

——变量定义

——控制流程规范

3.1.2 程序结构

针对 PLC 程序编程结构做出明确的规定

3.1.3 变量定义

明确规定了 PLC 程序变量的命名规范与定义方法

3.1.4 控制流程规范

针对产线设备的启动控制、报警操作流程控制做出统一的要求

3.2 缩写

3.2.1 PLC

可编程逻辑控制器（Programmable Logic Controller，简称 PLC），一种具有微处理器的数字电子设备，用于自动化控制的数字逻辑控制器，可以将控制指令随时加载存储器内存储与运行。可编程控制器由内部 CPU，指令及数据存储器、输入输出单元、电源模块、数字模拟等单元所模块化组合成。

3.2.2 FC

FC 块在 PLC 中是功能代码（Function Code）的简称，它相当于一个函数或子程序。FC 块用于执行特定的运算或控制任务，通常不包含自己的数据存储区域，而是在运行时使用临时数据区。FC 块可以多次调用，简化了对经常重复发生的任务的编程

3.2.3 FB

函数块（Function Block，简称 FB）是一种在工业自动化领域中广泛使用的编程概念。它是根据功能划分的程序组织单元，用于实现特定的功能或任务。每个函数块都可以看作是一个独立的子程序，拥有输入、输出和内部变量等参数。

3.2.4 OB

OB 块，英文名为 Organization Block，中文名为组织块，可以视为 PLC 程序中的主函数（main function），负责协调 PLC 的各种操作和任务。它的功能涵盖了从 CPU 启动、程序循环执行到错误处理的多个方面，是 PLC 程序运行的核心。OB 块的执行优先级由编号决定，编号小的优先级高。

4、细则/描述

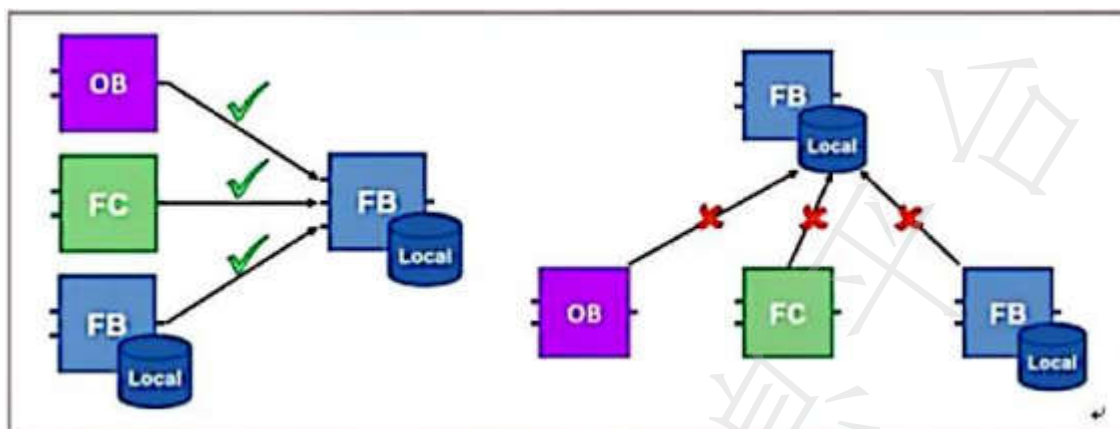
4.1 程序结构

4.1.1 程序调用方法

——所有功能的主要函数（FC）/逻辑程序需要在启动程序或主程序中进行调用。

——不同功能的函数之间不进行互相调用，而只使用相关数据块变量进行数据交换和逻辑运算。

- 需要细分的子程序和函数，可以由相关功能的主要函数/程序进行调用。
- 函数程序调用逻辑应该简洁直观，按照此功能进行独立。如图所示



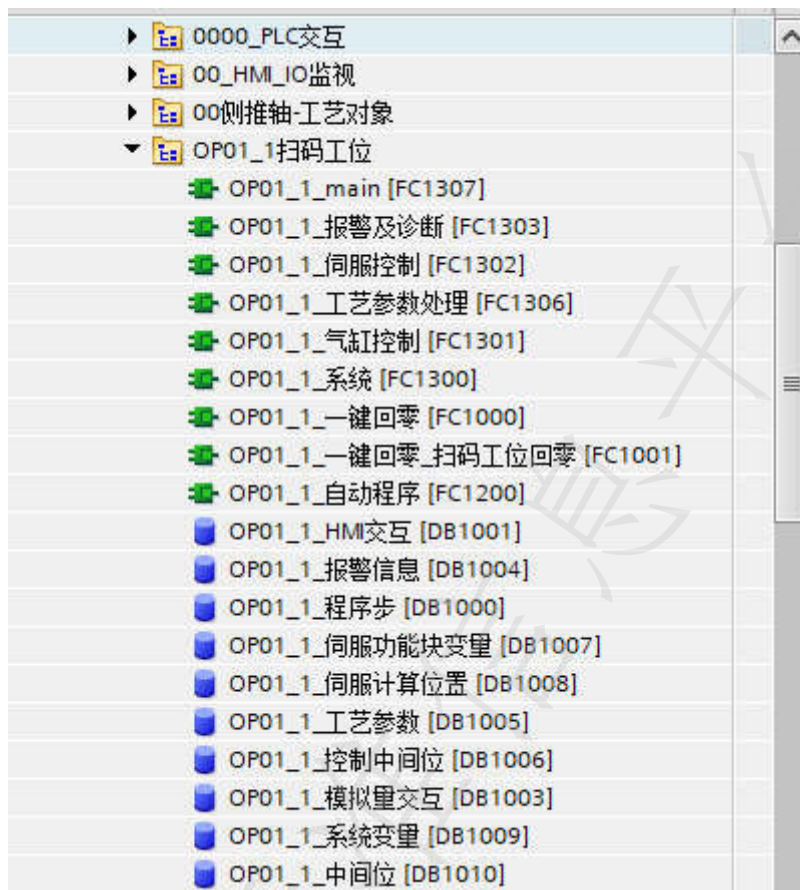
程序调用方式示意

FB 应具有详细的功能说明，且能正确实现每一个对象的控制，其控制逻辑、报警处理、型号交互等应简洁明了、便于查看，确保设备维护人员能快速掌握功能块结构。

4.1.2 程序树内容

程序项目树需要进行明确分组，分组依据为程序编程的具体功能，主程序树应包含但不局限以下内容，简易设备及单一功能程序可对其中未使用的模块留空，但应保留组项以保证功能区分的统一。

- MasterCopies 程序结构模板用于保存程序初始样例，作为编程结构大纲使用，核心控制程序样例同样保存于此程序组中，以便调取和修改使用。
- Type_PLC 程序组用于编写核心工艺功能块，规范规定常用于工艺需要尽可能使用封装功能块的功能的方式编程，每个功能程序组会有不同数量的功能块需要编写，所以 Type_PLC 程序组中需要再次细分组别，分组结构依据与主程序树相同
- 编写 Organization_Blocks 用于统一调用各功能模块子程序，仅作调用功能，力求简洁高效，不进行逻辑判断及额外功能编程。
- 当使用带安全功能的 PLC 时，应在 Safety 程序中编写安全相关逻辑程序。若 PLC 未集成安全功能，则也将系统安全联锁、急停等相关功能编写在 Safety 组内。
- 调用初始化程序功能块，恢复程序和相关参数至初始状态，检查报警信息是否正常。
- 程序树目录中每个子程序的注释必须要清楚的解释对应子程序的功能及内容，遵循实际物理工序排布，工位自动程序优先，中间不能穿插功能程序，方便维修人员快速定位目标程序位置处理设备异常。
- 人机界面管理程序组用于在固定显示屏与移动操作面板上实现操作模式管理、画面功能的调用，以及切换操作模式等。
- 运行模式程序组用于编写操作模式的调用指令，并且启动不同模式下的顺序控制指令。顺控相关功能块的函数应该根据不同操作模式，分别存放于操作模式名称相对应的子程序中。
- 生产信息程序组用于调用生产相关数据功能块和数据块。数据配合 HMI 设置，将生产信息完整无误的显示于人机界面。
- 诊断信息程序组用于调用系统诊断相关功能函数，将诊断所需的数据块编制在组内，对于需诊断子设备较多的产线，需要对诊断数据块进行分组。



程序数目录示意图

4.1.3 编程语言

- a) 主程序只能用梯形图 LAD 或者 SFC 作为编程语言，禁止使用 STL、ST 及 SCL 等高级编程语言。
- b) 运算和特殊功能块（和映射相关）可以用高级编程语言（仅限于功能块内或重复功能的子程序，避免重复调用，功能块不得加密）。
- c) 子程序的编写与主程序编写要求一致。

4.1.4 编程规则

- a) 程序中的输入输出型号（I/O），需要建立 PLC 变量中转，以便在 PLC 输入输出触点变化时可以快速修改程序。
- b) 自动模式切换到手动状态可单工位复位，跳转步用工序名称加步号。
- c) 电机气缸等执行机构需在自动切换手动时可在满足安全的情况下进行操作，切换成自动后可正常运行。
- d) 报警类目需细分为：一级报警、二级报警、提示信息，一级报警为设备处于急停状态，此时设备无法进行任何动作，运行指示灯亮红色且蜂鸣器报警，二级报警为对产品的质量/安全、对设备的效率/DT 时间产生影响的故障，指示灯及蜂鸣器状态与一级一致，提示信息为工站缺料提示、保养提示等不影响设备运行的信息。
- e) 报警程序需要置位自锁，人员完成故障确认后并通过按钮完成报警清除。
- f) 程序执行机构（伺服、气缸、直线电机等包括但不限于）的驱动信号要有安全限制条件，必须在满足安全保护条件下才能执行动作（如不满足安全条件需进行信息提示）。

4.1.5 程序功能要求

- a) 在程序中建立动作步流程功能块、并且要在 HMI 中显示当前设备执行的步数，各步序动作当前状态（自动运行中、手动中、故障中），需要在 HMI 中清晰的体现出来，方便维护人员快速定位

故障点，排除问题。

- b) 程序中应具备通讯状态监控，（通讯 IP 地址/设备心跳/上位机/激光器/MES/CCD）显示在 HMI 中，方便人员快速判断。
- c) 自动化的每一步动作都可实现单独手动操作，除全自动模式外，需要增加半自动功能，即自动化设备需要具备如下模式：A 调机模式：在设备工艺调试阶段，在此模式下运行，B 全自动模式：自动循环模式。
- d) 设备有自我故障诊断功能，可以在触摸屏上显示相应故障，实现人机交互，PLC 要求具备故障信息位置显示分析功能，针对设备故障进行警示，操作人员根据设备故障点显示快速查找设备故障点，根据故障点信息的触摸操作可以调出故障的分析信息，便于故障排除。例如：根据信息分析可以查找故障位置、相关电气元件编号、电路图纸标号、处理故障方法等。

4.1.6 功能块编写

- a) 采用复合 IEC-6113-3 标准的编程语言进行 PLC 程序设计，应选择 LAD（梯形图）作为块的编程语言。
- b) 功能块程序访问应避免访问全局变量。
- c) 数据始终存储在 CPU 的 RAM 中，仅在特殊情况下才允许使用装载存储器存储数据块。
- d) 优先使用多实例代替单实例。
- e) 功能块之间数据访问通过输入输出接口访问。
- f) 块注释必须以一行或几行的形式放在相应的代码段前面，每个块注释内容格式包含以下各项。
 - Company 公司名称
 - Library 库名称
 - A firmware version of the tested PLC(eg S7-1511 V1.6) 已测试固件版本的 PLC（例如 S7-1511 V1.6）
 - Software version at the time of creation 创建时的软件版本
 - Use restrictions (such as certain OB types) 使用限制（例如某些 OB 类型）
 - Requirements (for example, other hardware) 要求（例如，硬件）
 - Function Description 功能说明
 - Tile version with author and date 带有作者和日期的图块版本

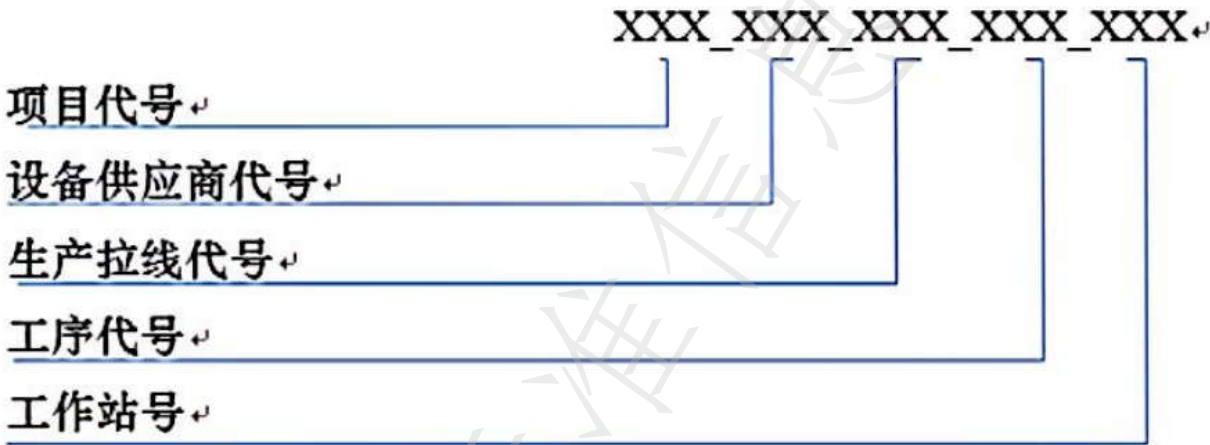
```
//=====
// Company// (c)Copyright (year)
//-----
// Library:      (that the source is dedicated to)
// Tested with:  (test system with FW version)
// Engineering:  TIA Portal (SW version)
// Restrictions: (OB types, etc.)
// Requirements: (hardware, technological package, memory needed, etc.)
// Functionality:(that is implemented in the block)
//-----
// Change log table:
// Version  Date      Expert in charge  Changes applied
// 01.00.00 dd.mm.yyyy (Name of expert)  First released version
//=====
```

- g) 要求在代码行的末尾或代码之前添加行注释。
- h) 功能块引脚
 - 输入输出没有前缀
 - 通过块接口进行交换
 - 如果多个 FB 或 FC 中需要数据，则通过块接口（输入，输出和 InOut 接口）处理数据交换，禁止直接访问 FB 外部的静态变量

- 对于数据类型（例如 WORD,DWORD,REAL,INT,TIME 类型），应使用输入或输出接口类型
 - 对于基本数据类型，仅当在块的外部和内部写入时才使用 InOut 接口类型
 - 如果传输了许多参数，则应尝试将它们封装在 PLC 数据类型中。然后应将此 PLC 数据类型声明为 InOut 标机
 - 对于结构化标签（例如，ARRAY,STRING 等类型）和 PLC 数据类型，通常应使用 InOut 接口类型
- i) 为了能够从代码中的传输和输出参数中清楚的分离出静态和临时标签，使用标签前缀，前缀静态标签：stat；临时标签：temp

4.2 变量定义

为了方便备份和程序查找需要按照如下标准命名项目，再根据日期来判断最新的备份数据，程序命名项目代号，设备供应商代号，生产线代号，工序代号，工作站号与 PLC 的 IP 站号一致。



4.2.1 变量分隔符

综合 PCL 程序编程软件和触摸屏程序的变量的命名规则，很多符号是禁止使用或者使用中风险的，比如<>,!,+等基本都不能使用。变量分隔符统一使用下划线“_”。

4.2.2 IO 定义

命名原则：为使得程序的使用者可以在短时间内定位到触点位置和功能。



4.2.3 常量定义

常量的定义使用大写字母和下划线，常量的名称总是以大写字母书写要检测单个单词或缩写，应在单个单词或缩写之间使用下划线。比如 MAX_VELOCITY。为了保证以后在库中使用这些块，在这些块中仅使用局部常量，这样可以保证在用户程序中编译不会由于缺少程序部件而导致错误。



4.2.4 中间变量定义

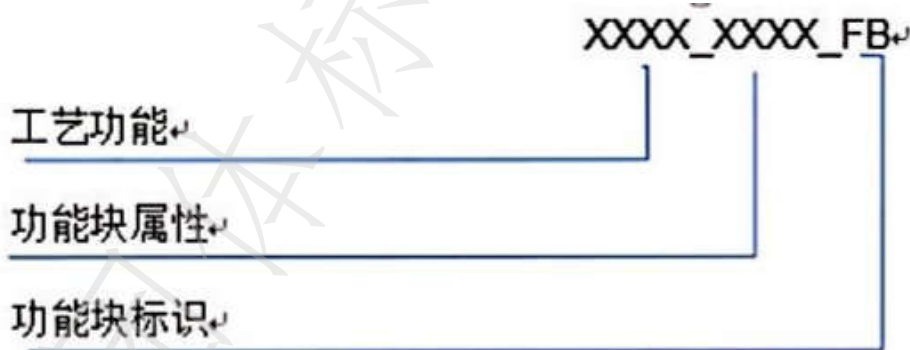
命名原则：用下划线分隔各个命名部分，第一部分首字母大写表明功能，第二部分首字母大写表明属性，如果功能统一，可使用一部分，当功能描述复杂，则按需增加分隔部分，规定首字母大写，示例：Timer_Operation。



4.2.5 循环因子定义

优先使用：i, j, k, l, m, n

4.2.6 功能块定义

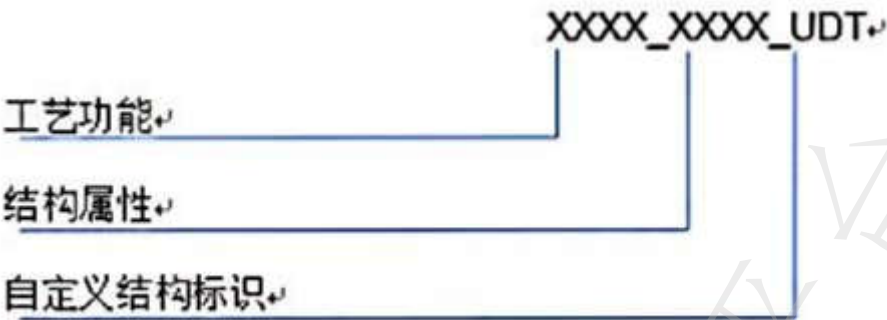


- a) 命名原则以工艺功能为首，字母大写，尽可能简略或使用通俗易懂的简称，使用下划线分隔，第二部分以大写字母表示功能块属性，使用下横线分隔，第三部分以 FB 结尾作为功能块标识，示例：POSDEV_1S2D2P_FB（单电控，二位三通阀）。
- b) 功能块引脚名称定义，全部使用大写字母，优先使用：功能/对象_属性，Xxx_Xxx 的形式，根据复杂意义的需求，可按需增加分隔，以简洁易懂为原则，示例 Alarm_Mode。

4.2.7 用户自定义结构定义

数据结构的定义相当重要，并尽量统一这些结构。

- a) 外层结构名称命名原则：以工艺功能为首，字母大写，尽可能简略或使用易懂的简称，使用下划线分隔，第二部分以大写字母表述结构属性，使用下横线分隔，第三部分以 UDT 结尾作为功能标识。示例：OPMODE_UDT



- b) 结构内部命名原则：用下横线分隔各个命名部分，第一部分首写字母表明功能，第二部分首字母大写表明属性，如果功能简单统一，可只使用一部分，但功能描述复杂，则需增加分隔部分，规则同首字母大写。示例：Plant_Identifier



4.2.8 Array 数组

数组索引以 0 开头，以常数结尾

4.3 设备控制流程

完整的启动流程应包含，但不仅限于：

4.3.1 启动流程

- 程序初始化：设备恢复默认开机状态，参数复原但不重置生产相关设定。
- 各站设备自检：对传感器信号进行前确认，PLC 及相关子设备诊断状态确认，确保产线通畅。
- 安全回路确认：确认程序内安全联锁功能、急停回路、集成安全功能无异常。
- 设备使能：设备动力回路接通，根据功能设定开始运转，等待物料。
- 每个启动步骤与运行状态，都应该包含完整的启动失败后退回操作，给出相应提示，根据安全级别不同进行不同等级的报错。

4.3.2 启动安全规定

启动开机过程中后，延时 3 秒提醒时间，3 秒内蜂鸣器播放开机提示声音，三色灯黄灯闪烁，过程中如出现安全报警、急停等安全相关故障信息，设备终止启动，退回开机状态，如无报警则 3 秒内开机成功，设备进入正常生产流程

4.3.2 生产中中断规定

生产中中断可能由多方面原因造成，如人工停止、设备报错停止、生产错误停止等。所有生产流程需进行状态记忆，特殊关键位置比如机械手上下料位置，需要添加检测装置与设备记忆互锁处理。在排除停止原因后，由停止时的生产步骤回复生产，尽可能避免全局初始化，保证生产效率和物料节约。

4.3.2 停机安全规定

- 急停：设备控制回路断电，所有运动机构立即停止
- 正常停止：完全当前一个物料加工后停止
- 设备安全回路通过安全继电器，禁止通过任何方式屏蔽安全回路