Koyo

Value & Technology

可编程序控制器 DL350 系列
用户手册

［增补版］

光洋电子(无锡)有限公司

目　　录

第一章　　开始使用.....1
第二章　　安装，接线和详细说明.....15
第三章　　CPU 规格及其操作.....67
第四章　　系统设计和设置.....102
第五章　　维护与故障排除.....135
附录 A　　菜单操作功能.....160
附录 B　　错误代码.....169
附录 C　　指令执行时间.....178
附录 D　　特殊继电器.....204
附录 E　DL305 产品重量.....208
附录 F　I/O 寻址的常规方法.....210

第一章　开始使用

1

开始使用

本章将介绍....

－介绍

－**DL305** 系统的各组成部分

－编程方法

－*Direct*LOGIC ™定义号系统

－设计一个成功的系统的步骤

介绍

编写目的

承蒙采用本公司 DL305 系列的产品。本手册将教您如何安装、编程及维护我们的产品，同时也能使您了解它是怎样和其他控制系统的装置协调工作的。本手册还包含一些重要的内容，它将涉及到 DL305 PLC 及其组件的安装和 PLC 编程。如果您对 PLC 系统已经比较了解，我们的手册能提供您建立及维护系统所需要的全部信息。



从哪里开始

如果您已经对 PLC 有所了解，请阅读第二章 “安装、接线和详细说明”，也可按您个人情况跳过一些章节。当您有说明问题的时候，可以把本手册当参考书来看。如果您是初次使用 DL305，我们建议您最好通读本手册以便了解 DL305 系列产品的多种特性。我们相信您一定会为自己能用 ***Direct*LOGIC** ™ 产品完成很多的任务而感到惊喜。

增补手册

如果您已经购买了操作界面或 ***Direct*SOFT**™，您还需要一些产品附带的手册来补充本手册。

技术支持

我们意识到即使我们努力做到了做好，一些您所需要的信息，由于安排方式上的原因而使您找不到。因此首先，查看以下帮助资源从而可以决定所需信息属于哪个部分。

- **目录表**——在本手册的开头，列有目录的各章节名称
- **目录的快速指南**——下一页上列出的各章概要
- **附录**——关键主题的参考材料，靠近本手册的结尾部分
- **索引**——按关键词的字母顺序来排列，在本手册末尾

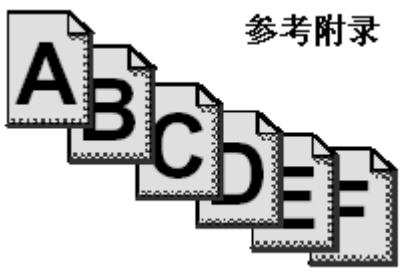
您也可以通过访问我们的网站来了解我们最新产品的信息。

- **互联网**——地址为 Http://antomationdirect.plcdirect.com

如果您还需要进一步的帮助，可以拨打 770－844－4200。我们的技术支持小组会很乐意回答您的问题。拨打电话请在工作日的 9：00～18：00 上午九点至下午六点，东部标准时间。如果您对我们的产品、服务或手册有什么问题或建议，请填写并寄回夹在本手册中的“回执卡”。

章号		本手册的主要内容都包含在一下章节中
1	**开始使用**	介绍 DL305 PLC 系统的各个组件
2	**安装、接线和详细说明**	解释如何准备系统安装，并提供安全的流程指导来确保人和机器的安全，本章还包含系统和 I/O 口的接线图以及 I/O 模块的详细介绍。
3	**CPU 的详细说明和相关操作**	列出了系统的操作和 DL305 CPU 通讯口的详细说明
4	**系统的设计和设置**	解释了各种扩展 I/O 口的方法和 DL305 CPU 的通讯口
5	**标准 RLL 指令**	描述了 150 条以上标准指令的作用，同时包含 DirectSOFT 和手持编程器的编程例子。注意：鼓形指令和级指令将在后面的章节中介绍。
6	**鼓形指令编程**	描述了 DL305 CPU 所支持的 4 个不同的定时器/事件鼓形控制器。本章以操作方面的细节问题开始，包括了控制技巧并揭示了鼓形控制器是如何解决顺序工作的问题。
7	**RLL^{+}级式编程**	很适合初学的编程人员－它将教您如何给您的工序作一个简单的状态变换图并把它们转换成级式语言。这种编程方法对编程和调试来说都是一种节省时间的好方法。
8	**PID 回路操作**	解释了 DL305 PID 回路功能如何实现了 4 个回路。讨论了包括 PID 运算，回路调整，报警，？/？协议子集等。
9	**维护与故障排除**	本章将帮助您诊断、修理和避免系统出问题，还包括如何查找I/O回路的问题，梯形图程序等错误的解决办法。

附录 有关 DL305 的一些附加的参考信息包含在一下五个附录中：



- A－辅助功能
- B－DL305 出错代码
- C－指令执行时间
- D－特殊继电器
- E－产品重量
- F－基本框架和 I/O 口的设置（常规方法）

DL305 系统组成

DL305 家族是一个种结构非常紧凑而且功能非常强大的产品系列。它的 CPU 虽小，但能提供许多只在大规模或很昂贵的系统中才使用的指令。在如今发展迅速的工控领域，模块化的设计能提供更灵活的选择。以下是 DL305 系统组成的概要。

CPU

在这一生产线上有三种功能增强的 CPU,DL330，DL340 和 DL350。本手册只涉及到 DL350 CPU。DL330 和 DL340 CPU 的详细情况包含在 DL305C 的用户手册里。DL350 CPU 含有内置的通讯口，一个大容量程序存储器，丰富的指令系统和高级诊断。同时，含有鼓形定时器，浮点运算功能，内置的能够实现自动调整的 PID 回路也是它的一个特色。

基本框架

共有三种基本框架：5 槽，8 槽和 10 槽。其中一个是插 CPU 模块的，其他的插各种 I/O 模块。所有的基本框架都内置了电源。最近新出了两种基本框架。型号为×××××－1 的基本框架能对 DL305 CPU 进行补充。这三种 CPU 都能在这几个基本框架上使用，并能融入某个系统中。当 DL305 CPU 用在某个老型号的基本框架上，或用在不同的基本框架所组成的系统中时，它在寻址和 I/O 设置上类似于 DL340 CPU (见附录 F)

I/O 设置

配合最新的基本框架，DL350 CPU 能支持最多 368 个输入/输出点。这些点可根据需要分配成输入点或输出点。DL305 系统可以通过增加远程 I/O 口来进行扩展。DL350 还提供内置的远程 I/O 口控制器。

I/O 模块

DL305 有一些在工业上很强大的模块。提供了支持 24VDC、110/220VAC 和触点电流高达 10A 的继电器输出等各种离散模块。

编程方法

DL350 CPU 支持两种编程方法， RLL（继电器梯形图逻辑）和 RLL^{+}（级式编程）。DirectSOFT 编程软件和手持编程器都能支持 RLL 和级式编程。

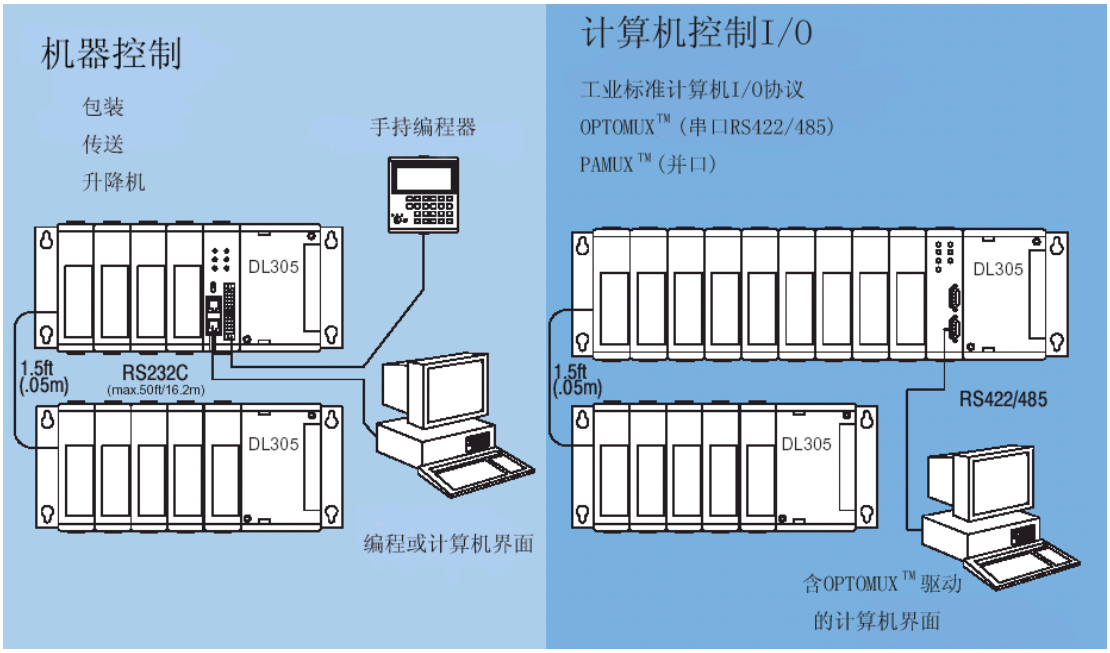
基于 Windows 操作系统的 DirectSOFT 编程

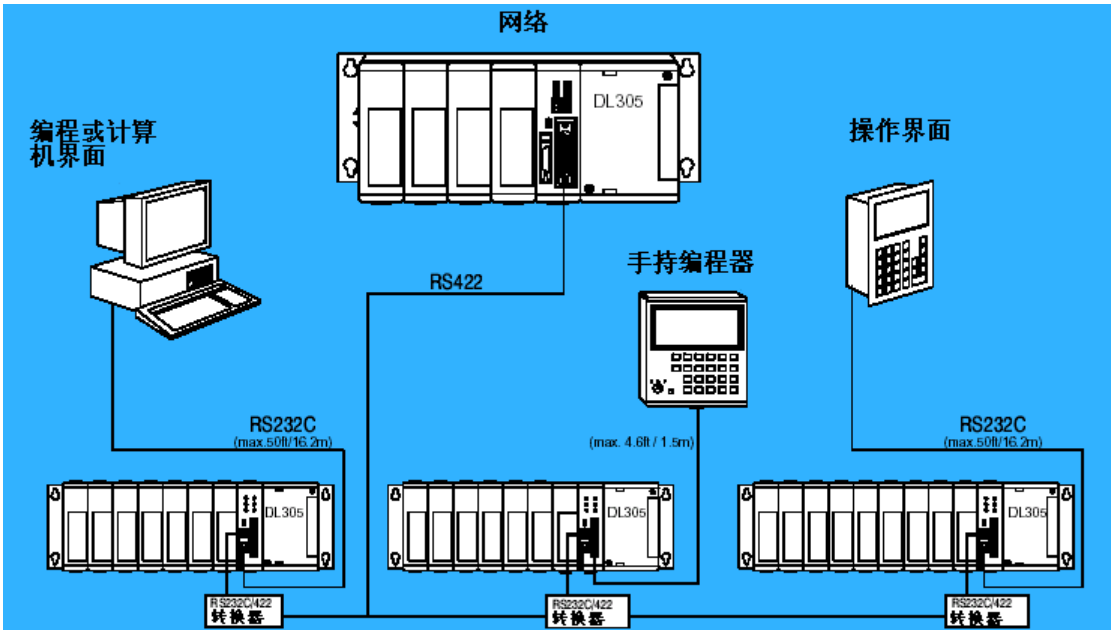
DL305 能用 DirectSOFT 来编程，它是工业领域里最先进的编程软件之一。

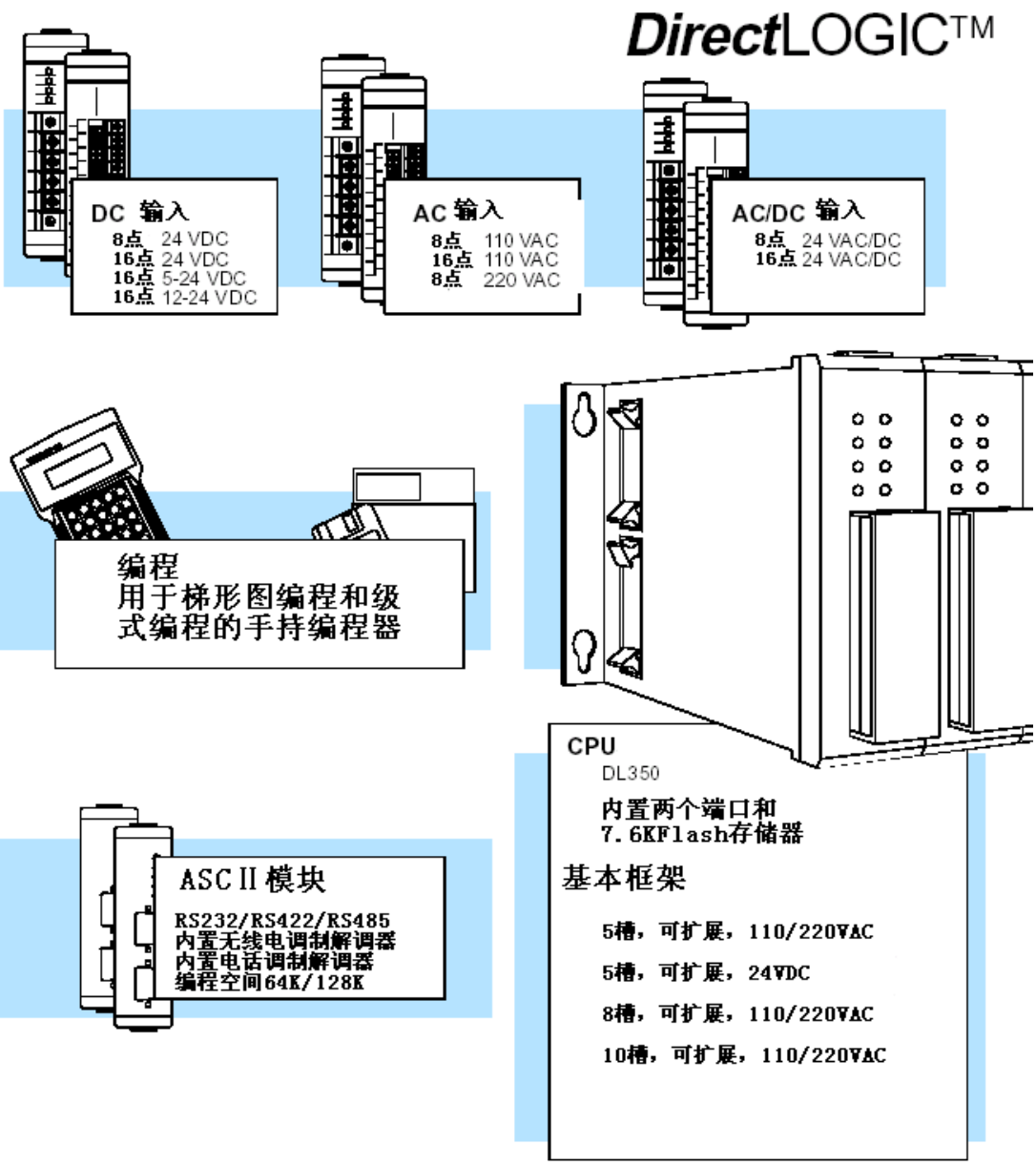
手持编程器

DL350 CPU 内置一个支持 DL205 所使用的编程器（D2－HPP）的编程口。该编程器可用来编写、修改和调试您的应用程序。为此您最好再看一下编程器使用手册。

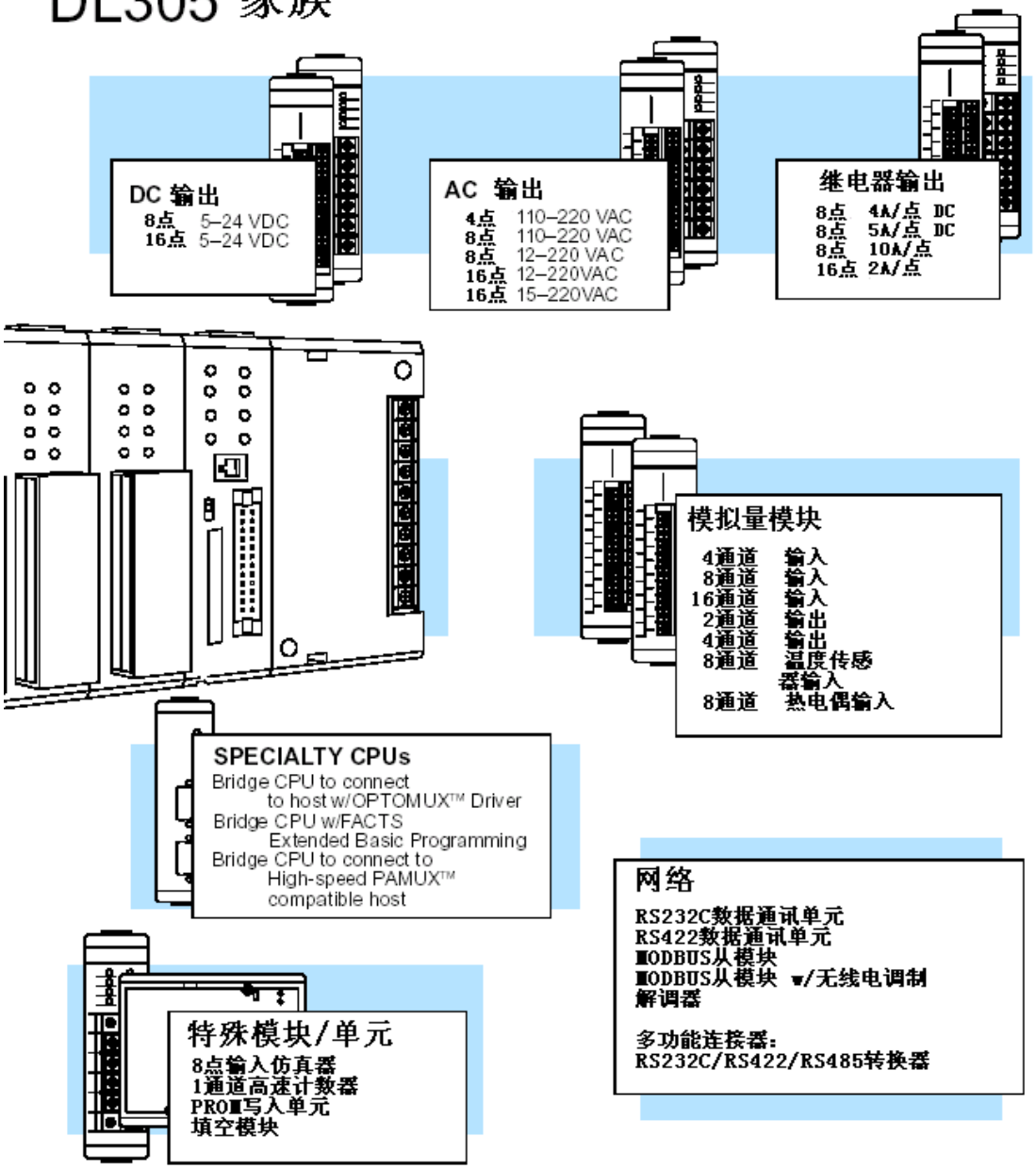
DL305 系统示意图 以下为 DL305 系统的主要部件及其配置方法的示意图。后面两页介绍了构建您的系统所需的一些特殊部件。






DirectLOGIC™
DC 输入
8点 24 VDC
16点 24 VDC
16点 5-24 VDC
16点 12-24 VDC
AC 输入
8点 110 VAC
16点 110 VAC
8点 220 VAC
AC/DC 输入
8点 24 VAC/DC
16点 24 VAC/DC
编程
用于梯形图编程和级式编程的手持编程器
CPU
DL350
内置两个端口和
7.6KFlash存储器
基本框架
5槽，可扩展，110/220VAC
5槽，可扩展，24VDC
8槽，可扩展，110/220VAC
10槽，可扩展，110/220VAC
ASCII模块
RS232/RS422/RS485
内置无线电调制解调器
内置电话调制解调器
编程空间64K/128K

DL305 家族


DC 输出
8点 5–24 VDC
16点 5–24 VDC
AC 输出
4点 110–220 VAC
8点 110–220 VAC
8点 12–220 VAC
16点 12–220VAC
16点 15–220VAC
继电器输出
8点 4A/点 DC
8点 5A/点 DC
8点 10A/点
16点 2A/点
模拟量模块
4通道 输入
8通道 输入
16通道 输入
2通道 输出
4通道 输出
8通道 温度传感器输入
8通道 热电偶输入
SPECIALTY CPUs
Bridge CPU to connect to host w/OPTOMUX™ Driver
Bridge CPU w/FACTS Extended Basic Programming
Bridge CPU to connect to High-speed PAMUX™ compatible host
网络
RS232C数据通讯单元
RS422数据通讯单元
MODBUS从模块
MODBUS从模块 w/无线电调制解调器
多功能连接器:
RS232C/RS422/RS485转换器
特殊模块/单元
8点输入仿真器
1通道高速计数器
PROM写入单元
填空模块

*Direct*LOGIC ™ 型号说明

当您在细查本手册时，会发现有许多不同类型等产品可供选用。一般来说要记住所有产品的详细情况是非常困难的。尽管如此，如果您花几分钟时间来了解一下产品的编号体系，您将会免去很多的麻烦。以下图表是不同种类产品的型号说明。电缆，电池，存储卡？等附件的零件编号是描述这些物品的典型缩写。

CPU 专用 CPU	
产品系列	D1/F1 D2/F2 D3/F3 D4/F4
CPU 种类/缩写	230…, 330…,430…
表示和类似模块有所不同	－1，－2，－3，－4

D4-	440DC	-1

基本框架	
产品系列	D2/F2 D3/F3 D4/F4
槽数	##B
插槽型号	DC 或无

D3-	05B	DC

离散 I/O 模块	
DL205 产品家族 DL305 产品家族 DL405 产品家族	D2/F2 D3/F3 D4/F4
点数	04/08/12/16/32
输入模块 输出模块 连接模块?	N T C
AC DC AC 和 DC 都有 继电器	A D E R
汇点 源点 汇点/源点	1 2 3
大电流 隔离 高速 I/O	H S F
表示和类似模块有所不同	－1，－2，－3，－4

D4 -	16	N	D	2	F

D3 -	16	N	D	2		-1

模拟 I/O 模块	
DL205 产品家族	D2/F2
DL305 产品家族	D3/F3
DL405 产品家族	D4/F4
通道数	02/04/08/16
输入（模数转换）	AD
输出（数模转换）	DA
连接模块	AND
隔离	S
表示和类似模块有所不同	－1，－2，－3，－4

F3-	04	AD	S	-01

其它模拟量模块的型号说明

F3-	08	THM	-n

注意：-n 表示热电偶的型号，例如：J,K,T,R,S 或 E

通信与网络 特殊 I/O 模块和设备 编程	
DL205 产品家族	D2/F2
DL305 产品家族	D3/F3
DL405 产品家族	D4/F4
名称缩写	见举例

D4-	DCM
D3-	HSC
D3-	HPP

DCM（通讯模块）
HSC（高速计数器）
HPP（级式手持编程器）

协处理器和 ASCⅡ？ 模块	
DL205 产品家族	D2/F2
DL305 产品家族	D3/F3
DL405 产品家族	D4/F4
协处理器	CP
ASCⅡ？	AB
64K 存储空间	64
128K 存储空间	128
512K 存储空间	512
无线电调制解调器	R
电话调制解调器	T

F4-	CP	128	-R

PLC 确认与编程的快速教程？

如果您曾经用过 PLC，或者想创建一个简单的例子，这一节将会教您如何来做。举这个例子并不是为了解释您创建系统所需的每样东西，只是为了向您提供一些使系统运作起来的基本示意图。

步骤 1：

打开 DL305 的包装并确认含有以下建立演示系统所必需的部件。

打开 DL305 包装

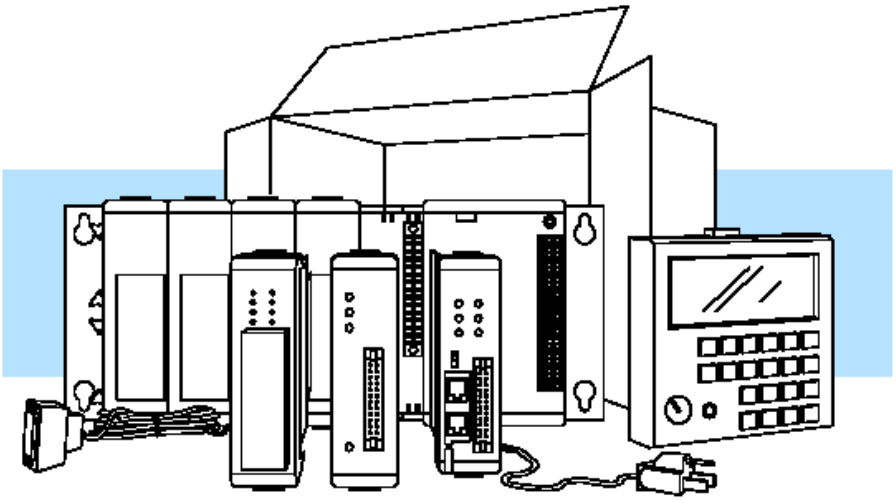
小部件如下：

- 基本框架
- CPU
- D3-08ND2 直流输入模块或 D3-08SIM 模拟输入模块
- D3-08TD2 直流输出模块
- *电源线
- *固定电线
- *A 24VDC 出发开关（如果不使用模拟输入模块）
- *一个螺丝起子，普通的或 Philips 型的

*此类物品不包含在 PLC 内

你必须选择以下两种编程方法之一来进行编程

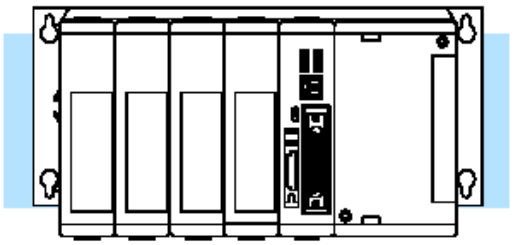
- DirectSOFT 编程软件，DirectSOFT 使用手册，和一根编程电缆（用来连接 CPU 和计算机）
- D2－HHP 手持式编程器和使用手册



步骤 2：安装 CPU 模和 I/O 模块

在基本框架上安装 CPU 模和 I/O 模块。CPU 模块必须插在第一个插槽（最靠近电源的地方）

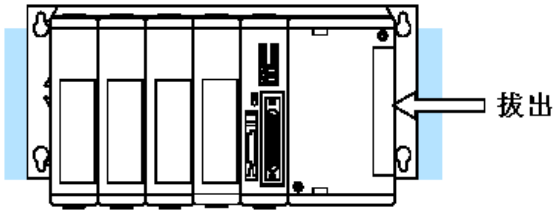
- 每个模块的顶部和底部都有一个塑料固定夹
- 把模块插入基本框架后，用上下两个引导部件来滑动
- 然后轻轻地向后拉，直到模块固定并且塑料固定夹也锁紧了



离散模块、模拟量模块、和继电器模块应该插在哪个插槽或哪个基本框架中并不重要。在本例中，将把输出模块插在 CPU 模块旁边，紧接着再插输入模块。其他模块的使用限制条件将在第四章介绍。您在系统配置时，必须保证不超过电源的最大供电能力。有关电源的参数也将在第四章中提到。

步骤 3：取下端子的盖板

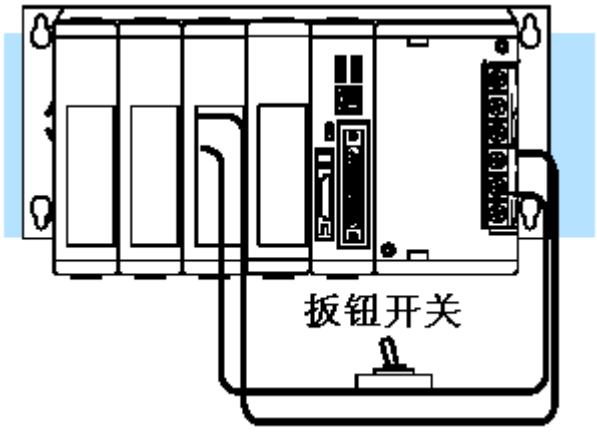
取下端子的盖板。此盖板是位于电源上的一个塑料片。



步骤 4：添加 I/O 仿真

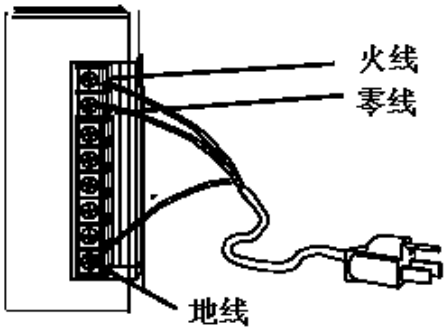
为了完成这个快速入门练习或学习本手册上的其他例子，你必须安装一个模拟输入的模块（或按下图所示连接一个开关），然后再安装一个输出模块。使用输入模拟器是检验一个系统或程序能否正常运行的最快的方法。任何一个离散的输出模块都可用来监测输出状态。

先把开关以及电源优先使用的一些场效应器件和系统先连接起来，从而来保证某个点在接线过程中没有被打开。如下图所示，把一个触发开关和一个输入模块（X0）连起来，并把 24V 辅助电源接到 CPU 的端子板上。第二章“安装、接线和详细说明”提供了 I/O 口的接线指导。



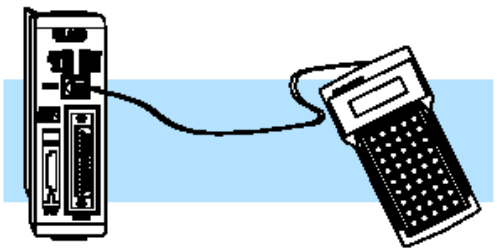
步骤 5：接电源线

如图所示接线。仔细阅读本手册在开头就提到的注意事项。有关接线步骤，详情请见第二章。当所有接线完成，把 CPU 和模块的盖板放回原处。注意暂且先不要通电。



步骤 6：连接手持编程器

用一根电缆把编程器连到 CPU 模块一个通讯口上（RJ 型的电话插座）。



步骤 7：打开系统电源

给系统上电并确认“PWR”指示灯是否点亮。如果没有，断开电源并检查所有接线是否正确。您可以参考第九章中“故障排除”一节相关内容。

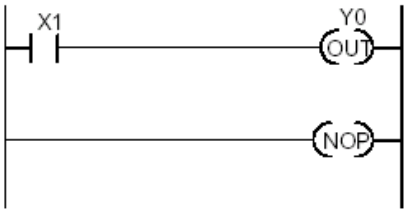
步骤 8：输入程序

把 CPU 模块上的模式开关先打到“STOP”位置，然后再打到“TERM”位置。此时 CPU 就进入了编程模式，我们就可以对 CPU 进行编程。

注意：没必要为您的系统特意分配 I/O 口，因为 DL305 CPU 会自动检测所安装的模块并进行正确的分配。

手持编程器按键顺序

$ STR	→	B 1	ENT
GX OUT	→	C 2	ENT

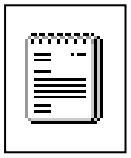


当输入一个简单的程序以后，把模式开关从“TERM”打到“RUN”然后再打回“TERM”。CPU 模块上的指示灯亮了就表示 CPU 已进入了运行模式。如果该指示灯没亮，请重复步骤 8 来确保程序输入正常，您也可以参考的第九章的“故障排除”。

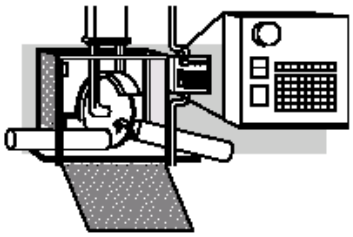
在运行过程中，输出状态指示灯能反映输出点的状态。当有输出信号时，相应的指示灯就会点亮。

设计一个成功的系统的步骤

步骤 1：
仔细查看安装向导

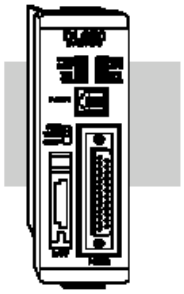


在任何系统的应用中，首先要保证的是安全问题。第二章的内容将会为您如何来建立一个更安全、更可靠的系统提供一些帮助。这一章还讲述了系统的各个部件是如何接线的。



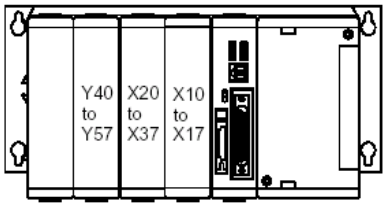
步骤 2：
理解 CPU 的组装程序

CPU 是一个自动化系统的心脏。确保您已经花时间了解了各种特性和所需组装的设备。



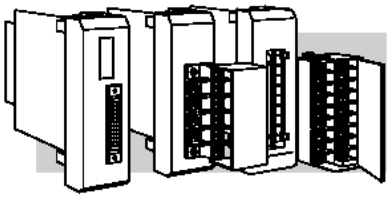
步骤 3：
理解 I/O 系统的配置

了解您的本地 I/O 口的设置是非常重要的。同时，电源负载大小的计算也是非常重要的。这些都会影响到您对 I/O 模块的放置及相关的设置。



步骤 4：
I/O 模块的选择接线描述

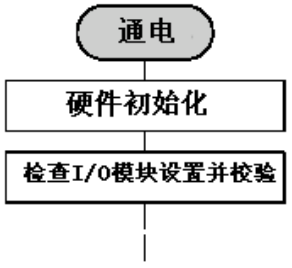
DL305 系统可以支持多种不同的 I/O 模块。第二章提供了离散 I/O 模块的详细介绍和具体接线图。



注意：本手册不包含一些特殊模块的资料，它们由专门的手册来介绍

步骤 5：
理解系统操作

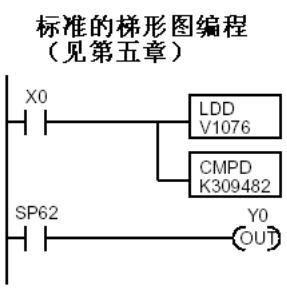
如果在您输入一个程序前就知道 DL305 系统是如何工作的，那将对您是非常有用的。这不仅涉及到程序的执行步骤，还涉及到操作模式和存储器的配置。详情请参考第三章。

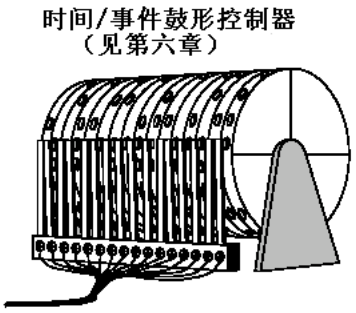


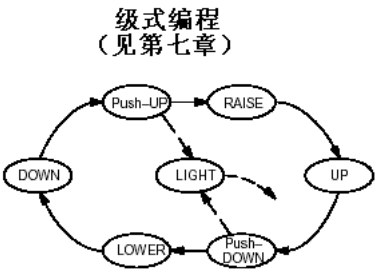
步骤 6：复习一下编程概念

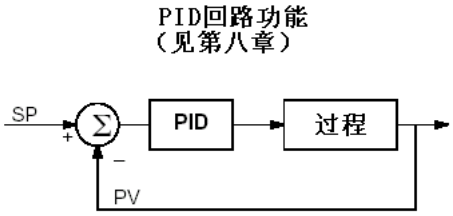
DL305 提供四种途径来来帮您解决编程问题，包括下面讲到的 PID 回路整定。

- 梯形图编程法（RLL）是用来解决一些布尔逻辑和通用的 CPU 的寄存器/累加器操作的最好的方法。它支持许多指令，包括鼓形控制指令，级指令和循环控制指令。
- DL350 有 4 个定时器/事件鼓形控制器，每个最多能 16 步。它们能提供基于时间和事件的步进控制器。鼓形控制器最适用于一个重复性的过程，因为它采用步进控制。？
- 级式编程方法（也称为 RLL^{+} ）是基于级的变换图。级式图把梯形图划分为几个部分，这有点类似于流程图中的级。
- DL305 的 PID 回路操作使用了设置表来设置 4 个回路。主要特性包括：自动调节，报警，设定值 ？/ ？生成器等。



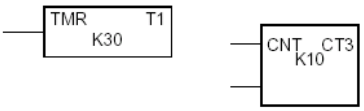






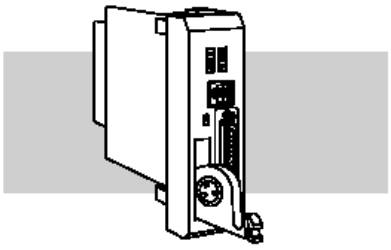
步骤 7：选择指令

一旦您已经安装好了系统并清楚了其工作原理，您就可以从功能最强大的指令集中选择你所需要的。



步骤 8：理解维护和故障排除的程序

设备在任何时候都有可能会发生故障。比如开关失灵，电池没电等。在大多数情况下，排除故障的主要时间都花在对故障的定位上。DL305 系统内置了许多功能可以帮助您快速地找到故障。有关诊断和故障排除的详情请参考第 9 章。



第二章　安装，接线和详细说明

2

本章将介绍…

——安全指南

——安装指南

——安装 **DL305** 的基本框架

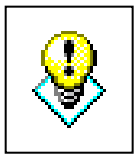
——安装基本框架的组件

——**I/O** 安装策略

——**I/O** 模块的位置，接线和详细说明

——详述语句的术语表

安全指南



警告：保证有一个对人员和设备都相对安全的工作环境，不仅是您的责任，也是您需要在系统的设计和安装过程中首先考虑的。自动化系统有时会发生一些故障，这些故障有可能会导致人员受伤或者设备损坏。不要指望自动化系统自己会提供一个安全的操作环境。您应该使用一些外部机电设备，例如：继电器，限位开关等。这些元件虽然独立于 PLC 系统之外，但可以用来保护有可能会产生人员伤亡或设备损坏的系统的任何一个部分。

每一个自动系统都是不同的，根据应用的不同，您可能会有一些特殊的需求。所以必须确保您的操作符合当地的安装及操作规范。

安全计划

要提供一个安全的操作环境，最好的方法就是把人员及设备的安全写入安全计划中。您最好仔细检查系统等每一个环节从而了解哪些部分对人和机器的安全是尤为重要的。

如果您对 PLC 系统的安装没有多少实际的经验，或者贵公司还尚未制定出安装指南，您最好通过一下三个来源来获取额外的信息。

- NEMA——国际电气制造业协会(The National Electrical Manufactures Association)，坐落于美国华盛顿特区，出版许多有关工业控制系统标准的文献。您可以直接向 NEMA 定购一些出版物，例如：
  ISC1,工业控制与系统通用标
  ISC3,工业系统
  ISC6,工业控制系统外壳？
- NEC——全国电气规程（美）（The National Electrical Code ），它提供了许多有关安装和使用各种不同电气设备的规则。一般您可以从电气设备销售人员或当地图书馆得到 NEC 手册的拷贝。
- 当地的机构——许多市政府和州政府除了 NEC 手册上的要求以外还有其它的一些要求。

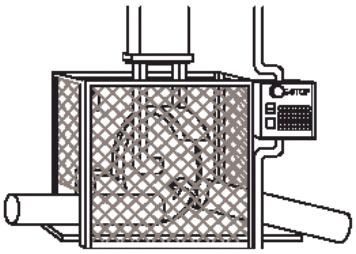
安全技术

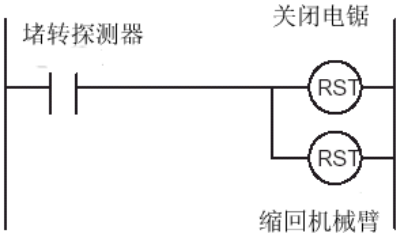
上述出版物提供了许多维护系统安全的方案和设备。使用下面的技术将有助于您减少安全问题的出现。

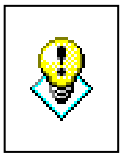
- 在 PLC 控制程序中要设计合理的系统断电顺序
- 系统断电方法（限位开关，紧急制动开关等）

依次关闭系统

第一级保护是由 PLC 的程序通过机械故障识别来完成的。分析您的实际应用并确定正确的断电顺序。一些典型的问题，例如零件的卡住或丢失，空箱等，都会产生人员受伤或设备损坏的风险。







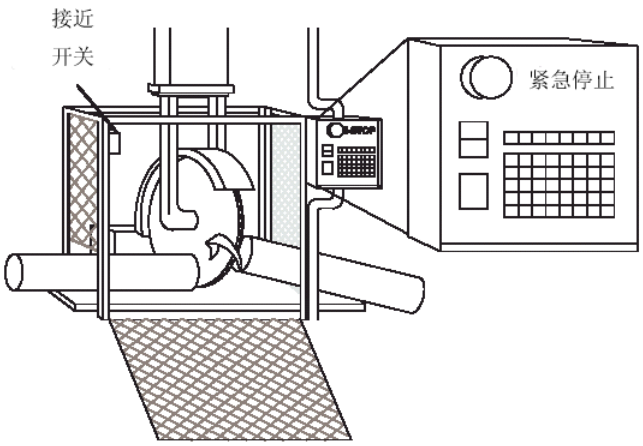
警告：控制程序决不能成为唯一的保护人身安全和设备安全的方法。

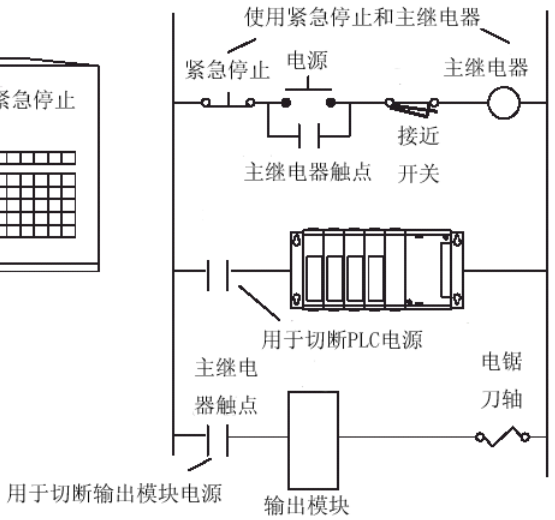
切断系统电源

通过某些机电装置，比如主控继电器和限位开关，您可以避免机器的误启动。只要安装正确，此类装置可以避免任何的机械误操作。

例如，机器的某个部分被卡住了，PLC 的控制程序可以关闭电锯并缩回刀轴。尽管如此，由于操作员会亲自打开隔离网来移走部件，您还必须使用一个接近开关。当隔离网一打开，就切断整个系统的电源。

同时，操作者也需要一个顺手的手动开关来切断所有系统电源。这可以通过使用一个标有“紧急停止”的开关来实现。





在紧急停止或者任何其它形式的电源中断后，也许会产生一些必须解决的问题。只有解决了这些问题，机器才能重新启动。例如，某个特殊寄存器的值必须被改变（或者维持断电以前的状态），系统才能重新工作。在这种情况下，您也许要用到控制程序中的停电记忆区域，或者常数来确定起始工作点。

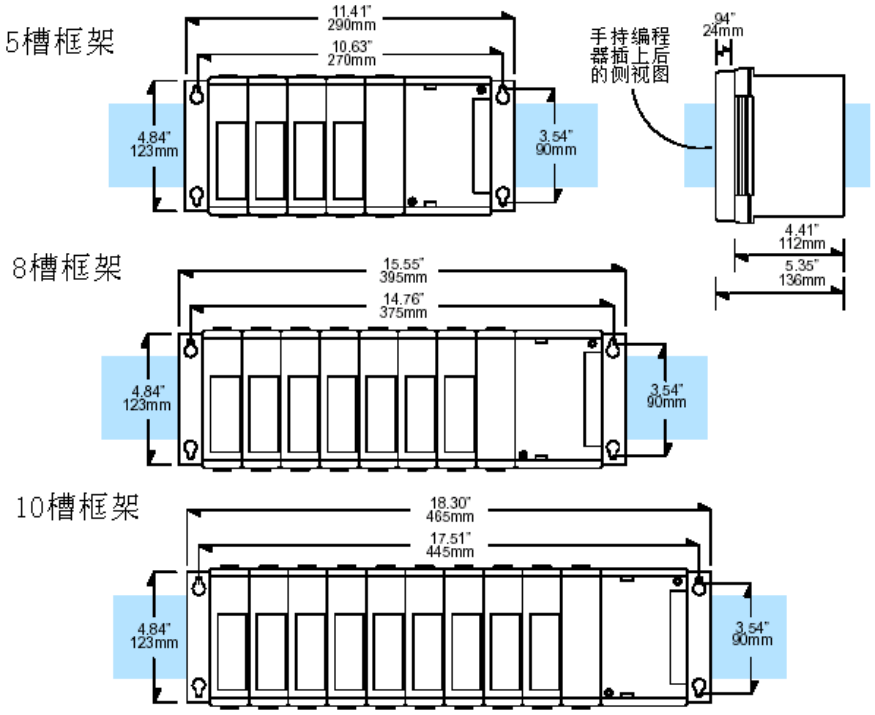
安装指南

在安装 PLC 系统之前,您还必须了解系统中用到的组件的尺寸。组件的尺寸将在下面的示意图中表示出来并且可以作为决定外壳规格的依据。记得要为以后的扩展设备留有一定的空间。

注意：如果您在系统中还用到其它的组件，请查阅相应的手册来了解它们的具体尺寸。

基本框架尺寸

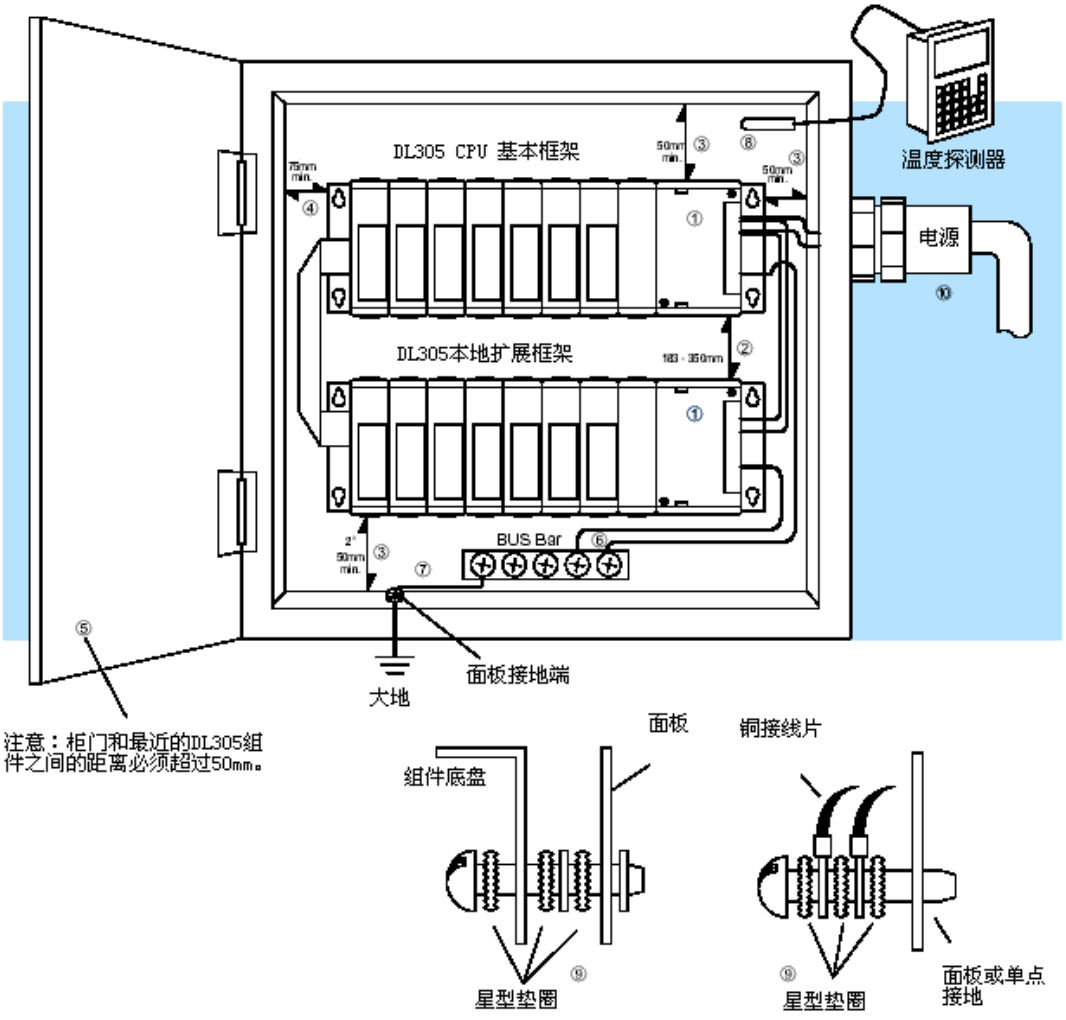
以下是有关安装尺寸的信息。所有基本框架的高度都是一致的。其深度由您所选的 I/O 模块来决定。基本框架的长度根据槽数的增加而增加。请确认您已经按照安装指南来留出空间了。



面板安装与布线

正确地设计面板将有助于确保 DL305 能工作在环境允许的条件之下，这一点是非常重要的。系统的安装应该按照所有相应的电气规范和标准。您的系统还应该符合实际应用的操作标准，从而确保运行正常。

1. 基本框架必须水平安装从而使通风顺畅。
2. 相邻两个基本框架之间的距离必须至少 183mm。
3. 基本框架和柜子的任何一部分之间必须最少有 50mm 的距离。基本框架和任何布线管之间的距离必须在 30mm 以上。
4. 柜门和 DL305 的组件之间至少要 50mm



5．DL305 基本框架上的接地端必须单点接地？。应该使用铜导线可以实现低阻抗要求。铜接线片？应该被弄弯然后焊接到导线的末端，这样才能保证有一个良好的接触面。您还必须去除阳极电镀？然后使用铜接线片和星型垫圈来接到段子上。通用的规则是为了使 DL305 和单点接地之间的直流电阻小于 0.1Ω 。

6．控制柜中所有需要接地返回？的设备都必须要有单点接地（例如铜的总线？）。单点接地必须接到控制柜的地上。

控制柜的接地端必须和大地相连。此连接必须至少使用 AWG12 的铜导线。最小线径，颜色的选择，通用安全实际操作都要按照当地的电气规范和标准来决定。

一个好的通用的参考地？（大地）对 DL305 的正确操作非常重要。DL305 系统及其组件都是用的同一个参考地。有几种方法可以用来提供足够的参考地，例如以下两种：

1）尽可能在靠近控制柜的地方安装一个接地棒。

2）接到电源地上

7．安装时还要考虑到周围的温度变化是否会超过允许的范围。要做到这一点只要在控制柜中放一个温度探测器，然后关上门让系统运行，直到稳定下来。如果周围的温度已经超过了 DL305 系统允许的范围，必须采取一些措施来使 DL305 系统的环境温度达到要求指标，比如安装一些加热/冷却装置。

8．设备安装螺钉以及接地？螺钉应该用#10 铜螺钉或同等零件。只要有可能，应该采用？孔而不是螺钉螺母。要保证有良好的接触，应该清除终端部位？接线处的杂物，如油漆，涂料，腐蚀物等。

9．DL305 系统的电源支持 110VAC，220VAC 和 24VDC，因此它可以贯穿于某个工业环境中使用。隔离变压器和降噪装置一般没必要使用，但它们对消除或降低电源不稳定的问题是有一定作用的。

外壳

选择一个良好的外壳将有助于您组成一个工作良好、安全可靠的 DL305 系统。根据 DL305 系统的不同应用，也许还需要一些额外的特性？外壳应至少包含以下要求：

- 符合电气标准
- 应避免不良的工业环境因素
- 公共参考地？
- 维持周围的温度
- 能够访问设备？
- 访问限制
- 提供足够的安装及维护设备的空间

环境技术要求

以下表格列出了 DL305 系统（包括 CPU,基本 I/O 模块）的通用环境技术要求。您的应用和周围温度的变化会使模块工作不稳定。要了解各种模块的特性曲线可以参考有关介绍 I/O 模块的章节。

技术要求	范围
储存温度	－20℃～70℃
工作温度	0℃～60℃
湿度	5%～95%相对湿度（不结露）
抗振	美国军用标准 810C, 方法 514.2
耐冲击	美国军用标准 810C, 方法 516.2
抗扰度	NEMA(ICS3-304),1μ S 方波
空气	无腐蚀性气体

*手持编程器和 DV-1000 的工作温度应该在 0℃～50℃

手持编程器和 DV-1000 的存储温度应该在－20℃～70℃

**设备必须工作在湿度 30%以下的环境中。尽管如此，在较低的湿度下一些静态的电气问题出现的频率要高一点。保证在触摸设备以前做好充分的防范措施。在低湿度的环境中，可以考虑使用接地带，防静电底板等物品。

电源

电源必须符合基本框架电源部分的电压和电流要求。

规格	D3-05B－1	D3-05BDC	D3-08B	D3-10B
输入电压范围	85~264VAC 47~63Hz	20.5~30VDC<10% 纹波	85~264VAC 47~63Hz	85~264VAC 47~63Hz
基本框架功率消耗	最大 85VA	48W	最大 85VA	最大 85VA
最大瞬间电流	30A	30A	30A	30A
绝缘耐压	1500VAC 持续 1 分钟，电源接线端子，运行输出，公共端，24VDC 之间	1500VAC 持续 1 分钟，在 24VDC 输入和运行输出之间	1500VAC 持续 1 分钟，电源接线端子，运行输出，公共端，24VDC 之间	2000VAC 持续 1 分钟，电源接线端子，运行输出，公共端，24VDC 之间
绝缘电阻	>10MΩ ,500VDC	>10MΩ ,500VDC	>10MΩ ,500VDC	>10MΩ ,500VDC
电源输出（电源范围和波动）	(5VDC)4.75~5.25V 小于 0.25V p-p (9VDC)8.0~10.0V 小于 0.45V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.25V p-p (9VDC)8.5~13.5V 小于 0.45V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.25V p-p (9VDC)8.0~10.0V 小于 0.45V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.25V p-p (9VDC)8.0~10.0V 小于 0.45V p-p (24VDC)20~28V 小于 1.2V p-p

机构认可

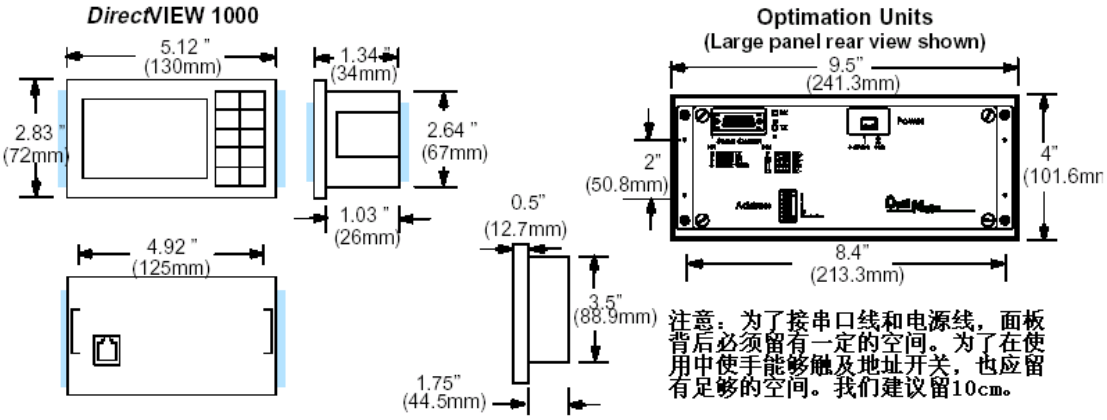
某些应用是需要机构认可的。典型的机构认可有以下几种

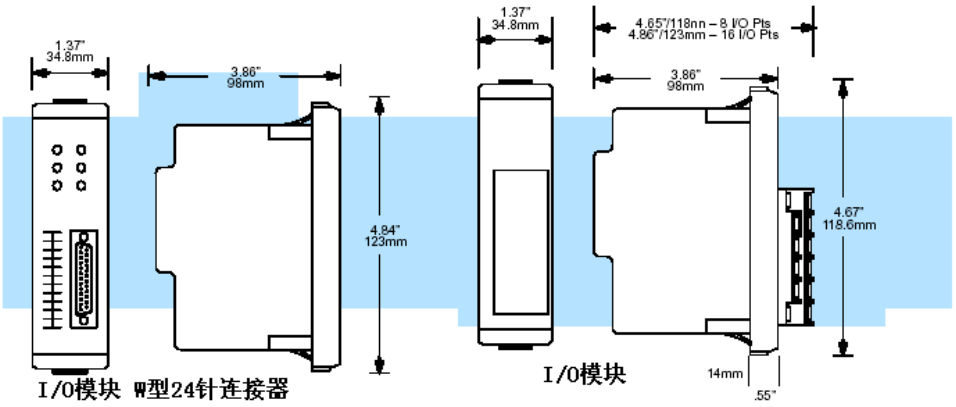
- UL（美国保险商实验所）
- CSA（加拿大标准协会—Canadian Standard Association）
- FM （工厂相互研究组织？）
- CUL（加拿大保险商实验所）

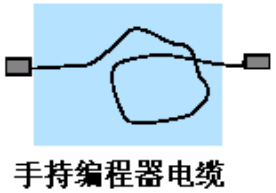
组件尺寸

在安装 PLC 系统之前,您还必须了解系统中用到的组件的尺寸。组件的尺寸将在后面的示意图中表示出来并且可以作为决定外壳规格的依据。记得要为以后的扩展设备留有一定的空间。附录 E 提供了各种组件的重量。

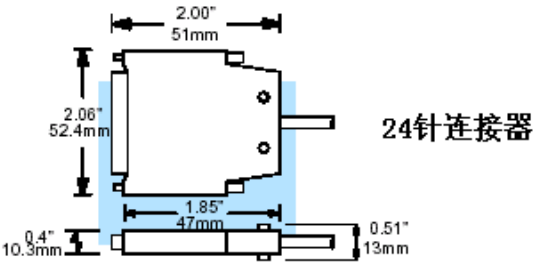
注意：如果您在系统中还用到其它的组件，请务必参考相关的手册来使得安装正确。







手持编程器电缆

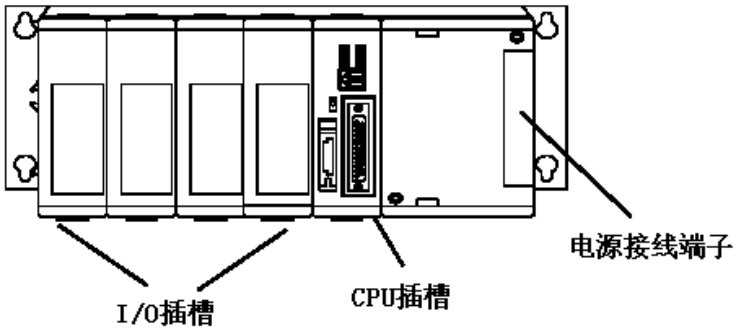


安装 DL305 基本框架

选择基本框架的型号

DL305 系统提供了三种不同型号的基本框架和两种电源。

下图是一个 5 槽框架的例子。

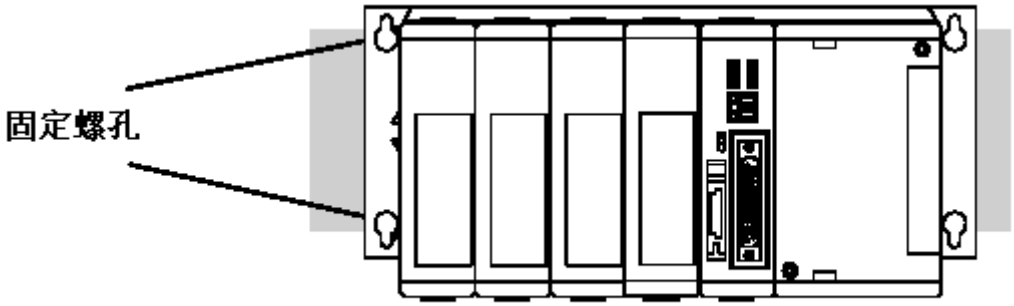


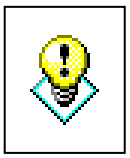
您可以根据以下三点来选择基本框架：

- 所需要的 I/O 模块数
- 电源输入要求（直流电源或交流电源）
- 允许的电源容量

安装基本模块

所有的 I/O 设置都要用到任何一个基本设置。由于在角上使用了 4 个 M4 螺丝钉，基本框架安装在设备上会很稳固。完整的尺寸在前一节中介绍过。

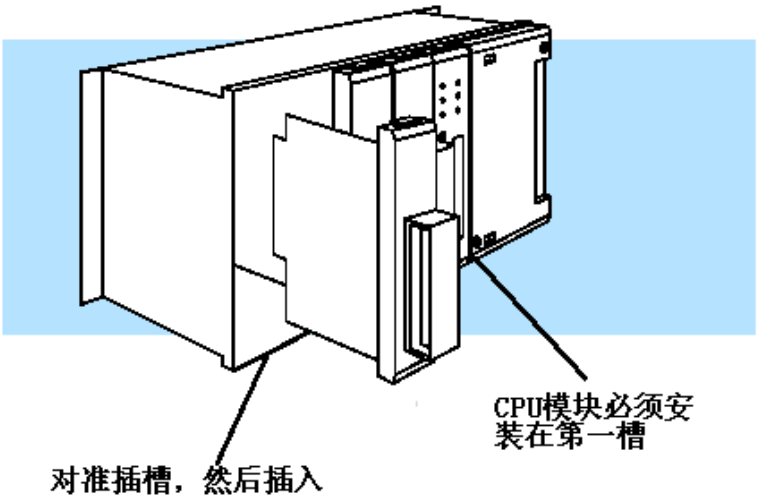


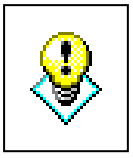


警告：为了降低触电，人身伤害或设备损坏的可能性，安装和拆卸任何部件时首先断电。

在基本框架中安装组件

在进行模块的安装时，把模块的 PCB 板对准框架的凹槽，然后垂直插入框架中直到与底板接触牢固可靠。一旦模块插入框架中，按下固定塑料夹（位于模块的底部和顶部）来保证插到位。





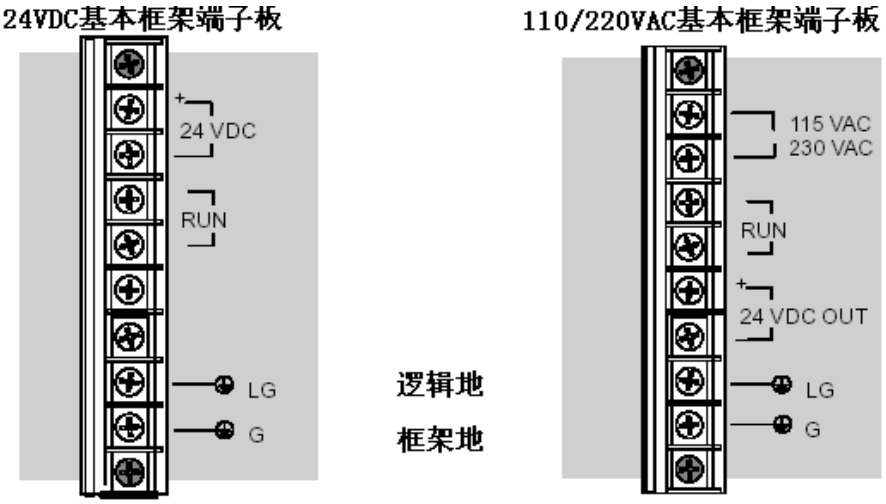
警告：为了避免触电，人身伤害或设备损坏，安装和拆卸任何部件时首先应断电。

基本框架接线指南

基本框架的接线

以下是位于 DL305×××××－1 基本框架电源上的接线端子示意图。基本框架能最大支持 16AWG 规格的电线。

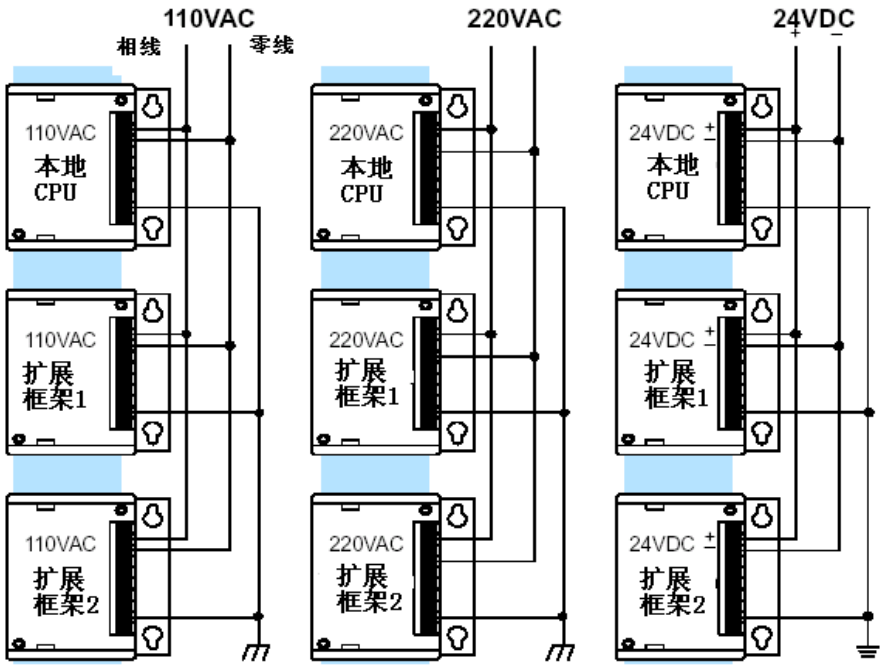
注意：您可以把 115VAC 或 220VAC 直接接到到电源上。当和其它的 *Direct*LOGIC ™产品一起使用时，没必要进行特殊的接线或跳线。



警告：接好电源线后，一定要装上保护盖。当拿走此保护盖时，如果您不小心用手碰到端子，将会有触电的危险。

扩展框架的接线

以下例子演示了扩展框架的接线方法。

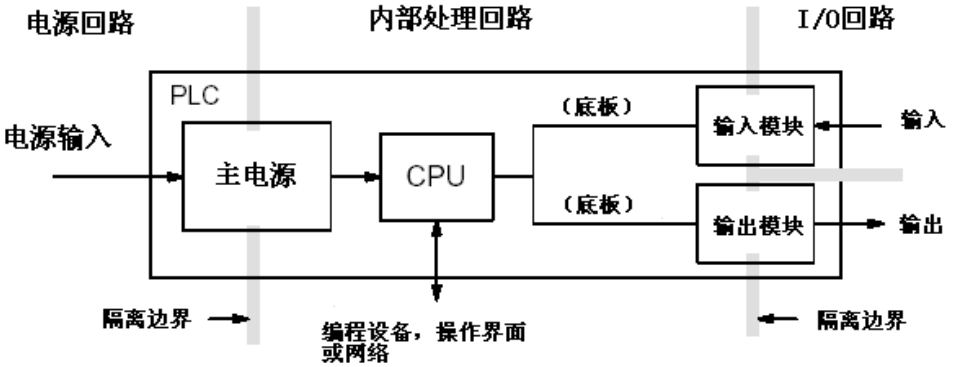


I/O 接线策略

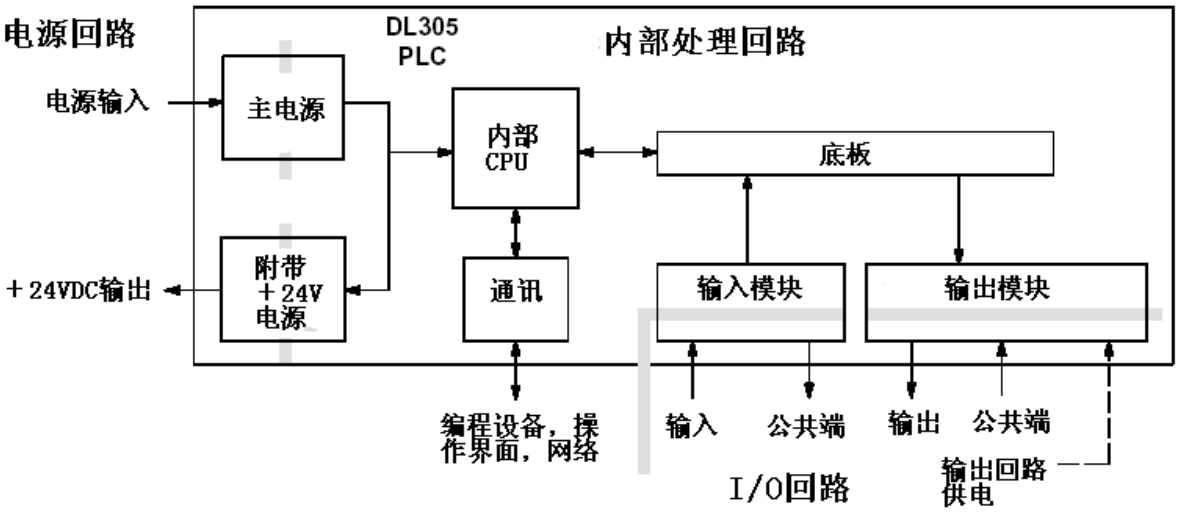
DL305 系统是非常灵活的，它能工作于不同的接线设置中。在实际操作前，通过学习这一节，您也许会发现在您的应用中最合理的接线策略。这可以帮您减少系统花费，接线错误和避免安全问题。

PLC 隔离边界

如下图所示，PLC 内部回路被隔离边界分为三个独立的区域。电气上的隔离能提供一定的安全性，所以，当某个区域出了问题，它不会损坏其它的区域。电源内部的变压器通过磁感应的方式把初级和次级隔离。光耦器件在信号的输入和输出时提供光隔离。此隔离把与现场装置直接相连的 I/O 回路和内部处理回路隔离起来。请注意输入点和输出点之间也是隔离的，因为它们都已经分别和处理回路隔离开来。隔离边界可以在电源接错或 I/O 接线错误时保护操作界面或者操作人员。当给一台 PLC 接线时，千万要防止内部处理回路和其它回路相连接。



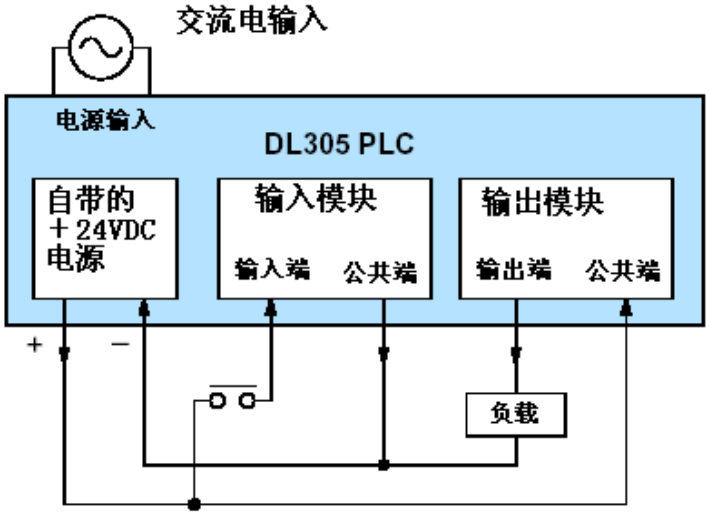
以下是 DL305 系统的布局图。除了给自身供电以外，基本框架在保证隔离的情况下还能向外提供一个＋24VDC 的电源。由于电源回路是和其它三个回路是隔离的，因此它能给 I/0 回路供电。



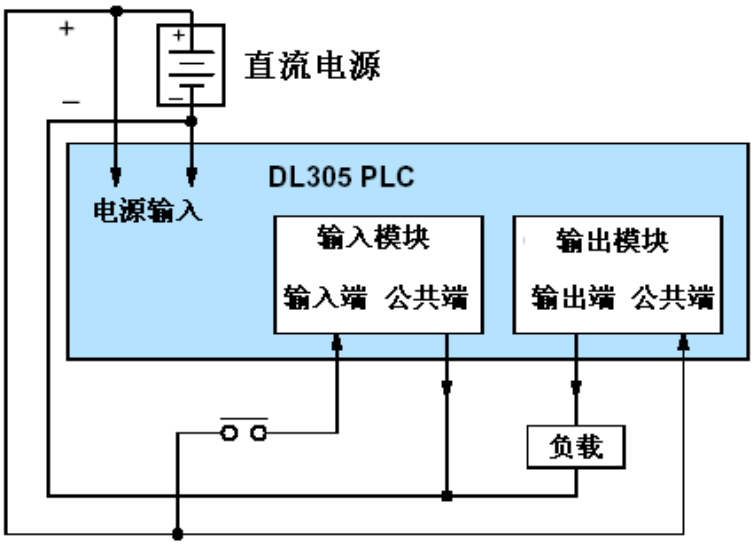
在一些情况下，使用 PLC 内置的＋24VDC 电源可以帮您节省费用。它能提供最大 100mA 的负载。使用时务必注意不要让电源过载。如果您是系统的设计者，您可以放心使用需要＋24VDC 供电的现场装置。

用 24VDC 给 I/O 回路供电

自带直流输出电源，这是所有采用交流供电的 DL305 基本框架的一个特点。如果输入设备和输出负载都需要＋24VDC 电源，PLC 自带的直流输出电源可以同时给两者供电。如下图所示：



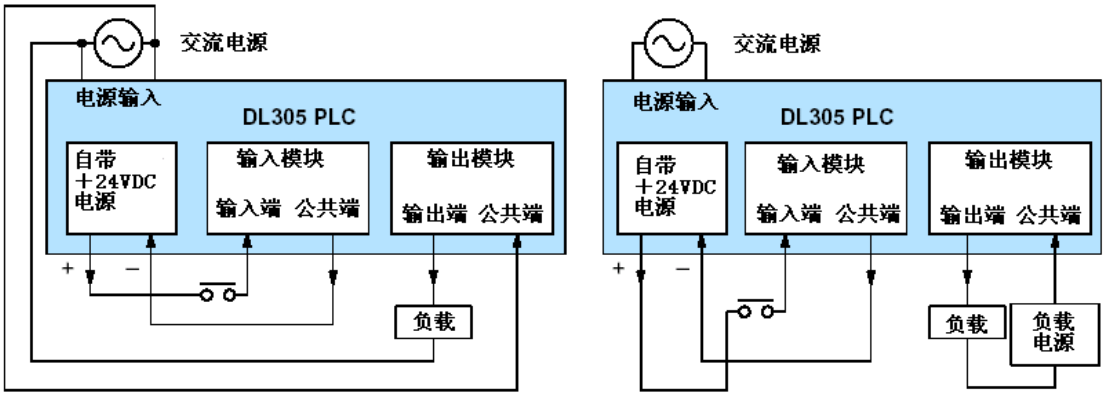
直流供电的 DL305 基本框架用于一些特殊的场合，比如直流电源比交流电源更容易得到。这些包括由许多电池组成的宽电压范围的场合，比如可移动的控制，车辆上的使用，便携式设备等。像在此类应用中，所有的输入装置和输出负载都使用同一个直流电源。



给 I/O 回路单独供电

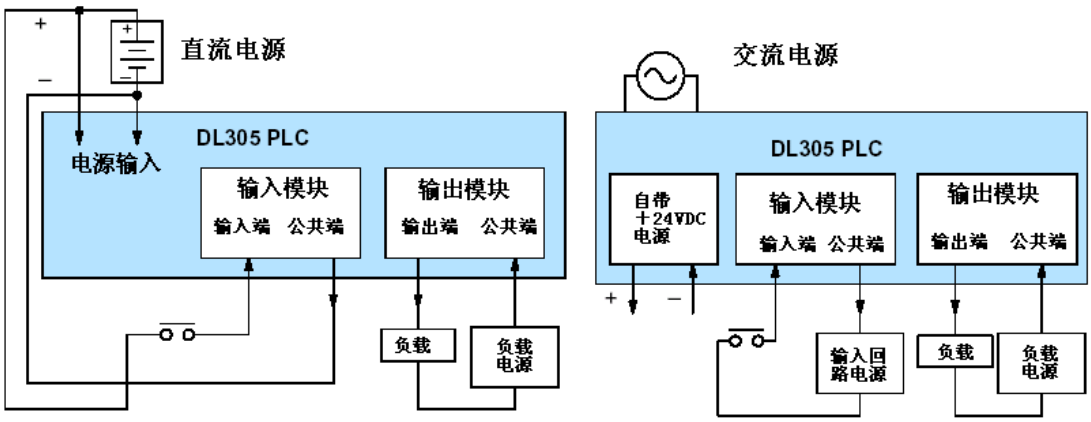
在大多数的情况下，有必要给输入设备和输出负载分别供电。输出负载经常需要高电压的直流电源，而输入传感器一般使用低电压的直流电。如果一个机器操作人员很有可能接触到输入接线，那么从安全方面来考虑，需要把它们和高电压的输出回路相隔离。如果负载能使用和 PLC 一样的电源，而且输入传感器能使用 PLC 自带的 24VDC 电源，那将是非常方便的。如下面左图所示。

如果负载不能由 PLC 外部电源供电，必须再使用一个独立的电源。如下面右图所示。



在一些情况下，输入回路和基本框架使用同一个电源。这主要用于直流供电的 PLC 上，如下面左图所示。此时，PLC 和输入回路使用同一个电源，而输出回路另外使用一个独立的电源。

从花费和复杂程度来考虑，最坏的情况就是 PLC,输入设备，输出负载都需要各自独立的电源。下面右图表示了如何来接线，但自带的直流电源输出没有用到。您应该尽量避免这种情况的发生。



汇点/源点的概念

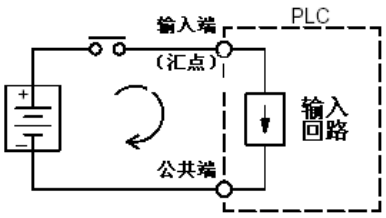
在深入学习接线策略之前，您还必须充分理解“汇点”和“源点”的概念。在讨论输入输出回路的时候会经常用到这两个词汇。编写本节的主要目的就是要让您很容易地理解这两个概念并进一步确保您的安装成功。先给出简短的定义。

汇点=提供一条与电源**负极**（—）相连的通路

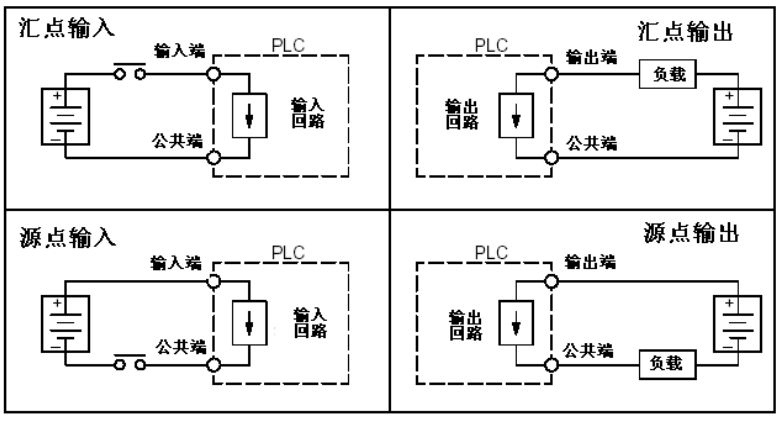
源点=提供一条与电源**正极**（+）相连的通路

首先，您会发现这两个概念只和直流回路有关而和交流回路无关，因为它涉及到正极和负极。所以这两个术语只能应用于直流的输入输出回路。指定为源点或汇点的输入点或输出点只能向一个方向传导电流。这意味着有可能把外部电源和现场设备接到 I/O 点上使得电流方向发生错误时，电路将不工作。尽管如此，理解了“源点”和“汇点”的概念以后，您能够每次都把电源和现场设备接对。

例如，右图就描述了一个“汇点”输入。要把外部电源正确地接好，您必须使输入提供一条与电源负极（—）相连的通路。电流从 PLC 的输入端流入，经过输入回路，再从公共端流出， 电源负极与公共端相连。在电源正极和输入端之间添加一个开关，整个电路就完成了。当开关闭合时，电流方向如图中箭头所示。

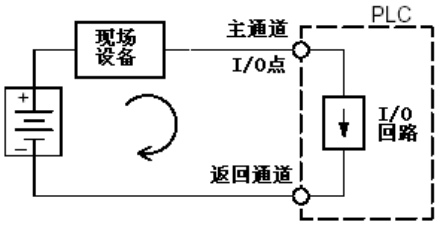


下图给出了汇点/源点的输入/输出电气原理图。本章的结尾部分将介绍 I/O 的规格和输入输出类型。

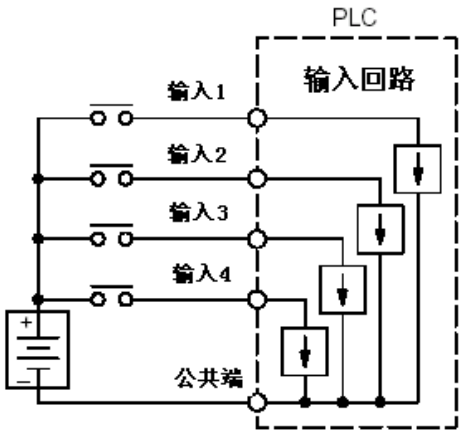


I/O 公共端的概念

一个 PLC 的 I/O 回路要工作起来，必须让电流从一个端子流入从另外一个端子流出。因此，每个 I/O 点需要两个端子。如右图所示，输入输出端子是电流通过的主要回路。要使电流返回电源负极，还必须要有额外的一个端子。



如果空间和预算都不受限制，每个 I/O 点都需要用专门的两个端子，如右面上图所示。然而，对大多数情况来说，这种方式没必要也不实际。因此，大多数的 PLC I/O 回路都采用了分组的方法，每组共用一个返回通道，即所谓的公共端。右图是 4 个输入点共用一个公共端的例子。这样，4 点的输入只需要 5 个端子，而不是原来的 8 个。



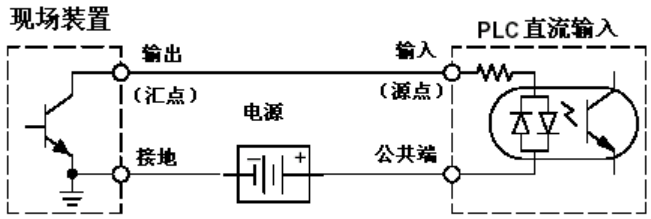
注意：在上面的电路图中，当 4 个点都有输入时，公共端上流经的电流是任何一个输入通道的 4 倍。这一点对输出回路尤为重要，公共端通常都需要更大规格的线。

直流 I/O 点和“固态”现场装置的连接

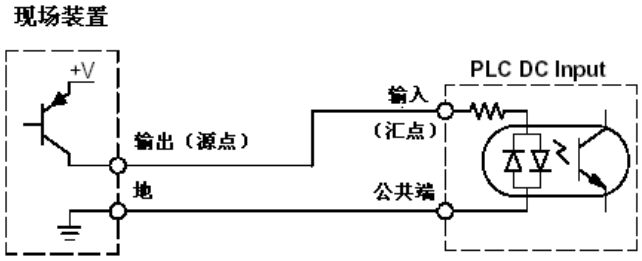
在前一节中提到了源点和汇点的概念，直流 I/O 回路有时候只能允许电流从一个方向流过。这对一些有固态特性（晶体管特性）的现场装置也是一样的。换句话说，现场装置也有源点和汇点的概念。当把两个现场装置接到一个直流回路中时，必须一个作为源点来接另一个作为汇点来接。

固态输入传感器

一些 DL305 的直流输入模块由于能够检测电流方向，使用起来非常方便，可以既作为源点来接也可以作为汇点来接。在下面的电路图中，其现场装置有一个集电极开路的 NPN 型晶体管输出。它能汇聚从 PLC 输入点（源点）流出来的电流。电源可以是 PLC 自带的 24VDC 电源，也可以是其它的电源（＋12VDC 或＋24VDC），只要输入特性相匹配。

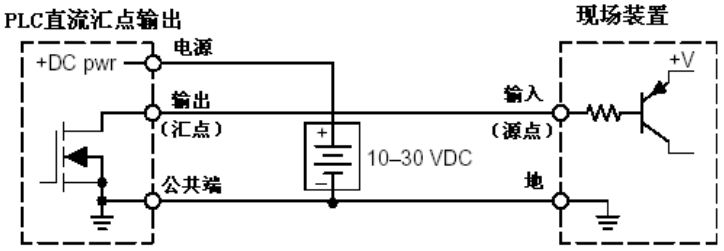


在下一个电路中，其现场装置有一个发射极开路的 PNP 型晶体管输出。工作时电流源于其输出，经过 PLC 的输入点汇于电源地。因为该现场设备是源点输出，所以没必要再另加电源。

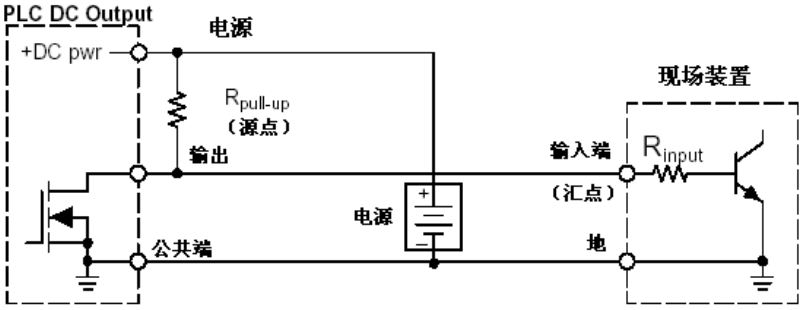


固态输出负载

在某些应用场合需要把 PLC 的输出点和固态输入的装置相连。这种类型的连接通常用于传送小信号，而不能用来传送驱动执行部件的直流电。一些 DL305 直流输出模块是汇点输出的。这意味着每个输出点在动作的时候提供了一条连着电源地的通路。在下面的电路图中，PLC 的输出点在动作的时候汇电流于电源地。与它相连的现场输入装置正好是源点输入的。



在下面的例子中，PLC 的直流汇点输出接到现场装置的汇点输入上。这是需要技巧的，因为 PLC 的输出点输出点和现场装置的输入点都是汇点型的。在回路中，必须一个是汇点型的另一个是源点型的，因此要通过使用一个上拉电阻来给 PLC 的输出点成为源点型的。在下面的电路图中，$R_{pull-up}$ 接在输出点和直流输出回路电源之间。



注意 1：不能通过上拉电阻的方式来驱动一个大的负载（>25mA）

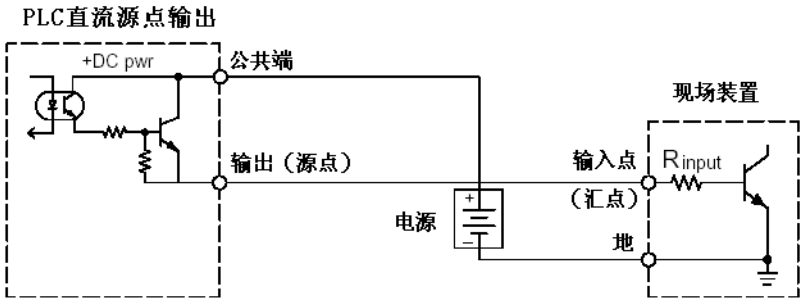
注意 2：使用上拉电阻的方式来获取源点将会颠倒输出点的逻辑。换句话说，从梯形图的角度来看，当 PLC 的输出点是 OFF 时，现场装置才动作。您在进行梯形图编程的时候，应该改变输出形式。或者，您也可以改变其它地方来消除这种效果，比如更改现场装置。

选择合适的 $R_{pull-up}$ 也是十分重要的。您首先需要知道现场装置动作时的工作电流。若没给出该值，您可以按下面的式子进行计算（典型的值是 15mA）。然后通过 I_{input} 和外部电源值来计算出 $R_{pull-up}$。接着计算 $R_{pull-up}$ 的功率从而选择合适的电阻。

$$I_{input} = \frac{V_{input\ (turn-on)}}{R_{input}}$$

$$R_{pull-up} = \frac{V_{supply} - 0.7}{I_{input}} - R_{input} \qquad P_{pull-up} = \frac{V_{supply}^{2}}{R_{pullup}}$$

当然，最简单的驱动一个汇点输入型的现场装置的办法就是使用直流源点输出模块，如下图所示。达林顿 NPN 型的管子有 1.5V 的导通饱和电压，但这对低电流的固态负载不成问题。



继电器输出指南

DL305 的 I/O 家族带继电器输出的有以下 4 种：D3-08TR, F3-08TRS-1, F3-08TRS-2, D3-16TR。继电器最适用于以下场合：

- ●驱动负载所用的电流晶体管输出无法满足
- ●低预算的场合
- ●一部分输出通道需要和其它输出点隔离（比如一些负载需的电压与其它输出点不同）

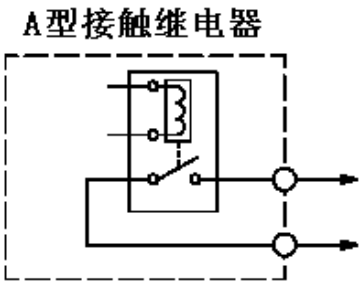
以下场合不宜使用继电器输出：

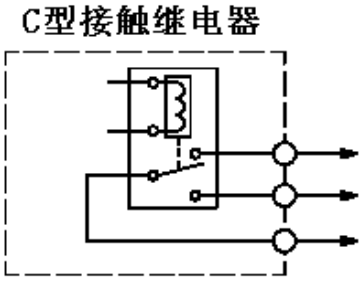
负载工作电流低于 10 mA

负载的开关频率较高或高强度的循环场合

DL305 输出模块的输出点有两种接触方式，如右图所示。A 型或称 SPST（单刀单掷——single pole,single throw）型是一个常开触点，这也是最为常用的一种类型。C 型或称 SPDT（单刀双掷——single pole,double throw）型有一个中心触点可以移向常开触点或常闭触点。此类接触形式可以提供一组常开和一组常闭触点。

一些继电器输出模块上的继电器有公共端，它们与继电器中的动触点相连接？。其它继电器模块都有各自完全独立的继电器。不管怎样，只要当输出点 ON 时，模块就驱动继电器线圈。





延长继电器的寿命

动作的次数，动作时产生的火花和空气中的污染物都会影响继电器触点的寿命。不管怎样，以下方法可以帮助您尽量延长继电器触点的寿命：

- ●只在系统确实需要的时候才让继电器动作
- ●如果可能的话，在电流最小的时候让继电器动作
- ●采取一定的措施来降低感性元件（比如接触器和螺线管）产生的感生电压。下面给出了具体电路图。



通过添加外部触点保护电路，可以让继电器的寿命超出继电器模块附带的规格表中列出的使用寿命。类似离合器、刹车装置、电动机、直接动作电磁阀和电机启动器等大电流的感性负载若采用了触点保护将大大受益。

RC 网络必须在靠近输出点附近连接。要确定 RC 缓冲区网络的参数，先要知道常开触点两端的电压以及闭合时通过触点的电流。如果负载使用的是交流电，就把其电流和电压转换成峰值：

这样您就可以按下面的式子计算出 R 和 C 的值：

$$C(\mu F)=\frac{I^2}{10} \qquad R(\Omega)=\frac{V}{10\times I^x} \quad ，其中\ x=1+\frac{50}{V}$$

C_{min}=0.001μ F，电压等级应大于 V，无极性

R_{min}＝0.5Ω，1/2W，精度±5%

例如，假设继电器要驱动的负载是 120VAC，0.5A。考虑到本例中的电源是交流电，首先要算出其电流电压的峰值。

$$I_P=I_{rms}\times 1.414=0.5\times 1.414=0.707\ （A）$$

$$V_P=V_{rms}\times 1.414=120\times 1.414=169.7\ （V）$$

然后确定 R 和 C 的值：

$$C(\mu F)=\frac{I^2}{10}=\frac{0.707^2}{10}=0.05\mu F,\ 电压等级大于\ 170V$$

$$R(\Omega)=\frac{V}{10\times I^x}，其中\ x=1+\frac{50}{V}$$

$$x=1+\frac{50}{169.7}=1.29 \quad R(\Omega)=\frac{169.7}{10\times 0.707^{1.29}}=26\Omega,\ 1/2\ W,\ \pm 5\%$$

如果触点控制的是一个直流的感性负载，尽可能在负载线圈旁边加一个二极管。当负载得电时，该二极管反偏（高阻抗状态）。当负载断电时，线圈中的储能以反向电压的形式释放。此时二极管正偏（低阻抗状态），并且把电流分流到接地端。这可以避免继电器触点受到动作瞬间所产生的高电压电弧的冲击。

为了取得最好的效果，请按以下指南使用一个降噪二极管：

- ●千万不能把此电路用于交流场合
- ●把二极管安装在尽可能靠近感性元件的地方
- ●二极管的反向击穿电压必须至少 100 PIV,正向电流 3A 以上。应使用快速恢复型（比如肖特基二极管）。千万不能使用类似 1N914,1N941 等的小信号二极管。
- ●通电前先确认二极管已安装正确。如果装反了，继电器一动作就会把电源烧坏。

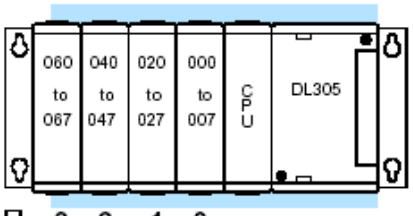
I/0 模块的位置，接线和详细说明

槽的编号

DL305 的基本框架提供不同数量的插槽用来插 I/0 模块。您会发现基本框架有 5 槽的，8 槽的等。其中一个是专门给 CPU 模块用的，所以您总是必须少用一个 I/O 插槽。比如，在 5 槽的基本框架上您只能使用 4 个 I/O 插槽。其编号为 0－3。CPU 插槽总是包含一个 CPU,因此不对它编号。

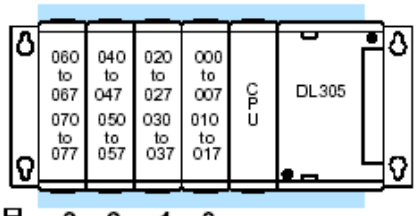
以下是在 5 槽框架上使用 8 点和 16 点 I/O 模块时分配 I/O 点的例子。

使用8点的I/0模块



插槽编号：3—2—1—0

使用16点的I/0模块



插槽编号：3—2—1—0

I/O 模块安放规则

在哪里插入什么模块是有一定的限制的。一些模块必须插在指定的插槽中，因此会影响到其它模块的安放及其编号。

如果您还有一些疑问，请继续往下看。您会逐渐消除您的疑问并找到适合您的设置方法。

在所有先前提到的设置中，将会用到的插槽编号会在扩展框架时用到。比如，如果要求声明某个模块必须放在六个插槽中最靠近 CPU 的那个插槽，那么系统必须由两个 5 槽

框架来组成，扩展框架的 1 号 2 号槽是有效的位置。

以下表格提供了 DL305 系统组件的一般安放规则。

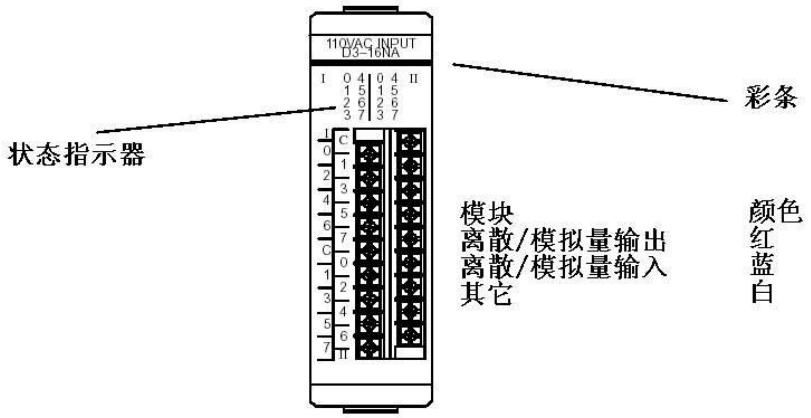
模块	限制
CPU	CPU 模块必须插在本地 CPU 框架的第一槽中。第一槽就是最靠近电源的那一槽。
16 点 I/O 模块	任意槽
模拟量模块	必须插在支持 16 点 I/O 的槽内
ASCⅡ处理模块	必须插在支持 16 点 I/O 的槽内
高速计数模块	D3-350 CPU 不支持高速计数模块

模块状态指示器

一些模块上有 LED 来指示当前的输入点状态。

I/O 模块颜色分类

DL305 家族的模块上有彩条，它将有助于您快速地辨别出是输入模块，输出模块，还是特殊模块。该彩条位于模块的正面。如下图所示：

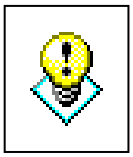


不同模块的接线

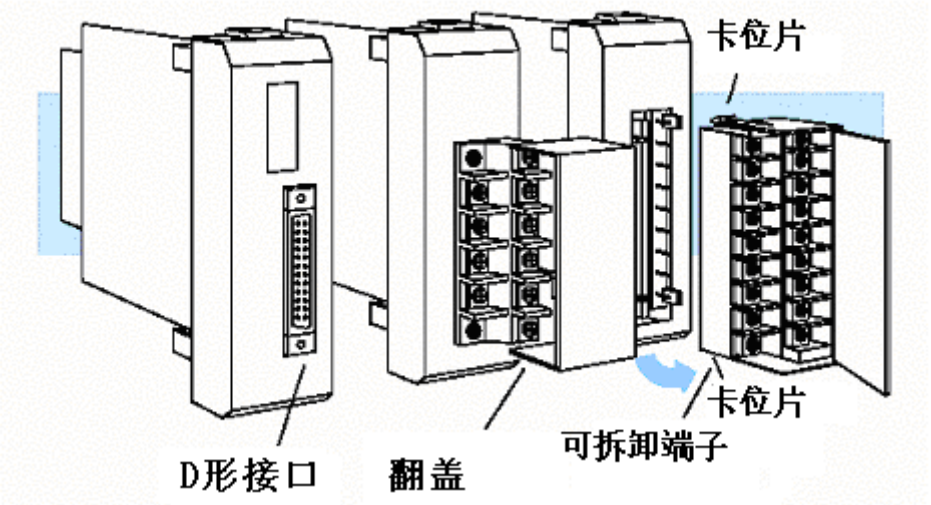
DL305 的 I/O 模块有三种不同的接口。一些模块有普通的螺丝钉接口。有些模块的接线端子是下凹的。下凹的端子可以防止人员不小心触及上面的接线。第三种是用来和专门的电缆连的 D 形接口。

两种螺丝钉式的端子都很容易拆卸。如果您仔细观察，您会发现在模块的顶部和底部有两个卡位片。要取出接线端子时，先按下两个卡位片，然后向外拔。

我们也有 DIN 导轨型的接线端子板和 DINEnector（请参考产品目录）。此 DINEnector 含有一个特殊的预装好的电缆，其 I/O 接口也已安装好了。



警告：对于一些模块，由于外接电源的存在，在 PLC 已经断电时也会带电。为了尽量避免触电的风险，拆卸时应保证外接的现场设备都已断电。



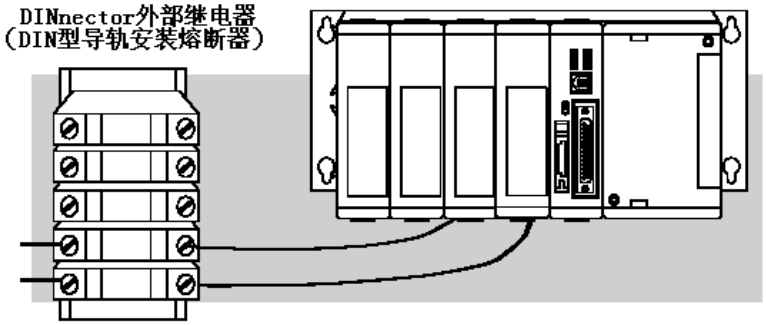
I/O 接线

在给您的系统接线时，请参考以下指南

1．不同模块支持的线径可能有所不同，请看下面的表格。线径稍小时，不会影响使用。

模块类型	最大 AWG
8 点	12AWG
16 点	16AWG

2．总是使用连续的导线，不要用两根线接起来来达到所需的长度
3．尽量缩短接线的长度
4．有可能就使用接线片
5．避免信号线靠近动力线
6．避免输入信号线靠近输出信号线
7．当接线较长时，为了减小压降，在返回线中使用多股线？
8．避免直流线和交流线相靠近
9．避免接线有过分的弯曲
10．为了避免模块烧坏，我们建议您最好添加一些外部熔断器。可以在每个公共端加一个熔断电流小于模块自带熔断器熔断电流的快速型的熔断器，或者加一个熔断电流略小于输出继电器触点电流的熔断器。请参考我们的产品目录来选取熔断块所用的 DINEnector 和 DIN 导轨。



注意：由于我们对自行拆机者将不负责任，对于一些熔断器是焊死的或不可更换的模块，如果其熔断器烧掉，我们建议您拿给我们来处理。

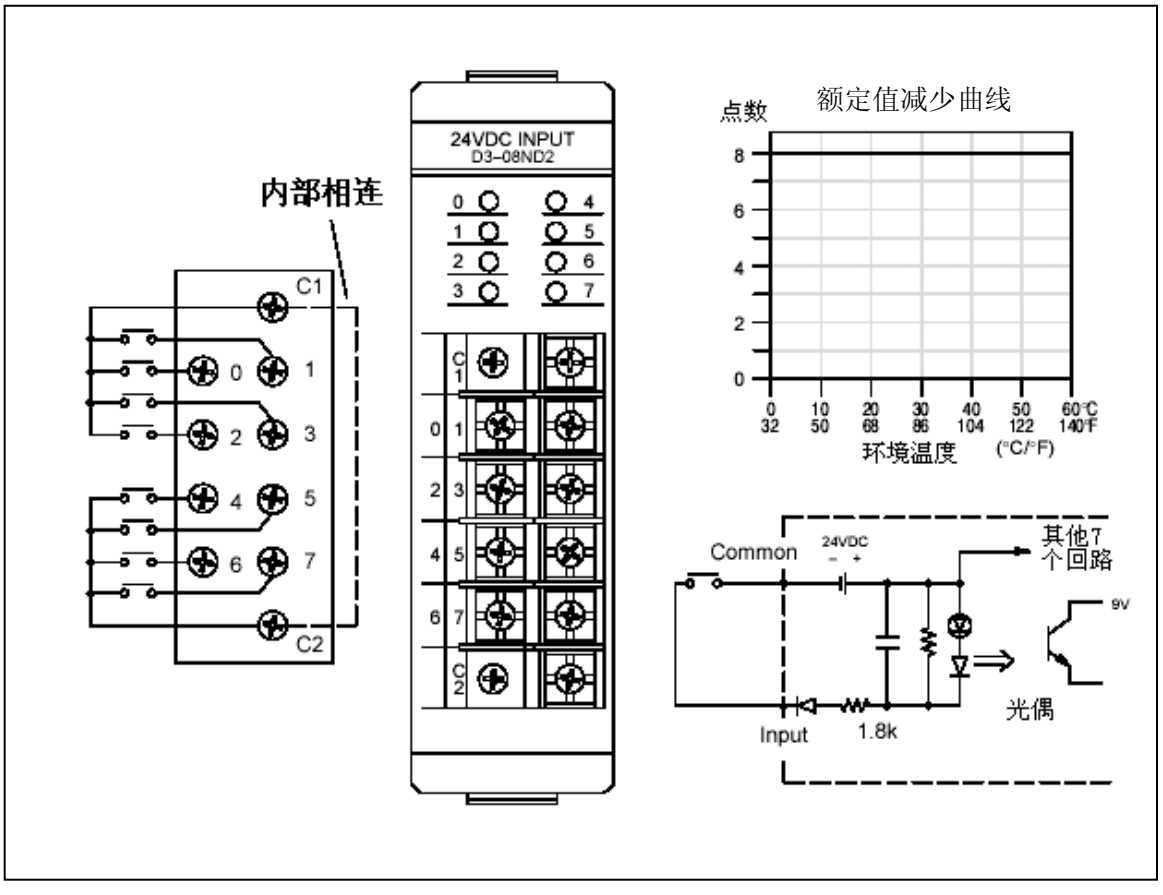
详细说明的术语表

Inputs or Outputs Per Module	说明每个模块的输入输出点数并指明是是源点型还是汇点型，或两者都能用
Commons Per Module	每个模块的公共端数量及其电气特性
Input Voltage Range	输入回路的工作电压范围
Output Voltage Range	输入回路允许的最大电压
Peak Voltage	输入回路允许的最大电压
AC Frequency	AC 模块只能在特定频率的交流电下工作
ON Voltage Level	让输入点为 ON 所需要的电压等级
OFF Voltage level	让输入点为 OFF 所需要的电压等级
Input Impedance	输入阻抗可以用来计算在特定电压下的工作电流
Input Currant	输入点为 ON 时的典型电流值
Minimum ON Current	输入回路要维持 ON 状态所需要的最小电流
Minimum OFF Current	输入回路要维持 OFF 状态所允许的最大电流
Minimum Load	输出回路工作时的最小负载电流
External DC Required	一些输出模块需要外接电源来给输出回路供电
ON Voltage Drop	也称“饱和电压”，当输出负载最大时在输出点和公共端之间测得的电压
Maximum Leakage Current	输出点为 OFF 状态时的最大电流

Maximum Inrush Current	一个输出点从 OFF 变为 ON 时瞬间所能承受的最大电流。其一般要比输出点 ON 时的电流小而且在交流回路中是感性负载的一个特性
Base Power Required	DL305 的自带的电源给其输入模块供电并随着模块的不同而不同。模块功率计算指南请参考 4—5。
OFF to ON Response	模块从 OFF 变为 ON 时所花的时间
ON to OFF Response	模块从 ON 变为 OFF 时所花的时间
Terminal Type	指明其端子类型是可拆卸式的或不可拆卸式的或其它形式的
Status Indicators	指示当前输入点状态的一组 LED。这些 LED 安装在输入回路的内部处理侧或现场侧。
Weight	说明模块的重量。DL305 的各种组件的重量在附录 E 上。
Fuses	输出回路的一个保护装置，当通过的电流超过其熔断值时就切断电路。有可更换式的和不可更换式的，或者内置的和外置的。
Module	模块
Sinking	汇点
Sourcing	源点

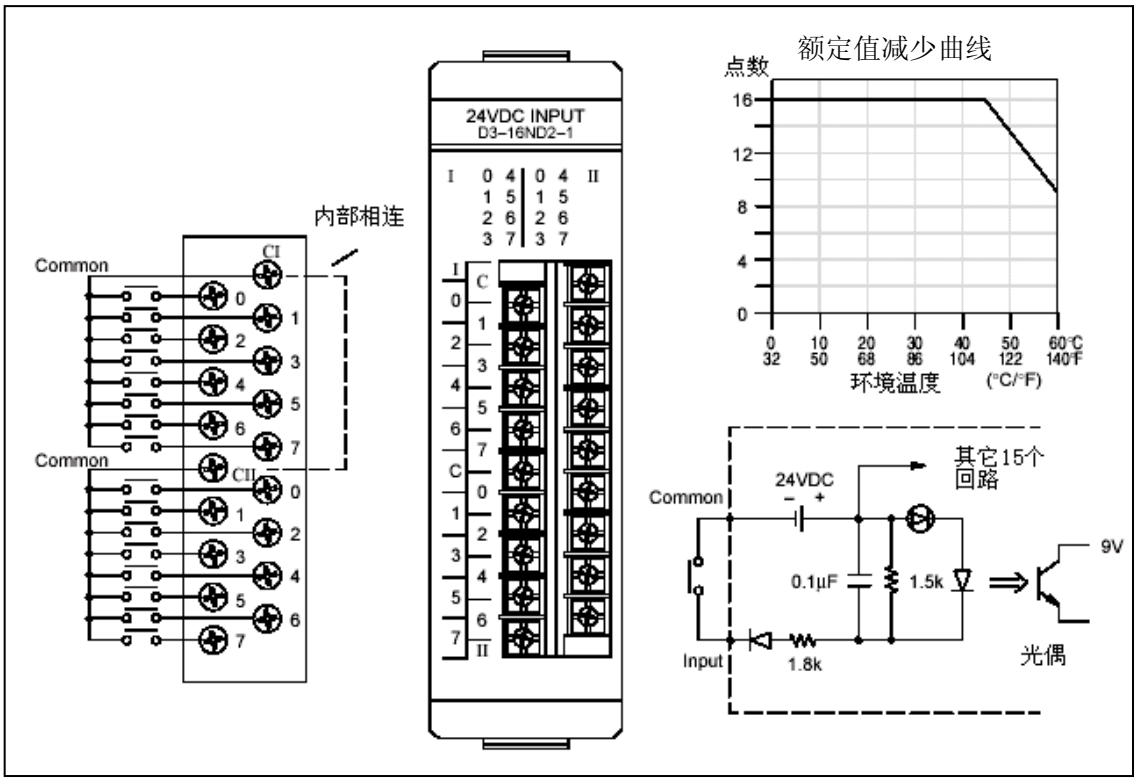
D3-08ND2,24VDC 输入模块

回路数	8 点		框架电流消耗	9V 最大 10mA 24V 14mA/ON 每点（最大 112mA）
公共端数量	2（内部相连）			
输入电压范围	18～36VDC			
输入电压	内部供电		OFF→ON 相应	4～15ms
峰值电压	40VDC		ON→OFF 相应	4～15ms
交流频率	N/A		端子类型	不可拆卸式
ON 电压	<3V		状态指示灯	现场侧
OFF 电压	>18V		重量	120g
输入阻抗	1.8KΩ			
输入电流	最大 12mA			
最小 ON 电流	7mA			
最大 OFF 电流	3mA			



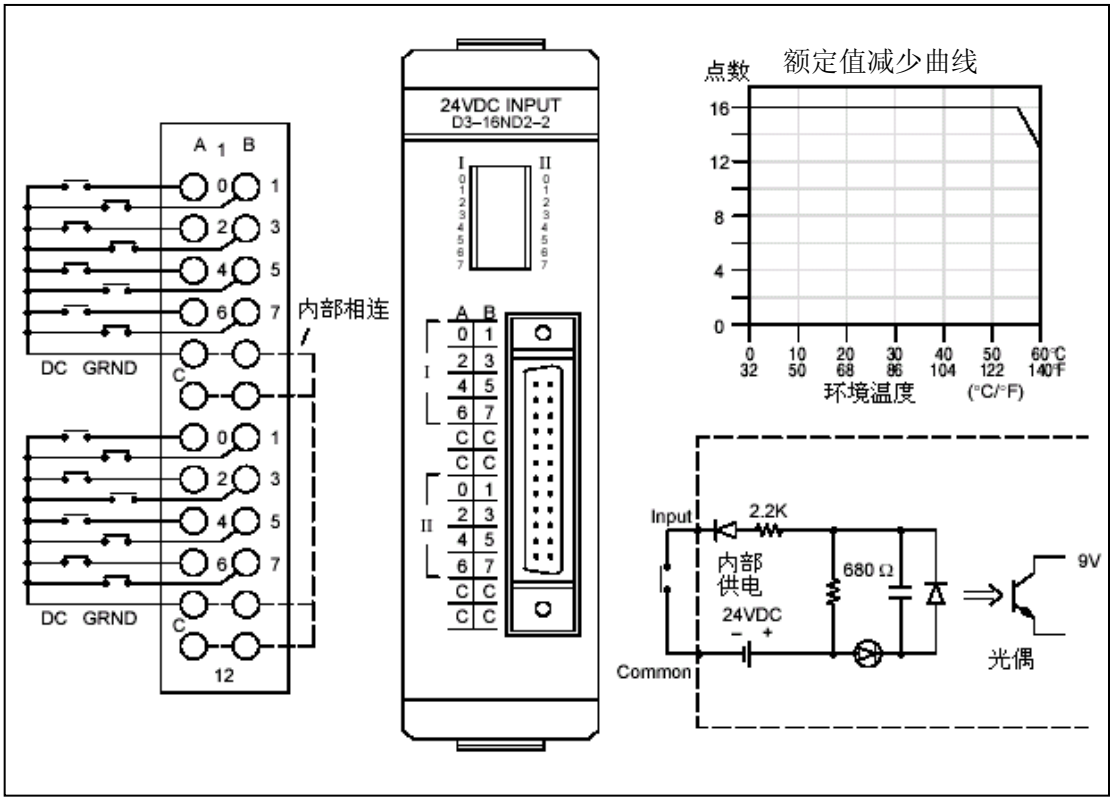
D3-16ND2-1, 24VDC 输入模块

回路数	16 点		框架电流消耗	9V 最大 25mA 24V 14mA/ON 每点（最大 224mA）
公共端数量	2（内部相连）			
输入电压范围	18～36VDC			
输入电压	内部供电		OFF→ON 相应	3～15ms
峰值电压	36VDC		ON→OFF 相应	4～15ms
交流频率	N/A		端子类型	可拆卸式
ON 电压	<3V		状态指示灯	现场侧
OFF 电压	>19V		重量	180g
输入阻抗	1.8KΩ			
输入电流	最大 20mA			
最小 ON 电流	5mA			
最大 OFF 电流	1mA			



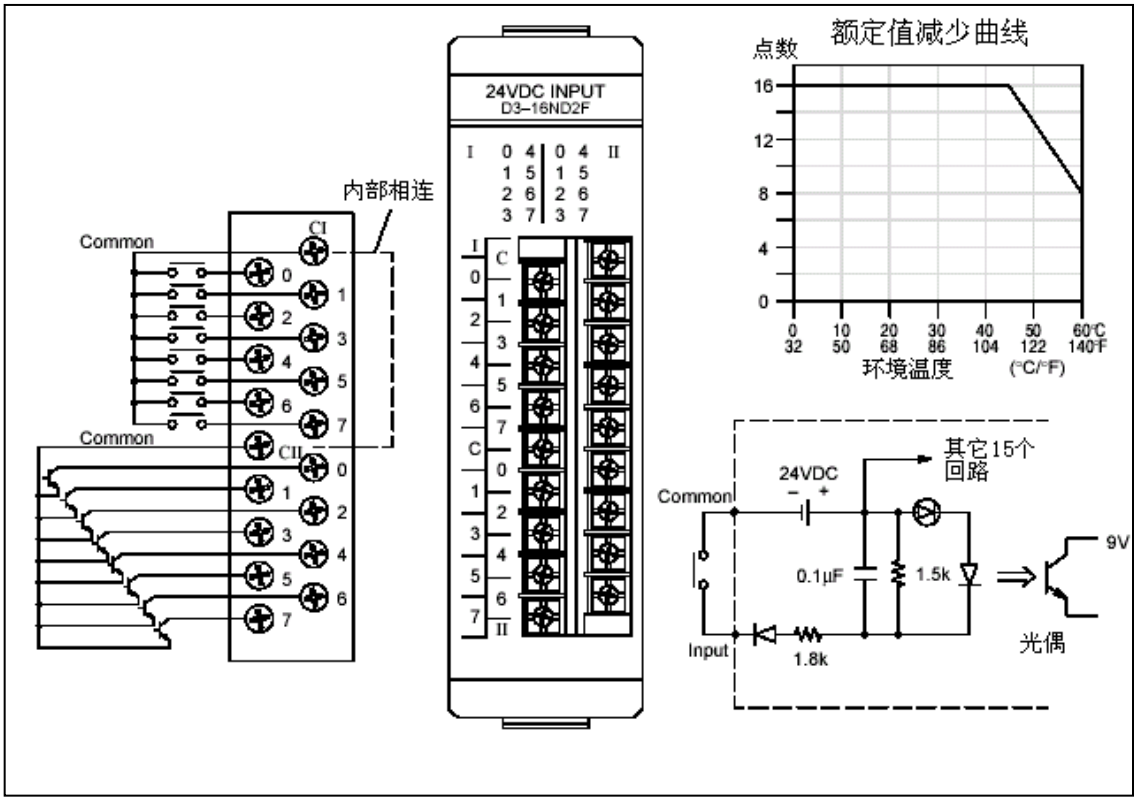
D3-16ND2-2, 24VDC 输入模块

回路数	16 点		框架电流消耗	9V 3mA+1.3mA/ON 每点（最大 24mA） 24V3mA+1.3mA/ON 每点（最大 209mA）
公共端数量	8（内部相连）			
输入电压范围	18～36VDC			
输入电压	内部供电			
峰值电压	36VDC		OFF→ON 相应	4～15ms
交流频率	N/A		ON→OFF 相应	4～15ms
ON 电压	<3V		端子类型	24P 可拆卸式
OFF 电压	>19V		状态指示灯	现场侧
输入阻抗	2.2KΩ		重量	150g
输入电流	最大 20mA			
最小 ON 电流	5mA			
最大 OFF 电流	2mA			



D3-16ND2F,24VDC 高速响应模块

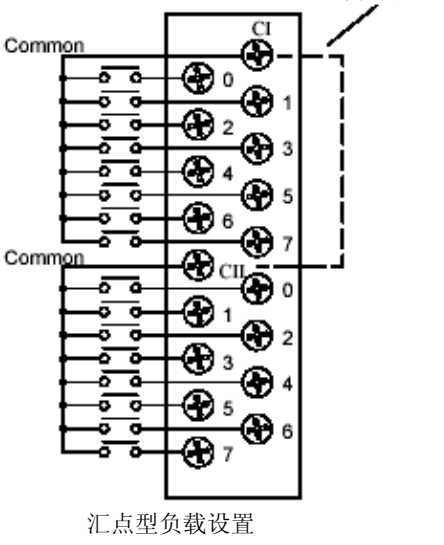
回路数	16 点（源点）		框架电流消耗	9V 最大 25mA 24V 14mA/ON 每点（最大 224mA）
公共端数量	2（内部相连）			
输入电压范围	18～36VDC			
输入电压	内部供电		OFF→ON 相应	0.8ms
峰值电压	36VDC		ON→OFF 相应	0.8ms
交流频率	N/A		端子类型	可拆卸式
ON 电压	<13V		状态指示灯	现场侧
OFF 电压	>19V		重量	180g
输入阻抗	1.8KΩ			
输入电流	最大 20mA			
最小 ON 电流	5mA			
最大 OFF 电流	1mA			



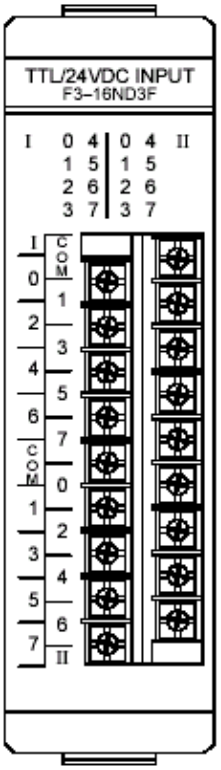
F3-16ND3F,TTL/24VDC 高速响应模块

回路数	16 点（源/汇点，通过跳线来选择）*		框架电流消耗	9V 最大 148mA 24V 最大 68mA
公共端数量	2（内部相连）		输入电流	5VDC:1mA 12～24VDC:3mA
输入电压范围	5VDC TTL 和 CMOS, 12~24VDC(通过跳线来选择)		输入阻抗	4.7K
输入电压	内部供电（汇点型负载） 外部供电（源点型负载）		OFF→ON 相应	1ms
			ON→OFF 相应	1ms
峰值电压	100VDC （35VDC 连续）		最高输入频率	500Hz
交流频率	N/A		最小 ON 电流	5VDC:0.4mA 12～24VDC:0.9mA
ON 电压	5VDC:0~1.5VDC 12~24VDC:0~4VDC		最大 OFF 电流	5VDC:0.8mA 12～24VDC:2.2mA
OFF 电压	5VDC:3.5~5VDC 12~24VDC:10~24VDC		端子类型	可拆卸式
			状态指示灯	逻辑侧
			重量	153g

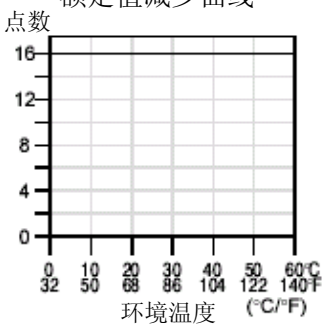
*12 路输入可以通过跳线来选择 5VDC/12~24VDC 及汇点型负载/源点型负载
4 路输入可以通过跳线来选择 5VDC/12~24VDC 及汇点型负载/源点型负载

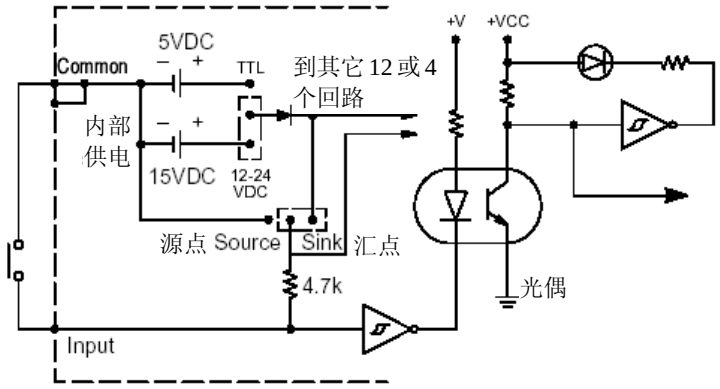


汇点型负载设置

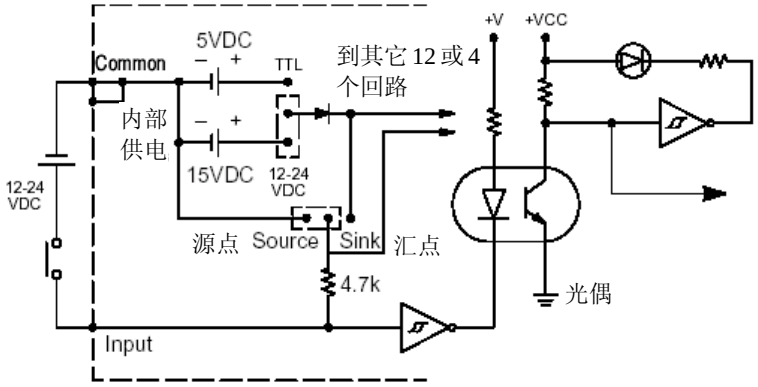


额定值减少曲线





通过跳线设定12－24VDC，汇点负载



通过跳线设定源点负载。必须使用外接电源。

设定工作方式

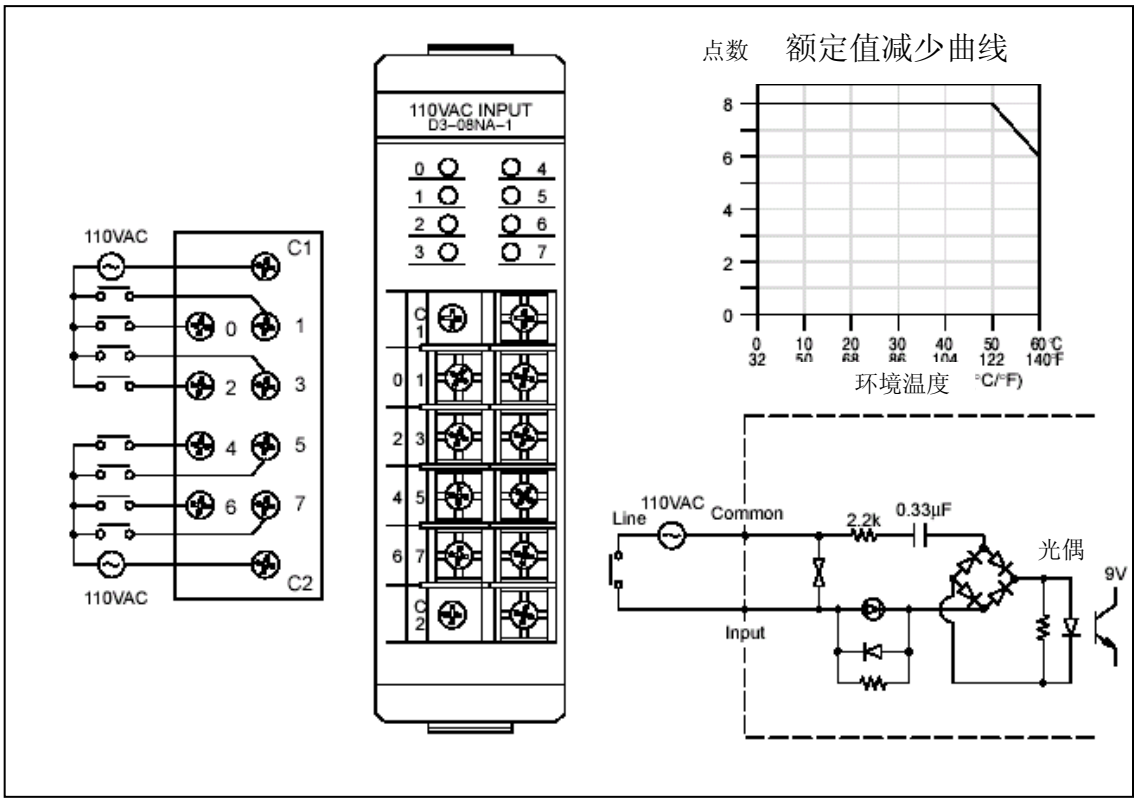
当跳线帽放在汇点信号侧时，用来检测输入状态的 DC 电源是框架自带的电源。当输入端和公共端短接时，汇点型输入就变成 ON。源点型输入信号在输入装置提供的电压使输入状态变为 OFF 时，它才呈现 ON 的状态。

每组电路的工作模式，比如 5VDC 或 12－24VDC 的汇点或源点输入，都可以通过改变电路板边缘的跳线帽的位置来实现。一共有 4 种跳线方式（每个跳线有三针），每一组有两个跳线位置。前两种设置方式用来设定前面的 12 点输入（比如 0－7 和 100－103）并标注为 12 回路。在第一组跳线上面标有 12/24V 和 5V。第二组跳线上标有 SINK（汇）和 SRC(SOURCE——源)。要选择 12 回路工作方式，把跳线帽插在最靠近对应标签的两根针上。比如，要设定前 12 个输入点为 24VDC 汇点输入，把一个跳线帽放在有 12/24V 标签的位置，把另一个放在有 SINK 标签的位置。后两种方式是用来设置后 4 点输入的（比如 104－107）并标注为 4 回路。这 4 回路可以设定成与前面 12 回路不同的工作方式。要设定正确，每一组都必须安放上跳线帽（总共 4 种跳线方式）。

注意：当一组输入用的是 TTL 电平，要为其选择 SINK 工作方式。“标准”的 TTL 能够汇几个毫安的电流，但只能源不到 1 毫安的电流。

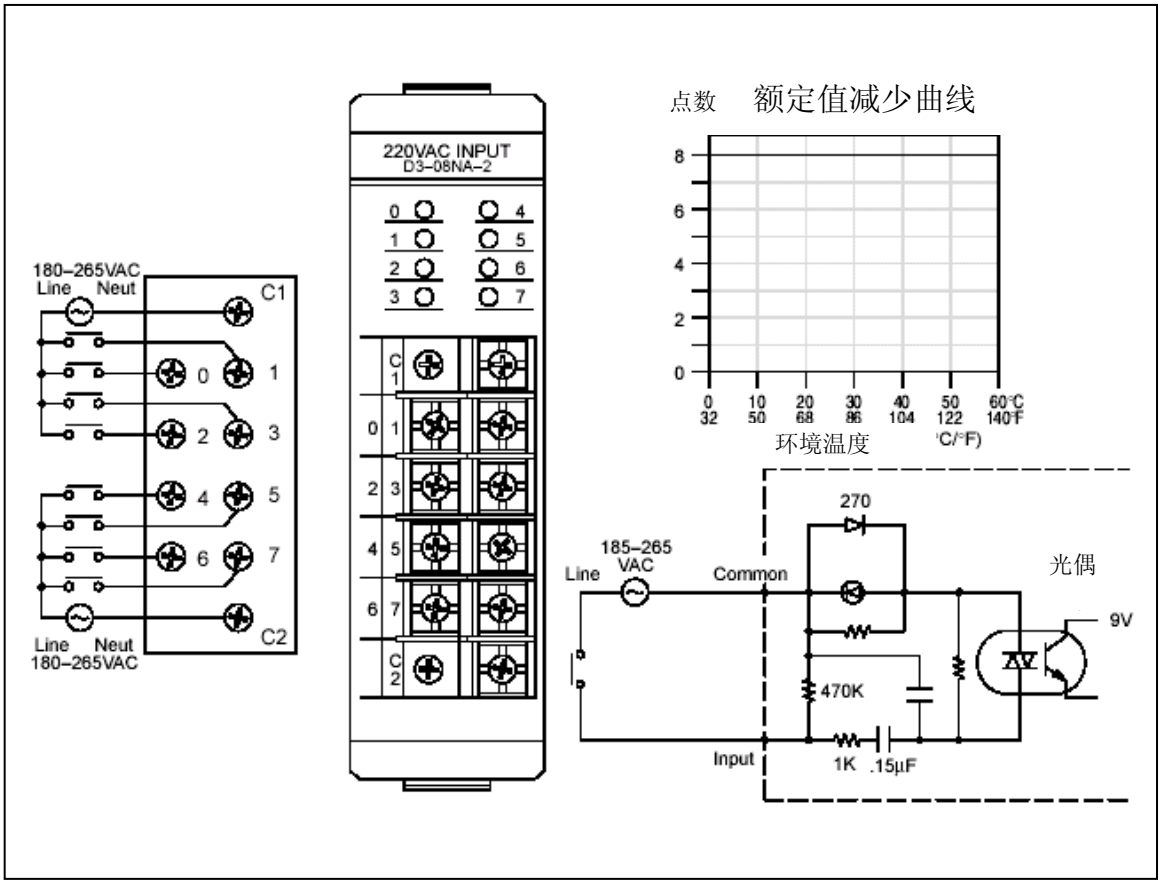
D3-08NA-1, 110VAC 输入模块

回路数	8		最小 ON 电流	8mA
公共端数量	2（隔离）		最大 OFF 电流	2mA
输入电压范围	85～132VAC		框架电流消耗	9V 最大 10mA 24V N/A
输入电压	外部供电			
峰值电压	132VAC		OFF→ON 相应	10～30ms
交流频率	47~63Hz		ON→OFF 相应	10～30ms
ON 电压	>80VAC		端子类型	不可拆卸式
OFF 电压	<20VAC		状态指示灯	现场侧
输入阻抗	10KΩ		重量	180g
输入电流	50Hz: 15mA 60Hz：18mA			



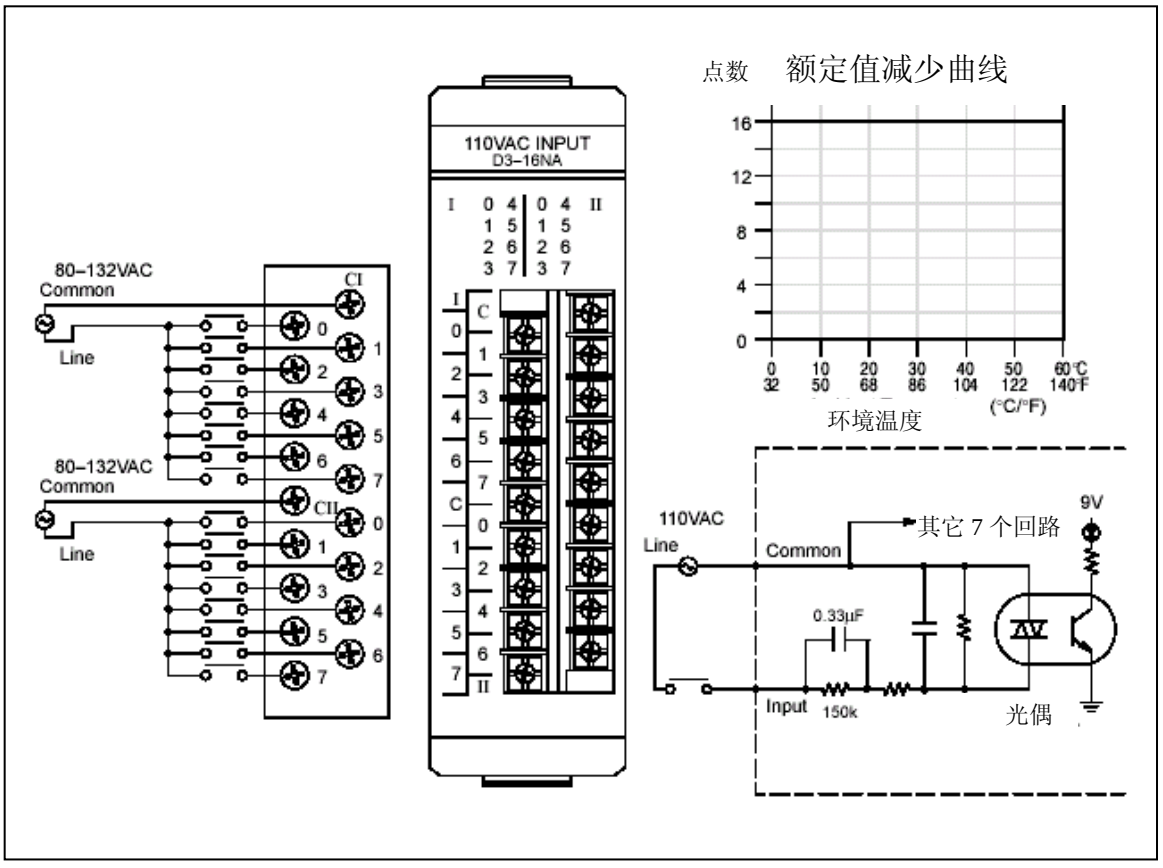
D3-08NA-2,220VAC 输入模块

回路数	8		最小 ON 电流	10mA
公共端数量	2（隔离）		最大 OFF 电流	2mA
输入电压范围	180～265VAC		框架电流消耗	9V 最大 10mA 24V N/A
输入电压	外部供电			
峰值电压	265VAC		OFF→ON 相应	5～60ms
交流频率	50～60Hz		ON→OFF 相应	5～60ms
ON 电压	>180VAC		端子类型	不可拆卸式
OFF 电压	<40VAC		状态指示灯	现场侧
输入阻抗	18KΩ		重量	140g
输入电流	50Hz: 13mA 60Hz：18mA			



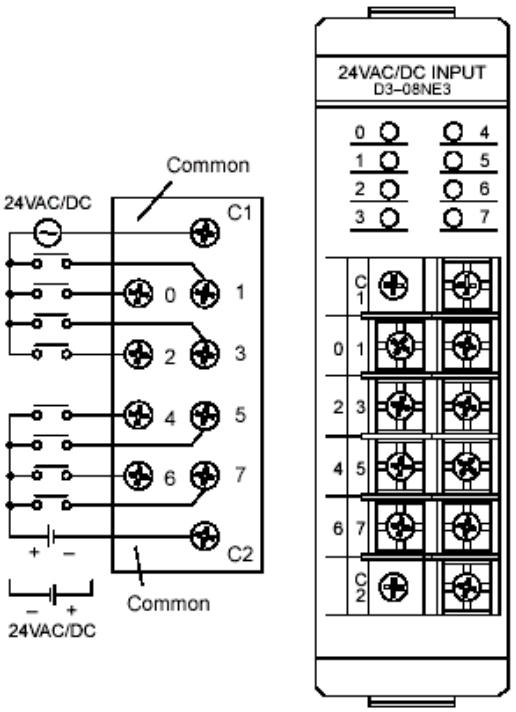
D3-16NA,110VAC 输入模块

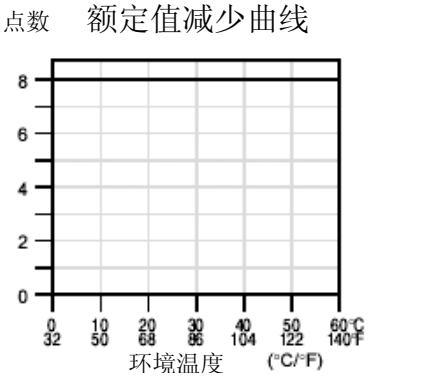
回路数	16		最小 ON 电流	8mA
公共端数量	2（隔离）		最大 OFF 电流	1.5mA
输入电压范围	80～132VAC		框架电流消耗	9V 最大 6.25mA/ON 每点最大 100mA
输入电压	外部供电			
峰值电压	132VAC			
交流频率	50～60Hz		OFF→ON 相应	5～50ms
ON 电压	>80VAC		ON→OFF 相应	5～60ms
OFF 电压	<15VAC		端子类型	可拆卸式
输入阻抗	8KΩ		状态指示灯	逻辑侧
输入电流	50Hz: 16mA 60Hz：25mA		重量	180g

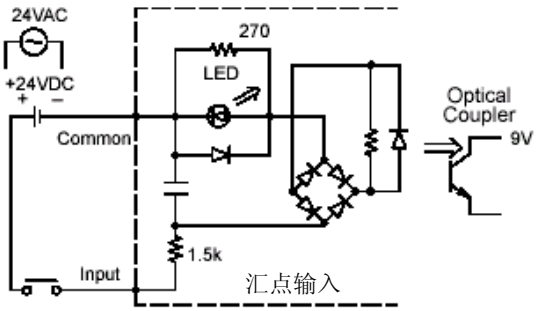


D3-08NE3,24VAC/DC 输入模块

回路数	8 点（汇/源点）		框架电流消耗	9V 最大 10mA 24V N/A
公共端数量	2（隔离）			
输入电压范围	20～28VAC/DC			
输入电压	外部供电			
峰值电压	28VAC/DC		OFF→ON 相应	AC:5～50ms DC:6～30ms
交流频率	47～63Hz			
ON 电压	>20V		ON→OFF 相应	AC/DC:5～60ms
OFF 电压	<6V		端子类型	不可拆卸式
输入阻抗	1.5KΩ		状态指示灯	现场侧
输入电流	最大 19mA		重量	120g
最小 ON 电流	10mA			
最大 OFF 电流	2mA			



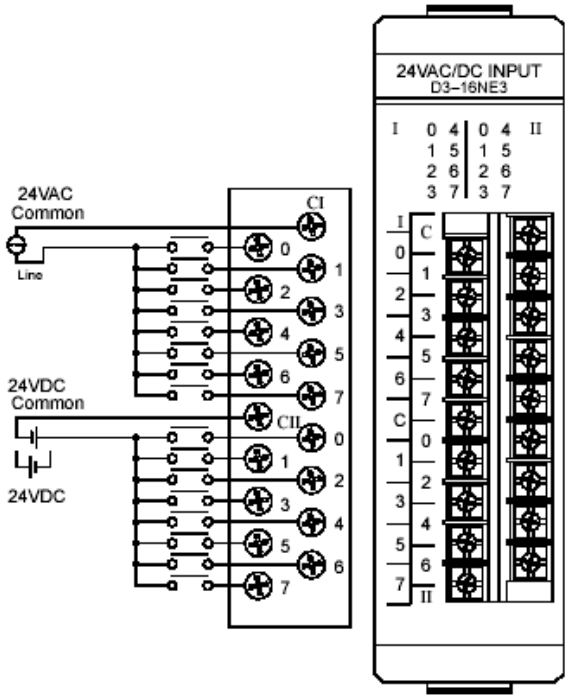




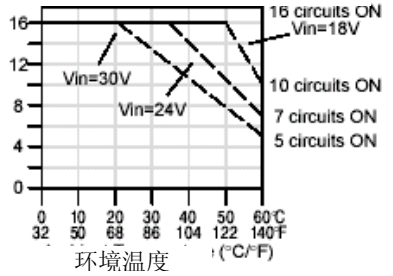
注意：该模块可以按源点的形式来接线，但是模块上的 LED 在有信号输入时也不会亮。

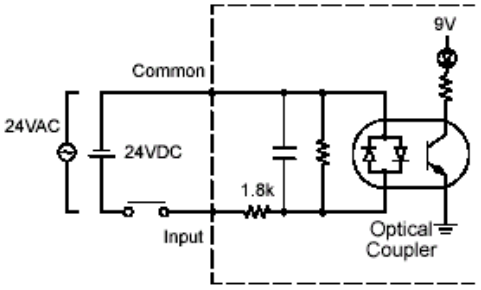
D3-16NE3,24VAC/DC 输入模块

回路数	16 点（汇/源点）		框架电流消耗	9V 2.5mA+4.5mA/ON 每点（最大 130mA） 24V N/A
公共端数量	2（隔离）			
输入电压范围	14～30VAC/DC			
输入电压	外部供电			
峰值电压	30VAC/DC		OFF→ON 相应	AC:5～30ms DC:5～25ms
交流频率	47～63Hz			
ON 电压	>14V		ON→OFF 相应	AC:5～30ms DC:5～25ms
OFF 电压	<3V		端子类型	可拆卸式
输入阻抗	1.8KΩ		状态指示灯	逻辑侧
输入电流	最大 16mA		重量	170g
最小 ON 电流	7mA			
最大 OFF 电流	2mA			



点数 额定值减少曲线

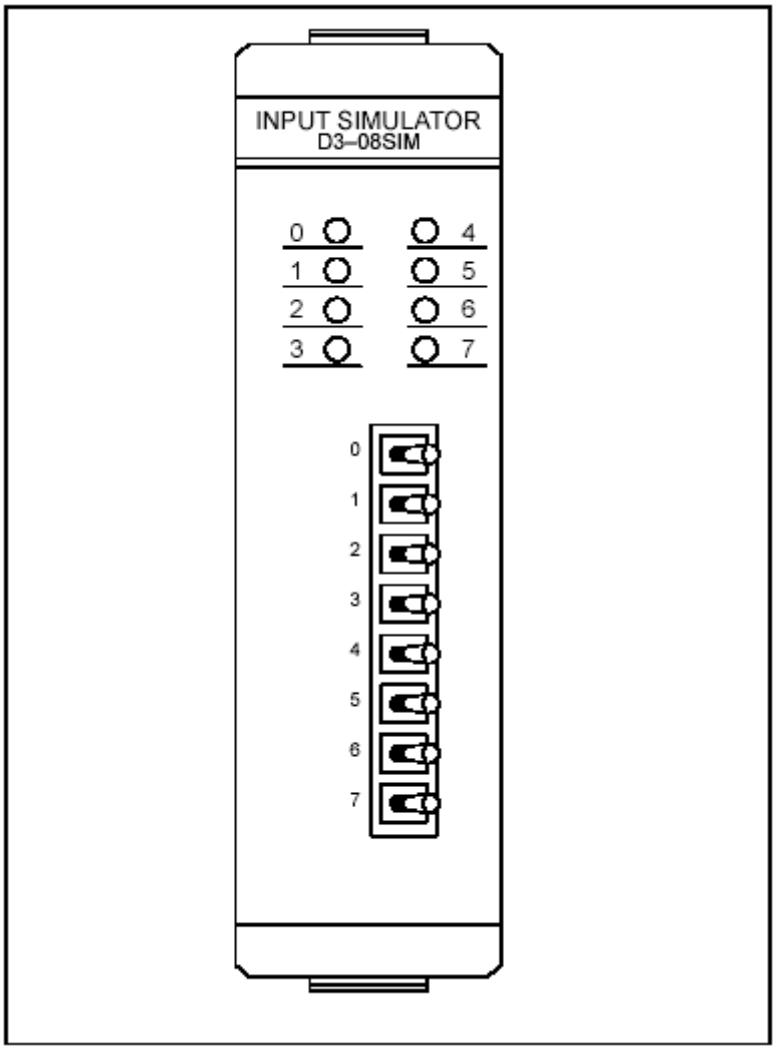




汇点模块设置

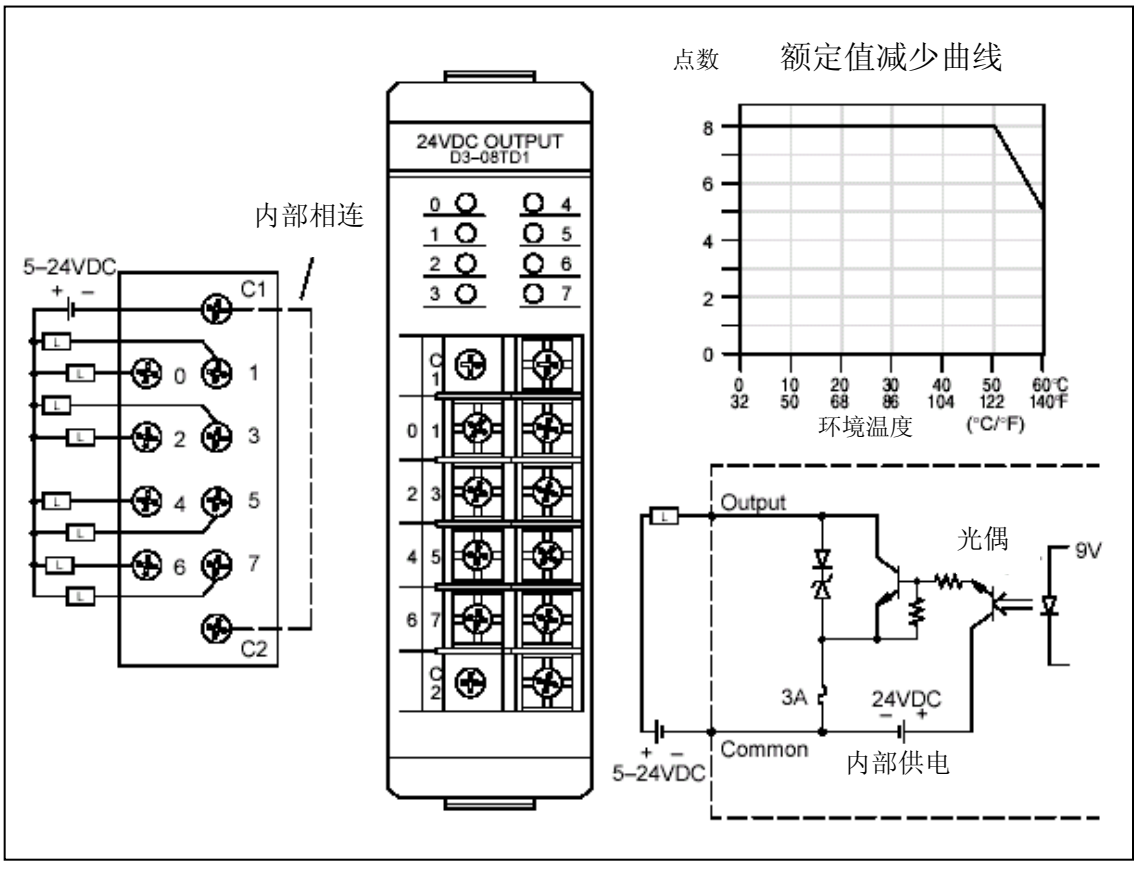
D3-16NA,110VAC 输入模块

回路数	16		最小 ON 电流	8mA
公共端数量	2（隔离）		最大 OFF 电流	1.5mA
输入电压范围	80～132VAC		框架电流消耗	9V 最大 6.25mA/ON 每点 最大 100mA
输入电压	外部供电			
峰值电压	132VAC			
交流频率	50～60Hz		OFF→ON 相应	5～50ms
ON 电压	>80VAC		ON→OFF 相应	5～60ms
OFF 电压	<15VAC		端子类型	可拆卸式
输入阻抗	8KΩ		状态指示灯	逻辑侧
输入电流	50Hz: 16mA 60Hz：25mA		重量	180g



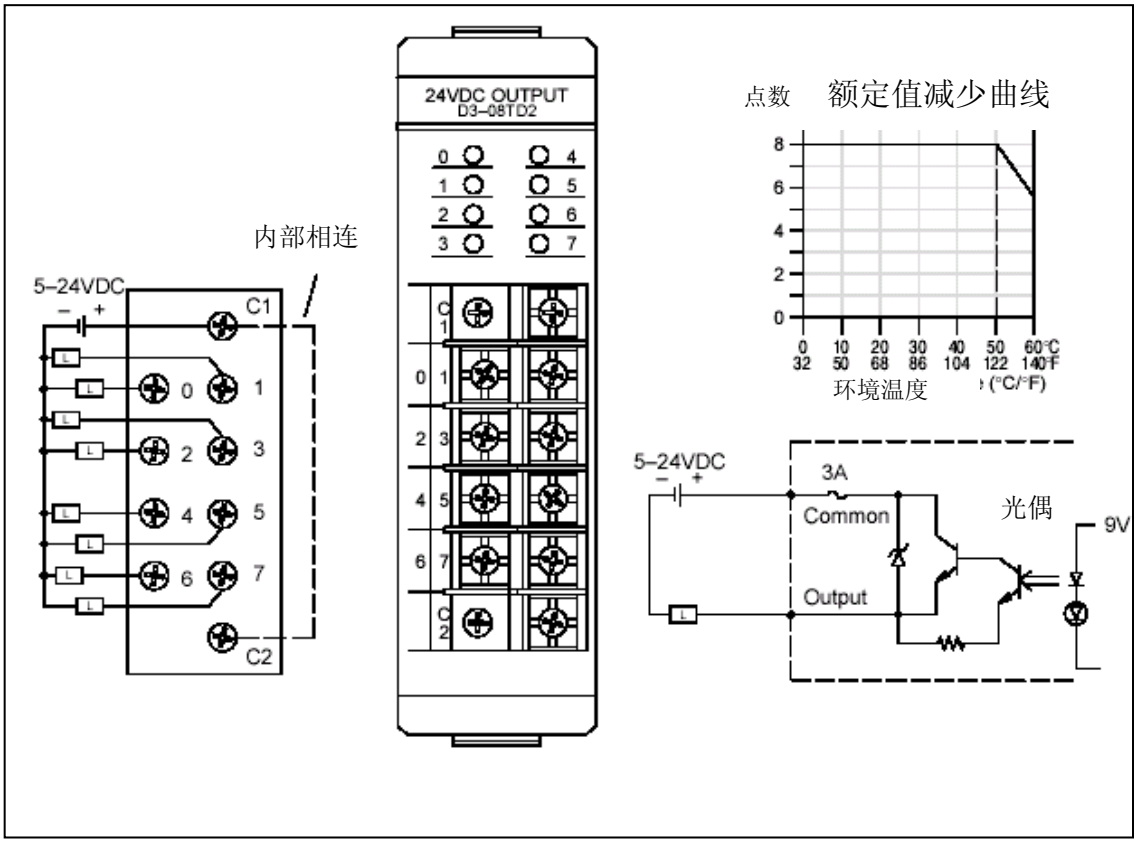
D3-08TD1,24VDC 输出模块

回路数	8 点（汇/源点）		最小负载	1mA
公共端数量	2（内部相连）		框架电流消耗	9V 最大 20mA 24V 3mA/每点 （最大 24mA）
工作电压范围	5～24VDC			
输出类型	NPN(集电极开路)			
峰值电压	45VDC		OFF→ON 相应	0.1ms
交流频率	N/A		ON→OFF 相应	0.1ms
ON 电压降落	0.5A 时 0.8V		端子类型	不可拆卸式
最大电流	0.5A/点 1.8A/公共端		状态指示灯	逻辑侧
			重量	170g
最大漏电流	40VDC 时 0.1mA		保险丝	（2）每个公共端用一个 3A 保险丝（不可更换式）
最大涌浪电流	3A/20ms 1A/100ms			



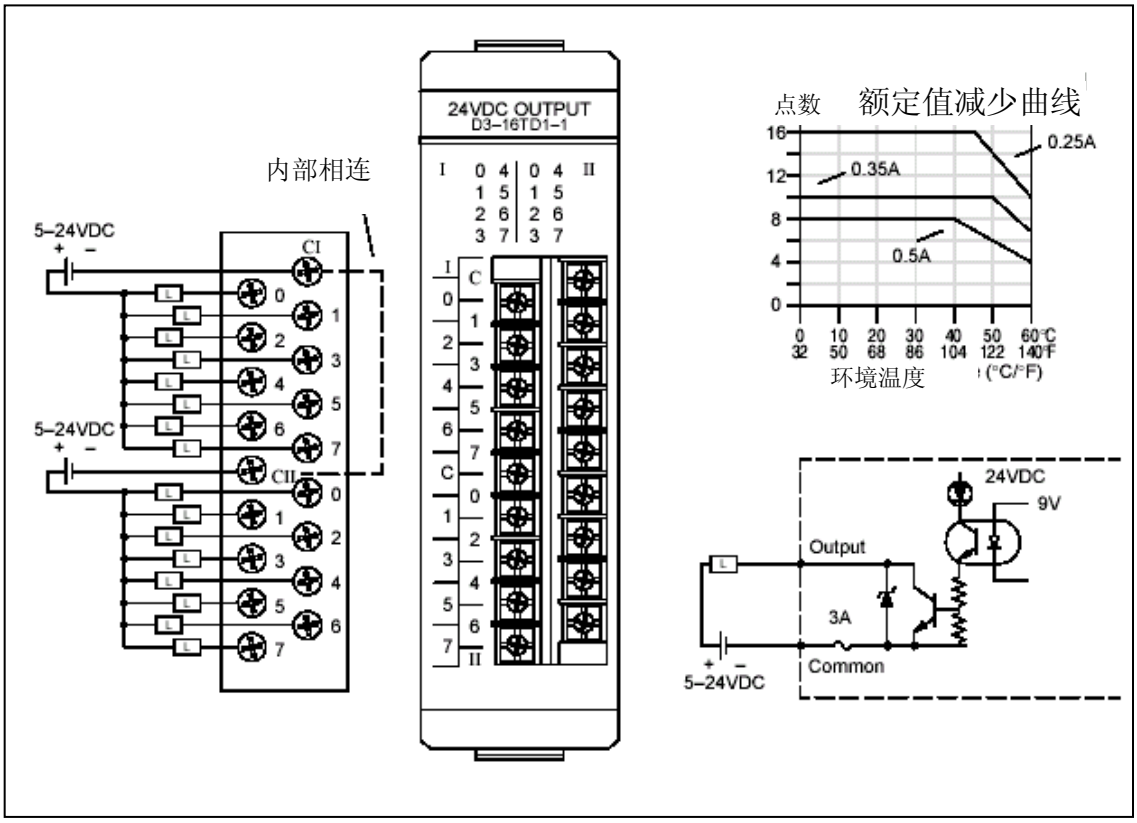
D3-08TD2,24VDC 输出模块

回路数	8 点（源点）		最小负载	1mA
公共端数量	2（内部相连）		框架电流消耗	9V 最大 20mA 24V N/A
工作电压范围	5～24VDC			
输出类型	NPN(发射极跟随)			
峰值电压	40VDC		OFF→ON 相应	0.1ms
交流频率	N/A		ON→OFF 相应	0.1ms
ON 电压降落	0.5A 时 1V		端子类型	不可拆卸式
最大电流	0.5A/点 1.8A/公共端		状态指示灯	逻辑侧
			重量	120g
最大漏电流	24VDC 时 0.1mA		保险丝	（2）每个公共端用一个 3A 保险丝（不可更换式）
最大涌浪电流	3A/20ms 1A/100ms			



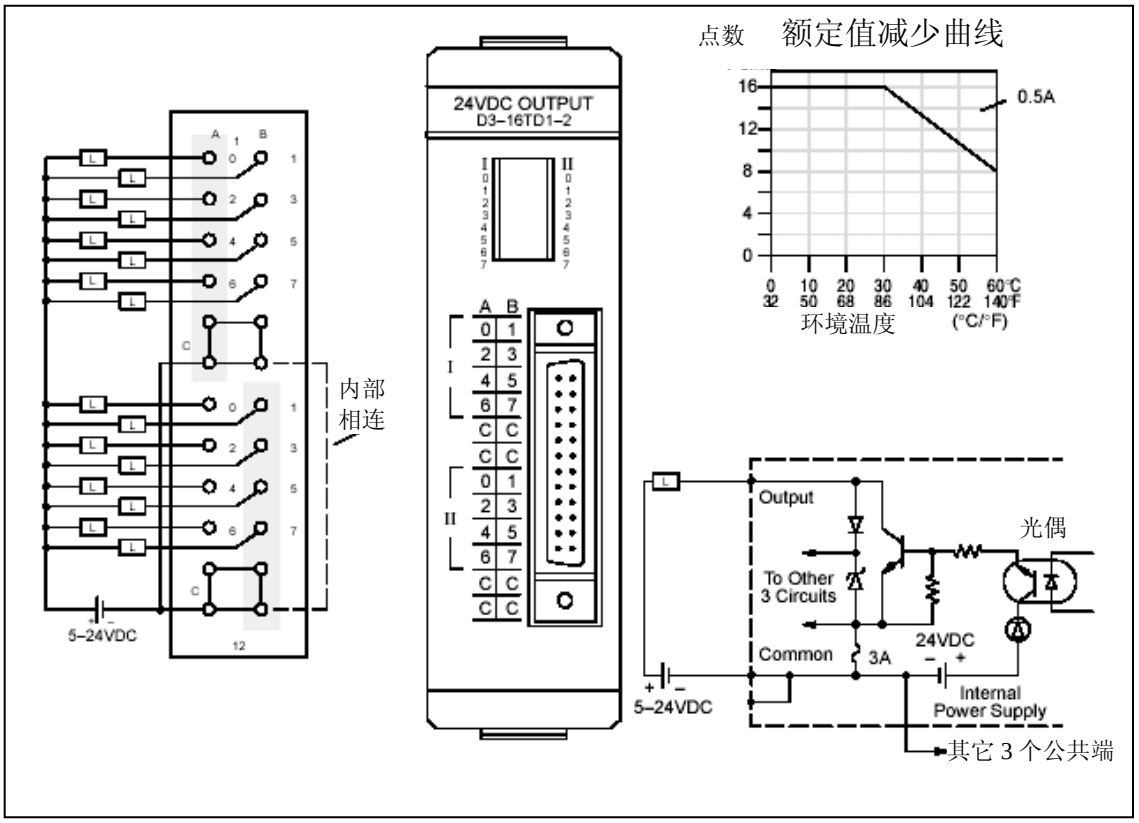
D3-16TD1-1,24VDC 输出模块

回路数	16 点（汇点）		最小负载	1mA
公共端数量	2（内部相连）		框架电流消耗	9V 最大 40mA 3mA+2.3mA/ON 每点 24V 最大 96mA 6mA/ON 每点
工作电压范围	5～24VDC			
输出类型	NPN(集电极开路)			
峰值电压	45VDC			
交流频率	N/A		OFF→ON 相应	0.1ms
ON 电压降落	0.5A 时 2V		ON→OFF 相应	0.1ms
最大电流	0.5A/点 2A/公共端		端子类型	可拆卸式
最大漏电流	40 VDC 时 0.1mA		状态指示灯	逻辑侧
			重量	160g
最大涌浪电流	3A/20ms 1A/100ms		保险丝	（2）每个公共端用一个 3A 保险丝（不可更换式）



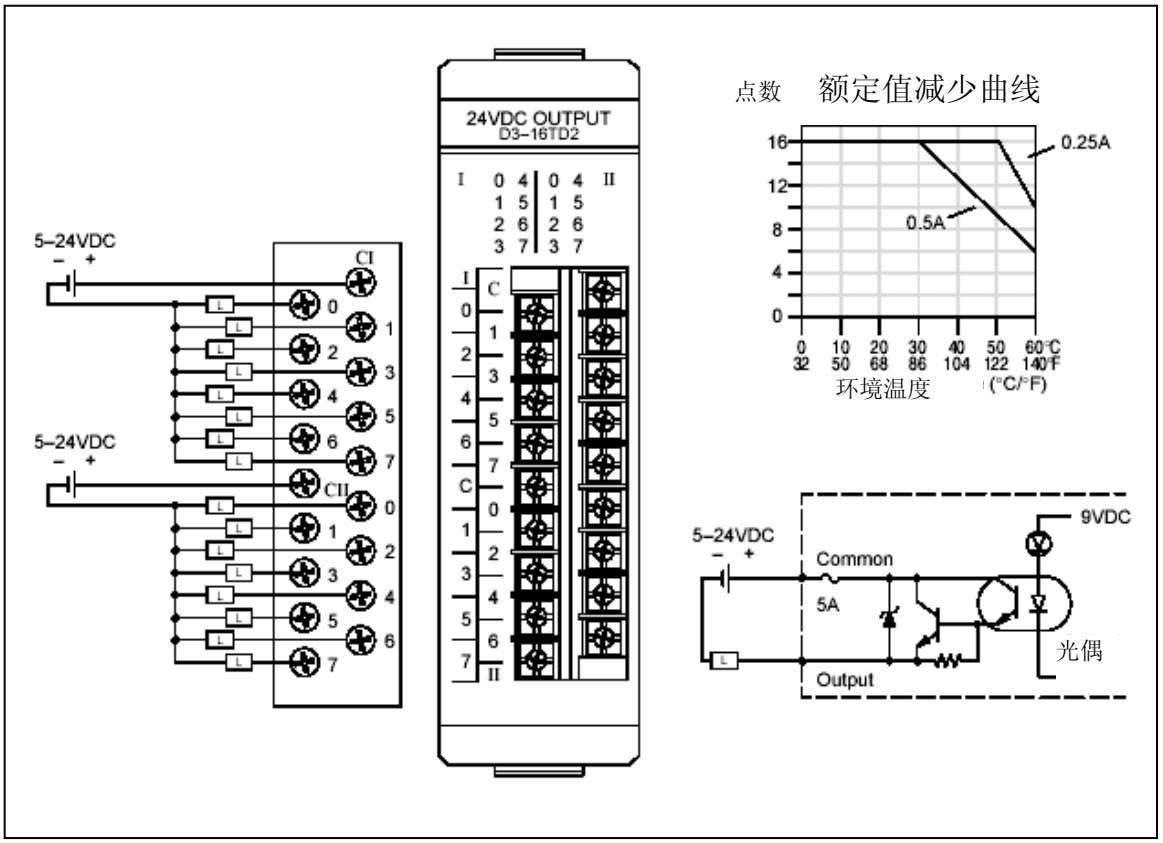
D3-16TD1-2,24VDC 输出模块

回路数	16 点（汇点）		最小负载	1mA
公共端数量	4（内部相连）		框架电流消耗	9V 最大 40mA 3mA+2.3mA/ON 每点 24V 最大 96mA 6mA/ON 每点
工作电压范围	5～24VDC			
输出类型	NPN(集电极开路)			
峰值电压	45VDC			
交流频率	N/A		OFF→ON 相应	0.1ms
ON 电压降落	0.5A 时 2V		ON→OFF 相应	0.1ms
最大电流	0.5A/点 1.8A/公共端		端子类型	可拆卸式
最大漏电流	40 VDC 时 0.3mA		状态指示灯	逻辑侧
			重量	160g
最大涌浪电流	3A/20ms 1A/100ms		保险丝	（2）每个公共端用一个 3A 保险丝（不可更换式）



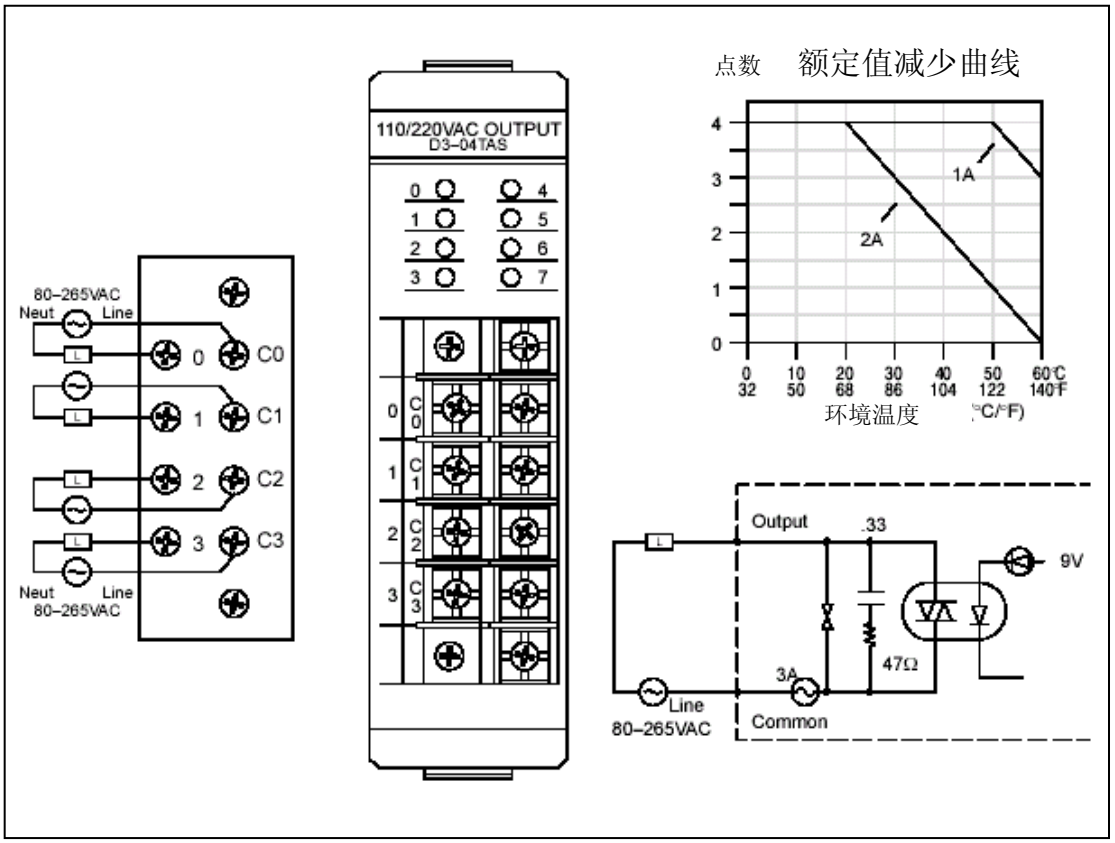
D3-16TD2,24VDC 输出模块

回路数	16 点（汇点）		最小负载	1mA
公共端数量	2（内部相连）		框架电流消耗	9V 7.5mA/ON 每点 （最大 180mA） 24V N/A
工作电压范围	5～24VDC			
输出类型	NPN(发射级跟随)			
峰值电压	40VDC		OFF→ON 相应	0.1ms
交流频率	N/A		ON→OFF 相应	1ms
ON 电压降落	0.5A 时 1.5V		端子类型	可拆卸式
最大电流	0.5A/点 3A/公共端		状态指示灯	逻辑侧
最大漏电流	40 VDC 时 0.01mA		重量	160g
最大涌浪电流	3A/20ms 1A/100ms		保险丝	（2）每个公共端用一个 3A 保险丝（不可更换式）



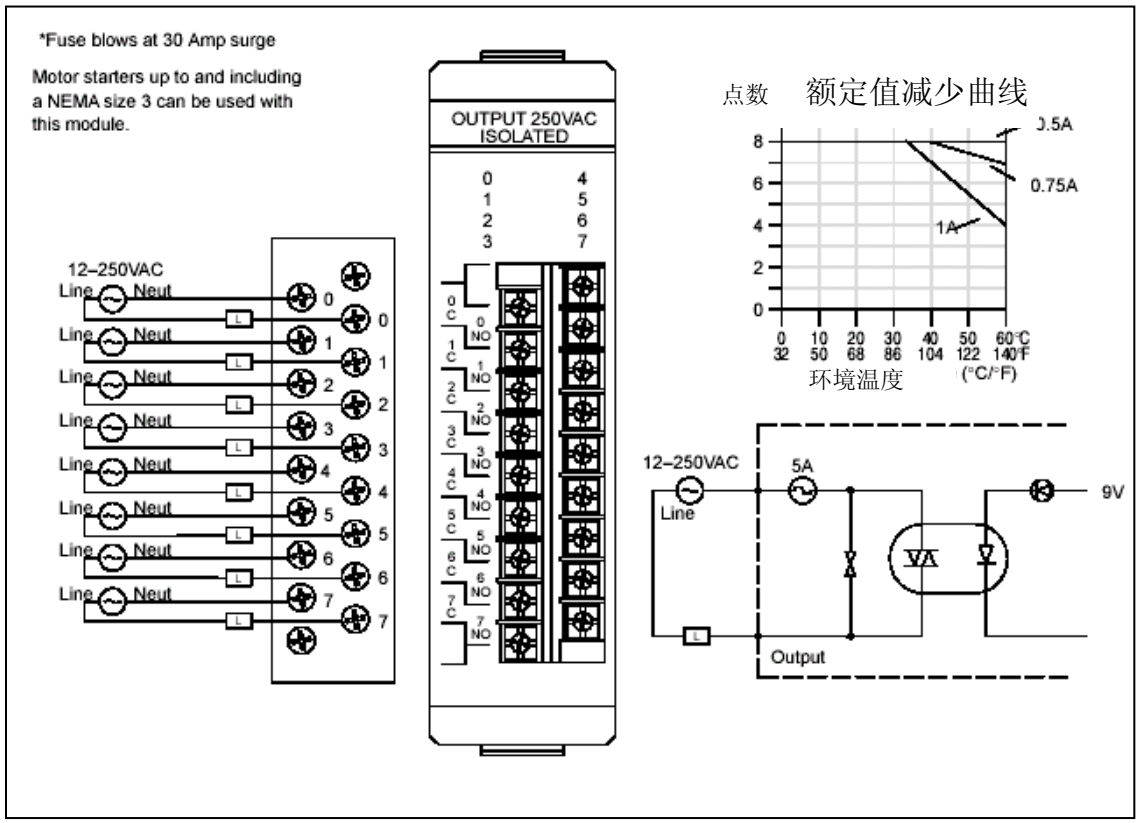
D3-04TAS,110~220VAC 输出模块

回路数	4 点		最小负载	10mA
公共端数量	4（隔离）		框架电流消耗	9V　最大 12mA 24V N/A
工作电压范围	80～265VDC			
输出类型	双向可控硅			
峰值电压	265VAC		OFF→ON 相应	1ms
交流频率	47~63Hz		ON→OFF 相应	10ms
ON 电压降落	2A 时 1.5V		端子类型	不可拆卸式
最大电流	2A/点 2A/公共端		状态指示灯	逻辑侧
最大漏电流	220VAC 时 7mA 110VAC 时 3.5mA		重量	180g
最大涌浪电流	20A/16ms 10A/100ms		保险丝	（4）每个公共端用一个 3A 保险丝（可更换式）



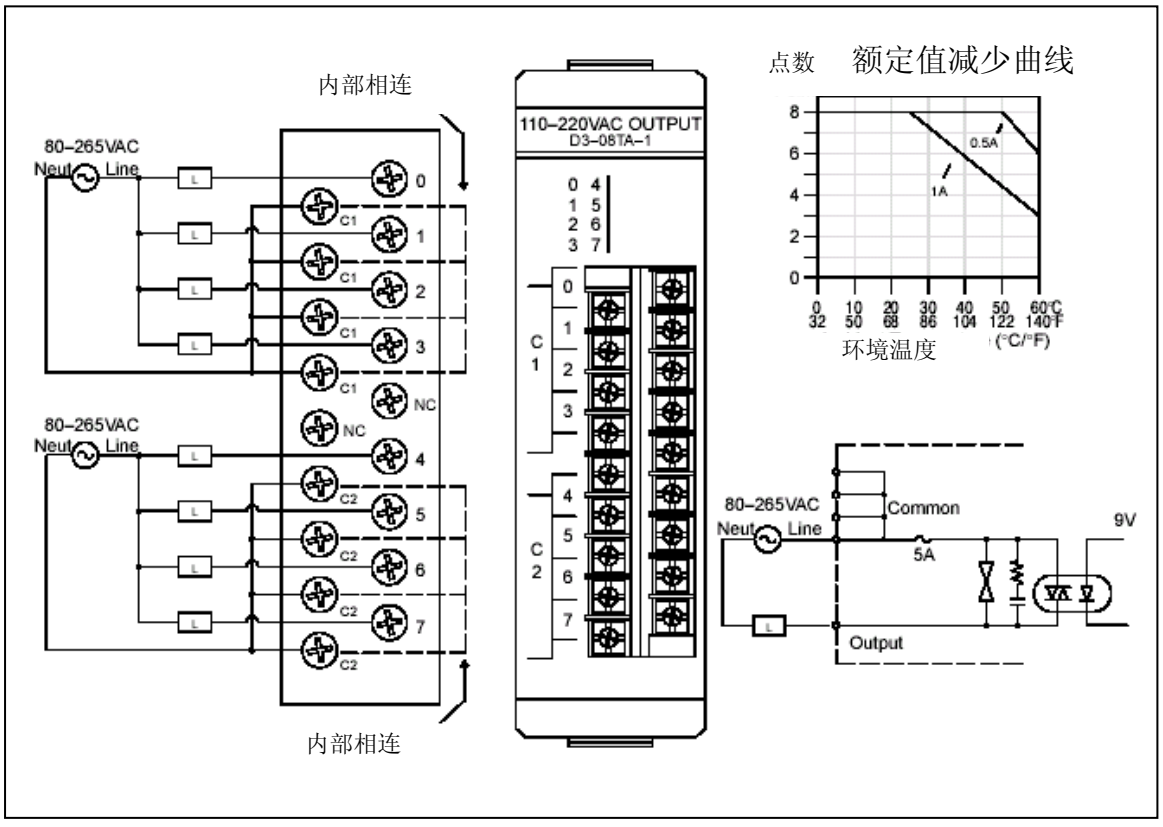
F3-08TAS,250VAC 隔离输出模块

回路数	8 点（点对点隔离）		框架电流消耗	9V 10mA/ON 每点 最大 12mA 24V N/A
公共点数量	8（隔离）			
工作电压范围	12～125VAC 125~250VAC 需要外部保险丝		OFF→ON 相应	最大 8ms
			ON→OFF 相应	最大 8ms
输出类型	双向可控硅		端子类型	可拆卸式
峰值电压	400VAC		状态指示灯	逻辑侧
交流频率	47~440Hz		重量	N/A at press time
ON 电压降落	1A 时 1VAC		保险丝 BK/PCE-5Bussman （含一个空余的保险丝）	（8）快速熔断型 每个回路有一个 5A （125V 快速熔断） 保险丝，可更换式
最大电流	1A/点			
最大漏电流	240VAC 时 10μ A			
最大涌浪电流	20A/16ms 3A/100ms			
最小负载	0.5mA			



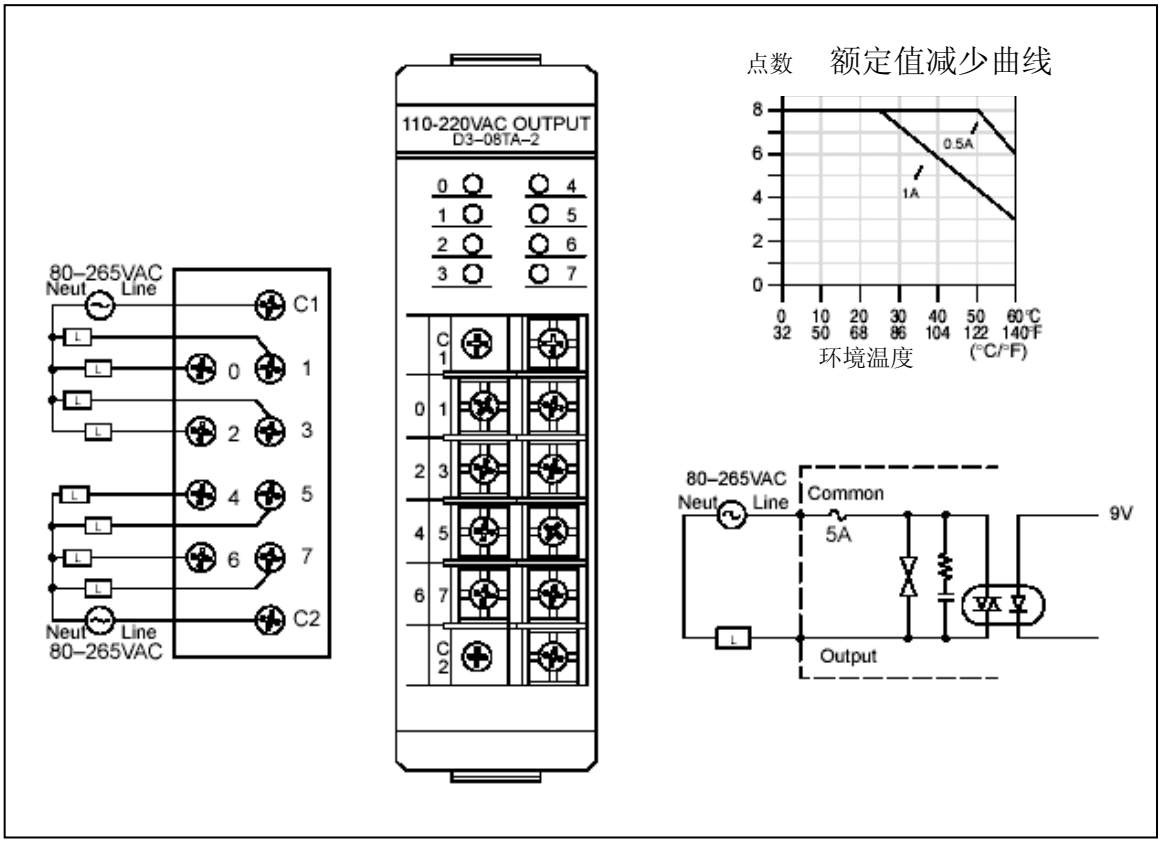
D3-08TA-1,110~220VAC 输出模块

回路数	8 点		最小负载	25mA
公共端数量	2（隔离）		框架电流消耗	9V 20mA/ON 每点（最大 160mA） 24V N/A
工作电压范围	80～265VDC			
输出类型	双向可控硅			
峰值电压	265VAC		OFF→ON 相应	最大 1ms
交流频率	47~63Hz		ON→OFF 相应	最大 8.33ms
ON 电压降落	1A 时 1.5VAC		端子类型	可拆卸式
最大电流	1A/点 3A/公共端		状态指示灯	逻辑侧
最大漏电流	220VAC 时 1.2mA 110VAC 时 0.52mA		重量	210g
最大涌浪电流	20A/16ms 5A/100ms		保险丝	（2）每个公共端用一个 5A 保险丝（不可更换式）



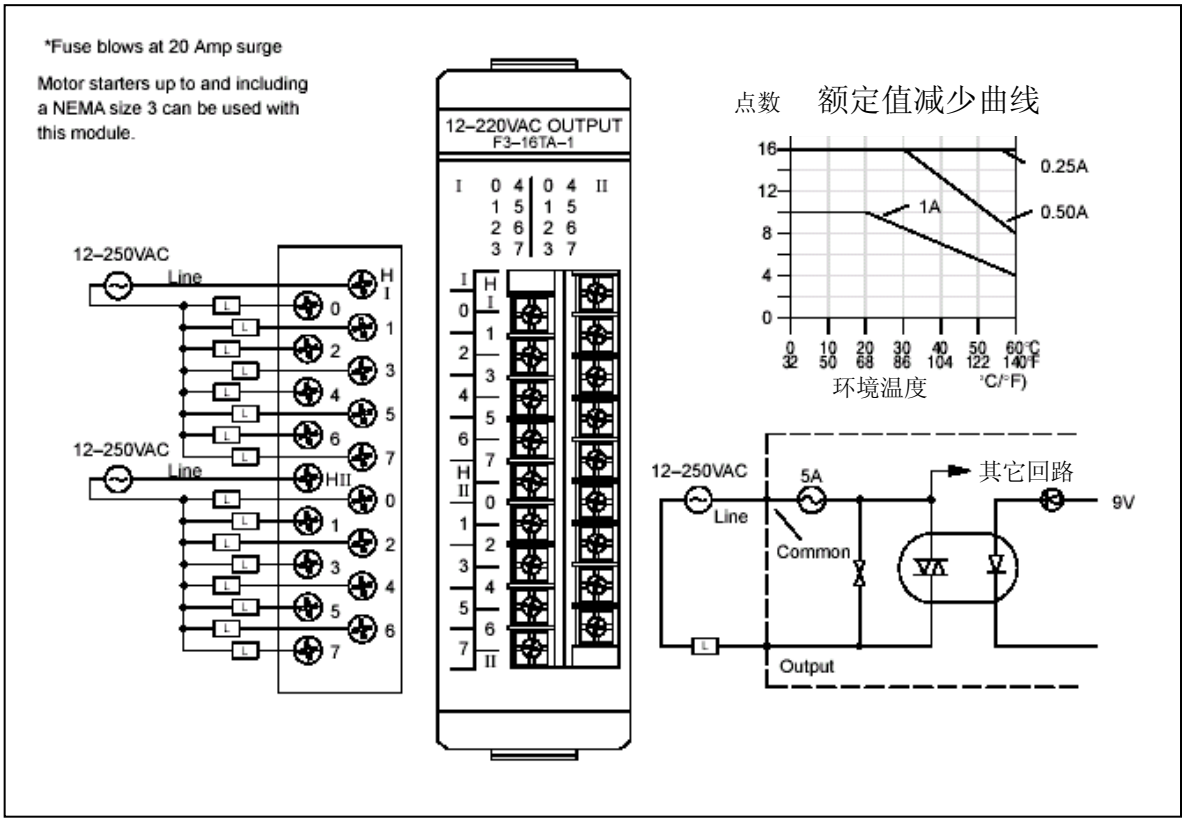
D3-08TA-2,110~220VAC 输出模块

回路数	8 点		框架电流消耗	9V 20mA/ON 每点 最大 160mA 24V N/A
公共点数量	2（隔离）			
工作电压范围	80～265VAC			
输出类型	双向可控硅		OFF→ON 相应	最大 1ms
峰值电压	265VAC		ON→OFF 相应	最大 8.33ms
交流频率	47~63Hz		端子类型	不可拆卸式
ON 电压降落	1A 时 1.5VAC		状态指示灯	逻辑侧
最大电流	1A/点 3A/公共端		量重	180g
最大漏电流	220VAC 时 1.2mA 110VAC 时 0.52mA		保险丝	（2）每个公共端使用一个 5A 保险丝，可更换式
最大涌浪电流	10A/16ms 5A/100ms			
最小负载	25mA			



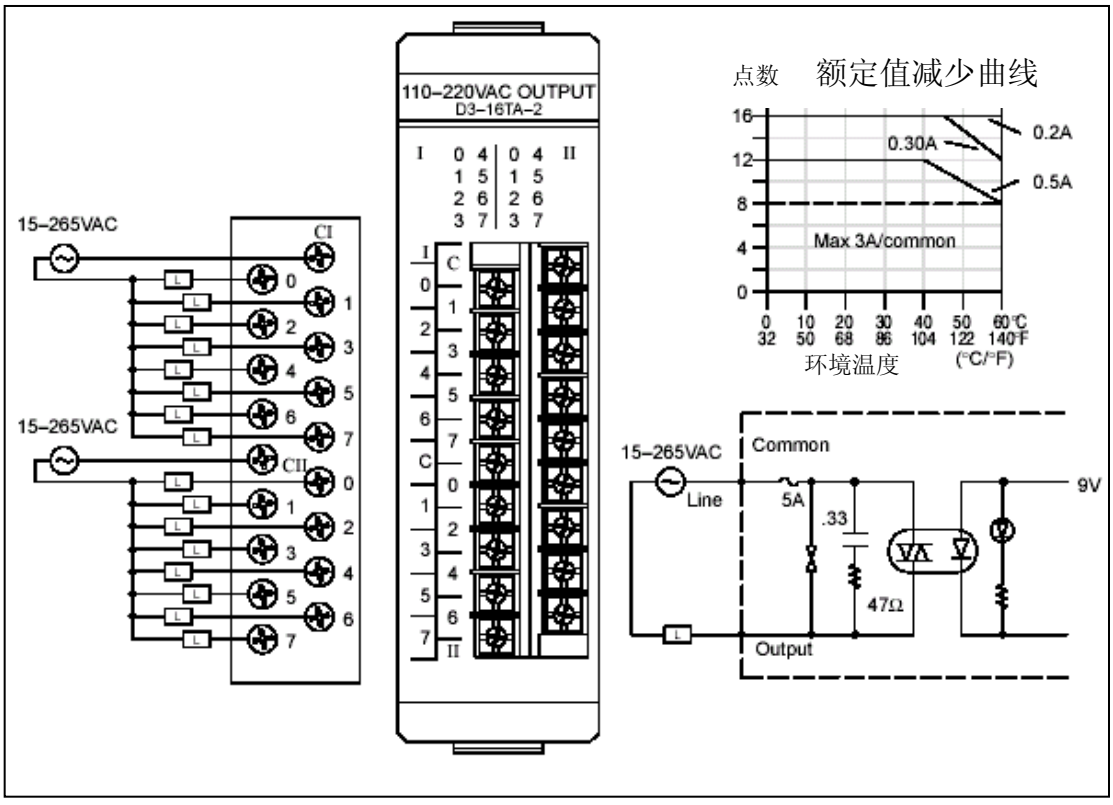
F3-16TA-1,12~220VAC 输出模块

回路数	16 点		最小负载	0.5mA
公共端数量	2（隔离）		框架电流消耗	9V 10mA/ON 每点（最大 160mA） 24V N/A
工作电压范围	12～125VAC 125～250VAC 需外部保险丝			
输出类型	双向可控硅		OFF→ON 相应	最大 8ms
峰值电压	540VAC		ON→OFF 相应	最大 8ms
交流频率	20～500Hz		端子类型	可拆卸式
ON 电压降落	1A 时 1VAC		状态指示灯	逻辑侧
最大电流	1A/点		重量	176g
最大漏电流	240VAC 时 10μ A		保险丝 BK/PCE-5Bussman（含一个空余的保险丝）	（4）每个公共端用一个 5A 保险丝（可更换式）
最大涌浪电流	20A/16ms 3A/100ms			



D3-16TA-2,15~220VAC 输出模块

回路数	16 点		最小负载	0.5mA
公共端数量	2（隔离）		框架电流消耗	9V 10mA/ON 每点（最大 160mA） 24V N/A
工作电压范围	12～125VAC 125～250VAC 需外部保险丝			
输出类型	双向可控硅		OFF→ON 相应	最大 8ms
峰值电压	540VAC		ON→OFF 相应	最大 8ms
交流频率	20～500Hz		端子类型	可拆卸式
ON 电压降落	1A 时 1VAC		状态指示灯	逻辑侧
最大电流	1A/点		重量	176g
最大漏电流	240VAC 时 10μ A		保险丝 BK/PCE-5Bussman（含一个空余的保险丝）	（4）每个公共端用一个 5A 保险丝（可更换式）
最大涌浪电流	20A/16ms 3A/100ms			

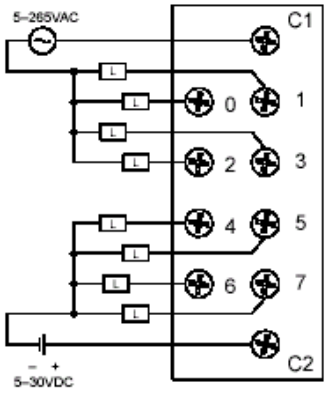


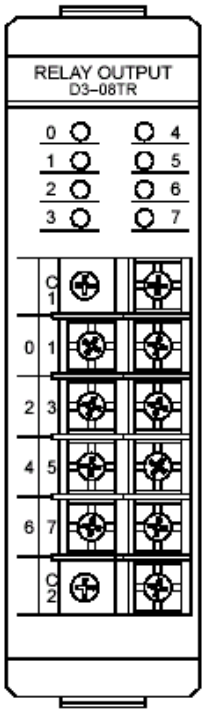
D3-08TR，继电器输出模块

回路数	8 点		最小负载	5V 时 5mA
公共端数量	2（隔离）		框架电流消耗	9V 45mA/ON 每点（最大 360mA） 24V N/A
工作电压范围	5～265VAC 5～30VDC			
输出类型	A 型（单刀单掷）		OFF→ON 相应	最大 5ms
峰值电压	265VAC/20VDC		ON→OFF 相应	最大 5ms
交流频率	47～63Hz		端子类型	不可拆卸式
ON 电压降落	N/A		状态指示灯	逻辑侧
最大电流	4A/点 AC 5A/点 DC 6A/公共端		重量	200g
最大漏电流	220VAC 时 1mA		保险丝	（2）每个公共端用一个 10A 保险丝（可更换式）
最大涌浪电流	5A			

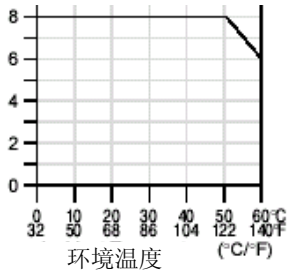
典型的继电器寿命

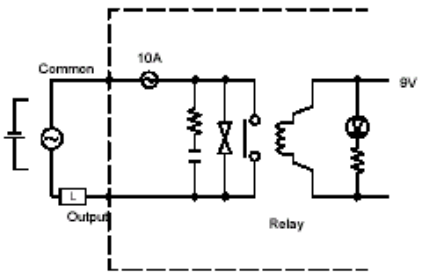
电压	电阻	电感	次数
220VAC	4A	0.5A	10 万
220VAC		0.05A	80 万
110VAC	4A	0.5A	10 万
110VAC		0.1A	65 万
24VDC	5A	0.5A	10 万





点数 额定值减少曲线



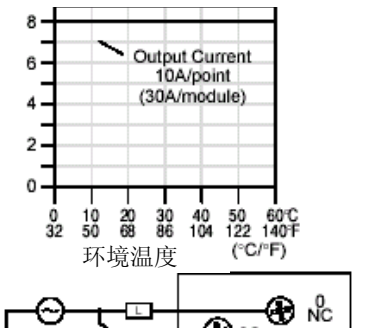


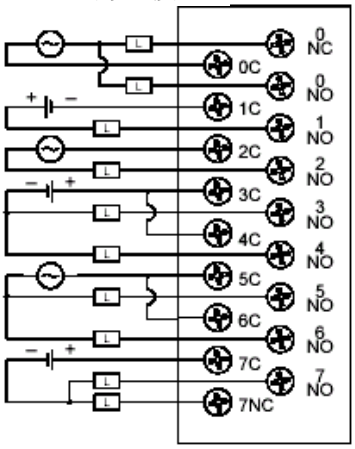
F3-08TRS-1，继电器输出模块

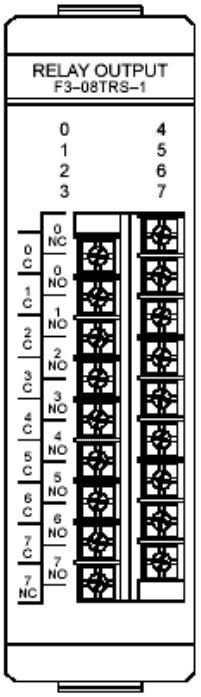
回路数	8 点		最大漏电流	N/A
公共端数量	8（隔离）		最大涌浪电流	10A　电感
工作电压范围	12～125VAC 125～250VAC 需要外部保险丝 12～30VDC		最小负载	12VDC 时 100mA
			框架电流消耗	9V　37mA/ON　每点（最大 296mA） 24V N/A
输出类型	6 点 A 型（单刀单掷） 2 点 C 型（单刀双掷）		OFF→ON 相应	最大 5ms
			ON→OFF 相应	最大 5ms
峰值电压	265VAC/120VDC		端子类型	可拆卸式
交流频率	47～63Hz		状态指示灯	逻辑侧
ON 电压降落	N/A		重量	252g
最大电流	10A/点　AC/DC 30A/模块 AC/DC		保险丝	（8）每个公共端用一个 10A(125V)保险丝（不可更换式）

注意：如果方法得当，可以适当延长接点的寿命。其技巧在本章的开头有过介绍。

点数　额定值减少曲线

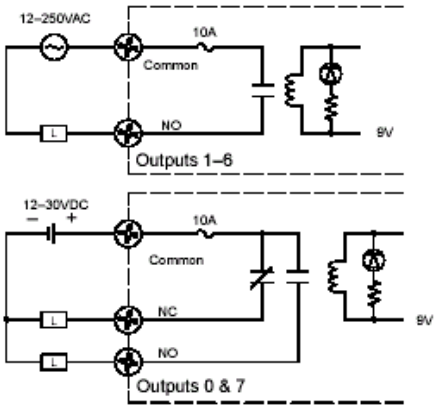






典型的继电器寿命

最大阻性或感性涌浪负载电流	工作电压		
	28VDC	120VAC	240VAC
1/4HP		25K	
10.0A	50K	50K	
5.0A	200K	100K	
3.0A	325K	125K	50K
0.05A	>50M		



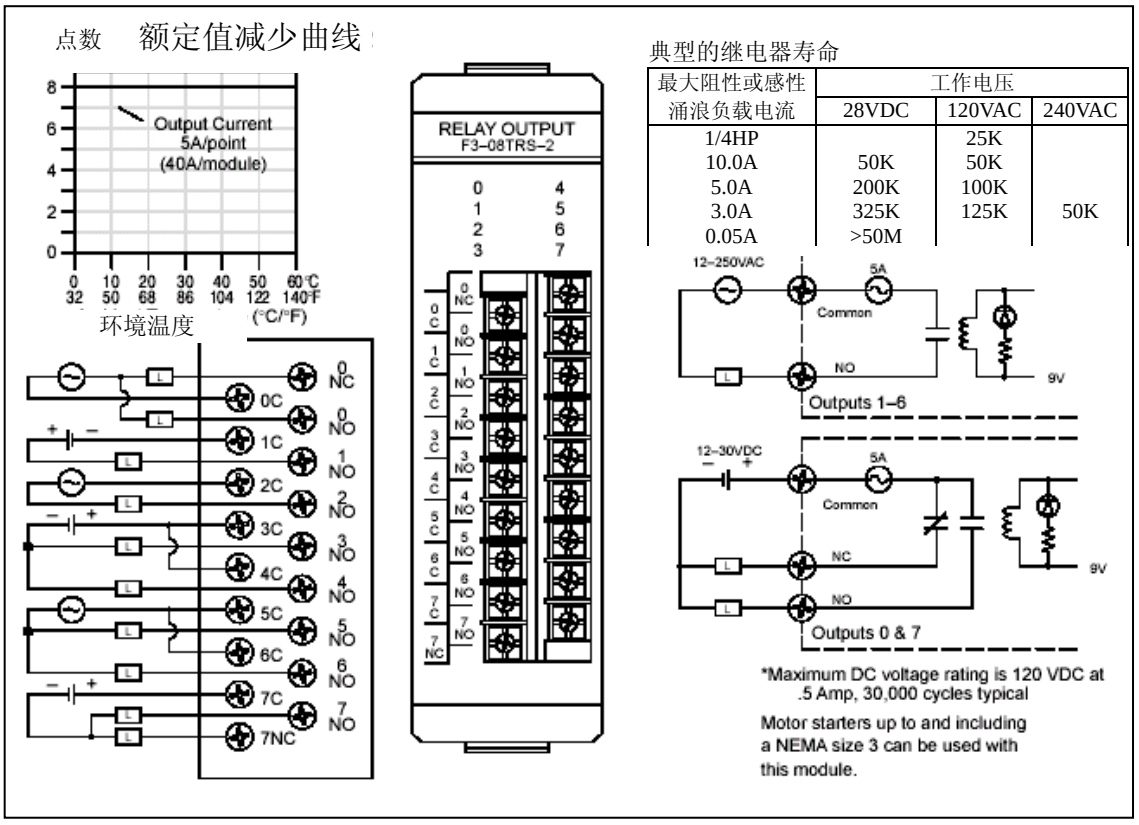
*Maximum DC voltage rating is 120 VDC at .5 Amp, 30,000 cycles typical

Motor starters up to and including a NEMA size 4 can be used with this module.

F3-08TRS-2,继电器输出模块

回路数	8 点		最大漏电流	N/A
公共端数量	8（隔离）		最大涌浪电流	10A 电感
工作电压范围	12～125VAC 12～30VDC		最小负载	12VDC 时 100mA
			框架电流消耗	9V 37mA/ON 每点（最大 296mA） 24V N/A
输出类型	6 点 A 型（单刀单掷） 2 点 C 型（单刀双掷）		OFF→ON 相应	最大 13ms
			ON→OFF 相应	最大 9ms
峰值电压	265VAC/120VDC		端子类型	可拆卸式
交流频率	47～63Hz		状态指示灯	逻辑侧
ON 电压降落	N/A		重量	255g
最大电流	5A/点 AC/DC 40A/模块 AC/DC		保险丝	（8）每个公共端用一个 5A(125V)保险丝（可更换式）

注意：如果方法得当，可以适当延长接点的寿命。其技巧在本章的开头有过介绍。

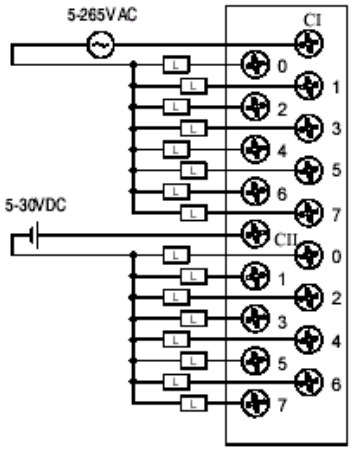


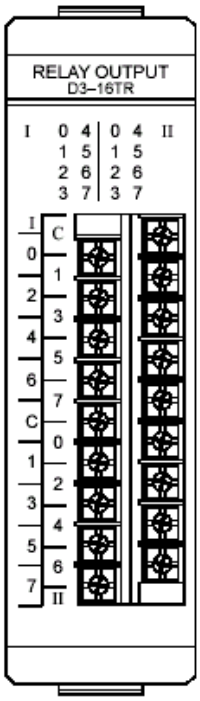
D3-16TR,继电器输出模块

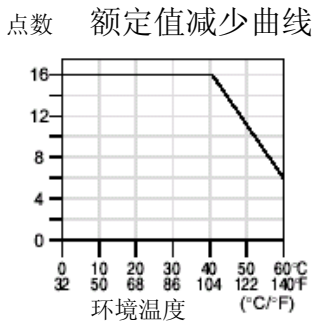
回路数	16 点		最小负载	5VDC 时 5mA
公共端数量	2（隔离）		框架电流消耗	9V 30mA/ON 每点（最大 480mA） 24V N/A
工作电压范围	5～265VAC 5～30VDC			
输出类型	16 点 A 型（单刀单掷）		OFF→ON 相应	最大 12ms
峰值电压	265VAC/30VDC		ON→OFF 相应	最大 12ms
			端子类型	可拆卸式
交流频率	47～63Hz		状态指示灯	逻辑侧
ON 电压降落	N/A		重量	248g
最大电流	2A/点 AC/DC（阻性） 8A/模块 AC/DC		保险丝	（8）每个公共端用一个 5A(125V)保险丝（可更换式）
最大漏电流	220VAC 时 0.1mA			
最大涌浪电流	2A			

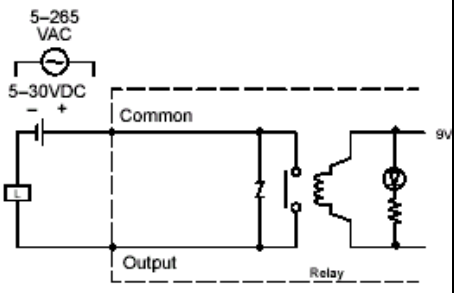
继电器典型寿命

电压	阻性	感性	次数
220VAC	2A	0.25A	100K
220VAC		0.03A	800K
110VAC	2A	0.25A	100K
110VAC		0.05A	650K
24VAC	2A	0.25A	100K









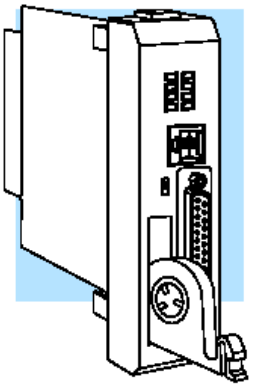
第三章　CPU 规格及其操作

3

本章将介绍：

——概要
——CPU 一般规格
——CPU 硬件特性
——使用电池备份
——选择程序存储媒体
——CPU 设置
——CPU 操作
——I/O 响应时间
——CPU 扫描周期
——PLC 数字系统
——存储器映射
——DL305 系统的 V－存储器
——I 输入/ Q 输出定义号表
——控制继电器定义号表
——Stage™ 控制/状态定义号表
——定时器/计数器状态定义号表

概要



CPU 是控制系统的心脏。几乎所有的控制系统都要由 CPU 来控制，因此 CPU 地设置和安装是否正确是十分重要的。本章提供了一些有关的信息：

- 不同的 CPU 模块之间的区别
- 设置和安装 CPU 的步骤

一般的 CPU 规格

DL350 是一个模块化的 CPU,它可以插在 5 槽、8 槽或 10 槽框架中。它支持所有 DL305 家族的 I/O 模块。DL350 CPU 提供了强大的处理功能，梯形图编程指令和级式编程指令（请参考第五章）。由于有广泛的自诊断功能，可以通过应用程序和操作界面来监控。DL350 与 DL305 家族的其它 CPU 略有不同。它支持 16 位的地址格式，而 DL330/340 是 8 位的。因此 DL350 多了一些指令和存储器空间，其特性很接近 DL205 和 DL405 的 CPU。

DL350 CPU 特性

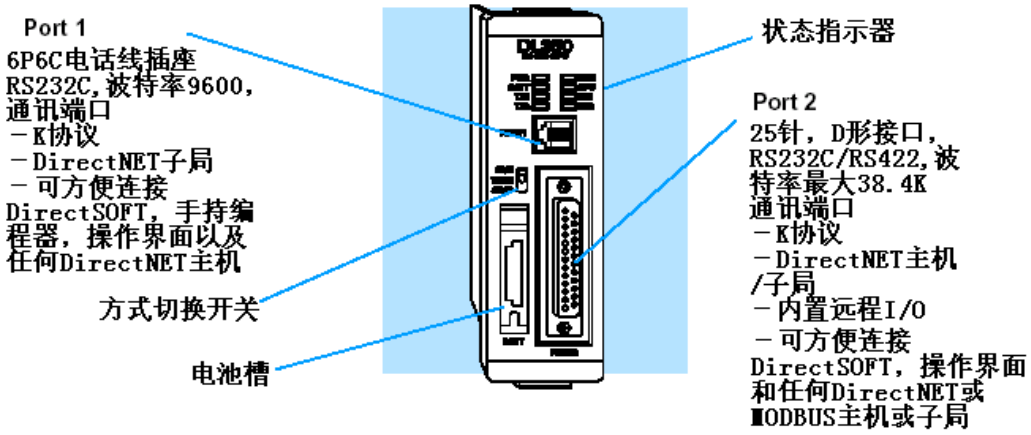
DL350 含有最大为 14.8K 的程序空间，其中 7.6K 为用户程序存储器，7.2K 为 R—存储器（数据寄存器）。它最多支持 368 点的本地 I/O 和 880 点的远端 I/O。为了增强处理性能，它还有一个内置的基于 RISC 的微处理器。DL350 有 150 条以上的指令，包括鼓形定时器，打印功能，浮点运算和 4 回路的 PID 功能。

DL350 有两个通讯口。上面一个口是 6 针型的，它是内置的 RS232 通讯口。它可以用来方便地和手持编程器，PC，或以 DirectNET 为协议的子局相连。下面一个口是 25 针的 RS232/RS422 通讯口。它可以用来连接 DirectSOFT，操作界面，内置远程 I/O，DirectNET 和 MODBUS RTU 主/从连接。

CPU 一般规格

特性	DL350
总的程序空间（字）	14．8K
用户程序空间（字）	7680（Flash）
R－存储器（字）	7168
不可挥发 R－存储器（字）	无
布尔执行/K	5－6ms
支持梯形图编程和级式编程	是
支持手持编程器	是
支持 Windows 环境下的 DirectSOFT	是
内置通讯口（RS232）	有
CMOS RAM	无
UVPROM	无
EEPROM	Flash
支持本地 I/O 点数	368
支持远程 I/O 点数	512
远程 I/O 通道	1
最多远程子局数	7
最多本地模拟量输入/输出通道	128/32
计数功能（脉冲输出、脉冲捕捉等）	无
一个模块上的 I/O 点数	8/16
槽数	5/8/10
指令数（见第五章）	170
中间继电器	1024
特殊继电器（系统定义）	144
级式编程中的最多级数	1024
定时器	256
计数器	128
直接输入输出	有
中断输入（硬件/定时）	无/有
支持子程序	是
支持鼓形定时器	是
For/Next 循环	有
算术	整数，浮点数
内置的 PID 回路	有
日历功能	有
运行时间编辑	有
内部诊断	有
口令设置	有
系统出错日志	有
用户出错日志	有
电池备份	有（可选）

CPU 硬件特性



方式切换开关

通过方式开关，可以切换成允许或禁止对 CPU 中程序的修改。除非打到 TERM 位置，在 RUN 和 STOP 位置时用界面设备（手持编程器，DirectSOFT 软件等）所作的修改都是无用的。如果打到 TERM 位置，并且没有口令的限制，包括进入用户程序在内的操作方式都是允许的（要使用编程或监控设备）。

切换位置	CPU 动作
RUN (Run Program)	如果没有遇到错误时，CPU 被强制进入 RUN 模式。不能通过编程或监控设备作任何修改。
TERM(Terminal)	此时支持 RUN,PROGRAM 和 TEST 模式。可通过编程或监控设备来修改当前模式或程序。
STOP(Stop Program)	CPU 被强制进入 STOP 方式。不可通过编程或监控设备来修改或监控。

改变 CPU 的方式有以下两种方法：

- 使用 CPU 的方式切换开关
- 把方式开关打到 TERM 位置时，可使用编程设备来改变操作方式。在此位置，您可以在 RUN 和 Program 方式之间进行切换。

状态指示器

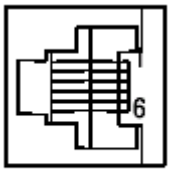
CPU 模块上面的 8 个状态指示 LED 有各自的特殊用处，它可以编程或排除故障时给予您帮助。

指示器	状态	含义
PWR	ON	电源正常
RUN	ON	CPU 在 RUN 方式
CPU	ON	CPU 自诊断出错
BATT	ON	CPU 电池电力不足
TX1	ON	端口 1 发送数据
RX1	ON	端口 1 接受数据
TX2	ON	端口 2 发送数据
RX2	ON	端口 2 接受数据

端口 1 规格

以下是 DL350 上的端口 1 的操作参数：

- 6 针电话插座（RJ12 电话插座）
- DirectNET（子局），K 协议
- RS232C,9600 波特率
- 可连接 DirectSOFT,D2-HPP,DV1000 或者 DirectNET 主机



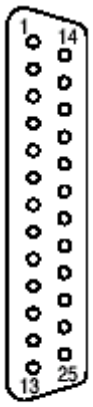
6针插座

端口 1 管脚描述（只限 DL350）		
1	0V	电源负极（即 GND）
2	5V	电源正极
3	RXD	接受数据（RS232C）
4	TXD	发送数据（RS232C）
5	5V	电源正极
6	0V	电源负极（即 GND）

端口 2 规格

DL350 的端口 2 是 25 针的 D 形接口。它可以用编程设备上的 AUX 功能来设置。

- 25 针插座接口
- 协议：K 协议，DirectNET 主机/子局，MODBUS RTU 主机 /子局，远程 I/O，无协议
- RS232C，非隔离，距离不超过 15m
- RS422C，非隔离，距离不超过 1000m
- 最大支持 38.4 波特
- 地址可选（1－90）
- 可连接 DirectSOFT，操作界面，和任何 DirectNET 或 MODBUS 的主机或子局。



25针D形插座

端口 2 管脚说明（DL350 CPU）		
1	未使用	
2	TXD	发送数据（RS232C）
3	RXD	接收数据（RS232C）
4	RTS	准备发送（RS232C）
5	CTS	清除发送
6	未使用	
7	0V	电源负极
8	0V	电源负极
9	RXD+	接收数据+(RS422)
10	RXD-	接收数据-(RS422)
11	CTS+	清除发送+(RS422)
12	TXD+	发送数据+
13	TXD-	发送数据-

端口 2 管脚说明（续）
14 TXD+ 发送数据+(RS422)
15 未使用
16 TXD－发送数据－(RS422)
17 未使用
18 RTS－ 请求发送－(RS422)
19 RTS+ 请求发送+(RS422)
20 未使用
21 未使用
22 未使用
23 CTS－ 清除发送－(RS422)
24 RXD＋ 接收数据＋
25 RXD－ 接收数据－

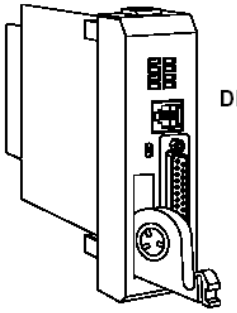
使用电池备份

当 DL350 没有外部电源时，一个备选的锂电池可用来维持系统的 RAM 的信息。一般的 CPU 电池的寿命为 5 年，这也是和一般的工作寿命相接近。但是，如果您已很久没有更换电池而且系统将要停用超过 10 天，也要考虑更换新的电池。

注意：在安装或更换您的 CPU 电池时，请备份您的 R—存储器数据和系统参数。您可以通过 DirectSOFT 来保存程序、R—存储器数据和系统参数到您的硬盘或软盘。

请按以下步骤来安装 D3-BAT CPU 电池：

1. 按下电池门上的卡位片并向外拉
2. 把新的电池放入电池盒
3. 关上电池门并确保到位
4. 对电池的更换日前作一个记录

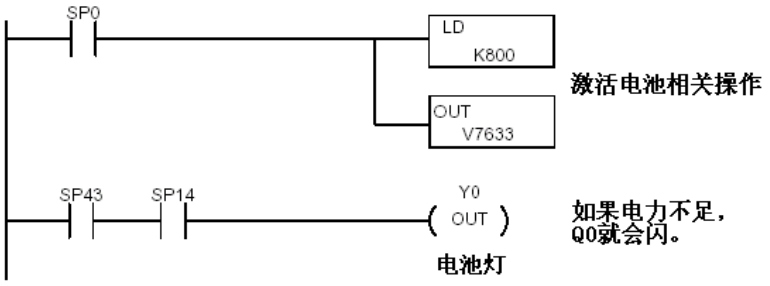


警告：不要尝试给电池充电或把旧电池扔到火中。因为电池有可能爆炸或释放出有害物质。

激活电池备份

可以通过给 R7633 中的第 12 位置位来激活电池（见下面的例子）。在此模式下，当电池电压低于 2.5VDC 时（SP43），电力不足指示灯就会亮而且出现 E41 报错。在此模式下，倘若电池是正常的，当断电时，CPU 会保持 M,S,T,C 以及 R 寄存器中的数据。电池的使用也可以决定当系统再次通电时会进入哪一种操作模式。详情请见后面的章节——CPU 的设置。

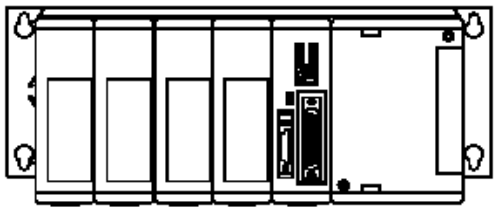
当您已经安装完了电池，电池回路也可以通过 R7633 中的第 12 位复位来禁止导通。但是，如果您已经安装了电池并选择了“No Battery”操作，则电池指示灯在电池电压很低的情况下也不会亮。



CPU 的设置

安装 CPU

CPU 模块**必须**安装在框架的第一槽中（最靠近电源的地方）。绝对不能插在其它槽中。要插入 CPU 模块时，先把模块的电路板对准插槽的上部和下部。然后垂直地插入其中直到接触牢固可靠。



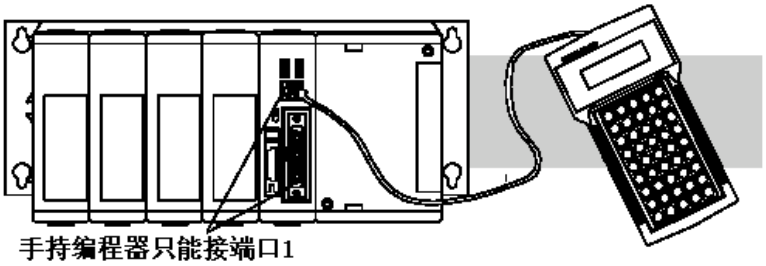
CPU模块必须安装在第一槽中！



警告：为了避免触电，人身伤害或机器损坏，在安装或拆除系统的任何组件时都要首先给系统断电。

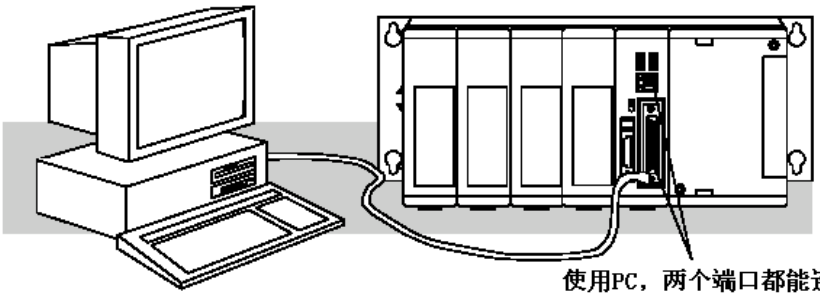
连接编程设备

手持编程器可通过手持编程器电缆来连接 CPU。您可以通过端口 1 连接手持编程器。其电缆是和手持编程器包装在一起的。此电缆约为 2m 长。



手持编程器只能接端口1

如果您通过一台装有 DirectSOFT 软件的 PC 来连接 CPU,那么，上下两个端口都可以使用。



使用PC，两个端口都能连

辅助功能

许多 CPU 的设置会涉及到辅助（AUX）功能的使用。AUX 功能提供了许多不同的操作，例如：清除程序存储器，显示扫描时间，拷贝程序到手持编程器的 EEPROM 中等。它们按照不同的系统参数来分类。附录 A 提供了 AUX 功能的描述。

您可以通过 DirectSOFT 或手持编程器来使用这些 AUX 功能。以上两种产品的用户手册都提供了使用 AUX 功能的步骤。其中有一些 AUX 功能是专门为手持编程器设计的，因此它们在 DirectSOFT 中也是无效的。下面的表格列举了对于不同的 CPU 和手持编程器的 AUX 功能。注意，手持编程器可能含有一些 DL350 CPU 不支持的 AUX 功能。

AUX 功能及其描述		350	HPP
AUX 2*—程序操作			
21	检查程序	✓	—
22	修改参数	⊗	—
23	清除程序块	✓	—
24	清除所有梯形图	✓	—
AUX 3*—R 寄存器操作			
31	清除 R 寄存器	✓	—
AUX 4*—I/O 设置			
41	显示 I/O 设置	✓	—
42	I/O 诊断	⊗	—
44	上电时 I/O 设置检查	⊗	—
45	选择设置	⊗	—
AUX 5*—CPU 设置			
51	修改程序名	✓	—
52	显示/修改日历	✓	—
53	显示扫描时间	✓	—
54	参数初始化	✓	—
55	监控定时器设定	✓	—
56	局号设定	✓	—
57	停电保持区域设定	✓	—
58	测试操作	⊗	—
59	位覆盖功能	⊗	—
5B	计数器界面设置	⊗	—
5C	显示出错日志	✓	—

AUX 功能及其描述		350	HPP
AUX 6*—手持编程器设置			
61	显示版本号	✓	
62	蜂鸣器 ON/OFF	⊗	✓
65	自诊断	⊗	✓
AUX 7*—EEPROM 操作			
71	读出	⊗	✓
72	写入	⊗	✓
73	比较	⊗	✓
74	空检查	⊗	✓
75	全清	⊗	✓
76	EEPROM 类型读出	⊗	✓
AUX 8*—口令操作			
81	修改口令	✓	—
82	口令打开	✓	—
83	口令关闭	✓	—

✓ 支持
⊗ 不支持
— 无效

清除已有的程序

在您写入一个新的程序之前，需要清除程序存储器中的内容。您可以使用 24 号功能来清除所有的程序。
您也可以通过其它的方法来清除其它的存储器空间。

- AUX 23—清除程序块
- AUX 24—清除所有程序
- AUX 31—清除 R 寄存器内容

设定时钟和日历

DL305 的时钟/日历功能可以达到多种目的。如果您需要用到这些功能，一些 AUX 功能可以用来设置相应的日期和时间。例如，您可以通过手持编程器的 AUX 52，显示/改变日历来改变时间和日期。如果要用 DirectSOFT，您可以使用 PLC 设置相关菜单，而且只能使用 K—协议。
CPU 使用以下的显示格式：

- 日期—年，月，日，星期（0—6，表示星期天—星期六）
- 时间—24 小时格式，时，分，秒

手持编程器显示

```
      23:08:17  97/05/20
```

存储器初始化

您可以通过 AUX 功能来改变任何日期或时间的组件。但是，但 CPU 不会自动纠正日期和星期之间的差异。比如，您把日期改到某月的 15 日，且这一天是星期四，您还必须再设定星期几（除非 CPU 本来就显示的是星期四）。星期几的设定只能通过手持编程器来完成。
DL350 CPU 把所涉及到的存储区域的系统参数像“中间结果暂存器”一样保留住。有时候，您对系统作的一些修改会被存在系统的存储器中。例如，如果您指定了一个中间继电器区域作为保留的，那么这些都会被存起来。
AUX 54 可以系统的存储器恢复成默认值。



警告：除非您要清除系统存储器中的所有设置，否则千万不要使用此功能。通常，如果您要改变程序或者旧的程序需要一些特殊的系统设置时，您只要把系统的存储空间初始化一下即可。

记住，此 AUX 功能将给所有的系统存储器复位。如果您已经设置了特殊的参数，比如保留区域等，它们都会被覆盖掉当 AUX 54 功能执行的时候。确保您在作此操作前已经考虑到了所有的后果。

设置 CPU 的局号

DL350 有一个内置的 DirectNET 端口。您可以使用通过手持编程器来设置局号和端口的通讯参数。默认设置如下：

- 局地址 1
- HEX 模式
- 奇校验
- 9600 波特率

DirectNET 的手册中有关于联网操作的通讯设置的详细说明。

设置保留存储区域

DL305 系统保留了一些存储空间。其默认的范围在许多情况下都适用，如果您的系统需要更多的保留空间或不需要保留空间，您也可以把它们改掉。其默认设置如下：

存储区域	DL305	
	默认范围	可用范围
中间继电器	M1000~M1777	M0~M1777
R 寄存器	R1400~R37777	R0~R37777
定时器	无	T0~T377
计数器	C0~C17	C0~C177
级	无	S0~S177

您可以通过 57 号功能来设置保留区域。同时您也可以用 DirectSOFT 的菜单来选择保留区域。



警告：DL305 CPU 不包含电池。其内部的大电容可以使数据在断电后短期内不挥发。如果保留区域的内容对您十分重要的话，您必须使用电池。

口令保护

DL305 CPU 所含有的口令功能帮助您尽量减少程序被非法复制或更改的可能性。DL305 CPU 提供了多重口令来进一步提高安全性。设置了密码以后，您就可以“锁住”CPU。当 CPU 被锁以后，在使用编程设备来改变任何系统参数之前就需要先输入口令。

您可以选择 8 位数字作为口令。CPU 的出厂口令为 00000000。全零可以取消口令保护。如果已经设置了新的口令，则无法通过输入 8 个“0”来进入。设置了口令以后，您也可以把口令重新设成 8 个“0”来去掉口令保护功能。

要了解有关口令的详细信息，请见相关附录。



警告：请您务必牢记口令。如果您忘记了口令，就不能对 CPU 进行操作。CPU 必须返回到厂家进行恢复。

CPU 操作

要使您的设备或程序达到一定的控制目标，您必须清楚地了解 CPU 是如何工作的。本节我们将研究 CPU 工作的四个方面。

- CPU 操作系统—CPU 管理系统控制的所有方面。
- CPU 工作模式—有三种基本模式，分别是 Program,Run 和 Test。
- CPU 定时—I/O 响应时间和 CPU 扫描周期。
- CPU 功能寄存器编号表—提供了各种系统资源的地址表，比如，T,C,I,Q 等。

CPU 操作系统

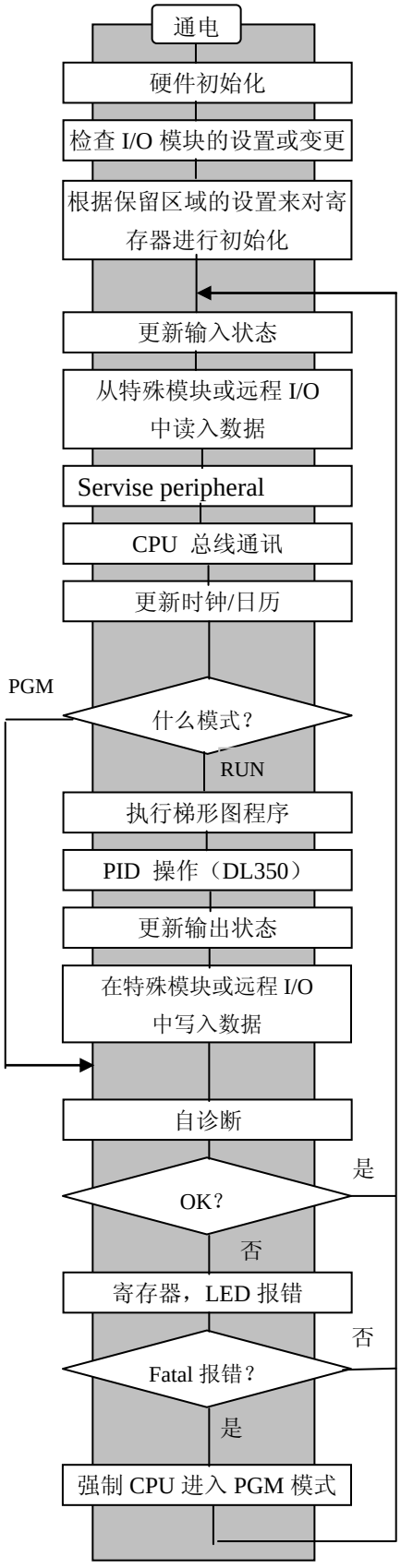
通电时，CPU 先对内部硬件作初始化。存储器在初始化时同时检测存储器设置。通常，除了保留区域，其他的存储器都清零（除非特殊情况）。

在完成了通电初始任务后，CPU 开始循环扫描。扫描流程图如右图所示。“扫描周期”是循环扫描的平均时间。请注意即使 CPU 在 Program 状态，也会不断读取输入信息。这就可以通过编程设备来随时监控输入状态。

输出的状态只在 RUN 模式下才改变。在 PROGRAM 模式下，它们都是关断的。

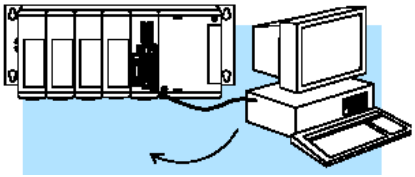
在 RUN 模式下，CPU 执行用户程序。紧接着，之前设置的 PID 回路开始工作（只限 DL305）。然后 CPU 把以上两个任务的结果立即写入输出点。

检错功能有两个等级。出现 Non-fatal 报错时，CPU 维持原运行状态。出现 Fatal 报错时，CPU 被强制进入 PROGRAM 模式，而且所有的输出都关断。



PROGRAM（编程）模式操作

在PROGRAM 模式中，CPU 并不执行程序或刷新输出状态。PROGRAM 模式主要是用来输入或修改您的用户程序。同时您也可以使用该模式来设置 CPU 的参数，比如局号，保留寄存器区域等。



程序下载

您可以通过模式开关来选择 DL350 CPU 进入 PROGRAM 模式。或者也可以先选择 TERM 模式，然后用手持编程器等编程设备来使 CPU 进入 PROGRAM 模式。

RUN（运行）模式操作

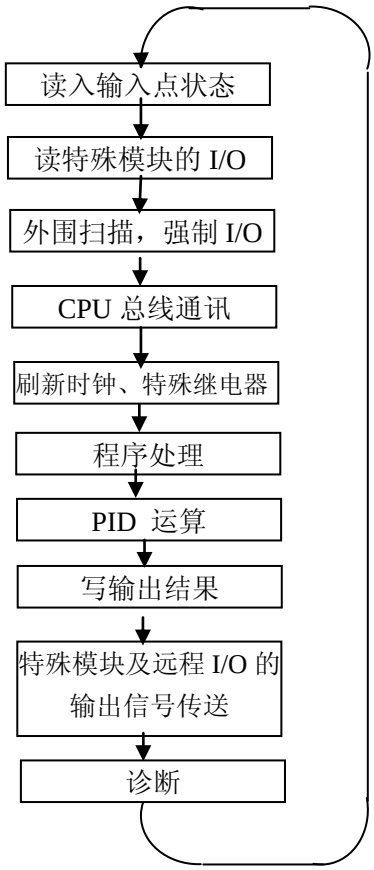
在 RUN 模式下，CPU 执行用户程序，作 PID 回路运算（只限 DL350），以及刷新 I/O 系统。

在 RUN 模式期间，您可以进行多种操作。包括：

- 监控和改变 I/O 点的状态
- 刷新 T/C 的预设值
- 刷新变量存储器

RUN 模式操作可以分为主要的几个部分。了解每个部分是如何来影响您的最终结果的，这是非常重要的。

您可以通过模式开关来选择 RUN 模式。或者也可以先选择 TERM 模式，然后用手持编程器等编程设备来让 CPU 进入 RUN 模式。



您也可以在 RUN 模式下来修改您的程序。RUN 模式下的修改并不是“bumpless”。相反，CPU 将维持上一次输出的状态而同时接受新的指令信息。如果程序中出现错误，CPU 会关断所有的输出并进入 PROGRAM 模式。



警告：只有授权个人完全熟悉系统的每一个部分才能修改程序。RUN 模式下的编程变得更快更有效。修改之前您应该彻底考虑清楚它有可能带来的影响并尽量减小人身伤害和设备损坏的可能性。

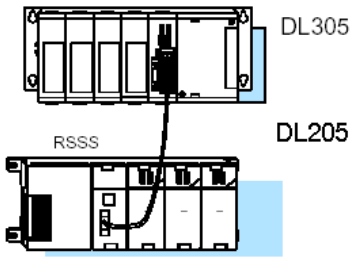
读输入状态

CPU 读取输入点的状态，然后存放到相应的输入点寄存器中。输入点寄存器位置是寄存器号后加一个“X”。当处理完了应用程序，这些寄存器又被 CPU 占用。

当然，输入点的状态可能在 CPU 读取以后发生变化。一般来说，CPU 的扫描周期是 ms 级的。如果您的输入变化在 CPU 的下一次扫描之前，那么您可以使用直接传送指令。这些指令不使用输入点寄存器的状态，而是直接从 I/O 模块上读取输入点的状态。尽管如此，这样还是会延长 CPU 的扫描时间，因为 CPU 必须再次读输入点的状态。有关直接指令的详情请参阅第五章的相关内容。

读特殊模块和远程 I/O 的信息

CPU 在读取输入点的状态以后，它再从任何安装在系统内的特殊模块中读入其信息。这同样也会占用一定的扫描周期。





注意：有可能每次扫描期间，远程 I/O 的状态都会刷新一次。这不太对。每个扫描周期内，CPU 从远程 I/O 的主机中获取相应的信息，但那个主机不一定能从其子局中得到刷新信息。

外围设备检测和强制 I/O

在 CPU 从输入模块中读取信息之后，它会接着检测外部设备。这是对任何已连接的外围设备作基本的通讯。例如，它会读取编程设备的信息，来判断是否有输入点、输出点或其它寄存器的状态发生了变化。

- 外围设备的强制信号—不是持续性的强制，只在一个扫描周期内有效

常规强制——这种强制可以临时对离散的某个位的状态进行操作。例如，您可能需要让某个输入点成为 ON，尽管它实际上是 OFF 的。您可以改变存在于 I/O 映射表中的输入/输出点的状态。其值可以维持下去直到下一个扫描周期的来临。当您想用一个信号来触发一个事件，这是一个最有效也是最基本的测试方法。

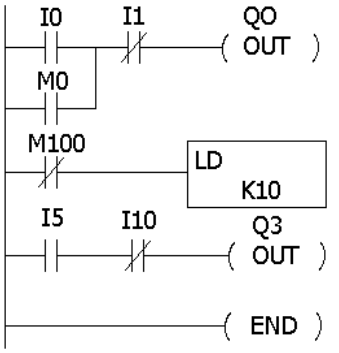
刷新时钟，特殊继电器和特殊寄存器

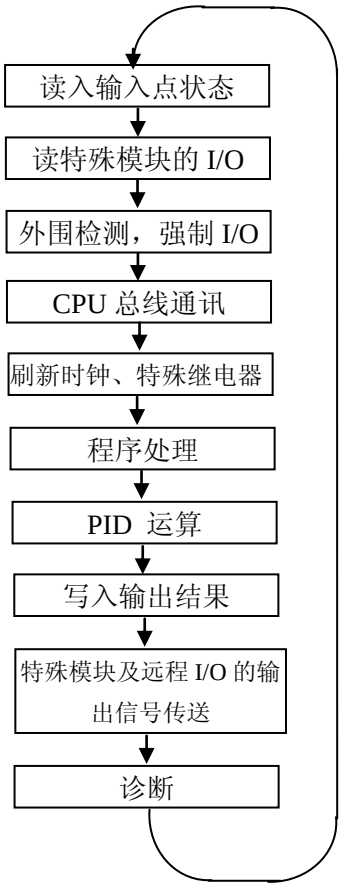
DL350 CPU 有一个内置的可以在用户程序中引用的时钟和日历。相关的时间信息存放在某几个 R 寄存器中。循环执行的这一部分确保了这一区域的值在每次扫描后更新一次。而且，有一些特殊继电器，比如自诊断继电器等的状态也不断更新。

处理应用程序

CPU 在对整个程序循环扫描时不断计算每个指令的结果。指令定义了输入条件和系统输出之间的关系。

CPU 的扫描顺序是从左到右，从上到下逐句运算直到遇到“END”。到那一刻，一个新的输出映射表就完成了。





在这段中，内部中间继电器，级，以及各种 R 寄存器的值都会刷新。您也许会想到 CPU 在扫描外围设备的时候可能已经获得并存储了一些强制信息。如果任何的 I/O 点或寄存器的状态被强制改变，其相应的输出映射表也会包含同样的信息。



注意：如果您的程序中用到了某个输出点，程序的运行结果会覆盖掉那些强制信号。例如，您用编程设备强制 Q0 为 ON，而程序中某一句的运算结果是 Q0 为 OFF，则在输出映射表中 Q0 显示为 OFF。当然，您也可以强制那些不参与程序运算的输出点。此时，强制信号会维持原样，因为没有任何运算结果来影响它。

PID 回路处理

DL350 CPU 支持 4 个 PID 回路。其 PID 运算是在梯形图程序处理完以后执行的，因此它是一个相对独立的任务。而且只有被设置过的回路才会根据内置的回路时间表执行运算。每个回路的采样时间都是可设定的。有关 PID 回路操作的详情，请参阅第八章的相关内容。

写入输出结果

当程序运算出结果并写入相应的寄存器中后，CPU 就把这些信息写到相应的本地框架的输出点上。请记住，CPU 也会把强制信号写入输出映射表，因此，被强制的点也会同时被刷新。

特殊模块及远程 I/O 的输出信号传送

CPU 刷新了本地和扩展框架的输出点之后，它就开始传送任何特殊模块所需要的输出点信息。例如，在这一扫描段中，把输出点的信息从输出映射表写入远程 I/O 框架中。



注意：看起来远程 I/O 点的状态每个扫描周期都会改变。其实不然。CPU 每个扫描周期会把信息传送到远程 I/O 的主局中。但远程主局会在主模块和从模块的下一次通讯的时候刷新远程模块的状态。请记住，远程 I/O 的通讯联接由远程主局（REMOTE MASTER）控制，而非 CPU。

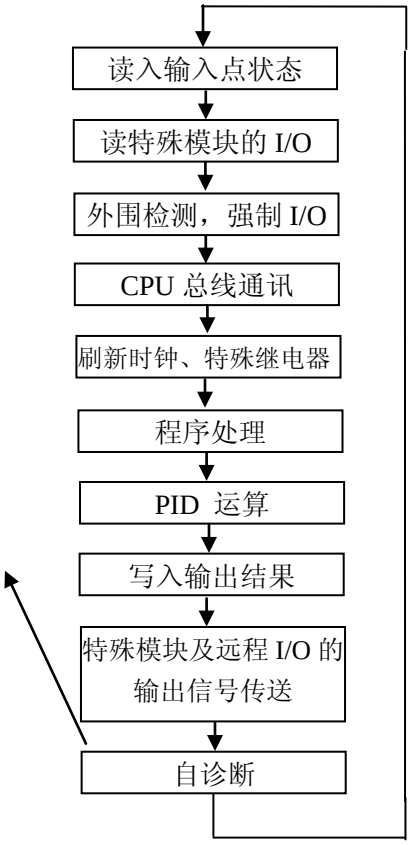
自诊断

扫描到此处时，CPU 进行所有的系统自诊断以及其他任务，例如：

- 计算扫描周期
- 刷新特殊继电器
- 看门狗定时器复位

DL350 CPU 可以自动检测并报告各种出错。附录 B 提供了 DL305 系统的错误代码表。

扫描周期的计算和看门狗定时器控制是尤为重要的自诊断功能。DL350 CPU 含有一个“看门狗”定时器，其中包含了 CPU 所允许的扫描周期的时间上限。出厂默认值为 200ms。假如扫描周期超过这个值，那么 CPU 就会进入 PROGRAM 模式，关断所有的输出点并报错。例如，当发现扫描超时，手持编程器会显示“E003 S/W TIMEOUT”。



您可以用 53 号功能来查看最小、最大、及当前扫描周期。也可以用 55 号功能来改变看门狗定时器的值。另外，您还可以在程序中使用 RSTWT 指令，在循环扫描的过程中复位看门狗定时器的值。

I/O 响应时间

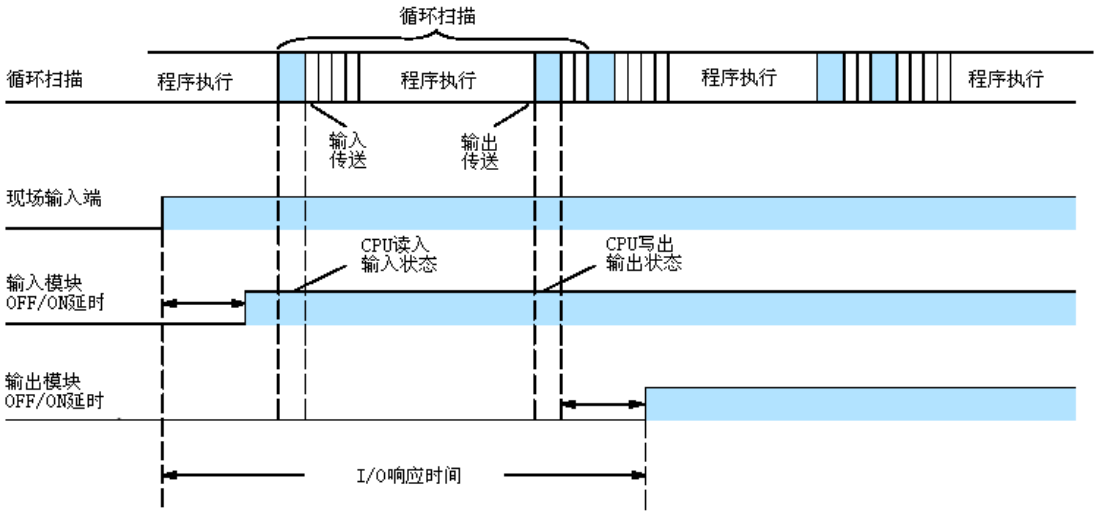
时间在您的应用中重要吗？

I/O 响应时间指的是控制系统从外界感知输入点的信号并刷新相应的输出点的状态所需要的时间。在大多数的应用中，CPU 在瞬间就能完成这个任务。但是，有些应用需要非常快的刷新时间。影响 I/O 响应时间的因素有以下 4 种：

- 输入点经过扫描的时候，其现场端发生了变化
- 输入模块从 OFF 到 ON 的延时
- CPU 的扫描周期
- 输出模块从 OFF 到 ON 的延时

通常最小的 I/O 响应

执行输入传送之前模块上就感知了输入的变化，此时输入响应是最短的。在这种情况下，输入状态被读入，然后执行程序并执行输出传送。以下是该情况的时序图。



在这种情况下，您可以按以下式子计算出响应时间。

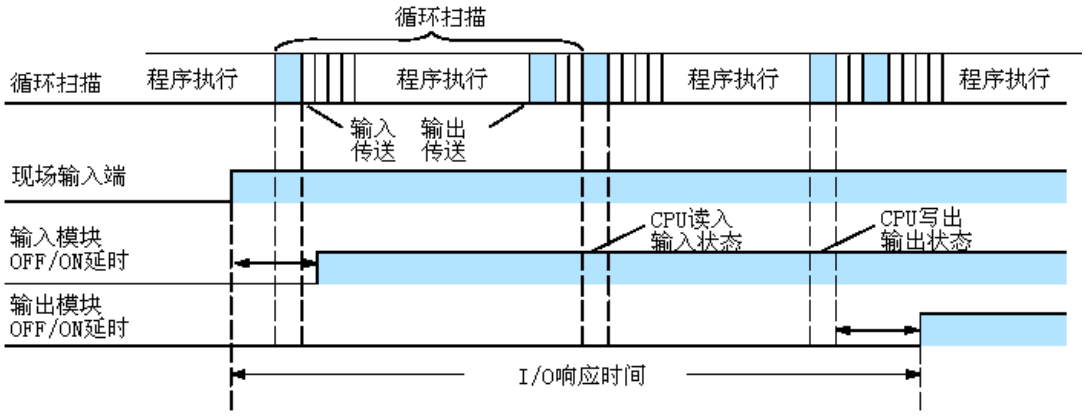
输入延时 + 扫描时间 + 输出延时 = 响应时间

通常最大 I/O 响应

执行了输入传送之后模块上才有输入状态的变化，此时 I/O 响应时间是最长的。这种情况下，输入状态表要在下一个周期中才发生变化。下面是这种情况的时序图。

在这种情况下，您可以按以下式子计算出响应时间。

输入延时 + （2×扫描周期）+ 输出延时 = 响应时间

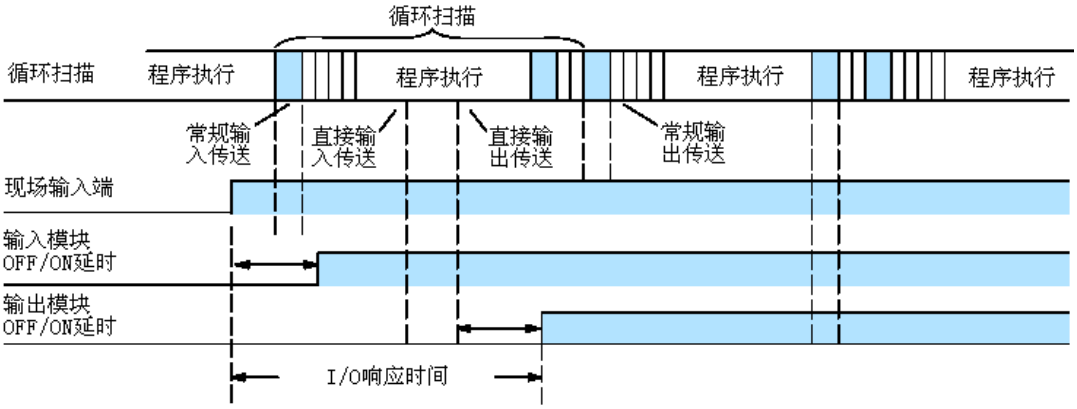


提高响应时间

以下方法可以帮您提高传送效率。

- 选择执行时间短的指令
- 使用直接传送指令（可在梯形图程序执行是刷新 I/O 点的状态）
- 选择高速响应模块

使用直接传送指令是一种非常有用的技巧。下图演示了直接传送指令及其效果。



在这种情况下，您可以按以下式子计算出响应时间。

输入延时 + 指令执行时间 + 输出延时 = 响应时间

指令执行时间是直接输入指令、直接输出指令、以及各种其它指令的执行时间的总和。



注意：使用直接输入指令时，输入状态表还是要由输入成批传送来更新。但是在程序执行时，不是取输入状态表的内容，而是按当时的输入状态执行。因此，任何紧接着的常规指令都会用到输入映射表中的值。任何紧接着的直接传送指令都会把其状态更新。

CPU 扫描周期

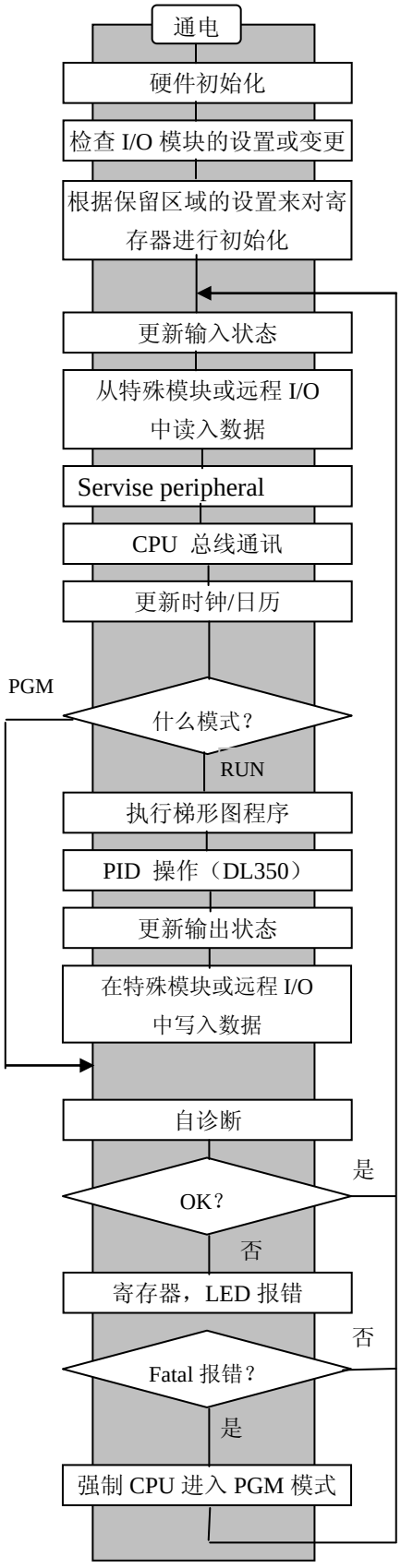
扫描周期覆盖了所有的操作系统所执行的任务。从 PROGRAM 模式转变为 RUN 模式之后，您可以通过 DirectSOFT 或手持编程器来查看最小、最大、及当前的扫描周期。这些信息对评估系统的性能来说是很重要的。

如前面所示，一个扫描周期由若干段组成。每一段完成其功能都需要一定的时间。在所有这些段中，只有程序执行的时间是受您控制的。因为不同的指令执行时间也是不同的。所以如果想获得更短的扫描周期，您可以使用高速指令。

您对 I/O 模块的选择和系统的设置，比如是扩展 I/O 还是远程 I/O，也会影响到扫描周期的长短。但是一般来说，这些都是由实际应用所决定的。

例如，您需要用到高速计数功能，那么您就必须选用一个高速计数模块。同样，如果有些 I/O 点在 CPU 以外一百多米的地方，您就需要用到远程 I/O。因为安装一个远程 I/O 比起延长每个 I/O 点的信号传输线来，安装更方便，成本也更低。

接下来的段落将介绍其中一些段工作所需的时间及相关内容。



初始化过程

在扫描期间，任何时候都由可能发生通讯请求，但 CPU 只在扫描到外围检测这一段时才“记载”相关请求。如果没有外围的联接，CPU 将不会在此上面花时间。

服务请求时间	DL350
最小	1.2μs
最大	1.5 – μs

外围检测

在扫描期间，任何时候都由可能发生通讯请求，但 CPU 只在扫描到外围检测这一段时才“记载”相关请求。如果没有外围的联接，CPU 将不会在此上面花时间。

记载请求时间（任何时候）		DL305
无任何联接	最小、最大	0μs
端口 1	传送 最小/最大	6.8/12.6μs
	接收 最小/最大	9.2/972μs
端口 2	传送 最小/最大	6.8/12.6μs
	接收 最小/最大	9.2/972μs

CPU 总线通讯

一些特殊模块可以通过 CPU 总线直接和 CPU 通讯。在循环的这一部分，CPU 完成所有的 CPU 通讯。所需时间取决于何种模块及何种请求模式。

注意：有些特殊模块对 CPU 的扫描时间有较大影响。如果您的应用对扫描周期要求比较高，请参考该模块附带的相关文档。

更新时钟/日历，特殊继电器，特殊寄存器

在这一阶段，时钟、日历及特殊继电器的值都进行更新并存入相应的 R 寄存器中。此更新只在 RUN 或 PROGRAM 模式下进行。

模式		DL350
PROGRAM 模式	最小	79.0μs
	最大	79.0μs
RUN 模式	最小	79.0μs
	最大	79.0μs

自诊断

DL305 CPU 可以进行多种系统自诊断。其所需时间取决于很多方面，比如安装的 I/O 模块数等。以下表格列出了一般期望的最小及最大时间。

自诊断时间	DL350
最小	104.0μs
最大	139.6μs

执行程序

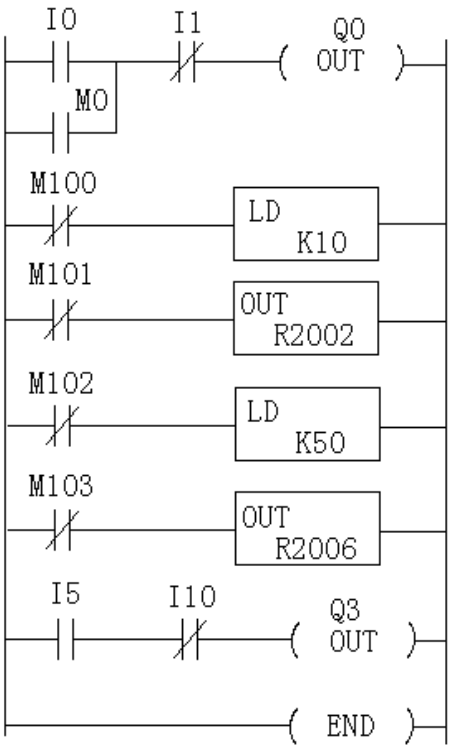
CPU 执行程序是从最顶部的指令一直到 END 指令，并按照从左向右、从上到下的顺序。当每个回路都进行计算时，相应的映射寄存器或存储器位置就更新。

处理程序所用的时间取决于您使用的指令的类型和数量以及总的程序执行时间。

您可以把所有指令执行时间相加来计算出总的程序执行时间。

例如，右图所示的程序可以用下面的方法来计算其程序执行时间。

指令	执行时间
STR I0	1.4 µs
OR M0	1.0 µs
ANDN I1	1.2 µs
OUT Q0	7.95 µs
STRN M100	1.6 µs
LD K10	62 µs
STRN M101	1.6 µs
OUT R2002	21.0 µs
STRN M102	1.6 µs
LD K50	62 µs
STRN M103	1.6 µs
OUT R2006	21.0 µs
STR I5	1.4 µs
ANDN I10	1.2 µs
OUT Q3	7.95 µs
END	16 µs
总计	210.5 µs

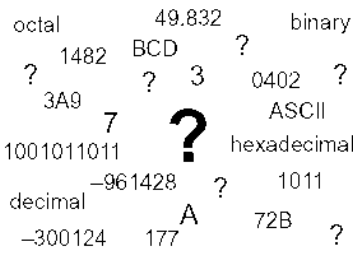


附录 C 提供了 DL350 CPU 所有指令的执行时间。

程序控制指令：DL350 CPU 还提供了一些可以改变程序执行方式的指令。包括 FOR/NEXT 循环，子程序以及中断子程序。这些指令可以打断正常程序的执行并能影响到程序的执行时间。第五章提供了有关这些不同类型指令是如何工作的详细信息。

PLC 数字系统

如果您是刚接触 PLC 的应用或首次使用我们的产品，请您先了解一下我们的产品所使用的数字。您会发现我们的 PLC 有其特有的使用数字的习惯。因此请您先花几分钟来对此熟悉一下。您在这里所了解到的信息可以应用于我们所有的 PLC 产品！



PLC 的存储和操作的数字都是二进制的形式：1 和 0。但为何我们要用不同形式的数字呢？不同的数字有不同的含义，对某个特定的目的，有些表示法用起来更加方便。有时我们用数字来表示一个尺寸大小或者一些东西的数量。另外有些数字指的是存储位置，地址或时间。在科学领域？，我们把工程单位和数字结合起来来表示某个特定的含义。

PLC 资源

PLC 提供了固定数量的资源，这取决于其型号和相应的配置。“资源”一词包含了变量存储器（R 寄存器），I/O 点数，定时器，计数器等。大多数模块式 PLC 允许你以 8 点为单位增加 I/O 点数。实际上，我们的 PLC 上的所有资源都是八进制来表示的。因为对计算机来说，使用八进制数比使用十进制更加方便，而且 8 是 2 的幂数。

使用八进制意味着每次计数以 8 个为一组。如右图所示有 8 个圆圈。其数量用十进制来表示就是“8”，但用八进制来表示却是“10”（8，9 在八进制中是没有的）。在八进制中，“10”意味着逢八进一。

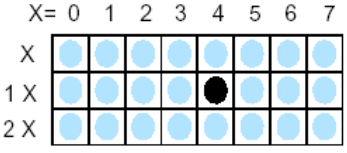


如下图所示，有两组 8 个圆圈。八进制数表示为“20”，可理解为两组 8 再加上 0。读作“two-zero octal”，而不是“twenty”。



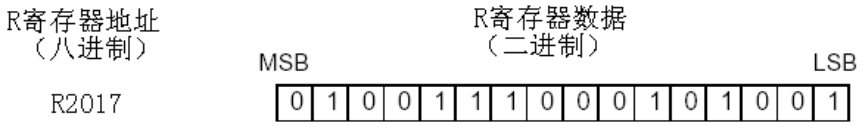
数 过 PLC 的资源以后，我们就要 *访问* PLC 的资源了（其中略有不同）。CPU 的指令集使用八进制地址来访问 PLC 资源。八进制地址和八进制的数量是类似的，只是它们从 0 开始。因为 0 对计算机来说是有一定的意义的，我们不能把它略去。

如右图所示，许多圆圈被排在方型区域内。要访问某个资源，PLC 指令会使用八进制数来指定其位置。如果这些都表示计数器，那么“C14”就是黑圆圈的位置。



R 寄存器

变量存储器（也称作 R 寄存器）用来存储梯形图程序及用户设置。R 寄存器位置和 R 寄存器地址是同一个意思，都是用八进制来表示的。例如，R2073 就是一个合法的地址，而 R1983 就是不合法的（8，9 在八进制中是没有的）。每个 R 寄存器能放一个“字”（word），即 16 位（bit）。设置寄存器时，我们的手册会提供每一个位的值。最低位（LSB）在右端，最高位（MSB）在左端。

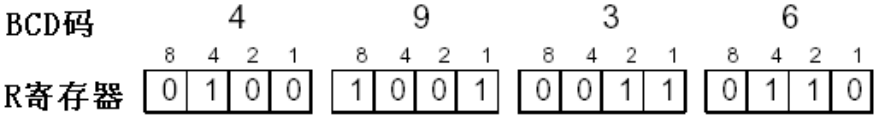


R 寄存器中的数据是 16 位二进制，但其数据寄存器一般不是一次设置一位的。指令或查看工具会使用到二进制，八进制，十进制和十六进制的数字。所有这些数字都被转化成二进制来让我们使用。

一个常见的问题是，“我们如何来判断一个数字是二进制，八进制，十进制或十六进制”？答案是我们一般不能从数字本身区分出来，但这并不要紧。重要的是：要写入寄存器的数据或机构和以后要从中读取数据的机构必须使用同样的数据类型（比如，八进制，十六进制，二进制等）。R 寄存器只是一个存储箱，它自身并不能转换或移动数据。

BCD（Binary-Coded Decimal）数字

因为通常人们习惯用十进制来计数，我们更愿意以十进制来操作或查看 PLC 的数据（通过操作界面）。但是，计算机使用单纯的二进制数的话，其工作效率会更高。有个折中的方案就是采用 BCD（Binary-Coded Decimal）码。一位 BCD 码可以表示 0 到 9，并占据 4 位二进制（a nibble——半位元组）。这样，每个 R 寄存器就能存储 4 个 BCD 数据，即十进制的 0000 到 9999。



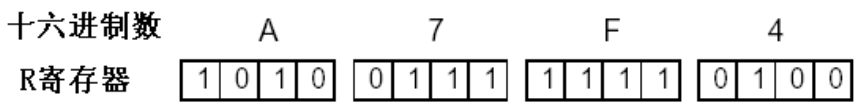
寄存器中如果单纯是二进制，一个字（16 位）能表示 0 到 65535 的数字。如果用来存储 BCD 码，那它只能表示 0 到 9999。许多算术指令都是用的 BCD 码，所以通过 DirectSOFT 或手持编程器我们可以以 BCD 码的形式来操作或查看数据。有些特殊的指令可以把 BCD 码转换成二进制数，或 vise-versa。

十六进制数

十六进制数和 BCD 码有些相似，只是前者能充分利用一定的存储空间。既然是十六进制，就需要 16 个不同的数字。因此，除了十进制中的 0 到 9，我们还使用 A 到 F（如下所示）。

十进制　0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
十六进制 0 1 2 3 4 5 6 7 8 9 A B C D E F

在一个 R 寄存器空间里，一个四位十六进制数可以表示 0 到 65535 的数。即从 0000 到 FFFF（HEX）。PLC 通常利用这个大范围来存放传感器的数据。

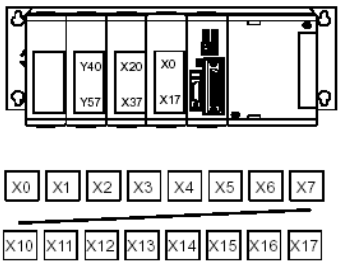


存储器映射

对任何的 PLC，您通常可以处理各种不同类型的信息。这包括输入装置的状态，输出装置的状态，各种定时元件，计数器等。例如，您需要直到系统是怎样来识别输入点、输出点及其它数据的。以下的章节将讨论 DL350 CPU 中的各种存储类型。存储器介绍的后面有一张存储器映射表。

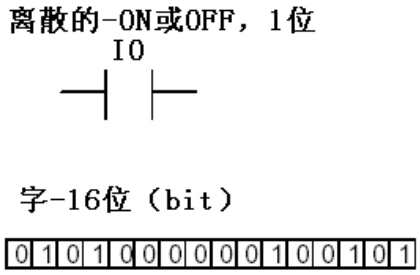
八进制编号系统

所有的存储器都是以八进制数来编号的。例如，右图表示了在离散输入点中所用到的八进制编号系统。请注意，八进制编号系统中绝对不含有 8 或 9。



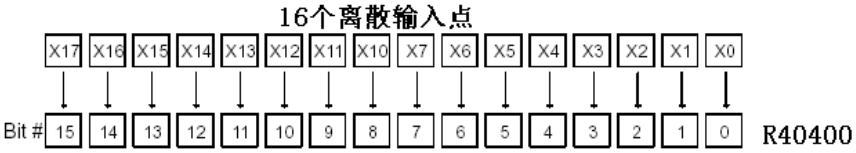
离散空间和字空间

当您对存储器的类型进行仔细观察时，您会发现 DL350 有两种存储器，即离散存储器和字存储器。离散存储器只占一位，存放 0 或 1。字存储器同 R 寄存器，是 16 位的，通常用来操作数据/数字或存储数据/数字等。有些信息会自动地存入 R 寄存器。例如，当前的时间值就是存在 R 寄存器中的。



离散存储器对应的 R 寄存器

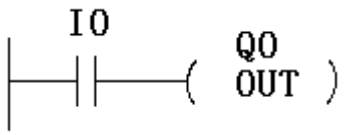
离散存储区域对应输入点、输出点、控制继电器、特殊继电器、级、定时器状态位、计数器状态位等。尽管如此，您还是可以以 R 寄存器的字来访问离散型数据。每个 R 寄存器空间包含 16 个离散空间。例如，下图表示了 X 输入点是如何与 R 寄存器相对应的。



这些离散的存储区域和其相应的 R 寄存器表在 DL350 CPU 一章中列出。

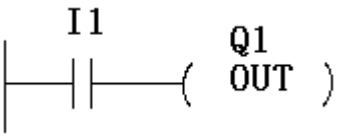
输入点（I）

I 表示离散输入点。DL350 CPU 最多支持 512 个输入点。例如，当 I0 接通，Q0 就接通。



输出点（Q）

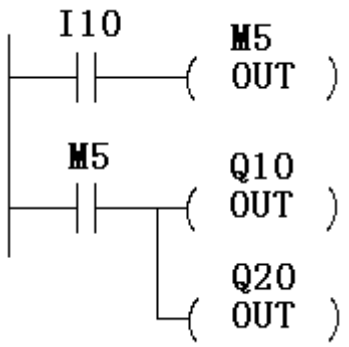
Q 表示离散输出点。。DL350 CPU 最多支持 512 个输出点。在这个例子中，当 I1 接通时，Q1 接通。



中间继电器（M）

中间继电器也是离散的，通常用来控制用户程序。中间继电器不是一个现实的“继电器”，所以它不能接任何的开关、输出线圈等。它们都是内藏于 CPU 之中的。中间继电器可以看作离散输入点或离散输出点来使用。它们的位置将会在程序中使用到，以位或字的形式。

在这个例子中，当 I10 为 ON 时，存储位置中的 M5 也就 ON。下面的第二个回路说明了如何把中间继电器当作输入点来使用。

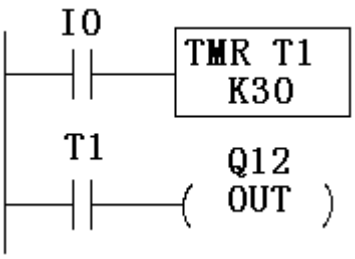


定时器

定时器状态位（T）

定时器的数量取决于您的 CPU 的型号。本节的最后提供了 DL350 CPU 的定时器数量。即使您不知道定时器的号码，您也可以访问定时器的状态位。它能反映某个定时器的设定值和经过值之间的关系。当经过值大于等于设定值，那么定时器状态位就变为 ON。

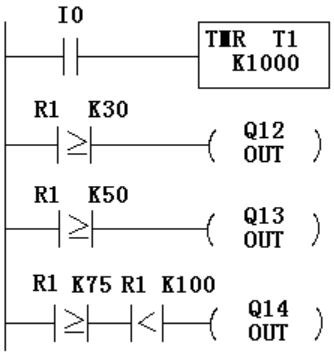
当输入 I0 接通，定时器 T1 就开始计时。当定时器达到预设的 3 秒时，定时器的状态就让 T1 接通。当 T1 接通，Q12 就接通。



定时器经过值

正如前面所讲的，有些信息会自动地存放到 R 寄存器中。定时器的经过值就是这样的。例如，R0 中存放 T0 的经过值，R1 中存放 T1 的经过值等。

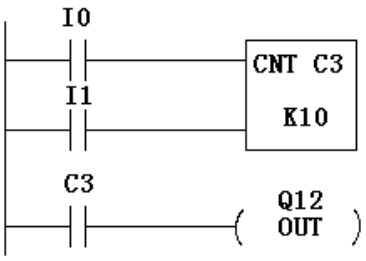
这样做是为了使编程更加方便灵活。右图说明了如何通过比较接点来监控一个定时器的多个经过值。



计数器

计数器状态位

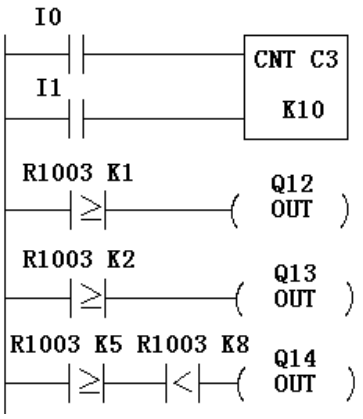
计数器的数量取决于您的 CPU 的型号。本节的最后提供了 DL350 CPU 支持的计数器的数量。即使您不知道计数器的号码，您也可以访问计数器的状态位。它能反映某个计数器的设定值和经过值之间的关系。当经过值大于等于设定值，那么计数器状态位就变为 ON。如右图，每当 I0 从 OFF 变为 ON 时，计数器的经过值就加 1。当 I1 接通时，计数器就被复位，经过值清零。当计数器的经过值达到了预设的 10 时，C3 就接通。当 C3 接通时，Q12 就接通。



计数器经过值

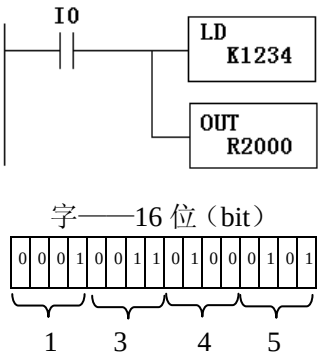
和定时器一样，计数器的经过值也是自动存放在 R 寄存器中的。例如，R1000 中存放 C0 的经过值，R1001 中存放 C1 的经过值等。

这样做是为了使编程更加方便灵活。右图说明了如何通过比较接点来监控一个计数器的多个经过值。



字存储器
（R 型）

字存储器即 R 寄存器，它是 16 位的，通常用来对数据/数字进行操作或者存储等。有些信息是自动地存入 R 寄存器的。例如，定时器的经过值是存在 R 寄存器中的。右边的例子表示 4 个 BCD 数装入累加器后再存入 R 寄存器中。

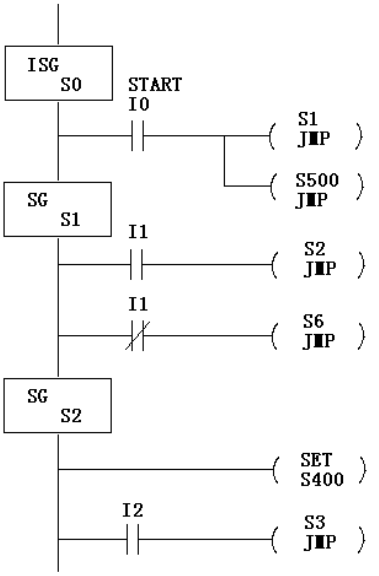


级
（S 型）

级用于 RLL PLUS 编程方式，类似于流程图。每个级都表示一个程序段。当该程序段或者说级为 ON，则该段中的程序执行。当级为 OFF，则该级不执行并且 CPU 跳到下一个为 ON 的级。有关 RLL PLUS 编程的详情请参阅第七章。

每个级都有一个状态位来表示它当前的状态。如果某个级为 ON，则和它对应的那个状态位也为 ON。反之，如果某个级为 OFF，则和它对应的那个状态位也为 OFF。这些状态位也可通过其他指令来强行设置，比如用 SET 或 RESET 指令。这样，您就可以在整个程序中方便地控制各个级。

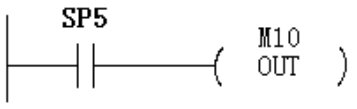
梯形表示法



特殊继电器
（SP 型）

特殊继电器是系统内按用途定义的离散的存储空间，它有许多不同类型。例如，有些是辅助编程的，有些是提供系统的状态信息的。附录 D 中有一个特殊继电器的完整列表。

在右边的例子中，中间继电器 M10 会不断 ON 50 ms 后接着 OFF 50 ms。因为 SP5 的功能就是产生周期为 100ms 的脉冲。



SP4：1 秒时钟
SP5：100ms 时钟
SP6：50ms 时钟

DL350 系统 R 寄存器

系统 R 寄存器	内容	默认值/范围
R7620-R7627	DV-1000 操作界面参数区	
R7620	设定数值存放 R 寄存器	R0-R3777
R7621	设定消息存放 R 寄存器	R0-R3777
R7622	设定显示 R 寄存器的数量（1-16）	1-16
R7623	设定存放以上数量的 R 寄存器	R0-R3777
R7624	设定显示字符存放的 R 寄存器	R0-R3777
R7625	存放功能号	I、O 或 M 对应的 R 寄存器
R7626	保留	0，1，2，3，12
R7627	保留	默认值=0000
R7630-R7632	保留	-
R7633	用户定义定时器中断/操作指令的标志位 Bit 0-7 40H 设置中断 Bit 12 电池标志位。ON 为使用，OFF 为不使用。 Bit 15 二进制指令标志位。ON 为使用标志位，OFF 为不使用。	
R7634	用户定义定时器中断	
R7640	回路表起始地址	R1400-R7340
R7641	允许的回路数量	1-4
R7642	错误代码-回路表出错的 R 寄存器位置	
R7643-R7647	保留	
R7650	端口 2 结束码设置（A55A），无协议通讯开始。	
R7651	端口 2 数据格式-无协议通讯格式设定	
R7652	端口 2 格式类型设定-无协议通讯码设定	
R7653	端口 2 终止码设定-无协议通讯结束码设定	
R7654	端口 2 存储 R 寄存器地址-无协议通讯数据存放的 R 寄存器地址	
R7655	端口 2 激活区域 0-7 通讯协议（flag 0），8-15 通讯时间/延时时间（flag 1）。	
R7656	端口 2 激活区域-0-15 通讯（flag 2，flag3）	
R7657	端口 2 激活区域-参数选择	
R7660-R7707	激活信息	
R7710-R7717	保留	
R7720-R7722	DV-1000 操作界面参数位置	
R7720	？定时器预设值指针	
R7721	？计数器预设值指针	
R7722	？定时器预设值大小	
R7730-R7737	对应 0-7 槽 D3-DCM	

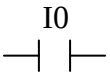
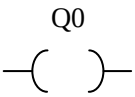
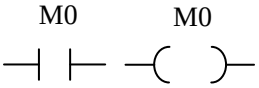
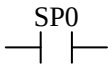
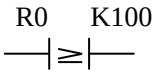
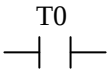
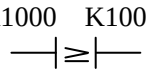
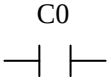
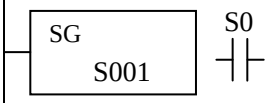
R7747	含有 10ms 计数器位置。该位置每 10ms 自动加 1。	

系统 R 寄存器	内容
R7750	保留
R7751	错误信息代码-执行 FAULT 指令时可向其中写入 4 为错误代码
R7752	保留
R7753	保留
R7754	保留
R7755	错误代码-写入 FATAL 报错码
R7756	错误代码-写入 MAJOR 报错码
R7757	错误代码-写入 MINOR 报错码
R7760-R7762	保留
R7763-R7764	语法错误码位置
R7765	
R7766	含有时钟“秒位”数字。(0-59)
R7767	含有时钟“分位”数字。(0-59)
R7770	含有时钟“小时”数字。(0-24)
R7771	含有“星期几”信息（Mon,Tue,等）
R7772	含有“日期”信息（1，2，3，4，……30 等）
R7773	含有“月份”信息（1-12）
R7774	含有“年份”信息（00-99）
R7775	扫描-写入当前扫描时间（毫秒）
R7776	扫描-从上一次 Program 打到 Run 后，程序执行的最小周期（毫秒）。
R7777	扫描-从上一次 Program 打到 Run 后，程序执行的最大周期（毫秒）。

以下中间继电器只能在 D3-350 CPU 的远程 I/O 的端口 2 上激活。

中间继电器	内容
M740	完成激活-远程 I/O 激活表写入完成以后，梯形图逻辑会把该继电器 ON
M741	删除已接收数据-把该继电器置 ON 可以删除通讯出错时接收的数据。
M743	重启-把该继电器置 ON 可以使因通讯出错而停止后重新开始。
M750-M757	激活出错-当激活表出错时，通讯继电器就 ON。(M750=MASTER，M751=SLAVE 1…M757=SLAVE 7)
M760-M767	准备通讯-当激活表中的数据有效时，通讯继电器就 ON。(M750=MASTER，M761=SLAVE 1…M767=SLAVE 7)

DL350 功能存储器编号表

存储器类型	定义号（八进制）	字存储器（八进制）	数量（十进制）	符号
输入点	I0~I777	R40400~R40437	512	
输出点	Q0~Q777	R40500~R40537	512	
中间继电器	M0~M1777	R40600~R40677	1024	
特殊继电器	SP0~SP777	R41200~R41237	512	
定时器经过值	无	R0~R377	256	
定时器状态位	T0~T377	R41100~R41117	256	
计数器经过值	无	R1000~R1177	128	
计数器状态位	C0~C177	R41140~R41147	128	
数据字	无	R1400~R7377 R10000~R17777	3076 4096	未定，用于多种指令
级	S0~S1777	R41000~S41077	1024	
系统参数	无	R7400~R7777	256	系统定义，用于多种目的

I/O 存储器定义号表

以下表格列出了与各 I/O 点对应的 R 寄存器中的位。

高位								DL350 输入点及输出点							低位	I 输入点	Q 输出点
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	地址	地址
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	R40400	R40500
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	R40401	R40501
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	R40402	R40502
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	R40403	R40503
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	R40404	R40504
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	R40405	R40505
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	R40406	R40506
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	R40407	R40507
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	R40410	R40510
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	R40411	R40511
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	R40412	R40512
277	276	275	274	273	272	271	270	267	276	265	264	263	262	261	260	R40413	R40513
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	R40414	R40514
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	R40415	R40515
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	R40416	R40516
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	R40417	R40517
417	416	415	414	413	412	411	410	407	306	305	304	303	302	301	300	R40420	R40520
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	R40421	R40521
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	R40422	R40522
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	R40423	R40523
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	R40424	R40524
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	R40425	R40525
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	R40426	R40526
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	R40427	R40527
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	R40430	R40530
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	R40431	R40531
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	R40432	R40532
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	R40433	R40533
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	R40434	R40534
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	R40435	R40535
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	R40436	R40536
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	R40437	R40537

中间继电器 M 定义号表

以下表格列出了与各中间继电器对应的 R 寄存器中的位。

高位								DL350 中间继电器（M）							低位	对应地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	R40600
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	R40601
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	R40602
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	R40603
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	R40604
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	R40605
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	R40606
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	R40607
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	R40610
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	R40611
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	R40612
277	276	275	274	273	272	271	270	267	276	265	264	263	262	261	260	R40613
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	R40614
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	R40615
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	R40616
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	R40617
417	416	415	414	413	412	411	410	407	306	305	304	303	302	301	300	R40620
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	R40621
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	R40622
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	R40623
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	R40624
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	R40625
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	R40626
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	R40627
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	R40630
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	R40631
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	R40632
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	R40633
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	R40634
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	R40635
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	R40636
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	R40637

高位								DL350 中间继电器（M）							低位	对应地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	R40640
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	R40641
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	R40642
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	R40643
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	R40644
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	R40645
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	R40646
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	R40647
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	R40650
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	R40651
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	R40652
1277	1276	1275	1274	1273	1272	1271	1270	1267	1276	1265	1264	1263	1262	1261	1260	R40653
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	R40654
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	R40655
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	R40656
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	R40657
1417	1416	1415	1414	1413	1412	1411	1410	1407	1306	1305	1304	1303	1302	1301	1300	R40660
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	R40661
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	R40662
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	R40663
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	R40664
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	R40665
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	R40666
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	R40667
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	R40670
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	R40671
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	R40672
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	R40673
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	R40674
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	R40675
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	R40676
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	R40677

级控制/状态定义号表

以下表格列出了每个级对应的控制位及其所在的寄存器。

高位								DL350 级（S）控制位							低位	对应地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	R41000
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	R41001
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	R41002
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	R41003
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	R41004
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	R41005
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	R41006
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	R41007
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	R41010
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	R41011
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	R41012
277	276	275	274	273	272	271	270	267	276	265	264	263	262	261	260	R41013
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	R41014
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	R41015
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	R41016
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	R41017
417	416	415	414	413	412	411	410	407	306	305	304	303	302	301	300	R41020
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	R41021
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	R41022
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	R41023
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	R41024
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	R41025
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	R41026
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	R41027
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	R41030
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	R41031
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	R41032
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	R41033
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	R41034
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	R41035
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	R41036
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	R41037

高位					DL350 附加级（S） 控制位（续）											低位	对应地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0		
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	R41040
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	R41041
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	R41042
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	R41043
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	R41044
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	R41045
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	R41046
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	R41047
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	R41050
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	R41051
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	R41052
1277	1276	1275	1274	1273	1272	1271	1270	1267	1276	1265	1264	1263	1262	1261	1260	R41053
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	R41054
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	R41055
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	R41056
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	R41057
1417	1416	1415	1414	1413	1412	1411	1410	1407	1306	1305	1304	1303	1302	1301	1300	R41060
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	R41061
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	R41062
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	R41063
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	R41064
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	R41065
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	R41066
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	R41067
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	R41070
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	R41071
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	R41072
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	R41073
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	R41074
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	R41075
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	R41076
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	R41077

定时器和计数器状态位表

以下表格列出了每个定时器和计数器的接点所对应的位及其相应的 R 寄存器地址。

高位 DL350 定时器（T）和计数器（C）接点 低位																定时器地址	计数器地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	R41100	R41140
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	R41101	R41141
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	R41102	R41142
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	R41103	R41143
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	R41104	R41144
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	R41105	R41145
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	R41106	R41146
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	R41107	R41147

下表列出了 DL350 的附加定时器接点所对应的位及其相应的 R 寄存器地址。

高位 DL350 附加定时器（T）接点 低位																定时器地址
17	16	15	14	13	12	11	10	7	6	5	4	3	2	1	0	
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	R41110
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	R41111
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	R41112
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	R41113
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	R41114
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	R41115
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	R41116
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	R41117

第四章　系统设计和设置

4

本章将介绍：

——DL305 系统的设计方法

——模块的安放

——功率预算的计算

——扩展 I/O

——远程 I/O

——与 MODBUS 和 ***Direct***NET 的网络连接

——主局的操作

——子局的操作

DL305 系统的设计方法

I/O 系统设置

DL350 提供了以下这几种类型的 I/O

- **本地 I/O**——由和 CPU 装在同一个框架中的 I/O 模块组成
- **远程 I/O**——由另一个框架上的 I/O 模块组成，该框架通过通过本地框架上的下面的端口和本地框架相连。
- **扩展 I/O**——由扩展框架中的 I/O 模块组成

DL305 系统可以对其配置采用多种不同的安排方式。所有的 I/O 配置都使用 DL305 的标准 I/O 模块和框架。

联网设置

DL305 CPU 可以通过以下途径来给系统进行联网：

- **DL350 通讯端口**——DL350 CPU 在 Port 2 上含一个 25 脚的连接器，它含有一个内置的 RTU MODBUS 连接。
- **MODBUS 主局模块**——它可以安装在任何一个插槽中来作为 MODBUS 网络的主局。
- **MODBUS 子局模块**——它可以安装在任何一个插槽中来作为 MODBUS 网络的子局。

模块	主局	子局
DL350 CPU	***Direct***NET MODBUS RTU	***Direct***NET K 协议 MODBUS RTU

基本框架设置

DL305 系统目前带有两种类型的基本框架。它们都有 5 槽，8 槽，10 槽三种类型。所有的 DL305 CPU 都能在这些基本框架上正常工作。型号为 XXXXX－1 的框架是为提高 DL350 CPU 的 特性来设计的，但其他的 DL305 CPU 也能用在这些框架上。您也可以在系统中同时采用各种不同的框架。在混用框架或随意安装 DL350 CPU 模块的情况下，您可能会失去 DL350 CPU 的一些功能。DL350 会回到 8 位寻址的状态，而且功能也会降至和 DL340 差不多。本节将着重介绍怎样在型号为 XXXXX－1 的框架上使用 DL350 CPU。如果您要在常规的框架上使用 DL350 CPU 或您在同一个系统中使用不同的框架，请参考附录 F 的有关信息。

型号为 XXXXX－1 的框架支持 8 位串行总线，当使用 DL350 作为 CPU 时，它可以使用智能模块。其寻址方式被简化，同时可以使用的 I/O 点数也增大了。由于电流容量超过 2.0A，所以您必须重新进行电流预算。

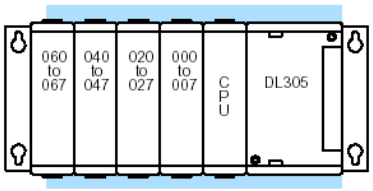
模块的放置

插槽的编号

DL305 框架按插槽数量的不同，分为 5 槽、8 槽等。其中有一个必须用来安装 CPU 模块，因此能使用的 I/O 插槽要比实际的少一个。例如，在 5 槽框架中您只能使用到 4 个 I/O 插槽。I/O 插槽的编号是 0－3。CPU 插槽中只能插 CPU 模块，因此不给该插槽编号。

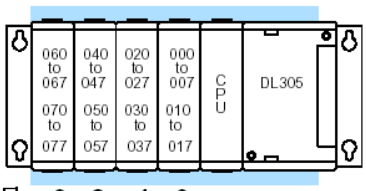
下面的两个例子说明了在 5 槽框架上安装 8 点 I/O 模块和在 5 槽框架上安装 16 点 I/O 模块的例子。

5 槽框架使用 8 点模块



槽号：3—2—1—0

5 槽框架使用 16 点模块



槽号：3—2—1—0

I/O 模块安装规则

您的某个模块应该安装在哪个槽中，这是有一定的限制的。有些模块有特定的安装位置，因此可能影响到其他模块的安装。4－6 和 4－7 上的两张表格列出了一些安装规则。

有时如果有必要的话，CPU 后面的几个槽位可以用作扩展插槽。例如，假使规则说明某个模块必须安装在 CPU 模块旁边，而且系统是由两个 5 槽框架组成的话，则只有扩展框架的 1 号和 2 号槽是有效的插槽。

以下表格提供了 DL305 组件的一般安装规则。

模块	限制
CPU	CPU 必须插在本地框架的第一个槽中。第一个槽位即与电源相邻的那个槽位。
16 点 I/0 模块	任意
模拟量模块	任意
ASCII Basic 模块	任意
高速计数模块	D3-350 CPU 不支持高速计数模块

I/0 设置

I/O 地址采用 8 进制数编号，从靠近 CPU 模块的那个模块开始编号。不管插槽中是否有模块，其地址是以 16 为一组进行分配的。输入模块和输出模块可以按任意次序安装，但对一些特殊模块的安装位置有一定的限制。

功率预算

管理您的电能资源

当您已经为您的 DL305 系统选好了 I/0 模块的类型和数量时，您还要记住很重要的一点是电源的负载能力是有限的。我们向您提供了一张图表来帮助您了解各框架的负载能力。这张图表在您进行 I/O 点选择时，能帮助您方便地计算电源所需要的功率。本节的末尾有一个电流预算的例子。



警告：进行电流预算是极其重要的。如果您在使用中超过了电流预算，系统有可能会发生不可预测的情况并由此产生对人员的伤害和对设备的损坏。

框架电源规格

下表说明了新的 XXXXX－1 型上各种电压输出所对应的最大电流。

框架	5V 电源输出电流 (A)	5V 电源输出电流 (A)	5V 电源输出电流 (A)	辅助 24 VDC 输出框架端子
D3-05B-1	1.0A(50℃) 0.7A(60℃)	2.0	0.6	100
D3-05BDC-1	1.0A(50℃) 0.7A(60℃)	2.0	0.6	无
D3-08-1	1.0A(50℃) 0.7A(60℃)	2.0	0.6	100
D3-08BDC-1	1.0A(50℃) 0.7A(60℃)	2.0	0.6	无
D3-10B-1	1.0A(50℃) 0.7A(60℃)	2.0	0.6	100
D3-10BDC	1.0A(50℃) 0.7A(60℃)	2.0	0.6	无

每个模块所需的 I/O 点数

每中模块都有一定数量的 I/O 点，包括一些特殊模块比如模拟量模块、计数模块等也是一样。下面的那张表格提供了每个模块对应的 I/O 点数。

模块的功率要求

接下来的三页将说明每个模块所需的最大电流。其中的需要外部电源是对应模块操作而不是现场接线。如果外部设备需要 24VDC，可以直接使用框架上的 24VDC 输出（最大 100mA），只要不超过它的容量即可。

	占用 I/O 点数	5V 电源需要（mA）	9V 电源需要（mA）	24V 电源需要（mA）	外部电源需要
CPU					
D3-350		500	20	0	无
直流输入模块					
D3-08ND2	8	0	10	112	无
D3-16ND2-1	16	0	25	224	无
D3-16ND2-2	16	0	24	209	无
D3-16ND2F	16	0	25	224	无
F3-16ND3F	16	0	148	68	无
交流输入模块					
D3-08NA-1	8	0	10	0	无
D3-08NA-2	16	0	130	0	无
D3-16NA	16	0	100	0	无
AC/DC 输入模块					
D3-08NE3	8	0	10	0	无
D3-16NE3	16	0	130	0	无
直流输出模块					
D3-08TD1	8	0	20	24	无
D3-08TD2	8	0	30	0	无
D3-16TD1-1	16	0	40	96	无
D3-16TD1-2	16	0	40	96	无
D3-16TD2	16	0	180	0	无
交流输出模块					
D3-04TAS	8	0	12	0	无
F3-08TAS	8	0	80	0	无
D3-08TA-1	8	0	96	0	无
D3-08TA-2	8	0	160	0	无
F3-16TA-1	16	0	160	0	无
D3-16TA-2	16	0	400	0	无

	占用 I/O 点数	5V 电源需要（mA）	9V 电源需要（mA）	24V 电源需要（mA）	所需外部电源
继电器输出模块					
D3-08TR	8	0	360	0	无
F3-08TRS-1	8	0	296	0	无
F3-08TRS-2	8	0	296	0	无
D3-16TR	16	0	480	0	无
模拟量					
D3-04AD	16	0	55	0	24VDC 最大 65mA
F3-04ADS	16	0	183	50	无
F3-08AD	16	0	25	37	无
F3-08TEMP	16	0	25	37	无
F3-08-THM-n	16	0	50	34	无
F3-16AD	16	0	33	47	无
D3-02DA	16	0	80	0	24VDC 最大 170mA
F3-04DA-1	16	0	144	108	无
F3-04DA-2	16	0	144	108	无
F3-04DAS	16	0	154	145	无
网络通讯模块					
FA-UNICON	0	0	0	0	24VDC 或 5VDC 最大 65mA
ASCII BASIC 模块					
F3-AB128-R	16	0	205	0	无
F3-AB128-T	16	0	205	0	无
F3-AB128	16	0	90	0	无
F3-AB64	16	0	90	0	无
特殊模块					
D3-08SIM	8	0	10	112	无
D3-HSC	16	0	70	0	无
编程器					
D2-HPP		200	50	0	可选

电流预算的例子　以下例子说明了如何来为 DL305 系统进行电流预算。

基本框架＃ 0	**模块类型**	**5VDC(mA)**	**9VDC(mA)**	**辅助电源 24VDC 输出（mA）**
允许的框架电流	**D3-05B**	**1000**	**2000**	**600**
CPU 插槽	D3-350	+500	+120	
0 号槽	D3-16NE3	+0	+130	+0
1 号槽	D3-16NE3	+0	+130	+0
2 号槽	F3-16TA-1	+0	+160	+0
3 号槽	F3-16TA-1	+0	+160	+0
4 号槽				
5 号槽				+0
6 号槽				+0
7 号槽				+0
其他				
手持编程器	D2-HPP	+200	+200	+0
总的电流		**700**	**900**	**0**
电流余量		1000-700=300	2000-900=1100	600-0=600

1．根据您的系统的组件来填表。首先，填入框架的电源数量。其次，列出对 CPU、I/O 模块，其他设备包括手持编程器等的要求。注意，尽管手持编程器或 DV-1000 并不安装在框架中，它们仍然会消耗系统的功率。同时，也要确保您有一些外部电源，比如模拟量模块所需的 24VDC。

2．依次把各槽的电流值，并在“**总的电流**”一行中填入适当的值。

3．把“**允许的框架电流**”中的值减去“**总的电流**”中的值，然后把结果填入“**电流余量**”一项中。

4．如果“**总的电流**”中的值比“**允许的框架电流**”中的值还要大，则说明超过了电流预算。使用该配置就有不安全因素，所以您应该重新进行 I/O 配置。



警告：进行电流预算是*极其*重要的。如果您在使用中超过了电流预算，系统有可能会发生不可预测的情况并由此产生对人员的伤害和对设备的损坏。

电流预算的工作表　以下空表格是提供给您进行电流预算的。

基本框架＃ 0	模块类型	5VDC(mA)	9VDC(mA)	辅助电源 24VDC 输出（mA）
允许的框架电流				
CPU 插槽				
0 号槽				
1 号槽				
2 号槽				
3 号槽				
4 号槽				
5 号槽				
6 号槽				
7 号槽				
其他				
手持编程器	D2-HPP			
总的电流				
电流余量				

1．根据您的系统的组件来填表。首先，填入框架的电源数量。其次，列出对 CPU、I/O 模块，其他设备包括手持编程器等的要求。注意，尽管手持编程器或 DV-1000 并不安装在框架中，它们仍然会消耗系统的功率。同时，也要确保您有一些外部电源，比如模拟量模块所需的 24VDC。

2．依次把各槽的电流值，并在“**总的电流**”一行中填入适当的值。

3．把“**允许的框架电流**”中的值减去“**总的电流**”中的值，然后把结果填入“**电流余量**”一项中。

4．如果“**总的电流**”中的值比“**允许的框架电流**”中的值还要大，则说明超过了电流预算。使用该配置就有不安全因素，所以您应该重新进行 I/O 配置。



警告：进行电流预算是*极其*重要的。如果您在使用中超过了电流预算，系统有可能会发生不可预测的情况并由此产生对人员的伤害和对设备的损坏。

本地 I/O 扩展

基本框架

下表可以帮助您了解怎样在您的控制系统中采用不同的 DL305 框架。

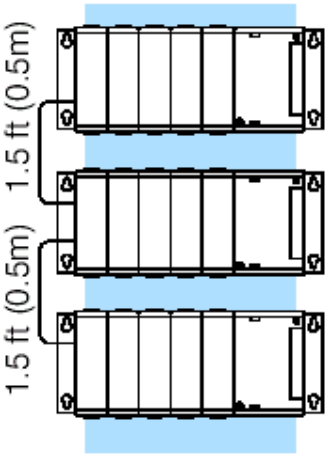
框架部分＃	槽数	能用作本地 CPU 框架	能用作远程 CPU 框架
D3-05B-1	5	能	能
D3-05BDC-1	5	能	能
D3-08B-1	8	能	能
D3-08BDC-1	8	能	能
D3-10B-1	10	能	能
D3-10BDC-1	10	能	能

本地/扩展连接

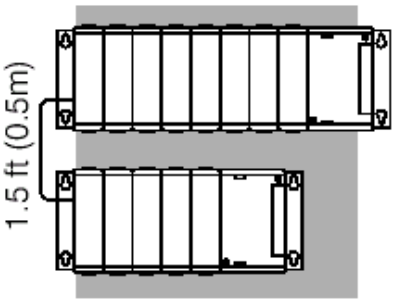
以下配置说明了 DL350 CPU 所支持的一些框架的连接方式。

注意：当您要扩展您的系统时，您可以使用以下方式中的一种。若您采用的连接方式不是以下其中的一种，则系统以可能不能正常工作。

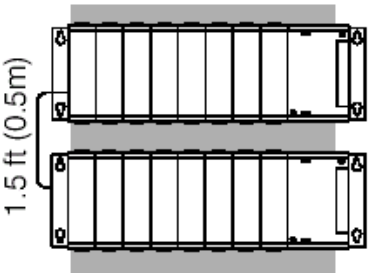
5 槽的本地 CPU 框架最多能接两个 5 槽的扩展框架



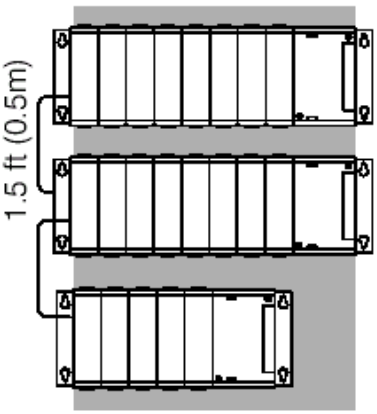
8 槽框架连接一个 5 槽扩展框架



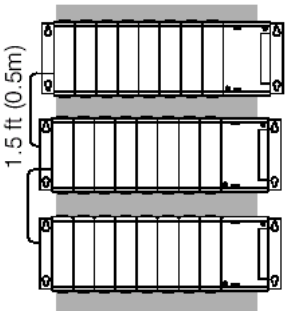
8 槽框架连接一个 8 槽框架



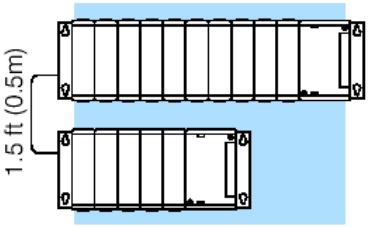
8 槽框架连接一个 8 槽和一个 5 槽的扩展框架。



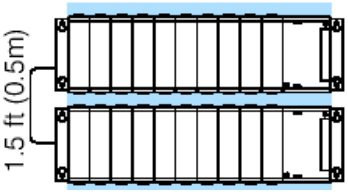
8 槽本地 CPU 框架连接两个 8 槽框架



10 槽框架连接一个 5 槽框架



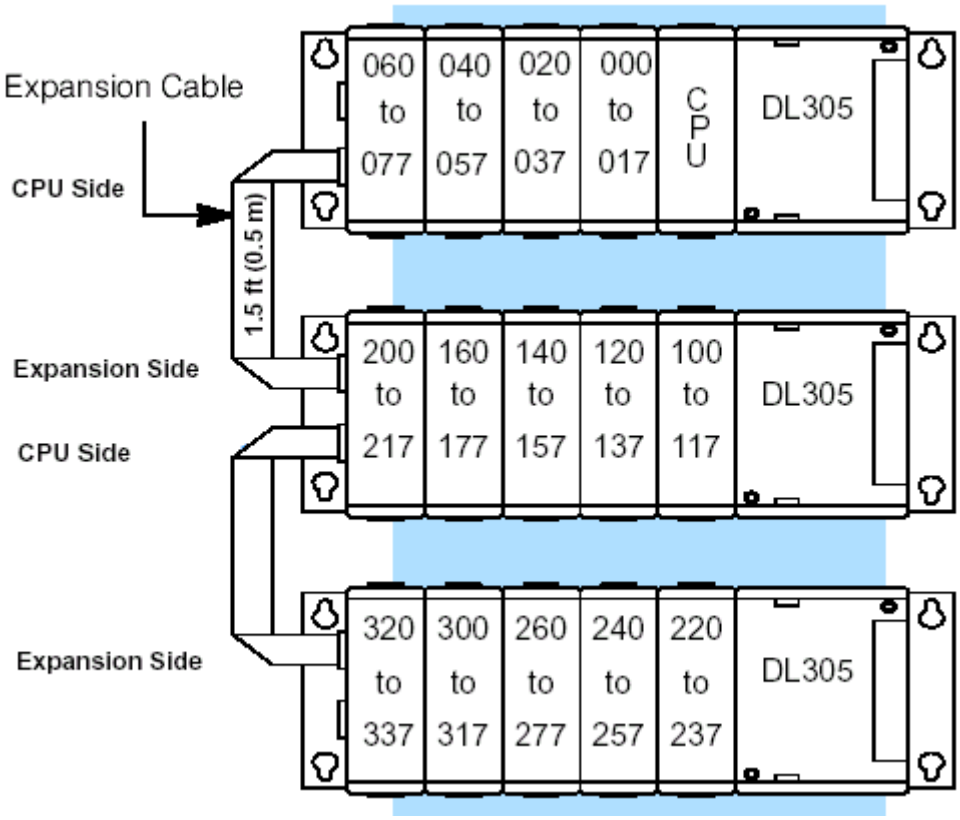
10 槽框架连接一个 10 槽框架



连接扩展框架

本地 CPU 框架通过一根 0.5M 的电缆（D3-EXCBL）与扩展框架相连接。

框架的上面一个连接器是前一个框架的输入端。而下面的一个连接器是是扩展框架的输出端。扩展电缆两端分别标有“CPU Side”和“Expansion Side”。其中“CPU Side”一端接在框架的下面一个连接器上，而“Expansion Side”一端接在下一个框架的上面一个连接器上。

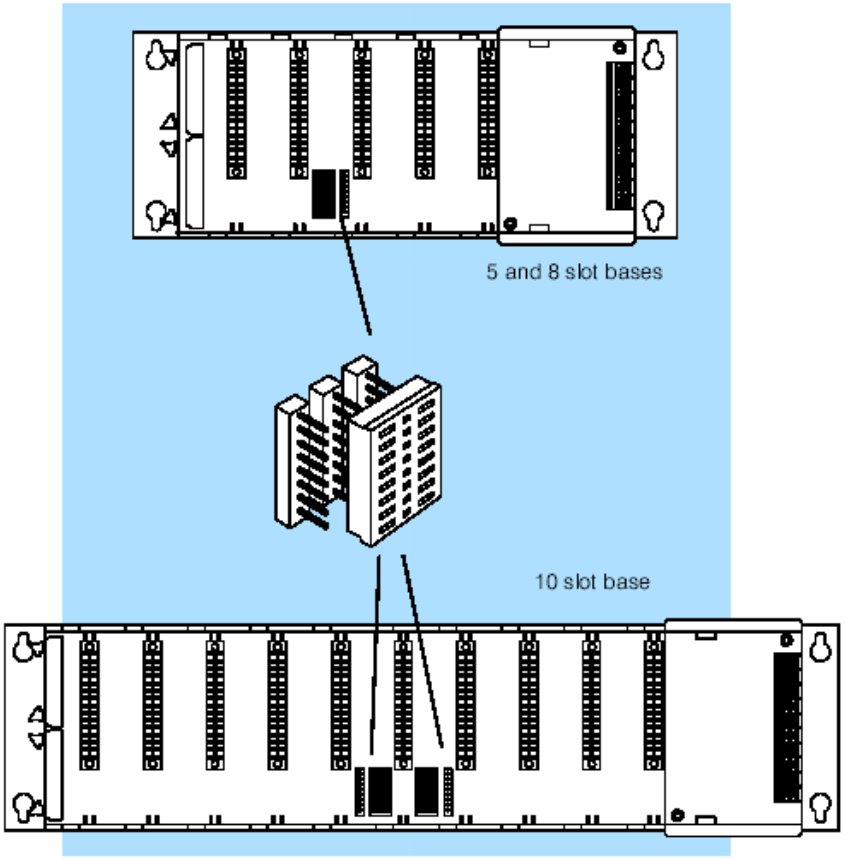


注意：不要把扩展电缆和其他普通 I/O 电缆放在一起。

设置框架的开关

跳线开关

对于 5 槽和 8 槽的框架，它们的 3 号槽和 4 号槽之间有一个跳线开关来把框架设置成本地 CPU 框架或扩展框架。而对于 10 槽框架，有两个跳线开关。一个在 4 号槽和 5 号槽之间，另一个在 5 号槽和 6 号槽之间。其中第二个开关用来设置使用 DL330/DL340 CPU 时的 I/O 寻址范围。当使用 DL350 CPU 时，必须把它放到右边的位置。



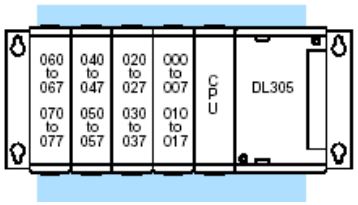
5 槽本地 CPU 框架的 I/O 设置

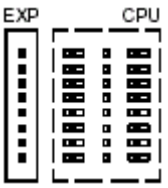
开关设置　　5 槽框架上的跳线开关可用来设置以下项目。

框架类型	跳线帽位置
本地 CPU	右
第一个扩展	左
最后一个扩展	右

5 槽框架

总共的 I/O 点数：
8 点模块－32
16 点模块－64



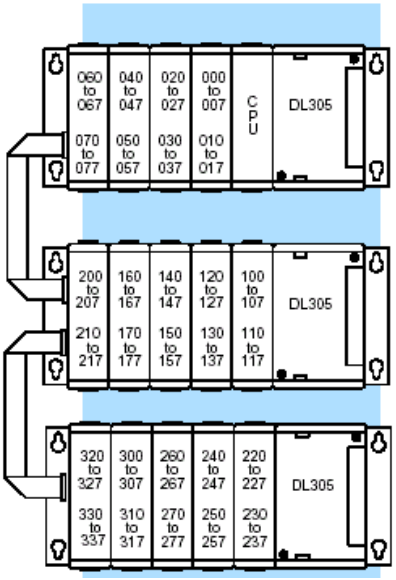


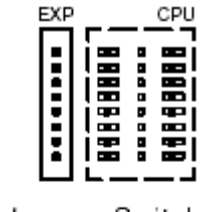
跳线位置

一个和两个 5 槽扩展框架

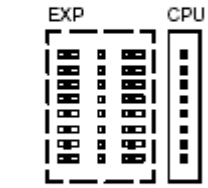
总共的 I/O 点数：
一个扩展框架
8 点模块－72
16 点模块－144

两个扩展框架
8 点模块－112
16 点模块－224

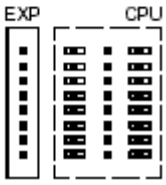




Jumper Switch



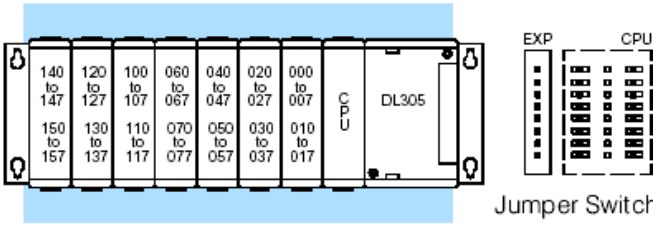
Jumper Switch



8 槽本地 CPU 框架的 I/O 设置

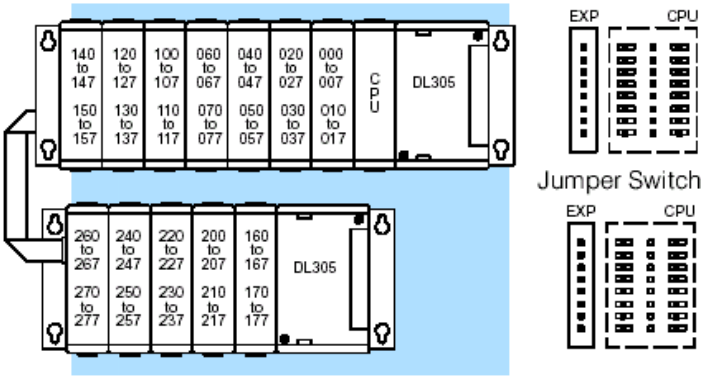
8 槽框架

总的 I/O 点数：
8 点模块－56
16 点模块－112



8 点框架和 5 槽扩展框架

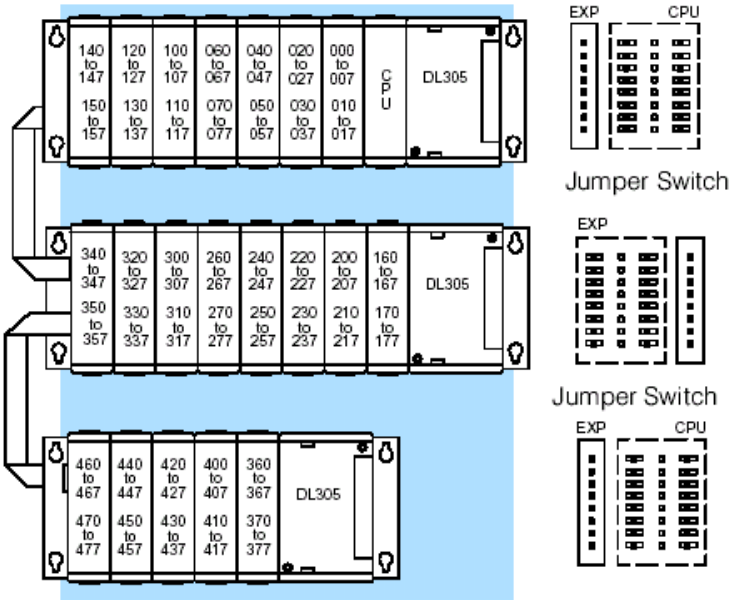
总的 I/O 点数：
8 点模块－96
16 点模块－192



8 槽框架和一个 8 槽框架、一个 5 槽框架

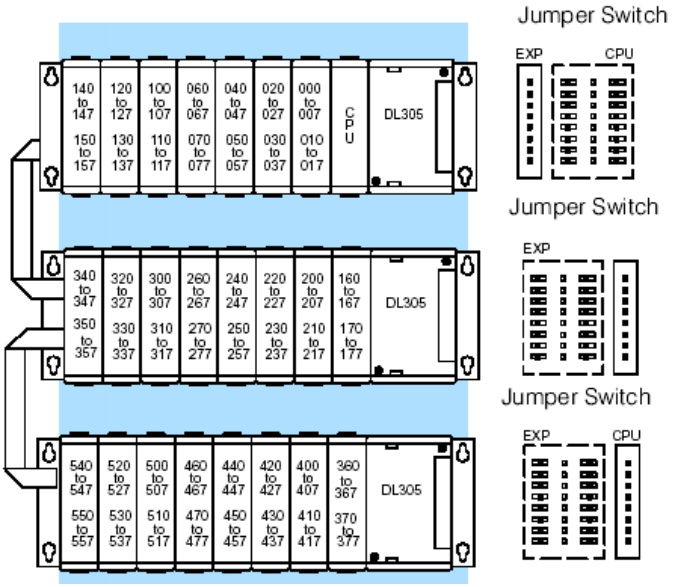
总的 I/O 点数：
第一扩展框架
8 点模块－120
16 点模块－240

第二扩展框架
1－8 槽 1－5 槽
8 点模块－160
16 点模块－320



8 槽框架和两个 8 槽扩展框架

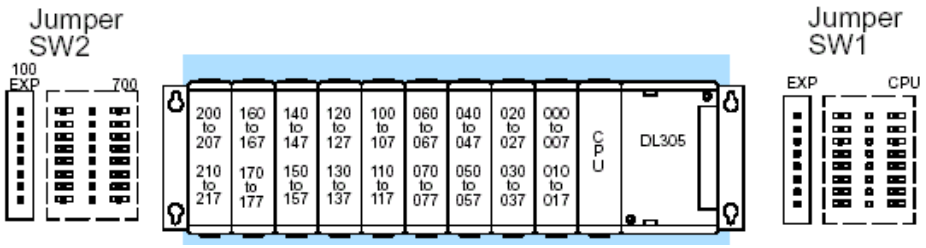
总的 I/O 电数：
2 个扩展框架
8 点模块－184
16 点模块－368



10 槽本地 CPU 框架的 I/O 设置

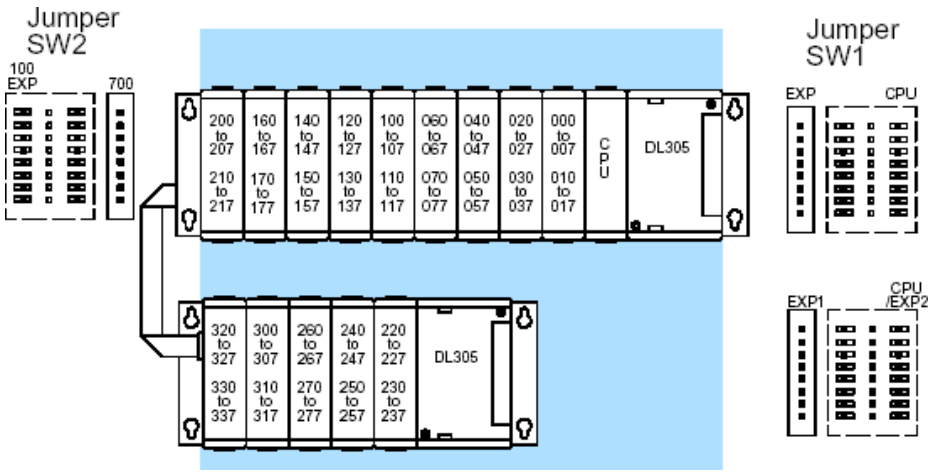
10 槽框架

总的 I/O 点数：
8 点模块－72
16 点模块－144



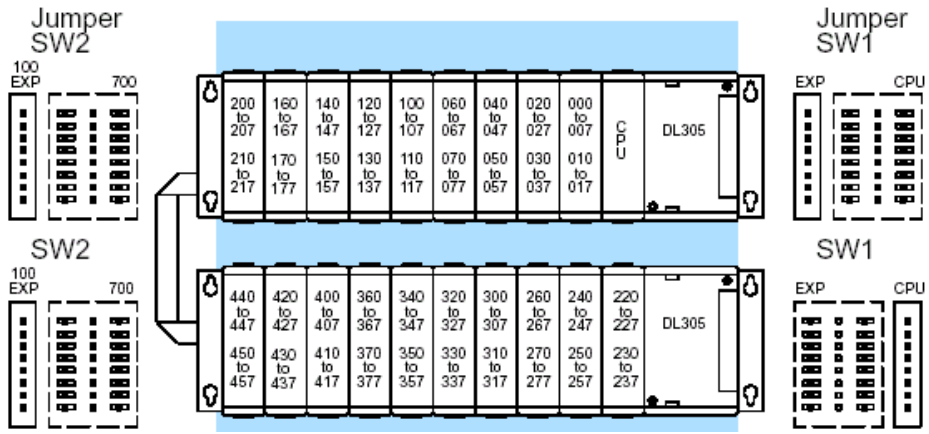
10 槽框架和 5 槽框架，使用 16 点 I/O 模块

总的 I/O 点数：
8 点模块－112
16 点模块－224



10 槽框架和 10 槽扩展框架，使用 16 点 I/O 模块

总的 I/O 点数：
8 点模块－152
16 点模块－304



远程 I/O 扩展

怎样添加一个远程 I/O 通道

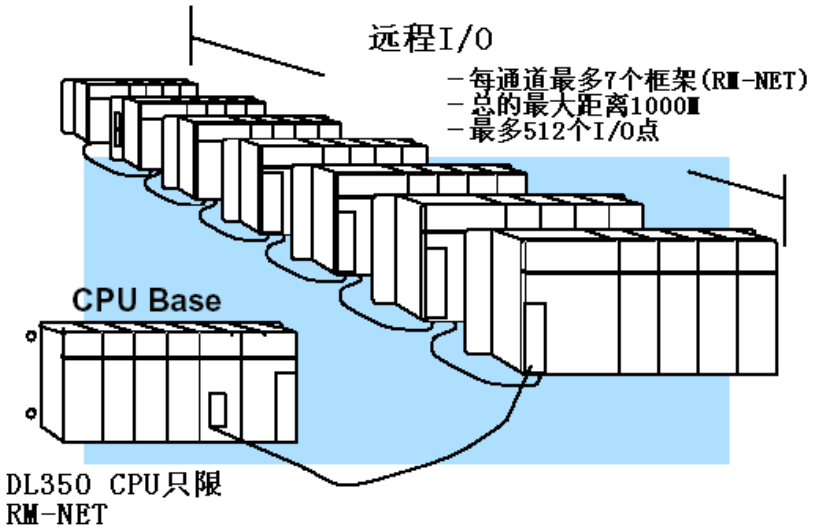
如果系统有许多传感器或其他现场设备离本地的中心 CPU 比较远（达到 1000M），这时远程 I/O 就非常有用了。DL350 支持一个内建的 remote master，但是 DL305 系列没有远程 I/O 模块。所以，您必须使用 DL205 或 DL405 框架来作为 slave channels。添加远程 I/O 模块的方法如下：

•**DL350 CPU**：该 CPU 的 Port2 是一个内置的远程 I/O 通道。

	DL350
本地 CPU 框架所支持的远程控制器的最大数量	1
CPU 的内置远程 I/O 通道	1
每个通道所支持的最大 I/O 点数	512
最大远程 I/O 点数	512
每个通道所支持的远程 I/O 框架的最大数量（RM-NET）	7

远程 I/O 点的状态是存放在不同的 CPU 寄存器中的，因此它不会减少本地的 I/O 点数。请查阅 DL205 远程 I/O 手册中有关远程 I/O 的设置及编号方法的章节。以下将介绍如何来设置内置的远程 I/O 通道。

下图是一个 CPU 框架连 7 个远程框架的例子。其中的远程框架是 DL205 或 DL405 框架。



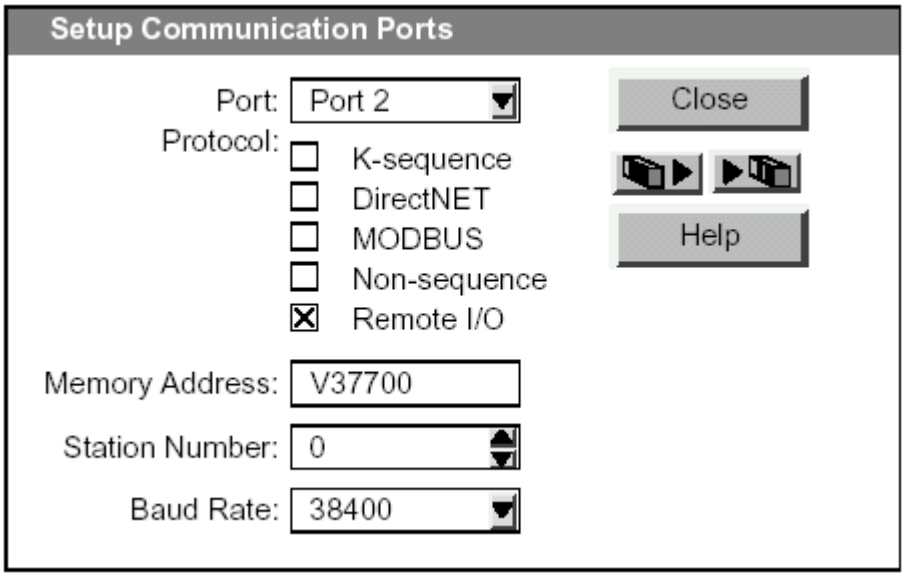
设置 CPU 的远程 I/O 通道

本节将介绍如何来设置 DL350 CPU 的远程 I/O 通道。一些详细内容将编在远程 I/O 手册的 D2-REMIO-M 相关章节中，这些对您设置远程从单元十分有帮助。

DL350 CPU 内置远程 I/O 通道的功能和 DL205 以及 DL405 CPU 的是相同的。它能连最多 7 个远程框架，I/O 点数可达 512，总的距离可达 1000M。您也许会想起上一章中提到 DL350 CPU 的 Port2 支持多种通信协议。若要用手持编程器来设置该端口的话，请使用 56 菜单并根据提示做相应的操作。若要使用 DirectSOFT，选择 PLC 菜单，然后选择 Setup 中的 Setup Sec. Comm Port…

•Port:选择“Port2”

•Protocol：在“Remote I/O”前的方框中点选，接着您就可以看到以下的对话框。



•**Memory Address**：选择一个 R 寄存器地址作为远程 I/O 分配区域的起始地址（出厂值为 R37700）。该区域独立于系统的远程控制器。

•**Station Number**：选择“0”让 DL350 作为主局。局号 1～7 作为远程子局保留。

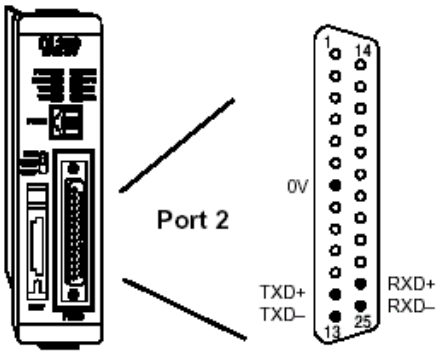
•**Baud Rate**：波特率可选择 19200～38400。这里我们一般选 38400，若发现数据传输出错或干扰问题，可以把它改成 19200。请注意：您必须在远程子局上设置相同的波特率（通过 DIP 开关）。

然后点击“”按钮，把 Port2 的设置内容写入 CPU 中，最后关闭。

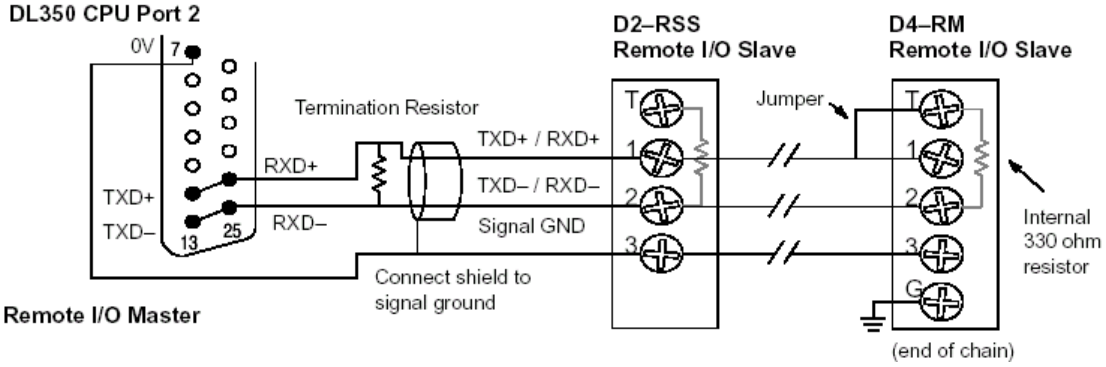
接下来的步骤就是把所要用到的设备都连起来。

DL350 CPU 上的 Port2 是一个 25 针的连接器，外形如右图所示。

- Pin 7　Signal GND
- Pin 12　TXD+
- Pin 13　TXD-
- Pin 24　RXD+
- Pin 25　RXD-



现在我们可以讨论一下 DL350 和远程框架之间是怎样接线的。远程 I/O 的连接是采用 3 线式、半双工的形式。尽管 DL350 的 Port2 是一个 5 线式的端口，我们必须把其中的两个针脚短接，如下图所示。

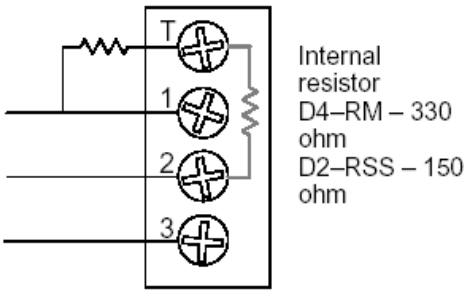


Port2 的接线如图所示。　注意屏蔽线一定要接信号地。终端电阻（Termination Resistor）要加在 CPU 的外面，并且尽可能靠近 CPU 的 25 针连接器。这样可以尽量减少由于电缆过长而带来的一些电气干扰。

注意在最后一个远程框架上接一根跳线和内部终端电阻。

最理想的情况就是电缆两端的终端电阻和电缆本身的阻抗相匹配。由于电缆的阻抗大于 330Ω，在最后一个远程框架上加一个电阻，如右图所示。如果小于 330Ω，在远程框架的 1 脚和 2 脚之间并联一个适当的电阻。请记住要按大小来依次排列从而匹配电缆的阻抗。电阻都应该在 100Ω～500Ω 之间。

外加一些电阻



设置远程 I/O 子局

完成 DL350 CPU 的 Port2 接线后，按照以下步骤来设置远程 I/O 子局。完整的使用说明在远程 I/O 的手册上。

- 设置和 CPU Port2 匹配的波特率
- 给每个子局分配一个局号，从 1 到 7。系统中的任意一个装置都必须有一个单独的局号。主局只有一个（局号为 0）。

设置远程 I/O 映射表

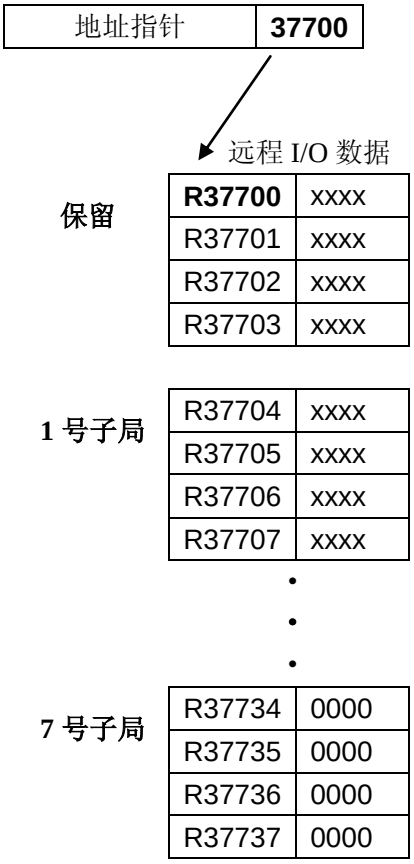
内置远程 I/O 通道的映射表的起始地址就是我们在 Port2 设置时所选的地址。每个远程子局都有由 4 个字组成的 I/O 映射表。R37700～R37703 为系统保留。

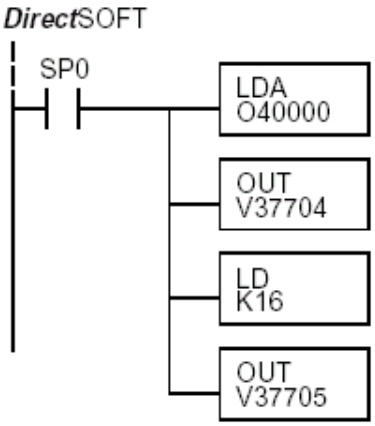
上电后，CPU 从映射表中读取数据并按以下规则来判断 4 个字的含义：

1. 子局输入数据的起始地址
2. 子局的输入点数量
3. 子局的输出点起始地址
4. 子局的输出点数量

这个映射表有 32 个字。如果您的系统中远程子局数量少于 7 个，则剩下的那些映射表都应该填入 0。例如，一个 3 子局系统会有 12 个字的数据和 16 个字的“0000”。梯形图程序中一部分必须在上电时对此进行设置。要把一个地址放入映射表中，可以如右图所示使用 LDA 指令。使用常规的 LD 指令来写入子局的输入点数或输出点数。

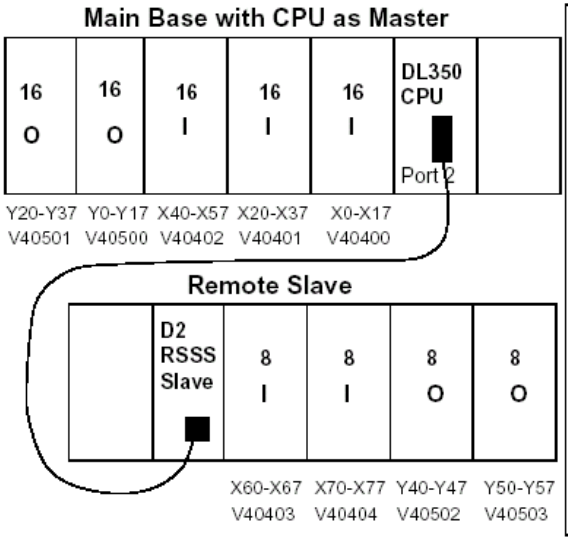
以下几页将介绍一个子局的例子。





请看以下一个采用远程 I/O 的系统。DL350 的内置远程 I/O 通道和一个子局相连，该子局局号我们定义为 1 号。主局和子局中的波特率都设成 38400。

我们可以把这些远程 I/O 点当作任何类型的 I/O 点，只要选择好正确的寄存器区域即可。由于我们有足够的空间来作为标准 I/O 地址（I 和 Q），我们可以把远程 I/O 点安排在主局的 I/O 点之后（分别为 I60 和 Q40）。



Remote Slave Worksheet

Remote Base Address 1 (Choose 1–7)

Slot Number	Module Name	INPUT		OUTPUT	
		Input Addr.	No. Inputs	Output Addr.	No.Outputs
0	08ND3S	X060	8		
1	08ND3S	X070	8		
2	08TD1			Y040	8
3	08TD1			Y050	8
4					
5					
6					
7					

Input Bit Start Address: X060 V-Memory Address:V 40403

Total Input Points 16

Output Bit Start Address: Y040 V-Memory Address:V 40502

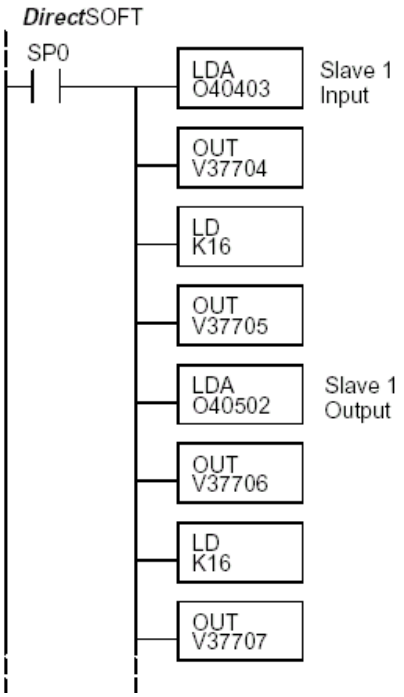
Total Output Points 16

远程 I/O 设置程序

使用以上的远程 I/O 工作表有助于我们在编写程序时组织好系统数据。其中的 4 个关键参数是在工作表的右下脚。有关地址的选用请参考第三章末尾的相关内容。

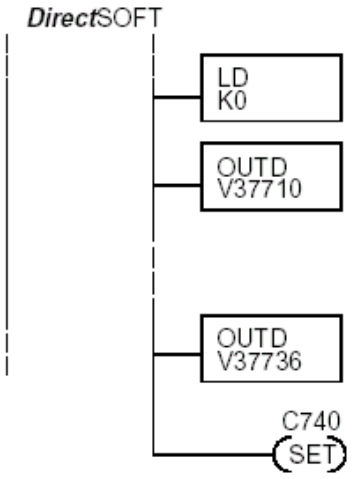
工作表中的计算结果要按照右边的程序来输入。请注意 LDA 和 LD 指令的区别。

以下几页将介绍剩余的一些设定程序。



当子局数量少于 7 个时，剩余的空间应写入零。因为 CPU 在工作时会把所有的非零区域当作子局的相关信息。

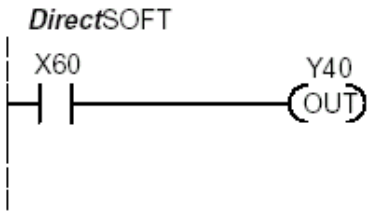
右边是一个给 2～7 号局所对应的地址写入零的例子。



上面程序的最后一行是用来给一个特殊继电器 M740 置位。这个特殊的接点是用来向 CPU 传达这样的信息：梯形图已完成了对远程 I/O 系统的设置。到时 CPU 就开始了和远程 I/O 的通讯。请不要忘记在远程 I/O 设置后加上该句。

远程 I/O 测试程序

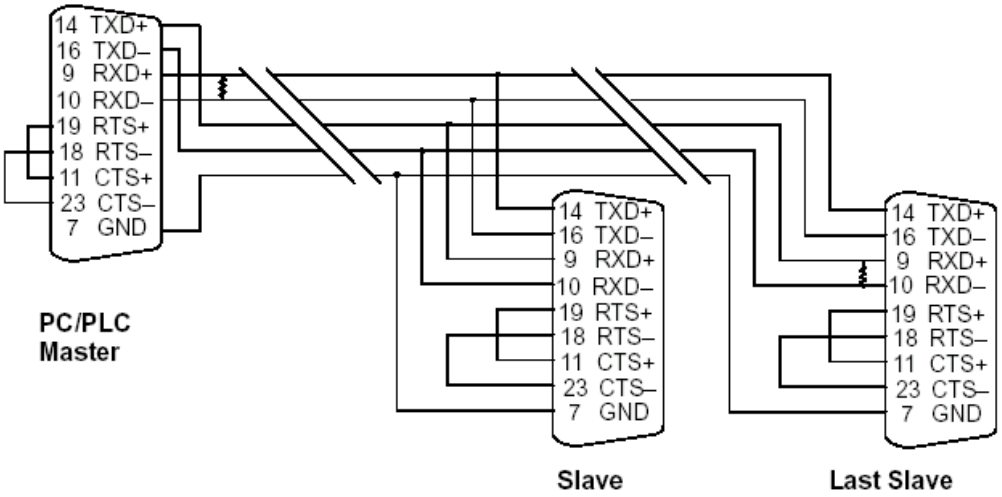
接下来我们可以来检验远程 I/O 的连接和设置程序是否正常工作。您可以简单地用右边所示的一句程序来检验。它以远程框架上的第一个输入点作为输入条件，以第一个输出点作为输出。当把 PLC 打到 RUN 之后，我们给远程框架上的那个输入点一个信号。若一切正常，则第一个输出点为 ON。



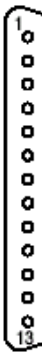
基于 MODBUS 和 *Direct*NET 的网络连接

设置 CPU 的端口

本节将介绍如何来设置 CPU 的内置端口，包括 MODBUS 和 DirectNET。这样您就可以通过 RTU 协议把 DL305 的 PLC 系统连接到 MODBUS 网络，或者 DirectNET 中的其他设备。MODBUS 主局系统必须能够发送 MODBUS 命令来读写适当的数据。要了解有关 MODBUS 协议的详细情况，请参考 Gould MODBUS Protocol reference Guide (P1-MBUS-300 Rev.B)。如果有一个更新的有效版本，在订购文件以前先和 MODBUS 提供者取得联系。若要了解 DirectNET 的详细信息，请订购我们的 DirectNET 手册，编号为 DA-DNET-M。



RS422 的电缆推荐使用 Belden 9855 或相当的。



25针
D型接

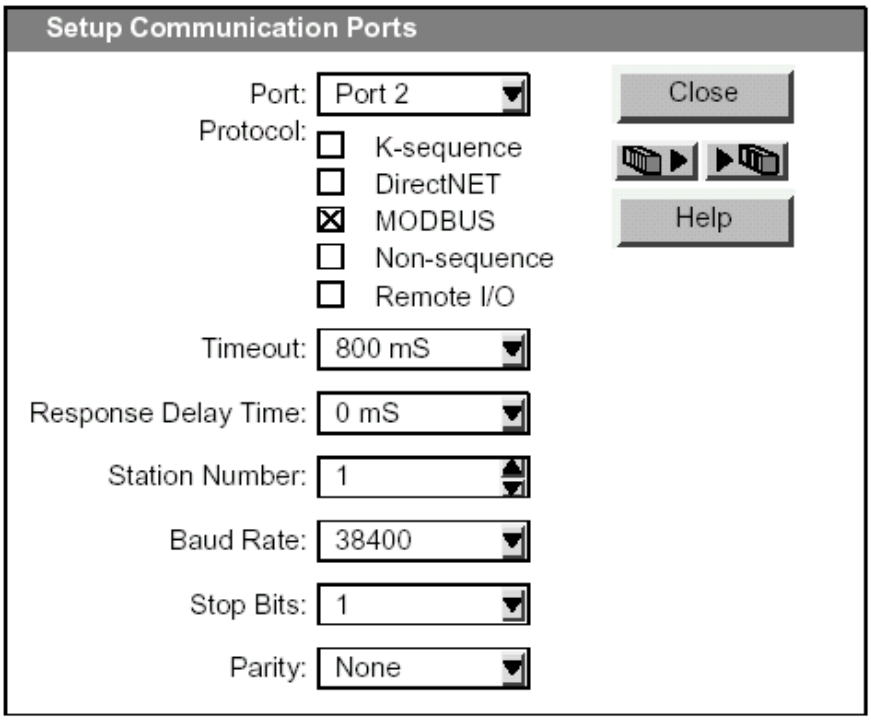
Port 2 Pin Descriptions (DL350 CPU)		
1	not used	
2	TXD	Transmit Data (RS232C)
3	RXD	Receive Data (RS232C)
4	RTS	Ready to Send (RS–232C)
5	CTS	Clear to Send (RS–232C)
6	not used	
7	0V	Power (–) connection (GND)
8	0V	Power (–) connection (GND)
9	RXD +	Receive Data + (RS–422)
10	RXD –	Receive Data (RS–422)
11	CTS +	Clear to Send + (RS422)
12	TXD +	Transmit Data + (REMIO)
13	TXD –	Transmit Data – (REMIO)

Port 2 Pin Descriptions (Cont'd)		
14	TXD +	Transmit Data + (RS–422
15	not used	
16	TXD –	Transmit Data – (RS–422)
17	not used	
18	RTS –	Request to Send – (RS–422)
19	RTS +	Request to Send – (RS–422)
20	not used	
21	not used	
22	not used	
23	CTS –	Clear to Send – (RS–422)
24	RXD +	Receive Data + (REMIO)
25	RXD –	Receive Data – (REMIO)

MODBUS 端口设置

在 DirectSOFT 中，选择 PLC 菜单，然后选择 Setup 中的“Setup Sec. Comm Port…”。

- **Port**：选择“Port 2”。
- **Protocol**：在“MODBUS”前的方框中点选（手持编程器上可使用 56 菜单，并选择“MBUS”），接着您就可以看到以下的对话框。



- Timeout: 端口在发出信号后在报错之前所等待响应的时间
- Response Delay Time:从激活 RTS 线路到传送数据时的经过时间。这适用于不采用 RTS/CTS 握手方式的装置。RTS 和 CTS 线路必须桥联在一起使 CPU 能够发送任意数据。
- Station Number:要使 CPU 端口作为 MODBUS 主局，选择 1。MODBUS 协议支持的子局数量为 247，但对于 DL350 只能使用 1～90。每个子局都必须有一个独立的局号。上点后端口自动作为子局，直到 DL350 执行到相关的网络设置指令后，该端口才变为主局。然后，该端口又变为子局模式直到梯形图程序再次使用它。
- Baud Rate:可选择的波特率有 300，600，900，2400，4800，9600，19200，和 38400。首先选择最高的波特率，若遇到数据出错再逐渐降低。注意：联网的设备要设定成相同的波特率。详细情况请参考相关产品手册。
- Stop Bits：设定通讯协议中的停止位，可为 1 或 2。
- Parity：可选 none（无校验）、even（偶校验）和 odd（奇校验）。

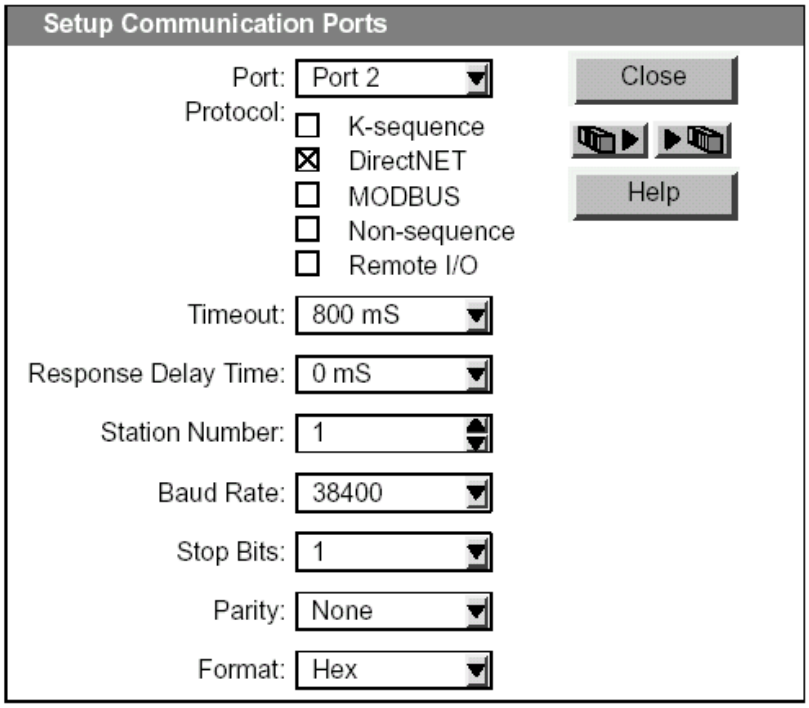
 按下该按钮来把设定好的参数写入 CPU，然后点击关闭。

DirectNET 端口设置

在 DirectSOFT 中，选择 PLC 菜单，然后选择 Setup 中的“Setup Sec. Comm Port…”。

• Port:选择“Port 2”。

• Protocol：选择“DirectNET”（或在手持编程器上使用 56 菜单，并接着选“DNET”），此时您应该可以看到以下对话框。



• Timeout: 端口在发出信号后在报错之前所等待响应的时间。

•Response Delay Time:从激活 RTS 线路到传送数据时的经过时间。这适用于不采用 RTS/CTS 握手方式的装置。RTS 和 CTS 线路必须桥联在一起使 CPU 能够发送任意数据。

•Station Number:要使 CPU 端口作为 MODBUS 主局，选择 1。MODBUS 协议支持的子局数量为 247，但对于 DL350 只能使用 1～90。每个子局都必须有一个独立的局号。上点后端口自动作为子局，直到 DL350 执行到相关的网络设置指令后，该端口才变为主局。然后，该端口又变为子局模式直到梯形图程序再次使用它。

•Baud Rate:可选择的波特率有 300，600，900，2400，4800，9600，19200，和 38400。首先选择最高的波特率，若遇到数据出错再逐渐降低。注意：联网的设备要设定成相同的波特率。详细情况请参考相关产品手册。

• Stop Bits：设定通讯协议中的停止位，可为 1 或 2。

•Parity：可选 none（无校验）、even（偶校验）和 odd（奇校验）。

• Format：选择：hex 或 ASCII 格式。


按下该按钮来把设定好的参数写入 CPU，然后点击关闭。

子局的操作

本节将介绍当您把一些设备设定成 DirectNET 子局或 MODBUS 子局时（DL350），这些设备是怎样来和 CPU 进行通讯的。MODBUS 的主局和子局之间必须通过 MODBUS RTU 协议来通讯。主局上的软件必须传送一个 MODBUS 功能码和 MODBUS 地址来确定 PLC 存储器位置。DirectNET 则是使用普通的 I/O 地址来访问 DL350 CPU 及其系统。不需要用梯形图编程来支持 MODBUS 和 DirectNET 子局的操作。

MODBUS 功能码

MODBUS 功能码决定了读写的状态，以及对象是一个数据点还是多个数据。DL350 支持的功能码如下所示：

MODBUS 功能码	功能	对应数据类型
01	读一组线圈	Q,M,T,C
02	读一组输入点	I,SP
05	Set/Reset 某个线圈	Q,M,T,CT
15	Set/Reset 一组线圈	Q,M,T,CT
03，04	从一个或多个寄存器中读取数据	R
06	向某个寄存器中写入数据	R
16	向多个寄存器中写入数据	R

MODBUS 地址设定

大部分主局软件的惯例允许您来指定某个 PLC 存储器空间，一般有两种方式可供选择：

- 通过指定 MODBUS 数据类型和及其地址
- 只通过对 MODBUS 地址的指定

如果您的主局要求数据类型及其地址

许多主局软件都允许您指定与PLC存储器对应的MODBUS数据类型及其地址。这是最简单的方法，但并不是所有的软件都支持这样的功能。

其实用来计算地址的方程式取决于您使用的 PLC 数据类型。PLC数据类型按以下两种用途分类：

- 离散——I,SP,Q,M,S,T,C（接点）
- 字——R,定时器经过值，计数器经过值

不管哪种情况，先把八进制地址转化成十进制，使之能适应MODBUS协议。以下是计算每组数据所用的等式。

DL350 寄存器类型	数量（十进制）	PLC 范围（八进制）	MODBUS 地址范围（十进制）	MODBUS 数据类型
离散型…把 PLC 地址转换成十进制 + 起始区域 + 数据类型				
输入（I）	512	I0~I777	2048~2560	输入
特殊继电器（SP）	512	SP0~SP777	3072~3584	输入
输出（Q）	512	Q0~Q777	2048~2560	线圈
控制继电器（M）	1024	M0~M1777	3072~4095	线圈
定时器（T）	256	T0~T377	6144~6399	线圈
计数器（C）	128	C0~C177	6400~6271	线圈
级（S）	1024	S0~S1777	5120~6143	线圈
字型… 把 PLC 地址转换成十进制 + 数据类型				
定时器经过值	256	R0~R377	0~255	输入寄存器
计数器经过值	128	R1000～R1177	512～639	输入寄存器
R 寄存器，用户数据	3072 4096	R1400~R7377 R10000~R17777	768～3839 4096 ～ 8191	保留寄存器
R 寄存器，系统数据	256	R7400~R7777	3480 ～ 3735	保留寄存器

以下例子表示如何来产生 MODBUS 主局的地址及其数据类型。

例 1：R2100

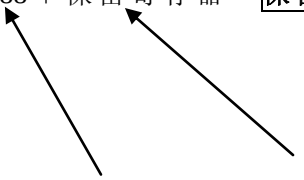
在用户寄存器中找到 R2100 作为 MODBUS 地址。

1. 在表格中找到 R 寄存器
2. 把 2100 转换成十进制数（1088）
3. 使用表格中的 MODBUS 数据类型

PLC 地址（Dec）＋数据类型

R2100=1088 十进制

1088＋保留寄存器＝**保留寄存器 1088**



R 寄存器 用户数据	3072 12288	R1400~R7377 R10000~R37777	768～3839 4096～16383	保留寄存器

例 2：Q20

找到输出点 Q20 的 MODBUS 地址。

1. 在表格里找到输出点
2. 把 20 转换成十进制数（16）
3. 加上起始地址（2048）
4. 使用表格中的 MODBUS 数据类型

PLC 地址（Dec）＋起始地址＋数据类型

Q20＝16 十进制

16＋2048＋线圈＝**线圈 2064**



输出点	1024	Q0~Q1777	2048~3071	线圈

例 3：T10 当前值

找到相应的 MODBUS 地址来获取定时器 T10 的当前值

1. 在表格中找到定时器当前值
2. 把 10 转换成十进制（8）
3. 使用表格中的 MODBUS 数据类型

PLC 地址（Dec）＋数据类型

T10＝8 十进制

8 ＋ 输入寄存器 ＝**输入寄存器 8**

定时器当前值	256	R0~R377	0~255	输入寄存器

例 4：M54

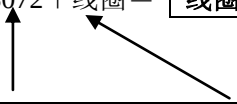
找到与 M54 对应的 MODBUS 地址

1. 找到表格中的中间继电器
2. 把 54 转换成十进制数（44）
3. 加上起始地址（3072）
4. 使用表格中的 MODBUS 数据类型

PLC 地址（Dec）＋起始地址＋数据类型

M54=44 十进制

44＋3072＋线圈＝ **线圈 3116**



中间继电器 M	2048	M0～M3777	3072 ～ 5119	线圈

若您的MODBUS主局软件需要一个的地址

有些主局的软件不支持用户对 MODBUS 数据类型及其地址的指定。相反，您只能指定一个地址。这种方法需要其他的方法来决定它的地址，这也是非常方便的。首先，MODBUS 地址也是分为数据类型和地址的。 这意味着单独地址可以不仅表示数据类型还可以表示地址。这经常作为“添加偏移”。有一点要注意的就是在您的主局软件中可能有以下两种寻址模式：

- 484 模式
- 584/984 模式

如果您的主局软件允许的话我们推荐您使用 584/984 寻址模式。这是因为该模式能够让您针对某个数据类型访问更多的寄存器空间。如果您的软件只支持 484 模式，则可能有些寄存器空间不能使用。其实用来计算地址的方程式取决于您使用的 PLC 数据类型。PLC 数据类型按以下两种用途分类：

- 离散——I,SP,Q,M,S,T,C（接点）
- 字——R,定时器经过值，计数器经过值

不管哪种情况，先把八进制地址转化成十进制，使之能适应 MODBUS 协议。以下是计算每组数据所用的等式。

DL350 寄存器类型	**数 量（十进制）**	**PLC 范围（八进制）**	**MODBUS 地址范围（十进制）**	**484 方式地址**	**584/984 方式地址**	**MODBUS 数据地址**
离散型…把 PLC 地址转换成十进制			**+ 起始区域**		**+ 相应的模式地址**	
输入 I	512	I0~I777	2048~2560	1001	10001	输入
特殊继电器 SP	512	SP0~SP777	3072~3584	1001	10001	输入
输出点 Q	512	Q0~Q777	2048~2560	1	1	线圈
中间继电器 M	1024	M0~M3777	3072~4095	1	1	线圈
定时器 T	256	T0~T377	6144~6399	1	1	线圈
计数器 C	128	C0~C177	6400~6527	1	1	线圈
级 S	1024	S0~S1777	5120~6143	1	1	线圈
字型… 把 PLC 地址转换成十进制				**+**	**相应的模式地址**	
定时器经过值	256	R0~R377	0~255	3001	30001	输入寄存器
计数器经过值	128	R1000~R1177	512~639	3001	30001	输入寄存器
用户存储区	3072 4096	R1400~R7377 R10000~R17777	768~3839 4096~8192	4001	40001	保留寄存器
系统存储区	256	R7400~R7777	3840~3735	4001	40001	保留寄存器

以下例子表示如何来产生 MODBUS 主局的地址。

例 1：R2100
584/984 模式

在用户寄存器中找到 R2100 作为 MODBUS 地址。

1．在表格中找到 R 寄存器
2．把 2100 转换成十进制数（1088）
3．加上此模式下的 MODBUS 起始地址（40001）

PLC 地址（Dec）＋ 模式地址
R2100=1088 十进制
1088＋40001＝ **41089**

R 寄存器 用户数据	320	R700~R777 R7400~R7777	488～768 3840～3735	4001	40001	保留寄存器

例 2：Q20
584/984 模式

找到输出点 Q20 的 MODBUS 地址。

1．在表格里找到输出点
2．把 20 转换成十进制数（16）
3．加上起始地址（2048）
5．加上该模式下的 MODBUS 地址（1）

PLC 地址（Dec）＋起始地址＋模式
Q20＝16 十进制
16＋2048＋1＝ **2065**

输出点 Q	1024	Q0~Q1777	2048~3071	1	1	线圈

例 3：T10 当前值
484 模式

找到相应的 MODBUS 地址来获取定时器 T10 的当前值

1．在表格中找到定时器当前值
2．10 转换成十进制（8）
3．加上此模式下的 MODBUS 起始地址（3001）

PLC 地址（Dec）＋数据类型
T10＝8 十进制
8 ＋ 3001 ＝ **3009**

定时器当前值	256	R0~R377	0~255	3001	30001	输入寄存器

例 4：M54
584/984 模式

找到与 M54 对应的 MODBUS 地址

1．到表格中的中间继电器
2．把 54 转换成十进制数（44）
3．加上起始地址（3072）
4．加上该模式下的 MODBUS 地址（1）

PLC 地址（Dec）＋起始地址＋模式
M54=44 十进制
44＋3072＋1＝ **3117**

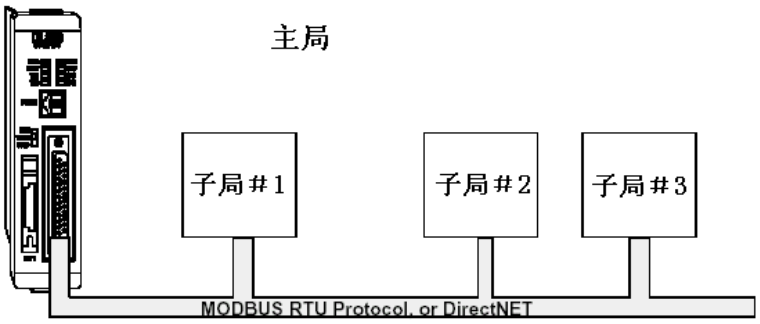
中间继电器 M	2048	M0～M3777	3072～5119	1	1	线圈

DirectNET 地址的确定

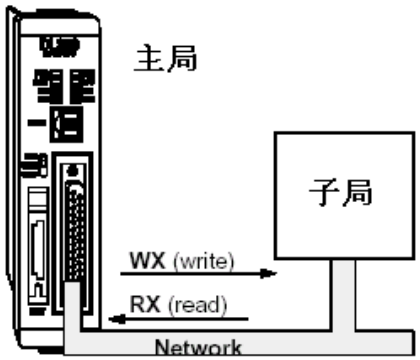
给 DirectNET 的子局分配存储器类型是非常方便的。用子局设备本身的地址即可。例如，若要通过 DirectNET 访问 PLC 中的 R2000，主局只要向子局的 R2000 发送请求即可。

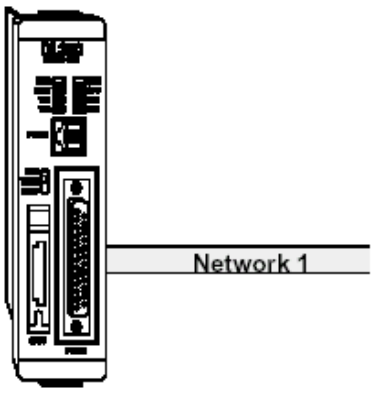
网络主局的操作

本节将介绍 DL350 作为主局怎样和 MODBUS 或 DirectNET 进行通讯。对于 MODBUS 网络，它采用 MODBUS RTU 协议。该协议必须由网络中的所有主局来解释。MODBUS 和 DirectNET 都是一主局/多子局形式的网络。本节将教您关于网络主局操作的梯形图编程。



当把 DL350 CPU 作为一个主局，您可以使用梯形图指令来进行初始化。WX 指令是用来进行写操作，RX 指令是用来进行读操作。在执行 WX 或 RX 指令之前，我们需要把相关的数据装入累加器堆栈中。当执行 WX 或 RX 指令时，它使用和指令中的数据相结合的存放在堆栈中的信息来彻底定义某个任务。

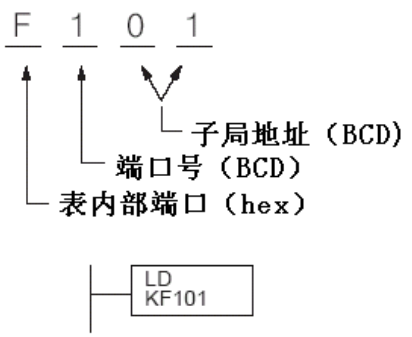




以下提供的信息可以帮助您一步一步地完成用来从子局收取数据的梯形图程序。

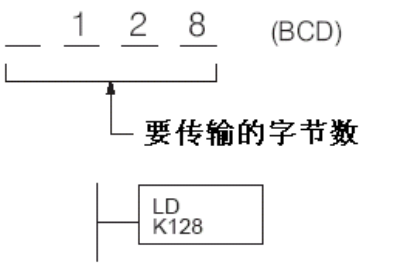
步骤 1：
定义主局端口号及子局号

第一个 LD 指令定义了主局的通讯端口号及子局的地址。该指令可以确定 90 个 MODBUS 子局或 90 个 DidrectNET 子局。字的格式如右所示。第一个字符“F”表示 CPU 端口是内部的（而不是插槽式的）。第二个字符“1”表示端口号为 1。这是逻辑上的端口号（0 表示上面的，1 表示下面的）。最后两位数字是包含的子局数量（01～90BCD）。



步骤 2：
写入要传送的字节数

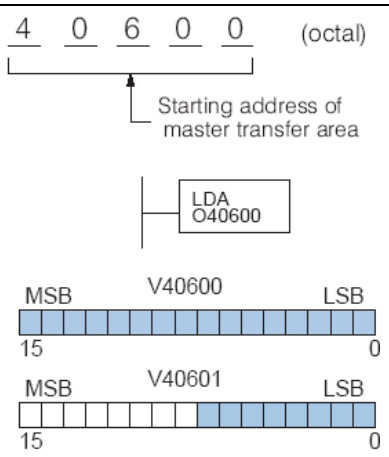
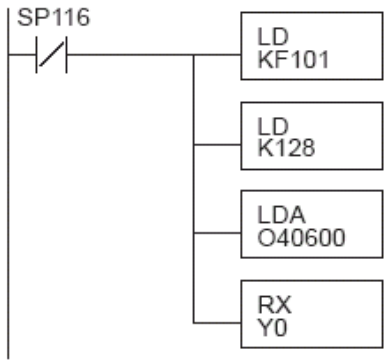
第二个 LD 指令决定了子局和主局之间传输的字节数。该值可以是 BCD 的 1～128。



该字节数还取决于您所获取的数据类型。例如，DL305 的输入点可以通过 R 寄存器来访问或直接作为输入点 I 来访问。但是，如果您只需要访问 I0~I27 的话，那么您必须使用输入点 I 的形式，因为 R 寄存器的访问是以两个字节为一个单位的。以下的表格说明了 DirectLOGIC 系列产品的各种不同的字节范围。

DL205/305405 寄存器	位数	字节数
R 寄存器 T/C 经过值	16 16	2 2
输入点（I,SP）	8	1
输出点 （Q,M,S,T/C）	8	1
scratch pad memory	8	1
诊断状态	8	1

DL305C（DL330/DL340 CPU）寄存器	位数	字节数
数据寄存器 T/C 累加器	8 16	1 2
I/O,内部继电器，移位寄存器位，T/C，级	8	1
scratch pad memory	8	2
诊断状态（5 字 R/W）	16	10

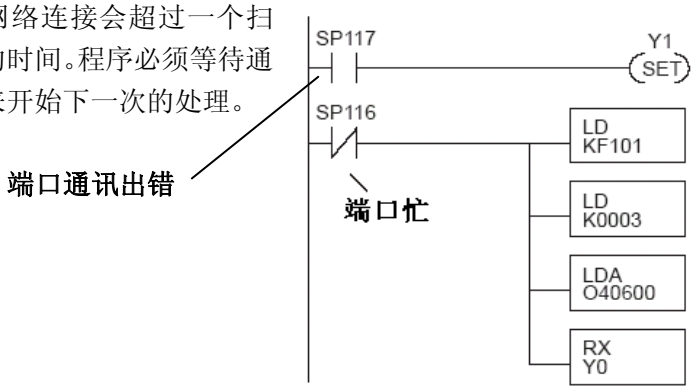
步骤 3： **确定主局的存储区域**	第三个指令是 Load Address（LDA）指令。这是用来装载将被传输的存储区域的首地址。LDA 指令的操作数是一个八进制数，执行后它被转换成十六进制并放入累加器中。 对于 WX 指令，DL350 CPU 将传送先前已确定的存储区域中的一组数据。 对于 RX 指令，DL350 CPU 从子局中读取一组数据并把它放到相应的寄存器中。	
	注意：由于 R 寄存器是 16 位的，您有可能不会用到整个字单元。例如，您只确定了 3 个字，并且是从子局中读取输出点 Q 的状态，那您只会用到 24 位的数据。因此，后面一个字只有低 8 位是有意义的。剩余的 8 个位没有用到。	
步骤 4： **确定子局的存储区域**	程序段的最后一个指令就是 WX 或 RX 本身。WX 指令是向子局写入数据的指令，而 RX 是从子局读出数据的指令。四条指令如右所示，您必须为子局确定一个起始地址和一个有效的数据。	
	• **DirectNET** 子局——在 WX 和 RX 指令中设定相同的地址作为子局的 I/O 地址 • MODBUS DL405,DL305(DL350 CPU),或 DL205 子局——在 WX 和 RX 指令中设定相同的地址作为子局的 I/O 地址 • MODBUS 305C（DL330/DL340 CPU）子局——如下表所示的方法来把 DL305 的地址转换成 MODBUS 地址。	

DL305C（DL330/DL340 CPU）系列 CPU 寄存器的 MODBUS 类型

PLC 存储器类型	PLC 基本地址	MODBUS 基本地址	PLC 存储器类型	PLC 基本地址	MODBUS 基本地址
T/C 经过值	R600	V0	T/C	CT600	GY600
I/O 点数	IO000	GY0	中间继电器	CT600	GY160
数据寄存器	R401, R400	V100	移位寄存器	SR400	GY400
级（只限 D3-330）	S0	GY200			

从梯形图程序进行的通讯

通常的网络连接会超过一个扫描周期的时间。程序必须等待通讯停止来开始下一次的处理。

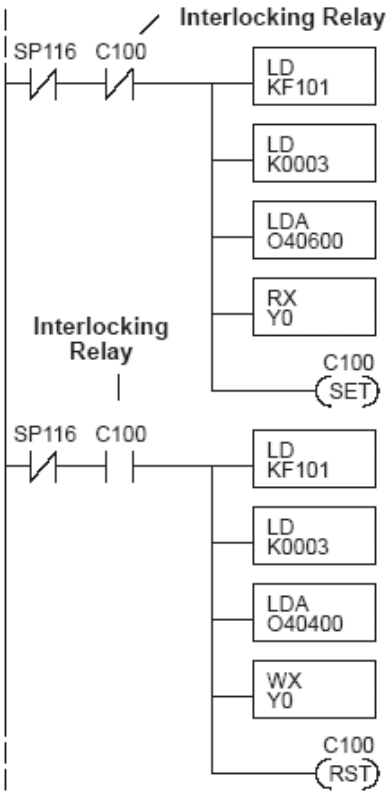


能作为主局的端口附带两个特殊继电器（请见附录 D）。一个表示“端口忙”（SP116），另一个表示“端口通讯出错”（SP117）。以上的例子说明了对于一个只读数据的网络主局是如何来运用这两个接点的。当 PLC 和子局进行通讯时，SP116 就 ON。当该位 OFF 以后，程序才能接受下一个请求。

多重读写的互锁

如果您要在梯形图程序中使用多重读写指令，您必须使用互锁来使每个程序都能够执行。如果您不使用互锁，那么 CPU 将只执行第一段程序。因为每个端口同一时间只能进行一次处理。在右边的例子中，RX 指令执行后，C100 被置位。当端口完成了通讯任务后，第二段程序就执行，C100 也被复位。

如果您使用级指令，您可以把两段程序放在不同的级中来确保它们都能够执行。



第五章　维护与故障排除

9

本章将介绍：

——硬件维护

——系统诊断

——**CPU** 指示灯

——**PWR** 指示灯

——**RUN** 指示灯

——**CPU** 指示灯

——**BATT** 指示灯

——通讯问题

——**I/O** 模块的故障排除

——噪音故障排除

——启动系统与程序调试

硬件维护

硬件维护

DL305 是一个低维护系统，只要定期对它做一些检查就能降低许多发生故障的可能性。常规的维护和检查应注意以下两个主要项目：

- 空气质量（环境温度，气流等）
- CPU 电池

空气质量维护

您的系统所处环境的空气质量会影响到系统的性能。如果您的系统是放在控制柜里的，请检查其环境温度是否超过了允许的范围。如果柜子中有空气过滤器，请在必要的时候给予清洗或更换来保证。建议您最好每隔一两个月来对系统来进行一次检查。请保证系统是工作在规定的环境条件下的。

电池报警灯

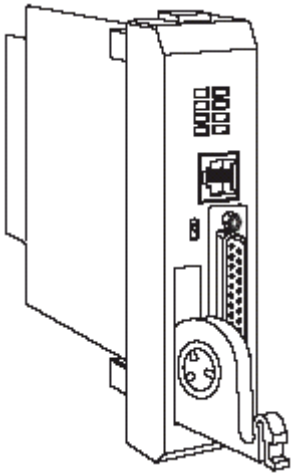
CPU 有一个电池 LED 来表示其中的电池是否还电量充足。您应该对其进行定期的检查并在必要的时候更换电池。您也可以通过编程来查看电池的电流是否低。SP43 这一特殊继电器在电池需要更换的时候会自动置 ON。

CPU 电池的更换

CPU 电池是用来保持 R 寄存器中的内容及系统参数的。电池使用寿命为 5 年。



注意：在安装或更换 CPU 电池以前，请备份您的 R 寄存器内容及系统参数。您可以通过 DirectSOFT 中保存程序，R 寄存器以及系统参数到个人计算机的硬盘或软盘上。



电池的安装步骤：

1. 按住电池舱门上的塑料片把舱门旋开。
2. 把电池放在纽扣槽中
3. 关上舱门并确保其已可靠到位
4. 把安装日期记录下来



警告：严禁对电池进行充电或把废旧电池扔入火中。因为电池有可能爆炸或产生有毒气体。

可以在编程时使用**系统诊断**

系统诊断

您的 DL305 系统在每个扫描周期都会完成预定义的诊断程序。该诊断功能能够检测到 CPU 和 I/O 模块的各种错误。基本的错误等级有两个，致命错误（fatal）和非致命错误（non-fatal）。

致命错误

致命错误是 CPU 检测到的可能使系统不能安全地或正常地工作的一些错误。如果 CPU 在运行模式下发生了致命错误，CPU 会自动切换到编程模式。（请记住，在编程模式下所有的输出点都会断开。）如果 CPU 在编程模式下发生了致命错误，则 CPU 在问题解决之前无法进入运行模式。

以下是一些致命错误的例子：

- 基本框架电源不正常
- 奇偶错误或 CPU 故障
- I/O 设置错误
- 某些程序错误

非致命错误

非致命错误是 CPU 标记出来的需要注意的错误。它们既不会禁止 CPU 从运行模式切换到编程模式，也不会阻止 CPU 进入运行模式。用户可以在编程时使用一些特殊继电器，使系统能够检测出这些非致命错误。然后有需要的话，您可以把系统安全关闭或切换到编程模式。

以下是一些非致命错误的例子：

- 电池电量低
- 所有的 I/O 模块错误
- 某些程序错误

查找诊断信息

诊断信息可以按消息细节的等级在许多地方查找到。

- CPU 会自动记录错误码和 FAULT 信息到两个独立的存储区域中。您可以通过手持编程器或 DirectSOFT 软件来查看。
- 手持编程器会显示错误码以及简短的描述。
- DirectSOFT 提供了错误码和一条错误信息
- 本手册的附录 B 中有一张完整的按错误码来排序的错误信息。

许多这些信息指向补充的存储区域。这些信息可以作为额外信息的参考。这些存储区域的参考是以 R 寄存器和特殊继电器的形式。以下两张表格给和某些类型的错误信息相关的存储器区域命名。特殊继电器表也包括可以在编程时使用的状态指示灯。有关每个特殊继电器的详细信息请参考附录 D。

与错误码相关的 R 存储器区域

错误等级	错误种类	诊断寄存器
用户定义	和 FAULT 指令一起使用的错误码	R7751
系统错误	致命错误码	R7755
	主要错误码	R7756
	细小错误码	R7757
语法	发生语法错误的地址	R7763
	语法检查时发现的错误码	R7764
CPU 扫描	上一次从编程模式切换到运行模式后的扫描次数	R7765
	当前扫描时间（ms）	R7775
	最小扫描时间（ms）	R7776
	最大扫描时间（ms）	R7777

与错误码相关的特殊继电器（SP）

上电与实时继电器	
SP0	初始复位
SP1	常时 ON
SP3	1 分钟时钟脉冲
SP4	1 秒钟时钟脉冲
SP5	100ms 时钟脉冲
SP6	50ms 时钟脉冲
SP7	每隔一次扫描 ON
CPU 状态继电器	
SP11	强制 RUN 状态
SP12	TERM RUN 状态
SP13	TEST RUN 状态
SP14	TEST HOLD 状态
SP15	TEST PROGRAM 状态
SP16	TERMINAL PROGRAM 状态
SP20	STOP 继电器
SP21	BREAK 指令已执行
SP22	中断许可继电器
系统监控继电器	
SP40	重度异常继电器
SP41	轻度异常继电器
SP43	电池异常继电器
SP46	通讯错误继电器
SP47	I/O 配置异常继电器
SP50	FAULT 指令已执行
SP51	看门狗超时
SP52	语法错误
SP53	无法处理逻辑
SP54	智能模块通讯出错

累加器状态继电器	
SP60	小于标志继电器
SP61	等于标志继电器
SP62	大于标志继电器
SP63	零标志继电器
SP64	半借位标志继电器
SP65	借位标志继电器
SP66	半进位标志继电器
SP67	进位标志继电器
SP70	符号标志继电器
SP71	接点指定出错标志
SP73	溢出标志继电器
SP75	数据出错标志继电器
SP76	读零标志继电器
通讯监控继电器	
SP116	端口 2 通讯标志
SP117	端口 2 通讯错误标志
SP120	0 号槽模块忙
SP121	0 号槽通讯错误
SP122	1 号槽模块忙
SP123	1 号槽通讯错误
SP124	2 号槽模块忙
SP125	2 号槽通讯错误
SP126	3 号槽模块忙
SP127	3 号槽通讯错误
SP130	4 号槽模块忙
SP131	4 号槽通讯错误
SP132	5 号槽模块忙
SP133	5 号槽通讯错误
SP134	6 号槽模块忙
SP135	6 号槽通讯错误
SP136	7 号槽模块忙
SP137	7 号槽通讯错误

错误信息表

DL350 CPU会自动记录系统错误码和您通过FAULT指令来创建的用户信息。CPU 会记录错误的代码及其发生的日期和时间。两个独立的区域存放着这些信息。

- 错误码表——系统可以记录最多 32 条错误。当错误发生时，以前记录的错误分别往下移，最近的错误放到顶部。当表格已放满时还发生错误，则排在最后一个错误将被丢掉。
- 信息表格——系统可以记录最多 16 条信息。当一条信息被触发，以前记录的信息分别往下移，最近的信息被放到顶部。当表格已放满时还发生错误，则排在最后一条消息将被丢掉。

以下表格说明了消息的错误表格。

日期	时间	消息
1993－05－26	08：41：51：11	*Conveyor-2 stopped
1993－04－30	17：01：11：56	*Conveyor-1 stopped
1993－04－30	17：01：11：12	*Limit SW1 failed
1993－04－28	03：25：14：31	*Saw Jam Detect

您可以通过DirectSOFT的PLC Diagnostic子菜单或手持编程器来访问错误码表格和消息表格。有关细节请查阅 DirectSOFT 和手持编程器的手册。

下面的例子说明了如何通过手持编程器的 5C 菜单来查看错误码。首先显示的是最近的错误码和消息。您可以通过 PREV 和 NXT 键来翻阅。

？？？？？？？？？？？？？？？？？？？？s-10hp 上无 5c 菜单？？？？？？？？

系统错误码

系统能够记录最多 32 个错误码。这些日志中的错误码是所有 DL305 系统生成的错误消息的子集。这些错误由 CPU 或手持编程器生成，这取决于实际的错误。附录 B 中有关于错误码的更详细的描述。

错误能在各种时候被检测到。尽管如此，绝大多数都是在系统上电，进入运行模式，或在手持编程器上键入信息时发生的错误或非法请求。

错误码	描述
E003	软件超时
E004	无效指令（RAM 奇偶错误）
E041	CPU 电池电量低
E043	存储卡电量低
E099	超出程序存储空间
E101	CPU 存储卡内容丢失
E104	写操作失败
E151	无效命令
E155	RAM 错误
E201	终端块丢失
E202	I/O 模块丢失
E203	保险丝断
E206	用户 24V 电源故障
E210	电源故障
E250	I/O 链通讯错误
E251	I/O 奇偶错误
E252	新的 I/O 配置
E262	I/O 超出范围
E312	通讯错误 2
E313	通讯错误 3
E316	通讯错误 6
E320	超时
E321	通讯错误
E499	打印指令中无效文本
E501	进入失败
E502	无效地址
E503	无效命令
E504	无效参数/数值
E505	无效指令
E506	无效操作

错误码	描述
E520	错误操作:CPU Run 模式
E521	错误操作:CPU Test Run 模式
E523	错误操作:CPU Test Program 模式
E524	错误操作:CPU Program 模式
E525	模式切换不在 TERM 模式
E526	离线单元
E527	在线单元
E528	CPU 模式
E540	CPU 被锁
E541	口令错误
E542	口令重置
E601	存储器满
E602	指令丢失
E604	参数丢失
E610	I/O 类型错误
E611	通讯 ID 错误
E620	超出存储器
E621	EEPROM 存储器非空
E622	无手持编程器 EEPROM
E624	只限 R 寄存器
E625	只限编程
E627	错误的写操作
E628	存储器类型错误（应为 EEPROM）
E640	比较错误
E650	手持编程器系统错误
E651	手持编程器 ROM 错误
E652	手持编程器 RAM 错误

编程错误码

下面的列表说明了编程时可能发生的错误。当您把 CPU 切换到 RUN 模式或使用手持编程器来检查程序时，这些错误会被检测到。同时 CPU 会把 SP52 置位并把错误码写入 R7755。附录 B 提供了有关错误码的更详细的信息。

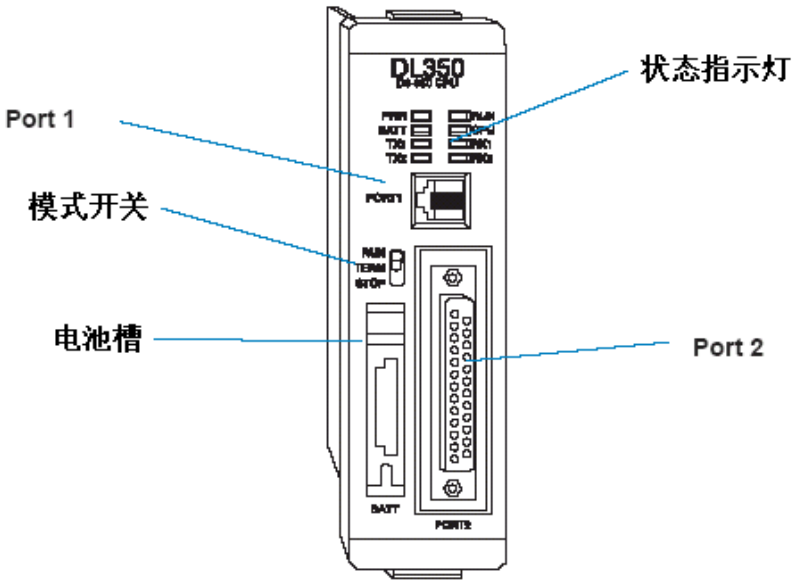
错误码	描述
E4**	CPU 中无程序
E401	缺少 END
E402	缺少 LBL
E403	缺少 RET
E404	缺少 FOR
E405	缺少 NEXT
E406	缺少 IRT
E412	SBR/LBL>64
E413	FOR/NEXT>64
E421	复制级参考
E422	复制 SBR/LBL 参考
E423	嵌套循环
E431	非法的 ISG/SG 地址
E432	非法的跳转（JUMP）地址
E433	非法的 SBR 地址
E434	非法的 RTC 地址
E435	非法的 RT 地址
E436	非法的 INT 地址
E437	非法的 IRTC 地址
E438	非法的 IRT 地址
E440	非法的数据地址
E441	ACON/NCON
E451	错误的 MLS/MLR
E452	输入线圈 I 用作输出线圈
E453	缺少 T/C
E454	错误 TMRA
E455	错误的 CNT
E456	错误的 SR

错误码	描述
E461	堆栈溢出
E462	堆栈下溢
E463	逻辑错误
E464	回路丢失
E471	复制线圈参考
E472	复制 TMR 参考
E473	复制 CNT 参考
E480	CV 位置错误
E481	CV 未连接
E482	CV 超出
E483	CVJMP 位置错误
E484	无 CV
E485	无 CVJMP
E486	BCALL 位置错误
E487	没有块被定义
E488	块位置错误
E489	CR 块定义错误
E490	没有块级
E491	ISG 位置错误
E492	BEND 位置错误
E493	BEND I 错误
E494	无 BEND

CPU 指示灯

DL350 CPU 的前部有一些指示灯，它们可以用来对系统进行诊断。以下的表格给出了对于不同的指示灯的状态可能出现的问题清单。其后有一个针对每个指示灯的详细分析。

指示灯状态	可能的问题
PWR（off）	1. 系统电压不正常 2. 电源/CPU 有故障 3. I/O 模块或其他组件的电源短路 4. 电功率消耗超出预算
RUN（亮不了）	1. CPU 编程错误 2. 开关在 TERM 位置 3. 开关在 STOP 位置
CPU（on）	1. 电气干扰 2. CPU 有缺陷
BATT（on）	1. CPU 电池电量低 2. 缺少 CPU 电池或接线不良
TX1	1．Port1 发送数据
RX1	1．Port1 接收数据
TX2	1．Port2 发送数据
RX2	1．Port2 接收数据



PWR 指示灯

CPU 的电源指示灯的不亮一般有 4 种情况：

1. 基本框架所使用的电源不正确或无法使用
2. 框架电源故障
3. 其他系统组件发生短路
4. 系统功率消耗超过预算

框架使用的电源不正常

如果框架使用的电源不正常，则 CPU 和（或）基本框架也许不能正常工作或根本无法工作。



警告：为了避免触电事故的发生，检查接线时一定要给系统断电。

1. 首先，切断系统的电源来检查所有的进线。
2. 如果您使用的是一个独立的中断面板，请检查那里的接线是否接错。
3. 如果检查无误，请给系统上电并测量框架上的电源端子，检查输入电压是否符合要求。如果电压不正常，请关闭系统并设法去解决问题。
4. 如果所有的接线和输入电源都正常，则请把框架寄给我们来维修。

CPU 故障

最好的检查 CPU 是否有故障的方法就是换一块新的 CPU 来试试。如果您发现主电源波动比较大，则 CPU 和基本框架已损坏。如果您怀疑这是电源损坏的原因，您可以在将来安装一个可以去除有害电压毛刺的整流器（line conditioner）。

装置或模块导致的电源不工作

如果使用系统的 5V 电源的模块或外部设备发生了故障，则有可能导致电源不能正常工作。该 5V 是来自框架上的端子或通讯端口。

测试此类设备的方法：

1. 去掉 CPU 的电源。
2. 断开所有的外部设备（也就是通讯电缆）和 CPU 断开。
3. 再次运行系统。

如果此时电源工作正常，则外部设备和电缆中至少有一个被短路了。如果此时电源还是工作不正常，则按照下面的步骤检查出问题的模块：

如果 PWR 指示灯亮，则问题可能出在某一个模块上。为了区分是哪个模块出了问题，切断系统的电源并每次拿走一个模块直到 PWR 灯亮为止。

请按照以下的程序：

- 切断框架的电源
- 拿走一个模块
- 重新给框架通电

弯曲了模块上的连接器针脚可能导致这种后果。检查连接部位是否有这种问题。

超出功率预算

如果在指示灯不亮以前，该机器已经在长时间内工正常好的话，一般不会是功率预算出了问题。超出功率预算的问题一般都是发生在上电的时候，因为此时 PLC 正处在工作状态而且输入点和输出点都需要更多的电流。



警告：如果超过了功率预算，PLC 会重启。如果您怀疑系统的功率预算有什么问题，请在此时仔细检查。超出了功率预算，将会产生一些导致人员伤害或设备损坏的不可预测的后果。请检验模块的功率损耗是否在所使用的框架的允许范围之内。详情请参考第四章系统设计和配置。

RUN 指示灯

如果 CPU 不能进入 RUN 模式（RUN 指示灯不亮），一般都是程序有问题，除非 CPU 发生了致命错误。如果发生了致命错误，CPU 灯会亮。您可以通过编程设备来纠正这个错误。
如果您在使用 DL350 并且试着用编程设备来切换模式，请确保开关是在 TERM 位置的。
对于这样的出错，手持编程器和 DirectSOFT 都会返回一条消息。您也可以通过 AUX 功能来检测出这个问题，这要取决于什么问题。大部分的常规编程出错是因为漏掉了“END”指令。所有的程序都要用“END”来结尾。附录 B 中有详细的错误码解释。

CPU 指示灯

如果 CPU 指示灯亮了，则 CPU 发生了一个致命错误。一般而言，这不是编程上的问题，而是硬件上的故障。您可以通过重新上电来解决这个问题。如果错误消除了，您应该监视系统并找出产生错误的原因。您会发现有时候问题是由于引入了外部电源的高频杂波而造成的。如果怀疑是接地问题，可以检查接地并安装电源滤波器。若重新上电后还是没有消除错误，或问题又产生了，您应该更换 CPU。

BATT 指示灯

如果 BATT 指示灯亮了，则说明 CPU 中的电池没有接好或需要更换。系统上电后会不断监视该电池的电压。

通讯问题

如果您不能和 CPU 进行通讯连接，请检查以下项目：
- 电缆没有连接好
- 电缆中有断线或内部接线有误
- 电缆没有正常中止或接地
- 与之相连的设备没有设定相应的波特率（上面的端口为 9600 波特，下面的端口可通过 56 菜单来设置）
- 与之相连的设备发送的数据有误
- 两者的地存在电位差
- 电源噪声产生连续的错误
- CPU 的端口坏了，需要更换
- 如果您使用的是 DirectSOFT，请参考快速手册中的故障排除一节

如果出现了错误，指示灯会一直亮直到通讯成功为止。

I/O 模块故障排除

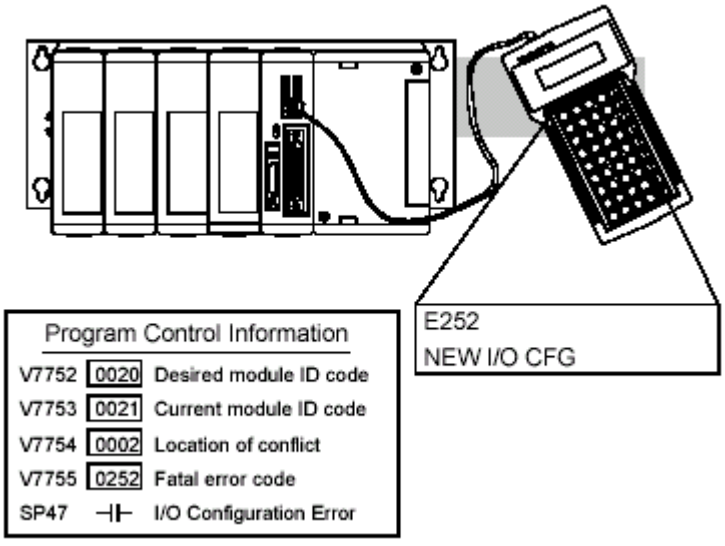
检查项目

如果您怀疑是 I/O 错误，可能有几种原因造成：

- 保险丝熔断
- 模块发生松动
- 24VDC 输出不正常
- 模块本身有故障
- I/O 配置检查检测到一个 I/O 配置的改动

I/O 诊断

如果模块没有提供一点问题的线索，请执行手持编程器上的 42 菜单或 DirectSOFT 中的 I/O diagnostics。这两种方法都能查看框架号，槽号及模块上的问题。 问题一旦解决，该灯就会灭掉。I/O 出错不会使 CPU 从 RUN 模式变为编程模式，但是有些特殊继电器可以用于梯形图程序来读取错误。应用程序就能采取相应的行动，比如进入编程模式或进行正常关闭。下图说明了出现错误的一个例子。



一些快速步骤

当处理 DL305 系列的 I/O 模块时，您必须知道一些情况。这些能够帮您更快地解决问题。

- 输出模块无法检测短路的或开路的输出点。如果您怀疑某个模块上的一个或几个输出点有问题，您应该测量公共端和该输出点之间的电压。请记住，当使用数字电压表时必须考虑到输出器件，比如双向可控硅或晶体管的漏电流。如果输出点没有接负载，则 OFF 的输出点有可能表现为 ON。
- 模块上的 I/O 指示灯只是逻辑上的指示灯。这意味着这些指示灯所表示的 ON 或 OFF 状态只是从 CPU 角度来说的。如果模块上的输出器件（双向可控硅或晶体管）被损坏了，则与之对应的指示灯依然能够正常工作。如果输入模块上的指示灯亮了，则说明输入回路是正常工作的。要确定是否工作正常，可以把输入信号去掉，再看看相应的指示灯是否灭掉。
- 当把现场设备接到 I/O 模块上时，漏电流的存在也许会成为一个问题。如果一个输出装置的漏电流过大，则有可能产生错误的输入信号。要解决这一问题，可以在输入回路中串一个电阻。阻值的大小取决于漏电流及电源电压的大小，但通常都是在 10K～20KΩ 之内。同时要考虑到电阻的功率应符合要求。
- 最简单的判断一个模块是否有问题的方法就是用一块好的来代替它。尽管如此，如果您怀疑是其他的设备导致了模块的问题，则更换模块后应该也会发生同样的问题。需要注意的一点是，您最好在用其他模块替代之前先检查一下设备的接线和电源电压。

测试输出点

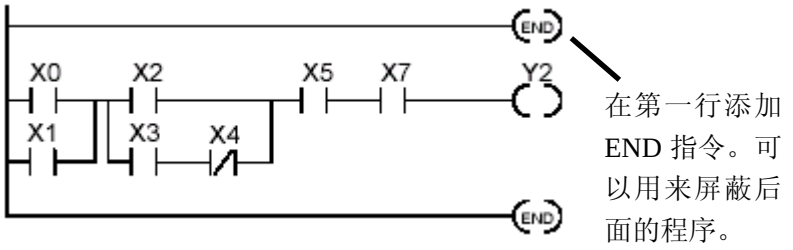
如果您要独立于程序外来检查 I/O 问题，请按以下的程序：

步骤	措施
1	使用手持编程器或 DirectSOFT 来连接 PLC。
2	切换到编程模式
3	找到地址 0
4	在地址 0 插入“END”指令。（这能够屏蔽用户的程序，从而防止 I/O 点的状态发生变化。）
5	切换到 RUN 模式
6	使用编程设备来测试相应的 I/O 点
7	当完成了 I/O 测试以后，删除地址 0 的“END”



警告：对某些系统应用而言，强制输出可能会产生一些不可预测机器动作，这有可能造成机器损坏或人员伤害。请确保您已对此作了充分的准备。

通过手持编程器按键来检测输出点



？？

？？

噪音故障排除

电气噪音问题

噪音问题是最难诊断的一个问题之一。电气噪音可以通过多种途径进入系统但可以归为两类：导电型的和辐射型的。判断噪音是如何进入系统有可能比较困难，但对两种噪音的处理方式却是类似的。

- 导电型的噪音是由于电线的接触，面板的接线等所产生的电气干扰。这些噪音有可能进入 I/O 模块，电源接线，通讯地，或底盘接地。
- 辐射型噪音是不通过直接的电气连接而是类似于无线电波的形式的电气干扰。

降低电气噪音

当电气干扰不能被消除时，我们可以降低它的程度使之不会影响系统的正常工作。

- 大多数的噪音问题是系统接地不良引起的。有一个良好的接地点是唯一的也是最好的解决噪音问题的办法。若接地点是无效的，那么就在尽可能靠近系统的地方安装一根接地棒。保证所有的接地线都是单点接地的并且不以菊花链的方式进行互连。用接地的金属来包围整个系统。在把噪音引入系统时，一根不固定的接线所产生的影响相当于一根大的天线；所以，您应该接好系统的每一根线。不固定的接地线比系统中的其他接线更容易产生问题。如果针对如何来给系统接地还有什么问题的话，您可以参考第二章中的相关内容。
- 电气噪音还有可能通过 CPU 和 I/O 模块的电源进入系统。安装一个隔离变压器可以解决这一问题。DC 电源应该有一个良好的接地。切换直流电源一般会产生比线性电源更多的噪音。
- 把输入接线和输出接线分开来。绝对不要把 I/O 接线靠近高电压的线路。

启动机器和调试程序

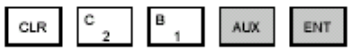
DL350 CPU 提供了一些功能可以帮助您在启动机器时或之前来调试程序。本节将对以下的主题进行讨论，这将会对您非常有帮助。

- 程序的语法检查
- 复制参数检查
- TEST 模式
- 特殊指令
- 运行时间编辑
- 强制 I/O

语法检查

尽管手持编程器和 DirectSOFT 都在写入程序时提供检查，但您有可能需要检查一个已修改的程序。两种编程设备都提供一种检查程序语法的方法。比如，您可以使用编程器上的 21 菜单，CHECK PROGRAM 或者 DirectSOFT 中的 PLC Diagnostics 菜单来检查程序的语法。这个功能能够检查出程序中的多种错误。以下例子说明了如何使用手持编程器上的这一功能。

使用 21 菜单来进行程序检查




```
AUX 21 CHECK PRO
1:SYN 2:DUP REF
```

选择语法检查（默认选择）



（若程序很短则不会显示“BUSY”）

```
BUSY
```

将会显示两者之一

错误显示（举例）

```
$00050     E401
MISSING END
```

语法无误显示

```
NO SYNTAX ERROR
?
```

附录 B 中有完整的错误码。如果您检查出一个错误，按 CLR 键就会显示发生错误的指令。纠正错误以后再次检查语法，直到显示“NO SYNTAX ERROR”。

检查重复线圈号

您可以查看同一线圈的重复使用。两种编程设备都有这个功能。例如，您可以通过使用编程器的 21 菜单，CHECK PROGRAM 或 DirectSOFT 中的 PLC Diagnostics 菜单选项来检查重复使用的线圈。下面的例子说明了如何用手持编程器来操作。

使用 21 菜单来进行程序检查




```
AUX 21 CHECK PRO
1:SYN 2:DUP REF
```

选择检查重复线圈



(You may not get the busy display if the program is not very long.)

```
BUSY
```

将会显示两者之一

错误显示（举例）

```
$00024  E471
DUP COIL REF
```

语法无误显示

```
NO DUP REFS
?
```

如果您检查到一个错误，按 CLR 键会显示发生线圈重复的指令。纠正问题后继续执行该操作直到没有发现重复线圈。



注意：您可以在不同的地方使用同一个线圈，尤其在使用级式编程和（或）OROUT 指令时。重复线圈号检查会找到这些输出点，尽管它们是被正确使用的。

TEST-PGR 与 TEST-RUN 模式

Test 模式允许 CPU 以 TEST-PGM 模式开始，进入 TEST-RUN 模式后运行若干个扫描周期后再回到 TEST-PGM 模式。扫描周期数可选择 1～65525。Test 模式同样允许您在 TEST-PGM 和 TEST-RUN 模式之间切换时保持输出点的状态。要切换到 Test 模式，用手持编程器可以按 MODE 键，用 DirectSOFT 可以选择 PLC MODES 菜单。使用 Test 模式的最基本的好处就在于可以在 CPU 回到 TEST-PGM 模式时保持某些输出点和参数的状态。而且切换到 TEST-PGM 模式后，CPU 可以保持定时器和计数器的当前值。



注意：只能通过 DirectSOFT 来设置扫描周期的数量。该功能在手持编程器上不支持。但是，您还是可以用手持编程器在 TEST-PGM 和 TEST-RUN 模式之间进行切换。

使用手持编程器，当您选择 TEST 模式后所进入的实际的模式取决与在这之前 CPU 处在何种模式下。如果 CPU 在 RUN 模式下，则切换成 TEST-RUN 模式。如果 CPU 在 PROGRAM 模式下，则切换成 TEST-PGM 模式。切换到 TEST 模式后，您就可以在 TEST-PGM 和 TEST-RUN 模式之间任意切换。DirectSOFT 的各种菜单选项提供了更加灵活的切换模式的方法。下面例子说明了如何来用手持编程器来选择 TEST 模式。

使用 MODE 键来选择 TEST 模式（例子假设为 RUN 模式下）

MODE NEXT ENT

```
*MODE CHANGE*
GO TO T-RUN MODE
```

按下 ENT 键来选择 TEST-RUN 模式

ENT

```
*MODE CHANGE*
CPU T-RUN
```

再次使用 MODE 键，可以回到 RUN 模式，进入 PROGRAM 模式，或进入 TEST-PGM 模式。

CLR MODE NEXT NEXT ENT

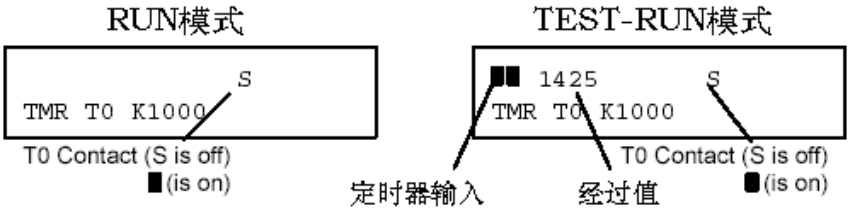
```
*MODE CHANGE*
GO TO T-PGM MODE
```

按下 ENT 来选择 TEST-PGM 模式

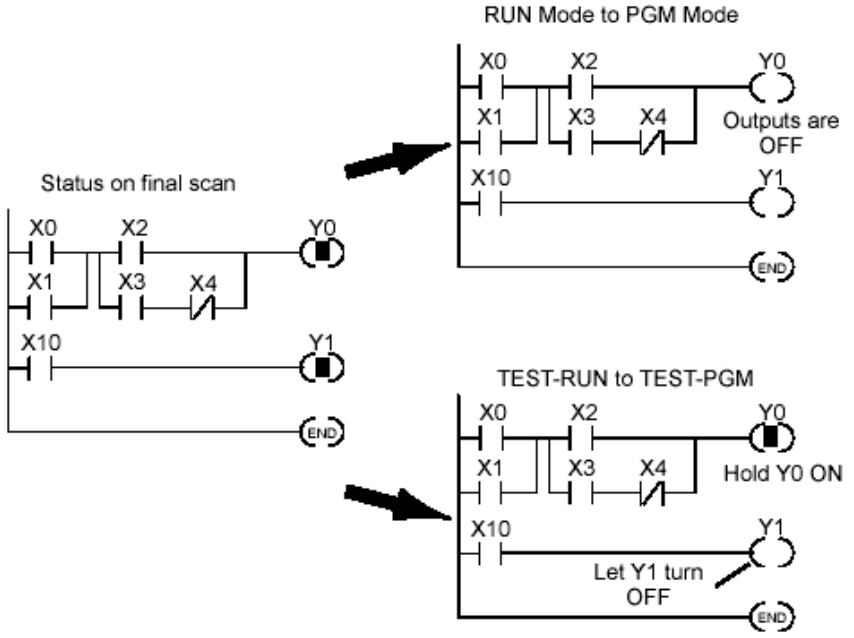
ENT

```
*MODE CHANGE*
CPU T-PGM
```

TEST 显示：在手持编程器上使用 TEST 模式时，您还可以查看到一些细节信息。对于一些指令，TEST-RUN 模式下看到的信息可能比 RUN 模式下看到的更详细。下面说明了定时器指令在两种模式下的不同显示。



输出保持：输出保持功能是非常有用的，因为它允许您维持一些关键的 I/O 点的状态。有时您需要修改程序，但又不希望某些操作停止。当在正常的 RUN 模式下，切换到编程模式时输出点都会 OFF。在 TEST-RUN 模式下，您可以设置每个输出点的状态或者让它保持先前的状态。该功能可以在 DirectSOFT 中设置。下图说明了 RUN 和 TEST-RUN 模式之间的不同。



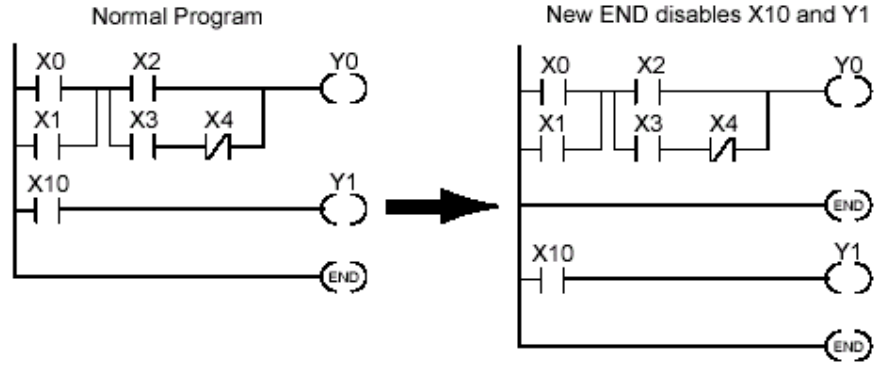
在您认为 TEST 模式是最好的选择之前，DL350 CPU 同样可以在 RUN 模式下修改程序。TEST 模式和 RUN TIME EDIT 功能的主要区别在于您不必逐个设定需要输出保持的输出点。当您使用 RUN TIME EDIT 功能时，CPU 在程序更新时会自动保持所有输出的当前状态。

特殊指令

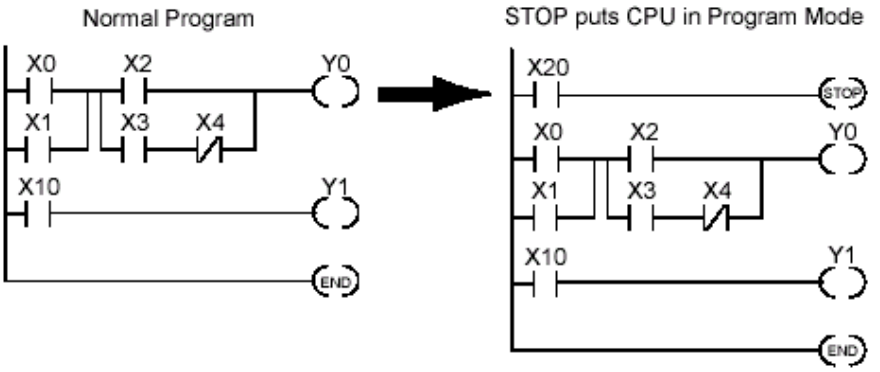
有一些特殊的指令有助于您在启动机器时顺利调试程序。

- END
- PAUSE
- STOP

END 指令：如果您需要快速地屏蔽程序，在不需要的程序处添加一条 END 指令。当 CPU 扫描到 END 指令，它将认为这是程序的末尾。下图说明了这个例子。



STOP 指令：在启动机器时，有时您需要快速关闭所有的输出点并回到编程模式。除了使用 TEST 模式，您还可以使用 STOP 指令。当执行该指令后，CPU 自动推出 RUN 模式并进入 PROGRAM 模式。记住，在 PROGRAM 模式下所有的输出点都 OFF。以下例子说明了使用一个条件来使 CPU 进入 PROGRAM 模式。



在上面的例子中，如果给 X20 一个触发信号，CPU 就会进入 PROGRAM 模式并关闭所有的输出点。

运行期间编程
（RUN TIME EDITS）

DL350 CPU 允许您在运行时修改程序。这些修改不是“bumpless”。相反，CPU 扫描不断地被中断（输出点维持原来的状态）直到程序完全修改完。这意味着如果输出点是 OFF 的，它会仍然保持 OFF 直到完成了程序的修改。同理，如果原来是 ON 的，它也会保持 ON。



警告：只有经授权的人员在对系统的每一个方面都非常了解的时候才能修改程序。RUN 模式下的修改会立即生效。请保证您已经彻底考虑了由此带来的后果并设法使人员伤害或机器损坏的可能性最小化。在运行期间编程的时候，一些重要的操作顺序会改变。

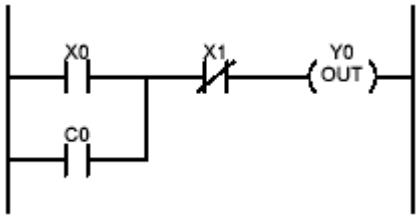
1. 如果新输入的指令中存在语法错误，CPU 将无法进入 RUN 模式。
2. 如果您删除的输出线圈当时为 ON，则该输出点的状态会一直保持直到您用编程设备来强制它为 OFF。
3. 在运行期间编程时输入点的状态不是得到普遍承认的。所以，如果您正在使用一个高速计数操作并且一个重要的输入点为 ON，则 CPU 有可能看不到这个变化。

并不是所有的指令都能用于运行期间编程。以下列举了能够的指令。

Mnemonic	Description
TMR	Timer
TMRF	Fast timer
TMRA	Accumulating timer
TMRAF	Accumulating fast timer
CNT	Counter
UDC	Up / Down counter
SGCNT	Stage counter
STR, STRN	Store, Store not
AND, ANDN	And, And not
OR, ORN	Or, Or not
STRE, STRNE	Store equal, Store not equal
ANDE, ANDNE	And equal, And not equal
ORE, ORNE	Or equal, Or not equal
STR, STRN	Store greater than or equal Store less than
AND, ANDN	And greater than or equal And less than

Mnemonic	Description
OR, ORN	Or greater than or equal Or less than
LD	Load data (constant)
LDD	Load data double (constant)
ADDD	Add data double (constant)
SUBD	Subtract data double (constant)
MUL	Multiply (constant)
DIV	Divide (constant)
CMPD	Compare accumulator (constant)
ANDD	And accumulator (constant)
ORD	Or accumulator (constant)
XORD	Exclusive or accumulator (constant)
LDF	Load discrete points to accumulator
OUTF	Output accumulator to discrete points
SHFR	Shift accumulator right
SHFL	Shift accumulator left
NCON	Numeric constant

使用程序的逻辑表示来描述该过程是如何工作的。在右边的例子中要把 X0 改成 C10。注意，现在假设 CPU 在 RUN 模式下。



使用 MODE 键切换到运行期间编程。

MODE NEXT NEXT ENT

MODE CHANGE
RUN TIME EDIT?

按下 ENT 键来选择运行期间编程

ENT

MODE CHANGE
RUNTIME EDITS

找到您所要修改的地方（X0）

SHIFT X SET A 0 SHIFT FD REF FIND

$00000 STR X0

按下方向键到 X 然后键入新的接点（C10）

→ → SHIFT C 2 B 1 A 0 ENT

RUNTIME EDIT?
STR C10

按 ENT 键来确定

ENT

OR C0

强制 I/O

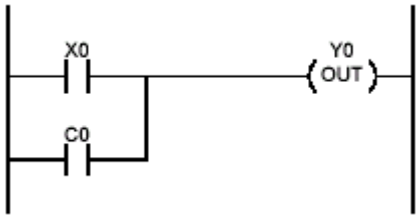
有些时候，尤其是在机器启动和排除故障的时候，您需要强制功能来使一个 I/O 点为 ON 或 OFF。在您用编程设备的强制功能之前，应该了解 DL350 是如何处理强制请求的。



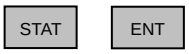
警告：只有经授权的人员在对系统的每一个方面都非常了解的时候才能修改程序。RUN 模式下的修改会立即生效。请保证您已经彻底考虑了由此带来的后果并设法使人员伤害或机器损坏的可能性最小化。

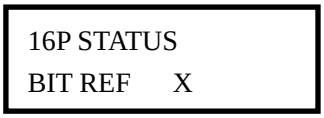
- **常规强制——**这类强制可以临时修改某个离散位的状态。例如，也许您需要使一个输入点 ON,尽管它是 OFF 的。您可以修改存放在镜像寄存器中的某个点的状态。在下一个扫描周期写入寄存器值时，该赋值才有效。这在测试情况下是最有用的，因为您也许需要强制某个位来触发一个动作。

下面举了一个简单的例子来说明如何使用手持编程器来强制 I/O 点。镜像寄存器的内容不随输入模块上的输入点状态的变化而变化。而且，程序的运行结果也不会刷新输出点的镜像寄存器。现在假设 CPU 在 RUN 模式下。

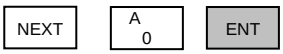


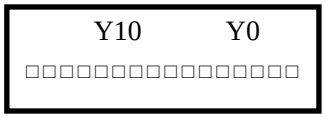
屏幕无任何显示的时候，使用以下的按键



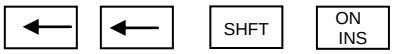


使用 PREV 和 NEXT 键来选择 Y 数据类型。（Y 出现后，按 0 使之从 Y0 开始。）

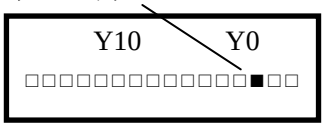




使用方向键来选择某个点，然后使用 ON 或 OFF 来改变它的状态



现在 Y2 为 ON



采用直接访问的常规强制

在屏幕无任何显示的时候，用以下的方式来给 **Y10** 为 **ON**。

SHFT | Y MLS | B 1 | A 0 | SHFT | ON INS



在屏幕无任何显示的时候，用以下的方式来给 **Y10** 为 **OFF**。

SHFT | Y MLS | B 1 | A 0 | SHFT | OFF DEL



附录 A　菜单操作功能

A

本附录将介绍：

——指令
——2*菜单：程序操作
——3*菜单：R 寄存器操作
——4*菜单：I/O 配置
——5*菜单：CPU 配置
——6*菜单：手持编程器配置
——7*菜单：EEPROM 操作
——8*菜单：口令设置

介绍

什么是菜单操作功能？

许多 CPU 设置都涉及到菜单操作功能（AUX）。该功能提供了许多不同的操作，比如清除程序，显示扫描时间，复制程序到手持编程器的 EEPROM 等。从影响系统参数的角度，它们被分为两类。您可以通过 DirectSOFT 或 DL205 手持编程器来使用菜单功能。与之配套的手册会教您具体的使用步骤。有些菜单功能只能在手持编程器上用，它们在 DirectSOFT 中是无效的。尽管如此，本附录提供了许多菜单操作的例子，您只需把编程设备的附带文件作为该信息的补充。注意，手持编程器的有些功能，DL350 CPU 有可能不支持。

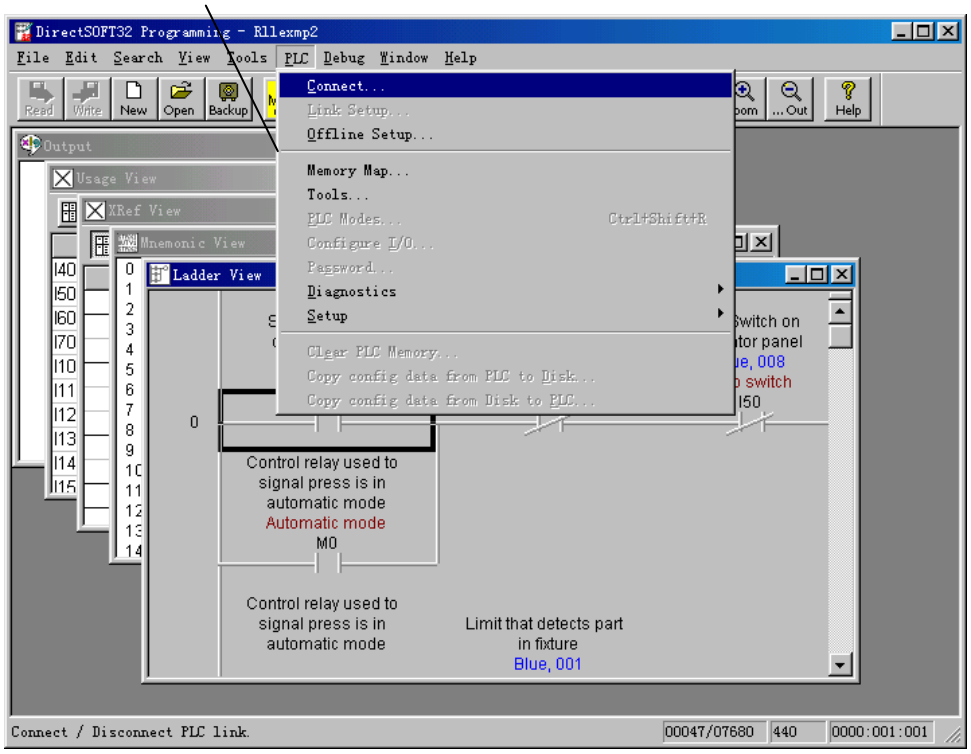
AUX 功能及其描述		350	HPP
AUX 2*—程序操作			
21	检查程序	✓	—
22	修改参数	⊗	—
23	清除程序块	✓	—
24	清除所有梯形图	✓	—
AUX 3*—R 寄存器操作			
31	清除 R 寄存器	✓	—
AUX 4*—I/O 设置			
41	显示 I/O 设置	✓	—
42	I/O 诊断	⊗	—
44	上电时 I/O 设置检查	⊗	—
45	选择设置	⊗	—
AUX 5*—CPU 设置			
51	修改程序名	✓	—
52	显示/修改日历	✓	—
53	显示扫描时间	✓	—
54	参数初始化	✓	—
55	监控定时器设定	✓	—
56	局号设定	✓	—
57	停电保持区域设定	✓	—
58	测试操作	⊗	—
59	位覆盖功能	⊗	—
5B	计数器界面设置	⊗	—
5C	显示出错日志	✓	—

AUX 功能及其描述		350	HPP
AUX 6*—手持编程器设置			
61	显示版本号	✓	✓
62	蜂鸣器 ON/OFF	⊗	✓
65	自诊断	⊗	✓
AUX 7*—EEPROM 操作			
71	读出	⊗	✓
72	写入	⊗	✓
73	比较	⊗	✓
74	空检查	⊗	✓
75	全清	⊗	✓
76	EEPROM 类型读出	⊗	✓
AUX 8*—口令操作			
81	修改口令	✓	—
82	口令打开	✓	—
83	口令关闭	✓	—

✓ 支持
⊗ 不支持
— 无效

通过 DirectSOFT 使用菜单功能

不管是在线编程时还是离线编程时，DirectSOFT 都提供了各种菜单选项。但有些功能只能在线编程时使用，有些只能离线编程时使用。下图表示 DirectSOFT 中的 PLC 操作菜单。



通过手持编程器使用菜单功能

您也可以通过手持编程器来使用菜单功能。有些菜单功能只能在手持编程器上使用。有时候菜单功能的名称或其他信息不能在一页中显示，您可以使用方向键来上下翻。同时，您也可以多按几下 CLR 键，这取决于当前的显示。

CLR AUX

```
AUX FUNCTION SELECTION
AUX 2* RLL OPERATION
```

使用 NXT 或 PREV 键来上下翻

NEXT

```
AUX FUNCTION SELECTION
AUX 3*    V OPERATION
```

按 ENT 键来选择子菜单

ENT

```
AUX 3*    V OPERATION
AUX 31 CLR V MEMORY
```

您也可以键入菜单号来直接进入子菜单

直接键入菜单号

CLR D 3 B 1 AUX

```
AUX 3*    V OPERATION
AUX 31 CLR V MEMORY
```

AUX 2*——程序操作

AUX 21，22，23和 24

您可以对程序作四种操作：

- AUX 21——程序检查
- AUX 22——定义号变更
- AUX 23——程序块删除
- AUX 24——程序全清

AUX 21 程序检查

在输入程序的同时，手持编程器和 DirectSOFT 都会自动检查错误。但是，当您要检查 CPU 中的程序使，可能有多种情况。有两种类型的检查可供选择：

- 语法
- 重复线圈

语法检查可以发现许多编程上的错误，例如漏掉 END 指令，不完整的 FOR/NEXT 循环等。如果您执行了此类检查并发现了错误，请查阅附录 B 中的错误代码。解决问题以后继续执行语法检查，直到屏幕显示“NO SYNTAX ERROR”。

使用重复线圈检查来确定您是否重复使用了输出线圈。注意，如果使用 OROUT 指令，该功能也会检查出重复的线圈。

该功能在 DirectSOFT 中的操作在 PLC 菜单下的 Diagnostic 子菜单中。

AUX 22 定义号变更

有时您需要改变 I/O 地址和中间继电器的定义号。您可以通过 AUX 22 作快速而又方便的修改。例如，您可以把所有的 X5 都改成 X10。

AUX 23 程序块删除

您会经常需要添加和删除程序的某一部分来解决问题。使用 AUX 23 您可以选择并删除某一段程序。DirectSOFT 没有与此对应的选项，但您可以通过其中的编辑工具来剪切某一段程序。

AUX 24 程序全清

AUX 24 的功能是删除 CPU 中的整个程序。在写入新的程序之前，必须执行该步骤。您也可以使用 DirectSOFT 中的 PLC 菜单下的 CLEAR PLC MEMORY 子菜单来实现该功能。

AUX 3*——R 寄存器操作

- AUX 31——R 寄存器全清

AUX 31 R 寄存器全清

普通使用的 R 寄存器内容全清。您也可以使用 DirectSOFT 中的 PLC 菜单下的 CLEAR PLC MEMORY 子菜单来实现该功能。

AUX 4*——I/0 配置

AUX 41 显示当前 I/O 配置

该功能用于显示当前的 I/O 配置。使用手持编程器您可以上下翻阅来查看每一个 I/O 槽的完整信息。屏幕上会显示每一个槽上的模块型号。DirectSOFT 中也有该功能，而且更加方便，因为您可以在屏幕中看到整个框架的情况。

AUX 5*——CPU 配置

AUX 51~58

该菜单下有多种操作，包括设置，查看和更改 CPU 设置。

- AUX 51——更改程序名
- AUX 52——显示/更改日历
- AUX 53——显示扫描时间
- AUX 54——参数初始化
- AUX 55——WATCH DOG 定时器设定
- AUX 56——通讯端口设定
- AUX 57——停电记忆区域设定
- AUX 5C——显示出错记录

AUX 51 更改程序名

DL305 系列的产品可以给 CPU 或手持编程器 EEPROM 中的程序命名。注意，EEPROM 中不能存储多个程序。程序名最多可以 8 个字符，包括字母和数字（A-Z,0-9）。AUX 51 是用于给程序改名，同样，您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。给程序命名以后，您只能通过 AUX 54 来清除程序名。但要保证您在执行该操作之前已经了解可能由此带来的后果！

AUX 52 显示/更改日历

DL350 CPU 有时钟和日历功能。如果您需要该功能，可以通过 AUX 52 来设置时间和日期。格式如下：

- 日期——年，月，日，星期（0～6，星期日～星期六）
- 时间——24 小时制，时，分，秒

您可以使用 AUX 功能来修改任何组件的时间和日期。但是，CPU 不会自动纠正日期与星期之间的不匹配。比如，您把时间调整到某月的 15 日，并且该天为星期四，您必须修改星期（除非 CPU 已经显示星期四）。同样，您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。

AUX 53 显示扫描时间

AUX 53 可以显示当前，最小，和最大的扫描时间。其中，最小和最大扫描时间是上一次从编程模式切换到 RUN 模式时的值。您也可以使用 DirectSOFT 中 PLC 菜单下的 Diagnostics 子菜单。

AUX 54 参数初始化

DL350 CPU 会保持特定的存储区域中的系统参数。有时，需要修改一些系统的参数。例如，如果您把一些中间继电器设置成停电记忆，这些将会被存储起来。



注意：只有在对系统存储器作了修改之后才有必要使用该功能。通常，只在修改程序或老的程序需要一种特殊的系统设定时才需要对系统存储器进行初始化。您也许经常会在修改了程序后忽略了系统存储器的初始化。

AUX 54 可以把系统存储器的值恢复到默认值。您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。

AUX 55 设置 Watchdog 定时器

DL350 CPU 含有一个用来监视扫描时间的 Watchdog 定时器。其出厂值为 200ms。如果扫描时间超过了 Watchdog 的时间限制，CPU 自动从 RUN 模式切换到编程模式。此时，手持编程器上会显示 E003 S/W TIMEOUT。

可以通过 AUX 55 来更改 Watchdog 定时器的设定值。您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。

AUX 56 CPU 局号设定

由于 DL350 CPU 有一个额外的通讯口，您可以通过手持编程器来给该端口设置地址和通讯参数。其默认设置如下：

- 1 号局
- HEX 模式
- 奇校验

该端口支持手持编程器，DirectSOFT 以及 DirectNET 通讯。DirectNET 的手册上提供了关于通讯设定的详细说明。



注意：只有当下面的端口在联网中使用时，您才需要使用此功能。否则，在默认值下系统就能正常工作。

使用 AUX 56 来设置局号和通讯参数。您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。

AUX 57 停电记忆区域设定

DL350 CPU 有些默认的区域是停电记忆的。这些区域能适合许多应用场合，您也可以根据自己的需要修改或不使用停电记忆区域。其默认设置如下：

存储区域	DL350	
	默认区域	可用区域
中间继电器	M1000~M1777	M0~M1777
R 寄存器	R1400~R37777	R0~R37777
定时器	无	T0~T377
计数器	C0~C177	C0~C177
级	无	S0~S1777

使用 AUX 57 可以改变停电记忆区域。您也可以使用 DirectSOFT 中 PLC 菜单下的 Setup 子菜单。



警告：DL350 CPU 没有电池。一旦发生停电，其内置的大电容会供电以保持其中的数据，但最多只支持一星期。有时这个时间会更短。如果该停电记忆区域对您十分重要，请选用相应的电池。

AUX 5C 显示出错记录

DL350 CPU 会自动记录系统的错误码和通过 FAULT 指令创建的用户信息。CPU 会记录错误码及其发生的日期和时间。以下两个独立的表格存放了这些信息。

- 错误码表格——系统最多能记录 32 条错误记录。当发生错误时，以前的错误记录分别往下移，最新的错误存放在第一行。当表格填满后，最后的一条错误码会被舍去。
- 信息表格——系统最多能记录 16 条信息。当信息被触发后，前面的信息分别往下移，最新的信息存放到第一行。当表格填满后，最后一条信息会被舍去。

下表是出错表格的一个例子。

日期	时间	信息
1997－05－26	08：41：51：11	*Conveyor-2 stopped
1997－04－30	17：01：11：56	*Conveyor-1 stopped
1997－04－30	17：01：11：12	*Limit SW1 failed
1997－04－28	03：25：14：31	*Saw Jam Detect

您可以使用 AUX 5C 来查看错误码和信息。您也可以使用 DirectSOFT 中 PLC 菜单下的 Diagnostics 子菜单来查看错误码和信息。

AUX 6*——手持编程其设置

AUX 61 显示版本号

大多数的工控产品都会不断完善其功能。有时这些的新的特性只有在固件（firmware）发布以后才能正常使用。使用 AUX 61 可以方便地查看 CPU 和手持编程器的版本号。您也可以使用 DirectSOFT 中 PLC 菜单下的 Diagnostics 子菜单来查看这些信息。

AUX 7*——EEPROM 操作

AUX 71~76

有些菜单功能是用于 CPU 中的程序和手持编程器中 EEPROM 中的程序互相转移的。

- AUX 71——把 CPU 中程序写入手持编程器 EEPROM
- AUX 72——把手持编程器 EEPROM 中内容写入 CPU
- AUX 73——比较 CPU 和手持编程器 EEPROM 中的内容
- AUX 74——空检查
- AUX 75——手持编程器 EEPROM 全清
- AUX 76——显示 EEPROM 类型（CPU 和手持编程器）

AUX 71 读出

使用 AUX 71 可以拷贝 CPU 存储器中的信息到手持编程器 EEPROM 中。您可以像上一张表格中的那样把手持编程器 EEPROM 中的某些内容拷贝到 CPU 中。其数量取决于 CPU。

AUX 72 写入

使用 AUX 71 可以拷贝手持编程器 EEPROM 中的信息到 CPU 存储器中。您可以像上一张表格中的那样把 CPU 中的某些内容读出。

AUX 73 比较

AUX 73 的功能是比较 CPU 和手持编程器 EEPROM 中的内容您可以像前一张表格所示比较不同类型的信息。其中有一个选项是“etc”，您可以用它来检查所有的区域，从而避免了重复操作。

AUX 74 空检查

使用 AUX 74 可以检查手持编程器 EEPROM 中是否为空。当您需要拷贝一个程序到 EEPROM 时，执行这个菜单将会是个好注意。

AUX 75 全清

使用 AUX 75 可以清除手持编程器 EEPROM 中所有内容。把程序从 CPU 中拷贝到手持编程器中时应该使用该菜单。

AUX 76 显示 EEPROM 类型

使用 AUX 76 可以查看手持编程器中的是何种类型的 EEPROM。

AUX 8* ——口令操作

AUX 81~83

有些 AUX 功能可以用来修改或激活 CPU 口令。您可以在在线通讯时使用 CPU 的这些特性，或者您也可以用装有 EEPROM 的手持编程器离线操作时使用此功能。您可以通过手持编程器修改程序或给程序口令保护。

- AUX 81——修改口令
- AUX 82——CPU 解锁
- AUX 83——CPU 上锁

AUX 81 修改口令

AUX 81 可以用来给 CPU 设置口令，这可以帮助您避免未授权的机器操作。 标准的口令必须使用 8 位数字 0～9。设置口令后要再清除它，只需把口令重新设成 8 个零。因为这是出厂口令。

DL350 还提供了由字母“A”和 7 个数字组成的多级口令。这种口令有别于标准口令的地方是，它允许操作界面访问并修改 R 寄存器的数据（比如置位）。但是，仍然不能对程序进行修改。一旦您设置了口令，别人就无法访问 CPU。用手持编程器给 CPU 设置口令有如下两种方法。

- 系统重新上电后 CPU 自动上锁（假定已设置口令）
- 您可以使用 AUX 82 和 AUX 83 来打开或关闭口令

同时您也可以通过 DirectSOFT 的 PLC 菜单下的 Password 子菜单来设置口令。该特性在 DirectSOFT 中的操作略有不同。当您设置了口令，关闭 DirectSOFT 后 CPU 会自动上锁。同样，系统重新上电后 CPU 也会自动上锁。



警告：在设置口令以前请确保您已牢记了口令。一旦 CPU 上锁后，您不能查看，更改或清除口令。如果您忘记了口令，您只能把 CPU 返回给我们来清除口令。



注意：D3-350 CPU 支持梯形图程序的多级口令保护。这种口令保护不会影响 PLC 和操作界面之间的通讯。要使用多级口令保护，只需把口令的第一个数字改成“A”，后面 7 个仍然是数字（例如：A1234567）。

AUX 82 CPU 解锁

AUX 82 可以用来给设置口令的 CPU 解锁。当使用 DirectSOFT 和设置口令的 CPU 进行通讯时，它会自动提示键入口令。

AUX 83 CPU 上锁

AUX 83 用来给 CPU 上锁。一旦 CPU 上锁以后，您必须输入口令才能访问 CPU。记住，对于 DirectSOFT 不需要此功能，因为在退出程序后 CPU 就自动上锁。

附录 B　错误代码

B

本章将介绍：

——错误代码表

DL305 错误代码	描述
E003 SOFTWARE TIME-OUT	如果程序的扫描时间超过了看门狗定时器的设定时间，将会发生这个错误。SP51 会 ON 并且错误代码会被装入 R7755。要解决这个问题，在 FOR NEXT 循环之间或子程序中添加 RSTWT 指令，或者用 AUX 55 来更改看门狗定时器的设定时间。
E041 CPU BATTERY LOW	CPU 电池电量低，应该及时更换。SP43 会 ON 并且错误代码会被装入 R7757。
E099 PROGRAM MEMORY EXCEEDED	如果编译过后的程序超过了 CPU RAM 的空间时将会出现这个错误。SP52 会 ON 并且错误代码会被装入 R7755。
E104 WRITE FAILED	写入 CPU 操作失败。先断电，取下 CPU，确保 EEPROM 没有被写保护。如果 EEPROM 没有被写保护，确保 EEPROM 安装正确。如果以上两点都没问题，则只能更换 CPU。
E151 BAD COMMAND	应用程序中出现奇偶校验错误。SP44 会 ON 并且错误代码会被装入 R7755。这个错误有可能是电气上的干扰产生的。清除程序后再重新写入程序。
E155 RAM FAILURE	系统的 RAM 中发生了求校验和错误。SP44 会 ON 并且错误代码会被装入 R7755。该问题的产生可能是由于电池电量低，电气干扰或 CPU RAM 损坏。清除寄存器并重新下载程序。确保所有的设备都接地良好。如果还有问题，则需要更换 CPU。
E202 MISSING I/O MODULE	某个 I/O 模块与 CPU 通讯有问题或根本没插在框架上。SP45 会 ON 并且错误代码会被装入 R7756。使用 AUX 42 来查看 I/O 诊断信息。
E210 POWER FUALT	给框架供电的主电源线供电有问题。
E250 COMMUNICATION FAILURE IN THE I/O CHAIN	本地 I/O 系统发生错误。问题可能是框架 I/O 总线或框架的直流电源。SP54 会 ON 并且错误代码会被装入 R7755。使用 AUX 42 来查看 I/O 诊断信息。
E252 NEW I/O CFG	这个错误发生在经过 CPU 自动配置检查后发现 I/O 配置改变时。您可以把模块放到它原来的位置或使用 AUX 45 来确认新的配置。SP47 会 ON 并且错误代码会被装入 R7755。
E262 I/O OUT OF RANGE	应用程序中出现了一个 I/O 超出范围的错误。纠正程序中的非法地址。SP54 会 ON 并且错误代码会被装入 R7755。

DL305 错误代码	描述
E312 HP COMM ERROR2	与 CPU 通讯时遇到了一个数据错误。清除错误并重新请求。如果问题还没解决，请依次检查连接电缆，更换手持编程器，需要的话更换 CPU。SP46 会 ON 并且错误代码会被装入 R7756。
E313 HP COMM ERROR3	与 CPU 通讯时遇到了一个地址错误。清除错误并重新请求。如果问题还没解决，请依次检查连接电缆，更换手持编程器，需要的话更换 CPU。SP46 会 ON 并且错误代码会被装入 R7756。
E316 HP COMM ERROR6	与 CPU 通讯时遇到了一个模式错误。如果问题还没解决，请尝试更换手持编程器，需要的话更换 CPU。SP46 会 ON 并且错误代码会被装入 R7756
E320 HP COMM TIME-OUT	CPU 未对手持编程器发送的请求进行应答。检查连接电缆是否工作正常。如果重新上电后还有问题，先更换 CPU，需要的话再更换手持编程器。
E321 COMM ERROR	与 CPU 通讯时遇到了一个数据错误。检查连接电缆是否工作正常。如果重新上电后还有问题，先更换 CPU，需要的话再更换手持编程器。
E4** NO PROGRAM	应用程序中遇到一个语法错误。一般都是在程序末尾忽略了 END。使用 AUX 21 来查看哪里出了错误。SP52 会 ON 并且错误代码会被装入 R7755。
E401 MISSING END STATEMENT	所有的应用程序都要用 END 来作结尾。请在程序中的适当位置添加 END。SP52 会 ON 并且错误代码会被装入 R7755。
E402 MISSING LBL	使用 GOTO,GTS,MOVMC 或 LDLBL 指令时缺少了对应的 LABEL。到编程手册上查阅这些指令的使用方法。SP52 会 ON 并且错误代码会被装入 R7755。
E403 MISSING RET	子程序结束时缺少 RET 指令。
E404 MISSING FOR	有了 NEXT 指令，必须有 FOR 指令。SP52 会 ON 并且错误代码会被装入 R7755。

DL305 错误代码	描述
E405 MISSING NEXT	有一个 FOR 指令必须有一个 NEXT 指令。SP52 会 ON 并且错误代码会被装入 R7755。
E406 MISSING IRT	中断子程序结尾缺少 IRT 指令。SP52 会 ON 并且错误代码会被装入 R7755。
E412 SBR/LBL>64	程序中的 SBR,LBL 和 DLBL 指令超过 64 个。程序中的 GTS 或 GOTO 指令数超过 128 个也会产生该错误。SP52 会 ON 并且错误代码会被装入 R7755。
E413 FOR/NEXT>64	程序中的 FOR/NEXT 循环个数超过 64 个。SP52 会 ON 并且错误代码会被装入 R7755。
E421 DUPLICATE STAGE REFERENCE	程序中有多个 SG 或 ISG 编号重复。每个级和初始级都要有一个独立的编号。
E422 DUPLICATE SBR/LBL REFERENCE	程序中有多个 SBR 或 LBL 编号重复。每个子程序或标签都要有一个独立的编号。
E423 NESTED LOOPS	DL350 CPU 不允许嵌套循环（在 FOR/NEXT 循环中再使用 FOR/NEXT 循环）。
E431 INVALID ISG/SG ADDDRESS	ISG 或 SG 不能作为子程序等放在 END 指令之后。SP52 会 ON 并且错误代码会被装入 R7755。
E432 INVALID JUMP (GOTO) ADDRESS	与 GOTO 指令对应的 LBL 不能放在 END 之后。SP52 会 ON 并且错误代码会被装入 R7755。
E433 INVALID SBR ADDRESS	SBR 必须放在 END 指令之后，不能放在主程序或中断子程序中。SP52 会 ON 并且错误代码会被装入 R7755。
E435 INVALID RT ADDRESS	RT 必须放在 END 指令之后，不能放在主程序或中断子程序中。SP52 会 ON 并且错误代码会被装入 R7755。

DL305 错误代码	描述
E436 INVALID INT ADDRESS	INT 必须放在 END 之后，不能放在主程序中。SP52 会 ON 并且错误代码会被装入 R7755。
E438 INVALID IRT ADDRESS	IRT 必须放在 END 之后，不能放在子程序中。SP52 会 ON 并且错误代码会被装入 R7755。
E440 INVALID DATA ADDRESS	DLBL 指令放在了主程序中，或 DLBL 指令的条件中含有输入接点。
E441 ACON/NCON	ACON 或 NCON 指令必须放在 END 之后。SP52 会 ON 并且错误代码会被装入 R7755。
E451 BAD MLS/MLR	用 MLS 来定义母线时，其编号必须是升序的方式。
E452 X AS OUTPUT	输入接点作为输出线圈。
E453 MISSING T/C	被使用的定时器或计数器接点缺少相应的触发条件。
E454 BAD TMRA	TMRA 指令缺少接点。
E455 BAD CNT	CNT 或 UDC 指令缺少接点。
E456 BAD SR	SR 指令缺少接点。
E461 STACK OVERFLOW	9 级以上的逻辑装入堆栈。检查 ORLD 或 ANDLD 指令的使用是否过多。
E462 STACK UNDERFOLW	不匹配的逻辑级被装入了堆栈。确认 ANDLD 和 ORLD 的数量是否和 LD 的数量相匹配。
E463 LOGIC ERROR	梯形图逻辑中缺少 LD 指令。
E464 MISSING CKT	梯形图逻辑的某一行没有正常结束。
E471 DUPLICATE COIL REFERENCE	某个 I/O 点重复使用 OUT 指令。
E472 DUPLICATE TMR REFERENCE	TMR 指令定义了相同的定时器。

DL305 错误代码	描述
E473 DUPLICATE CNT REFERENCE	CNT 指令使用了相同的计数器。
E480 INVALID CV ADDRESS	CV 指令放在了子程序或中断子程序中。CV 指令只能放在主程序中。
E481 CONFLICTING INSTRUCTION	合流的级之间存在指令。
E482 MAX.CV INSTRUCTION EXCEEDED	CV 指令数超过了 17。
E483 INVALID CVJMP ADDRESS	CVJMP 使用在子程序或中断子程序中。
E484 MISSING CV INSTRUCTION	CVJMP 指令后面缺少 CV 指令。CVJMP 指令必须紧跟着 CV 指令。
E485 NO CVJMP	CVJMP 指令必须放在 CV 和 SG,ISG,BLK,BEND,END 指令之间。
E486 INVALID BCALL ADDRESS	BCALL 使用在子程序或中断子程序中。BCALL 指令只能使用在主程序中。
E487 MISSING BLK INSTRUCTION	BCALL 指令后面缺少 BLK 指令。
E488 INVALID BLK ADDRESS	BLK 指令使用在子程序或中断子程序中。或者另一个 BLK 指令使用在 BCALL 和 BEND 指令之后。
E489 DUPLICATED CR REFERENCE	与 BLK 指令相关的中间继电器被当作输出使用在其他地方。

DL305 错误代码	描述
E490 MISSING SG INSTRUCTION	BLK 指令后缺少 SG 指令。
E491 INVALID ISG INSTRUCTION ADDRESS	BLK 和 BEND 指令中有一个 ISG 指令。
E492 INVALID BEND ADDRESS	BEND 指令使用在子程序或中断子程序之后。BEND 指令后面缺少 BLK 指令。
E493 MISSING REQUIRED INSTRUCTION	CV,SG,ISG,BLK,BEND 指令后面必须紧跟着 BEND 指令。
E494 MISSING BEND INSTRUCTION	BLK 指令后面缺少 BEND 指令。
E499 PRINT INSTRUCTION	非法使用 PRINT 指令。引用和（或）空间使用不当。
E501 BAD ENTRY	在手持编程器上进行了无效的按键操作。
E502 BAD ADDRESS	手持编程器上使用了无效的地址或地址超出范围。
E503 BAD COMMAND	在手持编程器上输入了无效的指令。
E504 BAD REF/VAL	指令中使用了无效的赋值和定义号。
E505 INVALID INSTRUCTION	手持编程器上输入了无效的指令。
E506 INVALID OPERATION	手持编程器检测到一个无效的操作。
E520 BAD OP-RUN	在 RUN 状态下，手持编程器检测到一个无效的操作。
E521 BAD OP-TRUN	在 TEST RUN 状态下，手持编程器检测到一个无效的操作。
E523 BAD OP-TPGM	在 TEST PROGRAM 状态下，手持编程器检测到一个无效的操作。
E524 BAD OP-PGM	在 PROGRAM 状态下，手持编程器检测到一个无效的操作。

DL305 错误代码	描述
E525 MODE SWITCH	当 CPU 的模式开关不在 TERM 位置时，手持编程器接受了一个操作。
E526 OFF LINE	手持编程器在 OFFLINE 模式。可以使用 MODE 键来切换到 ONLINE 模式。
E527 ON LINE	手持编程器在 ONLINE 模式。可以使用 MODE 键来切换到 OFFLINE 模式。
E528 CPU MODE	进行了 RUN TIME EDIT 时的无效操作。
E540 CPU LOCKED	CPU 设置了口令。使用 AUX 82 来输入口令。
E541 WRONG PASSWORD	口令错误。
E542 PASSWORD RESET	CPU 上电时发现口令错误并把口令复位到 00000000。可以通过 AUX 81 来重新输入口令。
E601 MEMORY FULL	CPU 存储空间已满。
E602 INSTRUCTION MISSING	使用查找功能后未找到相关指令。
E604 REFERENCE MISSING	使用查找功能后未找到相关定义号。
E610 BAD I/O TYPE	应用程序定义模块部正确。
E620 OUT OF MEMORY	错误的 LDLBL,MOV,或 MOVMC 命令。CPU 和手持编程器之间进行数据传输时，某个设备无法接收数据。
E621 EEPROM NOT BLANK	试图向无空间的 EEPROM 写入内容。擦除 EEPROM 中的内容并重试。
E622 NO HPP EEPROM	手持编程器中无 EEPROM。
E623 SYSTEM EEPROM	包含系统的 EEPROM 接收到一个功能请求。
E624 V-MEMORY ONLY	EEPROM 中只含有 V-MERMORY。
E625 PROGRAM ONLY	EEPROM 中只含有程序。

DL305 错误代码	描述
E627 BAD WRITE	对写保护的 EEPROM 进行写操作。检查写保护跳线或更换 EEPROM。
E640 COMPARE ERROR	比较 EEPROM 和 CPU 时发现错误。
E650 HPP SYSTEM ERROR	手持编程器上遇到了一个系统错误。重新给手持编程器上电，如果还有问题，请尝试更换编程器。
E651 HPP ROM ERROR	手持编程器上的 ROM 发生了一个错误。重新给手持编程器上电，如果还有问题，请尝试更换编程器。
E652 HPP RAM ERROR	手持编程器上的 RAM 发生了一个错误。重新给手持编程器上电，如果还有问题，请尝试更换编程器。

附录 C　指令执行时间

C

本章将介绍：
——简介
——顺序指令
——比较指令
——直接输入输出指令
——定时器，计数器，移位寄存器指令
——累加器数据指令
——逻辑运算指令
——算术运算指令
——位指令
——数字转换指令
——表格指令
——CPU 控制指令
——程序控制指令
——中断指令
——网络指令
——消息指令
——级指令

简介

本附录提供了 DL350 CPU 的各种指令的执行时间。您会发现许多指令的执行时间和它所使用的操作数类型有很大关系。例如，一些使用 R 寄存器的指令将进一步分为以下两种情况：

- 数据寄存器
- 位寄存器

R 寄存器的数据寄存器

一些 R 寄存器被作为数据寄存器。例如，一些存放定时器或计数器经过值的 R 寄存器，或者常规的 R 寄存器都可以当作 R 寄存器的数据寄存器。您可以改变这些寄存器中的某一位。以下区域被视作数据寄存器。

数据寄存器	DL350
定时器经过值	R0~R377
计数器经过值	R1000~R1177
用户数据字	R1400~R7377 R10000~R17777

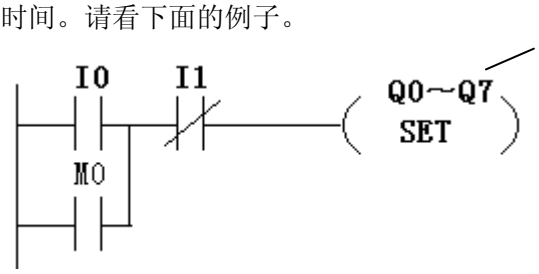
R 寄存器的位寄存器

您应该还记得像一些I,Q,M等离散点都是自动放在相应的R寄存器中的。以下是这些离散点在存储器中的位置。

位寄存器	DL350
输入（I）	R40400~R40437
输出（Q）	R40500~R40537
中间继电器（M）	R40600~R40677
定时器状态位	R41100~R41117
计数器状态位	R41040~R41147
级	R41000~R41077

怎样读懂表格

有些指令支持多种操作数，所以表格中的执行时间取决于操作数的类型及其数量。例如，SET 指令可以置位某个离散点也可以置位若干个离散点。如果您仔细查看了表格，您会发现两种情况的对应的指令执行时间。请看下面的例子。



SET	1#：I,Q,M,S 2#：I,Q,M,S（N 点）	17.4μ s 12.0μ s +5.4μ s *N
RST	1#：I,Q,M,S 2#：I,Q,M,S（N 点）	19.5μ s 10.5μ s+5.2μ s*N

执行时间取决于操作数的类型及其数量

顺序指令

顺序指令			DL350	
指令	功能	操作数	执行	不执行
STR	逻辑运算开始常开接点	I,Q,M,T,C,S,SP	0.74μ s	0.74μ s
STRN	逻辑运算开始常闭接点	I,Q,M,T,C,S,SP	0.68μ s	0.74μ s
OR	逻辑或运算常开接点	I,Q,M,T,C,S,SP	0.56μ s	0.56μ s
ORN	逻辑或运算常闭接点	I,Q,M,T,C,S,SP	0.6μ s	0.6μ s
AND	逻辑与运算常开接点	I,Q,M,T,C,S,SP	0.46μ s	0.46μ s
ANDN	逻辑与运算常闭接点	I,Q,M,T,C,S,SP	0.56μ s	0.56μ s
ANDSTR	逻辑组串联	无	0.4μ s	0.4μ s
ORSTR	逻辑组并联	无	0.4μ s	0.4μ s
OUT	线圈接通	I,Q,M	2.0μ s	2.0μ s
OUTH	线圈接通	I,Q,M	1.1μ s	1.1μ s
OROUT	输出线圈相同时使用	I,Q,M	2.4μ s	2.4μ s
PD	上升沿微分，一次扫描输出	I,Q,M	16.6μ s	16.6μ s
SET	线圈接通保持	1#：I,Q,M,S 2#：I,Q,M,S （N 点）	10.6μ s 11.4μ s +0.9 μ s *N	1.1μ s 1.1μ s
RST	线圈断开	1#：I,Q,M,S 2#：I,Q,M,S （N 点）	10.6μ s 11.4μ s +0.9 μ s *N	1.1μ s 1.1μ s
		1#：T,C 2#：T,C （N 点）	10.6μ s 11.4μ s +0.9 μ s *N	1.1μ s 1.1μ s

比较指令

比较指令				DL350	
指令	功能	操作数		执行	不执行
STRE	逻辑运算开始等于接点	1st	2nd		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
STRNE	逻辑运算开始不等于接点	1st	2nd		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		

比较指令			DL350	
指令	功能	操作数	执行	不执行
ORE	逻辑或运算等于接点	1st 2nd		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（数据） R：数据寄存器	32.6μ s	32.6μ s
			60.7μ s	60.7μ s
		R：位寄存器		
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（位） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）		
ORNE	逻辑或不等于接点	1st 2nd		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（数据） R：数据寄存器	32.6μ s	32.6μ s
			60.7μ s	60.7μ s
		R：位寄存器		
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（位） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）		

比较指令			DL350	
指令	功能	操作数	执行	不执行
ANDE	逻辑与运算等于接点	1st 2nd		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（数据） R：数据寄存器	32.6μ s	32.6μ s
			60.7μ s	60.7μ s
		R：位寄存器		
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（位） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）		
ANDNE	逻辑与运算不等于接点	1st 2nd		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（数据） R：数据寄存器	32.6μ s	32.6μ s
			60.7μ s	60.7μ s
		R：位寄存器		
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（位） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）		

比较指令				DL350	
指令	功能	操作数		执行	不执行
STR	逻辑运算开始 大于等于常开 接点	1st T,C	2nd R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
STRN	逻辑运算开始 大于等于常闭 接点	1st T,C	2nd R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		

比较指令			DL350	
指令	功能	操作数	执行	不执行
OR	逻辑或运算大于等于常开接点	1st T,C 2nd R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器		
		K：常数	5.5μ s	5.5μ s
		P：间接（数据）	35.9μ s	35.9μ s
		P：间接（位）		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接(数据） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）	35.6μ s	35.6μ s
		P：间接（位） R：数据寄存器		
		R：位寄存器	32.6μ s	32.6μ s
		K：常数	60.7μ s	60.7μ s
		P：间接（数据）		
		P：间接（位）		
ORN	逻辑或运算大于等于常闭接点	1st T,C 2nd R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器		
		K：常数	5.5μ s	5.5μ s
		P：间接（数据）	35.9μ s	35.9μ s
		P：间接（位）		
		R：数据寄存器 R：数据寄存器	8.7μ s	8.7μ s
		R：位寄存器	5.5μ s	5.5μ s
		K：常数	35.9μ s	35.9μ s
		P：间接（数据）		
		P：间接（位）	8.7μ s	8.7μ s
		R：位寄存器 R：数据寄存器	5.5μ s	5.5μ s
		R：位寄存器	35.9μ s	35.9μ s
		K：常数		
		P：间接（数据）	35.6μ s	35.6μ s
		P：间接（位）		
		P：间接（数据） R：数据寄存器	32.6μ s	32.6μ s
		R：位寄存器	60.7μ s	60.7μ s
		K：常数		
		P：间接（数据）		
		P：间接（位）	35.6μ s	35.6μ s
		P：间接（位） R：数据寄存器		
		R：位寄存器	32.6μ s	32.6μ s
		K：常数	60.7μ s	60.7μ s
		P：间接（数据）		
		P：间接（位）		

比较指令				DL350	
指令	功能	操作数		执行	不执行
AND	逻辑与运算大于等于常开接点	1st	2nd		
		T,C	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
ANDN	逻辑与运算大于等于常闭接点	1st	2nd		
		T,C	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：数据寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		R：位寄存器	R：数据寄存器	8.7μ s	8.7μ s
			R：位寄存器		
			K：常数	5.5μ s	5.5μ s
			P：间接（数据）	35.9μ s	35.9μ s
			P：间接（位）		
		P：间接（数据）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		
		P：间接（位）	R：数据寄存器	35.6μ s	35.6μ s
			R：位寄存器		
			K：常数	32.6μ s	32.6μ s
			P：间接（数据）	60.7μ s	60.7μ s
			P：间接（位）		

直接输入输出指令

直接输入输出指令			DL350	
指令	功能	操作数	执行	不执行
STRI	逻辑开始常开接点	I	78.6μ s	78.6μ s
STRNI	逻辑开始常闭接点	I	78.6μ s	78.6μ s
ORI	逻辑或运算常开接点	I	78.6μ s	78.6μ s
ORNI	逻辑或运算常闭接点	I	78.6μ s	78.6μ s
ANDI	逻辑与运算常开接点	I	78.6μ s	78.6μ s
ANDNI	逻辑与运算常闭接点	I	78.6μ s	78.6μ s
OUTI	线圈接通	Q	91.0μ s	91.0μ s
OROUTI	输出线圈重复时使用	Q	94.0μ s	94.0μ s
SETI	线圈置位	1 st#：Q 2 nd#：Q（N 点）	87.6μ s 97.5μ s +16.25 μ s *N	1.1μ s 1.1μ s
RSTI	线圈复位	1 st#：Q 2 nd#：Q（N 点）	87.6μ s 97.5μ s +16.25 μ s *N	1.1μ s

定时器，计数器，移位寄存器指令

定时器，计数器，移位寄存器指令			DL350	
指令	功能	操作数	执行	不执行
TMR	0.1 秒定时器	1 st 2 nd T： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	38.6μ s 23.0μ s 54.3μ s	24.6μ s 24.6μ s 52.0μ s
TMRF	0.01 秒定时器	1 st 2 nd T： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	61.2μ s 57.6μ s 90.4μ s	23.0μ s 19.4μ s 37.5μ s
TMRA	0.1 秒累加定时器	1 st 2 nd T： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	58.2μ s 53.6μ s 90.4μ s	27.1μ s 22.4μ s 59.2μ s
TMRAF	0.01 秒累加定时器	1 st 2 nd T： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	64.5μ s 59.9μ s 96.7μ s	27.6μ s 22.4μ s 59.2μ s
CNT	计数器（带复位端）	1 st 2 nd C： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	36.1μ s 32.5μ s 97.1μ s	24.6μ s 21.0μ s 56.8μ s
SGCNT	计数器（不带复位）	1 st 2 nd C： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	35.2μ s 33.7μ s 67.4μ s	27.7μ s 27.1μ s 54.3μ s
UDC	加减计数	1 st 2 nd C： R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	47.4μ s 42.7μ s 81.7μ s	40.0μ s 35.3μ s 72.1μ s
SR	移位寄存器	M（N 点）	17.8 μ s+1.0 μ s*N	12.6μ s

累加器数据指令

累加器/堆栈及数据输出指令			DL350	
指令	功能	操作数	执行	不执行
LD	读入对象（16 位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	47.4μ s 42.7μ s 81.7μ s	47.4μ s 42.7μ s 81.7μ s
LDD	读入对象（32 位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	47.4μ s 42.7μ s 81.7μ s	47.4μ s 42.7μ s 81.7μ s
LDF	读入任意位（1～32 位）	1 st 2 nd I,Q,M, K：常数 S,T,C,SP	10.5μ s +3.45 μ s *N	1.4μ s
LDA	读入对象寄存器号	O：表示地址的八进制常数	10.4μ s	1.1μ s
LDSX	读入 4 位常数	K：常数	14.6μ s	1.5μ s
OUT	累加器的内容写入目标（16 位）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	10.7μ s 41.9μ s	1.1μ s
OUTD	累加器的内容写入目标（32 位）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	11.7μ s 42.6μ s	1.1μ s
OUTF	累加器内容写入任意位（1～32 位）	1 st 2 nd I,Q,M, K：常数	43.8μ s+6.2μ s*N	1.1μ s
POP	数据堆栈弹出	无	7.8μ s	1.0μ s

逻辑运算指令

逻辑运算（累加器）指令			DL350	
指令	功能	操作数	执行	不执行
AND	逻辑与运算常开接点	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	9.1μ s 39.8μ s	1.1μ s 1.1μ s
ANDD	与运算（32位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	10.2μ s 6.5μ s 40.9μ s	1.1μ s 1.1μ s 1.1μ s
OR	逻辑与运算常开接点	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	9.3μ s 40.2μ s	1.1μ s 1.1μ s
ORD	或运算（32位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	10.4μ s 6.7μ s 41.1μ s	1.1μ s 1.1μ s 1.1μ s
XOR	异或运算（16位）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	9.2μ s 40.0μ s	1.1μ s 1.1μ s
XORD	异或运算（32位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	10.3μ s 6.2μ s 41.0μ s	1.1μ s 1.1μ s 1.1μ s
CMP	比较（16 位）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	10.8μ s 41.5μ s	1.1μ s 1.1μ s
CMPRD	比较（32 位）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	11.4μ s 7.7μ s 42.1μ s	1.2μ s 1.2μ s 1.2μ s
CMPS	堆栈比较	无	—	—

算术指令

算术指令（累加器）			DL350	
指令	功能	操作数	执行	不执行
ADD	4 位加法（BCD）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	93.3μ s 129.8μ s	1.2μ s 1.1μ s
ADDD	8 位加法（BCD）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	99.2μ s 80.6μ s 129.8μ s	1.2μ s 1.2μ s 1.2μ s
SUB	4 位减法（BCD）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	98.1μ s 121.9μ s	1.1μ s 1.1μ s
SUBD	8 位减法（BCD）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	98.2μ s 78.6μ s 127.8μ s	1.1μ s 1.1μ s 1.1μ s
MUL	4 位乘法（BCD）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	341.1μ s 367.8μ s 371.8μ s	1.1μ s 1.1μ s 1.1μ s
MULD	8 位乘法（BCD）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	1075.8μ s 1106.5μ s	1.1μ s 1.1μ s
DIV	4 位除法（BCD）	R：数据寄存器 R：位寄存器 K：常数 P：间接（数据） P：间接（位）	466.6μ s 492.8μ s 538.2μ s	1.1μ s 1.1μ s 1.1μ s
DIVD	8 位除法（BCD）	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	510.6μ s 501.1μ s	1.1μ s 1.1μ s
INCB	BIN 增 1	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	15.2μ s 45.9μ s	1.1μ s 1.1μ s
DECB	BIN 减 1	R：数据寄存器 R：位寄存器 P：间接（数据） P：间接（位）	15.2μ s 45.2μ s	1.1μ s 1.1μ s

位指令

位指令（累加器）			DL350	
指令	功能	操作数	执行	不执行
SHFR	右移	R：数据寄存器（N 位） R：位寄存器（N 位） K：常数（N 位）	9.8μ s+0.2μ s*N 7.9μ s+0.2μ s*N	1.2μ s
SHFL	左移	R：数据寄存器（N 位） R：位寄存器（N 位） K：常数（N 位）	9.8μ s+0.2μ s*N 7.9μ s+0.2μ s*N	1.2μ s
ROTR	循环右移	R：数据寄存器（N 位） R：位寄存器（N 位） K：常数（N 位）	15.7μ s 12.3μ s	1.2μ s
ROTL	循环左移	R：数据寄存器（N 位） R：位寄存器（N 位） K：常数（N 位）	15.7μ s 12.3μ s	1.2μ s
ENCO	编码	无	40.3μ s	1.0μ s
DECO	译码	无	6.5μ s	1.0μ s

数字转换指令

数字转换指令（累加器）			DL350	
指令	功能	操作数	执行	不执行
BIN	BIN 码变换	无	128.4μ s	1.0μ s
BCD	BCD 码变换	无	122.0μ s	1.0μ s
INV	取反	无	2.9μ s	1.0μ s
BCDCPL	十进制补码变换	无	74.5μ s	1.0μ s
ATH	ACCII→HEXA 代码变换	R	29.2μ s	1.0μ s
HTA	HEXA→ACCII 代码变换	R	29.2μ s	1.0μ s
SEG	7 段译码	无	12.6μ s	1.0μ s
GRAY	格雷码→BDC 码	无	142.0μ s	1.0μ s
SFLDGT	位替换指令	无	26.6μ s	1.0μ s

表格指令

表格指令			DL350	
指令	功能	操作数	执行	不执行
MOV	数据块传送	R：数据到 R：数据 R：位到 R：数据 R：数据到 R：位 R：位到 R：位 N＝传送字数	63μ s＋16×N	1.2μ s
MOVMC	E^2 和数据寄存器之间的数据传送	R：数据到 E^2 R：位到 E^2 E^2 到 R：数据 E^2 到 R：位	50μ s＋15×N	1.2μ s
LDLBL	数据标号地址读出	K	7.4μ s	1.5μ s

CPU 控制指令

CPU 控制指令			DL350	
指令	功能	操作数	执行	不执行
NOP	空操作	无	0.6μ s	0.6μ s
END	程序结束	无	14.7μ s	14.7μ s
STOP	STOP 方式停止	无	4.1μ s	1.0μ s
RSTWT	监控定时器复位	无	5.4μ s	1.0μ s
NOT	接点的逻辑组合取非	无	1.0μ s	1.0μ s

程序控制指令

程序控制指令			DL350	
指令	功能	操作数	执行	不执行
GOTO	程序跳转	K	5.0μ s	4.9μ s
LBL	跳转目的标号	K	0.6μ s	0.6μ s
FOR	循环指令（起点）	R ,K	110μ s	7.9μ s
NEXT	循环指令（终点）	无	48.4μ s	0μ s
GTS	调用子程序	K	12.5μ s	6.3μ s
SBR	子程序编号	K	0.5μ s	0μ s
RT	子程序结束	无	11.4μ s	11.4μ s
MLS	母线开始	K(1~7)	4.2μ s	4.2μ s
MLR	母线返回	K(0~7)	4.0μ s	4.0μ s

中断指令

中断指令			DL350	
指令	功能	操作数	执行	不执行
ENI	中断许可	无	45.8μ s	1.1μ s
DISI	中断禁止	无	5.7μ s	1.1μ s
INT	中断	0（0～7）	0μ s	0μ s
IRT	中断返回	无	1.5μ s	—
IRTC	中断返回条件	无	0.5μ s	0.5μ s

网络指令

网络指令			DL350	
指令	功能	操作数	执行	不执行
RX	读出，通过通讯模块读出其他 PC 的数据，程序，方式等	I,Q,M,T,C,SP,S R：数据寄存器 R：位寄存器	2024.1μ s	1.4μ s
WX	写入，通过通讯模块向其他 PC 写入数据，程序，方式等	I,Q,M,T,C,SP,S R：数据寄存器 R：位寄存器	2224.1μ s	1.4μ s

消息指令

消息指令			DL350	
指令		操作数	执行	不执行
FAULT	外部诊断码信息表示	R：数据寄存器 R：位寄存器 K：常数	108.9μ s 108.9μ s 96.2μ s	1.4μ s 1.4μ s 1.4μ s
DLBL	数据区标号	K	0μ s	0μ s
NCON	数值数据登陆	K	0μ s	0μ s
ACON	ASCII 数据登陆	A	0μ s	0μ s
PRINT	打印信息		104.0μ s	1.4μ s

级式指令

级式指令			DL350	
指令	功能	操作数	执行	不执行
ISG	初始级登记	S	24.3μ s	21.5μ s
SG	级登记	S	24.3μ s	21.5μ s
JMP	条件成立时级转移	S	24.4μ s	4.3μ s
NJMP	条件不成立时级转移	S	24.4μ s	4.6μ s
CV	级合流登记	S	13.9μ s	13.9μ s
CVJMP	级合流转移	S (N 级，1～16)	12.6μ s	12.6μ s
BCALL	级组起动	C	17.1μ s	17.1μ s
BLK	级组开始指令	C	22.1μ s	22.6μ s
BEND	级组结束指令	无	8.7μ s	0μ s

时钟/日历指令

时钟/日历指令			DL350	
指令	功能	操作数	执行	不执行
DATE	设定日期	R：数据寄存器 R：位寄存器	21.3μ s 21.3μ s	1.9μ s 1.9μ s
TIME	设定时间	R：数据寄存器 R：位寄存器	21.3μ s 21.3μ s	1.9μ s 1.9μ s

鼓型指令

鼓型指令			DL350	
指令	功能	操作数	执行	不执行
DRUM	时间驱动型凸轮	C	340.0μ s	62.6μ s
EDRUM	时间与事件型凸轮	C	243.0μ s	100.0μ s
MDRUM	带掩膜时间与事件驱动型凸轮（离散点输出）	C	206.0μ s	142.0μ s
MDRMW	带掩膜时间与事件驱动型凸轮（字输出）	C	150.0μ s	94.0μ s

指令对照表：

美国习惯	中国习惯
顺序指令	
STR	LD
STRN	LDN
OR	OR
ORN	ORN
AND	AND
ANDN	ANDN
ANDSTR	ANDLD
ORSTR	ORLD
OUT	OUT
OUTH	
OROUT	OUT
PD	PD
SET	SET
RST	RST
比较指令	
STRE	LDEQ
STRNE	LDNEQ
ORE	OREQ
ORNE	ORNEQ
ANDE	ANDEQ
ANDNE	ANDNEQ
STR	LDGE
STRN	LDNGE
OR	ORGE
ORN	ORNGE
AND	ANDGE
ANDN	ANDNGE

美国习惯	中国习惯
直接输入输出指令	
STRI	LDDI
STRNI	LDNDI
ORI	ORDI
ORNI	ORNDI
ANDI	ANDDI
ANDNI	ANDNDI
OUTI	OUTDI
OROUTI	OUTDI
SETI	SETDI
RSTI	RSTDI
定时器，计数器，移位寄存器指令	
TMR	TMR
TMRF	HTMR
TMRA	ATMR
TMRAF	AHTMR
CNT	CNT
SGCNT	GCNT
UDC	UDCNT
SR	SR
累加器/堆栈及数据输出指令	
LD	LDW
LDD	LDD
LDF	LDF
LDA	LDR
LDSX	LDS
OUT	OUTW
OUTD	OUTD
OUTF	OUTF
POP	POP

美国习惯	中国习惯
逻辑运算（累加器）指令	
AND	ANDW
ANDD	ANDD
OR	ORW
ORD	ORD
XOR	XORW
XORD	XORD
CMP	CMPR
CMPD	CMPRD
CMPS	
算术指令	
ADD	ADD
ADDD	ADDD
SUB	SUB
SUBD	SUBD
MUL	MUL
MULD	MULD
DIV	DIV
DIVD	DIVD
INCB	BINC
DECB	BDEC
位指令	
SHFR	SHFR
SHFL	SHFL
ROTR	ROTR
ROTL	ROTL
ENCO	ENCO
DECO	DECO

美国习惯	中国习惯
数字转换指令	
BIN	BIN
BCD	BCD
INV	INV
BCDCPL	BCDCPL
ATH	ATH
HTA	HTA
SEG	SEG
GRAY	GRAY
SFLDGT	SFLDGT
表格指令	
MOV	MOVE
MOVMC	MOVMC
LDLBL	LDLBL
CPU 控制指令	
NOP	NOP
END	END
STOP	STOP
RSTWT	WDOGR
NOT	NOT
程序控制指令	
GOTO	GOTO
LBL	GLBL
FOR	FOR
NEXT	NEXT
GTS	CAL
SBR	CLBL
RT	CEND
MLS	MLS
MLR	MLR

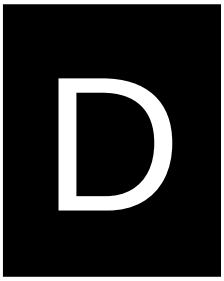
定义号对照表：

美国习惯	中国习惯
X	I
Y	Q
C	M
T	T
CT	C
S	S
P	P
SP	SP
V	R
K	K

美国习惯	中国习惯
中断指令	
ENI	INE
DISI	INH
INT	ILBL
IRT	RETI
IRTC	IEND
网络指令	
RX	RX
WX	WX
消息指令	
FAULT	FALT
DLBL	DLBL
NCON	NCON
ACON	ACON
PRINT	PRINT
级式指令	
ISG	ISG
SG	SG
JMP	JMP
NJMP	NJMP
CV	CV
CVJMP	CVJMP
BCALL	BREQ
BLK	BLK
BEND	BEND

美国习惯	中国习惯
时钟/日历指令	
DATE	DATE
TIME	TIME
鼓型指令	
DRUM	DRUM
EDRUM	EDRUM
MDRUM	MDRUM
MDRMW	MDRMW

附录 D　特殊继电器



本附录将介绍：

——DL350 CPU 的特殊继电器

DL350 CPU 特殊继电器

上电与实时继电器

SP0	第一扫描周期	上电运行或程序由编程方式改变为运行时的第一个扫描周期为 ON。
SP1	常 ON	每个扫描周期都接通的接点
SP2	常 OFF	每个扫描周期都断开的接点
SP3	1 分钟时钟	30 秒 ON，30 秒 OFF。
SP4	1 秒时钟	0.5 秒 ON，0.5 秒 OFF。
SP5	100ms 时钟	50ms 为 ON，50ms 为 OFF。
SP6	50ms 时钟	25ms 为 ON，25ms 为 OFF。
SP7	扫描时钟	一次扫描周期隔一次扫描周期 ON。

CPU 状态继电器

SP11	强制 RUN 方式	切换开关在 RUN 位置时 ON
SP12	TERM-RUN 方式	CPU 开关在 TERM 位置，CPU 处于 RUN 方式
SP13	TEST-RUN 方式	CPU 开关在 TERM 位置，CPU 处于 TEST-RUN 方式
SP14	TEST-HOLD 方式	CPU 开关在 TERM 位置，CPU 处于 TEST-HOLD 方式
SP15	TEST-PGM 方式	CPU 开关在 TERM 位置，CPU 处于 TEST-PGM 方式
SP16	TERM PGM 方式	CPU 开关在 TERM 位置，CPU 处于 PGM 方式
SP17	强制 STOP 继电器	切换开关在 STOP 位置
SP20	强制 STOP 方式	STOP 指令执行
SP21	BREAK 继电器	BREAK 指令执行，CPU 改变为 RUN 时 OFF
SP22	中断允许	ENI 指令允许中断时 ON
SP25	CPU 无电池方式	CPU 无电池方式时 ON

系统监视继电器

SP40	重度错误	重度错误例如 I/O 通讯异常时 ON
SP41	警告	轻度错误例如电池电压低时 ON
SP43	电池电压低	CPU 电池电压低时 ON
SP44	保留	
SP45	保留	
SP46	通讯错误	CPU 任何通讯口发生错误时 ON
SP47	I/O 配置错误	I/O 配置错误时 ON。CPU 上电时 I/O 配置检查必须允许。
SP50	FALT 指令	FALT 指令执行
SP51	监控定时器超时	CPU 的监控定时器超时
SP52	语法错误	CPU 运行或语法检查时语法错误。R7755 包含错误代码。
SP53	逻辑错误	CPU 不能运算
SP54	智能 I/O 错误	智能模块通讯错误

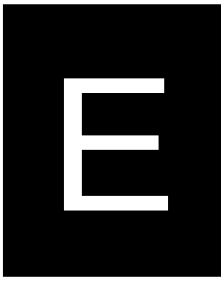
累加器状态继电器

SP60	小于	累加器的值小于指令值
SP61	等于	累加器的值等于指令值
SP62	大于	累加器的值大于指令值
SP63	零	指令结果为 0（累加器）
SP64	半借位	16bit 减指令结果发生借位
SP65	借位	32bit 减指令结果发生借位
SP66	半进位	16bit 加指令结果发生进位
SP67	进位	32bit 加指令结果发生进位
SP70	符号	累加器的值为负
SP71	无效的八进制	间接寻址指针（P）指定的 R 寄存器无效
SP72	无效的实数	累加器无效的实数
SP73	溢出	带符号数据运算时累加器发生溢出
SP74	下溢出	累加器浮点数下溢出（太接近 0.0）
SP75	数据错误	BCD 运算时，运算的数据不是 BCD 数时 ON
SP76	读零	指令读入累加器的数据为零

通讯监视继电器

SP116	CPU 通讯	当 PORT2 正在通讯时 ON
SP117	通讯口错误 2	当 PORT2 遇到通讯错误时 ON
SP120	0 号槽模块忙	当 0 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP121	0 号槽通讯错误	本地框架 0 号槽中的通讯模块发生错误时 ON。
SP122	1 号槽模块忙	当本地框架的 1 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP123	1 号槽通讯错误	本地框架 1 号槽中的通讯模块发生错误时 ON。
SP124	2 号槽模块忙	当本地框架的 2 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP125	2 号槽通讯错误	本地框架 2 号槽中的通讯模块发生错误时 ON。
SP126	3 号槽模块忙	当本地框架的 3 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP127	3 号槽通讯错误	本地框架 3 号槽中的通讯模块发生错误时 ON。
SP130	4 号槽模块忙	当本地框架的 4 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP131	4 号槽通讯错误	本地框架 4 号槽中的通讯模块发生错误时 ON。
SP132	5 号槽模块忙	当本地框架的 5 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP133	5 号槽通讯错误	本地框架 5 号槽中的通讯模块发生错误时 ON。
SP134	6 号槽模块忙	当本地框架的 6 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP135	6 号槽通讯错误	本地框架 6 号槽中的通讯模块发生错误时 ON。
SP136	7 号槽模块忙	当本地框架的 7 号槽中的通讯模块发送或接收数据时 ON。在使用 RX 或 WX 指令时必须要用该特殊继电器来避免在通讯模块忙时执行 RX 或 WX 指令。
SP137	7 号槽通讯错误	本地框架 7 号槽中的通讯模块发生错误时 ON。

附录 E　DL305 产品重量



本章将介绍：
——产品重量一览

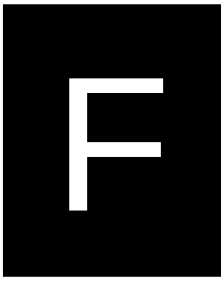
产品重量一览

CPU	重量
D3-330	178g
D3-330P	178g
D3-340	146g
D3-350	140g
特殊 CPU	
F3-OMUX-1	182g
F3-OMUX-2	182g
F3-PMUX	104g
F3-RTU	190g
框架	
D3-05B-1	1050g
D3-05BDC-1	1050g
D3-08-1	1250g
D3-10B-1	1450g
D3-05B	964g
D3-05BDC	964g
D3-08B	1253g
D3-10B	1432g
直流输入模块	
D3-08ND2	120
D3-16ND2-1	180g
D3-16ND2-2	150g
D3-16ND2F	180g
F3-16ND3F	153g
交流输入模块	
D3-08NA-1	140g
D3-08NA-2	140g
D3-16NA	180g
交流/直流输入模块	
D3-08NE3	120g
D3-16NE3	170g

直流输出模块	重量
D3-08TD1	120g
D3-08TD2	120g
D3-16TD1-1	160g
D3-16TD1-2	160g
D3-16TD2	210g
交流输出模块	
D3-04TAS	180g
D3-08TAS	
D3-08TA-1	210g
D3-08TA-2	180g
F3-16TA-1	176g
D3-16TA-2	210g
继电器输出模块	
D3-08TR	200g
F3-08TRS-1	252g
F3-08TRS-2	255g
D3-16TR	248g
模拟量模块	
D3-04AD	200g
F3-04ADS	195g
F3-08AD	154g
F3-08TEMP	147g
F3-08THM-n	170g
F3-16AD	152g
D3-02DA	200g
F3-04DA-1	180g
F3-04DA-1	180g
F3-04DAS	200g

通讯和联网	重量
D3-232-DCU	427g
D3-422-DCU	419g
ASCII 模块	
F3-AB128-R	146g
F3-128-T	175g
F3-AB128	154g
特殊模块	
D3-08SIM	85g
D3-HSC	147g
D3-PWU	368g
D3-FILL	30g
编程器	
D3-HP	202g
D3-HPP	204g
D2-HPP	220g

附录 F　I/O 寻址的常规方法



本章将介绍：

——了解常规的 I/O 分配

——常规框架的一般规格

——本地和扩展 I/O 系统

——框架上跳线开关的使用

——I/O 配置的例子

了解常规的 I/O 分配

本章介绍了在常规框架上安装 DL350 CPU 或在混合系统中安装 DL350 CPU 时所必须了解的一些信息。DL350 的使用有两种方案，即 16 位或 8 位寻址。第四章中提供了有关 xxxx－1 型框架的信息以及 16 位寻址的方案。

DL305 I/O 配置历史

多年来，DL305 系列的产品性能得到了多次的改进。 改进后，大部分都能和以前的产品向兼容。但有些改进，比如增加 I/O 点数、支持 16 点模块，会影响到 I/O 分配系统。为了帮您了解编号方案，以下提供了一些信息。

- 当 16 点模块插在 8 点模块的位置的时，其各个点的编号并不是连续的。这是为了与 8 点的系统相兼容。这意味着这 16 点被分为编号连续的两组，比如 000～007 和 100～107。
- 当原来的 I/O 点数从原来的最多 112 点增加到最多 176 点（DL330/DL330P CPU）或最多 184 点（DL340/DL350 CPU）时，大部分新的 I/O 地址并不是接在 112 点后面。这意味这中间有一个大的跳跃。

8 进制定义号

常规的 DL305 I/O 是以 8 进制来分配的。8 进制的编号系统中部含有数字 8 和 9。以下是两种编号方式的比较。

8 进制	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**20**	**21**	**22**	**23**	**24**	**…**
10 进制	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**…**

固定的 I/O 定义号

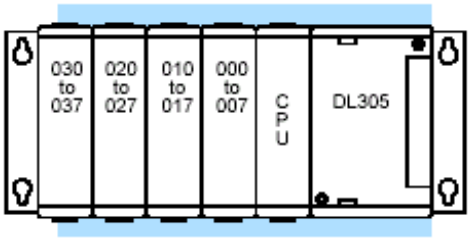
DL305 框架的 I/O 编号是固定的，您无法改变某个点的 I/O 地址，因为系统给每个插槽都定义了地址。其 I/O 编号的范围从 0～177 和 700～767。每个插槽的 I/O 编号取决于以下两点。

1. 框架的配置，即使用的是哪一种框架及是否使用扩展框架。
2. 每个模块 I/O 点数及模块在框架中的安放位置。

I/O 定义号向导

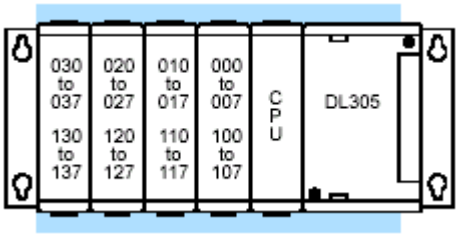
I/O 编号从 000 开始，CPU 旁边的模块被称作第一块模块。每个模块都依次占用 8 个点。对于 8 点模块，其 I/O 地址是 8 个连续的地址。而对 16 点模块，其 I/O 地址分为连续的两组。其中第一组的地址和 8 点模块是一样的，第二组的地址是在第一组的地址上加 100 得到的。

5 槽框架使用 8 点模块



槽号：3－ 2－ 1－ 0

5 槽框架使用 16 点模块



槽号：3－ 2 －1 －0

模块的占用的点数如下

直流输入模块	
D3-08ND2	8
D3-08ND2-1	16
D3-16ND2-2	16
D3-16ND2F	16
F3-16ND3	16
交流输入模块	
D3-08NA-1	8
D3-08NA-2	8
D3-16NA	16
AC/DC 输入模块	
D3-08NE3	8
D3-16NE3	16

直流输出模块	
D3-08TD1	8
D3-08TD2	8
D3-16TD1-1	16
D3-16TD1-2	16
D3-16TD2	16
交流输出模块	
D3-04TAS	8*
F3-08TAS	8
D3-08TA-1	8
D3-08TA-2	8
F3-16TA-1	16
D3-16TA-2	16

继电器输出模块	
D3-08TR1	8
F3-08TRS-1	8
F3-08TRS-2	8
D3-16TR	16
模拟量模块	
D3-04AD	16
F3-04ADS	16
F3-08AD	16
F3-08TEMP	16
F3-08THM-n	16
F3-16AD	16
D3-02DA	16

模拟量模块（续）	
F3-04DA-1	16
F3-04DA-2	16
F3-04DAS	16
ASCII 模块	
F3-AB128R	16
F3-AB128T	16
F3-AB128	16
F3-AB64	16
特殊模块	
D3-08SIM	8
D3-HSC	16

*这是一个 4 点模块，但每个模块至少占用 8 个点。

I/O 模块安装规则

对于模块应该放在哪一个插槽，这是有一定限制的。有些模块必须安放在特定位置，这有时会影响到其他模块的安放。我们试图给出一个关于管理模块安放的规则，但我们发现有时一张示意图可以代替千言万语。如果您对我们的说明有任何的疑问，请查阅本章中的相关示意图。它可以帮助您方便地解决许多问题。

如果需要的话，有些模块可以放到扩展框架上。例如规定某个模块必须放在 CPU 模块的旁边并且系统是由两个 5 槽框架组成的，则扩展框架的第一槽和第二槽也是有效的位置。

以下表格提供了 DL305 模块的一般安放规则。

模块	限制
CPU	CPU 模块必须放在本地框架的的第一个插槽中。第一个插槽是紧靠电源部分的那个插槽。
16 点 I/O 模块	系统中最多能用 8 个 16 点的模块，但这要取决于 CPU 的型号及所使用的模块的型号。16 点模块必须安放在 CPU 后面的 8 个插槽中，在扩展框架上的也算。这 8 个插槽中的任意一个 16 点模块都能改成 8 点模块。
模拟量模块	模拟量模块必须安放在支持 16 点 I/O 模块的插槽中。
ASCII 模块	ASCII 模块必须安放在支持 16 点 I/O 模块的插槽中。
高速计数模块	高速计数模块应该安放在本地 CPU 框架的前 4 个插槽中。

常规框架一般规格

以下表格提供了 DL305 常规框架的一般特性。xxxx－1 型的框架在第二章中也有一定的介绍。

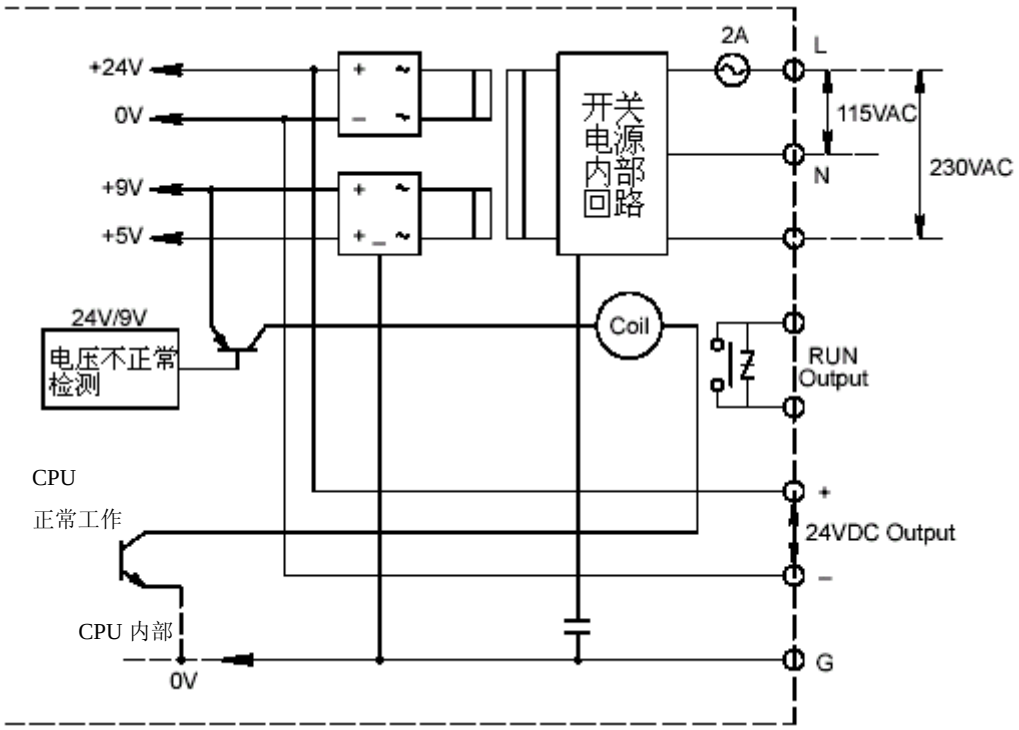
	D3-05B	D3-05BDC	D3-08B	D3-10B
总槽数	5	5	8	10
本地框架	能	能	能	能
扩展框架	能	能	不能	能
输入电压范围	97～132VAC 194~264VAC 47~63Hz	20.5~30VDC<10% 波动	97～132VAC 194~264VAC 47~63Hz	97～132VAC 194~264VAC 47~63Hz
框架功率消耗	最大 70VA (46W)	48W	最大 70VA (57W)	最大 70VA (57W)
最大瞬间电流	30A	30A	30A	30A
绝缘耐压	1500VAC 持续 1 分钟，电源接线端子，输出点，公共端，24VDC 之间	1500VAC 持续 1 分钟，在 24VDC 输入和输出点之间	1500VAC 持续 1 分钟，电源接线端子，输出点，公共端，24VDC 之间	2000VAC 持续 1 分钟，电源接线端子，输出点，公共端，24VDC 之间
绝缘电阻	>10MΩ,500VDC	>10MΩ,500VDC	>10MΩ,500VDC	>10MΩ,500VDC
电源输出（电压及其波动范围）	(5VDC)4.75~5.25V 小于 0.25V p-p (9VDC)8.0~13.5V 小于 0.2V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.1V p-p (9VDC)8.5~13.5V 小于 0.2V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.1V p-p (9VDC)8.0~12.0V 小于 0.2V p-p (24VDC)20~28V 小于 1.2V p-p	(5VDC)4.75~5.25V 小于 0.1V p-p (9VDC)8.0~12.0V 小于 0.2V p-p (24VDC)20~28V 小于 1.2V p-p
5V 电源最大电流	1.4A*	1.4A	1.4A（50℃） 1.0A（60℃）	1.4A（50℃） 1.0A（60℃）
9V 电源最大电流	0.8A*	0.8A	1.7A（50℃） 1.4A（60℃）	1.7A（50℃） 1.4A（60℃）
24V 电源最大电流	0.5A*	0.5A	0.6A	0.6A
辅助 24V 电源输出	100mA max	无	100mA max	100mA max
继电器输出	250VAC, 4A（阻性负载）	250VAC, 4A（阻性负载）	250VAC, 4A（阻性负载）	250VAC, 4A（阻性负载）
保险丝	2A（250V） 可自行更换	4A（250V） 可自行更换	2A（250V） 可自行更换	2A（250V） 可自行更换
尺寸 W×H×D	290×123×112 mm	290×123×112 mm	395×123×112 mm	465×123×112 mm
重量	964g	964g	1253g	1432g

*D3-05B 的总电流不能超过 2.3A。

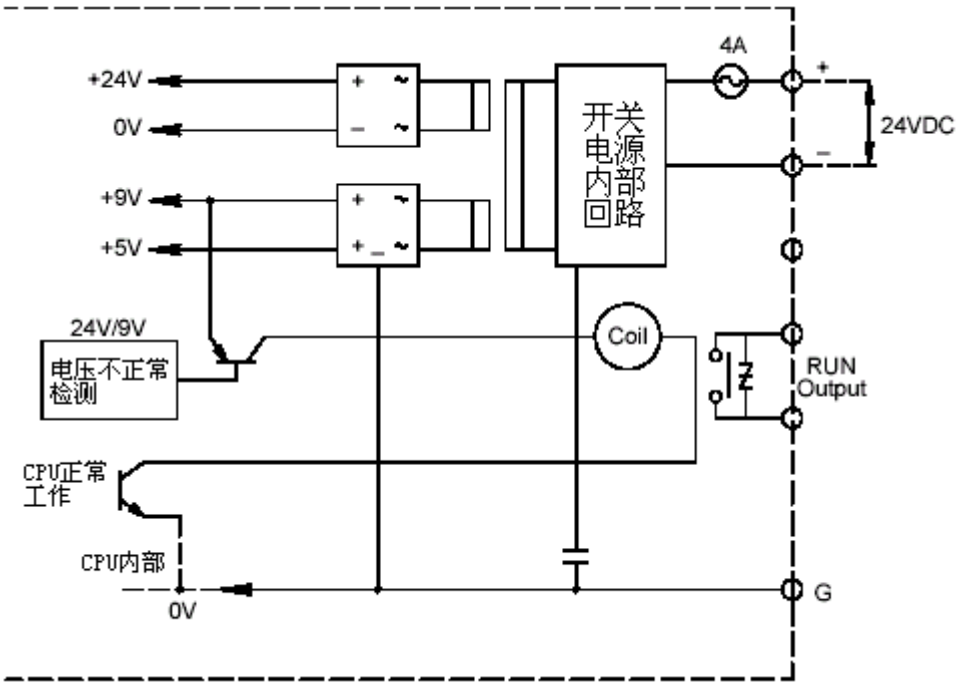
辅助 24VDC 输出 除了 D3-05BDC,所有的框架上都带有 24VDC 输出。该 24VDC 用来给外部设备或自身的某些模块供电。 该 24VDC 的使用会减少系统内部 24VDC 的供电能力。

电源示意图　　以下示意图有助于您详细了解上一页所述的 DL305 框架的一些规格。

D3-05B,D3-08B,D3-10B 的示意图



D3-05BDC 的示意图

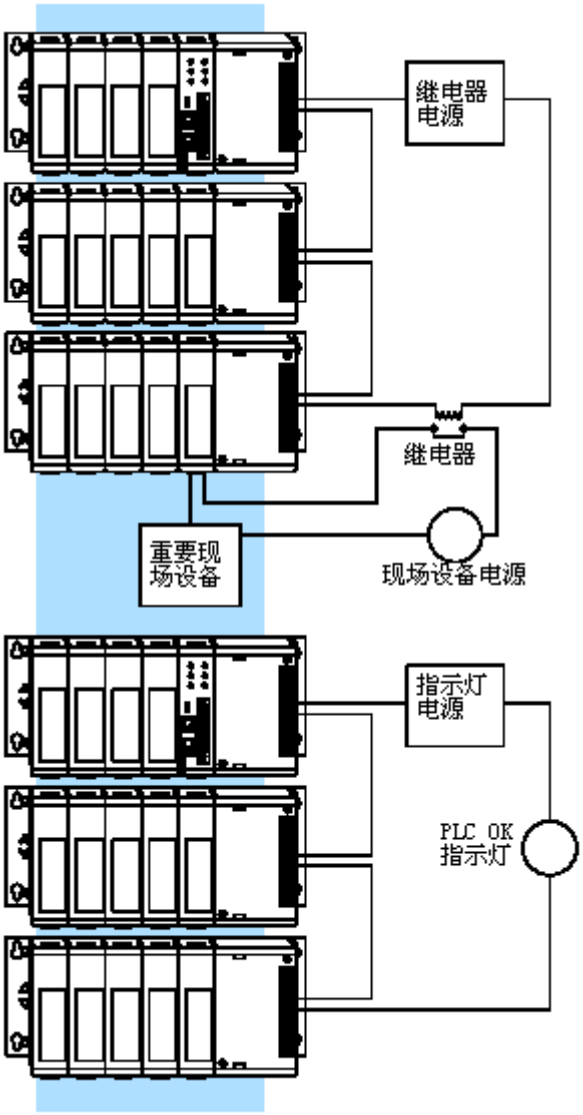


使用电源部分的运行继电器

DL305 框架的电源部分上有一个运行继电器输出，这可以检测本地框架或扩展框架是否发生了错误。

事件	本地框架运行继电器动作	本地框架运行继电器动作
从 PROGRAM 模式切换到 RUN 模式	吸合	不变
CPU 检测到致命错误	复位	不变
本地框架离开 RUN 模式	复位	不变
框架断电	复位	复位
9VDC 或 24VDC 发生故障	复位	复位

以下是使用运行继电器的一个例子。



使用运行继电器，可以在检测到错误时关闭重要现场设备

使用运行继电器来监视系统运行

本地和扩展 I/O 系统

框架使用表

这可以帮助您了解如何在您的控制系统中使用 DL305 系列的框架。以下是各个框架的使用注意点。

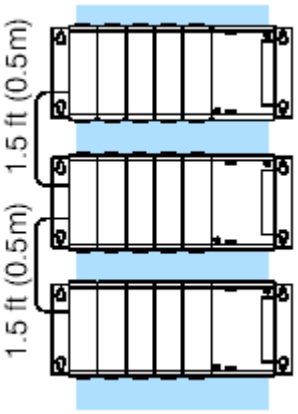
框架	总槽数	作为本地 CPU 框架	作为扩展框架
D3-05B	5	能	能
D3-05BDC	5	能	能
D3-08B	8	能	不能
D3-10B	10	能	能

本地/扩展连接

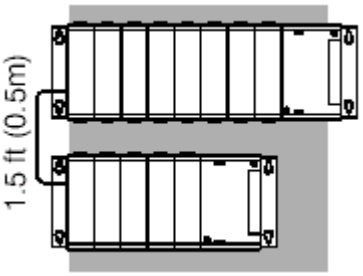
下图说明了本地框架和扩展框架的连接。

注意：在设计一个带扩展的系统时，您只能使用其中的一种。如果您使用的方法和以下所示都不同，则很有可能您的系统不能正常工作。

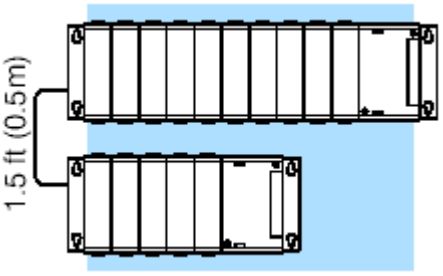
5 槽本地框架最多能带两个 5 槽扩展框架



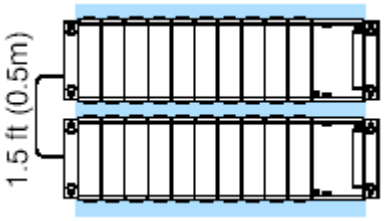
8 槽本地框架带一个 5 槽框架



10 槽框架带一个 5 槽框架



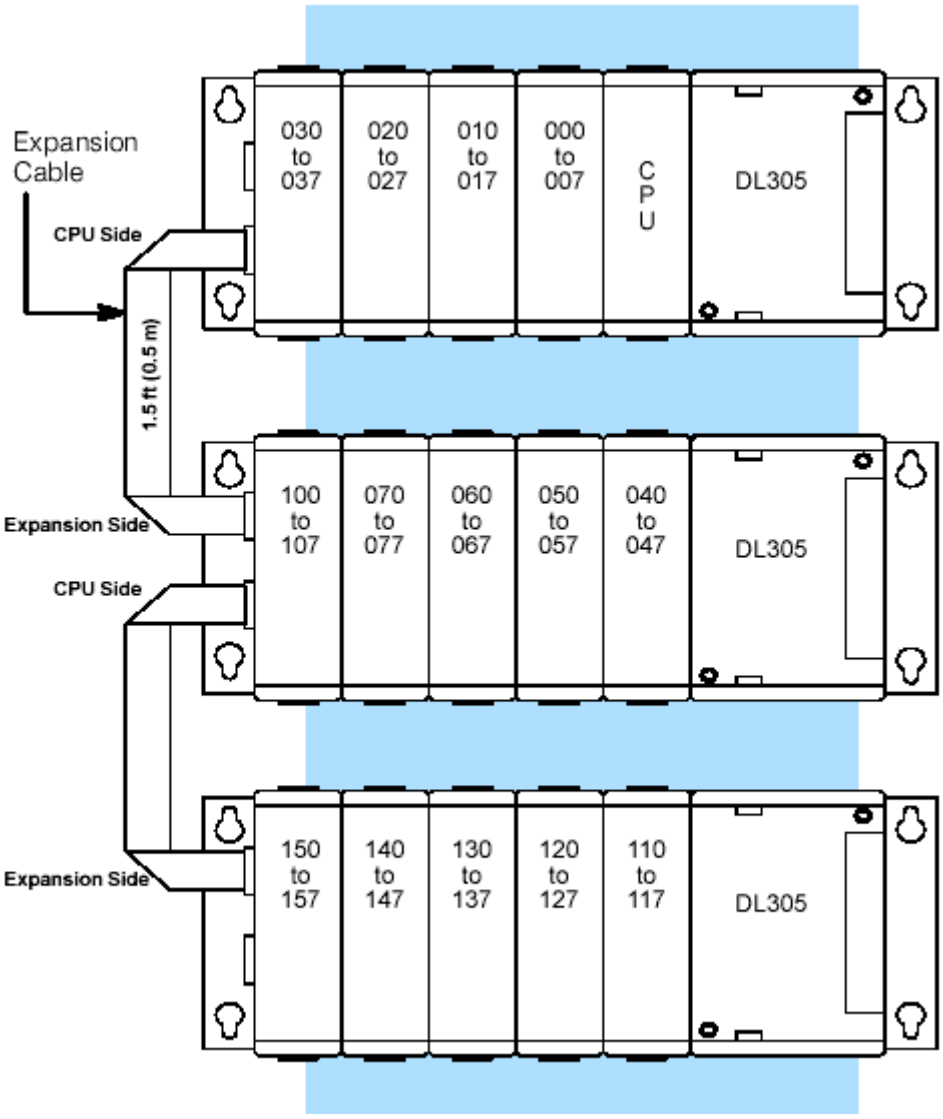
10 槽本地框架带一个 10 槽扩展框架



连接扩展框架

本地框架和扩展框架之间是用一根 50cm 长的电缆（D3-EXCBL）。应按下图所示连接。

框架上的上面一个接口用于接收前面一个框架的输入信号，而下面一个接口用于向下一个扩展框架发送输出信号。扩展电缆上标有“CPU Side”和“Expansion Side”。电缆的“CPU Side”端是接在框架的下面那个接口上，而“Expansion Side”是接在下一个框架的上面那个接口上。



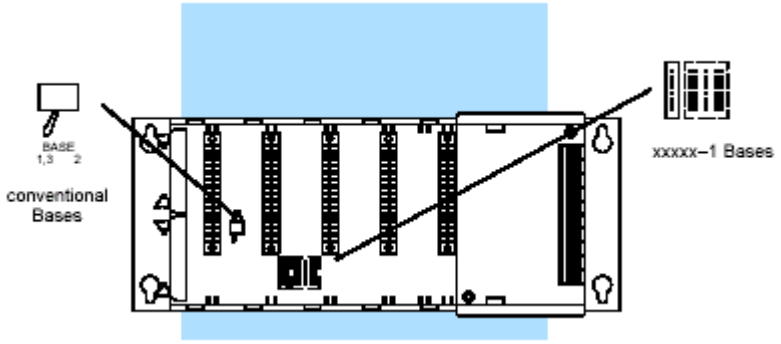
注意：扩展电缆应和 I/O 接线和电源接线分开放置。

设置框架开关

根据系统的不同，应对常规 5 槽和 10 槽框架上的跳线进行相应的设置。但 8 槽框架不含此类跳线开关。所有的 xxxx－1 型框架都有一个跳线开关，其中 10 槽框架有两个。

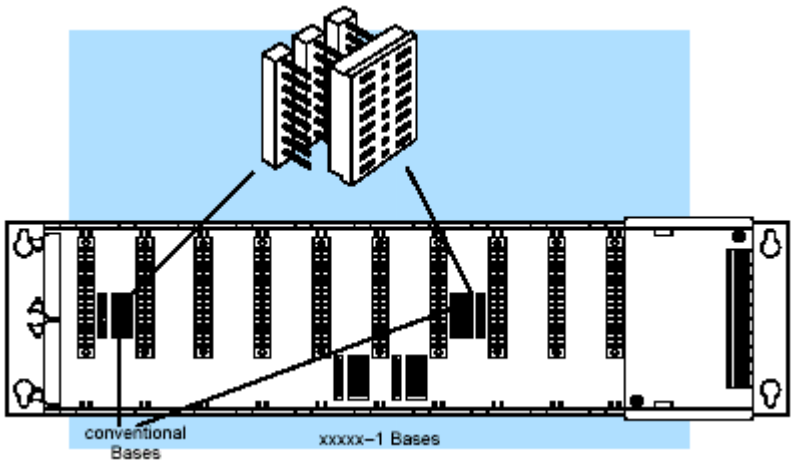
5 槽框架

常规 5 槽框架有一个两档的拨动开关用来把框架设置成本地框架，第一扩展框架，或第二（最后）扩展框架。xxxx－1 型的框架在第三槽和第四槽之间有个跳线开关。如果框架是本地 CPU 框架系统中的第三个框架，则该开关应放到“1，3”的位置。



10 槽框架

10 槽框架的第三槽和第四槽之间有一个跳线开关，用它可以把框架设置成本地 CPU 框架或扩展框架。同时在第九槽和第十槽之间（xxxx－1 型框架是第四槽和第五槽之间）还有一个跳线开关，用它可以把第十槽的地址设成 100～107 或 700～707。

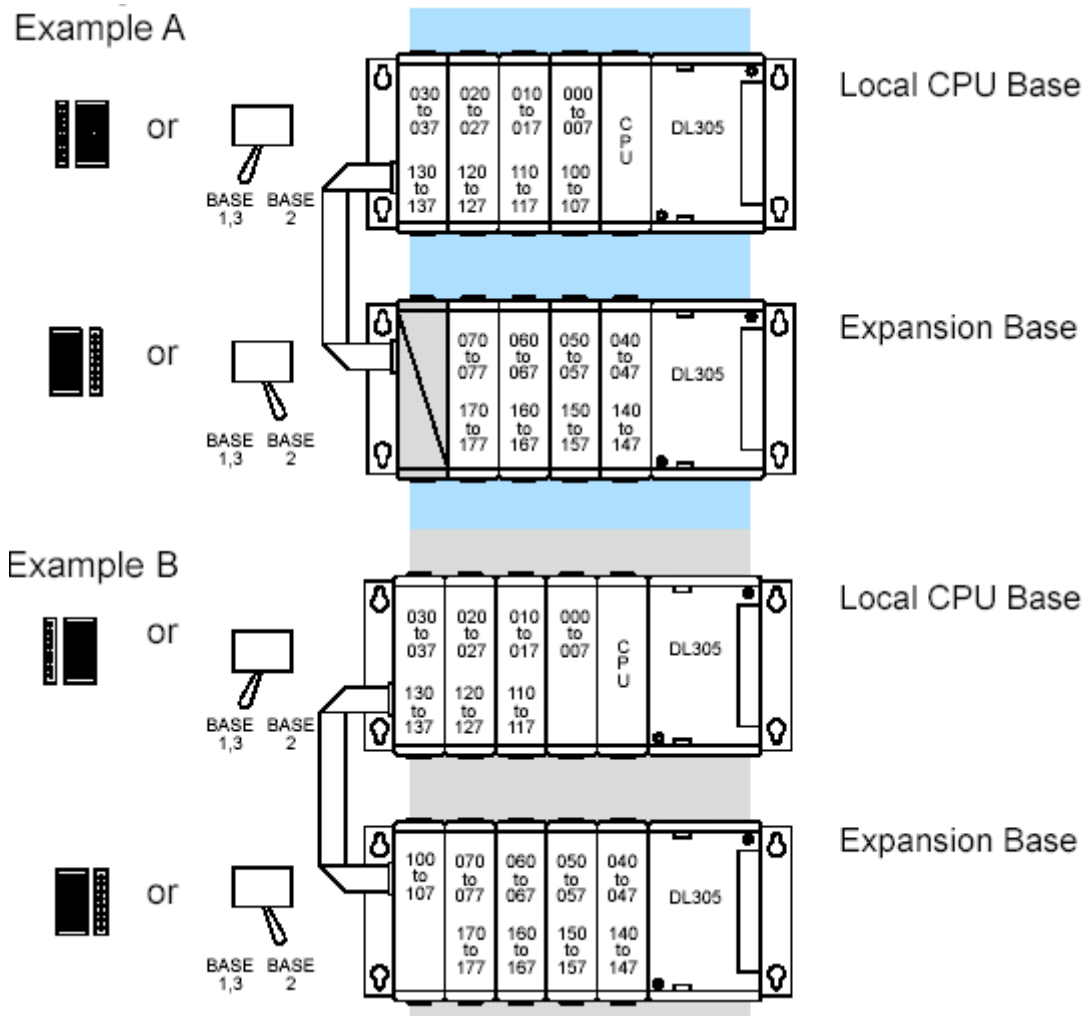


I/O 设置的例子

以下的系统设置的例子有助于您快速地对系统进行设置。这些系统设置了例子提供了 I/O 编号以及跳线开关应放的位置。

16 点 I/O 分配例子

当使用 16 点模块时，该模块后面 8 点的地址有可能已经被其他框架上的模块所占用了。下面的例子中 Example A 表示 CPU 旁边的插槽使用的是 16 点模块并占用地址 000～007 和 100～107。则扩展框架上的最后一个插槽不能使用，因为它占用的地址也是 100～107。Example B 表示在 CPU 旁边的插槽中使用 8 点模块，同时在扩展框架的最后一个插槽上也使用一个 8 点模块，因为它们的地址没有冲突。这两种设置的方法都是有效的。



最大限度使用 I/O 点的例子

以下是通过使用 16 点模块来达到最大数量 I/O 点数的例子。减掉例子中的任何一个模块都是允许的。8 点模块可以改成 16 点模块，同时其他模块的 I/O 分配也不受影响。当把 16 点模块改成 8 点模块时，16 点模块后 8 个点所占用的地址有可能不能使用在其他槽位中，这要取决于您的系统配置。

5 槽本地 CPU 框架的 I/O 设置

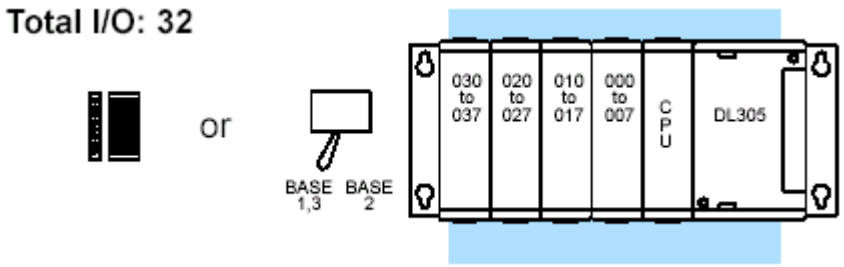
以下及后面几页是 5 槽框架插 8 点 I/O 模块，16 点模块，带一个扩展框架和两个扩展框架的例子。

开关设置

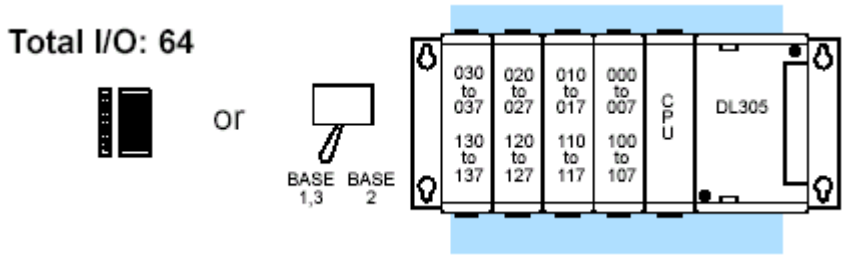
5 槽的框架里面有一个拨动开关或跳线开关，可以用它来选择：

框架类型	开关位置 （常规框架）	跳线开关位置 （xxxx－1 型框架）
本地 CPU	Base 1，3	跳线放右边
第一扩展	Base 2*	跳线放左边
第二扩展	Base 1，3	跳线放右边

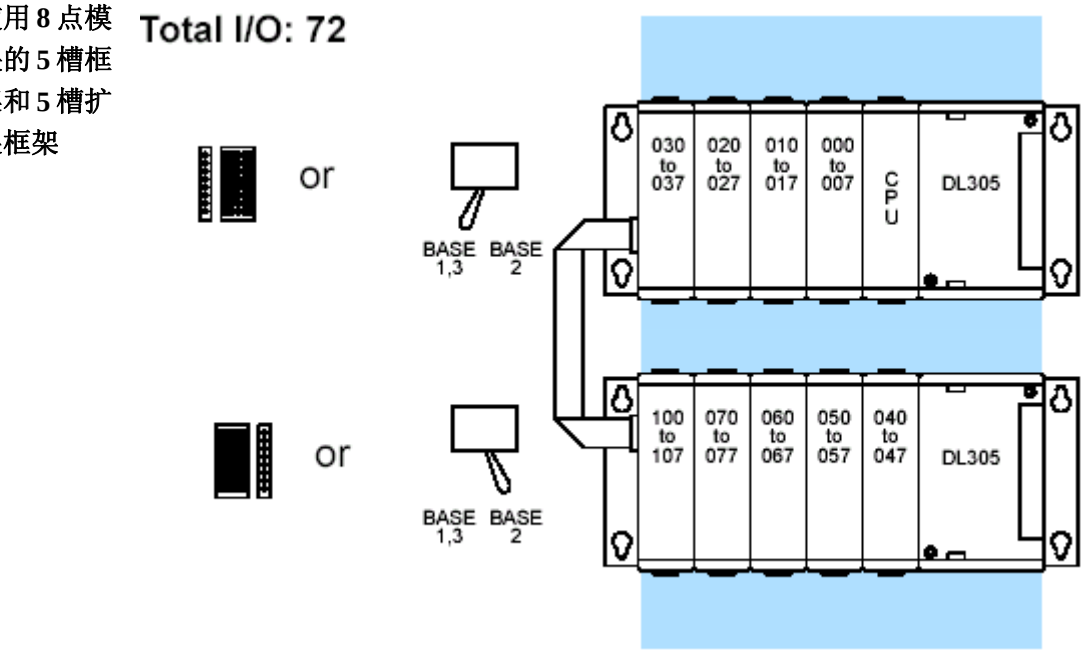
使用 8 点模块的 5 槽框架



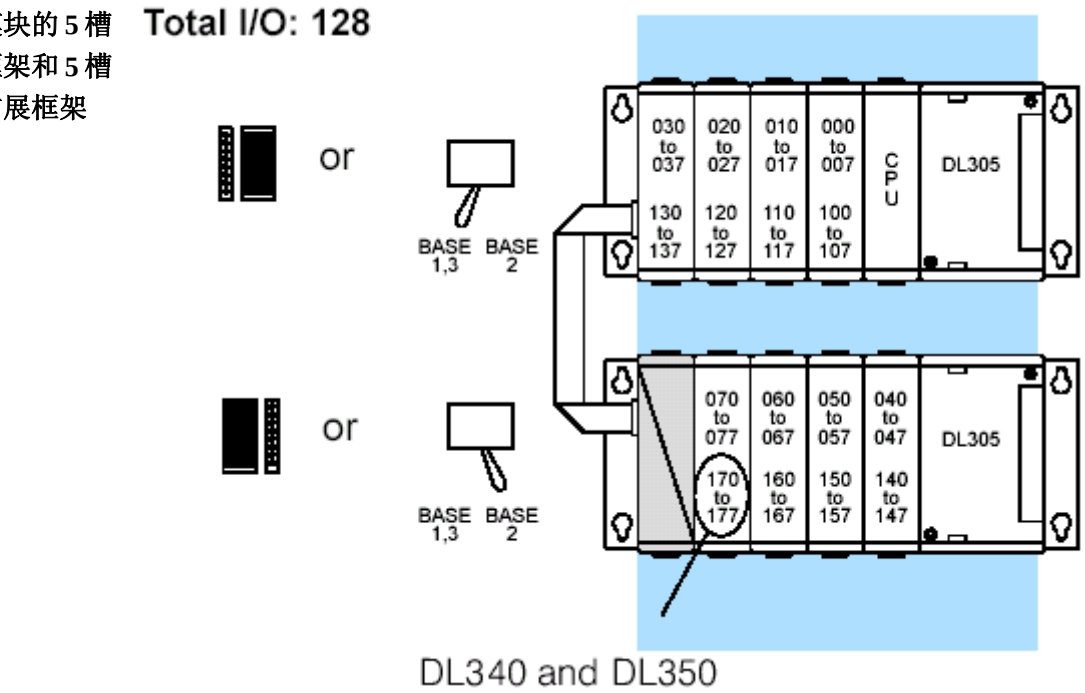
使用 16 点模块的 5 槽框架



使用 8 点模块的 5 槽框架和 5 槽扩展框架



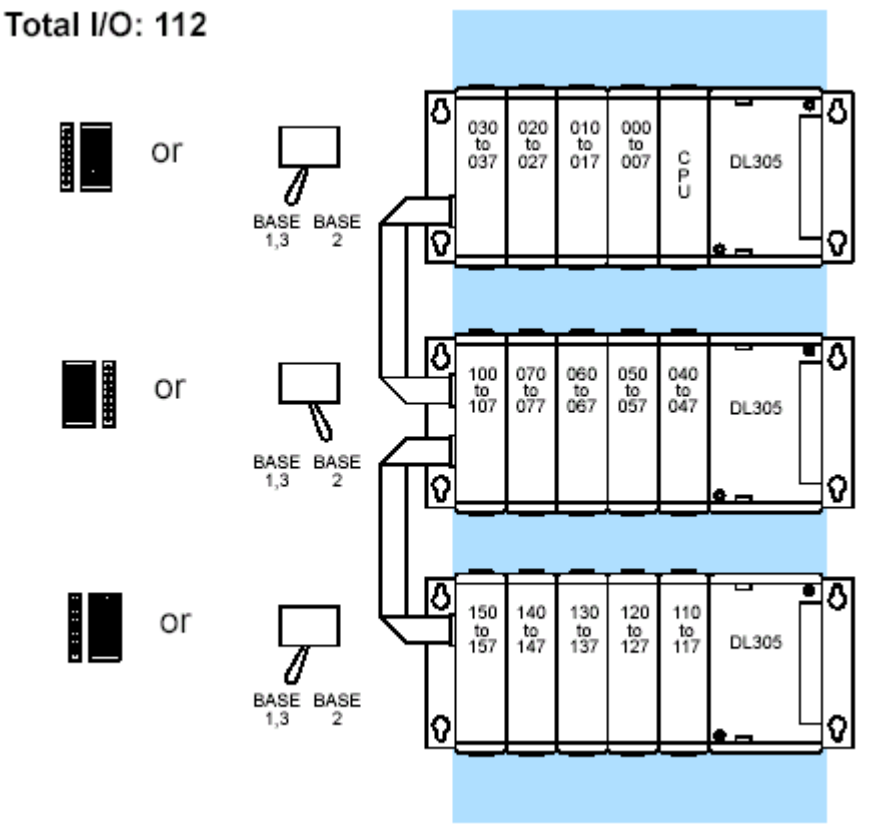
使用 16 点模块的 5 槽框架和 5 槽扩展框架



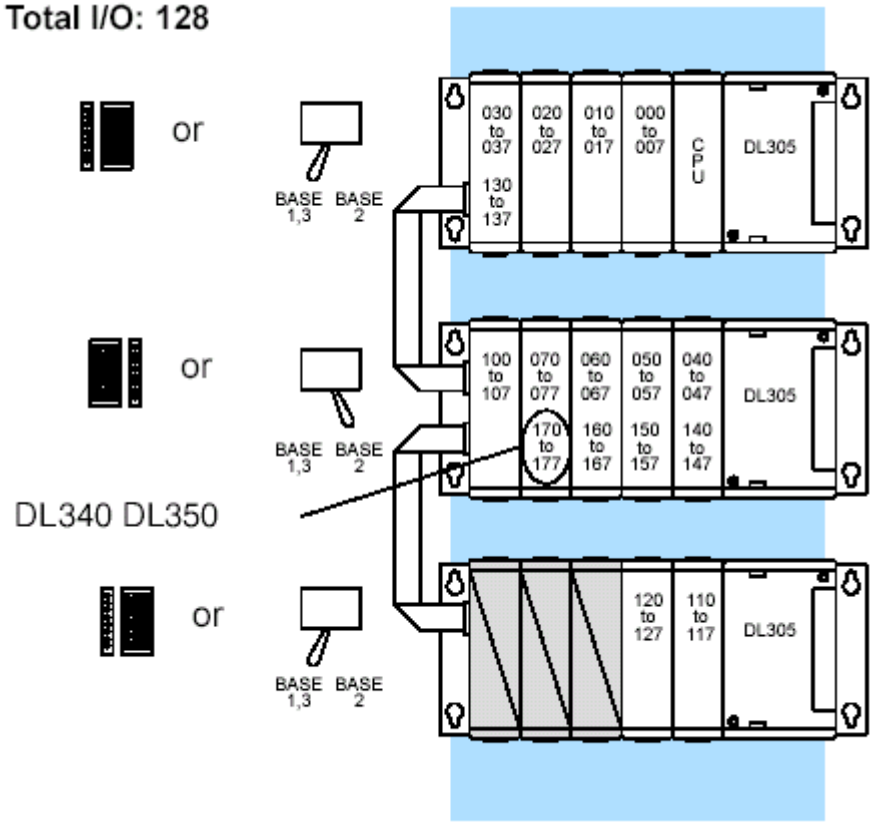
DL340 and DL350

注意：如果 16 点模块用在扩展框架的最后两个槽中，则程序中不能使用 160～177 号中间继电器。尽管这些点是作为 I/O 点，但在 DirectSOFT 中编程时还是要把它们写作 M160~M177。

使用 8 点模块的 5 槽框架和两个 5 槽扩展框架



使用 16 点模块的 5 槽框架和两个 5 槽扩展框架



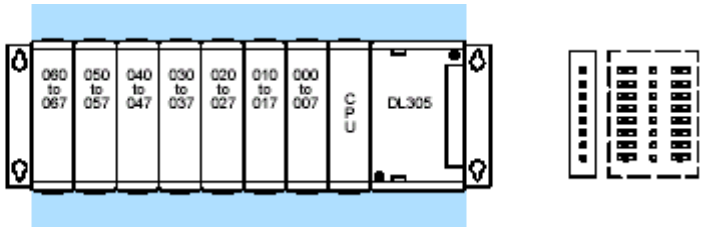
注意：如果 16 点模块用在扩展框架的最后两个槽中，则程序中不能使用 160～177 号中间继电器。尽管这些点是作为 I/O 点，但在 DirectSOFT 中编程时还是要把它们写作 M160~M177。

8 槽本地 CPU 框架的 I/O 设置

下面几张图将教您如何在 8 槽框架上使用 8 点或 16 点模块以及如何带 5 槽的扩展框架。其中框架右边的图是 xxxx－1 型框架的跳线的状态。

8 槽框架使用 8 点模块

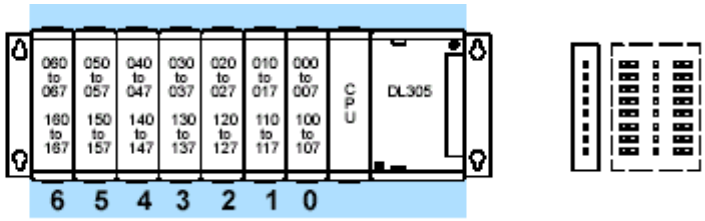
Total I/O: 56



8 槽框架使用 16 点模块

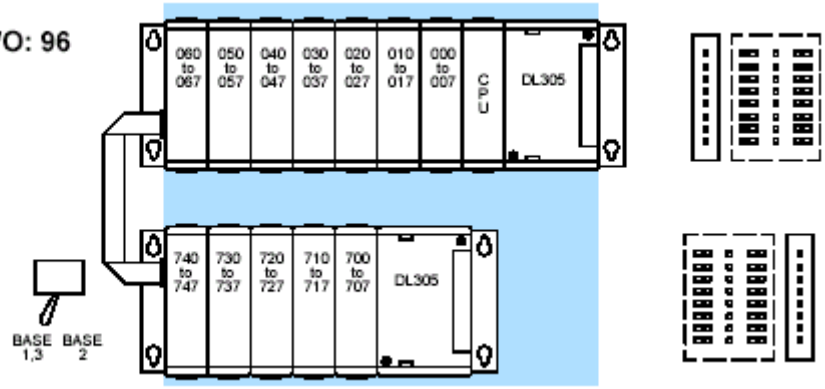
Total I/O: 112

*对于I/O点160～167，请参考底下的“注意”。



8 槽框架和 5 槽扩展框架都使用 8 点模块

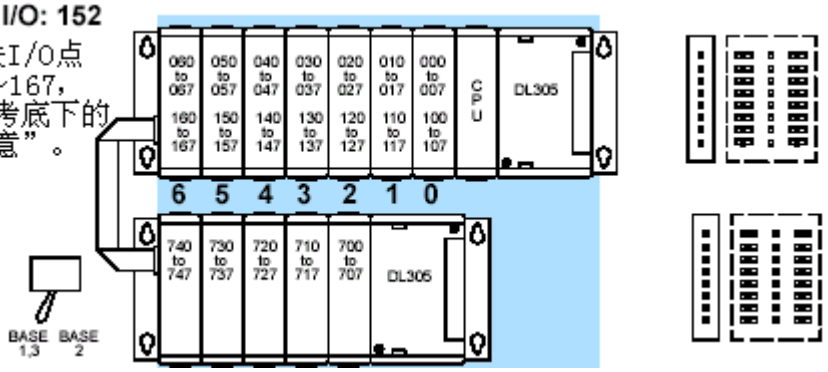
Total I/O: 96



8 槽框架和 5 槽扩展框架都使用 16 点模块

Total I/O: 152

*有关I/O点160～167，请参考底下的“注意”。



注意：如果 16 点模块用在第六槽，160～167 将不能作为中间继电器来使用。尽管这些点是作为 I/O 点，但在 DirectSOFT 中编程时还是要把它们写作 M160~M177。

10 槽本地 CPU 框架的 I/O 设置

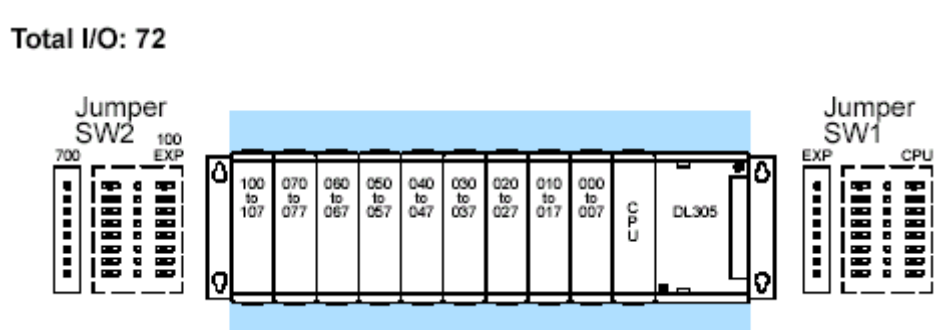
下面几张图将教您如何在 10 槽框架上使用 8 点或 16 点模块以及如何带 5 槽或 10 槽的扩展框架。

跳线设置

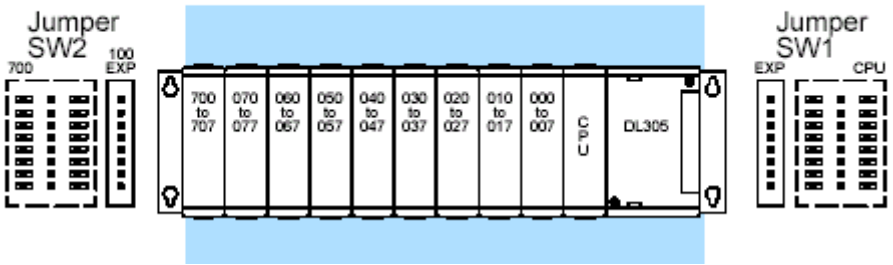
10 槽框架上有两个用来设置框架类型和地址范围的跳线开关。这两个跳线开关分别在第三槽和第四槽之间（SW1）以及第九槽和第十槽之间（SW2）。其中 SW1 是用来把框架设置成本地 CPU 框架或扩展框架。而 SW2 是用来把本地 CPU 框架上的第十槽的地址设成 100～107 或 700～707。如果把第十槽的地址设成 700～707，则可以在 CPU 旁边的插槽中使用 16 点模块。但是，这个跳线的位置也会影响扩展模块的 I/O 编号。

注意：扩展框架上的 SW2 必须设置成“100 EXP”。

最后一槽的地址范围 100～107



最后一槽的地址范围 700～707

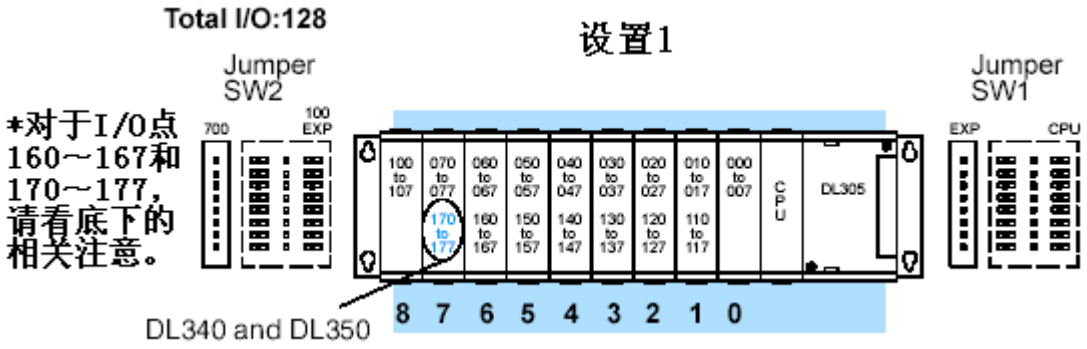


10 槽框架使用 16 点模块

下面的两张图说明了本地 CPU 框架使用 16 点模块以及有关如何最大限度使用 I/O 点数的两种设置方法。

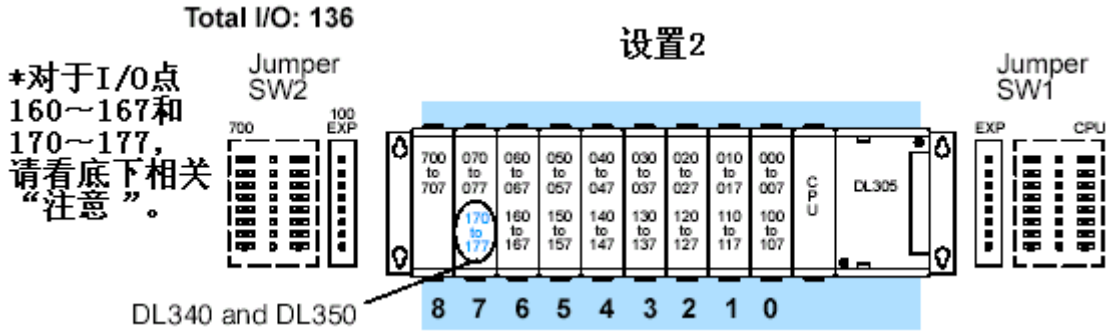
设置 1

在设置 1 中，CPU 旁边的槽中使用 8 点模块，因此最后一个槽可以使用地址 100～107。当 SW2 放到“100 EXP”位置时，最后一个槽的地址是 100～107，因此第一槽的地址只能 000～007。这意味着如果您要使用该种配置，第一个插槽只能放 8 点的模块。



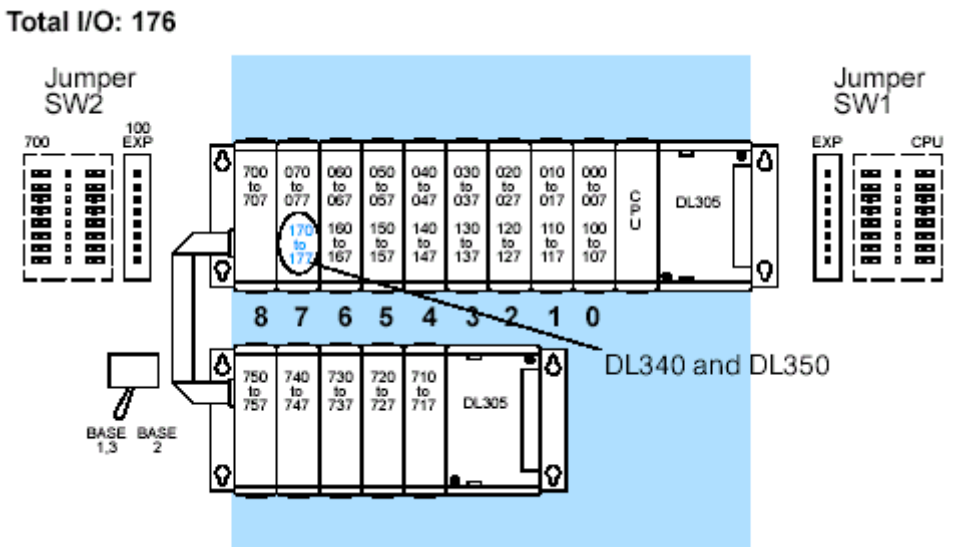
设置 2

在设置 2 中，CPU 旁边的槽中使用 16 点模块并且最后一块模块占用地址 700～707。这是 10 槽本地 CPU 框架 I/O 数量设置最多的一种方法。如果 SW2 放在“700”的位置，则第一槽中可以放 16 点模块。SW2 可以限制最大的 I/O 点数，因为扩展 I/O 的编号一般是从 700 开始的。



注意：当使用 DL330 或 DL330P CPU 时，如果在第六槽中使用 16 点模块。则 160～167 不能当作中间继电器来使用。当使用 DL340 或 DL350 CPU 时，如果第六或第七槽中使用 16 点模块，160～167 和 170～177 不能当作中间继电器来使用。尽管您可以把这些点当作 I/O 来使用，但在 DirectSOFT 中您还是要把这些 I/O 点写成 M160~C167 和 M170~C177。

10 槽框架和 5 槽扩展框架使用 16 点模块



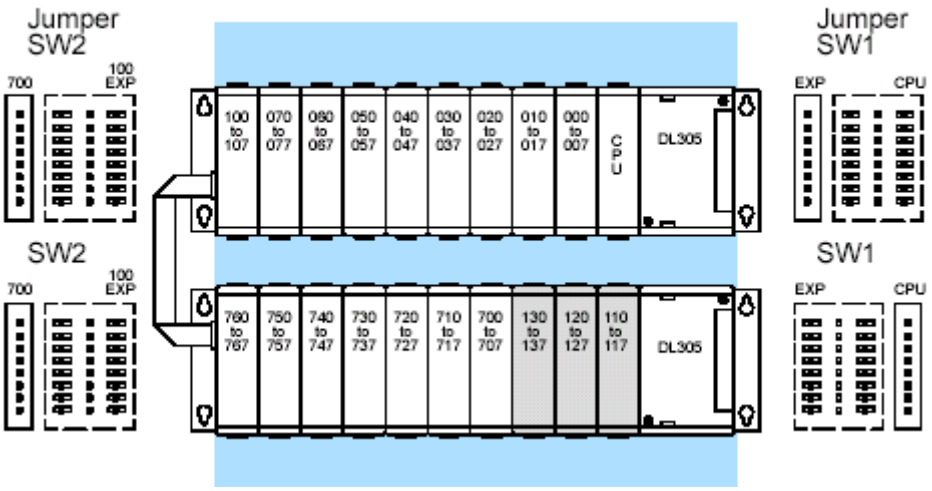
注意：当使用 DL330 或 DL330P CPU 时，如果在第六槽中使用 16 点模块。则 160～167 不能当作中间继电器来使用。当使用 DL340 或 DL350 CPU 时，如果第六或第七槽中使用 16 点模块，160～167 和 170～177 不能当作中间继电器来使用。尽管您可以把这些点当作 I/O 来使用，但在 DirectSOFT 中您还是要把这些 I/O 点写成 M160~C167 和 M170~C177。

扩展地址取决于本地 CPU 框架的设置

I/O 地址的改变取决于本地 CPU 框架的设置。注意，当本地 CPU 框架只含 8 点模块时，在扩展模块中可以使用 110～117，120～127，130～137。如果本地 CPU 框架含有 16 点模块而且 I/O 地址 110～117，120～127 和 130～137 已被占用，则在扩展框架中不能再使用这些 I/O 点。

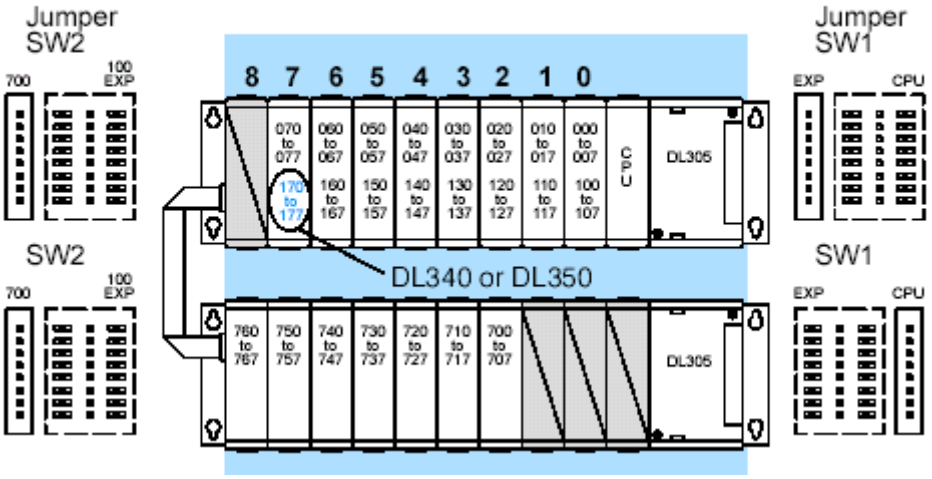
10 槽框架 10 槽扩展框架使用 8 点模块

Total I/O: 152



10 槽框架和 10 槽扩展框架使用 16 点模块

Total I/O: 184



注意：当使用 DL330 或 DL330P CPU 时，如果在第六槽中使用 16 点模块。则 160～167 不能当作中间继电器来使用。当使用 DL340 或 DL350 CPU 时，如果第六或第七槽中使用 16 点模块，160～167 和 170～177 不能当作中间继电器来使用。尽管您可以把这些点当作 I/O 来使用，但在 DirectSOFT 中您还是要把这些 I/O 点写成 M160~C167 和 M170~C177。

光洋电子(无锡)有限公司
Koyo ELECTRONICS （WUXI） CO., LTD.
地址：江苏省无锡市滨湖区建筑西路 599 号 1 栋 21 层
邮编：214072
电话：0510-85167888 传真：0510-85161393
http：//www.koyoele.com.cn

KEW-M3111A

2015 年 8 月