

LISTEN.
THINK.
SOLVE.SM



Logix5000 控制器设计要点

1756 ControlLogix

1756 GuardLogix

1768 CompactLogix

1769 CompactLogix

1789 SoftLogix5800

1794 FlexLogix

PowerFlex 700S with DriveLogix

参考手册

重要用户信息

固态设备的工作特性与机电设备有很大区别。《固态控制系统应用、安装及维护安全性指南》(出版物SGI-1.1可从用户所在地的罗克韦尔自动化销售办公室或在线<http://literature.rockwellautomation.com>网站获得)描述了固态设备与硬接线机电设备之间的一些重要区别。由于这种区别和固态设备的应用种类广泛，负责应用这些设备的全部人员必须保证该设备的每个计划应用都能够接受。

由于产品的应用所造成的间接损害，罗克韦尔自动化公司不承担任何责任。

本手册中的图表、示例程序仅用于演示目的。由于实际的安装配置中有许多特殊要求和不确定因素，因此对于依据本手册中的实例和图表进行的实际应用，罗克韦尔自动化公司不对其可靠性承担责任。

使用该手册中提及的信息、电路图、设备或软件，罗克韦尔自动化公司不承担专利权责任。

本手册的版权归罗克韦尔自动化公司所有。在没有罗克韦尔自动化公司书面同意的情况下，禁止任何人 / 单位全部或者部分复制本手册。

警告 	标识能够在危险环境中引起爆炸，从而导致人身伤害或死亡、财产损失、或经济损失的实践或环境信息。
重要事项	标识成功应用和了解产品的关键信息。
注意事项 	标识可能会导致人身伤害或死亡、财产损失，或经济损失的实践或环境信息。该标志帮助用户识别危险、避免危险、考虑后果。
警告 	标签可能安装在设备(例如: 变频器或者电机)表面或内部警告人们此处可能有危险电压。
烧伤危险 	标签可能安装在设备(例如: 变频器或者电机)表面或内部警告人们其表面温度可能危险。

Allen-Bradley、Rockwell Automation、ControlLogix、GuardLogix、CompactLogix、SoftLogix、FlexLogix、PowerFlex 700S、DriveLogix、FactoryTalk、FactoryTalk Administration Console、FactoryTalk Alarms and Events、FactoryTalk、FactoryTalk Live Data、FactoryTalk View、FactoryTalk View Studio、Data Highway Plus、SynchLink、PLC-5、SLC、SLC 500、RSLinx、RSBizWare Batch、ControlFlash、Ultra3000 和 PanelView 是罗克韦尔自动化公司的商标。

所有不属于罗克韦尔自动化的商标均属各自所有公司的财产。

Logix5000 控制器比较特性:

特性:	1756 ControlLogix	1756 GuardLogix	1768 CompactLogix	1769 CompactLogix	1789 SoftLogix5800	1794 FlexLogix	PowerFlex 700S2 DriveLogix
控制器任务:	100 个任务 - 事件型任务 - 支持所有的事件触发器	100 个任务 - 事件型任务 - 支持所有的事件触发器	16 个任务 - 事件型任务: 支持消费者标签、EVENT 指令、运动轴和运动事件触发器	1769-L35CR, -L35E: 8 个任务 1769-L32C, -L32E: 6 个任务 - 事件型任务: 支持消费者标签、EVENT 指令触发器	100 个任务 - 事件型任务: 支持所有的事件触发器和外部(Outbound)及 Windows 事件	8 个任务 - 事件型任务: 支持消费者标签和 EVENT 指令触发器	8 个任务 - 事件型任务: 支持运动轴和运动事件触发器
用户内存	1756-L55M12:750K 字节 1756-L55M13:1.5M 字节 1756-L55M14:3.5M 字节 1756-L55M16:7.5M 字节 1756-L55M22:750K 字节 1756-L55M23:1.5M 字节 1756-L55M24:3.5M 字节 1756-L60M03SE:750K 字节 1756-L61:2M 字节 1756-L62:4M 字节 1756-L63:8M 字节 1756-L64:16M 字节	1756-L61S: 2M 字节标准的 1M 字节安全的 1756-L61S: 4M 字节标准的 1M 字节安全的	1768-L43:2M 字节	1769-L31:512K 字节 1769-L32x:750K 字节 1769-L35x:1.5M 字节	1789-L10: 2 M 字节; 3 槽;无运动控制 1789-L30: 64 M 字节; 5 槽 1789-L60: 64 M 字节; 16 槽	1794-L34:512 K 字节	256 K 字节 768 K 字节带内存扩展
非易失性用户内存	1756-L55M12:无 1756-L55M13:无 1756-L55M14:无 1756-L55M16:无 1756-L55M22:有 1756-L55M23:有 1756-L55M24:有 1756-L6x:CompactFlash	CompactFlash	CompactFlash	CompactFlash	无	有	有(扩展内存)
内置的通讯端口	1 个 RS-232 串行通讯口	1 个 RS-232 串行通讯口	1 个 RS-232 串行通讯口	1769-L31:2 个 RS-232 串行通讯口 1769-L32C, -L35CR: 1 个 ControlNet 端口和 1 个 RS-232 串行通讯口 1769-L32E, -L35E: 1 个 EtherNet/IP 端口和 1 个 RS-232 串行通讯口	取决于个人计算机	1 个 RS-232 串行通讯口 2 个用于安装 1788 通讯卡的插槽	1 个 RS-232 串行通讯口 1 个用于安装 1788 通讯卡的插槽
通讯选项	- EtherNet/IP - ControlNet - DeviceNet - DH+ - Remote I/O - SynchLink	- EtherNet/IP (标准的;安全的) - ControlNet (标准的; 安全的) - DeviceNet (标准的; 安全的) - DH+ - Remote I/O - SynchLink	- EtherNet/IP - ControlNet - DeviceNet	- EtherNet/IP - ControlNet - DeviceNet	- EtherNet/IP - ControlNet - DeviceNet	- EtherNet/IP - ControlNet - DeviceNet	- EtherNet/IP - ControlNet - DeviceNet
串行口通讯	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485 - Modbus	- ASCII - DF1 full/half duplex - DF1 radio modern - DH-485
连接	100 个 ControlNet 连接 128 个 EtherNet/IP 连接 64 个 TCP/IP 连接	48 个 ControlNet 连接 128 个 EtherNet/IP 连接 64 个 TCP/IP 连接	48 个 ControlNet 连接 128 个 EtherNet/IP 连接 64 个 TCP/IP 连接	32 个 ControlNet 连接 32 个 EtherNet/IP 连接 32 个 TCP/IP 连接	48 个 ControlNet 连接 - EtherNet/IP 连接数量受通讯卡的类型和数量的限制	32 个 ControlNet 连接 32 个 EtherNet/IP 连接 64 个 TCP/IP 连接	32 个 ControlNet 连接 32 个 EtherNet/IP 连接 64 个 TCP/IP 连接
控制器冗余	完全支持	不支持	不适用	不适用	不适用	通过 DeviceNet 进行备份	不适用
简单的运动控制	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器	- 步进器(Stepper) - 通过 DeviceNet 的伺服 - 模拟量交流变频器
集成的运动控制	SERCOS 接口 模拟量选项: 编码器输入 LDT 输入 SSI 输入	SERCOS 接口 模拟量选项: 编码器输入 LDT 输入 SSI 输入	SERCOS 接口	不适用	SERCOS 接口 模拟量编码器输入	不适用	1 个完整的伺服 1 个反馈轴
编程语言	- 梯形图 - 结构文本 - 功能块 - 顺序功能图	- 梯形图 - 结构文本 - 功能块 - 顺序功能图	- 梯形图 - 结构文本 - 功能块 - 顺序功能图	- 梯形图 - 结构文本 - 功能块 - 顺序功能图	- 梯形图 - 结构文本 - 功能块 - 顺序功能图 - 外部例程 (使用 C/C++ 开发)	- 梯形图 - 结构文本 - 功能块 - 顺序功能图	- 梯形图 - 结构文本 - 功能块 - 顺序功能图

注释:

更新摘要

该版本的设计指南适用于 16 版本的 Logix5000 控制器固件。

更新内容	页码
连续型任务执行时间至少 1 毫秒	25
事件型任务的输出处理	26
组态系统内务处理时间片的未使用部分	28
Add-On 指令	31
增加 LINT 数据类型	41
电子锁	64
控制 HART 设备	68
控制 FOUNDATION 现场总线设备	69
运行时添加通用 ControlNet 和 EtherNet/IP 模块	73
在 EtherNet/IP 网络上单播(Unicast)生产者和消费者标签	77
DeviceNet 标签生成器工具	83
FactoryTalk 报警和事件系统	91
控制器初始化的固件升级	117
支持 SERCOS 变频器的自动固件升级	118

此外，下面这些模块可供使用：

- 1768-L43 CompactLogix 控制器
- 1768-CNB, 1768-CNBR ControlNet 模块
- 1768-EWEB EtherNet/IP web 服务器模块
- 1756-L64 ControlLogix 控制器
- 1756-CN2, 1756-CN2R ControlNet 模块
- 1756-EN2T EtherNet/IP 模块

更新摘要

注释：

前言	
设计 Logix5000 系统	简介.....11
第 1 章	
Logix5000 控制器资源	简介.....13
	控制器连接.....15
	确定全部的连接需求.....17
第 2 章	
将逻辑划分为任务、程序、 例程和 Add-On 指令	简介.....19
	确定何时使用任务、程序、例程.....20
	指定任务的优先级.....21
	管理用户的任务.....22
	影响任务执行的考虑因素.....23
	组态连续型任务.....25
	组态周期型任务.....25
	组态事件型任务.....25
	组态事件型任务的指南.....26
	选择系统内务处理百分数.....27
	管理系统内务处理时间片百分数.....28
	开发例程中的应用代码.....29
	与子例程传递参数的指南.....31
	用户自定义 Add-On 指令的指南.....31
	编程方法.....34
	逻辑的控制器预扫描.....36
	SFC 逻辑的控制器后扫描.....37
	定时器的执行.....37
	在线编辑 SFC.....39
第 3 章	
与其它的控制器共享标签数据 (生产者和消费者标签)	简介.....41
	数据类型的指南.....42
	数组.....43
	数组的指南.....44
	数组的间接寻址.....45
	数组索引的指南.....46
	数组索引的预扫描.....46
	用户自定义结构体的指南.....47
	为位类型标签选择数据类型.....48
	连续的位地址.....49
	字符串数据类型的指南.....50
	PLC-5/SLC 500 的字符串存取.....50
	组态标签.....51
	基本型标签的指南.....52
	创建别名标签.....53
	数据作用域的指南.....53
	标签名称的指南.....54
	标签的说明.....55

第 4 章

地址数据

简介	57
生产者和消费者标签的指南	58
为生产者和消费者标签指定RPI速率的指南	59
管理生产者和消费者标签连接的指南	59
组态基于消费者标签的事件型任务	59
比较消息和生产者/消费者标签	60

第 5 章

与 I/O 的通讯

简介	61
缓存的I/O数据	61
为 I/O 模块指定 RPI 速率的指南	62
I/O 模块的通讯格式	63
管理 I/O 连接的指南	65
控制1771 I/O模块	67
与HART设备进行通讯	68
与FOUNDATION现场总线设备进行通讯	69
为I/O数据创建标签	70
控制器所属关系	71
运行时/在线添加I/O模块	72

第 6 章

确定合适的网络

简介	75
选择一种网络	75
EtherNet/IP网络拓扑结构	76
EtherNet/IP网络的指南	77
EtherNet/IP系统中交换机的指南	78
ControlNet 网络拓扑结构	79
ControlNet网络的指南	79
非规划ControlNet网络的指南	81
比较规划的和非规划的ControlNet通讯	82
DeviceNet网络拓扑结构	82
DeviceNet 网络的指南	83

第 7 章

与其它设备的通讯

简介	85
缓冲的消息	86
消息缓存	86
消息的指南	89
管理消息连接的指南	89
块传送消息的指南	90
映射标签	90

第 8 章

FactoryTalk 报警和事件系统

简介	91
基于 Logix 的报警指令的指南	92
组态基于Logix的报警指令	93
报警过程	95
缓存的报警	97
可编程访问的报警信息	97

第 9 章		
优化运动控制应用	简介.....	99
	粗略的更新速率.....	99
	轴限制.....	100
	性能限制.....	100
	运动事件任务触发器.....	101
第 10 章		
优化使用 HMI 的应用	简介	103
	比较RSView Enterprise和RSView32软件.....	105
	RSView SE 软件的指南.....	105
	RSLinx 软件如何与 Logix5000 控制器进行通讯.....	106
	比较 RSLinx Classic 和 RSLinx Enterprise 软件.....	107
	RSLinx 软件的指南.....	108
	组态控制器标签的指南.....	109
第 11 章		
为批处理控制开发设备状态 (Equipment Phases)	简介.....	111
	Equipment Phases的指南.....	111
	Equipment Phase指令.....	112
第 12 章		
优化过程控制应用	简介	113
	比较PID和PIDE指令.....	113
	编程PID回路的指南.....	114
	高级的过程指令	115
	面板(Faceplates)	116
	比较Active-X面板和图形库组件.....	116
第 13 章		
管理固件	简介.....	117
	管理控制器固件的指南.....	117
	比较固件选项.....	118
	RSLogix 5000固件管理者的指南.....	119
	访问固件.....	121

注释：

设计 Logix5000 系统

简介

本手册提供用户进行系统优化的指南。本手册提供用户需要的系统设计的信息。除了包含在每章中的控制器指定标题外，手册的后面还包括：

- 常用术语表
- 相关出版物列表

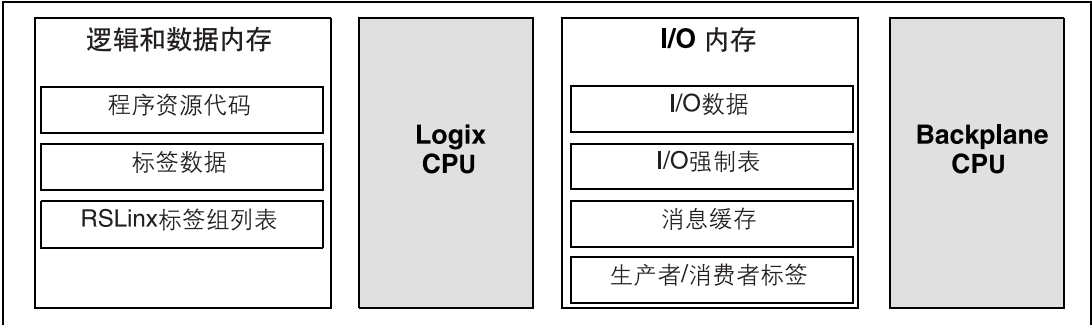
本手册适合于有经验的Logix系统程序员。手册中的信息假设用户已经了解如何执行这些指导。手册后面是相关出版物的网站列表，用户可以利用这些列表以获取更多关于如何执行这些指导的详细资料。

注释：

Logix5000 控制器资源

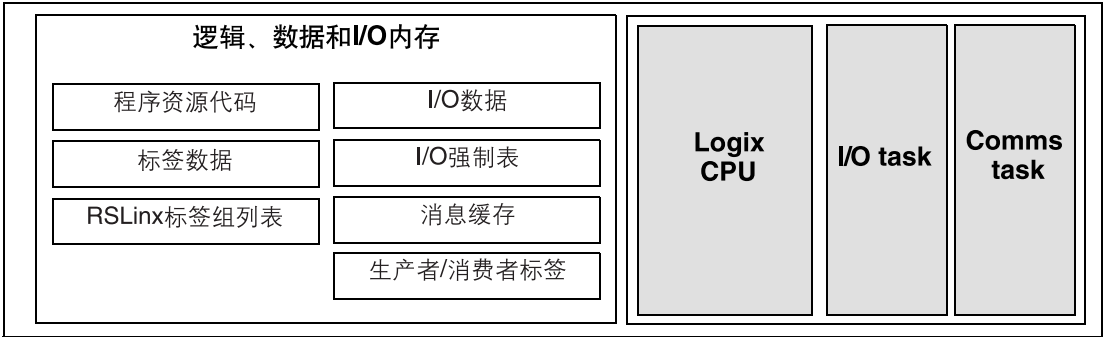
简介 控制器不同，资源也不同。

1768 CompactLogix 和 ControlLogix 控制器 - 有两个独立的内存。



- Logix CPU 执行应用项目代码和消息
- 背板 CPU 与进行 I/O 通讯，并发送数据给背板或者从背板接收数据。这个 CPU 与 Logix CPU 是相互独立的，所以它发送和接收 I/O 信息与程序的执行是不同步的。

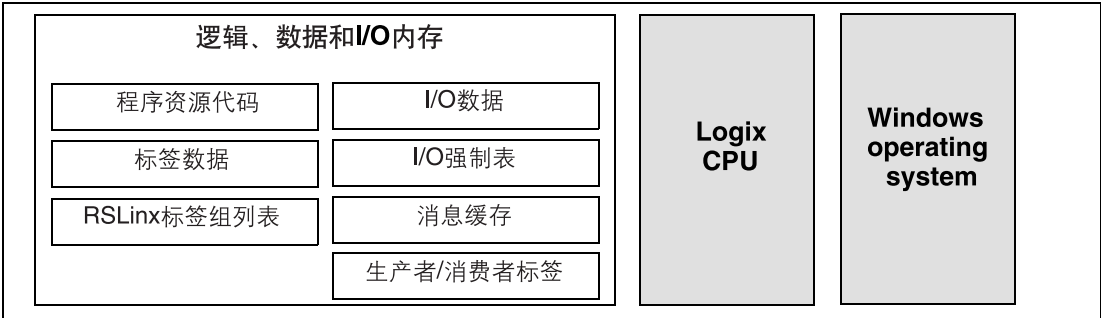
1769 CompactLogix、FlexLogix 和 DriveLogix 控制器 - 只有一个内存。



这些控制器只有一个单独的 CPU，执行所有操作。独立的任务执行 I/O 和网络通讯。这些任务模拟背板 CPU。

控制器	I/O 任务优先级	通讯任务优先级
1769 CompactLogix、FlexLogix 和 DriveLogix	6	12

SoftLogix 控制器 - 只有一个内存。



SoftLogix 控制器有一个独立的 CPU，与 Windows 操作系统协同工作执行所有的操作。为了让 I/O 任务和通讯任务使用控制器优先级，SoftLogix 控制器处理这些任务时使用 Windows 优先级。

控制器	I/O 任务优先级	通讯任务优先级
SoftLogix	Windows 优先级 16(空闲状态下)	Windows 优先级 16(空闲状态下)

对于所有控制器，内存存在运行时用来进行：

- 消息处理来缓存输入和输出消息。
- RSLinx 数据处理来存储标签组。
- 在线编辑来存储编辑梯级。
- 图形趋势图来缓存数据。

减少内存使用

这些等式提供控制器内存使用估计。

控制器任务(Controller Tasks)	_____	*4000 =	_____	字节(最小一个需求)
数字量 I/O 点(Digital I/O Points)	_____	*400 =	_____	字节
模拟量 I/O 点(Analog I/O Points)	_____	*2600 =	_____	字节
DeviceNet 模块(DeviceNet modules ¹)	_____	*7400 =	_____	字节
其它通讯模块(Other Communication modules ²)	_____	*2000 =	_____	字节
运动轴(Motion axis)	_____	*8000 =	_____	字节
FactoryTalk 报警指令(FactoryTalk Alarm Instruction)	_____	*1000 =	_____	字节(每个报警)
FactoryTalk 订阅者(FactoryTalk Subscriber)	_____	*10000=	_____	字节(每个订阅者)
		总计 =	_____	字节

1 第一个 DeviceNet 模块是 7400 字节。其它的 DeviceNet 模块每个是 5800 字节。

2 统计系统中所有的通讯模块，不只是这些本地框架。这包括设备连接模块、适配器模块和 PanelView 终端端口。

预留 20-30% 的控制器内存便于系统升级。

RSLink 软件使用的 Logix5000 控制器内存

RSLink 软件的内存需求取决于 RSLink 软件读取的数据类型。这些等式提供内存估计。

RSLink 内务处理

(每个连接)	*1345 = _____	字节(缺省状态下是 4 个连接)
单个标签	*45 = _____	字节
数组 / 结构体	*7 = _____	字节
	总计 = _____	字节

将标签整理到数组或者结构体中能够减少在获得数据时的通讯内务处理和连接数。

比较 PLC/SLC 内存

Logix5000 控制器使用已编译的指令来提供比 PLC 或者 SLC 处理器更快的执行次数。相对于 PLC 和 SLC 处理器的指令来说，已编译的指令使用更多的内存。

如果有一个 PLC/SLC 程序，用户能够通过以下的公式估计它在 Logix5000 控制器中的字节数。

$$PLC/SLC \text{ 字数} * 18 = \text{Logix5000 字节数}$$

控制器连接

Logix5000 控制器使用连接在两个设备之间建立通讯链路。这些连接可以是：

- 控制器与本地 I/O 模块或者本地通讯模块。
- 控制器与远程 I/O 或者远程通讯模块。
- 控制器与远程 I/O(机架优化)模块。
有关更多的 I/O 连接的信息，请参阅第 5 章，“与 I/O 的通讯”。
- 生产者消费者标签。
有关更多的信息，请参阅第 4 章，“与其它的控制器共享标签数据(生产者和消费者标签)”。
- 消息。
有关更多的信息，请参阅第 7 章，“与其它设备的通讯”。
- 访问 RSLogix 5000 编程软件。
- RSLink 软件访问 HMI 或者其它的软件应用项目。

控制器有不同的通讯限制。

通讯属性	ControlLogix 和 SoftLogix	CompactLogix	FlexLogix 和 DriveLogix
连接	250	100	100
缓存消息 ¹	消息和块传送相结合 32	消息和块传送相结合 32	消息和块传送相结合 32
面向非连接接收缓存	3	3	3
面向非连接发送缓存	10(能够增加到 40)	10(能够增加到 40)	10(能够增加到 40)

1 有关消息和缓存的更多信息请参阅第 7 章。

连接的限制可能最终由用来连接的通讯模块来确定。如果消息的路径通过通讯模块进行路由，与消息相关联的连接也可以用来计算通讯模块的连接限制。

控制器	通讯设备	支持的连接
ControlLogix	1756-CN2, 1756-CN2R	100 个 CIP 连接 (任何预定的连接和消息连接的组合)
	1756-CNB, 1756 -CNBR	64 个 CIP 连接，取决于 RPI，推荐只使用 48 个连接 (任何预定的连接和消息连接的组合)
	1756-EN2T, 1756-ENBT 1756-EWEB	128 个 CIP 连接 64 个 TCP/IP 连接
1768 CompactLogix	1768-ENBT 1768-EWEB	64 个 CIP 连接 32 个 TCP/IP 连接
1769 CompactLogix	1769-L32C, 1769-L35CR	32 个 CIP 连接，取决于 RPI，推荐 22 个连接可以预定 其余连接能够用做消息连接(或者所有 32 个连接，如果用户没有预定的连接)
	1769-L32E, 1769-L35E	32 个 CIP 连接 64 个 TCP/IP 连接
FlexLogix 包含 DriveLogix 的 PowerFlex 700S	1788-CN _x , 1788-CN _x R	32 个 CIP 连接，取决于 RPI，推荐 22 个连接可以预定 其余连接能够用做消息连接(或者所有 32 个连接，如果用户没有预定的连接)
	1788-ENBT	32 个 CIP 连接 64 个 TCP/IP 连接
SoftLogix5800	1784-PCICS	128 个 CIP 连接 127 个能够用来预定连接

确定总连接需求

Logix5000 控制器的总连接包括本地连接和远程连接。对于 FlexLogix 或者 CompactLogix 控制器来说统计本地连接不存在问题, 因为两种控制器支持系统中允许的最大模块数目。相对于其它控制器来说 ControlLogix 和 SoftLogix 控制器支持更多的通讯模块, 所以用户必须统计本地连接确保将连接数控制在 250 个连接限制内。使用下表来统计本地连接。

连接类型	设备数量	×	每个模块的连接	=	总连接
本地 I/O 模块(总是一个直接连接)		×	1	=	
运动伺服模块		×	3	=	
ControlNet 通讯模块		×	0	=	
EtherNet/IP 通讯模块		×	0	=	
DeviceNet 通讯模块		×	2	=	
DH+/Remote I/O 通讯模块		×	1	=	
RSLogix 5000 编程软件访问控制器		×	1	=	
Total					

用户选择的通讯模块决定有多少个可用的远程连接。使用下表统计远程连接。

连接类型	设备数量	×	每个模块的连接	=	总连接
远程 ControlNet 通讯模块 作为直接连接组态 作为机架优化连接组态		×	0 或者 1	=	
通过 ControlNet 的分布式 I/O 模块 (直接连接)		×	1	=	
远程 EtherNet/IP 通讯模块 作为直接连接组态 作为机架优化连接组态		×	0 或者 1	=	
通过 EtherNet/IP 的分布式 I/O 模块 (直接连接)		×	1	=	
通过 DeviceNet 网络的远程设备 (本地 DeviceNet 模块的机架 优化连接)		×	0	=	
其它远程通讯适配器		×	1	=	
生产者标签和第一个消费者 每个其它的消费者		×	1 1	=	
消费者标签		×	1	=	
已连接的消息 (CIP 数据表读 / 写和 DH+)		×	1	=	
块传送消息		×	1	=	
RSLink 软件访问 HMI 或者其它的软件应用项目		×	4	=	
RSLink Enterprise 软件访问 HMI 或者其它的软件应用项目		×	5	=	
总计					

注释：

将逻辑划分为任务、程序、例程和 Add-On 指令

简介

控制器运行系统是一个有优先权的多任务系统，符合 IEC 61131-3 编程标准。

任务用来组态 控制器执行

该任务为一个或者一系列程序提供时序和优先级信息。用户能够组态连续型、周期型和事件型的任务。

程序用来分组 数据和逻辑

任务中包含程序，每个程序带有自己的例程和程序范围标签。当任务触发时，分配到任务中的所有程序按照它们在控制器管理器(Controller Organizer)中的列表执行。

程序用在多个程序员开发的项目中。在开发期间，一个程序中的代码使用的程序范围标签可以复制在第二个程序中，并将标签名冲突的可能性减少到最小。

15 以上的版本，任务能够包含程序和设备阶段(Phase)。

例程用来将 可执行代码封装 在单一的编程语言中

例程包含可执行代码。每个程序有一个主例程，主例程是在程序中第一个执行的例程。使用逻辑指令，如跳转到子程序(JSR)指令，来访问其他的例程。用户也能够指定一个可选择的故障例程。

关于选择编程语言的更多信息，请参阅 29 页的开发例程中的应用项目代码。

Add-On 指令用来将可执行代码 封装在用户自定义指令中

一个 Add-On 指令是一个用户创建的封装逻辑指令。Add-On 指令有助于将控制器项目划分成更小的、更易于管理的单元。

选择何时使用任务、程序、例程和 Add-On 指令 使用以下注意事项来确定何时使用任务、程序和例程。

比较	任务	程序和设备阶段 (Phase)	例程	Add-On 指令
可用的数量	根据不同的控制器使用的数量也不同 (4、6、8 或者 32)	每个任务 32 个程序和设备阶段(Phase) (ControlLogix 和 SoftLogix 控制器是 100 个)	每个程序的例程数量没有限制	在项目中，Add-On 指令的数量没有限制
功能	确定如何和何时代码将会执行	管理需要共享通用数据区的例程组	包括可执行的代码 (梯形图，功能块，顺序功能图和结构文本)	包括可执行的代码 (梯形图，功能块和结构文本)
使用	- 大多数代码运行在连续型任务中 - 对于较慢的过程或者当运行对时间要求严格时使用周期型任务 - 对于操作需要与一个特殊的事件同步的情况，使用时间型任务	- 将主要设备或者车间单元分配在独立的程序中 - 使用程序区分不同的程序员或者创建可重复使用的代码 - 一个任务内已经组态的执行顺序 - 区分单一的批处理状态或者不连续的机械运行	- 在一个例程中区分机械或者单元的功能 - 根据控制过程使用适合的编程语言 - 将代码模块化成子程序，能够多次调用	- 标准化代码模块 - 指定的或者专用的操作 - 扩展到基础指令集 - 以一种编程语言形式封装，可以在另一种编程语言中使用 - 用在逻辑和数据监控场合
注意事项	- 数量太多的任务不易于调试 - 可能需要禁止一些任务的输出处理来改善性能 - 任务能被禁止以停止执行	- 多个程序调用的数据必须在控制器作用域区域内 - 按照执行顺序在控制器管理器中列表排序	- 多次调用的子程序不易于调试 - 可以在程序范围内和控制器作用域内调用数据 - 调用多个的例程会影响扫描时间 - 按照 Main(主例程)、Fault(故障例程)和字母表顺序在控制器管理器中列表排序	- 如果用户有许多参数或者专门的选项，建议使用多个 Add-On 指令 - 调用多个 Add-On 指令会影响扫描时间 - 必须使用对照参考或者查找来定位调用一个 Add-On 指令 - 只能离线编辑 - 支持有限的数据类型 - 改变数据值必须应用到每个程序中

指定任务的优先级

控制器的每个任务有优先级。一个较高的优先级(例如: 1)会中断任何较低的优先级任务(例如: 15)。连续型任务有最低的优先级, 总是被周期型任务或者事件型任务中断。

Logix5000 控制器	控制器支持的用户任务	控制器支持的优先级
ControlLogix	32	15
1768-L43 CompactLogix	16	15
1769-L35CR, 1769-L35E CompactLogix	8	15
1769-L32C, 1769-L32E CompactLogix	6	15
1769-L31 CompactLogix	4	15
FlexLogix	8	15
PowerFlex 700S with DriveLogix	8	15
SoftLogix5800	32	3

Logix5000 控制器有这些任务类型

优先级	用户任务	描述
最高的	N/A	CPU 内务处理 – 串口和通用 CPU 运行
	N/A	运动规划 – 在过程刷新速率下执行
	N/A	冗余任务 – 在冗余系统中与 1757-SRM 通讯
	N/A	趋势图数据采集 – 趋势图数据高速采集
	优先级 1 事件型 / 周期型	N/A
	优先级 2 事件型 / 周期型	N/A
	优先级 3 事件型 / 周期型	N/A
	优先级 4 事件型 / 周期型	N/A
	优先级 5 事件型 / 周期型	N/A
	优先级 6 事件型 / 周期型	在基于框架 RPI 设置的周期型任务下 1769 CompactLogix, FlexLogix, 和 DriveLogix 控制器处理 I/O
	优先级 7 事件型 / 周期型	N/A
	优先级 8 事件型 / 周期型	N/A
	优先级 9 事件型 / 周期型	N/A
	优先级 10 事件型 / 周期型	N/A
	优先级 11 事件型 / 周期型	N/A
最低的	优先级 12 事件型 / 周期型	DriveLogix 与变频器通讯 CompactLogix 和 FlexLogix 通讯和预定连接保持
	优先级 13 事件型 / 周期型	N/A
	优先级 14 事件型 / 周期型	N/A
	优先级 15 事件型 / 周期型	N/A
	连续型	消息管理器 – 基于系统内务处理时间片

如果一个周期型任务或者事件型任务执行时, 另一个任务触发, 而且两个任务有相同的优先级, 任务时间片执行时间增加 1ms 直到其中一个任务完成执行。

管理用户任务 用户能够组态这些用户任务。

如果用户想要指令逻辑	则	描述
任何时候	连续型任务	连续型任务在后台运行。CPU 时间不会分配给其它的操作或者任务用于执行连续型的任务。 - 连续型任务在任何时候都运行。当连续型任务完成一次扫描，它会立即重新扫描。 - 项目不需要使用连续型任务。如果需要用，只有一个连续型任务。
- 固定的周期里 (例如每 100ms) - 其它逻辑的扫描内执行多次	周期型任务	周期型任务在指定的时间间隔内执行功能。无论何时周期型任务触发时，周期型任务： - 中断任何较低的优先级任务。 - 执行一次。 - 在连续型任务中断的地方返回重新控制。
当事件发生时立即执行	事件型任务	事件型任务在指定的事件发生(触发)时执行功能。无论何时事件型任务触发时，事件型任务： - 中断任何较低的优先级任务。 - 执行一次。 - 在连续型任务中断的地方返回重新控制。 关于事件型任务的详细信息，请参阅 25 页的组态事件型任务。一些 Logix5000 控制器不支持所有的触发。

创建的用户任务出现在控制器的任务文件夹中。这些预定义系统任务不出现在任务文件夹中，它们不计入控制器的任务限制：

- 运动规划
- I/O 处理
- 系统内务处理
- 输出处理

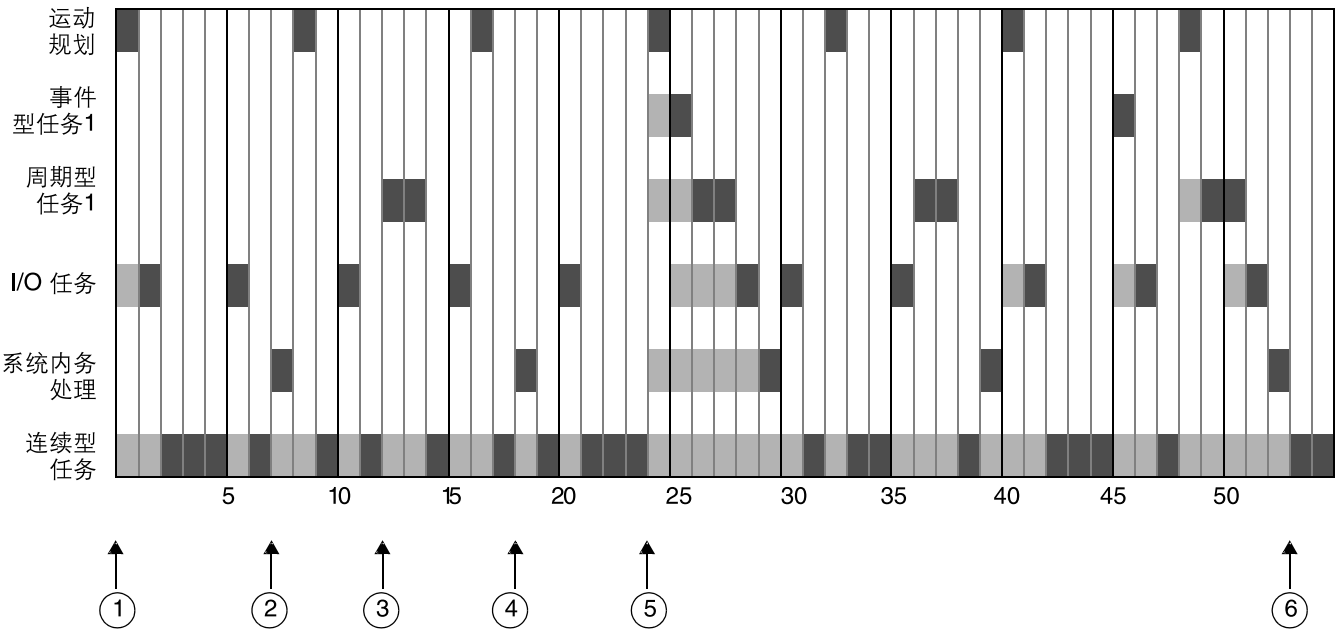
影响任务执行的考虑因素

考虑因素	描述
运动规划 请参阅 99 页的“优化运动控制应用项目”	无论任何优先级，运动规划中断所有其它的任务。 - 运动组的轴数和粗略更新周期影响运动规划执行的次数和时间。 - 如果运动规划正在执行，一个任务触发，任务等待直到运动规划结束才执行。 - 如果在任务执行时，粗略更新时间触发，任务停止让运动规划执行。
I/O 处理	CompactLogix、FlexLogix、DriveLogix 和 SoftLogix 控制器使用指定的周期型任务来处理 I/O 数据。这个 I/O 任务： - 对于 CompactLogix、FlexLogix 和 DriveLogix 控制器来说，运行在优先级 6。对于 SoftLogix 控制器来说，运行在 Windows 优先级 16(空闲下)。 - 较高的优先级接管 I/O 任务并能够影响处理。 - 在给系统规定的最快的 RPI 下执行。 - 在扫描已组态的 I/O 模块时执行。 - 对于本地 I/O，在每个任务结束时进行刷新。
系统内务处理 请参阅 27 页的“选择系统内务处理百分数”。	系统内务处理是控制器执行消息通讯和后台任务花费的时间。 - 消息通讯是没有在项目的 I/O 组态文件夹中的通讯，例如 MSG 指令。 - 消息通讯只发生在周期型任务和事件型任务没有运行时。如果用户使用多个任务，确保它们的扫描次数和执行时间间隔为消息通讯预留足够的时间。 - 系统内务处理能够中断周期型任务。 - 系统内务处理时间片指定控制器专用于消息通讯的时间(除了周期型任务和事件型任务的时间)百分数。 - 控制器每次执行消息通讯超过 1ms 并重新执行连续型任务。 - 调整任务的刷新率，在执行逻辑和服务消息通讯之间需要获得最佳的交替转换。 - 未使用的分配给系统内务处理的时间用来增加连续型任务的扫描速率，这会导致扫描时间变化。要减少这种变化，通过控制器属性设置给通讯预留时间。
输出处理	在一个任务结束时，控制器为系统输出模块执行输出处理。输出处理可能影响系统 I/O 模块的刷新。
许多的任务	如果有用户有很多任务，则： - 连续型任务会由于时间太长而不能完成。 - 其它的任务可能经历重叠。如果一个任务被中断的过于频繁或者中断时间太长，在它重新触发之前可能无法执行。 - 控制器通讯会变慢。 - 如果应用项目用于数据采集，尽量避免多个任务。因为在多个任务之间的转换会限制通讯带宽。

本例描述了包含这些任务的项目执行。

任务	优先级	周期	执行时间	持续时间
运动规划	N/A	8 ms (过程刷新率)	1 ms	1 ms
事件型任务 1	1	N/A	1 ms	1...2 ms
周期型任务 1	2	12 ms	2 ms	2...4 ms
I/O 任务 — 对于 ControlLogix 和 SoftLogix 控制器不适用	7	5 ms (最快的 RPI)	1 ms	1...5 ms
系统内务处理	N/A	时间片 = 20%	1 ms	1...6 ms
连续型任务	N/A	N/A	20 ms	48 ms

图表符号：  任务执行  任务中断(暂停)



描述	
①	最初，控制器执行运动规划和 I/O 任务(如果一个存在)。
②	在执行连续型任务 4ms 后，控制触发系统内务处理。
③	周期型任务 1 的周期结束(12ms)，所以任务中断连续型任务。
④	在连续型任务执行 4ms 后，控制触发系统内务处理。
⑤	事件型任务 1 触发。 事件型任务 1 等待直到运动规划执行完成。 较低优先级的任务经历更长的延时等待。
⑥	连续型任务自动重新扫描。

组态连续型任务

当用户打开 RSLogix5000 项目时就会自动创建一个连续型任务。连续型任务与 PLC-5 和 SLC500 处理器的逻辑执行方式相似。Logix5000 控制器支持一个连续型任务,但不是必须有连续型任务。用户能够组态是否在连续型任务执行结束时刷新输出模块。用户能够将连续型任务修改成周期型任务或者事件型任务。

CPU 将连续型任务和系统内务处理之间的时间分成时间片。在用户任务和系统内务处理任务之间的每个任务的转换需要另外的 CPU 时间来加载并重新存储任务信息。

无论系统内务处理的时间片如何, 16 版本的 RSLogix5000 软件对于连续型任务强制执行至少 1ms。这样有利于系统资源更有效率的使用, 因为允许更少的连续型任务执行次数意味着任务转换更加频繁。

组态周期型任务

周期型任务按照预先组态好的时间间隔自动执行。这个任务与 PLC-5 和 SLC500 处理器中的可选择定时中断相类似。用户能够组态任务是否在周期型任务结束时刷新输出模块。在任务执行之后, 直到组态时间间隔完成才能重新执行。

如果用户的应用项目需要进行大量的通讯处理(例如, message 指令或者 RSLinx 通讯), 推荐使用周期型任务而不是连续型任务。这会避免与任务转换有关的内务处理, 能够改善系统性能。

组态事件型任务

如果事件触发或者事件没有在指定的时间间隔内触发, 事件型任务自动执行。用户能够组态任务是否在事件型任务结束时刷新输出模块。在任务执行之后, 直到事件重新触发任务才能重新执行。每个事件型任务需要指定的触发。

触发	描述
模块输入数据状态改变	根据模块的状态改变(COS)组态, 输入模块(数字量或者模拟量)触发事件型任务。对于模块上的一点, 使能 COS。如果用户对多点使能 COS, 会发生事件型任务重叠。 ControlLogix 的顺序事件模块(1756-IB16ISOE 和 1756-IH16ISOE)使用使能 CST 捕获特性代替使用 COS。
消费者标签	只有一个消费者标签能够触发指定的事件型任务。使用生产控制器的 IOT 指令发送信号给新数据的生产。
轴注册 1 或者 2	注册输入触发事件型任务。
轴监控	监控位置触发事件型任务。
运动组执行	运动组的粗略刷新周期触发运动规划和事件型任务的执行。因为运动规划中断所有任务, 它是首先执行的。
EVENT 指令	多个 EVENT 指令能够触发相同的任务。

有关事件型任务的更多信息，请参阅：

- 《Logix5000 控制器通用过程编程手册》，出版物 1756-PM001。
- 《使用 Logix5000 控制器的事件型任务》，出版物 LOGIX-WP003。

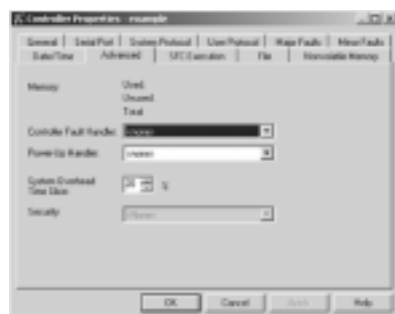
组态事件型任务指南

指南	描述
I/O 模块用来触发控制器相同框架上的事件。	在远程框架上的 I/O 模块添加额外的网络通讯和处理响应时间。
将数字量输入的事件限制在模块的单个输入位。	模块上所有的输入触发一个事件，所以使用多个位会增加任务重叠的可能。组态模块来检查触发输入并关闭其它位的状态改变。
设置控制器的最高优先级作为事件型任务的优先级。	如果事件型任务的优先级低于周期型任务的优先级，事件型任务将必须等待周期型任务执行完成后才能够执行。
限制事件型任务的数量	增加的事件型任务数量减少可用的 CPU 带宽并增加任务重叠的可能。

事件型任务其它的注意事项

注意事项	描述
事件型任务代码数量	每个逻辑组件(例如，梯级、指令或者结构文本)会增加扫描时间。
任务优先级	如果事件型任务不是最高的优先级任务，较高的优先级任务可能延时或者中断事件型任务的执行。
CPS 和 UID 指令	如果两个指令之一被使用，事件型任务不能中断当前正在执行的任务。(这些任务中使用 CPS 和 UID 指令)。
通讯中断	无论任务的优先级，串口通讯中断任务。
趋势图	趋势图数据采集优先于事件型任务。
输出处理	用户能够在任务结束时禁止输出处理以减少任务处理时间。16 版本 RSLogix5000 软件的控制器管理器显示是否有输出处理被禁止。

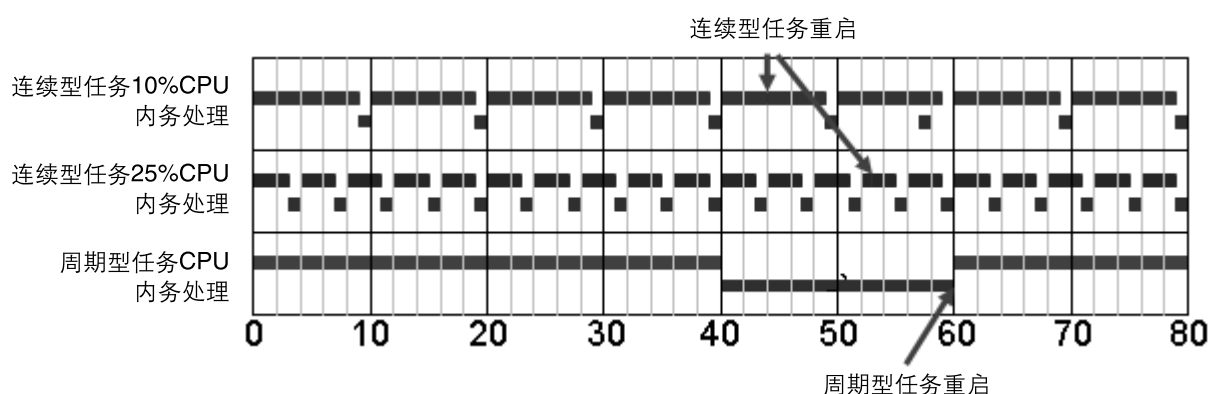
选择系统内务处理百分数



系统内务处理时间片指定了专用于通讯和后台冗余功能的连续型任务执行时间的百分数。系统内务处理功能包括以下内容：

- 编程通讯与 HMI 设备通讯(例如, RSLogix5000 软件)
- 响应消息
- 发送消息
- 串口消息和指令处理

控制器执行一次系统内务处理功能需要至少 1ms 的时间。如果控制器完成内务处理功能少于 1ms，它会重新开始执行连续型任务。下图比较连续型任务和周期型任务。



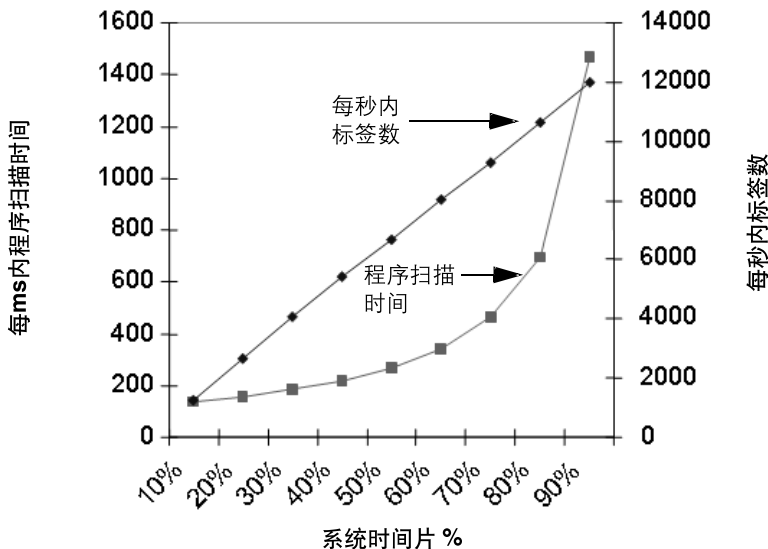
例子	描述
连续型任务 10%CPU 内务处理	在上面的例子中，系统内务处理时间片设置成 10%。设置代码执行时间为 40ms，连续型任务在 44ms 内执行完成。在 60ms 时间内，控制器用 5ms 进行通讯处理。
连续型任务 25%CPU 内务处理	将系统内务处理时间增加到 25%，控制器在 57ms 内完成连续型任务扫描，用 60ms 内的 15ms 时间进行通讯处理。
周期型任务	周期型任务中的相同代码给通讯处理预留更多的时间。下面的例子假设代码是 60ms 周期型任务。代码执行完成并在 60ms 停止，根据时间来触发。当任务停止时，所有的 CPU 带宽能够全部用来通讯。由于代码只用 40ms 进行执行，控制器能够用 20ms 进行通讯处理。取决于在 20ms 时间内通讯处理的数量，任务会有一些延迟来等待系统中的其它模块处理完成所有通讯的数据。

Logix5000 CPU 将连续型任务和系统内务处理之间的时间分成时间片。用户任务和系统内务处理之间的每个任务转换需要额外的CPU时间来加载并重新存储任务信息。用户能够使用下面的公式计算连续型任务的时间间隔：

$$\text{连续时间} = (100 / \text{系统内务处理时间片 \%}) - 1$$

管理系统内务处理
时间片百分数

当系统内务处理时间片百分数增加时, 分配给执行连续型任务的时间增加。如果没有控制器的通讯需要管理, 控制器使用通讯时间来执行连续型任务。

需要考虑的问题	描述																														
影响通讯和扫描时间	当增加通讯执行时，增加系统内务处理时间片百分数减少连续型任务的执行时间。 增加系统内务处理时间片百分数也能增加用来执行连续型任务的时间－增加全部的扫描时间。  <table><caption>Estimated data from the graph</caption><thead><tr><th>系统时间片 %</th><th>程序扫描时间 (ms)</th><th>每秒内标签数</th></tr></thead><tbody><tr><td>10%</td><td>150</td><td>1500</td></tr><tr><td>20%</td><td>180</td><td>2500</td></tr><tr><td>30%</td><td>220</td><td>3500</td></tr><tr><td>40%</td><td>280</td><td>5000</td></tr><tr><td>50%</td><td>350</td><td>7000</td></tr><tr><td>60%</td><td>450</td><td>9500</td></tr><tr><td>70%</td><td>580</td><td>12500</td></tr><tr><td>80%</td><td>750</td><td>16000</td></tr><tr><td>90%</td><td>1400</td><td>13500</td></tr></tbody></table>	系统时间片 %	程序扫描时间 (ms)	每秒内标签数	10%	150	1500	20%	180	2500	30%	220	3500	40%	280	5000	50%	350	7000	60%	450	9500	70%	580	12500	80%	750	16000	90%	1400	13500
系统时间片 %	程序扫描时间 (ms)	每秒内标签数																													
10%	150	1500																													
20%	180	2500																													
30%	220	3500																													
40%	280	5000																													
50%	350	7000																													
60%	450	9500																													
70%	580	12500																													
80%	750	16000																													
90%	1400	13500																													
系统内务处理时间片的未使用部分	使用 16 版本的 RSLogix5000 软件，用户能够组态任何未使用的系统内务处理时间片来： - 运行连续型任务，会使应用项目代码执行得更快并增加程序扫描的可变性。 - 处理通讯，会使连续型任务的扫描时间具有更多的可预见性和确定性。																														

个别的应用项目可能不同, 但是总体上对通讯和扫描时间的影响是相同的。上面的数据是基于包含 5000 个标签(没有数组或者自定义结构体)的 ControlLogix5555 控制器运行在连续型任务中。

开发例程中的应用代码

每个例程包含一种编程语言逻辑。选择一种基于应用项目的编程语言。

通常情况下，如果代码的内容代表	则选择这种编程语言
多个操作(不是按照顺序)连续或者并行执行	梯形图逻辑(LD)
布尔型或者基于位的操作	
复杂的逻辑操作	
消息和通讯处理	
机械互锁	
服务或者维护人员的操作可能必须要中断以便进行机械或者过程的故障处理	
伺服运动控制	
连续过程和变频器控制	功能块(FBD)
回路控制	
计算电路流	
多个操作的高级管理	顺序功能图(SFC)
重复顺序的运行	
批处理过程	
运动控制先后顺序(通过带嵌入结构文本的顺序功能图)	
状态机操作	
复杂数学操作	结构文本(ST)
指定的数组或者表的循环处理	
ASCII 字符串或者协议处理	

编程语言比较

比较	梯形图逻辑	功能块	顺序功能图	结构文本
指令种类	- 布尔型 - 通用运算和三角函数 - 定时器和计数器 - 数组管理 - 诊断 - 串口与消息 - ASCII 处理 - 专用的 CPU 控制 - 运动控制	- 通用运算和三角函数 - 定时器和计数器 - 位逻辑 - 高级过程控制 - 高级变频器控制	- 嵌入式结构文本的 步序动作 - 结构文本比较的转换 - 同步的可选择分支 - 停止符	- 通用运算和三角函数 - 定时器和计数器 - 位逻辑 - 数组管理 - 诊断 - 串口与消息 - 专用的 CPU 控制 - 运动控制 - 高级过程控制 - 高级变频器控制
编辑格式	- 图形梯级 - 不限制梯级	- 图形自由形式的制图 - 不限制页	- 图形自由形式的制图 - 不限制方格区域	- 文本的 - 不限制行
监控	- 梯级动画 - 数据值动画 - 强制状态	输出和输入引脚数据 值动画	- 激活步序动画 - 自动显示拖动条 - 分支和转换强制状态	- 标签监控面板 - 文本颜色
注释	- 标签 - 梯级	- 标签 - 文本框	- 标签 - 文本框 - 储存在 CPU 中的嵌入 结构文注释本	- 多行 - 如果行结束 - 储存在 CPU 中的注释

与子例程传递参数的指南

指南	描述
输入和返回参数取决于子程序的逻辑。	如果子程序需要任何返回参数先前的状态(在工程的其它场合需要使用这个值),这些值应该是输入参数: - 如果子程序包含锁存 / 非锁存逻辑(保持电路), 子程序预期的输出应该送入子程序并从子程序返回。 - 如果子程序不包含锁存 / 非锁存逻辑, 子程序预期的输出只需要从子程序返回。
将全部的定时器送入子程序然后从子程序返回。	如果子程序需要定时器, 将全部定时器标签送入子程序作为输入, 并将返回的全部定时器标签作为输出。将缓存标签中的定时器存储在子程序外部。

用户自定义 Add-On 指令的指南

用户自定义 Add-On 指令是用户创建的指令, 用于封装代码和本地数据。

指南	描述
为了在项目中反复调用, 使用 Add-On 指令来创建代码的标准模块。例程进行管理。	使用 Add-On 指令来: - 封装指定的或者强制性的操作, 例如电机或者阀门的动作。传送带或者容器动作最好使用 - 创建基于控制器指令的扩展。例如, 创建 Add-On 指令来执行 Logix5000 控制器中不可用的 SLC500 或者 PLC 控制器指令。 - 使用一种编程语言封装的指令可以在另一种编程语言中使用。例如, 创建一个功能块指令能够在梯形图是使用。
在梯形图、功能块或者结构文本编程语言中创建 Add-On 指令。	支持所有 Add-On 指令和固有的指令。JSR/SBR/RET、JXR、FOR/BRK(梯形图)、SFR、SFP、SAR、IOT 和 EVENT 指令除外。 在 Add-On 指令中的 GSV/SSV 指令不能涉及模块、消息、轴、运动组或者同类系统名称。 Add-On 指令支持功能块、梯形图和结构文本编程语言。每个 Add-On 指令逻辑范围是任何编程语言。例如, 主逻辑是功能块, 预扫描逻辑是梯形图。 用户能够嵌套 Add-On 指令多达七级

指南	描述
Add-On 指令支持以下参数: • 输入(拷入) • 输出(拷出) • InOut(由引用给定)	• 限制总数为 512: 输入参数 + 输出参数 + 本地标签(在 InOut 参数的数量上无限制) • 2MB 最大数据场合(参数与本地) • 报警、轴、轴控制组、坐标系、消息、运动组和生产者 / 消费者标签必须在程序或者控制器作用域内, 并作为 InOut 参数 • 给定包括控制器标签、程序标签和立即数标签。 • 输入参数与输出参数对原始数据类型限制(BOOL, SINT, INT, DINT, REAL)。对于 LINT、用户自定义的和结构数据类型, 使用 InOut 参数 • DINT 数据类型提供最佳的执行效果。 • 当标签已经创建为指令的数据类型时, 参数缺省值和本地标签用来初始化数据结构。当存在的参数或者本地标签的缺省值已经修改时, 存在的标签表明指令没有刷新。当参数或者本地标签添加到指令定义中时, 标签的缺省值用在已存在的标签中。
只能在离线时创建和修改	在线运行支持监控。 修改 Add-On 指令必须离线。修改 Add-On 指令会作用于所有的应用程序。
Add-On 指令像例程一样执行。	有更高执行优先级的任务能中断 Add-On 指令。使用 UID/UIE 指令串能够中断任务转换来校验在转换为另一个任务时 Add-On 指令完全执行。
Add-On 指令内的代码只能够通过参数或者本地自定义访问指定的数据。	如果用户想要编程时在 Add-On 指令外部访问本地数据, 可以将本地数据复制为参数。
使用可选的扫描模式逻辑来设置、初始化或者复位 Add-On 指令代码。	Add-On 指令能够有除了主逻辑之外的逻辑。 • 在控制器启动时预扫描逻辑执行 • 在 SFC 自动复位时后扫描逻辑执行 • 当梯级为假时使能假逻辑执行

子程序和 Add-On 指令的比较

比较	子程序	Add-On 指令
可访问性	程序中(多次复制)	控制器的任何地方(单独复制)
参数	通过数值	通过数值或者通过 InOut 给定
数字型参数	无变换, 用户必须管理	对于输入和输出参数的自动数据类型变换 InOut 参数必须与确定的类型精确匹配
参数数据类型	原始数据类型、数组数据类型、结构体数据类型	- 原始数据类型作为输入参数或者输出参数 - LINT、用户自定义和结构体数据类型作为 InOut 参数
参数校验	无, 用户必须管理	确认校验
数据封装	程序或者控制器作用域内所有的数据 (可访问任何位置)	本地数据被隔离(只能够访问指令内部)
监控 / 调试	多次调用的包含混合数据的逻辑	单独调用的数据逻辑
支持编程语言	FBD、LD、SFC、ST	FBD、LD、ST
可以调用	FBD、LD、SFC、ST	FBD、LD、SFC、ST
保护	锁定只能查看	锁定只能查看
文件	例程、梯级、文本、命令行	指令描述、版本信息、用户、梯级、文本、命令行、扩展帮助
执行性能	- JSR/SBR/RTN 增加内务处理 - 所有的数据被复制	- 调用更加有效 - 给定 InOut
内存使用	非常小	- 调用需要较多的内存 - 所有给定需要额外的内存
编辑	代码和数据都能够在运行的控制器中离线和在线修改	项目文件离线才可以修改代码并需要重新下载 相关的数据值能够在线和离线修改
输入 / 输出	所有的例程由全部的项目.L5K 文件输入和输出 (受保护的例程可能排除在外或者被加密) 单独的 LD 梯级、给定和标签 /UDT 能够通过. L5K 文件输入 / 输出	所有的 Add-On 指令能够在全部的项目.L5K 文件中输入 / 输出(受保护的例程可能排除在外或者被加密) 单独的 Add-On 指令定义和代码能够通过.L5K 文件输入 / 输出

编程方法

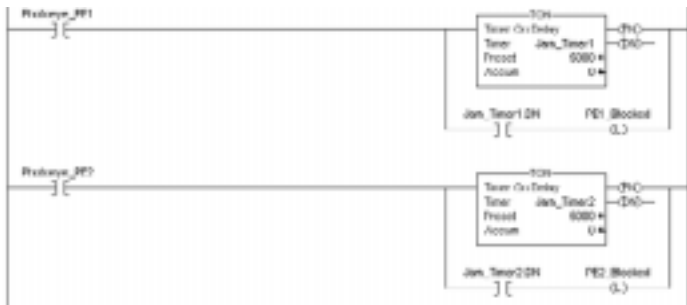
Logix5000 控制器能够使用不同的编程方法。在选择编程方法时需要权衡考虑。

内嵌副本

优势

- 使用更多内存
- 很快的执行时间，因为所有标签引用在运行时间前都被定义
- 很容易维护，因为梯级动画匹配标签值
- 需要更多的时间进行创建和修改

使用不同的标签引用，编写多次拷贝的代码。

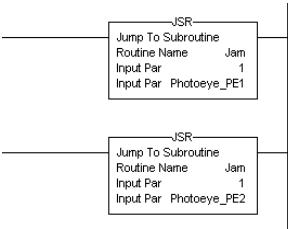


索引例程

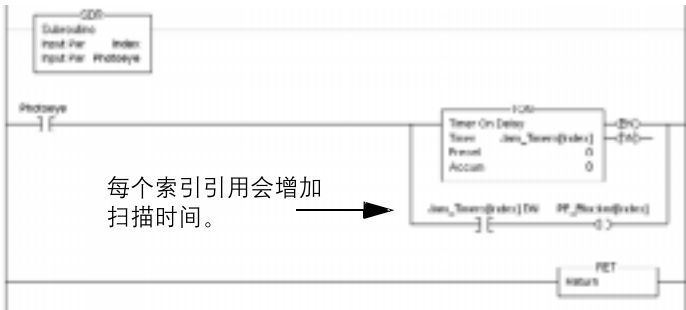
优势

- 拷贝代码比开发更快
- 最短的执行时间，因为所有标签引用都在运行时间内计算
- 难于维护，因为数据监控与执行不同步

编写拷贝代码并对存储在数组中数据使用索引引用。



JSR指令传送索引。

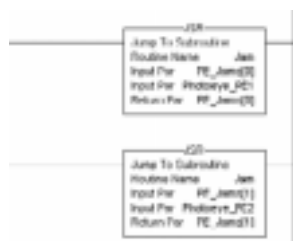


每个索引引用会增加扫描时间。

缓存例程

优势

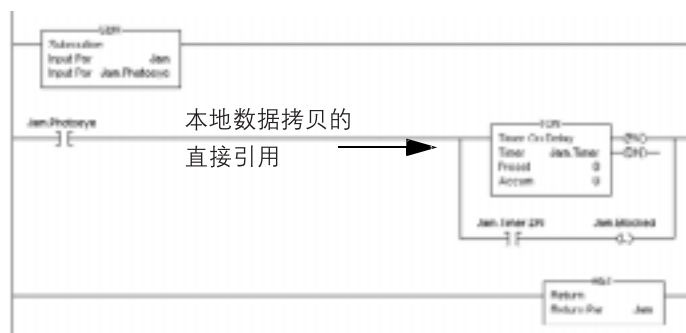
- 一个拷贝操作比多次索引偏移量要快得多
- 无需在运行时计算数组的偏移量
- 代码的数量增加，但是优势也很明显
- 难于维护，因为监控数据与执行不同步



JSR指令传送所有的控制数据



用户自定义结构体
打包控制数据



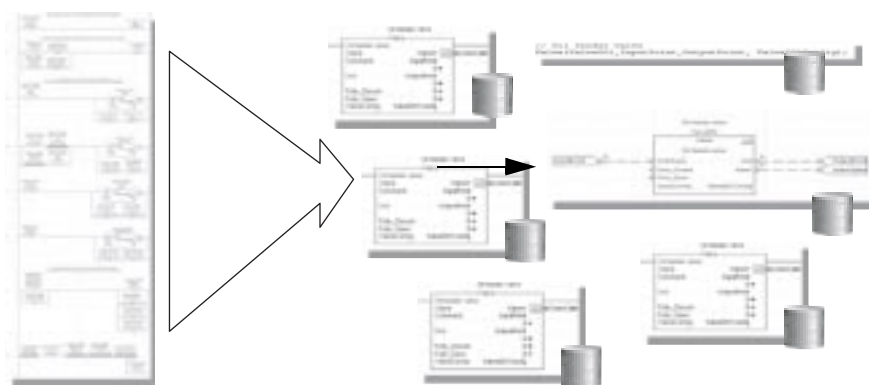
优势

- 一个拷贝操作比多次索引
- 用户自定义指令可以重复使用在一个或多个项目中
- 封装例程类型的代码，可以使用多次
- 每条指令的使用都有其本身的后台数据

Add-On 指令

Add-On 指令将逻辑封装在可重复使用的用户自定义指令中。

通过项目重复使用Add-On指令



逻辑的控制器预扫描

上电时，控制器使用初始化指令预扫描逻辑。控制器复位所有状态指令，例如输出值(OTE)和定时器(TON)。一些指令也在预扫描期间执行操作。例如，ONSR 指令关掉存储位。预扫描的信息，请参阅：

- 《Logix5000 控制器通用指令参考手册》，出版物 1756-RM003。
- 《Logix5000 控制器过程控制和驱动指令参考手册》，出版物 1756-RM006。

在预扫描期间，输入值不是当前值而且输出值也无法写入。

预扫描影响	描述
梯形图逻辑	控制器复位非保持型的 I/O 和内部值。
功能块逻辑	除了复位非保持 I/O 和内部值，控制器为每个功能块清空使能输入(EnableIn)参数。
结构文本逻辑	控制器复位位标签并强制数字标签归零(0)。 使用相等的赋值运算符(:=)来强制预扫描时复位数值。 如果用户想要标签保留在它最后的状态，使用不等的(:=)赋值运算符。
顺序功能图逻辑	嵌入的结构文本遵循以上同样的规则。
数组索引值	在预扫描时数组索引值能够使控制器故障。如果数组索引值大于数组的维数，控制器在预扫描时会检测出一个主要故障。为了避免这种情况出现，在预扫描时要确保该索引设置正确或者使用故障例程来处理这个错误。请参阅 46 页的数组索引的预扫描。

预扫描不同于第一次扫描，在预扫描时控制器不执行逻辑。第一次扫描时控制器执行逻辑。控制器在以下时刻置位 S:FS 进行一次扫描：

- 在预扫描之后的第一次扫描。
- 当程序未禁止时的第一次扫描。
- 每次步序是第一次扫描(当步.FS 被置位时)。从上一步的第一次扫描时的逻辑动作执行情况，用户能够看到 S:FS 位被置位。

Add-On 指令预扫描逻辑

Add-On 指令预扫描逻辑在主逻辑执行预扫描模式后执行。在执行前使用预扫描逻辑初始化标签值。例如，在第一次执行前将PID 指令设置为0%输出的手动模式。

- 当 Add-On 指令执行在预扫描模式下时，任何要求的参数需要传送如下数值：
- 指令调用中的自变量值传入输入参数。
 - 自定义在指令调用中的自变量传出到输出参数。

SFC 逻辑的控制器后扫描

- SFCs支持自动复位选项,当转换指示步序完成时复位选项执行相关步序动作的后扫描。同样，每个跳转到子程序(JSR)指令使控制器后扫描已调用的例程。在后扫描期间：
- 输出(OTE)指令关闭同时非保持定时器复位。
 - 在结构文本代码中，使用相等的赋值运算符([:=])复位标签。
 - 在结构文本代码中,使用不等的赋值运算符(:=)使标签保持在最后的状态。

Add-On 指令预扫描逻辑

当Add-On指令被SFC动作的逻辑调用时，同时自动复位选择复位，Add-On指令在后扫描模式执行。主逻辑在后扫描模式执行之后，Add-On指令后扫描例程执行。使用后扫描逻辑复位内部状态和状态值或者当SFC动作完成时禁止指令输出。

定时器的执行

PLC、SLC和Logix5000控制器的定时器在每次扫描时存储一部分实时时钟。下次扫描时，定时器将当前时钟与存储时钟值进行比较，通过差值调整ACC的值。

PLC/SLC 控制器	Logix5000 控制器
在 PLC/SLC 控制器中，定时器以 10ms/ 位存储 8 位。 这使得在定时器重叠前有 2.56 秒(2**8/100)的余量。 如果程序执行跳过定时器，程序中定时器好像暂停一样。 实际上，定时器发生溢出。当定时器逻辑下一次执行时，丢失的时间在 0 到 2.56 秒间变化。	Logix5000 控制器使用 32 位数据，所以有更多的空间来存储时间。定时器以 1ms/ 位存储 22 位，这相当于 69.905 分(2**22/1000 毫秒每分 /60 秒每分)。 如果控制器执行跳过定时器，它比 PLC/SLC 控制器需要更长时间才能发生溢出。 当定时器代码下次执行时这导致流失时间上更大的跳转。

程序执行可以跳跃正在执行的定时器是由于：

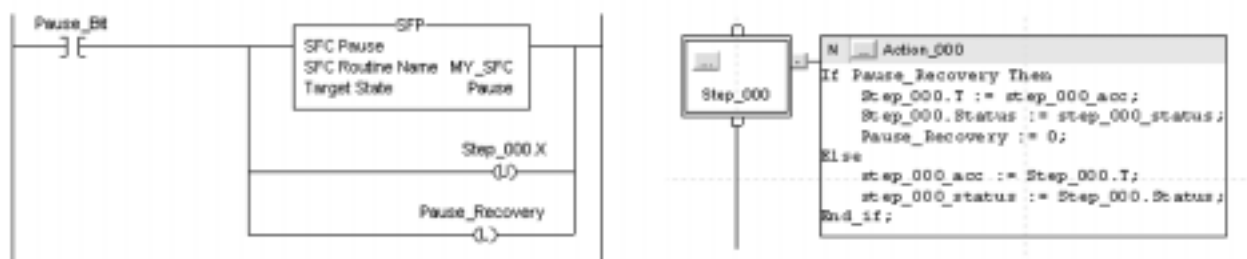
- 子程序没有被调用
- 跳转超过代码
- SFC 动作
- 非激活的 SFC 步序
- 事件型任务或者周期型任务没有执行
- 设备阶段(Phase)状态例程

SFC 步序定时器的执行

每次步序执行时，SFC 定时器存储时钟时间。在后来的步序扫描中，控制器将当前的时钟时间与最后的扫描比较并通过差值更新步序定时器的 ACC。

当暂停 SFC 然后再释放 SFC 时，步序定时器在暂停期间向前跳转。如果用户想要步序定时器在暂停时保持在它的位置：

- 当顺序功能图暂停释放时锁存恢复位。
- 增加一个步序动作用来存储步序定时器的 ACC 值并当暂停恢复位置位时重新存储该值。



在线编辑 SFC

13 固件版本增加了支持在线编辑 SFCs 的功能。当在线编辑 SFC 时，软件首先离线修改工程。确定修改时，将它们下载控制器。如果用户将控制器转换为测试或者非测试编辑时，控制器复位 SFC 并在初始步序开始执行。如果用户在线编辑 SFC：

- 当用户测试或者非测试编辑时，设计与 SFC 执行初始步序相符。
- 定位子程序中的结构文本逻辑最小化在线编辑的影响。
- 使用 SFR 指令来编程转换 SFC 执行到预期的步序。

在某些情况下，这会导致 SFC 与设备不同步。如果需要的话，初始步序的编程逻辑校验最后的状态并使用 SFR 指令修改成适当的步序。一种方法是在每个步序动作中设置索引编号。当重新启动时，使用 SFR 指令跳转到基于索引值的适当的步序。

注释:

地址数据

简介

Logix5000 控制器支持 IEC 61131-3 基本数据类型, 如 BOOL、SINT、INT、DINT、LINT 和 REAL。控制器也支持混合数据类型, 如数组、预定义结构体(如计数器和计时器)和用户自定义结构体(UDT)。

数据类型	说明	
基本数据类型 (BOOL、SINT、INT、DINT、REAL)	优点	要点
	- 独立名称 - 标签的数量没有限制 - 标签编辑器和数据监视器可以过虑个别的标签, 并且可以告诉您它们是否被引用了 - 在标签编辑器和数据监视器中始终按字母顺序排列 - 完整的别名标签支持(标签的根以及它的每一位) - 可在线添加 - 在 AOI 中支持输入或输出参数	- 每个标签使用 32 位的内存 - 相对于混合数据类型, 需要进行更多的额外通讯和更多的控制器内存 - 只能在离线时改变一个标签的数据类型 - 在标签编辑器和数据监视器中按字母顺序排列根标签, 按照结构体定义顺序排列结构体成员
特殊用途基本数据类型 (LINT)	优点	要点
	- 用 64 位整型数值存储数据和时间值 - 数据监视器显示数据和时间基数, 允许显示 LINT 类型数值, 如年、月、日、时间、分钟、秒和微秒	- 支持的指令仅限于: GSV、SSV、ALMD、COP 和 CPS - 进行数学运算和比较时, 先将 LINT 类型数值拷贝到两个 DINT 类型中, 再通过代码进行操作 - 在 ADD-On 指令中, 仅限于作为 InOut 参数。
混合数据类型 (数组、结构体)	优点	要点
	- 允许使用指定的名称和用户自定义的结构 - 在控制器内存中组合信息 - 优化通讯时间以及内存的影响 - 对数组进行动态寻址 - 可以在线创建新的数组标签 - 可以创建别名标签指向用户自定义结构体, 数组成员, 以及成员的某位	- 每个用户自定义结构体或数组限制为 2MB 数据 - 用户自定义结构体可能会进行填补, 这会强制 32 位数据排队成列 - 别名标签不能指向数组的根标签 - 标签编辑器和数据监视器的过滤功能有限 - 只能在离线时创建或修改用户自定义结构体 - 只能在离线时更改数组 - 在 ADD-On 指令中, 仅限于作为 InOut 参数。

Logix CPU 读取和操作 32 位数据值。所有数据从 32 位偏移量开始，因此分配给标签的最小内存为 4 字节。当创建了存储数据小于 4 字节的独立标签后，控制器为其分配 4 字节内存，但是数据仅仅占用了该内存部分空间。

数据类型	字节 3	字节 2	字节 1	字节 0						
	31...0	31...0	31...0	31	16	15	8	7	1	0
BOOL	未用	未用	未用	未用						0 或 1
SINT	未用	未用	未用	未用					-128...127	
INT	未用	未用	未用	未用			-32,768...32,767			
DINT	未用	未用	未用	-2,147,483,648...2,147,483,647						
REAL	未用	未用	未用	-3.40282347E ³⁸ ...-1.17549435E ⁻³⁸ (负数) 0 1.17549435E ⁻³⁸ ...3.40282347E ³⁸ (正数)						
LINT	有效日期 / 时间范围从 1/1/1970 12:00:00 AM UTC(通用时间)到 1/1/3000 12:00:00 AM UTC									

操作 SINT 或 INT 数据时，控制器首先将数值转换成 DINT 类型，执行编程操作，再将结果以 SINT 或 INT 数值返回。在执行相同操作时，与 DINT 类型数值相比较而言，SINT 和 INT 数据需要额外的内存和执行时间。

数据类型指南

指南	说明
在可能的情况下，尽量使用 DINT 数据类型	Logix5000 控制器执行 DINT(32 位)和 REAL(32 位)数学操作。DINT 比其它数据类型使用更少的内存和执行时间。用途： • DINT 适用多数数值和数组索引 • REAL 适用于浮点型、模拟量数值 • SINT(8 位)和 INT(16 位)主要用于用户自定义结构体或与不支持 DINT 数值的外部设备通讯

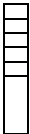
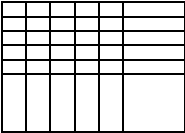
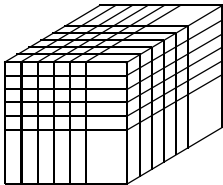
	SINT	INT	DINT	REAL
为独立标签保留的内存	4 字节	4 字节	4 字节	4 字节
为用户自定义结构体中的数据保留的内存	1 字节 (一行连续的 8 位)	2 字节 (一行连续的 16 位)	4 字节 (一行连续的 32 位)	4 字节 (一行连续的 32 位)
用于存取 ADD 指令中标签的内存	236 字节	260 字节	28 字节	44 字节
1756-L63 控制器执行 ADD 指令所需的执行时间	3.31μs	3.49μs	0.26μs	1.45μs

将 BOOL 数值组合成数组	使用 BOOL 数值时，将它们组合成 DINT 数组，优化控制器内存的使用，以及使 FBC 和 DDT 指令实现位操作
----------------	---

数组

数组分配了连续的内存块，将特殊的数据类型存储为数据表。

- 标签支持一维、二维或三维数组。
- 用户自定义结构体可以包含单维数组元素。

数组	存储数据如下	示例				
一维		标签名称	类型	维数0	维数1	维数2
		<i>one_d_array</i>	DINT[7]	7	--	--
		元素总数=7 有效下标范围 DINT[x], 其中 x=0...6				
二维		标签名称	类型	维数0	维数1	维数2
		<i>two_d_array</i>	DINT[4,5]	4	5	--
		元素总数 = 4 * 5 = 20 有效下标范围 DINT[x,y], 其中 x=0...3; y=0...4				
三维		标签名称	类型	维数0	维数1	维数2
		<i>three_d_array</i>	DINT[2,3,4]	2	3	4
		元素总数 = 2 * 3 * 4 = 24 有效下标范围 DINT[x,y,z], 其中 x=0...1; y=0...2; z=0...3				

数组数据类型决定了怎样使用连续的内存块。

BOOL[96]=12 字节内存

BOOL 数组使用 32 位的内存增量

3	3	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0		
1	0	9	7	7	6	5	5	4	3	2	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	
6	6	6	6	6	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	2	
3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2
9	9	9	9	9	9	8	8	8	8	8	8	8	8	8	8	8	8	7	7	7	7	7	7	7	7	7	6	6	6	6	6
5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4

SINT[10]=12 字节内存(2 字节没使用)

SINT 数组会在结尾处进行填补，以便消耗掉在左侧剩余的字节

3	2	1	0
7	6	5	4
未用	未用	9	8

INT[5]=12 字节内存(2 字节没使用)

INT 数组会在结尾处进行填补，以便消耗掉在左侧剩余的字节

1	0
3	2
未用	4

DINT[3]=12 字节内存，REAL[3]= 字节内存

DINT 和 REAL 数组使用 4 字节的内存增量

0
1
2

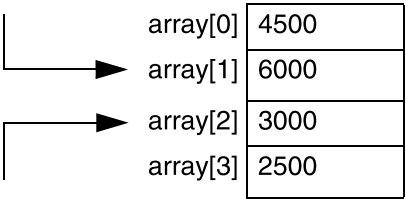
数组的指南

指南	说明	
多数数据类型均可创建数组，但以下数据类型除外：ALARM、AXIS、COORDINATE_SYSTEM、MOTION_GROUPH 和 MESSAGE	下标标识了数组内的各个元素。下标从 0 开始，以元素总数减 1 所得的数值结束(从 0 开始索引)。 • 单维数组占用更少的内存，并且比二维或三维数组的执行速度快。 • 直接引用数组元素的执行速度比间接引用的扫描速度快。 • 数组最大可为 2MB。 • 如果创建了结构体数组，根据结构体定义为每个元素分配内存。	
数组类型	优点	要点
一维	• 受固有文件指令更好的支持 • 完全支持用户自定义结构体和数组 • 间接引用的影响很小(扫描时间和内存) • 可以在线创建新数组	• 多个数组不能像在 PLC 或 SLC 处理器中一样间接引用(例如：N[N7:0]:5) • 文件指令不能直接支持 BOOL 数组 • 必须离线更改
二维或三维	• 可以为物理系统提供更精确的数据显示 • 可以用两维数组间接模仿 PLC 文件 / 字 • 可以在线创建新数组	• 间接引用的影响很大(扫描时间和内存) • 文件处理命令需要为文件指令使用额外的代码 • 必须离线更改
嵌套数组	文件指令为数组提供有限的支持。处理数组数据，创建用户自定义结构体，并将一个数组作为结构体成员。然后创建用户自定义结构体数据类型的数组标签。	
选择基于数据的数组数据类型，并选择处理该数据的指令	虽然SINT和INT数组能够将更多的数值压缩到一个给定内存区时，但与数组相关的每条指令需要额外的内存和扫描时间。	
数组最多存放 2MB 数据	数组最大内存为 2MB。当创建的数组过大时，软件将显示警告。即使 1.5…2MB 大小的数组是有效的，软件同样会显示警告。	
在线和离线编辑数组	在线和离线均可创建新数组。然而，仅可以在离线时修改现有数组的大小或数据类型。	

数组的间接寻址

使用数组下标的标签(间接寻址), 可以用指令存取数组中的不同元素。改变标签值, 即改变了逻辑相关的数组元素。

当 *index=1* 时, *array[index]*指这里



当 *index=2* 时, *array[index]*指这里

直接引用数组中的元素(例如 **MyArray[20]**), 比间接引用(例如 **MyArray[MyIndex]**)使用更少的内存, 同时扫描速度更快。也可以直接寻址标签的位(**MyDint.[Index]**)。

使用**DINT**标签进行间接寻址, 因为其它数据类型需要额外的转换, 并且执行速度慢。每次对数据的间接存取, 控制器都会重新计算数组的索引。如果多次存取一个指定数组元素, 先将数据从数组中拷贝到一个固定标签, 在后面的逻辑执行中使用这个标签。

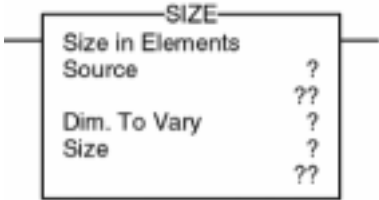
也可以使用表达式指定索引值。例如: **MyArray[10+MyIndex]**。

- 表达式使用运算符计算数值。
- 控制器计算表达式结果并将其作为索引使用。
- 如下所示为有效的运算符。

运算符	说明	最佳类型
+	加	DINT, REAL
-	减	DINT, REAL
*	乘	DINT, REAL
/	除	DINT, REAL
**	指数(xy)	DINT, REAL
ABS	绝对值	DINT, REAL
ACS	反余弦	REAL
AND	位与	DINT
ASN	反正弦	REAL
ATN	反正切	REAL
COS	余弦	REAL
DEG	弧度转换成角度	DINT, REAL
FRD	BCD 转换成整数	DINT

运算符	说明	最佳类型
LN	自然对数	REAL
LOG	以 10 为底的对数	REAL
MOD	取整	DINT, REAL
NOT	按位取补	DINT
OR	位或	DINT
RAD	角度转换成弧度	DINT, REAL
SIN	正弦	REAL
SOR	平方根	DINT, REAL
TAN	正切	REAL
TOD	整数转换成 BCD	DINT
TRN	舍位	DINT, REAL
XOR	按位异或	DINT

数组索引的指南

指南	说明
使用 SIZE 指令设定数组元素的个数。	在运行时设定运行时一个数组的元素个数，可以根据每个应用实例的要求调整代码值，重复写入代码。 
使用立即数引用数组元素。	SIZE 指令将元素个数返回。数组从 0 开始索引，所以最后一个元素的位置号为元素个数减 1。
使用 DINT 标签进行数组索引。	使用立即数引用数组元素比间接引用的处理和扫描速度快。
使用 DINT 标签进行数组索引。	DINT 标签的执行速度最快。SINT、INT 和 REAL 标签需要额外的转换代码，在操作过程中需要更多的扫描时间。
数组元素不能作为索引。	Logix5000 控制器不直接支持将一个数组元素作为索引，查询其它数组值。当需要进行这类操作时，可以为元素创建一个别名，将该别名作为索引使用。或者将元素拷贝到一个固定标签，将该标签作为索引使用。

数组索引的预扫描

在预扫描过程中，控制器会复位基本指令(例如，输出和计时器)的状态。一些由程序执行初始化的索引计算可能导致索引的地址超出范围，如果程序没有执行，则不能初始化索引。使用故障处理例程可以避免此情况的发生。

- 在第一个任务的第一个程序中，编写一个无条件的梯级，并且输出指令 OTE 引出一个内部位。预扫描时，预扫描位将关闭。正常扫描时，预扫描位会一直打开。
- 如果发生了索引地址超出范围故障和预扫描位关断，只需复位故障然后继续执行。

有关处理故障的信息和示例代码，请参阅 Logix5000 控制器程序编程手册，版本号 1756-PM001。

重要事项

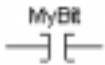
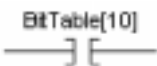
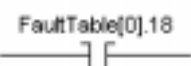
13和更高版本的控制器固件中没有预扫描条件。无需编写故障处理例程解决索引地址超出范围的故障。

用户自定义结构体的指南

指南	说明
组合一个结构体内相同数据类型的成员。	大多数数据类型都可以创建结构体成员，但以下数据类型除外：ALARM，AXIS，COORDINATE_SYSTEM，MOTION_GROUP，和 MESSAGE 数据类型。 按顺序存放相同数据类型的成员。 Logix5000 控制器为不同的数据类型分配了不同的存储位：SINT，8 位；INT，16 位；DINT 和 REAL，32 位；BOOL，8 位。如果 BOOL 型数据在用户自定义结构体内相邻排列，则将它们映射到同一字节内。
结构体只能使用一维数组。	只能将一维数组作为结构体成员。用户自定义结构体不支持多维数组。
必须将结构体使用的 I/O 数据拷贝到成员中。	如果结构体中包含表示 I/O 设备的成员，必须通过逻辑从相应的 I/O 标签中将数据拷贝给结构体成员。 确保结构体成员数据类型与 I/O 数据类型相同，避免数据类型转换。
用户自定义结构体成员个数限制为 500 个。	Logix5000 控制器将用户自定义成员个数限制为 500。如果有更多需要，可以在主结构体中使用嵌套结构。
用户自定义结构体存放数据限制为 2MB。	UDT 的最大内存为 2MB。如果创建了占用内存过大的 UDT，软件将显示警告。当 UDT 的范围在 1.5 到 2MB 时，虽然该范围是有效的，软件同样会显示警告。
当用户自定义结构体用于通讯时，对其大小有所限制。	生产者和消费者标签在进行背板通讯时限制在 500 字节内存，当进行网络通讯时限制在 480 字节内存。 RSLink 软件可以将用户自定义结构体优化到 480 字节范围内。
使用合适的指令将数据下载到结构体。	在程序的起始处将输入值加载到用户自定义结构体中，在程序的结尾处将输出值从用户自定义结构体中拷贝出来。 • 单独位 - 使用检查是否闭合(XIC)和输出线圈(OTE)指令 • 连续位 - 使用位域分配(BTD)指令 • 单一数值 - 使用文件拷贝(COP)指令 • 多个连续数值 - 使用同步拷贝(CPS)指令
使用结构体注释自动创建标签注释。	使能 Pass-through Description 工作站选项(Tools>Options>Display)，显示使用结构体数据类型标签的说明，这些说明是在该结构体成员中添加的。
在线和离线编辑。	可以在线或离线创建用户自定义结构体。但是，只能在离线时修改现有结构体。

为位类型标签选择数据类型

位类型可以在Logix5000控制器中以如下形式存在: BOOL 标签, BOOL 数组中的位, SINT, INT, DINT 数组元素中的位, 用户自定义结构体成员, 或用户自定义结构体中 SINT, INT, DINT 成员中的位。

标签类型	说明	
BOOL 标签 MyBit:BOOL 	每个标签存取一个指定的位。每个标签使用 4 字节。	
	优点 • 每位有一个指定的标签	要点 • 需要额外的带宽进行通讯 • 使用较多的内存: 每位 32 位 • 不能使用 FBC/DDT 位文件指令
BOOL 数组 BitTable:BOOL[32] 	BOOL 数组将多个位组合到相邻的字中(32 位的字)。	
	优点 • 将多个位组合到一个单独的字中 • 更好的使用内存 • 使用间接寻址可以访问数组中的所有位	要点 • 只有位指令支持 BOOL 数据类型 • 不可以使用文件指令、拷贝指令或 DDT/FBC 指令
DINT 数组 FaultTable:DINT[3] 	DINT 将多个位组合到相邻的字中。	
	优点 • 将多个位组合到一个单独的字中 • 文件指令、拷贝指令和 DDT/FBC 指令支持 DINT 数组 • 允许通过元素(字)和位号存取位	要点 • 需要额外规划间接寻址位 • 使用间接寻址很难访问数组的位
用户自定义结构体 BitStructure Bit1:BOOL Bit2:BOOL Fault:BitStructure 	用户自定义结构体将多个位组合到相邻的字中, 并独立命名。	
	优点 • 基于对象 • 将多个位组合到一个单独的字中	要点 • 第三方的 MMI/EOI 产品不直接支持结构体(RSView32 不支持 32 位标签和结构体) • 不能使用 FBC/DDT 位文件指令

连续的位地址 PLC-5 和 SLC 500 处理器内的 BOOL 数据表支持两种寻址模式，可以寻址到相同位。

寻址模式	说明
连续的位 在 PLC-5 或 SLC 软件中， 用位表示寻址模式	连续的位地址具有将所有位作为一个连续的位列表(数组)访问的能力。例如，如果要访问 B 文件第 2 个字的第 3 位，指定 B3/18 即可。这种模式与 Logix5000 控制器中的 BOOL 数组相似，在 Logix5000 中指定该位为 FaultBit[18]。
字的位 在 PLC-5 或 SLC 软件中， 用字 / 位表示寻址模式	字的位地址在一个字的范围内标识位。例如，B3:1/2 与连续的位地址 B3/18 相同。这种模式与 Logix5000 控制器存取 SINT，INT，DINT 数组的位相似，在 Logix5000 中该位指定为 FaultTable [1].2。

Logix5000 控制器支持上述两种寻址模式，但是不能使用两种模式访问同一个数组中的位，这是因为要遵循 IEC 61131-3 标准。选择最适合应用项目的方式进行访问。这两种方式也可以用来拷贝数组间的数据。

也可以使用表达式间接访问具有连续位号的 DINT 数组。例如：

```
Tag
  MyBits : DINT[10]
  BitRef : DINT
EndTag

MOV(34, BitRef)
XIC(MyBits[BitRef / 32] . [BitRef AND 31])
```

这里：

表达式	计算
[BitRef / 32]	DINT 数组的元素
如果标签 MyBits 是 INT 或 SINT 类型，除数分别为 16 或 8。	
[BitRef AND 31]	元素的位
如果标签 MyBits 是 INT 或 SINT 类型，掩码值分别为 15 或 7。	

诊断检测(DDT)和文件按位比较(FBC)指令将位号作为其操作结果。这些指令仅限于 DINT 数组，因此可以用它们查找上述示例返回的位号。

字符串数据类型的指南

字符串数据类型是保持ASCII码的结构体。结构体的第1个成员定义了字符串的长度，第2个成员是一个保持实际ASCII码的数组。

Name:emailString

Description:string type used for email source/destination data

Maximum Characters:512

Members:

	Name	Data Type	Style	Description
	LEN	DINT	Decimal	
	DATA	SINT[512]	ASCII	

Data Type Size: 516

指南	说明
创建的字符串数据类型的长度可以与默认值不同	默认的字符串数据类型包括82个字符，但是也可以根据实际需要创建自定义长度的字符串数据类型。
只有部分指令支持字符串数据类型	如下比较指令支持字符串标签：EQU，NEQ，GRT，GEG，LES，LEQ，CMP。 如下串行端口指令支持字符串标签：ARD，ARL，AWA，AWT。 如下字符串处理指令支持字符串标签：STOD，DTOS，STOR，RTOS，CONCAT，MID，FIND，DELETE，INSERT，UPPER，LOWER，SIZE。 如下文件指令支持字符串数组：FAL，FFL，FFU，LFL，LFU，COP，CPS，FSC。
使用 SIZE 指令设定字符串中字符的个数。	在运行时设定字符串中字符的个数，可以根据实例的需要调整代码值，重复写入代码。
使用 DTOS，RTOS 和 CONCAT 指令可以在字符串中插入标签值。	SLC 500 处理器具有在字符串中插入数据表访问地址的能力(在线间接访问)。SLC 500 AWA 和 AWT 指令可以查找数据值，并将一个 ASCII 表达式放入输出字符串中。Logix 5000 控制器不直接支持这种能力。使用 DTOS 或 RTOS 指令将数值转换成字符串，再用 CONCAT 指令将字符与其它字符串合并。
设置 LEN 字段指示当前字符个数。	在字符串结构中 LEN 字段指示字符串中有多少字符。RSLogix 5000 软件和控制器的操作字符串指令用 LEN 值决定字符串 DATA 数组中有多少位置包含有效字符。一旦达到 LEN 值，RSLogix 5000 软件和指令均停止对 DATA 数组的操作。

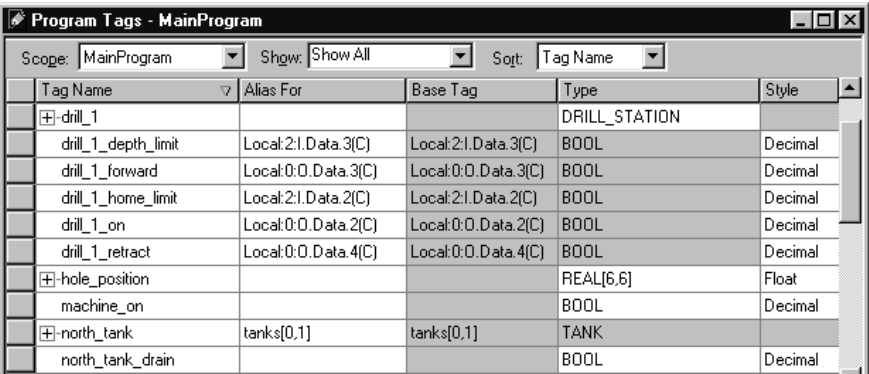
PLC-5/SLC 500 的字符串存取

PLC-5 和 SLC 500 处理器的 ASCII(A)数据表使用字符串格式，与 Logix 字符串数据类型相似。其最大的区别是 PLC-5/SLC 500 处理器的 LEN 字段(length)是16位的，即INT值，而Logix5000控制器的LEN字段(length)是32位的，即DINT值。这个区别可能会影响转换逻辑和数据通讯。Logix5000 控制器将 LEN 字段转换成适当的值和大小，而 PLC-5/SLC 500 用消息格式读写字符串。

组态标签

标签是控制器内存中存储数据的某一区域的文本性的名称。标签是分配内存、逻辑访问数据和监视数据的基本机制。

标签用途	选择类型
存储项目的逻辑应用值	Base
对现有标签的数据重新命名(帮助将较长的、预先确定的标签名简化，如 I/O 数据或用户自定义结构体)	Alias
向其它控制器发送(广播)数据	Produced
从其它控制器接收数据	Consumed



有关 I/O 标签的更多信息，请参阅第 5 章 “与 I/O 的通讯”。

基本型标签的指南

指南	说明
创建独立的基本标签。	控制器支持预先定义的、独立的标签。 - 基本标签直接列于标签编辑器和数据监视器中,通过浏览字母顺序列表可以轻松查找标签。 - 可以在线创建基本标签,但是只能离线修改数据类型。 当需要传送和处理更多的信息时,仅使用基本标签会影响 HMI 通讯性能。
创建用户自定义结构体	用户自定义结构体(数据类型)允许将数据组合以满足机械和工艺需求。 - 一个标签包含系统某一方面的全部数据。这样就能够保证将相关数据存放在一起并容易访问,而无需考虑这些数据的类型。 - 每个数据(成员)都有一个描述性名称。 - 可以用结构体创建具有相同数据规划的多个标签。 - 只能离线修改用户自定义结构体。 RSLinx 软件对用户自定义结构体的优化作用大于独立标签。
使用类似于文件的数组能够快速创建一组相似的标签。	数组在通用标签名称下创建多个数据类型的实例。 - 数组允许将使用相同数据类型和执行相似功能的一些标签组合。 - 可以根据数据表达方式组合一维、二维或三维数据。 - 只能离线修改数组。 - RSLinx 软件对数组数据类型的优化作用大于独立标签。 尽可能减少 BOOL 数组的使用。许多数组指令对 BOOL 数组不进行操作。这就增加了初始化和清除数组数据的难度。
利用程序域标签的优势。	如果需要多个相同名称的标签,在不同的程序域内定义每个标签(程序域标签)。这样可以在多个程序中重复使用标签名称和逻辑。 避免在控制器域和程序域内使用相同的标签名称。在程序域,当某个标签名称与控制器标签名称相同时,将不能够引用该名称的控制器域标签。
命名时混合使用大小写字母,并使用下划线字符。	尽管标签不区分字母大小写(大写字母 A 和小写字母 a 相同),但是混合使用大小写字母的命名易于阅读。例如, Tank_1 比 tank1 可读性强。
将字母顺序排列作为考虑因素。	RSLogix 5000 软件按字母顺序显示相同作用域的标签。为了更容易地监视相关联的标签,对希望存放在一起的标签以相似的初始字符命名。例如,标签名称 Tank_North 和 Tank_South 要优于 North_Tank 和 South_Tank。
当标签命名中涉及数字时,以 0 作为起始号。	RSLogix 5000 软件在标签编辑器和数据监视器中使用简单分类方式按照字母顺序排列标签名。这意味着软件显示标签 Tag1, Tag2, Tag11, 和 Tag12 时按照如下顺序: Tag1, Tag11, Tag12, 然后是 Tag2。如果希望按数字序号顺序显示标签,应分别将它们命名为 Tag01, Tag02, Tag11, 和 Tag12。

创建别名标签

创建别名标签，用来表示其它标签。

- 两个标签共享基本型标签定义的数据值。
- 当基本型标签值改变时，该标签的指向(别名)标签随之改变。

指南	说明
别名标签指向基本标签。	指定别名时，应避免： • 嵌套别名(不能用别名标签指向别名标签)。 • 多个别名指向同一个标签。 在上载过程中，软件反编译程序并使用物理内存地址确定哪个标签正在被代码引用。如果存在被引用的标签，则该基本标签的所有引用都恢复成别名。如果多个别名指向相同的标签，RSLogix 5000 软件必须假定它可以找到找到第一个别名标签(按字母顺序)。
别名标签不影响控制器的执行。	在下载过程中，程序被编译成机器可执行的代码和物理内存地址。需要在控制器内存中存放别名的名称，程序对引用别名标签的或对其相关的基本标签执行相同的操作。
从 RSLinx 软件中存取别名标签。	因为在RSLinx软件中别名标签作为一个独立的标签存在，指向复杂的数组或结构体的别名标签可能需要额外的通讯时间。当从RSLinx软件或其他HMI中引用标签时，其速度比直接引用基本型标签快。

数据作用域的指南



数据作用域定义了存取标签的位置。控制器作用域标签可以被所有程序使用。程序作用域标签只能被程序的代码使用。状态(phase)作用域标签只能被指定的设备程序段代码使用。

操作	作用域分配
在同一个项目的多个程序中使用相同的标签	控制器作用域(控制器标签)
消息(MSG)指令使用的标签	
生产者或消费者数据	
使用运动标签	
与 PanelView 端子通讯	程序作用域(程序标签)
在控制器的不同部分或过个程中多次使用相同的标签名称。	
由多个程序员编写逻辑，并将这些逻辑合并到一个控制项目中。	
	程序段作用域(程序段标签)

将机械部分或不同的站点程序段隔离成独立的程序或设备程序段，使用程序作用域或程序段作用域标签。这样：

- 隔离了各个程序和设备程序段。
- 防止标签名称冲突。
- 提高重用代码的能力。

标签名称的指南

指南	说明
创建具有描述性的名称并且要保持它们足够简洁。	标签名称可以为 1 到 40 个字符长度。 • 标签名称的每个字符使用 1 字节的控制器内存，占用 4 字节内存。 例如，1 到 4 个字符的标签名称占用 4 字节内存。5 个字符的标签名称占用 8 字节内存。 • 标签名称存储在控制器中。 • 使用结构体减少使用标签的数量和大小。 程序上载会保护标签名称。
创建命名规范。	根据电子制图或机械设计惯例开发标签命名规范。例如，Conv1_Full_PE101 结合了传感器的功能和光眼的编号。
在标签名称中使用正确的字符。	Logix5000 标签名称遵循 IEC 61131-3 标准。使用： • 字母 A 到 Z。 • 数字 0 到 9。 • 下划线字符()。 标签名称以字母开始(避免与逻辑表达式混淆)，剩余字符可以是任何符合要求的字符。
插入名称改善排序。	RSLogix 5000 软件按字母顺序显示标签。如果在标签名称中使用数字，在字母前插入 0 以便按照正确的顺序将名称排序。 例如，标签名称：TS1，TS2，TS3，TS10，TS15，TS20，TS30。 显示顺序：TS1，TS10，TS15，TS2，TS20，TS3 和 TS30。 将 0 插入到标签中，则显示顺序为：TS01，TS02，TS03，TS10，TS15，TS20，TS30。

标签的说明

RSLogix 5000 软件会搜索标签源，以便加载第一个可用的注释。这样可以减少需要输入的的注释数量，以及确保引用的标签具有可以显示的描述。

指南	说明	
RSLogix 5000 软件根据标签源显示标签注释。	标签类型	RSLogix 5000 软件显示的注释
	基本标签	该标签的注释是唯一可以显示的，标签可定义为 BOOL，SINT，INT，DINT 或 REAL。
	别名标签	首先是别名标签的注释，然后是基本标签。
	用户自定义结构体和 ADD-On 指令标签	结构体的所有成员都使用标签的描述，除非给特定的成员添写了注释。 例如，如果 MyTimer.DN 没有注释，那么它会使用 MyTimer 的注释。
	基本数组标签	- 所有对数组的引用都使用数组的注释，除非给特定的数组元素添写了注释。 例如，如果MyTable[10]没有注释，那么它会使用MyTable的注释。 - 所有对数组的间接引用都使用数组的注释。 例如：MyTable[Index]使用 MyTable 的注释。
	结构体数组标签	在数组中，对所有结构体成员的引用缺省使用数组的定义，除非给数组的结构体成员添写了注释。 例如，如果Table[0].Field1 没有特定的注释，那么它会使用Table的注释。

相关更多信息，请参阅《自动创建带用户自定义结构体类型的标签注释白皮书》，LOGIX-WP004。

注释:

与其它的控制器共享标签数据 (生产者和消费者标签)

简介

Logix5000 控制器支持生产(广播)和消费(接收)系统共享标签的能力。

为了实现两个控制器共享生产者或消费者标签,必须将它们连接到相同的控制网络(例如, ControlNet 或 EtherNet/IP 网络)。不可以跨越两个网络连接生产者和消费者标签。

Logix5000控制器可以在如下网络生产和消费标签(网络长度与它们对这些网络通讯所支持的长度相同):

- ControlLogix 背板
- ControlNet 网络
- EtherNet/IP 网络

如果没有其它连接, 控制器支持标签个数。

身份	控制器支持
生产者	生产者标签个数 ≤ 127
消费者	消费者标签个数 ≤ 250(或控制器最大连接能力)

控制器支持的消费者和生产者标签的总数是:

$$\text{生产者标签个数} + \text{消费者标签个数} + \text{其它连接个数} \leq 250(\text{或控制器最大连接能力})$$

重要事项

实际上, 组态一个项目的生产者和消费者标签, 标签个数要与进行通讯的模块的连接数相匹配。

生产者和消费者标签的指南

指南	说明
不能跨越不同网络连接生产者和消费者标签。	为了实现两个控制器共享生产者或消费者标签，必须将它们连接到相同的控制网络。可以通过 ControlNet 或 EtherNet/IP 网络生产和消费标签。
在控制器作用域创建标签。	只能生产和消费(共享)控制器作用域的标签。
标签所占内存 ≤ 500 字节。	如果传送大于 500 字节的标签，应创建逻辑将数据打包传输。 如果通过 ControlNet 网络上消费标签，标签应不大于 480 字节，这是 ControlNet 网络的限制，而非控制器的限制。
将传送给同一个控制器的数据组合。	如果向同一个的控制器发送多个标签： • 创建一个或多个用户自定义结构体存放这些数据。这样比分别传送每个标签使用的连接数少。 • 根据相似的更新时间间隔组合数据。为了节省带宽，对较少的关键数据使用较大的 RPI。
使用如下数据类型之一： • DINT • REAL • DINT 或 REAL 数组 • 用户自定义结构体	要共享除 DINT 或 REAL 以外的数据类型，创建用户自定义结构体保存需要的数据。 生产者标签以及相关的消费者或其他标签使用相同的数据类型。
使用用户自定义结构体生产或消费 INT 或 SINT 数据。	创建带有 INT 或 SINT 成员的用户自定义结构体，从而生产或消费 INT 或 SINT 数据。成员可以是独立的 INT 或 SINT 数据，也可以是 INT 或 SINT 数组的成员。最终可以生产和消费用户自定义结构体。
生产者和消费者的数据类型必须匹配。	生产者和消费者中的生产者或消费者标签的数据类型必须相同。
向非 Logix 设备生产基于用户自定义数据结构体的标签。	控制器生产 32 位字的标签。以其它的存储位，如 16 位的通讯设备作目标设备，其获得的最终数可能不准确。为了避免差错，应在用户自定义结构体中构造生产者数据。
要是实现数据交换需要应用程序握手。	生产者数据以 RPI 为基准进行连续传送，所以很难知道新数据何时到达。可以设置一个位或在生产者标签中嵌入一个计数器，从而识别消费者接收到了新数据。也可以通过一个相反的生产者/消费者标签提供返回握手，使初始生产者得知消费者已经收到并处理了标签。
使用 CPS 指令缓冲生产者和消费者数据。	在生产者中，使用 CPS 指令将数据拷贝到输出标签。在消费者中，使用另一个 CPS 标签把数据拷贝到缓冲标签。 CPS 指令为数据结构大于 32 位的数据提供了数据完整性。 要点：当执行 CPS 指令时，控制器禁止所有中断。
使用单播 EtherNet/IP 通讯，从而减少广播网络流量。	为了减小带宽的使用和保护网络的完整性，一些设备块多点发送 Ethernet 信息包。16 版本的 RSLogix 5000 软件，具有组态生产者/消费者标签并使用多点发送或单播通讯的功能。单播通讯： • 减少网络带宽。 • 简化 Ethernet 交换机组态。

为生产者和消费者标签指定 RPI 速率的指南

在组态生产者和消费者标签时，可以设置请求信息包间隔(RPI)速率。RPI 值是控制器试图与模块通讯的速率。

指南	说明
确保 RPI 大于或等于 NUT。	使用 RSNetWorx 软件为软件选择网络更新时间(NUT)和规划网络连接时间。 如果网络上的模块和/或生产者/消费者标签的 RPI 比网络更新时间快，RSNetWorx 软件不能规划 ControlNet 网络的时间。
最小(最快)的消费者 RPI 决定了生产者标签的 RPI。	如果多个消费者需要相同的标签，最小(最快)的请求决定了指向所有消费者的生产者标签 RPI。

管理生产者和消费者标签连接的指南

指南	说明
使生产者和消费者标签数量最少。	使生产者和消费者标签数量最少，以便减少网络流量。同样，减少对高速、确定性数据，如联动装置的生产者和消费者标签的使用。
使用数组或用户自定义结构。	当向同一个控制器发送多个标签时，创建数组或用户自定义结构体来存放这些数据。每个生产者和消费者标签的字节限制为 500(≤ 500)。
正确组态消费者的数量。	确保组态消费者的数量与连接到每个生产者标签的消费者控制器数量一致。如果设置的数量大于实际控制器的数量，会浪费连接数。 缺省设置为每个生产者标签连接两个消费者。
连接多个生产者/消费者连接。	如果两个控制器之间有多个生产者和消费者连接，并且其中一个连接失败，所有的生产者和消费者连接都将失败。 建议将所有生产者和消费者数据组和到一个结构体或数组中，这样两个控制器之间只需要一个连接即可。

组态基于消费者标签的事件型任务

事件型任务根据预先组态的事件发生顺序自动地执行。消费者标签的到达也可以成为这样的事件。

- 只有一个消费者标签可以触发指定的事件型任务。
- 典型地，在生产者控制器中使用 IOT 指令下达生产新数据的指令。
- 当消费者标签触发事件型任务时，事件型任务在所有数据到达后才开始执行。

有关组态事件型任务的更多信息，请参阅第2章，将逻辑划分为任务、程序、例程和 Add-On 指令。

比较消息和生产者 / 消费者标签

方法	优点	要点
读 / 写消息	• 可编程启动 • 仅在需要时使用通讯和网络资源 • 支持大数据包的自动分段存储和重新组合，最多可包含 32767 个成员。 • 可以高速缓存一些连接，节省重新传输时间。 • 一般的 CIP 消息适用于第三方设备。	• 需要的资源不可用时，可能发生延迟。 • MSG 指令和处理过程影响控制器扫描(系统内务处理时间片)。 • 程序扫描时数据异步到达(使用可编程握手或使用 UID/UIE 指令对以便减少影响，不支持事件型任务)。
生产者 / 消费者标签	• 一次组态并根据请求数据包间隔(RPI)自动发送数据。 • 多个消费者可以同时受到同一个生产者标签的相同数据。 • 当消费者数据到达时可以触发事件型任务。 • 使用 ControlNet 时必须确定时间。 • 保留 ControlNet 资源。 • 不影响控制器的扫描。	• 仅被 Logix5000 和 PLC-5 控制器，以及 1784-KTCS I/O Linx 支持，并选择第三方设备。 • 背板通讯限于 500 字节，网络通讯限于 480 字节。 • 程序扫描时数据异步到达(使用可编程握手或 CPS 指令和事件型任务实现同步)。 • 必须分别获得连接状态。 • 在 EtherNet/IP 网络中，可以使用多播或单播连接组态生产者 / 消费者标签。

与 I/O 的通讯

简介

在 Logix5000 控制器中，I/O 值以 RPI(请求数据包间隔时间)周期进行刷新，这个周期是通过在工程的 I/O 组态文件夹中的 Module Property(模块属性)对话框中进行设置的。这个刷新值与逻辑的执行异步。

模块在指定的 RPI 内发送输入数据到控制器。因为这个传输是异步于逻辑执行的，控制器中的 I/O 值在扫描中可以改变。

缓存的 I/O 数据

如果多次引用一个 I/O 标签，则当程序扫描期间标签数值发生变化时，应用程序会受影响，用户必须在第一次引用该标签之前将这个 I/O 值复制缓冲的标签中。在用户程序代码中，引用缓存的标签而不引用 I/O 标签。

重要事项

对缓存的 I/O 数据使用同步复制指令(CPS)。当 CPS 指令复制数据的时候，没有 I/O 刷新或其他任务改变数据的值。对 CPS 指令有干扰作用的任务将被延时执行直到指令执行完成。过度使用 CPS 指令会抑制通讯任务的执行，影响控制器的性能。

缓存的 I/O 数据：

- 预防一个输入或输出值在程序执行期间改变(I/O 刷新异步于逻辑的执行)
- 复制一个输入输出标签到结构体的成员或数组的元素中
- 预防生产或消费数据的值在程序执行期间改变
- 确保所有的生产和消费数据以组的形式到达或发送(在多传送到时候不混合)
- 如果用户想要缓冲的 I/O 数据大于 32 位(或 4 个字节)只能使用 CPS 指令

过度使用 CPS 指令会严重影响 CPU 的效率。

如果使用用户自定义结构体并且成员代表 I/O 设备，那么用户必须逻辑程序将数据从相应的 I/O 标签复制到结构体的成员中。

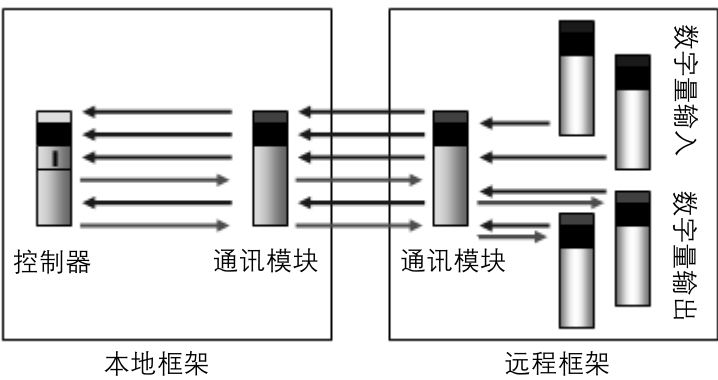
为 I/O 模块指定 RPI 速率指南 设置每一个模块的RPI速率(ControlLogix和SoftLogix)或每一个控制器的RPI速率(CompactLogix 和 FlexLogix)。RPI 的值为控制器与模块进行通讯的速率。

指南	说明
指定 RPI 为实际需要速率的 50%	设置比实际应用更快的 RPI 会浪费网络资源, 比如 ControlNet 规划带宽, 网络处理时间和 CPU 处理时间 例如, 如果用户每 80ms 需要一次信息, 设置 RPI 为 40ms。数据异步于控制器的扫描, 所以采样速率频率为所需的两倍这样可以保证更多的当前数据。
在同一个模块上具有相似性能需求的组设备	对于在同一个模块上具有相似性能需求的组设备, 用户合并数据传送到一个模块要好于传送到多个模块, 这样将节省网络带宽
设置 ControlNet 网络的刷新时间(NUT)等于小于 RPI 的最快值	当组态一个 ControlNet 网络的时候, 设置网络刷新时间(NUT)等于或小于系统中 I/O 模块和生产或者/消费者标签的最快 RPI 值。例如, 如果用户的 RPI 为 10ms, 设置 NUT 为 5ms 可以使规划网络更灵活。
在一个 ControlNet 系统中, RPI 应为 NUT 的偶数倍	设置 RPI 为 NUT 的二进制倍数。例如, 如果 NUT 为 10ms, 设置 RPI 可以为 10ms, 20ms, 40ms, 80ms, 160ms 等等。
在一个 ControlNet 系统中, 隔离 I/O 通讯	如果用户使用非规划 ControlNet 通讯或想要在运行时可以添加 ControlNet I/O(见 72 页), 将 ControlNet 网络专用于 I/O 通讯。在专用 I/O 网络中, 确保: • 没有 HMI 通讯 • 没有 MSG 通讯 • 没有编程工作站 • 没有多处理器系统结构的点对点互锁
在一个 EtherNet/IP 系统中, 基于状态改变的模块限于 RPI 的 1/4	如果用户通过 EtherNet/IP 网络在远程机架连接中组态模块为基于状态改变, 这个模块可以和其 RPI 同样的速率发送数据。起初, 模块立刻发送数据。然而, 当输入改变的时候, 模块数据在适配器中保持 1/4 RPI 时间以避免 therNet/IP 网络模块通讯的过载
基于控制器的数据传送	控制器类型决定数据传送速率 • ControlLogix 和 SoftLogix 控制器以模块组态的 RPI 传送数据 • CompactLogix 和 FlexLogix 控制器以 2ms 的幂指数倍传送数据(例如 2, 4, 8, 16, 64 或 128)。例如, 如果用于指定 RPI 为 100ms, 则数据实际传送为 64ms

I/O 模块的通讯格式

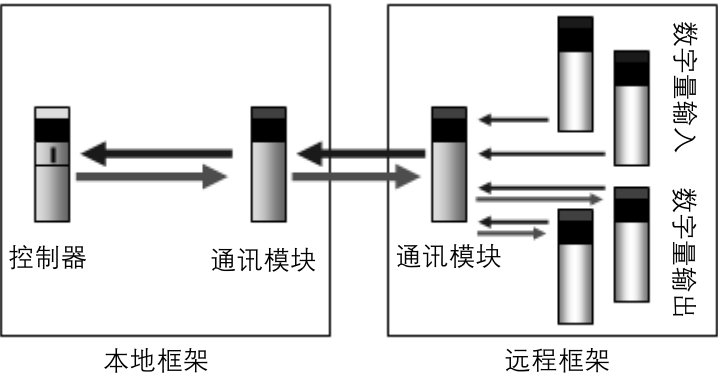
通讯格式决定了控制器是通过直接连接还是机架优化连接连接到 I/O 模块。通讯格式也决定了模块提供或使用的信息的类型和数量。

直接连接 每一个模块都单独的通过控制器接收或发送数据。通讯模块作为网络传送数据的桥梁。



好处	注意事项
• 每一个模块可以决定各自的速率(RPI) • 每一个模块可以发送更多的数据，例如诊断数据和模拟量数据 • 支持事件型任务通讯	• 需要额外的连接和网络资源 • 这是在本地机架中所支持的唯一方法 • I/O 数据将作为个体标签发送

机架优化连接 处于远程机架的通讯模块将多个模块的数据打成一个数据包然后将这个数据包作为一个连接发送到控制器。



好处	注意事项
• 一个连接可以服务于满机架的数字量模块 • 减少网络资源和负载	• 所有模块以相同的速率发送数据 • 不使用的槽仍然通讯 • 对于模拟量数据和诊断数据仍然需要直接连接 • 仅限远程机架 • 对于每个模块的 I/O 数据以别名标签的数组的形式呈现

机架优化格式限制一个机架中的每个模块的数据为一个32位的输入字。如果在机架中放置一个诊断模块，机架优化格式将消除诊断模块提供的数值。在这个例子中，最好使用直接连接，这样的话所有模块的诊断信息都可以传递到控制器中。

电子锁

当用户组态模块的时候,要进行电子锁配置来决定在同一个机架槽或位置中放置不同的模块。

选项	说明
精确匹配	任何以后要插入该机架槽的模块必须与之前插入的模块相同并且具有相同的代码，目录号，主版本号 and 次版本号
兼容匹配	任何以后要插入该机架槽的模块必须与之前插入的模块相同并且具有相同的代码，目录号，主版本号。另外，次版本号必须等于或大于之前的设置。 兼容匹配也允许模块替换其他相似的模块，例如，用户可以用1756-ENET(组态为兼容匹配)模块替换 1756-ENET， 1756-ENBT 或 1756-EN2T 模块
禁止电子锁	不需要设置电子锁。用户可以在同一个机架槽中放置不同的模块

警告



改变 RPI 和电子锁设置可能造成连接的终端并且数据丢失。

当禁止电子锁的时候要非常小心。如果不正确的使用，可能导致人身的伤害或死亡，财产的损害或经济的损失。

管理 I/O 连接指南

指南	说明
I/O 模块的类型决定了连接的类型	模拟量模块经常使用直接连接，除了使用连接信息的 1771 模拟量模块 数字量模块可以使用直接连接或机架优化连接。在标题中包含“optimization”的通讯格式属于机架优化连接；其他说有的连接属于直接连接
基于远程 I/O 模块来选择远程适配器的通讯格式	为远程适配器选择一个或一些格式
选择	如果
None	如果远程机架只包含模拟量模块，诊断数字量模块，带有保险丝的输出模块或通讯模块 在 ControlNet 网络中，在控制器运行时使用“None”在网络中添加新框架
Rack-Optimized	如果远程框架仅包含标准的数字量输入和输出模块(没有诊断模块或装有保险丝的输出模块) ControlNet 网络仅支持预定义通讯的机架优化。在控制器运行时，用户可以向一个存在的机架优化连接中添加新的数字量模块，但是新的机架优化连接只能在离线状态下添加。一个 EtherNet/IP 网络支持在在线和离线两种状态下添加新的机架优化连接。更多的信息，参见 72 页
Listen Only Rack-Optimized	如果用户想要从由另一个控制器控制的已进行机架优化的远程框架中接收 I/O 模块和框架槽的信息 只听机架优化连接的运行性能同机架优化连接一样

指南	说明
使用机架优化连接来节省连接的使用	如果用户想要限制控制器和网络连接的数目，可以使用机架优化连接
在一些情况中，直接连接的工作性能最好	对于远程适配器模块组态为机架优化连接，框架中的每一个槽都要发送数据，即使这个槽是空的或放置一个直接连接的模块。在控制器和远程适配器模块之间的机架优化传送 12 个字节的数据。另外，远程适配器模块每个槽发送 8 个字节的数据到控制器；控制器每个槽发送 4 个字节的数据到远程适配器 如果在一个多槽的框架中具有少数的数字量模块，最好使用直接连接因为传送满机架的信息需要更多的系统带宽
例子	说明
远程 17 槽框架 0 槽：1756-CNBR/D 1-15 槽：模拟量模块 16 槽：标准数字量模块	选项 1：选择远程适配器的通讯格式为机架优化。这个例子中使用 16 个控制器连接(15 个用于模拟量模块和 1 个机架优化连接)。这个例子中的传输为： • 机架优化 12 字节 • 数字量模块 12 字节 • 每一个模拟量模块 12 字节，一共 180 字节 选项 2：选择远程适配器的通讯格式为“None”。这个例子中使用 16 个控制器连接(每一个 I/O 模块一个直接连接)。没有机架优化数据的传送。 推荐：推荐选项 2，因为它避免了不必要的网络通讯，而且改善了网络性能
远程 17 槽框架 0 槽：1756-CNBR/D 1-8 槽：模拟量模块 9-16 槽：数字量模块	选项 1：选择远程适配器的通讯格式为机架优化。这个例子中使用 9 个控制器连接(8 个用于模拟量模块和 1 个机架优化连接)。这个例子中的传输为： • 机架优化 12 字节 • 每一个数字量模块 12 字节，一共 96 字节 • 每一个模拟量模块 12 字节，一共 96 字 选项 2：选择远程适配器的通讯格式为“None”。这个例子中使用 16 个控制器连接(每一个 I/O 模块一个直接连接)。没有机架优化数据的传送。 推荐：这个例子中最好的选项要取决于数字量 I/O 模块的类型和其他控制器的连接。如果系统中有很多模拟量模块，诊断模块，带保险丝的输出模块或生产者/消费者标签，选项 1 可以节省控制器的连接。如果有大量可用的控制器连接，选项 2 可以减少不必要的网络通讯

控制 1771 I/O 模块

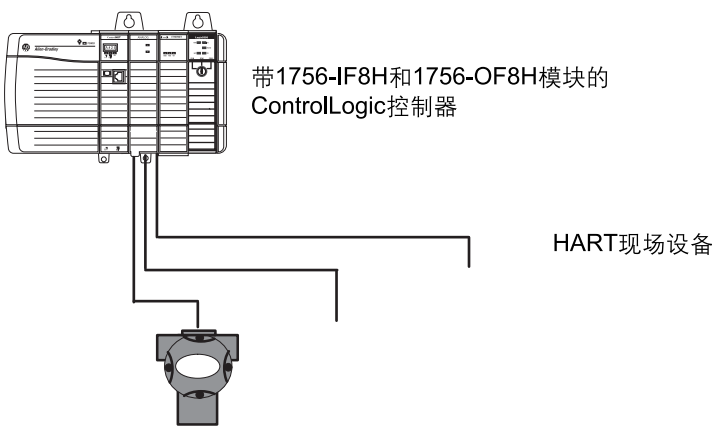
- Logix5000 控制器支持：
- 与 1771 数字量和模拟量模块的远程 I/O 通讯
 - 与 1771 数字量 I/O 模块的 ControlNet 通讯
 - 与 1771 模拟量和智能 I/O 模块通过 ControlNet 网络执行块传送信息指令

指南	说明
分布式 1771 模拟量 I/O 模块	1771 模拟量 I/O 模块分布多个框架可以减少单个 1771-ACN15、1771-ACNR15、或 1771-ASB 适配器信息块传送到数量 在不同的网络中隔离不同的 1771 框架可以使通讯多样化, 这样可以避免单一的网络通讯模块处理所有的块传送信息
ControlNet 的块传送增加 ControlNet 非规划带宽的数量	可用的非规划通讯时间的数量取决于预定义通讯的通讯量负载的大小 增加控制器系统的开销以分配更多的 CPU 时间来处理消息和块传送
块传送编程	每个 ControlNet NUT 的每个节点的非规划信息数据限制为 510 字节。1756-CNB 每个传送限制为 128 个字，如果需要，数据将以多个数据包发送。 数据传送与程序的扫描异步 有关块传送更多的信息，参见 90 页

与 HART 设备进行通讯

HART(Highway Addressable Remote Transmitter)是一个开放的协议，这个协议是为工业过程测量应用中连接模拟量设备所设计的。这个协议使用测量中广泛应用的 4-20mA 电流环。

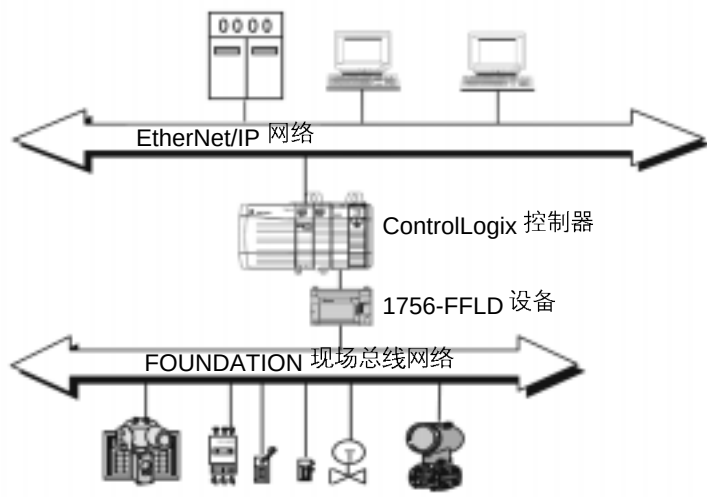
1756-IF8H 和 1756-OF8H 模块提供了在一个模块中模拟量和 HART 的连通性。用户可以将模块放置在本地的控制器或远程的ControlNet, EtherNet/IP 网络中。用户不需要额外的硬件来访问 HART 信号。



指南	说明
在需要的通道中使能 HART	所有的通道共享 HART 调制解调器，所以仅在需要的通道使能 HART 可以提高 HART 的响应时间
HART 部分标签的更新速率比模拟量标签慢	HART 网络的通讯量不同，更新速率也不同。如果将八个通道的 HART 都使能，更新速率在 10s 的范围内。确保在控制策略中考虑到了这个响应时间。另外，检查拥有 HART 的数据品质
设备变量状态(PVStatus, SVStatus, TVStatus, FVStatus)是 HART 系统中相关的新特性	如果用户的 HART 设备不支持设备变量状态，1756-IF8H 和 1756-OF8H 模块的状态值是基于 HART 设备的通讯状态的。 动态变量不像模拟量信号更新的那样快。实际的速率取决于组态 HART 通道的数量，通过指令的数量，手持通讯设备或其它从设备的存在和设备的响应时间
1756 HART 模块支持资产管理软件	HART 必须在任何资产管理访问可用前使能，包括多元扫描。RSLinx 专业版软件，RSLinx Gateway 软件和 RSLinx OEM 软件使得资产管理软件通过 RSLinx 网络和 1756 背板进行通讯 Endress+Hauser FieldCare 资产管理软件是 FDT 框架应用。这个框架应用运行 DTM 文件。DTM 文件为控制和设备卖主的可执行文件。有通讯 DTM 和设备 DTM。罗克韦尔自动化提供为 RSLinx 软件和 1756 背板提供了通讯 DTM 并且为 1756 HART 模块提供了两个其他的通讯 DTM。 Endress+Hauser 公司为他们仪表和阀提供了设备 DTM。设备 DTM 提供了需要组态，监视和维护设备的可见参数

与 FOUNDATION 现场总线设备进行通讯

1757-FFLD 连接设备将 EtherNet/IP 网络和 FOUNDATION 现场总线桥接在一起。



指南	说明
每一个 1757-FFLD 设备支持 16 个逻辑交换功能块	每一个逻辑交换功能块支持八个模拟量的输入和八个模拟量的输出。每一个功能块使用一个 CIP 连接。 一个 1756-ENBT 通讯模块支持 125 个 CIP 连接，支持 7 个 1757-FFLD 设备，每一个设备支持 16 个功能块。添加额外的 756-ENBT 模块支持额外的 1757-FFLD 设备。
设备的类型影响每个 H1 网段上的 FOUNDATION 现场总线设备的最大数量	每一个 1757-FFLD 设备支持 2 个或 4 个 H1 网段。 • 平均每个 H1 网段 8-10(最大 16 个)个仪表 • 对于阀，每个网段 4-6 个，因为阀具有更多的数据
每个 H1 网段不要超过实际通讯关联 (VCRs) 的最大数量	一个 VCR 是一个提供在 FOUNDATION 现场总线设备间传送数据的通道。需要发送和接收数据的 VCR 的数量取决于设备和数据类型。 1757-FFLD 对每个 H1 网段最大支持 16 个发布方(publisher)和 16 个接收方(subscriber)VCR。每一个通过 Logix5000 控制器的参数都使用一个 VCR。一些设备(例如：阀)使用更多的 VCR。
确保每一个 FOUNDATION 现场总线设备都有正确的设备描述(DD)	DD 就好象 DeviceNet 设备中的 EDS 文件。用户可以在厂商 / 组织机构的网站或随设备附带的媒体上找到 DD。运行 DD 服务的主机可以对现场设备的 DD 中的所有参数进行定义。
使用正确的导线和连接产品	常用的是树型或改进的树型拓扑结构。不使用菊花链设备。 噪声是最常见的问题，应归于： • 错误的接线 • 不正确的接地 • 劣质的连接器
为了达到最佳的执行效果，需要了解 FOUNDATION 现场总线系统的详细信息。	参见如下参考信息： • http://www.fieldbus.org 上的 FOUNDATION 现场总线技术概览 • http://relcominc.com 上的 RelcomWiring 指南

为 I/O 数据创建标签

当通过编程软件组态模块的时候每一个I/O的标签都是自动的创建的。每一个标签的命名遵循下面的格式：

位置:槽号:类型.成员名称.子成员名称.位

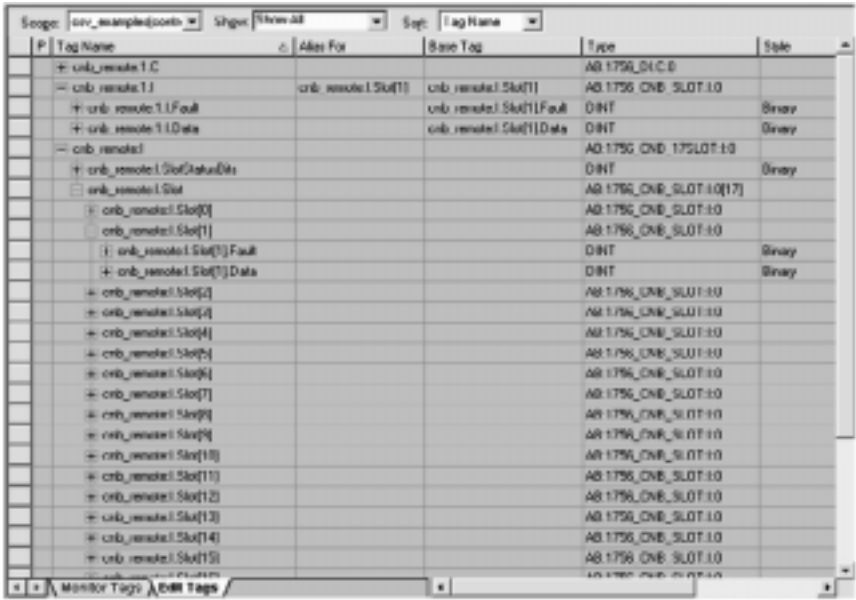
地址变量	IS
位置	鉴别所处网络位置 LOCAL= 本地框架或 DIN 导轨 ADAPTER_NAME= 远程适配器或网桥
槽号	I/O 模块所处框架的槽号
类型	数据类型: I= 输入, C= 组态, O= 输出, S= 状态
成员名称	来自于 I/O 模块的数据, 例如数据和错误, 取决于模块
子成员名称	与成员名称相关的数据
位(可选)	I/O 模块的点, 取决于 I/O 模块的尺寸(对于 32 点模块为 0-31)

如果用户组态了一个机架优化连接, 软件创建一个远程通讯模块的机架对象标签。用户可以引用单独的机架机架优化的I/O模块, 或者机架对象标签中的元素。

例如, 一个远程 ControlNet 通讯模块 (remote_cnb)在 1 槽中有一个 I/O 模块。

这是为远程框架中1槽的I/O模块所创建的单独标签。

这是远程通讯模块的机架对象标签的入口, 标识了I/O模块处于远程 1 槽中。



控制器所属关系 当选择了通讯格式，用户还要选择是否建立模块的所有者或只听关系。

模式	说明
所有者	所有者控制器可以写模块的组态字并创建模块连接
只听	使用只听连接的控制器只能监视模块。不能对模块的组态字进行写操作并且当所有者控制器控制 I/O 模块的时候只能保持对 I/O 模块的一个连接

输入模块的所有权相对于输出模块有一个明显的差别。

控制	所有权	说明
输入模块	所有者	一个输入模块由控制器组态，这个控制器作为所有者建立了一个连接，是建立所有者连接的第一个控制器。 当输入模块被组态后(并为一个控制器所有)，其他的控制器仍然可以建立这个模块的所有者连接。这允许如果在最初的所有者控制器断掉连接的时候其他所有者控制器也可以接收组播的数据。所有其他的所有者必须同原始的所有者控制器具有相同的组态数据和通讯格式，否者连接请求将被拒绝。
	只听	当输入模块被组态后(并为一个控制器所有)，其他控制器可以建立这个模块的只听连接。这些控制器可以接收组播的数据。如果所有的所有者控制器都断开了与输入模块的连接，那么只听连接的控制器将不再接收组播的数据。
输出模块	所有者	一个输出模块由控制器组态，这个控制器作为所有者建立了一个连接。输出模块仅允许有一个所有者连接。如果其他的控制器试图建立所有者连接，连接请求将被拒绝。
	只听	当一个输出模块被组态的时候(并为一个控制器所有)，其他的控制器可以建立这个模块的只听连接。这些控制器可以接收组播数据。如果所有者控制器断开与输出模块的连接，那么只听连接的控制器将不再接收组播的数据。

运行时 / 在线添加 I/O 模块

使用 15 版本的 RSLogix5000 软件，用户可以在控制器运行的时候添加 1756 I/O 模块。

- 在运行的时候用户只能添加 1756 I/O 模块
- 用户可以在本地框架中，通过 ControlNet 网络非规划部分远程和通过 EtherNet/IP 网络添加 1756 I/O 模块

网络

注意事项

ControlNet

用户可以在运行的时候向机架优化连接或直接连接中添加 ControlNet I/O 模块(当运行时添加 ControlNet I/O 模块时不可以创建机架优化连接)

用户可以使用：

- 1756-CN2, 1756-CN2R 任何系列的模块
- 1756-CNB, 1756-CNBR D 系列或更新的通讯模块

禁止数字量输入模块的基于状态改变(COS)特性因为它可以造成输入数据传送比 RPI 更快

仅将一个 ControlNet 网络用于 I/O 通讯，对于专用的 I/O 网络，确保：

- 没有 HMI 通讯
- 没有 MSG 通讯
- 没有编程工作站

非规划模块的 RPI 快于 25ms 将使 1756-CNB, 1756-CNBR 通讯模块过载

- 使用 10ms 或更多的 NUT
- 尽可能的保持 SMAX 和 UMAX 的值最小

用户可以添加 I/O 模块直到到达：

- 1756-CNB, 1756-CNBR 通讯模块的 75% 利用率。取决于 RPI，每增加一个 I/O 模块利用率增加 1-4%
- 1756-CNB, 1756-CNBR 通讯模块达到 48 个连接
- ControlNet 网络保留的非规划带宽<350000 字节

EtherNet/IP

在线添加 EtherNet/IP 的 I/O 和离线时添加 EtherNet/IP 的 I/O 的步骤相同。EtherNet/IP I/O 即时通讯取决于模块的 RPI。不需要规划，所以 EtherNet/IP 通讯可以看作是非预规划的

EtherNet/IP I/O 模块在运行时可以添加到已存在的机架优化连接，新的机架连接或直接连接中(用户可以运行添加 EtherNet/IP I/O 的时候创建新的机架优化连接)

用户可以添加 I/O 模块直到达到通讯模块所限制的数量

1756-EN2T 模块	1756-ENBT 模块	1756-ENBT/B 模块
10000pps	5000pps	900pps
128 TCP 连接	64 TCP 连接	64 TCP 连接
256 CIP 连接消息	128 CIP 连接消息	160 CIP 连接消息

运行 / 在线添加 I/O 模块

1756 模块类型和 连接方法	位于本地框架中		位于经过 ControlNet 网络的远程框架中				位于经过 EtherNet/IP 网络的远程框架中	
	离线	运行 ⁽¹⁾	离线 规划	非规划	运行 ⁽¹⁾ 规划	非规划	离线	运行 ⁽¹⁾
运动 - 直接	Yes	No	N/A	N/A	N/A	N/A	N/A	N/A
数字 - 直接	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
数字 - 机架优化	N/A	N/A	Yes	No	Yes	No	Yes	Yes
模拟量 - 直接	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
第三方 - 直接	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
1756-DNB	Yes	No	Yes	No	No	No	Yes	Yes
1756-DHRIO	Yes	No	Yes	No	No	No	Yes	Yes
1756-CNx- 无连接	Yes	Yes	Yes	Yes	No	Yes	N/A	N/A
1756-CNx- 机架优化	N/A	N/A	Yes	N/A	N/A	N/A	N/A	N/A
ControlNet 第三方 - 直接	N/A	N/A	Yes	Yes	No	Yes	N/A	N/A
1756-EN2T, 1756-ENBT, 1756-ENET- 无连接	Yes	Yes	N/A	N/A	N/A	N/A	Yes	Yes
1756-EN2T, 1756-ENBT, 1756-ENET- 机架优化	N/A	N/A	N/A	N/A	N/A	N/A	Yes	Yes
EtherNet/IP 第三方 - 直接	N/A	N/A	N/A	N/A	N/A	N/A	Yes	Yes

(1) 支持使用 15 版本的 RSLogix5000 编程软件添加

运行 / 在线添加 I/O 设计注意事项

当用户设计网络的时候,如果需要在运行时添加I/O模块,则参考这些注意事项。

设计要点	说明
I/O 模块	通常，只能在运行时添加 1756 I/O 模块 在本地框架，ControlNet 上的远程框架或 EtherNet/IP 上的远程框架上为想要添加的 I/O 模块保留空间
输入传输速率	确保 RPIs 对想要发送和接收的数据起作用 确保所添加的 I/O 依赖于基于状态改变的数据
网络拓扑	在一个 ControlNet 网络中，安装备用的分接器以便在运行时添加 1756 I/O 模块的时候可以不中断网络。每一个分接器必须进行终结，以至于系统不暴露于地。检查 ControlNet 系统的需求来决定用户的网络可以支持多少个备用的分接器。 • 在具有冗余电缆的 ControlNet 网络中，用户可以中断干线添加一个新的分接器，但是在添加模块的过程中冗余电缆将丢失 • 在 ControlNet 环中，添加一个新的分支环或添加一个新的同轴电缆的节点并且中断网络的相应部分 • 用户可以移除已经存在的单节点并添加一个中继器，然后从新添加这个节点并且可以在新的网段中添加新的节点 在 EtherNet/IP 中，保留交换机的一些连接点以便在将来可以添加新的节点或交换机
网络组态	在 ControlNet 网路中，设计哪些通讯需要规划而哪些不需要规划 在 EtherNet/IP 网络中，所有的通讯都是即时的并且基于模块的 RPI(也可以看作是非规划的) 如果用户知道以后需要添加一个装有数字量模块的新框架,组态网络并且将它添加到I/O组态树中作为机架优化，然后禁止通讯适配器知道需要添加新框架的时候
网络性能	用户可以在运行时添加 I/O 模块知道影响到了通讯模块的通讯能力。 确保有足够的通讯模块用于计划添加的连接

对于更多的信息，参见《在 ControlNet 和 EtherNet/IP 上运行 / 在线添加 ControlLogix(1756)I/O 白皮书》，出版物 LOGIX-WP006。

确定合适的网络

简介

NetLinx 开放网络架构是罗克韦尔自动化使用开放网络技术的策略，这个技术对管理层和车间层的网络进行综合无缝连接。NetLinx 体系架构的网络包括 DeviceNet、ControlNet 和 EtherNet/IP，共享一个通用的通讯服务集。这些网络是 Logix 控制系统推荐使用的网络。

选择一种网路

比较	EtherNet/IP 网络	ControlNet 网络	DeviceNet 网络
控制 I/O	较好	最好	低密度
组态设备	最好	最好	最好
采集数据	最好	较好	好
对等互锁	较好	最好	好
设备	较好	较好	最好
拓扑	星型， 需要交换机	干线 / 分支， 中继器起始	干线 / 分支
节点数量	许多节点	99	63
性能	最好	最好	好
运行 / 在线添加 I/O 模块	支持 ⁽¹⁾	支持 ⁽¹⁾	即将支持 ⁽²⁾

⁽¹⁾ 更多的信息，参见 72 页中的“运行 / 在线添加 I/O 模块”。

⁽²⁾ 1756-DNB 7 或更新的固件版本和相应的 RSNetWorx for DeviceNet 软件，用户可以在运行的网络上添加节点。在固件版本不支持的情况下，用户只能在扫描器处于空闲状态的时候向 DeviceNet 网络上添加节点。

设计网络的时候遵循这些指南

设计要点	说明
网络拓扑	为将来的连接做好计划。 计划好将来添加 I/O 模块所需要额外的控制器和 / 或通讯模块。
网络组态	在 ControlNet 网络中，计划那些通讯需要规划哪些则不需要规划。 在 EtherNet/IP 网络中，所有的 I/O 通讯都是基于模块的 RPI(可以看作是非规划)。 如果用户知道以后需要添加一个装有数字量模块的新框架，组态网络并且将它添加到 I/O 组态树中作为机架优化，然后禁止通讯适配器知道需要添加新框架的时候。
网络性能	确保有足够的通讯模块用于计划添加的连接。
框架	巩固多模块连接到单一网络节点的通讯连接。一组数字量 I/O 模块作为一个机架优化连接可以减少通讯量和网络带宽。
输入传输速率	确保 RPIs 对想要发送和接收的数据起作用。 确保所添加的 I/O 依赖于基于状态改变的数据。

关于设计运行 / 在线添加 I/O 模块更多的信息，请参见 72 页。

EtherNet/IP 网络拓扑结构

EtherNet/IP 网络	拓扑
• 一个 EtherNet/IP 网络支持消息，生产者 / 消费者标签和分布式 I/O • 一个 EtherNet/IP 网络支持全 / 半双工 10Mbps 或 100Mbps 通讯 • 一个 EtherNet/IP 网络不需要网络规划和路由表 • 配置 EtherNet/IP 网络设备的参数有一些方法可用，但不是任何时候所有的方法都有效。这些方法由设备和组态决定： – DHCP – 罗克韦尔自动化 BOOTP/DHCP 工具 – RSLinx 软件 – RSLogix 5000 软件 – RSNetWorx for EtherNet/IP 软件 应用思想 • 连接多台计算机 • 业务系统的默认网关 • 星型拓扑结构最适于少量的节点和较短的距离	例子 1 <pre>graph TD; Device1[Device] --- Switch[Switch]; Switch --- Device2[Device]; Switch --- Device3[Device];</pre> 例子 2 <pre>graph TD; Router[Router] --- Switch1[Switch]; Router --- Switch2[Switch]; Switch1 --- Device1[Device]; Switch1 --- Device2[Device]; Switch2 --- Device3[Device]; Switch2 --- Device4[Device];</pre>

EtherNet/IP 网络的指南

指南	说明								
参见这些出版物	当安装 EtherNet/IP 网络的时候参见这些出版物： • 《Logix5000 控制系统的 EtherNet/IP 模块用户手册》，出版物 ENET-UM001 • 《EtherNet/IP Web 服务器模块用户手册》，出版物 ENET-UM527 • 《EtherNet/IP 介质设计和安装指南》，ENET-IN001								
确保交换机具有要求的特性	对于 EtherNet/IP 控制，使用工业等级的交换机。 <table> <tr> <th>必须或推荐</th><th>交换机特性</th></tr> <tr> <td>必须</td><td>所有端口都具有全双工能力</td></tr> <tr> <td>推荐</td><td> • VLAN • 自协商和手动设置速度 / 双工 • 线速交换结构 • SNMP </td></tr> <tr> <td>在要求管理交换机的系统中推荐 (参见 78 页中的“EtherNet/IP 系统交换机指南”)。</td><td> • IGMP snooping – 限制了与指定 IP 组播组相关端口的组播流量 – 多数的交换机需要 IGMP snooping 的路由器 – 在一个独立的网络中，如果没有路由器，确保交换机支持 IGMP snooping • 端口诊断 • 端口镜像(在排查故障时需要) </td></tr> </table>	必须或推荐	交换机特性	必须	所有端口都具有全双工能力	推荐	• VLAN • 自协商和手动设置速度 / 双工 • 线速交换结构 • SNMP	在要求管理交换机的系统中推荐 (参见 78 页中的“EtherNet/IP 系统交换机指南”)。	• IGMP snooping – 限制了与指定 IP 组播组相关端口的组播流量 – 多数的交换机需要 IGMP snooping 的路由器 – 在一个独立的网络中，如果没有路由器，确保交换机支持 IGMP snooping • 端口诊断 • 端口镜像(在排查故障时需要)
必须或推荐	交换机特性								
必须	所有端口都具有全双工能力								
推荐	• VLAN • 自协商和手动设置速度 / 双工 • 线速交换结构 • SNMP								
在要求管理交换机的系统中推荐 (参见 78 页中的“EtherNet/IP 系统交换机指南”)。	• IGMP snooping – 限制了与指定 IP 组播组相关端口的组播流量 – 多数的交换机需要 IGMP snooping 的路由器 – 在一个独立的网络中，如果没有路由器，确保交换机支持 IGMP snooping • 端口诊断 • 端口镜像(在排查故障时需要)								
考虑使用紧密合作伙伴的交换机。 数据传输依赖于控制器。	这些紧密合作伙伴具有可以满足需要的特性的交换机：Cisco、Hirschmann 和 N-Tron。 Logix5000 控制器的类型决定了数据传输的速率。 • ControlLogix 和 SoftLogix 控制器以用户为模块组态的 RPI 传输数据。 • CompactLogix、FlexLogix 和 DeviceLogix 控制器以 2ms 的幂数次倍传输数据(例如 2、4、8、16、64 或 128)。例如，如果用户指定 RPI 为 100ms，实际的数据传输速率为 64ms。								
用户可以在运行时添加 I/O 模块。	使用 15 版本固件的控制器，用户可以在控制器运行时通过 EtherNet/IP 网络在远程机架连接添加 1756 I/O 模块。可以组态为直接连接或机架优化连接。更多的信息，请参见 72 页。								
数据传输速率依赖于 RPI。	一个 EtherNet/IP 网络以设置的 RPI 向控制器广播 I/O 信息。基于状态改变(COS)设置使能并且： • 没有数据改变，EtherNet/IP 模块在每个 RPI 都生产数据。 • 数据改变，EtherNet/IP 模块以 RPI/4 的最大速率生产数据。								
使用单播 EtherNet/IP 通讯减少网络广播流量。	为了减少带宽使用和维护网络的完整性，一些措施阻止组播以太网数据包。使用 16 版本固件的 RSLogix5000 软件，用户可以组态一个生产者 / 消费者标签来使用组播或单播连接。单播连接： • 减少网络带宽。 • 简化以太网交换机设置。								

EtherNet/IP 系统中交换机的指南

使用管理交换机	使用无管理交换机
• EtherNet/IP 控制系统是通过交换机或路由器直接连接到业务系统的。将控制系统和业务系统进行适当的隔离通常是一个好的设计习惯。 • 系统采用的是非罗克韦尔自动化的 EtherNet/IP 设备连接到网络(个人计算机除外)。这些设备可能不会适当的处理 I/O 设备产生的组播流量。 • 如果系统需要执行排查故障。对于排查故障，用户需要使用仅管理交换机才支持的端口镜像。 • 通过每个设备的流量(数据包 / 秒)要小于每个设备的容量。	在 I/O 系统中： • 在一个隔离的 EtherNet/IP 体系中，控制系统不是直接连接到业务系统上的。或者控制系统是通过一个 ControlLogix 网关连接到业务系统(例如，一个 ControlLogix 框架包含两个 1756-ENBT 模块，一个连接到控制系统另一个连接到业务系统)。 • EtherNet/IP 控制系统仅包含罗克韦尔自动化设备(个人计算机除外)。 在非 I/O 系统中： • 网络中 EtherNet/IP 的流量仅包含消息(MSG 指令，HMI，程序上载 / 下载)。这这种情况下，不论网络的结构如何，使用无管理交换机是合适的选择。

如果使用无管理交换机，用户将不能使用系列功能：

- 交换机端口诊断
- 端口镜像
- 强制双工速度
- SNMP
- IGMP snooping
- 用于组态和诊断的网页浏览

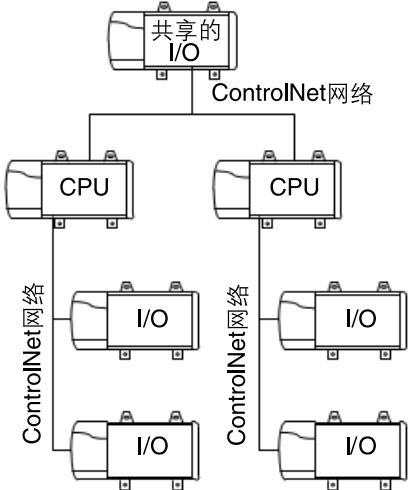
确定用户的系统是否操作正确

罗克韦尔自动化 EtherNet/IP 设备(例如 156-ENBT 扫描器)内嵌有诊断网页。

在这个网页上	期望
以太网统计页 • 所有介质计数器 • In Error 和 Out Error 计数器 • 拒绝数据包计数器	这些数值应该为 0 并且不增加。
诊断概览页	• 没有连接超时。 • 数据包 / 秒的数量应在每个设备容量的范围内。 • I/O 数据包计数器统计表中的 MISSED 计数器应为 0。

如果连接经常断开或 HMIs 的显示更新很慢，应减小流量负荷。如果情况和组播有关，最好还是使用带 IGMP snooping 的管理交换机。

ControlNet 网络拓扑结构

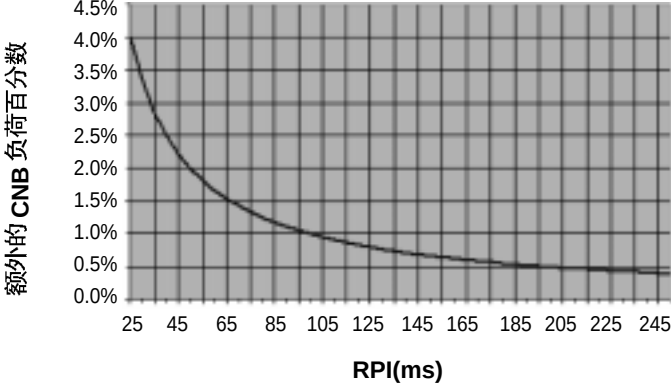
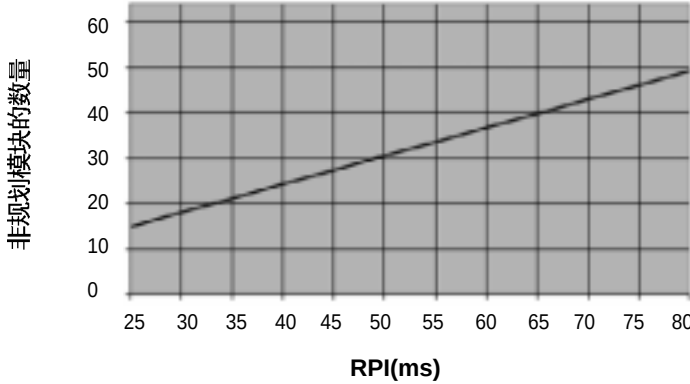
ControlNet 网络	拓扑
- ControlNet 网络允许在同一根导线上传递 I/O 数据和消息数据。 - 多个控制器和他们各自的 I/O 数据也可以放在同一条 ControlNet 导线上。 - 当添加了新的 I/O 或一个已存在的 I/O 模块的通讯结构发生变化时，用户必须使用 RSNetWorx for ControlNet 软件来重新规划网络。 - 如果网络时间改变，网络上的每一个规划的设备的通讯将受到影响。 - 为了减小改变带来的影响，将每一个 CPU 和各自的 I/O 置于隔离的 ControlNet 网络上。 - 将共享 I/O 和生产者 / 消费者标签置于公用的网络上，每一个 CPU 都可以使用这些信息。 应用思想 - 默认 Logix 网络。 - 最好替换为通用远程 I/O。 - 多重分布式 DeviceNet 网络高速链路。 - 对等互锁网络。 - 公用设备包括：Logix5000 控制器、PanelView 终端、I/O 模块和变频器。	 The diagram illustrates a ControlNet network topology. At the top, a '共享的 I/O' (Shared I/O) module is connected to a 'ControlNet 网络' (ControlNet network). This network branches into two separate 'ControlNet 网络' segments. Each segment contains a 'CPU' module and two 'I/O' modules, all connected to the same network line. The segments are labeled 'ControlNet 网络' on the left and right sides.

ControlNet 网络的指南

指南	说明
参阅这些出版物。	当安装 ControlNet 网络的时候参阅这些出版物： 《ControlNet 同轴电缆介质设计和安装指南》，出版物 CNET-IN002 《ControlNet 光纤电缆介质设计和安装指南》，出版物 CNET-IN001 《Logix5000 控制系统 ControlNet 模块用户手册》，出版物 CNET-UM001
每个 ControlNet 网络节点的数量限制为 40。	ControlNet 的每个网络节点数限制为 99 个，但是节点数量会降低网络的性能。每个网络最大 40 个节点具有最佳的性能并且可以为其他通讯保留带宽。
调整 RSNetWorx for ControlNet 默认设置。	在 RSNetWorx for ControlNet 软件中改变这些设置。 - UMAX(一个网络中最大的非规划节点号) - 默认为 99 - 网络需要花相应的时间来处理在此设置的所有数量的节点，即使实际的网络上没有这么多的设备。 - 改变网络上活动设备和可能连接的额外设备的响应级别。 - SMAX(一个网络中最大的规划节点号) - 默认为 1。 - 这个数对于所有系统都需要改变。 - 设置 SMAX<UMAX。

指南	说明
设计时至少要留有 400KB 可用的, 非规划的网络带宽。	为非规划的网络带宽保留少量的内存将导致较差的消息吞吐量和较慢的工作站响应。 ControlNet 上非规划数据传输异步于程序的扫描并且每个 ControlNet NUT 最大支持 510 字节 / 节点。1756-CNB 和 1756-CNBR 通讯模块每个数据传送最大支持 128 个字。如果需要, 非规划数据还可以多个数据包的形式发送。
将设备网(1756-DNB)和串口(1756-MVI) 通讯模块放置在本地框架中。	DeviceNet(1756-DNB)和串口(1756-MVI)通讯模块具有500字节倍数的数据包将影响预定义带宽。将这些模块放置在各控制器同一个框架中可以避免这些数据在 ControlNet 网络中被规划。 如果用户必须将这些通讯设备放置在远程机架中, 设置输入和输出的大小要与 RSNetWorx for DeviceNet 软件中的设置相一致, 这将减少需要传输的数据量。
限制 1756-CNB, 1756-CNBR 的连接。	为了得到最佳的性能, 将 1756-CNB, 1756-CNBR 的连接限制为 40-48。如果用户需要更多的连接则需要 在 同一个框架中添加更多的模块。添加更多的模块和凌乱的线路会影响系统的性能。 如果包含 CNB 模块的框架也包含多和数字量 I/O 模块, 设置 CNB 模块的通讯格式为机架优化, 否则, 选 None。 1756-CNB, 1756-CNBR 模块要使用在仅含有 I/O 模块的框架中, 1756-CN2, 1756-CN2R 通讯模块要与控制器在同一个框架中。
对于额外的连接考虑 1756-CN2, 1756-CN2R 模块。	每一个 1756-CN2, 1756-CN2R 通讯模块支持 100 个连接。 除 100 个连接之外, 模块还具有更高的性能。 1756-CN2, 1756-CN2R 通讯模块要与控制器在同一个框架中。 1756-CNB, 1756-CNBR 模块要使用在仅含有 I/O 模块的框架中。
如果用户改变网络设置, 重新保存 每一个控制器的工程。	任何时候用户使用 RSNetWorx for ControlNet 软件并且保存或覆盖编辑, 关系到系统中的每一个控制器各自的 RSLogix5000 工程文件和任务的保存。这将 ControlNet 的设置拷贝到离线数据库文件中并且确保将来下载到控制器并上线的时候不需要运行 RSNetWorx for ControlNet 软件。
用户可以在运行时添加 I/O 模块。	使用带有 15 版本固件的控制器, 用户可以在控制器运行时通过 EtherNet/IP 在远程机架连接中添加 1756 I/O9 模块。可以组态为直接连接或机架优化连接
数据传输依赖于控制器。	Logix5000 控制器的类型决定了数据传输的速率 • ControlLogix 和 SoftLogix 控制器以用户为模块组态的 RPI 传输数据。 • CompactLogix, FlexLogix 和 DeviceLogix 控制器以 2ms 的幂数次倍传输数据(例如 2, 4, 8, 16, 64 或 128)。例如, 如果用户指定 RPI 为 100ms, 实际的数据传输速率为 64ms。

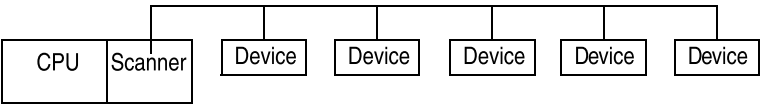
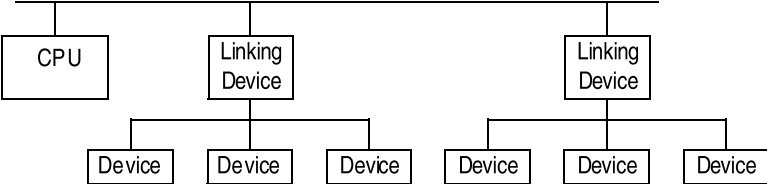
非规划 ControlNet 网络的指南

指南	说明																										
用户可以将整个 ControlNet 网络以非规划运行。	一个非规划网络： • 提供更简单的网络组态 • 对于 I/O 更新比较慢的应用是很有用的 用户仍然需要运行RSNetWorx for ControlNet软件来组态NUT、SMAX、UMAX和介质组态设置。																										
如果用户将 I/O 放置在了非规划的 ControlNet 网络中要进行适当的设计。	对于非规划 ControlNet 网络的 I/O 参考下列建议： • 用户必须使用 1756-CNB、1756-CNBR D 系列或更新的版本。 • 禁止数字量输入模块的基于状态改变(COS)特性因为这将造成输入数据的发送比 RPI 快。 • 将 ControlNet 网络专用于 I/O。 • 不要超过 1756-CNB、1756-CNBR 通讯模块利用率的 75%。 • 1756-CNB、1756-CNBR 通讯模块不要超过哦 48 个连接。 • 使用 10ms 或更长的 NUT。 • 保持 SMAX 和 UMAX 尽可能的小。																										
设置 RPI 为 25ms 或更慢。	对于非规划的模块使用 25ms 或更慢的 RPI 可以避免 1756-CNB、1756-CNBR 通讯模块过载。 每增加一个 I/O 模块通讯模块的负载将增加 1-4%，依赖于 RPI 额外的 1756-CNB、1756-CNBR 负荷  <table border="1"><caption>额外的 1756-CNB、1756-CNBR 负荷数据表</caption><thead><tr><th>RPI (ms)</th><th>额外的 CNB 负荷百分数</th></tr></thead><tbody><tr><td>25</td><td>4.0%</td></tr><tr><td>45</td><td>2.5%</td></tr><tr><td>65</td><td>1.8%</td></tr><tr><td>85</td><td>1.4%</td></tr><tr><td>105</td><td>1.1%</td></tr><tr><td>125</td><td>0.9%</td></tr><tr><td>145</td><td>0.8%</td></tr><tr><td>165</td><td>0.7%</td></tr><tr><td>185</td><td>0.6%</td></tr><tr><td>205</td><td>0.5%</td></tr><tr><td>225</td><td>0.4%</td></tr><tr><td>245</td><td>0.3%</td></tr></tbody></table>	RPI (ms)	额外的 CNB 负荷百分数	25	4.0%	45	2.5%	65	1.8%	85	1.4%	105	1.1%	125	0.9%	145	0.8%	165	0.7%	185	0.6%	205	0.5%	225	0.4%	245	0.3%
RPI (ms)	额外的 CNB 负荷百分数																										
25	4.0%																										
45	2.5%																										
65	1.8%																										
85	1.4%																										
105	1.1%																										
125	0.9%																										
145	0.8%																										
165	0.7%																										
185	0.6%																										
205	0.5%																										
225	0.4%																										
245	0.3%																										
RPI 将影响可以放置 I/O 模块的数量。	这个表格表示了 RPI 和模块数量的关系，不要超过 1756-CNB、1756-CNBR 通讯模块利用率的 75%。 在非规划网络中 I/O 模块的最大数量  <table border="1"><caption>在非规划网络中 I/O 模块的最大数量数据表</caption><thead><tr><th>RPI (ms)</th><th>非规划模块的数量</th></tr></thead><tbody><tr><td>25</td><td>15</td></tr><tr><td>30</td><td>18</td></tr><tr><td>35</td><td>21</td></tr><tr><td>40</td><td>24</td></tr><tr><td>45</td><td>27</td></tr><tr><td>50</td><td>30</td></tr><tr><td>55</td><td>33</td></tr><tr><td>60</td><td>36</td></tr><tr><td>65</td><td>39</td></tr><tr><td>70</td><td>42</td></tr><tr><td>75</td><td>45</td></tr><tr><td>80</td><td>48</td></tr></tbody></table>	RPI (ms)	非规划模块的数量	25	15	30	18	35	21	40	24	45	27	50	30	55	33	60	36	65	39	70	42	75	45	80	48
RPI (ms)	非规划模块的数量																										
25	15																										
30	18																										
35	21																										
40	24																										
45	27																										
50	30																										
55	33																										
60	36																										
65	39																										
70	42																										
75	45																										
80	48																										

比较规划和非规划 ControlNet 通讯

规划 ControlNet 通讯	非规划 ControlNet 通讯
确定性的	比规划通讯具有较小的确定性 当不需要规划网络的时候允许更简单的控制网安装
在 ControlNet 上添加规划 I/O 用户必须： - 在离线的 RSLogix5000 工程中添加 I/O - 下载 RSLogix5000 工程到控制器 - 运行 RSNetWorx 进行网络规划(规划网络时必须将网络停止并且将控制器置于编程模式) - 保存 RSLogix5000 工程	在不影响规划的情况下进行在线更改 通过非规划带宽新模块可能影响其他模块的通讯
RPI 和 NUT 决定模块的通讯速率	RPI 决定模块的通讯速率
MSG 和 HMI 可以在同一个网络中发生因为他们在非规划通讯中是隔离的 MSG 和 HMI 流量可能影响 I/O 通讯	ControlNet 网络仅用于 I/O 通讯 MSG 和 HMI 流量可能影响 I/O 通讯
直接和机架优化连接到 I/O	仅直接连接到 I/O(导致使用更少的 I/O 模块的数量因为连接数限制控制器和通讯模块)
支持任何固件版本的 ControlNet 通讯模块	用户可以使用任何版本的 1756-CN2, 1756-CN2R 通讯模块, 如果使用 1756-CNB, 1756-CNBR 通讯模块的话, 必须为 D 系列或更新的版本
支持可以通过 ControlNet 网络通讯的任何 I/O 平台	使用 15 版本的 RSLogix 软件, 支持 1756 I/O 模块

DeviceNet 网络拓扑结构

DeviceNet 网络	拓扑
- 用户需要 DeviceNet 扫描器来连接控制器和 DeviceNet 设备。 - 用户必须使用 RSNetWorx for DeviceNet 软件来组态设备并创建扫描器列表。 - 用户可组态的网络比特率为 125Kbps (缺省设置并且具有较好的起始点)、250 Kbps 或 500 Kbps。 - 如果网络上的每一个设备(扫描器除外)发送的输入数据 ≤ 4 字节并且接收的输出数据 ≤ 4 字节, 用户可以使用扫描器的自动扫描特性来组态网络。 应用思想 - 分布式设备 - 变频器网络 - 诊断信息	单一网络  多个小分布式网络(子网络) 

DeviceNet 网络的指南

指南	说明
参阅这些出版物。	当安装 DeviceNet 网络的时候参阅这些出版物： • 《DeviceNet 电缆系统手册》，出版物 DN-UM072 • 《Logix5000 控制系统 DeviceNet 模块用户手册》，出版物 DNET-UM001
使用 DeviceNet 标签生成工具。	16 版本的 RSLogix5000 软件包含 DeviceNet 标签生成器工具，这个工具可以创建基于 RSNetWorx for DeviceNet 软件设备特性结构的标签和逻辑。逻辑在 DNB 的数组标签和设备的标签之间拷贝数据，所以数据的更新与程序的扫描同步。
将设备网扫描器模块(DNB)置于本地框架。	将 DNB 模块置于本地框架中可以获得最佳性能，尤其在 ControlLogix 系统中。按照比实际设备多出 20% 来设置 DNB 模块的输入和输出映像区，用来为以后的时候保留。如果用户将 DNB 模块置于远程机架中，输入输出映像区的设置都最佳性能的影响至关重要。
检查网络数据总量没有超过 DNB 数据表的	一个 DNB 支持： • 124, 32 位输入字 • 123, 32 位输出字 • 32, 32 位状态字 用户可以使用 RSNetWorx for DeviceNet 软件离线评估网络数据，如果网络上的数据超过一块 DNB 所能支持的容量，可以使用第二块 DNB。
先设置从设备。	在将设备添加到组态列表之前先组态其参数。许多设备在添加到列表后将不能改变组态参数。如果用户先组态扫描器，有可能出现扫描器的组态与当前设备的组态不一致的情况。如果组态不一致，在扫描网络的时候将找不到设备。
63 号节点不使用，以用于添加新节点。	设备在出厂时缺省的节点号为 63。将 63 号节点保留这样可以在网络上添加新设备，然后再改变新设备的节点地址。
62 号节点不使用，以用于连接计算机。	通常 62 作为连接到网络上的计算机的节点地址，用于排查故障和组态。
不要忘记将扫描器的运行位置位。	要将扫描器置于运行模式，控制器必须处于运行模式并且控制器中的逻辑必须将扫描器的运行位置位。
	
用户确保有当前设备的 EDS 文件。	RSNetWorx for DeviceNet 软件使用 EDS 文件来识别设备。如果软件不能正确的识别设备，用户可能是安装了错误的 EDS 文件，对于一些设备，用户可以通过上载设备的信息来创建 EDS 文件。或从网站 http://www.ab.com/networks/eds 来获取 EDS 文件。

注释:

与其他设备的通讯

简介

MSG 指令与其他设备数据块的读写时异步的。

如果目标设备为	选择这些消息类型中的一个
Logix5000 控制器	CIP 数据表读
	CIP 数据表写
需要使用 RSLogix5000 软件组态的 I/O 模块	模块重新设置
	CIP generic
SERCOS 驱动器	SERCOS IDN 读
	SERCOS IDN 写
PLC-5 控制器	PLC5 类型读
	PLC5 类型写
	PLC5 字范围读
	PLC5 字范围写
SLC 控制器	SLC 类型读
MicroLogix 控制器	SLC 类型写
块传送模块	块传送读
	块传送写
PLC-3 控制器	PLC3 类型读
	PLC3 类型写
	PLC3 字范围读
	PLC3 字范围写
PLC-2 控制器	PLC2 无保护读
	PLC2 无保护写

缓冲的消息

一些类型的消息使用连接接收和发送数据。当消息传送完毕的时候有些会提示用户选择打开连接或关闭连接。下面的表格表示了哪一个消息使用连接并且用户是否可以缓冲连接。

消息类型	使用通讯方式	使用连接	哪个可以缓冲
CIP 数据表读或写	CIP	×	×
PLC2, PLC3, PLC5 或 SLC(全系列)	CIP		
	带 ID 源的 CIP		
	DH+	×	×
CIP generic	N/A	用户选择 ⁽¹⁾	用户选择 ⁽¹⁾
块传送读或写	N/A	×	×

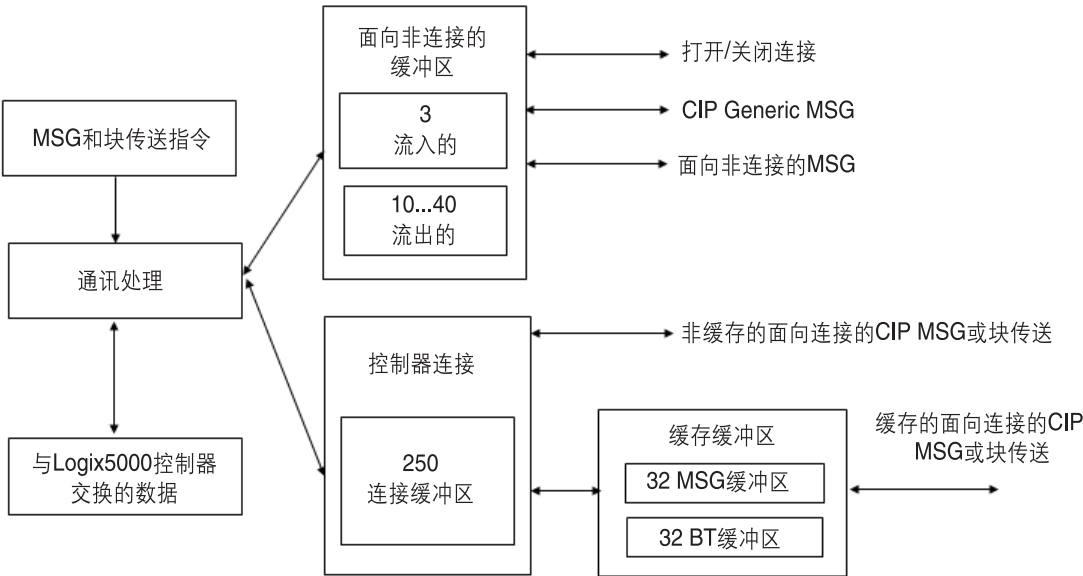
(1) 用户可以连接 CIP generic 消息，但是对于大多数应用我们建议用户不连接 CIP generic 消息。

一个缓冲的连接保持打开直到下列条件发生：

- 控制器处于编程模式。
- 用户将消息置于非缓冲的。
- 初始化另一个消息并且需要一个缓冲区。
- 在连接中一个中间节点失效。

消息缓存

一个Logix5000控制器具有面向非连接信息和缓冲信息的缓冲区。缓冲区存储输入的和输出的数据信息直到控制器可以处理数据。



早于12版本的控制器固件仅有16个缓存缓冲区，用于MSG和块传送。

缓冲区	说明
10 个流出的面向非连接的缓冲区 用户可以使用 CIP generic 消息指令将这个数字增加到 40。每一个添加的缓冲区大约使用 1.2KB 的 I/O 内存。参见《Logix5000 控制器通用指令参考手册》的 MSG 部分，出版物 1756-RM003	流出的面向非连接缓冲区是用于： • 建立 ControlNet、EtherNet/IP 和远程 I/O 网络上的本地 I/O 模块和远程设备的 I/O 连接 • 在 Ethernet 或 ControlNet(CIP 和带源 ID 的 CIP)网络上执行面向非连接的 PLC2、PLC3、PLC5 或 SLC(全系列)消息。 • 在 DH+ 网络上初始化消息(使用 2 个缓冲区，一个用于打开连接，一个用于传送数据)。 • 初始化非缓冲块传送。 • 初始化非缓冲 CIP 读 / 写消息指令。 • 初始化缓冲块传送。 • 初始化缓冲 CIP 读 / 写消息指令。 • CIP generic 消息指令。
3 个流入的面向非连接的缓冲区	流入的面向非连接的缓冲区用于： • 初始化接收缓冲的 CIP 消息指令。 • 接收非缓冲 CIP 消息指令。 • 接收 DH+ 网络的消息。 • 接收 CIP generic 消息指令。 • 接收来自 ControlNet PanelView 终端的读或写请求(面向非连接消息)。 • 最初接收来自 EtherNet/IP PanelView 终端的读请求(面向连接消息)。 • 接收来自 EtherNet/IP PanelView 终端的写请求(面向非连接消息)。 • 接收来自 RSLogix5000 软件上线初始化请求。 • 初始化 RSLinx 连接的接收。

缓冲区	说明
缓存的缓冲区	缓存的缓冲区是消息和块传送的流出的缓冲区。一个缓存的连接可以提高消息性能，因为连接保持打开并且在下一个时间执行的时候不需要重新建立。缓存的连接对控制器总连接数的限制进行计数。缓存的连接以 RPI 的时间进行更新。当控制器置于编程模式的时候所有存储入口将关闭。
12 或更新的版本	使用 12 或更新的固件版本，用户可以缓存 32 个消息和块传送(任何组合)。更早的固件版本的控制器允许用户缓存 16 个消息和 16 个控制块。
32 个缓存的缓冲区，用于消息和块传送的任意组合	缓存消息第一次执行时，将使用 10 个流出的面向非连接的缓冲区中的一个。当连接建立起来后数据将被移到缓存的缓冲区中。
11 或更早的版本	为了获得最佳的性能，缓存的消息或块传送的数量不要超过缓存的缓冲区的数量。如果用户存储了比可用缓存的缓冲区数目更多的消息或块传送，控制器将会寻找长时间处于不活动状态的连接，关闭那个连接，然后允许一个新的连接来替换它。控制器将关闭缓存的消息或块传送，取决于哪一个在长时间内处于不活动状态。如果所有 32 个缓存的连接都使用，消息将使用流出的面向非连接缓冲区中的一个。如果所有的面向非连接缓冲区都处于使用状态，消息指令将产生错误代码 301(没有缓冲区内存)或 302(无可用带宽)。
16 个缓存的缓冲区仅用于消息， 16 个缓存的缓冲区仅用于块传送	用户可以复用缓存的连接，如果一个连接处于不活动状态并且消息指令执行与其具有相同的目标和路径，指令将使用这个连接。例如，如果用户对同一个模块进行块传送读或写，读和写将互锁确保在一段时间内只有一个处于活动状态。然后当进行缓存时，它们将使用同一个缓存的连接。

流出的面向非连接缓冲区

缓冲区	说明
1...10	前 10 个缓冲区(缺省)被面向非连接消息、初始化面向连接消息、建立 I/O 连接和建立生产者/消费者连接所共享。
11	第 11 个缓冲区用于建立 I/O 和生产者/消费者连接
12...40	第 12 到第 40 个缓冲区仅被用于初始化面向连接消息和执行面向非连接消息。为了将流出的缓冲区数值增加到 11 以上，执行 CIP generic 消息进行设置，以便于当从编程模式切换到运行模式时改变生效。

消息的指南

指南	说明
消息标签必须作为基本标签存在于控制器域中。	操作系统访问消息标签中的信息异步于程序扫描。除消息标签的可见区域外，其隐藏的部分仅由操作系统后台所访问。
用户可以在一个程序中使用多于 32 个消息。	控制器支持 32 个活动的缓存的消息。如果用户决定使用消息的数目多于 32 个，将不能存储所有的消息。用户需要额外的程序来确保同时不超过 32 个消息处于活动状态。12 版本之前的固件，控制器一次只支持 16 个活动的缓存的消息。
用户可以使用一个消息用来发送大量的数据。	即使有网络数据包的限制(例如 ControlNet 为 500 字节 DH+ 为 244 字节)，控制器仍可以通过 MSG 指令发送大量的数据。当组态消息的时候，选择一个源/目的地标签的数组并且选择想要发送元素的数量(32767 个元素)。控制器自动将数组拆分成小的片段然后将所有的片段发送到目的地。在接收端，数据出现在片段中，所以需要一些应用代码来识别最后一个片段的到达。

管理消息连接的指南

指南	说明
创建用户自定义结构体或数组。	用户自定义结构体可以使用户组织起与机器或过程相匹配的数据。 • 一个标签包含与用户系统具体内容相关的所有数据。这可以使所有相关的数据集合到一起并容易的进行定位，无论数据的类型如何。 • 每一个单独的数据片(成员)有一个描述性的名称。这将为用户的逻辑创建一个初始水平的文件。 • 用户可以使用结构体来创建相同数据区的复用标签。 • 除了独立标签，RSLinx 还可以优化用户自定义结构体。
在适当的时候缓存连接。	如果消息重复的执行，则缓存连接。这将保持连接打开并且优化执行时间。每一次都打开一个连接将使消息的执行时间增加。 如果一个消息执行的不频繁，则不缓存连接。这将在消息执行完毕后关闭连接，这可以释放这个连接以用于其他的消息。

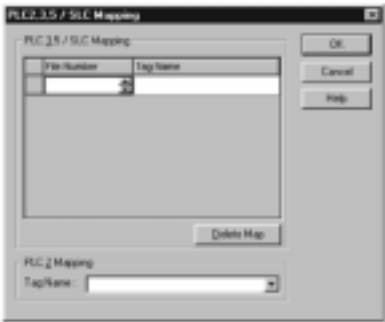
用户为系统设置的系统开销时间片百分率决定了控制器用于通讯和处理后台功能的时间比(包括周期型和事件型任务)。包括发送和接收消息。

关于系统开销时间片百分比的更多信息，参见 27 页中的“选择系统开销时间片百分比”。

块传送消息的指南

指南	说明
分布于多个框架的 1771 模拟量模块。	分布于多个框架的 1771 模拟量模块减少了单独的 1771-ACN 或 1771-ASB 模块需要管理的块传送数量。
将不同的 1771 框架隔离在不同的网络。	将不同的1771框架隔离在不同的网络将使通讯多样化,这样可以不用单一的网络或通讯模块来处理所有的通讯。
增加 ControlNet 非规划带宽	如果在 ControlNet 网络上进行通讯,增加 ControlNet 非规划带宽的数量来允许网络上额外的时间用于数据交换。 关于 ControlNet 网络的非规划带宽的更多信息, 请参见 82 页。
增加系统开销时间片百分率。	增加 Logix5000 控制器的系统开销时间片来分配 CPU 更多的时间来处理连续型任务的通讯。
同一模块的块传送读或写通讯互锁。	在程序中实现块传送对同一模块读和写消息的互锁确保在同一时间不会出现两个操作同时活动的情况。
对带有大量的块传送模块的系统使用 1757-ABRIO 模块。	1757-ABRIO 模块提供了 ControlLogix 框架到 1771 I/O 和其他模块通过远程 I/O 的连通性。1757-ABRIO 模块减轻了控制器块传送的负担并且增加了可执行块传送的数量。

映射标签



Logix5000控制器将标签的名称存储在控制器中,所以其他的设备读写数据的时候不需要知道其物理地址。但是许多设备仅识别 PLC/SLC 的地址表, 所以 Logix5000控制器提供了一个 PLC/SLC 映射功能, 这个功能可以使用户将 Logix 标签名称和物理地址相映射。

- 用户仅需要映射在信息中使用的文件号, 其他文件号不需要映射。
- 映射表会加载到控制器中并且当逻辑地址访问数据的时候会用到。
- 用户仅能访问控制器域标签(全局数据)。

按照下面的步骤进行标签映射。

- 不要使用 0, 1 和 2 号文件。在 PLC5 控制器中这些文件是为输出, 输入和状态文件保留的
- 使用 PLC-5 仅能为数据类型为 INT, DINT 或 REAL 型的标签进行映射。试图映射系统结构体的元素可能会产生不良的影响
- 使用这些文件类型和标识符

Logix5000 的数组类型	PLC 文件标识符
INT 数组	N 或 B
DINT 数组	L
REAL 数组	F

FactoryTalk 报警和事件系统

简介

FactoryTalk 报警和事件系统通过在 Logix5000 控制器中嵌入报警引擎，集成了 FactoryTalk View SE 应用软件和 Logix5000 控制器的报警。需要：

- 16 或更新版本的 RSLogix 5000 软件。
- 5.0 或更新版本的 FactoryTalk View SE。
- Logix5000 控制器。

固件	版本
ControlLogix 非冗余的控制器	16.20 或更新
ControlLogix 冗余的控制器	16.60 或更新
ControlLogix 控制器	16.20 或更新
DriveLogix 系统	16.20 或更新
SoftLogix 控制器	16.40 或更新

梯形图、结构化文本和功能块可用两种基于 Logix 报警指令。

- 数字量报警(ALMD)指令根据布尔(真 / 假)条件检测报警。
- 模拟量报警(ALMA)指令根据模拟量数值变化的等级或速率检测报警。

基于 Logix 的报警指令的指南

指南	说明														
估算每个报警增加的控制器内存。	报警指令使用新的报警数据类型，包含每个报警的状态信息和时间戳。估算这些报警在控制器中的内存使用： - 接收控制器报警的每个FactoryTalk报警和事件订阅者(subscriber)占用100KB内存。 每个控制器最多支持16个订阅者。多数应用软件只要求一个订阅者对应一个控制器，向多个 FactoryTalk View SE 客户提供数据。 - 每个报警 1KB。														
报警指令增加总的控制器扫描时间。	如下执行时间显示了 ALMD 指令和 ALMA 指令如何影响总扫描时间。 <table><tr><th rowspan="2">梯级状态</th><th colspan="2">执行时间</th></tr><tr><th>ALMD 指令</th><th>ALMA 指令</th></tr><tr><td>假</td><td>N/A(ALMD 将梯级状态作为输入状态)</td><td>17μs</td></tr><tr><td>真，并且未扫描到报警状态变化</td><td>7μs</td><td>65μs</td></tr><tr><td>真，扫描到报警状态变化</td><td>25 118μs</td><td></td></tr></table> 报警状态的改变指报警状态改变的任何事件，例如报警确认或抑制报警。通过创建报警相关性，可以减少大量报警状态同时改变(报警爆炸)的可能。大规模报警爆炸可能对应用程序代码的扫描时间产生很大影响。	梯级状态	执行时间		ALMD 指令	ALMA 指令	假	N/A(ALMD 将梯级状态作为输入状态)	17μs	真，并且未扫描到报警状态变化	7μs	65μs	真，扫描到报警状态变化	25 118μs	
梯级状态	执行时间														
	ALMD 指令	ALMA 指令													
假	N/A(ALMD 将梯级状态作为输入状态)	17μs													
真，并且未扫描到报警状态变化	7μs	65μs													
真，扫描到报警状态变化	25 118μs														
可以在线编辑或添加报警指令。	在线编辑新建的和现有的报警，并自动将它们发送给用户。无需重新预定，即可收到更新信息。在线所做的更改自动的从控制器报警结构体传播到架构的剩余部分。														
在梯形图上，对指令面板 faceplate 上报警值的定义，决定了是否可以通过报警结构体编程存取那些值。	当创建报警指令时，也可以为那个报警创建一个报警数据类型。例如，报警 MyDigitalAlarm 的数据类型是 DigitalAlarm。在梯形图上，指令面板也可以使用如下值： - ProgAck - ProgReset - ProgDisable - ProgEnable 在梯形图上，如果为面板参数输入数值或分配标签(例如 AckSection 1All)，每次扫描指令时，将数值或标签值自动地写入报警结构体。 在梯形图上，如果想要编程存取报警结构体，必须向面板分配结构体标签。例如，为了在逻辑中使用 MyAnalogAlarm.ProgAck，在面板上将标签 MyAnalogAlarm.ProgAck 分配给 ProgAck 参数。														
RSLogix 5000 软件的测试报警行为。	在报警对话框Status列表中监视报警条件、确认报警、禁用报警、抑制报警或复位报警。使用对话框选项即可查看报警怎样动作，而无需可操作的 HMI。														

组态基于 Logix 的报警指令

选项	说明														
消息字符串	消息字符串(最多包含255个字符, 包括内部文本)包含向报警操作员显示的报警信息。除输入文本外, 还可以插入变量信息。在报警消息编辑器里, 选择需要的变量信息, 并将其添加到消息字符串的任何位置。 <table> <tr> <th>变量</th><th>说明</th></tr> <tr> <td>报警名称</td><td>报警的标签名称。 /*S:0 % 报警名称 */</td></tr> <tr> <td>状态名</td><td>报警的状态(例如, 真、假、高高或低)。 /*S:0 % 状态名 */</td></tr> <tr> <td>输入值</td><td>真、假或模拟量输入的当前值。 /*N:5 % 输入值 NOFILL DP:0*/</td></tr> <tr> <td>极限值</td><td>引起报警的极限或条件。 /*N:5 % 极限值 NOFILL DP:0*/</td></tr> <tr> <td>严重等级</td><td>分配给报警的重要性。 /*N:5 % 重要性 NOFILL DP:0*/</td></tr> <tr> <td>相关联标签的数值</td><td>并与报警一起发送的被选标签数值。 /*N:5 % 标签 1 NOFILL DP:0*/</td></tr> </table> 这个信息总是和报警一起发送, 可以被操作员看到, 并且无论是否将其插入到消息字符串中, 该信息都将存入历史记录。 用户不能通过编程从报警标签访问报警消息字符串。为了改变基于具体事件的报警消息, 组态一个相关联的标签作为字符串数据类型, 并在消息中嵌入这个相关联的标签。 支持多语言版本的消息。通过导入/导出功能输入不同语言。有关更多信息, 请参阅第 94 页。	变量	说明	报警名称	报警的标签名称。 /*S:0 % 报警名称 */	状态名	报警的状态(例如, 真、假、高高或低)。 /*S:0 % 状态名 */	输入值	真、假或模拟量输入的当前值。 /*N:5 % 输入值 NOFILL DP:0*/	极限值	引起报警的极限或条件。 /*N:5 % 极限值 NOFILL DP:0*/	严重等级	分配给报警的重要性。 /*N:5 % 重要性 NOFILL DP:0*/	相关联标签的数值	并与报警一起发送的被选标签数值。 /*N:5 % 标签 1 NOFILL DP:0*/
变量	说明														
报警名称	报警的标签名称。 /*S:0 % 报警名称 */														
状态名	报警的状态(例如, 真、假、高高或低)。 /*S:0 % 状态名 */														
输入值	真、假或模拟量输入的当前值。 /*N:5 % 输入值 NOFILL DP:0*/														
极限值	引起报警的极限或条件。 /*N:5 % 极限值 NOFILL DP:0*/														
严重等级	分配给报警的重要性。 /*N:5 % 重要性 NOFILL DP:0*/														
相关联标签的数值	并与报警一起发送的被选标签数值。 /*N:5 % 标签 1 NOFILL DP:0*/														
相关联的标签	最多可以从控制器项目中选择 4 个附加标签与报警相连。这些标签与报警消息一起传送给报警服务器。连接的标签可以是如下数据类型: BOOL、INT、SINT、DINT、REAL 或字符串。例如, 安全阀的数字量报警也应包括泵速度和系统压力, 以及容器温度这些信息。 可以将相关联的标签随意嵌入消息文本字符串。														
严重等级	可组态的严重等级范围从 1 到 1000, 可以通过它们设置报警的重要等级。严重等级 1 适用于优先级低的报警; 严重等级 1000 适用于紧急状况。在 FactoryTalk 报警服务器中, 严重等级的缺省设置为: • 1 到 250 是低级报警。 • 251 到 500 是中级报警。 • 501 到 750 是高级报警。 • 751 到 1000 是紧急报警。 可以组态 FactoryTalk 范围向操作员显示的方式。处理器同样可以对报警等级进行筛选。例如, 维护工程师可以通过筛选设置仅查看严重等级为 128 的报警。														

选项	说明
报警分类	通过报警分类将相关报警分组。正确规定报警分类，以及相同类别中使用的每个报警。报警分类是面向事件的。 例如，指定 Tank Farm A 分类为某特定领域全部容器报警的组合。或者指定控制回路分类为全部 PID 回路报警的组合。 可以在分类基础上通过 HMI 显示和过滤报警。例如，操作员可以显示所有容器报警或 PID 回路报警。 报警分类不替换具体报警的订阅。FactoryTalk View SE 报警对象图形上有组态选项，该选项可以决定操作员可查看哪个控制器的报警。
查看命令	当操作员请求某个报警时，执行操作员工作站的命令。这样使操作员可以执行任何标准 FactoryTalk View 的命令，例如请求具体面板和显示、执行宏指令、访问帮助文件和运行外部应用项目。当报警状况发生并向操作员显示该状况时，操作员可以看到报警条，并通过报警汇总上的按钮运行相关查看命令。
缺省设置	在报警指令属性的参数表中可以定义指令参数。可以将参数返回给 FactoryTalk 缺省设置，并定义用户指令缺省设置。用户指定的指令缺省设置仅对指令的这个实例适用。

报警消息的多种语言文本

可以以多种语言保持报警消息。在相关语言文本的 RSLogix 5000 编程软件中或导入 / 导出(.CSV 或.TXT)文件中均可输入不同语言。

可以从导入 / 导出(.CSV 或.TXT)文件获得报警消息，并可以在初始消息字符串中添加额外的翻译文本行。不同语言的消息在 TYPE 栏中使用 ISO 语言代码。操作员文本存放在 DESCRIPTION 列。SPECIFIER 对报警类型进行识别。

TYPE	NAME	DESCRIPTION	SPECIFIER
ALMSG:en-us	ALMA Logix Tag Name	HH alarm text for operator in English	HH
ALMSG:en-us	ALMA Logix Tag Name	H alarm text for operator in English	H
ALMSG:en-us	ALMA Logix Tag Name	L alarm text for operator in English	L
ALMSG:en-us	ALMA Logix Tag Name	LL alarm text for operator in English	L L
ALMSG:en-us	ALMA Logix Tag Name	ROC positive alarm text for operator	POS
ALMSG:en-us	ALMA Logix Tag Name	ROC negative alarm text for operator	NEG
ALMSG:en-us	ALMA Logix Tag Name	HH Mitteilung für den Operator auf Deutsch	HH
ALMSG:en-us	ALMA Logix Tag Name	H Mitteilung für den Operator auf Deutsch	H

使用导入/导出功能创建和翻译多种语言的消息字符串。.TXT 导入/导出格式支持所有语言，包括汉语、日语和韩语在内。.CSV 导入/导出格式不支持汉语、日语和韩语。

导入和导出消息总是执行合并操作。在.CSV文件中删除的消息并不能相应地在.ACD文件中删除。要想删除消息，需要导入.CSV文件，该文件包括类型、名称和指定的区域文件，但是将注释栏置空。

在 HMI 中查看报警消息时：

- 这里没有缺省的语言字符串。如果没有指定消息文本的语言，FactoryTalk View 软件将其找到的第一条消息字符串的语言文本显示出来。
- 日期和时间格式并不随语言的变化而改变。它们与操作系统格式一致。
- 非组态对话框，如带注释对话框的 ACK，不可以转换语言。它们与操作系统使用相同的语言。

报警过程

当报警系统上电时，报警按照如下方式建立与控制器的初始连接：

1. RSLinx Enterprise 服务器启动对报警的订阅。
2. 控制器服务器为预订保留 100 KB 的缓冲区。
3. 控制器将其全部报警信息发送给订阅者：
 - 路径和标签信息
 - 报警组态
 - 所有组态语言中的消息字符串
4. 一旦用户收到发现信息，它将向报警请求订阅。

在这个阶段每个报警通常向用户传送 500 字节的数据。一般而言，对于具有 1000 个报警(500 个模拟量报警和 500 个数字量报警)的系统，其发现阶段将持续 35 秒左右的时间。这个时长与控制器加载、网络加载以及消息字符串大小和语言有关。

该阶段报警系统正常运行时，控制器按照如下方式向用户发送报警数据：

1. 报警事件发生时，控制器为报警信息附上时间戳并将其发送给用户。

通常，一个模拟量报警的状态信息包是 500 字节，一个数字量报警的状态信息包是 250 字节。

2. 用户向适当的客户应用程序和历史数据库发送报警数据。
3. 操作员确认报警信息，需要将操作员已确认报警的这个信息记录到历史数据库。(这个时间戳来自于操作员工作站。)
4. RSLinx Enterprise 服务器将确认请求发送给控制器。
5. 控制器收到确认请求，将报警作为确认标识，并将完成的动作时间戳返回给用户。
6. 控制器将带有时间戳的确认证实返回给用户。(这个时间戳来源于控制器。)
7. 用户将确认发送给适当的客户和历史数据库。

因为在正常报警运行期间，时间戳由多个位置产生，它对协调系统工作站和控制器的时钟具有非常重要的作用。有关更多信息，请参阅带 ControlLogix 控制器的时钟同步的不同方式，Knowledgebase 文档 40467。

缓冲的报警

控制器为每个用户保留了 100 KB 的空间用以缓存报警数据，防止用户与控制器的连接断开。通常，这个缓冲区可以保持 1000 个事件。

指南	说明
一旦用户与控制器的连接断开，应尽快重新建立连接。	控制器的报警缓冲区继续缓冲新的报警，直到所有缓冲区都排满(100KB)或缓冲时间结束。 在 RSLinx Enterprise 软件中组态报警服务器时，可以组态缓冲区超时，从 1 分钟到 2 小时(缺省设置为 20 分钟)。如果在缓冲时间内用户与控制器重新连接失败，控制器将清除缓冲，并收回为控制器正常运行保留的 100 KB 缓冲区。
通过查看 RSLogix 5000 软件的指令面板，可以检查用户与控制器的连接状态。	RSLogix 5000 编程软件中报警属性的状态表标识了控制器是否正在缓冲报警数据。这个值将在下一事件发生时更新。

可编程访问的报警信息

每个报警指令都包含存放报警组态和执行信息的报警结构体。报警结构体包含控制编程元素和操作员元素。报警指令并不通过模式设置来决定激活程序访问还是激活操作员访问，因此这些因素一直是激活的。

如下是在报警指令中执行动作的三种方式。

访问	报警结构体元素	要点
控制程序	- ProgAck - ProgReset - ProgSuppress - ProgDisable - ProgEnable	使用控制器逻辑可编程访问报警系统的元素。例如，控制程序可以决定禁止一些列于一个单独的根目标相连的报警。控制程序可以访问标签 MyDigitalAlarm.ProgDisable，禁用报警指令 MyDigitalAlarm，其数据类型为 DigitalAlarm。
用户 HMI	- OperAck - OperReset - OperSuppress - OperDisable - OperEnable	通过用户面板访问报警系统标签。例如，如果操作员需要移动一个工具，而不是通过报警屏幕分别手操禁止或抑制报警，操作员可以按下禁用键，以便访问标签 MyDigitalAlarm.OperDisable 标签。
标准 HMI 对象	不可用	通常，操作员交换作用发生于 FactoryTalk View 应用程序的报警汇总和报警条对象。这个交换作用与上述的用户 HMI 选项相似，但是这里不能编程可见或作用。

经由梯形图指令访问报警元素，执行全局报警操作。例如，将 **BOOL** 标签分配给所有 **ProgDisable** 区域，该区域位于 **ToolA** 中与梯形图面板相连的报警里。然后用 **DisableToolA** 禁用所有使用该标签的报警的操作。

重要事项

如果为报警面板上的 **ProgAck**, **ProgReset**, **ProgDisable**, or **ProgEnable** 功能指定了标签，则在报警数据类型中无需使用报警结构体元素完成相同的功能。例如，如果指定用 **DisableToolA** 禁用报警 **MyDigitalAlarm**，就不用编程访问 **MyDigitalAlarm.ProgDisable** 标签了。这样做可能会引起这样的状况，即面板要求一种操作，而报警标签又要求进行另一种操作。

如果想要使用报警结构体元素可编程的改变报警系统，为这些元素分配一个面板。例如，在 **ProgDisable** 的报警面板上，从报警结构体标签 **MyAlarmTag.ProgDisable** 中分配标签。这样就可以在其他代码单元编程访问 **MyAlarmsTag.ProgDisable**，而不存在冲突。

在 **HMI** 和事件日志中，任何控制器驱动的事件，无论来自报警结构体还是报警对话框，都作为离散事件(**Discrete Events**)记录在历史数据库中。这也包括任何经操作员元素(**OPERxxx**)访问相同信息的 **HMI** 界面。所以当时间戳和时间在日志中记录后，日志并不包含什么引发事件的标记。例如，在这种情况下，因为操作员和 workstation 不通过 **Factory Talk** 报警图形对象发生动作，因此不对它们进行追踪。

抑制或禁用报警

抑制报警指移除 **HMI** 中已经存在，并被用户得知的报警，但是仍然保持报警的有效性。这样就可以在解决已知报警时抑制报警汇总，而无需继续查看报警信息。抑制报警并不出现在操作员报警汇总或报警条上，但是抑制报警仍然会被发送给用户、记录在历史数据库中，同样会传输报警状态、时间戳，以及对其他可编程的或操作员的作用作出回应。

禁用报警指视报警不存在于控制器中一样处理报警。禁用报警不能够将报警状态传输或记录在历史数据库中。在 **Factory Talk View SE** 软件中的 **Alarm Status Explorer** 中，仍就追踪禁用报警，并可以将其重新使能。

优化运动控制应用

介绍

Logix5000 控制器包含一个高速运动任务，该任务可以执行运动命令(梯形图和结构化文本)和生成位置和速度轮廓信息。控制器将轮廓信息发送到一个或多个运动模块。RSLogix 5000 编程软件提供了完整的轴组态并支持运动编程。

关于运动的更多信息，请参阅：

- Motion Analyzer。
- 《Logix5000 控制器运动指令参考手册》，出版物 1756-RM00。
- 《ControlLogix 运动模块设置和组态手册》，1756-UM006。

粗略更新速率

粗略更新速率决定了运动任务执行的周期。在执行运动指令时，运动任务周期性计算发送到运动模块的伺服命令位置、速度和加速度。

要计算粗略更新速率：

- $2 \times (\text{任务执行时间} + \text{每个轴执行动作的数量})$ 。
- 将结果除以 1000，取整为最近的毫秒。

如果粗略速率太小，控制器可能没有时间来执行非运动的逻辑。通常，为了使控制器的执行时间合理，运动任务中的每个轴需要 1ms。

运动规划使用几乎是它的整个最小粗略迭代时间。每个轴的粗略迭代时间可最小设置为 1ms。因此，如果有一个周期型任务运行周期为 5ms，并且有 2 轴的运动，运动规划将运行 2 次，花费约为 5ms 时间段中的 4ms。在这种情况下，可能永远不能执行完成该周期型任务。

轴限制

控制器	支持的运动模块和轴数量	应用
ControlLogix	1756-M03SE(3 轴)	RA SERCOS 驱动器
	1756-L60M03SE(3 轴)	RA SERCOS 驱动器
	1756-L60 控制器内嵌 SERCOS 接口	
	1756-M08SE(8 轴)	RA SERCOS 驱动器
	1756-M016SE(16 轴)	RA SERCOS 驱动器
	1756-M02AE(2 轴)	RA 和第三方： • 模拟量命令信号 • 差分编码器反馈
	1756-HYD02 • 模拟量命令信号 • 线性传感器反馈	RA 和第三方：
	1756-M02AS • 模拟量命令信号 • SSI 反馈	RA 和第三方：
SoftLogix	1784-PM16SE(16 轴) • 每台计算机最多 4 块 1784-PM16SE 卡 • 一个控制器只能与一块 1784-PM16SE 卡相关联	RA SERCOS 驱动器
	1784-PM02AE(2 轴) • 每台计算机最多 4 块 1784-PM02AE 卡 • 最多可以有 4 块 1784-PM02AE 卡与控制器相关联 • 不能将一块 1784-PM02AE 运动卡与已经和 1784-PM16SE 卡相关联的同一控制器相关联	RA 和第三方： • 模拟量命令信号 • 差分编码器反馈

性能限制

运动规划可以中断所有其它任务，而不会考虑其权限。

- 运动组的轴数量和粗略更新周期影响运动规划执行的时间和频率。
- 如果运动规划正在执行时，一个任务被触发，那么该任务会在运动规划执行完毕再执行。
- 如果粗略更新速率发生在一个任务正在执行时，那么该任务会暂停并执行运动规划。

运动事件任务触发器

一个事件型任务会基于预先组态的发生事件自动执行。其中包含多种基于运动的事件。

当发生以下情况时， 触发一个事件型任务	使用触发器	需要考虑的事项
一个轴的注册输入打开 Registration	轴注册 1 或 2 (或关闭)。	• 为了利用注册输入来触发事件型任务，首先要执行一个 Motion Arm (MAR) 指令。这将使轴探测注册输入并依次触发事件型任务。 • 一旦注册输入触发了事件型任务，再次执行 MAR 指令可以使轴重新为下一个注册输入作准备。 • 如果正常逻辑的扫描时间没有足够快到使轴重新为下一个注册输入作准备，那么考虑将 MAR 指令放入事件型任务中执行。
轴到达定义为检测点 的位置。	轴检测	• 为了利用注册输入来触发事件型任务，首先执行一个 Motion Arm Watch (MAW) 指令。这将使轴探测检测位置并依次触发事件型任务。 • 一旦检测位置触发了事件型任务，再次执行 MAW 指令可以使轴重新为下一个检测位置作准备。 • 如果正常逻辑的扫描时间没有足够快到使轴重新为下一个检测位置作准备，那么考虑将 MAW 指令放入事件型任务中执行。
运动规划完成执行。	运动组执行	• 运动组的粗略更新周期会同时触发运动规划和事件型任务。 • 因为运动规划可以中断所有其它的任务，因此它首先执行。如果将事件型任务指定为最高优先权的任务，那么它会在运动规划后执行。

关于组态事件型任务的信息，请参阅第 2 章“将逻辑划分为任务、程序、例程和 Add-On 指令”。

注释:

优化使用 HMI 的应用

介绍

罗克韦尔自动化提供了以下的 HMI(人机界面)平台。

平台	描述
PanelView Plus 终端	专用的、机器级的 HMI，运行 RSVIEW Machine Edition 软件
RSVIEW Enterprise 软件	平台，产品家族包括： • 用于开放式、机器级的 RSVIEW ME (Machine Edition) 软件；也可运行在 PanelView Plus 终端 • 用于单工作站、管理层的 RSVIEW SE(Supervisory Edition) Station 软件 • 用于多服务器、多客户端和管理层 HMI 的 RSVIEW SE 分布式软件
RSVIEW32 软件	单工作站或单服务器、多客户端、监管级 HMI

软件产品为 HMI 应用提供了车间层设备连接，其中包括：

- RSLinx Classic 软件，也称为 RSLinx 2.x。
- RSLinx Enterprise 软件。

决定如何使用 HMI

方法	优势	考虑事项
单一 HMI	- 所有的 HMI/EOI 支持该方法 - 少数控制器连接 - 无服务器设置和管理 - 本地控制和监视	- 用于可视化的单点故障 - 每次只有一个人可以监视单一的显示画面
多个、独立的 HMI	- 所有的 HMI/EOI 支持该方法 - 同一 HMI 屏幕信息可以在多个站点查看 - 同时多人监视系统的各个部分 - 每个 HMI 获取其自身的数据 - 无中央服务器设置和管理 - 本地控制和监视	- 需要较多的控制器连接 - 服务所有的通讯(程序扫描影响)对控制器产生额外的负担 - 无数据共享, 除了经过控制器 - 添加额外的 HMI 可以进一步扩大系统
客户端 / 服务器 HMI	- 同一 HMI 屏幕信息可以在多个站点查看 - 服务器提供数据给多个客户端 - 需要较少的控制器连接 - 与具有多个 HMI 相比, 对系统的影响较小 - 在服务器中管理应用, 而不是单独在客户端或多个独立的 HMI	- 如果未使用冗余, 对于所有 HMI 而言, 服务器是单点故障 - 如果每个客户端需要不同的数据, 较少的通讯开销节省 - 需要网络知识

与以上多种 HMI 方法相似, 大多数第三方 HMI 仅限于直接通讯。

比较 RSView Enterprise 和 RSView32 软件

HMI 产品	优势	考虑事项
RSView SE	- 支持 Window 2000、Windows XP 和 Window Server 2003 操作系统 - 对于 RSView SE 和 RSView ME 软件 (包括 PanelView Plus 和 VersaView CE 终端)具有通用的 RSView Studio 开发环境 - FactoryTalk enabled	- 不支持 Windows NT 操作系统 - RSView SE 仍在开发某些 RSView32 功能
RSView32	- 支持 Windows NT、Windows 2000、Windows XP 和 Windows Server 2003 操作系统 - FactoryTalk enabled(7.0 版本和更新的) - RSView32 软件只支持单服务器架构	- RSView32 开发环境只支持 RSView32 软件 - 用于 PanelView 终端的 PanelBuilder 软件

RSView SE 软件指南

要成功设置 RSView Supervisory Edition HMI 系统，最多：

- 5 个 RSView Studio 客户端可以同时访问一个 RSView SE 应用项目。
- 50 个 RSView SE 客户端可以同时访问一个 RSView SE 应用项目。

在非冗余应用项目中，最多：

- 在一个 RSView SE 应用项目中可以有 10 个 RSView SE 服务器。
- 在一台计算机中可以有 2 个 RSView SE 服务器作为主机。

在冗余应用项目中，最多：

- 在一台计算机中可以有一个 RSView SE 服务器作为主机。

对于冗余服务器应用项目或需要多于 2 个 RSView SE 服务器和 20 个 RSView SE 客户端的应用项目，请联系 Rockwell Software。

RSLinx 软件如何与 Logix5000 控制器进行通讯

RSLinx 软件作为数据服务器优化与 HMI 应用项目的通讯。RSLinx 软件将数据项目组合成单一的网络信息包,从而减少通过网络发送和需要被控制器处理的报文数量。

重要事项

如果无其它说明，RSLinx 软件包括 RSLinx Classic 软件和 RSLinx Enterprise 软件。

1.

当 RSLinx 软件最初连接到一个 Logix5000 控制器时，它将查询标签数据库并上载所有控制器作用域标签的定义。如果在控制器作用域存在多层、用户自定义的结构体，RSLinx 软件只查询顶层。
2.

当 HMI 客户端需要数据时，RSLinx 软件查询程序范围标签的定义和多层用户自定义结构的底层。
3.

RSLinx 软件接收来自本地或远程 HMI/EOI 客户端数据项目的请求，并将多个请求在优化的信息包中结合。每个数据项目可以是一个简单的 Logix 标签、数组或用户自定义结构体。每个优化的信息包最大为 480 字节的数据，并且可以包含一个或多个数据项目。
4.

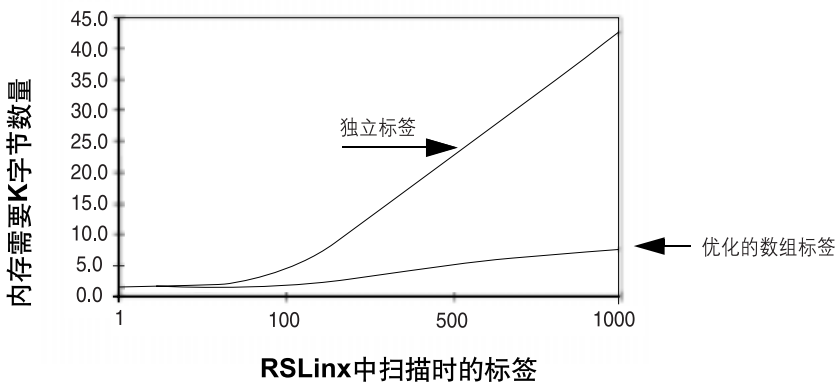
Logix5000 控制器将分配未使用的系统 RAM，从而创建一个包含请求数据项目的优化缓冲区。

— 一个优化缓冲区可以包含尽可能多的数据来适合一个 480 字节的信息包(优化限于 480 字节)。

— 目前，RSLinx Enterprise 软件只提供数据标签的优化。

— 如果使用 RSLogix 5000 软件来监控控制器 RAM，可以看到使用的内存增加。

— 在扫描中，控制器为每个 RSLinx 优化信息包创建一个优化缓冲区。



比较 RSLinx Classic 和 RSLinx Enterprise 软件

比较	RSLinx Classic (RSLinx 2.x) 软件	RSLinx Enterprise 软件
支持平台	- Windows 98 - Windows ME - Windows NT - Windows 2000 - Windows XP - Windows Server 2003	- Windows CE - Windows 2000 - Windows XP - Windows Server 2003
架构	单线程	多线程
数据服务器	OPC 数据服务器 对于 PLC/SLC 平台和需要复杂网络路由的应用项目，其可作为首选数据服务器 每个数据服务器最多包含 10 个客户端	FactoryTalk Live 数据服务器 对于 Logix5000 平台，其可作为首选数据服务器 每个数据服务器最多包含 20 个客户端
PLC/SLC 系统	通过以太网，每个数据服务器最多可以包含 20 个控制器	通过以太网，每个数据服务器最多可以包含 20 个控制器
Logix5000 系统	最多： - 通过以太网，每个数据服务器可以包含 10 个控制器 - 每个数据服务器可以有 10000 个活动(处于扫描状态)的标签 - 每个控制器可以有 3 个 RSLinx 数据服务器	最多： - 通过以太网，每个数据服务器可以包含 20 个控制器 - 每个数据服务器可以有 20000 个活动(处于扫描状态)的标签 - 每个控制器可以有 3 个 RSLinx 数据服务器
用户接口和事件日志	是	- 可获得的用户接口是 FactoryTalk Studio 软件和 FactoryTalk Administration Console 软件 - 事件日志仍处于开发中
优势	- 支持与冗余 ControlLogix 系统切换的主题 - 支持用户自定义标签优化 - RSLinx Gateway 软件合并了多个 HMI 请求，从而减少了网络通讯量	- 使用 4 个读和 1 个写单方向连接(与 RSLinx 软件相比较少) - 自动处理 Logix 标签改变 - FactoryTalk Live Data 软件合并了多个 HMI 请求，从而减少了网络通讯量
考虑事项	- 如果 Logix5000 控制器重新进行了加载而且扫描时标签发生了改变，则需要 HMI 重新启动 - 使用 4 个双方向连接 - 目前不支持 OPC - ActiveX 面板需要一个独立的 OPC 服务器	- 不支持在冗余 ControlLogix 系统中的主题切换 - 仅限于数组标签优化

RSLinx 软件指南

指南	描述
使用 RSLinx 软件作为多个 HMI 的数据服务器。	对于多个 HMI 站点： • 平衡了远程 OPC(RSLinx Classic 软件)或 FactoryTalk(RSLinx Enterprise 软件)软件用于数据采集。 • 只有 RSLinx 数据服务器应该有一个激活的主题。 • 不可以组态或使用 HMI 站点中的主题。 • 在 HMI 站点不需要 RSLinx 软件。
不能使用过多的 RSLinx 站点。	标签采集的性能会随着从同一控制器采集数据的 RSLinx 站点数量的增加而下降。 使用一个 RSLinx Gateway 站点和使另一个数据采集站点使用远程 OPC 进行数据采集。
当添加 / 移除扫描的标签时，解决延迟时间。	当从一个 HMI 屏幕信息切换至另一个时，需要花费时间将条目放入控制器进行扫描和将条目从控制器中取出停止扫描。时间延迟的部分是由于控制器为优化缓冲区分配系统 RAM。 为了消除该延迟，当进行 HMI 屏幕信息切换时，将条目放入 HMI 屏幕信息中进行扫描，并留下它们进行扫描。例如，可以创建一个数据日志来保持条目处于扫描状态。然后当进行 HMI 屏幕信息切换时，数据采集会继续进行而无需中断。 RSLinx Enterprise 和 RSView SE 软件解决了该时间延迟。当 HMI 屏幕信息改变时，这些应用项目会使标签无效，而不是将其从扫描中移除。

组态控制器标签指南

指南	描述
与第三方产品使用 INT 数据类型。	多数第三方操作员界面产品不支持 DINT(32 位)数据类型。 然而，在使用 INT 数据类型时，还需要考虑额外的性能和内存使用事项。请参阅 42 页数据类型使用指南。 RSView 软件支持 Logix5000 具有的数据类型(包括 BOOL、SINT、INT、DINT 和 REAL)、结构体和数组。
在数组中组合相关数据。	多数第三方操作员界面产品不支持用户自定义结构体。数组同样可以确保数据在邻近的内存中，这样优化了控制器和 RSLinx 软件或其它操作员界面之间的数据传输。 数组标签的传输更加快速，并且与独立标签的组合相比，占有更少的内存。
将标签映射到 PLC 地址中。	为了优化控制器和 RSLinx 软件或其它操作员界面之间的数据传输，使用 PLC 映射的标签。 RSLinx 主题必须使 Optimize Poke Packets 使能。 RSView 应用项目必须通过 DownloadALL 命令或 WritingPendingValues VBA 方法写入数值。
使用 RSLinx OPC 服务。	使用 RSLinx OPC 服务将多个标签请求捆扎成一个报文，从而减少通讯开销。 OPC 提供的优化优于 DDE。

在 RSView 软件中引用控制器数据

下表说明了如何引用 RSView 标签地址中的数据。

Logix5000 数组数据类型	描述	PLC 文件标识符	RSView 标签数据类型
BOOL	数值为 0、1 或 -1	B	数字式
SINT	8 位整型	A	字节
INT	16 位整型	N	整型
DINT	32 位整型	L	长整型
LINT	用于存储日期和时间数值的 64 位整型数值	无 PLC 标识符	不支持
REAL	浮点型	F	浮点型

当分配一个 Logix5000 字符串标签时，使用地址的语法[OPC_Topic]StringTag.Data[0]。SC82 来分配一个 SINT 数组。该字符串数据存储在字符串标签的 SINT 数组.Data 中，而且用户分配数组中第一个元素的地址(.Data[0])。在一个 STRING 标签中的最大字符数量是 82。如果需要更多的字符，创建用户自定义结构体来容纳这些字符。

如果要写入数据到一个 HMI 或外部源的 Logix5000 字符串标签，必须设置 LEN 域来说明字符串中字符的数量。RSLogix 5000 软件 and 控制器使用 LEN 数值来确定出现的字符数量。

注释:

为批处理控制开发设备状态

介绍

RSLogix 5000 软件的 PhaseManager 选件(在 15 版本中引入)为用户设备提供一个状态模型。它包括以下的组件：

- 运行状态模型的阶段(Phase)
- 用于编程的设备阶段(Phase)指令
- PHASE 数据类型

Equipment Phases 指南

指南	描述
设备活动使用各自独立的阶段(Phase)。	阶段(Phase)是设备执行的一个特定活动。 • 单个机器使用一个阶段(Phase)。 • 确保每个阶段(Phase)进行一个独立的活动。 • 确保项目中的阶段(Phase)和程序总数处于控制器程序的限制内。 • 列出运行某阶段(Phase)的设备。
每阶段(Phase)完成一个状态模型。	每个阶段(Phase)运行它自身的一组状态。状态模型将设备的运行周期划分成一系列状态。 • 决定上电后哪个状态用于起始状态。 • 以起始状态开始，并且运行在整个模型中。 • 只使用所需的状态；跳过那些不需使用的状态。 • 使用子例程来运行和旁路状态切换。 设备阶段(Phase)的状态模型与 S88 状态模型(U.S.标准 ISA S88.01-1995)相似，并且它的 IEC 同类标准 IEC61512-1-1998 通常被认为是 S88。它是一组适合于制造业系统设计和运行的模型、术语和良好的实践。
把设备控制与阶段(Phase)控制代码分开。	阶段(Phase)的优势，即针对如何制造产品，它可以让用户把设备控制代码与制造产品的生产工序(配方)分开。这样使得用同一台设备为生产不同的产品而执行不同的工序步骤变得更加容易。

指南	描述
把异常情况与正常执行分离。	状态模型使得从正常操作与一些异常情况(故障、失效和非正常条件)分离更加容易。 - 针对故障，使用一个预先状态例程进行检测。 预先状态例程不是一个阶段(Phase)状态例程。创建预先状态例程和创建任一程序步骤雷同。 - 使用状态位来限制代码程序在一特定状态。 - RSLogix 5000软件会自动为每个状态创建一个标签。该状态标签都有确定阶段(Phase)状态的标志位。例如，My_Phase.Running。
只能非冗余控制器中使用 Equipment Phase	ControlLogix 控制器冗余需要大量额外的验证确认。因此，PhaseManager 选项只通过了在非冗余控制器环境中测试。如果需要冗余控制器，使用 Phase Logic Interface(PLI)应用代码或将 Equipment Phase 程序下载到非冗余控制器。

Equipment Phase 指令

设备阶段(Phase)指令适用于梯形图和结构化文本编程语言。用户可以在梯形图例程、结构化文本例程和 SFC 动作中使用它们。

如果	使用指令
向设备阶段(Phase)发出信号，指示状态例程已完成因而进入下一状态。	阶段(Phase)状态完成 (PSC)
更改设备阶段(Phase)的状态或子状态	设备阶段(Phase)命令(PCMD)
发出信号表明设备阶段(Phase)失效	设备阶段(Phase)失效 (PFL)
清除设备阶段(Phase)的失效代码	设备阶段(Phase)清除失效 (PCLF)
启动与 RSBizWare Batch 软件的通讯	设备阶段(Phase)外部请求 (PXRQ)
清除设备阶段(Phase)的 NewInputParameter 位	设备阶段(Phase)新参数 (PRNP)
在设备阶段(Phase)逻辑中设置断点	设备阶段(Phase)暂停 (PPD)
占有阶段(Phase)操作权 - 防止其它程序或 RSBizWare Batch 软件向设备阶段(Phase)发命令。 - 确保其它程序或 RSBizWare Batch 软件未占用设备阶段(Phase)。	附加到设备阶段(Phase) (PATT)
释放阶段(Phase)的所属权	脱离设备阶段(Phase) (PDET)
重写一个命令	设备阶段(Phase)重写(POVR)

更多的信息，请参阅《PhaseManager 用户手册》，LOGIX-UM001。

优化过程控制应用

介绍

Logix5000控制器将功能块图表编辑器和多个过程控制指令结合起来。控制器通常可以执行比典型的应用所需的更多回路。

比较 PID 和 PIDE 指令

功能块 PIDE 指令提供高于梯形图 PID 指令额外的性能提升。

增强型 PID (PIDE)	标准 PID
增量式算法在偏差值改变时工作 该算法与与多数 DCS 系统所使用的类型相同。 同时也可以使执行适应增益更加容易。	位置式算法随着偏差值工作
全套模式 - 程序 / 操作员控制 - 串级 / 比例 - 自动 - 手动 - 超驰 - 手操	有限的模式 - 自动 - 软件手册(与 PIDE 手动模式相似) - 手动(与 PIDE 手操模式相似)
可选的时间选择模式 - 周期型 - 过采样 - 实时采样	无时间选择模式
处理 PV/CV 故障 PIDE 块具有内嵌的 PVFault 和 CVFault 成员。	不能处理 PV/CV 故障
完全无扰动切换到串级模式或者再切换回来	不能无扰动切换到串级模式或者再切换回来

编程 PID 回路指南

指南	描述
将 PID 回路放入周期型任务中。	组态周期型任务以期望的速率执行。
基于任务执行时间估计回路数量	如下估计可以执行的 PID 回路数量： (周期型任务的执行时间，以 ms 为单位)/2 这可以为控制器管理低权限任务中其它逻辑留出足够的时间。

对于典型的应用，估计每个回路有 1ms 的程序扫描时间。其中包括 PID 指令和其它与回路相关的代码。

高级过程指令

指令	描述
报警(ALM)	为任意模拟信号提供报警功能。
增强型 PID(PIDE)	在标准的 PID 指令之上提供增强的功能。该指令使用 PID 算法的速度公式。增益值应用于偏差或 PV 的变化值，而不是偏差或 PV 值。
斜坡 / 渗透(RMPS)	提供一定数量的交互的斜坡和渗透时间段。
整定(SCL)	将一个非整定的输入值转换成浮点型工程量。
位置比例(POSP)	以用户定义的周期时间,使用与期望位置 and 实际位置差值成比例的脉冲宽度周期打开或关闭触点来打开或关闭设备。
脉宽调制(SRTP)	将 PID 回路的 0-100% 的输出转化成周期性脉冲信号来驱动加热和制冷的数字量输出触点。
超前 - 滞后补偿(LDLG)	为输入信号提供一个超前 - 滞后的相位补偿。
函数发生器(FGEN)	基于分段线性函数对输入进行转换。
累加器(TOT)	累加一定时间范围内的模拟量输入值。
死区时间(DEDT)	实现单个输入信号延时功能。用户可以选择死区延时的时间长度。
离散式 2 态设备(D2SD)	控制只有两种状态的离散型设备，例如只有开 / 关和打开 / 闭合。
离散式 3 态设备(D3SD)	控制具有三种状态的离散型设备，例如快 / 慢 / 停，正转 / 停止 / 反转等。

面板

RSLogix5000 软件包括某些功能块指令的面板。这些面板是可以读取整个数据结构的 Active-X 控件。与 RSView 软件和任何其它应用使用

重要事项

RSLogix 5000 编程软件不是一个有效的 Active-X 容器。

这些面板通过RSLinx Classic软件中的RSLinx OPC服务器与控制器进行通讯。RSLinx OPC 服务器不能在与 RSLogix 5000 软件一起使用的 RSLinx Lite 软件中获得。用户必须购买一个软件包，例如 RSLinx Classic OEM、Professional 或 Gateway 软件。

以下指令具有面板：

- 报警(ALM)
- 增强型选择(ESEL)
- 累加器(TOT)
- 斜坡 / 渗透(RMPS)
- 离散型 2 态设备(D2SD)
- 离散型 3 态设备(D3SD)
- 增强型 PID(PIDE)

FactoryTalk View 软件，CPR7 或更新版本，也包含一组与 FactoryTalk View SE 和 ME 应用项目一起使用的指令面板。这些面板要求 RSLinx Enterprise 软件，并且使用 FactoryTalk Studio 开发环境中固有的对象。这些面板可以与 FactoryTalk Studio 软件一同安装。

比较 Active-X 面板和图形库组件

Active-X 面板	图形库组件
Requires RSLinx Classic 软件	需要 RSLinx Enterprise 软件
适合于与 Active-X 容器一起使用，例如 RSView 32 或 Microsoft Office 产品	适合于与 RSView SE 或 RSView ME 软件一起使用
只在个人计算机平台支持	支持所有的管理版和机器版平台
面板将复选框和单选按钮组合	图形将按钮和文本框组合
不允许用户自定义	允许用户自定义

管理固件

介绍

Logix 控制器、I/O 模块和其它设备均使用可以由用户自行升级的固件。用户可以选择固件版本级别，并决定何时更新固件。

管理控制器固件指南

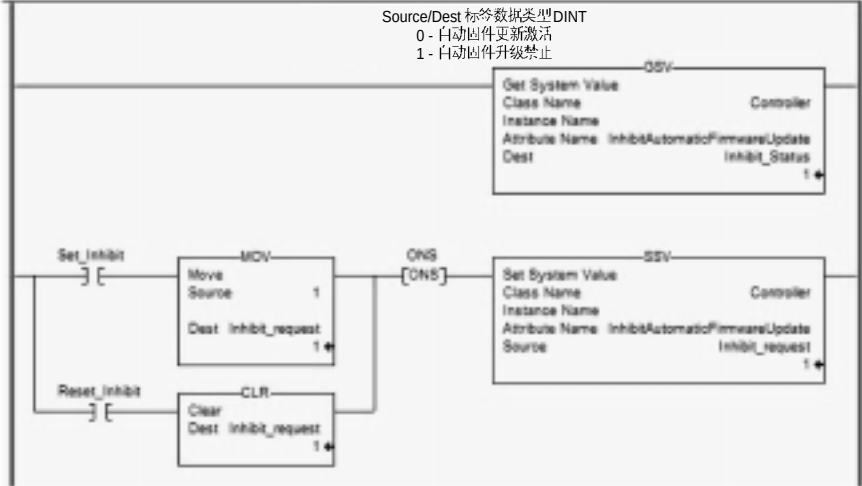
指南	描述						
将软件版本和固件版保持在相同的主版本级别。	在版本发布时，软件的特殊版本支持固件特殊版本中的特点和功能。要使用特殊版本的固件，用户必须有响应的软件版本。应考虑软件和硬件结合的兼容性问题。 版本发布号包括主版本和次版本号，格式为 xx.yy。 <table><tr><th>何处</th><th>是</th></tr><tr><td>xx</td><td>主版本 更新每个发布版本时，会有一个功能变化</td></tr><tr><td>yy</td><td>次版本 任何时间更新时，会有变化，该变化不会影响功能和界面</td></tr></table>	何处	是	xx	主版本 更新每个发布版本时，会有一个功能变化	yy	次版本 任何时间更新时，会有变化，该变化不会影响功能和界面
何处	是						
xx	主版本 更新每个发布版本时，会有一个功能变化						
yy	次版本 任何时间更新时，会有变化，该变化不会影响功能和界面						
文档固件版本。	在电子绘图和其它的工程文档中包括软件版本和固件版本信息。						
阅读相关的发布文件。	在安装前必须阅读附带新软件版本和固件版本的发布文件。这些发布文件帮助用户理解改进和变化的部分，并且可以帮只用户确定是否需要因产生的变化而修改应用项目。在多数情况下，应用项目将在更新后正常运行。						
组态模块，以便控制器自动更新固件。	控制器固件， 16 版本，包含的固件管理特点使得控制器可以自动快速更新设备。要使用固件管理器： • 当本地和远程模块处于编程或运行模式时，只要它们的电子锁组态设置为精确匹配，并且它们均被 ControlFlash 应用程序支持，那么可以进行快速更新。 • 固件工具箱必须位于控制器的 CompactFlash 卡中。						

RSLogix 5000 固件管理器的指南

从控制器固件 16 版本开始，当用户更换系统中的设备时，RSLogix 5000 固件管理器可以自动加载固件。

- OEM，一个月建造多种机器，可以使控制器快速更新系统中的所有模块和设备，而无需用户干涉。
- 具有严格规则的机器需要特殊的固件版本对设备进行维护证明。固件管理器帮助确保设备处于正确的固件版本。
- 维护人员更换发生故障的硬件，安装代替的设备，并且控制器将自动用正确的固件版本更新该设备。

指南	描述
固件管理器将快速更新任何罗克韦尔自动化的设备： • 设备可以位于 I/O 组态树下。 • 设备使电子锁组态为精确匹配。 • 通过使用 ControlFlash 软件被正常更新设备。	固件管理器可以作用于本地 I/O 模块和 EtherNet/IP、SERCOS 和 ControlNet 网络上的分布式模块。在 DeviceNet 网络上，固件管理器只支持本地设备，例如扫描器和位于控制器工程 I/O 树下的连接设备。由于用户不能直接将远程 DeviceNet 设备放到 I/O 树下，因此固件管理器不能管理远程 DeviceNet 设备。 固件管理器支持： • 支持 CompactFlash 记忆卡的 Logix5000 控制器，除了 GuardLogix 安全控制器和 ControlLogix 冗余控制器。 固件管理器不能管理在 I/O 组态树下的其它控制器的固件。这些控制器通过其自带的 CompactFlash 卡来管理固件。 • SERCOS 伺服驱动器支持通过 SERCOS 网络快速更新。这包括： - 1394 伺服驱动器，固件版本 1.85 和更新。 - Kinetix 6000 伺服驱动器，固件版本 1.85 和更新。 - Ultra3000 伺服驱动器，固件版本 1.50 和更新。 - 8720MC 伺服驱动器，固件版本 3.85 和更新。 • 直接为网络中以部分的非模块式、分布式 I/O 产品，无需适配器。不支持需要适配器的分布式 I/O 产品，例如 POINT I/O 或 FLEX I/O 模块。然而，固件管理器可以管理适配器的固件。 PanelView Plus 终端不支持 ControlFlash 软件，因此固件管理器不支持它们。
对于管理设备固件的固件管理器，设备的电子锁必须组态为精确匹配。	未组态为精确匹配的其它模块可能存在于 I/O 组态中，但是固件管理器将不会维护这些模块的固件。 对于一个特殊设备，要禁止固件管理器： • 将该设备的电子锁改为其它模式，除了精确匹配。 • 通过 SSV 指令或控制器属性的非易失性内存选项卡禁止固件管理器。
CompactFlash 卡必须格式化为 FAT16。	如果 CompactFlash 卡格式化为 FAT32，必须将其重新格式化为 FAT16。

指南	描述
每个控制器必须存储固件文件，用于在 CompactFlash 中由固件管理器管理模块	通过控制器属性的非易失性内存选项卡，使能固件管理器。单击 Load/Store。通过自动固件升级下拉菜单，选择 Store 将它复制到 CompactFlash 卡中。 计算机运行 RSLogix5000 软件必须具有： - 已安装的 ControlFlash 应用程序 - 在 ControlFlash 缺省目录下所需的固件工具箱，适用于固件管理器要维护的模块。 RSLogix 5000 软件将固件工具箱从用户的计算机中移动到控制器 CompactFlash 卡中，提供给固件管理器使用。 控制器固件和应用逻辑通过在非易失性内存选项卡中的固件管理器管理。固件管理器增加该功能来存储控制器固件和 CompactFlash 中的逻辑。如果用户禁止固件管理器，则只能禁止固件管理器更新，而不是控制器固件更新。当控制器映像重新载入时，控制器固件更新仍然会出现。
通过使用 GSV 和 SSV 指令使能或禁止自动固件更新。	
用户可以管理自动固件更新的状态。	在控制器属性的非易失性内存选项卡中管理自动固件更新的状态。 要对一个特殊模块的自动固件更新状态进行管理，使用 GSV 指令。以下示例说明了固件管理器遇到的 1756-OB16D 模块错误固件版本。 

访问固件

RSLogix 5000 软件具有固件更新工具箱。固件版本也可在罗克韦尔自动化网站获得。

1. 进入 support.rockwellautomation.com
2. 在左侧 Download 下，单击 Firmware updates。

注释:

A

术语	定义
Add-On 指令	Add-On 指令是用户自定义指令，其将可执行逻辑和数据进行压缩。
数组	同一名称下具有相同数据类型的序列数据。一个数组标签在控制器中占用连续的内存块，每个元素按顺序排列。
原子数据类型	BOOL、SINT、INT、DINT、LINT 和 REAL 数据类型。

B

术语	定义
缓冲区	临时内存区域用于将进入和输出的报文排队。设备的缓冲区域决定了可以排队处理的报文数量。

C

术语	定义
高速缓冲存储器	为一个反复执行的 MSG 指令留下一个开放式连接。
粗略更新速率	决定了运动任务执行的周期。在执行运动指令时，运动任务周期性计算发送到运动模块的伺服命令位置、速度和加速度。
复合数据类型	数组、结构体和字符串数据类型。
连接	2个设备之间的通讯链路，例如在一个控制器和一个 I/O 模块、PanelView 终端或另一个控制器之间。 - 连接是资源的分配，其可以在设备之间提供比未连接报文更可靠的通讯。 - 用户通过组态控制器与系统中的其它设备通讯，可以间接地确定控制器所使用的连接数量。
消费者标签	该标签接收由生产者标签通过 EtherNet/IP 网络、ControlNet 网络或 ControlLogix 背板广播的数据。一个消费者标签必须是： - 控制器作用域。 - 与远程标签(生产标签)相同的数据类型(包括任意数组维数)。 请参阅生产者标签。
连续型任务	连续型任务在后台连续运行。所有未分配给其它操作(例如运动、通讯和周期型任务)的 CPU 时间都将用于执行连续型任务中的程序。
控制器作用域	在控制器中的任意处均可访问的数据。控制器包含可以被任意程序中的例程和别名标签，以及控制器作用域中别名标签引用的一组标签。 请参阅程序范围。

D

术语	定义
直接连接	直接连接是在控制器和 I/O 模块之间的一种实时数据传输链路。控制器维护并监控其与模块之间的连接。此连接的任何中断，例如模块故障或者在带电状态下从移除该模块，都将使得控制器在与模块相关联的数据区中的故障位置位。 请参阅机架优化连接。

E

术语	定义
元素	可分配地址的单元数据是大单元数据中的子单元。数组或结构体中的单一单元。
设备阶段(Phase)	设备阶段(Phase)是一种类型的程序。它具有例程和一组独立的标签。它还具有： • 状态模型。 • 状态机。 • PHASE 数据类型。
事件型任务	基于一个发生的触发事件或如果在一个特定时间间隔触发事件没有发生，那么事件型任务自动执行。
显性	一种对时间无苛刻要求，并且本身是请求/回答的连接。执行一个MSG 指令或执行一个程序上载均为显性连接。显性引用的基本信息(例如源地址、数据类型和目的地址)包含于每个报文中。 请参阅隐性。

F

术语	定义
固件版本	对于具有固件成份的产品而言，通过产品ID 标记可以识别固件版本。版本可以表示出设备所使用的操作系统。固件版本通常有 2 个号码，通过句点将它们分开。例如，固件版本 10.02，第一个号码(10)定义了主版本，第二个号码(02)定义了次版本。 请参阅软件版本。

H

术语	定义
HART 协议	HART(可寻址远程传感器高速通道)是一种连接模拟量设备的开放式协议。对于HART 的连通性，可以从罗克韦尔自动化及其周边合作伙伴的产品中选择。

I

术语	定义
隐性	一种本身对时间具有苛刻要求的连接。其包括I/O 和生产者/消费者标签。隐性引用的信息(例如源地址、数据类型和目的地址)在报文中暗指出，但没有包含于报文中。 请参阅显性。
索引	用于执行数组中元素的引用。

L

术语	定义
本地连接	一种与本地框架、扩展本地框架，或用于控制器组态的I/O 组中任意的一个模块的连接。通讯在背板、或虚拟背板中进行，并且不需要额外的通讯模块或适配器。

M

术语	定义
成员	结构体的元素具有其自身的数据类型和名称。 - 成员也可以是结构体，可以创建嵌入式结构数据类型。 - 结构体中的每个成员可以是不同的数据。
报文	报文可以异步地将块数据读取或写入到另一个设备中。

N

术语	定义
网络刷新时间(NUT)	在 ControlNet 网络中可以发送数据的重复时间间隔。网络刷新时间的范围是 2 到 100 毫秒。

P

术语	定义
参数	参数是传到指令中或从指令中返回的数值或标签。Add-On 指令支持以下参数： - 输入(复制进入) - 输出(复制输出) - 入出(经过引用)
周期型任务	周期型任务基于预先组态的时间间隔自动执行。该任务与 PLC-5 和 SLC 500 处理器中可选的定时中断相似。
PhaseManager 选项	RSLogix 5000 软件(在 15 版本中引入)的 PhaseManager 选项为用户设备提供了一个状态模型。使用 PhaseManager 选项来船检设备阶段(Phase)程序。
快速扫描	控制器的一种功能，即在禁止程序之前，为了复位指令和数据，检查程序中的逻辑。
预扫描	预扫描是在转换到运行模式下的一种中间过程扫描。 - 当用户将控制器从编程模式转换为运行模式，控制器执行预扫描。 - 预扫描检查所有的程序和指令，并且初始化基于结果的数据。 - 一些指令在预扫描过程中的执行会与在正常扫描过程的执行有所不同。
生产者标签	由一个控制器生成，并且可以被其它控制器使用的标签。生产者标签总是在控制器作用域。 请参阅消费者标签。
产品定义数据类型	一种由软件 and 控制器自动定义的结构体数据类型。通过组态 I/O 模块，用户将为模块添加产品定义数据类型。
程序	组态相关的例程和标签，每个程序包含程序标签、主执行例程、其它例程和可选的故障例程。
程序范围	只可在当前程序中访问数据。每个程序包含一组只可被例程和该程序中的别名标签引用的标签。 请参阅控制器作用域。

R

术语	定义
机架优化连接	对于数字量 I/O 模块，用户可以选择机架优化通讯。一个机架优化连接巩固了在框架(或 DIN 导轨)上的控制器和 I/O 模块之间的连接使用。其胜于对每个 I/O 模块使用独立、直接连接。机架优化连接为整个框架(或 DIN 导轨)提供了一个连接。
远程连接	一种与远程框架或 DIN 导轨上的模块进行的连接。通讯需要通讯模块和 / 或适配器。
请求数据包间隔(RPI)	当通过网络进行通讯， - 通常，该间隔组态以毫秒为单位。 - 实际生产的数据受限于网络刷新时间的最大倍数，网络刷新时间小于所选的 RPI。
例程	用单一编程语言编写的一组逻辑指令，例如梯形图。例程为控制器中(与 PLC 或 SLC 控制器中的程序文件相似)的工程提供了可执行代码。

S

术语	定义
规划连接	规划连接是 ControlNet 通讯独有的一种连接方式。规划连接允许用户以预先确定的速率反复发送和接收数据，预先确定的速率即为请求数据包间隔(RPI)。例如，与 I/O 模块的连接是规划连接，因为用户能够以设定的速率反复从模块接收数据。其它的规划连接包括： - 与通讯设备的连接。 - 生产者 / 消费者标签。 在 ControlNet 网络中，用户必须使用 RSNetwork for ControlNet 软件来使能所有的规划连接，并设定网络刷新时间。
软件版本	软件产品的产品 ID 标记标识了软件版本。版本表示了软件的功能版本。软件版本通常有 2 个号码，通过句点将它们分开。例如，在软件版本 10.02 中，第一个号码(10)定义了主版本，第二个号码(02)定义了次版本。 请参阅固件版本。
状态机	状态机： - 对一个激活的状态调用主例程(状态例程)。 - 使用最小的编码管理状态之间的转变。 - 确保设备沿着允许的路径经由逐个状态。
状态模型	状态模型将用户设备的运行周期分成一系列状态。每个状态在设备的运行中都是一个瞬态。它是在假设时间段内设备的动作或条件。
结构体	某些数据类型是结构体。 - 结构体存储了一组数据，它们中的每一个可以是不同的数据类型。 - 在一个结构体中，每个独立的数据类型被成为一个成员。 - 与标签一样，成员均有一个名称和数据类型。 - 用户可以通过使用单独标签和多数其它结构体的组合创建用户自定义结构体。 - 要复制数据到结构体，使用 COP 指令。
系统内务处理时间片	明确说明了用于通讯和后台功能(系统支出)的控制器时间的百分比(除周期型任务时间之外)。

T

术语	定义
标签	控制器内存中用于存储数据的命名区域。标签是分配内存、逻辑中的引用数据和管理数据的基础结构。
任务	执行程序的规划机制。缺省情况下, 每个新工程文件包含一个预先组态好的连续型任务。用户可以根据需要组态另外的周期和事件型任务。

U

术语	定义
面向非连接的消息	面向非连接的消息是一种不需要连接资源的消息。面向非连接的消息作为单一的请求/响应发送。
用户自定义数据类型(UDT)	UDT 是用户定义的数据结构体。用户自定义数据类型将不同类型的数据组成一个单一名称的统一体。用户可以定义用户自定义数据类型的成员。与标签一样, 成员具有名称和数据类型。

V

术语	定义
虚拟通讯关系(VCR)	VCR 是为基金会现场总线设备之间提供数据传输的通道。发送数据和接收数据要求的 VCR 数量依赖于使用的设备和数据类型。VCR 的类型决定了传输是否为规划或非规划。 • 客户端/服务器 VCR 适合于排队的、非规划的、用户初始化的和一对一的通讯。 • 报告分发 VCR 适合于排队的、非规划的、用户初始化的和一对多的通讯。 • 发布方/接收方 VCR 适合于缓冲的、一对多的通讯。

注释：

出版物

用户可以在 <http://www.literature.rockwellautomation.com> 网站查看或者下载出版物。要订购印刷版的技术文档, 请与用户当地的罗克韦尔自动化分销商或者销售代表取得联系。

RSLogix 5000 编程软件中包含了 PDF 版本的下列出版物, 以及在线帮助和指南。

资源	说明
《Logix5000 控制系统的 EtherNet/IP 模块用户手册》, ENET-UM001 《Logix5000 控制系统的 ControlNet 模块用户手册》, CNET-UM001 《Logix5000 控制系统的 DeviceNet 模块用户手册》, DNET-UM004	网络
《Logix5000 控制器快速入门》, 1756-QS001 《Logix5000 通用过程编程手册》, 1756-PM001 《SFC 和 ST 编程语言编程手册》, 1756-PM003(本手册是从《Logix5000 控制器通用过程编程手册》中摘录出来的) 《Logix5000 控制器通用指令参考手册》, 1756-RM003 《Logix5000 控制器过程控制和驱动指令参考手册》, 1756-RM006 《Phase Manager 用户手册》, LOGIX-UM001 《Logix5000 控制器运动控制指令参考手册》, 1756-RM007 《Logix5000 控制器系统参考手册》, 1756-QS107 《Logix5000 控制器导入 / 导出参考手册》, 1756-RM084G 《将 PLC-5 或者 SLC 500 逻辑转换成 Logix5000 逻辑参考手册》, 1756-RM085	Logix5000 控制器
《ControlLogix 控制器安装指南》, 1756-IN101 《ControlLogix 系统用户手册》, 1756-UM531 《ControlLogix 运动控制模块设置和组态手册》, 1756-UM006	ControlLogix 控制器
《1768 CompactLogix 控制器安装指南》, 1768-IN004 《1768 CompactLogix 系统用户手册》, 1768-UM001	1768 CompactLogix 控制器
《1769-L31 CompactLogix 控制器安装指南》, 1769-IN069 《1769-L32E, -L35E CompactLogix 控制器安装指南》, 1769-IN020 《1769-L32C, -L35CR CompactLogix 控制器安装指南》, 1769-IN070 《CompactLogix 系统用户手册》, 1769-UM011	1769 CompactLogix 控制器
《FlexLogix 控制器安装指南》, 1794-IN002 《FlexLogix 系统用户手册》, 1794-UM001	FlexLogix 控制器
《SoftLogix 控制器安装指南》, 1789-IN001 《SoftLogix 系统用户手册》, 1789-UM002	SoftLogix 控制器

网站

资源	说明
http://www.ab.com/logix/	Logix 产品信息
http://www.ab.com/networks/	NetLinx 产品信息
http://support.rockwellautomation.com 在 Downloads 下的左侧面板中, 选择 Software Updates。	软件更新(需要产品序列号)
http://support.rockwellautomation.com 在 Downloads 下的左侧面板中, 选择 Firmware Updates。	固件更新(需要产品序列号)
http://www.ab.com/networks/eds/	罗克韦尔自动化 EDS 文件
http://literature.rockwellautomation.com	罗克韦尔自动化手册
http://samplecode.rockwellautomation.com	RSLogix 5000 示例代码

注释:

罗克韦尔自动化的技术支持

罗克韦尔自动化在网站上提供技术信息帮助用户使用自己的产品。在 <http://support.rockwellautomation.com> 网站上，用户可以找到技术手册、FAQs 知识库、技术和应用记录、实例代码和与软件补丁包的链接而且用户可以自定义 MySupport 功能以更好的使用这些工具。

对于附加级别用于安装、组态和诊断的电话技术支持，我们提供 TechConnect 支持程序。如果需要更多信息，请于本地分销商或罗克韦尔自动化代表联系，或访问网站 <http://support.rockwellautomation.com> 网站。

安装助理

如果用户在安装硬件模块的最开始的 24 小时里遇到了问题，请重新阅读该手册中的相关信息。用户还可以联系专用的客户支持电话号码，该号码为用户模块启动和运行提供初始帮助：

美国	1.440.646.3223 星期一到星期五，早 8 点到晚 5 点(美国东部时间)
美国境外	任何技术支持问题，请联系当地罗克韦尔自动化代表

新产品满意返厂

罗克韦尔对全部产品进行测试，以确保它们出厂后可以稳定运行。然而，如果用户的产品不运行并且需要返厂：

美国	联系分销商。用户必须向分销商提供一个用户支持代码(使用上面的电话号码来获得一个)以便完成退货过程。
美国境外	请与用户本地罗克韦尔自动化公司代表联系办理返厂手续

www.rockwellautomation.com.cn

动力、控制与信息解决方案

Americas: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1)414 382.2000, Fax: (1)414 382.4444
亚太地区 - 香港数码港道 100 号数码港 3 座 F 区 14 楼 电话: (852)28874788 传真: (852)25109436

北京 - 北京市建国门内大街 18 号恒基中心办公楼 1 座 4 层 邮编: 100005 电话: (8610)65217888 传真: (8610)65217999
天津 - 天津市河西区南京路 20 号金皇大厦写字楼 3816 室 邮编: 300042 电话: (8622)23312285 传真: (8622)23312265
青岛 - 青岛市香港中路 40 号数码港旗舰大厦 2206 室 邮编: 266071 电话: (86532)86678338 传真: (86532)86678339
西安 - 西安市高新区科技路 33 号高新国际商务中心数码大厦 1201 室 邮编: 710075 电话: (8629)88152488 传真: (8629)88152466
郑州 - 郑州市中原中路 220 号裕达国际贸易中心 A 座 1216-1218 室 邮编: 450007 电话: (86371)67803366 传真: (86371)67803388
太原 - 山西省太原市府西街 69 号山西国际贸易中心 B 座 8 层 801 室 邮编: 030002 电话: (86351)8689580 传真: (86451)8689580
上海 - 上海市仙霞路 319 号远东国际广场 A 幢 7 楼 邮编: 200051 电话: (8621)61206007 传真: (8621)62351099
南京 - 南京市中山南路 49 号商茂世纪广场 44 楼 A3-A4 座 邮编: 210005 电话: (8625)86890445 传真: (8625)86890142
武汉 - 武汉市建设大道 568 号新世界国贸大厦 I 座 2202 室 邮编: 430022 电话: (8627)68850233 传真: (8627)68850232
长沙 - 长沙市韶山路 159 号通程国际大酒店 1712 室 邮编: 410011 电话: (86371)5450233/5456233 传真: (86371)5456233 ext. 608
杭州 - 杭州市杭大路 15 号嘉华国际商务中心 1203 室 邮编: 310007 电话: (86571)87260588 传真: (86571)87260599
广州 - 广州市环市东路 362 号好世界广场 2703-04 室 邮编: 510060 电话: (8620)83849977 传真: (8620)83849989
深圳 - 深圳市深南东路 5047 号深圳发展银行大厦 15L 邮编: 518001 电话: (86755)25847099 传真: (86755)25870900
厦门 - 厦门市湖里区湖里大道 41 号联泰大厦 4A 单元西侧 邮编: 361006 电话: (86592)2655888 传真: (86592)2655999
南宁 - 南宁市民族大道 92-1 号新城国际大厦 1415 室 邮编: 530000 电话: (86771) 5536784 传真: (86771)5534713
成都 - 成都市总府路 2 号时代广场 A 座 906 室 邮编: 610016 电话: (8628)86726886 传真: (8628)68726887
重庆 - 重庆市渝中区邹容路 68 号大都会大厦 3112-13 室 邮编: 400010 电话: (8623)63702668 传真: (8623)63702558
昆明 - 昆明市东风西路 123 号三合商利写字楼 13 层 C 座 邮编: 650000 电话: (86871)3635448/ 3635458/ 3635468 传真: (86871)3635428
沈阳 - 沈阳市沈河区青年大街 219 号新华国际大厦 15-F 单元 邮编: 110015 电话: (8624)23961518 传真: (8624)23963539
大连 - 大连市西岗区中山路 147 号森茂大厦 2305 层 邮编: 116011 电话: (86411)83687799 传真: (86411)83679970
哈尔滨 - 哈尔滨市南岗区红军街 15 号奥威斯发展大厦七层 E 座 邮编: 150001 电话: (86451)84879066 传真: (86451)84879088

