



GE Fanuc Automation

可编程控制产品

Series 90™-30
DeviceNet模块

GFK-2196

中国工控网

www.chinakong.com

资料中心

November 2002

章 1

系列 90-30 PLC DeviceNet 模块

本手册的信息

第 1 章: 系列 90-30 PLC 的 DeviceNet 模块, 介绍 Series 90™-30 系列 DeviceNet 主站模块 (IC693DNM200) 和 系列 90-30 系列 DeviceNet 从站模块(IC693DNS201)的信息.

第 2 章: 安装, 讨论电源消耗, 电缆规格, 接地, 安装模块, 和 LED 指示灯含义.

第 3 章: DeviceNet 主站的 PLC 配置, 解释如何在 90-30 系列 PLC 配置内添加一个 DeviceNet 主站模块信息. 还解释如何配置通讯连接, 如何配置模块以及如何配置 DeviceNet 网络.

第 4 章: DeviceNet 从站的 PLC 配置, 解释如何在 90-30 系列 PLC 配置内添加一个 DeviceNet 从站模块信息.

第 5 章: 模块工作原理, 说明 90-30 系列的 DeviceNet 主站模块和系列 90-30 DeviceNet 从站模块的功能.

第 6 章: 通讯编程(COMMREQs), 解释如何在应用程序中使用和读取状态信息.

第 7 章: 系列 90-30 模块的 DeviceNet Objects, 数目 系列 90-30 系列 DeviceNet 模块中定义的信息对象.

附录 A: DeviceNet EDS 文件, 包含了 90-30 系列的 DeviceNet 主站模块和系列 90-30 DeviceNet 从站模块的电子数据表(EDS) 文件.

关于 DeviceNet 的细节的信息

关于 DeviceNet 更细节的信息, 请与 Open DeviceNet Vendor Association 联系.

Open DeviceNet Vendor Association, Inc.
20423 State Road 7
Suite 499
Boca Raton, FL. 33498
phone: (954) 340-5412
FAX: (954) 340-5413
Internet: [HTTP://WWW.ODVA.ORG](http://WWW.ODVA.ORG)
Email: <ODVA@POWERINTERNET.COM>

系列 90-30PLC 使用的 DeviceNet 模块

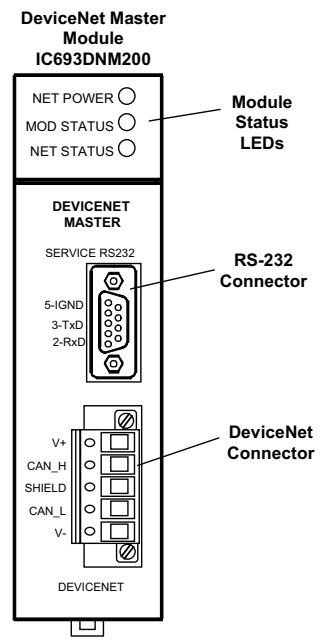
系列 90-30PLC 使用的 DeviceNet 模块共有两类:

- 90-30 系列 DeviceNet 主站模块(IC693DNM200) 用作网络的主站.它和网络上最多 63 台设备交换数据和发送自定义消息. 在有更高级别计算机的 DeviceNet 网络中, 这个模块的内置服务器功能可以使用自动数据传送和自定义消息和主站通讯.
- 90-30 系列 DeviceNet 从站模块 (IC693DNS201) 用作网络的从站(从站). 它能够自动地和网络主站交换 PLC 数据, 并且应答主站发送的自定义消息.

这两种模块均可安装到 系列 90-30PLC 的任意机架. 每块模块都可成为 DeviceNet 网络上的单独节点.

90-30 系列 DeviceNet 主站模块

90-30 系列 DeviceNet 主站模块 (IC693DNM200) 允许 90-30 系列 CPU 通过 DeviceNet 网络收发数据. 它作为主站最多可与 63 个 DeviceNet 从站通讯. 它还可通过配置同时既作主站又作其他主站的从站.



本模块有三种 DeviceNetLED 指示灯,这些指示灯可以显示它的工作和通讯状态. 在升级固件时, RS-232 串口 (9 针 D 形接口) 用作和计算机连接. DeviceNet 连接器是一个可拆卸的弹簧端子. 它可以和总线连接, 也可在不需要总线时取下.

特性

- 支持所有标准 DeviceNet 通讯速率 (125k, 250k, 500k 波特率)
- 每个从站最多 255 个字节输入/255 个字节输出, 每个主站最多 3972 个字节输入/3972 个字节输出.
- 支持每从站一个代理连接的非连接消息管理器 (UCMM)
- 每个从站可配一至两个 I/O 连接. 每个从站的 I/O 连接可以设置成以下模式: 定时询问, 选通, 循环或状态改变(COS). 通常一个连接被设置成定时询问而另一个的设置成选通、循环或 COS.
- 对于定时询问和 COS/循环设备可以分别设置更新频率
- 全局扫描频率可配置
- PLC-应用程序使用 COMMREQ 命令 启动
- 和从站通讯的状态可在 PLC 故障表(可配置)中找到. 提供 64 个网络设备状态位
- 可配置当通讯丢失时采取的故障动作

系列90-30 DeviceNet 主站模块的规格

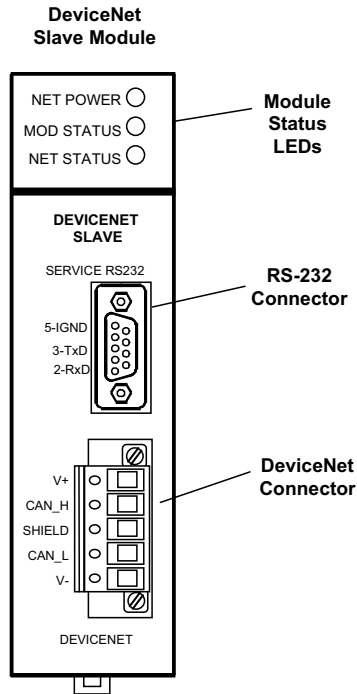
产品目录编号	IC693DNM200
描述	90-30 系列 DeviceNet 网主站模块
安装位置	90-30 系列背板 (CPU, 扩展, 或远程) 上除第 1 槽的其他任何槽位
环境温度	存储温度: -40°C 至 85°C 工作温度: 0°C 至 60°C
背板电流消耗	450mA 5VDC (通常情况)
通讯速率	支持所有标准的 DeviceNet 通讯速率(125k, 250k, 和 500k 波特率)

兼容性

- 除了IC693CPU321 和IC693CPU340以外,兼容所有系列 90-30CPU.CPU311/313/331 对于配置大小的限制在第 3 章会有详细的说明.
- 要求 CPU 固件的版本是 8.0 . 如果 CPU 的版本有 10.0, 则推荐使用.
- 要求 SIMPLICITY Machine Edition Logic Developer PLC 编程软件版本是 3.0 以上, 且安装了 Service Pack for DeviceNet.
- 与 VersaPro™, Control, or Logicmaster™ 编程软件不兼容.
- 90-30 系列手持编程器 (IC693PRG300) 不能配置本模块.

系列 90-30 DeviceNet 从站模块

系列 90-30 DeviceNet 从站模块 (IC693DNS201) 作为 系列 90-30 PLC 的 DeviceNet 网络接口，由其他主站设备控制。



本模块有三种 DeviceNet LED 指示灯,这些指示灯可以显示它的工作和通讯状态. 在升级固件时, RS-232 串口 (9 针 D 形接口) 用作和计算机连接. The DeviceNet 连接器是一个可拆卸的弹簧端子. 用它可以和总线连接, 也可在不需要总线时取下.

特性

- 支持所有标准 DeviceNet 通讯速率(125k, 250k, 500k 波特率)
- 最多 255 个字节输入 / 255 个字节输出.
- 可配置的通讯模式: 定时询问,选通, 循环和 COS I/O 连接,
- 支持 Unconnected Message Manager (UCMM), 允许同时最多有 250 个 explicit messaging 连接.
- 可以配置一个至两个 I/O 连接和 explicit messaging.每个 I/O 连接可以设置成以下模式: 定时询问,选通, 循环或状态改变(COS). 通常一个连接被设置成定时询问而另一个的设置成选通、循环或 COS..
- 支持 集合对象和访问每个配置过的从站的输入输出数据 (I/O 1 区和 I/O 2 区) 使用 SET_ATTRIBUTE_SINGLE 和 GET_ATTRIBUTE_SINGLE 服务可以将从站 I/O 区分成输入和输出两个区(I/O 1 区和 I/O 2 区). 使用 SET_ATTRIBUTE_SINGLE 可以配置最多 255 个字节的属性.
- 可配置当通讯丢失时采取的故障动作.

- UCMM-capable Group 2 服务器

系列90-30DeviceNet 从站模块的规格

产品目录编号	IC693DNS201
描述	90-30 系列 DeviceNet 从站模块
安装位置	90-30 系列背板 (CPU, 扩展, 或远程) 上除第 1 槽的其他任何槽位
环境温度	存储温度: -40°C 至 85°C 工作温度: 0°C 至 60°C
背板电流消耗	5VDC 450mA (典型)

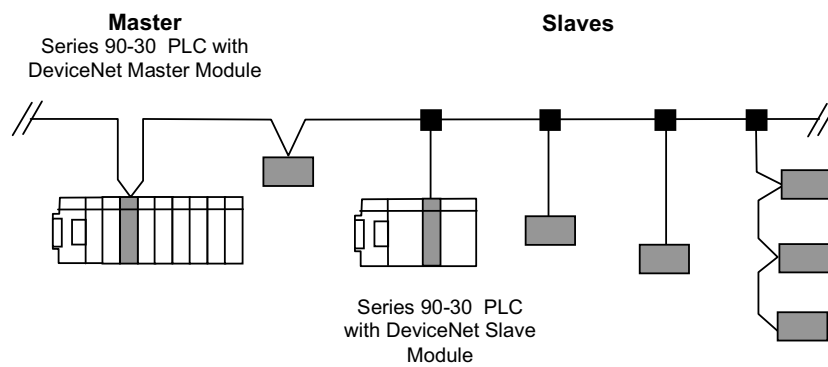
兼容性

- 兼容除了IC693CPU321 和IC693CPU340以外所有系列 90-30CPU.
- 要求 CPU 固件的版本是 8.0. 如果 CPU 的版本有 10.0, 则推荐使用.
- 需要 CIMPLICITY Machine Edition Logic Developer PLC 编程软件 3.0 版本以上, 且装有 Service Pack for DeviceNet, or later.
- 不兼容 VersaPro™, Control, or Logicmaster™ 编程软件.
- 系列 90-30 手持编程器 (IC693PRG300) 不能用于配置这个模块.

DeviceNet 网络

DeviceNet 是一个在控制系统(如: PLCs, PCs, VMEbus computers, and 机器人控制器) 与 分布式工业设备如开关,传感器, 阀, 电机启动器,条码读取器,驱动器,显示,及操作接口之间交换数据的通讯网络。网络还可连接智能接口模块如 VersaPoint DeviceNet 网络接口单元 (NIU) 和 VersaMax DeviceNet NIU. NIU 是远程 I/O 站的网络接口模块。

DeviceNet 网络 为线型结构. 由主干线路和连到设备的支路构成. 电源和信号都在一根电缆中传输. 系列 90-30DeviceNet 模块可以直接连到主干也可连到支路还可以使用级联连接。



系列 90-30 DeviceNet 模块的 DeviceNet 通讯

DeviceNet 使用生产者-消费者的通讯技术. 拥有数据的设备向网络上发送数据. 所有需要这些数据的设备都监听消息. 当设备识别出恰当的表示符, 它就使用这些数. 消息不指定具体的源或目的, 一个消息可以被多个设备使用. 例如, 一个消息可以同时控制多台电机的启动装置.

一个 DeviceNet 消息字段可以包含 0 至 8 个字节. 消息长于 8 个字节时需要分段. 系列 90-30 DeviceNet 模块保证每个网络节点的数据完整性.

I/O Messaging

I/O Messaging 用于在设备间自动交换数据. 单个 I/O 消息最多可有 255 个字节. I/O 消息建立一个生产数据设备或多个消费数据的设备间的专用通讯路径. 系统配置中设定好这些生产数据和消费数据设备间的连接参数. 一旦连接建立, 设备就自动地进行通讯.

I/O 消息有四种基本类型: 定时询问, 选通, 循环, 以及 状态变换(COS). 系列 90-30 DeviceNet 主站模块可以给每个从站设备设置一至两种类型的 I/O 通讯连接(例如, 一个循环 I/O 通讯连接和一个状态变化 I/O 通讯连接). 详见 第 3 章, "在 PLC 中配置 DeviceNet 主站".

Explicit Messaging

Explicit messaging 提供两个设备间的点对点的通讯连接. Explicit messaging 经常用于从站配置和诊断. DeviceNet 协议定义了 Explicit messages . 对于系列 90-30 的 DeviceNet 主站模块, Communications Request (COMMREQs) 用于发送 Explicit messages. 详见第 6 章, "可编程通讯". 除了那些使用 COMMREQ 进行用户自定义对象的 DeviceNet 模块, 所有的模块都会自动接收 Explicit message .

章 2

安装

本章介绍 系列 90-30 DeviceNet 模块的基本安装说明:

- 电源要求: PLC 系统电源和 DeviceNet 网络电源
- DeviceNet 电缆: 技术规格, 长度, 终端, 夹子
- 接地: DeviceNet 电缆, DeviceNet 电源, DeviceNet 系统
- 安装 DeviceNet 模块 in the PLC
- 模块指示灯 LEDs: Module Status, Network Status, Network Power
- RS-232 串口

附加信息

正确安装电缆、电源、和其他网络硬件需要了解一些 DeviceNet 技术规范的细节. 推荐读者们访问www.ODVA.org查找详细信息.

标准一致性

在安装 GE Fanuc 产品的场合, 需要按照联邦通讯委员会,加拿大通讯局或欧盟的标准。请参考 GE Fanuc 的*标准一致性的安装要求*, GFK-1179.

电源

使用系列 90-30 DeviceNet 模块,需要考虑使用两个单独的电源: PLC 电源和 DeviceNet 网络电源.

电源

系列 90-30 DeviceNet 模块需要 PLC 背板提供 450mA 5VDC (典型).推荐使用高容量的系列 90-30 电源如 IC693PWR330 或 IC693PWR331, 尤其是在使用 CPU350 及更高型号的 CPU 时, 或 PLC 还有以太网模块和/或多块 DeviceNet 模块时.

CIMPLICITY Machine Edition 配置软件会自动计算出 PLC 电源的负荷. 如果估算电源负荷请参考 GFK-0356, 系列 90-30 产品安装和硬件手册.

DeviceNet 电源

系列 90-30 DeviceNet 模块需要网络电源为其网络收发器提供 24VDC. DeviceNet 网络电源推荐使用线性电源. DeviceNet 电源不得同时作为设备电源使用. 如果不按要求使用, I/O 设备可能导致网络出错甚至网络瘫痪.

DeviceNet 规范推荐使用电源端子 将电源和网络连接起来. 电源端子 应该选择和电缆容量的相当的. 网络电流的最大值取决于电缆的类型.

系列 90-30 模块 需要消耗 DeviceNet 网上的 24VDC 的 xxxma 电流 (典型).

粗缆电流限制

粗缆网络电流最大值为 16 安培. 不过,每个单独网段的电流仅能有 8 安培. 在两个网段中央位置接入一个可供 16 安的独立电源, 由它为每个网段提供 8 安的电流.

细缆电缆限制

细缆电流的最大值为 3 安培.

系列 90-30 模块 DeviceNet 电缆

系列 90-30 DeviceNet 模块既可以用 DeviceNet 粗缆也可以用 DeviceNet 细缆。粗缆可以有更长的距离和更高的电流容量。一般,粗缆用于主干电缆。细缆一般比较短适合于分支电缆及布线要求灵活的场合。

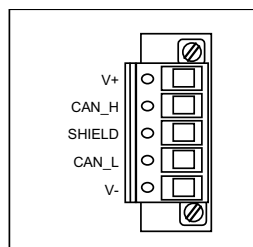
粗缆和细缆都是 5 芯, 多导线铜缆。两根用于网络通讯。另外两根用于传导网络电流。第 5 号根线 形成电磁屏蔽层。大多数电缆用颜色分别芯的不同, 其颜色和系列 90-30 DeviceNet 模块端子上的颜色一一对应。

电缆和网络的技术规格

粗缆基本规格	两对屏蔽信号线- 同轴屏蔽电缆
	屏蔽编织层- 65%遮盖率; 最小 36 AWG 或 0.12 毫米 的镀锡铜缆(独立镀锡)
	屏蔽线- 最小#18 铜.; 最小 19 股(独立镀锡)
	外径- 0.410 英寸(最小)到 0.490 英寸(最大.) 圆形半径偏差控制在 0.5 外径的 15%以内.
细缆基本规格	两对屏蔽信号线- 同轴屏蔽电缆
	屏蔽编织层- 65%遮盖率; 最小 36 AWG 或 0.12 毫米 的镀锡铜缆(独立镀锡)
	屏蔽线- 最小#22 铜.; 最小 19 股(独立镀锡)
	外径- 0.240 英寸(最小)到 0.280 英寸(最大.) 圆形半径偏差控制在 0.5 外径的 20%以内.
网络拓扑结构	分支限制的总线型(主干线路/分支线路)
冗余	不支持
节点设备的网络电源	常规 24 V 直流电源误差 ±4%
允许节点数 (不带网桥)	64 个
数据包长度	0-8 个字节, 允许信息分段
冲突地址检测	带上电时地址验证功能
错误检测/ 纠错	CRC – 在没有收到确认信息时重发消息

总线连接器的端子分布

系列 90-30 DeviceNet 模块的 DeviceNet 连接器带 5 个有颜色区分的螺丝卡紧的端子。连接器用于连接总线; 可以在不中断网络工作的情况下取下模块。



信号	端子	颜色
V+	5	Red
CAN_H	4	White
Shield	3	Bare
CAN_L	2	Blue
V-	1	Black

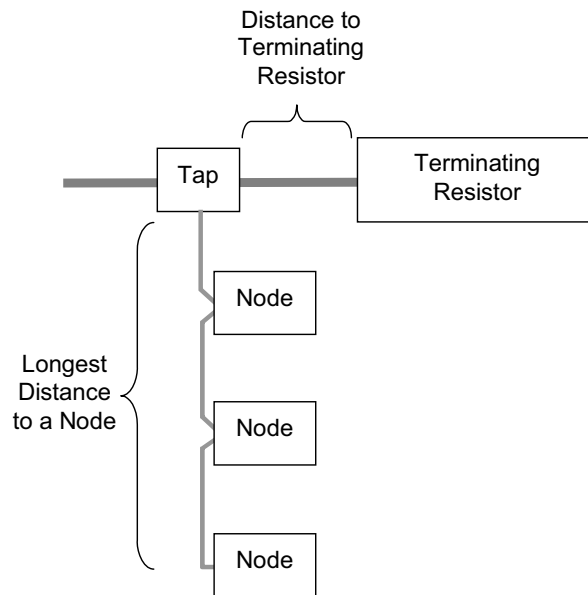
总线长度

主干电缆和支线电缆的总长度取决于电缆类型和通讯速率. 单个分支电缆不能超过 6 米, 而且每个分支仅可一个网络节点. 不过, 节点可以是多端口的.

通讯速率	125kbps	250kbps	500kbps
粗缆, 主干长度	500m (1640ft)	250m (820ft)	100m (328ft)
细缆, 主干长度	100m (328ft)	100m (328ft)	100m (328ft)
分支线路长度最大值	6m (20ft)	6m (20ft)	6m (20ft)
全部支路的总长度	156m (512ft)	78m (256ft)	39m (128ft)

每个波特率下的分支线路总长度是网络中两种电缆所有支路的长度之和.

另外, 如果 tap 到最远设备的距离长于 tap 到最近的终端电阻, 就像下图所示, 支线的长度也计算到主干电缆长度里面(即全部分支长度).

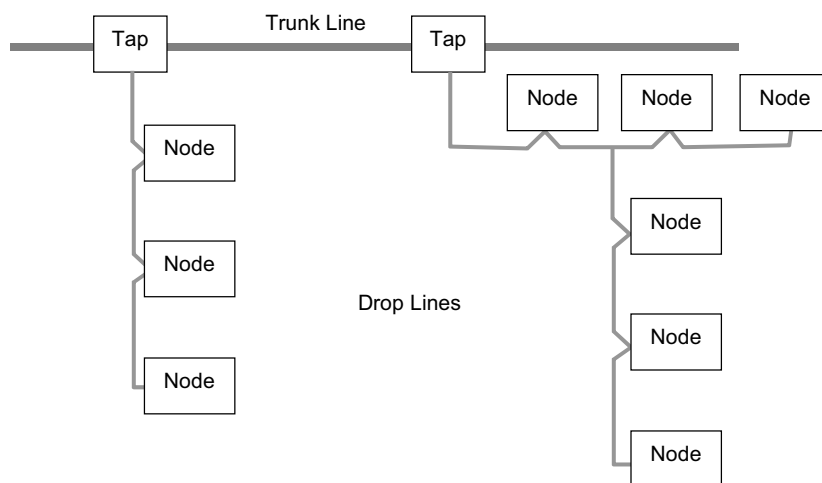


网络终端

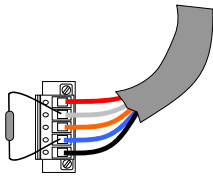
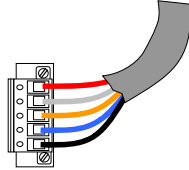
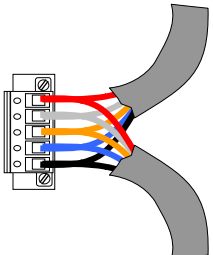
在 DeviceNet 网的两端必须安装 121 欧姆, 1% ¼瓦的终端电阻. 每个终端电阻要连接到数据通讯信号的 2 针 (CAN_L) 和 4 针 (CAN_H).

夹子 级联与分支

Devices 设备可以直接连到主干电缆, 也可以连到与主干电缆相连的支路上. 夹子 可以装在分线箱或面板内. 支线和级联经常在控制柜内使用, 而这些控制柜内有多个设备相邻. 使用级联式支线 和分支式支线, 从 tap 至最远端不能超过 20 英尺.



系列 90-30 的 DeviceNet 主站模块接线方式由它在网络所处的位置决定:

<p>如果系列 90-30 DeviceNet 模块位于主干线路的一端, it is 它需要接一个根电缆和一个终端电阻:</p> 	<p>如果模块安装在支线或支段的一端, 它只需要接一根电缆.</p> 	<p>如果模块直接安装到主干电缆上或作为支路级联电缆的一部分, 它要把进线和出线电缆都接上:</p> 
--	---	--

接地

DeviceNet 电缆接地

和设备连接的所有 DeviceNet 屏蔽层都要接地。可以把屏蔽层接到连接器的 3 针(屏蔽)上。

DeviceNet 电源接地

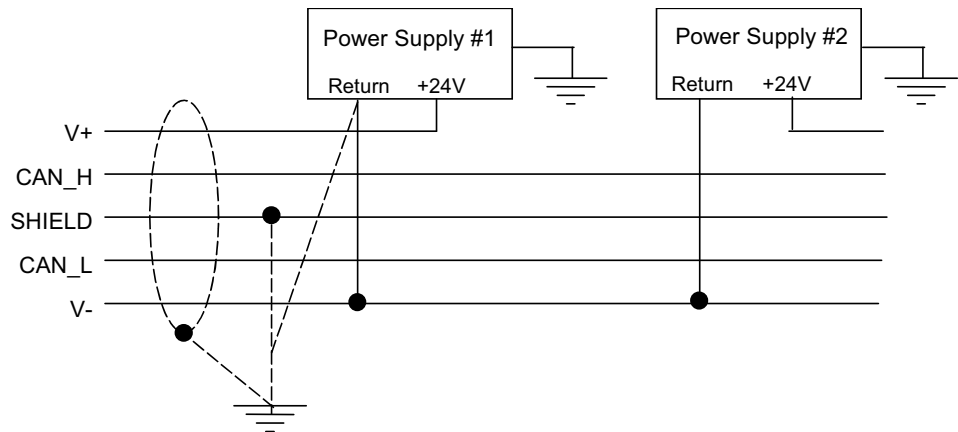
DeviceNet 网络电源也必须接地, 而且只能接在一个地上。The V- 信号必须接到电源的保护地。如果有多个电源,只能是接到 V-的那个电源接到各自得地上。

DeviceNet 系统接地

DeviceNet 通讯只能接到一个地上。基本上是接到电源所处的控制柜内的地上。

DeviceNet 电源 (-V)线, 屏蔽线(裸线) 和电缆屏蔽层必须直接接到地上。。理想的情况是在中央位置接地。连接线应该使用 25 毫米(1 英寸.)的编织铜线或#8 AWG 线, 其长度不能超过 3 米(10 英尺.)。

下图显示了一个有两个电源的网络如何接地。每个电源的底盘都接到地上。



地线尺寸

系列 90-30 DeviceNet 模块上的 DeviceNet 螺钉接线端子的接地导线的最小尺寸是 2.5mm² (14 AWG) 线。其他网络设备可能需要更粗的电缆。

PLC 机架中安装 DeviceNet 模块

系列 90-30 DeviceNet 模块可以安装到主机架的第 2 槽及以后的槽位上，也可安装到扩展机架的第 1 槽及以后的槽位上。

1. 切断机架电源。
2. 拿住模块的顶部顺着机架上的凹槽向下用力，直到模块牢固地插到底板上。
3. 将 DeviceNet 电缆连接到模块。
4. 根据需要终结网络。

注意：关于安装系列 90-30 机架系统和模块，参考系列 90-30 安装手册和硬件手册，GFK-0356。

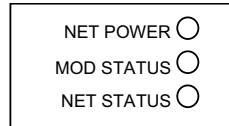
取下机架里的模块

若取下机架里的模块，按照以下步骤操作：

1. 切断机架电源。
2. 去掉模块上连接的电缆。
3. 按照模块底部的解锁扣缓慢地把模块从下部抬起脱离底槽。

模块指示灯 LEDs

模块有三个指示灯 LEDs，显示工作和通讯状态：



网络电源指示灯 ED

指示灯 LED	含义
红色	没有检测到网络电源.
绿色	检测到网络电源.

模块状态值 ED

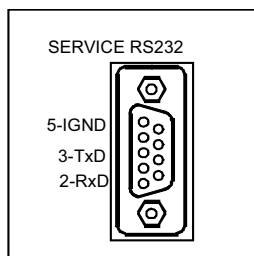
指示灯 LED	含义
熄灭	底板电源没有送电.
绿色	模块正常工作.
闪烁绿色	模块处于待机状态. 当它的配置可能丢失、不完整、不正确时. 模块可能处于待机状态.
闪烁红色	可恢复故障
红色	模块遇到不可恢复的故障; 可能需要复位或更换.
红绿交替闪烁	模块正在自检.

网络状态指示灯 ED

指示灯 LED	含义
熄灭	<ul style="list-style-type: none"> ▪ 模块离线, 或 ▪ 模块没有完成 MACID 冲突检测, 或 ▪ 模块没有供电. 参见模块状态指示灯 LED.
闪烁绿色	<ul style="list-style-type: none"> ▪ 模块在线但是没有连接处于连接成功状态, 或 ▪ 模块通过 MACID 冲突检测, 而且已经在线, 但是没有和其他节点建立连接.
绿色	模块已经在线并且有连接处于连接成功状态.
闪烁红色	有 I/O 连接处于超时状态.
红色	模块不能在网上通讯.
红绿交替闪烁	模块检测到一个网络访问错误并处于通讯故障状态.

RS-232 串口

RS-232 串口 是一个 9-针 D 形连接器. 这个端口在升级固件时与计算机相连.



升级 DeviceNet 模块的固件

当有新的固件可用时,可以按照如下步骤升级模块.

1. 下载升级的固件到计算机. 可以到 GE Fanuc 网站www.gefanuc.com/support/plc/ 查找升级固件
2. 使用直通串口电缆将计算机 和 DeviceNet 模块面板上的串口连接起来. 仅用 RX 和 TX 线电缆, 要符合 RS232 转 RS485 kit (IC693ACC903).
3. 可以利用计算机里的标准通讯软件与模块通讯. 模块支持 19200 波特率, 无奇偶校验, 8 位数据位, 1 位停止位, 以及无流控制.
4. 切断 模块所在的系列 90-30 PLC 机架电源, 再上电.
5. 在计算机端, 按 ENTER 键直到看到初始开机界面. 开机界面显示引导程序正在等待新的固件. 在通讯软件内使用 Xmodem 发送模块的固件文件. 注意: 你必须在再次上电时按 ENTER 键. 推荐电源开关一合, 就按住 ENTER 键.
6. 如果模块完成保存固件的工作, 它会自动地重启并且启用新固件.
7. 如果固件没有成功传输, 开机界面再次出现. 重新传输.

如果升级成功, 断开串口电缆, 切断系列 90-30 PLC 机架电源后再上电.

章 3

DeviceNet 主站的 PLC 配置

本章解释如何在系列 90-30 PLC 配置中添加系列 90-30 DeviceNet 主站模块 (IC693DNM200)。也会解释如何配置系列 90-30 DeviceNet 主站模块和 DeviceNet 网的连接。

- 配置步骤
- 在 PLC 配置中添加一块 DeviceNet 主站模块
- 配置 DeviceNet 主站模块的参数
 - DeviceNet 主站模块的参数
 - DeviceNet 主站模块的网络参数
- 给 DeviceNet 主站配置添加从站信息
 - 添加 Device 的 EDS 文件
- 配置添加到主站的从站网络设置
 - 分配 MAC ID 和 波特率
 - 配置 I/O 通讯连接
 - 配置 DeviceNet Explicit 通讯
- 配置 DeviceNet 主站作为从站工作时的网络参数
 - 分配 MAC ID 和波特率
 - 配置 DeviceNet 主站作为从站工作时的 I/O 通讯连接
 - 配置 DeviceNet Explicit 通讯

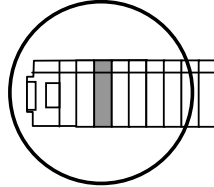
这些配置过程针对基本熟悉 CIMPLICITY Machine Edition Logic Developer 编程软件 和系列 90-30 PLC 的用户所写。如何使用编程软件, 请见编程软件的附带帮助系统。

注意: DeviceNet 主站仅在 CIMPLICITY Machine Edition Logic Developer 编程软件中得到支持。Logic 90™, VersaPro™, 和 Control 软件 均不支持这些系列 90-30 DeviceNet 模块。

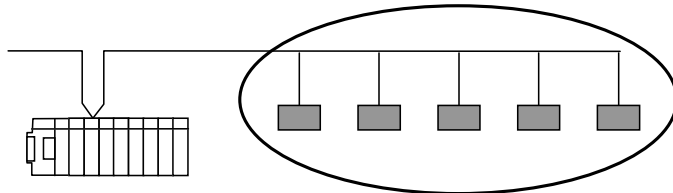
配置步骤

配置 DeviceNet 主站模块需三个基本步骤:

- 把模块插到 PLC 机架并配置它的工作参数.



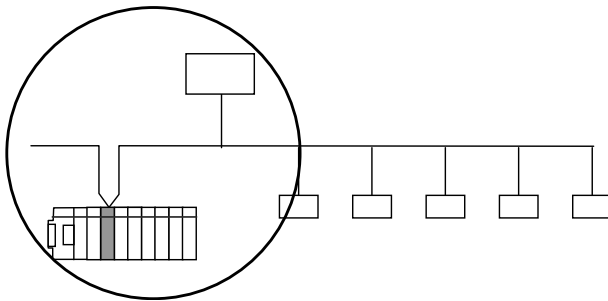
- 修改 DeviceNet 主站需要访问的从站信息和它们的网络设置参数



从站设备的数量和类型以及它们所能交换数据的数量取决于 CPU 的内存大小. CPU 的内存数量取决于 CPU 的型号、CPU 固件的版本、其他模块的数量和类型、从站地数量和类型以及与扩展编程器 HMI 设备通讯的数量和类型. 使用 Logic Developer-PLC 软件, 当前配置的大小可以从硬件配置的“数据 View”窗口找到并且可以添加各个组件的大小. LD-PLC 程序不允许配置超过 65,535 个字节. DeviceNet 配置的大小还受下列模板的用户配置空间的限制:

CPU 311/313	4,736 个字节可用
CPU 331	4,673 个字节可用

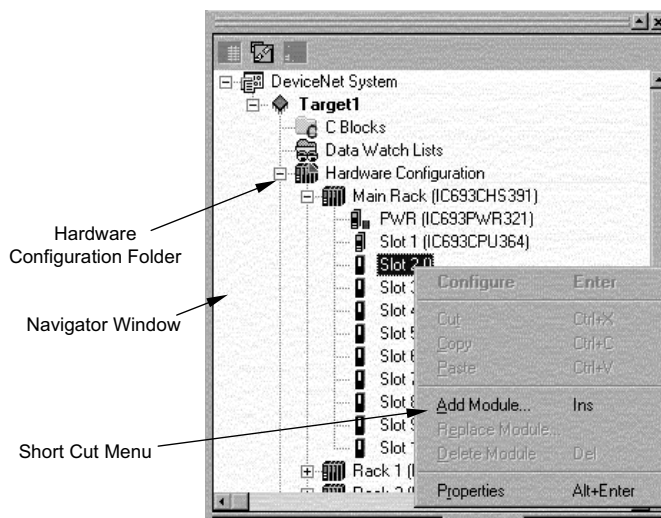
- (可选) 配置 DeviceNet 主站模块 自己的网络设置. 这种情况仅在 DeviceNet 主站模块 作为另一个网络主站的服务器工作时才需要. 例如, 可能和更高级的主机控制器交换它带的从站数据.



在 PLC 配置中添加 DeviceNet 主站模块

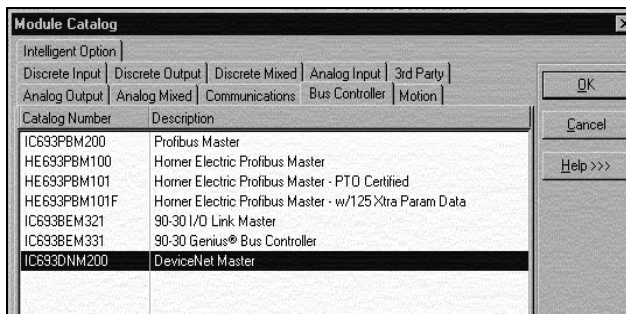
首先, 在 PLC 机架配置中添加系列 90-30 DeviceNet 主站模块 (IC693DNM200) . 这个模块除了 IC693CPU321 和 IC693CPU340 以外, 兼容所有型号的系列 90-30 CPU . 它要求 CPU 的固件版本至少是 8.0 . 如果你使用的版本有 10.6 或更高, 推荐使用.

1. 在 Navigator 窗口的工程页中, 展开硬件配置文件夹.
2. 在硬件配置文件夹内, 在想添加 DeviceNet 主站模块的 PLC 槽位右击鼠标 .
3. 可以选择 CPU 机架内除 1 槽以外的任何槽位 .
4. 选择弹出菜单中的 Add 模块.



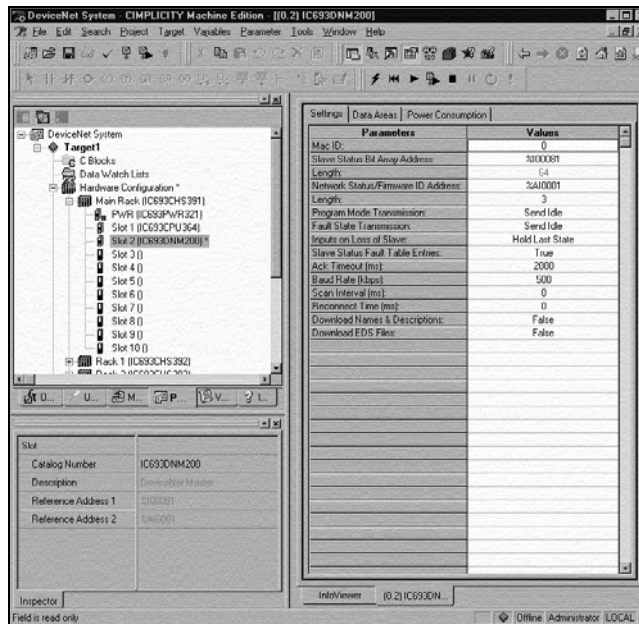
显示出模块 Catalog 对话框.

5. 添加 DeviceNet 主站模块, 在总线控制器页上单击鼠标左键. 显示出总线控制器模块列表. 选择列表中的 IC693DNM200 DeviceNet 主站点击 OK.



配置 DeviceNet 主站模块的参数

在 Navigator 窗口的 PLC 配置中添加了 DeviceNet 主站模块之后, 模块的参数编辑器窗口会出现在 InfoViewer 窗口中。



DeviceNet 主站模块的参数

设置页	
Mac ID	DeviceNet 网主站的 Mac ID(媒介访问控制标识). 有效范围: 0 - 63. 默认: 0.
Slave Status Bit Array Address	存放表示 DeviceNet 网上每个节点状态的位串的起始地址. 必须是未被占用的 %AI, %I, %Q, %G, %AQ, %R, %T, 或 %M 地址范围. 它默认使用下一个未被定义的 %I 地址. 一个从站状态地址等于起始地址+ 从站的站地址. 例如, 如果状态位串被映射到 %I00001, 站地址是 5 的从站的状态位 就是 %I00001 + 5 = %I00006. 主站的状态位的计算方法和从站一样(起始地址+ 站地址). 主站站地址默认配置是 0, 但是可以设置成任何一个有效地址(0-64).
Length (of 从站 status bits)	(只读.) 长度是 64, 对应 64 个网络设备.
Network Status/Firmware ID Address	存放模块/网络状态信息的三个字的起始地址. 默认使用下一个未被定义过的 %AI 地址. 网络状态/固件 ID 地址必须是在未被占用的 %AI, %I, %Q, %G, %AQ, %R, %T, 或 %M 的地址范围. 在系统工作时, 模块和网络状态数据以及模块固件 ID 将会存放它指定的地址上.
Length (of network status /firmware ID)	(只读) 存放网络状态/ 固件 ID 的地址的长度, 3 个字.
Program Mode Transmission	当 PLC 处于编程状态时(停止模式), DeviceNet 主站模块既可以发送空闲数据包也可以发送空数据. 默认为发送空闲数据包.

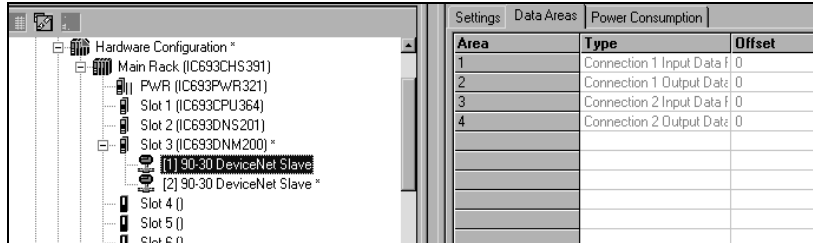
设置页	
Fault State Transmission	当 DeviceNet 主站模块 发现一个 PLC 故障 (由于 PLC 不能正常的从主站模块更新 I/O 数据), 既可以发送空闲数据包也可以发送空数据. 默认为发送空闲数据包..
inputs on Loss of slave	如果 DeviceNet 主站无法与某个从站通讯时, 它可以维持 CPU 最后一次读入时的输入值(默认 the default), 或清空输入数据.
slave Status Fault 页面 le Entries	当从站通讯状态发生时(丢失和重连), DeviceNet 主站模块 既可以报告错误给故障表也可以不报告. 如果设置成 True (默认), 主站会产生故障表项. 如果设置成 False, 从站状态不会报告给故障表.
Ack Timeout (ms)	等待冲突 MacID 检测 (在启动过程中的) CAN 确认的时间直, 以毫秒为单位。冲突 MacID 检测使用报告收到(确认)来发现冲突. 有效值范围: 0 至 65,535. 默认 2,000 毫秒 (2 秒).
Baud Rate (kbps)	DeviceNet 主站模块的数据传送速率. 可用波特率的最大值由总线的长度和电缆的类型决定. 更详细的信息参见第 2 章. 可选: 125K, 250K, 或 500K.
Scan Interval (ms) for 选通 连接	扫描相邻两个被选通从站连接的时间间隔. 默认值为 0. 如果有从站连接被设置成选通模式, 则必须指定一个时间值. 有效范围 0 至 65,535ms. 所有被选通的连接都以这个时间间隔被扫描.
Reconnect Time (ms)	如果从站连续三次扫描都没作成回应, 从站被标记成不存在并且主站会重试与之连接. 这个参数规定了主站需要等待多久才尝试重新连接. 默认时间是 0.有效范围 100 至 65535 毫秒.
Download Names & Descriptions	<p>这个设置决定是否在 PLC 配置下装时也把名称和描述也下装下去.</p> <p>默认, 这个参数是 False , 不把名称和描述下装到 PLC. 这是推荐设置, 因为下装的名称和描述会占用 PLC 大量的内存. 名称和描述仅是为了方便才使用. 下装时忽略它们不会影响系统工作.不过, 如果这个参数设置成 False, 在以后从 PLC 上传配置给编程器时, 只会包含默认的名称和说明.</p> <p>如果这个参数设置成 True, 为从站和主站输入的名称和描述会被下装到 PLC 并且以后从 PLC 上传到编程器时这些名称和描述还会存在于配置信息中</p>
Download EDS Files	<p>这个配置决定是否在下装配置信息时把配置信息所用的 EDS 文件也一同下装到 PLC.</p> <p>这个参数默认是 False 即 EDS 文件不会被下装到 PLC. 这是推荐的选择, 因为下装 EDS 文件可能会占用 PLC 过多的内存. 不过, 当设置成 False, 以后从 PLC 上传到编程器时, 配置中就不包含 EDS 文件了.你可能需要其他的 EDS 文件(比如, 磁盘上的) 来配置给定类型的更多模块. 如果没有 EDS 文件, 唯一的办法是添加更多的普通类型的模块.</p> <p>如果这个参数设置成 True, 当配置下载到 PLC 时, 用到的 EDS 文件会一同下装下去. 如果以后从 PLC 上传配置到编程器, EDS 文件会被恢复到配置的 Toolchest 中.</p>

数据区页面

这页显示了当 DeviceNet 主站模块被用作从站时，分配给网络设定的 PLC 应用程序的地址:

这页只有在 DeviceNet 主站模块 作为另一个 DeviceNet 主站设备的从站工作时才有效.

注意: 如果 DeviceNet 主站模块 没有用作从站请不要在这个页面输入. 如果 DeviceNet 主站没有用作从站, 而在页面输入了值会导致 DeviceNet 主站不能与从站通讯.



功率消耗页面

功率消耗

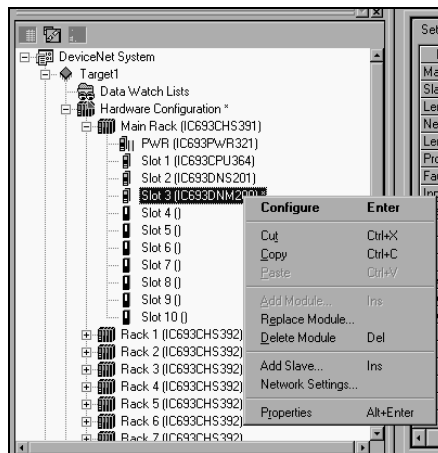
这个只读页显示了 DeviceNet 从站模块消耗背板电源的功率值. 这个功率值为模块工作时的消耗值.

Parameters	Values
+5VDC (Watts)	2.25
+24VDC Relay Power	0
+24VDC Isolated (Watts)	0

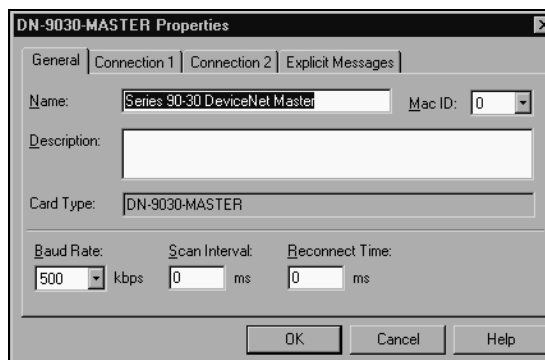
DeviceNet 从站模块还需要为它的 DeviceNet 收发器从 DeviceNet 网上的 24 伏直流电源上引入电源.

DeviceNet 主站模块的网络设置

配置 DeviceNet 主站模块的网络设置, 在配置窗口右击 DeviceNet 主站,然后选择 网络设置:



出现网络设置对话框.



在 General 页可以为模块取一个名称及填写描述信息. 还可选择 MACID 及波特率. 这些参数也可在配置参数屏幕中找到; 在这两个地方都可以修改.

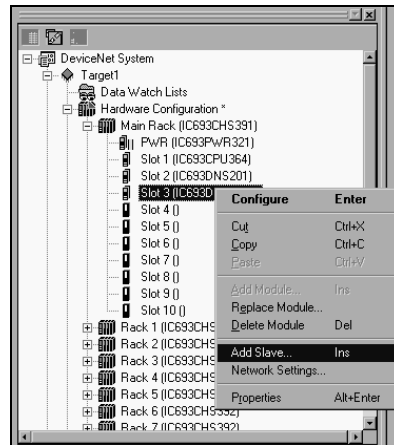
其余的页设定模块使用的 messaging 连接..

其余的页面只在系列 90-30 DeviceNet 主站模块 作为其他主机的服务器 (从站)工作时才有效. 如果 DeviceNet 主站不是其他 DeviceNet 主站的从站, 这三个页面应该保持空白.关于网络设置的详细信息, 请查询本章后面的“配置网络设置”一节.

通过向网络添加从站设置 DeviceNet 主站所访问的从站信息

在 PLC 机架内添加 DeviceNet 主站模块之后,应该告诉主站 网上有哪些需要通讯的从站设备, 这要通过向主站添加从站来完成. 在网络配置中添加从站有两个办法:

- 在 Navigator 窗口, 右击 IC693DNM200 主站 模块并且选择添加从站.



从列表中选择从站类型. 选择好从站之后点击 OK. 例如, 添加一个系列 90-30 DeviceNet 从站 模块 (IC693DNS201)作为网络的从站,你应该选择:



- 还可以直接从 Toolchest 拖放到 DeviceNet 主站. 在 Tools 工具栏上点击 Toolchest 按钮打开 Toolchest. 选择 DeviceNet Devices drawer. 选择一个从站设备.

如果你正在编辑一个从 PLC 上传的配置,设备 EDS 的(或者设备名称和描述)取决于 DeviceNet 主站模块 的配置, 这在本章的前面有过介绍.如果 EDS 文件和名称以及描述, 没有下装过,它们也不会上传的配置中, 并且 EDS 文件也不会出现在 Toolchest 中.

无论在 Navigator 窗口添加设备还是使用拖放的方法添加设备,如果没有找到你想添加的设备类型也没有这个设备的 EDS 文件,你只能将从站配置成一般设备.

添加 Device s ED 文件

如果你想配置的设备在列表和 Toolchest 中都没有, 可以使用 EDS 文件, 这个文件由设备制造商提供, 点击 Have Disk. 在 Open 对话框, 找到 EDS 文件并点击 Open. (当使用这种方法选择一个 EDS 文件, 它会添加到从站目录和 Toolchest 的 DeviceNet Devices drawer 中.)

配置添加到主站的从站的网络设置

网络设置 包括 MAC ID、波特率、和设备间的 messaging 连接.

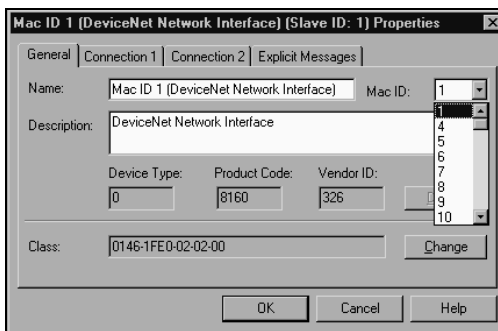
分配 MAC ID 和波特率

确保在 Machine Edition 配置中的每个从站输入的 MAC ID 与这些从站自身设置的 MAC ID 一致.

许多设备使用 DIP 开关设定 MAC ID.有些需要主站发送的配置命令设置 MAC ID.所有软件配置的设备开始的默认 MAC ID 都是: 63. 如果配置一个软件配置的设备,添加一个 MAC ID 是 63 的一般设备然后使能 Explicit Messaging 以便配置消息可以被发送到该设备.因此,给网络上的设备分配的 MAC ID 是 63 时要特别小心, 防止添加一个新从站时出现重复 MAC ID 冲突的故障.因为所有的软件配置的从站开始的默认 MAC ID 都一样, 这些从站应该一次向网上添加一个. 当每个新从站连上时,它的 MAC ID 应该使用发送服务 Explicit COMMREQ 来设置,并且它的工作状态应该在配置下一个从站前检查一下.记住当设备在收到配置消息后将使用新的 MAC ID.

配置从站设备的 MAC ID

网络设置: General 页,配置设备的 MAC ID. MAC ID 的默认值时下一个可用地址而不是跳过的地址.下拉列表显示了 Machine Edition 配置中还没有用过的 MAC ID. 本例中, ID 2 和 3 已经指派过了, 所以这些数字不出现在列表中.



名称

用于给从站命名一个标识名称

描述

用于添加从站的描述信息

配置添加到主站的从站的 I/O Messaging 连接

I/O Messaging 是 DeviceNet 系统中的一个术语，用于在主站和从站之间路由和自动交换数据。每个配置好的 I/O 消息定义了一个生产设备与一至多个消费设备之间的确定通讯路径。一旦连接建立起来，I/O Messaging 通讯就在系统工作时自动进行。

DeviceNet 系统中的每个系列 90-30 设备最多可设置两个不同的 I/O Messaging 连接。每个连接都可被禁止(默认)，也可设置为轮询、选通、状态改变，或循环工作。连接要根据应用的需要进行配置。例如，主站可以选通所有只有输入数据的从站色设备而轮询其余的从站设备。

选择了一个 I/O Messaging 连接类型也就决定了另外一个连接可选择的类型范围了，如下表所示。例如，你仅可为模块选择一个轮询连接。

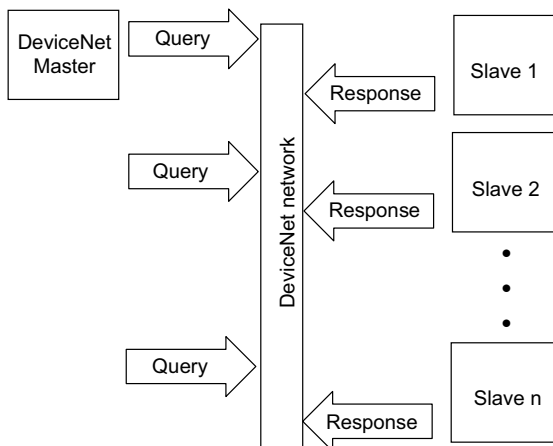
<i>选择类型的一个连接</i>	<i>另一连接可用的类型</i>
禁止	禁止, 轮询, 选通, 状态改变, 循环
轮询	禁止, 选通, 状态改变, 循环
选通	禁止, 轮询, 状态改变, 循环
状态改变	禁止, 轮询, 选通
循环	禁止, 轮询, 选通

每种连接类型的配置在下面几页中进行介绍。

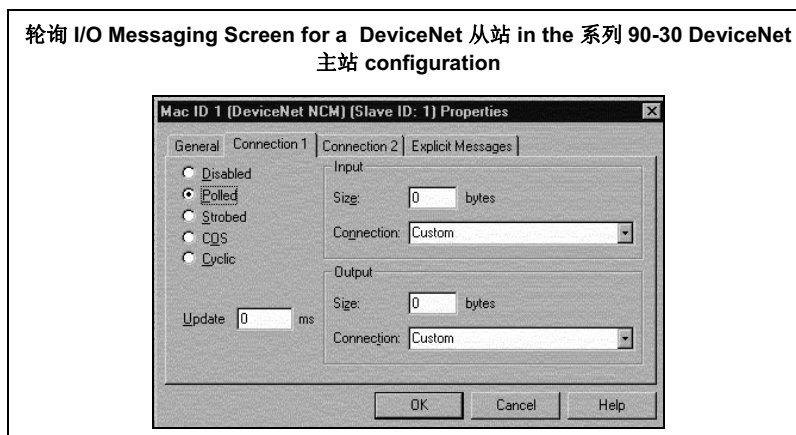
配置 轮询 I/O Messaging 连接

在轮询 I/O 模式下(执行 I/O 最常见的方法), DeviceNet 主站自动地发送一个消息, 消息中包含的被配置成轮询连接模式的所有从站的输出数据. 从站则发回一个响应, 响应中包含了从站的输入数据.因此轮询需要 2 个消息才能更新 每个 被轮询设备的 I/O 数据.

主站 尽可能快地发送选举序列.选举是最有精确但效率最低的更新 I/O 数据模式. 这种模式非程适合于应用程序中可靠性很高的控制数据.



将从站的连接 1 或者连接 2 配置成轮询模块, 需要选择从站属性菜单的 Polled 一项.



对于输入资源, 要指定从站模块发给主站的数据字节数.

对于输出资源, 要指定 DeviceNet 从站模块从主站接收的字节数.

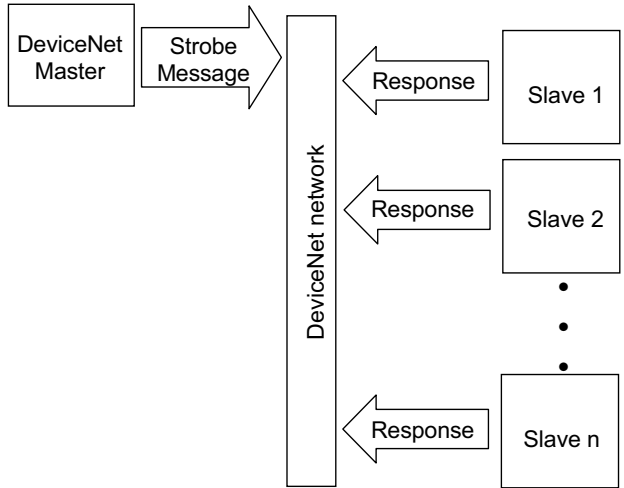
对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0.

注意: 这里的输入输出的字节数必须和从站及主站的设置一致, 否则无法合从站通讯.

配置选通 I/O Messaging 连接

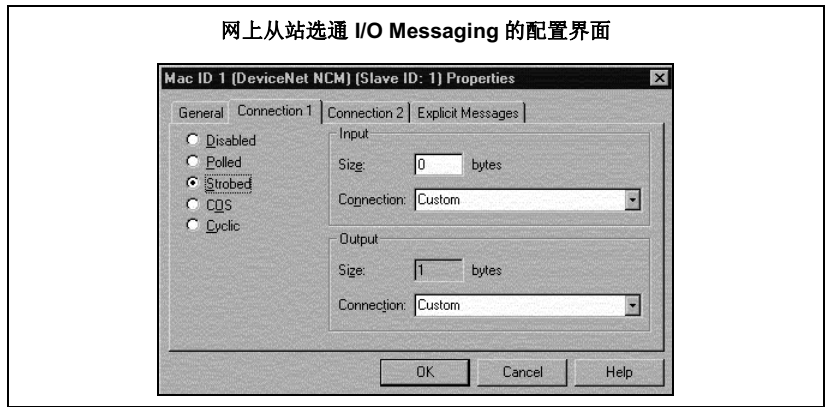
在选通 I/O 模式时, 主站产生一个单个选通请求消息, 这个请求消息被所有配置成选通连接模式的设备所消费, 请求获得从站的当前状态. 由 DeviceNet 主站模块的扫描间隔参数指定的速率执行.

每个选通的设备将它们的输入数据发回给主站. 设备按照它们的 MAC ID 顺序回应, 从最低的 MAC ID 设备开始. MAC ID 可以按照 I/O 的优先级分配给从站.



选通 I/O Messaging 比 轮询 I/O messaging 更有效, 因为主站 不需要为每一个从站发送一个独立的选通请求. 选通 I/O Messaging 对于象传感器之类的只有输入数据的从站更为有效.

将连接配置成选通 I/O Messaging, 在从站属性菜单选择选通.

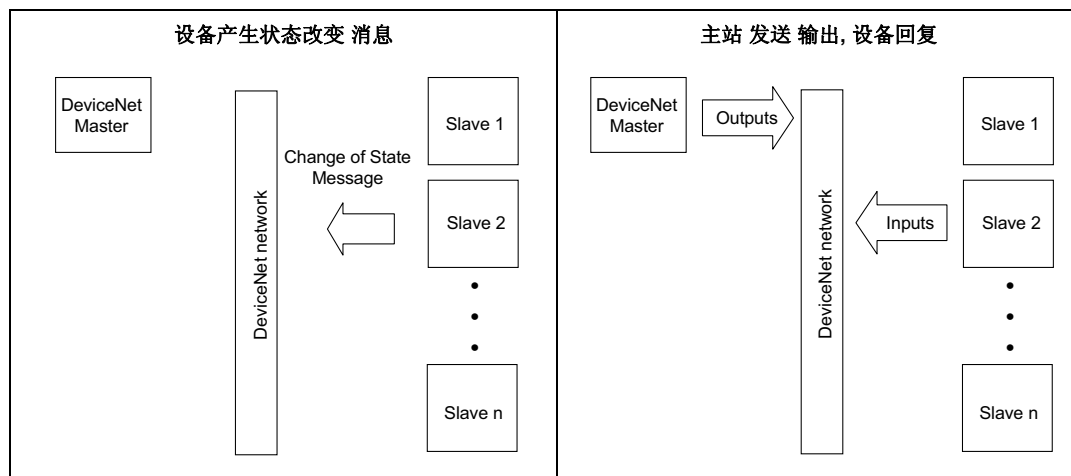


对于输入资源, 指定发送给主站的数据的字节数.

输出资源的长度自动设置成 1 个字节. 主站发给从站的消息即它们要把它们的输入数据发送给主站, 这个消息是 1 个字节的消息. 它反映了选通请求消息中 I/O 位的状态: 置位 (1) 或复位 (0).

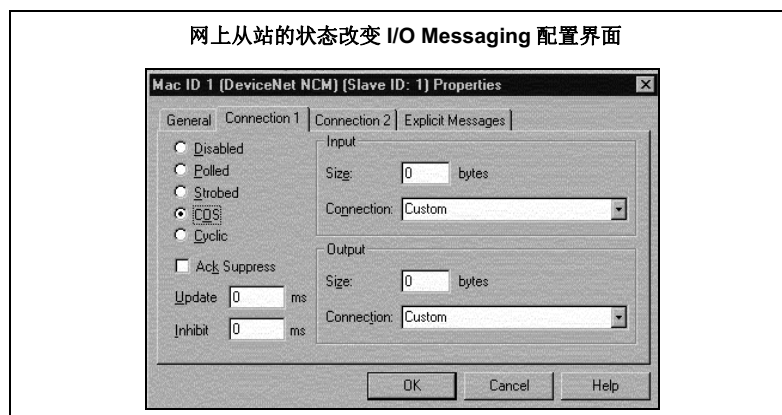
配置状态改变 cos I/O Messaging 连接

配置成状态改变 (cos) I/O Messaging 的连接只有在模块发送一个报告状态变化的消息给主站时才会被激活. 主站随后发送一个输出 消息给这个设备并且设备返回一个包含输入数据的回复信息.



状态改变 I/O Messaging 是网上效率最高的 messaging 类型, 但是比其他方法可能精度要差.

将连接配置成状态改变 I/O Messaging, 需选择连接页菜单上的状态改变.



对于输入资源,需指定模块发送给主站的字节数.

对于输出资源, 需指定模块从主站接收的字节数.

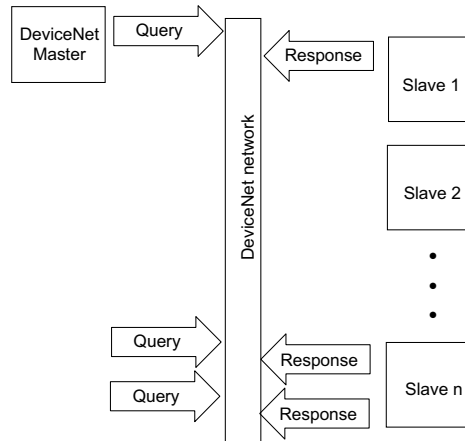
如果禁止确认被选中,主站不会等待模块发送的确认消息.

如果使用抑制选项, 设定一个两次数据生产之间的最小延时时间, 以毫秒为单位. 例如, 一个从站 由一个状态改变连接, 每隔 5 毫秒变化一次数据,但是控制程序每隔 25 毫秒更新一次数据. 设定抑制值为 25, 就能让从站每个 25 毫秒发送一次数据从而避免浪费网络带宽.

对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0.

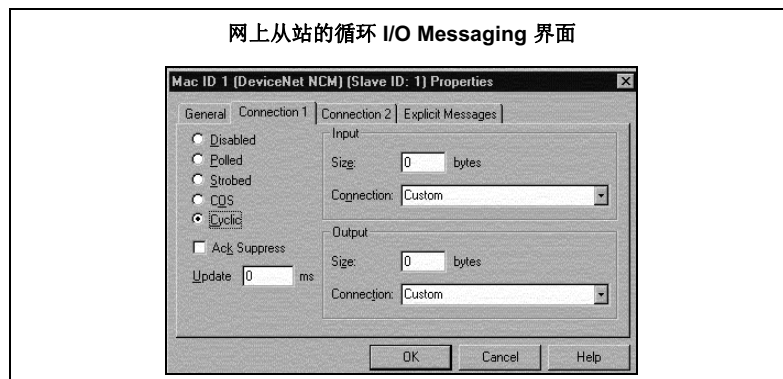
配置循环 I/O Messaging 连接

循环 I/O Messaging 模式和轮询 I/O Messaging 模式类似, DeviceNet 主站自动的发送一个消息, 消息包含了发给被配置成循环更新连接模式的设备的输出数据. 模块发回它们的输入数据. 和轮询一样, 循环 I/O Messaging 需要 2 个消息才能完成更新设备的 I/O 数据. 与轮询 messaging 的区别是, 循环 messaging 可以为每个从站配置不同的时间间隔



循环 messaging 适于那些如模拟量输入传感器之类的设别. 例如, 把温度传感器设置成循环消息, 以每隔 500 毫秒的频率报告它的测量值. 循环 messaging 可以降低网络的流量而又可以精确测量输入仪表的某种类型值. 在应用程序中使用它效率会比较高. 循环 I/O 连接可以作为一个'心跳'来保证设备能够继续工作, 这个从站设备同时还使用状态更新 I/O 连接来更新它的 I/O 状态

将连接配置成循环 I/O Messaging, 需选择从站属性菜单的 Cyclic..



对于输入资源, 需指定模块发送给主站的字节数.

对于输出资源, 需指定模块从主站接收的字节数.

如果禁止确认被选中, 主站不会等待模块发送的确认消息.

对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0.

配置 DeviceNet Explicit Messaging

Explicit Messaging 是优先级最高的 messaging. Explicit messaging 除能够访问默认的 I/O 连接集,和为 explicit 连接创建缓冲区还能够访问对象. 不是所有的从站都支持 explicit 服务, 包括开关量 I/O 设备在内的许多设备都不支持..

一些设备会根据配置的参数返回 Explicit Messaging. 例如, 一些 VersaPoint 模块可被配置成使用 Explicit Messaging. 要配置成 Explicit Messaging, 首先按照下列的步骤配置 Explicit Messaging 连接.

如果 DeviceNet 从站模块要使用 Explicit Messaging, 点击 Explicit 消息 页上的 Enable Explicit 连接.

另外, 对于系列 90-30 DeviceNet 从站模块和作为从站工作的系列 90-30 DeviceNet 主站模块, 需要指定消息请求和消息回复的大小. 确保指定的大小足够大. 系列 90-30 DeviceNet 模块通过应用程序中的 COMMREQ 指令实现 Explicit Messaging. 第 6 章中介绍了这些 COMMREQ 消息, "通讯编程". 每个 explicit 消息类型的消息大小在该章有详细说明.



配置作为从站工作的 DeviceNet 主站的网络设置

网络设置包括 MAC ID、波特率、及网上设备间的 messaging. 这个概念已在前面一节"配置添加到主站的从站的网络设置"中介绍过了.

为作为从站工作的 DeviceNet 主站配置 I/O Messaging 连接

I/O Messaging 是 DeviceNet 系统中的一个术语, 用于在主站和从站之间路由和自动交换数据. 每个配置好的 I/O 消息定义了一个生产设备与一至多个消费设备之间的确定通讯路径. 一旦连接建立起来, I/O Messaging 通讯就在系统工作时自动进行.

DeviceNet 系统中的每个系列 90-30 设备可以配置最多个不同类的 I/O Messaging 连接. 每个连接都可被禁止(默认), 也可设置为轮询, 选通, 状态改变, 或循环工作. 连接要根据应用的需要进行配置. 例如, 主站可以选通所有只有输入数据的从站色设备而轮询其余的从站设备.

选择了一个 I/O Messaging 连接类型也就决定了另外一个连接可选择的类型范围了, 如下表所示. 例如, 你仅可为模块选择一个轮询连接.

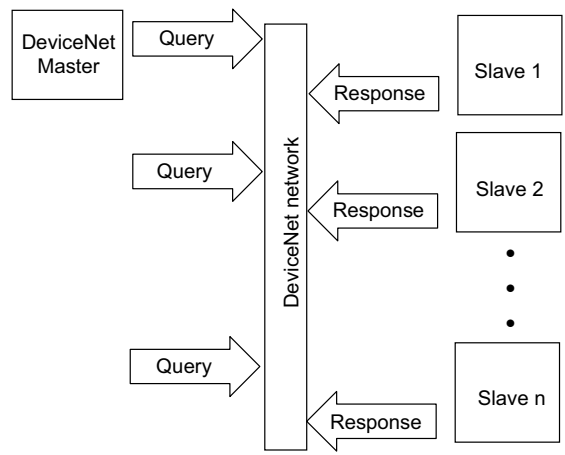
<i>选择类型的一个连接</i>	<i>另一连接可用的类型</i>
禁止	禁止, 轮询, 选通, 状态改变, 循环
轮询	禁止, 选通, 状态改变, 循环
选通	禁止, 轮询, 状态改变, 循环
状态改变	禁止, 轮询, 选通
循环	禁止, 轮询, 选通

每种连接类型的配置在下面几页中进行介绍.

配置轮询 I/O Messaging 连接

在轮询 I/O 模式下(访问 I/O 最常用的模式), DeviceNet 主站自动地发送一个消息, 消息中包含的被配置成轮询连接模式的所有从站的输出数据. 从站则发回一个响应, 响应中包含了从站的输入数据.因此轮询需要 2 个消息才能更新 每个 被轮询设备的 I/O 数据.

主站 尽可能快地发送选举序列.选举是最有精确但效率最低的更新 I/O 数据模式. 这种模式非程适合于应用程序中可靠性很高的控制数据.



将从站的连接 1 或者连接 2 配置成轮询模块, 需要选择从站属性菜单的 Polled 一项..



对于输入资源, 要指定从站模块发给主站的数据字节数.

对于输出资源, 要指定 DeviceNet 从站模块从主站接收的字节数.

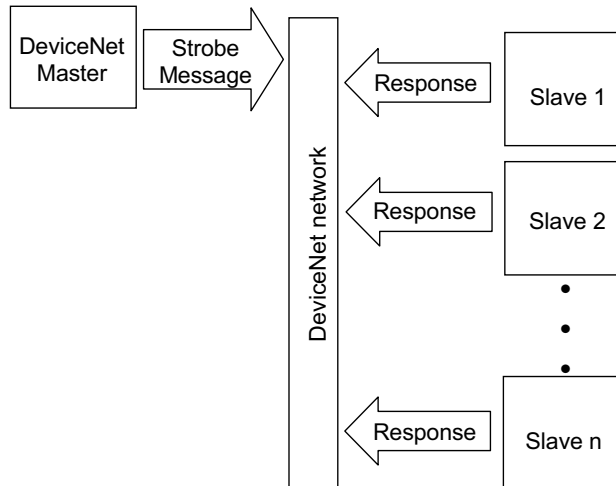
对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0.

注意: 这里的输入输出的字节数必须和从站及主站的设置一致, 否则无法合从站通讯.

配置选通 I/O Messaging 连接

在选通 I/O 模式时, 主站产生一个单个选通请求消息, 这个请求消息被所有配置成选通连接模式的设备所消费, 请求获得从站的当前状态. 由 DeviceNet 主站模块的扫描间隔参数指定的速率执行.

每个选通的设备将它们的输入数据发回给主站. 设备按照它们的 MAC ID 顺序回应, 从最低的 MAC ID 设备开始. MAC ID 可以按照 I/O 的优先级分配给从站.



选通 I/O Messaging 比 轮询 I/O messaging 更有效, 因为主站 不需要为每一个从站发送一个独立的选通请求. 选通 I/O Messaging 对于象传感器之类的只有输入数据的从站更为有效.

将连接配置成选通 I/O Messaging, 在从站属性菜单选择选通.

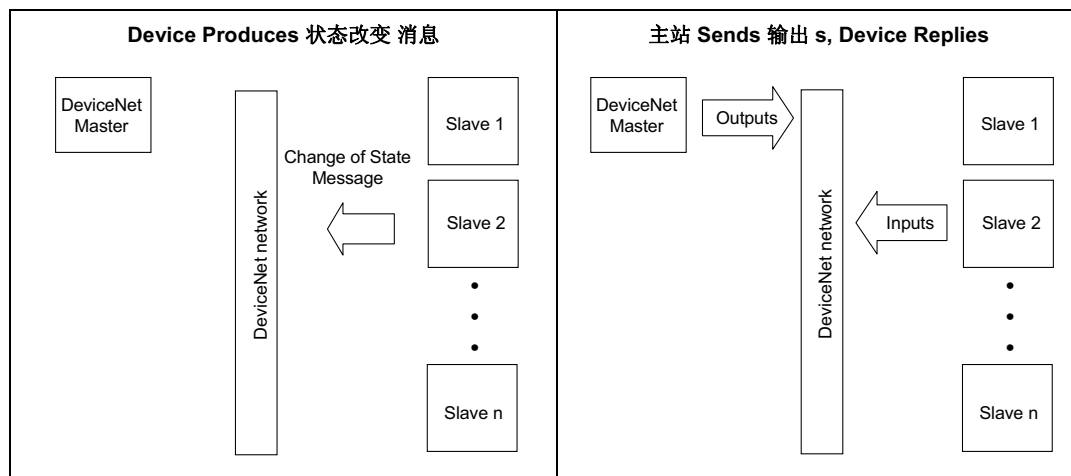


对于输入资源, 指定发送给主站的数据的字节数.

输出资源的长度自动设置成 1 个字节. 主站发给从站的消息即它们要把它们的输入数据发送给主站, 这个消息是 1 个字节的消息. 它反映了选通请求消息中 I/O 位的状态: 置位 (1) 或复位 (0).

配置状态改变 cos I/O Messaging 连接

配置成状态改变 (cos) I/O Messaging 的连接只有在模块发送一个报告状态变化的消息给主站时才会被激活. 主站随后发送一个输出 消息给这个设备并且设备返回一个包含输入数据的回复信息.



状态改变 I/O Messaging 是网上效率最高的 messaging 类型, 但是比其他方法可能精度要差.

将连接配置成状态改变 I/O Messaging, 需选择连接页菜单上的状态改变.



对于输入资源,需指定模块发送给主站的字节数.

对于输出资源, 需指定模块从主站接收的字节数.

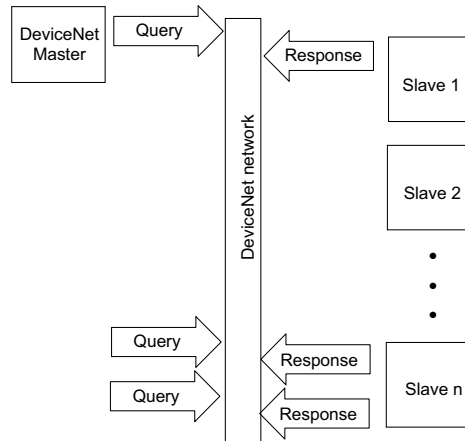
如果禁止确认被选中,主站不会等待模块发送的确认消息.

如果使用抑制选项, 设定一个两次数据生产之间的最小延时时间, 以毫秒为单位. 例如, 一个从站 由一个状态改变连接, 每隔 5 毫秒变化一次数据,但是控制程序每隔 25 毫秒更新一次数据. 设定抑制值为 25, 就能让从站每个 25 毫秒发送一次数据从而避免浪费网络带宽.

对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0.

配置循环 I/O Messaging 连接

循环 I/O Messaging 模式和轮询 I/O Messaging 模式类似, DeviceNet 主站自动的发送一个消息, 消息包含了发给被配置成循环更新连接模式的设备的输出数据. 模块发回它们的输入数据. 和轮询一样, 循环 I/O Messaging 需要 2 个消息才能完成更新设备的 I/O 数据. 与轮询 messaging 的区别是, 循环 messaging 可以为每个从站配置不同的时间间隔



循环 messaging 适于那些如模拟量输入传感器之类的设别. 例如, 把温度传感器设置成循环消息, 以每隔 500 毫秒的频率报告它的测量值. 循环 messaging 可以降低网络的流量而又可以精确测量输入仪表的某种类型值. 在应用程序中使用它效率会比较高. 循环 I/O 连接可以作为一个'心跳'来保证设备能够继续工作, 这个从站设备同时还使用状态更新 I/O 连接来更新它的 I/O 状态

将连接配置成循环 I/O Messaging, 需选择从站属性菜单的 Cyclic.



对于输入资源, 需指定模块发送给主站的字节数.

对于输出资源, 需指定模块从主站接收的字节数.

如果禁止确认被选中, 主站不会等待模块发送的确认消息.

对于更新, 需要指定一个毫秒值, 以指明 DeviceNet 主站更新这个设备的数据的间隔. 如果希望尽可能快的更新输入 0

配置 DeviceNet Explicit Messaging

Explicit Messaging 是优先级别最高的 messaging. Explicit messaging 除能够访问默认的 I/O 连接集,和为 explicit 连接创建缓冲区还能够访问对象. 不是所有的从站都支持 explicit 服务, 包括开关量 I/O 设备在内的许多设备都不支持.

一些设备会根据配置的参数返回 Explicit Messaging. 例如, 一些 VersaPoint 模块可被配置成使用 Explicit Messaging. 要配置成 Explicit Messaging, 首先按照下列的步骤配置 Explicit Messaging 连接.

如果 DeviceNet 从站模块要使用 Explicit Messaging, 点击 Explicit 消息 页上的 Enable Explicit 连接.

另外, 对于系列 90-30 DeviceNet 从站模块和作为从站工作的系列 90-30 DeviceNet 主站模块, 需要指定消息请求和消息回复的大小. 确保指定的大小足够大. 系列 90-30 DeviceNet 模块通过应用程序中的 COMMREQ 指令实现 Explicit Messaging. 第 6 章中介绍了这些 COMMREQ 消息, "通讯编程". 每个 explicit 消息类型的消息大小在该章有详细说明.



章 4

PLC 配置 DeviceNet 从站

本章将解释如何把 90-30 系列 DeviceNet 从站模块(IC693DNS201) 添加到 90-30 系列 PLC 的配置中. 还将解释如何配置 90-30 系列 系列 DeviceNet 从站模块和 DeviceNet 网之间的通讯连接.

- 配置步骤
- 添加一个 DeviceNet 从站模块到 PLC 配置
- 配置 DeviceNet 从站模块的参数
- 配置 DeviceNet 从站模块的网络设置项
 - 配置 MAC ID
 - 配置 I/O Messaging 连接
 - 配置 DeviceNet Explicit Messaging

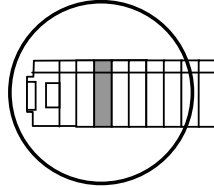
这些配置过程适用于了解 CIMPLICITY Machine Edition Logic Developer 软件和 90-30 系列 PLC 的用户. 如何使用编程软件,请查阅软件的内置帮助系统.

注意: 只有 CIMPLICITY Machine Edition Logic Developer 软件支持 DeviceNet 从站模块. Logimaster™, VersaPro™, 和 控制软件都不支持该模块.

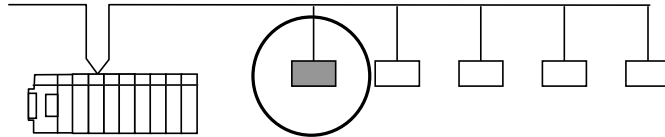
配置步骤

配置 DeviceNet 从站模块需三个基本步骤：

- 配置安装模块的 PLC CPU 的模块参数. 下页详细介绍.



- 配置模块的网络设置项. 参见本章的"配置网络设置项"一节.

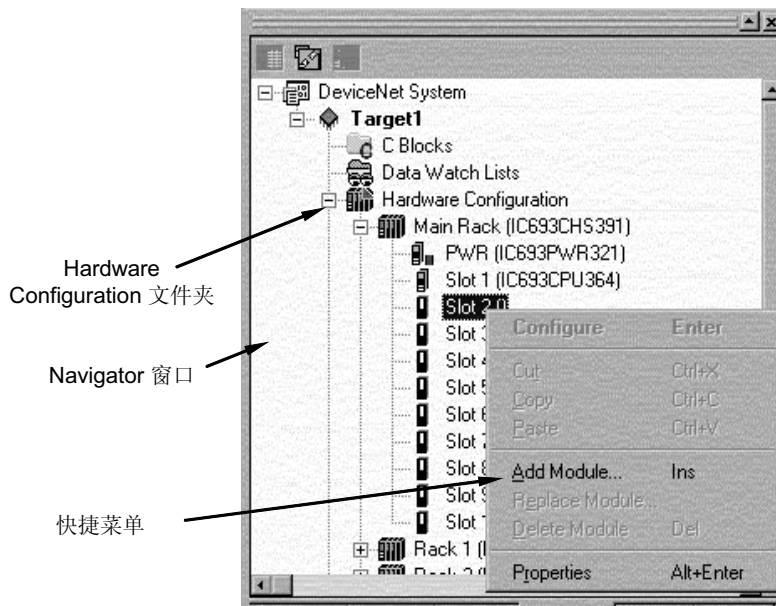


- 配置模块使之成为网络主站所属的一个从站. 在第 3 章, 介绍配置 90-30 系列 DeviceNet 主站模块的网络参数. 本手册不介绍其他类型的主站配置过程.

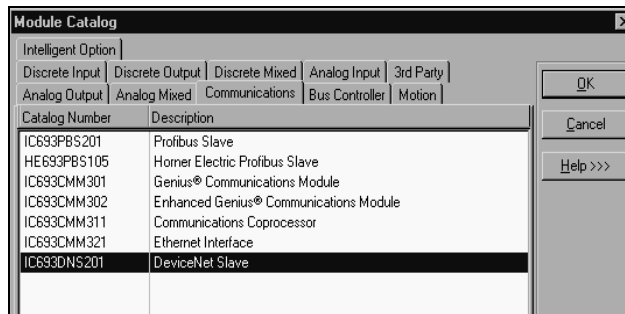
添加一个 DeviceNet 从站模块到 PLC 的配置中

首先, 在 PLC 机架配置中添加 90-30 系列 DeviceNet 从站模块 (IC693DNS201) . IC693DNS201 兼容除IC693CPU321和IC693CPU340以外的 90-30 系列 CPU . 它要求 CPU 固件版本至少是 8.0. 如果使用的 CPU 有 10.6 或更高的版本的固件, 则推荐使用.

1. 在该配置的 Navigator 窗口内的 Project tab 中 , 展开 Hardware Configuration 文件夹.
2. 在 Hardware Configuration 文件夹内, 选择安装 DeviceNet 从站模块的 PLC 槽, 右击鼠标. 模块可以安装到 CPU 机架内除槽 1 以外的其他槽上.
3. 在快捷菜单中选则 Add Module.

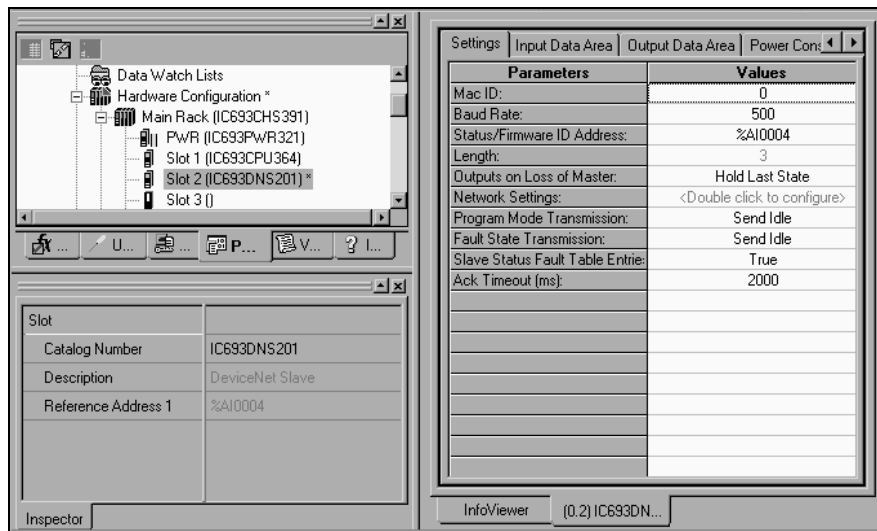


4. Module Catalog 对话框显示. 因为要添加 DeviceNet 从站模块, 所以点击 Communications tab. 在列表中选择 IC693DNS201 DeviceNet Slave 然后点击 OK.



配置 DeviceNet 从站模块的参数

在选择模块之后, 其参数编辑器窗口会显示在 InfoViewer 窗口区域.

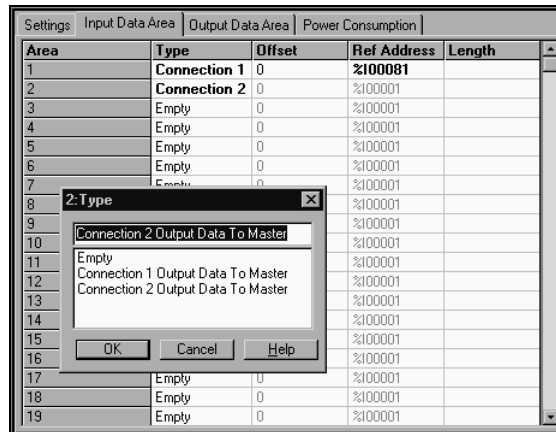


DeviceNet 从站模块的参数

设置页	
Mac ID	DeviceNet 网上从站的 Mac ID (介质访问控制标识) . 有效范围: 0 - 63. 默认: 0. 如需设置位于 DeviceNet 从站属性对话框中这个参数, 找到 Navigator 窗口, 右击 DeviceNet 从站并选择网络设置. DeviceNet 从站属性对话框弹出.
Baud Rate (kbps)	DeviceNet 从站模块的通讯速率. 可用的最大波特率取决于总线的长度和电缆的类型. 详情参见第 2 章. 可选择: 125K, 250K, 或 500K. 默认为 500.
Status/ Firmware ID Address	<p>占用三个字大小的模块状态信息的起始地址. 默认为 %AI 类型内存. 网络状态/固件 ID 的地址范围不能重叠, 可选的内存类型: %AI, %I, %Q, %G, %AQ, %R, %T, 或 %M. 默认偏移量为选用内存类型的下一个可用地址.</p> <p>固件 ID 字为当前 DeviceNet 主站模块中的固件版本. 主修订版本位于高字节而次修订版本号位于低字节.</p>
Length (of module status/firmware ID)	这是上面提到的网络状态/固件 ID 地址内存的长度. 长度是 3 个字或 48 个位(只读).
Outputs on Loss of Master	这个参数决定从站与主站通讯中断时从站如何处理输入/输出. 默认为保持最后状态. 可以被设置成清零.
Network Settings	双击鼠标编辑从站的网络通讯设置. 参见本章的"配置网络设置".
Program Mode Transmission	当 PLC 处于编程模式(即停止模式), the 模块既可发送空闲数据包也可不发送数据. 默认为发送空闲数据包.
Fault State Transmission	当模块检验到一个 PLC 故障, 模块既可发送空闲数据包也可将数据清零. 默认为发送空闲数据包.
Slave Status Fault Table Entries	当从站通讯状态出现问题时(丢失和重连), DeviceNet 主站模块既可在故障表中报告错误也可不报告. 如果设定值是 True (默认), 主站产生故障记录. 如果设定值是 False, 从站状态出现问题是不向故障表报告.
Ack Timeout (ms)	等待重复 MacID 检测(在启动过程中执行的) CAN 确认消息的时间值, 单位为毫秒. 如果在这段时间内没有收到这个确认消息, 模块报告一个确认失败故障. 有效范围: 0 至 65,535. 默认为 2,000 毫秒(2 秒).

数据区页

这些页显示了被所选模块分配的 PLC 应用程序变量:这些变量的分配可以编辑,或使用默认值.注意输入数据区页显示的是主站的输入值,即模块的输出值.输出数据区页显示的是主站的输出值,即模块的输入值.对于输入和输出数据区,可以为连接 1 和连接 2 分别配置.



输入和输出的偏移量是主站 PLC 的内存区起始地址.必须小于在 Network Settings (网络设置) 对话框中指定的数据区大小.要指定数据区大小,在 Navigator 窗口中,右击 DeviceNet 从站然后选 Network Settings (网络设置). DeviceNet Slave Properties 对话框出现.在连接 1 或者连接 2 tab,在资源下设置大小.

变量地址是主站 PLC 内存中的地址,数据被映射到这里.如果大小被设置成 0,这个区域是只读的.地址可以是 %AI, %I, %Q, %G, %AQ, %R, %T, or %M. 如果字节数是奇数,只能使用开关量地址(%I, %Q, %G, %M, %T).

功率消耗页

功率消耗

这个只读页显示了 DeviceNet 从站模块消耗背板电源的功率值.这个功率值为模块工作时的消耗值.

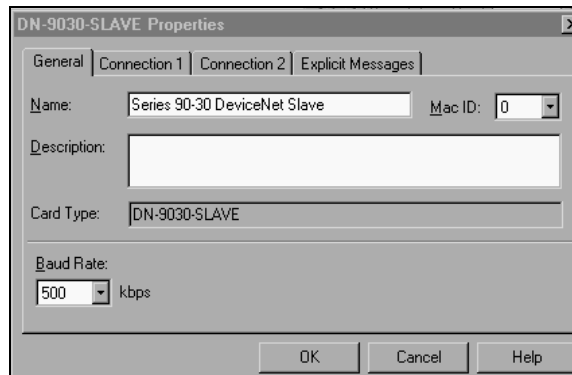
Parameters	Values
+5VDC [Watts]	2.25
+24VDC Relay Powe	0
+24VDC Isolated [W	0

DeviceNet 从站模块还需要为它的 DeviceNet 收发器从 DeviceNet 网上的 24 伏直流电源上引入电源.

配置 DeviceNet 从站模块的网络设置

配置 DeviceNet 从站模块的网络配置, 在 PLC 配置中, 右击 DeviceNet 从站, 然后选择 Network (网络设置)。

网络配置对话框出现。



在 General 页可以为模块取一个名称及填写描述信息. 还可选择 MACID 及波特率. 这些参数也可在配置参数屏幕中找到; 在这两个地方都可以修改.

其余的页设定模块使用的 messaging 连接.

配置 MAC ID

所有的可软件配置的设备开始都有同一个默认的 MAC ID: 63. 因此, 如果可能尽量避免将一个设备的 MAC ID 设置成 63,防止在添加一个新的从站时出现重复 MAC ID 冲突的问题.

配置 I/O Messaging 连接

I/O Messaging 是 DeviceNet 系统中的一个术语，用于在主站和从站之间路由和自动交换数据。每个配置好的 I/O 消息定义了一个生产设备与一至多个消费设备之间的确定通讯路径。一旦连接建立起来，I/O Messaging 通讯就在系统工作时自动进行。

DeviceNet 从站模块最多可设置两个不同的 I/O Messaging 连接。每个连接都可被禁止(默认)，也可设置为轮询、选通、状态改变，或循环工作。连接要根据应用的需要进行配置。

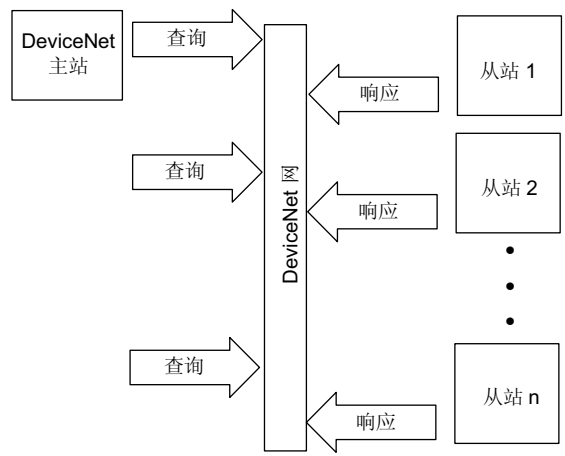
选择了一个 I/O Messaging 连接类型也就决定了另外一个连接可选择的类型范围了，如下表所示。例如，你仅可为模块选择一个轮询连接。

<i>选择类型的一个连接</i>	<i>另一连接可用的类型</i>
禁止	禁止, 轮询, 选, 状态, 循环
轮询	禁止, 选通, 状态改变, 循环
选通	禁止, 轮询, 状态改变, 循环
状态改变	禁止, 轮询, 选通
循环	禁止, 轮询, 选通

每种连接类型的配置在下面几页中进行介绍。

配置轮询 I/O Messaging 连接

在轮询 I/O 模式下, DeviceNet 主站自动地发送一个消息, 消息中包含的被配置成轮询连接模式的所有从站的输出数据. 从站则发回一个响应, 响应中包含了从站的输入数据. 因此轮询需要 2 个消息才能更新 每个 被轮询设备的 I/O 数据.



将从站的连接 1 或者连接 2 配置成轮询模块, 需要选择从站属性菜单的 Polled 一项.



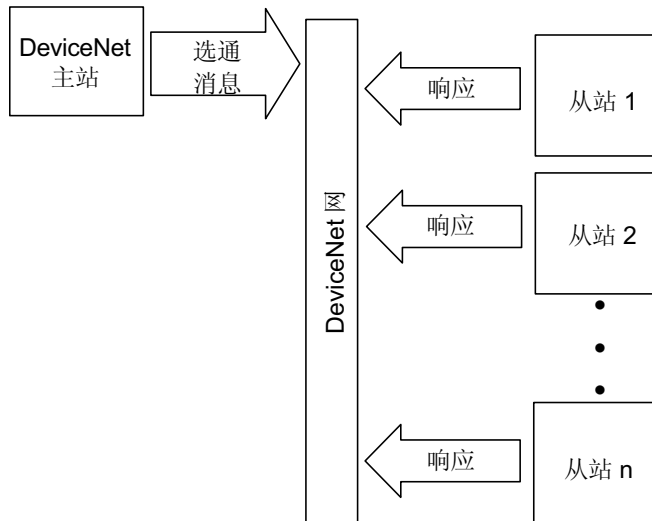
对于输入资源, 要指定从站模块发给主站的数据字节数.

对于输出资源, 要指定 DeviceNet 从站模块从主站接收的字节数.

配置选通 I/O Messaging 连接

在选通 I/O 模式时, 主站产生一个单个选通请求消息, 这个请求消息被所有配置成选通连接模式的设备所消费, 请求获得从站的当前状态。

每个选通的设备将它们的输入数据发回给主站. 设备按照它们的 MAC ID 顺序回应, 从最低的 MAC ID 设备开始. MAC ID 可以按照 I/O 的优先级分配给从站。



将连接配置成选通 I/O Messaging, 在从站属性菜单选择 Strobed.

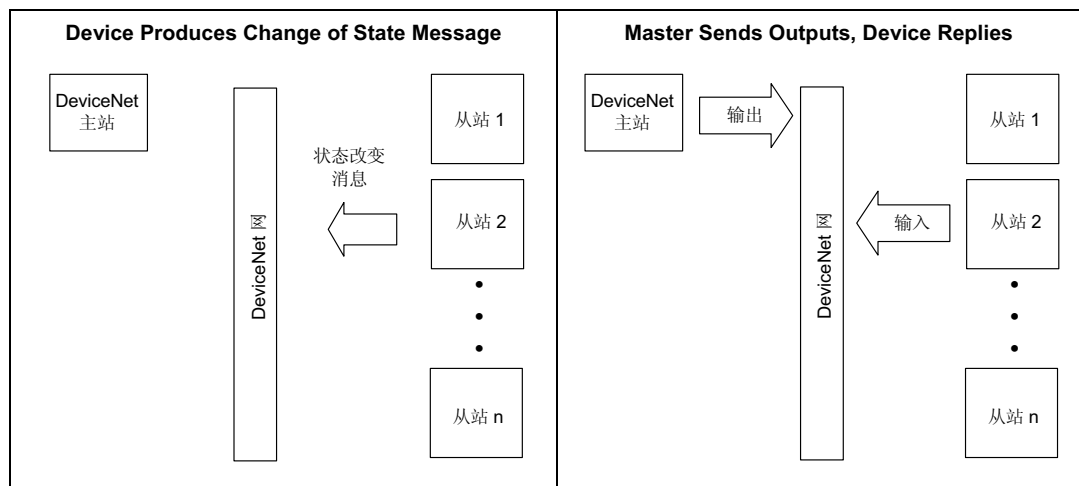


对于输入资源, 指定发送给主站的数据的字节数。

输出资源的长度自动设置成 1 个字节. 主站发给从站的消息即它们要把它们的输入数据发送给主站, 这个消息是 1 个字节的消息. 它反映了选通请求消息中 I/O 位的状态: 置位 (1) 或复位 (0).

配置状态改变 I/O Messaging 连接

配置成状态改变 (COS) I/O Messaging 的连接只有在模块发送一个报告状态变化的消息给主站时才会被激活. 状态把输出消息发送给模块然后模块会把它的输入数据发给主站.



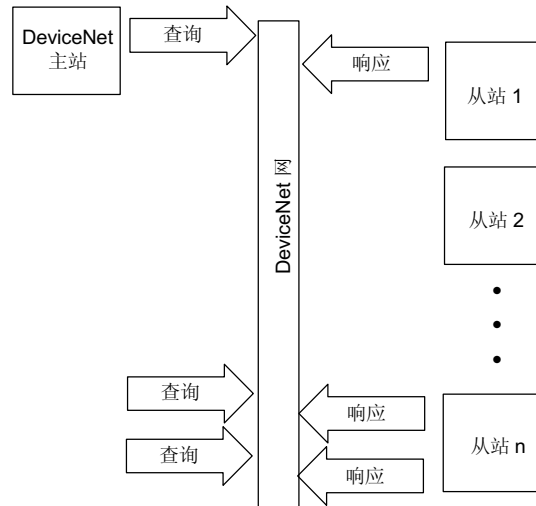
将连接配置成状态改变 I/O Messaging ,需选择连接页菜单上的 COS.



对于输入资源,需指定模块发送给主站的字节数.
对于输出资源, 需指定模块从主站接收的字节数.
如果禁止确认被选中,主站不会等待模块发送的确认消息.

配置循环 I/O Messaging 连接

循环 I/O Messaging 模式和轮询 I/O Messaging 模式类似, DeviceNet 主站自动的发送一个消息, 消息包含了发给被配置成循环更新连接模式的设备的输出数据. 模块发回它们的输入数据. 和轮询一样, 循环 I/O Messaging 需要 2 个消息才能完成更新设备的 I/O 数据. 与轮询 messaging 的区别是, 循环 messaging 可以为每个从站配置不同的时间间隔.



循环 I/O 连接可以作为一个'心跳'来保证设备能够继续工作, 这个从站设备同时还使用状态更新 I/O 连接来更新它的 I/O 状态.

将连接配置成循环 I/O Messaging, 需选择从站属性菜单的 Cyclic.



对于输入资源, 需指定模块发给主站的字节数.

对于输出资源, 需指定模块接收主站的字节数.

如果禁止确认被选中, 主站不会等待模块发送的确认消息.

配置 DeviceNet Explicit Messaging

Explicit Messaging 是优先级最高的 messaging. Explicit messaging 除了能够访问默认的 I/O 连接集还能访问对象, 并且可以为 explicit 连接新建一个缓冲区.

如果 DeviceNet 从站模块要使用 Explicit Messagin, 点击 Explicit Messages 页上的 Enable Explicit Connection.

同时还要指定消息请求和消息响应的大小. 确保给定的大小足够大. 90-30 系列 DeviceNet 模块通过程序中的 COMMREQ 指令 来实现 Explicit Messaging through. 这些 COMMREQ 消息在第 6 章中解释, "通讯编程". 每种消息类型的消息长度也在那章讲解.



章 5

模块工作

本章介绍系列 90-30DeviceNet 主站模块与系列 90-30DeviceNet 从站模块在 90-30 系列 PLC 系统中如何工作.

- 系列 90-30DeviceNet 主站模块的工作情况
 - PLC 处于停止模式时的工作情况
 - PLC 处于运行模式时的工作情况
- 系列 90-30 从站(主站) DeviceNet 模块的工作
- 系列 90-30 DeviceNet 模块的故障表条目
 - I/O 故障表条目
 - PLC 故障表条目
- 90-30 系列 DeviceNet 模块的 PLC 状态参考量
- 系列 90-30DeviceNet 主站模块的设备状态位串

系列 90-30 DeviceNet 主站模块工作

系列 90-30 DeviceNet 主站模块 在与网络设备的连接建立好之后会自动交换数据.不过,它向网络设备发送的数据内容由 PLC 的工作模式决定.

处于停止模式时 DeviceNet 主站模块的工作方式

当 PLC 处于停止状态, DeviceNet 主站模块 将网络设备发送过来的输入数据更新到内部内存,并且接收来自 PLC CPU 的输出数据.只是 DeviceNet 主站模块不把 PLC 的输出数据发送到网上,直到 PLC 返回到运行状态.根据它的配置,它可以在停止状态下发送以下两种数据:

- **Idle Messages (zero length)** – 传送空闲 是默认状态.在传送空闲状态下, DeviceNet 主站模块 发送给所有已配置的 DeviceNet 设备而且长度为 0 的 I/O 消息.每个设备检测这个消息为“收到空闲 Receive Idle”条件,这是在 DeviceNet 规范中定义的.设备然后按照配置好的安全状态动作执行
- **Zeroed Data** – 在发送清零数据状态, DeviceNet 主站模块 将所有发给设备的输出数据置 0.

如果 DeviceNet 主站模块 在 PLC 处于停止状态时接收一个新配置,它将关闭与所有 DeviceNet 设备建立的连接并处理新的配置.

当 从停止状态切换到运行状态操作

当 PLC 从停止状态切换到运行状态, DeviceNet 主站模块 把最新的 PLC CPU 发送的输出数据 发送给 DeviceNet 网上的节点.如果没有收到过 CPU 的输出数据,网络输出数据都是 0.

处于运行状态时 DeviceNet 主站模块的工作

当 PLC 处于运行状态, DeviceNet 主站模块 和网络上设备按照配置好的连接属性 (轮询、选通、状态改变、循环通讯) 进行通讯: .

- DeviceNet 主站模块 首先发送选通多播消息.被配置成选通 I/O 通讯的设备按照它们 MAC ID 的顺序开始回应.
- 模块接着询问每个被配置成轮询的设备连接.轮询可以在收到选通回应时开始.模块会尽可能快地进行轮询.
- 模块按照配置的更新速率询问每个被配置成循环通讯的设备.而后这些设备把自己的输入数据返给模块.
- 配置成状态改变的网络设备只有在它们的状态发生改变时才会给主站发送消息.在收到一个来自设备的状态改变消息之后,主站发送一个请求消息给这个设备.该设备则把它的新数据发过来.

DeviceNet 主站模块把从网上收到的数据存入它自己的内存.当正常工作时, CPU 会在它正常输入扫描时从系列 90-30 DeviceNet 主站模块读入输入数据. PLC CPU 把新的数据更新到每个设备被分配的内存地址上.

当执行完应用程序之后, PLC CPU 发送输出数据给 DeviceNet 主站模块. 模块按照网络设备的 I/O 通讯类型 把输出数据发送给网络设备.

在 处于运行模式下如果 不能从模块读入输入数据时的工作方式

如果 PLC CPU 在 1 秒的间隔内没有从 DeviceNet 主站模块读取输入数据, 模块默认 CPU 不再通讯. 模块继续监视来自网上的输入数据但不把数据发送给 PLC CPU. 它还停止向网络发送输出数据. 改为它在停止模式下要么发送 idle 消息要么发送 zeroed 数据的工作方式. 网络设备进入它们自己配置的安全状态.

Detecting 检测总线丢失错误

如果发生 DeviceNet bus-off 故障, 模块不能与网络通讯了. 模块的状态数据(在这章给出) 中的 CAN 状态部分标明一个 bus-off 情况发生了. 对于 DeviceNet 主站模块, 在内部连接超时发生时, 每个节点的状态都标明这个设备不再是活动的了. 应用程序既可以监视 CAN 状态字也可以监视从站的状态位来检测 bus-off 情况是否发生或者有从站丢掉.

fatal Errors 致命错误

如果系列 90-30 DeviceNet 模块发生了致命错误, 如果可能它会采取以下动作:

- 将模块状态 LED 置为红色常亮并且熄灭网络状态 LED.
- 通知 PLC CPU.
- 停止一切当前的网络操作.

模块尝试维持与 PLC CPU 的通讯, 并且当它收到 COMMREQ #9"读模块头部"时, 它会作出响应. 另外, 模块接着开始发送 Module Header 信息到 RS-232 串口, 使用以下通讯参数: 19200 波特率, 8N1 (8 位数据位, 没有校验, 1 位停止位):

- 整个 Module Header 以 ASCII 的十六进制编码.
- Module Header 的消息字段使用可读形式.

系列 90-30 从站(服务器) DeviceNet 模块的工作方式

系列 90-30 DeviceNet 从站模块作为 DeviceNet 主站的服务器进行工作。系列 90-30 DeviceNet 主站模块也可以配置成服务器, 并且可以执行与 90-30 系列 DeviceNet 从站模块一样的。对于 DeviceNet 主站模块, 服务器属性的使用是可选的。

服务器属性使用模块的网络设置进行配置。模块可以被配置成两种 I/O 消息连接。也可以配置使用或不使用 DeviceNet Explicit Messaging。

系列 90-30 DeviceNet 模块的 I/O Messaging

PLC 参数被分配给为服务器配置的每个 I/O Messaging 连接。DeviceNet 网上的主站使用这些 I/O Messaging 连接读写系列 90-30 PLC CPU 内最多两个内存区的数据。一个内存区用于轮询一个内存区用于循环、选通或 COS I/O。

90-30 系列 DeviceNet 模块的 Explicit Messaging

如果网络设置中打开 Explicit Messaging 功能, 模块可以接收来自网络主站的 DeviceNet Explicit Messages。模块自动地回复 Explicit messages, 除非是包含用户定义的对象 Explicit Messages, 这些对象被串行地传送给系列 90-30 PLC CPU, 并由用户的应用程序负责处理它们。PLC CPU 中的应用程序逐个地读出排列着地 queued Explicit Messages 并且作相应的回复。这个处理过程在第 5 章中解释, "通讯编程"。

错误处理

如果发生错误, 系列 90-30 DeviceNet 从站模块会按照上一頁的 DeviceNet 主站模块如何工作的步骤处理。

系列 90-30 DeviceNet 模块的故障表条目

系列 90-30 DeviceNet 模块能自动地报告错误信息到系列 90-30 PLC 的故障表屏幕。

PLC Date/Time: 02-11-2000 19:29:42		Fault Table Viewer					Status																														
Last Cleared: 02-11-2000 19:29:32							Online																														
I/O Fault Table (Displaying 2 of 2 faults, 0 Overflowed)																																					
Loc (rds/slt)	CIRC No.	Ref. Address	Fault Category	Fault Type	Date/Time																																
0,2	0	%I 00209	Add'n of Device	undefined	02-11-2000 19:30:06																																
<table border="1"> <thead> <tr> <th>I/O Bus</th> <th>Bus Address</th> <th>Point Address</th> <th>Group</th> <th>Action</th> <th>Category</th> <th>Fault Type</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>n/a</td> <td>0</td> <td>7</td> <td>2:Diagnostic</td> <td>131</td> <td>0</td> </tr> <tr> <td colspan="8">Fault Extra Data: 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td> </tr> <tr> <td colspan="8">Fault Description: undefined</td> </tr> </tbody> </table>								I/O Bus	Bus Address	Point Address	Group	Action	Category	Fault Type	1	n/a	0	7	2:Diagnostic	131	0	Fault Extra Data: 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00								Fault Description: undefined							
I/O Bus	Bus Address	Point Address	Group	Action	Category	Fault Type																															
1	n/a	0	7	2:Diagnostic	131	0																															
Fault Extra Data: 00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00																																					
Fault Description: undefined																																					
0,2	0	%I 00209	Loss of Device	-	02-11-2000 19:29:47																																
<table border="1"> <thead> <tr> <th>I/O Bus</th> <th>Bus Address</th> <th>Point Address</th> <th>Group</th> <th>Action</th> <th>Category</th> <th>Fault Type</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>n/a</td> <td>0</td> <td>3</td> <td>2:Diagnostic</td> <td>130</td> <td>0</td> </tr> <tr> <td colspan="8">Fault Extra Data: 00 03 03 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00</td> </tr> <tr> <td colspan="8">Fault Description: undefined</td> </tr> </tbody> </table>								I/O Bus	Bus Address	Point Address	Group	Action	Category	Fault Type	1	n/a	0	3	2:Diagnostic	130	0	Fault Extra Data: 00 03 03 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00								Fault Description: undefined							
I/O Bus	Bus Address	Point Address	Group	Action	Category	Fault Type																															
1	n/a	0	3	2:Diagnostic	130	0																															
Fault Extra Data: 00 03 03 00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00																																					
Fault Description: undefined																																					

这个自动故障报告功能可以通过设置模块的软件配置中的 **Slave Status Fault Table Entries** 参数为 False 关闭。无论是否把信息显示到故障表, 状态信息仍能从模块的状态参数获得。参数含义下页给出。

关于不同类型的故障会导致 PLC 的工作产生哪些反应, 参考系列 90-30 系统手册, GFK-1411 或系列 90-30 参考手册, GFK-0467。

I/O 故障表条目

I/O 故障表可以显示以下几个系列 90-30 DeviceNet 模块故障:

设备丢失 - 标明配置的从站中有个 DeviceNet 故障. 例如, 如果主站发现从站超时但是网络没有丢失,且从站状态表中的那个设备位从 1 变到 0 时, 主站会记录一个设备丢失故障, .

设备增加 - 标明从站状态表中的某一位设备状态位从 0 变到 1 时, 说明一个从站设备在 DeviceNet 扫描表中是活动的.

运行时错误- 一个运行时错误会影响 DeviceNet 模块或网络的工作情况. 配置错误, 初始化错误, 和总线错误都是可能的运行时错误. 因为这些故障的本质, 有些不会报告给故障表. 但是, PLC 应用程序可监控模块的三个状态字来发现多种错误. 如果发现一个故障, 程序可一个通过给模块发送 Read Module Header COMMREQ 命令来获得更多的故障信息. 更多信息, 参见第 5 章, "可编程的通讯".

故障表条目

PLC 故障表可列举出 系列 90-30 DeviceNet 模块的一个故障:

系统复位 - 这个故障 会在由 Identity 复位服务引起 或 DeviceNet 网络电源的再次上电引起的一个 DeviceNet 请求时产生.

90-30 系列 DeviceNet 模块的 PLC 状态参数

90-30 系列 DeviceNet 模块自动报告存放状态信息的三个字, 这三个字的位置由 PLC 配置参数(网络状态/固件 ID)给出. 应用程序逻辑可以监视这些参数并且采取适当的动作以响应特别的变化. 这可以使用连接来或监视故障表里的故障.

这三个状态字的格式对于两种模块来说都是一样的. DeviceNet 主站模块里的 word 1 (服务器状态) 表示仅当本模块被设置成网络上另一台主站设备的从站(服务器)时有效.

映射到状态字的相同信息可以使用 COMMREQ #6, 获取状态信息, 直接从模块读取该命令的使用在第 5 章中有描述, "通讯编程".

Word 1		Server Status							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
byte 0	res.	AKS	CYC	COS	res.	ST	P	EX	
byte 1	reserved				SERA	IDLE2	IDLE1	G3	
		Group 2 only I/O connections		AKS	确认抑制使能				
				CYC	循环 I/O 连接被分配				
				COS	状态改变 I/O 连接被分配				
				ST	位选通 I/O 连接被分配				
				P	轮询 I/O 连接被分配				
		Group 2 Explicit Connections		EX	Explicit 连接被分配				
		Group 3 Connection		G3	至少有一个 Group 3 (UCMM) 连接被分配				
		Status Bits		IDLE1	输出区 1 收到 idle 状态位.				
				IDLE2	输出区 2 收到 idle 状态位				
				SERA	服务器 Explicit 请求可用. 使用接收 explicit 命令接收请求				
Word 2		CAN 网络状态							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	
byte 0	ML	RO	TO	TA	A	BO	BW	OL	
byte 1	SA	O5	O2	O1	RE	reserved	BP	ER	
		<i>应用程序专用标志</i>							
		SA		扫描器是活动的(至少建立了一个连接)					
		O5		在线 通讯速率 500 Kbaud					
		O2		在线 通讯速率 250 Kbaud					
		O1		在线 通讯速率 125 Kbaud					
		RE		固件正在复位 DeviceNet I/O 数据无效					
		<i>通用标志</i>							
		BP		总线有电(如果电源 sense 不支持则为 0)					
		ER		CAN 通讯错误					
		ML		消息丢失(CAN 控制器 / 收到 ISR)					
		RO		接收缓冲区溢出 (主机应用程序清理接收队列的速度太慢)					
		TO		超时, 传送失败(网络阻塞)					
		TA		确认错误, 传送失败(没有连接到其他节点)					
		A		检测到网络有活动(收到消息或发送消息)					
		BO		掉线(错误太多, 导致节点被断开)					
		BW		总线警告(这个节点正遇到大量的错误)					
		OL		在线, CAN 接口已被初始化					

Word 3	固件 ID, 小修订版本号:	以 BCD 编码的四位十六进制数字. 例如, 修订版 1.10 = 01 10 hex.
	固件 ID, 主修订版本号:	参见上行.

系列 90-30 DeviceNet 主站模块的设备状态位

除了维护前面描述的三个状态字之外系列 90-30 DeviceNet 主站模块还在 PLC 内存中维持 64 个设备状态位. 每一位 对应每一个可能的设备. 这个状态区的内存的类型和起始偏移量是可配置的.

一个单独设备的状态位等于起始地址加设备的站地址(MAC ID). 例如, 如果状态位区起始于 %I00001, 站地址是 5 的设备位为 %I00001 + 5, 即 %I00006. DeviceNet 主站模块的状态位也对它的 MAC ID. DeviceNet 主站模块自身的状态位一直为 0.

对于其他设备, 如果状态位为 0, 则对应这个 MAC ID 的设备不是活动的(未配置或有故障). 如果状态位为 1, 则对应这个 MAC ID 的设备是活动的. 这些位从 0 跃迁至 1 或从 1 跃迁到 0, 都会作为添加新模块或模块丢失故障添加到 I/O 故障表中.

本例中所示的状态位代表有 25 台设备 MAC ID 被分配成从 0 到 24. 系列 90-30 DeviceNet 主站模块 被配成 MAC ID #0 并且 MAC ID #8 的设备故障, 所以 %I00001, %I00009, 和 %I00026 至 %I00064 都是 0:

1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	%I00001
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	%I00017
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	%I00033
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	%I00049

通过使用 COMMREQ #6, 也可获得设备状态信息, 该命令为 获取状态信息, 在第 6 章, "通讯编程"中描述.

本章阐述应用程序如何使用通讯请求 与系列 90-30DeviceNet 主站模块或 系列 90-30 DeviceNet 从站模块交换信息. COMMREQ 用于读取 DeviceNet 模块本身的信息和发送或者回复 DeviceNet explicit 消息 s. 本章分为以下几节:

- **系列 90-30DeviceNet 模块使用的 COMMREQ**
- **使用 COMMREQ 对通讯编程**(概述)
- **COMMREQ 编程要求和建议**
- **读取识别码、状态和错误信息**(从系列 90-30 DeviceNet 主站或从站模块)
 - 模块型号、模块 ID、模块修订版本.
 - CAN 内核标识码和修订版本.
 - DeviceNet 序列号.
 - 错误代码.
 - CAN 网络状态.
- **取得网络从站的状态**(只适用于系列 90-30 DeviceNet 主站模)
 - 是否包含在主站的配置设备列表之中
 - 是否正被扫描
 - 配置错误状态
 - 连接 1 和 连接 2 输入状态
- **取得从站状态信息**(系列 90-30DeviceNet 主站 或从站模块的)
 - 模块是否设置成从站工作模式
 - 模块的输出连接状态
 - 模块是否发送过 DeviceNet explicit 消息.
 - 怎样 设置模块的 I/O messaging .
- **取得来自设备的输入状态**(从 系列 90-30DeviceNet 主站模块)
 - 网上每个 MAC ID 的网络活动状态
 - DeviceNet 主站模块自身的网络配置.
 - DeviceNet 主站模块的当前网络状态.
 - DeviceNet 主站模块的固件 ID.
- **发送一个 DeviceNet Explicit 消息到网络**(系列 90-30DeviceNet 主站)
- **读取或回应来自网上的 DeviceNet Explicit 消息**(系列 90-30DeviceNet 从站模块或被配置成按从站工作的系列 90-30 主站模块)

关于使用 COMMREQ 编程的一般信息，请参考编程软件的联机帮助。

用于系列 90-30 DeviceNet 模块的 COMMREQ

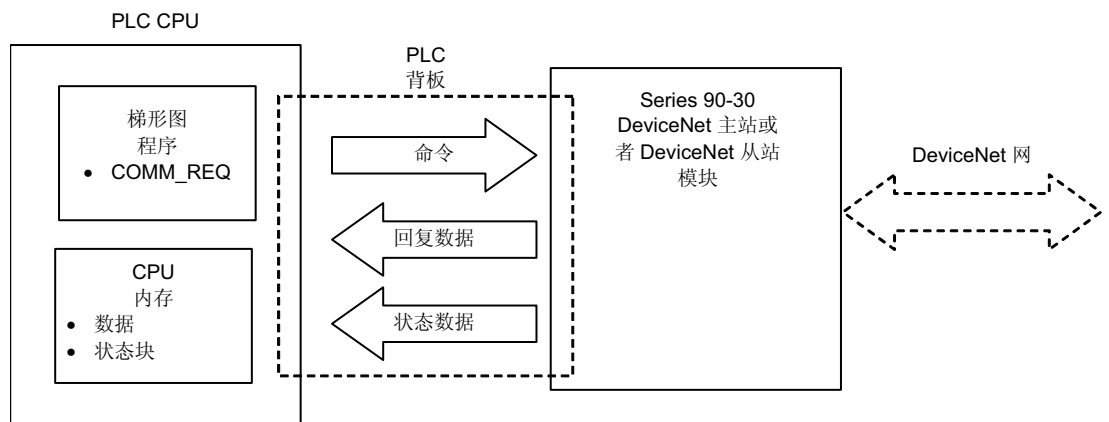
系列 90-30 DeviceNet 主站 模块 和 从站模块 可以使用下列 COMMREQ. 关于 COMMREQ 的详细描述在接下来的几页中介绍.

命令代码	支持特性	命令 Name	描述
1	主站	发送 设备 Explicit	用于通过 DeviceNet 网络发送和接收 explicit 消息给服务器设备. 最多 238 个服务数据字节.
2	从站	接收服务器 Explicit	用于取得模块内的一个发送给服务器的 explicit 请求.
3	从站	发送 服务器 Explicit	用于 提供发送给服务器的 explicit 请求。比如回复客户端的请求. 最多 238 个服务数据字节.
4	主站	取得详细的设备状态	用于取得特定网络节点的详细状态信息.
5	从站	取得详细的服务器状态	用于取得模块的服务器功能的细节信息.
6	主站 & 从站	取得状态信息	用于取得模块状态字节, 这些字节也可用于配置 PLC I/O. 另外的一种访问这个数据的方法. 包括设备状态, 服务器状态, CAN 状态和 固件 ID.
7	主站	发送设备 Explicit Extended	扩展版本的发送设备 Explicit (命令代码 1). 没有 238 个字节的限制并且允许是 PLC 内存中不同的数据区.
8	从站	发送服务器 扩展的 Explicit	扩展版本的发送 服务器 Explicit (命令代码 3). 没有 238 个字节的限制并且允许是 PLC 内存中不同的数据区.
9	主站	读取模块头部	用于 取得模块内客户端的详细状态.

使用 COMMREQ 进行通讯编程

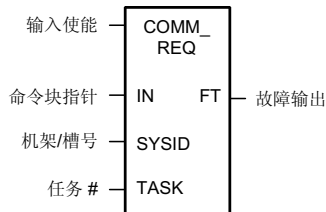
当 PLC 应用程序中的 COMMREQ 梯形图指令 激活后, 通讯请求启动. CPU 发送 COMMREQ 给 PLC 系统中的一个智能模块, 这个地方指的是 系列 90-30 DeviceNet 主站或从站模块. 模块收到该命令并执行被请求的操作. 一些 DeviceNet 主站或从站使用 COMMREQ 读写模块数据. 另外一些 COMMREQ 可以让 DeviceNet 模块去执行网络上的 DeviceNet 消息. 这种情况下, COMMREQ 和 DeviceNet 消息的数据以能被 DeviceNet 接收端识别的格式表示

每个 COMMREQ 执行完毕后, PLC CPU 将工作状态放入 CPU 内指定的内存.



通讯请求功能块的格式

通讯请求(COMMREQ) 功能块有如下参数:



OMM E 功能块的参数

输出/输出	选项	描述												
Enable	Flow	这个功能块得电, 执行通讯请求. 要使能 COMMREQ 指令必须是逻辑 1. 使能逻辑可以使用一个瞬时接通线圈作触点.												
IN	R, AI, AQ	指向命令块首字的指针. 例如, IN 为 %R00100, 则意味着命令块的起始地址是 %R00100. 命令块的长度取决于执行什么样的命令. 所有系列 90-30DeviceNet 模块命令的命令块在本章给出.												
SYSID	I, Q, M, T, G, R, AI, AQ, constant	用十六进制数表示目标设备的机架号(靠左的字节)和槽号(靠右的字节). 如: 机架 0 槽 4 = 00 04 机架 2 槽 9 = 02 09												
TASK	1	无论 DeviceNet 主站还是 DeviceNet 从站模块都必须置 1												
FT	flow, none	<p>FT (故障) 输出 可以给程序提供一个输出来验证通讯请求命令是否被成功地执行了. 在以下情况下, FT 输出会变成高位:</p> <ul style="list-style-type: none"> 指定的目标地址不存在(例如, 系统只有机架 0, 却指定了机架 1). 指定的任务号对设备无效. 数据长度被置成 0. <p>FT 输出 可以被连接到其他的设备, 如置位线圈, 或什么都不接.</p> <p>注意: 如果 COMMREQ 指令通过“瞬时接通”被执行, 那么 FT 输出状态值只在一次 PLC 扫描之内有效.</p> <p>FT 输出可有这样的状态:</p> <table border="1"> <thead> <tr> <th>使能输出状态</th> <th>是否有错误</th> <th>输出</th> </tr> </thead> <tbody> <tr> <td>活动的</td> <td>无</td> <td>低</td> </tr> <tr> <td>活动的</td> <td>有</td> <td>高</td> </tr> <tr> <td>不活动的</td> <td>不执行</td> <td>低</td> </tr> </tbody> </table>	使能输出状态	是否有错误	输出	活动的	无	低	活动的	有	高	不活动的	不执行	低
使能输出状态	是否有错误	输出												
活动的	无	低												
活动的	有	高												
不活动的	不执行	低												

COMMREQ 编程要求和建议

- 在执行 COMMREQ 指令前, 数据必须放在命令块中. 有些命令还要设置更多的数据区. 因为正常的 PLC 扫描顺序是从上到下, 所以这些操作应该在 COMMREQ 块所处的行之前完成.
- 如果你使用 MOVE 指令来给命令块 寄存器装入值, 使用字-类型的 MOVE 装入一个十六进制的数, 使用整型-类型的 MOVE 装入一个十进制数.
- COMMREQ 指令应该由一个跃迁线圈激活.
- 如果使能输出不是活动的, 则 FT 输出一直为 False . 如果 COMMREQ 被一个跃迁 (one-shot)触点激活并且发生了一个故障, FT 输出仅在当次 PLC 扫描时变为高位. 要在故障发生时采取行动, 可以把故障输出设置成一个置位线圈, 这个线圈不会在本次扫描中变为复位状态. 其余的逻辑就需要在故障被确认之后和下次执行 COMMREQ 之前, 复位故障输出线圈.
- 使用一个设备编程, 比如, COMMREQ FT 输出端上的置位线圈是可选的. FT 输出可以是开路的.
- COMMREQ 必须按顺序发送. *新 COMMREQ 只有在前一个 COMMREQ 完成之后* (要么成功要么失败)才可能通过状态块 *发送出去*. 当一个梯形图程序使用多个 COMMREQ 时, 在执行一个 COMMREQ 之前要检查前一个 COMMREQ 是否已经成功执行. 通过检查状态字和 FT (Fault) 输出可有达到这个要求.

检查 COMMREQ 的执行情况

当 COMMREQ 功能块有某个故障时, FT 输出变为 高位。这可以用于故障检测. 除了处理使用 FT 输出, 程序还应该监视第一个 COMMREQ 状态字, 并且使用错误消息逻辑生成操作员接口设备上显示的文字. 这时, 特定的状态字错误代码可以显示在显示器上, 给操作工恰当的操作信息.

如果 COMMREQ 被成功地执行了, CPU 把 COMMREQ 的“状态” 状态字[状态字 1]置为 1. 如果检测到有错误, 会返回一个大于 1 的值. 要在梯形图中检测错误, 可用大于(GT) 比较指令来检查状态字的值是否为负值(小于零). 如果有错误发生了, GT 指令的输出 (Q)变成高位. 由输出端驱动的线圈可用于激活故障处理或故障报告逻辑.

要动态地检查状态状态, 可以在每次和它关联的 COMMREQ 执行前, 写一个超出范围 0x01-0x0D 或 0x0E-0xFF 的无符号正整数 (常用 0) 给所有四个状态字. 如果指令成功执行, CPU 把“状态” 状态字置 1. 如果是数字 1, 表示最后一个 COMMREQ 被成功的执行的; 1 不表示前一个 COMMREQ 执行的结果.

当应用程序扫描中有多个 COMMREQ 时, 在执行下一个 COMMREQ 之前检测前一个 COMMREQ 是否被成功执行, 这点十分重要. 而检测状态字就可以做到.

OMM E 发生错误时应该采取的正确行动

处理 COMMREQ 错误采取什么样的对应行动取决于应用程序。如果在启动或调试梯形图程序过程中发生了错误,需要检查 COMMREQ 参数。还应该在修改完程序随即就发生错误时,检查 COMMREQ 参数。

当正常运行的无误程序发送错误时,出错原因可能是与硬件有关。在排除 COMMREQ 错误时,可以检查 PLC 故障表以取得更多信息。

使用 OMM E 进行 DeviceNet Explicit Messaging

当使用 COMMREQ 命令来发送或接收 explicit 消息时,按照正确的步骤操作十分重要。DeviceNet explicit 消息接口是半双工接口,这意味着被一个配置好的设备只可以处理一个命令。模块总会对 explicit 命令提供响应[在 COMMREQ 状态内]。如果这个设备没有响应,模块自己产生一个错误响应以代表未响应的设备,从而使得在 PLC 一级的命令/回复序列不会被挂起。恰当的序列为:

- a.) 将完整的 COMMREQ 命令信息装载到 PLC 内由 COMMREQ 命令指定的 explicit 消息传送缓冲区内存。
- b.) 将命令指出的 COMMREQ 状态内存(4 个字)值设置到一个已知的,清除的状态。
- c.) 梯形图中,使用跃迁 (one-shot)执行 COMMREQ 命令。
- d.) PLC 应用程序一定要检查 COMMREQ 状态是成功状态(状态 = 1),或是失败状态(状态 > 1)。如果命令没有成功,经过一个延时就会返回一个失败状态。进入失败状态后,再决定采取一定措施:重试命令,向操作员报警,增加故障计数,停止 PLC,或跳到下一个消息。
- e.) 如果处于成功状态,处理回复数据的方式取决于被执行的命令。你在 COMMREQ 命令中指定存放回复数据的存储位置。不过只有被证明是成功之后,才可以把回复缓冲起来并将它复制到 PLC 的其他位置。缓冲区是 PLC 内存内的一个区域,用于存储回复数据直到验证了它是命令成功执行的结果时为止。当 COMMREQ 状态= 1,数据可以被复制到它最终位置。如果在故障状态,回复数据不是预计的数据而是多个描述故障模式的数值。在应用程序中那些需要测试一个回复是否正确地方,推荐使用缓冲区。
- f.) 对于下一个 explicit 消息从命令序列开始的地方启动。

注意

PLC 程序不要设置针对 explicit messaging 命令回复的超时值,这一点非常重要。如过设置了超时值,命令/回复序列就有可能失去同步机制并阻止其他正常的 explicit 消息产生。

系列90-30DeviceNet 模块使用的 OMM E 状态块

COMMREQ 状态块是一个四个字长.每个状态字的数据类型是 UINT. 状态块可位于任何 PLC 支持的内存区域.

在每个 COMMREQ 前, PLC 应用程序**必须**设定状态块的值为 0.

字	名称	描述	
1	状态: 当前 COMMREQ 请求的状态		
	0x00	模块还未处理完 COMMREQ	
	0x01	命令完成 注意: 这个状态不一定表示成功. 有些带回复数据的命令必须进行检查.	
	0x02	忙 - 命令正在处理, 现在还没有完成 注意: 不保证在完成或终止前状态一定变成忙.	
	0x03	命令被终止 - 无效命令	
	0x04	命令被终止- 无效命令数据	
	0x05	命令被终止- 数据不足	
	0x06	保留	
	0x07	命令被终止- 回复区域的内存不足 命令 没有为回复指定足够的 PLC 内存. 命令 将被忽略.	
	0x08	命令被终止- 命令-特殊错误. 参见状态块中的错误代码和附加代码以取得更多的信息.	
	0x09	命令被终止- 无效的 COMMREQ	
	0x0A	命令被终止- 特殊段选或 COMMREQ 回复不被支持	
	0x0B	命令被终止- 回复写 PLC 内存失败	
	0x0C	命令被终止- 特殊段选或 COMMREQ 数据不被支持	
	0x0D	命令被终止-读 PLC 内存失败	
0x0E to 0xFF	保留		
2	丢失 命令	最后一个命令的命令代码丢失. 如果没有命令丢失则为 0.	
3	错误 代码: 含义取决于命令编号		
	命令	错误代码	
		0	保留
	1,3,7,8	1	Explicit 数据对于共享内存缓冲区太大了. 附加代码字保存共享内存缓冲区实际大小.
	1, 4, 7	2	指定了无效的 MacID.
	1,2,3,7,8	3	未被配置的 Explicit 连接.
2	4	Explicit 请求不可用.	
4	附加代码: 用于报告错误.		

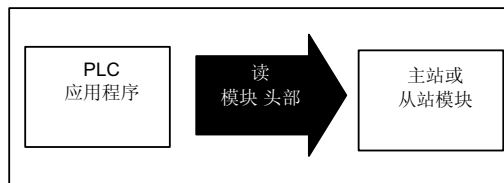
命令代码 9: 读取识别信息、状态和错误信息

使用 **COMMREQ #9, 读模块头部**, 从系列 90-30 DeviceNet 主站或从站模块中取得下列信息,:

- 模块型号, 模块 ID, 模块修订版本.
- CAN 内核标识符和修订版本.
- DeviceNet 序号.
- 已出现故障的错误代码.
- CAN 网络状态.

这个命令读取模块内部内存的数据;不需要向网络发送 DeviceNet 消息.

90-30 系列 PLC



一旦检测到错误, PLC 应用程序能给模块发送一个读模块头部 COMMREQ. 除非故障阻止了背板正常工作, 否则模块能够返回一个回复, 而故障信息在回复数据中. 错误代码在本章中列出.

读模块头部 OMM E 举例

下面是使用 COMMREQ 的例子:

- 取得模块头部数据
- 返回 COMMREQ 状态字存放在 %R10-%R13
- 返回设备状态存放在 %R251-%R283.

字#	Dec	(Hex)	描述
1	00004	(0004)	命令块长度: 对于命令 9 一直为 8 个字节(4 个字).
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字的内存类型(本例为 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: 状态字的起始地址减 1. (本例是 %R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00009	(0009)	命令代码: 读模块头部; 命令 9
8	00008	(0008)	回复段选: 回复数据的内存类型(本例使用的是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复内存偏移量: 回应内存类型的偏移量 减 1. (本例是 %R251).
10	00065	(0041)	回复内存大小: 回复的最大长度(以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22). 最大 2048 个字节. 注意: 对于命令 9 必须是 130 个字节(65 个字)或更多, 或返回到 COMMREQ 状态中的错误 然后这个命令会被忽略.

读模块头部 回复数据格式

字	描述								
1	命令 代码. 数据块正在回复的命令代码 (0x0009)								
2	模块型号.其值为 "DN" (0x444E)或"ER" (0x4552) 如果检测到严重的错误时								
3	窗口大小: 标明主机接口窗口的大小. 0 = 16K, 1 = 32K, 2 = 64K, 3=128K								
4	保留								
5	内核标识符. 0x0001 = CAN 2.0A 内核								
6	内核修订版本								
7	模块 ID, 0x0017 (系列 90-30DeviceNet 模块)								
8	模块 修订版本 以二进制编码的十进制数(BCD), 4 位十六进制数字 XX.XX (i.e. rev 1.0 = 0x0100, rev 1.10 = 0x0110)								
9,10	DeviceNet 序号								
11 - 18	卡型号, IC693DNM200 或 IC693DNS201								
19 - 22	模块 序号 (i.e. "9409001")								
23, 24	保留								
25	主程序错误代码. 参见下页的错误代码列表.								
26	CAN 网络状态字.								
		bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	字节 0	ML	RO	TO	TA	A	BO	BW	OL
	字节 1	SA	O5	O2	O1	RE		BP	ER
	应用程序-特殊标志								
	SA	扫描器处于活动状态(至少有连接建立成功)							
	O5	在线 500 K 波特率							
	O2	在线 250 K 波特率							
	O1	在线 125 K 波特率							
	RE	固件正在执行 DeviceNet 复位, I/O 数据是无效的							
	普通标志								
	BP	总线有电(如果不支持电源感应器, 其值为 0)							
	ER	CAN 通讯错误							
	ML	消息丢失(CAN 控制器 / 接收 ISR)							
	RO	接收缓冲区溢出(主机应用程序清理接收队列的速度太慢)							
	TO	超时引起传送失败(网络繁忙)							
	TA	由于确认信息错误引起传送失败(没有与其他节点连接上)							
	A	检测到网络活动信息(接收到消息或有消息发送)							
	BO	掉线(由于过的的错误本节点被断开)							
	BW	总线警告(本节点有许多错误)							
	OL	在线, CAN 接口已被初始化							
27	CAN 传送计数器. 在有消息被提交给 CAN 控制器时增加.								
28	CAN 确认错误计数器. 如果传送的消息由于没有被其他站确认而被终止时计数器增加. 当这个计数器增加之后, CAN 传送计数器(字 27)会减少以补偿没有被真正传送的消息.								
29	CAN 接收计数器. 当收到消息时增加. 消息即便被接收过滤器过滤, 计数器也会增加.								
30	CAN 通讯错误计数器. 如果发现一个 CAN 帧错误,则计数器就会增加.								
31	CAN 丢失消息计数器. 如果在前一个信息被放入接收队列之前又收到一个 CAN 消息, 则计数器增加.								
32	CAN 接收队列溢出计数器. 如果 CAN 消息 由于接收队列已满而丢失, 则计时器增加.								
33	附加应用代码代码. 参见下页的错误代码列表.								
34 - 63	当字 2 的模块型号是 "DN"时,存储的是模块标识字符串.例如: "DeviceNet Module 1.00.00\n(C) 2002 GE Fanuc Automation." 格式是: 主版本号.次版本号.生成序号 如果模块型号是"ER",存放的是内核错误字符串.								
64	主 Tick 间隔 (相当于系统时间基准)								
65	每个主 tick 间隔中的小 Tick 数								

模块头部的运行时错误代码

运行时错误[读模块头部的字 2 ,回复数据 = "DN" (0x444E)]之后, 回复数据的主错误代码 [字 25] 和 附加代码 [字 33]中的值描述运行时错误. 主错误代码字是 0 时, 表示没有错误.

目录名	主代码	错误名称	附加代码	描述
没有错误	0x0000	-	0x0000	主代码为 0 表示没有错误
配置文件错误	0x0001	未知版本	0x0001	不正确的 /不支持的版本
初始化文件错误	0x0002	未知版本	0x0001	不正确的 /不支持的版本
		未知的头部 Id	0x0002	初始化文件的头部 ID 不能被识别或无效的
		无效的块定义计数	0x0003	块的定义计数是无效的
		未知的块类型	0x0004	块类型无法识别
		无效的块校验和	0x0005	块校验和是无效的
		无效的共享内存偏移量	0x0006	共享内存偏移量 不在 0x1000 至 to 0x3FFF 范围之内
		未知的 9030Io 型号	0x0007	I/O 型号 代码不能识别
		无效的 Mac Id	0x0008	数据指针列表中的 Mac Id 不在 0 至 64 之内的范围,也不是 255.
		未知的 内存区域类型	0x0009	内存区类型代码不能识别
		无效的块大小	0x000A	初始化文件中的块的大小与预计的大小不匹配,或比固件支持的最大值还要大.
		无效的块偏移量	0x000B	块定义记录中的偏移量指向位置超出初始化文件定义的最大值.
		重复块	0x000C	使用相同类型代码的块已经存在于初始化文件.
		数据指针超范围	0x000D	数据指针指向的位置超出了共享内存的范围.
		丢失块	0x000E	初始化文件中有一个或多个所需块 1, 3, 和 4 丢失.
增加设备错误	0x0003	重复的设备	0x0005	设备已经在扫描列表的.
		无效的共享的内存偏移量	0x000D	共享的内存偏移量不在 0x1000 至 0x3FFF 范围内.
		无效的连接标志	0x000F	标志字段中的位组合是无效的.
		无效的 Explicit 缓冲区大小	0x0010	Explicit 缓冲区大小是无效的.
		无效的选通缓冲区大小	0x0011	选通缓冲区大小是无效的. 注意输出大小必须是 1.
		无效的路径缓冲区	0x0012	路径缓冲区没有初始化.
在线错误	0x0004	无效的 Mac Id	0x0002	服务器 配置块中的 MacId 不在 0 至 63 的范围内.
		无效的 波特率	0x0003	波特率 没有被设置成 0, 1,或 2 (分别对应. 125K, 250K,或 500K)
		重复的 Mac Id 故障	0x0004	在尝试切换成在线状态时发现了重复的 MacId.
		在线	0x0009	在线已经在线. 内部固件错误;向厂商报告.
		掉线	0x000E	检测的总线故障.
		无效的连接标志	0x000F	服务器 配置块的标志字段中的位组合是无效的.
		无效的 Explicit 缓冲区大小	0x0010	Explicit 缓冲区大小是无效的.
		无效的选通缓冲区大小	0x0011	选通缓冲区大小是无效的. 注意输出大小必须是 1.
		无效的路径缓冲区	0x0012	路径缓冲区没有初始化.
		确认故障	0x0013	在重复的 MacId 序列时没有收到 CAN 确认信息
启动扫描	0x0005	总线离线	0x0007	总线还未在线
		扫描器启动	0x000A	扫描器已经启动
		扫描器停止	0x000C	扫描器正在停止

模块头部中严重的错误代码

如果模块能够在发生严重的错误之后还能执行和报告一个错误,那么模块头部数据的模块型号字段变成“ER”(0x4552).

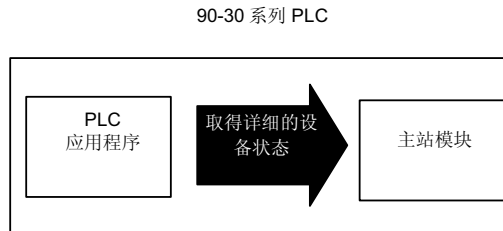
- 如果网络是活动的,则停止全部 DeviceNet 网络活动. 模块不能保证 DeviceNet 关闭消息能够正常的发送出去.
- 模块状态 LED 变成红色并且网络状态 LED 熄灭.
- 整个模块头部(以十六进制 ACSII 码编码)和模块头部的消息字段(以字符串形式)由 RS232 串口以 19200 波特率,无奇偶校验 8 位数据位 1 位停止位的模式持续向外界发送.
- 在 PLC 故障表中会增加状态标志和信息一个条目.
- 背板通讯继续工作(可以限制)以便能够使用 PLC COMMREQ (命令 9)读取头部数据.

下列的特殊错误没有任何源代码信息,主代码变为 0xFFFF 而且附加代码字段中错误代码是下表中某一个错误.能报告源代码位置的严重错误会把源文件号放在主代码字段 把源代码的行号放在附加代码.

附加错误代码	错误	描述
1	RAM 数据 测试失败	测试 RAM 数据总线失败. 模块需退回修理.
2	RAM 地址测试失败	测试 RAM 地址总线失败. 模块需退回修理.
3	RAM A16 地址测试失败	测试 RAM 地址总线 A16 失败. 模块需退回修理
4	RAM A17 地址测试失败	测试 RAM 地址总线 A17 失败. 模块需退回修理
5	模块校验和是无效的	可能为无法检测的内存错误. 如果多个应用程序模块都发生的这个错误,模块需退回修理.
6	清除 CAN 复位标志失败	检测 CAN 控制器失败. 模块需退回修理
7	CAN 数据测试失败	检测 CAN 控制器数据总线失败. 模块需退回修理
8	CAN 地址测试失败	检测 CAN 控制器地址总线. 模块需退回修理
9	无效的 NVRAM 数据	模块的 非易失性内存中有无效的信息. 模块需退回修理
10	执行请求被拒绝	这个 模块不可配置成执行应用模块. 与应用模块的经销商联系以获取帮助.
11	应用程序初始化错误	初始化应用模块失败. 向应用模块的经销商报告出错的情况.
12	未知的应用程序初始化代码	初始化应用模块模块. 向应用模块的经销商报告出错的情况
13	应用程序终止	应用模块被终止(异常的情况). 向应用模块的经销商报告出错的情况。
14	应用程序严重的错误	发生严重的运行时错误. 向应用模块的经销商报告出错的情况
15 - 21	XXX 中断	检测到一个意外的中断. 向应用模块的经销商报告出错的情况. 记录发生错误的环境条件.
22	事件队列溢出	向应用模块的经销商报告出错的情况. 记录发生错误的环境条件.
23	用户计时器中断嵌套	
24	无效的 CAN 中断	
25	系统计时器中断嵌套	
26	不完整的中断	向应用模块的经销商报告出错的情况. 记录发生错误的环境条件. 这个错误因主机总线适配器向模块产生了的不正确的中断而发生.
27	栈溢出	向应用模块的经销商报告出错的情况. 记录发生错误的环境条件.
99	发生意外的情况	发生严重的运行时错误. 向应用模块的经销商报告出错的情况

命令代码 4: 取得网络设备的状态

使用 **COMMREQ #4**, 取得详细的设备状态从系列 90-30 DeviceNet 主站模块取回某个网络设备的详细状态信息.



本命令可以得到指定节点的下列信息:

- 是否包含在主站的配置设备列表里
- 是否正被扫描
- 配置信息错误状态(无效的厂商 id、设备型号、产品代码、I/O 连接等等)
- 它的连接 1 和连接 2 的输出状态

这是系列 90-30 PLC 内部的功能; 它不产生 DeviceNet 网络消息.

取得详细的设备状态 OMM E 举例

下例中的 COMMREQ 执行以下操作:

- 从 DeviceNet 主站模块取得 MAC ID 为 #4 的从站设备的状态.
- 返回的 COMMREQ 状态存放在 %R10-%R13
- 返回的设备状态存放在 %R251-%R260.

字#	Dec	(Hex)	描述
1	00005	(0005)	命令数据块的长度: 对于取得详细的设备状态 COMMREQ, 一直是 5
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字的内存类型(本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: COMMREQ 状态字的地址减 1 (%R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00004	(0004)	命令代码: 取得详细的设备状态命令编号 4
8	00008	(0008)	回复段选: 回复数据的内存类型 (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复内存偏移量 回应中内存类型的偏移量减 1. 本例中的 %R251. (偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22)
10	00009	(0009)	回复内存大小: 回复长度的最大值 (以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22). 最大值为 2048 个字节. 注意: 对于命令 9 必须是 18 字节(9 字 s)或以上, 或为一个被返回到 COMMREQ 状态中的一个错误值然后该命令会被忽略.

11	00004	(0004)	MAC ID: 网络设备的. 本例是, 4.
----	-------	--------	------------------------

取得详细的设备状态 回复数据格式

当收到 PLC CPU 发送的 COMMREQ 4, 系列 90-30DeviceNet 主站模块产生一个回复, 回复中有指定 MAC ID 的状态数据.

字#	描述			
1	本数据块回复的命令编号. (4)			
2 低字节	状态代码: 编号表示连接到设备的客户端的状态.			
	状态	含义	状态	含义
	0x 00	设备不在设备列表中	0x 0D	无效的 I/O 连接 1 输出大小
	0x 01	设备空闲 (不能被扫描)	0x 0E	读 I/O 连接 1 输出大小错误
	0x 02	设备正在被扫描	0x 0F	无效的 I/O 连接 1 输出大小
	0x 03	设备超时	0x 10	读 I/O 连接 1 输出大小错误
	0x 04	UCMM 连接错误	0x 11	无效的 I/O 连接 2 输出大小
	0x 05	主站/从站连接集忙	0x 12	读 I/O 连接 2 输出大小错误
	0x 06	分配主站/从站连接集错误	0x 13	无效的 I/O 连接 2 输出大小
	0x 07	无效的厂商 id	0x 14	读 I/O 连接 2 输出大小错误
	0x 08	读厂商 id 错误	0x 15	设置 I/O 连接 1 数据包速率错误
	0x 09	无效的设备型号	0x 16	设置 I/O 连接 2 数据包速率错误
	0x 0A	读设备型号错误	0x 17	M/S 连接集同步故障
	0x 0B	无效的产品代码	0x 18	设置生产约束时间错误
	0x 0C	读产品代码错误	0x 19 - FF	保留
	2 高字节	状态标志: 这些位表示从站的连接 1 和 连接 2 输入中的连接状态.		
bits 0-4		保留, 应该被忽略		
bit 5		1 = 输出区 1 接收空闲条件		
bit 6		1 = 输出区 2 接收空闲条件		
bit 7		保留, 应该被忽略		
3 至 9	保留, 应该被忽略			

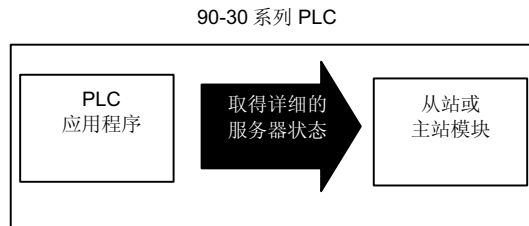
命令代码 5: 取得系列 90-30 DeviceNet 从站模块或主站模块的服务器功能的状态信息

使用 **COMMREQ #5**, 取得详细服务器状态, 获取系列 90-30 DeviceNet 从站模块状态信息以及以服务器模式工作的 DeviceNet 主站模块 的状态信息。

这命令提供下列模块信息:

- 模块是否被配置成从站(它的网络设置已被配置)
- 模块的输出连接状态
- 模块是否已经发送了 DeviceNet explicit 消息 (由之前的发送服务器回应 COMMREQ 引起的).
- 模块的 I/O messaging 设置是如何配置的.

这个功能是系列 90-30 PLC 内置的; 它不会产生 DeviceNet 网络消息.



获取详细的服务器状态 OMM E 举例

本例中, 应用程序发送一个获取详细的服务器状态 COMMREQ 给作为从站工作的系列 90-30 DeviceNet 主站模块 .

这个 COMMREQ 指定了命令编号和命令长度, 及 COMMREQ 状态和回复消息存储的内存位置.

字#	Dec	Hex	描述
1	00004	(0004)	命令数据块的长度: 本命令一直为 4 个字.
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字内存类型 (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: COMMREQ 状态字地址减 1 (本例为 %R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00005	(0005)	命令代码: 取得详细的服务器 状态命令 (5)
8	00008	(0008)	回复 段选: 回复数据的内存类型. (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复 内存偏移量: 回复内存类型的偏移量减 1. 本例中是 %R251.
10	00009	(0009)	回复 内存大小: 回复的最大长度 (以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22). 最大值为 2048 个字节. 注意: 对于命令 5 必须是 18 个字节(9 个字)或更多, 或是一个被返回到 COMMREQ 状态的错误值然后命令被忽略.

取得详细服务器状态 回复数据格式

取得详细服务器 状态 COMMREQ 的回应消息中有模块网络配置的详细信息. 这个回应也表明了模块是否已经将此前的发送服务器 Explicit 消息 发送出去了.

字#	十进制 二进制	十六进制	描述	
1	00005	(0005)	这个数据块要回复的命令编号(5)	
2 低字节	00000001	(01)	表明是否模块设置为从站状态(配置的网络设置). 本例中, 模块正在被扫描.	
			0x00	空闲 (Group 2 主站/从站连接未被配置, 从站服务器没有活动, 没有主站扫描).
			0x01	活动的 (Group 2 主站/从站连接被分配, 从站服务器是活动的并正被主站设备扫描).
			0x02- 0xFF	保留, 这些位应该被忽略.
2 高字节	位串表示多个连接的状态. 在本例, 模块的 1 和 2 连接都配置成接收空闲, 并且它有个 UCMM 连接.			
	11100000	(E0)	<i>bits 0 - 4</i> 保留, 这些位应该被忽略.	
			<i>bit 5</i> 1 = 输出 连接 1 接收空闲条件	
			<i>bit 6</i> 1 = 输出 连接 2 接收空闲条件	
			<i>bit 7</i> 1 = Group 3 UCMM 连接(s)配置的.	
3	00000	(0000)	保留, 这些位串应该被忽略.	
4 低字节	位串表示自最后一次取得详细服务器状态请求以来 explicit 消息的状态. 在准备下次调用 COMMREQ 前, 这些位串自动地被这个 COMMREQ 清零. 本例中, 模块已经发出 explicit 消息响应.			
	00001	(0001)	<i>bit 0</i>	1 = Explicit 回应已发出. 当扫描器提交到发送服务器 Explicit 消息的回应时, 置位.
			<i>bits 1 - 7:</i> 保留	
4 高字节	这些位表明模块已配置的属性. 本例中, 系列 90-30DeviceNet 模块从站服务器 被设置成 explicit messaging 和轮询 I/O 工作模式.			
	00000011	(03)	<i>bit 0</i>	1 = Explicit 连接已分配
			<i>bit 1</i>	1 = 轮询 I/O 连接已分配
			<i>bit 2</i>	1 = 位选通 I/O 连接已分配
			<i>bit 3</i>	未用
			<i>bit 4</i>	1 = 状态改变 I/O 连接已分配
			<i>bit 5</i>	1 = 循环 I/O 连接已分配
			<i>bit 6</i>	1 = 禁止确认使能
<i>bit 7</i>			未用	
5至9			保留, 这些位应该被忽略.	

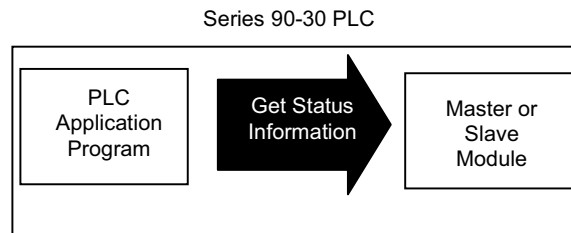
命令代码 6: 取得设备输出状态

使用 **COMMREQ #6, 取得状态信息**, 读取通常映射到系列 90-30 DeviceNet 模块的 3 个状态字的信息, 以及映射到 DeviceNet 主站模块的 64 个设备状态位串的所有信息. COMMREQ #6 可用访问到这个信息.

作为这个命令回应, 模块返回以下信息以响应:

- 网上每个 MAC ID 的活动状态
- 模块自身的网络配置.
- 模块当前的网络状态.
- 模块的固件 ID.

被命令读取的信息直接来自模块内部; 本命令不产生 DeviceNet 消息.



取得状态信息 OMM E 举例

这个例子中 COMMREQ 执行以下操作:

- 从 DeviceNet 主站取得状态信息.
- 返回的 COMMREQ 状态字存放在 %R10-%R13
- 返回的设备状态存放在 %R251-%R260.

字#	Dec	(Hex)	描述
1	00004	(0004)	命令数据块的长度: 对于取得状态信息, 长度是 4 个字(8 个字节).
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字的内存类型 (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: COMMREQ 状态字的起始地址减 1 (本例中是 %R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00006	(0006)	命令代码: 取得状态信息的命令编号(6)
8	00008	(0008)	回复段选: 回复数据的内存类型. (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复内存偏移量: 回复内存类型的偏移量 (从 0 开始). 本例中它是 %R251. (偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22).
10	00008	(0008)	回复内存大小: 回复的最大长度 (以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22). 最大值为 2048 个字节. 注意: 对于命令 6 必须是 16 个字节(8 个字)或以上, 或是一个被返回到 COMMREQ 状态的错误值, 那么该命令就会被忽略

取得状态信息 回复数据的格式

字#	描述							
1	这个数据块将要回复的命令代码 (6)							
2 - 5	设备状态. 每个位对应一个单独的设备 MAC ID. 这位的状态表示设备的状态: 0 = 设备处于不活动的状态(没有配置,故障,等...) 1 = 设备处于活动的状态, 可被扫描到 对于主站自己的 MAC ID, 该状态位一直为 0.							
	<i>bit 7</i>	<i>bit 6</i>	<i>bit 5</i>	<i>bit 4</i>	<i>bit 3</i>	<i>bit 2</i>	<i>bit 1</i>	<i>bit 0</i>
字节0	7	6	5	4	3	2	1	0
字节1	15	14	13	12	11	10	9	8
字节	23	22	21	20	19	18	17	16
字节	31	30	29	28	27	26	25	24
字节	39	38	37	36	35	34	33	32
字节	47	46	45	44	43	42	41	40
字节	55	54	53	52	51	50	49	48
字节	63	62	61	60	59	58	57	56
6	服务器状态							
	<i>bit 7</i>	<i>bit 6</i>	<i>bit 5</i>	<i>bit 4</i>	<i>bit 3</i>	<i>bit 2</i>	<i>bit 1</i>	<i>bit 0</i>
字节0	res.	AKS	CYC	COS	res.	ST	P	EX
字节1	保留				SERA	空闲2	空闲1	G3
	Group 2 只有 I/O 连接			AKS	禁止确认被使能			
				CYC	循环 I/O 连接被分配			
				COS	状态改变 I/O 连接被分配			
				ST	位选通 I/O 连接被分配			
				P	轮询 I/O 连接被分配			
	Group 2 Explicit 连接			EX	Explicit 连接被分配			
	Group 3 连接			G3	至少分配了一个 Group 3 (UCMM) 连接			
	状态位串			空闲1	输出区 1 接收空闲状态位.			
				空闲2	输出区 2 接收空闲状态位			
				SERA	服务器 Explicit 请求可用. 使用接收服务器 explicit 命令来取得这个请求			
7	CAN 网络状态.							
	<i>bit 7</i>	<i>bit 6</i>	<i>bit 5</i>	<i>bit 4</i>	<i>bit 3</i>	<i>bit 2</i>	<i>bit 1</i>	<i>bit 0</i>
字节0	ML	RO	TO	TA	A	BO	BW	OL
字节1	SA	O5	O2	O1	RE	保留	BP	ER

	<p><i>应用程序特殊标志</i></p> <p>SA 扫描器是活动的 (至少有一个连接建立成功)</p> <p>O5 在线 500 K 波特率</p> <p>O2 在线 250 K 波特率</p> <p>O1 在线 125 K 波特率</p> <p>RE 固件正在复位, DeviceNet I/O 数据是无效的</p> <p><i>普通标志</i></p> <p>BP 总线有电(如果不支持电源感应, 则为 0)</p> <p>ER CAN 通讯错误</p> <p>ML 消息丢失(CAN 控制器/接收 ISR)</p> <p>RO 接收缓冲区溢出 (主机应用程序清除接收队列的速度太慢)</p> <p>TO 超时引起发送失败(网络繁忙)</p> <p>TA 由于确认错误发送失败(连接的没有节点)</p> <p>A 检测到网络活动(收到消息或发送消息)</p> <p>BO 掉线(由于过多的错误这个节点被断开)</p> <p>BW 总线警告(这个节点有许多错误)</p> <p>OL 在线, CAN 接口已被初始化</p>	
8	固件 ID, 次修订版本:	四位 BCD 十六进制数字. 例如, 修订版本 1.10 = 01 10 十六进制.
	固件 ID, 主修订版本:	见上.

命令代码 1 和 7: 发送 DeviceNet Explicit 消息

使用 **COMMREQ #1, 发送设备 Explicit**, 发送最多 238 个字节的 DeviceNet explicit 消息到指定设备. 回复数据不能超过 2048 个字节.

使用 **COMMREQ #7, 发送扩展的设备 Explicit**, 发送多于 238 个字节的数据或使用 PLC 中独立的数据内存区. 回复的长度不能超过 2048 个字节.

这两个 COMMREQ 会命令 DeviceNet 主站模块向网上发送 DeviceNet explicit 消息. 被寻址的设备在系列 90-30 DeviceNet 主站模块的配置中必须被配置成 explicit 消息类型连接而且要有足够的缓冲区内内存, 以能够存储 COMMREQ 所能产生的最长消息或由设备产生的最长回复.

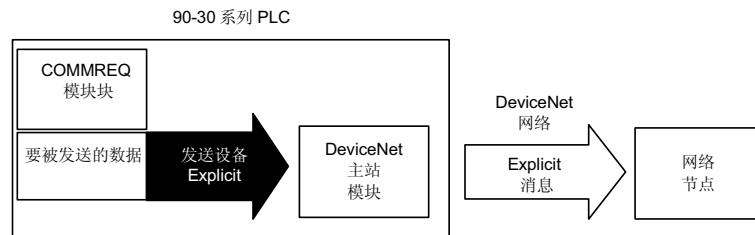
配置可寻址的系列 90-30 DeviceNet 从站设备(网络设置, explicit 消息页)时, 在“explicit 消息大小”字节内填写一个最小值十分重要, 全部数据的长度为数据内存大小 (COMMREQ 7 的字 12) 减去“数据 字节偏移量” (命令 7, 服务数据头部的数据字 6) 中给定的跳过字节数. 在后面的命令 7 的例子中: 网络地址是 4 的设备(S2K 控制器) 为存放 explicit 消息配置的存储区最小为 141 个字节(142 个字节的数据块减去一个跳过的字节).



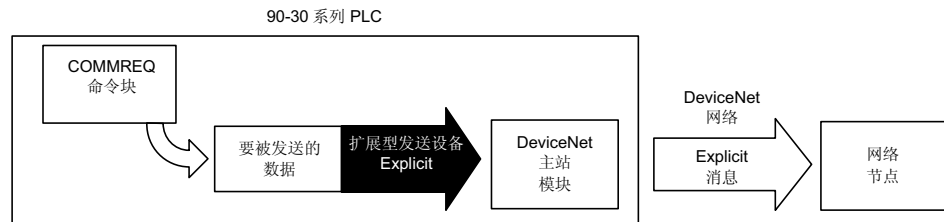
如果设备没有被配置成 explicit messaging 或者配置的字节数不足, COMMREQ 调用失败且 COMMREQ 状态字变为代码 8 .

发送设备 Explicit 和扩展型发送设备 Explicit 之间的区别

发送设备 Explicit 和扩展型发送设备 Explicit 之间的区别在于如何在 PLC 内存中存放被发送的数据。对于发送设备 Explicit, 被发送的数据与 COMMREQ 命令块在同一个内存区域。



对于扩展型发送设备 Explicit, 被发送的数据则放在由 COMMREQ 命令块的指针所指向的一个独立内存区域内。这样就可以存储和发送更多的数据以及将数据和命令内存分开。



发送设备 Explicit OMM E 举例

发送设备 Explicit COMMREQ 命令块中包含被发送的数据, 这些数据存放在 explicit 消息中 (数据在距命令块末尾的一个偏移位置上)。本例中, 要配置 VersaPoint 模拟量模块上的几个通道。应用程序反复使用带不同实例 [其他的通道] 的消息。为保证能正常工作, 即便只调用一个 COMMREQ, 应用程序也需要检查 COMMREQ 状态。因为 VersaPoint DeviceNet NIU (网络接口单元) 可能离线。当发送多个命令给一个设备 (MAC ID) 时, 在执行下一个命令前要检查前一个命令是否成功完成, 这点是十分关键的。

本例中 COMMREQ 执行了以下操作:

- 发送一个 explicit 消息给设备 # 4 (VersaPoint DeviceNet NIU)
- 返回的 COMMREQ 状态字存放在 %R10-%R13
- 设定模拟量输出 1 的范围为 4-20mA。

字#	Dec	(Hex)	描述
1	00012	(000C)	命令的长度: 本 COMMREQ 命令块的长度。 对于发送设备 Explicit (命令 1) 命令的长度为 10 个字加上服务数据的字数. COMMREQ 头部(字 1-6) 不加作命令长度. 注意: 服务数据是以字节为单位, 如果是字为单位, 需要先除以 2 再向上取整. 服务数据长度会根据执行消息的不同而不同; 查询厂商提供的可寻址服务器设备的文档. 本例中: 12 个字 = 服务数据为 3 个字节(转换成 2 个字) + 10 个字 命令 长度.
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字的内存类型(本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: COMMREQ 状态字的起始地址减 1. (本例中是 %R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00001	(0001)	命令代码: 发送设备 Explicit 命令编号(1)
8	00008	(0008)	回复段选: 回复数据的内存类型. (本例中是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复内存偏移量: 回复的内存类型的偏移量减 1. 偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22. 本例中是 %R251.
10	00006	(0006)	回复内存大小: 存放命令的回复的最大长度: 以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22) 对于命令 1, 大小必须是 10 个字节(5 个字)或更多, 或是一个被存放到 COMMREQ 状态字的错误值并且请求被忽略. 注意: 回复的长度取决于使用的服务和被访问的实例. 查询服务器 设备文档. 服务器回复数据需添加 10 个字节(5 个字) 作为回复的头部. 回复内存大小可以大于消息中回复数据的大小, 但不能小于消息中回复数据的大小.
11	00004	(0004)	MAC ID: 发送消息的设备的 Mac ID, 范围(0 - 63). 本例中 VersaPoint 网络接口单元使用 MAC ID #4.
12	00002	(0002)	服务数据大小: 需发送的服务数据字节数. 需要查看 DeviceNet 服务器的文档以决定要发送的信息大小. 例如, 2 字节= 1 属性 字节+ 1 个数据. 注意: 服务代码是 0x10 或 0x0E, 服务数据中的属性字节位于字节 0.
13	00016	(0010)	DeviceNet 服务代码: 参考服务器设备厂商的文档. 本例中, VersaPoint DeviceNet NIU 的服务代码 0x10 (设置单个属性服务) 用于写数据. 而服务代码 0x0E (取得单个属性 服务) 用于读数据.
14	00010	(000A)	类/对象: 被请求的对象类. 参考服务器设备厂商的文档. 本例中, 对象类是 0x0A (模拟量 输出点对象).
15	00001	(0001)	实例: 被请求对象类的特定实例. 参考服务器设备厂商的文档. 本例的实例代表了要设定的那个 VersaPoint 模拟量通道.
16	00001	(0001)	服务数据 字节偏移量: 如果偏移量是 0, 那么服务数据紧跟在该数据字后的内存(字 17, 见下). 这里输入的值是从这个字到服务数据开始之间的字节数. 例如, 如果偏移量是 2, 那么两个字节会被"跳过"即服务数据从字 18 开始.
17	1792	(00)	跳过字节: 这个 字节会被跳过 因为数据字节偏移量是 "1" 字 16. 本例中. 多个字节可能被跳过. 跳过字节中数据被忽略.
		(07)	服务数据 字节 0, 属性: 服务代码 0x10 和 0x0E 消息中用作属性. 属性是单-字节字段, 总位于服务数据的字节 0. "属性"字段对于其他消息服务不起作用. 这个字节实际上是服务数据的起始字节. 因为数据字节偏移量有一个-字节的跳过. 例如属性 7 指 VersaPoint, 模拟量 输出点对象, 范围设定.
18	00003	(0003)	服务数据: 这个数据的起始偏移量取决于服务数据的字节偏移量. 命令 1 的服务数据长度最大为 238 个字节. 例子中 "范围" 3 是厂商代码, 代表 VersaPoint 模拟量 输出 4-20 毫安 设定且数据类型是 USINT (2 个字节). 注意: 清楚使用的数据类型很重要, 这样才能由 COMMREQ 的字 1 和字 12 计算出长度设定

			值.
19 至结束			服务数据: 消息要求的附加服务数据 . 在本例中没有用到.

扩展型发送设备 Explicit OMM E 举例

扩展型发送设备 Explicit COMMREQ 命令块中使用一个指针指向要被发送的数据，这些数据是 explicit 消息中的一部分。程序员可以使用这个功能在每次发送时指向不同的消息而不必再计算命令长度。命令 7 还没有命令 1 的 238-字节服务数据限制，因此可以增加服务数据的最大值设置。本命令只能用于主站模块。

本例中的 COMMREQ 执行以下操作：

- 发送一条 explicit 消息给 Mac ID 4 (一个 GE Fanuc S2K DeviceNet 运动控制器)
- 返回的 COMMREQ 状态字存放到%R10-%R13
- 设置(写)数据(32 DINT)变量数组到 S2K 整型内存中(VI 寄存器)。

	字#	值	描述
COMMREQ Header	1	00007	命令长度: 扩展型发送设备 Explicit 命令的长度。一直是 7 个字。
	2	00000	一直为 0 (非等待模式请求)
	3	00008	状态内存选择: COMMREQ 状态字的内存类型(本例中是%R)。 (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
	4	00009	状态内存偏移量: COMMREQ 状态字的起始地址减 1 (本例中是%R10)
	5		保留
	6		保留
COMMREQ 命令	7	00007	命令代码: 发送 设备 Explicit Extended 命令编号(7)
	8	00008	回复段选: 回复数据的内存类型。(本例中是%R)。(8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
	9	00250	回复内存偏移量: 回复内存类型的偏移量减 1。 本例是 %R251。(偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22)。
	10	00005	回复内存大小: 存放本命令回复所要求的最大长度:(以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22)。在预计的回复大小基础上再增加 10 个字节。 注意: 必须是 10 个字节(5 个字)或更多,或是一个将被报告给 COMMREQ 状态字的一个错误值, 然后这个请求就被忽略了。实际的长度根据要发送什么样的消息决定; 查询目标设备厂商提供的信息。最大值为 2048 个字节。
	11	00008	数据段选: 服务数据的内存类型。(本例中是%R)。(8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
	12	00300	数据内存偏移量: 服务数据存放在指定内存类型的偏移量, 起始地址减 1。(偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22)。本例中, 本例是%R301。
	13	00071	数据内存大小: 要被发送的数据大小, 以所选数据类型的大小为单位(以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22)。 指定的大小必须能放下全部的 explicit 数据块。全部数据块的计算方法为: 服务数据的头部 12 个字节(6 个字) + 跳过 字节(在服务数据的头部字 6 中给出)+ 服务数据。 注意: 知道服务使用的数据类型很重要, 只有知道它才能准确的计算出最小长度。用到属性时, 属性字节在服务数据字节 0 并且需要加到数据长度中。大小需要向上取整。 本例中有 71 个字; 服务数据头部 6 个字 + 1 跳过 字节+ 1 属性字节+ 32 DINT 数据 (64 个字) 服务数据。

扩展型发送设备 Explicit 数据块格式

下面的数据根据数据内存偏移量必须存放在命令指定的 PLC 内存位置上。

One use of 数据 字节偏移量 (见下) 将被“指向”PLC 内存中的一大段数据。下例中数据字节偏移量用于维护 PLC 内存内的数据的字边界位置，甚至我们要求服务数据包含属性值。

	字#	(十六进制)	描述
服务数据的头部	1	(0004)	MAC ID: 发送消息设备的地址(0 - 63).
	2	(0081)	服务数据字节数: 需要参考厂商提供的 DeviceNet 服务器文档来决定.本例中服务数据 0x81 (129 bytes) = 1 个字节属性 + 128 个字节(32 DINT) 数据.
	3	(0010)	DeviceNet 服务代码: 查看厂商提供的服务器设备的文档. 在本例中, 服务是 0x10 (设置单个属性服务), 表示写数据.
	4	(0004)	对象类: 被请求的对象类.参见服务器设备的文档. 本例中, 对象类是 0x04 (S2K 组合对象).
	5	(0300)	实例: 本请求所指的对象类的实例.参见 服务器的文档. 本例中实例十进制数 768 (0300h)指向 S2K 内的 VI001, VI001 是 S2K 中 32 个可写 DINT 类型变量的第一个 .
	6	(0001)	数据字节偏移量: 在这个字和被发送的服务数据起始字之间的字节数. 如果偏移量是 0, 表示服务数据紧跟在这个数据字 (在字 7,见下)之后. 比如,如果偏移量是 2,那么两个字节会被"跳过" 即数据从字 8 开始.
	7	(00)	LSB: 跳过 – 因为字 6 的设置此低字节"跳过".
服务数据		(03)	服务数据 字节 0, 属性 – 服务 0x10 和 0x0E 消息中使用属性.了解特定属性的含义, 需查看目标服务器设备的文档. 因为字 6 "跳过"一个字节所有这个字节是服务数据的起始位置. 在此开始存放不带属性的消息数据. 当然也可能在不同的位置上, 这取决于字 6 中的值.
	8, 9	DINT	服务数据: 也可能位于其他的偏移量上, 这要根据字 6 的值来决定. 使用字 6 允许的偏移量, 在本例中, DINT 数据 需要以字边界对齐.
	10 至末尾	-	服务数据: 本例中 服务数据的末尾在字 71 [6 个头部字+ 1 个 跳过字节+ 1 个 属性 字节+ 64 个数据字].

发送设备 Explicit 及扩展型 回复数据格式

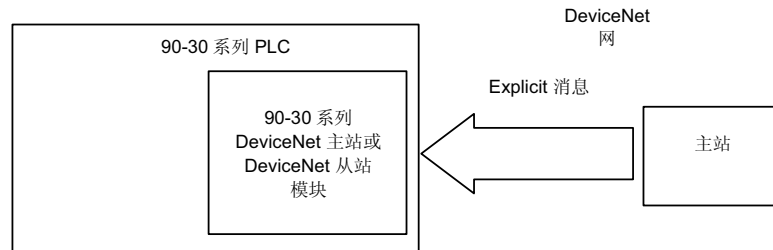
系列 90-30DeviceNet 主站模块回复发送设备 Explicit 或扩展型发送设备 Explicit COMMREQ. 回复数据以下面的格式:

字#	描述			
1	数据块要回复的命令代码 (1 代表 发送设备 Explicit, 7 代表 扩展型发送设备 Explicit)			
2	explicit 消息的状态. 位 0 和 1 都是 0.			
	b10	1 = 为了能够放入共享的内存缓冲区, Explicit 消息回应信息被截断了. 设备 explicit 缓冲区被配置的空间太小.		
	b11	1 = 为了能放入回复内存, Explicit 消息回应消息被截断. COMMREQ 中分配的回复缓冲区太小.		
	bt 1	保留, 应该被忽略.		
3	生产这个回复的设备的 MAC ID .			
4	回复数据消费的字节数. 注意: 如果分配的缓冲区不足, 这个值应该是回复数据的实际大小. 分配回复大小 至少比服务数据多 10 个字节(对应是回复字 1-5).			
5	DeviceNet 服务代码 / 内部结果代码. 小于 0xFF 的值: explicit 消息回复中服务代码的低字节包含了与 explicit 消息相同的服务. 因为消息要答复由 COMMREQ 发出的 explicit 服务, 所以低字节的最高位被置成 1. 例如: GET_ATTRIBUTE_SINGLE 的服务代码是 0x0E DeviceNet 回应将最高位置位: 0x8E SET_ATTRIBUTE_SINGLE 的服务代码是 0x10 回应的最高位置位 0x90 DeviceNet 错误使用服务代码 0x14, 并且由于错误成为了回应, 最高位被置位: 0x94. 例如: GET_ATTRIBUTE_SINGLE: 0x0E DeviceNet 错误回应: 0x94 (主代码和附加代码使用下列的字节)			
	值	错误描述	值	错误描述
	0x00 - 01	保留	0x12	保留
	0x02	执行请求服务对象所需的资源不能用.	0x13	服务没有提供足够的以执行请求的服务
	0x03 - 07	保留	0x14	请求中指定的属性不被支持
	0x08	请求的服务在对象类/实例中没有实现或定义	0x15	服务给出了多于预计的数据
	0x09	检测到无效的的属性数据	0x16	指定的对象在设备中不存在
	0x0A	保留	0x17	保留
	0x0B	服务请求的对象已经处于请求模式或请求状态	0x18	对象的属性数据没有先于请求服务存储起来
	0x0C	在当前的模式/状态下对象不能执行所需服务	0x19	对象的属性数据没有被对象保存起来
	0x0D	保留	0x1A - 1E	DeviceNet 保留
	0x0E	收到一个试图修改一个不能修改的属性的请求	0x1F	厂商特定的错误
	0x0F	允许/权限检查失败	0x20	无效的参数
	0x10	设备的当前模式或状态禁止执行请求的服务	0x21 - CF	保留
	0x11	要被发送的数据的长度比分配的回应缓冲区大	0xD) - FF	厂商特定的对象和类错误
	大于 0xFF 的值是系列 90-30DeviceNet 主站 模块 代码的内部值 (见下).			
	0x0100	Explicit 连接不能被建立		
	0x0101	Explicit 正文的格式不能表示被请求的类. (例如. 类 > 255 且连接正文格式是 8/8 或 8/16)		
	0x0102	Explicit 正文格式不能表示被请求的实例. (例如. 实例 > 255 并且 连接正文的格式是 8/8 或 8/16)		
	0x0103	发送 explicit 消息所需资源不可用		
	0x0104 - FFFF	保留		

6 - 末尾	服务所需的可选数据。这段数据的大小由字 4 指定
--------	--------------------------

命令代码 2, 3 和 8: 读取和响应客户端 *Explicit* 消息

网上的客户端可以发送 DeviceNet explicit 消息给一个系列 90-30 DeviceNet 从站模块或作为 explicit messaging 服务器的系列 90-30 DeviceNet 主站 模块。



不需要 编程的对象

一些客户端 explicit 消息是系列 90-30 DeviceNet 模块自身的服务并不需要 PLC 应用程序介入。模块固件中定义的和为自动回复保留的对象有：

- 标识对象 (0x01)
- 消息路由器对象 (0x02)
- DeviceNet 对象 (0x03)
- 组合对象 (0x04)
- 连接对象 (0x05)
- PLC 数据 对象 (0x64).

你不需要 COMMREQ 命令回复与这些对象关联的服务，由系列 90-30 DeviceNet 模块产生回复。

对于许多应用程序,使用自动消息对象的客户端消息可以达到应用程序目的. 例如, PLC 数据对象 允许像个人电脑或其他 PLC 之类的设备读写安装系列 90-30 DeviceNet 模块的 PLC 内的数据内存阵列. 它可以是发送一组数据给 PLC 的 %R 内存或者是根据需要获得一组状态数据. 使用单一 PLC 数据类型, 仅需少量的编程就可以完成数据的解包或打包, 这些程序可以产生几乎所有的数据结构并且可以被客户端访问. 使用自动应答通常可以简化 PLC 的应用程序. 在设计整个系统时这是一个优先考虑的问题.

使用 OMM E 发送自定义 Explicit Messaging

本章介绍的 COMMREQ 命令可以允许 PLC hosting the 系列 90-30DeviceNet 模块回复“自定义” explicit 消息. 一种利用自定义-explicit messaging 的方法是允许系列 90-30DeviceNet 模块响应已存在的客户端的消息,或模仿部分 DeviceNet 设备模板的响应. 例如系列 90-30DeviceNet 模块可以模仿 emulate 来自 DeviceNet 电机控制中心或位置控制器设备的响应. 唯一的限制是上面提到的保留自动化对象不能用于自定义-explicit 类对象标识. DeviceNet 对象类标识指在 1-65535 之间的一个给定值. 系统仅有六个保留标识,所有有很大的空间留给自定义-explicit 标识使用. 一般地, 使用对象值大于 0x64 来定义自定义-explicit 消息以减少发生 DeviceNet 对象冲突的可能性.

通过使用自定义-explicit messaging, PLC 程序可以实现应用对象并且可以让 DeviceNet 网上的其他客户端设备读写这些对象的参数. 系列 90-30PLC CPU 中的应用程序可以读和响应下面介绍的 DeviceNet 自定义-explicit 消息.

模块如何处理自定义-Explicit 消息

当系列 90-30DeviceNet 服务器 收到一个网络上的自定义-explicit 消息时,它存储消息到一个内部队列中. 在一个给定时间内, 消息队列最多可以有 255 个连接. 不过,这个消息队列可以被模块内的所有客户端或服务器的消息功能块使用. 例如系列 90-30DeviceNet 主站模块与从站的轮询连接是一个客户端连接. 客户端设备发送给系列 90-30DeviceNet 模块 服务器的 explicit 消息会打开一个活动的服务器连接直到客户端消息被回复或超时. 在连接空闲片刻之后(通常是 2-3 秒) 它被释放, 让队列可以被后续消息使用. 如果一个客户端设备在空闲超时之前产生一个一秒的 explicit 消息给系列 90-30DeviceNet 模块, 那么这个连接还可以用于处理新的消息.

另外服务器 状态字 SERA 位至少又一次 PLC 扫描时为逻辑 ON. 服务器状态字通过配置映射到 PLC I/O 而且通过 COMMREQ # 6 可以被使用. 应用程序应监视这个服务器 状态字而不能总是发送命令 6 COMMREQ 去查询自定义-explicit 消息是否到达. PLC 中的应用程序可以监视 SERA 位以判断队列中是否有客户端的 自定义-explicit 消息 . PLC 应用程序必须处理客户端的消息并且在客户等待回应超时前产生一个回复.

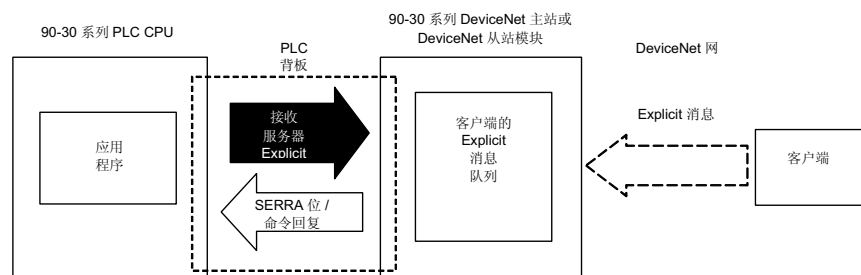
因此, PLC 应用程序响应自定义-explicit 消息首先应该使用 **COMMREQ #2** 读取下一个网上客户端设备发送的自定义-explicit 请求. 使用 **COMMREQ #2 接收服务器 Explicit**.

PLC 应用程序在读取一个 DeviceNet 模块 explicit 消息之后,使用 **COMMREQ #3** 回复自定义-explicit 客户端请求. 对于最多 238 个字节的消息使用 **COMMREQ #3, 发送服务器 Explicit**, 或对于字节数更多的消息使用 **COMMREQ #8, 发送服务器 Explicit Extended**.

为了判断 DeviceNet 模块是否完成发送自定义-explicit 消息的回复, 程序可以使用 **COMMREQ #5, Get Detailed 服务器状态**. 命令 5 的回复数据 表明了模块是否已发送了由前一次发送 服务器 Explicit COMMREQ 调用而产生的 DeviceNet explicit 回复消息.

读取 DeviceNet 模块的 DeviceNet 自定义-Explicit 消息

接收服务器 Explicit COMMREQ 命令 DeviceNet 从站模块或以从站模式工作的 DeviceNet 主站 模块返回一个最后一次收到的 DeviceNet 客户端消息.



系列 90-30 DeviceNet 模块 在回应第一个消息前，可以接收多个网络主站返回的 explicit 请求消息。

命令代码 接收服务器 Explicit OMM E 格式举例

这个例子中的接收服务器 Explicit COMMREQ 使用命令码(2),且设置了 PLC CPU 内的内存类型、起始地址及模块要回复数据的长度. 测试 SERA 位来启动这个命令.

字#	Dec	(Hex)	描述
1	00004	(0004)	命令长度: 命令数据块的长度. 接收服务器 Explicit 命令的长度是 4.
2	00000	(0000)	一直为 0 (非等待模式请求)
3	00008	(0008)	状态段选: COMMREQ 状态字的 内存类型 (本例是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	00009	(0009)	状态内存偏移量: COMMREQ 状态字的起始地址减 1 (本例是 %R10)
5	00000	(0000)	保留
6	00000	(0000)	保留
7	00002	(0002)	命令代码: 接收服务器 Explicit (2)
8	00008	(0008)	回复段选: 回复数据的内存类型. (本例是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	00250	(00FA)	回复内存偏移量: 回复内存类型的偏移量减 1. (偏移量是以字为单位的内存类型有: 8, 10, 12; 偏移量是以字节为单位的内存类型有: 16, 18, 20, 22). 本例是 %R251.
10	00010	(000A)	回复大小: 这个命令需要保存回复的长度的最大值: (以字为单位的内存类型有: 8, 10, 12; 以字节为单位的内存类型有: 16, 18, 20, 22). 注意: 必须是 15 个字节或更多,或是一个返回到状态字的错误然后这个请求将会被忽略.

接收服务器 Explicit 回复数据格式

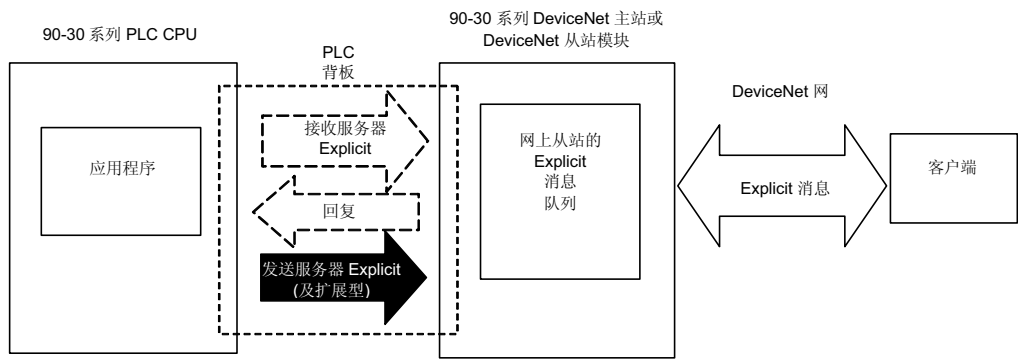
当 DeviceNet 模块收到一个接收服务器 Explicit COMMREQ, 它会回复一个消息, 消息中既包括内部内存中下一个 explicit 请求, 或在处理队列中没有 explicit 请求的标识.

回复包含关于发 explicit 送请求设备的信息, 以及被请求的服务描述. 应用程序必正确地响应请求.

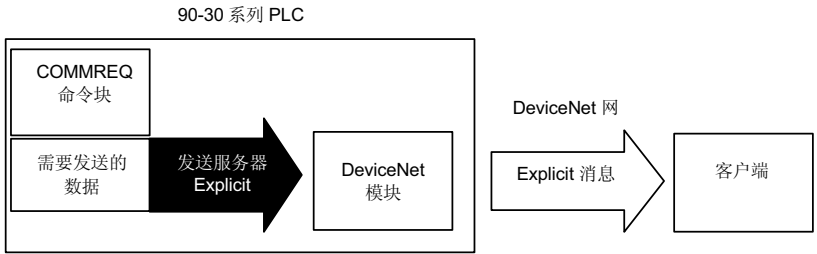
字#	描述	
1	这个数据块正在回复的命令代码. (2)	
2	接收服务器 Explicit 命令的状态. 如果下列位被置位, 则那一个对应的条件为真. 0 表示取得了一个消息.	
	bit 0	1 = 没有 explicit 请求. 模块把剩余的回复数据清零.
	bits 1 - 15	保留.
3	连接 ID: 对应这个请求的连接 ID.	
4	大小: 服务数据 字节数	
5	服务: 被请求的 DeviceNet 服务代码.	
6	对象类: 指向的对象类.	
7	实例: 这个请求指向的特定对象类的实例.	
8 - end	服务数据: 服务所需的可选数据. 这个数据的大小由字 4 给出.	

命令代码 3 和 回复 DeviceNet Explicit 消息

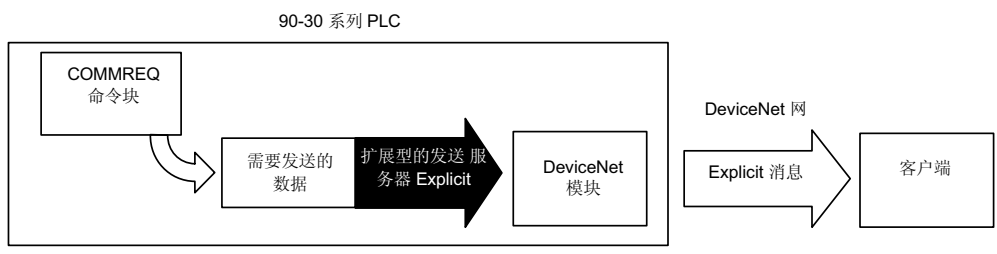
当应用程序取得从系列 90-30 DeviceNet 从站模块或以从站模式工作的系列 90-30 主站模块返回的 DeviceNet explicit 消息,它可以使用 **COMMREQ #3, 发送服务器 Explicit** 回应网络主站. 如果回应的内容超过 238 个 explicit 数据字节,使用 **COMMREQ #8, 扩展型发送服务器 Explicit**.



发送服务器 Explicit 和扩展型发送服务器 Explicit 的区别是如何在 PLC 内存中保存要发送的数据. 对于发送服务器 Explicit, 要发送的数据存放于 COMMREQ 命令块占用的内存区.



对于扩展型发送服务器 Explicit, 要发送的数据 存放在一个由 OMMREQ 命令块指针指向的一个独立内存区域. 这样就可以保存和发送更多的数据.



命令代码3 发送服务器 Explicit OMM E 描述

对于发送服务器 Explicit COMMREQ 没有命令的回复. 检验 COMMREQ 状态字判断命令是否成功时需要注意. 你必须有先于这个命令前一个来自命令 2 的连接 ID.

字#	描述
1	命令长度: 命令的长度. 命令的长度是 5 个字加服务数据字的大小 (注意: 服务数据 是以字节为单位, 除以 2 在向上取整). 本例中服务数据是 5 个字节= 3 个字
2	一直为 0 (非等待模式请求)
3	状态段选: 内存类型 COMMREQ 状态字 (本例是 %R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	状态内存偏移量: COMMREQ 状态字 地址减 1 (本例是 %R10)
5	保留
6	保留
7	命令代码: 发送服务器 Explicit (3)
8	连接 ID: 用于满足请求/响应匹配的需要. 来自用于提供这个客户端的回复命令 2 的 COMMREQ.
9	大小: 服务数据的字节数
10	服务: DeviceNet 服务代码.
11	服务数据 偏移量: 在这个字与需被发送的服务数据的起始位置之间的字节数. 如果偏移量是 0, 服务数据 放入紧跟在这个数据字(字 12, 见下面)后面. 例如, 如果偏移量是 2, 那么“跳过”两个字节即数据从字 13 开始.
12	跳过 (可选): 服务数据 字节 0, 属性:
13 to end	服务数据:

命令代码 扩展型发送服务器 OMM E

字#	描述
1	命令长度: 对于命令 8 一直是 4 个字
2	一直是 0 (非等待模式请求)
3	状态段选: 内存类型 of COMMREQ 状态字 (本例中是%R). (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
4	状态内存偏移量: COMMREQ 状态字 地址减 1 (本例是%R10)
5	保留
6	保留
7	命令代码: 扩展型发送 服务器 Explicit (8)
8	数据段选择: 将被发送数据的内存类型. (8 = R, 10 = AI, 12 = AQ, 16 = I, 18 = Q, 20 = T, 22 = M)
9	数据 内存偏移量: 将被发送数据的内存类型中的偏移量. (偏移量是字的内存类型: 8, 10, 12; 偏移量是字节的内存类型: 16, 18, 20, 22)
10	数据 内存大小: 大小数据 块 to 发送 (大小单位是字的内存类型: 8, 10, 12; 大小单位是字节的内存类型: 16, 18, 20, 22). 必须足够大以放下下面定义的全部 explicit 数据块. 以字节为单位,需要的大小为“服务数据字节大小+ 8 + 服务数据字节偏移量.” 对于单位不是字节的需要单位转换.

发送扩展的服务器 Explicit 数据格式

由命令指定的 PLC 内存区域必须按照下列数据格式填写.

字#	描述
1	连接 ID: 服务器 连接 ID 用于满足请求/响应匹配的需要. 来自用于启动服务的命令 2 到这个请求.
2	服务数据大小: 服务数据字节大小
3	服务代码: DeviceNet 服务代码.
4	服务数据字节偏移量: 从本字至将被发送的服务数据起始位置之间的字节数.如果偏移量是 0,服务数据放置与紧跟这个数据字(在字 7, 见下). 例如,如果偏移量是 2,那么“跳过”两个字节即数据从字 8 开始.
5	跳过(可选):
	服务数据字节 0,属性:
6 至结束	服务数据:

章 7

系列 90-30 模块的 DeviceNet 对象

本章介绍 系列 90-30 DeviceNet 主站模块 IC693DNM200 和从站模块 IC693DNS201 中定义的 DeviceNet 对象，

这些模块在信息交换过程中使用下列类型的对象：

- 标识对象
- 消息路由对象
- DeviceNet 对象
- Assembly 对象
- 连接对象
- PLC 数据对象

标识对象

对象类 0x01

标识对象 包含 模块的标识信息.

这个对象只有一个实例.

类属性

<i>Id</i>	<i>描述</i>	<i>Get</i>	<i>Set</i>	<i>限制</i>
1	修订版	<input type="radio"/>	<input type="radio"/>	
2	Max Instance 最大实例	<input type="radio"/>	<input type="radio"/>	
6	类属性的最大 ID	<input type="radio"/>	<input type="radio"/>	
7	Max ID of 实例属性实例属性的最大 ID	<input type="radio"/>	<input type="radio"/>	
● 表示支持 ○ 表示不支持				

类服务

<i>Service 服务</i>	<i>参数.选项</i>
Get_Attributes_All	<input type="radio"/>
Reset	<input type="radio"/>
Get_Attribute_Single	<input type="radio"/>
Find_Next_Object_Instance	<input type="radio"/>
● 表示支持 ○ 表示不支持	

实例属性

<i>Id</i>	<i>描述</i>	<i>Get</i>	<i>Set</i>	<i>限制</i>
1	Vendor	●	<input type="radio"/>	326
2	Device Type	●	<input type="radio"/>	0x0C
3	Product Code	●	<input type="radio"/>	Master: 2 Slave: 1
4	Revision ¹	●	<input type="radio"/>	
5	Status	●	<input type="radio"/>	
6	Serial Number	●	<input type="radio"/>	
7	Product Name	●	<input type="radio"/>	Master: "90-30 DeviceNet Master" Slave: "90-30 DeviceNet Slave"
8	State	<input type="radio"/>	<input type="radio"/>	
● 表示支持 ○ 表示不支持				
1 DeviceNet 指定主和次版本不为 0, 最低有效次版本号为 1				

实例服务

<i>服务</i>	<i>参数.选项</i>
Reset	●
Get_Attribute_Single	●
● 表示支持 ○ 表示不支持	

消息路由对象

对象类 0x02

消息路由对象 为 explicit messages 指定路由到其他的对象.

这个对象只有一个实例.

类属性

Id	描述	Get	Set	限制
1	Revision	<input type="radio"/>	<input type="radio"/>	
4	Optional Attribute List	<input type="radio"/>	<input type="radio"/>	
5	Optional Service List	<input type="radio"/>	<input type="radio"/>	
6	Max ID of 类属性	<input type="radio"/>	<input type="radio"/>	
7	Max ID of 实例属性	<input type="radio"/>	<input type="radio"/>	
● 表示支持 ○ 表示不支持				

类服务

服务	参数.选项
Get_Attributes_All	<input type="radio"/>
Get_Attribute_Single	<input type="radio"/>
● 表示支持 ○ 表示不支持	

实例属性

Id	描述	Get	Set	限制
1	Object List	<input type="radio"/>	<input type="radio"/>	
2	Maximum connections supported	<input type="radio"/>	<input type="radio"/>	
3	Number of active connections	<input type="radio"/>	<input type="radio"/>	
4	Active connections list	<input type="radio"/>	<input type="radio"/>	
● 表示支持 ○ 表示不支持				

实例服务

服务	参数.选项
Get_Attributes_All	<input type="radio"/>
Get_Attribute_Single	<input type="radio"/>
● 表示支持 ○ 表示不支持	

DeviceNet 对象

对象类 0x03

DeviceNet 对象 包含 DeviceNet 工作的参数.

这个对象只有一个实例.

类属性

Id	描述	Get	Set	限制
1	Revision	●	○	
● 表示支持 ○ 表示不支持				

类服务

服务	参数.选项
Get_Attribute_Single	●
● 表示支持 ○ 表示不支持	

实例属性

Id	描述	Get	Set	限制
1	MAC ID	●	○	
2	Baud Rate	●	○	
3	BOI	○	○	
4	Bus-off counter	○	○	
5	Allocation information	●	○	
6	MAC ID switch changed	○	○	
7	Baud rate switch changed	○	○	
8	MAC ID switch value	○	○	
9	Baud rate switch value	○	○	
● 表示支持 ○ 表示不支持				

实例服务

服务	参数.选项
Get_Attribute_Single	●
Allocate M/S connection set	●
Release M/S connection set	●
● 表示支持 ○ 表示不支持	

组合对象

对象类 0x04

组合对象能够把多个应用程序对象的数据组成一个消息.

这个对象最多可有四个实例.

类属性

Id	描述	Get	Set	限制
1	修订版本	○	○	
2	最大实例	○	○	
● 表示支持 ○ 表示不支持				

类服务

服务	参数.选项
Get_Attributes_All	○
Get_Attribute_Single	○
● 表示支持 ○ 表示不支持	

实例属性

Id	描述	Get	Set	限制
1	列表中的成员数量	○	○	
2	成员列表	○	○	
3	数据	●	●*	
● 表示支持 ○ 表示不支持				
* 只针对非建立的输出连接				

实例服务

服务	参数.选项
Get_Attribute_Single	●
Set_Attribute_Single	●*
● 表示支持 ○ 表示不支持	
* 只针对非建立的输出连接	

连接对象

对象类 0x05

连接对象 处理模块连接. 每个连接对象代表连接的一个端点. 在一个 I/O 消息中, 消息中如何处理数据的全部信息都被包含于该 I/O 消息的连接对象中.

以任何一种连接类型组合里连接对象最多可有四个实例 .

类属性

<i>Id</i>	<i>描述</i>	<i>Get</i>	<i>Set</i>	<i>限制</i>
1	Revision	<input type="radio"/>	<input type="radio"/>	
● 表示支持 ○ 表示不支持				

类服务

<i>Service</i>		<i>参数.选项</i>
Reset	<input type="radio"/>	
Create	<input type="radio"/>	
Delete	<input type="radio"/>	
Get_Attribute_Single	<input type="radio"/>	
Find_Next_Object_Instance	<input type="radio"/>	
● 表示支持 ○ 表示不支持		

实例属性

Id	描述	Get	Set	属性限制/固定值									
				1	2	3	4	5	6	7	8	9	10
1	State	●	○										
2	Instance Type	●	○	0	0	1	1	1	1	1	1	1	1
3	Transport class trigger	●	○	0x23	0x83	0x22	0x83	0x80	0x83	0x03 or 0x13	0x03 or 0x13	0x00 or 0x10	0x00 or 0x10
4	Produced connection ID	●	○										
5	Consumed connection ID	●	○									0xFFFF	0xFFFF
6	Initial comm. characteristics	●	○			0x10	0x01	0xF0	0x02	0x01	0x01	0x0F	0x0F
7	Produced connection size	●	○	0xFFFF	0xFFFF								
8	Consumed connection size	●	○	0xFFFF	0xFFFF								
9	Expected packet rate	●	●	2500	2500	250		250		250	250	250	250
12	Watchdog timeout action(1)	●	●	0x01	0x01 or 0x03	0	0	0	0	0	0	0	0
13	Produced connection path length	●	○	0	0								
14	Produced connection path	●	○	-	-								
15	Consumed connection path length	●	○	0	0						4	0	0
16	Consumed connection path	●	○	-	-						0x20 0x2B 0x24 0x01	-	-
17	Production Inhibit Time (2)	●	●	-	0xFFFF	-	-	-	-	-	-	-	-

● 表示支持 ○ 表示不支持

(1) 仅服务器 explicit 通讯连接能够支持设置属性服务

(2) 仅在客户端 trigger 连接中能够支持设置属性服务

连接类型:

1. Explicit 客户端连接属性限制
2. Explicit 服务器端连接属性限制
3. 轮讯客户端连接属性限制
4. 轮讯服务器连接属性限制
5. 选通客户端连接属性限制
6. 选通服务器连接属性限制
7. COS/循环客户端连接属性限制 (确认)
8. COS/循环服务器连接属性限制 (确认)
9. COS/循环客户端连接属性限制 (未确认)
10. COS/循环服务器连接属性限制 (未确认)

实例服务

服务	参数.选项
Reset	○
Delete	○
Apply_Attributes	○
Get_Attribute_Single	●
Set_Attribute_Single	●
● 表示支持 ○ 表示不支持	

PLC 数据对象

对象类 0x64

实例数: 7

特别服务

服务代码	实现		服务名称
	类	实例	
0x32	是	否	GET_PLC_DATA
0x33	是	否	SET_PLC_DATA

特别实例

The Series 系列 90-30 DeviceNet 模块支持使用以下段选择器访问 PLC 内存. 段选择器置于 Explicit 请求的实例 ID 中.

名称	段选择器 / 实例 ID (dec)	描述
REGISTER_TBL_WORD	8	R 内存
ANALOG_IN_TBL_WORD	10	AI 内存
ANALOG_OUT_TBL_WORD	12	AQ 内存
DISCRETE_IN_BYTE	16	字节模式的 I 内存
DISCRETE_OUT_BYTE	18	字节模式的 Q 内存
DISCRETE_TEMP_BYTE	20	字节模式的 T 内存
DISCRETE_INTERNAL_BYTE	22	字节模式的 M 内存

E D e vice 0x3

服务特别请求数据

名称	数据类型	描述
Offset	UINT2	内存区域中的偏移量. 单位取决于 Seg_Selector. Offset 偏移量从 1 开始, 而不是从 0 开始.
Length	UINT2	读数据的长度. 单位取决于 Seg_Selector

服务特别成功响应数据

如果成功, Explicit Service 字段会被置 0xB2 然后包含以下数据.

名称	数据类型	描述
Data	Array	被请求的数据

E D e vice 0x33

服务特别请求数据

名称	数据类型	描述
Offset	UINT2	内存区域中的偏移量. 单位取决于 Seg_Selector. 偏移量从 1 开始, 而不是从 0 开始.
Length	UINT2	写数据的长度. 单位取决于 Seg_Selector
Data	UNIT[Length]	被写的数据 数据大小等于长度 * 单位大小.

服务特别成功响应数据

如果成功, Explicit Service 字段会被置 0xB3 并且不含数据.

服务特别错误响应数据

错误响应把 0x94 存入 explicit message 的服务代码字段中, 而且一个错误代码和附加代码分别存放在 1 和 2. 以下错误代码可能被返回:

Byte 1	Byte 2	描述
0x08	0xFF	不支持的服务(无效的服务代码)
0x0C	0xFF	对象状态冲突(比如. 内部邮箱系统不可用)
0x11	0xFF	返回数据过长(需传送的数据比内部传送缓冲区要长(255 个字节))
0x13	0xFF	数据不足
0x15	0xFF	数据太多
0x16	0xFF	对象不存在(无效的实例 ID 或类 ID)
0x20	0x01	寄存器基地址使用不正确的形式
0x20	0x02	寄存器的数量使用不正确的形式
0x20	0x03	寄存器基地址超出本型号 CPU 范围
0x20	0x04	寄存器的数量超出 本型号 CPU 或 DN 模块的范围
0x20	0x05	寄存器基地址与数量相加的和超出本 CPU 的范围
0x20	0x06	与 PLC 通讯出错 on GET_PLC_RGS service

0x20	0x07	与 PLC 通讯出错 on SET_PLC_RGS service
------	------	-----------------------------------

附录

A

DeviceNet EDS 文件

这个章介绍 90-30 系列 DeviceNet 主站模块, IC693DNM200 和 90-30 系列 DeviceNet 从站模块, IC693DNS201 使用的电子表单(EDS) 文件. 这些文件仅作为参考使用;随 GE Fanuc 编程软件 CIMPLICITY Machine Edition Logic Developer 一起提供的电子版 EDS 文件是最新版的. EDS 文件(和所有更新文件) 也可以在 GE Fanuc WEB 网站www.gefanuc.com/support/plc/上

DeviceNet 主站模块使用的电子表单文件

\$ DeviceNet Electronic Data Sheet

\$ Copyright (C) 2002 GE Fanuc Automation North America

[File]

```
DescText = "90-30 DeviceNet Master";
CreateDate = 06-05-2002;
CreateTime = 12:59:12;
ModDate = 10-23-2002;
ModTime = 12:34:56;
Revision = 1.04;
```

[Device]

```
VendCode = 326;
VendName = "GE Fanuc Automation NA Inc.";
ProdType = 12;
ProdTypeStr = "Communication Adapter";
ProdCode = 32;
MajRev = 1;
MinRev = 16;
ProdName = "90-30 DeviceNet Master";
Catalog = "IC693DNM200";
```

DeviceNet 从站模块使用的电子表单文件



\$ DeviceNet Electronic Data Sheet

\$ Copyright (C) 2002 GE Fanuc Automation North America

[File]

DescText = "90-30 DeviceNet Slave";

CreateDate = 06-05-2002;

CreateTime = 12:59:12;

ModDate = 10-23-2002;

ModTime = 12:34:56;

Revision = 1.04;

[Device]

VendCode = 326;

VendName = "GE Fanuc Automation NA Inc.";

ProdType = 12;

ProdTypeStr = "Communication Adapter";

ProdCode = 31;

MajRev = 1;

MinRev = 16;

ProdName = "90-30 DeviceNet Slave";

Catalog = "IC693DNS201";