

DeviceNet™ 规范简介

广州周立功单片机发展有限公司

2004年7月14日

目录

一、DeviceNet 概述	1
二、DeviceNet 协议特性	1
三、DeviceNet 对象模型	3
四、DeviceNet 的连接及报文协议	5
五、设备描述与 EDS 文件	7
六、一致性测试	7

一、DeviceNet 概述

DeviceNet™ 规范由 Rockwell 自动化公司开发，并将其作为一个基于 CAN 协议的开放式现场总线标准而公布。最初的 DeviceNet 产品在 1995 年初出现。DeviceNet 协议特别为工厂自动控制而定制，它在美国和亚洲扮演了非常重要的角色。在欧洲，越来越多的系统方案使用 DeviceNet 来实现。

ODVA (Open DeviceNet Vendor Association) 是所有 DeviceNet 产品开发者的组织，它成立于 1995 年，并获得了 Rockwell 自动化公司所有知识产权的转让。该组织按照公司的原则进行运作，并确保所有成员都有同等的发言权。它负责 DeviceNet 标准的制定和更新。此外，ODVA 还致力于 DeviceNet 在全球的推广和市场化。

开发基于 DeviceNet 的产品必须遵循 DeviceNet 规范。DeviceNet 规范分 Volume I、Volume II 两部分。用户可以从 ODVA 协会寻找关于 DeviceNet 开发源代码的信息；基于 CAN-bus 的硬件则可以从 PHILIPS、Intel 等半导体供货商那里获得。

表 1.1 DeviceNet 的特点和功能

网络大小	最多 64 个节点	
网络长度	可选的端 — 端网络长度随网络传输速度变化	
	波特率	距离
	125kbp	500m (1, 640ft)
	250kbp	250m (820ft)
	500kbp	100m (328ft)
数据包	0~8 字节	
总线拓扑结构	线性 (干线/支线)；电源和信号在同一网络电缆中	
总线寻址	带多点传送 (一对多) 的点对点；多主站和主/从；轮询或状态改变 (基于例外)	
系统特性	支持设备的热插拔，无需网络断电	

二、DeviceNet 协议特性

DeviceNet 协议是一个简单、廉价而且高效的协议，适用于最低层的现场总线，例如：过程传感器、执行器、阀组、电动机起动机、条形码读取器、变频驱动器、面板显示器、操作员接口和其他控制单元的网络。可通过 DeviceNet 连接的设备包括从简单的挡光板到复杂的真空泵各种半导体产品。

DeviceNet 也是一种串行通信链接，可以减少昂贵的硬接线。DeviceNet 所提供的直接互连性不仅改善了设备间的通信，而且同时提供了相当重要的设备级诊断功能，这是通过硬接线 I/O 接口很难实现的。

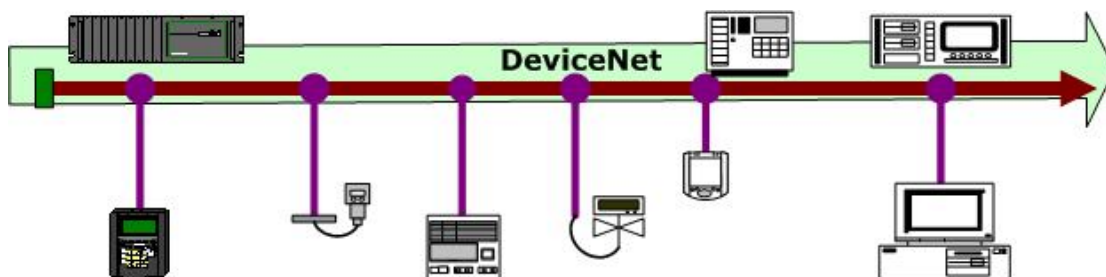


图 2.1 DeviceNet 通讯连接举例

除了提供 ISO 模型的第 7 层（应用层）定义之外，DeviceNet 规范还定义了部分第 1 层（物理收发器）和第 0 层（传输介质）。图 2.2 所示为 DeviceNet 在 ISO 模型中的相关层。对 DeviceNet 节点的物理连接也作了清楚的规定（见 4.5.3.2）。连接器、电缆类型和电缆长度，以及与通信相关的指示器、开关、相关的室内铭牌都作了详细规定。

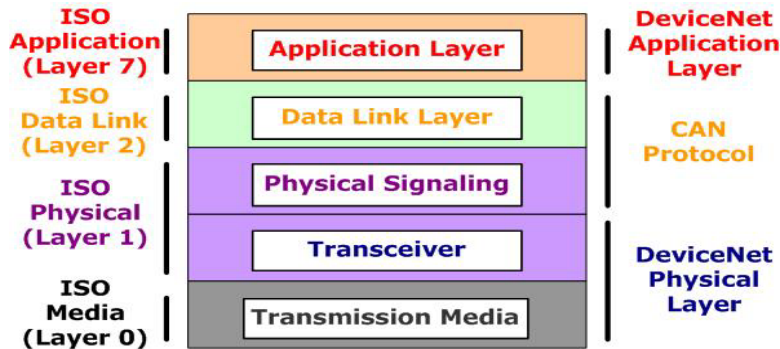


图 2.2 ISO 模型中的 DeviceNet 相关层

DeviceNet 网络最大可以操作 64 个节点，可用的通讯波特率分别为 125kbps、250kbps 和 500kbps 三种。设备可由 DeviceNet 总线供电（最大总电流 8A）或使用独立电源供电。

DeviceNet 网络电缆传送网络通讯信号，并可以给网络设备供电。宽范围的应用导致规定了不同规格的电缆：粗电缆、细电缆和扁平电缆，以能够适用于工业环境。电缆的选用见下表所示。

表 2.1 干线和支线电缆的长度推荐值

数据通讯 速率	干线长度			总支线长度	单根支线 最大长度
	粗电缆	细电缆	扁平电缆		
125kbps	500 m	100 m	420 m	156 m	6 m
250kbps	250 m	100 m	200 m	78 m	6 m
500kbps	100 m	100 m	100 m	39 m	6 m

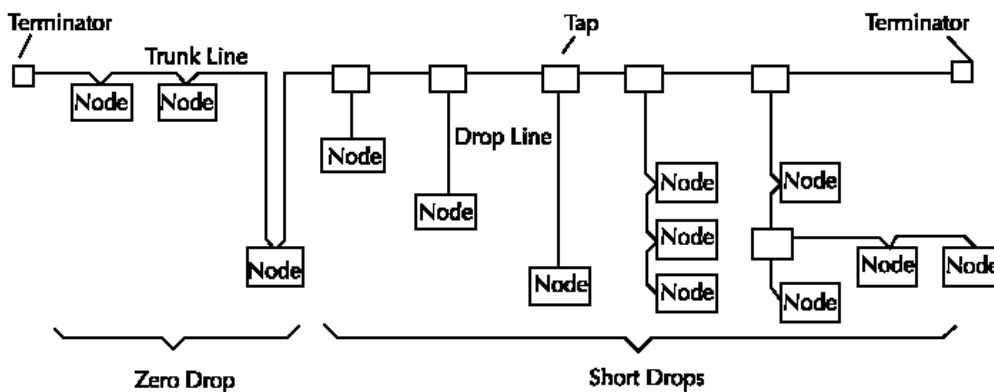


图 2.3 DeviceNet 网络拓扑结构

DeviceNet 设备的物理接口可在系统运行时连接到网络或从网络断开，并具有极性反接保护功能。可通过同一个网络，在处理数据交换的同时对 DeviceNet 设备进行配置和参数设置，这样使复杂系统的试运行和维护变得比较简单；而且现在有许多的高效工具供系统集成者使用，开发变得容易。

DeviceNet 使用“生产者-消费者”通讯模型以及 CAN 协议的基本原理。DeviceNet 发送节点生产网络上的数据，而 DeviceNet 接收节点则消费网络上的数据；两个或多个设备之间的通信总是符合基于连接的通讯模式。

三、DeviceNet 对象模型

DeviceNet 通过抽象的对象模型来描述网络中所有可见的数据和功能。一个 DeviceNet 设备可以定义成为一个对象的集合。这种基于对象的描述提供了一个清晰的设备模型。DeviceNet 设备最重要的对象如图 3.1 所示。

一个对象代表设备内一个部件的抽象描述。对象由它的数据或属性、功能或服务以及它所定义的行为决定。

属性代表数据，设备通过 DeviceNet 生产这些数据。其中可能包括：对象的状态、定时器值、设备序列号或者温度、压力或位置等过程数据。

服务用于调用一个对象的功能或方式。它可对独立属性（如 Get_Attribute_Single/Set_Attribute_Single）进行读或写操作。另外还可创建新的对象实例，或删除现有对象。

对象的行为定义了如何对外部或内部事件进行响应。内部事件可以是定时器的运行事件，外部事件可以是设备要响应的新的过程数据。

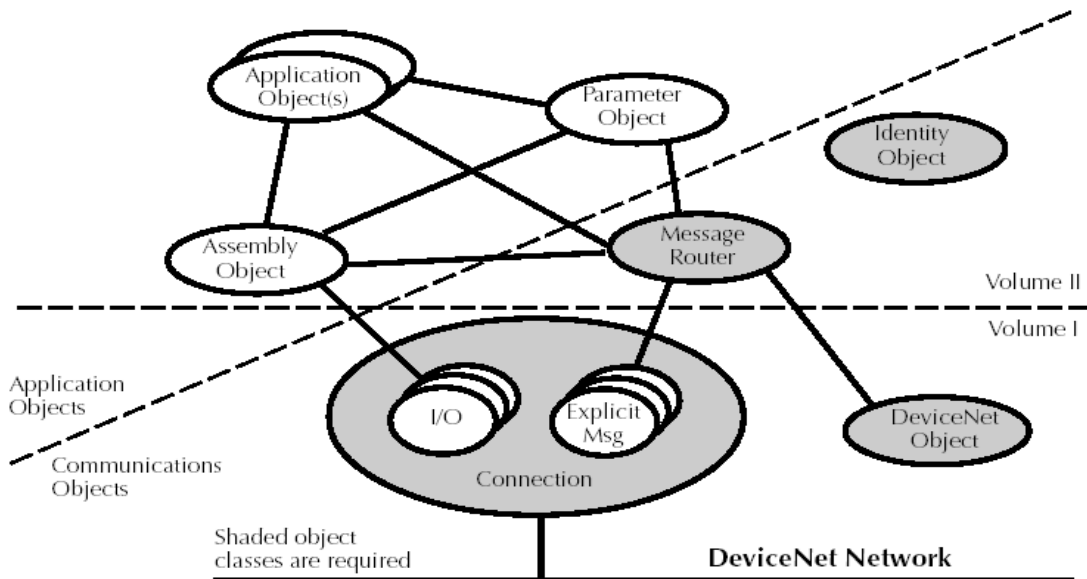


图 3.1 DeviceNet 对象模型

对象分类定义了所有属性、服务和同一类对象行为的描述。如果设备中存在一个对象，我们把它看作是一个分类实例或者对象实例。所能建立的一个分类的实例数目取决于设备的容量。当对象的分类被定义时，对象的功能和行为也随之定义。一个分类的所有实例都支持相同的服务、相同的行为并具有相同的属性。对于每个独立的属性来说，每个实例都有自身的状态和值。

一个对象的数据和服务通过一个分层的寻址概念进行寻址，它包括以下部分：

- 设备地址（MAC ID）
- 分类 ID
- 实例 ID
- 属性 ID
- 服务代码

在 DeviceNet 中通常使用标识符 (ID) 来定义分类、实例、属性和服务。每个 ID 通常用 8 位整数来表示分类、实例和属性, 7 位整数表示服务。这样分类、实例和属性就有多达 256 个可用的 ID, 而服务则有 128 个可用 ID。分类和实例也可以使用 16 位整数, 这样它们的地址空间就扩展为 65536 个不同的 ID。但 16 位模式只被少数设备所支持。

分类、实例、属性和服务的 ID 并不完全供用户自由使用。其中一些保留作将来的规范扩展之用, 还有一些保留给厂商使用。

在 DeviceNet 设备中典型的对象类如表 3.1 所示:

表 3.1 DeviceNet 设备典型对象

对象类编号	对象类名称	规范参考
1	Identity	Vol II, Rel 2.0, page 6-3
2	Message Router	Vol II, Rel 2.0, page 6-17
3	DeviceNet	Vol I, Rel 2.0, page 5-50
4	Assembly	Vol II, Rel 2.0, page 6-23
5	Connection	Vol I, Rel 2.0, page 5-5
6	Parameter	Vol II, Rel 2.0, page 6-86

- Identity (标识对象 Vol II, Rel 2.0, page 6-3)

DeviceNet 设备有且只有一个标识对象类实例 (实例号为 1)。该实例具有以下属性: 供应商 ID、设备类型、产品代码、版本产品名称, 以及检测脉冲周期等。实例必须支持服务 Get_Attribute_Single (服务代码: 0x0e)。

- Message Router (信息路由对象 Vol II, Rel 2.0, page 6-17)

DeviceNet 设备有且只有一个信息路由对象类实例 (实例号为 2)。信息路由对象将显式信息转发到相应的对象, 对外部并不可见。

注意: 信息路由对象的属性 (支持最多连接数、活动连接 ID、系统组件使用的连接数)。

- DeviceNet (DeviceNet 类对象 Vol I, Rel 2.0, page 5-50)

DeviceNet 设备有且只有一个 DeviceNet 对象类实例 (实例号为 3)。DeviceNet 对象具有以下属性: 节点 MAC ID、通讯波特率、BOI (离线中断)、分配信息。实例必须支持服务: Get_Attribute_Single (服务代码: 0x0e)、Set_Attribute_Single (服务代码: 0x10), 对象所提供的分类特殊服务 Allocate_Master/Slave_Connection_Set (服务代码: 0x4B)、Release_Group_2_Identifier_Set ((服务代码: 0x4c))。

- Assembly (组合对象 Vol II, Rel 2.0, page 6-23)

DeviceNet 设备可能具有一个或者多个组合对象类实例 (实例号为 4)。组合对象类实例的主要作用是将不同应用对象的属性 (数据) 组合成为一个单一的属性, 从而可以通过一个报文发送。

- Connection (连接对象 Vol I, Rel 2.0, page 5-5)

DeviceNet 设备至少具有两个连接类实例 (实例号为 5)。每个连接对象表示网络上两个节点之间虚拟连接的一个端点。连接对象分为显式信息连接、I/O 信息连接。显式报文用于属性寻址、属性值以及特定服务; I/O 报文中数据的处理由连接对象 I/O 连接实例决定。

- Parameter (参数对象 Vol II, Rel 2.0, page 6-86)

参数对象 (实例号为 6) 是可选的, 用于具有可配置参数的设备中。每个实例分别代表不同的配置参数。参数对象为配置工具提供了一个标准的途径, 用于访问所有的参数。

- Application (应用对象)

通常除了组合对象和参数对象外，设备中至少有一个应用对象。在卷 II 第 6 章描述了许多已标准化的应用对象。

四、DeviceNet 的连接及报文协议

4.1 概述

DeviceNet 是一个基于连接的通信网络系统。一个 DeviceNet 的连接提供了多个应用之间的路径。当建立连接时，与连接相关的传送会被分配一个连接 ID (CID)。如果连接包含双向交换那么应当分配两个连接 ID 值。连接与连接 ID 见图 4.1 所示。



图4.1 连接及连接ID

4.2 CAN 标识符使用

DeviceNet 建立在标准 CAN2.0A 协议之上，并使用 11 位标准报文标识符，可分成 4 个单独的报文组，见表 4.1 所示。同样，基于扩展 CAN2.0B 协议的 CAN 节点也可以兼容设计成一个 DeviceNet 设备。

表 4.1 报文组的定义

连接 ID = CAN 标识符 (bits 10:0)											标识用途
10	9	8	7	6	5	4	3	2	1	0	
0	报文 ID				源 MAC ID						报文组 1
1	0	MAC ID					报文 ID				报文组 2
1	1	报文 ID			源 MAC ID						报文组 3
1	1	1	1	1	报文 ID						报文组 4
1	1	1	1	1	1	1	x	x	x	x	无效 CAN 标识符

在 DeviceNet 中，CAN 标识符被称为连接 ID。它包含报文组 ID、该组中的报文 ID、设备 MAC ID。源和目标地址都可作为 MAC ID。定义取决于报文组和报文 ID。系统中报文的含义由报文 ID 确定。

4 个报文组分别有以下用途：

- 报文组 1

分配了 1024 个 CAN 标识符 (000H~3FFH)，占有所有可用标识符的一半。该组中每个设备最多可拥有 16 个不同的报文。该组报文的优先级主要由报文 ID (报文的含义) 决定。如果 2 个设备同时发送报文，报文 ID 号较小的设备总是先发送。以这种方式可以相对容易地建立一个 16 个优先级的系统。报文组 1 通常用于 I/O 报文交换应用数据。

- 报文组 2

分配了 512 个标识符 (400H~5FFH)。该组的大多数报文 ID 可选择定义为“预定义主/从连接集”。其中 1 个报文 ID 定义为网络管理。优先级主要由设备地址 (MAC ID) 决定，其次由报文 ID 决定。如果要考虑各位的具体位置，那么带 8 位屏蔽的 CAN 控制器可以根据 MAC ID 滤除自身的报文组 2 报文。

- 报文组 3

分配了 448 个标识符（600H~7BFH），具有与报文组 1 相似的结构。与报文组 1 不同的是，它主要交换低优先级的过程数据。此外，该组的主要用途是建立动态的显式连接。每个设备可有 7 个不同的报文，其中 2 个报文保留作未连接报文管理器端口（UCMMPort）。

- 报文组 4

分配了 48 个 CAN 标识符（7C0H~7EFH），不包含任何设备地址，只有报文 ID。该组的报文只用于网络管理。通常分配 4 个报文 ID 用于“离线连接集”。

其它 16 个 CAN 标识符（7F0H~7FFH）在 DeviceNet 中被禁止。

4.3 DeviceNet 信息协议

DeviceNet 定义了两种类型的报文：显式信息报文、I/O 信息报文。

I/O 信息报文用于在 DeviceNet 网络中传输应用和过程数据。相关的 I/O 数据总是从一个生产应用传输到多个消费应用。I/O 报文通常使用高优先级的报文标识符，连接标识符提供了 I/O 报文的相关信息。I/O 报文传送通过 I/O 信息连接对象来实现。在 I/O 报文被传输之前，I/O 信息连接对象必须已经建立。

I/O 信息报文格式的最重要的特性是完全利用了 CAN 数据场来传输过程数据。连接的端点通过 CAN 报文标识符来识别过程数据的重要性。每个 I/O 报文使用 1 个 CAN 标识符。

显式信息报文用于 DeviceNet 网络中两个设备之间的一般性数据交换。显式报文通常使用低优先级的报文标识符。显式报文为点对点传送，采用典型的请求/响应通讯模式，通常用于设备配置、故障诊断。显式报文传送通过显式信息连接对象来实现，在设备中建立显式信息连接对象。显式报文请求指明了对象、实例和属性，以及所要调用的特定分类服务，并由报文路由对象传递到相应的对象。

显式信息报文格式最重要的特性是 CAN 标识符场的任何一部分都不用于显式报文传输协议。所有协议都包含在 CAN 数据场当中。CAN 标识符场用作连接 ID。设备之间的每个显式连接通道需要 2 个 CAN 标识符，一个用于请求报文，另一个用于响应报文。标识符在连接建立时确定。

4.4 DeviceNet 连接的建立

DeviceNet 是一个基于连接的网络系统。只有当对象之间已建立一个连接时，才能通过网络进行报文传送。DeviceNet 规定了两种类型连接：

- I/O 连接在一个生产应用及一个或多个消费应用之间提供了专用的，具有特殊用途的通讯路径。特定的应用和过程数据通过这些路径传输；
- 显式信息连接在两个设备之间提供了一个通用的、多用途的通讯路径。显式信息连接提供典型的面向请求/响应的网络通讯方式。

DeviceNet 中的报文总是以基于连接的方式进行交换，因此，在进行通讯之前，首先必须建立连接对象。DeviceNet 节点在开机后能够立即寻址的唯一端口是“非连接信息管理器端口”（UCMM 端口）和预定义主/从连接组的“Group2 非连接显式请求端口”。当通过 UCMM 端口或者 Group2 非连接显式请求端口建立一个显式报文连接后，这个连接可用于从一个节点向其它节点传送信息，或建立 I/O 信息连接。一旦建立了 I/O 信息连接，就可以在网络设备之间传送 I/O 数据。

通过 UCMM 端口可以动态的建立显式信息连接。一个支持预定义主/从连接组，并且具有 UCMM 功能的设备称为 Group 2 服务器。一个 Group 2 服务器可被一个或多个客户机通过一个或多个连接进行寻址。

预定义主/从连接组用于简单而快速地建立一个连接。当使用预定义的主/从连接组时，客户机（主站）和服务器（从站）之间只允许存在一个显式连接。由于在预定义主/从连接组定义内已省略了创建和配置应用与应用之间连接的许多步骤，可以使用较少的网络和设备资源来实现 DeviceNet 通讯。

不具有 UCMM 功能，只支持预定义主/从连接组的从设备，被称为 DeviceNet 中的仅限 Group 2 服务器。只有分配它的主站才可以寻址仅限 Group 2 的服务器。仅限 Group 2 的设备能够接收的所有报文都在报文组 2 中被定义。支持预定义主/从连接组对设备制造商来说代表了一个简单实现的方案。

绝大多数现有的 DeviceNet 设备都是基于预定义的主/从连接组，因为这在终端设备上实现起来比较简单。

五、设备描述与 EDS 文件

为了实现同类设备的互用性并促进其互换性，同类设备间必须具备某种一致性，即每种设备类型必须有一个标准的模型。

设备描述（Device Profiles）通过定义标准的设备模型，促进不同厂商同类设备的互操作性，并促进其互换性。ODVA 已经规定了一些工业自动化中常用产品的设备描述。例如，通用 I/O（离散或模拟）、驱动器、位置控制器等等。

在 DeviceNet 规范中设备描述分为 3 个部分。

- 设备类型的对象模型

对象模型定义了设备中所必需和可选的对象分类。对象模型还指定了实现的对象实例的个数，这些对象如何影响设备的行为，及其与这些对象的接口。

- 设备类型的 I/O 数据格式

在设备描述中指定了 I/O 数据的格式。通常也包括组合对象的定义，组合对象属性包括了特定的数据的映射。

- 配置数据和访问该数据的公共接口

描述了配置数据以及数据的公共接口实现。通常包含在电子数据文档（EDS）中，EDS 包含在设备的用户文件中。

DeviceNet 规范规定了电子数据文档（EDS）的格式，EDS 文件提供访问和改变设备可配置参数的所有必要信息。当使用电子数据文档（EDS）时，供货商可以将产品的特殊信息提供给其它供货商。这样可以具有友好的用户配置工具，可以很容易的更新，无需经常修正配置软件工具。

六、一致性测试

ODVA 定义了 DeviceNet 设备和系统的测试和批准程序。会员厂商有机会将它们的设备交给当前 3 个独立的 DeviceNet 兼容性测试中心之一进行一致性检测。所有 DeviceNet 设备只作两个关键性测试：互操作性和互换性。

互操作性表示所有厂商的 DeviceNet 设备都可在网络上互相操作。互换性比其更进一步，可以用相同类型的设备（即它们符合相同的设备描述）在逻辑上互相置换，不管这些设备是由哪个厂商制造的。

一致性测试可以分成以下 3 个部分：

- 软件测试对 DeviceNet 协议的功能进行验证。在测试时，根据设备复杂性的不同，可传输多达数千个报文。
- 硬件测试检测物理层的兼容性。该测试检测规范的所有要求，例如断线保护、过压、接地和绝缘、CAN 收发器等等。该测试对于不符合 DeviceNet 规范的设备可能是破坏性的。
- 系统互用性测试，可以验证在一个多达 64 个节点和众多不同厂商扫描仪的网络中设备的功能。

一致性测试软件可直接从 ODVA 获得。它是基于 Windows 的工具，运行在不同供应商的几个 PC-CAN 接口上。厂商在进行正式的 ODVA 测试之前可以对其设备进行测试。至现在为止，硬件测试和系统互用性测试只能由 3 个独立测试中心完成。

如果设备通过了测试，那么可以说它已是以通过 DeviceNet 一致性测试，并加以标记。许多 DeviceNet 用户现在都要求有该标识。通过一致性测试的设备在市场上会有显著的优势。

ZLGMCU

2004-07-14