DeviceNet
ODVA

TECHNOLOGY OVERVIEW SERIES

CIP on CAN Technology

PUB00026R5 APRIL 2021

DeviceNet® – CIP on CAN Technology

DeviceNet® has been solving manufacturing automation applications since the mid-1990's, and today boasts an installed base numbering in the millions of nodes. DeviceNet is a member of a family of networks that implements the Common Industrial Protocol (CIP™) at its upper layers. CIP encompasses a comprehensive suite of messages and services for a variety of manufacturing automation applications, including control, safety, security, energy, synchronization, motion, configuration, diagnostics and information. As a truly media-independent protocol that is supported by hundreds of vendors around the world, CIP provides users with a unified communication architecture throughout the manufacturing enterprise.

With media independence comes the ability to choose the CIP Network best suited for each application. One of these possible choices is DeviceNet, which adapts CIP to CAN Technology. Why adapt CIP to CAN? CAN is the same network technology used in automotive vehicles for communication between smart devices and has a total installed base numbering in the billions of node. By leveraging the economies of scale in this proven commercial technology, DeviceNet provides users with the ability to distribute and manage simple devices throughout their architecture in a cost-effective manner.

DeviceNet offers several unique advantages for manufacturing automation applications:

- Comprehensive producer-consumer services let you simultaneously control, configure and collect data from intelligent devices over a single network;
- Support for up to 64 nodes and baud rates up to 500 kilobits per second (kbps).
- Robust physical layer, designed for high noise and other challenging environments, provides robust signal transmission while providing the user with a flexible network architecture offering a range of data rates — 125, 250 and 500 kbps — and trunk-line distances up to 500 meters (125 kbps and thick cable);
- Power (24 Vdc, 8 Amps) and signal on the same wire with the ability to remove and replace nodes under power and power taps at any point on the network; and
- Rugged installation options include round cable that allows for flexible cabling topologies, including daisy-chain and trunk-line, with range of connector options available, including screw-terminal and hard-wired (IP20) and mini and micro-style plugs (IP67), or flat cable with flat trunk connectors (IP67).

Here's a more in-depth look at the technology behind every DeviceNet-compliant product.

What is DeviceNet?

DeviceNet, like other CIP Networks, follows the Open Systems Interconnection (OSI) model, which defines a framework for implementing network protocols in seven layers: physical, data link, network, transport, session, presentation and application. Networks that follow this model define a complete suite of network functionality from the physical implementation through the application or user interface layer. As with all CIP Networks, DeviceNet implements CIP at the Session layer and above while adapting CIP to the specific DeviceNet technology at the Transport layer and below. This network architecture is shown in Figure 1.

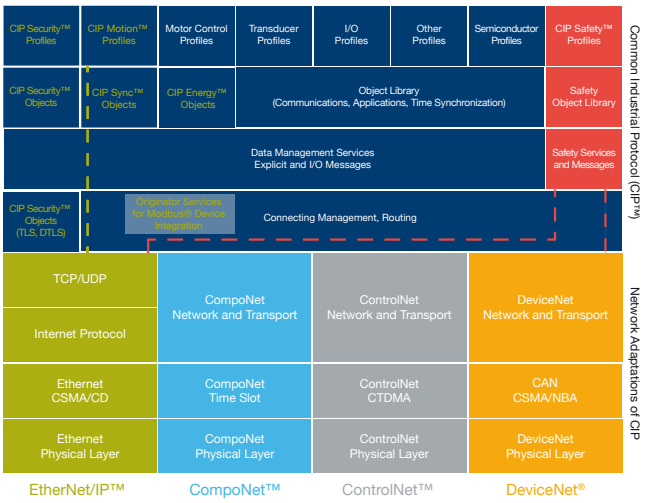

Figure 1: DeviceNet as Part of the CIP OSI Model

The DeviceNet network infrastructure is passive, making node functionality independent of physical location and the network itself inherently tolerant to individual lost node connections. Although the network infrastructure is passive, the network cable can convey device power over the same cable with communication messages. This feature is extremely valuable for devices with small physical size and power requirements, such as photo cells, where DeviceNet simplifies the number of required system components and connections.

To further decrease complexity, DeviceNet systems require only a single point of connection for both configuration and control. This is because DeviceNet supports both I/O (or implicit) messages—those that typically contain time-critical control data—and explicit messages—those in which the data field carries both protocol information and specific service requests. And, as a producer-consumer network that supports multiple communication hierarchies and message prioritization, DeviceNet provides more efficient use of bandwidth than a device network based on a source-destination model. DeviceNet systems can be configured to operate either in a controller/device or distributed control architecture using peer-to-peer communication.

The Physical Layer

DeviceNet incorporates a trunkline-dropline topology that allows the use of separate twisted pair buses for both signal and power distribution. The possible variants of this topology are shown in Figure 2.

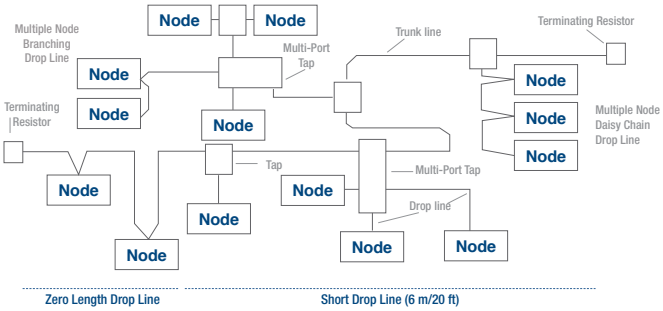

Figure 2: DeviceNet Topology Options

Nodes can be removed or inserted with the network on, helping to reduce production downtime. Power taps can be added at any point on the network, enabling the use of redundant power supplies. The trunkline current rating is 8 Amps maximum, depending on cable type.

DeviceNet supports three possible data rates, and the user may choose from several cable options. In general, these cables can be used as trunkline or as dropline, but end-to-end network length will vary depending on the cable type selected and the date rate used, as shown in Table 1.

Cable Type	125 kbps	250 kbps	500 kbps
Thick Round Cable	500 m (1,640 ft)	250 m (820 ft)	100 m (328 ft)
Thin Round Cable	100 m (328 ft)	100 m (328 ft)	100 m (328 ft)
Flat Cable	420 m (1,378 ft)	200 m (656 ft)	75 m (246 ft)
Maximum Drop Length	6 m (20 ft)	6 m (20 ft)	6 m (20 ft)
Cumulative Drop Length	156 m (512 ft)	78 m (256 ft)	39 m (128 ft)

Table 1: DeviceNet end-to-end network distance as a function of data rate and cable type

DeviceNet supports devices with physical layer implementations that are isolated or non-isolated. However, since the DeviceNet physical layer must be optically isolated from the rest of the device, externally powered devices (e.g., AC drive starters and solenoid valves) can share the same bus cable, helping to save space and reduce wiring complexity.

DeviceNet provides a choice of screw-terminal or pluggable connector types, as shown in figures 3 through 5.

Figure 3: Screw-terminal & hard-wired connectors (IP20)

Figure 4: Mini & Micro-style Pluggable connectors (IP67)

Figure 5: Flat cable with flat trunk connectors (IP67)

The Data Link Layer

DeviceNet uses standard, unmodified CAN for its Data Link layer. The relatively minimal overhead required by CAN at the Data Link layer makes DeviceNet efficient in message handling. Minimal network bandwidth is used to package and transmit CIP messages over DeviceNet, and minimal processor overhead is required by a device to parse such messages.

Although the CAN specification defines several types of message frames (e.g., data, remote, overload and error), DeviceNet uses primarily only the data frame. The format for the CAN data frame is shown in Figure 6.

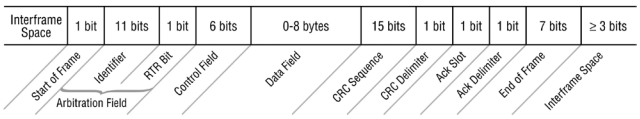

Figure 6: CAN Data Frame Format

The CAN specification defines two logical bus states called “dominant” (logic 0) and “recessive” (logic 1). Any transmitter can drive the bus to a dominant state, while the bus can be in the recessive state only when no transmitter is in the dominant state. This is an important element of the bus arbitration scheme employed by CAN. In addition, CAN is a “carrier sense” network, meaning that nodes will only attempt to transmit a message when no other nodes are transmitting. Carrier sense networks provide inherent peer-to-peer connection capabilities. CAN also utilizes a non-destructive, bit-wise arbitration mechanism to resolve the potential conflict between multiple nodes accessing the bus simultaneously, and does so with no loss of data or bandwidth.

When a “Start of Frame” bit is transmitted, all receivers on a CAN network synchronize to the transition from the recessive to the dominant state. The CAN Identifier and the Remote Transmission Request bit (RTR2) together form the arbitration field, which is used to facilitate media access priority. When a device transmits, it also monitors each bit it sends by simultaneously receiving each bit transmitted in order to validate the data transmitted and to enable immediate detection of simultaneous transmission. If a node transmitting a recessive bit receives a dominant bit while sending the arbitration field, it stops transmitting. The winner of arbitration

between all nodes transmitting simultaneously is the one with the lowest-numbered 11-bit identifier.

The CAN Control Field contains six bits. The content of two bits is fixed, while the other four are used for a "length field," which specifies the length of the forthcoming Data Field from zero to eight bytes.

The CAN Data Frame is followed by the Cyclic Redundancy Check (CRC) field to detect frame errors as well as several frame formatting delimiters. By employing several types of error detection and fault confinement methods—including CRC and automatic retries—that are mostly transparent to the network and that prevent a faulty node from disrupting the network, CAN provides highly robust error checking and fault confinement capabilities.

The Network and Transport Layers

DeviceNet is a connection-based network, meaning that a relationship (i.e., connection) with a device must first be established in order to exchange data with that device. Connections are established via either an Unconnected Message Manager (UCMM) or a Group 2 Unconnected Port. In addition, DeviceNet supports two types of messages, Explicit and Implicit (often referred to as I/O Messages). Explicit messages are used for request/response-oriented transactions typically associated with configuration or data collection that are not time critical, whereas Implicit messages are used to communicate real-time I/O data.

Devices may be clients, servers or both, and, in turn, clients and servers may be producers, consumers or both. A typical client device's connections produce requests and consume responses. A typical server device's connections consume requests and produce responses. DeviceNet allows several variations on this model. For example, some client and/or server connections that may only consume messages are the destinations for cyclic or change-of-state messages. Similarly, some client and/or server connections may only produce messages, and these connections are the sources for cyclic or change-of-state messages. The use of cyclic and change-of-state connections can substantially reduce network bandwidth requirements.

Whenever communicating with a device, an Explicit Messaging Connection is established first via the UCMM or Group 2 Unconnected port. This explicit connection can then be used to move information from one node to the other, or to establish an Implicit (e.g., I/O) connection. Once an I/O connection has been established, the 11-bit CAN identifier field serves as the unique identifier of this data. The remainder of the frame contains no protocol data, only application data. When the application requires more than eight bytes of data in a message, it is utilizes the DeviceNet fragmentation protocol to move the data in several successive messages or "fragments."

In order to take full advantage of DeviceNet's producer-consumer capabilities, the uniqueness of each connection ID is strictly controlled. To achieve this goal, DeviceNet uses the 11-bit CAN identifier to define the connection ID and divides the 11-bit CAN identifier into four groups. The first three defined groups contain two fields: one 6-bit field for Media Access Code Identifier (MAC ID) and the other for Message ID. The combined fields define the connection ID. By design, nodes in a DeviceNet system are responsible for managing their own identifiers. These identifiers are distributed throughout the entire range of message priorities that are available to each node, regardless of their MAC ID. Through a duplicate MAC ID algorithm, the uniqueness of CAN identifiers is guaranteed without the need for a central tool or record for each network.

<table>
<tr><th colspan="11">IDENTIFIER BITS</th><th>HEX RANGE</th><th>IDENTITY USAGE</th></tr>
<tr><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td><td></td></tr>
<tr><td>0</td><td colspan="3">Group 1 Message ID</td><td colspan="4">Source MAC ID</td><td colspan="3"></td><td>000 – 3ff</td><td>Message Group 1</td></tr>
<tr><td>1</td><td>0</td><td colspan="5">MAC ID</td><td colspan="3">Group 2 Message ID</td><td>400 – 5ff</td><td>Message Group 2</td></tr>
<tr><td>1</td><td>1</td><td colspan="3">Group 3 Message ID</td><td colspan="4">Source MAC ID</td><td colspan="3"></td><td>600 – 7bf</td><td>Message Group 3</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td colspan="6">Group 4 Message ID (0-2f)</td><td colspan="3"></td><td>7c0 – 7ef</td><td>Message Group 4</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>X</td><td>X</td><td>X</td><td>X</td><td>7f0 – 7ff</td><td>Invalid CAN Identifiers</td></tr>
<tr><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td><td></td></tr>
</table>

Figure 7: DeviceNet allocations within the 11-bit CAN Identifier Field.

Because DeviceNet uses a device address inside the CAN identifier field, it incorporates automatically a mechanism for detecting nodes with duplicate addresses. This is important because it is more efficient to prevent duplicate addresses than to locate them after they occur. Another key benefit to nodes managing their identifiers is that a user can add and delete nodes, add additional peer-to-peer messages among existing nodes at any time, or both, without having to know the existing setup. That is, no centralized record must be located or reconstructed. Since nodes know which IDs are already in use, a tool must simply request that an I/O connection be added between the two devices, specifying priority level, the data path (class, instance, attribute) and the production trigger (cyclic, poll or change-of-state).

The Upper Layers

DeviceNet uses the Common Industrial Protocol (CIP), a strictly object-oriented protocol, at the upper layers. Each CIP object has attributes (data), services (commands) and behaviors (reactions to events). CIP's producer-consumer communication model provides more efficient use of network resources than a source-destination model by allowing the exchange of application information between a sending device (e.g., the producer) and many receiving devices (e.g., the consumers) without requiring data to be transmitted multiple times by a single source to multiple destinations. In producer-consumer networks, a message is identified by its connection ID, not its destination address (as is the case with source-destination networks). CIP's message structure makes it possible for multiple nodes to consume data produced by a single source based solely on the connection ID to which the message refers. Thus, the producer-consumer model provides a clear advantage for users of CIP Networks by making efficient use of network resources in the following ways:

- If a node wants to receive data, it only needs to ask for it once to consume the data each time it is produced.
- If a second (third, fourth, etc.) node wants the same data, all it needs to know is the connection ID to receive the same data simultaneously with all other nodes.

CIP also includes "device types" for which there are "device profiles." For a given device type, the device profile will specify the set of CIP objects that must be implemented, configuration options and I/O data formats. This consistency in object implementation for a given device type provides another clear advantage for users of CIP Networks by promoting a common application interface for a given device type and interoperability in networks comprised of devices from multiple vendors. For applications where unique functionality is required, it is also possible for a DeviceNet vendor to define additional vendor-specific objects in a DeviceNet-compliant product in order to support the functional requirements of particular applications.

Seamless bridging and routing is perhaps the most significant advantage for users of CIP Networks for it is this mechanism that most protects the user's investment for the future. The ability to originate a message on one CIP Network, such as DeviceNet, and then pass it to another CIP Network, such as EtherNet/IP, with no presentation at the Application Layer, means that users can incorporate incremental application improvements to existing installations and/or integrate systems with diagnostic, prognostic and/or IT applications. Seamless bridging and routing between homogeneous and heterogeneous CIP Networks is enabled by a set of objects that defines mechanisms for a device to use when forwarding the contents of a message produced on one network port to another. This mechanism does not alter the contents of a message during the routing process. When using this mechanism, the user's only responsibility is to describe the path that a given message must follow. CIP ensures that the message is handled correctly, independent of the CIP Networks involved.

The Predefined Controller/ Device Connection Set

DeviceNet provides for an alternate, simplified communication scheme based on a controller/device relationship. Called the "Predefined Controller/Device Connection Set," this scheme simplifies both the packaging and the movement of data contained in the I/O messages most often used in control applications.

Because one of DeviceNet's key unique advantages is "power" on the network, many DeviceNet-compliant sensors and actuators are designed to perform a predetermined function on power-up. In this case, both the type and amount of data the device will produce and/or consume is also known on power-up. The Predefined Controller/Device Connection Set provides connection objects that are almost entirely configured at the time the device powers-up.

After powering up the network, the only remaining step necessary to begin the flow of data is for a "controller" device to claim ownership of this predefined connection set within its "device(s)." Devices can produce data using one or more of the message types described in Table 2. The message type used is determined based on how the device is configured and the requirements of the application.

Type	Description of Operation
Polled	A device configured for polled I/O will receive "output" data from the controller in a sequential order that is defined by the controller's scan list and will transmit its "input" data in response to the controller's poll. The controller's polling rate is determined by: the number of nodes in the scan list; the DeviceNet baud rate; the size of messages produced by the controller and each node in its scan list; and the internal timing of the controller. For a given system configuration, this messaging method will provide deterministic behavior. Polled I/O "output" data is unicast and "input" data is multicast.
Cyclic	A device configured to produce a cyclic I/O message will produce its data at a precisely defined interval. This type of I/O messaging allows each user to configure the system to produce data at a rate appropriate for the application. Depending on the application, cyclic I/O messaging can reduce the amount of traffic on the wire and more efficiently use the available bandwidth.
Change-of-state	A device configured to produce change-of-state (COS) messages will produce data whenever it changes, or at a base "heartbeat" rate. This adjustable heartbeat rate provides a way for the consuming device to know that the producer is still alive and active. DeviceNet also defines a user-configurable Production Inhibit Time that limits how often COS messages are produced to prevent nodes from "flooding" the bandwidth. Users can adjust these parameters to provide optimum bandwidth utilization in a given application scenario.

Table 2: Device I/O message types in the DeviceNet predefined controller/device connection set

Management of the DeviceNet Technology

DeviceNet is managed by ODVA, an international association of the world's leading automation companies. ODVA's DeviceNet management responsibilities include:

- Publishing The DeviceNet Specification;
- Overseeing the process to incorporate new enhancements to the DeviceNet Specification;
- Licensing the DeviceNet Technology to companies desiring to make and/or sell DeviceNet-compliant products;
- Promoting industry awareness of DeviceNet and its benefits; and
- Helping to ensure compliance of DeviceNet products with the specification through conformance testing and conformity reporting.

For more information about DeviceNet, CIP or ODVA, visit ODVA at **www.odva.org**.

About ODVA

Founded in 1995, ODVA is a global association whose members comprise the world's leading automation companies. ODVA's mission is to advance open, interoperable information and communication technologies in industrial automation. ODVA recognizes its media independent network protocol, the Common Industrial Protocol or "CIP" – and the network adaptations of CIP – EtherNet/IP, DeviceNet, CompoNet and ControlNet – as its core technology and the primary common interest of its membership. For future interoperability of production systems and the integration of the production systems with other systems, ODVA embraces the adoption of commercial-off-the-shelf (COTS) and standard, unmodified Internet and Ethernet technologies as a guiding principle wherever possible. This principle is exemplified by EtherNet/IP – the world's number one industrial Ethernet network. For more information about ODVA, visit odva.org.

ODVA
Ann Arbor, Michigan, USA
TEL: +1 734-975-8840
FAX: +1 734-922-0027
WEB: www.odva.org

PUB00026R5