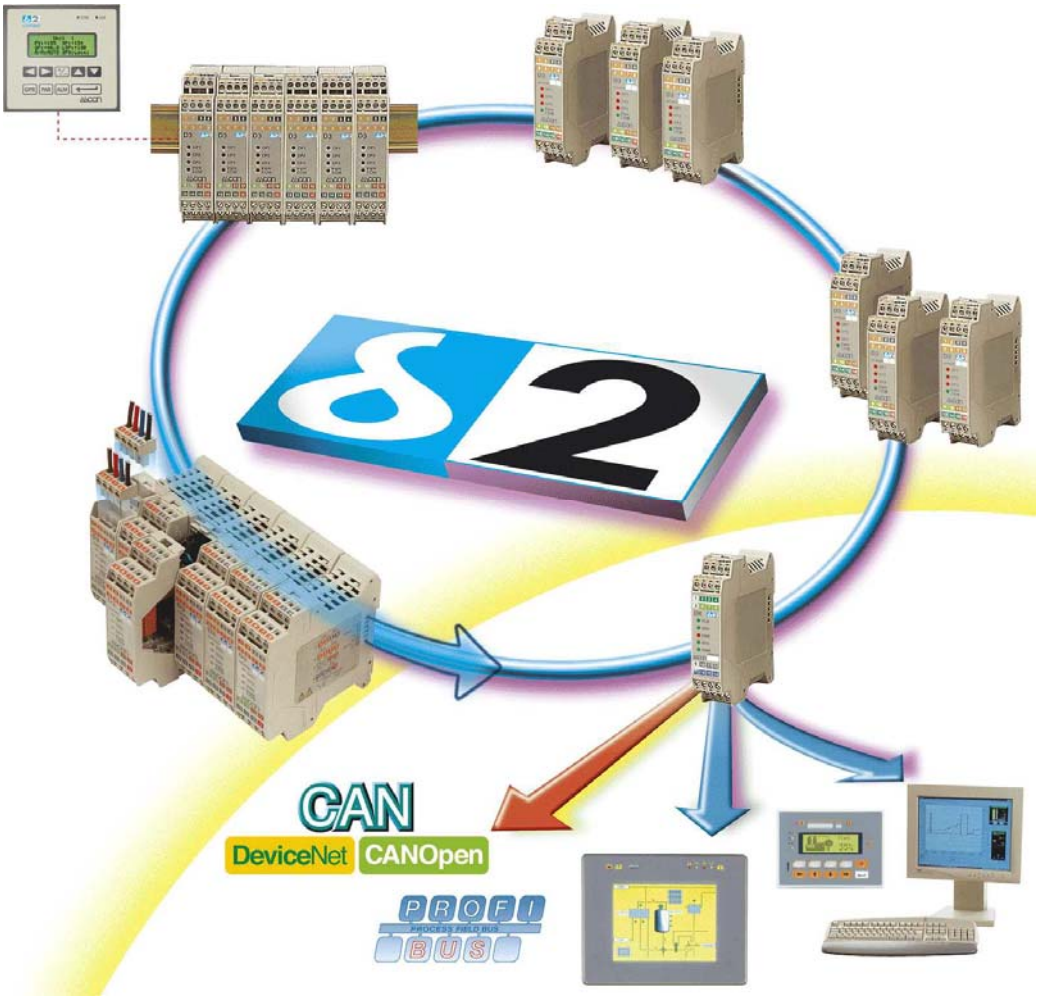
Delta2 Series

DIN rail mounting



DX Manager

DeviceNet Interface

Copyright 2007 ASCON spa

All rights reserved

Any part of this document can be reproduced or transferred in any form without previous authorization by ASCON spa.

ASCON guarantees the accuracy of the information contained in this document. To sustain technological leadership, ASCON products receive continuous improvements; this could requires modifications to the information contained in this document. ASCON will not be responsible for technical or editorial errors, omissions or incidental damages related to supply or use of this material.

All other brand or product names are the copyright property of the respective company.

ASCON spa
Via Falzarego 9/11
20021 Baranzate (Milano) - ITALY
Phone: 02 333371
Fax: 02 3504243
Internet: http://www.ascon.it
e-mail: info@ascon.it

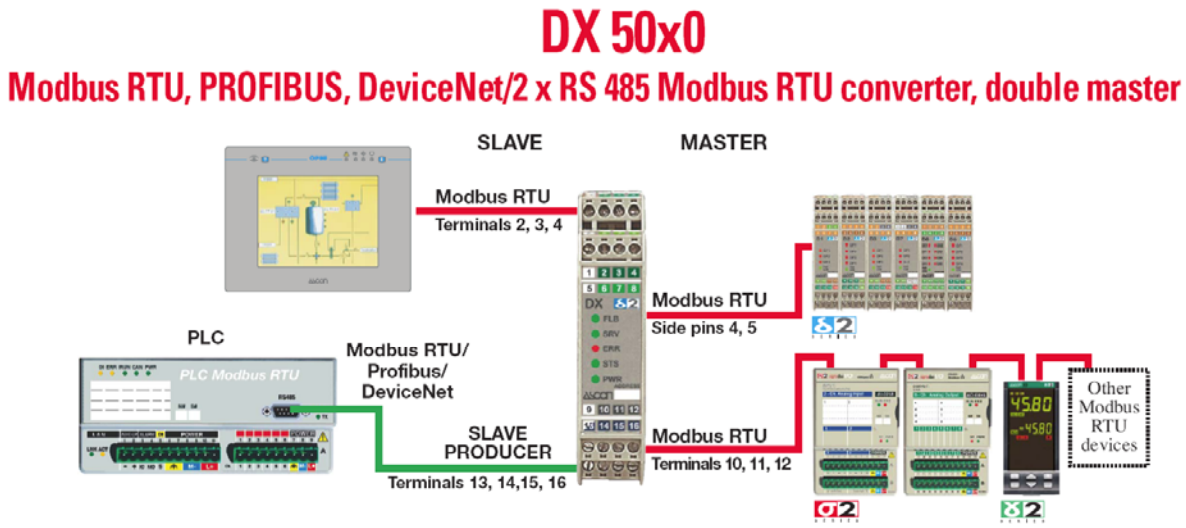
ASCON Corporation
1884 East Fabyan Parkway
Batavia, Illinois 60510
Phone: 630 – 482 – 2950
Fax: 630 – 482 – 2956
Internet: http://www.asconcorp.com
e-mail: info@asconcorp.com

Contents

1. A DeviceNet DX Manager network's example..... page 4
2. The DeviceNet Implementation..... page 4
3. What about CAN?..... page 5
   3.1 Features Description..... page 5
4. DX Manager Features..... page 5
   4.1 I/O Messaging..... page 6
      4.1.1 Bit Strobe Messages..... page 6
      4.1.2 Poll Messages..... page 7
      4.1.3 Change Of State (COS)/Cyclic Messages..... page 7
   4.2 Explicit Messaging..... page 7
   4.3 Fragmentation Protocol Byte..... page 8
   4.4 Additional Bytes..... page 8
5. Object Dictionary..... page 9
6. Service Channel..... page 10
7. Configuration from Service Channel..... page 10

1. A DeviceNet DX Manager network's example

This would be a possible topology representation of a solution by using the DX Manager with a Fieldbus port connected to a generic PLC and an HMI interface (Touch panel, SCADA system on PC or Server, etc.).



- Two masters can access simultaneously the instrumentation connected on Modbus RTU networks.
- One of the two masters must be Modbus RTU.
- The second master can be PROFIBUS or DeviceNet or Modbus RTU, the DX 50x0 converts these protocols to Modbus RTU.
- For example: the RS232 or RS485 service ports of DX 50x0 can be connected to an Autolink self configuring SCADA or to a local operator panel; the slave Fieldbus port can be connected to any PLC using PROFIBUS, DeviceNet or Modbus RTU protocols.

2. The DeviceNet Implementation

The implementation provides a Group 2 only Server, supporting all optional DeviceNet features, i.e.: Polled I/O, COS/Cyclic, Bit Strobe, Heartbeat Messaging.
For further information on the definition of these functions, please refer to the ODVA DeviceNet specifications.
DeviceNet masters exchange data with DeviceNet slave devices by using I/O messages; are available 4 different types of I/O messages: strobe, poll, change-of-state (COS) and cyclic.
The strobe I/O message, which is 8 bytes (64 bits) in length, is broadcast by a master to all devices on the network. Each bits corresponds to one of the node addresses (0 to 63).
Each device on the network that supports strobe I/O messages, responds by placing its input data, which can be a maximum of 8 bytes per device, on the network.
The poll I/O message (which can have 255 bytes of max length) is sent by a master to a specific slave device on the network. This is effectively peer-to-peer communication. The Input data from a slave device are read continuously and the output data to the device are written continuously.
A change-of-state (COS) I/O message is sent by a slave device to a master whenever the input data status changes or, in some cases, with a user-configurable rate (heartbeat). A COS I/O message does not solicit a response from the scanner. This is a peer-to-peer communication.
A cyclic I/O message is sent by a slave device to a master at a user-configurable rate. A cyclic I/O message does not solicit a response from the master. This is a peer – to - peer communication.

The granularity of data transmitted on a DeviceNet network is one byte. This means that even if just a single bit of data needs to be transmitted a whole byte has to be used.
The basic setup of the DX Manager is available through the configuration Service Port.
On the DeviceNet network are available 200 registers and 128 coils in Read, Write or Read/Write mode: it is also available the Service Channel mode that allow to read and/or write any single register on the Delta2 devices connected.

3. What about CAN?

The DeviceNet is based on communications protocol called CAN (Controller Area Network). The CAN is a deterministic protocol that has some interesting and advanced features as a fast response time and is highly reliable. With CAN, any node may transmit when the bus is not busy. If two or more nodes try to transmit at the same time, the message with the lowest CAN ID will complete the transmission. DeviceNet adds a layer above CAN that allows logical connections to exist among nodes and defines message formats. A single DeviceNet network may have up to 64 nodes, each with a unique address (MAC ID). DeviceNet supports baud rate of 125, 250, or 500 Kbps. As the baud rate increases, the maximum allowable length of cable between all the devices decreases. There is only one baud rate allowed per network, and all devices must operate at the same baud rate. The table below lists some of the more important DeviceNet features.

3.1 Features Description

- Network Size Up to 64 nodes;
- Network Length Selectable end-to-end network distance varies with speed;
- Baud Rate Distance;
- 125 Kbps 500 m (1640 ft);
- 250 Kbps 250 m (820 ft);
- 500 Kbps 100 m (328 ft);
- Data Packets 0-8 bytes;
- Bus Topology Linear (trunk-line/drop-line): power and signal on the same cable;
- Bus Addressing Peer-to-Peer with Multicast (one-to-many);
- Multi-Master and Master/Slave special case;
- polled or change-of-state (exception-based);
- Hot Swap of the devices from the network under power.

4. DX Manager Features

DeviceNet allows explicit message connections and I/O message connections. Explicit messages are more generic and flexible than I/O messages: are typically used for device configuration and diagnostics. I/O messages have less protocol overhead and are more efficient than explicit messages: are typically used to transfer control information. Most devices use predefined Master/Slave connection message sets for I/O data exchange. This set provides strobe, polled, change of state or cyclic transmissions. Whit a strobe connection, a master sends one strobe message and all enabled slaves will respond. With a polled connection, a master sends individual poll messages to each device and will expects a response from every single one. Change-of-State or Cyclic connections do not require command from the master: a slave device using either change of state (COS) or Cyclic connection, will send data when itself detects a monitored value(s) change. These messages restart the interval time and are used to prevent that fluctuating values will overload the network communication by sending continuous messages. A slave device with cyclic or COS connection sends data on a fixed (configurable) interval time and the COS connection will send also a message if detects data change before the end of interval time.
The DX Manager has pre-configured DeviceNet “objects” to access controller’s registers.

There are two basic message types: I/O messages and explicit messages.
I/O messages provide dedicated special-purpose communications between a producing application and one or more consuming applications.
Explicit messages provide generic, multi-purpose communication paths between two devices and are used to perform a particular task and report results of task performing.

4.1 I/O Messaging

I/O messages consist of small set of messages that may contain fixed amount of data. For example, Bit-Strobe messages can only transmit one bit of data from a master to a slave device and can only receive 8 bytes (64 bits) of data from each slave. Poll messages are used to send and receive significant amounts of data between masters and slaves and are only limited by bandwidth constraints. Cyclic/COS messages are also used to transmit large amounts of data from slaves to masters when data changes or at regular intervals.

I/O messages have low overhead.
Messages of 8 bytes or less have no communications overhead. Larger packs are less efficient because they require one byte of overhead plus an extra message pack for every 7 data bytes. For example, if you have 8 data bytes or less, the message is not fragmented and requires only the usual CAN header and trailer.
However, if you are sending 9 bytes of data, then the message is fragmented and an additional fragmentation protocol byte must be included in the first CAN message, which only leaves room for 7 data bytes. The remaining 2 data bytes require a 2nd CAN message with a CAN header, trailer and an additional fragmentation byte. Therefore, it takes 3 additional bytes to complete a 9-byte message. Although longer messages have a smaller percentage of overhead, they still consume network bandwidth.

If you are using a more complex device with configurable I/O, you might be able to change the message contents when I/O is added, but the message data structure is still determined by the manufacturer. I/O messages are also part of Group 2 and the DeviceNet specification states that a device is limited to only one Group 2 connection. Therefore, the device is allowed only to exchange I/O messages with a single master device.

Bit Strobe is related to the restriction on the size of Bit Strobe data. A value of 0 indicates that no data is available. From 0 to 8.

4.1.1 Bit-Strobe Messages

Bit-Strobe commands can rapidly transmit small amounts of data between a master and its bit-strobe slaves. Related of need to send/receive data quickly, the command response messages are limited to 64 bits (8 bytes) in length. Each bit corresponds to a MAC ID (0-63) supported on the network. Although the master device transmits all 64 bits, the network does not have to contain 64 devices. In addition, the message is only “consumed” by devices that are configured for Bit/Strobe messages from that particular master device. You can configure slave devices to ignore either the Bit-Strobe command, to consume the command and its output data or to consume the command and use it to trigger a response. Commands are ignored until a bit-strobe connection is established. The master then uses the connection to update data on the slave device and to receive data from the slave in the response (if configured).

4.1.2 Poll Messages

Poll commands can move any amount of I/O data between a master and its polled slaves and back again if required. Unlike the Bit-Strobe command, which broadcasts its message to the entire network, the Poll command is directed to one specific slave device. You can configure the slave device to either ignore the poll command and its output data or consume the command and use it to trigger a poll response. Poll commands, like Bit-Strobe commands, are also used by the master to update data on the slave device and to receive data from the slave in response.

4.1.3 Change of State / Cyclic Messages

Cyclic/COS messages are used for slave-to-master transmission but are not related to a specific request from the master. Although this message can consist of unlimited amounts of I/O data that is directed towards a single master device: maximum message size and content may be limited by the design of your device. In many cases, these items cannot be altered because some manufacturers of simple devices with fixed I/O have pre-set the contents of messages as well as message length.

Both COS and Cyclic messages can send messages at a fixed (configurable) interval time as well as when data change. When data changes, messages are sent out and the interval timer is restarted, which determines the maximum time between COS messages. To prevent that rapid data changes will overload the communication data transfer, the COS connection allows you to define a second interval time. This interval is a sort of delay that prevents the device from sending additional messages before the production inhibit timer has elapsed. This also allows you to define the minimum time between COS messages and also ignores changes of data during this time. When you establish a connection to a network, you can configure the module to automatically send next message without an acknowledgment or to postpone sending another message until the module receives an acknowledgment from master device. COS/Cyclic messages can also be combined with a Poll message to send data in response to scanner requests, at specified intervals or when data changes. The response to a Poll command will also re-start the interval timer mentioned above.

4.2 Explicit Messaging

Explicit messages allow a master device to retrieve and even change any of the DeviceNet objects within a module. The master has also access to all the data on the DeviceNet network and can issue commands to a module and its subsystems. However, these advantages require additional overhead on the message such as specifying the requested operation and the required objects in the module. Explicit messaging may also be provided by using UCMM services, which allow multiple connections to any module. The maximum number of connections and 0perations depend on the module's implementation. This information is provided with the module's specifications or with the statement of compliance. Another disadvantage by using Explicit messaging is that data is limited to the specific one in a particular object. If required data is not available into a single object, then multiple messages are needed to gather the data.

The protocol byte is a message header that is **only** used in the explicit message fragmented data field format. It is not used with I/O message fragmentation.

4.3 Fragmentation Protocol Byte

Both explicit messages and I/O messages contain a fragmentation byte that consists of a start, middle, or end fragment. The lower 6 bits are the sequence number and it represents the fragment numbers into a message. The fragment number starts with 0 and increases with each fragment, which ensures that no fragments are lost or repeated.
For explicit messages, the message recipient must acknowledge each message fragment before the next fragment is transmitted. This acknowledgment message sets both high bits to 1 and returns the sequence number in the lower 6 bits. If no acknowledgment is received within the fragment time-out, the fragment is resent.
For I/O messages, there is no acknowledgment received for fragmented messages, so each fragment is sent only once. The time-out for I/O messages is based on a complete message being received, so if a fragment is lost, the entire message is treated as lost and the transaction should time out.

4.4 Additional Bytes

The next byte in the message is the service code. The 3 most useful codes are Read Single Attribute (0EH), Write Single Attribute (10H), and Reset (05H). In the response, if the most significant bit is set, the service is successfully executed. If the service fails, the error response will have a special service code to indicate it.

Service codes are followed by object, instance and attribute numbers. Objects and instances are either 8 or 16 bits and are dependent to the connection type. 16-bit values are structured with the lower byte followed by highest byte.

Write commands are sequenced so data follows the object, instance and attribute information.

Reset commands may not require the attribute although attributes can be used to specify the type of reset.

5. Object Dictionary

This is list of parameters available in the DX Manager

Parameter	Class	Instance	Attribute	R/W	Length	Meaning
Register1 Output	100	1	1	RW	2	First register mapped as Write
…	…	…	…	…		…
Register200 Output	100	1	200	RW	2	Last register mapped as Write
Register1 Input	100	2	1	R	2	First register mapped as Read
…	…	…	…	…		…
Register200 Input	100	2	200	R	2	Last register mapped as Read
Coil Output	100	1	201	RW	16	Coils mapped as Write
Coil Input	100	2	201	R	16	Coil mapped as Read
Block64RegOutput	100	1	202	RW	128	Output Registers from 1 to 64
Block64RegInput	100	2	202	R	128	Input Registers from 1 to 64
Block64RegOutput	100	1	203	RW	128	Output Registers from 65 to 128
Block64RegInput	100	2	203	R	128	Input Registers from 65 to 128
Block100RegOutput	100	1	204	RW	144	Output Registers from 129 to 200
Block100RegInput	100	2	204	R	144	Input Registers from 129 to 200
Ass Obj Sel	100	1	205	RW	1	Select Ass Obj length
Block56RegOutput	100	1	206	RW	112	Output Registers from 1 to 56
Block56RegInput	100	2	206	R	112	Input Registers from 1 to 56
Block63RegOutput	100	1	207	RW	126	Output Registers from 57 to 119
Block63RegInput	100	2	207	R	126	Input Registers from 57 to 119
Bit Strobe Coil	100	2	208	R	8	First 64 coils for Bit Strobe
CLASS 101						
Redirect ID	101	1	1	RW	1	Service Channel ID
Redirect Addr	101	1	2	RW	2	Service Channel Address
Redirect Cmd/Func	101	1	3	RW	1	Service Channel Command and Status
Redirect Value	101	1	4	RW	2	Service Channel Value

With Parameter 100,1,205 is possible to select length of assembly object: the possible value (number of byte) are: 56, 64, 119, 128 or 200.

For every value, the corresponding number of bytes is:

Number of Bytes = (Value * 2) + Coils byte;

- 56 Registers --> (56 * 2) + 16 = 128 Bytes (attributes 201 and 206);
- 64 Registers --> (64 * 2) + 16 = 144 Bytes (attributes 201 and 202);
- 119 Registers --> (119 * 2) + 16 = 254 Bytes (attributes 201, 206 and 207);
- 128 Registers --> (128 * 2) + 16 = 272 Bytes (attributes 201, 202 and 203);
- 200 Registers --> (200 * 2) + 16 = 416 Bytes (attributes 201, 202, 203 and 204);

6. Service Channel

The Redirect parameters of Service channel are available trough explicit message. Redirect Node ID and Address mean node and address of Modbus Registers. All operations are controlled from Redirect Command/Status parameter.

Related to the writing operations:
- a value = 0 enables a reading command;
- a value = 1 enables a write command.

Related to the reading operations:
- a value = 0 means FREE, no operations ;
- a value = 1 means BUSY, an operation in progress;
- a value = 2 means READY, operation concluded;
- a value = 3 means ERROR, last operation had an error.

Redirect Value means value to write or value read to/from Modbus;

If Status is BUSY no write operation are possible on other parameters.

7. Configuration from Service Serial

From service port is possible configure the following settings:

Parameter	Meaning	Mapping	Note
Address	DX node on DeviceNet network	Holding Register 10000	From 1 to 63
Baudrate	DeviceNet Baud Rate	Holding Register 10001	0 = 125 KB 1 = 250 KB 2 = 500kB
Assembly Obj	Assembly object selection	Holding Register 10008	0 = 56 Registers 1 = 64 Registers 2 = 119 Registers 3 = 128 Registers 2 = 200 Registers
Swap	Byte order selection	Holding Register 10009	0 = Intel format 1 = Motorola