Supplement for
HE500OCS100, HE500OCS200,
HE5000OCS250 HE800RCS210
and Mini OCS Modules

DeviceNet™ Implementation Using Control Station Modules

Second Edition
28 July 2000

SUP0326-02

KEEP WITH USER MANUAL

PREFACE

This supplement is provided for Horner APG's Control Station modules for use in a DeviceNet™ network.

Information in this document is subject to change without notice and does not represent a commitment on the part of Horner APG, LLC.

For user manual updates, contact Horner APG, Technical Support Division, at (317) 916-4274 or visit our website at www.heapg.com.

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Horner APG, LLC. ("HE-APG") warrants to the original purchaser that Control Station modules for use in a DeviceNet™ network manufactured by HE-APG are free from defects in material and workmanship under normal use and service. The obligation of HE-APG under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within two (2) years from the date of manufacture or eighteen (18) months from the date of installation by the original purchaser whichever occurs first, such defect to be disclosed to the satisfaction of HE after examination by HE-APG of the allegedly defective part or parts. THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES AND HE-APG NEITHER ASSUMES, NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR HE-APG, ANY OTHER LIABILITY IN CONNECTION WITH THE SALE OF THESE Control Station modules for use in a DeviceNet™ network. THIS WARRANTY SHALL NOT APPLY TO THESE Control Station modules for use in a DeviceNet™ network OR ANY PART THEREOF WHICH HAS BEEN SUBJECT TO ACCIDENT, NEGLIGENCE, ALTERATION, ABUSE, OR MISUSE. HE-APG MAKES NO WARRANTY WHATSOEVER IN RESPECT TO ACCESSORIES OR PARTS NOT SUPPLIED BY HE-APG. THE TERM "ORIGINAL PURCHASER", AS USED IN THIS WARRANTY, SHALL BE DEEMED TO MEAN THAT PERSON FOR WHOM THE Control Station modules for use in a DeviceNet™ network ARE ORIGINALLY INSTALLED. THIS WARRANTY SHALL APPLY ONLY WITHIN THE BOUNDARIES OF THE CONTINENTAL UNITED STATES.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall HE-APG or its suppliers be liable of any special, consequential, incidental or penal damages including, but not limited to, loss of profit or revenues, loss of use of the modules or any associated equipment, damage to associated equipment, cost of capital, cost of substitute modules, facilities, services or replacement power, down time costs, or claims of original purchaser's customers for such damages.

To obtain warranty service, return the product to your distributor with a description of the problem, proof of purchase, post paid, insured and in a suitable package.

ABOUT PROGRAMMING EXAMPLES

Any example programs and program segments in this manual or provided on accompanying diskettes are included solely for illustrative purposes. Due to the many variables and requirements associated with any particular installation, Horner APG cannot assume responsibility or liability for actual use based on the examples and diagrams. It is the sole responsibility of the system designer utilizing Control Station modules for use in a DeviceNet™ network to appropriately design the end system, to appropriately integrate the Control Station modules for use in a DeviceNet™ network and to make safety provisions for the end equipment as is usual and customary in industrial applications as defined in any codes or standards which apply.

Note: The programming examples shown in this manual are for illustrative purposes only. Proper machine operation is the sole responsibility of the system integrator.

Revisions to this Manual

1. Added Appendix A: OCS Explicit Messaging.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Scope

This supplement covers the implementation of Control Station modules in a DeviceNet network. Chapter One provides general information about Control Station modules. Also covered are the capabilities of the modules in various networks including DeviceNet. Chapter Two covers the installation implementation of the Control Station Modules in a DeviceNet network. Chapter Three covers setup and configuration. Chapter Four provides operational information about an example network setup.

1.2 Control Station Modules

Table 1.1 lists the Control Station Modules.

Table 1.1 – Control Station Modules

Module	Part Number
Operator Control Station (OCS), Graphic OCS and Remote Control Station (RCS)	HE500OCS100; HE500OCS200 HE500OCS250 HE800RCS210
Mini OCS Modules (Network Versions only for DeviceNet)	HE500OCSxxx

1.3 Networks

1.3.1 Types of Networks Used with Control Station Modules

The Control Station Modules are used in two types of networks: CsCAN or DeviceNet.

a. CsCAN is a licensed network protocol created by Horner APG, LLC.

b. DeviceNet is an open network protocol that was originally developed by Allen-Bradley. The DeviceNet specification is currently administered by the Open DeviceNet Vendor Association, Inc. (ODVA).

DeviceNet and CsCAN use Controller Area Network (CAN) protocol. However, they can <u>not</u> co-exist on the same physical CAN network.

The capabilities of the Control Station modules vary in accordance with the type of network the modules are configured for. Table 1-2 lists the differences in the capabilities of the Control Station modules when implemented into a DeviceNet *or* CsCAN network.

Table 1-2 – OCS/RCS Capabilities in a DeviceNet or CsCAN Network

Capability	DeviceNet	CsCAN
Upload/Download Ladder Code over network.	No	Yes
Ability to control the PLC and perform de-bugging.	No	Yes
a. Must directly upload/download programming into *each* individual device on network. b. Permits the upload/download of programming for all devices on the network via one device.	a. Yes b. No	a. No b. Yes
Allows the use of slave devices made by various manufacturers.	Yes	No-Uses Horner Slaves.
Uses Cscape Software to configure and program the Control Modules. Also uses Cscape Software to configure, program and download Ladder Code into the modules.	Yes	Yes
Supports 125K, 250K, and 500K baud CAN Network	Yes	Yes
Supports 1Megbaud CAN Network	No	Yes
Provides for peer-to-peer communication	*****No	Yes
* To provide for peer-to-peer communications in a DeviceNet network, the "Unconnected Message Manager" [UCMM] feature must be available. Currently, the Control Station Modules do <u>not</u> support the UCMM method of communication in a DeviceNet network.		

1.4 Additional DeviceNet Information

For more information on DeviceNet networks or the Open DeviceNet Vendor Association, Inc. (ODVA), visit the ODVA web site at **www.odva.org**.

1.5 Before Getting Started

Wiring diagrams, specifications, and other pertinent information for the Control Station modules are covered in the user manuals listed in Table 1-3.

Table 1-3 – Additional References for the Control Station Modules

Module	GE Part Number
Operator Control Station (OCS100 & OCS200), Graphic OCS (OCS250) and Remote Control Station (RCS210)	***These modules are covered in User Manual: GFK-1631***
SmartStack™ Modules (Individual data sheets for various models.)	***These modules are covered in User Manual: GFK-1601***
Mini OCS Modules (OCSxxx) (Only network versions are supported using DeviceNet.) (Individual data sheets for various models.)	***These modules are covered in User Manual: GFK-1687***

NOTES

CHAPTER 2: DEVICENET INSTALLATION AND IMPLEMENTATION

2.1 General

Chapter Two covers the features and requirements of Control Station modules in a DeviceNet network.

2.2 Installation

Installation information is contained in the user manual for each Control Station product. Wiring diagrams, connectors, specifications, and other pertinent information are covered in the installation chapter for each Control Station module.

To install the Control Station modules in a DeviceNet network, be sure to check the connectors and pin-outs, total cable length vs. CAN Network baudrate, power supply and grounding information. (See Table 1-3 in this manual for a list of applicable user manuals for the OCS, Graphic OCS, RCS, and Mini OCS Control Station products.)

2.3 DeviceNet Features Supported Using Control Station Modules

It is important to determine the features that the Control Station Modules support (when used in a DeviceNet network) *before* programming, configuring, and setting up the network.

a. Communication Method

The DeviceNet Specification provides two communications protocols or methods that can be used to communicate with a DeviceNet slave device. First is the Unconnected Message Manager (UCMM) and the other is the Group 2 Only Server. Only one of these two protocols or methods can be implemented by any specific DeviceNet slave node. The Control Station Modules implement the Group 2 Only Server protocol.

The DeviceNet Specification defines a Group 2 Only Server as:
A slave (server) device that is UCMM incapable and must use the Predefined Master/Slave Connection Set to establish communications (at a minimum, the Predefined Master/Slave Explicit Messaging Connection must be supported). A Group 2 Only device can transmit and receive only those identifiers defined by the Predefined Master/Slave Connection Set.

b. Message Types / Message Priority

Both Explicit Messaging and I/O Connections are supported using Group 2 Only and the Predefined Master/Slave Connection Set.

c. Types of Messages / Connections Supported

The **Explicit Messaging** and **Polled Connection** features are supported. The following terms are defined per the DeviceNet™ Specification, Release (2.0), Volume I: DeviceNet Communication Model and Protocol, Sections 1-4, 1-4.2, and 7-3.

Explicit Messaging: Used in typical request/response-oriented networks. Explicit Messages are used to command the performance of a particular task and to report the results of performing the task.

Polled Connection: Responsible for receiving the Master's Poll Command and returning the associated response.

“Explicit Messaging” is used for the transfer of small amounts of infrequently accessed data such configurations or tuning.

The “Polled Connection” is used for data that is sent automatically between the slave and the DeviceNet Master (Scanner) without intervention. The DeviceNet Master sends a **Polled Command** to a DeviceNet Slave. The DeviceNet Slave produces or sends a **Polled Response** back to the DeviceNet Master. (In the Control Station, the **Polled Response** consists of 16 words of data.) All Control Station modules monitor (or “eavesdrop” on) **Polled Response** messages on the network. (See **Polled Snooping** in Section 2.3, item d.)

The Control Station Modules do <u>not</u> support **Bit Strobed**, **Cyclic** and **Change-of-State** connections.

d. Additional Feature available in Control Station Modules

The Horner extension of the DeviceNet protocol known as **Polled Snooping** is also provided. Polled Snooping is totally non-intrusive to DeviceNet traffic. The Control Station Modules monitor (or “eavesdrop” on) all **Polled Response** messages on the network. The data in the **Polled Responses** can used by the ladder code program in each of the Control Station Modules. Their ladder code programs can make decisions based on the “snooped” data. (See **Polled Connection** in Section 2.1.1, item c.)

As **Polled Response** messages are received by other Control Station Modules, the first 16 words of the response data are copied into an input map array indexed by the Media Access Control Identifier (MAC ID). (The MAC ID is the DeviceNet Node Address of the transmitting node.) Through a process called **Mapping,** any 16 words of data within the Control Station Modules input map array can be selected to be copied into the %AIG registers of the Control Station module. These 16 %AIG registers can then participate in the ladder code running in the Control Station modules.

2.4 Configuration and Programming / Software Requirements

All configuration and ladder programming is performed through the use of Cscape™ Software. Cscape is a 32-bit Windows program that runs on a Personal Computer (PC) using the Windows 95, Windows 98 or Windows NT operating system. Cscape provides for a serial connection between the PC and the target module. See the *Cscape Software User Manual* (GFK-1630) for more information regarding the features and operation of Cscape.

Note: It is <u>not</u> currently possible to perform ladder code updates, operating firmware updates or other administration and control functions over a DeviceNet network. These function can only be performed by a direct serial connection between a PC running Cscape and each individual front panel serial port of the Control Station modules.

2.5 Requirements to Integrate Control Station Modules in a DeviceNet Network

The following requirements must be met to integrate Control Station Modules into a DeviceNet network and to establish communications with a DeviceNet Master.

a. DeviceNet Node Address (MAC ID)

In order to communicate with a DeviceNet Master, a unique MAC ID must be assigned to each node on the network. Valid DeviceNet MAC IDs are within the range of 0 - 63 inclusive (at least one master and 63 slave and/or master nodes).

b. DeviceNet Baud Rate

Each device connected to a DeviceNet network must be configured to operate at the same baud rate. The baud rate for the Control Station Modules can be selected to be 125K, 250K or 500Kbaud. All devices on a DeviceNet must be configured to operate at the same baudrate.

c. Polled Input Data Assembly

The Control Station Modules “consume” (or expect to receive) 16 words of data whenever a polled command is received. These 16 words are stored in the Input Map array (at Index 64) in the Control Station Module.

d. Polled Output Data Assembly

After the poll command is received, the Control Station Modules “produce” (or send) 16 words of data to the DeviceNet Master as a response to a polled command.

2.6 DeviceNet LEDs

Table 2-1 Module Status LED (MS)	
LED Indicator	**State**
Off	No power applied to device.
Green	Device operates in a normal condition.
Flashing Green	Device in Standby due to incomplete, incorrect or missing configuration.
Flashing Red	Minor Fault - is recoverable.
Red	Device has unrecoverable fault - may need replacing.
Flashing Red-Green	The device is in Self Test.

Table 2-2 Network Status LED (NS)	
LED Indicator	**State**
Off	Not powered or Device is not on-line.
Green	Link is OK, Device is on-line and connected.
Flashing Green	Device is on-line but is not connected in the established state.
Flashing Red	One or more I/O connections are in the timed-out state.
Red	Critical link failure. Device detects error which prevents communication over network.

2.7 Self-Test Diagnostics

To ensure system integrity, the Control Stations maintain user-accessible diagnostic information. This information is available in the %S2 System Register bit (NET_OK) and in the %SR4 System Register word (TEST_RESULT).

The %S2 bit is ON only if the network is functioning properly. If %S2 is OFF, the cause (or causes) appears in %SR4. %SR4 is a 16-bit register with each bit corresponding to the Pass/Fail result of a specific Control Station self-test. Normally, all bits in %SR4 are OFF. If one of %SR4's bits is ON, it indicates that the corresponding self-test failed.

Each fault bit in the %SR4 register falls into 1 of 3 categories:

1. Critical Fault Controller fault; prevents controller from running user program
2. Internal Fault Controller fault; user program determines remedial action, if any
3. Network Fault Controller network fault; user program determines remedial action, if any

The following table shows how fault bits are mapped into the %SR4 register.

Table 2.3 - %SR4 (TEST_RESULT) Fault Mapping			
%SR4 Bit	**Fault Name**	**Fault Type**	**Fault Description**
%SR4.1	BIOS_ERROR	Critical	Controller BIOS firmware checksum test failed.
%SR4.2	ENGINE_ERROR	Critical	Controller engine firmware CRC test failed.
%SR4.3	LADDER_ERROR	Critical	User program CRC test failed (defaults loaded).
%SR4.4	RAM_ERROR	Critical	Controller RAM memory test failed.
%SR4.5	DUP_ID_ERROR	Network	Duplicate network ID test failed; another node, which has been assigned this node's MAC ID, is operating on the network.
%SR4.6	BAD_ID_ERROR	Network	Illegal network ID detected; this node's assigned MAC ID is not in the 0 to 63 range (forced to be node 63).
%SR4.7	IO_CFG_ERROR	Internal	This node's I/O configuration does not match the set of SmartStack modules actually plugged in.
%SR4.8	BAD_NET_ERROR	Network	Network power is OFF or no other nodes were found on the network.
%SR4.9	BAD_LOGIC_ERROR	Critical	User program contains an illegal ladder instruction.
%SR4.10	BAD_CLOCK_ERROR	Internal	Controller's real-time-clock has not been initialized.
%SR4.11	DEVICENET_ERROR	Network	Connection to Master DeviceNet scanner has not been established, or the Master DeviceNet scanner is in Program Mode instead of in Run Mode.
%SR4.12	Reserved	-	Reserved for future use (should be 0)
%SR4.13	Reserved	-	Reserved for future use (should be 0)
%SR4.14	Reserved	-	Reserved for future use (should be 0)
%SR4.15	Reserved	-	Reserved for future use (should be 0)
%SR4.16	Reserved	-	Reserved for future use (should be 0)

NOTES

CHAPTER 3: SETUP AND CONFIGURATION

3.1 Initial Procedures and Requirements

The initial procedures and requirements to install and configure the Control Station modules are covered in Chapter 3. Before starting, consider the following information.

1. Layout a DeviceNet scheme on paper and define I/O's, node address references, etc.

It is important that the initial DeviceNet network be properly designed at the front-end of the project and take in consideration any drops and nodes that will be added in the future. Haphazard additions of extra "drops" and nodes at a later date often cause problems. Unterminated drops can be tolerated on a DeviceNet network. Adding nodes later in the project can make configuration of the DeviceNet Master (and its associated PLC) much more difficult and prone to human error.

2. Configure the DeviceNet Master.

Consult the manufacturer's user guide for procedures to configure the master.

The following items are required for communication between a DeviceNet Master and Control Station Modules.

a. DeviceNet Node Address (MAC ID)

In order to communicate with a DeviceNet Master, a unique MAC ID must be assigned to each node on the network. Valid DeviceNet MAC IDs are within the range of 0 - 63 inclusive. See the applicable user manual for the controller –refer to Table 1-3. Also refer to the user manual to **setup the MAC ID for the Control Station Modules,** (configured via Cscape). **Other slave devices on the network may need different procedures.** Consult the slave manufacturer's user manual for procedures to set the MAC ID.

To set the MAC ID for the RCS210, refer to the Control Station Hardware Manual (MAN0227), Section 6.3, item b.

b. DeviceNet Baud Rate

Each device connected to a DeviceNet network must be configured to operate at the same baud rate. The baud rate for the Control Station Modules can be selected to be 125K, 250K or 500Kbaud and is configured via Cscape. Consult the slave manufacturer's user manual for procedures.

To set the MAC ID for the RCS210, refer to the Control Station Hardware Manual (MAN0227), Section 6.3, item b.

c. Polled Input Data Assembly (Consumption Size)

The Control Station Modules "consume" (or expect to receive) 16 words of data whenever a polled command is received. The words of data are stored in the Input Map Array (at Index 64) in the Control Station Module. A process called "Polled Snooping" is described in 2.1.1 item d.

d. Polled Output Data Assembly (Production Size)

After the poll command is received, the Control Station Modules "produce" (or send) 16 words of data to the DeviceNet Master as a response to a polled command.

3.2 Configuration Procedures

The following procedures are used to configure a Control Station Module for use in a DeviceNet network. The configuration for the Control Station Module is accomplished using Cscape Software. **For the example configuration, the Control Station Module is an OCS100.**

Note: Other slave devices on the network may require different configuration procedures. Consult the slave manufacturer's user manual for configuration procedures.

3.2.1 Selecting a Controller

1. Go to the Main Screen and press Controller. Select Configure.

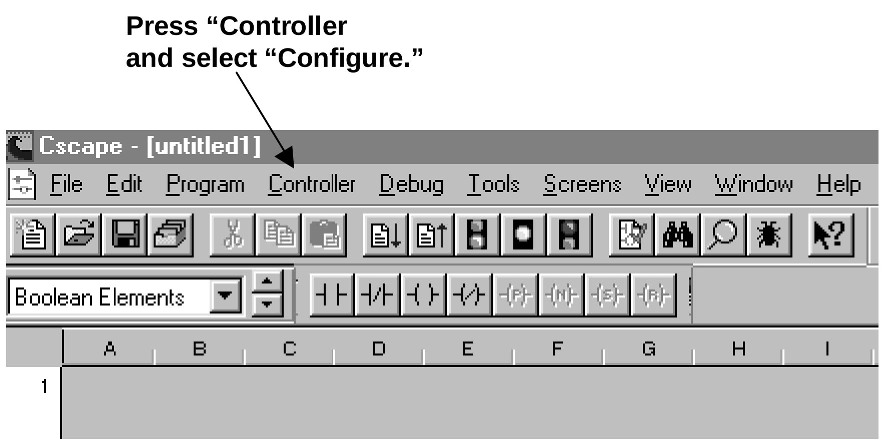


Figure 3.1 – Select Controller

2. The following screen appears. The IC300OCS100-CsCAN is shown as the selected Control Station module. In this configuration example, however, the desired Control Station module controller is the **IC300OCS100-DevNet**. Click the Config button located next to the IC300OCS100 slot.

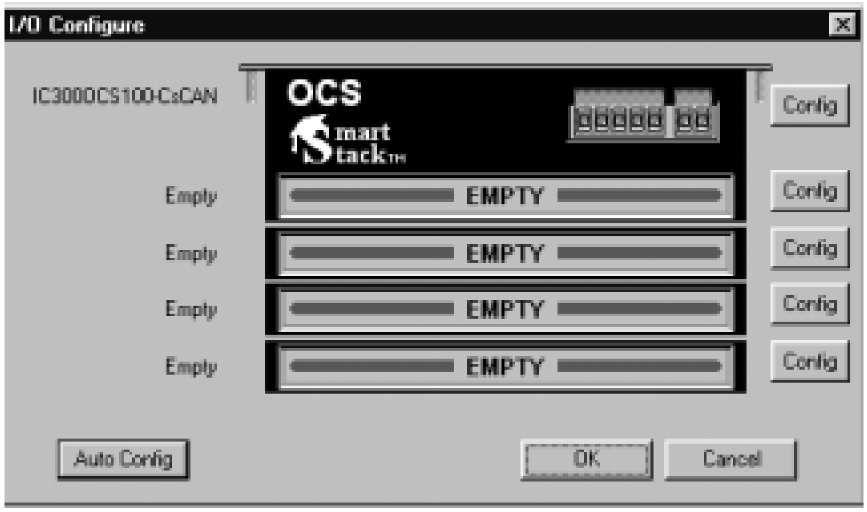


Figure 3.2 – Configure Controller Type Dialog

3. The following screen appears. Select the desired Control Station from the pull-down menu, and press OK.

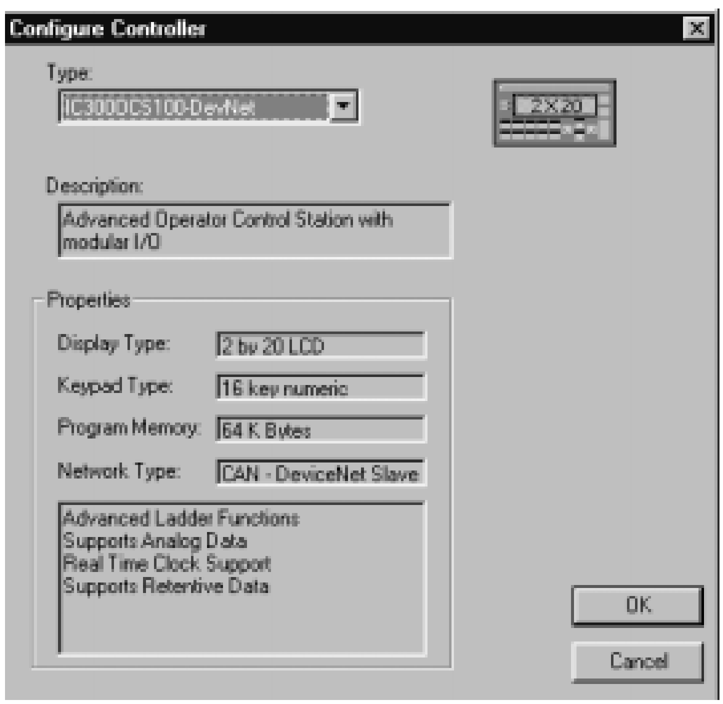


Figure 3.3 – Selecting a Control Station Module for Use in a DeviceNet Network

4. The following screen appears. The desired controller is now selected. Press the OK button.

Figure 3.4 – OCS100-DevNet Selected

3.2.2 Network Configuration

5. From the Main Screen, press Program and select “Network Config.”

Press Program. Select “Network Config.”

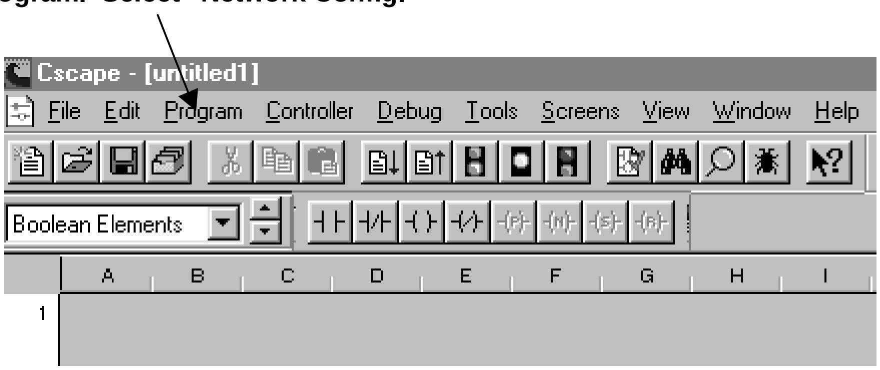


Figure 3.5 – Selecting “Network Config”

6. The following screen appears.

a. The **Point** node is the Control Station that is being configured.
b. The **Source Word** represents a different Control Station on the network than the **Point** node.
c. The **Net ID** address is that of the **Source Word**.

Configuring this screen allows the Point node to receive data from:

a. Various **Source Word's %AQG** registers and to have the data sent to the **Point** node's **%AIG** registers. (See Steps #6 through #8.)
b. The DeviceNet **Master** (using a special index address of **"64" or "M" for "Master"** and to have the data sent to the **Point** node's **%AIG** registers. (See Steps #9 through #11.)

<u>Example Configuration:</u> This example depicts a Point node receiving data from a Source Node (Refer to Figure 3.12.)

In this example configuration, the user wants the ***%AQG8*** data from a ***Source Word*** (with a ***Net ID of 5***) to be sent to the ***Point*** node's ***%AIG2*** register.

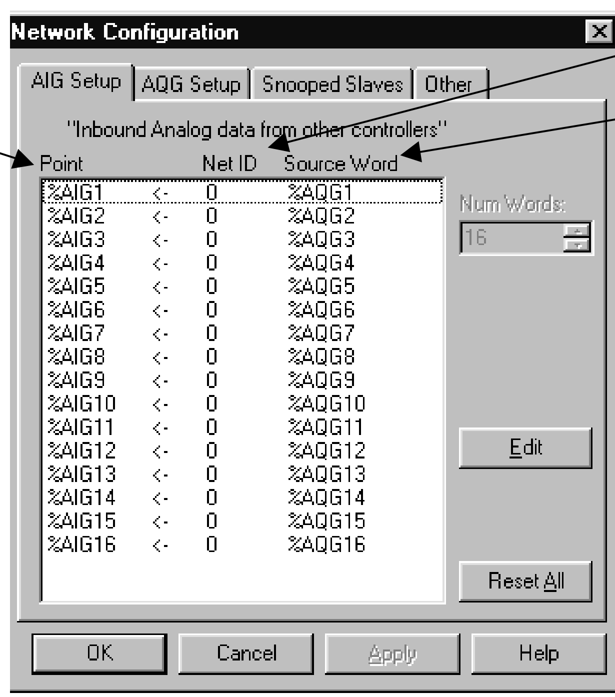
Note: An example of a Point node receiving data from a DeviceNet Master is illustrated later in Step #9.

To edit the screen and select the appropriate %AIG and %AQG registers and Net ID, use the mouse arrow and double-click on the desired **Point** node's **%AIG2** register.

1. The user needs to edit this screen to achieve the following transfer of data:

The user wants the %AQG**8** register data from the **Source Word** to be sent to the **Point node's** %AIG**2** register.

The **Source Word** has a **Net ID** address of **5**.



Network ID: Represents the node address of the **Source Word**, which is a *different* Control Station on the network than the **Point**.

The user wants to edit this screen so that the **Source Word's** %AQG8 data is sent to the **Point** node's %AIG**2** register.

Figure 3.6 – Network Configuration Screen (Before Configuration Parameters are Set)

7. The following screen appears. Using pull-down menus, select the node address of the **Source Word** and the %AQG.register from which the data is to be obtained. ***In this example, the data is to be obtained from a Control Station (Souce Word) with a Net ID of 5, and the information is to be obtained from %AQG8 within the Source Word.*** Press OK.

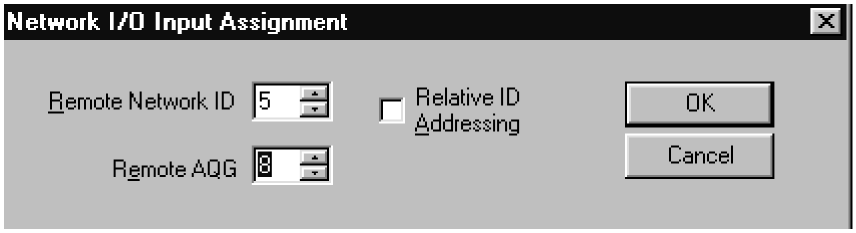


Figure 3.7 – Network I/O Input Assignment Screen

8. The following screen indicates that the **Source Word's** %AQG8 data is to be sent to the **Point** node's %AIG2 register. The **Source Word's Net ID** is 5.

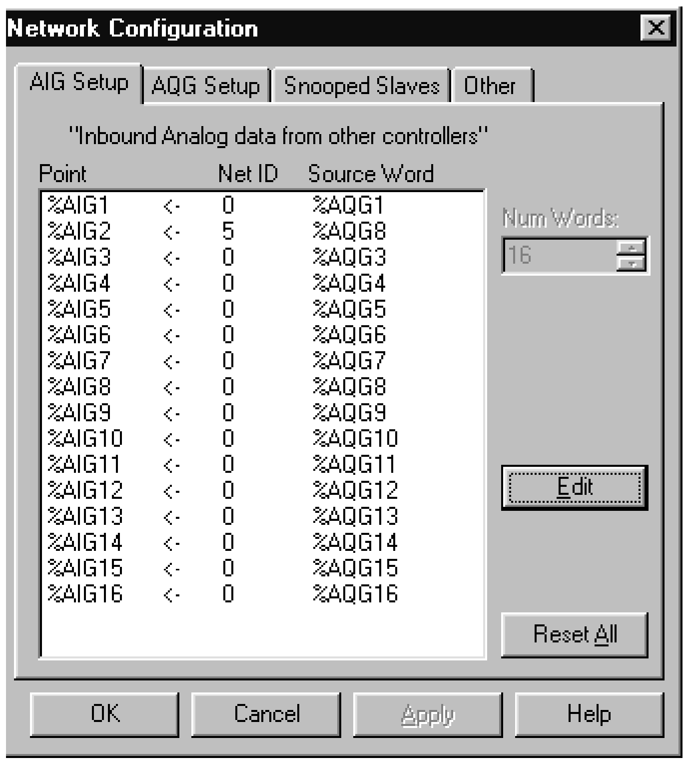


Figure 3.8 – Network Configuration Screen
(Shows Status of Example Configuration%AIG2)

9. Steps #6 through #8 illustrated an example of the **Point** node receiving data from a **Source Word** (Control Station Module).

The following **example** illustrates how the **Point** node receives data from the **DeviceNet Master**. *The DeviceNet* ***Master*** *uses special index address of* ***"64"*** or the user can type in ***"M"*** for **"Master."**

Example Configuration:

In this example configuration, the user wants the ***%AQG1*** data from a ***DeviceNet Master*** (with ***a special Net ID of "64"*** or the user can type in ***"M"*** *for* ***"Master."***) to be sent to the ***Point*** node's ***%AIG1*** register.

To edit the screen and select the appropriate %AIG and %AQG registers and Net ID, use the mouse arrow and double-click on the desired **Point** node's **%AIG1** register.

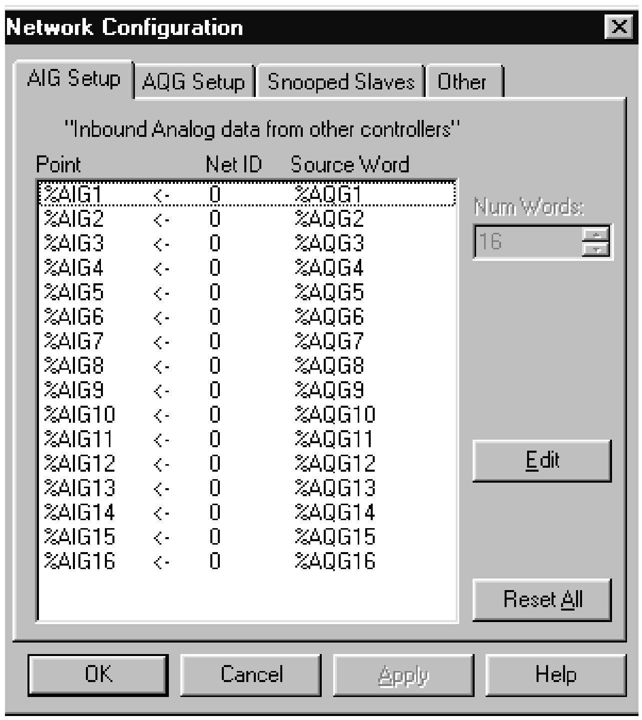


Figure 3.9 – Network Configuration Screen (Before parameters have been set.)

10. The following screen appears. Using pull-down menus, select the node address of the **DeviceNet Master** and the %AQG.register from which the data is to be obtained. ***In this example, the data is to be obtained from a DeviceNet Master (Source Word) with a special Net ID of "64" or "M" for "Master," and the information is to be obtained from %AQG1 within the Source Word.*** Press OK.

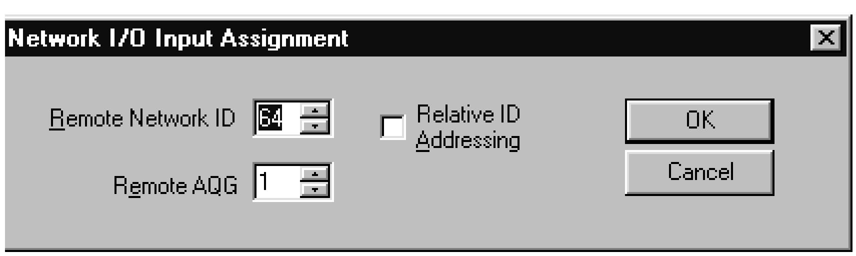


Figure 3.10 – Selecting the DeviceNet Master's Special Address of "64"

11. The following screen appears. It indicates that the **Point** node's %AIG1 register is to receive data from the **DeviceNet Master's** %AQG1 register (Source). The **DeviceNet Master** has a special **Net ID** of **"64"** or the user can type in ***"M" for "Master."***

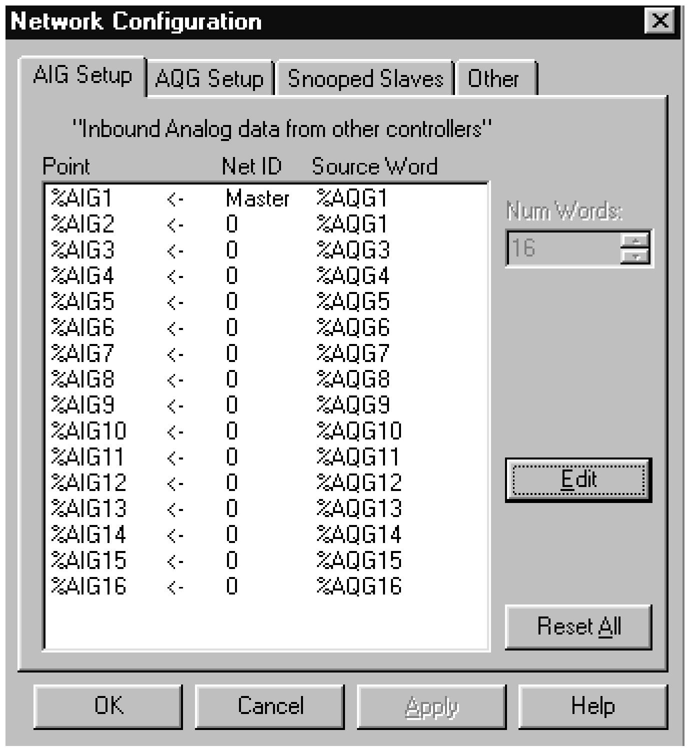


Figure 3.11 – Network Configuration Screen
(Shows Status of Example Configuration%AIG1)

3.2.3 Relative ID Addressing

9. The **Relative ID Addressing** feature simplifies the configuration process involving a *number* of Control Station Modules that are located on different conveyor segments. If *each* Control Station is going to be configured such that:

a. The ***same %AIG registers*** are going to be configured in *each* Point node to store specific data sent from the Source nodes AND
b. The ***same %AGQ registers*** of the Source nodes are going to be configured to send specific data to the Point node's %AIG registers, THEN
c. The **Relative ID Addressing** feature allows a single ladder file (which includes logic, configuration, and the network map) to be used by a number of Control Station s on the network, which are performing the same functions.

Example: Configuration using Relative ID Addressing Feature

In this configuration example , the Relative ID Addressing feature is selected. In Figure 3.9, the user wants the **Point** node to be configured such that the **Source** nodes (on conveyor segments located just before and after the **Point** node) send the data from the same %AQG registers to the same %AIG registers as in the **Point** node. (The process is described in more detail in Section 3.2.2.)

Note that **Source -1** is one Net ID address *lower* than the Point's Net ID address. Also note that **Source+1** and **Source +2** Net ID addresses are *greater* than the Point's Net ID address by 1 and 2 respectively.

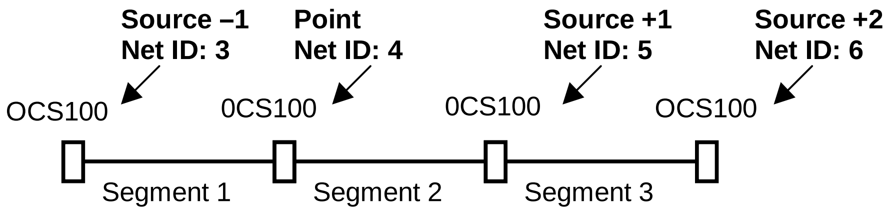


Figure 3.12 – Example Conveyor Line

In Figure 3.13, the user has clicked on the box to select "Relative ID Addressing." The user has typed in "-1" for the Remote Network ID. This means that the Net ID of the Source is relative to the Point's Net ID by an *offset of minus one*. Press OK.

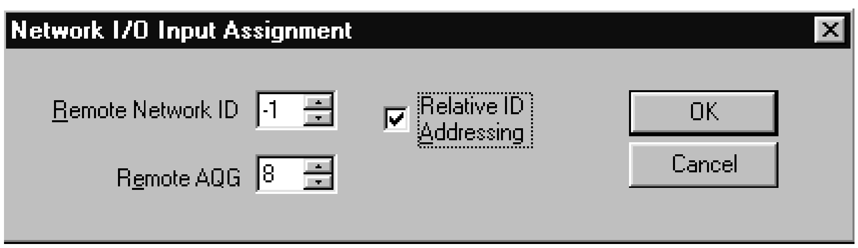


Figure 3.13 – Network Configuration Screen

The following screen appears. The screen indicates that the Point node is to receive data from two Source nodes that are:

a. One Net ID address **greater** than (+1) the Point's Net ID address, and
b. One Net ID address **less** than (-1) the Point's Net ID address.

The screen also depicts that the Point node's %AIG**1** register is to receive data from the +1 Source's %AQG**6** register. The Point node's %AIG**2** register is to receive data from the -1 Source's %AQG**8** register.

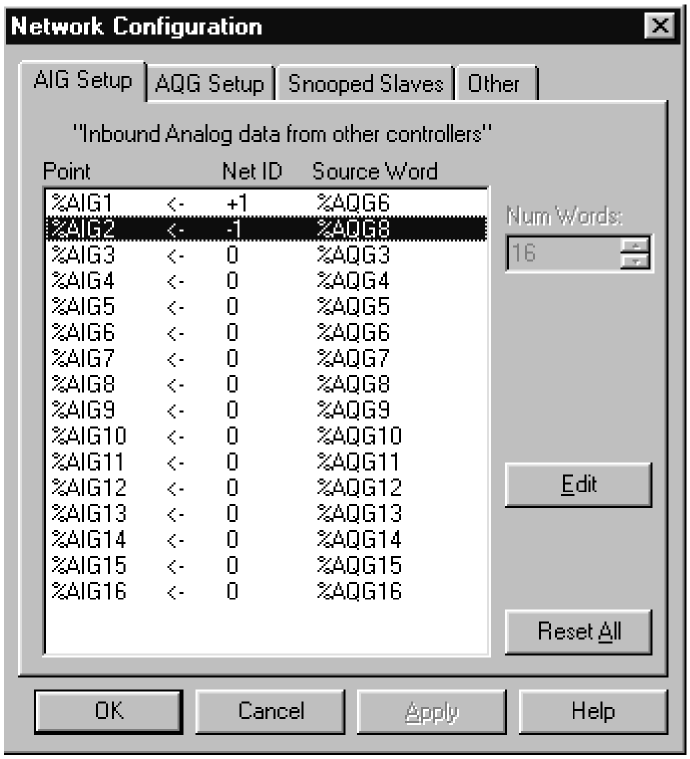


Figure 3.14 – Network Configuration Screen
(Shows Status of Relative ID Addressing)

3.2.4 Setting up Snooped DeviceNet Slaves

Snooping allows the Control Station to gather data from other slaves on a network as the master polls that slave device. Polled response data sent from the slave to the master can be in one of two formats depending on the amount of data the slave needs to send to the master.

a. If the data sent from the salve is 8 bytes or less, the data is not fragmented, and the box next to the Mac ID of that slave is not checked.

b. If the slave produces *more* than 8 bytes of data, the fragmented format is used, and the box next to the MAC ID of that slave needs to be checked.

Note: This information is only needed if the controller being configured needs to gather data or snoop on data from another slave on the network. Setting this incorrectly will not affect the performance or behavior of the DeviceNet network, but may prevent the controller from being able to obtain correct data from the snooped slave.

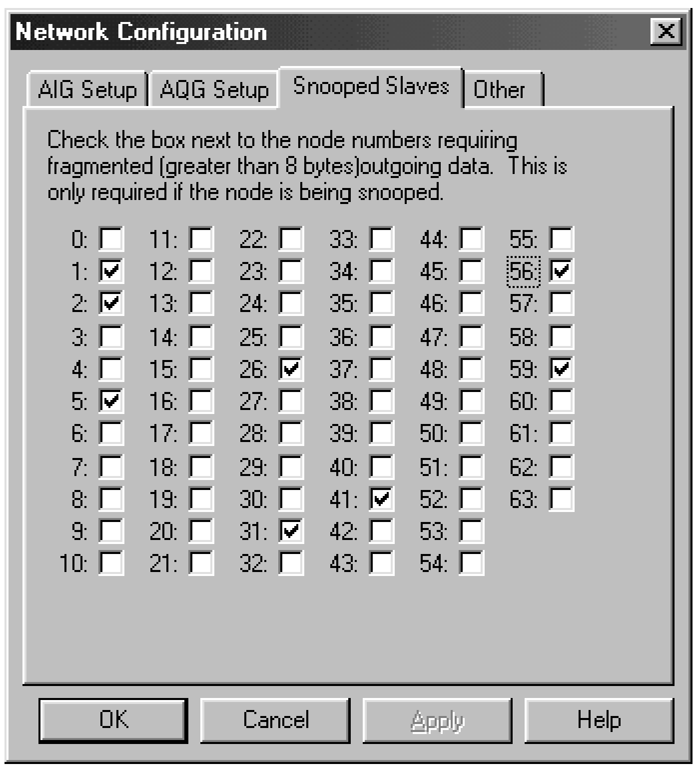


Figure 3.15 – Setting Up Snooped Slaves

NOTES

CHAPTER 4: OPERATION

4.1 DeviceNet Overview

Chapter Four describes an example DeviceNet network using Control Station Modules **(OCS100)** in a conveyor application. Refer to Figure 4.1.

1. The **DeviceNet Network** allows a DeviceNet Master to communicate with slave nodes over a common bus. Figure 4.1 depicts a DeviceNet network using OCS100 devices. Conveyors are typically laid out in segments. Each segment typically uses a Control Station (i.e., OCS100) to control a specific segment and to communicate with Control Station Modules located on other segments of the conveyor. The OCS100 also communicates with the DeviceNet Master. A total of 64 nodes is available on the network including the DeviceNet master. Up to 63 slave devices can be used.

In the example configuration, two segments of a conveyor operation are shown. Each segment has a Control Station Module (OCS100) and one other slave device connected to it.

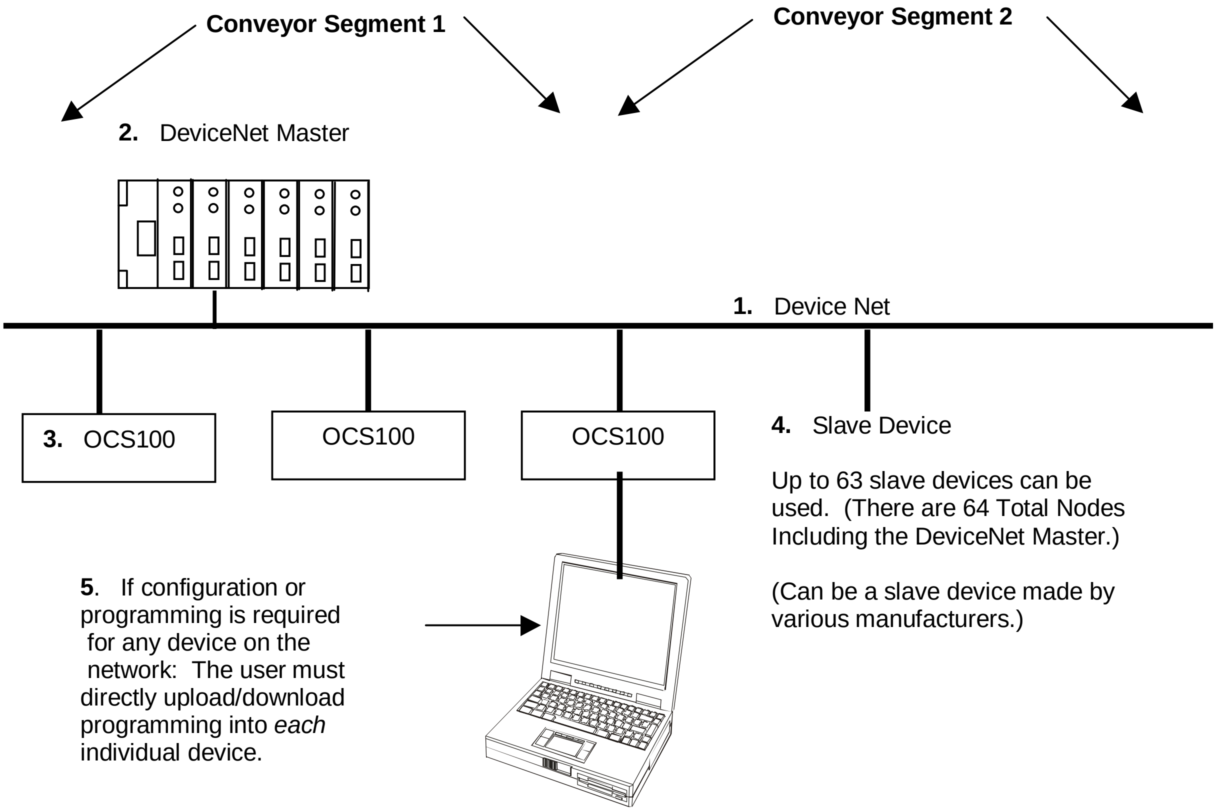


Figure 4.1 – Using Control Station Modules in a DeviceNet Network

2. The **DeviceNet Master** sends two types of messages to the slaves. The first message is an **Explicit Message**, which is used for the transfer of small amounts of infrequently accessed data such as configuration or tuning. The second type of message is a **Polled Connection**. It is used for data that is sent automatically between the slave and the DeviceNet Master (Scanner) without intervention. When the slave responds with a “Polled Response,” all of the Control Station Modules “snoop” the data as described in item 3. (See Appendix A for details of Explicit Messaging as implemented in the Control Station Modules.)

3. The DeviceNet Master sends data *to* the **Control Station modules** (“Polled Commands”) and receives data *from* the **Control Station modules** (“Polled Responses”). Using a feature called “**Polled Snooping,”** all Control Station modules (i.e., OCS100) monitor (or “eavesdrop” on) the “Polled Response” messages on the network. “Polled Snooping” is non-intrusive to DeviceNet traffic.

The data in the “Polled Responses” can be used by the ladder code program in each of the Control Station Modules (i.e., OCS100). The individual ladder code programs of the Control Station modules can then make decisions based on the “snooped” data.

Each of these processes are described as follows:

DeviceNet Master sends data to a Control Station module , and the Control Station module responds; The other Control Station Modules in the network “snoop” the data, which is sent into their %IG registers via “Mapping”:

a. The DeviceNet Master sends a “Polled Command” to up to 63 **DeviceNet Slaves** on the network. Each of the Slaves that were sent the “Polled Command” produces or sends a “Polled Response” back to the DeviceNet Master. (The “Polled Response” for Control Stations consists of 16 words of data.). All other Control Station Modules (i.e., OCS100) snoop (or “eavesdrop” on) all “Polled Response” messages on the network.

b. The “Polled Command” data that is send from the DeviceNet Master to specific Control Station Modules is sent to a special address or “storage location” in the individual Control Station Module’s **Input Map Array** at Index 64. The Input Map Array has sixty-five addresses or “storage locations” (0-64). There is an index for each of the 64 nodes on the network.

c. As “Polled Response” messages are received by the other Control Station modules, the first 16 words of the response data are copied into each of the Control Station modules’ Input Map Array.

d. Through a process called **“Mapping,”** any 16 words of data within each Control Station module’s Input Map Array can be selected to be copied into its %AIG registers. (The user determines the selection of the 16 words via the configuration process.) These 16 %AIG registers can then participate in the ladder code running in the Control Station Modules. “Mapping” occurs at the start of every PLC Ladder scan.

CAUTION

When the controller stops, all output data is set to zero.

The Ladder Code sends data to the DeviceNet Master via a "Polled Response" message; Other Control Station Modules "snoop" data which is sent to their Input Map Arrays:

a. Depending upon how its written, the ladder code can respond to the "Polled Responses" received by the Control Station Modules. By sending "global data" (%AQG) to the DeviceNet Master, this output data is "snooped" by all Control Station Modules on the network and is placed into their **Input Map Arrays**. For example, if one segment of the conveyor stops momentarily, the ladder code can be written such that the rest of the conveyor line is "notified" and can respond appropriately. The user can configure the Control Station Modules to receive the data in the %AQG registers of other Control Station Modules or the DeviceNet Master.

For a step-by-step explanation on how to configure the Control Station Modules (i.e.,OCS100) and to use a feature known as **Relative ID Addressing**, refer to Sections 3.2.2 and 3.2.3 in Chapter 3.

4. Up to 63 **DeviceNet Slaves** can be used in the network. There can be up to 64 total nodes including the DeviceNet Master. The slaves can be made by various manufacturers. In order to communicate with a DeviceNet Master, a unique DeviceNet Node Address (MAC ID) must be assigned to each node on the network. . Control Station Modules (i.e., OCS100) have a selectable baudrate of 125K, 250K or 500Kbaud. Thus, *all* nodes on the network must be set at the same baudrate.

5. To install upgrades or perform configuration, the user must directly upload/download programming into *each* individual Control Station device.

NOTES

APPENDIX A: OCS EXPLICIT MESSAGING

1 General

While most necessary data can be transferred quickly and easily using the Polled Messaging, situations arise where specific data need to be transferred using Explicit Messaging. It is important to note that Explicit Messaging is significantly slower than Polled Messaging, and thus should be used only for relatively minor, infrequently accessed data. Explicit Messaging should not be used for important or quickly changing data such as Emergency Stop bits, instantaneous speed values, etc.

In operation, the DeviceNet Master (Scanner) builds and transmits Explicit Message sequences to the OCS/DeviceNet Slave (OCS). The OCS then, following the receipt of an explicit command message, builds and transmits explicit response messages.

The actual building of Explicit Messages is performed by the DeviceNet Master (Scanner) through the programming language used to control it (Ladder Logic for most PLCs).

Please refer to the manuals for the DeviceNet Master (Scanner) and the programming language used for the specifics of building and sending an Explicit Messages.

For more information regarding Explicit Messages, refer to the *DeviceNet™ (ODVA) Specifications*, Vol. 1, Section 4.2.5.

2 Explicit Message Command Format

2.1 Explicit Message Packet

When sending an Explicit Message, a message packet is created in a specific format. The message packet consists of the following:

MAC ID Address of the receiving node.
Service Code Indicates the desired command (i.e., Service Code 50: Get OCS Registers).
Class ID A vendor-specific value. (Set to 100)
Instance ID Correlates to a register type depending upon the Service Code that is selected.
Service Data This field contains Sevice Code specific data. (See Section 2.3.)

2.2 Components of Explicit Message Command Format

This section covers vendor-specific information, which is used to access registers within the OCS from the master device (or scanner) when utilizing explicit messages.

Note: All numbers referenced in this manual document are in Base 10 or decimal format.

Explicit messages contain the following components:

a. **Service Code Supported -** Indicates the desired command:

Service Codes	
14	Get Attribute Single
16	Set Attribute Single
50	Get OCS Registers
51	Set OCS Registers

b. **Class** - Is a vendor-specific value. Set to 100.

c. **Instance ID -** Correlates to a register type depending upon which Service Code is selected. **(**See Table A1 and A2.)

Table A1 - Instance ID for use with Service Codes 14 and 16

INSTANCE ID	REGISTER TYPE	REGISTER SIZE	ACCESS RIGHTS	ALLOWABLE ATRIBUTE ID RANGE	REPRESENTS REGISTERS
0	%IG	Bit	NO ACCESS	0 – 255	%IG1 – 256
1	%QG	Bit	NO ACCESS	0 – 255	%QG1 – 256
2	%S	Bit	RD ONLY	0 – 255	%S1 – 256
3	%K	Bit	RD ONLY	0 – 255	%K1 – 256
4	%AIG	Word	RD ONLY	0 – 31	%AIG1 – 32
5	%AQG	Word	RD/WR	0 – 31	%AQG1 – 32
6	%SR	Word	RD ONLY	0 – 63	%SR1 – 64
7	%F	Word	NO ACCESS	N/A	N/A
8	%I	Bit	RD/WR	0 – 255	%I1 – 256
9	%Q	Bit	RD/WR	0 – 255	%Q1 – 256
10	%T	Bit	RD/WR	0 – 255	%T1 – 256
11	%M	Bit	RD/WR	0 – 255	%M1 – 256
12	%AI	Word	RD/WR	0 – 255	%AI1 – 256
13	%AQ	Word	RD/WR	0 – 255	%AQ1 – 256
14	%R	Word	RD/WR	0 – 255	%R1 – 256
15	%R	Word	RD/WR	0 – 255	%R257 – 512
16	%R	Word	RD/WR	0 – 255	%R513 – 768
17	%R	Word	RD/WR	0 – 255	%R769 – 1024
18	%R	Word	RD/WR	0 – 255	%R1025 – 1280
19	%R	Word	RD/WR	0 – 255	%R1281 – 1536
20	%R	Word	RD/WR	0 – 255	%R1537 – 1792
21	%R	Word	RD/WR	0 – 255	%R1793 – 2048

Table A2 - Instance ID for use with Service Codes 50 and 51					
INSTANCE ID	REGISTER TYPE	REGISTER SIZE	ACCESS RIGHTS	ALLOWABLE ATRIBUTE ID RANGE	REPRESENTS REGISTERS
0	%IG	Bit	NO ACCESS	0 – 255	%IG1 – 256
1	%QG	Bit	NO ACCESS	0 – 255	%QG1 – 256
2	%S	Bit	RD ONLY	0 – 255	%S1 – 256
3	%K	Bit	RD ONLY	0 – 255	%K1 – 256
4	%AIG	Word	RD ONLY	0 – 31	%AIG1 – 32
5	%AQG	Word	RD/WR	0 – 31	%AQG1 – 32
6	%SR	Word	RD ONLY	0 – 191	%SR1 – 192
7	%F	Word	NO ACCESS	N/A	N/A
8	%I	Bit	RD/WR	0 – 2047	%I1 – 2048
9	%Q	Bit	RD/WR	0 – 2047	%Q1 – 2048
10	%T	Bit	RD/WR	0 – 2047	%T1 – 2048
11	%M	Bit	RD/WR	0 – 2047	%M1 – 2048
12	%AI	Word	RD/WR	0 – 511	%AI1 – 512
13	%AQ	Word	RD/WR	0 – 511	%AQ1 – 512
14	%R	Word	RD/WR	0 – 2047	%R1 – 2048

d. **Attribute ID**

The Attribute ID points to the first (or only) OCS register number to be accessed. In the case of Service Codes 14 and 16, the Attribute ID argument is 1 byte in length. In the case of Service Codes 50 and 51, the Attribute ID argument is 2 bytes in length. See the Instance ID tables (Item c) for the available ranges of each of the OCS registers types available.

2.3 Building an Explicit Message

a. Explicit Command Byte Sequences

For Service Code 14 (Get Attribute Single), the explicit command byte sequence is:

BYTE	GENERIC TAGS	AS IMPLEMENTED
0	MAC ID	MAC ID
1	Service Code	14
2	Class ID	100
3	Instance ID	0 – 21 See Table A1
4	Attribute ID	0 – 255 See 2.2, item d

For Service Code 16 (Set Attribute Single), the explicit command byte sequence is:

BYTE	GENERIC TAGS	AS IMPLEMENTED
0	MAC ID	MAC ID
1	Service Code	16
2	Class ID	100
3	Instance ID	0 – 21 See Table A1
4	Attribute ID	0 – 255 See 2.2, item d
5	Data Byte 1	If the register type specified by the Instance ID byte is a bit type register, byte 5 will be a Boolean variable (0 or 1 value) and byte 6 does not exist. Otherwise bytes 5 and 6 will together contain a 16 bit integer variable. Byte 5 will contain the LSB and byte 6 will contain the MSB.
6	Data Byte 2	

For service code 50 (Get OCS Registers), the explicit command byte sequence is:

BYTE	FUNCTION GENERIC	FUNCTION AS IMPLEMENTED
0	MAC ID	MAC ID
1	Service Code	50
2	Class ID	100
3	Instance ID	0 – 14 See Table A2
4	Attribute ID	*Register Base LSB
5	Data Byte 1	*Register Base MSB
6	Data Byte 2	Byte Count
* The **Register Base** is the first OCS register (number – 1) to be accessed. For any bit access, the Register Base must be on a byte boundary. That is, it must be evenly divisible by 8. Example: To access registers %I49-64, put in 48 and a count of 2.		

For service code 51 (Set OCS Registers), the explicit command byte sequence is:

BYTE	FUNCTION GENERIC	FUNCTION AS IMPLEMENTED
0	MAC ID	MAC ID
1	Service Code	51
2	Class ID	100
3	Instance ID	0 – 14 See Table A2
4	Attribute ID	*Register Base LSB
5	Data Byte 1	*Register Base MSB
6	Data Byte 2	Byte Count
7 - n	Additional Data (as required)	Additional Data (as required)
* The **Register Base** is the first OCS register (number – 1) to be accessed. For any bit access, the Register Base must be on a byte boundary. That is, it must be evenly divisible by 8. Example: To access registers %Q1-16, put in 0 and a count of 2.		

Service Codes 50 and 51 only:
The **Byte Coun**t is the number of bytes of data that are to be read or written. Word registers are 2 bytes in length. Bit type registers are handled in groups of 8 bits at a time. 8 bit type registers are packed into a single byte. Bit type registers can only be accessed in groups of 8. The Register Base when referencing bit type registers must by on a byte boundary, that is it must be divisible by 8.

Note that most DeviceNet scanners handle DeviceNet Explicit Header bytes and Fragmentation bytes transparently to the user. For that reason, the message format sequences included in this document do not include these bytes in the illustrations. Consult the documentation for your specific DeviceNet scanner concerning how to handle these bytes.

2.4 Explicit Command Message Examples

a. Service Code 14 – Get Attribute Single

Function: READ the value of register %R28 from the OCS at node address 3.

BYTE	FUNCTION	VALUE
0	MAC ID	3
1	Service Code	14
2	Class ID	100
3	Instance ID	14
4	Attribute ID	27

b. Service Code 16 – Set Attribute Single

Function: WRITE the value 51 to register %AQ5 of the OCS at node address 7.

BYTE	FUNCTION	VALUE
0	MAC ID	7
1	Service Code	16
2	Class ID	100
3	Instance ID	13
4	Attribute ID	4
5	Data Byte LSB	51
6	Data Byte MSB	

c. Service Code 50 – Get OCS Registers

Function: READ the values of Registers %R28 - %R35 from the OCS at node address 22.

BYTE	FUNCTION	VALUE
0	MAC ID	22
1	Service Code	50
2	Class ID	100
3	Instance ID	14
4	* Register Base LSB	27
5	* Register Base MSB	
6	Byte Count	16
*The **Register Base** is the first OCS register (number – 1) to be accessed. For any bit access, the Register Base must be on a byte boundary. That is, it must be evenly divisible by 8. Example: To access registers %I49-64, put in 48 and a count of 2.		

d. Service Code 51 – Set OCS Registers

Function: WRITE the values 15, 82 and 2153 to registers %AQ10 through %AQ12 of the OCS at node address 7.

BYTE	FUNCTION	VALUE(DECIMAL)
0	MAC ID	7
1	Service Code	51
2	Class ID	100
3	Instance ID	13
4	* Register Base LSB	9
5	* Register Base MSB	
6	Byte Count	6
8	Data 1 LSB	15
9	Data 1 MSB	
10	Data 2 LSB	82
11	Data 2 MSB	
12	Data 3 LSB	2153
13	Data 3 MSB	
* The **Register Base** is the first OCS register (number – 1) to be accessed. For any bit access, the Register Base must be on a byte boundary. That is, it must be evenly divisible by 8. Example: To access registers %Q1-16, put in 0 and a count of 2.		

2.5 Explicit Response Message Format

Upon receiving a command message from the master, the OCS will respond with an explicit response message using the following response message format:

BYTE	CONTENTS
0	**MAC ID** DeviceNet node address of the responding OCS.
1	**Service Code** Is identical to the Service Code that was sent to the OCS in the command message except bit 7 is set. This indicates that this is a Response Message. If an error was detected by the OCS, this byte contains the value 148 (94 Hex).
2 - n	**Returned Data (as required)** If the requested service was a command to “Read OCS Registers”, the Returned Data byte(s) will contain the requested OCS register values. If the requested service was a command to “Write OCS Registers”, the Returned Data byte(s) will not be present. If an error was detected by the OCS, as indicated by a value of 148 (94 Hex) in the Service Code byte, the first two Returned Data bytes will contain the error code indicating he problem. Byte #2 will indicate the General Error Code (see table below), and byte #3 will contain the additional error code.

NOTE: The programmer's should insure that the response data buffer is of sufficient size to handle the largest possible response message.

APPENDIX A

The following table defines the General Error Codes (Byte 2) as defined by DeviceNet.

Table A3 - General Error Codes (Byte 2)

ERROR CODE (IN HEX)	NAME	DESCRIPTION OF ERROR
00 – 01	Reserved	
02	Resource Unavailable	Resource needed for the object to perform the requested service were not available
03 – 07	Reserved	
08	Service Not Supported	The requested service was not implemented or not defined for this Object Class/Instance
09	Invalid Attribute Value	Invalid attribute data detected
0A	Reserved	
0B	Already in requested mode	The object is already in the requested mode or state requested by the service
0C	Object State Conflict	The object can not perform the requested service in it's current mode or state
0D	Reserved	
0E	Attribute Not Settable	A request to modify a non-modifiable attribute was received
0F	Privilege Violation	Permission/privilege check failed
10	Device State Conflict	The device's current mode or state prohibits the requested service
11	Reply Data Too Large	The data to be transmitted is large than the allocated response buffer
12	Reserved	
13	Not Enough Data	The service did not supply enough data to perform the requested service
14	Attribute Not Supported	the attribute specified in the request is not supported
15	Too Much Data	The serve supplied more data than was expected
16	Object Does not Exist	The specified object does not exist in the device
17	Reserved	
18	No stored attribute data	The attribute data of the object was not stored prior to the requested service
19	store operation failure	The attribute data of this object was not saved by the object
1A – 1E		Reserved by DeviceNet
1F	Vendor Specific error	A vendor-defined error has occurred. See the Additional Byte for further information
20	invalid parameter	A parameter associated with the service was determined to be not valid
21 – CF	Reserved	
D0 – FF	Object Class and Service Errors	Vendor-specific Object and Class errors.

2.6 Additional Error Codes (Byte 3)

When the General Error Code is a 1F Hex, a vendor specific error message is being sent. In this case, the following table contains the additional error codes as defined for the OCS.

Table A4 – Additional Error Codes (Byte 3)

BYTE 3	DESCRIPTION OF ERROR
1	The register type is invalid
2	The byte count is invalid
3	The requested registers result in an overflow past the end of the register bank
4	The command length is invalid
5	The OCS is busy ((*) see below description below)
6	The response from the OCS ladder engine is invalid
7	Attempted to write to a read only or a no access register or attempted to read a no access register
8	Attempted to access a bit register with an improper register base value. Must be on a byte boundary (8n + 1) (Service Codes 50 & 51 only).
9	The OCS is not in the RUN mode.
(*) Note: Within the OCS, the explicit messaging of items in Class ID of 100 share a command processor with the OCS's RS-232 serial port. In the situation where the serial port is connected to a PC running Cscape, there exists the possibility that the command processor may be busy processing data to/from the serial port when a DeviceNet explicit request is received from the DeviceNet master. This condition is especially apparent if Cscape is running a "Remote Text Window", a "Data Watch Window" or is uploading or downloading to/from the OCS. If this error message is received, simply re-send the same command buffer to the OCS again. On the second request the command message will most likely be processed promptly.	

INDEX