How to configure and connect the Alpha® Gateway Interface to a DeviceNet network

Contents

Technical support

For additional information, programming examples, and other data that may be available for your specific network configuration, check the Adaptive® website *http://www.adaptivedisplays.com*.

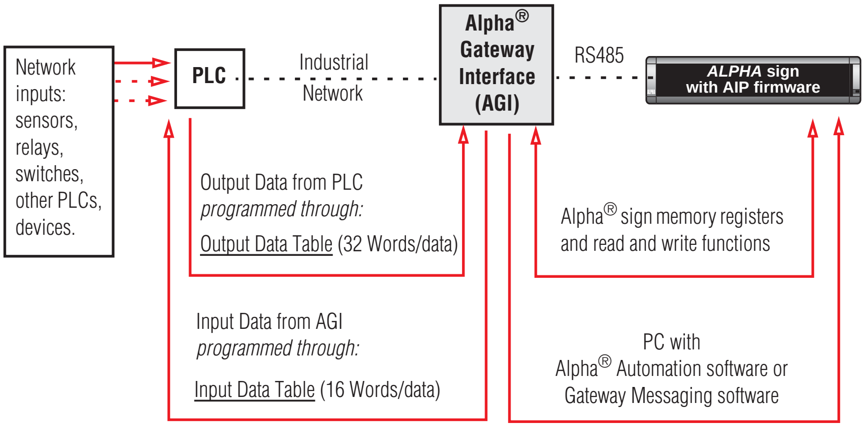
Introduction

The Alpha® Gateway Interface (the “AGI”) allows for the exchange of data between a DeviceNet PLC and Alpha® signs to display messages and real-time data on a system.

This document outlines preliminary connection requirements for setting up an AGI to act as an interface between a DeviceNet industrial network and an Alpha® sign network.

Below is a high-level diagram of how the AGI works:

NOTE: In the event of a communication failure between any two points of the information pathway in the diagram, messages may fail to display on a sign.



Network inputs: sensors, relays, switches, other PLCs, devices.
PLC
Industrial Network
Alpha® Gateway Interface (AGI)
RS485
ALPHA sign with AIP firmware
Output Data from PLC programmed through: Output Data Table (32 Words/data)
Input Data from AGI programmed through: Input Data Table (16 Words/data)
Alpha® sign memory registers and read and write functions
PC with Alpha® Automation software or Gateway Messaging software

INFORMATION FLOW — In a DeviceNet network, a “device” is any point in the information pathway capable of sending or receiving a data signal. In the most basic network configuration, above (one input, one PLC, one AGI, one sign), the PLC, AGI, and sign are all capable of both sending and receiving data signals.

Figure 1: Information flow

Adaptive Micro Systems • 7840 North 86th Street • Milwaukee, WI 53224 USA • 414-357-2020 • 414-357-2029 (fax) • http://www.adaptivedisplays.com

Due to continuing product innovation, specifications in this manual are subject to change without notice.

Basic setup procedure

Follow these basic steps to set up the AGI in an Alpha® sign and DeviceNet industrial network:

1. Create sign messages using a computer with the Alpha® Automation software or the Gateway Messaging software. (Refer to the software's online help.)
2. Connect the computer, Alpha® sign(s), and, if needed, the AGI using the "Interconnection diagram" on page 5.
3. Download the sign messages to one or all of the signs on the network using the messaging software.
4. Determine the network baud rate and node address for the AGI.
5. Program the PLC's DeviceNet Scanner Module for the Input and Output data tables required for the AGI.
6. Program the PLC to trigger messages and update variables. (See "Configuration data for Alpha® Gateway networks" on page 23.)

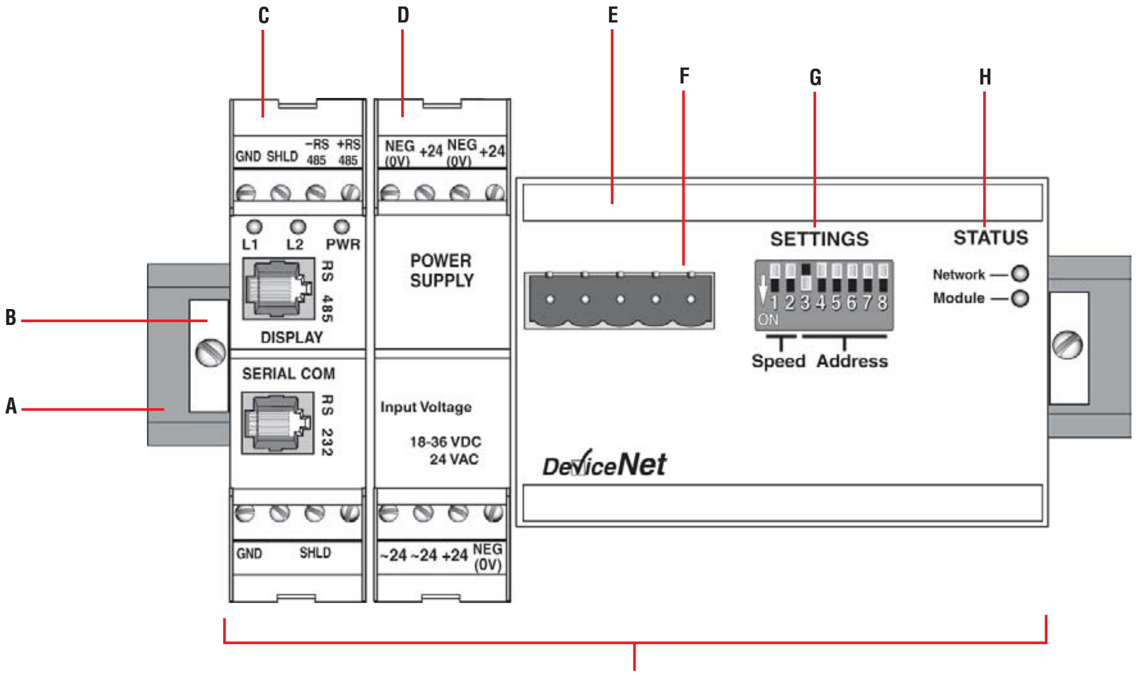
Related documents

Most of these documents are available on the Adaptive® web site — *http://www.adaptivedisplays.com*:

Document name	Part number	Description
Alpha® Automation software online help system	—	Describes how to use the Alpha® Automation software to compose and display messages on Alpha® signs.
Networking Alpha® Signs	9700-0112	Describes wired, modem, and wireless Alpha® sign networks.
Alpha® Industrial Protocol (AIP)	9711-8814	Describes how the Modbus protocol is incorporated into the Alpha® industrial protocol sign firmware. This firmware allows signs to connect to a Modbus communications network.

Alpha® Gateway Interface

The Alpha® Gateway Interface is a 3-module unit: (1) The Network Interface Module is the interface between a PLC and the Alpha® network. (2) The Power Supply Module supplies power for both the CPU Module and the Network Interface Module. (3) The CPU Module is the interface between the Network Interface Module and Alpha® signs. For more information, see "Technical specifications" on page 18.



The 3-modules can be arranged in either one of two ways:

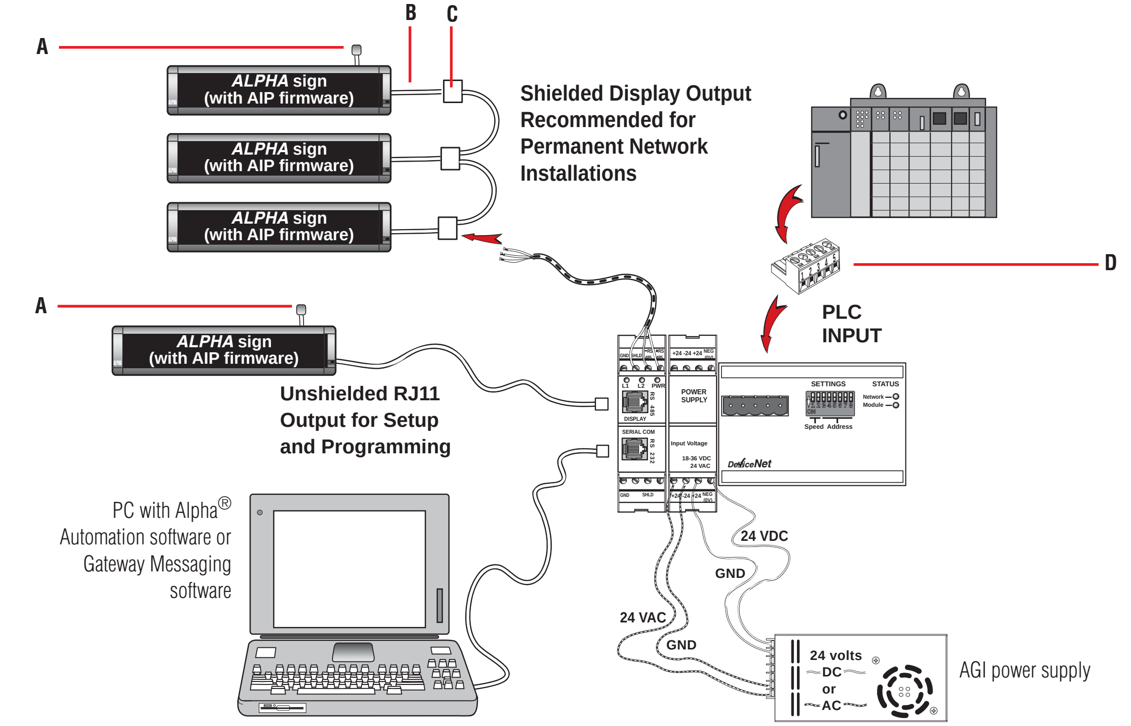
CPU Module — Power Supply Module — Network Interface Module
or
Power Supply Module — CPU Module — Network Interface Module

Item	Part number		Name	Description
A	—		DIN rail	The rail should be connected to earth ground because the rail serves as chassis for all the modules.
B	—		End clamp	Use these to hold the modules in place.
C	1188-9203	E9920-1116	CPU Module	RS232 input. RS485 output (hardwired or standard RJ11 jack).
D		E9920-1113	Power Supply Module	Accepts 24 VAC or 18 to 24 VDC power input.
E		E9920-1115 (see NOTE 1 below)	Network Interface Module	Connects Alpha® sign network to PC and provides an interface between the DeviceNet PLC and an Alpha® sign network.
F	—		Combicon 5-lead connector	Connecting point to the PLC network

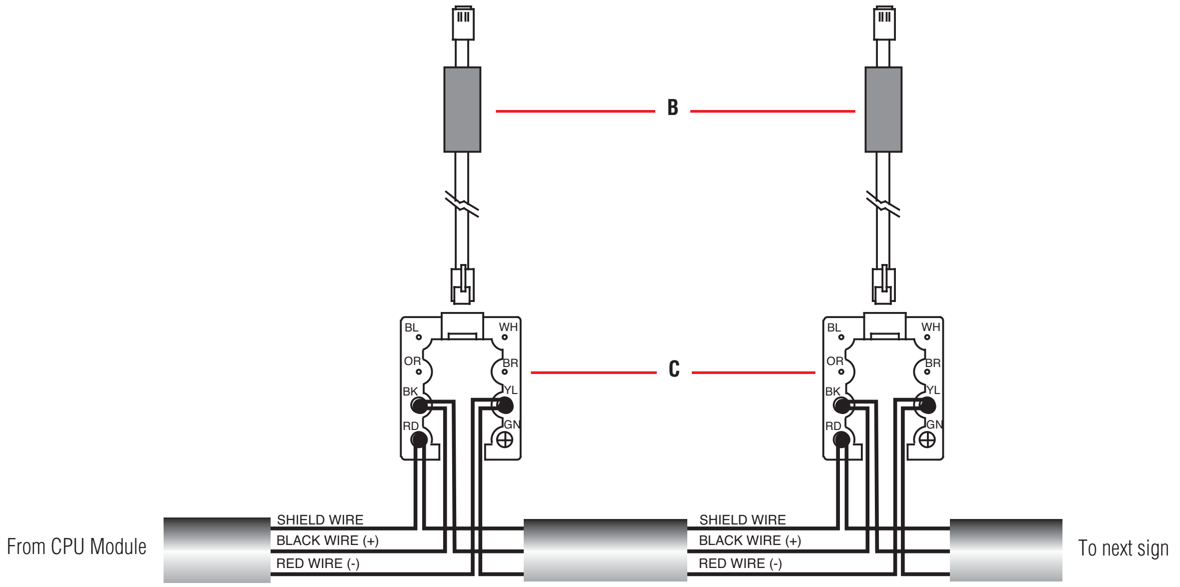
<table>
<tr>
<td rowspan="2"><b>G</b></td>
<td rowspan="2">—</td>
<td rowspan="2">SETTINGS (see NOTE 2 below)</td>
<td>Speed</td>
<td>DIP switches 1 and 2 set DeviceNet baud rate:
<table>
<tr><th colspan="2">DIP switches (0 = OFF, 1 = ON)</th><th colspan="2" rowspan="2">Baud rate</th></tr>
<tr><th>1</th><th>2</th></tr>
<tr><td>0</td><td>0</td><td>125 kb</td><td rowspan="4">Longer cable connections may not support faster baud rates. See “Cabling reference” on page 17.</td></tr>
<tr><td>0</td><td>1</td><td>250 kb</td></tr>
<tr><td>1</td><td>0</td><td>500 kb</td></tr>
<tr><td>1</td><td>1</td><td>reserved</td></tr>
</table>
</td>
</tr>
<tr>
<td>Address</td>
<td>DIP switches 3 through 8 set the binary-encoded network node address:
<table>
<tr><th colspan="6">DIP switches (0 = OFF, 1 = ON)</th><th rowspan="2">Network node address</th></tr>
<tr><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td></td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>62 (see NOTE 3 below)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>63 (see NOTE 3 below)</td></tr>
</table>
</td>
</tr>
<tr>
<td rowspan="2"><b>H</b></td>
<td rowspan="2">—</td>
<td rowspan="2">STATUS</td>
<td>Network</td>
<td>The <b>Network</b> LED provides the following information:
<table>
<tr><th>LED</th><th>Meaning</th><th>Suggested action</th></tr>
<tr><td>Off</td><td>Offline or no power</td><td>• Apply power to unit or to DeviceNet network.</td></tr>
<tr><td>Steady green</td><td>Ok</td><td>• No action required</td></tr>
<tr><td>Flashing green</td><td>No communication</td><td>• Check 2-digit error code on PLC.
• Configure scan list table to add device.</td></tr>
<tr><td>Flashing red</td><td>At least one device in scan list has failed.</td><td>• Check failed device and the scan list table.</td></tr>
<tr><td>Steady red</td><td>Communication failure</td><td>• Reset Network Interface Module.
• Replace module if necessary.</td></tr>
</table>
</td>
</tr>
<tr>
<td>Module</td>
<td>The <b>Module</b> LED provides the following information:
<table>
<tr><th>LED</th><th>Meaning</th><th>Suggested action</th></tr>
<tr><td>Off</td><td>No power</td><td>• Apply power to unit.</td></tr>
<tr><td>Green</td><td>Ok</td><td>• No action required</td></tr>
<tr><td>Flashing red</td><td>Invalid configuration</td><td>• Check configuration setup with the software that was used for initialization.
• Check physical connections.</td></tr>
<tr><td>Steady red</td><td>Fatal fault</td><td>• Check physical connections.
• Look for evidence of electrical damage.
• Replace module.</td></tr>
</table>
</td>
</tr>
<tr>
<td colspan="5">NOTES:
(1) When ordering a Network Interface Module, the type of network (DeviceNet, Profibus DP, or Modbus Plus) must be specified.
(2) To change DIP switch settings, turn the unit off, change the DIP switches, then turn the unit on again. Do NOT change DIP switches when the unit is on.
(3) Network node addresses 62 and 63 may be reserved for other devices on some networks.</td>
</tr>
</table>

Interconnection diagram

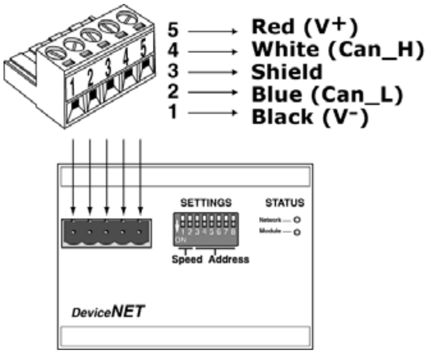
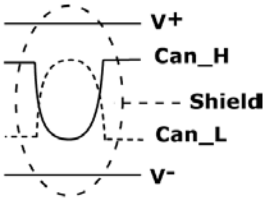
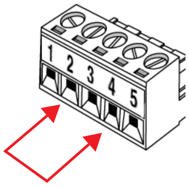
Overview



Modular Network Adapter wiring



Item	Part number	Name	Description
A	1088-9107	End-Of-Line (EOL) terminator	Termination is required for the last (or only) sign in a network. For more information, see **Networking Alpha® Signs**.
B	1088-8636	RS485 cable	1-foot RS485 cable that connects a Module Network Adapter to an Alpha® sign
C	4331-0602	Modular Network Adapter	Used to connect multiple Alpha® signs together.

D	—	PLC input	Pin-out diagram for DeviceNet “combicon” style 5-lead connector that is supplied with the Network Interface Module.  1 2 3 4 5 5 Red (V+) 4 White (Can_H) 3 Shield 2 Blue (Can_L) 1 Black (V-) SETTINGS STATUS Network Module ON 1 2 3 4 5 6 7 8 Speed Address DeviceNET External 12 VDC power supply required: V^+ — supply (24 VDC) V^- — bus common (0 VDC) Can_H — high differential data connection Can_L — low differential data connection  Connect 121-ohm terminating resistor across contacts 2 and 4 to attenuate data transfer signal. Connect a second 121-ohm resistor at the PLC (opposite end of the cable). See “Network termination test” on page 16 for a field test procedure to check impedance values in your network. 

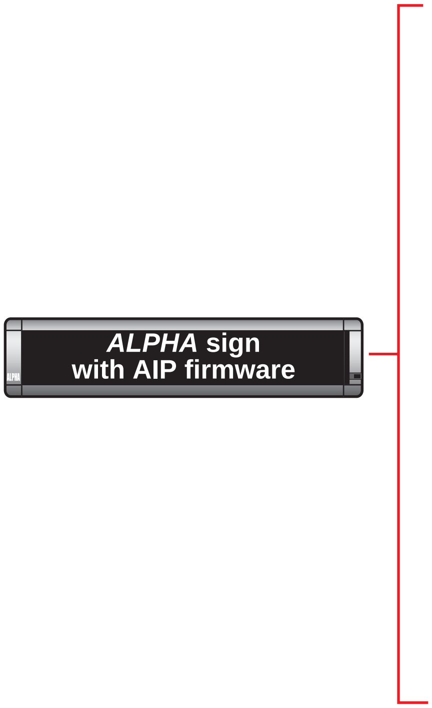
Network/Gateway data pathway

The Alpha® Gateway Interface allows for the exchange of data between a PLC and Alpha® sign(s) to show messages and real-time data on a system. The AGI is connected to Alpha® signs via a multi-drop (RS485) network which will support up to 32 drops before requiring a repeater. Alpha® signs can be assigned addresses from 001 to 254 by using Alpha® Automation software.

Messages are created using Alpha® Automation software or Gateway Messaging software and then stored in each sign's memory. An Alpha® sign can store up to 4000 messages (1 - 4000) and can support up to 100 (1 - 100) variables.

Memory map

The following table shows an Alpha® sign's memory allocations and functions:



Register #	Type	Access	Description
001	Variables	Read / Write	Variable #1 (± integer value)
002			Variable #2 (± integer value)
003			Variable #3 (± integer value)
. . .			. . .
100			Variable #100 (± integer value)
101	Message processing	Write only	Message Activation Register (message # to turn on)
102			Message Deactivation Register (message # to turn off)
103	Message queue	Read/Write	All currently running messages
104		Read only	
. . .			
167			

Message #	Access	Description
1	Write during message download only	Message #1
2		Message #2
3		Message #3
. . .		. . .
4000 (see NOTE)		Message 4000

NOTE:
The size of a message determines how many messages can be stored in an Alpha® sign. For example, a sign can store up to 4000 messages if the average message length equals 50 characters. On the other hand, a sign could only store 2000 messages if the average message length equals 100 characters

DeviceNet EDS file

Each device on a DeviceNet network is associated with an *.eds file (short for "electronic data sheet") which is installed by the Alpha® Automation software. This file contains all of the parameters, such as baud rate and table formats, along with all of the other necessary data required by the Scanner Module when the AGI is configured on a DeviceNet network. When loading the EDS file to your system, the input and output parameters are set to the following:

- 32 words (64 bytes) out for the PLC Output Data table from the PLC to the AGI
- 16 words (32 bytes) in for the PLC Input Data table from the AGI to the PLC

The following section explains how to load the required EDS file.

DeviceNet required initialization sequence

In order to add an AGI to a DeviceNet network, you must have an intermediate communication interface (such as an Allen-Bradley 1772-KFD) which will allow the DeviceNet PLC and the AGI to exchange data. Because DeviceNet is an "open protocol", there is a range of available options from Allen-Bradley and from third-party vendors.

Whatever type of intermediate communication interface is selected, there are three, basic steps for initialization on a DeviceNet network:

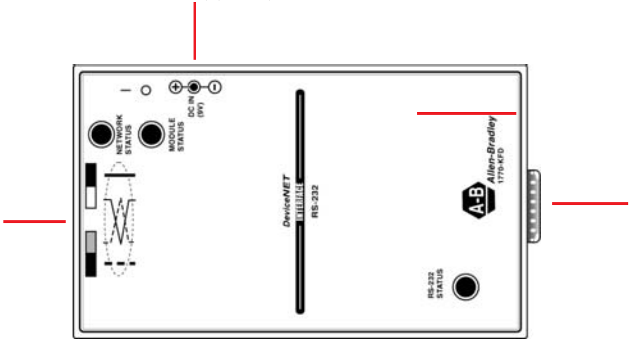
- First, verify connections for the AGI to the network as well as the Network Interface Module settings.
- Second, transfer AGI properties to the initialization software and install the EDS file using your specific software application.
- Third, map I/O data tables from the Scanner Module to the Network Interface Module.

NOTE: While the specific instructions may differ depending upon the DeviceNet PLC and the communication interface selected, the following steps are essentially the same.

Verify all hard-wired connections and the Network Interface Module data rate and node address settings.

1. For AGI connection information, see "Interconnection diagram" on page 5.
2. For Network Interface Module data rate and node address settings, see "Alpha® Gateway Interface" on page 3.
3. Follow the manufacturer's instructions to connect the communication interface between the AGI and the network. As an example, the Allen-Bradley 1772-KFD is connected as shown below:

9-volt external power supply connection (optional)

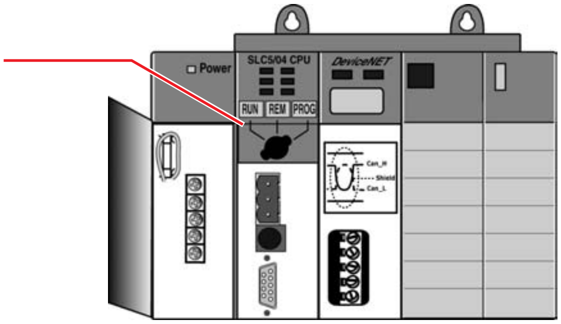


Combicon 5-pin connector (female)

DB9 RS232 input from PC

Figure 2: Allen-Bradley 1772-KFD

4. Complete any other required configuration to allow for software initialization. For example, an Allen-Bradley SLC has a “mode” switch which must be turned to the *PROGRAM* position to program the Scanner Module:

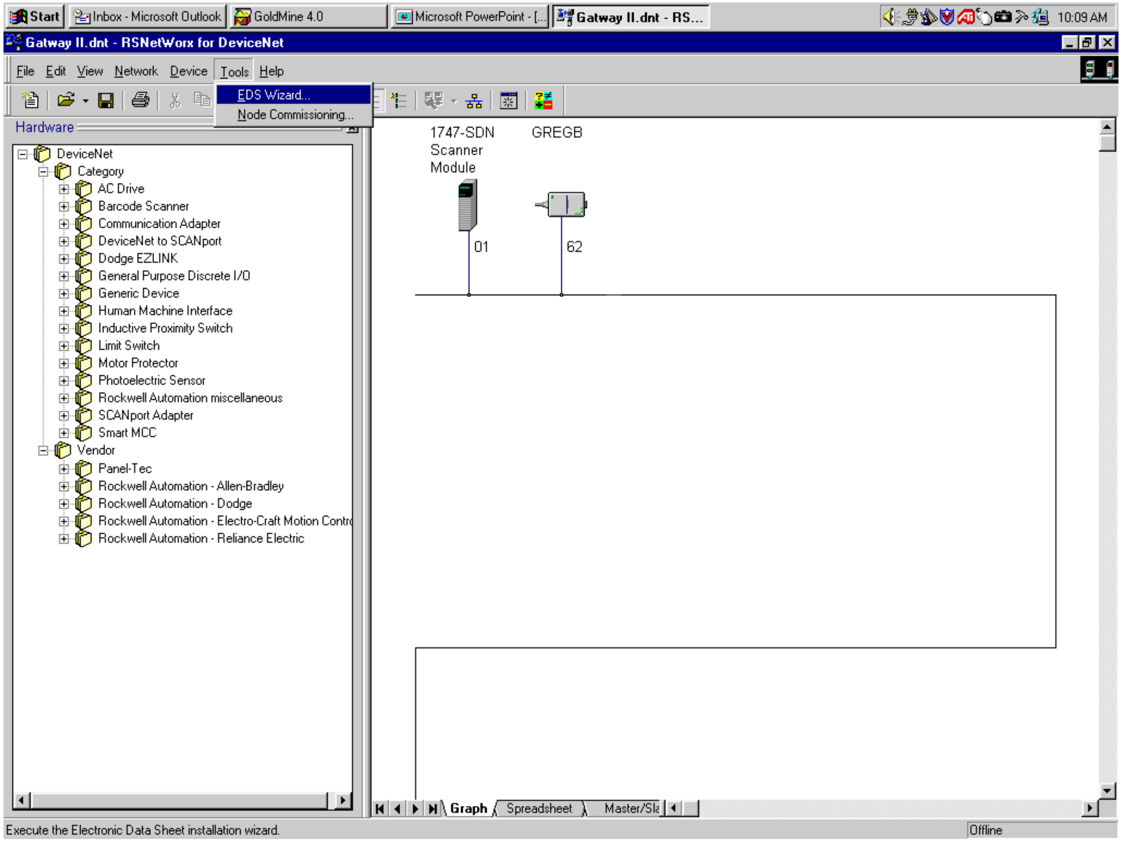


This PLC must be set to *PROGRAM* mode.

Figure 3: Allen-Bradley SLC

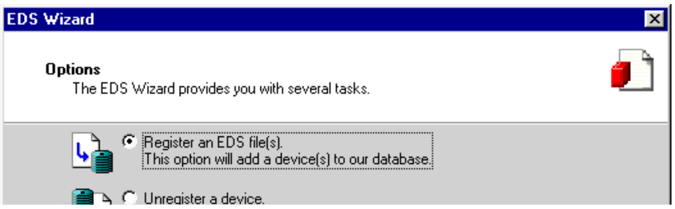
Transfer AGI properties to the initialization software and install the EDS file using your specific software application.

1. As an example, Allen-Bradley’s *RSNetWorx* application is used to start the initialization sequence: Select the *EDS Wizard* from the *Tools* pull-down menu. The *EDS Wizard* is used to load Gateway properties into *RSNetWorx*:



Execute the Electronic Data Sheet installation wizard.

2. The specific software application used for this purpose may differ, but the initialization sequence should be similar. In the *RSNetWorx* example, the *EDS Wizard* brings up several options. Select *Register an EDS file(s)*:



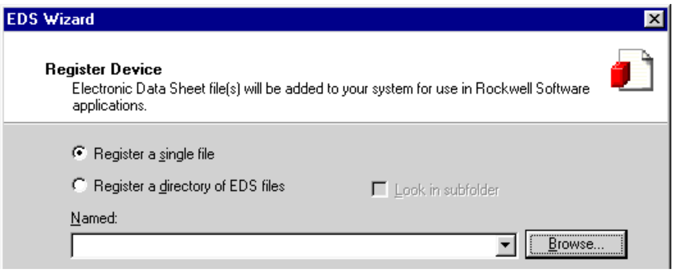
EDS Wizard

Options
The EDS Wizard provides you with several tasks.

● Register an EDS file(s).
This option will add a device(s) to our database.

○ Unregister a device.

3. Next, select *Register a single file* and select the ABS.eds file by browsing to C:\Program Files\Adaptive Micro Systems\Alpha Automation Software\Alpha Gateway Interface Files. Then click *Next*. Each of the four screens that follow to load the file configuration in the directory. However, you may also choose to custom configure the installation by selecting other options available through the *EDS Wizard*, which are presented to you in the screens that follow.



EDS Wizard

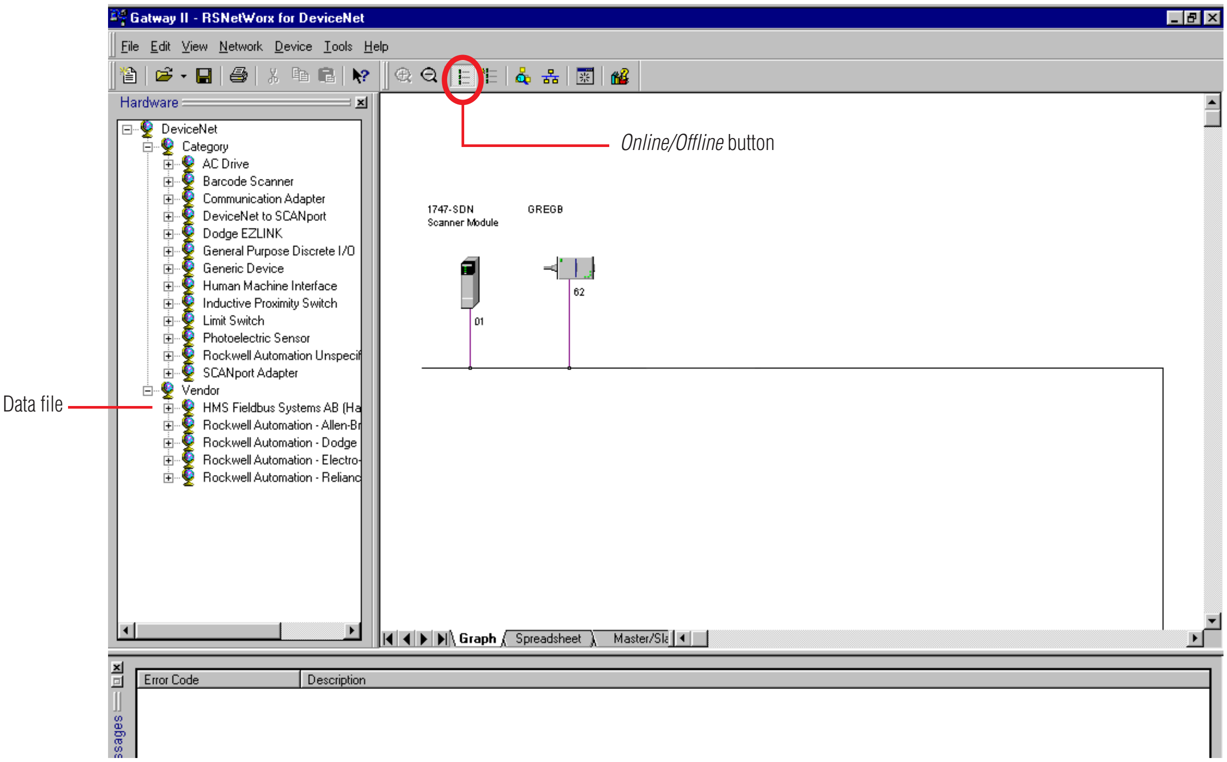
Register Device
Electronic Data Sheet file(s) will be added to your system for use in Rockwell Software applications.

● Register a single file

○ Register a directory of EDS files □ Look in subfolder

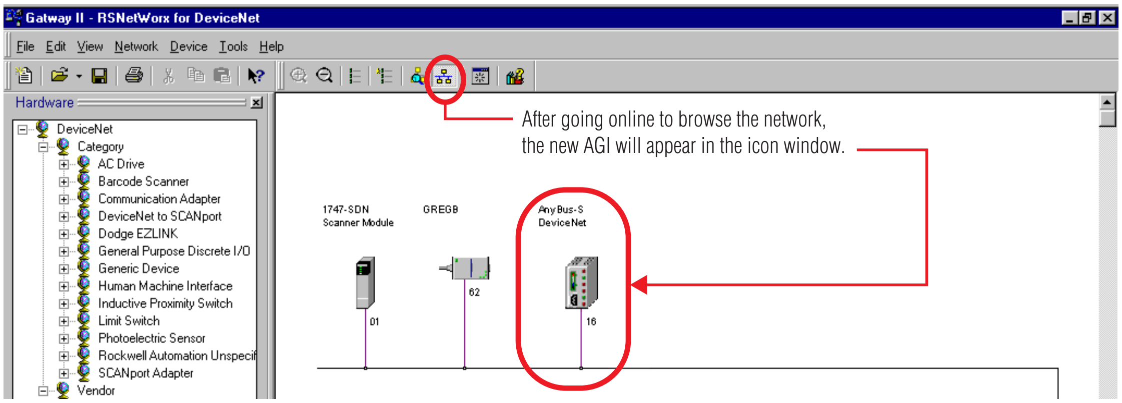
Named:

4. Check to make sure that the information has been transferred by going to the *View* pull-down menu and selecting *Hardware* from the toolbar. You should see a screen that identifies the new EDS data file. The icon for the Alpha® Gateway network device may not appear in the window on the right, however. Click the *Online/Offline* button to the online (button depressed) position:



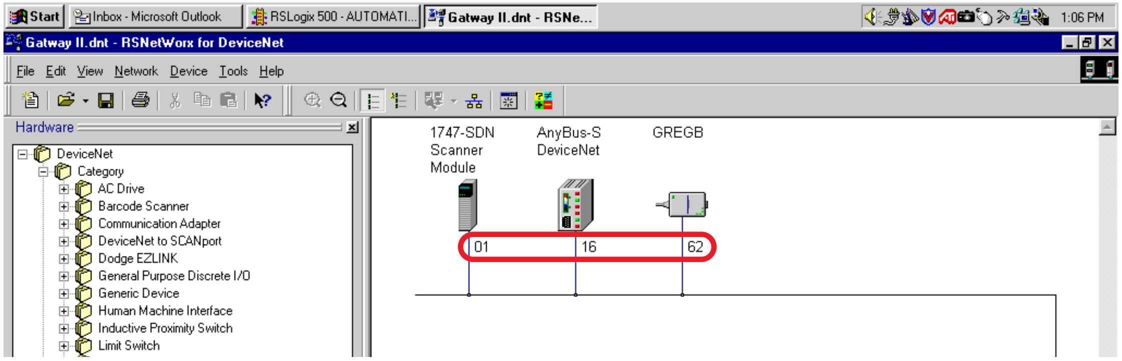
Gatway II - RSNetWorx for DeviceNet

5. Once placed into online mode, the software will browse the network to identify all devices:



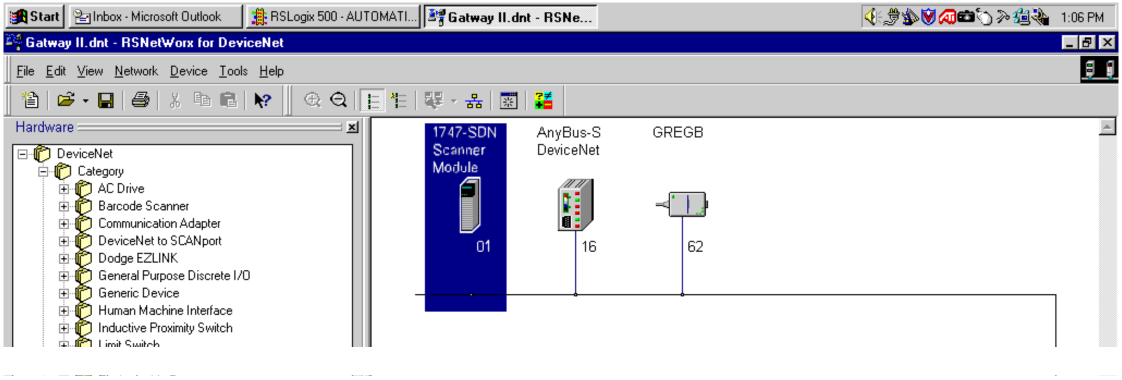
After going online to browse the network, the new AGI will appear in the icon window.

6. Click the **Refresh** button (F5) to put the network devices into numerical order:

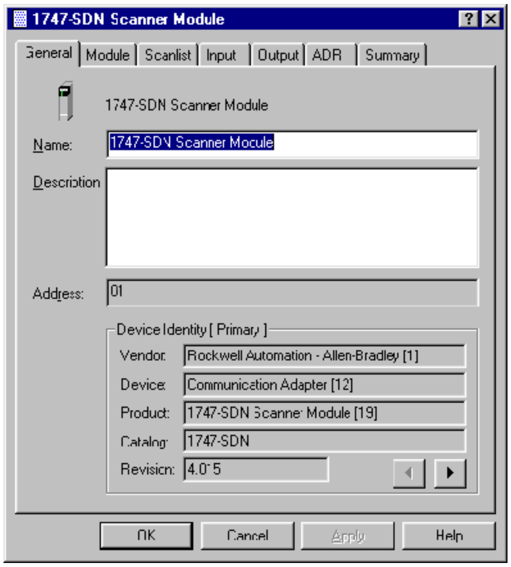


Map I/O data tables from the Scanner Module to the Network Interface Module:

1. Place the cursor on the Scanner Module and double click.



2. The Scanner Module properties box will appear. Verify the device name and address. Then click on the *Scanlist* tab:


1747-SDN Scanner Module
General | Module | Scanlist | Input | Output | ADR | Summary
OK
Cancel
Apply
Help

1747-SDN Scanner Module

Name: 1747-SDN Scanner Module

Description:

Address: 01

Device Identity [Primary]

Vendor: Rockwell Automation - Allen-Bradley [1]

Device: Communication Adapter [12]

Product: 1747-SDN Scanner Module [19]

Catalog: 1747-SDN

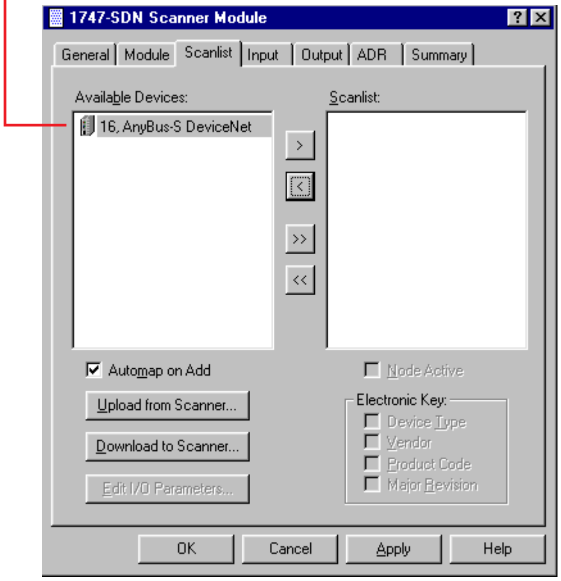
Revision: 4.015

Verify device name and address

3. At the prompt, select *Download*. The following will appear:

(1) Select the AGI device

(2) Move the device to the *Scanlist*.



Available Devices: 16, AnyBus-S DeviceNet

Scanlist:

☑ Automap on Add

☐ Node Active

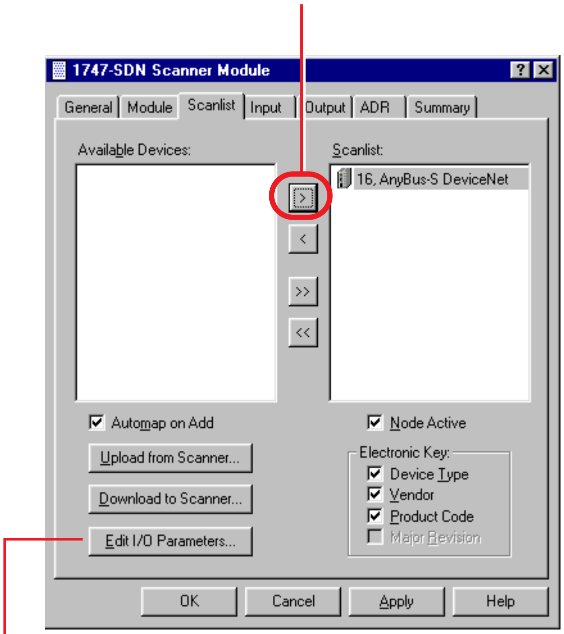
Electronic Key:

☐ Device Type

☐ Vendor

☐ Product Code

☐ Major Revision



Available Devices:

Scanlist: 16, AnyBus-S DeviceNet

☑ Automap on Add

☑ Node Active

Electronic Key:

☑ Device Type

☑ Vendor

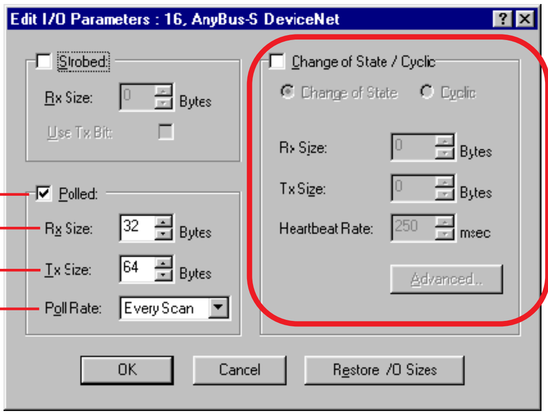
☑ Product Code

☐ Major Revision

(3) Select *Edit I/O Parameters...*

4. After selecting *Edit I/O Parameters...*, the following screen will appear:

Recommended defaults for Gateway network data exchange are shown.



Edit I/O Parameters : 16, AnyBus-S DeviceNet

□ Strobed:
Rx Size: 0 Bytes
Use Tx Bit: □

☑ Polled:
Rx Size: 32 Bytes
Tx Size: 64 Bytes
Poll Rate: Every Scan

□ Change of State / Cyclic
● Change of State ○ Cyclic
Rx Size: 0 Bytes
Tx Size: 0 Bytes
Heartbeat Rate: 250 msec

Polled = checked

Rx Size = 32 bytes in

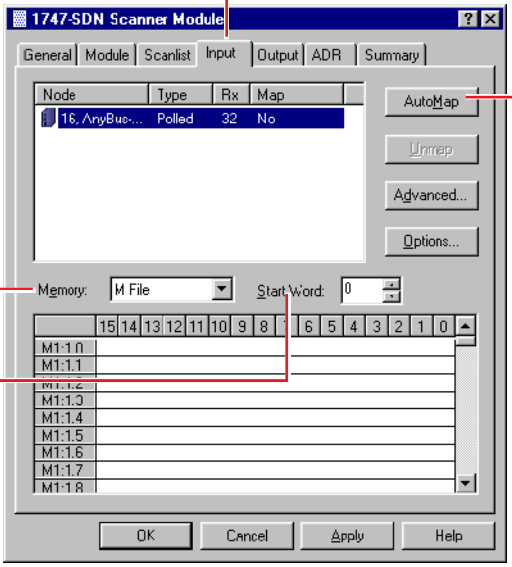
Tx Size = 64 bytes out

Poll Rate = Every Scan

Depending on specific factors in your network environment, as long as the transmit and receive data parameters are set to “32 Bytes In” and “64 Bytes Out, you could also choose “Change of State” I/O parameters (data exchanged only after a PLC change state), or “Cyclic” *(*data exchanged at regular, periodic intervals).

5. After clicking *OK*, the following screen will appear:

(1) Select the *Input* tab to begin mapping the data exchange pathway.



1747-SDN Scanner Module

Node	Type	Rx	Map
16, AnyBus-...	Polled	32	No

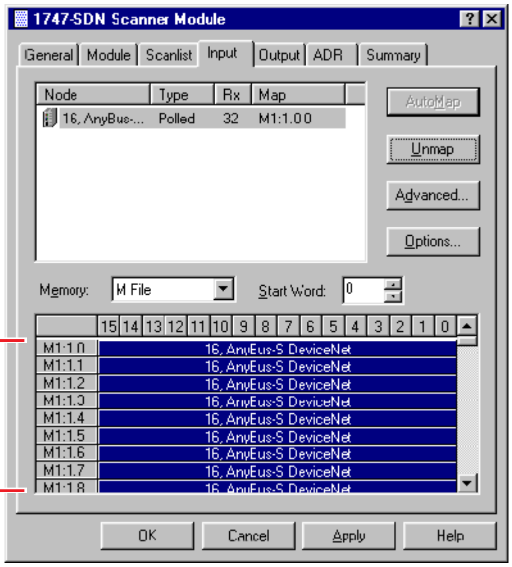
Memory: M File
Start Word: 0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1:1.0																
M1:1.1																
M1:1.2																
M1:1.3																
M1:1.4																
M1:1.5																
M1:1.6																
M1:1.7																
M1:1.8																

(2) Verify that *Memory* is set correctly. The SLC PLC shown in this example uses “M” file setting. Verify *your* usage. For example, ControlLogix PLCs use the “Assembly Data” memory setting.

(3) Select the *Start Word*. The value for this will depend on current system configuration and what is already in use by other devices.

(4) Click *AutoMap* button and the selected pathway is automatically mapped.



1747-SDN Scanner Module

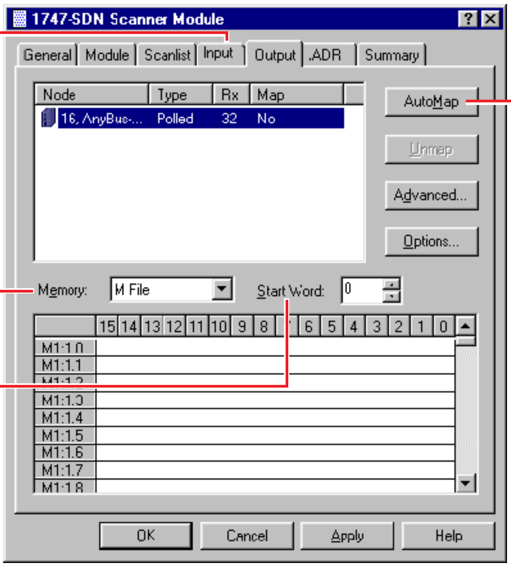
Node	Type	Rx	Map
16, AnyBus-...	Polled	32	M1:1.00

Memory: M File
Start Word: 0

	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
M1:1.0	16, AnyBus-S DeviceNet
M1:1.1	16, AnyBus-S DeviceNet
M1:1.2	16, AnyBus-S DeviceNet
M1:1.3	16, AnyBus-S DeviceNet
M1:1.4	16, AnyBus-S DeviceNet
M1:1.5	16, AnyBus-S DeviceNet
M1:1.6	16, AnyBus-S DeviceNet
M1:1.7	16, AnyBus-S DeviceNet
M1:1.8	16, AnyBus-S DeviceNet

(5) As the pathway is defined, highlighted files are listed, as shown in the screen below.

6. Select the *Output* tab and repeat the previous procedure:



Node	Type	Rx	Map
16, AnyBus-...	Polled	32	No

Memory: M File

Start Word: 0

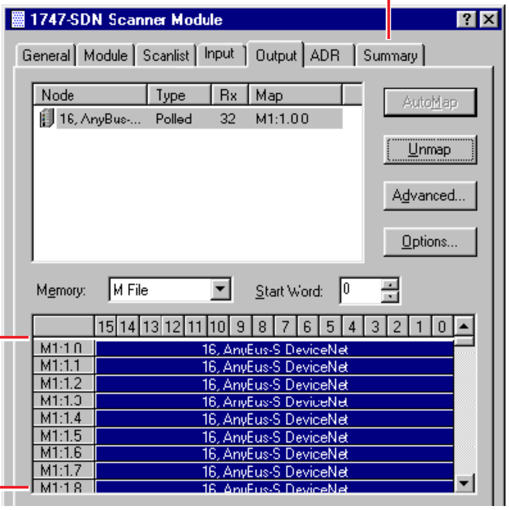
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1:1.0																
M1:1.1																
M1:1.2																
M1:1.3																
M1:1.4																
M1:1.5																
M1:1.6																
M1:1.7																
M1:1.8																

(1) Select the *Output* tab to begin mapping the data exchange pathway.

(2) Verify that *Memory* is set correctly. The SLC PLC shown in this example uses "M" file setting. Verify *your* usage. For example, ControlLogix PLCs use the "Assembly Data" memory setting.

(3) Select the *Start Word*. The value for this will depend on current system configuration and what is already in use by other devices.

(4) Click *AutoMap* button and the selected pathway is automatically mapped.



Node	Type	Rx	Map
16, AnyBus-...	Polled	32	M1:1.0.0

Memory: M File

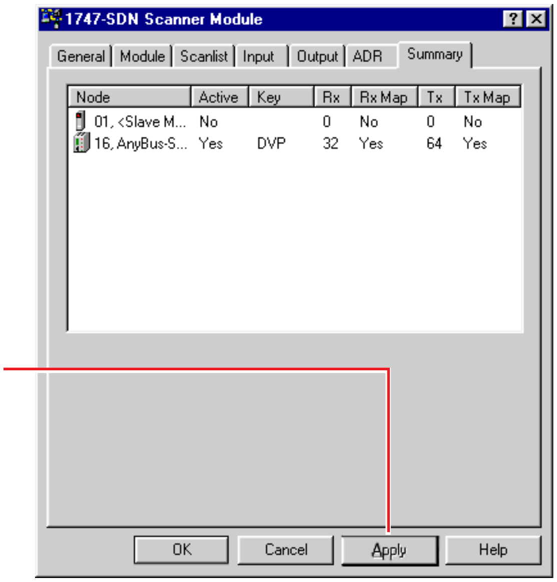
Start Word: 0

	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
M1:1.0	16, AnyBus-S DeviceNet
M1:1.1	16, AnyBus-S DeviceNet
M1:1.2	16, AnyBus-S DeviceNet
M1:1.3	16, AnyBus-S DeviceNet
M1:1.4	16, AnyBus-S DeviceNet
M1:1.5	16, AnyBus-S DeviceNet
M1:1.6	16, AnyBus-S DeviceNet
M1:1.7	16, AnyBus-S DeviceNet
M1:1.8	16, AnyBus-S DeviceNet

(6) After mapping the pathways, select the *Summary* tab.

(5) As the pathway is defined, highlighted files are listed, as shown in the screen below.

7. After selecting the *Summary* tab, verify that mapping has been correctly defined:

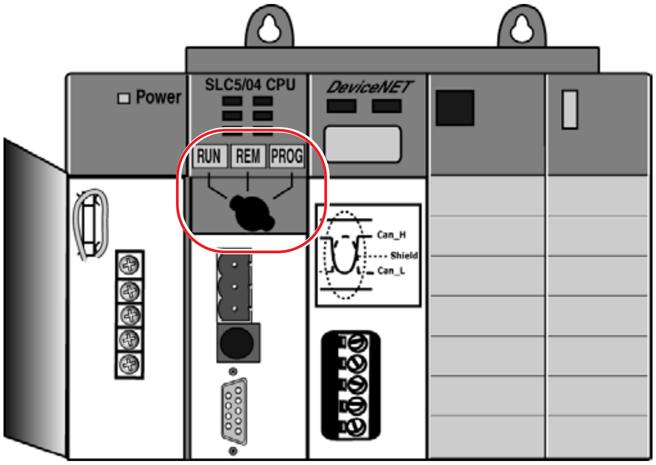


Node	Active	Key	Rx	Rx Map	Tx	Tx Map
01, <Slave M...	No		0	No	0	No
16, AnyBus-S...	Yes	DVP	32	Yes	64	Yes

After verifying mapping, select *Apply*.

Then download all information to the scanner module.

8. In this example, turn the PLC switch back to *Run* to start using the network for messaging control:



Troubleshooting and testing

Troubleshooting

When an Alpha® sign is successfully connected to a DeviceNet network, there should *always* be some type of message on the sign. If there isn't, use this table to troubleshoot:

Problem	Possible causes
No message appears on a sign	• Network wiring fault • PLC fault • Sign hardware failure • PLC trying to display a message that was not programmed into the sign • Message(s) too long for preset file size • Message is just composed of a "space" character • Device not plugged in or switched on
"No Network Activity" appears on sign	• Network wiring fault • PLC fault • Sign hardware failure • Sign timeout — no network activity for at least 3 seconds • AGI offline/connected to RS232 line
"NO BACKGROUND MESSAGE" appears on sign[1]	• Sign address not correct • Sign has not received any message to display (not an error condition) • Sign has not received any valid serial data. • Information is being sent but it is not for this sign or is invalid.
Message error — a specific message number is displayed (for example "Message #0024")	• Blank message — either this message was never edited (and never downloaded to the sign) or • Message is invalid — there were syntax errors in the message which prevented it from being downloaded to the sign.

[1] This is called the "background" message. Alpha® Automation software or Gateway Messaging software can be used to change the wording of this message.

Network termination test

Termination is used to match the impedance values between the network node and the transmission line being used. If the values are not balanced, transmitted signals may not be completely absorbed by the load causing a portion of the transmitted signal to be reflected back into the transmission line. If the source, transmission line, and load impedance are equal, reflected signals do not cause a problem.

To measure the series resistance of the DeviceNet data pair conductors and the attached terminating resistors, do the following:

1. Turn all network power supplies off.
2. Measure and record the DC resistance between CANH and CANL at each end of the network and also in the middle.

3. Measured resistance should be between 50 and 70 ohms. If a value is outside of this range, these are the suggested actions:

Measured resistance	Suggested action
< 50 ohms	• Check for short circuit between CANH and CANL wiring. • Check for more than 2 terminating resistors. • Check nodes for faulty transceivers. Use an ohm meter to measure current leakage through all of the CAN circuits: (1) Measure resistance between CANH and V^-, CANH and V^+, CANL and V^- and CANL and V^+. (2) Values obtained should be in the approx. range between 1M and 4M ohms. (3) Check for electrical damage in the circuit if values are well outside recommended range.
50 — 70 ohms	• Normal range. No action required.
71 — 125 ohms	• Check for open circuits in CANH or CANL wiring. • Check for a missing terminating resistor.
> 125 ohms	• Add termination resistors. • Check for open circuits in CANH and CANL wiring.

Cabling reference

In a DeviceNet network the *Trunk Line* is the main network cable. The *Trunk Line* is usually a “thick” cable, between the two 121 ohm resistors. that anchor the two ends of the network (one at each end), serving as “end of line terminators”. A *Drop Line* is the network branching cable between the trunk line tap or junction box and the individual network node. A *Drop Line* must not exceed 6 meters (19 feet, 8inches) in length, no matter what data transfer baud rate is being used.

NOTE: DeviceNet supports a maximum of 64 nodes on a single network. Standard 11-bit identifier CAN architecture is used for addressing. (There is no use of extended 29-bit syntax.)

NOTE: There are trade-offs between the longest *Trunk* and *Drop* cable lengths that you can use, and the selected data transfer speed. The baud rate selected can affect, and/or restrict, each of the following variables.

- The available number of network nodes,
- the total overall length of the *Trunk Line,*
- the type of cable used for the *Trunk Line*, and
- the sum total length of all individual *Drop Lines*

Refer to the tables, below, to make sure that you do not exceed the recommended guidelines for cabling, data rate, and cumulative drops:

DATA Transfer Rate	Longest overall length of trunk cable supported	
	Using 100% “thick” cable	*Using 100% “thin” cable*
125k baud per second	500 meters/1640 feet	100 meters/410 feet
250k baud per second	250 meters/820 feet	100 meters/410 feet
500k baud per second	100 meters/410 feet	100 meters/410 feet

DATA Transfer Rate	Longest drop cable length supported	
	Each drop, maximum length	*Cumulative length, all drops*
125k baud per second	6 meters/19.67 feet	156 meters/512 feet
250k baud per second	6 meters/19.67 feet	78 meters/256 feet
500k baud per second	6 meters/19.67 feet	39 meters/128 feet

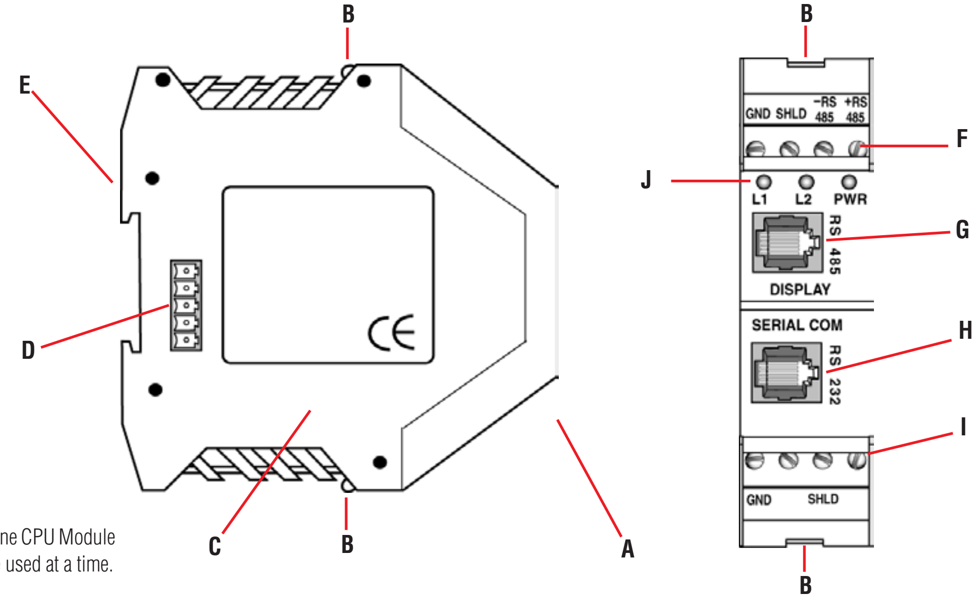
Technical specifications

The Alpha® Gateway Interface is assembled with three modules that are described in the tables that follow. Note that the CPU Module and Power Supply Module are physically interchangeable. Either one can be mounted next to the Network Interface Module.

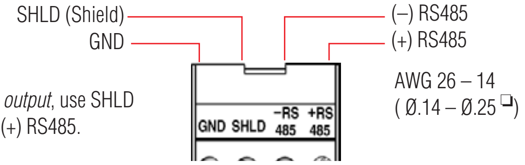
- CPU Module — interface between the AGI and Alpha® signs.
- Power Supply Module — supplies power to the CPU Module and Network Interface Module.
- Network Interface Module — interface between the PLC and Alpha® sign network.

CPU Module

Description



Only one CPU Module can be used at a time.

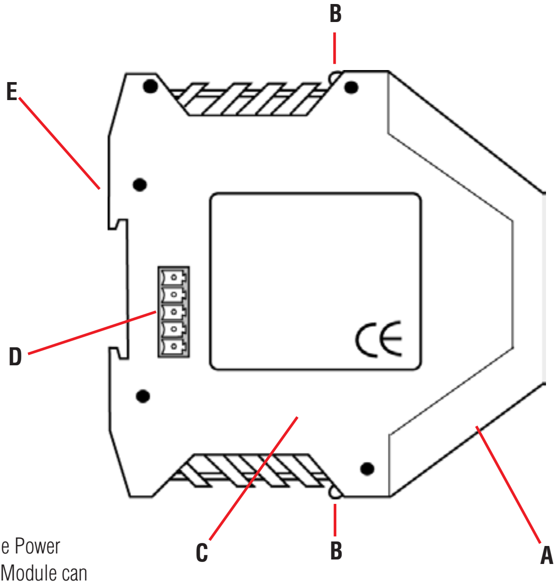
Item	Name	Description
A	Module top	The internal PCB is attached to the Module top.
B	Release buttons	Use a pen tip or other pointed tool to push Release buttons and pull off the Module bottom.
C	Module bottom	After Release buttons (Item B) are unlatched, the Module bottom can be pulled away.
D	Intermodule connector	Used to pass power and signals between modules. When attached to a conductive DIN rail, the rail serves as the CHASSIS GND for all modules connected to the DIN rail.
E	DIN rail latch	Used to attach the module to a DIN EN 5Ø Ø22 mounting rail.
F	Serial output	Shielded RS485 output to one or more Alpha® signs. NOTE: Because of the signal protection afforded by shielding, this is the recommended way of connecting Alpha® signs to the CPU Module.  SHLD (Shield) GND (–) RS485 (+) RS485 AWG 26 – 14 (Ø.14 – Ø.25 ❑) GND SHLD −RS 485 +RS 485 For *shielded RS485 output*, use SHLD with (–) RS485 and (+) RS485.

G	DISPLAY		Unshielded RS485 output to a single Alpha® sign or display. Quick-connect jack is used to: • program messages into a sign • trigger messages already in a sign NOTE: See connection diagram on page 5 (This is not a telephone connector.)
H	SERIAL COM		*Unshielded RS232 input* from a PC. Used to program messages and send them to a CPU Module that is *no farther than 50 feet* from the PC. Use 6-conductor RS232 cable such as Adaptive® pn 1088-8625 (25 feet) or 1088-8627 (50 feet) This is not a telephone connector.
I	Serial input		not used
J	Pass through mode Receiving data	L1 (Red)	Flashes once a second while downloading data to the Alpha® sign network. Flashes briefly when receiving data from an Alpha® sign
	Fault indication Clearing variable data Heartbeat Transmitting data	L2 (Yellow)	Flashes when the AGI encounters a fault. Flashes continuously when power is cycled, clearing variable data/Alpha® sign registers. Flashes once every 500 ms to indicate that the Heartbeat is enabled. Flashes when transmitting data to an Alpha® display.
	Power Indicator	PWR	Always green while unit has power

Specifications

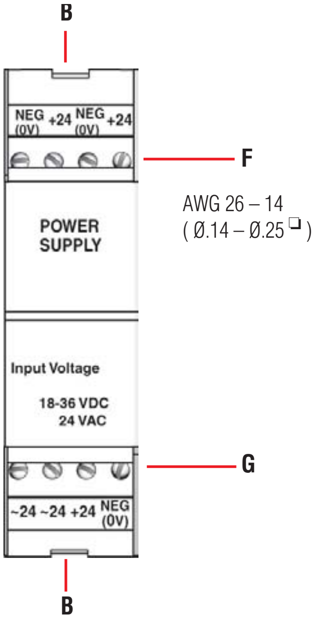
Dimensions:	2.75"W x 4.25"H x 1"D
Weight:	4 oz. per module
Operating temperature:	60°C
Humidity range:	10 – 95% non-condensing
Mounting:	DIN rail 35 x 7 mm
Operating voltage:	5 V
Current draw:	150 mA
Power consumption:	0.75 W
Serial (in):	Communication type: RS232 Terminal type: RJ11 Protocol: EZ95 (modified)
Display (out):	Communication type: RS485 Terminal type: RJ11 Protocol: Modbus ASCII
Terminals (out):	Communication type: RS485 Terminal type: Screw Wire size: AWG 26 - 14 (US) / 0, 14-2, 5^2(Europe) Protocol: Modbus ASCII Max. number of drops: 32 Max. distance: 4000 ft (1200 m)

Description

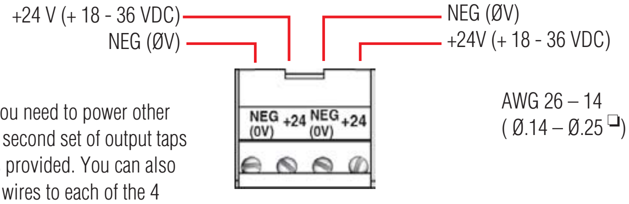
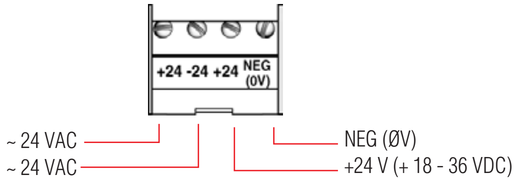


Only one Power Supply Module can be used at a time.

Side view



Front view

Item	Name	Description
A	Module top	The internal PCB is attached to the Module top.
B	Release buttons	Use a pen tip or other pointed tool to push Release buttons and pull off the Module bottom.
C	Module bottom	After Release buttons (Item B) are unlatched, the Module bottom can be pulled away.
D	Intermodule connector	Used to pass power and signals between modules.
E	DIN rail latch	Used to attach the module to a DIN EN 5Ø Ø22 mounting rail.
F	Power output	 +24 V (+ 18 - 36 VDC) NEG (ØV) NEG (ØV) +24V (+ 18 - 36 VDC) NEG (0V) +24 NEG (0V) +24 AWG 26 – 14 (Ø.14 – Ø.25 ❑) NOTE: If you need to power other devices, a second set of output taps (4 total) is provided. You can also connect 2 wires to each of the 4 terminals (8 wires out, total), to power up 4 separate 24V devices.
G	Power input	 +24 -24 +24 NEG (0V) ~ 24 VAC ~ 24 VAC NEG (ØV) +24 V (+ 18 - 36 VDC)

Specifications

Dimensions:	2.75"W x 4.25"H x 1"D
Weight:	4 oz. per module
Operating temperature:	60°C
Humidity range:	10 – 95% non-condensing
Mounting:	DIN rail 35 x 7 mm
AC input voltage[1]	
Max. AC voltage:	25 Vrms
Min. AC voltage:	14 Vrms
Power consumption:	15W @ 24 Vrms
DC input voltage	
Max. DC voltage:	36 VDC
Min. DC voltage:	18 VDC
Output voltage 24 VDC	
Max. voltage:	36 VDC
Min. voltage:	18 VDC
Max. current:	700 mA
Bus output voltage 5 VDC	
Max. voltage:	5.05 V
Min. voltage:	4.95 V
Max. current:	500 mA
Protection	
Type:	Poly switch
Self-resetting:	Yes
Terminals	
Type:	Screw
Wire size:	US spec:AWG 26 - 14/Euro spec: 0, 14-2, 5[2]

[1] Only one power supply, 18 – 36 VDC or 24 VAC, can be used to power this product.

Network Interface Module

Description

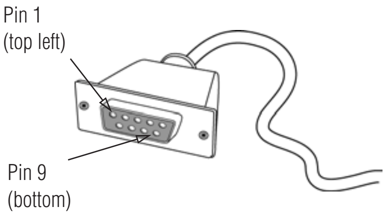
See “Alpha® Gateway Interface” on page 3.

Specifications

Dimensions:	Width = 90mm x Height = 75mm x Depth = 105mm
Weight:	190 grams
Voltage	Min = 4.75 Volts; Typical = 5.00 Volts; Max = 5.25 Volts
Current	Min = 300 mA; Typical = 350 mA; Max = 450 mA
Ambient Environmental Conditions	Temperature Range: 5° C– 60° C Humidity: 10%–95% (no condensation)
Protection	In accordance with DeviceNet standards
DeviceNET:	• DeviceNet certification • Bus powered by embedded +5V supply
EMC Compliance	CE compliant
Mounting:	DIN rail 35 x 7 mm

Other devices on the network

Most DeviceNet connections are custom-configured. However, there is also a protocol that allows the use of a standard DB9 serial interface in a DeviceNet network. The pin-out is shown below for reference in case a device in your network is configured with the DB9 serial connection or you wish to add a device that is so configured:



Pin 1	No connection
Pin 2	Can_L
Pin 3	V –
Pin 4	No connection
Pin 5	Shield
Pin 6	Optional Ground (V-)
Pin 7	Can_H
Pin 8	No connection
Pin 9	V +

FOR REFERENCE ONLY:
Other devices in your network may have DB9 DeviceNET connections.

Configuration data for Alpha® Gateway networks

Overview

The Alpha® Gateway Interface maps Output and Input Data tables in the PLC to the Gateway interface's Input and Output Data tables. Once mapped, the PLC, operating through the Alpha® Gateway Interface (the "AGI"), controls the function and operations of the Alpha® sign network.

The Gateway network is configured to allow two different types of messaging, "Add/remove messages" and "Priority messaging".

- "Add/remove messages" allows several different messages to all be active at once. The messaging control function allows different priority levels to be applied to the messages that are programmed with *Alpha® Automation Software* (or *Gateway Messaging Software*.)
- "Priority messaging" activates only one message at a time, leaving full control of messaging display with the PLC.

Variable data corresponding to values in the Output Data Table and in the *Alpha® Automation Software* (or *Gateway Messaging Software*) may be included in any message.

NOTE: All Alpha® signs used in a Gateway network must have Alpha Industrial Protocol firmware.

Information flow

In an Alpha® sign network, a device is any point in the information pathway that is capable of sending and receiving a data signal.

In the most basic network configuration (one input, one PLC, one AGI, and one sign), the PLC, AGI, and sign are all capable of both sending and receiving data.

See the illustration in "Introduction" on page 1 for a description of information flow.

Functional capacity

Data mapping between the PLC and AGI allows the industrial network to control the following operations of an Alpha® sign network:

- Priority messaging
- Add/remove messages in queue
- Read currently-running messages in a sign
- Read variable data
- Update variable data
- Monitor the status of signs on the network, individually or all at once
- Determine if the AGI is processing data
- Clear any status bits set by the AGI
- Clear all messages currently being displayed

Alpha® sign network data storage

An Alpha® Gateway Interface is connected to the Alpha® signs via a multi-drop RS485 network. This network will support up to 32 drops before requiring a repeater. These signs can be addressed from 001 to 254 by using *Alpha® Automation Software* (see the *Alpha® Automation Software* online Help system) or through each sign's internal DIP switches. See "Memory map" on page 7 to see how information is stored in an Alpha® sign.

Network device GSD and EDS files

The AGI operating on an industrial network may need to be identified to the network using an initialization file. For example, Profibus DP networks use *.gsd files. DeviceNet networks use *.eds files. Check your network documentation.

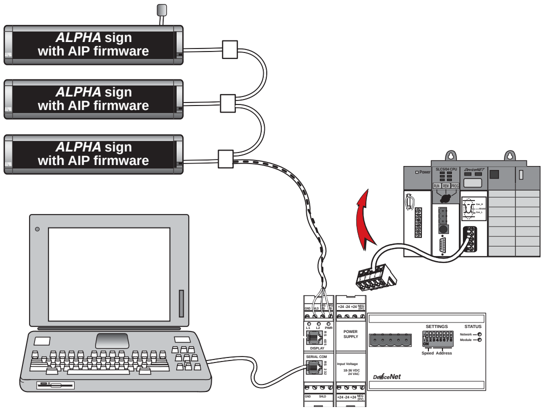
Look for GSD and EDS files in the same directory path used to install *Alpha® Automation Software* (or *Gateway Messaging Software*.) When loading the GSD or EDS files to your system, set the input and output parameters as follows:

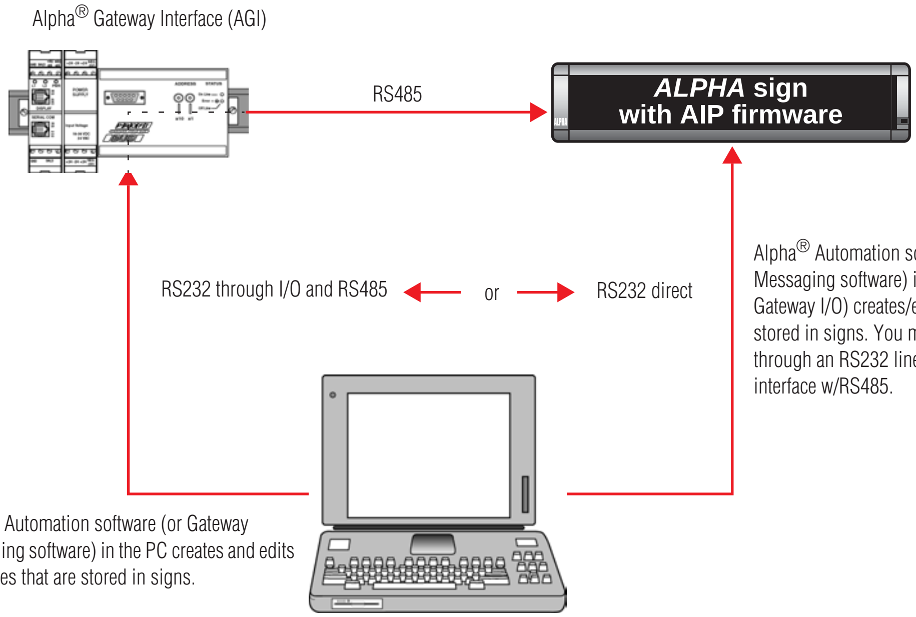
- 32 Words (64 Bytes) out for the PLC Output Data table from the PLC to the Gateway
- 16 Words (32 Bytes) in for the PLC Input Data table from the Gateway to the PLC

Downloading messages to an Alpha® sign network

To download messages from the *Alpha® Automation Software* (or *Gateway Messaging Software*) to the Alpha® sign network, *physically disconnect the Network Interface Module from the industrial network*. Reconnect the network input to the Network Interface Module only after you have completed all downloads.

Also, you could bypass the Network Interface Module and physically connect your PC to the Alpha® sign network directly via an RS232 connection:





Alpha® Automation software (or Gateway Messaging software) in the PC (through Gateway I/O) creates/edits messages that are stored in signs. You may download directly, through an RS232 line **or** through Gateway interface w/RS485.

Alpha® Automation software (or Gateway Messaging software) in the PC creates and edits messages that are stored in signs.

Output Data table

The Output Data table mapped in the PLC to the AGI Input Data table is used to control and execute all available operations of the message sign network.

The Output Data table consists of 32 words, reserved for the functions described in the table below. Valid values are listed for each word. (Invalid values are shown in Table 9 on page 32.)

Table 1: Output Data table

Word	Description	Valid values (decimal)
0	Control Word	2 = Write 3 = Read Message Queue 4 = Read Variable Data -1 = Clear Status bits
1	Sign Address (SA)	1 – 254, 255 is broadcast address.
2	Add Message Number to queue (AM) (When the Control Word is a 2)	1 – 4000 and 4095 are valid message numbers; 4001–4094 are *not* valid. (4095 identifies the background message.)
	Clearing a single bit in Words 1 and 2 of the Input Data table (CB) (When the Control Word is a -1)	0 = Used for clearing a Status bit for an individual sign address
	Starting Register Index (RI) (When the Control Word is a 3)	1-6 when reading Message queue data
	Starting Register Index (RI) (When the Control Word is a 4)	1-9 when reading Variable data.
3	Remove Message Number from queue (When the Control Word is a 2)	0 = Used for adding multiple messages to the queue 1 – 4000 and 4095 valid Message Numbers: 4001–4094 *not* valid. 4095 always used to identify the background message -1 = Used for priority messaging and clearing the message queue
4	Variable data #0	0 – 65535
5	Variable data #1	0 – 65535
6	Variable data #2	0 – 65535
7	Variable data #3	0 – 65535
8	Variable data #4	0 – 65535
9	Variable data #5	0 – 65535
10	Variable data #6	0 – 65535
11	Variable data #7	0 – 65535
12	Variable data #8	0 – 65535
13	Variable data #9	0 – 65535
14	Variable data #10	0 – 65535
15	Variable data #11	0 – 65535
16	Variable data #12	0 – 65535
17	Variable data #13	0 – 65535
18	Variable data #14	0 – 65535
19	Variable data #15	0 – 65535
20	User defined Variable ID #1	16 – 99
21	Variable data	0 – 65535
22	User defined Variable ID #2	16 – 99
23	Variable data	0 – 65535
24	User defined Variable ID #3	16 – 99
25	Variable data	0 – 65535
26	User defined Variable ID #4	16 – 99
27	Variable data	0 – 65535
28	User defined Variable ID #5	16 – 99
29	Variable data	0 – 65535
30	User defined Variable ID #6	16 – 99
31	Variable data	0 – 65535

Displaying messages on Alpha® signs

There are three different ways in which messages can be displayed or managed on an Alpha® sign:

- **Add/remove messages —** Allows more than one message to be displayed at a time, with priority levels set for each message. This allows the ability to manage display sequences and timing.
- **Priority messaging —** Used to display one message at a time.
- **Clearing the Message Queue —** Removes messages from the message queue, to display the background message.

Table 2: Messaging methods

Method	Description	Examples	Benefit	Valid values
Add/remove messages	Allows multiple messages to run concurrently on the same sign(s). When this method is used it also allows the use of priority levels to control the timing and display sequence for all the messages in the queue.	• Table 3 on page 27 • Table 4 on page 27	Allows some scheduling of messaging display during initial creation of the message.	• Word 0 = 2 (only valid) • Word 1 = 1-255d* • Word 2** = 1-4000, 4095d • Word 3** = 1-4000, 4095d
Priority messaging	Allows one message number to be used to show a single message while automatically removing the single message that was running previously. This frees you from having to remove the message that was running before you display a new message	• Table 5 on page 28.	This method allows the PLC to have full control of message display.	• Word 0 = 2 • Word 1 = 1-255d* • Word 2** = 1-4000, 4095d • Word 3** = 0xFFFF(-1d)
Clearing the Message Queue	Allows you to clear all currently running messages on a sign, by storing 0 in word 2 and –1d (0xFFFF) in word 3, as shown in Table 2 on page 26. When this is done, the background message that was programmed using the Alpha® Automation software (or Gateway Messaging software) is displayed. (This is the default background message "NO BACKGROUND MESSAGE" if no other has been programmed.)	• Table 12 on page 33.	Clears all message queue data.	• Word 0 = 2 • Word 1 = 1-255d* • Word 2** = 0 (only valid) • Word 3** = 0xFFFF(-1d)
* When the sign address is 255 decimal/0xFFFF hexadecimal, this is a broadcast message. There is NO reply from sign(s) on the network. ** The value of Words 2 and 3 can not be the same. The Gateway will not process the information. See Output Data Table Integrity Bit Table.				

Add/remove messages example 1

This is an example of the steps required to add three different messages, one at a time, on the same Alpha® sign with *Add/remove messages* method.

In this example, we want to display messages 40, 42, and 500 individually on the same sign. The sign address is 6 (decimal).

Using information found in Table 1 on page 25, store the desired message numbers and sign information into words 0-3 of the Output Data table.

Table 3: Add/Remove messages: Example 1

Word	Step 1	Step 2	Step 3	Step 4
0	2	2	2	2
1	6	6	6	6
2	0	40	42	500
3	-1	0	40	42
Result	Message queue is cleared and the background message is displayed.	Message 0 is removed and Message 40 is added to the queue.	Message 40 is removed and Message 42 is added to the queue.	Message 42 is removed and Message 500 is added to the queue.
NOTE: All values for words 0 through 3 are shown in decimal.				

Add/remove messages example 2

This example shows how to add three different messages to the queue at the same time on an Alpha® sign. This requires that each message be added without removing the previous message.

In this example, we want to display messages 84, 589 and 34 at the same time on a sign. The sign address is 55 (decimal).

Using information found in Table 1 on page 25, store the desired message numbers and sign information into words 0-3 of the Output Data table.

Table 4: Add/Remove messages: Example 2

Word	Step 1	Step 2	Step 3	Step 4
0	2	2	2	2
1	55	55	55	55
2	0	84	589	34
3	-1	0	0	0
Result	Message queue is cleared and the background message is displayed.	Message 0 is removed and message 84 is displayed.	Message 0 is removed and messages 84 and 589 are displayed.	Message 42 is removed and messages 84, 589, and 34 are displayed.
NOTE: All values for words 0 through 3 are shown in decimal.				

NOTE: Messages can be removed individually (see steps 3 and 4 in Table 3) or all at the same time (step 1 in Table 5).

Priority messaging example

This example shows how to display individual messages using *Priority messaging*, and the steps required to add three different messages on an Alpha® sign with *Priority messaging*. This means that only the number of the message to be displayed is stored in word 2.

In this example, we want to display messages 954, 26, and 50 individually on the same sign. The sign address is 3 (decimal). Using information found in Table 1, store the desired message numbers and sign information into words 0-3 of the Output Data table.

Table 5: Priority Message example

Word	Step 1	Step 2	Step 3	Step 4
0	2	2	2	2
1	3	3	3	3
2	0	954	26	50
3	-1	-1	-1	-1
Result	Message Queue is Cleared and the background message is displayed.	Message 954 is displayed.	Message 26 is displayed.	Message 50 is displayed.
NOTE: All values shown in decimal.				

Displaying variable data on Alpha® signs

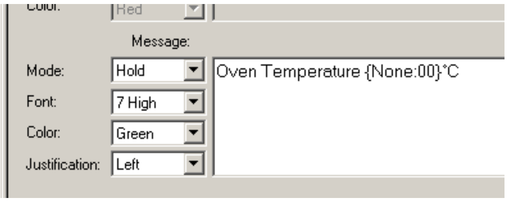
An Alpha® sign has the capability to store up to 100 variables in its memory. These variables are stored in a sign's memory register (001 to 100) corresponding to the variable number (0 to 99) used in the Alpha® Automation software (or Gateway Messaging software.)

There are 16 words in the Output Data table that are reserved for variable numbers 0-15. Words 20-31 are used to allow access to variable numbers 16-99. Variable data stored in words 4-31 are sent to all signs by using defined variable ID numbers and variable data in the next word following (see Output Data Table, odd-numbered variable data in words 20-31 must correspond to even-numbered variable ID numbers in the preceding word), once the Gateway receives the change of state for a particular variable.

The two examples that follow show display of messages containing embedded variable data.

Displaying variable data example 1

In this first example, message 5 contains variable 0, used for a processor's oven temperature. Message 5 is created in Alpha® Automation software (or Gateway Messaging software) in this format:



Color: Red

Message:

Mode: Hold

Font: 7 High

Color: Green

Justification: Left

Oven Temperature {None:00}°C

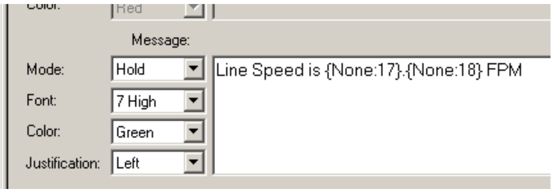
We want to display this on the sign with address 20 using the *Add/remove messages* method. This only requires that the message be triggered once and the variable data that is required stored in word 4 of the Output Data table.

Table 6: Displaying Variable data: Example 1

Word	Step 1	Step 2	Step 3	Step 4
0	2	2	2	2
1	20	20	20	20
2	0	5	5	5
3	-1	0	0	0
4	0	30	31	32
Result	Message queue is cleared and the background message is displayed.	Message 0 is removed and message 5 is displayed as "Oven Temperature 30˚C".	Message 0 is removed and message 5 is displayed as "Oven Temperature 31˚C".	Message 0 is removed and message 5 is displayed as "Oven Temperature 32˚C".
NOTE: All values shown in decimal.				

Displaying variable data example 2

In this second example, message 30 contains variables 17 and 18 that are used for a manufacturer's production line speed in XX.X FPM. Message 30 is created using the Alpha® Automation software (or Gateway Messaging software) in a format that will look like this:



Color: Red

Message:

Mode: Hold

Font: 7 High

Color: Green

Justification: Left

Line Speed is {None:17}.{None:18} FPM

We want to display this message on the sign with address 4 using the *Priority messaging* method. This only requires that the message be triggered once, with variable data stored in the appropriate words of the Output Data table (in this case words 20–23).

Table 7: Displaying Variable data: Example 2

Word	Step 1	Step 2	Step 3	Step 4
0	2	2	2	2
1	4	4	4	4
2	0	30	30	30
3	-1	-1	-1	-1
20	17	17	17	17
21	6	14	14	14
22	18	18	18	18
23	0	3	2	4
Result	Message queue is cleared and the background message is displayed.	Message 30 is displayed as "Line Speed is 14.3 FPM".	Message 30 is displayed as "Line Speed is 14.2 FPM".	Message 30 is displayed as "Line Speed is 14.4 FPM".
NOTE: All values shown in decimal.				

Input Data table from the AGI to the PLC

The Input Data table consists of 16 words that are used in conjunction with the Output Data table for the function that is being used.

Various words are used to determine the status of a sign, the status of the Gateway and requested data.

The Input Data table also gives feedback to the PLC. If there is a problem trying to communicate with a specific sign address (1-31), it will alert the PLC if there is a major fault and store data values requested by the PLC.

The functions/features in the Input Data table are:

- Operation Counter
- Operation Status of the Gateway
- Output Data Table Integrity (O.D.T.I.)
- Sign Status
- Requested Data

Table 8: Input Data table

Word	Meaning	Valid data (decimal)
0	Operation Counter	0 – 65535
1	Error Status (sign address #1-15) / Output Data Table Integrity	Output Data Table Integrity bit (Bit 0) 0 = Gateway completed processing of data or no error 1 = Improper data in Output Data Table Error Status (Bits 1-15) 0 = Gateway received a response from a sign 1 = Gateway did not receive a response from a sign
2	Error Status (sign address #16-31)	Error Status (Bits 0-15) 0 = Gateway received a response from a sign 1 = Gateway did not receive a response from a sign
3	Reserved	Reserved for future use
4	Requested Data #1	0 – 65535
5	Requested Data #2	
6	Requested Data #3	
7	Requested Data #4	
8	Requested Data #5	
9	Requested Data #6	
10	Requested Data #7	
11	Requested Data #8	
12	Requested Data #9	
13	Requested Data #10	
14	Requested Data #11	
15	Requested Data #12	

Operation Counter/Output Data table Integrity bit (ODTI)

The Operation Counter (word 0) is used to let the PLC know that it has completed a task. When the Gateway receives data from the PLC, it begins to process the operation immediately. Once it has successfully completed the operation, the Operation Counter is incremented by 1. The counter automatically resets itself back to zero when it reaches 65,535 (0xFFFF).

The Output Data Table Integrity bit (or ODTI, word 1, bit 0) is similar to the Operation Counter, but it lets the PLC know that the Gateway is currently processing data. If the bit is set high (1), it indicates that the Gateway is processing data. If the bit is set low (0), it indicates that the Gateway has completed processing of data. If the Gateway encounters a problem, such as invalid data sent by the PLC, it will also set this bit high (1). This will remain high until the Gateway receives valid information. The Operation Counter is then incremented.

Table 9: ODTI bit invalid data

If Output Data table Word 0 is:	Then Word(s):	(whose function is:)	will have Word 1, Bit 0 set to 1 (invalid data) if:
2	1	Sign Address (SA)	SA > 255 OR S A = 0
	2	Add Message (AM)	AM < 1 or (≥ 4001 and ≤ 4094) or ≥ 4096
	3	Remove Message (RM)	RM (> -1 and < 1) or (≥ 4001 and ≤ 4094) or ≥ 4096
	2, 3	Add Message / Remove Message	AM = RM
	20, 22, 24, 26, 28, 30	Valid Variable ID (VID)	VID = 0 or < 0 or >100
3	1	Read message queue for Sign Address (SA)	SA > 254 or SA = 0
	2	Read Register Index (RI) for message queue	RI > 6 or RI = 0
4	1	Read variable data for Sign Address (SA)	SA > 254 or SA = 0
	2	Read Register Index (RI) for variable data	RI > 9 or RI = 0

Sign Status

The Sign Status bit consists of two words that indicate whether the sign gave an acknowledge back to the Gateway.

When the Gateway receives an acknowledgement, it will clear (0) the appropriate bit. The Gateway will set the bit if the sign does not give an acknowledgement back after two attempts, 500 milliseconds apart.

- Word 1 — Bit 0 of the word is for the Output Data Table Integrity bit (ODTI). Bits 1-15 of the word contain the Error Status for sign addresses 1-15.
- Word 2 — Bits 0-15 of the word contain the Error Status for sign addresses 16-31.

Table 10: Word 1 of Input Data table

Word 1	Most Significant Byte								Least Significant Byte (ODTI = Output Data Table Integrity bit)							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sign Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	ODTI
Data	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11: Word 2 of Input Data table

Word 2	Most Significant Byte								Least Significant Byte							
	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Sign Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Data	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Clearing the Sign Status bits

There are two ways to clear the bits in words 0-2: all at once or individually:

- To clear all the bits in words 0-2, configure the Output Data table using the information found in step 1 of Table 12.
- To clear a bit in words 1 and 2 representing a specific sign address, configure the information in the Output Data table using the information found in step 2 of Table 12:

Table 12: Methods of clearing the Sign Status bits

Word	Step 1	Step 2
0	-1 (0xFFFF)	-1 (0xFFFF)
1	3	3
2	1–4000, 4095	0
Result	Clears words 0-2 of the Input Data table. Sets all bits to 0.	Clears only bit 3 in word 1 representing sign number 3.

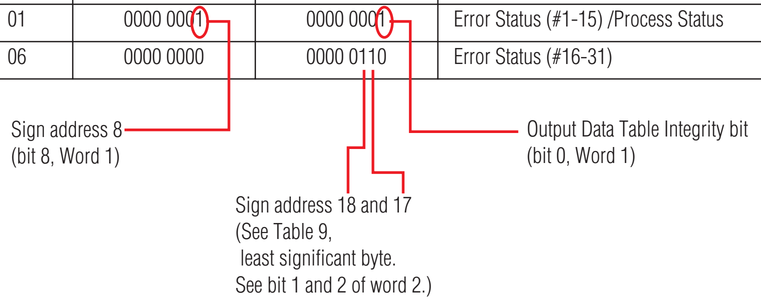
Input Data table words 0 - 2 example

The following table shows information relating to Words 0, 1 and 2 in the Input Data table.

In this example, the Gateway has completed 100 operations (word 0). Bit 0 word 1 is set high (1) and is indicating that the Gateway is currently processing data or that the PLC sent invalid data. Bit 8 of word 1 and bits 1 and 2 of word 2 are set high (1) representing that the Gateway did not receive any response from sign numbers 8, 17, and 18:

Table 13: Input Data table example

Word	Decimal	MSB	LSB	MSB	LSB	Meaning
0	00100 operations	00	64	0000 0000	0110 0100	Operation Process Counter
1	00257	01	01	0000 0001	0000 0001	Error Status (#1-15) /Process Status
2	00006	00	06	0000 0000	0000 0110	Error Status (#16-31)



Requesting data from an Alpha® sign

When data is requested by the PLC, the PLC stores the requested information in words 4-15 of the Input Data table.

This data can either be currently running messages (in the Message Queue) or variable data from a sign. These words are reserved in the Input Data table that the Alpha® Gateway Interface uses only for returning message information or variable data to the PLC.

Requesting currently running messages in a sign

The Alpha® Gateway Interface allows you to read currently running messages in a sign's queue by using the Read Message Queue function.

The message queue allows for up to 64 messages running at one time on any given sign. These message numbers are stored in the sign's memory from registers 103 to 166. Register 103 represents the most recent message number to be added. Register 166 represents the message number that has not been activated for the longest interval.

Since there are only 12 words available in the Input Data table for the requested data, the data is returned to the PLC in 6 groups of 12 words.

The group of 12 words that are returned to the PLC is determined by the Starting index (word 3 of the Output Data table). The Starting Index value and the associated registers that will be returned to the PLC are shown in the following section of the Output Data table.

Table 14: Requesting Message Queue data (Output Data table)

Word	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7
0	Control Word	3	3	3	3	3	3
1	Sign #	6	6	6	6	6	6
2	Starting Index	1	2	3	4	5	6
Meaning	Read registers from sign #6	Read registers 103-114	Read registers 115-126	Read registers 127-138	Read registers 139-150	Read registers 151-162	Read registers 163-166 NOTE: Words 8–15 set to -1d (0xFFFF)

The requested data is stored in the Input Data table in words 4 through 15.

Requesting variable data from a sign

The Gateway allows you to read the variable data that is currently stored in a sign.

All signs on the network will have the same variable data. There are 100 variable registers in a sign. This variable data is stored in the sign memory from registers 001 to 100. Since there are only 12 words available in the Input Data table for the requested data, the data is returned to the PLC in one of 9 groups of 12 words.

The group of 12 words that are returned to the PLC is determined by the Starting index (word 3 of the Output Data table).

The Starting index value and the associated registers that will be returned to the PLC are shown in the following section of the Output Data table:

Table 15: Requesting variable data (Output Data table)

Word	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9
0	Control Word	4	4	4	4	4	4	4	4
1	Sign #	6	6	6	6	6	6	6	6
2	Starting Index	1	2	3	4	5	6	7	8
Meaning	Read registers from sign #6	Read registers 001-013	Read registers 014-026	Read registers 027-039	Read registers 040-052	Read registers 053-065	Read registers 066-078	Read registers 079-091	Read registers 092-100 NOTE: Words 13–15 set to -1d (0xFFFF)

The data requested will be sent back to the PLC using words 4–15 in the Input Data table.