Aromat Matsushita/NAIS Ethernet Device Driver Help

Table of Contents

Aromat Matsushita/NAIS Ethernet Device Driver Help

Help version 1.005

CONTENTS

Overview

The Aromat Matsushita/NAIS Ethernet Device Driver was designed specifically for use with 32 bit OPC Server products running on Intel microprocessor based computers. It is intended for use with Aromat Matsushita/NAIS FP devices with ET-LAN units. For Operating System (OS) requirements, refer to the OPC Server's Help documentation.

Device Setup

Supported Devices

Any Aromat Matsushita/NAIS FP series PLC with optional ET-LAN Ethernet unit.

Connection Timeout

This parameter specifies the time that the driver will wait for a connection to be made with a device. Depending on network load, the connect time may vary with each connection attempt. The default setting is 3 seconds. The valid range is 1 to 60 seconds.

Request Timeout

This parameter specifies the time that the driver will wait on a response from the device before giving up and going on to the next request. Longer timeouts only affect performance if a device is not responding. The default setting is 1000 milliseconds. The valid range is 100 to 9999 milliseconds.

Retry Attempts

This parameter specifies the number of times that the driver will retry a message before giving up and going on to the next message. The default setting is 3 retries. The valid range is 1 to 10.

Device IDs

This is the IP address of the destination device.

Note: This driver does not support multi-level networks where messages are relayed from one PLC to another. It is possible to communicate with PLCs on other network levels if messages are relayed through a router.

See Also:

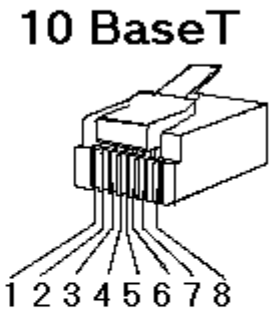
Communications Parameters
Station Numbers
Request Size

ET-LAN Connection Settings

Cable Diagrams

Patch Cable (Straight Through)

TD +	1	OR/WHT	OR/WHT	1 TD +
TD -	2	OR	OR	2 TD -
RD +	3	GRN/WHT	GRN/WHT	3 RD +
	4	BLU	BLU	4
	5	BLU/WHT	BLU/WHT	5
RD -	6	GRN	GRN	6 RD -
	7	BRN/WHT	BRN/WHT	7
	8	BRN	BRN	8
		RJ45		RJ45



Crossover Cable

TD +	1	OR/WHT	GRN/WHT	1 TD +
TD -	2	OR	GRN	2 TD -
RD +	3	GRN/WHT	OR/WHT	3 RD +
	4	BLU	BLU	4
	5	BLU/WHT	BLU/WHT	5
RD -	6	GRN	OR	6 RD -
	7	BRN/WHT	BRN/WHT	7
	8	BRN	BRN	8
		RJ45		RJ45

Communications Parameters

Protocol

The protocol setting specifies whether the driver should connect to the ET-LAN unit using the User Datagram Protocol (UDP) or the Transfer Control Protocol (TCP). The driver requires Winsock V1.1 or higher.

Open Method

The ET-LAN unit must be configured to use one of three methods to process connection open requests when the TCP protocol is used: "Active Open," "Unpassive Open" and "Full Passive Open." Each connection that could possibly be made to the ET-LAN unit must be individually configured in the unit. See ET-LAN Connection Settings for more information. If the unit is configured to use the Active Open method, it will actively attempt to initiate the connection. The Active Open method cannot be used to connect with this driver, however you may still use the Active Open method for PLC-to-PLC connections. If the unit is configured to use the Unpassive Open method, it will wait passively for another node to initiate the connection. This node may use any IP and Port number. If the Full Passive Open method is used, the ET-LAN unit will wait passively for a node, which uses a specified IP and Port, to initiate the connection. Select the method the ET-LAN unit is configured to use for a connection with this driver.

Important: There is a known operating system issue that prevents the driver from reusing a given source Port and IP with TCP/IP for 4 or more minutes after a server shutdown. Because of this, we strongly recommend the use of the TCP/Unpassive or UDP connection methods over the TCP/Full Passive method.

Source Port Number

The Source Port Number specifies what port the driver should use to send and receive messages. This only applies to UDP and TCP/Full Open. The operating system will automatically select an unused source Port number if TCP/Unpassive mode is used. It is recommended that Port number greater than 1024 be used. The default Port number is 1025. Certain restrictions apply to Port numbers. For more information, refer to **ET-LAN Connection Settings**.

Destination Port Number

The Destination Port Number specifies what Port the ET-LAN is configured to send and receive messages. The default Port number is 1025. It is recommended that Port numbers greater than 1024 be used. Certain restrictions apply to Port numbers. For more information, refer to **ET-LAN Connection Settings**.

Station Numbers

Source Station Number

Each node on the network must be assigned a unique station number. This includes the connection points in the driver (one per device object). Specify a station number that will uniquely identify the driver's connection point. Valid numbers range from 1 to 64, where 1 is the default.

Note: For more information, refer to **ET-LAN Connection Settings**.

Destination Station Number

Specify the station number of the target device. Valid numbers range from 1 to 64, where 1 is the default.

See Also: **Device Setup**

Request Size

Request size refers to the number of bytes that may be requested from a device at one time. To refine the performance of this driver, the request size may be configured to one of the following settings: 32, 64, 128, 256 and 512 bytes. The default value is 64 bytes. If a large number of closely spaced addresses will be read, it is most likely advantageous to specify a larger request size. If a few addresses that are widely separated will be read, then it is most likely advantageous to use a smaller request size.

ET-LAN Connection Settings

Overview

Each ET-LAN unit can maintain up to eight connections. Connections can be made to other PLCs, routers or to this driver. Each connection must be described ahead of time in the PLC's shared memory area. Connection descriptions include: protocol, open processing method, source and destination IP, source and destination Port numbers, source and destination station numbers and routing information. The shared memory area is configured using a ladder program.

Connection Types

The ET-LAN unit supports communication using the **User Datagram Protocol (UDP)** and the **Transfer Control Protocol (TCP)**. Requests to make connections using TCP must be processed using one of three methods: **Active Open**, **Unpassive Open** and **Full Passive Open**.

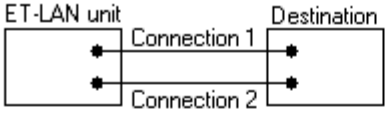
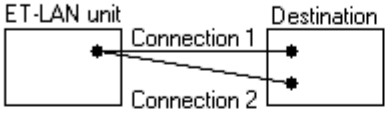
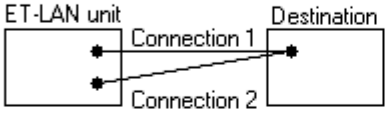
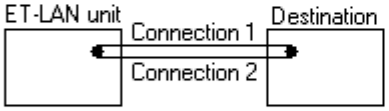
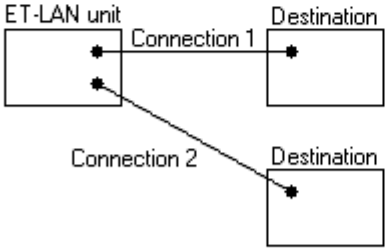
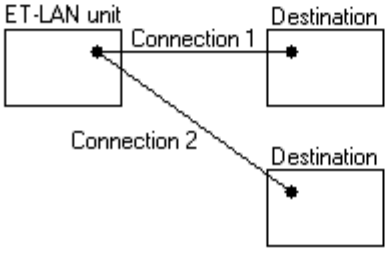
UDP connections are initiated by the driver. The IP and Port used by the driver must be specified in the PLC's shared memory area. When programming the device to use UDP, it may be necessary to cycle the power for changes to take effect. The three methods are described as follows:

- **TCP/Active Open** connections are initiated by the PLC and are not supported by this driver. Active Open connections can still be used for PLC-to-PLC communication.
- **TCP/Unpassive Open** connections are initiated by the driver. The IP and Port used by the driver need not be specified in the PLC's shared memory area. This is the preferred connection method.
- **TCP/Full Passive Open** connections are initiated by the driver. The IP and Port used by the driver must be specified in the PLC's shared memory area. There is a known operating system issue that prevents the driver from reusing a given source Port and IP with TCP/IP for 4 or more minutes after a server shutdown. Because of this, we strongly recommend the use of the TCP/Unpassive or UDP connection methods instead.

Port Assignment

Port numbers are used to distinguish each of the various communication processes. It is recommended that Port

numbers be greater than 1024, since many of the lower Port numbers are commonly reserved by computer operating systems. When multiple connections are made to an ET-LAN unit, there are further restrictions on the Port numbers illustrated below. A similar table may be found in the ET-LAN documentation. The descriptions given below explain how the configuration relates to a server project.

Combination			Availability	
Destination configuration	Port number settings (each circle is a unique port number)	Description	Communication method	
			TCP	UDP
ET-LAN is connected to one destination node		Destination is another PLC or two device objects using different ports on same channel or on different channels using same IP.	1	1
		Destination is another PLC or two device objects using different ports on same channel or on different channels using same IP.	1, 2	N/A
		Destination is another PLC or two device objects using the same port on same channel or on different channels using same IP.	1	N/A
		Destination is another PLC or two device objects using the same port on same channel or on different channels using same IP.	N/A	N/A
ET-LAN is connected to multiple destination nodes		Each destination could be another PLC or a device object. If two device objects, they are on different channels using different IP's.	1, 3	1, 3
		Each destination could be another PLC or a device object. If two device objects, they are on different channels using different IP's.	1, 2, 4	N/A

1. If the destination is two server device objects, they each must be assigned a unique station number.

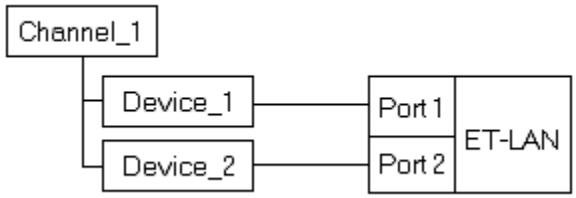
2. Not available with FP2 ET1 unit.

3. This configuration will also work if the channels use the same IP and each device uses a unique Port. If channels use different IP's, then device objects may use the same Port number.

4. A potential message routing problem exists here. If destinations are two server device objects, they must not use the same IP and Port numbers.

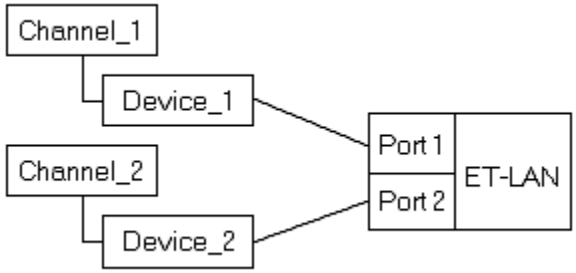
Recommended Configuration for Multiple Connections

There should generally be a one-to-one relationship between server device objects and PLCs. It is possible to configure multiple server device objects that connect with the same PLC, possibly resulting in a small increase in performance. If you wish to create multiple connections to a single PLC, we recommend that each device object use a unique combination of IP (selected from channel network adaptor) and source Port number illustrated below. For more information on how to create multiple source IP's, refer to **Multihoming**.



Multiple connections, single channel

Device_1 and Device_2 should use unique Port and station numbers.



Multiple connections, multiple channels

If Channel_1 and Channel_2 use the same IP, then Device_1 and Device_2 should use unique Port numbers. If the channels use different IP's, then the devices may use the same Port number.

Both devices should use unique station numbers.

Station Numbers

Each device object that connects with the same PLC should have a unique source station number. Server device objects that connect with different PLCs may use the same source station number.

Configuring the Shared Memory Area

Each possible connection to the PLC must be described in the shared memory area. The shared memory area is configured using a ladder program such as the one illustrated below (see your ET-LAN documentation for more information).

This program configures a single ET-LAN unit, in slot 0, to connect to two external devices. The external devices may be two other PLC's, two server device objects, or a combination. Each connection uses the UDP protocol. The PLC uses a unique Port for each connection, 6004 and 6005. The first external device uses IP 192.168.111.27 (0xC0A86F1B), Port 1025, and station number 10. The second external device uses IP 192.168.111.70 (0xC0A86F46), Port 1026, and station number 11. A subnet mask of 255.255.255.255 (0xFFFFFE00) and a default router/gateway IP of 192.168.111.1 (0xC0A86F01) are specified, but are not required in this particular application.

Allocation of handshake area for internal relays:
| |
| |
| R9010 |
|----| |----[F150 READ, H 0, H 360, K 2, WR 0] |
| [F151 WRT, H 0, WR 4, K 2, H 368] |
| |
| |
Initialization processing:

```
| |
| |
| R9013 |
|----| |----[F1 DMV, H C0A86F83, DT 10] |
| [F0 MV, H 1, DT 12] |
| [F0 MV, K 1, DT 13] |
| [F151 WRT, H 0, DT 10, K 4, H 200] |
| |
| R9014 RD R4C |
|----| |--------|/|-----------------------------------------[ ]----|
| |
| |
```

Initialization of routing information:

```
| |
| |
| RD  |
|----| |----[F150 READ, H 0, H 2D0, K1, DT 300] |
| |
| R9013 |
|----| |----[F1 DMV, H FFFFFE00, DT 14] |
| [F1 DMV, H C0A86F01, DT 16] |
| [F151 WRT, H 0, DT 14, K 4, H 230] |
| |
| |
```

Initialization of connection 1 information:

```
| |
| |
| R9013 |
|----| |----[F0 MV, H 8000, DT 20] |
| [F0 MV, K 6004, DT 21] |
| [F0 MV, H C0A86F1B, DT 22] |
| [F0 MV, K 1025, DT 24] |
| [F0 MV, K 10, DT 25] |
| [F151 WRT, H 0, DT 20, K 6, H 250] |
| |
| RC R11 R50 |
|----| |--------|/|-----------------------------------------[ ]----|
| |
| |
```

Initialization of connection 2 information:

```
| |
| |
| R9013 |
|----| |----[F0 MV, H 8000, DT 30] |
| [F0 MV, K 6005, DT 31] |
| [F0 MV, H C0A86F46, DT 32] |
| [F0 MV, K 1026, DT 34] |
| [F0 MV, K 11, DT 35] |
| [F151 WRT, H 0, DT 30, K 6, H 260] |
| |
| RC R13 R52 |
|----| |--------|/|-----------------------------------------[ ]----|
| |
| |
```

RC is initialization complete signal
RD is initialization error signal
R11 is open error signal (connection 1)
R13 is open error signal (connection 2)
R4C is initialization request signal
R50 is open complete signal (connection 1)
R52 is open complete signal (connection 2)
R9010 is always ON relay
R9013 is initial ON relay
R9014 is initial OFF relay

Connection Information Shared Memory Area

The connection information shared memory area is allotted as follows:

Connection	Address Range
1	0x250-0x25F
2	0x260-0x26F
3	0x270-0x27F
4	0x280-0x28F
5	0x290-0x29F
6	0x2A0-0x2AF
7	0x2B0-0x2BF
8	0x2C0-0x2CF

Offset	Description
0	Protocol and Open Method: 0x8000 UDP 0x0300 TCP/Full Passive Open 0x0200 TCP/Unpassive Open 0x0000 TCP/Active Open (not supported by driver)
1	Source node Port number
2	Partner node IP (low word)
3	Partner node IP (high word)
4	Partner node Port number
5	Partner node station number
6	N/A (partner node physical address)
7	N/A (partner node physical address)
8	N/A (partner node physical address)
9	Reserved
A	Reserved
B	Reserved
C	Reserved
D	N/A (Receive request data size)
E	Reserved
F	N/A (Transmission request data size)

Multi-Homing

Some applications require a unique IP address be associated with each channel. In these cases, the host computer must be **multihomed**. A computer is multihomed when it is configured to have more than one IP address. This may be accomplished by installing multiple **Network Adapter Cards** (NICs) in the computer or by assigning multiple IP addresses to a single NIC.

Adding IP Addresses to a Single NIC on Windows NT

1. Double-click the **My Computer** icon. Select **Control Panel**.
2. Click **Network** and then select the **Protocols** tab.
3. Select **TCP/IP Protocol**.
4. Click **Properties** and then select the **IP Address** tab.
5. Click **Advanced**.
6. Click **Add**.
7. Enter the additional **IP address** and **subnet mask**.

8. Click **OK**.

Adding IP Addresses to a Single NIC on Windows 2000

1. Click the **My Computer** icon. Select **Control Panel**.
2. Click the **Network and Dial-Up Connections** icon.
3. Select either the **Local Area Connection** icon or other icon associated with the **NIC** of interest.
4. Click **Properties** and then select **Internet Protocol (TCP/IP)**
5. Click **Properties**.
6. Click **Advanced** and then select the **IP Settings** tab.
7. Click **Add**.
8. 6. Enter the additional **IP address** and **subnet mask**.
9. Click **OK**.

Notes

1. Multihoming is not supported under Windows 95 or 98.

2. Windows NT will allow up to 5 IP addresses to be added for each NIC via the control panel. If more IP addresses are necessary, they can be added to the registry manually by browsing under **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services**. Next, select the service associated with the **adapter card** in question. Under the service, go to the **Parameters\Tcpip** subkey. Add the IP addresses to **IPAddress**. Edit **SubnetMask** and add an entry for each new IP address.

3. Windows 2000 does not impose a limit on the number of IP addresses you may add via the control panel.

4. The system may need to be restarted before newly added IPs can be used.

5. There will be additional operating system overhead when running on a multihomed system. Unless a very fast device is being used, this overhead should not entirely cancel out the performance gain achieved from distributing the communications load over multiple channels.

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit

BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Float	32 bit Floating point value The driver interprets two consecutive registers as a Floating-point value by making the second register the high word and the first register the low word.

Address Descriptions

Address specifications vary depending on the model in use. The address ranges shown below may exceed the range available for a particular device. If an address is requested that is not supported by the device, the Aromat Matsushita/ NAIS driver will mark the requested data item in error.

Note: The default data types are shown in **bold**.

Address Types

Type	Valid Tag Prefixes	Valid Data Types
I/Os (bits)	X, Y	**Boolean**
I/Os (words)	WX, WY	**Word**, DWord*, Short, Long*, BCD, LBCD*
Relays (bits)	R	**Boolean**
Relay (words)	WR	**Word**, DWord*, Short, Long*, BCD, LBCD*
Timer/Counter Contacts	T/C	**Boolean**
Data Registers	DT	**Word**, DWord*, Short, Long*, Float*, BCD, LBCD*
Timer/Counter Registers	SV, EV	**Word**, Short
Index Registers	IX, IY ID	**Word**, Short, **DWord**, Long

*When using these 32 bit data types, two consecutive 16bit registers will be used; meaning, if address DT1 is declared type DWord, both addresses DT1 and DT2 will be used to store the 32 bitvalue.

Address Specifications

Address Type	Bit Number	Bit Range	Word Number	Word Range
External Inputs (read only)	X<xxx>.<y> xxx-Decimal y-Hex X<xxxxx> xxxxx-Decimal	X000.0-X000.F … X999.0-X999.F X0-X15999	WX<xxx> xxx-Word Number	WX000-WX999
External Ouputs	Y<xxx>.<y> xxx-Decimal y-Hex Y<xxxxx> xxxxx-Decimal	Y000.0-Y000.F … Y999.0-Y999.F Y0-Y15999	WY<xxx> xxx-Word Number	WY000-WY999
Internal Relays	R<xxx>.<y> xxx-Decimal y-Hex R<xxxxx> xxxxx-Decimal	R000.0-R000.F … R999.0-R999.F R0-R15999	WR<xxx> xxx-Word Number	WR000-WR999
Timer/Counter Contacts (read only)	T<xxx> xxx-Decimal	T000-T999	N/A	N/A

	C<xxx> xxx-Decimal	C000-C999		
Data Registers	N/A	N/A	DT<xxxxx> xxxxx-Decimal	DT00000-DT65535
Special Registers	N/A	N/A	DT<xxxxx> xxxxx-Decimal	DT90000-DT99999
Set Value Area (Timers / Counters)	N/A	N/A	SV<xxxx> xxxx-Decimal	SV0000-zV9999
Elapsed Value Area (Timers / Counters)	N/A	N/A	EV<xxxx> xxxx-Decimal	EV0000-EV9999
Index Registers	N/A	N/A	N/A	IX, IY, ID

Note: <y> bits are only valid between 0 and F hexadecimal. The bit reference used when accessing X,Y and R memory is only required when using the <xxx>.<y> address format. Normally direct access to X,Y and R can be done using standard Aromat Matsushita/NAIS addressing such as X50, Y122, or R140.

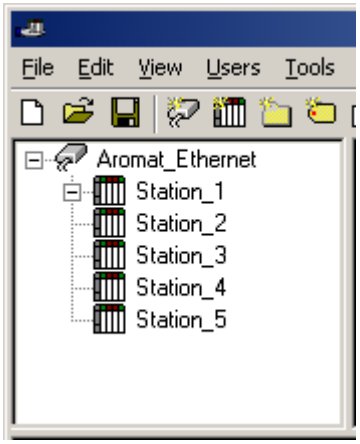
Examples

1. Y15 Output Relay 15.
2. T000 Timer Contact 0.
3. C127 Counter Contact 127.

Optimizing Aromat Ethernet Communications

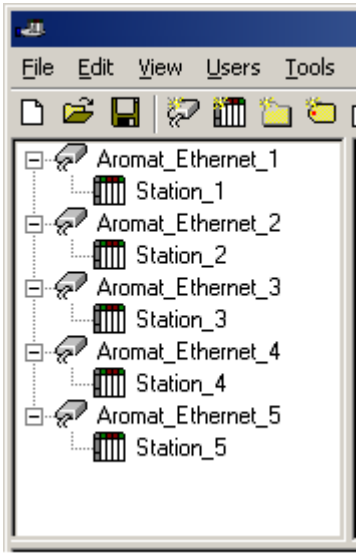
The Aromat Matsushita/NAIS Ethernet driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the Aromat Matsushita/NAIS Ethernet driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance.

Our server refers to communications protocols like Aromat Matsushita/NAIS Ethernet as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single Aromat Matsushita/NAIS controller from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the Aromat Matsushita/NAIS Ethernet driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Each device appears under a single Aromat Matsushita/NAIS Ethernet channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the Aromat Matsushita/NAIS Ethernet driver could only define one single channel, then the example shown above would be the only option available; however, the Aromat Matsushita/NAIS Ethernet driver can define up to 16 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has 16 or fewer devices, it can be optimized exactly how it is shown here.

The performance will improve even if the application has more than 16 devices. While 16 or fewer devices may be ideal, the application will still benefit from additional channels. Although by spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

Block Size, which is available on each defined device, can also affect the Aromat Matsushita/NAIS Ethernet driver's performance. Block Size refers to the number of bytes that may be requested from a device at one time. To refine the performance of this driver, configure Block Size to one of the following settings: 32, 64, 128, 256, or 512 bytes. Depending on the Aromat Matsushita/NAIS device model, the Block Size setting affect the application's performance drastically. A default value of 64 bytes is recommended. If an application has large requests for consecutively ordered data, however, block size should be increased.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only
Array size is out of range for address '<address>'
Array support is not available for the specified address: '<address>'

Device Status Messages

Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Driver Error Messages

Winsock initialization failed (OS Error = n)
Winsock V1.1 or higher must be installed to use the Aromat Matsushita/NAIS Ethernet device driver
Unable to bind IP '<IP in hex>' to Port '<Port>' - address is in use by another application
Response to <read/write> request to '<tag address>' on device '<device name>' contained error code '<xx>'
Response to <read/write> request to '<tag address>' on device '<device name>' had an unexpected format
Response to <read/write> request to '<tag address>' on device '<device name>' failed BCC check

Address Validation

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has no length.

Solution:
Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Address <address>' is out of range for the specified device or register

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically references a location that is beyond the range of supported locations for the device.

Solution:
Verify the address is correct; if it is not, re-enter it in the client application.

Data Type '<type>' is not valid for device address '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Array size is out of range for address '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically is requesting an array size that is too large for the address type or block size of the driver.

Solution:
Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically contains an array reference for an address type that doesn't support arrays.

Solution:
Re-enter the address in the client application to remove the array reference or correct the address type.

Device Status Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

'Device <Device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is broken.
2. The IP address assigned to the device is incorrect.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify the IP address given to the named device matches that of the actual device.

Unable to write to '<address>' on device '<device name>'

Error Type:
Serious

Possible Cause:
1. The connection between the device and the host PC is broken.
2. The named device may have been assigned an incorrect IP address.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify the IP address given to the named device matches that of the actual device.

Driver Error Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Driver Error Messages

Winsock initialization failed (OS Error = n)

Error Type:
Fatal

OS Error	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is not ready for network communication.	Wait a few seconds and restart the driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock V1.1 or higher must be installed to use the Aromat Matsushita/NAIS Ethernet device driver

Error Type:
Fatal

Possible Cause:
The version number of the Winsock DLL found on the system is less than 1.1.

Solution:
Upgrade Winsock to version 1.1 or higher.

Unable to bind IP '<IP in hex>' to Port '<Port>' - address is in use by another application

Error Type:

Serious

Possible Cause:
The driver was unable to bind the specified source Port and IP because another application is already using this combination of protocol (UDP or TCP/IP), IP, and Port number.

Solution:
Aromat Matsushita/NAIS Ethernet requires the driver to bind the connection's source IP and Port number when using UDP or TCP/IP Full Passive. If this is not possible, there are three solutions:

1. Determine what application is using the specified IP and Port and then re-configure it to use another address.
2. Use a different combination of source IP and Port.
3. Use the TCP/IP Unpassive connection type.

Response to <read/write> request to '<tag address>' on device '<device name>' contained error code '<xx>'

Error Type:
Serious

Possible Cause:
The device returned an error code in response to a read or write request.

Solution:
Consult the Aromat Matsushita/NAIS documentation for meaning of error code and take appropriate corrective measures.

Response to <read/write> request to '<tag address>' on device '<device name>' had an unexpected format

Error Type:
Serious

Possible Cause:
The response to a read or write request was not in the correct format.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. If problem does not appear to be related to noise, contact Technical Support.

Response to <read/write> request to '<tag address>' on device '<device name>' failed BCC check

Error Type:
Serious

Possible Cause:
The received byte check code was incorrect, indicating a corrupted response.

Solution:
Verify the cabling between the PC and the PLC device. Take appropriate shielding measures to reduce electrical noise.

Index

- A -

- B -

- C -

- D -

- E -

- F -

- L -

- M -

- N -

- O -

- R -

- S -

- U -

- W -