Omron Host Link Driver Help

Table of Contents

Omron Host Link Driver Help

Help version 1.024

CONTENTS

Overview

The Omron Host Link Driver provides an easy and reliable way to connect Omron Host Link devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with SYSMAC C-Series devices.

Device Setup

Supported Devices

C20H
C200H
CQM1
Open

Communication Protocol

Omron Host Link

Supported Communication Parameters

Baud Rate: 9600
Parity: Even or Odd
Data Bits: 7 or 8
Stop Bits: 1 or 2

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. It may be invoked through the COM ID dialog in Channel Properties. For more information, refer to the OPC server's help documentation.

Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 100. The maximum number of supported devices is 32.

Device IDs

The Host Link protocol supports up to 32 devices. Each device has its own Unique ID (unit number). The valid range is 0 to 31.

Flow Control

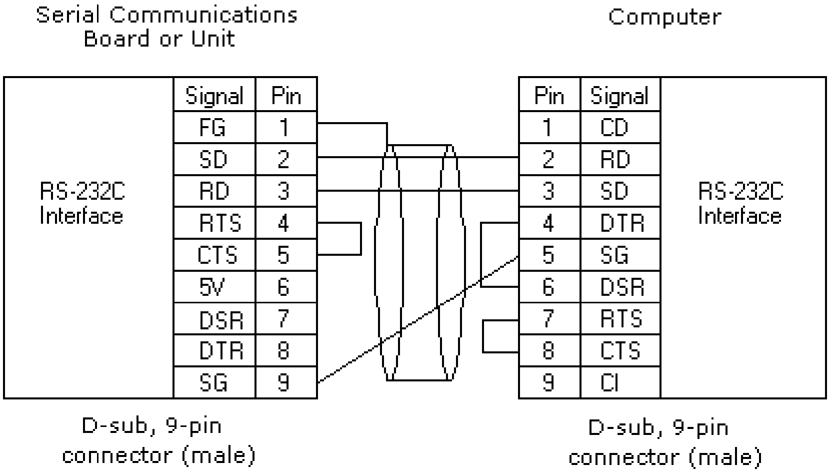
When using an RS232/RS485 converter, the type of flow control that is required depends on the needs of the converter. Some converters do not require any flow control, whereas others require RTS flow. Consult the converter's documentation in order to determine its flow requirements. An RS485 converter that provides automatic flow control is recommended.

Note 1: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control setting of **RTS** or **RTS Always** in the Channel Properties.

Note 2: When running on platforms that do not enforce proper flow control, it may be necessary to set the flow control in the server's communications settings.

Cable Diagrams

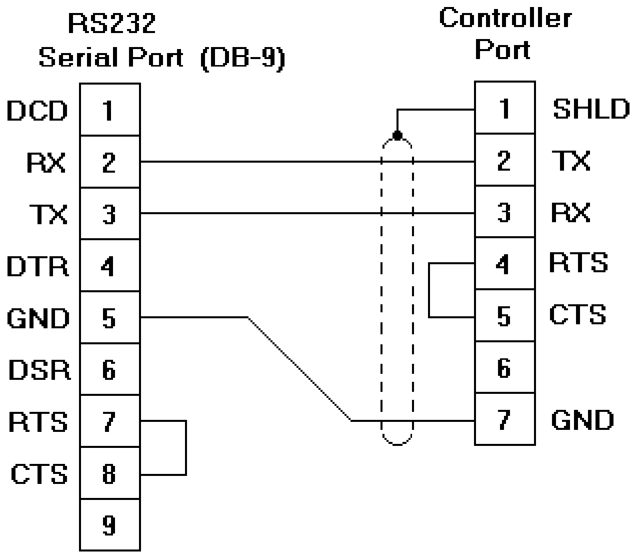
New Style


Serial Communications Board or Unit
Computer
RS-232C Interface
RS-232C Interface
D-sub, 9-pin connector (male)
D-sub, 9-pin connector (male)

Signal	Pin
FG	1
SD	2
RD	3
RTS	4
CTS	5
5V	6
DSR	7
DTR	8
SG	9

Pin	Signal
1	CD
2	RD
3	SD
4	DTR
5	SG
6	DSR
7	RTS
8	CTS
9	CI

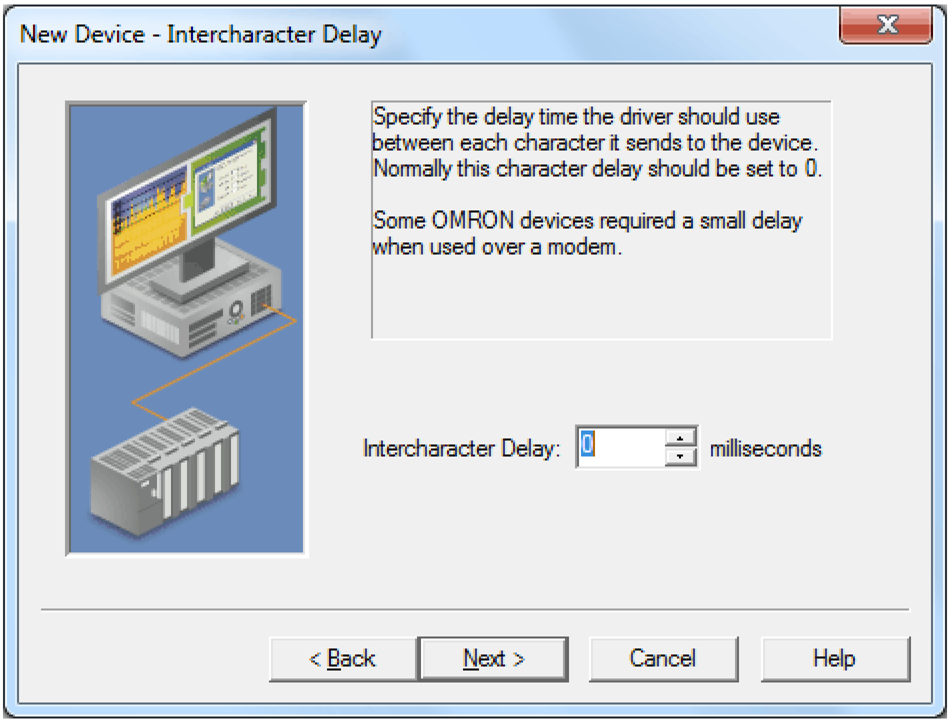
Old Style



Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC server's help documentation.

Inter-Character Delay


New Device - Intercharacter Delay

Specify the delay time the driver should use between each character it sends to the device. Normally this character delay should be set to 0.

Some OMRON devices required a small delay when used over a modem.

Intercharacter Delay: 0 milliseconds

Description of the parameter is as follows:

- **Intercharacter Delay:** This parameter specifies the controlled delay between each character that the Omron Host Link Driver sends to an Omron device. The valid range is 0 to 200 milliseconds. The default setting is 0 milliseconds.

  Note 1: It is generally recommended that this value be left at the default setting; however, users may need to adjust it for successful communications when using the driver over a modem. For a modem connection of 1200 baud, a typical setting would be 30 milliseconds.

Note 2: Although this delay is geared toward modem communications, each specific device may not need to use the Inter-Character Delay when communicating via modem. The delay introduced by this value will reduce the communication's speed.

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
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.
Long	Signed 32 bit value
DWord	Unsigned 32 bit value
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.
Float Example	If register DM100 is specified as a float, bit 0 of register DM100 would be bit 0 of the 32 bit float, and bit 15 of register DM101 would be bit 31 of the 32 bit float.
String	Null terminated ASCII string. Support includes string lengths up to 58 characters, and selection of HiLo byte order, LoHi byte order, Only High byte, and Only Low byte.

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

C20H Addressing
C200H Addressing
CQM1 Addressing
Open Addressing

C20H Addressing

The default data types are shown in **bold**. For more information, refer to **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR00.56H-AR27.02H .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR00.56L-AR27.02L .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as String using Only the High Order byte of each word	AR00.28D-AR27.01D .l is string length, range 1 to 28 chars	**String**	Read/Write
Auxiliary Relay as String using Only the Low Order byte of each word	AR00.28E-AR27.01E .l is string length, range 1 to 28 chars	**String**	Read/Write
Data Memory	DM0000-DM0999 DM0000-DM0998 DMxxxx.00-DMxxxx.15 DM1000-DM1999 DM1000-DM1998 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean **Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write Read Only
Data Memory as String with HiLo Byte Order	DM0000.58H-DM0999.02H .l is string length, range 2 to 58 chars DM1000.58H-DM1999.02H .l is string length, range 2 to 58 chars	**String** **String**	Read/Write Read Only
Data Memory as String with LoHi Byte Order	DM0000.58L-DM0999.02L .l is string length, range 2 to 58 chars DM1000.58L-DM1999.02L .l is string length, range 2 to 58	**String** **String**	Read/Write Read Only

	chars		
Data Memory as String using Only the High Order byte of each word	DM0000.29D-DM0999.01D .l is string length, range 1 to 29 chars DM1000.29D-DM1999.01D .l is string length, range 1 to 29 chars	**String** **String**	Read/Write Read Only
Data Memory as String using Only the Low Order byte of each word	DM0000.29E-DM0999.01E .l is string length, range 1 to 29 chars DM1000.29E-DM1999.01E .l is string length, range 1 to 29 chars	**String** **String**	Read/Write Read Only
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.58H-HR99.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String with LoHi Byte Order	HR00.58L-HR99.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String using Only the High Order byte of each word	HR00.29D-HR99.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay as String using Only the Low Order byte of each word	HR00.29E-HR99.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Internal Relay as String with HiLo Byte Order	IR000.58H-IR255.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String with LoHi Byte Order	IR000.58L-IR255.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String using Only the High Order byte of each word	IR000.29D-IR255.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay as String using Only the Low Order byte of each word	IR000.29E-IR255.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays	LR00-LR63 LR00-LR62	**Word**, Short, BCD, Long, DWord, Float, LBCD	Read/Write

	LRxx.00-LRxx.15	Boolean	
Link Relays as String with HiLo Byte Order	LR00.58H-LR63.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String with LoHi Byte Order	LR00.58L-LR63.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String using Only the High Order byte of each word	LR00.29D-LR63.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays as String using Only the Low Order byte of each word	LR00.29E-LR63.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Timer/Counter	RC000-RC511 RCxxx.00-RCxxx.15	Word, Short, **BCD** Boolean	Read/Write
Timer/Counter status	TC000-TC511	**Boolean**	Read/Write

String Support

The C20H model supports reading and writing auxiliary relay (AR), data memory (DM), holding relay (HR), internal relay (IR) and link relays (LR) as an ASCII string. When using any of these device types for string data, each register can contain either two bytes (two characters) of ASCII data or one. The order of the ASCII data within a given register, or the byte to use within a given register can be selected when the string is defined.

When using two bytes of ASCII data per register the length of the string can be from 2 to 58 characters (or 2 to 56 for AR) and is entered in place of a bit number. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register the length of the string can be from 1 to 29 characters (or 1 to 28 for AR) and is entered in place of a bit number. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address.

Examples
1. To address a string starting at DM100 with a length of 50 bytes and HiLo byte order, enter:
DM100.50H

2. To address a string starting at DM110 with a length of 8 bytes and LoHi byte order, enter:
DM110.08L

3. To address a string starting at DM200 with a length of 15 bytes and Only the High Order byte, enter:
DM200.15D

4. To address a string starting at DM220 with a length of 7 bytes and Only the Low Order byte, enter:
DM220.07E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes cannot exceed 116 bytes. This limit is imposed by the protocol. Since this driver uses an ASCII protocol, there are 4 bytes for each word, short and BCD, and 8 bytes for each DWord, long, LBCD and float. For example, a 4 X 7 array of words results in an array size of 28 words times 4 bytes for each word = 112 bytes, which would fit within the maximum request size of 116 bytes.

Note: Use caution when modifying 32 bit values (DWord, Long, LBCD, and Float). Each address that allows these data types will start at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. As an example, when using DWords, users may want to use DM0, DM2, DM4 and so on in order to prevent overlapping Words.

C200H Addressing

The default data types are shown in **bold**. For more information, refer to **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR000-AR999 AR000-AR998 ARxxx.00-ARxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR000.58H-AR999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR000.58L-AR999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Auxiliary Relay as String using Only the High Order byte of each word	AR000.29D-AR999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Auxiliary Relay as String using Only the Low Order byte of each word	AR000.29E-AR999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Data Memory	DM0000-DM9999 DM0000-DM9998 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Data Memory as String with HiLo byte order	DM0000.58H-DM0999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.58L-DM0999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String using Only the High Order byte of each word	DM0000.29D-DM0999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Data Memory as String using Only the Low Order byte of each word	DM0000.29E-DM0999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay	HR000-HR999 HR000-HR998 HRxxx.00-HRxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Holding Relay as String with HiLo byte order	HR000.58H- HR999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String with LoHi Byte Order	HR000.58L-HR999.02L	**String**	Read/Write

	.l is string length, range 2 to 58 chars		
Holding Relay as String using Only the High Order byte of each word	HR000.29D-HR999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay as String using Only the Low Order byte of each word	HR000.29E-HR999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay	IR000-IR999 IR000-IR998 IRxxx.00-IRxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Internal Relay as String with HiLo byte order	IR000.58H-IR999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String with LoHi Byte Order	IR000.58L-IR999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String using Only the High Order byte of each word	IR000.29D-IR999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay as String using Only the Low Order byte of each word	IR000.29E-IR999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays	LR000-LR999 LR000-LR998 LRxxx.00-LRxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Link Relays as String with HiLo Byte Order	LR000.58H-LR999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String with LoHi byte order	LR000.58L-LR999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String using Only the High Order byte of each word	LR000.29D-LR999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays as String using Only the Low Order byte of each word	LR000.29E-LR999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Timer/Counter	RC000-RC999 RCxxx.00-RCxxx.15	Word, Short, **BCD** Boolean	Read/Write
Timer/Counter status	TC000-TC999	**Boolean**	Read/Write

String Support

The C200H model supports reading and writing auxiliary relay (AR), data memory (DM), holding relay (HR), internal relay (IR) and link relays (LR) as an ASCII string. When using any of these device types for string data, each register can contain either two bytes (two characters) of ASCII data or one. The order of the ASCII data within a given register, or the byte to use within a given register can be selected when the string is defined.

When using two bytes of ASCII data per register the length of the string can be from 2 to 58 characters and is entered in place of a bit number. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register the length of the string can be from 1 to 29 characters and is entered in place of a bit number. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address.

Examples
1. To address a string starting at DM100 with a length of 50 bytes and HiLo byte order, enter:
DM100.50H

2. To address a string starting at DM110 with a length of 8 bytes and LoHi byte order, enter:
DM110.08L

3. To address a string starting at DM200 with a length of 15 bytes and Only the High Order byte, enter:
DM200.15D

4. To address a string starting at DM220 with a length of 7 bytes and Only the Low Order byte, enter:
DM220.07E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes cannot exceed 116 bytes. This limit is imposed by the protocol. Since this driver uses an ASCII protocol, there are 4 bytes for each word, short and BCD, and 8 bytes for each DWord, long, LBCD and float. For example, a 4 X 7 array of words results in an array size of 28 words times 4 bytes for each word = 112 bytes, which would fit within the maximum request size of 116 bytes.

Note: Use caution when modifying 32 bit values (DWord, Long, LBCD, and Float). Each address that allows these data types will start at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, for example, users may want to use DM0, DM2, DM4 and so on in order to prevent overlapping Words.

CQM1 Addressing

The default data types are shown in **bold**. For more information, refer to **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR00-AR27 AR00-AR26 ARxx.00-ARxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR00.56H-AR27.02H .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR00.56L-AR27.02L .l is string length, range 2 to 56 chars	**String**	Read/Write
Auxiliary Relay as String using Only the High Order byte of each word	AR00.28D-AR27.01D .l is string length, range 1 to 28 chars	**String**	Read/Write
Auxiliary Relay as String using	AR00.28E-AR27.01E	**String**	Read/Write

Only the Low Order byte of each word	.l is string length, range 1 to 28 chars		
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Data Memory as String with HiLo Byte Order	DM0000.58H-DM6655.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.58L-DM6655.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String using Only the High Order byte of each word	DM0000.29D-DM6655.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Data Memory as String using Only the Low Order byte of each word	DM0000.29E-DM6655.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Holding Relay as String with HiLo Byte Order	HR00.58H-HR99.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String with LoHi Byte Order	HR00.58L-HR99.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String using Only the High Order byte of each word	HR00.29D-HR99.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay as String using Only the Low Order byte of each word	HR00.29E-HR99.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Internal Relay as String with HiLo Byte Order	IR000.58H-IR255.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String with LoHi Byte Order	IR000.58L-IR255.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String using Only the High Order byte of each word	IR000.29D-IR255.01D .l is string length, range 1 to 29 chars	**String**	Read/Write

Internal Relay as String using Only the Low Order byte of each word	IR000.29E-IR255.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Link Relays as String with HiLo Byte Order	LR00.58H-LR63.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String with LoHi Byte Order	LR00.58L-LR63.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String using Only the High Order byte of each word	LR00.29D-LR63.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays as String using Only the Low Order byte of each word	LR00.29E-LR63.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Timer/Counter	RC000-RC511 RCxxx.00-RCxxx.15	Word, Short, **BCD** Boolean	Read/Write
Timer/Counter status	TC000-TC511	Boolean	Read/Write

String Support

The CQM1 model supports reading and writing auxiliary relay (AR), data memory (DM), holding relay (HR), internal relay (IR) and link relays (LR) as an ASCII string. When using any of these device types for string data, each register can contain either two bytes (two characters) of ASCII data or one. The order of the ASCII data within a given register, or the byte to use within a given register can be selected when the string is defined.

When using two bytes of ASCII data per register the length of the string can be from 2 to 58 characters (or 2 to 56 for AR) and is entered in place of a bit number. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register the length of the string can be from 1 to 29 characters (or 1 to 28 for AR) and is entered in place of a bit number. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address.

Examples

1. To address a string starting at DM100 with a length of 50 bytes and HiLo byte order, enter:
DM100.50H

2. To address a string starting at DM110 with a length of 8 bytes and LoHi byte order, enter:
DM110.08L

3. To address a string starting at DM200 with a length of 15 bytes and Only the High Order byte, enter:
DM200.15D

4. To address a string starting at DM220 with a length of 7 bytes and Only the Low Order byte, enter:
DM220.07E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes cannot exceed 116 bytes. This limit is imposed by the protocol. Since this driver uses an ASCII protocol, there are 4 bytes for each word, short and BCD, and 8 bytes for each DWord, long, LBCD and float. For example, a 4 X 7 array of words results in an array size of 28 words times 4 bytes for each word = 112 bytes, which would fit within the maximum request size of 116 bytes.

Note: Use caution when modifying 32 bit values (DWord, Long, LBCD, and Float). Each address that allows these data types will start at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, for example, users may want to use DM0, DM2, DM4 and so on in order to prevent overlapping Words.

Open Addressing

The following memory map is open for all memory types to support newer devices. Consult the manufacturer's documentation for device specific address ranges. The default data types are shown in **bold**. For more information, refer to **String Support** and **Array Support**.

Device Type	Range	Data Type	Access
Auxiliary Relay	AR0000-AR9999 AR0000-AR9998 ARxxxx.00-ARxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Auxiliary Relay as String with HiLo Byte Order	AR0000.58H-AR9999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Auxiliary Relay as String with LoHi Byte Order	AR0000.58L-AR9999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Auxiliary Relay as String using Only the High Order byte of each word	AR0000.29D-AR9999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Auxiliary Relay as String using Only the Low Order byte of each word	AR0000.29E-AR9999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Data Memory	DM0000-DM9999 DM0000-DM9998 DMxxxx.00-DMxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Data Memory as String with HiLo Byte Order	DM0000.58H-DM9999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String with LoHi Byte Order	DM0000.58L-DM9999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Data Memory as String using Only the High Order byte of each word	DM0000.29D-DM9999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Data Memory as String using Only the Low Order byte of each word	DM0000.29E-DM9999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay	HR0000-HR9999	**Word**, Short, BCD, Long,	Read/Write

	HR0000-HR9998 HRxxxx.00-HRxxxx.15	DWord, Float, LBCD Boolean	
Holding Relay as String with HiLo Byte Order	HR0000.58H-HR9999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String with LoHi Byte Order	HR0000.58L-HR9999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Holding Relay as String using Only the High Order byte of each word	HR0000.29D-HR9999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Holding Relay as String using Only the Low Order byte of each word	HR0000.29E-HR9999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay	IR0000-IR9999 IR0000-IR9998 IRxxxx.00-IRxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Internal Relay as String with HiLo Byte Order	IR0000.58H-IR9999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String with LoHi Byte Order	IR0000.58L-IR9999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Internal Relay as String using Only the High Order byte of each word	IR0000.29D-IR9999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Internal Relay as String using Only the Low Order byte of each word	IR0000.29E-IR9999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays	LR0000-LR9999 LR0000-LR9998 LRxxxx.00-LRxxxx.15	**Word**, Short, BCD, Long, DWord, Float, LBCD Boolean	Read/Write
Link Relays as String with HiLo Byte Order	LR0000.58H-LR9999.02H .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String with LoHi Byte Order	LR0000.58L-LR9999.02L .l is string length, range 2 to 58 chars	**String**	Read/Write
Link Relays as String using Only the High Order byte of each word	LR0000.29D-LR9999.01D .l is string length, range 1 to 29 chars	**String**	Read/Write
Link Relays as String using Only the Low Order byte of each word	LR0000.29E-LR9999.01E .l is string length, range 1 to 29 chars	**String**	Read/Write
Timer/Counter	RC0000-RC9999	Word, Short, **BCD**	Read/Write

	RCxxxx.00-RCxxxx.15	Boolean	
Timer/Counter Status*	TC0000-TC9999	**Boolean**	Read/Write

***Note:** Some models do not support writes to Timer/Counter status (TCxxxx).

String Support

The Open model supports reading and writing auxiliary relay (AR), data memory (DM), holding relay (HR), internal relay (IR) and link relays (LR) as an ASCII string. When using any of these device types for string data, each register can contain either two bytes (two characters) of ASCII data or one. The order of the ASCII data within a given register, or the byte to use within a given register can be selected when the string is defined.

When using two bytes of ASCII data per register the length of the string can be from 2 to 58 characters and is entered in place of a bit number. The length must be entered as an even number. The range of registers spanned by the string cannot exceed the range of the device type. The byte order is specified by appending either a "H" or "L" to the address.

When using one byte of ASCII data per register the length of the string can be from 1 to 29 characters and is entered in place of a bit number. The range of registers spanned by the string cannot exceed the range of the device type. The byte to use within a register is specified by appending either a "D" or "E" to the address.

Examples
1. To address a string starting at DM100 with a length of 50 bytes and HiLo byte order, enter:
DM100.50H

2. To address a string starting at DM110 with a length of 8 bytes and LoHi byte order, enter:
DM110.08L

3. To address a string starting at DM200 with a length of 15 bytes and Only the High Order byte,s enter:
DM200.15D

4. To address a string starting at DM220 with a length of 7 bytes and Only the Low Order byte, enter:
DM220.07E

Array Support

Arrays are supported for all data types except Boolean. There are two methods of addressing an array. Examples are given using data memory locations.

DMxxxx [rows] [cols]
*DMxxxx [cols]**

*This method assumes "rows" is equal to one.

Rows multiplied by cols multiplied by data size in bytes cannot exceed 116 bytes. This limit is imposed by the protocol. Since this driver uses an ASCII protocol, there are 4 bytes for each word, short and BCD, and 8 bytes for each DWord, long, LBCD and float. For example, a 4 X 7 array of words results in an array size of 28 words times 4 bytes for each word = 112 bytes, which would fit within the maximum request size of 116 bytes.

Note: Use caution when modifying 32 bit values (DWord, Long, LBCD, and Float). Each address that allows these data types will start at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Writing to DM0 will also modify the value held in DM1. It is recommended that users utilize these data types so that overlapping does not occur. When using DWords, for example, users may want to use DM0, DM2, DM4 and so on in order to prevent overlapping Words.

Error Descriptions

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

Address Validation

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>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address

Serial Communications

Communications error on '<channel name>' [<error mask>]
COMn does not exist
COMn is in use by another application
Error opening COMn
Unable to set comm parameters on COMn

Device Status Messages

Bad address in block [<start address> to <end address>], tag '<tag name>' on device '<device name>'
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Address Validation

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

Address Validation

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>' contains a syntax error
Device address '<address>' is not supported by model '<model name>'
Device address '<address>' is Read Only
Missing address

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

Error Type:
Warning

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

Solution:
Verify that 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 statically has been assigned an invalid data type.

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

Device address '<address>' contains a syntax error

Error Type:
Warning

Possible Cause:

A tag address that has been specified statically contains one or more invalid characters.

Solution:

Re-enter the address in the client application.

Device address '<address>' is not supported by model '<model name>'

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

Solution:

1. Verify that the address is correct; if it is not, re-enter it in the client application.
2. Verify that the selected model name for the device is correct.

Device address '<address>' is Read Only

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically 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.

Missing address

Error Type:

Warning

Possible Cause:

A tag address that has been specified statically has no length.

Solution:

Re-enter the address in the client application.

Serial Communications

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

Serial Communications

Communications error on '<channel name>' [<error mask>]

Error Type:

Serious

Error Mask Definitions:

B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.

P = Received byte parity error.
T = TX buffer full.

Possible Cause:

1. The serial connection between the device and the Host PC is bad.
2. The communications parameters for the serial connection are incorrect.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.

COMn does not exist

Error Type:

Fatal

Possible Cause:

The specified COM port is not present on the target computer.

Solution:

Verify that the proper COM port has been selected.

COMn is in use by another application

Error Type:

Fatal

Possible Cause:

The serial port assigned to a device is being used by another application.

Solution:

Verify that the correct port has been assigned to the channel.

Error opening COMn

Error Type:

Fatal

Possible Cause:

The specified COM port could not be opened due to an internal hardware or software problem on the target computer.

Solution:

Verify that the COM port is functional and may be accessed by other Windows applications.

Unable to set comm parameters on COMn

Error Type:

Fatal

Possible Cause:

The serial parameters for the specified COM port are not valid.

Solution:

Verify the serial parameters and make any necessary changes.

Device Status Messages

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

Device Status Messages

Bad address in block [<start address> to <end address>] on device '<device name>
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Bad address in block [<start address> to <end address>], tag '<tag name>' on device '<device name>'

Error Type:
Serious

Possible Cause:
An attempt has been made to reference a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device. Eliminate ones that reference invalid locations.

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

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

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Index