

Omron FINS Serial Driver Help

© 2013 Kepware Technologies

Table of Contents

Table of Contents	2
Omron FINS Serial Driver Help	4
Overview	4
Device Setup	5
Communication Parameters	6
Run Mode Settings	6
Modem Setup	7
FINS Networks	7
Data Types Description	11
Address Descriptions	12
C200H Addressing	12
C500 Addressing	16
C1000H Addressing	19
C2000H Addressing	22
CV500 Addressing	25
CV1000 Addressing	29
CV2000 Addressing	33
CVM1-CPU01 Addressing	37
CVM1-CPU11 Addressing	41
CVM1-CPU21 Addressing	44
CS1 Addressing	48
CJ1 Addressing	53
CJ2 Addressing	57
Error Descriptions	61
Address Validation	61
Address '<address>' is out of range for the specified device or register.....	62
Data Type '<type>' is not valid for device address '<address>'.....	62
Device address '<address>' contains a syntax error.....	62
Device address '<address>' is not supported by model '<model name>'.....	62
Device address '<address>' is Read Only.....	62
Missing address.....	62
Unable to write to '<address>' on device '<device name>'.....	63
Serial Communications	63
Communications error on '<channel name>' [<error mask>].....	63
COMn does not exist.....	63
COMn is in use by another application.....	64
Error opening COMn.....	64
Unable to set comm parameters on COMn.....	64
Omron FINS Serial Device Specific Messages	64

Device '<device name>' access right denied (Tag '<address>'). [Main, Sub: '<main code, sub code>'].... 65

Device '<device name>' cannot accept command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main ... code, sub code>']..... 65

Device '<device name>' cannot process command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main . code, sub code>']..... 65

Device '<device name>' deactivated due to network settings conflict with device '<device name>'..... 65

Device '<device name>' is not responding..... 65

Device '<device name>' responded with a command format error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']..... 66

Device '<device name>' responded with a command parameter error (Tag '<address>', Size '<bytes>'). ... [Main, Sub: '<main code, sub code>']..... 66

Device '<device name>' responded with a communications error. [Main, Sub: '<main code, sub code>'].. 66

Device '<device name>' responded with destination node error. [Main, Sub: '<main code, sub code>']... 66

Device '<device name>' responded with error '<error code>' (Tag '<address>', Size '<bytes>'). [Main, ... Sub: '<main code, sub code>']..... 67

Device '<device name>' responded with error in unit. [Main, Sub: '<main code, sub code>']..... 67

Device '<device name>' responded with Fatal CPU Unit error. [End Code: '<end code>']..... 67

Device '<device name>' responded with local node error. [Main, Sub: '<main code, sub code>']..... 67

Device '<device name>' responded with Non-fatal CPU Unit error. [End Code: '<end code>']..... 68

Device '<device name>' responded with read not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: . '<main code, sub code>']..... 68

Device '<device name>' responded with routing table error. [Main, Sub: '<main code, sub code>']..... 68

Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: . '<main code, sub code>']..... 68

The current project was created with an older version of this driver. Examine device properties to val-... idate network parameters..... 68

Main and Sub Error Codes..... 69

Index..... 76

Omron FINS Serial Driver Help

Help version 1.040

CONTENTS

[Overview](#)

What is the Omron FINS Serial Driver?

[Device Setup](#)

How do I configure a device for use with this driver?

[Data Types Description](#)

What data types does this driver support?

[Address Descriptions](#)

How do I address a data location on a Omron FINS Serial device?

[Error Descriptions](#)

What error messages does the Omron FINS Serial Driver produce?

Overview

The Omron FINS Serial Driver provides an easy and reliable way to connect Omron FINS Serial devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

Device Setup

Supported Devices

This driver supports the FINS protocol via the Sysmac Way: Host Link Interface. For a current list of models that support the Sysmac Way: Host Link Interface (reference: Omron's *CX-Server Runtime - User Manual*), refer to the manufacturer's web site.

Communication Protocol

Omron FINS

Supported Communication Parameters

Baud Rate: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 bps

Parity: Even, Odd, or None

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 server's help documentation.

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. To determine the converter's flow requirements, refer to its documentation. 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 Channel Properties.

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

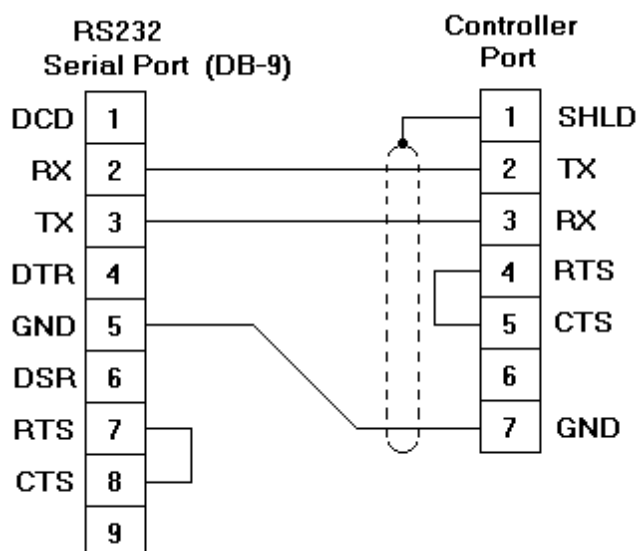
Device IDs

The FINS command system supports communication with devices on local and remote Omron networks. The Device ID is a three-layer network address that uniquely identifies the target device. The format of the Device ID is *UU.AAA.NNN*, where:

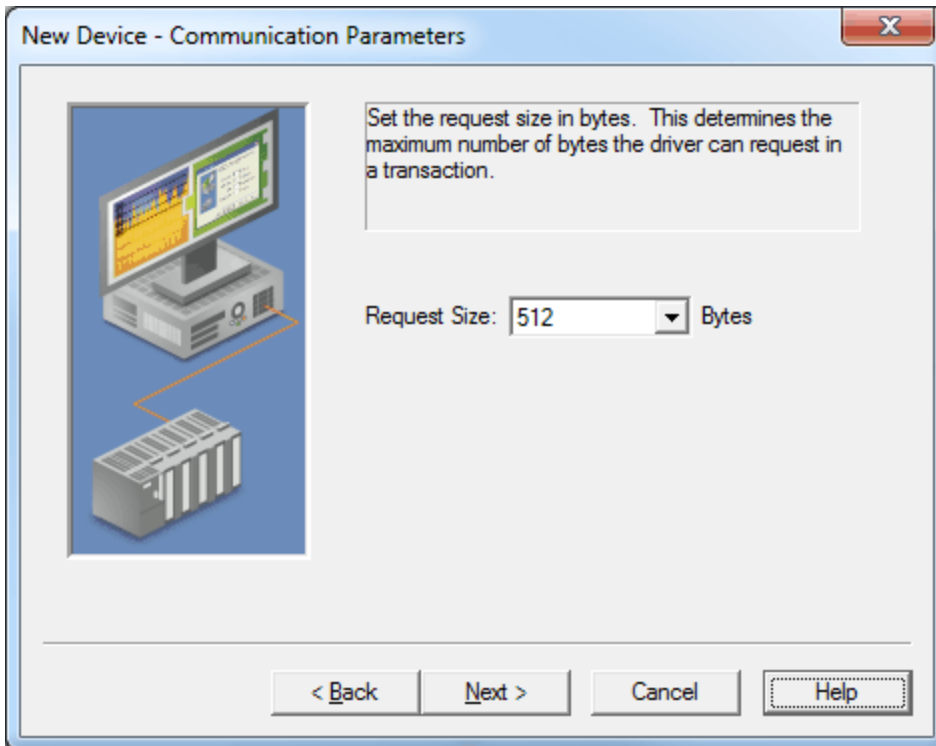
- **UU:** Unit Number of the Host Link Unit used for PC interface (0 to 31 decimal).
- **AAA:** FINS Destination Network Address (0 to 127 decimal).
- **NNN:** FINS Destination Node Number (0 to 254 decimal)

Note: For more information, refer to [FINS Networks](#).

Cable Diagram



Communication Parameters

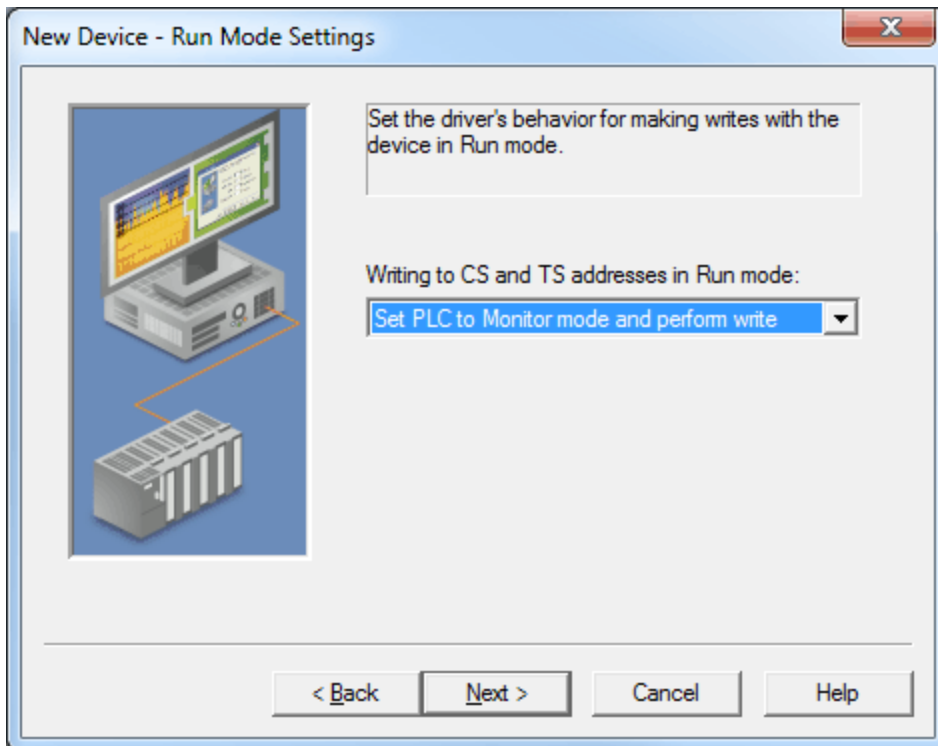


Description of the parameter is as follows:

- **Request Size:** This parameter specifies the number of bytes that may be requested from a device at one time. To refine the driver's performance, set the request size to one of the following settings: 32, 64, 128, 256, or 512 bytes. The default setting is 512 bytes.

Note: Because this driver uses an ASCII protocol, there are four bytes transmitted for each Word, Short, and BCD. Eight bytes are transmitted for each DWord, Long, LBCD, and Float.

Run Mode Settings



Description of the parameter is as follows:

- **Writing to CS and TS addresses in Run mode:** This parameter specifies the driver's behavior when the device makes writes while in Run mode. Options include Fail write and log message, Set PLC to Monitor mode and perform write, and Set PLC to Monitor mode, write, reset to Run. The default setting is Set PLC to Monitor mode and perform write. Descriptions of the options are as follows:
 - **Fail write and log message:** When selected, this option will fail the write and then post a message to the Event Log.
 - **Set PLC to Monitor mode and perform write:** When selected, the PLC will be set to Monitor mode before performing the write.
 - **Set PLC to Monitor mode, write, reset to Run:** When selected, the PLC will be set to Monitor mode before performing the write. Once complete, the PLC will reset to Run mode.

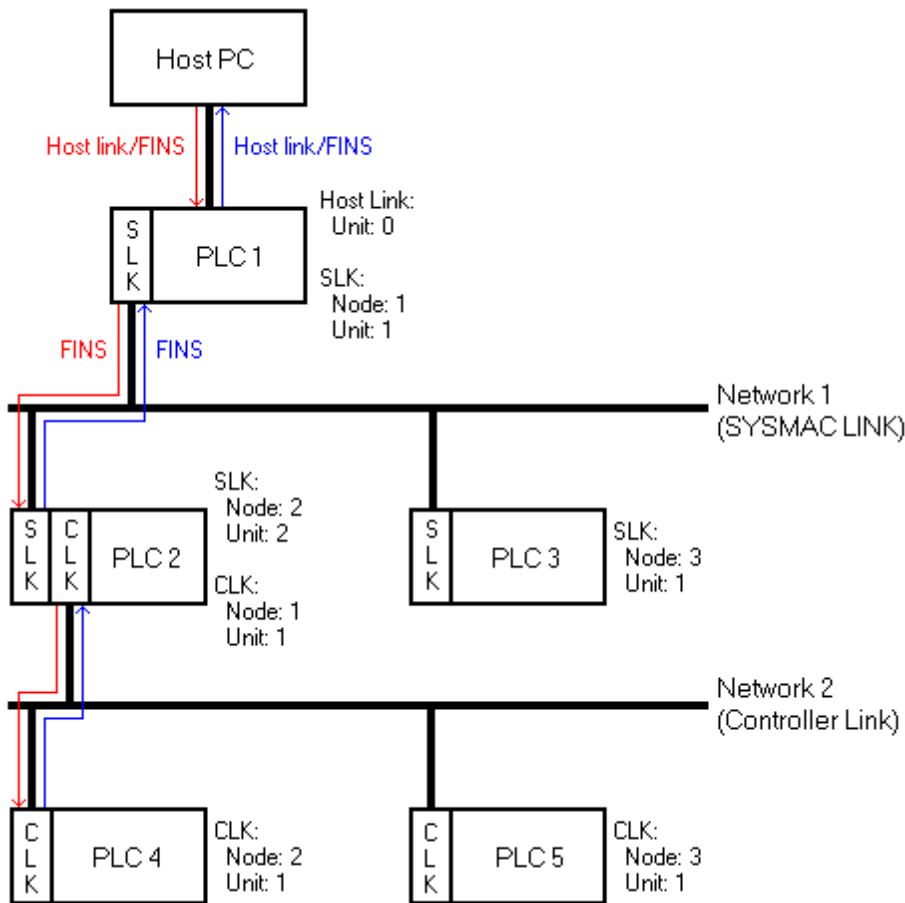
Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC Server Help documentation.

FINS Networks

The FINS communications service was developed by Omron to provide a consistent way for PLCs and computers on various networks to communicate. Compatible network types include Ethernet, Host Link, Controller Link, SYSMAC LINK, SYSMAC WAY, and Toolbus. FINS allows communications between nodes up to three network levels. Direct links between a PC and a PLC via Host link is not counted as a network layer.

The diagram below shows a FINS network comprised of interconnected SYSMAC LINK and Controller Link networks, and will serve as an example for the following discussion. FINS allows communication between any pair of devices in the diagram. PLC 1 serves as the host computer's Host Link interface to network 1. PLC 2 serves as a gateway between the networks 1 and 2. The Host PC sends a data request command (shown in red) to PLC 4 via PLC 1 and 2. The response is shown in blue. In principle, PLC 4 or 5 could act as a gateway to a third network layer that could also be reached by the Host PC.



FINS Messages

FINS messages contain two parts: a header and a data portion. The header contains source and destination information, among other things. The data portion contains command codes and optional command parameters. The six source and destination parameters contained in the header are as follows:

- **DNA:** Destination Network Address.
- **DA1:** Destination Node Number.
- **DA2:** Destination Module Address.
- **SNA:** Source Network Address.
- **SA1:** Source Node Number.
- **SA2:** Source Module Address.

This driver will always set DA2 and SA2 to zero, meaning that communications will be between the host computer and the destination node's CPU module. This driver also sets SNA and SA1 to zero to indicate that communications will be through the Host Link port of the interface device. The interface device (PLC 1) will reset SNA and SA1 as needed when forwarding messages to other PLCs.

Host Link Communications

This driver is able to communicate with any FINS compatible device with a Host Link port. That device can then process a given FINS command if it is the destination node, or relay the message to another device if it is not. In the figure above, is desired to send a data request from the Host PC to PLC 4. The driver constructs a FINS data request message with appropriate source and destination parameters set in the header, encapsulates that message in a Host Link wrapper, and sends it off to PLC 1. PLC 1 then examines the FINS message header and determines that node 2 on network 2 (PLC 4) is the intended destination. By referring to its routing tables, PLC 1 determines that the message must be sent to the gateway device PLC 2 in order to reach the destination node on network 2. Since PLC 2 is not on a Host Link network, the Host Link wrapper is removed. PLC 2 in turn sends the message to PLC 4, which sends its reply back to the Host PC via PLC 2 and 1. PLC 1 wraps the FINS response in a Host Link wrapper, which this driver can recognize and process.

For this example, the server must be configured as follows:

1. To start, create a channel that uses this driver. Then, create a device that represents the destination node (PLC 4).
2. Next, set the Device ID. The Host Link unit number of interface device is 0, the FINS destination network address (DNA) is 2, and the FINS destination node number (DA1) is 2. Therefore, the Device ID in this example is 0.2.2.

Note: As previously mentioned, SNA, SA1, SA2 and DA2 are automatically set to zero by the driver.

3. Similar device objects must be created for PLC 1, 2, 3, and 5.

Routing Tables

With multi-level networks, additional information must be programmed into the PLCs so they can send messages to other nodes in the system. This is accomplished using FINS routing tables. FINS routing tables come in two varieties: Local and Remote. Local routing tables associate a network number with one of the communication modules or Special Input/Output Units (SIOU) installed in the PLC's rack. Remote routing tables instruct how to reach one of the next network levels. The routing tables for this example would appear as follows:

PLC 1 (Local)

Network Number	Unit
1	1

Since PLC 1 is only connected to the SYSMAC LINK network, it only has one entry in its local routing table.

PLC 1 (Remote)

Remote Network Number	Relay Network	Relay Node
2	1	2

Network 2 is a remote network for PLC 1. To send a message to a node on network 2, PLC 1 must send that message to a relay node (or gateway) on one of its local networks. This local network is called the relay network, and must be network 1 in this example. The SYSMAC LINK module in the gateway (PLC 2) is node number 2. Thus, the relay node for PLC 1 is 2.

PLC 2 (Local)

Network Number	Unit
1	2
2	1

There are two local networks for PLC 2. The SYSMAC LINK and Controller Link network numbers are assigned as 1 and 2 respectively. The SYSMAC LINK and Controller Link modules are assigned unit numbers 2 and 1 respectively. The network node number of each of these modules is configured by the user, and must be unique within its associated network. In this example, both of the communications modules in PLC 2 could be called node 10 because they are on different networks.

PLC 3 (Local)

Network Number	Unit
1	1

The local routing table for PLC 3 has only one entry, because the device is only connected to network 1.

PLC 3 (Remote)

Remote Network Number	Relay Network	Relay Node
2	1	2

Network 2 is a remote network for PLC 3. Like PLC 1, the relay network is 1, and the relay node is 2 (the SLK module of the gateway device PLC 2).

Routing tables for PLC 4 and 5 must appear like the following. In this example, they are the same because the Controller Link modules in both PLCs are unit number 1. These modules must be assigned node numbers that are unique in network 2.

PLC 4 and 5 (Local)

Network Number	Unit
2	1

PLC 4 and 5 (Remote)

Remote Network Number	Relay Network	Relay Node
1	2	1

Note: For more information, refer to the Omron documentation.

Data Types Description

Data Type	Description
Boolean	Single bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 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
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Float	32-bit real
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.
String	Null terminated ASCII string. Support includes string lengths up to 256 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.

[C200H Addressing](#)

[C500 Addressing](#)

[C1000H Addressing](#)

[C2000H Addressing](#)

[CV500 Addressing](#)

[CV1000 Addressing](#)

[CV2000 Addressing](#)

[CVM1-CPU01 Addressing](#)

[CVM1-CPU11 Addressing](#)

[CVM1-CPU21 Addressing](#)

[CS1 Addressing](#)

[CJ1 Addressing](#)

[CJ2 Addressing](#)

C200H Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [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, LBCD, Float Boolean	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56 chars	String	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .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	AR00.028D-AR27.001D .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	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	String	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	DM0000.256H-DM6655.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	DM0000.256L-DM6655.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.128D-DM6655.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM0000.128E-DM6655.001E .l is string length, range 1 to 128 chars	String	Read/Write

	.l is string length, range 1 to 128 chars		
Expansion Data Memory (current bank)	EM0000-EM6143 EM0000-EM6142 EMxxx.00-EMxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	EM0000.256H-EM6143.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	EM0000.256L-EM6143.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	EM0000.128D-EM6143.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	EM0000.128E-EM6143.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	EM00:0000-EM07:6143 EM00:0000-EM07:6142 EMx:x.00-EMxx:xxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	EM00:0000.256H-EM07:6143.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	EM00:0000.256L-EM07:6143.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	EM00:0000.128D-EM07:6143.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	EM00:0000.128E-EM07:6143.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	String	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	String	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	String	Read/Write
Holding Relay as string	HR00.100E-HR99.001E	String	Read/Write

Using Only the Low Order byte of each word	.l is string length, range 1 to 100 chars		
Internal Relay	IR000-IR511 IR000-IR510 IRxxx.00-IRxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Internal Relay as a string with HiLo byte order	IR000.256H-IR511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string with LoHi byte order	IR000.256L-IR511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR000.128D-IR511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.128E-IR511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Link Relay as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	String	Read/Write
Link Relay as string with LoHI byte order	LR00.128L-LR63.002L .l is string length, range 2 to 128 chars	String	Read/Write
Link Relay as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	String	Read/Write
Link Relay as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64 chars	String	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer/Counter	TC000-TC511	BCD , Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.256H-TC511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string with LoHi byte order	TC000.256L-TC511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D-TC511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E-TC511.001E .l is string length, range 1 to 128 chars	String	Read/Write

Timer/Counter Status	TS000-TS511	Boolean	Read/Write
----------------------	-------------	---------	------------

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C200H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H

2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L

3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D

4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

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 the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use DM0, DM2, DM4, and so on to prevent overlapping Words.

C500 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Data Memory	DM000-DM511 DM000-DM510 DMxxx.00-DMxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	DM000.256H-DM511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	DM000.256L-DM511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM000.128D-DM511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM000.128E-DM511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	HR00-HR31 HR00-HR30 HRxx.00-HRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Holding Relay as string with HiLo byte order	HR00.064H-HR31.002H .l is string length, range 2 to 64 chars	String	Read/Write
Holding Relay as string with LoHi byte order	HR00.064L-HR31.002L .l is string length, range 2 to 64 chars	String	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.032D-HR31.001D .l is string length, range 1 to 32 chars	String	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.032E-HR31.001E .l is string length, range 1 to 32 chars	String	Read/Write
Internal Relay	IR00-IR63 IR00-IR62 IRxx.00-IRxx.15	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write

		Boolean	
Internal Relay as a string with HiLo byte order	IR00.128H-IR63.002H .l is string length, range 2 to 128 chars	String	Read/Write
Internal Relay as a string with LoHi byte order	IR00.128L-IR63.002L .l is string length, range 2 to 128 chars	String	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR00.064D-IR63.001D .l is string length, range 1 to 64 chars	String	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR00.064E-IR63.001E .l is string length, range 1 to 64 chars	String	Read/Write
Link Relays	LR00-LR31 LR00-LR30 LRxx.00-LRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Link Relays as string with HiLo byte order	LR00.064H-LR31.002H .l is string length, range 2 to 64 chars	String	Read/Write
Link Relays as string with LoHi byte order	LR00.064L-LR31.002L .l is string length, range 2 to 64 chars	String	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.032D-LR31.001D .l is string length, range 1 to 32 chars	String	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.032E-LR31.001E .l is string length, range 1 to 32 chars	String	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer/Counter	TC000-TC127	BCD , Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.256H-TC127.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string with LoHi byte order	TC000.256L-TC127.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D-TC127.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E-TC127.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter Status	TS000-TS127	Boolean	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C500 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at DM100 with a length of 100 bytes and HiLo byte order, enter:
DM100.100H
2. To address a string starting at DM110 with a length of 78 bytes and LoHi byte order, enter:
DM110.078L
3. To address a string starting at DM200 with a length of 55 bytes and Only the High Order byte, enter:
DM200.055D
4. To address a string starting at DM220 with a length of 37 bytes and Only the Low Order byte, enter:
DM220.037E

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 the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use DM0, DM2, DM4, and so on to prevent overlapping Words.

C1000H Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [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, LBCD, Float Boolean	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56 chars	String	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .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	AR00.028D-AR27.001D .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	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	String	Read/Write
Data Memory	DM0000-DM4095 DM0000-DM4094 DMxxxx.00-DMxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	DM0000.256H-DM4095.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	DM0000.256L-DM4095.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.128D-DM4095.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	DM0000.128E-DM4095.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	String	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	String	Read/Write

	.l is string length, range 2 to 200 chars		
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	String	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E .l is string length, range 1 to 100 chars	String	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Internal Relay as a string with HiLo byte order	IR000.256H-IR255.002H .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string with LoHi byte order	IR000.256L-IR255.002L .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR000.128D-IR255.001D .l is string length, range 1 to 128 chars	String	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.128E-IR255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Link Relays as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	String	Read/Write
Link Relays as string with LoHi byte order	LR00.128L-LR63.002L .l is string length, range 2 to 128 chars	String	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	String	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64 chars	String	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer/Counter	TC000-TC511	BCD , Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.256H-TC511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string with LoHi byte order	TC000.256L-TC511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string	TC000.128D-TC511.001D	String	Read/Write

Using Only the High Order byte of each word	.l is string length, range 1 to 128 chars		
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E-TC511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter Status	TS000-TS511	Boolean	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C1000H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H

2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L

3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D

4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

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 the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use DM0, DM2, DM4, and so on to prevent overlapping Words.

C2000H Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [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, LBCD, Float Boolean	Read/Write
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H .l is string length, range 2 to 56 chars	String	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L .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	AR00.028D-AR27.001D .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	AR00.028E-AR27.001E .l is string length, range 1 to 28 chars	String	Read/Write
Data Memory	DM0000-DM6655 DM0000-DM6654 DMxxxx.00-DMxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	DM0000.256H-DM6655.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	DM0000.256L-DM6655.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	DM0000.128D-DM6655.001D .l is string length, range 1 to 128 chars	String	Read/Write

Data Memory as string Using Only the Low Order byte of each word	DM0000.128E-DM6655.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	HR00-HR99 HR00-HR98 HRxx.00-HRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H .l is string length, range 2 to 200 chars	String	Read/Write
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L .l is string length, range 2 to 200 chars	String	Read/Write
Holding Relay as string Using Only the High Order byte of each word	HR00.100D-HR99.001D .l is string length, range 1 to 100 chars	String	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E .l is string length, range 1 to 100 chars	String	Read/Write
Internal Relay	IR000-IR255 IR000-IR254 IRxxx.00-IRxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Internal Relay as a string with HiLo byte order	IR000.256H-IR255.002H .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string with LoHi byte order	IR000.256L-IR255.002L .l is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string Using Only the High Order byte of each word	IR000.128D-IR255.001D .l is string length, range 1 to 128 chars	String	Read/Write
Internal Relay as a string Using Only the Low Order byte of each word	IR000.128E-IR255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Link Relays	LR00-LR63 LR00-LR62 LRxx.00-LRxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Link Relays as string with HiLo byte order	LR00.128H-LR63.002H .l is string length, range 2 to 128 chars	String	Read/Write
Link Relays as string with LoHi byte order	LR00.128L-LR63.002L .l is string length, range 2 to 128 chars	String	Read/Write
Link Relays as string Using Only the High Order byte of each word	LR00.064D-LR63.001D .l is string length, range 1 to 64 chars	String	Read/Write
Link Relays as string Using Only the Low Order byte of each word	LR00.064E-LR63.001E .l is string length, range 1 to 64	String	Read/Write

	chars		
Temporary Relay	TR TR0-TR7	Word, Short Boolean	Read/Write
Timer/Counter	TC000-TC511	BCD, Word, Short	Read/Write
Timer/Counter as string with HiLo byte order	TC000.256H-TC511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string with LoHi byte order	TC000.256L-TC511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D-TC511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E-TC511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer/Counter Status	TS000-TS511	Boolean	Read/Write

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The C2000H model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at DM1000 with a length of 100 bytes and HiLo byte order, enter:
DM1000.100H
2. To address a string starting at DM1100 with a length of 78 bytes and LoHi byte order, enter:
DM1100.078L
3. To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter:
DM2000.055D
4. To address a string starting at DM2200 with a length of 37 bytes and Only the Low Order byte, enter:
DM2200.037E

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 the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords DM0 and DM1 overlap at word DM1. Thus, writing to DM0 will also modify the value held in DM1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use DM0, DM2, DM4, and so on to prevent overlapping Words.

CV500 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC1023	Boolean	Read Only
Auxiliary Relay	A000-A255	Word , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	Word , Short, BCD	Read Only
	A256-A510	Long, DWord, LBCD, Float	
Auxiliary Relay as string with HiLo byte order	A000.00-A000.15-A255.00-A255.15	Boolean	Read/Write
	A256.00-A256.15-A511.00-A511.15	Boolean	Read Only
	A000.256H-A255.002H A256.256H-A511.002H .l is string length, range 2 to 256 chars	String String	Read/Write Read Only

Auxiliary Relay as string with LoHi byte order	A000.256L-A255.002L A256.256L-A511.002L .l is string length, range 2 to 256 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A255.001D A256.128D-A511.001D .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A255.001E A256.128E-A511.001E .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO2555.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO2555.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C000-C511	BCD , Word, Short	Read/Write
Counter as string with HiLo byte order	C000.256H-C511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C000.256L-C511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C000.128D-C511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C000.128E-C511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS000-CS511	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L	String	Read/Write

	.l is string length, range 2 to 256 chars		
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D0000.256H-D8191.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D0000.256L-D8191.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D0000.128D-D8191.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D0000.128E-D8191.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR0-DR2	Word , Short, BCD*	Read/Write
Index Register	IR0-IR2	Word , Short, BCD*	Read/Write
Step Timer	ST000-ST511	Word , Short, BCD*	Read/Write
Step Timer Status	STS000-STS511	Boolean	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer	T000-T511	BCD , Word, Short	Read/Write
Timer as string with HiLo byte order	T000.256H-T511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T000.256L-T511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T000.128D-T511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T000.128E-T511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS000-TS511	Boolean	Read/Write
Transition Flag	TN000-TN511	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV500 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D1000 with a length of 100 bytes and HiLo byte order, enter:
D1000.100H
2. To address a string starting at D1100 with a length of 78 bytes and LoHi byte order, enter:
D1100.078L
3. To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter:
D2000.055D
4. To address a string starting at D2200 with a length of 37 bytes and Only the Low Order byte, enter:
D2200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CV1000 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC2047	Boolean	Read Only
Auxiliary Relay	A000-A255	Word , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	Word , Short, BCD	Read Only
	A256-A510	Long, DWord, LBCD, Float	
	A000.00-A000.15-A255.00-A255.15	Boolean	Read/Write
	A256.00-A256.15-A511.00-A511.15	Boolean	Read Only
Auxiliary Relay as string with HiLo byte order	A000.256H-A255.002H	String	Read/Write
	A256.256H-A511.002H	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte order	A000.256L-A255.002L	String	Read/Write
	A256.256L-A511.002L	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A255.001D	String	Read/Write
	A256.128D-A511.001D	String	Read Only
	.I is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A255.001E	String	Read/Write
	A256.128E-A511.001E	String	Read Only
	.I is string length, range 1 to 128 chars		
CIO	CIO0000-CIO2555	Word , Short, BCD	Read/Write
	CIO0000-CIO2554	Long, DWord, LBCD, Float	
	CIOxxxx.00-CIOxxxx.15	Boolean	
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO2555.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		

CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO2555.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C1023	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H-C1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C0000.256L-C1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.128D-C1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.128E-C1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS0000-CS1023	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D24575.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D24575.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D24575.001D .l is string length, range 1 to 128 chars	String	Read/Write

Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D24575.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR0-DR2	Word, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32765.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128D-E32765.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H - E07:32765.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.256L - E07:32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D- E07:32765.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E- E07:32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Index Register	IR0-IR2	Word, Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	Word, Short, BCD*	Read/Write
Step Timer Status	STS0000-ST51023	Boolean	Read/Write
Temporary Relay	TR TR0-TR7	Word, Short Boolean	Read/Write
Timer	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L .l is string length, range 2 to 256	String	Read/Write

	chars		
Timer as string Using Only the High Order byte of each word	T0000.128D-T1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.128E-T1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS0000-TS1023	Boolean	Read/Write
Transition Flag	TN0000-TN1023	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IROD @ Short = -50
IROD @ Word = 8050 (MSB set)
IRO @ BCD = 8050

IROD @ Short = 50
IROD @ Word = 50
IRO @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV1000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H

2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L

3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D

4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]

Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CV2000 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Action Flag	AC0000-AC2047	Boolean	Read Only
Auxiliary Relay	A000-A255	Word , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	Word , Short, BCD	Read Only
	A256-A510	Long, DWord, LBCD, Float	
Auxiliary Relay as string with HiLo byte order	A000.00-A000.15-A255.00-A255.15	Boolean	Read/Write
	A256.00-A256.15-A511.00-A511.15	Boolean	Read Only
	A000.256H-A255.002H A256.256H-A511.002H	String String	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	.l is string length, range 2 to 256 chars		
	A000.256L-A255.002L A256.256L-A511.002L	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	.l is string length, range 2 to 256 chars		
	A000.128D-A255.001D A256.128D-A511.001D	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	.l is string length, range 1 to 128 chars		
	A000.128E-A255.001E A256.128E-A511.001E	String String	Read/Write Read Only

	.l is string length, range 1 to 128 chars		
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO2555.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO2555.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C1023	BCD , Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H-C1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C0000.256L-C1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.128D-C1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.128E-C1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS0000-CS1023	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	Word , Short, BCD Long, DWord, LBCD,	Read/Write

		Float Boolean	
Data Memory as string with HiLo byte order	D00000.256H-D24575.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D24575.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D24575.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D24575.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR0-DR2	Word , Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32765.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128D-E32765.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H - E07:32765.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.256L - E07:32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D- E07:32765.001D .l is string length, range 1 to 128 chars	String	Read/Write

Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E- E07:32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Index Register	IR0-IR2	Word, Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	Word, Short, BCD*	Read/Write
Step Timer Status	STS0000-ST1023	Boolean	Read/Write
Temporary Relay	TR TR0-TR7	Word, Short Boolean	Read/Write
Timer	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.128D-T1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.128E-T1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS0000-TS1023	Boolean	Read/Write
Transition Flag	TN0000-TN1023	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IROD @ Short = -50
IROD @ Word = 8050 (MSB set)
IRO @ BCD = 8050

IROD @ Short = 50
IROD @ Word = 50
IRO @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CV2000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

Arrays are supported for all data types except Boolean, Data Register, Index Register and Step Timer. There are two methods of addressing an array. Examples are given using data memory locations.

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CVM1-CPU01 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255	Word , Short, BCD	Read/Write
	A000-A254	Long, DWord, LBCD, Float	
	A256-A511	Word , Short, BCD	Read Only
	A256-A510	Long, DWord, LBCD, Float	
	A000.00-A000.15-A255.00-A255.15	Boolean	Read Only

	A256.00-A256.15-A511.00-A511.15	Boolean	
Auxiliary Relay as string with HiLo byte order	A000.256H-A255.002H A256.256H-A511.002H .l is string length, range 2 to 256 chars	String String	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.256L-A255.002L A256.256L-A511.002L .l is string length, range 2 to 256 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A255.001D A256.128D-A511.001D .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A255.001E A256.128E-A511.001E .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO2555.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO2555.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C000-C511	BCD , Word, Short	Read/Write
Counter as string with HiLo byte order	C000.256H-C511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C000.256L-C511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C000.128D-C511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C000.128E-C511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS000-CS511	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write

		Boolean	
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D0000-D8191 D0000-D8190 Dxxxx.00-Dxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D0000.256H-D8191.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D0000.256L-D8191.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D0000.128D-D8191.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D0000.128E-D8191.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR0-DR2	Word , Short, BCD*	Read/Write
Index Register	IR0-IR2	Word , Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer	T000-T511	BCD , Word, Short	Read/Write
Timer as string with HiLo byte order	T000.256H-T511.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T000.256L-T511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T000.128D-T511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T000.128E-T511.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS000-TS511	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IROD @ Short = -50
IROD @ Word = 8050 (MSB set)
IRO @ BCD = 8050

IROD @ Short = 50
IROD @ Word = 50
IRO @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CVM1-CPU01 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D1000 with a length of 100 bytes and HiLo byte order, enter:
D1000.100H
2. To address a string starting at D1100 with a length of 78 bytes and LoHi byte order, enter:
D1100.078L
3. To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter:
D2000.055D
4. To address a string starting at D2200 with a length of 37 bytes and Only the Low Order byte, enter:
D2200.037E

Array Support

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

Dxxxx [rows] [cols]

Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CVM1-CPU11 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255 A000-A254	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A256-A511 A256-A510	Word , Short, BCD Long, DWord, LBCD, Float	Read Only
	A000.00-A000.15-A255.00- A255.15	Boolean	Read/Write
	A256.00-A256.15-A511.00- A511.15	Boolean	Read Only
Auxiliary Relay as string with HiLo byte order	A000.256H-A255.002H A256.256H-A511.002H	String String	Read/Write Read Only
	.l is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte order	A000.256L-A255.002L A256.256L-A511.002L	String String	Read/Write Read Only
	.l is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A255.001D A256.128D-A511.001D	String String	Read/Write Read Only
	.l is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A255.001E A256.128E-A511.001E	String String	Read/Write Read Only
	.l is string length, range 1 to 128 chars		
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
	CIO0000.256H-CIO2555.002H	String	Read/Write
	.l is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L	String	Read/Write
	.l is string length, range 2 to 256 chars		
CIO Memory as string	CIO0000.128D-CIO2555.001D	String	Read/Write

Using Only the High Order byte of each word	.l is string length, range 1 to 128 chars		
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C1023	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H-C1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C0000.256L-C1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.128D-C1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.128E-C1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS0000-CS1023	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	Word, Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D24575.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D24575.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D24575.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string	D00000.128E-D24575.001E	String	Read/Write

Using Only the Low Order byte of each word	.l is string length, range 1 to 128 chars		
Data Register	DR0-DR2	Word , Short, BCD*	Read/Write
Index Register	IR0-IR2	Word , Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	Word , Short Boolean	Read/Write
Timer	T0000-T1023	BCD , Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.128D-T1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.128E-T1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS0000-TS1023	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50
IR0D @ Word = 8050 (MSB set)
IR0 @ BCD = 8050

IR0D @ Short = 50
IR0D @ Word = 50
IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CVM1-CPU11 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the

ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

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

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CVM1-CPU21 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A255 A000-A254	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A256-A511 A256-A510	Word , Short, BCD Long, DWord, LBCD, Float	Read Only
	A000.00-A000.15-A255.00- A255.15	Boolean	Read/Write
	A256.00-A256.15-A511.00- A511.15	Boolean	Read Only

Auxiliary Relay as string with HiLo byte order	A000.256H-A255.002H A256.256H-A511.002H .l is string length, range 2 to 256 chars	String String	Read/Write Read Only
Auxiliary Relay as string with LoHi byte order	A000.256L-A255.002L A256.256L-A511.002L .l is string length, range 2 to 256 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A255.001D A256.128D-A511.001D .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A255.001E A256.128E-A511.001E .l is string length, range 1 to 128 chars	String String	Read/Write Read Only
CIO	CIO0000-CIO2555 CIO0000-CIO2554 CIOxxx.00-CIOxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO2555.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO2555.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO2555.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO2555.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C1023	BCD , Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H-C1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C0000.256L-C1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.128D-C1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string Using Only the Low Order byte of each word	C0000.128E-C1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter Status	CS0000-CS1023	Boolean	Read/Write
CPU Bus Link	G000-G255 G000-G254 Gxxx.00-Gxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write

CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L .l is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string Using Only the High Order byte of each word	G000.128D-G255.001D .l is string length, range 1 to 128 chars	String	Read/Write
CPU Bus Link as string Using Only the Low Order byte of each word	G000.128E-G255.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory	D00000-D24575 D00000-D24574 Dxxxxx.00-Dxxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D24575.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D24575.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D24575.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D24575.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR0-DR2	Word , Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765 E00000-E32764 Exxxxx.00-Exxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32765.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128D-E32765.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E07:32765 E00:00000-E07:32764 Ex:x.00-Exx:xxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H-	String	Read/Write

	E07:32765.002H .l is string length, range 2 to 256 chars		
Expansion Data Memory as string with LoHi byte order	E00:00000.256L-E07:32765.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D-E07:32765.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E-E07:32765.001E .l is string length, range 1 to 128 chars	String	Read/Write
Index Register	IR0-IR2	Word, Short, BCD*	Read/Write
Temporary Relay	TR TR0-TR7	Word, Short Boolean	Read/Write
Timer	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.128D-T1023.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.128E-T1023.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS0000-TS1023	Boolean	Read/Write

*Arrays are not supported.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

IR0D @ Short = -50

IR0D @ Word = 8050 (MSB set)
 IR0 @ BCD = 8050

IR0D @ Short = 50
 IR0D @ Word = 50
 IR0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CCM1-CPU21 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

- To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
- To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
- To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
- To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

Array Support

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

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Note: Use caution when modifying 32-bit values (DWord, Long, LBCD, and Float). Each address, for which these data types are allowed, starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1. Thus, writing to D0 will also modify the value held in D1. It is recommended that these data types be used so that overlapping does not occur. For example, when using DWords, use D0, D2, D4, and so on to prevent overlapping Words.

CS1 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446	Word , Short, BCD Long, DWord, LBCD,	Read Only

	A448-A959 A448-A958 A000.00-A000.15-A447.00-A447.15 A448.00-A448.15-A959.00-A959.15	Float Word , Short, BCD Long, DWord, LBCD, Float Boolean Boolean	Read/Write Read Only Read/Write
Auxiliary Relay as string with HiLo byte order	A000.256H-A447.002H A448.256H-A959.002H .l is string length, range 2 to 256 chars	String String	Read Only Read/Write
Auxiliary Relay as string with LoHi byte order	A000.256L-A447.002L A448.256L-A959.002L .l is string length, range 2 to 256 chars	String String	Read Only Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A447.001D A448.128D-A959.001D .l is string length, range 1 to 128 chars	String String	Read Only Read/Write
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A447.001E A448.128E-A959.001E .l is string length, range 1 to 128 chars	String String	Read Only Read/Write
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO6143.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO6143.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO6143.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO6143.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C4095	BCD , Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H-C4095.002H .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string with LoHi byte order	C0000.256L-C4095.002L .l is string length, range 2 to 256 chars	String	Read/Write
Counter as string Using Only the High Order byte of each word	C0000.128D-C4095.001D .l is string length, range 1 to 128 chars	String	Read/Write
Counter as string	C0000.128E-C4095.001E	String	Read/Write

Using Only the Low Order byte of each word	.l is string length, range 1 to 128 chars		
Counter Status	CS0000-CS4095	Boolean	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR00-DR15 DR00-DR14	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128E-E32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Exx:xxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H - E12:32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.256L - E12:32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as	E00:00000.128D-	String	Read/Write

string Using Only the High Order byte of each word	E12:32767.001D .l is string length, range 1 to 128 chars		
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E- E12:32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Holding Relay as string with HiLo byte order	H0000.256H-H1535.002H .l is string length, range 2 to 256 chars	String	Read/Write
Holding Relay as string with LoHi byte order	H0000.256L-H1535.002L .l is string length, range 2 to 256 chars	String	Read/Write
Holding Relay as string Using Only the High Order byte of each word	H0000.128D-H1535.001D .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay as string Using Only the Low Order byte of each word	H0000.128E-H1535.001E .l is string length, range 1 to 128 chars	String	Read/Write
Index Register	IR00-IR15	DWord , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	Boolean	Read Only
Timer	T0000-T4095	BCD , Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T4095.002H .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T4095.002L .l is string length, range 2 to 256 chars	String	Read/Write
Timer as string Using Only the High Order byte of each word	T0000.128D-T4095.001D .l is string length, range 1 to 128 chars	String	Read/Write
Timer as string Using Only the Low Order byte of each word	T0000.128E-T4095.001E .l is string length, range 1 to 128 chars	String	Read/Write
Timer Status	TS0000-TS4095	Boolean	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Working Relay as string with HiLo byte order	W000.256H-W511.002H .l is string length, range 2 to 256 chars	String	Read/Write

Working Relay as string with LoHi byte order	W000.256L-W511.002L .l is string length, range 2 to 256 chars	String	Read/Write
Working Relay as string Using Only the High Order byte of each word	W000.128D-W511.001D .l is string length, range 1 to 128 chars	String	Read/Write
Working Relay as string Using Only the Low Order byte of each word	W000.128E-W511.001E .l is string length, range 1 to 128 chars	String	Read/Write

*Use caution when modifying 32-bit values such as DWord, Long, LBCD, and Float. Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1: writing to D0 will also modify the value held in D1. It is recommended that these data types be used in order to prevent overlapping. For example, to prevent overlapping Words, users should use D0, D2, D4, and so on for DWords. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CS1-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CS1 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been

assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

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.

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Writing to DR and IR Registers

DR and IR registers can be written to only when the device is in Programming Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. Because the write will succeed, no error message be returned. The device's value will not change because the device is in Run Mode.

CJ1 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446	Word , Short, BCD Long, DWord, LBCD, Float	Read Only
	A448-A959 A448-A958	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A000.00-A000.15-A447.00- A447.15	Boolean	Read Only
	A448.00-A448.15-A959.00- A959.15	Boolean	Read/Write
Auxiliary Relay as string with HiLo byte order	A000.256H-A447.002H A448.256H-A959.002H .l is string length, range 2 to 256 chars	String String	Read Only Read/Write
	A000.256L-A447.002L A448.256L-A959.002L .l is string length, range 2 to 256 chars	String String	Read Only Read/Write
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A447.001D A448.128D-A959.001D .l is string length, range 1 to 128	String String	Read Only Read/Write

	chars		
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A447.001E A448.128E-A959.001E .l is string length, range 1 to 128 chars	String String	Read Only Read/Write
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO6143.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO6143.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D-CIO6143.001D .l is string length, range 1 to 128 chars	String	Read/Write
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO6143.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C4095	BCD , Word, Short	Read/Write
Counter Status	CS0000-CS4095	Boolean	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR00-DR15 DR00-DR14	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32767.002L .l is string length, range 2 to 256 chars	String	Read/Write

Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128D-E32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E12:32767 E00:00000-E12:32766 Ex:x.00-Exx:xxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H-E12:32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.256L-E12:32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D-E12:32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E-E12:32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Index Register	IR00-IR15	DWord , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK31	Boolean	Read Only
Timer	T0000-T4095	BCD , Word, Short	Read/Write
Timer Status	TS0000-TS4095	Boolean	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write

*Use caution when modifying 32-bit values such as DWord, Long, LBCD, and Float. Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1: writing to D0 will also modify the value held in D1. It is recommended that these data types be used in order to prevent overlapping. For example, to prevent overlapping Words, users should use D0, D2, D4, and so on for DWords. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CJ1-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CJ1 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

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.

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Writing to DR and IR Registers

DR and IR registers can be written to only when the device is in Programming Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. Because the write will succeed, no error message be returned. The device's value will not change because the device is in Run Mode.

CJ2 Addressing

The default data types for dynamically defined tags are shown in **bold** where appropriate. For more information on notes and restrictions, refer to [BCD Support](#), [String Support](#), and [Array Support](#).

Device Type	Range	Data Type	Access
Auxiliary Relay	A000-A447 A000-A446	Word , Short, BCD Long, DWord, LBCD, Float	Read Only
	A448-A1471 A448-A1470	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write
	A10000-A11535 A10000-A11534	Word , Short, BCD Long, DWord, LBCD, Float	Read Only
	A000.00-A000.15-A447.00- A447.15	Word , Short, BCD Long, DWord, LBCD, Float	Read Only Read/Write
	A448.00-A448.15-A1471.00- A1471.15	Boolean Boolean	Read Only
	A10000.00-A10000.15- A11535.00-A11535.15	Boolean	
Auxiliary Relay as string with HiLo byte order	A000.256H-A447.002H A448.256H-A1471.002H A10000.256H-A11535.002H	String String String	Read Only Read/Write Read Only
	.l is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte order	A000.256L-A447.002L A448.256L-A1471.002L A10000.256L-A11535.002L	String String String	Read Only Read/Write Read Only
	.l is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the High Order byte of each word	A000.128D-A447.001D A448.128D-A1471.001D A10000.128D-A11535.001D	String String String	Read Only Read/Write Read Only
	.l is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the Low Order byte of each word	A000.128E-A447.001E A448.128E-A1471.001E A10000.128E-A11535.001E	String String String	Read Only Read/Write Read Only
	.l is string length, range 1 to 128 chars		
CIO	CIO0000-CIO6143 CIO0000-CIO6142 CIOxxxx.00-CIOxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H-CIO6143.002H .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string with LoHi byte order	CIO0000.256L-CIO6143.002L .l is string length, range 2 to 256 chars	String	Read/Write
CIO Memory as string	CIO0000.128D-CIO6143.001D	String	Read/Write

Using Only the High Order byte of each word	.l is string length, range 1 to 128 chars		
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E-CIO6143.001E .l is string length, range 1 to 128 chars	String	Read/Write
Counter	C0000-C4095	BCD , Word, Short	Read/Write
Counter Status	CS0000-CS4095	Boolean	Read/Write*
Data Memory	D00000-D32767 D00000-D32766 Dxxxxx.00-Dxxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Data Memory as string with HiLo byte order	D00000.256H-D32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L-D32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the High Order byte of each word	D00000.128D-D32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Data Memory as string Using Only the Low Order byte of each word	D00000.128E-D32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Data Register	DR00-DR15 DR00-DR14	Word , Short, BCD Long, DWord, LBCD, Float	Read/Write*
Expansion Data Memory (current bank)	E00000-E32767 E00000-E32766 Exxxxx.00-Exxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H-E32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L-E32767.002L .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the High Order byte of each word	E00000.128D-E32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E-E32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory	E00:00000-E24:32767 E00:00000-E24:32766 Ex:x.00-Exx:xxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Expansion Data Memory as string with HiLo byte order	E00:00000.256H- E24:32767.002H .l is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as	E00:00000.256L-	String	Read/Write

string with LoHi byte order	E24:32767.002L .l is string length, range 2 to 256 chars		
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D- E24:32767.001D .l is string length, range 1 to 128 chars	String	Read/Write
Expansion Data Memory as string Using Only the Low Order byte of each word	E00:00000.128E- E24:32767.001E .l is string length, range 1 to 128 chars	String	Read/Write
Holding Relay	H0000-H1535 H0000-H1534 Hxxxx.00-Hxxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write
Index Register	IR00-IR15	DWord , Long, LBCD, Float	Read/Write*
Task Flag	TK00-TK127	Boolean	Read Only
Timer	T0000-T4095	BCD , Word, Short	Read/Write
Timer Status	TS0000-TS4095	Boolean	Read/Write*
Working Relay	W000-W511 W000-W510 Wxxx.00-Wxxx.15	Word , Short, BCD Long, DWord, LBCD, Float Boolean	Read/Write

*Use caution when modifying 32-bit values such as DWord, Long, LBCD, and Float. Each address for which these data types are allowed starts at a word offset within the device. Therefore, DWords D0 and D1 overlap at word D1: writing to D0 will also modify the value held in D1. It is recommended that these data types be used in order to prevent overlapping. For example, to prevent overlapping Words, users should use D0, D2, D4, and so on for DWords. The exception to this is IR tags, which are native 32-bit values with MSB to LSB byte ordering for the CJ1-series PLCs.

BCD Support

16-Bit and 32-Bit data can be referenced as signed and unsigned BCD values.

Unsigned BCD support is accomplished by appending a 'D' to any Word or DWord tag address or by using the BCD and LBCD data types. The value range follows:

Word as unsigned BCD
0 - 9999

DWord as unsigned BCD
0 - 99999999

Signed BCD support is accomplished by appending a 'D' to any Short or Long tag address. The most significant bit of the Short/Long is set to 1 for negative values, 0 for positive values. The value range follows:

Short as signed BCD
+/-7999

Long as signed BCD
+/-79999999

Example

D0D @ Short = -50
D0D @ Word = 8050 (MSB set)
D0 @ BCD = 8050

D0D @ Short = 50
D0D @ Word = 50
D0 @ BCD = 50

Note: Boolean, Float and String data types do not support appending 'D' to the tag address. Furthermore, 'D' cannot be appended to BCD or LBCD tags, or tags with a default data type of BCD or LBCD (such as Counter and Timer tags).

String Support

The CJ1 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register will contain two bytes (two characters) of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 256 characters and is entered in place of a bit number. The string length cannot exceed the request size that has been assigned to the device. The length must be entered as an even number. The range also cannot exceed the request size that has been assigned to the device. Since this driver uses an ASCII protocol, there are 2 bytes transmitted for each character. 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 128 characters and is entered in place of a bit number. The string length times 4 cannot exceed the request size that has been assigned to the device. 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. For more information, refer to [Device Setup](#).

Examples

1. To address a string starting at D01000 with a length of 100 bytes and HiLo byte order, enter:
D01000.100H
2. To address a string starting at D01100 with a length of 78 bytes and LoHi byte order, enter:
D01100.078L
3. To address a string starting at D02000 with a length of 55 bytes and Only the High Order byte, enter:
D02000.055D
4. To address a string starting at D02200 with a length of 37 bytes and Only the Low Order byte, enter:
D02200.037E

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.

Dxxxx [rows] [cols]
Dxxxx [cols] (this method assumes "rows" is equal to one)

Rows multiplied by cols multiplied by data size in bytes cannot exceed the request size that has been assigned to the device. 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 10 X 10 array of words results in an array size of 400 bytes, which would require a request size of 512. The next smaller request size choice, 256, would be too small. For more information, refer to [Device Setup](#).

Writing to DR and IR Registers

DR and IR registers can be written to only when the device is in Programming Mode. If the device is in Run Mode and a write is attempted to one of these registers, the value in the device will not change. Because the write will succeed, no error message be returned. The device's value will not change because the device is in Run Mode.

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](#)

[Unable to write to '<address>' on device '<device name>'](#)

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](#)

Omron FINS Serial Device Specific Messages

[Device '<device name>' access right denied \(Tag '<address>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' cannot accept command \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' cannot process command \(Tag '<address>', Size '<bytes>'\) \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' deactivated due to network settings conflict with device '<device name>'](#)

[Device '<device name>' is not responding](#)

[Device '<device name>' responded with a command format error \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with a command parameter error \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with a communications error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with destination node error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with error '<error code>' \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with error in unit. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with Fatal CPU Unit error. \[End Code: '<end code>'\]](#)

[Device '<device name>' responded with local node error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with Non-fatal CPU Unit error. \[End Code: '<end code>'\]](#)

[Device '<device name>' responded with read not possible \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with routing table error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with write not possible \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[The current project was created with an older version of this driver](#)

See Also: [Main and Sub Error Codes](#)

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](#)

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

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 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:

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

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.

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>' \[<error mask>\]](#)

[COMn does not exist](#)

[COMn is in use by another application](#)

[Error opening COMn](#)

[Unable to set comm parameters on COMn](#)

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 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.

Omron FINS Serial Device Specific Messages

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

Omron FINS Serial Device Specific Messages

[Device '<device name>' access right denied \(Tag '<address>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' cannot accept command \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' cannot process command \(Tag '<address>', Size '<bytes>'\) \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' deactivated due to network settings conflict with device '<device name>'](#)

[Device '<device name>' is not responding](#)

[Device '<device name>' responded with a command format error \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with a command parameter error \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with a communications error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with destination node error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with error '<error code>' \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with error in unit. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with Fatal CPU Unit error. \[End Code: '<end code>'\]](#)

[Device '<device name>' responded with local node error. \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with Non-fatal CPU Unit error. \[End Code: '<end code>'\]](#)

[Device '<device name>' responded with read not possible \(Tag '<address>', Size '<bytes>'\). \[Main, Sub: '<main code, sub code>'\]](#)

[Device '<device name>' responded with routing table error. \[Main, Sub: '<main code, sub code>'\]](#)

Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

The current project was created with an older version of this driver

Device '<device name>' access right denied (Tag '<address>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' cannot accept command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' cannot process command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' deactivated due to network settings conflict with device '<device name>'.

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

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.

Device '<device name>' responded with a command format error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with a command parameter error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with a communications error. [Main, Sub: '<main code, sub code>']

Error Type:

Serious

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with destination node error. [Main, Sub: '<main code, sub code>']

Error Type:

Serious

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with error '<error code>' (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with error in unit. [Main, Sub: '<main code, sub code>']

Error Type:

Serious

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with Fatal CPU Unit error. [End Code: '<end code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with local node error. [Main, Sub: '<main code, sub code>']

Error Type:

Serious

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with Non-fatal CPU Unit error. [End Code: '<end code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with read not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with routing table error. [Main, Sub: '<main code, sub code>']

Error Type:

Serious

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']

Error Type:

Warning

Possible Cause:

Refer to [Main and Sub Error Codes](#) for a description of the Main and Sub Codes. The description includes possible causes.

Solution:

Refer to [Main and Sub Error Codes](#) topic for a description of the Main and Sub Codes. The description includes corrections.

The current project was created with an older version of this driver. Examine device properties to validate network parameters

Error Type:

Warning

Possible Cause:

The current project was created using a version of this driver that checked for network parameter conflicts during Runtime. This version only performs Runtime checks on parameters that can be changed with System Tags. All other parameters are now checked when the Device Properties are edited.

Solution:

Examine the properties of each device. Change the value of any property and then change it back to its original value in order to alter the property sheet. Then, click "Accept" or "OK" to validate all of the network parameters associated with the device. Once the devices are validated, save the project.

Note:

This solution will not have to be repeated. Any new devices that are added to the project will be validated automatically.

See Also:

[FINS Networks](#)

Main and Sub Error Codes**Main Codes 00-05**

Main Code	Sub-Code	Check Point	Probable Cause	Correction
00: Normal completion	00: Normal completion	N/A.	N/A.	N/A.
	01: Service canceled	Data link status.	1. Service was canceled. 2. Service was canceled.	1. Check the capacity of the destination area in the third node. 2. Check the status of the data link.
01: Local node error	01: Local node not in network	Network status of local node.	Local node is not participating in the network.	Connect the node to the network.
	02: Token timeout	Maximum node address.	Token does not arrive.	Set the local node to within the maximum node address.
	03: Retries failed	N/A.	Send was not possible during the specified number of retries.	Execute a communications test between the nodes and re-examine the system environment if it fails.
	04: Too many send frames	Number of enabled send frames.	Cannot send because maximum number of event frames exceeded.	Check event execution on the network and reduce the number of events per cycle. Increase the maximum number of event frames.
	05: Node address range error	Node address.	Node address setting error occurred.	Check the settings of the rotary switches to be sure that the address is within range and that each address is set only once in the same network.
02: Destination node error	01: Destination node not in network	INS indicator on Unit.	The destination node is not in the network.	Add the destination node to the network.
	02: Unit missing	Instruction control data.	There is no unit with the specified unit address.	Check the destination unit address.
	03: Third node missing	1. Instruction control data. 2. Command data.	1. The third node does not exist. 2. Broadcasting was specified.	1. Check the unit address of the third node. Check the node address of the third node in the

	04: Destination node busy 05: Response time-out	N/A. 1. N/A. 2. Instruction control data. 3. Error history.	The destination node is busy. 1. The message was destroyed by noise. 2. The response monitor time is too short. 3. The send/receive frame was discarded.	send data for CMND(490). 2. Specify only one node for the third node. Increase the number of retries or review the system so that the destination node does not receive so many messages. 1. Increase the number of retries or test communications between nodes to see if there is too much noise. 2. Increase the length of the response monitor time. 3. Take appropriate measures based on the error history.
03: Controller error	01: Communications controller error 02: CPU Unit error 03: Controller error 04: Unit number error	Unit / Board indicators. CPU Unit indicators at destination node. Board indicators. Unit number.	An error occurred in the communications controller. A CPU error occurred in the destination CPU Unit. A response was not returned because an error occurred in the Board. The unit number was set incorrectly.	Take appropriate measures based on the operation manuals for the related Units/Boards. Clear the error from the CPU Unit based on its operation manuals. Check network communications status and restart the Board. If the problem persists, replace the Board. Set the rotary switches correctly, being sure the unit numbers are within range and that each number is used only once.
04: Service unsupported	01: Undefined command 02: Not supported by model/version	Command code. Unit model and version.	The Unit/Board does not support the specified command code. The command cannot be executed because the model or version is incorrect.	Check the command code. Check the model number and version.
05: Routing table error	01: Destination address setting error 02: No routing tables 03: Routing table error 04: Too many relays	Routing table. Routing table. Routing table. Network configuration.	The destination network or node address is not set in the routing tables. Relaying is not possible because there are no routing tables. There is an error in the routing tables. An attempt was made to send to a network that was over 3 networks away	Register the destination network and node in the routing tables. Set routing tables in the source node, designation node, and relay nodes. Set the routing tables correctly. Reconstruct the networks or change the routing tables so that commands are sent within a range of 3 networks or less.

Main Codes 10-20

Main Code	Sub-Code	Check Point	Probable Cause	Correction
10: Command format error	01: Command too long	Command data.	The command is longer than the maximum permissible length.	Check the command format and correct the command data.

	<p>02: Command too short</p> <p>03: Elements / data don't match</p> <p>04: Command format error</p> <p>05: Header error</p>	<p>Command data.</p> <p>Command data.</p> <p>Command data.</p> <p>Routing table.</p>	<p>The command is shorter than the minimum permissible length.</p> <p>The designated number of elements differs from the number of write data items.</p> <p>An incorrect format was used.</p> <p>Either the relay table in the local node or the local network table in the relay node is incorrect.</p>	<p>Check the command format and correct the command data.</p> <p>Check the number of elements and set data for each element.</p> <p>Check the command format and correct the command data.</p> <p>Set the routing tables correctly.</p>
11: Parameter error	<p>01: Area classification missing</p> <p>02: Access size error</p> <p>03: Address range error</p> <p>04: Address range exceeded</p> <p>06: Program missing</p> <p>09: Relational error</p> <p>0A: Duplicate data access</p> <p>0B: Response too long</p> <p>0C: Parameter error</p>	<p>Memory area code in command data.</p> <p>Access size specification in command data.</p> <p>Starting address in command data.</p> <p>1. Starting address and number of elements in command data. 2. Data link tables.</p> <p>Program number in command data.</p> <p>1. Command data. 2. Data link table.</p> <p>1. I/O access in CPU Unit. 2. Data link tables.</p> <p>Number of elements in command data.</p> <p>1. Parameters in command data. 2. Data link table file.</p>	<p>The specified word does not exist in the memory area or there is no EM Area.</p> <p>The access size specification is incorrect or an odd word address is specified.</p> <p>The start address in command process is beyond the accessible area.</p> <p>1. The end address in command process is beyond the accessible area. 2. The total number of words is beyond the limit.</p> <p>FFFF Hex was not specified.</p> <p>1. A node not set in the common link parameters is set as a refresh parameter. 2. A node not set in the common link parameters is set as a refresh parameter.</p> <p>1. Differential monitoring was specified during data tracing or data tracing was specified during differential monitoring. 2. The same node address is specified more than once.</p> <p>The response format is longer than the maximum permissible length.</p> <p>1. There is an error in one of the parameter settings. 2. There is an error in the file.</p>	<p>Check the memory areas and parameter codes in the command and correct the command data.</p> <p>Check the memory areas and access size and correct the access size.</p> <p>Check the area being processed and set the correct range.</p> <p>1. Check the area being processed and set the correct range. 2. Correct the data link tables.</p> <p>Specify FFFF Hex.</p> <p>1. Check the command data and correct the relationship between the elements. 2. Correct the data link tables.</p> <p>1. Abort the current process or wait until it ends before executing the command. 2. Correct the data link tables.</p> <p>Check the command format and correct the number of elements.</p> <p>1. Check the command data and correct the parameters. 2. Check the contents of the file.</p>

20: Read not possible	02: Protected	N/A.	The program area is protected.	Release protection from a Programming Device and then execute the command.
	03: Table missing	Table.	1. A table has not been registered. 2. There is an error in the table.	1. Register a table. 2. Correct the table.
	04: Data missing	N/A.	The search data does not exist.	N/A.
	05: Program missing	Program number in command data.	A non-existing program number has been specified.	Check the program numbers and specify a valid one.
	06: File missing	File name and file device.	The file does not exist at the specified file device.	Check the path and file name, and correct them.
	07: Data mismatch	Contents of memory being compared.	1. A data being compared is not the same. 2. A file read operation failed.	1. Check memory contents and use the correct data. 2. Check the contents of the file.

Main Codes 21-25

Main Code	Sub-Code	Check Point	Probable Cause	Correction
21: Write not possible	01: Read Only	N/A.	The specified area is Read Only.	If the area is protected using a switch setting, release protection and then execute the command. If the area is permanently Read Only, the command cannot be executed.
	02: Protected Cannot write data link table	1. N/A. 2. PLC Setup.	1. The program area is protected. 2. Writing is not possible because automatic data link table generation has been specified.	1. Release protection from a Programming Device and then execute the command. 2. Change the PLC Setup so that the data link tables can be manually written.
	03: Cannot register	1. Number of files in file device. 2. Number of files open.	1. The file cannot be created because the limit has been exceeded. 2. The maximum number of files has already been opened for the system limit.	1. Delete any unnecessary files or create more file memory. 2. Close one or more files and then execute the command.
	05: Program missing	Program number in command data.	A non-existing program number has been specified.	Check the program numbers and specify a valid one.
	06: File missing	File name.	The file does not exist at the specified file device.	Correct the file name and then execute the command.
	07: File name already exists	File name.	A file with the same name already exists in the specified file device.	Change the name of the file being written and then execute the command.
	08: Cannot change	Contents of memory being changed.	The change cannot be made because doing so would create a problem.	N/A.
	22: Not executable in current mode	01: Not possible during execution	1. N/A. 2. Data link	1. The mode is incorrect. 2. The data link is operating

	<p>02: Not possible while running</p> <p>03: Wrong PLC mode</p> <p>04: Wrong PLC mode</p> <p>05: Wrong PLC mode</p> <p>06: Wrong PLC mode</p> <p>07: Specified node not polling node</p> <p>08: Step cannot be executed</p>	<p>status.</p> <p>1. N/A. 2. Data link status.</p> <p>N/A.</p> <p>N/A.</p> <p>N/A.</p> <p>N/A.</p> <p>N/A.</p> <p>N/A.</p>	<p>1. The mode is incorrect. 2. The data links are active.</p> <p>The PLC is in PROGRAM mode.</p> <p>The PLC is in DEBUG mode.</p> <p>The PLC is in MONITOR mode.</p> <p>The PLC is in RUN mode.</p> <p>The specified node is not the polling node.</p> <p>The mode is incorrect.</p>	<p>links.</p> <p>1. Check the mode. 2. Check the status of the data links.</p> <p>Check the modes of the PLC and computer.</p> <p>Check the modes of the PLC and computer.</p> <p>Check the modes of the PLC and computer.</p> <p>Check the modes of the PLC and computer.</p> <p>Check node functioning as the polling node for the network.</p> <p>Check step status.</p>
23: No such device	<p>01: File device missing</p> <p>02: Memory missing</p> <p>03: Clock missing</p>	<p>Unit configuration.</p> <p>N/A.</p> <p>N/A.</p>	<p>The specified memory does not exist as a file device.</p> <p>There is no file memory.</p> <p>There is no clock.</p>	<p>Mount memory or format EM as file memory.</p> <p>Check the file memory to see if it is mounted.</p> <p>Check the model.</p>
24: Cannot start / stop	01: Table missing	Data link tables.	The data link tables have not been registered or they contain an error.	Set the data link tables.
25: Unit error	<p>02: Memory error</p> <p>03: I/O setting error</p> <p>04: Too many I/O points</p> <p>05: CPU bus error</p> <p>06: I/O duplication</p> <p>07: I/O bus error</p> <p>09: SYSMAC BUS /</p>	<p>Contents of memory being processed.</p> <p>I/O Unit configuration.</p> <p>Number of I/O in registered I/O tables.</p> <p>CPU bus line.</p> <p>Rack numbers, Unit numbers, and I/O addresses in PLC Setup.</p> <p>I/O bus line.</p> <p>SYSMAC BUS / 2</p>	<p>The contents of memory contains an error.</p> <p>The registered I/O tables do not agree with the actual I/O configuration.</p> <p>There are too many I/O points and remote I/O points registered.</p> <p>An error occurred in data transfer between the CPU and a CPU Bus Unit.</p> <p>The same number / address was set more than once.</p> <p>An error occurred in data transfer between the CPU and an I/O Unit.</p> <p>An error occurred in data</p>	<p>Transfer the correct contents to memory.</p> <p>Correct the I/O tables or the I/O configuration.</p> <p>Change the registered I/O table so that it is within the limit.</p> <p>Check Units, Boards, and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.</p> <p>Check the PLC Setup and correct the numbers / addresses so that each is used only once.</p> <p>Check Units, Boards, and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.</p> <p>Check Units, Boards, and cables</p>

	2 error	transmission path.	transfer on the SYSMAC BUS / 2 line.	to be sure they are connected correctly and then execute the ERROR CLEAR command.
	0A: CPU Bus Unit error	CPU Bus Unit transmission path.	An error occurred in data transfer for a CPU Bus Unit.	Check Units, Boards, and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	0D: SYSMAC BUS No. duplication	Word settings.	The same word is allocated more than once.	Check the I/O tables and correct the allocations.
	0F: Memory error	Status of memory being processed.	A memory error has occurred in internal memory, a memory card, or EM file memory.	For internal memory, write the correct data and then execute the command. For a memory card or EM file memory, the file data has been destroyed. Execute the FILE MEMORY FORMAT command. If the problem persists, replace the memory.
	10: SYSMAC BUS terminator missing	N/A.	Terminators have not been set.	Set the terminators correctly.

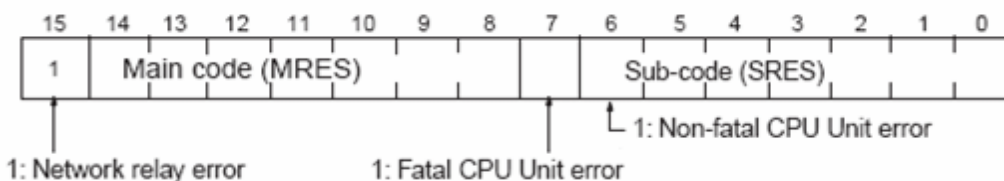
Main Code 26-40

Main Code	Sub-Code	Check Point	Probable Cause	Correction
26: Command error	01: No protection	Command protection for program area.	The specified area is not protected.	An attempt was made to clear protection on an area that is not protected, i.e., there is no reason to clear protection.
	02: Incorrect password	N/A.	An incorrect password has been specified.	Specify the correct password.
	04: Protected	1. N/A. 2. Number of commands being executed.	1. The specified area is protected. 2. The node receiving the command is already processing 5 commands.	1. Clear protection from a Programming Device and then execute the command. 2. Wait for current processing to end or force the end of a current process and then execute the command.
	05: Service already executing	N/A.	The service is being executed.	Wait for the service to end or force the end of the service and then execute the command.
	06: Service stopped	N/A.	The service is not being executed.	If necessary, start the service.
	07: No execution right	1. LNK indicator on Unit / Board. 2. N/A.	1. The right to execute the service has not been obtained. 2. A response was not returned because a buffer error occurred.	1. The local node is not in the data link. Execute the command from a node that is participating in the data link. 2. Restart the Board. If the problem persists, replace the Board.
	08: Settings not complete	Settings required before execution.	The settings required before executing the service have not been made.	Make the required settings.
	09: Necessary items not set	Command data.	The required elements have not been set in the command data.	Check the command format and set the required elements in the command data.

	0A: Number already defined 0B: Error will not clear	Action numbers and transition numbers of program in program area. Cause of error being cleared.	The specified action/transition number has already been registered in a previous program. The cause of the error has not been removed.	Check the action/transition numbers to ones that are not being used and then execute the command. Remove the cause of the error and then execute ERROR CLEAR.
30: Access right error	01: No access right	N/A.	The access right is held by another device. (Online editing is being executed from another node or ACCESS RIGHT ACQUIRE or ACCESS RIGHT FORCE ACQUIRE has been executed by another node.)	Wait until the access right is released and then execute the command. ACCESS RIGHT ACQUIRE or ACCESS RIGHT FORCE ACQUIRE can be executed to obtain the access right, but this may adversely affect processing by the node that previously held the access right.
40: Abort	01: Service aborted	N/A.	Service was aborted with ABORT command.	

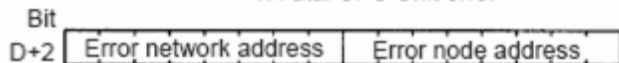
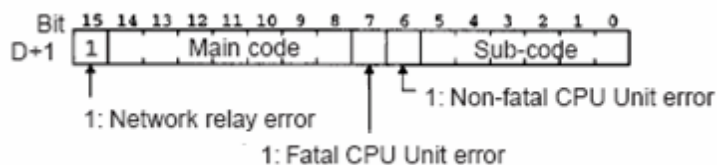
Flags

There are specific flags in the end code word (bits 6, 7, and 15) that may also be ON. The following diagram displays the contents of the end code word.



If bit 6 or 7 is ON, an error has occurred in the destination CPU Unit. If this occurs, refer to the operation manuals for the CPU Unit where the error occurred and then remove the cause of the error.

If bit 15 is ON, an error has occurred during a network relay operation. The end code contains an additional two-byte (one word) network relay error code that can be used to determine the location of the relay error. The following diagram displays the response data of a network relay error. This information can be used to determine the node where the error occurred so that appropriate measures may be taken.



Error network address: 00 to 7F (0 to 127)
Error Node Address

- Ethernet: 01 to 7E (1 to 128)
- SYSMAC NET: 01 to 7E (1 to 126)
- Controller Link: 01 to 20 (1 to 32)
- SYSMAC LINK: 01 to 3E (1 to 62)

Index

A

Address '<address>' is out of range for the specified device or register.....	62
Address Descriptions.....	12
Address Validation.....	61

B

BCD.....	11
Boolean.....	11

C

C1000H Addressing.....	19
C2000H Addressing.....	22
C200H Addressing.....	12
C500 Addressing.....	16
CJ1 Addressing.....	53
CJ2 Addressing.....	57
Communication Parameters.....	6
Communications error on '<channel name>' [<error mask>].....	63
COMn does not exist.....	63
COMn is in use by another application.....	64
CS1 Addressing.....	48
CV1000 Addressing.....	29
CV2000 Addressing.....	33
CV500 Addressing.....	25
CVM1-CPU11 Addressing.....	41
CVM1-CPU21 Addressing.....	44

D

Data Type '<type>' is not valid for device address '<address>'.....	62
Data Types Description.....	11

Device '<device name>' responded with a communications error. [Main, Sub: '<main code, sub code>']	66
Device '<device name>' access right denied (Tag '<address>'). [Main, Sub: '<main code, sub code>']	65
Device '<device name>' cannot accept command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	65
Device '<device name>' cannot process command (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	65
Device '<device name>' deactivated due to network settings conflict with device '<device name>'	65
Device '<device name>' is not responding	65
Device '<device name>' responded with a command format error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	66
Device '<device name>' responded with a command parameter error (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	66
Device '<device name>' responded with destination node error. [Main, Sub: '<main code, sub code>']	66
Device '<device name>' responded with error '<error code>' (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	67
Device '<device name>' responded with error in unit. [Main, Sub: '<main code, sub code>']	67
Device '<device name>' responded with Fatal CPU Unit error. [End Code: '<end code>']	67
Device '<device name>' responded with local node error. [Main, Sub: '<main code, sub code>']	67
Device '<device name>' responded with Non-fatal CPU Unit error. [End Code: '<end code>']	68
Device '<device name>' responded with read not possible (Tag '<address>', Size '<bytes>').	68
Device '<device name>' responded with routing table error. [Main, Sub: '<main code, sub code>']	68
Device '<device name>' responded with write not possible (Tag '<address>', Size '<bytes>'). [Main, Sub: '<main code, sub code>']	68
Device address '<address>' contains a syntax error	62
Device address '<address>' is not supported by model '<model name>'	62
Device address '<address>' is Read Only	62
Device ID	5
Device Setup	5

E

Error Descriptions	61
Error opening COMn	64

F

FINS Networks	7
Framing	63

M

Main and Sub Error Codes	69
Mask	63
Missing address	62
Modem Setup	7

N

Network	5
----------------------	----------

O

Omron FINS Serial Device Specific Messages	64
Overrun	63
Overview	4

P

Parity	63
---------------------	-----------

R

Run Mode Settings	6
--------------------------------	----------

S

Serial Communications	63
Short	11
Supported Devices	5

T

The current project was created with an older version of this driver. Examine device prop-... erties to validate network parameters	68
--	-----------

U

Unable to set comm parameters on COMn	64
Unable to write tag '<address>' on device '<device name>'	63

V

VM1-CPU01 Addressing	37
-----------------------------------	-----------

W

Word	11
-------------------	-----------