# **Omron FINS Serial Driver**

© 2025 PTC Inc. All Rights Reserved.

# **Table of Contents**

Omron FINS Serial Driver	-
Table of Contents	
Welcome to the Omron FINS Serial Driver Help Center	
Overview	5
Setup	5
Channel Properties – General	6
Tag Counts	6
Channel Properties – Serial Communications	
Channel Properties – Write Optimizations	
Channel Properties – Advanced	
Device Properties – General	
Operating Mode	
Tag Counts	
Device Properties – Scan Mode	
Device Properties – Timing  Device Properties – Auto-Demotion	
Device Properties – Run Mode Settings	
Device Properties – Communication Parameters	
Device Properties – Redundancy	
FINS Networks	
Data Types Description	
Address Descriptions	
C200H Addressing	
C500 Addressing	
C1000H Addressing	
•	
C2000H Addressing	
CV500 Addressing	
CV1000 Addressing	
CV2000 Addressing	
CVM1-CPU01 Addressing	
CVM1-CPU11 Addressing	49
CVM1-CPU21 Addressing	53
CS1 Addressing	58
CJ1 Addressing	63
CJ2 Addressing	67
Event Log Messages	72
Device responded with error.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	72
Device responded with local node error.   Main error = <code>, Sub error = <code>.</code></code>	72
Device responded with destination node error.   Main error = <code>, Sub error = <code></code></code>	72
Device responded with a communications error.   Main error = <code>, Sub error = <code>.</code></code>	73

	= <code>, Sub error = <code>.</code></code>	
	Device responded with routing table error.   Main error = <code>, Sub error = <code>.</code></code>	. 73
	Device responded with a command format error.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	73
	Device responded with a command parameter error.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	74
	Device responded with read not possible.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	74
	Device responded with error in unit.   Main error = <code>, Sub error = <code>.</code></code>	74
	Device cannot accept command.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	
	Access right denied.   Tag address = ' <address>', Main error = <code>, Sub error = <code>.</code></code></address>	. 75
	Device responded with write not possible.   Tag address = ' <address>', Data size = <byte number="">, Main error = <code>, Sub error = <code>.</code></code></byte></address>	75
	Unable to write to tag. The device is in Run Mode.   Tag address = ' <address>'.</address>	75
	Device responded with fatal CPU unit error.   End code = <code>.</code>	75
	Device responded with non-fatal CPU unit error.   End code = <code>.</code>	75
	Failed to set device back to run mode following write to tag.   Tag address = ' <address>'</address>	76
	Error Mask Definitions	76
ı	Main and Sub Error Codes	76
Inc	dex	84

# Welcome to the Omron FINS Serial Driver Help Center

This help center is the user documentation for Kepware Omron FINS Serial Driver. This help center is updated regularly to reflect the latest functionality and information.

#### Overview

What is the Omron FINS Serial Driver?

#### 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?

## **Event Log Messages**

What messages does the Omron FINS Serial Driver produce?

Version 1.062

© 2025 PTC Inc. All Rights Reserved.

## Overview

The Omron FINS Serial Driver provides a reliable way to connect Omron FINS serial devices to OPC client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications.

## Setup

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 website.

#### 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

## **Channel and Device Limits**

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

## **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 set through **channel properties**.

#### Flow Control

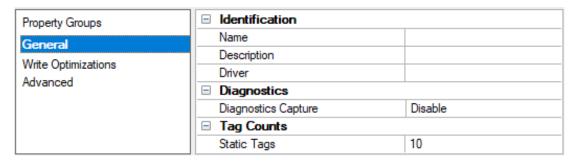
When using an RS232 / RS485 converter, the type of flow control that is required depends on the needs of the converter. Some converters do not require any flow control whereas others require RTS flow. To determine the converter's flow requirements, refer to its documentation. An RS485 converter that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

#### Notes:

- 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.
- 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.

## **Channel Properties – General**

This server supports the use of multiple simultaneous communications drivers. Each protocol or driver used in a server project is called a channel. A server project may consist of many channels with the same communications driver or with unique communications drivers. A channel acts as the basic building block of an OPC link. This group is used to specify general channel properties, such as the identification attributes and operating mode.



#### Identification

**Name**: Specify the user-defined identity of this channel. In each server project, each channel name must be unique. Although names can be up to 256 characters, some client applications have a limited display window when browsing the OPC server's tag space. The channel name is part of the OPC browser information. The property is required for creating a channel.

For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

**Description**: Specify user-defined information about this channel.

Many of these properties, including Description, have an associated system tag.

**Driver**: Specify the protocol / driver for this channel. Specify the device driver that was selected during channel creation. It is a disabled setting in the channel properties. The property is required for creating a channel.

Note: With the server's online full-time operation, these properties can be changed at any time. This includes changing the channel name to prevent clients from registering data with the server. If a client has already acquired an item from the server before the channel name is changed, the items are unaffected. If, after the channel name has been changed, the client application releases the item and attempts to re-acquire using the old channel name, the item is not accepted. Changes to the properties should not be made once a large client application has been developed. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.

#### **Diagnostics**

**Diagnostics Capture**: When enabled, this option makes the channel's diagnostic information available to OPC applications. Because the server's diagnostic features require a minimal amount of overhead processing, it is recommended that they be utilized when needed and disabled when not. The default is disabled.

- Note: This property is not available if the driver or operating system does not support diagnostics.
- For more information, refer to Communication Diagnostics and Statistics Tags in server help.

## Tag Counts

**Static Tags**: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

## **Channel Properties – Serial Communications**

Serial communication properties are available to serial drivers and vary depending on the driver, connection type, and options selected. Below is a superset of the possible properties.

Click to jump to one of the sections: <u>Connection Type</u>, <u>Serial Port Settings</u>, and <u>Operational Behavior</u>.

Notes:

- With the server's online full-time operation, these properties can be changed at any time. Utilize proper user role and privilege management to prevent operators from changing properties or accessing server features.
- Users must define the specific communication parameters to be used. Depending on the driver, channels
  may or may not be able to share identical communication parameters. Only one shared serial connection
  can be configured for a Virtual Network (see Channel Properties Serial Communications).

Property Groups	☐ Connection Type	
General	Physical Medium	COM Port
Serial Communications	□ Serial Port Settings	
Write Optimizations	COM ID	39
Advanced	Baud Rate	19200
Advanced	Data Bits	8
	Parity	None
	Stop Bits	1
	Flow Control	RTS Always
	□ Operational Behavior	
	Report Communication Errors	Enable
	Close Idle Connection	Enable
	Idle Time to Close (s)	15

## Connection Type

**Physical Medium**: Choose the type of hardware device for data communications. Options include Modem, COM Port, and None. The default is COM Port.

- None: Select None to indicate there is no physical connection, which displays the <u>Operation with no Com-</u> munications section.
- 2. COM Port: Select Com Port to display and configure the Serial Port Settings section.
- 3. **Modem**: Select Modem if phone lines are used for communications, which are configured in the <u>Modem</u> Settings section.
- 4. **Shared**: Verify the connection is correctly identified as sharing the current configuration with another channel. This is a read-only property.

#### Serial Port Settings

**COM ID**: Specify the Communications ID to be used when communicating with devices assigned to the channel. The valid range is 1 to 9991 to 16. The default is 1.

Baud Rate: Specify the baud rate to be used to configure the selected communications port.

Data Bits: Specify the number of data bits per data word. Options include 5, 6, 7, or 8.

Parity: Specify the type of parity for the data. Options include Odd, Even, or None.

**Stop Bits**: Specify the number of stop bits per data word. Options include 1 or 2.

**Flow Control**: Select how the RTS and DTR control lines are utilized. Flow control is required to communicate with some serial devices. Options are:

- None: This option does not toggle or assert control lines.
- DTR: This option asserts the DTR line when the communications port is opened and remains on.
- RTS: This option specifies that the RTS line is high if bytes are available for transmission. After all buffered bytes have been sent, the RTS line is low. This is normally used with RS232/RS485 converter hardware.
- RTS, DTR: This option is a combination of DTR and RTS.

- RTS Always: This option asserts the RTS line when the communication port is opened and remains on.
- RTS Manual: This option asserts the RTS line based on the timing properties entered for RTS Line Control. It is only available when the driver supports manual RTS line control (or when the properties are shared and at least one of the channels belongs to a driver that provides this support). RTS Manual adds an RTS Line Control property with options as follows:
  - Raise: Specify the amount of time that the RTS line is raised prior to data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Drop**: Specify the amount of time that the RTS line remains high after data transmission. The valid range is 0 to 9999 milliseconds. The default is 10 milliseconds.
  - **Poll Delay**: Specify the amount of time that polling for communications is delayed. The valid range is 0 to 9999. The default is 10 milliseconds.
- **Tip**: When using two-wire RS-485, "echoes" may occur on the communication lines. Since this communication does not support echo suppression, it is recommended that echoes be disabled or a RS-485 converter be used.

## **Operational Behavior**

- Report Communication Errors: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- Close Idle Connection: Choose to close the connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- Idle Time to Close: Specify the amount of time that the server waits once all tags have been removed before closing the COM port. The default is 15 seconds.

## **Modem Settings**

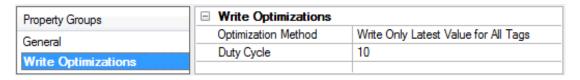
- Modem: Specify the installed modem to be used for communications.
- Connect Timeout: Specify the amount of time to wait for connections to be established before failing a read or write. The default is 60 seconds.
- Modem Properties: Configure the modem hardware. When clicked, it opens vendor-specific modem properties.
- **Auto-Dial**: Enables the automatic dialing of entries in the Phonebook. The default is Disable. *For more information, refer to "Modem Auto-Dial" in the server help.*
- Report Communication Errors: Enable or disable reporting of low-level communications errors. When enabled, low-level errors are posted to the Event Log as they occur. When disabled, these same errors are not posted even though normal request failures are. The default is Enable.
- Close Idle Connection: Choose to close the modem connection when there are no longer any tags being referenced by a client on the channel. The default is Enable.
- Idle Time to Close: Specify the amount of time that the server waits once all tags have been removed before closing the modem connection. The default is 15 seconds.

#### **Operation with no Communications**

• Read Processing: Select the action to be taken when an explicit device read is requested. Options include Ignore and Fail. Ignore does nothing; Fail provides the client with an update that indicates failure. The default setting is Ignore.

## Channel Properties – Write Optimizations

The server must ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties to meet specific needs or improve application responsiveness.



#### **Write Optimizations**

Optimization Method: Controls how write data is passed to the underlying communications driver. The options are:

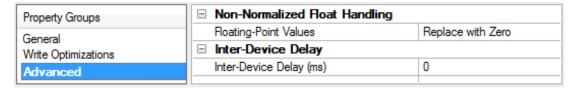
- Write All Values for All Tags: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing data to the device as quickly as possible. This mode ensures that everything written from the client applications is sent to the target device. This mode should be selected if the write operation order or the write item's content must uniquely be seen at the target device.
- Write Only Latest Value for Non-Boolean Tags: Many consecutive writes to the same value can accumulate in the write queue due to the time required to actually send the data to the device. If the server updates a write value that has already been placed in the write queue, far fewer writes are needed to reach the same final output value. In this way, no extra writes accumulate in the server's queue. When the user stops moving the slide switch, the value in the device is at the correct value at virtually the same time. As the mode states, any value that is not a Boolean value is updated in the server's internal write queue and sent to the device at the next possible opportunity. This can greatly improve the application performance.
   Note: This option does not attempt to optimize writes to Boolean values. It allows users to optimize the operation of HMI data without causing problems with Boolean operations, such as a momentary push button.
- Write Only Latest Value for All Tags: This option takes the theory behind the second optimization mode and applies it to all tags. It is especially useful if the application only needs to send the latest value to the device. This mode optimizes all writes by updating the tags currently in the write queue before they are sent. This is the default mode.

**Duty Cycle**: is used to control the ratio of write to read operations. The ratio is always based on one read for every one to ten writes. The duty cycle is set to ten by default, meaning that ten writes occur for each read operation. Although the application is performing a large number of continuous writes, it must be ensured that read data is still given time to process. A setting of one results in one read operation for every write operation. If there are no write operations to perform, reads are processed continuously. This allows optimization for applications with continuous writes versus a more balanced back and forth data flow.

• **Note**: It is recommended that the application be characterized for compatibility with the write optimization enhancements before being used in a production environment.

## Channel Properties – Advanced

This group is used to specify advanced channel properties. Not all drivers support all properties; so the Advanced group does not appear for those devices.



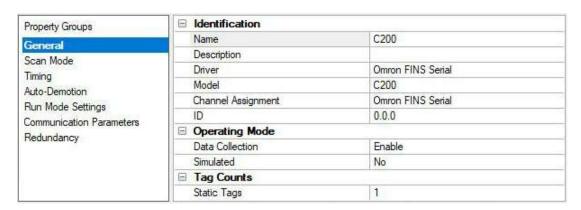
**Non-Normalized Float Handling**: A non-normalized value is defined as Infinity, Not-a-Number (NaN), or as a Denormalized Number. The default is Replace with Zero. Drivers that have native float handling may default to Unmodified. Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. Descriptions of the options are as follows:

- Replace with Zero: This option allows a driver to replace non-normalized IEEE-754 floating point values with zero before being transferred to clients.
- **Unmodified**: This option allows a driver to transfer IEEE-754 denormalized, normalized, non-number, and infinity values to clients without any conversion or changes.
- Note: This property is disabled if the driver does not support floating-point values or if it only supports the option that is displayed. According to the channel's float normalization setting, only real-time driver tags (such as values and arrays) are subject to float normalization. For example, EFM data is not affected by this setting.
- For more information on the floating-point values, refer to "How To ... Work with Non-Normalized Floating-Point Values" in the server help.

**Inter-Device Delay**: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

Note: This property is not available for all drivers, models, and dependent settings.

## Device Properties – General



#### Identification

Name: User-defined identity of this device.

**Description**: User-defined information about this device.

Channel Assignment: User-defined name of the channel to which this device currently belongs.

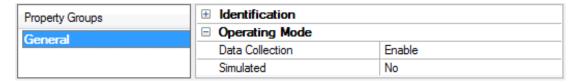
**Driver**: Selected protocol driver for this device.

**Model**: The specific version of the device. For a list of models that support the FINS Communications Service, refer to the manufacturer's website.

**ID**: The ID specifies the three-layer network address that uniquely identifies the target device. The format of the 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).
- For more information, refer to FINS Networks.
- See Also: Operating Mode.

#### **Operating Mode**



**Data Collection**: This property controls the device's active state. Although device communications are enabled by default, this property can be used to disable a physical device. Communications are not attempted when a device is disabled. From a client standpoint, the data is marked as invalid and write operations are not accepted. This property can be changed at any time through this property or the device system tags.

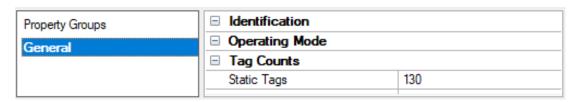
**Simulated**: Place the device into or out of Simulation Mode. In this mode, the driver does not attempt to communicate with the physical device, but the server continues to return valid OPC data. Simulated stops physical

communications with the device, but allows OPC data to be returned to the OPC client as valid data. While in Simulation Mode, the server treats all device data as reflective: whatever is written to the simulated device is read back and each OPC item is treated individually. The item's memory map is based on the group Update Rate. The data is not saved if the server removes the item (such as when the server is reinitialized). The default is No.

#### Notes:

- 1. Updates are not applied until clients disconnect and reconnect.
- 2. The System tag (\_Simulated) is read only and cannot be written to for runtime protection. The System tag allows this property to be monitored from the client.
- 3. In Simulation mode, the item's memory map is based on client update rate(s) (Group Update Rate for OPC clients or Scan Rate for native and DDE interfaces). This means that two clients that reference the same item with different update rates return different data.
- 4. When a device is simulated, updates may not appear faster than one (1) second in the client.
  - Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

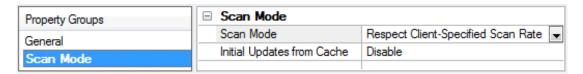
## **Tag Counts**



**Static Tags**: Provides the total number of defined static tags at this level (device or channel). This information can be helpful in troubleshooting and load balancing.

## Device Properties - Scan Mode

The Scan Mode specifies the subscribed-client requested scan rate for tags that require device communications. Synchronous and asynchronous device reads and writes are processed as soon as possible; unaffected by the Scan Mode properties.



**Scan Mode**: Specify how tags in the device are scanned for updates sent to subscribing clients. Descriptions of the options are:

- Respect Client-Specified Scan Rate: This mode uses the scan rate requested by the client.
- Request Data No Faster than Scan Rate: This mode specifies the value set as the maximum scan rate. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
  - Note: When the server has an active client and items for the device and the scan rate value is increased, the changes take effect immediately. When the scan rate value is decreased, the changes do not take effect until all client applications have been disconnected.
- Request All Data at Scan Rate: This mode forces tags to be scanned at the specified rate for subscribed clients. The valid range is 10 to 99999990 milliseconds. The default is 1000 milliseconds.
- Do Not Scan, Demand Poll Only: This mode does not periodically poll tags that belong to the device nor perform a read to get an item's initial value once it becomes active. It is the OPC client's responsibility to poll for updates, either by writing to the \_DemandPoll tag or by issuing explicit device reads for individual items. For more information, refer to "Device Demand Poll" in server help.
- Respect Tag-Specified Scan Rate: This mode forces static tags to be scanned at the rate specified in their static configuration tag properties. Dynamic tags are scanned at the client-specified scan rate.

**Initial Updates from Cache**: When enabled, this option allows the server to provide the first updates for newly activated tag references from stored (cached) data. Cache updates can only be provided when the new item reference shares the same address, scan rate, data type, client access, and scaling properties. A device read is used for the initial update for the first client reference only. The default is disabled; any time a client activates a tag reference the server attempts to read the initial value from the device.

## **Device Properties – Timing**

The device Timing properties allow the driver's response to error conditions to be tailored to fit the application's needs. In many cases, the environment requires changes to these properties for optimum performance. Factors such as electrically generated noise, modem delays, and poor physical connections can influence how many errors or timeouts a communications driver encounters. Timing properties are specific to each configured device.

Property Groups	☐ Communication Timeouts		
General	Connect Timeout (s)	3	
Scan Mode	Request Timeout (ms)	1000	
Timing	Attempts Before Timeout	3	
Tilling			

#### **Communications Timeouts**

**Connect Timeout**: This property (which is used primarily by Ethernet based drivers) controls the amount of time required to establish a socket connection to a remote device. The device's connection time often takes longer than normal communications requests to that same device. The valid range is 1 to 30 seconds. The default is typically 3 seconds, but can vary depending on the driver's specific nature. If this setting is not supported by the driver, it is disabled.

Note: Due to the nature of UDP connections, the connection timeout setting is not applicable when communicating via UDP.

Request Timeout: Specify an interval used by all drivers to determine how long the driver waits for a response from the target device to complete. The valid range is 50 to 9999999 milliseconds (167 minutes). The default is usually 1000 milliseconds, but can vary depending on the driver. The default timeout for most serial drivers is based on a baud rate of 9600 baud or better. When using a driver at lower baud rates, increase the timeout to compensate for the increased time required to acquire data.

Attempts Before Timeout: Specify how many times the driver issues a communications request before considering the request to have failed and the device to be in error. The valid range is 1 to 10. The default is typically 3, but can vary depending on the driver's specific nature. The number of attempts configured for an application depends largely on the communications environment. This property applies to both connection attempts and request attempts.

## **Timing**

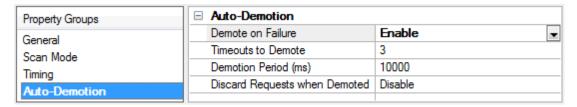
Inter-Request Delay: Specify how long the driver waits before sending the next request to the target device after receiving the response to the previous request. It overrides the normal polling frequency of tags associated with the device, as well as one-time reads and writes. This delay can be useful when dealing with devices with slow turn-around times and in cases where network load is a concern. Configuring a delay for a device affects communications with all other devices on the channel. It is recommended that users separate any device that requires an inter-request delay to a separate channel if possible. Other communications properties (such as communication serialization) can extend this delay. The valid range is 0 to 300,000 milliseconds; however, some drivers may limit the maximum value due to a function of their particular design. The default is 0, which indicates no delay between requests with the target device.

Note: Not all drivers support Inter-Request Delay. This setting does not appear if it is not available.



## Device Properties – Auto-Demotion

The Auto-Demotion properties can temporarily place a device off-scan in the event that a device is not responding. By placing a non-responsive device offline for a specific time period, the driver can continue to optimize its communications with other devices on the same channel. After the time period has been reached, the driver reattempts to communicate with the non-responsive device. If the device is responsive, the device is placed on-scan; otherwise, it restarts its off-scan time period.



**Demote on Failure**: When enabled, the device is automatically taken off-scan until it is responding again.

Tip: Determine when a device is off-scan by monitoring its demoted state using the \_AutoDemoted system tag.

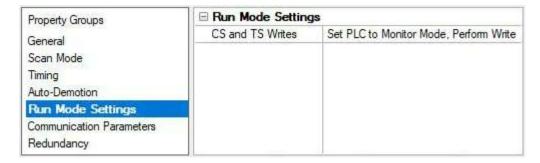
**Timeouts to Demote**: Specify how many successive cycles of request timeouts and retries occur before the device is placed off-scan. The valid range is 1 to 30 successive failures. The default is 3.

**Demotion Period**: Indicate how long the device should be placed off-scan when the timeouts value is reached. During this period, no read requests are sent to the device and all data associated with the read requests are set to bad quality. When this period expires, the driver places the device on-scan and allows for another attempt at communications. The valid range is 100 to 3600000 milliseconds. The default is 10000 milliseconds.

**Discard Requests when Demoted**: Select whether or not write requests should be attempted during the off-scan period. Disable to always send write requests regardless of the demotion period. Enable to discard writes; the server automatically fails any write request received from a client and does not post a message to the Event Log.

# Device Properties – Run Mode Settings

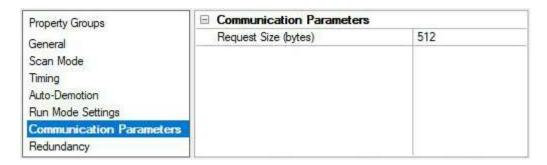
This group specifies the driver's behavior when the device makes writes to Timer Status and Counter Status while in Run Mode



CS and TS Writes: Select how the device performs writes in Run mode. The default setting is Fail Write, Log Message.

- Fail Write, Log Message: posts a message to the Event Log when the Write command fails.
- Set PLC to Monitor Mode, Perform Write: changes the PLC to Monitor Mode before performing the
  write.
- Set PLC to Monitor Mode, Write, Reset to Run: changes the PLC to Monitor Mode before performing the write. Once complete, the PLC is reset to Run mode.

## **Device Properties – Communication Parameters**



**Request Size**: 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.

## Device Properties – Redundancy

Property Groups	☐ Redundancy	Redundancy	
General	Secondary Path	Channel.Device1	
Scan Mode	Operating Mode	Switch On Failure	
	Monitor Item		
Timing Auto-Demotion	Monitor Interval (s)	300	
	Return to Primary ASAP	Yes	
Redundancy			

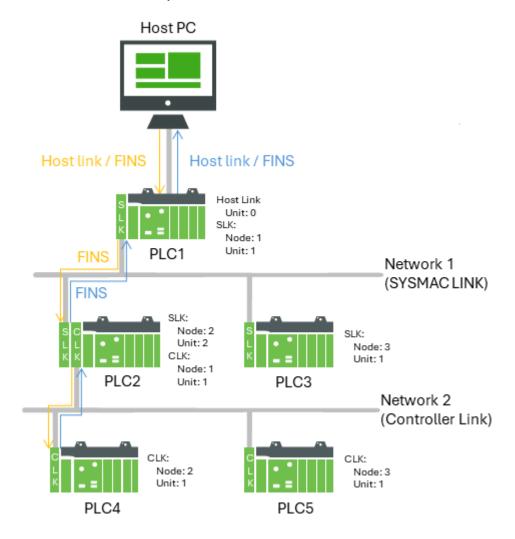
Redundancy is available with the Media-Level Redundancy Plug-In.

Consult the website, a sales representative, or the user manual for more information.

### **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 serves 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 sets DA2 and SA2 to zero, meaning that communications is between the host computer and the destination node's CPU module. This driver also sets SNA and SA1 to zero to indicate that communication is through the Host Link port of the interface device. The interface device (PLC 1) resets 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 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 SYSMACK 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

• 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	Word, Short, BCD Long, DWord, LBCD,	Read/Write
	ARxx.00-ARxx.15	Float Boolean	
Auxiliary Relay as string with HiLo byte order	AR00.056H-AR27.002H I is string length, range 2 to 56 chars	String	Read/Write
Auxiliary Relay as string with LoHi byte order	AR00.056L-AR27.002L I 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 I 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 I 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 .I is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	DM0000.256L- DM6655.002L .I is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string Using Only the	DM0000.128D-	String	Read/Write

Device Type	Range	Data Type	Access
High Order byte of each word	DM6655.001D		
	.l is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	DM0000.128E- DM6655.001E	String	Read/Write
	.l is string length, range 1 to 128 chars		
Expansion Data Memory (current bank)	EM0000-EM6143	Word, Short, BCD	Read/Write
	EM0000-EM6142	Long, DWord, LBCD,	
	EMxxxx.00-EMxxxx.15	Float Boolean	
Expansion Data Memory (current bank) as string with HiLo byte order	EM0000.256H- EM6143.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string with LoHi byte order	EM0000.256L- EM6143.002L	String	Read/Write
	.l is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte of	EM0000.128D- EM6143.001D	String	Read/Write
each word	.I is string length, range 1 to 128 chars		
Expansion Data Memory (current bank) as string Using Only the Low Order byte of	EM0000.128E- EM6143.001E	String	Read/Write
each word	.I is string length, range 1 to 128 chars		
Expansion Data Memory	EM00:0000-EM07:6143	Word, Short, BCD	Read/Write
	EM00:0000-EM07:6142	Long, DWord, LBCD,	
	EMx:x.00-EMxx:xxxx.15	Float Boolean	
Expansion Data Memory as string with	EM00:0000.256H-	String	Read/Write
HiLo byte order	EM07:6143.002H		
	.l is string length, range 2 to 256 chars		
Expansion Data Memory as string with LoHi byte order	EM00:0000.256L- EM07:6143.002L	String	Read/Write
	.l is string length, range 2 to 256 chars		
Expansion Data Memory as string Using Only the High Order byte of each word	EM00:0000.128D- EM07:6143.001D	String	Read/Write
	.l is string length, range 1 to 128 chars		
Expansion Data Memory as string Using Only the Low Order byte of each word	EM00:0000.128E- EM07:6143.001E	String	Read/Write
	.l is string length, range 1 to 128 chars		
Holding Relay	HR00-HR99	Word, Short, BCD	Read/Write
	HR00-HR98	Long, DWord, LBCD,	
	HRxx.00-HRxx.15	Float Boolean	

Device Type	Range	Data Type	Access
order	.I is string length, range 2 to 200 chars		
Holding Relay as string with LoHi byte	HR00.200L-HR99.002L	String	Read/Write
order	.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	String	Read/Write
,	.l is string length, range 1 to 100 chars		
Holding Relay as string Using Only the	HR00.100E-HR99.001E	String	Read/Write
Low Order byte of each word	.l is string length, range 1 to 100 chars		
Internal Relay	IR000-IR511	Word, Short, BCD	Read/Write
	IR000-IR510	Long, DWord, LBCD, Float	
	IRxxx.00-IRxxx.15	Boolean	
Internal Relay as a string with HiLo byte	IR000.256H-IR511.002H	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Internal Relay as a string with LoHi byte	IR000.256L-IR511.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Internal Relay as a string	IR000.128D-IR511.001D	String	Read/Write
Using Only the High Order byte of each word	I is string length, range 1 to 128 chars		
Internal Relay as a string Using Only the	IR000.128E-IR511.001E	String	Read/Write
Low Order byte of each word	I is string length, range 1 to 128 chars		
Link Relays	LR00-LR63	Word, Short, BCD	Read/Write
	LR00-LR62	Long, DWord, LBCD, Float	
	LRxx.00-LRxx.15	Boolean	
Link Relay as string with HiLo byte order	LR00.128H-LR63.002H	String	Read/Write
	.I is string length, range 2 to 128 chars		
Link Relay as string with LoHI byte order	LR00.128L-LR63.002L	String	Read/Write
	.I is string length, range 2 to 128 chars		
Link Relay as string Using Only the High	LR00.064D-LR63.001D	String	Read/Write
Order byte of each word	.l is string length, range 1 to 64 chars		
Link Relay as string Using Only the Low	LR00.064E-LR63.001E	String	Read/Write
Order byte of each word	.I is string length, range 1 to 64 chars		
Temporary Relay	TR	Word, Short	Read/Write
Timov/Counter	TR0-TR7	Boolean	Dood Marie
Timer/Counter	TC000-TC511	BCD, Word, Short	Read/Write Read/Write
Timer/Counter as string with HiLo byte order	TC000.256H- TC511.002H	String	Read/vvrite
T. (0	.l is string length, range 2 to 256 chars	l Oudre or	D 1011
Timer/Counter as string with LoHi byte	TC000.256L-TC511.002L	String	Read/Write

Device Type	Range	Data Type	Access
order	.I is string length, range 2 to 256 chars		
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D- TC511.001D	String	Read/Write
	.I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
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 contains 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 Setup.

### **Examples**

- 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
- 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 and String. 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 Setup.

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 also modifies 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 <u>BCD Support</u>, <u>String Support</u>, and <u>Array Support</u>.

Device Type	Range	Data Type	Access
Data Memory	DM000-DM511	Word, Short, BCD	Read/Write
	DM000-DM510	Long, DWord, LBCD,	
	DMxxx.00-DMxxx.15	Float	
		Boolean	
Data Memory as string with HiLo byte	DM000.256H-	String	Read/Write
order	DM511.002H		
	.I is string length, range 2		
	to 256 chars		
Data Memory as string with LoHi byte	DM000.256L-	String	Read/Write
order	DM511.002L		
	.l is string length, range 2		
	to 256 chars		
Data Memory as string Using Only the	DM000.128D-	String	Read/Write
High Order byte of each word	DM511.001D		
	.l is string length, range 1		
	to 128 chars		
Data Memory as string Using Only the	DM000.128E-	String	Read/Write
Low Order byte of each word	DM511.001E		
	.l is string length, range 1		
	to 128 chars		

Device Type	Range	Data Type	Access
Holding Relay	HR00-HR31	Word, Short, BCD	Read/Write
	HR00-HR30	Long, DWord, LBCD,	
	HRxx.00-HRxx.15	Float	
		Boolean	
Holding Relay as string with HiLo byte	HR00.064H-HR31.002H	String	Read/Write
order	I is string length, range 2 to 64 chars		
Holding Relay as string with LoHi byte	HR00.064L-HR31.002L	String	Read/Write
order	I is string length, range 2 to 64 chars		
Holding Relay as string Using Only the	HR00.032D-HR31.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 32 chars		
Holding Relay as string Using Only the	HR00.032E-HR31.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 32 chars		
Internal Relay	IR00-IR63	Word, Short, BCD	Read/Write
	IR00-IR62	Long, DWord, LBCD,	
	IRxx.00-IRxx.15	Float	
		Boolean	
Internal Relay as a string with HiLo byte order	IR00.128H-IR63.002H	String	Read/Write
	I is string length, range 2 to 128 chars		
Internal Relay as a string with LoHi byte	IR00.128L-IR63.002L	String	Read/Write
order	I is string length, range 2 to 128 chars		
Internal Relay as a string Using Only the	IR00.064D-IR63.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 64 chars		
Internal Relay as a string Using Only the	IR00.064E-IR63.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 64 chars		
Link Relays	LR00-LR31	Word, Short,	Read/Write
	LR00-LR30	Long, DWord, LBCD,	
	LRxx.00-LRxx.15	Float	
		Boolean	
Link Relays as string with HiLo byte order	LR00.064H-LR31.002H	String	Read/Write
	.I is string length, range 2 to 64 chars		
Link Relays as string with LoHi byte order	LR00.064L-LR31.002L	String	Read/Write
	I is string length, range 2 to 64 chars		
Link Relays as string Using Only the High Order byte of each word	LR00.032D-LR31.001D	String	Read/Write
	.I is string length, range 1 to 32 chars		
Link Relays as string Using Only the Low	LR00.032E-LR31.001E	String	Read/Write
Order byte of each word	.I is string length, range 1 to 32 chars		
Temporary Relay	TR	Word, Short	Read/Write
	TR0-TR7	Boolean	
Timer/Counter	TC000-TC127	BCD, Word, Short	Read/Write

Device Type	Range	Data Type	Access
Timer/Counter as string with HiLo byte order	TC000.256H- TC127.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Timer/Counter as string with LoHi byte	TC000.256L-TC127.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Timer/Counter as string Using Only the High Order byte of each word	TC000.128D- TC127.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E- TC127.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
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

IROD @ Short = -50 IROD @ Word = 8050 (MSB set) IRO @ 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 contains 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 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 and String. There are two methods of addressing an array. Examples are given using at a 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 **Setup**.

● 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 also modifies 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	Word, Short, BCD	Read/Write
	AR00-AR26	Long, DWord, LBCD,	
	ARxx.00-ARxx.15	Float	
		Boolean	
Auxiliary Relay as string with HiLo byte	AR00.056H-AR27.002H	String	Read/Write
order	.l is string length, range 2 to 56 chars		
Auxiliary Relay as string with LoHi byte	AR00.056L-AR27.002L	String	Read/Write
order	.l is string length, range 2 to 56 chars		
Auxiliary Relay as string Using Only the	AR00.028D-AR27.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 28 chars		
Auxiliary Relay as string Using Only the	AR00.028E-AR27.001E	String	Read/Write
Low Order byte of each word	.l is string length, range 1 to 28 chars		

Device Type	Range	Data Type	Access
Data Memory	DM0000-DM4095	Word, Short, BCD	Read/Write
	DM0000-DM4094	Long, DWord, LBCD,	
	DMxxxx.00-DMxxxx.15	Float	
		Boolean	
Data Memory as string with HiLo byte	DM0000.256H-	String	Read/Write
order	DM4095.002H		
	I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	DM0000.256L- DM4095.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	DM0000.128D- DM4095.001D	String	Read/Write
	.l is string length, range 1 to 128 chars		
Data Memory as string Using Only the	DM0000.128E-	String	Read/Write
Low Order byte of each word	DM4095.001E		
	I is string length, range 1 to 128 chars		
Holding Relay	HR00-HR99	Word, Short, BCD	Read/Write
	HR00-HR98	Long, DWord, LBCD,	
	HRxx.00-HRxx.15	Float	
		Boolean	
Holding Relay as string with HiLo byte order	HR00.200H-HR99.002H	String	Read/Write
oluei	.l is string length, range 2 to 200 chars		
Holding Relay as string with LoHi byte	HR00.200L-HR99.002L	String	Read/Write
order	.I is string length, range 2 to 200 chars		
Holding Relay as string Using Only the	HR00.100D-HR99.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 100 chars		
Holding Relay as string Using Only the	HR00.100E-HR99.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 100 chars		
Internal Relay	IR000-IR255	Word, Short, BCD	Read/Write
	IR000-IR254	Long, DWord, LBCD,	
	IRxxx.00-IRxxx.15	Float	
		Boolean	
Internal Relay as a string with HiLo byte	IR000.256H-IR255.002H	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Internal Relay as a string with LoHi byte	IR000.256L-IR255.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Internal Relay as a string Using Only the	IR000.128D-IR255.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 128 chars		
Internal Relay as a string Using Only the	IR000.128E-IR255.001E	String	Read/Write
Low Order byte of each word	.l is string length, range 1 to 128 chars		

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

#### **Examples**

- 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
- 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 and String. 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 **Setup**.

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 also modifies 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	Word, Short, BCD	Read/Write
	AR00-AR26	Long, DWord, LBCD,	
	ARxx.00-ARxx.15	Float	
		Boolean	
Auxiliary Relay as string with HiLo byte	AR00.056H-AR27.002H	String	Read/Write
order	.l is string length, range 2 to 56 chars		
Auxiliary Relay as string with LoHi byte	AR00.056L-AR27.002L	String	Read/Write
order	.l is string length, range 2 to 56 chars		
Auxiliary Relay as string Using Only the	AR00.028D-AR27.001D	String	Read/Write
High Order byte of each word	.l is string length, range 1 to 28 chars		
Auxiliary Relay as string Using Only the	AR00.028E-AR27.001E	String	Read/Write
Low Order byte of each word	I is string length, range 1 to 28 chars		
Data Memory	DM0000-DM6655	Word, Short, BCD	Read/Write
	DM0000-DM6654	Long, DWord, LBCD,	
	DMxxxx.00-DMxxxx.15	Float	
	D140000 050U	Boolean	D 1047 ::
Data Memory as string with HiLo byte order	DM0000.256H- DM6655.002H	String	Read/Write
	I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	DM0000.256L- DM6655.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	DM0000.128D- DM6655.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	DM0000.128E- DM6655.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
Holding Relay	HR00-HR99	Word, Short, BCD	Read/Write
	HR00-HR98	Long, DWord, LBCD,	
	HRxx.00-HRxx.15	Float	
		Boolean	
Holding Relay as string with HiLo byte	HR00.200H-HR99.002H	String	Read/Write
order	.I is string length, range 2 to 200 chars		
Holding Relay as string with LoHi byte order	HR00.200L-HR99.002L	String	Read/Write
	.I 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	String	Read/Write

Device Type	Range	Data Type	Access
	.l is string length, range 1 to 100 chars		
Holding Relay as string Using Only the Low Order byte of each word	HR00.100E-HR99.001E I 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 I is string length, range 2 to 256 chars	String	Read/Write
Internal Relay as a string with LoHi byte order	IR000.256L-IR255.002L I 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 .I 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 I 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 I is string length, range 2 to 128 chars	String	Read/Write
Link Relays as string with LoHi byte order	LR00.128L-LR63.002L I 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 I 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 I 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	String	Read/Write
Timer/Counter as string with LoHi byte order	to 256 chars  TC000.256L-TC511.002L  I 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 .I is string length, range 1 to 128 chars	String	Read/Write

Device Type	Range	Data Type	Access
Timer/Counter as string Using Only the Low Order byte of each word	TC000.128E- TC511.001E	String	Read/Write
	I is string length, range 1 to 128 chars		
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 contains 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 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
- To address a string starting at DM2000 with a length of 55 bytes and Only the High Order byte, enter: DM2000.055D
- 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 and String. 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 Setup.

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 also modifies 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,	Read Only
	A256-A511	Float	Read/Write
	A256-A510	Word, Short, BCD	Read Only
	A000.00-A000.15- A255.00-A255.15	Long, DWord, LBCD, Float	
	A256.00-A256.15-	Boolean	
	A511.00-A511.15	Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A255.002H	String	Read/Write
order	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
	.l is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A255.001E	String	Read/Write
Low Order byte of each word	A256.128E-A511.001E	String	Read Only

Device Type	Range	Data Type	Access
	.I is string length, range 1 to 128 chars		
CIO	CIO0000-CIO2555	Word, Short, BCD	Read/Write
	CIO0000-CIO2554	Long, DWord, LBCD,	
	CIOxxxx.00-CIOxxxx.15	Float	
	0100000 05011	Boolean	D 1044 ::
CIO Memory as string with HiLo byte order	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
	.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	String	Read/Write
	.I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
Counter	C000-C511	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C000.256H-C511.002H	String	Read/Write
	.l is string length, range 2 to 256 chars		
Counter as string with LoHi byte order	C000.256L-C511.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Counter as string Using Only the High	C000.128D-C511.001D	String	Read/Write
Order byte of each word	.I is string length, range 1 to 128 chars		
Counter as string Using Only the Low	C000.128E-C511.001E	String	Read/Write
Order byte of each word	.I is string length, range 1 to 128 chars		
Counter Status	CS000-CS511	Boolean	Read/Write
CPU Bus Link	G000-G255	Word, Short, BCD	Read/Write
	G000-G254	Long, DWord, LBCD,	
	Gxxx.00-Gxxx.15	Float	
		Boolean	
CPU Bus Link as string with HiLo byte order	G000.256H-G255.002H	String	Read/Write
ordei	.l is string length, range 2 to 256 chars		
CPU Bus Link as string with LoHi byte	G000.256L-G255.002L	String	Read/Write
order	.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	String	Read/Write
	.I is string length, range 1 to 128 chars		
CPU Bus Link as string Using Only the	G000.128E-G255.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1		

Device Type	Range	Data Type	Access
	to 128 chars		
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	String	Read/Write
	.l is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	D0000.256L-D8191.002L .I 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 .I 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 .I 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 I is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T000.256L-T511.002L I 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 I 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 I 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

<sup>\*</sup>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 contains 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 <a href="Setup">Setup</a>.

#### **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
- To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter: D2000.055D
- 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, Step Timer, and String. 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 **Setup**.

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 also modifies 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,	Read Only
	A256-A511	Float	Read/Write
	A256-A510	Word, Short, BCD	Read Only
	A000.00-A000.15- A255.00-A255.15	Long, DWord, LBCD, Float	
	A256.00-A256.15-	Boolean	
	A511.00-A511.15	Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A255.002H	String	Read/Write
order	A256.256H-A511.002H	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte	A000.256L-A255.002L	String	Read/Write
order	A256.256L-A511.002L	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the	A000.128D-A255.001D	String	Read/Write
High Order byte of each word	A256.128D-A511.001D	String	Read Only
	.I is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A255.001E	String	Read/Write
Low Order byte of each word	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,	
	CIOxxxx.00-CIOxxxx.15	Float	
		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	String	Read/Write
	.I is string length, range 1		

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

Device Type	Range	Data Type	Access
	.I is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	D00000.128E- D24575.001E .I 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 .I 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 .I 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 .I 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 .I 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 .I 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 .I 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 .I 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 .I 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 Temporary Relay	STS0000-STS1023 TR TR0-TR7	Boolean Word, Short Boolean	Read/Write Read/Write

Device Type	Range	Data Type	Access
Timer	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H I is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L I 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 I 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 I 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

<sup>\*</sup>Arrays are not supported.

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 CV1000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register contains 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 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
- 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, Step Timer, and String. 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 Setup.

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 also modifies 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,	Read Only
	A256-A511	Float	Read/Write
	A256-A510	Word, Short, BCD	Read Only
	A000.00-A000.15-	Long, DWord, LBCD,	
	A255.00-A255.15	Float	
	A256.00-A256.15-	Boolean	
	A511.00-A511.15	Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A255.002H	String	Read/Write
order	A256.256H-A511.002H	String	Read Only
	.I is string length, range 2 to 256 chars		

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

Device Type	Range	Data Type	Access
CPU Bus Link as string with HiLo byte	G000.256H-G255.002H	String	Read/Write
order	.l is string length, range 2 to 256 chars		
CPU Bus Link as string with LoHi byte	G000.256L-G255.002L	String	Read/Write
order	.l is string length, range 2 to 256 chars		
CPU Bus Link as string Using Only the	G000.128D-G255.001D	String	Read/Write
High Order byte of each word	.l is string length, range 1 to 128 chars		
CPU Bus Link as string Using Only the	G000.128E-G255.001E	String	Read/Write
Low Order byte of each word	I is string length, range 1 to 128 chars		
Data Memory	D00000-D24575	Word, Short, BCD	Read/Write
	D00000-D24574	Long, DWord, LBCD, Float	
	Dxxxxx.00-Dxxxxx.15	Boolean	
Data Memory as string with HiLo byte order	D00000.256H- D24575.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	D00000.256L- D24575.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	D00000.128D- D24575.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	D00000.128E- D24575.001E	String	Read/Write
·	.I is string length, range 1 to 128 chars		
Data Register	DR0-DR2	Word, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765	Word, Short, CD	Read/Write
	E00000-E32764	Long, DWord, LBCD, Float	
	Exxxxx.00-Exxxxx.15	Boolean	
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H- E32765.002H	String	Read/Write
as same marrines syle order.	I is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L- E32765.002L	String	Read/Write
, ,	.I is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte	E00000.128D- E32765.001D	String	Read/Write
of each word	.l is string length, range 1 to 128 chars		
Expansion Data Memory (current bank) as string Using Only the Low Order byte of	E00000.128E- E32765.001E	String	Read/Write
each word	.I is string length, range 1		

Device Type	Range	Data Type	Access
	to 128 chars		
Expansion Data Memory	E00:00000-E07:32765	Word, Short, BCD	Read/Write
	E00:00000-E07:32764	Long, DWord, LBCD,	
	Ex:x.00-Exx:xxxxx.15	Float	
		Boolean	
Expansion Data Memory as string with HiLo byte order	E00:00000.256H - E07:32765.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Expansion Data Memory as string with LoHi byte order	E00:00000.256L - E07:32765.002L	String	Read/Write
	.I 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- E07:32765.001D	String	Read/Write
	.I 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- E07:32765.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
Index Register	IR0-IR2	Word, Short, BCD*	Read/Write
Step Timer	ST0000-ST1023	Word, Short, BCD*	Read/Write
Step Timer Status	STS0000-STS1023	Boolean	Read/Write
Temporary Relay	TR	Word, Short	Read/Write
	TR0-TR7	Boolean	
Timer	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Timer as string with LoHi byte order	T0000.256L-T1023.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Timer as string Using Only the High Order	T0000.128D-T1023.001D	String	Read/Write
byte of each word	.I is string length, range 1 to 128 chars		
Timer as string Using Only the Low Order	T0000.128E-T1023.001E	String	Read/Write
byte of each word	.I is string length, range 1 to 128 chars		
Timer Status	TS0000-TS1023	Boolean	Read/Write
Transition Flag	TN0000-TN1023	Boolean	Read/Write

<sup>\*</sup>Arrays are not supported.

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

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 CV2000 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register contains 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 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
- 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, Index Register, Step Timer, and String. 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 **Setup**.

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

Device Type	Range	Data Type	Access
CIO Memory as string Using Only the Low Order byte of each word	CIO0000.128E- CIO2555.001E	String	Read/Write
	.l is string length, range 1 to 128 chars		
Counter	C000-C511	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C000.256H-C511.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Counter as string with LoHi byte order	C000.256L-C511.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Counter as string Using Only the High	C000.128D-C511.001D	String	Read/Write
Order byte of each word	.l is string length, range 1 to 128 chars		
Counter as string Using Only the Low	C000.128E-C511.001E	String	Read/Write
Order byte of each word	I is string length, range 1 to 128 chars		
Counter Status	CS000-CS511	Boolean	Read/Write
CPU Bus Link	G000-G255	Word, Short, BCD	Read/Write
	G000-G254	Long, DWord, LBCD, Float	
	Gxxx.00-Gxxx.15	Boolean	
CPU Bus Link as string with HiLo byte	G000.256H-G255.002H	String	Read/Write
order	.l is string length, range 2		Tread/Wille
	to 256 chars		
CPU Bus Link as string with LoHi byte	G000.256L-G255.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
CPU Bus Link as string Using Only the	G000.128D-G255.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 128 chars		
CPU Bus Link as string Using Only the	G000.128E-G255.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 128 chars		
Data Memory	D0000-D8191	Word, Short, BCD	Read/Write
	D0000-D8190	Long, DWord, LBCD, Float	
	Dxxxx.00-Dxxxx.15	Boolean	
Data Memory as string with HiLo byte order	D0000.256H- D8191.002H	String	Read/Write
Cidal	I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte	D0000.256L-D8191.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	D0000.128D- D8191.001D	String	Read/Write
Thigh Older byte of each word	I is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	D0000.128E- D8191.001E	String	Read/Write

Device Type	Range	Data Type	Access
	.I 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	Word, Short	Read/Write
	TR0-TR7	Boolean	
Timer	T000-T511	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T000.256H-T511.002H	String	Read/Write
	.l is string length, range 2 to 256 chars		
Timer as string with LoHi byte order	T000.256L-T511.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Timer as string Using Only the High Order	T000.128D-T511.001D	String	Read/Write
byte of each word	.l is string length, range 1 to 128 chars		
Timer as string Using Only the Low Order	T000.128E-T511.001E	String	Read/Write
byte of each word	.l is string length, range 1 to 128 chars		
Timer Status	TS000-TS511	Boolean	Read/Write

<sup>\*</sup>Arrays are not supported.

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-CPU01 model supports reading and writing numerous device types as an ASCII string. When using data memory for string data, each register contains 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 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
- To address a string starting at D2000 with a length of 55 bytes and Only the High Order byte, enter: D2000.055D
- 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 String. 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 **Setup**.

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 also modifies 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 <u>BCD Support</u>, <u>String Support</u>, and <u>Array Support</u>.

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

Device Type	Range	Data Type	Access
Auxiliary Relay as string with HiLo byte	A000.256H-A255.002H	String	Read/Write
order	A256.256H-A511.002H	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte	A000.256L-A255.002L	String	Read/Write
order	A256.256L-A511.002L	String	Read Only
	.l is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the	A000.128D-A255.001D	String	Read/Write
High Order byte of each word	A256.128D-A511.001D	String	Read Only
	.l is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A255.001E	String	Read/Write
Low Order byte of each word	A256.128E-A511.001E	String	Read Only
	.l is string length, range 1 to 128 chars		
CIO	CIO0000-CIO2555	Word, Short, BCD	Read/Write
	CIO0000-CIO2554	Long, DWord, LBCD,	
	CIOxxxx.00-CIOxxxx.15	Float	
CIO Momony on string with Hill a byta	CIOOOOO SEGH	Boolean	Read/Write
CIO Memory as string with HiLo byte order	CIO0000.256H- CIO2555.002H	String	Read/write
	.l is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte	CIO0000.256L-	String	Read/Write
order	CIO2555.002L		
	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	String	Read/Write
riigii Ordei byte oi eacii 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	String	Read/Write
order byte or each word	.l is string length, range 1		
	to 128 chars		
Counter	C0000-C1023	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H- C1023.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Counter as string with LoHi byte order	C0000.256L-C1023.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Counter as string Using Only the High Order byte of each	C0000.128D- C1023.001D	String	Read/Write
word	.I is string length, range 1 to 128 chars		
Counter as string Using Only the Low Order byte of each word	C0000.128E- C1023.001E	String	Read/Write
Order Byte of each word	.l is string length, range 1		
	to 128 chars		

Device Type	Range	Data Type	Access
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 I is string length, range 2 to 256 chars	String	Read/Write
CPU Bus Link as string with LoHi byte order	G000.256L-G255.002L I 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 I 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 I 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 .I is string length, range 2 to 256 chars	String	Read/Write
Data Memory as string with LoHi byte order	D00000.256L- D24575.002L .1 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 .I 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 .I 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	T0000-T1023	BCD, Word, Short	Read/Write
Timer as string with HiLo byte order	T0000.256H-T1023.002H I is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T1023.002L I 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 I is string length, range 1 to 128 chars	String	Read/Write

Device Type	Range	Data Type	Access
Timer as string Using Only the Low Order	T0000.128E-T1023.001E	String	Read/Write
byte of each word	.I is string length, range 1 to 128 chars		
Timer Status	TS0000-TS1023	Boolean	Read/Write

<sup>\*</sup>Arrays are not supported.

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 contains 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 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
- 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 String. 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 **Setup**.

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 also modifies 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 <u>BCD Support</u>, <u>String Support</u>, and <u>Array Support</u>.

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

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

Device Type	Range	Data Type	Access
CPU Bus Link as string Using Only the	G000.128E-G255.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 128 chars		
Data Memory	D00000-D24575	Word, Short, BCD	Read/Write
	D00000-D24574	Long, DWord, LBCD,	
	Dxxxxx.00-Dxxxxx.15	Float Boolean	
Data Memory as string with HiLo byte order	D00000.256H- D24575.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	D00000.256L- D24575.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	D00000.128D- D24575.001D	String	Read/Write
	.l is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	D00000.128E- D24575.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
Data Register	DR0-DR2	Word, Short, BCD*	Read/Write
Expansion Data Memory (current bank)	E00000-E32765	Word, Short, BCD	Read/Write
	E00000-E32764	Long, DWord, LBCD, Float	
	Exxxxx.00-Exxxxx.15	Boolean	
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H- E32765.002H	String	Read/Write
	.l is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L- E32765.002L	String	Read/Write
	.l is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte	E00000.128D- E32765.001D	String	Read/Write
of each word	.l is string length, range 1 to 128 chars		
Expansion Data Memory (current bank) as string Using Only the Low Order byte of	E00000.128E- E32765.001E	String	Read/Write
each word	.l is string length, range 1 to 128 chars		
Expansion Data Memory	E00:00000-E07:32765	Word, Short, BCD	Read/Write
	E00:00000-E07:32764	Long, DWord, LBCD,	
	Ex:x.00-Exx:xxxxx.15	Float Boolean	
Expansion Data Memory as string with HiLo byte order	E00:00000.256H- E07:32765.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		

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

<sup>\*</sup>Arrays are not supported.

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 contains 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 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
- 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 String. 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 **Setup**.

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 also modifies 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	Word, Short, BCD	Read Only
	A000-A446	Long, DWord, LBCD,	Read/Write
	A448-A959	Float	Read Only
	A448-A958	Word, Short, BCD	Read/Write
	A000.00-A000.15- A447.00-A447.15	Long, DWord, LBCD, Float	
	A448.00-A448.15- A959.00-A959.15	Boolean Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A447.002H	String	Read Only
order	A448.256H-A959.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte	A000.256L-A447.002L	String	Read Only
order	A448.256L-A959.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the	A000.128D-A447.001D	String	Read Only
High Order byte of each word	A448.128D-A959.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A447.001E	String	Read Only
Low Order byte of each word	A448.128E-A959.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
CIO	CIO0000-CIO6143	Word, Short, BCD	Read/Write
	CIO0000-CIO6142	Long, DWord, LBCD,	
	CIOxxxx.00-CIOxxxx.15	Float	
		Boolean	
CIO Memory as string with HiLo byte order	CIO0000.256H- CIO6143.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte order	CIO0000.256L- CIO6143.002L	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- CIO6143.001D	String	Read/Write
	.I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
Counter	C0000-C4095	BCD, Word, Short	Read/Write
Counter as string with HiLo byte order	C0000.256H- C4095.002H	String	Read/Write

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

Device Type	Range	Data Type	Access
	to 128 chars		
Expansion Data Memory (current bank) as string Using Only the Low Order byte of each word	E00000.128E- E32767.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
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	String	Read/Write
	to 256 chars		
Expansion Data Memory as string with LoHi byte order	E00:00000.256L - E12:32767.002L .I is string length, range 2	String	Read/Write
	to 256 chars		
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D- E12:32767.001D	String	Read/Write
	I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
Holding Relay	H0000-H1535	Word, Short, BCD	Read/Write
	H0000-H1534	Long, DWord, LBCD, Float	
	Hxxxx.00-Hxxxx.15	Boolean	
Holding Relay as string with HiLo byte order	H0000.256H- H1535.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Holding Relay as string with LoHi byte	H0000.256L-H1535.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Holding Relay as string Using Only the High Order byte of each word	H0000.128D- H1535.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Holding Relay as string Using Only the Low Order byte of each word	H0000.128E- H1535.001E	String	Read/Write
ŕ	.I is string length, range 1 to 128 chars		
Index Register	IR00-IR15	<b>DWord</b> , 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 I is string length, range 2 to 256 chars	String	Read/Write
Timer as string with LoHi byte order	T0000.256L-T4095.002L	String	Read/Write

Device Type	Range	Data Type	Access
	.l is string length, range 2 to 256 chars		
Timer as string Using Only the High	T0000.128D-T4095.001D	String	Read/Write
Order byte of each word	.I is string length, range 1 to 128 chars		
Timer as string Using Only the Low Order	T0000.128E-T4095.001E	String	Read/Write
byte of each word	.I is string length, range 1 to 128 chars		
Timer Status	TS0000-TS4095	Boolean	Read/Write*
Working Relay	W000-W511	Word, Short, BCD	Read/Write
	W000-W510	Long, DWord, LBCD,	
	Wxxx.00-Wxxx.15	Float	
		Boolean	
Working Relay as string with HiLo byte	W000.256H-W511.002H	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Working Relay as string with LoHi byte	W000.256L-W511.002L	String	Read/Write
order	.I is string length, range 2 to 256 chars		
Working Relay as string Using Only the	W000.128D-W511.001D	String	Read/Write
High Order byte of each word	.I is string length, range 1 to 128 chars		
Working Relay as string Using Only the	W000.128E-W511.001E	String	Read/Write
Low Order byte of each word	.I is string length, range 1 to 128 chars		

\*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 also modifies the value held in D1. It is recommended that these data types be used 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 contains 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 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 and String. 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 **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 does not change. Because the write succeeds, no error message is returned. The device's value does 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	Word, Short, BCD	Read Only
	A000-A446	Long, DWord, LBCD,	Read/Write
	A448-A959	Float	Read Only
	A448-A958	Word, Short, BCD	Read/Write
	A000.00-A000.15- A447.00-A447.15	Long, DWord, LBCD, Float	
	A448.00-A448.15- A959.00-A959.15	Boolean Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A447.002H	String	Read Only
order	A448.256H-A959.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string with LoHi byte	A000.256L-A447.002L	String	Read Only
order	A448.256L-A959.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the	A000.128D-A447.001D	String	Read Only
High Order byte of each word	A448.128D-A959.001D	String	Read/Write
	.I is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A447.001E	String	Read Only
Low Order byte of each word	A448.128E-A959.001E	String	Read/Write
	.I is string length, range 1 to 128 chars		
CIO	CIO0000-CIO6143	Word, Short, BCD	Read/Write
	CIO0000-CIO6142	Long, DWord, LBCD,	
	CIOxxxx.00-CIOxxxx.15	Float	
		Boolean	
CIO Memory as string with HiLo byte order	CIO0000.256H- CIO6143.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte order	CIO0000.256L- CIO6143.002L	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- CIO6143.001D	String	Read/Write
	.I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
Counter	C0000-C4095	BCD, Word, Short	Read/Write
Counter Status	CS0000-CS4095	Boolean	Read/Write*
Data Memory	D00000-D32767	Word, Short, BCD	Read/Write

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

Device Type	Range	Data Type	Access
	to 256 chars		
Expansion Data Memory as string Using Only the High Order byte of each word	E00:00000.128D- E12:32767.001D	String	Read/Write
	.I 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	String	Read/Write
	.I is string length, range 1 to 128 chars		
Holding Relay	H0000-H1535	Word, Short, BCD	Read/Write
	H0000-H1534	Long, DWord, LBCD,	
	Hxxxx.00-Hxxxx.15	Float	
		Boolean	
Index Register	IR00-IR15	<b>DWord</b> , 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	Word, Short, BCD	Read/Write
	W000-W510	Long, DWord, LBCD,	
	Wxxx.00-Wxxx.15	Float	
		Boolean	

\*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 also modifies the value held in D1. It is recommended that these data types be used 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 contains 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 Setup.

#### **Examples**

- 1. 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
- 3. 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 and String. 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 **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 does not change. Because the write succeeds, no error message is returned. The device's value does 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	Word, Short, BCD	Read Only
	A000-A446	Long, DWord, LBCD,	Read/Write
	A448-A1471	Float	Read Only
	A448-A1470	Word, Short, BCD	Read Only
	A10000-A11535	Long, DWord, LBCD,	Read/Write
	A10000-A11534	Float	Read Only
	A000.00-A000.15-	Word, Short, BCD Long, DWord, LBCD,	
	A447.00-A447.15	Float	
	A448.00-A448.15- A1471.00-A1471.15	Boolean	
	A10000.00-A10000.15-	Boolean	
	A11535.00-A11535.15	Boolean	
Auxiliary Relay as string with HiLo byte	A000.256H-A447.002H	String	Read Only
order	A448.256H-A1471.002H	String	Read/Write
	A10000.256H-	String	Read Only
	A11535.002H	Cumg	Troud Only
	.I is string length, range 2		
	to 256 chars		
Auxiliary Relay as string with LoHi byte	A000.256L-A447.002L	String	Read Only
order	A448.256L-A1471.002L	String	Read/Write
	A10000.256L- A11535.002L	String	Read Only
	.I is string length, range 2 to 256 chars		
Auxiliary Relay as string Using Only the	A000.128D-A447.001D	String	Read Only
High Order byte of each word	A448.128D-A1471.001D	String	Read/Write
	A10000.128D- A11535.001D	String	Read Only
	.I is string length, range 1 to 128 chars		
Auxiliary Relay as string Using Only the	A000.128E-A447.001E	String	Read Only
Low Order byte of each word	A448.128E-A1471.001E	String	Read/Write
	A10000.128E- A11535.001E	String	Read Only
	.I is string length, range 1 to 128 chars		
CIO	CIO0000-CIO6143	Word, Short, BCD	Read/Write
	CIO0000-CIO6142	Long, DWord, LBCD,	
	CIOxxxx.00-CIOxxxx.15	Float	
		Boolean	
CIO Memory as string with HiLo byte order	CIO0000.256H- CIO6143.002H	String	Read/Write
	.I is string length, range 2 to 256 chars		
CIO Memory as string with LoHi byte order	CIO0000.256L- CIO6143.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		

Device Type	Range	Data Type	Access
CIO Memory as string Using Only the High Order byte of each word	CIO0000.128D- CIO6143.001D	String	Read/Write
	.I 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	String	Read/Write
2.22. 2,12 0. 000	.I is string length, range 1 to 128 chars		
Counter	C0000-C4095	BCD, Word, Short	Read/Write
Counter Status	CS0000-CS4095	Boolean	Read/Write*
Data Memory	D00000-D32767	Word, Short, BCD	Read/Write
	D00000-D32766	Long, DWord, LBCD,	
	Dxxxxx.00-Dxxxxx.15	Float Boolean	
Data Memory as string with HiLo byte order	D00000.256H- D32767.002H	String	Read/Write
ordo.	.I is string length, range 2 to 256 chars		
Data Memory as string with LoHi byte order	D00000.256L- D32767.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Data Memory as string Using Only the High Order byte of each word	D00000.128D- D32767.001D	String	Read/Write
	I is string length, range 1 to 128 chars		
Data Memory as string Using Only the Low Order byte of each word	D00000.128E- D32767.001E	String	Read/Write
	.l is string length, range 1 to 128 chars		
Data Register	DR00-DR15	Word, Short, BCD	Read/Write*
	DR00-DR14	Long, DWord, LBCD, Float	
Expansion Data Memory (current bank)	E00000-E32767	Word, Short, BCD	Read/Write
,	E00000-E32766	Long, DWord, LBCD,	
	Exxxxx.00-Exxxxx.15	Float	
	F00000 05011	Boolean	D 1044 ::
Expansion Data Memory (current bank) as string with HiLo byte order	E00000.256H- E32767.002H	String	Read/Write
as suring with the byte order	.I is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string with LoHi byte order	E00000.256L- E32767.002L	String	Read/Write
	.I is string length, range 2 to 256 chars		
Expansion Data Memory (current bank) as string Using Only the High Order byte	E00000.128D- E32767.001D	String	Read/Write
of each word	.I is string length, range 1 to 128 chars		
Expansion Data Memory (current bank)	E00000.128E-	String	Read/Write
as string Using Only the Low Order byte of each word	E32767.001E  I is string length, range 1		
	1.13 Suning length, range 1		

Device Type	Range	Data Type	Access
	to 128 chars		
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 .I is string length, range 2 to 256 chars	String	Read/Write
Expansion Data Memory as string with LoHi byte order	E00:00000.256L- E24:32767.002L .I 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- E24:32767.001D .I 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 .I 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	<b>DWord</b> , 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 also modifies the value held in D1. It is recommended that these data types be used 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 contains 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 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 and String. 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 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 does not change. Because the write succeeds, no error message is returned. The device's value does not change because the device is in Run Mode.

# **Event Log Messages**

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the OPC server help on filtering and sorting the Event Log detail view. Server help contains many common messages, so should also be searched. Generally, the type of message (informational, warning) and troubleshooting information is provided whenever possible.

▼ Tip: Messages that originate from a data source (such as third-party software, including databases) are presented through the Event Log. Troubleshooting steps should include researching those messages online and in vendor documentation.

Device responded with error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

## **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

# Device responded with local node error. | Main error = <code>, Sub error = <code>.

#### **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

# Device responded with destination node error. | Main error = <code>, Sub error = <code>.

#### **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

# Device responded with a communications error. | Main error = <code>, Sub error = <code>.

# Error Type:

Warning

#### Possible Cause:

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device cannot process command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

# **Error Type:**

Warning

#### Possible Cause:

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

# Device responded with routing table error. | Main error = <code>, Sub error = <code>.

# Error Type:

Warning

#### Possible Cause:

The error codes indicate possible causes.

# Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device responded with a command format error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

#### Error Type:

Warning

#### Possible Cause:

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device responded with a command parameter error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

# **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device responded with read not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

# **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

# **Possible Solution:**

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device responded with error in unit. | Main error = <code>, Sub error = <code>.

#### Error Type:

Warning

# **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device cannot accept command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

# **Error Type:**

Warning

## **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

# See Also:

Main and Sub Error Codes

# Access right denied. | Tag address = '<address>', Main error = <code>, Sub error = <code>.

# **Error Type:**

Warning

#### **Possible Cause:**

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

Device responded with write not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>.

# Error Type:

Warning

#### Possible Cause:

The error codes indicate possible causes.

# Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

#### See Also:

Main and Sub Error Codes

# Unable to write to tag. The device is in Run Mode. | Tag address = '<address>'.

# **Error Type:**

Warning

# Device responded with fatal CPU unit error. | End code = <code>.

# **Error Type:**

Warning

# Possible Cause:

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

# Device responded with non-fatal CPU unit error. | End code = <code>.

#### **Error Type:**

Warning

# Possible Cause:

The error codes indicate possible causes.

#### Possible Solution:

Refer to Main and Sub Error Codes for a description possible solutions and corrections.

# Failed to set device back to run mode following write to tag. | Tag address = '<address>'.

# **Error Type:**

Warning

# **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

# Main and Sub Error Codes

Main Code	Sub-Code	Check Point	Probable Cause	Correction
00: Normal completion	00: Normal completion			
	01: Service can- celed		Service was can- celed.	Check the capacity of the destination area in the third node.
		Data link status	Service was can- celed.	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		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.
	06: Node address duplication	Node addresses	The same node address has been set twice in the same network.	Change the address of one of the nodes with the same address.
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 miss-	Instruction control	The third node does	Check the unit address of the

Main Code	Sub-Code	Check Point	Probable Cause	Correction
	ing	data	not exist.	third node. Check the node address of the third node in the send data for CMND(490).
		Command data	Broadcasting was specified.	Specify only one node for the third node.
	04: Destination node busy		The destination node is busy.	Increase the number of retries or review the system so that the destination node does not receive so many messages.
	05: Response timeout		The message was destroyed by noise.	Increase the number of retries or test communications between nodes to see if there is too much noise.
		Instruction control data	The response monitor time is too short.	Increase the length of the response monitor time.
		Error history	The send / receive frame was discarded.	Take appropriate measures based on the error history.
03: Controller error	01: Com- munications con- troller error	Unit / Board indic- ators	An error occurred in the communications controller.	Take appropriate measures based on the operation manuals for the related units / boards.
	02: CPU Unit error	CPU Unit indicators at destination node	A CPU error occurred in the destination CPU Unit.	Clear the error from the CPU Unit based on its operation manuals.
	03: Controller error	Board indicators	A response was not returned because an error occurred in the Board.	Check network com- munications status and restart the Board. If the problem per- sists, replace the Board.
	04: Unit number error	Unit number	The unit number was set incorrectly.	Set the rotary switches cor- rectly, being sure the unit num- bers are within range and that each number is used only once.
04: Service unsupported	01: Undefined command	Command code	The unit / board does not support the specified command code.	Check the command code.
	02: Not supported by model / version	Unit model and version	The command can- not be executed because the model or version is incor- rect.	Check the model number and version.
05: Routing table error	01: Destination address setting error	Routing table	The destination network or node address is not set in the routing tables.	Register the destination network and node in the routing tables.
	02: No routing tables	Routing table	Relaying is not possible because there are no routing tables.	Set routing tables in the source node, designation node, and relay nodes.
	03: Routing table error	Routing table	There is an error in the routing tables.	Set the routing tables correctly.
	04: Too many relays	Network con- figuration	An attempt was made to send to a	Reconstruct the networks or change the routing tables so

Main Code	Sub-Code	Check Point	Probable Cause	Correction
			network that was over 3 networks away	that commands are sent within a range of 3 networks or less.
10: Com- mand 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.
	02: Command too short	Command data	The command is shorter than the minimum permissible length.	Check the command format and correct the command data.
	03: Elements / data don't match	Command data	The designated number of elements differs from the number of write data items.	Check the number of elements and set data for each element.
	04: Command format error	Command data	An incorrect format was used.	Check the command format and correct the command data.
	05: Header error	Routing table	Either the relay table in the local node or the local net- work table in the relay node is incor- rect.	Set the routing tables correctly.
11: Para- meter error	01: Area clas- sification missing	Memory area code in command data	The specified word does not exist in the memory area or there is no EM Area.	Check the memory areas and parameter codes in the command and correct the command data.
	02: Access size error	Access size spe- cification in com- mand data	The access size specification is incorrect or an odd word address is specified.	Check the memory areas and access size and correct the access size.
	03: Address range error	Starting address in command data	The start address in command process is beyond the accessible area.	Check the area being processed and set the correct range.
	04: Address range exceeded	Starting address and number of ele- ments in command data	The end address in command process is beyond the accessible area.	Check the area being processed and set the correct range.
		Data link tables	The total number of words is beyond the limit.	Correct the data link tables.
	06: Program miss- ing	Program number in command data	FFFF Hex was not specified.	Specify FFFF Hex.
	09: Relational error	Command data	A large-small relationship in the elements in the command data is incorrect.	Check the command data and correct the relationship between the elements.
		Data link table	A node not set in the common link	Correct the data link tables.

Main Code	Sub-Code	Check Point	Probable Cause	Correction
			parameters is set as a refresh para- meter.	
	0A: Duplicate data access	I/O access in CPU Unit	Differential mon- itoring was spe- cified during data tracing or data tra- cing was specified during differential monitoring.	Cancel the current process or wait until it ends before executing the command.
		Data link tables	The same node address is specified more than once.	Correct the data link tables.
	0B: Response too long	Number of ele- ments in command data	The response format is longer than the maximum permissible length.	Check the command format and correct the number of elements.
	0C: Parameter error	Parameters in command data	There is an error in one of the parameter settings.	Check the command data and correct the parameters.
		Data link table file	There is an error in the file.	Check the contents of the file.
20: Read not possible	02: Protected		The program area is protected.	Release protection from a Programming Device and then execute the command.
	03: Table missing	Table	A table has not been registered.	Register a table.
			There is an error in the table.	Correct the table.
	04: Data missing		The search data does not exist.	
	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	A data being compared is not the same.	Check memory contents and use the correct data.
			A file read operation failed.	Check the contents of the file.
21: Write not possible	01: Read Only		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 Can- not write data link table		The program area is protected.	Release protection from a Programming Device and then execute the command.
		PLC Setup	Writing is not possible because automatic data link table	Change the PLC Setup so that the data link tables can be manually written.

Main Code	Sub-Code	Check Point	Probable Cause	Correction
			generation has been specified.	
	03: Cannot register	Number of files in file device	The file cannot be created because the limit has been exceeded.	Delete any unnecessary files or create more file memory.
		Number of files open	The maximum number of files has already been opened for the system limit.	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.	
22: Not executable in current mode	01: Not possible during execution		The mode is incorrect.	Check the mode.
		Data link status	The data link is operating.	Check the status of the data links.
	02: Not possible while running		The mode is incorrect.	Check the mode.
		Data link status	The data links are active.	Check the status of the data links.
	03: Wrong PLC mode		The PLC is in PROGRAM mode.	Check the modes of the PLC and computer.
	04: Wrong PLC mode		The PLC is in DEBUG mode.	Check the modes of the PLC and computer.
	05: Wrong PLC mode		The PLC is in MONITOR mode.	Check the modes of the PLC and computer.
	06: Wrong PLC mode		The PLC is in RUN mode.	Check the modes of the PLC and computer.
	07: Specified node not polling node		The specified node is not the polling node.	Check node functioning as the polling node for the network.
	08: Step cannot be executed		The mode is incorrect.	Check step status.
23: No such device	01: File device missing	Unit configuration	The specified memory does not exist as a file device.	Mount memory or format EM as file memory.
	02: Memory missing		There is no file memory.	Check the file memory to see if it is mounted.
	03: Clock missing		There is no clock.	Check the model.

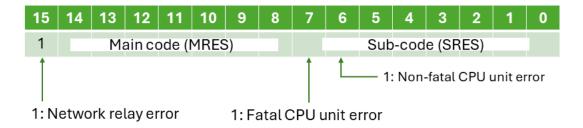
Main Code	Sub-Code	Check Point	Probable Cause	Correction
24: Cannot start / stop	01: Table missing	Data link tables	The data link tables have not been registered or they contain an error.	Set the data link tables.
25: Unit error	02: Memory error	Contents of memory being processed	The contents of memory contains an error.	Transfer the correct contents to memory.
	03: I/O setting error	I/O Unit con- figuration	The registered I/O tables do not agree with the actual I/O configuration.	Correct the I/O tables or the I/O configuration.
	04: Too many I/O points	Number of I/O in registered I/O tables	There are too many I/O points and remote I/O points registered.	Change the registered I/O table so that it is within the limit.
	05: CPU bus error	CPU bus line	An error occurred in data transfer between the CPU and a CPU Bus Unit.	Check Units, Boards, and cables to be sure they are connected correctly and then execute the ERROR CLEAR command.
	06: I/O duplication	Rack numbers, Unit numbers, and I/O addresses in PLC Setup	The same number / address was set more than once.	Check the PLC setup and correct the numbers / addresses so that each is used only once.
	07: I/O bus error	I/O bus line	An error occurred in data transfer between the CPU and an I/O unit.	Check units, boards, and cables to be sure they are connected correctly, then execute the ERROR CLEAR command.
	09: SYSMAC BUS / 2 error	SYSMAC BUS / 2 transmission path	An error occurred in data transfer on the SYSMAC BUS / 2 line.	Check units, boards, and cables to be sure they are connected correctly, then execute the ERROR CLEAR command.
	OA: CPU Bus Unit error	CPU Bus Unit trans- mission 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		Terminators have not been set.	Set the terminators correctly.
26: Com- mand error	01: No protection	Command pro- tection 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

D2: Incorrect password	Main Code	Sub-Code	Check Point	Probable Cause	Correction
word word has been specified.  1 The specified area is protected.  Number of commands being executed being executed water process to end or force the end of a current process, then execute the command.  1 The node receiving the commands.  1 The node receiving the commands.  1 The node receiving the commands.  2 Service already executed.  1 The service is being executed.  2 The service is not being executed.  2 The service is not being executed.  3 The service is not being executed.  4 A response was not returned because a buffer error occurred.  2 A response was not returned because a buffer error occurred.  3 Settings not complete  3 Settings required before execution defined  4 Check the command format and set the required elements have not been set in the command data.  4 OA: Number already defined  4 Action numbers and transition numbers of program in program area  5 DA: Error not clear  2 Cause of error  2 The specified area is protected.  5 The node receiving the eceiving the command.  6 Wait for current process to end or force the end of a current process, then execute the command.  7 The service is not being executed.  7 The required in the cause of the execute the command from a node that is participating in the data link. Execute the command from a node that is participating in the data link.  8 Restart the board.  9 Restart the board. If the problem persists, replace the board.  9 Settings required before executing the service have not been made.  1 The required elements in the command data.  2 Check the command data.  3 Check the action / transition numbers already been registered in a previous program.  9 Settings required lements in the command.					reason to clear protection.
Sprotected.   Gramming Device and then execute the command.				word has been specified.	Specify the correct password.
mands being executed the command is already processing 5 commands.  O5: Service already executing The service is being executed.  O6: Service stopped Executed The service is not being executed.  O7: No execution right Executed The service is not being executed.  O7: No execution right Executed The service is not been obtained.  O8: Settings not complete Execution Performent Execution the service has not been obtained.  O8: Settings not complete Execution Execution The settings required before execution tempts already defined Executed Execution Program in program area  O8: Number already defined Exercise The setting already been registered in a previous program.  O8: Error not clear Cause of error The cause of the Exercise Individual Salt for the execute the command is already been remote sheing executed.  Wait for the service to end or force the end of a current process, then execute the command. Wait for the service to end or force the end of a current process, then execute the command. Wait for the service to end or force the end of a current process, then execute the command. Wait for the service to end or force the end of force the end of force the end of the service and then execute the command.  Wait for the service to end or force the end of the service and then execute the command.  If necessary, start the service.  The local node is not in the data link. Execute the command from a local tall ink.  Restart the board. If the problem persists, replace the board.  Make the required settings.  Check the command format and set the required elements in the command data.  Check the command format and set the required elements in the command data.  Check the action / transition numbers and transition numbers and transition numbers and previous program.		04: Protected		-	gramming Device and then
executing    executed.   force the end of the service and then execute the command.			mands being	the command is already processing	end or force the end of a cur- rent process, then execute the
Deing executed.		-		_	force the end of the service and then execute the com-
right  Unit / Board  the service has not been obtained.  A response was not returned because a buffer error occurred.  O8: Settings not complete  Settings required before execution  O9: Necessary items not set  OA: Number already defined  OA: Number already defined  OB: Error not clear  OB: Error not clear  Cause of error  The settings required before execution been set in the command data.  The settings required before executing the service have not been set in the command data.  Check the command format and set the required elements in the command data.  Check the action / transition numbers has already been registered in a previous program.  The cause of the  Remove the cause of the error		06: Service stopped			If necessary, start the service.
returned because a buffer error occurred.  O8: Settings not complete  Settings required before execution  O9: Necessary items not set  OA: Number already defined  OA: Number already defined  OB: Error not clear  Cause of error  Returned because a buffer error occurred.  The settings required before executing the service have not been made.  The required elements have not been set in the command data.  The specified action / transition numbers and transition numbers has already been registered in a previous program.  Remove the cause of the error				the service has not	data link. Execute the com- mand from a node that is par-
complete before execution required before executing the service have not been made.  O9: Necessary items not set Command data  OA: Number already defined Action numbers and transition numbers of program in program area  OB: Error not clear Cause of error The cause of the Remove the cause of the executing the service have not been set in the command format and set the required elements in the command data.  Check the command format and set the required elements in the command data.  Check the action / transition number has already been registered in a previous program.				returned because a buffer error	lem persists, replace the
items not set  ments have not been set in the command data.  OA: Number already defined  Action numbers and transition numbers of program in program area  OB: Error not clear  Cause of error  ments have not been set in the command data.  The specified action / transition number has already been registered in a previous program.  The cause of the Remove the cause of the error				required before executing the service have not been	Make the required settings.
defined and transition number bers of program in program area program.  / transition number has already been registered in a previous program.    OB: Error not clear   Cause of error   The cause of the   Remove the cause of the error			Command data	ments have not been set in the com-	and set the required elements
			and transition num- bers of program in	/ transition number has already been registered in a pre-	numbers to ones that are not being used and then execute
removed. CLEAR.		0B: Error not clear		error has not been	and then execute ERROR
30: Access right error  The access right is held by another device. (Online editing is being executed from another node or ACCESS RIGHT ACQUIRE OR ACCESS RIGHT ACCES RIGHT ACCESS RIGHT ACCES RI		01: No access right		held by another device. (Online edit- ing is being executed from another node or ACCESS RIGHT ACQUIRE or	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
has been executed				Service was can-	

Main Code	Sub-Code	Check Point	Probable Cause	Correction
	celled		celled with CANCEL (ABORT) command.	

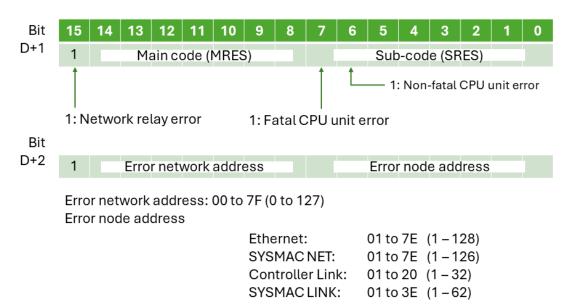
# **Flags**

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.



# Index

# Α

```
Access right denied. | Tag address = '<address>', Main error = <code>, Sub error = <code>. 75
Address Descriptions 19
Attempts Before Timeout 12
Auto-Demotion 13
Auto-Dial 8
```

# В

Baud Rate 7 BCD 18 Boolean 18

# C

C1000H Addressing 26 C2000H Addressing 30 C200H Addressing 19 C500 Addressing 23 Channel Assignment 10 Channel Properties - Advanced 9 Channel Properties - General 6 Channel Properties - Serial Communications 6 Channel Properties - Write Optimizations 8 CJ1 Addressing 63 CJ2 Addressing 67 Close Idle Connection 8 COM ID 7 COM Port 7 Communication Parameters 5, 13 Communication Protocol 5

Connect Timeout 8, 12

^ .: T 7

Connection Type 7

CS and TS Writes 13

CS1 Addressing 58

CV1000 Addressing 37

CV2000 Addressing 41

CV500 Addressing 33

CVM1-CPU11 Addressing 49

#### CVM1-CPU21 Addressing 53

# D

Data Bits 7

Data Collection 10

Data Types Description 18

Demote on Failure 13

Demotion Period 13

Device cannot accept command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 74

Device cannot process command. | Tag address = '<address>', Data size = <byte number>, Main error = <code>. Sub error = <code>. 73

Device Properties - Auto-Demotion 13

Device Properties - Redundancy 14

Device Properties - Timing 12

Device responded with a command format error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 73

Device responded with a command parameter error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 74

Device responded with a communications error. | Main error = <code>, Sub error = <code>. 73

Device responded with destination node error. | Main error = <code>, Sub error = <code>. 72

Device responded with error in unit. | Main error = <code>, Sub error = <code>. 74

Device responded with error. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 72

Device responded with fatal CPU unit error. | End code = <code>. 75

Device responded with local node error. | Main error = <code>, Sub error = <code>. 72

Device responded with non-fatal CPU unit error. | End code = <code>. 75

Device responded with read not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 74

Device responded with routing table error. | Main error = <code>, Sub error = <code>. 73

Device responded with write not possible. | Tag address = '<address>', Data size = <byte number>, Main error = <code>, Sub error = <code>. 75

Diagnostics 6

Discard Requests when Demoted 13

Do Not Scan, Demand Poll Only 11

Driver 10

Drop 8

DTR 7

Duty Cycle 9

DWord 18

#### Ε

Error Mask Definitions 76

# Event Log Messages 72

# F

Fail Write, Log Message 13
Failed to set device back to run mode following write to tag. | Tag address = '<address>'. 76
FINS communications service 15
FINS Messages 15
FINS Networks 15
FINS protocol 5
Flags 83
Float 18
Flow Control 5, 7
Framing 76

# Н

Hardware break 76 Host Link Communications 16 Host Link Interface 5

# ı

I/O error 76
ID 10
Identification 6, 10
Idle Time to Close 8
Initial Updates from Cache 12
Inter-Device Delay 10

# L

LBCD 18 Local 16 Long 18

# М

Main and Sub Error Codes 76 Model 10 Modem 7-8 Modem Settings 8

# Ν

Network 5

Non-Normalized Float Handling 9

None 7

# 0

Operating Mode 10

Operation with no Communications 8

Operational Behavior 8

Optimization Method 9

Overrun 76

Overview 5

# Р

Parity 7, 76

Physical Medium 7

Poll Delay 8

# R

Raise 8

Read Processing 8

Redundancy 14

Remote 16

Replace with Zero 9

Report Communication Errors 8

Request Size 14

Request Timeout 12

Respect Tag-Specified Scan Rate 11

Routing Tables 16

RS-485 8

RTS 7

Run Mode Settings 13

RX buffer 76

# S

Scan Mode 11

Serial Communications 6

```
Serial Port Settings 7
Set PLC to Monitor Mode, Perform Write 13
Set PLC to Monitor Mode, Write, Reset to Run 13
Setup 5
Shared 7
Short 18
Signed 18
Simulated 11
Stop Bits 7
String 18
Sysmac
      Way 5
Т
Tag Counts 6, 11
Timeouts to Demote 13
Timing 12
TX buffer 76
U
Unable to write to tag. The device is in Run Mode. | Tag address = '<address>'. 75
Unmodified 9
Unsigned 18
٧
VM1-CPU01 Addressing 46
W
Word 18
Write All Values for All Tags 9
Write Only Latest Value for All Tags 9
```

Write Only Latest Value for Non-Boolean Tags 9