Siemens S5 Driver

© 2016 PTC Inc. All Rights Reserved.

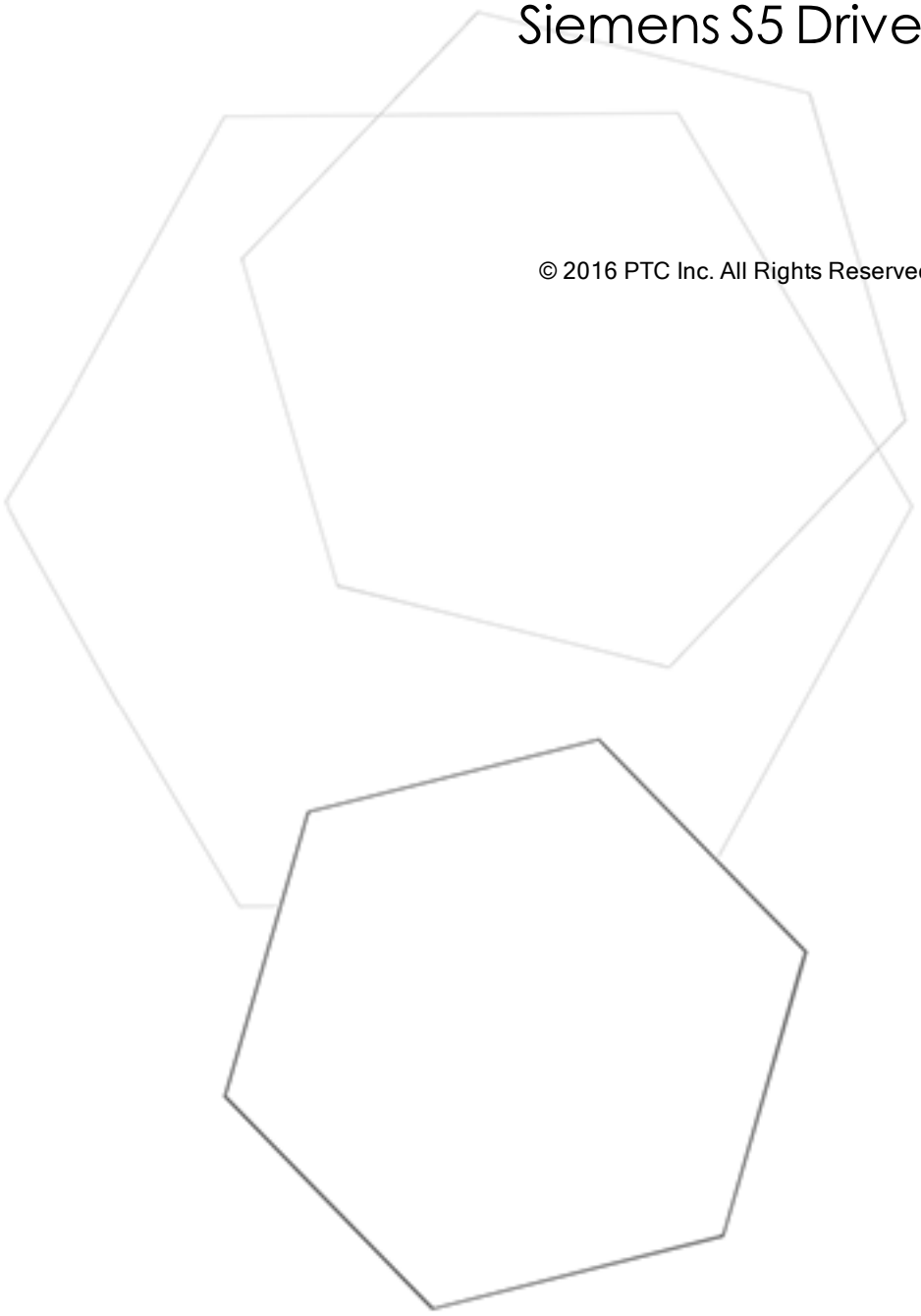


Table of Contents

Siemens S5 Driver 1
Table of Contents 2
Siemens S5 Driver 4
Overview 4
Setup 5
Channel Properties 5
Channel Properties - General 6
Channel Properties - Serial Communications 6
Channel Properties - Write Optimizations 9
Channel Properties - Advanced 9
Channel Properties - Communication Serialization 10
Driver Device Properties 11
Device Properties - Identification 12
Device Properties - Operating Mode 12
Device Properties - Scan Mode 13
Device Properties - Timing 13
Device Properties - Auto-Demotion 14
Device Properties - Redundancy 15
Data Types Description 16
Address Descriptions 17
Siemens S5 (AS511) 90U Address Descriptions 17
Siemens S5 (AS511) 95U Address Descriptions 19
Siemens S5 (AS511) 100U-100 Address Descriptions 21
Siemens S5 (AS511) 100U-101 Address Descriptions 23
Siemens S5 (AS511) 100U-103 Address Descriptions 25
Siemens S5 (AS511) 101U Address Descriptions 27
Siemens S5 (AS511) 115U-941 Address Descriptions 29
Siemens S5 (AS511) 115U-942 Address Descriptions 31
Siemens S5 (AS511) 115U-943 Address Descriptions 33
Siemens S5 (AS511) 115U-944 Address Descriptions 35
Siemens S5 (AS511) 115U-945 Address Descriptions 37
Siemens S5 (AS511) 135U-921 Address Descriptions 39
Siemens S5 (AS511) 135U-922 Address Descriptions 41
Siemens S5 (AS511) 135U-928 Address Descriptions 42
Siemens S5 (AS511) 155U-946 Address Descriptions 44
Siemens S5 (AS511) 155U-947 Address Descriptions 46
Event Log Messages 49
Failure reading device configuration. 49
Protocol error. Incorrect number of bytes received. | Received = <number> (bytes), Expected = <number> (bytes). 49

Requested data block is not defined and has been disabled. | Data block = 'DB<block number>'. 49
Data block is not defined. Write operation has failed. | Data block = 'DB<block number>'. 50
Error Mask Definitions 50
Index 51

Siemens S5 Driver

Help version 1.026

Overview
What is the Siemens S5 Driver?

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

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Siemens S5 Driver?

Event Log Messages
What messages does the Siemens S5 Driver produce?

Overview

The Siemens S5 Driver provides a reliable way to connect Siemens S5 (AS511) devices to client applications; including HMI, SCADA, Historian, MES, ERP, and countless custom applications. It is intended for use with Siemens S5 PLCs communicating via the front programming port using AS511 protocol (which is specific for each Siemens device). This driver has been designed to operate with a set range of Siemens equipment: it is not recommended for use on devices that are not supported.

The Siemens S5 PLC family has a unique memory structure. Data within the PLC is not at fixed locations within the PLC's memory space. As the PLC logic is created and modified, this memory space is continuously updated and revised. When these revisions occur, the location of the key data elements (such as flags, timers, counters, I/O, and data blocks) can move around in the PLC's memory. The Siemens S5 Driver has been designed to read the location of these memory elements when the driver begins operation or detects a communications error. If the PLC configuration changes, users must restart the Siemens S5 Driver or pull and replace the cable connection. Both of these actions cause the driver to reacquire the location of all PLC memory elements.

Setup

Supported Devices

Siemens S5-90U
Siemens S5-95U
Siemens S5-100U-100
Siemens S5-100U-101
Siemens S5-100U-103
Siemens S5-101U
Siemens S5-115U-941
Siemens S5-115U-942
Siemens S5-115U-943
Siemens S5-115U-944
Siemens S5-115U-945
Siemens S5-135U-921
Siemens S5-135U-922
Siemens S5-135U-928
Siemens S5-155U-946
Siemens S5-155U-947

Communication Protocol

AS511 Current Loop

Supported Communication Properties

Baud: 9600 (Fixed)
Parity: Even (Fixed)
Data Bits: 8 (Fixed)
Stop Bit: 1 (Fixed)

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows communications with serial devices attached to an Ethernet network using a terminal server or device server. It may be invoked through COM ID in Channel Properties. When used directly with a serial port, this driver only supports a single connection to a single controller per serial port. When operating in Ethernet Encapsulation Mode, the driver supports up to 30 controllers per channel. In this mode, a single controller can be paired with a terminal server/device server to form a single node. For more information, refer to the server's help documentation.

Note: The Siemens S5 AS511 protocol is sensitive to timing and gaps in the communications stream. If the network experiences heavy packet loss or delay while using Ethernet Encapsulation, the Siemens S5 Driver may report a large number of timeout errors or be unable to communicate. In some cases, using a switched network can help reduce these delays; however, it is not a guaranteed solution.

Modem Setup

This driver supports modem functionality. For more information, refer to Modem Support in the server help documentation.

See Also:
Channel Properties
Device Properties

Channel Properties

This server supports the use of simultaneous multiple 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.

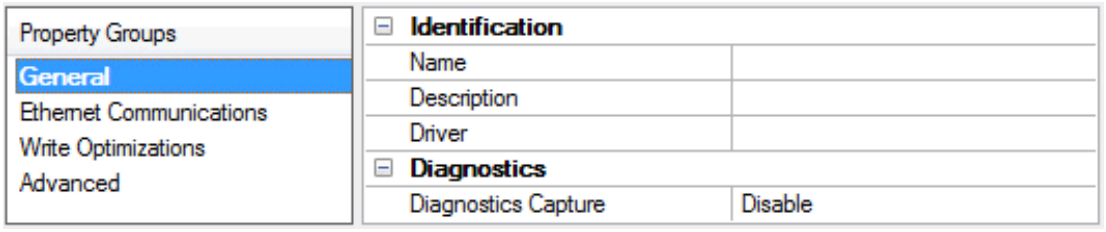
The properties associated with a channel are broken in to logical groupings. While some groups are specific to a given driver or protocol, the following are the common groups:

General

Ethernet or Serial Communications
Write Optimization
Advanced

Channel Properties - General

This server supports the use of simultaneous multiple 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: 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.
For information on reserved characters, refer to "How To... Properly Name a Channel, Device, Tag, and Tag Group" in the server help.

Description: User-defined information about this channel.
Many of these properties, including Description, have an associated system tag.

Driver: Selected protocol / driver for this channel. This property specifies the device driver that was selected during channel creation. It is a disabled setting in the channel properties.

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. With this in mind, changes to the properties should not be made once a large client application has been developed. Utilize the User Manager to prevent operators from changing properties and restrict access rights to server features.

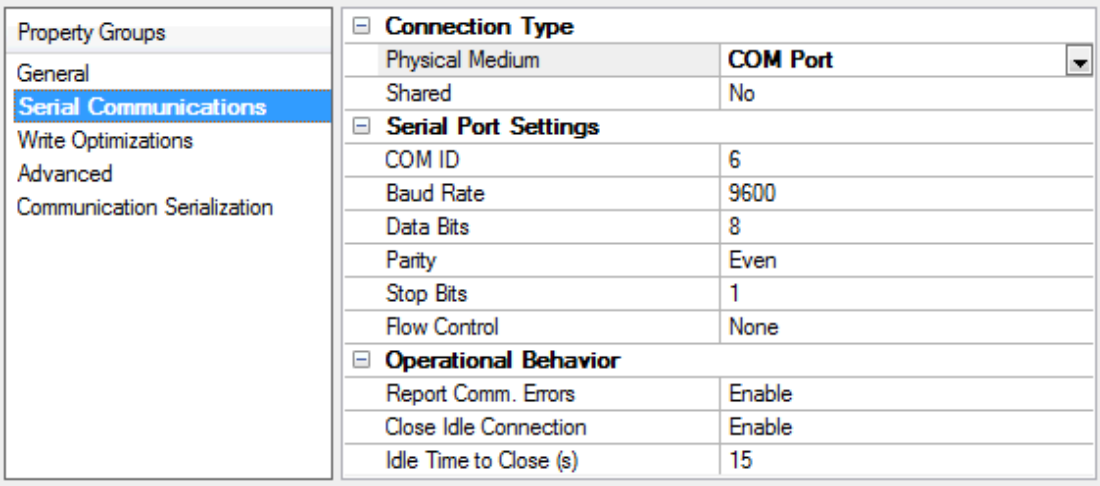
Diagnostics

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.
For more information, refer to "Communication Diagnostics" in the server help.
Not all drivers support diagnostics. To determine whether diagnostics are available for a particular driver, open the driver information and locate the "Supports device level diagnostics" statement.

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: Connection Type, Serial Port Settings or Ethernet Settings, and Operational Behavior.

Note: With the server's online full-time operation, these properties can be changed at any time. Utilize the User Manager to restrict access rights to server features, as changes made to these properties can temporarily disrupt communications.



Connection Type

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

- **None**: Select None to indicate there is no physical connection, which displays the Operation with no Communications section.
- **COM Port**: Select Com Port to display and configure the Serial Port Settings section.
- **Modem**: Select Modem if phone lines are used for communications, which are configured in the Modem Settings section.
- **Ethernet Encap.**: Select if Ethernet Encapsulation is used for communications, which displays the Ethernet Settings section.
- **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**: This property specifies 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**: This property specifies 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**: This property specifies 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 Comm. 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.

Ethernet Settings

Ethernet Encapsulation provides communication with serial devices connected to terminal servers on the Ethernet network. A terminal server is essentially a virtual serial port that converts TCP/IP messages on the Ethernet network to serial data. Once the message has been converted, users can connect standard devices that support serial communications to the terminal server. The terminal server's serial port must be properly configured to match the requirements of the serial device to which it is attached. *For more information, refer to "How To... Use Ethernet Encapsulation" in the server help.*

- **Network Adapter**: Indicate a network adapter to bind for Ethernet devices in this channel. Choose a network adapter to bind to or allow the OS to select the default.
  Specific drivers may display additional Ethernet Encapsulation properties. For more information, refer to Channel Properties - Ethernet Encapsulation.

Modem Settings

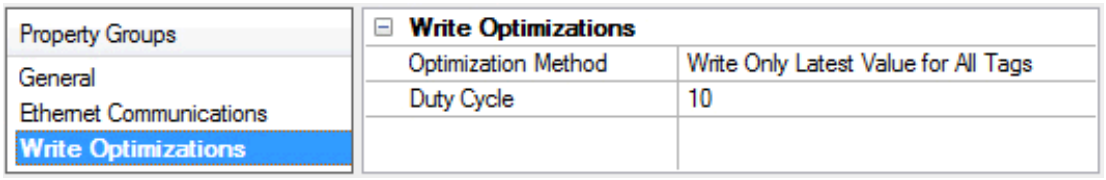
- **Modem**: Specify the installed modem to be used for communications.
- **Connect Timeout**: Specify the amount of time to wait for connections to be established before failing a read or write. The default is 60 seconds.
- **Modem Properties**: Configure the modem hardware. When clicked, it opens vendor-specific 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 Comm. 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

As with any OPC server, writing data to the device may be the application's most important aspect. The server intends to ensure that the data written from the client application gets to the device on time. Given this goal, the server provides optimization properties that can be used 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:

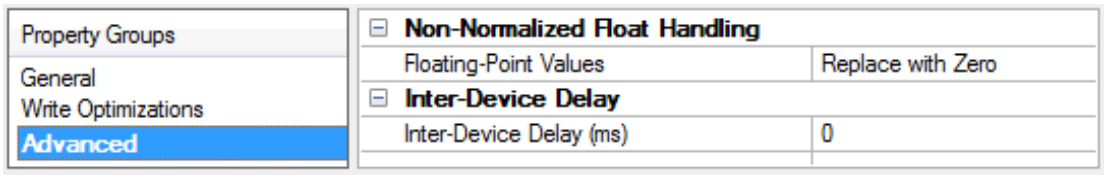
- **Write All Values for All Tags**: This option forces the server to attempt to write every value to the controller. In this mode, the server continues to gather write requests and add them to the server's internal write queue. The server processes the write queue and attempts to empty it by writing 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: Non-normalized float handling allows users to specify how a driver handles non-normalized IEEE-754 floating point data. 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. 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.lin

For more information on the floating point values, refer to "How To ... Work with Non-Normalized Floating Point Values" in the server help.

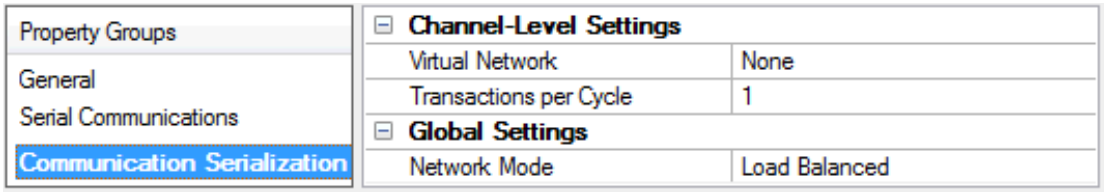
Inter-Device Delay: Specify the amount of time the communications channel waits to send new requests to the next device after data is received from the current device on the same channel. Zero (0) disables the delay.

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

Channel Properties - Communication Serialization

The server's multi-threading architecture allows channels to communicate with devices in parallel. Although this is efficient, communication can be serialized in cases with physical network restrictions (such as Ethernet radios). Communication serialization limits communication to one channel at a time within a virtual network.

The term "virtual network" describes a collection of channels and associated devices that use the same pipeline for communications. For example, the pipeline of an Ethernet radio is the master radio. All channels using the same master radio associate with the same virtual network. Channels are allowed to communicate each in turn, in a “round-robin” manner. By default, a channel can process one transaction before handing communications off to another channel. A transaction can include one or more tags. If the controlling channel contains a device that is not responding to a request, the channel cannot release control until the transaction times out. This results in data update delays for the other channels in the virtual network.



Channel-Level Settings

Virtual Network This property specifies the channel's mode of communication serialization. Options include None and Network 1 - Network 50. The default is None. Descriptions of the options are as follows:

- **None**: This option disables communication serialization for the channel.
- **Network 1 - Network 50**: This option specifies the virtual network to which the channel is assigned.

Transactions per Cycle This property specifies the number of single blocked/non-blocked read/write transactions that can occur on the channel. When a channel is given the opportunity to communicate, this number of transactions attempted. The valid range is 1 to 99. The default is 1.

Global Settings

- **Network Mode**: This property is used to control how channel communication is delegated. In **Load Balanced** mode, each channel is given the opportunity to communicate in turn, one at a time. In **Priority** mode, channels are given the opportunity to communicate according to the following rules (highest to lowest priority):
  - Channels with pending writes have the highest priority.
  - Channels with pending explicit reads (through internal plug-ins or external client interfaces) are prioritized based on the read's priority.
  - Scanned reads and other periodic events (driver specific).

  The default is Load Balanced and affects *all* virtual networks and channels.

Devices that rely on unsolicited responses should not be placed in a virtual network. In situations where communications must be serialized, it is recommended that Auto-Demotion be enabled.

Due to differences in the way that drivers read and write data (such as in single, blocked, or non-blocked transactions); the application's Transactions per cycle property may need to be adjusted. When doing so, consider the following factors:

- How many tags must be read from each channel?
- How often is data written to each channel?
- Is the channel using a serial or Ethernet driver?
- Does the driver read tags in separate requests, or are multiple tags read in a block?
- Have the device's Timing properties (such as Request timeout and Fail after *x* successive timeouts) been optimized for the virtual network's communication medium?

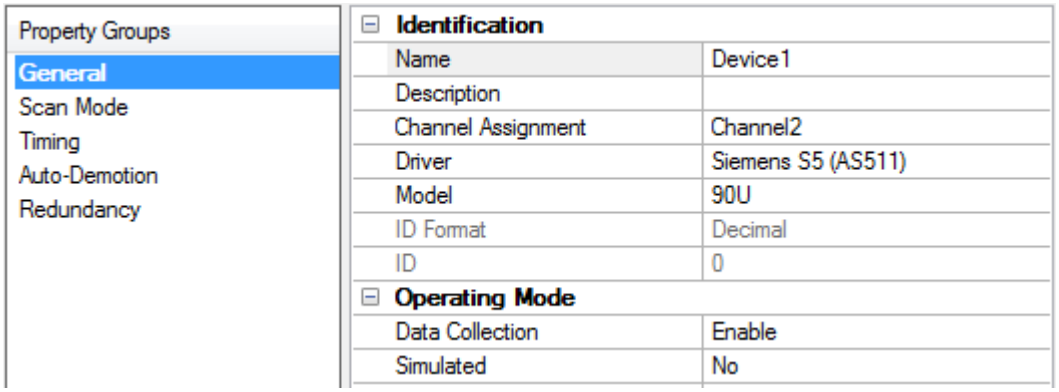
Driver Device Properties

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

Device properties are organized into the following groups. Click on a link below for details about the settings in that group.

Identification
Operating Mode
Scan Mode
Communication Timeouts
Auto-Demotion
Redundancy

Device Properties - 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.

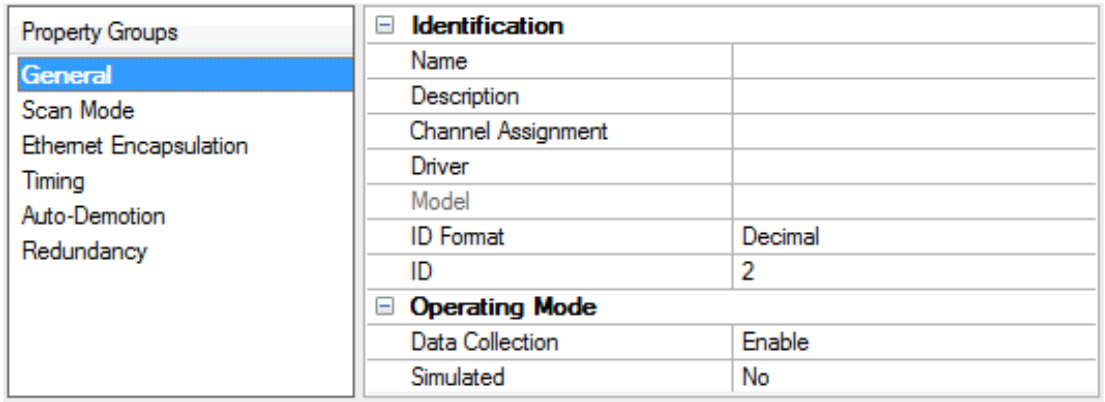
Driver: Selected protocol driver for this device.

Model: The specific version of the device, the C64 CNC Controller.

ID Format: Select how the device identity is formatted. Options include Decimal, Octal, and Hex.

ID: the unique identity of the device for communication with the driver.

Device Properties - 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: This option places the device into 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.

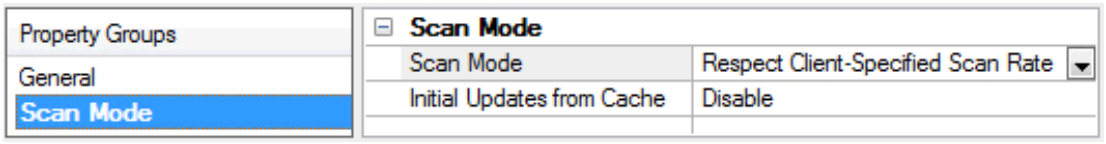
Notes:

1. This 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.
2. 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.

Simulation Mode is for test and simulation purposes only. It should never be used in a production environment.

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.



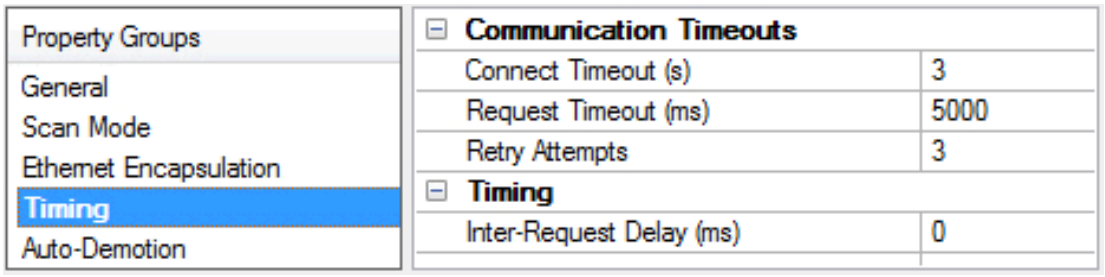
Scan Mode: specifies how tags in the device are scanned for updates sent to subscribed clients. Descriptions of the options are:

- **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 maximum scan rate to be used. 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 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 Communications Timeouts 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. Communications Timeouts properties are specific to each configured device.



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: This property specifies 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 9,999,999 milliseconds (167.6667 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.

Retry Attempts: This property specifies how many times the driver retries 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 retries configured for an application depends largely on the communications environment.

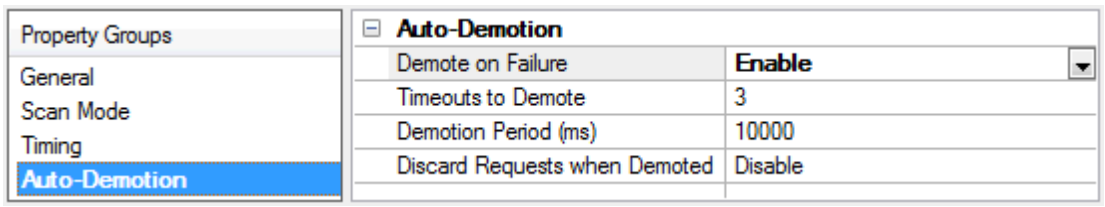
Timing

Inter-Request Delay: This property specifies how long the driver waits before sending the next request to the target device. 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 turnaround 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 supported by the driver.

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.



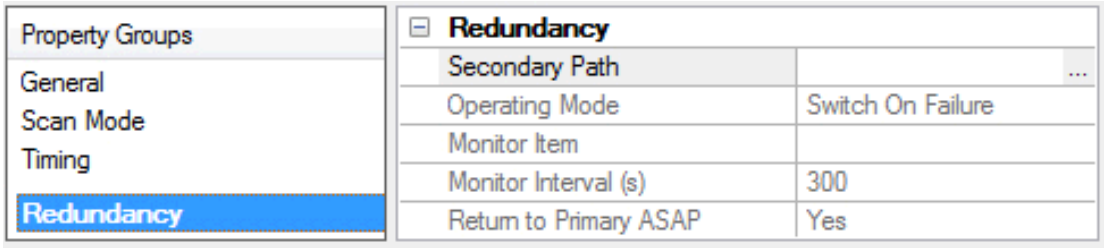
Demote on Failure: When enabled, the device is automatically taken off-scan until it is responding again.
Tip: Determine when a device is off-scan by monitoring its demoted state using the _AutoDemoted system tag.

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 - Redundancy



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

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

Data Types Description

Data Type	Description
Boolean	Single bit of an 8-bit value*
Byte	Unsigned 8-bit value
Word	Unsigned 16-bit value
Short	Signed 16-bit value
DWord	Unsigned 32-bit value
Long	Signed 32-bit value
Float	32-bit floating point value The driver interprets two consecutive registers as a floating-point value by making the second register the high word and the first register the low word.
String	Null-terminated ASCII string Includes Hi-Lo or Lo-Hi byte order selection.

**For more information, refer to Address Descriptions.*

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.

Siemens S5 (AS511) 90U
Siemens S5 (AS511) 95U
Siemens S5 (AS511) 100U-100
Siemens S5 (AS511) 100U-101
Siemens S5 (AS511) 100U-103
Siemens S5 (AS511) 101U
Siemens S5 (AS511) 115U-941
Siemens S5 (AS511) 115U-942
Siemens S5 (AS511) 115U-943
Siemens S5 (AS511) 115U-944
Siemens S5 (AS511) 115U-945
Siemens S5 (AS511) 135U-921
Siemens S5 (AS511) 135U-922
Siemens S5 (AS511) 135U-928
Siemens S5 (AS511) 155U-946
Siemens S5 (AS511) 155U-947

Siemens S5 (AS511) 90U Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write

Address Type	Range	Type	Access
Note: Q and A access the same memory area	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10

- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 95U Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b *	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write

Address Type	Range	Type	Access
Note: Q and A access the same memory area.	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory Note: F and M access the same memory area.	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 100U-100 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126		Read/Write

Address Type	Range	Type	Access
	QD0-QD124	**Word**, Short **DWord**, Long	Read/Write
Discrete Outputs **Note:** Q and A access the same memory area.	A0.b-A127.b* AB0-AB127 AW0-AW126 AD0-AD124	**Boolean** **Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write Read/Write
Internal Memory	F0.b-F255.b* FB0-FB255 FW0-FW254 FD0-FD252	**Boolean** **Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write Read/Write
Internal Memory **Note:** F and M access the same memory area.	M0.b-M255.b* MB0-MB255 MW0-MW254 MD0-MD252	**Boolean** **Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30

- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers
The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters
Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 100U-101 Address Descriptions
The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**,	Read/Write

Address Type	Range	Type	Access
		Long	
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b *	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10

- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8, ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 100U-103 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write

Address Type	Range	Type	Access
Note: Q and A access the same memory area.	AB0-AB127 AW0-AW126 AD0-AD124	**Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write
Internal Memory	F0.b-F255.b* FB0-FB255 FW0-FW254 FD0-FD252	**Boolean** **Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write Read/Write
Internal Memory Note: F and M access the same memory area.	M0.b-M255.b* MB0-MB255 MW0-MW254 MD0-MD252	**Boolean** **Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.-KS255.IH*** DB1-N:KS0.-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 101U Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write

Address Type	Range	Type	Access
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory

types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers
The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters
Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 115U-941 Address Descriptions
The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write

Address Type	Range	Type	Access
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15.	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers
The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters
Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 115U-942 Address Descriptions
Default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write

Address Type	Range	Type	Access
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the

values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters
Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 115U-943 Address Descriptions
Default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**,	Read/Write

Address Type	Range	Type	Access
		Long	
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters
Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 115U-944 Address Descriptions
The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write

Address Type	Range	Type	Access
Note: F and M access the same memory area.	MB0-MB255 MW0-MW254 MD0-MD252	**Byte** **Word**, Short **DWord**, Long	Read/Write Read/Write Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 115U-945 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I127.b*	**Boolean**	Read/Write
	IB0-IB127	**Byte**	Read/Write
	IW0-IW126	**Word**, Short	Read/Write
	ID0-ID124	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E127.b*	**Boolean**	Read/Write
	EB0-EB127	**Byte**	Read/Write
	EW0-EW126	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED124	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q127.b*	**Boolean**	Read/Write
	QB0-QB127	**Byte**	Read/Write
	QW0-QW126	**Word**, Short	Read/Write
	QD0-QD124	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A127.b*	**Boolean**	Read/Write
	AB0-AB127	**Byte**	Read/Write
	AW0-AW126	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD124	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write

Address Type	Range	Type	Access
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Right Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used

to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 135U-921 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I511.b*	**Boolean**	Read/Write
	IB0-IB511	**Byte**	Read/Write
	IW0-IW510	**Word**, Short	Read/Write
	ID0-ID508	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E511.b*	**Boolean**	Read/Write
	EB0-EB511	**Byte**	Read/Write
	EW0-EW510	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED508	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q511.b*	**Boolean**	Read/Write
	QB0-QB511	**Byte**	Read/Write
	QW0-QW510	**Word**, Short	Read/Write
	QD0-QD508	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A511.b*	**Boolean**	Read/Write
	AB0-AB511	**Byte**	Read/Write
	AW0-AW510	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD508	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write

Address Type	Range	Type	Access
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH*** DB1-N:KS0.l-KS255.lL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *l* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 135U-922 Address Descriptions

Default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I511.b*	**Boolean**	Read/Write
	IB0-IB511	**Byte**	Read/Write
	IW0-IW510	**Word**, Short	Read/Write
	ID0-ID508	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E511.b*	**Boolean**	Read/Write
	EB0-EB511	**Byte**	Read/Write
	EW0-EW510	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED508	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q511.b*	**Boolean**	Read/Write
	QB0-QB511	**Byte**	Read/Write
	QW0-QW510	**Word**, Short	Read/Write
	QD0-QD508	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A511.b*	**Boolean**	Read/Write
	AB0-AB511	**Byte**	Read/Write
	AW0-AW510	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD508	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write

Address Type	Range	Type	Access
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T127	**Long**	Read/Write
Counter Current Values	C0-C127	**Word**, Short	Read/Write
Counter Current Values	Z0-Z127	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 135U-928 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I511.b*	**Boolean**	Read/Write
	IB0-IB511	**Byte**	Read/Write
	IW0-IW510	**Word**, Short	Read/Write
	ID0-ID508	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E511.b*	**Boolean**	Read/Write
	EB0-EB511	**Byte**	Read/Write
	EW0-EW510	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED508	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q511.b*	**Boolean**	Read/Write
	QB0-QB511	**Byte**	Read/Write
	QW0-QW510	**Word**, Short	Read/Write
	QD0-QD508	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A511.b*	**Boolean**	Read/Write
	AB0-AB511	**Byte**	Read/Write
	AW0-AW510	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD508	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write

Address Type	Range	Type	Access
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T255	**Long**	Read/Write
Counter Current Values	C0-C255	**Word**, Short	Read/Write
Counter Current Values	Z0-Z255	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note: Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 155U-946 Address Descriptions

Default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I511.b*	**Boolean**	Read/Write
	IB0-IB511	**Byte**	Read/Write
	IW0-IW510	**Word**, Short	Read/Write
	ID0-ID508	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E511.b*	**Boolean**	Read/Write
	EB0-EB511	**Byte**	Read/Write
	EW0-EW510	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED508	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q511.b*	**Boolean**	Read/Write
	QB0-QB511	**Byte**	Read/Write
	QW0-QW510	**Word**, Short	Read/Write
	QD0-QD508	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A511.b*	**Boolean**	Read/Write
	AB0-AB511	**Byte**	Read/Write
	AW0-AW510	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD508	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write

Address Type	Range	Type	Access
Data Block String	DB1-N:KS0.I-KS255.IH*** DB1-N:KS0.I-KS255.IL***	**String**	Read/Write
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T255	**Long**	Read/Write
Counter Current Values	C0-C255	**Word**, Short	Read/Write
Counter Current Values	Z0-Z255	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note:Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) is added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Siemens S5 (AS511) 155U-947 Address Descriptions

The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I0.b-I511.b*	**Boolean**	Read/Write

Address Type	Range	Type	Access
	IB0-IB511	**Byte**	Read/Write
	IW0-IW510	**Word**, Short	Read/Write
	ID0-ID508	**DWord**, Long	Read/Write
Discrete Inputs	E0.b-E511.b*	**Boolean**	Read/Write
	EB0-EB511	**Byte**	Read/Write
	EW0-EW510	**Word**, Short	Read/Write
Note: I and E access the same memory area.	ED0-ED508	**DWord**, Long	Read/Write
Discrete Outputs	Q0.b-Q511.b*	**Boolean**	Read/Write
	QB0-QB511	**Byte**	Read/Write
	QW0-QW510	**Word**, Short	Read/Write
	QD0-QD508	**DWord**, Long	Read/Write
Discrete Outputs	A0.b-A511.b*	**Boolean**	Read/Write
	AB0-AB511	**Byte**	Read/Write
	AW0-AW510	**Word**, Short	Read/Write
Note: Q and A access the same memory area.	AD0-AD508	**DWord**, Long	Read/Write
Internal Memory	F0.b-F255.b*	**Boolean**	Read/Write
	FB0-FB255	**Byte**	Read/Write
	FW0-FW254	**Word**, Short	Read/Write
	FD0-FD252	**DWord**, Long	Read/Write
Internal Memory	M0.b-M255.b*	**Boolean**	Read/Write
	MB0-MB255	**Byte**	Read/Write
	MW0-MW254	**Word**, Short	Read/Write
Note: F and M access the same memory area.	MD0-MD252	**DWord**, Long	Read/Write
Data Block Boolean	DB1-N:KM0.b-KM255.b** .b is Bit Number 0-15	**Boolean**	Read/Write
Data Block Left Byte	DB1-N:KL0-KL255**	**Byte**	Read/Write
Data Block Right Byte	DB1-N:KR0-KR255**	**Byte**	Read/Write
Data Block Unsigned Word	DB1-N:KH0-KH255**	**Word**, Short	Read/Write
Data Block Signed Word	DB1-N:KF0-KF255**	**Short**, Word	Read/Write
Data Block Signed Long	DB1-N:KD0-KD254**	**Long**, DWord	Read/Write
Data Block Float	DB1-N:KG0-KG254**	**Float**	Read/Write
Data Block String	DB1-N:KS0.l-KS255.lH***	**String**	Read/Write

Address Type	Range	Type	Access
	DB1-N:KS0.I-KS255.IL***		
Data Block Timer	DB1-N:KT0-KT255**	**Long**	Read/Write
Data Block Counter	DB1-N:KC0-KC255**	**Word**, Short	Read/Write
Timer Current Values	T0-T255	**Long**	Read/Write
Counter Current Values	C0-C255	**Word**, Short	Read/Write
Counter Current Values	Z0-Z255	**Word**, Short	Read/Write

*.*b* specifies the bit number, and may range from 0 to 7.
** *1-N* specifies the block number.
*** *1-N* specifies the block number. *I* specifies the string length, and may range from 2 to 254. *H* specifies the high byte order; *L* specifies the low byte order. *H* is assumed if no byte order is specified.

Note: All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

Examples

- To access bit 3 of Internal Memory F20, declare an address as follows: F20.3
- To access Data Block 5 as word memory at element 30, declare an address as follows: DB5:KH30
- To access Data Block 2 element 20 and bit 7, declare an address as follows: DB2:KM20.7
- To access Data Block 1 as left byte memory at element 10, declare an address as follows: DB1:KL10
- To access Internal Memory F20 as a DWORD, declare an address as follows: FD20
- To access Input Memory I10 as a Word, declare an address as follows: IW10

Note:Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 modifies the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use FD0, FD4, FD8 ... and so on to prevent overlapping bytes.

Timers

The Siemens S5 Driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type is scaled using the appropriate Siemens time base. As a result, the values are always returned as a count of milliseconds. When writing to T or KT memory types, the Siemens time base is applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

Counters

Counters are stored as three BCD digits on the device. The largest value that can be read or written to a counter is 999.

Strings

String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) are used to store the string data. When writing strings shorter than the maximum specified length (5 in this example), a null terminator (0x00) are added to the end of the string. When strings are read, the full range of registers are read (3 in this example). Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. Appending either an "H" or "L" to the address specifies the byte order.

Event Log Messages

The following information concerns messages posted to the Event Log pane in the main user interface. Consult the 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.

Failure reading device configuration.

Error Type:
Warning

Possible Cause:

1. A device configuration transaction timed out because the serial connection between the device and the host PC is invalid.
2. A device configuration transaction timed out because the communications parameters for the serial port connection are incorrect.

Possible Solution:

1. Verify the cabling between the PC and the device is connected and intact.
2. Verify that the correct baud rate and parity are specified for the device.

Protocol error. Incorrect number of bytes received. | Received = <number> (bytes), Expected = <number> (bytes).

Error Type:
Warning

Possible Cause:

1. Misalignment of packets due to disconnection between PC and device.
2. There is a problem with the cabling connecting the devices, causing noise.

Possible Solution:
While the driver can recover from this error without intervention, check the cabling or the device itself.

Requested data block is not defined and has been disabled. | Data block = 'DB<block number>'.

Error Type:
Warning

Possible Cause:
An attempt was made to reference a non-existent location in the specified device.

Possible Solution:
Verify that the tags assigned to addresses in the specified range on the device and eliminate any that reference invalid locations.

Data block is not defined. Write operation has failed. | Data block = 'DB<block number>'.

Error Type:
Warning

Possible Cause:
An attempt was made to write to a non-existent location in the specified device.

Possible Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate any that reference invalid locations.

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

Index

A

Address Descriptions 17
Advanced Channel Properties 9
AS511 protocol 4
Auto Dial 8

B

Baud 5
Baud Rate 7
Boolean 16
Byte 16

C

Channel Assignment 12
Channel Properties 5
Channel Properties - General 6
Channel Properties - Write Optimizations 9
Character buffer overrun 50
Close Idle Connection 8
COM ID 7
Communication Serialization 10
Communications Timeouts 13-14
Connect Timeout 14
Connection Type 7

D

Data Bits 7
Data block is not defined. Write operation has failed. | Data block = 'DB<block number>'. 50
Data Collection 12
Data Types Description 16
Demote on Failure 15
Demotion Period 15
Device Properties 11
Device Properties - Auto-Demotion 14
Diagnostics 6
Discard Requests when Demoted 15

Do Not Scan, Demand Poll Only 13
Driver 6, 12
Duty Cycle 9
DWord 16

E

Error Mask Definitions 50
Ethernet Encapsulation 5
Event Log Messages 49

F

Failure reading device configuration. 49
Float 16
Flow Control 7
Framing 50

G

Global Settings 11

H

Hardware 50

I

ID 12
ID Format 12
Identification 12
Idle Time to Close 8
IEEE-754 floating point 10
Initial Updates from Cache 13
Inter-Request Delay 14

L

Load Balanced 11
Long 16

M

Model 12
Modem 8
Modem Setup 5

N

Network 5
Network Adapter 8
Network Mode 11
Non-Normalized Float Handling 10

O

Operating Mode 12
Operational Behavior 8
Optimization Method 9
Overview 4

P

Parity 5, 7
Physical Medium 7
Priority 11
Protocol error. Incorrect number of bytes received. | Received = <number> (bytes), Expected = <number> (bytes). 49

R

Read Processing 9
Redundancy 15
Report Comm. Errors 8
Request All Data at Scan Rate 13
Request Data No Faster than Scan Rate 13
Request Timeout 14
Requested data block is not defined and has been disabled. | Data block = 'DB<block number>'. 49
Respect Client-Specified Scan Rate 13
Respect Tag-Specified Scan Rate 13
Retry Attempts 14
RX buffer overrun 50

S

S5 PLC 4
Scan Mode 13
Serial Communications 6
Serial Port Settings 7
Setup 5
Short 16
Siemens S5 (AS511) 100U-100 Address Descriptions 21
Siemens S5 (AS511) 100U-101 Address Descriptions 23
Siemens S5 (AS511) 100U-103 Address Descriptions 25
Siemens S5 (AS511) 101U Address Descriptions 27
Siemens S5 (AS511) 115U-941 Address Descriptions 29
Siemens S5 (AS511) 115U-942 Address Descriptions 31
Siemens S5 (AS511) 115U-943 Address Descriptions 33
Siemens S5 (AS511) 115U-944 Address Descriptions 35
Siemens S5 (AS511) 115U - 945 Address Descriptions 37
Siemens S5 (AS511) 135U-921 Address Descriptions 39
Siemens S5 (AS511) 135U-922 Address Descriptions 41
Siemens S5 (AS511) 135U-928 Address Descriptions 42
Siemens S5 (AS511) 155U-947 Address Descriptions 46
Siemens S5 (AS511) 155U - 946 Address Descriptions 44
Siemens S5 (AS511) 90U Address Descriptions 17
Siemens S5 (AS511) 95U Address Descriptions 19
Siemens S5 PLCs 4
Simulated 12
Stop Bits 7
String 16
Supported Devices 5

T

Timeouts to Demote 15
Transactions 11
TX buffer full 50

V

Virtual Network 10

W

Word 16
Write All Values for All Tags 9
Write Only Latest Value for All Tags 9
Write Only Latest Value for Non-Boolean Tags 9
Write Optimizations 9