Siemens S7-200 Driver Help

Table of Contents

Siemens S7-200 Driver Help

Help version 1.016

CONTENTS

Overview

The Siemens S7-200 Driver provides an easy and reliable way to connect Siemens S7-200 devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Siemens S7-200 devices. This driver also supports the use of the 10 bit or 11 bit setting of the PPI programming cable. In 11 bit mode, the normal S7-200 Model should be selected. If using the 10 bit mode (specifically, the EM 241 Modem Module is required), then the S7-200 PPM mode should be used.

Device Setup

Supported Devices

Siemens S7-200 devices

Supported Cables

A special cable is required in order to communicate with the S7-200 PLC. Please use the cable recommended by the manufacturer.

Communication Protocol

Point-to-Point (PPI) S7-200 Communications Protocol (11 bit Mode)
Point-to-Point Modem (PPM) S7-200 Communications Protocol (10 bit Mode)

The driver normally operates using the standard 11 bit PPI protocol. If the use of the EM 241 modem module is required, the S7-200 PPM model must be selected. The S7-200 PPM model allows the driver to operate in a 10 bit mode (which is more compatible with a wide range of off-the-shelf modems). The 10 bit PPM mode can also be used directly on the programming port of the PLC. To enable 10 bit PPM mode, the S7-200 programming cable must be set to 10 bit mode.

Supported Communication Parameters*

Baud Rate: 9600 or 19200
Parity: Even (11 Bit Mode) None (10 Bit PPM Mode)
Data Bits: 8
Stop Bits: 1

*Not all devices support the listed configurations.

Maximum Number of Channels and Devices

The maximum number of channels and devices supported by the Siemens S7-200 Driver are as follows:
256 channels
127 devices

Ethernet Encapsulation

This driver supports Ethernet Encapsulation, which allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server or device server. Ethernet Encapsulation mode is invoked by selecting it from the COM ID dialog on the Channel Properties page. For more information, refer to the main OPC Server's help documentation.

Master ID**

Valid Master IDs are 0-126. This is the node number used by the Siemens S7-200 driver on the network.

Device IDs**

Valid Device IDs are 0-126.

**Each channel should have a unique Master ID. Any devices defined under this channel should not use a Device ID that conflicts with the Master ID.

Flow Control

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

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

Modem Setup

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

Data Types Description

Data Type	Description
Boolean	Single bit of a 16 bit value.*
Byte	Unsigned 8 bit value. bit 0 is the low bit bit 7 is the high bit
Word	Unsigned 16 bit value. bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value. bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value. bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value. bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
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

*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.

S7-200 Addressing
S7-200 PPM Addressing

S7-200 Addressing

The S7-200 addressing format is the same as the S7-200 PPM addressing format. The model selection in this case determines whether the driver is using PPI protocol (normal S7-200 Mode) or PPM (S7-200 in Point to Point Modem) mode. In both cases, the addressing is the same.

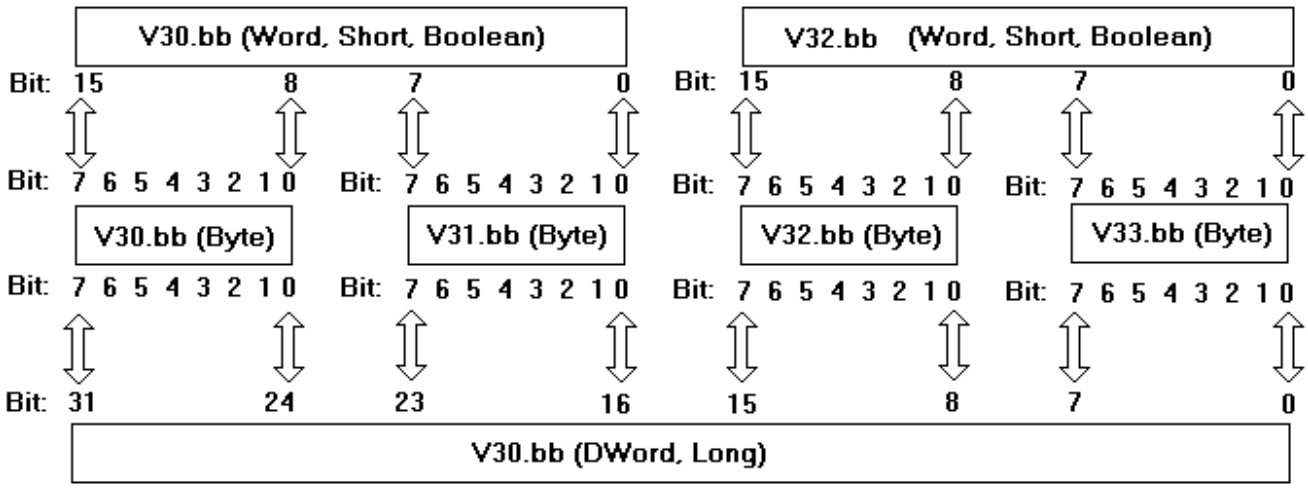
The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I00000-I65535 I00000-I65534 I00000-I65532 I00000.bb-I65535.bb* I00000.bb-I65534.bb* I00000.bb-I65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Discrete Outputs	Q00000-Q65535 Q00000-Q65534 Q00000-Q65532 Q00000.bb-Q65535.bb* Q00000.bb-Q65534.bb* Q00000.bb-Q65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Internal Memory	M00000-M65535 M00000-M65534 M00000-M65532 M00000.bb-M65535.bb* M00000.bb-M65534.bb* M00000.bb-M65532.bb*	Byte, **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Special Memory	SM00000-SM65535 SM00000-SM65534 SM00000-SM65532 SM00000.bb-SM65535.bb* SM00000.bb-SM65534.bb* SM00000.bb-SM65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write SM0-SM29 are Read Only
Variable Memory	V00000-V65535 V00000-V65534 V00000-V65532 V00000.bb-V65535.bb* V00000.bb-V65534.bb* V00000.bb-V65532.bb*	Byte, **Word**, Short DWord, Long, Float, String **Byte** Boolean, **Word**, Short **DWord**, Long, String	Read/Write
Timer Current Values	T00000-T65535	**DWord**, Long	Read/Write
Timer Status Bits	T00000-T65535	Boolean**	Read Only
Counter Current Values	C00000-C65535	**Word**, Short	Read/Write

Counter Status Bits	C00000-C65535	Boolean**	Read Only
High Speed Counters	HC00000-HC65535	**DWord**, Long	Read Only
Analog Inputs	AI00000-AI65534***	**Word**, Short	Read Only
Analog Outputs	AQ00000-AQ65534***	**Word**, Short	Write Only

*For Byte, Word, Short, DWord or Long data types, an optional .bb (dot bit) can be appended to the address to reference a bit in a particular value. The valid ranges for the optional bit is 0-7 for Byte types; 0-15 for Word, Short and Boolean types; 0-31 for DWord and Long types; and 1-211 for String types. Float types do not support bit operations. Boolean and String types require a bit number. The bit number for String types specifies the number of characters in the string.

Dynamic addresses with bit numbers in the range of 0 to 7 will default to Byte, 8 to 15 will default to Word, and 16 to 31 will default to DWord. V Memory addresses with a bit number larger than 31 will default to String. The following diagram illustrates how the driver maps bits within the controller.



[e.g., V30.10@bool, V30.2@byte and V30.26@DWord all reference the same bit in the controller.]

**For Timer and Counter status bits, a dot bit notation is not used. The status bit for timer 7 would be T7 declared as Boolean.

***For Analog Inputs and Outputs, the address must be even (AI0, AI2, AI4...). Analog Outputs (AQ) are Write Only because there isn't a method to read the value of Analog Outputs from the device. Write Only types in this driver will return the last value written when read if an initial write to device has completed. If an initial write has not completed, then the driver will always return a value of 0 when read. This only applies while a client is connected to the server.

The actual number of addresses of each type depends on the Siemens S7-200 device in use (each type does not necessarily support an address of 0 to 65535). For address ranges, refer to the device's documentation.

Arrays

In addition to the address formats listed above, certain memory types (I, Q, M, SM, V, AI, AQ) support an array operation. Boolean arrays are not allowed at this time. To specify an array address, append [rows][cols] to the end of an address. If only [cols] is specified, [rows] will default to 1. With the array type, it is possible to read and write a block of 200 bytes at one time.

The maximum array size for Word and Short types is 100, and for DWord, Long and Float types is 50. The array size is determined by the multiplication of rows and cols.

Note: The maximum array size also depends on the maximum block size of the device being used.

Examples

1. To read and write an array of 10 Variable Memory Float values starting with V10, declare an address as follows: V10 [1][10] (choose Float for the data type.)

This array will read and write values to registers V10, V14, V18, V22 ... V46.

2. To read and write to bit 23 of Internal Memory Long M20, declare an address as follows:
M20.23 (choose Long for the data type.)

Strings

The driver allows for variable length strings to be stored in Variable Memory locations. The bit number specifies the string length (1-211) in characters. String data that is sent to the device, but is smaller in length than the string character count (bit number) is null terminated. String data that meets or exceeds the character length is truncated to the character count and sent to the device without a null terminator.

To read and write a string starting at V5 for a length of 10 characters (V Memory locations V5-V14 would be used to store this 10 character string), declare an address as follows: V5.10 (choose string for the data type.)

Not all devices will support up to 211 character requests in a single transaction. To determine the maximum number of characters that can be requested in a transaction, consult the device's documentation. This value is the largest string the driver can Read/Write to/from the device.

Caution: When modifying Word, Short, DWord, Long and Float types remember that each address starts at a byte offset within the device. Therefore, Words V0 and V1 overlap at byte 1. Writing to V0 will also modify the value held in V1. Similarly, DWord, Long and Float types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. As an example, when using DWords, use V0, V4, V8 and so on to prevent overlapping bytes.

S7-200 PPM Addressing

The S7-200 PPM addressing format is the same as the S7-200 addressing format. The model selection in this case determines whether the driver is using PPI protocol (normal S7-200 Mode) or PPM (S7-200 in Point to Point Modem) mode. In both cases, the addressing is the same. PPM mode is used with the target PLC is connected via the EM241 Modem module or via the programming port running in 10 bit mode.

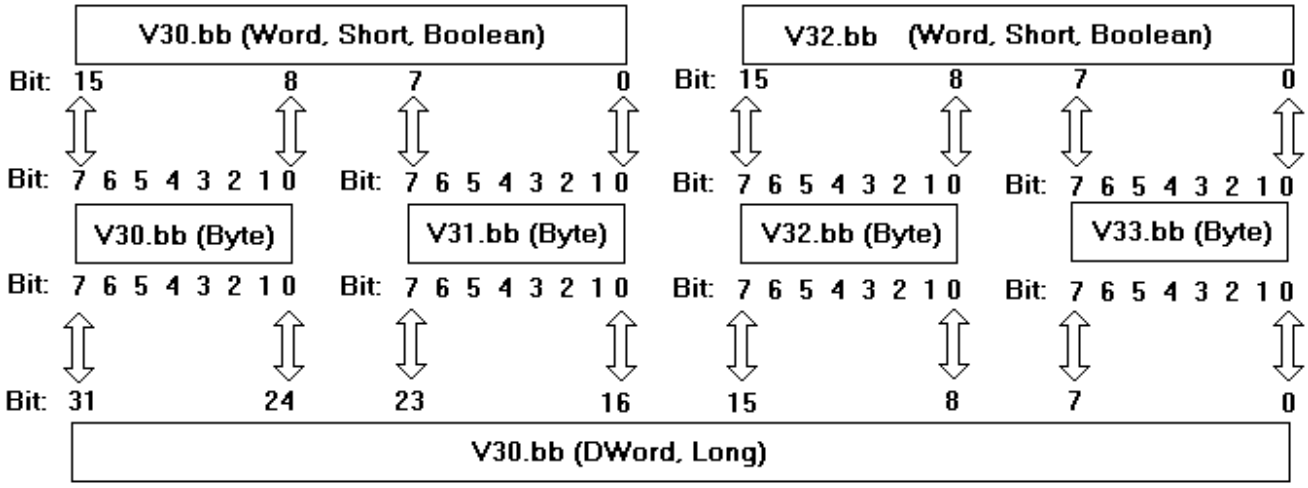
The default data types for dynamically defined tags are shown in **bold**.

Address Type	Range	Type	Access
Discrete Inputs	I00000-I65535 I00000-I65534 I00000-I65532 I00000.bb-I65535.bb* I00000.bb-I65534.bb* I00000.bb-I65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Discrete Outputs	Q00000-Q65535 Q00000-Q65534 Q00000-Q65532 Q00000.bb-Q65535.bb* Q00000.bb-Q65534.bb* Q00000.bb-Q65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Internal Memory	M00000-M65535 M00000-M65534 M00000-M65532 M00000.bb-M65535.bb* M00000.bb-M65534.bb* M00000.bb-M65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write
Special Memory	SM00000-SM65535 SM00000-SM65534 SM00000-SM65532 SM00000.bb-SM65535.bb* SM00000.bb-SM65534.bb* SM00000.bb-SM65532.bb*	Byte **Word**, Short DWord, Long, Float **Byte** Boolean, **Word**, Short **DWord**, Long	Read/Write SM0-SM29 are Read Only
Variable Memory	V00000-V65535	Byte	Read/Write

	V00000-V65534 V00000-V65532 V00000.bb-V65535.bb* V00000.bb-V65534.bb* V00000.bb-V65532.bb*	**Word**, Short DWord, Long, Float, String **Byte** Boolean, **Word**, Short **DWord**, Long, String	
Timer Current Values	T00000-T65535	**DWord**, Long	Read/Write
Timer Status Bits	T00000-T65535	Boolean**	Read Only
Counter Current Values	C00000-C65535	**Word**, Short	Read/Write
Counter Status Bits	C00000-C65535	Boolean**	Read Only
High Speed Counters	HC00000-HC65535	**DWord**, Long	Read Only
Analog Inputs	AI00000-AI65534***	**Word**, Short	Read Only
Analog Outputs	AQ00000-AQ65534***	**Word**, Short	Write Only

*For Byte, Word, Short, DWord or Long data types, an optional .bb (dot bit) can be appended to the address to reference a bit in a particular value. The valid ranges for the optional bit is 0-7 for Byte types; 0-15 for Word, Short and Boolean types; 0-31 for DWord and Long types; and 1-211 for String types. Float types do not support bit operations. Boolean and String types require a bit number. The bit number for String types specifies the number of characters in the string.

Dynamic addresses with bit numbers in the range of 0 to 7 will default to Byte, 8 to 15 will default to Word, and 16 to 31 will default to DWord. V Memory addresses with a bit number larger than 31 will default to String. The following diagram illustrates how the driver maps bits within the controller.



[e.g., V30.10@bool, V30.2@byte and V30.26@DWord all reference the same bit in the controller.]

**For Timer and Counter status bits, a dot bit notation is not used. The status bit for timer 7 would be T7 declared as Boolean.

***For Analog Inputs and Outputs the address must be even (AI0, AI2, AI4...). Analog Outputs (AQ) are Write Only because there isn't a method to read the value of Analog Outputs from the device. Write Only types in this driver will return the last value written when read if an initial write to device has completed. If an initial write has not completed, then the driver will always return a value of 0 when read. This only applies while a client is connected to the server.

The actual number of addresses of each type depends on the Siemens S7-200 device in use (each type does not necessarily support an address of 0 to 65535). Refer to the device documentation for address ranges.

Arrays

In addition to the address formats listed above, certain memory types (I, Q, M, SM, V, AI, AQ) support an array operation. Boolean arrays are not allowed at this time. To specify an array address, append [rows][cols] to the end of an address. If only [cols] is specified, [rows] will default to 1. With the array type, it is possible to read and write a block of 200 bytes at one time.

The maximum array size for Word and Short types is 100, and for DWord, Long and Float types is 50. The array size is determined by the multiplication of rows and cols.

Note: The maximum array size also depends on the maximum block size of the device being used.

Examples

1. To read and write an array of 10 Variable Memory Float values starting with V10, declare an address as follows: V10 [1][10] (choose Float for the data type.)
This array will read and write values to registers V10, V14, V18, V22 ... V46.

2. To read and write to bit 23 of Internal Memory Long M20, declare an address as follows:
M20.23 (choose Long for the data type.)

Strings

The driver allows for variable length strings to be stored in Variable Memory locations. The bit number specifies the string length (1-211) in characters. String data that is sent to the device, but is smaller in length than the string character count (bit number) is null terminated. String data that meets or exceeds the character length is truncated to the character count and sent to the device without a null terminator.

To read and write a string starting at V5 for a length of 10 characters (V Memory locations V5-V14 would be used to store this 10 character string), declare an address as follows: V5.10 (choose string for the data type.)

Not all devices will support up to 211 character requests in a single transaction. To determine the maximum number of characters that can be requested in a transaction, refer to the device documentation. This value is the largest string the driver can Read/Write to/from the device.

Caution: When modifying Word, Short, DWord, Long and Float types remember that each address starts at a byte offset within the device. Therefore, Words V0 and V1 overlap at byte 1. Writing to V0 will also modify the value held in V1. Similarly, DWord, Long and Float 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 V0, V4, V8 ... and so on, to prevent overlapping bytes.

Error Descriptions

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

Address Validation

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

Serial Communications

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

Device Status Messages

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

Device Specific Messages

Bad block starting at '<start address>' for a length of '<block size in bytes>' on device '<device name>'

Address Validation

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

Address Validation

Missing address

Error Type:
Warning

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

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

Device address '<address>' contains a syntax error

Error Type:
Warning

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

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

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

Error Type:
Warning

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

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

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

Error Type:
Warning

Possible Cause:
A tag address that has been specified dynamically references a location that is valid for the communications protocol

but not supported by the target device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application. Also verify that the selected model name for the device is correct.

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

Error Type:
Warning

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

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

Device address '<address>' is Read Only

Error Type:
Warning

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

Solution:
Change the access mode in the client application.

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

Error Type:
Warning

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

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

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

Error Type:
Warning

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

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

Serial Communications

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

Serial Communications

COMn does not exist

Error Type:
Fatal

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

Solution:
Verify that the proper COM port has been selected.

Error opening COMn

Error Type:
Fatal

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

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

COMn is in use by another application

Error Type:
Fatal

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

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

Unable to set comm parameters on COMn

Error Type:
Fatal

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

Solution:
Verify the serial parameters and make any necessary changes.

Communications error on COMn [<error mask>]

Error Type:
Serious

Error Mask Definitions:

B = Hardware break detected.
F = Framing error.
E= I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

Possible Cause:

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

Solution:

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

Device Status Messages

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

Device Status Messages

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

Device '<device name>' is not responding

Error Type:

Serious

Possible Cause:

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

Solution:

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

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

Error Type:

Serious

Possible Cause:

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

Solution:

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

Device Specific Messages

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

Device Specific Messages

Bad block starting at '<start address>' for a length of '<block size in bytes>' on device '<device name>'

Error Type:

Serious

Possible Cause:

An attempt has been made to reference a block of memory that contains at least one nonexistent location in the specified device.

Solution:

Verify that the tags assigned to addresses are within the specified range on the device and eliminate ones that reference invalid locations.

Index

- A -

- B -

- C -

- D -

- E -

- F -

- L -

- M -

- N -

- O -

- P -

- S -

- U -

- W -