AVEVA™ Communication Drivers Pack – Schneider Electric - SOMAC Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868892

Contact Information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

AVEVA

Chapter 1

Getting Started

This document describes the SOMAC Communication Driver, and the device and protocol environment in which it works. It includes application-level and bus-level communications protocols, item naming conventions, and Communication Driver features.

About the SOMAC Communication Driver

The Wonderware Operations Integration Schneider Electric SOMAC Communication Driver (referred to as the Communication Driver through the remainder of this user's guide) is a Microsoft® Windows® application program that acts as a communications protocol server. This Communication Driver is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in, which is part of the Operations Control Management Console (OCMC) suite of utilities.

This Communication Driver allows other Windows application programs access to data in PLCs (also referred to as devices) attached to an Ethernet network. The Communication Driver requires a TCP/IP package that supports the WinSock interface standard. It can access data directly via the Ethernet in Schneider Electric SOMAC family of controllers, such as the M241 and M251. The server can operate in either stand-alone mode or connect with any OPC, DDE, or SuiteLink compliant client application.

This Communication Driver documentation covers only the information you need to configure and run the Communication Driver component. See the documentation that comes with the related components for details on their operation. You can find installation instructions in a help file on the distribution CD. Many high-level functions and user-interface elements of the OI Server Manager are universal to all Communication Drivers, and only the documentation for the OI Server Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the OI Server Manager is critical to understanding this user's guide. To read the documentation about the MMC and OI Server Manager, click the **Help Topics** on the OCMC **Help** menu.

Note: The shortcut menu items described in this document typically represent only a subset of any actual shortcut menu. Most items in each shortcut menu are standard Windows commands. See the MMC Help for more information about those commands.

Support Client Protocols

The client applications connect to the SOMAC Communication Driver using following protocols:

- OPC
- SuiteLink
- DDE/FastDDE
- PCS

For more information refer to the "Support Client Protocols" section of the AVEVA Communication Drivers Pack Help.

Supported Hardware and Software

The SOMAC Communication Driver provides connectivity to the following Schneider Electric logic and motion controllers from the Modicon family:

Logic Controllers:

- Modicon M241 PLC
- Modicon M251 PLC

Motion Controllers:

- PacDrive 3 LMC Eco / Pro / Pro 2

The SOMAC Communication Driver **does not** support connectivity to the following Schneider Electric logic controllers from the Modicon family:

- Modicon M258
- Modicon TSX Micro
- Modicon M221
- Modicon Easy M100
- Modicon Easy M200

Conformance

The following hardware and software was used for conformance testing of this Communication Driver.

- Schneider Electric Modicon M241/M251
- PacDrive 3 LMC Eco / Pro / Pro 2

Supported Device Protocols

The SOMAC Communication Driver is designed to provide direct connectivity to the Schneider Electric controllers from the Modicon family. It uses TCP/IP (Transmission Control Protocol/Internet Protocol) bus-level protocol to communicate with all devices across an Ethernet network.

TCP is the lower-level transport and data-link vehicle for data delivery over an IP network. It provides reliable connection-oriented full-duplex data stream transport. IP is the basic protocol for the Internet which uses an IP address scheme to send data in packets across networks.

Licensing for SOMAC Communication Driver

The SOMAC Communication Driver supports the activation-based licensing to acquire the license both locally and remotely.

For more information on activation-based licensing, see "Centralized (Activation-Based) Licensing" in the Communication Drivers Pack Help.

Chapter 2

Configuring the SOMAC Communication Driver

Determining the SOMAC Communication Driver Hierarchy

Each Communication Driver is identified by a unique program name (ProgID) under the OCMC. The ProgID for the SOMAC Communication Driver is: OI.SOMAC.1. You can find it under the local node of the default group of the OI Server Manager, on the computer where the Communication Driver is installed.

You do not need to install the OI Server Manager on the same computer as the Communication Driver. When you access the Communication Driver remotely, you will not find the Communication Driver node under the local node. You must locate and identify the Communication Driver on a computer in one of the node groups.

To find the Communication Driver

1. Start the Operations Control Management Console (From the Windows **Start** menu, point to **Programs**, **AVEVA**, and then click the **Operations Control Management Console** icon).
2. In the Communication DriverOI Server Manager tree, under the **Local** node, navigate to the **Schneider Electric - SOMAC** Communication Driver.

To determine the SOMAC Communication Driver Hierarchy

Before configuring your Communication Driver, you should determine the hierarchical structure of your network/PLC environment to establish communications to each of the controllers. The SOMAC hierarchy in the Communication Driver starts with the Channel Selector connection, followed by the Device Selector connections.

1. Start the **Operations Control Management Console**.
2. In the OI Server Manager tree, under the **Local** node, navigate to the SOMAC Communication Driver.
3. Expand the Communication Driver, and then click **Configuration**.

   The **Global Parameters** tab appears in the details pane.

4. Configure all the global parameters as required for this Communication Driver. The default **Poke Mode** settings for the Communication Driver is **Optimization mode**. For more information about the **Global Parameters** dialog box, including descriptions of the different poke modes, see the Communication Drivers Pack Help.

   Note: Any global parameters fields that appear disabled are not supported for the Communication Driver.

5. When the SOMAC Communication Driver hierarchy build is completed, you can start configuring the respective devices for communications.
   - You can create the desired device groups in the **Device Groups** section with each of the PLC objects.
   - You can create the desired device items under the **Device Items** section with each of the PLC objects.

Object Naming Convention within a Hierarchy

1. The format of the default name of an hierarchy object is
   ```
   New_<ObjectName>_###
   ```
   where,

   <ObjectName>: name of the object type

   ###: numeric value starting from "000" enumerated sequentially per hierarchy object
2. The hierarchy object name can contain up to 32 characters.
3. The link name for the OPC items is constructed by assembling the respective object names of the nodes along the hierarchy tree in the logical order, starting from the data source root down to the leaf. Therefore, the link name is always unique.

Setting Up a SOMAC Communication Driver for the First Time

If you are setting up an Communication Driver for the first time, perform the following tasks in the order listed:

1. Locate the Communication Driver in the Operations Control Management Console (OCMC). In the OI Server Manager tree, under the Local node, the Communication Driver base instance name is OI.SOMAC.
2. Configure the global parameters. See "Configuring Global Parameters" in the Communication Drivers Pack help.
3. Add one or more channel selector connections. See Adding and Configuring Channel Selector.
4. Add one or more device selector connections. See Adding and Configuring Device Selector.
5. Add one more device groups. See Device Group Definitions.
6. Add device items. See Device Item Definitions.
7. Activate the Communication Driver. See "Activating/Deactivating the OI Server" in the Communication Drivers Pack Help.
8. Troubleshoot any problems. See Troubleshooting the SOMAC Communication Driver.

Adding and Configuring Channel Selector

The server-specific configuration portion of the SOMAC Communication Driver hierarchy tree under the OI Server Manager starts at the Channel Selector object. This object lets you set server parameters for communication with agents (devices) in the hierarchy tree.

To add a ChannelSelector connection to your SOMAC hierarchy

1. In the console tree, right-click **Configuration** and then click **Add ChannelSelector Connection**. The **New_ChannelSelector_000** object appears in the hierarchy.
2. Edit the object name to appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. You can rename the hierarchy entry later.

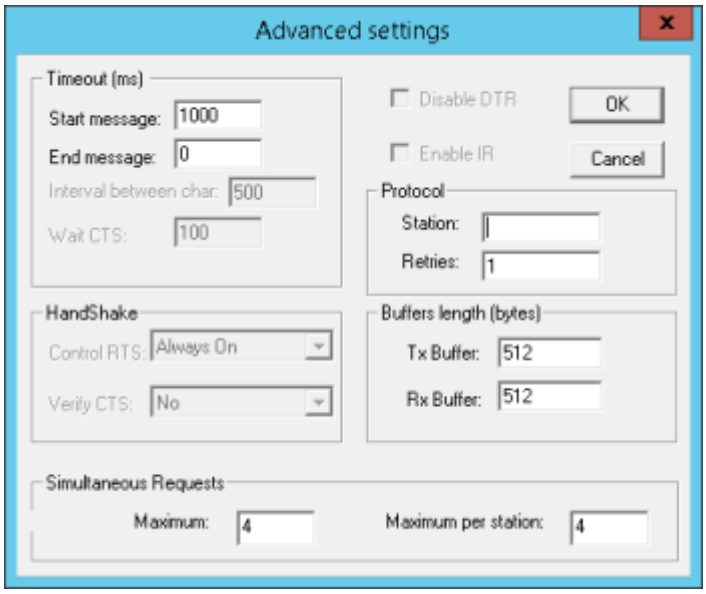
   The **New_ChannelSelector_000 Parameters** view is displayed.

Configuring Advanced Settings for Channel Selector connection

To configure the Advanced Settings

1. In the **New_ChannelSelector_000 Parameters** view, click **Advanced..**

   The **Advanced settings** dialog appears.



2. Change the settings as needed.
3. Click **OK** to close the **Advanced Settings** dialog box.
4. Click **Save** to save the changes.

The configuration view of the **Advanced Settings** dialog displays the following parameters.

Timeout (ms)

- **Start message**: The timeout (in milliseconds) to receive the start of a message. The default value is **1000** ms.
- **End message**: The timeout (in milliseconds) to receive the end of a message. The default value is **0** ms.

- **Interval between char**: (non-configurable) The interval (in milliseconds) between characters in a message. The pre-set value is **500**.
- **Wait CTS**: (non-configurable) The timeout for the Clear to Send wait. The pre-set value is **100**.

HandShake

- **Control RTS**: (non-configurable) Specifies whether to use the Request to Send control. The pre-set value is **Always On**.
- **Verify CTS**: (non-configurable) Specifies whether to use the Clear to Send verification type. The pre-set value is **No**.

Protocol

- **Station**: The station number or ID of the channel, according to the device protocol being used. Some master/slave protocols consider the Communication Driver to be another slave device and therefore require it to have its own station ID. For more information, see the documentation for the specific Communication Driver.
- **Retries**: The number of times that the Communication Driver will retry the same command before generating a communication error. The default value is **1**.

Buffer length (bytes)

- **Tx Buffer**: The memory for data transmission to the SoMachine devices. The default value is **512** bytes.
- **Rx Buffer**: The memory for data received from the SoMachine devices. The default value is **512** bytes.

Simultaneous Requests

- **Maximum**: The maximum number of requests (up to 32) that can be sent at the same time to all devices in the channel. The default value is **4**.

  Note: There is a limit of 100 total simultaneous requests across all channels in a server instance, which means you can have three channels at 32 each, four channels at 24 each, five channels at 20 each, and so on up to 100 channels at 1 each, or any combination thereof.

- **Maximum per station**: The maximum number of requests that can be sent at the same time to a single device in the channel. The default value is **4**.

Disable DTR

When disabled, no DTR signal is sent before starting a communication. It is a non-configurable parameter, and the check-box is not selected by default.

Enable IR

Available only on Windows Embedded target systems. Enables use of Infrared interface (COM2 port) rather than a standard serial port to communicate with devices. It is a non-configurable parameter, and the check-box is not selected by default.

Adding and Configuring Device Selector

The SOMAC Communication Driver can connect to different Windows agents, PLCs, and other data sources. These connections are modeled in the hierarchy by means of Device Selector objects, each of which models the end-point of the communications path.

From the **ChannelSelector** branch of the Communication Driver hierarchy, create the new **DeviceSelector** object.

To add a Device Selector connection to your SOMAC hierarchy

1. Right-click the **New_ChannelSelector_000** object, and select **Add DeviceSelector Connection**.

   The **New_DeviceSelector_000** object is created.
2. Rename the object as appropriate.
3. **The New_DeviceSelector_000 Parameters** configuration view is displayed.

To configure the Device Selector connection

- Configure the **Station**.

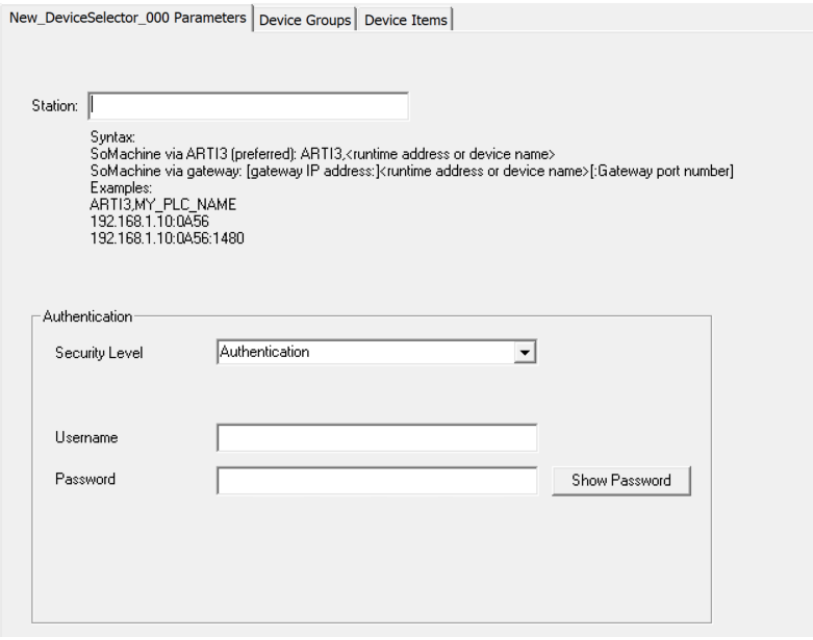
  The Station field cannot be empty. The syntax of the Station depends on the ARTI3 or the gateway being used by the Communication Driver. For more information, see Setting Station ID of a Device.

To configure a secure device connection

1. In the **Authentication** section, click the **Security** dropdown and select **Authentication.**
2. If the connection to the data source is unsecure, select **No Security**.
3. In the **Username** field, enter the name of the user with access to the secure device.
4. In the **Password** field, enter the password that is associated with the secure device.

Authenticating the Connection

The **Authentication** section allows you to configure the security settings while connecting to a secure (ARTI3) device.



To configure a secure connection

1. In the **Authentication** section, click the **Security** dropdown and select Authentication.

   If the connection to the data source is unsecure, select **No Security**.

2. In the **Username** field, enter the name of the user associated with the secure device.
3. In the **Password** field, enter the password that is associated with the secure device.

Setting Station ID of a Device

Set the station ID for a selected device so that the Communication Driver can identify and communicate with it on the network.

Syntax

To connect to a SoMachine device via ARTI3, use the following syntax:

```
ARTI3,<runtime address or device name>
```

To connect to a SoMachine device via gateway, use the following syntax:

```
[gateway IP address:]<runtime address or device name>[:Gateway port number]
```

To connect to a SoMachine device via CFILE, user the following syntax:

```
CFILE,<Configuration File>,<Configuration ID>
```

where,

- **gateway IP address:** The IP address of the SoMachine Gateway server that is managing communication with the SoMachine device(s). If no address is specified, it will connect directly using ARTI.
- **runtime address or device name**: The node name (case sensitive) or hexadecimal address of the Schneider Electric device.

  The node name is configured during the communication setup within SoMachine for the respective PLC. It can be obtained from the **Controller Selection** tab for the controller. The Controller NodeName defaults to the controller part number and MAC address. However, it is recommended to change the Controller NodeName so the configuration works even if the PLC is replaced and to avoid other problems.
- **gateway port number**: The port number of the SoMachine Gateway server that is managing communication with the Schneider Electric device(s). If no port is specified, the default is 1217.
- **runtime IP address**: The IP address of the Schneider Electric device. If no address is specified, the default is 127.0.0.1 (i.e., localhost).
- **Configuration File**: The path relative to the application path.
- **Configuration ID**: The configuration ID within the configuration file.

Examples:

ARTI3,MY_PLC_NAME

192.168.1.10:0A56

192.168.1.10:0A56: 1480

CFILE,.\A\File.ini,PLC:PLCWinNT

AVEVA

Chapter 3

Device Groups and Device Items

Device Group Definitions

Use the **Device Groups** configuration view, to create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box. To open the Device Groups dialog box, in the Device Selector configuration editor, click the **Device Groups** tab.

Note: When you select another part of the Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

To create or add device groups

1. Right-click anywhere in the table, and then click **Add**. A device group is added with a default name and update interval.
2. Enter a unique name up to 32 characters long for the device group.

To delete device groups

1. Right-click the device group to be deleted, and then click **Delete**.
2. Read the warning, and then click **Yes**.

To edit device groups

Use the **Edit** option from the **Device Groups** tab only for configuring the Communication Driver's unsolicited message handling.

To configure default update intervals

To configure a default update interval for the object, right-click in the **Device Groups** box and then click **Config Default Update Interval**.

To edit update intervals

To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

or

Right-click its value in the **Update Interval** column and then click **Modify Update** Interval.

The update interval is the frequency, in milliseconds, that the Communication Driver acquires data from the topics associated with that device group.

Different topics can be polled at different rates from a PLC by defining multiple device group names for the same PLC and setting a different update interval for each device group.

Device Item Definitions

The device item name is an is an “alias” or a label for the data in the device. It is an alternative name for the item reference, and ca be used instead of the item reference when you create the client application. Device item configuration is optional, but is strongly recommended.

To create or add device items

1. Right-click anywhere in the table, and then click **Add**.
2. In the **Name** column, type a unique item name. The maximum is 32 characters.
3. In the corresponding line, double-click the **Item Reference** column and enter the correlated item reference for the name you created.

To rename device items

Right-click the device item to be renamed and click **Rename**. Make the changes.

To delete device items

Right-click the device item to be deleted from the list and click **Delete**.

To clear all device items

Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

NOTE: You can import a .csv file containing your item definitions to help streamline configuration. See "Exporting and Importing CSV Files" in the Communication Drivers Help Pack.

AVEVA

Chapter 4

SOMAC Communication Driver References

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the Communication Driver itself.

This section only describes the item reference syntax and options for the SOMAC driver. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the Communication Drivers Pack Help.

Item Reference Syntax

Item references in this Communication Driver use the following syntax. For more information about the referenced addresses, see the manufacturer's documentation for your device.

For local and global variables in a SoMachine device, use the following syntax:

```
<application name>.<object name>.<variable name>
```

where,

- **application name**: the name of the application
- **object name**: the name of the program organization unit (POU), global variable list, or other programming object that contains the variable. For example, PLC_PRG.
- **variable name**: the name of the variable

Supported Data Types

The data type is specified as a suffix in the item syntax. This Communication Driver supports the following data types:

- BYTE, WORD, DWORD, SINT, USINT, INT, REAL, UINT, DINT, UDINT, STRING, TIME, DATE, BOOL, LWORD, LINT, ULINT, LREAL, DATE_AND_TIME, DT, TIME_OF_DAY, TOD, WSTRING, LTIME
- User-defined structures (DUT), enumerators and so on.

- Multi-dimensional arrays and an array of arrays that start with positive or negative indices.

Examples of SOMAC Item References

These are examples of valid item references for this Communication Driver.

Local	**Global**
Application.PLC_PRG.initPosition1	Application.GVL.initPosition1
Application.PLC_PRG.Timer2[1,3,0].StartTime	Application.Global_POU.Timer2[1,3,0].StartTime
Application.PLC_PRG.DINT_Array_3Dim_2[1,-5,0]	Application.GVL.DINT_Array_3Dim_2[1,-5,0]
Application.PLC_Local_POU.struct[9][5].Bool	Application.Global1.struct[9][5].Bool
Application.PLC_PRG.Dint_Array_negative_index[-5].member1	Application.Global_POU.Dint_Array_negative_index[-5].member1

For more information about the referenced addresses, see the documentation provided by the manufacturer for your device.

AVEVA

Chapter 5

Troubleshooting the SOMAC Communication Driver

SOMAC Communication Driver Error Codes

The following tables describe the error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
1	PLC not connected	• Lost connection to the PLC due to a hardware failure,such as PLC in error node,or cables issues. • Wrong **Station** field configuration.	• Check the **Station** field configuration,confirming that the IP Addresses for the Gateway (if it is used) and the PLC are correct, as well as the PLC ID number in hexadecimal format. • Check if the PLC is running and if you can ping it.
2	Login to PLC has failed	Some devices only allow a log-in of one application.	If there is another program connected to the PLC, you need to disconnect it (i.e.,log off). Then you should be able to communicate with the PLC.
3	No cyclic list has been found	Invalid list or no list variables to read.	Internal error related to the PLC Handler functions CycDefineVarList and CycEnterVarAccess.

4	PLCHandler is inactive	PLCHandler instance is not set active.This error happens when you use the INI file option and it is misconfigured.	Properly configure the INI file and the Station field.
5	Loading of the symbols has failed	There is no symbol configuration in the application.	Create the Symbol Configuration accordingly.
6	The defined communication interface is not valid or not supported	The interface is not supported(ARTI,Gateway). This error happens when trying to establish a connection with the PLC.	Check if your SOMAC configuration supports the desired interface(GATEWAY,ARTI, INI file).
7	Communication error occurred during action	• Error while trying to start the communication with the PLC. • Exceeded number of retries to receive a response from the PLC before throwing a COMM_FATAL. Related to the PLCHanlder PlcConfig Struct.	Check if your PLC is properly configured and reachable.
8	Wrong or erroneous configuration of the PLCHandler	No configuration for this PLCHandler instance (Id unknown).This error happens when trying to establish a connection with the PLC and you are using a INI file that is not properly configured for that PLC instance.	Properly configure the INI file.
9	Invalid parameter	Invalid function parameters (fore.g.NULL).Usually happens when trying to retrieve the Variable Names from the PLC.	Internal error related to the PLCHandler functions GetAllItems, GetItem, and CycEnterVarAccess.

10	Communication interface not resp.Incorrectly installed (e.g.,Gateway Dlls not available)	The interface can't start successfully (missing interface-dependent DLLs).This error happens when trying to establish a connection with the PLC.	If you are using the Gateway,check to see if it properly installed and running.
11	Method not yet supported resp.implemented	Spare error.	Not applicable.
12	Exception occurred during action	An exception occurred in the underlying interface. This error happens when transferring any application service to the PLC.	Internal driver error related to the PLCHandler function SyncSendService.
13	Timeout time exceeded	Time for the answer on a data package from the PLC exceeded. This could be caused by a wrong Station field configuration or the PLC is unreachable.	• Check the **Station** field. • Check if you can have access to the PLC using pinging and testing the TCP/IP ports
14	PLC already connected(at a further::Connect function call)	The driver tried to reconnect to a PLC that is already connected.	Internal error related to the PLCHandler function Connect.
15	Reconnect thread already active	Reconnect thread is still active.This error happens when trying to establish a connection with the PLC.	Internal error related to the PLCHandler function Connect.
16	Symbols available offline	Cannot open connection to the PLC but could load the symbol file offline. This error happens when trying to establish a connection with the PLC.	Internal error related to the PLCHandler function Connect.
17	Asynchronous operation	Asynchronous operation(e.g.,cyclic read of variables)has not yet finished.	Internal PLCHandler error that should never happen on this driver.Contact technical support if this error occurs.

18	ctiveX error	Internal error.	The communication driver does not use this capability of PLCHandler, if you see this error it is probably a problem with the PLCHandler.Please contact technical support.
19	Target ID mismatch	PLC does not match to the passed target ID specified.	Use the programming software to scan the network and find the correct PLC ID
20	Object not found	No object found for the required action(e.g., tried to get an element beyond the end of the list).	Contact technical support.
21	Components not loaded	No object found for the required action(e.g.,tried to get an element beyond the end of the list).	Components required to establish communication are missing. Please contact your supplier to receive the additional files.
22	Busy	Last action still in progress, cannot start the required one	• The driver tried to start a communication task before the previous one was completed.Contact the technical support. • If you are seeing intermittent communication problems because of this issue, please try increasing the number of retries.
23	Disabled	Driver tried to use the log feature but logging is disabled.	Contact technical support.
50	Invalid type	Results returned by the PLCHandler or specified by the driver are invalid.	Contact technical support.

51	Symbols not found	None of the variables specified match the symbols currently present in the PLC.	• Make sure that your symbols are properly added to the controller. • Verify if the name specified in the driver work sheet matches the variable name in the PLC.
52	Initialization error	The operating system does not have enough resources for the driver initialization.	Enable the protocol analyzer and run the driver again to retrieve further details.
53	Memory allocation error	• The driver could not allocate memory. • Internal programming error in the driver.	• Verify the memory available on your device. • If enough memory is available, contact technical support.
54	Driver is closing	Driver could not be initialized because it is in shutdown process.	Wait for until the driver close and then retry.
55	PLCHandler returned invalid code	PLCHandler function GetLastError returned zero after a read or write failure.	Contact technical support.
0	OK	Communicating without error.	None required.
-15	Time out waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode,or in error mode. • Wrong station number. • Wrong parity(for serial communication). Wrong RTS/CTS configuration(for serial communication).	• Check cable wiring. • Check the PLC mode—it must be RUN. • Check the station number. • Increase the time out in the driver's advanced settings. • Check the RTS/CTS configuration(for serial communication).

Maximum Simultaneous Request Settings

When using ARTI3 communication (UDP), if you configure the **Maximum** and **Maximum per station** fields under the **Simultaneous Requests** section with a value greater than 4 for each field, the SOMAC Communication Driver cannot connect to more than 4 devices.

The real maximum simultaneous requests settings cannot exceed 4 since there is a limitation in the SOMAC protocol library PLCHandler.

Each device attempts to create 1 UDP port. If the number of UDP ports created from all devices is greater than the driver's maximum simultaneous request, then it stops creating new UDP ports.

Regardless of how many devices there are, when the maximum simultaneous requests are less than or equal to 4, the driver distributes and shares those ports to all connected devices. Each device has multiple connections to the PLC (based on the maximum simultaneous requests settings), and also depends on the device's unused available channels/connection limit.

If the maximum simultaneous requests are greater than 4, the attempts of the device to open more ports fail, and the Communication Driver logic assigns an exclusive port per device. This is the reason why the driver cannot connect to more than 4 devices as all ports are being held exclusively per device if the maximum simultaneous requests are set to 4.

In conclusion, the SOMAC Communication Driver has the following limitations in functionality:

- Port sharing: The driver opens a new port for each device upto 4 UDP ports and shares them as long as maximum request setting is lesser than 4.
- Exclusive port assignment to each device: The driver does not share the ports assigned to each device with any other device once the maximum number of UDP ports are opened.

To avoid exclusive port assignment to each device, maximum simultaneous requests and maximum simultaneous requests per station should be limited to 4 each under 1 channel.