

**NLMPI Communication Driver**

Driver for TCP/IP Communication  
with Siemens Devices Using NetLink-MPI Converter

**Contents**

**INTRODUCTION .....2**

**GENERAL INFORMATION.....3**

    DEVICE CHARACTERISTICS .....3

    LINK CHARACTERISTICS.....3

    DRIVER CHARACTERISTICS .....3

    CONFORMANCE TESTING .....4

**INSTALLING THE DRIVER .....5**

**CONFIGURING THE DRIVER .....6**

    SETTING THE COMMUNICATION PARAMETERS .....6

    CONFIGURING THE DRIVER WORKSHEETS .....8

**EXECUTING THE DRIVER ..... 12**

**TROUBLESHOOTING ..... 13**

**SAMPLE APPLICATION ..... 15**

**REVISION HISTORY..... 16**

## Introduction

The NLMPI driver enables communication between the Studio system and devices using the NetLink-MPI protocol communicating over TCP/IP, according to the specifications discussed in this document.

This document was designed to help you install, configure, and execute the NLMPI driver to enable communication with these devices. The information in this document is organized as follows:

- **Introduction:** Provides an overview of the NLMPI driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the NLMPI driver.
- **Installing the Driver:** Explains how to install the NLMPI driver.
- **Configuring the Driver:** Explains how to configure the NLMPI driver.
- **Executing the Driver:** Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Troubleshooting:** Lists the most common error codes for this protocol and explains how to fix these errors.
- **Sample Application:** Explains how to use a sample application to test the NLMPI driver configuration.
- **Revision History:** Provides a log of all modifications made to the driver and the documentation.



### Notes:

- This document assumes that you have read the “Development Environment” chapter in the Studio *Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows XP environment. If you are unfamiliar with Windows XP, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

## General Information

This chapter explains how to identify all the hardware and software components used to implement communication between the Studio NLMPI driver and the Siemens Devices on the MPI port, using the Hilscher NetLink-MPI adapter.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

### Device Characteristics

To establish communication, you must use devices with the following specifications:

- **Manufacturer:** Hilscher
- **Compatible Equipment:**
  - Siemens S7-300/400 family
- **Programming Software:** Sycon
- **Device Runtime Software:** None

For a list of the devices used for conformance testing, see “Conformance Testing.”

### Link Characteristics

To establish communication, you must use links with the following specifications:

- **Device Communication Port:** MPI Port
- **Physical Protocol:** Converter from Ethernet to MPI
- **Logic Protocol:** MPI
- **Specific PC Board:** Any Ethernet card

### Driver Characteristics

The NLMPI driver is composed of the following files:

- **NLMPI.INI:** Internal driver file. *You must not modify this file.*
- **NLMPI.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **NLMPI.PDF:** Document providing detailed information about the NLMPI driver.
- **NLMPI.DLL:** Compiled driver.

#### **Notes:**

- All of the preceding files are installed in the /DRV subdirectory of the Studio installation directory.

You can use the NLMPI driver on the following operating systems:

- Windows XP/7/8
- Windows CE

For a list of the operating systems used for conformance testing, see “Conformance Testing” on page 4.

The NLMPI driver supports the following registers:

Register Type	Length	Write	Read	Bit
DB (Data Block)	1 Byte	•	•	•
M (Memory)	1 Byte	•	•	•
I (Input)	1 Byte	•	•	•
Q (Output)	1 Byte	•	•	•
C (Counter)	1 Word	•	•	•
T (Timer)	1 Word	•	•	•
ST (Status)	1 Word	•	•	•

**Conformance Testing**

The following hardware/software was used for conformance testing:

- **Cable:** Ethernet cable

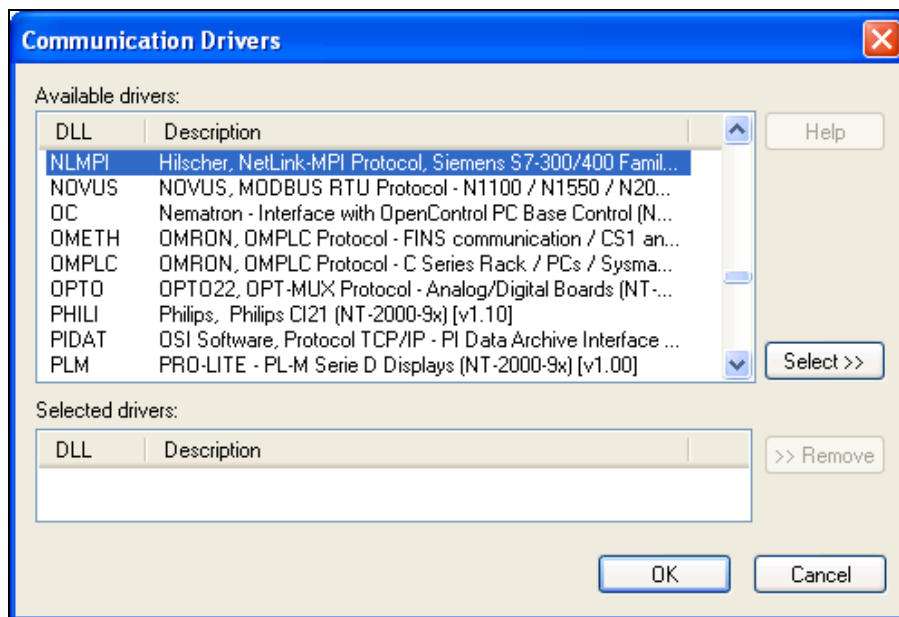
Driver Version	Studio Version	Operating System	Equipment
1.3	7.0	WinXP/7	S7-300 CPU315-2DP

## Installing the Driver

When you install Studio version 5.1 or higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the application you are using.

Perform the following steps to select the driver from within the application:

1. Open Studio from the **Start** menu.
2. From the Studio main menu bar, select **File** → **Open Project** to open your application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication drivers* dialog.
4. Select the **NLMPI** driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.



*Communication Drivers Dialog Box*

5. When the **NLMPI** driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.



**Note:**

It is not necessary to install any other software on your computer to enable communication between the host and the device.



**Attention:**

For safety reasons, you must use special precautions when installing the physical hardware. Consult the hardware manufacturer's documentation for specific instructions in this area.

## Configuring the Driver

After opening Studio and selecting the NLMPI driver, you must configure the driver. Configuring the NLMPI driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEETS* (or Communication tables)

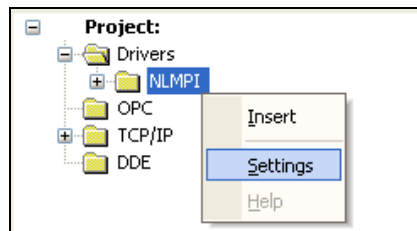
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**, and **Address** fields only.

**Note:**  
For a detailed description of the Studio *STANDARD DRIVER SHEETS*, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

### Setting the Communication Parameters

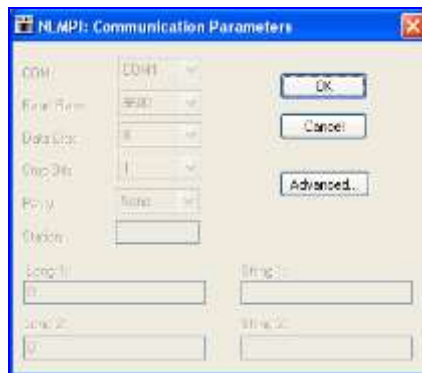
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

1. From the Studio development environment, select the *Comm* tab located below the *Workspace*.
2. Click on the *Drivers* folder in the *Workspace* to expand the folder.
3. Right-click on the NLMPI subfolder and when the pop-up menu displays, (as shown in the following figure) select the **Settings** option.



Select Settings from the Pop-Up Menu

The *NLMPI: Communications Parameters* dialog displays (as follows).



Communication Parameters Dialog

This Driver does not have custom parameters.

4. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings that are necessary.



**Notes:**

- Do not change any of the other *Advanced* parameters at this time. You can consult the Studio *Technical Reference Manual* for information about configuring these parameters for future reference.

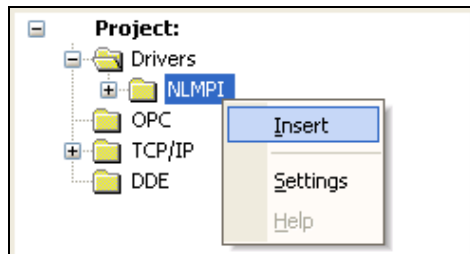
## Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEETS* (or communication tables) to associate application tags with the device addresses. You can configure multiple Driver Worksheets — each of which is divided into a *Header* section and *Body* section.

### Configuring the *STANDARD DRIVER SHEET*

Use the following steps to create a new *STANDARD DRIVER SHEET*:

1. From the Studio development environment, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *<Driver Name>* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



**Inserting a New Worksheet**

**Note:**

To optimize communication and ensure better system performance, you must tie the tags in different driver worksheets to the events that trigger communication between each tag group and the period in which each tag group must be read or written. Also, we recommend configuring the communication addresses in sequential blocks to improve performance.



The *STANDARD DRIVER SHEET* displays (similar to the following figure).

	Tag Name	Address	Div	Add
1	Tag[1]	1		
2	Tag[2]	W2		
3	Tag[3]	UW4		
4	Tag[4]	B6		
5	Tag[5]	7.4		
6	Tag[6]	8		
7	Tag[7]	W9.3		

**STANDARD DRIVER SHEET**

In general, all parameters on the Driver Worksheet (except the **Station**, **Header**, and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. For detailed information about configuring the standard parameters, consult the *Studio Technical Reference Manual*.

4. Use the following information to complete the **Station**, **Header**, and **Address** fields on this worksheet.

- **Station** field: Specify the device using the following syntax:

**<IP Address>:<Port Number>:<PLC ID>**

Where:

- **IP Address** is the PLC IP address in the TCP/IP network.
- **Port Number** is the port number for the NetLink protocol device (usually 1099).
- **PLC ID** is the PLC identification number (from 0 to 126).

- **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device and a reference to the initial address. (Default value is DB:0)

These variables must comply with the following syntax:

**<Type>:<AddressReference>** (For example: DB:10)

Where:

- **Type** is the register type. (DB, M, I, Q, C, T or ST)
- **AddressReference** is the initial address (reference) of the configured type. For DB, **Type** is the Data Block number.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

Also, you can type a tag string in brackets { **Tag** } into the **Header** field, but you must be certain that the tag's value is correct and that you are using the correct syntax, or you will get an invalid Header error.

The following table lists all of the data types and address ranges that are valid for the NLMPI driver.

Header Field Information			
Operands	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Default Data Type
DB (Data Block)	DB:20	Varies according to the equipment	UB (Unsigned Byte)
M (Memory)	M:15	Varies according to the equipment	UB (Unsigned Byte)
I (Input)	I:12	Varies according to the equipment	UB (Unsigned Byte)
Q (Output)	Q:120	Varies according to the equipment	UB (Unsigned Byte)
C (Counter)	C:7	Varies according to the equipment	BCD
T (Timer)	T:0	Varies according to the equipment	BCD
ST (Status)	ST:0	Varies according to the equipment	UW (Unsigned Word)

- **Address** field: Use this field to associate each tag to its respective device address.

Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to an address on the device. The address must comply with the following syntax:

*[Data type]<AddressOffset>. [Bit]* (For example: W 10, B20, 40, 10.5)

Where:

- **Data type** (*optional parameter*) Defines the data type of the operand. If not defined, Studio uses the default Data Types (See Table above). The following Data Types are available:
  - B (signed Byte),
  - UB (unsigned Byte),
  - W (signed Word – 2 bytes)
  - UW (unsigned Word – 2 bytes)
  - DW (signed double Word – 4 bytes)
  - UDW (unsigned double Word – 4 bytes)
  - F (Float Point – 4 bytes)
  - BCD (2 bytes)
- **AddressOffset** is a parameter added to the **AddressReference** parameter (configured in the **Header** field) to compose the group address configured in the **Header** field. If you configured DB in the **Header** field, this parameter defines the address of the Data Block.
- **Bit** (*optional parameter*) is the bit number to be read from the device.

**Attention:**

- The Word and BCD values are stored in two consecutive registers (byte), where the address value corresponds to the first register position. You must ensure that you do not configure a non-existent address or a conflict will occur.
- The Floating point and Double words value is stored in four, consecutive registers (byte), where the address value corresponds to the first register position. You must ensure that you do not configure a non-existent address or a conflict will occur.
- This driver supports Bit reading only; it cannot execute Bit writing.

Address Configuration Sample		
Device Address	Header Field	Address Field
DB5:100	DB:5	100
DB3:20 and DB3:21	DB:3	W20
M12	M:0	12
M12	M:10	2
M12 and M13	M:10	W2
M12 (bit 7)	M:2	10.7
T50	T:0	50
T50	T:25	25

➔ **Attention:**

You must not configure a range of addresses greater than the maximum block size (data buffer length) supported by each PLC within the same worksheet. The maximum data buffer length for this driver is 216 bytes per *STANDARD DRIVER SHEET*.

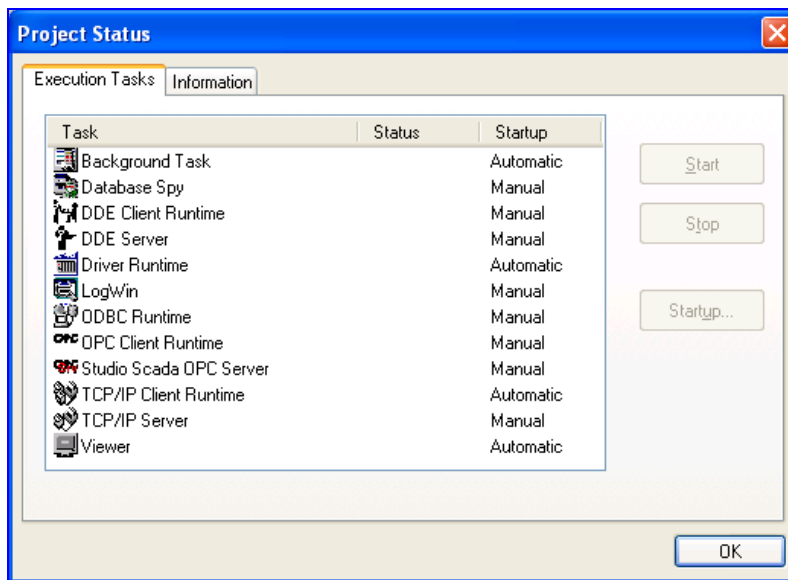
## Executing the Driver

After adding the NLMPI driver to a project, Studio sets the project to execute the driver automatically when you start the run-time environment.

To verify that the driver run-time task is enabled and will start correctly, perform the following steps:

1. Select **Project** → **Status** from the main menu bar.

The *Project Status* dialog box displays, as follows.



*Project Status Dialog Box*

2. Verify that the *Driver Runtime* task is set to **Automatic**.
  - If the setting is correct, click **OK** to close the dialog box.
  - If the **Driver Runtime** task is set to **Manual**, select the **Driver Runtime** line. When the **Startup** button becomes active, click the button to toggle the *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

## Troubleshooting

If the NLMPI driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required.
103	Error Connect	Wrong IP Address or Port Number	Use "Telnet" or "Ping" tools to check your network configuration and try to find the PLC with the computer on which you are running Studio.
104	Invalid Number of Message	Low timeout configured in communication parameters	Increase Timeout.
105	Invalid Answer Length	Equipment failed or is out of order, indicating that the framed request will generate a response whose size exceeds the available data size	Check the equipment state.
106	Error Writing bit	Driver only supports reading bits. Driver shows this error when trying to write to bit.	Write function is not supported for bits.
-15	Timeout Start Message	<ul style="list-style-type: none"> <li>▪ Disconnected cables</li> <li>▪ PLC is turned off, in stop mode, or in error mode</li> <li>▪ Wrong station number</li> <li>▪ Wrong RTS/CTS control settings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check cable wiring.</li> <li>▪ Check the PLC state – it must be RUN.</li> <li>▪ Check the station number.</li> <li>▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.</li> </ul>
-17	Timeout between rx char	<ul style="list-style-type: none"> <li>▪ PLC in stop mode or in error mode</li> <li>▪ Wrong station number</li> <li>▪ Wrong parity</li> <li>▪ Wrong RTS/CTS configuration settings</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check cable wiring.</li> <li>▪ Check the PLC state – it must be RUN.</li> <li>▪ Check the station number.</li> <li>▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.</li> </ul>
-34	Invalid Address	<ul style="list-style-type: none"> <li>▪ The address field specified is invalid or out of the supported range</li> </ul>	<ul style="list-style-type: none"> <li>▪ Check the supported ranges and specify an address within the supported range.</li> </ul>

➔ **Tip:**

You can verify communication status using the Studio development environment *Output* window (*LogWin* module). To establish an event log for **Field Read Commands**, **Field Write Commands**, and **Serial Communication**, right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing a Windows CE target, you can use the Remote LogWin of Studio (**Tools** → **Remote Logwin**) to get the log events from the target unit remotely.

If you are unable to establish communication with the PLC, try to establish communication between the PLC Programming Tool and the PLC. Quite frequently, communication is not possible because you have a hardware or cable problem, or a PLC configuration error. After successfully establishing communication between the device's Programming Tool and the PLC, you can retest the supervisory driver.

To test communication with Studio, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Studio version**: To find this information, select **Help** → **About**.
- **Driver Version**: To find this information, read the full description of the driver on the **Communication Drivers** Dialog Box.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands**, and **Serial Communication** for the LogWin window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

## Sample Application

This driver currently does not have a Sample application

## Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.0	Fábio H.Y.Komura	Jun/22/2004	Initial version
B	1.1	Fábio H.Y.Komura	Nov/23/2004	Solved problem of communicating with more than 1 device
C	1.2	Fábio H.Y.Komura	Feb/25/2005	Solved problem of communicating with more than 2 devices
D	1.3	Paulo Balbino	Jul/7/2011	Solved issue with Exceptions being generated because of modifications on the DrvStd.dll
E	1.3	Anushree Phanse	Jun/4/2015	Driver Documentation updated for error codes. No change in the driver.