

MOLOW Communication Driver

Driver for Serial Communication
with Devices Using the MOLOW Protocol

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS.....3

 DRIVER CHARACTERISTICS4

 CONFORMANCE TESTING4

INSTALLING THE DRIVER6

CONFIGURING THE DEVICE7

CONFIGURING THE DRIVER8

 SETTING THE COMMUNICATION PARAMETERS8

 CONFIGURING THE STANDARD DRIVER WORKSHEET 11

EXECUTING THE DRIVER 15

TROUBLESHOOTING 16

SAMPLE APPLICATION 18

REVISION HISTORY..... 19

Introduction

The MOLOW driver enables communication between the Studio system and a set of devices using the MOLOW protocol by serial link, according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the MOLOW driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the MOLOW driver.
- **Installing the Driver:** Explains how to install the MOLOW driver.
- **Configuring the Device:** Explains how to configure the MOLOW device.
- **Configuring the Driver:** Explains how to configure the communication 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 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 product’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows NT/2000/XP environment. If you are unfamiliar with Windows NT/2000/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 serial communication between the Studio MOLOW driver and devices using the MOLOW protocol.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics
- Conformance Testing

Device Characteristics

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

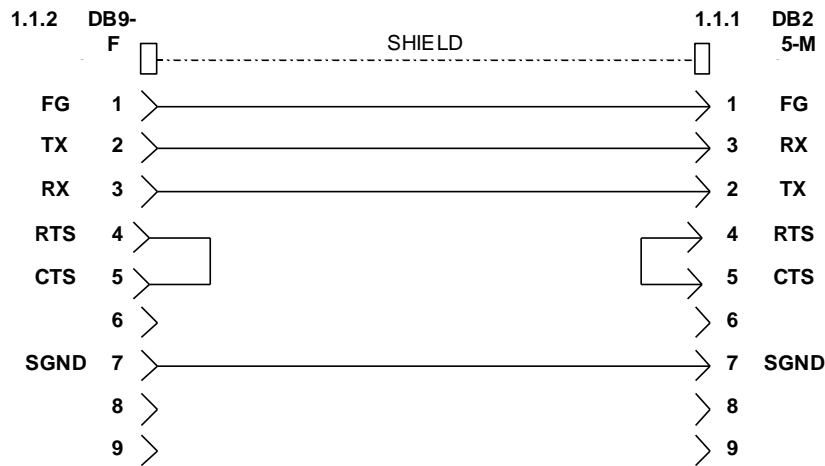
- **Manufacturer:** Modicon or any other device using the MOLOW protocol for serial communication
- **Compatible Equipment:**
 - AEG CPU 984 Series
 - AEG CPU Compact
 - Siemens S7 S200 (using a special ladder program to emulate the MOLOW protocol on the free port)
 - Any device that is compatible with the MOLOW protocol
- **Modicon Programmer Software:** ModSoft

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

Link Characteristics

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

- **Device Communication Port:** MOLOW Port (AEG 985E) and Free Port (S7-S200)
- **Physical Protocol:** RS232
- **Logic Protocol:** MOLOW
- **Device Runtime Software:** None
- **Specific PC Board:** None
- **Cable Wiring Scheme:**




Cable Wiring Scheme

Driver Characteristics

The MOLOW driver is composed of the following files:

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

 **Notes:**

- All of the preceding files are installed in the /DRV subdirectory of the Studio installation directory.
- You must use Adobe Acrobat® Reader™ (provided on the Studio installation CD-ROM) to view the **MOLOW.PDF** document.

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

- Windows 9x
- Windows 2000
- Windows NT

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

The MOLOW driver supports the following registers:

Register Type	Write	Read	Integer	Float	STRING
ID	–	•	–	–	•
START	•	–	–	–	–
STOP	•	–	–	–	–
GET	–	•	•	•	•
SETK	•	–	•	•	–
GETLM	–	•	•	•	•
INIT	•	–	–	–	–
CONFIG	•	–	•	•	–
GETSK	–	•	•	•	•
DPW	•	–	•	•	–
DPR	–	•	•	•	•

Conformance Testing

The following hardware/software was used for conformance testing:

- **Configuration:**
 - **Baud Rate:** 19200
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **COM Port:** COM1
 - **Parity:** even
- **Cable:** Use specifications described in the “Link Characteristics” section.

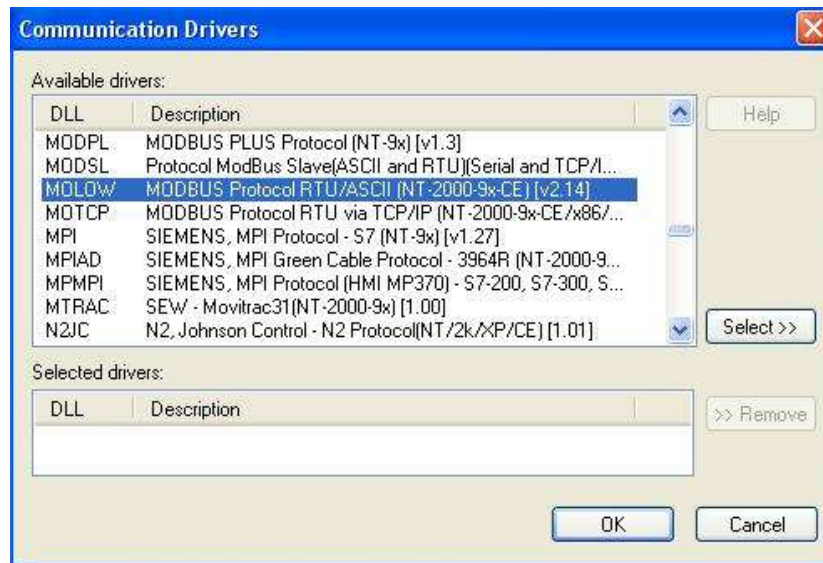
Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.00	6.0	WinXP	WinXP	PLC LCDAS (Low Cost Data Acquisition)

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, or double-click the Studio shortcut icon on your desktop.
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 **MOLOW** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

5. When the **MOLOW** 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 Studio and the device. However, to download the custom program to your device, you must install a MOLOW programmer software package (such as *ModSoft*). Consult the MOLOW documentation for installation instructions.

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 Device

Because there are several brands of equipment that use the MOLOW protocol, it is impossible to define a standard device configuration. Therefore, we suggest using the following default configuration:

- **Baud Rate:** 19200
- **Data Bits:** 8
- **Stop Bits:** 1
- **Parity:** Even

Configuring the Driver

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

- Specifying communication parameters (only one configuration needed).
- Defining communication tags and controls in the Communication tables or *Driver* worksheets (Standard and Main Driver Worksheets).

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 and Main Driver Worksheets, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

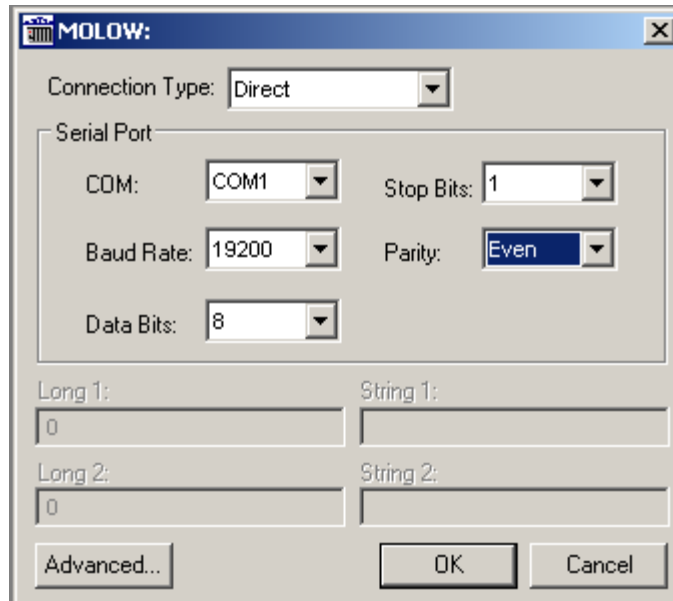
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 *MOLOW* subfolder. When the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu

The *MOLOW: Communication Parameter* dialog displays:

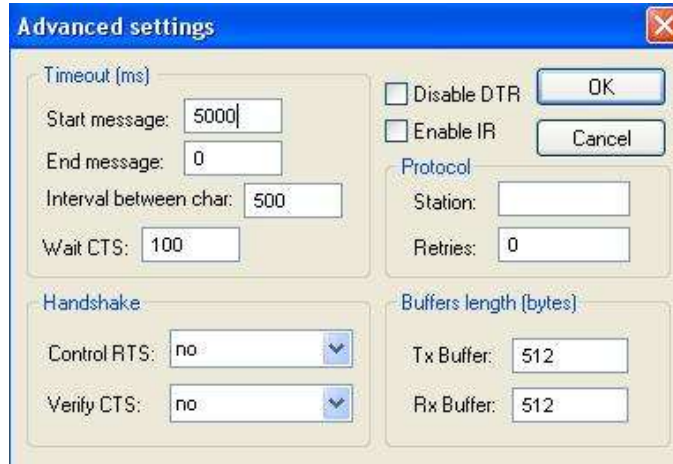


MOLOW: Communication Parameters Dialog

Note:

These parameters must be configured *exactly the same* as those you configured for the MOLOW driver in the *Communications Parameters* dialog.

4. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog:



Advanced Settings Dialog

5. Use the following table to specify the **Control RTS** (*Request to Send*) parameter:

Parameter	Default Value	Valid Values	Description
Control RTS	No	no yes yes + echo	<p>Specify this parameter if the RTS handshake signal is set before communication and if there is an echo in the communication.</p> <ul style="list-style-type: none"> If you are using Windows 95 or CE with the correct RS232–RS485 Converter (without RTS Control), select no. If you are using Windows NT with the Cutler Hammer RS232–485 adapter, you must select yes. <p>Important: Setting this parameter incorrectly prevents the driver from working, and generates a Timeout waiting to start a message error message.</p>



Notes:

- Do not change any of the other default Advanced Settings parameters at this time. You can consult the Studio *Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change these parameters only if you are using a DCE (Data Communication Equipment) converter (such as 232/485), modem, and so forth between your computer, the driver and the host. However, before adjusting the advanced communication parameters, you must be familiar with the characteristics of the DCE.

Configuring the Standard Driver Worksheet

This section explains how to configure a *Standard Driver Worksheet* (or Communication table) to associate application tags with the PLC addresses. You can configure multiple *Driver* worksheets — each of which is divided into a **Header** section and a **Body** section.

Use the following steps to create a new Standard Driver Worksheet:

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 *MOLOW* subfolder.
3. When the pop-up menu displays, 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 *MOLOW.drv* dialog displays (similar to the following figure):

	Tag Name	Address	Div	Add
1	st	1		
2				
3				
4				

MOLOW Driver Worksheet

4. Use the following information to complete the **Station**, **Header** and **Address** fields on this worksheet:
- **Station** field: Use this field to specify the ID (*node*) of the device (*unit name*). Valid values are 0–247 (*no default*).
 - **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 also a reference to the initial address. The default value is **0X:0**.

These variables must comply with the following syntax:

<Type>: <AddressReference> (for example: **START:0**)

Where:

- * **<Type>** is the register type. Valid values: **start, stop, id, init, config, get, setk, getsk, getlm, dpw, dpr**.
- * **<AddressReference>** is the initial address (reference) of the register type you configured.

After you edit the **Header** parameter, the system checks that the syntax is valid. If the syntax is invalid, Studio automatically inserts the default value (**ID:0**) into 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; otherwise, you will get an **Invalid Header** error.

The following table lists all of the data types and address ranges that are valid for the **Header** field:

Header Field Information			
Data Types	Sample Syntax	Valid Range of Initial Addresses	Comments
ID	ID:0	Not used	Returns the equipment's identification
START	START:0	Not used	Starts the LCDAS
STOP	STOP:0	Not used	Stops the LCDAS
GET	GET:0	1-9	Address 9 = Counter (Number of measurements since start)
SETK	SETK:0	1-4	Address 1 = Pressure and Temperature calc Address 2 = P&T wavelength Address 3 = DP wavelength Address 4 = 2P&2T wavelength
GETLM	GETLM:0	1-7	Address 1 = Number of results Address 2 = Number of sensors Address 3 = Time Address 4 = Sensor 1 result Address 5 = Sensor 2 result Address 6 = Sensor 3 result Address 7 = Sensor 4 result
INIT	INIT:0	1-31	Address 01 = P0 Address 02 = TR Address 03 = a11 Address 04 = a12 Address 05 = a21 Address 06 = a22 Address 07 = l1rt1 Address 08 = l2rt1 Address 09 = l1rt2 Address 10 = l2rt2 Address 11 = aa11

Header Field Information			
Data Types	Sample Syntax	Valid Range of Initial Addresses	Comments
			Address 12 = aa12 Address 13 = aa21 Address 14 = aa22 Address 15 = Grad Address 16 = Sens Address 17 = lamMin1 Address 18 = lamMax1 Address 19 = lamMin2 Address 20 = lamMax2 Address 21 = lamMin3 Address 22 = lamMax3 Address 23 = lamMin4 Address 24 = lamMax4 Address 25 = Distance Address 26 = a1 Address 27 = a2 Address 28 = a3 Address 29 = a4 Address 30 = a5 Address 31 = a6
CONFIG	CONFIG:0	Not used	Sets the Start window
GETSK	GETSK:0	Not used	Reads back the calibration data
DPW	DPW:0	1-3	To calculate DP
DPR	DPR:0	Varies according to equipment	Reads DP results

Address field: The body of the *Driver* worksheet allows you to associate each tag to its respective address in the device. 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.

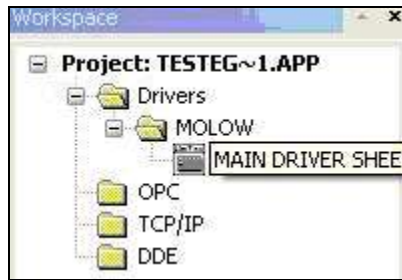
Where:

- * **Signed/Unsigned** (*optional parameter used for integer values only*): Valid values are **s** (signed) and **u** (unsigned).
- * **AddressOffset** is a parameter added to the **AddressReference** parameter (configured in the **Header** field) to compose the address to be read from or written to the device.

➡ Attention:

- Studio stores the floating-point value in two consecutive words. In this case, the **Address** value corresponds to the first Holding Register position. You must not configure a non-existing address, or it will create a conflict.
- In the *Driver* worksheet, the start address is defined by the sum of the **<AddressReference>** parameter (from the **Header** field) and the lowest **<AddressOffset>** parameter (in the **Address** field). The resulting start address *must be a non-zero value*. A zero value will cause a conflict.
- You must not configure a range of addresses greater than the maximum block size (*data buffer length*) supported by each PLC within the same *Driver* worksheet. The maximum data buffer length for this driver is 64 bytes per *Standard Driver Worksheet*.

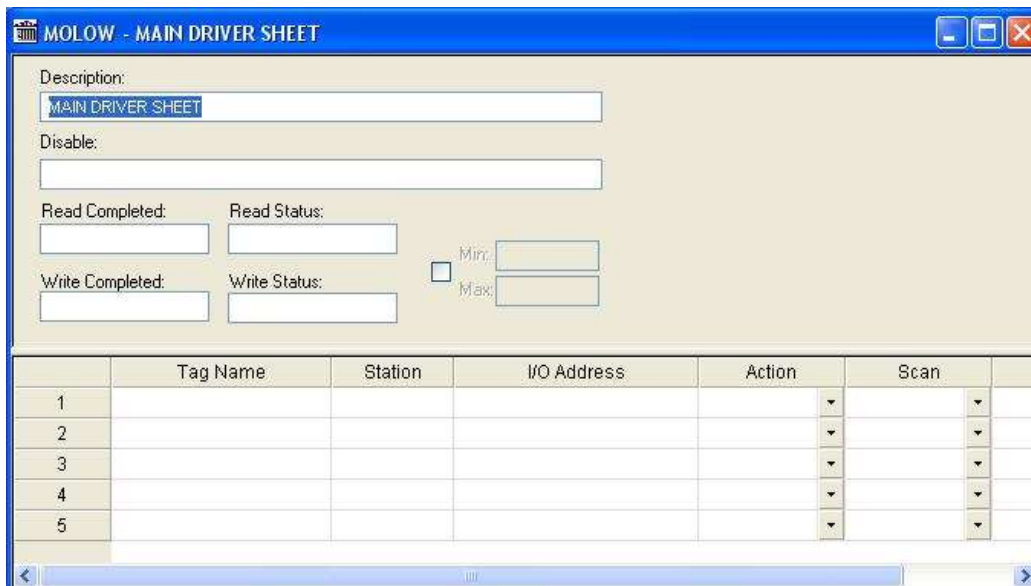
When you add the MOLOW driver to your application, the program automatically adds the *MAIN Driver Worksheet (MDS)* to the *MOLOW* driver folder as shown:



You can use the MDS to associate Studio tags to addresses in the PLC. Most MDS parameters are standard for any driver, and are not discussed in this publication. For information about configuring these parameters, consult the *Studio Technical Reference Manual*.

Use the following instructions to configure the parameters that are specific to the MOLOW driver:

1. Double-click on the *Main Driver Sheet* icon to open the following worksheet:



2. Configure the following fields on this worksheet:

- **Station** field: Type the PLC address ID number
- **I/O Address** field: Type the address of each PLC register, using the following syntax:

<Type><Address>: 0 (for example, **Id: 0**) or

Where:

- * **<Type>** is the register type. Valid values: **start, stop, id, init, config, get, setk, getsk, getlm, dpw, dpr**.
- * **<Address>** is the register address of the device.

➔ **Attention:**
 You must use a non-zero value in the **Station** field, and you cannot leave the field blank.

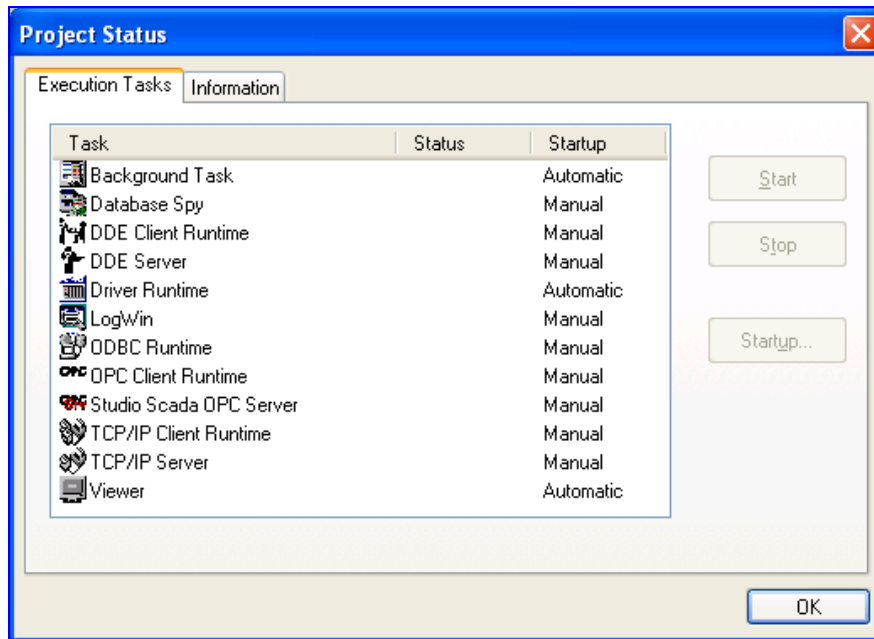
Executing the Driver

After adding the MOLOW 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 displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
 - If the setting is correct, click **OK** to close the dialog.
 - 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 MOLOW 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 the failure that occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
10	Invalid Header field	Specified invalid tag value in Header field	Specify a valid Header tag value.
11	Invalid Address field	Specified invalid Address	Specify a valid address.
12	Invalid block size	Offset greater than maximum allowed	Specify a valid offset or create a new worksheet. Typically, maximum offset is 64.
13	Invalid CRC	Invalid CRC in response message	<ul style="list-style-type: none"> ▪ Check the cable wiring. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see <i>Studio Technical Reference Manual</i> for valid configurations).
18	Invalid BCD Value	Tried reading an invalid BCD value	Check the value on PLC to make sure that it is a valid value.
19	Invalid BCD Value	Tried writing a negative BCD value	Only positive BCD values can be written.
20	Illegal function	Wrong sub-code, or the function might be already running	Check to see if the equipment is already running (started).
21	Slave Device Failure	Hardware problem	Contact the manufacturer.
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected cables ▪ PLC is turned off, in stop mode or in error mode. ▪ Wrong station number ▪ Wrong RTS/CTS control settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see the <i>Studio Technical Reference Manual</i> for valid configurations).
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC is in stop mode or in error mode. ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see “Link Characteristics” for valid RTS/CTS configurations).

⇒ **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 enable the log at the unit (**Tools** → **LogWin**) and verify the `ce1log.txt` file created at the target unit.

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**.
- **Project Information**: To find this information, select **Project** → **Status**.
- **Driver Version** and **Communication Log**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application in the `/COMMUNICATION EXAMPLES/MOLOW` directory. We strongly recommend that you use this sample application to test the MOLOW driver before configuring your own application, for the following reasons:

- To better understand the information provided in each section of this document.
- To verify that your configuration is working satisfactorily.
- To certify that the hardware used in the test (device, adapter, cable and PC) is working satisfactorily before you start configuring your own, customized applications.



Note:

This application sample is not available for all drivers.

Use the following procedure to perform the test:

1. Configure the device's communication parameters using the manufacturer's documentation.
2. Open and execute the sample application.
3. Execute the *Viewer* module in Studio to display information about the driver communication.



Tip:

You can use the sample application screen as the maintenance screen for your custom applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Fabio Carvalho	May/30/2005	First driver version
B	1.1	Paulo Balbino	Sept/16/2011	Fix Ini version and Doc