

MBLAU Communication Driver

Driver for Serial Communication
with devices using Microblau Protocol

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	7
DEVICE CONFIGURATION	12
EXECUTING THE DRIVER	13
TROUBLESHOOTING	14
SAMPLE APPLICATION	16
REVISION HISTORY.....	17

Introduction

The MBLAU driver enables communication between the Studio system and devices using the Microblau protocol communicating over Serial, according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the MBLAU driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the MBLAU driver.
- **Installing the Driver:** Explains how to install the MBLAU driver.
- **Configuring the Driver:** Explains how to configure the MBLAU 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 MBLAU 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 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 communication between the Studio MBLAU driver and the Microblau controller.

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:** Microblau Indústria Eletrônica
- **Compatible Equipment:**
 - TD3000 and TD300
 - Any controller that is fully compatible with the Microblau protocol
- **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:** Serial Port (RS 485)
- **Physical Protocol:** Serial
- **Logic Protocol:** Microblau
- **Specific PC Board:** None

Driver Characteristics

The MBLAU driver is composed of the following files:

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

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

- Windows NT/2K/XP
- Windows CE

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

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration:**
 - **COM Port:** COM1
 - **Baud Rate:** 19200
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **Parity:** Even
 - **Station:** 99

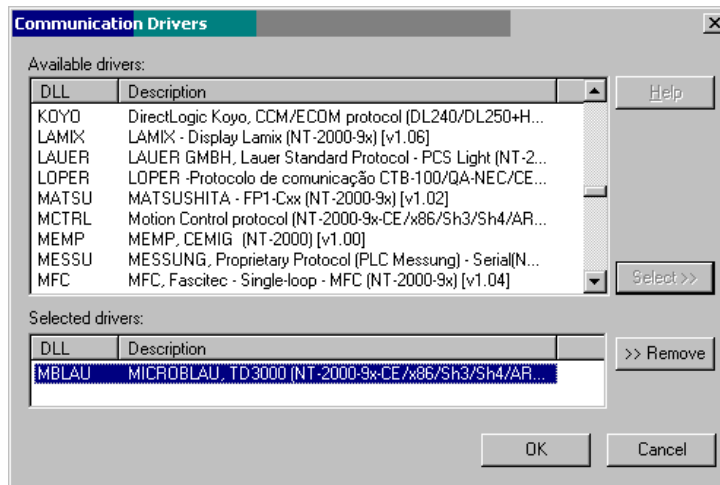
Driver Version	Studio Version	Operating System	Equipment
2.00	3.0	WinXP	TD3000

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 MBLAU driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

5. When the MBLAU 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 MBLAU driver, you must configure the driver. Configuring the MBLAU driver is done in two parts:

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

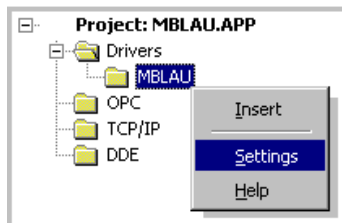
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**,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

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

Setting the Communication Parameters

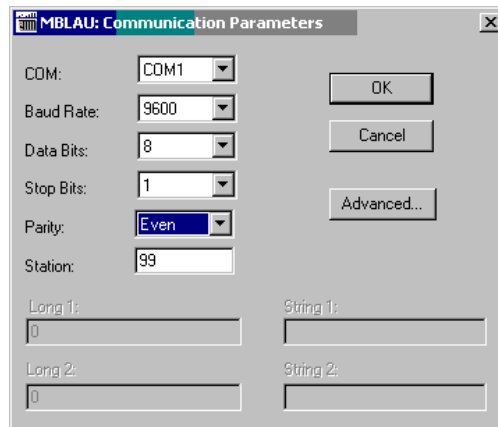
Use the following steps to configure the communication parameters, which are valid for all *Driver* worksheets configured in the system:

1. From the Studio development environment, select the **Comm** tab located below the *Workspace* pane.
2. Click on the *Drivers* folder in the *Workspace* pane to expand the folder.
3. Right-click on the *MBLAU* subfolder and when the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu

The *MBLAU: Communications Parameters* dialog displays:



Communication Parameters Dialog

4. Specify the custom parameters as noted in the following table:

Parameters	Default Values	Valid Values	Description
Station	–	99	Computer Address



Note:

The device must be configured with *exactly the same* parameters that you configured in the *MBLAU Communication Parameters* dialog.

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



Notes:

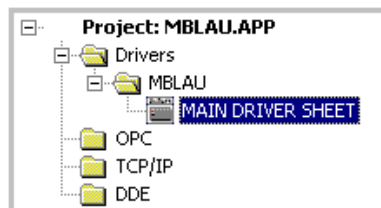
- Do not change any of the other *Advanced* parameters at this time. You can consult the *Studio Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem, and so forth between the PC, driver, and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

Configuring the Driver Worksheets

This section explains how to configure the *MAIN* and *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.

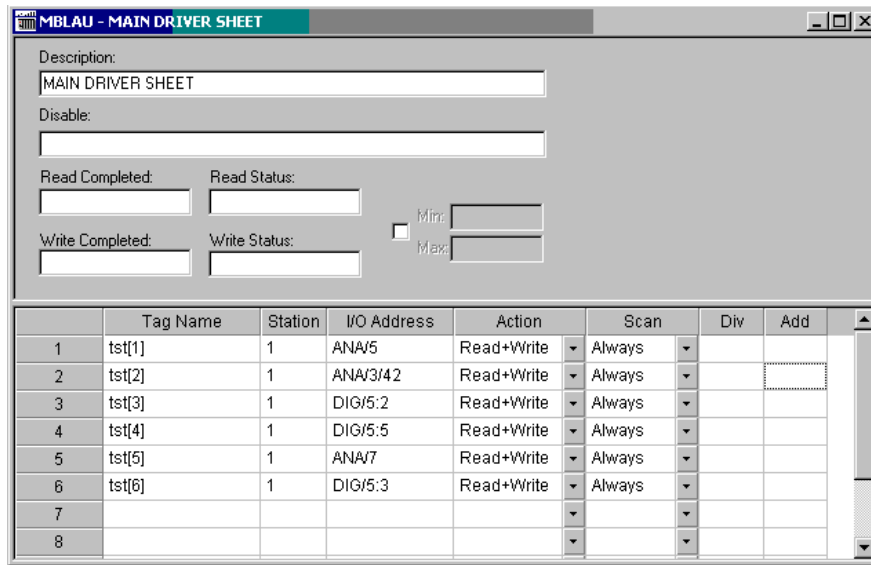
Configuring the MAIN DRIVER SHEET

When you add the MBLAU driver to your application, Studio automatically adds a *MAIN DRIVER SHEET* to the driver folder, as shown:



MAIN DRIVER SHEET

You use this worksheet (similar to the following figure) to associate Studio tags to addresses in the PLC:



MAIN DRIVER SHEET



Note:

Most of the *MAIN DRIVER SHEET* parameters are standard for all drivers, and are not discussed in this document. Instructions for configuring these standard parameters are provided in the Studio *Technical Reference Manual*.

Use the following information to configure the **Station** and **Address** parameters specific to this driver:

- **Station:** Type the MicroController Address (ID number).
- **I/O Address:** Type the address of each register in the controller using the following syntax (for Input and Outputs):

<Header>/<Address>

Where:

- **Header:** Operation types and the complement, if necessary, in accordance with the table below:

Operation Types	Comments
01	Read command of analog registers
02	Write command of analog registers
03	Read command of digital registers
04	Write command of digital registers
06:<Line>	Write command of line of hourly programming
07:<Line>	Read command of line of hourly programming
P:<Line>	Write command of line of hourly programming; possible to change the dot, command, day, hour, and minute in Runtime mode
ANA	Read command and Write command of analog registers
DIG	Read command and Write command of digital registers

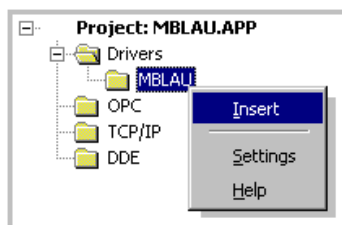
- **Address:** Complement of the operation type specified in the Header, in accordance with the following table:

Header	Address	Description
01	<register>	Read register
02	<register>	Write register
03	<block>:<register>	Read block and register
04	<block>:<register>	Write block and register
06:<Line>	<dot>:<cmd>:<day>: <hour>:<min>	<dot> : Activation dot (1..127) <cmd> : Command on dot (0..1) <day> : Day (0..7) <hour> : Hour (0..23) <min> : Minute (0..59)
07:<Line>	<register>	Read of the information type of line of hourly programming. Valid information type: 1 : Activation dot 2 : Command on dot 3 : Day 4 : Hour 5 : Minute
P:<Line>	0..4	0 : Activation dot (1..127) 1 : Command on dot (0..1) 2 : Day (0..7) 3 : Minute (0..59) 4 : Hour (0..23)
ANA	<register>	Read and/or write registers
DIG	<block>:<register>	Read and write block and register

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, select the **Insert** option:

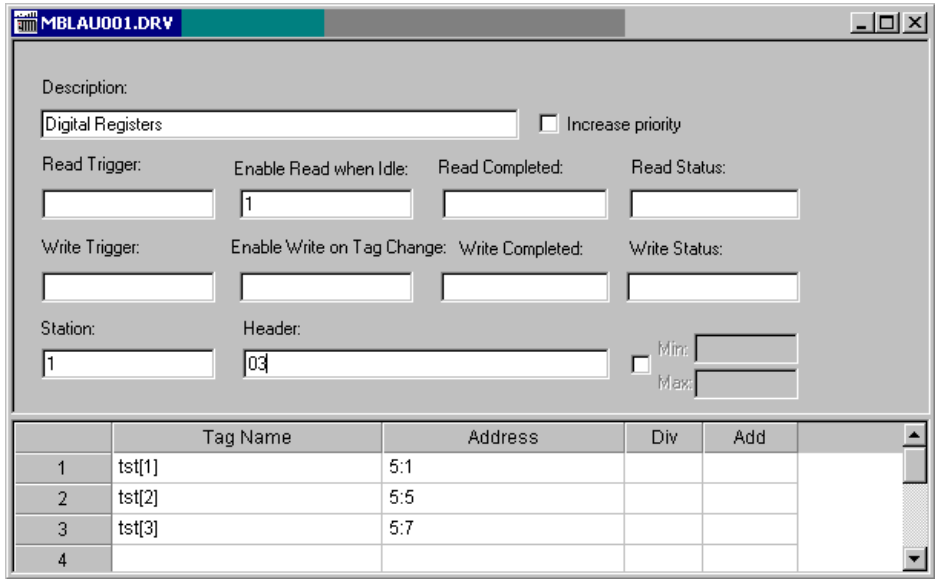


Inserting a New Worksheet

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):



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: **Station**: Type the MicroController Address (ID number)
 - **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.

<Operation Type>

Operation Types	Comments
01	Read command of analog registers
02	Write command of analog registers
03	Read command of digital registers
04	Write command of digital registers
06: <Line>	Write command of line of hourly programming
07:<Line>	Read command of line of hourly programming
P:<Line>	Write command of line of hourly programming, possible the change of the dot, command, day, hour, and minute in Runtime mode.
ANA	Read command and write command of analog registers
DIG	Read command and write command of digital registers

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.

- **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:

Header	Address	Description
01	<register>	Read register
02	<register>/<Code Format>	Write register Code format: Number of the decimal house (optional) Valid Code: 10 – Y 20 – YY 11 – Y,Y 40 – YYYY 31 – YYY,Y 60 – YYYYYY 51 – YYYYY,Y 42 – YYYYY,YY
03	<block>:<register>	Read block and register
04	<block>:<register>	Write block and register
06:<Line>	<dot>:<cmd>:<day>: <hour>:<min>	<dot> : Activation dot (1..127) <cmd> : Command on dot (0..1) <day> : Day (0..7) <hour> : Hour (0..23) <min> : Minute (0..59)
07:<Line>	<register>	Read of the information type of line of hourly programming. Valid information type: 1 : Activation dot 2 : Command on dot 3 : Day 4 : Hour 5 : Minute
P:<Line>	0..4	0 : Activation dot (1..127) 1 : Command on dot (0..1) 2 : Day (0..7) 3 : Minute (0..59) 4 : Hour (0..23)
ANA	<register>	Read and/or write registers
DIG	<block>:<register>	Read and write block and register

Sample Address	
Header Field	Address Field
01	0
	10
	20
02	0
	10/42
	20/11
03	5:2
	5:5
	5:7
04	5:2
	5:5
	5:7
06:3	120:1:5:20:15
	3:1:0:12:25
	7:0:1:5:40
07:12	3
	5
	2
P:1	0
	1
	4
ANA	0
	10
	20
DIG	5:2
	5:5
	5:7

Device Configuration

Because there are multiple devices that use the Modbus protocol, we cannot define a standard device configuration.

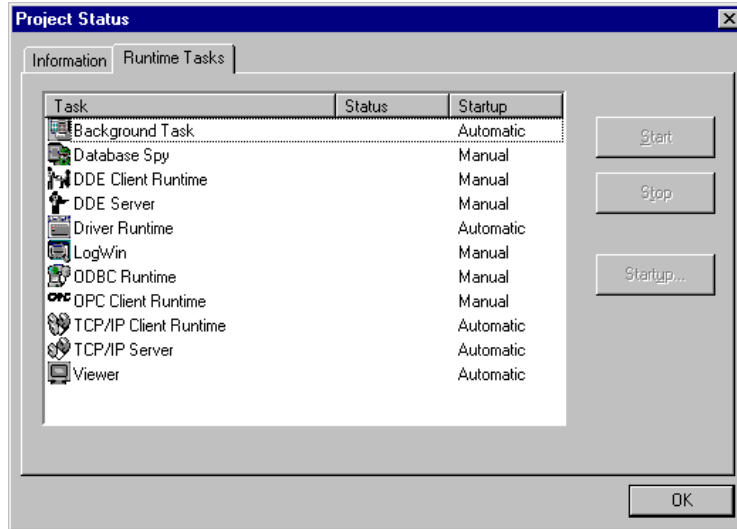
Executing the Driver

After adding the MBLAU 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 MBLAU 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.
2	Invalid Header	Specified an invalid header.	Type a valid header.
3	Invalid Address	Specified an invalid address. The data address received in this query was not an allowable address for the slave.	Type a valid address.
5	Protocol Error	Invalid message received.	<ul style="list-style-type: none"> ▪ Check the equipment state. ▪ Check the communication parameters for the device and the Studio software.
6	Checksum Error	Invalid checksum in response message.	<ul style="list-style-type: none"> ▪ Check the cable wiring. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see Studio <i>Technical Reference Manual</i> for valid configurations).
8	Invalid Slave Address	Wrong station address.	Check the station address.
9	Invalid Master Address	Wrong computer address.	Check the computer address.
10	Answer Unknown	Invalid message received.	<ul style="list-style-type: none"> ▪ Check the equipment state. ▪ Check the communication parameters for the device and the Studio software.
11	Minimum Parameters Error	Parameters number configured in Address is wrong.	Check the Address configuration.
12	Invalid Read Command	Invalid Read Command.	Check if the header is valid to read.
14	Invalid Write Error	Invalid Write Command.	Check if the header is valid to write.
15	Field Out of Range	A parameters configured in Address is Out of Range.	Check the valid value.
-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 configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC 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 configuration. See <i>Studio Technical Reference Manual</i> for information about 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 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 in the *Communication Drivers* dialog.
- **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

You will find a sample application for drivers in the /COMMUNICATION EXAMPLES/<Driver Name> directory. We strongly recommend that you check if there is a sample application for this driver and use it to test the driver before configuring your own customized application, for the following reasons:

- To better understand the information provided in the 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.

 **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	2.00	Eric Vigiani	Jan/06/2004	Modified to be compatible with others devices. Included MDS (Main Driver Sheet)