

TXRX Communication Driver

Driver for Sending Data to or Receiving Data from
a Serial Port or a TCP/IP Connection

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 WORKSHEET 8

 CONFIGURING THE DEVICE 10

EXECUTING THE DRIVER 11

TROUBLESHOOTING 12

SAMPLE APPLICATION 13

REVISION HISTORY 14

Introduction

The TXRX driver enables communication between the Studio system and devices using the ASCII protocol, according to the specifications discussed in this document.

This document was designed to help you install, configure and use the TXRX driver to send data to or receive data from a Serial port or TCP/IP connection. The information in this document is organized as follows:

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

The information is organized into the following sections:

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

Device Characteristics

You can use this driver with any device that exchanges data using a simple ASCII protocol. (The devices used for conformance testing are listed in “Conformance Testing” on page 4.)

Link Characteristics

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

- **Device Communication Port:** RS232 or Ethernet port
- **Physical Protocol:** RS232/RS485 or TCP/IP
- **Logic Protocol:** ASCII
- **Device Runtime Software:** None
- **Specific PC Board Adapters:** None
- **Adapters/Converters:** Device-dependent
- **Cable Wiring:** Device-dependent

Driver Characteristics

The TXRX driver is composed of the following files:

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

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

- Windows 9x
- Windows 2000
- Windows NT
- Windows XP
- Windows CE

(The operating systems used for conformance testing are listed in “Conformance Testing” on page 4.)

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration (a):**
 - PLC Program: None
 - Baud Rate: 9600
 - Protocol: ASCII
 - Data Bits: 7
 - Stop Bits: 1
 - Parity: None
 - COM Port: 1
 - TCP/IP Port: 0
 - Cable: Serial Cross-over Cable

Driver Version	Studio Version	Operating System (development)	Operating System (runtime)	Equipment
1.0	4.1	Windows NT	Windows NT/CE	<ul style="list-style-type: none"> – Two PCs connected with a Serial cable – Two PCs connected with a TCP/IP connection

- **Driver Configuration (b):**
 - PLC Program: None
 - Baud Rate: Not used
 - Protocol: ASCII
 - Data Bits: Not used
 - Stop Bits: Not used
 - Parity: Not used
 - COM Port: Not used
 - TCP/IP Port: 1234
 - Cable: Ethernet Cable

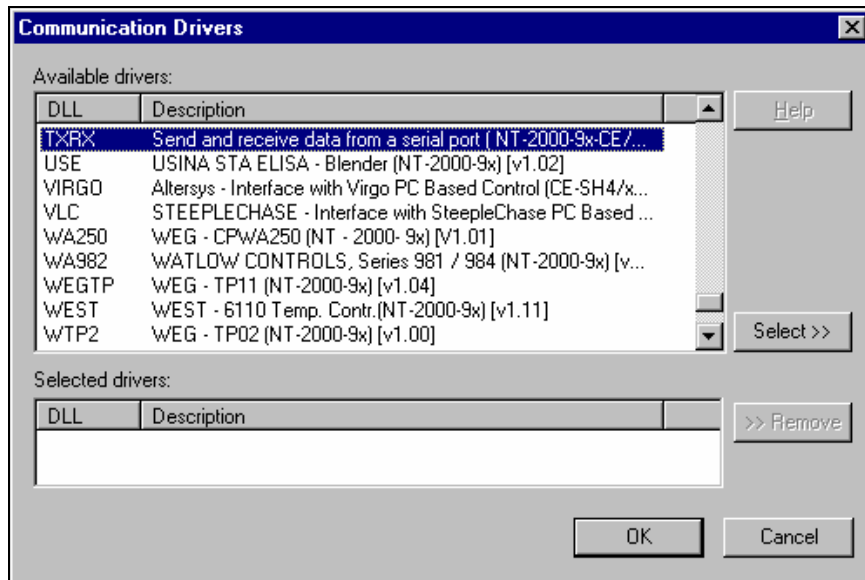
Driver Version	Studio Version	Operating System	Equipment
1.03	4.1	Windows 2000 + SP1	<ul style="list-style-type: none"> – Two PCs connected with a Serial cable – Two PCs connected with a TCP/IP connection

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 **TXRX** 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 **TXRX** 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.



Caution:

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 TXRX driver, you must configure the driver. Configuring the TXRX driver is done in two parts:

- Specifying communication parameters
- Defining communication tags and controls in the Communication tables or Driver Worksheet

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.

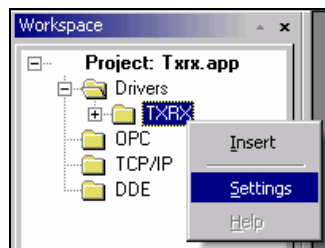
 **Note:**

For a detailed description of the Studio Standard Driver Worksheets, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

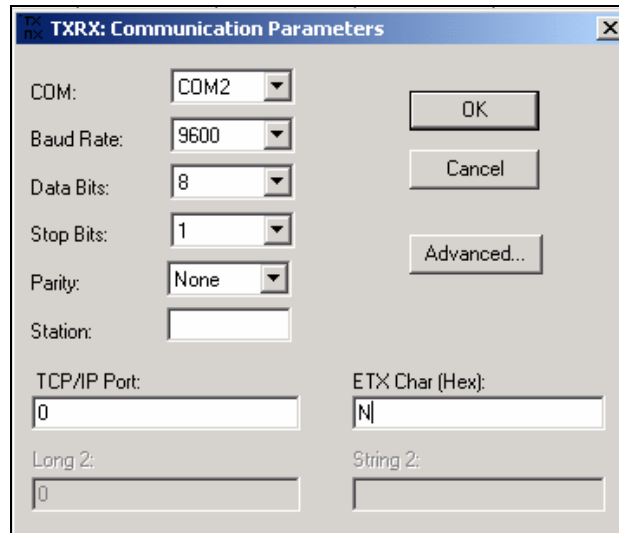
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 *TXRX* subfolder. When the pop-up menu displays (as shown in the following figure), select the **Settings** option.



Select Settings from the Pop-Up Menu

The TXRX: Communication Parameters dialog displays (as follows).



TXRX: Communication Parameters Dialog

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

Parameters	Default Values	Valid Values	Description
COM	COM2	COM1 to COM8	Serial port of the PC used to communicate with the device
Baud Rate	9600	110 to 57600 bps	Data communication rate
Data Bits	8	5 to 8	Number of data bits used in the protocol
Stop Bits	1	1 or 2	Number of stop bits used in the protocol
Parity	None	even, odd, none, space, or mark	Protocol parity
Station	0	0	Not used for this driver
TCP/IP Port	0	Integer value	TCP/IP port number used to receive messages from other devices. Important: If you want to use Serial communication, you must specify zero (0) for this parameter.
ETX Char (Hex)	0	0-FF or N (none)	ETX character indicates the end of a message. You can specify multiple ETX characters using the following syntax: <1st ETX>, <2nd ETX>, <3rd ETX> ... For example, you can specify 0D, which is a carriage return in ASCII. Configure an N value if you do not want to use the ETX character.

Notes:

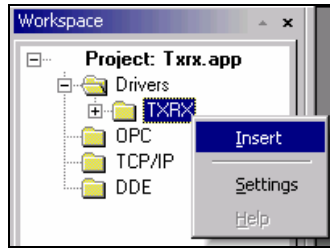
- If you specify an ETX character in the **ETX Char (Hex)** field, Studio will put that character at the end of every message in a Write command.
- Clicking the **Advanced** button opens the *Advanced Settings* dialog, which contains some additional communication parameters. Although you *should not change the default settings at this time*, you can consult the *Studio Technical Reference Manual* for information about configuring these parameters for future reference.

Configuring the 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:

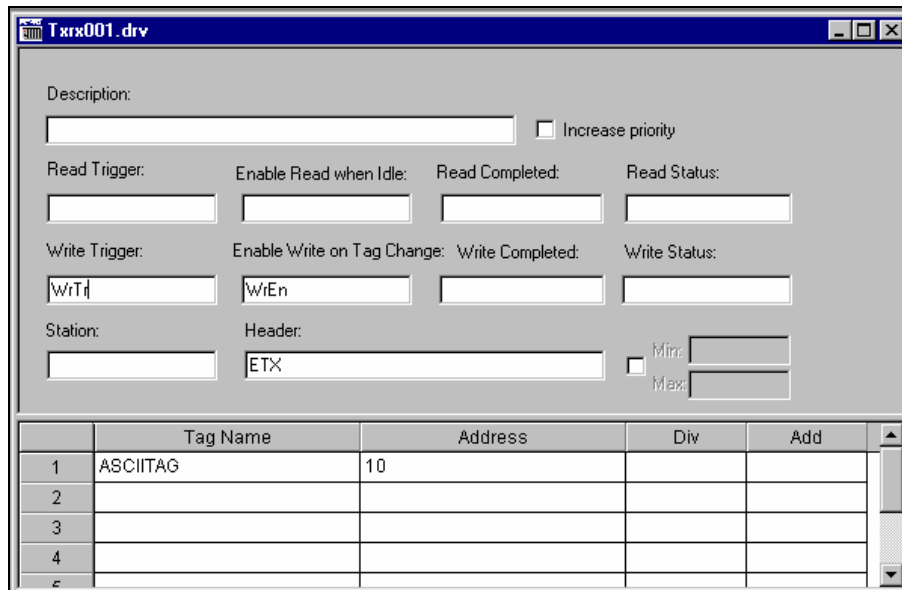
1. From the Studio development environment, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder, and right-click the *TXRX* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



Inserting a New Worksheet

Note:
 This driver does not use Read fields; instead, the driver uses incoming messages to receive data from the device.

The *Txx001.drv* dialog box displays (similar to the following figure).



TXRX Driver Worksheet

All parameters on the Driver Worksheet (except the **Station**, **Header** and **Address** fields) are standard for all communication drivers and 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 (*not used for Serial communication*): Specify the target device's IP address and TCP/IP port to which you send messages. (There is no default value.) This parameter must comply with the following syntax:

<IP Address>:<TCP/IP port> (For example: **192.168.2.15:1234**)

Where:

- **<IP Address>** is the IP address of the target device (For example: **192.168.2.15**). *You specify this parameter for Write commands only.*
- **<TCP/IP port>** is the TCP/IP port to which you send messages. (For example: **1234**)
- **Header** field: Define the specific incoming and outgoing messages to be sent from the device.

You can specify two types of headers for incoming messages:

- **RXn**: Writes incoming messages to a specified tag when *n* characters arrive.
- **ETX**: Writes incoming messages to a specified tag only when one of the **ETX** characters you configured for the driver arrives.


To specify a header for outgoing messages:

- **TX:<ETX char (Hex value)>**: Sends outgoing messages to the device. Messages consist of a specified tag in ASCII format along with the specified **ETX** character.

For example:

- TX** sends messages only. If you do not specify an **ETX** character in the **Header**, Studio automatically uses the **ETX** character from the **Settings** field for Write commands. (For example: **TX:0D**)
- TX:0A** sends messages only. Studio uses the **0A** character (as the **ETX** character) at the end of the outgoing message.
- RX10** receives incoming messages of ten (10) characters.
- ETX** receives incoming messages ending with one of the **ETX** characters you specified in the *Communication Parameters* dialog.

You can type {**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.

 **Note:** Studio transfers incoming messages to tags in the tag fields as strings.

- **Address field:** Use the information provided in the following table to divide each message into its respective component parts.
 - 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.
 - For Write commands, the address must specify the number of characters that can be sent for each line of the worksheet. The maximum value per line is 80 characters.
 - For incoming messages or responses, use the following syntax
L:<Message Length>, S:<Separator Char (Hex Value)> or <Any Integer Value>
(For example: w10.2)

Where:

- **<Message Length>** is the number of characters (starting with the last character defined in the Driver Worksheet) the address tag will receive.
- **<Separator Char>** is an ASCII character used to end and separate messages; for example, **s : 41**.
- **<Any Integer Value>** is the number of characters (starting with the last character defined in the Driver Worksheet) the address tag will receive.

Address Configuration Sample		
Address on the Device	Header Field	Address Field
"1234567890"	ETX	L:10
"DriverSheet"	ETX	S:20
"Example"	ETX	S:20
"1234567890"	ETX	L:10

Configuring the Device

You must use the same configuration parameters for the device that you specified for the TXRX driver in the Communication Parameters dialog.

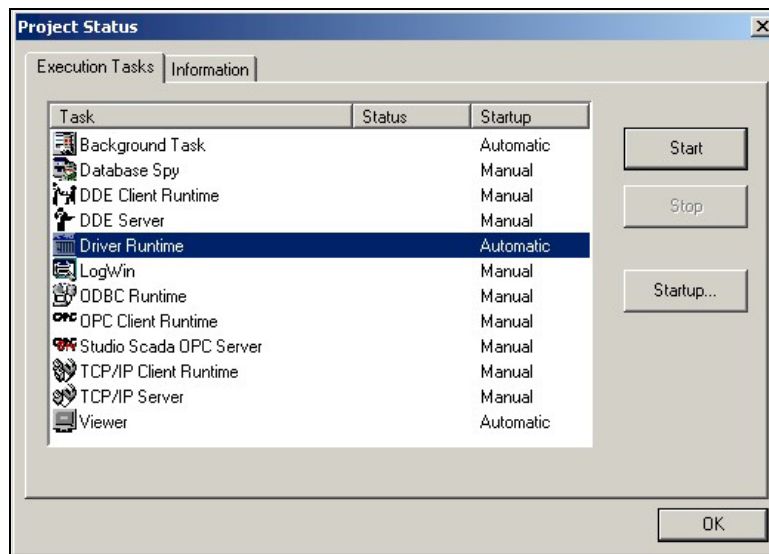
Executing the Driver

After adding the TXRX 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 TXRX 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	Communicating without problems	Not required
2	Failure to allocate memory	Not enough memory to allocate the driver buffer	Increase the system memory.
10	Invalid Header	Invalid Header provided or tag in the Header field has an invalid configuration	Type a valid Header or a valid tag value in the Header field. Consult the Studio <i>Technical Reference Manual</i> for a list of valid headers.
16	Invalid command	Invalid command specified in the Driver Worksheet	Disable all Read fields in the Driver Worksheet.
17	Invalid ETX	Invalid ETX value specified	Type a valid ETX value (between 0 and FF).
18	Error connecting	Invalid IP address or TCP/IP port specified	Type a valid IP Address (TCP/IP) in the Station field or for the tag value.

⇒ **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/TXRX` directory. We *strongly* recommend that you use this sample application to test the TXRX driver before configuring your own customized 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.

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	Lourenço Teodoro	9 Mar 2001	Initial version
B	1.01	Lourenço Teodoro	27 Apr 2001	Improved unsolicited messages treatment
C	1.02	Lourenço Teodoro	19 Nov 2001	<ul style="list-style-type: none">▪ Fixed the lock-up bug▪ Implemented the receiving feature with more than one line in the address
D	1.03	Robert Vigiani Jr.	5 Dec 2001	<ul style="list-style-type: none">▪ Included TCP/IP communication▪ Included configuration of ETX char in the send messages▪ Included message treatment by length or by separator char
E	1.04	Lourenço Teodoro	3 May 2002	The parameter Interval Between RX char is used to reset the RX buffer. The previous version had a fixed time of 500ms.
F	1.05	Lourenço Teodoro	9 Oct 2002	Fixed bug when using Interval Between RX char with TCP/IP
G	1.06	Bryan Morgan	14 Feb 2003	Fixed bug preventing the driver from using the ETX char when sending data
H	1.07	Lourenço Teodoro	01 Dec 2003	Fixed send problem when the driver was compiled for UNICODE. This problem could happen under Windows CE.
I	1.08	Bruno A. Crepaldi	12 Nov 2004	Fixed the problem of memory leak
J	1.09	Leandro Coeli	06 Sep 2005	Fixed the ETX problem