

**TWCAT Communication Driver**

Driver for TCP/IP Communication  
with TWINCAT PLC Using TwinCAT ADS Library

**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

**EXECUTING THE DRIVER ..... 13**

**TROUBLESHOOTING ..... 14**

**SAMPLE APPLICATION ..... 16**

**REVISION HISTORY ..... 17**

## Introduction

The TWCAT driver enables communication between the Studio system and TwinCAT PLC using the TwinCAT ADS Interface PLC over TCP/IP, according to the specifications discussed in this document.

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

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

The information is organized into the following sections:

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

### Device Characteristics

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

- **Manufacturer:** Beckhoff
- **Compatible Equipment:** Beckhoff TwinCAT Software
- **Programmer Software:** Beckhoff TwinCAT (System Manager and PLC Control)

### Link Characteristics

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

- **Device Communication Port:** TwinCAT TCP/IP Port
- **Physical Protocol:** Ethernet/TCP/IP
- **Logic Protocol:** Proprietary
- **Device Runtime Software:** TwinCAT Software
- **Third Party Library:** TwinCAT ADS Interface PLC Software (`tcAdsD11.dll` or `tcAdsD11CE.dll`)
- **Specific PC Board:** Any TCP/IP Adapter (Ethernet Board)

### Driver Characteristics

The TWCAT driver is composed of the following files:

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

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

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

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

The TWCAT driver supports the following flair registers:

Register Type	Length	Write	Read	Bit	Integer	Float	DWord
%M	1 Byte	•	•	•	•	•	•
%MX	1 Byte	•	•	•	–	–	–
%Q	1 Byte	•	•	•	•	•	•
%QX	1 Byte	•	•	•	–	–	–
%I	1 Byte	•	•	•	•	•	•
%IX	1 Byte	•	•	•	–	–	–
NAME	–	•	•	•	•	•	•
REFRESH	–	–	•	–	–	–	–

### **Conformance Testing**

The following hardware/software was used for conformance testing:

- **Driver Configuration:** PLC Program Machine
- **Cable:** Ethernet Cable

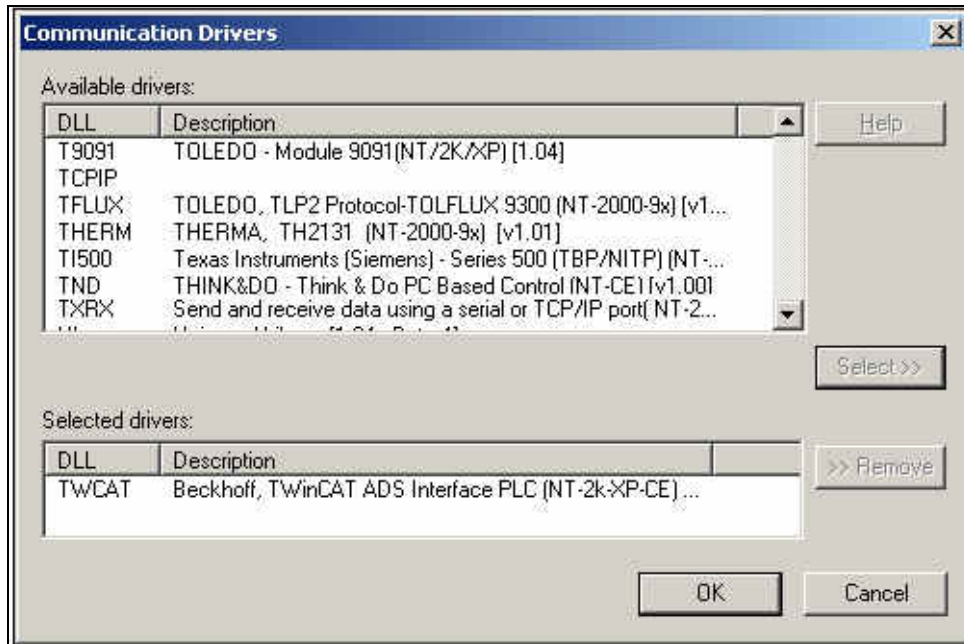
Driver Version	Studio Version	Operating System (Development)	Operating System (Target)	Equipment
1.02	6.0	Windows XP + Service Pack 2	<ul style="list-style-type: none"> <li>▪ Windows XP + Service Pack 2</li> <li>▪ Windows CE v4.2</li> </ul>	<b>Equipment:</b> Beckhoff TwinCAT PLC

## 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 **TWCAT** driver from the *Available Drivers* list, and then click the **Select** button:



*Communication Drivers Dialog*

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

**Note:**

It is necessary to install the TwinCAT ADS Interface PLC 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 TWCAT driver, you must configure the driver. Configuring the TWCAT driver is done in two parts:

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

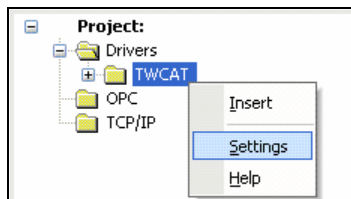
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**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

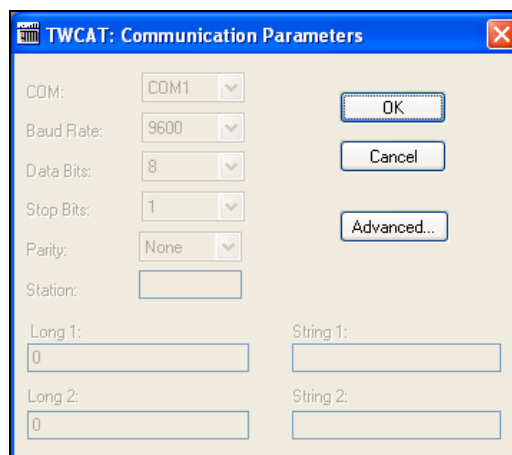
Use the following steps to configure the communication parameters, which are valid for all *Driver* worksheets configured in the system:

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 *TWCAT* subfolder and when the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu

The *TWCAT: Communication Parameters* dialog displays:



Communication Parameters Dialog

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

Parameters	Default Values	Valid Values	Description
Station	0	0	Not used for this driver

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

**Notes:**

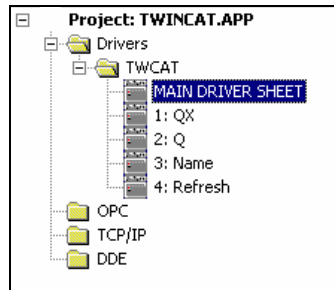
- Do not change any of the other *Advanced* parameters at this time. You can consult the Studio *Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem and so forth between the PC, the 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 TWCAT driver to your application, Studio automatically adds a *MAIN DRIVER SHEET* to the driver folder, as shown in the following figure:



**Main Driver Sheet**

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

Description: <input type="text" value="MAIN DRIVER SHEET"/>							
Disable: <input type="text"/>							
Read Completed:		Read Status:					
<input type="text"/>		<input type="text"/>					
Write Completed:		Write Status:		<input type="checkbox"/> Min: <input type="text"/> <input type="checkbox"/> Max: <input type="text"/>			
<input type="text"/>		<input type="text"/>					
	Tag Name	Station	I/O Address	Action	Scan	Div	Add
1	Data[1]	192.168.1.72.1.1:1	%Q0:B10	Read+Write	Always		
2	Data[2]	192.168.1.72.1.1:1	%Q0:W3	Read+Write	Always		
3	Data[3]	192.168.1.72.1.1:2	.switch	Read	Always		
4	Data[4]	192.168.1.72.1.1:2	.devSpeed	Read+Write	Always		
5	Data[5]	192.168.1.53.1.1:2	%M10:F4	Read+Write	Always		
6	Data[6]	192.168.1.53.1.1:1	.timerUp	Read+Write	Always		
7	Data[7]	192.168.1.53.1.1:1	.timerDown	Read+Write	Always		
8	Data[8]	192.168.1.53.1.1:1	.engine	Read+Write	Always		
9	Data[9]	192.168.1.64.1.1:4	.deviceUp	Read+Write	Always		
10	Data[10]	192.168.1.64.1.1:4	.DeviceDown	Read+Write	Always		
11							

**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 PLC AmsNetID Address (ID number) and the Runtime number or station port.  
 <AmsNetID>:<Runtime or Port Number> (for example: 192.168.1.72.1.1:1 or 192.168.1.72.1.1:301)

Where:

- <AmsNetID>: AmsNet ID of the PLC.
- <Runtime>: Runtime number of PLC program.
- <Port Number>: PLC/NC/IO port used to do the communication.

- **I/O Address:** Type the address of each register in the PLC using the following syntax (for Input and Outputs):  
 <Type>[<StartAddress>:<Datatype>]<Offset>. [Bit] (for example, %Q0:W0.0, %QX0.0, .engine)

Where:

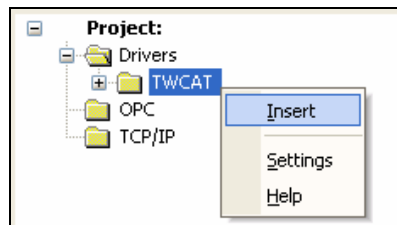
- <Type>: Register type (%M, %MX, %Q, %QX, %I, %IX, <Variable Name>). When the <Variable Name> is configured, the **Datatype** is populated automatically. The variables declared in the “Global Variables” need to use the prefix “.” (for example, .engine, .speed).
- <StartAddress> (optional): Start address of the device’s register. This value must be configured only with the %Q, %I and %M operands. When this value is configured, the **Datatype** must be configured too.

- **<Datatype>** (*optional*): Format of data – **B** (byte), **W** (word), **D** (double word) and **F** (float). It must be used only with the **%Q**, **%I** and **%M** operands. When the operand is **%QX**, **%MX** or **%IX**, the data type is **bit** by default.
- **<Offset>**: This value is the offset of the **StartAddress**. When the **StartAddress** is not configured, the value is 0 (zero) by default.
- **<Bit>** (*optional*): Indicates the bit number to be read from/written to the device.

### 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 *TWCAT* 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.



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 TWCAT driver.

Header Field Information			
Data Types	Sample Syntax	Valid Range of Initial Addresses per Worksheet	Comments
<b>%M</b>	%M:0	Varies according to the equipment	Memory: Read and write data value in the memory area. The data type can be BYTE, WORD, DWORD or FLOAT.
<b>%MX</b>	%MX:0	Varies according to the equipment	Memory: Read and write data value in the memory area. The data type can be only BIT.
<b>%Q</b>	%Q:0	Varies according to the equipment	Output: Read and write data value in the outputs. The data type can be BYTE, WORD, DWORD or FLOAT.
<b>%QX</b>	%QX:0	Varies according to the equipment	Output: Read and write data value in the output. The data type can be only BIT.
<b>%I</b>	%I:0	Varies according to the equipment	Input: Read and write data value in the inputs. The data type can be BYTE, WORD, DWORD or FLOAT.
<b>%IX</b>	%IX:0	Varies according to the equipment	Input: Read and write data value in the input. The data type can be only BIT.
<b>NAME</b>	NAME	Varies according to the equipment	Read and write data value using the variable name in the address. The data type will be discovered automatically.
<b>REFRESH</b>	REFRESH	Varies according to the equipment	Refresh the driver internal list with variable information configured in the TwinCAT PLC.

- **Address field:** Use the information in the next table 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:  
**<Type><AddressOffset>.[Bit]** (for example: **B10, W20.5, DW1, .engine**).

Where:

- **<Type>** Format of data– **B** (byte), **w** (word), **D** (double word) and **F** (float). It must be used only with the **%Q**, **%I** and **%M** operands. When the operand is **%QX**, **%MX** or **%IX**, the data type must be **BYTE** and the **Bit** must be configured.
- **<AddressOffset>** is a parameter added to the **AddressReference** parameter (configured in the **Header** field) to compose the group address configured in the **Header** field.
- **<Bit>** (*optional parameter*) is the bit number to be read from or written to the device. The bit value must be configured with **%QX**, **%MX** or **%IX** operands. If it is not configured, the zero value will be **get** by default.

Address Configuration Sample		
Device Address	Header Field	Address Field
%QX0.0	%QX0	B0.0
%QX0.5	%QX0	B0.5
%QX1.0	%QX0	B1.0
%QX0.0	%Q0	B0.0
%Q0 (8 bits)	%Q0	B0
%QX1.1	%Q0	W0.9
%M0 (Word)	%M0	W0
%M5 (BYTE)	%M0	B5
%M5 (BYTE)	%M5	B0
%MX5.0	%MX0	B5.0
%IX10.7	%IX0	B10.7
%IX10.7	%I0	B10.7
%I10 and %I11 (16bits)	%I	W10
engine (Global Variable)	NAME	.engine
speed (Global Variable)	NAME	.speed
down (Global Variable)	NAME	.down
balance (main program)	NAME	MAIN.balance

**⚠ Caution:**

You must not configure a range of addresses or variable 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 128 addresses or variable name, per *Standard Driver* worksheet.

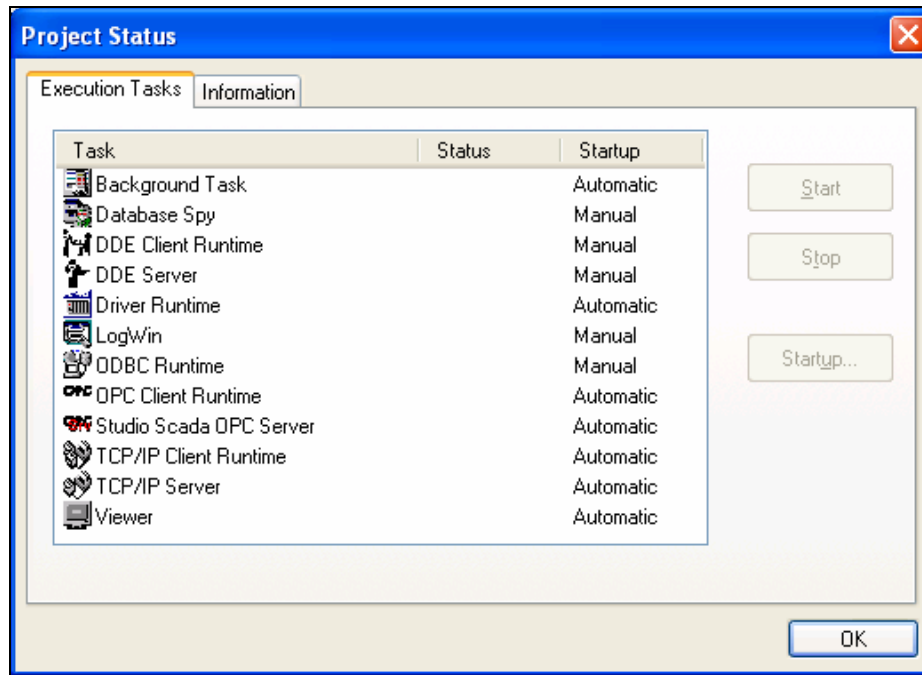
## Executing the Driver

After adding the TWCAT 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 Runtime* 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 TWCAT 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
1	Invalid AmsNet ID Station	Invalid AmsNet ID Station. The AmsNet ID must have 6 numbers.	Type a valid AmsNet ID.
30	Error loading third party library	The third party library is not installed (tcAdsDll.dll or tcADSDIICE.dll).	Install the third party library <i>TwinCAT ADS Interface PLC</i> software.
40	Error loading extern function	The extern function can not be loaded.	Install the correct tcAdsDll.dll or tcADSDIICE.dll library. The library exists, but the extern functions cannot be loaded.
50	Invalid command	Header in the Driver Worksheet is invalid.	Type a valid header.
60	Invalid datatype	Specified address contains an invalid value	Type a valid address.
70	Error in the GetLocalAddress function	The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Contact your Studio technical support representative.
90	Error writing data	The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Note the error text in the LogWin, and contact your Studio technical support representative.
100	Error in the ADSIGRP_SYM_READWRITE (InfoByNameEx) service	Cannot load the TwinCAT PLC variables information. The TwinCAT ADS Interface PLC is not working properly, TwinCAT software is not running, or the configured variable name is not configured in the TwinCAT PLC software.	Note the error text in the LogWin, and contact your Studio technical support representative.
110	Error in the ADSIGRP_SYM_READWRITE service	Cannot read the TwinCAT PLC variables values. The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Contact your Studio technical support representative.
120	Error processing value	The configured variable name or address is not configured in the TwinCAT PLC software or is out of range.	Note the error text in the LogWin, and contact your Studio technical support representative.
130	BlockSize Error	Address or variable configured in Driver sheet is over the limit.	Only 128 addresses or variables can be configured per Driver Sheet.
140	Too many connections	AmsNetID is over the limit.	Only 64 AmsNetID addresses can be configured.
-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 Studio <i>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 Studio <i>Technical Reference Manual</i> for information about valid RTS/CTS configurations.</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** and **Field Write Commands**, 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.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands** and **Field Write Commands** for the *LogWin* window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

## **Sample Application**

This driver does not have a sample application.

## Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Roberto Vigiani, Jr.	Jan/28/2004	First driver version
B	1.01	Roberto Vigiani, Jr.	Feb/04/2004	Modified the extern function to dynamic mode
C	1.02	Fabio H.Y. Komura	Oct/27/2004	<ul style="list-style-type: none"><li>- Implemented remote access</li><li>- Accessed all runtimes</li><li>- Increased read performance</li></ul>
D	1.03	Bruno A. Crepaldi	Jan/06/2005	Modified to accept any port number in the Station field.
E	1.04	Leandro G. Coeli	Apr/25/2005	<ul style="list-style-type: none"><li>- Implemented String Type</li><li>- Implemented Signed Type</li></ul>
F	1.05	Leandro G. Coeli	Sep/6/2005	Fixed problems with GPF in ARMs processors
G	1.06	Leandro G. Coeli	Sep/26/2005	Fixed problems with string writing