

Communication Driver MISTC

Driver for serial communication with OPTO22 devices using RS-232 serial interface

Index

- 1 INTRODUCTION2**
- 2 GENERAL CHARACTERISTICS.....3**
 - 2.1 DEVICE CHARACTERISTICS.....3
 - 2.2 LINK CHARACTERISTICS.....4
 - 2.3 DRIVER CHARACTERISTICS4
 - 2.4 INFORMATION ABOUT CONFORMANCE TESTING.....5
- 3 INSTALLATION.....6**
 - 3.1 INSTALLING THE DRIVER6
 - 3.2 OTHER SOFTWARE REQUIREMENTS.....6
- 4 DRIVER CONFIGURATION.....7**
 - 4.1 SETTINGS - COMMUNICATION PARAMETERS7
 - 4.2 DRIVER WORKSHEET.....10
 - 4.3 STATION AND HEADER CONFIGURATION11
 - 4.4 ADDRESS CONFIGURATION.....12
 - 4.5 DEVICE CONFIGURATION13
- 5 EXECUTION14**
- 6 TROUBLESHOOTING15**
- 7 APPLICATION SAMPLE.....17**
- 8 HISTORY OF VERSIONS18**

1 Introduction

The MISTC driver enables communication between Studio system and some of the OPTO22 devices by using “Mistic” protocol (RS 232), in accordance with the characteristics covered in this document.

This document contains 8 parts, as follow:

- Introduction: Provides an overview of the driver documentation.
- General characteristics: Provides information necessary to identify all the required components (hardware and software) necessary to implement the communication and global characteristics about the communication.
- Installation: Explains the procedures that must be followed to install the software and hardware required for the communication.
- Driver configuration: Provides the required information to configure the communication driver such as the different permutations for configuration and its default values.
- Execution: Explain the steps to test whether the driver was correctly installed and configured.
- Troubleshooting: Supplies a list of the most common error codes for this protocol and the procedures to fix them.
- Application Sample: Provides a sample application for testing the configuration the driver.
- History of versions: Provides a log of all the modifications done in driver.

☞ Note: This document presumes that the user has read the chapter *Driver Configuration* of the Studio's Technical reference manual.

2 General Characteristics

2.1 Device Characteristics

- Manufacturer: OPTO22
- MISTIC 200 Model
- Display Rev 1.11
- OPTO Kernel R3.0a
- Compatible Equipment :

Controller Name	ROM Configuration	Controller Type
G4LC32 (obsolete)	UVROM	216
G4LC32SX (obsolete)	UVROM	217
G4LC32	1 Megabit	218
G4LC32ISA	1 Megabit	219
G4LC32SX	1 Megabit	220
M4RTU	1 Megabit	221
G4LC32	4 Megabit	222
G4LC32ISA	4 Megabit	223
G4LC32SX	4 Megabit	224
M4RTU	4 Megabit	225
G4LC32ISA-LT	Fixed	226
M4	1 Megabit	227
M4	4 Megabit	228
M4IO	1 Megabit	229
M4IO	4 Megabit	230
SNAP LCSX	1 Megabit	232
SNAP LCSX Plus	1 Megabit	234
SNAP LCM4	4 Megabit	236
Runtime PC	N/A	231

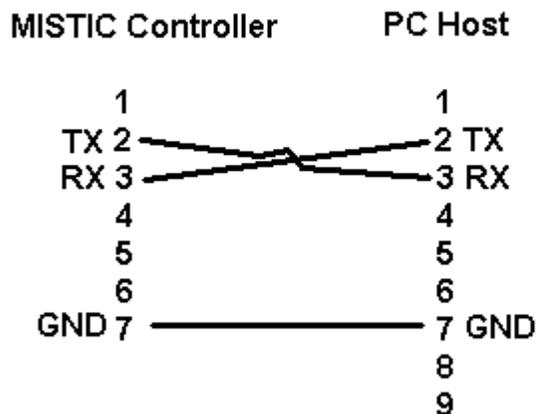
- OPTO22 programmer software:
 OPTOControl (Strategy configuration) or
 OPTOTerm (Controller configuration and Communication test)

 Note: Please refers to section 2.4 to see the Equipment used in the standard conformance tests for this driver.

2.2 Link Characteristics

- Device communication port: RS232 port
- Physical protocol: RS232
- Logic protocol: Mistic **ASCII** Protocol or Mistic **Binary** Protocol
- Device Runtime software: None
- Specific PC Board: None
- Adapters / Converters: None
- Cable Wiring: G4LC32 cable. Note: You must connect cable in the Com 0 or Com 1 of Controller. These ports implement RS232 serial communication.

Communication Cable



2.3 Driver Characteristics

- Operating System:
 - Windows CE
 - Windows XP

☞ Note: Please refer to section 2.4 to see the Operating System used in the conformance tests for this driver.

The driver is composed of the following files:

- MISTC.INI: Internal file of the driver, it should not be modified by the user.
- MISTC.MSG: This file contains the error messages for each error code. It is an internal file of the driver, the user should not modify it.
- MISTC.PDF: This document provides detailed documentation about the driver.

- MISTC.DLL: This is the compiled library for the driver.

☞ Note: All the files above must to be in the subdirectory /DRV of the Studio's installation directory.

2.4 Information about conformance testing

- Equipment: G4LC32

Configuration:

PLC program: InitVariables

Baud Rate: 19200

Protocol: Mistic ASCII protocol and Mistic BINARY protocol

Data Bits: 8

Stop Bits: 1

Parity: None

COM port: COM1

- Cable: According link specification, section 2.2.
- Operating System (development): Windows NT 4.0 + Service pack 6
- Operating System (target): Windows NT 4.0 + Service Pack 6; Windows CE v2.11 (x86 and SH3)
- Studio Version: 3.0 – Service Pack 1
- Driver version: 1.01

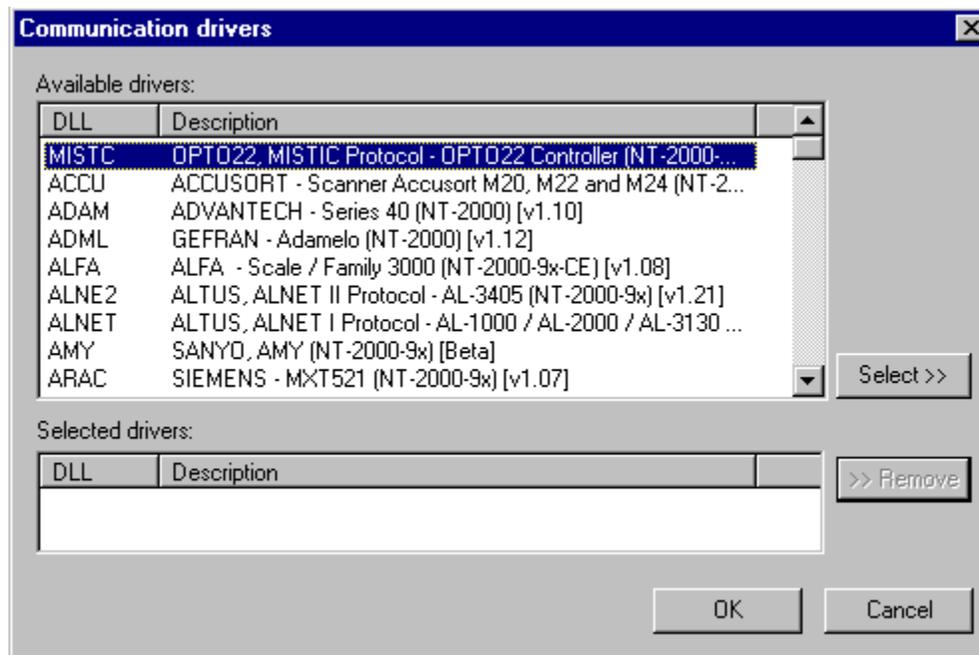
3 Installation

3.1 Installing the Driver

When you install the Studio v3.0 or higher, the communication drivers are already installed. You need now to select the driver at the applications where it will be used.

The steps to select the driver inside an application are:

1. Execute the Studio and select the proper application.
2. Select the menu *Insert + Driver...*
3. In the column **Available Drivers**, select the **MISTC Driver** and push the button **ADD>>>** (the driver MISTC must appear in the column **Selected Drivers**).
4. Press **OK**.



3.2 Other software requirements

It is not necessary to install any other software in the PC to enable the communication between the host and the Device. However, to download the custom program to the device, it is necessary to install one of the OPTO22 programmer software, for example, OPTOControl. Please see the OPTOControl documentation about the procedure to install their software.

☞ Note: Special care must be taken when installing the physical hardware. Refer to the hardware manufacturer documentation for specific instructions in this area.

4 Driver Configuration

After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

The driver configuration is two parts:

The Settings or Communication parameters, it is only one configuration to the whole driver, then you have the communication tables or Driver Worksheets, where the communication tags are defined.

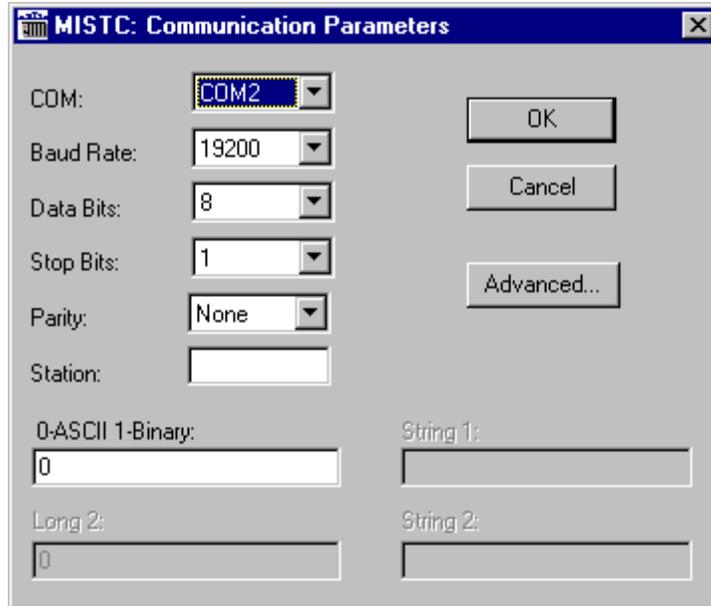
4.1 Settings - Communication Parameters

These parameters are valid for all driver worksheets configured in the system. To open the window for configuring the **Communication parameters**, follow these steps:

1. In the **Workspace** of the Studio environment, select the **Comm** table.
2. Expand the folder **Drivers** and select the subfolder **MISTC**.
3. Right click on the **MISTC** subfolder and select the option **Settings**.



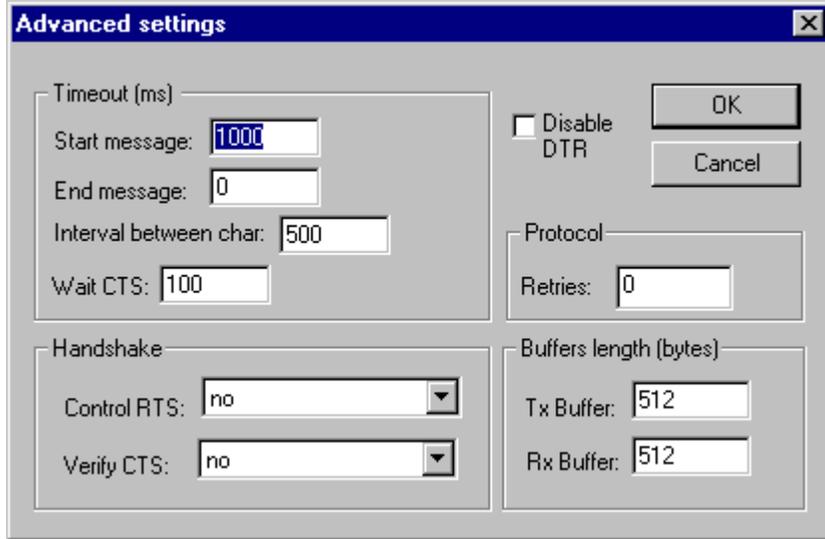
When selecting the Settings, there is the following dialog to configure:



Parameter	Default Value	Valid values	Description
COM	COM1	COM1 to COM8	Serial port of the PC used to communication with the device
Baud Rate	19200	110 to 57600bps	Communication rate of data
Data Bits	8	5 to 8	Number of data bits used in the protocol
Stop Bits	1	1or 2	Number of stop bits used in the protocol
Parity	None	even, odd, none, space or mark	Parity of the protocol
Station	0	0	Not used for this driver
0-Ascii 1-Binary	0	0 or 1	0 – Set ASCII protocol to driver communication. 1 – Set BINARY protocol to driver communication. (The setting of protocol must be the same of controller configuration)

Note: These Parameters must be just the same as the configured on the OPTO22 device. The Opto22 Programmable Controller Manual indicates the following serial communication settings:
Baud Rate: 19200, Data bits: 8, Stop bits: 1, Parity: None

By clicking on the button **Advanced...** in the window **Communication Parameters**, you open additional communication parameters.



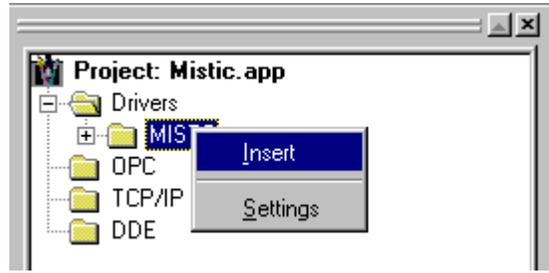
The Advanced setting parameters are explained at the Studio Technical Reference Manual, and you should keep the default values to all fields. Only the field described at the next table should be configured:

Parameter	Default Value	Valid values	Description
Control RTS	No	no, yes or yes + echo	Define if the handshake signal of RTS (Request to Send) is set before communication and if there is an echo in the communication. You must to choose the option " no ". Important: Wrong settings on this field will not let the driver work, having the Timeout waiting start a message error.

4.2 Driver Worksheet

It is possible to configure many driver worksheets, each one will be composed of a Header and Body. To create a new driver worksheet, follow these steps:

1. In the **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **MISTC**.
3. Right click on the **MISTC** subfolder and select the option **I**nsert.



Note: To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

When creating a communication table, you have the following window:

Description: Increase read priority

Read Trigger: Enable Read when Idle: Read Completed: Read Status:

Write Trigger: Enable Write on Tag Change: Write Completed: Write Status:

Station: Header: Min: Max:

	Tag Name	Address	Div	Add
1	Float_tab[1]	0		
2	Float_tab[2]	1		
3				

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Communication Driver documentation about the configuration of the standard fields. This document describes the Station, Header and Address fields, which are specific to each communication driver.

4.3 Station and Header configuration

Parameter	Default Value	Valid values	Description
Station	-	0 to 255	The PLC's Address. This value must be just the same as the configured on the OPTO22 controller.
Header	I	Vide next table	Defines the type of variable to be read or written from or to the device.

The **Header** field defines the type of variables which will be read/written from/to the device. It complies with the syntax: <variables type> or <table variables type>:<table variables name>. After editing the field **Header**, the system will check if it is valid or not. If the syntax is incorrect, the default value (I) or (I:Table name) will be automatically placed in this field.

You can type Tag between curly brackets into this field, but be sure that the Tag's value is correct, with the right syntax, or you will get the Invalid Header error. The right syntax, both for the field typing and Tag value is described below:

- **Variables type or Table variables type:**
 - I – Integer variable type or Integer table variable type.
 - F – Float-point variable type or Float-point table variable type.
 - S – String variable type or String table variable type.
- **Table variables name:** Table name configured in Opto22 controller with the same type of “table variables type” value.

Important: The “Table variable name” is case sensitive, it must be the exactly the same of the controller configuration.

Information regarding the parameter “Header”			
Type	Sample of syntax	Valid range of initial Address	Comment
I	I or I:<INTEGER_TABLE>	-	Integer type: <ul style="list-style-type: none"> • I – Read and write integer variables. This type can only execute item commands. • I:<INTEGER_TABLE> - Read and write integer table variables. This type can only execute write item command but it can execute read group command.
F	F or F:<FLOAT_TABLE>	-	Float-point type: <ul style="list-style-type: none"> • F – Read and write float-point variables. This type can only execute item commands. • I:<FLOAT_TABLE> - Read and write float table variables. This type can only execute write item command but it can execute read group command.
S	S or S:<STRING_TABLE>	-	String type: <ul style="list-style-type: none"> • S – Read and write string variables. This type can only execute item commands. • S:<STRING_TABLE> - Read and write string table variables. This type can only execute item commands.

4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the column **Tag Name**, you must type the tag from your application database. This tag will receive or send values from or to an address on the device.

The address cells complies to the following syntax:

- **Single variables**
The address must receive the variable name configured in the controller. There is not write and read group. There is only write and read item.
- **Table variables**
The address must receive the position of table configured in the Header.

Sample of Addressing Configuration		
Address on the Device	Header Field	Address Field
Integer variable – INTEGER_VARIABLE	I	INTEGER_VARIABLE
Integer variable – Int1	I	Int1
Integer variable – Int_Val1	I	Int_Val1
Float-point variable – FLOAT_VARIABLE	F	FLOAT_VARIABLE
Float variable – Float1	F	Float1
Float variable – FloatVal1	F	FloatVal1
String variable – STRING_VARIABLE	S	STRING_VARIABLE
String variable – Str1	S	Str1
String variable – Str2	S	Str2
Table Integer variable INTEGER_TABLE position 0	I:INTEGER_TABLE	0
Table Integer variable INTEGER_TABLE position 5	I:INTEGER_TABLE	5
Table Integer variable INT_TAB position 1	INT_TAB	1
Table Integer variable INT_TAB position 7	INT_TAB	7
Table Float variable – FLOAT_TABLE position 0	FLOAT_TABLE	0
Table Float variable – FLOAT_TABLE position 9	FLOAT_TABLE	9
Table Float variable – RealValue position 11	RealValue	11
Table Float variable – RealValue position 15	RealValue	15
Table String variable – STRTAB position 0	STRTAB	0
Table String variable – STRTAB position 1	STRTAB	1
Table String variable – STRTAB position 2	STRTAB	2
Table String variable – STRING_TAB position 0	STRING_TAB	0
Table String variable – STRING_TAB position 10	STRING_TAB	10

4.5 Device Configuration

The Opto22 controller must have the following configuration.

Baud Rate: 19200
Data bits: 8
Stop bits: 1
Parity: None
Protocol : ASCII or Binary

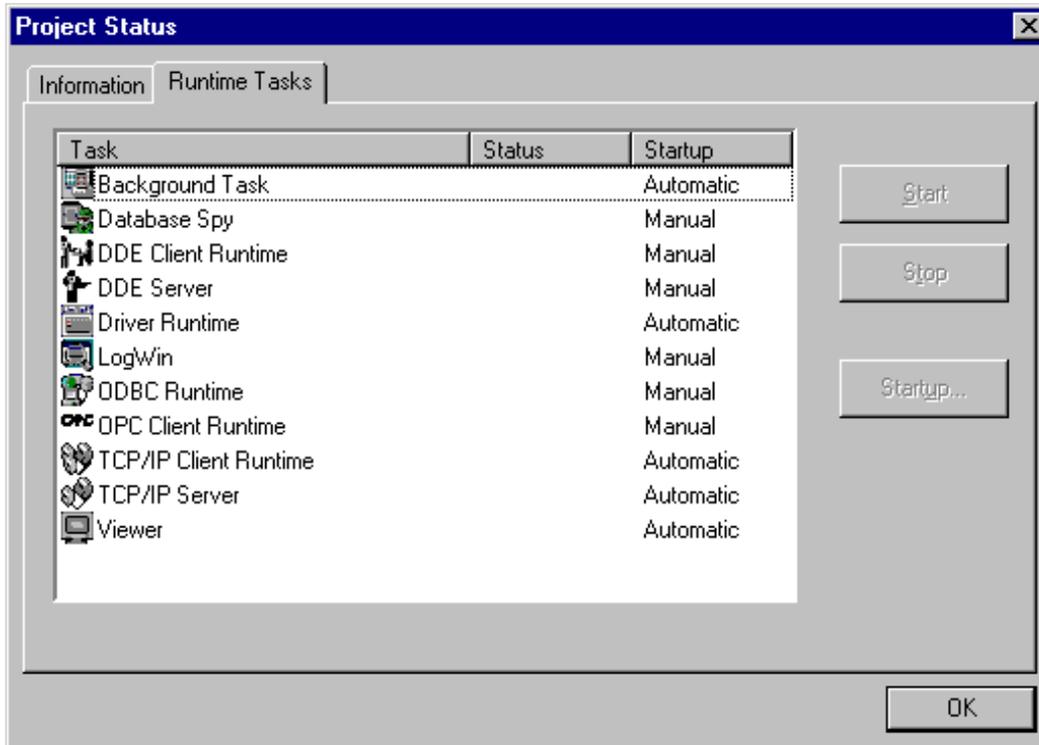
Besides that, it is required to select the PLC station number, by using the setup in the PLC panel..

Use the OptoControl or the OptoTerm to have sure of communication status is OK.

- **OptoControl:**
You can read and write values with debug mode.
- **OptoTerm:**
You can use the OptoTerm to send the “Up Clear Command” typing:
‘A’ and <ENTER>.
You can read a integer variable typing :
^INTEGER_VARIABLE @@ .

5 Execution

- When installing the driver, it is automatically selected to execute when you start-up the Runtime Environment. To verify the if the driver is correctly enabled to start, use the menu option **Project + Status...**, and verify the task Driver Runtime



6 Troubleshooting

After each attempt to communicate using this driver, the tag configured in the field **Read Status** or **Write Status** will receive the error code regarding the kind of failure that occurred. The error messages are:

Error Code	Description (*)	Possible causes	Procedure to solve
0	OK	Communication without problems	
2	Controller received a message with a bad CRC	<ul style="list-style-type: none"> - Incorrect or loose communication wiring. - High noise level on the communication line. - Missing terminator on the ends of the communication cable (RS-485/422 only). - Twisted pair cable not used (RS-485/422 only). 	
4	Controller power-up clear	<ul style="list-style-type: none"> - Controller lost power since last communication. 	<ul style="list-style-type: none"> - Controller need of a Power-UP Clear. - Type 'A' and <ENTER> in the OptoTerm.
5	Controller received insufficient data in a particular data field	<ul style="list-style-type: none"> - Error in command syntax. 	<ul style="list-style-type: none"> - Type a valid Header either on the header field or on the tag value. - Type a valid Address either on the Address field.
10	Invalid Header	<ul style="list-style-type: none"> - An invalid Header has been typed or the tag that is inside this field has an invalid configuration. 	<ul style="list-style-type: none"> - Type a valid Header either on the header field or on the tag value. A lot of different valid headers are shown on the section 4.2
11	Invalid Address	<ul style="list-style-type: none"> - An invalid Address has been typed or the tag that is inside this field has an invalid configuration 	<ul style="list-style-type: none"> - Type a valid Address either on the Address field.
6	Checksum Error	<ul style="list-style-type: none"> - It's a protocol error 	<ul style="list-style-type: none"> - Check the serial communication configuration. Verify if the settings on the Communication Parameters and on the device are the same.
30	Invalid port number	<ul style="list-style-type: none"> - Sending a peer message to the same controller (analogous to talking to yourself). - Firmware error. 	<ul style="list-style-type: none"> - Check Firmware version.
31	Send timeout	<ul style="list-style-type: none"> - Sending long strings to a host port. - Port timeout delay too short. - Firmware error. 	<ul style="list-style-type: none"> - Increase delay timeout in the controller. - Check Firmware version.
32	Bad table index	<ul style="list-style-type: none"> - Negative table index value. - Table index value greater than the table length. 	<ul style="list-style-type: none"> - Check the value of table index.
33	Numeric overflow	<ul style="list-style-type: none"> - The result of a calculation is larger than the numeric type used. 	<ul style="list-style-type: none"> - Check calculations and numeric type used.
35	Not a number	<ul style="list-style-type: none"> - A math operation resulting in a complex or imaginary number, such as 	<ul style="list-style-type: none"> - Check math operations.

		the square root of a negative number.	
36	Divide by zero	- A math operation tried to divide a constant or variable by 0.	- Check math operations.
39	Port already in use	- Attempt to have more than one port "open" in a chart.	- Check if port already is used.
45	String too short to hold data	- String variable too short for data specified. - Attempt to put the date or time in a string with a length less than eight.	- Check String and data specified.
-15	Timeout waiting start a message.	- Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings.	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
-17	Timeout between rx char.	- PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.

Note: The results of the communication may be verified in the **output** Window of the Studio's environment. To set a log of events for **Field Read Commands**, **Field Write Commands** and **Serial Communication** click with the right button of the mouse on the output window and chose the option setting to select these log events. When testing under a Windows CE target, you can enable the log at the unit (Tools/Logwin) and verify the file celog.txt created at the target unit.

When you are not able to establish the communication with the PLC, you should first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication it is not possible due to a hardware or cable problem, or due an error or lack of configuration at the PLC. Only after the communication between the PLC Programming Software and the PLC is working fine, you can test again the supervisory.

When testing the communication with the Studio, you should first use the application sample described at item 7, instead of the new application that you are creating.

If is required to contact technical support, please have the following information available:

- Operating System (type and version): To find this information use the Tools/System Information option
- Project information: It is displayed using the option Project/Status from the Studio menu
- Driver version and communication log: Available from Studio Output when running the driver
- Device model and boards: please refer to hardware manufacture's documentation

7 Application Sample

The Studio contains a configured project to test the driver. It is strongly recommended to do some tests with this application before beginning the configuration of the customized project, for the follow reasons:

- To understand better the information covered in section 4 of this document.
- To verify that your configuration is working.
- To certify that the hardware used in the test (device + adapter + cable + PC) is in working conditions before beginning the configuration of the applications.

☞ Note: The Application Sample is not available for all drivers.

The Studio application is in the directory: /COMMUNICATION EXAMPLES/<Driver Name>

To perform the test, you need to follow these steps:

- Configure the device communication parameters using manufacturer programmer software..
- Open the application /COMMUNICATION EXAMPLES/<Driver Name>
- Execute the application
- To display the following screen with some information about the communication, please execute the Viewer module in the Studio.

☞ Note: The application for testing may be used like a maintenance screen for the custom application.

8 History of Versions

Version	By	Date	Description of changes
1.01	Roberto V. Junior	17-feb-2000	<ul style="list-style-type: none">▪ First driver version▪ Driver available for Windows CE
1.03	André Körbes	23-Sep-2010	<ul style="list-style-type: none">▪ Fixed driver initialization issues