Communication Driver BERK

Driver for communication with Berkeley BXi Controller

Index

1 Introduction

The **BERK** driver enables communication between Studio system and Berkeley BXi controller devices using the Berkeley proprietary protocol, in accordance with the characteristics covered in this document.

This document contains 8 parts, as follows:

- **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.

➲**Attention:** This driver is compatible only with the UNICODE versions of Studio released as of Oct/23/2001.

2 General Characteristics

2.1 Device Characteristics

- **Manufacturer**: Berkeley process control
- **Compatible Equipment**:
  - BXi controllers
- **BXi controller programmer software**: MachineWorks

↳**Tip:** 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**:: Ethernet port
- **Physical protocol**: Ethernet
- **Logic protocol**: Berkeley proprietary
- **Device Runtime software**: None
- **Specific PC Board**: Ethernet card
- **Cable**: Ethernet cable

2.3 Driver Characteristics

- **Operating System**:
  - Windows 9x
  - Windows 2000
  - Windows NT
  - Windows CE (x86 / SH3 / SH4 / MIPs / ARM / PPC)

↳**Tip:** 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:

- **BERK.INI:** Internal file of the driver, it should not be modified by the user.
- **BERK.MSG:** Error messages for each error code. It should not be modified.
- **BERK.PDF:** Provides detailed documentation about the driver.
- **BERK.DLL:** Compiled driver.

✍ **Note:** All the files above must to be in the subdirectory /DRV of the Studio’s installation directory.

- **Supported Registers:**

Register Type	Length	Write	Read	Bit	Integer	Float
Axis	8 Bytes	–	●	–	–	●
PC Variables	8 Bytes	●	–	–	–	●
Network Variables	8 Bytes	–	●	–	–	●
I/O	*See Note	–	●	●	●	–
User Mail Messages	*See Note	●	●	–	–	–

✍ **Note:** The I/O data length will depend on the I/O type: Bit for digital I/O and Integer for Analog I/O. The user mail messages are string values that can be sent or read to or from the controller.

2.4 Information about conformance testing

- Equipment: BXi controller

Configuration:

- Cable: see section 2.2
- Operational System (development): Windows 2K + Service pack 1
- Operational System (target): Windows 2K + Service Pack 1
- Studio Version: 4.4 + Service Pack 2
- Driver version: 1.00

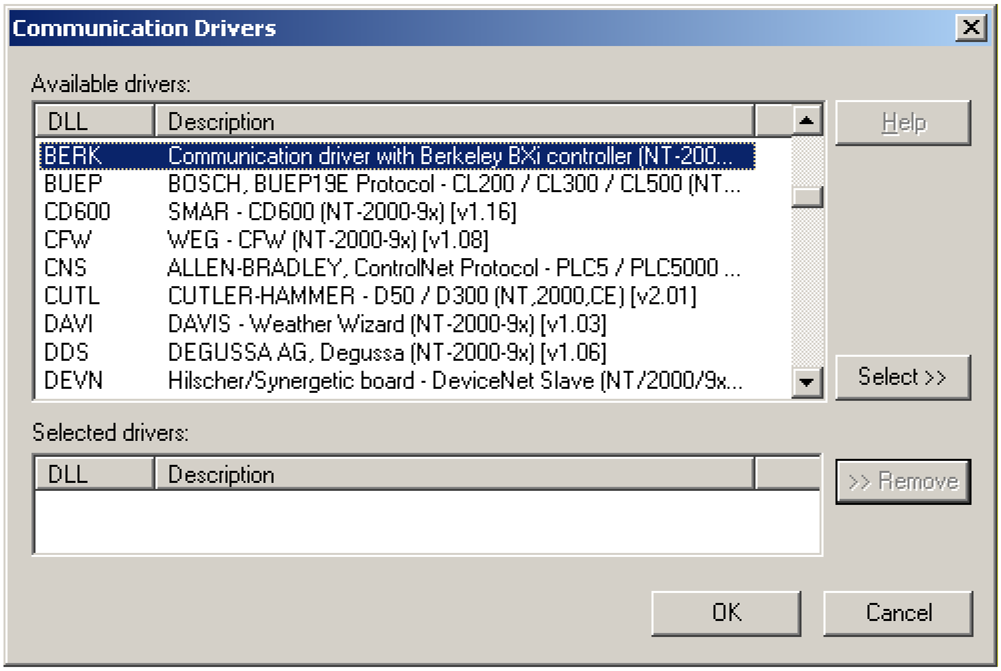
3 Installation

3.1 Installing the Driver

When you install the Studio, 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 **BERK Driver** and push the button **Select >>** (the driver BERK must appear in the **Selected Drivers** area).
4. Press **OK**.



Available drivers:

DLL	Description
BERK	Communication driver with Berkeley BXi controller (NT-200...
BUEP	BOSCH, BUEP19E Protocol - CL200 / CL300 / CL500 (NT...
CD600	SMAR - CD600 (NT-2000-9x) [v1.16]
CFW	WEG - CFW (NT-2000-9x) [v1.08]
CNS	ALLEN-BRADLEY, ControlNet Protocol - PLC5 / PLC5000 ...
CUTL	CUTLER-HAMMER - D50 / D300 (NT,2000,CE) [v2.01]
DAVI	DAVIS - Weather Wizard (NT-2000-9x) [v1.03]
DDS	DEGUSSA AG, Degussa (NT-2000-9x) [v1.06]
DEVN	Hilscher/Synergetic board - DeviceNet Slave (NT/2000/9x...

Selected drivers:

DLL	Description

3.2 *Other software requirements*

It is not necessary to install any other software in the PC to enable the communication between the Studio and the Device. It is only needed to enable TCP/IP protocol in the Operating System.

➲**Attention:** Special precautions must be taken when installing the physical hardware. Refer to the hardware manufacturer documentation for specific instructions in this area.

4 Driver Configuration

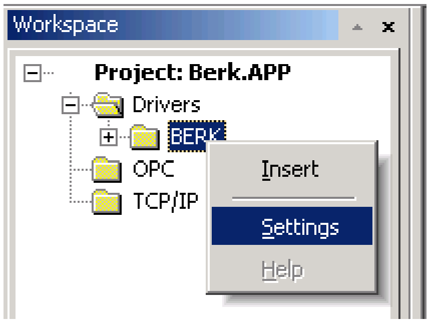
After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

The driver configuration is two parts:
- The Settings or Communication parameters, it is only one configuration to the whole driver;
- The communication tables or Driver Worksheets, where the communication tags are defined. There are two types of communication tables: **Standard Tables** and **MAIN DRIVER SHEET**. BERK driver specifically implements only **Standard Tables**.

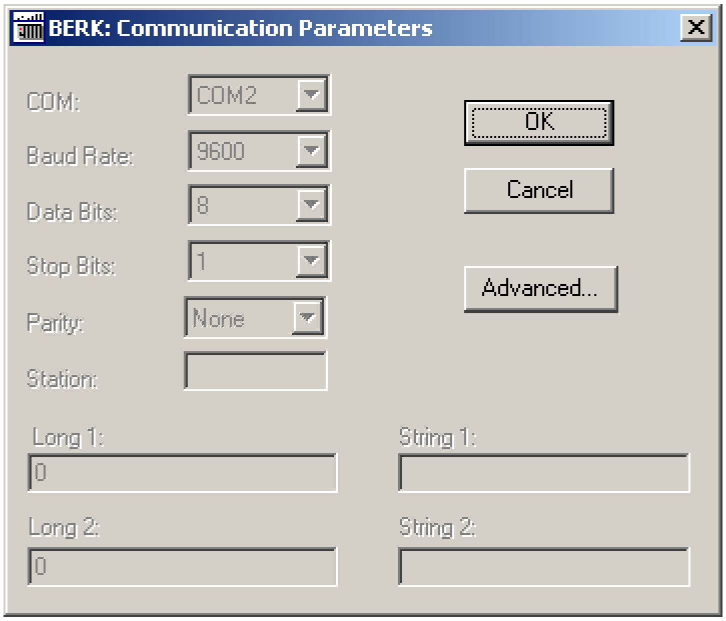
4.1 Settings - Communication Parameters

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 **BERK**.
3. Right click on the **BERK** subfolder and select the option **Settings**.



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



COM: COM2

Baud Rate: 9600

Data Bits: 8

Stop Bits: 1

Parity: None

Station:

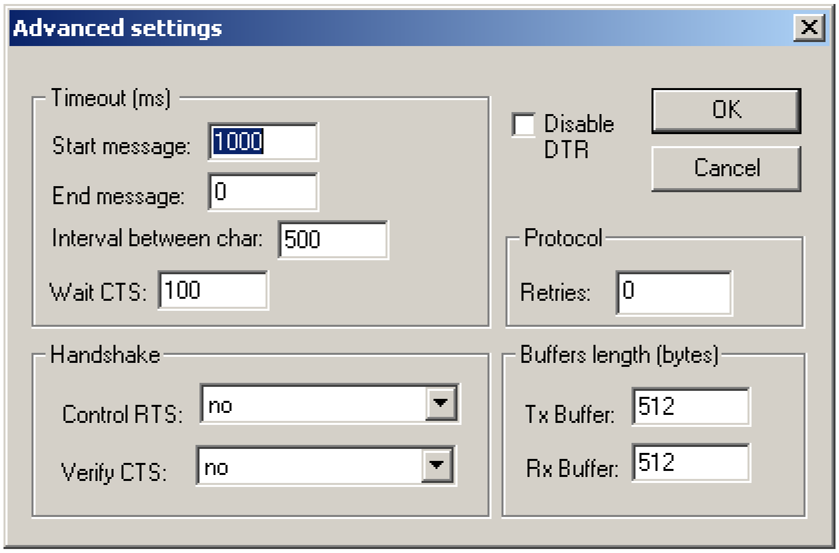
Long 1: 0

String 1:

Long 2: 0

String 2:

By clicking on the button **Advanced…** in the windows **Communication Parameters,** you will be able to configure other communication parameters. The only one that is applied to BERK driver is the Start message time out. It is the time out when sending user mail messages and to indicate the connection status, you can keep the default values to all fields.



Advanced settings

Timeout (ms)

Start message: 1000

End message: 0

Interval between char: 500

Wait CTS: 100

□ Disable DTR

Protocol

Retries: 0

Handshake

Control RTS: no

Verify CTS: no

Buffers length (bytes)

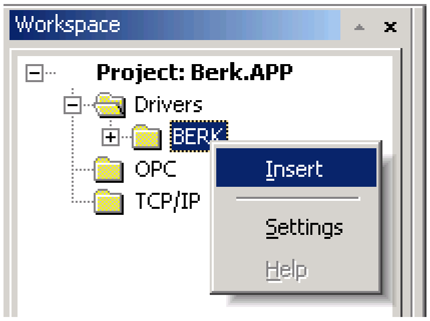
Tx Buffer: 512

Rx Buffer: 512

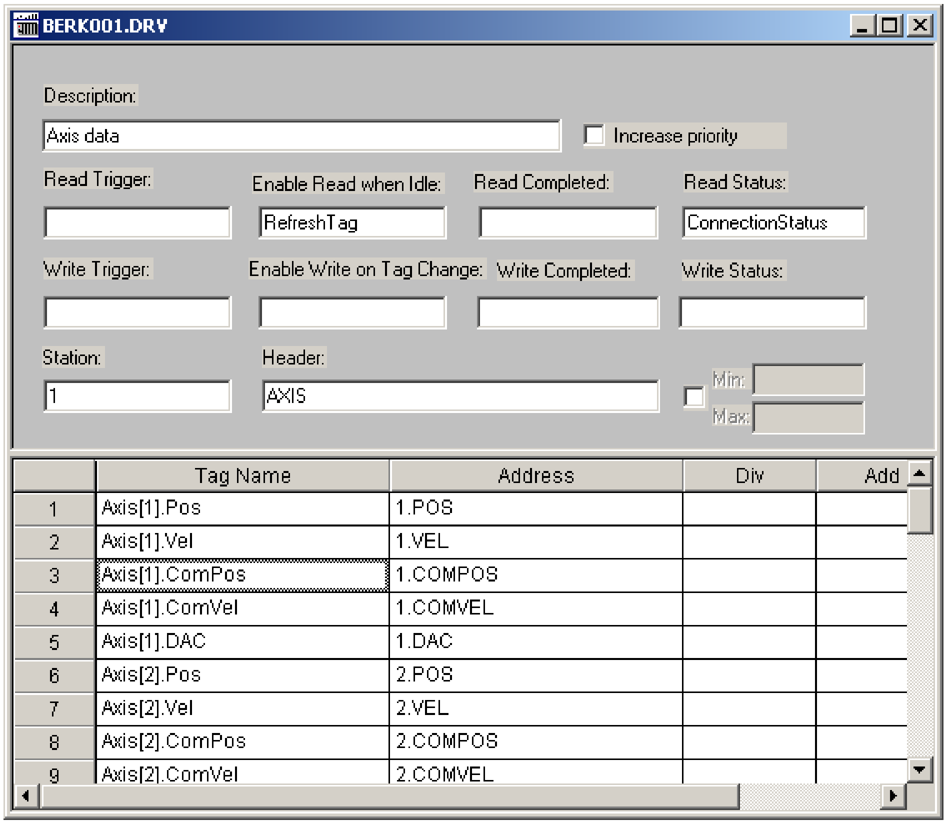
4.2 Standard Driver Worksheet

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 **BERK**.
3. Right click on the **BERK** subfolder and select the option **Insert**.



Tip: 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.



BERK001.DRV

Description: Axis data

□ Increase priority

Read Trigger:	
Enable Read when Idle:	RefreshTag
Read Completed:	
Read Status:	ConnectionStatus
Write Trigger:	
Enable Write on Tag Change:	
Write Completed:	
Write Status:	
Station:	1
Header:	AXIS

□ Min: Max:

	Tag Name	Address	Div	Add
1	Axis[1].Pos	1.POS		
2	Axis[1].Vel	1.VEL		
3	Axis[1].ComPos	1.COMPOS		
4	Axis[1].ComVel	1.COMVEL		
5	Axis[1].DAC	1.DAC		
6	Axis[2].Pos	2.POS		
7	Axis[2].Vel	2.VEL		
8	Axis[2].ComPos	2.COMPOS		
9	Axis[2].ComVel	2.COMVEL		

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Technical Reference Manual 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

The parameter **Station** defines the device address to be read or written. It complies with the following syntax:

<Controller ID> (e.g.: 2)

The parameter **Header** defines the type of variables that will be read or written from or to the host. It complies with the following syntax:

<VariableType> (e.g.: AXIS)

- *VariableType*: Variable type to be read or written (AXIS, IO, RACKSTA, PCVAR, NTVAR, USERMAIL);

After editing the field **Header**, the system will check if it is valid or not. If the syntax were incorrect, the default value (AXIS) will be automatically placed in this field.

You can type string Tag between curly brackets into this field and in the Station 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:

Type	Header
Axis Data	AXIS
IO Data	IO
PC Variables	PCVAR
Network Variables	NTVAR
User Mail	USERMAIL

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:

For Axis data:
<AxisNumber>.<Information> (e.g.: 2.POS)

- AxisNumber*:* Number of the Axis to retrieve the information (1~8).

- Information*:* Information that will be retrieved from the axis (POS = Position, COMPOS = Commanded position, VEL = Velocity, COMVEL = Commanded velocity or DAC).

For IO data:
<Rack>:<Module>.<Point> (e.g.: 2:1.1)

- Rack*:* Rack number (0 = INTERNAL RACK).

- Module*:* Module number (1~10).

- Point*:* Point number (1~64).

For PC Variables and Network Variables data:
<Category>.<Variable> (e.g.: 1.1)

- Category*:* Variable category number (1~12).

- Variable*:* Variable number (1~20).

For user mail data:
<Length> (e.g.: 10)

- Length*:* Amount of chars that will be sent to the controller. If it is 0 (zero) the amount to be sent is the string in the tag configured in the tag column.

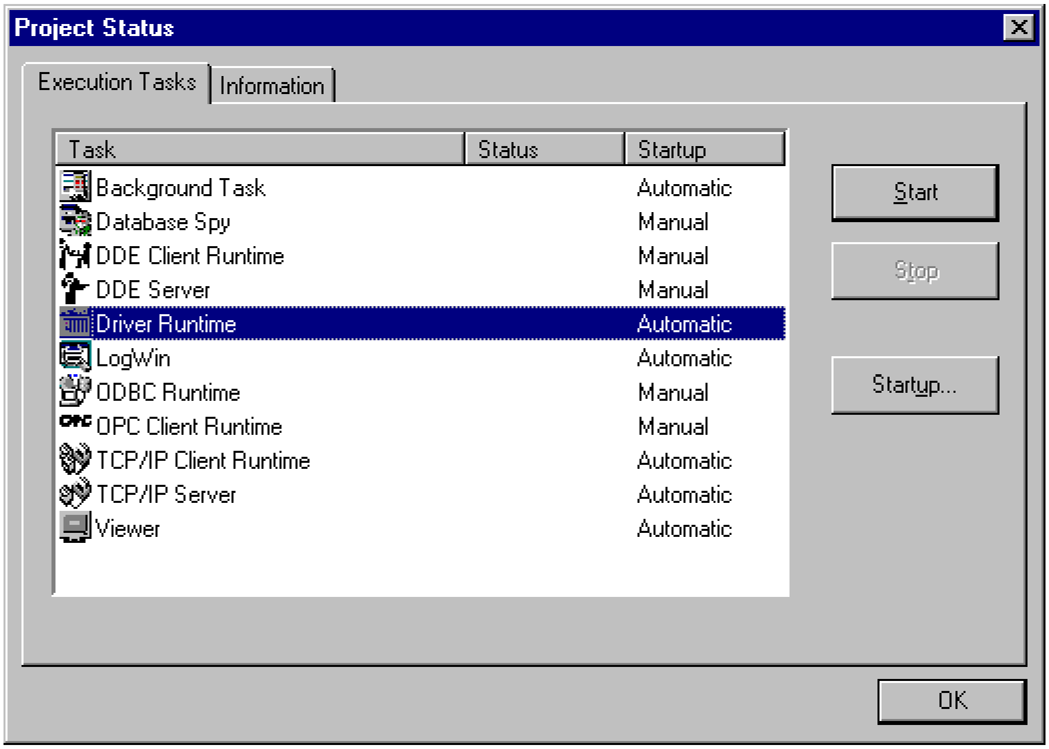
Sample of Addressing Configuration		
Device Variable	**Header Field**	**Address Field**
Position from Axis 1	AXIS	1.POS
Commanded position from Axis 2	AXIS	2.COMPOS
Velocity from Axis 3	AXIS	3.VEL
I/O Point #10 from Module #1 Rack #2	IO	2:1.10
PC Variable 1 from category 3	PCVAR	3.1
Network Variable 10 from category 5	NTVAR	5.10

4.5 Device Configuration

Please refer to Berkeley manuals to get information about device configuration.

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**



Task	Status	Startup
Background Task		Automatic
Database Spy		Manual
DDE Client Runtime		Manual
DDE Server		Manual
Driver Runtime		Automatic
LogWin		Automatic
ODBC Runtime		Manual
OPC Client Runtime		Manual
TCP/IP Client Runtime		Automatic
TCP/IP Server		Automatic
Viewer		Automatic

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	None
1	Invalid Header	Wrong Header typed at the driver configuration worksheet.	- See at the section 4.3 Station and Header Configuration
2	Invalid Address	Wrong Address typed at the driver configuration worksheet.	- If you're using a TAG on the Header Field, check if this TAG's value is valid to the configured addresses. - If you are not using tags, it is possible that you have changed the header after configuring the Addresses, and then, to the new header, the addresses is invalid.
3	Invalid Station	Wrong Station typed at the driver configuration worksheet.	- See at the section 4.3 Station and Header Configuration
21	Block Size Error	The offset on the Driver Configuration's worksheet is too big, and the message can not be framed.	- Change the offsets, or even though create a new worksheet.
23	Invalid Answer	Problems in the communication medium	Check all the cables, hubs and rounters
25	Disconnected	Controller is not connected	Check the controller address and all the cables, hubs and rounters.
100	Invalid I/O	The I/O information that was requested from the communication worksheet is not available. The controller has already sent the I/O configuration but no data has arrived.	Wait for some seconds until the controller sends the I/O data
101	Invalid Rack	The Rack number configured in the address field of the driver worksheet is not configured in the controller	Check the Rack number in the address field and the controller I/O configuration.
102	Invalid Module	The Module number configured in the address field of the driver worksheet is not configured in the controller	Check the Module number in the address field and the controller I/O configuration.
103	Invalid Point	The Point number configured in the address field of the driver worksheet is not configured in the controller	Check the Point number in the address field and the controller I/O configuration.
104	Invalid Axis	The Axis information to the Axis whose number was configured in the driver worksheet was not received from the controller yet.	Wait for some seconds until the controller sends the Axis data
105	Invalid Network Variable	The Variable information to the Network variable whose number was configured in the driver worksheet was not received from the controller yet.	Wait for some seconds until the controller sends the network variables data
106	Invalid Mail Size	The amount of chars to be sent to the controller if greater than the maximum size allowed by the UDP protocol	Reduce the message size and send it again.

-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.

↳**Tip:** The communication status can be verified by the **output** Window of the Studio's environment or by the **LogWin** module. 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 Device, first of all establish the communication between the Device Programming Tool and the PLC. Very frequently the communication 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 driver.

When testing the communication with the Studio, you should first use the application sample described at item 7 (if it's available), 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

Studio provides 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.

↳**Tip:** 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
1.00	José L. Teodoro	23-Oct-2001	▪ Initial driver version
1.01	José L. Teodoro	28-Jun-2002	▪ Improved the mail receiving functionality