

Communication Driver Axiom

Driver for communication with Axiom devices

Index

1 INTRODUCTION..... 2

2 GENERAL CHARACTERISTICS 3

2.1 DEVICE CHARACTERISTICS 3

2.2 LINK CHARACTERISTICS..... 3

2.3 DRIVER CHARACTERISTICS 3

2.4 INFORMATION ABOUT CONFORMANCE TESTING..... 3

3 INSTALLATION 4

3.1 INSTALLING THE DRIVER 4

3.2 OTHER SOFTWARE REQUIREMENTS 4

4 DRIVER CONFIGURATION 5

4.1 SETTINGS - COMMUNICATION PARAMETERS..... 5

4.2 DRIVER WORKSHEET..... 7

4.3 STATION AND HEADER CONFIGURATION..... 8

4.4 ADDRESS CONFIGURATION 9

4.5 DEVICE CONFIGURATION 11

5 EXECUTION 12

6 TROUBLESHOOTING..... 13

7 APPLICATION SAMPLE..... 15

8 HISTORY OF VERSIONS..... 15

1 Introduction

The Axiom driver enables communication between Studio system and some of the Axiom devices using their Host protocol, 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: Axiom
- Compatible Equipment
 - Analog and Digital PC boards from Axiom.
- Axiom configuration software, for example: Device Manager(Axiom driver).

☞ 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: PC Boards (ex: A5411)
- Logic protocol: Proprietary Host Protocol
- Device Runtime software: Axiom driver (library of boards access)
- Specific PC Board: Analog and digital PC boards from Axiom.

2.3 Driver Characteristics

- Operating System:
 - Windows NT

☞ 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:

- AXIOM.INI: Internal file of the driver, it should not be modified by the user.
- AXIOM.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.
- AXIOM.PDF: This document provides detailed documentation about the driver.
- AXIOM.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

This driver still do not have a conformance testing.

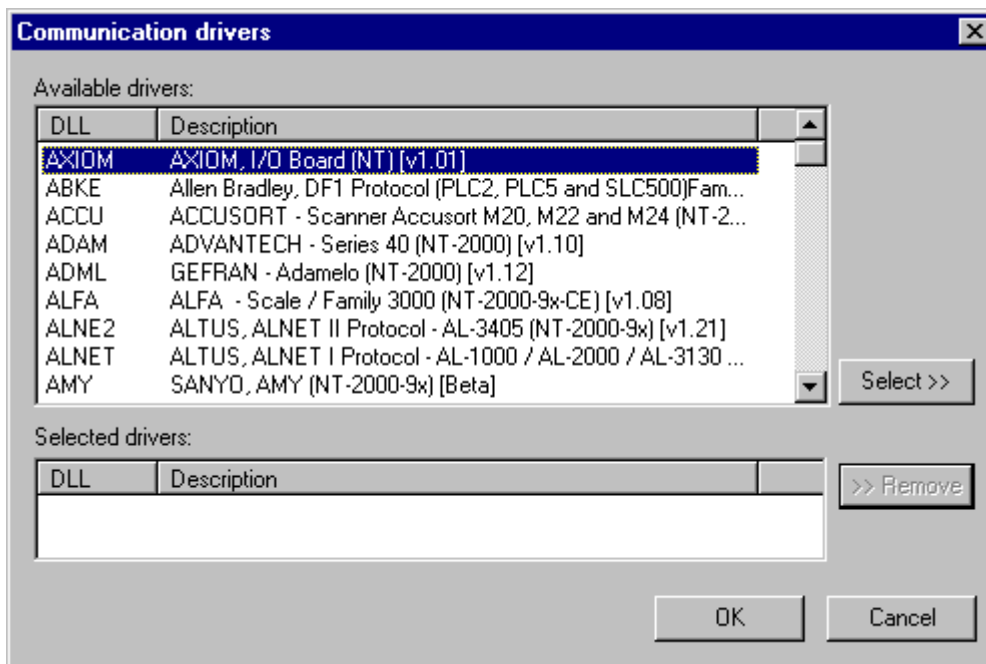
3 Installation

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 **Axiom Driver** and push the button **ADD>>>** (the driver Axiom must appear in the column **Selected Drivers**).
4. Press **OK**.



3.2 Other software requirements

You must install Axiom board library. This library is installed when you install Axiom configuration software for analog and digital PC boards (Device Manager).

After installing library you must configure the board with the configuration software, Device Manager. This software will create a configuration file (default : axdrv.ini).

☞ Note: 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.

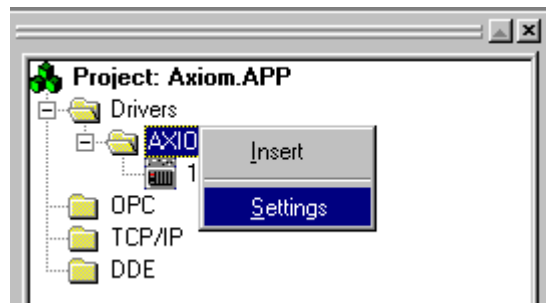
The driver configuration is two parts:

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

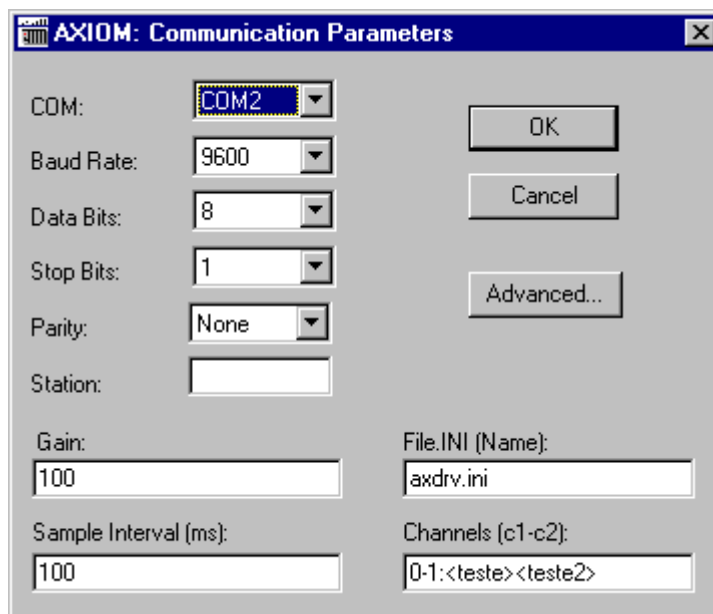
4.1 Settings - Communication Parameters

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

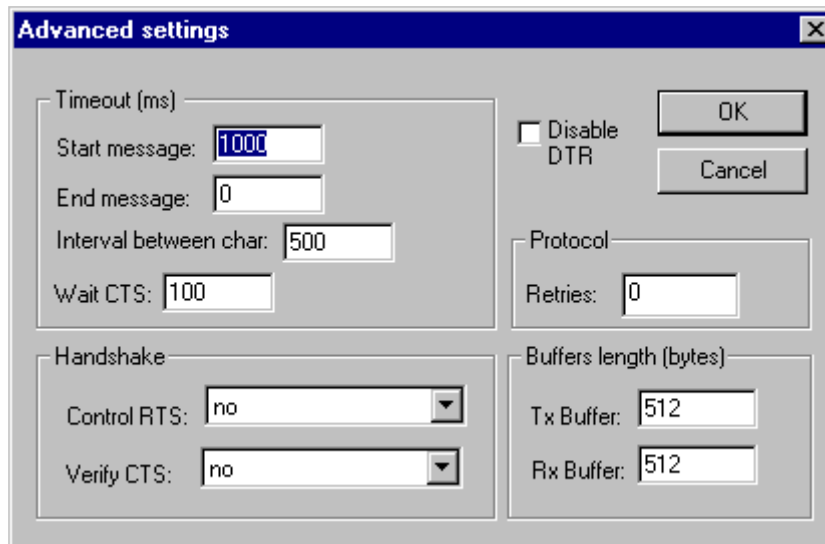


When selecting the Settings, there is the following dialog to configure:



Parameter	Default Value	Valid values	Description
COM	COM2	COM1 to COM8	Serial port of the PC used to communication with the device. <i>(Not used for this driver)</i>
Baud Rate	9600	110 to 57600bps	Communication rate of data. <i>(Not used for this driver)</i>
Data Bits	8	5 to 8	Number of data bits used in the protocol. <i>(Not used for this driver)</i>
Stop Bits	1	1or 2	Number of stop bits used in the protocol. <i>(Not used for this driver)</i>
Parity	None	even, odd, none, space or mark	Parity of the protocol. <i>(Not used for this driver)</i>
Station	0	0	<i>(Not used for this driver)</i>
File.Ini (Name)	Axdrv.ini	String value.	Name of configuration file created by Device Manager. This file have the configuration boards.
Channels (c1-c2)	-	<c1> - Numeric value. <c2> - Numeric value. <TagHiLimit>-Tag Name(string) <TagLoLimit>-Tag Name(string)	<i>Sintax:</i> <c1>-<c2>:<TagHiLimit><TagLoLimit> <c1> - Number of first channel. <c2> - Number of second channel. TagHiLimit – Tag Name with Hi limit of first channel. TagLoLimit – Tag Name with Low limit of second channel. <i>This field is specific to “T” command.</i>
Gain	1	1, 2, 4, 8, 16	Gain of channels. <i>This field is specific to “T” command.</i>
Sample Interval (ms)	0	Numeric value.	Sample interval of channels value. <i>This field is specific to “T” command.</i>

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

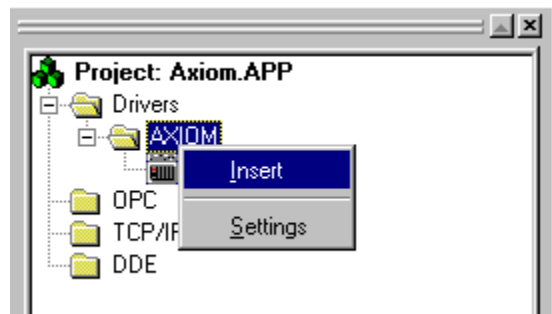


The Advanced setting parameters are explained at the Studio Technical Reference Manual, and you should keep the default values to all fields.

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



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	test[1]	INT0:data[1]		
2	test[2]	INT1:data2[1]		
3				
4				
5				
6				
7				

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	-	Numeric value	Device Number: This value is generated by the Device Manager(axiom) to any board configured.
Header	AI	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 that will be read or written from or to the device. It complies with the syntax:<Operand type> or T:<Status>:<Sample number>. After editing the field **Header**, the system will check if it is valid or not. If the syntax were incorrect, the default value (AI) 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 bellow:

- **Operand Type:** type of operands that will be read or written from or to device.
 AI - Analog Input. (read only)
 AO - analog Output. (write only)

DI - Digital Input. (read only)
 DO - Digital Output. (write only)
 TMP - Temperature. (read only)
 EVT – Event Counter. (read/write)
 FRQ - Frequency. (read/write)
 CON - Counters. (read/write)
 STP - To EVT, FRQ or CON. (Write)

- **T:<Status>:<Sample number>** : Read of analog input with dedicate treat.
 <Status> - Tag Name to receive status of values reads.
 <Sample> - Tag Name with number o sample values.

Information regarding the parameter “Header”			
Type	Sample of syntax	Valid range of initial Address	Comment
AI	AI	-	Analog input reading.
AO	AO	-	Analog output writing.
DI	DI	-	Digital input reading.
DO	DO	-	Digital output writing.
TMP	TMP	-	Temperature reading
EVT	EVT	-	Event counter reading and writing.
FRQ	FRQ	-	Frequency reading and writing.
CON	CON	-	Counters reading and writing.
STP	STP	-	Stops event counter, frequency or counters.
T	T:<ValStatus>:<NumSample>	-	<ValStatus> - Tag name to receive status of analog input reading. <NumSanple> - Tag name with the number of sample value storage in the tag of address field.

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:

- To **AI** header:
 Syntax : <channel>:<gain code>
 <channel> - analog input channel number for reading.
 <gain code> - gain code for this channel.
- To **DI** header:
 Syntax : <channel> or <channel>.<bit>
 <channel> - digital input channel number for reading.(8 bits)
 <bit> - bit value of channel reading value.
- To **TMP** header:
 Syntax: <channel>:<gain code>:<thermocouple type>:<unit>
 <channel> - temperature channel number for reading.

<gain code> - gain code for this channel.
 <thermocouple type> - J type thermocouple.
 <unit> - unit of read value.

- To **AO** header:
 Syntax: <channel>
 <channel> - analog output channel number for writing.

- To **DO** header:
 Syntax : <channel> or <channel>.<bit>
 <channel> - digital output channel number for writing.(8 bits)
 <bit> - bit number to write in digital output channel.

- To **EVT** header:
 Syntax: <Event Cnt>
 <Event Cnt> - Number of Event counter for writing.

- To **FRQ** header:
 Syntax: <Gate>:<Measurement>:<Cascade>:<GateTime>
 <Gate> - Gate(time-base) counter.
 <Measurement> - Measurement counter.
 <Cascade> - counter#2 cascade to counter#1
 <GateTime> - Time Base (milliseconds)

- To **CON** header:
 Syntax: <Cnt>
 <Cnt> - Number of Counter for writing.

- To **STP** header:
 Syntax: EVT:<cnt> or
 FRQ:<Gate>:<Measurement>:<Cascade>:<GateTime> or
 CON:<Cnt>
 <cnt> - Number of Event Counter to stop.
 <Gate> - Gate(time-base) counter.
 <Measurement> - Measurement counter.
 <Cascade> - counter#2 cascade to counter#1
 <GateTime> - Time Base (milliseconds)
 <Cnt> - Number of Counter to stop.

- To **T:<Status>:<NumSample>** header:
 Syntax: INT<channel>:<sample tag>
 <channel> - number of channel for reading.
 <sample tag> - tag name of a array type, where the sample value are stored.
 The tag in the Tag Name column will receive the last value read of device.

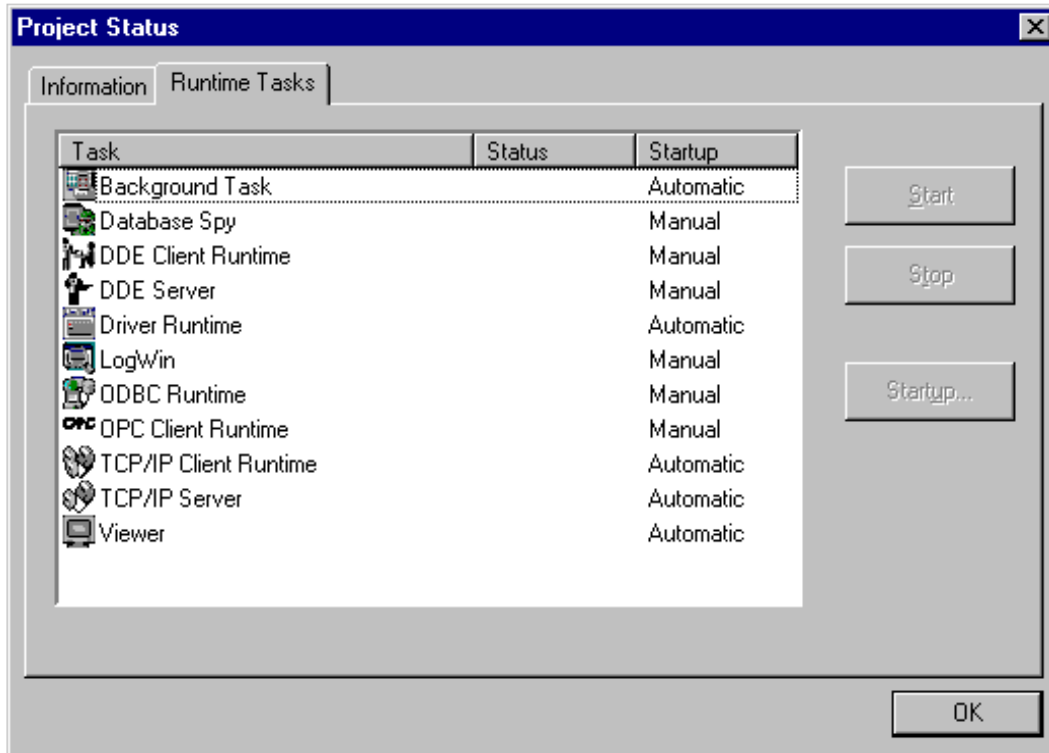
Sample of Addressing Configuration		
Address on the Device	Header Field	Address Field
AI – channel 0	AI	0:0
DI – channel 1 (8 bits)	DI	1
DI – channel 0 bit 0	DI	0.0
TMP – channel 0	TMP	0:0:0:0
AO - channel 0	AO	0
DO – channel 0 (8 bits)	DO	0
DO – channel 1 (bit 2)	DO	1.2
EVT – event counter 0	EVT	0
FRQ – frequency	FRQ	10:0:10:10
CON – counter 1	CON	1
STP – event counter 3	STP	EVT:3
STP – frequency	STP	FRQ:10:0:10:10
STP – counter 0	STP	CON:0
AI (dedicate mode –channel 0)	T:<tagsts><tagnumsample> Where: <tagsts> - Integer type tag. <tagnumsample> - Integer type tag.	INT0:valvect[1] Where: Valvect – Real array type tag with 512 position.
AI (dedicate mode –channel 1)	T:<tagsts><tagnumsample> Where: <tagsts> - Integer type tag. <tagnumsample> - Integer type tag.	INT1:tagvect[1] Where: tagvect – Real array type tag with 512 position.

4.5 Device Configuration

To configure the Axiom board it is necessary to use itself programmer (supplied by Axiom - Device Manager). The board configuration is saved in the configuration file (default: axdrv.ini) . In additional it is necessary configure the name of configuration file in the communicating parameters.

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	-
106	Error board initialize.	Driver can not initialize board.	Check the name of configuration file created by the Device Manager from Axiom.
107	Error Invalid Header	An invalid Header has been typed or the tag that is inside this field has an invalid configuration.	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
108	Error Invalid Address	An invalid Address has been typed.	Type a valid Address either on the address field . A lot of different valid headers are shown on the section 4.4
109	Error Invalid Command	- The read command was use with a write trigger. - The write command was use with a read trigger.	- Check the type of command (read or write type).
110	Error Invalid tag on channel	- The tag configured in field of communication parameters do not exist.	- Check the tag used in the communication parameters. This tag must is in the tag list of studio.
111	Error invalid syntax.	- The configuration of communication parameters is wrong.	- Verify the configuration in the communication parameters. The tags must be between <>.
112	Error reading tag.	- Error accessing tag configured in the communication parameters.	- Check the tag used in the communication parameters. This tag must is in the tag list of studio.
113	Error writing tag.	- Error accessing tag configured in the communication parameters.	- Check the tag used in the communication parameters. This tag must is in the tag list of studio.
-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-jan-00	▪ First driver version
1.02	Roberto V. Junior	20-jan-00	▪ Fixed bug of treating float data like a integer.
1.03	Roberto V. Junior	07-jan-03	▪ Fixed bug of read value.
1.04	Lourenço Teodoro	15-Jan-10	▪ Driver compiled for Unicode