GPIBN Communication Driver

Driver for Communication
with GPIB National Instruments Board

Contents

Introduction

The GPIBN driver enables communication between the Studio system and a GPIB National board, according to the specifications discussed in this document.

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

- **Introduction**: Provides an overview of the GPIBN driver documentation
- **General Information**: Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the GPIBN driver
- **Installing the Driver**: Explains how to install the GPIBN 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 product’s *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 communication between the Studio GPIBN driver and a GPIB National Board.

The information is organized into the following sections:

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

Device Characteristics

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

- **Manufacturer**: National Instruments

For a list of the devices used for conformance testing, see the "Conformance Testing" section.

Link Characteristics

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

- **Logic Protocol**: IEE 488.1
- **Device Runtime Software**: None
- **Specific PC Board**: GPIB

Driver Characteristics

The GPIBN driver is composed of the following files:

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

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

- Windows 2000
- Windows XP

For a list of the operating systems used for conformance testing, see the "Conformance Testing" section

The GPIBN driver supports the following Commands:

Commands
IBRD
IBWRT
IBWAIT
IBRSP
IBTRG
IBDEV
IBLOC

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration**:
  - **Baud Rate**: -
  - **Data Bits**: -
  - **Stop Bits**: -
  - **Parity**: -
  - **COM Port**: -

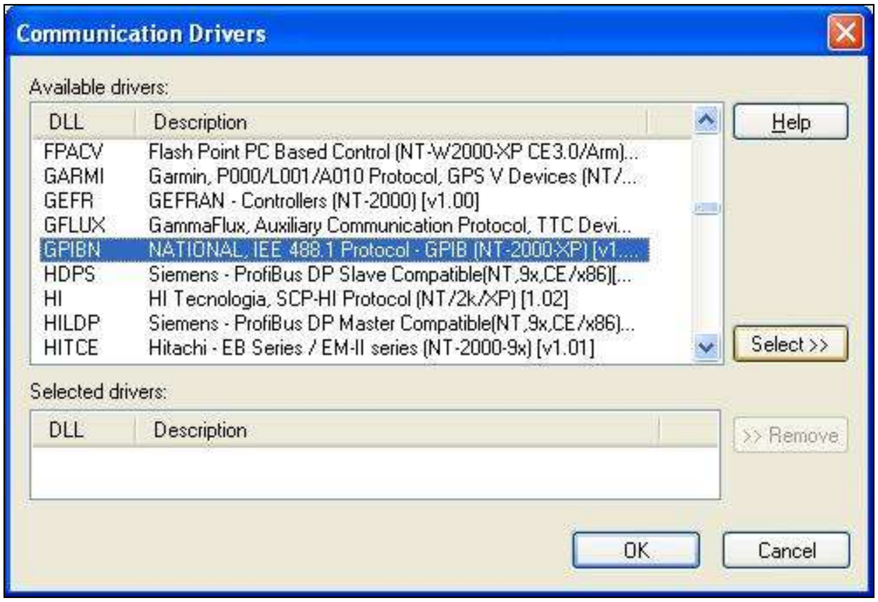
Driver Version	Studio Version	Operating System (development)	Operating System (target)	Equipment
1.20	7.1+SP3+P9	WinXP	WinXP	2 GPIB boards (multiple GPIB interface communication)

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 or double-click the Studio shortcut icon on your desktop.
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 **GPIBN** driver from the *Available Drivers* list, and then click the **Select** button.



Available drivers:

DLL	Description
FPACV	Flash Point PC Based Control (NT-W2000-XP CE3.0/Arm)...
GARMI	Garmin, P000/L001/A010 Protocol, GPS V Devices (NT/...
GEFR	GEFRAN - Controllers (NT-2000) [v1.00]
GFLUX	GammaFlux, Auxiliary Communication Protocol, TTC Devi...
GPIBN	NATIONAL, IEE 488.1 Protocol - GPIB (NT-2000-XP) [v1....
HDPS	Siemens - ProfiBus DP Slave Compatible(NT,9x,CE/x86)[...
HI	HI Tecnologia, SCP-HI Protocol (NT/2k/XP) [1.02]
HILDP	Siemens - ProfiBus DP Master Compatible(NT,9x,CE/x86)...
HITCE	Hitachi - EB Series / EM-II series (NT-2000-9x) [v1.01]

Selected drivers:

DLL	Description

Communication Drivers Dialog

5. When the **GPIBN** 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 Studio and the device.

> **Attention:**
> 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 (*GPIB NI-488.2 User Manual for Windows*, National Instruments, and/or *GPIB NI-488.2 Function Reference Manual for DOS/Windows*).

Configuring the Driver

After opening Studio and selecting the **GPIBN** driver, you must configure the driver. Configuring the GPIBN driver is done in two parts:

- Specifying communication parameters (only one configuration needed)
- Defining communication tags and controls in the Communication tables or *Driver* worksheets (Standard Driver Worksheets)

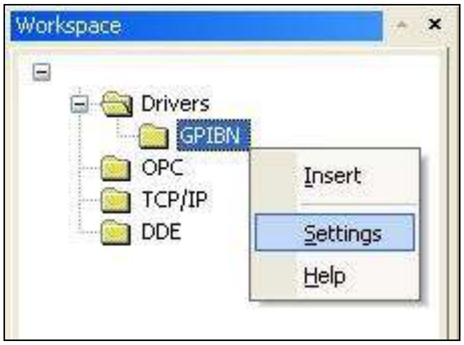
  Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**,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

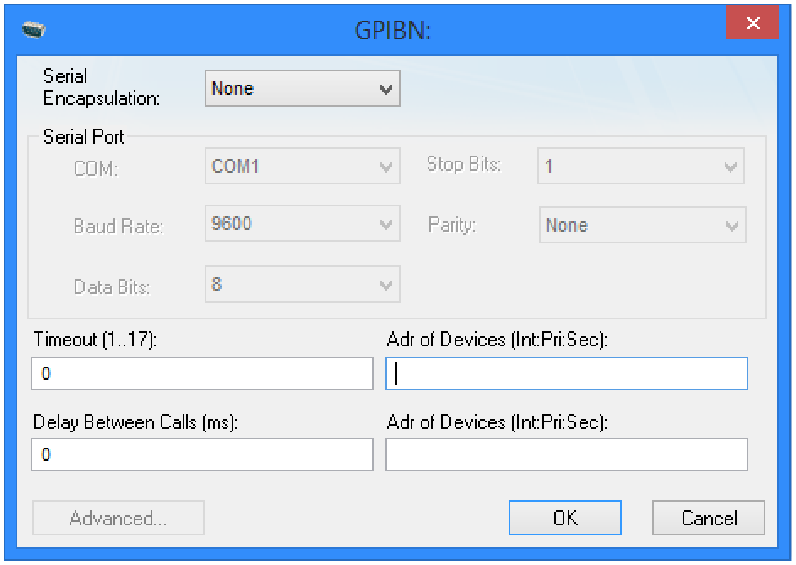
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 *GPIBN* subfolder. When the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu

The *GPIBN: Communication Parameters* dialog displays:



Serial Encapsulation: None

COM: COM1

Stop Bits: 1

Baud Rate: 9600

Parity: None

Data Bits: 8

Timeout (1..17): 0

Adr of Devices (Int:Pri:Sec):

Delay Between Calls (ms): 0

Adr of Devices (Int:Pri:Sec):

GPIBN: Communication Parameters Dialog

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

Parameters	Default Values	Valid Values	Description
Delay Between Calls (ms)	0	500..1000	Delay between calls into same sheet
Timeout	0	1..17	1 Timeout of 10µs 2 Timeout of 30µs 3 Timeout of 100µs 4 Timeout of 300µs 5 Timeout of 1ms 6 Timeout of 3ms 7 Timeout of 10ms 8 Timeout of 30ms 9 Timeout of 100ms 10 Timeout of 300ms 11 Timeout of 1s 12 Timeout of 3s 13 Timeout of 10s 14 Timeout of 30s 15 Timeout of 100s 16 Timeout of 300s 17 Timeout of 1000s
Address of Devices (Int:Pri:Sec)	-	-	List of addresses of devices, in this format <Int1>:<Pri1>:<Sec1> , <Int1>:<Pri2>:<Sec2>.. , <IntN>:<PriN>:<SecN> Here: IntN or Interface is the board number used. This is an optional parameter. If not informed the driver assumes the default value 0 PriN is the primary device address

			SecN is the secondary device address
Address of Devices (Int:Pri:Sec)	-	-	Continuation of the list of addresses of devices (same as the previous field in this table). <Int1>:<Pri1>:<Sec1> , <Int1>:<Pri2>:<Sec2>] .. , <IntN>:<PriN>:<SecN> Here: IntN or Interface is the board number used. This is an optional parameter. If not informed the driver assumes the default value 0 PriN is the primary device address SecN is the secondary device address

Examples of device addresses:

- **1** (interface/board 0 with primary address 1)
- **1:2** (interface/board 0 with primary address 1 with secondary address 2)
- **3:1:2** (interface/board 3 with primary address 1 with secondary address 2)
- **1:10** (interface/board 0 with primary address 1, and interface 0/board with primary address 10)
- **1:2,10:20** (interface 0 with primary address 1 with secondary address 2, and interface 0 with primary address 10 with secondary address 20)
- **2:1:2,3:10:20** (interface 2 with primary address 1 with secondary address 2, and interface 3 with primary address 10 with secondary address 20)

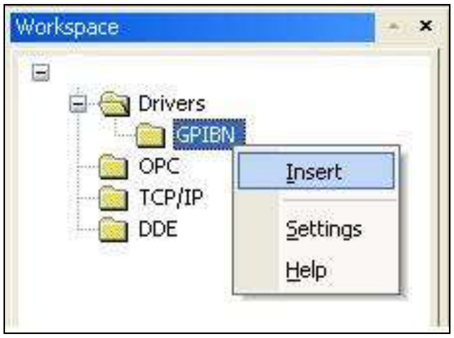
Note:
These parameters must be configured *exactly the same* as those you configured for the GPIBN driver in the *Communications Parameters* dialog.

Configuring the Standard 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:

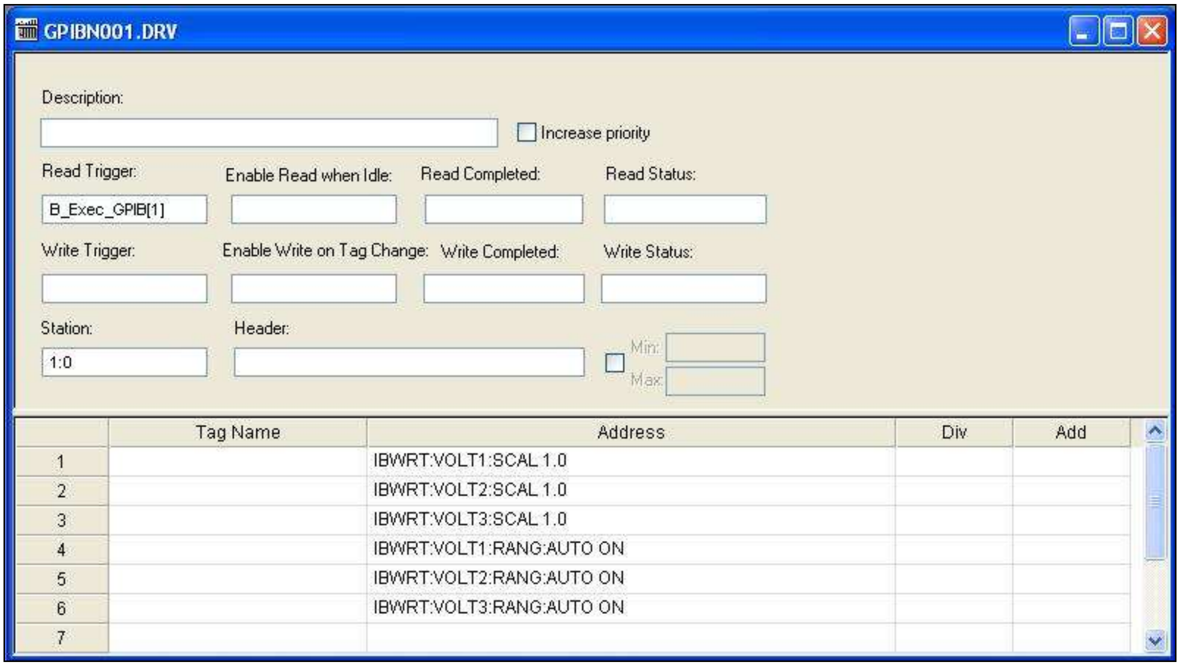
1. From the Studio development environment, select the **Comm** tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder, and right-click the *GPIBN* subfolder.
3. When the pop-up menu displays, select the **Insert** option:



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.

The *GPIBN.drv* dialog displays (similar to the following figure):



GPIBN001.DRV

Description:		☐ Increase priority	
Read Trigger: B_Exec_GPIB[1]	Enable Read when Idle:	Read Completed:	Read Status:
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:
Station: 1:0	Header:	☐ Min: / Max:	

	Tag Name	Address	Div	Add
1		IBWRT:VOLT1:SCAL 1.0		
2		IBWRT:VOLT2:SCAL 1.0		
3		IBWRT:VOLT3:SCAL 1.0		
4		IBWRT:VOLT1:RANG:AUTO ON		
5		IBWRT:VOLT2:RANG:AUTO ON		
6		IBWRT:VOLT3:RANG:AUTO ON		
7				

GPIBN Driver Worksheet

4. Use the following information to complete the **Station**, **Header** and **Address** fields on this worksheet:

– **Station** field: Address of device. It must comply with the following syntax:

`<Interface>:<Primary Device>`

Where:

- `<Interface>` is the number of the board. This is an optional parameter. If not informed the driver assumes the default value 0
- `<Primary Device>` is the number that identifies the primary device address

Examples:

- **1** (interface 0 with primary device address 1 – it will also use a secondary address, if it had been configured in the Driver Settings > Adr. Of Devices (Pri:Sec) parameter, pairing with the primary address 1)
- **10** (interface 0 with primary device address 10 – it will also use a secondary address, if it had been configured in the Driver Settings > Adr. Of Devices (Pri:Sec) parameter, pairing with the primary address 10)
- **3:1** (interface 3 with primary device address 1 – it will also use a secondary address, if it had been configured in the Driver Settings > Adr. Of Devices (Pri:Sec) parameter, pairing with the interface 3 and primary address 1)

Note:

The Interface (Int) is an optional parameter in the Settings field *Address of Devices (Int:Pri:Sec)* of the driver. If not informed the driver assumes the default value 0.

In this case for successful communication with interface 0, primary device address 1, secondary device address 2: The user can define *1:2* or *0:1:2* in the Settings field of the *Address of Devices (Int:Pri:Sec)* and use *0:1* or *1* as the *Station* field on the driver sheet. The secondary device address 2 is used since it was configured in the Settings field.

– **Header** field: Not used

– **Address** field: The body of the *Driver* worksheet allows you to associate each tag to its respective address in the device. 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:

```
<Command>:<Parameter>
```

Where:

- `<Command>`is the command sent to the device. Valid values: **IBRD**, **IBWRT**, **IBWAIT**, **IBRSP**, **IBTRG**, **IBDEV** and **IBLOC**
- `<Parameter>`is the complement of the command.

Command	Parameters
IBRD	`<Tag Name - String Type>:<Number of bytes waited >`
IBWRT	`<Command String>` If there are \\ (two back slashes) in the Command String, just send one \. E.g. `IBWRT:Teste\\Driver` ➔ `Teste\Driver` If there are \ (one back slash) and two characters in the Command String, send just the two characters in Hexa. E.g. IBWRT:Teste\\Driver\0d ➔ Teste\Driver<byte 0D hexa>
IBWAIT	`<Event Code>`
IBRSP	Not Used
IBTRG	Not Used
IBDEV	`<Configuration String>`
IBLOC	Not Used

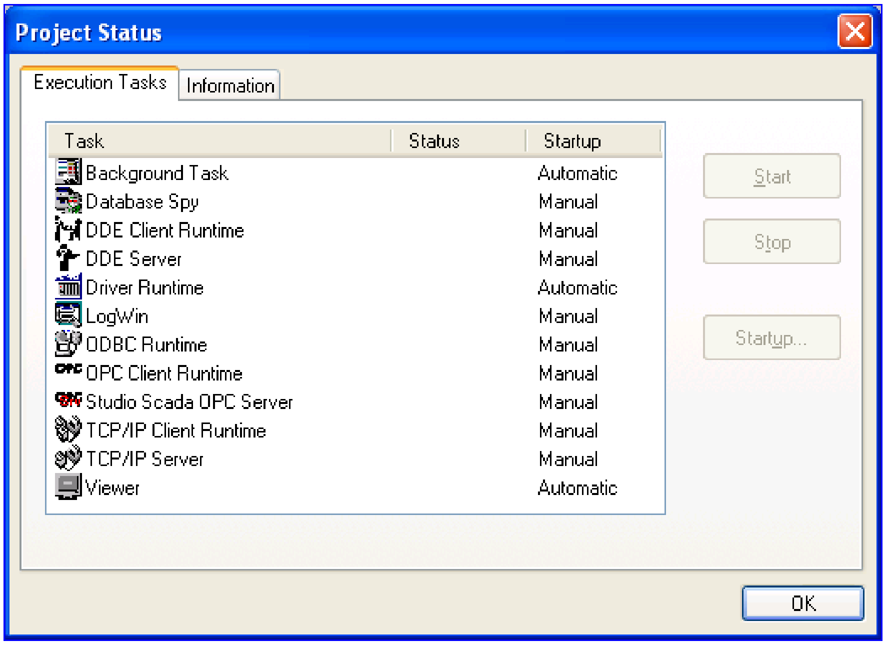
Executing the Driver

After adding the GPIBN 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 displays:



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

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 GPIBN 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	Error in the IBDEV function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
2	Error in the IBCLR function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
3	Error in the IBWRT function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
4	Error in the IBRD function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
5	Error in the IBTRG function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
6	Error in the IBWAIT function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
7	Error in the IBRSP function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
8	Error in the IBLOC function	Configuration of the Driver Sheet is wrong	Verify the configuration of the Driver Sheet.
9	Error loading DLL.	Board is not installed correctly	Verify the installation of the board.
10	Error in the initialization	Board is not installed correctly	Verify the installation of the board.
20	Invalid Address	Address is not configured correctly	Verify the configuration of the Address field.
23	Too many devices configured in Communication Parameters	Many devices configured in Communication Parameters	Verify the configuration of devices in Communication Parameters.
24	Station not configured in Communication Parameters	Station not configured in Communication Parameters	Configure the Station in Communication Parameters.
-15	Timeout Start Message	▪ Disconnected cables ▪ PLC is turned off, in stop mode or in error mode. ▪ Wrong station number ▪ Wrong RTS/CTS control settings	▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see the Studio *Technical Reference Manual* for valid configurations).
-17	Timeout between rx char	▪ PLC is in stop mode or in error mode. ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings	▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the RTS/CTS configuration (see "Link Characteristics" for valid RTS/CTS configurations).

⇨ **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 **Protocol Analyzer,** right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events.

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**: This 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

This driver does not have a sample application

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.14	Eric Vigiani	Apr / 14 / 2005	Modified the IBWRT command from 32 to 256 characters
B	1.15	Fellipe Peternella	Mar / 13 /2009	Modified Driver to respect Character Case on the Commands
C	1.16	Paulo Balbino	Sept/16/ 2011	Fixed Time Stamp function
D	1.17	André Körbes	Apr/4/2012	Increased the maximum number of devices to 32
E	1.18	André Körbes	May/22/2013	Improved driver safety
F	1.19	Eduardo Castro	Feb/01/2016	Improved read operation stability
G	1.20	Eduardo Castro	Jan/09/2017	Added support for multiple boards/interfaces