GAUGE Communication Driver

Driver for Serial Communication
With Devices Using Gauge devices

Contents

Introduction

The GAUGE driver enables communication between the Studio system and devices using the Gauge protocol communicating over serial, according to the specifications discussed in this document.

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

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

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**: Vacuum Gauges Technology
- **Compatible Equipment**:
  - Vacuum Gauge – CM3

Link Characteristics

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

- **Device Communication Port**: Serial Port
- **Physical Protocol**: Serial
- **Logic Protocol**: Gauge
- **Device Runtime Software**: None

Driver Characteristics

The GAUGE driver is composed of the following files:

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

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

- Windows 9x
- Windows 2000
- Windows NT

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

The GAUGE driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	DWord
H	String	–	•	–	–	–
K	String	–	•	•	•	•
F	String	•	–	•	•	•
P	String	•	–	•	•	•
R	String	•	–	•	•	•
S	String	–	•	•	•	•

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration**:
  - **PLC Program**: Inficon – Vacuum Gauge CM3
  - **Baud Rate**: 9600
  - **Protocol**: Inficon
  - **Data Bits**: 8
  - **Stop Bits**: 1
  - **COM Port**: COM1

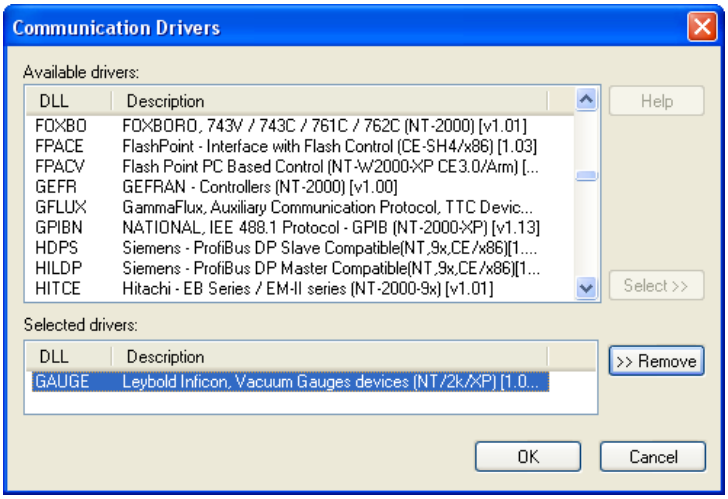
Driver Version	Studio Version	Operating System (development)	Equipment
1.01	6.0	Windows XP SP2	Vacuum Gauge

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.



Communication Drivers Dialog Box

4. Select the **GAUGE** driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.
5. When the **GAUGE** 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 the host 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.

Configuring the Driver

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

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

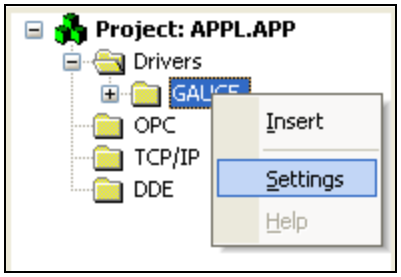
  Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**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

> **Note:**
> For a detailed description of the Studio *MAIN* and *STANDARD DRIVER SHEET*s, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

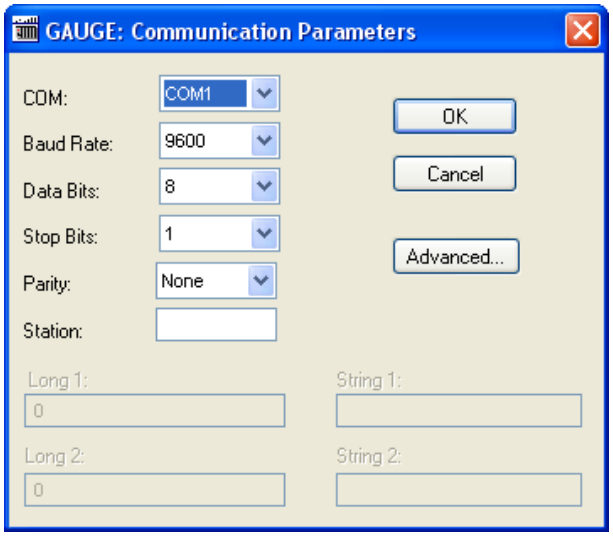
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 Gauge subfolder. When the pop-up menu displays (as shown in the following figure), select the **Settings** option.



Select Settings from the Pop-Up Menu

The *GAUGE: Communication Parameters* dialog displays (as follows).



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

> **Note:**
> The device must be configured with *exactly the same* parameters that you configured in the *GAUGE Communication Parameters* dialog.

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

> **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

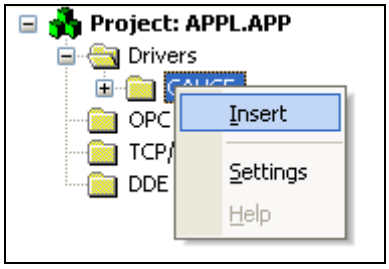
This section explains how to configure the *STANDARD DRIVER SHEETs* (or communication tables) to associate application tags with the device addresses. You can configure multiple Driver Worksheets — each of which is divided into a *Header* section and *Body* section.

Configuring the *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. In the *Workspace* pane, expand the *Drivers* folder and right-click the *GAUGE* subfolder.

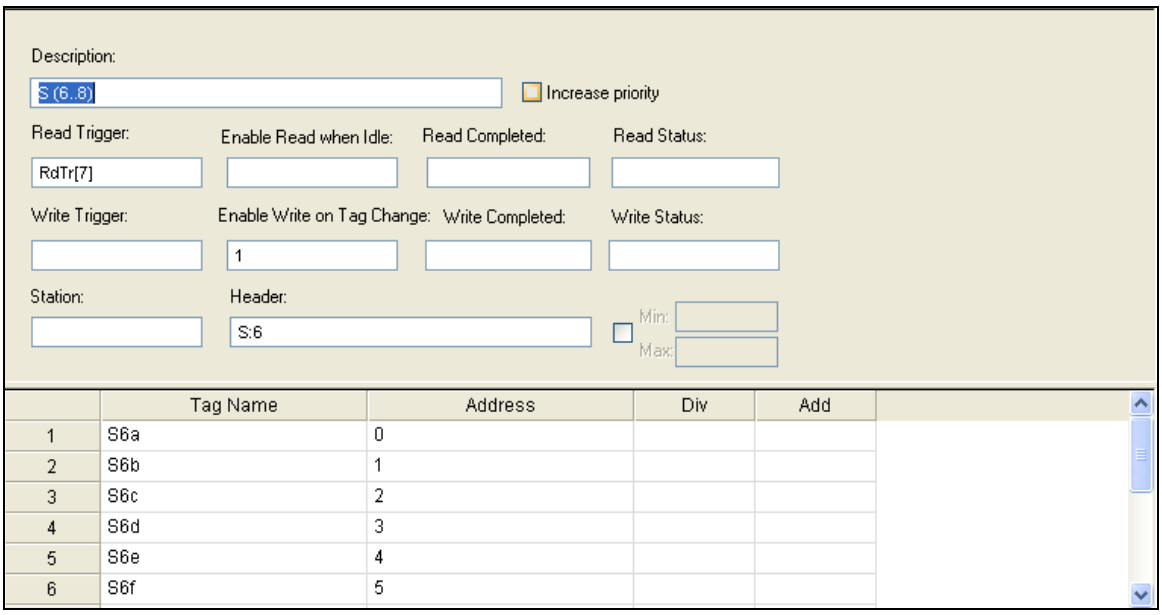
When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



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 *STANDARD DRIVER SHEET* displays (similar to the following figure).



STANDARD DRIVER SHEET

In general, all parameters on the Driver Worksheet (except the **Station**, **Header** and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. For detailed information about configuring the standard parameters, consult the Studio *Technical Reference Manual*.

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

   - **Station** field: Not used
   - **Header** field: Use the information in the following table to define the type of variables that will be read from or written to the device, and a reference to the initial address. Default value is **`K:N`**.

     These variables must comply with the following syntax:

   * For **H, K** Data Type:

     `<Type>` (For example: **`H`**)

   * For **F, R** Data Type:

     `<Type>` (For example: **`F`**)It also requires the ID number in the address field

   * For **S, P** Data Type:

     `<Type>:<Id>` (For example: **`S:00`**)

Where:

- **Type** is the register type. (**K, H, F, P, R, S**)
- **Id** is the identification command of the configured type.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

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 GAUGE driver.

Header Field Information			
Data Types	**Sample Syntax**	**Valid Range of Initial Addresses per Worksheet**	**Comments**
H	H	-	HELLO – Determine Presence and type of instrument
K	K	-	Communications Diagnostics
F	F:64	Varies according to the equipment	Read Parameter Value
P	P:64	Varies according to the equipment	Set Parameter Value
R	R:00	Varies according to the equipment	Remote Command Send
S	S:00	Varies according to the equipment	Read Equipment Status

- **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:

For F Data Type:

Address	Description	Response format - STRING
64	Relay 1 set point, upper	Char 1 – ACK Char 2 – Sensor # "1", "2", "3" Char 3 – Always space Char 4 – MSD data field Char 5 thru 7 – Rest of data field Char 8 – Always "E" character Char 9 – Exponent sign "+", "-" Char 10 – MSD of exponent data Char 11 – LSD of exponent data
65	Relay 1 set point, lower	
66	Relay 2 set point, upper	
67	Relay 2 set point, lower	
68	Relay 3 set point, upper	
69	Relay 3 set point, lower	
70	Linear exponent sensor 1	Char 1 – ACK Char 2 – Exponent sign "+", "-" Char 3 – Number digit
71	Linear exponent sensor 2	
72	Linear exponent sensor 3	

73	Gain factor. (ion, cold cathode)	Char 1 – ACK Char 2 – MSD data Char 3 – Data Char 4 – Always decimal point Char 5 – LSD data
74	Secs Timer 1, character timeout	
75	Secs Timer 2, protocol timeout	
76	Secs Retry Counter	Char 1 – ACK Char 2 – MSD data Char 3 – LSD data
77	Secs upper device ID	Char 1 – ACK Char 2 – MSD data Char 3 – Data Char 4 – LSD data
78	Secs lower device ID	
79	Crossover parameter	Char 1 – ACK
80	Crossback parameter	Char 2

Code	Torr	Millibar	Pascal
01	1E-3	1.3E-3	1.3E-1
02	2E-3	2.6E-3	2.6E-1
03	3E-3	3.9E-3	3.9E-1
04	4E-3	5.3E-3	5.3E-1
05	5E-3	6.6E-3	6.6E-1
06	6E-3	7.9E-3	7.9E-1
07	7E-3	9.3E-3	9.3E-1
08	8E-3	1.06E-2	1.06E+0
09	9E-3	1.19E-2	1.19E+0
10	10E-3	1.33E-2	1.33E+0

81	Autozero parameter	Char 1 – ACK

Char 2	Code	Torr	Millibar	Pascal
	01	1E-5	1.33E-5	1.33E-3
	02	2E-5	2.66E-5	2.66E-3
	03	3E-5	3.99E-5	3.99E-3
	04	4E-5	5.33E-5	5.33E-3
	05	5E-5	6.66E-5	6.66E-3
	06	6E-5	7.99E-5	7.99E-3
	07	7E-5	9.33E-5	9.33E-3
	08	8E-5	1.06E-4	1.06E-2
	09	9E-5	1.19E-4	1.19E-2
	10	1E-4	1.33E-4	1.33E-2

For P Data Type:

Address Offset	Description	Command format - STRING
64	Relay 1 set point, upper	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character Char 5 – Sensor # (1, 2, 3) Char 6 – Always space character Char 7 – MSD data Char 8 – Always decimal point Char 9 thru 10 – LSD's data Char 11 – "E" character Char 12 – Exponent sign "+", "-" Char 13 thru 14 – Exponent data MSD first
66	Relay 2 set point, upper	
68	Relay 3 set point, upper	
65	Relay 1 set point, lower	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character Char 5 – MSD data Char 6 – Always decimal point Char 7 thru 8 – LSD's data Char 9 – "E" character Char 10 – Exponent sign "+", "-" Char 11 thru 12 – Exponent data MSD first
67	Relay 2 set point, lower	
69	Relay 3 set point, lower	
70	Linear recorder exponent sensor 1	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character Char 5 – Exponent sign "+", "-" Char 6 – MSD exponent data Char 7 – LSD exponent data
71	Linear recorder exponent sensor 2	
72	Linear recorder exponent sensor 3	
73	Gain factor. (ion, cold cathode)	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character Char 5 – MSD data Char 6 – Always decimal point Char 7 thru 8 – LSD data
74	Secs Timer 1, character timeout	
75	Secs Timer 2, protocol timeout	
76	Secs Retry Counter	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character Char 5 – MSD data Char 6 – LSD data
79	Crossover parameter	Char 1 – "P" Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character

Address Offset	Description	Command format - STRING				
80	Crossback parameter	Char 5 thru 6	**Code**	**Torr**	**Millibar**	**Pascal**
			01	1E-3	1.3E-3	1.3E-1
			02	2E-3	2.6E-3	2.6E-1
			03	3E-3	3.9E-3	3.9E-1
			04	4E-3	5.3E-3	5.3E-1
			05	5E-3	6.6E-3	6.6E-1
			06	6E-3	7.9E-3	7.9E-1
			07	7E-3	9.3E-3	9.3E-1
			08	8E-3	1.06E-2	1.06E+0
			09	9E-3	1.19E-2	1.19E+0
			10	10E-3	1.33E-2	1.33E+0
81	Autozero parameter	Char 1 – “P” Char 2 – MSD ID# Char 3 – LSD ID# Char 4 – Always space character				
		Char 5 thru 6	**Code**	**Torr**	**Millibar**	**Pascal**
			01	1E-5	1.33E-5	1.33E-3
			02	2E-5	2.66E-5	2.66E-3
			03	3E-5	3.99E-5	3.99E-3
			04	4E-5	5.33E-5	5.33E-3
			05	5E-5	6.66E-5	6.66E-3
			06	6E-5	7.99E-5	7.99E-3
			07	7E-5	9.33E-5	9.33E-3
			08	8E-5	1.06E-4	1.06E-2
			09	9E-5	1.19E-4	1.19E-2
			10	1E-4	1.33E-4	1.33E-2

For S Data Type:

Address Offset	Description	Address	Response format - STRING
00	Read pressure sensor 1	-	Char 1 – ACK Char 2 Thru 6 – Value data MSD first, decimal point floats Char 7 – “E” character Char 8 – Exponent sign “+”, “-” Char 9 Thru 10 – Exponent value MSD first
01	Read pressure sensor 2		
02	Read pressure sensor 3		
03	Read sensor 1 type	**Address**	**Description**
04	Read sensor 2 type	0	Capacitance diaphragm gauge 1000 torr
05	Read sensor 3 type	1	Capacitance diaphragm gauge 100 torr
		2	Capacitance diaphragm gauge 10 torr
		3	Capacitance diaphragm gauge 1 torr
		4	Pirani
		5	Cold cathode
		6	Ion
06	Read board configuration sensor 1	**Address**	**Description**
		0	AutoRange, 1 = off, 0 = on
		1	AutoEmis, 1 = off, 0 = on

<table>
<tr><th>Address Offset</th><th>Description</th><th>Address</th><th>Response format - STRING</th></tr>
<tr><td>07</td><td>Read board configuration sensor 2</td><td rowspan="2">2
3
4
5
6</td><td rowspan="2">Lin/log recorder, 1 = log, 0 = linear
Nitrogen/Argon, 1 = Nitrogen, 0 = Argon
AutoDegas, 1 = off, 0 = on
Gain, 1 = Default, 0 = Programmable gain
AutoZero, 1 = off, 0 = on</td></tr>
<tr><td>08</td><td>Read board configuration sensor 3</td></tr>
<tr><td>09</td><td>Read errors in sensor 1</td><td rowspan="3">-</td><td rowspan="3">Char 1 – ACK
Char 2 thru 3 – Error code (represented below)
<table><tr><th>Code</th><th>Description</th></tr><tr><td>00</td><td>No errors</td></tr><tr><td>10</td><td>Emission error</td></tr><tr><td>11</td><td>180 volt error</td></tr><tr><td>12</td><td>Degas error</td></tr><tr><td>13</td><td>Cold cathode turn on error</td></tr><tr><td>20</td><td>Over pressure</td></tr><tr><td>21</td><td>Cable disconnected</td></tr><tr><td>22</td><td>Emission off</td></tr></table></td></tr>
<tr><td>10</td><td>Read errors in sensor 2</td></tr>
<tr><td>11</td><td>Read errors in sensor 3</td></tr>
<tr><td>12</td><td>CPU configuration</td><td>Address
0
1
2
3
4</td><td>Description
Baud: 00 = 300, 01 = 1200, 10 = 2400, 11 = 9600
1 = RS232, 0 = SECS
Units: 01 mBar, 10 = pascal, 11 = torr
Keyboard lock, 1 = off, 0 = on
Notation: 1 = scientific, 0 = engineering</td></tr>
<tr><td>13</td><td>Get displayed sensor number</td><td>-</td><td>Char 1 – ACK
Char 2 – 1, 2 or 3</td></tr>
<tr><td>14</td><td>Get emission status</td><td>-</td><td>Char 1 – ACK
Char 2 – Code number, 0 = off, 1 = on</td></tr>
<tr><td>15</td><td>Get degas status</td><td>-</td><td>Char 1 – ACK
Char 2 – Code number, 0 = off, 1 = on</td></tr>
<tr><td>16</td><td>Read I/O board status</td><td>Address
0
1
2
3</td><td>Description
1 to 0 = select sensor 1
1 to 0 = select sensor 2
1 to 0 = select sensor 3
1 to 0 = emission on.0 to 1 = emission off.</td></tr>
<tr><td>17</td><td>Read relay output status</td><td>Address
0
1
2
3</td><td>Description
System Relay, 0 = de-energized, 1 = energized
Relay 1, 0 = de-energized, 1 = energized
Relay 2, 0 = de-energized, 1 = energized
Relay 3, 0 = de-energized, 1 = energized</td></tr>
<tr><td>18</td><td>Read zero mode adjustment indicator</td><td>-</td><td>Char 1 – ACK
Char 2 – Sign of trend “+” or “-”
Char 3 – MSD data (0 or 1)
Char 4 – LSD data (0 thru 9)</td></tr>
<tr><td>19</td><td>Get communications errors</td><td>Address
0
1
2</td><td>Description
RS232 buffer full, 0 = no error, 1 = error
IEEE488 buffer full, 0 = no error, 1 = error</td></tr>
</table>

Address Offset	Description	Address	Response format - STRING
			RS232 framing or overrun error, 0 = no error, 1 = error

For R Data Type:

Address	Description
00	Display sensor 1
01	Display sensor 2
02	Display sensor 3
03	Lock out parameters
04	Unlock parameters
05	Zeroing on
06	Zeroing off
07	Increase zero
08	Decrease zero
09	Turn emission on
10	Turn emission off
11	Turn degas on
12	Turn degas off
13	Lockout zero Adj.
14	Unlock zero Adj.
15	Enable sensor #2 autozero
16	Disable sensor #2 autozero
17	Enable sensor #3 autozero
18	Disable sensor #3 autozero

Where:

– **AddressOffset** is a parameter added to the **AddressReference** parameter (configured in the **Header** field) to compose the group address configured in the **Header** field.

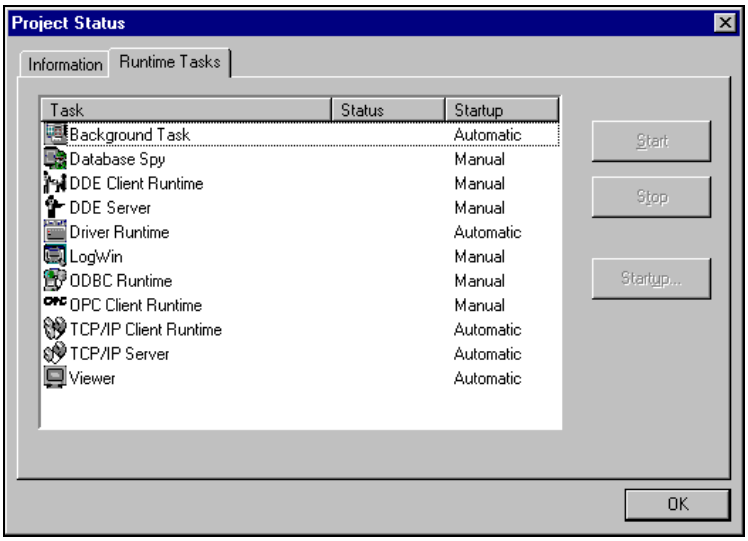
Executing the Driver

After adding the GAUGE 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 box displays, as follows.



Project Status Dialog Box

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, click **OK** to close the dialog box.
   - 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 GAUGE 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 the failure that occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required
1	ERROR OF PROTOCOL	Unexpected data	▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. Check the configuration. See Studio *Technical Reference Manual* for information about valid RTS/CTS configurations.
2	INVALID COMMAND	The command is not valid for this driver	Check the commands available for this driver.
3	ILLEGAL COMMAND CODE	The command is not valid for this driver	Check the commands available for this driver.
5	ILLEGAL COMMAND ID NUMBER	Invalid identification number for the command	Check the identification number in the header field or in the address field, according to the command.
9	Checksum error	Protocol error	Contact your Studio technical support representative.
-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 configuration. See Studio *Technical Reference Manual* for information about valid RTS/CTS configurations.
-17	Timeout between rx char	▪ PLC 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 configuration. See Studio *Technical Reference Manual* for information about 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 **Serial Communication**, 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 box.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands** and **Serial Communication** for the LogWin window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application for drivers in the **/COMMUNICATION EXAMPLES/GAUGE** directory. We strongly recommend that you check for a sample application for this driver and use it to test the driver before configuring your own customized application, for the following reasons:

- To better understand the information provided in each section of this document.
- To verify that your configuration is working satisfactorily.
- To certify that the hardware used in the test (device, adapter, cable and PC) is working satisfactorily before you start configuring your own, customized applications.

> **Note:**
> This application sample is not available for all drivers.

Use the following procedure to perform the test:

1. Configure the device's communication parameters using the manufacturer's documentation.
2. Open and execute the sample application.

> **Tip:**
> You can use the sample application screen as the maintenance screen for your custom applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Fabio Pereira de Carvalho	Mar/28/2005	Initial version