

SSUE Communication Driver

Driver for Serial Communication with Devices
Using the Serial Output of User E Protocol

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS.....3

 DRIVER CHARACTERISTICS3

INSTALLING THE DRIVER4

CONFIGURING THE DRIVER5

 SETTING THE COMMUNICATION PARAMETERS5

 CONFIGURING THE DRIVER WORKSHEETS7

EXECUTING THE DRIVER9

TROUBLESHOOTING 10

SAMPLE APPLICATION 11

REVISION HISTORY..... 12

Introduction

The SSUE driver enables communication between the Studio system and devices communicating over RS232 using the Serial Output of User E, according to the specifications discussed in this document.

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

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

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

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

- **Manufacturer:** Any device using the Serial Output of User E protocol communicating over RS232
- **Compatible Equipment:** Cemig

Link Characteristics

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

- **Device Communication Port:** Serial Output of User E
- **Physical Protocol:** Serial (RS232)
- **Device Runtime Software:** None
- **Specific PC Board:** None

Driver Characteristics

The SSUE driver is composed of the following files:

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

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

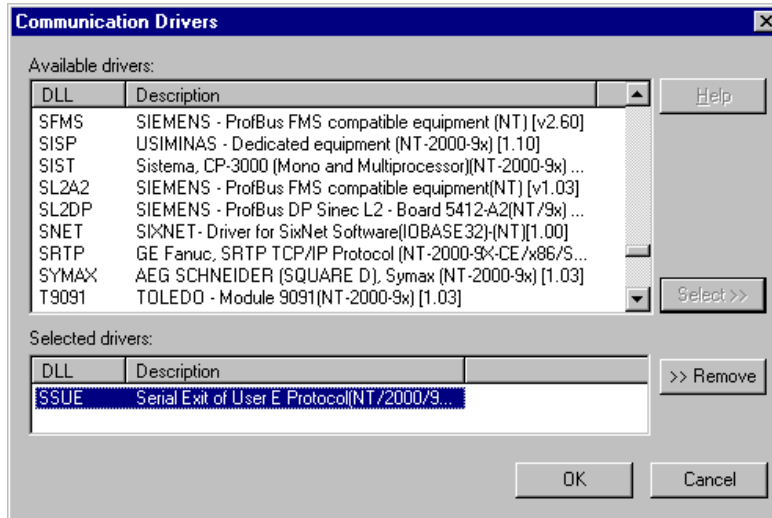
- Windows 9x
- Windows 2000
- Windows NT

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.
4. Select the **SSUE** driver from the *Available Drivers* list, and then click the **Select** button:



Communication Drivers Dialog

5. When the **SSUE** 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 SSUE driver, you must configure the driver. Configuring the SSUE driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEETS* (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.



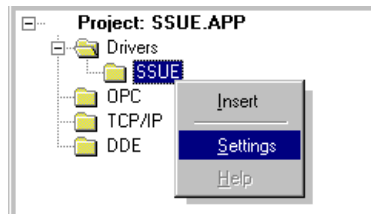
Note:

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

Setting the Communication Parameters

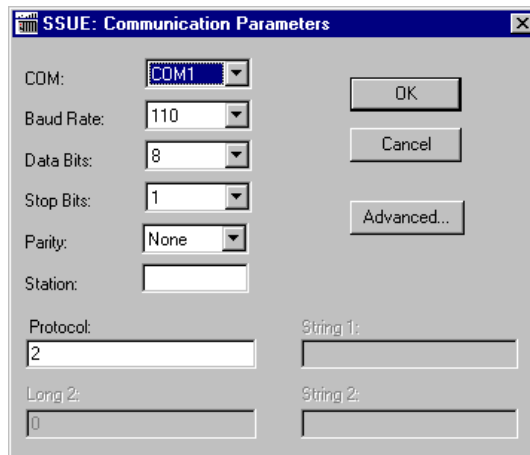
Use the following steps to configure the communication parameters, which are valid for all *Driver* worksheets configured in the system:

1. From the Studio development environment, select the **Comm** tab located below the *Workspace* pane.
2. Click on the *Drivers* folder in the *Workspace* pane to expand the folder.
3. Right-click on the *SSUE* subfolder and when the pop-up menu displays, select the **Settings** option:



Select Settings from the Pop-Up Menu


The *SSUE: Communications Parameters* dialog displays:




Communication Parameters Dialog

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

Parameters	Default Values	Valid Values	Description
Protocol	2	2	Serial Output Extended

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

5. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings that are 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, driver, and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

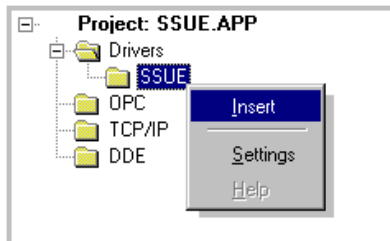
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEET*s (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.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *SSUE* subfolder.
3. When the pop-up menu displays, select the **Insert** option:



Inserting a New Worksheet

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

Description:				
<input type="text" value="Serial Exit Extended"/>				<input type="checkbox"/> Increase priority
Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Station:	Header:	<input type="checkbox"/> Min: <input type="text"/>		
<input type="text"/>	<input type="text" value="EXTENDED"/>	<input type="checkbox"/> Max: <input type="text"/>		

	Tag Name	Address	Div	Add
1	tag[1]	0		
2	tag[2]	1		
3	tag[3]	2		
4	tag[4]	3		
5	tag[5]	4		
6	tag[6]	5		
7	tag[7]	6		
8	tag[8]	7		
9	tag[9]	8		

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*.

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

- **Header** field: Use the information in the table below to define the type of variables that will be read from the device.

These variables must comply with the following syntax:

<Type of Serial Output>

Type of valid Serial Output: **EXTENDED** (to configure in agreement) was configured in the *Communication* parameters.

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.

- **Address** field: Use this field 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:

EXTENDED Header:

Address Field	Description
0	Number of seconds until the end of the interval of demand of current active energy.
1	Invoice pointer. It is complemented to each spare operation of demand.
2	Pointer of end of UFER interval. It is complemented to each end of reactive interval.
3	Reactive rank : 0 = none 1 = capacitive 2 = inductive 3 = both
4	Hourly rank : 0 = reserved (c) 1 = tip (a) 2 = outside-tip (b) 3 = 4(rank D)
5	Quadrant : 1 = quadrant 1 2 = quadrant 2 3 = quadrant 3 4 = quadrant 4
6	Tariff of Reactive: 0 = deactivated 1 = activated
7	Number of pulses of Active Energy since the beginning of the interval of current demand (pulses kWh-d, if quadrant 1 or 4; pulses kWh-r, if quadrant 2 or 3).
8	Number of pulses of Reactive Energy since the beginning of the interval of current demand (pulses kVArh-Q1, if quadrant 1; pulses kVArh-Q2 if quadrant 2; pulses kVArh-Q3, if quadrant 3; and pulses kVArh-Q4, if quadrant 4).

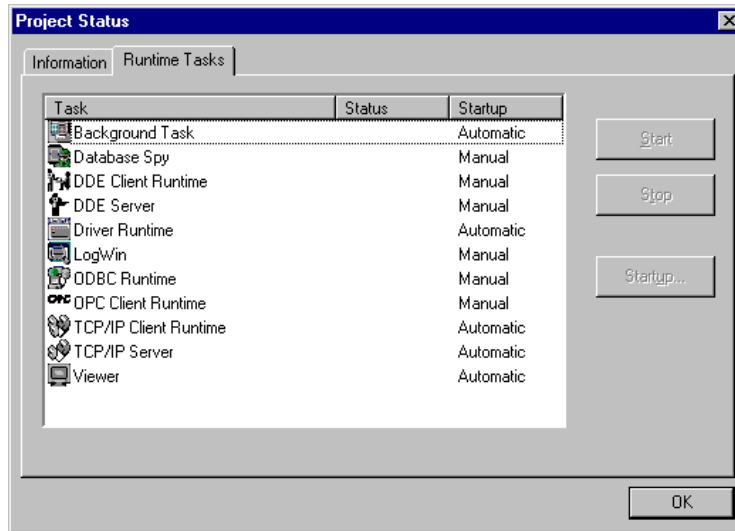
Executing the Driver

After adding the SSUE driver to a project, Studio sets the project to execute the driver automatically when you start the Runtime Environment.

To verify that the driver *Runtime 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:



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 SSUE 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.
4	Invalid Address	The sheet was configured with the wrong Address.	Check that the address is correct.
5	Invalid Command	Invalid read/write commands.	Read/write commands are not needed; the Driver automatically receives messages from the equipment.
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected cables. ▪ PLC is turned off, in stop mode, or in error mode. ▪ Wrong station number. ▪ Wrong RTS/CTS control settings. 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode. ▪ Wrong station number. ▪ Wrong parity. ▪ Wrong RTS/CTS configuration settings. 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the configuration. ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> 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.
- **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/<Driver Name> directory. We strongly recommend that you check if there is 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 the 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	Eric Vigiani	Aug/26/2003	Initial version.
B	1.01	Eric Vigiani	Feb/13/2004	Modified the quadrant parameters in the EXTENDED header.