

**SPA Communication Driver**

Driver for SPA-bus Communication Protocol

**Contents**

**INTRODUCTION .....2**

**GENERAL INFORMATION.....3**

    DEVICE CHARACTERISTICS .....3

    LINK CHARACTERISTICS .....3

    DRIVER CHARACTERISTICS .....3

    CONFORMANCE TESTING .....4

**INSTALLING THE DRIVER .....5**

**CONFIGURING THE DRIVER .....6**

    SETTING THE COMMUNICATION PARAMETERS .....6

    CONFIGURING THE DRIVER WORKSHEETS .....8

    DEVICE CONFIGURATION ..... 10

**EXECUTING THE DRIVER ..... 11**

**TROUBLESHOOTING ..... 12**

**SAMPLE APPLICATION ..... 14**

**REVISION HISTORY..... 15**

## Introduction

The SPA driver enables communication between the Studio system and devices using the SPA-bus Communication Protocol, according to the specifications discussed in this document.

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

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

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:** ABB Relays
- **Compatible Equipment:** Any device that is fully compatible with the SPA-bus Communication protocol
- **Programmer Software:** None

For a list of the devices used for conformance testing, see “Conformance Testing”.

### Link Characteristics

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

- **Device Communication Port:** Serial Port
- **Physical Protocol:** RS232
- **Logic Protocol:** SPA-bus
- **Device Runtime Software:** None
- **Specific PC Board:** None

### Driver Characteristics

The SPA driver is composed of the following files:

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

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

- Windows XP

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

The SPA driver supports the following registers:

Register Type	Write	Read
I (Input Data)	•	•
O (Output Data)	•	•
S (Setting values)	•	•
V (Variables)	•	•
M (Memory Data)	•	•
F (slave identification)	•	•
L (last events)	•	•
B (last events from backup buffer)	•	•
T (Time)	•	•

**Notes:**

- All slaves do not provide data of all categories since a wide latitude exists in data available from different kinds of slaves.
- The slaves provide data of categories F, T, L, B and A only on channel 0. Data of categories I, O, S, V, and M is available on both channel 0 and other channels.

### **Conformance Testing**

The following hardware/software was used for conformance testing:

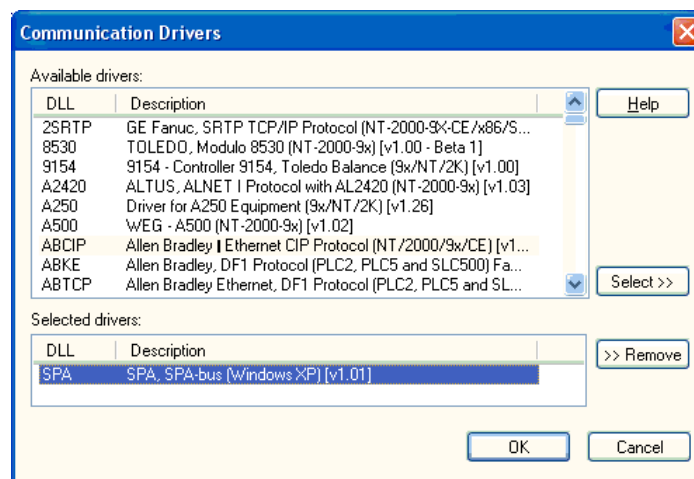
- **Equipment:** Slave devices SPA-bus compatible
- **Driver Configuration:**
  - **PLC Program:** None
  - **Baud Rate:** 9600
  - **Data Bits:** 7
  - **Stop Bits:** 1
  - **Parity:** Even
  - **COM Port:** COM1
- **Cable:** RS232
- **Operating System (development):** Windows XP
- **Operating System (runtime):** Windows XP
- **Studio Version:** 5.1
- **Driver Version:** 1.01

## 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 **SPA** driver from the *Available Drivers* list, and then click the **Select** button.



**Communication Drivers Dialog**

5. When the **SPA** 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.

## Configuring the Driver

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

- Specifying communication parameters.
- Defining communication tags and controls in the *Communication* tables or *Driver* worksheet. There are two types of communication tables: **STANDARD TABLES** and the **MAIN DRIVER SHEET**. This driver version uses only the *Standard 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.

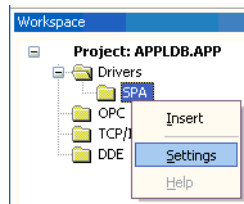
 **Note:**

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

### Setting the Communication Parameters

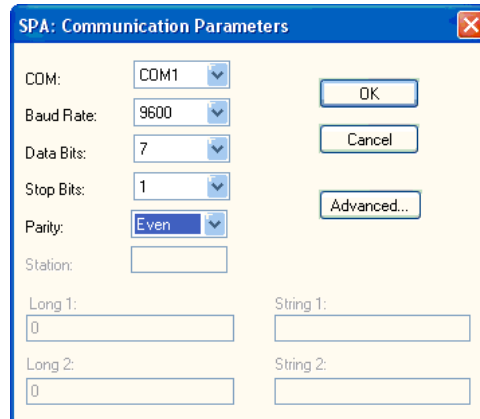
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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



**Select Settings from the Pop-Up Menu**

The *SPA: Communications Parameters* dialog displays:



**Communication Parameters Dialog**

 **Note:**

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

4. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings as necessary.

 **Notes:**

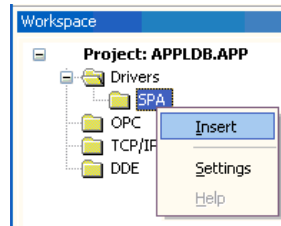
- Do not change any of the other *Advanced* parameters at this time. You can consult the *Studio Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication 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

### 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 *<Driver Name>* 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 *STANDARD DRIVER SHEET* displays (similar to the following figure):

	Tag Name	Address	Div	Add
1	tag_input[1]	1		
2	tag_input[1]	3		
3	tag_input[1]	4		
4				

**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: Specify the device using the following syntax:

**<Slave Number>:<Channel>**

Where:

- **Slave Number** is the device address. It is a decimal number from 1 to 999.
- **Channel** is the device slot number. It is a decimal number from 0 to 999.
- **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 I)
- These variables must comply with the following syntax:  
**<Type>** (for example: I)

Where:

- **Type** is the register type. (I, O, S, M, V, T, F, L or B)

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 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:
- **<AddressOffset>** (For example: 1, 6, 8, 17)

➔ **Attention:**

- AddressOffset result cannot equal zero (0). SPA operands must start in an address that is greater than zero.
- The address is not required for Types T, F, L, and B. If the user types anything in the field address, it will be ignored.

Address Configuration Sample		
Device Address:Channel	Header Field	Address Field
1:4	I	7
3:2	O	10
5:0	F	-
21:0	T	-

**Device Configuration**

Because there are multiple devices that use the SPA protocol, we cannot define a standard device configuration.

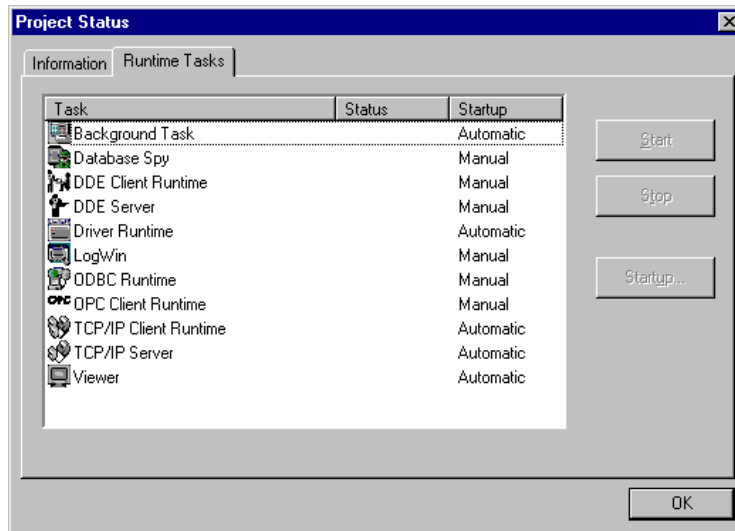
## Executing the Driver

After adding the SPA 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:



*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 SPA 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.
06	Invalid Address	Specified an invalid address in the <i>Driver</i> worksheet. The data address received in this query was not an allowable address for the slave.	Type a valid address.
07	Invalid Header	Specified header contains an invalid value that is not allowed for the slave.	Specify a valid tag value in the <b>Header</b> field.
10	Error in checksum or parity	Protocol Error.	Contact your Studio technical support representative.
11	Slave Busy	Protocol Error.	Contact your Studio technical support representative.
12	Overflow of slave input buffer	Protocol Error.	Contact your Studio technical support representative.
13	Message from Master too complicated for the slave	Protocol Error.	Contact your Studio technical support representative.
14	Reserved for use in higher levels of the communication	Protocol Error.	Contact your Studio technical support representative.
15	Syntax error	Protocol Error.	Contact your Studio technical support representative.
16	Slave does not contain all data requested in the message	Protocol Error.	Contact your Studio technical support representative.
17	Addressed data is impossible to write or read	Protocol Error.	Contact your Studio technical support representative.
18	Data in write message not validated	Protocol Error.	Contact your Studio technical support representative.
19	Undefined negative acknowledgment	Protocol Error.	Contact your Studio technical support representative.

⇒ **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	1.01	Lidiane A. Moreira	Aug/20/2003	First version.