SPO52 Communication Driver

Driver for serial Communication using
SPO52 Protocol

Contents

Introduction

This document will help you to select, configure and execute the SPO52 driver, and it is organized as follows:

- **Introduction**: This section, which provides an overview of the document.
- **General Information**: Identifies all of the hardware and software components required to implement communication between the Studio system and the target device.
- **Selecting the Driver**: Explains how to select the SPO52 driver in the Studio system.
- **Configuring the Device**: Describes how the target device must be configured to receive communication from the SPO52 driver.
- **Configuring the Driver**: Explains how to configure the SPO52 driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver**: Explains how to execute the SPO52 driver during application runtime.
- **Troubleshooting**: Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **Sample Application**: Explains how to use a sample application to test the SPO52 driver configuration
- **Revision History**: Provides a log of all changes made to the driver and this documentation.

General Information

Driver Characteristics

The SPO52 driver package consists of the following files, which are automatically installed in the **\DRV** subdirectory of Studio:

- **SPO52.INI:** Internal driver file. *You must not modify this file*.
- **SPO52.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file*.
- **SPO52.PDF:** This document, which provides detailed information about the SPO52 driver.
- **SPO52.DLL:** Compiled driver.

> **Note:**
> You must use Adobe Acrobat® Reader™ to view the **SPO52.PDF** document. You can install Acrobat Reader from the Studio installation CD, or you can download it from Adobe's Web site.

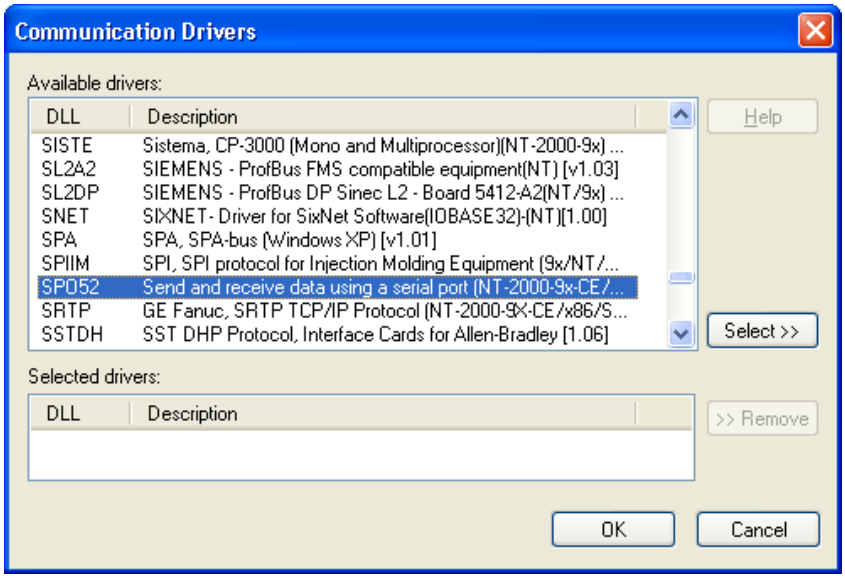
You can use the SPO52 driver on the following operating systems:

- Windows XP/7/8
- Windows CE

Selecting the Driver

When you install Studio, all of the communication drivers are automatically installed in the **\DRV** subdirectory but they remain dormant until manually selected for specific applications. To select the SPO52 driver for your Studio application:

1. From the main menu bar, select **Insert** → **Driver** to open the *Communication Drivers* dialog.
2. Select the **SPO52** driver from the *Available Drivers* list, and then click the **Select** button.



Communication Drivers Dialog

3. When the **SPO52** driver is displayed in the **Selected Drivers** list, click the **OK** button to close the dialog. The driver is added to the *Drivers* folder, in the *Comm* tab of the Workspace.

> **Note:**
> It is not necessary to install any other software on your computer to enable communication between Studio and your target device. However, this communication can only be used by the Studio application; it cannot be used to download control logic to the device. To download control logic to a Mitsubishi device, you must also install the Mitsubishi programming software. For more information, please consult the documentation provided by the device manufacturer.

> **Attention:**
> For safety reasons, you must take special precautions when installing any physical hardware. Please consult the manufacturer's documentation for specific instructions.

Configuring the Driver

Once you have selected the SPO52 driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver's communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

Configuring the Communication Settings

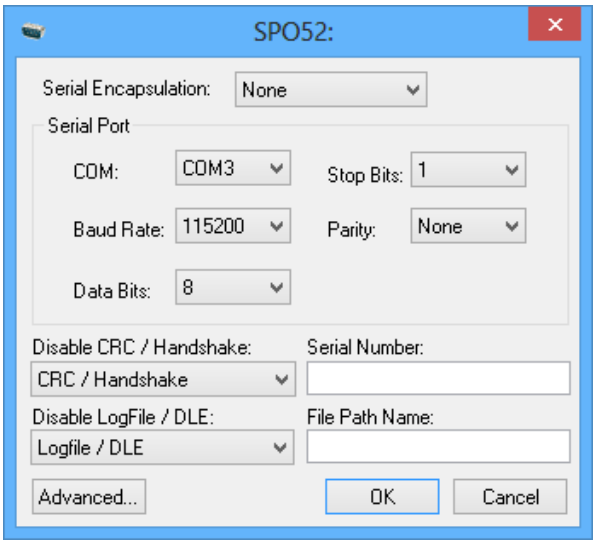
The communication settings are described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

For the purposes of this document, only SPO52 driver-specific settings and procedures will be discussed here. To configure the communication settings for the SPO52 driver:

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The SPO52 driver is listed here as a subfolder.

2. Right-click on the *SPO52* subfolder and then select the **Settings** option from the pop-up menu:

   The *SPO52: Communication Settings* dialog is displayed:



SPO52: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. To ensure error-free communication, the driver settings must *exactly match* the corresponding settings on the device. Please consult the manufacturer's documentation for instructions how to configure the device and for complete descriptions of the settings.

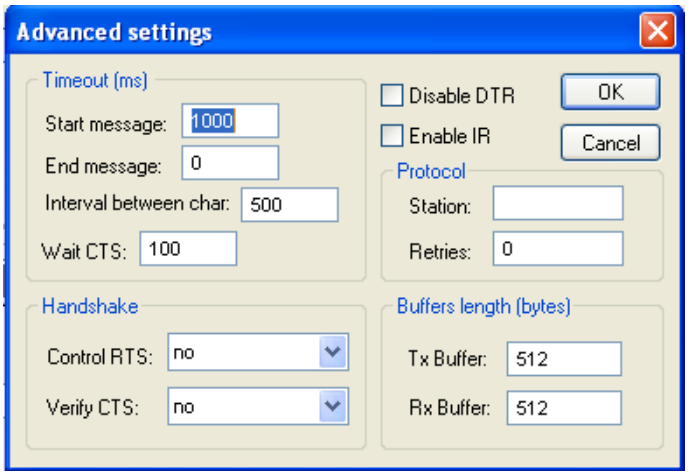
   Depending on your circumstances, you may need to configure the driver *before* you have configured your target device. If this is the case, then take note of the driver settings and have them ready when you later configure the device.

   > ➲ **Attention:**
   > For safety reasons, you **must** take special precautions when connecting and configuring new equipment. Please consult the manufacturer's documentation for specific instructions.

   The communication settings and their possible values are described in the following table:

Parameters	Default Values	Valid Values	Description
Serial Number			▪ The Device Serial number to be added in the Fault historical register.
File Path Name			▪ The first file to be generated by the Fault Header type.
Disable CRC / Handshake	`CRC / Handshake`	`CRC / Handshake` or `CRC / No Handshake` or `No CRC/ No Handshake`	▪ Disable CRC check and/or Handshake. When Handshake is enabled, the driver writes 0x06 as ACK after receiving a valid FAULT package and 0x15 as NACK after receiving an invalid FAULT package. After three consecutive invalid FAULT packages, the driver adds a new line on the error file with zeros.
Disable LogFile / DLE	`Logfile / DLE`	`Logfile / DLE` or `Logfile / No DLE` or `No Logfile / DLE` or `No Logfile / No DLE`	▪ Disable the saving of LogFile and/or the treating of 0x10 as DLE.

4. If you are using a Data Communication Equipment (DCE) converter (e.g., 232/485) between your PC and your target device, then you must also adjust the **Control RTS** (Request to Send) setting to account for the converter. In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:



Advanced Settings Dialog

You can consult the Studio *Technical Reference Manual* later for more information about configuring these settings.

5. IClick **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Settings* dialog.

> **Note:**
> During tests with INCA-Board, it has been noticed the FAULT package received requires a larger buffer size. This can be set on the *Rx Buffer* field under *Buffer length (bytes)* zone at the Driver Advanced Settings shown above. The recommended length, based on these tests, in order to avoid buffer errors is 2048 bytes or greater for the *Rx Buffer*.

Configuring the Driver Worksheets

A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with registers on the target device. Each worksheet is triggered by specific application behavior, so that the tags / registers defined on that worksheet are scanned only when necessary – that is, only when the application is doing something that requires reading from or writing to those specific tags / registers. Doing this optimizes communication and improves system performance.

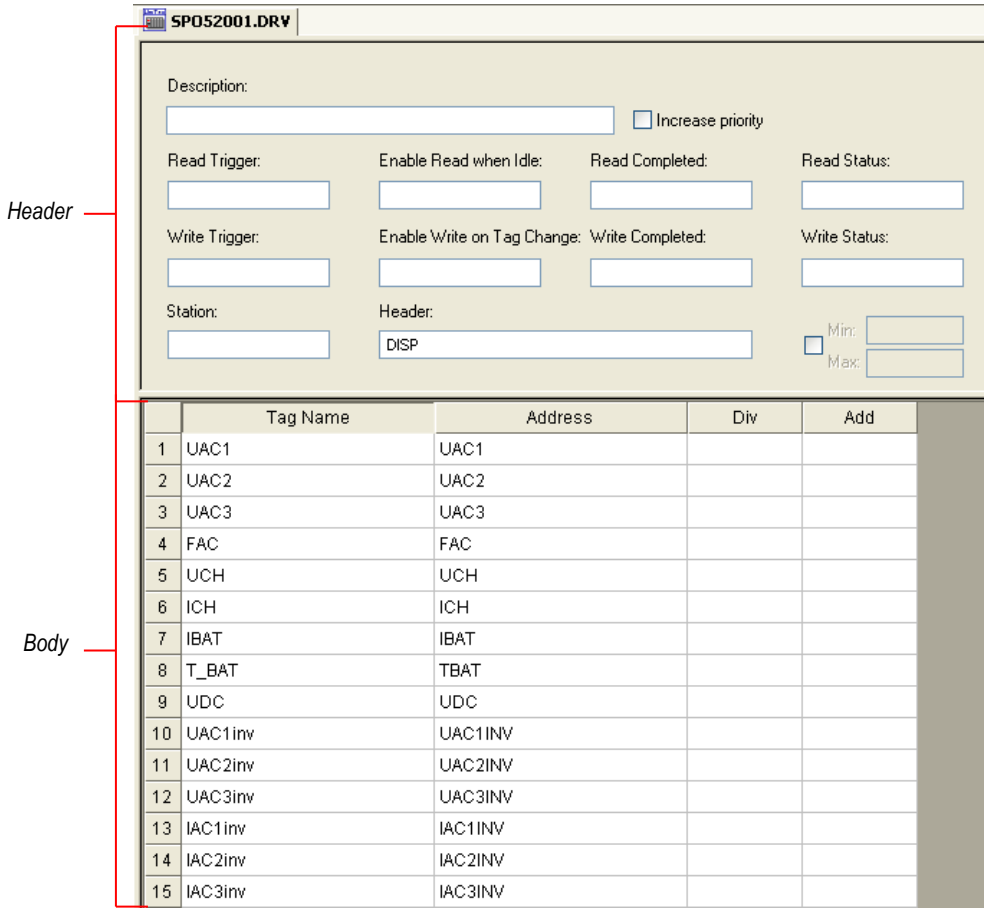
The configuration of these worksheets is described in detail in the "Communication" chapter of the Studio *Technical Reference Manual*, and the same general procedures are used for all drivers. Please review those procedures before continuing.

> **Note:**
> We recommend configuring device registers in sequential blocks in order to maximize performance.

To insert a new driver worksheet:

1. In the *Comm* tab, open the *Drivers* folder and locate the *SPO52* subfolder.
2. Right-click on the *SPO52* subfolder, and then select **Insert** from the pop-up menu:

   A new SPO52 driver worksheet is inserted into the *SPO52* subfolder, and the worksheet is opened for configuration:



SPO52 Driver Worksheet

> **Note:**
> Worksheets are numbered in order of creation, so the first worksheet is **`SPO52001.drv`**.

Most of the fields on this worksheet are standard for all drivers; see the "Communication" chapter of the *Technical Reference Manual* for more information on configuring these fields. However, the **Station**, **Header**, and **Address** fields use syntax that is specific to the SPO52 driver.

3. Configure the **Header** fields as follows:

   - **Header** field: Specify the kind of operation. The **Header** field uses the following syntax:

     <Type>

     Example — **DISP**

     Where:

     <Type> is the operation type.

     You can also specify an indirect tag (e.g. **`{header}`**), but the tag that is referenced must follow the same syntax and contain a valid value.

> **Attention:**
> You cannot leave the **Header** fields blank; you must specify some value.

The following table lists all of the operation types that are valid for the SPO52 driver:

Register Type	Comments
DISP	Receive Display Information (Affichage)
REQ	Send a Requisition to the Device (Requête Utilisateur)
FAULT	Receive Fault Information (Défauts)

4. For each table row (i.e., each tag/register association), configure the **Address** field using the following syntax…

   For DISP header:

   Station: Not used.

   Address: See table below.

	For DISP Header
Address	**Description**
UAC1	RMS network tension AC phase 1
UAC2	RMS network tension AC phase 2
UAC3	RMS network tension AC phase 3
FAC	Network Frequency
UCH	Charger output tension
ICH	Charger current
IBAT	Battery current
T_BAT	Battery temperature
UDC	Continue tension DC

UAC1inv	Charger RMS output tension AC phase 1
UAC2inv	Charger RMS output tension AC phase 2
UAC3inv	Charger RMS output tension AC phase 3
IAC1inv	Charger RMS current AC phase 1
IAC2inv	Charger RMS current AC phase 2
IAC3inv	Charger RMS current AC phase 3
FINV	Charger output frequency
DCH1	Charger fault word 1
DCH2	Charger fault word 2
WCH1	Charger warning 1
WCH2	Charger warning 2
SCH1	Charger status word 1
SCH2	Charger status word 2
DINV1	Charger fault word 1
DINV2	Charger fault word 2
WINV1	Charger warning word 1
WINV2	Charger warning word 2
SINV1	Charger status word 1
SINV2	Charger status word 2

For REQ header:

Station: Not used.

Address: See table below.

	For REQ Header
Symbol	**Description**
IDV	Request Frame Number

For FAULT header:

Station: Path for saving a csv file with the following format:

IDD, n, <Timestamp>, UAC1-1, UAC2-1, UAC3-1, Uout1-1, Uout2-1, Uout3-1, Iout1-1, Iout2-1, Iout3-1

> ➲ **Attention:**
> The Write Trigger in the Standard Driver Sheet is used to start the Error File writing. Before the write trigger be toggled the Error File will be written in the Settings File Path Name configured.

Address: See table below.

	For FAULT Header
Address	**Description**
UAC1_1	Network tension AC phase 1 (Fault information)
UAC2_1	Network tension AC phase 2 (Fault information)

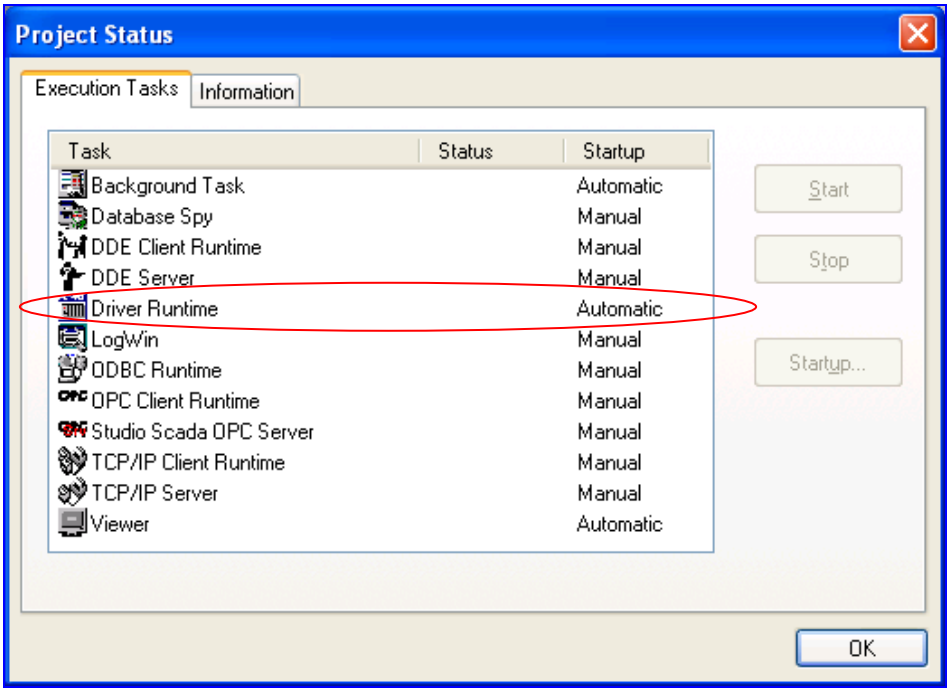
UAC3_1	Network tension AC phase 3 (Fault information)
UOUT1_1	Output tension AC phase 1 (Fault information)
UOUT2_1	Output tension AC phase 2 (Fault information)
UOUT3_1	Output tension AC phase 3 (Fault information)
IOUT1_1	Output current AC phase 1 (Fault information)
IOUT2_1	Output current AC phase 2 (Fault information)
IOUT3_1	Output current AC phase 3 (Fault information)
IDD	Index of the Fault Frame (Fault Information Frame (Défauts))

Executing the Driver

By default, Studio will automatically execute your selected communication driver(s) during application runtime. However, you may verify your application's runtime execution settings by checking the *Project Status* dialog.

To verify that the the communication driver(s) will execute correctly:

1. From the main menu bar, select **Project** → **Status**. The *Project Status* dialog displays:



Project Status Dialog

2. Verify that the *Driver Runtime* task is set to **Automatic**.
   - If the setting is correct, then proceed to step 3 below.
   - If the **Driver Runtime** task is set to **Manual**, then select the task and click the **Startup** button to toggle the task's *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the SPO52 driver fails to communicate with the target device, then the database tag(s) that you configured for the **Read Status** or **Write Status** fields of the Main Driver Sheet will receive an error code.

> ⇨ **Tip:**
>
> You can monitor communication status by establishing an event log in Studio's *Output* window (*LogWin* module). To establish a log for **Field Read Commands**, **Field Write Commands** and **Serial Communication,** right-click in the *Output* window and select the desired options from the pop-up menu.
>
> You can also use the *LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. The log is saved on the unit in the **`celog.txt`** file, which can be downloaded later.

If you are unable to establish communication between Studio and the target device, then try instead to establish communication using the device's own programming software. Quite often, communication is interrupted by a hardware or cable problem or by a device configuration error. If you can successfully communicate using the programming software, then recheck the driver's communication settings in Studio.

To test communication between Studio and the device, 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**: Displays in the Studio *Output* window when the driver is running.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Plinio Santana	26-Nov-2007	First driver version.
B	1.01	Eric Vigiani	26-Aug-2008	Revision after changes in the protocol
B	1.02	Lourenço Teodoro	16-Dec-2008	Updated driver version, no changes in the contents.
C	1.3	Nelson Inagaki	07-Mar-2012	Updated driver version, no changes in the contents.
D	1.4	Paulo Balbino	11-Apr-2013	Released driver
E	1.5	Ricardo Marroni	20-Feb-2014	Implemented CRC and Handshake Modified the data type FINV to be Unsigned Fixed values being read on the IBAT and TBAT data types Improved stability on Windows CE devices