

CFW Communication Driver

Driver for Serial Communication
with CFW Devices

Contents

INTRODUCTION2

GENERAL INFORMATION3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING4

INSTALLING THE DRIVER5

CONFIGURING THE DRIVER.....6

 SETTING THE COMMUNICATION PARAMETERS6

 CONFIGURING THE DRIVER WORKSHEETS.....8

EXECUTING THE DRIVER 12

TROUBLESHOOTING..... 13

SAMPLE APPLICATION..... 14

REVISION HISTORY 15

Introduction

The CFW driver enables communication between the Studio system and Weg CFW devices communicating over Serial, according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the CFW driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the CFW driver.
- **Installing the Driver:** Explains how to install the CFW driver.
- **Configuring the Driver:** Explains how to configure the CFW 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 CFW 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 CFW driver and the CFW 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:** Weg
- **Compatible Equipment:**
 - CFW-05
 - SCA-04
- **Programming Software:** None
- **Device Runtime 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:** Serial RS232
- **Logic Protocol:** Proprietary
- **Specific PC Board:** None

Driver Characteristics

The CFW driver is composed of the following files:

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

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

- Windows NT/2K/XP

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

The CFW driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float	DWord
V (Variables)	1 Byte	•	•	•	•	–	–
P3 (Parameters)	1 Byte	•	•	•	•	–	–

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration:**
 - Baud Rate: 9600
 - Data Bits: 8
 - Stop Bits: 1
 - Parity: Even
 - COM Port: COM1
- **Cable:** According to specifications described in the “Link Characteristics” section.

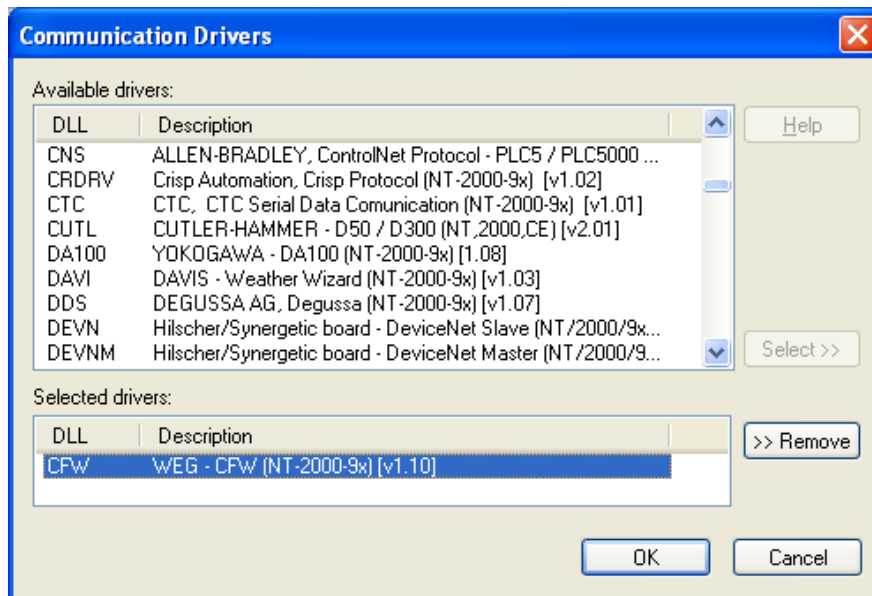
Driver Version	Studio Version	Operating System	Equipment
1.10	6.0	Win2K	SCA-04

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 **CFW** driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.



Communication Drivers Dialog Box

5. When the **CFW** 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 CFW driver, you must configure the driver. Configuring the CFW 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.

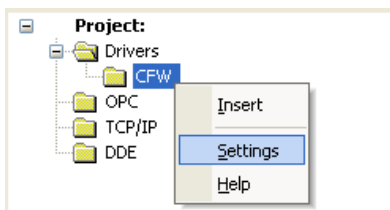
Note:

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

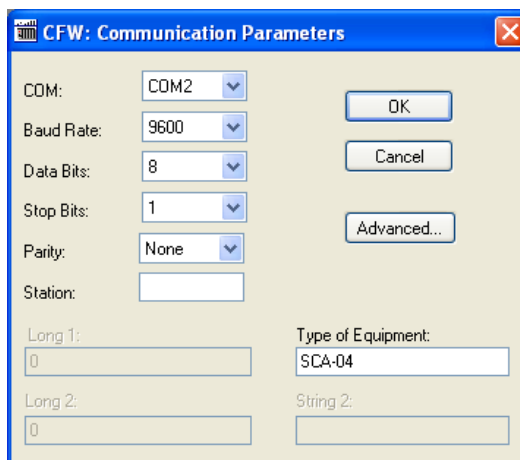
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

1. From the Studio development environment, select the *Comm* tab located below the *Workspace*.
2. Click on the *Drivers* folder in the *Workspace* to expand the folder.
3. Right-click on the CFW 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 *CFW: Communications Parameters* dialog displays (as follows).




Communication Parameters Dialog

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

Parameters	Default Values	Valid Values	Description
Station	0	0-31	Devices identification numbers. 0 for Point-to-Point communication 31 for Broadcast
Type of Equipment	-	CFW-05 SCA-04 Other types	Type of equipment

 **Note:**
The device must be configured with *exactly the same* parameters that you configured in the *CFW 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.

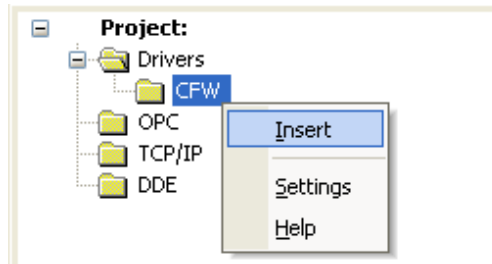
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER 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.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *CFW* subfolder.
3. 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).

The screenshot shows a configuration window for a driver. At the top, there is a 'Description' field containing 'Driver test' and an 'Increase priority' checkbox. Below this are sections for 'Read Trigger' and 'Write Trigger', each with fields for 'Enable Read when Idle', 'Read Completed', 'Read Status', 'Write Trigger', 'Enable Write on Tag Change', 'Write Completed', and 'Write Status'. The 'Station' field is set to '1' and the 'Header' field is set to 'V02'. There are also 'Min:' and 'Max:' fields with checkboxes. At the bottom is a table with 8 rows and 5 columns: Tag Name, Address, Div, and Add.

	Tag Name	Address	Div	Add
1	Tag[1]	1.0		
2	Tag[2]	1.1		
3	Tag[3]	1.2		
4	Tag[4]	1.3		
5	Tag[5]	1.4		
6	Tag[6]	1.5		
7	Tag[7]	1.6		
8	Tag[8]	1.7		

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:

<PLC ID>

Where:

- PLC ID is the PLC identification number (from 0 to 31).

- **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.

These variables must comply with the following syntax:

<Register Type>:<Reference> (For example: V02 or P308)

Where:

- **Register Type** is the register type. (**V** or **P3**)
- **Reference** is the reference of register type (must have two digits).

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 CFW driver.

- **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:

<Address> . [Bit] (For example: *10, 20, 40, 10.5*)

Where:

- **Address** is the Byte number of register type configured in the **Header** field.
- **Bit** (*optional parameter*) is the bit number to be read from or written to the device.

Basic Variables (V)		
Register Reference	Address	Comments
V00 (Indication of the converter model – read)	0	Indication of the converter model
V02 (indication of the state of the converter - read)	0	Error State
	1.0	if 0, set for inactive ramp if 1, set for active ramp
	1.1	if 0, set for inactive general if 1, set for active general
	1.2	if 0, counterclockwise if 1, clockwise
	1.3	if 0, inactive JOG if 1, active JOG
	1.4	if 0, local if 1, remote
	1.5	if 0, without sub if 1, with sub
	1.6	if 0, manual if 1, automatic
V03 (Selection of logical command - write)	0.0	if 0, deactivate for ramp if 1, activate for ramp
	0.1	if 0, deactivate for general if 1, activate for general

Basic Variables (V)		
Register Reference	Address	Comments
	0.2	if 0, counterclockwise if 1, clockwise
	0.3	if 0, inactive JOG if 1, active JOG
	0.4	if 0, local if 1, remote
	0.5	Not used
	0.6	Not used
	0.7	If in error state, change from 0 to 1 in this bit causes "reset" in the converter.
	1.0	If 1, block for ramp
	1.1	If 1, block for general
	1.2	If 1, rotation direction
	1.3	If 1, JOG
	1.4	If 1, local/remote
	1.5	Not used
	1.6	Not used
	1.7	Converter "reset"
V04 (Reference of frequency for Serial-read/write)	0	Reference of frequency for Serial
V05 (Commands activated for Serial - read)	0.0	If 1, local reference via Serial
	0.1	If 1, local selection of direction of rotation via Serial
	0.2	If 1, local Turn on/Turn off via Serial
	0.3	If 1, local JOG via Serial
	0.4	If 1, remote reference via Serial
	0.5	If 1, remote selection of direction of rotation via Serial
	0.6	If 1, remote Turn on/Turn off via Serial
	0.7	If 1, remote JOG via Serial
	1	If 1, selection of local/remote via Serial

Parameters of Serial Communications (P3)		
Register Reference	Address	Comments
P307 (Baud rate)	0	0 – Serial deactivated 1 - 300 2 - 600 3 - 1200 4 - 2400 5 - 4800 6 - 9600
P308 (Converter ID)	0	1 - 30

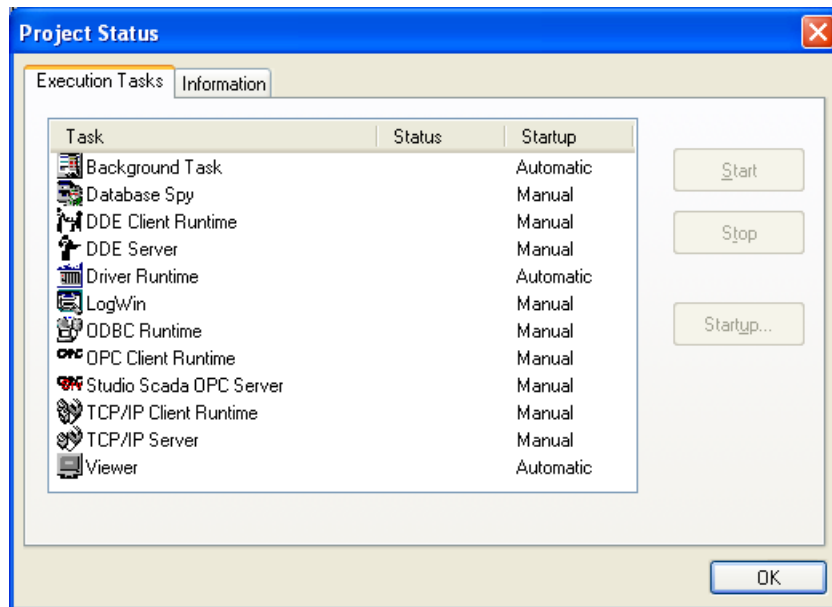
Executing the Driver

After adding the CFW 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 CFW 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.
2	NACK	Message too big or invalid message	Check the address to be sure it exists on the device. Check the equipment state. Check the communication parameters for the device and the Studio software.
3	Invalid Message	Receive a corrupted message. Message without '=' character.	Check cable wiring.

→ **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/<Driver Name>` 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
	1.04	Roberto V. Junior	Dec/1998	Fix problem with Timeout.
	1.06	Roberto V. Junior	Feb/1999	Include Retries in SendMessage
	1.09	Roberto V. Junior	Sep/2003	Included new equipment SCA.
A	1.10	Fábio H.Y. Komura	May/2004	Fix problem when a corrupted message is received.