# ALFA Communication Driver

Driver for Serial
Communication with Alfa Devices

## Contents

# Introduction

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

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

- **Introduction**: Provides an overview of the ALFA driver documentation.
- **General Information**: Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the ALFA driver.
- **Installing the Driver**: Explains how to install the ALFA driver.
- **Configuring the Driver**: Explains how to configure the ALFA 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 ALFA 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 XP/CE environment.
  If you are unfamiliar with Windows XP/CE, 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 ALFA driver and the Alfa 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**: ALFA
- **Compatible Equipment**:
  - Any device that is fully compatible with the ALFA devices family 3000

## *Link Characteristics*

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

- **Physical Protocol**: Serial
- **Logic Protocol**: Proprietary
- **Device Runtime Software**: None
- **Specific PC Board**: None

## *Driver Characteristics*

The ALFA driver is composed of the following files:

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

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

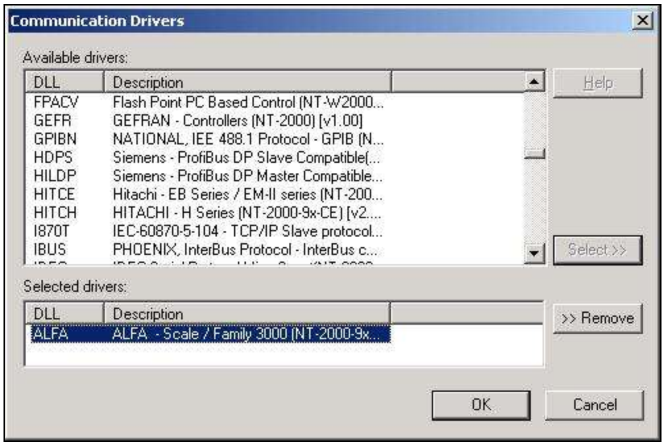
- Windows XP
- Windows CE

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



Communication Drivers

Available drivers:

| DLL | Description |
|---|---|
| FPACV | Flash Point PC Based Control (NT-W2000... |
| GEFR | GEFRAN - Controllers (NT-2000) [v1.00] |
| GPIBN | NATIONAL, IEE 488.1 Protocol - GPIB (N... |
| HDPS | Siemens - ProfiBus DP Slave Compatible(... |
| HILDP | Siemens - ProfiBus DP Master Compatible... |
| HITCE | Hitachi - EB Series / EM-II series (NT-200... |
| HITCH | HITACHI - H Series (NT-2000-9x-CE) [v2.... |
| I870T | IEC-60870-5-104 - TCP/IP Slave protocol... |
| IBUS | PHOENIX, InterBus Protocol - InterBus c... |

Selected drivers:

| DLL | Description |
|---|---|
| ALFA | ALFA - Scale / Family 3000 (NT-2000-9x... |

***Communication Drivers Dialog Box***

5. When the **ALFA** 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. However, to download the custom program to your device, you must install the ALFA programmer. Consult your ALFA programmer software documentation for installation instructions.

> **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 ALFA driver, you must configure the driver. Configuring the ALFA driver is done in two parts:

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

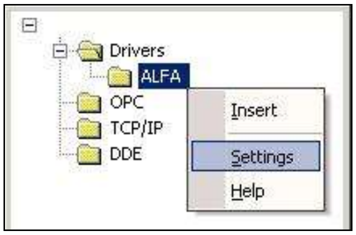
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**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

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

## ***Setting the Communication Parameters***

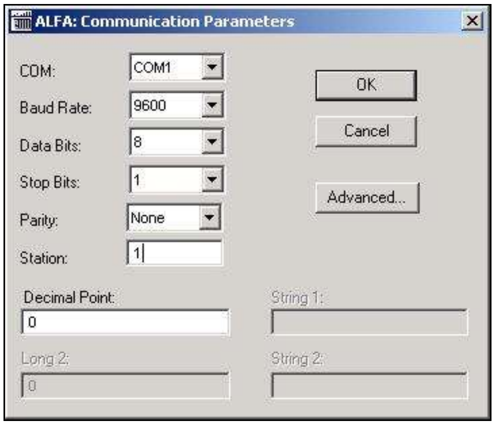
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 ALFA 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 *ALFA: Communications Parameters* dialog displays (as follows).



COM: COM1
Baud Rate: 9600
Data Bits: 8
Stop Bits: 1
Parity: None
Station: 1
Decimal Point: 0
String 1:
Long 2: 0
String 2:

***Communication Parameters Dialog***

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

| Parameters | Default Values | Valid Values | Description |
|---|---|---|---|
| **Station** | None | - | Computer Station |
| **Decimal Point** | 0 | 0.. 5 | Decimal Point |

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

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

> ✍ **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.

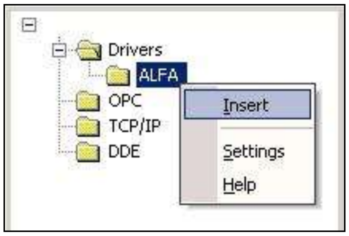
## ***Configuring the Driver Worksheets***

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*:

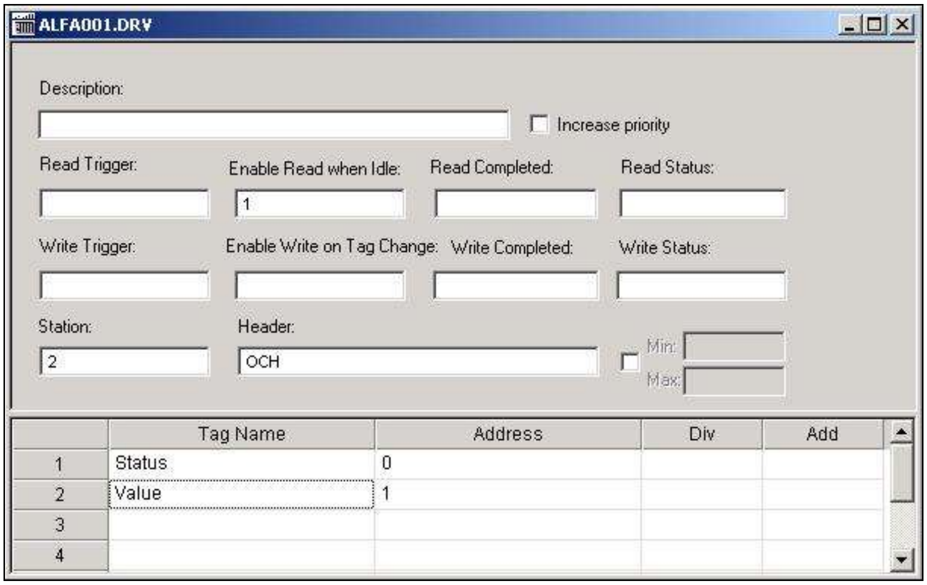
1. From the Studio development environment, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *ALFA* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



***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).



ALFA001.DRV

Description:

☐ Increase priority

Read Trigger:

Enable Read when Idle: 1

Read Completed:

Read Status:

Write Trigger:

Enable Write on Tag Change:

Write Completed:

Write Status:

Station: 2

Header: OCH

☐ Min:

Max:

| | Tag Name | Address | Div | Add |
|---|---|---|---|---|
| 1 | Status | 0 | | |
| 2 | Value | 1 | | |
| 3 | | | | |
| 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: Device number.
- **Header** field: Use the information in the following table to define the type of command.

  These variables must comply with the following syntax:

  ***<Command>*** *(For example: OCH)*

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 ALFA driver.

| Header Field Information | |
|---|---|
| **Read Command** | **Comments** |
| 04H | Verification of the SetPoints. |
| 06H | Verification of the SetPoint configuration |
| 0BH | Verification of the weight band for operation of the analogical output |
| 0CH | Verification of the SetPoints |
| 51H | Verification of the accumulated value |
| 54H | Verification of the calibration parameters |
| 08H | Solicitation of Weights and general Status |
| 18H | Solicitation of Weight and storage Status |
| 19H | Clean storage command |
| **Write Command** | **Comments** |
| 02H | Unique initialization of the slaves |
| 03H | Verification of the SetPoints |
| 05H | Verification of the SetPoint configuration |
| 07H | Programming of the slave address |
| 0AH | Verification of the weight band for operation of the analogical output |
| 50H | Transition of the calibration parameters |
| 09H | Activation of key over net |

- **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:

| Header Field | Address Field | Comments |
|---|---|---|
| 04H | 0 | SetPoints Number |
| | 1 | SetPoints 1 |
| | 2 | SetPoints N |
| 06H | 0 | Hysteresis (%) |
| | 1 | Relay 0 – Open<br>1 – Close |
| | 2 | SetPoints Number |

| | | |
|---|---|---|
| | 3 | Lock 1 |
| | 4 | Lock N |
| 0BH | 0 | Reference: 0 – Gross weights<br>1 – Net weights |
| | 1 | 4mA or 0mA weights |
| | 2 | 20mA weights |
| OCH | 0 | Status |
| | 1 | Accumulated value |
| | 2 | Decimal Point |
| | 3 | Degree |
| | 4 | Calibration weights |
| | 5 | Maximum capacity |
| | 6 | Auto zero: 0 – Zero manual and automatic disable<br>1 – Zero automatic<br>2 – Zero manual and automatic enable |
| | 7 | Digital Filter: 0 – Fast<br>1 – Normal<br>2 – Slow |
| | 8 | Tare: 0 – Not successive<br>1 – Not memory successive<br>2 – Successive<br>3 – Memory successive |
| 54H | 0 | Status: 0 – No error<br>1 – Empty >= calibration weights<br>2 – Insufficient span<br>3 – Calibration weights > Maximum capacity |
| 08H | 0 | Position of the Decimal Point |
| | 1 | Negative weights in the display |
| | 2 | Detection of movement of weights |
| | 3 | Saturation alarm |
| | 4 | Overweight alarm |
| | 5 | Transition by cut level 1 |
| | 6 | Transition by cut level 2 |
| | 7 | Transition by cut level 3 |
| | 8 | Transition by cut level 0 |
| | 9 | Current position of the display |
| | 10 | Local parameters changed over display/key |
| | 11 | Display weights |
| | 12 | Tare |
| 18H | 0 | Storage weights |
| | 1 | Tare |
| 19H | None | |
| 02H | None | |
| 03H | 0 | Status: 0 – Not save in memory not volatile<br>1 – Save in memory not volatile |
| | 1 | SetPoint number |
| | 2 | SetPoint 1 |
| | 3 | SetPoint 2 |

| | N | SetPoint N |
|---|---|---|
| 05H | 0 | Hysteresis (00.99%) |
| | 1 | Relay : 0 – Open<br>1 – Close |
| | 2 | Status: 0 – Not save in memory not volatile<br>1 – Save in memory not volatile |
| | 3 | SetPoint Number |
| | 4 | Lock 1 (0 – Without lock, 1 – with lock) |
| | 5 | Lock 2 (0 – Without lock, 1 – with lock) |
| | 6 | Lock N (0 – Without lock, 1 – with lock) |
| 07H | 0 | New address |
| 0AH | 0 | Reference: 0 – Gross weights<br>1 – Net weights |
| | 1 | 4mA or 0mA weights |
| | 2 | 20mA weights |
| 50H | 0 | Decimal Point (0, 1, 2, 3 or 4) |
| | 1 | Degree (1, 2 or 5) |
| | 2 | Calibration weights |
| | 3 | Maximum capacity |
| | 4 | Auto zero: 0 – Zero manual and automatic disable<br>1 – Zero automatic<br>2 – Zero manual<br>3 – Zero manual and automatic enable |
| | 5 | Digital Filter: 0 (Fast)….8 (Slow) |
| | 6 | Tare: 0 – Not successive<br>1 – Not memory successive<br>2 – Successive<br>3 – Memory successive |
| 09H | 0 | Zero (0 or 1) |
| | 1 | Tare (0 or 1) |
| | 2 | Not tare (0 or 1) |
| | 3 | Unlock (0 or 1) |

## ***Device Configuration***

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

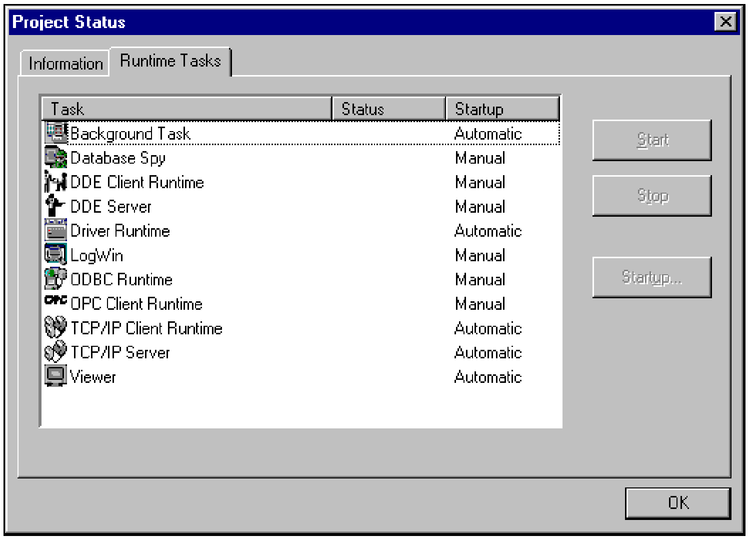
# Executing the Driver

After adding the ALFA 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.



| Task | Status | Startup |
|---|---|---|
| Background Task | | Automatic |
| Database Spy | | Manual |
| DDE Client Runtime | | Manual |
| DDE Server | | Manual |
| Driver Runtime | | Automatic |
| LogWin | | Manual |
| ODBC Runtime | | Manual |
| OPC Client Runtime | | Manual |
| TCP/IP Client Runtime | | Automatic |
| TCP/IP Server | | Automatic |
| Viewer | | Automatic |

***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 ALFA 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 |
| 1 | Error of Protocol | Receive an invalid message | Check the computer and device numbers. |
| 2 | Invalid Command | The command configured in the Header field is invalid. | Check the command configured in the Header field, and make sure it can read or write. |
| 3 | Invalid Response | Receive an invalid message | Check the computer and device numbers. |
| 4 | Invalid Block Size | The number the line configured in the Address field is wrong. | Check the number configured in the Address field. |
| -15 | Timeout Start Message | ▪ Disconnected cables<br>▪ PLC is turned off, in stop mode, or in error mode<br>▪ Wrong station number<br>▪ Wrong RTS/CTS control settings | ▪ Check cable wiring.<br>▪ Check the PLC state – it must be RUN.<br>▪ Check the station number.<br>▪ 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<br>▪ Wrong station number<br>▪ Wrong parity<br>▪ Wrong RTS/CTS configuration settings | ▪ Check cable wiring.<br>▪ Check the PLC state – it must be RUN.<br>▪ Check the station number.<br>▪ 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

This driver does not have a sample application

# Revision History

| Doc. Revision | Driver Version | Author | Date | Description of changes |
|---|---|---|---|---|
| A | 1.10 | Eric Vigiani | Sep/28/2004 | Modified to avoid CPU usage in 100%. |
| B | 1.11 | André Körbes | Sep/23/2010 | Generated driver for Windows CE |