

**RTP Communication Driver**

Driver for Ethernet Communication (TCP/IP)  
with RTP 2000 / RTP 2300 Series Devices

**Contents**

**INTRODUCTION .....2**

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

    DEVICE SPECIFICATIONS.....3

    NETWORK SPECIFICATIONS.....3

    DRIVER CHARACTERISTICS .....3

**SELECTING THE DRIVER .....4**

**CONFIGURING THE DRIVER .....5**

    CONFIGURING THE COMMUNICATION SETTINGS .....5

    CONFIGURING THE DRIVER WORKSHEETS .....6

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

**TROUBLESHOOTING ..... 12**

**REVISION HISTORY..... 13**

## Introduction

The RTP driver enables Ethernet communication between the Studio system and RTP 2000 or RTP 2300 Series devices, according to the specifications discussed in this document.

This document will help you to select, configure and execute the RTP 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 RTP driver in the Studio system.
- **Configuring the Device:** Describes how the target device must be configured to receive communication from the RTP driver.
- **Configuring the Driver:** Explains how to configure the RTP driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver:** Explains how to execute the RTP 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 RTP driver configuration
- **Revision History:** Provides a log of all changes made to the driver and this documentation.

### Notes:

- This document assumes that you have read the “Development Environment” chapter in Studio’s *Technical Reference Manual*.
- This document also assumes that you are familiar with the Microsoft Windows NT/2000/XP environment. If you are not familiar with Windows, then we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

## General Information

This chapter identifies all of the hardware and software components required to implement serial communication between the RTP driver in Studio and a RTP device.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics

### Device Specifications

To establish communication, your target device must meet the following specifications:

- **Manufacturer:** RTP Corporation
- **Compatible Equipment:** 2000 and 2300 Series Data Control and Acquisition systems
- **Programmer Software:** NetSuite
- **Device Runtime Software:** None

For a description of the device(s) used to test driver conformance, see “Conformance Testing” on the next page.

### Network Specifications

To establish communication, your device network must meet the following specifications:

- **Device Communication Port:** Ethernet Port
- **Physical Protocol:** TCP/IP
- **Logic Protocol:** Proprietary (ASCII)
- **Specific PC Board:** Any TCP/IP adapter (Ethernet board)

### Driver Characteristics

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

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

 **Note:**

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

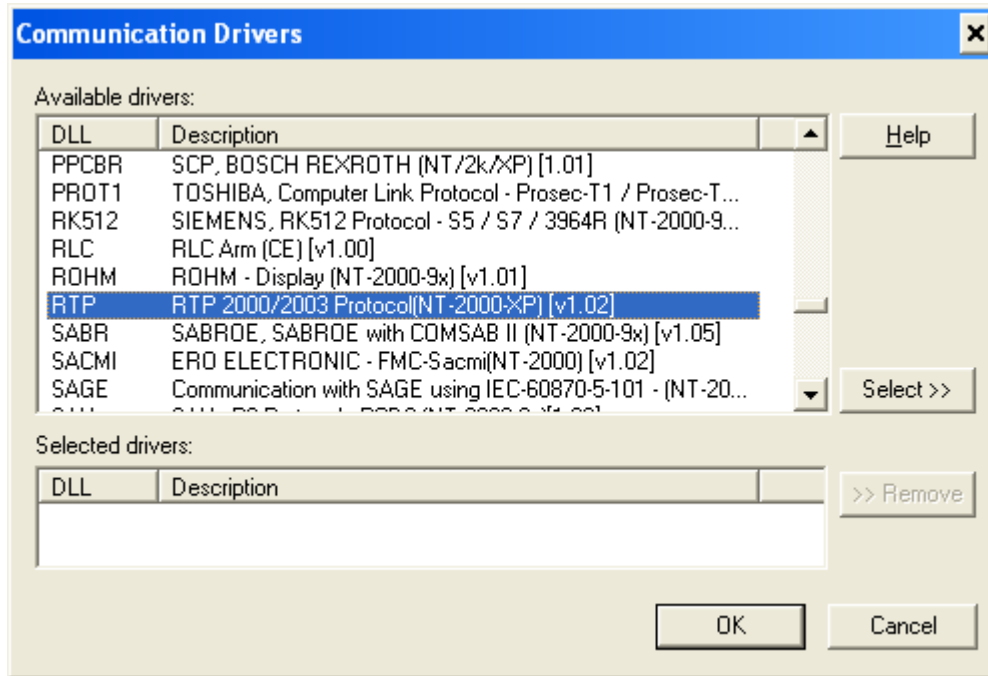
You can use the RTP driver on the following operating systems:

- Windows 2000/XP/Vista

## 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 RTP driver for your Studio application:

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



*Communication Drivers Dialog*

3. When the **RTP** 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 RTP device, you must also install the RTP programming software (e.g. NetSuite). 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.

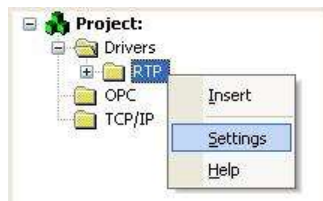
## Configuring the Driver

Once you have selected the RTP 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

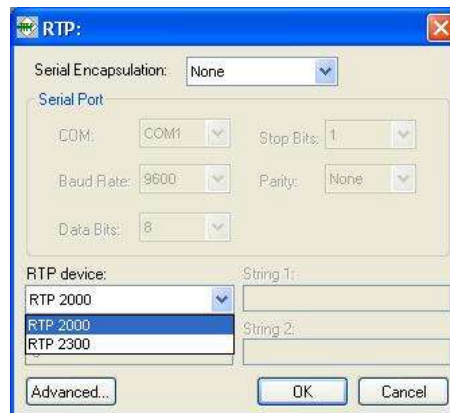
There is two options of devices RTP and you need to choose one of these in the driver's settings.

1. In the *Workspace* pane, select the *Comm* tab and then expand the *Drivers* folder. The RTP driver is listed here as a subfolder.
2. Right-click on the *RTP* subfolder and then select the **Settings** option from the pop-up menu:



Select Settings from the Pop-Up Menu

The *MOTCP: Communication Settings* dialog is displayed:



MOTCP: Communication Settings Dialog

3. In the *Communication Settings* dialog, configure the driver settings to enable communication with your target device. Select RTP 2000 or RTP 2300 (based in your device) to the driver work properly.

## Configuring the Driver Worksheets

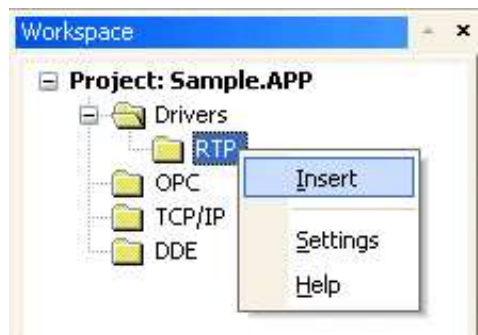
A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with registers on the target device. Each worksheet is triggered by specific application behavior, so that the tags / registers defined on that worksheet are scanned only when necessary – that 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 *RTP* subfolder.
2. Right-click on the *RTP* subfolder, and then select **Insert** from the pop-up menu:



*Inserting a New Worksheet*

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

The screenshot shows a configuration window for 'RTP001.DRV'. It is divided into a 'Header' section and a 'Body' section. The 'Header' section contains several input fields: 'Description' (FP Variable Read/Write), 'Read Trigger', 'Write Trigger', 'Station' (89.7.1.69), and 'Header' (FV:100). There are also checkboxes for 'Increase priority', 'Enable Read when Idle', 'Enable Write on Tag Change', 'Min', and 'Max'. The 'Body' section is a table with the following data:

	Tag Name	Address	Div	Add
1	tag_fv	0		
*				
*				

**XRCM Driver Worksheet**

**Note:**  
 Worksheets are numbered in order of creation, so the first worksheet is **RTP001.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 RTP driver.

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

- **Station** field: Specify the IP Address of the device, using the following syntax:

**<IP Address>**

Example — 89.7.1.69

...where **<IP Address>** is the device’s IP address on the TCP/IP network.

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

**Attention:**  
 You must use a non-zero value in the **Station** field, and you cannot leave the field blank.

- **Header field:** Specify the address of the first register of a block of registers on the target device. The addresses declared in the *Body* of the worksheet are simply offsets of this **Header** address. When Read/Write operations are executed for the entire worksheet (see **Read Trigger** and **Write Trigger** above), it scans the entire block of registers from the first address to the last.

The **Header** field uses the following syntax:

**<Type>: <AddressReference>**

Example — **FV:100**

...where:

- **<Type>** is the register type (Please refer to table below)
- **<AddressReference>** is the initial address (reference) of the configured type.

After you edit the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is invalid, then Studio automatically inserts a default value of **VCF:0**.

You can also specify an indirect tag (e.g. {**header**}), but the tag value that is referenced must be valid by following the same syntax described above.

4. Support for the **Read Trigger**, **Write Trigger**, **Enable Read when Idle** and **Enable Write on Tag Change** operations varies according to the register type specified in the **Header** field. The following table shows which operations are supported by which register types. Configure the operations as described in the “Communication” chapter of the *Technical Reference Manual*.

Type	Description	RTP2000	RTP3000	Read Trigger	Write Trigger	Enable Read	Enable Write
VCF	Virtual Coil Force	YES	YES	NA	Allowed	NA	Allowed
VC	Virtual Coil Read/Write	YES	YES	Allowed	Allowed	Allowed	Allowed
DIF	Digital Input Force	YES	NO	NA	Allowed	NA	Allowed
DI	Digital Input Read/Write	YES	NO	Allowed	Allowed	Allowed	Allowed
DOF	Digital Output Force	YES	NO	NA	Allowed	NA	Allowed
DO	Digital Output Read/Write	YES	NO	Allowed	Allowed	Allowed	Allowed
AIF	Analog Input Force	YES	NO	NA	Allowed	NA	Allowed
AI	Analog Input Read/Write	YES	NO	Allowed	Allowed	Allowed	Allowed
AOF	Analog Output Force	YES	NO	NA	Allowed	NA	Allowed
AO	Analog Output Read/Write	YES	NO	Allowed	Allowed	Allowed	Allowed
VAF	Variable Force	YES	YES	NA	Allowed	NA	Allowed
VA	Variable Read/Write	YES	YES	Allowed	Allowed	Allowed	Allowed
FVF	FP Variable Input Force	YES	YES	NA	Allowed	NA	Allowed
FV	FP Variable Read/Write	YES	YES	Allowed	Allowed	Allowed	Allowed
CLK	Clock Read/Write	YES	YES	Allowed	Allowed	Allowed	NA
SCC	Get Symbol Count	YES	YES	Allowed	NA	Allowed	NA
SCI	Get Symbol by Index	YES	YES	Allowed	NA	Allowed	NA



Type	Description	RTP2000	RTP3000	Read Trigger	Write Trigger	Enable Read	Enable Write
SCN	Get Symbol by Name	YES	YES	Allowed *	NA	Allowed *	NA
STU	Get Status	YES	YES	Allowed	NA	Allowed	NA
MSG	Get Message	YES	YES	Allowed	NA	Allowed	NA

\* Reads only one item per driver sheet

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

**<AddressOffset>**

Examples — 0, 10, 50

...where **<AddressOffset>** is a parameter that is added to the **<AddressReference>** parameter of the **Header**, to compose the specific address of the register in the block. The sum of the two parameters cannot equal zero (0).

Header	Valid Addresses	Description	Data Type
VCF	0 thru 65535	Sets the Force Flag on Virtual Coil Registers.	Boolean
VC	0 thru 65535	Reads/Writes to Virtual Coil Registers.	Boolean
DIF	0 thru 65535	Sets the Force Flag on Digital Input Registers.	Boolean
DI	0 thru 65535	Reads/Writes to Digital Input Registers.	Boolean
DOF	0 thru 65535	Sets the Force Flag on Digital Output Registers.	Boolean
DO	0 thru 65535	Reads/Writes to Float Digital Output Registers.	Boolean
AIF	0 thru 65535	Sets the Force Flag on Analog Input Registers.	Boolean
AI	0 thru 65535	Reads/Writes to Analog Input Registers.	Integer
AOF	0 thru 65535	Sets the Force Flag on Analog Output Registers.	Boolean
AO	0 thru 65535	Reads/Writes to Analog Output Registers.	Integer
VAF	0 thru 65535	Sets the Force Flag on Variable Registers.	Boolean
VA	0 thru 65535	Reads/Writes to Float Point Variable Registers.	Integer
FVF	0 thru 65535	Sets the Force Flag on Float Point Variable Registers.	Boolean
FV	0 thru 65535	Reads/Writes to Float Point Variable Registers.	Real
CLK	HUNDREDTH, SECOND, MINUTE, HOUR, DOW ( <i>day of the week</i> ), DAY, MONTH, YEAR	Reads/Writes to the RTP Driver's Internal Clock. <b>Note:</b> When writing to the clock, any parameter not included in the driver sheet will be set to its minimum value.	Integer
SCC	0	Reads the Symbol Count from the RTP Program.	Integer
SCI	TYPE, ADDRESS, ELEMENTS, NAME	Reads the Properties of a Symbol Given Its Index.	Integer, Integer, String
SCN	<Name of the Symbol>	Reads the Address of a Symbol Given Its Name.	Real
STU	SYSINFO, ENGSTATUS, BUFFERSIZE, SBOFFSET, APOFFSET	Reads Status Information from the RTP Device.	Integer
MSG	ENGSTATUS, MESSAGE	Reads Engine Status and Error Messages from the RTP Device.	Integer, String

For examples of how device registers are specified using **Header** and **Address**, see the following table:

Device Register	Header	Address
VCF:10	<b>VCF : 0</b>	<b>10</b>
VCF:10	<b>VCF : 5</b>	<b>5</b>
CLK:SECOND	<b>CLK</b>	<b>SECOND</b>
SCC	<b>SCC</b>	<b>0</b>

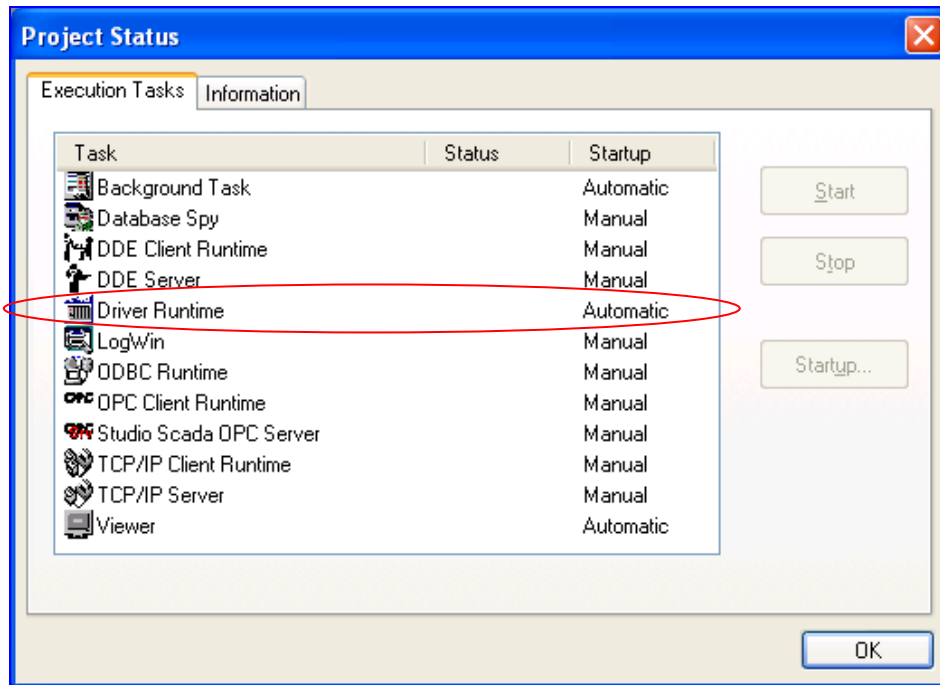
For more information about device registers and addressing, please consult the manufacturer's documentation.

## 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 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 RTP 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. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes
0	OK	Communication completed successfully.
3	Invalid Station	Wrong IP Address configured in the Driver Worksheet <i>Station</i> field
23	Invalid Answer	PLC reply was not recognized.
27	Write Violation	Register Type does not support the write function.
28	Invalid Message Length	Length of the PLC's response was invalid.
29	Read Violation	Register Type does not support the read function.
30	No Symbol Table	No Symbol Table found in the device.
31	No Message	No Message found for the corresponding address.
32	Wrong CRC	Wrong Cyclic Redundancy Check (CRC) causes the message to be considered invalid. This usually is caused when the message arrived from the PLC misses part of the information
-15	Time Out	<ul style="list-style-type: none"> <li>▪ Disconnected cables</li> <li>▪ PLC is turned off, in stop mode, or in error mode</li> <li>▪ Wrong station number</li> </ul>

⇒ **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 **Protocol Analyzer**, right-click in the *Output* window and select the desired options from the pop-up menu.

You can also use the *Remote LogWin* module (**Tools** → **Remote LogWin**) to establish an event log on a remote unit.

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 (e.g., NetSuite). 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	Bryan Morgan	24 Sep 2002	First driver version
B	1.00	Bryan Morgan	04 Apr 2003	Corrected reading float and bit registers error
C	1.01	Plínio M. Santana	16 Feb 2007	Made modification to sign the CRC correctly
D	1.01	Plínio M. Santana	27 Feb 2007	Document corrections.
E	1.01	Michael D. Hayden	30 Mar 2007	Edited for language and usability.
F	1.02	Eric Vigiani	04 Sep 2008	Added support to RTP2300 device.