

SATCH Communication Driver

Driver for Serial Communication
with Satchwell IAC Devices

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS.....3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING4

INSTALLING THE DRIVER5

CONFIGURING THE DRIVER6

 SETTING THE COMMUNICATION PARAMETERS6

 CONFIGURING THE DRIVER WORKSHEETS8

EXECUTING THE DRIVER 12

TROUBLESHOOTING 13

SAMPLE APPLICATION 15

REVISION HISTORY..... 16

Introduction

The SATCH driver enables communication between the Studio system and Gammaflux TTC devices using the Gammaflux TTC Auxiliary protocol communicating over Serial, according to the specifications discussed in this document.

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

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

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics
- Conformance Testing

Device Characteristics

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

- **Manufacturer:** Satchwell Control System
- **Compatible Equipment:**
 - IAC200/400 & IAC600 family
- **Programming Software:** None
- **Device Runtime Software:** None

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

Link Characteristics

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

- **Device Communication Port:** Serial
- **Physical Protocol:** Serial RS422/485
- **Logic Protocol:** Proprietary (SatchNet)

Driver Characteristics

The SATCH driver is composed of the following files:

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

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

- Windows NT/2K/XP
- Windows CE

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

The SATCH driver supports the following registers:

Register Type	Length	Write	Read	Bit	Integer	Float
Tables	Varies according to table	•	•	•	•	•

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration:**
 - **Baud Rate:** 1200
 - **Data Bits:** 7
 - **Stop Bits:** 1
 - **Parity:** Even
 - **COM Port:** COM2
- **Cable:** Serial RS232 with RS232/RS485 converter

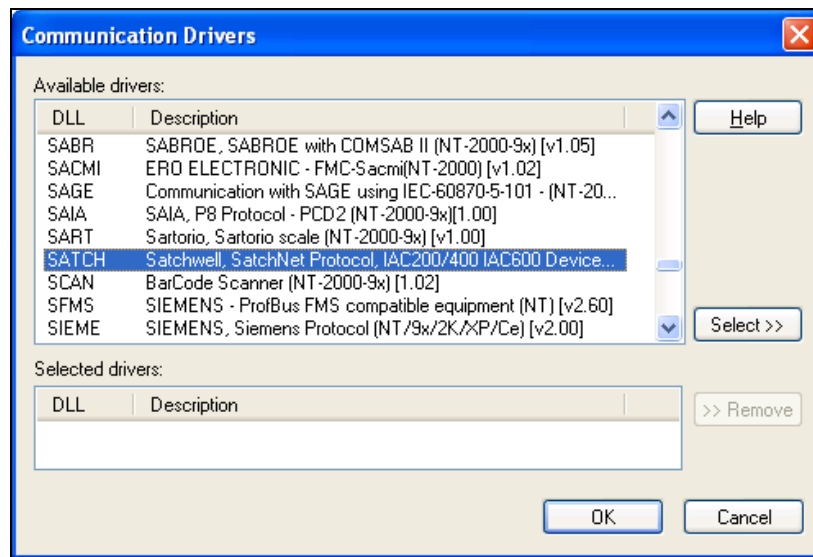
Driver Version	Studio Version	Operating System	Equipment
1.00	6.0	WinXP	Satchwell Control System IAC 600

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

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

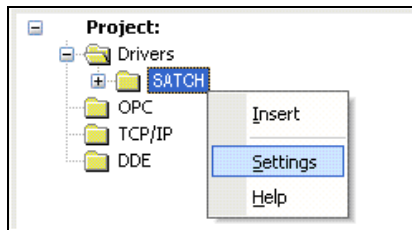
Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header** and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**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

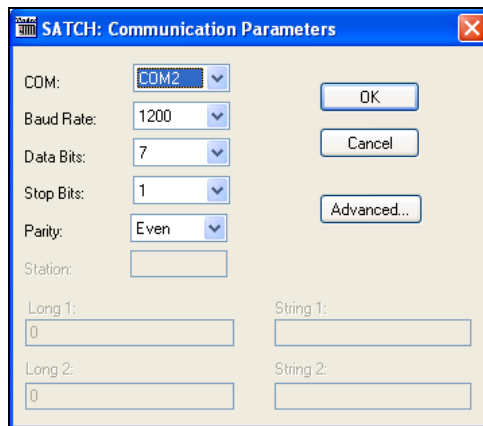
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

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 SATCH 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 *SATCH: Communication Parameters* dialog displays (as follows).



Communication Parameters Dialog

This Driver does not have custom parameters.

4. 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.

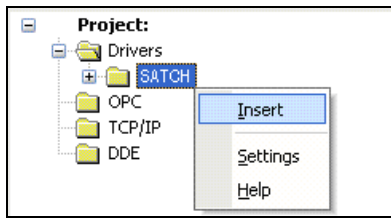
Configuring the Driver Worksheets

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 a *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 *SATCH* 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).

Description:				
<input type="text"/>				<input type="checkbox"/> Increase priority
Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:	
<input type="text" value="rd"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:	
<input type="text" value="wr"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Station:	Header:			<input type="checkbox"/> Min: <input type="text"/>
<input type="text" value="1"/>	<input type="text" value="T74"/>			Max: <input type="text"/>
	Tag Name	Address	Div	Add
1	TagA	0		
2	TagB	B1		
3	TagC	W2		
4	Tagd	F3		

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 Users Guide and 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.

- **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. The default value is **T0**.

These variables must comply with the following syntax:

T<Table Number> (for example, **T10**)

Where:

- **<Table Number>** is the table number (any number).

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 (**T0**) 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; otherwise, you will get an invalid Header error.

The following table lists some of the Tables and address ranges that are valid for the SATCH driver.

Header Field Information (Examples)			
Table	Sample Syntax	Table Offset (bytes)	Comments
69	T69	<ul style="list-style-type: none"> • 0: Seconds (0-59) • 1: Minutes (0-59) • 2: Hours (0-23) • 3: Day of Week (1-7) • 4: Day (1-31) • 5: Week Number (1-53) • 6: Month (1-12) • 7: Year (0-99) • 8: Update flag (0-1) 	Time (Data format is single byte.)
70	T70	0: Problem flag (0-1)	SatNet Problem Flag (Data format is 1 byte.)
71	T71	<ul style="list-style-type: none"> • 0: Group address 1 (32 - 127) • 1: Group address 2 (32 - 127) • 2: Group address 3 (32 - 127) • 3: Group address 4 (32 - 127) • 4: Group address 5 (32 - 127) • 5: Group address 6 (32 - 127) • 6: Group address 7 (32 - 127) • 7: Group address 8 (32 - 127) • 8: Group address 9 (32 - 127) • 9: Group address 10 (32 - 127) 	Group Addresses (Data format is 1 byte.)
72	T72	<ul style="list-style-type: none"> • 0: Tag 1 (0 - 65535) • 2: Value 1 (-10000 - 10000) • 4: Tag 2 (0 - 65535) • 6: Value 2 (-10000 - 10000) • 8: Tag 3 (0 - 65535) • 10: Value 3 (-10000 - 10000) • 12: Tag 4 (0 - 65535) • 14: Value 4 (-10000 - 10000) 	Global Message Data (Data format is 2 byte.)
74	T74	<ul style="list-style-type: none"> • 0: Address (0 - 65535) • 2: Baud Rate (0 - 65535) • 4: True Baud Rate (0 - 65535) • 6: Protocol (0 - 65535) 	Serial Link Parameters (Data format is 2 byte.)
81	T81	Varies according to the equipment	State Input Nodes (SI) (Data format is single byte.)
82	T82	Varies according to the equipment	State Output Nodes (SO) (Data format is single byte.)
83	T83	Varies according to the equipment	Value Input Nodes (VI) (Data format is float point 4 bytes.)
84	T84	Varies according to the equipment	Value Output Nodes (VO) (Data format is float point 4 bytes.)
90	T90	Varies according to the equipment	Node Table (Data format is 2 bytes.)

To access other Tables, see the device documentation.

- **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:

[Data type]<AddressOffset>.[Bit] (for example, **W10, B20, 40, 10.5**)

Where:

- **[Data type]** (*optional parameter*) defines the data type of the operand. If not defined, Studio uses the default Data Types. The following Data Types are available:
 - B (signed Byte),
 - UB (unsigned Byte) (default)
 - W (signed Word – 2 bytes)
 - UW (unsigned Word – 2 bytes)
 - DW (signed double Word – 4 bytes)
 - UDW (unsigned double Word – 4 bytes)
 - F (Float Point – 4 bytes)

- **<AddressOffset>** is a parameter (in bytes) added to the **AddressReference** parameter (configured in the **Header** field) to compose the group address configured in the **Header** field.
- **[Bit]** (*optional parameter*) is the bit number to be read from the device.

Attention:

- The Floating point and Double words value are stored in four consecutive registers (byte), where the address value corresponds to the first register position. You must not configure a non-existent address or a conflict will occur.
- This driver supports Bit reading only; it cannot execute Bit writing.

Address Configuration Sample		
Device Address	Header Field	Address Field
Table 69 Offset 0 (byte)	T69	0
Table 69 Offset 1 (byte)	T69	1
Table 72 Offset 2 (word)	T72	W2
Table 82 Offset12 (float)	T82	F12
Table 82 Offset12 (byte)	T82	B12

Attention:

You must not configure a range of addresses greater than the maximum block size (data buffer length) supported by each PLC within the same worksheet. The maximum data buffer length for this driver is 128 bytes per *STANDARD DRIVER SHEET*.

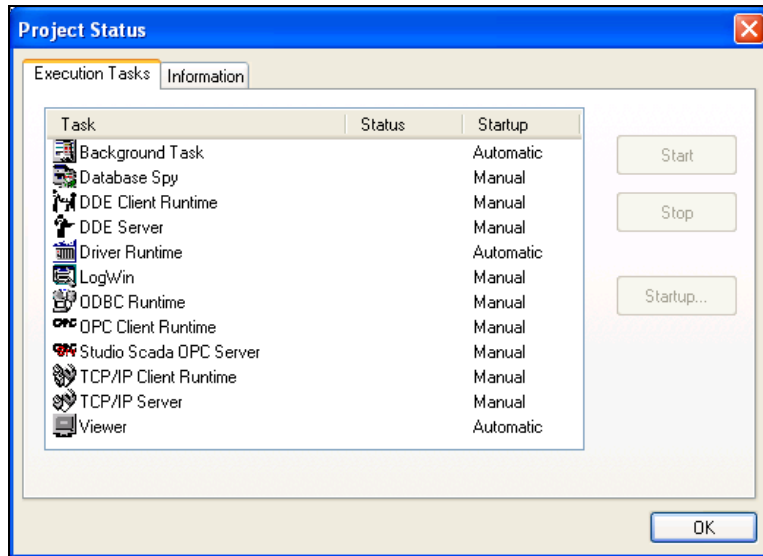
Executing the Driver

After adding the SATCH 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 SATCH 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
1	Protocol error	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
2	Invalid read command in received message	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
3	Wrong Station in received message	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
4	Receive NACK from PLC	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
5	Wrong Table in received message	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
6	Wrong Length in received message	<ul style="list-style-type: none"> ▪ Wrong station number ▪ Wrong RTS/CTS control settings ▪ PLC in stop mode or in error mode 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected cables ▪ PLC is turned off, in stop mode or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> ▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See <i>Studio Users Guide and Technical Reference Manual</i> for information about valid

Error Code	Description	Possible Causes	Procedure to Solve
			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

You will find a sample application for drivers in the `/COMMUNICATION EXAMPLES/<SATCH>` directory. We strongly recommend that you check for a sample application for this driver and use it to test the driver before configuring your 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
A	1.00	Fábio H.Y.Komura	March 18, 2005	Initial version