Communication Driver Hitch

Driver for serial communication with
Hitachi H-Series devices using RS-232

Index

1 Introduction

The Hitch driver enables communication between Studio system and some of the Hitachi devices using their H-Series protocol by RS 232, in accordance with the characteristics covered in this document.

This document contains 8 parts, as follow:

- Introduction: Provides an overview of the driver documentation.
- General characteristics: Provides information necessary to identify all the required components (hardware and software) necessary to implement the communication and global characteristics about the communication.
- Installation: Explains the procedures that must be followed to install the software and hardware required for the communication.
- Driver configuration: Provides the required information to configure the communication driver such as the different permutations for configuration and its default values.
- Execution: Explain the steps to test whether the driver was correctly installed and configured.
- Troubleshooting: Supplies a list of the most common error codes for this protocol and the procedures to fix them.
- Application Sample: Provides a sample application for testing the configuration the driver.
- History of versions: Provides a log of all the modifications done in driver.

☞ Note: This document presumes that the user has read the chapter *Driver Configuration* of the Studio's Technical reference manual.

2 General Characteristics

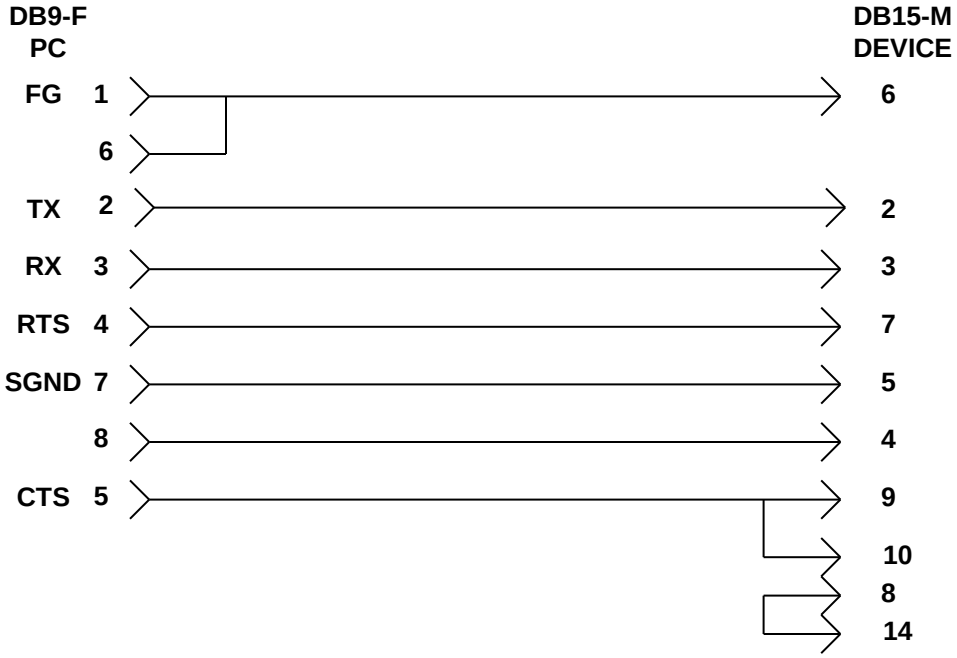
2.1 Device Characteristics

- Manufacturer: Hitachi
- Compatible Equipment
  - H - Series
- Hitachi H-Series PLC programmer software, for example: Actsip-H.

☞ Note: Please refers to section 2.4 to see the Equipment used in the standard conformance tests for this driver.

2.2 Link Characteristics

- Device communication port: RS232 port (COMM) / Peripheral port (CPU)
- Physical protocol: RS232(RS422)
- Logic protocol: Hitachi H-Series Protocol
- Device Runtime software: None
- Specific PC Board: None
- Adapters / Converters: None
- Cable Wiring: Look it on the following picture bellow.



2.3 Driver Characteristics

- Operating System:
  - Windows 9x
  - Windows 2000
  - Windows NT
  - Windows CE

☞ Note: Please refer to section 2.4 to see the Operating System used in the conformance tests for this driver.

The driver is composed of the following files:

- HITCH.INI: Internal file of the driver, it should not be modified by the user.
- HITCH.MSG: This file contains the error messages for each error code. It is an internal file of the driver, the user should not modify it.
- HITCH.PDF: This document provides detailed documentation about the driver.
- HITCH.DLL: This is the compiled library for the driver.

☞ Note: All the files above must to be in the subdirectory /DRV of the Studio's installation directory.

2.4 Information about conformance testing

- **Equipment**: Hitachi H-Series PLC H-200

- **Configuration:**

Baud Rate: 19200
Protocol: Hitachi H-Series Protocol
Data Bits: 7
Stop Bits: 1
Parity: Even
COM port: COM1

- **Cable**: According link specification, section 2.2.
- **Operating System (development)**: Windows NT 4.0 + Service pack 5
- **Operating System (target)**: Windows NT 4.0 + Service Pack 5; Windows CE v2.11 – Processor x86
- **Studio Version**: 3.0
- **Driver version**: 2.01

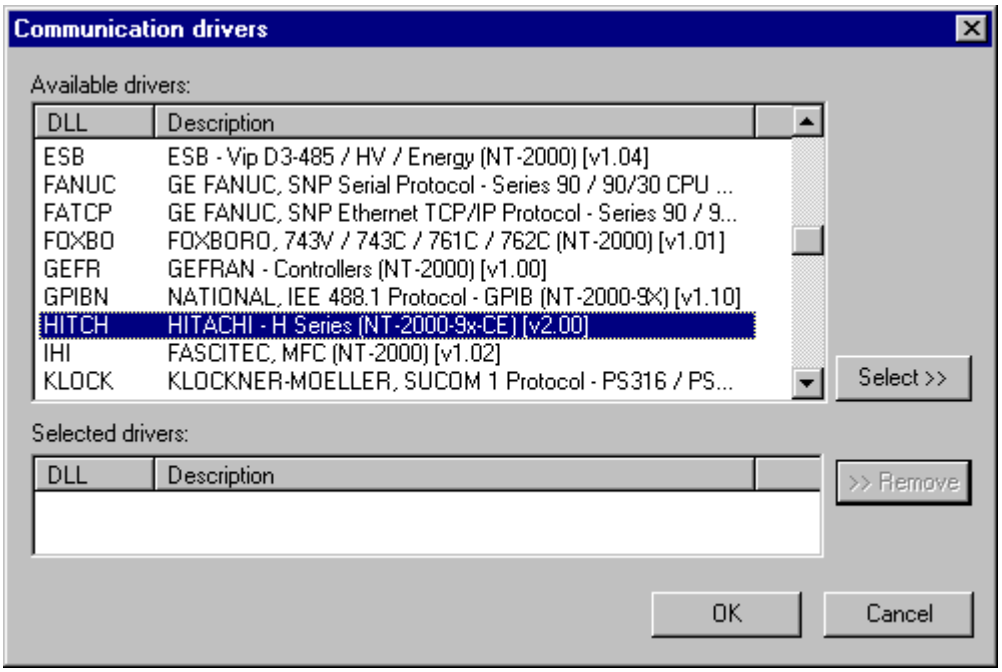
3 Installation

3.1 Installing the Driver

When you install the Studio v3.0 or higher, the communication drivers are already installed. You need now to select the driver at the applications where it will be used.

The steps to select the driver inside an application are:

1. Execute the Studio and select the proper application.
2. Select the menu *Insert + Driver...*
3. In the column **Available Drivers**, select the **Hitch Driver** and push the button **ADD>>>** (the driver Hitch must appear in the column **Selected Drivers**).
4. Press **OK**.



3.2 Other software requirements

It is not necessary to install any other software in the PC to enable the communication between the host and the Device. However, to download the custom program to the device, it is necessary to install one of the Hitachi programmer softwares, for example, Actsip-H. Please see the Actsip-H documentation about the procedure to install it.

☞ Note: Special attentions must be taken when installing the physical hardware. Refer to the hardware manufacturer documentation for specific instructions in this area.

4 Driver Configuration

After the driver is installed and selected in the Studio (see section 3.1), you should proceed to the driver configuration.

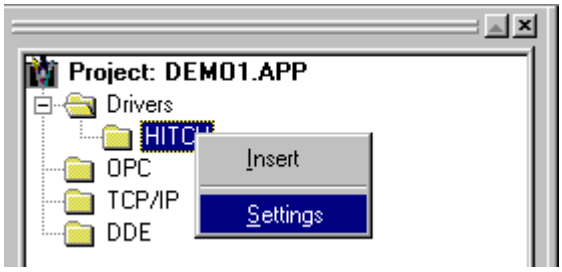
The driver configuration is two parts:

The Settings or Communication parameters, it is only one configuration to the whole driver, them you have the communication tables or Driver Worksheets, where the communication tags are defined.

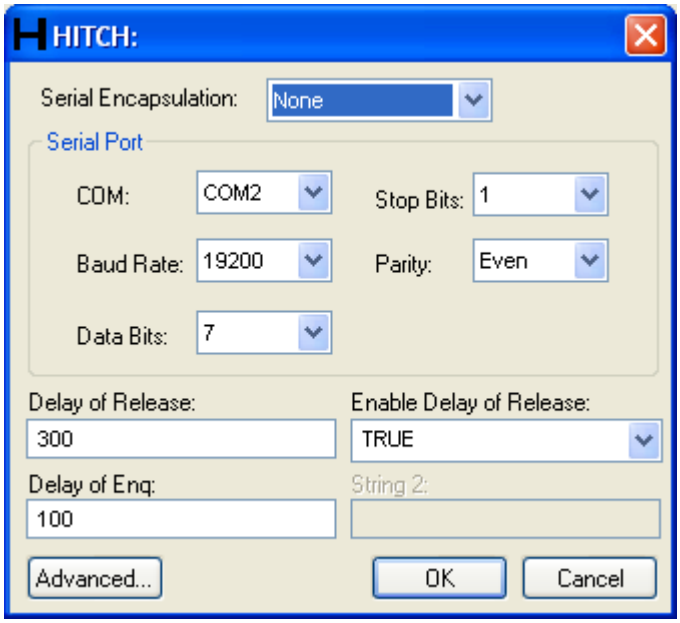
4.1 Settings - Communication Parameters

These parameters are valid for all driver worksheets configured in the system. To open the window for configuring the **Communication parameters**, follow these steps:

1. In the **Workspace** of the Studio environment, select the **Comm** table.
2. Expand the folder **Drivers** and select the subfolder **HITCH**.
3. Right click on the **HITCH** subfolder and select the option **Settings**.



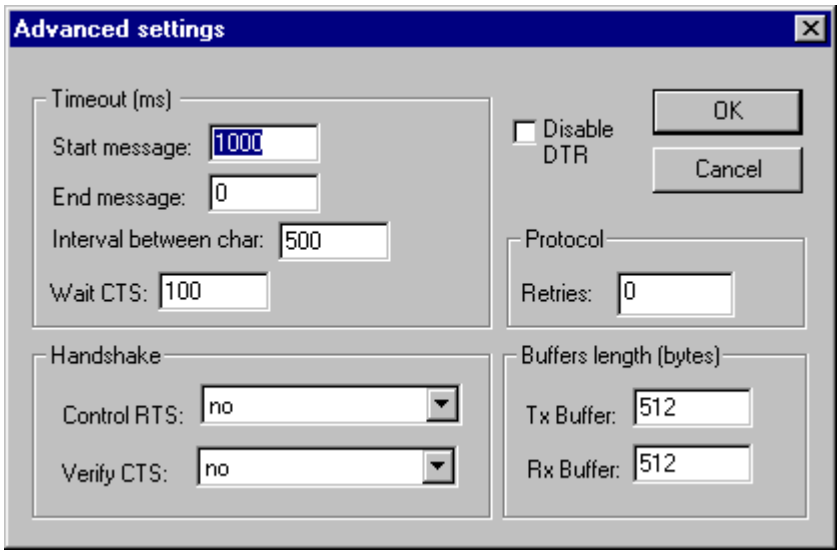
When selecting the Settings, there is the following dialog to configure:



Parameter	Default Value	Valid values	Description
COM	COM2	COM1 to COM8	Serial port of the PC used to communication with the device
Baud Rate	19200	110 to 57600bps	Communication rate of data
Data Bits	7	5 to 8	Number of data bits used in the protocol
Stop Bits	1	1or 2	Number of stop bits used in the protocol
Parity	Even	even, odd, none, space or mark	Parity of the protocol
Station	0	0	Not used for this driver
Delay of Release	300	Numeric value (ms)	Delay in millisecond between Release Message and Data Message sender to device. This parameters can assume a different value according the Host and Device models. Modify this parameters just when you can not launch the communication.
Delay of Enq	100	Numeric value (ms)	Delay in millisecond to send Enq Message to device. This parameters can assume a different value according the Host and Device models. Modify this parameters just when you can not launch the communication.
Enable Delay of Release	TRUE	TRUE or FALSE	Enable/Disable the delay of release.

☞ Note: These Parameters must be exactly the same as the configured on the Hitachi device. The Hitachi H-Series Instruction Manual indicates the following serial communication settings:
Baud Rate: 19200, Data bits: 7, Stop bits: 1, Parity: Even

By clicking on the button **Advanced…** in the window **Communication Parameters,** you open additional communication parameters.



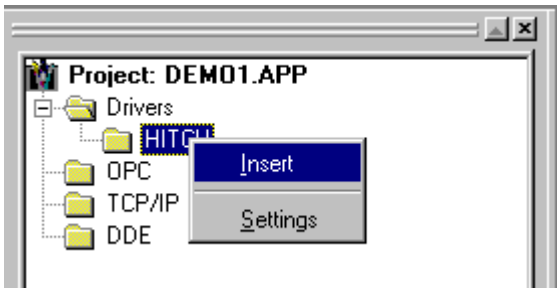
The Advanced setting parameters are explained at the Studio Technical Reference Manual, and you should keep the default values to all fields. Only the field described at the next table should be configured:

Parameter	Default Value	Valid values	Description
Control RTS	No	No, yes or yes + echo	Define if the handshake signal of RTS (Request to Send) is set before communication and if there is an echo in the communication. You must to choose the option "**no**". **Important**: Wrong settings on this field will not let the driver work, having the Timeout waiting start a message error.

4.2 Driver Worksheet

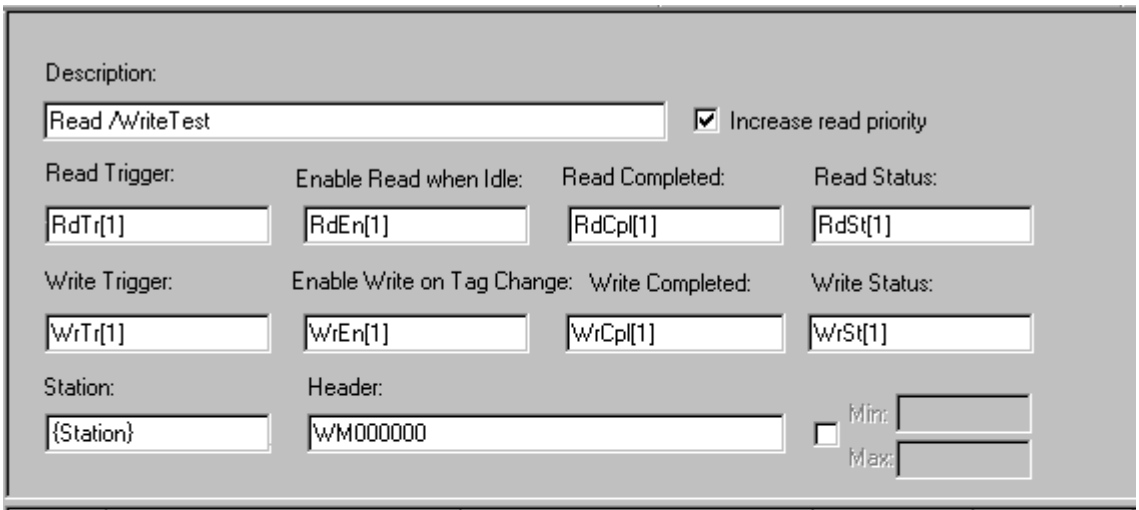
It is possible to configure many driver worksheets, each one will be composed of a Header and Body. To create a new driver worksheet, follow these steps:

1. In the **Workspace** of the Studio environment, select the table **Comm**.
2. Expand the folder **Drivers** and select the subfolder **HITCH**.
3. Right click on the **HITCH** subfolder and select the option **Insert**.



☞ Note: To optimize communication and ensure better performance for the system, it is important to tie the tags in different driver sheets according to the events that must trigger the communication of each group of tags and the periodicity for which each group of tags must be written or read. In addition, it is recommended to configure the addresses of communication in sequential blocks.

When creating a communication table, you have the following window:



	Tag Name	Address	Div	Add
1	RDWR[0]	0		
2	RDWR[1]	1		
3	RDWR[2]	2		
4	RDWR[3]	3		
5	RDWR[4]	4		
6	RDWR[5]	5		
7	RDWR[6]	6		

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Communication Driver documentation about the configuration of the standard fields. This document describes the Station, Header and Address fields, which are specific to each communication driver.

4.3 Station and Header configuration

Parameter	Default Value	Valid values	Description
Station	-	-	Not used for the driver
Header	WM000000	Vide next table	Defines the type of variable to be read or written from or to the device and the reference of the initial address.

The **Header** field defines the type of variables that will be read or written from or to the device. It complies with the syntax: <type of operator data><initial address reference>. After editing the field **Header**, the system will check if it is valid or not. If the syntax were incorrect, the default value (WM000000) or the seamless value will be automatically placed in this field.

You can type Tag between curly brackets into this field, but be sure that the Tag's value is correct, with the right syntax, or you will get the Invalid Header error. The right syntax, both for the field typing and Tag value is described bellow:

- **Type of operator data**: it can be **X** ((Remote/Local)External bit input), **Y** ((Remote/Local)External bit output), **L** (Bit CPU link area), **R** (Internal bit output), **M** (Bit data area), **N** (Timer counter), **CL** (Elapsed count clear), **WX** ((Remote/Local)External word input), **WY** ((Remote/Local)External word output), **WL** (Word CPU link area), **WR** (Internal word output), **WM** (Word data area), **TC** (Timer counter elapsed time), **DIF** (Rising edge detection) and **DFN** (Falling edge detection).

- **Initial address**: As the communication driver establishes serial communications using messages blocks, this driver has been developed to use an initial address and a block size. So, all you need to do to type a right device's variable address is to type here the initial address in **hexadecimal** format and on the *Address* cells (Driver Configuration's Body), just type the offset regarding this initial address. The tables below have a lot of examples how to get the same variable's value using different headers.

Important: If you use the type of operator data with X, Y, WX or WY the initial address must be following syntax:

Local:

- **X | u | s | b |**
- **Y | u | s | b |**
- **WX | u | s | m |**
- **WY | u | s | m |**

Remote:

- **X | r | St | s | b |**
- **Y | r | St | s | b |**
- **WX | r | St | s | b |**
- **WY | r | St | s | b |**

where,

u - Unit number 0 to 2(H250 - 0 to 1).
s - Slot number 0 to 9(H250 - 0 to 8).
b - Intra-module bit number 00 to 5F(must be two digit in hexadecimal).
m - Intra-module word number 0 to 7(must be one digit).
r -Remote host station number (1 to 4)
St - Remote local station number (0 to 7)

For others operators types you must configure the initial address directly in hexadecimal format.

Information regarding the parameter "Header"			
Type	**Sample of syntax**	**Valid range of initial Address**	**Comment**
(Remote/Local) External bit input	X000000	Depends of the CPU model and the amount of I/O card. Local : 928 points (H250, 512 points) Remote : 512 points (Up to 128 points for one host station)	Physical Inputs: "X00<u><s><b>" or "X0<r><St><s><b>" • <u> is the unit number (0 to 2)(H250:0 to 1) • <s> is the slot number (0 to 9)(H250:0 to 8) • <b> is the intra-module bit number(00 to 5F) in hexadecimal. (Two Digit) • <r> is the remote host station number (1 to 4) • <St> is the remote local station (0 to 7) Important: All values are configured in hexadecimal.
(Remote/Local) External bit output	Y000100	Depends of the CPU model and the amount of I/O card. Local : 928 points (H250, 512 points) Remote : 512 points (Up to 128 points for one host station)	Physical Outputs: "Y00<u><s><b>" or "Y0<r><St><s><b>" • <u> is the unit number (0 to 2)(H250:0 to 1) • <s> is the slot number (0 to 9)(H250:0 to 8) • <b> is the intra-module bit number(00 to 5F) in hexadecimal. (Two Digit) • <r> is the remote host station number (1 to 4) • <St> is the remote local station (0 to 7) Important: All values are configured in hexadecimal.

Bit CPU link area	L000000	Depends of the CPU model. Area 1 16.384 points (L000000 to L003FFF) Area 2 16.384 points (L010000 to L013FFF)	Bit CPU link area 1 and Bit CPU link area 2. It is possible to read or write these bits.
Internal bit output	R000000	Depends of the CPU model. 1.984 points (R000000 to R0007BF) Internal special bit output : 64 points (R0007C0 to R0007FF)	Internal bit output and Internal special bit output. It is possible to read or write these bits.
Bit data area	M000000	Depends of the CPU model. 16.384 points (M000000 to M003FFF)	Bit data area. It is possible to read or write these bits.
Timer counter	N000000	Depends of the CPU model. (N000000 to N0001FF)	Timer counter. It is possible to read or write these bits.
Elapsed count clear	CL000000	Depends of the CPU model. 512 points (CL000000 to CL0001FF)	Elapsed count clear It is possible to read or write these bits.
(Remote/Local) External word input	WX000000	Depends of the CPU model and the amount of I/O card. Local : 928 points(58 words) (H250, 512 points(32 words)) Remote : 512 points(32 words) (Up to 128 points for one host station)	Physical Inputs: "WX00<u><s><m>" or "WX0<r><St><s><m>" • <u> is the unit number (0 to 2)(H250:0 to 1) • <s> is the slot number (0 to 9)(H250:0 to 8) • <m> is the intra-module word number (0 to 7) in hexadecimal. (One Digit) • <r> is the remote host station number (1 to 4) • <St> is the remote local station (0 to 7) Important: All values are configured in hexadecimal.
(Remote/Local) External word output	WY000010	Depends of the CPU model and the amount of I/O card. Local : 928 points(58 words) (H250, 512 points(32 words)) Remote : 512 points(32 words) (Up to 128 points for one host station)	Physical Outputs: "WY00<u><s><m>" or "WY0<r><St><s><m>" • <u> is the unit number (0 to 2)(H250:0 to 1) • <s> is the slot number (0 to 9)(H250:0 to 8) • <m> is the intra-module word number (0 to 7) in hexadecimal. (One Digit) • <r> is the remote host station number (1 to 4) • <St> is the remote local station (0 to 7) Important: All values are configured in hexadecimal.
Word CPU link area	WL000000	Depends of the CPU model. Area 1 16.384 points (1.024 words) (WL000000 to WL0003FF) Area 2 16.384 points (1.024 words) (WL001000 to WL0013FF)	Bit CPU link area 1 and Bit CPU link area 2. It is possible to read or write these words.
Internal word output	WR000000	Depends of the CPU model. 1.024 words (H250) (WR000000 to WR0003FF) 17.408 words(H252) (WR000000 to WR0043FF) Internal special word output : 512 words (R00F000 to R00F1FF)	Internal word output and Internal special word output. It is possible to read or write these words.
Word data area	WM000000	Depends of the CPU model. 1.024 words	Word data area. It is possible to read or write these words.

		(WM000000 to WM0003FF)	
Timer counter elapsed time	TC000000	Depends of the CPU model. 512 words (TC000000 to TC0001FF)	Timer counter elapsed time. It is possible to read or write these words.
Rising edge detection	DIF000000	Depends of the CPU model. 512 points (DIF000000 to DIF0001FF)	Rising edge detection It is possible to read or write these bits.
Falling edge detection	DFN000000	Depends of the CPU model. 512 points (DFN000000 to DFN0001FF)	Falling edge detection It is possible to read or write these bits.

4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the column **Tag Name,** you must type the tag from your application database. This tag will receive or send values from or to an address on the device.

The address cells complies to the following syntax:

<offset>.<bit number>

offset: the **offset** is a number in **decimal** that when added to the **initial address reference** configured in the **Header** field defines an address in the device. If you're using the operator data word, it's the Word Offset. If you're using the operator data bit, it's the bit offset.
When you're working with word data block, the highest allowed offset is 120 (0 to 119), but if you're working with bit data block, the highest allowed offset is 240 (0 to 239).

Sample of Addressing Configuration		
Address on the Device	**Header Field**	**Address Field**
X000000	X000000	0
X000005	X000000	5
X000005	X000005	0
Y000100	Y000100	0
Y00010A	Y00010A	0
Y00010A	Y000100	10
R000000	R000000	0
R0000A8	R0000A8	0
R0000A8	R0000A0	8
L001000	L001000	0
L001007	L001006	1
L001007	L001000	7
M00010B	M00010B	0
M00010B	M000109	2
M00010C	M000100	12
Timer Counter 0	N000000	0
Timer Counter 1	N000000	1
Timer Counter 1	N000001	0
CL00001C	CL00001C	0
CL00001C	CL000018	4
CL00001D	CL000019	4
WX000000	WX000000	0
X000000 (WX)	WX000000	0.0
X000001 (WX)	WX000000	0.1
WY000010	WY000010	0
Y000100 (WY)	WY000010	0.0

Y000106 (WY)	WY000010	0.6
WR000000	WR000000	0
WR00001A	WR00001A	0
WR00001B	WR00001A	1
WR000000 (BIT 0)	WR000000	0.0
WM000000	WM000000	0
WM00000F	WM00000E	1
WM000010 (BIT 1)	WM00000E	2.1
TC000000	TC000000	0
TC00005F	TC00005F	0
TC000060	TC00005F	1
TC000060 (BIT 0)	TC00005F	1.0
TC000060 (BIT 4)	TC00005E	2.4
DIF000000	DIF000000	0
DIF000009	DIF000000	9
DIF000009	DIF000009	0
DFN00000A	DFN00000A	0
DFN00000A	DFN000000	10
DFN000000	DFN000000	0
DFN0000AA	DFN0000A0	10

☞ Note: In the previous table there are several ways to set the same variable on the device because of the variable's number defined by the sum of the **initial address reference** defined in the field header and the **offset** defined in the Address Field.

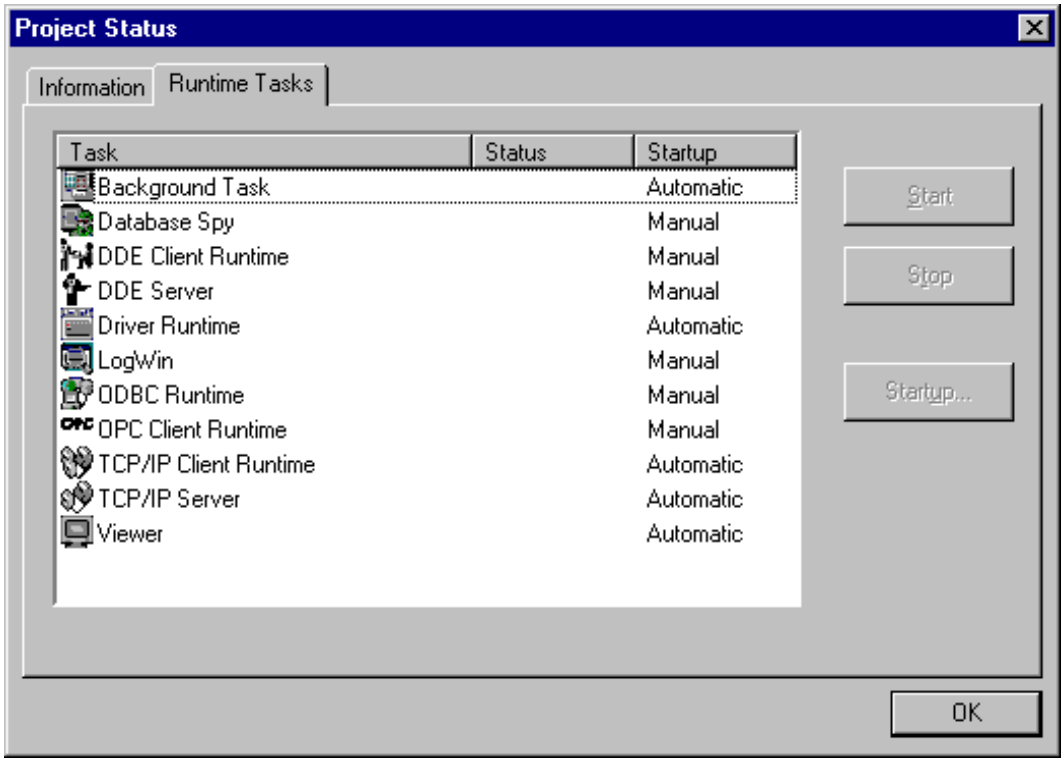
4.5 Device Configuration

The Hitachi H-Series Manual indicates the following serial communication settings:

Baud Rate: 19200
Data bits: 7
Stop bits: 1
Parity: Even

5 Execution

- When installing the driver, it is automatically selected to execute when you start-up the Runtime Environment. To verify the if the driver is correctly enabled to start, use the menu option **Project** + **Status…,** and verify the task Driver Runtime



6 Troubleshooting

After each attempt to communicate using this driver, the tag configured in the field **Read Status** or **Write Status** will receive the error code regarding the kind of failure that occurred. The error messages are:

Error Code	Description (*)	Possible causes	Procedure to solve
0	OK	Communication without problems	-
5	Invalid Protocol (Invalid PLC response)	- Wrong Communication Parameters Configuration. - Wrong RTS/CTS configuration settings. - Invalid PLC configuration.	Check the serial communication configuration. Verify if the settings on the Communication Parameters and on the device are the same.
9	Invalid Header	An invalid Header has been typed or the tag that is inside this field has an invalid configuration.	Type a valid Header either on the header field or on the tag value. A lot of different valid headers are shown on the section 1.1
11	Invalid Address	An invalid Address has been typed.	Type a valid Address either on the Address field. A lot of different valid address are shown on the section 1.2
12	Invalid Block Size	The length of block data size request is very large.	Check the initial address and the major address configured in the sheet. When you're working with word data block, the highest allowed offset is 120 (0 to 119), but if you're working with bit data block, the highest allowed offset is 240 (0 to 239).
13	Invalid Length of the Answer	- Wrong Communication Parameters Configuration. - Wrong RTS/CTS configuration settings. - Invalid PLC configuration.	Check the serial communication configuration. Verify if the settings on the Communication Parameters and on the device are the same.
15	Invalid Number of Points Requested	- Invalid Sheet Configuration.	Check the initial address, the major address configured in the sheet and the type operator data.
17	Invalid Range of I/O Number	- Invalid Sheet Configuration. - The specified operator don't exist in the PLC.	- Check the sheet configuration. - Check PLC configuration.
-15	Timeout waiting start a message.	- Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings.	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.
-17	Timeout between rx char.	- PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings	- Check the cable wiring - Check the PLC state. It must be RUN - Check the station number. - Check the right configuration. See on the section 2.2 the different RTS/CTS valid configurations.

☞ Note: The results of the communication may be verified in the **output** Window of the Studio's environment. To set a log of events for **Field Read Commands**, **Field Write Commands** and **Serial Communication** click with the right button of the mouse on the output window and chose the option setting to select these log events. When testing under a Windows CE target, you can enable the log at the unit (Tools/Logwin) and verify the file celog.txt created at the target unit.

When you are not able to establish the communication with the PLC, you should first of all establish the communication between the PLC Programming Tool and the PLC. Very frequently the communication it is not possible due to a hardware or cable problem, or due an error or lack of configuration at the PLC. Only after the communication between the PLC Programming Software and the PLC is working fine, you can test again the supervisory.

When testing the communication with the Studio, you should first use the application sample described at item 7, instead of the new application that you are creating.

If is required to contact technical support, please have the following information available:

- Operating System (type and version): To find this information use the Tools/System Information option
- Project information: It is displayed using the option Project/Status from the Studio menu
- Driver version and communication log: Available from Studio Output when running the driver
- Device model and boards: please refer to hardware manufacture's documentation

7 Application Sample

The Studio contains a configured project to test the driver. It is strongly recommended to do some tests with this application before beginning the configuration of the customized project, for the follow reasons:

- To understand better the information covered in section 4 of this document.
- To verify that your configuration is working.
- To certify that the hardware used in the test (device + adapter + cable + PC) is in working conditions before beginning the configuration of the applications.

☞ Note: The Application Sample is not available for all drivers.

The Studio application is in the directory: /COMMUNICATION EXAMPLES/<Driver Name>

To perform the test, you need to follow these steps:

- Configure the device communication parameters using manufacturer programmer software.
- Open the application /COMMUNICATION EXAMPLES/<Driver Name>
- Execute the application
- To display the following screen with some information about the communication, please execute the Viewer module in the Studio.

☞ Note: The application for testing may be used like a maintenance screen for the custom application.

8 History of Versions

Version	By	Date	Description of changes
2.01	Roberto V. Junior	30-jul-1999	▪ Driver available for Windows CE
2.03	Eric Vigiani	05-Jan-2009	▪ Implemented an option in the communication parameter to Disable the Release command