

Communication Driver THERM

Driver for serial communication with THERMA
Controller using its RS-485 serial interface

Index

1	INTRODUCTION	2
2	GENERAL CHARACTERISTICS	3
2.1	DEVICE CHARACTERISTICS	3
2.2	LINK CHARACTERISTICS	3
2.3	DRIVER CHARACTERISTICS	3
2.4	INFORMATION ABOUT CONFORMANCE TESTING	4
3	INSTALLATION	5
3.1	INSTALLING THE DRIVER	5
3.2	OTHER SOFTWARE REQUIREMENTS	5
4	DRIVER CONFIGURATION	6
4.1	SETTINGS - COMMUNICATION PARAMETERS	6
4.2	STANDARD DRIVER WORKSHEET	8
4.3	STATION AND HEADER CONFIGURATION	9
4.4	ADDRESS CONFIGURATION	9
5	EXECUTION	10
6	TROUBLESHOOTING	11
7	APPLICATION SAMPLE	12
8	HISTORY OF VERSIONS	13

1 Introduction

The THERM driver enables communication between Studio system and Therma TH2131 device using the proprietary protocol, in accordance with the characteristics covered in this document.


This document contains 8 parts, as follows:

- **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:** THERMA
- **Compatible Equipment:**
 - Controller (model TH2131)


 **Tip:** 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::** Serial RS485 port
- **Physical protocol:** Serial RS232/RS485
- **Logic protocol:** Proprietary
- **Adapters/Convertes:** None
- **Specific PC Board:** None


2.3 Driver Characteristics

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

 **Tip:** 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:

- **THERM.INI:** Internal file of the driver, it should not be modified by the user.
- **THERM.MSG:** Error messages for each error code. It should not be modified.
- **THERM.PDF:** Provides detailed documentation about the driver.
- **THERM.DLL:** Compiled driver.

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

▪ **Supported Commands:**

Command Type	Write	Read
R	–	•
W	•	–
M	•	–

2.4 Information about conformance testing

- **Equipment:** Controller (TH2131 model)
- **Configuration:**
 - Baud Rate: 2400
 - Protocol: Proprietary
 - Data Bits: 8
 - Stop Bits: 1
 - Parity: Odd
 - COM port: COM1
- **Operating System (development):** Windows 2000 + Service Pack 2
- **Operating System (target):** Windows 2000 + Service Pack 2
- **Studio Version:** 4.1
- **Driver version:** 1.00

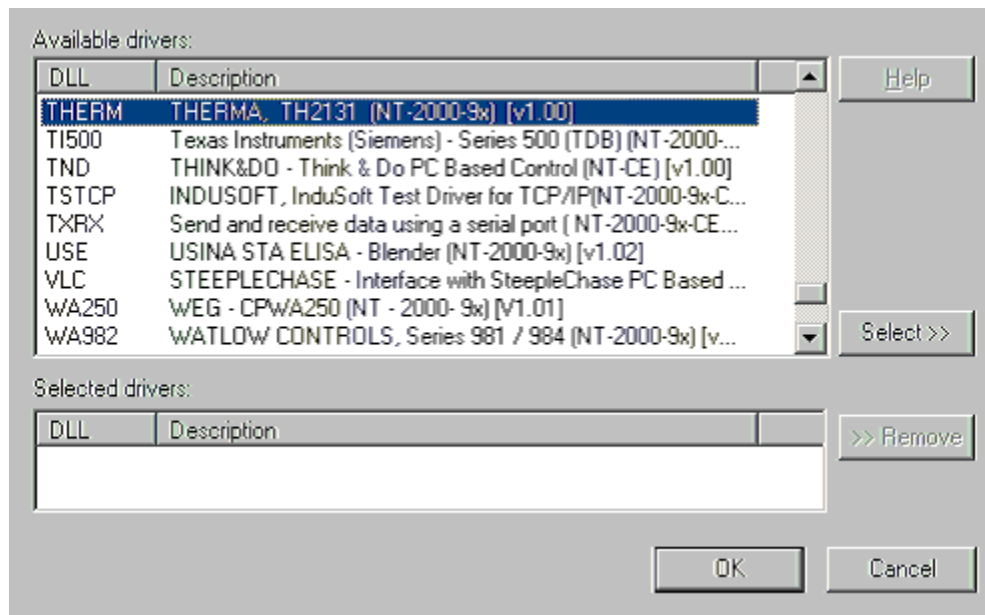
3 Installation

3.1 Installing the Driver

When you install the Studio, 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 **THERM Driver** and push the button **ADD>>>** (the driver THERM 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 Studio and the Device.

Attention: Special precautions 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.

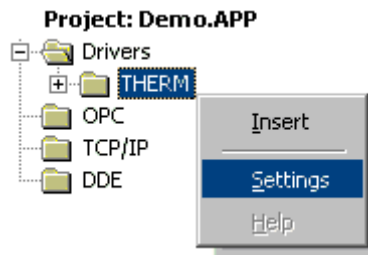
The driver configuration is two parts:

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

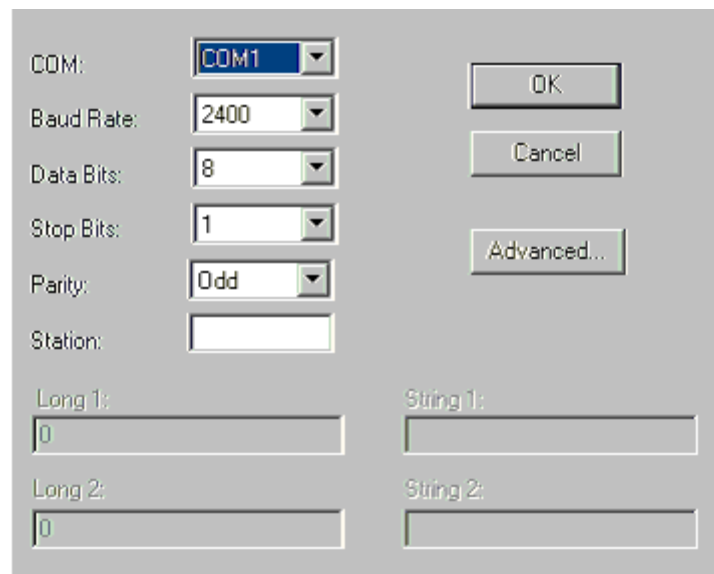
4.1 Settings - Communication Parameters

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 **THERM**.
3. Right click on the **THERM** subfolder and select the option **Settings**.



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

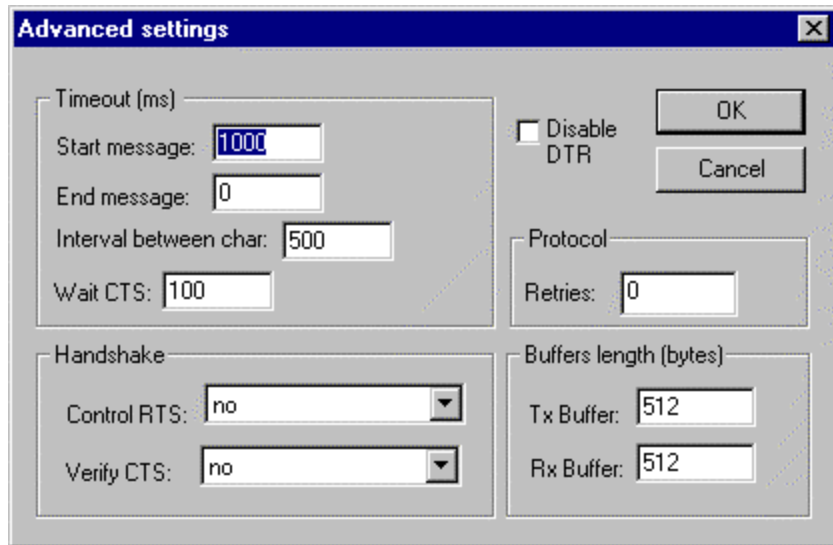
A screenshot of a communication settings dialog box. The dialog has a light gray background and contains the following fields and controls:

- COM: A dropdown menu showing 'COM1'.
- Baud Rate: A dropdown menu showing '2400'.
- Data Bits: A dropdown menu showing '8'.
- Stop Bits: A dropdown menu showing '1'.
- Parity: A dropdown menu showing 'Odd'.
- Station: An empty text input field.
- Long 1: A text input field containing '0'.
- String 1: An empty text input field.
- Long 2: A text input field containing '0'.
- String 2: An empty text input field.
- Buttons: 'OK', 'Cancel', and 'Advanced...' are located on the right side of the dialog.

Parameter	Default Value	Valid values	Description
Station	<Blank>	-	Not used for this driver
Signed=1 or Unsigned=0	0	0 or 1	0=Signed values; 1=Unsigned values. This parameter is the default treatment to integer values.

Note: The device MUST be configured with the SAME values defined in the **Communication Parameters** window of the THERM driver (Baud-rate=2400 ; Stop Bits=1 and Parity=Odd).

By clicking on the button **Advanced...** in the windows **Communication Parameters**, you will be able to configure other serial 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:

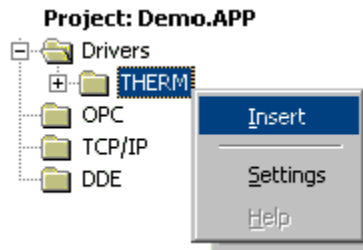
Parameter	Default Value	Valid values	Description
Control RTS	No	no, yes or yes + echo	Define if the handshake signal of RTS (Request to Send) is set before communication and if there is an echo in the communication. If we are using Windows 95 or CE with the right RS 232 – RS 485 Converter (without RTS Control), choose the option “no”. But, if we are under Windows NT and if the Cutler Hammer RS232 – 485 adapter is used, the choice must be “yes”. Important: Wrong settings on this field will not let the driver work, having the Timeout waiting start a message error.

Tip: Usually, these parameters must be changed when using a DCE (Data Communication Equipment) - converter (232/485, for example), modem, etc - between the PC and the device. It is necessary to know the characteristics of the DCE before adjusting these parameters.

4.2 Standard 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 **THERM**.
3. Right click on the **THERM** subfolder and select the option **Insert**.



Tip: 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.

Description:				
<input type="text"/>				<input type="checkbox"/> Increase priority
Read Trigger:	Enable Read when Idle:	Read Completed:	Read Status:	
<input type="text"/>	<input type="text" value="1"/>	<input type="text"/>	<input type="text"/>	
Write Trigger:	Enable Write on Tag Change:	Write Completed:	Write Status:	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
Station:	Header:	<input type="checkbox"/> Min: <input type="text"/> <input type="checkbox"/> Max: <input type="text"/>		
<input type="text" value="1"/>	<input type="text" value="R"/>			

	Tag Name	Address	Div	Add
1	sv	SV		
2				

All entries at the Driver Worksheet, exception by the **Station**, **Header** and **Address** are standard to all communication drivers. You should refer to Studio Technical Reference Manual 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	<None>	1-100	Define address of the controller
Header	R	See next table	Defines the command type

The parameter **Header** defines the command type. It complies with the following syntax:

- To R, M and W:
 <**Command**> (e.g.: W);

- **Command**: Command type (**R** = Read data from the controller, **M** = Move data temporarily to the controller (volatile data); **W** = Write data into the controller EEPROM).

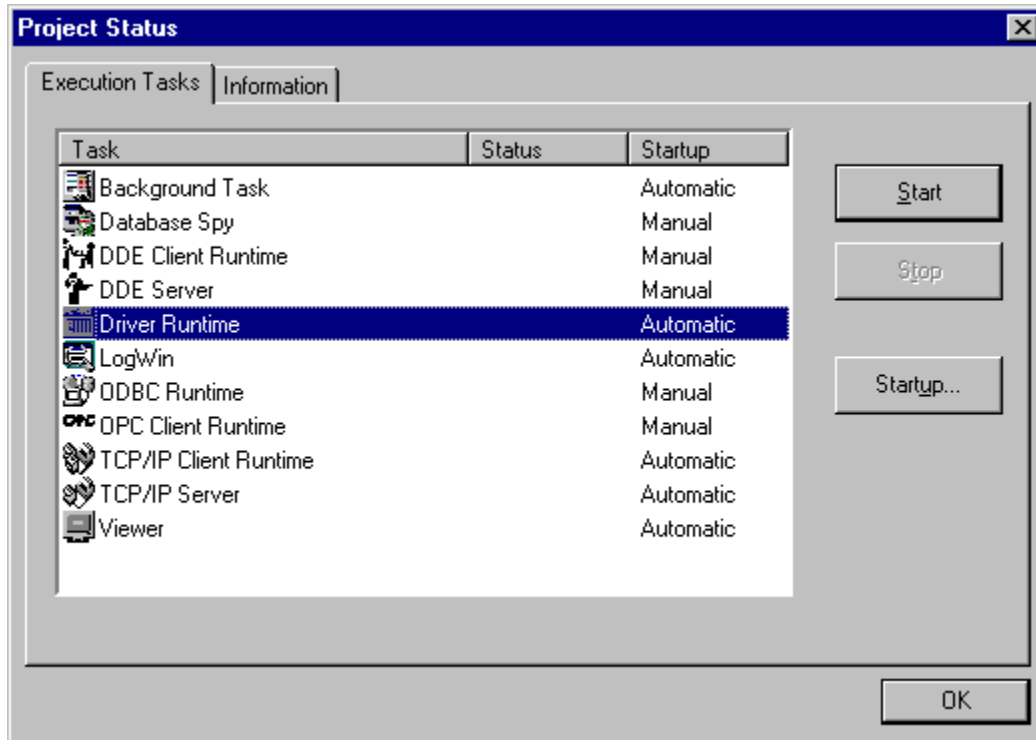
4.4 Address Configuration

The body of the driver worksheet allows you to associate each tag to its respective address in the device. In the **Tag Name** column, 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 with the following syntax <**Address Value**>.

Addressing Configuration						
Header	Address					
R or M or W	PV	SV_4	OUT22	I1	ANL2	SVOS
	OUT%	TM_4	SV_32	D1	ANH2	PVOS
	SV	OUT4	TM_32	DB1	ALD1	UNIT
	OUTL	SV_5	OUT32	ATVL	ALT1	SOFT
	AT	TM_5	SV_42	CYT1	ALD2	CASC
	AL1	OUT5	TM_42	HYS1	ALT2	ODU
	AL2	SV_6	OUT42	P2	ALD3	OPAD
	AL3	TM_6	SV_52	I2	ALT3	HZ
	PTN	OUT6	TM_52	D2	HYSA	SET1
	SEG	SV_7	OUT52	CYT2	CLO1	SET2
	TIMR	TM_7	SV_62	HYS2	CHO1	SET3
	SV_1	OUT7	TM_62	GAP1	CLO2	SET4
	TM_1	SV_8	OUT62	GAP2	CHO2	SET5
	OUT_1	TM_8	SV_72	LCK	CLO3	SET6
	SV_2	OUT8	TM_72	INP1	CHO3	SET7
	TM_2	SV_12	OUT72	ANL1	RUCY	SET8
	OUT2	TM_12	SV_82	ANH1	WAIT	SET9
	SV_3	OUT_12	TM_82	DP	SETA	SET0
	TM_3	SV_22	OUT82	LSPL	IDNO	INP2
	OUT3	TM_22	P1	USPL	BAUD	OUTY

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	-
8	Invalid Command to Read	The command configured in the header is invalid to read.	Verify the table "Supported Commands" in the 2.3 section and fix the operation with this command.
9	Invalid Command to Write	The command configured in the header is invalid to write.	Verify the table "Supported Commands" in the 2.3 section and fix the operation with this command.
-15	Timeout waiting start a message.	<ul style="list-style-type: none"> - Disconnected cables - PLC turned off, or in Stop or error mode - Wrong Station number - Wrong RTS/CTS control settings. 	<ul style="list-style-type: none"> - 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.	<ul style="list-style-type: none"> - PLC in stop or error mode - Wrong station number - Wrong parity - Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> - 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.

 **Tip:** The communication status can be verified by the **output** Window of the Studio's environment or by the **LogWin** module. 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 controller, first of all establish the communication between the controller Programming Tool and the controller. Very frequently the communication is not possible due to a hardware or cable problem, or due an error or lack of configuration at the controller. Only after the communication between the controller Programming Software and the controller is working fine, you can test again the supervisory driver.

When testing the communication with the Studio, you should first use the application sample described at item 7 (if it's available), 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

Studio provides 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.

 **Tip:** The application for testing may be used like a maintenance screen for the custom application.

8 History of Versions

Doc. Revision	Driver Version	Author	Date	Description of changes
A	1.00	Eric Vigiani	Sep/12/2001	▪ Initial version
B	1.01	Eric Vigiani	Apr/02/2002	▪ Included Signed or Unsigned values