

COWAF Communication Driver

Driver for Serial Communication
with Omron E5ZN Temperature Controller
Using CompoWay/F Protocol

Contents

INTRODUCTION	2
GENERAL INFORMATION.....	3
DEVICE SPECIFICATIONS.....	3
NETWORK SPECIFICATIONS.....	3
DRIVER CHARACTERISTICS	3
CONFORMANCE TESTING	4
SELECTING THE DRIVER	5
CONFIGURING THE DEVICE	6
CONFIGURING THE DRIVER	6
CONFIGURING THE COMMUNICATION SETTINGS	6
CONFIGURING THE DRIVER WORKSHEETS	7
EXECUTING THE DRIVER	13
TROUBLESHOOTING	14
REVISION HISTORY.....	16

Introduction

The COWAF driver enables communication between the Studio system and an Omron E5ZN Temperature Controller using the CompoWay/F protocol, according to the specifications discussed in this document.

This document will help you to select, configure and execute the COWAF 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 COWAF driver in the Studio system.
- **Configuring the Device:** Describes how the target device must be configured to receive communication from the COWAF driver.
- **Configuring the Driver:** Explains how to configure the COWAF driver in the Studio system, including how to associate database tags with device registers.
- **Executing the Driver:** Explains how to execute the COWAF driver during application runtime.
- **Troubleshooting:** Lists the most common errors for this driver, their probable causes, and basic procedures to resolve them.
- **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 COWAF driver in Studio and an Omron E5ZN Temperature Controller using the CompoWay/F protocol.

The information is organized into the following sections:

- Device Specifications
- Network Specifications
- Driver Characteristics
- Conformance Testing

Device Specifications

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

- **Manufacturer:** Omron
- **Compatible Equipment:** E5ZN Temperature Controller
- **Programmer Software:** CX-Thermo

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:** RS232 Serial Port
- **Physical Protocol:** RS232
- **Logic Protocol:** CompoWay/F (proprietary)
- **Device Runtime Software:** None
- **Adapters/Converters:** RS-232/RS-485 (E5ZN)

Driver Characteristics

The COWAF driver package consists of the following files, which are automatically installed in the `\DRV` subdirectory of Studio:

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

 **Note:**

You must use Adobe Acrobat® Reader™ to view the **COWAF.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 COWAF driver on the following operating systems:

- Windows NT/2000/XP
- Windows CE

For a description of the operating systems used to test driver conformance, see “Conformance Testing” below.

The COWAF driver supports the following registers:

Variable Area	Length in Bytes	Write	Read
C0	4	–	•
C1	4	•	•
C3	4	•	•
C5	4	•	•
C7	4	•	•

Conformance Testing

The following hardware/software was used for conformance testing:

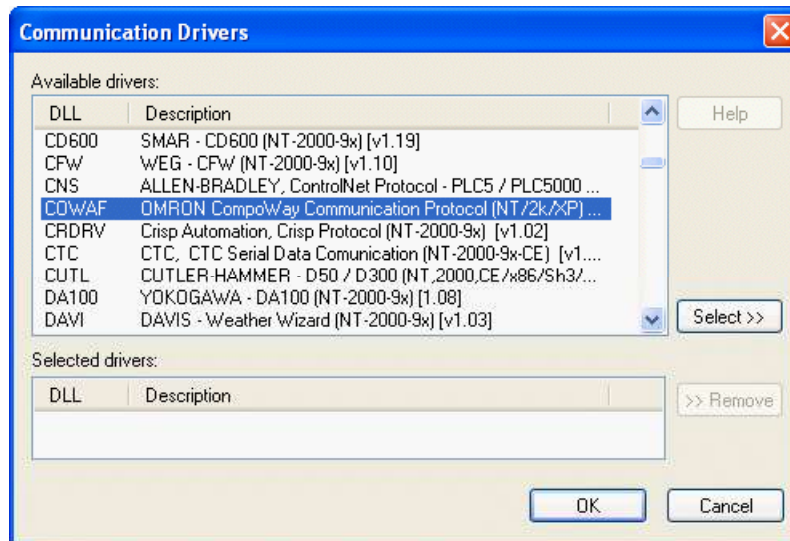
- **Serial Port Configuration:**
 - **Baud Rate:** 9600
 - **Protocol:** Proprietary
 - **Parity:** Even
 - **Data Bits:** 7
 - **Stop Bits:** 2
 - **COM Port:** COM1

Driver Version	Studio Version	Operating System	Equipment
1.00	6.1	Windows XP	E5ZN Temperature Controller

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 COWAF driver for your Studio application:

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



Communication Drivers Dialog

3. When the **COWAF** 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 an E5ZN, you must also install the programming software (e.g., CX-Thermo). 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 Device

We suggest using the following configuration:

- **Protocol:** Proprietary
- **COM Port:** COM1
- **Baud Rate:** 9600
- **Data Bits:** 7
- **Stop Bits:** 2
- **Parity:** Even

Configuring the Driver

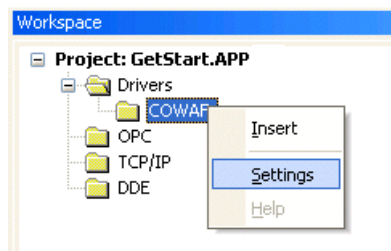
Once you have selected the COWAF driver in Studio, you must properly configure it to communicate with your target device. First, you must set the driver’s communication settings to match the parameters set on the device. Then, you must build driver worksheets to associate database tags in your Studio application with the appropriate addresses (registers) on the device.

Configuring the Communication Settings

The communication settings are 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.

For the purposes of this document, only COWAF driver-specific settings and procedures will be discussed here. To configure the communication settings for the COWAF driver:

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



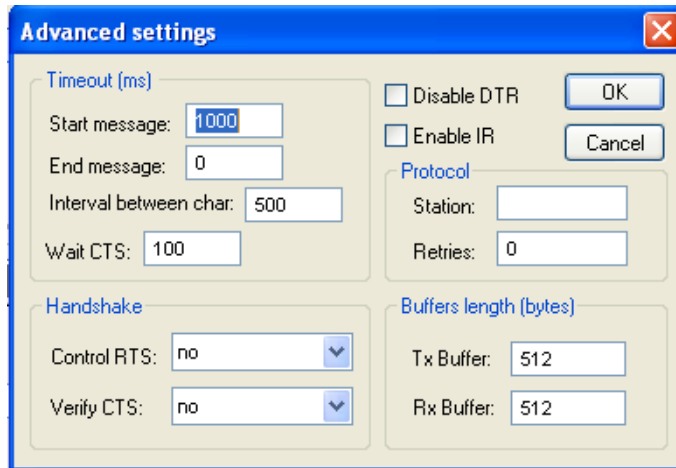
Select Settings from the Pop-Up Menu



COWAF: Communication Parameters Dialog

3. Verify the Serial Port settings and change them if necessary.

- If you are using a Data Communication Equipment (DCE) converter (e.g., 232/485) between your PC and your target device, then you must also adjust the **Control RTS** (Request to Send) setting to account for the converter. In the *Communication Settings* dialog, click the **Advanced** button to open the *Advanced Settings* dialog:



Advanced Settings Dialog

When the dialog is displayed, configure the **Control RTS** setting using the following information:

Setting	Default	Values	Description
Control RTS	no	no	Do not set the RTS (Request to Send) handshake signal. IMPORTANT: If you are using Windows 95/98 or Windows CE with the correct RS232/RS485 adapter (i.e. without RTS control), then you must select this option.
		yes	Set the RTS (Request to Send) handshake signal before communication. IMPORTANT: If you are using Windows NT and the Cutler-Hammer RS232/RS485 adapter, then you must select this option.
		yes+echo	Set the RTS (Request to Send) handshake signal before communication, and echo the signal received from the target device.

➔ **Attention:**
 If you incorrectly configure the **Control RTS** setting, then runtime communication will fail and the driver will generate an error. See “Troubleshooting” for more information.

You do not need to change any other advanced settings at this time. You can consult the *Studio Technical Reference Manual* later for more information about configuring these settings.

- Click **OK** to close the *Advanced Settings* dialog, and then click **OK** to close the *Communication Parameters* dialog.

Configuring the Driver Worksheets

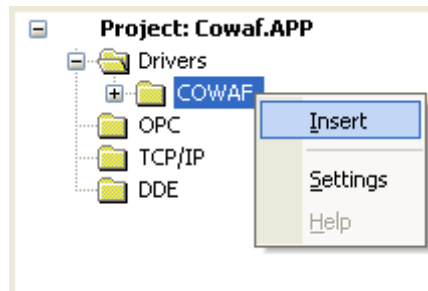
A selected driver includes one or more driver worksheets, which are used to associate database tags in Studio with registers on the target device. Each worksheet is triggered by specific application behavior, so that the tags / registers defined on that worksheet are scanned only when necessary – that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



Inserting a New Worksheet

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

#	Tag Name	Address	Div	Add
1	tagTest[16]	0000		
2	tagTest[17]	0100		
*				

COWAF Driver Worksheet

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 COWAF driver.

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


- **Station** field: Identify the target device, using the following syntax:

<Node ID>

Example — 32

Where **<Node ID>** is the ID number (00 to 99) of the target device on the serial network, as determined by the device configuration. (If you want to broadcast a command to all devices on the serial network, then set the **<Node ID>** to **XX**.)

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 cannot leave the **Station** field blank; you must specify some value.

- **Header** field: Specify a section of memory on the target device. The addresses declared in the *Body* of the worksheet will automatically be of this section. 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** uses the following syntax:

<VariableArea>

Example — c1

Where **<VariableArea>** is the section of memory on the E5ZN Temperature Controller. Valid values include **c0** (read only), **c1**, **c3**, **c5** and **c7**.

VariableArea	Valid Range of Initial Addresses	Channel	Description
c0	0000 to 0006	1	Read-only data for settings area 0 in double-word format
	0100 to 0106	2	
c1	0000 to 0019	1	Read/write data for settings area 0 in double-word format
	0100 to 0119	2	
c3	0000 to 003E	1	Read/write data for settings area 1 in double-word format
	0100 to 013E	2	
c5	0000	1	Read/write data for settings area 0 in double-word format
	0100	2	
c7	0000 to 0001	–	Read/write data for settings area 1 in double-word format
	0100 to 0101	–	

 **Note:**

- For a complete description of any address information, please refer to the manufacturer's documentation for the E5ZN Temperature Controller.
- Before executing the command, use an operation command to enable communications writing(CWR).

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 **C0**.

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

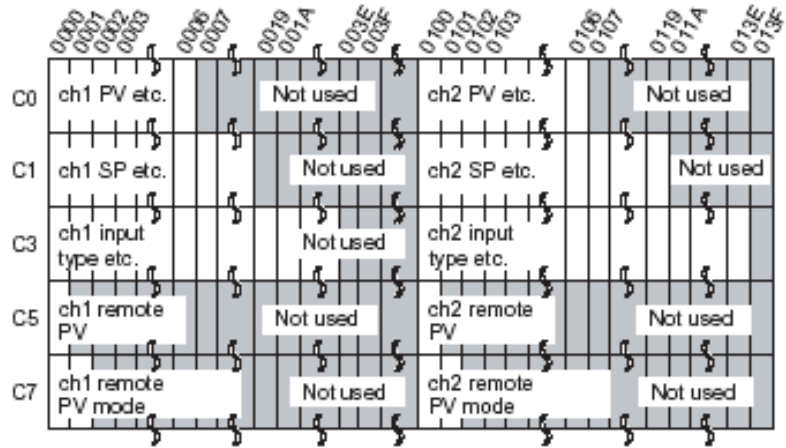
<Address>

Examples — **0003, 001A**

Where **<Address>** is a parameter that is combined with the **<VariableArea>** parameter of the **Header**, to compose the specific address of the register on the device. This value is a four-digit hexadecimal.

For examples of how register addresses are composed using the values in the **Header** and **Address** fields, see the following table and illustration:

Device Register	Header Value	Address Value
C0000	C0	0000
C10003	C1	0003
C30008	C3	0008
C50000	C5	0000
C70001	C7	0001
C00100	C0	0100
C10011	C1	0011
C3000A	C3	000A
C50100	C5	0100
C70100	C7	0100



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

OPERATION COMMANDS (RAM, RESET, MOVE1, CWR, WRITEMODE, RUNSTOP AND AUTOTUNE)

As an alternative to defining tag/register associations, you can configure a driver worksheet to send operation commands to the target device. The commands are sent according to **Write Trigger** and **Enable Write on Tag Change**, just as with a normal worksheet. Available commands include:

- **RUNSTOP** starts or stops the target device. This command can be used in either setting area 0 or 1.
- **AUTOTUNE** stops or executes autotuning on the target device. This command can be used in setting area 0 only; if it is used in setting area 1, then the device will return an operation error.
- **RAM** writes operation/adjustment level setting data to internal nonvolatile memory. This command can be used for either setting areas 0 or 1.
- **RESET** returns the Temperature Controller to a initial state when it was turned ON. This command can be used for either setting areas 0 or 1.
- **MOVE1** move to settings area1. Use this command in setting area 0.
- **CWR** enables/disables Communications Writing, If communications writing is disabled any command will be rejected. Default is it enabled.

- **WRITEMODE** selects backup mode and RAM mode. Default is RAM write mode. This command can be used in either setting area 0 or 1. When switching from RAM write mode to backup mode, the operation/adjustment level setting data is written to internal non-volatile memory.

 **Note:**

- For a complete description of operation commands, please refer to the manufacturer's documentation for the E5ZN Temperature Controller.
- Before executing the command, use an operation command to enable communications writing (CWR).

To configure a driver worksheet to send operation commands:

1. Insert a new driver worksheet, as described previously.
2. Configure the **Station** and **Header** fields as follows:
 - **Station** field: Identify the target device, using the following syntax:
<Node ID>
Example — 32

Where **<Node ID>** is the ID number (00 to 99) of the target device on the serial network, as determined by the device configuration. (If you want to broadcast a command to all devices on the serial network, then set the **<Node ID>** to **XX**.)

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.
 - **Header** field: Specify the desired operation command (**RAM**, **RESET**, **MOVE1**, **CWR**, **WRITEMODE**, **RUNSTOP** or **AUTOTUNE**) for this worksheet.
3. For each table row, specify the associated database tag in the **Tag Name** field and then enter a value of 0 in the **Address** field.

For operation commands, use the following syntax:

<Address>

Example — 0000

Where **<Address>** must be zero, following the example.

 **Attention:**

If you configure more than one table row (i.e., more than one tag association), then you run the risk of having multiple operation commands conflict with and override each other. Unless it cannot be avoided, only one database tag should be configured to control when and how an operation command is sent to the target device.

Keep in mind that the operation command configured in the **Header** field is only a general command that has several available options. To send a desired command option to the target device during runtime, you must develop your application to write the corresponding value to the database tag configured in the **Tag Name** field. The command options and their corresponding values are described in the following table:

Operation Command in Header field	Value Written to Database Tag	Address field	Command Option Sent to Device
RUNSTOP	0	0000	Run channel 1
	1		Stop channel 1
	2		Run channel 2
	3		Stop channel 2
	4		Run channels 1 and 2 (see note)
	5		Stop channels 1 and 2 (see note)
AUTOTUNE	0	0000	Stop autotuning on channel 1
	1		Execute autotuning on channel 1
	2		Stop autotuning on channel 2
	3		Execute autotuning on channel 2
	4		Stop autotuning on channels 1 and 2 (see note)
	5		Execute autotuning on channels 1 and 2 (see note)
RAM	0	0000	Writes operation/adjustment level setting data to internal memory
	2		
RESET	0	0000	Reset the Temperature Controller
	2		
MOVE1	0	0000	Move to setting area 1
	2		
CWR	0 or 2	0000	Communication Writing prohibited (OFF)
	1 or 3		Communication Writing not prohibited (ON)
WRITEMODE	0 or 2	0000	Backup mode
	1 or 3		RAM write mode



Note:

The indicated command options are supported on upgraded pulse output and analog output models *only*.



Tip:

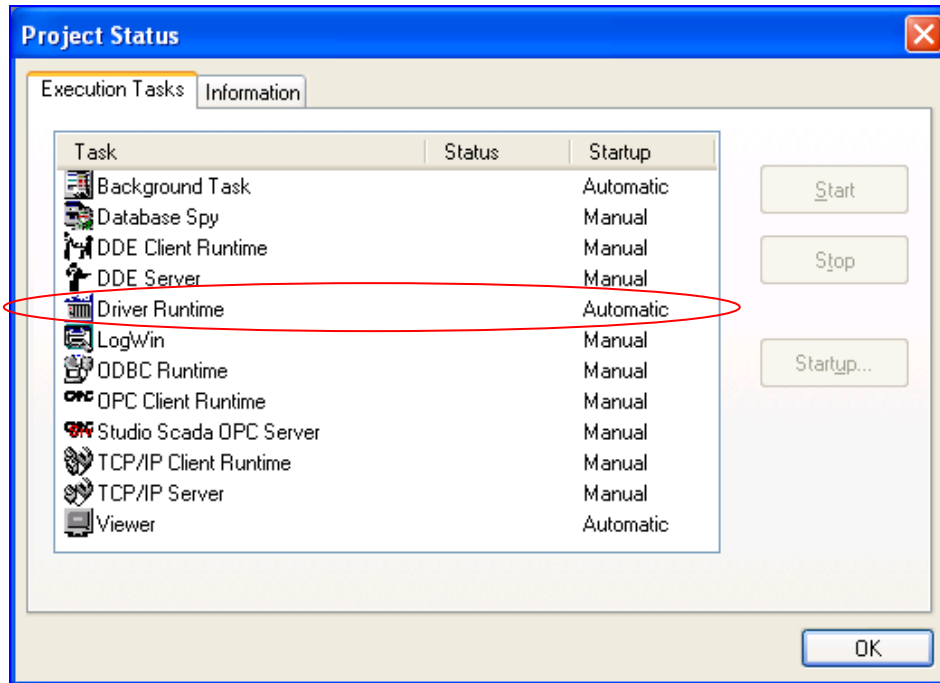
You can configure the worksheet's **Enable Write on Tag Change** field so that an operation command is immediately sent to the target device whenever a new value is written to the database tag configured in the **Tag Name** field.

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 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 COWAF 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	Procedure to Solve
-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 on General Information section <i>Network Specification</i>.
-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 on General Information section <i>Network Specification</i>..
0	Normal Completion	Communication without problems	None required
10	Parity Error	The sum of the bits (received data (1)) does not match.	Check parity at Studio Communication Parameters.
11	Framing Error	The number of character stop bits for the command frame was 0.	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see “Network Specifications” for valid RTS/CTS configurations).
12	Overrun Error	A data transfer was attempted when the reception data buffer was already full.	Contact PLC manufacturer.
13	BCC Error	The received BCC was different from the calculated BCC.	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see “Network Specifications” for valid RTS/CTS configurations). ▪ Check device and Studio Communications Parameters.
14	Format Error	The command text contains characters other than “0” to “9” and “A” to “F.” No SID and command text. MRC and SRC in the command text were not included in the command text.	Check the address field. Replace all the configured invalid addresses.
16	Sub-Address Error	No sub-address, SID, and command text. This error is not covered by the echoback test. The size of the sub-address was less than two characters, and no SID and command text were found.	Contact the manufacturer’s technical support.
18	Frame length Error	The size of the received frame exceeded the specified number of bytes.	Reduce the number of elements at Standard Worksheet.
0F	FINS Command Error	The specified FINS command could not executed.	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the RTS/CTS configuration (see “Network Specifications” for valid RTS/CTS configurations). ▪ Check device and Studio Communications Parameters.
100	Error Invalid Header	The configured header is invalid	Check the documentation to choose a valid header

101	Error Nonexisting Operation	Tried a invalid operation	Choose a valid operation
102	Error Invalid Command	Reading or writing when it is not permitted	Check the documentation to configure the Worksheet appropriately for each header. Check if the reading and writing are allowed.
103	Error Invalid Option	Tried to reading or writing a invalid value	Check the commands to know the valid values for each case.
104	Error Invalid Read	The reading could not be completed	Check the error that came from device in Logwin.
105	Error Invalid Write	The writing could not be completed	Check the error that came from device in Logwin.

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

You can also use the *LogWin* module (**Tools** → **LogWin**) to establish an event log on a remote unit that runs Windows CE. The log is saved on the unit in the `ce1log.txt` file, which can be downloaded later.

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., CX-Thermo). 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.

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	Diego Barros	Apr/13/2006	<ul style="list-style-type: none">▪ Initial version▪ Driver available for Windows NT/2000/XP/CE
B	1.00	Michael D. Hayden	Jun/28/2006	Edited for language and usability.
C	1.00	Graziane C. Forti	Oct/05/2006	<ul style="list-style-type: none">▪ Implemented RAM, RESET and MOVE1 commands▪ Implemented CWR command (Communications Writing)
D	1.00	Eric Vigiani	Oct/24/2006	Implemented WRITEMODE command
E	1.00	Graziane C. Forti	Jan/03/2007	<ul style="list-style-type: none">▪ Optimized Driver▪ Fixed problem RESET command
F	1.1	André Körbes	Jan/07/2013	<ul style="list-style-type: none">▪ Fixed station validation when using 0 as Node ID