CTC Communication Driver

Driver for Serial Communication with CTC
Devices from 2200 / 2400 / 2800 Controllers

Contents

Introduction

The CTC driver enables communication between the Studio system and CTC devices for 2200 / 2400 / 2800 Controllers communicating over Serial, according to the specifications discussed in this document.

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

- **Introduction**: Provides an overview of the CTC driver documentation.

- **General Information**: Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the CTC driver.

- **Installing the Driver**: Explains how to install the CTC driver.

- **Configuring the Driver**: Explains how to configure the CTC 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 CTC 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 CTC driver and the CTC devices for 2200 / 2400 / 2800 Controllers.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

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

- **Manufacturer**: Control Technology Corporation (CTC)
- **Compatible Equipment**:
  – 2200 / 2400 / 2800 Controllers
- **Programming Software**: None
- **Device Runtime Software**: None

For a list of the devices used for conformance testing, see "Conformance Testing."

Link Characteristics

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

- **Device Communication Port**: RS232 Port
- **Physical Protocol**: RS232/RS485
- **Logic Protocol**: Binary (Proprietary)
- **Specific PC Board**: None

Driver Characteristics

The CTC driver is composed of the following files:

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

You can use the CTC 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 5.

The CTC driver supports the following registers:

Register Type	Length	Write	Read
Register	2 Word	●	●
Register Group (16)	2 Word	–	●
Register Group (50)	2 Word	–	●
Analog Output	1 Word	●	●
Outputs Group (8)	1 bit	●	●
Outputs Group (128)	1 bit	●	●
Analog Input	1 Word	–	●
Inputs Group (8)	1 bit	–	●
Inputs Group (128)	1 bit	–	●
Servo Position	2 Word	–	●
Servo Error	2 Word	–	●
Servo Inputs	1 bit	–	●
System Configuration	1 bit	●	●
Status Byte	1 bit	–	●
Data Table Row	1 Word	●	●
Data Table Dimensions	1 Word	●	●
Flag	1 bit	●	●
I/O Count Request	1 Word	–	●
Misc. I/O Count Request	1 Byte	–	●
Controller Step Status Request	1 Word	–	●
Controller Status	1 bit	●	–

Conformance Testing

The following hardware/software was used for conformance testing:

- **Driver Configuration**:
    - **Baud Rate**: 9600
    - **Protocol**: Binary
    - **Data Bits**: 8
    - **Stop Bits**: 1
    - **Parity**: None
    - **COM Port**: COM1

- **Cable**: RS232 cable

Driver Version	Studio Version	Operating System	Equipment
1.02	6.0	WinXP	CTC 2200 Controller

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 **CTC** driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.
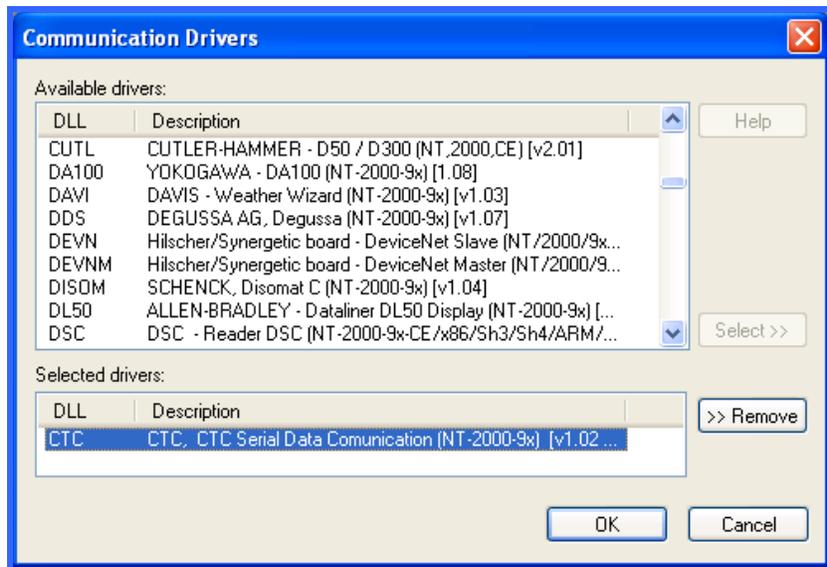
Communication Drivers Dialog Box

5. When the **CTC** 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 CTC driver, you must configure the driver. Configuring the CTC driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEET*s (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 SHEET*s, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

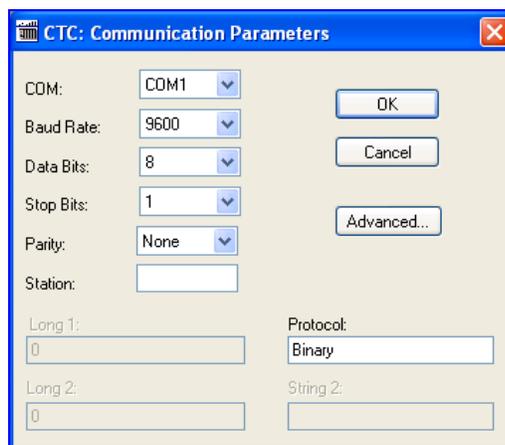## *Setting the Communication Parameters*

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 CTC subfolder and when the pop-up menu displays (as shown in the following figure), select the **Settings** option.
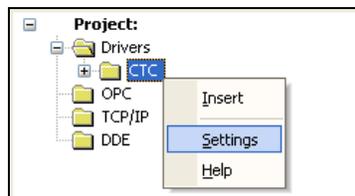
Select Settings from the Pop-Up Menu

The *CTC: Communication Parameters* dialog displays (as follows).

Communication Parameters Dialog

4. Specify the custom parameters as noted in the following table:

Parameters	Default Values	Valid Values	Description
Station	0	0	Not used for this driver
Protocol	Binary	Binary, Terminal or Computer	Protocol used to communicating with devices.

☞ **Note:**

The device must be configured with *exactly the same* parameters that you configured in the *CTC Communication Parameters* dialog.

---

5. 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.


<u>Configuring the Driver Worksheets</u>

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 *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 *CTC* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.
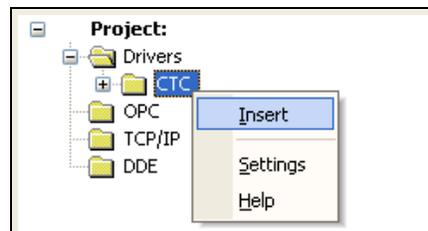
Inserting a New Worksheet

✎ **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).

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 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 Controller address (from *1* to 255).

   ▪ **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.

   ▪ **Address** field: Use this field to associate each tag to its respective device address.

      Use the following table to Configure the **Header** and **Address** fields:

Register Type	Standard Driver Sheet Configuration		Address Description	Action (Read=R ; Write=W)
	Header	Address		
Register	R	1 .. 65535	Register number	RW
Register Group (16)	R16:<first register> (1 .. 986)	1 .. 16	Register number	R
Register Group (50)	R50:<bank> (1 .. 20)	1 .. 50	Register number	R
Analog Output	AO	1 .. 256	Analog Output number	RW
Outputs Group (8)	O : <bank> (1 .. 128)	1 .. 8	Output number	RW
Outputs Group (128)	O128 : <bank> (1 .. 8)	1 .. 128	Output number	RW
Analog Input	AI	1 .. 256	Analog Input number	R
Inputs Group (8)	I: <bank> (1 .. 128)	1 .. 8	Input number	R
Inputs Group (128)	I128: <bank> (1 .. 8)	1 .. 128	Input number	R
Servo Position	SP	1 .. 16	Servo Axis number	R
Servo Error	SE	1 .. 16	Servo Axis number	R
Servo Inputs	SI: <Servo Axis number> (1..16)	0	"HOME" input	R
		1	"START" input	R
		2	"LOCAL/REMOTE" input	R
		3	"REVERSE LIMIT" input	R
		4	"FORWARD LIMIT" input	R
System Configuration	SCONF	0	'1' if using input #1 for START function	RW
		1	'1' if using input #2 for STOP function	RW
		2	'1' if using input #3 for RESET function	RW
		3	'1' if using input #4 for STEP function	RW
Status Byte	STB	0	0=running; 1=stopped	R
		1	0=normal mode; 1=programming mode	R
		2	0=status ok; 1=software fault	R
		3	0=mid-program; 1=fresh reset	R
Data Table Row	TR<Data Table Row number>	0 .. X	Column number	RW
Data Table Dimensions	TD	0	Amount of Table Rows configured in the current program	RW
		1	Amount of Table Columns configured in the current program	RW
Flag	F	1 .. 32	Flag number	RW
I/0 Count Request	CR	0	Amount of Flags configured in the current program	R
		1	Amount of Inputs configured in the current program	R

		2	Amount of Outputs configured in the current program	R
		3	Amount of Stepping Motor Axes configured in the current program	R
		4	Amount of Servo Axes configured in the current program	R
		5	Amount of Analog Inputs configured in the current program	R
		6	Amount of Analog Outputs configured in the current program	R
Misc. I/0 Count Request	MCR	0	Amount of Prototyping Boards in the controller	R
		1	Amount of High-Speed Counting Channels in the controller	R
		2	Amount of Thumbwheel Arrays connected to the controller	R
		3	Amount of Numeric Displays connected to the controller	R
Controller Step Status Request	STR:<bank> <bank> = 1 to Tasks 1 to 8 = 2 to Tasks 9 to 16 = 3 to Tasks 17 to 24 = 4 to Tasks 25 to 32	0	0=controller running; 1=controller stopped	R
		1	Software Fault Code	R
		2	Step number for Software Fault	R
		3	Data relating to the Software Fault	R
		4	Step being executed by the task #1	R
		5	Status of the task #1	R
		6	Step being executed by the task #2	R
		7	Status of the task #2	R
		8	Step being executed by the task #3	R
		9	Status of the task #3	R
		10	Step being executed by the task #4	R
		11	Status of the task #4	R
		12	Step being executed by the task #5	R
		13	Status of the task #5	R
		14	Step being executed by the task #6	R
		15	Status of the task #6	R
		16	Step being executed by the task #7	R
		17	Status of the task #7	R
		18	Step being executed by the task #8	R
		19	Status of the task #8	R
Controller Status	CST	0	0=to start controller; 1=to stop controller	W
		1	0=to not reset; 1= to reset	W

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 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, or you will get an invalid Header error.

Device Configuration

You must be sure to specify the same parameters for the CTC device (*Baud Rate*, *Stop Bits*, and so forth) that you configured for the CTC driver.

Executing the Driver

After adding the CTC 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.
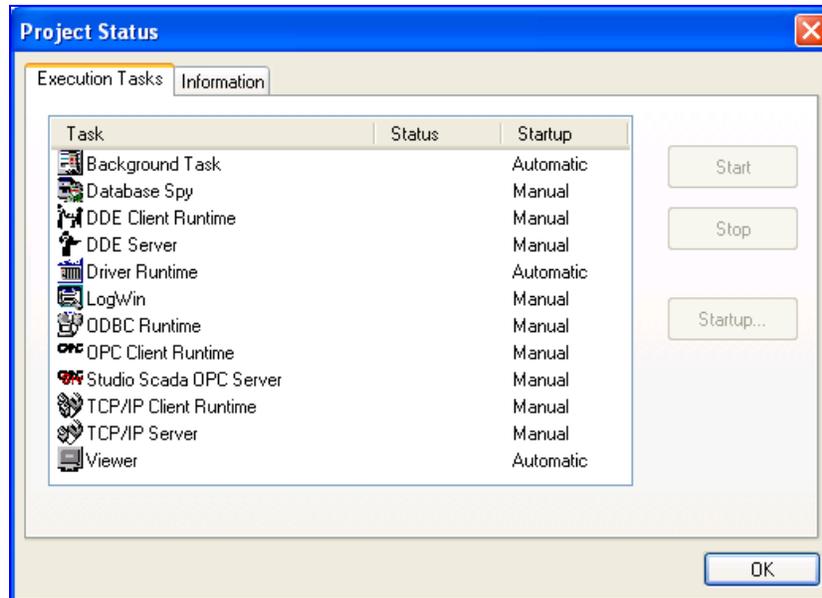
Project Status Dialog Box

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 CTC 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 what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required.
1	Invalid Address	You typed an invalid address, or the tag in this field has an invalid configuration.	Type a valid address in the **Address** field or the tag value.
2	Invalid Header	You typed an invalid Header, or the tag in this field has an invalid configuration.	Type a valid header in the **Header** field or the tag value.
4	Invalid Command	Addressed module did not accept the sent command	Verify the module and be sure that you are not trying to write in the input module, for example.
-15	Timeout Start Message	▪ Disconnected cables ▪ PLC is turned off, in stop mode, or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings	▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See *Studio Technical Reference Manual* for information about valid RTS/CTS configurations.
-17	Timeout between rx char	▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings	▪ Check cable wiring. ▪ Check the PLC state – it must be RUN. ▪ Check the station number. ▪ Check the configuration. See *Studio Technical Reference Manual* for information about valid 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 when 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/CTC** directory. We strongly recommend that you check for a sample application for this driver and use it to test the driver before configuring your own 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
-	1.00	Eric Vigiani	Sep/20/2002	First driver version
A	1.01	Roberto V. Jr	Jan/07/2003	Modified to work in the ANSI version
B	1.02	Fabio H.Y. Komura	Jun/24/2004	▪ Implemented write operations for Output Registers ▪ Fixed some problems with Data Table, System Configuration and Controller Status