

Communication Driver MCTRL

Driver for Ethernet and Serial Communication (TCP/IP and RS-232)
with Motion Control Devices

Contents

CONTENTS1

INTRODUCTION2

GENERAL CHARACTERISTICS.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS.....3

 DRIVER CHARACTERISTICS3

INSTALLING THE DRIVER4

CONFIGURING THE DRIVER5

 SETTING THE COMMUNICATION PARAMETERS5

 CONFIGURING THE STANDARD DRIVER WORKSHEET7

Configuring the Header Field8

Configuring the Address Field 10

EXECUTING THE DRIVER 13

TROUBLESHOOTING 14

REVISION HISTORY..... 15

Introduction

The MCTRL driver enables communication between the Studio system and Motion Control devices according to the specifications discussed in this document.

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

- **Introduction:** Provides an overview of the MCTRL driver documentation.
- **General Characteristics:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the MCTRL driver.
- **Installing the Driver:** Explains how to install the hardware and software components required for the MCTRL driver.
- **Configuring the Driver:** Explains how to configure the communication driver, including the different permutations for configuration and the driver's default values.
- **Configuring the Device:** Provides the required information to setup and configure the communication.
- **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.
- **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 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 Characteristics

This chapter explains how to identify all the hardware and software components used to implement communication between Studio's MCLTR driver and Motion Control devices.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

This driver has been tested with the following devices:

- **Galil:** DMC-1415 and DMC-2180
- **Yaskawa:** SMC3010

Link Characteristics

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

- **Device Communication Port:** RS-232 Port or Ethernet Port
- **Physical Protocol:** RS-232 or TCP/IP
- **Logic Protocol:** Proprietary (ASCII)
- **Device Run-time Software:** None
- **Specific PC Board:** None

Driver Characteristics

The MCTRL driver is composed of the following files:

- **MCTRL.INI:** Internal driver file. *You must not modify this file.*
- **MCTRL.MSG:** Error messages for each error code. *You must not modify this file.*
- **MCTRL.PDF:** Document providing detailed information about the MCTRL driver.
- **MCTRL.DLL:** Compiled driver.

Notes:

- All of the preceding files have been 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 MCTRL.PDF document.

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

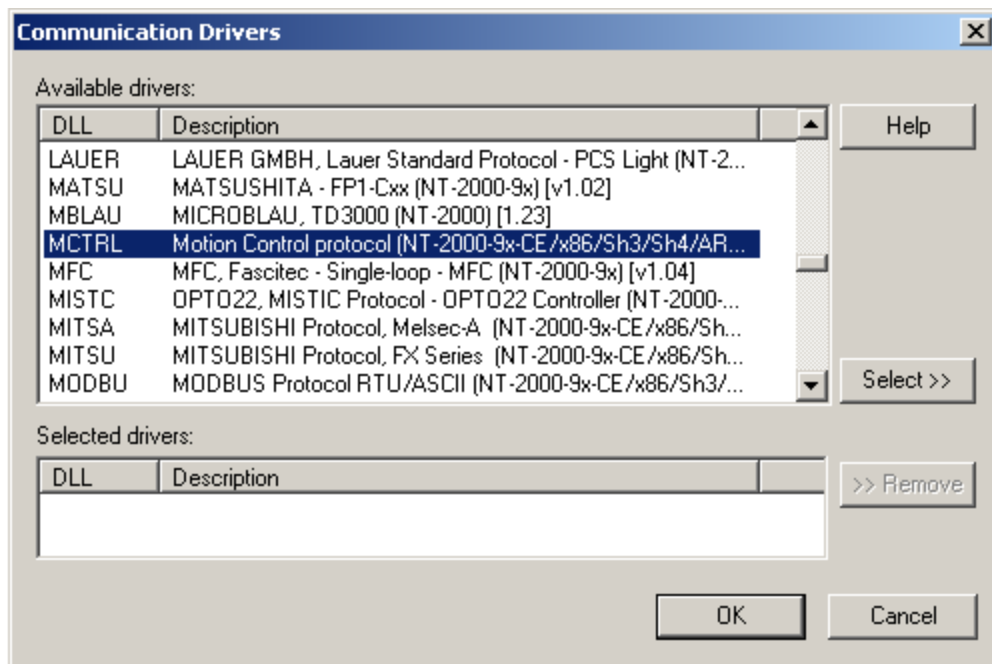
- Windows 2000
- Windows NT
- Windows XP
- Windows CE

Installing the Driver

When you install Studio, 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. Execute the Studio development environment.
2. When the Studio application screen launches, select **File** → **Open Project** from the main menu bar to open the appropriate application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication drivers* dialog (as shown in the following figure).



Communication Drivers Dialog Box

4. Select the **MCTRL** driver in the **Available Drivers** list, and then click the **Select** button.
5. When the **MCTRL** driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.

It is not necessary to install any other software on your PC to enable communication between the host and the device.

Configuring the Driver

After installing the MCTRL driver and opening Studio, you can configure the driver.

You configure a driver in two stages:

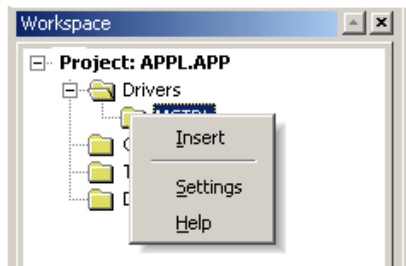
- Set the communication parameters (only one configuration for the whole driver)
- Define the communication tags by completing the communication tables or Driver Worksheets

The following sections provide instructions for setting the parameters and completing the Worksheets.

Setting the Communication Parameters

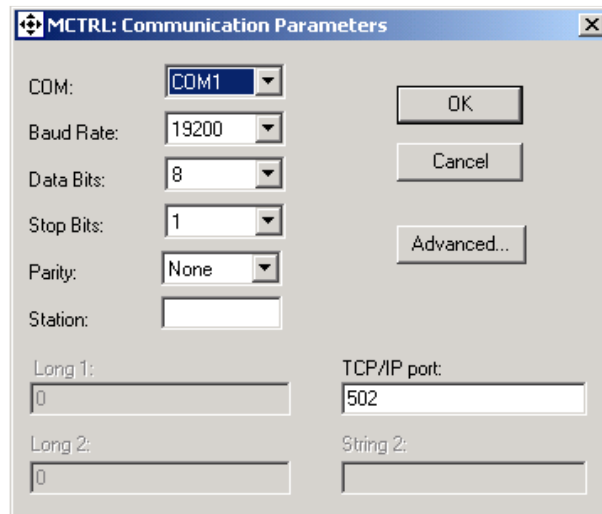
When you specify the communication parameters, they are valid for all Driver Worksheets configured in the system. Use the following steps to configure the communication parameters for the driver:

1. From the Studio application screen, click the *Comm* tab located below the *Workspace* pane.
2. From the *Workspace* pane, expand the *Drivers* folder.
3. Right-click on the *MCTRL* 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 *MCTRL: Communication Parameters* dialog displays as follows:

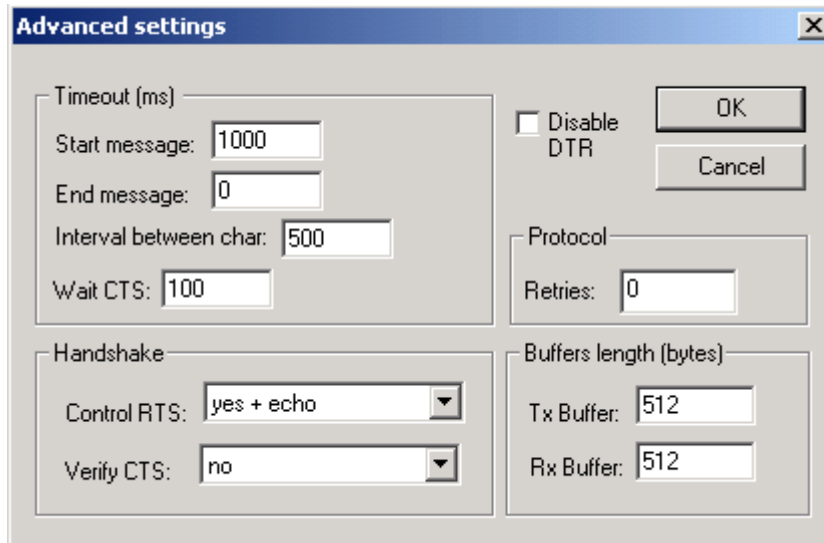


MCTRL: Communications Parameters Dialog Box

4. Configure the following parameters on the *Communications Parameters* dialog:

Parameter	Default Value	Valid Values	Description
COM	COM1	COM1 to COM8	PC serial port used to communicate with the device
Baud Rate	19200	110 to 57600bps	Data communication rate
Data Bits	8	5 to 8	Number of data bits used in the protocol
Stop Bits	1	1 or 2	Number of stop bits used in the protocol
Parity	None	Even, odd, none, space, or mark	Protocol parity
Station	0	0	Not used for the driver
TCP/IP port	(Blank)	(Blank) or 1 to 5000	Number of the TCP/IP port to use. If left blank, specifies a serial connection.

Note:
 You must configure the device using the same values you defined for the MCTRL driver in the *Communication Parameters* dialog. Click the **Advanced** button in the *Communication Parameters* dialog to open the *Advanced settings* dialog (as shown in the following figure).



Advanced Settings Dialog Box

5. Set the **Control RTS** (Request to Send) parameter, using the following table:

Parameter	Default Value	Valid Values	Description
Control RTS	No	<ul style="list-style-type: none"> • no(<i>Default</i>) • yes • yes+echo 	Define this parameter if the RTS (Request to Send) handshake signal is set before communication and if there is an echo in the communication.

Important:

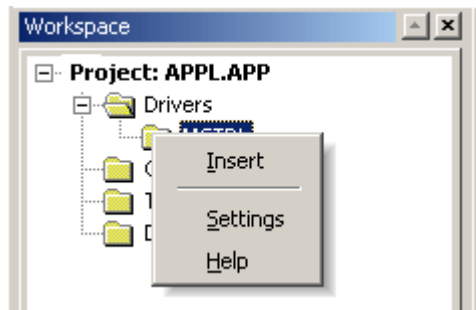
- Using the wrong settings on this field will prevent the driver from working, and cause **Timeout** error messages.
- Although you can configure other serial communication parameters from this dialog, **do not** change any of the default parameters at this time except **Control RTS**.
- For additional information about parameters on the *Advanced settings* dialog, refer to the Technical Reference Manual.

Configuring the Standard Driver Worksheet

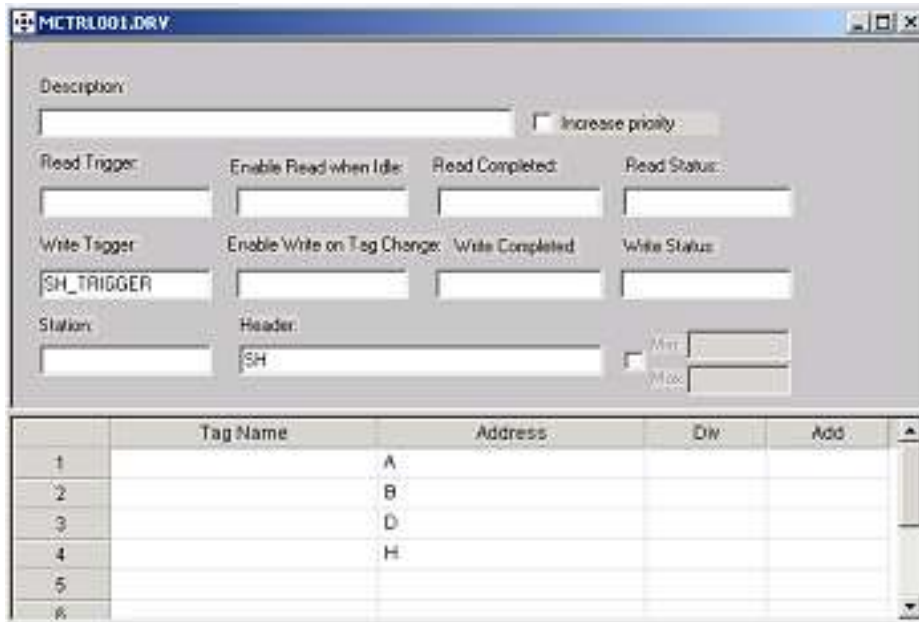
This section explains how to configure a Standard Driver Worksheet (or communication table) to define communication tags. You can configure multiple Driver Worksheets — each of which is divided into a Header and Body.

Use the following steps to create a new Standard Driver Worksheet:

1. From the Studio application screen, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the *MCTRL* subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



The <drivername>.drv dialog box displays (similar to the following figure).



MCTRL001.DRV Dialog Box

The fields on a Standard Driver Worksheet are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. Consequently, this document explains how to configure the **Station**, **Header**, and **Address** fields.

For detailed information about the configuring the standard fields refer to the Technical Reference Manual.

Configuring the Station Field

The **Station** field specifies the IP address of the target device. This field is not used for serial communications.

Configuring the Header Field

The **Header** field specifies the type of command to be sent to the device. The following table lists all of the valid headers for this driver and the operations you can use with each header.

Header	Description	Read Trigger	Write Trigger	Enable Read	Enable Write
AC	Acceleration	Allowed	Allowed	Allowed	Allowed
BG	Begin Move	NA	Allowed	NA	Allowed
CE	Configure Encoder	Allowed	Allowed	Allowed	Allowed
CN	Configure	NA	Allowed	NA	Allowed
DC	Deceleration	Allowed	Allowed	Allowed	Allowed

Header	Description	Read Trigger	Write Trigger	Enable Read	Enable Write
DE	Dual Encoder Position	Allowed	Allowed	Allowed	Allowed
ER	Error Limit	Allowed	Allowed	Allowed	Allowed
IP	Increment Position	Allowed	Allowed	Allowed	Allowed
JG	Jog Axis	Allowed	Allowed	Allowed	Allowed
KD	Derivative Constant	Allowed	Allowed	Allowed	Allowed
KI	Integrator	Allowed	Allowed	Allowed	Allowed
KP	Proportional Constant	Allowed	Allowed	Allowed	Allowed
MO	Motor Off	NA	Allowed	NA	Allowed
MT	Motor Type	Allowed	Allowed	Allowed	Allowed
OE	Off On Error	Allowed	Allowed	Allowed	Allowed
PA	Position Absolute	Allowed	Allowed	Allowed	Allowed
PR	Position Relative	Allowed	Allowed	Allowed	Allowed
RP	Reference Position	Allowed	NA	Allowed	NA
SH	Motor On	NA	Allowed	NA	Allowed
SP	Speed	Allowed	Allowed	Allowed	Allowed
ST	Stop Move	NA	Allowed	NA	Allowed
TD	Tell Dual Encoder	Allowed	NA	Allowed	NA
TL	Torque Limit	Allowed	Allowed	Allowed	Allowed
TP	Tell Position	Allowed	NA	Allowed	NA
TS	Tell Switches	Allowed	NA	Allowed	NA
TT	Tell Torque	Allowed	NA	Allowed	NA
TV	Tell Velocity	Allowed	NA	Allowed	NA
CMD	Individual Commands	*	Allowed	*	Allowed
VARF	Internal Float Variables	*	Allowed	*	Allowed
VARS	Internal String Variables	*	Allowed	*	Allowed

- Reads only one item per driver sheet

Configuring the Address Field


You can use the body of a Standard Driver Worksheet to associate each command with a specific axis or group of axes. If the command returns a value Studio will write that value to the tags in the tag column. If you do not specify an address, the driver sheet will not run.

The Address field must comply with the syntax described in the following table:

Header	Valid Addresses	Description	Tag Data Type
AC	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the acceleration value for specified axes.	Integer
BG	A,B,C,D,E,F,G,H,X,Y,Z,W	Starts motion on the specified axes.	Any
CE	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the encoder configuration for specified axes.	Integer
CN	M,N,O,P	Writes the limit, home, latch, and selective abort configuration.	Integer
DC	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the deceleration value for specified axes.	Integer
DE	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the auxiliary encoder positions for specified axes.	Integer
ER	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the error limit for specified axes.	Integer
IP	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the current position for specified axes.	Integer
JG	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the jog speed for specified axes.	Integer
KD	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the derivative constant for specified axes.	Real
KI	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the integral gain for specified axes.	Real
KP	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the proportional constant for specified axes.	Real
MO	A,B,C,D,E,F,G,H,X,Y,Z,W	Shuts off the control algorithm on the specified axes.	Any
MT	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the motor type for specified axes.	Real
OE	A,B,C,D,E,F,G,H,X,Y,Z,W	Enables/Disables the Off-On-Error function for specified axes.	Integer
PA	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the stop position for specified axes.	Integer
PR	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the incremental distance for specified axes.	Integer

Header	Valid Addresses	Description	Tag Data Type
RP	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the reference position for specified axes.	Integer
SH	A,B,C,D,E,F,G,H,X,Y,Z,W	Turns on the control algorithm on the specified axes.	Any
SP	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the slew speed for specified axes.	Integer
ST	A,B,C,D,E,F,G,H,X,Y,Z,W	Stops motion on the specified axes.	Any
TD	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the dual encoder position for specified axes.	Integer
TL	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads/Writes the torque limit for specified axes.	Real
TP	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the current position for specified axes.	Integer
TS	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the switch states for specified axes.	Integer
TT	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the current torque for specified axes.	Real
TV	A,B,C,D,E,F,G,H,X,Y,Z,W	Reads the current velocity for specified axes.	Integer
CMD	AB #	Aborts Motion (<i>Write Only</i>)	Integer
CMD	BN	Burns Parameters (<i>Write Only</i>)	None
CMD	BP	Burns Program (<i>Write Only</i>)	None
CMD	BV	Burns Variables (<i>Write Only</i>)	None
CMD	CB #	Clears Output Bit (<i>Write Only</i>)	Integer
CMD	SB #	Sets Output Bit (<i>Write Only</i>)	Integer
CMD	MG_MTn	Gets Motor Type (<i>Read Only</i>)	Integer
CMD	MG_SCn	Gets Stop Code (<i>Read Only</i>)	Integer
CMD	MG@AN[n]	Gets Analog Input Value (<i>Read Only</i>)	Real
CMD	MG@IN[n]	Gets Digital Input Value (<i>Read Only</i>)	Integer

Header	Valid Addresses	Description	Tag Data Type
CMD	MG@OUT[n]	Gets Digital Output Value (<i>Read Only</i>)	Integer
CMD	RS	Resets (<i>Write Only</i>)	None
CMD	TM?	Gets Sample Time (<i>Read Only</i>)	Integer
CMD	TM #	Sets Sample Time (<i>Write Only</i>)	Integer
CMD	XQm,n	Executes Program (<i>Write Only</i>)	None
VARF	{Variable Name}	Allows internal variables to be set (or declared) and read.	Real
VARS	{Variable Name}	Allows internal variables to be set (or declared) and read.	String

 **Notes:**

- In this Motion Control protocol:
 - A, M, and X are equivalent
 - B, N and Y are equivalent
 - C, O and Z are equivalent
 - D, P and W are equivalent
- The Maximum size for a String Variable is 6 characters long.
- For further explanation about these commands, please see the Command Reference Guide for your specific Motion Control device.

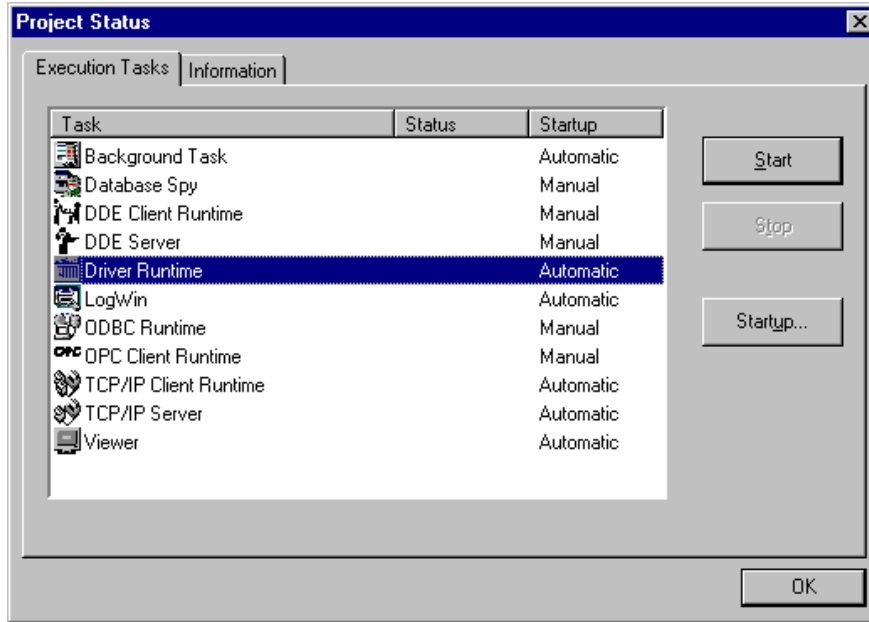
Executing the Driver

After you add the MCTRL driver to a project, Studio sets the project to automatically execute the driver 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.



Project Status Dialog Box

2. Verify that the **Driver Runtime Startup** task is set to **Automatic**.
 - If the setting is correct, click **OK** to close the dialog box.
 - If the **Driver runtime** task is not set to **Automatic**, select the **Driver Runtime** line and when the **Startup** button becomes active, click the button (a toggle) to set the **Startup** mode to **Automatic**.
3. Start the application to run the driver.

Troubleshooting

If the MCTRL driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields will receive an error code. You can use this error code to identify the type of failure that occurred.

The following table describes the error codes for this driver:

Error Code	Description	Possible Causes
0	OK	Command sent successfully.
3	Invalid Station	The IP address in the station field is invalid.
23	Command Error	Errors while sending command, execute TC to receive error code.
26	Connection Error	Error occurred while connecting to the target device.
27	Read Violation	Header does not support any Read functions.
-15	Time Out	Timeout waiting for a response from the controller.

⇒ **Tip:**

You can verify communication status using the Studio environment *Output* window or the *LogWin* module. To establish an event log for **Field Read Commands**, **Field Write Commands**, and **Serial Communication**, right-click on the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing under a Windows CE target, you can enable the log at the unit (**Tools** → **Logwin**) and verify the `ce1og.txt` file created at the target unit.

If you are unable to establish communication with the device, you must first try to establish communication between the device's Programming Software and the device. Quite frequently, communication is not possible because you have a hardware or cable problem, or a device configuration error. After you successfully establish communication between the device's Programming Software and the device, you can retest the supervisory driver.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of Changes
A	1.00	Bryan Morgan	16-May-02	First driver version
B	1.00	Bryan Morgan	12-July-02	Expanded function list
C	1.00	K. C. Francis	16-July-02	Changes to layout and content of the document
D	1.01	Bryan Morgan	7-Aug.-02	Added Ethernet capability
E	1.02	Bryan Morgan	21-Aug.-02	Correct document error foe VARF header and correct Read Error message number
F	1.03	Lourenço Teodoro	09-Mar.-04	Fixed bug when not all axis are present. The Read function was always reading all the axis.
G	1.03	Andre Bastos	01-10-09	Updated the content of the documentat