AVEVA™ Communication Drivers Pack – Texas Instruments – TI500 Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868893

Contact Information

AVEVA Group Limited
High Cross
Madingley Road
Cambridge
CB3 0HB. UK

https://sw.aveva.com/

For information on how to contact sales and customer training, see https://sw.aveva.com/contact.

For information on how to contact technical support, see https://sw.aveva.com/support.

To access the AVEVA Knowledge and Support center, visit https://softwaresupport.aveva.com.

Contents

Chapter 1

Getting Started with TI500 Communication Driver

This document describes the technical specifications and configuration options for the Texas Instruments TI500 Communication Driver.

About the TI500 Communication Driver

The TI500 Communication Driver can communicate with Simatic/TI 500 series devices as well as CTI 2500 series devices. It supports the following protocols:

- NITP (Non-Intelligent Terminal Protocol)
- TBP (Transport Binary Protocol)
- CAMP (Common ASCII Messaging Protocol)

Note: This Communication Driver is hosted by the OI Server Manager, a Microsoft Management Console (MMC) snap-in, which is a part of the Operations Control Management Console (OCMC) suite of utilities. Many high-level functions and user-interface elements of the OI Server Manager are universal to all Communication Drivers, and only the documentation for the OI Server Manager contains descriptions of those universal functions/UI elements. Therefore, reading the documentation for both the MMC and the OI Server Manager is critical to understanding this user's guide. To read the documentation about the MMC and OI Server Manager, right-click the OI Server Manager icon and select the Help menu. Both the MMC Help and the Communication Drivers Pack Help are displayed.

Supported Hardware and Software

The TI500 Communication Driver connects to the following 500 Series PLCs from Texas Instruments (Siemens):

- TI545
- TI565
- TI575
- CTI2500

For RS232/ Ethernet communication, CTI2500 - C100 PLCs can use CAMP, TBP, or NITP protocol. CAMP protocol supports TCP and UDP. For UDP, use CTI 2572-B module connecting to CTI2500 PLCs. Select the specific protocol in the communication settings.

To program CTI2500 - C100 PLCs, you need PLC WorkShop Suite for Siemens 505. For more information, see

Conformance

The following hardware and software was used for conformance testing of this Communication Driver.

Device: CTI2500-C100 with CTI 2572-B module to enable UDP

Configuration 1:

- Serial/Ethernet: Serial (NITP)
- Databits: 7
- Parity: Odd

Configuration 2:

- Serial/Ethernet: Serial (TBP)
- Databits: 8
- Parity: None

Configuration 3:

- Serial/Ethernet: TCP

Configuration 4:

- Serial/Ethernet: UDP

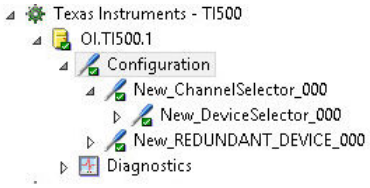
Chapter 2

Configuring the TI500 Communication Driver

Working with the TI500 Communication Driver Configuration

Each server instance has its own hierarchy of objects, and each object has parameters that you need to configure in order to establish communication between the Communication Driver and individual devices on the network.

You can view a TI500 server instance's configuration hierarchy under its **Configuration** node.



This section only describes how to configure object parameters for a TI500 server instance. For more general information about adding and configuring objects, see "Configuring Your Communication Driver" in the Communication Drivers Pack Help.

Setting Up the TI500 Communication Driver for the First Time

If you are setting up an Communication Driver for the first time, perform the following tasks in the order listed:

1. Locate the Communication Driver in the Operations Control Management Console (OCMC). In the OI Server Manager tree, under the Local node, the Communication Driver base instance name is OI.TI500.1.

2. Configure the global parameters. See "Configuring Global Parameters"in the Communication Drivers Pack help.
3. Add one or more channel selector connections.
4. Add one or more device selector connections.
5. Add one or more device groups.
6. Add one or more device items.
7. Activate the Communication Driver. See "Activating/Deactivating the OI Server"in the Communication Drivers Pack help.
8. Troubleshoot any problems. See Troubleshooting the TI500 Communication Driver.

Setting the Station ID of a Device

Set the station ID for a selected device so that the Communication Driver can identify and communicate with it on the network.

Syntax

A station ID is required only if you select **TCP** or **UDP** option under the **Serial/Ethernet** drop down list in the **ChannelSelector** screen. For **Serial (TBP)** and **Serial (NITP)** options, the **Station** field in the **DeviceSelector** can be left blank.

Note: If you are upgrading from an old version of the Communication Driver which has NITP protocol selected with Serial Encapsulation as TCP/IP, the **Serial/Ethernet** selection is updated with TCP in the **ChannelSelector** screen and the **Station** field in the **DeviceSelector** screen is updated with the IP address and Port number.

The station ID for a target device must use the following syntax:

```
<IP address>:<Port Number>
```

IP address

The IP address of the target device.

Port Number

The port number of the PLC program running on the target device. The default port is 1505, but you can use the PLC programming software to change it if necessary.

Examples:

192.168.0.5:1505

10.13.63.90:1505

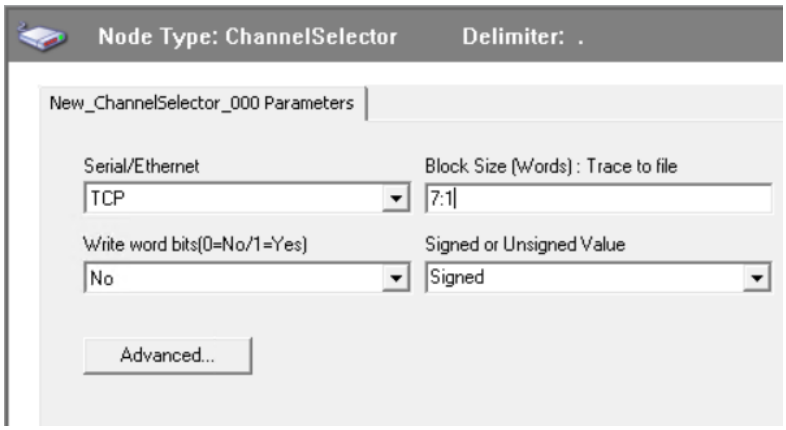
Adding and Configuring a Channel Selector Object

The server-specific configuration portion of the TI500 Communication Driver hierarchy tree under the OI Server Manager starts at the Channel Selector object. This object lets you set server parameters for communication with agents (devices) in the hierarchy tree.

Adding Channel Selector Objects

To add a Channel Selector object to your TI500 hierarchy

- In the console tree, right-click **Configuration** and then click **Add Channel Selector Connection**. The "New_ChannelSelector_000" object appears in the hierarchy.



Edit the object name to appropriately describe components of your specific hardware environment. If you do not rename the object at this time, a numeric sequencing system is applied. You can rename the hierarchy entry later.

Configuring Channel Selector Objects

To configure a Channel Selector object

1. In the **Serial/Ethernet** dropdown list, select any one of the options:
   - **Serial (TBP)**: for direct serial communication with the device using Transparent Byte Protocol (TBP).
   - **Serial (NITP)**: for direct serial communication with the device using Non-Intelligent Terminal Protocol (NITP).
   - **TCP**: to communicate over a TCP Ethernet connection.
   - **UDP**: to communicate over a UDP Ethernet connection.

     Both TCP and UDP use CAMP protocol for communication.
2. In the **Block Size (Words) Trace to file**:

   Default is 0 or disabled. If the you want to enable this field, change value to 1. For example: The field **Block Size (Words) Trace to file** with the value 7, 1 can be used to enable additional logs saved in an external file. If this field is enabled and if Protocol Analyzer is enabled on the output window, the location of this external file (.txt) will be indicated on the driver startup in the output window messages. Advanced driver task logs will be saved to this file during runtime. Each runtime of the driver will generate a new log file.
3. In the **Write word bits (0=No/1=Yes)** field**:**
   - Enter 1, to allow writing to the individual bits of a Word.
   - Enter 0, to not allow writing to individual bits of a Word.
4. In the **Signed or Unsigned Value** dropdown list, select the type of the register value.

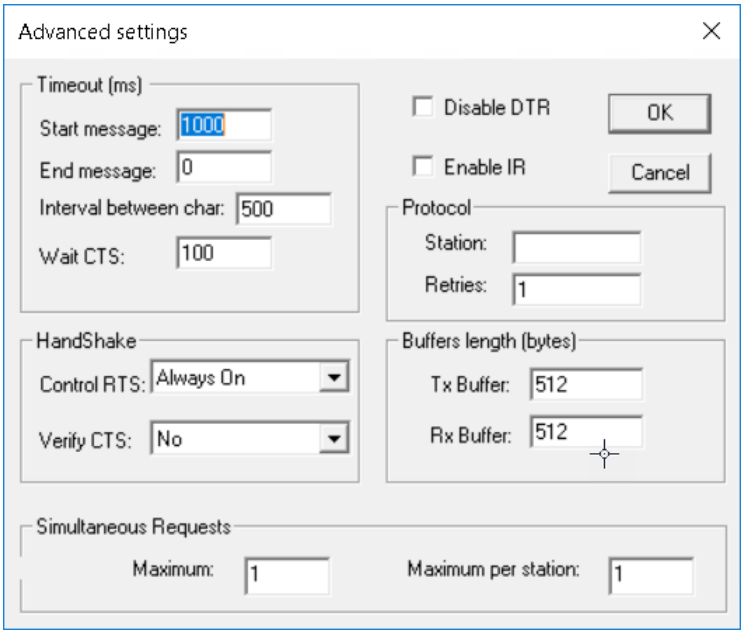
5. In the **Serial/Ethernet** dropdown list if you select **Serial (TBP)** or **Serial (NITP),** the **Serial Port** section appears. Under this section,you can configure the direct serial communication settings:
   - In the **COM** dropdown list, select the COM port to which the target device is connected.
   - In the **Baud Rate**, **Data Bits**, **Stop Bits**, and **Parity** dropdown lists, configure the serial communication settings to match the settings that have already been configured on the target device.
6. Use **Advanced** option to modify additional communication settings. For more information see Advanced Settings.

Note: If you are adding more than one device selectors to the same channel, except the **Serial/Ethernet** field you can leave the other fields blank.

Advanced Settings

This option allows you to modify additional communication settings such as timeouts, retries, and buffer sizes.These settings should be modified only during an unexpected behavior of Communication Driver. In all other cases, keep the default settings unchanged.

Note: The **Timeout**, **Simultaneous Requests**, and **Retries** fields are editable. We recommend retaining the default values for all other fields.



The following parameters are configurable in the TI500 Communication Driver:

Parameter	Default	Description
Timeout: Start message	1000 ms	Specify the timeout for the message start.
Protocol: Retries	1	Enter value to specify the number of attempts to execute the same communication before considering a communication error for this command.

Simultaneous Requests: Maximum	1	Specify the maximum number of requests that can be sent simultaneously to all connected devices.
Simultaneous Requests: Maximum per station	1	Specify the maximum number of requests that may be sent simultaneously to a single device.

The following parameters are not applicable in the TI500 Communication Driver, or are pre-set and are not configurable in the TI500 Communication Driver. Descriptions are provided here for information purposes:

Parameter	**Description**
Timeout: End message	Reserved. Do not modify.
Timeout: Interval between char	Specifies the timeout between each character.
Timeout: Wait CTS	Specifies the timeout for the Clear to Send wait.
Handshake: Control RTS	Specifies whether to use the Request to Send control.
Handshake: Verify CTS	Specifies whether to use the Clear to Send verification type.
Disable DTR	When disabled, no DTR signal is sent before starting a communication.
Enable IR	Not applicable, as the Wonderware System Platform does not support Windows Embedded.
Protocol: Station	Used to specify a slave address where required.
Buffers length: Tx Buffer	Reserved. Do not modify.
Buffers length: Rx Buffer	Reserved. Do not modify.

For more information about these settings, see ""Advanced Settings" n the Communication Drivers Pack Help.

Adding and Configuring Device Selector Objects

The TI500 Communication Driver can connect to PLCs. These connections are modeled in the hierarchy by means of Device Selector objects, each of which models the end-point of the communications path.

From the **ChannelSelector** branch of the Communication Driver hierarchy, create the new **DeviceSelector** object.

To add a Device Selector connection to your TI500 hierarchy

1. In the console tree, right-click the **ChannelSelector** object, and then click **Add DeviceSelector Connection**. The **New_DI_000** object and associated **Parameters** configuration view appear.
2. Rename the object as needed to reflect the connection.
3. Configure the **Station** field.

Note: This field is mandatory, if you select **TCP** or **UDP** option under the **Serial/Ethernet** drop down list in the **Channel Selector** screen. You should leave this field blank if you select **Serial (NITP)** or **Serial (TBP)** option under the **Serial/Ethernet** drop down list in the **Channel Selector** screen.

For **TCP** or **UDP** options, use the following syntax:
`<IP Address>:<Port Number>`

Parameter	Default Value	Description
IP Address	none	The IP network address of the target device. **Required**.
Port Number	none	The port of the CTI or TI controller, supports up to three TCP connections. You can connect using the TCP port 4450 or 1505.

Example: 198.2.65.1:1505

For more information see Setting Station ID of a Device.

4. Configure **Device Groups** and **Device Items**.

Device Group Definitions

Use the **Device Groups** configuration view, to create, add, delete, and define device groups. You can also configure default update intervals for the objects and edit update intervals in this dialog box. To open the **Device Groups** dialog box, in the **Device Selector** configuration editor, click the **Device Groups** tab.

Note: When you select another part of the Communication Driver tree hierarchy, you are prompted to save the modifications to the configuration set.

To create or add device groups

1. Right-click anywhere in the table, and then click **Add**. A device group is added with a default name and update interval.
2. Enter a unique name up to 32 characters long for the device group.

To delete device groups

1. Right-click the device group to be deleted, and then click **Delete**.
2. Read the warning, and then click **Yes**.

To edit device groups

Use the **Edit** option from the **Device Groups** tab only for configuring the Communication Driver's unsolicited message handling.

To configure default update intervals

To configure a default update interval for the object, right-click in the **Device Groups** box and then click **Config Default Update Interval**.

To edit update intervals

To edit the update interval for an object, double-click its value in the **Update Interval** column and make the edits.

or

Right-click its value in the **Update Interval** column and then click **Modify Update** Interval.

The update interval is the frequency, in milliseconds, that the TI500 Communication Driver acquires data from the topics associated with that device group.

Different topics can be polled at different rates from a PLC by defining multiple device group names for the same PLC and setting a different update interval for each device group.

Device Item Definitions

The device item name is an “alias” or a label for the data in the device. It is an alternative name for the item reference, and can be used instead of the item reference when you create the client application. Device item configuration is optional, but is strongly recommended.

To create or add device items

1. Right-click anywhere in the table, and then click **Add**.
2. In the **Name** column, type a unique item name. The maximum is 32 characters.
3. In the corresponding line, double-click the **Item Reference** column and enter the correlated item reference for the name you created.

To rename device items

Right-click the device item to be renamed and click **Rename**. Make the changes.

To delete device items

Right-click the device item to be deleted from the list and click **Delete**.

To clear all device items

Right-click in the **Device Items** box and click **Clear All**. All the device items listed are cleared after you confirm their deletion.

NOTE: You can import a .csv file containing your item definitions to help streamline configuration. See "Exporting and Importing CSV Files" in the Communication Drivers Pack Help.

AVEVA

Chapter 3

TI500 Communication Driver Reference

Use item references to access data stored in memory registers in connected devices, as well as to access standard system items in the Communication Driver itself.

This section only describes the item reference syntax and options for the TI500 server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the Communication Drivers Pack Help.

Item Reference Syntax

Item references in this Communication Driver use the following syntax.

- For digital registers (i.e., register types X, Y, C), use the following syntax:

  `<register type><address>`

  The following syntax diagram shows all of the possible options:

  `{ X | Y | C }address`

  For example: X1

- For analog registers (i.e., register types V, K, TCP, TCC, WX, WY), use either of the following syntaxes:

  `<register type><address>.[bit]`

  For example: V1.2

  `<register type><address><data type>`

  For example: V1D

  The following syntax diagram shows all of the possible options:

  `{ V | K | TCP | TCC | WX | WY }address{ | .bit | F | .Step | D | S | U | ST<length> | STS<length> }`

Register type

The type of memory register on the target device:

Type	Description
X	Discrete Input
Y	Discrete Output
C	Control Relay
V	Variable Memory
K	Constant Memory
TCP	Timer/ Count Preset Memory
TCC	Timer/ Count Count Memory
WX	Word Input Memory
WY	Word Output Memory
STW	System Status Memory
DCC	Drum Current Count Memory
DCP	Drum Count Preset Memory
DSP	Drum Step Preset Memory

Address

The address of the memory register on the target device.

Bit

The bit number (1 to 16) to be accessed in the memory register.

This parameter is optional; if no bit is specified, the entire 16-bit Word is used by default.

Data type

The data type of the memory register: The data type is specified as a suffix in the item syntax. The TI500 Communication Driver supports the following data types.

DataType	Suffix	Description
Bit(Boolean)	.Bit	Single bit. Bit 1 is the most significant bit. Bit 16 is the least significant bit.
Float	F	Floating point or real number (32-bit).
DWord	D	Unsigned, 32-bit decimal value. Bit 1 is the most significant bit and Bit 32 is the least significant bit.

Integer	S	Signed 16-bit value. **Note:** If the S data type is is used, then registers can receive signed values even if the channel selector property is set to unsigned.
Integer	U	Unsigned 16-bit value. **Note:** If the U data type is used, then registers can receive unsigned values even if the channel selector property is set to signed.
Integer	DS	Signed 32-bit value. **Note:** If the DS data type is is used, then registers can receive signed values even if the channel selector property is set to unsigned.
Integer	DU	Unsigned 32-bit value. **Note:** If the DU data type is used, then registers can receive unsigned values even if the channel selector property is set to signed.
String	ST<length>	String. <length> denotes the number of characters in the string. Every word in PLC contains two characters; the first byte in the word is the first character, and the second byte is the second character.
String	STS<length>	String with byte swap. <length> denotes the number of characters in the string. Every word in PLC contains two characters. Byte swap means that the first byte in the word is the second character, and the second byte is the first character.

This parameter is optional; if no type is specified, it is considered to be Word (i.e., signed, 16-bit) by default.

Suffix data types

- All registers support the suffix ST<length> and STS<length> except X, Y, and C and DCP.
- The maximum string length that can be specified depends on the register it is used with.
- The length is a required parameter when using ST or STS.
- In the **Channel Selector** screen, if you select the **Serial/Ethernet** field as **Serial (NITP)** or **Serial (TBP)**, the String data types ST and STS are not supported.
- The DCP register requires a .Step suffix. It does not support any other suffixes.

Address Descriptions

The address descriptions consist of the register type, its item name and the allowable range of values, the default data type, allowable suffixes (if any), and allowable access methods.

Register Type	Item Name	Data Type	Access
Discrete Input	X	Bit	Read Only
Discrete Output	Y	Bit	Read/Write
Control Relay	C	Bit	Read/Write
Constant Memory	V	Word	Read/Write
Variable Memory	K	Word	Read/Write
Timer/Count Preset Memory	TCP	Word	Read/Write
Timer/Count Count Memory	TCC	Word	Read/Write
Word Input Memory	WX	Word	Read/Write
Word Output Memory	WY	Word	Read/Write
System Status	STW	Word	Read/Write
Drum Step Preset	DSP	Word	Read/Write
Drum Count Preset	DCP	Bit	Read/Write
Drum Current Count	DCC	Word	Read Only

Note: In the **Channel Selector** screen, if you select the **Serial/Ethernet** field as **Serial (NITP)** or **Serial (TBP),** the Drum memory registers DCC, DCP, and DSC are not supported.

Supported register types

The following registers are implemented in the Communication Driver which are PID Control and Analog Alarm Data Elements.

Register Type	Item Name
Loop Process Variable	LPV
Loop Setpoint	LSP
Loop Output	LMN
Loop Bias	LMX
Loop Error	LERR

Loop Gain	LKC
Loop Rate	LTD
Loop Reset	LTI
Loop V-flags	LVF
RAMP/SOAK flags	LRSF
Analog Alarm Process Variable	APV
Analog Alarm Setpoint	ASP
Analog Alarm flags	AVF
Loop Process Variable High Limit	LPVH
Loop Process Variable Low Limit	LPVL
Analog Alarm Process Variable High Limit	APVL
Analog Alarm Process Variable Low Limit	APVH
Loop Sample Rate (in seconds)	LTS
Analog Alarm Sample Rate (in seconds)	ATS
Loop High Alarm Limit	LHA
Loop Low Alarm Limit	LLA
Loop Orange Deviation Alarm Limit	LODA
Loop Yellow Deviation Alarm Limit	LYDA
Loop Setpoint High Limit	LSPH
Loop Setpoint Low Limit	LSPL
Most-significant word of Loop C-flags	LCFH
Least-significant word of Loop C-flags	LCFL
Loop C flags (combination of LCFH and LCFL)	LCF
Loop High-High Alarm Limit	LHHA
Loop Low-Low Alarm Limit	LLLA
Loop Rate-of-Change Alarm Limit (in Engineering Units per minute)	LRCA
Loop Alarm Deadband	LADB
Analog Alarm High Alarm Limit	AHA
Analog Alarm Low Alarm Limit	ALA

Analog Alarm Orange Deviation Alarm Limit	AODA
Analog Alarm Yellow Deviation Alarm Limit	AYDA
Analog Alarm Setpoint High Limit	ASPH
Analog Alarm Setpoint Low Limit	ASPL
Most-significant word of Analog Alarm C-flags	ACFH
Least-significant word of Analog Alarm C-flags	ACFL
Analog Alarm C flags (combination of ACFH and ACFL)	ACF
Analog Alarm High-High Alarm Limit	AHHA
Analog Alarm Low-Low Alarm Limit	ALLA
Analog Alarm Rate-of-Change Alarm Limit (in Engineering Units per minute)	ARCA
Analog Alarm Alarm Deadband	AADB
Analog Alarm Error	AERR
Loop Derivative Gain-limiting coefficient	LDK
Loop RAMP/ SOAK Step Number	LRSN
Loop Alarm/ Alarm Acknowledge flags	LACK
Analog Alarm/ Alarm Acknowledge flags	AACK
Loop Peak Elapsed Time Value Represents the elapsed time from when the process is scheduled until it completes execution (TI545,TI555,TI575).	LPET
Analog Alarm Peak Elapsed Time Value Represents the elapsed time from when the process is scheduled until it completes execution (TI545 , TI555 , TI575).	APET
SF PGM Peak Elapsed Time Value Represents the elapsed time from when the process is scheduled until it completes execution(TI545, TI555,TI575).	PPET

Note: In order to use these registers, you need to configure them in the PLC using the PID Loop Directory and Analog Alarm Directory menus.

Examples of Item References

These are examples of valid item references for this Communication Driver. For more information about the referenced addresses, see the documentation provided by the manufacturer for your device.

Register Type	Address on the Device	Item Reference
Discrete Input	X01000	X1000
Discrete Output	Y00007	Y7
Control relay	C01400	C1400
Variable Memory	V00001(Float format)	V1F
		V1.15
	V00100(DWord format)	V100D
		V100.15
	V00001 (Signed 16-bit value)	V1S
	V00001 (Unsigned 16-bit value)	V1U
	V00001 (Signed 32-bit value)	V1DS
	V00001 (Unsigned 32-bit value)	V1DU
Constant Memory	K00007	K7 K7.7
	K00026	K26
		K26.6
Timer/Count Preset Memory	TCP00010 TCP00053	TCP10 TCP53
Timer/Count Count Memory	TCC00017 TCC00020	TCC17 TCC20
Word Input Memory	WX00000 (DWord format) WX00000 (Float format)	WX0D WX0F
Word Output Memory	WY00007 WY00021	WY7 WY21.1
System Status	STW00020 -	STW20
Drum Current Count	DCC00001	DCC1
Drum Step Preset	DSP00023	DSP23
Drum Count Preset	DCP00003.3 (step= 3)	DCP3.3
Drum Step Counter	DSC00002 (float)	DSC2F
String With Swap	K00100-K00109 (String with swap)	K100STS20

String	V00100-V00101 (String)	V100ST3

Similarly for other register types and memory locations, you can derive the item reference from the address on the device. For information regarding PID and Alarms, see Supported register types section under Address Descriptions.

Chapter 4

Troubleshooting the TI500 Communication Driver

TI500 Communication Driver Error Codes

The following tables describe the additional error codes that you might receive when poll/poke requests and operations fail.

Code	Description	Possible Causes	Solution
1	Protocol Error	Communication Driver settings do not match the PLC settings.	Check the PLC settings, and then configure the Communication Driver settings to match.
2	Invalid Command	Communication Driver settings do not match the PLC settings.	Check the PLC settings, and then configure the Communication Driver settings to match.
3	Invalid Response	Communication Driver settings do not match the PLC settings.	Check the PLC settings, and then configure the Communication Driver settings to match.
4	Invalid Block Size	The PLC model does not accept the number of bytes requested or sent.	Adjust the Max Block Size (in the channel settings).
5	Invalid Header	One or more of the specified item references does not use the required syntax.	Verify the item references according to the syntax provided in this documentation, and then correct them if necessary.
6	Invalid Address	One or more of the specified item references does not use the required syntax.	Verify the item references according to the syntax provided in this documentation, and then correct them if necessary.
9	Word bit write is not enabled	Attempted to write to a word bit without enabling it in the channel settings.	Select the Write word bits option(in the channel settings).

14	Busy Error	• Disconnected cables. • Device is turned off, or in Stop or error mode. • Station number is wrong.	• Check the cable wiring. • Check the device state. It must be RUN. • Check the station number. • Check the configuration.
15	Invalid Entry	Invalid entry or entries for channel settings.	Check Write word bits or Protocol (in the channel settings) and make sure they are each set to either 0 or 1, according to the descriptions provided in this documentation.
16	Configured block size is not supported by PLC	The Communication Driver is requesting more registers than what is supported by the PLC model.	Set the Max Block Size (in the channel settings) to a value that is supported by the PLC model. The default value is 7.
17	Invalid entry for block size	The specified maximum block size is invalid.	Set Max Block Size (in the channel settings) to an integer value from 1 to 15.
18	Invalid block size	The specified maximum block size is invalid.	Set Max Block Size(in the channel settings) to an integer value from 1 to 15.
19	Failed to connect to the PLC	Connection to the PLC failed.	Check PLC Settings and configure Driver Settings to match with it.
20	Unsupported TI500 type	A tag with invalid type has been inserted into the Address field	• Check if the address is supported by the driver. • Check the address is valid on the PLC
21	Invalid Message ID.	Invalid entry or entries in the driver settings or the driver sheet.	• Check the driver settings and the address that is failing. • Check the PLC connectivity and that the settings and entries on the driver match with it. • Close and reopen the project and start runtime . • If issue still persists contact technical support.
0	OK	Communicating without error.	None required.
-15	Time out waiting for message to start	• Disconnected cables. • PLC is turned off, in stop mode,or in error mode.	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number.

		• Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/ CTS configuration (for serial communication).	• Increase the time out in the advanced settings of the TI500 Communication Driver. • Check the RTS/ CTS configuration (for serial communication).

List of Error Codes for the protocol sent by the PLC (NITP and CAMP)

This group of error codes is returned by the remote PLC when an error is encountered processing a task code.

Error Code	Description
129 [01]	Reset Current Transaction
130 [02]	Address out of Range (other than ladder logic)
131 [03]	Requested data not found
132 [04]	Illegal Task Code Request
133 [05]	Request exceeds available memory
134 [06]	Diagnostic fail on power up
135 [07]	Fatal error detected
136 [08]	Keylock/password protection error
137 [09]	Incorrect amount of data sent with request
138 [0A]	Illegal request in current operational mode
139 [0B]	Network was not deleted
140 [0C]	Attempted write operation did not verify
141 [0D]	Illegal number of ASCII characters received
142 [0E]	Illegal request when running from EEPROM or flash
143 [0F]	Data not inserted
144 [10]	Data not written
145 [11]	Illegal data sent with command

146 [12]	Invalid operation with NIM local/remote mode (obsolete)
146 [12]	The store and forward buffer is busy
148 [14]	No response from special function module
149 [15]	Illegal instruction found in program memory (may include memory address)
150 [16]	Attempted to write to protected variable (e.g. TCC, TCP)
151 [17]	No response from PLC (e.g. single scan not performed)
152 [18]	Requested memory size exceeds total available memory
153 [19]	Requested memory size is not multiple of block allocation size
154 [1A]	Requested memory size is less than minimum defined value
155 [1B]	Requested memory size is larger than maximum defined value
156 [1C]	PLC busy – cannot complete requested operation
157 [1D]	Communications error in HOLD mode – Transition to Run not allowed
158 [1E]	Port Lockout is active
159 [1F]	Attempting to delete active program via reconfiguration
160 [20]	Program load in progress or invalidated
161 [21]	I/O configuration error – too many points
162 [22]	I/O configuration error – attempt to assign Output point to multiple applications
191 [3F]	Bus error detected
192 [40]	Operating system error detected
193 [41]	Invalid control block type
194 [42]	Control block number out of range
195 [43]	Control block does not exist
196 [44]	Control Block already exists
197 [46]	Offset out of range

198 [47]	Arithmetic error detected while writing Loop or Loop Alarm parameters
199 [48]	Invalid SF program type
200 [49]	Instruction number or RAMP/SOAK step number out of range
201 [4A]	Attempt to access an integer variable as a real
202 [4B]	Attempt to access a real variable as an integer
203 [4C]	Trask code buffer overflow – too much data requested
204 [2D]	Control block size error (cannot exceed 32767 bytes)
208 [50]	Task code request buffer too large
209 [51]	Invalid SF statement size
210 [52]	Invalid return value
211 [53]	Attempt to execute a cyclic statement in a non-cyclic SF program
212 [54]	Control block is disabled
213 [55]	Control block is not disabled
214 [56]	Attempt to perform a FSTR_OUT SF statement on an empty FIFO
215 [57]	Attempt to perform a FSTR_INT SF statement on a full FIFO
216 [58]	Stack overflow while evaluating a MATH, IF-THEN, or IMATH statement
217 [59]	Maximum SF subroutine nesting level exceeded (maximum = 4)
218 [1A]	Arithmetic Overflow
219 [5B]	Invalid operator in and IF, MATH, or IMATH expression
220 [5C]	S memory overflow
221 [5D]	Attempt to divide by 0
224 [60]	Invalid data type code
225 [61]	RAMP/SOAK step type mismatch

List of Error Codes for the protocol sent by the PLC (CAMP)

This group of error codes is returned by the remote PLC when an error is encountered processing a task code.

Error Code	Description
30 [6E]	NITP Protocol error
35 [73]	Bad or missing delimiter
36 [74]	Bad clock check character
37 [75]	Invalid Type
38 [76]	Invalid Data Character
39 [77]	Odd number of characters
40 [78]	Invalid device code
48 [80]	Invalid error character
49 [81]	No words to write
50 [82]	Invalid word count
51 [83]	Memory Address = 0
52 [84]	Write Unsuccessful
53 [85]	Invalid command code
63 [8F]	Invalid number of words
64 [90]	Unsupported address class or device class
65 [91]	Request Too Large
67 [93]	CAMP maximum response exceeded
68 [94]	Maximum number of task codes per message
69 [95]	Invalid Task Code Character Count
92 [AC]	Memory Read Error
93 [AD]	Memory Write Error
119 [C7]	Message Queue Fill