AVEVA™ Communication Drivers Pack – Beckhoff – TWINCAT Driver

User Guide

Publication date: Tuesday, May 9, 2023

Publication ID: 868900

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 TWINCAT Communication Driver

Introduction to the TWINCAT Communication Driver

These are the technical specifications for Beckhoff TWINCAT Communication Driver.

Requirements

To communicate with TwinCAT PLCs and runtimes, you must have the TwinCAT Automated Device Specification (ADS) software installed and configured on the same computer where this Communication Driver is installed. The ADS software allows the computer to present itself as a TwinCAT node on the network, and this Communication Driver communicates with other nodes through it. The ADS software is installed as part of the full TwinCAT programming software, so if you already have the programming software installed on the computer, there is nothing more you need to do. Otherwise, you need to install and configure the ADS software separately. For more information, see Installing and Configuring the TwinCAT ADS Software at the end of this document.

Conformance

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

- Driver Configuration: PLC program Machine.pro
- Equipment:
  - Beckhoff TwinCAT PLC
  - Beckhoff CX1000
  - Beckhoff CX1020
  - Beckhoff BC9000
  - Beckhoff CP6607
- Cable: Ethernet

Supported Hardware and Software

Once you have installed and configured the ADS software, this Communication Driver should be able to communicate over TCP/IP with any Beckhoff device that is compatible with the version of the ADS software that you installed. The following devices were used in conformance tests with the TC3 ADS software (product number TC1000):

- Beckhoff TwinCAT PLC/IO on Windows
- CX1000 and CX1020 with Windows Embedded Compact 5.0 and 6.0
- PLC BC9000, BX9000 and CP6607 with Windows Embedded Compact 5.0 for ARMV4i processor

The TC3 ADS software should also be compatible with CX1010, CX1030, CX2000, CX5000, CX5100, CX8000, CX9000, and any other device that hosts a TwinCAT 2 or TwinCAT 3 runtime.

If you are unsure whether your specific device is supported, please contact your Beckhoff support representative and confirm that your device is compatible with the ADS software

Chapter 2

Configuring the TWINCAT Communication Driver

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 the configuration hierarchy of a TWINCAT server instance under its Configuration node.

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

This section include:

Adding and Configuring a Channel Selector Object

The server-specific configuration of the TWINCAT 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 ChannelSelector connection to your TWINCAT hierarchy

1. In the console tree, right-click **Configuration** and then click **Add ChannelSelector Connection**. The **New_ChannelSelector_000** object appears in the hierarchy.
2. 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.

   The **New_ChannelSelector_000 Parameters** view is displayed.

Configuring Channel Selector Objects

Configure the communication settings for a selected channel to ensure uninterrupted communication with the device network.

Time Out

The internal timeout (in milliseconds) of the TwinCAT library. This timeout is valid only when trying to find a remote TwinCAT runtime. If the TwinCAT runtime is found but the system is not running, this timeout will not apply.

Max Block Size

The maximum number of addresses that will be read in the same block. If you type 0, it will default to 512.

Import Symbol Table

This option determines when the symbols are imported from the TwinCAT PLC or runtime.

- **After (re)connecting**

  The symbol table is dynamically imported every time this Communication Driver communicates with the target device. This option is supported by all versions of TwinCAT.

- **On startup only**

  The symbol table is imported only once, when this Communication Driver communicates with the target device for the first time after startup. This option is more efficient, but it is supported only by TwinCAT 3.1 or later.

Handle Optimization

The driver creates ADS connections or handles to the PLC in runtime. The default setting for this field is 0, where default number of handles is not defined. If this field is set to 1, the driver optimizes the number of connections and sets the maximum number of connections possible is 1024.

Advanced

Click this button to open the **Advanced Settings** dialog box, which provides access to additional communication settings such as timeouts, retries, and buffer sizes. You might need to change these settings if the Communication Driver behaves unexpectedly during run time, but the default settings should work for most network configurations. For more information about these settings, see "Advanced Settings" in the Communication Drivers Pack Help.

Adding and Configuring a Device Selector Object

The TWINCAT Communication Driver can connect to different Windows agents, PLCs, and other data sources. 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 TWINCAT hierarchy

1. Right-click the **New_ChannelSelector_000** object, and select **Add DeviceSelector Connection**.

   The **New_DeviceSelector_000** object is created.

2. Rename the object as appropriate.

3. **The New_DeviceSelector_000 Parameters** configuration view is displayed.

To configure the Device Selector connection

- Configure the **Station**.

  The Station field cannot be empty. The syntax of the Station depends on the ARTI3 or the gateway being used by the Communication Driver. For more information, see Setting the Station ID of a Device.

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

The station ID must use the following syntax:

```
<AMS Net ID>:<runtime or port number>
```

The following syntax diagram shows all of the possible options:

```
AMS Net ID :{ runtime or port number | 301 | 800 | 801 | 811 | 821 | 831 | 851 }
```

where,

AMS Net ID: The AMS Net ID of the TwinCAT PLC or runtime. This ID consists of six numeric values separated by periods (e.g., 5.7.46.126.1.1), and although it can be based on the target's IP address, it is not the same. If you have properly created an AMS route between the computer and the target, you should know the target's ID. For more information, see "Installing and Configuring the TwinCAT ADS Software" at the end of this document.

runtime or port number: The PLC/NC/IO port used to do the communication.

TwinCAT 3 uses fully configurable port numbers. The default value is 851 for TwinCAT PLC, but you can specify anything as long as it matches the port number that has been configured in the TwinCAT 3 programming software.

For TwinCAT 2, the following values have been implemented and tested:

Value	Description
1, 2, 3, 4 301	Runtime number TwinCAT I/O
800	BC9000 or BX9000
801, 811, 821, 831	Equivalent to runtime numbers 1,2,3,4

Examples

Examples of valid station IDs:

5.7.46.126.1.1:1

192.168.1.72.1.1:301

28.39.5.87.1.1:851

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 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 is an "alias" or a label for the data in the device. It is an alternative name for the item reference, and ca 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 "Advanced Settings" in the Communication Drivers Pack Help.

Chapter 3

TWINCAT 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 TWINCAT server. For more general information about item references, see "Managing Device Items" and "Item Reference Descriptions" in the Communication Drivers Pack Help.

This section includes:

Item Reference Syntax

Item references in this Communication Driver use the following syntax.

```
PLC I/O Addresses
```

Note: This type of addressing is supported for TwinCAT 2.x only. If your target device runs TwinCAT 3.x, you must use the variable addressing described in the next section below.

You can access specific PLC I/O addresses for memory and I/O registers using the following syntax:

```
<register type><data type><address number>[.bit]:<variable type>
```

For strings:

```
<register type><data type><address number>:STRING[<length>]
```

The following syntax diagram shows all of the possible options:

```
{ register type | %M | %Q | %I }{ data type | B | X | W | D }address number{ | .bit }:
{ variable type | BOOL | BYTE | WORD | DWORD | SINT | USINT | INT | UINT | DINT | UDINT |
REAL | LREAL | TIME | TOD | DATE | DT | STRING{ | [length] } }
```

where,

register type: The register type: %M (memory), %Q (output), or %I (input).

data type: The data type: B (byte), X (byte for bit access), W (word), or D (double word).

address number: The memory or I/O address number on the PLC.

bit: Indicates the bit number to be read from/written to the device. In order to access individual bits,data type must be X (e.g. %IX0.0). This parameter is optional.

length: The length of the string in bytes/characters. This parameter is optional; if no length is specified, the default length is 80. The maximum possible length on the device is 1024.

Note: Length suffix for string is not supported on TwinCAT version 3.x.

variable type: The variable type that is configured for this address. The valid types are: BOOL, BYTE, WORD, DWORD, SINT, USINT, INT, UINT, DINT, UDINT, REAL, LREAL, TIME, TOD (TIME_OF_DAY), DATE, DT (DATE_AND_TIME), and STRING. If you do not indicate the variable type, it will automatically fill this field with the default type according to the configured data type: BOOL for X, BYTE for B, WORD for W, and DWORD for D.

Note: If the specified STRING length is greater than the actual STRING length on the device, the device might return a "string overflow” error.

Global and Local Variables

You can also access Global and Local Variables, but only on Windows-based runtimes (e.g., CX1000, CX1020, WinCE, WinXP, WinXPe).

For Global Variables on TwinCAT 2.x only, type a period and then the variable name
(e.g., .engine, .speed, .counter): .<variable name>

For Global Variables on TwinCAT 3.x only, as well as for Local Variables on both TwinCAT 2.x and TwinCAT 3.x, type the POU name followed by the period and then the variable name:

<POU name>

.<variable name>

The following syntax diagram shows all of the possible options:

```
{ | POU name }.variable name{ | [index] }{ | :length }
```

where,

POU name: The name of the program organization unit (POU) that contains the variable.

index: Both Global and Local Variables support an optional array index, which is used to access arrays and structures. If you declare arrays in your TwinCAT PLC program as…

MyRealArray:ARRAY[0..10]OF REAL;

MyDINT:ARRAY[0..10]OF DINT;

the corresponding addresses (assuming the first position of each array) should be:

.MyRealArray[0]

.MyDINT[0]

variable name: The name of the Global or Local Variable.

length: Both Global and Local Variables also support an optional string length, which is used to predict the block size that will be necessary to create virtual groups that will not be bigger than allowed. If not set, no assumption about the length is made, and the Max Block Size setting in the communication settings will determine the number of tags on each block. The length is especially useful when reading several string variables that may or may not have its size determined on the TwinCAT program, and not setting its length on the address may lead to errors of Invalid Block Size. Also, if using the length, you must type the same length as used on the TwinCAT program.

If you declare string variables in your TwinCAT PLC program as…

string1:STRING; string2:STRING(150);

then the corresponding addresses should be:

.string1:80

.string2:150

In the first case the string length is taken as 80, which is the default value to ensure correct separation of virtual groups.

The length field syntax is also valid for members of arrays and structures.

Note: Length suffix for string is not supported on TwinCAT version 3.x.

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	TwinCAT Version	Item Name	Description
Memory(%M)	2.x only	%M	Memory: Read and write data value in the memory area

Output(%Q)
Input(%I)
	2.x only
	2.x only
		%Q
		%I
			Output: Read and write data value in the outputs area.
			Input: Read and write data value in the inputs area.

Local or Global Variable	2.x and 3.x	NAME	Variable Name: Up to 512 in the same device. See Max Block Size in the communication settings.
Array Variable	2.x only	.VAR[0..10]	Array Variable Name – the limit is the variable array size. This is supported only in one-dimensional arrays.

Note: Communication with Windows-based TwinCAT runtime systems (e.g., Windows XP, CX1000, CX1020, Windows Embedded Compact) is possible only through Variable Names. Communication with a Beckhoff BC9000/BX9000 is possible only through the %M, %Q, and %I registers.

Supported Data Types

The data type is specified as a suffix in the item syntax. This Communication Driver supports the following data types.

DataType	Suffix	Description
Boolean	BOOL	Boolean value
Byte	BYTE	Unsigned,8-bit variable.
Word DWord	WORD DWORD	Unsigned,16-bit decimal value.(Word) **Note:**This is the default when no data type is specified. Unsigned,32-bit decimal value (double Word).
Signed Short Integer Unsigned Short Integer	SINT USINT	Signed,8-bit decimal value. Unsigned,8-bit decimal value.
Signed Integer	INT	Signed,16-bit decimal value.
Unsigned Integer Signed Integer	UINT DINT	Unsigned,16-bit decimal value. Signed, 32-bit decimal value.
Unsigned Integer Floating Point	UDINT REAL	Unsigned, 32-bit decimal value. Floating point, 32-bit value.
Floating Point	LREAL	Floating point, 64-bit value.
Duration time	TIME	The most significant digit is one millisecond, 32-bit value.
Time of day Date	TOD DATE	The most significant digit is one millisecond, 32-bit value. The most significant digit is one second, 32-bit value.
Date and time	DT	The most significant digit is one second, 32-bit value.

Note: The table above shows suggested usages and addressing possibilities for a TwinCAT data type given the address formats available. All of these data types are used only in I/O addressing, which is supported only by TwinCAT 2.x. 64-bit integers (i.e., LWORD, LINT, ULINT) are not supported

Examples of Item References

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

Examples for TwinCAT 2.x

Device Address	Item Reference
%QX0.0	%QX0.0:BOOL

%QX0.5	%QX0.5:BOOL
%QX1.0	%QX1.0:BOOL
%QX7.7 %QB0(8bits)	%QX7.7:BOOL %QB0:BYTE
%QW0	%QW0:WORD
%MW0(Word)	%MW0:WORD
%MW5(INT)	%MW5:INT
%MB5(BYTE) %MX5.0	%MB5:BYTE %MX5.0:BOOL
%MD15(REAL)	%MD15:REAL
%IX10.7	%IX10.7:BOOL
%IW10 %I10and%I11(16bits)	%IW10:WORD %IW10

engine(GlobalVariable)
speed(GlobalVariable)
	.engine
	.speed

down(GlobalVariable)	.down

engine(GlobalVariableonTwinCAT3)
balance(mainprogram)
	GVL.engine
	MAIN.balance

timerUp.StartTime(Global) MyDINT[3](Global)	.timerUp.StartTime .MyDINT[3]
MyArray[3,3,3](Global)	.MyArray[3,3,3]

Examples for TwinCAT 3.x

Device Address	**Item Reference**
engine(GlobalVariable,List:GVL)	GVL.engine

speed(GlobalVariable,List:Glbl1)
down(GlobalVariable,List:Global)
	Glbl1.speed
	Global.down

engine(GlobalVariableonTwinCAT3)
balance(mainprogram)
	GVL.engine
	MAIN.balance

MyDINT[3,3,3](mainprogram)	MAIN.MyDINT[3,3,3]

Chapter 4

Troubleshooting the TWINCAT Communication Driver

TWINCAT 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	Invalid AMS Net ID Station	Invalid AMS Net ID Station. The AMS Net ID must have 6 fields.	Type a valid AMS Net ID.
30	Error loading third party library	The third party library is not installed (tcAdsDll.dll or tcADsDllCE.dll).	Install the third party library TwinCAT ADS Interface PLC software.
40	Error loading extern function	The extern function cannot be loaded.	Install the correct tcAdsDll.dll or tcAdsDllCE.dll library. The library exists, but the extern functions cannot be loaded.
60	Invalid datatype	Specified address contains an invalid value.	Type a valid address.
70	Error in the GetLocalAddress function	The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Contact your application technical support representative.
90	Error writing data	The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Note the error text in the LogWin, and contact your application technical support representative.

100	Error in the ADSIGRP_SYM_READ WRITE (InfoByNameEx) service	Cannot load the TwinCAT PLC variables information. The TwinCAT ADS Interface PLC is not working properly, TwinCAT software is not running, or the configured variable name is not configured in the TwinCAT PLC software.	Confirm that the variable name is configured the same as in the PLC program.
110	Error in the ADSIGRP_SYM_READ WRITE service	Cannot read the TwinCAT PLC variables values. The TwinCAT ADS Interface PLC is not working properly, or TwinCAT software is not running.	Confirm that the variable name is configured the same as in the PLC program.
120	Error processing value	The configured variable name or address is not configured in the TwinCAT PLC software or is out of range.	Confirm that the variable name is configured the same as in the PLC program.
200	Too many connections	Ams Net ID is over the limit.	Only 64 Ams Net ID addresses can be configured.
211	Invalid Read Block Size	Not enough data buffer.	Configure upto 7 KB of data.
213	Invalid Header	The configured Header is not valid.	Check the documentation for a valid Header configuration.
215 216	Invalid Block Size ARRAY Invalid Symbol	Not enough data buffer ARRAY. Invalid NAME	Configure upto 7 KB of data. Correct the invalid NAME
217	Invalid Connection	Invalid connection configuration.	Check the connection configuration
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. • Wrong station number. • Wrong parity (for serial communication). • Wrong RTS/CTS configuration (for serial communication).	• Check cable wiring. • Check the PLC mode — it must be RUN. • Check the station number. • Increase the time out in the driver's advanced settings. • Check the RTS/CTS configuration (for serial communication).

Installing and Configuring the TwinCAT ADS Software

This section describes how to install and configure the TwinCAT Automation Device Specification (ADS) software that is required for communication with TwinCAT PLCs and runtimes.

Download and install the ADS software on the local computer

To communicate with TwinCAT PLCs and runtimes, you must have the ADS software installed and configured on the same computer where this Communication Driver is installed (hereafter called "the local computer"). The ADS software allows the local computer to present itself as a TwinCAT node on the network, and this Communication Driver communicates through it.

The ADS software is installed as part of the full TwinCAT software, so if you already have the full TwinCAT software installed on the local computer, there is nothing more you need to do. Otherwise, you need to install and configure the ADS software separately.

At the time this document was written, you could download the ADS software installer from the following location: www.beckhoff.com/english/twincat/tc1000.htm

After you download the installer, run it and follow the instructions. You will need to restart the local computer to finish the installation, and when you do, the software will run automatically. By default, the software is installed at: C:\TwinCAT\

Your use of the ADS software is subject to the License Agreement that is installed with the software. For more information about the License Agreement, please contact Beckhoff.

Add an AMS route between the local computer and the target

To establish communication between the local computer and a target PLC or runtime, you need to add an AMS route between the two. This can be done on either the local computer or the target,if both have valid AMS Net IDs.

Each TwinCAT node on the network or each PLC, runtime, or computer that has the ADS software installed, has a unique AMS Net ID that consists of six numeric values separated by periods (e.g., 5.7.46.126.1.1). When you install the ADS software on a computer, that computer is given a default AMS Net ID based on the IP address of the computer. The AMS Net ID is separate from the IP address, however, if you change the IP address, the AMS Net ID is not updated to match. You can manually change the AMS Net ID, if necessary.

To add the AMS route on the local computer:

1. In the notification area of the Windows taskbar, right-click the TwinCAT icon, and then on the shortcut menu, click **Router** > **Edit Routes**. (You might need to expand the notification area if the TwinCAT icon is hidden.) The **TwinCAT Static Route**s dialog box is displayed.
2. In the **TwinCAT Static Routes** dialog box, click **Add**. The **Add Route** dialog box is displayed.
3. If the target is located on the same network as the local computer, you should be able to select it:
   a. Click **Broadcast Search** to get a list of targets that broadcast their presence on the network.
   b. Select your target in the list. The route settings are automatically configured for the selected target.
4. If the target is not located on the same network as the local computer, you need to manually configure the route settings:

   a. In the **Route Name (Target)** box, type a name for the target. This is the name that will be displayed in the local list of routes of the computer, after you add the route.
   b. In the **Route Name (Remote)** box, type a name for the local computer. This is the name that will be displayed in the list of routes of the target, after you add the route. The default name is the host name of the local computer, and you can change it if necessary.
   c. In the **AmsNetID** box, type the ID of the target. If you do not know the ID, either use the TwinCAT programming software to get it or use Broadcast Search on another computer on the network of the target.
   d. In the **Transport Type** list, select the transport type or protocol of the network. In most cases, you should select TCP_IP (i.e., TCP/IP). For all other options, please contact Beckhoff.
   e. In the **Address Info** box, type the host name or IP address of the target.Make sure the corresponding option either **Host Name** or **IP Address** is selected.

   Tip: You can use the ping command, at the Windows command prompt, to confirm that the specified host name or IP address is valid and accessible.

5. Click **Add Route**. The **Add Route** dialog box is closed, and the route is added to the list of routes of the local computer.
6. Close the **TwinCAT Static Routes** dialog box.

Alternatively, if you want to add the AMS route on the target, see the documentation of the manufacturer for that PLC or runtime.

Test the AMS route that you have added

After you have added the AMS route between the local computer and the target, you should test the route to make sure they can communicate with each other. To test the route:

1. On the local computer, locate and run the ADS test program (TcAdsTest.exe). The TcAdsTest window is displayed.

   Note: There are three copies of TcAdsTest.exe included in the ADS software. If the software was installed at its default location, the three copies should be located at:

   - C:\TwinCAT\AdsApi\TcAdsDll\TcAdsTest.exe
   - C:\TwinCAT\AdsApi\TcAdsTest\TcAdsTest.exe
   - C:\TwinCAT\Common32\TcAdsTest.exe

     All three copies function the same, so you can use any one of them.

2. In the **TcAdsTest** window, click **AdsPortOpen**. An alert message is displayed to inform you that the ADS port of the computer has been opened for communication.
3. In the **TcAdsTest** window, click **Test**. The Test window is displayed.
4. In the **Test** window, in the **AmsNetId** box, type the ID of the target.
5. Click **Start** to start the test. The number of successful operations (e.g., n Successful) should be displayed in the **Output** box, and the number should keep increasing as long as the test is running.
6. Click **Stop** to stop the test.
7. Close the **Test** window, and then close the **TcAdsTest** window.

If the test results confirm that the local computer and the target can communicate with each other, this Communication Driver should also be able to communicate with the target through the ADS software.