

SIXNET EtherTRAK Driver Help

© 2012 Kepware Technologies

Table of Contents

Table of Contents	2
SIXNET EtherTRAK Driver Help	3
Overview	3
Device Setup	4
Addressing RS-485 RemoteTRAK Devices Over the Ethernet	6
Optimizing Your SIXNET EtherTRAK Ethernet Communications	7
Data Types Description	8
Error Descriptions	10
Address Validation	10
Missing address	10
Device address '<address>' contains a syntax error	10
Address '<address>' is out of range for the specified device or register	11
Data Type '<type>' is not valid for device address '<address>'	11
Device address '<address>' is Read Only	11
Array size is out of range for address '<address>'	11
Array support is not available for the specified address: '<address>'	11
Device Status Messages	11
Device '<device name>' is not responding	12
Unable to write to '<address>' on device '<device name>'	12
Driver Error	12
Winsock initialization failed (OS Error = n)	12
Winsock V1.1 or higher must be installed to use the SIXNET EtherTRAK device driver	12
Device Specific Messages	13
Bad address in block [<start address> to <end address>] on device '<device name>'	13
Block size mismatch reading [<start address> to <end address>] on device '<device name>'	13
Block request [<start address> to <end address>] on device '<device name>' responded with exception. = n	13
Index	14

SIXNET EtherTRAK Driver Help

Help version 1.014

CONTENTS

[Overview](#)

What is the SIXNET EtherTRAK Driver?

[Device Setup](#)

How do I configure a device for use with this driver?

[Addressing RemoteTRAK Devices Over the Ethernet](#)

How do I address RS-485 RemoteTRAK devices over Ethernet?

[Optimizing Your SIXNET EtherTRAK Ethernet Communications](#)

How do I get the best performance from the SIXNET EtherTRAK driver?

[Data Types Description](#)

What data types does this driver support?

[Address Descriptions](#)

How do I address a data location on a SIXNET EtherTRAK device?

[Error Descriptions](#)

What error messages does the SIXNET EtherTRAK driver produce?

Overview

The SIXNET EtherTRAK Driver provides an easy and reliable way to connect SIXNET EtherTRAK devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with SIXNET EtherTRAK devices that support the Modbus Open TCP/UDP protocol. This driver utilizes UDP socket communications to provide maximum performance with minimal overhead.

Device Setup

Supported Devices

SIXNET EtherTRAK I/O modules (firmware version 2.10 or later)
SIXNET RemoteTRAK I/O connected through an EtherTRAK I/O module.*
SIXNET VersaTRAK RTUs (firmware version 2.12 or later)
SIXNET SIXTRAK gateways (firmware version 2.12 or later)

*Both the RemoteTRAK and EtherTRAK must have firmware version 2.01 or later.

Communication Protocol

Modbus Open Protocol over Ethernet using Winsock V1.1 or higher.

Maximum Number of Channels

The maximum number of supported channels is 100.

Device ID (EtherTRAK IP Network Address Without RemoteTRAK RS-485 Bridging)

SIXNET EtherTRAK devices are networked using standard IP addressing. Determine and set the IP of the SIXNET EtherTRAK modules using the SIXNET Remote IO Toolkit. In general, the Device ID has the following format: YYY.YYY.YYY.YYY, where YYY designates the device IP address. Each YYY byte should be in the range of 0 to 255. If intending to address RemoteTRAK modules hung from the SIXNET EtherTRAK module's RS-485 port, refer to [Addressing RS-485 RemoteTRAK Devices Over Ethernet](#).

Port Number

This parameter specifies the TCP/IP port number that the remote device is configured to use. The default setting is 502.

Coil Block Sizes

Coils can be read from 8 to 800 points (bits) at a time. A higher block size means more points will be read from the device in a single request. If data needs to be read from non-contiguous locations within the device, the block size can be reduced. The default setting is 32.

Note: The Device Properties refer to Coils as Discretes.

Register Block Sizes

Registers can be read from 1 to 120 locations (words) at a time. A higher block size means more register values will be read from the device in a single request. If data needs to be read from non-contiguous locations within the device, the block size can be reduced. The default setting is 32.

Connection Timeout

This parameter specifies the amount of time the driver should wait while connecting to a device. If the timeout is reached before making a connection to a device, the driver will post an error and move onto the next request. The valid range is 1 to 30 seconds. The default setting is 3 seconds.

Request Timeout

This parameter specifies how long the driver should wait for a response from the devices on the channel before timing out. The valid range is 100 to 30000 milliseconds. The default setting is 1000 milliseconds.

Failure Count

This parameter specifies how many times the driver should retry a message before considering communications with the device to be lost. Once this count is reached, the driver will set the ERROR tag for this device to TRUE. The valid range for the Failure count is 1 to 10 retries. The default setting is 3 retries.

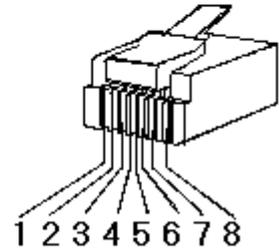
Cable Connections

Patch Cable (Straight Through)

TD + 1	OR/WHT	OR/WHT	1	TD +
TD - 2	OR	OR	2	TD -
RD + 3	GRN/WHT	GRN/WHT	3	RD +
4	BLU	BLU	4	
5	BLU/WHT	BLU/WHT	5	
RD - 6	GRN	GRN	6	RD -
7	BRN/WHT	BRN/WHT	7	
8	BRN	BRN	8	

RJ45 RJ45

10 BaseT



Crossover Cable

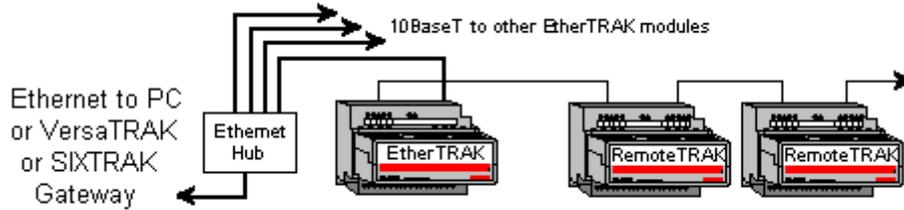
TD + 1	OR/WHT	GRN/WHT	1	TD +
TD - 2	OR	GRN	2	TD -
RD + 3	GRN/WHT	OR/WHT	3	RD +
4	BLU	BLU	4	
5	BLU/WHT	BLU/WHT	5	
RD - 6	GRN	OR	6	RD -
7	BRN/WHT	BRN/WHT	7	
8	BRN	BRN	8	

RJ45 RJ45

8-pin RJ45

Addressing RS-485 RemoteTRAK Devices Over the Ethernet

The SIXNET EtherTRAK Driver can address RemoteTRAK I/O connected on a RS485 party-line by connecting through the SIXNET ET-GT-485-1 Ethernet to Modbus RS485 gateway. Any SIXNET EtherTRAK I/O module messages are passed through to the RS485 port.



The intelligent Modbus interfaces accept Modbus Ethernet commands and convert them to traditional Modbus messages. They then pass the command along to the RS485 port on the Ethernet interface. Replies from the station are returned to Modbus Ethernet format and passed back on the Ethernet network to the SIXNET EtherTRAK OPC server.

Device ID (EtherTRAK IP Network Address with RemoteTRAK RS-485 Bridging)

SIXNET EtherTRAK devices are networked using standard IP addressing. Users can determine or set the IP of the SIXNET EtherTRAK modules using the SIXNET Remote IO Toolkit. In general, the Device ID has the following format `YYY.YYY.YYY.YYY`, where `YYY` designates the device IP address. Each `YYY` byte should be in the range of 0 to 255.

When addressing RemoteTRAK devices via the SIXNET EtherTRAK module's RS-485 port, include the station number of the RemoteTRAK device as part of the SIXNET EtherTRAK IP address. Using the same format as the IP address, adding the RemoteTRAK station number would take the following format: `YYY.YYY.YYY.YYY.ZZZ`. The normal IP address remains the same as denoted by the `YYY.YYY.YYY.YYY`; however, the station number of the desired RemoteTRAK unit on the SIXNET EtherTRAK module's RS-485 port is denoted by the `.ZZZ`. The valid station number range for `ZZZ` is 1 to 247.

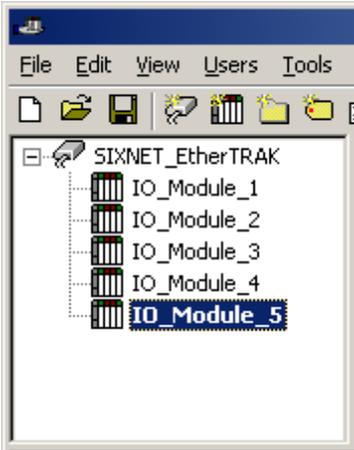
Example

Assume that the SIXNET EtherTRAK module is at IP address `10.1.1.10`. To this EtherTRAK module, the user desires to attach four RemoteTRAK modules that have already been configured with Modbus Station numbers of 1, 2, 3 and 4. In the SIXNET EtherTRAK OPC Server, the user would add four separate devices to the SIXNET EtherTRAK project. The first device would have a Device ID of `10.1.1.10.1`; the second `10.1.1.10.2`; the third `10.1.1.10.3`; the fourth `10.1.1.10.4`. Thus, although each Device ID has the same IP address, the last field contains the actual station number of each RemoteTRAK device attached to the RS-485 port.

Optimizing Your SIXNET EtherTRAK Ethernet Communications

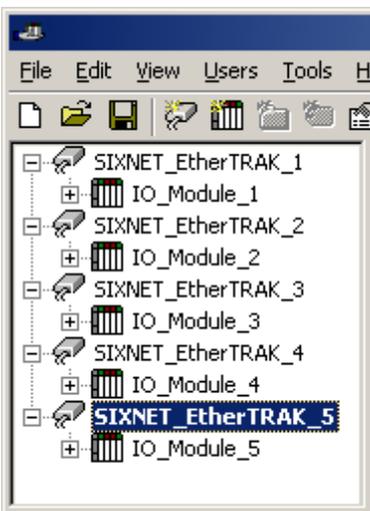
The SIXNET EtherTRAK driver driver has been designed to provide the best performance with the least amount of impact on the system's overall performance. While the SIXNET EtherTRAK driver driver is fast, there are a couple of guidelines that can be used in order to control and optimize the application and gain maximum performance.

Our server refers to communications protocols like EtherTRAK driver as a channel. Each channel defined in the application represents a separate path of execution in the server. Once a channel has been defined, a series of devices must then be defined under that channel. Each of these devices represents a single EtherTRAK I/O module from which data will be collected. While this approach to defining the application will provide a high level of performance, it won't take full advantage of the SIXNET EtherTRAK driver driver or the network. An example of how the application may appear when configured using a single channel is shown below.



Each device appears under a single SIXNET EtherTRAK driver channel. In this configuration, the driver must move from one device to the next as quickly as possible in order to gather information at an effective rate. As more devices are added or more information is requested from a single device, the overall update rate begins to suffer.

If the SIXNET EtherTRAK driver driver could only define one single channel, then the example shown above would be the only option available; however, the SIXNET EtherTRAK driver driver can define up to 100 channels. Using multiple channels distributes the data collection workload by simultaneously issuing multiple requests to the network. An example of how the same application may appear when configured using multiple channels to improve performance is shown below.



Each device has now been defined under its own channel. In this new configuration, a single path of execution is dedicated to the task of gathering data from each device. If the application has fewer devices than channels, it can be optimized exactly as it is shown here.

The performance will improve even if the application has more devices than channels. While one device per channel is ideal, the application will still benefit from additional channels. Although spreading the device load across all channels will cause the server to move from device to device again, it can now do so with far less devices to process on a single channel.

Block Size, which is available on each defined device, can also affect the SIXNET EtherTRAK driver driver's performance. Block Size refers to the number of bytes that may be requested from a device at one time. To refine the performance of this driver, configure Block Size from 1 to 120 registers per request. The coil block size can be adjusted from 8 to 800. Increase the block size setting for the device if the application consists of large requests for consecutively ordered data.

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value bit 0 is the low bit bit 7 is the high bit
Char	Signed 8 bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
DWord Example	If register 40001 is specified as a dword, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type. The reverse is true when this is not selected.
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Long Example	If register 40001 is specified as a long, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type. The reverse is true when this is not selected.
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
Float	32 bit floating point value. The driver interprets two consecutive registers as a floating point value by making the second register the high word and the first register the low word.
Float Example	If register 40001 is specified as a float, bit 0 of register 40001 would be bit 0 of the 32 bit word, and bit 15 of register 40002 would be bit 31 of the 32 bit word.

Note: The descriptions above assume first word low data handling of 32 bit data types.

Address Descriptions

Access to I/O of different types is supported by the Modbus messaging protocol via address ranges. To access the following SIXNET data types, use the following Modbus address ranges.

SIXNET Data Type	SIXNET Address	Modbus Data Address
Discrete In:	0 .. 9999	10001 .. 20000
Discrete Out:	0 .. 9999	00001 .. 10000
Analog In:	0 .. 2999	30001 .. 33000
Analog Out:	0 .. 2999	40001 .. 43000
Short Integer In	0 .. 1999	33001 .. 35000
Long Integer In	0 .. 1999	35001 .. 37000
Floating Point In:	0 .. 2999	37001 .. 40000
Short Integer Out:	0 .. 1999	43001 .. 45000
Long Integer Out:	0 .. 1999	45001 .. 47000
Floating Point Out:	0 .. 2999	47001 .. 50000

Examples

1. Modbus data address 10001 equates to SIXNET discrete input 0.
2. Modbus data address 30006 equates to SIXNET analog input 5.

Generic Modbus Addressing Decimal Format

The default data types for dynamically defined tags are shown in **bold**.

Address	Range	Data Type	Access
Output Coils [Function Codes (decimal): 01, 05, 15]	000001-065536	Boolean , Byte, Char, Word, Short*	Read/Write
Input Coils [Function Code (decimal): 02]	100001-165536	Boolean , Byte, Char, Word, Short*	Read Only
Internal Registers [Function Code (decimal): 04]	300001-365536 300001-365535 3xxxxx.0-3xxxxx.15	Word , Short, BCD Float, DWord, Long, LBCD Boolean	Read Only
Holding Registers [Function Codes (decimal): 03, 06, 16]	400001-465536 400001-465535 4xxxxx.0-4xxxxx.15	Word , Short, BCD Float, DWord, Long, LBCD Boolean	Read/Write

*When accessing coils as a byte or char, the address specified must lie on a byte boundary (such as xxxxx1, xxxxx9, xxxxx17 and so forth). When accessed as a word or short, the address specified must lie on a word boundary (such as xxxxx1, xxxxx17, xxxxx33 and so forth).

Examples

1. To access SIXNET Discrete Output 0, enter a Modbus address of 00001.
2. To access SIXNET Analog In 3, enter a Modbus address of 30004.
3. To access SIXNET Analog Out 2, enter a Modbus address of 40003.

Array Support

Arrays are supported for internal and holding register locations for all data types except for Boolean. There are two methods of addressing an array. Examples are given using holding register locations.

4xxxx [rows] [cols]
4xxxx [cols] this method assumes rows is equal to one.

Rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register type. For arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

[Missing address](#)

[Device address '<address>' contains a syntax error](#)

[Address '<address>' is out of range for the specified device or register](#)

[Data Type '<type>' is not valid for device address '<address>'](#)

[Device address '<address>' is Read Only](#)

[Array size is out of range for address '<address>'](#)

[Array support is not available for the specified address: '<address>'](#)

Device Status Messages

[Device '<device name>' is not responding](#)

[Unable to write to '<address>' on device '<device name>'](#)

Driver Error Messages

[Winsock initialization failed \(OS Error = n\)](#)

[Winsock V1.1 or higher must be installed to use the SIXNET EtherTRAK device driver](#)

Device Specific Messages

[Bad address in block \[<start address> to <end address>\] on device '<device name>'](#)

[Block size mismatch reading \[<start address> to <end address>\] on device '<device name>'](#)

[Block request \[<start address> to <end address>\] on device '<device name>,' responded with exception = n](#)

Address Validation

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

[Missing address](#)

[Device address '<address>' contains a syntax error](#)

[Address '<address>' is out of range for the specified device or register](#)

[Data Type '<type>' is not valid for device address '<address>'](#)

[Device address '<address>' is Read Only](#)

[Array size is out of range for address '<address>'](#)

[Array support is not available for the specified address: '<address>'](#)

Missing address

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically has no length.

Solution:

Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically contains one or more invalid characters.

Solution:

Re-enter the address in the client application.

Address '<address>' is out of range for the specified device or register

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically references a location that is beyond the range of supported locations for the device.

Solution:

Verify that the address is correct; if it is not, re-enter it in the client application.

Data Type '<type>' is not valid for device address '<address>'

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically has been assigned an invalid data type.

Solution:

Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically has a requested access mode that is not compatible with what the device supports for that address.

Solution:

Change the access mode in the client application.

Array size is out of range for address '<address>'

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically is requesting an array size that is too large for the address type or block size of the driver.

Solution:

Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: '<address>'

Error Type:

Warning

Possible Cause:

A tag address that has been specified dynamically contains an array reference for an address type that doesn't support arrays.

Solution:

Re-enter the address in the client application to remove the array reference or correct the address type.

Device Status Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Status Messages

[Device '<device name>' is not responding](#)

Unable to write to '<address>' on device '<device name>'**Device '<device name>' is not responding**

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Unable to write to '<address>' on device '<device name>'

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Driver Error

The following error/warning messages may be generated. Click on the link for a description of the message.

Driver Error Messages[Winsock initialization failed \(OS Error = n\)](#)[Winsock V1.1 or higher must be installed to use the SIXNET EtherTRAK device driver](#)**Winsock initialization failed (OS Error = n)**

Error Type:

Fatal

OS Error	Indication	Possible Solution
10091	Indicates that the underlying network subsystem is not ready for network communication.	Wait a few seconds and restart the driver.
10067	Limit on the number of tasks supported by the Windows Sockets implementation has been reached.	Close one or more applications that may be using Winsock and restart the driver.

Winsock V1.1 or higher must be installed to use the SIXNET EtherTRAK device driver

Error Type:

Fatal

Possible Cause:

The version number of the Winsock DLL found on the system is less than 1.1.

Solution:

Upgrade Winsock to version 1.1 or higher.

Device Specific Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Specific Messages

[Bad address in block \[<start address> to <end address>\] on device '<device name>'](#)

[Block size mismatch reading \[<start address> to <end address>\] on device '<device name>'](#)

[Block request \[<start address> to <end address>\] on device '<device name>.' responded with exception = n](#)

Bad address in block [<start address> to <end address>] on device '<device name>'

Error Type:

Serious

Possible Cause:

An attempt has been made to reference a nonexistent location in the specified device.

Solution:

Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Block size mismatch reading [<start address> to <end address>] on device '<device name>'

Error Type:

Fatal for addresses falling in this block.

Possible Cause:

The driver attempted to read a block of memory in the PLC. The PLC responded with no error, but did not provide the driver with the requested block size of data.

Solution:

Ensure that the range of memory exists for the PLC.

Block request [<start address> to <end address>] on device '<device name>' responded with exception = n

Error Type:

Fatal for addresses falling in this block.

Possible Cause:

The driver attempted to read a block of memory in the PLC. The PLC responded with the exception error shown.

Solution:

Ensure that the range of memory or data type exists for the PLC.

Index

A

Address '<address>' is out of range for the specified device or register.....	11
Address Descriptions.....	8
Address Validation.....	10
Addressing RS-485 RemoteTRAK Devices over the Ethernet.....	6
Array size is out of range for address '<address>'.....	11
Array support is not available for the specified address.....	11
Array support is not available for the specified address:'<address>'.....	11

B

Bad address in block [<start address> to <end address>] on device '<device name>'.....	13
BCD.....	8
Block request [<start address> to <end address>] on device '<device name>' responded.....	13
with exception = n.....	
Block Size.....	4
Block size mismatch reading [<start address> to <end address>] on device '<device name>'.....	13
Boolean.....	8

C

Coils.....	9
------------	---

D

Data Type '<type>' is not valid for device address '<address>'.....	11
Data Types Description.....	8
Device '<device name>' is not responding.....	12
Device address '<address>' contains a syntax error.....	10
Device address '<address>' is read only.....	11
Device ID.....	4
Device Setup.....	4
Device Specific Messages.....	13

Device Status Messages	11
Driver Error Messages	12
DWord	8

E

Error Descriptions	10
---------------------------------	-----------

F

Float	8
--------------------	----------

L

LBCD	8
Long	8

M

Missing address	10
------------------------------	-----------

N

Network	4
----------------------	----------

O

Optimizing Your SIXNET EtherTRAK Communications	7
Overview	3

R

Registers	9
------------------------	----------

S

Short 8

U

Unable to write to '<address>' on device '<device name>' 12

W

Winsock initialization failed (OS Error = n) 12

Winsock V1.1 or higher must be installed to use the SIXNET EtherTRAK device driver 12

Word 8