

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8705-15
Driver for
Networked Robotics
NTMS4 Temperature Sensor Modules

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

TABLE OF CONTENTS

TABLE OF CONTENTS	2
1. Networked Robotics - NTMS4 Temperature Sensor Driver	3
2. Driver Scope of Supply	4
2.1. Supplied by FieldServer Technologies for this driver	4
2.2. Provided by the Supplier of 3 rd Party Equipment	4
2.2.1. <i>Required 3rd Party Hardware</i>	4
2.2.2. <i>Required 3rd Party Software</i>	4
2.2.3. <i>Required 3rd Party Configuration</i>	4
3. Hardware Connections	5
3.1. Hardware Connection Tips / Hints	5
4. Configuring the FieldServer as a NTMS4e Client	6
4.1. Data Arrays/Descriptors	6
4.2. Client Side Connection Descriptions	7
4.3. Client Side Node Descriptors	8
4.4. Client Side Map Descriptors	9
4.4.1. <i>FieldServer Related Map Descriptor Parameters</i>	9
4.4.2. <i>Driver Related Map Descriptor Parameters</i>	10
4.4.3. <i>Timing Parameters</i>	10
4.4.4. <i>Map Descriptor Example 1 – Read Temperature</i>	11
5. Configuring the FieldServer as a Networked Robotics - NTMS4 Temperature Sensor Server	13
Appendix 1. Useful Features	14
Appendix 2. Troubleshooting tips	15
2.1.1. Connection Tips & Hints	15
2.1.2. Driver Error Messages	17
2.1.3. Exposing Driver Stats	18
Appendix 3. Revision History	20

1. Networked Robotics - NTMS4 Temperature Sensor Driver

The NTMS4 driver can read temperatures from Networked Robotics NTMS4 modules using TCP/IP connections.

The driver can process data from multiple modules. The IP Address and Socket Port can be configured for each module.

The driver reads the temperature and cannot perform configuration of the module. It expects that the modules are correctly configured using the Vendor’s software or Instructions. The driver can store three data elements per module. The time stamp (in seconds since 1970) of the last time good data was obtained, the temperature data and a status value which is set non-zero when the module reports and error. If the module cannot be reached then the FieldServer will (after a configurable number of retries) mark the node offline. This state and the 3 data elements can be monitored using another protocol.

The driver can only be used as a client. Minimal server functionality is provided only to support our ongoing quality assurance program by facilitating automated testing of the driver. It is not documented or supported. If required please contact the FST sales group to discuss your requirements.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	Many	The FieldServer will be able to poll multiple modules. The maximum is 200. The maximum number of sensors tested by CAS is 75.
Server	0	Not supported or documented.

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
-	Every FieldServer is shipped with a generic RJ45 Ethernet patch cable. That cable is suitable to connect the FieldServer to a network to which NTMS4 modules are connected.
FS-8705-15	Driver Manual.

2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Hardware

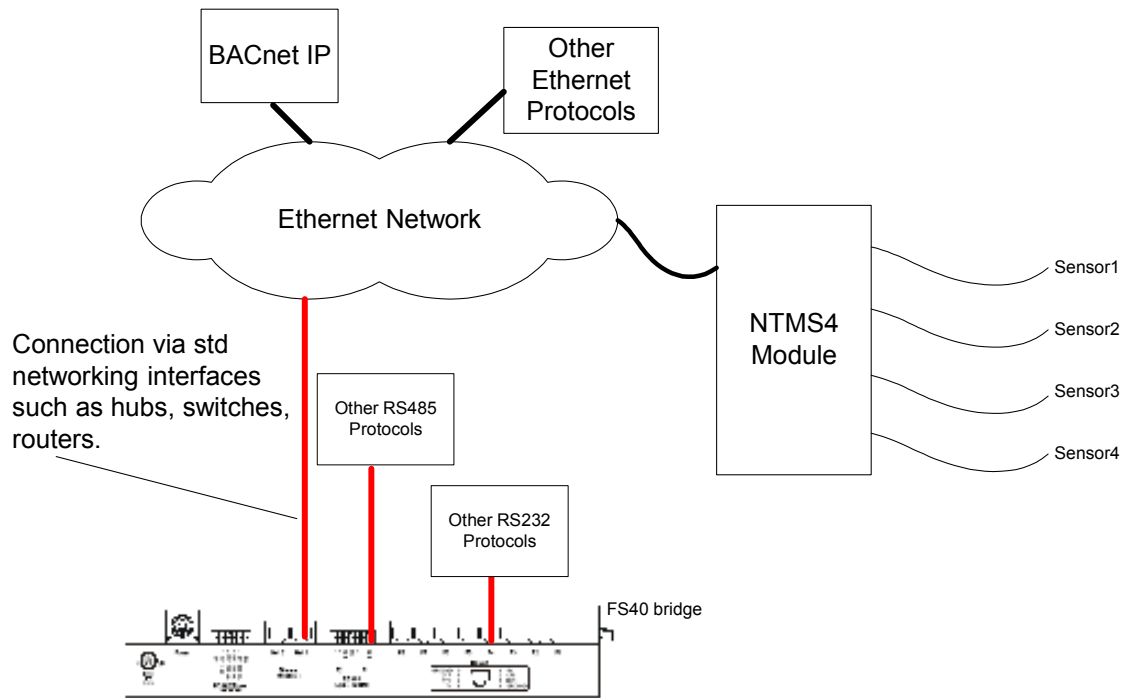
Part #	Description

2.2.2. Required 3rd Party Software

2.2.3. Required 3rd Party Configuration

Modules must be configured correctly before they can be read with this driver.

3. Hardware Connections



3.1. Hardware Connection Tips / Hints

Make sure the FieldServer and NTMS modules are correctly sub-netted and/or routed.

4. Configuring the FieldServer as a NTMS4e Client

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See “.csv” sample files provided with the FS).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Networked Robotics - NTMS4 Temperature Sensor Module

4.1. Data Arrays/Descriptors

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for NTMS4 Serial Driver communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, the destination device addresses need to be declared in the “Client Side Nodes” section, and the data required from the servers needs to be mapped in the “Client Side Map Descriptors” section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

Section Title		
Data Arrays		
Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, UInt32, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10,000

Example

// Data Arrays		
Data_Arrays		
Data_Array_Name,	Data_Format,	Data_Array_Length,
DA_AI_01,	UInt16,	200
DA_AO_01,	UInt16,	200
DA_DI_01,	Bit,	200
DA_DO_01,	Bit,	200

4.2. Client Side Connection Descriptions

Create one connection for each adapter you will be using.

Section Title		
Adapters		
Column Title	Function	Legal Values
Adapter	Specify which port the device is connected to the FieldServer	N1,N2
Protocol	Specify protocol used	nrobot_ntms4
Poll_Delay*	Time between internal polls	0-32000 seconds, 0.0 second
NR_use_error_value	Optional Parameter If the driver reads a module and the module responds reporting an error then you can tell the driver to overwrite the last good temperature read with a specific value indicating an error. This is different from the driver not being able to connect to the NTMS4 module – which is reported as Node Offline and which can be monitored using the Node_Status Bits.	Yes, No
NR_error_value	Optional Parameter If NR_use_error_value is set to yes then the driver uses this value to overwrite the last good temperature read from the module.	Any floating point number. Suggested=999.9 so that it easy for an operator to see something is wrong.

Example

```
// Client Side Connections

Adapters
Adapter, Protocol,
N1, nrobot_ntms4,
```

4.3. Client Side Node Descriptors

Create one Node per connection only.

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	Station address of physical server node This parameter is not used directly by the driver. We recommend that a unique Node ID's be given to each node so that the status of each node can be monitored by another protocol	0-255
Protocol	Specify protocol used	nrobot_ntms4
Adapter	Specify which port the device is connected to the FieldServer	N1,N2
IP_Address	IP Address of the NTMS4 Server	Valid IP Address in the format 192.168.1.101
NR_PortNumber	Default is Port 8088 to 8091 for sensors 1 to 4 on the4 NTMS4 module.. Specify Port number for TCP Socket.	Positive Integer. May be specified in hex in the format 0x1234
timeout	Default is 2.0seconds. Large pages from slow servers may require settings at large as 5.0seconds	Specify a floating point number with at most two digits after the decimal point.

Example

Nodes					
Node_Name,	Node_ID,	Protocol,	Adapter,	IP_Address,	NR_PortNumber
Mod01Sens01,	1,	nrobot_ntms4,	N1,	72.29.75.91,	8088

4.4. Client Side Map Descriptors

4.4.1. FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer The temperature is stored here.	One of the Data Array names from “Data Array” section above We recommend that you use a ‘FLOAT’ Data Array since the temperatures are reported as floating point numbers.
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in “Data Array” section above
Function	Function of Client Map Descriptor. Only read is supported.	RDBC

4.4.2. Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in “Client Node Descriptor” above
Data_Type	This commonly used parameter is not used by this driver.	
Length	Length of Map Descriptor Reserves space in the Data Array. Set to 1.	1
Address	This commonly used FieldServer parameter is not used by this protocol.	
DA_Float_Name	If this parameter is specified then the driver stores the timestamp here. The Timestamp is only stored when a valid temperature is stored. The offset is controlled by Data_Array_Offset	One of the Data Array names from “Data Array” section above
DA_Byte_Name	This parameter is optional. If the module reports an error then the driver specifies the error in this Data Array if it is specified. The offset is controlled by Data_Array_Offset	One of the Data Array names from “Data Array” section above

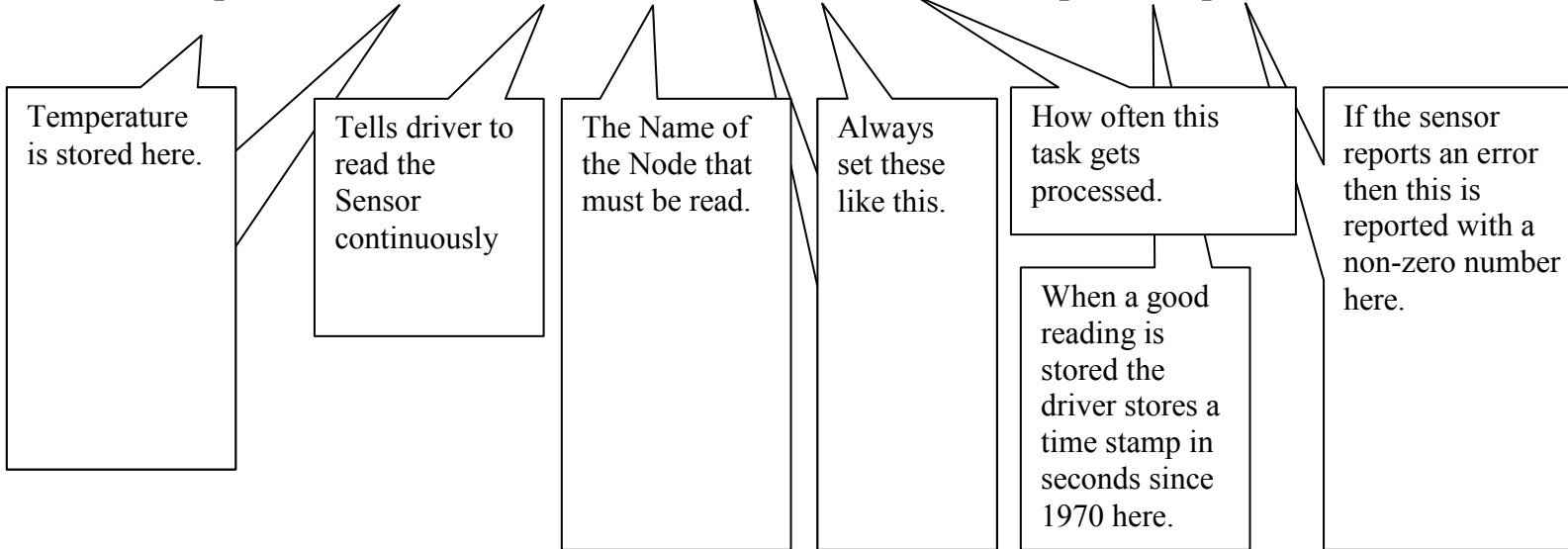
4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled	≥0.001s

4.4.4. Map Descriptor Example 1 – Read Temperature

In this example the driver reads the temperature. It is stored in DA_Sensor[0] and the time stamp is stored in DA_TIMESTAMP[0]. If instead of a temperature the module responds reporting an error then it gets stored in DA_STATUS[0]. This task is executed at best every 5 seconds.

```
Map_Descriptors
Map_Descriptor_Name ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,Address ,Length ,Scan_Interval ,DA_Float_Name ,DA_BYTE_NAME
ReadSens000 ,DA_SENSOR ,0 ,RDBC ,SENSOR000 ,0 ,1 ,5.000s ,DA_TIMESTAMP ,DA_STATUS
```



This Configuration Fragment Illustrates a configuration to read 3 sensors. If the module is offline then the bit corresponding to the Node_ID in the Data Array named DA_Node_Status will be set to zero (online=1). If the sensor responds with an error then the driver will overwrite the value stored in DA_Sensor with 999.9.

```
//=====
//
//   Data Arrays
//
Data_Arrays
Data_Array_Name ,Data_Format ,Data_Array_Length
DA_SENSOR      ,Float      ,200
DA_TIMESTAMP   ,UINT32     ,200
DA_STATUS      ,UINT16     ,200
nrobot_stats   ,UINT16     ,200

Data_Arrays
Data_Array_Name ,Data_Format ,Data_Array_Length ,Data_Array_Function
DA_Node_Status  ,Bit         ,100                ,Node_Status

//=====
//
//   Client Side Connections
//
Connections
Adapter ,Protocol      ,Poll_Delay ,NR_error_value ,NR_use_error_value
N1      ,nrobot_ntms4 ,0.100s   ,999.9          ,Yes

//=====
//
//   Client Side Nodes
//
Nodes
Node_Name ,Node_ID ,Protocol      ,Adapter ,IP_Address      ,NR_PortNumber
SENSOR000 ,0       ,nrobot_ntms4 ,N1      ,63.250.243.40  ,8088
SENSOR001 ,1       ,nrobot_ntms4 ,N1      ,63.250.243.41  ,8088
SENSOR002 ,2       ,nrobot_ntms4 ,N1      ,63.250.243.42  ,8088

//=====
//
//   Client Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,Address ,Length ,Scan_Interval ,DA_Float_Name
,DA_Byte_Name
ReadSens000        ,DA_SENSOR      ,0                ,RDBC    ,SENSOR000 ,0      ,1      ,5.000s        ,DA_TIMESTAMP ,DA_STATUS
ReadSens001        ,DA_SENSOR      ,1                ,RDBC    ,SENSOR001 ,0      ,1      ,5.000s        ,DA_TIMESTAMP ,DA_STATUS
ReadSens002        ,DA_SENSOR      ,2                ,RDBC    ,SENSOR002 ,0      ,1      ,5.000s        ,DA_TIMESTAMP ,DA_STATUS
```

5. Configuring the FieldServer as a Networked Robotics - NTMS4 Temperature Sensor Server

This driver can emulate a NTMS4 module by serving temperature data. However, the functionality has not been documented and is not supported as it was implemented as part of FieldServer's quality assurance program. If you are interested in using Sever Side features then please contact Chipkin Automation Systems.

Appendix 1. Useful Features

Time Stamp Information

The time stamp of the most recent temperature update is made by the FieldServer. The NTMS4 does not time stamp the data. The time stamp is stored as a 32 bit unsigned number which represents the number of seconds since the 1st of January 1970.

For example: The value 1240999200 represents Wed, 29 Apr 2009 10:00:00 GMT

Converters may be found on the web.

Eg. http://www.onlineconversion.com/unix_time.htm

Status Information

If the Map Descriptor parameter named 'DA_Byte_Name' is specified then the driver will store a number indicating the status of the last sensor read. The driver does not store an error code if it cannot connect to the port on the NTMS4.

You are advised to read the NTMS4 manual to check for an updated list of error codes.

Error Number	Interpretation
0	Temperature was read without error
1	The device/sensor is not configured
2	The device/sensor is configured but the NTMS4 cannot communicate with it.

Appendix 2. Troubleshooting tips

2.1.1. Connection Tips & Hints

There are two categories of common problem. Ethernet Connection problems and Sensor Problems.

Ethernet Connection Issues

1. If you can't ping the NTMS4 IP address from a computer on the same subnet as the FieldServer then you will need to work with your network administrator to resolve routing issues.

Start : Run

Type: CMD <Enter>

Type: ping 192.168.1.101<Enter> (where 192.168.1.101 is the IP of the NTMS4)

2. If you can't telnet to the NTMS4 port with the sensor you are testing then either that port is blocked or the NTMS4 has been configured to use a different port than the one you are trying. Use the Networked Robotics software to check the configuration of the device. If you are sure the port number is correct then check your network administrator to see if the port is blocked.

Start : Run

Type: CMD <Enter>

Type: telnet 192.168.1.101 8088<Enter> (where 192.168.1.101 is the IP of the NTMS4 and 8088 is the port of the sensor being tested)

Use Ctrl-] to end the session or read <http://www.chipkin.com/articles/how-to-end-a-telnet-session-windows>

If the FieldServer can't open a socket connection to the port or if multiple consecutive reads fail (from an Ethernet point of view) then it regards that Node as offline.

Monitoring the Ethernet Connection Status

The following config file fragment outlines how you can use this status information. In this example, the Node_ID is assigned as zero (line 11). This corresponds to the 0th item in the Data Array DA_Node_Status. If the Node_ID had been assigned to 1 then the node status will be reported in offset 1 of the Data Array DA_Node_Status. If the node is offline the the Data Array offset is set to zero. If the node is online then the offset is set to 1. You can map these Data Array items onto a server map descriptor so that the remote client can monitor the Ethernet Connection Status.

```

1. Data_Arrays
2. Data_Array_Name ,Data_Format ,Data_Array_Length ,Data_Array_Function
3. DA_Node_Status ,Bit ,100 ,Node_Status
4.
5. //=====
6. //
7. // Client Side Nodes
8. //
9. Nodes
10. Node_Name ,Node_ID ,Protocol ,Adapter ,IP_Address ,NR_PortNumber
11. SENSOR000 ,0 ,nrobot_ntms4 ,N1 ,63.250.243.40 ,8088

```

Sensor Errors

If a connection to the sensor port can be established over the Ethernet Network there are still some potential problems. For example, the sensor may not be connected, may have failed or may not have been configured correctly.

These type of problems can be monitored in two ways.

1. You can have the driver set the sensor value to an error value such as 999.9 or you can define a Data Array where the driver will store a number reporting the success or failure of reading the temperature. Again, we emphasize, these errors are not the same as Ethernet connection errors which are reported differently. In the configuration fragment below the driver has been configured to set the temperature value to 999.9 if there is an error reading the sensor.

```

Connections
Adapter ,Protocol ,Poll_Delay ,NR_error_value ,NR_use_error_value
N1 ,nrobot_ntms4 ,0.100s ,999.9 ,Yes

```

2. You can have the driver update a data array with a number reporting the success/error of the last read. In the configuration fragment below the driver will store this number in the Data Array name 'DA_STATUS' at offset zero

```

Map_Descriptors
Map_Descriptor_Name , Data_Array_Name ,Data_Array_Offset ,.....,DA_Byte_Name
ReadSens000 , DA_SENSOR ,0 ,..... ,DA_STATUS

```


2.1.2. Driver Error Messages

Error Message	Explanation and corrective action
<p>We have shown place holders for the parts of the message which change.</p> <p>%s is a place holder for a text string. %d is a place holder for a number %c is a place holder for an alpha character.</p>	
<p>NTMS4:#01 FYI. Use an Array called <%s> to expose diagnostic info.</p>	<p>The driver exposes operation stats in a specially named Data Array. Read more in section 2.1.3</p>
<p>MTMS4:#02 FYI Using sockets on Port=%d (Hex:%#04x)</p>	<p>The default socket port is 80 (decimal). This message is printed if the socket port is configured using the NR_PortNumber parameter. No corrective action is required.</p>
<p>NTMS4:#03 Err Cant connect to %s (%s)</p>	<p>A TCP/IP socket connection could not be made with the remote device. Its possible the IP Address and/or the Port are incorrect. Check that you can ping the IP Address from your location.</p>
<p>NTMS4:#04 Err Timeout. No response from %s (%s)</p>	<p>A connection to the device was established but we could not read data we could interpret from the device. Perhaps the port is wrong and the driver has been configured (incorrectly) to read the help file.</p> <p>Its conceivable that this error could occur if the format of the NTMS4 responses has changed since the driver was developed.</p> <p>If you get this error please capture a wireshark log.</p>
<p>NTMS4:#05 FYI Debug mode is on NTMS4:#05 FYI Debug mode turned off</p>	<p>If debug mode is on then the driver prints a message to remind you to turn it off as leaving it on means that any important messages will be lost as the error screen is limited in size. Read section 2.1.3</p>

2.1.3. Exposing Driver Stats

The diver makes some of its operating statistics available in a Data Array where they can be read by a remote client. The lines from the example below can be cut and pasted into a configuration file.

```
Data_Arrays,
Data_Array_Name,      Data_Format,      Data_Array_length,
nrobot_stats,        UINT32,          1000,
```

Offset	Description
1	Set this to 1 to turn on some debug diagnostics. Set to zero to turn them off. It may take several seconds (up to 30) for the driver to respond. When debugging the driver prints the messages it receives from the remote field devices.
2	Each time the driver timed out because it could not find a response or the response could not be interpreted this value is incremented by 1. This particular stat is incremented when the driver is waiting for a response from the sensor.
3	Each time the driver timed out because it could not find a response or the response could not be interpreted this value is incremented by 1. This particular stat is incremented when the driver is has a partial response but is waiting for a complete response.
4	Each time the driver timed out trying to establish a socket connection to the sensor this value is incremented by 1.
5	Each time the driver timed out trying to read data or send a poll because the socket was closed by the sensor this value is incremented by 1.
6	Each time the driver timed out because it could not find a response or the response could not be interpreted this value is incremented by 1. This statistic is the sum of the two stats found at offset 2,3,4,5.

Appendix 3. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
21 Apr 2009	PMC		1.00a	0	Document Created
29 Apr 2009	PMC		1.00a	1	Corrections and added Trouble Shooting Tips.
12 May 09	PMC		1.00d	2	Corrections, new stats and updated error messages.