FieldServer
FS-8700-66 GE-SNP

Driver Manual

(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after June 2019.

Driver Revision: 1.06

Document Revision: 1.B

Technical Support

Please call us for any technical support needs related to the FieldServer product.

Sierra Monitor Corporation
1991 Tarob Court
Milpitas, CA 95035

Website: www.sierramonitor.com

U.S. Support Information:

+1 408 964-4443
+1 800 727-4377

Email: support@sierramonitor.com

EMEA Support Information:

+31 33 808 0590

Email: support.emea@sierramonitor.com

TABLE OF CONTENTS

1 GE-SNP DESCRIPTION

The GE-SNP Serial driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using GE-SNP Serial protocol. The FieldServer can emulate either a Server or Client.

The FieldServer provides functions to read and write PLC memory and change the privilege level. Standard SNP mailbox messages are used. The driver does not support Datagram messages and cannot parse them. These messages are defined by the SNP protocol to allow multiple data types to be packed into one message. They are not commonly used by the HMI and 3rd party applications and are inconsistent with the FieldServer's *Write Through and Port Expander* capabilities.

The driver can expose communications statistics in a Data Array so that downstream devices can monitor them.

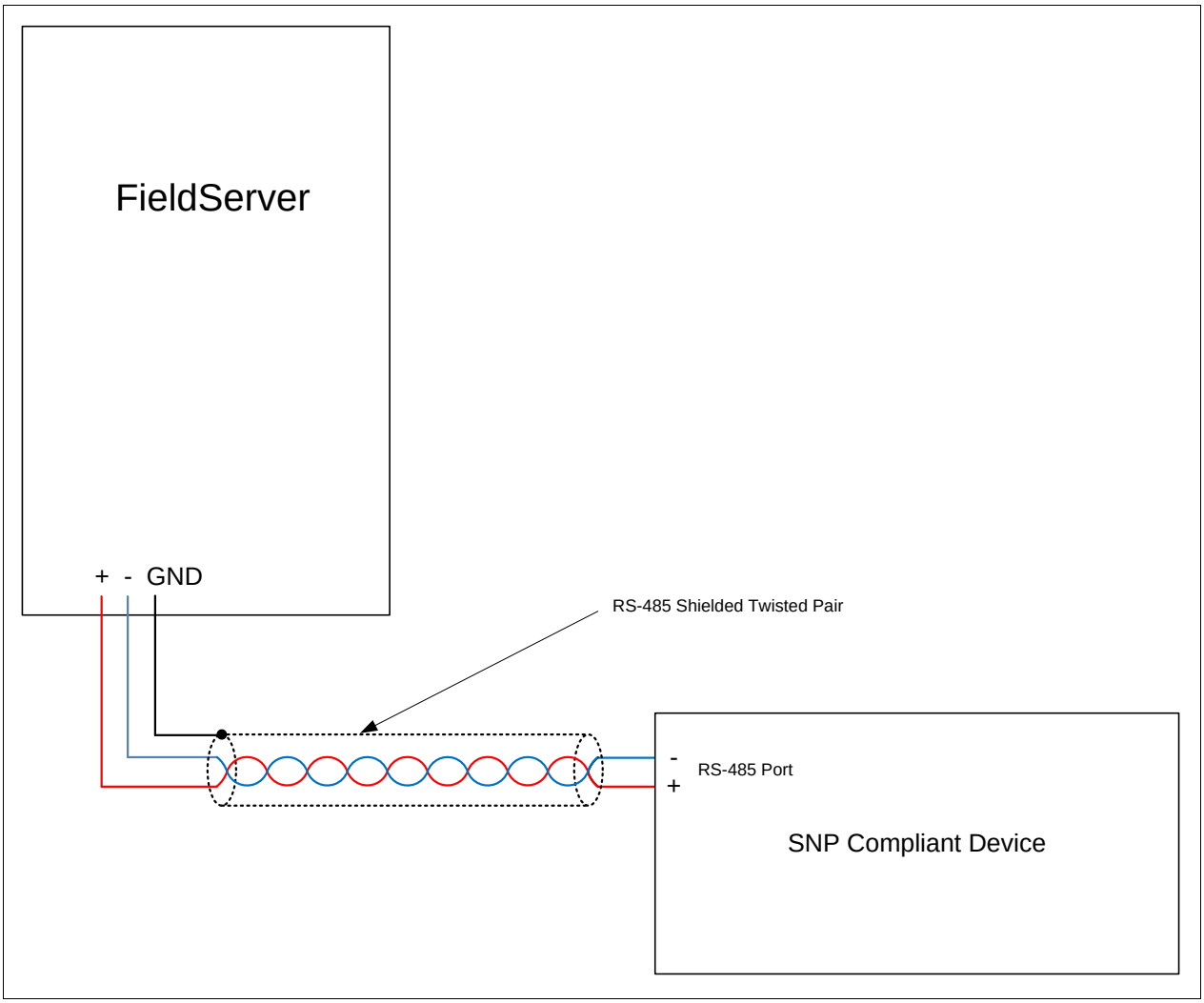
2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by Sierra Monitor Corporation

Part #	Description
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-03	RJ45 to DB9M connector adapter
	Driver Manual

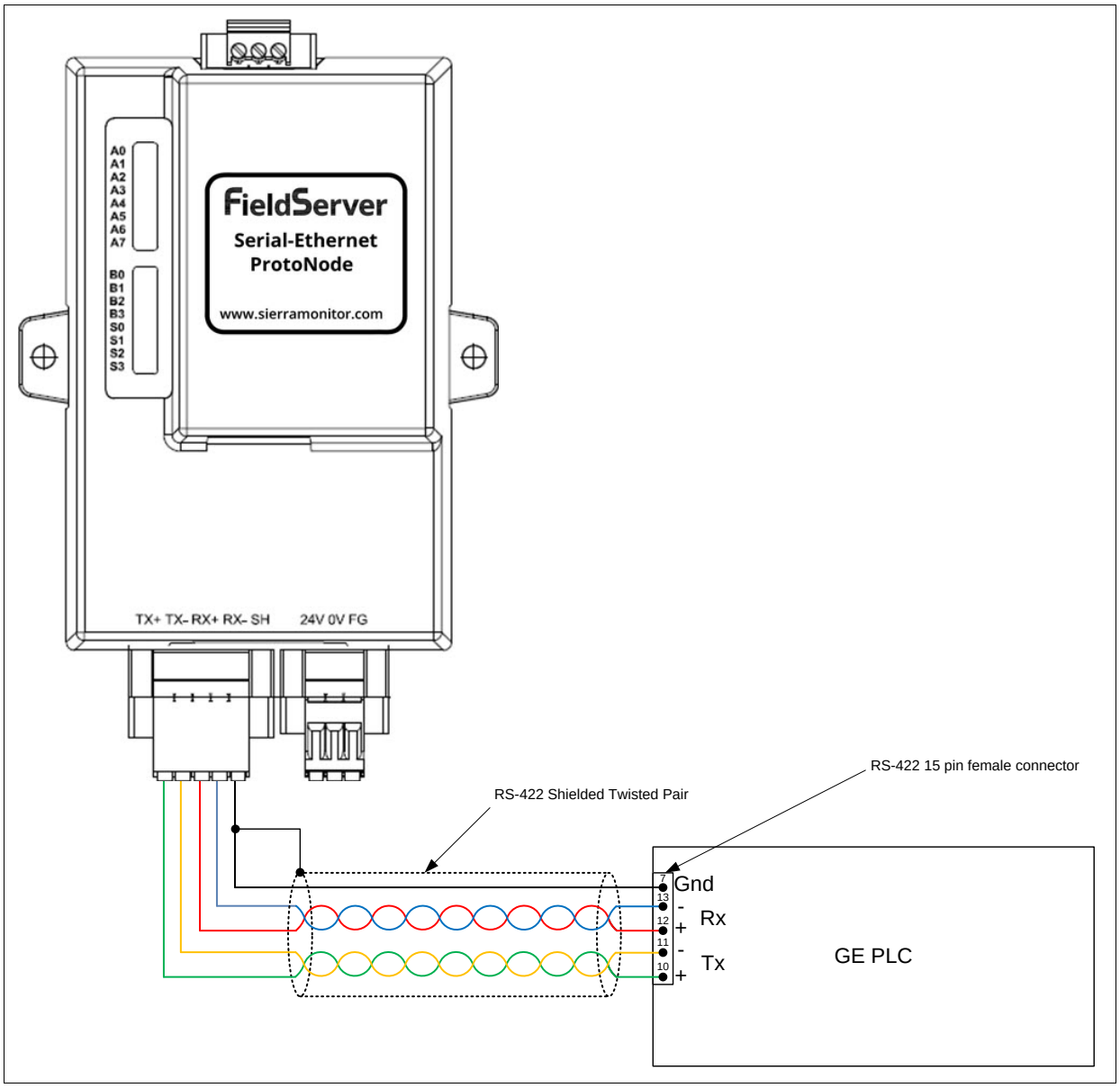
3 HARDWARE CONNECTIONS

3.1 RS-485 Connection Diagram



3.2 RS-422 Connection Diagram

RS-422 is supported by the FS-QS-1030 or the FS-QS-1031 which are available from Sierra Monitor Corporation.



3.3 Connection Notes

If a bridge other than the QuickServer is being used, an external RS-485 to RS-422 converter is required.

4 DATA ARRAY PARAMETERS

Data Arrays are “protocol neutral” data buffers for storage of data to be passed between protocols. It is necessary to declare the data format of each of the Data Arrays to facilitate correct storage of the relevant data.

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, Byte, Uint16, Uint32, Sint16, Sint32
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.	1-10000

Example

```
//  Data Arrays
Data_Arrays
Data_Array_Name   , Data_Array_Format   , Data_Array_Length
DA_AI_01          , UInt16              , 200
DA_AO_01          , UInt16              , 200
DA_DI_01          , BIT                 , 200
DA_DO_01          , BIT                 , 200
```

5 CONFIGURING THE FIELDSERVER AS A GE_SNP CLIENT

For detailed information on FieldServer configuration, 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 FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a GE_SNP Server.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for GE_SNP 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: In the tables below, * indicates an optional parameter, with the bold legal value as default.

5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[1]
Protocol	Specify protocol used.	SNP
Baud*	Specify baud rate.	110 – 19200, standard baud rates only (Vendor limitation), **19200**
Parity*	Specify parity.	Even, **Odd**, None, Mark, Space
Data_Bits*	Specify data Bits.	7, **8**
Stop_Bits*	Specify stop Bits.	**1** (Vendor limitation)
Poll _Delay*	Time between internal polls.	0-32000 seconds, **1 second**
Timeout	The timeout specified for the connection is used as the data link layer timeout. It is also used as the timeout for each Map Descriptor if a timeout is not explicitly defined for a Map Descriptor.	

Example

```
//  Client Side Connections

Connections
Port    , Protocol    , Baud     , Parity      , Poll_Delay
P1      , SNP         , 19200    , Odd         , 0.100s
```

GE Port Defaults			GE Port Capabilities
Parameter	**Series 90-30 PLC's**	**Series 90-70 PLC's**	**Legal Values**
Baud	19,200	19,200	300, 600, 1200, 2400, 4800, 9600, **19200**
Data Bits	8	8	**8**
Parity	Odd	Odd	**Odd**, Even, None
Stop Bits	1	1	**1**, 2
Max Link Idle Time	10 Secs	5 Secs	1 to 60 Secs

[1] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.2 Client Side Node Descriptor Parameters

Section Title		
Nodes		
Column Title	**Function**	**Legal Values**
Node_Name	Provide name for Node. The driver uses the Node name to establish a connection with the Server if the Node_ID is non-zero. If the Node_ID is zero then the driver ignores the Node name when logging onto the PLC and logs on using a Null nameThsu the driver is able to log onto a Node whose name is unknown.	Up to 32 alphanumeric characters
Node_ID*	The parameter is used in conjunction with Node_Name to control how the driver logs onto the PLC.	0,**1**
Protocol	Specify protocol used.	SNP
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[2]
SNP_Longbreak*	Number of characters used to calculate the time for the specified baud rate at which the FieldServer will send a break signal to the PLC.	1-65535, **4**
SNP_Break_Offdelay*	Delay (in milliseconds) before a message is sent to the PLC from the FieldServer after the break is released.	1-65535, **70**
Route*	Provide Destination ID Field.	Period separated four decimal values, **10.16.0.0**

Example

```
//  Client Side Nodes
Nodes
Node_Name   , Node_ID   , Protocol   , Port   , SNP_Longbreak   , SNP_Break_Offdelay
PLC 1       , 1         , SNP        , P1     , 50              , 200
```

[2] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

5.3 Client Side Map Descriptor Parameters

5.3.1 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.	One of the Data Array names from **Section 4**
Data_Array_Offset	Starting location in Data Array.	0 to maximum specified in **Section 4**
Function	Function of Client Map Descriptor.	Rdbc, Wrbc, Wrbx

5.3.2 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 **Section 5.2**
Data_Type	Data type	Discrete Inputs (%I) Discrete Outputs (%Q) Discrete Temporaries (%T) Discrete Internals (%M) Genius Global Data (%G) Analog Inputs (%AI) Analog Outputs (%AQ) Registers (%R) %SA Discrete %SB Discrete %SC Discrete %S Discrete (%S)
Length	Length of Map Descriptor. Length may not exceed the table in the PLC.	1-1000
Address	Starting address of read block / write block. The 1st element of each Data type Table is referred to as address one.	1 , 2 , 3 Positive whole numbers
Format*	BIT tables are normally read by reading whole Bytes at a time (If the Length is 10 then 10 Bytes of data are read and placed in 10 Data Array locations). This parameter can be used to read BITs instead. (If the Length is 10 then 10 BITs are read and each BIT is stored in its own location).	BIT, **Byte**; The format for %AI, %AQ and %R cannot be changed
GE_Func*	This parameter is only specified when a write Map Descriptor is created to change the privilege level.	CPL, **RW**

5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled.	≥0.1s

5.4 Map Descriptor Examples

5.4.1 Simple Read

This example provides a Map Descriptor to read 10 Bytes of Discrete Input states, starting at the very first Discrete Input. The data is stored in a Data Array called DA_DI and the first input is stored at location 100 in the array (101st element). The PLC is polled every 2 seconds.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Node_Name	, Address	, Length	, Scan_Interval	, Data_Type
Read D1	, DA_D1	, 100	, Rdbc	, PLC-1	, 1	, 10	, 2.0	, %1

Map Descriptor names may be used in driver error messages. It is useful to use unique names.

The name of the Data Array in which the driver will store the data. The name must correspond to a Data Array defined in **Section 4**.

Location in the Data Array where the first element of data will be stored. An offset of 100 indicates the 101st element of the array.

RDBC = read block continuous. The driver will read data from the PLC continuously.

The Node as defined in **Section 5.2**. The Node Name connects the Map Descriptor to a Node which in turn connects the Map Descriptor to a port.

The address and length specify the first element and the number of elements that must be read from the PLC's data tables. GE PLC's reference data tables starting at element 1. Unless otherwise specified, the driver reads bytes and words. This Map Descriptor reads bytes of data covering %I1 to %I79.

A Read is performed every 2.0s.

Use the % symbol as you would if you were programming a GE PLC.

5.4.2 Simple Write

This example writes data from the FieldServer Data Array called DA_AO to the PLC identified as NODE1. The write is repeated every 5 seconds. Ten word values are written to the PLC's %AQ Data Table starting at location 20.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Node_Name	, Address	, Length	, Scan_Interval	, Data_Type
Write_AO	, DA_AO	, 0	, Wrbc	, Node1	, 20	, 10	, 5.0	, %AQ

Write continuously.

Write to PLC Table element number 20 to 29 inclusive (10 elements).

Write to the %AQ (Analog Output) table in the PLC.

5.4.3 Handling BITs

This example writes data from the FieldServer Data Array called DA_AO to the PLC identified as NODE1. The write is repeated every 5 seconds. Ten word values are written to the PLC's %AQ Data Table starting at location 20.

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Node_Name	, Address	, Length	, Scan_Interval	, Data_Type	, Format
Read_DI	,DA_T	, 100	, Rdbc	, PLC-1	, 8	, 2	, 2.0	, %T	, Bit

The format parameter tells the driver to override the default data type and read Bits specifically. In this case the Bits are stored in the Data Array in two separate elements at DA_T[100] and DA_T[101].

6 CONFIGURING THE FIELDSERVER AS A GE_SNP SERVER

For detailed information on FieldServer configuration, 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" files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a GE_SNP Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for GE_SNP communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the FieldServer virtual Node(s) needs to be declared in the "Server Side Nodes" section, and the data to be provided to the Client needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

NOTE: In the tables below, * indicates an optional parameter, with the bold legal value as default.

6.1 Server Side Connection Descriptors

Section Title		
Connections		
Column Title	**Function**	**Legal Values**
Port	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2[3]
Protocol	Specify protocol used.	SNP
Baud*	Specify baud rate.	110 – 115200, standard baud rates only
Parity*	Specify parity.	Even, Odd, **None**, Mark, Space
Data_Bits*	Specify data Bits.	7, **8**
Stop_Bits*	Specify stop Bits.	**1**
Poll Delay	Time between internal polls.	0-3200s, **1s**

Example

```
//  Server Side Connections

Connections
Port    , Protocol   , Baud     , Parity
P1      , SNP        , 19200    , Odd
```

[3] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

6.2 Server Side Node Descriptors

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for Node. The driver uses the Node name to establish a connection with the Server if the Node_ID is non-zero. Refer to **Appendix A.1**.	Up to 32 alphanumeric characters
Node_ID	The parameter is used to control how the driver logs onto the PLC.	**0**,1
Protocol	Specify protocol used.	SNP
Port	Port must be defined if the FieldServer is emulating more than one node and polling more than one SNP device.	P1-P2, R1-R2[4]

Example

```
//  Server Side Nodes

Nodes
Node_Name   , Node_ID   , Protocol   , Port
PLC 1       , 1         , SNP        , P1
```

6.3 Server Side Map Descriptors

6.3.1 FieldServer Specific 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.	One of the Data Array names from "Data Array" section above
Data_Array_Offset	Starting location in Data Array.	0 to maximum specified in "Data Array" section above
Function	Function of Server Map Descriptor.	Passive

[4] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

6.3.2 Driver Specific Map Descriptor Parameters

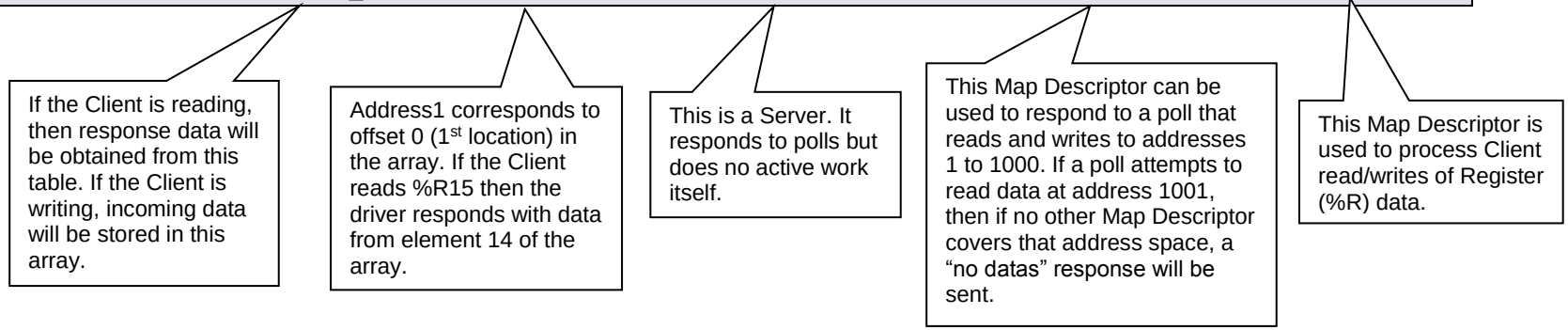
Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from.	One of the Node names specified in "Client Node Descriptor" above
Data_Type	Data type Use one of the Data Types specified in brackets.	Discrete Inputs (%I) Discrete Outputs (%Q) Discrete Temporaries (%T) Discrete Internals (%M) Genius Global Data (%G) Analog Inputs (%AI) Analog Outputs (%AQ) Registers (%R) %SA Discrete %SB Discrete %SC Discrete %S Discrete (%S)
Length	Length of Map Descriptor – may not exceed the table length in the PLC.	1 - 1000
Address	Starting address of read block / write block The 1st element of each Data type Table is referred to as address one.	1 , 2 , 3 Positive whole numbers

6.3.3 Timing Parameters

Column Title	Function	Legal Values
Scada_Hold_Timeout	Specifies time Server side waits before responding to Client that Node is offline on FieldServer Client side.	>1.0s

6.3.4 Map Descriptor Example

Map_Descriptor_Name	, Data_Array_Name	, Data_Array_Offset	, Function	, Node_Name	, Address	, Length	, Data_Type
Server-R-Data	, TABLE_R	, 0	, Passive	, PLC-1	, 1	, 1000	, %R



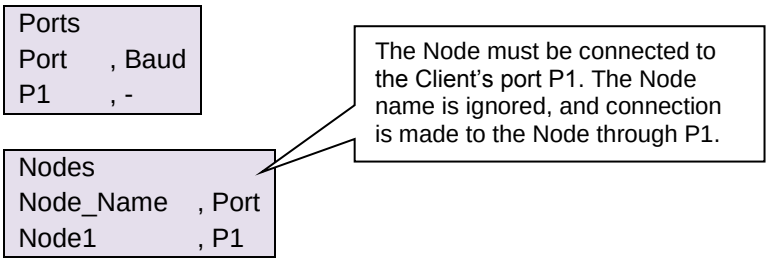
Appendix A. Useful Features

Appendix A.1. SNP Node Names

The following notes describe how the Node_Name is used when the FieldServer is acting as a Server.

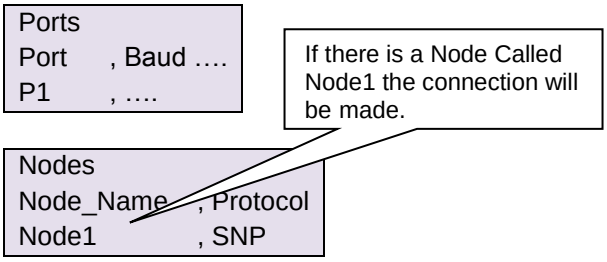
When a Client attempts to establish a connection, it may send the name of the Node it wishes to connect to or a Null Node name. If it sends a Null Node name, then the FieldServer will connect if any SNP Nodes have been defined for the same port as the Client using the Null name. The Node must be connected to a port. Only one SNP node can be connected per port.

Example:



If the Client uses the Node name during a connection attempt, then the FieldServer uses the name to find a matching Node. The name must be an exact match. If a match is found, then the connection can be established. In this case the Node does not have to be tied to a particular port and thus one Node could respond to different Clients polling it on different ports. If this method is used and the Node is tied to a port then the Client must poll on the same port.

Example: Client uses a specific name to connect. In this case provided there is a Node with a matching name the connection will be made



Appendix A.2. Scaling[5]

Scaling is applied by the driver when Bytes and registers are stored.

When BIT data is transferred in BIT format than scaling is not applied. When BIT data is transferred using bye format's then scaling may be applied.

Refer to the FieldServer Configuration Manual for more information on how scaling is handled by the FieldServer.

[5] Valid for driver Version 1.02 and later.

Appendix B. Troubleshooting

- **Every 2nd Message Times Out**
  A possible cause is that the poll frequency is so low that the time between polls is greater than the Max Link Idle Time (of the PLC). If the time is exceeded the connection is closed. Thus when the next poll is received by the PLC it ignores it and hence it times out in the FieldServer. After timing out, the driver re-establishes the connection for the next poll.

- **Very Slow Comms Rate**
  A timer known as the T1 timer is the minimum time that must pass before a response to a message can be sent. Both ends of a SNP connection have their own T1 timer (hard coded in the driver). When a connection is made the timers are exchanged and the larger of the two is used. The drivers T1 timer is set to a very small number. SNP message #6 indicates the size of the T1 timer. You may need to configure the PLC to improve communications.

- **Multidrop RS-485 Networks**
  SNP protocol specifies that there may only be one Master on a network at a time. When using nameless Nodes (Node_ID=0) there may be only one PLC on the connection.

Appendix C. Reference

Appendix C.1. Error Messages

Message	Explanation
SNP:#1 FYI. The MapDesc called <%s> is too short.	The length of the Map Descriptor defined to expose driver statistics must be at least 660.[6]
SNP:#2 FYI. You could have used a MapDesc called <%s> to expose diagnostic info.	You may safely ignore this message. Read 0 for information on how to expose the driver's communication statistics using a Data Array.
SNP:#3 FYI. SNP device requires inter msg timer < timeout. Nid=<%s>.	The driver is required to use the larger of its own or the PLC's T1 (Inter message) timer. This message is produced when the PLC returns a T1 timer that is larger than the driver's timeout. Using this timer will mean that every message will timeout. Increase the timeout for the Node, connection or Map Descriptor as appropriate.[6]
SNP:#4 Error. Cant emulate Node=<%s>	A connection request has been received for the Node named in the error message but this Node has not been defined in the CSV file. The connection has not been established.[6]
SNP:#5 Err. Total Bytes expected/rcvd=%d/%d	This could result from a corrupted message or it may indicate an unexpected/unsupported aspect of the SNP protocol. If the message occurs frequently, take a log and report the error to FieldServer.
SNP:#6 FYI. T1 timer set to <%ld> mSec.	This message requires no action. It reports the T1 timer value selected by the slave in milliseconds. If the value is large then this will slow the communications rate and you may wish to reconfigure the slave.
SNP:#7 FYI. T1 timer reset to <%ld> mSec.	This is the FieldServer slave reporting its T1 timer after a Client has established a connection No action is required.
SNP:8 FYI. Subsequent connect response ignored. NName=<%s>	No action is required. A Client has attempted to re-establish a connection that is already open.
SNP:9 Err. Device declined request. Mjr=%x(h) Min=%x(h) Possible Offending Poll =<%s> addr=%d length=%d	This message uses two lines of the error log. It is printed when a slave declines a request. The major and minor faults are reported. These numbers are printed in hexadecimal. See **Appendix C.3** for further details. The message reports the Map Descriptor which generated the rejected poll, the data type, the address and the number of elements being requested. Often this message will be the result of a poll that requests data past the end of a PLC data table or when the length is based on a BIT count but the poll actually requests Bytes.[6]
SNP:10 Err. Wrong msg type=%d(h) mbox=%x(h)	This message is reported when the driver receives an unexpected SNP message. Record this information, take a log and report the error to FieldServer.

[6] Edit the CSV, correct the problem and reset the FieldServer.

Message	Explanation
SNP:11 Err. Max Tbuf Len. FieldServer cant go bigger than %d Bytes Possible Offending Poll =<%s> addr=%d length=%d	The length of some messages is variable and is dependent on settings or the particular implementation of the SNP. Normally this driver adjusts to use the largest size required by a Client or slave response but this message indicates that the driver's maximum had been reached. The data has been discarded so it is important that the situation be remedied. The data type, address and the number of elements being requested are reported. Two possible solutions exist. Change the Client's communications configuration and reduce the size of the text buffer length to 4096 Bytes. Create two Map Descriptors that each read half the data.
SNP:12 FYI. Max Tbuf Len. FieldServer != Device. Changed FieldServer to %d Bytes	This message requires no response. It indicates that the slave requires a longer text buffer than the driver's default. The driver has adjusted its buffer to the new length indicated.
SNP:13 Err. Wrong msg type=%d(h) mbox=%x(h)	The driver has received an unexpected reply in response to a poll. Make a log, record the values reported in this message and contact FieldServer for support.
SNP:#14 Error. Attach Response. Unexpected !	The driver has been polled by a Client and it has received an unexpected message. If the error occurs repeatedly then make a log, record the values reported in this message and contact FieldServer for support. There are a few variations of this message but they all indicate the same problem.
SNP:#15 FYI. Node Station - forcing to 0	The Node_ID has not been defined for a SNP Node. Read the notes in **Appendix A.1** and in **Sections 5.2** and **6.2**. [7]
SNP:#16 Err. Addresses start at 1. MapDesc=<%s>	GE devices number the elements of data tables starting at element 1.[7]
SNP:#17 Err. MapDesc=<%s> has bad data type.	Refer to **Section 5.3.2** for a list of valid data types.[7]
SNP:#18 Err. MapDesc=<%s> has bad data type.	
SNP:#19 Err. Mapdesc=<%s> has invalid length.	A Map Descriptor must have a length. The minimum length is "1". [7]
SNP:#20 Err. do diagnostic x	If you see this error, report it to FieldServer tech support. The message is produced when a QA diagnostic is produced and should only ever be seen during testing by FieldServer Inc.
SNP:#21 Err. do diagnostic x	
SNP:#22 Err. do diagnostic x	
SNP:#23 Err. Illegal Node_ID [%d] - Set to 1	Legal Node_ID's are 0-65535.[7]
GE_SNP:#24 FYI. Parse Error: Attach: CPU ID is not Null terminated.	The CPU Id is a max 10 characters including the terminating Null character. The driver has found one that isn't correctly formatted. This error may be ignored. It will not affect operation of the driver. It does mean the driver cannot store the name of the connected device in the stats array.

[7] Edit the CSV, correct the problem and reset the FieldServer.

Appendix C.2. Driver Stats

The GE-SNP Serial Driver records statistics differently from the way that FieldServer driver normally records statistics. This difference arises from the fact that this driver is not a simple poll response driver - a single poll can generate a large number of response fragments.

Fragment Ack/Nack messages are NOT counted as messages but the Bytes sent/received are counted. Connection messages are counted as messages and the Bytes sent/rcvd are counted.

This driver can expose these and additional statistics by writing data to a Data Array.

A special Map Descriptor is required. The driver recognizes the Map Descriptor by its name which must be "***SNP-stats***".

Example: Configuration of this special Map Descriptor

```
Nodes
Node_Name          , Protocol
Null_Node          , SNP

Data_Arrays
Data_Array_Name    , Data Format      , Data_Array_Length
SNP_STATS          , UINT32           , 660
```

```
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Node_Name    , Length
snp-stats           , SNP_STATS       , Null_Node    , 660
```

The Driver uses the Data Array SNP_STATS (in this example) to store driver specific statistics. Only one of these Map Descriptors may be specified (per tier) per FieldServer.

The driver stores the following stats for each port. The offset into the Data Array can be found by multiplying the port number by 60.

#	Message	Description
0	GE_STAT_CPU_ID	This is 10 Bytes long. The numbers placed here correspond to the ASCII characters of the responding Node's name.
11	GE_STAT_T1_TIMER	The current value of the T1 timer is reported here. The driver uses the larger of its own or the slave T1 Timer.
12	GE_STAT_MBOX_NAK_MAJOR	Most recent Mailbox Nak error code. See additional table below.
13	GE_STAT_MBOX_NAK_MINOR	
14	GE_STAT_MBOX_NAK_CNT	Mailbox Nak error Count.
15	GE_STAT_MBOX_PROG_NUM	Most recent Piggyback info.
16	GE_STAT_MBOX_SWEEP	
17	GE_STAT_MBOX_PLC_STAT	
18	GE_STAT_MBOX_PRIV_LVL	
19	GE_STAT_NO_NODES	If a poll for a SNP Node is received and there are no Nodes defined or there is no matching Node.
20	GE_STAT_CONNECT_ATTEMPT	Counts the number of time a connection attempt is made
21	GE_STAT_CONNECT_RESPONSE	Counts the number of times the driver responds to connection attempts when acting as a Server.
22	GE_STAT_CONNECT_NO_RESPONSE	Counts the number of connection attempts that timed out.
23	DRV_DLL_CLIENT_SENDS_MSG	Counts the number of connection and poll messages sent by the Client (whole messages only).

#	Message	Description
24	DRV_DLL_CLIENT_SENDS_ACKNAK	Counts the number of simple Ack/Nak's sent by Client. These Ack/Nak's indicate messages if formatted correctly, but the device can not necessarily respond to the request.
25	DRV_DLL_CLIENT_SENDS_BYTES	Byte count includes Ack/Nak's sent and all fragments and connection.
26	DRV_DLL_SERVER_SENDS_MSG	Same as for Client.
27	DRV_DLL_SERVER_SENDS_ACKNAK	
28	DRV_DLL_SERVER_SENDS_BYTES	
29	DRV_DLL_CLIENT_RCVS_MSG	Counts ack/nak's, and all message fragments excluding attach responses.
30	DRV_DLL_CLIENT_RCVS_BYTES	Byte count from above messages.
31	DRV_DLL_SERVER_RCVS_MSG	Counts message fragments and ack/nak's rcvd from Client but excludes Client responses to Server fragments.
32	DRV_DLL_SERVER_RCVS_BYTES	Byte count for above messages
33	DRV_DLL_TIMEOUT	Count of timeouts caused by no response or ic_timeout at the DLL layer.
34	DRV_DLL_ERROR	Count of the number of times that the DLL layer went into an error state.
35	DRV_DLL_ERROR_CODE	The error code from the most recent DLL error is stored here.
36	GE_STAT_DRV_ACT_NODE_OFFLINE_R	Corresponds to slave driver actions in responding to a poll. The polled Node is offline.
37	GE_STAT_DRV_ACT_NO_MAPDESC_R	Corresponds to slave driver actions in responding to a poll. There is no Map Descriptor that can be used to process the poll from the Client.
38	GE_STAT_DRV_ACT_NO_DATA_R	Corresponds to slave driver actions in responding to a poll. There is no data available.
39	GE_STAT_DRV_ACT_NO_NODE_R	Corresponds to slave driver actions in responding to a poll. This should never happen as a connection would have been refused.
40	GE_STAT_DRV_ACT_NAK_R	The slave cannot respond to a poll for one of the reasons above.
41	GE_STAT_DRV_ACT_NORMAL_R	The Server has responded to a poll normally.
42	GE_STAT_CD_UPD_CHECK	A message is invalid because the checksum failed.
43	GE_STAT_CD_UPD_FUNCTION	A message is invalid because the message type is not known.
44	GE_STAT_CD_UPD_NO_START	A message is invalid because it begins with the wrong codes.
45	GE_STAT_CD_UPD_PROTO	A message is invalid because it is unexpected in its current context or some other reason.
46	GE_STAT_CLIENT_SENDS_FRAG	Counts the number of read/write message fragments sent by the Client.
47	GE_STAT_CD_UPD_IC_TIMEOUT	Counts the number of times that an IC timeout error occurred at the DLL layer.
48	GE_STAT_CD_UPD_MSG_IGNORED	Counts the number of times a message was ignored. A message will be printed to the error log with the reason.
49	GE_STAT_CD_UPD_NAK	Counts the number of times that the slave received a short nak from the Client.
50	GE_STAT_CD_UPD_STREAMING	Counts the number of times the DLL layer's buffer was too full to process additional incoming data.

Appendix C.3. Server Response NAK, Major and Minor Error Codes[8]

When the Server responds with a MailBox NAK it now allocates different Major and Minor error codes to differentiate the NAK reasons.

Error Code	NAK Reason	Description
0x05 0xbe	No Such Node	The request Node is not known by the Server.
0x05 0xbd	No Data	The Data age is greater than the Cache_Age setting.
0x05 0xbc	Other Problems	Eg. No Server Map Descriptors defined for requested data.

Appendix C.3.1. Major Error Status Codes

Error Status	Description
01h Illegal Service Request	Either not defined or not supported.
02h Insufficient Privilege	Minor status field contains the privilege level required for the service request.
04h Protocol Sequence Error	The CPU has received a message that is out of order.
05h Service Request Error	Minor status field contains the request specific error code. See table of Minor Error Status Codes below.
06h Illegal Mailbox Type	Service request mailbox type is either undefined or unexpected.
07h The PLC CPU's Service Request Queue is full. The master should retry later.	It is recommended that the master wait a minimum of 10 ms before sending another service.

Appendix C.3.2. Minor Error Status Codes

Decimal	Hex	Description
-1	0FF	Service request has been aborted.
-2	0FE	No privilege for attempted operation.
-3	0FD	Unable to perform auto configuration.
-4	0FC	I/O configuration is invalid.
-5	0FB	Cannot clear I/O configuration.
-6	0FA	Cannot replace I/O module.
-7	0F9	Task address out of range.
-8	0F8	Invalid task name referenced.
-9	0F7	Required to log in to a task for service.
-10	0F6	Invalid sweep state to set.
-11	0F5	Invalid password.
-12	0F4	Invalid input parameter in request.
-13	0F3	I/O configuration mismatch.
-14	0F2	Invalid program cannot log in.
-15	0F1	Request only valid from programmer.
-16	0F0	Request only valid in stop mode.
-17	0EF	Programmer is already attached.
-18	0EE	Could not return block sizes.
-19	0ED	VMEbus error encountered.
-20	0EC	Task unable to be created.
-21	0EB	Task unable to be deleted.
-22	0EA	Not logged in to process service request.

[8] Valid for driver Version 1.02 and later.

Decimal	Hex	Description
-23	0E9	Segment selector not valid in context.
-24	0E8	No user memory is available to allocate.
-25	0E7	Configuration is not valid.
-26	0E6	CPU model number does not match.
-27	0E5	DOS file area not formatted.
-28	0E4	Segment for this selector does not exist.
-29	0E3	CPU revision number does not match.
-30	0E2	IOS could not delete configuration or bad type.
-31	0E1	No I/O configuration to read or delete.
-32	0E0	Service in process cannot login.
-33	0DF	Invalid Datagram connection address.
-34	0DE	Size of Datagram connection invalid.
-35	0DD	Unable to locate given connection ID.
-36	0DC	Unable to find connection address.
-37	0DB	Invalid segment selector in Datagram.
-38	0DA	Null pointer to data in segment selector.
-39	0D9	Transfer type invalid for this selector.
-40	0D8	Point length not allowed.
-41	0D7	Invalid Datagram type specified.
-42	0D6	Datagram connection boundary exceeded.
-43	0D5	Invalid block name specified in Datagram.
-44	0D4	Mismatch of configuration checksum.
-45	0D3	User Program Module (UPM) read or write exceeded block end.
-46	0D2	Invalid write mode parameter.
-47	0D1	Packet size or total program size does not match input.
-48	0D0	One or more PLC modules configured have unsupported revision.
49	0CF	Specified device is unavailable in the system (not present).
-50	0CE	Specified device has insufficient memory to handle request.
-51	0CD	Attempt was made to read a device but no data has been stored on it.
-52	0CC	Data stored on device has been corrupted and is no longer reliable.
-53	0CB	A comm or write verify error occurred during save or restore.
-54	0CA	Device is write protected.
-55	0C9	Login using non-zero buffer size required for block commands.
-56	0C8	Password(s) already enabled and cannot be forced inactive.
-57	0C7	Passwords are set to inactive and cannot be enabled or disabled.
-58	0C6	Control Program (CP) tasks exist but requestor not logged into main CP.
-59	0C5	No task-level Rack/Slot configuration to read or delete.
-60	0C4	Verify with FA Card or EEPROM failed.
-61	0C3	Text length does not match traffic type.
-62	0C2	The OEM key is NULL (inactive).
-63	0C1	Invalid block state transition.
-64	0C0	Bad OMF record checksum in store.
-65	0BF	Illegal OMF record type/data contents.
-66	0B	Bad Block Type given in Load/Store.
-67	0BD	Block Set (subblock name) not found.
-68	0BC	Block Type (e.g., data) not found.
-69	0B	Maximum length of a partial store exceeded.
-70	0B	Block Set already exists, cannot create.

Decimal	Hex	Description
-71	0B9	Executable flag in TYPDEF record not set.
-72	0B8	Size of the Segment Selector Table in TYPDEF record is not correct.
-73	0B7	Segment length in Verify not equal to the segment length of block in the PLC.
-74	0B6	Cyclic Redundancy Check (CRC) checksum comparison in Verify failed.
-75	0B5	Additive checksum comparison in Verify failed.
-76	0B4	Attempt to alter interrupt list in MAIN DECL BLOCK during RUN MODE.
-77	0B3	Length limit exceeded; includes read past end of transferred data, writes past of program block.
-78	0B2	Program block already exists and cannot be replaced.

Appendix C.3.3. Minor Error Status Codes: Program Load and Store Requests

Decimal	Hex	Description
80	50	Problem with sending mail to the slave Service Request task. (Series 90-70 PLC CPUs only).
81	51	Problem with getting mail from the slave Service Request task. (Series 90-70 PLC CPUs only).
85	55	Slave SNP task timed out before receiving SRP response. (Series 90-70 PLC CPUs only).
86	56	Slave SNP task could not find the requested Datagram connection. (Series 90-70 PLC CPUs only).
87	57	Slave SNP task encountered an error in trying to write the Datagram. (Series 90-70 PLC CPUs only).
88	58	Slave SNP task encountered an error in trying to update the Datagram. Series 90-70 PLC CPUs only).