



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-111 PSP

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after February 2012

Driver Version: 1.03
Document Revision: 2

TABLE OF CONTENTS

1	PSP Description	3
2	Driver Scope of Supply	3
2.1	Supplied by FieldServer Technologies for this driver.....	3
3	Hardware Connections.....	3
1	Data Array Parameters.....	4
4	Configuring the FieldServer as a PSP Client	5
4.1	Client Side Connection Descriptions.....	5
4.2	Client Side Node Descriptors	6
4.3	Client Side Map Descriptors.....	7
4.3.1	<i>FieldServer Related Map Descriptor Parameters.....</i>	<i>7</i>
4.3.2	<i>Driver Related Map Descriptor Parameters.....</i>	<i>7</i>
4.3.3	<i>Timing Parameters.....</i>	<i>7</i>
4.4	Map Descriptor Example – Read Data from the Server.....	8
4.5	Map Descriptor Example - Use of PSP_CMD	9
4.5.1	<i>Update Node_Id (PSP_Cmd = ND).....</i>	<i>9</i>
4.5.2	<i>Update Baud Rate (PSP_Cmd = BD).....</i>	<i>9</i>
4.5.3	<i>Read or Change the Status of a Node.....</i>	<i>9</i>
5	Configuring the FieldServer as a PSP Server	10
5.1	Server Side Connection Descriptors	10
5.2	Server Side Node Descriptors	10
5.3	Server Side Map Descriptors.....	10
Appendix A. Troubleshooting		11
Appendix A.1. Driver Error Messages		11
Appendix A.2. Driver Stats		12

1 PSP DESCRIPTION

ProtoCessor Simple Protocol (PSP) is an ASCII based protocol which provides a simple way to communicate with ProtoCessor. The Serial PSP driver allows the FieldServer to transfer data to and from devices over RS-232 or RS-485 using PSP protocol. The FieldServer can emulate either a Server or Client. The Client side has been developed primarily to test the Server side driver as part of FieldServer's QA program.

As a Server, the Driver responds to read/write requests and can update remote device configurations. The Server can accept or serve data values in decimal or hex format as determined by the Client. The Server responds with an error number when for any reason it is unable to successfully respond to a valid request.

As a Client, the driver polls to read or write data. The Driver can receive or send data values in decimal or in hex format. The Driver can also poll the Server to change upstream node or connection parameters such as Node_ID or Baud rate.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Single client per port.
Server	0	Protocol is Node-less. Server serves ports.

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8915-10	UTP cable (7 foot) for RS-232 use

3 HARDWARE CONNECTIONS

The ProtoCessor is integrated into the customer's hardware at design time. No additional connection information is required.

1 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, UInt16, Sint16, 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_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
```

4 CONFIGURING THE FIELDSEVER AS A PSP 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 FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a PSP Server. The Client side has been developed primarily to test the Server side driver as part of FieldServer's QA program

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for PSP 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.

4.1 Client Side Connection Descriptions

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ¹
Protocol	Specify protocol used	PSP, Protoessor_Simple
Baud*	Specify baud rate	38400
Parity*	Specify parity	Even, Odd, None
Data_Bits*	Specify data bits	7 , 8
Stop_Bits*	Specify stop bits	1 , 2
Poll_Delay*	Time between internal polls	0-32000s, 1s

Example

```
// Client Side Connections

Connections
Port , Protocol , Baud , Parity , Poll_Delay
P8 , PSP , 38400 , None , 0.100s
```

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.

4.2 Client Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	PSP station address of physical server node	1-255
Protocol	Specify protocol used	PSP.
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ¹

Example

```
// Client Side Nodes

Nodes
Node_Name , Node_ID , Protocol , Connection
HOST 1 , 1 , PSP , P1
```

4.3 Client Side Map Descriptors

4.3.1 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 1.
Data_Array_Offset	Starting location in Data Array	0 to (Data_Array_Length-1) as specified in Section 1.
Function	Function of Client Map Descriptor	Rdbc, Wrbc, Wrbx

4.3.2 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 4.2
Length	Length of Map Descriptor	1 (Single item)
PSP_Data_Format ^{*2}	Format of Data value to or from Server.	Hex, D, U, F, -
Serv_Data_Array*	Name of target Data Array at Server. Required for Write thru operations.	Up to 32 alphanumeric characters, Data_Array_Name specified in Section 1.
Serv_Array_Offset*	Target Data Array Offset at Server. Required for Write thru operations.	0-32767, Data_Array_Offset specified in Section 1.
PSP_Cmd*	ND - update field protocol Node_ID BD - update field protocol Baud Rate Node_Status - get status of field protocol Node Node_Enabled - enable field protocol Node Node_Disabled - disable field protocol Node Refer to Section 4.5.3 for more information.	ND, BD, Node_Status, Node_Enabled, Node_Disabled, -.

4.3.3 Timing Parameters

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

² Hex = Hexadecimal Integer, D = Signed Decimal Integer, U = Unsigned Integer, F = Floating Point.

4.4 Map Descriptor Example – Read Data from the Server

This Map Descriptor reads data from the Server.

```
// Client Side Map Descriptors
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , PSP_Data_Format , Lengthn , Scan_Interval , Serv_Data_Array , Serv_Array_Offset
A1 , DA_1 , 0 , Rdbc , Node_A , Hex , 1 , 1s , DA_serv , 0
```

Client stores returned data in this Data Array.

Client stores returned data at this Offset.

One of the Nodes declared under Section 4.2. This reflects the physical server's communication configuration.

Always 1, as only one value can be read in a single transaction.

Name of target Data Array at Server - if not specified Data_Array_Name will be used.

Forces the driver to perform a read poll for every Scan_Interval time period.
If WRBC, the driver will issue a write command with the same parameters

Optional parameter - specifies the format in which the Client expects the data value to be formatted by the Server.
If "Hex" returned value will be considered as hex representation. If "-" or parameter not defined returned value will be considered as decimal value coded as ASCII characters.

Time period between polls.

Target Data Array offset at Server - if not specified Data_Array_Offset will be used.

4.5 Map Descriptor Example - Use of PSP_CMD

4.5.1 Update Node_Id (PSP_Cmd = ND)

The following Map Descriptor can be configured to update the Node_Id of a Node belonging to the Non-PSP protocol.

Map_Descriptors						
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Node_Name	Function	Length	PSP_Cmd
MDCLI_1	DA_1	0	NODE_01	Wrbx	1	ND

4.5.2 Update Baud Rate (PSP_Cmd = BD)

The following Map Descriptor can be configured to update the Baud rate of a Node belonging to the Non-PSP protocol.

Map_Descriptors						
Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Node_Name	Function	Length	PSP_Cmd
MDCLI_1	DA_1	0	NODE_01	Wrbx	1	BD

4.5.3 Read or Change the Status of a Node

The following Map Descriptors can be configured to read the status of or disable/enable a Node belonging to the Non-PSP protocol.

Map_Descriptors								
Map_Descriptor_Name	Scan_Interval	Data_Array_Name	Data_Array_Offset	Node_Name	Function	Length	PSP_Data_Format	PSP_Cmd
MD_NS	2.0s	DA_NS	0	NODE_01	Rdbc	1	-	Node_Status
MD_ND	-	DA_ND	0	NODE_01	WRBX	1	-	Node_Disabled
MD_NE	-	DA_NE	0	NODE_01	WRBX	1	-	Node_Enabled

5 CONFIGURING THE FIELDSEVER AS A PSP SERVER

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 FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a PSP Client. The Driver responds to Client’s read/write requests.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for PSP communications, the driver independent FieldServer buffers need to be declared in the “Data Arrays” section, and the data to be provided to the clients needs connections to be declared in the “Server Side Connections” section. Details on how to do this can be found below.

The Server responses with an error number when for any reason it is unable to successfully respond to a valid request. See Appendix A.1 for detail

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

5.1 Server Side Connection Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 ³
Protocol	Specify protocol used	PSP, Protoceessor_Simple
Baud*	Specify baud rate	38400
Parity*	Specify parity	Even, Odd, None
Data_Bits*	Specify data bits	7, 8
Stop_Bits*	Specify stop bits	1, 2

Example

```
//      Server Side Connections

Connections
Port   , Protocol   , Baud   , Parity
P1    , PSP        , 38400  , None
```

5.2 Server Side Node Descriptors

This section is not relevant to this driver.

5.3 Server Side Map Descriptors

This section is not relevant to this driver.

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.

Appendix A. Troubleshooting

Appendix A.1. Driver Error Messages

Some configuration errors produce an error every time a poll is generated. This would fill the error buffer quickly without adding any clarity. For this reason the driver suppresses subsequent similar messages and it is possible for the same error produced by multiple Map Descriptors to produce only one error message. Subsequent error messages can be seen on the driver message screen.

Note : In the actual message %d is replaced by an integer, %s by text indicating a Data Array name or Map Descriptor name and %x by two hex characters.

Error Message	Description and Action Required
PSP:#01 ERR. Cannot store at %u da %s DA length %u	This error will be generated when the driver tries to store data but the corresponding Data Array is not long enough. Set the Data Array length accordingly. ⁴
PSP:#02 FYI. Response with unknown err. MD=%s",md->mapdesc_name	The Server has sent an error message with a number that is unknown to the Client. This is for information only. Check the indicated Map Descriptor to determine which request generated this response.
PSP:#03 FYI. Response has error=%04d MD=%s",err_num,md->mapdesc_name	The Server has sent an error message. See 0 for the description corresponding to the error number.
PSP:#04 FYI. do diagnostic 1	The Client is polling the Server with the first part of the request only.
PSP:#05 FYI. do diagnostic 2	The Client is polling the Server with the second part of the request only.
PSP:#06 FYI. do diagnostic 3 waived, No Checksum	The Client is polling with the wrong checksum, but the action will be ignored as protocol does not support checksum.
PSP:#07 FYI. do diagnostic 4	You are forcing the client to do a poll proceeded with a few garbage characters.
PSP:#32 FYI. You could use an Array called <%s>to expose diagnostic info	You could use an Array named "psp-stats" to expose stats - error numbers sent by the Server will be stored in this array. Refer to 0

⁴The CSV file needs to be edited, downloaded to the FieldServer and the FieldServer reset for the changes to take effect.

Appendix A.2. Driver Stats

In addition to the standard FieldServer operating statistics the driver exposes certain key stats in a Data Array if required. A Server device can then monitor these stats. The following must be added to the configuration file to activate these stats.

```
// Expose Driver Operating Stats.

Data_Arrays
Data_Array_Name , Data_Format , Data_Array_Length
psp-stats      , UINT32      , 200
```

Length must be sufficient to store error messages.

Number	Description
0001	Data Array does not exist
0002	Data Array Offset does not exist
0003	Illegal Format code for FMT - keyword
0004	Corrupted message received.
0005	Configuration update command did not processed successfully (e.g. Baud or Node_ID update)
0006	Data object is expired or offline

The portion of the Data Array used will depend upon the port being used for PSP communication. The following table shows the offset where any error number will be stored according to the port in use.

Port	Offset where Error Number is Stored							
	0001	0002	0003	0004	0005	0006
P1	1	2	3	4	5	6
P2	51	52	53	54	55	56
P3	101	102	103	104	105	106
P4	151	152	153	154	155	156
P5	201	202	203	204	205	206
P6	251	252	253	254	255	256
P7	301	302	303	304	305	306
P8	351	352	353	354	355	356
R1	401	402	403	404	405	406
R2	451	452	453	454	455	456