Chipkin
Automation Systems

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8705-12
Federal Signal Ultravoice - Electronic Siren Controllers

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

© **Chipkin Automation Systems**, Box 211, 3495 Cambie St, Vancouver, BC, Canada, V5Y 4R3
Tel: (866) 383-1657, **Fax:** (416) 915-4024
Email: dfs@chipkin.com **Website:** www.chipkin.com

TABLE OF CONTENTS

TABLE OF CONTENTS **2**

1. Federal Signal Ultravoice (FSU) - Electronic Siren Controllers (ESC) Serial Driver Description **3**

2. Driver Scope of Supply **5**
2.1. Supplied by FieldServer Technologies for this driver 5
2.2. Provided by the Supplier of 3rd Party Equipment 5
2.2.1. Required 3rd Party Hardware *5*
2.2.2. Required 3rd Party Software *5*
2.2.3. Required 3rd Party Configuration *5*
2.2.4. Optional Items *5*

3. Hardware Connections **6**
3.1. Hardware Connection Tips / Hints 6

4. Configuring the FieldServer as a FSC - Electronic Siren Controllers Serial Driver Client **7**
4.1. Data Arrays/Descriptors 7
4.2. Client Side Connection Descriptions 8
4.3. Client Side Node Descriptors 9
4.4. Client Side Map Descriptors 10
4.4.1. FieldServer Related Map Descriptor Parameters *10*
4.4.2. Driver Related Map Descriptor Parameters *10*
4.4.3. Timing Parameters *11*
4.4.4. Map Descriptor Example 1 – Read Status. *12*
4.4.5. Map Descriptor Example 2 – Sending Commands to the controller *14*
4.4.6. Map Descriptor Example 3 – Sending Commands to the controller *15*
4.5. Interpreting the Status Report values found in the Data Arrays 17

5. Configuring the FieldServer as a FSC - Electronic Siren Controllers Serial Driver Server **21**

Appendix 1. Advanced Topics **22**

Appendix 2. Troubleshooting tips **23**
2.1.1. Connection Tips & Hints 23
2.1.2. Driver Error Messages 24
2.1.3. Exposing Driver Stats 28

Appendix 3. Revision History **30**

1. Federal Signal Ultravoice (FSU) - Electronic Siren Controllers (ESC) Serial Driver Description

The FSU - Electronic Siren Controllers (ESC) Serial Driver allows the FieldServer to transfer data to and from devices over RS232 using Federal Signal Ultravoice - Electronic Siren Controllers Serial Driver protocol.

The FieldServer can emulate a Client. As a client the driver can poll for status information and send commands to the FSU controller.

The driver is a serial driver using a RS232 serial port to connect between the FieldServer and the CHC-MF. An RS485 port together with a converter can also be used for the connection.

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	1	Only 1 FSU-ESC node per connection
Server	0	Not supported or documented.

Supported Functions
ARM
CANCEL
VOICE
DISARM
ZONEA-D
REPORT
MSG_A-P
WAIL
P_WAIL
A_WAIL
STEADY
P_STEADY
A_STEADY
PHASE+-
LOWPWR
CODE01-CODE50

Status Items Monitored with 'Report' Function
Siren Type
Function State (Code running)
Unit ID
Amplifier status for each amp in
the unit depending on siren type
Audio A
Audio B
Master Current
Battery
Charger
AC Power
Control Box Intrusion
Battery Box Intrusion
False Alarm/Local Activation
Rotation

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
-	No specific cables are shipped with this driver. A generic RJ45 Ethernet cable must be shipped with this driver.
-	A generic male and Female connector kit must be shipped with this driver.
FS-8709-12	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

No special configuration of the CHC interface is required.

2.2.4. Optional Items

PART #	Vendor/Manufacturer	Description

3. Hardware Connections

The FieldServer is connected to the the serial port of the controller. Only 3 wires are connected.

Xmit
Receive
Ground.

3.1. Hardware Connection Tips / Hints

This section is blank.

4. Configuring the FieldServer as a FSC - Electronic Siren Controllers Serial Driver 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 FSC - Electronic Siren Controllers Serial Driver Server.

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 FSC - Electronic Siren Controllers 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 CHC. Each connection can only be used to connect to a single CHC interface.

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	FSU_ESC
Baud*	Specify baud rate	Driver Supports : 110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 28800; 38400; 57600; 115200 Baud Vendor Equipment Supports : 1200 Baud
Parity*	Specify parity	Driver Supports : Odd, Even, **None** Vendor Equipment Supports: **None**
Data_Bits*	Specify data bits	Driver Supports : 7,**8** Vendor Equipment Supports : 8
Stop_Bits*	Specify stop bits	Driver Supports : 1,2 Vendor Equipment Supports: 1
Handshaking*	Specify hardware handshaking	**None**
Poll _Delay*	Time between internal polls	0-32000 seconds, **0.2 second**

[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.

Example

```
//      Client Side Connections

Connections
Port,       Protocol,   Baud,   Parity,   Data_Bits,   Stop_Bits
P1,         FSU_ESC,    1200,   Even,     8,           1
```

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.	1-258
Protocol	Specify protocol used	FSU_ESC
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[2]

Example

```
//  Client Side Nodes

Nodes
Node_Name       Node_ID     Protocol    Connection
SirenNode       1           FSU_ESC     P1
```

[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.

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	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 Client Map Descriptor	RDBC, WRBC, WRBX Passive_Client

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	Data type This commonly used parameter is not used by this driver.	Register, Coil, AI, DI
Length	Length of Map Descriptor Tells the driver how much space in the Data Array is reserved for this function	Set Length = 34 for report Reading and equal to 1 for all other Map Descriptos
Address	This commonly used FieldServer parameter is not used by this protocol.	
FSU_ESC_Function_Name	Used to define the command to be sent to the controller	GENERIC ARM CANCEL VOICE Q_TEST DISARM ZONE REPORT RESET

		MSG WAIL P_WAIL A_WAIL STEADY P_STDY A_STDY AUX PHASE PHASE PHASE LOWPWR CODE

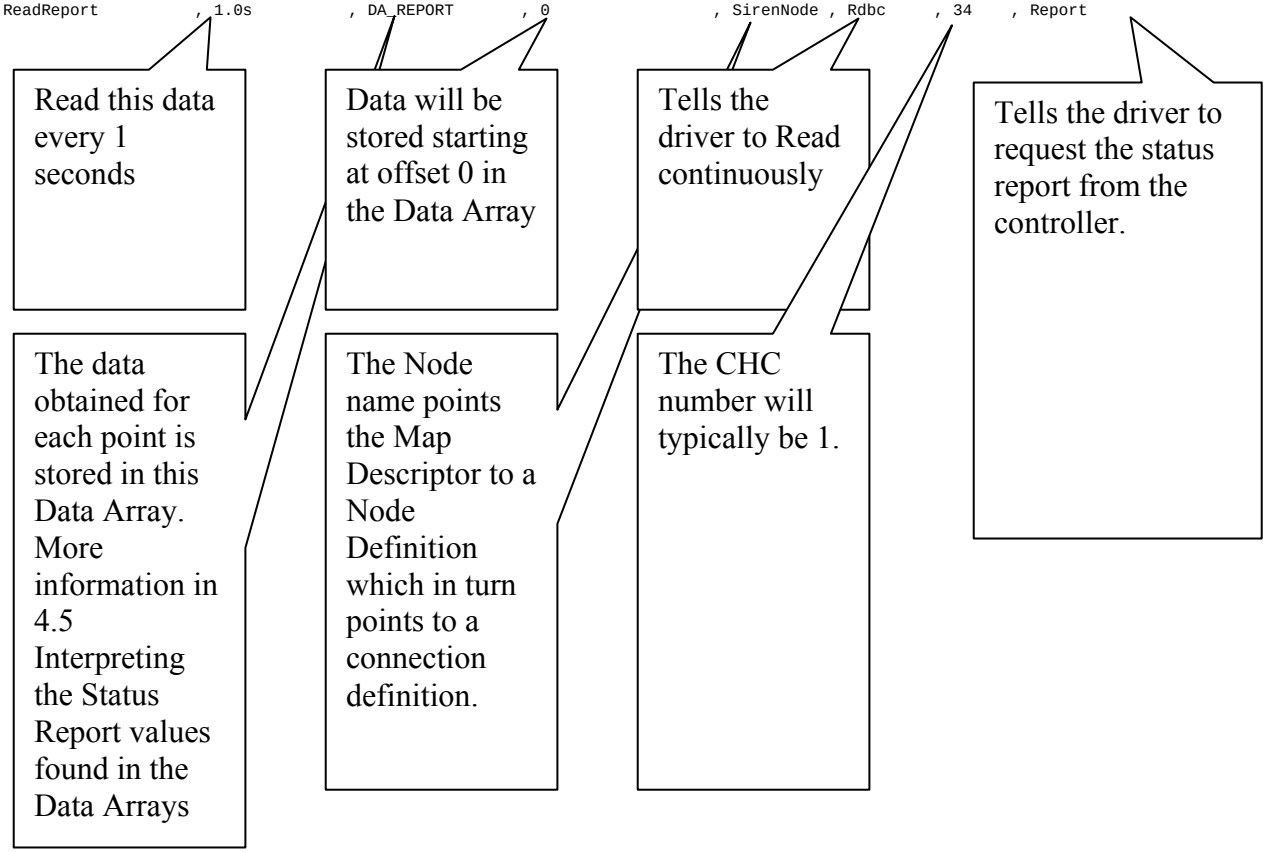
4.4.3. Timing Parameters

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

4.4.4. Map Descriptor Example 1 – Read Status.

In this example the driver reads status data from the controller. It reads the data every 1 second because the `Scan_Interval` has been set to 1 seconds and the `function` has been set to RDBC – Read Block Continuous.

```
Map_Descriptors
Map_Descriptor_Name , Scan_Interval , Data_Array_Name , Data_Array_Offset , Node_Name , Function , Length, FSU_ESC_Function_Name
ReadReport          , 1.0s          , DA_REPORT       , 0                 , SirenNode , Rdbc     , 34    , Report
```



4.4.5. Map Descriptor Example 2 – Sending Commands to the controller

This example is typical for all commands except for 'Phase', 'Msg', 'Code' and 'Zone'

Each time one of these commands is executed the driver sends the appropriate message to the controller. In this example, the command wait to be triggered by some other protocol updating the specified Data Array offset in the relevant Data Array. We have called the Data Array 'Not Used' for the reason that the driver does not extract any values from the Data Array to send in the message to the controller. Rather, the Data Array is used to trigger the commands. For example, each time the FieldServer's other protocol writes to DA_NOT_USED[5] the driver will send a reset command to the controller. The value written doesn't have to change to trigger the command. It is the update of the Data Array element that triggers the command.

```
Map_Descriptors
Map_Descriptor_Name ,Scan_Interval ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,FSU_ESC_Function_Name ,Length
Arm                 ,1.0s          ,DA_NOT_USED     ,0                 ,wrbx     ,NodeA     ,Arm                    ,1
Cancel              ,1.0s          ,DA_NOT_USED     ,1                 ,wrbx     ,NodeA     ,Cancel                 ,1
voice               ,1.0s          ,DA_NOT_USED     ,2                 ,wrbx     ,NodeA     ,voice                  ,1
Q_test              ,1.0s          ,DA_NOT_USED     ,3                 ,wrbx     ,NodeA     ,q_test                 ,1
Disarm              ,1.0s          ,DA_NOT_USED     ,4                 ,wrbx     ,NodeA     ,disarm                 ,1
Reset               ,1.0s          ,DA_NOT_USED     ,5                 ,wrbx     ,NodeA     ,reset                  ,1
Wail                ,1.0s          ,DA_NOT_USED     ,6                 ,wrbx     ,NodeA     ,wail                   ,1
p_wail              ,1.0s          ,DA_NOT_USED     ,7                 ,wrbx     ,NodeA     ,p_wail                 ,1
a_wail              ,1.0s          ,DA_NOT_USED     ,8                 ,wrbx     ,NodeA     ,a_wail                 ,1
steady              ,1.0s          ,DA_NOT_USED     ,9                 ,wrbx     ,NodeA     ,steady                 ,1
p_stdy              ,1.0s          ,DA_NOT_USED     ,10                ,wrbx     ,NodeA     ,p_stdy                 ,1
a_stdy              ,1.0s          ,DA_NOT_USED     ,11                ,wrbx     ,NodeA     ,a_stdy                 ,1
Aux                 ,1.0s          ,DA_NOT_USED     ,12                ,wrbx     ,NodeA     ,aux                    ,1
LowPwr              ,1.0s          ,DA_NOT_USED     ,13                ,wrbx     ,NodeA     ,LowPwr                 ,1
Phase+              ,1.0s          ,DA_NOT_USED     ,14                ,wrbx     ,NodeA     ,Phase+                 ,1
Phase-              ,1.0s          ,DA_NOT_USED     ,15                ,wrbx     ,NodeA     ,Phase-                 ,1
```

4.4.6. Map Descriptor Example 3 – Sending Commands to the controller

This example is typical for 'Phase', 'Msg', 'Code' and 'Zone' commands.

These commands need a value from the Data Array to know exactly what command to send. For example is the command is configured to send a 'CODE' command. The driver looks in the data array, extracts the value – 5 for example and then sends CODE05 command. For zones valid Data Array values are 1-4 for ZONEA-ZONED. For Msgs, valid Data Array values are 1-16 for MSG_A to MSG_P. For codes, valid Data Array values are 1-50 for CODE01-CODE50. For phases, valid Data Array values are 1 or 2 for PHASE+ or PHASE-.

Each time one of these commands is executed the driver sends the appropriate message to the controller. In this example, the commands wait to be triggered by some other protocol updating the specified Data Array offset in the relevant Data Array. The value written doesn't have to change to trigger the command. It is the update of the Data Array element that triggers the command.

```
Map_Descriptors
Map_Descriptor_Name ,Scan_Interval ,Data_Array_Name ,Data_Array_Offset ,Function ,Node_Name ,FSU_ESC_Function_Name ,Length
Phase               ,1.0s          ,DA_PHASE       ,0                 ,wrbx     ,NodeA     ,Phase                  ,1
Msg                 ,1.0s          ,DA_MSG         ,0                 ,wrbx     ,NodeA     ,Msg                    ,1
Code                ,1.0s          ,DA_CODE        ,0                 ,wrbx     ,NodeA     ,Code                   ,1
Zone                ,1.0s          ,DA_ZONE        ,0                 ,wrbx     ,NodeA     ,Zone                   ,1
```

If another protocol writes the value 1 to DA_CODE[0] then the driver will send a single CODE01 command. It will do this each time some other protocol writes the value 1 to DA_CODE[0]. If the value written is invalid then no command will be sent.

4.5. Interpreting the Status Report values found in the Data Arrays

The offsets specified in the table below are relative to the offset specified on the Map Descriptor.

Table 1: How Status Report Data is stored.

Offset	Meaning	Notes
1	Unit Type	See Table 2
2	Function State	See Table 3
3	Unit Number	As a decimal Number
4	Sensor status - amps 1-4	As Ascii Char - See Table 4
5	Sensor status - amps 5-8	As Ascii Char - See Table 4
6	Sensor status - amps 9-12	As Ascii Char - See Table 4
7	Sensor Status A	As Ascii Char - See Table 4
8	Sensor Status B	As Ascii Char - See Table 4
9	Sensor Status C	As Ascii Char - See Table 4
10	Status - Amp 1	0=Ok 1=Bad or Inactive
11	Status - Amp 2	0=Ok 1=Bad or Inactive
12	Status - Amp 3	0=Ok 1=Bad or Inactive
13	Status - Amp 4	0=Ok 1=Bad or Inactive
14	Status - Amp 5	0=Ok 1=Bad or Inactive
15	Status - Amp 6	0=Ok 1=Bad or Inactive
16	Status - Amp 7	0=Ok 1=Bad or Inactive
17	Status - Amp 8	0=Ok 1=Bad or Inactive
18	Status - Amp 9	0=Ok 1=Bad or Inactive
19	Status - Amp 10	0=Ok 1=Bad or Inactive
20	Status - Amp 11	0=Ok 1=Bad or Inactive
21	Status - Amp 12	0=Ok 1=Bad or Inactive
22	Battery	0=Ok 1=Bad or Inactive
23	Master Current	0=Ok 1=Bad or Inactive
24	Audio B	0=Ok 1=Bad or Inactive
25	Audio A	0=Ok 1=Bad or Inactive
26	Intrusion Cabinet 1	0=Ok 1=Bad or Inactive
27	Not Used	0=Ok 1=Bad or Inactive
28	AC Power	0=Ok 1=Bad or Inactive
29	Charger	0=Ok 1=Bad or Inactive
30	Rotation	0=Ok 1=Bad or Inactive
31	Not Used	0=Ok 1=Bad or Inactive
32	Spare	0=Ok 1=Bad or Inactive
33	False Alarm	0=Ok 1=Bad or Inactive

Table 2: Unit Type

Data Array Value	ASCII Equivalent	Meaning
48	0	MOD6024 & MOD6048
49	1	Not defined
50	2	Not defined
51	3	Not defined
52	4	MOD1004
53	5	MOD2008
54	6	MOD3012
55	7	EOWS-612
56	8	MOD4016
57	9	MOD5020
42	*	Custom

Table 3: Function State

Data Array Value	ASCII Equivalent	Meaning
48	0	Wail
49	1	Pulsed Wail
50	2	Alt. Wail
51	3	Steady
52	4	Pulsed Steady
53	5	Alt. Steady
54	6	Aux
55	7	Alarm
56	8	Quiet Test
57	9	Cancel
58	:	Public Address
59	;	Armed
60	<	Standby
61	=	Digital Voice
65	A	Code 1
66	B	Code 2
67	C	Code 3
68	D	Code 4
69	E	Code 5
70	F	Code 6
71	G	Code 7
72	H	Code 8
73	I	Code 9
74	J	Code 10
75	K	Code 11
76	L	Code 12
77	M	Code 13
78	N	Code 14
79	O	Code 15

80	P	Code 16
81	Q	Code 17
82	R	Code 18
83	S	Code 19
84	T	Code 20
85	U	Code 21
86	V	Code 22
87	W	Code 23
88	X	Code 24
89	Y	Code 25
90	Z	Code 26
91	[	Code 27
92	\	Code 28
93	]	Code 29
94	^	Code 30
95	_	Code 31
96	`	Code 32
97	a	Code 33
98	b	Code 34
99	c	Code 35
100	d	Code 36
101	e	Code 37
102	f	Code 38
103	g	Code 39
104	h	Code 40
105	i	Code 41
106	j	Code 42
107	k	Code 43
108	l	Code 44
109	m	Code 45
110	n	Code 46
111	o	Code 47
112	p	Code 48
113	q	Code 49
114	r	Code 50

Table 4: Status Codes

Data Array Value	ASCII Equivalent	Meaning
49	1	0001
50	2	0010
51	3	0011
52	4	0100
53	5	0101
54	6	0110
55	7	0111
56	8	1000
57	9	1001
48	0	1010
42	*	1011
35	#	1100
65	A	1101
66	B	1110
67	C	1111
68	D	0000

5. Configuring the FieldServer as a FSC - Electronic Siren Controllers Serial Driver Server

This driver has a server side implemeted but it is used for FieldServer's Quality Assurance program and is not docuemnted or supported. If you are interested in using Sever Side features then please contact Chipkin Automation Systems.

Appendix 1. Advanced Topics

This section is blank.

Appendix 2. Troubleshooting tips

2.1.1. Connection Tips & Hints

This section is blank.

2.1.2. Driver Error Messages

Error Message We have shown place holders for the parts of the message which change. %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.	**Explanation and corrective action**
FSU_ESC:#1 Err FSU_ESC_Function='%s' is unknown	The driver has been configured incorrectly. The Map Descriptor parameter *FSU_ESC_Function_Name* must be specified correctly. Review Chapter 4 and the examples. Correct the configuration and then download the corrected configuration file and reset the FieldServer for the changes to take effect.
FSU_ESC:#2 Err. The Function Name must be specified.	See Error #1
FSU_ESC:#3 Err. Zone command out of range. %s[%d]Value=%d Where %s[%d] specifies a Data Array and offset.	Valid Entries in the Data Array are 1-4 for zones A-D The most likely cause for this message is either the driver has started up and there are zero's in the Data Arrays or the other protocol has written a number to the FieldServer that is out of the valid range. You can preload values into Data Arrays using the configuration file. The FieldServer configuration manual has notes on how to do that.
FSU_ESC:#4 Err. MSG command out of range. %s[%d]Value=%d Where %s[%d] specifies a Data Array	Valid Entries in the Data Array are 1-16 for Msgs A-P The most likely cause for this message is

and offset.	either the driver has started up and there are zero's in the Data Arrays or the other protocol has written a number to the FieldServer that is out of the valid range. You can preload values into Data Arrays using the configuration file. The FieldServer configuration manual has notes on how to do that.
FSU_ESC:#5 Err. MSG command not recognized. (%s)	The Message command speficied in the configuration file must be 'MSG' you cannot specify 'MSG_A' or any other format. Read example 3 in chapter 4 for example on how to configure this command. Correct the configuration and then download the corrected configuration file and reset the FieldServer for the changes to take effect.
FSU_ESC:#6 Err. PHASE command out of range. %s[%d]Value=%d Where %s[%d] specifies a Data Array and offset.	Valid Entries in the Data Array are 1 or 2 for PHASE+ or PHASE- The most likely cause for this message is either the driver has started up and there are zero's in the Data Arrays or the other protocol has written a number to the FieldServer that is out of the valid range. You can preload values into Data Arrays using the configuration file. The FieldServer configuration manual has notes on how to do that.
FSU_ESC:#7 Err. CODE command out of range. %s[%d]Value=%d Where %s[%d] specifies a Data Array and offset.	Valid Entries in the Data Array are 1-50 for CODE01-CODE50 The most likely cause for this message is either the driver has started up and there are zero's in the Data Arrays or the other protocol has written a number to the FieldServer that is out of the valid range. You can preload values into Data Arrays using the configuration file. The

	FieldServer configuration manual has notes on how to do that.
FSU_ESC:#8 Err. Unknown Function Code=%s	The CODE command speficied in the configuration file must be 'CODE' you cannot specify 'CODE01' or any other format. Read example 3 in chapter 4 for example on how to configure this command. Correct the configuration and then download the corrected configuration file and reset the FieldServer for the changes to take effect.
FSU_ESC:#9x Err. Diagnostic	If any of these messages are printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#10 Err. DA=%s is too short. Min length=%d	The Data Array is too short to store the data from the Status report. Correct the configuration and then download the corrected configuration file and reset the FieldServer for the changes to take effect.
FSU_ESC:#11 Err. Cmd=%s not recognized.	Read example 3 in chapter 4 for example on how to configure this command.
FSU_ESC:#12 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#13 Err. Cmd=%s not recognized	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#14 Err. Recieved Cmd=%s. Require DA with name=%s to Store	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#15 Err. Cmd=%s not recognized.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#16 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#17 Err. Cmd=%s not	If this message is printed please take a

recognized.	log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#18 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#19 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#20 Err. Recieved Cmd=%s. Require DA with name=%s to Store.	If this message is printed please take a log, send the log to Tech Support using email and follow up with a call.
FSU_ESC:#21 FYI. Use an Array called <%s> to expose diagnostic info.	Read Chapter 2.1.3 Exposing Driver Stats
FSU_ESC:#22 Report: %s	The driver prints this message every time it receives a status report from the controller. No action is required and the message can be ignired.
FSU_ESC:#23 Invalid Report Character=%d(dec)=0x%02x=%c	If a status message is not correctly composed this message is printed. If it is printed frequently then this probably indicates noise on the line. Please take a log, send the log to Tech Support using email and follow up with a call.

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 an upstream device. 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,
fsu-esc-stats,          UINT32,             1000,
```

Offset	Description
1	Incremented each time the driver tries to send a ZONE command but the ZONE number is invalid.
2	Incremented each time the driver tries to send a MSG command but the MSG number is invalid.
3	Incremented each time the driver tries to send a MSG command but the MSG number is invalid.
4	Incremented each time the driver tries to send a PHASE command but the the value in the Data Arra wasn't a 1 or 2.
5	Incremented each time the driver doesn't recognize the command to be sent
6	Increments each time a "REPORT" query is sent to the controller.
7	Total number of bytes sent to the controller for Report Queries.
8	Increments each time a command is sent to the controller (Writes).
9	Total number of bytes sent to the controller for messages that command the controller.
10	Number of times the driver processed a map descriptor but could not send a poll because the command was invalid.
11	Number of times the driver timed out trying to send a command to the controller.
12	Number of times the drivers's receive buffer overflowed. Buffer overflows occur when the driver receives bytes but cant recognize the messages so it cant clear them out.
13	Number of times the 1st or last characters in a Status report are invalid.
14	The number of times a complete response to the Status Report query are received.
15	Number of times a complete and valid response to the Status Report query are received.
16	Number of times a complete but invalid response to the Status Report query are

Offset	Description
	received.
17	Increments each time a Report msg is sent and no response is received within the timeout period.
18	Increments each time Error Msg #10 is printed.

Appendix 3. Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
14 Aug 2008	PMC		1.00a	0	Document Created