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

Driver Manual
(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after August 2019.

Driver Version:	1.01
Document Revision:	3.C

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 FEDERAL SIGNAL ULTRAVOICE (FSU) – ELECTRONIC SIREN CONTROLLERS (ESC) DESCRIPTION

The FSU – Electronic Siren Controllers (ESC) Serial Driver allows the FieldServer to transfer data to and from devices over RS-232 using Federal Signal Ultravoice – Electronic Siren Controllers Serial 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 RS-232 serial port to connect between the FieldServer and the CHC-MF. An RS-485 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 needed, contact Sierra Monitor to discuss your requirements.

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

NOTE: The FSU needs to be configured in a specific way before it will communicate with a FieldServer. More information is provided in the Section 3.

NOTE: The driver expects the CD (Carrier Detect) LED to be on. This may be the case if a radio is connected or if the jumper labeled 'CAR DET on the UV Control Card is on. (The jumper was known as JP5 in versions of the card that predate rev H).

2 DRIVER SCOPE OF SUPPLY

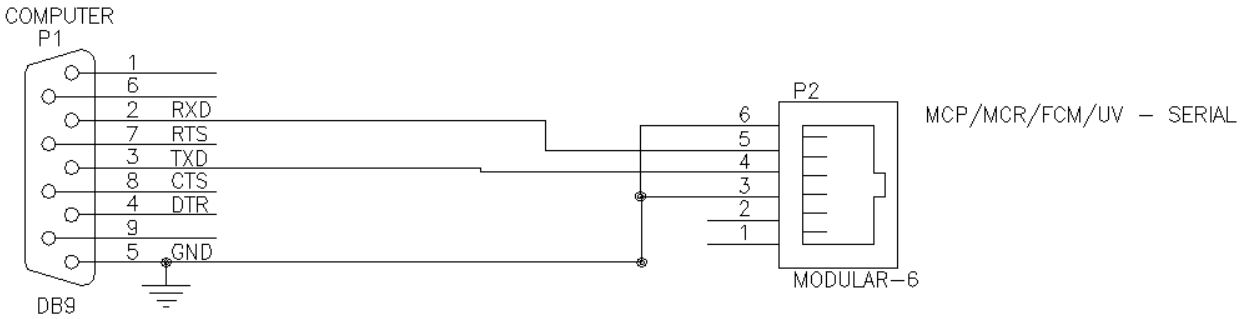
2.1 Supplied by Sierra Monitor Corporation

Part #	Description
	RJ45 Ethernet cable
	Male and female connector kit

3 HARDWARE CONNECTIONS

1. The first step in establishing communication between the FieldServer and the Ultravoice panel is to change the communication setting in the panel, using federal signals programming software.

   a. Two things are needed to change those settings a copy of Federal Signals programming software (these instructions are for SFCDWare) and a RJ11 to DB9 serial connector. Federal Warning System's Customer Care Center is available at 1-800-524-3021 or http://www.federalwarningsystems.com to get a copy of their software and more importantly a key to unlock that software. A schematic of the RJ11 to DB9 Connector for Serial Communication is shown below.



   b. Build the connector shown above and connect the RJ11 end to the port labeled RS232 on the UV panel and the DB9 end to the computer. If these connections are good the software should be able to communicate with the panel in the next step. AMP#5 – 555042-3 RJ11 connectors made a huge difference in the quality of the connection on our panel.

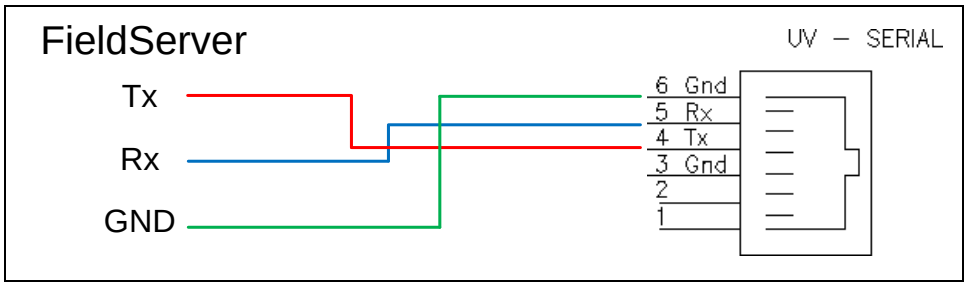
   c. To change the settings, perform the following:

      i. Under System Setup the DTMF 2-Way System needs to be checked. The other setting should be fine as defaulted. Mode=modem, com port=com 1, front porch=1000 ms, and # Tries=1. Then Save.

      ii. Under RTU then Configuration check that the site number is correct. The DIP switch address is on the Rx module. The card needs to be pulled out of the panel to see it. Once the site number is correct press RTU Configuration and check all the above.

      iii. The panel should update and acknowledged messages should appear after each step.

2. The second step is to connect the panel to the FieldServer using a Cat-5E patch cable with a RJ11 connector on the panel end and direct wire to pin connection on the FieldServer end.



3.1 Hardware Connection Tips

The AMP#5 – 555042-3 connectors made a much better connection than any other RJ11 connectors tried. Without these connectors, communication was sporadic at best unless the connector was sitting just right.

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 FSC – ELECTRONIC SIREN CONTROLLERS SERIAL CLIENT

For detailed information on FieldServer configuration, refer to the FieldServer instruction 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 FSC - Electronic Siren Controllers Serial Driver Server.

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 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 and the bold legal values are default.

5.1 Client Side Connection Parameters

NOTE: 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.	R1-R2[1]
Protocol	Specify protocol used.	FSU_ESC
Baud*	Specify baud rate.	Driver: 110; 300; 600; 1200; 2400; 4800; **9600**; 19200; 28800; 38400; 57600; 115200 Vendor Equipment: 1200
Parity*	Specify parity.	Driver: Odd, Even, **None** Vendor Equipment: **None**
Data_Bits*	Specify data bits.	Driver: 7, **8** Vendor Equipment: 8
Stop_Bits*	Specify stop bits.	Driver: 1, 2 Vendor Equipment: 1
Poll _Delay*	Time between internal polls.	0-32000 seconds, **0.2 second**

Example

```
//  Client Side Connections

Connections
Port    , Protocol    , Baud    , Parity    , Data_Bits    , Stop_Bits    , Poll_Delay
R1      , FSU_ESC     , 1200    , Even      , 8            , 1            , .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.

5.2 Client Side Node Parameters

NOTE: 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.	R1-R2[2]

Example

```
//  Client Side Nodes

Nodes
Node_Name   , Node_ID   , Protocol    , Port
SirenNode   , 1         , FSU_ESC     , R1
```

5.3 Client Side Map Descriptor Parameters

5.3.1 FieldServer Related Map Descriptor Parameters

Section Title		
Map_Descriptor		
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

[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.

5.3.2 Driver Related Map Descriptor Parameters

Section Title		
Map_Descriptor		
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	This commonly used driver parameter is not used by the driver.	
Address	This commonly used driver parameter is not used by the driver.	
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 Descriptors
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

5.3.3 Timing Parameters

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

5.4 Map Descriptor Examples

5.4.1 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.

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Length
ReadReport            , DA_REPORT         , 0                   , RDBC       , SirenNode    , 34
```

```
, Scan_Interval   , FSU_ESC_Function_Name
, 1.0s            , Report
```

Example comments:

- Data_Array_Name – The data obtained for each point is stored in this Data Array. More information can be found in **Section 5.5**.
- Data_Array_Offset – Data will be stored starting at offset 0 in the Data Array.
- Function – Tells the driver to Read continuously.
- Node_Name – The Node name points the Map Descriptor to a Node Definition which in turn points to a connection definition.
- Length – The CHC number will typically be 1.
- Scan_Interval – Read this data every 1 seconds.
- FSU_ESC_Function_Name – Tells the driver to request the status report from the controller.

5.4.2 Sending Commands to the Controller

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 waits 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.

```
//    Client Side Map Descriptors

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

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.

```
//    Client Side Map Descriptors

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

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.

5.5 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.

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

Unity Type Values		
Data Array Value	**ASCII Equivalent**	**Meaning**
48	0	MOD6024 & MOD6048
49	1	Not defined
50	2	
51	3	
52	4	MOD1004
53	5	MOD2008
54	6	MOD3012
55	7	EOWS-612
56	8	MOD4016
57	9	MOD5020
58	:	MOD6024/48
59	;	EIWS
61	=	UV-0
64	@	UV-1
65	A	UV-2
66	B	UV-3
67	C	UV-3R
68	D	UV-4
69	E	UV-5
70	F	UV-6
71	G	UV-7
72	H	UV-8
73	I	UV-9
74	J	UV-10
75	K	UV-11
76	L	UV-12
77	M	UV-13
78	N	UV-14
79	O	UV-15
80	P	UV-16

Function State Values		
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

Function State Values		
Data Array Value	**ASCII Equivalent**	**Meaning**
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

Sensor Status Values	
Data Array Value	**Meaning**
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
0	0000

6 CONFIGURING THE FIELDSERVER AS A FSC – ELECTRONIC SIREN CONTROLLERS SERIAL SERVER

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

Appendix A. Troubleshooting

Appendix A.1. Driver Error Messages

The following placeholders are found in the table below in place of error message text where appropriate.

%s is a placeholder for a text string.

%d is a placeholder for a number.

%c is a placeholder for an alpha character.

Error Messages	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 **Section 5.3** for 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. See the FieldServer Configuration Manual for more information.
FSU_ESC:#4 Err. MSG command out of range. %s[%d]Value=%d Where %s[%d] specifies a Data Array and offset.	Valid Entries in the Data Array are 1-16 for Msgs A-P 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. See the FieldServer Configuration Manual for more information.
FSU_ESC:#5 Err. MSG command not recognized. (%s)	The Message command specified in the configuration file must be 'MSG' you cannot specify 'MSG_A' or any other format. See **Section 5.4.2** for examples 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. See the FieldServer Configuration Manual for more information.

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. See the FieldServer Configuration Manual for more information.
FSU_ESC:#8 Err. Unknown Function Code=%s	The CODE command specified in the configuration file must be 'CODE' you cannot specify 'CODE01' or any other format. See **Section 5.4.2** for examples 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.	See **Section 5.4.2** for examples 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 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:#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.	See **Appendix A.2**.
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 ignored.
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.

Appendix A.2. Exposing Driver Operating Statistics

In addition to the standard FieldServer operating statistics the driver exposes certain key stats in a Data Array if required. These stats can then be monitored by an upstream device. Add the following to the configuration file to activate these stats.

```
// Expose Driver Operating Stats

Data_Arrays
Data_Array_Name    , Data_Format    , Data_Array_Length
Fsu-esc-stats      , UINT32         , 1000
```

Stat Relative Offset	Notes
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 value in the Data Array 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' receive buffer overflowed. Buffer overflows occur when the driver receives bytes but can't 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 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.