



A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-128 Safetran SCS-128

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version:	1.00
Document Revision:	6

TABLE OF CONTENTS

1. SAFETRAN-SCS-128 DESCRIPTION 3

2. DRIVER SCOPE OF SUPPLY 4

 2.1. Supplied by FieldServer Technologies for this driver..... 4

 2.2. Provided by the Supplier of 3rd Party Equipment 4

 2.2.1. *Required 3rd Party Configuration*..... 4

3. HARDWARE CONNECTIONS..... 5

4. CONFIGURING THE FIELDSEVER AS A SAFETRAN-SCS-128 CLIENT 6

 4.1. Data Arrays/Descriptors 6

 4.2. Client Side Connection Descriptions 7

 4.3. Client Side Node Descriptors 7

 4.4. Client Side Map Descriptors..... 8

 4.4.1. *FieldServer Related Map Descriptor Parameters*..... 8

 4.4.2. *Driver Related Map Descriptor Parameters*..... 8

 4.4.3. *Timing Parameters* 8

 4.5. Map Descriptor Example: Read Indications..... 9

 4.6. Map Descriptor Example: Write Controls 10

5. CONFIGURING THE FIELDSEVER AS A SAFETRAN-SCS-128 SERVER..... 11

 5.1. Server Side Connection Descriptors 11

 5.2. Server Side Node Descriptors..... 12

 5.3. Server Side Map Descriptors 13

 5.3.1. *FieldServer Specific Map Descriptor Parameters*..... 13

 5.3.2. *Driver Specific Map Descriptor Parameters* 13

 5.3.3. *Map Descriptor Example*..... 14

APPENDIX A. ADVANCED TOPICS 15

 Appendix A.1. Data Format..... 15

1. Safetran-SCS-128 Description

The Safetran-SCS-128 driver allows the transfer of data to and from devices over RS-232 SCS-128 protocol. Vital Harmon Logic Controller (VHLC) supports SCS-128 protocol. SCS-128 80938-098 Rev A04 is the basis of the Safetran-SCS-128 driver. The FieldServer can act as a Client or a Server.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 virtual client per port allowed on multidrop systems, which can poll 127 Server devices
Server	127	This is a protocol limit, maximum devices can be polled by Client

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
FS-8917-04	RJ45 to DB25F connector adapter
FS-8700-128	Driver Manual.

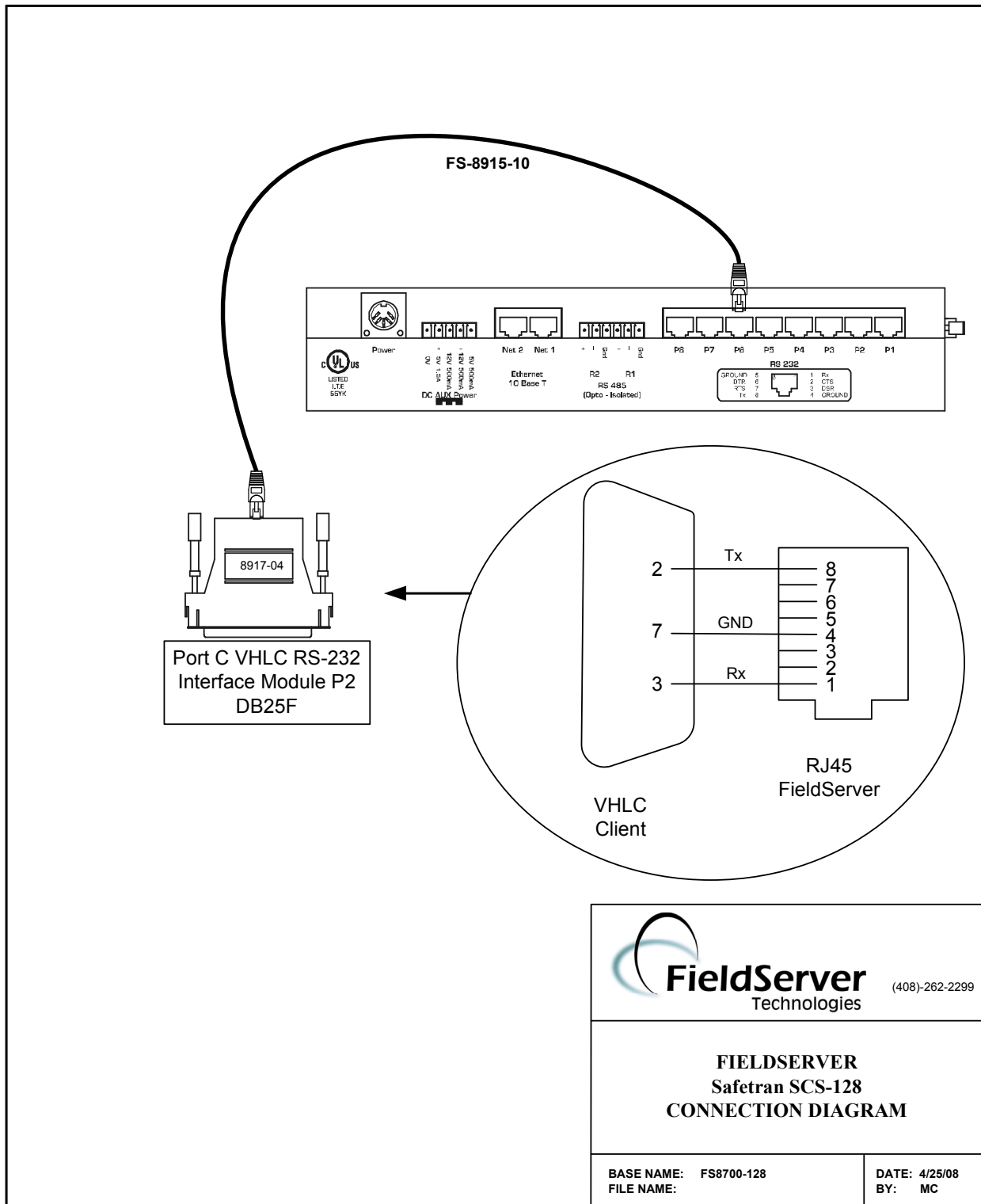
2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Configuration

If the FieldServer is configured as a Server, then the SCS-128 client must be configured to use 'recall' for indication status polling.

3. Hardware Connections

The FieldServer is connected to the VHLC as shown in connection drawing.



4. Configuring the FieldServer as a Safetran-SCS-128 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 Safetran-SCS-128 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 Safetran-SCS-128 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, 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_DI_01,             Bit,                     256
DA_DO_01,             Bit,                     256
DA_8I_01,             Byte,                    32
DA_8O_01,             Byte,                    32
DA_16I_01,            UInt16,                  16
DA_16O_01,            UInt16,                  16
DA_32I_01,            UInt32,                  8
DA_32O_01,            UInt32,                  8
```

4.2. 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 ¹
Protocol	Specify protocol used	SCS-128 SCS128 Safetran SCS-128
Baud*	Specify baud rate	75, 150, 300 ,600, 1200, 2400, 9600 (Vendor limitation)
Parity*	Specify parity	Even (Vendor limitation)
Data_Bits*	Specify data bits	8 (Vendor limitation)
Stop_Bits*	Specify stop bits	1 (Vendor limitation)
Handshaking*	Specify hardware handshaking	None
Poll_Delay*	Time between internal polls	0-32000 seconds, 0.05 seconds

Example

```
// Client Side Connections

Connections
Port,          Protocol,      Baud,      Parity,      Handshaking,      Poll_Delay
P8,           SCS-128,      9600,      Even,        None,             0.100s
```

4.3. 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	SCS-128 station address of physical Server Node	1-127
Protocol	Specify protocol used	SCS-128 SCS128 Safetran SCS-128
Connection	Specify which port the device is connected to the FieldServer	P1-P8 ¹
SCS128_Ack*	If ON, Ack will be sent to Server upon receipt of every indication response	OFF, ON

Example

```
// Client Side Nodes

Nodes
Node_Name,      Node_ID,      Protocol,      Connection,      SCS128_Ack
NODE_A,         1,           SCS-128,      P8,             OFF
```

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.

4.4. 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

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
SCS128_Poll_Type*	Specify Poll_type 'Recall' or 2 always poll for current indications 'Poll' or 1 request for indications if changed 'Controls' or 3 writes controls to target device 'Cutoff' or 5 or 6 send 'cutoff' command to target device	For Read Map Descriptors : Recall, 2 , Poll, 1, For Write Map Descriptors : Controls, 3 Cutoff, 5, 6
Data_Type*	'Flag' - each byte of Indication statuses will be stored in 8 consecutive locations in the Data Array. On write; one Control byte will be composed from 8 consecutive locations. 'Byte'- each byte of Indication statuses will be stored as an 8 Bit Integer at the specified location. On Write; one Control byte will be composed from a single location. 'Register' - two bytes of Indication statuses will be stored as a 16 Bit integer at the specified location. On Write; two Control bytes will be composed from a single location. '32-Bit_Reg' four bytes of Indication statuses will be stored as 32 Bit integers at the specified location. On Write, four Control bytes will be composed from a single location. Refer to Appendix A	Flag , Byte, Register, 32-Bit_Reg
Length*	Length of Map Descriptor Number of indications to store or number of controls to write as per Data_Type.	Flag: 8-256, 16 Byte: 1-32, 2 Register: 1-16, 1 32-Bit_Reg: 1-8, 1

4.4.3. Timing Parameters

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

4.5. Map Descriptor Example: Read Indications

The following Map Descriptors will read 256 indication statuses from physical Server Node_A (described in Client Side Node Descriptors section) after every 1 second. One such Map Descriptor is required for each Node.

Map_Descriptors	Data_Array_Name	Data_Array_Offset	Function	Node_name	Data_Type	SCS128_Poll_Type	Length	Scan_Interval
CMD_DI1,	DA_DI,	0,	RDBC,	Node_A,	Flag,	Recall,	256,	1
CMD_DI2,	DA_8I_01,	0,	RDBC,	Node_A,	Byte,	Poll,	32,	1
CMD_DI3,	DA_16I_01,	0,	RDBC,	Node_A,	Register,	2,	16,	1
CMD_DI4,	DA_32I_01,	0,	RDBC,	Node_A,	32-Bit_Reg.	1,	8,	1

Name of one of the Data Arraya defined in the Data_Array section. Indication statuses will be stored in this Data Array

Starting offset in Data Array to hold 1st indication status

One of the Nodes declared in Node Section. Indications will be read from the station address belonging to this Node

Number of consecutive locations in the Data array to be used for storing data from the Server.

Frequency (in seconds) to read Indication Statuses

4.6. Map Descriptor Example: Write Controls

The following Map Descriptor will write 256 controls to physical Server Node_A (described in Client Side Node Descriptors section) whenever any of the control statuses changes/updates. One such Map Descriptor is required per Node.

```
// Client Side Map Descriptors
Map_Descriptors
Map_Descriptor_Name, Data_Array_Name, Data_Array_Offset, Function, Node_name, Data_Type, Length, SCS128_Poll_Type
CMD_DO1, DA_DO_01, 0, Wrbx, Node_A, Flag, 256, Controls
CMD_DO2, DA_80_01, 0, Wrbx, Node_A, Byte, 32, Controls
CMD_DO3, DA_160_01, 0, Wrbx, Node_A, Register, 16, Controls
CMD_DO4, DA_320_01, 0, Wrbx, Node_A, 32-Bit_Reg, 8, Controls
```

Name of one of the Data Arrays defined in Data_Array section. This Data Array holds control statuses to be written to other devices.

Starting offset in the Data Array which holds the 1st control status.

Wrbx: send command to the Server when the associated Controls updates or changes.
 write
 To write continuously use Wrbx as function and add a Scan_Interval parameter to configure how often to write.

One of the Nodes declared in the Node Section. Indications will read from the station address belonging to this Node

Number of consecutive locations in the Data array to use when issuing the Write command.

5. Configuring the FieldServer as a Safetran-SCS-128 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 Safetran-SCS-128 Client. When configured as a Server, the FieldServer is unable to keep synchronism with other Servers and Clients on the network if it is being polled using ‘Poll’ Function (i.e. single byte message). Therefore the FieldServer must be polled with ‘Recall’ request for indication statuses. The FieldServer will always respond with the current statuses.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Safetran-SCS-128 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 clients needs to be mapped in the “Server 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.

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 ²
Protocol	Specify protocol used	SCS-128 SCS128 Safetran SCS-128
Baud*	Specify baud rate	75, 150, 300 ,600, 1200, 2400, 9600 (Vendor limitation)
Parity*	Specify parity	Even (Vendor limitation)
Data_Bits*	Specify data bits	8 (Vendor limitation)
Stop_Bits*	Specify stop bits	1 (Vendor limitation)
Handshaking*	Specify hardware handshaking	None
Server_Hold_Timeout	Specifies time FieldServer will reserve Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s

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.

Example

```
// Server Side Connections

Connections
Port,          Protocol,          Baud,          Parity,          Handshaking
P8,           SCS-128,          9600,          None,            None
```

5.2. Server Side Node Descriptors

Section Title		
Nodes		
Column Title	Function	Legal Values
Node_Name	Provide name for Node	Up to 32 alphanumeric characters
Node_ID	Modbus station address of physical Server Node	1-255
Protocol	Specify protocol used	SCS-128 SCS128 Safetran SCS-128
Server_Hold_Timeout	Specifies time FieldServer will reserve Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s

Example

```
// Server Side Nodes

Nodes
Node_Name,    Node_ID,    Protocol    *
PLC 1,        1,          SCS-128
```

* Note that no connection information is necessary on Server side

5.3. Server Side Map Descriptors

5.3.1. 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	Server
Server_Hold_Timeout	Specifies time FieldServer will reserve Server side connection while waiting for the Client side to update data in Data_Array (if necessary)	>1.0s

5.3.2. 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
SCS128_Poll_Type	Specify type of requests to handle from client. 'Indications' Handle all read requests from client. 'Controls' Handle all write requests from client	Indications, Controls
Data_Type	'Flag' each byte of Controls will be stored in 8 consecutive locations in the Data Array. Similarly while responding Indications; status byte will be composed with data from 8 consecutive locations. 'Byte' each byte of Controls will be stored as 8 Bit Integer at specified location. Similarly while responding Indications; status byte will be composed with data from single location. 'Register' every two bytes of Controls will be stored as 16 Bit integer at specified location. Similarly while responding indications; two Control bytes will be composed with data from single location. '32-Bit_Reg' every four Control bytes will be stored as 32 Bit integer at specified location. Similarly while responding indications; four Control bytes will be composed with data from single location. Refer to Appendix A	Flag , Byte, Register, 32-Bit_Reg
Length	Length of Map Descriptor Number of indications to serve or number of controls to store as per Data_Type.	Flag: 8-256, 16 Byte: 1-32, 2 Register: 1-16, 1 32-Bit_Reg: 1-8, 1
Address	Irrelevant parameter for this driver	

5.3.3. Map Descriptor Example.

The following Map Descriptors will Serve 256 indication statuses emulating Server Node_A (described in Client Side Node Descriptors sections). One such Map Descriptor is required for each Node.

```
// Server Side Map Descriptors
```

Map_Descriptors	Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name	Data_Type,	SCS128_Poll_Type,	Length,	Scada_Hold_Timeout
SMD_DI1,	DA_DI,	0,	Server,	Node_A,	Flag,	Indications,	256,	4	
SMD_DI2,	DA_8I_01,	0,	Server,	Node_A,	Byte,	Indications,	32,	4	
SMD_DI3,	DA_16I_01,	0,	Server,	Node_A,	Register,	Indications,	16,	4	
SMD_DI4,	DA_32I_01,	0,	Server,	Node_A,	32-Bit_Reg,	Indications,	8,	4	
SMD_DO1,	DA_DO,	0,	Server,	Node_A,	Flag,	Controls,	256,	4	
SMD_DO2,	DA_8O_01,	0,	Server,	Node_A,	Byte,	Controls,	32,	4	
SMD_DO3,	DA_16O_01,	0,	Server,	Node_A,	Register,	Controls,	16,	4	
SMD_DO4,	DA_32O_01,	0,	Server,	Node_A,	32-Bit_Reg,	Controls,	8,	4	

Name of one of the Data Arrays defined in the Data_Array section. Indications statuses will be served to clients from this Data Array.

One of the Nodes declared in the Node Section. The Node holds the Node_Id

See Driver Specific Map Descriptor Parameters

Specify what type of requests from client should be processed

Number of indication statuses Node will have. It also depends upon Data_Type. All example lengths have the same meaning. Flag - 1 bit, Byte - 8 bits Register - 16 bits 32-Bit_Reg - 32 bits

Appendix A. Advanced Topics

Appendix A.1. Data Format

- LSB bit in Indication/Control Byte is the 1st status/control value
- Suppose Data_Type is 'Byte' and only the 1st and 2nd indication status values are 1 (out of 8 statuses), all others being zero, Driver will store a numeric value of 3 in Data Array.
- Suppose Data_Type is 'Register' and only the 1st and 9th indication statuses values are 1 (out of 16 statuses), all others being zero, Driver will store numeric value of 257 in the Data Array.
- Suppose Data_Type is '32-Bit_Reg' and only the 1st 9th and 17th indication statuses values are 1 (out of 32 statuses), all others being zero, Driver will store the numeric value 65793 in the Data Array.