



---

A Sierra Monitor Company

**Driver Manual**  
(Supplement to the FieldServer Instruction Manual)

**FS-8700-96 SMC5000**

**APPLICABILITY & EFFECTIVITY**

**Effective for all systems manufactured after May 1, 2001**

<b>Driver Version:</b>	<b>1.00</b>
<b>Document Revision:</b>	<b>2</b>

## TABLE OF CONTENTS

<b>1. SMC5000 Description .....</b>	<b>3</b>
<b>2. Driver Scope of Supply.....</b>	<b>4</b>
2.1. Supplied by FieldServer Technologies for this driver .....	4
2.2. Provided by the Supplier of 3 <sup>rd</sup> Party Equipment.....	4
2.2.1. <i>Hardware</i> .....	4
<b>3. Hardware Connections .....</b>	<b>5</b>
3.1. Hardware Connection Tips / Hints .....	5
<b>4. Configuring the FieldServer as a SMC5000 Client.....</b>	<b>6</b>
4.1. Data Arrays/Descriptors .....	6
4.2. Client Side Connection Descriptors .....	7
4.3. Client Side Node Descriptors .....	7
4.4. Client Side Map Descriptors .....	8
4.4.1. <i>FieldServer Related Map Descriptor Parameters</i> .....	8
4.4.2. <i>Driver Related Map Descriptor Parameters</i> .....	8
4.4.3. <i>Timing Parameters</i> .....	8
4.4.4. <i>Map Descriptor Example</i> .....	9
<b>5. Configuring the FieldServer as a SMC5000 Server.....</b>	<b>10</b>
5.1. Server Side Connection Descriptors.....	10
5.2. Server Side Node Descriptors .....	11
5.3. Server Side Map Descriptors.....	12
5.3.1. <i>FieldServer Specific Map Descriptor Parameters</i> .....	12
5.3.2. <i>Driver Specific Map Descriptor Parameters</i> .....	12
5.3.3. <i>Map Descriptor Example</i> .....	13
<b>Appendix A. Advanced Topics .....</b>	<b>14</b>
Appendix A.1. Poller_Client Map Descriptor Data Array .....	14
Appendix A.2. SMC_Functions – Client Application .....	15
Appendix A.3. SMC_Functions – Server Application.....	15

## 1. SMC5000 Description

The SMC5000 driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using SMC5000 protocol. The FieldServer can emulate either a Server or Client, although the SMC5000 driver is only available in a Client configuration. Any Server drivers are for FieldServer testing purposes only.

### Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	1	Only 1 Client node allowed on multidrop systems
Server	255	A maximum of 255 nodes can be connected on a multidrop network, although performance declines as more nodes are added. In non-multidrop systems only 1 Server node is allowed.

The SMC5000 protocol is used to provide report-by-exception (RBE) capability when communicating with the Sierra Monitor's Model 5000 Sentry. The Sentry always acts as a SMC5000 Server node. Changes of critical Sentry registers (e.g. concentrations, alarms, etc.) are thus sent to the Client at a higher frequency than normal registers.

## 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 Ethernet connection
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-02	RJ45 to DB9F connector adapter
FS-8917-01	RJ45 to DB25M connection adapter
FS-8917-21	RS-485 connection adapter
FS-8700-96	Driver Manual.

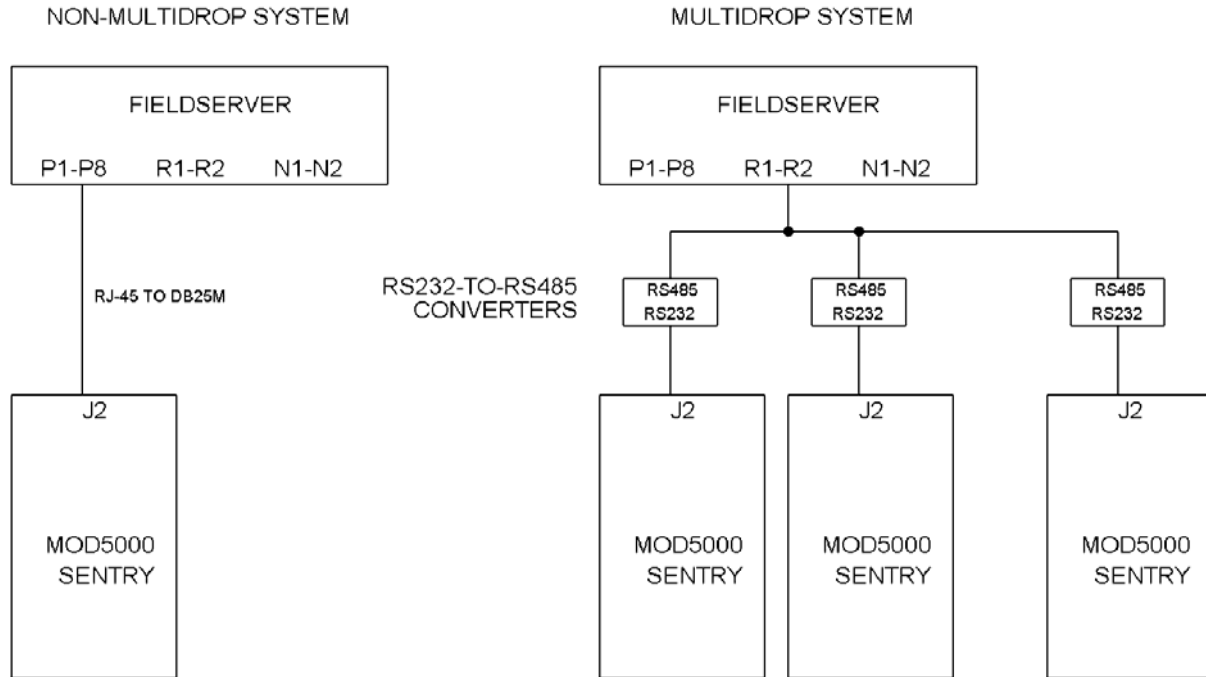
### 2.2. Provided by the Supplier of 3<sup>rd</sup> Party Equipment

#### 2.2.1. Hardware

Part #	Description
	Sierra Monitor Model 5000 Sentry

### 3. Hardware Connections

The FieldServer is connected to the Model 5000 Sentry as shown in the connection drawing. Configure the Model 5000 Sentry according to manufacturer's instructions



#### 3.1. Hardware Connection Tips / Hints

When using the X40 model, make sure you connect the FieldServer to the network using one or both of the RS-485 ports marked R1 and R2. If more ports are required, P1 – P8 may be used in conjunction with an RS-232-to-RS-485 converter.

When using the X20 model, make sure the serial port is configured as a RS-485 port. Refer to **ENOTE0017 - Setting FS-B20 up for RS-485**.

## 4. Configuring the FieldServer as a SMC5000 Client

For a detailed discussion on FieldServer configuration, please 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 SMC5000 Server. The FieldServer should always be configured as a Client driver when the user wants it to communicate directly with one or more Model 5000 Sentry devices.

### 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 SMC5000 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, 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 Descriptors

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>1</sup>
Protocol	Specify protocol used	SMC5000
Baud*	Specify baud rate	<b>1200</b> or 110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, <b>None</b> , Mark, Space
Data_Bits*	Specify data bits	<b>7, 8</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>
Poll_Delay*	Time between internal polls	0-32000 seconds, <b>1 second</b>

### Example

// Client Side Connections					
Connections					
Port,	Protocol,	Baud,	Parity,	Handshaking,	Poll_Delay
P8,	SMC5000,	1200,	None,	None,	0.050s

## 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	Modbus station address of physical Server node (Sentry)	1-255
Protocol	Specify protocol used	SMC5000
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>1</sup>

### Example

// Client Side Nodes			
Nodes			
Node_Name,	Node_ID,	Protocol,	Connection
SENTRY_1,	1,	SMC5000,	P8

<sup>1</sup> 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

Section Title		
Map Descriptors		
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	COS_POLLER, RDBC, PASSIVE, SERVER, COS_READ, WRBX

##### 4.4.2. Driver Related Map Descriptor Parameters

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
SMC_Function	Additional field describing the Function or Task to be performed.	Poller_Client, Passive_Client (See Driver Notes)
Node_Name	Name of Node to fetch data from	One of the node names specified in "Client Node Descriptor" above
Length	Length of Map Descriptor	1-500
Address	Starting address of read block	40001-49999 (See also SMC5000/Modbus Address Map of Sentry)

##### 4.4.3. Timing Parameters

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



**4.4.4. Map Descriptor Example.**

```
// Client Side Map Descriptors
```

Map Descriptors	Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	SMC_Function	Node_name	Address	Length	Scan_Interval
A_Poller,	A_STATUS,	0,	0,	COS_POLLER,	Poller_Client,	Node_A,	42001,	5,	0.1
A1,	DA_A11,	0,	0,	COS_READ,	Passive_Client,	Node_A,	42001,	100,	-
A2,	DA_A12,	0,	0,	PASSIVE,	Passive_Client,	Node_A,	40635,	16,	-
A3,	DA_A13,	0,	0,	WRBX,	Passive_Client,	Node_A,	40634,	1,	-

When a Passive\_Client is set as COS\_READ, it enables it to be Quick-loaded at FieldServer startup or when the node comes back on-line.

When a Passive\_Client is set as WRBX, it enables it to send a write message to the Server when the data in its Data Array has changed. This will cause the relevant data on the Server to be updated to the changed data.

Client Side Map Descriptors should always start with a Poller\_Client Map Descriptor as shown here. It is mapped to a Data Array that will provide an overview of the types of messages received. See Appendix A.1.

Following the Poller\_Client Map Descriptor are the Passive\_Client Map Descriptors for that node, which will fill their respective Data Arrays with the data received from the Servers.

These Addresses together with the lengths corresponds to SMC5000 or Modbus Addresses on the Server

The Polling speed is set by this value.

## 5. Configuring the FieldServer as a SMC5000 Server

For a detailed discussion on FieldServer configuration, please 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” files provided with the FieldServer.)

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a SMC5000 Client. The FieldServer should be configured as a Server when it must emulate one or more Model 5000 Sentry devices.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for SMC5000 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, R1-R2
Protocol	Specify protocol used	SMC5000
Baud*	Specify baud rate	<b>1200</b> or 110 – 115200, standard baud rates only
Parity*	Specify parity	Even, Odd, <b>None</b> , Mark, Space
Data_Bits*	Specify data bits	7, <b>8</b>
Stop_Bits*	Specify stop bits	<b>1</b>
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>
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 Connections				
Connections				
Port,	Protocol,	Baud,	Parity,	Handshaking
P8,	SMC5000,	1200,	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	SMC5000/Modbus station address of physical Server node	1-255
Protocol	Specify protocol used	SMC5000
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
SENTRY_1,            1,                SMC5000
```

### 5.3. Server Side Map Descriptors

#### 5.3.1. FieldServer Specific Map Descriptor Parameters

Section Title		
Map Descriptors		
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, WRBX
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

Section Title		
Map Descriptors		
Column Title	Function	Legal Values
SMC_Function	Additional Field for defining the function of the Map Descriptor.	LCD_SERVER, IDLE_SERVER, COS_SERVER. (See <b>Error! Reference source not found.</b> )
Node_Name	Name of Node to fetch data from	One of the node names specified in "Client Node Descriptor" above
Length	Length of Map Descriptor	1 – 1000
Address	Starting address of read block	40001-49999 (See also SMC5000/Modbus Address Map of Sentry)

### 5.3.3. Map Descriptor Example.

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

Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	SMC_Function,	Node_name	Address	Length
A_IDLE,	DA_IDLE,	0,	SERVER,	IDLE_SERVER,	Node_A,	42001,	500,
A_LCD_TRIGGER,	DA_KEYPRESS,	0,	SERVER,	LCD_SERVER,	Node_A,	40634,	1,
A_LCD_DISPLAY,	DA_LCD,	0,	WRBX,	LCD_SERVER,	Node_A,	40635,	16,
A_COS1,	DA_COS,	0,	WRBX,	COS_SERVER,	Node_A,	42001,	1,
A_COS2,	DA_COS,	1,	WRBX,	COS_SERVER,	Node_A,	42002,	1,

The IDLE\_SERVER Map Descriptors define the data and their addresses that would be sent to the Client at a low priority and frequency. Their function fields have to be set to SERVER.

A LCD\_SERVER defined as SERVER function will mark the Remote Keypress register that will activate the LCD COS responses. These responses will cease when no new Keypress was received a minute after the previous

A LCD\_SERVER defined as WRBX function will mark the LCD registers. For a minute after a Remote Keypress the contents and addresses of these registers will be sent to the Client each time its contents change.

Each Change-of-State (COS) register needs to be defined in its own COS\_SERVER Map Descriptor and needs its Function field to be set to WRBX

## Appendix A. Advanced Topics

### Appendix A.1. Poller\_Client Map Descriptor Data Array

On the SMC5000 Client the Poller\_Client Map Descriptor is mapped to a UINT16 Data Array that will provide an overview of the types of messages received from the Server. The five registers in the Data Array are defined as follows:

1	2	3	4	5
COS Messages Received	IDLE Messages Received	LCD-COS Messages Received	COS Queue Overruns Received	Synchronize Status

Register	Description
COS Messages Received	This register will keep count of the Change-of-State (COS) responses received from the Server node or Sentry device.
IDLE Messages Received	This register will keep count of the Idle responses received from the Server node.
LCD-COS Messages Received	This register will keep count of the LCD Change-of-State COS responses received from the Server node.
COS Queue Overrun Messages Received	This register will keep count of the COS overrun responses received from the Server node. Change-of-State events are stored in the Server's COS queue until the Client has successfully received all these events. Should this COS Queue overrun for some reason, it would indicate that an COS events might have been lost. See next paragraph for more information.
Synchronize Status	After FieldServer start-up or after a COS queue overrun message has been received from the Server, the synchronize status will be zero until all the COS and Idle registers have been read at least once from the Server, at which point this register will contain a value of 1.

## Appendix A.2. SMC\_Functions – Client Application

Map Descriptors pertaining to a specific node require a Poller\_Client at the top, followed by sufficient Passive\_Client Map Descriptors to adequately cover all the possible Register Addresses found on the Server or Sentry device. See the SMC5000/Modbus Address Map for the relevant information.

The Poller\_Client Map Descriptor Function field must be set to COS\_POLLER or RDBC. The Passive\_Client's Function field can be set to either SERVER, PASSIVE, COS\_READ or WRBX. The differences are described in the following table:

Function	Description
SERVER (or PASSIVE)	This is the most basic functionality. This Map Descriptor would only specify the Data Array locations to store the data received from the Server falling within its address range.
COS_READ	COS_READ Map Descriptors are the same as the SERVER types described above, with the added functionality of Quickloads. A Quickload is when all data for the Map Descriptor is read from the Server in a single message. A Quickload would be done when the FieldServer is started or when a node comes back on-line.
WRBX	When the Passive_Client Map Descriptor has been set as a WRBX function it will still perform as a SERVER type previously described, but with the added functionality of being able to send data in a write message to the Server. A write would be issued only when its Data Array's contents had changed. The result of the write will be that the relevant registers on the Server would be updated with the data sent.

## Appendix A.3. SMC\_Functions – Server Application

SMC5000 Server Map Descriptors are divided into COS\_SERVER, IDLE\_SERVER and LCD\_SERVER as follows:

Function	Description
COS_SERVER	COS or change-of-state Server Map Descriptors are used to monitor high-priority data such as alarms and concentrations on the Sentry. The Function field must be set to WRBX.
IDLE_SERVER	Idle Server Map Descriptors are used to monitor the low priority registers/data such as set points on the Sentry which will be sent to the Client at a much lower frequency. Function field must be set to SERVER.
LCD_SERVER	LCD Server Map Descriptors will define the registers/data eligible for change-of-state monitoring specific to the Sentry's front panel LCD display. These LCD data responses are enabled only when a write to the Remote Keypress register is received on the Server and disabled when no write to this register is received for 1 minute. The Remote Keypress register must be defined in the Function field as a SERVER, while the actual LCD data is defined with a WRBX function (refer to 5.3.3)

The SMC5000/Modbus Address Map provides the register addresses together with their descriptions for the Model 5000 Sentry device.

THIS PAGE INTENTIONALLY LEFT BLANK