



---

A Sierra Monitor Company

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

**FS-8700-42 Spectronics**

**APPLICABILITY & EFFECTIVITY**

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

<b>Driver Version:</b>	<b>1.00</b>
<b>Document Revision:</b>	<b>4</b>

## Table of Contents

<b>1. Spectronics 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 Supplier of 3 <sup>rd</sup> Party Equipment.....	4
<b>3. Hardware Connections.....</b>	<b>5</b>
<b>4. Configuring the FieldServer as a Spectronics 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 Spectronics 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.....	11
5.3.1. <i>FieldServer Specific Map Descriptor Parameters</i> .....	11
5.3.2. <i>Driver Specific Map Descriptor Parameters</i> .....	11
5.3.3. <i>Map Descriptor Example.</i> .....	12
<b>Appendix A. Advanced Topics – Spectronics .....</b>	<b>13</b>
Appendix A.1. Default Data Types.....	13
Appendix A.2. Single Writes .....	13
Appendix A.3. Write-Thru Operation.....	13
Appendix A.4. Connection to York Modbus Microgateway.....	14

## 1. Spectronics Description

The Spectronics driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Spectronics protocol. The FieldServer can emulate either a Server or Client.

The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer.

## 2. Driver Scope of Supply

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

FieldServer Technologies PART #	Description
FS-8915-10	7' Patch Cable
FS-8917-01	RJ45 to DB25M connector adapter
FS-8700-01	Driver Manual

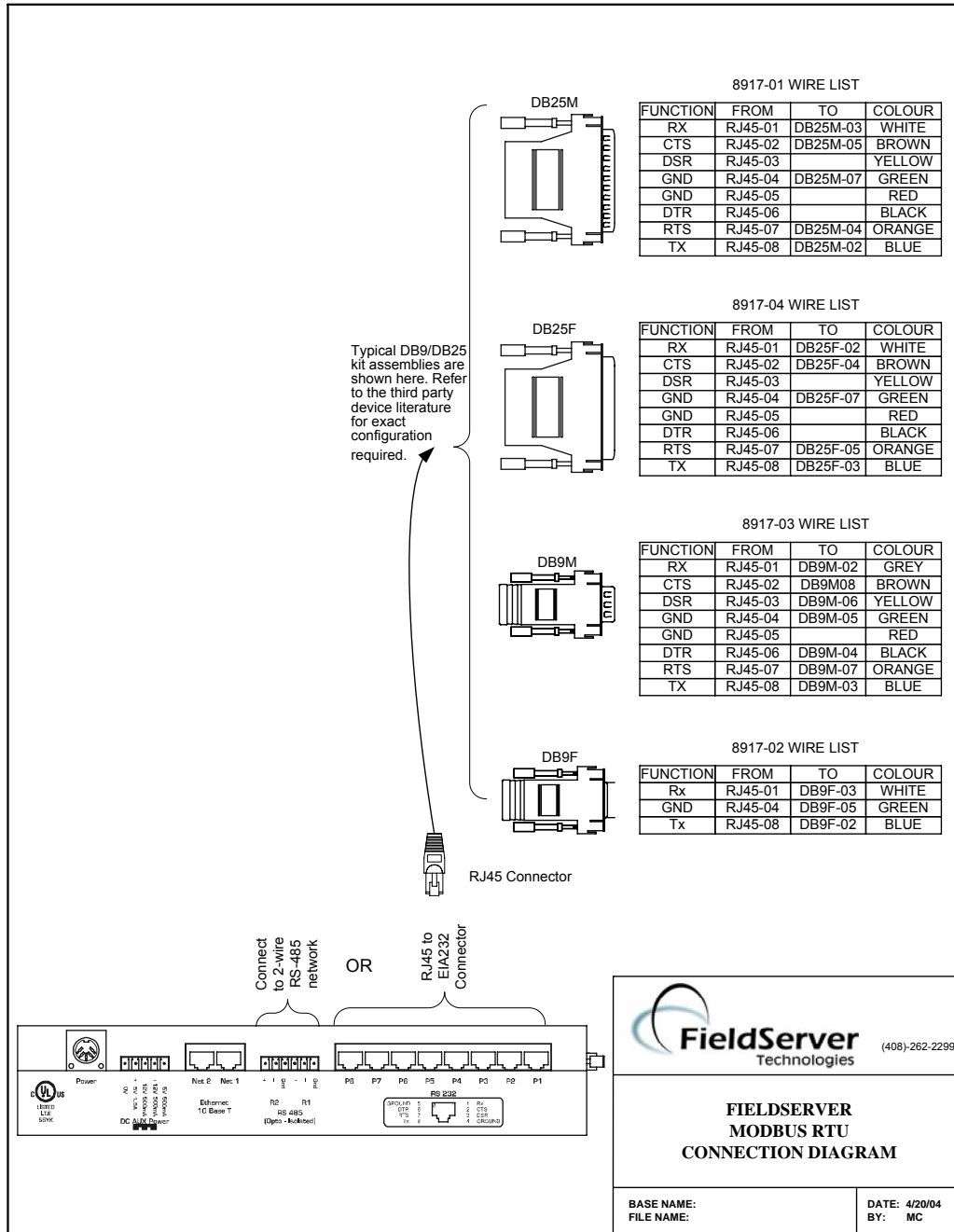
### 2.2. Provided by Supplier of 3<sup>rd</sup> Party Equipment

PART #	DESCRIPTION
	Spectronics Device

### 3. Hardware Connections

It is possible to connect a Spectronics device to any of the existing serial ports on the FieldServer<sup>1</sup>. These ports simply need to be configured for Spectronics in the configuration file.

Configure the Spectronics device according to manufacturer's instructions.



<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. Configuring the FieldServer as a Spectronics 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 Spectronics 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 Spectronics 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>2</sup>
Baud*	Specify baud rate	110 – 115200, standard baud rates only
Parity*	Specify parity	<b>None</b> (Vendor limitation)
Data_Bits*	Specify data bits	<b>8</b> (Vendor limitation)
Stop_Bits*	Specify stop bits	<b>1</b> (Vendor limitation)
Protocol	Specify protocol used	Spectronics
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>
Poll Delay*	Time between internal polls	0-32000 s, <b>1 s</b>

### Example

// Client Side Connections					
Connections					
Port,	Protocol,	Baud,	Parity,	Handshaking,	Poll_Delay
P8,	Spectronics,	9600,	None,	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	Station address of physical server node	1-255
Protocol	Specify protocol used	Spectronics
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2 <sup>2</sup>

### Example:

// Client Side Nodes			
Nodes			
Node_Name,	Node_ID,	Protocol,	Port
Spec_device1,	1,	Spectronics,	P8

<sup>2</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

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

##### 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
Length	Length of Map Descriptor	1-125 (Register, AI)
Address	Starting address of read block	40001, 30001, etc
Data_Array_Low_Scale*	Scaling zero in Data Array	-2,147,483,648 to 2,147,483,647, <b>0</b>
Data_Array_High_Scale*	Scaling max in Data Array	-2,147,483,648 to 2,147,483,647, <b>100</b>
Node_Low_Scale*	Scaling zero in Connected Node	-2,147,483,648 to 2,147,483,647, <b>0</b>
Node_High_Scale*	Scaling max in Connected Node	-2,147,483,648 to 2,147,483,647, <b>100</b>

##### 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.

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

Map_Descriptors							
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_Name,	Address,	Length,	Scan_Interval
CMD_AI_01,	DA_AI_01,	0,	RDBC,	Spec_Device1,	30001,	20,	1.000s
CMD_AO_01,	DA_AO_01,	0,	RDBC,	Spec_Device1,	40001,	20,	1.000s
CMD_DI_01,	DA_DI_01,	0,	RDBC,	Spec_Device1,	10001,	20,	1.000s
CMD_DO_01,	DA_DO_01,	0,	RDBC,	Spec_Device1,	00001,	20,	1.000s

## 5. Configuring the FieldServer as a Spectronics 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 Spectronics Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Spectronics 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 <sup>3</sup>
Baud*	Specify baud rate	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> (Vendor limitation)
Protocol	Specify protocol used	Spectronics
Handshaking*	Specify hardware handshaking	RTS, RTS/CTS, <b>None</b>

#### Example

// Server Side Connections				
Connections				
Port,	Protocol,	Baud,	Parity,	Handshaking
P1,	Spectronics,	9600,	None,	None

<sup>3</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.

**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	Node ID of physical server node	1 – 255
Protocol	Specify protocol used	Spectronics

**Example**

Nodes			
Node_Name,		Node_ID,	Protocol
Spec_Srv_11,		11,	Spectronics

**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

**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
Data_Type	Data type	Register, Coil, AI, DI
Length	Length of Map Descriptor	1 - 125
Address	Starting address of read block	40001, 30001, etc
Data_Array_Low_Scale*	Scaling zero in Data Array	-32767 to 32767, <b>0</b>
Data_Array_High_Scale*	Scaling max in Data Array	-32767 to 32767, <b>100</b>
Node_Low_Scale*	Scaling zero in Connected Node	-32767 to 32767, <b>0</b>
Node_High_Scale*	Scaling max in Connected Node	-32767 to 32767, <b>100</b>

### 5.3.3. Map Descriptor Example.

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

Map Descriptors										
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_name,	Address,	Length,	Data_Array_Low_Scale,	Data_Array_High_Scale	Node_Low_Scale	Node_High_Scale
SMD_AI_01,	DA_AI_01,	0,	Server,	Spec_Srv_11,	30001,	200,	0,	100,	0,	10000
SMD_AO_01,	DA_AO_01,	0,	Server,	Spec_Srv_11,	40001,	200,	0,	100,	0,	10000
Map Descriptors										
Map_Descriptor_Name,	Data_Array_Name,	Data_Array_Offset,	Function,	Node_name,	Address,	Length				
SMD_DI_01,	DA_DI_01,	0,	Server,	Spec_Srv_11,	10001,	200				
SMD_DO_01,	DA_DO_01,	0,	Server,	Spec_Srv_11,	00001,	200				

## Appendix A. Advanced Topics – Spectronics

### Appendix A.1. Default Data Types

When a Spectronics address range is specified, a particular Data Type is implied. The defaults are as follows:

Address range	Data_Type	Function Code (Write)	Function Code (Read)
40001 - 49999	Register	16	3
30001 - 39999	Analog_Input	n/a.	4
10001 - 19999	Digital_Input	n/a.	2
00001 - 09999	Coil	15	1

### Appendix A.2. Single Writes

When writing the default data types can be overwritten using the “Single\_Coil” and “Single\_Register” settings as part of the Map Descriptor configuration. In that case the Function codes for writes will be as follows:

Address range	Data_Type	Function Code (Write)
40001 - 49999	Single_Register	6
30001 - 39999	Coil	5.

#### **Example: FC 6 = Write Single Register**

Add a parameter to the Spectronics client side Map Descriptor called Data\_Type.

If you specify the Data\_Type as Single\_Register and the Function as WRBC or WRBX, then a Spectronics poll with FC 6 will be generated.

Of course Single\_Register implies a length of one, and even if you try to set the length longer in the csv file, the length is limited to 1 in the driver.

### Appendix A.3. Write-Thru Operation

Consider a client Map Descriptor that is configured to read data using a FC 3 (Read Multiple Register) operation. It is now possible do a write thru operation on the existing Client Map Descriptor by storing data to the Client Map Descriptors Data Array.

The Write Cache Map Descriptor that is created will use the following function codes depending on the Cache Map Descriptor Length and the Clients Node Type.

Single Register Write - FC 6 - When the length of the Cache Map Descriptor is one. This will normally be the case.

Multiple Register Write – FC 16 - When the length of the Cache Map Descriptor is larger than one or the Node\_Type has been set to “Block\_Mode”.

***Appendix A.4. Connection to York Modbus Microgateway***

If connecting the FieldServer to a York Modbus Microgateway, the Node\_ID of the Microgateway is defined by the address DIP switches. If switch 4 is set to 'On' and the other switches are set to 'off' then Node\_ID of the Microgateway is '247', the parity is 'Even', and the stop bits are 1. Other Node\_ID combinations can be found in the York Modbus Microgateway Installation Manual.

THIS PAGE INTENTIONALLY LEFT BLANK