



**FieldServer**  
**FS-8700-32 Modbus Tek-Air**  
**Driver Manual**  
(Supplement to the FieldServer Instruction Manual)

**APPLICABILITY & EFFECTIVITY**

Effective for all systems manufactured after August 2019.

Driver Version: 1.02  
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](http://www.sierramonitor.com)

U.S. Support Information:

+1 408 964-4443

+1 800 727-4377

Email: [support@sierramonitor.com](mailto:support@sierramonitor.com)

EMEA Support Information:

+31 33 808 0590

Email: [support.emea@sierramonitor.com](mailto:support.emea@sierramonitor.com)

**TABLE OF CONTENTS**

- 1 Modbus Tek-Air Description..... 4**
- 2 Driver Scope of Supply ..... 4**
  - 2.1 Supplied by Sierra Monitor Corporation ..... 4
  - 2.2 Provided by Supplier of 3<sup>rd</sup> Party Technology ..... 4
- 3 Hardware Connections..... 5**
- 4 Data Array Parameters ..... 6**
- 5 Configuring the FieldServer as a Tek-Air Modbus Client..... 7**
  - 5.1 Client Side Connection Parameters ..... 7
  - 5.2 Client Side Node Parameters ..... 7
  - 5.3 Client Side Map Descriptor Parameters ..... 8
    - 5.3.1 FieldServer Related Map Descriptor Parameters ..... 8
    - 5.3.2 Driver Related Map Descriptor Parameters ..... 8
    - 5.3.3 Timing Parameters ..... 8
      - 5.3.1 Map Descriptor Example ..... 8
- 6 Configuring the FieldServer as a Tek-Air Modbus Server..... 9**
  - 6.1 Server Side Node Parameters..... 9
  - 6.2 Server Side Map Descriptor Parameters..... 10
    - 6.2.1 FieldServer Related Map Descriptor Parameters ..... 10
    - 6.2.2 Driver Related Map Descriptor Parameters ..... 10
    - 6.2.3 Timing Parameters ..... 10
      - 6.2.1 Map Descriptor Example ..... 10
- Appendix A Useful Features ..... 11**
  - Appendix A.1 Node\_Offline\_Response ..... 11
  - Appendix A.2 Map Descriptor Functions ..... 12
- Appendix B Troubleshooting..... 13**
  - Appendix B.1 Effect of the Length Parameter – Data Read/Write ..... 13

## 1 MODBUS TEK-AIR DESCRIPTION

The Tek-Air Modbus RTU driver allows the FieldServer to transfer data to and from Tek-Air devices over either RS-232 or RS-485 using the Enhanced Tek-Air Modbus RTU protocol. The driver was developed for Modbus Application Protocol Specification V1.1a" from Modbus-IDA. The specification can be found at [www.modbus.org](http://www.modbus.org). Modbus Tek-Air is the same as Modbus\_RTU, except that it has the ability to concatenate bytes of data in a packet to create floating point values. The order in which the bytes are combined and the address range used was developed specifically for the Tek-Air/Modbus interface. To accommodate this, the driver must be configured to poll using odd numbered addresses as two integer registers are used for every floating point value. If this value is stored in a floating point data array, it will be retrievable in the correct format on the Server side. Note that the Tek-Air "Double" data type is not supported.

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 Sierra Monitor Corporation

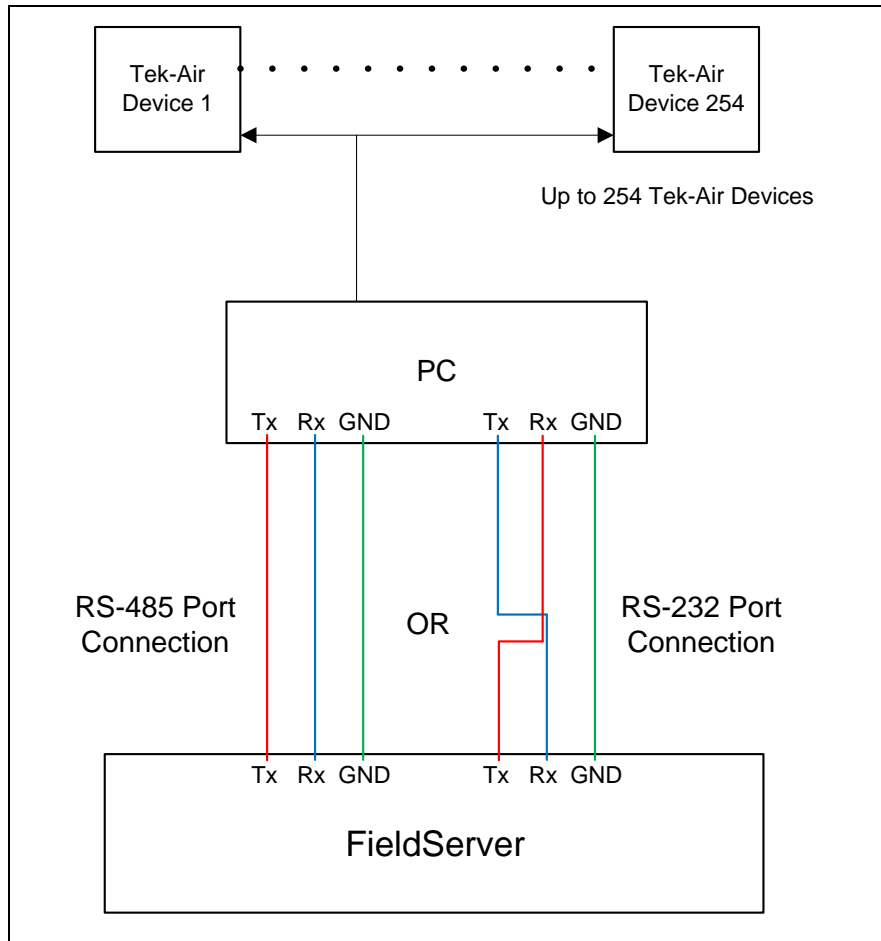
Part #	Description
FS-8917-16	UTP cable (7 foot) for RS-232 use

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

Part #	Description
	Tek-Air device

### 3 HARDWARE CONNECTIONS

The FieldServer is connected to the Modbus Tek-Air as shown in connection drawing. Configure the Modbus Tek-Air according to manufacturer's instructions.



## 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           , Float             , 100
DA_AO           , Float             , 100
DA_DI           , Bit              , 100
DA_DO           , Bit              , 100
    
```

## 5 CONFIGURING THE FIELDSEVER AS A TEK-AIR MODBUS 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 Tek-Air Modbus Server.

### 5.1 Client Side Connection Parameters

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer.	R1, R2 <sup>1</sup>
Baud*	Specify baud rate.	300-9600-38400
Parity*	Specify parity.	Even, Odd, <b>None</b>
Data_Bits*	Specify data bits.	7, 8
Protocol	Specify protocol used.	Tek-Air
Poll Delay*	Time between internal polls.	0-32000s, <b>1s</b>
Stop_Bits	Specify stop bits.	1, 2

**Example**

```
// Client Side Connections

Connections
Port , Baud , Parity , Data_Bits , Stop_Bits , Protocol , Poll_Delay , Poll_Delay
R1 , 9600 , None , 8 , 1 , Tek_Air , 0.100s , 0.100s
```

### 5.2 Client Side Node Parameters

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.	Tek-Air
Port	Specify which port the device is connected to the FieldServer.	R1, R2 <sup>1</sup>

**Example**

```
// Client Side Nodes

Nodes
Node_Name , Node_ID , Protocol , Port
Tek_01 , 1 , Tek-air , R1
```

<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.

### 5.3 Client Side Map Descriptor Parameters

#### 5.3.1 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 defined in <b>Section 4</b>
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length -1) as defined in <b>Section 4</b>
Function	Function of Client Map Descriptor. Refer to <b>Appendix A.2</b> .	Rdbc, Wrbc, Wrbcx, Arcs

#### 5.3.2 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 <b>Section 5.2</b>
Address	Starting address of read block.	30001, 30003, 30005 ... 40001, 40003, 40005 ... 10001, 10002, 10003 ... 00001, 00002, 00003 ...
Length	Specifies how many register bits etc. to read. Refer to <b>Appendix B.1</b> .	0 - 125
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>

**NOTE: For information on applying scaling, refer to the Configuration Manual.**

#### 5.3.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval*	Seconds per Scan	0-32000, <b>20</b>

#### 5.3.1 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_1 , DA_AI , 0 , Rdbc , Tek_01 , 30001 , 3 , 20.000s
CMD_AO_1 , DA_AO , 0 , Rdbc , Tek_01 , 40001 , 3 , 20.000s
CMD_DI_1 , DA_DI , 0 , Rdbc , Tek_01 , 10001 , 3 , 20.000s
CMD_DO_1 , DA_DO , 0 , Rdbc , Tek_01 , 1 , 3 , 20.000s
```



## 6 CONFIGURING THE FIELDSEVER AS A TEK-AIR MODBUS SERVER

For detailed information FieldServer configuration, 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 Modbus Tek\_Air Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Modbus Tek\_Air 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: In the tables below, \* indicates an optional parameter and the bold legal values are default.**

Section Title		
Connections		
Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer.	R1, R2 <sup>2</sup>
Baud*	Specify baud rate.	110 – 115200; standard baud rates only, <b>9600</b>
Parity*	Specify parity.	Even, Odd, <b>None</b>
Data_Bits*	Specify data bits.	<b>7, 8</b>
Stop_Bits*	Specify stop bits.	<b>1</b> (Vendor limitation)
Protocol	Specify protocol used.	Tek-Air

**Example**

Connections										
Port	,	Baud	,	Parity	,	Data_Bits	,	Stop_Bits	,	Protocol
R1	,	9600	,	None	,	8	,	1	,	Tek-Air

### 6.1 Server Side Node Parameters

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.	Tek-Air
Port	Specify which port the device is connected to the FieldServer.	R1, R2 <sup>1</sup>
Node_Offline_Response*	Set the FieldServer response to the Modbus Tek-Air when the Server Node supplying the data has gone offline.	<b>No_Response</b> , Old_Data, Zero_Data, FFFF_Data; refer to <a href="#">Appendix A.1</a> for further information

**Example**

Nodes						
Node_Name	,	Node_ID	,	Protocol	,	Port
Tek_01	,	1	,	Tek-Air	,	R1

<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.

## 6.2 Server Side Map Descriptor Parameters

### 6.2.1 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 defined under <b>Section 4</b>
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length -1) as defined in <b>Section 4</b>
Function	Function of Server Map Descriptor.	Passive

### 6.2.2 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 <b>Section 6.1</b>
Address	Starting address of read block.	30001, 30003, 30005 ... 40001, 40003, 40005 ... 10001, 10002, 10003 ... 00001, 00002, 00003 ...
Length	Specifies how many register bits etc. to read.	0 - 125
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>

**NOTE: For information on how to apply scaling, refer to the Configuration Manual.**

### 6.2.3 Timing Parameters

Column Title	Function	Legal Values
Scan_Interval*	Seconds per Scan	0-32000s, <b>20s</b>

### 6.2.1 Map Descriptor Example

```
// Server Side Map Descriptors
//
Map_Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Address , Length , Scan_Interval
CMD_AI_1 , DA_AI , 0 , Passive , Tek_01 , 30001 , 3 , 20.000s
CMD_AO_1 , DA_AO , 0 , Passive , Tek_01 , 40001 , 3 , 20.000s
CMD_DI_1 , DA_DI , 0 , Passive , Tek_01 , 10001 , 3 , 20.000s
CMD_DO_1 , DA_DO , 0 , Passive , Tek_01 , 1 , 3 , 20.000s
```

**Appendix A Useful Features**

**Appendix A.1 Node\_Offline\_Response**

In systems where data is being collected from multiple Server Nodes and made available on a FieldServer configured as a Modbus TEK\_AIR Server, when a Server Node goes offline the default behavior of the FieldServer would be to stop responding to polls for this data. This might not be what the user wants. Various options exist making it possible to signal that the data quality has gone bad without creating error conditions in systems sensitive to the default option.

The following options can be configured under the Node parameter, Node\_Offline\_Response, to set the response of the FieldServer to the Modbus TEK\_AIR Client when the Server Node supplying the data is offline:

- No\_Response – This is the default option. The FieldServer simply does not respond when the corresponding Server Node is offline.
- Old\_Data – The FieldServer will respond, but with the last known value of the data. This maintains the communication link in an active state, but may hide the fact that the Server Node is offline.
- Zero\_Data – The FieldServer will respond, but with the data values set to zero. If the user normally expects non-zero values, this option will signal the offline condition without disrupting communications.
- FFFF\_Data – The FieldServer will respond, but with the data values set to FFFF (hex). If the user normally expects other values, this option will signal the offline condition without disrupting communications.

When configured as a Server this parameter can force a desired exception response as follows:

- Node\_Offline\_Message or Exception\_4 - FieldServer's response will be Exception 4
- Gateway\_Path\_Unavailable or Exception\_A - FieldServer's response will be Exception A
- Gateway\_Device\_Failed or Exception\_B - FieldServer's response will be Exception B

**Example**

Nodes				
Node_Name	Node_ID	Protocol	Node_Offline_Response	Port
DEV11	, 11	, Tek-Air	, No_Response	, -
DEV12	, 12	, Tek-Air	, Old_Data	, -
DEV15	, 15	, Tek-Air	, Zero_Data	, -
DEV16	, 16	, Tek-Air	, FFFF_Data	, -
DEV17	, 17	, Tek-Air	, Exception_4.	, -
DEV18	, 18	, Tek-Air	, Gateway_Path_Unavailable	, -

**Appendix A.2 Map Descriptor Functions**

Function	Description
Rdbc	This is an active Map Descriptor which polls the device once every scan interval. The function will also execute writes from the Data Array to a remote device when a change is detected, but only when used in conjunction with writeable addresses.
Arcs	This function will perform the same operation as an Rdbc (Arc) function, but will sequence through the range of addresses starting at "Address" and wrapping at "Address + Length". A length of 1 will be used for every one of the Addresses that gets polled.
Wrbx	The Wrbx function will write data from the Data Array to the remote device. The write is triggered by a change in the associated Data Array. If the associated Data Array is updated a write will occur, even if the value/s within the Data Array have not changed. The "Scan_Interval" parameter is not required for this function as writes are event driven and not continuous.
Wrbc	This is similar to the Wrbx function, except that the writes occur at a regular interval rather than on an event driven basis. The frequency of the writes is determined by the "Scan_Interval" parameter.

**Appendix B Troubleshooting****Appendix B.1 Effect of the Length Parameter – Data Read/Write**

When using WRBC or WRBX as functions to write data, then the number of points written is determined by the length parameter. However, when using RDBC the points are written one at a time even if Length is greater than 1.