

Driver Manual

FS-8700-15 Modbus Daniels

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after March 2021.



Driver Revision: 1.01
Document Revision: 2.A



fieldserver

MSA Safety
1991 Tarob Court
Milpitas, CA 95035
Website: www.MSAsafety.com

U.S. Support Information:
+1 408 964-4443
+1 800 727-4377
Email: smc-support@msasafety.com

EMEA Support Information:
+31 33 808 0590
Email: smc-support.emea@msasafety.com

Contents

1	Description	4
2	Driver Scope of Supply	4
2.1	Provided by the Supplier of 3 rd Party Equipment	4
2.1.1	Required 3 rd Party Hardware	4
3	Hardware Connections	5
4	Data Array Parameters	6
5	Client Side Configuration	7
5.1	Client Side Connection Parameters	7
5.2	Client Side Node Descriptors	7
5.3	Client Side Map Descriptor Parameters	8
5.3.1	FieldServer Specific Map Descriptor Parameters	8
5.3.2	Driver Related Map Descriptor Parameters	8
5.3.3	Timing Parameters	8
5.4	Map Descriptor Example	9

1 Description

The Modbus Daniels driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using Modbus Daniels protocol (3-9000-674 REV 2.1). The FieldServer can only emulate a Client.

Modbus Daniels is an inclusion into Modbus ASCII, allowing floats to be used. Modbus Daniels is designed to support floats in the address range 7000 to 7999. The “Double” data type is not supported. Other standard Modbus data types are supported.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Client	255	The maximum achievable number of devices on the network may be less depending on device loading and network speed characteristics. Only one client device allowed on multidrop systems.

2 Driver Scope of Supply

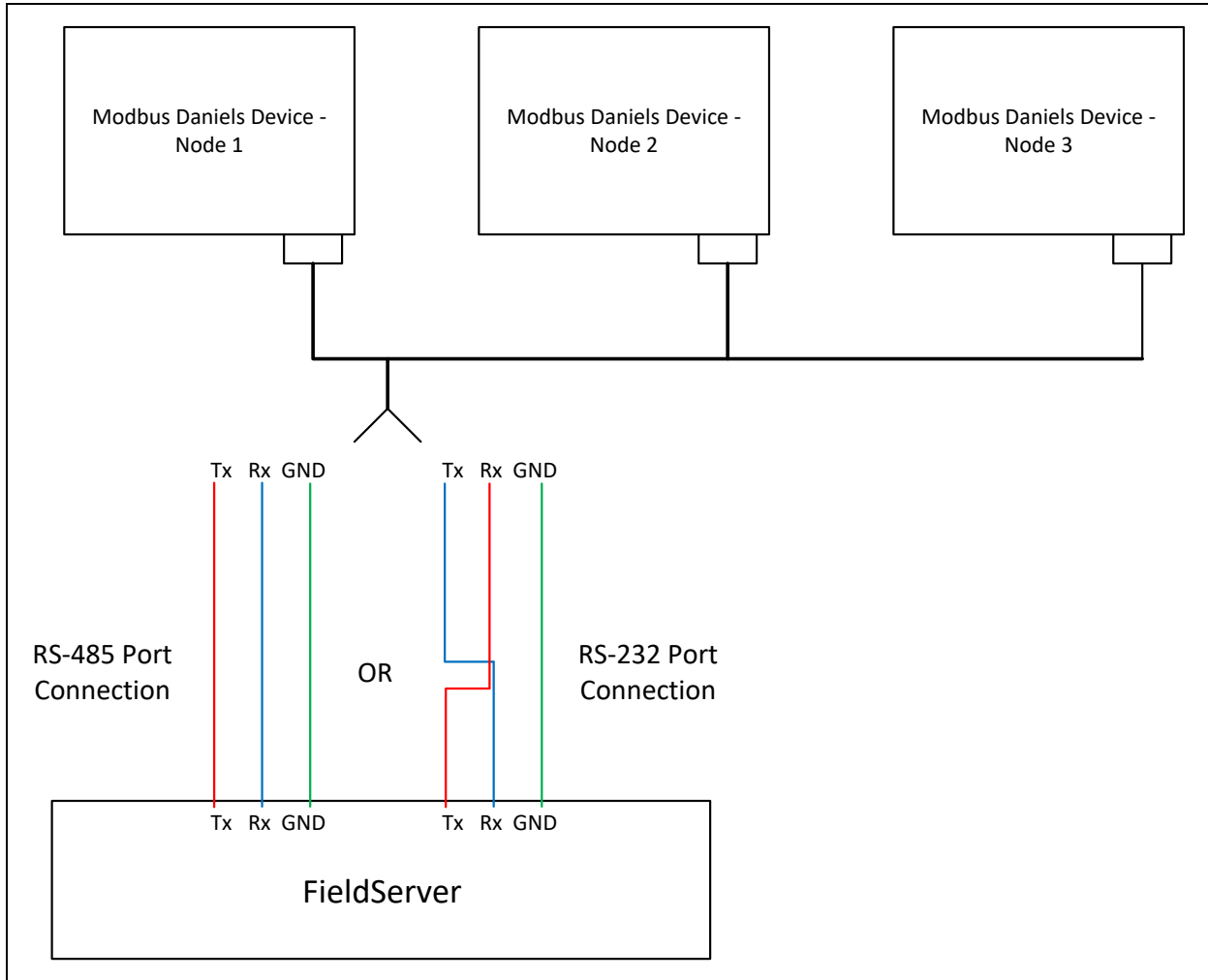
2.1 Provided by the Supplier of 3rd Party Equipment

2.1.1 Required 3rd Party Hardware

Part #	Description
	The Modbus Daniels device must be set to use Modbus ASCII protocol

3 Hardware Connections

The FieldServer is connected to the Modbus Daniels Device as shown in the connection drawing. Configure the Modbus Daniels device 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_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_Format      ,
DA_AI_01             , UInt16             , 200
DA_AO_01             , UInt16             , 200
DA_DI_01             , Bit                , 200
DA_DO_01             , Bit                , 200
```

5 Client Side Configuration

For detailed information on 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 FS).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a Modbus Daniels Server.

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

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.	P1-P2, R1-R2 ¹
Protocol	Specify protocol used.	Modbus_Daniels
Baud*	Specify baud rate.	110 – 115200, standard baud rates only, 9600
Parity*	Specify parity.	Even, Odd, None, Mark, Space
Data_Bits*	Specify data bits.	7, 8
Stop_Bits*	Specify stop bits.	1
Poll_Delay*	Time between internal polls.	0-32000 seconds, 1 second

Example

```
// Client Side Connections
Connections
Port          , Protocol          , Baud   , Parity   , Poll_Delay
R1            , Modbus_Daniels   , 9600   , Even    , 0.100s
```

5.2 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.	1-255
Protocol	Specify protocol used.	Modbus_Daniels
Connection	Specify which port the device is connected to the FieldServer.	P1-P2, R1-R2 ¹

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.

Example

```
// Client Side Nodes
Nodes
Node_Name      , Node_ID      , Protocol      , Connection
PLC 1          , 1          , Modbus_Daniels , R1
```

5.3 Client Side Map Descriptor Parameters

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 Section 4
Data_Array_Offset	Starting location in Data Array.	0 to (Data_Array_Length-1) as specified in Section 4
Function	Function of Client Map Descriptor.	RDBC – Node Type set to Block Mode; FC16 is used; length of 1 is used WRBC – FC16 is used; length of MD is used WRBX – FC16 is used; length of MD is used

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 “Client Node Descriptor” above
Data_Type*	Data type.	Register, Coil, AI, DI
Length	Length of Map Descriptor.	1-125 (Register, AI) 1-800 (Coil, DI)
Address	Starting address of read block.	0000 – 6999 (Coil) 7000 – 7999 (Float) 10000 (DI) 30000 (AI) 40000 – 49999 (Register)
Data_Array_Low_Scale*	Scaling zero in Data Array.	-32767 to 32767, 0
Data_Array_High_Scale*	Scaling max in Data Array.	-32767 to 32767, 100
Node_Low_Scale*	Scaling zero in Connected Node.	-32767 to 32767, 0
Node_High_Scale*	Scaling max in Connected Node.	-32767 to 32767, 100

5.3.3 Timing Parameters

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

5.4 Map Descriptor Example

Map_Descriptor_Name	Data_Array_Name	Data_Array_Offset	Function	Node_Name	Address	Length	Scan_Interval
CMD_AI	DA_AI	0	RDBC	PLC 1	30000	20	1s

In the above example:

- **Data_Array_Name** – The data array where the read data will be stored is specified here. Must be previously declared in **Section 4**.
- **Function** – Note the use of the RDBC function here which will cause this map descriptor to continuously read data every scan interval from the remote device. The function will also execute writes from the Data Array to the remote device when a change is detected, but only when used in conjunction with writeable addresses.
- **Node_Name** – This Map Descriptor will only respond to the node settings defined under **Section 5.2**.