MSA
The Safety Company

fieldserver

Driver Manual

FS-8700-145 Firecom

APPLICABILITY & EFFECTIVITY
Effective for all systems manufactured after March 2021.

Driver Revision: 1.01
Document Revision: 4.A

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

The Firecom Protocol driver allows the FieldServer to store data received from devices over RS-232 using the Firecom LSN 2000 ASCII protocol. The Firecom system provides detection, monitoring and control of various fire and safety devices manufactured by the Firecom Inc. (www.firecominc.com).

The FieldServer can emulate a Client Firecom device which processes unsolicited messaged from the Firecom panel. The FieldServer cannot send any messages to a Firecom device and is therefore a Passive Client only device.

The driver supports messages sent from the Firecom Life Safety Net 2000 panel. The FieldServer, connected to a single Firecom panel, is capable of processing any messages received on that connection across any Node, Slot and Address ranges depending on the configuration used and number of data points supported. The Driver supports a subset of Firecom commands, all other commands are discarded. The Driver also ignores the message code and time parameters of any received packet. The supported Firecom commands and the respective states are listed in **Section 5.1**.

2   Hardware Connections

2.1   RS-232 Connection to Firecom LSN 2000 Panel
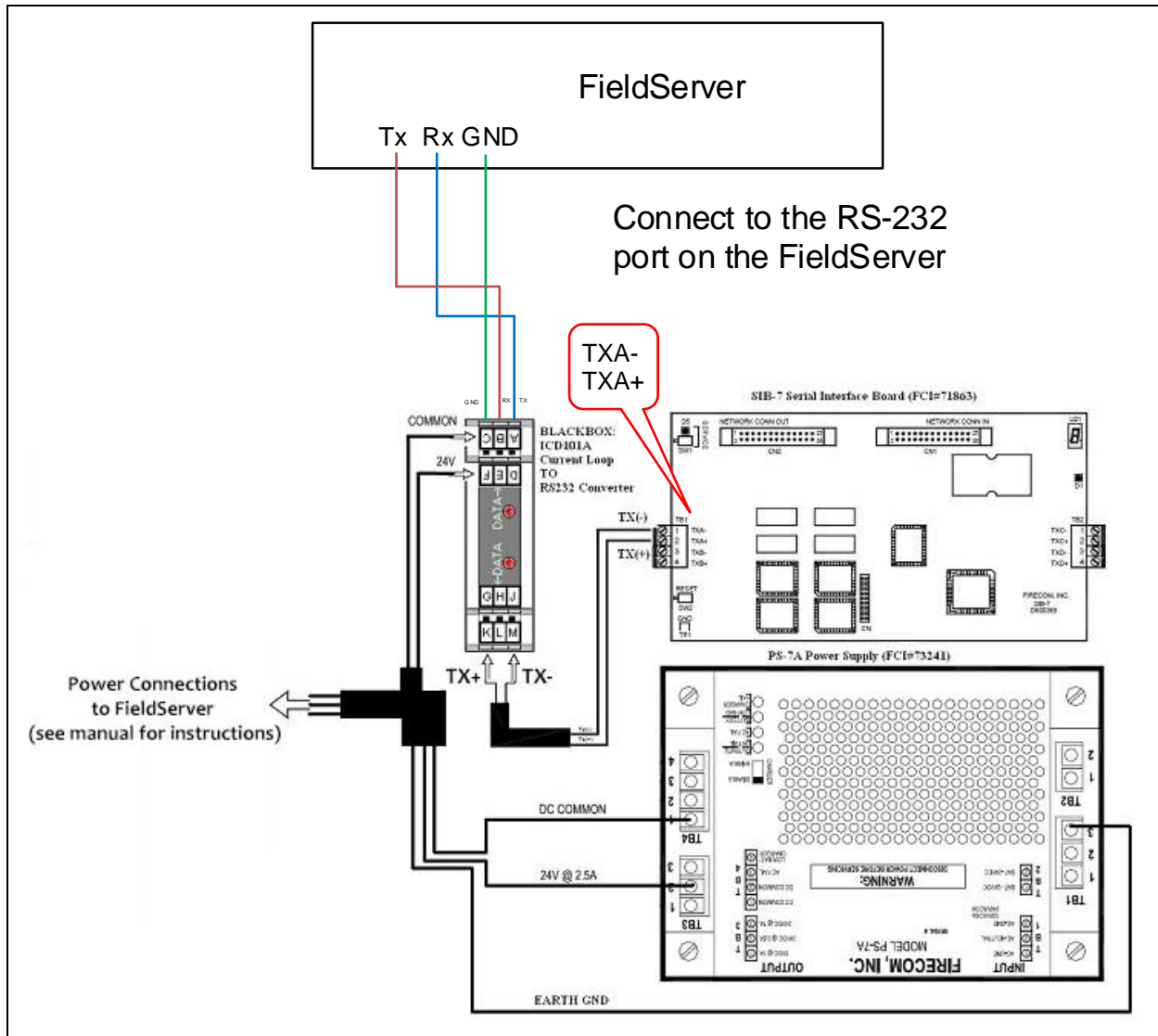
The FieldServer is connected to the Firecom panel via the sib-4 serial interface board port a, b, c or d. The port must be configured as Firecom ASCII protocol in the Firecom ISN 2000 data entry program.

3   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. The Firecom driver always stores a single byte value between 0 and 255, thus it is recommended to use byte format Data Arrays.	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    , Data_Array_Length
DA_FC_01           , Byte           , 255
DA_FC_02           , Byte           , 20
```

4   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 FieldServer).

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

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Firecom Device 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 tables below, * indicates an optional parameter, with the bold legal value as default.

4.1   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]
Baud*	Specify baud rate.	**9600** (Vendor limitation)
Parity*	Specify parity.	**None**, (Vendor limitation)
Data_Bits*	Specify data bits.	**8** (Vendor limitation)
Stop_Bits*	Specify stop bits.	**1** (Vendor limitation)
Protocol	Specify protocol used.	Firecom (makes the port the exclusive domain of Firecom devices)

Example

```
//   Client Side Connections
Connections
Port    , Baud    , Data_Bits    , Stop_Bits    , Parity    , Protocol
P1      , 9600    , 8            , 1            , None      , Firecom
```

---

[1] Not all ports shown may be supported by the hardware. Consult the appropriate Instruction manual for details of the hardware.

4.2   Client Side Node Descriptors

Section Title	
Nodes	

Column Title	Function	Legal Values
Node_Name	Provide name for node.	Up to 32 alphanumeric characters
Protocol	Specify protocol used.	Firecom
Node_ID	This parameter tells the driver which Firecom node to monitor on this FieldServer. Note that one node needs to be created for each Firecom node to be monitored.	0 to 255

Example

```
//   Client Side Nodes
Nodes
Node_Name     , Node_ID     , Protocol
Firecom_00     , 0               , Firecom
Firecom_70     , 70             , Firecom
```

4.3   Client Side Map Descriptor Parameters

4.3.1   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.	Passive_Client
Node_Name	Provide name for Node.	Up to 32 alphanumeric characters

4.3.2   Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Slot	This parameter specifies which Firecom Slot will be associated with this Map Descriptor.	0 to 9
Address	This parameter specifies which Firecom address (or starting address when used with a length greater than 1) is to be associated with this Map Descriptor. This field, together with the Slot and Node's Node_ID, identifies the Firecom points to map to the specified data array.	**1** - 255
Firecom_store_option*	This parameter specifies whether the Firecom command code or the respective state will be stored in the specified data array.	**Store_command** Store_state
Length	Length of Map Descriptor.	1 - 999

4.4   Map Descriptor Examples

The driver processes messages from a Firecom panel that relays the events for any device on the Firecom network. A Firecom message contains several fields to uniquely identify the device that triggered an event. The driver uses these fields to determine which device or range of devices to monitor for events. A Map Descriptor is specific to a single Node and Slot, but can include a range of addresses, so a Map Descriptor needs to be created for each Node and Slot range that is of interest.

The Firecom panel has several codes that correspond to events, but all of these can be related to one of three states (*Normal, Alarm or Trouble*) that the device is in after an event occurred. The Map Descriptor determines whether the original Firecom command or the relevant state is stored to a Data Array. These values can then be directly read by any other Client devices.

4.4.1   Store Firecom Command

The driver will use this Map Descriptor to store the Firecom command codes to a Data Array. This Example assumes there is a Node FC_node_1 configured with Node_ID = 1. The command code of any message received with the Node = 1; Slot = 0 and any address between 10 and 29 will be stored to data array DA_1. The storage offset is determined from the Address parameter. Thus for a message with address parameter 15, the command will be stored to location 5 of the Data Array.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name  , Data_Array_Offset  , Function       , Node_Name  , Slot  , Address  , Length  , Firecom_store_option
FC_command            , DA_1             , 0                  , Passive_Client , FC_node_1  , 0     , 10       , 20      , Store_command
```

In the above example:

- Data_Descriptor_Name – Each Map Descriptor can be given a unique name.
- Data_Array_Offset – The data will be stored from position 0 in the data array.
- Node_Name – The Node must be set up for the correct Node number.
- Address – Monitor messages from devices with addresses starting from 10.
- Firecom_store_option – The Firecom Command will be stored.

4.4.1   Store State Command

The Driver will use these Map Descriptors to store the representative state of the mapped device to Data Array DA_1 and the Firecom command to DA_2.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name  , Data_Array_Offset  , Function       , Node_Name  , Slot  , Address  , Length  , Firecom_store_option
FC_command            , DA_1             , 0                  , Passive_Client , FC_node_1  , 0     , 10       , 20      , Store_state
FC_command            , DA_2             , 0                  , Passive_Client , FC_node_1  , 0     , 10       , 20      , Store_command
```

5   Driver Notes

5.1   Driver Scope

This driver supports a subset of the possible Firecom commands, all other commands are discarded. The Driver also ignores the message code and time parameters of any received packet. The supported Firecom commands and the respective states are listed in the table below.

Received Message Type (command)	Respective Point State After Message	Firecom Command Data Array Value	State Data Array Value
Supervisory (156)	ALARM	156	1
Supervisory Reset (157)	NORMAL	157	0
Alarm (160)	ALARM	160	1
Alarm Reset (161)	NORMAL	161	0
Trouble (162)	TROUBLE	162	2
Trouble Reset (163)	NORMAL	163	0
Warden Station Alarm (184)	ALARM	184	1
Warden Station Alarm Reset (185)	NORMAL	185	0
System Restart message: "F0000000000000xxxxxxxxxxxxxxxx"	All points: NORMAL	0	0
All Nodes Offline: "FEEEEEEEEEEEEEEEEEEEEEEEEEEEE"	All points: OFFLINE	255	255

5.2   Driver Stats

Firecom panels produce data messages for slave devices to consume. The messages can come at any time depending on the events that occur on the network. If no events occur the panel will send out a keep-alive message approximately every 5 seconds. If no message is received for 12 seconds, it is assumed that at least 2 alive messages were missed and all Nodes on the Firecom connection will be marked as offline. The driver counts all incoming messages of interest as the PLC_READ_MSG_RECD statistic.