FieldServer Driver - Serial
FS-8700-40
Siemens Fire Safety MXL

Devices Supported: Siemens Fire Safety MXL system with NIM-1R Network Interface Module and 555-193085 cable from Siemens Cerberus

Physical Interface: EIA232 with NIM-1R configured for Foreign System Interface (FSI) by setting all the switches in SW2 to open (or OFF)

Overview

This FieldServer driver can be used to poll a Siemens Cerberus MXL system or to emulate a Siemens Fire Safety MXL system with attached modules. Either configuration only supports the remote monitoring commands which are adequate for remote data and status information collection.

Client Configuration File Structure

In FSI mode, the NIM-1R allows the FieldServer to gather data from multiple (up to 63) Cerberus Panels when they are connected together with an MXL network. The default configuration that comes in the FieldServer will monitor two panels (1 and 2) with 8 modules (1 to 8) each.

There are two sets of data that are collected by the Cerberus driver. The first is a collection of 14 counters per panel. Each 16-bit counter is incremented when ever the corresponding event occurs. A client can read these counters to quickly determine if a new event has been reported to the server. The counters are in the following order:

Event	Offset
Fire Alarm In	0
Fire Alarm Out	1
Fire Alarm Acknowledge	2
Trouble In	3
Trouble Out	4
Trouble Acknowledge	5
Supervisory In	6
Supervisory Out	7
Supervisory Acknowledge	8
Security Alarm In	9
Security Alarm Out	10
Security Alarm Acknowledge	11
Status In	12
Status Out	13

The second set of data is a collection of bit maps that can be queried to determine which device has reported the event. When an alarm event arrives from a device, the corresponding bit in two different arrays is set. The bit in one array indicates the alarm has occurred, and the bit in the other array indicates that the alarm has not been acknowledged. When the alarm clear event arrives, the bit in the alarm array is cleared. If an alarm acknowledge event arrives, the bit in the un-acknowledged array is cleared. Each of these arrays are optional. To enable one, there must be a map descriptor in the configuration that has an address that corresponds to the array as shown in this table:

Array	Address
Fire Alarm	1
Fire Alarm Un-Acknowledged	2
Trouble Alarm	3
Trouble Un-Acknowledged	4
Supervisory Alarm	5
Supervisory Un-Acknowledge	6
Security Alarm	7
Security Alarm Un-Acknowledged	8
Status	9

Note:
The client driver can only show the status of alarms and acknowledgments. It cannot send a “clear alarm” or “acknowledge alarm” message to cerberus equipment to actually acknowledge or clear alarms.

The map descriptors are used to determine which bit to set in the data array. When an event arrives from a panel, the FieldServer will scan for map descriptors with the Node_ID matching the panel number reported in the event. This map descriptor will point into a data array that will be modified. The map descriptor can further refine the location by specifying a starting module within the panel. This is specified in the Block_Number field. Each module represents 256 bits that can be set or reset by an event. A map descriptor that is 512 bits long with a starting module (Block_Number) of 3 represents the events for Module 3 and 4. A map descriptor that is 10 bits long will only map the first 10 devices for that module. The combination of Block_Number and Data_Array_Location can be used to map multiple panels with various numbers of modules into a packed array easily read by server side of the FieldServer.

When the customer receives a FieldServer with the Cerberus driver installed, the configuration has been set to poll 2 panels (1 and 2) once a second. As well as two sets of event counters, there are alarm bit arrays set up for 8 modules (1 to 8) per panel. If the customer’s configuration fits within this size, he needs only to configure the Server side of the FieldServer, the interface to the non-Cerberus device. As an example, if the user were connecting to a device using Modbus RTU communications protocol, they would configure the interface just like they would do if the FieldServer were another Modbus PLC. They would create a map descriptor to assign a Modbus address (such as 40001) to the Counts data array so that the Modbus bus master could read it.

If the customer has different panel numbers, the Nodes section of the configuration file will need to be changed. If the customer has more panels, Node entries and map descriptors will need to be added by copying the original ones and modifying the Panel number entries and data array offsets.

To access panels with more modules, first increase the data array lengths. If the extra modules are contiguous, the map descriptors for that panel can have their lengths increased also. If there are large

gaps in the numbering of the module, it will be more efficient to add additional map descriptors by copying the ones that are given and modifying the block number to start at the next existing module number.

Server Configuration File Structure

The driver can also be used to emulate a MXL server. Other protocol drivers could then poll remote devices and access the local MXL server data to set or clear events. In this configuration an existing Siemens Cerberus MXL panel could be replaced with an MXL emulation. Existing clients could poll the emulation driver on the FieldServer to get the same data as from a conventional MXL server.

Up to 100 panels can be emulated with the driver. Each panel has to be on a unique port and have a unique node id assigned. A map descriptor has to be defined for each type of remote device. The following types can be used:

Device Type
Fire Alarm In
Fire Alarm Out
Fire Alarm Acknowledge
Trouble In
Trouble Out
Trouble Acknowledge
Supervisory In
Supervisory Out
Supervisory Acknowledge
Security Alarm In
Security Alarm Out
Security Alarm Acknowledge
Status In
Status Out

Revision History

Date	Driver Version	Document Revision	Resp	Comment
6/24/02	1.01b			Releasing