FieldServer Driver
FS8705-21
Driver for
Integlex Fire Alarm Panel

Description

The Integlex driver is a passive client driver intended for connection to the serial port of a Integlex FACP. A passive client driver waits for messages to be sent to it (by the panel). The driver cannot send messages to the panel and hence it cannot request the state of any point in the panel.

The driver can process alarm and trouble events, system troubles and some other messages from the panel.

The driver can only be used as a client. Minimal server functionality is provided only to support our ongoing quality assurance program by facilitating automated testing of the driver. Server operation is not documented or supported.

Synchronization

A consequence of the fact that this is a passive client driver, is that the FieldServer must be synchronized with the panel by restarting the FieldServer and then resetting the panel to reset and resend the active data.

Driver Functionality

When an event message is received by the FieldServer the FieldServer will inspect the message to learn the Node Number and for device events it will also learn the Loop, Device and Zone Numbers.

The driver will determine if the event reports an on/off state and will set/clear an element of its data arrays (internal database) to correspond to the state of the point being reported. These Data Array elements are mapped onto the server protocol such as BACnet IP.

Available Information for Mapping

Panel Troubles
For Panel Troubles, there is one Data Array of troubles configurable for each FACP on the network. The list of troubles for each panel is fixed, but a capability has been added within the driver to add to the standard list of troubles if needed.

Device Activation Status
For Device Activation Events data points are mapped in based on specific activation status type, node and loop. When the event message is received, the driver uses the node, loop number and the activation status (Active/ Inactive Status part of the message), looks through the configuration and finds the Data Array to use for storage. It then uses the device number as the offset into the Data Array and sets or clears the data at that offset depending on whether it is an active/inactive status event. A system reset will also clear device status where relevant.

Device Trouble Status
For Device Trouble Events data points are mapped in based on specific activation status type, node and loop. When the event message is received, the driver uses the node, loop number and the Trouble status (Active/ Inactive Status part of the message), looks through the configuration and finds the Data Array to use for storage. It then uses the device number as the offset into the Data Array and sets or clears the data at that offset depending on whether it is an active/inactive status event.

Device Summary Status
For all Device Events the driver can provide a summary to report if a device has any active status. There are 2 summary systems – one uses different bits for each possible event so activations and troubles can be stored in the same data element. This is known as the Full Summary. The other (partial summary) requires different array items for activations and troubles to allow for unique activation and trouble summaries.

Device MultiState Status
The MultiState method of reporting is also a summary but instead of using a different bit in each word used to represent a device, the Multistate method uses a number. For activations the number reports 1) normal or 2) the highest priority activation. For Troubles the number reports 1) normal 2) the trouble type (an enumeration) or 3) that there is more than one trouble active for the device.

Device Classification
When device messages are received, the driver looks in the classification part of the message. It then uses a lookup table, finds a matching classification in the table, extracts the classification number and stores the number in the Data Array configured to store classification numbers. Note that due to the capture method, the driver can only report classifications of devices that have changed state.

Database Rest through System Reset
When a system reset is received, the driver clears all previously set data in any data array configured to store data from the Integlex panel. If there are points in an active state, when the system reset is performed, the panel will send new messages reporting the active states and thus the FieldServer data and the panel data will be synchronized.

Unsupported Features

Simulation Notification
Some messages produced by the panel indicate the event was generated by simulation. The driver will ignore the fact that these are simulated events and will set / clear data as if they were non-simulated events.

Zone reporting
Since the Protocol specification does not allow for the reporting of specific zone status directly, this feature has not been supported. Individual events do report zones, but since they only do it on change of status any attempt to capture zone status using these messages would be incomplete.

Switch Operations
Although the driver uses System Reset for management of supported features, the actual reporting of operations made at the panel (Acknowledge, Drill, Silence, System Reset etc) is not possible.

Menu Operations
The reporting of Menu operations made is not supported.

ZOCU and OCU Status
The status of ZOCU and OCU LED's and Pushuttons are not monitored.

Other
Any features not specifically stated as supported in this document should be regarded as not supported. Please feel free to contact us for further information.

Max Nodes Supported

FieldServer Mode	Nodes	Comments
Passive Client	1	*Each port on the FieldServer can only be connected to 1 panel. Multiple networked panels can be supported through this connection to one panel.*
Active Server	0	*Not supported or documented.*

Formal Driver Type

Serial
Passive Client

Compatibility Matrix

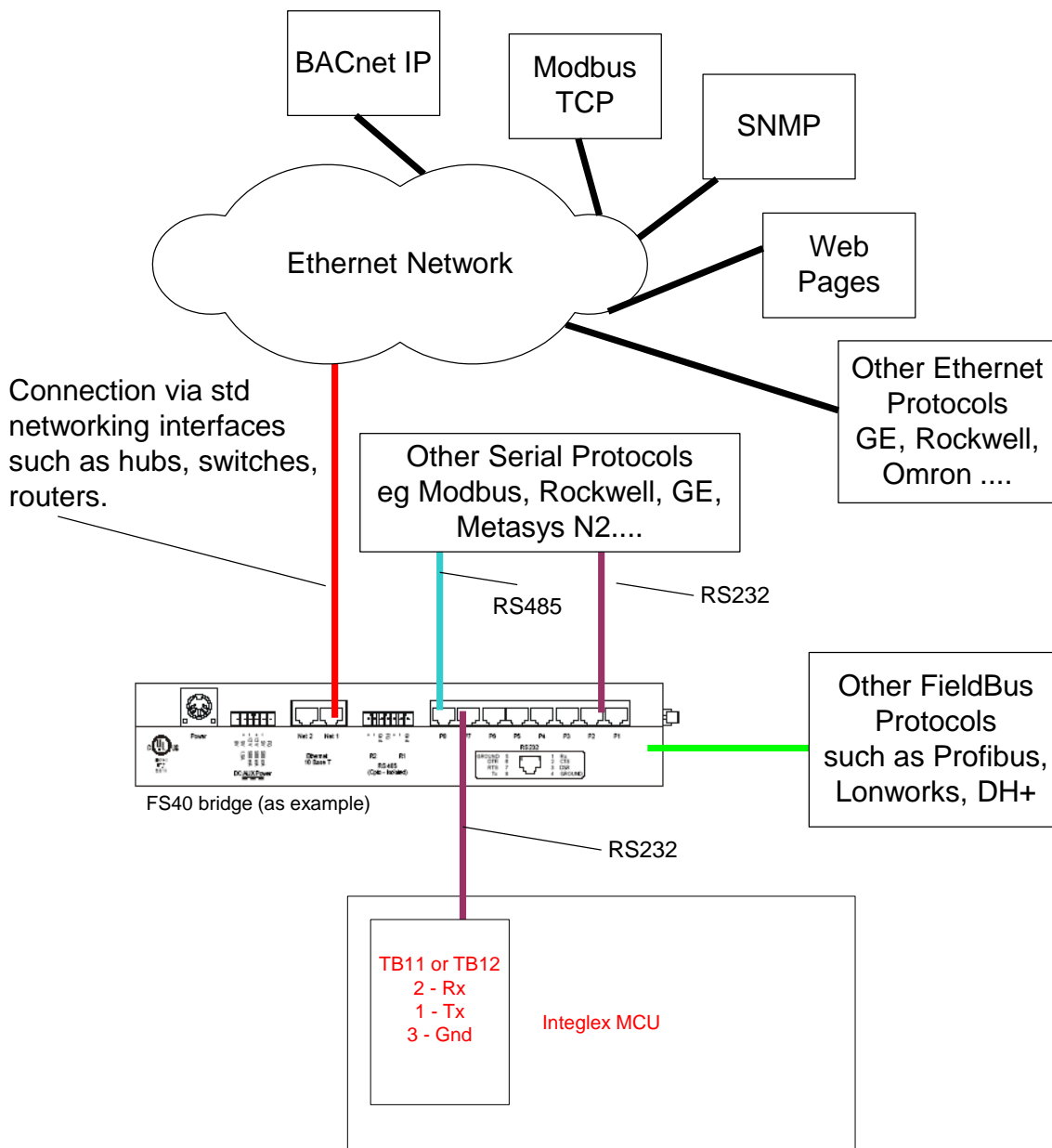| FieldServer Model	Compatible with this driver
FS-x2010	No
FS-x2011	No
FS-x40	No
FS-X30	Yes
QuickServer	Yes

Connection Information

Connection type:	EIA232
Baud Rates:	Driver Supports : 110; 300; 600; 1200; 2400; 4800; 9600; 19200; 28800; 38400; 57600 Baud *Integlex* supports**: 9600**
Data Bits:	Driver Supports : 7,8 *Integlex* supports: **7**
Stop Bits:	Driver Supports : 1,2 *Integlex* supports: **1**
Parity:	Driver Supports : Odd, Even, None *Integlex* supports: **Even**
Hardware interface:	N/A
Multidrop Capability	No

Devices tested

Device	Tested (FACTORY, SITE)
Integlex FACP UL864 9th Edition	Sierra Foothills (September 2012)

Connection configurations

Multiple workstation protocols and connections supported. See list of FieldServer Drivers.

BACnet IP

Modbus TCP

SNMP

Ethernet Network

Web Pages

Other Ethernet Protocols GE, Rockwell, Omron ....

Connection via std networking interfaces such as hubs, switches, routers.

Other Serial Protocols eg Modbus, Rockwell, GE, Metasys N2....

RS485

RS232

Other FieldBus Protocols such as Profibus, Lonworks, DH+

FS40 bridge (as example)

RS232

TB11 or TB12
2 - Rx
1 - Tx
3 - Gnd

Integlex MCU

Support

This driver was developed by Chipkin Automation Systems (CAS), a FieldServer Approved Integrator®. CAS are proud to provide support for the driver. For support please call CAS at (866) 383-1657.

Revision History

Date	Resp	Format	Driver Ver.	Doc. Rev.	Comment
2012Jul12	PMC		0.00	0	Created
2012Jul26	GFM		0.00	1	Clarified Driver Support
2012Aug30	PMC		0.00aD	2	Updated notes on summary.
2012Sep10	GFM		1.00aA	3	QC