FieldServer
FS-8700-48 Fike Cheetah

Driver Manual

(Supplement to the FieldServer Instruction Manual)

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after November 2015

Kernel Version: 6.03
Document Revision: 1

Contact Information:

Thank you for purchasing the FieldServer.

Please call us for Technical support of the FieldServer product.

Contact Information:

Sierra Monitor Corporation
1991 Tarob Court
Milpitas, CA 95035

Contact number:
+1 408 262-6611
+1 800 727-4377

Email: info@sierramonitor.com

Website: www.sierramonitor.com

TABLE OF CONTENTS

LIST OF FIGURES

1 CHEETAH DEVICE DESCRIPTION

The Cheetah Protocol driver allows the FieldServer to transfer data to and from devices over either RS-232 or RS-485 using the Cheetah device protocols (Legacy Cheetah Classic, Cheetah Xi and Cheetah MIM).

The driver supports messages sent from the Cybercat panel. Specifically, the driver supports message 1.02 which reports panel, zone and device states.

The FieldServer can emulate either a Server or Client but it should be noted that it can only process unsolicited messages from the Cheetah devices. Thus, it does not provide an active Client driver. It is best to consider this driver as a consumer only driver with the data being produced by a Cheetah controller.

2 DRIVER SCOPE OF SUPPLY

2.1 Supplied by Sierra Monitor Corporation for this driver

Sierra Monitor Corporation PART #	Description
21723-1	Flat 6 way cable (7 foot) for RJ11 connection
FS-8915-10	UTP cable (7 foot) for RS-232 use
FS-8917-04	RJ45 to DB25F connector adapter
FS-8917-14	RJ11 to DB25M connection adapter

2.2 Provided by Supplier of 3rd Party Equipment

Part #	Description
	Fike Cheetah panel and power supply.
	RS-485 cable, if required.
	Longer UTP cable, if required.

3 HARDWARE CONNECTIONS[1]

The FieldServer is connected to the Cheetah panel's Peripheral Port (see below).

This port is either a serial port (identified as jack 4) or RS-485 port (use the terminals adjacent to jack 4).

Configure the Cheetah Panel according to manufacturer’s instructions

3.1 Connection to Fike Cheetah Panel

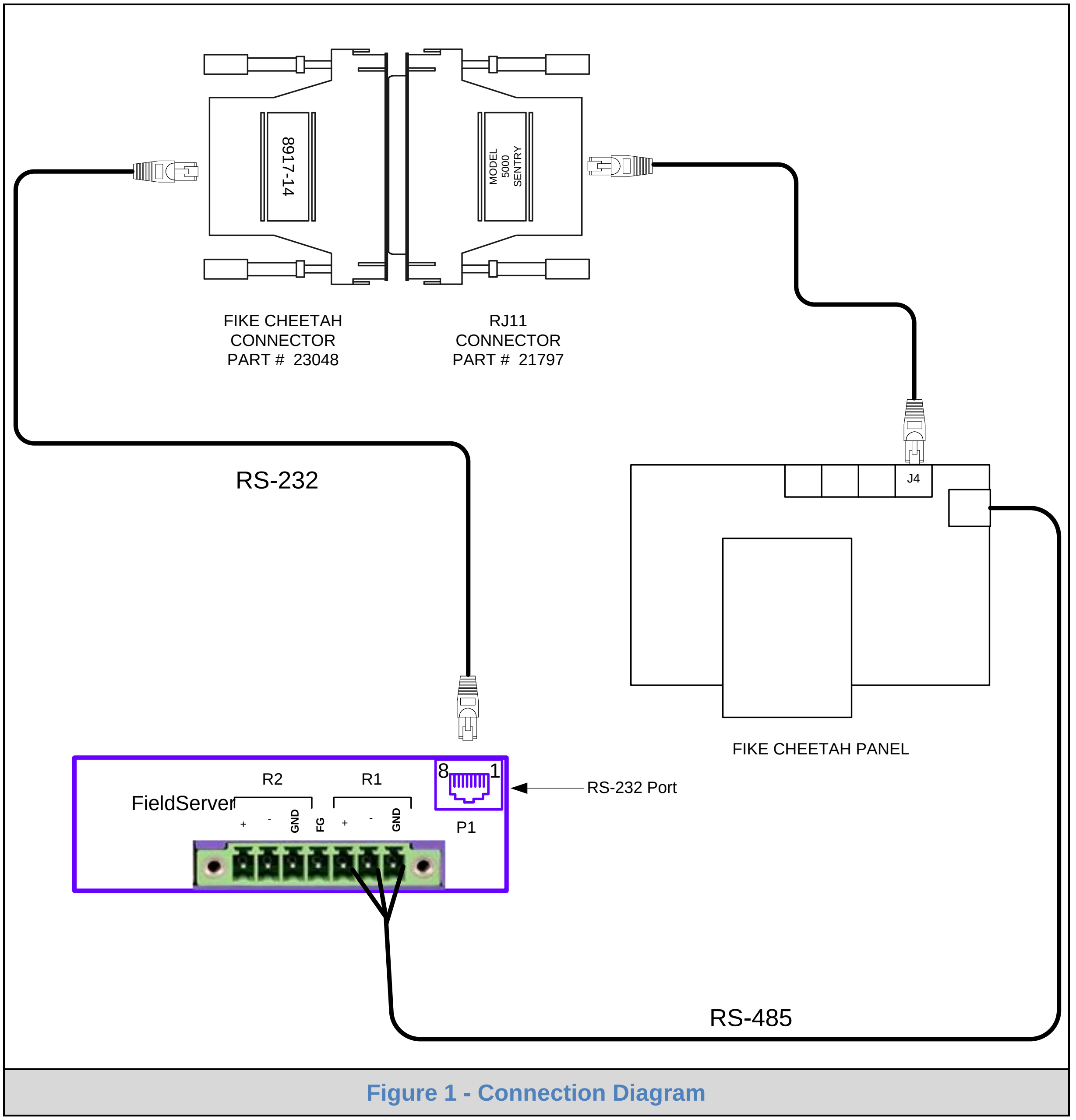


Figure 1 - Connection Diagram

[1] The connection diagrams are for Non Cybercat Panels. Connection info was not available at the time this document was last updated.

3.2 RS-485 Connection to Fike Cheetah Xi Panel

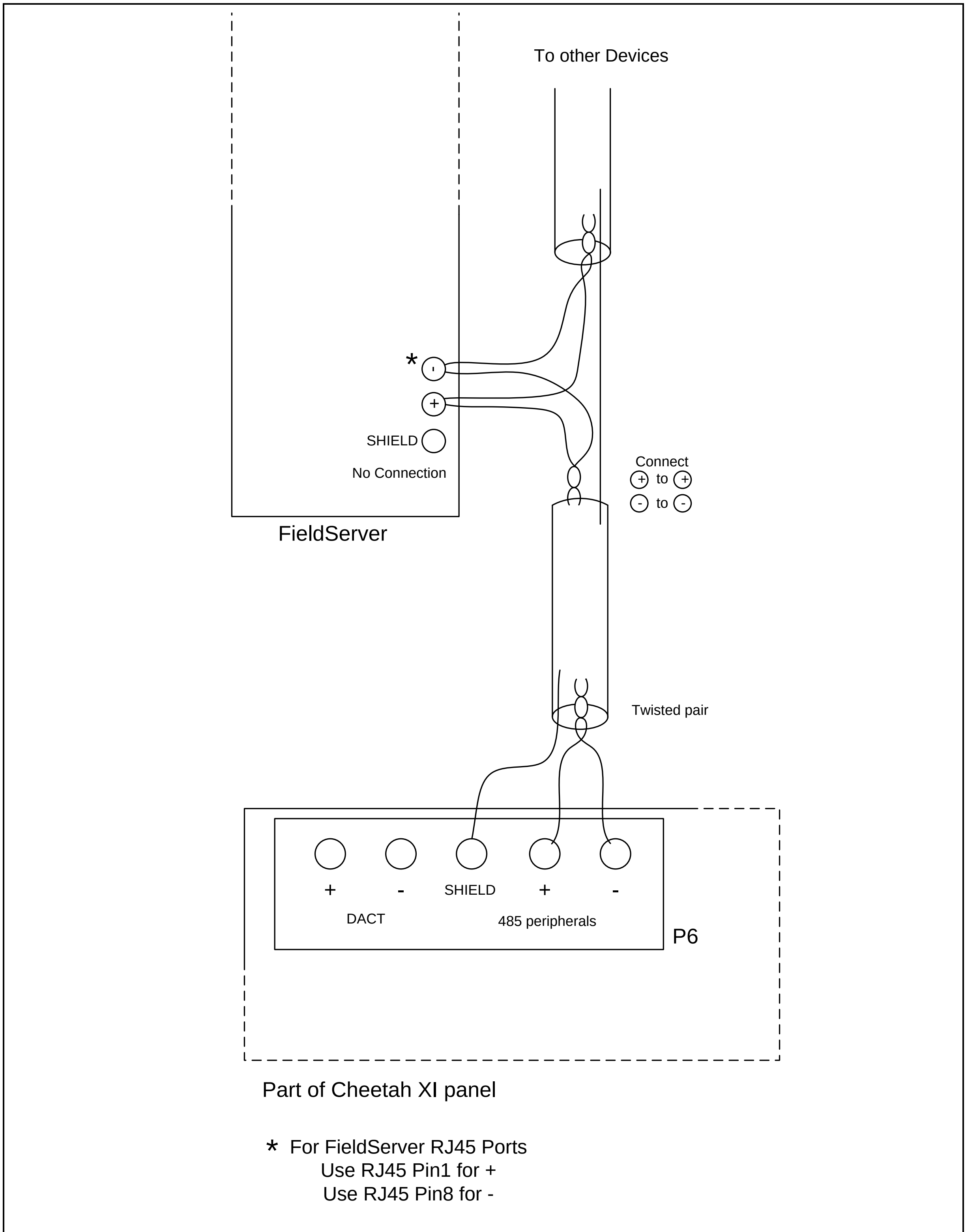
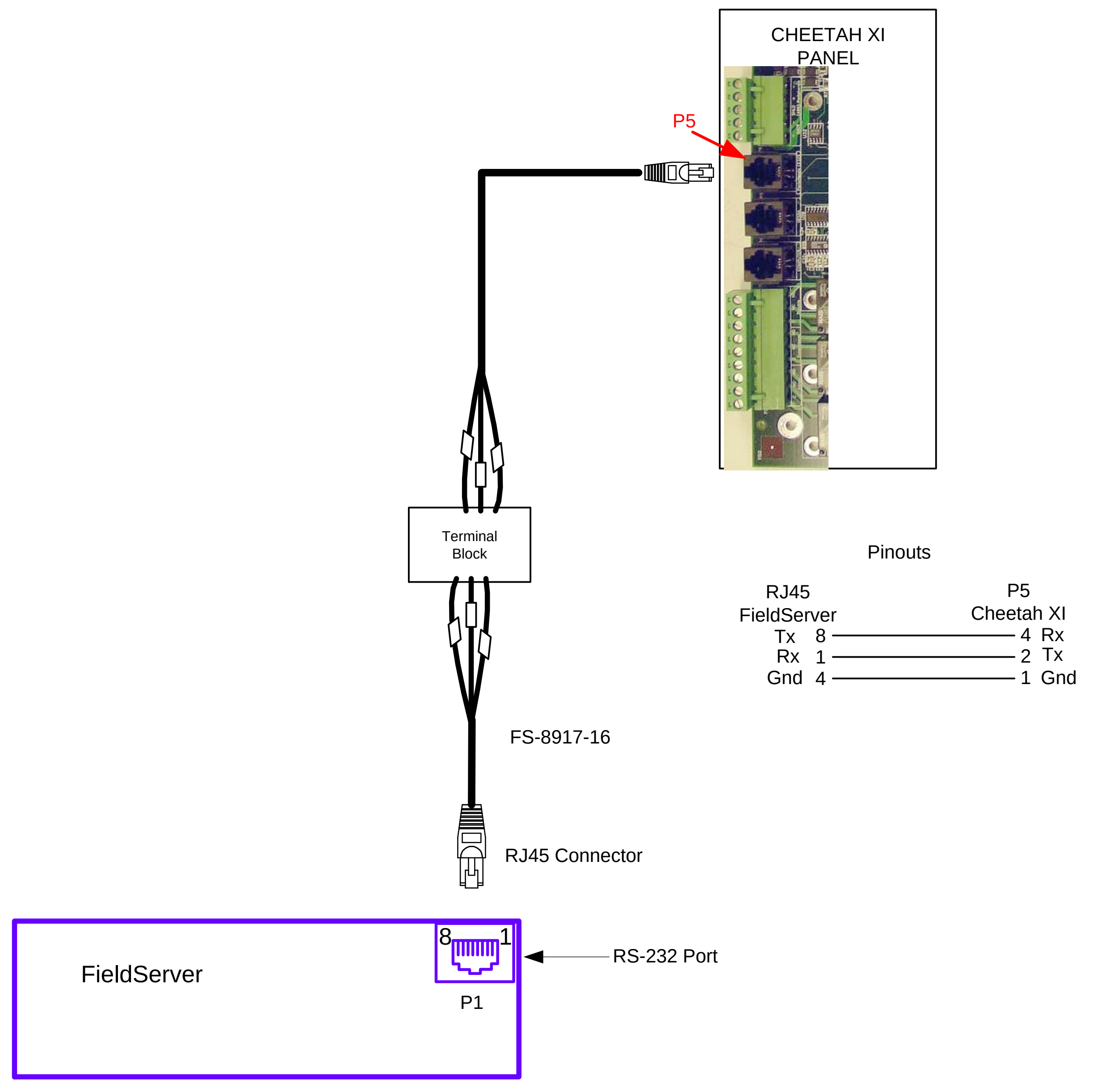


Figure 2 - Connection Diagram from RS-485 to Fike Cheetah Xi Panel

3.3 RS-232 Connection to Fike Cheetah Xi Panel



Pinouts

RJ45 FieldServer		P5 Cheetah XI	
Tx	8	4	Rx
Rx	1	2	Tx
Gnd	4	1	Gnd

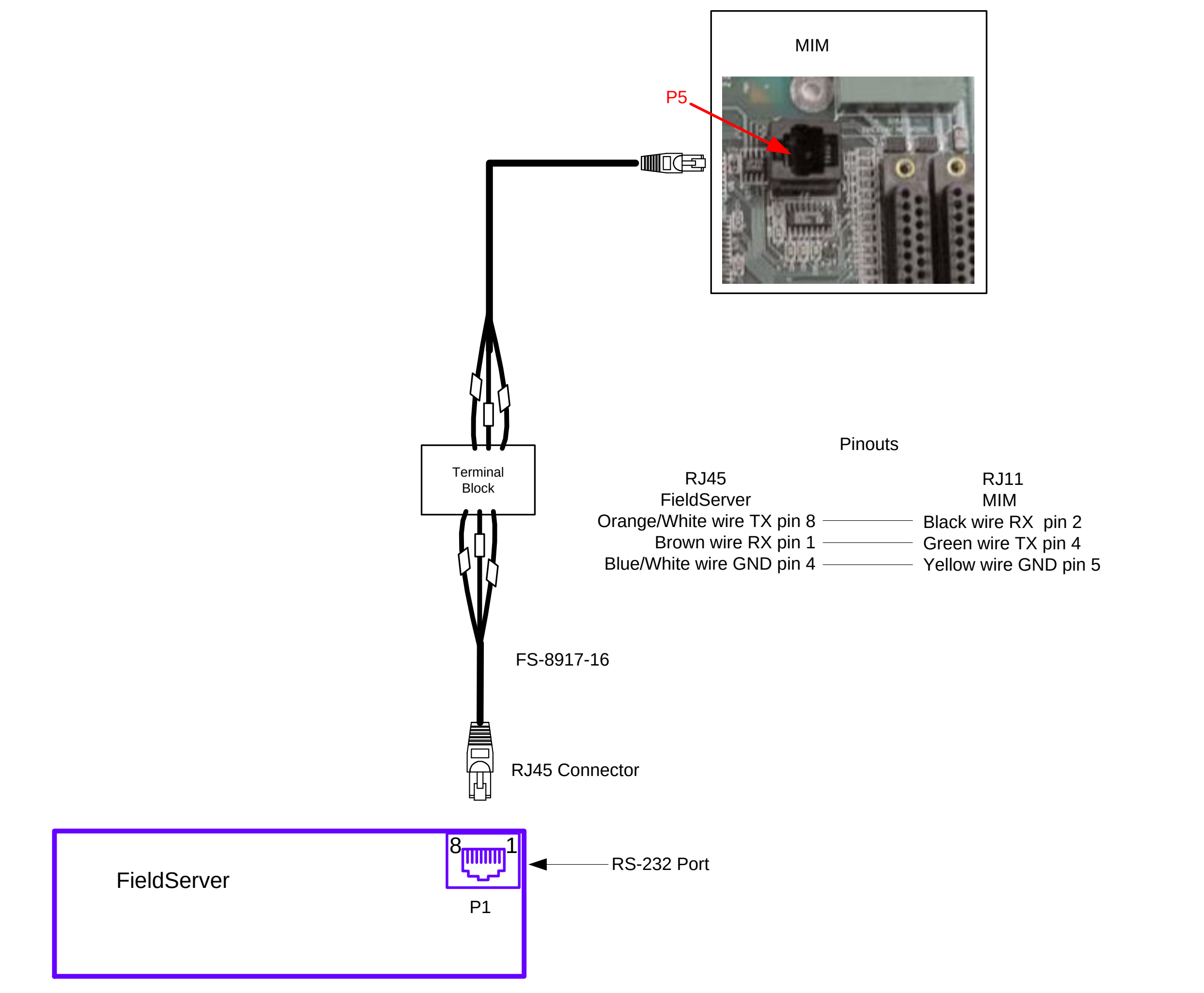
Figure 3 - Connection Diagram from RS-232 to Fike Cheetah Xi Panel

3.3.1 Connection Notes:

- Communication on the RS-232 port is untested.
- We recommend that a RS-232 optical isolator is used on the connection between the FieldServer and the Cheetah XI panel to isolate any potential ground differential issues.

3.4 Connecting the FieldServer to the MIM (Multi-Interface Module)

- The RS-232 port of the FieldServer connects to the P5 (RJ11) RS-232 port of the MIM board.



Pinouts

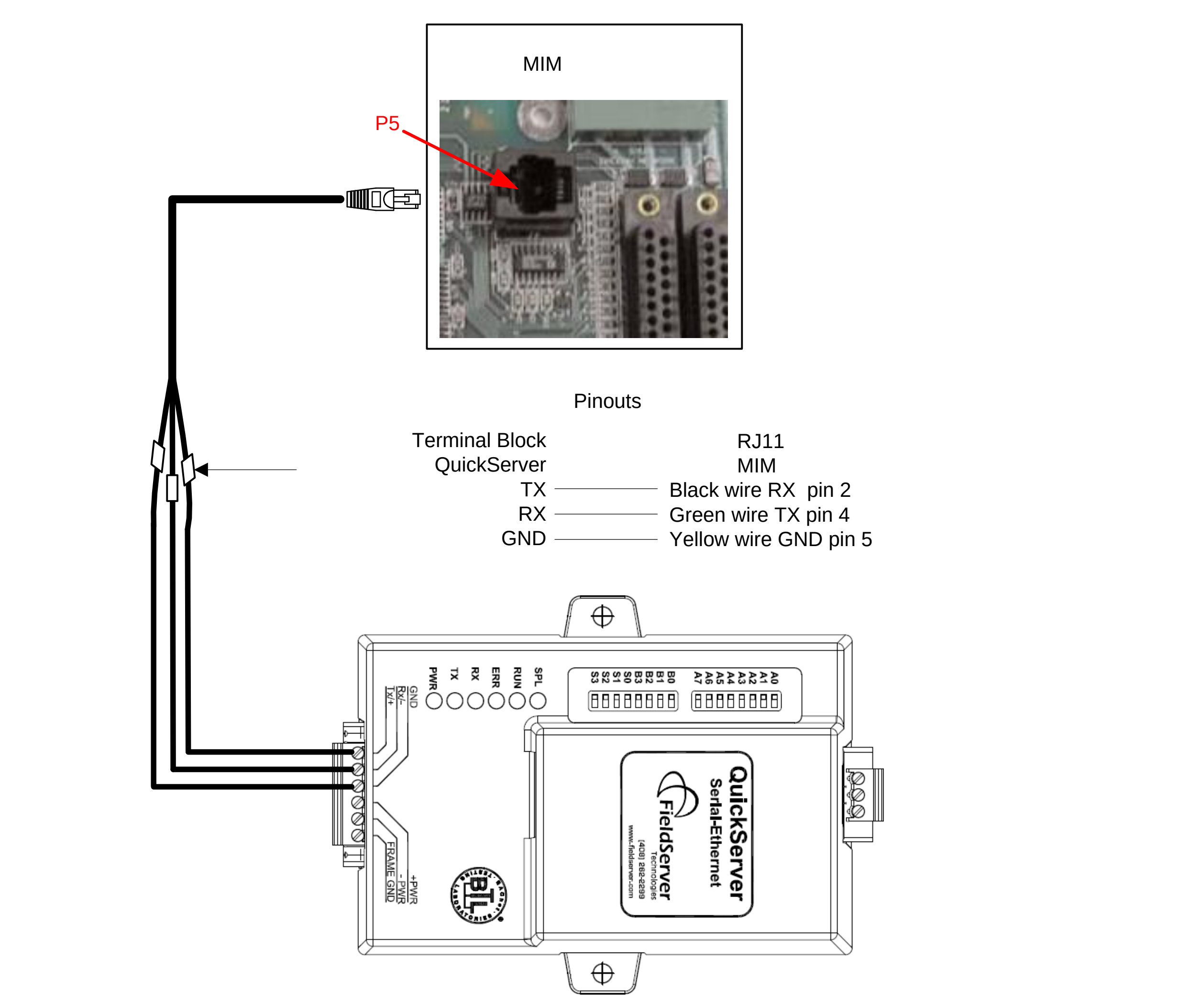
RJ45 FieldServer	RJ11 MIM
Orange/White wire TX pin 8	Black wire RX pin 2
Brown wire RX pin 1	Green wire TX pin 4
Blue/White wire GND pin 4	Yellow wire GND pin 5

Figure 4 - Connecting a FieldServer to the MIM

3.4.1 Connection Notes

- The Peripherals menu of the Fike Panel needs to be updated:
  - Hit ‘ESC’ until “Top Level Menu” is on the screen
  - Hit ‘F1’ for “Config”
  - Hit ‘F6’ for “Menu 2”
  - Hit ‘F6’ for “Menu 3”
  - Hit ‘F1’ for “Periph”
  - Hit ‘F1’ for “Device”
  - Choose address of MIM
  - Set “Type” to “Computer”
  - Set “Supervise” to “No”

3.5 Connecting the QuickServer to the MIM (Multi-Interface Module)



Pinouts

Terminal Block QuickServer	RJ11 MIM
TX	Black wire RX pin 2
RX	Green wire TX pin 4
GND	Yellow wire GND pin 5

Figure 5 - Connecting a QuickServer to the MIM

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. The Cheetah driver always sets Data Array elements to a zero or one. Thus, the use of bit arrays is suggested but is not mandatory.	Float, BIT, UInt16, SInt16, Byte.
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.	1-10,000. If you use the 'All' keyword when setting the parameter Cheet_Zone/Device then the minimum length is 128.

Example

```
//   Data Arrays
Data_Arrays
Data_Array_Name      , Data_Array_Format    , Data_Array_Length
ZONE_ALARMS          , Bit                  , 256
PANEL_DATA           , Float                , 1000
DA_HIST              , Float                , 1000
DEVICE_L1_STATE      , Float                , 256
```

5 CONFIGURING THE FIELDSERVER AS A CHEETAH DEVICE 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 Cheetah Device Client.

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for Cheetah 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 "Server Side Nodes" section and the data required from the Client needs to be mapped in the "Server Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the 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-P8, R1-R2[2]
Baud	Specify baud rate	38400
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	Cheetah (makes the port the exclusive domain of Cheetah devices.) This keyword is not required when specifying the port.
MIM_Enabled*	Enable multi-panel communications via the MIM module[3]	Yes, **No**

Example

```
//    Client Side Connections

Port    , Baud     , Protocol    , Parity    , Data_Bits     , Stop_Bits     , MIM_Enabled
P1      , 38400    , Cheetah     , None      , 8             , 1             , Yes
```

[2] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

[3] Node_ID for each panel/Node must be specified when MIM is enabled

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*	Specify Remote Networked panel's Node_ID. Only used when MIM_Enabled = Yes	1-255
Protocol	Specify protocol used	Cheetah
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[4]
PLC_Type*	This parameter tells the driver what type of panel you are connecting to. If not specified, the driver expects a legacy Cheetah panel.	Cybercat , **Cheetah**

Example for MIM_Enabled = No

```
//    Client Side Nodes

Nodes
Node_Name         , Protocol     , Port     , PLC_Type
Cheet_Master1     , Cheetah      , P1       , Cheetah
```

Set PLC_Type to Cybercat for Xi Panels

Example for MIM_Enabled = Yes

```
//    Client Side Nodes

Nodes
Node_Name         , Node_ID     , Protocol     , Port     , PLC_Type
Cheet_Master1     , 11          , Cheetah      , P1       , Cheetah
Cheet_Master1     , 12          , Cheetah      , P1       , Cheetah
Cheet_Master1     , 13          , Cheetah      , P1       , Cheetah
```

[4] 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 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	Passive
Node_Name	Provide name for Node	Up to 32 alphanumeric characters

5.3.2 Driver Specific Map Descriptor Parameters

Column Title	Function	Legal Values
Cheet_Zone*	A Map Descriptor may be used to store data for one, all or no zones. To store data from multiple zones, multiple Map Descriptors must be declared, each specifying the zone of interest. When specifying “All”, the data for zone 0 is stored in the first element of the Data Array defined by the *Data_Array_Name & Data_Array_Offset* and the data for zone 127 in the 128th element of the Data Array. Depending on the firmware version of the Cheetah panel some or all of the following states are available. Abort, Trouble, Supervisory, Zone Disable, Pre-Alarm, Alarm, Pre-Discharge, Release, Process. To store data for multiple states, multiple Map Descriptors must be declared - One per state of interest.	None, **All**, 0-127 Must be *None* when Cheet_Device is not equal to *None*.
Cheet_Device*	Define one or more Map Descriptors to store data from the 0-127 addressable devices. Each Map Descriptor must have the Cheet_Zone set to *None*. Devices belong to one of 4 possible loops. Thus when Cheet_Device is set to *All* or to a specific device number, the Cheet_Loop number must be set to a value from 1 to 4. If Cheet_Device is set to *All* then 128 states are stored. The data for device 0 is stored in the first element of the Data Array defined by the *Data_Array_Name & Data_Array_Offset* and the data for device 127 in the 128th element of the Data Array.	**None**, All, 0-127 Must be *None* when Cheet_Zone is not equal to *None*.
Cheet_Loop	Specify this parameter when the value of Cheet_Device is not equal to *none*.	**None**, 1-3 Must be *None* when Cheet_Zone is not equal to *None*.

Column Title	Function	Legal Values
Cheet_DT*	Data Type. Multiple Map Descriptors are required to store multiple states in one/more Data Arrays.	Abort, Trouble, Supervisory, Disable, Pre-Alarm, Alarm, Pre-Dis, Release, Process, **Any**, All, History, Panel Only *Alarm* and *Trouble* are valid when storing device data
Length	The length of the Data_Array that will be used to store the information. Ensure that the length is sufficient to store all information (e.g. for Zone Data the minimum length is 241 (Enough space for Zones 0-240).	1-10,000
Cheet_Func*	Use for active Map Descriptors only	Port Response, **-**
DA_Byte_Name	Full details of the most recent event (any device) may be stored in the data array DA_HIST_EVENT. See sections 5.4.14 and 5.4.15. Appendix D.3 maps the layout of this data.	DA_Hist_Event

MIM enabled only:		
Loop_Number	The circuit loop to store events from.	1-4
History_Event_Name	The type of history event to store data from.	Device_Trouble, Device_General, Device_Supervisory, Device_Process, System_Trouble, Device_Alarm, Vesda_Trouble, Vesda_Pre-Alarm, Vesda_Alarm, Gateway
Address	The start device on the loop to store data from.	0-254
Length	The number devices on the loop to store data from.	Device_Trouble, Device_General, Device_Supervisory, Device_Process, Device_Alarm: 1-255 System_Trouble: 24 Vesda_Trouble: 9 Vesda_Pre-Alarm: 4 Vesda_Alarm: 4 Gateway: 7
DA_Device Event_List	The Data Array to store multiple events from devices.	One of the Data Array names from Section 4
Vesda_Zone	The Vesda Zone to store events from.	1-254
Gateway_Zone	The Gateway Zone to store classic Cheetah events from.	1-254

5.4 Map Descriptor Examples (MIM Disabled)

The driver processes messages from the panel that relay the panel's current status as well as new history events. These messages contain composite data and the contents cannot simply be stored in a Data Array to read by a Client device.

Map Descriptors are used to store portions of this composite data from the following categories. At least one Map Descriptor is required for each category.

5.4.1 Zone Status

There are two methods of storing Zone status data:

- The driver can store the state (trouble, alarm, pre-alarm...) of each zone in a separate array as a a bit state (1 or 0). See sections 5.4.6 and 5.4.7. A separate Map Descriptor is required per zone state (9 possible).
- The driver can store a number to indicate normal or abnormal state of each zone. (The value of the number indicates the states.) See section 5.4.8. All data is stored in a single Data Array and one Map Descriptor is required for all zones.

5.4.2 Device Status

There are two methods of storing Device status data.

- The driver can store the state (trouble, alarm, pre-alarm...) of each device in a separate array as a bit state (1 or 0). See sections 5.4.9 and 5.4.10.. A separate Map Descriptor is required per zone state (9 possible). A separate set of Map Descriptors is required for each loop (4 possible).
- The driver can store a number in a separate Data Array to indicate normal or abnormal state of each device. See section 5.4.11. All data is stored in a single Data Array and one Map Descriptor is required for all zones.

5.4.3 Panel Information

Information about the panel itself such as evento counters, board status and LED status can be stored by the driver. See section 5.4.12. This data is stored in consecutive array locations. Appendix D.3 maps the layout of this data.

5.4.4 History Events

History events can be stored in two formats:

- Event codes for all devices are stored in a single data array at a location based on the source device's address. This gives an array of the most recent events for all devices. See section 5.4.13
- The entire history event record for the most recent event (any device or a specific device) can be stored in the Data Array DA_HIST_EVENT which must be defined as described in Section 4. Appendix D.3 maps the layout of this data.

5.4.5 Acknowledging Alarms

There are significant limitations on the driver’s ability to send alarm acknowledgements to the panel. Refer to Appendix A.1 for more information. Section 5.4.16 describes a Map Descriptor which can be used to get the driver to acknowledge alarms.

5.4.6 Map Descriptor Example 1. (All Zones Data)

This Map Descriptor may be used to store Zone data sent by the panel. The message sent by the panel is dependent on the panel's firmware version. This Map Descriptor will use 241 consecutive array locations to store data for the zones. Zone 0's data will be stored at the first location and Zone 240's state will be stored at the 241st location. The base location in the array is determined by the Data Array offset,

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name    , Data_Array_Offset   , Function   , Node_Name   , Cheet_Zone   , Cheet_Loop   , Cheet_Device   Cheet_DT       , Length
CHEETAH01             , ZONE_ABORTS        , 0                   , Passive    , Panel_01    , All          , None         , None           , Abort        , 256
CHEETAH02             , ZONE_TROUBLES      , 0                   , Passive    , Panel_01    , All          , None         , None           , Trouble      , 256
CHEETAH03             , ZONE_SUPERS        , 0                   , Passive    , Panel_01    , All          , None         , None           , Supervisory  , 256
CHEETAH04             , ZONE_DISABLES      , 0                   , Passive    , Panel_01    , All          , None         , None           , Disable      , 256
```

Each Map Descriptor can be given a unique name.

The data is stored in this Data Array.

Starting at element zero.

Data for All Zones is stored (241 data

A separate Map Descriptor is required for each Data Type.

5.4.7 Map Descriptor Example 2.(Specific Zone Data)

In this example the Map Descriptors store data for one zone each. This variation allows the manipulation of the arrangement of data in Data Arrays.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Cheet_Zone   , Cheet_DT   , Data_Array_Name   , Data_Array_Offset   , Function   , Node_Name   , Length
Zone_Alarms1          , 1            , Release    , DA_1              , 0                   , Passive    , Node_A      , 256
Zone_Alarms2          , 2            , Release    , DA_2              , 0                   , Passive    , Node_A      , 256
```

Data for the specified zone is stored.

Release state data is stored.

The data is stored into a Data Array called DA_1 for zone 1 and DA_2 for zone 2...

Starting at element zero.

5.4.8 Map Descriptor Example 3 – Zone Status as a numeric value

In this example, the driver stores zone data for any zone. It will store data for all possible states that the panel reports for each of the zones in the form of a number in the Data Array. The number can be interpreted to determine which states are active.

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Cheet_Zone  , Cheet_Loop  , Cheet_Device  Cheet_DT  , Length
CHEETAH10            , ZONE_STATES      , 0                  , Passive   , Panel_01   , All         , None        , None          , Any     , 256
```

For the driver to effectively report the status as a number the Data Array format must be suitable for storing the number. UINT16, UINT32 and FLOAT formats are supported.

This data type tells the driver that this Map Descriptor must be used to store zone status data of ***any*** type.
The driver writes a number into the array location for each device. The value of the number indicates the status of the zone. The value is based on which bits in the binary number are set.
Bit 0: Abort State
Bit 1: Trouble State
Bit 2: Supervisory State
Bit 3: Zone is disabled
Bit 4: Pre Alarm State
Bit 5: Alarm State
Bit 6: Pre-Discharge State
Bit 7: Release State
Bit 8: Process State
Example : Value = 32 indicates an alarm state
Example : Value = 96 indicates an alarm & pre-discharge state

5.4.9 Map Descriptor Example 4 – All Devices

In this example separate Map Descriptors are provided to store the Pre-Alarm, Alarm and Trouble States of all devices on loops 1 and 2.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name , Data_Array_Name   , Data_Array_Offset , Function , Node_Name , Cheet_Zone , Cheet_Loop , Cheet_Device Cheet_DT   , Length
CHEETAH11           , DEV_L1_PREALMS , 0                   , Passive  , Panel_01  , None       , 1          , All                 , Pre-Alarm , 256
CHEETAH12           , DEV_L1_ALARM   , 0                   , Passive  , Panel_01  , None       , 1          , All                 , Alarm     , 256
CHEETAH13           , DEV_L1_TRBLS   , 0                   , Passive  , Panel_01  , None       , 1          , All                 , Trouble   , 256

CHEETAH14           , DEV_L2_PREALMS , 0                   , Passive  , Panel_01  , None       , 2          , All                 , Pre-Alarm , 256
CHEETAH15           , DEV_L2_ALARM   , 0                   , Passive  , Panel_01  , None       , 2          , All                 , Alarm     , 256
CHEETAH16           , DEV_L2_TRBLS   , 0                   , Passive  , Panel_01  , None       , 2          , All                 , Trouble   , 256
```

A different array is used for each state. Another strategy could have been to use the same array but vary the offset.

When storing device data, the Cheet_Zone must be set to None.

Additional Map Descriptors are required for each loop.

The Data Type describes which Map Descriptor to use for each device state's storage. If the device state doesn't match one of these data types then the message will be ignored.

5.4.10 Map Descriptor Example 5 (Specific Device)

In this example, a Map Descriptor has been defined for the storage of the state of one specific device. (Device 20)

```
//  Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Cheet_Zone , Cheet_DT , Cheet_Device , Cheet_Loop , Length
Device20_L1_Alm     , DA_DI1          , 20                , Passive  , Node_A    , None       , Alarm    , 20           , 1          , 256
```

Each Map Descriptor in this example reads data for one device only. Thus each Map Descriptor must point to a different Data Array or as is the case in this example, to a different location in the same Data Array.

They are passive because this driver is a data consumer.

The driver is node independent; however Node_A ties this Map Descriptor to a Node Descriptor which thus connects the Map Descriptor to a protocol and to a port.

When storing device data the Cheet_Zone must be set to None.

The alarm state is being stored.

The Device number is 20.

The device belongs to loop1

5.4.11 Map Descriptor Example 6 - Device States as a Numeric Value

In this example the normal or abnormal state of all the devices of loops 1-4 will be stored by this Map Descriptor.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Cheet_Zone  , Cheet_Loop  , Cheet_Device  Cheet_DT  , Length
CHEETAH23            , DEV_L1_STATE     , 0                  , Passive   , Panel_01   , None        , 1           , All           , Any      , 256
CHEETAH24            , DEV_L2_STATE     , 0                  , Passive   , Panel_01   , None        , 2           , All           , Any      , 256
CHEETAH25            , DEV_L3_STATE     , 0                  , Passive   , Panel_01   , None        , 3           , All           , Any      , 256
CHEETAH26            , DEV_L4_STATE     , 0                  , Passive   , Panel_01   , None        , 4           , All           , Any      , 256
```

The array's format must be suitable for storing the state number which can range from 0-15.
Thus BYTE, UINT16, UINT32 and FLOAT are suitable formats for the Data Array.

The ***Any*** keyword tells the driver to store the device state as a number
The value of the number indicates the device state. The number is a binary number and its value is determined by which bits are set.
Bit 0: Alarm
Bit 1: Pre-Alarm
Bit 2: Trouble

5.4.12 Map Descriptor Example 7 – Panel Data

This example provides a Map Descriptor which tells the driver where to store the non-zone/device specific data obtained from a panel. Appendix D.3 of the manual maps how the data is stored. Ensure that the Data Array is long enough to store all the data.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Cheet_Zone  , Cheet_Loop  , Cheet_Device  Cheet_DT  , Length
CHEETAH27            , PANEL_DATA       , 0                  , Passive   , Panel_01   , None        , None        , None          , Panel    , 100
```

The ***Panel*** keyword is used to store the panel data using this Map Descriptor.

5.4.13 Map Descriptor Example 8 – History Data (All Devices)

In this example, 4 Map Descriptors process all history events on all four loops. One Data Array is used and loop #2's data is stored at an offset location of 240 (max number of devices per loop) in the Data Array. The Device is set to ALL to tell the driver to process all devices on the loop using this Map Descriptor. If a history event for device 100 on loop 3 is received then the driver will store the event code at location 480(=base offset for loop 3)+100 (=device address). The event code will be stored as a number and the meaning of the number may be obtained by reading Appendix D.3

```
//   Client Side Map Descriptors

Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  ,Cheet_Zone  , Cheet_DT  , Cheet_Device  , Cheet_Loop  , Length
Device1_L1_Hist      , DA_HIST          , 0                  , Passive   , Node_A     , None        , History   , All           , 1           , 256
Device1_L2_Hist      , DA_HIST          , 240                , Passive   , Node_A     , None        , History   , All           , 2           , 256
Device1_L3_Hist      , DA_HIST          , 480                , Passive   , Node_A     , None        , History   , All           , 3           , 256
Device1_L4_Hist      , DA_HIST          ,  720               , Passive   , Node_A     , None        , History   , All           , 4           , 256
```

History events relate to devices and thus the zone must be set to *None*.

Map Descriptor will store History data

One Map Descriptor is required per loop.

5.4.14 Map Descriptor Example 9 – Full History Event Record – Specific Device

Full History Event records contain composite data which require at least 65 consecutive Data Array locations for storage. If the Data_Array_Offset is not carefully specified the storage areas will overlap.

```
//   Client Side Map Descriptors

Map_Descriptor_Name , Data_Array_Name , Data_Array_Offset , Function , Node_Name , Cheet_Zone , Cheet_DT , Cheet_Device , Cheet_Loop , DA_Byte_Name     , Length
Device1_L1_Hist     , DA_HIST         , 20                , Passive  , Node_A    , None       , History  , 20           , 1          , DA_HIST_EVENT , 256
```

Event codes are stored in this Data Array.

One device is processed using this Map Descriptor. Thus only events relating to device 20 of loop 1 will be stored using this Map Descriptor. Events relating to other devices will be discarded unless additional Map Descriptors are defined.

Event history records are stored in this Data Array. The contents of the Data Array locations are described in Appendix D.3. Each event record uses at least 65 consecutive elements of the array so the choice of an offset must be made carefully.

5.4.15 Map Descriptor Example 10 – Full History Event Record – Any Device

Only the most recent history event is stored using this Map Descriptor. The full record is stored but is overwritten when a new event is received irrespective of the event's device address.

```
//  Client Side Map Descriptors

Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Cheet_Zone  , Cheet_DT  , Cheet_Device  , Cheet_Loop  , DA_Byte_Name     , Length
Device1_L1_Hist      , DA_HIST          , 0                  , Passive   , Node_A     , None        , History   , ALL           , 1           , DA_HIST_EVENT    , 256
```

Differs from Example 9 in that the device is specified as *ALL* - now the driver stores the event for any device at the same location.

5.4.16 Map Descriptor Example 11 – Alarm Ack

This example illustrates a Map Descriptor which can be used to acknowledge / silence or reset the panel. Read Appendix A.1 to understand the limitations of this functionality. The value of the array element at offset zero in the array named DA_ACK is used to send a signal to the Cheetah panel.

```
//   Client Side Map Descriptors

Map_Descriptor_Name    , Data_Array_Name    , Data_Array_Offset    , Function    , Node_Name    , Cheet_Func
Ack_Mapdesc            , DA_ACK             , 0                    , Passive     , Node_A       , Port Response
```

Only one element of this Data Array is used.
The value determines how the Cheetah panel will be affected.
If bit 0 is set then the panel will be reset.
If bit 1 is set then the panel will be silenced.
If bit 2 is set then the panel will be acknowledged.

The Map Descriptor must always be passive. This is because this Map Descriptor is used to respond to the port supervision query.

This tells the driver that this Map Descriptor is to be used for acknowledging alarms.

5.5 Map Descriptor Examples (MIM Enabled)

MIM Enabled mode allows the monitoring of more than one Fike panel. A Node with associated panel ID for each remote networked panel has to be created in the configuration file.

5.5.1 Map Descriptor Example 1 – Troubles (from Devices)

Trouble events are reported by devices on panel loops and are stored in data arrays. The storage offset equals the device's address on the loop. The last reported device trouble is stored in the Data Array specified under Data Array Name. The specific types of device troubles per device are stored in 16 Bit words in the Data Array specified under DA_Device_Event_List according to the following table:

Device Trouble	Store value	Event clearable?	Moved to Bit Data Array Offset
Device missing.	1	Yes	0
GCA Open trouble.	2	Yes	1
Low Power trouble.	4	Yes	2
Line Short trouble.	8	Yes	3
Line Open trouble.	16	Yes	4
Maintenance trouble.	32	Yes	5
Walktest alarm.	64	Yes	6
General trouble.	128	Yes	7
Device disabled trouble.	256	Yes	8
Pre-alarm #2.	512	Yes	9
Pre-alarm #1.	1024	Yes	10
Device disabled.	2048	Yes	11
Isolation trouble.	4096	Yes	12
Power supply low.	8192	Yes	13
Invalid Abort active.	16384	Yes	14

As per the example below, use the bit extract moves function to isolate each trouble type into a bit Data Array at offsets per the above table.

```
Moves
Function      , Source_Data_Array    , Source _Offset    , Target_Data_Array    , Target_Offset    , Length
Bit_Extract   , P11L01Dev_Trbls      , 0                 , Target_Bit_DA        , 0                , 15
```

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name     , Data_Array_Offset , Function , Node_Name , Loop_Number , History_Event_Name , Address , Length , DA_Device_Event_List
PNL11_TRB            , DEVICE_L2_TRBLS     , 0                 , passive  , Panel_11  , 1           ,  Device Trouble    , 0       , 255    , P11L01Dev_Trbls
PNL12_TRB            , PNL12_L2_TRBLS      , 0                 , passive  , Panel_12  , 1           ,  Device Trouble    , 0       , 255    , P12L01Dev_Trbls
```

5.5.2 Map Descriptor Example 2 – System Troubles

System trouble events are reported per panel. One mapdesc per networked panel is required. There are 24 possible system trouble types. The specific type of system trouble is stored in the Data Array at an offset according to the following table:

System Trouble	Data Array Offset	Store values	Event clearable?
Panel in configuration mode.	0	1	Requires panel reset to clear
Network no response. Network communication trouble.	1	0, 1	Yes
Network Class-A trouble.	2	0, 1	Yes
Network wire trouble.	3	0, 1	Yes
Panel missing.	4	0, Missing Panel ID	Yes
Peripheral missing.	5	0, Peripheral address of the missing peripheral	Yes
Loop Line short trouble.	6	Loop number	Requires panel reset to clear
Loop Class A trouble.	7	Loop number	Requires panel reset to clear
NAC circuit Short trouble.	8	0, NAC circuit number	Yes
NAC circuit Open trouble.	9	0, NAC circuit number	Yes
NAC or Panel Relay or Aux NAC circuit disabled.	10	0, 1	Yes
Panel AC Power Low trouble.	11	0, 1	Yes
Panel Battery trouble.	12	0, 1	Yes
Ground fault low trouble.	13	0, 1	Yes
Ground fault high trouble.	14	0, 1	Yes
Supplemental Power supply Fan trouble.	15	0, 1	Yes
Auxiliary power supply output trouble.	16	0, 1	Yes
Panel memory/communication trouble.	17	1	Requires panel reset to clear
Panel memory/firmware trouble.	18	1	Requires panel reset to clear

DACT general trouble.	19	0, 1	Yes
Printer trouble.	20	0, 1	Yes
Zone Disabled.	21	0, zone number	Yes
Panel in walktest mode.	22	1	Requires panel reset to clear
Enter walktest mode.	23	1	Requires panel reset to clear

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name      , Data_Array_Offset  , Function   , Node_Name   , History_Event_Name
PNL11_SYS_TRB        , PNL11_SYS_TRBLS,  , 0                     , passive    , Panel_11    , System_Trouble
PNL12_SYS_TRB        , PNL12_SYS_TRBLS,  , 0                     , passive    , Panel_12    , System_Trouble
```

5.5.3 Map Descriptor Example 3 – Device Alarms

Alarm events are reported by devices on panel loops and are stored in data arrays. The storage offset equals the device's address on the loop. The last reported device alarm is stored in the Data Array specified under Data Array Name. The specific types of device alarm per device are stored in 16 Bit words in the Data Array specified under DA_Device_Event_List according to the following table:

Device Alarm	Store value	Event clearable?	Moved to Bit Data Array Offset
Pre-discharge input.	1	Requires panel reset to clear	0
Manual release	2	Requires panel reset to clear	1
Watermist Alarm	4	Requires panel reset to clear	2
Detector in Alarm	8	Requires panel reset to clear	3
Waterflow Alarm	16	Requires panel reset to clear	4
Manual pull Alarm	32	Requires panel reset to clear	5
Alarm type #2	64	Requires panel reset to clear	6
Alarm type #1	128	Requires panel reset to clear	7

As per the example below, use the bit extract moves function to isolate each alarm type into a bit Data Array at offsets per the above table.

```
Moves
Function     , Source_Data_Array   , Source _Offset    , Target_Data_Array   , Target_Offset     , Length
Bit_Extract  , P11L01_Alarms       , 0                 , Target_Bit_DA       , 0                 , 8
```

NOTE: For a panel to report more than one alarm state per device is an unlikely event, but the protocol specification allows for such a condition to occur.

```
//   Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name  , Data_Array_Offset  , Function  , Node_Name  , Loop_Number  , History_Event_Name  , Address  , Length  , DA_Device_Event_List
PNL11_DEV_ALM        , PNL11_DV_ALMS    , 0                  , passive   , Panel_11   , 1            , Device Alarm        , 0        , 255     , P11L01_Alarms
```

5.5.4 Map Descriptor Example 4 – Vesda Events

Some Fike panels may support reporting Vesda panel events. The History_Event_Name indicates the type of event to capture.

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name  , Data_Array_Name    , Data_Array_Offset  , Function  , Node_Name  , Vesda_Zone  , History_Event_Name
PNL12_VESDA_TRBL     , PNL12_VSD_TRBL     , 0                  , passive   , Panel_12   , 1           , Vesda Trouble
PNL12_VESDA_PALM     , PNL12_VSD_PALM     , 0                  , passive   , Panel_12   , 2           , Vesda Pre-Alarm
PNL12_VESDA_ALRM     , PNL12_VSD_ALRM     , 0                  , passive   , Panel_12   , 3           , Vesda Alarm
```

Vesda alarm events are reported by panel zones and are stored in the data array. The type of alarm is stored at an offset according to the following table:

Vesda Alarm	Data Array Offset	Store values	Event clearable?
VESDA Fire-1 level.	0	0, 1	Yes
VESDA Fire-2 level.	1	0, 1	Yes
VESDA Sector Fire-1 level.	2	0, 1	Yes
VESDA Sector Fire-2 level.	3	0, 1	Yes

Vesda pre-alarm events are reported by panel zones and are stored in the data array. The type of pre-alarm is stored at an offset according to the following table:

Vesda Pre-alarm	Data Array Offset	Store values	Event clearable?
VESDA Alert level (Pre-alarm 1).	0	0, 1	Yes
VESDA Action level (Pre-alarm 2).	1	0, 1	Yes
VESDA Sector Alert level (Pre-alarm 1).	2	0, 1	Yes
VESDA Sector Action level (Pre-alarm 2).	3	0, 1	Yes

Vesda troubles are reported by panel zones and are stored in the data array. The type of trouble is stored at an offset according to the following table:

Vesda Trouble	Data Array Offset	Store values	Event clearable?
VESDA missing.	0	0, 1	Yes
VESDA Major fault.	1	0, 1	Yes
VESDA Minor fault.	2	0, 1	Yes
VESDA Isolate fault.	3	0, 1	Yes
VESDA Power fault.	4	0, 1	Yes
VESDA Network fault.	5	0, 1	Yes
VESDA Airflow fault.	6	0, 1	Yes
VESDA Filter fault.	7	0, 1	Yes
VESDA Configuration error trouble.	8	0, 1	Yes

5.5.5 Map Descriptor Example 5 – Device General, Supervisory and Process Events

Device General, Supervisory and Process events are reported by devices on panel loops and are stored in data arrays. The storage offset equals the device's address on the loop. The last reported device event is stored in the Data Array specified under Data Array Name. The specific types of device event per device are stored in 16 Bit words in the Data Array specified under DA_Device_Event_List according to the following table:

Device General event	Store value	Event clearable?	Moved to Bit Data Array Offset
Input Event Active	1	Yes	0
Abort Input	2	Requires panel reset to clear	1

Device Supervisory event	Store value	Event clearable?	Moved to Bit Data Array Offset
Supervisory input	1	Yes	0
Device disabled	2	Yes	1

Device Process event	Store value	Event clearable?	Moved to Bit Data Array Offset
Process input	1	Yes	0

As per the example below, use the bit extract moves function to isolate each event type into a bit Data Array at offsets per the above table.

```
Moves
Function      , Source_Data_Array   , Source _Offset   , Target_Data_Array   , Target_Offset   , Length
Bit_Extract   , P11L01Dev_Gen       , 0                , Target_Bit_DA1      , 0               , 2
Bit_Extract   , P11L01Dev_Sup       , 0                , Target_Bit_DA2      , 0               , 2
```

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name    , Data_Array_Offset   , Function   , Node_Name   , Loop_Number   , History_Event_Name    , Address   , Length   , DA_Device_Event_List
PNL11_GEN_EVTS        , PNL11_GEN_EVTS     , 0                   , passive    , Panel_11    , 1             , Device General        , 0         , 255      , P11L01Dev_Gen
PNL11_SUP_EVTS        , PNL11_SUP_EVTS     , 0                   , passive    , Panel_11    , 1             , Device Supervisory    , 0         , 255      , P11L01Dev_Sup
PNL11_PRC_EVTS        , PNL11_PRC_EVTS     , 0                   , passive    , Panel_11    , 1             , Device Process        , 0         , 255      , P11L01Dev_Proc
```

5.5.6 Map Descriptor Example 6 – Classic Cheetah Events from Gateway

Some Fike panels may support reporting Classic Cheetah panel events. The specific type of event is stored in the Data Array at an offset according to the following table:

Gateway Event	Data Array Offset	Store values	Event clearable?
Gateway Alarm active.	0	1	Requires panel reset to clear
Gateway Supervisory active.	1	0, 1	Yes
Gateway Trouble active.	2	0, 1	Yes
Gateway Pre-alarm 1 active.	3	0, 1	Yes
Gateway Pre-alarm 2 active.	4	0, 1	Yes
Gateway Process active.	5	0, 1	Yes
Gateway Zone disabled.	6	0, 1	Yes

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name   , Data_Array_Name    , Data_Array_Offset   , Function   , Node_Name   , Gateway_Zone   , History_Event_Name
CLASSIC_GW_EVT        , CLASSIC_GW_EVT     , 0                   , passive    , Panel_GW    , 1              , Gateway
```

6 CONFIGURING THE FIELDSERVER AS A CHEETAH DEVICE SERVER

This Driver cannot act as a Server, i.e. it cannot write data to the Cheetah controller or devices. Thus it cannot be used to acknowledge alarms or reset states.

Appendix A. USEFUL FEATURES

Appendix A.1. Acknowledging Alarms

This driver can be used to acknowledge alarms, reset or silence the Cheetah panel.

This functionality is limited. For this function to operate the Cheetah panel must be configured to supervise the port that the FieldServer is connected to. In addition to enabling this function, port supervision means that the panel will go into alarm if the FieldServer does not respond to the supervision messages. In fact the request to ack /reset/ silence the panel is included in the driver's response to the supervision poll from the panel. The panel ignores unsolicited messages. A consequence of this is that the driver cannot control the timing of when the ack /reset/ silence message is sent to the panel.

In using this functionality you should also understand that the Cheetah panel protocol does not acknowledge message receipt so this driver cannot report whether the message was received by the panel and whether it was acted on.

The port supervision response message is sent even if you do not define a ***Port Response*** Map Descriptor in the CSV file. In this case the command data will always be zero. Once you define a Map Descriptor then the driver uses one Data Array element to determine the command data to send to the panel. The value from this array element determines the action the panel will take.

Bit	Action	Decimal Value
0	Reset	1
1	Silence	2
2	Acknowledge	4

Example: To Silence the Panel, then set the value of the Data Array element to 2.

Appendix B. VENDOR INFORMATION

Appendix B.1. Fike Cheetah Panel Firmware version sensitivity

The following limitations to older firmware have been identified. FieldServer is unable to correlate this change with a particular Fike firmware version number at the current time.

Messages 1.1 and 6.0 (graphics update msg) are sent by older panels. Only message 6.0 contains information that can be used by the driver to reflect zone and device states. The 6.0 message is sent infrequently (typically 1 in 60 messages) and this results in a zone/device state update every few minutes.

Appendix B.1.1. Message 6.0 limitations

- Can only report data for 127 zones. Data is limited to Alarm, Trouble, Pre-Discharge and Released states
- Can only report data for 127 devices on loops 1-4. Data is limited to Alarm, Trouble states.

Appendix B.1.2. Message 1.1 - Older firmware

- No useful information about zones or devices.

Appendix B.1.3. Message 1.1 - Newer Firmware

- Supports zones 1-240. For zones the following states are reported: Abort, Trouble, Supervisory, Disabled, Pre-alarm, Alarm, Pre-Discharge, Released, Processed.
- Supports devices 1-127 on loops 1-4. For devices the following states are reported: Alarm, Pre-Alarm and Trouble

Appendix B.2. Fike XI Panel Limitations and capabilities

The XI panels send messages which contain zone and device data. The message is known by Fike as a '1.2' message.

The message reports if a device is normal, alarm, trouble or in a pre-discharge state. 256 Devices on loops 1-4 are reported. 255 Zones report Process, Trouble, Supervisory, Disable, Pre-Alarm, Alarm states.

Appendix C. TROUBLESHOOTING

Appendix C.1. Driver Error Messages

Multiple protocol drivers may exist on a FieldServer. Each driver may produce its own error messages and the FieldServer itself may produce error messages.

All messages produced by this driver begin "Cheetah:"

Error Message	Action Required
Cheetah:#1 Unknown simulation function.	Report these messages to FieldServer Technologies.
Cheetah:#2 Simulation function unknown.	
Cheetah:#3 Protocol Error (Start), Incoming msg ignored - Waiting for next msg.	Warning messages only. An incoming message was discarded because the identifiers which mark the beginning of a message could not be found. You cannot take any action to correct this message. If it occurs often check wiring, noise and installation.
Cheetah:#4 Protocol Error (Stop), Incoming msg ignored - Waiting for next msg.	
Cheetah:#5 Protocol Error (Chksum), Incoming msg ignored - Waiting for next msg.	
Cheetah:#6 Protocol Error (Unknown), Incoming msg ignored - Waiting for next msg.	Report this error to FieldServer Technologies.
Cheetah:#7a Err. Zone=%d Max=%d Cheetah:#7b MapDesc Error. Zone value error. (%d) Cheetah:#7c FYI. Warning. Zone=%d. Max zone is panel type dependent. Read Manual.	An invalid zone has been specified. The zone causing the problem is printed in parenthesis.[5] The largest possible zone number is 255. Only Cybercat panels support this number of zones. Legacy panels support up to zone 239. Very old panels that can only send message #6 can only report information for 127 panels. If you are connected to a Cybercat panel and you get the FYI message you can ignore it. If you are connected to some other panel and you get either the Err or FYI message then you will need to correct the CSV file.
Cheetah:#8 MapDesc Error. Device value error. (%d)	An invalid device has been specified. The device causing the problem is printed in parenthesis.[5]
Cheetah:#9 MapDesc Error. Loop value error. (%d)	An invalid loop has been specified. The loop causing the problem is printed in parenthesis. [55]
Cheetah:#10 MapDesc Error. Zone & Device Specified.	One Map Descriptor cannot be used to store data for zones and devices. Either the keyword Cheet_Zone or Cheet_Device must be set to *None*[5].

[5] Modify the CSV file, download to the FieldServer and restart the FieldServer for the changes to take effect.

Error Message	Action Required
Cheetah:#11 MapDesc Error. With devices only alarm & trouble available.	For devices only the alarm & trouble states are available. Set the Cheet_DT values appropriately in the configuration file.[5]
Cheetah:#12 Message on Cheetah port but no mapDesc found.	A port has been reserved for the Cheetah protocol and a message has been received on this port but there is no Map Descriptor defined for this port.[6]
Cheetah:#13 Data Array to short. MapDesc=<%s> RQD=%d.	The Data Array associated with the Map Descriptor in question is too short. Adjust the length as required by the error message. Note that the error may be repeated for a single Map Descriptor when a new zone or device is stored because the storage location may be based on the zone or device number. Generally for zone storage the array must have at least 240 locations (and 128 for older Cheetah firmware.)[6]
Cheetah:#15 Err. MD length is required - defaulting to 1	The Map Descriptor length must be sufficient to store all the data. The maximum device number is 255 and the maximum zone number is 255, therefore to store all zones and devices the MD's must be 256 elements long. Some legacy panels and message don't support the full number of devices/zones. For example some panels only support 241 zones. Try and determine the correct length otherwise please use 256. If you are unsure ask tech support to provide the template file server.csv
Cheetah:#17. Err. DIAG_USER_1	An internal diagnostic has been activated. This should not happen on a live system. Take a log and contact tech support
Cheetah:# 18 Err. Bad msg start= %#x	Messages are expected to begin with a Carriage return or SOH (0x01). The message has been rejected because it starts with the reported byte. Perhaps the vendor has changed firmware. If this error occurs repeatedly then take a log and contact tech support. If it occurs rarely then assume it is noise and ignore it if you are satisfied you are getting good data updates.
Cheetah:#19 FYI. Ignoring 0x0100 messages from Cybercat.	Cybercat panels transmit legacy message which must be ignored. This message confirms the driver is ignoring them. No corrective action is required. If you are connected to a Cybercat panel and never see this message printed (checked the system and driver error message screen) then please take a log and contact Tech support.
Cheetah:#20 Err. DA too short. Zone=%d MD=<%s> RQD=%d	The driver is attempting to store zone status information from a Cybercat panel. The Data Array is too short. Adjust the length of the Data Array and the length of the Map Descriptor.[6]
Cheetah:#21. Err. DA too short. MapDesc=<%s> RQD=%d DA=%s	The driver is attempting to store device status information from a Cybercat panel. The Data Array is too short. Adjust the length of the Data Array and the length of the Map Descriptor.[6]
Cheetah:#22 FYI. Node=%s is a Cybercat panel.	These messages report the type of panel that the configuration is suitable for. If this isn't what you expect, review the configuration against the manual and make changes as required. Refer to section 0 for more information.[6]
Cheetah:#23 FYI. Node=%s is a Cheetah panel.	
Cheetah:#24 Error, Node_ID must be specified when MIM enabled!	Specify Node ID of remote panel or remove MIM enabled setting. Note: for storing data from multiple panels MIM enabled should be specified.

[6] Modify the CSV file, download to the FieldServer and restart the FieldServer for the changes to take effect.

Error Message	Action Required
Cheetah:#25 Invalid MIM_Enabled setting [Yes;No], defaulting to No!	Use either Yes or No for the MIM enabled setting.

Appendix C.2. Driver Stats

Cheetah panels produce data messages for slave devices to consume. The type and frequency of the messages depends on the Cheetah firmware revision.

The driver counts all incoming messages of interest as the PLC_READ_MSG_RECD statistic. Other legal messages which do not contain the data this driver is interested in are discarded and are counted as the MSG_IGNORED statistic.

The PLC_READ_MSG_RECD statistic is incremented once by each Map Descriptor which extracts data from an incoming message. Thus, one incoming message and three associated Map Descriptors would cause the statistic to increase by three (when viewed from the connection's point of view.)

The driver ignores messages 0x0100 from Cybercat panels. These legacy messages contain contradictory information. If the node is configured as a Cybercat panel then the driver ignores the messages and increases the Ignored Messages stat on the connection.

Appendix C.3. Map Descriptor Specific Errors

Some errors produced by the driver are Map Descriptor specific. They can only be seen when using the Ruidebug program and looking at the Map Descriptor debugging screens. For more information on how to do this please refer to the FieldServer Utilities manual.

Appendix C.4. Multiple Cheetah Panels

Some of the broadcast messages produced by the Cheetah panel are node-less. This means that these messages do not identify the node of origin. Unfortunately, the message this driver uses to determine zone and device alarms is a node-less message. This limits the number of Cheetah panels per port to one. (If there were more than one the driver would not be able to determine the node of origin.)

Appendix D. REFERENCE

Appendix D.1. Driver Scope

This driver has implemented cheetah command #6.0. These messages are produced by a Cheetah device controller and are consumed by this driver. The driver has no control on the frequency of the messages and thus cannot guarantee fresh data. In addition the protocol has no method for acknowledging messages so that in the event of this driver having to discard a corrupt message, the message producer does not know and will not re-send.

Appendix D.2. Panel Firmware Versions

The driver supports older versions of panel firmware which transmit a shorter version of the Panel Status command. This shorter version contains only panel data whereas the newer version contains panel, zone & device data.

Appendix D.3. Storing Panel Data

Panel data is stored in consecutive locations as described in the map below. For additional explanations on the meaning of each data element contact FIKE directly.

Array Location	Data Category	Data stored at this location
0	Active Counter (0..65535)	Active events in normal state
1		Active events in Process state
2		Active events in Trouble state
3		Active events in Supervisory state
4		Active events in Drill state
5		Active events in ? state
6		Active events in ? state
7		Active events in ? state
8		Active events in Abort state
9		Active events in Pre-Alarm1 state
10		Active events in Pre-Alarm2 state
11		Active events in Alarm state
12		Active events in Pre-Discharge state
13		Active events in Release state
14		Active events in Zone Discharge state
15		Active events in Water Mist state
16	Board Status Indicators (0 or 1)	Alarm silence active 1=yes
17		Supervisory Silence Active 1=yes
18		Trouble Silence Active 1=yes
19		AC Power on Main Board 1=ok
20		AC Power on Aux Board 1=ok
21		Walk test Condition 1=on
22		Ground Fault Condition 1=ok
23		Ack Condition 1=on
24	Date & Time	Month
25		Day of Month
26		Year
27		Day of Week

Array Location	Data Category	Data stored at this location
28		Hour
29		AM/PM
30		Minute
31		Second
32	Current Event Counter	Number of current events present on cheetah
33	LED Status Indicators	Pre-Alarm
34		Fire Alarm
35		Supervisory Alarm
36		Trouble
37		AC Power
38		Alarm Silence
39		Supervisory Silence
40		Trouble Silence

Appendix D.4. How History Events are Stored

History events records are send by the panel each time an event occurs. The driver can parse these messages and store the data in either of two ways.

- The event code is stored at an array location that is based on the device address and loop number. See Section 5.4.13
- Additional event data is stored in a secondary Data Array. See Section 5.4.14

The array locations of the additional data are indicated below

Location	Contents	Notes
0	Month	
1	Day of Month	
2	Year less 1900	
3	Day of Week (1..7)	
4	Hour (0-12)	
5	AM-PM (A,P)	
6	Minute	
7	Second	
8	Event Data (0-65535)	
9	Custom Message	20 bytes
29	Fire Point	
30	Zero Point	
31	Device Type	
32	Device Function	
33	Zones device belongs to	32 bytes. Bit 0 -> Zone1 …. Bit 17->zone 18 …
65	Current History Counter (0-65535)	

The event codes are defined in the table below.

#	Description	Notes
0	no record	-
1	aux power unconfig	auxiliary power module present but unconfigured.
2	aux power corrected	auxiliary power module replaced after being missing.
3	aux power missing	auxiliary power module not present but configured.
4	ac1 power restored	ac for main board input restored.
5	ac1 power trouble	ac for main board input missing/brownout.
6	ac2 power restored	ac for power module input restored.
7	ac2 power trouble	ac for power module input missing/brownout.
8	batttery1 restored	battery for main board input restored.
9	batttery1 trouble	battery for main board input missing.
10	battery2 restored	battery for power module input restored.
11	battery2 trouble	battery for power module input missing.
12	aud1 trb restored	audible #1 restored.
13	aud1 open trouble	audible #1 open circuit.
14	aud1 short trouble	audible #1 short circuit.
15	aud2 trb restored	audible #2 restored.
16	aud2 open trouble	audible #2 open circuit.
17	aud2 short trouble	audible #2 short circuit.
18	gnd fault restored	ground fault trouble restored to normal.
19	gnd fault present	ground fault present.
20	device present	addr device trouble: device returns canceling missing trouble
21	device missing	addr device trouble: device was polled but not found
22	commun restored	addr device trouble: communication to device has returned to normal
23	bad communication	addr device trouble: communication to device has errors (parity checksum etc.)
24	restored multiple	addr device trouble: multiple devices trouble has cleared
25	multiple devices	addr device trouble: multiple devices detected at this address
26	restored device kind	addr device trouble: correct kind of device has been restored to this address
27	wrong device kind	addr device trouble: wrong kind of devices has been found
28	contaminated device	addr device trouble: analog device is contaminated
29	loop module restored	loop mod missing trouble has cleared
30	loop module missing	loop module not in place but there are configured devices on the loops.
31	config menu accessed	trouble stored when the configuration menu has been entered
32	factory initialized	history stored when a factory initialization has been invoked
33	uart failure	history stored when the uart has been corrupted by esd and then re-initialized.
34	uart restored	history stored when preceding trouble has cleared.
35	system reset	complete reset including power-down of loops for ten seconds.
36	trouble restored	trouble function input restore.
37	trouble input active	trouble function input present.
38	process restored	process control function input restored.

#	Description	Notes
39	process input active	process control function input present.
40	supervisory restored	supervisory function input restore.
41	supervisory active	supervisory function input present.
42	remote reset release	remote reset function input released.
43	trouble unsilence	all troubles unsilenced for all zones.
44	trouble silence	all troubles silenced for all zones.
45	supervisory unsilence	all supervisory unsilenced for all zones.
46	supervisory silence	all supervisory silenced for all zones.
47	power-up reset	reset record when panel is powered up
48	external reset	reset record when external reset switch pressed or esd hit causes reset
49	remote reset active	remote reset switch activated
50	drill input restored	drill function input restore.
51	drill input active	drill function input present.
52	alarm unsilence	all alarm states unsilenced for all zones.
53	alarm silence	all alarm states silenced for all zones.
54	abort pressed	abort function input present.
55	abort released	abort function input restored.
56	palarm1 restore-----	device restored below pre-alarm1 threshold. analog data appended to this message
57	palarm1 active -----	device above pre-alarm1 threshold. analog data appended to this message
58	palarm2 restore-----	device restored below pre-alarm2 threshold. analog data appended to this message
59	palarm2 active -----	device above pre-alarm2 threshold. analog data appended to this message
60	alarm active -----	device has reached the alarm threshold. analog data appended to this message
61	manual alarm	manual pull function input present.
62	manual pre-discharge	pre-discharge function input present.
63	manual release	manual release function input present.
64	waterflow active	waterflow input function activated.
65	waterflow restored	waterflow input function restored to normal.
66	invalid abort active	abort switch pressed in zone without an alarm present
67	inv abort restored	abort switch released in zone without an alarm present
68	cfg resto 9: mr-----	config error #9 restored. zone number appended to this message.
69	cfg error 9: mr-----	config error #9: manual release is in an alarm only zone.
70	network reset	history stored when a reset command is recorded from cheetah net
71	network acknowledge	history stored when an acknowledge command is recorded from cheetah net
72	process restored	process state for this zone de-activated.
73	process present	process state for this zone activated.
74	trouble restored	trouble state for this zone de-activated.
75	trouble present	trouble state for this zone activated.
76	supervisory restored	supervisory state for this zone de-activated.
77	supervisory present	supervisory state for this zone activated.

#	Description	Notes
78	drill restored	drill state for this zone de-activated.
79	drill present	drill state for this zone activated.
80	net member#a missing	network member #a is missing.
81	net member#a return	network member #a has returned on line.
82	net manager missing	network manager is missing.
83	net manager return	network manager has returned on line.
84	network silence	history stored when a silence command is recorded from cheetah net
85	network drill presnt	drill state activated from network - all zone only.
86	abort restored	abort state for this zone de-activated.
87	abort present	abort state for this zone activated.
88	pre-alarm1 restored	pre-alarm 1 state for this zone de-activated.
89	pre-alarm1 present	pre-alarm 1 state for this zone activated.
90	pre-alarm2 restored	pre-alarm 2 state for this zone de-activated.
91	pre-alarm2 present	pre-alarm 2 state for this zone activated.
92	network drill restor	drill state restored from network - all zone only.
93	alarm present	alarm state for this zone activated.
94	periph #aa missing	peripheral device of id#aa is missing.
95	pre-discharge	pre-discharge state for this zone activated.
96	periph #aa return	peripheral device of id#aa has returned on line.
97	release	release state for this zone activated.
98	zone enabled	zone disable state for this zone de-activated.
99	zone disabled	zone disable state for this zone activated.
100	watermist inactive	not used since release state never clears
101	watermist active	watermist state for this zone activated
102	network alarm unsil	network command - unsilence alarms
103	network superv unsil	network command - unsilence supervisories
104	checksum ok -----	zone configuration checksum error restored. zone number appended to this message.
105	checksum error -----	zone configuration checksum error. zone number appended to this message.
106	network troubl unsil	network command - unsilence troubles
107	vesda not configured	trouble event stored when the hli is present but no vesda is configured.
108	vesda configur clear	trouble clear event for the above trouble.
109	config menu exited	trouble clear event when the configuration menu has been exited
110	aux1 power restored	auxiliary power input trouble restored - main board
111	aux1 power trouble	auxiliary power input trouble - main board
112	aux2 power restored	auxiliary power input trouble restored - power module
113	aux2 power trouble	auxiliary power input trouble - power module
114	aud switch disabled	enable/disable switch for audibles is in the disable position
115	aud switch restored	enable/disable switch for audibles has returned to the enable position
116	diagnostics entered	indicates the operator has accessed the diagnostic menu selections
117	zone disable restore	frcm assigned to zone disable function has cleared
118	zone disable active	frcm assigned to zone disable function has activated

#	Description	Notes
119	internal fault	device trouble from the 0x08 polling command
120	int fault restored	trouble restored from the 0x08 polling command
121	wiring fault-open	open circuit fault on supervised circuit: frcm/som/srm
122	wiring fault-short	short circuit fault on frcm/som/srm
123	acknowledge switch	acknowledge switch pressed.
124	test point bad	analog device fire test point is out of hochiki specified range.
125	calibration restored	calibration fault has been restored. device now has valid calibration data.
126	calibration fault	fault detected during calibration of this device.
127	configure restored	fault restored from som/r2m/srm configuration trouble.
128	configure fault	fault detected when configuring som/r2m/srm.
129	reconfigured device	addressable device has been configured
130	enabled device	addressable device has been enabled.
131	disabled device	addressable device has been disabled.
132	enabled loop 1	loop 1 has been enabled.
133	disabled loop 1	loop 1 has been disabled.
134	enabled loop 2	loop 2 has been enabled.
135	disabled loop 2	loop 2 has been disabled.
136	enabled loop 3	loop 3 has been enabled.
137	disabled loop 3	loop 3 has been disabled.
138	enabled loop 4	loop 4 has been enabled.
139	disabled loop 4	loop 4 has been disabled.
140	enabled zone -----	zone enabled. zone number appended to this message.
141	disabled zone -----	zone disabled. zone number appended to this message.
142	enabled ckt -----	board level output circuit enabled. circuit number appended to this message.
143	disabled ckt -----	board level output circuit disabled. circuit number appended to this message.
144	loop 1 calibration	all configured analog devices on loop 1 have been calibrated.
145	loop 2 calibration	all configured analog devices on loop 2 have been calibrated.
146	loop 3 calibration	all configured analog devices on loop 3 have been calibrated.
147	loop 4 calibration	all configured analog devices on loop 4 have been calibrated.
148	l1 open class a	class a wiring has an open circuit trouble - loop 1.
149	l2 open class a	class a wiring has an open circuit trouble - loop 2.
150	l3 open class a	class a wiring has an open circuit trouble - loop 3.
151	l4 open class a	class a wiring has an open circuit trouble - loop 4.
152	l1 class a restored	class a wiring trouble has restored to noramal - loop 1.
153	l2 class a restored	class a wiring trouble has restored to noramal - loop 2.
154	l3 class a restored	class a wiring trouble has restored to noramal - loop 3.
155	l4 class a restored	class a wiring trouble has restored to noramal - loop 4.
156	walktest active	panel has enterred the walktest mode.
157	walktest expired	walktest mode has expired.
158	cfg resto 1: mr-----	config error #1 restored. zone number appended to this message.
159	cfg error 1: mr-----	config error #1 (no manrel in this zone). zone number appended to this message.
160		cheetah network device clear
161		cheetah network device activation

#	Description	Notes
162	config resto 3: al v	config error #3 restored.
163	config error 3: al v	config error #3 (alarm verification on device in a suppression zone) trouble.
164	system power low	system power below voltage level for reliable operation.
165	system power ok	system power low trouble restored.
166	checksum error -----	config checksum error trouble: devices & loops (loop number appended).
167	checksum ok -----	config checksum error trouble restored: devices & loops (loop number appended).
168	checksum ok -----	board level output config checksum restored. circuit number appended to this message.
169	checksum error -----	board level output config checksum error. circuit number appended to this message.
170	password level 1	password level 1 enterred.
171	password level 2	password level 2 enterred.
172	password level 3	password level 3 enterred.
173	password level 4	password level 4 enterred.
174	password level 5	password level 5 enterred.
175	password expired	password has timed-out.
176	software error	erroneous interrupt source or illegal program instruction fetched.
177	keyword error	erroneous program flow.
178	wiring fault restore	frcm/som/srm wiring fault restored to normal.
179	config resto 4: sens	config error #4 restored
180	config error 4: sens	config error #4 (analog phot has too high alarm sensitivity)
181	bat1 charge volt low	battery #1 charger has a low voltage output
182	bat1 charge restored	battery #1 chrager has returned to normal voltage output
183	bat2 charge volt low	battery #2 charger has a low voltage output
184	bat2 charge restored	battery #2 chrager has returned to normal voltage output
185	cfg resto 5: w------	config error #5 restored. zone number appended.
186	cfg error 5: w------	config error #5 (watermist zone needs an srm). zone number appended.
187	cfg resto 6: w------	config error #6 restored. zone number appended.
188	cfg error 6: w------	config error #6 (watermist zone has an abort). zone number appended.
189	cfg resto 7: on-time	config error #7 restored
190	cfg error 7: on-time	config error #7 (srm assigned to watermist & zone has no on-time)
191	cfg resto 8: zn-type	config error #8 restored.
192	cfg error 8: zn-type	config error #8 (srm assigned to watermist & alarm type of zone).
193	pc configuring panel	configuration is currently being changed by the pc
194	pc configuring done	the pc has finished configuring the panel
195	history is erased	the history buffers have been erased
196	cfg resto 10: vesda#	config error #10 restored
197	vesda commun error	communication to the vesda hli device has failed
198	vesda commun restore	communication to the vesda hli device has restored to normal
199	cfg error 10: vesda#	config error #10 (two vesda's with the same unit number)
200	alert level restore	vesda detector alert level restored. analog data appended to this message

#	Description	Notes
201	alert 0.000%/0.000%	vesda detector alert level activated. analog data appended to this message
202	action level restore	vesda detector action level restored. analog data appended to this message
203	action 0.000%/0.000%	vesda detector action level activated. analog data appended to this message
204	fire-1 0.000%/0.000%	vesda detector fire 1 level activated. analog data appended to this message
205	fire-2 0.000%/0.000%	vesda detector fire 2 level activated. analog data appended to this message
206	vesda fault: major	vesda detector trouble
207	vesda clear: major	vesda detector trouble
208	vesda fault: minor	vesda detector trouble
209	vesda clear: minor	vesda detector trouble
210	vesda fault: isolate	vesda detector trouble
211	vesda clear: isolate	vesda detector trouble
212	vesda fault: system	vesda detector trouble
213	vesda clear: system	vesda detector trouble
214	vesda fault: zone	vesda detector trouble
215	vesda clear: zone	vesda detector trouble
216	vesda fault: urgent	vesda detector trouble
217	vesda clear: urgent	vesda detector trouble
218	vesda fault: power	vesda detector trouble
219	vesda clear: power	vesda detector trouble
220	vesda fault: network	vesda detector trouble
221	vesda clear: network	vesda detector trouble
222	vesda fault: airflow	vesda detector trouble
223	vesda clear: airflow	vesda detector trouble
224	vesda fault: filter	vesda detector trouble
225	vesda clear: filter	vesda detector trouble
226	vesda fault id = ---	
227	vesda clear id = ---	
228	invalid vesda zone	vesda zone configured but not present on the vesdanet.
229	invalid v-zone clear	clear message for above fault.
230	invalid vesda fault	fault code 201 message purpose unclear at this time.
231	invalid fault clear	clear message for above fault.
232	grnd fault disabled	ground fault detection has been disabled.
233	grnd fault enabled	clear message for above fault.
234	network clr: -------	network event message: zone-state has cleared
235	network act: -------	network event message: zone-state has activated
236	periph #aa reset	peripheral device of id#aa has active reset sw.
237	periph #aa silence	peripheral device of id#aa has active silence sw.
238	periph #aa acknowl	peripheral device of id#aa has active acknowledge sw.
239	periph #aa trouble	peripheral device of id#aa has an active trouble.
240	periph #aa trb clear	peripheral device of id#aa trouble has cleared.
241	periph #aa zone err	peripheral device of id#aa has a zone mis-match with cheetah configuration
242	periph #aa zone clr	peripheral device of id#aa zone mis-match trouble has cleared
243		

#	Description	Notes
244	alert sect-n 0.000%	vesda detector has reached alert level in sector 'n'
245	periph #aa drill on	peripheral device of id#aa has active drill sw.
246	periph #aa drill off	peripheral device of id#aa has de-active drill sw.
247	network module trubl	trouble detected on the network card
248	network module clear	trouble restored from the network card
249	device needed reset	output device required reset to restore its configuration
250	action sect-n 0.000%	vesda detector has reached alert level in sector 'n'
251	fire-1 sect-n 0.000%	vesda detector has reached action level in sector 'n'
252	fire-2 sect-n 0.000%	vesda detector has reached alarm level in sector 'n'
253		event 253 reserved for the history module
254		event 254 reserved for the history module
255		event 255 reserved for the history module