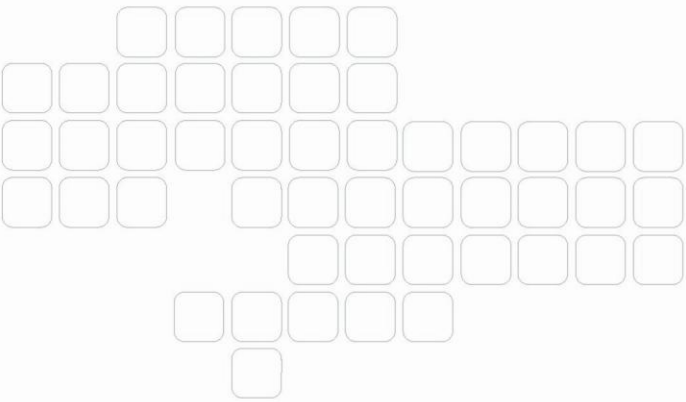


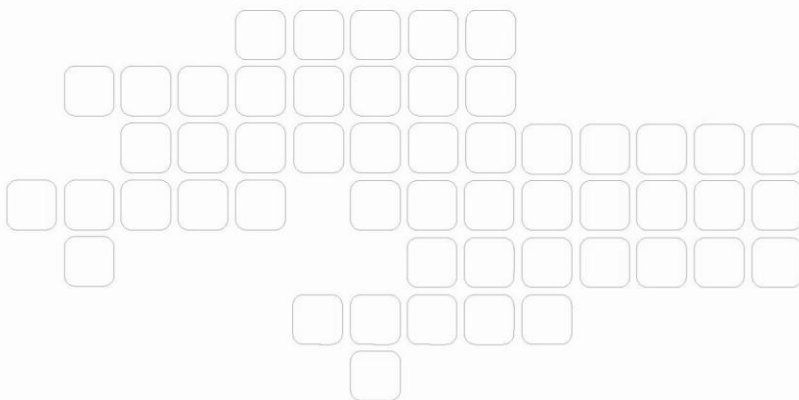


USER'S MANUAL

# FireSync Host Protocol

Platform Version 4.11.5.5

Protocol Version 1.1



Copyright, © 2006, LMI Technologies, Inc. All rights reserved.

## **Proprietary**

This document, submitted in confidence, contains proprietary information which shall not be reproduced or transferred to other documents or disclosed to others or used for manufacturing or any other purpose without prior written permission of LMI Technologies Inc.

No part of this publication may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine readable form without prior written consent of LMI Technologies, Inc.

## **Trademarks and Restrictions**

DynaVision®, chroma+scan®, Selcom®, FireSync®, and Sensors That See® are registered trademarks of LMI Technologies, Inc. Any other company or product names mentioned herein may be trademarks of their respective owners. Information in this manual is subject to change.

This product is designated for use solely as a component and as such it does not comply with the standards relating to laser products specified in U.S. FDA CFR Title 21 Part 1040.

LMI Technologies, Inc.  
1673 Cliveden Ave.  
Delta, BC V3M 6V5  
Canada

Telephone: +1 604 636 1011  
Facsimile: +1 604 516 8368  
[www.sensorthatsee.com](http://www.sensorthatsee.com)

# Table of Contents

|                                |    |
|--------------------------------|----|
| Introduction .....             | 5  |
| FireSync Host Protocol.....    | 6  |
| Communication Mechanism.....   | 6  |
| Data Types.....                | 6  |
| IP Address .....               | 8  |
| Command Header .....           | 8  |
| Reply Header .....             | 8  |
| System States.....             | 9  |
| System Operation Commands..... | 10 |
| Get Protocol Version .....     | 10 |
| Start System .....             | 11 |
| Stop System.....               | 12 |
| Pause System.....              | 12 |
| Resume System.....             | 13 |
| Set Operation Mode .....       | 13 |
| Get Operation Mode.....        | 14 |
| Write File.....                | 14 |
| Write Volatile File .....      | 15 |
| Read File.....                 | 15 |
| Delete File.....               | 16 |
| Enumerate Sensor .....         | 16 |
| Get System Current Time.....   | 17 |
| Data Channel Enable .....      | 17 |
| Data Channel Disable .....     | 18 |
| Run System .....               | 18 |
| Ping Command .....             | 19 |
| Start Event Channel .....      | 20 |
| Stop Event Channel .....       | 20 |
| Get System Start Delay.....    | 21 |
| Backup Storage.....            | 21 |
| Restore Storage.....           | 22 |
| Get Server Info.....           | 22 |
| Get Sensor Info.....           | 23 |
| Set Parameter.....             | 24 |
| Get Parameter.....             | 24 |
| Enable Data Keep-alive.....    | 25 |
| Disable Data Keep-alive.....   | 26 |
| Management Commands.....       | 27 |
| Get Protocol Version .....     | 27 |
| Start Health Channel .....     | 27 |
| Stop Health Channel .....      | 28 |
| Set Public IP .....            | 28 |
| Ping Command .....             | 29 |
| Get Private IP .....           | 30 |
| Set Private IP .....           | 30 |
| Get Health Log.....            | 31 |
| Get Station Dumps.....         | 31 |

|                                   |    |
|-----------------------------------|----|
| Get Report .....                  | 32 |
| Upgrade Commands.....             | 33 |
| Get Protocol Version .....        | 33 |
| Start Upgrade.....                | 33 |
| Get Status .....                  | 34 |
| Get Log .....                     | 35 |
| UDP Broadcast Protocol .....      | 36 |
| Get IP Settings .....             | 36 |
| Set IP Command.....               | 37 |
| Data Channels and Format .....    | 39 |
| FireSync Result Format .....      | 40 |
| FireSync Sensor Data Channel..... | 41 |
| Keep-alive Message Format.....    | 41 |
| FireSync Health Data Channel..... | 41 |
| Health Data Format.....           | 41 |
| FireSync Event Data Channel.....  | 42 |
| Health Indicators .....           | 43 |
| Sensor Indicators.....            | 44 |
| Server Indicators .....           | 45 |
| Event Indicators.....             | 47 |

## Introduction

This document describes the FireSync®<sup>1</sup> Host protocol. The FireSync Host protocol is a standard protocol based on TCP/IP that allows customers to communicate with a FireSync server or a standalone sensor. Unless specially stated otherwise, a FireSync server and a standalone sensor accept FireSync Host protocol commands. Therefore, both FireSync servers and standalone sensors will be referred to simply as servers.

---

<sup>1</sup> FireSync is a registered trademark of LMI

## FireSync Host Protocol

### Communication Mechanism

A client application operates a FireSync server by sending commands and listening for responses over various command channels. These channels are TCP connections made by the client to predefined ports on the server. Different channels support different groups of commands. Below is a table of these predefined ports.

| Channel          | Port | Description                              |
|------------------|------|------------------------------------------|
| System Operation | 3190 | Core system operation control channel    |
| Management       | 3191 | System monitoring and management channel |
| Upgrade          | 3192 | System upgrade channel                   |

The system operation port is able to serve multiple connections concurrently. Commands are atomic operations; that is, if multiple commands are issued at the same time through multiple connections, they are processed in sequence. The other ports should be treated as capable of single connection only.

The commands and replies are sent as binary messages. Each command and reply begins with a header. The general data types are described in the next section, followed by the specification of the headers and commands.

In addition to the TCP command channels, there is a UDP-based broadcast protocol that can be used to set and get the IP addresses of the stations. This can be used on stations with unknown network configurations. It is described in detail in a later section.

### Data Types

The command specification uses a shorthand notation for data types. The data type definitions are:

| Data Types Name | Description              |
|-----------------|--------------------------|
| Char            | Unsigned 8-bit character |
| Byte            | Single byte              |
| 32u             | 32-bit unsigned integer  |
| 64s             | 64-bit signed integer    |

|         |                                           |
|---------|-------------------------------------------|
| 64u     | 64-bit unsigned integer                   |
| type[x] | array of x number of <i>type</i> elements |

All numbers are little-endian, i.e. lower order byte has lower memory address, unless otherwise stated.

## IP Address

The IP address and subnet mask are represented as arrays of 4 bytes, with the first element corresponding to the first value in the dotted decimal format. For example, IP address 192.168.1.10 is specified as:

```
address[0] = 192  
address[1] = 168  
address[2] = 1  
address[3] = 10
```

## Command Header

Each command begins with a header, defined as:

| Field   | Type | Description                                     |
|---------|------|-------------------------------------------------|
| length  | 64s  | Message length in bytes (including header size) |
| command | 64s  | Command Identifier                              |

## Reply Header

Similar to command, each reply begins with a header, defined as:

| Field   | Type | Description                                     |
|---------|------|-------------------------------------------------|
| length  | 64s  | Message length in bytes (including header size) |
| command | 64s  | Command Identifier                              |
| status  | 64s  | Status code                                     |

The status code is defined as:

| Label              | Value | Description                                                                                                                    |
|--------------------|-------|--------------------------------------------------------------------------------------------------------------------------------|
| Invalid State      | -1000 | The command failed because it was received during an invalid state. The various system states are described below.             |
| Item not found     | -999  | The command failed because the requested item does not exist                                                                   |
| Invalid Command    | -998  | The command ID is not recognized                                                                                               |
| Invalid Parameters | -997  | The parameters for the command are incorrect. A parameter could be missing, of the wrong size, or simply have the wrong value. |
| Failed             | 0     | The command failed                                                                                                             |
| Ok                 | 1     | The command was successfully executed                                                                                          |



## System States

A FireSync system can be in one of several states. The current state of the system determines what commands can be issued to the server. These states are explained below:

| State   | Value | Description                                                                                                      |
|---------|-------|------------------------------------------------------------------------------------------------------------------|
| Idle    | 1     | The system has started up but has detected errors that prevent the system from being able to run.                |
| Ready   | 2     | The system is ready to run.                                                                                      |
| Running | 3     | The system is in full-operation mode. The data transfer channel is connected and data is sent when appropriate.  |
| Standby | 4     | The system is paused. The data transfer channel remains connected and the system is ready to resume immediately. |

## System Operation Commands

The following is a table of the core system commands. These commands are used in typical sensor operations.

| Command                 | Identifier |
|-------------------------|------------|
| Get Protocol Version    | 0x1100     |
| Start System            | 0x1000     |
| Stop System             | 0x1001     |
| Pause System            | 0x1002     |
| Resume System           | 0x1003     |
| Set Operation Mode      | 0x1004     |
| Get Operation Mode      | 0x1005     |
| Write File              | 0x1006     |
| Write Volatile File     | 0x1017     |
| Read File               | 0x1007     |
| Delete File             | 0x1008     |
| Enumerate Sensors       | 0x1009     |
| Get System Current Time | 0x100A     |
| Data Channel Enable     | 0x100B     |
| Data Channel Disable    | 0x100C     |
| System Run Command      | 0x100D     |
| System Ping Command     | 0x100E     |
| Start Event Channel     | 0x1010     |
| Stop Event Channel      | 0x1011     |
| Get System Start Delay  | 0x1012     |
| Backup Storage          | 0x1013     |
| Restore Storage         | 0x1014     |
| Get Server Info         | 0x1015     |
| Get Sensor Info         | 0x1016     |
| Set Parameter           | 0x1018     |
| Get Parameter           | 0x1019     |
| Enable Data Keep-alive  | 0x1020     |
| Disable Data Keep-alive | 0x1021     |

### Get Protocol Version

Use this command to retrieve the protocol version. The protocol version described by this document can be found on the title page (major.minor).

This command is available in protocol version 1.1 and above.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description          |
|--------------|--------------|----------------------|
| header       | Reply Header | Command reply header |
| majorVersion | 64s          | Major version.       |
| minorVersion | 64s          | Minor version.       |

Return Value:

OK

## Start System

This command starts up the system. Upon receiving the command, the FireSync server initiates a TCP connection to the specified TCP end-point (based on *address* and *port* fields). If the connection is successful, the sensor data channel is established and data will be sent on this channel. The sensor data channel is described in a later section.

This command is used to start a system which requires only one data channel. To start the system with multiple data channels, use the enable/disable data channel commands, and the run system command.

Command Message:

| Field       | Type           | Description                                                              |
|-------------|----------------|--------------------------------------------------------------------------|
| header      | Command Header | Command message header                                                   |
| reserved[4] | byte           | Reserved                                                                 |
| address[4]  | byte           | IP address of data receiving TCP connection                              |
| port        | 64s            | Port of data receiving TCP connection                                    |
| startTime   | 64u            | The start time of the system. Set to 0 to start the system without delay |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value   | Description                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------|
| Invalid state | Returned if the server is in booting or idle state, or a system is already running              |
| Failed        | FireSync server cannot establish TCP connection, or if operation failed on more or more sensors |

## Stop System

This command stops the system. The data TCP connection will be terminated by the FireSync server upon receiving this command

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value | Description                                |
|-------------|--------------------------------------------|
| Failed      | If operation failed on one or more sensors |

## Pause System

This command puts all sensors into standby mode. All light sources (eg. lasers) turn off.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value   | Description                                                                                    |
|---------------|------------------------------------------------------------------------------------------------|
| Invalid state | Returned if the server is not running.                                                         |
| Failed        | FireSync server cannot establish TCP connection, or if operation failed on one or more sensors |

## Resume System

This command resumes all sensors from standby mode. All light sources (e.g. lasers) are turned on.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value   | Description                                                                                    |
|---------------|------------------------------------------------------------------------------------------------|
| Invalid state | Returned if the server is not in standby.                                                      |
| Failed        | FireSync server cannot establish TCP connection, or if operation failed on one or more sensors |

## Set Operation Mode

This command sets the operation mode of the sensor

Command Message:

| Field    | Type           | Description            |
|----------|----------------|------------------------|
| header   | Command Header | Command message header |
| Mode[16] | char           | Name of mode to set    |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

This command sets the current mode by its name.

Return Value:

OK if successful.

| Error Value   | Description                                                |
|---------------|------------------------------------------------------------|
| Invalid state | Returned if the server is currently running, or in standby |

|        |                                    |
|--------|------------------------------------|
| Failed | If the name of the mode is invalid |
|--------|------------------------------------|

## Get Operation Mode

This command retrieves the name of the currently active operation mode.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field    | Type         | Description                  |
|----------|--------------|------------------------------|
| header   | Reply Header | Command reply message header |
| Mode[16] | char         | Name of the current mode     |

Return Value:

This command always returns OK.

## Write File

The write file command is used to create a new file or update an existing one on the server. If a file already exists with the given name, it is over-written; otherwise a new file is created.

Command Message:

| Field              | Type           | Description               |
|--------------------|----------------|---------------------------|
| header             | Command Header | Command message header    |
| filename[64]       | char           | Name of the file to write |
| fileSize           | 64s            | Size of the file          |
| fileData[fileSize] | byte           | File contents             |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value | Description                          |
|-------------|--------------------------------------|
| Failed      | File could not be updated or created |

## Write Volatile File

The Write Volatile File command is identical to the “Write File” command, except that the files written are not saved to non-volatile storage; i.e. any new files, or changes to existing files are discarded after power cycling. The distinction exists only for stand-alone devices and sensors. This command should not be used to write files to a FireSync server as it behaves the same way as the “Write File” command.

It should be noted that, if the “Write File” command is used after “Write Volatile File” commands, then any changes written by the latter are saved to storage.

Command Message:

| Field              | Type           | Description               |
|--------------------|----------------|---------------------------|
| header             | Command Header | Command message header    |
| filename[64]       | char           | Name of the file to write |
| fileSize           | 64s            | Size of the file          |
| fileData[fileSize] | byte           | File contents             |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value | Description                          |
|-------------|--------------------------------------|
| Failed      | File could not be updated or created |

## Read File

This command is used to read the contents of a file from the server. Simply send a command containing the name of the file that is to be accessed, and the server responds with the file data (and length) if the file exists.

Command Message:

| Field        | Type           | Description              |
|--------------|----------------|--------------------------|
| header       | Command Header | Command message header   |
| filename[64] | char           | Name of the file to read |

Reply Message:

| Field    | Type         | Description          |
|----------|--------------|----------------------|
| header   | Reply Header | Reply message header |
| fileSize | 64s          | Size of the file     |

|                    |      |               |
|--------------------|------|---------------|
| fileData[filesize] | byte | File contents |
|--------------------|------|---------------|

Return Value:

OK if successful.

| Error Value    | Description                                     |
|----------------|-------------------------------------------------|
| Failed         | File could not be read, or file does not exist. |
| Item not found | File does not exist                             |

## Delete File

Send this command to delete a file with the given name from the storage area of the server.

Command Message:

| Field        | Type           | Description              |
|--------------|----------------|--------------------------|
| header       | Command Header | Command message header   |
| filename[64] | char           | Name of the file to read |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value    | Description               |
|----------------|---------------------------|
| Failed         | File could not be deleted |
| Item not found | File does not exist       |

## Enumerate Sensor

Retrieve a list of sensors currently presented in the system. The number of sensors and their serial numbers are returned.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field | Type | Description |
|-------|------|-------------|
|-------|------|-------------|



|                        |              |                                     |
|------------------------|--------------|-------------------------------------|
| header                 | Reply Header | Command message header              |
| numSensors             | 64s          | Number of sensors present in system |
| serialNums[numSensors] | 64s          | Sensor serial numbers               |

Return Value:

This command always returns OK.

## Get System Current Time

Retrieves the number of microseconds since the system is powered up.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description            |
|--------------|--------------|------------------------|
| header       | Reply Header | Command message header |
| Current time | 64u          | Time passed            |

Return Value:

OK if successful.

| Error Value | Description                        |
|-------------|------------------------------------|
| Failed      | The system time could not be read. |

## Data Channel Enable

Enable an output data channel by providing a channel identifier and a TCP end-point (based on *address* and the *port* fields). The FireSync server will initiate connection to all enabled data channels when it starts.

This command can be processed only when system is stopped.

Command Message:

| Field       | Type           | Description                                 |
|-------------|----------------|---------------------------------------------|
| header      | Command Header | Command message header                      |
| reserved[4] | byte           | Reserved                                    |
| address[4]  | byte           | IP address of data receiving TCP connection |
| port        | 64s            | Port of data receiving TCP connection       |
| channel ID  | 64s            | Between 0 to 63                             |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value   | Description                            |
|---------------|----------------------------------------|
| Failed        | The data channel could not be enabled. |
| Invalid State | The system is not currently stopped    |

## Data Channel Disable

Disable an output data channel. This command can be processed only when system is stopped.

Command Message:

| Field      | Type           | Description            |
|------------|----------------|------------------------|
| header     | Command Header | Command message header |
| channel ID | 64s            | Between 0 to 63        |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value   | Description                             |
|---------------|-----------------------------------------|
| Failed        | The data channel could not be disabled. |
| Invalid State | The system is not currently stopped     |

## Run System

This command starts up the system. Upon receiving the command, the FireSync server initiates a TCP connection to each enabled data channel.

Command Message:

| Field     | Type           | Description                                                              |
|-----------|----------------|--------------------------------------------------------------------------|
| header    | Command Header | Command message header                                                   |
| startTime | 64u            | The start time of the system. Set to 0 to start the system without delay |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value   | Description                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------|
| Invalid state | Returned if the server is in booting or idle state, or a system is already running.              |
| Failed        | FireSync server cannot establish TCP connection, or if operation failed on more or more sensors. |

## Ping Command

This command allows the client to ping the server to detect if the connection to the server is still alive.

The timeout field sets the ping timeout timer for the system operation channel.

If the timeout period is set to non-zero, the client must send the next ping command to the server at an interval no longer than the specified timeout period. The server will disconnect the command channel if it does not receive the next ping command within the specified period.

If the timeout period is set to zero, the ping monitoring mechanism is disabled, and the client has no obligation to send the ping command. This is the default behavior on system startup.

Command Message:

| Field   | Type           | Description                         |
|---------|----------------|-------------------------------------|
| header  | Command Header | Command message header              |
| timeout | 64s            | Ping timeout period. (microseconds) |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

This command always returns OK.

## Start Event Channel

This command starts the transmission of asynchronous events to a TCP server (specified by the *address* and *port* field). Similar to the System Start command, the FireSync server initiates the TCP connection. The event data channel is described in a later section.

The TCP server must be located at a subnet that could be connected by the Master, which is located in the 90.x.x.x subnet.

Command Message:

| Field       | Type           | Description                                 |
|-------------|----------------|---------------------------------------------|
| header      | Command Header | Command message header                      |
| reserved[4] | byte           | Reserved                                    |
| address[4]  | byte           | IP address of data receiving TCP connection |
| port        | 64s            | Port of data receiving TCP connection       |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value | Description                                 |
|-------------|---------------------------------------------|
| Failed      | If the TCP connection cannot be established |

## Stop Event Channel

This command stops event data streaming and disconnects the health data connection.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value | Description                             |
|-------------|-----------------------------------------|
| Failed      | The health channel could not be stopped |

## Get System Start Delay

This command returns the maximum time (in microseconds) that it takes the server to process a Start command and reply to the client. This value is useful when multiple servers must be started in a synchronized fashion. When starting such a system, it must be guaranteed that all servers have received and processed the start command before the start time value in the command.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field      | Type         | Description                                                                     |
|------------|--------------|---------------------------------------------------------------------------------|
| header     | Reply Header | Command reply message header                                                    |
| startDelay | k64s         | The maximum time this server takes to process a Start command, in microseconds. |

Return Value:

OK if successful.

| Error Value | Description                             |
|-------------|-----------------------------------------|
| Failed      | The health channel could not be stopped |

## Backup Storage

Issue this command to retrieve an image of the current server storage. If the operation is successful, the server responds with the image length and data.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field                | Type         | Description          |
|----------------------|--------------|----------------------|
| header               | Reply Header | Reply message header |
| imageSize            | 64s          | Size of the image    |
| imageData[imageSize] | byte         | Image data           |

Return Value:

OK if successful.

| Error Value | Description                        |
|-------------|------------------------------------|
| Failed      | Backup image could not be created. |

## Restore Storage

This command can be used to restore the storage of the server with a backup image file. All existing files are overwritten on restore.

Command Message:

| Field               | Type           | Description              |
|---------------------|----------------|--------------------------|
| header              | Command Header | Command message header   |
| imageSize           | 64s            | Size of the backup image |
| fileData[imageSize] | byte           | Backup image data        |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value | Description                        |
|-------------|------------------------------------|
| Failed      | Backup image could not be restored |

## Get Server Info

This command retrieves information about the server such as its serial number and model. This command will fail if the system is running.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

|              |           |                                                                                                                                                                                  |
|--------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| serialNumber | 64s       | Station serial number                                                                                                                                                            |
| modelName    | char[32]  | Station model name                                                                                                                                                               |
| storageFree  | 64s       | Free storage space in bytes                                                                                                                                                      |
| appVersion   | 64s       | Server application version.<br>(Bits 63 – 32: Reserved<br>Bits 31 – 24: Major Version<br>Bits 23 – 16: Minor Version<br>Bits 15 – 8: Build Version<br>Bits 7 – 0: Patch Version) |
| reserved     | byte[240] | Reserved for future and internal use                                                                                                                                             |

Return Value:

OK if successful.

| Error Value | Description                    |
|-------------|--------------------------------|
| Failed      | Could not read the information |

## Get Sensor Info

This command retrieves information about the server such as its serial number and model. This command will fail if the system is running.

Command Message:

| Field        | Type           | Description                 |
|--------------|----------------|-----------------------------|
| header       | Command Header | Command message header      |
| serialNumber | 64s            | Serial number of the sensor |

Reply Message:

| Field        | Type         | Description                                                                                                                                                                      |
|--------------|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| header       | Reply Header | Reply message header                                                                                                                                                             |
| serialNumber | 64s          | Sensor serial number                                                                                                                                                             |
| modelName    | char[32]     | Sensor model name                                                                                                                                                                |
| appVersion   | k64s         | Sensor application version.<br>(Bits 63 – 32: Reserved<br>Bits 31 – 24: Major Version<br>Bits 23 – 16: Minor Version<br>Bits 15 – 8: Build Version<br>Bits 7 – 0: Patch Version) |
| reserved     | byte[248]    | Reserved for future and internal use                                                                                                                                             |

Return Value:

OK if successful.

| Error Value | Description                    |
|-------------|--------------------------------|
| Failed      | Could not read the information |

## Set Parameter

The Set Parameter command allows the configuration of mode parameters. Unlike the XML configuration file, mode parameters are not stored in non-volatile memory. Parameters are identified by the mode name containing the parameter, and the parameter name itself.

Mode parameters are available only on some sensors, and cannot be set while the system is running or paused.

Command Message:

| Field          | Type           | Description            |
|----------------|----------------|------------------------|
| header         | Command Header | Command message header |
| modeName       | char[16]       | Name of the mode       |
| parameterName  | char[16]       | Name of the parameter  |
| parameterValue | 64s            | New parameter value.   |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value       | Description                                       |
|-------------------|---------------------------------------------------|
| Failed            | Could not set the value                           |
| Invalid Parameter | Could not find the parameter                      |
| Invalid State     | Could not set the value because system is running |

## Get Parameter

The Get Parameter command retrieves mode parameters. See Set Parameter for more information.

Mode parameters are available only on some sensors, and cannot be read while the system is running or paused.

Command Message:

| Field         | Type           | Description            |
|---------------|----------------|------------------------|
| header        | Command Header | Command message header |
| modeName      | char[16]       | Name of the mode       |
| parameterName | char[16]       | Name of the parameter  |

Reply Message:



| Field          | Type         | Description              |
|----------------|--------------|--------------------------|
| header         | Reply Header | Reply message header     |
| parameterValue | 64s          | Current parameter value. |

Return Value:

OK if successful.

| Error Value       | Description                                       |
|-------------------|---------------------------------------------------|
| Failed            | Could not get the value                           |
| Invalid Parameter | Could not find the parameter                      |
| Invalid State     | Could not get the value because system is running |

## Enable Data Keep-alive

This command enables the sending of keep-alive messages over the data channels.

Some network stacks can exhibit higher than normal latency when data is sent over a TCP connection that had been inactive for too long. When keep-alive is enabled, a message is sent over any data channel that has remained inactive for a certain amount of time.

The keep-alive message has an ID of 900 and no extra data. Refer to the Data Channels and Format section for the format.

This command is available in protocol version 1.1 and above.

Command Message:

| Field  | Type           | Description                                                                                                                                    |
|--------|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| header | Command Header | Command message header                                                                                                                         |
| period | 64s            | Maximum idle time (microseconds).<br><br>If a channel has been inactive for this number of microseconds, a keep-alive message is sent over it. |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value | Description                      |
|-------------|----------------------------------|
| Failed      | Could not enable data keep-alive |

## Disable Data Keep-alive

This command disables the sending of keep-alive messages.

It is available in protocol version 1.1 and above.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Reply message header |

Return Value:

OK if successful.

| Error Value | Description                       |
|-------------|-----------------------------------|
| Failed      | Could not disable data keep-alive |

## Management Commands

The management commands are used for sensor system monitoring and server management.

| Command              | Identifier |
|----------------------|------------|
| Get Protocol Version | 0x2100     |
| Start Health Channel | 0x2000     |
| Stop Health Channel  | 0x2001     |
| Set Public IP        | 0x2002     |
| Ping                 | 0x2003     |
| Get Private IP       | 0x2004     |
| Set Private IP       | 0x2005     |
| Get Health Log       | 0x2006     |
| Get Station Dumps    | 0x2007     |
| Get Report           | 0x2008     |

### Get Protocol Version

Use this command to retrieve the protocol version. The protocol version described by this document can be found on the title page (major.minor).

This command is available in protocol version 1.1 and above.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description          |
|--------------|--------------|----------------------|
| header       | Reply Header | Command reply header |
| majorVersion | 64s          | Major version.       |
| minorVersion | 64s          | Minor version.       |

Return Value:

OK

### Start Health Channel

This command starts the transmission of system health information to a TCP server (specified by the *address* and *port* field). Similar to the System Start command, the

FireSync server initiates the TCP connection. The health data channel is described in a later section.

Command Message:

| Field       | Type           | Description                                 |
|-------------|----------------|---------------------------------------------|
| header      | Command Header | Command message header                      |
| reserved[4] | byte           | Reserved                                    |
| address[4]  | byte           | IP address of data receiving TCP connection |
| port        | 64s            | Port of data receiving TCP connection       |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

OK if successful.

| Error Value | Description                                 |
|-------------|---------------------------------------------|
| Failed      | If the TCP connection cannot be established |

## Stop Health Channel

This command stops health data streaming and disconnects the health data connection.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value | Description                             |
|-------------|-----------------------------------------|
| Failed      | The health channel could not be stopped |

## Set Public IP

This command changes the IP address of the public interface of the FireSync server. If the command returns success, the IP will be changed when the management connection is broken from the server. If the address field is 0, then automatic IP configuration by DHCP will be enabled.

Note: The subnet 90.0.0.0/255.0.0.0 is reserved for private use; therefore any address in this subnet is not valid for this command.

Command Message:

| Field       | Type           | Description            |
|-------------|----------------|------------------------|
| header      | Command Header | Command message header |
| reserved[4] | byte           | Reserved               |
| address[4]  | byte           | New network IP address |
| reserved[4] | byte           | Reserved               |
| mask[4]     | byte           | New subnet mask        |
| reserved[4] | byte           | Reserved               |
| gateway[4]  | byte           | New gateway address    |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value | Description                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------|
| Failed      | IP cannot be changed. This happens if the IP specified is invalid, or within the subnet of 90.xxx.xxx.xxx |

## Ping Command

This command allows the client to ping the server to detect if the connection to the server is still alive.

The timeout field sets the ping timeout timer for the management channel.

If the timeout period is set to non-zero, then the client must send the next ping command to the server at an interval no longer than the specified timeout period. The server will disconnect the management channel if it does not receive the next ping command within the specified period.

If the timeout period is set to zero, then the ping monitoring mechanism is disabled, and the client has no obligation to send the ping command. This is the default behavior on system startup.

Command Message:

| Field   | Type           | Description                         |
|---------|----------------|-------------------------------------|
| header  | Command Header | Command message header              |
| timeout | 64s            | Ping timeout period. (microseconds) |

Reply Message:

| Field  | Type         | Description          |
|--------|--------------|----------------------|
| header | Reply Header | Command reply header |

Return Value:

This command always returns OK.

### Get Private IP

This command retrieves the private interface IP of the FireSync server.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field       | Type         | Description                  |
|-------------|--------------|------------------------------|
| header      | Reply Header | Command reply message header |
| reserved[4] | byte         | Reserved                     |
| Address[4]  | byte         | New IP address               |

Return Value:

OK if successful.

| Error Value | Description                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Failed      | The IP address of the private interface cannot be retrieved. This error can occur if the sensor interface is disconnected from any switch device (master or slave). |

### Set Private IP

The private interface IP can be changed through this command. In a system configuration where there are multiple servers on the same sensor data network, the servers must have different IPs on that network.

The valid addresses are 90.0.0.1 through 90.0.0.5. To ensure that there is no conflict, the Get Private IP can be used to retrieve the private addresses of the other servers.

Command Message:

| Field       | Type           | Description                    |
|-------------|----------------|--------------------------------|
| header      | Command Header | Command message header         |
| reserved[4] | byte           | Reserved                       |
| address[4]  | byte           | New private network IP address |

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value | Description                                                        |
|-------------|--------------------------------------------------------------------|
| Failed      | IP cannot be changed. This happens if the IP specified is invalid. |

## Get Health Log

The station keeps a internal log of the health data, which can be retrieved via this command. The health data can be analyzed to aid in troubleshooting. The file has a proprietary format and should be sent to technical support for analysis.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field      | Type         | Description                  |
|------------|--------------|------------------------------|
| header     | Reply Header | Command reply message header |
| size       | 64s          | Size of the log file         |
| Data[size] | byte         | Log file data                |

Return Value:

OK if successful.

| Error Value | Description                     |
|-------------|---------------------------------|
| Failed      | The log could not be retrieved. |

## Get Station Dumps

In the event of a station process failure, some debug information is saved onto the station. This command retrieves this information as a single package. This package should be sent to technical support for analysis.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field      | Type         | Description                  |
|------------|--------------|------------------------------|
| header     | Reply Header | Command reply message header |
| size       | 64s          | Size of the log file         |
| Data[size] | byte         | Dump archive data            |

Return Value:

OK if successful.

| Error Value | Description                            |
|-------------|----------------------------------------|
| Failed      | The dump files could not be retrieved. |

## Get Report

Download the health analysis report as a text file.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field      | Type         | Description                  |
|------------|--------------|------------------------------|
| header     | Reply Header | Command reply message header |
| size       | 64s          | Size of the report file      |
| data[size] | byte         | Analysis report file.        |

Return Value:

OK if successful.

| Error Value | Description                             |
|-------------|-----------------------------------------|
| Failed      | The report file could not be retrieved. |



## Upgrade Commands

The upgrade commands are used for system upgrades. Typical usage is to transmit an upgrade package file through the Start Upgrade command, and then monitor the progress with the Get Status command until the process is finished. The Get Log command can be used to retrieve an error log if problem occurs.

| Command              | Identifier |
|----------------------|------------|
| Get Protocol Version | 0x0100     |
| Start Upgrade        | 0x0000     |
| Get Status           | 0x0001     |
| Get Log              | 0x0002     |

### Get Protocol Version

Use this command to retrieve the protocol version. The protocol version described by this document can be found on the title page (major.minor).

This command is available in protocol version 1.1 and above.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description          |
|--------------|--------------|----------------------|
| header       | Reply Header | Command reply header |
| majorVersion | 64s          | Major version.       |
| minorVersion | 64s          | Minor version.       |

Return Value:

OK

### Start Upgrade

This command begins the upgrade process by transmitting an upgrade package file to the server.

Command Message:

| Field       | Type           | Description                      |
|-------------|----------------|----------------------------------|
| header      | Command Header | Command message header           |
| packageSize | k64s           | Size of the upgrade package file |

|                      |      |                                   |
|----------------------|------|-----------------------------------|
| package[packageSize] | byte | Image of the upgrade package file |
|----------------------|------|-----------------------------------|

Reply Message:

| Field  | Type         | Description                  |
|--------|--------------|------------------------------|
| header | Reply Header | Command reply message header |

Return Value:

OK if successful.

| Error Value   | Description                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Failed        | The upgrade cannot be started due to unspecified error. The Get Log command can be used to retrieve detailed description of the failure. |
| Invalid State | The system is currently in a state that does not allow upgrade (running or paused).                                                      |

## Get Status

This command queries the current status of the upgrade progress.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description                                                  |
|--------------|--------------|--------------------------------------------------------------|
| header       | Reply Header | Command reply message header                                 |
| step         | 64s          | The current upgrade process step (see below)                 |
| stepProgress | 64s          | Progress of the currently running step as a percentage value |

Here is a table of the possible upgrade step values

| Step              | Value | Description                                 |
|-------------------|-------|---------------------------------------------|
| Upgrade Completed | 0     | The previous upgrade completed successfully |
| Upgrade Failed    | -1    | The previous upgrade failed                 |
| Upgrading Server  | 1     | Upgrading server files                      |
| Upgrading Sensors | 2     | Upgrading sensor firmware                   |
| Resetting Sensors | 3     | Resetting all sensors after firmware load   |

Remarks: Once the upgrade process stops due to completion or error, the current step is either Upgrade Completed or Upgrade Failed.

Return Value:

OK if successful.

| Error Value | Description                         |
|-------------|-------------------------------------|
| Failed      | Could not retrieve upgrade progress |

## Get Log

In case of upgrade programs, the Get Log command can be used to retrieve the upgrade log and aid in troubleshooting.

Command Message:

| Field  | Type           | Description            |
|--------|----------------|------------------------|
| header | Command Header | Command message header |

Reply Message:

| Field        | Type         | Description                      |
|--------------|--------------|----------------------------------|
| header       | Reply Header | Command reply message header     |
| logSize      | k64s         | Size of the upgrade log          |
| Log[logSize] | char         | The upgrade log in string format |

Return Value:

OK if successful.

| Error Value | Description                        |
|-------------|------------------------------------|
| Failed      | Could not retrieve the upgrade log |

## UDP Broadcast Protocol

The UDP broadcast protocol can be used to retrieve and set the network configuration of a station with unknown settings. To send command to a station, a UDP packet must be send to the following broadcast destination:

| Destination Address | Destination Port |
|---------------------|------------------|
| 255.255.255.255     | 3220             |

Each command identifies its target with a serial number. After the command is processed, the station generates another broadcast UDP packet that is sent to the port of the command sender.

| Destination Address | Destination Port       |
|---------------------|------------------------|
| 255.255.255.255     | Port of command sender |

### Get IP Settings

Send this command to retrieve the current network settings of a station. If a network interface is not connected at the time, then “0.0.0.0” is returned for that interface.

Command Message:

| Field        | Type | Description                          |
|--------------|------|--------------------------------------|
| length       | 64s  | Total message size (0x20)            |
| id           | 64s  | 0x0001                               |
| signature    | 64s  | 0x0000504455494D4C                   |
| serialNumber | 64s  | Serial number of destination station |

Reply Message:

| Field             | Type | Description                                  |
|-------------------|------|----------------------------------------------|
| length            | 64s  | Total message size (0x48)                    |
| id                | 64s  | 0x1001                                       |
| status            | 64s  | Return code                                  |
| signature         | 64s  | 0x0000504455494D4C                           |
| serialNumber      | 64s  | Serial number of source station              |
| publicIsDhcp      | 64s  | 1 if public interface uses DHCP. 0 otherwise |
| reserved[4]       | byte | Reserved                                     |
| publicAddress[4]  | byte | Public interface address                     |
| reserved[4]       | byte | Reserved                                     |
| publicMask[4]     | byte | Public interface mask                        |
| reserved[4]       | byte | Reserved                                     |
| publicGateway[4]  | byte | Public interface gateway                     |
| reserved[4]       | byte | Reserved                                     |
| privateAddress[4] | byte | Private interface address                    |

Return Value:

OK if successful.

| Error Value | Description           |
|-------------|-----------------------|
| Failed      | Could not get address |

## Set IP Command

This command is used to set the network settings of the station. If the public interface does not use DHCP and the address is specified to be "0.0.0.0", then the public address is not set. Likewise, specifying "0.0.0.0" for the private address causes it to be ignored also.

Command Message:

| Field             | Type | Description                                                                                                    |
|-------------------|------|----------------------------------------------------------------------------------------------------------------|
| length            | 64s  | Total message size (0x40)                                                                                      |
| id                | 64s  | 0x0002                                                                                                         |
| signature         | 64s  | 0x0000504455494D4C                                                                                             |
| serialNumber      | 64s  | Serial number of destination station                                                                           |
| publicIsDhcp      | 64s  | 1 if public interface uses DHCP. 0 otherwise. If DHCP is used, then the static address information is ignored. |
| reserved[4]       | byte | Reserved                                                                                                       |
| publicAddress[4]  | byte | Public interface address                                                                                       |
| reserved[4]       | byte | Reserved                                                                                                       |
| publicMask[4]     | byte | Public interface mask                                                                                          |
| reserved[4]       | byte | Reserved                                                                                                       |
| publicGateway[4]  | byte | Public interface gateway                                                                                       |
| reserved[4]       | byte | Reserved                                                                                                       |
| privateAddress[4] | byte | Private interface address                                                                                      |

Reply Message:

| Field        | Type | Description                     |
|--------------|------|---------------------------------|
| length       | 64s  | Total message size (0x28)       |
| id           | 64s  | 0x1002                          |
| status       | 64s  | Return code                     |
| signature    | 64s  | 0x0000504455494D4C              |
| serialNumber | 64s  | Serial number of source station |

Return Value:

OK if successful.

| Error Value | Description                   |
|-------------|-------------------------------|
| Failed      | Could not set the address(es) |



### **Data Channels and Format**

While the command channels are used to operate the system, information from a server is sent back to the client over data channels. There are three such channels: the sensor data channel, the health data channel and the event channel.

A common generic data format is used over all of these channels. This format is referred to as the FireSync Result format and is described in details in this chapter.

## FireSync Result Format

The FireSync Result format has the following general structure.

| Field           |                   | Type  | Description                                               |
|-----------------|-------------------|-------|-----------------------------------------------------------|
| size            |                   | 64s   | Total size of message                                     |
| id              |                   | 64s   | Message identifier                                        |
| attributeCount  |                   | 64s   | Number of attributes                                      |
| itemCount       |                   | 64s   | Number of items                                           |
| attributes      |                   | 64s   | first attribute                                           |
|                 |                   | ...   |                                                           |
|                 |                   | 64s   | n <sup>th</sup> attribute (n specified by attributeCount) |
| itemDescriptors | length0           | 64s   | First-dimension length of first item                      |
|                 | length1           | 64s   | Second-dimension length of first item                     |
|                 | length2           | 64s   | Third-dimension length of first item                      |
|                 | type              | 64s   | Item type of first item                                   |
|                 | ...               |       |                                                           |
|                 | length0           | 64s   | First-dimension length of n <sup>th</sup> item            |
|                 | length1           | 64s   | Second-dimension length of n <sup>th</sup> item           |
|                 | length2           | 64s   | Third-dimension length of n <sup>th</sup> item            |
|                 | type              | 64s   | Item type of n <sup>th</sup> item                         |
|                 | ...               |       |                                                           |
| itemData        | data0             | block | Content of first item                                     |
|                 | ...               |       |                                                           |
|                 | data <sub>n</sub> | block | Content of of n <sup>th</sup> item                        |

Each item descriptor describes the block of data at the corresponding index in the data section. The data block is the flatten result of an array which can be in any of one, two, or three dimensions. The length fields in the descriptor contain the lengths in the three dimensions. Only length0 is valid for a one dimensional array, and length0 and length1 for a two dimensional array. The lengths are followed by the item type, which is an identifier for a primitive type, and is defined by the following table.

| Type Identifier | Type Description             |
|-----------------|------------------------------|
| 1               | 8-bit unsigned integer       |
| 2               | 8-bit signed integer         |
| 3               | 16-bit unsigned integer      |
| 4               | 16-bit signed integer        |
| 5               | 32-bit unsigned integer      |
| 6               | 32-bit signed integer        |
| 7               | 64-bit unsigned integer      |
| 8               | 64-bit signed integer        |
| 9               | byte                         |
| 10              | single byte character        |
| 11              | 64-bit floating point number |

To access the data of a two dimensional array, the item at subscripts [i][j] is located at item offset  $i * \text{length1} + j$ ; for a three dimensional array, the item at [i][j][k] is located at item offset  $i * \text{length1} * \text{length2} + j * \text{length2} + k$ .



## FireSync Sensor Data Channel

When a Start System command is issued by the client to the server, a connection is made by the server to the TCP end-point specified in the command. This connection is referred to as the sensor data channel. Once the system is started and the connection made, data is automatically streamed to the client at a pace defined by the particular sensor system application.

The general format of the data is compatible with the FireSync Result format; however the exact format depends on the sensor system application.

### Keep-alive Message Format

Keep-alive messages are sent when a data channel remains inactive and keep-alive has been enabled via the command. It has the following format (each field is a 64s):

| Field    | Value |
|----------|-------|
| size     | 32    |
| id       | 900   |
| reserved | 0     |
| reserved | 0     |

## FireSync Health Data Channel

When the Start Health Channel command is issued to the server and the health channel is established successfully by the server to the client machine on the specified port, health information is pushed automatically through on a pre-configured interval. Issuing a Stop Health Channel command stops this streaming and terminates the connection.

### Health Data Format

The health data channel uses the FireSync Result format which is described in the previous section. Each node in the system, sensor and server alike, sends out one FireSync Result message at a pre-configured interval.

The ID field of the result message indicates the type of the source node:

| Value | Type           |
|-------|----------------|
| 1000  | Health Message |

The attributes contain the serial number of the originating node:

| Attribute Index | Description                       |
|-----------------|-----------------------------------|
| 0               | Serial number of originating node |

The data is a 2-dimensional array of 64s, where each row is referred to as a health “indicator”. The number of indicators per sensor or server depends on the application, but is constant throughout system operation.

Each health indicator has the following format:

| Type | Description       |
|------|-------------------|
| 64s  | Unique Identifier |
| 64s  | Instance Id       |
| 64s  | Value             |

The content of a FireSync result block for probes are:

| Field           |            | Value                                          |
|-----------------|------------|------------------------------------------------|
| size            |            | Variable depending on number of probes         |
| id              |            | 1000                                           |
| attributeCount  |            | 1                                              |
| itemCount       |            | 1                                              |
| attributes      |            | Serial Number                                  |
| itemDescriptors | length0    | Number of indicators                           |
|                 | length1    | 3 (Unique identifier + Instance Id + value)    |
|                 | length2    | 0 (Not used)                                   |
|                 | type       | 8 (64-bit signed integer)                      |
| Indicators      | Indicator0 | First Indicator                                |
|                 | ...        | Up to number of indicator specified by length0 |

For a list of available indicators and their corresponding value field meanings, please refer to the health indicator section.

## FireSync Event Data Channel

The operation is identical to that of the health data channel, except Start/Stop Event Channel are used to control the channel, and the ID field of 1001 is used.

## Chapter 4

### **Health Indicators**

This section describes the indicators that are provided by the FireSync platform.

## Sensor Indicators

The following table describes the indicators that are transmitted for each sensor. Unless otherwise specified, all logging intervals begin at sensor power-up.

| Name           | ID   | Instance ID | Description                                                                                                                                                                               |
|----------------|------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App Version    | 1000 | N/A         | Sensor application firmware version.<br>(Bits 63 – 32: Reserved<br>Bits 31 – 24: Major Version<br>Bits 23 – 16: Minor Version<br>Bits 15 – 8: Build Version<br>Bits 7 – 0: Patch Version) |
| Temperature    | 1001 | N/A         | Current temperature.<br>(degrees C)                                                                                                                                                       |
| Memory Level   | 1002 | N/A         | Current memory usage.<br>(percentage)                                                                                                                                                     |
| Voltage Status | 1003 | N/A         | Current status of sensor voltages.<br>(0: Normal; 1: Error)                                                                                                                               |
| Voltage Errors | 1004 | N/A         | Count of detected voltage errors.                                                                                                                                                         |
| Camera Status  | 1200 | Camera ID   | Current camera status.<br>(0:Normal; 1:Error)                                                                                                                                             |
| Camera Errors  | 1201 | Camera ID   | Count of detected camera errors.                                                                                                                                                          |
| Camera Losses  | 1202 | Camera ID   | Count of lost camera frames.                                                                                                                                                              |
| Sync Status    | 1203 | N/A         | Current status of synchronization<br>communication between the sensor and the<br>FireSync Master.<br>(0:Normal; 1:Error)                                                                  |
| Sync Errors    | 1204 | N/A         | Count of detected synchronization<br>communication errors.                                                                                                                                |
| Data Status    | 1300 | N/A         | Current status of data communication<br>between the sensor and the FireSync Server.<br>(0:Normal; 1:Error)                                                                                |
| Data Errors    | 1301 | N/A         | Count of detected data communication<br>errors.                                                                                                                                           |
| Data Losses    | 1302 | N/A         | Count of detected data communication<br>losses.                                                                                                                                           |

## Server Indicators

The following table describes the indicators that are transmitted for each server. Unless otherwise specified, all logging intervals begin at server power-up.

| Name                           | ID   | Instance ID            | Description                                                                                                                                                                      |
|--------------------------------|------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App Version                    | 2000 | N/A                    | Server application version.<br>(Bits 63 – 32: Reserved<br>Bits 31 – 24: Major Version<br>Bits 23 – 16: Minor Version<br>Bits 15 – 8: Build Version<br>Bits 7 – 0: Patch Version) |
| Temperature                    | 2001 | 1: Master              | Current temperature of the components.<br>(degrees C)                                                                                                                            |
| Encoder Frequency              | 2002 | N/A                    | Current input encoder frequency detected by the system.                                                                                                                          |
| Encoder Value                  | 2003 | N/A                    | Current input encoder value.                                                                                                                                                     |
| Encoder Index                  | 2004 | N/A                    | Current input encoder index value.                                                                                                                                               |
| Digital I/O States             | 2005 | N/A                    | Current states digital I/Os (0 – inactive, 1 – active)<br>Bit 0: Input 0 State<br>Bit 1: Input 1 State<br>Bit 2: Input 2 State<br>Bit 3: Input 3 State                           |
| System Time                    | 2006 | N/A                    | Current FireSync network time                                                                                                                                                    |
| System Restarts                | 2200 | N/A                    | Count of system restarts due to error.                                                                                                                                           |
| System State                   | 2201 | N/A                    | Current system state.<br>(see FireSync Host Protocol Reference)                                                                                                                  |
| Server Uptime                  | 2202 | N/A                    | Uptime of the server<br>( $\mu$ s)                                                                                                                                               |
| Last Start Time                | 2203 | N/A                    | Target time sent in the last start command                                                                                                                                       |
| Communication Interface Status | 2300 | 0: Sensor<br>1: Client | Status of server communication interfaces.<br>(0:Normal; 1:Error)                                                                                                                |
| Data Losses                    | 2301 | N/A                    | Count of data frames dropped by server.                                                                                                                                          |
| Sensor Communication Status    | 2302 | Sensor S/N             | Status of control communication between server and sensor.<br>(0:Normal; 1:Missing; 2:Error)                                                                                     |
| Sensor Data Losses             | 2303 | Sensor S/N             | Count of data frames dropped by sensor.                                                                                                                                          |
| Sensor Restarts                | 2304 | Sensor S/N             | Count of sensor restarts due to error.                                                                                                                                           |
| Sensor State                   | 2305 | Sensor S/N             | State of the sensor.<br>(1: Error<br>2: Ready<br>3: Running<br>4: Paused<br>5: Boot Failure)                                                                                     |

|                     |      |             |                                                                                                            |
|---------------------|------|-------------|------------------------------------------------------------------------------------------------------------|
| Client Data Rate    | 2306 | N/A         | Data rate (in kB/s) of output to client                                                                    |
| Startup Latency     | 2307 | N/A         | Latency of the previous start command; measured between command arrival and reply transmission (µs)        |
| Sensor Link Speed   | 2308 | Sensor S/N  | Sensor link speed<br>100: 100 mbps<br>1000: 1 gbps                                                         |
| Network Health      | 2400 | N/A         | Network health<br>0: No Error<br>1: Error detected                                                         |
| Station Health      | 2401 | Station S/N | Station health<br>0: No Error<br>1: Error detected                                                         |
| Master Health       | 2402 | Master S/N  | Master health<br>0: No Error<br>1: Error detected                                                          |
| Sensor Health       | 2403 | Sensor S/N  | Sensor health<br>0: No Error<br>1: Error detected                                                          |
| Link Speed Mismatch | 2404 | N/A         | Whether or not a mismatch in link speeds of sensors has been detected:<br>0: No error<br>1: Error detected |
| Start Expire Count  | 2405 | N/A         | Number of start commands that failed due to low target time                                                |
| CPU Usage           | 7002 | N/A         | Current CPU usage                                                                                          |
| CPU Usage Min       | 7003 | N/A         | Minimum CPU usage (reset every 30 seconds)                                                                 |
| CPU Usage Max       | 7004 | N/A         | Maximum CPU usage (reset every 30 seconds)                                                                 |

## Event Indicators

The following table describes the indicators that are transmitted on the event data channel. Unless otherwise specified, all logging intervals begin at master power-up.

Some of the information overlaps with those from the server. The event data channel is intended for transmitting indicators at higher frequency (100Hz).

| Name               | ID   | Instance ID | Description                                                                                                                                                                      |
|--------------------|------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App Version        | 3000 | N/A         | Master application version.<br>(Bits 63 – 32: Reserved<br>Bits 31 – 24: Major Version<br>Bits 23 – 16: Minor Version<br>Bits 15 – 8: Build Version<br>Bits 7 – 0: Patch Version) |
| Encoder Value      | 3003 | N/A         | Current input encoder value.                                                                                                                                                     |
| Encoder Index      | 3004 | N/A         | Current input encoder index value.                                                                                                                                               |
| Digital I/O States | 3005 | N/A         | Current states digital I/Os (0 – inactive, 1 – active)<br>Bit 0: Input 0 State<br>Bit 1: Input 1 State<br>Bit 2: Input 2 State<br>Bit 3: Input 3 State                           |