AI-7160

Ringing Signal Generator

Programmers Guide

Direct Control

Advent Instruments Inc.

Release 1.3a

Printed in Canada

Advent Instruments Inc.
111 - 1515 Broadway Street
Port Coquitlam, BC, V3C6M2
Canada

Internet: techsupport@adventinstruments.com
sales@adventinstruments.com

Web Site: http://www.adventinstruments.com

Telephone: (604) 944-4298
Fax: (604) 944-7488

Contents

1. Introduction 1

2. Communication Protocol 1

2.1 The Basics 1
Physical Interface 1
Data Layer 1
2.2 Device Properties 2
2.3 Commands 3
2.4 Data Types 4
Integer 4
Hexadecimal 4
Fixed Point 5
ASCII String 5
2.5 The GET Command 6
2.6 The SET Command 6
2.7 The DO Command 7
2.8 The TAG Command 8

3. Property Reference 9

3.1 Overview 9
3.2 Ringing Generator Properties 9
3.3 Configuration Properties 13
3.4 Off Hook Detector Properties 15
3.5 Measurement Properties 17
3.6 Digital Input/Output Properties 20
3.7 Analog Input/Output Properties 22
3.8 Waveform Capture Properties 24
3.9 Miscellaneous Properties 28
3.10 System Properties 28

4. Asynchronous Messages 32

4.1 Overview 32
4.2 Power Up 33
4.3 System Errors 33
4.4 System Faults 34
4.5 Binary Waveform Captures 35

Appendix A: Firmware Revisions 37

Appendix B: Properties by Number 39

Appendix C: Command Errors 43

Appendix D: Support 47

1. Introduction

This document describes how to control and use the AI-7160 Ringing Generator via its RS-232 serial port. While the supplied Windows based software with its graphical user interface is useful for configuring the instrument, in automated testing applications direct control is usually a favored approach. Any program that is able to access a host's (PC) serial port will be able to control and use the AI-7160.

The remainder of this document is divided into the following sections, each covering an aspect of controlling the instrument.

- **2. Communication Protocol**
  Describes the fundamental aspects of sending commands and receiving responses.
- **3. Property Reference**
  Lists all of the properties that can be accessed by sending commands. These properties manage all aspects of the AI-7160's operation.
- **4. Asynchronous Messages**
  Provides information on how and when the AI-7160 may send a message back to the host outside of the normal command then response protocol.

2. Communication Protocol

2.1 The Basics

Physical Interface

The AI-7160's rear panel DE-9 serial port connector operates at a fixed baud rate of 115,200. The serial line format is fixed at 8 data bits, 1 stop bit, and no parity. No hardware flow control (CTS/RTS) is used. The 9 pin connector is configured as a DCE (Data Communications Equipment) and can connect directly with a host PC (DTE). The following table lists the signals and pins used by the 9 pin connector:

Pin	Direction	Description
2 (RXD)	To Host	Asynchronous serial data sent from the AI-7160 to the host. The least significant bit of each byte is sent first.
3 (TXD)	To AI-7160	Asynchronous serial data sent from the host to the AI-7160. The least significant bit of each byte is sent first.
5 (GND)	n/a	Signal ground.
7 (RTS)	To AI-7160	The AI-7160 ignores the state of this signal.
8 (CTS)	To Host	Indicates to the host it may send data to the AI-7160. This signal is not used and always asserted by the AI-7160.

Note, pins 1, 4, 6, and 9 have no connection within the AI-7160.

Data Layer

All data bytes sent to or coming from the AI-7160 consists of printable ASCII characters. A simple terminal program can be used to send command lines and view the corresponding response lines.

The host (PC) and AI-7160 operate in a master/slave relationship. The host sends a command line and only then does the AI-7160 return a response line. The AI-7160 will not send any data unless a command line is sent to it first, with a few limited exceptions. These exceptions are used for error and status reporting and described in more detail in section 4. Asynchronous Messages

The rules for sending command lines are as follows:

- All command lines are terminated with a carriage return character <CR> 0x0d.
- If more than one command is contained within a command line, each command must be separated by a single colon character ':'.
- The maximum length of a command line 512 bytes, including the terminating <CR>.
- Upon reception of character code 26 (CTRL-Z) the AI-7160's clears its command buffer ignoring all prior characters sent.
- Upon reception of character code 8 (CTRL-H / backspace) the AI-7160 removes the prior character from its command buffer.
- No command lines are to be sent until at least one character from the response line of a prior command line has been received.

Every command line sent generates a response line from the AI-7160. The rules for the response lines are as follows:

- All response lines start with the '$' character.
- All response lines end with the <CR> 0x0d character.
- If the command line consisted of more than one command then the response to each command is separated by a colon character ':'.
- Sending an empty command line (only a <CR>), returns an empty response line consisting of '$' followed by <CR>.
- The maximum response line length is 512 bytes including the terminating <CR>. If a command line generates a response longer than this limit the response line is terminated after 512 characters and an error response is generated.

2.2 Device Properties

The AI-7160 can be viewed as a collection of properties, each one representing a separate aspect of the instrument. Changing a setting consists of sending a command to change or modify the value of a property. Likewise, reading a measurement consists of sending a command to return the value of a property.

Logically the properties are grouped into families of related functions. Groups of properties exist for controlling the AI-7160's AC & DC voltage source, its terminal configurations, its measurement results, and other functions.

Every property is assigned a fixed and unique positive integer value. This property identification number is used by the commands as each command specifies which property it wishes to access.

The relationship between commands and properties is similar to verbs and nouns. The commands act on properties just as verbs act on nouns. It is important to note that not all commands can be used with all properties. Some properties do not support certain commands.

A complete listing of all the properties is provided in section 3. Property Reference.

2.3 Commands

In general, commands always act on a single device property. This is by either reading or writing to some aspect of a property. The AI-7160 supports four different commands. Each one is represented by a different character and is described as follows:

Command	Character	Description
Get	?	Return one or more settings from a property specified by its ID number. Example: get the RMS level of the ringing generator command: ?25 response: $50
Set	>	Modify one or more settings of the specified property with the specified operator. Operators can be used to assign new values, increment settings or decrement settings, among others. Example: increment the ringing frequency by 5 Hz. command: >21+=5 response: $*OK
Do	#	Perform an operation on a property that modifies one or more of its settings. This command is similar to the 'Set' command except it does not support operators and returns the result of its operation. Example: return the voltage (DC) and current (DC) measurements command: #34(4,13) response: $-48.0259,0.00058
Tag	@	Marks a command line with an identification number and optionally a checksum. This is the only command that does not act on a property. Rather it is used as an optional aid for synchronizing response lines to command lines in situations where the serial connection may be unreliable.

The details and usage of every command is described in more detail in the following sections:

- 2.5 The GET Command
- 2.6 The SET Command
- 2.7 The DO Command
- 2.8 The TAG Command

It is important to note that commands sent to the AI-7160 must not contain any extra white space such as space and tab characters. They cause the return of an error response. Anytime a command cannot be understood or contains invalid data an error response is returned. For information on the format of an error response, see: Appendix C: Command Errors.

2.4 Data Types

All commands either send some data or expect a response containing data. This data is either numeric or character string based. Specific rules are used for the encoding and decoding of the data fields.

The following data types may be used when sending commands and are returned by the AI-7160's responses.

Integer

This data type expresses a positive or negative integer numeric value in the range of +/- 2,147,483,647. It is composed of an optional minus sign followed by up to 10 digits consisting of 0 to 9. No other characters can be used.

```
[-] (0-9)
```

Examples:

123	Valid
-09	Valid
3.14	Invalid - not an integer
5,600	Invalid - contains a comma
5 600	Invalid - contains a space
3000000000	Invalid - exceeds maximum of 2147483647

Hexadecimal

This data type can represent any 32 bit value in the range of 0x0000 0000 to 0xFFFF FFFF. It begins with a lower case 'x' character followed by up to 8 hexadecimal digits. The hexadecimal digits must be one of 0 to 9, A to F, or a to f.

```
X ( 0-9 | A-F | a-f )
```

Examples:

xAF	Valid
x000c34fe	Valid
x 56	Invalid - no space characters allowed
x123456789	Invalid - more than 8 digits
x-81	Invalid - negative sign not allowed
D345	Invalid - missing 'x' prefix
Xe3	Invalid – ‘x’ must be lower case

Fixed Point

This data type represents numeric values in the range of +/-32767 to a precision of approximately +/- 0.00002. Fixed point values are formatted as follows:

```
[-] [ (0-9) ] [ . [0-9] ]
```

Examples:

3.1415	Valid
-45	Valid
-.567	Valid
- 13.4	Invalid - space character after minus sign not allowed
123456.768	Invalid - exceeds 32767
-0,679	Invalid - must use '.' as decimal point.

ASCII String

This data type represents a sequence of ASCII characters. All strings must begin with the single quotation mark character '. All of the characters in the string must be printable with an ASCII character code greater than or equal to 32 and must not include either : ,) % (colon, comma, right bracket, percent). An escape sequence may be used to include a character that fails the previous rule. The escape sequence is a '%' character followed by the 2 digit hexadecimal representation of the character code.

```
' ( <printable chars> | <escaped chars> )
```

Escaped characters:

```
% (0-9 | A-F) (0-9 | A-F)
```

Examples:

'hello	Valid
'%48%69 there	Valid - represents 'Hi there'
'A, or B	Invalid - comma must be escaped
'Ctrl-C is %3	Invalid - escaped characters must use 2 hex digits
'Include %3c	Invalid - escaped hex digits cannot be lower case.

2.5 The GET Command

The GET '?' command returns one or more values from a property. The number of values returned and their data type depends on the property specified. The format of the GET command is:

```
? <property ID number>
```

Examples:

Send	Response	Description
?30	$0	Return off-hook state.
?21	$26.99912	Return ringing generator frequency in Hz.
?48	$0,1	Return the rear panel BNC output settings: Mode value is 0 (off) Gain value is 1
?33	$111.11472,50,3,10,0	Return measurement parameters: Current integration time: 111.11472 ms Minimum integration time: 50 ms Minimum number of integration cycles: 3 Averaging length: 10 Current measurement range: 0 (high)

2.6 The SET Command

The SET '>' command modifies a property's value be either assigning it a new value or modifying it by an operator and value. The response to a set command is always *OK. The format of the SET command is:

```
> <property ID number> <operator> <value>
```

The possible operators are:

```
=       Assign value to property
+=      Add value to property
-=      Subtract value from property
&=      Bit-wise AND value with the property
|=      Bit-wise OR value with the property
^=      Bit-wise XOR value with the property
~=      Bit-wise AND NOT value with the property
```

Note, depending on the values data type not all operators may be supported. For strings, only the assign and add (append) operator can be used. Also not all properties may support all operators. If a property does not support the specified operator it will return an error response.

Examples:

Send	Response	Description
>25=85.6	$*OK	Set the ringing generator AC RMS level to 85.6 Vrms.
>21+=6.5	$*OK	Increase ringing frequency by 6.5 Hz.
>46|=x8	$*OK	Set bit 3 of the terminal configuration property to ensure output polarity is reversed.
>48~=x8	$*OK	Clear bit 3 of the terminal configuration property to ensure output polarity is not reversed.
>48^=x8	$*OK	Toggle bit 3 of the terminal configuration property which controls the output polarity.

2.7 The DO Command

The DO '#' command passes one or more values to a property causing it to perform an action. The response from the property is a list of one or more values. In many cases it can be considered as a combination of the SET & GET commands but lacking the different operators supported by the SET command. The format for the DO command is:

```
# <property ID number> ( <value> [ , <value> ] )
```

Examples:

Send	Response	Description
#33(4,1)	$1	Set the current measurement mode to its low range. The response returns the new current measurement range setting.
#34(4,5)	$0,-48.02545	Returns the last AC and DC voltage measurements.
#39(2)	$2	Set the rear panel digital output 'A' mode to mirror the ringing generator state.
#42(1,2)	$1,2,0	Configure the rear panel digital input 'A' to start the ringing generator on a rising edge, turn off on a falling edge. The response returns the two settings along with the state of the input pin.
#48(1,3)	$3,1	Configure the rear panel BNC output to indicate the measured voltage. The response returns the new mode and the current gain setting.

2.8 The TAG Command

The TAG '@' command does not act on a property but rather is used to mark a command line with an integer value and optionally a checksum. Its purpose is to ensure that the response lines returned by the AI-7160 can be synchronized to the command lines sent and verified for correctness by using a simple checksum. The format of the command is:

```
@ <command line ID> [ , <checksum> ]
```

The command line ID number can be any integer or hex value. The same value is echoed back in the response line. In addition, an optional command line checksum value may follow the ID number. This is an integer or hex value from 0 to 255 that represents the sum of all byte values (modulo 256) from the start of the command line up to (but not including) the tag command character '@'. If a checksum value is supplied then this value is compared against a computed checksum on the received command line. If they do not match an error response is returned.

The response to the TAG command is always two values consisting of the sent command line ID and a checksum that represents the sum of all byte values (modulo 256) from the start of the response line up to (but not including) the response to the TAG command.

```
<command line ID> , <response line checksum>
```

Examples:

Send	Response	Description
?25:@123	$83.4:123,43	Get the ringing generator level and use a command line ID of 123. The response indicates a level of 83.4 Vrms with the same command line ID of 123 and a response line checksum of 43. The value 43 is the modulo 256 sum of all response characters up to and including the colon, but not including the TAG command response.

3. Property Reference

3.1 Overview

All of the properties representing the capabilities of the AI-7160 are described in this section. Accessing them via the GET, SET, DO commands allows a host (PC) to control the operation of the AI-7160 along with returning measurement and configuration information.

Each property is represented by a unique identification number and this number must be used when sending a command. Note that not all properties support using all commands. Accessing a property with a command it does not support returns an error response.

The properties are organized into groups based on their function. The details of each property group below are described in the following sections.

- 3.2 Ringing Generator Properties
- 3.3 Configuration Properties
- 3.4 Off Hook Detector Properties
- 3.5 Measurement Properties
- 3.6 Digital Input/Output Properties
- 3.7 Analog Input/Output Properties
- 3.8 Waveform Capture Properties
- 3.9 Miscellaneous Properties
- 3.10 System Properties

For a listing of each property by its identification number, see Appendix B: Properties by Number.

3.2 Ringing Generator Properties

These properties control the high voltage AC and DC signal generator of the AI-7160. This includes the frequency, AC level, DC offset voltage, wave shape, starting phase angle, and ending phase angle settings.

All of these properties support the GET and SET commands, but not the DO command.

21	**Ring Generator Frequency** In units of Hz between the limits of 13 and 70. Default value is 22 Hz.
GET:	Returns the ring frequency.
Fixed	Frequency in Hz.
SET:	Sets the ring frequency.
Fixed	Frequency in Hz.

Example:

Send	Response	Description
>21=68	$*OK	Set frequency to 68 Hz.

22	**Ring Generator DC Voltage** In units of Volts between the limits of -200 and +200. Default is -48 volts.
GET:	Returns the DC voltage.
Fixed	In Volts.
SET:	Sets the DC voltage.
Fixed	In Volts.

Example:

Send	Response	Description
>22=30	$*OK	Set dc voltage to +30.

23	**Ring Generator Wave Shape Type** Determines the wave-shape of the ringing signal. Can be one of the following types: 0 = Sine (default) 1 = Square 2 = Trapezoidal (Ramping over 25% of the period) 3 = Trapezoidal (Ramping over 50% of the period) 4 = Trapezoidal (Ramping over 75% of the period) 5 = Triangle
GET:	Returns the wave shape type.
Int	Wave shape type.
SET:	Sets the wave shape type.
Int	Wave shape type.

Example:

Send	Response	Description
>23=3	$*OK	Set the wave shape to trapezoidal.

24	**Ring Generator Peak AC Level** In units of Volts between the limits of -233 and +233. Changing either the wave shape setting or the AC RMS level causes this setting to change in order to reflect a different wave shapes crest factor and/or a different AC RMS level. Default value is 70.7.
GET:	Returns the peak AC level.
Fixed	In Volts.
SET:	Sets the peak AC level.
Fixed	In Volts

Example:

Send	Response	Description
>23=0:>25=80:?24	$*OK:*OK:113.1372	Set the wave shape to sine and the RMS level to 80, and then return the peak voltage setting.

25	**Ring Generator RMS AC Level** In units of Volts between the limits of 0 and 160. Changing the peak AC level setting causes this setting to change. The default value is 50.
GET:	Returns the AC RMS level.
Fixed	In Volts RMS
SET:	Sets the AC RMS level.
Fixed	In Volts RMS

Example:

Send	Response	Description
>25-=10:?25	$*OK:70	Reduce the AC RMS level by 10 volts and then return the current setting.

26	**Ring Generator State** Turns on or off the ringing generator and returns the current state of the generator. Default state is 0 (off). Note: Turning on the ring generator always forces the current measurement range to its high range (100 mA).
GET:	Returns the ring generator state and warning flags.
Int	Ring generator state: 0 = off 1 = active - generating a waveform with specified shape, frequency, and level. 2 = pending off - will be turned off when the waveform phase meets the turn off criteria. 3 = muted - generator is active, but output is muted. This state is entered if off-hook detection is configured to mute the output.
Int	Ring generator warning flags: bit 0: if set the output is potentially clipped (AC and DC settings produce voltage peaks exceeding max). bit 1: if set the voltage supply for the ringing generator is off. bit 2: if set the voltage supply is operating in a low voltage mode in order to reduce power dissipation. Note: If clipping is detected the flag (bit 0) will remain set for 1 second after clipping is no longer detected.
SET:	Sets the ring generator state.

Int	Ring generator state: 0 = turn off - if currently active or muted, the generator will enter the pending off state until the turn off criteria is reached. 1 = turn on - enables the generator at the specified shape, frequency, and level.

Example:

Send	Response	Description
>26=1	$*OK	Turn on the ringing generator.

27	**Ring Generator Turn Off Mode** Controls how the ring generator will be turned off. The valid settings are as follows: 0 = Immediately when the command is issued (default). 1 = Wait until the generator reaches the specified ending phase (property #29). 2 = Wait until the generator reaches either a phase angle of 180 or 360 degrees.
GET:	Returns the turn off mode.
Int	Turn off mode.
SET:	Sets the turn off mode.
Int	Turn off mode.

Example:

Send	Response	Description
>27=2	$*OK	The ringing will only turn off when the phase angle reaches 180 or 360 degrees.

28	**Ring Generator Starting Phase** Controls the initial phase angle of the waveform generator when ringing is set to active. In units of degrees between 0 and 359.9. Default is 0.
GET:	Returns the starting phase.
Fixed	Phase in degrees.
SET:	Sets the starting phase.
Fixed	Phase in degrees. Values less than zero are clamped to zero. Values greater or equal to 360 are set to zero.

Example:

Send	Response	Description
>29=90	$*OK	Set the starting phase angle to 90 degrees.

29	**Ring Generator Ending Phase** Controls the phase angle at which the generator turns off if the turn-off mode setting is set to 1. In units of degrees between 0 and 359. Default is 0.
GET:	Returns the ending phase.
Fixed	Phase in degrees.
SET:	Sets the ending phase.
Fixed	Phase in degrees. Values less than zero are clamped to zero. Values greater or equal to 360 are set to zero.

Example:

Send	Response	Description
>29=270:>27=1	$*OK:*OK	Set the generator ending phase to 270 degrees and then set the turn off mode to wait until the specified phase angle is reached.

3.3 Configuration Properties

This group of properties determines how the AI-7160's high voltage ringing generator is connected to the various front panel terminals.

All of these properties support the GET and SET commands, but not the DO command.

44	**Internal Feeding Resistance** Determines the internal feeding resistance. This is the resistance between the generator and the output terminals. Up to five resistors may be engaged in a series configuration. A fixed resistance of 200 ohms is always present and is in series with any of the other five selectable resistors. Each resistor selection is controlled by a single bit as follows. Bit 0 = if set, insert a 30 ohm resistor into the generator to output path. Bit 1 = if set, insert a 200 ohm resistor into the generator to output path. Bit 2 = if set, insert a 320 ohm resistor into the generator to output path. Bit 3 = if set, insert a 450 ohm resistor into the generator to output path. Bit 4 = if set, insert a 1050 ohm resistor into the generator to output path. The default setting value is 2. This sets a total feeding resistance of 400 ohms (200 + 200 ohms).
GET:	Returns the internal resistance selector bits.
Hex	Selector bits.
Fixed	Resistance of all resistors selected, not including the always present 200 ohms).
SET:	Sets the internal resistance selector bits.
Int	Selector bits.

Example:

Send	Response	Description
>44=x18:?44	$*OK:x18,1500	Engage 1050 and 450 ohm resistors, for a total resistance of 1500 ohms + 200 ohms.

45	**External Feed Resistance Select** Specify if an external feed resistance is to be used, or one or more of the internal feed resistances is used. Setting a non-zero value causes an externally supplied resistance to be used, while a value of zero selects the internal resistance specified by property #44. Note: When using an external resistance a fixed resistance of 200 ohms is always in the path of the generator. Any external resistance is always in addition to the fixed 200 ohms. The default value is 0 (internal feed resistance).
GET:	Return setting.
Int	0 if internal resistance used, 1 if external resistance used.
SET:	Sets external resistance selection.
Int	Pass a non-zero value to use an external feed resistance. Pass a value of zero to use the internal feed resistance

Example:

Send	Response	Description
>45=1	$*OK	Use an externally supplied feeding resistance.

46 Output Terminal Connections

Sets the configuration of the output terminals. The bits of a single integer determines the state of four internal switches as follows:

bit 0 = if set, disconnect (float) the positive output terminal. If cleared the terminal is connected to the generator.
bit 1 = if set, disconnect (float) the negative output terminal. If cleared the terminal is connected to the generator.
bit 2 = if set, short the positive and negative terminals together.
bit 3 = if set, reverse the generator connection to the plus and minus output terminals.

All other bits are not used and forced to zero.

If the ringing generator is active when changing any of the above bits the AC signal is muted for 5 ms prior to the change, and 5 ms following the change. Once the 10 ms delay has elapsed, the AC signal is restored. However, if the ringing generator is not active then changing any of the above bits causes an immediate change to the switch states.

The default value is zero (no bits are set).

GET:	Returns the terminal configuration.
Int	Switch settings indicated by bits 0 to 3.
SET:	Sets the terminal configuration.
Int	Switch settings indicated by bits 0 to 3.

Example:

Send	Response	Description
>46=x6	$*OK	Disconnect the minus output terminal from the generator and short the plus and minus output terminals together.

47 Generator Earth Ground Select

Determines if the ringing generator has its negative end connected to earth ground or is floating. Setting a non-zero value causes the generator to be grounded, while a value of zero floats it w.r.t. earth ground.

The default value is 0 (floating).

GET:	Return setting.
Int	0 if floating, 1 if earth grounded.
SET:	Sets earth ground selection.
Int	Pass a non-zero value to connect the negative end of the generator to earth ground. Pass a value of zero to float the generator.

Example:

Send	Response	Description
>47=1	$*OK	Ground reference the ringing generator output.

3.4 Off Hook Detector Properties

This group of properties controls the operation of the off-hook line state detector.

30	**Off Hook State** Returns the hook state as either: 0 = on-hook 1 = off-hook The determination of the hook state depends on whether or not the ringing generator is active. **When ringing is not active:** Off-hook state is entered when the current drawn exceeds a threshold level for a specified period of time. When the current measurement range is set to high (default), the current threshold can be set between the limits of 1 to 20 mA. When the current measurement range is set to low, the current threshold is fixed at 0.75 mA. The time period the current must exceed the threshold is adjustable between the limits of 1 to 1000 ms. The state is returned to on-hook when the current falls below 90% of the threshold value for the same specified period of time. **When ringing is active:** Off-hook state is entered when the measured external DC resistance falls below a threshold level for a specified number of ringing cycles. When the current measurement range is set to high (default), the threshold is adjustable between the limits of 0.1 to 20 kOhms. When the current measurement range is set to low, the threshold is adjustable between the limits of 0.1 to 20 Mohms. The number of ringing cycles the resistance must be below the threshold is adjustable between the limits of 1 to 100. The state is returned to on-hook when the resistance exceeds 12% above the threshold for the same number of ringing cycles. Note: Over a complete ringing cycle if the measured averaged voltage and current are less than 0.1V or 0.2mA (2uA in the low range) respectively, then the line state is deemed to be on-hook. Note: When the ringing generator is turned on or turned off the hook state detector is disabled for an adjustable period of time between 1 to 1000 ms. During this period no change to the hook state are detected. This period allows for startup and turn off voltage or current transients to settle. Caution should be observed if the power delivered by the ringing generator can exceed the rating of any connected devices for this time interval.
GET:	Returns the current hook state.
Int	Hook state (0=on-hook, 1=off-hook).

Example:

Send	Response	Description
?30	$0	Return the current hook state as on-hook.

31	**Off-Hook Action** Controls what (if any) action is performed when an off-hook condition is detected. The valid setting are: 0 = nothing 1 = Mute the ringing generator. If the line state returns to on-hook then unmute the generator. 2 = Stop the ringing generator. If the line state returns to on-hook the ring generator remains off. 3 = Stop both the ringing generator and the command sequencer. If the line state returns on-hook both remain off. The default value is 3.
GET:	Returns the off-hook action.
Int	Action to perform when off-hook is detected.
SET:	Sets the off-hook action.
Int	Action to perform when off-hook is detected.

Example:

Send	Response	Description
>31=2	$*OK	Ensure the ringing generator is turned off when an off-hook condition is detected.

32	**Off-Hook Parameters** Accesses the settings for the off-hook detector. The settings controlling the operation of the off-hook detector are as follows: **Current Threshold (mA)**: Only used when ringing is not active. The measured current must exceed this threshold for a specified time period in order to reach the off-hook state. When the current falls bellows 90% of the threshold for the same period of time, the on-hook state is reached. The valid range is from 1 to 20 mA. Default value is 10 mA. Note, if the current measurement is in its low range this setting is ignored and a fixed threshold of 0.75 mA is used instead. **Resistance Threshold (kohms or Mohms)**: Only used when ringing is active. The measured resistance must fall below this threshold for a specified number of ringing cycles in order to reach the off-hook state. When the resistance rises above 12% this threshold for the same number of ringing cycles the on-hook state is reached. The units of this parameter depend on the current measurement range. In the high range it is kohms, while in the low range it is Mohms. The valid range is from 0.1 to 20 (kohms or Mohms). Default value is 0.8 **Current Threshold Time (ms)**: Only used when ringing is not active. Time period the measured current must either exceed or fall below the specified threshold before a change in hook state is detected. The valid range is from 1 to 1000 ms. Default value is 2. **Resistance Cycle Count**: Only used when ringing is active. The number of ringing cycles the measured resistance must either exceed or fall below the specified threshold before a change in hook state is detected. The valid range is from 1 to 100 cycles. Default value is 2. **Detector Blind Time (ms)**: Time period in which the off-hook detector is disabled when the ringing generator is either started or stopped. No change in state is detected during this time period. The valid range is from 1 to 1000 ms. Default value is 50.
GET:	Return all parameters.
Fixed	Current threshold (in mA) used to determine the off-hook state when ringing is not active.
Fixed	Resistance threshold in either kOhms or Mohms (depending on current measurement range) used to determine the off-hook state when ringing is active.
Int	Time period (in ms) the current draw must exceed the threshold when ringing is not active.
Int	Number of ringing cycles the measured impedance must be below the threshold when ringing is active.
Int	Time period (in ms) the off-hook detector is disabled for when ringing is either started or stopped.
DO:	Set the value of a single parameter.
Inputs	Description
Int	Specifies which parameter to set. 1 = current threshold 2 = resistance threshold 3 = current threshold time period 4 = resistance threshold number of ringing cycles 5 = disable time during ringing start or stop.
Fixed	Value for the parameter.
Outputs	Description
Fixed	Value being used for the parameter. Note, this can be different than the input parameter value if the value is outside the acceptable limits.

Example:

Send	Response	Description
#32(1,15):#32(3,5)	$15:5	Configure the hook state detector for a current threshold of 15 mA and a time threshold of 5 ms.

3.5 Measurement Properties

The following properties control the operation of the voltage and current meters, along with returning measurement values.

33	**Measurement Parameters** Accesses the measurement parameters. There are four parameters controlling the operation of the measurements. They are: **Minimum Integration Time (ms)**: Sets the minimum amount of integration time for a single measurement. The valid range is from 50 to 1000 ms. Default is 50. **Minimum Number of Cycles**: Sets the minimum number of ringing cycles to integrate over for a single measurement. The valid range is from 1 to 100. Default is 3. **Averaging Length**: Sets the number of measurements to average over. The averaging results are calculated from a straight arithmetic average using the specified number of measurements. The valid range is from 2 to 50. Default is 10. **Current Measurement High/Low Range Select**: 0 = high current range (+/- 100 mA) 1 = low current range (+/- 1 mA) Default is 0 (high range)
GET:	Return all measurement parameters.
Fixed	Integration time (ms). Note, this value cannot be set. It is calculated as the greater of the minimum integration time parameter or the ringing cycle time multiplied by the minimum number of ringing cycles parameter.
Fixed	Minimum integration time (ms).
Int	Minimum number of ringing cycles to integrate over.
Int	Number of integration cycles to average over.
Int	Current measurement channel range mode (0=high, 1=low).
DO:	
Inputs	Description
Int	Sets which parameter to change. 1 = Set minimum integration time (ms). 2 = Set minimum number of ringing cycle to integrate over. 3 = Set averaging length (2 to 50). 4 = Set high (0) or low (1) current measurement range.
Fixed	New value for the measurement parameter.
Outputs	Description
Fixed	Value applied to the measurement parameter.

Example:

Send	Response	Description
#33(4,1)	$1	Set the current measurement meter to its low range (+/-1 mA).

34 **35** **36**	Measurement Readings - Group A, Group B, Group C All three properties operate in the same manner. They are used to return any of the 28 available measurement readings. Up to 7 measurements can be returned at a time. Which of the 28 measurements returned are specified with the 'DO' command by passing a list of measurement IDs. Each type of measurement is identified by its ID value as listed below: 0 = Last voltage sampled (V) 1 = Minimum voltage sampled (V) 2 = Maximum voltage sampled (V) 3 = Integrated RMS voltage (Vrms) 4 = Integrated DC voltage (V) 5 = Integrated AC voltage (Vrms) 6 = Average of integrated RMS voltage (Vrms) 7 = Average of integrated DC voltage (V) 8 = Average of integrated AC voltage (Vrms) 9 = Last current sampled (mA or uA) 10 = Minimum current sampled (mA or uA) 11 = Maximum current sampled (mA or uA) 12 = Integrated RMS current (mArms or uArms) 13 = Integrated DC current (mA or uA) 14 = Integrated AC current (mArms or uArms) 15 = Average of integrated RMS current (mArms or uArms) 16 = Average of integrated DC current (mA or uA) 17 = Average of integrated AC current (mArms or uArms) 18 = Integrated DC resistance (kohms or Mohms) 19 = Average of integrated DC resistance (kohms or Mohms) 20 = Integrated AC impedance (kohms or Mohms) 21 = Phase (number of degrees the current lags the voltage by) (+/-180) 22 = Average of integrated AC impedance (kohms or Mohms) 23 = Average of phase (number of degrees the current lags the voltage by) (+/-180) 24 = Voltage measurement status flags (see register #37 for details). 25 = Current measurement status flags (see register #37 for details). 26 = Resistance measurement status flags (see register #37 for details). 27 = Impedance measurement status flags (see register #37 for details). 28 = Phase measurement status flags (see register #37 for details). Note: When the current range is set to high the readings are returned in units of mA and the resistance/impedance in units of kohms. When set to the low range the units change to uA and Mohms respectively. Note: To measure the resistance or impedance, at least 0.2 mA must be measured in the high current range and 2 uA in the low current range. Note: The phase measurement requires a measured voltage of at least 1Vrms and a current draw of at least either 1mA or 10uA depending the range setting.
GET:	
Fixed (multiple)	Returns the measurement readings specified by the IDs passed in the last 'DO' command issued.
SET:	N/A
DO:	
Inputs	Description
Int (multiple)	List of measurement IDs. Update to 7 IDs can be specified.
Outputs	Description
Fixed (multiple)	List of measurement readings that correspond to the IDs number passed.

Example:

Send	Response	Description
#34(18,20):#35(24,25)	$1000,1000:0,0	Set property #34 to return the DC resistance and AC impedance measurements and property #35 to return the voltage and current measurement flags.
?34	$1000,1000	Returns the latest measurements as specified by the

		prior 'DO' command for property #34.
?35	$0,0	Returns the latest measurements as specified by the prior 'DO' command for property #35.

37	**Measurement Reset** Resets some or all of the measurement settings and results. The passed values determine what aspect of the measurement system is reset. They are specified as: 1 = Reset all measurement parameters to default and set all readings to zero. 2 = Set the voltage minimum and maximum readings to be equal to the last sampled value. 3 = Set the current minimum and maximum readings to be equal to the last sampled value. 4 = Reset averaging by ignoring any prior measurements.
DO:	Reset measurement settings and/or results.
Inputs	Description
Int (multiple)	Value indicating what part of the measurement system to reset (as listed above). Specifying a value not listed above is ignored. Multiple values can be passed.
Outputs	Description
Int (multiple)	Same as the input parameters, except if a passed value was outside the acceptable range it will be returned as zero.

Example:

Send	Response	Description
#37(2,3)	$2,3	Reset the minimum and maximum readings for the voltage and current measurements.
#37(4)	$4	Reset averaging by ignoring any prior measurements.

38	**Measurement Status** Returns status flags for the various measurements as well as the number of integration cycles remaining until a complete average is achieved. The bit meanings are the same for the voltage, current, resistance, impedance, and phase measurements. Those flags are: Bit 0 = Set if an over-range condition is detected on either the voltage or current channel. Bit 1 = Set if an over-range condition was detected on either channel since the last time this register was read. Bit 2 = Set if the measurement is stopped because of a change in the channel's range setting. Bit 3 = Set if the measurement is stopped because of insufficient voltage and/or current (applies to phase only). Bit 4 = Set if the measurement is stopped because the ringing generator is turned off (applies to phase only). Bit 5 = Set if the last resistance/impedance measurement returned was clamped to its maximum limit.
GET:	Return measurement status flags.
Hex	Voltage measurement flags.
Hex	Current measurement flags
Hex	Resistance measurement flags
Hex	Impedance measurement flags
Hex	Phase measurement flags
Int	Number of integration cycles left before a complete average over the specified number of cycles is achieved.

Example:

Send	Response	Description
?38	$x0,x0,x0,x0,x8,0	Phase measurement cannot be performed because of insufficient measured voltage and current.

3.6 Digital Input/Output Properties

The digital input/output properties control the operation of the three rear panel digital outputs and two digital inputs. The outputs are termed 'A', 'B', and 'C'. The inputs are termed 'A' and 'B'.

39	**Digital Output A** Sets the operating mode of digital output 'A'. It can be in one of the following modes: 0 = set low (default) 1 = set high 2 = tracks the state of the ringing generator (high when active, low when off).
GET:	Returns configuration mode.
Int	Mode of operation.
DO:	Change operating mode.
Inputs	Description
Int	Operating mode: 0 = force output low 1 = force output high 2 = track ringing generator state 3 = toggle output state if existing mode is either 0 (low) or 1 (high). Any other value is ignored.
Outputs	Description
Int	Returns operating mode.

Example:

Send	Response	Description
#39(2)	$2	Set output 'A' to track the ringing generator enable state.

40	**Digital Output B** Sets the operating mode of digital output 'B'. It can be in one of the following modes: 0 = set low (default) 1 = set high 2 = tracks the state of the off-hook detector (high when off-hook, low when on-hook).
GET:	Returns configuration mode.
Int	Mode of operation.
DO:	Change operating mode.
Inputs	Description
Int	Operating mode: 0 = force output low 1 = force output high 2 = track off-hook detector state 3 = toggle output state if existing mode is either 0 (low) or 1 (high). Any other value is ignored.
Outputs	Description
Int	Returns operating mode.

Example:

Send	Response	Description
#40(2)	$2	Set output 'B' to track the off-hook detector state.

41	**Digital Output C** Sets the operating mode of digital output 'C'. It can be in one of the following modes: 0 = set low (default) 1 = set high 2 = follows the state of the command sequencer (high when running, low when stopped).
GET:	Returns configuration mode.
Int	Mode of operation.
DO:	Change operating mode.
Inputs	Description
Int	Operating mode: 0 = force output low 1 = force output high 2 = track command sequencer state 3 = toggle output state if existing mode is either 0 (low) or 1 (high). Any other value is ignored.
Outputs	Description
Int	Returns operating mode.

Example:

Send	Response	Description
#41(2)	$2	Set output 'C' to track the state of the command sequencer.

42	**Digital Input A** Controls the operation of digital input 'A'. Different actions can be performed when either a rising or falling edge is detected at the input pin. The actions are: 0 = do nothing on an edge change (default). 1 = change the ringing generator state on a rising edge. 2 = change the ringing generator state on a falling edge.
GET:	Returns digital input configuration and state.
Int	Start ringing generator on (0=never, 1=rising edge, 2=falling edge).
Int	Stop ringing generator on (0=never, 1=rising edge, 2=falling edge).
Int	Logical state of the input pin. Return 1 if high, 0 if low.
DO:	
Inputs	Description
Int	Start ringing generator on (0=never, 1=rising edge, 2=falling edge).
Int	Stop ringing generator on (0=never, 1=rising edge, 2=falling edge).
Outputs	Description
Int	Start ringing generator on (0=never, 1=rising edge, 2=falling edge).
Int	Stop ringing generator on (0=never, 1=rising edge, 2=falling edge).
Int	Logical state of the input pin. Return 1 if high, 0 if low.

Example:

Send	Response	Description
#42(1,2)	$1,2,0	Configure input 'A' to start the ringing generator on a rising edge and stop it on a falling edge. The current input state is low (0).

43	**Digital Input B** Controls the operation of digital input 'B'. Different actions can be performed when either a rising or falling edge is detected at the input pin. The actions are: 0 = do nothing on edge change (default). 1 = change the command sequencer state on a rising edge. 2 = change the command sequencer state on a falling edge.
GET:	Returns digital input configuration and state.
Int	Start command sequencer on (0=never, 1=rising edge, 2=falling edge).
Int	Stop command sequencer on (0=never, 1=rising edge, 2=falling edge).
Int	Logical state of the input pin. Return 1 if high, 0 if low.
DO:	
Inputs	Description
Int	Start command sequencer on (0=never, 1=rising edge, 2=falling edge).
Int	Stop command sequencer on (0=never, 1=rising edge, 2=falling edge).
Outputs	Description
Int	Start command sequencer on (0=never, 1=rising edge, 2=falling edge).
Int	Stop command sequencer on (0=never, 1=rising edge, 2=falling edge).
Int	Logical state of the input pin. Return 1 if high, 0 if low.

Example:

Send	Response	Description
#43(1,2)	$1,2,0	Configure input 'B' to start the command sequencer on a rising edge and stop it on a falling edge. The current input state is low (0).

3.7 Analog Input/Output Properties

The analog input/output properties control the operation of the rear panel BNC input connector and output connector.

48	**Output Control** The rear panel BNC output connector can be configured to supply a voltage that represents either the ringing generator output or the measured voltage or current. Its operation is determined by a mode setting which is one of the following values: 0 = off (output voltage fixed at zero). 1 = ringing generator output (mirrors the output of the ringing generator with a scaling of 1V to 100V). 2 = ringing generator AC wave-shape (outputs the ringing wave shape with a level of 1V peak). 3 = voltage measurement (outputs the measurement with a scaling of 1V to 100V). 4 = current measurement (outputs the measurement with a scaling of 1V to 50mA or 1V to 500uA depending on the current measurement range setting). The default mode is 0 (off). An output gain adjustment can be used to modify the scaling factors for all of the output modes. By default the gain setting is unity. However it can be set above unity to increase the output voltage or less than unity to reduce the output voltage. A negative gain setting inverts the output voltage polarity. Note, the maximum BNC output voltage is +/- 3 volts. It has a source impedance of 600 ohms.
GET:	Returns the output mode and gain.
Int	Output mode
Fixed	Output gain.
DO:	Change either the output mode setting or the output gain setting.
Inputs	Description

Int	Setting to change: 1 = output mode, 2 = output gain, anything else is ignored.
Fixed	New mode or gain setting.
Outputs	Description
Int	Output mode.
Int	Output gain.

Example:

Send	Response	Description
#48(1,2):#48(2,3)	$2,1:2,3	Configure the output to mirror the generated wave shape, followed by a command to adjust the gain to 3 yielding a wave shape output of +/-3 Volts.

49	**Input Control** The rear panel BNC input can be used to supply custom ringing generator signals. Its operation is set by a mode setting with the following values: 0 = off (voltage present at BNC input is ignored). 1 = Voltage combined with ringing generator signals. The default mode is 0 (off). An input gain factor is used to set the scaling between the BNC input voltage and the ringing generator output voltage. By default the gain is 10. This means every volt present at the BNC input translates to 10 volts at the ringing generator output. The gain setting can be increased or decreased as needed. A negative gain setting inverts the effect on the ringing generator output. The maximum BNC input voltage is +/- 4 volts.
GET:	Returns the input mode, measured voltage, and input gain.
Int	Input mode
Fixed	Voltage present at the BNC input (in volts).
Fixed	Input gain adjustment.
SET:	N/A
DO:	Change either the input mode setting or the input gain setting.
Inputs	Description
Int	Setting to change: 1 = input mode, 2 = input gain, anything else is ignored.
Fixed	New mode or gain value.
Outputs	Description
Int	Input mode.
Fixed	Voltage present at the BNC input (in volts).
Fixed	Input gain adjustment.

Example:

Send	Response	Description
#49(1,1):#49(2,50)	$1,-0.096,10:1, -0.096,50	Apply the measured voltage to the ringing generator output with a gain scaling of 50V for every 1V measured. The current reading at the input connector is -0.096 volts.

3.8 Waveform Capture Properties

The waveform capture properties operate in a similar manner to an oscilloscope. When the specified trigger conditions are satisfied the voltage and current measurement samples are written into a buffer. The contents of this buffer can then be automatically transferred to the host upon completion, or a transfer can be manually initiated by sending a command. The captured data samples are transferred to the host as any asynchronous binary message. See the section 4.5 Binary Waveform Captures for more information.

50	**Capture General Settings** Configures the general settings for voltage and current waveform capture. Up to 4000 voltage and current samples can be captured and stored. Once captured the samples may be transferred to the host either automatically or manually initiated via a binary asynchronous message. Four different general settings are used control the capture process. They are: **Sample Rate:** Determines how many voltage and current samples per second are recorded. The units are in ksamples/sec. The valid settings are 1, 2 or 4. As the amount of memory allocated to the capture buffer is fixed, increasing the sample rate decreases the maximum possible duration of the capture. The default value is 4 ksamples/s. **Capture Count:** The 4000 sample memory buffer used to store the voltage and current samples can be sub-divided into multiple smaller capture buffers, each the same size. Each smaller buffer is utilized one after the other as trigger conditions are satisfied. Multiple capture buffers are useful if multiple triggers can occur in rapid succession. Each trigger condition starts a capture process in a successive buffer. Once the last buffer is filled the entire capture process is either stopped (trigger mode = single) or started again from the first buffer (trigger mode = normal). The valid range for the capture buffer count is between 1 and 10. The default value is 1 capture buffer. **Capture Depth:** Sets the recording depth of each capture buffer in units of seconds. The maximum possible duration is based on the sample rate and number of capture buffers. Applied a value of zero or negative forces the capture depth to the maximum possible duration. This setting can be used to reduce the number of samples captured to a value less than the maximum. The default value is 0.1 seconds. **Automatic Host Transfer Count:** If set to a non-zero value, upon the completion of a capture the buffer's contents are automatically transferred to the host by sending a single asynchronous binary message. Once a transfer is initiated, this setting is decremented by one if its value is positive. If the value reaches zero then no more transfers are automatically initiated. Capture buffer data transfers may be manually initiated by using property #52. The default value is 0 (automatic transfer disabled). Note: Changing any of the above settings, except 'Automatic Host Transfer Count' stops any capture in progress.
GET:	Returns the general capture settings.
Int	Sample rate in units of ksamples/s.
Int	Number of capture buffers.
Int	Automatic host transfer count.
Fixed	Depth of each capture buffer in seconds.
Fixed	Maximum allowed depth of each capture buffer in seconds.
DO:	Change any of the general capture settings.
Inputs	Description
Int	Setting to change: 1 = sample rate, 2 = capture count, 3 = capture depth (in seconds), 4 = automatic transfer count. anything else is ignored.
Fixed	New setting value.

Outputs	Description
Fixed	Value of the setting.

Example:

Send	Response	Description
#50(4,1)	$1	Set the automatic transfer count. Following a capture the buffer contents are automatically sent to the host.

51	**Capture Trigger** Controls the trigger settings for voltage and current captures. A trigger condition is required to initiate the storage of samples into the capture buffer. Five different settings are used to control the trigger. They are: **Mode**: The capture process operates in one of three different modes. They are 'Off', 'Single', and 'Normal'. 0 = **Off**: A trigger condition will not initiate a capture. Any captures in progress are terminated. 1 = **Single**: A trigger condition starts storing the voltage and current samples to a capture buffer. Once the buffer is filled the capture is complete. If more than one capture buffer is configured then any additional trigger conditions starts a capture process on the subsequent buffer. Once the last capture buffer has been filled no more captures are made. If the trigger source includes 'manual' then captures are started immediately upon this setting changing to 1. 2 = **Normal**: Operates in the same manner as 'Single', except when the last capture buffer is filled the process begins again from the first capture buffer. If only one capture buffer is specified, then captures are continually initiated when valid trigger conditions are detected. The default mode is 0 (off). **Source**: Selects the trigger sources. A trigger source is an event or condition that initiates a capture. Multiple sources can be selected at the same time. When any of the selected source conditions occurs the capture is started. Each source is assigned a bit position in the setting. The valid bit positions are: Bit 0: Manual trigger: Changing the mode to a non-zero value becomes the trigger event. Bit 1: Voltage - Measurement must cross the level setting in the specified direction (polarity). Bit 2: Current - Measurement must cross the level setting in the specified direction (polarity). Bit 3: (reserved) Bit 4: Off-hook detected. Bit 5: On-hook detected. Bit 6: (reserved) Bit 7: (reserved) Bit 8: Digital input 'A' rising edge. Bit 9: Digital input 'A' falling edge. Bit 10: Digital Input 'B' rising edge. Bit 11: Digital Input 'B' falling edge. **Position**: Specifies the position of the trigger in relation to the capture buffer. The trigger position uses units of seconds and ranges from -10 to the duration of the capture buffer. The duration of the capture buffer is specified by the Capture Depth general setting. The trigger position value can be interpreted as follows: Negative (<0): Represents the delay from when the trigger condition is detected to when the first sample is stored into the capture buffer. Use negative values when the point of interest occurs sometime after the trigger condition. Zero (0): The trigger condition represents the first sample stored into the capture buffer. Use a value of zero when the point of interest occurs immediately after or shortly after the trigger condition. Between 0 and Capture Depth setting: The trigger condition represents a sample taken within the capture buffer. This is useful when the point of interest occurs before or immediately at the trigger condition. Note: If the sample rate setting is increased or the number of capture buffers increased, the trigger position is recalculated to ensure it does not exceed the duration of the capture buffer. **Level**: Specifies a voltage or current threshold used as a trigger condition. If the voltage or current (depending on the source setting bits) crosses the trigger level setting in the direction specified by the polarity setting, then a trigger event occurs. The units for the trigger level are V or mA. Unless either of the voltage or current trigger source flags are set this setting has no effect. **Polarity**: Specifies the polarity used to cause a voltage or current trigger event. A value of zero represents a rising slope polarity while any other value represents a falling slope polarity. If the voltage or current crosses the trigger level in the direction specified by this setting, then a trigger event occurs. Unless either of the voltage or current trigger source flags are set this setting has no effect.
GET:	Returns the trigger settings.
Int	Mode (0=off, 1=single, 2=normal)
Int	Source flags (see bit flag values abo ve)

Fixed	Trigger position in seconds (-10 to Capture Depth).
Fixed	Trigger threshold for voltage and current (in units of V or mA)
Fixed	Trigger polarity for voltage and current (0=rising slope, otherwise falling slope).
DO:	Change any of the trigger settings.
Inputs	Description
Int	Setting to change: 1 = mode, 2 = source flags, 3 = position, 4 = threshold level, 5 = polarity. anything else is ignored.
Fixed	New setting value.
Outputs	Description
Int	Value of the setting.

Example:

Send	Response	Description
#51(2,x2):#51(3,0.1):#51(1,1)	$0:0.09999:1	Set trigger source to 'voltage'. Set trigger position to 0.1 seconds. Set trigger mode to 'single'.

52 Capture Status

Returns the status of the capture process or initiates a transfer of a capture buffer to the host.

Reading this property returns the number of completed captures and the status of the current or last capture. The number of captures ranges from 0 to the maximum value specified by the general setting 'Capture Count'. The status value indicates the capture state as one of the following:

0 = Idle: No capture in process.
1 = Arm: Capture buffer is being prepared to accept a trigger condition.
2 = Trigger: Waiting for a trigger condition to complete the capture.
3 = Acquire: Trigger has occurred and the capture is occurring.
4 = Complete: The capture is finished.

If more than one capture buffer is configured then when the 'Complete' state is reached the subsequent capture buffer is prepared for the trigger condition by entering the 'Arm' state. This process continues until the last capture buffer reaches the 'Complete' state.

When at least one capture buffer has reached the 'Complete' state the contents of a buffer may be transferred to the host. This is initiated by writing to the property and passing the capture buffer number. The transfer begins immediately and completes before the response to the command is returned.

Note: The transfer of the capture buffer occurs as a single asynchronous binary message.

GET:	Returns the capture count and status.
Int	Number of capture buffers in the completed state.
Int	Status of the capture process.
DO:	Initiate a transfer of the capture buffer contents to the host.
Inputs	Description
Int	Capture buffer number to transfer (from 1 to number of completed captures).
Outputs	Description
Int	Capture buffer number being transferred. Zero is returned if the specified buffer number is not valid or it is not in a complete state.

Example:

Send	Response	Description
?52	$1,4	One capture taken, current status is complete.
#52(1)	<async binary message> $1	Transfer contents of capture buffer #1 in as a binary message, followed by the command response.

3.9 Miscellaneous Properties

20	**Execute Application Operation** Performs a special operation that is only used for testing or trouble-shooting purposes. Error codes: 50: DO command was passed a string for the first parameter. 51: DO command was passed an invalid operation identifier value.
GET:	N/A
SET:	N/A
DO:	Perform specified operation.
Inputs	Description
Int	Operation identifier. See following table for list of valid identifier values.
Outputs	Description
	Various - Depends on passed operation identifier. See the following table.

Operation ID	Response	Description
100	1	Turn on the ringing generator high voltage supply. Turn on power dissipation monitoring.
101	1	Turn off the ringing generator high voltage supply.
102	1	Force the voltage supply into low mode and turn off power dissipation monitoring.
103	1	Force the voltage supply into high mode and turn off power dissipation monitoring.

3.10 System Properties

1	**Device Summary** Returns general device information.
GET:	
String	Model and name: AI-7160 Ringing Generator
String	Device serial number: in the form of 'SN15yyyy'
Hex	Device model identifier Bits 16 to 31: Product family code: 0x02 Bits 0 to 16: Product variant code: 0x01
Hex	System version code: Bits 24 to 31: major code: 0x01 Bits 16 to 23: minor code: 0x01 Bits 0 to 15: build code: (varies)
Hex	Unit identifier code - high 32 bits.
Hex	Unit identifier code - low 32 bits.
DO:	

Inputs	Description
Int	Category of device information to return: 1 = Same as for the 'GET' command. 2 = Application firmware version code (hex), and hardware version code (hex) Both values formatted in the same manner as the 'System version code' listed above. 3 = Device birth date (string) and last calibration date (string). Both strings are formatted as "y<y>-m<m>-d<d>" for example: Feb. 10, 2012 will return: 'y2012-m02-d10'

2	**Installed Options** Return information about any installed options.
GET:	
Int	Number of options installed or supported.

3	**Reset** Reset application settings or the entire device.
DO:	
Inputs	Description
Int	Type of reset to perform: 1 = restore applications settings to default values. 2 = re-boot the device following a short delay. The delay is so the command response can be returned to the PC.

4	**Read Device Parameter** Return the value of one or more device parameters. These are internal parameter used for calibration and device identification purposes. Access to these parameters is only meant for calibration or trouble shooting purposes. If the ID values passed are not integers the command fails with error 'n' where 'n' is the parameter index. If the ID value is not valid the command fails with error 'n' where 'n' represents 10 + the parameter index that is the invalid item ID.
DO:	
Inputs	Description
Int	One or more parameter ID numbers. Up to 7 item ID's can be specified.
Outputs	Description
Int	One or more parameter ID values. The number of item values returned equals the number of IDs specified. The data type of each returned depends on the item ID. Some items are returned as integer, hex, or in fixed point format.

5	**Write Device Parameter** Set the value of a device parameter. Access to these parameters is only meant for calibration or trouble shooting purposes. Note: Changing parameter values does not automatically save the new value to non-volatile memory. Note: If the parameter ID number is invalid a command error is returned.
DO:	
Inputs	Description
Int	Item ID number to set.
Int	Item value to set. Data type to use depends on the item ID.
Outputs	Description

Int	Item ID number that was written too.

6	**Execute System Operation** Perform a system related function. The number and type of the passed parameters are variable.
DO:	
Inputs	Description
	Various
Outputs	Description
	Various

7	**System Error Counts** Returns the number of system errors recorded.
GET:	
Int	Total number of errors that have been recorded.
Int	Number of those errors that have been flagged as critical.
DO:	
Inputs	Description
Int	Clear all errors if a non-zero value is passed.
Outputs	Description
	Same as for GET above.

8	**System Error Details** Obtain details on a specific error type.
GET:	Return information on the first error type that has a non-zero count.
Int	The next error type that has a non-zero count (-1 if none).
Int	Error type being reported.
Int	The bit-wise OR of the flags field for all errors recorded (of this type).
Int	The details field for the last error recorded (of this type).
Int	The number of errors of this type that have been recorded.
Int	The timestamp field for the last error recorded (of this type). Units are in 1 ms increments.
String	The message string for the last error recorded (of this type).
DO:	
Inputs	Description
Int	Error type to report details on. Pass a negative value to get the details on the first error type that has a non-zero count.
Outputs	Description
	Same as for GET above.

9	**Command Sequencer Text** Sets the command sequencer text string or returns information about the sequencer text string. Note: Do not change the sequencer commands while it is running. This can lead to undefined operation.
GET:	Returns the number of characters in the sequencer command string and the amount of free space still available.
Int	Number of characters in the sequencer command string.
Int	Amount of space left in the sequencer buffer.
SET:	Sets the sequencer command string or appends text to the existing command string. If not enough space exists in the buffer, an error code of 100 is returned.

Inputs	Description
String	'=' operator: Set the string as the new sequencer commands. '+=' operator: Append the string to the end of any existing sequencer text string.

10	**Command Sequencer Control** Controls the operation of the sequencer or returns status information about its execution.
GET:	Return sequencer status information.
Int	Current execution status: 0= stopped, 1=paused, 2=running, 3=stopped with error. Note, if an error is encountered the error status remains until either a stop or run command is issued. Note, if the sequencer runs to the end of the last command without error the position is 1 more than the sequencer string length.
Int	Current execution position: 0=not yet started, else=character position in sequencer text string.
DO:	Control sequencer execution
Inputs	Description
Int	Control value: 0 = stop, 1 = pause (only functions if currently running), 2 = run (start execution from the beginning of the command sequence string). Any other value is ignored.

11	**Command Sequencer Errors** Returns information about any command sequencer errors encountered. Note, starting execution of a sequence resets the error code to zero and clears the error message string.
GET:	
Int	Error code: 0 = none, 1 = 'get' command error 2 = 'set' command error 3 = 'do' command error else = unknown command character
String	Message string containing information about the error encountered.

4. Asynchronous Messages

4.1 Overview

Normally the AI-7160 only responds to commands sent to it. However there are a few situations in which it does send a message outside the normal command/response flow. These are termed asynchronous messages and may occur anytime, including after the reception of a command but before the transmission of the response. As such the host (PC) must be able to identify and separate the asynchronous messages from normal command responses.

It is important to note that an asynchronous message will never split apart a response message, nor a response message split apart an asynchronous message. The transmission of each type of message is atomic.

As with the normal response messages, asynchronous messages follow a near identical structure. The only difference is that they start with an exclamation mark character '!' instead of the dollar sign '$'. As with the response messages an asynchronous message is terminated by a <CR> character and consist only of ASCII printable characters.

The contents of an asynchronous message consist of 1 to 7 fields, each using the same available data types and format (integer, fixed point, hexadecimal, character strings) as normal response messages.

The conditions that cause an asynchronous message are as follows:

- **Power-up or reboot:** Message is sent to the host on power-up or re-boot containing basic information about the device. This includes its name, serial, version, and other fields.
- **System errors:** When internal and recoverable errors are detected within the device, the error information is logged and an asynchronous message is sent. The message contains information about the type of error and any available details.
- **System faults:** When internal non-recoverable errors occur, the AI-7160 enters a fault mode. It outputs a message once every 3 seconds and flashes a code with the READY front panel indicator. This is a terminal error and the unit must be power cycled. No further communication is possible.

Note, a special type of asynchronous message is used to transfer the contents of waveform capture buffers to a host. These messages contain binary data and follow a different format from the text based asynchronous messages. Details of the message format is described in section: 4.5 Binary Waveform Captures

4.2 Power Up

The power-up or re-boot message is generated shortly after the AI-7160 is first turned on or has been rebooted by sending the command '#3(2)'. The message can be used by the host (PC) to identify that a power-up or reboot has occurred and provides basic information about the device. This includes its name, serial number, product identifier, system & communication protocol version, and a unique 64-bit identifier. The format of the power-up/reboot message is as follows:

```
!*PUP, <name>, <SN>, <PID>, <SysVrs>, <UIDhigh>, <UIDlow>
Where:
        <name> is the name of the device:
                'AI-7160 Ringing Generator'
        <SN> is a string containing the product's serial
                number.  Its format follows:  'SN<nnnnnn>'
                where <nnnnnn> represents 6 decimal digits
                0 to 9.
        <PID> is an integer representing the product
                identification number.  All AI-7160's will
                report the same ID number of 0x00020001.
        <SysVrs> is an integer representation of the system &
                communication version.  The format is:
                   Bits 24 to 31:  Major revision
                   Bits 16 to 23:  Minor revision
                   Bits 0 to 15:  Build revision
        <UIDhigh> is the high 32 bits of a unique 64 bit
                integer.
        <UIDlow> is the low 32 bits of a unique 64 bit
                integer.
```

4.3 System Errors

If the AI-7160 detects an internal recoverable error it will log the error and send an asynchronous message to the host (PC). The message contains details on the error encountered, including its type or class of error, specific bit flags used to indicate a sub-type of error, additional details provided by an integer, the number of errors of this type/class that have occurred, a timestamp (since power-up/reboot), and optionally a text message.

```
!*ERR, <type>, <flags>, <details>, <count>, <time>, <msg>
Where:
        <type> is an integer representing the type or class
                of error.
        <flags> integer bit flags representing the specific
                error conditions for all errors of this
                type.
```

```
<details> integer value containing additional
          information for the last error of this
          type.
<count> an integer value representing the number of
          errors that have occurred of this type.
<time> an integer value representing the time stamp
          of when the last error of this type
          occurred.  The units are in ms.
<msg> a text string optionally providing more
          information on the last error of this
          type.
```

The possible error types/classes and sub-types are defined as follows:

Type/Class	Sub-Type Bit Flag	Description
0 - System	0x0001	Under voltage detected on internal power supply.
	0x0002	(reserved)
	0x0004	Queue failure - Property change descriptor.
	0x0008	Queue failure - Property change notification.
	0x0010	Queue failure - Unable to send command response.
	0x0020	Command response exceeds buffer size.
	0x0040	Invalid property data type signature.
	0x0080	EEPROM - Checksum failure - default settings used instead.
	0x0100	EEPROM - Unable to modify - invalid write position.
	0x0200	EEPROM - Unable to modify - invalid word count specified.
	0x0400	Calibration data not loaded.

Type/Class	Sub-Type Bit Flag	Description
1 - Comm	0x0001	Loss of communication synchronization.
	0x0002	Queue failure - Unable to write.
	0x0004	Command line exceeds maximum limits.
	0x0008	Error in reception of character. possibly incorrect baud, framing, stop-bit.

Type/Class	Sub-Type Bit Flag	Description
2 - Measure	0x0001	Synchronization error with data converters.

4.4 System Faults

If the AI-7160 traps a non-recoverable internal error it enters into a fault mode. In this mode it turns off all front panel indicators except for READY. Once every three seconds the READY indicator flashes. The number of flashes indicates a fault code. Also once every three seconds it sends an asynchronous message to the host (PC). This message contains the fault code along with a text string providing more information.

Once in fault mode the AI-7160 must be power cycled for recovery.

The format of the fault message is as follows:

```
!*FLT, <fault>, <message>
Where:
          <fault> is an integer value representing the fault
                    code.
          <message> is a text string providing more information
                    on the fault cause or condition.
```

Please contact technical support if a fault code is encountered.

4.5 Binary Waveform Captures

Waveform captures containing voltage and current measurement samples are transferred to the host using a binary asynchronous message format. The structure of this message format is as follows:

```
! <ESC> <data length> <checksum> <data>
Where:
          <ESC> is the byte code 0x1B
          <data length> is a 16 bit value (sent LSB first) that
                    represents the number of bytes in the data
                    field.
          <checksum> a 32 bit value (sent LSB first) that
                    represents the 2's complement of the sum
                    of all 32 bit words in the data field (LSB
                    first). Only for the purposes of the
                    checksum calculation, if the number of
                    data bytes does not fall on a 32 bit word
                    boundary, then additional zero value bytes
                    are added in order to complete the 32 bit
                    word.
          <data> zero to a maximum of 16384 data bytes.
```

Unlike for the text based messages, there is no special terminating character (carriage return <CR>) for binary messages.

The captured voltage and current waveform samples are stored within data field of the binary message. For capture records, the data field always contains an integral number of 32-bit words. Four words are used to form a header which contains details concerning the capture buffer and the capture conditions. Any additional 32-bit words are voltage and current sample pairs. 16 bits are used to express the voltage and 16-bits for current.

The structure of this waveform capture record is defined in the following table:

Bits 24 to 31 (MSB)	Bits 16 to 24	Bits 8 to 15	Bits 0 to 7 (LSB)	Description

<table>
<tr><td>'1'</td><td>'P'</td><td>'A'</td><td>'C'</td><td>Identifier: The four ASCII characters 'CAP1' identify the binary message data payload as captured waveform samples.</td></tr>
<tr><td colspan="2">Sample Count</td><td>Sample Rate</td><td>Capture</td><td>Capture: The capture buffer number being transferred. Ranges from 1 to maximum number of capture buffers available.

Sample Rate: Sample rate in units of k-samples/second. Possible values are 1, 2, and 4.

Sample Count: Number of voltage and current samples contained in this data record.</td></tr>
<tr><td colspan="2">Post Trigger Count</td><td colspan="2">Trigger Source Flags</td><td>Trigger Source: Contains bit flags indicating which trigger condition(s) initiated the capture.

Post Trigger Count: The number of voltage and current samples captured after the trigger event.</td></tr>
<tr><td colspan="2">Status Flags</td><td colspan="2">Auto Transfer Count</td><td>Auto-Transfer Count: Contains the value of the auto-transfer count setting at the time the capture data record was sent.

Status Flags: Bit flags defined as follows:

Bit 0: Set if current measurement in low range.

(all other bits are reserved and will contain zero).</td></tr>
<tr><td colspan="2">Current (#1)</td><td colspan="2">Voltage (#1)</td><td>Voltage sample #1: 16-bit signed integer representing the voltage in units of 1/32 V.

Current sample #1: 16-bit signed integer representing the current in either units of 1/256 mA (high current range) or 1/16 uA (low current range).</td></tr>
<tr><td colspan="2">Current (#2)</td><td colspan="2">Voltage (#2)</td><td>Subsequent voltage and current sample (provided Sample Count is greater than 1).</td></tr>
<tr><td colspan="2">…</td><td colspan="2">…</td><td></td></tr>
<tr><td colspan="2">Current (#n-1)</td><td colspan="2">Voltage (#n-1)</td><td>Second to last voltage and current sample.</td></tr>
<tr><td colspan="2">Current (#n)</td><td colspan="2">Voltage (#n)</td><td>Last voltage and current sample, where 'n' is equal to Sample Count.</td></tr>
</table>

Appendix A: Firmware Revisions

Updates to the AI-7160's firmware are periodically released which may change the operation in one of the following manners:

- Addition of new properties.
- Expanded capability of existing properties.
- Perform fixes to known bugs.

Any time new properties are added or their capabilities expanded all attempts are made to preserve backwards compatibility. This permits the firmware to be upgraded anytime without concern for breaking applications that were developed with older versions of the firmware.

The AI-7160's firmware can be updated by using the Windows graphical control program.

For the most current listing of the AI-7160's firmware's revision history, please see:

http://www.adventinstruments.com/Products/AI-7160/Support/Firmware%20Revisions

Appendix B: Properties by Number

All of the available properties for controlling the AI-7160 are listed by index number in the following table.

#	Name	GET/SET/DO Command Format
1	**Device Summary**	GET → Name, SN, Product Identifier, SysVrs, UIDhigh, UIDlow DO (1) →(same as GET) DO (2) →Firmware Version, Hardware Version DO (3) →Birthdate, Last Calibration Date
2	**Installed Options**	GET →Number of options
3	**Reset/Restore**	DO (1) →Restore settings to default DO (2) →Reboot
4	**Read Parameter**	DO (ID number,...) →Value of parameter (up to 7 of them)
5	**Write Parameter**	DO (ID number, value) →ID number
6	**System Operation**	DO (OpType,…) →Depends on operation type.
7	**Error Count**	GET →Total count, Critical count DO (n) → Reset count if (n) is non-zero. Return counts.
8	**Error Details**	GET →Next type, type, flags, details, count, timestamp, message DO (n) →Get details on error type (n). Pass -1 to get first type.
9	**Sequencer Text**	GET →Number chars in sequence, Amount of space left SET →Text to assign or append.
10	**Sequencer Control**	GET →Status (0=stopped, 1=paused, 2=running, 3=error), Execution position DO (n) →0=stop, 1=pause, 2=run from beginning
11	**Sequencer Error**	GET →Error code (0=none), message string
20	**App Operation**	DO (OpType, …) → Depends on operation type.
21	**Ring Frequency**	GET/SET → Frequency in Hz
22	**Ring DC Voltage**	GET/SET →Voltage in V
23	**Ring Wave Shape**	GET/SET → Waveshape (0=sine, 1=square, 2-4=trap, 5=triangle)
24	**Ring AC Peak Level**	GET/SET → AC Peak voltage in V
25	**Ring AC RMS Level**	GET/SET →AC RMS voltage in Vrms
26	**Ringing State**	GET →state (0=off, 1=active, 2=pending off, 3=muted), flags (bit 0=clipped, bit 1=HV supply off, bit 2=low mode) SET → State (0=turn off, 1=turn on)
27	**Ring Turn Off Mode**	GET/SET →0=immediately, 1=at end phase, 2=at 180 or 360 deg.
28	**Ring Starting Phase**	GET/SET → Starting phase in deg.
29	**Ring Ending Phase**	GET/SET →Ending phase in deg.
30	**Off Hook State**	GET →0=on hook, 1=off hook
31	**Off Hook Action**	GET/SET →0=nothing, 1=mute ring, 2=stop ring 3=stop ring and sequencer

32	**Off Hook Parameters**	GET →Ithreshold, Zthreshold, Itime, Zcycles, Blind time DO (1, Ithreshold) →Ithreshold DO (2, Zthreshold) → Zthreshold DO (3, Itime) →Itime DO (4, Zcycles) → Zcycles DO (5, Blind Time) → Blind Time
33	**Measure Parameters**	GET → IntTime, MinIntTime, MinCycles, AverageCnt, Irange DO (1, MinIntTime) →MinIntTime DO (2, MinCycles) → MinCycles DO (3, AverageCnt) → AverageCnt DO (4, Irange) → Irange
34	**Measurement Readings**	GET →Readings specified by last DO command. DO (reading ID, …) →Readings specified by readingID
35	**Measurement Readings**	GET →Readings specified by last DO command. DO (reading ID, …) →Readings specified by readingID
36	**Measurement Readings**	GET →Readings specified by last DO command. DO (reading ID, …) →Readings specified by readingID
37	**Measurement Reset**	DO (1) →Restore default parameters and zero readings. DO (2) →Set voltage min/max to last reading DO (3) →Set current min/max to last reading DO (4) →Reset averaging to last reading
38	**Measurement Status**	GET →Voltage flags, Current Flags, Res Flags, Zflags, Phase flags
39	**Digital Output 'A'**	GET → Operating mode (0=low, 1=high, 2=mirror ring generator) DO (n) →Set mode (0=low, 1=high, 2=ring gen, 3=toggle)
40	**Digital Output 'B'**	GET → Operating mode (0=low, 1=high, 2=mirror hook status) DO (n) →Set mode (0=low, 1=high, 2=hook status, 3=toggle)
41	**Digital Output 'C'**	GET → Operating mode (0=low, 1=high, 2=mirror sequencer) DO (n) →Set mode (0=low, 1=high, 2=sequencer, 3=toggle)
42	**Digital Input 'A'**	GET → Start ring action, Stop ring action, Input state DO (Start ring action, stop ring action) →(return same as GET)
43	**Digital Input 'B'**	GET → Start seq. action, Stop seq. action, Input state DO (Start seq. action, stop seq. action) →(return same as GET)
44	**Internal Feed Resistance**	GET →Selector bits, resistance selected in ohms. SET →Selector bits
45	**External Resistance**	GET → 0=internal resistance, 1=external resistance SET → 0=internal resistance, else external resistance
46	**Terminal Configuration**	GET→ Switch flags SET → Switch flags
47	**Ground Generator**	GET → 0=floating, 1=grounded SET → 0=floating, else grounded
48	**BNC Output Control**	GET →Mode, Gain DO (1, Mode) → Mode, Gain DO (2, Gain) → Mode, Gain
49	**BNC Input Control**	GET → Mode, Voltage, Gain DO (1, Mode) → Mode, Voltage, Gain DO (2, Gain) → Mode, Voltage, Gain
50	**Capture General Settings**	GET →Rate, #Cap Buffers, Xfer Count, Cap Depth, Max Cap Depth DO (1, Rate) → Rate DO (2, #Cap Buffers) → #Cap Buffers DO (3, Cap Depth) → Cap Depth DO (4, Xfer Count) → Xfer Count
51	**Capture Trigger Settings**	GET → Mode, Source, Position, Level, Polarity DO (1, Mode) →Mode DO (2, Source) → Source

		DO (3, Position) → Position DO (4, Level) → Level DO (5, Polarity) → Polarity
52	**Capture Status**	GET → Captures Complete Count, Status DO (Buffer #) → Buffer #

Appendix C: Command Errors

Sending a command that the AI-7160 cannot understand causes it to return an error response instead of the normal command response. An error response can be returned under the following conditions:

- Invalid command character
- Property number or operator that does not exist
- Incorrect formatting of data types (numbers and strings)
- White space (space or tab characters) used within the command
- Illegal parameter values specified with the DO or SET commands.

When the error is detected all further processing of the command or any other commands on the same line are terminated and an error response is returned. If a prior command (on the same line) was processed without error its response is returned before the error response.

The format of the error response is as follows:

```
*ERR, <error code>, <position>, <details>
Where:
          <error code> is an integer value.
          <position> is the character position in the command
                     line where the error was detected.
          <details> is additional information dependent on the
                     error code
```

The following table lists all of the possible error codes and their meaning.

Error Code	Description:
	Note: For all of the following errors the returned <details> field is the ASCII character code of the character that caused the error, except where noted.
1	**Unknown Command:** The first character of a command is unknown. The only valid command characters are: ? > #@.
2	**Invalid Property ID:**

	The property ID number (following the command code character) is invalid. It may be because either no numeric characters were found or the numeric value is outside the valid range.
3	**Invalid Terminator:** The command is not terminated properly. All commands must end with a <CR> (end of line), or the ':' character in the case of multiple commands on the same command line.
4	**Invalid Operator:** The operator specified for the SET command is invalid. Only the following operators are possible: =, +=, -=, &=, |=, ^=, ~= Not all of the properties support all of the above listed operators.
5	**Wrong Character:** A specific character was not found at the expected location. This occurs when the brackets '( )' surrounding the value list for the DO command are not found.
6	**Invalid Data Type Identifier:** A value's data type is unknown. This occurs when integers do not start with either - or 0-9, or hexadecimal values do not start with 'x' (lower case only), or strings do not start with the apostrophe character.
7	**Too Many Values:** More values have been specified in a value list than is supported. Currently the maximum number of values in a list is 7.
8	**Missing Value:** Occurs when a value is incomplete for the following data types: Integer: No digits following the negative sign. Hexadecimal: Invalid hex chars following the 'x' character. The only valid characters are: 0-9, A-F, a-f Strings: Require two hex characters following the '%' character.
9	**Out of Range:** Caused by: 1) Too many hex chars: Hexadecimal value contains more than 8 digits. 2) The absolute value of a fixed point number exceeds or equals 32768.
10	**Non-Printable Char:** String contains a non-printable character (char code less than 32).
12	**Invalid Escape Format:** The two characters following the '%' escape character are not hexadecimal digits (0-9, or A-F). Note, lower case 'a-f' characters cannot be used for escaping characters.
13	**Input Values Mismatch:** The number of values and the data type of those values passed to the SET or DO command does not match that required by the property. This error

	code is also issued if automatic type conversion fails (ie, from an integer greater than 32767 to a fixed point value). Automatic type conversion is only attempted between integer and fixed point values.
13	**Command Not Supported**: The specified command (GET, SET, or DO) is not supported by the specified property.
14	**Command Failed:** The command was formatted properly but the command failed and returned an error code. This is usually due a passed command value being outside valid limits. The error code value is returned in the <details> field of the error response.
15	**Frame Checksum Failed:** The checksum of the received frame does not match the checksum value specified by the tag '@' command. The <details> field of the error response contains the calculated checksum of what was received. Note: The checksum is from the start of the frame up to but not including the tag command character '@'.

Appendix D: Support

For assistance regarding any of the topics discussed in this document, or any general questions, please contact us in one of the following methods.

- Email: Technical Questions:
  techsupport@adventinstruments.com
  Sales Inquiries:
  sales@adventinstruments.com

- In North America:

  Tel: (604) 944-4298
  Fax: (604) 944-7488
  Mail: Advent Instruments Inc.
  111 - 1515 Broadway Street
  Port Coquitlam, BC, V3C6M2
  Canada

- In Asia:

  Tel: (852) 8108-1338
  Fax: (852) 2900-9338
  Mail: Advent Instruments (Asia) Ltd.
  Unit No. 7, 9/F, Shatin Galleria
  18 - 24 Shan Mei Street
  Fotan, Shatin, N.T.
  Hong Kong