icListen Command & Control Telemetry

Feb 26, 2019

Ocean Sonics Ltd.
Hill House, 11 Lornevale Road,
Great Village, NS, B0M 1L0 Canada
Phone: +1 902 655 3000
www.OceanSonics.com

Table of Contents

1 Introduction

This document applies to the following **icListen** models:

icListen LF v2.7 and below
icListen HF v2.2 (Release 32) and below
icListen AF v2.2 (Release 32) and below

If your **icListen** model is not included in the above list, or is a higher firmware version than those listed above, please contact Ocean Sonics for the latest revision of this document.

This document details the telemetry format for communicating with an **icListen** unit over the command and control channel. All **icListen** models support this communication channel. **icListen HF** and **AF** also support streaming channels. For more information on streaming channel telemetry, please see the *icListen Streaming Telemetry* document.

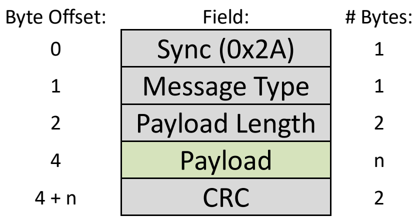
Communications are done in a master-slave format with the PC (master) issuing commands (generally via the **Lucy** software) and the **icListen** (slave) responding with the requested data. **icListen** will not send any data through command and control channels until it has been requested, or if certain actions have taken place on the web interface of Ethernet models.

2 Overview of Messages

This is a brief overview of the commands accepted by the **icListen** command and control channel. Unless specified otherwise, all fields in a message are little endian, and in binary format. Messages sent to **icListen LF** cannot exceed 2048 bytes, and messages sent to **icListen HF** and **icListen AF** can be up to 65541 bytes. These limits include the message framing (sync, message type, length, and CRC bytes).

2.1 Basic Message Structure

All messages sent to and from **icListen**'s command and control channel share a common structure:



Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type	1
2	Payload Length	2
4	Payload	n
4 + n	CRC	2

Figure 2-1: Basic Command Structure

Sync: The first character in all messages must be the sync character. This character is an ASCII '*' character (0x2A).
Message Type: This byte identifies the actual type of message being transmitted.
Payload Length: This is the number of bytes in the payload of the message. A payload length of 0 is valid.
Payload: The payload contains any necessary parameters that must accompany a command. The payload will vary in length, based on the type of message being transmitted.
CRC: This is a cyclic redundancy check value, which is used to protect against data corruption within the message during transmission. See *Cyclic Redundancy Check* for more detail.

2.2 icListen Commands

Command	***Code***	***Description***
Set Time	‘A’ (0x41)	Sets icListen’s internal time
Set Baud Rate	‘B’ (0x42)	Sets icListen’s Baud Rate
Collect Data	‘C’ (0x43)	Requests data on the channels indicated
Job Setup	‘D’ (0x44)	Updates icListen’s configuration
Enquire Device	‘E’ (0x45)	Requests device type, status and firmware version
Enquire Job Setup	‘F’ (0x46)	Requests the current configuration details
Read File	‘H’ (0x48)	Retrieves file data from icListen
Write File	‘I’ (0x49)	Sends file data to icListen
Delete File	‘J’ (0x4A)	Instructs icListen to delete a file
Directory Listing	‘K’ (0x4B)	Requests a list of directories and files
File Size	‘L’ (0x4C)	Requests the size of a specific file
Rename File	‘M’ (0x4D)	Instructs icListen to rename a file
Copy File	‘N’ (0x4E)	Instructs icListen to make a copy of a file
Delete Folder	‘O’ (0x4F)	Instructs icListen to delete a folder and its contents
Enquire File System (LF)	'P' (0x50)	Requests list of pending file system commands from icListen LF
Stop Data Streams	‘Q’ (0x51)	Stops/Disconnects all data streams
Enquire Triggers	'R' (0x52)	Requests list of active triggers
Setup Trigger	‘S’ (0x53)	Sets up an epoch trigger
Remove Trigger	‘T’ (0x54)	Removes an existing epoch trigger
Clear Triggers	‘U’ (0x55)	Deletes all of the epoch triggers from icListen
Reset Password	‘V’ (0x56)	Resets the user password for icListen
Reset	‘Y’ (0x59)	Resets the icListen unit
Enquire Battery	‘Z’ (0x5A)	Enquires the battery status
Enquire Time Sync	‘[’ (0x5B)	Enquires the time sync status
File System Command	‘\’ (0x5c)	Performs various actions on icListen’s file system (icListen AF/HF)

2.3 Command Compatibility

The command set used between all **icListen** models is shared, but not all commands will receive a response on all models and firmware versions. The following table shows the compatibility of each **icListen** model with each of the commands.

Command	***icListen LF***	***icListen HF***	***icListen AF***
Set Time	All versions	All versions	All versions
Set Baud Rate	All versions	v2.2 (Release 31) and up	v2.2 (Release 31) and up
Collect Data	All versions	All versions	All versions
Job Setup	All versions	All versions	All versions
Enquire Device	All versions	All versions	All versions
Enquire Job Setup	All versions	All versions	All versions
Read File	All versions	None	None
Write File	All versions	None	None
Delete File	All versions	None	None
Directory Listing	All versions	None	None
File Size	All versions	None	None
Rename File	All versions	None	None
Copy File	All versions	None	None
Delete Folder	All versions	None	None
Enquire File System	All versions	None	None
Stop Data Streams	None	All versions	All versions
Enquire Triggers	All versions	v1.4 (Release 15) and up	All versions
Setup Trigger	All versions	v1.4 (Release 15) and up	All versions
Remove Trigger	All versions	v1.4 (Release 15) and up	All versions
Clear Triggers	All versions	v1.4 (Release 15) and up	All versions
Reset Password	None	All versions	All versions
Reset	v2.5 and up	All versions	All versions
Enquire Battery	None	v1.6 (Release 17) and up	All versions
Enquire Time Sync	v2.7 and up	v2.1 (Release 28) and up	v2.1 (Release 28) and up
File System Command	None	v2.2 (Release 31) and up	v2.2 (Release 31) and up

2.4 Differences in Messages between Firmware Versions/Models

As new updates to the firmware are performed, some commands and responses may occasionally vary between **icListen** firmware versions, and models.

The command descriptions will describe the behaviour of each command for the most recent release of the firmware.

Any differences between firmware versions will be noted in the sections for the affected commands.

3 Configuration

Commands detailed in this section are used either for retrieving or changing **icListen** configuration settings.

The commands detailed in this section are:

Enquire Device
Enquire Job Setup
Job Setup
Set Time
Set Baud Rate
Reset Password
Reset
Enquire Battery
Enquire Time Sync

For details on the *Enquire Triggers* command, see the *Epoch Commands* section.

3.1 Enquire Device

The purpose of the enquire device command is to retrieve the type, status, and firmware version of the connected device. This command is supported by the entire **icListen** family of products.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x45)	1
2	Payload Length (0)	2
4	CRC	2

Figure 3-1: Enquire Device Command

No payload is required for the Enquire device command. Example code for setting up an enquire message can be found in *Appendix A*.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x45)	1
2	Payload Length (33+n)	2
4	Serial Number	2
6	Firmware Version	8
14	Build Date	18
32	Device Type	1
33	Status	1
34	Reserved	2
36	Enquire Version	1
37	Version Specific Payload	n
37 + n	CRC	2

Figure 3-2: Enquire Device Response

Serial Number: This is the serial number for this device. It is unique to each unit of a given device type.
Firmware Version: This is a null terminated ASCII string which shows the firmware version of the unit (ex: “v1.0.01”).
Build Date: This is a null terminated string indicating the date on which the firmware was built. (ex: “25 Apr 2013 15:00”).
Device Type: This is a code representing the type of device that this unit is. See Appendix C for a list of valid device type codes.
Status: This code indicates the operating status of **icListen**. See Appendix C for a list of valid codes.
Reserved: These bytes hold no meaning and should be ignored.

Enquire Version: This is the version of the enquire response. This value is a code which dictates what is contained in the Version Specific Payload Field.

Version Specific Payload: This is a variable field, which changes based on the enquire version. For any given device, as the enquire version number increases, new values are added to this part of the payload, and the existing fields from the previous version are still maintained.

3.1.1 icListen LF Version Specific Enquire Payloads

icListen LF v2.6 and below use version 1 payloads, and, v2.7 uses payload version 2.

Byte Offset:	Field:	Version	# Bytes:
37	Full Scale mV	1-2	2
39	Hydrophone Sensitivity	1-2	2
41	Max HW Gain	1-2	1
42	Link Type	1-2	1
43	Temperature	2	2
45	Humidity	2	2
47	Sync Status	2	1
48	System Time Status	2	1

Figure 3-3: LF Enquire Payload

Full Scale mV: This is the value in millivolts that a full scale count in the waveform data corresponds to.

Hydrophone Sensitivity: This is the sensitivity of the hydrophone in dB re 1µPa.

Max HW Gain: This is the maximum hardware gain that is applied to data. Any additional gain past this value will be applied digitally in software.

Link Type: This code represents the type of communications link used. See Appendix C for valid codes.

Temperature: This is internal temperature of **icListen** in tenths of °C (123 = 12.3 °C).

Humidity: This is the internal relative humidity of **icListen** in tenths of % (321 = 32.1 % rh).

Sync Status: This code indicates the status of PPS synchronization. Valid codes are:

0 = Sync input/output disabled

1 = Sync output enabled

2 = Sync output enabled (with PPS encoded time)

3 = Sync input enabled (falling edge), no PPS detected

4 = Sync input enabled (falling edge), syncing to detected PPS

5 = Sync input enabled (falling edge), synced to PPS

6 = Sync input enabled (rising edge), no PPS detected

7 = Sync input enabled (rising edge), syncing to detected PPS

8 = Sync input enabled (rising edge), synced to PPS

System Time Status: This code indicates how the system time was set. Valid codes are:

0 = Time value has not been set

1 = Time was manually set

2 = Time set from RTC at start-up

3 = Time was set from file system at start-up

4 = Time was set from NTP

5 = Time was set from GPS

6 = Time was set from PTP

128 -> 137 = Time set from PPS encoded time. Subtracting 128 from the value will give the time status code of the PPS source (ex: 129 – 128 = 1 = Time set manually).

3.1.2 icListen AF/HF Version Specific Enquire Payloads

icListen HF/AF v1.1 (Release 12) and below use version 1 payloads, v1.2 to v1.5 (Release 13 to 16) use version 2 payloads, v1.6 to v2.0 (Release 17 to 27) use version 3 payloads, v2.1 (Release 28 to 30) use version 4 payloads, and v2.2 (Release 31) uses version 5 payloads.

Byte Offset:	Field:	Version	# Bytes:
37	Hydrophone Sensitivity	1-5	2
39	Temperature	3-5	2
41	Humidity	3-5	2
43	Battery Status	3-5	1
44	Battery Percent	3-5	1
45	HW Rev	3-5	1
46	Link Type	3-5	1
47	Release	4-5	8
55	Sync Status	4-5	1
56	System Time Status	4-5	1
57	Model	5	16

Figure 3-4: AF/HF Enquire Payload

Hydrophone Sensitivity: This is the sensitivity of the hydrophone in dB re 1µPa.
Temperature: This is internal temperature of **icListen** in tenths of °C (123 = 12.3 °C).
Humidity: This is the internal relative humidity of **icListen** in tenths of % (321 = 32.1 % rh).
Battery Status: This code indicates the charging state of the internal battery. Valid codes are:

0 = No Data Available
1 = Battery is Charging
2 = Battery is Discharging
3 = Battery is not Charging or Discharging

Battery Percent: This is the estimated charge % of the internal battery. A value of 0xFF indicates that this data is unavailable.
HW Rev: This is the hardware revision of this **icListen**. Device behaviour may vary based on this.
Link Type: This code represents the type of communications link used. See Appendix C for valid codes.
Release: This is a null terminated ASCII string describing the firmware release.
Sync Status: This code indicates is the status of PPS synchronization. Valid codes are:

0 = Sync input/output disabled
1 = Sync output enabled
2 = Sync output enabled (with PPS encoded time)
3 = Sync input enabled (falling edge), no PPS detected
4 = Sync input enabled (falling edge), syncing to detected PPS
5 = Sync input enabled (falling edge), synced to PPS
6 = Sync input enabled (rising edge), no PPS detected
7 = Sync input enabled (rising edge), syncing to detected PPS
8 = Sync input enabled (rising edge), synced to PPS

Model: This is a null terminated ASCII string describing the model of icListen. (ex: RB9-ETH)

System Time Status: This code indicates how the system time was set. Valid codes are:

0 = Time value has not been set

1 = Time was manually set

2 = Time set from RTC at start-up

3 = Time was set from file system at start-up

4 = Time was set from NTP

5 = Time was set from GPS

6 = Time was set from PTP

128 -> 137 = Time set from PPS encoded time. Subtracting 128 from the value will give the time status code of the PPS source (ex: 129 – 128 = 1 = Time set manually).

3.2 Job Setup

The Job Setup command is used to change the current setup of the instrument. This command is supported by the entire **icListen** family of products.

In the event of a successful setup, this command also results in the update of the internal job setup configuration file. This allows **icListen** to boot into the last known configuration. This file is located at "/ICLISTEN/JOBSETUP.CFG" on **icListen LF** units, and at "/home/icListen/Config/JOBSETUP.CFG" on **icListen HF** and **icListen AF** units.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x44)	1
2	Payload Length (2+n)	2
4	Type	2
6	Type Specific Payload	n
6 + n	CRC	2

Figure 3-5: Job Setup Message

Type: This code sets the setup type of the following type specific payload.
Type Specific Payload: This payload varies based on the value of the Type field.

The response to this message is in the same form as the command.
If there was any error in the settings in the command, the entire message will be rejected, and the setup type in the response will indicate that there was an error.

If the setup was successful, the response will indicate the same type of setup that was sent to the instrument, and the response will contain the new setup values which the unit is now operating with. In some cases the new settings may vary slightly from the requested settings, but should always be equivalent to those sent. More details on the specific setup types can be found in the sections below.

Firmware Differences:
-HF versions before Release 20 stored the setup as "/home/icListen/Config/JOBCONFIG.CFG"

3.2.1 Tag Based Setup (code 20)

This setup type is recognized by **icListen HF** as of release 23, and is the only setup type supported by **icListen AF**. Though **icListen HF** still recognizes some older setup types for backwards compatibility, it is recommended that this setup type be used for **icListen HF** as well. This setup type is planned to be the setup type used by any future **icListen** models as well.

This setup consists of a list of tags and values for settings which are to be configured. The list can include all configurable options in **icListen**, or if desired can also contain only those parameters that must be updated. For example, this allows a user to adjust the waveform settings, without changing the spectrum settings, or adjust the waveform log length without changing any of the other waveform data collection settings.

Byte Offset:	Field:	# Bytes:
0	Number of Tags	2
2	Tag 1	2
4	Tag 1 Value Length	2
6	Value 1	n
6 + n	Tag 2	2
8 + n	Tag 2 Value Length	2
10 + n	Value 2	m
10 + n + m	…	

Figure 3-6: Tag Based Setup (20) Payload

Number of Tags: This is the number of setup tags that will follow in the message.
Tag #: This is the tag which the following value corresponds to.
Tag # Value Length: This is the length in bytes of the value for this tag. Each tag has a minimum length which it must be. The length can be extended however, and the extra bytes will simply be ignored. This is done to allow a driver to adjust their alignment for improved efficiency if desired.
Value #: This is the value that is set for the given tag. The meaning, length, and limits depend on the given tag.

Tag descriptions and lengths can be seen in the following sub-section.

3.2.1.1 Setup Tag Values

The following list describes the meaning of each setup tag code. Values in this table all have a minimum length of 4 bytes. Limits on the values depend on the **icListen** model, which can be found in Appendix D.

0 (Auxiliary Pin Function):

This is the function of the auxiliary pin on icListen. Valid settings are:
0 = Sync to falling edge of PPS
1 = Produce falling edge PPS
2 = Sync to rising edge of PPS
3 = Reserved
4 = Ignore aux pin state
5 - 31 = Reserved
32 = Sync to falling edge of PPS, and wake from standby when pin goes low
33 = Produce falling edge PPS, and wake from standby when pin goes low
34 = Sync to rising edge of PPS, and wake from standby when pin goes low
35 = Reserved
36 = Wake from standby when pin goes low

1 (Log Start Time):

This is the UNIX time at which both waveform and spectrum logging will begin. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC. Setting this to a negative value will cause logging to begin immediately when data is available.

2 (Spectrum Sample Rate):

This is the sample rate in Hz that spectrum data is sampled at.

3 (Spectrum Reference Level):

This is the reference level of spectrum data results in dB relative to 1V.

4 (Points Per FFT):

This is the number of data points used per FFT calculation.

5 (Samples Between FFT's):

This is the number of new samples gathered per FFT calculation.

6 (FFT Processing Type):

This is the type of processing performed on FFT's, and its result depends on the FFT's accumulated and in some cases the FFT weighting factor. Valid types are:
Mean (4): Avg = sum[Value(0)…Value(# Accumulated)]/(# Accumulated)
Peak (5): Maximum signal strength at each frequency over (# Accumulated) FFT's.
Exponential Moving Average(EMA) (6): Avg(t) = [(Weight-1) * Avg(t-1) + Value(t)]/Weight

7 (FFT's Accumulated):

This is the number of FFT's accumulated per spectrum data result.

8 (FFT Weighting Factor):

This is the weighting of new FFT data sets used in the EMA FFT processing type.

9 (Spectrum Logging Mode):

This is the spectrum data logging mode. Valid settings are: disabled (0), active (1)), triggered (2).

10 (Spectrum Log File Length):

This is the length of spectrum log files in minutes.

11 (Waveform Duty Cycle Time):

This is the duration of the duty cycle applied to waveform data logging, in minutes.

12 (Waveform Duty Cycle Active Time):

This is the active portion of the waveform data logging duty cycle, in minutes.

13 (Waveform Duty Cycle Ignore Time Changes):

If this field is set to 1, the duty cycle will not be interrupted when the system time is updated. If it is set to 0, the duty cycle will be adjusted so that the duty cycle will operate by stopping and starting at the minutes that it otherwise would have.

Example:

Logging begins at 1:00:00.

Duty Cycle is set to 3 minutes active, 2 minutes idle.

At 1:01:30 on the instrument, the time is re-synched to 1:02:00 (+30 seconds)

Ignore Time Changes = 0 (default):

This active part of the cycle will end at 1:03:00, and the next active portion will begin at 1:05:00. This preserves the time of day that logging begins/ends at.

Ignore Time Changes = 1:

This active part of the cycle will now end at 1:03:30, and the next active portion will begin at 1:05:30. This preserves the current cycle length.

14 (Waveform Sample Rate):

This is the sample rate in Hz that waveform data is sampled at.

15 (Waveform Data Bit Depth):

This is the number of bits per sample returned in waveform data.

16 (Waveform Data Gain):

This is the gain applied to waveform data in dB.

17 (Waveform Data Endianness):

This controls the endianness of waveform data. Valid settings are big endian (0), or little endian (1). If logging is not disabled, this must be set to little endian. Most computers are little endian machines as well, so if unsure, set to little endian.

18 (Waveform Data Logging Mode):

This is the waveform data logging mode. Valid settings are: disabled (0), active (1), triggered (2).

19 (Waveform Log File Length):

This is the length of waveform log files in minutes.

20 (Auxiliary Pin Messages):

This setting is a bitmask used to configure what messages are used on the auxiliary pin. Enabled messages are sent when configured to output a sync pulse, and read when configured to accept a sync pulse. If the sync input/output is disabled, no messages will be transmitted or received on the auxiliary pin. Currently the only available message is the system time (UNIX time), which can be used to precisely synchronise the time of day between icListen devices. Due to the slow data rate of this link (1 bit per second), it may take up to 2 minutes to synchronise the system time.

Valid settings:

0 = No messages sent/received

1 = Time of data sent with sync out/received with sync in

3.2.2 icListen LF Setup (code 12)

This setup type is recognized only by **icListen LF**. It is used to set all settings for the instrument in one message.

Byte Offset:	Field:	# Bytes:
0	Gain	2
2	Sample Rate	4
6	Mode of Operation	2
8	Reserved	4
12	FFT Processing Type	2
14	Processing Parameters	4 or 8

Figure 3-7: icListen LF Setup (12) Payload

Gain: This is the gain applied to both waveform and spectrum data, set in dB. The maximum gain that may be applied to the waveform data is the maximum hardware gain of that instrument. Any gain set above this will be applied digitally to the spectrum data.

Sample Rate: This is sample rate at which the data is collected for both waveform and spectrum data, in Hz.

Mode of Operation: This is a code which determines how data is logged/transmitted in/from the instrument. For a full list of these codes see Appendix C.

Reserved: This field is reserved and should be set to 0.

FFT Processing Type: This is a code which determines how FFT data is processed by the instrument. The type of processing set in the command may be altered in the response to an equivalent type for certain processing types.

Processing Parameters: This is a set of parameters which varies depending on the FFT processing type selected. It is either 2 or 4, 16 bit parameters. The processing parameters may be altered to equivalent parameters in the response for certain processing types.

3.2.2.1 FFT Processing Types & Parameters

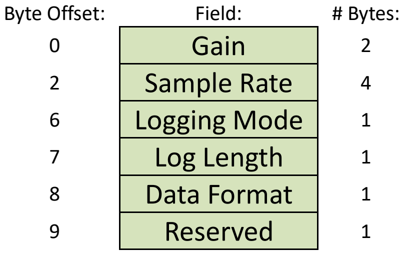
There are 3 types of FFT processing available for icListen: Mean Average, Peak Detect, and Exponential Moving Average (EMA). All of these processing methods can be combined with a sliding window overlap. If the processing type sent in a setup message does not combine sliding window with another type of processing, the response may be modified from the command. The modified response will be an equivalent setup (methods sent without sliding window will be converted to sliding window types with no overlap, and sliding window without other processing will be converted to mean of 1 with the desired overlap).

The sliding window parameters will also potentially modified in the returned value. These parameters will be modified so that the second overlap parameter is 1024 (which is the number of points used for each FFT calculation). So if the set values were 1 and 2, the response will contain 512 and 1024.

Processing Type	#Params	Details
0 – Mean Average *modified to type 4 in response	2	Avg = sum[Value(0)…Value(P1)]/P1 *P1 cannot be 0 *P2 is reserved (set to 0)
1 – Sliding Window *modified to type 4 in response	2	Overlap = 1 – P1/P2 *P2 must not be greater than P1
2 – Peak Detect *modified to type 5 in response	2	P1 = Number of points that peak is found over *If P1 is 0, the peak values are held until scanned, rather than using a set number of FFT's *P2 is reserved (set to 0)
3 –EMA *modified to type 6 in response	2	Avg(t) = [(P1-1) * Avg(t-1) + Value(t)]/P1 P2 is the number of FFT's between reported values *P1 and P2 cannot be 0
4 – Mean with Sliding Window	4	Avg = sum[Value(0)…Value(P1)]/P1 Overlap = 1 – P3/P4 *P1 cannot be 0 *P2 is reserved (set to 0) *P3 must not be greater than P4
5 – Peak with Sliding Window	4	P1 = Number of points that peak is found over Overlap = 1 – P3/P4 *If P1 is 0, the peak values are held until scanned, rather than using a set number of FFT's *P2 is reserved (set to 0) *P3 must not be greater than P4
6 – EMA with Sliding Window	4	Avg(t) = [(P1-1) * Avg(t-1) + Value(t)]/P1 Overlap = 1 – P3/P4 *P1 and P2 cannot be 0 *P3 must not be greater than P4

3.2.3 Waveform Setup (code 14)

This setup type is recognized only by **icListen HF**. It has been superseded by the tag based setup type (20), but for backwards compatibility is still recognized by the instrument, and is returned with the default setup enquire response. It is used to set most waveform data related setup parameters.



Byte Offset:	Field:	# Bytes:
0	Gain	2
2	Sample Rate	4
6	Logging Mode	1
7	Log Length	1
8	Data Format	1
9	Reserved	1

Figure 3-8: Waveform Setup (14) Payload

Gain: This is the gain applied to waveform data, set in dB. Gain in **icListen HF** is applied digitally.
Sample Rate: This is sample rate at which the data is collected, in samples per second.
Logging Mode: This code determines how data is logged internally. Logging may be disabled (0), active (1), or active when triggered (2).
Log Length: This is the maximum log file length, set in minutes.
Data Format: This is a code which determines the endianness and bit-depth of the waveform data. Valid settings are:

2 = 16 bit Big Endian
3 = 24 bit Big Endian
4 = 32 bit Big Endian (24 bit data packed into 32 bits)
130 = 16 bit Little Endian
131 = 24 bit Little Endian
132 = 32 bit Little Endian (24 bit data packed into 32 bits)

Reserved: This Field is reserved, and should be set to 0.

3.2.4 Spectrum Setup (code 15)

This setup type is recognized only by **icListen HF**. It has been superseded by the tag based setup type (20), but for backwards compatibility is still recognized by the instrument, and is returned with the default setup enquire response. It is used to set most spectrum data related setup parameters.

Byte Offset:	Field:	# Bytes:
0	Reference Level	2
2	Sample Rate	4
6	Points Per FFT	2
8	FFT Processing Type	2
10	Samples Between FFT’s	2
12	FFT’s Accumulated	2
14	FFT Weighting Factor	2
16	Logging Mode	1
17	Log Length	1

Figure 3-9: Spectrum Setup (15) Payload

Reference Level: This is the reference level that spectrum data results are relative to in dB relative to 1V.
Sample Rate: This is sample rate at which the data is collected.
Points Per FFT: This is the number of data points used for each FFT calculation.
FFT Processing Type: This is a code which determines how FFT data is processed by the instrument. The following processing types are valid:

> **4 (Mean Average):** Avg = sum[Value(0)…Value(Accumulations)]/ Accumulations
> **5 (Peak Detect):** Max value at each frequency taken over the set number of “Accumulations”
> **6 (Exponential Moving Average):** Avg(t) = [(Weight-1) * Avg(t-1) + Value(t)]/Weight

Samples Between FFT’s: This is the number of new data points gathered between each FFT calculation.
FFT’s Accumulated: The number of FFT data sets accumulated together for each spectrum data result.
FFT Weighting Factor: This value is used only when the processing type is set to exponential moving average. It determines the weight on new FFT data sets with respect to the rest of the data in the average by the following formula.
Logging Mode: This code determines how data is logged internally. Logging may be disabled (0), or active (1), or active when triggered (2).
Log Length: This is the maximum log file length, set in minutes.

3.2.5 Log Start Time Setup (code 18)

This setup type is recognized only by **icListen HF**. It has been superseded by the tag based setup type (20), but for backwards compatibility is still recognized by the instrument, and is returned with the default setup enquire response. It is used to set the time that internal logging will begin at.

Byte Offset:	Field:	# Bytes:
0	Reserved	2
2	Start Time	4

Figure 3-10: Log Start Time Setup (18) Payload

Reserved: This field is reserved and must be set to 0.
Start Time: This is the UNIX time value at which logging sill start for both waveform and FFT logs. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC. If a negative value is set, logging will begin as soon as data becomes available, without any delay.

3.2.6 Duty Cycle Setup (code 19)

This setup type is recognized only by **icListen HF**. It has been superseded by the tag based setup type (20), but for backwards compatibility is still recognized by the instrument, and is returned with the default setup enquire response. It is used to set the duty cycle applied to waveform data logging.

Byte Offset:	Field:	# Bytes:
0	Active Time	2
2	Idle Time	2
4	Ignore Time Changes	1
5	Reserved	1

Figure 3-11: Duty Cycle Setup (19) Payload

Active Time: This is the active portion of the duty cycle in minutes.

Idle Time: This is the idle portion of the duty cycle in minutes.

Ignore Time Changes: If this field is set to 1, the duty cycle will not be interrupted when the system time is updated. If it is set to 0, the duty cycle will be adjusted so that the duty cycle will continue to operate by stopping and starting at the minutes that it otherwise would have. Example:

> Logging begins at 1:00:00.
>
> Duty Cycle is set to 3 minutes active, 2 minute idle.
>
> At 1:01:30 on the instrument, the time is re-synched to 1:02:00 (+30 seconds)
>
> **Ignore Time Changes** = 0 (default):
>
> This active part of the cycle will end at 1:03:00, and the next active portion will begin at 1:05:00. This preserves the time of day that logging begins/ends at.
>
> **Ignore Time Changes** = 1:
>
> This active part of the cycle will now end at 1:03:30, and the next active portion will begin at 1:05:30. This preserves the current cycle length.

Reserved: This field is reserved and must be set to 0.

3.2.7 Old Duty Cycle Setup (code 16)

This setup type is recognized only by **icListen HF** v1.4 (release 15) and older. It has been superseded by the tag based setup type (20), and is not supported by new firmware releases. It was responsible for setting the start time for waveform data logging, and the duty cycle used by waveform data logging.

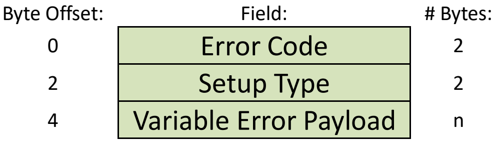
Byte Offset:	Field:	# Bytes:
0	Reserved	2
2	Active Duration	2
4	Idle Duration	2
6	Start Delay	4

Figure 3-12: Old Duty Cycle Setup (16) Payload

Reserved: This field is reserved and must be set to 0.
Active Duration: This is the active portion of the duty cycle in minutes.
Idle Duration: This is the idle portion of the duty cycle in minutes.
Start Delay: This is the number of seconds after the current time to wait before logging waveform data.

3.2.8 Error Setup (code 65535)

The error setup type is sent when a setup command was rejected due to some of the settings being invalid. This setup is generated as a response to an invalid setup, and will not be accepted as a valid setup type by **icListen**. This setup response can come from **icListen LF**, **icListen HF**, and **icListen AF**.



Byte Offset:	Field:	# Bytes:
0	Error Code	2
2	Setup Type	2
4	Variable Error Payload	n

Figure 3-13: Error Setup (65535) Payload

Error Code: This code indicates what error was encountered when trying to apply a setup. For a list of valid error codes, see the following sub-section.
Setup Type: This is the setup type that this error is a response to.
Variable Error Payload: For most setup types this payload field is empty. This field will hold data for the following error codes:

> **65523 (Unrecognized tag):** The payload will contain a single 16bit field holding the unrecognized tag value.
> **65524 (Invalid tag length):** The payload will contain a single 16bit field holding the tag value that has a bad length sent with it.
> **65525 (Check Tags):** The payload will contain a 16bit value indicating the number of tag errors that follow, and a list of tag (16bit)/tag error (16bit) pairs. The list will contain all tags sent with the setup, and their error status. Additional tags may also be contained in the response if the error was due to a conflict with those settings. For a list of tag error codes, see the following sub-section.

3.2.8.1 Setup Error Codes

The following table describes the error codes that may be returned when a setup fails:

Code	Description
0	Unrecognized setup type
1 – 15	If the error code is between 1 and 15, these bits may be used as a bit field to describe specific invalid settings: Bit 0: Invalid Gain Bit 1: Invalid Sample Rate Bit 2: Invalid Mode of Operation(LF), Invalid Waveform Data Format(HF) Bit 3: Invalid FFT Type or Parameters Bit 4: Invalid Logging Setup
65523	Unrecognized setup tag
65524	Invalid tag length
65525	Check tags for more details. See the following table for tag error codes.
65526	Invalid time
65527	Invalid (non -0) reserved bytes
65528	Invalid duty cycle settings
65529	Communications to DSP failed
65530	Memory allocation failed
65531	Message payload too small for setup
65532	Reserved – this code is not in use
65533	Setup is incompatible with one or more triggers, as it could generate too much data to log (**icListen LF** only)
65534	Too much data to log (reduce waveform sample rate, or change spectrum processing parameters to remedy this.)
65535	Unit is not ready to accept setups

The following table describes tag errors returned with the “Check Tags” error:

Code	Description
0	No error (success).
1	Value is not valid (generic error).
2	Value is in conflict with another setup parameter (the other parameter will also be marked with this error code).
3	Value was not checked for validity. This may occur if a value which this parameter depends on was not valid.
4	Value is too high.
5	Value is too low.

3.3 Enquire Job Setup

The Enquire Job Setup command is used to read the current setup and status from the instrument. This command is supported by the entire **icListen** family of products.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x46)	1
2	Payload Length (2)	2
4	Type	2
6	CRC	2

Figure 3-14: Enquire Job Setup Command

Type: This is the setup type which is to be enquired. This is an optional field. If this field is empty or is an unrecognized type, the default setup type (or types) for the instrument will be returned.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x46)	1
2	Payload Length (3+n)	2
4	Status	1
5	Type	2
7	Type Specific Payload	n
7 + n	CRC	2

Figure 3-15: Enquire Job Setup Response

Status: This code indicates the operating status of **icListen**. See Appendix C for a list of valid codes.
Type: This code indicates the setup type which fills the following type specific payload.
Type Specific Payload: This payload varies based on the value of the Type field. See the sub-sections of the "Job Setup" command section for more details on the type specific payloads.

If a specific setup type is requested, and that type of setup is recognized, a single setup message of that type will be sent in response. If no specific type is requested, or the requested setup type is not recognized by the instrument, the default setup response will be sent. In the case of icListen HF, the default case consists of 4 separate setup responses. Here are the default setup response types for each **icListen** model:

icListen LF: Type 12 (icListen LF setup)
icListen HF: Types 14 (waveform), 15 (spectrum), 18 (log start time), 19 (duty cycle)
icListen AF: Type 20 (tag based setup)

Firmware Differences:

- HF versions 1.4 (Release 15) and below will return types 14 (waveform), 15 (spectrum), 16 (Old Duty Cycle)

- HF versions 1.8 (Release 22) and below will not recognize the type field and will return the defaults

3.4 Set Time

The set time command is used to update or enquire the system time of **icListen**.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x41)	1
2	Payload Length (4)	2
4	Time	4
8	CRC	2

Figure 3-16: Set Time Message

Time: This is the UNIX time value that **icListen**'s system time will be updated to. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC. A time value of 0 or less may be sent to enquire the system time from the instrument without updating it. The set time command cannot be used to set the system time to a date before Jan 1, 2010.

The response is the same as the message, with the time field now containing the system time of the instrument at the time the response was sent. If the set time command fails, a time value of -1 will be sent in the response.

When icListen boots up, it will acquire its initial time differently based on the icListen model. For **icListen LF** units, the time will be set to the time of the last logged file, or to January 1st 1980 if no files have been logged. **icListen HF** and **icListen AF** make use of an on-board RTC to retrieve the time on start-up, which is reset to January 1, 2010 if the RTC has lost power due to fully drained internal batteries.

Firmware Differences:

-LF versions before 2.6, and HF versions before 1.7 (Release 19) will return the current time in the unit if the set time command fails instead of -1

-LF versions before 2.7 and HF versions before 2.0 (Release 23) will allow the time of day to be set before Jan 1, 2010

3.5 Set Baud Rate

The set baud rate command is used to update the baud rate used by serial **icListen** devices.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x42)	1
2	Payload Length (4)	2
4	Baud Rate	4
8	CRC	2

Figure 3-17: Set Baud Rate Command

Baud Rate: This is baud rate that the unit is being set to communicate at.

If a baud rate is sent, no response will be sent for this message. For **icListen HF/AF**, if the payload is left empty (payload length of 0, not a baud rate of 0), the active baud rate will be returned. This information may be useful when using SLIP to communicate with serial devices.

After the baud rate has been updated, **icListen** must receive another command within 5 seconds (preferably not the "Set Baud Rate" command again), at the new baud rate, in order for the new baud rate to be accepted. If no other command is received within those 5 seconds, the baud rate will revert to the previous baud rate. This is to ensure that the unit is capable of communicating with the user at the new rate before saving the configuration.

The baud rate used by **icListen** at boot time is found in the configuration files. If this file is not found, the device will use the default rate for your instrument.

The following baud rates will be accepted by **icListen**:

icListen LF: 1200, 2400, 4800, 9600, 19200, 115200, 25600, 1250000

icListen HF/AF: 19200, 115200, 230400, 460800, 921600

Firmware Differences:
-LF versions before 2.5.02 do not support baud rates below 19200

3.6 Reset Password

On **icListen** models that make use of a password, this command is used to reset the password to a blank value. Passwords may be used by **icListen HF** and **icListen AF** for SSH and on the web interface.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x56)	1
2	Payload Length (0)	2
4	CRC	2

Figure 3-18: Reset Password Message

The response to this message is the same as the command, and neither message contains a payload.

Firmware Differences:
-HF versions before 2.0 (Release 23) will reset the SSH password, but not the webserver password. All passwords are reset as of release 23

3.7 Reset

This command will cause **icListen** to reset. In the case of **icListen HF** and **icListen AF**, this command will shut down **icListen**, and only restart it if external power is supplied to the device.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x59)	1
2	Payload Length (0)	2
4	CRC	2

Figure 3-19: Reset Message

The response to this message is the same as the command, and neither message contains a payload.

3.8 Enquire Battery

This command will request the battery status from **icListen**. This data includes the charge status, as well as some more in depth data which can be used to diagnose battery issues. This command is currently supported by **icListen HF** and **icListen AF**.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5A)	1
2	Payload Length (0)	2
4	CRC	2

Figure 3-20: Enquire Battery Command

This command requires no payload.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5A)	1
2	Payload Length (2+n)	2
4	Response Version	2
6	Version Specific Payload	n
6 + n	CRC	2

Figure 3-21: Enquire Battery Response

Response Version: This version is used to determine what contents will follow in the version specific payload.
Version Specific Payload: This field will vary based on the **icListen** model and the response version.

3.8.1 icListen AF/HF Version Specific Battery Payloads

icListen HF and **icListen AF** share the same battery enquire response payload. Currently, only version 0 of this payload exists.

Byte Offset:	Field:	# Bytes:
6	Charging Status	1
7	Charge Percent	1
8	Charge	2
10	Capacity	2
12	Battery Current	2
14	Battery Voltage	2
16	Battery Temperature	2
18	Board Voltage	2
20	Microprocessor Voltage	2

Figure 3-22: Enquire Battery v0 Payload

Charging Status: This code indicates the charging state of the battery. Valid codes are:

0 = No Data Available
1 = Battery is Charging
2 = Battery is Discharging
3 = Battery is not Charging or Discharging

Charge Percent: This is the estimated charge % of the internal battery. A value of 0xFF indicates that this data is unavailable.
Charge: This is the estimated charge currently stored in the battery, measured in mAh.
Capacity: This is the capacity of the battery when fully charged, measured in mAh.
Battery Current: This is current flowing into the battery, measured in mA. A negative current indicates that current is flowing out of the battery (discharging).
Battery Voltage: This is the voltage measured on the battery terminals, measured in mV.
Battery Temperature: This is temperature of the battery, measured in tenths of °C (303 = 30.3 °C).
Board Voltage: This is regulated supply voltage to the **icListen**, measured in mV.
Microprocessor Voltage: This is the supply voltage to the main microprocessor, measured in mV.

The battery state can report a status code of 3 (NOT charging or discharging), for one of 3 reasons: the battery may be fully charged, the status may have been updated during the switch between charging and discharging, or the charging may have been stopped due to high battery temperature.

3.9 Enquire Time Sync

This command will request the time synchronization status from **icListen**. This data includes the PPS sync mode/status, system time set mode/status, offset from PPS, and a monotonic PPS counter. This command is supported by all **icListen** models.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5B)	1
2	Payload Length (0)	2
4	CRC	2

Figure 3-23: Enquire Time Sync Command

This command requires no payload.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5B)	1
2	Payload Length (2+n)	2
4	Response Version	2
6	Version Specific Payload	n
6 + n	CRC	2

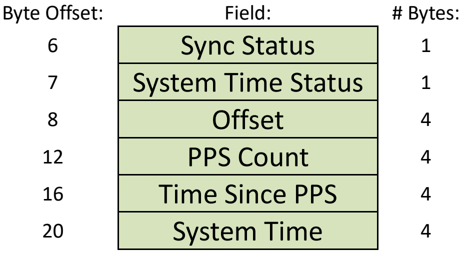
Figure 3-24: Enquire Time Sync Response

Response Version: This version is used to determine what contents will follow in the version specific payload.

Version Specific Payload: This is a variable field, which changes based on the response version. For any given device, as the enquire version number increases, new values are added to this part of the payload, and the existing fields from the previous version are still maintained. This allows a driver which was written for an older firmware version to continue to parse data from the message correctly, even when the firmware is updated.

3.9.1 Version Specific Enquire Time Sync Payloads

All **icListen** devices currently share the same version specific payload (version 0).



Byte Offset:	Field:	# Bytes:
6	Sync Status	1
7	System Time Status	1
8	Offset	4
12	PPS Count	4
16	Time Since PPS	4
20	System Time	4

Figure 3-25: Enquire Time Sync v0 Payload

Sync Status: This code indicates the status of synchronization. Valid codes are:

0 = Sync input/output disabled
1 = Sync output enabled
2 = Sync output enabled (with PPS encoded time)
3 = Sync input enabled (falling edge), no PPS detected
4 = Sync input enabled (falling edge), syncing to detected PPS
5 = Sync input enabled (falling edge), synced to PPS
6 = Sync input enabled (rising edge), no PPS detected
7 = Sync input enabled (rising edge), syncing to detected PPS
8 = Sync input enabled (rising edge), synced to PPS

System Time Status: This code indicates how the system time was set. Valid codes are:

0 = Time value has not been set
1 = Time was manually set
2 = Time set from RTC at start-up
3 = Time was set from file system at start-up
4 = Time was set from NTP
5 = Time was set from GPS
6 = Time was set from PTP
128 -> 137 = Time set from PPS encoded time. Subtracting 128 from the value will give the time status code of the PPS source (ex: 129 – 128 = 1 = Time set manually).

Sync Offset: This is the measured time offset in nanoseconds, between the detected PPS edge, and this system's second edge.

PPS Count: This is a monotonic counter of PPS inputs detected when configured for sync input. This value will be set to 0 if sync input is not enabled.

Time Since PPS: This is the number of seconds that have elapsed since a PPS signal has been verified.

System Time: This is the UNIX time value of **icListen**'s system clock. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC

4 Data Collection

Data can be retrieved on **icListen**’s command and control channel by sending the Collect command, with a scan mask set appropriately for the desired data. Data on **icListen HF** may also be retrieved using the streaming data channels. See the *icListen Streaming Telemetry* document for more details on using the streaming channels.

The commands detailed in this section are:

Collect Data
Stop Data Streams

4.1 Collect Data

This command is used to request sensor data from **icListen**.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x43)	1
2	Payload Length (1)	2
4	Scan Mask	1
5	Extended Scan Mask	(1)
5+(1)	CRC	2

Figure 4-1: Collect Command

Scan Mask: This is a bitmask used to select which pieces of data will be requested from **icListen**. A description of the data associated with each bit in the scan mask can be found in a following sub-section.
Extended Scan Mask: This is an optional second bitmask used to select more data to request from **icListen**. A description of the data associated with each bit in the scan mask can be found in the Extended Collect Data-section.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x43)	1
2	Payload Length (1+n)	2
4	Scan Mask	1
5	Mask Specific Payload	n
5 + n	CRC	2

Figure 4-2: Collect Response

Scan Mask: This is a bitmask indicates what data is included in the response message. The data in the response may not always be the same as what was requested, as some requested data may not yet be ready, or not be supported by the **icListen** model. A description of the data associated with each bit in the scan mask can be found in the following sub-section.
Mask Specific Payload: This field is variable, and contains any requested data, which was available. See the following sub-sections for more detail on the contents of this payload.

4.1.1 Collect Scan Mask Description

The following table describes what data each bit in the scan mask is used for.

Scan Mask Bit Field Description:

Bit	Name	Description
0	Guest Sensor Data	Voltage measurement of the guest sensor (where available). This is measured in tenths of Volts (54 = 5.4V). This is a 16 bit field
1	Humidity	This is the internal relative humidity in tenths of % (321 = 32.1 % rh). This is a 16 bit field
2	Temperature	This is internal temperature in tenths of °C (123 = 12.3 °C). This is a 16 bit field
3	Extended Mask	This bit is used if an additional mask byte is to be used to request more data. This mask byte contains its own bitfield which is used to request more data which will follow the data requested by the first mask byte.
4	Spectrum Data	This is a data chunk which contains the next available set of spectrum data, as well as the setup parameters used when the data was collected. See more detail in the following sub-section.
5	Waveform Data	This is a data chunk which contains the next available samples of waveform data, as well as the setup parameters used when the data was collected. See more detail in the following sub-section.
6	File System Data	This is a data chunk containing the requested file system data. See more detail in the following sub-section.
7	Data Waiting	This bit indicates that more data is already available for immediate scanning. No additional data is sent when this bit is set, and if it is set in the command, it will be ignored.

It is recommended that no other data is scanned while File System Data is being scanned, in order to reduce the size of the response messages.

If the “Extended Mask” bit is set, the firmware will check for a second mask byte for the extended data requests. This second mask byte will not be returned with the response. Instead, all of the data returned will contain a tag indicating what type of data it is. Check the “Extended Collect Data”-section for more details on the extended data.

Not all fields will be present in the response every time. If data is not available, or if it has not been requested, it will not be present. No space is left for absent fields (ie: if only temperature was requested, the total payload will be only 3 bytes long – mask (1 byte)/temperature (2 bytes).

Data in the response is generally filled in the same order as their position in the bitmask. Guest Sensor, followed by Humidity, followed by, Temperature, etc. The only exception to this is the extended data, which is always at the end of the response.

Firmware Differences:
-HF/AF versions before 2.2 (Release 31) do not support the Extended Mask, Waveform Data, File System Data or Data Waiting bits

4.1.2 Spectrum Data

The spectrum data chunk returned with the collect response contains its own header information. This header information includes hardware and setup information required for interpreting the data.

Byte Offset:	Field:	# Bytes:
0	Sequence Number	2
2	Length	2
4	Gain	1
5	Sample Rate	4
9	Hydrophone Sensitivity	2
11	FFT Processing Type	2
13	Processing Parameters	n
13 + n	Spectrum Data	m

Figure 4-3: Collect Spectrum Data

Sequence Number: This sequence number is incremented for each new spectrum data set produced by the instrument. It can be used to determine if any data has been missed.
Length: This is the number of spectrum data points returned in the message.
Gain: This is the gain applied to the spectrum data, in dB.
Sample Rate: This the sample rate at which the data was collected, in Hz.
Hydrophone Sensitivity: This is the sensitivity of the hydrophone in dB re 1µPa.
FFT Processing Type: This is a code identifies how the FFT data was processed. See the “Job Setup” command section for more details.
Processing Parameters: This is a set of parameters which varies depending on the FFT processing type selected. It is either 2 or 4, 16 bit parameters. See the “Job Setup” command section for more details.
Spectrum Data: This is the actual power spectrum data. The data points are in ½ dB counts (2 counts per dB) relative to 1 µV.

4.1.3 Waveform Data

The waveform data chunk returned with the collect response contains its own header information. This header information includes hardware and setup information required for interpreting the data.

Byte Offset:	Field:	# Bytes:
0	Sequence Number	2
2	Data Format	1
3	Number of Bytes	2
5	Gain	1
6	Sample Rate	4
10	Hydrophone Sensitivity	2
12	Maximum Full Scale mV	2
14	Waveform Data	n

Figure 4-4: Collect Waveform Data

Sequence Number: This sequence number is incremented for each new waveform buffer produced by the instrument. It can be used to determine if any data has been missed.
Data Format: This is a code which determines the endianness and bit-depth (bits per sample) of the waveform data. The high bit is used to indicate endianness (0 = little endian, 1 = big endian), and the rest of the bits are used to indicate bit depth.
Number of Bytes: This is the number of bytes of waveform data returned in the message.
Gain: This is the gain applied to the waveform data, in dB.
Sample Rate: This is the rate at which the data was collected, in Hz.
Hydrophone Sensitivity: This is the sensitivity of the hydrophone in dB re 1µPa.
Maximum Full Scale mV: This is the value in millivolts that a full scale count corresponds to, without compensating for gain (ie: if 6dB of gain is applied, the full scale count would correspond to a voltage 6dB below this value).
Waveform Data: This is the actual waveform data.

Firmware Differences:

-HF/AF versions before 2.2 (Release 31) do not support the Waveform Data collect response

4.1.4 File System Chunk (icListen LF only)

The File System Chunk contains the Collect Responses for each of the File System commands. It is recommended that no other mask bits are set when collecting file system data, to reduce the size of the response messages. These responses are covered individually for each command in the *File System Commands* section.

4.1.5 Extended Collect Data

The following table describes what data each bit in the “Extended Scan Mask” byte is used for.

Extended Scan Mask Bit Field Description:

Bit	Name	Description
0	Heading Data	This requests heading data from icListen. Accelerometer and Magnetometer readings are sent with the response.
1..3	Reserved	These bits are reserved for future use, and should be set to 0.
4	Spectrum Data Time	This requests timing data which can be used to determine the precise time of the first sample used in the spectrum data returned with this message. This data will not be sent if spectrum data is not included in the message.
5	Waveform Data	This requests timing data which can be used to determine the precise time of the first sample of waveform data returned with this message. This data will not be sent if waveform data is not included in the message.
6,7	Reserved	These bits are reserved for future use, and should be set to 0.

Data returned in response to the extended mask is returned in chunks, which include a data type code and length. No modified mask byte is returned for extended data. If the requested data is unavailable, the requested data chunk is not returned in the response.

Byte Offset:	Field:	# Bytes:
0	Data Type	2
2	Length (n)	2
4	Payload	n

Figure 4-5: General Extended Collect Data Chunk

Data Type: This field is used to indicate the type of data in the following payload. The data type will have a high byte of 0, and the low byte will be the same as the Extended Mask byte with the relevant bit set for the following data. (ie: Spectrum data Time is bit 4(0x10) in the mask, so the Data Type field for this data would be 0x0010).
Length: This is the number of bytes in the payload of this chunk.
Payload: This is a variable length field, which contains the actual data.

If new data is added to these chunks in the future, that data will be appended to the end of the payload. This allows a driver to skip new data at the end of the chunk that it does not understand, while still being able to read the beginning of the payload which it does understand.

The following sub-sections describe the contents of the extended collect data response chunks.

Firmware Differences:

-LF versions before 2.7 do not support the extended collect data
-HF/AF versions before 2.2 (Release 31) do not support the extended collect data

4.1.5.1 Heading Data

This data includes readings taken from icListen's internal accelerometer and magnetometer.

Byte Offset:	Field:	# Bytes:
0	Data Type (0x0001)	2
2	Length (12)	2
4	Acceleration (X)	2
6	Acceleration (Y)	2
8	Acceleration (Z)	2
10	Magnetic Field (X)	2
12	Magnetic Field (Y)	2
16	Magnetic Field (Z)	2

Figure 4-6: Heading Data Chunk

Data Type: This field is used to indicate the type of data in the payload. For heading data this value is 0x0001.

Length: This is the number of bytes in the payload of this chunk (12 bytes).

Acceleration (X,Y,Z): These are the acceleration readings along the x, y, and z axes. These readings are given in milli g-units (1 g-unit = 1 grav). These are signed 16bit values.

Magnetic Field (X,Y,Z): These are the magnetic field readings along the x, y, and z axes. These readings are given in milligauss (mG). Multiplying this value by 100 gives nanotesla (nT) values. These are signed 16bit values.

4.1.5.2 Spectrum Data Time

This data includes the information necessary for determining the precise time of the spectrum data. The spectrum data time chunk will not be sent if no spectrum data is included in the message.

Byte Offset:	Field:	# Bytes:
0	Data Type (0x0010)	2
2	Length (8)	2
4	Samples Into Second	4
8	Time	4

Figure 4-7: Spectrum Data Time Chunk

Data Type: This field is used to indicate the type of data in the payload. For spectrum data time this value is 0x0010.

Length: This is the number of bytes in the payload of this chunk (8 bytes).

Samples Into Second: This is the number of samples into the second of the first raw sample used for the spectrum data calculations. The sample rate, which is returned with the spectrum data, can be used to precisely convert from samples to seconds. It should be noted that spectrum data does not always begin at the top of a second, so the first data set returned for any given second may not be at 0 samples.

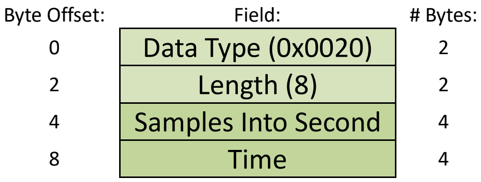
Time: This is the UNIX time value of the second containing the first sample used for the spectrum data calculations. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC.

The precise time into the second that the data begins at is calculated as follows:

Precise Time = Unix Time + Samples Into Second (samps) / Sample Rate (samps/second)
ex: 4000 kS/s data, Unix Time = 1393437870(26 Feb 2014 18:04:30), Samps in Second = 200
Full Time = 1393437870(Unix Time) + 200(samples into second)/4000(sample rate)
= 26 Feb 2014 18:04:30.050

4.1.5.3 Waveform Data Time

This data includes the information necessary for determining the precise time of the waveform data. The waveform data time chunk will not be sent if no waveform data is included in the message.



Byte Offset:	Field:	# Bytes:
0	Data Type (0x0020)	2
2	Length (8)	2
4	Samples Into Second	4
8	Time	4

Figure 4-8: Waveform Data Time Chunk

Data Type: This field is used to indicate the type of data in the following payload. For waveform data time this value is 0x0020.

Length: This is the number of bytes in the payload of this chunk (8 bytes).

Samples Into Second: This is the number of samples into the second of the first waveform data sample in this collect message.

Time: This is the UNIX time value of the first waveform data sample. UNIX time is defined as the number of seconds since 00:00:00 January 1, 1970 UTC.

The precise time into the second that the data begins at is calculated as follows:

Precise Time = Unix Time + Samples Into Second (samps) / Sample Rate (samps/second)
ex: 4 kS/s data, Unix Time = 1393437870(26 Feb 2014 18:04:30), Samps in Second = 200
Full Time = 1393437870 + 200/4000
= 26 Feb 2014 18:04:30.050

4.2 Stop Data Streams

This command is used to stop streaming all data, and disconnect all connections to streaming ports.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x51)	1
2	Payload Length (0)	2
4	CRC	2

Figure 4-9: Stop Data Streams Message

The response to this message is the same as the command, and neither message contains a payload.

Firmware Differences:

-HF versions before 1.2 do not send a response to this message

5 File System Commands

Unless otherwise noted, the commands for the **icListen** file system are performed in a 2 step process. First the command is issued and an immediate response is sent, to confirm the command was accepted. The results of the command are later issued as a collect response, sent in the body of a *Collect Data* command. Collect responses are retrieved by sending a “Collect” command with the “File System Data” bit set in the scan mask.

All file system commands include a 16 bit universally unique identifier (UUID), to ensure the proper order of messages. For **icListen LF** the UUID is set by the issuer of the command, and for other **icListen** models the UUID is issued by the device. This allows for multi-message file transfers. It is the responsibility of the user to maintain a record of UUIDs issued to **icListen**, and what commands/parameters were issued with those commands.

5.1 icListen AF/HF File System Commands

icListen AF/HF can queue up to 8 file system requests at a time. If the queue is full, response messages must either be collected, or requests cancelled in order to make room for new requests. Sending the file system enquire command will indicate the UUIDs and status of all queued file system requests.

Filenames (including the path) have a maximum size of 240 characters, and are case sensitive. Filenames must only be made up of printable ACII characters, and cannot include the '?' or '*'(wild card characters). All '/' characters will be considered as folder division. All file paths are considered relative to the "/home/icListen/" folder. These are the contents of the "/home/icListen" folder:

Config/ - This folder contains the job setup files, and epoch trigger setup files used by the instrument

Data/ - This folder contains all logged data on the instrument

update/ - This folder is where firmware update packages are placed

History.txt - This is an ASCII text file containing a log of certain events for the device, such as start-ups, firmware updates, time synchronisation, and certain error conditions

Response data retrieved using the collect messages are limited to 4096 bytes of data (including the UUID, error code, and length fields).

Filesystem commands will vary in their payload, based on the type of command being sent.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5C)	1
2	Payload Length (4+n)	2
4	Command	2
6	Request ID	2
8	Command Specific Payload	n
8 + n	CRC	2

Figure 5-1: icListen AF/HF File System Command

Command: This code is the command that is being requested. The following commands are supported:

0 = Login to filesystem
1 = Enquire File System
2 = Cancel File Request
3 = List File(s)
4 = Read File
5 = Write File
6 = Delete File(s)

Command: This value is specified by the requestor, and is returned in the immediate response. Its purpose is to allow the requestor to verify which request the immediate response corresponds to, and should be unique.

Command Specific Payload: The contents of this field varies based on the command, and are covered in the following command sections.

The immediate responses to file system commands contain the following fields:

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x5C)	1
2	Payload Length (12+n)	2
4	Command	2
6	Request ID	2
8	UUID	2
10	Error Code	2
12	Reserved	2
14	Length (n)	2
16	Command Specific Payload	n
16 + n	CRC	2

Figure 5-2: icListen AF/HF File System Command Response

Command: This value is the command that was requested.
Request ID: This value is specified by the requestor in the command message. Its purpose is to allow the requestor to verify which request the immediate response corresponds to, and should be unique.
UUID: This value is a unique identifier, which can be used to match file system responses with requests. A value of 0xFFFF will be returned if there was an error, and the request could not be accepted. It will also be set to 0xFFFF for any command that requires no further action after the immediate response is sent (login, enquire, and cancel commands). It is set by the device rather than the requestor to guarantee uniqueness.
Error Code: This code will indicate if the command was accepted, and why it was not if it failed. Possible error codes in the immediate response are:

0 = No error (command accepted).

1 = Unsupported command.

2 = Not enough memory to complete the request).

3 = Maximum number of requests are already queued.

4 = Command payload is too small (see command sections below command specific payloads).

5 = UUID not found in queue (this error is sent in response to Cancel File Requests).

6 = Invalid user ID.

7 = Incorrect password.

8 = Invalid filename.

9 = Access denied (logged in user does not have permission to use file).

10 = Path points to a folder. The path should not be a folder for read/write/delete commands. Using the delete command with a wildcard character can also result in folders matching the wildcard which will return this error (files that match will still be deleted).

11 = Invalid offset.

Reserved: This field is reserved to force 32bit alignment at the start of the command specific payload.
Length: This is the length of the command specific payload.
Command Specific Payload: This field is variable in length, and changes depending on the command that was issued.

Collect responses to file system requests are returned in the response to collect messages, and have the following structure:

Byte Offset:	Field:	# Bytes:
0	Command	2
2	UUID	2
4	Error Code	2
6	Length (n)	2
8	Command Specific Payload	n

Figure 5-3: icListen AF/HF File System Collect Response

Command: This is the command that this message is responding to.
UUID: This is the UUID that was assigned to this request.
Error Code: This code is used to indicate if any errors have occurred for this request. Possible error codes in the collect response are:

0 = No error (command completed successfully).

2 = Not enough memory to complete the request).

9 = Access denied (logged in user does not have permission to use file).

10 = Path points to a folder. The path should not be a folder for read/write/delete commands. Using the delete command with a wildcard character can also result in folders matching the wildcard which will return this error (files that match will still be deleted).

12 = Operation failed (message parameters appeared valid, but file system failed to perform the requested action).

13 = File not found.

14 = Too much data for response. This error may be returned by the List File(s) or Read File commands, if the requested data would exceed the limit on the response size. The maximum amount of data that can be transferred will be returned in the message payload. To retrieve the rest of the data, another request will need to be issued, skipping past the files (list command) / bytes (read command) returned with this message.

15 = End of file. This code will be returned when a read command has requested data that would go past the end of the file. The full available data to read will have been returned with the response.

16 = Error getting file stats (optional fields could not be read)

Length: This is the length of the command specific payload.
Command Specific Payload: This section contains a variable payload which is different for each filesystem command.

5.1.1 Login to File System

This command has an immediate response, and does not require a collect message to be sent. It is used to log into the hydrophone. The main user for icListen is user ID 500, which uses an empty password.

The Command Specific Payload for this operation is either empty (for logout), or contains the user id number and the password required for login:

Byte Offset:	Field:	# Bytes:
0	User ID	4
4	Password	N

Figure 5-4: icListen AF/HF Login to File System Command Specific Payload

User ID: This is the ID number of the user that wished to be logged in. 500 is the main icListen user. This is an unsigned 32 bit value.

Password: This is the password required for this user. It is a NULL terminated ASCII string, and is case sensitive.

Byte Offset:	Field:	# Bytes:
0	User ID	4

Figure 5-5: icListen AF/HF Login to File System Response

User ID: This is the ID number of the user that is currently logged in. If no user is logged in, this field is not included in the response. This is an unsigned 32 bit value.

There is no collect response is for this command.

5.1.2 Enquire File System

This command has only an immediate response, and does not require a collect message to be sent. It requests the status of all queued file system commands. Up to 8 file system commands may be queued at one time. To remove requests from the queue, the resultant data must be collected, or the request must be cancelled.

The Command Specific Payload for this operation is empty.

The command specific payload for this command's response is shown in the following figure:

Byte Offset:	Field:	# Bytes:
0	Number of Commands (n)	2
2	Command 1 UUID	2
4	Command 1 Status	2
…	…	
…	Command n UUID	2
…	Command n Status	2

Figure 5-6: icListen AF/HF Enquire File System Response

Number of Commands: This is the number of commands that are in the queue.
Command # UUID: This is the UUID of the command at position "#" in the queue.
Command # Status: This is the status of the command at position "#" in the queue. Valid status codes are:

0 = Command is waiting in queue
1 = Command is being acted on currently
2 = Complete/Response waiting to be collected
3 = Command is being cancelled

The payload will contain "n", UUIDs and status's (where n is the number of commands). If the number of commands is 0, no UUIDs or statuses will be returned.

There is no collect response for this file system command.

5.1.3 Cancel File Request

This command has an immediate response, and does not require a collect message to be sent. This command is used to remove unwanted file system commands from the queue, in order to make room for new commands to be issued. If a file write or delete command is cancelled mid-operation, the state of the files being operated on is unknown (file may or may not yet be deleted, and requested write may or may not have yet been completed).

The command specific payload for this command is as follows:

Byte Offset:	Field:	# Bytes:
0	UUID	2

Figure 5-7: Cancel File Request Command Specific Payload

UUID: This is the UUID of the command which is to be cancelled. If the UUID is set to 0xFFFF, all queued requests will be cancelled.

The Command Specific Payload for the immediate response is empty for this command, and no collect response is sent.

5.1.4 List File(s)

The List File command is used to acquire a listing of file names, and optionally other status fields for the files as well.

The command specific payload for this command is as follows:

Byte Offset:	Field:	# Bytes:
0	From File	4
4	Number of Files	2
6	Flags	2
8	Path	n

Figure 5-8: List Files(s) Command Specific Payload

From File: This value is used to skip some files from being listed in the response. For example, if you wish to list the last 3 files in a folder containing 10 files, this value should be set to 7. This is an unsigned 32 bit value.

Number of Files: This is the maximum number of files to return in the response. Setting this to 0 will cause no files to be listed, but will update the “From File” field in the collect response to indicate how many files in total are in the directory that match the Path string. This is an unsigned 16 bit value.

Flags: This is a bitmask which can be used to select optional fields to retrieve with the listing. The Bit values are as follows:

Bit 0 (0x0001) = Show Link
Bit 1 (0x0002) = File Mode
Bit 2 (0x0004) = Owner
Bit 3 (0x0008) = Group
Bit 4 (0x0010) = File Size
Bit 5 (0x0020) = Modified Date
Bits 6..15 (0xFFC0) = Reserved

Path: This is the path to the file or folder being listed. It is a NULL terminated ASCII string, and is case sensitive. A wildcard character (* or ?) may be used in order to list multiple similar files (ex: Data/*.wav will list all wav files in the data folder).

The command specific payload of the immediate response message is empty.

The command specific payload for the collect response for this command contains the actual data requested:

Byte Offset:	Field:	# Bytes:
0	From File	4
4	Number of Files (n)	2
6	Flags	2
8	File 1 Name	N
8+N	File 1 Link	M
8+N+M	File 1 File Mode	2
10+N+M	File 1 Owner	4
14+N+M	File 1 Group	4
18+N+M	File 1 Size	8
22+N+M	File 1 Modified Date	8
30+N+M	…	…

Figure 5-9: List Files(s) Collect Response

From File: This value indicates how many files in the folder were skipped in the listing. This is an unsigned 32 bit value.

Number of Files: This is the number of files returned in the response. This may be lower than the requested maximum, if the full data would be too much to include in the payload. This is an unsigned 16 bit value.

Flags: This bitmask indicates which pieces of data are included for each file. The bit values are as follows:

Bit 0 (0x0001) = Show Link
Bit 1 (0x0002) = File Mode
Bit 2 (0x0004) = Owner
Bit 3 (0x0008) = Group
Bit 4 (0x0010) = File Size
Bit 5 (0x0020) = Modified Date
Bits 6..15 (0xFFC0) = Reserved

File # Name: This is the NULL terminated ASCII filename string.

File # Link: This optional field is the NULL terminated ASCII filename string of the file that a linked file points to. If the file is not a link, this string will be NULL.

File # File Mode: This optional field represents the file mode. The bits have the following meanings:

Bits 0-2 = Other Permissions: Read (0x0004) / Write (0x0002) / Execute (0x0001)
Bits 3-5 = Group Permissions: Read (0x0020) / Write (0x0010) / Execute (0x0008)
Bits 6-8 = User Permissions: Read (0x0100) / Write (0x0080) / Execute (0x0040)
Bits 9-11 = Special Permissions: SetUID (0x0800) / SetGID (0x0400) / Sticky (0x0200)
Bits 12-15 = File type:
Unknown (0x0000), FIFO (0x1000), Character Device (0x2000), Directory (0x4000)
Block Device (0x6000), Regular File (0x8000), Link (0xA000), Socket (0xC000)
Bits 21..31 = Reserved

File # Owner: This optional field is the user ID number that owns the file (500 is the main icListen user). This is an unsigned 32 bit value.
File # Group: This optional field is the group ID number that owns the file (500 is the main icListen user). This is an unsigned 32 bit value.
File # Size: This optional field is the size of the file in bytes. This is a signed 64 bit value.
File # Modified Date: This optional field is the time that the file was last modified (Unix time). This is a signed 64 bit value.

Note that all of the “File #” fields are optional (with the exception of the file name), and can be enabled/disabled by setting the “Flags” field appropriately. For most users, only the file name, mode, and possibly the file size will be of interest. The requested fields are repeated for each file.

Also note that the listed files are returned in the order that they were found on the filesystem (not sorted). This is done in order to improve memory/time efficiency in the device. It is the responsibility of the driver/software writer to sort the list of files that are returned, if this feature is desired.

If the requested listing would use more data than can fit in a response, the amount that can fit will be returned, and the “Number of Files” will be adjusted to reflect this. Also, if the internal memory runs out during the operation, as much data as could be retrieved will be returned. If either situation occurs, the error code returned with the response will indicate what happened.

5.1.5 Read File

The Read File command is used to read/download the contents of a file.

The command specific payload for this command is as follows:

Byte Offset:	Field:	# Bytes:
0	Offset	8
4	Number of Bytes	2
6	Path	n

Figure 5-10: Read Files Command Specific Payload

Offset: This is the offset in bytes where the data should be read from. This is a signed 64 bit value.
Number of Bytes: This is the maximum number of bytes to return in the response. Less than this many bytes may be read if the end of file is reached, or too much data for the response was requested. Setting this value to 0 will request that the maximum allowable amount of data be read. This is an unsigned 16 bit value.
Path: This is the path to the file being read from. It is a NULL terminated ASCII string, and is case sensitive.

The command specific payload of the immediate response message is empty.

The command specific payload for the collect response for this command contains the actual data requested:

Byte Offset:	Field:	# Bytes:
0	Offset	8
2	Number of Bytes	2
4	Data	n

Figure 5-11: Read Files Collect Response

Offset: This is the offset in bytes where the data was read from. This is a signed 64 bit value.
Number of Bytes: This is the number of data bytes returned with the message, which in some cases will be less than the amount requested. This is an unsigned 16 bit value.
Data: This is the actual data that was read from the file.

If more data was requested than could fit in the response or the end of file is reached, the amount that could fit will be returned, and the "Number of Bytes" will be adjusted to reflect this. If either of these situations occurred, the error code returned with the response will indicate this.

5.1.6 Write File

The Write File command is used to write/upload the contents of a file to the **icListen**.

The command specific payload for this command is as follows:

Byte Offset:	Field:	# Bytes:
0	Offset	8
4	Number of Bytes	2
6	Data	n
6+n	Path	m

Figure 5-12: Write Files Command Specific Payload

Offset: This is the offset in bytes where the data should be written to. This is a signed 64 bit value.
Number of Bytes: This is the number of bytes to write. This is an unsigned 16 bit value.
Data: This is the actual data which is to be written.
Path: This is the path to the file being written from. It is a NULL terminated ASCII string, and is case sensitive.

The command specific payload of the immediate response message is empty for this command.

The command specific payload for the collect response for this command indicates how much data was written:

Byte Offset:	Field:	# Bytes:
0	Offset	8
2	Number of Bytes	2

Figure 5-13: Write Files Collect Response

Offset: This is the offset in bytes where the data was written to. This is a signed 64 bit value.
Number of Bytes: This is the number of data bytes that were actually written, which in some cases will be less than the amount requested. This is an unsigned 16 bit value.

5.1.7 Delete File(s)

The Delete File command is used to delete files from **icListen's** file system.

The command specific payload for this command contains only the path of the file(s) to be deleted.

Byte Offset:	Field:	# Bytes:
0	Path	n

Figure 5-14: Delete Files Command Specific Payload

Path: This is the path to the file(s) being deleted. It is a NULL terminated ASCII string, and is case sensitive. A wildcard character (* or ?) may be used to delete multiple files. The wildcard character can be used on filenames only, and will not be accepted in folder names (ex: "Data/*" will deleted all files within the Data folder, but "Da*/" will not be accepted).

Note that this command will not delete folders, and will not recursively delete files within folders.

If an error code is returned with the collect response, some files may still have been deleted. For this reason, it is recommended that the List File(s) command be used to determine which files (if any) have been deleted.

When using the wildcard character to delete multiple files, if any folders are found which also match the wildcard string, an error will be set in the response indicating was folder was found. The folder will not be deleted, but all files matching the wildcard string should still be deleted. It should be noted that the "." And ".." folders will not cause this error to be set if they are found through a wildcard string match.

The command specific payload of the immediate response message is empty, and there is no command specific payload for the collect response to this command.

5.2 icListen LF File System Commands

The **icListen LF** file system has a maximum directory structure depth of 5. Directory names may be up to 12 characters long, and all files are in 8.3 format (maximum 8 character name, a dot '.', and a 3 character extension). Filenames are not case sensitive.

All **icListen LF** file system message payloads are in big endian format. The message framing (sync, command byte, length, CRC) is little endian format. All immediate command responses to file system messages have the same structure:

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type	1
2	Payload Length (3)	2
4	UUID	2
6	Command Accepted	1
7	CRC	2

Figure 5-15: icListen LF File System Command Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Command Accepted: This is used to indicate if a file system command is accepted. A value of 0 indicates that the command was rejected, and a value of 1 indicates that it was accepted.

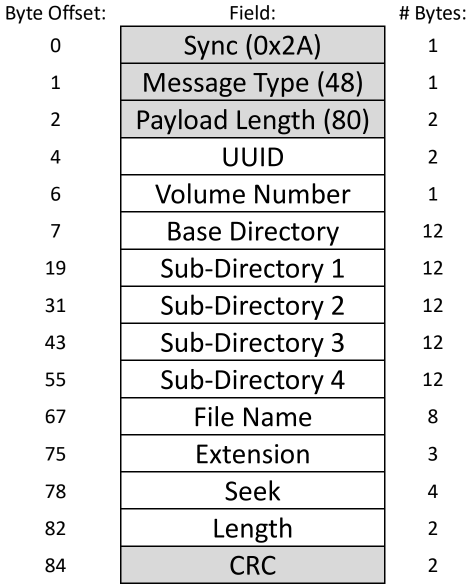
5.2.1 File System Error Codes

All files system commands that provide a response to the collect command provide an error code with the response. The following table gives descriptions for each of the error codes.

Code	Description
0	No Error
1	Invalid Volume Requested
2	File Open Failed
3	File Seek Failed
4	File Write Failed
5	File Read Failed
6	Reached End of File
7	File Close Failed
8	File Delete Failed
9	Directory Listing Failed
10	Copy Failed
11	Rename Failed
12	File System Access Error
13	File System Idle
14	File System Busy for USB
15	Volume Full
16	Corrupt Sector

5.2.2 Read File

This command is used to read data from the **icListen LF** file system.



Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (48)	1
2	Payload Length (80)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	File Name	8
75	Extension	3
78	Seek	4
82	Length	2
84	CRC	2

Figure 5-16: Read File Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Base Directory:
Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.
File Name: This is name of the file (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.
Seek: This is the offset into the file where the data will begin to be read from, in bytes.
Length: This is the number of bytes that will be read from the file. The maximum length that may be requested is 4096 bytes.

This is the collect response sent when the data has been retrieved from the file system.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1
4	Length	4
8	Data	n

Figure 5-17: Read File Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.

Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.

Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

Length: This is the number of bytes that will follow in the Data portion of the payload.

Data: This is the actual binary data read from the file.

5.2.3 Write File

This command is used to write data to the **icListen LF** file system.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (49)	1
2	Payload Length (80+n)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	File Name	8
75	Extension	3
78	Seek	4
82	Length	2
84	Data	n
84 + n	CRC	2

Figure 5-18: Write File Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Base Directory:
Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.
File Name: This is name of the file (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.
Seek: This is the offset into the file where the data will begin to be written to, in bytes.
Length: This is the number of bytes that will be written to the file. The maximum length to write is the 1962 bytes.
Data: This is the actual binary data to be written.

This is the collect response sent when the data has been written to the file system.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1

Figure 5-19: Write File Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

5.2.4 Delete File

This command is used to delete a file from the **icListen LF** file system.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4A)	1
2	Payload Length (74)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	File Name	8
75	Extension	3
78	CRC	2

Figure 5-20: Delete File Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.

Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

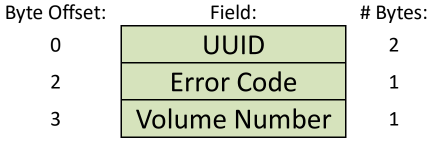
Base Directory:

Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.

File Name: This is name of the file (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.

Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.

This is the collect response sent when the file has been deleted from the file system.



Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1

Figure 5-21: Delete File Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

5.2.5 Directory Listing

This command is used to retrieve a list of files in a directory on the **icListen LF** file system.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4B)	1
2	Payload Length (63)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	CRC	2

Figure 5-22: Directory Listing Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the directory to list exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**’s.
Base Directory:
Sub-Directory 1 - 4: These directories point to where the directory to list exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.

This is the collect response sent when the data has been retrieved from the file system.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1
4	Length	4
8	Data	n

Figure 5-23: Directory Listing Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the directory to list exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Length: This is the number of bytes that will follow in the Data portion of the payload.
Data: This is a list of 12 byte ASCII strings describing the files contained in the directory. All strings that are not 12 bytes long are padded to 12 bytes with NULL characters. All objects containing a dot and extension can be considered files, and those without are directories. The maximum number of returned objects is 170 (including the "." and ".." entries).

5.2.6 File Size

This command is used to request the size of a file from the **icListen LF** file system.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4C)	1
2	Payload Length (74)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	File Name	8
75	Extension	3
78	CRC	2

Figure 5-24: File Size Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Base Directory:
Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.
File Name: This is name of the file (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.

This is the collect response sent when the data has been retrieved from the file system.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1
4	Size	4

Figure 5-25: File Size Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the directory to list exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Length: This is the size of the file in bytes.

5.2.7 Rename File

This command is used to rename a file on the **icListen LF** file system. The rename command will not move the file to a new location.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4D)	1
2	Payload Length (85)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	Old File Name	8
75	Old Extension	3
78	New File Name	8
86	New Extension	3
89	CRC	2

Figure 5-26: Rename File Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Base Directory:
Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.
Old File Name: This is name of the file (without the dot or extension or). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
Old Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.
New File Name: This is the new name that the file will be given (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
New Extension: This is the new extension that the file will be given. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.

This is the collect response sent when the file has been renamed.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1

Figure 5-27: Rename File Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**’s.

5.2.8 Copy File

This command is used to copy a file on the **icListen LF** file system to a new name/location. If a file already exists on the file system with the new file name, that file will be overwritten.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4E)	1
2	Payload Length (145)	2
4	UUID	2
6	Volume Number	1
7	Old Base Directory	12
19	Old Sub-Directory 1	12
31	Old Sub-Directory 2	12
43	Old Sub-Directory 3	12
55	Old Sub-Directory 4	12
67	Old File Name	8
75	Old Extension	3
78	New File Name	8
86	New Extension	3
89	New Base Directory	12
101	New Sub-Directory 1	12
113	New Sub-Directory 2	12
125	New Sub-Directory 3	12
137	New Sub-Directory 4	12
149	CRC	2

Figure 5-28: Copy File Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.
Old Base Directory:
Old Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.
Old File Name: This is name of the file (without the dot or extension or). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
Old Extension: This is the extension of the file. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.
New File Name: This is the new name that the file will be given (without the dot or extension). This is an ASCII string, padded to 8 bytes using NULL characters if necessary.
New Extension: This is the new extension that the file will be given. This is an ASCII string, padded to 3 bytes using NULL characters if necessary.
New Old Base Directory:
New Old Sub-Directory 1 - 4: These directories point to where the file exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.

This is the collect response sent when the file has been copied.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1

Figure 5-29: Copy File Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

5.2.9 Delete Directory

This command is used to delete a directory on the **icListen LF** file system. All contents of the directory will also be deleted

Warning: For a full 16 Gb card, deleting all of the data may take more than an hour. The collect response will not be sent until this operation is complete, and no other file system operations may take place during this time.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (4F)	1
2	Payload Length (63)	2
4	UUID	2
6	Volume Number	1
7	Base Directory	12
19	Sub-Directory 1	12
31	Sub-Directory 2	12
43	Sub-Directory 3	12
55	Sub-Directory 4	12
67	CRC	2

Figure 5-30: Delete Directory Command

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.

Volume Number: This is the volume number that the directory exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

Base Directory:

Sub-Directory 1 - 4: These directories point to where the directory to delete exists. These are ASCII strings, padded to 12 bytes using NULL characters if necessary.

This is the collect response sent when the directory has been deleted.

Byte Offset:	Field:	# Bytes:
0	UUID	2
2	Error Code	1
3	Volume Number	1

Figure 5-31: Delete Directory Collect Response

UUID: This value is a unique identifier, which can be used to identify file system command responses with requests.
Error Code: This code is used to indicate the cause of failure (or success) of the file system operation.
Volume Number: This is the volume number that the file exists on. **icListen LF** has two volumes: 1 and 2. Volume 2 is not available on all **icListen LF**'s.

5.2.10 Enquire File System

This command has an immediate response, and does not require a collect message to be sent. The purpose of this command is to retrieve a list of file system commands that are waiting to be processed. As requests are made, they are added to the list. As soon as the request has been serviced, the command is removed from the list, and its collect response is queued up to be transmitted upon request.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x50)	1
2	Payload Length (0)	2
4	CRC	2

Figure 5-32: icListen LF Enquire File System Command

No payload is required for this command

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x50)	1
2	Payload Length (1+n)	2
4	Number of UUIDs	1
5	UUID List	n
5 + n	CRC	2

Figure 5-33: icListen LF Enquire File System Response

Number of UUIDs: This is the number of command UUIDs that are in the following list.
UUID List: This is list of 16bit UUIDs that correspond to file system commands that are waiting to be processed. It is the responsibility of the user that sent the commands to maintain a list of UUIDs that were transmitted, in order to determine the nature of the requests in the response.

6 Epoch Commands

Currently there are four commands for controlling the Epoch mode in **icListen**. These commands are:

Enquire Triggers
Setup Trigger
Remove Trigger
Clear Triggers

6.1 Enquire Triggers

The purpose of this command is to retrieve a list triggers that are currently configured in the system.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x52)	1
2	Payload Length (0)	2
4	CRC	2

Figure 6-1: Enquire Triggers Command

No payload is required for this command

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x52)	1
2	Payload Length	2
4	Number of Triggers	1
5	Setup Length 1 (2 + n)	2
7	Test Type 1	2
9	Type Specific Payload 1	n
9 + n	…	m
9 + n + m	CRC	2

Figure 6-2: Enquire Triggers Response

Number of Triggers: This is the number of trigger setups that are in the following list.
Setup Length: This is the length of the following setup in bytes (including the trigger type field).
Test Type: This type determines the type of trigger setup payload that follows. See the following sub-section for test types.
Type Specific Payload: This payload contains the actual setup parameters for this trigger. See the following sub-section for the valid type specific setup payloads.

The setup length, test type, and type specific payload are reported for each configured trigger.

6.1.1 Trigger Setup Payloads

The valid trigger setup types for **icListen** are described in the following table:

Type	Trigger	Description
0	Spectrum Data Above Threshold v1	This trigger tests if any value in the test frequency range (specified as the frequency bin number) is above the test threshold.
1	Spectrum Data Below Threshold v1	This trigger tests if any value in the test frequency range (specified as the frequency bin number) is below the test threshold.
4	Remove Trigger	This setup indicates that the given trigger number is not configured, and can be sent to remove a configured trigger.
5	Spectrum Data Above Threshold v2	This trigger tests if any value in the test frequency range (specified in Hz) is above the test threshold.
6	Spectrum Data Below Threshold v2	This trigger tests if any value in the test frequency range (specified in Hz) is below the test threshold.

Trigger types 0 and 1 are supported by **icListen LF**, and trigger types 4, 5, and 6 are supported by **icListen HF** and **icListen AF**.

Trigger types 0 and 1 share the same setup payload, and trigger types 5 and 6 share the same payload.

The following figures show the setup payloads used by all valid trigger setups.

Byte Offset:	Field:	# Bytes:
0	Signal Duration	2
2	Event Hold	2
4	Min Frequency Bin	2
6	Max Frequency Bin	2
8	Effect	2
10	Threshold	4

Figure 6-3: Trigger Type 0 and 1 Payload

Signal Duration: The number of spectrum results a signal must be present for, to activate the trigger.
Event Hold: The number of spectrum results after a signal goes away that a trigger remains active for.
Min Frequency Bin: This is the minimum frequency bin in the range checked by this trigger. The frequency at the center of the bin depends on the sample rate:

Bin Frequency = Bin # * Sample Rate / 1024.

Max Frequency Bin: This is the maximum frequency bin in the range checked by this trigger. The frequency at the center of the bin depends on the sample rate:

Bin Frequency = Bin # * Sample Rate / 1024.

Effect: This is a bitmask which is used to determine what effects are asserted when a trigger goes active. See the following sub-section for a description of the bit fields in this mask.
Threshold: This is the threshold against which data is checked, set in ½ dB counts relative to 1 µV.

Byte Offset:	Field:	# Bytes:
0	Trigger Number	1
1	Reserved	1

Figure 6-4: Trigger Type 4

Trigger Number: This is the number associated with this trigger. **icListen** can configure up to 5 triggers. Trigger numbers are a 0-based index (0 is trigger 1, 1 is trigger 2, etc).
Reserved: This byte is reserved, and should be set to zero when sending this setup.

Byte Offset:	Field:	# Bytes:
0	Trigger Number	1
1	Enabled	1
2	Signal Duration	2
4	Event Hold	2
6	Min Frequency	4
10	Max Frequency	4
14	Threshold Type	2
16	Threshold	2
18	Effect	2
20	Pre-Trigger	2

Figure 6-5: Trigger Type 5 and 6 Payload

Trigger Number: This is the number associated with this trigger. **icListen** can configure up to 5 trigger. Trigger numbers are a 0-based index (0 is trigger 1, 1 is trigger 2, etc).
Enabled: This flag is used to indicate if a trigger is enabled or not. If the value is 0, the trigger is not enabled, and if the value is 1, the trigger is enabled.
Signal Duration: The number of seconds a signal must be present for, to activate the trigger.
Event Hold: The number of seconds after a signal goes away that a trigger remains active for.
Min Frequency: This is the minimum frequency in the range checked by this trigger, set in Hz.
Max Frequency: This is the maximum frequency in the range checked by this trigger, set in Hz.
Threshold Type: This is a code describing what units the threshold is set in. Valid settings are 0 for dB re 1µPa and 1 for dB re 1V.
Threshold: This is the threshold against which data is checked, set in dB.
Effect: This is a bitmask which is used to determine what effects are asserted when a trigger goes active. See the following sub-section for a description of the bit fields in this mask.
Pre-Trigger: This is the number of seconds that we wish to assert our effect on prior to validation of the signal. Pre-triggering is not yet supported by icListen, so this value must be set to 0.

6.1.2 Trigger Effect Bitmask

The effects asserted when a trigger goes active are set using a bit field. The following table describes the meaning of each bit in this field.

Bit	Description
0	Drive signal output (requires device to be wired for signal output).
1	Transmit waveform data.
2	Transmit spectrum data.
3	Log waveform data.
4	Log spectrum data.
5	Transmit trigger start/stop message.
6 – 15	Reserved for future use.

*Effects for bits 0, 1, and 2 are available only on **icListen LF**.
*Effect for bit 5 is only available on **icListen HF** and **icListen AF**.

6.2 Setup Trigger

This message is used to configure triggers in **icListen**.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x53)	1
2	Payload Length (2+n)	2
4	Test Type	2
6	Type Specific Payload	n
6 + n	CRC	2

Figure 6-6: Setup Trigger Command

Test Type: This type determines the type of trigger setup payload that follows.
Type Specific Payload: This payload contains the actual setup parameters for this trigger.

The setup trigger message sets all of the parameters in **icListen** for a triggered event. See the "Enquire Trigger" command sub-sections for more details on the test type and type specific payload fields.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x53)	1
2	Payload Length (3+n)	2
4	Error Code	1
5	Test Type	2
7	Type Specific Payload	n
7 + n	CRC	2

Figure 6-7: Setup Trigger Response

Error Code: This code describes any errors that may have occurred during the trigger setup process. See the following sub-section for a description of the different error codes.
Test Type: This type determines the type of trigger setup payload that follows. It will be the same as the type sent in the setup command.
Type Specific Payload: This payload contains the actual setup parameters for this trigger. This payload will be the same as that sent in the command.

6.2.1 Trigger Setup Error Codes

When setting up and removing triggers from icListen, the response will always contain an 8 bit error code. The following table describes the possible errors that may be returned by the instrument:

Error Code	Description
0	No Error
1 – 31	If the error code is between 1 and 31, these bits may be used as a bit field to describe specific invalid settings: Bit 0: Low Limit Out of Range Bit 1: High Limit Out of Range Bit 2: Too Much Data to Log WAV files Bit 3: Too Much Data to Log FFT files Bit 4: Threshold Invalid
245	Trigger setup payload too small
246	Internal Message Queue Error
248	Invalid Threshold Type
249	Min Frequency > Max Frequency
250	Trigger Not Found
251	Invalid Test
252	Invalid Effect
253	Limit to number of triggers has been reached
254	System is out of memory and cannot service the request
255	Unit Not Ready

6.3 Remove Trigger

The remove trigger command is used to remove a single trigger from **icListen**. The payload of this message is different in **icListen LF**, from the other **icListen** models.

For **icListen AF** and **icListen HF**, the “Setup Trigger” command may also be used to remove individual triggers, by setting the trigger type to “Remove Trigger”.

6.3.1 icListen LF Remove Trigger Command

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x54)	1
2	Payload Length (2+n)	2
4	Test Type	2
6	Type Specific Payload	n
6 + n	CRC	2

Figure 6-8: icListen LF Remove Trigger Command

Test Type: This type determines the type of trigger setup payload that follows.
Type Specific Payload: This payload contains the actual setup parameters for this trigger.

When removing triggers individually from **icListen LF**, the exact configuration used when setting up the trigger in the first place must be sent to remove that trigger. The "Clear Triggers" command may also be used to remove triggers, if all triggers are to be removed instead of individual ones.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x53)	1
2	Payload Length (3+n)	2
4	Error Code	1
5	Test Type	2
7	Type Specific Payload	n
7 + n	CRC	2

Figure 6-9: icListen LF Remove Trigger Response

Error Code: This code describes any errors that may have occurred during the trigger removal process. See the error code sub-section under the "Setup Trigger" command for a description of the different error codes.
Test Type: This type determines the type of trigger setup payload that follows. It will be the same as the type sent in the remove trigger command
Type Specific Payload: This payload contains the actual setup parameters for the removed trigger. This payload will be the same as that sent in the command.

6.3.2 icListen AF / HF Remove Trigger Command

The “Setup Trigger” command may also be used to remove a trigger, by setting the trigger type to “Remove Trigger”.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x54)	1
2	Payload Length (4)	2
4	Test Type (4)	2
6	Trigger Number	1
7	Reserved	1
8	CRC	2

Figure 6-10: Remove Trigger Command

Test Type: This type determines the type of trigger setup payload that follows. For this command the type must always be set to 4 (Remove Trigger).
Trigger Number: This is the number associated with this trigger. **icListen** can configure up to 5 triggers. Trigger numbers are a 0-based index (0 is trigger 1, 1 is trigger 2, etc).
Reserved: This field is reserved, and should always be set to 0.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x54)	1
2	Payload Length (5)	2
4	Error Code	1
5	Test Type (4)	2
7	Trigger Number	1
8	Reserved	1
9	CRC	2

Figure 6-11: Remove Trigger Response

Error Code: This code describes any errors that may have occurred during the trigger removal process. See the error code sub-section under the “Setup Trigger” command for a description of the different error codes.
Test Type: This type determines the type of trigger setup payload that follows. For this command the type must always be set to 4 (Remove Trigger).
Trigger Number: This is the number associated with this trigger. **icListen** can configure up to 5 triggers. Trigger numbers are a 0-based index (0 is trigger 1, 1 is trigger 2, etc).
Reserved: This field is reserved, and should always be set to 0.

6.4 Clear Triggers

This command may be used to remove all configured triggers from **icListen**.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x55)	1
2	Payload Length (0)	2
8	CRC	2

Figure 6-12: Clear Triggers Command

No payload is required for this command.

Byte Offset:	Field:	# Bytes:
0	Sync (0x2A)	1
1	Message Type (0x55)	1
2	Payload Length (1)	2
4	Error Code	1
8	CRC	2

Figure 6-13: Clear Triggers Response

Error Code: This code describes any errors that may have occurred during the trigger removal process. See the error code sub-section under the “Setup Trigger” command for a description of the different error codes.

7 Cyclic Redundancy Check (CRC)

At the end of every **icListen** message there is a 16 bit CRC, which is calculated on all of the bytes in the message (including the sync and command characters). When checking the CRC, the calculation is performed on all bytes in the message (including the 2 bytes used for the CRC). If the CRC is correct, the calculated value will be 0. This ensures that the command was not modified, through error or attack, during the transmission from **icListen** to the requester, or vice versa.

A C code example for calculating the CRC, and the look-up table it uses, can be found in *Appendix B*.

8 Start-up Procedure

A typical start-up procedure contains the following commands:

Enquire Device
Set Time
Enquire Job Setup (Optional)
Job Setup

The *Enquire Device* command returns the status details of the connected **icListen**. This verifies that the correct unit and firmware version is connected.

The *Set Time* command is next. This ensures that the device is set to a known time for file logging.

The *Enquire Job Setup* command is optional during start-up. This command returns the current configuration of the **icListen**. This can be skipped if a *Job Setup* command with the desired configuration is sent.

The *Job Setup* command is used to set the **icListen** to a desired configuration. This is used during start-up so the **icListen** is performing the appropriate tasks during operation.

See the *Command Compatibility* section for a complete list of the commands recognized by each **icListen** model and firmware version.

9 Data Collection

For data collection operations over the command and control channel, the only required command is the *Collect Data* command. Using the scan mask, the type of data returned is set. The device also must be configured to collect the requested data, in order for it to be made available.

To acquire data over the streaming data channels (available on icListen AF and icListen HF), please refer to the *icListen Streaming Telemetry* document.

10 File Retrieval

To retrieve a logged file from an **icListen LF** the following commands are required:

Directory Listing
File Size
Read File
Collect Data

First the *Directory Listing* command is required to locate the desired file.

Once the desired file is located the *File Size* command is used to determine the size of the file.

After the size is known the *Read File* and *Collect Data* commands are alternated to request the data (in chunks of a maximum size of 4096 bytes), and to receive the data respectively. The *Enquire File System* command may be used between the *Read File* and *Collect Data* commands, in order to determine if the file read request has been completed and data is ready for collecting (the request is completed if the UUID of the request no longer appears in the *Enquire File System* response).

Data from an **icListen HF** and **icListen AF** is retrieved using Lucy, FTP, SFTP, SCP, or via the webserver. FTP and SFTP can be done using a program such as FileZilla. For SFTP and SCP, the username is "icListen", the port is 22, and by default there is no password. FTP requires no user name or password. Any modern web browser may be used to retrieve data from the webserver.

Firmware Differences:
-HF versions before 2.0 (Release 23) do not support FTP

Appendix A – Example Enquire Command Setup

```
/* Max message reply length is 6706 */
#define MAX_PAYLOAD_LENGTH 6700

/* Message format */
typedef struct
{
      /* Sync character */
      char Sync;
      /* Command Character */
      char Command;
      /* Length of the data payload */
      char PayloadLength[2];
      /* Contains the data for the command (if applicable) */
      char Payload[MAX_PAYLOAD_LENGTH];
      /* CRC used to verify message validity */
      char Crc[2];
} T_MESSAGE;

/*********************************************************
An example function for preparing an enquire command
ls_byte returns the least significant byte of a short.
ms_byte returns the most significant byte of a short.
*/
int SendEnquireCommand(void)
{
      T_MESSAGE Msg;
      ushort Crc,
      DataLength;

      /* Setup the Enquire message */
      /* '*' is the icListen Sync character */
      Msg.Sync = '*';
      /* 'E' is the command character for enquire */
      Msg.Command = 'E';
      /* Length of payload = 0 for enquire */
      DataLength = 0;
      /* The least significant byte of the payload length */
      Msg.PayloadLength[0] = ls_byte(DataLength);
      /* The high byte of the payload length, Enquire payload length is
      0 */
      Msg.PayloadLength[1] = ms_byte(DataLength);
      /* The payload of an enquire command is empty */
      /* Calculate the CRC */
```

```
        Crc = CalculateTxCrc(&Msg);
        /* Insert CRC into Message */
        Msg.Crc[0] = ls_byte(Crc);
        Msg.Crc[1] = ms_byte(Crc);

        /* Send message via serial or ethernet link */
        SendMessage(&Msg);
}

/* CRC functions are in Appendix B */
```

Appendix B – CRC Code

```
/******************************************************************
CalculateTxCrc:

Calculates the Tx message checksum.

*/
ushort CalculateTxCrc( T_MESSAGE *Msg )
{
      ushort checksum;
      int i;

      checksum = 0;

      checksum = crc_update(Msg->Sync, checksum);
      checksum = crc_update(Msg->Command, checksum);
      checksum = crc_update(ls_byte(Msg->PayloadLength), checksum);
      checksum = crc_update(ms_byte(Msg->PayloadLength), checksum);
      for(i = 0; i < Msg->PayloadLength; i++)
      {
            checksum = crc_update ( Msg->Payload[i], checksum );
      }
      return ( checksum );
}

/******************************************************************
Calculate CRC-16 using table-lookup method

*/
unsigned short crc_update( char data, unsigned short accum )
{
      unsigned char comb_value;

      comb_value = ( accum&0x0ff ) ^ (unsigned char)data;
      accum = ( accum >> 8 ) ^ crc16_table[ comb_value ];
      return ( accum );
}
```

```
/*************************************************************
The CRC-16 lookup table.

*/
static unsigned short crc16_table[256] =
{
      0x0000, 0xc0c1, 0xc181, 0x0140, 0xc301, 0x03c0, 0x0280, 0xc241,
      0xc601, 0x06c0, 0x0780, 0xc741, 0x0500, 0xc5c1, 0xc481, 0x0440,
      0xcc01, 0x0cc0, 0x0d80, 0xcd41, 0x0f00, 0xcfc1, 0xce81, 0x0e40,
      0x0a00, 0xcac1, 0xcb81, 0x0b40, 0xc901, 0x09c0, 0x0880, 0xc841,
      0xd801, 0x18c0, 0x1980, 0xd941, 0x1b00, 0xdbc1, 0xda81, 0x1a40,
      0x1e00, 0xdec1, 0xdf81, 0x1f40, 0xdd01, 0x1dc0, 0x1c80, 0xdc41,
      0x1400, 0xd4c1, 0xd581, 0x1540, 0xd701, 0x17c0, 0x1680, 0xd641,
      0xd201, 0x12c0, 0x1380, 0xd341, 0x1100, 0xd1c1, 0xd081, 0x1040,
      0xf001, 0x30c0, 0x3180, 0xf141, 0x3300, 0xf3c1, 0xf281, 0x3240,
      0x3600, 0xf6c1, 0xf781, 0x3740, 0xf501, 0x35c0, 0x3480, 0xf441,
      0x3c00, 0xfcc1, 0xfd81, 0x3d40, 0xff01, 0x3fc0, 0x3e80, 0xfe41,
      0xfa01, 0x3ac0, 0x3b80, 0xfb41, 0x3900, 0xf9c1, 0xf881, 0x3840,
      0x2800, 0xe8c1, 0xe981, 0x2940, 0xeb01, 0x2bc0, 0x2a80, 0xea41,
      0xee01, 0x2ec0, 0x2f80, 0xef41, 0x2d00, 0xedc1, 0xec81, 0x2c40,
      0xe401, 0x24c0, 0x2580, 0xe541, 0x2700, 0xe7c1, 0xe681, 0x2640,
      0x2200, 0xe2c1, 0xe381, 0x2340, 0xe101, 0x21c0, 0x2080, 0xe041,
      0xa001, 0x60c0, 0x6180, 0xa141, 0x6300, 0xa3c1, 0xa281, 0x6240,
      0x6600, 0xa6c1, 0xa781, 0x6740, 0xa501, 0x65c0, 0x6480, 0xa441,
      0x6c00, 0xacc1, 0xad81, 0x6d40, 0xaf01, 0x6fc0, 0x6e80, 0xae41,
      0xaa01, 0x6ac0, 0x6b80, 0xab41, 0x6900, 0xa9c1, 0xa881, 0x6840,
      0x7800, 0xb8c1, 0xb981, 0x7940, 0xbb01, 0x7bc0, 0x7a80, 0xba41,
      0xbe01, 0x7ec0, 0x7f80, 0xbf41, 0x7d00, 0xbdc1, 0xbc81, 0x7c40,
      0xb401, 0x74c0, 0x7580, 0xb541, 0x7700, 0xb7c1, 0xb681, 0x7640,
      0x7200, 0xb2c1, 0xb381, 0x7340, 0xb101, 0x71c0, 0x7080, 0xb041,
      0x5000, 0x90c1, 0x9181, 0x5140, 0x9301, 0x53c0, 0x5280, 0x9241,
      0x9601, 0x56c0, 0x5780, 0x9741, 0x5500, 0x95c1, 0x9481, 0x5440,
      0x9c01, 0x5cc0, 0x5d80, 0x9d41, 0x5f00, 0x9fc1, 0x9e81, 0x5e40,
      0x5a00, 0x9ac1, 0x9b81, 0x5b40, 0x9901, 0x59c0, 0x5880, 0x9841,
      0x8801, 0x48c0, 0x4980, 0x8941, 0x4b00, 0x8bc1, 0x8a81, 0x4a40,
      0x4e00, 0x8ec1, 0x8f81, 0x4f40, 0x8d01, 0x4dc0, 0x4c80, 0x8c41,
      0x4400, 0x84c1, 0x8581, 0x4540, 0x8701, 0x47c0, 0x4680, 0x8641,
      0x8201, 0x42c0, 0x4380, 0x8341, 0x4100, 0x81c1, 0x8081, 0x4040,
};
```

Appendix C – Miscellaneous Response Codes

Device Codes:

Code	Device
0x01	icTalk LF
0x02	icListen LF
0x03	Standalone Guest Sensor
0x04	icTalk HF
0x05	icListen HF
0x07	icListen AF
0x41	icListen MF

Device Status Codes:

Code	Status
0	Ready
1	Data Acquisition Not Ready
2	Device Not Ready
3	Start-up Configuration Failed
4	Baud Rate Reconfiguring
5	File System Fault

Link Type Codes:

Code	Status
0	Unknown
1	RS232
2	USB
3	Ethernet
	RS422
	RS485

Battery Status Codes:

Code	Status
0	No data available
1	Battery is charging
2	Battery is discharging (no external power)
3	Battery charge state is not changing (externally powered, but not charging)

Sync Status Codes:

Code	Status
0	Sync in/out is disabled
1	Sync out
2	Sync out (with encoded time of day)
3	Sync in/no PPS detected (falling edge)
4	Sync in/syncing to PPS (falling edge)
5	Sync in/synced to PPS (falling edge)
6	Sync in/no PPS detected (rising edge)
7	Sync in/syncing to PPS (rising edge)
8	Sync in/synced to PPS (rising edge)

System Time Status Codes:

Code	Status
0	Time value has not been set
1	Time was manually set
2	Time set from RTC at start-up
3	Time was set from file system at start-up
4	Time was set from NTP
5	Time was set from GPS
6	Time was set from PTP
128	Time set from PPS encoded time (from unset source)
129	Time set from PPS encoded time (from manually set source)
130	Time set from PPS encoded time (from source set by RTC at start-up)
131	Time set from PPS encoded time (from source set by file system at start-up)
132	Time set from PPS encoded time (from source set by NTP)
133	Time set from PPS encoded time (from source set by GPS)
134	Time set from PPS encoded time (from source set by PTP)

Modes of Operation (LF):

Feature	Format	Code																																			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
FFT Transmission	Continuous	🟩			🟩		🟩		🟩	🟩	🟩																	🟩	🟩	🟩	🟩	🟩	🟩				
	Triggered													🟩			🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩										
Wave Transmission	Continuous		🟩			🟩			🟩	🟩	🟩	🟩					🟩					🟩	🟩	🟩						🟩				🟩			
	Triggered														🟩			🟩							🟩	🟩	🟩	🟩			🟩	🟩	🟩		🟩	🟩	🟩
Internal Logging	WAV			🟩	🟩	🟩				🟩									🟩			🟩			🟩						🟩				🟩		
	FFT						🟩	🟩			🟩	🟩								🟩			🟩			🟩						🟩				🟩	
	Triggered															🟩					🟩			🟩			🟩		🟩	🟩			🟩	🟩			🟩

*Shaded cells indicate an active feature/format

*Triggered logs can be either FFT or WAV files, depending on the active triggers

icListen FFT Processing Types:

Code	Type
0	Mean Average
1	Overlap
2	Peak Value
3	Exponential Moving Average
4	Overlap
5	Peak Value with Overlap
6	Exponential Moving Average with Overlap

icListen Job Setup Types:

Code	Description
0 - 11	Reserved
12	icListen LF setup
13	Reserved
14	icListen HF waveform data setup – superseded by type 20 icTalk LF setup (v1.1)
15	icListen HF spectrum data setup – superseded by type 20
16	icListen HF duty cycle setup (Release 15 and down) – superseded by type 20
17	icTalk setup (v1.2 and up)
18	icListen HF log start time setup (Release 16 and up) – superseded by type 20
19	icListen HF duty cycle setup (Release 16 and up) – superseded by type 20
20	Tag based setup (icListen HF Release 23 and up, and icListen AF)
65535	Error response

Job Setup Error Codes:

Code	Description
0	Unrecognized setup type
1 – 15	If the error code is between 1 and 15, these bits may be used as a bit field to describe specific invalid settings: Bit 0: Invalid Gain Bit 1: Invalid Sample Rate Bit 2: Invalid Mode of Operation(LF), Invalid Waveform Data Format(HF) Bit 3: Invalid FFT Type or Parameters Bit 4: Invalid Logging Setup
65523	Unrecognized setup tag
65524	Invalid tag length
65525	Check tags for more details. See the following table for tag error codes.
65526	Invalid time
65527	Invalid (non-0) reserved bytes
65528	Invalid duty cycle settings
65529	Communications to DSP failed
65530	Memory allocation failed
65531	Message payload too small for setup
65532	Reserved – this code is not in use
65533	Setup is incompatible with one or more triggers, as it could generate too much data to log (**icListen LF** only)
65534	Too much data to log (reduce waveform sample rate, or change spectrum processing parameters to remedy this.)
65535	Unit is not ready to accept setups

Setup Tag Error Codes:

Code	Description
0	No error (success).
1	Value is not valid (generic error).
2	Value is in conflict with another setup parameter (the other parameter will also be marked with this error code).
3	Value was not checked for validity. This may occur is a value which this parameter depends on was not valid.
4	Value is too high.
5	Value is too low.

icListen LF File System Error Codes:

Code	Description
0	No Error
1	Invalid Volume Requested
2	File Open Failed
3	File Seek Failed
4	File Write Failed
5	File Read Failed
6	Reached End of File
7	File Close Failed
8	File Delete Failed
9	Directory Listing Failed
10	Copy Failed
11	Rename Failed
12	File System Access Error
13	File System Idle
14	File System Busy for USB
15	Volume Full
16	Corrupt Sector

icListen Waveform Data Formats:

Code	Description
2	16 bit big endian
3	24 bit big endian
4	24 bit big endian data packed into 32 bits
130	16 bit little endian
131	24 bit little endian
132	24 bit little endian data packed into 32 bits

icListen EPOCH Trigger Effect Bit-field Values:

Bit	Effect Description
0	Voltage signal output
1	Transmit WAV data (must issue Collect command to receive this data)
2	Transmit FFT data (must issue Collect command to receive this data)
3	Log WAV data
4	Log FFT data
5	Epoch Message
6 - 15	Reserved

icListen EPOCH Trigger Error Codes:

Code	Description
0	No Error
1 - 31	If the error code is between 1 and 31, these bits may be used as a bit field to describe specific invalid settings: Bit 0: Low Limit Out Of Range Bit 1: High Limit Out Of Range Bit 2: Too Much Data To Log WAV files Bit 3: Too Much Data To Log FFT files Bit 4: Threshold Invalid
246	Internal Message Queue Error
248	Invalid Threshold Type
249	Min Frequency > Max Frequency
250	Trigger Not Found
251	Invalid Test
252	Invalid Effect
253	Limit to number of triggers has been reached
254	System is out of memory and cannot service the request
255	Unit Not Ready

Appendix D – Device Specific Settings

icListen LF Settings:

Setting	Accepted Values
Gain (dB)	0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72
Sample Rate (Hz)	250, 500, 1000, 2000, 4000, 8000, 16000
Mode of Operation	0 - 35
FFT Processing Type	0, 1, 2, 3, 4, 5, 6
Mean (Type 0) Parameters	P1: 1 - 65535 P2: 0
Overlap (Type 1) Parameters	P1, P2: 1 - 1024 P2 >= P1, P1/P2 rounded up to even value
Peak (Type 2) Parameters	P1: 0 - 65535 P2: 0
Exponential (Type 3) Parameters	P1, P2: 1 - 65535
Mean with Overlap (Type 4) Parameters	P1: 1 - 65535 P2: 0 P3, P4: 1 - 1024 P4 >= P3, P3/P4 rounded up to even value
Peak with Overlap (Type 5) Parameters	P1: 0 - 65535 P2: 0 P3, P4: 1 - 1024 P4 >= P3, P3/P4 rounded up to even value
Exponential with Overlap (Type 6) Parameters	P1, P2: 1 - 65535 P3, P4: 1 - 1024 P4 >= P3, P3/P4 rounded up to even value
Baud Rates	1200, 2400, 4800, 9600 19200, 115200, 256000, 1250000 *Baud rates below 19200 not available prior to v2.5.02
Epoch Trigger Types	0, 1

*See *Appendix C* for details on FFT processing type and mode of operation

*See “Job Setup” command section for details on FFT processing parameters

icListen HF Settings:

Setting	Accepted Values
Gain (dB)	0, 6, 12, 18, 24, 30, 36, 42, 48 *Prior to v1.3 (Release 14), 0dB is always used
Waveform Sample Rate (Samples/sec)	1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000, 256000, 512000 *Prior to v2.1 (Release 28) 512000 Sample/sec data was only available as 16bit.
Logging Mode	0 = None 1 = Enabled 2 = On Epoch Trigger
Log length (minutes)	1-255 *Limited to 1GB file size
Data Format (code describing bit-depth and endianness in streaming data headers and deprecated waveform setup messages)	2, 3, 4, 130, 131, 132 *2, 3,4 and 132 are available only for streaming live data *Prior to v1.3 (Release 14) type 24bit big endian (3) is always used for streamed data, and 24bit little endian (131) for WAV log data.
Bit Depth	16, 24, 32 (24bits packed into 32)
Endianness	0 (big endian), 1 (little endian) *Logged data must be little endian
Spectrum Sample Rate (Samples/sec)	1000, 2000, 4000, 8000, 16000, 32000, 64000, 128000, 256000, 512000
FFT Processing Type	4, 5, 6
Points Per FFT	1024
Samples Between FFTs	512
Accumulations	1-65535
Filtered Weighting Factor	1-65535
Spectrum Reference Level (dBV)	-120 to -180
Duty Cycle Time (minutes)	0-131070
Duty Cycle Active Time (minutes)	0-65535
Duty Cycle Idle Time (minutes) -only applicable to deprecated duty cycle setup messages	0-65535
Duty Cycle Start Delay (minutes) -only applicable to v1.4(Release 15) and lower	(-2^{31}) -> $(2^{31}-1)$
Logging Start Time (Seconds since 1970)	0 -> $(2^{31}-1)$ *Negative values will cause logging to start immediately
Baud Rates	19200, 115200, 230400, 460800, 921600
Epoch Trigger Types	5, 6
Number of Epoch Triggers Available	5 (trigger numbers 0 - 4)

*See *Appendix C* for details on FFT processing type and Waveform Data Formats

icListen AF Settings:

Setting	Accepted Values
Gain (dB)	0, 6, 12, 18, 24, 30, 36, 42, 48
Waveform Sample Rate (Samples/sec)	1000, 2000, 4000, 8000, 16000, 32000
Logging Mode	0 = None 1 = Enabled 2 = On Epoch Trigger
Log length (minutes)	1-255 *Limited to 1GB file size
Bit Depth	16, 24, 32 (24bits packed into 32)
Endianness	0 (big endian), 1 (little endian) *Logged data must be little endian
Spectrum Sample Rate (Samples/sec)	1000, 2000, 4000, 8000, 16000, 32000
FFT Processing Type	4, 5, 6
Points Per FFT	1024
Samples Between FFTs	512
Accumulations	1-65535
Filtered Weighting Factor	1-65535
Spectrum Reference Level (dBV)	-120 to -180
Duty Cycle Time (minutes)	0-131070
Duty Cycle Active Time (minutes)	0-65535
Logging Start Time (Seconds since 1970)	0 -> (2^{31}-1) *Negative values will cause logging to start immediately
Baud Rates	19200, 115200, 230400, 460800, 921600
Epoch Trigger Types	5, 6
Number of Epoch Triggers Available	5 (trigger numbers 0 - 4)

*See *Appendix C* for details on FFT processing type and Waveform Data Formats

Glossary

CRC	Cyclic Redundancy Check
dB	Decibels
FFT	Fast Fourier Transform
Hz	Hertz
µPa	micro Pascals
UTC	Coordinated Universal Time
UUID	Universally Unique Identifier
V	Volts