AquaSeNT ANE-00009 OFDM Modem

User Manual

Firmware Revisions

This manual applies directly to instruments that have the firmware revision 5.11.01.04

AquaSeNT
11 Professional Park Road
Storrs, CT 06268, USA
www.aquasent.com
Version 1.2

Notices

Revisions

The manual's revision number indicates its current edition. The revision number changes when a new edition is printed (minor corrections and updates that are incorporated at reprint do not cause the revision number to change).

Version 1.0 :	July 2015, The first revision
Version 1.1 :	May 2016, Revised to reflect the changes in the modem firmware
Version 1.2 :	Dec 2016, Revised to reflect the changes in the modem firmware

Contents

1 Unpacking and Preparation

This chapter describes how to set up and start the AquaSeNT ANE-00009 Acoustic Modem.

1.1 Checking the Shipment

After you receive the ANE-00009, carry out the following checks during unpacking according to the procedure below:

Step 1. Check that the shock-absorbing material used to package the modem and accessories has not been damaged.

Step 2. Check the packaged items supplied with the modem for any damage or defects.

Step 3. Check that all packaged items supplied with the modem have been provided as per the specified options (refer to Table 1).

Step 4. After checking, contact the AquaSeNT sales and service office if one of the following applies:

- The shocking-absorbing material used to package the modem and accessories has been damaged.
- A packaged item supplied with the modem is missing.
- A fault has been detected in the subsequent operation check of the modem.

Table 1: Items packaged with the ANE-00009

Name	Part Number	Qty
Modem	ANE-00009	1
Hydrophones		2
Power Adapter	XST-70W-002	1
Serial RS232 Cable		1
USB-to-Serial Converter	ICUSB232V2	1
USB flash drive	AquaSeNT	1

1.2 Preparation before Use

Confirm that the power supplied to the ANE-00009 is 16V with 0.5A current capacity. If the packaged power adapter is used, please confirm that the voltage selection switch is pointed to the 16V. To do this, the user can put the red voltage selection switch to the left most position and then move it one position to the right.

Confirm that the two packaged hydrophones are connected to the modem. The hydrophones should be submerged in the water.

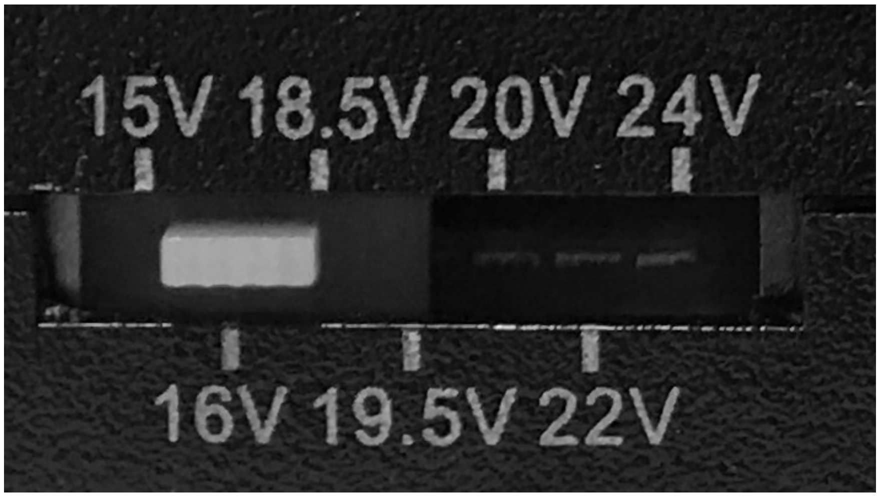

Figure 1: Power adapter voltage settings

! → The ANE-00009 is designed for laboratory use only. Its housing is not water-proof. Do NOT put the ANE-00009 into the water.

1.3 Starting the Acoustic Modem

This section describes how to start the modem for the first time after unpacking. The modem is in default settings. It is not applicable after user changes the modem settings.

Before powering on the modem, the user needs to set up a serial port terminal to communicate with the modem. If this is the first time the modem is powered on after unpacking, the serial baud rate uses the default value of 9600. The terminal should also be set to send a carrage return followed by a line feed as line end.

The ANE-00009 does not have a power switch. When the user connects the power adapter plug into the modem's power port, the modem is turned on.

The ANE-00009 has two LEDs to inidicate its status. One is labelled as power and the other is labelled as communication. When the modem is in the initialization status after power up, the two LEDs are red. After initialization is done, the power LED turns green and communication LED turns off. The starting messages are printed on the terminal.

```
Modem started
```

```
MID:1
RTC:2000-01-01T00:00:05
TXPWR:-10
RXGAIN:0
UIMODE:CMD
CMDTERM:CRLF(In),CRLF(Out)
BIV:5.11.01.04
```

Now the modem is in normal operation. The user can follow the rest of the manual to do the communication and configuration with the modem.

When the modem is receiving acoustic packets, the communication LED turns green; When it is transmitting, the LED turns red. The power LED is always green during normal opertions.

2 Overview

2.1 Product Introduction

The AquaSeNT ANE-00009 is an acoustic modem for underwater communications. The ANE-00009 is designed for encouraging underwater networking and communication research in the laboratory.

The ANE-00009 is used for acoustic communication over a frequency range of 21 kHz to 27 kHz. The sampling rate is 96 kHz and resolution is 16 bit. It employs Orthogonal Frequency Division Multiplexing (OFDM) modulation. It uses two hydrophones to receive acoustic signal and one of them to transmit. The ANE-00009 also supports waveform playing.

The ANE-00009 uses one DB9 Serial RS232 cable to communicate with the host device. It needs only the TX and RX wires of RS232. No other wires are needed. It supports standard baud rates from 4800 bps to 230400 bps.

Every modem has an ID number. The ID is used to identify the source and destination during commnucation. The user can assign the modem with a preferred ID. The valid ID value is from 2 to 255. 0 is reserved for broadcasting.

2.2 Acoustic Packet

The modem uses packets to transfer data. A packet starts with a preamble and several data blocks may follow the preamble. Guard intervals are inserted between the preamble and the first data block and between two data blocks. The packet structure is shown Figure 2.

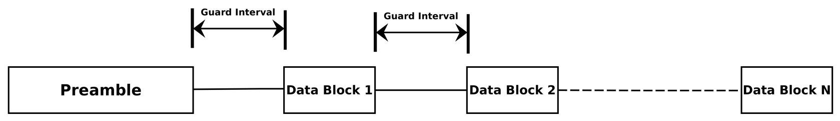

Figure 2: Acoustic packet structure

The preamble synchronizes transmission timing between two or more modems. It carries some packet informations as well. With the information in the preamble, the modem knows how to process the remaining data blocks, if there are any. The preamble carries three basic packet parameters:

- Source modem ID. This is the ID of the modem from which the packet is transmitted.
- Destination modem ID. This is the ID of the modem to which the packet is transmitted. The value 0 means this is a broadcasting packet.
- Packet type. The modem decides how to process the rest of the packet based on the type of the packet. The modem supports 4 types which are described in Table 2.

The ANE-00009 implements a new scheme for preamble detection that improves the performance of acoustic communication with a single transducer. To be compatible with ANG-02009 (AMN-OFDM-13A), the ANE-00009 supports the preamble of ANG-02009 (AMN-OFDM-13A) as well. The new preamble is labelled as mode 10 while the previous one as mode 0 throughout the modem firmwares and the modem documentations.

Table 2: Packet types

Type	Preamble Mode	Payload	Number of Data Blocks	Payload Size
0	Table 3	Data	16	Table 3
1	0 / 10	Data (short)	None	6
64	0 / 10	Waveform	Replaced	N/A
65	0 / 10	Range information	None	N/A

- Packet of type 0 is used for data transmission. Data are carried on the data blocks. One packet can have at most 16 data blocks.
- Packet of type 1 is also used for data transmission. If there are no more than 6 bytes, the user can send them in a type 1 packet. The

data are carried in the preamble followed by no data blocks. This is useful to send an RTS/CTS packet.

- Packet of type 64 is used for waveform transmission. The user can upload a pass band waveform to the modem and transmit it. When another modem receives the type 64 packet, it removes the preamble and saves the user waveform in a file.

- Packet of type 65 is used for the ranging function of the modem. The distance between two modems can be measured using packets of this type.

The preamble carries additional parameters if it has one or more block blocks:

- Packet mode. The packet mode reflects how much data one data block can carry. The modem achieves a higer data rate when it transmits/receives packets of a higher data mode. Table 3 shows the number of bytes of different modes.

- Guard interval. Guard intervals are used to ensure the preamble and data blocks do not interfere with the one following it when acoustic channels are complex. The modem supports 50 ms, 100 ms and 150 ms guard intervals. The user can choose shorter guard intervals to achieve a higher data rate in good environments or choose longer guard intervals to achieve robust transmission in challenging environments.

Table 3: Packet modes.

Mode	Pramble Mode	Number of Bytes per Block	Maximum Number of Bytes per Packet
1	0	38	608
2	0	80	1280
3	0	122	1952
4	0	164	2624
5	0	248	3968
11	10	8	128
12	10	18	288

The user can set the packet parameters in the transmitting commands or in the modem configuration registers. Detailed information can be found in Chapter 3 and Chapter 4.

2.3 User Interface

The user interacts with the modem using a text-based interface via the serial port. The modem has two User Interface (UI) modes.

Data Mode: In Data Mode, all UI input data received by the modem are transmitted. UI input data are put into an input buffer. If there is no data from UI input during a time interval, the modem will put the bufferred data in a packet and transmit it. The time interval is called packet gap. It prevents the modem from waiting indefinitely.
The packets built in UI Data Mode are of type 0. The other parameters are retrieved from the modem configuration registers: `PDST` and `PMODE`. The user can change these configuration registers by using the `HHCRW` command when the modem UI is in Command Mode.
When the data in the input buffer reaches the packet size limit set by packet mode before a packet gap appears, the modem will transmit the data within the size limit in a packet and continue to process the rest of the data.
The user can use an escape string to switch from Data Mode to Command Mode. The escape string must be followed by a packet gap with no other data between them. The default escape string is "`+++A`". The user can change it to any of 4 regular characters described in Section 3.2.

Command Mode: In Command Mode, UI data exchanged between the modem and host follow the NMEA 183 standard. The data are divided and parsed as modem commands. With modem commands, the user can transmit packets of all types, access modem configuration registers and request file system service. Chapter 3, Chapter 4 and Chapter 5 describe how to use modem commands to operate the modem in detail.
The user can use the `HHCTD` command to switch from Command Mode to Data Mode.

3 Modem Command Reference

This chapter describes the modem command reference for the OFDM modem.

3.1 Notational Conventions

This section describes the rules to read the descriptions of the commands.

A part with the heading "Syntax" describes the syntax of a command exchanged between the modem and the host. A syntax has a command part. If there is one or more than one parameter part, a comma is used to separate the command part and the first parameter part, as well as two parameter parts.

The definitions of symbols used in the syntax are as follows.

<> :	Characters enclosed are necessary parameters for the command.
[ ] :	Characters enclosed are optional parameters for the command.
{} :	One of the items enclosed must be selected. Individual items are separated by a vertical bar (|).

3.2 Command Format and Special Characters

A valid command starts with a "$", followed by the command name, optional parameters, an optional checksum and ends with command terminators.

Syntax : `$<COMMAND>[,OPTION,[OPTION,[...]]][,CHECKSUM]<terminators>`

The command names and options are in uppercase.
All commands start with a `$` symbol.
Each command consists of five characters. The first two identify the sender and divide all the commands into categories. There are two categories in this firmware version. `HH` category contains the commands from the host device (e.g. a computer) to the modem. `MM` category contains the commands from the modem to the host. The next three characters indicate the type of the command, which will be described in detail later in this chapter.
Options vary with different commands and will be described for each command.
A checksum can be attached to a command. The checksum is indicated by the character `*`. It is the XOR of all characters between `$` and `*`. For the `HH` commands, if the host appends a checksum the modem will do the check. For the `MM` commands, the user can choose whether to append the checksum by setting the configuration register `MMCHK`.
Terminators indicate the end of a command. The `HH` commands share one `IN` terminator and `MM` commands share one `OUT` terminators. [1] The user can set the `IN` and the `OUT` terminators as `CR`, `LF`, `CRLF`, `LFCR` [2] by setting the configuration register `CMDTERM`.

There are several specicial characters that are limited in use.

[1] The `IN` and `OUT` are command directions in the modem's view.
[2] `CR` is carriage return, the ASCII character of value 13. `LF` is the line feed, the ASCII character of value 10.

`$` : `$` (ASCII 0x24) is reserved as the start of a command. It cannot appear in the remainning part of a command. If there are multiple `$`'s within a command string, the modem splits the string with the `$` and process each substring as a command.

`LF, CR` : `CR` (ASCII 0xD) and `LF` (ASCII 0xA) are reserved as terminators of a command. They indicate the end of a command.

`*` : `*` (ASCII 0x2A) is reserved for the checksum of a command. It indicates that the rest of the command is checksum.

`,` : `,` (ASCII 0x2C) is reseved to seperate command items.

`Backspace, DEL` : `Backspace` (ASCII 0x8) and `DEL` (ASCII 0x7F) delete the last character in the command buffer.

`Ctrl+U` : `Ctrl+U` clears the command buffer.

Except the special characters above, all other characters that have an ASCII value smaller than 0x20 (Space) are control characters and cannot appear in a command. Non-ASCII characters cannot appear in a command.

The maximum length of a single `HH` command is 2600 bytes without the terminators. A command longer than the maximum length will not be parsed and an error message will be reported. The `MM` commands do not have length limitation.

A special command: The escape string is the only command that can be used in both DATA mode and COMMAND mode. No matter which UI mode the modem is in, the escape string puts the modem into COMMAND mode. The escape string can be followed by correct input terminators.

→ Section 2.3

Detailed infomation can be found in Section 2.3 about the DATA/COMMAND mode of modem's user interface.

3.3 Commands from Modem to Host

This section describes the commands sent from the modem to the host device. These commands start with `MM` as described in Section 3.2.

MMERR: The modem sends this command to the host to report an error. If the error is caused by a command from the host to the modem, the command and its options are included in the `MMERR` command. The modem can report either the error code or a message that expains the error code. The user can choose which one to use by setting the configuration register `ERRFMT`. The user can find the description for the errors in the Appendix A.

Syntax : `$MMERR,[COMMAND,[OPTION,[OPTION, ... ]]]{Code|Message}<OUT TERM>`

Example : If the user sends the command below to the modem,

```
> $HHTXA,0,M1,0,hello$world\r\n
```

Because `$` is reserved, the modem splits it into two commands,

```
> $HHTXA,0,M1,0,hello
> $world\r\n
```

For the first commands, there are no terminators required; For the second command, the command name `world` is not valid, thus the modem sends two `MMERR` commands back to the host as the responses.

```
$MMERR,HHTXA,0x0016
$MMERR,0x0002
```

If the modem is configured to display error messages, the two `MMERR` commands are:

```
$MMERR,HHTXA,UI:IllegalChar/NoTerminators
$MMERR,UI:UnkownCommand
```

! $\rightarrow$ In this manual, the *Examples* describe the commands exchanged between the host and the modem. For clear illustration, the command from the host has a '>' followed by a space before itself while the command from the modem does not have. In real operation on the modem, the '>' followed by a space is not needed. For the rest part of the manual, if not specified, the commands in the *Examples* have suitable terminators attached to them without explicitly printed as `\r` and `\n`.

MMOKY: The modem sends this command to the host to report the result after executing an `HH` command successfully. The `HH` command and its option may be included in the `MMOKY` command. The result part varies with different `HH` commands and will be described later with each `HH` command.

Syntax : `$MMOKY[,COMMAND[,OPTION[,OPTION...]]][,result]<OUT TERM>`

Example : User sends the escape string,

```
> +++A
```

Modem sends back a `MMOKY` after it puts the modem in the COMMAND mode,

```
$MMOKY,+++A
```

MMRXD: The modem sends this command to report the received packet with the ASCII representation in HEX format. It lifts the limitation of the non-appearance of the special charactors and control charactors discussed in Sector 3.2.

Syntax : `$MMRXD,<SRC ID>,<DST ID>,<PACKET DATA><OUT TERM>`

Example : The modem receives a packet containing four characters "$123". The sender's modem ID is 99 and this is a broadcasting packet in which the destination modem ID is 0.

```
$MMRXD,99,0,24313233
```

MMRXA: The modem sends this command to report the received packet in ASCII format.

Syntax : `$MMRXA,<SRC ID>,<DST ID>,<PACKET DATA><OUT TERM>`

Example : The modem receives a packet containing three characters "123". The sender's modem ID is 99 and this is a broadcasting packet in which the destination modem ID is 0.

```
$MMRXA,99,0,313233
```

! $\rightarrow$ The user can choose to use `MMRXD` or `MMRXA` by setting the configuration register `RXFMT`. However, if the packets contains any special characters, control character or non-ASCII characters described in Section 3.2, the modem reports the packet with `MMRXD` no matter what the setting is in the register `RXFMT`.

MMTDN: The modem sends the `MMTDN` command to report the result of sending a packet. This command consists of a packet number and an error code. The packet number is assigned to the packet when it is registered in the `HHTXA` or `HHTXD` command. If the packet has been transmitted successfully, the error code is 0. The user can find the description for the error codes in Appendix A. The user can enable/disable the `MMTDN` by setting the configure register `TXFLG`.

Syntax : `$MMTDN,<ERROR>,<PACKET NUM>`

Example : The user sends a `HHTXA` command to the modem. The modem registers this command and generates a packet with number 12. After this packet has been transmitted, the modem reports a `MMTDN` packet.

```
$MMTDN,0,12
```

3.4 Commands from Host to Modem

This section describes the commands sent from the host device to the modem. These commands start with `HH` as described in Section 3.2.

HHHLP: The host sends this command to check the syntax of the `HH` commands. The user can find the detailed expanation for the syntax of the commands in this section.

Syntax : `$HHHLP[,{HHCTD|HHTXD|HHTXA|HHTXW|HHCRW|HHCRR|HHFSA}]<IN TERM>`

Example : If `HHHLP` is sent without option, it reports the syntax of itself.

```
> $HHHLP
$MMOKY,HHHLP,HHHLP[,{HHCTD|HHTXD|HHTXA|HHTXW|HHCRW|HHCRR|HHFSA}]
```

If `HHHLP` is sent with an `HH` command specified, it reports the syntax of that command.

```
> $HHHLP,HHCRW
$MMOKY,HHHLP,HHCRW,<REG>[,FIELD][,VAL]
```

HHCTD: The host sends this command to put the modem into `Data Mode`. Afterwards, the host can use the escape string to put the modem back into `Command Mode`.

Syntax : `$HHCTD<IN TERM>`

Example : Put the modem into DATA mode.

```
> $HHCTD
$MMOKY,HHCTD
```

HHTXD: The host sends this command to have the modem transmit a packet. The packet payload in this command is in hexadecimal format. The command must specify the destination modem ID, packet type and mode, and packet amplitude level to send out a packet.

- The destination modem ID specifies which modem will receive this packet. It will be encoded into the packet preample. The valid `modem`

ID values range from 1 to 255. If the destination modem ID is 0, the modem sends out a broadcasting packet.

- The packet type and mode specify the parameters of the packet. Detailed infomation about the type and mode of the OFDM packet can be found in Section 2.2. By default, the packet type is 0. User can specify the packet mode by using number 1 - 5, 11 - 12. If user inputs 0, the value in the modem configuration register `PMODE` will be used as packet mode. If user wants to send packets of other types rather than type 0, a type number following a 'T' will do. In this case the default preamble mode is 0. The user can explicitly specify the preamble mode by adding a 'M' followed by the preamble mode number.
- The packet power level controls the amplitude of the digital signal of the packet. The unit is represented as a percentage. The value 100 means full amplitude. The lowest value is 10 which means 10% of full amplitude. The value 0 is acceptable and means full amplitude. The user can use this field to adjust the transmitting power for an individual packet.

! $\rightarrow$ The packet amplitude level option in the `HHTXD` or `HHTXA` command differs from the modem configuration register `TXPWR`. The packet power level adjusts the amplitude of digital signal while the `TXPWR` adjusts the amplitude of the analog signal.

When the modem processes the `HHTXD` successfully it returns a `MMOKY` command to the host. This `MMOKY` command consists of its commmand name, the name of `HHTXD` and a packet number. The packet number is assigned during the processing of the `HHTXD` command. If the user enables the `MMTDN` command, the modem will send an `MMTDN` command containing this packet number when the packet has been transmitted.

Syntax :
```
$HHTXD[[,<DST>,{0|T<TYPE>[M<PREAMBLE MODE>]|<MODE>},
<PAL>],<HEX>]<IN TERM>
```

Return :
```
$MMOKY,HHTXD,<PACKET NUM><OUT TERM>
$MMERR,HHTXD,<ERROR><OUT TERM>
```

The maximum payload length for different modes are listed in Table 3. The hexadecimal repressentation of the packet payload doubles its length. A single `HHTXD` command cannot acommodate such a long payload. The `HHTXD` command supports combining the payloads from multiple `HHTXD` commands into one packet. The combining process involes three forms of `HHTXD`:

1. `$HHTXD,<HEX><IN TERM>`

2. `$HHTXD,<DEST>,{0|T<TYPE>[M<PREAMBLE MODE>]|<MODE>},`
   `<PAL>[,<HEX>]<IN TERM>`
3. `$HHTXD<IN TERM>`

Form 1 of `HHTXD` is used to put the payload into an internal buffer of modem. Form 2 is used to specify the parameters of the packet and send it out. The payload is not neccessary in form 2. Form 3 is used to clear the internal buffer. The user could use multiple form 1 of `HHTXD` to accumulate payload in the internal buffer and send it out with form 2 of `HHTXD`.

! → The returned `MMOKY` to the form 1 of `HHTXD` does not contain a valid packet number because the packet has not been generated by the modem. Instead -1 appears as the packet number in the `MMOKY`.

Example : Broadcast the string "helloworld" (68656C6C6F776F726C64) in a packet of type 0, mode 2 and 50% amplitude level.

```
> $HHTXD,0,2,50,68656C6C6F776F726C64
$MMOKY,HHTXD,0
```

Send to the modem with ID 99 the string "helloworld" in a packet of full amplitude. Use the type and mode value stored in the configuation register `PTYPE` and `PMODE`.

```
> $HHTXD,99,0,0,68656C6C6F776F726C64
$MMOKY,HHTXD,1
```

Combines several payloads and send them in one packet.

1. Put "dummy" (64756D6D79) into the internal buffer of modem

```
> $HHTXD,64756D6D79
$MMOKY,HHTXD,-1
```

2. Clear the internal buffer to remove "dummy"

```
> $HHTXD
$MMOKY,HHTXD,-1
```

3. Put "hello" and "world" in two commands

```
> $HHTXD,68656C6C6F
$MMOKY,HHTXD,-1
> $HHTXD,776F726C64
$HHOKY,HHTXD,-1
```

4. Broadcasting the packet with the mode 2 specified.

```
> $HHTXD,0,2,0
$MMOKY,HHTXD,2
```

Send a T65 packet on a modem to measure the distance from itself to the destination modem.

```
> $HHTXD,13,T65,0
$MMOKY,HHTXD,1
$MMTDN,0,1
$MMOKY,RANGE,-0.866272m
```

The preamble mode in the packet above is 0. To send a T65 packet with preamble type 10 to measure the distance to another modem:

```
> $HHTXD,13,T65M10,0
$MMOKY,HHTXD,1
$MMTDN,0,1
$MMOKY,RANGE,-0.866272m
```

After registering the ranging packet, the modem reports an `MMOKY` to reponse the `HHTXD` command and then an `MMTDN` command to indicate that the packet has been sent out. After receiving the response from the destination node, the modem reports the ranging result in a `MMOKY` command. The ranging result in this example is measured with two modems beside each other. Due to the resolution of ranging measurement, the -0.866272 m can be considered as 0 m.

HHTXA: The host sends this command to have the modem sending out a packet. The packet payload in this command is in ASCII character. The payload should not contain the special charactors and control charactors discussed in Sector 3.2.
The usage of `HHTXA` is same as the `HHTXD`. The only difference is that the payload of `HHTXA` are valid ASCII characters.

Syntax : `$HHTXA[[,<DST>,{0|T<TYPE>[M<PREAMBLE MODE>]|<MODE>},`
`<PAL>],<ASCII>]<IN TERM>`

Return : `$MMOKY,HHTXA,<PACKET NUM><OUT TERM>`
`$MMERR,HHTXA,<ERROR><OUT TERM>`

! → When using multiple `HHTXx` commands to combine the payloads, the user can mix the `HHTXD` and `HHTXA` commands as long as the payload in each command complies with the requirement of the command.

HHTXW: The host sends this command to have the modem send out a waveform stored in its file system. The waveform file consists of sampling points represented with 16-bit two's complement numbers in the little-endian format. The sampling rate and frequency band must be the same as the modem. The information of modem's sampling rate and frequency band can be found in Section 2.1.

The user can use the HHFSA command to upload the waveform files to the modem's file system. When the modem receives a packet sent by the `HHTXW` command, it stores the pass band waveform data in the filesystem. The user can find detailed information on the packet data saving in Section 5.1.

The desination modem ID specifies which modem will receive this.

The path to the waveform file should be the full absolute path.

Syntax : `$HHTXW,<DST>,[PREAMBLE MODE,]<PATHTOWAVE><IN TERM>`

Return : `$MMOKY,HHTXW,<PACKET NUM><OUT TERM>`
`$MMERR,HHTXW,<ERROR><OUT TERM>`

Example : Broadcast a waveform named "CDMA.DAT" under /WAVE direcotry in the modem's file system

```
> $HHTXW,0,/WAVE/CDMA.DAT
$MMOKY,HHTXW,4
```

The preamble mode in the packet above is 0. To send the waveform with preamble modem 10:

```
> $HHTXW,0,10,/WAVE/CDMA.DAT
$MMOKY,HHTXW
```

HHCRW: The host sends this command to change one of the modem's configuration registers. Detailed information of the modem registers and usage of this command can be found in Chapter 4.

Syntax : `$HHCRW,<REG>[,FIELD],<VAL><IN TERM>`

Return : `$MMOKY,HHCRW,<REG><OUT TERM>`

HHCRR: The host sends this command to read one of the modem's configuration registers. Detailed information of the modem registers and usage of this command can be found in Chapter 4.

Syntax : `$HHCRR,<REG>[,FIELD]<IN TERM>`

Return : `$MMOKY,HHCRR,<REG>[,FIELD],<VAL><OUT TERM>`

HHFSA: The host sends this command to operate on the modem's file system. The `HHFSA` command consists of its own name, a file system operation and some argments used with the file system command. Besides the `MMOKY` or `MMERR` commands, the modem may return extra messages which are producted by file system operations. Detailed information of file system operations can be found in Section 5.2.

Syntax : `$HHFSA,<FSOP>[,PATH[,ARG[,ARG]]]<IN TERM>`

Return : `[FS MSG]<OUT TERM>$MMOKY,HHFSA<OUT TERM>`

4 Modem Registers

Users can use modem registers to configure the modem and control its behavior. Each register has a unique name. The user can access the registers by their names with the `HHCRR` or `HHCRW` command.

There are three register attributes: `Read-Only` (`RO`), `Write-Only` (`WO`) and `Non-Volatile` (`NV`). If a register is labeled with Non-Volatile, its value is stored in non-volatile memory space and won't be lost when the modem is shutdown.

A register may have several fields. The user can access an individual field by specifying the field name in the Option parameter in the `HHCRR` or `HHCRW` command. Naturally, the fields inherites the attributes from the register.

MID: Modem ID `(Non-Volatile)`
The physical address of the modem for the communication. A valid ID value ranges from 1 to 255. 0 is reserved for broadcasting.

Default : `1`

Syntax :

```
$HHCRW,MID,<mid><IN TERM>
$HHCRR,MID<IN TERM>
```

Example : Assign the value 99 to a modem's ID and read it back:

```
> $HHCRW,MID,99
$MMOKY,HHCRW,MID
> $HHCRR,MID
$HHOKY,HHCRR,MID,99
```

UART: Uart Configuration `(Non-Volatile)`

This register stores configurations of the serial port connection between the modem and the host. There are four configurable parameters: baud rate, data bits, parity check and stop bits. The user can access them by using fields:

- `BAUD` The supported baud rates are `4800, 9600, 19200, 38400, 57600, 115200` and `230400` bps.
- `DATABITS` The 7-bit and 8-bit words are supported.
- `PARITY` The odd parity, even parity and no parity are supported. In this field, 0 means no parity, 1 means odd parity and 2 means even parity.
- `STOPBITS` 1 stop bit and 2 stop bits are supported.

Default :

```
BAUD: 9600
DATABITS: 8
PARITY: 0
STOPBITS: 1
```

Syntax :

```
$HHCRW,UART,<BAUD|DATABITS|PARITY|STOPBITS>,<value><IN TERM>
$HHCRR,UART,<BAUD|DATABITS|PARITY|STOPBITS><IN TERM>
```

Example : Change the baud rate to 115200 and read it back:

```
> $HHCRW,UART,BAUD,115200
$MMOKY,HHCRW,UART,BAUD
> $HHCRR,UART,BAUD
$MMOKY,HHCRR,UART,BAUD,115200
```

Use the odd parity check

```
> $HHCRW,UART,PARITY,1
$MMOKY,HHCRW,UART,PARITY
```

Use 7-bit word:

```
> $HHCRW,UART,DATABITS,7
$MMOKY,HHCRW,UART,DATABITS
```

Use 2 stop bits:

```
> $HHCRW,UART,STOPBITS,2
$MMOKY,HHCRW,UART,STOPBITS
```

! → The baud rate change takes effect instantly. The user may not be able to see the `MMOKY` response to the baud rate write command.

ECHO: Echo on serial port `(Non-Volatile)`

Toggle echo on/off on the modem serial port.

Default : `OFF`

Syntax : `$HHCRW,ECHO,{ON|OFF}<IN TERM>`
`$HHCRR,ECHO<IN TERM>`

Example : Toggle the serial port echo first on and then off:

```
> $HHCRW,ECHO,ON
$MMOKY,HHCRW,ECHO
> $HHCRW,ECHO,OFF
$MMOKY,HHCRW,ECHO
```

LOG: Log file `(Non-Volatile)`

Toggle log on/off in the modem file system. More infomation about modem log can be found in Chapter 5.

Default : `ON`

Syntax : `$HHCRW,LOG,{ON|OFF}<IN TERM>`
`$HHCRR,LOG<IN TERM>`

Example : Toggle the modem log first off and then on:

```
> $HHCRW,LOG,OFF
$MMOKY,HHCRW,LOG
> $HHCRW,LOG,ON
$MMOKY,HHCRW,LOG
```

MMCHK: Checksum for modem user interface `(Non-Volatile)`

Toggle checksum on/off for the modem `MM` commands. When it is tuned on, a checksum is appended to each command from the modem to the host. If the checksum is present in a command from the host to the modem, the modem will check it and reply with an error if the checksum fails.

Default : `OFF`

Syntax : `$HHCRW,MMCHK,{ON|OFF}<IN TERM>`
`$HHCRR,MMCHK<IN TERM>`

Example : Toggle the checksum on:

```
> $HHCRW,MMCHK,ON
```

```
$MMOKY,HHCRW,MMCHK*7F
```

Send a commands with checksum to toggle the checksum off:

```
> $HHCRW,MMCHK,OFF*49
$MMOKY,HHCRW,MMCHK
```

PGAP: Packet gap `(Non-Volatile)`

The gap time to separate two packets in data mode. If the modem does not receive a character in a time period specified by this register, it considers that the current packet has been received and sends out the data that have been accumulated in the packet buffer. The unit is in *millisecond*. The valiad values ranges from `50ms` to `150,000ms`.

Default : `2000`

Syntax : `$HHCRW,PGAP,<gap time><INTERM>`
`$HHCRR,PGAP<IN TERM>`

Example : Set the gap to `1,000 ms` and read it back:

```
> $HHCRW,PGAP,1000
$MMOKY,HHCRW,PGAP
> $HHCRR,PGAP
$MMOKY,HHCRR,PGAP,1000
```

ESC: Escape string `(Non-Volatile)`

Escape string to switch the modem to the command mode from the data mode. The escape string must consist of four characters. The characters in the escape string cannot be control characters or special characters.

Default : `+++A`

Syntax : `$HHCRW,ESC,<four character string><IN TERM>`
`$HHCRR,ESC<IN TERM>`

Example : Set the escape string to `^^^A` and read it back:

```
> $HHCRW,ESC,^^^A
$MMOKY,HHCRW,ESC
> $HHCRR,ESC
$MMOKY,HHCRR,ESC,^^^A
```

! $\rightarrow$ The escape string has to be between two gaps in the data mode.

RXFMT: Received data format `(Non-Volatile)`

The received data format is either `ASCII` or `HEX`. When this register is set to `HEX`, the modem reports the received packet in hexadecimal format with `MMRXD`. When it is set to `ASCII`, the modem checks whether there is any special or control character in the packet. If there is none, the modem reports the packet in ASCII format with `MMRXA`. Otherwise it reports the packet in hexadecimal format with `MMRXD`.

Default : `ASCII`

Syntax :

```
$HHCRW,ESC,{ASCII|HEX}<IN TERM>
$HHCRR,ESC<IN TERM>
```

Example : Set the received data format to `ASCII` and read it back:

```
> $HHCRW,RXFMT,ASCII
$MMOKY,HHCRW,RXFMT
> $HHCRR,RXFMT
$MMOKY,HHCRR,RXFMT,ASCII
```

ERRFMT: Error display format `(Non-Volatile)`

The error format is either `MSG` or `CODE`. Each error has an error code and an error message explaining the code. The user can find the description for the errors in the Appendix A. When this register is set to `MSG`, the modem reports an error message when it detects an error in the `HH` commands or in another process. When set to `CODE`, the modem reports an error code instead.

Default : `MSG`

Syntax :

```
$HHCRW,ERRFMT,{MSG|CODE}<IN TERM>
$HHCRR,ERRFMT<IN TERM>
```

Example : Set the error format to be `MSG` and `CODE`. Use an unformatted command to check the difference.

```
> $HHCRW,ERRFMT,MSG
$MMOKY,HHCRW,ERRFMT
> $HHCRR,,ERRFMT
$MMERR,Unkonw register
> $HHCRW,ERRFMT,CODE
$MMOKY,HHCRW,ERRFMT
```

```
> $HHCRR,,ERRFMT
$MMERR,0x5001
```

CMDTERM: Command terminators (`Non-Volatile`)

Terminators of modem commands (input terminators) and reponses (output terminators). Input terminators indicate the end of a command, and output terminators are appened to all modem response to the host. The terminators can be `CR`, `LF`, `CRLF`, `LFCR`. This register has two fields: `IN` and `OUT`. The user must specify one field in the `HHCRW` and `HHCRR` commands.

Default :
```
IN:  CRLF
OUT: CRLF
```

Syntax :
```
$HHCRW,CMDTERM,{IN|OUT},<terminators><IN TERM>
$HHCRR,CMDTERM,{IN|OUT}<IN TERM>
```

Example : Set the input terminators to `CR`:

```
> $HHCRW,CMDTERM,IN,CR
$MMOKY,HHCRW,CMDTERM
```

Now change the terminal's line end to `CR`, and read the input terminators:

```
> $HHCRR,CMDTERM,IN
$MMOKY,HHCRR,CMDTERM,IN,CR
```

RTC: Real-time clock

The user can get and set the real-time clock of the modem. This register has two fields: `TIME` and `DATE`. The user must specify one field in the `HHCRW` and `HHCRR` commands. The time is formated as `hour:minutes:seconds` in 24-hour notation. The date is formated as `year:month:day`.

Default :

Syntax :
```
$HHCRW,RTC,{TIME|DATE},{time|date}<IN TERM>
$HHCRR,RTC,{TIME|DATE}<IN TERM>
```

Example : Set time to 01:23:45 and read it back:

```
> $HHCRW,RTC,TIME,01:23:45
$MMOKY,HHCRW,RTC
> $HHCRR,RTC,TIME
$MMOKY,HHCRR,RTC,TIME,01:23:45
```

Set the date to Jan 1, 2015 and read it back:

```
> $HHCRW,RTC,2015-01-01
$MMOKY,HHCRW,RTC
> $HHCRR,RTC,DATE
$MMOKY,HHCRR,RTC,DATA,2015-01-01
```

! → The year in the date must be of 4 digits

GDT: Packet block guard time (`Non-Volatile`)

The guard time is the time interval between two date blocks, the synchronization block and the first data block. The unit is millisecond. The valid values are `50`, `100` and `150 ms`. Using a smaller guard time increases the communication data rate but deceases the communication robustness in complicated mutil-channel environments.

Default : `150`

Syntax : `$HHCRW,GDT,{50|100|150}<IN TERM>`
`$HHCRR,GDT<IN TERM>`

Example : Set guard time to 50 and read it back:

```
> $HHCRW,GDT,50
$MMOKY,HHCRW,GDT
> $HHCRR,GDT
$MMOKY,HHCRR,GDT,50
```

RXGAIN: Peceiving gain (`Non-Volatile`)

The user can set receiving pre-amplifier gain in decibel. The valid value ranges from `0` to `32dB`.

Default : `0`

Syntax : `$HHCRW,RXGAIN,<dB><IN TERM>`
`$HHCRR,RXGAIN<IN TERM>`

Example : Set receiving gain to 10dB and read it back:

```
> $HHCRW,RXGAIN,10
$MMOKY,HHCRW,RXGAIN
> $HHCRR,RXGAIN
$MMOKY,HHCRR,RXGAIN,10
```

TXPWR: Transmitting Power `(Non-Volatile)`

The user can set the transmitting power level in decibel. The valid input value ranges from `0` to `-20dB`.

Default : `-10`

Syntax :

```
$HHCRW,TXPWR,<dB><IN TERM>
$HHCRR,TXPWR<IN TERM>
```

Example : Set transmitting power level to -3dB and read it back:

```
> $HHCRW,TXPWR,-3
$MMOKY,HHCRW,TXPWR
> $HHCRR,TXPWR
$MMOKY,HHCRR,TXPWR,-3
```

RXFLG: Receiving flags `(Non-Volatile)`

These flags control the receiving behavior of the modem. They are stored in one register and can be accessed individually by fields:

- `TCHL:` trigger channel number in a multiple receiving channel modem. The trigger channel is always on to monitor the incomming acoustic packet while other channels are only turned on after a valid trigger is detected. In a `N`-channel modem, the valid value is from `0` to `N-1`.
- `RALL:` receiving all flag. When the `RALL` flag is set `TRUE`, the modem does not check the `destination ID` of the incoming packets and receives all of them. Otherwise, the modem receives only the broadcasting packets and the packets with the `destination ID` as same as its modem ID.
- `ISNR:` input Signal-to-Noise Ratio calculation flag. When the `ISNR` flag is set `TRUE`, the modem calculates the Signal-to-Noise Ratio of the incoming packets. Otherwise it does not.
- `ZONE:` zone information calculation flag. When the `ZONE` flag is set `TRUE`, the modem calculates the channel cluster zone information of the incoming packets. Otherwise it does not.
- `RAW:` Receiving packet raw data saving flag. When the `RAW` flag is set `TRUE`, the modem saves the pass band raw data of the incoming packets to its file system. Otherwise it does not.
- `DBG:` Receiving debug printing flag. When the `DBG` flag is set `TRUE`, the modem prints out the receiving debug information of the incoming

packets. Otherwise it does not.

Default :

```
TCHL: 0
RALL: ON
ISNR: OFF
ZONE: OFF
RAW:  OFF
DBG:  OFF
```

Syntax :

```
$HHCRW,RXFLG,<TCHL>,<channel number><IN TERM>
$HHCRW,RXFLG,<RALL/ISRN/ZONE/RAW/DBG>,<ON/OFF><IN TERM>
$HHCRR,RXFLG,<TCHL/RALL/ISRN/ZONE/RAW/DBG><IN TERM>
```

Example : Set trigger channel to channel 1 and read it back:

```
> $HHCRW,RXFLG,TCHL,1
$MMOKY,HHCRW,RXFLG
> $HHCRR,RXFLG,TCHL
$MMOKY,HHCRR,RXFLG,TCHL,1
```

Toggle debug print on:

```
> $HHCRW,RXFLG,DBG,ON
$MMOKY,HHCRW,RXFLG
```

Toggle raw data saving off:

```
> $HHCRW,RXFLG,RAW,OFF
$MMOKY,HHCRW,RXFLG
```

TXFLG: Transmitting flags `(Non-Volatile)`

These flags control the transmitting behavior of modem. They are stored in one register and can be accessed individually by fields:

- `DONE:` transmitting done report flag. When the `DONE` flag is set to `ON`, the modem sends a `MMTDN` command to the host when a packet has been transmitted. Otherwise, the modem does not send the `MMTDN` command.
- `DBG:` Transmitting debug printing flag. When the `DBG` flag is set `TRUE`, the modem prints out the transmitting debug information of the outgoing packets. Otherwise it does not.

Default :

```
DONE:  OFF
DBG:  OFF
```

Syntax : `$HHCRW,RXFLG,{DONE|DBG},{ON|OFF}<IN TERM>`
`$HHCRR,RXFLG,{DONE|DBG}<IN TERM>`

Example : Toggle DONE on to enable the `HHTDN` command:

```
> $HHCRW,TXFLG,DONE,ON
$MMOKY,HHCRW,TXFLG
```

Read the transmitting debug status:

```
> $HHCRW,TXFLG,DBG
$MMOKY,HHCRR,TXFLG,DBG,OFF
```

PDST: Packet destination Modem ID `(Non-Volatile)`
The destination modem ID is effective in all transmitted packets when the modem works in `DATA` mode. When the modem works in the `COMMAND` mode, the destination modem ID is specified in the `HHTXA` and `HHTXD` commands. The valid value of this register ranges from `0` to `255`, `0` means broadcasting.

Default : 0

Syntax : `$HHCRW,PDST,<broadcasting or destination modem ID><IN TERM>`
`$HHCRR,PDST<IN TERM>`

Example : Set the destination modem ID to 99 and read it back

```
> $HHCRW,PDST,99
$MMOKY,HHCRW,DST
> $HHCRR,PDST
$MMOKY,HHCRR,DST,-99
```

PMODE: Packet mode `(Non-Volatile)`
The packet mode is effective in all transmitted packets when the modem works in `DATA` mode and when the mode is not specified in the the `HHTXA` and `HHTXD` commands in `COMMAND` mode. The valid values of this register are the modes that the modem supports.

Default : 1

Syntax : `$HHCRW,PMODE,<mode><IN TERM>`
`$HHCRR,PMODE<IN TERM>`

Example : Set the packet mode to 2 and read it back

```
> $HHCRW,PMODE,2
```

```
$MMOKY,HHCRW,PMODE
> $HHCRR,PMODE
$MMOKY,HHCRR,PMODE,2
```

PTYPE: Packet type `(Non-Volatile)`
The packet type is effective in all transmitting packets when the type is not specified in the the `HHTXA` and `HHTXD` commands in `COMMAND` mode. The valid values of this register are the types that the modem supports.
The packet type has no effect when modem is UI data mode.

Default : 0

Syntax : `$HHCRW,PTYPE,<type><IN TERM>`
`$HHCRR,PTYPE<IN TERM>`

Example : Set the packet type to 1 and read it back

```
> $HHCRW,PTYPE,1
$MMOKY,HHCRW,PTYPE
> $HHCRR,PTYPE
$MMOKY,HHCRR,PTYPE,1
```

FWVER: Firmware version `(Read-Only, Non-Volatile)` User can read this register to get the firmware version of the modem.

Syntax : `$HHCRR,FWVER<IN TERM>`

Example : Read the firmware version

```
> $HHCRW,FWVER
$MMOKY,HHCRR,FWVER,5.11.01.02
```

FXN: Functionalities `(Write-Only)` The register allows the user to perform special functions. Function is specified in the `FIELD` argument in the `HHCRW` command.

- `CFGSAVE`: Configuration save. This function saves to current values of the non-volatile registers.
- `FACSET`: Factory setting. This function restors all registers with default values.

- `MANUALRX`: Manual receiving. This function issues a manual trigger to the receiver of the modem. This function is mainly for debug purpose.

Syntax : `$HHCRW,FNX,<function name><IN TERM>`

Example : Save current configurations

```
> $HHCRW,FXN,CFGSAVE
$MMOKY,HHCRW,FXN,CFGSAVE
```

Restore default settings

```
> $HHCRW,FXN,FACSET
$MMOKY,HHCRW,FXN,FACSET
```

Force the manual trigger

```
> $HHCRW,FXN,MANUALRX
$MMOKY,HHCRW,FXN,MANUALRX
```

5 File System

This chapter describes the file system of the modem.

The internal storage of the modem is formatted as a FAT disk. The user can store log files and raw data of modem packets in the file system.

5.1 Naming Files and Directories

The modem automatically creates two directories in the root directory after formatting the internal stroage. One directory contains the log files and is named `LOG`, the other contains the acoustic packet data files and is named `DATA`. The user can create other directories and upload files to the file system. There are two naming rules for the files and directories.

- File or directory names must comply with the 8.3 filename rule. 8.3 filenames have at most eight characters, optionally followed by a period '`.`' and a filename extension of at most three characters. For files with no extension, the '`.`', if present, has no significance. File and directory names are uppercase.
- The absolute path to a file or a directory must consists of no more than 32 characters.

The `DATA` directory is read-only to the user. The modem creates and writes the subdirectories and files in the `DATA` directory. If the user enables the `RAW` field in the configuration register RXFLG, upon receiving an acoustic packet the modem creates a new data file and writes the pass band packet data to it. The data file name consists of a prefix, an index number of 7 hex digits and an extension of "DAT". The index number starts from 0 and increases by 1 every time when a new data file is created. The data files are stored in subdirectories. Each subdirectoy contains 1022 data files. Once the number of files in a subdirectory reaches 1022, a new subdirectory is created and named with the index number of the next data file. The prefix in the data file name represents the decode result of the packet data.

G :	The packet has been decoded completely and successfully.
F :	The decode fails on one or more data block in the packet.
E :	The decode fails on the preamble.
M :	The packet is a customized waveform transmitted by the user.
I :	The destination modem ID in the non-broadcasting packet is different from the modem's ID, the modem doesn't decode and save the data blocks. [3]

The LOG directory is read-only to the user. After formatting, the modem creates a default log file named DEFAULT.LOG. If the user sets the configuration register LOG, the modem will save in the default log file the acoustic packet data, debug information of acoustic packets, MM commands and the formatted HH commands. The user can save the default log with another name using LOGSAVE of HHFSA. After saving, the modem creates a new DEFAULT.LOG and writes to it.

5.2 File System Operations

The user can access the modem's file system with the HHFSA command. This section describes the supported file system operations

LS: List directory contents. List information about the files and directories in the target directory.

Syntax : `$HHFSA,LS,<PATH>,<SKIP>,<NITEMS><IN TERM>`

PATH :	Absolute path to the target directory
SKIP :	If *SKIP* is non-negative, the list operation starts from the *(SKIP+1)*'th item in the target directory.

[3] The I files appear only when the RALL field of RXFLG is set OFF.

NITEMS : If *NITEMS* is 0, *SKIP* is ignored and the list opeartion returns only the number of files and number of directory in the target directory. If *NITEMS* is positive, the list opertion returns only *NITEMS* files and directories in total if no less.

Example : The user does not know how many raw data have been stored in the `/DATA` directory, so the user uses the `LS` opertation to check:

```
> $HHFSA,LS,/DATA,0,0
```

The modem returns that 0 file and 1 directory is in `/DATA` directory:

```
      In /DATA
         0 file(s),    1 Dir(s)
$MMOKY,HHFSA
```

The user checks the name of the only directory in the `/DATA` directory:

```
> $HHFSA,LS,/DATA,0,10
```

Because there is only one directory and no file in the `/DATA`, only 1 item is listed:

```
      In /DATA
      D---- 2015-01-01T12:55         0  00000000
         0 file(s),    1 Dir(s)
$MMOKY,HHFSA
```

The user checks how many data files are in the directory `/DATA/00000000`

```
> $HHFSA,LS,/DATA/00000000,0,0
```

The modem returns that 34 files and 0 directory is in `/DATA/00000000`

```
      In /DATA/00000000
        34 file(s),    0 Dir(s)
$MMOKY,HHFSA
```

The user may want to get the information of the last 4 data files:

```
> $HHFSA,LS,/DATA/00000000,30,34
```

The modem returns the detailed information of the last 4 files:

```
      In /DATA/00000000
      ----A 2015-01-01T13:11    252416  G000001E.DAT
      ----A 2015-01-01T13:11    252416  G000001F.DAT
      ----A 2015-01-01T14:10    252416  E0000020.DAT
```

```
        ----A 2015-01-01T14:15    252416  G0000021.DAT
           4 file(s),    0 Dir(s)
$MMOKY,HHFSA
```

RM: Remove files or directories. `RM` removes each specified file. By default it does not remove directories. The user can use the recursive option to remove specified directories, along with all of its content.

Syntax : `$HHFSA,RM,<path>{,R}<IN TERM>`

PATH : Absolute path to the target file or directory

R : `Recursive` option. If specified, `RM` removes the directory and its content.

Example : Remove a log file named `TEMP.LOG` in the `/LOG` direcotory"

```
> $HHFSA,RM,/LOG/TEMP.LOG
$HHOKY,HHFSA
```

Remove a non-empty `/WAVE` directory

```
> $HHFSA,RM,/WAVE,R
$HHOKY,HHFSA
```

! → Although possible, it is not recommended for user to remove the `/DATA` or `/LOG` or their subdirectories.

DISKFREE: Get the free space of the file system.

`DISKFREE` returns the free space left in the file system. The space is reported as Megabytes.

Syntax : `$HHFSA,DISKFREE<IN TERM>`

Example : Check how much free space is left in the file system:

```
> $HHFSA,DISKFREE
     14904.97MB free space
$HHOKY,HHFSA
```

FMT: Format the file system. The user can format the file system with `FMT`. The `FMT` automatically creates a `/DATA` and a `LOG` directory in the root directory. After formmating, lost data on the file system will be lost. To avoid formatting the file system by mistake, a `Confirm` is neccessary in the command.

Syntax : `$HHFSA,FMT,Confirm<IN TERM>`

Confirm : A neccessary argument to avoid accidental formatting.

Example : Format the file system:

```
> $HHFSA,FMT,Confirm
$HHOKY,HHFSA
```

! → The `Confirm` is case-sensitive in this command.

LOGSAVE: Save the log file.

The user can use `LOGSAVE` to save the `DEFAULT.LOG` or save it as another log file. If the user saves the `DEFAULT.LOG` as another file, a new `DEFAULT.LOG` will be created and log will be written to it. Both files are in the `/LOG` directory.

! → The `DEFAULT.LOG` is regularly saved after accumulating a certain amount of log data. However, to prevent the data loss, the user is recommended to use the `LOGSAVE` to do manually saving before powering off the modem.

Syntax : `$HHFSA,LOGSAVE[,LOG NAME]<IN TERM>`

LOG NAME : If *LOG NAME* is omitted, the `DEFAULT.LOG` will be saved. If *LOG NAME* is given, the `DEFAULT.LOG` will be saved in the name of *LOG NAME*.

Example : Save the `DEFAULT.LOG` as `TEMP.LOG`

```
> $HHFSA,LOGSAVE,TEMP.LOG
$HHOKY,HHFSA
```

Save the `DEFAULT.LOG`:

```
> $HHFSA,LOGSAVE
$HHOKY,HHFSA
```

! → Use only file name for *LOG NAME*. Do not include path. The new log will be placed in the `/LOG` directory.

MKDIR: Create a directory.

The user can use `MKDIR` to create a directory if it does not exist.

Syntax : `$HHFSA,MKDIR,<PATH><IN TERM>`

PATH : The absolute path to the directory to be created.

Example : Create a `WAVE` directory in the root directory

```
> $HHFSA,MKDIR,/WAVE
$HHOKY,HHFSA
```

Create a `FSK` directory in the `/WAVE` directory.

```
> $HHFSA,MKDIR,/WAVE/FSK
$HHOKY,HHFSA
```

! → Although possible, it is not recommended for user to create directories in the `/DATA` or `/LOG` directory.

PUT: Put a file into the file system.
The user can use `PUT` to upload a file to the file system.
Following this command, the modem uses the file transfer protocol `Y MODEM` to transfer data with the host device. The user can consult the `File transfer between Modem and Linux PC (AN010)` for more information on how to use `Y MODEM` protocol to upload files to the modem.

Syntax : `$HHFSA,PUT,<PATH><IN TERM>`

PATH : The absolute path to the file to be created. The uploaded data will be written to the file.

Example : Upload a file named `FSK001.BIN` to the `/FSK` directory

```
> $HHFSA,PUT,/FSK/FSK001.BIN
     Ymodem Receiver Started
```

Now user can start the YMODEM protocol to transfer data.

! → Although in the `YMODEM` protocol the file name is tranferred, the modem uses the file name in the *PATH* to create the file.

GET: Get a file from the file system.

The user can use `GET` to download a file from the file system.
Following this command, the modem uses file transfer protocol `Y MODEM` to transfer data with the host device. The user can consult the `File transfer between Modem and Linux PC (AN010)` for more information on how to use `Y MODEM` protocol to download files from the modem.

Syntax : `$HHFSA,PUT,<PATH>,<SKIP>,<LENGTH><IN TERM>`

PATH : The absolute path to the file to be downloaded.

SKIP : If *skip* is non-negative, the downloaded data starts from the *(skip+1)*'th position in the file.

LENGTH : If *LENGTH* is 0, `GET` stops at the end of the file. If *LENGTH* is positive, `GET` downloads only *LENGTH* bytes of data.

Example : Download a file named `G0000001.DAT` in the `/DATA/00000000` directory. Skip the first 1000 bytes and download the rest of the file.

```
> $HHFSA,GET,/DATA/00000000/G0000001.DAT,1000,0
    Ymodem Sender Started
```

Now user can start the YMODEM protocol to transfer data.

CAT: Catenate a file or part of it. The user can use `CAT` to display the contents of a file. `CAT` displays printable ASCII characters and carriage return (`CR`) and line feed (LF). For other characters, `CAT` displays them in hexadecimal value with `\x` prefix.

Syntax : `$HHFSA,CAT,<PATH>,<SKIP>,<LENGTH><IN TERM>`

PATH : The absolute path to the file to be displayed.

SKIP : If *skip* is non-negative, the displayed data starts from the *(skip+1)*'th position in the file.

LENGTH : If *LENGTH* is 0, `CAT` stops at the end of the file. If *LENGTH* is positive, `CAT` displays only *LENGTH* bytes of data.

Example : The `DEFAULT.LOG` is of 1078 bytes. Skip the first 1000 bytes and display the last 78 bytes.

```
> $HHFSA,CAT,/LOG/DEFAULT.LOG,1000,100
    DAT,1000,10
    $MMOKY,HHFSA
    $HHFSA,LS,/LOG,0,10
    $MMOKY,HHFSA
    $HHFSA,LOGSAVE


    $MMOKY,HHFSA
```

Display the first 16 characters of `/DATA/00000000/G0000001.DAT`. [4]

```
> $HHFSA,CAT,/DATA/00000000/G0000001.DAT,0,16
    \x00\x00O\x00\x00\x00I\x00\x00\x00Q\x00\x00\x00H\x00

    $MMOKY,HHFSA
```

6 Troubleshooting

This chapter provides troubleshooting guides which can help identify and correct some basic problems that may be encountered during modem setup and operation. In the following, a list of symptoms and the corresponding suggested actions for them are described. Hopefully the suggested actions can help quickly identify and solve the problem. However, if the problem still remains, please contact our customer service.

Symptom: ***No display in the serial terminal when the modem powers up.***

1. Verify whether the power source is in normal condition, and is correctly connected.
   (a) The power source should provide 16 voltage DC output.
   (b) The polarity of the wire connecting to the power source should match the power source's polarity.
2. Verify the serial connection is setup successfully.
   (a) Check whether the modem is correctly connected to the controller (PC, embedded system and so on) through serial port.
   (b) Check whether the serial terminal opens the correct serial port that the modem is on.
   (c) Check whether the Baud rate is set correctly.
3. Verify whether the modem cable being used is good. Switch to another cable and see whether it solves the problem.

Symptom: ***Serial terminal displays garbled characters or only hex digits.***

In this case, it is likely the serial port Baud rate is wrong. Try to reopen the serial port with different Baud rates.

[4] Because some characters in the data file happen to be printable, there are `O`, `I`, `Q` and `H` in the display contents.

Symptom: ***After successful start, the modem doesn't respond to any user input, or it always replies with a line starting with "$MMERR" for all the input.***

1. For the serial terminal, line feed (LF) and carriage return (CR) should be checked.
2. Check whether the keyboard is working appropriately, whether there is any stuck key.
3. Check whether the cable is still steadily connected in both ends (check the connection of the cable to the modem; check the connections of the cable to the serial port and power source).

Symptom: ***Modem can't transmit successfully.***

If the transmit debug information is on (*$HHCRW,DEBUG,T,1*), sometimes users can see *PAMPERR* in the serial terminal.

1. If the modem is transmitting in air, it will stop sending when the transmit power level is set too high, as a function of power amplifier protection.

   In air environment, it's always recommended to use small power level for transmission.

2. If the modem is connected with a very long cable, and the power source is not strong enough (provides less than 15 V), then the modem can't transmit in full power level. Since in this case, there is no sufficient power provided.

Symptom: ***After the transmitter modem's successful transmission, there is no response at the receiver modem side.***

1. Receiver side debug information is off and the decoding failed. In case the receiver side's debug information is turned off, there won't be any response in receiver's serial terminal when its decoding failed.

   To turn on receiver debug information, use *$HHCRW,DEBUG,R,1*.

2. Verify whether receiver modem is on and working normally.

Symptom: ***The receiver modem failed in decoding.***

1. If the test environment is confined, such as in a small water tank, it's recommended to use small transmission power. Otherwise, the signal

received in receiver modem will get saturated easily, which leads to decoding failure.

If the received signal still saturates in this confined area, users can reduce the water level in the tank, so as to make the water only submerge some part of the hydrophones, while leaves the transducer in air. This can significantly reduce the problem of signal saturation.

2. AquaSeNT OFDM modem supports 7 working modes, their data rates are: 0.375, 0.75, 1.5, 3, 4.5, 6, and 9 kbps correspondingly. The higher the data rate, the higher the mode requires on channel condition. If the receiver failed in decoding, a lower data rate mode can be used to transmit, and see whether the receiver can decode or not.
3. If the test is carried out in long distance, a larger transmit power level can be tried. In the receiver side, SNR values and decoding performance can be observed to see whether system performance is improved.
4. Sometimes, the channel delay spread in the test environment could be very long, and even longer than the pre-defined OFDM guard interval. In this case, the decoding performance will be degraded.

   The user can also try to increase the guard interval length and see whether it improves the SNR and decoding performance.

Symptom: ***Significant difference in receiver channels' SNR values.***

In tests where modems are placed very close to each other (especially in air environment), the receiver's SNR values in different channels could be very different, since some hydrophones' received signals maybe blocked.

User can try to rotate the receiver modem, or place the transmitter modem and receiver modem further away from each other. In this way, the SNR values in the receiver are supposed to be closer to each other.

Appendix

A Error Messages

The AquaSeNT ANE-00009 provides error messages to indicate its operating status. This appendix describes the error messages in order of their error codes.

0x0001 :	UI:Unknown error When this error occurs, contact AquaSeNT's Sales and Service or the company from which you bought the device.
0x0002 :	UI:Unkown command User inputs an unknown Command.
0x0003 :	UI:Not hex string The string is not hexadecimal representation of binary values.
0x0004 :	UI:Unsupported FS opeartion The operation is not supported in `HHFSA`.
0x0005 :	UI:Unsupported FS argument The argument is not supported in `HHFSA`.
0x0006 :	UI:FS I/O error The I/O error during processing of `HHFSA`.
0x0007 :	UI:FS path too long The file path exceeds the FS path limitation.
0x0012 :	UI:Invalid char The character from user input is not allowd.
0x0013 :	UI:Inbox queue full Too much data for the UI input buffer.
0x0014 :	UI:Inbox overrun Same as Error 0x0013.
0x0015 :	UI:Command too long The length of command exceeds the limitaion.
0x0016 :	UI:Illegal char/No terminators. Improper occurance of terminator characters.
0x0021 :	UI:Invalid format The command is in invalid format.

0x0022 :	UI:Invalid checksum	The checksum of command does not pass.
0x1001 :	ACPKT:Illegal modem id	The modem ID in the acoustic packet is improper.
0x1002 :	ACPKT:Invalid mode	The mode of the acoustic packet is improper.
0x1003 :	ACPKT:Invalid type	The type of the acoustic packet is improper.
0x1004 :	ACPKT:Invalid power level	The power level of the acoustic packet is improper.
0x1005 :	ACPKT:Guard time	THe guard time of the acoustic packet is not allowed.
0x1006 :	ACPKT:Memory allocation error	The memory allocation for the acoustic packet failed.
0x1007 :	ACPKT:No payload or payload too long	The length of payload of acoustic packet excceeds the limit of the selected mode.
0x1008 :	ACPKT:No free packet	The acoustic packet queue is full.
0x2001 :	MREG:Unknown register	The register name is unknown.
0x2002 :	MREG:Unknown field	The field is not found in the register.
0x2003 :	MREG:Illegal value	The value is improper for the register.
0x2004 :	MREG:Error in I/O layer	Error occurs in processing of the register operation.
0x2005 :	MREG:Error returned by function	Error occurs in running the functionality.
0x3001 :	AFS:not ready	The file system is not ready for operations.
0x3011 :	AFSD:Unexpected file index	Error occures in auto indexing for raw data saving.
0x4001 :	FS:DISK I/O error	

0x4002 :	FS:Assertion failed
0x4003 :	FS:Physical driver cannot work
0x4004 :	FS:Could not find the file
0x4005 :	FS:Could not find the path
0x4006 :	FS:Invalid path name format
0x4007 :	FS:Access denied or directory full
0x4008 :	FS:Object exist
0x4009 :	FS:Invalid file/directory
0x400A :	FS:Physical driver is Write protected
0x400B :	FS:Invalid logic drive
0x400C :	FS:No work area
0x400D :	FS:No valid FAT volume
0x400E :	FS:Format parameter error
0x400F :	FS:Volume accessing timeout
0x4010 :	FS:Operation rejected
0x4011 :	FS:LNF buffer allocation failed
0x4012 :	FS:Too many open files
0x4013 :	FS:Invalid parameters
0x5001 :	RANGE:No Ack Received No acknowlege returned from the remote modem during ranging.
0x6001 :	YMODEM:Done The YMODEM protocol finishes successfully.
0x6002 :	YMODEM:Cancel The YMODEM protocol has been cancelled.
0x6004 :	YMODEM:Bad frame A corrupt YMODEM frame.
0x6005 :	YMODEM:Frame missing A missing frame detected during the YMODEM protocol.
0x6006 :	YMODEM:CRC Error A YMODEM frame does not pass CRC check.

0x6007 :	YMODEM:Header Error Error occurs in the file infomation header.
0x6008 :	YMODEM:Data Error Error occurs int the dat frame.
0x6009 :	YMODEM:Unrecognized Data Too much unformatted data.
0x600A :	YMODEM:Unexpected frame An unexpected frame appears.
0x600B :	YMODEM:Bus error Data bus error occurs during YMODEM protocol.

Index