WOODS HOLE OCEANOGRAPHIC INSTITUTION

Acoustic Communications Group

Micro-Modem Software Interface Guide

Micro-Modem Software Interface Guide

This version of the manual corresponds to software release AUV0.93.0.02 and higher

Table of Contents

Tables

Release Notes for AUV 0.86+ build (since AUV 0.79)

New features:
1. Acknowledgement of individual data packets (or frames).
2. Ping command to measure travel time to specific units or determine if they are in range.
3. Remote control of hardware output lines on the modem (for example to drop a weight).
4. Sleep command to put a modem into low power mode, either locally or remotely.
5. Command to read the on-board analog-to-digital converter.
6. New transmit rates available.

Release Notes for AUV 0.90+ build (since AUV 0.86)

New commands and features:
1. Narrow-band ping command added and documented (CCPNT)
2. Noise measurement command added and documented (CCCFR).
3. New messages and other changes:
4. The CCMEC command used to set general-purpose output lines on the modem has been expanded to allow control of 2 RTS lines.
5. The thresholds for navigation have been modified so that they are consistent with the communications thresholds. The new defaults for NPT (navigation power threshold), and NDT (navigation detection threshold) are listed in Table 5: Modem NVRAM Configuration Parameter Summary.
6. Relative Doppler between source and receiver is tracked and reported in the new message CADOP.
7. The full software revision number is now reported in the CAREV message (was for example 0.86, now is 0.86.1.17).
8. Always Send Data (ASD) flag added to the NVRAM configuration parameters. When this parameter is 1 the modem will transmit test data when the user does not respond to a data request. This feature is useful for testing modems without an external host. In the case of multi-frame packets, all frames are filled immediately after the first frame request times-out.
9. All of the NVRAM parameters may be reset to the factory profile using the command CCCFG,ALL,0.

Release Notes for AUV 0.92+ build (since AUV 0.90)

1. Mini-packet. The mini-packet that is used for the cycle-init command, ping command, sleep command, etc. is also available to the user. 13 bits may be transmitted in one mini-packet. See CCMUC and associated messages.
2. High rate PSK support with coprocessor card is available.
3. Up to four channels of PSK data can be received with the addition of an analog 4-channel card.
4. Variable length packets with variable number of frames can now be sent.
5. The modem can process GPS messages on an auxiliary serial port.
6. All communication packets can be sent synchronized to a PPS clock and received with exact arrival time information.
7. Extended 8-bit addressing with high-rate packets and 7-bit addressing with FSK packets is now available with limited support.

Features Described but not yet Implemented

1. Auto-sleep feature (Activity Time-Out). If this feature is enabled the modem will automatically go to hibernate state after N seconds of no acoustic or serial port activity (either will keep it awake). It will wake from this mode if the EXTWAKE input pin is pulled high OR at a specified duty cycle, typically 10 seconds out of 100 seconds. The low duty cycle mode allows the modem to look for incoming acoustic signals on an intermittent basis.

2. Built-in data FIFO buffer for data storage prior to transmission upon request from another modem (CCFFL, CCFST).

Interfacing to the Micro-Modem

All Micro-Modem applications use NMEA 0183 messaging.

Getting Started with Acoustic Communication

Only a few commands are necessary to get started transmitting and receiving data via the acoustic modem. The system may be polled (master-slave) for communication, but also supports random access. Active navigation using the REMUS Digital Transponders is asynchronous, but can be coordinated centrally if desired using the Cycle-Initialization command. While many messages are used to provide status, most of these can be turned off if desired. The interface is designed to be NMEA compatible, and thus the interface commands and status messages are all ASCII. Transfer of binary data is handled by hex-encoding it within an NMEA sentence. However, all of the data that is actually transmitted acoustically is sent in compact binary form in fixed-length packets. A short introduction to the interface is provided below. More detail is available in the sentence descriptions and applications chapter.

Topside (or Central) Control: The computer controlling the master modem will use the Cycle Initialization command, $CCCYC, to send or request data to or from other units in the network. Data received back from a remote unit will be sent over the serial port in the $CARXA (ASCII) or $CARXD (binary) messages, depending on which has been enabled. Designation of the 'Master' is actually arbitrary, any modem can be told by the user to initiate a transaction with the Cycle Initialization command.

Remote (or Slave) Units: A computer connected to any modem will receive notification that a Cycle Initialization command has been received. The information that indicates the function of the current cycle is then provided to the user in the $CACYC message. If there is incoming data, it is sent in the $CARXA (ASCII) or $CARXD (binary) messages, depending on which has been enabled. If the master is requesting data, the modem passes that request to the user with the $CADRQ (Data Request) message. After the user provides the data to the modem it is transmitted back to the requestor.

Autonomous Networks: There is no requirement for a 'Master' in the network. At any time any member of the network can make a transmission. Thus, while monitoring of the network and queries for sensor or status information may come from a buoy or ship with a human in the loop, vehicles operating independently may communicate among themselves as desired. For example, vehicles may be launched and the queried for status at the start of a mission, then left to interact asynchronously. However, as message traffic grows, the probability of one system interfering with the other increases. Use of the acknowledgement capability ensures delivery of important packets. Often there is a clear flow of data, for example, vehicles performing initial surveys will pass location data to other vehicles

outfitted with different sensors. Then it makes sense for the survey vehicle to initiate all transactions, and for the other vehicles to transmit only when told to. Important information (as differentiated from state-of-health) should be transmitted with the acknowledgement bit set.

Modem Setup

Setting up a new Micro-Modem requires just a few steps to ensure that it is configured for the intended use. The setup is done using the $CCCFG command. When this command is entered the modem will echo back the value in the $CACFG message. The format of the command is:

$CCCFG,NNN,vv*CS

Where **NNN** is the parameter name, and **vv** is the value. Values can be queried using the $CCCFQ command, which is

$CCCFQ,NNN

The modem echoes back the value of parameter NNN in the $CACFG message.

The basic steps to set up are as follows:

1. Set the unit number (SRC): **$CCCFG,SRC,1**

2. Set the Cycle-Init timeout time (CTO): **$CCCFG,CTO,10**

3. Set data output format flag as desired, for example: **$CCCFG,RXA,1** to enable ASCII data output, or **$CCCFG,RXD,1** for hex-encoded.

Testing that the transmitter works simply requires sending a cycle-init command: $CCCYC,1,1,4,0,0,1 (transmit unit 1 to unit 4, rate 0, no ACK, one packet in the transaction). The modem will send the short cycle-init packet, then request data from the user to transmit. If no data is provided it will generate an error message. A test packet will be transmitted if the ASD parameter is set (Always Send Data).

Packets, Frames and Acknowledgement

As with many communications systems, all data transmitted by the Micro-Modem is broken up into packets. Depending upon the data rate, there may be more than one frame of data within a packet. The current version of the Micro-Modem software supports multiple transmit and receive rates. The high rates contain multiple frames. The integrity of each frame is protected with a cyclic-redundancy check (CRC). The FSK packet contains one 32 byte frame. At the lowest-rate PSK data rate there are three 32-byte frames per packet. The highest current PSK rate has eight 256-byte frames per packet. The data rates and packet formats are listed under the CCCYC command.

Acknowledgement that packets have been delivered without error is provided to the sender when the sender sets the ACK bit in a transmit data message. Then the modem sets the ACK bit high in the acoustic transmission, which prompts the receiving modem to reply with an ACK message acoustically. When the acoustic ACK is received at the modem that originally sent the data it sends a message out the serial port to the user to indicate successful receipt. Note that automatic re-transmissions are not done by the modem, it is left to the user to decide if the data is to be re-transmitted, or if new data is to be generated.

Downlink Transaction without Acknowledgement

A *downlink* is a data transmission originated by the host computer controlling the modem. The sequence of events for a downlink transaction without acknowledgement is as follows:

1. The host sends a CCCYC command to the modem with command type (1 for unit-to-unit), the destination, and data rate (0 for low-rate FSK).
2. The transmitting modem then sends the short Cycle-Init command. All other modems that receive the Cycle-Init report it to their hosts using the CACYC message.
3. After the Cycle-Init command is sent the transmitting modem queries the surface host with the CADRQ message for the data to be transmitted and then the surface modem transmits the data.
4. At the remote modem the data packet is received and if decoded correctly (CRC checks), the data frames are sent to the user, one frame per CARXD message. All modems that decode the data correctly provide the data to their host.
5. For high rate PSK data packets, the cycle init fields are incorporated into the data packet header, thus obviating the need to send the cycle init packet.

Uplink Transaction with Acknowledgement

An *uplink* is a poll for data from another modem. The sequence of events for an uplink transaction with acknowledgement is as follows:

1. The user sends a Cycle-Init (CCCYC) command to the modem with the source ID, receiver ID (itself), packet type, and other parameters.
2. The modem transmits the Cycle-Init packet. The data rate indicated in the Cycle-Init sets the data rate and thus the size of the expected response.
3. When the Cycle-Init command is received by the polled modem it sends the Cycle Init message to its host, then requests data to transmit, making one request for each frame of data using the data request (CADRQ) command. If the polled unit would like an

acknowledgement for a particular packet, it sets the ACK bit in the data response message (CCTXA or CCTXD).

4. The requesting modem receives the data packet from the polled unit and sends the data frames to the polling user, each in a CARXD message. If any of the frames have the ACK bit set, the polling modem then transmits a short packet with the acknowledgement bits for that packet back to the polled modem.

5. At the polled modem the acknowledgement(s) of receipt by the polling unit are sent to the user with the CAACK message.

USER FLAG in Cycle Init Command

There is also a spare bit in the Cycle Init command for use by the originator of the cycle. Several uses are suggested, though the user may employ this flag as appropriate for a given application.

1. ACK request indicator. If the FLAG is set to 1, this indicates that the unit requesting data is asking for data that will require a positive acknowledgement. For example, high-priority data.

2. QUEUE request indicator. If the FLAG is set to 0, then data from the STATUS queue is requested. If 1, then normal data (e.g. from an on-going transaction) is requested.

The modem passes on the FLAG to the user in the cycle-init message, but the user may or may not set the FLAG in the data message it sends back to the modem.

This feature may be used to set up a two-priority queue on client systems. The master may use the FLAG to differentiate between high-priority and state-of-health information. In the case of a remote unit performing event detection, all of the events can be placed in a queue that will require positive acknowledgement. The surface can use the FLAG to indicate that it wants the high-priority information, and the vehicle will provide data from that queue until it is empty, then send other, lower priority, data. The transition from high priority to low may be inferred by the content of the data, or if the ACK feature is set by the transmitter when priority is high, when the ACK bit is no longer set high in data messages.

NMEA 0183 Sentences

The following messages are used to interact with the Micro-Modem. Generally, the messages are NMEA 0183 sentences. Sentences begin with "$" followed by a 5 character talker and message identifier, then some number of comma delimited data fields, a checksum field "*<8 bit HEX checksum>" and finally, the termination sequence <CR><LF> (HEX 0D 0A). Different talker identifiers (the first two characters after the "$" in a talker sentence) are used for the different acoustic tasks the modem can perform, as shown in the table below. Not all have been implemented yet ($DF, $YX, and $XB).

Table 1: NMEA-0183 Talker IDs known by Micro-Modem

Talker ID	Function	Description
$CC	Control Computer	Host Computer connected to modem via serial port
$CA	Acoustic Communications	Data and configuration for acoustic communications
$SN	LBL Navigation	Data and configuration for LBL navigation
$GP, $PG	GPS Receiver	Messages from GPS receiver, only interpreted on the auxiliary serial port when the GPS flag is enabled. See the "GPS" configuration parameter.
$DF	USBL Navigation	Data and configuration for USBL navigation (Not yet implemented)
$YX	Transducer Diagnostics	Data and control for acoustic diagnostics (interference, noise, etc.) (Not yet implemented)
$XB	Transponder Beacon	Data and control for acoustic transponder (Not yet implemented)

NMEA Messages

The NMEA commands and messages supported by the modem are summarized in the following tables, which are sorted alphabetically by message.

Table 2: Micro-Modem NMEA Messages – Host to Modem

Message	Description	Applications	Page
$CCCFG	User sets a modem parameter	All	31
$CCCFQ	User queries for current value of modem parameter	All	35
$CCCLK	User sets real-time clock	All	31
$CCCYC	User commands network cycle	All communication	17
$CCMEC	User controls external hardware I/O lines	All	23
$CCMPC	User commands active navigation cycle (ping)	Active Navigation	19
$CCMSC	User commands modem to sleep	All	21
$CCMUC	User sends User Mini-Packet	Communication	26
$CCPDT	User commands ping to REMUS digital transponders.	Active Navigation	28
$CCTXA	User provides data ASCII data to transmit	Communication	15
$CCTXD	User provides data hex-encoded data to transmit	Communication	14

Table 3: Micro-Modem NMEA Messages – Modem to Host

Message	Description	Applications	Page
$CAACK	Modem reports positive acknowledgement that a frame of data has been correctly received by another unit.	All	12
$CACFG	Modem reports NVRAM configuration parameter	All	4
$CACLK	Modem reports current time (real-time clock)	All	31
$CACYC	Modem reports current synchronous network command	All communication	18
$CADBG	Modem reports low-level debugging messages	All	44
$CADOP	Modem reports relative Doppler (m/sec)	All Communication	44
$CADQF	Modem reports decision quality factor for last FSK packet	All Communication	43
$CADRQ	Modem requests data from the user.	All Communication	12
$CAERR	Modem reports an error	All	45
$CAMFD	Modem reports comms matched-filter detector value	All Communication	43
$CAMSA	A sleep command was received acoustically	All	21
$CAMSC	Echo of sleep command	All	21
$CAMSG	Modem reports a system information message such as a time-out when expecting a data reply.	All Communication	42
$CAMSR	An acoustic reply to a sleep command was received	All	22
$CAMUA	Mini-Packet received acoustically	Communication	26
$CAMUC	Echo of User Mini-Packet command	Communication	26
$CAMUR	Reply to Mini-Packet received	Communication	27
$CAREV	Modem reports application name and revision number	All	42
$CARXA	Modem reports received data in ASCII format	All Communication	13

Message	Description	Applications	Page
$CARXD	Modem reports received data in binary format	All Communication	13
$CARXP	Modem reports detection of start of packet	All Communication	14
$CATXA	Modem confirms receipt of data in CCTXA to transmit	All Communication	15
$CATXD	Modem confirms receipt of data in CCTXD to transmit	All Communication	15
$CATXF	Modem reports end of packet transmission	All Communication	16
$CATXP	Modem reports start of packet transmission	All Communication	16
$SNMFD	Modem reports nav matched-filter detector value(s)	Navigation	30

Quick-Reference Guide for Commonly Used Messages

$CACYC, Command Type, SRC, DEST, Packet Type, ACK Request Bit, Number of frames

Command Types: 0 – Unit transmits to All, 1 – Unit transmits to Unit. 2 – Unit transmits to Group.

$CADRQ, Time, SRC, DEST, ACK Request Bit, Max # Bytes Requested, Frame Counter

$CAERR, Time, Error Name, Error Number

$CAMSG, Message Type, Message Number

$CARXA, SRC, DEST, ACK bit, Frame Number, ASCII data message

$CARXD, SRC, DEST, ACK bit, Frame Number, Hex encoded data message

$CATXD, SRC, DEST, ACK bit, # Bytes in CCTXD message

$CATXA, SRC, DEST, ACK bit, # Bytes in CCTXA message

$CCCFG, Parameter Name, Setting

$CCCFQ, Parameter Name, Setting

$CCCYC – same as CACYC

$CCPDT, Group, Interrogation Channel, Synch Flag, Synch Timeout (ms) AF, BF, CF, DF

Where Group is typically 1 or 2, Interrogation Channel 1-4. Set xF to 1 to enable specific transponders

$CCTXD, SRC, DEST, ACK bit, Hex encoded data message

$CCTXA, SRC, DEST, ACK bit, ASCII data message

$SNTTA, Travel Time 1, Travel Time 2, Travel Time 3, Travel Time 4, System Time

Acoustic Communications Sentences

CAACK – Acknowledgment of a transmitted packet

Message from modem to host indicating that an acknowledgement was received for a previously transmitted data frame.

$CAACK,SRC,DEST,Frame#,A*CS

SRC	Source (unit designated as transmitter of the ack)
DEST	Destination (unit designated as receiver of the ack)
Frame#	Frame number
A	ACK bit, always set to 1.
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Example: $CAACK,2,0,1,1

This ack was sent by node 2 to node 0 indicating that it successfully received frame one from the node 0. Node 0 can then remove the data corresponding to the successfully transacted frame from the queue or buffer.

CADRQ – Data request message, modem to host

This message is from modem to host requesting data to transmit. The host may reply to the modem with either the CCTXA (ASCII data) or CCTXD (hex encoded binary data) messages. If the host does not respond within the time set with the DTO parameter (default is 3 seconds), a timeout error message: $CAERR, DATA_TIMEOUT is sent the serial port by the modem and no acoustic reply is transmitted back to the requester. The number of bytes N is determined by the packet type (see CCCYC message for more information).

$CADRQ,HHMMSS,SRC,DEST,ACK,N,F#*CS

HHMMSS	Time of Request
SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
ACK	ACK bit, 0 or 1.
N	Max number of bytes requested by modem
F#	Frame number
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example: $CADRQ,134351,1,4,0,32,1
At time 13:43:51 the modem requests the host to provide up to 32 bytes of data to transmit to unit 4 for frame number 1. The ACK flag is set to 0 in this request.

CARXA – Received ASCII message, modem to host

Received ASCII data message from the modem to the host. This message is sent to the host when good data (correct CRC) is received by the modem. This message can be enabled or disabled by sending $CCCFG,RXA,x, to the modem where x is 1 to enable and 0 to disable. Default is disabled. Note that binary data transmitted from one modem and received on another with this message enable can produce unreadable results or may create a string that does not conform to the NMEA standard. When data is received but the CRC does not check the message $CAMSG,BAD_CRC,2 is sent to the host.

$CARXA,SRC,DEST,ACK,F#,m…m*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
ACK	ACK bit, 0 or 1.
F#	Frame number
m…m	ASCII text string
***CS**	Hex coded checksum (8 bit XOR of sentence)

Note: ASCII messages must NOT contain the following NMEA reserved characters: carriage return [0x0D], linefeed[0x0A], "$"[0x24], ""[0x2A], ","[0x2C], "!"[0x21],"\"[0x5C],"^" [0x5E],"~"[0x7E].

Example: $CARXA,1,4,1,1,Cycle Test From SecureCRT*67
The message 'Cycle Test from SecureCRT' has been received in frame number 1 of a message originating from unit 1 and destined for unit 4, and the ACK bit was set by the transmitter, and the RXA flag is set so that the modem provides the data in ASCII form.

CARXD - Received binary message, modem to host

Received hex encoded binary data message from the modem to the host. This message is sent to the host when good data (correct CRC) is received by the modem. This message can be enabled or disabled by sending $CCCFG,RXD,x, where x is 1 to enable and 0 to disable. Default is enabled. When data is received but the CRC does not check the message $CAMSG,BAD_CRC,2 is sent to the host.

$CARXD,SRC,DEST,ACK,F#,HH…HH*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
ACK	ACK bit set by transmitter, 0 or 1.
F#	Frame number
HH	Hex coded data bytes, 2 characters each, eg. 00-FF to represent binary numbers from 0-256
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example:
CARXD,4,6,1,1,4379636c6520546573742046726f6d20536563757265435254*3D

The message 'Cycle Test from SecureCRT' has been received in frame number 1 of a message transmitted by unit 4 to unit 6, and the ACK bit was set by the transmitter, and the RXD flag is set so that the modem provides the data in hex-encoded form.

CARXP – Incoming packet detected, modem to host

The modem sends this message when it acoustically detects the start of a packet. It is sent before the full packet is received. This message is only sent if the RXP configuration parameter is set to 1. By default, RXP is disabled (0).

$CARXP,Type*CS

Type   Incoming packet type: 0 if FSK, 1 if PSK.

*CS   Hex coded checksum (8 bit XOR of sentence)

Example: $CCRXP,0*444

CCTXD – Transmit binary data message, host to modem

Binary data message from host to modem for transmission. This message (or alternatively, CCTXA) is sent by the host to the modem in response to the $CADRQ (data request) message. The number of bytes in this message should be the same or less than the number of bytes that were requested by the modem using $CADRQ. The user sets the ACK bit to 1 when an acknowledgement is desired for this packet. The source and destination addresses should be the same as those in the data request message and are included for symmetry with the CARXD message.

$CCTXD,SRC,DEST,A,HH…HH*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
A	ACK bit set by transmiter, 0 or 1.
HH…HH	Hex coded data bytes, 2 characters each, e.g. 00-FF.
*CS	Hex coded checksum (8 bit XOR of sentence) optional

Example: $CCTXD,4,6,0,546573742046726f6d2042756f79*0F

The host provided the hex-encoded data destined for unit 6 from unit 4.

CCTXA – Transmit ASCII data message, host to modem

ASCII data message from host to modem for transmission. This message (or alternatively, $CCTXD) is generated by the host and sent to the modem in response to the $CADRQ (data request) message. The number of bytes in this message should be the same or less than the number of bytes that were requested by the modem using $CADRQ. The user sets the ACK bit to 1 when an acknowledgement is desired for this particular frame of data. The source and destination addresses should be the same as those in the data request message and are included for symmetry with the CARXD message.

$ CCTXA,SRC,DEST,A,MM…M*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
A	ACK bit set by transmitter, 0 or 1.
MM…M	ASCII message
*CS	Hex coded checksum (8 bit XOR of sentence) optional

Note: ASCII messages must NOT contain the following NMEA reserved characters: carriage-return[0x0D], linefeed[0x0A], "$"[0x24], ""[0x2A], "," [0x2C],"!"[0x21],"\"[0x5C],"^" [0x5E],"~"[0x7E.

Example: $CCTXA,4,6,1,ASCII Test Message

CATXD – Echo back of transmit binary data message

CATXA – Echo back of transmit ASCII data message

These two messages are the companions to CCTXD and CCTXA. They are sent back to the user over the serial port in response to data transmit messages. They indicate that the CCTXD and CCTXA message were received correctly and will be transmitted.

$ CATXD,SRC,DEST,A,#Bytes*CS (or CATXA)

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
A	ACK bit set by transmitter, 0 or 1.
# Bytes	Number of bytes in the CCTXD or CCTXA message
*CS	Hex coded checksum (8 bit XOR of sentence)

CCTXP – Start of packet transmission, modem to host

The modem sends this message when it begins acoustically transmitting a packet.

$CCTXP,#Bytes*CS

#Bytes   Number of bytes in the packet
*CS   Hex coded checksum (8 bit XOR of sentence), optional

Example: $CCTXP,4*46

CCTXF – End of packet transmission, modem to host

The modem sends this message when it has finished acoustically transmitting a packet.

$CCTXF,#Bytes*CS

#Bytes   Number of bytes in the packet
*CS   Hex coded checksum (8 bit XOR of sentence), optional

Example: $CCTXF,2*56

Acoustic Network Protocol Sentences

CCCYC – Network Cycle Initialization Command

This command starts a network cycle and is typically used by the network master, though any unit can originate a transaction. This message sets up transactions as listed in the Network Protocol Command Table shown below. Note that if the command is to transmit data, then the modem will send a message requesting the data to the user ($CADRQ). The command may be used by any unit, but there is a risk of collision if multiple units simultaneously start a network cycle. Thus it is typically used only by the Master, though remote units can use this command to talk to others, including the Master. Normally this command is used by the Master to talk directly with another unit, in which case it puts its own address in ADR2. However, this command allows the Master to tell a unit to transmit to a different one, (i.e. unit 2 to talk to 3). In that case the transmitting unit is ADR1 and the receiving unit is ADR2. See " AUV Application Examples: Communication" (page 50) for additional examples.

$CCCYC,CMD,ADR1,ADR2,Packet Type,ACK,Npkt*CS

CMD	Deprecated. Use any number from 0-7 in this field.
ADR1	Address 1 - Unit or Group # (depends on command type)
ADR2	Address 2 – Unit or Group# (depends on command)
Packet Type	Packet type, 0-5. Current packet types include:
	0 80 bps FH-FSK
	1 250 bps 1/31 spreading (unimplemented)
	2 500 bps 1/15 spreading
	3 1200 bps 1/7 spreading
	4 1300 bps 1/6 rate block code (unimplemented)
	5 5300 bps 9/14 rate block code.
ACK	Deprecated. Use either 0 or 1 in this field. To send acknowledgment mini-packets, set the ACK bit in the $CCTXD message.
Nframes	Number of frames to send in packet
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

All Cycle Initialization packets are mini-packets. For FH-FSK data packets, they are sent using Packet Type 0, FH-FSK. For higher rates, the mini-packet modulation depends on the value of the MOD parameter. If it is 1, the mini-packet is set using PSK modulation. Data packets are transmitted using the indicated type. The approximate length of all data packet transmissions with maximum number of allowable frames is 3.5 seconds. The data payload for each packet is as follows:

Table 4: Packet Types

Packet Type	Max. num. of Frames	Bytes/Frame	Bytes/Packet
0	1	32	32
1	3	64	192
2	3	64	192
3	2	256	512
4	2	256	512
5	8	256	2048
6	6	32	192

CACYC – Echo of Network Cycle Initialization command

This message is the echo back to the user of the CCCYC message. It provides a check for the user that the command was received by the modem. The parameters of the echo are the same as the command.

Ping Sentences

The Ping command is used to check if a particular unit is within communication range, and to measure the travel time to the unit. The Ping command is done using a mini-packet (just like the Cycle-Init command), so that it is relatively short (less than 1 second long). The ping command is sent using FSK modulation or PSK modulation based on the value of the NVRAM MOD parameter. A Ping transaction involves several NMEA sentences as described below.

CCMPC – Mini-Packet Ping command, host to modem

To ping another unit the CCMPC command is used. The SRC is the originator of the command, the DEST is the unit to be pinged.
$CCMPC,SRC,DEST*CS

SRC	Source (unit designated as ping originator)
DEST	Destination (unit designated as receiver of the ping)
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

CAMPC – Echo of Ping command, modem to host

When the modem sees a CCMPC command it responds back with a CAMPC message as a confirmation to the user that the command is received.

$CAMPC,SRC,DEST*CS

SRC	Source (unit designated as ping originator)
DEST	Destination (unit designated as receiver of the ping)
***CS**	Hex coded checksum (8 bit XOR of sentence)

CAMPA – A Ping has been received, modem to host

When a ping is received at any unit, the CAMPA message is sent over the serial port to the user. The information provides awareness of network activity. The SRC is the originator of the ping command, the DEST is the unit being pinged. When the ping is received at DEST the modem automatically responds. Thus the CAMPA is simply notification and no action is necessary by the user.

$CAMPA,SRC,DEST*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver of the ping)
***CS**	Hex coded checksum (8 bit XOR of sentence)

CAMPR – Reply to Ping has been received, modem to host

When the modem that transmits a ping command receives a response it computes the one-way travel time and provides that information in the CAMPR message. Note that the SRC and DEST now reflect the fact that the unit that was pinged is now transmitting. Thus, the value of DEST in this message will match SRC in the original CCMPC command.

When the reply to a ping is received by other modems a CAMPR message is generated for the user. However, these other units have no knowledge of the transmission time of the ping command, so there is no travel time available. The travel time field is empty in this case.

$CAMPR,SRC,DEST,TRAVELTIME*CS

SRC	Source (unit that responded to the ping)
DEST	Destination (originator of the ping)
Travel Time	One-way travel time (seconds) if available
*CS	Hex coded checksum (8 bit XOR of sentence)

CCRSP – Pinging with an FM sweep

This command is used to send out an FM sweep without a packet.

$CCRSP,RXSIG,TXSIG,TIME OUT*CS

RXSIG	Received signal, obsolete. Set to 0
TXSIG	Signal to send. 1 sends an FSK packet FM sweep in the current band , 2 sends a PSK packet FM sweep in the current band.
TIME OUT	Time out in seconds for received signal, obsolete. Set to 0
*CS	Hex coded checksum (8 bit XOR of sentence)

CARSP – Respose to FM sweep ping command

Echo of the **CCRSP** command. For example, if the user issues the command

$CCRSP,0,1,0

The modem echoes the command before transmitting the sweep:

$CARSP,0,1,0*4E

Sleep Sentences

The Sleep command is used to power down a modem for up to one day in coarsely quantized increments. This may be done locally, or via acoustic control. When done acoustically the command is sent using a mini packet. A sleep transaction involves several NMEA sentences as described below. The power draw of a modem in its off state is several milli-watts as opposed to almost 200 mW typically.

CCMSC – Sleep command, host to modem

The local or remote sleep command is CCMSC. The SRC is the originator of the command, the DEST is the unit to be put to sleep. If SRC is the same as DEST then there is no acoustic transmission and the local modem acknowledges the command and then goes to sleep.

An argument within the sleep command is used to specify a time to sleep. Three modes are available. The argument -1 indicates sleep until hardware wakeup (power cycle or hardware line EXTWAKE pulled high). 0 indicates sleep for 0 seconds which simply forces a hard reboot. Arguments larger than 0 and less than or equal to 1524 indicate the number of minutes to sleep before waking up, rounded down to the nearest multiple of 6 minutes. The exception is that values from 1 to 6 are rounded up to 6, and thus the shortest sleep interval is 6 minutes. 1524 minutes corresponds to 25 hours and 24 minutes. Out of range values cause NMEA range error messages.

The sleep command is available in both FSK and PSK modulation schemes, based on the value of the MOD parameter.

$CCMSC,SRC,DEST,ARG*CS

SRC	Source (unit giving the command)
DEST	Destination (unit to sleep)
ARG	-1: Sleep until hardware wakeup. 0: Sleep for 0 seconds (force reboot). 1-1524: Sleep for ARG minutes with quantization as described above.
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

CAMSC – Echo of Sleep command, modem to host

When the modem sees a CCMPC command it responds back with a CAMPC message as a confirmation to the user that the command was received. If SRC is the same as DEST the modem then goes to sleep and the acoustic messages, CAMSA and CAMSR are not used. If the sleep command is destined for a remote unit the originating host should wait for and expect a CAMSR message as confirmation. The arguments and format of the echo are the same as for the command except that the modem always appends the checksum to the message where for input it is optional.

CAMSA – A Sleep was received acoustically, modem to host

When any modem receives a mini-packet with the sleep command it informs the local host whether or not the command is for it. This allows the local host to observe control activity in the channel. The arguments of the CAMSA are the same as in the CCMSC and CAMSC messages.

CAMSR – A Sleep reply was received, modem to host

A modem that receives a sleep command destined for it will respond with a reply message as confirmation before it goes to sleep. That reply message is provided to the user via the CAMSR message. As with the ping reply message, SRC indicates the SRC of the replying unit, i.e. the unit going to sleep, and the DEST is the unit that commanded the sleep.

$CAMSR,SRC,DEST,ARG*CS

SRC	Source (unit confirming the command)
DEST	Destination (unit that originated the command)
ARG	An argument, -1 means sleep until hardware wakeup. 0 means sleep for 0 seconds which simply causes a reboot.
***CS**	Hex coded checksum (8 bit XOR of sentence)

External Hardware Line Control Sentences

The external hardware line control command is used to read or write several hardware output lines on the modem. This may be done locally, or via acoustic control. When done acoustically the command is sent using a mini packet. This transaction involves several NMEA sentences as described below. When using this line to control a burn wire or other high-current device it should be opto-isolated and buffered. Even if it is not used in this way it is suggested that the line be opto-isolated.

CCEXL – External hardware control command, local modem only

Use this command to set the hardware I/O lines on a local modem. The argument to the hardware control command is used to specify which lines are to be manipulated and how. The default state of the output lines is 0 volts upon power up. If the modem does a hard reboot the lines are also cleared to the default state.

$CCEXL,MODE*CS

MODE	Two hex characters (8 bits total) controlling hardware lines. Currently the bits that are assigned are: Bit 0: Toggle EXTSEL1, 0 for low, 1 for high Bit 1: Toggle EXTSEL2, 0 for low, 1 for high Bit 6: Toggle GPO, 0 for low, 1 for high

For example, to set the external select line 1 (EXTSEL1) high, use:

$CCEXL,01

CCMEC – External hardware control command, host to modem

Use this command to set and read the hardware I/O lines on a local or remote modem. The SRC is the originator of the command and the DEST is the unit whose hardware lines are to be controlled. If SRC is the same as DEST then there is no acoustic transmission and the local modem acknowledges the command and performs the requested operation. Arguments to the hardware control command are used to specify which lines are to be manipulated and how. At present two modes are available, read and set. The default state of the output lines is 0 volts (for EXTSEL1 and 1) or -5.5 volts (for RTS1 and 2) upon power up. If the modem does a hard reboot the lines are also cleared to the default state.

Available in both PSK and FSK mini-packet formats.

$CCMEC,SRC,DEST,LINE,MODE,ARG*CS

SRC	Source (unit giving the command)
DEST	Destination (unit whose lines are being controlled)

LINE	1 - EXTSEL1
	2 - EXTSEL 2
	3 - GPIO 1 (GPIO daughter board required)
	4 - GPIO 2
	5 - GPIO 3
	6 - GPIO 4
	7 - COM1 RTS/CTS (reads CTS, writes RTS)
	8 - COM2 RTS/CTS (reads CTS, writes RTS)
MODE	0 – Read (ARG is then 0)
	1 – Set (ARG is 0 to set the line to 0, 1 to assert)
	2 – Toggle high (ARG is duration specified below)
	3 – Toggle low (ARG is duration specified below)
ARG	An integer argument for toggle time duration.
	If in READ mode the argument is not used and should be set to 0.
	0 - 1 ms
	1 - 0.5 sec
	2 - 1 sec
	3 - 2 sec
	4 - 6 sec
	5 - 10 sec
	6 - 20 sec
	7 - 30 sec
*CS	Hex coded checksum (8 bit XOR of sentence) optional

Example: $CCMEC,0,6,1,1,1

Unit 0 is commanding unit 6 to set line 1 to its high state (3 volts). The line will stay high until explicitly set low by a command **$CCMEC,0,6,1,1,0**.
Note that the line will also be reset to the default state (0) when the modem does a hard reboot.

CAMEC – Echo of hardware control command, modem to host

When the modem sees a CCMEC command it responds back with a CAMEC message as a confirmation to the user that the command was received. If SRC is the same as DEST the modem then acts on the command locally and the acoustic messages, CAMEA and CAMER are not used. If the command is destined for a remote unit the originating host should wait for and expect a CAMER message as confirmation.

CAMEA – Hardware control command received acoustically

When any modem receives a mini-packet with the hardware control command it informs the local host whether or not the command is for it. This allows the local host to observe control activity in the channel. If the DEST in the command is the same as the local address the action indicated in the command is performed.

CAMER – Hardware control command reply received

A modem that receives a hardware control command destined for it will respond with a reply message as confirmation. That reply message is provided to the user via the CAMER message. As with the ping reply message, SRC indicates the SRC of the replying unit, i.e. the unit taking the action, and the DEST is the unit that commanded it. If a read command was given, then the ARG is the state of the line on the modem. The arguments for LINE and MODE are the same as in the CCMEC message.

$CAMER,SRC,DEST,LINE,MODE,ARG*CS

SRC	Source (commanded unit now replying)
DEST	Destination (unit originating command)
LINE	1 - EXTSEL1 2 - EXTSEL 2 3 - GPIO 1 (GPIO daughter board required) 4 - GPIO 2 5 - GPIO 3 6 - GPIO 4 7 - COM1 RTS/CTS (reads CTS, writes RTS) 8 - COM2 RTS/CTS (reads CTS, writes RTS)
MODE	0 – Read (ARG is then the state of the line on the unit) 1 – Set (ARG is 0 to set the line to 0, 1 to set it to 3 Volts) 2 – Toggle high (ARG is duration,0 = 1 msec, 1-31 are decoded as durations of 0.5 sec to 15.5 sec) 3 – Toggle low (ARG is duration as specified above)
ARG	An integer argument for toggle time duration, or if in READ mode the state of the line (0 or 1). 0 - 1 ms 1 - 0.5 sec 2 - 1 sec 3 - 2 sec 4 - 5 sec 5 - 10 sec 6 - 20 sec 7 - 30 sec
*CS	Hex coded checksum (8 bit XOR of sentence)

User Mini-Packet Sentences

The user mini-packet command sends very short messages in the same packet size as used by the Cycle-Init command and ping commands. The mini-packet is relatively short (less than 1 second). A user mini-packet transaction involves several NMEA sentences as described below.

CCMUC – User Mini-Packet command, host to modem

A user mini packet is sent using the CCMUC sentence. The data format is ASCII-encoded hex. The data payload is 13 bits, which is encoded in the NMEA sentence as 2 8-bit hex values (4 characters). As with all transactions, all modems that receive the packet report it.

$CCMUC,SRC,DEST,HHHH*CS

SRC	Source (data originator), 4 bits.
DEST	Destination (data receiver), 4 bits.
HHHHHH	ASCII-coded hex data (2 Hex values). Values in the range of 0 to 1FFF are legal.
*CS	Hex coded checksum (8 bit XOR of sentence) optional

CAMUC – Echo of user Mini-Packet, modem to host

When the modem receives the CCMUC NMEA sentence it responds with the CAMUC sentence as confirmation.

$CAMUC,SRC,DEST,HHHH*CS

SRC	Source
DEST	Destination
HHHHHH	ASCII-coded hex data (2 Hex values). Note that the data is zero-padded, e.g. an F input in CCMUC is echoed back as 000F.
*CS	Hex coded checksum (8 bit XOR of sentence)

CAMUA – Mini-Packet received acoustically, modem to host

When a mini-packet is received at any unit, the CAMUA sentence is sent over the serial port to the user. The information provides awareness of network activity. When the mini-packet is received at DEST the modem automatically responds to acknowledge receipt.

$CAMUA,SRC,DEST,HHHH*CS

SRC	Source
DEST	Destination
HHHH	ASCII-coded hex data (2 Hex values). Note that the data is zero-padded, e.g. an F input in CCMUC is sent as 000F.
*CS	Hex coded checksum (8 bit XOR of sentence)

CAMUR – Reply to Mini-Packet received, modem to host

When the modem that transmits a mini-packet receives an acoustic confirmation that the Mini-Packet was received, the CAMUR sentence is provided as confirmation that the data was delivered and acknowledged. Note that the SRC and DEST now reflect the fact that the unit receiving the min-packet is now transmitting. Thus, the value of DEST in this message will match SRC in the original CCMUC command.

When the reply to a Mini-Packet is received by other modems a CAMUR message is generated also.

$CAMUR,SRC,DEST*CS

SRC	Source (unit that received the mini-packet)
DEST	Destination (originator of the mini-packet)
***CS**	Hex coded checksum (8 bit XOR of sentence)

Navigation Sentences

CCPDT – Ping REMUS digital transponder, host to modem

REMUS active LBL navigation ping message from host to modem.

$CCPDT, GRP, CHANNEL, SF, STO, Timeout, AF, BF, CF, DF*CS

GRP	Transponder group designator (at present only 1 is available, which corresponds to a set A-D).
CHANNEL	Interrogation channel, 1-4.
SF	Synchronization Flag. Set to 1 if hardware synchronous outgoing pulse desired, 0 otherwise.
STO	Synch time-out in milliseconds. Set to 0 if not used.
Timeout	Maximum time to listen (milliseconds).
AF,BF,CF,DF	Transponder flags. Set to 1 to enable detection of individual beacons.
***CS**	Hex coded checksum (8 bit XOR of sentence) optional.

Example: $CCPDT,1,1,0,0,2500,1,1,0,0
Ping transponder group 1 on channel 1, no hardware synchronization, 2.5 second time out, listen for transponders A and B (for example, transponders labeled DTxA and DTxB, x might be 2 or 4 depending on the number of channels it is capable of receiving). When two systems are sharing transponders they should use different interrogation codes. As of October 2003 (Micro-Modem software release 0.86) all four interrogation channels are supported by the Micro-Modem and the Digital Transponders.

The 1-way travel time information is returned in the SNTTA message.

Note: the transponder turn around time parameter must be sent to the correct value in the modem non-volatile memory. For the REMUS transponders this is 50 msec. The command to set this is: $CCCFG,TAT,50

CCPNT – Ping narrowband transponder, host to modem

Narrowband active LBL navigation ping message from host to modem.

$CCPNT, Ftx, Ttx, Trx, Timeout, FA, FB, FC, FD,Tflag*CS

Ftx	Transmit frequency to interrogate narrowband transponders
Ttx	Length of transmit ping in milliseconds
Trx	Length of receive pings in milliseconds
Timeout	Maximum time to listen for in milliseconds
FA,FB,FC,FD	Receive frequency list (in Hz), Maximum – minimum frequency should not exceed 5kHz.
Tflag	Flag to indicate whether the transmit signal is actually sent (1), or whether the ping is initiated with an external synchronization pulse (0).
***CS**	Hex coded checksum (8 bit XOR of sentence) optional.

Example 1, Deep-Ocean Transponder Ping:
$CCPNT,9000,10,10,10000,9000,10000,10500,11000,1

Ping at 9 kHz for 10 msec, then listen for transponders that have return pulse length of 10 msec for 10000 msec. The transponder frequencies are 9 kHz, 10 kHz, 10.5 kHz, and 11 kHz. The 1-way travel time information is returned in the SNTTA message.

Example 2, Benthos UAT-376 Ping: $CCPNT,26000,5,5,4000,25000,0,0,0,1

Ping at 26 kHz for 5 msec, then listen for 4000 msec at 25 kHz.

Note: the transponder turn around time parameter must be set to the correct value in the modem non-volatile memory. For UAT-376 the turn-around time is 20 msec. The command to set this is: $CCCFG,TAT,20.

Sample output after $CCPNT is shown below for the case of a bench test at near-zero range. The MFD message is printed if the MFD flag is set.

```
$SNPNT,26000,5,5,4000,25000,0,0,0,0,22118*6E
$SNMFD,01,1393,0154,0904*56
$SNTTA,-0.0005,,,,150347.00*6C
```

SNTTA – Transponder travel times, modem to host

Active LBL navigation one-way travel time message from modem to host. Note that if no signal is detected the travel times fields are empty but the commas remain. *The one-way travel time has the transponder turn-around time already subtracted.* The turnaround time is entered with the CCCFG command: $CCCFG,TAT,xx, where xx is the turn-around time in milliseconds.

$SNTTA,TA,TB,TC,TD,hhmmsss.ss*CS

TA	Travel time from transponder A
TB	Travel time from transponder B
TC	Travel time from transponder C
TD	Travel time from transponder D
hhmmss.ss	Time of ping (Modem time from real-time clock)
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

Example: $SNTTA,0.0733,0.0416,,,014524.00*5C

The 1-way travel time to transponder A is 0.0733 seconds, or 110 meters, and the 1-way travel time to transponder B is 0.0416 s, or 62 meters.

SNMFD – Nav matched filter information, modem to host

Matched-filter detector message from modem to host provided upon reception of a navigation pings. This message can be enabled or disabled by sending $CCCFG,MFD,x, where x is 1 to enable and 0 to disable. Default is disabled.

$SNMFD,NN,MFPK,MFPWR,MFRATIO*CS

NN	Transponder channel
MFPK	Matched-filter peak value.
MFPWR	Incoherent broad-band power at the peak
MFRATIO	Ratio MFPK and MFPWR used for the detector test .
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example: $SNMFD,01,0858,1632,00052

Configuration Sentences

CCCLK – Set clock, host to modem

Set clock message from host to modem. The modem responds with the CACLK message showing the time.

$CCCLK,YYYY,MM,DD,hh,mm,ss*CS

YYYY	Year
MM	Month
DD	Day
HH	Hours
MM	Minutes
SS	Seconds
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example: $CCCLK, 2004,3,28,10,20,0

CCCFG – Set NVRAM configuration parameter, host to modem

Send NVRAM configuration parameter message.

$CCCFG,NNN,vv*CS

NNN	Name of NVRAM parameter to set (See Table below)
Vv	New value
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example: $CCCFG,SRC,1

Set the address of the unit (its source address) to 1. The modem will echo the new setting back with the same sentence and include the checksum as well: $CCCFG,SRC,1*33.
To show all settings at once use $CCCFQ,ALL

Example: $CCCFG,ALL,0

Set ALL of the NVRAM parameters to profile 0 (the factory defaults).

Table 5: Modem NVRAM Configuration Parameter Summary.

Parameter Name (nnn)	Range	Description	Default
AGC	0-1	Turn on automatic gain control	0
AGN	0-255	Analog Gain (50 is 6 dB, 250 is 30 dB)	250
ASD	0-1	Always Send Data. Tells the modem to send test data when the user does not provide any.	0
BBD	0-1	PSK Baseband data dump to serial port	0
BND	0-3	Frequency Bank (1, 2, 3 for band A, B, or C, 0 for user-defined PSK only band)	
BR1	0-7	Baud rate for serial port 1	3
	0	2400	
	1	4800	
	2	9600	
	3	19200	
	4	38400	
	5	57600	
	6	115200	
	7	230400	
BR2	0-7	Baud rate for serial port 2 (see above)	3
BRN	0-1	Run bootloader at next revert	0
BSP	1-2	Boot loader serial port	1
BW0	INT	Bandwidth for Band 0 PSK, Hz Allowable bandwidths are 1000,1250,2000,2500,4000,5000	4000
CPR	0-1	Coprocessor power toggle switch	1
CRL	Int	Cycle init reverb lockout (ms)	50
CST	0-1	Cycle statistics message	1
CTO	0-255	Cycle init timeout (sec)	10
DBG	0-1	Enable low-level debug messages	0
DGM	0-1	Diagnostic messaging	0
DOP	0-1	Whether or not to send the $CADOP message	0
DQF	0 or 1	Whether or not to send the $CADQF message	1
DTH	Int	Matched filter signal threshold, FSK	108
DTO	1-30	Data request timeout (sec)	2
DTP	Int	Matched filter signal threshold, PSK	90
ECD	Int	Delay at end of cycle (ms)	50
EFF	Int	Feedforward taps for the LMS equalizer	20

Parameter Name (nnn)	Range	Description	Default
EFB	Int	Feedback taps for the LMS equalizer	10
FMD	Int	PSK FM probe direction,0 up, 1 down	1
FML	Int	PSK FM probe length, symbols	40
FC0	INT	Carrier at Band 0 PSK only, increments in 250 Hz steps	25000
GPS	0-1	GPS parser on aux. serial port	0
HFC	0-1	Hardware flow control on main serial port. **NOTE**: If RTS is high when the modem boots, it will enter the bootloader. If you use hardware flow control, your interface software should be aware of this condition and include provisions to exit the bootloader.	0
IRE	0-1	Print impulse response of FM sweep	0
MCM	0-1	Enable current mode hydrophone power supply on Rev. C Multi-Channel Analog Board. Must be set to 1 for Rev. B Multi-Channel Analog Board.	1
MFC	Int	MFD calibration value (samples)	-2
MFD	0 or 1	Whether or not to send the MFD messages	0
MOD	0 or 1	0 sends FSK minipacket, 1 sends PSK minipacket	0
MPR	0 or 1	Enable power toggling on Multi-Channel Analog Board	1
MSE	0-1	Print symbol mean squared error (dB) from the LMS equalizer	0
MVM	0-1	Enable voltage mode hydrophone power supply on Multi-Channel Analog Board	1
NDT	Int	Detect threshold for nav detector	150
NPT	Int	Power threshold for nav detector	50
NRL	Int	Navigation reverb lockout (ms)	25
NRV	Int	Number of CTOs before hard reboot	20
PAD	Int	Power-amp delay (ms)	2
PCM	0-255	Passband channel mask	0
POW	Int	Detection power threshold (dB)	-100
PRL	Int	Packet reverb lockout (ms)	50
PTH	Int	Matched filter detector power threshold	180
PTO	1-30	Packet timeout (sec)	14
REV	0 or 1	Whether or not to send the $CAREV message	1
RXA	0 or 1	Whether or not to send the $CARXA message	0

Parameter Name (nnn)	Range	Description	Default
RXD	0 or 1	Whether or not to send the $CARXD message	1
RXP	0 or 1	Whether or not to send the $CARXP message	0
SCG	0 or 1	Set clock from GPS	0
SGP	0-1	Show GPS messages on main serial port	0
SHF	0 or 1	Whether or not to send the $CASHF message	0
SNR	0-1	Turn on SNR stats for PSK comms	0
SNV	0-1	Synchronous transmission of packets	0
SRC (see note 4)	0-127	Default Source Address	0
TAT	0-255	Navigation turn-around-time (msec)	50
TOA	0-1	Display time of arrival of a packet (sec)	0
TXD	Int	Delay before transmit (ms)	250
TXP	0-1	Start of transmit message	1
TXF	0-1	End of transmit message	1
XST	0-1	Turn on transmit stats message, CAXST	1

Important Notes

1. Never change the frequency band (BND) to anything other than the band that the hardware is configured for as this may damage the power amplifier. The factory setting should not be changed unless the hardware is changed.

2. The transmit and receive hopping codes (THC or RHC) must be consistent across the network.

3. BW0 and FC0 can only be set and printed out if BAND is set to 0.

4. Exception to the 7-bit source and destination IDS is the user mini-packet command which uses 4-bit addressing.

CCCFQ – Query configuration parameter, host to modem

Query a modem configuration parameter. Used to check a value or setting on the modem.
$CCCFQ,NNN *CS

NNN   Name of parameter (Table 10), or "ALL" to show all
*CS   Hex coded checksum (8 bit XOR of sentence)

Example: $CCCFQ,SRC
Query the address of the unit. The modem will respond with: $CCCFG,SRC,x*CS where x is the source number. For example: $CCCFG,SRC,1*33.

Selected Configuration Settings for detector lockouts

CRL – cycle reverberation lockout

This is used to lockout the detector after receiving a packet so as to not detect on end-of-packet noise. Defaults to 50 msec.

ECD – End of Cycle Delay

This is used to lockout the detector after transmitting a data or mini packet so as to not detect on outgoing packet reverberation . Defaults to 50 msec.

PRL – Packet reverberation lockout

This is used to lockout the detector after transmitting a cycle init and before setting up the receiver so as to not detect on end-of-packet noise. Defaults to 50 msec.

TXD – Transmit delay

Insert minimum delay between cycle init and packet. Defaults to 600 msec.

Diagnostic and Information Sentences

CABBD – Dump of baseband data to serial port, modem to host

NOTE: This message is used for diagnostic purposes only. If BBD is enabled during normal operation, serial errors may occur. The CABBD message format and behavior may change in future firmware releases.

Modem prints complex, interleaved n-channel PSK data to serial port as a hex string. Available in firmware rev 0.93.0.0+

To enable this message, set the BBD configuration parameter to 1 (by issuing $CCCFG,BBD,1). When this message is enabled, one $CABBD… message will be printed for every packet detected by the modem. The length of this message is not fixed, but it can be determined using the **nwords** argument.

Disable this message once the desired baseband data has been retrieved by setting the BBD configuration parameter to 0 ($CCCFG,BBD,0).

This message terminates with a CRLF character sequence, but does not include the NMEA 0183 checksum.

$CABBD,nwords, <hhhhhhhh…..>

nwords	Twice the number of samples being printed. Each sample is represented as a hex-encoded 32-bit complex value, or 8 hex characters. So, the number of hex characters in this message is equal to **nwords** * 4.
<hhhhhhhh…..>	Baseband samples represented as hex-encoded 32-bit complex values (8 ASCII characters per sample).

CCCFR -- Measure noise level at receiver, host to modem

Message from host to modem used to command the modem to measure the level of signal at the hydrophone in the frequency bands used for communication.

$CCCFR,TTTTT

TTTTT	Time to average in milliseconds

Example: $CCCFR,1000
Command the modem to average for 1 second and report back with the average noise values in the $SNCFR message

SNCFR – Noise report, modem to host

Response from modem to host with noise estimates made in the frequency bands used for communication. The FH-FSK standard uses 7 pairs of frequencies, so there are 14 measurements made and reported to the user.

$SNCFR,<v1>,<v2>,…,<v14>*CS

<v1> to <v14>	Voltages in dB re Volt (dBV)
*CS	Hex coded checksum (8 bit XOR of sentence)

Example: $SNCFR,-117,-116,-116,-116,-112,-88,-111,-115,-115,-113,-115,-115,-115,-111*7F

Frequencies Corresponding to the 14 Bins			
Number	Band A	Band B	Band C
1	7680	12800	23040
2	8000	13120	23360
3	8320	13440	23680
4	8640	13760	24000
5	8960	14080	24320
6	9280	14400	24640
7	9600	14720	24960
8	9920	15040	25280
9	10240	15360	25600
10	10560	15680	25920
11	10880	16000	26240
12	11200	16320	26560
13	11520	16640	26880
14	11840	16960	27200

CACST – Communication cycle receive statistics

Message from modem at the end of a communication packet receive. Used to print out all statistics related to the received packet. Firmware rev 0.92.0.99+

The CST message is subject to change and has a version number starting from firmware rev 0-93.0.52.

$CACST,<Field1>,<Field2>, …, <Field26>*CS

Table 6: Communications Packet Receive Statistics Message Fields

Field number (version 0)	Field number (version 6)	Field Name	Description
N/A	1	Version number	if less than 6, then version 0.
N/A	2	Date	YYYYMMDD
1	4	Mode	0 if good, 1 if bad CRCs, 2 if packet timeout
2	3	TOA time	Time of Arrival
3	5	TOA mode	Clock status
4	6	MFD peak	See $CAMFD description
5	7	MFD power	See $CAMFD description
6	8	MFD ratio	See $CAMFD description
7	9	SPL	See $CAMFD description
8	10	SHF-AGN	See $CASHF description
9	11	SHF-AINPSHIFT	See $CASHF description
10	12	SHF-AINSHIFT	
11	13	SHF-MFDSHIFT	See $CASHF description
12	14	SHF-P2BSHIFT	See $CASHF description
13	15	Rate	Packet rate
14	16	SRC	Packet Source ID
15	17	DEST	Packet destination ID
16	18	PSK Error Code	0. No Error 1. " Bad Modulation header" 2. "Bad CRC, Data Header" 3. "Bad CRC" on any frame 4. "Error accessing coproc" - if modem loses r/w connection to the coproc 5. "Equalizer timeout" - if the coproc never returns 6. "Missed start of PSK packet"
17	19	Packet Type	-1. Unknown 1: FSK 2: FSK Mini 3: PSK 4: PSK Mini
18	20	Nframes	Number of frames expected
19	21	nbad	Number of frames with bad crcs
20	22	SNR-RSS	See $CASNR description
21	23	SNR in	See $CASNR description
22	24	SNR out	See $CASNR description
23	25	Symbols SNR	See $CASNR description
24	26	MSE	Mean Squared Error from Equalizer
25	27	DQF	Data Quality Factor, FSK only

Field number (version 0)	Field number (version 6)	Field Name	Description
26	28	DOP	Doppler
27	29	Stdev noise	Std dev. of noise, dB
28	30	Carrier	Carrier frequency of received packet , Hz
29	31	Bandwidth	Bandwidth of received packet, Hz
N/A	32	PCM	Multi-channel mask, 0 for main board only

CAXST – Communication cycle transmit statistics

Message from modem at the end of a communication packet transmit. Used to print out all statistics related to the transmitted packet. Use NVRAM setting **XST** to turn it on or off. Default setting, ON. Firmware rev 0.93.0.46+

$CAXST,<Field1>,<Field2>, …, <Field26>*CS

Table 7: Communications Packet Transmit Statistics Message Fields

Field number	Field Name	Description
1	Date	YYYYMMDD
2	Time	Time of transmit, hhmmss.sss
3	Timing mode	Clock status
4	Mode	0. Transmit successful 1. FETS too hot 2. Extsync Timeout 3. TX inhibited 4. Data Timeout
5	Probe Length	Length of FM probe, in symbols
6	Bandwidth	Bandwidth of transmitted packet, Hz
7	Carrier	Carrier, Hz
8	Rate	Packet rate
9	SRC	Packet Source ID
10	DEST	Packet destination ID
11	ACK	Ack expected/not expected (1/0)
12	Nframes expected	Number of frames expected, from the CACYC message
13	Nframes sent	Number of frames sent
14	Packet Type	-1. Unknown 1: FSK 2: FSK Mini 3: PSK 4: PSK Mini
15	Nbytes	Total number of data bytes sent

CAMSG – Transaction message, modem to host

Message from modem to host used to inform the user about the status of the acoustic link. At present there are two CAMSG types: bad CRC of a received packet (either cycle init or data), and timeout when an acoustic data packet was requested (or expected) and not received.

$CAMSG, Type, Number*CS

Type	Type of message. Current message types include:
	  BAD_CRC
	  PACKET_TIMEOUT

Number	Number of message. (For BAD_CRC and PACKET_TIMEOUT, this is the packet type. For a list of packet types, see CCCYC – Network Cycle Initialization Command, page 17.)
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Examples:

$CAMSG, BAD_CRC, 2
A cycle-init or data packet was received with a CRC that did not check.

$CAMSG, PACKET_TIMEOUT, 1
The modem timed-out on reception of an acoustic packet. This can occur when the originator makes a request for data and there is no reply, or when a data packet is expected after a cycle-init packet indicates that it is arriving, but it is not received.

CAREV – Software revision message, modem to host

Modem software revision number message from modem to host provided at boot and at the end of a cycle, including a cycle time-out (e.g. after CTO seconds). This message can be enabled or disabled by sending $CCCFG,REV,x, where x is 1 to enable and 0 to disable. Default is enabled.

$CAREV,HHMMSS,IDENT,V.VV.V.VV*CS

HH	Hours
MM	Minutes
SS	Seconds
IDENT	Software identifier string (for example AUV), or INIT at boot time.
V.VV.V.VV	Revision number (for example, 0.89.0.17).
***CS**	Hex coded checksum (8 bit XOR of sentence)

CADQF – Data quality factor information, modem to host

Data qualify factor message from modem to host provided upon reception of an FSK data packet. This message can be enabled or disabled by sending $CCCFG,DQF,x, where x is 1 to enable and 0 to disable. Default is enabled.

$CADQF,DQF,P*CS

DQF	Data quality factor (0-255) for last packet. Values above 200 normally represent decodable packets. Values from 230-250 represent high SNR.
P	Packet type, 1 data packet, 2, mini-packet, e.g. cycle init.
***CS**	Hex coded checksum (8 bit XOR of sentence)

CASHF – Shift information, modem to host

Buffer shift information to convert 12-bit incoming data to full-scale 16-bit data, published upon reception of all communications packets. Higher shifts imply low signal strengths and vice versa. This message can be enabled or disabled by sending $CCCFG,SHF,x, where x is 1 to enable and 0 to disable. Default is disabled.

$CASHF,AGN,AINPSHIFT, AINSHIFT, MFDSHIFT, P2BSHIFT*CS

AGN	Analog gain setting.
AINPSHIFT	Shift of previous input buffer
AINSHIFT	Shift of current input buffer
MFDSHIFT	Shift of the basebanded buffer containing a detection
P2BSHIFT	Shift performed during basebanding incoming data
***CS**	Hex coded checksum (8 bit XOR of sentence)

CAMFD – Comms matched filter information, modem to host

Matched-filter detector message from modem to host provided upon reception of any data packet reception. This message can be enabled or disabled by sending $CCCFG,MFD,x, where x is 1 to enable and 0 to disable. Default is disabled.

$CAMFD,MFPK,MFPWR,MFRATIO,SPL*CS

MFPK	Matched-filter peak value.
MFPWR	MFPK compensated for shift and converted to dB
MFRATIO	Ratio of MFPK and incoherent broad-band noise used for the detector test .
SPL	Sound Pressure Level at receiver in dB re. 1 µPa
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example:

$CAMFD,6441,32,0440, 0138*41

CACLK – Time/Date message, modem to host

Time and date message from modem to host.
$CACLK,YYYY,MM,DD,hh,mm,ss*CS

YYYY	Year
MM	Month
DD	Day
Hh	Hours
Mm	Minutes
Ss	Seconds
***CS**	Hex coded checksum (8 bit XOR of sentence)

CASNR – SNR statistics on the incoming PSK packet

Received signal strength and packet SNR on received PSK comms packets.
Firmware rev. auv-0-92-0-94+
$CASNR,RSS,SNR in,SNR out,Symbol SNR, Noise level*CS

RSS	Received signal strength in dB re. 1 µPa
SNR in	Input SNR, dB, channel 1 only
SNR out	SNR at the output of the equalizer, dB
Symbol SNR	Symbol SNR for spread spectrum packets only, dB
Noise level	Noise level measurement, taken before start of packet,channel 1 only
***CS**	Hex coded checksum (8 bit XOR of sentence)

CADOP – Doppler speed message, modem to host

This message reports the calculated relative speed between the transducer attached to this modem and the transmitting transducer. It is shown only if the DOP configuration parameter is set to 1.
$CADOP,Speed*CS

Speed	Relative speed in meters per second
***CS**	Hex coded checksum (8 bit XOR of sentence)

Example:

$CADOP,0.0*5B

CADBG – Low level debug message, modem to host

Message from modem to host that provides insight into low-level modem processes. This message is not intended for use by end users. Contact the Acoustic Communications Group for more information or assistance with these messages.

$CADBG, Information*CS

Information	Varies
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Error Messages

CAERR – Error message, modem to host

Error messages from modem to host regarding errors that result from serial input by the user. Error messages include things that are typically correctable by the user, such as NMEA parsing and not supplying data to the modem when a data request (CADRQ) is made. CAMSG is used to inform the user about acoustic link status (timeout, bad CRC, etc.).

$CAERR,HHMMSS,Module,NN,Message*CS

HHMMSS	Time of error
Module	Name of the software module reporting the error. For example, all errors related to NMEA messages are labeled "NMEA".
NN	Error Number
Message	Human-readable error message
*CS	Hex coded checksum (8 bit XOR of sentence) optional

Example:
$CAERR,163553,NMEA,12,Unknown command*4B

The user sent an unrecognized NMEA command.

Chapter

2

The AUV Application with LBL Navigation

This application is based on work done for the ONR Very Shallow Water and Surf Zone (VSW/SZ) mine countermeasures project. It provides central control or random access within a potentially dynamic network. Navigation is performed using fixed transponders (the REMUS digital transponders).

Communications is broken up into *cycles* that are described by the *cycle initialization* packet that is typically broadcast by a central controller, which may be fixed or mobile, but may be initiated by any user if desired.

The different actions that may occur during a cycle include communications to and from any two units or groups. A cycle starts with the originator sending a short FM sweep used for initial timing and acquisition, followed by the *cycle initialization* packet (described completely in the $CCCYC command). LBL navigation is done asynchronously on demand from the user (AUV), which may be coordinated via the Master using a cycle-init command if desired.

REMUS-Compatible LBL

LBL navigation using REMUS transponders is done through use of signals compatible with proprietary codes developed by the Oceanographic Systems Laboratory (OSL) at WHOI. The codes are BPSK signals selected by Tom Austin for use in multi-access transponder applications. The REMUS digital transponders are configured to be interrogated by one to four users, and each interrogation is answered with a different code. Thus multiple vehicles can simultaneously use a set of transponders without the requirement to time-share or coordinate.

The transponders are used in groups, typically up to four in a group. Group 1 is transponders labeled A-D. Only one group may be interrogated at a time. If transponders from two different groups are used they must be interrogated separately.

The REMUS transponders use signals in the 20-30 kHz range. The NMEA sentence to command a modem to ping is $CCPDT (Ping Digital Transponder), and is used as shown in the Navigation sentences section of the previous chapter. This command can be used at any time except with the modem is in the middle of a network cycle transaction, i.e. not between the time when a Cycle Init is received ($CACYC) and when a cycle is finished ($CAREV) because the modem (or another one in the network) is then receiving data. The response from the modem to the user after the $CCPDT (Ping Digital Transponder) command is the $SNTTA (Travel Times, Active) message.

For initial configuration make sure that the following parameters are set using $CCCFG command.

Configuring for Ping Digital Transponder Command

Parameter	Setting or Range of Settings
AGN (Analog input gain)	200 to 250 (maps to 25-30 dB)
NPT (Navigation ping power threshold)	50 (water)
NDT (Navigation ping detection threshold)	120-150
MFD (Flag to turn on match filter message)	1 (see SNMFD for information)

Test Notes:
1. When testing in air only test one transponder at a time.
2. Hold the modem transducer and the transponder very close or touching.

AUV Application Example: LBL Navigation

Example 1A: Ping Digital Transponder Command	
Vehicle Host to Modem	**Vehicle Modem to Vehicle Host**
REMUS LBL ping commanded by user (ping transponders A and B from group 1 on interrogation channel 1, with no hardware synch and 3 sec timeout): `$CCPDT,1,1,0,0,3000,1,1,0,0`	*Modem then performs ping*
	Output from modem after ping with travel times from transponders A and B at 12:30:00 `$SNTTA,1.0552,1.2345,,,123000`

Note that a CCPDT command will stop the modem from looking for an incoming data packet that is expected based on a cycle-init command telling the unit to prepare for downlink, but that once the receiver has acquired the incoming packet the CCPDT command will be buffered until the packet reception if finished. Thus, in general the CCPDT command should be used only after a cycle is completed.

Narrow Band LBL

The modem can perform LBL navigation using standard narrow-band pings. The navigation cycle is set up using the $CCPNT command (see Navigation sentences). The modem pings the transponders using the frequency given by the user in the command and listens for returns for <timeout> seconds. After that, it returns the one way travel time from each responding transponder in the $SNTTA (Travel Times, Active) message. The maximum allowable pulse width is 10 milliseconds. All frequencies must be within a 5kHz bandwidth interval.

This command can be used at any time except with the modem is in the middle of a network cycle transaction, i.e. not between the time when a Cycle Init is received ($CACYC) and when a cycle is finished ($CAREV) because the modem (or another one in the network) is then receiving data..

For initial configuration make sure that the following parameters are set using $CCCFG command.

Configuring for Ping Narrow band Transponder Command

Parameter	Setting or Range of Settings
AGN (Analog input gain)	200 to 250 (maps to 25-30 dB)
NPT (Navigation ping power threshold)	50 (water)
NDT (Navigation ping detection threshold)	120-150
MFD (Flag to turn on match filter message)	1 (see SNMFD for information)
TAT (turn around time)	Matched to transponder specifications, typically 3 ms for Benthos transponders.

Example 1B: Ping narrow band LBL Transponder Command

Vehicle Host to Modem	Vehicle Modem to Vehicle Host
Narrowband LBL ping commanded by user (ping two Benthos transponders at frequencies 10.5 kHz and 11 kHz with an interrogation frequency of 10kHz , a pulse length of 10 ms and 3 sec timeout): `$CCPNT,10000,10,10,3000,10500,11000,0,0,1`	*Modem echoes the command, then performs ping*
	Output from modem after ping with travel times from both transponders at 12:30:00 `$SNTTA,1.0552,1.2345,,,123000`

AUV Application Examples: Communication

Example 1-A: Unit 0 downlinks to unit 6, *Unit 0 side*

Surface Host to Surface Modem (0)	Surface Modem (0) to Surface Host
Surface host computer sent: `$CCCYC,1,0,6,0,0,1` $CCCYC,Cmd,Src,Dest,Rate,ACK,#frames	
	Surface modem acknowledges cycle-init command. `$CACYC,1,0,6,0,0,1`
	Surface modem then transmits cycle-init packet
	Surface modem then requests data to transmit: `$CADRQ,134351,0,6,0,32,1` $CADRQ,Time,Src,Dst,ACK,NBytes, Frame#
Surface host provides data with ACK bit 0: `$CCTXA,0,6,0,Requested Data` $CCTXA,Src,Dst,ACK,ascii data	*Surface modem then transmits data packet*
	Or, if the surface host provides no data within the data timeout (DTO), the modem sends an error back over the serial port to the surface host: `$CAERR,DATA_TIMEOUT,3`

Example 1-B: Unit 0 downlinks to unit 6, *Unit 6 side*

Vehicle Host to Vehicle Modem (6)	Vehicle Modem (6) to Vehicle Host
	Vehicle modem receives cycle-init command and tells host about cycle:. `$CACYC,1,0,6,0,0,1` $CACYC,Cmd,Src,Dest,Rate,ACK,#frames
	Vehicle modem then receives data packet.
	If data packet is received successfully, the vehicle modem sends data to its host in this form if RXA flag is set: `$CARXA,0,6,0,1,Requested Data` $CARXA,Src,Dst,ACK, Frame#,ascii data
	And/or, the modem sends data to its host in this form if RXD flag is set and CRC is good: `$CARXD,0,6,0,1,526571756573746564204 4617461*65`
	Or, if the packet is received with bad CRC the vehicle modem sends message to host: `$CAMSG,BAD_CRC,2`
	Or, if no packet is received at all within the data timeout (3 sec), the vehicle modem sends this message to host: `$CAMSG,PACKET_TIMEOUT,3`

Example 2-A: Unit 0 uplinks from unit 6, *Unit 0 side*

Surface Host to Surface Modem (0)	Surface Modem (0) to Surface Host
Surface host computer sent: `$CCCYC,1,6,0,0,0,1` Requesting that unit 6 send data to unit 0. $CCCYC,Cmd,Src,Dest,Rate,ACK,#frames	
	Surface modem acknowledges cycle-init command: `$CACYC,1,6,0,0,0,1`
	Surface modem then transmits cycle-init and starts waiting for a reply from Unit 6.
	If the data packet is received successfully the surface modem sends data to its host in this form if RXA flag is set: `$CARXA,6,0,0,1,Requested Data` $CARXA,Src,Dst,ACK, Frame#,ascii data
	And/or, the surface modem sends data to its host in this form if RXD flag is set and CRC is good: `$CARXD,6,0,0,1,52657175657374656420 44617461*65`
	Or, if the packet is received with bad CRC the surface modem sends message to host: `$CAMSG,BAD_CRC,1`
	Or, if no reply is received at all, the surface modem sends this message to host: `$CAMSG,PACKET_TIMEOUT,2`

Example 2-B: Unit 0 uplinks from unit 6, *Unit 6 side*

Vehicle Host to Vehicle Modem (6)	Vehicle Modem (6) to Vehicle Host
	Vehicle modem receives cycle-init command. `$CACYC,1,6,0,0,0,1`
	Vehicle modem requests data to transmit: `$CADRQ,134351,6,0,0,32,1` $CADRQ,Time,Src,Dst,ACK,#Bytes, Frame#

Vehicle host provides data in ASCII form with CCTXA (here shown with ACK bit 0):
`$CCTXA,6,0,0,Requested Data`
 	
Or, vehicle host provides data in hex-encoded form with CCTXD (here shown with ACK bit 0):
`$CCTXD,6,0,0,52657175657374656420 44617461`
	Vehicle modem acknowledges the CCTXA:
	`$CATXA,6,0,0,14`
	
	$CATXA,Src,Dst,ACK,#Bytes
	or CCTXD:
	$CATXD,Src,Dst,ACK,#Bytes
	Vehicle modem then transmits data packet

	Or, if the vehicle host provides no data, then the modem sends this message over the serial port back to the vehicle host: `$CAERR,NO_DATA,-1`

Example 3-A: Unit 0 downlinks to unit 6, *Unit 0 side, with ACK*

Surface Host to Surface Modem (0)	Surface Modem (0) to Surface Host

Surface host computer sent: `$CCCYC,1,0,6,0,0,1` $CCCYC,Cmd,Src,Dest,Rate,ACK,#frames	
	Surface modem acknowledges cycle-init command. `$CACYC,1,0,6,0,0,1`
	Surface modem then transmits cycle-init packet
	Surface modem then requests data to transmit: `$CADRQ,134351,0,6,0,32,1` $CADRQ,Time,Src,Dst,ACK,NBytes, Frame#
Surface host provides data with ACK bit 1: `$CCTXA,0,6,1,Requested Data` $CCTXA,Src,Dst,ACK,ascii data	*Surface modem then transmits data packet with the ACK bit set high in the packet header*
	In the case where the remote unit successfully receives the data it then sends back an ACK message in a mini-packet, and the ACK is sent out the serial port to the user: `$CAACK,6,0,1,1` $CAACK,Src (of ACK),Dst,Frame#,ACK bit

Example 3-B: Unit 0 downlinks to unit 6, *Unit 6 side, with ACK*

Vehicle Host to Vehicle Modem (6)	Vehicle Modem (6) to Vehicle Host
	Vehicle modem receives cycle-init command and tells host about cycle:. `$CACYC,1,0,6,0,0,1` $CACYC,Cmd,Src,Dest,Rate,ACK,#frames
	Vehicle modem then receives data packet.
	If data packet is received successfully, the vehicle modem sends data to its host in this form if RXA flag is set: `$CARXA,0,6,1,1,Requested Data` $CARXA,Src,Dst,ACK, Frame#,ascii data

	And/or, the modem sends data to its host in this form if RXD flag is set and CRC is good:
	`$CARXD,0,6,1,1,5265717565573746564204`
	`4617461*65`

	Because the ACK bit is set in the acoustic message, the receiving modem automatically transmits the ACK message in a mini-packet back to the originator.

The uplink case with ACK is similar, except that the surface system receiving the data sends back the ACK message, and the sub-sea system is provided the ACK message over the serial port when it is received acoustically. Important note for both cases: An ACK message may have been transmitted, but not received, by the unit requesting the ACK. In this case the mini-packet with the ACK may have a bad CRC.

High-rate (PSK) cycles

High-rate PSK cycles follow the same pattern with one important exception. Cycle init fields are incorporated into the data packet header thus eliminating the need for sending a cycle init packet for downlinking data. Cycle inits are still sent when requesting data from another unit.

PSK Mini-Packets

Mini-packets currently available in PSK:

1. CCCYC - network initialization command

2. CAACK – Acknowledgement of a transmitted packet

3. Command mini-packets:

      a. CCMPC – Ping command

      b. CCMSC – sleep command

      c. CCMEC – external hardware control command

PSK mini-packet usage

To use a PSK mini-packet in the modem's standard bands, set the MOD parameter to 1. This will ensure that all mini-packet transactions are in PSK. For Band 0, the PSK only user band, the value of the MOD bit does not matter. All transactions will be automatically in PSK.

CCCYC: An FSK communications cycle will still use FSK cycle-initializations (cycle-inits), however a PSK communication cycle will use a PSK cycle init if needed. ACKs for a PSK data packet will be sent as PSK ACKS if the modem requesting the ACK is equipped with a coprocessor.

Cycle Timing

There are several different cycle timings to be aware of as summarized below:

CASE 1, Downlink with no ACK. No special time-keeping is required. In this case the initiator of the downlink should just wait a few seconds after sending the data packet to the modem to transmit before continuing to the next cycle. However, navigation may be performed in the period following a cycle so time for that should be allowed, and the amount of time depends partially on the size of the network.

CASE 2, Downlink with ACK. In this case there are two possibilities, one the remote unit will respond with the ACK message, in which case the next cycle can begin, OR, there will be no ACK message and the initiator must re-try. In this case it is up to the user to decide how long to wait for the ACK before re-trying. The amount of time to wait may be computed from: (1) the data packet length (3.5 s), (2) two times the one-way travel time, and (3) one short packet time for the ACK message (1 s). In a 2.25 km network this is 3.5+3+1 = 7.5 s. If applicable, a few seconds should also be allowed for navigation before the next cycle.

CASE 3, Uplink with no ACK. No special time-keeping is required. In this case the user should wait until the data is received correctly, or until there is a CRC error message, or a PKT_TIMEOUT message is received. The Packet Timeout (PTO) should be set to twice the travel-time plus the maximum time

that the remote user will take to supply the requested data. That maximum is specified by Data Timeout, DTO.

CASE 4, Uplink, with ACK. The new cycle can begin after any of the following:

- The timeout has expired on the uplink resulting in the PKT_TIMEOUT message.

- A data packet was received incorrectly, resulting in a BAD_CRC message.

- The data and CAACK messages from the modem are received by the user indicating that the data was successfully received.

Compact Control Language

Using the Micro-Modem with the Compact Control Language CCL developed by Roger Stokey for acoustically communicating with the REMUS vehicle (and its variants) is straightforward. The Compact Control Language was designed for 32 byte packets, and thus maps directly into the lowest rate Micro-Modem packet.

Notes:

1. The binary-encoded hex message command must be used to provide data to the modem for transmission (CCTXD).

2. The RXD flag must be set to 1 (which is the factory default) so that the data messages that are received will be provided to the user in binary-encoded hex in the CARXD message.

To send a CCL message from one system to another the user simply converts the 32 byte binary structure to hex and puts it into a CCTXD message. For example, unit 6 sends to unit 0 without an ACK:

```
$CCTXD,6,0,0,094F43114511461190087D0849088F6A1189FA1187FA11C520C9BE20C9
B620C9
```

This message is 09, the bathymetry message containing three position, depth, altitude, bathymetry measurements as shown in the example in the CCL manual.

References

[1] Freitag, L., M. Johnson, M. Grund, S. Singh and J. Preisig, "Integrated Acoustic Communication and Navigation for Multiple UUVs," *Proc. Oceans 2001*, Honolulu, HI, pp. 2065-2070, 2001.

Index