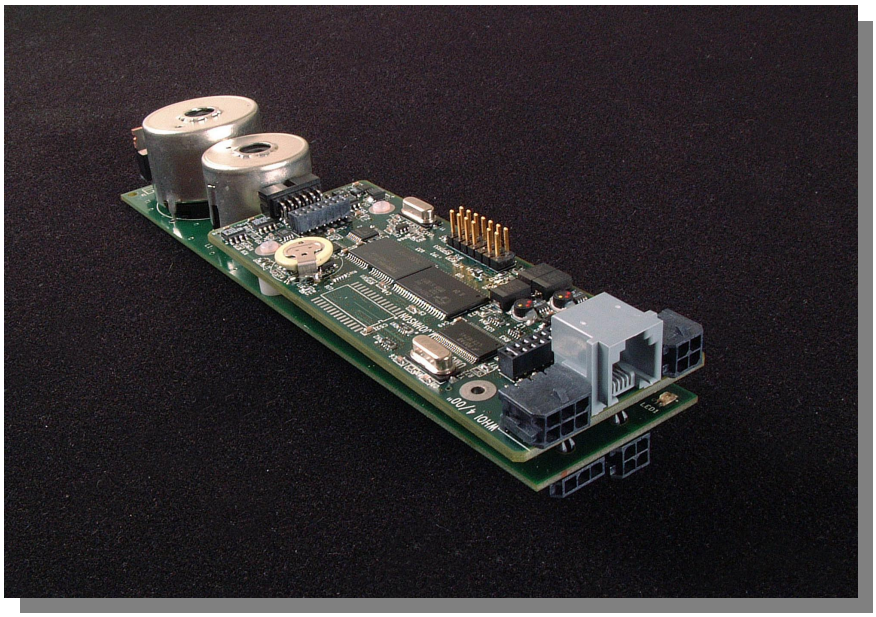
WOODS HOLE OCEANOGRAPHIC INSTITUTION

Acoustic Communications Group



Micro-Modem Software

Interface Guide

ACOUSTIC COMMUNICATIONS GROUP

Micro-Modem Software Interface Guide

Version 2.3 - Revised March 20, 2003

Woods Hole Oceanographic Institution
Woods Hole, MA 02543
Send comments and questions to:
Matthew Grund, mgrund@whoi.edu

This version of the manual corresponds to software release AUV0.73 and higher

Table of Contents

Release Notes for AUV 0.73+ build (since MCM 1.29)

1. Modified CADRQ, CARXD, CARXA, CCTXD, CCTXA to support ACK bit for acknowledged packet transfer and full source and destination fields.
2. Changed error messages.
3. The CAMSG message is now used to inform the user of communications link issues, in particular, if a CRC does not check, and if a time-out has been exceeded on packet reception.
4. Fixed hex-coded data message description (no commas are used).

New sections: *Cycle Timing*

Features Described but not yet Implemented

1. Acknowledgement message CAACK. However, the data request message and the data transmit and receive messages contain the ACK flag and they will not change when the ACK features are added.

2. Multi-frame packet transactions. These will be added when the higher data rates are added.

Chapter 1

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 by the systems programmer. For example, vehicles may be launched and the queried for status at the start of a mission, then left with both vehicles acting as the initiator occasionally. However, as message traffic increases, the probability of one system interfering with the other increases. Use of the acknowledgement capability ensures that important data is delivered with certainty in this case. In many cases there is a clear data flow, for example, vehicles performing initial surveys will pass location data to other vehicles outfitted with different sensors. In that case it makes sense for the survey vehicle to initiate all transactions, and for the other vehicles to transmit only when told to. The target information would be sent with the acknowledgement 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 and no data packet will be transmitted.

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 one data rate, but higher data rates will allow larger packets of data, which may contain multiple frames. Frames are individually protected with a cyclic-redundancy check, and they may be individually acknowledged or re-transmitted. In the FSK system the packet size is 32 bytes for the user and only one frame is used. At the lowest-rate PSK data rate there are 3 32-byte frames per packet. The highest PSK rate has 8 256-byte frames per packet. Multi-frame transactions on the Micro-Modem will be handled in the same way as on the Utility Modem.

Downlink Transaction without Acknowledgement

A downlink is defined as 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 (for example the REMUS control program) 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). Note that ACK is not available for broadcast transmissions.

2. The surface modem then transmits the short Cycle-Init command. All vehicle modems that receive the Cycle-Init report it to their hosts using the CACYC message. This indicates the destination for the data. Only the modem designated as the destination will turn on its receiver to get the data.

3. After the Cycle-Init command is transmitted the surface modem queries the surface host with the CADRQ message for the data to be transmitted and then the surface modem then transmits the data

4. At the sub-sea modem the data packet is received and if decoded correctly (CRC checks) it sends the data frames to the user, one frame per message.

5. At the surface the topside modem receives the data packet and sends the data frames to the user, each in a data message. If any of the frames have the ACK bit set, the topside modem then transmits a short packet with the acknowledgement bits for that packet back to the sub-sea modem.

If a broadcast to all units is desired, then the command type in the CCCYC is set to 0 (unit transmits to all). In that case all of the units receiving the Cycle Init command will turn on their receivers for the incoming data.

Uplink Transaction with Acknowledgement

An uplink is defined as a poll for data originated by the host computer controlling the modem. The sequence of events for an uplink transaction with acknowledgement is as follows:

1. A user (for example the REMUS control program) sends a CCCYC command to the modem to query vehicle N at a particular data rate.

2. The surface modem then sends the Cycle-Init message to vehicle N. The data rate determines how many frames will be necessary to fill a packet.

3. When the Cycle-Init command is received sub-sea the modem first tells the vehicle the contents of the Cycle Init message, then the modem requests data from the vehicle, making one request for each frame of data using the CADRQ command. If the vehicle would like an acknowledgement for that particular frame of data, it sets the ACK bit in the CCTXA message (if ASCII, in the CCTXD message if hex-encoded binary is being transmitted). The modem passes on that bit to the surface.

4. At the surface the topside modem receives the data packet and sends the data frames to the user, each in a data message. If any of the frames have the ACK bit set, the topside modem then transmits a short packet with the acknowledgement bits for that packet back to the sub-sea modem.

5. At the sub-sea side the modem provides the acknowledgement(s) to the user with the CAACK message.

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
$DF	USBL Navigation	Data and configuration for USBL navigation
$YX	Transducer Diagnostics	Data and control for acoustic diagnostics (interference, noise, etc.)
$XB	Transponder Beacon	Data and control for acoustic transponder

NMEA Messages

The NMEA commands and messages are summarized in the following table. Messages high-lighted in gray below are not currently available in the software build noted on the front page of the manual.

Table 2: Micro-Modem NMEA Messages

Message	Description	Applications
$CAACK	Modem reports positive acknowledgement that a frame of data has been correctly received by another unit.	All
$CACFG	Modem reports configuration parameter	All
$CACLK	Modem reports current time (real-time clock)	All
$CACYC	Modem reports current synchronous network command	All communication
$CADBG	Modem reports detailed version number at reset.	All
$CADRQ	Modem requests data from the user.	All Communication
$CADQF	Modem reports decision quality factor for last FSK packet	All Communication
$CAERR	Modem reports an error	All
$SNMFD	Modem reports nav matched-filter detector value(s)	Navigation
$CAMFD	Modem reports comms matched-filter detector value	All Communication
$CAMSG	Modem reports a system information message such as CRC error or time-out.	All Communication
$CAREV	Modem reports application name and revision number	All
$CARXA	Modem reports received data in ASCII format	All Communication
$CARXD	Modem reports received data in binary format	All Communication
$CATXA	Modem confirms receipt of data in CCTXA to transmit	All Communication
$CATXD	Modem confirms receipt of data in CCTXD to transmit	All Communication
$CATXF	Modem reports end of packet transmission	All Communication
$CATXP	Modem reports start of packet transmission	All Communication

Message	Description	Applications
$CCCFG	User sets a modem parameter	All
$CCCFQ	User queries for current value of modem parameter	All
$CCCLK	User sets real-time clock	All
$CCCYC	User commands network cycle	All communication
$CCPDT	User commands ping to REMUS digital transponders.	Active Navigation
$CCPNG	User commands active navigation cycle (ping)	Active Navigation
$CCSNB	User set navigation buoy information	Navigation
$CCTXA	User provides data ASCII data to transmit	Communication
$CCTXD	User provides data hex-encoded data to transmit	Communication
$SNTTA	Navigation receiver reports active navigation travel times	Navigation

Quick-Reference Guide for Commonly Used Messages

Command Summary

$CACYC, Command Type, Transmitter (SRC), Receiver (DEST), Rate, ACK, Number of Packets Command Types: 0 – Unit transmits to All, 1 – Unit transmits to Unit. 2 – Unit transmits to Group. Rate: 0 – 80 bps.
$CADRQ, Time, SRC, DEST, ACK 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

Message from modem to host indicating that an acknowledgement has been received for a previously transmitted data frame or frames.

$CAACK,A…A*CS

A…A	ACK bits, 0 or 1. The ACK bit in the first position corresponds to frame 1 in the transaction, etc.
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Example: $CAACK,100

The first frame of 3 in the previous transaction has been positively acknowledged as being received, while frames 2 and 3 were not. The host can then remove the data corresponding to the successfully transacted frame from the queue or buffer where it was saved pending notification that it had been successfully received at its destination.

CADRQ

Message 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.

$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 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 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, 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,4,6,1,1,4379636c65205465737420467 26f6d20536563757265435254*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.

CCTXD

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 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.

$CCTXD,SRC,DEST,A,HH…HH*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
A	ACK bit, 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

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, 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,Cycle Test From SecureCRT

Acoustic Network Protocol Sentences

CCCYC

Network cycle initialization command transmit message from host to modem, typically the network master. This message sets up transactions as listed in the Network Master Command Table. Note that if the Master is to transmit data, then the modem will send a message requesting the data to the user ($CADRQ). This command can be used by any unit, but there is always the risk of collision if used by multiple units simultaneously. 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 request data from another unit, in which case it puts its own address in ADR2. However, this command can also be used by 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 the applications section for additional examples.

$CCCYC,CMD,ADR1,ADR2,RATE,ACK,Npkt*CS

CMD	Command type
ADR1	Address 1 - Unit or Group # (depends on command type)
ADR2	Address 2 – Unit or Group# (depends on command)
RATE	Packet type. Current supported rates include 0, which is 80 bps FH-FSK.
ACK	Use acknowledge scheme for command (0 or 1).
Npkt	Number of packets in cycle (1 in the current version)
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

Table 3: Network Protocol Commands

CMD	Description	ADR1	ADR2	ACK	NPKT
0	Unit Transmits to All	TX unit	n/a	n/a	1-16
1	Unit Transmits to Unit	TX unit	RX unit	0-1	1-16
2	Unit Transmits to Group	TX unit	RX grp	n/a	1-16
3	<reserved>				
4	Group Transmits to All	TX grp	d/c	n/a	1-16
5	Unit Performs Active NAV	unit	d/c	n/a	n/a
6	<reserved>				
7	<reserved>				

Notes: d/c = Don't care, n/a = not available. unit=node id (0-15), group= group id: up to 16 different groups (0-15) can be configured.

Navigation Sentences

CCPDT

REMUS active LBL navigation ping message from host to modem.

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

GRP	Transponder group designator (normally 1 or 2). Transponders labeled DTxA through DTxD are group 1, DTxE through DTxH are group 2 where x is 1 or 2 and is matched to a particular REMUS.
CODE	Interrogation code. 1-4, normally 1 and 3
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 with code 1, no hardware synchronization, 2.5 second time out, listen for transponders A and B (for example, transponders labeled DT2A and DT2B). When two systems are sharing transponders they should use different interrogation codes. As of January 2003 two interrogation codes are supported by the Micro-Modem and the Digital Transponders, 1 and 3.

SNTTA

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 trip travel time has the REMUS Digital Transponder turn-around time (50 msec) already subtracted.

$SNTTA,tt1,tt2,tt3,tt4,hhmmsss.ss*CS

tt1	Travel time from transponder 1
tt2	Travel time from transponder 2
tt3	Travel time from transponder 3
tt4	Travel time from transponder 4
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 the DT1A transponder is 0.0733 seconds, or 110 meters, and the 1-way travel time to the DT1B transponder is 0.0416 s, or 62 meters.

SNMFD

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.

$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 message from host to modem.

$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)

CCCFG

Set configuration parameter message from host to modem.

$CCCFG,NNN,vv*CS

NNN	Name of 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**

Table 4: Modem Configuration Parameter Summary

Parameter Name (nnn)	Value	Description	Default
SRC	0-15	Default Source Address	1
DST	0-15	Default Destination Address	1
BND	1-3	Frequency Bank (1, 2, 3 for band A, B, or C)	3
RHC	1-3	RX Hopping code (channel number)	1
THC	1-3	TX Hopping code (channel number)	1
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
PTH	Int	MFD Noise Power Threshold	800
DTH	Int	MFD Detect Threshold	175
CSD	Int	Factory setting – do not change	15020
REV	0 or 1	Whether or not to send the $CAREV message	1
DQF	0 or 1	Whether or not to send the $CADQF message	1
AGC	0 or 1	Whether or not to send the $CAAGC message	1
AGN	0-255	Analog Gain (50 is 6 dB, minimum for comms)	50
		Analog Gain (250 is 30 dB, typical for nav)	250
CLK	0 or 1		0
RXA	0 or 1	Whether or not to send the $CARXA message	0
RXD	0 or 1	Whether or not to send the $CARXD message	1
MFD	0 or 1	Whether or not to send the MFD messages	0
SHF	0 or 1	Whether or not to send the $CASHF message	0
NPT	Int	Power threshold for nav detector	600
NDT	Int	Detect threshold for nav detector	40
TXD	Int	Delay before transmit (ms)	0
ECD	Int	Delay at end of cycle (ms)	250
PRL	Int	Packet reverb lockout (ms)	50
CRL	Int	Cycle init reverb lockout (ms)	50
NRL	Int	Navigation reverb lockout (ms)	25
PTO	Int	Packet timeout (sec)	4
CTO	Int	Cycle init timeout (sec)	5

NTO	Int	Navigation reply timeout (sec)	3
TAT	Int	Navigation beacon turn-around time (ms)	50
DTO	Int	Data request timeout (sec)	3

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 always be maintained.

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

CCCFQ

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.

Diagnostic and Information Sentences

CAMSG

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
Number	Number of message
***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

Modem software revision number message from modem to host provided at boot and at the end of a cycle. 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,vv.vv*CS

hh	Hours
mm	Minutes
ss	Seconds
ident	Software identifier string (for example MCM or AUV)
vv.vv	Revision number
***CS**	Hex coded checksum (8 bit XOR of sentence)

CADQF

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*CS

DQF	Data quality factor (0-255) for last packet. Values above 200 normally represent decodable packets. Values from 230-250 represent high SNR.
***CS**	Hex coded checksum (8 bit XOR of sentence)

CAMFD

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*CS

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:

$CAMFD,10927,3006,0363

CACLK

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)

CAERR

Error messages from modem to host. 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,XXXX,NN*CS

HHMMSS	Time of error
XXXX	Name of warning, for example, NO_DATA, or the name of the reporting module in the case of NMEA warnings.
NN	Warning Number
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

Example: $CAERR,041318,N0_DATA,2,*38

The modem made a request for data from the user but none was supplied within the data timeout setting (DTO).

Table 5: Error Numbers for NMEA

Module Name	Code	Description
NMEA		NMEA message handler
	9	No error
	11	Checksum error
	12	Unknown command
	13	Sync error: “$” not found
	14	Syntax error: separator “,” not found
	15	Wrong argument count
	16	Checksum syntax: “*” not found
	17	Argument value out of range
	18	Required argument is blank
	19	Termination error
	20	Unknown Talker ID

Chapter 4

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 transponder boards 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, group 2 is E-H. Only one group may be interrogated at a time. If transponders from two different groups are used they must be interrogated separately.

This feature is only available with the Band C hardware and firmware because 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)	300-800 (air) 300 (water)
NDT (Navigation ping detection threshold)	40-60 (start with 45)
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 1: 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.

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` $CCCYC,Cmd,Src,Dest,Rate,ACK,#pkts	
	Surface modem acknowledges cycle-init command. `$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` $CACYC,Cmd,Src,Dest,Rate,ACK,#pkts
	Vehicle modem then receives data packet.
	If data packet is received successfully, the vehicle modem sends data to its host in this form if RXA flag is set: `$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,0,6,0,1,526571756573746564 2044617461*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. $CCCYC,Cmd,Src,Dest,Rate,ACK,#pkts	
	Surface modem acknowledges cycle-init command: `$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,6,0,0,1,526571756573746564 2044617461*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`

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:

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

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

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

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