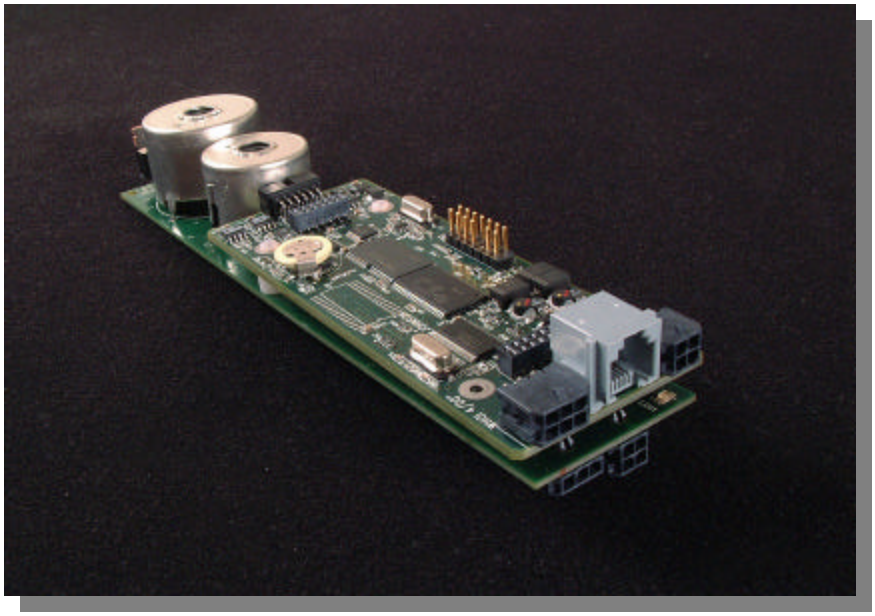
WOODS HOLE OCEANOGRAPHIC INSTITUTION

Acoustic Communications Group



Micro-Modem User's Guide

ACOUSTIC COMMUNICATIONS GROUP

Micro-Modem User's Guide

Version 2.01 - Revised March 12, 2003

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

Table of Contents

Table References

Chapter

1

The Micro-Modem

The WHOI Micro-Modem is a versatile low-power embedded signal-processing platform with a collection of hardware peripherals and software applications allowing it to fill many underwater acoustic needs such as underwater acoustic modem, acoustic navigation beacon, acoustic network node, and passive navigation device.

Figure 1: WHOI Micro-Modem with power amp and pressure housing.

The WHOI Micro-Modem is a collection of hardware and software solutions, organized around a signal processing core and a real-time operating system. The modem motherboard relies on the Texas Instruments TMS320C5410 DSP, together with a high speed 12 bit analog to digital (A2D) converter, and other peripherals such as FLASH memory, a low rate A2D, real time clock, and 2 serial ports. The modem hardware set also includes several different class-D power amplifier designs, an external 1.2GB FLASH board, Lithium ion batteries and an external 8 channel high-speed A2D board. All boards are designed to fit in a 2”ID water-tight housing.

The modem software consists of a real-time signal-processing core, developed at WHOI specifically for the TMS320C5410, and a user specific application, which may enable some or all of the communication and navigation capabilities. In general, applications are written to use an RS-232 serial user interface and NMEA messaging. Several applications ranging from a simple low-rate point-to-point modem to multi-user networked communication and navigation solution are available, and new applications are constantly being developed.

Hardware Specifications

The board set, transducers, batteries and housings are designed and assembled at WHOI. This means that they are optimized for at-sea deployments: having a small footprint, simple, robust design, and low power consumption. However, production times can be long, and the number of units available is small. Modem configurations vary considerably with desired features.

Table 1: Available Micro-Modem Hardware Components

Component	Capabilities	Needed	Power (Max.)	Size
Micro-Modem mainboard	DSP, power supply, RS-232 ports, 1 channel receiver	Required	200mW	4.2”x1.75”x0.6”
Power Amplifier	1 channel transmit	Optional	30 W	7.25”x1.3”x1.5”
1.2GB flash card	Data storage, raw acoustic sampling	Optional	50mW	3.5”x1.5”x0.25”
8 channel A2D card	4 or 8 channel receiver, high speed sampling (250KHz)	Optional	200mW	4.4”x1.5”x0.6”

Micro-Modem Main Board

The Micro-Modem main board is the signal-processing core of the micro-modem. It also provides the serial user interface, single channel receiver, and supplies power for all daughter cards (excluding the high current output rails for the power amp card).

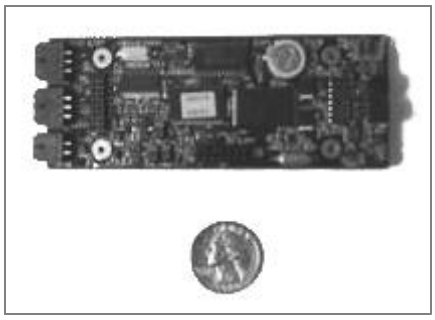


Figure 2: Micro-Modem Main board version 1.1.

Availability

The Micro-Modem main board version 1.1 is currently available. Version 1.2 will be designed and fabricated in early 2003. Enhancements include broad input voltage range for the power supply, 1 pulse-per-second input, and faster DSP. The following table summarizes version 1.1 specifications.

Table 2: Micro-Modem Mainboard Specifications

Version	1.1
Power input	5-17 V, 200 mW
Processor	Texas Instruments TMS320C54010 96MHz or 54016.
Size	4.2”x1.75”x0.6”
Interface	2 RS-232 serial ports, 119200bps (max)
Storage	64KW RAM (on chip), 256KW SRAM (external), 1MB Flash, 32KB EPROM, 96 byte NVRAM
Receiver	1 channel 12-bit A2D 93kHz max sample rate

Micro-Modem Power Amplifier

The Micro-Modem power amplifier card provides transmit capabilities. It also provides transmit/receive (T/R) switching and additional input amplification, when a single transducer is used for transmit and receive.

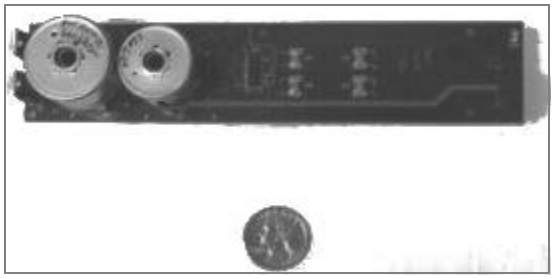


Figure 3: Micro-Modem Power Amplifier version 1.0.

The amplifier output stage has an inductor and step-up transformer that must be matched to amplifier input voltage, and transducer type.

Availability

The Micro-Modem power amplifier version 1.0 and 1.1 are currently available. Each board must be fitted with electromagnetic matching components. An inductor and transformer are designed and fabricated at WHOI, to match the amplifier and transducer optimally.

Table 3: Micro-Modem Power Amplifier Specifications

Version	1.0	1.1
Power input	12-17 V, 20W transmit,	12-32 V
Size	7.25”x1.3”x1.5”	7.25”x1.3”x1.5”

Micro-Modem External FLASH Card

The Micro-Modem external FLASH storage card provides additional storage capabilities. The card can be fitted with up to 1.2GB of storage. The card is typically used to store acoustic data in streaming fashion, or log raw received packets. Currently the flash cards are used with a custom application that rewards and stores passband data based on user commands, or on a pre-programmed schedule. A future software release will integrate this capability with the standard application.

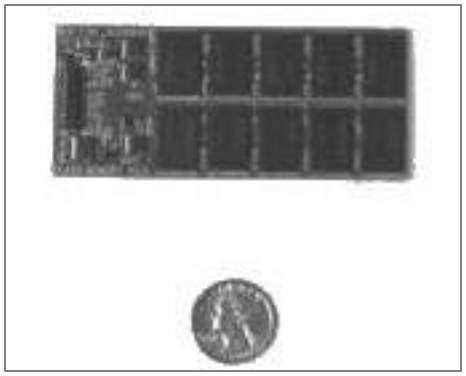


Figure 4: Micro-Modem External FLASH Card version 1.0.

Availability

The Micro-Modem external FLASH storage card version 1.0 is currently available.

Table 4: Micro-Modem External FLASH Card Specifications

Version	**1.0**
Power	50mW read, program, erase 0.5mW standby
Storage	1.2 GB
Size	3.5”x 1.5” x 0.25”

Micro-Modem 8 Channel A2D Card

The Micro-Modem 8 channel Analog to Digital (A2D) card provides additional receiver channels and high rate sampling capabilities. The card can operate in either a 4 channel or 8 channel mode. The card is intended to support high rate PSK communication, USBL navigation applications, and diversity reception for FH-FSK.

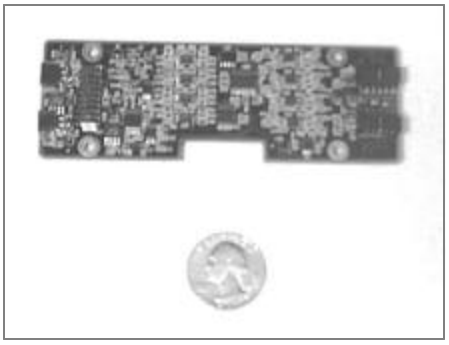


Figure 5: Micro-Modem 8 channel high speed A2D card version 1.0.

Availability

The Micro-Modem 8 channel Analog to Digital (A2D) card is version 1.0 is currently available. The device driver software is still under development. The card is expected to be operational by December 2002.

Table 5: Micro-Modem 8 Channel A2D Card Specifications

Version	1.0
Power	200mW
Receiver	4 channels at 250KHz, or 8 channels at 125KHz
Size	4.4”x1.6”x0.6”

Chapter 2

Applications

WHOI Micro-Modem applications are constantly under development. Sometimes, if an existing application does not suit your needs, an application can be created quickly by pasting together features that already exist in other applications. Other times a new application will require a new feature, which can then be added to the library of existing features, for use by other applications.

The following table is a snapshot of current applications in use, and under development.

Table 6: Application List

APPLICATION NAME	VERSION No.	STATUS
MCM passive	1.30	Beta Test
Comms	1.0	Fielded
MCM Active	0.9	Development

Chapter

3

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 is designed to be polled (master-slave) for communication, and asynchronous for active navigation. 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 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.

Sub-sea: A computer connected to a slave modem will receive notification that an incoming Cycle Initialization command has been detected with the $CADET message. The information in the cycle init command that indicates what the current cycle is for is then provided in the $CACYC message. If there is incoming data, it is provided to the user in the $CARXA (ASCII) or $CARXD (binary) messages, depending on which has been enabled. If the master is requesting data, the sub-sea 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 master.

On either side the current configuration is set using $CCCFG, and examined using $CCCFQ. To set up a network one unit is set to have source address 0 (master), and the others use numbers from 1 to 15.

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.

Table 7: 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 WHOI Micro-Modem uses several different NMEA commands and messages as listed in the following table and described below.

Table 8: Micro-Modem NMEA Messages

Message	Description	Applications
$CACFG	Modem reports configuration parameter	All
$CACLK	Modem reports current time (real-time clock)	All
$CACYC	Modem reports current synchronous network command	Network based communication
$CADET	Modem reports signal detection	All Communication, Navigation
$CADQF	Modem reports decision quality factor for last FSK packet	All Communication
$CAERR	Modem reports an error	All
$CAREV	Modem reports application name and version	All
$CARXA	Modem reports received data in ASCII format	All Communication
$CARXD	Modem reports received data in binary format	All Communication
$CATXF	Modem reports end of packet transmission	All Communication
$CATXP	Modem reports start of packet transmission	All Communication
$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 synchronous network cycle	Network Communications, passive navigation
$CCPDT	User commands ping to REMUS digital transponding	Active Navigation

$CCPNG	User commands active navigation cycle (ping)	Active Navigation
$CCSNB	User set navigation buoy information	Navigation
$CCTXA	User provides data for transmit in ASCII format	Communication
$CCTXD	User provides data for transmit in binary format	Communication
$SNBXY	Navigation receiver reports beacon referenced position fix	Navigation
$SNGLL	Navigation receiver reports latitude and longitude	Navigation
$SNTTA	Navigation receiver reports active navigation travel times	Navigation
$SNTTP	Navigation receiver reports passive navigation travel times	Navigation
$XBPNG	Navigation transponder beacon reports ping transmission	Navigation

Acoustic Communications Sentences

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 approximately 3 seconds, a timeout error message is sent by the modem and no acoustic reply is transmitted back to the requester.

$CADRQ,hhmmss,ss,dd,n*CS

hhmmss	Time of Request
ss	Sender address (0-15)
dd	Destination address (0-15)
n	Number of bytes requested by modem (max)
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Example: $CADRQ,000331,6,0,32*43

At time 00:03:31 unit 6 is requested by unit 0 to send up to 32 bytes of data.

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 $CAERR,CRC,-1 is sent to the host.

$CARXA, ss,dd,m…m*CS

ss	Sender address(0-15)
dd	Destination address (0-15)
m…m	ASCII text string (1-32 bytes)
***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: $CARXA,6,0,Test From Node*71

Unit 6 has sent the message ‘Test from Node’ to unit 0 and the RXA flag is set and thus the message is printed out in ASCII.

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 $CAERR,CRC,-1 is sent to the host.

$CARXD,ss,dd,HH…HH*CS

Ss	Sender address (0-15)
dd	Destination address (0-15)
HH	Hex coded data bytes, 2 characters each, eg. 00-FF to represent binary numbers from 0-256 (1-32 bytes)
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Example: $CARXD,6,0,546573742046726f6d2042756f79*0F

Unit 6 has sent the message ‘Test from Node’ to unit 0, and the RXD (hex encoded binary message) flag is set and thus the message is printed out in hex.

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 $CCDRQ (data request) message.

$CCTXD,ss,dd,HH,…,HH*CS

ss	Sender address (0-15)
dd	Destination address (0-15)
HH,…,HH	Hex coded data bytes, 2 characters each, e.g. 00-FF (1-32 bytes)
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

Example:
$CATXD,6,0,54,65,73,74,20,46,72,6f,6d,20,42,75,6f,79*0F

Unit 0 has sent a request for data to unit 6, the modem has received that request acoustically and has asked the unit to provide data. The data is provided in this format when binary data is to be transferred.

CCTXA

ASCII data message from host to modem for transmission. This message (or alternatively,CCTXD) is sent by the host to the modem in response to the $CCDRQ (data request) message.

$ CCTXA,ss,dd,mmmm*CS

Ss	Source Address (0-15)
Dd	Destination Address (0-15)
Mmmm	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: $CATXD,6,0,Test From Node*71

Unit 0 has sent a request for data to unit 6, the modem has received that request acoustically and has asked the unit to provide data. The data is provided in this format when ASCII data is to be transferred.

Acoustic Network Communications Sentences

CCCYC

Network cycle initialization command transmit message from host to modem 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).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 are: 5 – 80 bps 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 9: Network Master 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, G, C, SF, STO, TO, AF, BF, CF, DF*CS

G	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.
C	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
TO	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, 3000, 1, 1, 0, 0

Ping transponder group 1 with code 1, no hardware synchronization, 3 second time out, listen for transponder A and B (for example, transponders labeled DT2A and DT2B). When two systems are sharing transponders they should use different interrogation codes.

SNTTA

Active LBL navigation one-way travel time message from modem to host. 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, 1.0552,1.2345, , , 123000

SNTTP

Passive LBL navigation travel time message from modem to host.

$SNTTP,tt1,tt2,tt3,tt4,hhmmss.ss*CS

tt1	Travel time for path: master ⇒ buoy 1 ⇒ client
tt2	Travel time for path: master ⇒ buoy 2 ⇒ client
tt3	Travel time for path: master ⇒ buoy 3 ⇒ client
tt4	Travel time for path: master ⇒ buoy 4 ⇒ client
hhmmss.ss	Time of ping (Modem time from real-time clock)
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

SNGLL

Global-referenced navigation position message from modem to host.

$SNGLL,ddmm.mmm,N,dddmmm.mmm,E,hhmmss.ss,N,M*CS

ddmm.mmm	Latitude (degrees and decimal minutes)
N	Latitude hemisphere (N or S)
dddmm.mmm	Longitude (degrees and decimal minutes)
E	Longitude hemisphere (E or W)
hhmmss.ss	UTC of position fix
N	Number of transponders
M	Mode Indicator: A = Active mode P = Passive mode
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

SNBXY

Beacon-referenced navigation position message from modem to host.

$SNBXY,xxxx.xx,SX,yyyy.yy,SY,hhmmss.ss,N,M*CS

xxxx.xx	Meters in X direction†
SX	Sign of X position (+ or -)
yyyy.yy	Meters in Y direction†
SY	Sign of Y position (+ or -)
hhmmss.ss	UTC of position fix
N	Number of transponders
M	Mode Indicator: A = Active mode P = Passive mode
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

†Note: the X direction is the vector from the lowest numbered beacon, to the next beacon. Y is perpendicular to X with positive being 90° counter-clockwise from X.

XBPNG

Notification from modem to host that a ping has been transmitted in response to an interrogation.

$XBPNG,dd,mm,hhmmss.ss*CS

Dd	Beacon address
Mm	Beacon mode: 00 - classic narrow band LBL 01 - MCM network beacon 02 - REMUS beacon 03 - USBL beacon (FM sweep transmit)
hhmmss.ss	Time sent
***CS**	Hex coded checksum (8 bit XOR of sentence)

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 10)
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.

Table 10: Modem Configuration Parameter Summary (MCM Version 1.29 and AUV Version 1.0)

Parameter Name (nnn)	Value	Description	Default
SRC	0-15	Default Source Address	
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	150
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 units is approximately 6 dB)	50
CLK	0 or 1	Whether or not to send the $CACLK message	0
RXA	0 or 1	Whether or not to send the $CARXA message	1
RXD	0 or 1	Whether or not to send the $CARXD message	1
MPF	0 or 1	Whether or not to send the $CAMFD message	0
SHF	0 or 1	Whether or not to send the $CASHF message	0

Table 11: Modem Configuration Parameter Summary (AUV Version 1.0)

Parameter Name (nnn)	Value	Description	Default[1]
NPT	Int	Power threshold for nav detector	300
NDT	Int	Detect threshold for nav detector	500
TXD	Int	Delay before transmit (ms)	250
ECD	Int	Delay at end of cycle (ms)	150
PRL	Int	Packet reverb lockout (ms)	50
CRL	Int	Cycle init reverb lockout (ms)	50
NRL	Int	Navigation reverb lockout (ms)	50
PTO	Int	Packet timeout (sec)	4
CTO	Int	Cycle init timeout (sec)	5
NTO	Int	Navigation reply timeout (sec)	10
TAT	Int	Navigation beacon turn-around time	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.

[1] New parameters available in this version in addition to Table 10.

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 Sentences

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)

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 message from modem to host.

$CAERR,hhmmss,XXXX,nn*CS

hhmmss	Time of error
XXXX	Reporting module name: i.e., NMEA, SIO, NAV, etc.
Nn	Error code
***CS**	Hex coded checksum (8 bit XOR of sentence) optional

Table 12

: Error Codes

Module Name	Error 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
SYS		System monitor
	1	Low battery voltage
	2	Clock not set
	3	Modem Settings not set
CRC		Packet Decoder
	1	Bad Checksum

Chapter

4

The MCM and AUV Applications with Navigation

The MCM application is based on work done for the ONR Very Shallow Water and Surf Zone (VSW/SZ) mine countermeasures project. It provides central control of a fixed network, and navigation is performed using broadcasts made from fixed, known locations. The operation of the system, and details of navigation solutions and accuracy is provided in [1]. The AUV application is essentially the same, but includes standard LBL navigation instead of the passive navigation capability. With the exception of the navigation differences the applications are very similar.

All of the communications activity in the network is controlled by the user from the Master node. Access is broken up into *cycles* that are described by the *cycle initialization* packet that is broadcast by the Master. In the MCM application each cycle also includes a passive navigation fix.

The different actions that may occur during a cycle include communications to and from any two units or groups. A cycle starts with the Master node sending a short FM sweep used for initial timing and acquisition, followed by the *cycle initialization* packet (described completely in the $CCCYC command). In the passive navigation mode the fixed nodes in the system are called *secondary nodes*. Each of them transmits a specific code at a fixed delay after the FM sweep is detected. All of these signals are received by the *clients* and the time differences of arrival between the Master's FM sweep and the Secondary nodes signals are used to provide a position fix. In the plain AUV application 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.

Each unit in the network has several states and the operation of the units can be described in terms of those states. Table 13 contains a summary of the states that the Client and Secondary nodes can take. Note that a cycle is defined as the time from when the FM signal is sent by the Master to initiate a passive navigation sequence to when the Client has completed uplink or downlink as controlled by the Cycle Initialization packet.

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 $CCPDT (Ping Digital Transponder), and is used as shown in the Navigation sentences section of the previous chapter. This command can be used at any time except with the modem is in the middle of a network cycle transaction, i.e. not between the time when a Cycle Init is detected ($CADET) and when a cycle is finished ($CAREV) because the modem 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.

Table 13: MCM Application States

Client States (5<=id<=15)			
#	Description	Message upon entering this state	Valid NMEA Commands
0	Waiting for Cycle Init Packet	$CAREV (revert to state 0)	$CCCLK,$CCCFQ,$CCCFG
1	Decoding CI packet	$CADET (detect)	None
2	Receiving NAV pings	$CADQF	$CCCLK,$CCCFQ,$CCCFG
3	Requesting Data from host	$CADRQ (data request)	$CCTXD (data reply to $CADRQ $CCCLK,$CCCFQ,$CCCFG Note: sending these will abort the cycle
4	Transmitting Data Packet	$CATXP (transmitting packet)	None
5	Receiving Data Packet	$SNTTP (if RX commanded in CI)	$CCCLK,$CCCFQ,$CCCFG Note: sending these will abort the cycle
6	Error Occurred	$CAERR	none
7	Reconfiguring	$CACFG	none
8	Decoding Packet (second half of #5)	$CADET	none

Secondary States (1<=id<=5)			
#	Description	Message upon entering this state	Valid NMEA Commands
0	Waiting for Cycle Init Packet	$CAREV (revert to state 0)	$CCCLK,$CCCFQ,$CCCFG
1	Transmitting Reply	$CADET (detect)	none
2	Error Occurred	$CAERR	none
3	Reconfiguring	$CACFG	None

MCM and AUV Application Examples: Master

The following examples apply to both the MCM and AUV application with the exception that the SNTTP messages do not appear in the AUV application.

Example 1: Master broadcasts to all, rate 5 with 7 data bytes (values ff-f9) encoded in hex.

Modem Input and Explanation	Modem Output
User sent: `$CCCYC,0,0,0,5,0,0,ff,fe,fd,fc,fb,fa,f9`	
Modem acknowledges cycle-init command.	`$CACYC,0,0,0,5,0,0,7*45`
Modem starts transmitting	`$CATXP,2*40`
Modem finishes transmitting	`$CATXF,2*56`
No travel times reported	`$SNTTP,,,,,,104605.00*65`
Packet transmit starts	`$CATXP,1*43`
Packet transmit ends	`$CATXF,1*55`
Cycle finished	`$CAREV,104627,MCM,1.09*3C`

Example 2: Master downlinks to unit 6, rate 5 with 7 data bytes (values ff-f9) encoded in hex.

Modem Input and Explanation	Modem Output
User sent: `$CCCYC,1,0,6,5,0,0,ff,fe,fd,fc,fb,fa,f9`	
Modem acknowledges cycle-init command.	`$CACYC,1,0,6,5,0,0,7*42`
Modem starts transmitting	`$CATXP,2*40`
Modem finishes transmitting	`$CATXF,2&56`
No travel times reported	`$SNTTP,,,,,,104627.00*65`
Packet transmit starts	`$CATXP,1*43`
Packet transmit ends	`$CATXF,1*55`
Cycle finished	`$CAREV,105139,MCM,1.09*35`

Example 3: Master attempts uplink from unit 6, rate 5 with no reply.

Modem Input and Explanation	Modem Output
User sent: `$CCCYC,1,6,0,5,0,0`	
Modem acknowledges cycle init command.	`$CACYC,1,6,0,5,0,0,0*45`
Modem starts transmitting	`$CATXP,2*40`
Modem finishes transmitting	`$CATXF,2*56`
No travel times reported	`$SNTTP,,,,,,105139.00*6C`
No data received from unit	`$CAERR,105424,RX_TIMEOUT,1*52`
Cycle finished	`$CAREV,105425,MCM,1.09*3D`

Example 4: Master uplinks from unit 6, rate 5, good data received.

Modem Input and Explanation	Modem Output
User sent: `$CCCYC,1,6,0,5,0,0`	
Modem acknowledges cycle init command.	`$CACYC,1,6,0,5,0,0,0*45`
Modem starts transmitting	`$CATXP,2*40`
Modem finishes transmitting	`$CATXF,2*56`
No travel times reported	`$SNTTP,,,,,,105139.00*6C`
Data packet detected	`$CADET,DATAPKT,2,29624,2458,0*03`
Data quality factor from decoder (high)	`$CADQF,244*4F`
Data reported in ascii	`$CARXA,6,0,Test From Node*71`
Data reported in hex	`$CARXD,6,0,546573742046726f6d2042756f79*0F`

Example 5: Master uplinks from unit 6, rate 5, data detected, but not decoded.

Modem Input and Explanation	Modem Output
User sent: `$CCCYC,1,6,0,5,0,0`	
Modem acknowledges cycle init command	`$CACYC,1,6,0,5,0,0,0*45`
Modem starts transmitting	`$CATXP,2*40`
Modem finishes transmitting	`$CATXF,2*56`
No travel times reported	`$SNTTP,,,,,,105139.00*6C`
Packet detected	`$CADET,DATAPKT,2,29624,2458,0*03`
Quality factor from decoder (low, <200)	`$CADQF,178*4F`
Packet has bad CRC	`$CAERR,CRC,-1*09`

MCM Application Examples: Client

Example 1: Client uplink data request with timeout from user.

Modem Input and Explanation	Modem Output
Client modem ready	`$CAREV,000302,MCM,1.07*35`
Detect cycle init	`$CADET,CYCLEINIT,46,32767,0,1*05`
No replies from secondary nodes	`$SNTTP,,,,,,000302.00*62`
Got a Cycle Init command	`$CACYC0,6,0,0,032,5302*44`
Data request sent to user	`$CADRQ,000315,6,0,32*45`
Timeout since the user didn't provide data	`$CAERR,000318,NODATA,3*43`

Example 2: Client uplink data request with data from user.

Modem Input and Explanation	Modem Output
Client modem ready	`$CAREV,000318,MCM,1.07*3E`
Detect cycle init	`$CADET,CYCLEINIT,4,32767,0,1*33`
No replies from secondary nodes	`$SNTTP,,,,,,000318.00*69`
Got a Cycle Init command	`$CACYC,0,6,0,0,0,32,5302*44`
Data request sent to user	`$CADRQ,000331,6,0,32*43`
User types: `$CCTXA,6,0,Test`	
Modem echos data	`$CATXA,6,0,Test`
Transmit starts	`$CATXP,1*43`
Transmit ends	`$CATXF,1*55`

AUV Application Examples: LBL Navigation

Example 1: Client uplink data request with timeout from user.

Modem Input and Explanation	Modem Output
Client modem ready	`$CAREV,000302,MCM,1.07*35`
REMUS LBL ping commanded by user (ping transponders A and B from group 1 on interrogation channel 1, with no hardware synch and 3 sec timeout): `$CCPDT,1,1,0,0,3000,1,1,0,0.`	
Output from modem after ping with travel times from transponders A and B at 12:30:00	`$SNTTA,1.0552,1.2345,,,123000`

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