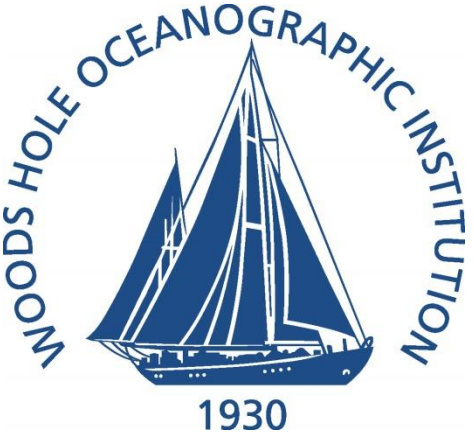
Micromodem-2 User's Guide

Acoustic Communications Group



Version number: 1.2

Table of Contents

Overview 1

New Features in the Micromodem-2 2

Hardware Improvement Highlights 2

More Information 2

Upgrading from the Micromodem-1 3

Hardware Differences 3

Software Interface Differences 3

Getting Started with Acoustic Communication 4

Modem Setup 5

NMEA 0183 Sentences 6

NMEA Messages 6

Hardware Interface 10

User Connections 11

Pinouts 11

Mating Connectors 12

Electrical Specifications 14

Absolute Maximum Ratings 14

Operating Conditions 14

Software Interface 15

Acoustic Communications Sentences 16

CAACK – Acknowledgment of a transmitted packet 16

CADRQ – Data request message, modem to host 16

CARXD - Received binary message, modem to host 17

CARXP – Incoming packet detected, modem to host 17

CCTXD – Transmit binary data message, host to modem 17

CCTXP – Start of packet transmission, modem to host 18

CCTXF – End of packet transmission, modem to host 18

Sending Sweeps and Tones 18

Sending M-Sequences 19
Passthrough Messages 19
NMEA API Level 22
CCALQ 22
CAALQ 22
Acoustic Network Protocol Sentences 23
CCCYC – Network Cycle Initialization Command 23
CACYC – Echo of Network Cycle Initialization command 23
Ping Sentences 24
CCMPC – Mini-Packet Ping command, host to modem 24
CAMPC – Echo of Ping command, modem to host 24
CAMPA – A Ping has been received, modem to host 24
CAMPR – Reply to Ping has been received, modem to host 25
CCRSP – Pinging with an FM sweep 25
CARSP – Response to FM sweep ping command 25
Recording Raw Data 26
CCREC 26
CARCI 27
CAREP 27
Example 28
CACFT – Fathometer ping reply report 30
CACIR – Fathometer ping reply IRE capture 30
Sleep and Hibernate Sentences 32
Hibernating 32
Sleep 37
External Hardware Line Control Sentences 39
CCEXL – External hardware control command, local modem only 39
CCMEC - External hardware control command 39
CAMEC – Echo of hardware control command, modem to host 40
CAMEA – Hardware control command received acoustically 41
CAMER – Hardware control command reply received 41
Packets, Rates, Frames and Acknowledgement 42

Flexible Data Protocol (FDP) 44
CCTDP- transmit (downlink) data packet 44
CATDP - Response to CCTDP command 44
CARDP- Reception of a FDP downlink data packet. 45
FDP Minipacket rates 46
User Mini-Packet Sentences 48
CCMUC – User Mini-Packet command, host to modem 48
CAMUC – Echo of user Mini-Packet, modem to host 48
CAMUA – Mini-Packet received acoustically, modem to host 48
CAMUR – Reply to Mini-Packet received, modem to host 49
Navigation Sentences 50
CCPGT – Ping Generic transponder, host to modem 50
CCPDT – Ping REMUS digital transponder, host to modem 51
CCPNT – Ping narrowband transponder, host to modem 52
SNTTA – Transponder travel times, modem to host 53
SNMFD – Nav matched filter information, modem to host 53
Diagnostic and Information Sentences 54
CABBD – Dump of baseband data to serial port, modem to host 54
CCCFR -- Measure noise level at receiver, host to modem 54
SNCFR -- Noise report, modem to host 55
CACST – Communication cycle receive statistics 55
CAXST – Communication cycle transmit statistics 58
CAMSG – Transaction message, modem to host 58
CAREV – Software revision message, modem to host 59
CADQF – Data quality factor information, modem to host 60
CASHF – Shift information, modem to host 60
CAMFD – Comms matched filter information, modem to host 60
CACLK – Time/Date message, modem to host 61
CASNR – SNR statistics on the incoming PSK packet 61
CADOP – Doppler speed message, modem to host 62
CADBG – Low level debug message, modem to host 62
BBD, Baseband data dump configuration parameter 62

SNNST 64
Configuration Sentences 66
CCCFG – Set configuration parameter, host to modem 66
CCCFQ – Query configuration parameter, host to modem 67
Selected Configuration Settings for detector lockouts 84
Synchronous Navigation 85
Using and Setting the Clock 86
CCTMS 86
CATMS 86
Examples 87
CCTMQ 87
CATMQ 88
CATMG 88
CCCLK – Set clock, host to modem 89
Firmware Update 91
UPMFW 92
UPMFWA 92
UPDAT 93
UPDATA 93
UPERR 93
UPDONE 94
pyAcomms Support 95
Acoustic NMEA command 96
CCACM 96
CAACM 96
CAACA 97
CAACR 97
USBL navigation 98
USBL Navigation Configuration Parameters 99
CCXSB 100
CAXSB 101
CCRSB 102

CAUSB 103
CAUIR 103
CAUXY 103
Error Messages 104
CAERR – Error message, modem to host 104
ModemTMA 105
pyAcomms Support Library 106
pyAcomms Installation Instructions: 106
pyAcomms Upgrade Instructions 107
Applications 108
Long-term deployment with periodic wakeups and data logging 109
Initial Configuration 109
Wake the modem from hibernate 110
Set the time when the host wakes the modem 110
Command the modem to hibernate (in the future) 110
Retrieving Logged Data 111
Example Real-Time Modem Output 111
Docking an AUV using USBL navigation with precision timing 113
Navigation using a tracking ping 116
CCPST Passthrough Message Examples 118
FATHOMETER Examples 120
Depth sounder, no Iridium link 120
References: 128
Document Revision History: 128

Table 1: NMEA-0183 Talker IDs known by Micro-Modem 6
Table 2: Micro-Modem NMEA Messages – Host to Modem 6
Table 3: Micro-Modem NMEA Messages – Modem to Host 7
Table 4: Quick-Reference Guide for Commonly Used Messages 9
Table 5: Rate Chart 42
Table 6: FDP minipackets 46
Table 7: FDP Miniframe sizes, in QPSK symbols 46
Table 8: FDP data packets 47

Table 9: Communications Packet Receive Statistics Message Fields 56
Table 10: Communications Packet Transmit Statistics Message Fields 58
Table 11: Routine Errors 59

Overview

New Features in the Micromodem-2

The Micromodem-2 offers many improvements to the Micromodem-1 while maintain the same form factor and supporting existing applications. It is acoustically interoperable with Micromodem-1 systems.

Hardware Improvement Highlights

- Configuration settings are stored in non-volatile memory that requires no power. Therefore, clock battery depletion will no longer result in loss of the modem configuration.
- The real-time clock battery is sized to operate for approximately 15 years (at room temperature).
- The supply voltage range has been expanded: 3.5V to 33V.
- The power supply input protection has been significantly improved: absolute max of -36V to +36V.
- The modem has two prioritized power supply inputs.
- The real-time clock is accurate to 2ppm.
- There are up to 14 reconfigurable GPIO pins available.
- GPIO pins can operate when driven from -6V to +6V.
- There are a total of 4 serial ports, including a RS485/422-capable port and a 3.3V serial port.
- The negative operating temperature range has been lowered to -40°C. (See the Temperature section for more details.)
- Acoustic performance is improved.
- The processing capabilities of the modem have been substantially improved.

More Information

For a more detailed discussion of the development of the Micromodem-2, see Reference [3].

Note that several specifications of the Micromodem-2 have changed since this paper was published. Use this manual when designing interoperable systems, and contact the Acoustic Communcations Group with questions.

Upgrading from the Micromodem-1

The Micromodem-2 is designed to be largely backward-compatible with the Micromodem-1, both in hardware and in the software interface.

Mechanically, the Micromodem-2 is the same size as the Micromodem-1 and includes the same mounting hole configuration. See the assembly drawing for more details.

Hardware Differences

Some of the connectors used on the Micromodem-2 are different from those used on the Micromodem-1. An adapter harness, WHOI Acomms Part 231024, may be used to connect a Micromodem-1 wiring harness to a Micromodem-2. For more details on the Micromodem-2 connectors, see the assembly drawing.

With current firmware, idle power consumption of the Micromodem-2 is higher than the Micromodem-1 (approximately 500mW). This is expected to decrease with future software updates.

Software Interface Differences

1. The version number of Micromodem-2 firmware is in a different format: 1.23.12345X. If parsing the version number (typically not necessary in user applications), each section of the version number should be parsed as follows:

<16-bit unsigned integer>.<16-bit unsigned integer>.<32-bit unsigned integer><optional ASCII char>

2. $CCTXA is no longer supported
3. $CCCFR is not yet implemented
4. $CCEXL is no longer supported since its functionality is present in $CCMEC.
5. $CCMEC functionality is slightly different, see CCMEC for current functionality.
6. $CACST supports a version number, see (version 6) in Table 8 of the Software Interface Guide.
7. The following configuration parameters are no longer implemented:
    a. EDR- same as ECD, end of cycle delay
    b. NPT – Noise threshold for navigation, same as PTH
    c. RSP – Receive signal power. Not yet implemented.
8. $CAXST supports a version number, see Table 9 in the Software Interface Guide
9. $CASNR, the input and output SNR fields are now reported to one decimal place as opposed to none.
10. $CAMSE, Mean Squared Error in dB, is reported to one decimal place as opposed to none.

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 $CARXD (hex-encoded 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 typically sent in $CARXD (hex-encoded binary) messages. 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,RXD,1** to enable hex-encoded data output.

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).

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.
$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	80
$CCCFQ	User queries for current value of modem parameter	All	**Error! Bookmark not defined.**
$CCCLK	User sets real-time clock	All	86
$CCCYC	User commands network cycle	All communication	23
$CCMEC	User controls external hardware I/O lines	All	**Error! Bookmark**

Message	Description	Applications	Page
			not defined.
$CCMPC	User commands active navigation cycle (ping)	Active Navigation	24
$CCMSC	User commands modem to sleep	All	37
$CCMUC	User sends User Mini-Packet	Communication	48
$CCPDT	User commands ping to REMUS digital transponders.	Active Navigation	50
$CCTXD	User provides data hex-encoded data to transmit	Communication	17

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	16
$CACFG	Modem reports NVRAM configuration parameter	All	5
$CACLK	Modem reports current time (real-time clock)	All	86
$CACYC	Modem reports current synchronous network command	All communication	23
$CADBG	Modem reports low-level debugging messages	All	62
$CADOP	Modem reports relative Doppler (m/sec)	All Communication	61
$CADQF	Modem reports decision quality factor for last FSK packet	All Communication	60
$CADRQ	Modem requests data from the user.	All Communication	16
$CAERR	Modem reports an error	All	104
$CAMFD	Modem reports comms matched-filter detector value	All Communication	60
$CAMSA	A sleep command was received acoustically	All	38

Message	Description	Applications	Page
$CAMSC	Echo of sleep command	All	37
$CAMSG	Modem reports a system information message such as a time-out when expecting a data reply.	All Communication	58
$CAMSR	An acoustic reply to a sleep command was received	All	38
$CAMUA	Mini-Packet received acoustically	Communication	48
$CAMUC	Echo of User Mini-Packet command	Communication	48
$CAMUR	Reply to Mini-Packet received	Communication	49
$CAREV	Modem reports application name and revision number	All	59
$CARXD	Modem reports received data in binary format	All Communication	17
$CARXP	Modem reports detection of start of packet	All Communication	17
$CATXD	Modem confirms receipt of data in CCTXD to transmit	All Communication	**Error! Bookmark not defined.**
$CATXF	Modem reports end of packet transmission	All Communication	18
$CATXP	Modem reports start of packet transmission	All Communication	18
$SNMFD	Modem reports nav matched-filter detector value(s)	Navigation	53

Table 4: Quick-Reference Guide for Commonly Used Messages

$CACYC, Command Type, SRC, DEST, Packet Type, ACK Request Bit, Number of frames
$CADRQ, Time, SRC, DEST, ACK Request Bit, Max # Bytes Requested, Frame Counter
$CAERR, Time, Error Name, Error Number
$CAMSG, Message Type, Message Number
$CARXD, SRC, DEST, ACK bit, Frame Number, Hex encoded data message
$CATXD, SRC, DEST, ACK bit, # Bytes in CCTXD message
$CCCFG, Parameter Name, Setting
$CCCFQ, Parameter Name
$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
$SNTTA, Travel Time 1, Travel Time 2, Travel Time 3, Travel Time 4, System Time

Hardware Interface

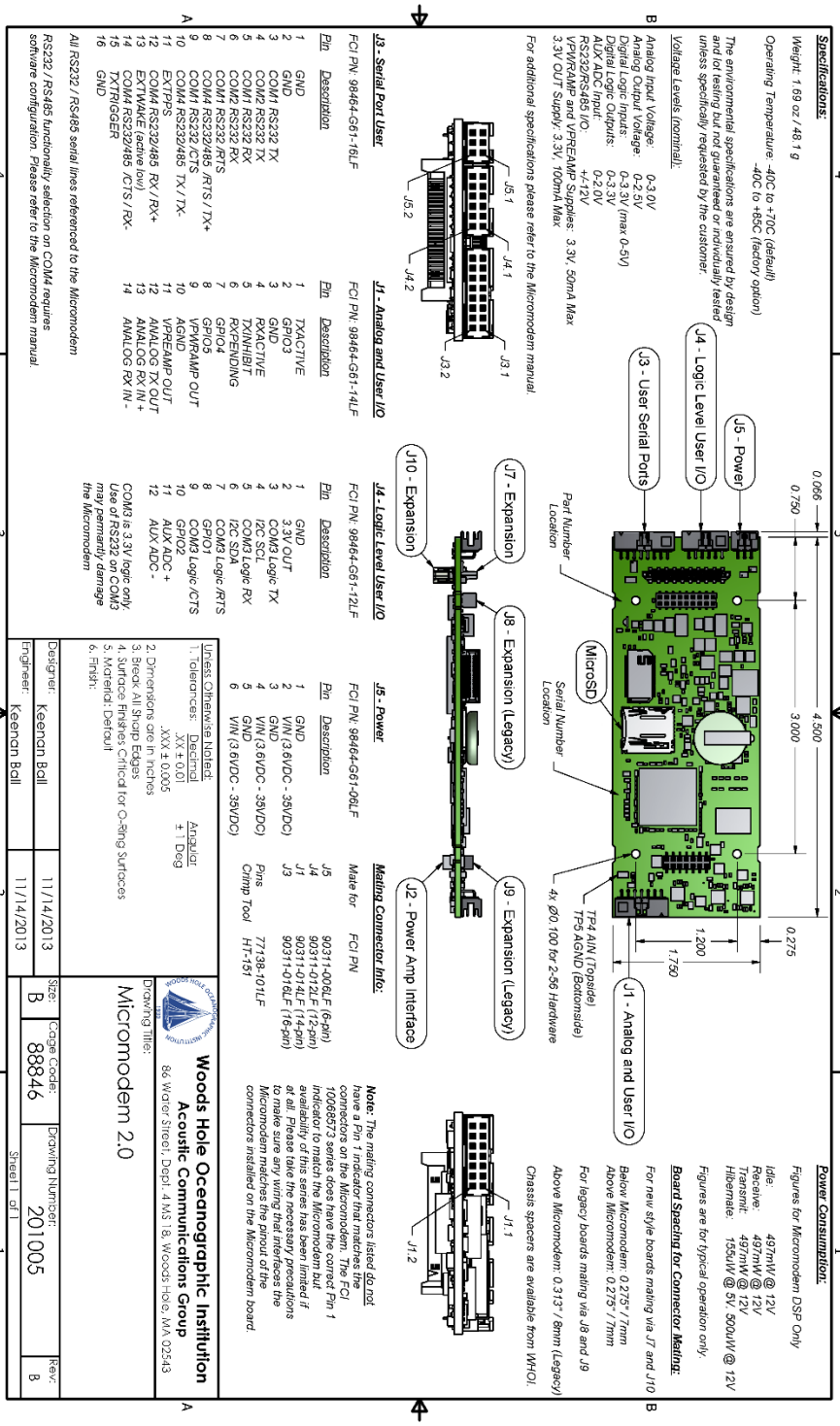
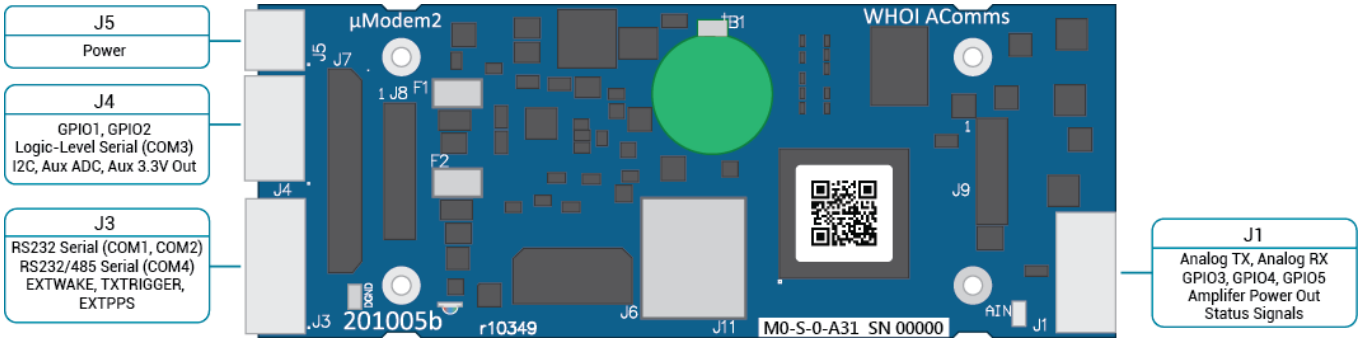


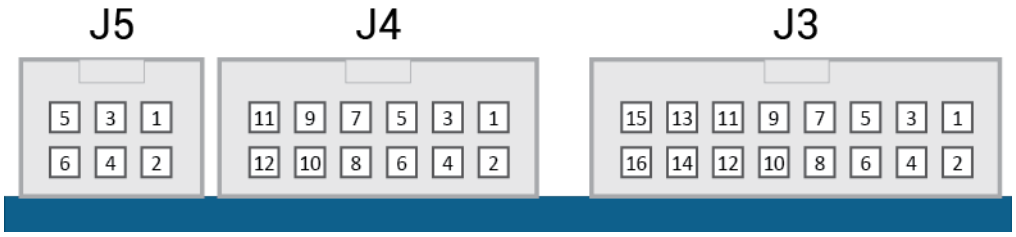
Figure 1: Assembly Drawing of Micromodem-2 showing edge connector pinouts

User Connections



Pinouts

View looking into edge of modem:



J5 Signals

Pin #	Type	Signal Name
1	Power Ground	Ground
2	Power Input	Edge Power
3	Power Ground	Ground
4	Power Input	Edge Power
5	Power Ground	Ground
6	Power Input	Edge Power

J4 Signals

Pin #	Type	Signal Name
1	Ground	Ground
2	Power Out	Aux 3.3V Out
3	3.3V Serial Out	COM3 TX
4	3.3V I2C Out	I2C SDA
5	3.3V Serial In	COM3 RX
6	3.3V I2C Bidirectional	I2C SCL
7	3.3V Serial Out	COM3 #RTS
8	Bidirectional GPIO	GPIO1
9	3.3V Serial In	COM3 #CTS

10	Bidirectional GPIO	GPIO2
11	Analog In (differential positive)	Aux ADC +
12	Analog In (differential negative)	Aux ADC -

J3 Signals

Pin #	Type	Signal Name
1	Signal Ground	Ground
2	Signal Ground	Ground
3	RS232 Out	COM1 TX
4	RS232 Out	COM2 TX
5	RS232 In	COM1 RX
6	RS232 In	COM2 RX
7	RS232 Out	COM1 #RTS
8	RS232 Out / RS485 Out (Input in RS485 half-duplex mode)	COM4 #RTS / TX+
9	RS232 In	COM1 #CTS
10	RS232 Out / RS485 Out (Input in RS485 half-duplex mode)	COM4 TX / TX-
11	Logic In (PPS)	EXTPPS (rising edge is PPS)
12	RS232 In / RS485 In	COM4 RX / RX+
13	Logic In	EXTWAKE (active low)
14	RS232 In / RS485 In	COM4 #CTS / RX-
15	Logic In	TXTRIGGER (rising edge)
16	Signal Ground	Ground

Mating Connectors

The connectors used on the edge of the modem are FCI Minitek 2mm headers. They are locking and polarized to prevent misinsertion. They mate with FCI connector bodies and crimp contacts.

Preassembled Harnesses

WHOI builds an adapter harness for connecting Micromodem-2's to Micromodem-1 wiring harnesses, part **231024**.

Crimp Contacts

Wire Size	FCI Part Number	Digi-Key Part Number	Mouser Part Number
26 AWG to 30 AWG	77138-101LF	609-2653-1-ND	649-77138-101LF
22 AWG to 24 AWG	10044403-101LF		649-10044403-101LF

Connector Bodies

Note that these connector bodies have pin 1 indicators that DO NOT match the connector on the modem. Please be careful to make sure that any wiring matches the pinouts illustrated above. (FCI obsoleted the connector bodies with the "correct" pin 1 indicators.)

Mate for	FCI Part Number	Digi-Key Part Number	Mouser Part Number

J1	90311-014LF	609-2743-ND	649-90311-014LF
J3	90311-016LF	609-2798-ND	649-90311-016LF
J4	90311-012LF	609-2797-ND	649-90311-012LF
J5	90311-006LF	609-2777-ND	649-90311-006LF

Crimpers

In addition to the crimper below, Molex part 11-01-0204 (now obsolete) and Molex 63819-0100 (Digi-Key WM9020-ND) have been successfully used. The Molex crimpers are preferred by several engineers and technicians at WHOI.

FCI Part Number	Digi-Key Part Number	Mouser Part Number
HT-151	609-2508-ND	649-HT-151

Electrical Specifications

Absolute Maximum Ratings

Stresses beyond those listed in this section may cause permanent damage to the modem. **Do not design to these specifications.** Exposure to any absolute maximum rating for extended periods may affect the reliability and lifetime of the modem.

Parameter	Value
Edge Power Input Voltage Stack Power Input Voltage	-36V to +36V
Voltage on GPIO1, GPIO2, GPIO3, GPIO4, GPIO5, EXTPPS, TXTRIGGER, COM3 TX, COM3 RX, COM3 RTS, COM3 CTS, TXACTIVE, RXACTIVE, TXINHIBIT	-8V to +8V
Voltage on EXTWAKE	-3.6V to +12V
Voltage on COM1 RX, COM1 CTS, COM2 RX, COM4 RX/RX+, COM4 CTS/RX-	-25V to +25V
Voltage on COM4 TX/TX-, COM4 RTS/TX+	-13.2V to +9V
Voltage on COM1 TX, COM1 RTS, COM2 TX	-13.2V to +13.2V
Voltage on Aux ADC -, Aux ADC +	-24V to +24V

All voltages specified relative to ground.
These ratings are determined by design, and are not individually tested.

Operating Conditions

Parameter	Conditions	Min	Typical	Max	Units
Input Supply Voltage (Vin)		3.6		33	V
Supply Current	Vin ≤ 5V			1	A
	Vin ≤ 12V			0.5	A
	Vin ≤ 24V			0.25	A
	Vin ≤ 33V			0.15	A
Hibernate Current	Vin = 5V		31	40	µA
	Vin = 12V		41	55	µA
	Vin = 24V		47	65	µA
Voltage on any GPIO pin		-6		6	V

Software Interface

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 2 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.

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,4,6,1,1,4379636c65205465737420467 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.

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.

If ACK is requested on the packet being sent, the ACK field must be set on all TXD messages that supply data for that packet.

Pre-loading a data frame: Both $CCTXD and $CCTXA can be used to pre-load a single 32-byte frame. The next $CCCYC command will not issue a $CCDRQ but send the pre-loaded frame immediately at the rate requested by the cycle init command. Trying to pre-load more than 1 32-byte frame will result in the 2nd frame being dropped.

$CCTXD,SRC,DEST,A,HH…HH*CS

SRC	Source (unit designated as transmitter)
DEST	Destination (unit designated as receiver)
A	ACK request bit set by transmitter, 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.

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

Sending Sweeps and Tones

The Micromodem can send FM sweeps and tones, using the CCSWP command. For example, to send three sweeps, 1 second apart, starting at 22500 Hz, ending at 27500 Hz, which are 50 msec long, use:

$CCSWP,225000,275000,5000,500,3,10000

CCSWP

Send an FM sweep

$CCSWP,start_freqx10, stop_freqx10, bw_Hz, duration_msx10, nreps, reptime_msx10*CS

start_freqx10	Start frequency, in Hzx10.
stop_freqx10	Stop frequency, in Hzx10.
bw_Hz	Bandwidth, Hz, if sweep, must be abs(stop freq-start freq).
duration_msx10	Duration of signal in milliseconds times 10.
nreps	Number of repetitions of the signal
reptime_msx10	Time between successive reps, milliseconds times 10, must be greater than the length of the signal. Ignored if nreps = 0.

Sending M-Sequences

The Micromodem can send maximal-length sequences, using the CCMSQ command. The 'cycles per chip' field in the command should be chosen so that the implied bandwidth (carrier_Hz/cycles_per_chip) is an allowable modem bandwidth. For example, to send 3 repetitions of a length 511 (k=9) M-sequence at a carrier of 700 Hz at 14 cycles per chip , use:

$CCMSQ,511,3,14,700

CCMSQ

Send a maximal-length sequence

$CCMSQ,seqlen_bits,nreps,cycles_per_chip,carrier_Hz*CS

seqlen_bits	Sequence length, bits. Should be 2^k-1, where k={4…16}
nreps	number of repetitions, between 3 to 60.
cycles_per_chip	carrier cycles per chip
carrier_Hz	carrier, Hz

Passthrough Messages

A "passthrough" sentence, $CCPST, allows serial strings to be sent to devices connected to other serial ports on the Micromodem-2. For example, a host computer on Micromodem-2 COM2 could send a configuration sentence to a GPS connected to Micromodem-2 COM4. The passthrough sentence is

configurable to allow a variety of options. The passthrough sentence can include the standard NMEA-style response, $CAPST…, or it can just include the user's string without a $CAPST… preamble, and with optional reformatting to generate a new NMEA sentence. See documentation and examples below.

CCPST

Pass a string from one Micromodem-2 serial port to other Micromodem-2 serial port(s), with optional reformatting.

$CCPST,target_ports,strip_msg,prepend_dollar,append_cksum,append_CRLF,supplied_cksum,string*CS

target_ports	0-15, bit mask: bits 0,1,2,3 correspond to passing the message through to COM1,2,3,4, respectively. target_ports does not necessarily have to include the originating port that sent the $CCPST command, in which case no message is sent back to the originating port.
strip_msg	0-1: if 0, pass through complete $CAPST string. If 1, strip **$CCPST,t,s,p,a,a,** from message prior to passing it through.
prepend_dollar	0-1: If strip_msg=1 and prepend_dollar=1, then the passthrough message will be **$string\r\n** with the **$CAPST,t,s,p,a,a,** stripped off and a dollar sign pre-pended. Otherwise, if strip_msg=0, then prepend_dollar must be 0.
append_cksum	0-1: If strip_msg=1 and prepend_dollar=1 and append_cksum=1, then the passthrough message will be **$string*CS\r\n** with the **$CAPST,t,s,p,a,a,** stripped off, a dollar sign pre-pended, and a checksum appended. Otherwise, if strip_msg=0, or if (strip_msg=1 and prepend_dollar=0) then append_cksum must be 0.
append_CRLF	**Set to 0 only.** [Future Implementation: 0-1: If strip_msg=1 and prepend_dollar=0, then if append_CRLF=1, pass through **string\r\n** . Otherwise, append_CRLF=0 and the passthrough string is not further modified.]
supplied_cksum	Empty, or two ASCII-hex chars: if supplied_cksum is not the empty string, and append_cksum=1 (and hence strip_msg=1 and prepend_dollar=1), then pass through **$string*CS\r\n** where CS is the supplied checksum. Otherwise, do not modify passthrough string.
string	The string to pass through to the target ports. Maximum length is 125 characters.

***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The modem will pass through the response to the CCPST command to the target ports as described below with the CAPST sentence.

CAPST

Response to CCPST passthrough command. Response is sent to each of the target_ports, not necessarily including the originating port that sent the $CCPST command unless that port is explicitly included in target_ports.

If strip_msg=0, response string includes $CAPST and the arguments:

$CAPST,target_ports,strip_msg,prpnd_dollar,appnd_cksm,supplied_cksm,string*CS

where fields are as described above in CCPST documention.
(For 2.0.18841 and earlier, append_CRLF is missing from $CAPST message.)

If strip_msg=1, then response string does NOT include $CAPST or the arguments, and only includes the "**string**" argument itself, with optional NMEA-style reformatting:

If prepend_dollar=1 and append_cksum=0:
$string\r\n
If prepend_dollar=1 and append_cksum=1 and supplied_cksum is empty:
$string*CS\r\n [CS is calculated checksum]
If prepend_dollar=1 and append_cksum=1 and supplied_cksum not empty:
$string*CS\r\n [CS is supplied checksum]
If prepend_dollar=0 and append_CRLF=0:
string
If prepend_dollar=0 and append_CRLF=1:
[FUTURE IMPLEMENTATION] **string\r\n**

CCPST Passthrough Message Examples can be found in the Applications section.

NMEA API Level

The NMEA interface provides a version number, called the "NMEA API Level" below, which only changes when the NMEA interface changes. Therefore, it doesn't change with each firmware update, but only when NMEA messages are changed or added. This version number is a single integer that monotonically increases.

The host can use the NMEA API Level when developing systems that interface with the Micromodem and can avoid parsing the firmware version number. The host software can query the NMEA API version using the CCALQ command.

CCALQ

Request NMEA API Level and other information useful to application developers

$CCALQ,0*CS

0	Reserved. Must be **0**.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The modem will respond to this command with the **CAALQ** message.

CAALQ

Response to NMEA API Level query

$CAALQ,app_name,nmea_api_level,reserved*CS

app_name	Application currently running on the modem. **MODEM** for a uModem2 running standard firmware **RECOVERY** when the modem is in recovery mode Other values may appear when using custom, application-specific firmware
nmea_api_level	NMEA API level (integer)
reserved	Reserved field, currently blank.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

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.

$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, 0-6.
ACK	Deprecated. Use either 0 or 1 in this field. To request acknowledgment mini-packets, set the ACK bit in the $CCTXD message.
Nframes	Number of frames to send in packet , only used in serial commands for downlink requests
***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 MOD 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 for bands 1, 2 and 3.

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,TIMEOUT*CS

RXSIG	Deprecated. Set to 0.
TXSIG	Signal type. 1 to transmit FM sweep for a FSK packet, 2 to transmit FM sweep for a PSK packet, both in current band.
TIMEOUT	Deprecated. Set to 0.
***CS**	Hex coded checksum (8 bit XOR of sentence)

CARSP – Response 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

Recording Raw Data

Using the $CCREC command, the modem can record up to 2.8Mbytes (1.4M words) of basebanded data. In addition to the command, the rec configuration parameter shows the current state of the recording.

CCREC

Start recording raw passband or baseband data

$CCREC,start_time,rec_secs,chmask,passbandflag,mem_loc,print_format,bw_Hz, reserved,serial_port*CS

start_time	**YYYY-MM-ddTHH:mm:ssZ:** time in ISO 8601 Extended format (including the "Z" time zone specifier), it corresponds to the time when the modem will start recording. This time must be in the future, and can be at most 5 hours from the current time. **2**-**18000**: the modem will start recording after these many seconds have elapsed, with the upper limit corresponding to 5 hours. **-1**: Stop any current recording. All other arguments are ignored
rec_secs	**0** Number of seconds to record. This cannot result in more than 1.4M words of data being requested.
chmask	**1-30** A bitmask of 5 bits, the least significant being the onboard channel, and the other four being the multichannel card. Either the onboard channel or the multichannel source can be selected for recording, but not both.
passbandflag	**0-1** baseband vs. passband recording. Currently works only for baseband
mem_loc	Location where data is stored **0** External SDRAM, default **1** SD card, currently not operational **2** Flash, currently not operational
print_format	Format of print messages **0** ASCII, not operational **1** Base 64, default **2** Base 252, not operational
bw_Hz	**10-10000** Bandwidth of basebanded data, can be different from modem's current detector settings for a limited set of values.

ain_shift	**0-12** Analog input left shift, used for data that may not be full range for a 16-bit A2D.
serial_port	**0-15** A bitmask of output ports to print recorded data to, with 0 used for no printing.
***CS**	Optional: The NMEA sentence checksum, hex coded (8- bit XOR or CRC-32). See Message Checksums.

The modem will respond to this command with a **CAREC** message with the start time field populated by the actual start time in ISO 8601 Extended format, unless the command was sent to stop recording.

CARCI

Informational message issued at the start and stop of a recording

$CARCI,START*CS

$CARCI,STOP*CS

CAREP

Print out recorded data at the end of the recording interval, if the serial port field in the CCREC command is non-zero. The data is chunked into messages, each of which is further composed of buffers with unique shift and length. Each message is represented by a CAREP.

$CAREP,msgnum,nmsgs,nbufs,start_time,time_source,nchan,iscomplex,sample_rate, buflength,shift,data*CS

msgnum	Message number, starting from 1 up to a maximum of nmsgs.
nmsgs	Total number of messages in this recording
nbufs_per_msg	The number of buffers in this message
start_time	Time of the first sample in this message, in Unix format with a subsecond field added.
time_source	5-bit field, with bits 1-3 defining the PPS source, and bits 4-5 defining the real time clock source. See CCTMQ for further information.
nchan	Number of channels being recorded, set by bitmask
iscomplex	Set to 1, if basebanded data
sample_rate	The sample rate of the recording, in Hz
buflengths	Buffer length, in 16-bit int, for each buffer in this message, printed as base 64
shift	The shift value of each buffer, printed as base 64

data	Total data in the buffers, printed as base 64. If multichannel baseband data, then stored as r,i for channel 1, r,i for channel 2 and so on for each buffer

Example

Set the modem to record 2 seconds after the receipt of the command for 10 seconds on all four channels of the multichannel card at a baseband bandwidth of 100 Hz.

$CCREC,2,10,30,0,0,0,100,0,1

$CAREC,2014-04-12T08:10:51Z,2014-04-12T08:11:01Z,30,0,0,1,0,0,0*7C

$CARCI,START*36

$CAREV,081054,AUV,2.0.20147*17

$CAREV,081054,COPROC,0.30.0.56*4B

$CARCI,STOP*6E

$CAREV,081104,AUV,2.0.20147*13

$CAREV,081104,COPROC,0.30.0.56*4F

$CAREC,2014-04-12T08:11:08Z,2014-04-12T08:11:18Z,30,0,0,1,0,0,1*78

$CARCI,START*36

$CAREV,081114,AUV,2.0.20147*12

$CAREV,081114,COPROC,0.30.0.56*4E

$CARCI,STOP*6E

$CAREP,1,11,11,1397290267.899122,9,4,1,200,oACgAKAAoACgAKAAoACgAKAAoACgAK==,CwACAAMABAAFAAYABwAHAAgABwAIAA==,AAAAAAAA///9/w4A/f+W/3oAUwCj/wYAXwDy/4//EgCYAPL/jv9MAGQAjP/G/1wASwC7/+H/bgANAJD/8P9iAL//Zv9HAFMAov+s/1sAjQCj/wYAXwDy/4//EgCYAPL/jv9MAGQAjP/G/1wASwC7/+H/bgANAJD/8P9iAL//Zv9HAFMAov+s/1sAjQAAAAAA+P8CACQA7f+Q/6/74P8ZAGQAjP/G/1wASwC7/+H/bgANAJD/8P9iAL//Zv9HAFMAov+s/1sAjQAAAAAA+P8CACQA7f+Q/6/74P8ZAAYA+f8BAAaAOv///89ABYAKwAvAOT/mf9b//T/Mv8WAbv/nwCCAL3/kC/rgCeAN3/JAPT/xwA4/4gAKP/J/4n/I/+HAD3/wABfAO3/rgCC/24Ab/+S/+L/Mv+WAIz/7wA6AP

//8BAAEAwAA7AD4A1wBt/0AA6//b/6j/fwBg/6L/zP81/5cAtv+rADUACQC4AF7/nwBE/9X/9P9r/4EAmf+dAIYA7//LADf/bgBm/7D/KABK/9IAoP9g/6L/zP81/5cAtv+rADUACQC4AF7/nwBE/9X/9P9r/4EAmf+dAIYA7//LADf/bgBm/7D/KABK/9IAoP//////AQD//wEAAQD//wEA/////17/nwBE/9X/9P9r/4EAmf+dAIYA7//LADf/bgBm/7D/KABK/9IAoP//////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////AQABAP//AQD/////AQD//wEAAQCMAFkADQDFAHf/dgDx/8r/tP90ALH/sf/r/zn/sABj/6sATAANAN8AZP9wAET/rP/u/1f/hQBr/6cAZAAOALgARv9gAEX/x/8AAEL/lQBZ/7H/sf/r/zn/sABj/6sATAANAN8AZP9wAET/rP/u/1f/hQBr/6cAZAAOALgARv9gAEX/x/8AAEL/lQBZ/////8BAP//AQABAP//AQD/////ZP9wAET/rP/u/1f/hQBr/6cAZAAOALgARv9gAEX/x/8AAEL/lQBZ/////8BAP//AQABAP//AQD/////AQD//wEAAQD//wEA/////wEA//8OALgARv9gAEX/x/8AAEL/lQBZ/////8BAP//AQABAP//AQD/////AQD//wEAAQD//wEA/////wEA//8BAAEAsgCTADMA5QCO/ycAvv++/2z/VAAR/1L/EgAh/7kAhP/CAEEAFgDPAEX/pQBB/6z/sv9C/9AAd/+0AEAA+f+xADT/MwBV/4T/0v8p/58Alv8R/1L/EgAh/7kAhP/CAEEAFgDPAEX/pQBB/6z/sv9C/9AAd/+0AEAA+f+xADT/MwBV/4T/0v8p/58Alv////AQD//wEAAQD//wEA////0X/pQBB/6z/sv9C/9AAd/+0AEAA+f+xADT/MwBV/4T/0v8p/58Alv/////AQD//wEAAQD//wEA////wEA//8BAAEA//8BAP////8BAP//AQABAP//AQD/////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////8BAP//AQABAP//AQD/////AQD//wEAAQCjAEQA8f/XAGv/NADE/2X/mQCY/78AOwAvANoAav+TAEf/uv+R/yv/jQBL/+YABABFAIsAd/+HAGH/8v+n/zn/n/aABE/9EAOQBfALwAhf+fAL8AOwAvANoAav+TAEf/uv+R/yv/jQBL/+YABABFAIsAd/+HAGH/8v+n/zn/n/aABE/9EAOQBfALwAhf+fAP////8BAP//AQABAP//AQD/////jQBL/+YABABFAIsAd/+HAGH/8v+n/zn/n/aABE/9EAOQBfALwAhf+fAP////8BAP//AQABAP//AQD/////AQD//wEAAQD//wEA/////wEA//+n/zn/n/aABE/9EAOQBfALwAhf+fAP////8BAP//AQABAP//AQD/////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////8BAP//AQABAD3/5/96AAd/+0AEAA+f+xADT/MwBV/4T/0v8p/58Alv/////AQD//wEAAQD//wEA/////wEA//8BAAEA//8BAP////8BAP//AQABAP//AQD

CACFT – Fathometer ping reply report

Report a depth from a fathometer.active,1 ping.

$CACFT,start_time,lat_deg,lon_deg,AGN,RTTT_sec,peak_val,peak_shift_val_squared*CS

start_time	**YYYY-MM-ddTHH:mm:ssZ:** time in ISO 8601 Extended format (including the "Z" time zone specifier), it corresponds to the time when the modem transmitted the outgoing ping
lat_deg.	Latitude in degrees, if available, else set to -100.0
lon_deg.	Longitude in dgrees, if available, else set to -200.0
AGN	Gain setting
RTTT_sec	Round trip travel time, seconds
peak_value	Peak value of captured IRE buffer
peak_shift_val_squared	shift compensation required for fixed-point peak value
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

CACIR – Fathometer ping reply IRE capture

Report impulse response from a fathometer.active,1 ping. There will be 'nmsgs' reported, each message will report 'nbufs', each of length 'buflen'.

$CACIR,msgnum,nmsgs,capture_start_time,nbufs,buflen,shift_val1,buf1_val,….shift_val*n*,buf*n*_vals*CS

msgnum	message number
nmsgs	Total messages for this report
start_time	**YYYY-MM-ddTHH:mm:ssZ:** time in ISO 8601 Extended format (including the "Z" time zone specifier), it corresponds to the start time of the first capture buffer
nbufs	Total number of buffers in this message
buflen	length of each buffer, in samples
shift_val1	shift compensation required for this buffer's samples
buf1_val	Impulse response squared, in base64 or hex based on configuration parameter.
shift_val*n*,buf*n*_vals	shift values and IRE squared values for nbufs
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Sleep and Hibernate Sentences

Hibernating

The modem can enter a low-power state during which it will not respond to acoustic or serial traffic.

The user can command the modem to enter hibernate either via a command, **CCHIB**, or via configuration settings that enable automatic hibernation on a schedule.

The modem can wake from hibernate via the following:

Power-on reset (power to the modem is removed and reinstated)

Signal on EXTWAKE pin (the polarity of this signal can be changed via a hardware option, but for most users the modem will wake when this signal is pulled low/to ground)

Signal on STACKWAKE pins – For the future expansion with highly integrated hardware. Please consult the Acoustic Communications Group before using this functionality.

Real-time clock alarm – This can be set via user command and configuration parameters as described in this section.

Hibernate Configuration Parameters

All parameters related to hibernating are controlled using parameters in the **hibernate** configuration group.

hibernate.wake_interval Number of seconds between scheduled automatic wakeups. This is typically used only if the modem must enter hibernate and wake multiple times without host interaction.

0 (Default)	Disable scheduled automatic wakeups
2 – 2419200	Number of seconds between automatic wakeup times. Maximum value corresponds to 28 days between wakeups.

hibernate.wake_reference Reference time used as an offset for scheduled automatic wakeups. Wakeup times are offset from this reference time by (signed) integer multiples of the wake interval.

YYYY-MM-ddTHH:mm:ssZ	Reference time in ISO 8601 Extended format. The time zone specifier must be included and set to “Z”. Default is 1970-01-01T00:00:00Z

hibernate.hibernate_after Number of seconds modem remains on after boot before automatically hibernating. This is typically used only if the modem must enter hibernate and wake multiple times without host interaction.

0 (Default)	Disable automatic hibernation
5 – 2419200	Number of seconds to remain on after modem boot. Maximum value corresponds to 28 days before automatically entering hibernate.

CCHIB

Command modem to hibernate

$CCHIB,hibernate_time,wake_time*CS

hibernate_time	0: hibernate immediately. The modem will hibernate after sending the response to this command. YYYY-MM-ddTHH:mm:ssZ: time in ISO 8601 Extended format (including the "Z" time zone specifier), it corresponds to the time when the modem will hibernate. This time must be in the future, and can be at most 28 days from the current time. If the hibernate.hibernate_after configuration parameter is set, the modem will pick the earlier of the two times to hibernate. 2-2419200: the modem will enter hibernate after these many seconds have elapsed, with the upper limit corresponding to 28 days. If the hibernate.hibernate_after configuration parameter is set, the modem will pick the earlier of the two times to hibernate. -1: Clear any hibernate time previously commanded by $CCHIB. Does not clear scheduled hibernate times from the hibernate.hibernate_after configuration parameter.
wake_time	0: Wake up on external trigger on the EXTWAKE or STACKWAKE pins, or a power reset. YYYY-MM-ddTHH:mm:ssZ: time in ISO 8601 Extended format (including the "Z" time zone specifier), it corresponds to the time when the modem will wake from hibernate. This time can be at most 28 days from the current time and must be at least 2 seconds after the hibernate_time, if specified. 2-2419200: the modem will wake from hibernate after these many seconds have elapsed, with the upper limit corresponding to 28 days.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

After the CCHIB command is sent, the modem will reply with a CAHIB message that indicates when the modem will hibernate and when it will wake.

CAHIB

Response to hibernate command

$CAHIB,hibernate_cause,hibernate_time,wake_cause,wake_time*CS

hibernate_cause	0 if the modem will be hibernating at hibernate_time due to a value set by the user in the CCHIB command 1 if the modem will be hibernating at hibernate_time due to the value of the hibernate.hibernate_after configuration parameter 2 if the modem will be hibernating at hibernate_time due to a local serial $CCMSC command. 3 if the modem will be hibernating at hibernate_time due to an acoustically received $CCMSC command. -1 if no hibernate time is set.
hibernate_time	Date and time at which the modem will hibernate, in ISO 8601 Extended format. If the user specified 0 (hibernate immediately) as the hibernate mode in the CCHIB command, this will be the current time. Field is blank if no hibernate time is set.
wake_cause	0 if the modem will wake at wake_time due to a value set by the user in the CCHIB command 1 if the modem will wake at wake_time due to the value of the hibernate.wake_interval and hibnerate.wake_reference configuration parameters 2 if the modem will wake at wake_time due to a local serial $CCMSC command. 3 if the modem will wake at wake_time due to an acoustically received $CCMSC command. -1 if no wake time is set.
wake_time	Date and time at which the modem will wake from hibernate, in ISO 8601 Extended format. If the modem will hibernate until an external signal occurs, this field is blank.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent in response to a **CCHIB** command, and also when hibernate or wake times change, for example due to changing the hibernate configuration parameters or receiving a $CCMSC sleep command.

CAHBR

Informational message issued before the modem goes into hibernation

$CAHBR,wake_time*CS

wake_time	Date and time at which the modem will wake from hibernate, in ISO 8601 Extended format. If the modem will hibernate until an external signal occurs, this field is blank.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent just before the modem goes into hibernate. Thus, the CAHIB sentence serves as an immediate acknowledgement of a change in hibernate or wake times, while the CAHBR sentence is an informational message about the modem's change of status before it shuts down.

pyAcomms Support

Hibernate is provided in pyAcomms via the start_hibernate method of the Micromodem class (acomms.Micromodem.start_hibernate). For more information, see the pyAcomms documentation.

Hibernate Examples

In examples, **messages in bold** are commands sent from the host to the modem and messages not in bold are messages sent from the modem to the host. *CS* is the NMEA message checksum.

Start hibernating immediately and wake only if an external signal occurs

This example assumes that the hibernate configuration parameters are set to their default values, and that the current time is 2012-12-01T01:04:05Z.

$CCHIB,0,0

$CAHIB,0,2012-12-01T01:04:05Z,0,**CS*

$CAHBR,**CS*

The modem will hibernate immediately. However, other NMEA messages may be printed before the modem hibernates and before the CAHBR is issued.

Start hibernating in 15 seconds and wake at 2012-12-01T06:00:00Z

This example assumes that the hibernate configuration parameters are set to their default values, and that the current time is 2012-12-01T01:05:00Z.

$CCHIB,15,2012-12-01T06:00:00Z

$CAHIB,0,2012-12-01T01:05:15Z,0,2012-12-01T06:00:00Z**CS*

After 15 seconds have passed and the modem is about to hibernate, it will issue the following message:

$CAHBR,2012-12-01T06:00:00Z**CS*

CCRST

Reset command. The modem reboots from given location in flash memory. Power is not lost during a reset. The syntax below is for firmware versions newer than 2.0.14791:

$CCRST,*slotCS**

slot	0: Reset and reboot into recovery firmware. 1,2: Reset and reboot into user-programmable firmware slot 1 or 2, respectively.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The reset command can be used to set the modem into a known boot state. It can also be used while programming new firmware onto flash. In this case, the user will issue $CCRST,0 to reboot into the recovery slot. After programming new firmware onto flash, $CCRST,1 or $CCRST,2 will reboot the modem into the firmware in slot 1 or 2, respectively.

Subsequent reboots and power-cycles of the modem will boot into the last slot successfully booted with the CCRST command.

See Firmware Update section for more details on updating firmware.

(Deprecated syntax: For firmware versions older than 2.0.14774, $CCRST,1 will reboot into the recovery firmware, and $CCRST,0 will reboot into the most recently programmed user firmware slot.)

Sleep

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. ***(Users should only use the Sleep commands when the command must be sent remotely; otherwise the Hibernate commands are recommended.)***

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 via EXTWAKE. 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

Use this command to set and read the hardware I/O lines on a local or remote modem.

$CCMEC,source,dest,line,mode,arg*CS

source	Address of unit issuing the command			
dest	ID of unit whose lines are being toggled. This can be the same as source if controlling the local modem.			
line	**Arg**	**MM-1**	**MM-2**	**Pin MM-2**
	1	EXTSEL1	Not implemented	None
	2	EXTSEL2	GPIO4	J1.7
	3	GPIO	GPIO3	J1.2
	4	reserved	GPIO5	J1.8
	5	reserved	GPIO1	J4.8

	6	reserved	GPIO2	J4.10
	7	COM1	Not implemented	None
	8	COM2	Not implemented	None

mode	**0** Read Line **1** Set Value **2** Toggle high **3** Toggle low
arg	Depending on mode, this has different values: If Mode is 0, this must be **0**. If Mode is 1, then this is: • **0** for Off • **1** for On. If Mode is 2 or 3, this is an integer argument for toggle time duration: • **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**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The source is the originator of the command and the DEST is the unit whose hardware lines are to be controlled. If Source 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 four modes are available, read, set, toggle high and toggle low. The default state of the output lines is 0 volts for lines upon power up. If the modem does a hard reboot the lines are also cleared to the default state.

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

(See description of SRC, DEST, LINE, MODE, ARG for $CCMEC above.)

Packets, Rates, 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 below.

Table 5: Rate Chart

Rate	ECC,Modulation Type	Bytes Per Frame	Maximum Frames	Maximum bytes per packet	Packet Payload (bps) at 5000 Hz bandwidth
0	Conv(2,1,9),FH-FSK	32	1	32	80
1	**BCH (128:8),QPSK**	**64**	**3**	**192**	**498**
2	DSSS-15,QPSK	64	3	192	520
3	DSS-7,QPSK	256	2	512	1223
4	**BCH(64:10), QPSK**	**256**	**2**	**512**	**1301**
5	**Hamming(14:9),QPSK**	**256**	**8**	**2048**	**5388**
6	DSS-15,QPSK	32	6	192	490

Rates number 1, 4, and 5 use block codes, which perform better than the DSSS spreading/convolutional codes (rates 2, 3, and 6) for the typical case of single-user access to the channel. Unless you have a specific backward-compatibility requirement, we recommend that you use rates 1, 4, and 5. Rate 6 is provided to allow PSK support for legacy 32-byte frames.

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 specifying the data rate.
2. If using data rate 0 (FH-FSK), the transmitting modem then acoustically transmits the short Cycle-Init command packet. All other modems that receive the Cycle-Init command packet will report it to their hosts using the CACYC message. (If using PSK rates, the transmitting modem does not transmit an explicit Cycle-Init command packet.)
3. The transmitting modem then queries its host with the CADRQ message for the data to be transmitted and then the modem acoustically transmits the data packet.

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 PSK data packets, the Cycle-Init fields are incorporated into the data packet header, thus obviating the need to send an explicit Cycle-Init acoustic packet. An NMEA $CACYC message is still reported, however.

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 $CCTXD data response message.
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.

Flexible Data Protocol (FDP)

The Micromodem-2, in addition to providing the standard Micromodem-1 data and mini packets, also implements a new communications protocol, the Flexible Data Protocol. This protocol is fully backwards compatible with the Micromodem I standard, but allows greater flexibility and ease of use.

CCTDP- transmit (downlink) data packet

Send up to 'maximum' number of bytes at the user defined rate. If the data bytes specified are 100 or less, and the rate chosen is compatible with a minipacket rate, a FDP minipacket is sent, else a full FDP data packet is sent.

$CCTDP,dest,rate,ack,base64,hexdata*CS

dest	ID of the destination
rate	The data rate corresponding to bits per second
ack	Currently set to 0. In future, 1 if an acknowledgement is requested
base64	1 for base64 encoded data, 0 for hex encoded data
data	The data to be sent, in hexadecimal or base64 format, not to exceed the maximum for the rate specified (see rate chart). Future feature: if this field is empty, the modem will first look at the data queue, if that is empty, the modem will issue a data request (CAFDQ).
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The modem will respond to this command with the **CATDP** message,followed by **CATXP** and **CATXF**, if they are set. The modem breaks up the data into frames, each protected by a CRC. The response to the **$CCTDP** command reports the number of bytes in each mini frame of the packet along with the number of bytes in each data frame. Note the nbytes fields within mini or data frames are separated by a semicolon delimiter.

CATDP - Response to CCTDP command

$CATDP,errflag,uniqueID,dest,rate,ack,base64,nbytesmf1;nbytesmf2;…,nbytesdf1;nbytesdf2;…,checksum*CS

errflag	Error flag, 0 no error, packet is queued; 1 error, packet is dropped
uniqueID	Packet ID for transmit queue (future implemetation,currently set to 0)
dest	ID of the destination

rate	The data rate corresponding to bits per second
ack	1 if an acknowledgement is requested
base64	1 for base64 encoded data, 0 for hex encoded data
nbytesmf1;nbytesmf2…	Number of bytes of data in each mini frame up to 8 mini frames, blank if there are no miniframes.
nbytesdf1;nbytesdf2…	Number of bytes of data in each data frame up to 16 data frames, blank if there are no data frames.
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

A modem, on receiving a FDP data packet, will print out a CARDP message along with the standard set of detection and packet statistics messages followed by the appropriate response to the packet received. The RDP message takes the place of the CACYC and the CARXD messages. The data bytes are printed in hexadecimal or base64 notation based on the value of the recv.base64 config parameter.

CARDP- Reception of a FDP downlink data packet.

$CARDP,src,dest,rate,ack,reserved,crccheck1;nbytes1;mf1;…crccheckn;nbtyesn;mfn,crccheck1;nbytesn;df1;…crccheckn;nbytesn;dfn*CS

src	ID of the originating modem
dest	ID of the destination modem
rate	The data rate corresponding to bits per second
ack	1 if an acknowledgement is requested
reserved	Reserved
crccheck1; **nbytes1;** **mf1;…**	CRC check on mini frame, 1 if passed, 0 if failed, Number of bytes in mini frame Mini frame bytes in hexadecimal or base 64 notation, if CRC check passed, blank if not.
crccheck1; **nbytes1;** **df1;…**	CRC check on data frame, 1 if passed, 0 if failed, Number of bytes in data frame Data frame bytes in hexadecimal or base64 notation, if CRC check passed, blank if not.
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

In addition to the **crccheck** field on the CARDP message which alerts the user to an erroneous frame, there are two other error messages possible. One, if the modulation header does not decode, causing the modem to print 'Bad Modulation Header', and two, if the data header does not decode, the modem will print "BAD_CRC, Data Header". See Table 11.

FDP Minipacket rates

A minipacket can be sent using different coding rates and data sizes. The following tables show a breakdown of the packet size and duration for varying rates with a bandwidth of 5000Hz. Although only full frames are listed, there is no penalty for sending partial frames. The packet duration does not include the FM probe and null time that precede each packet. Grayed out rates are not yet implemented. The duration of a packet can be calculated using the following formula:

d = (framesyms x nframes + overheadsyms)/BW
where:

d = Packet duration (sec)
framesyms = total number of symbols in a frame
nframes = number of frames
overheadsyms = 702, packet overhead, symbols
BW = symbol rate bandwidth, Hz

Table 6: FDP minipackets

Rate	Number of frames	Frame size (symbols)	Data bytes in packet	Packet length (QPSK symbols)	Packet duration (milliseconds)	Burst rate (bps)
1	1-8	896	1-100	7870	1574	508
3	1-8	384	1-100	3774	755	1060
5	1-8	91	1-100	1430	286	2797

The first miniframe of a packet can hold 9 data bytes, subsequent frames can hold 13. Thus the breakdown of packet size with respect to number of frames sent is:

Table 7: FDP Miniframe sizes, in QPSK symbols

Nframes	Databytes	Packet length (rate1)	Packet Length(rate 3)	Packet length(rate 5)
1	1-9	1598	1086	793
2	10-22	2494	1470	884
3	23-35	3390	1854	975
4	36-48	4286	2238	1066
5	49-61	5182	2622	1157
6	62-74	6078	3006	1248
7	75-87	6974	3390	1339
8	88-100	7870	3774	1430

A data packet can be sent using different coding rates and data sizes. The following tables show a breakdown of the packet size and duration for varying rates with a bandwidth of 5000Hz. Only full

frames are sent, partial frames are padded to full length. The packet duration does not include the FM probe and null time that precede each packet.

Table 8: FDP data packets

Rate	Number of frames	Frame size (bytes)	Data bytes in packet	Packet length (QPSK symbols)	Packet duration (milliseconds)	Burst rate (bps)
1	1-3	64	192	15358	3072	500
5	1-8	256	2048	15031	3000	5460

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

CCPGT – Ping Generic transponder, host to modem

Ping Generic Transponder from host to modem.

$CCPGT,Ftx,nbits,tx_seq_code,transponder_timeout_ms,Frx,rx_seq_code1, rx_seq_code2,rx_seq_code3,rx_seq_code4, Bandwidth_Hz,reserved2*CS

Ftx	Transmit frequency for outgoing ping, Hz
nbits	number of bits to use from the signal sequence, same for outgoing as well as incoming pings
tx_seq_code	outgoing sequence bits, packed into a 64 bit int
transponder_timeout_ms	Maximum time to listen for in milliseconds
Frx	Receive frequency for incoming ping, Hz
rx_seq_code1,2,3,4	which signals to receive, 0 being no signal
Bandwidth_Hz	chiprate/bandwidth, of outgoing and receive signals
reserved2	future use, potentially modulation scheme
***CS**	Hex coded checksum (8 bit XOR of sentence) optional.

Example 1: Sending $CCPGT with a Turyn code (1 1 0 1 1 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 1 1 0 0 0) at a chiprate of 4 kHz and carrier frequency of 26 kHz and listening for a reply of the Turyn code at 24 kHz:

$CCPGT,26000,28,DA444780,1000,24000, DA444780,0,0,0,4000,0*68

$SNPGT,26000,28,DA444780,1000,24000,FDA44478,0,0,0,4000,0*09
$CATXP,280*48
$CATXF,280*5E
$SNTTA,,,,,144456.86*56

CCPDT – Ping REMUS digital transponder, host to modem

REMUS active LBL navigation ping message from host to modem. Dprecated, use $CCPGT instead

$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. If set to 1, transmits on rising edge of EXTPPS hardware input. If set to 0, transmits immediately and does not wait for EXTPPS.
STO	Synch time-out in milliseconds. Set to 0 if not used.
Timeout	Maximum time to listen acoustically (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 turnaround 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. In passive mode, the navigation statistics message $SNNST must be turned on to get the detection on the outgoing ping as well as all four replies. Additionally, the passive nav travel times are reported as direct times with no compensation for Transponder Turnaround Time(TAT).

$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), difference between Maximum and minimum frequency should not exceed 5kHz.	
Tflag	Transmit Flag	0 – wait for external sync (deprecated) 1 – transmit outgoing ping 2 – passive mode, listen for outgoing ping and replies
***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.

In passive mode, the 'Time of ping' field is populated with detection time of the outgoing ping. This is used as the reference time to calculate the travel time offsets for transponders A, B, C and D.

$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

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. Versions 6+ of the CST message are supported in the Micromodem-II.

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

Table 9: Communications Packet Receive Statistics Message Fields

Field number (version 0)	Field number (version 6, micromodem-2)	Field Name	Description
N/A	1	Version number	if less than 6, then version 0.
N/A	N/A	Date	YYYYMMDD
1	2	Mode	0 if good, 1 if bad CRCs, 2 if packet timeout
2	3	TOA time	Time of Arrival Version 6: yyyyMMddHHmmss.uuuuuu Version <6: HHmmss.ffff
3	4	TOA mode	Clock status
4	5	MFD peak	See $CAMFD description
5	6	MFD power	See $CAMFD description
6	7	MFD ratio	See $CAMFD description
7	8	SPL	See $CAMFD description
8	9	SHF-AGN	See $CASHF description
9	10	SHF-AINPSHIFT	See $CASHF description
10	11	SHF-AINSHIFT	
11	12	SHF-MFDSHIFT	See $CASHF description
12	13	SHF-P2BSHIFT	See $CASHF description
13	14	Rate	Packet rate
14	15	SRC	Packet Source ID
15	16	DEST	Packet destination ID
16	17	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	18	Packet Type	-1. Unknown 1: FSK 2: FSK Mini 3: PSK 4: PSK Mini
18	19	Nframes	Number of frames expected
19	20	nbad	Number of frames with bad CRCs
20	21	SNR-RSS	See $CASNR description

Field number (version 0)	Field number (version 6, micromodem-2)	Field Name	Description
21	22	SNR in	See $CASNR description
22	23	SNR out	See $CASNR description
23	24	Symbols SNR	See $CASNR description
24	25	MSE	Mean Squared Error from Equalizer
25	26	DQF	Data Quality Factor, FSK only
26	27	DOP	Doppler
27	28	Stdev noise	Std dev. of noise, dB
28	29	Carrier	Carrier frequency of received packet , Hz
29	30	Bandwidth	Bandwidth of received packet, Hz
N/A	TBD	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 10: Communications Packet Transmit Statistics Message Fields

Field number	Field number Version 6	Field Name	Description
N/A	1	Version number	Version number of the xst message
1	2	Date	YYYYMMDD
2	3	Time	Time of transmit, hhmmss.sss
3	4	Timing mode	Clock status
4	5	Mode	0. Transmit successful 1. FETS too hot 2. Extsync Timeout 3. TX inhibited 4. Data Timeout
5	6	Probe Length	Length of FM probe, in symbols
6	7	Bandwidth	Bandwidth of transmitted packet, Hz
7	8	Carrier	Carrier, Hz
8	9	Rate	Packet rate
9	10	SRC	Packet Source ID
10	11	DEST	Packet destination ID
11	12	ACK	Ack expected/not expected (1/0)
12	13	Nframes expected	Number of frames expected, from the CACYC message
13	14	Nframes sent	Number of frames sent
14	15	Packet Type	-1. Unknown, 1: FSK, 2: FSK Mini 3: PSK, 4: PSK Mini
15	16	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 23.)
***CS**	Hex coded checksum (8 bit XOR of sentence), optional

Table 11: Routine Errors

Type	Description
Bad Modulation Header	Modulation header did not decode on a PSK packet
BAD_CRC, Data Header	Data Header did not decode on a PSK packet
BAD_CRC	A specific frame did not decode on a comms packet
Error accessing coproc	The modem could not find the coprocessor
Equalizer Timeout	A PSK packet took too long to decode
Missed start of PSK pkt	A PSK packet probe was detected but the start of the packet was missed
Unknown minipacket command	A command minipacket was received with unknown command type
Unknown minipacket type	A minipacket was received with unknown type
TX inhibited	The modem 'TXINHIBIT' line is high, causing modem to drop a packet queued for transmit
PACKET_TIMEOUT	A modem, expecting a data packet within 'PTO' seconds, timed out on a detection

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 □Pa
SNR in	Input SNR, dB, channel 1 only, reported to 1 decimal place in version 6 of CACST
SNR out	SNR at the output of the equalizer, dB, reported to 1 decimal place in version 6 of CACST
Symbol SNR Noise level	Symbol SNR for spread spectrum packets only, dB 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

BBD, Baseband data dump configuration parameter

BBD Baseband data dump level

0 (default)	Baseband data dump disabled
1(legacy)	Dumps all baseband data from every PSK packet received, using CABBD format sentences.
2 (legacy)	Dumps all baseband data from 'Bad Mod Hdr' error packets only, using CABBD format sentences
3	Dump only training and header baseband data in a packet using the new BDD sentences
4	Dump the entire packet, using the new BDD sentences

CABBD

Dump of baseband data to serial port, modem to host(legacy). This includes 75 msec of pre-packet samples, which are used to estimate noise, and an extra 100 samples of padding at the end of the message.

$CABBD,nwords,hhhh…

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.
hhhh….	Baseband samples represented as hex-encoded 32-bit complex values (8 ASCII characters per sample).

CABDD

Dump of baseband data to serial port, modem to host. This includes 75 msec of pre-packet samples, which are used to estimate noise, and an extra 100 samples of padding at the end of the message. Each sentence has 256 bytes of hex encoded data samples. The last of these sentences can be zero padded.

$CABDD,msgnum,total_msgs,databytes,hhhh…*CS

msgnum	Message number being printed
total_msgs	Total number of messages to expect
databytes	Data bytes in current sentence, exclusive of padding .
hhhh	Baseband samples represented as hex-encoded 32-bit complex values (8 ASCII characters per sample).
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Assuming a serial baud rate of 19200, and single-channel receive, a full data packet dump can take up to 160 seconds (BBD=11), a training and header only dump (BBD=3) will take 11 seconds. At 115200 baud it drops to 2.4 seconds. The following table shows the number of $CABDD messages printed for each legacy packet type, along with the time take to print out all the messages at two different baud rates assuming a bandwidth of 2 kHz. Because the pre-packet padding is fixed at 75 milliseconds, the number of messages will change with the bandwidth.

Packet type	Number of messages,BDD=3	Number of messages,BDD=11	Total seconds to print full packet, 19200 baud	Total seconds to print full packet, 115200 baud
1	34	488	148	25
2	34	469	142	24
3	34	530	160	27
4	34	498	150	25
5	34	482	145	24

To print out a baseband dump of 8 bytes of data using the new Flexible Data Protocol at a bandwidth of 5000 Hz at the lowest rate (rate 1), requires 62 messages.

SNNST

Navigation statistics message, printed at the end of an LBL navigation cycle. Controlled by the nav.nst configuration parameter.

$SNNST,version,TX_Hz,TX_ms,yyyymmddhhmmss.ssssss,timing_mode,AGN,TAT,NAV_mode,det0_peak,det0_pow,det0_ratio,xpond1,xpond1_freq,det1_peak,det1_pow,det1_ratio,xpond1_owtt,… xpond4,xpond4_freq,det4_peak, det4_pow,det4_ratio,xpond4_owtt*CS

version	Version number of message, currently 0	
TX_Hz	Outgoing ping frequency, Hz	
TX_ms	Outgoing ping and replies width, ms	
yyyymmddhhmmss.ssssss	Timestamp of outgoing ping. In passive mode, timestamp of start of the receiver	
timing_mode	4-bit bitmask: bits 0-1 RTC source	
	00	RTC none
	01	RTC onboard
	10	RTC user command synced
	11	RTC GPS synced
	Bits 2-3 PPS source	
	00	PPS none
	01	PPS from RTC
	10	PPS from external pin
	11	PPS from RTC, sync'd to the last external PPS pulse.
AGN	Analog gain setting at receiver	
TAT	Turnaround time, ms, for transponders	
NAV mode	0, active nav, transmit ping 1, passive nav, do not transmit ping, listen for outgoing ping and replies.	
det0_peak	MFD detection peak, outgoing ping, blank if active nav cycle	
det0_pow	MFD detection power, dB, outgoing ping, blank if active nav cycle	
det0_ratio	MFD detection ratio, outgoing ping, blank if active nav cycle	
xpond1	Transponder number, 1-4	
xpond1_freq	Reply frequency, Hz, of transponder 1	
det1_peak	MFD detection peak, transponder 1, blank if no detect	

det1_pow	MFD detection power, dB, transponder 1, blank if no detect
det1_ratio	MFD detection ratio, transponder 1, blank if no detect
xpond1_owtt	One way travel time, ms, transponder 1, blank if no detect
xpond2	…
xpond4_owtt	One way travel time, ms, transponder 4, blank if no detect
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Configuration Sentences

CCCFG – Set configuration parameter, host to modem

Send NVRAM configuration parameter message. **See** CCCFQ – Query configuration parameter, host to modem **for a complete listing of the configuration parameters.**

$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 their factory defaults.

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.

Example: $CCCFQ,ALL

Query all configuration parameters.

In the Micromodem-2, configuration parameters are grouped together based on their functionality. Legacy parameters are still available, but the new parameters will supersede legacy parameters in case of overlapping functionality. $CCCFQ is used to query the value of a single parameter or a group. $CCCFG is used to set the value of a single parameter.

CCCFQ,TOP- Query all configuration parameters

A comprehensive list of all configuration parameters. Each grouping is further described in the section relevant to its functionality. For example, to learn more about the 'hibernate' group, see the section on hibernating.

$CCCFQ,TOP

In response to this query, the modem prints:

$CACFG,<configuration parameter>,<value>*CS

For convenience, the list is broken up into subgroups and tabulated, each with its own query command.

CCCFQ,ALL- Query legacy configuration parameters

Listing of legacy parameters only.

$CCCFQ,ALL

Table 10: Legacy Configuration Parameters

Parameter Name	Description	Minimum value	Maximum Value	Default Value
AGC	Turn on automatic gain control	0	1	1
AGN	Analog Gain (50 is 6 dB, 250 is 30 dB)	0	255	250
AGX	Transmit AGC sequence for PSK	0	1	1
ASD	Always Send Data. Tells the modem to send test data when the user does not provide any.	0	1	0
BBD	PSK Baseband data dump to serial port	0	4	0
BND	Frequency Bank (1, 2, 3 for band A, B, or C, 0 for user-defined PSK only band)	0	3	3
BR1	UART1 baud rate setting bitrate 0 2400 1 4800 2 9600 3 19200 4 38400 5 57600 6 115200 7 230400	0	7	3
BR2	UART2 baud rate	0	7	3
BR3	UART3 baud rate	0	7	3
BR4	UART4 baud rate	0	7	3
BRN	Run bootloader at next revert	0	1	0
BSP	Boot loader serial port	1	4	1
BW0	Bandwidth for Band 0 PSK, Hz Allowable bandwidths are 1000,1250,2000,2500,4000,5000	0	250000	4000
CPR	Coprocessor power toggle switch	0	1	1
CRL	Cycle init reverb lockout (ms)	0	32767	50
CST	Cycle statistics message	0	1	1

CTO	Cycle init timeout (sec)	0	255	10
DBG	Enable low-level debug messages	0	1	0
DGM	Diagnostic messaging	0	1	0
DOP	Whether or not to send the $CADOP message	0	1	0
DQF	Whether or not to send the $CADQF message	0	1	1
DTH	Matched filter signal threshold, FSK	0	32767	108
DTO	Data request timeout (sec)	1	30	2
DTP	Matched filter signal threshold, PSK	0	32767	90
ECD	Delay at end of cycle (ms)	0	32767	50
EFB	Feedforward taps for the LMS equalizer	10	100	20
EFF	Feedback taps for the LMS equalizer	10	100	10
FC0	Carrier at Band 0	0	100000	25000
FMD	PSK FM probe direction,0 up, 1 down	0	1	1
FML	PSK FM probe length, symbols	10	255	200
GPS	GPS parser on aux. serial port	0	1	0
HFC	Hardware flow control on main serial port . Currently, non-functional.	0	1	0
IRE	Print impulse response of FM sweep	0	1	0
MCM	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.	0	1	1
MFD	Whether or not to send the MFD messages	0	1	0
MOD	0 sends FSK minipacket, 1 sends PSK minipacket	0	1	0
MPR	Enable power toggling on Multi-Channel Analog Board	0	1	1
MSE	Print symbol mean squared error (dB) from the LMS equalizer	0	1	0

MVM	Enable voltage mode hydrophone power supply on Multi-Channel Analog Board	0	1	1
NDT	Detect threshold for nav detector	0	32767	120
NRL	Navigation reverb lockout (ms)	0	32767	25
NRV	Number of CTOs before hard reboot	0	255	150
PAD	Power-amp delay (ms)	0	100	2
PCM	PSK multichannel channel mask	0	255	0
POW	Detection power threshold (dB)	-128	128	-100
PRL	Packet reverb lockout (ms)	0	32767	50
PTH	Matched filter detector power threshold	0	32767	50
PTO	Packet timeout (sec)	0	1000	14
REV	Whether or not to send the $CAREV message	0	1	1
RXA	Whether or not to send the $CARXA message	0	1	0
RXD	Whether or not to send the $CARXD message	0	1	1
RXP	Whether or not to send the $CARXP message	0	1	0
SCG	Set clock from GPS	0	1	0
SGP	Show GPS messages on main serial port	0	1	1
SHF	Whether or not to send the $CASHF message	0	1	0
SNR	Turn on SNR stats for PSK comms	0	1	0
SNV	Synchronous transmission of packets	0	1	0
SRC	Default Source Address	0	255	0
TAT	Navigation turn-around-time (msec)	0	32767	50
TOA	Display time of arrival of a packet (sec)	0	1	0
TXD	Delay before transmit (ms)	0	32767	600
TXP	Start of transmit message	0	1	1
TXF	End of transmit message	0	1	1
XST	Turn on transmit stats message, CAXST	0	1	1

CCCFQ,uart<n>- Query UART configuration parameters

Listing of UART parameters. There are four UARTs on the Micromodem-2:

uart1: RS232 with optional hardware flow control (RTS/CTS).

uart2: RS232 without hardware flow control.

uart3: logic-level (3.3V) with optional hardware flow control.

uart4: Switchable between RS232 (optional hardware flow control) and RS485 (full- or half-duplex).

$CCCFQ,uart<n>

Parameter Name	Description	Min Value	Max Value	Default Value	Legacy Parameter
uart<n>.bitrate	Baudrate, allowable values are: 2400, 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600	2400	921600	19200	BR1, BR2, BR3, BR4
uart<n>.task	0 OFF 1 NMEA 2 reserved 3 reserved 4 RAW	0	4	1	N/A
uart<n>.parse_gps	parse incoming GPS messages	0	1	0	GPS=uart2.parse_gps
uart<n>.show_gps	pass GPS messages through	0	1	0	SGP=uart1.show_gps
uart<n>.set_clk_GPS	set modem clock from GPRMC and GPZDA sentences	0	1	0	SCG=uart2.set_clk_GPS
uart<n>.flowcontrol (uart1, uart4)	use hardware flow control. Currently non-functional	0	1	0(OFF)	HFC=uart1.flowcontrol
uart<n>.crc32	New 32 bit CRC(1) vs. legacy 8-bit CRC(0)	0	1	0	N/A
uart4.rs485	0=RS232	0	2	0	N/A

	1=full-duplex RS485 2=half-duplex RS485				

CCCFQ,log- Query logging configuration parameters

Listing of logging parameters. See NMEA Data Logging for further details on how to use them.

$CCCFQ,log

Parameter Name	Description	Minimum value	Maximum Value	Default Value
log.nmea.level	NMEA log level	0	2	0
log.nmea.location	log location	1	1	1, Read-only
log.nmea.lastretrieval	last time a log file was retreived, in microseconds			0
log.nmea.correntdbSize	current size of SQLite database bytes	0	59768832	0

CCCFQ,fsk- Query FSK packet configuration parameters

Listing of FSK packet parameters. Only legacy_addressing is settable. Carrier is set using the legacy parameter, BND. Bandwidth, nulltime and modulation are fixed parameters.

$CCCFQ,fsk

Parameter Name	Description	Minimum value	Maximum Value	Default Value	Notes
fsk.packet.modulation	modulation type	0	0	0	Read-only
fsk.packet.bandwidth_Hz	bandwidth	4000	4000	4000	Read-only
fsk.packet.carrier_Hz	carrier, Hz	See BND	See BND	25120	Read-only
fsk.packet.nulltime_ms	nulltime, ms	200	200	200	Read-only
fsk.packet.legacy_addressing	4-bit addressing	0	1	0	

CCCFQ,psk- Query PSK packet configuration parameters

Listing of PSK packet parameters.

$CCCFQ,psk

Parameter Name	Description	Minimum value	Maximum Value	Default Value
psk.packet.mod_hdr_version	0 legacy	0	1	0

	1 FDP protocol			
psk.packet.modulation	modulation type	0	0	0
psk.packet.bandwidth_Hz	current bandwidth, Hz	10	10000	5000
psk.packet.carrier_Hz	current carrier, Hz	100	125000	25000
psk.packet.bandwidth0_Hz	band 0 bandwidth, Hz	10	10000	5000
psk.packet.carrier0_Hz	band 0 carrier, Hz	100	125000	5000
psk.packet.nulltime_ms	nulltime, ms	100	5000	250

CCCFQ,detector- Query detector configuration parameters

Listing of detector parameters. Settings common to all detectors are listed under 'detector' while specific settings are listed under 'detector<n>', where n=1,2 or 3.Typically detector1 listens for FSK packets, detector2 listens for PSK packets and detector3 listens for navigation pings.

$CCCFQ,detector

Parameter Name	Description	Minimum value	Maximum Value	Default Value
detector.power_thresh	power threshold for detection	-128	128	-100
detector.noise_thresh	noise threshold	0	32767	50
detector1.channel_mask	8-bit bitmask bit chan 0 onboard 1-4 MCA card 5-7 reserved	0	511	1
detector1.type	The signal to listen for(FM sweep)	0	0	0
detector1.fm_dir	Direction of FM sweep	0	1	0
detector1.length_sym	length of sweep, symbols	10	1000	40
detector1.fm_bw_Hz	Bandwidth, Hz	10	10000	4000
detector1.carrier_Hz	Carrier, Hz	100	125000	25120
detector1.thresh	detection threshold	0	32767	108
detector2.channel_mask	8-bit bitmask bit chan 0 onboard 1-4 MCA card 5-7 reserved	0	511	1
detector2.type	The signal to listen for(FM sweep)	0	0	0

detector2.fm_dir	Direction of FM sweep	0	1	1
detector2.length_sym	length of sweep, symbols	10	1000	200
detector2.fm_bw_Hz	Bandwidth, Hz	10	10000	4000
detector2.carrier_Hz	Carrier, Hz	100	125000	25120
detector2.thresh	detection threshold	0	32767	90
detector3.channel_mask	8-bit bitmask bit chan 0 onboard 1-4 MCA card 5-7 reserved	0	511	1
detector3.type	The signal to listen for(tone)	0	2	1
detector3.fm_dir	Direction of FM sweep	0	1	0
detector3.length_sym	length of sweep, symbols	10	1000	40
detector3.fm_bw_Hz	Bandwidth, Hz	10	10000	4000
detector3.carrier_Hz	Carrier, Hz	100	125000	25120
detector3.thresh	detection threshold	0	32767	50

CCCFQ,recv- Query receiver configuration parameters

Listing of receiver parameters. These operate on the receiver after a detection has taken place.

$CCCFQ,recv

Parameter Name	Description	Minimum value	Maximum Value	Default Value
recv.mca.gain	MCA gain	0	255	250
recv.onboard_gain	Onboard gain setting	0	255	250
recv.onboard_equalizer	Use onboard equalizer	0	1	0
recv.agn_legacy	Use legacy AGN setting instead of recv.mca.gain and recv.onboard_gain	0	1	1
recv.p2b_chmask	8-bit bitmask for packet bit chan 0 onboard 1-4 MCA card 5-7 reserved	0	511	1
recv.agc_length_ms	AGC sequence length, ms	0	1000	100
recv.agc_legacy	AGC on noise seq.	0	1	1

recv.digital_rcvr	Bypass analog in and use baseband digital data over SPI interface. Not for general use, special hardware required	0	1	0
recv.base64data	Encode data in CARDP as base64	0	1	0

CCCFQ,xmit- Query transmitter configuration parameters

Listing of transmitter parameters.

$CCCFQ,xmit

Parameter Name	Description	Minimum value	Maximum Value	Default Value
xmit.poweramp_gating	gate power amp for null time	0	1	0
xmit.txinhibit	Check TXINHIBIT input pin	0	1	0
xmit.uart_disable_mask	Disable specified UARTs during transmit (bit mask)	0	15	0

CCCFQ,timing- Query detector configuration parameters

Listing of timing parameters.See section on Using and Setting the Clock for further details.

$CCCFQ,timing

Parameter Name	Description	Minimum value	Maximum Value	Default Value
timing.clock_source	clock source	0	3	0, Read-only
timing.PPS_source	PPS source	0	4	0, Read-only
timing.syncnav	transmit on a PPS edge	0	1	0
timing.pps_timeout	PPS receipt timeout	0	300	3

CCCFQ,hibernate- Query hibernate configuration parameters

Listing of hibernate parameters. See section on Hibernating for further details

$CCCFQ,hibernate

Parameter Name	Description	Minimum value	Maximum Value	Default Value
hibernate.wake_interval	wake interval in sec	0	2419200	0
hibernate.wake_reference	wake reference, sec	0	2147483647	0
hibernate.hibernate_after	hibernate-after time, sec	0	2419200	0

CCCFQ,nav- Query navigation configuration parameters

Listing of navigation parameters. See section on USBL navigation and on Tracking ping for further details.

$CCCFQ,nav

Parameter Name	Description	Minimum value	Maximum Value	Default Value
nav.soundspeed_mps	sound speed, meters per second	0	3000	1430
nav.nst	show nav stat msg	0	1	0
nav.usbl.mode	USBL mode	0	3	0
nav.usbl.dbg	Debug flag	0	1	0
nav.usbl.array_type	Use to set default array locations for commonly used array heads	0	10	0
nav.usbl.array_locs<n>_x_mm	USBL array locations, x-position (mm), n=1-4	-32768	32767	{-15, +15, +15, -15}
nav.usbl.array_locs<n>_y_mm	USBL array locations, y-position (mm)	-32768	32767	{+15, +15, -15, -15}
nav.usbl.array_locs<n>_z_mm	USBL array locations, z-position (mm)	-32768	32767	{0, 0, 0, 0}
nav.usbl.channel_mask	channel mask	0	511	30
nav.trackping.mode	Tracking Pinger mode	0	4	0

nav.trackping.timing_mode	Timing mode	0	1	0
nav.trackping.carrier_Hz	Carrier for the pinger signal, Hz	100	125000	13500
nav.trackping.cycles_per_sym	Used instead of bandwidth for user defined signals	1	1000	7
nav.trackping.period	Repetition period, seconds	1	255	1
nav.trackping.reference_time	Reference time, unix seconds . 0 indicates no reference time required	0	4294967295	1449187200
nav.trackping.data	If a sequence is being sent, use this data as packed unsigned 64-bit int	0	2^{64}	0
nav.trackping.sequence_len	Length of sequence in bits	1	64	47
nav.trackping.bandwidth_Hz	Bandwidth of signal	10	50000	4000
nav.trackping.duration_msx10	Duration of signal,if not user defined in millisecods times 10.	0	32767	35

CCCFQ,rec Query recording configuration parameters

Listing of recording information(read only) .

$CCCFQ,rec

Parameter Name	Description	Minimum value	Maximum Value	Default Value
rec.state	recording state 0 Idle, not recording 1 recording 2 Done recording, printing	0	4	0

rec.start_time	recording start time, unix, seconds. If not recording, this is set to -1	-1	2147483647	0
rec.end_time	recording end time, unix, seconds. This shows the end time of the current recording, or if the modem is done recording, the end time of the previous recording.	-1	2147483647	0

CCCFQ,pwramp Query poweramp configuration parameters

Listing of power amp settings. Only valid with 202006d and 202009c and newer variants of power amps.

$CCCFQ,pwramp

Parameter Name	Description	Minimum value	Maximum Value	Default Value	Notes
pwramp.temperature_degC	temperature, ˚C	-274.0	+274.0	N/A	Read-only
pwramp.vbat	battery Voltage	0.0	26.0	N/A	Read-only
pwramp.vtransmit	power amp voltage during transmit	0.0	26.0	N/A	Read-only
pwramp.iout_sense_amps	Power Amp IOut Sense current	0.0	26.0	N/A	Read-only
pwramp.fault	Faulted? 0/1	0	1	N/A	Read-only
pwramp.txlevel	Bitfield 0 High 1 Med-High 2 Med-Low 3 Low	0	3	0	

CCCFQ,info Query board information parameters

Listing of software and hardware information. **(These parameters are read-only.)**

$CCCFQ,info

Parameter Name	Description	Typical Value

info.part_number	part number	201005
info.serial_number	serial number	511
info.board_rev	board revision	68 (ASCII 'D')
info.bom_variant	BOM variant	77 (ASCII 'M')
info.bom_rev	BOM revision	1
info.fpga_version	FPGA version	2.0.21446
info.fpga_api_level	FPGA API level	3
info.fpga_flashrom_version	FPGA flashrom version	2
info.build_oem	build OEM	65 (ASCII 'A')
info.build_year	build year	2013
info.build_month	build month	1
info.build_options	build options	0
info.coproc_version	coprocessor version	x.xx.x.xx.xxxxx
info.firmware_version	firmware version	x.x.xxxxx
info.loader_version	loader version	17175
info.booted_slot	current firmware booted slot	1

CCCFQ,fathometer Query fathometer parameters

Listing of fathometer parameters. See fathometer application for an example application.

`$CCCFQ,fathometer`

Parameter Name	Description	Minimum value	Maximum value	Default Value
fathometer.active	transmit schedule active	0	1	0
fathometer.hibernate_after	hibernate after npings otherwise de-activate after npings	0	1	0
fathometer.t0_ms	lockout end-of-txtime to start-of-capture-window (ms)	0	20000	1333
fathometer.tcapture_ms	total capture window (ms)	0	20000	5333

fathometer.report_full_window	report full fathometer window rather than window around peak	0	1	0
fathometer.dt_peak_ms	peak report window duration (ms)	0	20000	300
fathometer.pre_peak_pct	peak report window - percentage of reported IRE which will be before peak	0	100	10
fathometer.decimate_by_2	decimate report window by 2	0	1	0
fathometer.npings_total	number of pings per cycle	1	1000	1
fathometer.npings_over_iridium	number of pings transmitted over Iridium per cycle	1	1000	0
fathometer.min_dx_m	minimum distance from last ping	0	100000	0
fathometer.last_lat_deg	latitude of last ping (decimal degrees)(RDONLY)	-100	100	-100
fathometer.last_lon_deg	longitude of last ping (decimal degrees)(RDONLY)	-200.0	200.0	-200
fathometer.diagnostics_over_iridium	send non-fathometer NMEA sentences over Iridium	0	1	0
fathometer.ire_over_iridium	send IRE NMEA sentences over Iridium	0	1	0
fathometer.use_base64	send IRE with base64 encoding rather than ASCII hex	0	1	1

CCCFQ,hwd Query and set board hardware parameters

Listing of software and hardware information. **(These parameters are read-only.)**

$CCCFQ,hwd

Notes on hwd.gpio<n>

Parameter Name	Description	Minimum value	Maximum Value	Default Value
hwd.batt_V	Input battery voltage	0.0	-	
hwd.aux_adc_v	Aux ADC voltage	0.0	6.0	Read-only
hwd.aux_adc_gain	Aux ADC gain	0	255	1
hwd.aux_adc_res	Aux ADC resolution, bits	0	255	12
hwd.aux33V_on	Aux 3.3V enabled	0	1	1
hwd.aux33V_fault	Aux 3.3V Fault	0	1	Read-only
hwd.stack33V_on	Stack 3.3V enabled	0	1	1
hwd.stack33V_fault	Stack 3.3V Fault	0	1	Read-only
hwd.Vpwramp_on	Vpwramp enabled	0	1	1
hwd.Vpwramp_fault	Vpwramp fault	0	1	Read-only
hwd.Vpreamp_on	Vpreamp enabled	0	1	1
hwd.Vpreamp_fault	Vpreamp fault	0	1	Read-only
hwd.VSDcard_on	VSDcard enabled	0	1	0
hwd.VSDcard_fault	VSDcard fault	0	1	Read-only
hwd.gpio<n>, n=1-5	Set or read hardware GPIO pins 1-5.	0	2	0=00: Output low 1=01: Output high 2=10: Input (low) 3=11: Input (high)
hwd.txinhibit	TXINHIBIT pin value	0	1	Read-only
hwd.temp	Temperature, ◦C	-274.0	+274.0	Read-only

Notes on hwd.gpio<n>:

The Micromodem-2 has five GPIO pins, GPIO 1-5.

To configure a pin, for example GPIO 1, do:

Set GPIO 1 as output, value low:	$CCCFG,hwd.gpio1,0
Set GPIO 1 as output, value high:	$CCCFG,hwd.gpio1,1
Set GPIO 1 as input:	$CCCFG,hwd.gpio1,2

To read the value of GPIO 1:	$CCCFQ,hwd.gpio1
Value read will give:	
GPIO 1 is an output, value low:	$CACFG,hwd.gpio1,0
GPIO 1 is an output, value high:	$CACFG,hwd.gpio1,1
GPIO 1 is an input, value low:	$CACFG,hwd.gpio1,2
GPIO 1 is an input, value high:	$CACFG,hwd.gpio1,3

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

Synchronous Navigation

The Micromodem-2 allows precise transmission of signals on a PPS edge. See Reference **Invalid source specified.** for implementation details. However, there are a few differences in how the PPS state is maintained in the modem. The Micromodem-2 can synchronize the RTC PPS to an external PPS. Hence, if external PPS is lost, the modem can still stay synchronized with clock state set to 'EXT_SYNC'. So a packet that is queued up for synchronous transmit is always sent so long as RTC PPS is available.

The Micromode-2 shows increased performance accuracy over the Micromodem-1 with respect to one-way travel time. Tests conducted over the wire (no cables or transducers) showed a mean error of 84 microseconds with a standard deviation of 42 microseconds at 4 kHz bandwidth, which is less than half a baseband sample (125 microseconds).

Using and Setting the Clock

The modem can set its real-time clock based on an external signal received on a GPIO pin if the user prepares the modem with the CCTMS command.

CCTMS

Set the modem clock, 'TiMeSet'

$CCTMS,YYYY-MM-ddTHH:mm:ssZ,mode *CS

YYYY-MM-ddTHH:mm:ssZ	Time that will be set, in ISO 8601 Extended format. The time zone specifier must be included and set to "Z".
mode	**0** Set time for the "current" second. If a PPS signal is present, this will synchronize the clock to the specified time based on the last PPS edge. **1** Set specified time on the next rising edge of the External PPS signal. If providing a single pulse on the External PPS line for synchronization, the rising edge of the pulse must occur within 2 seconds of sending this command.
***CS**	Optional: The NMEA sentence checksum, hex coded (8- bit XOR or CRC-32). See Message Checksums.

The modem will respond to this command with a **CATMS** message once the time is set or a 3-second timeout elapses (only applicable when mode is 1). The **CATMS** message is therefore not sent immediately in response to this command.

Additionally, if the clock is set successfully and the clock source has changed, an informational message, CATMG, will be printed.

If using the External PPS (EXTPPS) signal, it must remain stable (both high and low) for at least 10us, and it must not be asserted until after the modem has processed the CCTMS message. The host serial port often introduces significant latency that cannot be predicted. Therefore, we suggest sending this command at least several hundred milliseconds before a rising edge on External PPS.

CATMS

Response to set clock command

$CATMS,timed_out,YYYY-MM-ddTHH:mm:ssZ*CS

timed_out	**0** (time was set) if the CATMS command mode was 0 or if the External PPS signal had a rising edge within the timeout period and the time was set successfully. **1** (time was not set) if mode was 1 and the timeout period elapsed. The time was not set.
YYYY-MM-ddTHH:mm:ssZ	Time that was set, in ISO 8601 Extended format. If `timed_out` is 1, this field will be empty.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message will be sent by the modem following a **CCTMS** command when the clock is set or when the timeout period elapses.

Examples

In examples, **`messages in bold`** are commands sent from the host to the modem and `messages not in bold` are messages sent from the modem to the host. *CS* is the NMEA message checksum.

Set the clock immediately

This example sets the modem clock to the specified time.

$CCTMS,2012-12-01T01:04:05Z,0

$CATMS,0,2012-12-01T01:04:05Z**CS*

Set the clock on the next rising edge of External PPS

This example shows the message that will appear if the rising edge on External PPS (EXTPPS) occurs within two seconds of the command being sent.

$CCTMS,2012-12-01T01:04:05Z,1

$CATMS,0,2012-12-01T01:04:05Z**CS*

This transaction shows what would occur if the timeout elapsed before the rising edge occurred.

$CCTMS,2012-12-01T01:04:05Z,1

$CATMS,1,**CS*

CCTMQ

Query modem time command

`$CCTMQ,0*CS`

0	Reserved field, must be 0.
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The modem will respond with the current time and time source information in the **CATMQ** message.

CATMQ

Response to time query (CCTMQ) command

$CATMQ,YYYY-MM-ddTHH:mm:ssZ,clock_source,pps_source*CS

YYYY-MM-ddTHH:mm:ssZ	Current time, in ISO 8601 Extended format.
clock_source	**NONE** If there is no clock source present. This is an error condition. **RTC** if the clock source is the internal RTC modem clock. This could have previously been synchronized with an external clock. **USER_CMD** If the clock was set using the user command **CCCLK** or **CCTMS**. **GPS** if the clock source is synchronized to a GPS string as controlled by the FRAM parameter **SCG**.
pps_source	**NONE** If there is no PPS source present. This is an error condition. **RTC** if the PPS source is the internal RTC modem PPS. **EXT** If the PPS source is the external PPS pin. **EXT_SYNC** if the PPS RTC pulse source is synchronized to the last valid external PPS pulse.
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

The parameters for this message are identical to those of the CATMG message (the only difference is the message ID).

CATMG

Informational message about timing source, printed when timing sources change

$CATMG,YYYY-MM-ddTHH:mm:ssZ,clock_source,pps_source*CS

YYYY-MM-ddTHH:mm:ssZ	Current time, in ISO 8601 Extended format.

clock_source	**NONE** If there is no clock source present. This is an error condition. **RTC** if the clock source is the internal RTC modem clock. This could have previously been synchronized with an external clock. **USER_CMD** If the clock was set using the user command **CCCLK** or **CCTMS**. **GPS** if the clock source is synchronized to a GPS string as controlled by the FRAM parameter **SCG**.
pps_source	**NONE** If there is no PPS source present. This is an error condition. **RTC** if the PPS source is the internal RTC modem PPS. **EXT** If the PPS source is the external PPS pin. **EXT_SYNC** if the PPS RTC pulse source is synchronized to the last valid external PPS pulse.
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message will be sent by the modem following a change of status in either the clock source or the PPS source. This could happen after a reboot, or the execution of a **CCTMS** command or a **CCCLK** command.

The parameters for this message are identical to those of the CATMQ message (the only difference is the message ID).

CCCLK – Set clock, host to modem

Deprecated for uM2; use CCTMS instead.
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

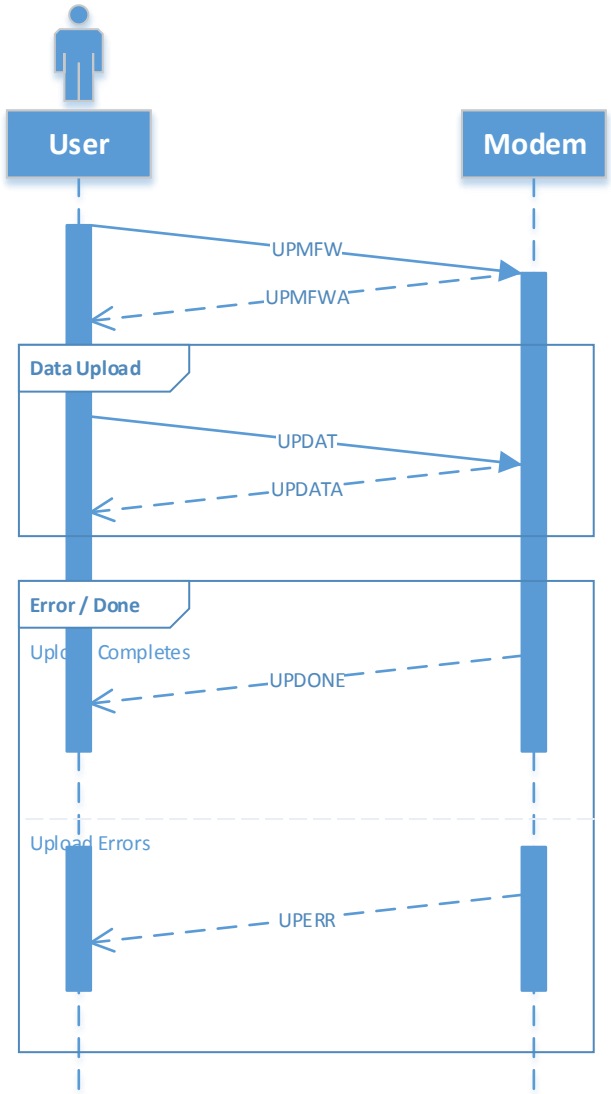
Firmware Update

The modem can update its firmware during operation. To update firmware over the serial port, the modem needs to boot into the recovery slot from flash using the $CCRST command.

The AComms Group can provide a python script or an executable to update the firmware, described below in the section on pyAComms Support. Alternately, if users need to implement their own firmware update software, the user can command the modem start the update process using the **UPMFW** command. The user then must send the **UPDAT** commands to load the firmware.

The modem can support two separate user update-able firmware slots.

Below is the Message Sequence Diagram:



UPMFW

Command modem to start upload of new firmware.

$UPMFW,firmware_slot,data_location,total_data_size,force_reboot,sha1_hash,firmware_filename *CS

firmware_slot	**1** if the modem will be flashing the first user slot **2** if the modem will be flashing the second user slot **0** if the modem will be flashing the recovery slot
data_location	**0** if modem is loading from NMEA
total_data_size	The number of bytes in firmware file.
force_reboot	**0** if the modem will be rebooted by the user to the new firmware **1** if the modem will be automatically rebooted upon completion
sha1_hash	SHA1 hash of the firmware file for upload verification in Hex coded bytes, 2 characters each, e.g. 00-FF.
firmware_filename	Firmware filename.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

After the UPMFW command is sent, the modem will reply with a UPMFWA message that indicates when the modem is ready to start accepting firmware data using the UPDAT message. The SHA1 Hash is generated following the sha1sum method of the Unix operating system utility.

UPMFWA

Response to upload start command

$UPMFWA,firmware_slot,data_location,total_data_size,sha1_hash,firmware_filename*CS

firmware_slot	**1** if the modem will be flashing the first user slot **2** if the modem will be flashing the second user slot **0** if the modem will be flashing the recovery slot
data_location	**0** if modem is loading from NMEA
total_data_size	The number of bytes in firmware file.
sha1_hash	SHA1 hash of the firmware file for upload verification
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent in response to a **UPMFW** command.

UPDAT

Upload of piece of firmware data.

$UPDAT,data_format,data*CS

data_format	**0** if the data is encoded in base64 data **1** if the data is encoded in Hex coded data bytes.
data	If data_format is 0, then data is base64 encoded bytes. Maximum of 1K per message. If data_format is 1, then data is hex coded data bytes, 2 characters each, e.g. 00-FF, Maximum of 512 bytes per message.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent after the UPMFWA message is received. This message is sent multiple times to handle the upload of each piece of the firmware file. Each data packet is assumed to be received sequentially with respect to the firmware file. The modem will respond with a UPDATA message after each data chunk is received.

UPDATA

Response to upload data command

$UPDATA,nbytes*CS

nbytes	Number of bytes received in data chunk.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent in response to a **UPDAT** command.

UPERR

Response to errors within firmware update process

$UPERR,errno,error_msg*CS

errno	Specific error number referencing error message below.
error_msg	Human Readable Error Message detailing the fault in the upload process.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-

	bit XOR or CRC-32). See Message Checksums.

This message is sent in response to any errors that occur within the update process. If received, modem has ignored this transfer and it needs to be restarted.

UPDONE

Response to upload firmware completion

$UPDONE,Update Complete,msg*CS

msg	Done Message.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

This message is sent in response to a completed and successful firmware update. Depending on the configuration of the update, the message will reference to the reboot option.

pyAcomms Support

Not Currently Supported. However, using a downloadable python script (fw_update.py) from our website, the firmware on the Micromodem-2 can be upgraded following these steps. The user will need access to the serial port on the modem.

1. Boot the modem into the recovery slot, using $CCRST,0
2. Disconnect from the serial terminal.
3. From a command prompt, type in:
   python update_fw.py <serial port> <baudrate> <firmwarefile.ldr>
   Where:
   `Serial port` is serial port identifier that the micromodem is connected on depending on operating environment. (I.E, if Windows then COMx or if Linux /dev/ttyXX etc.).
   `Baudrate` is the baud rate the modem UART is set to, which is also the serial terminal baudrate.
   `firmwarefile.ldr` is the firmware updater ldr file.
4. The script will establish a connection with the com port and upload the new firmware into the requested slot. It will show the $UPDATA message.
5. Once it is done, the $UPDONE message will be printed along with statistics on time to upload, filesize and number of packets sent. The user can now re-connect to the serial port on the modem and wait for it to reboot into the newly programmed slot. If there was an error in programming the firmware, the modem will reboot into the recovery slot.
6. To force the modem to boot into the newly-programmed slot, type $CCRST,1 or $CCRST,2
7. The script changes the baudrate to 115200 to improve the speed of the upload.

Acoustic NMEA command

An acoustic nmea command takes an NMEA command and sends it acoustically at a given rate using the new FDP minipacket.

CCACM

Acoustic NMEA command packet.

$CCACM,dest,rate,ack,cmd_string*CS

dest	Destination ID for command
rate	The rate at which to send the command, currently only 1 and 3 are valid.
ack	Expect an acknowledgement from the destination modem, if set to 1, else no ack is expected.
cmd_string	The NMEA command to be sent, minus the dollar sign and checksum
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Upon receipt of the command on the originating modem, a $CAACM… is issued and the command is sent acoustically as a FDP minipacket. Currently this command exists only for the USBL CCXSB command. Upon receipt of the packet, if the destination ID matches the source ID of the receiving modem, an ack is sent if requested, and the command is implemented.

CAACM

Response to acoustic NMEA command packet.

$CAACM,dest,rate,ack,cmd_string*CS

dest	Destination ID for command
rate	The rate at which to send the command, currently only 1 and 3 are valid.
ack	Expect an acknowledgement from the destination modem, if set to 1, else no ack is expected.
cmd_string	The NMEA command to be sent, minus the dollar sign and checksum
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

CAACA

Reciept of acoustic NMEA command packet.

$CAACA,dest,rate,ack,cmd_string*CS

dest	Destination ID for command
rate	The rate at which the command was received.
ack	Echo of the acknowledgement field.
cmd_string	Echo of the NMEA command that was sent, minus the dollar sign and checksum
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Upon reception of an acoustic command minipacket, the destination modem issues a $CAACA message. Additionally if the destination ID on the command matches the source ID of the modem, it executes the command.

CAACR

Acknowledgement of an acoustic NMEA command packet.

$CAACR,dest,rate,ack,cmd_string*CS

dest	Destination ID for command (source ID of the modem issuing the ack)
rate	The rate at which the command was received, same as the rate at which it was sent.
ack	Set to 0.
cmd_string	The NMEA command received, minus the dollar sign and checksum
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

USBL navigation

The Micromodem USBL mode provides azimuth and elevation to another Micromodem system which is configured as a USBL transmitter. It also provides range to the transmitter when both are equipped with either GPS PPS or another stable time source with a hardware pulse output.

A Micromodem-2 equipped with a multi-channel receiver card and a 4-element USBL array functions as the receiver. The receiver may or may not include a transmit capability. A Micromodem-2 with source operates as the source. Host computers at both the dock and the homing vehicle are assumed to exist and provide high-level control of the modems.

This version of the firmware uses an approach which allows the host computers to control all aspects of the USBL mode of operation. Future versions could be more sophisticated, but this approach makes visible to the user every step of the process.

Unambiguous Ranging Discussion

The first release of the capability described here will require the repetition rate to be such that the range is not ambiguous, i.e. there is not more than one ping in the water at a time. Future versions will have an additional several bits of tag data to follow each ping such that their emission time can be determined at the receiver.

As an example, if the transmitter is configured at 1 Hz the vehicle can home from less than 1 second, or 1500 meters, away by assuming that a ping, when received, was transmitted at the next ping time minus the inter-ping delay. In other words for the 1 second case, if a reception occurs at xx:0.70 sec, the range is computed by assuming the ping was sent at xx:0.0, and thus the travel time is 0.70 seconds. For a 4 Hz rate, when the signal is received at 0.7 seconds it is assumed to be transmitted at xx:0.5 seconds and the travel time is 0.2 seconds. At four Hz the maximum unambiguous range is 375 m. Note that the PING command (CCPNG) can be used at any time before the USBL mode is enabled to measure the actual range.

USBL Navigation Configuration Parameters

All parameters related to USBL navigation are controlled using parameters in the **nav** configuration group.

nav.soundspeed_mps Sound speed in meters per second in urethane.

1430	default

nav.usbl.mode0/[1,2,3], off/[on-FSK, on-PSK, on-both]

0	Default, no USBL receiver, unless explicitly commanded by $CCRSB.
1	USBL receiver on all received FSK probes, comms on PSK probes
2	USBL receiver on all received PSK probes, comms on FSK probes
3	USBL receiver and comms on FSK and PSK probes (not yet implemented)

nav.dbg debug (verbose) flag.

0	Default, no debug message
1	Print detector performance messages if the corresponding config parameters are set. They are MFD, TOA, SHF and IRE. Additionally, print the $CAUIR and $CAUXY messages.

nav.array_type use array locations for a particular array type, ignore array_locs setting.

0	Default, no array type entered, use array_locs
1	Use trackpoint array locations
2	Use MSI array locations
3	Use Btech array locations

nav.usbl.array_locs1_x_mm element 1 array x-location in mm

-15 (default)	Element 1 array x-location, mm

nav.usbl.array_locs1_y_mm element 1 array y-location in mm

15 (default)	Element 1 array y-location, mm

nav.usbl.array_locs1_z_mm element 1 array z-location in mm

0 (default)	Element 1 array z-location, mm

nav.usbl.array_locs2_x_mm element 2 array x-location in mm

15 (default)	Element 2 array x-location, mm

nav.usbl.array_locs2_y_mm element 2 array y-location in mm

15 (default)	Element 2 array y-location, mm

nav.usbl.array_locs2_z_mm element 2 array z-location in mm

0 (default)	Element 2 array z-location, mm

nav.usbl.array_locs3_x_mm element 3 array x-location in mm

15 (default)	Element 3 array x-location, mm

nav.usbl.array_locs3_y_mm element 3 array y-location in mm

-15 (default)	Element 3 array y-location, mm

nav.usbl.array_locs3_z_mm element 3 array z-location in mm

0 (default)	Element 3 array z-location, mm

nav.usbl.array_locs4_x_mm element 4 array x-location in mm

-15 (default)	Element 4 array x-location, mm

nav.usbl.array_locs4_y_mm element 4 array y-location in mm

-15 (default)	Element 4 array y-location, mm

nav.usbl.array_locs4_z_mm element 4 array z-location in mm

0 (default)	Element 4 array z-location, mm

nav.usbl.channel_mask The bitwise channel mask used by the detector to listen to the USBL array, channel one being the one used for detection.

30	Default, all four channels are on.

CCXSB

Command modem to start or stop transmitting USBL pings at a specified repetition rate for nseconds.

$CCXSB,enable_flag,signal_type,reprate_hzx10,nsecs,timing_mode*CS

enable_flag	**0** – Disable USBL mode, stop transmitting pings, if currently active. All other fields of the message are ignored. **1** – Enable USBL mode, start transmitting pings.
signal_type	**0 - FSK_probe -**The default probe for a FSK packet using current band settings**.** **1 - PSK_probe -**The default probe for a PSK packet using current band settings**.**
reprate_hzx10	The ping repetition rate in tens of Hz. Allowable values are 1, 2, 5, 10,20,40,50,100,200, corresponding to rates of 0.1, 0.2, 0.5, 1, 2, 4, 5, 10 and 20 Hz.
nsecs	The number of seconds to transmit, range is 0 to 3600. The number of pings transmitted is max(1,nsec*reprate_hzx10/10)
timing_mode	Set to zero. This is a reserved field for potential future use to convey information about PPS state.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Once the modem receives the $CCXSB command, it will echo it back (or issue NMEA error if need be). Acoustic communications packets can still be received in between ping transmissions, if they do not interfere with an outgoing transmission or reverberation from that transmission. However, the transmission of pings is paused during that time. The intent is to allow the vehicle that is homing to the dock the capability to turn off the pinging once docking is completed or aborted. In future, a User Mini Packet received at the transmitter could be used to abort the transmission.

CAXSB

Modem echo of the $CCXSB command.

$CAXSB,enable_flag,signal_type,reprate_hzx10,nsecs,timing_mode*CS

enable_flag	**0** – Disable USBL mode, stop transmitting pings, if currently active. All other fields of the message are ignored. **1** – Enable USBL mode, start transmitting pings.
signal_type	**0 - FSK_probe -**The default probe for a FSK packet using current band settings **1 - PSK_probe -**The default probe for a PSK packet using current band settings
reprate_hzx10	The ping repetition rate in tens of Hz. Allowable values are 1, 2, 5, 10,20,40,50,100,200, corresponding to rates of 0.1, 0.2, 0.5, 1, 2, 4, 5, 10 and 20 Hz.
nsecs	The number of seconds to transmit, range is 0 to 3600. The number of pings transmitted is max(1,nsec*reprate_hzx10/10)
timing_mode	Set to zero. This is a reserved field for potential future use to convey information about PPS state.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

CCRSB

Command modem to start listening to USBL pings at a specified repetition rate for nseconds.The modem switches the detector to the multichannel card for the USBL mode.

$CCRSB,enable_flag,signal_type,reprate_hzx10,nsecs,timing_mode*CS

enable_flag	**0** – Disable USBL mode, stop listening for USBL pings, if currently active. All other fields of the message are ignored. **1** – Enable USBL mode, start the USBL detector.
signal_type	The signal to listen for: **0 - FSK_probe -**The default probe for a FSK packet using current band settings, the modem will be able to receive PSK packets. **1 - PSK_probe -**The default probe for a PSK packet using current band settings, the modem will be able to receive FSK packets.
reprate_hzx10	The ping repetition rate in tens of Hz. Allowable values are 1, 2, 5, 10,20,40,50,100,200, corresponding to rates of 0.1, 0.2, 0.5, 1, 2, 4, 5, 10 and 20 Hz.
nsecs	The number of seconds to listen, range is 0 to 3600. The number of pings transmitted is max(1,nsec*reprate_hzx10/10)
timing_mode	Set to zero. This is a reserved field for potential future use to convey information about PPS state.
***CS**	Optional: The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

Once the modem receives the $CCRSB command, it will echo it back (or issue NMEA error if need be). If another $CCRSB command is issued before the first one times out, it will supersede the previous one.

The modem, while configured for receiving USBL navigational pings on one channel, will still be able to receive communication packets on the second channel. For example, if the `signal_type` is set to 0, the modem will still receive PSK packets, and if it is set to 1, it will still receive FSK packets.

CAUSB

The USBL navigational message.

$CAUSB, YYYY-MM-ddTHH:mm:ss.ssssZ,azimuth_deg,el_deg,owtt,timing_mode*CS

YYYY-MM-ddTHH:mm:ss.ssssZ	Time of arrival, in ISO 8601 Extended format.
azimuth_deg	Arrival angle, azimuth, degrees -180^0 to 180^0
el_deg	Arrival angle, elevation, degrees, -180^0 to 180^0
owtt	One way travel time from dock, if available, in ss.ssss. This is calculated assuming it is always less than the interval between successive pings. In future, more sophisticated signals can be sent to resolve the ambiguity for larger timing delays.
timing_mode	Timing mode on received ping, same as TOA mode
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

CAUIR

The USBL Peak Impulse Response message, all channels. Use in conjunction with channel_mask to pick out valid channels.

$CAUIR, ch1re,ch1im,ch2re,ch2im,ch3re,ch3im,ch4re,ch4im*CS

ch<1-4>re	Peak value of the impulse response estimate, real, q.15 format
ch<1-4>im	Peak value of the impulse response estimate, imaginary, q.15 format
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

CAUXY

The USBL unit vector. The unit vector assumes the USBL receiver to be the origin with the x-axis pointing towards the source.

$CAUXY,xloc,yloc,zloc*CS

xloc	Unit vector in x (pointing from the USBL receiver to the source). Q.15 format
yloc	Unit vector in y. Q.15 format
zloc	Unit vector in z. Q.15 format
***CS**	The NMEA sentence checksum, hex coded (8-bit XOR or CRC-32). See Message Checksums.

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.

ModemTMA

The Acoustic Communications Group provides a basic Windows GUI tool for interaction with Micromodems over a variety of connections. This tool allows basic functionality such as sending and receiving packets, setting configuration parameters, updating firmware, sending pings, etc. It also provides more advanced options for different users. Users can download the most recent version from:

http://acomms.whoi.edu/umodem/downloads.html

pyAcomms Support Library

The Acoustic Communications Group provides a basic interaction library for the Micromodem system through Python 2.7.x. This is a basic library which provides support for Micromodem functions. This section provides the installation instructions for the pyAcomms library. The library supports both windows, mac and linux development.

The current release of the pyAcomms library is located on the Python package Index (PyPI). The information links is here:

https://pypi.python.org/pypi/acomms

pyAcomms Installation Instructions:

Installation of Python:

Python can be installed following the instructions available from the python.org wiki here:

https://wiki.python.org/moin/BeginnersGuide/Download

Python 2 is the version that is supported and used by our libraries.

Installation of Python Package Index:

Python Package Index (PIP) is the tool which allows downloading of the current release of the pyAcomms repository. To install, follow the instructions from the provider here: http://pip.readthedocs.org/en/latest/installing.html

Installation of the pyAcomms Library and its dependencies:

Once the installation of python and the PIP tool are complete, then the installation of the library is as follows:

1) Open a command terminal.
2) Execute the following command:

```
pip install acomms
```

This will install all current dependencies for the pyAcomms library. These dependencies currently include:

- bitstring
- pyserial
- isodate
- timer2
- python-dateutil
- enum

- apscheduler
- setuptools
- crcmod

pyAcomms Upgrade Instructions

If an older version of pyAcomms is installed, one can upgrade using the same PIP tool. To upgrade:

1) Open a command terminal.
2) Execute the following command:

```
pip install --upgrade acomms
```

Applications

Long-term deployment with periodic wakeups and data logging

The Micromodem-2 may be deployed for long periods of time during which it monitors and logs data from sensors, and this data can be periodically retrieved by a host computer.

To increase mission duration, a host can command the modem to hibernate periodically to reduce energy consumption.

If a high-accuracy clock, such as an atomic clock, is available, the host may set the clock on the modem each time it wakes from hibernate so that timing-critical functions such as acoustic navigation can be performed accurately.

The following instructions deal with the following scenario:

- A host system has a high-accuracy clock, which it will use to set the time on the modem.
- The host will wake the modem from hibernate via the EXTWAKE pin.
- The host will command the modem to hibernate.
- The modem will log all NMEA messages.
- A host will retrieve logged NMEA messages at a later time.

In the following instructions, **`messages in bold`** are commands sent from the host to the modem and `messages not in bold` are messages sent from the modem to the host. *CS* is the NMEA message checksum.

Initial Configuration

First, the modem should be configured to enable NMEA logging. (The values shown below are also the default values, so they don't need to be set unless they were previously changed.)

$CCCFG,log.nmea.level,2

$CACFG,log.nmea.level,2**CS*

$CCCFG,log.nmea.location,1

$CACFG,log.nmea.location,1**CS*

Because the host will be commanding the modem to hibernate and waking it via the EXTWAKE pin, the automatic hibernate schedule should be disabled. (This, again, is the default for these configuration parameters.)

$CCCFG,hibernate.hibernate_after,0

$CACFG,hibernate.hibernate_after,0,2**CS*

$CCCFG,hibernate.wake_interval,0

$CACFG,hibernate.wake_interval,0**CS*

Wake the modem from hibernate

Assuming the modem has been previously commanded to hibernate, wake it by driving the EXTWAKE pin LOW.

Set the time when the host wakes the modem

EXTPPS is used to latch the time on the modem, and can be used once to set the time, or may be provided at a 1 second interval.

Adjust the time in this command as necessary. It is recommended that the host set the time at least 500ms after this command is issued, so the time specified here should be one or two seconds in the future.

$CCTMS,2012-12-01T01:04:05Z,1

After issuing the command, drive the EXTPPS pin high at the time specified in the message. The modem will reply with

$CATMS,0,2012-12-01T01:04:05Z**CS*

The host software should prepare to handle possible error conditions, as well. See the **CCTMS** command in the User's Guide.

Command the modem to hibernate (in the future)

After setting the time, the host software can command the modem to stay awake (and collect/log data) for 10 minutes and then hibernate.

Send the hibernate command with a relative hibernate time of 10 minutes (600 seconds), and set the wake mode to 0, because the modem should hibernate until the host drives the EXTWAKE pin.

$CCHIB,600,0

$CAHIB,0,*2012-12-01T01:15:21Z*,0,**CS*

The timestamp in the response will correspond to the current time + 10 minutes. After 10 minutes, the modem will issue a $CAHBR message before shutting down.

$CAHBR,**CS*

See the **CCHIB** command in the User's Guide for more details.

Retrieving Logged Data

When the host wishes to retrieve the data that was logged by the modem while it was operating, the host software can issue the **CCRBR** command.

To request all logged messages that haven't already been retrieved:

```
$CCRBR,1,0,0,0,0
```

The modem will then respond with a series of **CARBR** messages containing logged data. The **CARBR** message syntax, along with a number of examples, is described in the NMEA Data Logging and Retrieval section of the User's Guide.

Example Real-Time Modem Output

A wav file is available to use in testing the modem and the host application software. It may be downloaded from http://acomms.whoi.edu/documents/MIZ_rate1_bw25_fc_900_fs_48000.wav. The receiving modem needs the following parameters set:

```
$CCCFG,BND,0

$CCCFG,BW0,25

$CCCFG,FC0,900

$CCCFG,psk.packet.mod_hdr_version,1

$CCCFG,psk.packet.nulltime_ms,4000

$CCCFG,recv.agc_length_ms,500
```

When played out of a sound card on a computer into the analog input of the modem using a sampling rate of 48 kHz, it will result in the following series of messages similar to below:

```
$CAREV,181916,AUV,2.0.14703*18

$CAREV,181916,COPROC,0.20.0.51*43

$CARXP,1*45

$CARDP,0,1,1,0,0,1;8;0001020304050607;,*6A

$CACST,6,0,20131008181917.655045,3,1539,25,0384,0150,141,01,01,01,01,1,000,001,0,5,1,0,150,49.2,26.4,-100,-23.4,-01,1.1,-6,900,0025*60

$CAREV,182031,AUV,2.0.14703*17
```

```
$CAREV,182031,COPROC,0.20.0.51*4C
```

To retrieve the messages, send the following command to the modem:

```
$CCRBR,1,0,0,0,0
```

The results provided back will be similar to below:

```
$CARBR,0,0,2013-08-23T19:14:26Z,15,CAREV,191426,AUV,2.0.10734*71

$CARBR,0,1,2013-08-23T19:14:26Z,15,CAREV,191426,COPROC,0.20.0.51*2B

$CARBR,0,2,2013-08-23T19:14:26Z,15,CAREV,191426,RECOVERY,2.0.10734*32

$CARBR,0,3,2013-08-23T19:14:33Z,15,CARXP,1*24

$CARBR,0,4,2013-08-23T19:14:35Z,15,CARDP,1,0,1,0,0,1;9;000102030405060708;,*03

$CARBR,0,5,2013-08-
23T19:14:35Z,15,CACST,6,0,20130823191433.543886,3,836,29,0361,0150,40,00,00,00,00,-1,-01,-
01,0,5,1,0,150,69.4,33.0,-100,-30.0,-01,0.0,00,10000,2000*34

$CARBR,0,6,2013-08-23T19:14:35Z,15,CAREV,191435,AUV,2.0.10734*77

$CARBR,0,7,2013-08-23T19:14:35Z,15,CAREV,191435,COPROC,0.20.0.51*2D

$CARBR,0,8,2013-08-23T19:14:35Z,15,CAREV,191435,RECOVERY,2.0.10734*38

$CARBR,0,9,2013-08-23T19:14:40Z,2,CCRBR,1,0,0,0,0*06

$CARBR,1,0,,,*6D
```

Docking an AUV using USBL navigation with precision timing

In this example, a vehicle equipped with a precision timing source (CSAC board synchronized to GPS) docks to a docking station, also synchronized to a GPS source. The CSAC board supplies timing information as $GPZDA messages in conjunction with a PPS signal which the vehicle modem can use to set its clock.

Assuming that the GPZDA messages are coming in over UART4 and are being echoed on to UART1 of the micromodem, the following commands are issued to the vehicle modem before the start of the mission:

1. Power on vehicle
2. Power on GPS unit
3. Wait a few minutes for the CSAC board to boot. When the board boots, the modem prints out its boot message on UART1:

```
$CAPST,2,0,0,0,,CSAC($Rev: 16967 $)*00
```

4. Set the uart configuration parameters to process GPS messages on the appropriate UART and echo them on another. Set the modem clock using GPS message string on the UART that is processing the incoming messages. In this example, the CSAC board is connected to UART4 of the modem using a baudrate of 19200, and the modem output is monitored on UART2.
   The configuration parameters need to be set only once and hold through a modem power cycle.

```
$CCCFQ,uart4

$CACFG,uart4.parse_gps,1*37

$CACFG,uart4.show_gps,0*40

$CACFG,uart4.set_clk_GPS,1*3B

$CACFG,uart4.crc32,0*0B

$CACFG,uart4.rs485,0*40

$CACFG,uart4.bitrate,19200*0F

$CCCFQ,uart2

$CACFG,uart2.parse_gps,0*30

$CACFG,uart2.show_gps,1*47

$CACFG,uart2.set_clk_GPS,0*3C

$CACFG,uart2.crc32,0*0D
```

```
$CACFG,uart2.bitrate,19200*0
```

5. Once the modem sees a valid GPS string come through along with an external PPS, it will set the clock. Check to make sure that the clock source is GPS and and the PPS source is EXT. The modem will print a CATMG message upon change of clock status, or the user can query status using the CCTMQ command. Keep in mind that the CSAC board only processes valid fixes and the GPS unit can take several minutes to acquire one.

```
$GPRMC,194010,A,4131.4776,N,07040.2588,W,000.0,000.0,051214,015.5,W*78

$CATMG,2014-12-05T19:40:11Z,GPS,EXT*7E

$CCTMQ,1

$CATMQ,2014-12-05T19:40:46Z,GPS,EXT*6A
```

6. At this point, it is safe to disconnect the GPS unit and start the vehicle mission. The CSAC board will maintain the clock state through a power cycle up to 2 minutes. If the vehicle reboots, the modem will automatically re-synchronize its clock from the CSAC board. If the vehicle power is turned off for longer than this time, the modem will have to be re-synchronized.
7. Once the vehicle is ready to home, it can initiate homing by sending the $CCXSB command to the docking station acoustically using the $CCACM command. A coprocessor firmware of 0.10.0.52 or greater is needed to send the acoustic command. In the following example, a vehicle with source ID 5, commands a docking station with source ID 6, to start pinging at a repetition rate of 1 Hz for 5 seconds. Bold face commands are serial input to the modem and normal face are serial output from the modem.

```
$CCACM,6,3,1,CCXSB,1,1,10,5,0

$CAACM,6,3,1,CCXSB,1,1,10,5,0*18

$CATXP,0*42

$CATXF,0*54

$CAXST,6,20140529,195047.278352,3,0,200,5000,25000,2,0,0,0,0,2,3,0*4B
```

The dock side, upon reception of the acoustic command packet, reports CAACA, sends the acknowledgement and then starts the pinger:

```
$CAACA,6,3,1,CCXSB,1,1,10,5,0*14

$CACST,6,0,20140529195037.609747,3,11385,40,0129,0150,100,00,00,00,00,3,000,006,0,5,2,0,150,63.7,
22.9,-100,-19.9,-01,0.0,26,25000,5000*44

$CATXP,0*42
```

```
$CATXF,0*54

$CAXST,6,20140529,195038.282801,3,0,200,5000,25000,2,0,0,0,0,3,3,0*4A

$CATXP,200*40

$CATXF,200*56

etc. …
```

Upon receipt of the acknowledgement the vehicle side reports:

```
$CAACR,6,3,0,CCXSB,1,1,10,5,0*06

$CACST,6,0,20140529195047.951611,3,8238,39,0093,0150,100,00,00,00,00,3,006,006,0,5,3,0,150,61.1,27.8,-100,-24.8,-01,0.0,27,25000,5000*7B
```

8. In the following example, the vehicle will listen for 1 hour for homing pings and calculate range based on a repetition rate of 4 Hz.

```
$CCRSB,1,1,40,3600,0

$CARSB,1,1,40,3600,3*5F
```

9. On the dock modem, the homing pings can start upon issuing the $CCXSB command. In this example, the dock will issue homing pings for one hour at a repetition rate of 4 Hz.

```
$CCXSB,1,1,40,3600,0

$CAXSB,1,1,40,3600,3*55

$CATXP,200*40

$CATXF,200*56
```

10. The vehicle should start printing $CAUSB strings upon successful reception of the homing pings. At any time in the homing processes, the vehicle or the docking station can stop the homing by re-issuing the appropriate USB command with the enable field set to 0. The modem also echoes a $CARSB,0… sentence upon expiration of the USBL receiver listen time.

```
$CAUSB,2014-12-05T19:48:03.000196Z,-4.6,4.5,00.0001,3*70

$CAREV,194803,AUV,2.0.16992*1D

$CAREV,194803,COPROC,0.10.0.46*47

$CAUSB,2014-12-05T19:48:03.250196Z,-4.6,4.5,00.0001,3*77

$CARSB,0,1,40,10,0*59
```

Navigation using a tracking ping

The modem can send out tracking pings on a schedule based on CCCFQ,nav- Query navigation configuration parameters settings. Once a ping mode is set, it holds through a modem reset/reboot cycle. It is advisable to set and synchronize the modem clock before starting the tracking pings. Configuration parameters are updated only during a mode change, i.e. once a pinger has been turned on, configuration setting changes will not take effect until the pinger is turned off. There are four distinct modes for sending out tracking pings.

nav.trackping.mode

0	Default, pinger inactive
1	User sequence, send out a user-defined sequence
2	Send a tone
3	Send an FM upsweep
4	Send a tone at the set period with a double tone every 10 seconds

To send a user-defined sequence, the user would use mode 1 and set the following additional configuration parameters:

nav.trackping.data packed data in a 64 bit unsigned int

0-2^{64}	Packed data bits

nav.trackping.sequence_len The number of bits to unpack and send

1-64	Length of sequence in bits

nav.trackping.cycles_per_sym this parameter can be used in lieu of bandwidth to set number of carrier cycles per symbol

1-1000	Cycles per symbol

For all other pings, the above parameters are ignored.

To send a 13.5 kHz, 3.5 ms tone at the top of every second with a second tone 35 ms later at the top of every 10th second, starting from the top of the minute, set the following parameters

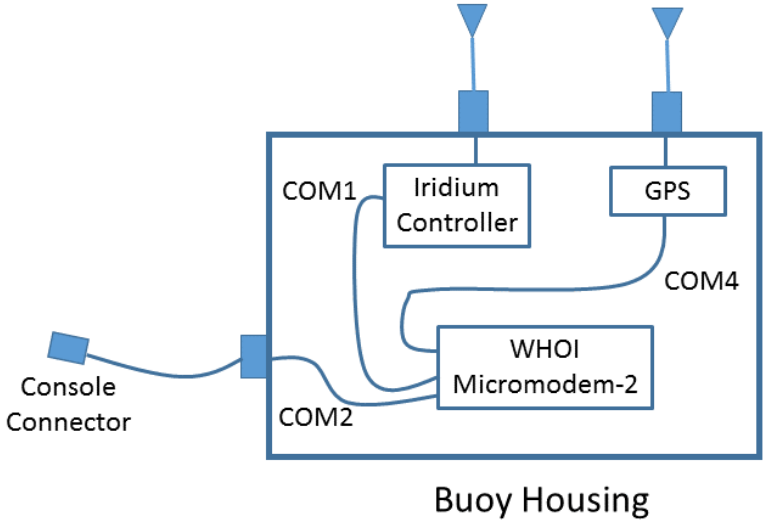
```
$CCCFG,nav.trackping.carrier_Hz,13500

$CCCFG,nav.trackping.period,1

$CCCFG,nav.trackping.reference_time,1449187200

$CCCFG.nav.trackping.duration_msx10,35

$CCCFG,nav.trackping.mode,4
```

To send a 10.5 kHz, 10 ms FM upsweep at the top of every 10th second starting from the top of the minute, set the following parameters

```
$CCCFG,nav.trackping.carrier_Hz,10500

$CCCFG,nav.trackping.period,10

$CCCFG,nav.trackping.reference_time,1449187200

$CCCFG.nav.trackping.duration_msx10,100

$CCCFG.nav.trackping.bandwidth_Hz,3000

$CCCFG,nav.trackping.mode,3
```

CCPST Passthrough Message Examples

The examples below reference the system drawn here:



In these examples, a Micromodem-2 is in a buoy housing with a console connector on COM2, an Iridium controller (such as WHOI-205102 Iridium 9523 Interface Board) on COM1, and a GPS on COM4. The Iridium controller in these examples relays all NMEA sentences that it receives back and forth between COM1 and the Iridium RF channel (like a "bridge", or a high-latency NMEA console). In addition, the Iridium controller board's microcontroller can also send sentences onto COM1 to the Micromodem-2.

Example 1: Iridium controller sends a sign-on message to the Micromodem-2 on COM1 to be passed through to the Console on COM2 as well as echoed back to itself (in turn sent over Iridium to shore, verifying on shore and at the console that the Micromodem-2 is functioning):

> Iridium Controller sends to Micromodem-2 on COM1: **$CCPST,3,0,0,0,0,,Iridium online\r\n**
> Micromodem-2 passes through on COM1 and COM2: **$CAPST,3,0,0,0,0,,Iridium online*CS\r\n**
> [In this example, append_CRLF field is included in $CAPST message]

Example 2: Iridium controller sends a debug message to the Micromodem-2 on COM1 to be passed through to the Console on COM2, but NOT echoed back to itself (so that it will not be sent over Iridium):

> Iridium Controller sends to Micromodem-2 on COM1: **$CCPST,2,0,0,0,0,,Debug\r\n**
> Micromodem-2 passes through message on COM2 only: **$CAPST,2,0,0,0,0,,Debug*CS\r\n**
> [In this example, append_CRLF field is included in $CAPST message]

Example 3: Host computer on COM2 console connector sends a configuration message to a u-blox GPS connected on COM4, also echoing back to itself:

Host computer sends to Micromodem-2 on COM2:

$CCPST,10,1,1,0,0,,PUBX,40,GLL,0,0,0,0,0,0,0\r\n

Micromodem-2 passes through message on COM4 and COM2:

$PUBX,40,GLL,0,0,0,0,0,0,0\r\n [disable $GPGLL on u-blox GPS]

Example 4: As with Example 3 above, but a shore-side computer sends the $CCPST as an Iridium message that is sent to Micromodem-2 on COM1, and is echoed to console on COM2, Iridium controller on COM1, and u-blox GPS on COM4:

Shore-side computer sends via Iridium to Micromodem-2 on COM1:

$CCPST,11,1,1,0,0,,PUBX,40,GLL,0,0,0,0,0,0,0\r\n

Micromodem-2 passes through message on COM4, COM2, and COM1:

$PUBX,40,GLL,0,0,0,0,0,0,0\r\n [disable $GPGLL on u-blox GPS]

FATHOMETER Examples

Depth sounder, no Iridium link

To use the modem as a depth sounder with local logging of serial data (no Iridium link), a full depth capture window of 4 seconds, transmit lockout of 100 ms, and base64 data for the **$CACIR** messages, use the following configuration parameter settings, other settings are ignored for this case:

$CACFG,fathometer.active,0*4D

$CACFG,fathometer.hibernate_after,0*10

$CACFG,fathometer.t0_ms,100*45

$CACFG,fathometer.tcapture_ms,4000*24

$CACFG,fathometer.report_full_window,1*51

$CACFG,fathometer.decimate_by_2,0*7E

$CACFG,fathometer.npings_total,1*70

$CACFG,fathometer.npings_over_iridium,0*25

$CACFG,fathometer.min_dx_m,0*5A

$CACFG,fathometer.diagnostics_over_iridium,0*5C

$CACFG,fathometer.ire_over_iridium,0*56

$CACFG,fathometer.use_base64,1*6B

When ready to ping, set "**fathometer.active**" to 1

$CCCFG,fathometer.active,1

The modem will send out an FM sweep using the settings defined for a PSK packet probe. After the lockout period (**t0_ms**), it will open the detector and record the impulse response for **tcapture_ms** milliseconds. At the end of that time, it will issue a report using $CACFT and $CACIR messaging. All messages are echoed onto uart1 and uart2. If there is a GPS feed , the last reported latitude and longitude will be reported in the $CACFT message.

$CACFG,fathometer.active,1*4C

$CATXP,2000*70

$CATXF,2000*66

$CAXST,6,20180522,184520.000040,3,0,200,1000,3500,-2,17229,16708,19795,0,0,-1,0*74

$CACFT,2018-05-22T18:45:20.000040Z,41.5244,-70.6711,250,2.0009,32767,0*53

$CACIR,1,7,2018-05-22T18:45:20.525426Z,100,12,18,AAAAAAEAAAABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQABAAEAAgABAAEAAgACAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAAAAAAAAAAAAAAAAEAAAABAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAEAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAQAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQACAAEAAQABAAEAAQACAAIAAQAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAQAAAAEAAgABAAEAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAACAAMABAAEAAIAAgABAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAEAAQABAAEAAQAAAAAA,18,AAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQAAAAEAAQABAAAAAAACAAEAAgACAAIA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAA,18,AAABAAEAAQABAAEAAgABAAEAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAABAAIA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAEAAAAAAAAAAAACAAMABAADAAIA,18,AQABAAEAAgADAAMABAAEAAUABQAEAAMA,18,AQAAAAAAAAAAAAAAAAAAAAAAAAAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAABAAIAAgACAAIAAQABAAIAAgACAAEA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAA,18,AAABAAMAAwACAAAAAgACAAMAAQAAAAEA,18,AQAAAAAAAQACAAMAAgACAAEAAAAAAAIA,18,AgACAAIAAQABAAAAAAAAAAAAAAAAAA,18,AAABAAEAAgADAAUABwAHAAgABQAEAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAEAAQAAAAAAAAABAAMAAgAAAAAA,18,AgADAAYACAAFAAUABwAHAAQAAgACAAIA,18,AQABAAEAAQAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAEABAADAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAQABAAAAAgABAAAA,18,AAAAAAAAABAAIABAAGAAQAAwACAAEA,18,AgABAAEAAAAAAAAAAAAAAAAAAAAAAA,18,AQAAAAAAAAABAAAAAAAAAAAAAAAA,18,AAADAAQABAADAAIAAwAEAAIAAAAAAAIA,16,AAAAAAAAAAAAAAAAAABAAEAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,4,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,6,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQABAAEAAQAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAA

AAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA,4,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,6,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
*56

$CACIR,2,7,2018-05-
22T18:45:20.525426Z,100,12,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAABAAA
A,18,AAAAAAAAAAAAAAAAQAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAEAAQAAAAAA,18,AAAAAAAAAA
AAAAAAAAAAAAAAAABAAEA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAA
AAAABAAAAAAAAAAA,16,AAAAAAAAAABAAAAAQAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAABAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAAAAAAAAAAAAAEAAgABAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAA
AA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18
,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAABAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAB
AAEAAQACAAEAAQABAAIAAwADAAIA,18,AgACAAIAAQABAAEAAQABAAEAAQACAAIA,16,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAQABAAEAAAA
AAAAAAAAAAEA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAEA
AQABAAEA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
8,AAABAAEAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAEAAQABAAAAAAAAAAA,16,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,18,AAAAAAAAQABAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAA
AAAA,18,AAAAAAAAAAAAAAAAQABAAIAAgABAAEA,18,AgACAAIAAgACAAEAAAAAAAAAAAAAAA,
18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAA
AAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAA
AAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAEA,18,AQAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AQAAAAAAAAAAAAAAAAAAAAAAAAAA
,18,AAAAAAAAAAAAAAAAAAAAAAAQAAAAEA,18,AQACAAEAAAAAAAAAAAABAAEAAAAAAAA,18,A
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQA
BAAEAAAAAAAAAAAAAAAA,18,AAAAAAAAAABAAEAAQABAAEAAQAAAAA,16,AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAA
AEAAQABAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAEA,18,AAAAAAMABAACAAMABAABAAAAAAAAA
A,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,
AAAAAAAAAAAAAAAAAAAAAAAAQABAAAA,18,AAAAAAAAAAAAAAAAAABAAEAAgACAAEA,16,AAABA
AAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAA

ABAAEAAQABAAAAAAAAAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAEAAgACAAIAAQAAAAAA,18,AAAAAAAAAAAAAAAAAAB
AAAAAAAAAAAA,18,AAAAAAAAAQAAAAEAAAAAAAAAAQABAAAA,18,AAAAAAAAAQAAAAAAAAAAAEAA
gACAAEA*5A

$CACIR,3,7,2018-05-
22T18:45:20.525426Z,100,12,18,AQAAAAAAAAAAAAAAAAAAAAAAgADAAEA,16,AAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,18,AQABAAAAAAABAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAEA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAA
AAAAAAAAABAAEAAQABAAEA,18,AAABAAEAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAgADAAUA,18,BAADAAIAAQAAAAAAQAD
AAMAAwADAAIA,18,AgAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAQABAAAAA
AAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAQABAAEA,18,AQABAAAAAAAAAAAAAAAAAEAAwAFAA
YA,18,BAACAAEAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18
,AQABAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAABAAEAA
AABAAEAAQAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAEAAAAAAAAA,18,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,18,AAAAAAEAAAAAAAAAAAAAAAEAAAAAAAAA,16,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQAAAAAAAAAAAAA
AAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQABAAEAAAAAAAAAAAAAAAAAAAAAAA,18,AAA
AAAEAAQABAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAA
AAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAA
AAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAA
AAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAA
AAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,0,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,0,AAAAAAAAAQAAAAAAA
AAAAAEAAAAAAAAA,0,AQAAAAAAAQABAAAAAAABAAAAAAAAAAEA,0,AAAAAAEAAQAAAAAAgAAAAAA
AgAAAAAA,0,AgABAAAAAgACAAAAAgACAAAAAgADAAAA,0,AgADAAAAAwACAAAABAACAAAABQAAAAI
A,0,BQAAAAUAAwAAAAcAAAAEAAUAAAAJAAEA,0,BAAHAAAACwABAAcACAACAA8AAAAPAAQA,0,CwA
OAAMAGAAAAB4AAgAcAA0AFwAeAA8A,0,NgAGAFEAAAB0AAAAqAAIAAQBIgDUAW4A,0,nwTUAcMf/3
9KabYFFwZHAdEBogDJAHIA,2,XwBcACwAUQAQAEcAAgA9AAAALgADAB4A,4,CwAPABMABAAXAAAAE
wACAAwACQADAA4A,6,AAAMAAMABQAIAAAACgABAAUABgAAAAcA,8,AQACAAYAAAAFAAMAAAAGAAAA
AQAEAAAA,10,AwADAAAAAwABAAAAAwABAAAAAwABAAAA,12,AwABAAAAAwABAAAAAQABAAAAAQABA
AAA,14,AAABAAAAAAABAAEAAAAAAAEAAAAAAAAA,16,AQAAAAAAAQABAAAAAAABAAEAAAAAAAEA,1
8,AQAAAAAAAAAAAAAAAAAAAAEAAQAAAAAA,18,AAAAAAEAAAAAAAAEAAAAAAAAAAAAAAAEA,16,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAA

AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA*78

$CACIR,4,7,2018-05-
22T18:45:20.525426Z,100,12,18,AQAAAAEAAQABAAEAAQABAAEAAQABAAEA,18,AQABAAAAAAA
AAAAAAQABAAEAAQACAAEA,18,AQABAAEAAgACAAIAAgACAAMAAwACAAIA,18,AQABAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAEAAgACAAEAAQABAAEAAQABAAEA,16,AAAAAAAAAAABAAEAAQAAA
AAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAACAAIAAwADAAMA,18,AwAEAAMAAwACAAEAAAAAAAAAAA
AAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAABAAAAAAAAAAAAAAAAAA
A,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAA
AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AQAAAAAAAAAAAAAAAAAAAEAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAA,16,AAAAAAAAAAAAAAEAAAABAAEAAQAAAAAA,16,AAAAAAAAAQACAAEAAAAAAAAAAAAAAA
AA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAgADAAMAAQABAAEAAAAAAAEA,16
,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAABAAEAAAAAAAAAAAAAAAAA,18,AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAEAAAAAAAAA,18,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAABAAEAAAAAAAAAAAAAAAAAAAAA
AAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
6,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAA,18,AAAAAAEAAQABAAEAAQABAAEAAQAAAAEA,18,AQAAAAEAAQABAAEAAQAAAAEAAQAB
AAEA,18,AQABAAIAAwAEAAMAAgAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,
18,AAAAAAAAAAAAAAAAAACAAIAAQACAAIA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAABAAEAAAAAAAAAAAAAAAEA,18,AQABAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAEAAQABAAEAAgAFAAkA,18,CgAHAAQAAQAA
AAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQAEAAMAAwABAAAAAQAAAAAA,16,AAAAAAAAAAAAAAAA
AAAAAEAAQAAAAAA,16,AAAAAAAAAAAAAAAAAQABAAEAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAA
AAAQAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQABAAAAAAAAAAAAAA
BAAEA,18,AQAAAAAAAAAAAAAAAAAAAAAQABAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
,18,AQABAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQABAAEAAQABAAEAAQAAAAAAAAAAAA,18,A
QAAAAAAAAAAAEAAQADAAIAAQAAAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAA
AAAAAAAAEAAQAAAAEAAQAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAA
AAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,18,AAAAAAAAAAAAAAAAAAAAAAEAAQABAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAA
AAAAAAAAAAAQACAAMAAwADAAIA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAEAAQABAAAAAAAAAAAA,18,AAAAAAAAAAAAAA

AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAQACAAIAAgACAAIA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AQAAAAAAAQABAAAAAAAAAAAA
AAAAAAA*75

$CACIR,5,7,2018-05-
22T18:45:20.525426Z,100,12,18,AQAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAQA
BAAAAAQACAAIAAgACAAMA,18,AgAEAAQABAADAAIAAAABAAIABAAGAAUA,18,BAACAAAAAAAAAAAA
AAADAAUABwAHAAgA,18,BQACAAEAAQABAAIAAQABAAEAAAAAAAAA,16,AAAAAAAAAAABAAAAAAAAA
AAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAABAAAAAQABAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAABAAEAAAAAAAAAAABAAEAAQAAAAA
A,18,AAAAAAEAAQAAAAAAAAAAAEAAAABAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,4,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,6,AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,
18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA,4,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,6,AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18
,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
6,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAA
AAAAAAAAAAAAAAAAAAAAAQABAAEA,18,AgACAAMAAwABAAIAAgACAAEAAAABAAEA,16,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAABAAEA,18,AQAAAAAAAAAAAAAAAABAAAAAAAA
AAEA,18,AQABAAEAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,
18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAEAAAAAAAAAAAAAAA,18,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAIAAQAAAAAAAAAAAEAAQABAAEA,18,AQAAAAE
AAAAAAAAAAAAAAEAAQADAAMA,18,AwADAAQABQAEAAMAAQABAAAAAAAAAAAA,18,AAABAAEAAQAB
AAEAAwADAAIAAQADAAQA,18,AwABAAAAAAACAAIAAgABAAEAAAAAAAAA,18,AAAAAAAAAABAAQAC
AALAAoABwADAAAA,18,AAAAAAAAAQAFAAgABwAEAAQABQAGAAMA,18,AAAAAAEABAAFAAIAAAAAAA
AAAAAAAAEA,18,AQADAAQAAwAAAAAAAAAAAIAAgACAAIA,18,BgAKAAgAAwAAAAIABgAEAAIAAAA
BAAQA,16,AAAAAAAAAgAEAAQAAgAAAAEAAQAAAAAA,18,AAAFAAUAAQAAAAMABQADAAEAAAAAAAAA
,18,AAAAAAEAAwADAAEAAAAAAAAAAAQACAAAA,16,AAAAAAAAAAAAAAAAAAAAAAEAAQAAAAAA,8,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,12,AAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAA
AAAAAAAAQABAAEA,18,AQABAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAA

AAAAAAAAAA,18,AQABAAEAAQABAAEAAAAAAAAAAAAAAAEA,16,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*41

$CACIR,6,7,2018-05-
22T18:45:20.525426Z,100,12,18,AAAAAAAAAAAAAAAAAAAAAAEAAQACAAIA,18,AQABAAEAAQA
BAAEAAwADAAQABAACAAAA,18,AAAAAAAAAQABAAEAAQAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAA,18,AAAAAAAAAAAAAAAAQAAAAAAAQAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAA
AAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAA
AA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18
,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQABAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAABAAEA,18,AQABAAAAAAAAAAAAAA
AAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAEAAQAAAAAAAAAAAAAAAAAAAA,16,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAA
AAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,20,AAAAAAAAAAAAAAAAQABAAEAAgACAAEA,18,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,18,AAABAAAAAAAAAEAAQABAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,
18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AA
AAAAAAAAAAAAAAAAAAAAAAAAAAEA,18,AAAAAAAAAAAAAAEAAQABAAAAAAABAAEA,18,AQABAAE
AAQAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAABAAIAA
gABAAEAAgABAAAA,18,AAAAAAAAAQABAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAA
AAAQACAAIA,18,AQABAAAAAAAAAAAAAABAAEAAAAAAAAA,18,AAABAAIAAwADAAIAAQAAAAAAAA
AAAAA,18,AAAAAAAAAAAAAAAAAAAAAEAAAAAAAAA,18,AAABAAIABQAIAAgACAAHAAcACAAIAAUA
,18,BAAGAAgABgAEAAQABwAGAAMAAgACAAIA,18,AQABAAAAAAABAAIABAAFAAMAAgAAAAAA,18,A
AAAAAEAAQAAAAAAAQAEAAYABwAKAAwA,16,AwADAAIAAQAAAAAAAAAAAAAAAAAAAA,18,AAAAAA
AAAAABAAEAAQAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,4,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,6,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,8,AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,12,AAAAAAAAAAAAAAAAAAAAAAAAAA
AAA,14,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,1
8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAA

AAAAAAAAAAAAAA,4,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,6,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAA,8,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,10,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
A*48

$CACIR,7,7,2018-05-
22T18:45:20.525426Z,66,12,12,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,14,AAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAA
AAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAA
AAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,A
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAA
AAAAAAAAAAAAAAAAAAQABAAAA,18,AAAAAAAAAAAAAAAAQACAAIAAgABAAAA,18,AAAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAEA,16,AAAAAAAAAAAAAAAAAAA
AAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAQAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
A,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAEAAQABAAAA,16,AAAAAAAAAA
AAAAEAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAABAAAAAQAB
AAEAAQACAAEA,18,AAAAAAAAAAAAAAAAAAAAEAAQAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAEA,18,AQABAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAQABAAIAAgACAAMA,18
,AgADAAIAAgACAAIAAQABAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAA
AAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAQAAAAAAAAAAAAAAQAAAAAA,16,AAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAABAAEAAQABAAAAAQA
BAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAEA,18,AgABAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AQAAAAAAAAAAAAAAAAAAAAAAABA
AAA,18,AAABAAIABAAFAAYABgAFAAMAAQAAAAAA,18,AQACAAEAAQABAAEAAAAAAAAAAAAAAAAA,1
8,AAAAAAAAAAAAAEAAQABAAAAAAABAAMA,16,AQABAAEAAAAAAAAAAAAAAAAAAAAAAA,18,AAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA,18,AgADAAQAAwADAAMAAwABAAAAAAAAAEA,16,AAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA,16,AAAAAAAAAAAAA
AAAAAAAAAAAAAAAAEA,16,AgACAAMAAwACAAEAAQABAAEAAQAAAAAA,16,AAAAAAAAAAAAAAAAA
AAAAAAAAAAAAA*6C

References:

[1] "Micro-Modem Software Interface Guide," Acoustic Communications Group.

Document Revision History:

10 Aug 2017: Merged Micromodem-1 manual with Micromodem-2 manual.