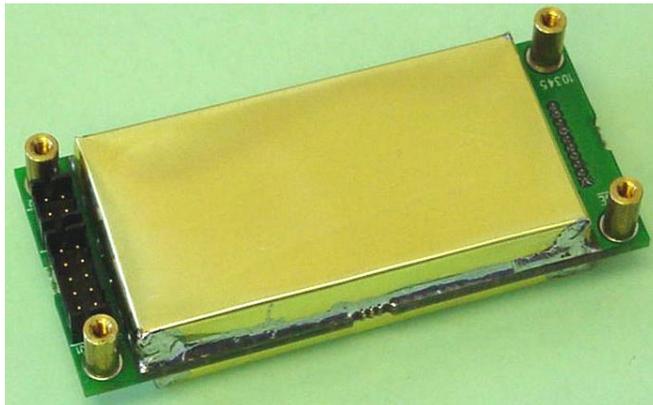


Inductive Modem Module (IMM)



Technical Reference Manual

Sea-Bird Electronics, Inc.
13431 NE 20th Street
Bellevue, Washington 98005 USA
Telephone: +1 425-643-9866
Fax: +1 425-643-9954
E-mail: seabird@seabird.com
Website: www.seabird.com

Manual Version #013, 3/22/21
Firmware Version 1.14 and later



Limited Liability Statement

Extreme care should be exercised when using or servicing this equipment. It should be used or serviced only by personnel with knowledge of and training in the use and maintenance of oceanographic electronic equipment.

SEA-BIRD ELECTRONICS, INC. disclaims all product liability risks arising from the use or servicing of this system. SEA-BIRD ELECTRONICS, INC. has no way of controlling the use of this equipment or of choosing the personnel to operate it, and therefore cannot take steps to comply with laws pertaining to product liability, including laws which impose a duty to warn the user of any dangers involved in operating this equipment. Therefore, acceptance of this system by the customer shall be conclusively deemed to include a covenant by the customer to defend, indemnify, and hold SEA-BIRD ELECTRONICS, INC. harmless from all product liability claims arising from the use or servicing of this system.

Declaration of Conformity

Sea-Bird Electronics, Inc.
13431 NE 20th Street, Bellevue, WA 98005 USA

DECLARATION OF CONFORMITY

Manufacturer's Name: Sea-Bird Electronics
Manufacturer's Address: 13431 NE 20th Street
Bellevue, WA 98005, USA

The Authorized Representative located within the Community is:
OTT MESSTECHNIK GmbH & Co.KG
P.O.Box: 2140 / 87411 Kempten / Germany
Ludwigstrasse 16 / 87437 Kempten
Internet: <http://www.ott.com>
Phone: +49 831 5617 – 100
Fax: +49 831 5617 - 209

Device Description: Various Data Acquisition Devices and Sensors

Model Numbers:

| | | | | | | | | |
|--------------------|-------|---------------------------------------|--------|-----------|--------------|--------|-----------|--------|
| 3S | 3F | 3plus | 4C | 4M | 5T | 5P | 5M | 7 |
| 8 | 9plus | 11plus | 14 | 16plus V2 | 16plus-IM V2 | | 17plus V2 | 18 |
| 19plus V2 | 21 | 25plus | 26plus | 27 | 29 | 32 | 32C | 32SC |
| 33 | 35 | 35RT | 36 | 37-IMP | 37-IM | 37-SMP | 37-SM | 37-SIP |
| 37-SI | 38 | 39 | 39-IM | 41 | 41CP | 43 | 43F | 44 |
| 45 | 49 | 50 | 52-MP | 53BPR | 54 | 55 | 56 | 63 |
| SIM | ICC | IMM | PDIM | AFM | 90488 | 90204 | 90402 | 90504 |
| Glider Payload CTD | | NiMH Battery Charger and Battery Pack | | | | | | |

Applicable EU Directives: Machinery Directive 98 / 37 /EC
EMC Directive 2004 / 108 /EC
Low Voltage Directive (73 / 23 /EEC) as amended by (93 / 68 /EEC)

Applicable Harmonized Standards:
EN 61326-1:2006 Class A Electrical Equipment for Measurement, Control, and Laboratory Use, EMC Requirement – Part 1: General Requirements
(EN 55011:2007 Group 1, Class A)

EN 61010-1:2001, Safety Requirements for Electrical Equipments for Measurement, Control, and Laboratory Use – Part 1: General Requirements

Declaration based upon compliance to the Essential Requirements and Letter of Opinion from CKC Certification Services, LLC., Notified Body 0976

I, the undersigned, hereby declare that the equipment specified above conforms to the above European Union Directives and Standards.

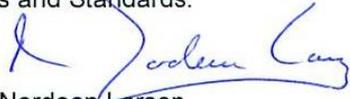
Authorized Signature: 
Name: Nordeen Larson
Title of Signatory: President
Date: 27 June 2012
Place: Bellevue, WA

Table of Contents

| | |
|--|-----------|
| Inductive Modem Module (IMM) | 1 |
| Limited Liability Statement | 3 |
| Declaration of Conformity | 4 |
| Table of Contents | 5 |
| 1 IMM Overview and Conceptual Model | 8 |
| 1.1 Scope and Intent of IMM Conceptual Model | 8 |
| 2 IMM Conceptual Model | 10 |
| 2.1 Hardware..... | 10 |
| 2.1.1 Host..... | 10 |
| 2.1.2 DPSK Transceiver with Tone Detect | 10 |
| 2.1.3 Signal Detector | 10 |
| 2.1.4 Inductive Cable Coupler..... | 10 |
| 2.1.5 Power Input | 11 |
| 2.1.6 HostFlag Digital Input..... | 11 |
| 2.1.7 IMFlag Digital Output..... | 11 |
| 2.2 Nonvolatile Structures | 11 |
| 2.2.1 Hardware Data..... | 11 |
| 2.2.2 Configuration Data..... | 11 |
| 2.2.3 Status Data..... | 12 |
| 2.2.4 Event Counters..... | 12 |
| 2.2.5 Host Data File..... | 12 |
| 2.2.6 Sample Data..... | 12 |
| 2.2.7 GData Reply Buffer | 12 |
| 2.2.8 IM Reply Buffer..... | 12 |
| 2.3 Volatile Structures | 13 |
| 2.3.1 Operating Mode | 13 |
| 2.3.2 Host Command Buffer | 13 |
| 2.3.3 IM Command Buffer..... | 13 |
| 3 Operating Modes | 14 |
| 3.1 Host Service | 14 |
| 3.2 IM Service..... | 14 |
| 3.3 IM Receive | 14 |
| 3.4 Sleep | 14 |
| 3.5 Power Cycling IMM..... | 15 |
| 4 Reply Tags and XML-like Formatting | 16 |
| 4.1 <Executed/> Tag | 16 |
| 4.2 Error Tags | 16 |
| 4.3 <ConfirmationRequired/> Tag | 17 |
| 4.4 <Warning> Tags..... | 17 |
| 4.4.1 Warning Tag Reporting Potentially Serious Consequences..... | 17 |
| 4.4.2 Warning Tag Reporting Unexpected Action by IMM..... | 17 |
| 4.5 <Executing/> Tag | 18 |
| 4.6 Comment Tags | 18 |
| 4.7 <HostService2MinTimeout/> Reply Tag..... | 18 |
| 4.8 Data Reply Tags | 18 |
| 5 IMM-Host Interface in Host Service Mode | 19 |
| 5.1 Entering Host Service Mode | 19 |
| 5.1.1 Entering Host Service Mode with Power Cycling..... | 19 |
| 5.1.2 Entering Host Service Mode with Serial Line Transition | 20 |
| 5.1.3 Entering Host Service Mode with HostFlag Transition..... | 20 |

| | | |
|-----------|--|-----------|
| 5.2 | Sending Commands from Host to IMM..... | 20 |
| 5.2.1 | Terminating Commands to IMM | 20 |
| 5.3 | IMM Prompts and Echoing..... | 20 |
| 5.4 | Backspace | 20 |
| 5.5 | Timeout Parameters in Host Service Mode | 21 |
| 5.5.1 | TModem2 Maximum Wait for Start of IM Reply..... | 21 |
| 5.5.2 | TModem3 Maximum IM Receive Time..... | 21 |
| 5.5.3 | TModem4..... | 21 |
| 5.6 | Exiting Host Service Mode | 21 |
| 6 | Sending Commands to Remote Devices | 22 |
| 6.1 | Capturing IM Line | 22 |
| 6.2 | Sending Data | 22 |
| 6.3 | Sending Wake-up Tone..... | 22 |
| 6.4 | Releasing IM Line | 22 |
| 7 | Receiving Commands from Remote Devices..... | 23 |
| 7.1 | Waking IMM from IM line | 23 |
| 7.2 | Receiving Non-Host Commands | 23 |
| 7.2.1 | Receiving Group and Global Non-Host Commands | 23 |
| 7.3 | Detecting Wake-Up Tones..... | 23 |
| 7.4 | Receiving Host Commands..... | 24 |
| 7.4.1 | Device ID and Serial Number Addressed Host Commands..... | 24 |
| 7.4.2 | Group Addressed and Global Host Commands | 24 |
| 7.5 | Disabling Signal Detector | 24 |
| 8 | IMM-Host Interface in IM Service Mode | 25 |
| 8.1 | Waking Host with IMM..... | 25 |
| 8.1.1 | Waking Host with Hardware Handshaking (IMFlag)..... | 25 |
| 8.1.2 | Waking Host with a Serial Byte..... | 27 |
| 8.1.3 | Waking Host with a Serial Break Condition..... | 27 |
| 8.2 | Confirming Host Wake-up..... | 27 |
| 8.2.1 | Confirming Host Wake-up with HostPrompt | 27 |
| 8.2.2 | Confirming Host Wake-up with HostFlag Input..... | 27 |
| 8.3 | Sending Command to Host..... | 27 |
| 8.4 | Terminating Command to Host | 28 |
| 8.5 | Host Replies | 28 |
| 8.5.1 | Host Echo..... | 28 |
| 8.5.2 | Termination of Host Reply | 28 |
| 8.5.3 | Termination Character from Host | 29 |
| 8.5.4 | HostFlag Transition..... | 29 |
| 8.5.5 | Inter-character Delay Termination (THost4) | 29 |
| 8.5.6 | Host Return to Sleep | 29 |
| 8.6 | Timeout Parameters in IM Service Mode | 29 |
| 8.6.1 | THost0 Maximum Wait for Wake-up Confirmation | 30 |
| 8.6.2 | THost1 Delay after Confirm..... | 30 |
| 8.6.3 | THost2 Maximum Wait for Start of Reply | 30 |
| 8.6.4 | THost3 Maximum IM Reply Transmission Time | 30 |
| 8.6.5 | THost4 Inter-Character Delay Timeout..... | 30 |
| 8.6.6 | THost5 IMFlag Hold after Reply Termination | 30 |
| 9 | Handling of Binary Data | 31 |
| 10 | Using Sample Data | 32 |
| 10.1 | Adding and Appending Samples | 32 |
| 10.2 | Retrieving List of Samples | 32 |
| 10.3 | Reading Samples | 33 |
| 10.4 | Erasing Samples | 33 |
| 10.5 | Overwriting Samples and Memory Overflow..... | 33 |
| 11 | IMM Commands | 34 |
| 11.1 | Command Addressing..... | 34 |
| 11.2 | Sending Commands..... | 34 |

| | | |
|-----------|---|-----------|
| 11.2.1 | Interrupting Replies to IM Commands | 35 |
| 11.3 | Group Commands | 35 |
| 11.4 | Command Summary Tables | 35 |
| 11.4.1 | Config Type 1 Commands | 35 |
| 11.4.2 | Config Type 1 Blocked Commands | 36 |
| 11.4.3 | Config Type 2 Commands | 36 |
| 11.5 | Detailed Command Descriptions | 39 |
| 11.6 | Config Type 2 Set Commands | 56 |
| 12 | Interface Mode Shortcuts | 64 |
| 13 | Electrical Specification | 65 |
| 13.1 | IMM Connector Description | 65 |
| 13.2 | Hardware I/O Line Characteristics | 66 |
| 13.3 | Absolute Maximum Ratings | 66 |
| 13.4 | Power Supply Input | 66 |
| 13.5 | Power Consumption Estimates | 67 |
| 14 | Manual Revision History | 68 |
| 15 | Firmware Revision History | 69 |
| 16 | Index | 71 |

1 IMM Overview and Conceptual Model

The IMM is a low-power microprocessor peripheral that interfaces new or existing instruments capable of logic-level serial, RS-232 to inductive moorings. IMM telemetry is compatible with existing instruments and Surface Inductive Modems (SIMs) from Sea-Bird. The IMM includes advanced capabilities that allow instrument self-discovery, inductive line status reporting, cable coupler performance measurement, reduced restrictions on Inductive Modem (IM) host devices, and a more convenient and robust communications interface. Most older Sea-Bird instruments require firmware or hardware upgrades to use these advanced features, but use of advanced features will not disturb older instruments on the same mooring.

The goals of an inductive modem system are:

- Serve as a minimally restrictive and highly reliable communications link.
- Provide specific features required to achieve scientific goals in spite of any limitations of the communications link.
- Provide simple and stable interfaces to help developers make reliable, maintainable, and reusable software at any scale.
- Provide advanced features to improve reliability and efficiency of system programming, setup, and maintenance.

1.1 Scope and Intent of IMM Conceptual Model

This section contains a high level view of the IMM conceptual model, providing a view of the basic properties and structures of concepts. In an instrument, the conceptual model serves as a reference with which to describe what aspects of the instrument are affected by each possible operation, whether a user command accepted by a device or a firmware function executing in the device. Careful command and function descriptions using sound conceptual models help reduce unintended consequences, both by instrument end users and by programmers. This can dramatically improve the ease of use and overall reliability of systems.

The IMM conceptual model must make sense in the context of models of individual instruments of all types from all manufacturers, as well as buoy systems and future network nodes. High-level users must design systems that communicate with and control a wide variety of devices. This task becomes more manageable if a high-level conceptual model may be applied to these devices.

The IMM will be used by multiple systems with fundamentally different models. To accommodate system designers, the IMM model must be stable and sufficiently specified to allow designers of the high level systems to easily interface the IMM model with their systems. The IMM conceptual model serves as part of a bridge between the specific features and limitations of inductive modems and oceanographic instruments and the more generalized requirements of higher-level network models and interfaces. Laying the foundation of this bridge now helps prevent changes in the IMM's interface to host instruments. Addition of Host IDs, Host Data, and Sample Data storage to the IMM and potential for XML support in this data may allow host devices to conform to high-level interfaces without changing their firmware. This may prove valuable for system integrators who have limited influence over the firmware of the instruments generating data.

The IMM's conceptual model fundamentally differs from that of an instrument:

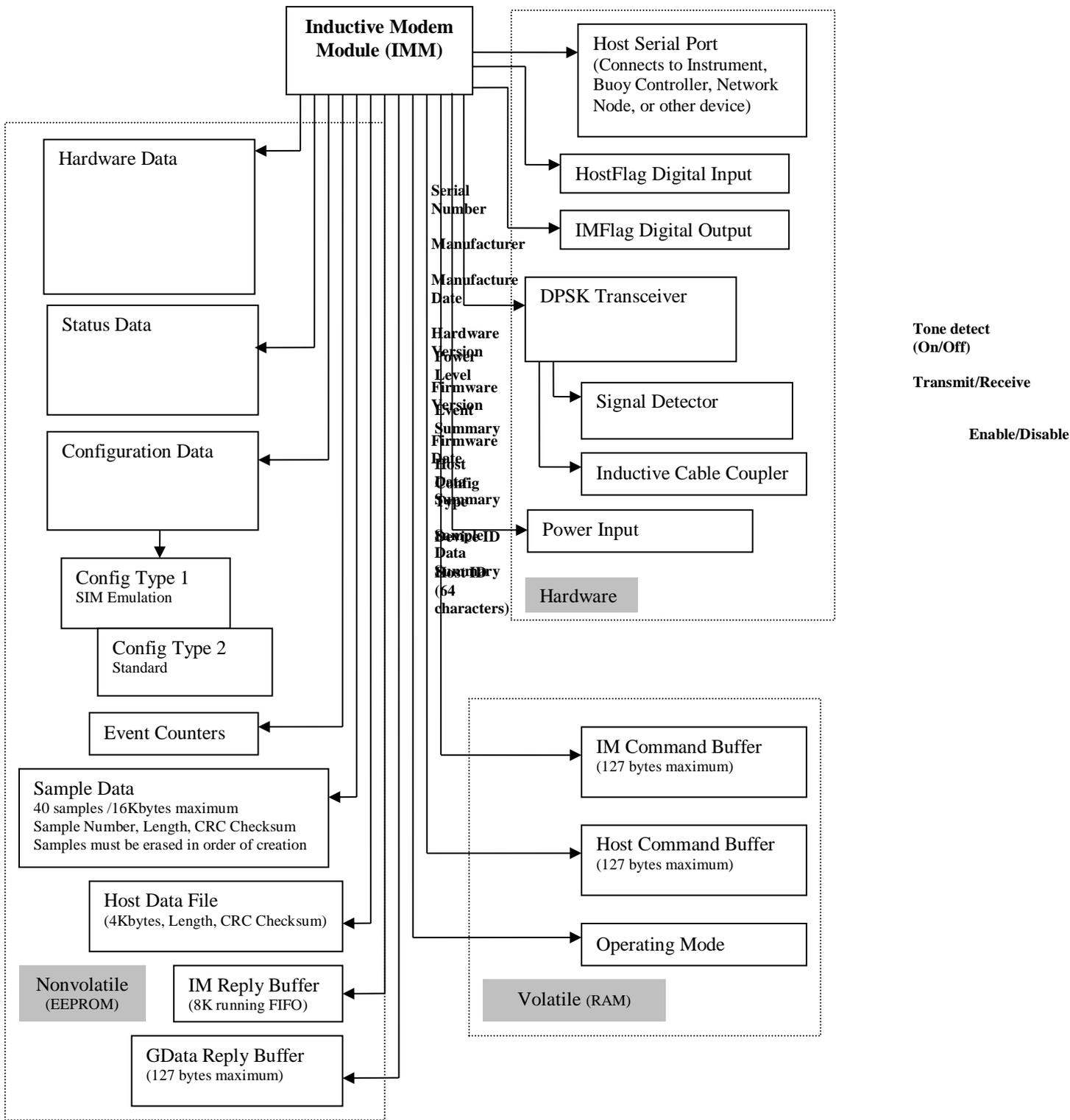
- It does not perform any environmental measurement or collect any scientific data directly.
- It is capable of controlling and communicating directly with an instrument or other device (some instruments may have this property).
- It is capable of communicating with a network of instruments through the IM line.

There is an attractive logical split between surface devices¹ and underwater² devices, but the IMM model intentionally avoids making any conceptual split on this basis. This split may impose unnecessary limitations on the system that may interfere with future expansion.

¹Like the existing Surface Inductive Modem (SIM)

² Like the existing SBE 44 Underwater Inductive Modem (UIM)

Figure 1.1: Diagram of IMM Conceptual Model



Each item in the IMM conceptual model diagram is described in *Section 2: IMM Conceptual Model*.

2 IMM Conceptual Model

The IMM model is divided into three logical sections:

- Hardware - physical devices and circuits.
- Nonvolatile - structures stored in EEPROM that persist when the IMM loses power.
- Volatile - structures stored in RAM that are cleared or reset when the IMM loses power.

2.1 Hardware

2.1.1 Host

Every IMM has a single logical host. The host may be an instrument, a buoy controller, a node of a larger network, or some unknown hardware capable of serial communications. In Sea-Bird products like the SBE 37-IMP or SBE 37-IMP-IDO, which have an IMM built into the instrument, the host instrument is the SBE 37.

2.1.2 DPSK Transceiver with Tone Detect

The DPSK Transceiver is the electronic system that transmits and receives data with Differential Phase Shift Keying (DPSK) on the inductive mooring line. The transceiver is usually off; it turns on in response to a host command or an event from the Signal Detector.

2.1.3 Signal Detector

The Signal Detector is a low-power electronic system that detects the presence of a wake-up tone or DPSK transmission on the inductive mooring line. When the detector identifies a signal, it enables the DPSK Transceiver to receive and decode the signal. The Signal Detector can be disabled to save a small amount of power if it is not required¹.

2.1.4 Inductive Cable Coupler

The Inductive Cable Coupler is an electrical device, analogous to an antenna, which inductively couples the IMM's DPSK Transceiver to an inductive mooring cable. If the Inductive Cable Coupler is damaged or incorrectly installed, the communications system will not be reliable. The IMM has the ability to measure the performance of an assembled cable coupler.

There are two classes of Inductive Cable Couplers, 100-turn and 20-turn. The IMM is optimized for low power with the 100-turn coupler, but does work with the older, 20-turn coupler. Although different classes of cable couplers *may* be used on the same mooring, it is generally not recommended.

The Inductive Cable Coupler is also available in various forms, such as for integration with the instrument (as in the SBE 37-IM, 37-IMP, 37-IMP-IDO, 39-IM, 16*plus*-IM V2), for use as an external coupler (as in the connection to a buoy – referred to as the ICC), or for use as a test coupler for laboratory testing of the system.

Note: An IMM in the buoy can be operated without a cable coupler if using a direct cable connection configuration; this requires an external transformer in place of the cable coupler.

¹ The detector should be disabled only in an IMM with a host device that initiates all mooring communications. This is typical of many buoy controllers.

2.1.5 Power Input

The IMM may be directly connected to batteries, or power may be supplied by the host instrument. The IMM will not power up until the input voltage rises above 6.0V. If the voltage applied to the IMM rises above 19V, a voltage limiting circuit activates. This circuit allows the IMM to operate with reduced efficiency at input voltages up to 30V. Systems with higher voltage power supplies should use the IMM Daughter board, which has a high-efficiency switching supply for 12V to 80V inputs.

The IMM has two internal, high-efficiency switching power supplies: one powers the low-voltage microcontroller, and the other provides a higher voltage for the DPSK transmitter. The transmitter power supply is bypassed and does not activate until the input voltage is greater than 9V, which is the maximum safe transmitting voltage. The IMM can measure the voltage of the transmitter power supply; this measurement nearly matches the input voltage when the input voltage is less than 9V.

2.1.6 HostFlag Digital Input

HostFlag is a digital input used to power-on the IMM or terminate commands or replies to commands.

2.1.7 IMFlag Digital Output

IMFlag is an open collector digital output used to signal arrival of data or completion of an operation.

2.2 Nonvolatile Structures

2.2.1 Hardware Data

Hardware Data contains data describing the IMM, and can be modified only by the factory. A standard command (GetHD) allows read-only access to Hardware Data. Hardware data includes: Serial Number, Manufacturer, Hardware Manufacture Date, Hardware Version, Firmware Version, and Firmware Date.

Each IMM and IM-enabled instrument (an instrument that communicates via IM telemetry, but does not include an integrated IMM) has a unique serial number greater than 100 and less than 2^{32} . Serial numbers are set by Sea-Bird and cannot be changed. In Sea-Bird IM-enabled instruments, the serial number of the inductive modem is the serial number of the instrument.

2.2.2 Configuration Data

Configuration data includes all settable parameters related to operation of the IMM, including all host communication settings. A standard command (GetCD) allows you to review Configuration Data. Configuration data does not include the IMM serial number, because the serial number cannot be changed, but the reply to GetCD includes the serial number for convenience.

Configuration data may be changed in the field, and may be updated by the host, by a PC with a special cable, or over the inductive modem. Some commands that edit configuration data must be confirmed by repeating the command.

The Configuration Type (Config Type) defines the set of available commands in the IMM. Configuration Types provide for backwards compatibility as well as future expansion. Config Type 1 emulates the SIM-coupled and SIM-direct modems. Config Type 2 is the standard IMM. Additional Configuration Types may be defined later for custom applications.

The Device ID is a user-programmable two-digit number from 00 to 99, and is used to address the IMM.

The Host ID is a user-programmable 64-character string, provided to help describe and identify the host.

2.2.3 Status Data

Status Data, describing the IMM, contains data that change while deployed. Status Data is not always directly modifiable, but the ability to clear or reset elements of Status Data is provided. A standard command (GetSD) allows read-only access for the host or through the IM line. Multiple commands may affect Status Data; see the command summaries for details. Status data includes Power Level, Event Counters, Discovered by List, Host Data Summary, and Sample Data Summary.

2.2.4 Event Counters

Event Counters are an array of counters recording the number of occurrences of common timeouts, power-on resets, watchdog timer resets, and other conditions. Event Counters are analogous to the *Black Box* in airplanes; they help identify the root cause of a malfunction. A standard command (GetEC) allows read-only access for the host or through the IM line; the event counters are cleared with ResetEC or ResetEventCounters.

2.2.5 Host Data File

The Host Data File is a nonvolatile memory buffer, physically located on the IMM, reserved for storage of host-specific information. This buffer may be used to store anything, including binary data. Some users may store host setup information, an XML file describing the host, or JAVA scripts for processing data from the host. The buffer is limited to 4 Kbytes, and is read- and write-accessible from both the serial port and the inductive modem². A Host Data summary is available with file length and checksum³.

2.2.6 Sample Data

The Sample Data structure is a nonvolatile memory⁴ space where the host may store data for later upload through the Inductive Modem. In buoy applications, if the host periodically writes to Sample Data, the buoy controller can upload the data from the IMM without disturbing the host. This data buffering allows host instruments to take measurements and report data independently of the buoy controller's update rate.

The Sample Data structure allows a variable maximum number of logical Samples (up to 40), each containing a variable number of bytes up to the full memory space (16 Kbyte). Sample Data is read- and write-accessible from both the serial port and the inductive modem. A Sample Data summary is available with length and checksum⁵ for each sample in memory.

A Sample ID is automatically assigned to each Sample as it is created. The Sample ID is a 32-bit unsigned integer, starting at 1 and incrementing each time a sample is created. There is no way to reset the assignment of Sample IDs in an IMM.

2.2.7 GData Reply Buffer

GData is a global IM command used to make time-synchronized measurements across multiple instruments of different types. It allows a single command to initiate a sample, even though different host instruments may require different command strings. The GData Reply Buffer is a 127 byte buffer that holds the reply to the last GData command. GData may also be sent as a group command or addressed to a specific instrument, e.g. !G0:GDATA or !00GDATA. Note that #00GDATA is generally an error - this causes the remote device with ID 00 to send the string "GDATA" to its host.

2.2.8 IM Reply Buffer

The IMM has an 8K buffer for replies from the host to the Inductive Modem. Bytes are placed in the buffer as they are received from the host, and they are removed from the buffer as they are transmitted to the IM line. If the number of bytes in the buffer exceeds 8192, data is lost and a DPSK_BUF_OFLOW event is logged in the Event Counters. It is possible to transmit up to 20 Kbytes in a reply, but we recommend limiting replies to 8 Kbytes or less to avoid complications.

² True binary support is provided with the HostFileWrite command, which is accepted only from the serial port, not from the inductive modem. The HostFileAppend command is accessible from both the serial port and the inductive modem, but the Host-to-IMM termination sequence ("\r\n") may make storage of binary data with this command difficult.

³ The Host Data File checksum is implemented in firmware V1.05 and later. The checksum will read as 0x00000000 for firmware versions < 1.05.

⁴ Sample Data is implemented in EEPROM that has a nominal endurance of 1 million erase/write cycles. The Sample ID allows a rough estimate of the number of cycles the EEPROM had accumulated. If an instrument records 6 samples per hour, each of which uses the entire memory space of the EEPROM, manufacturer's data suggest 99.9 % of EEPROMs will last more than 20 years.

⁵ Sample Data checksums are implemented in firmware V1.05 and later. Checksums will read as 0x00000000 for firmware versions < 1.05.

2.3 Volatile Structures

Volatile structures are reset when the IMM executes a power-on reset.

2.3.1 Operating Mode

IMM operating mode describes the action the IMM is performing at a particular moment; it is not a user-settable parameter. There are four operating modes: Host Service, IM Service, IM Receive, and Sleep. Power-off might be considered an additional operating mode in some applications. The IMM may be configured to enter Sleep mode or Host Service mode on power up.

2.3.2 Host Command Buffer

The Host Command Buffer is a 127 byte RAM buffer used only during Host Service mode. Characters received from the host are stored in this buffer until the termination sequence arrives (carriage return followed by line feed, as generated by the Enter key on Windows computers). When the termination sequence arrives, the IMM attempts to execute the contents of the Host Command Buffer as a command. All host commands must fit in this buffer or the IMM will generate an error. This size limitation does not apply to data sent after executing a command (e.g., the SampleAdd command, which prompts for up to 16 Kbytes of additional data).

2.3.3 IM Command Buffer

The IM Command Buffer is a 127 byte RAM buffer where incoming commands from the IM line are stored. All IM commands must fit in this buffer or the command will be rejected as transmission errors and will not be executed. This size limitation does not apply to replies to commands.

3 Operating Modes

Operating mode describes the action the IMM is performing at a particular moment; it is not a user-settable parameter. There are four operating modes: Host Service, IM Service, IM Receive, and Sleep. Power-off might be considered an additional operating mode in some applications.

3.1 Host Service

The IMM enters Host Service mode when communication is initiated by the IMM's host. In Host Service mode the host may send commands to the IMM to store data, change settings, or initiate communication with remote devices. The IM receive buffer is cleared after processing each command from the host⁶. The IMM remains in Host Service mode until it receives the PwrOff command from the host or the 2-minute timer expires.

During Host Service mode, the IMM cannot process commands arriving from the IM line:

- If the IMM is in Host Service mode but is not processing a command from the host, the IMM responds to properly addressed incoming IM signals with a <Busy/> tag.
- If the IMM is in Host Service mode and is processing a command from the host, the IM line is ignored.

3.2 IM Service

The IMM enters IM Service mode when any valid command (including a command that relays data to the host) arrives from the IM line. The IMM remains in this mode until the command is executed. When in IM Service mode, the IMM does not respond to commands or wake-up attempts from the host, nor does it respond to additional commands arriving from the IM line. The IMM does not send <Busy/> replies to the IM line in IM Service mode.

3.3 IM Receive

The IMM enters IM Receive mode if the signal detector is enabled and a signal is detected on the IM line. A wake-up tone is **not** required to switch an IMM into IM Receive mode (Note: A wake-up tone **is** required for IM-enabled instruments that do not have an integral IMM to enter the analogous mode; e.g., SBE 16*plus*-IM, 16*plus*-IM V2, 37-IM, 37-IMP (firmware < 4.0), 39-IM, 44).

Every 2 minutes the IMM tests the line for a valid signal⁷. If there is no signal, the IMM returns to Sleep mode. Arrival of a valid IM command resets the 2-minute timer. If the PwrOff command arrives from the IM line, the IMM pauses for 100 milliseconds, then switches to Sleep mode.

In IM Receive mode, the IMM responds to wake-up attempts from the host by switching directly to Host Service mode. At the end of Host Service mode, the IMM re-tests the IM line. If a valid signal is present, the IMM resets the 2-minute timer and resumes IM Service (Note: The same timer is used for IM Service and Host Service; it is reset on every mode change).

3.4 Sleep

Sleep mode is the IMM's low-power-on state. If the signal detector is enabled and a signal is identified on the IM line, the IMM switches to an intermediate mode for a few milliseconds to determine if the signal is a valid IM signal. If the signal is valid, the IMM switches to IM Receive mode; otherwise, it returns to Sleep.

⁶ If the Host takes several seconds to enter data, as in a typical SampleAdd command, the IMM might act on the command stored in its IM receive buffer when it finishes processing the host command. This could cause the IMM to send a <Busy/> reply tens of seconds after the IM command actually arrived. This late reply might collide with other signals on the IM line. Clearing the IM receive buffer after each host command prevents this problem.

⁷ This test should not interfere with incoming communications from the IM line.

3.5 Power Cycling IMM

The IMM contains EEPROM memory, which may be unreliable if power is removed during a write operation. The safest way to power down the IMM is to use the following sequence:

1. Use ReleaseLine to explicitly terminate the IM session (if the IM line was captured),
2. Send the PwrOff command,
3. Wait 100 milliseconds,
4. Disconnect power and set the logic level serial input to floating or logic low state.

This power-off procedure is primarily a concern for applications where the IMM will be power cycled in the field. If the IMM is power cycled in the field, remove power for at least 15 seconds before re-connecting. The IMM will not fully power on until the input voltage rises above 6 volts.

4 Reply Tags and XML-like Formatting

The IMM uses an XML-like format to pass information from the IMM to the host in Host Service mode and from the IMM to the remote device in IM Service mode. If the IMM connected to a host never operates in Host Service mode, the host will never see any XML-like tags. The XML-like format is not used when the IMM forwards incoming commands to a host device or when the IMM relays normal replies from a host device⁸.

XML-like tags are designed to help host programmers identify common IMM responses and Errors with non-command-specific code. The purpose of the similarity to XML is to take advantage of a format already understood by most developers. IMM reply tags should not be considered XML.

4.1 <Executed/> Tag

The Executed tag indicates the IMM has completed processing the last command. It is always the last thing sent by the IMM after processing a command⁹ (except the prompt), and the IMM never sends more than one Executed tag in response to a command¹⁰. If the IMM returns an Executed tag with no Error or ConfirmationRequired tags, the command was understood and successfully processed. The IMM does not add Executed tags to host responses to commands received from the IM line.

4.2 Error Tags

Error tags, of the form <ERROR type='Type' msg='Message'/>, provide a standard format for the IMM to report error conditions and common negative results. Each Error tag contains a type and a message (msg). The Error type is intended for machine parsing; the number of types will be kept to a minimum and tied to the IMM's conceptual model. The msg parameter is intended for system debugging. Users should avoid creating dependencies on the content of the msg parameter.

Any command processed in Host Service mode or received from the IM line in IM Service mode may result in one or more Error tags. The IMM may send more than one Error tag in response to a single command.

Table 4.2.1 Error Types

| Type | Description |
|------------------|--|
| INVALID COMMAND | <ul style="list-style-type: none"> Command not recognized by instrument Instrument command buffer overflowed (IMM 127 byte buffer) |
| INVALID ARGUMENT | <ul style="list-style-type: none"> Improperly formatted argument Argument out of fixed permissible range Argument out of dynamic permissible range (IMM sample data - attempt to download data for sample ID not in memory) |
| NOT ALLOWED | <ul style="list-style-type: none"> Valid command not allowed in current operating mode or configuration (e.g., not allowed while logging or when IM line not captured) Valid command not allowed via specific interface (e.g., disabling IM receiver via IM interface) |
| FAILED | <ul style="list-style-type: none"> Negative result (e.g., EEPROM test failed, CaptureLine command failed, cable coupler test failed) |
| ABORTED | <ul style="list-style-type: none"> User aborted active process (e.g., escape key in IM transmit or discovery) |
| IM RECEIVE | <ul style="list-style-type: none"> Inductive Modem received invalid characters |
| OVERFLOW | <ul style="list-style-type: none"> Buffer or data structure overflowed and data was truncated |
| TIMEOUT | <ul style="list-style-type: none"> Timing parameter violated |
| POWER FAIL | <ul style="list-style-type: none"> Low battery or low power condition |
| FACTORY SETUP | <ul style="list-style-type: none"> A factory setting was not entered; customers should never see this error |

⁸ The Error tag may be appended to host replies to incoming commands if there is an error in interacting with the host, such as a timeout condition.

⁹ Config Type 1 does not use Executed tags for commands that mimic SIM behavior. In Config Type 2, the *ID= command does not use Executed tags.

¹⁰ Note that a command sent to a remote device may get an Executed tag from the remote device within a <RemoteReply> tag - followed by an Executed tag from the IMM itself, which will be outside the RemoteReply tag.

4.3 <ConfirmationRequired/> Tag

The IMM sends the ConfirmationRequired tag in response to commands that must be repeated before they take effect. This is common when changing critical elements in Configuration Data, such as the serial type or baud rate. The ConfirmationRequired tag is often, but not always, accompanied by a Warning tag.

Example:

```
IMM>setbaudrate=19200
<WARNING>
IMM will power down
next power up after confirm will use
new baud rate
</WARNING>
<ConfirmationRequired/>
<Executed/>
IMM>setbaudrate=19200
<Executed/>
```

In the example, the first time SetBaudRate= is sent, the ConfirmationRequired tag indicates the command did not take effect. When SetBaudRate= is sent a second time, the ConfirmationRequired tag is not sent again, indicating the parameter is set.

4.4 <Warning> Tags

Warning tags have two significant functions.

- Report potentially serious consequences that may result from confirming and executing a command which returns the ConfirmationRequired tag.
- Notify the user or host program of unexpected actions by the IMM in response to a command.

The function of a specific Warning tag depends on the command that generated the tag. Refer to the detailed command descriptions for interpretation of Warning tags for particular commands.

To ensure maximum upgradeability, any user program that parses IMM output should be capable of processing warning tags. Programs may depend on the existence of Warning tags in response to specific commands, but there must be no dependency on the content of Warning tags. Warning tags may expand to include a type parameter in the opening tag, similar to the type parameter in Error tags. Unlike Error tags, Warning tags are not self-closing.

4.4.1 Warning Tag Reporting Potentially Serious Consequences

The first function of warning tags is seen primarily in *Set* commands. Several Set commands return Warning tags when execution of the command may have serious consequences.

Example:

```
IMM>setserialtype=1
<WARNING>
IMM will power down
next power up after confirm will use
RS232 Serial interface
</WARNING>
<ConfirmationRequired/>
<Executed/>
IMM>setserialtype=1
<Executed/>
```

4.4.2 Warning Tag Reporting Unexpected Action by IMM

This second function of Warning tags is used by the SampleAdd command. The SampleAdd command may return a warning tag if an old sample was automatically erased to make room for the new sample. This is not an error condition, since the new sample was created; but the host may need to identify this condition, so a Warning is generated.

4.5 <Executing/> Tag

The Executing tag indicates the IMM is processing the command, but additional time is required to complete the command. The Executing tag may be sent any number of times in response to a single command. In general, the IMM sends an Executing tag at least once per second if an operation takes more than 1 second to complete. The IMM does not interrupt open reply tags (e.g., <RemoteReply> or <SampleData>) to send Executing tags. The IMM does not send an Executing tag in response to # commands received from the IM line.

4.6 Comment Tags

Comment tags are presented in the standard XML format, as in <!--this is a comment-->. Comments carry data intended to help users become familiar with the operation of the IMM and keep track of the IMM state when controlling it with a terminal program. Comment tags are enabled when the DebugLevel is 3 or higher.

Comments are specifically not intended for machine parsing. To ensure maximum upgradeability, any user program that parses IMM output should be capable of ignoring comment tags and should have no dependency on the existence or content of comment tags.

4.7 <HostService2MinTimeout/> Reply Tag

If the IMM is in Host Service mode and the host does not enter a valid command to the IMM for 2 minutes, the IMM times out and goes back to sleep. The IMM sends the HostService2MinTimeout tag to indicate it has timed out. After sending the tag, but before the IMM goes to sleep, the IMM pauses for 250 milliseconds¹¹. During this pause, the IMM will not respond to any input from the host. After the pause the IMM clears its serial buffers and goes to sleep immediately. Note that this reply tag is sent automatically by the IMM upon timeout, while all other reply tags are sent in response to a command.

The 2-minute timeout can be prevented by setting EnableHostFlagWakeup=1 and holding the HostFlag signal in a logic high state.

4.8 Data Reply Tags

Data reply tags are returned in response to particular commands, and are closely linked to the IMM conceptual model and to Sea-Bird's evolving standard instrument model. Data reply tags for each command are described in detail with the command in *Section 11: IMM Commands*.

Examples of Data Reply Tags:

```
<EventSummary numEvents=' [UINT16] ' />
<Sample ID=' [HEX32] ' Len=' [HEX16] ' CRC=' [HEX32] ' />
<Power Vtx=' [% .1f] ' / >
<LineStatus S=' [STRING10] ' />
```

Reply Tag Value Definitions (note – brackets do not appear in data):

[UINTn] = unsigned n-bit integer (*Example*: UINT16 is 16-bit integer).

[HEXn] = unsigned n-bit integer in Hex format (*Example*: HEX16 is 16-bit integer in Hex).

[% .nf] = floating point number with n digits to right of decimal place (*Example*: %.1f is a number with 1 digit to the right of the decimal place, such as 8.5 or 19.5)

[STRINGn] = n-character string (*Example*: STRING10 is 10-character string)

¹¹ The 250 millisecond pause guarantees that the IMM will not immediately wake up if the Host echoes the timeout tag.

5 IMM-Host Interface in Host Service Mode

Host Service mode allows the host to directly send configuration commands, store or retrieve data, or use the IMM to send information to other devices in the inductive loop. Host Service mode is initiated by the host¹². After entering Host Service mode, the IMM remains ready to receive commands until the 2-minute timer expires or the host sends the PwrOff command.

Nearly all host-IMM interaction in Host Service mode takes the form of a command string sent by the host to the IMM, termination of the command, and the IMM's response to the command. The only exception is the <HostService2MinuteTimeout/> flag, which the IMM sends in response to a timeout condition. The host may send any number of commands to the IMM in a single Host Service session, provided the delay between completion of a command and arrival of the next valid command is no longer than 2 minutes.

5.1 Entering Host Service Mode

There are three ways to enter Host Service mode. Each method can be independently disabled. Some applications require all three methods to be disabled, preventing the IMM from ever entering Host Service mode¹³.

The host may initiate Host Service mode at any time except when the IMM is in IM Service mode or during a blackout period after a PwrOff command, power cycle, or 2-minute timeout. When the IMM enters Host Service mode, it notifies the host with a prompt and the IMFlag signal. Any characters received after the IMM confirms wakeup are added to the command buffer, so the host should initialize its serial port before initiating Host Service mode. If spurious characters are sent during wakeup, the host may need to send the termination sequence ('\r\n') to clear the command buffer before sending the first command¹⁴.

Table 5.1.1 Summary of Host Service Mode Interfacing

| <i>IMM</i> Wake-up | Confirm Wake-up | Command Termination | Reply Termination | <i>IMM</i> Return To Sleep |
|--|--|--|---|---|
| <ul style="list-style-type: none"> • Power-up • Serial byte or break • HostFlag low-high transition | <ul style="list-style-type: none"> • IMFlag pulled to GND • IMM prompt | <ul style="list-style-type: none"> • Termination character sequence (CR followed by LF, '\r\n') • HostFlag low-high transition • 2-minute timeout | <ul style="list-style-type: none"> • Termination character • Serial line break • Inter-character delay (2 seconds) • Prompt | <ul style="list-style-type: none"> • Power-down • PwrOff command • 2-minute timeout <p>Note: IMM will not sleep if Hostflag is High and EnableHostFlagWakeup=1</p> |

5.1.1 Entering Host Service Mode with Power Cycling

This is intended for applications like simple buoy controllers, where all IM activity is initiated by the buoy controller. If EnableHostServeOnPwrUp=1, the IMM automatically enters Host Service mode when power is applied. Before turning off power, the host should send the PwrOff command to the IMM and wait 100 milliseconds. Power must remain off for at least 15 seconds to guarantee complete power down¹⁵. If the logic level serial input is used, the serial line must be either floating or in a logic low state or the IMM cannot properly power down. This is a good conservative approach to IM system design, because it guarantees a clean software state in the IMM each time the buoy controller initiates communication.

Enable/disable this feature with SetEnableHostServeOnPwrup=.

¹² Host Service mode can also be initiated by power cycling the IMM if EnableHostServeOnPwrup=1.

¹³ Host Service mode is typically completely disabled in simple instruments that send data only in response to incoming commands from the IM line.

¹⁴ The IMM ignores any strings from the host with fewer than 3 characters, excluding the termination sequence.

¹⁵ Systems with high input voltages should wait longer than 15 seconds before re-applying power to the IMM. A system with 30V input should wait 1 minute.

5.1.2 Entering Host Service Mode with Serial Line Transition

If EnableSerialIMMWakeup=1, the IMM enters Host Service mode in response to a high-low transition on the serial input (logic level or RS-232).

Enable/disable this feature with SetEnableSerialIMMWakeup=.

Note: EnableSerialIMMWakeup cannot be set to 0 if EnableSignalDetector is set to 0, and vice versa. This prevents the IMM from entering a state from which it cannot wake up.

5.1.3 Entering Host Service Mode with HostFlag Transition

If EnableHostFlagWakeup=1, the IMM enters Host Service mode in response to a low-high transition on the HostFlag input. If EnableHostFlagWakeup=1, the IMM also wakes up in response to a static high condition on HostFlag when it prepares to go to sleep; this may occur when the IMM wakes in response to IM line signals, even if no commands address the IMM. When EnableHostFlagWakeup=1 and the IMM is in Host Service mode, the IMM will not time out after 2 minutes when HostFlag is high. Users may hardwire HostFlag high to guarantee the IMM will not exit Host Service mode.

Enable/disable this feature with SetEnableHostFlagWakeup=.

5.2 Sending Commands from Host to IMM

After waking the IMM and receiving wakeup confirmation (prompt and/or IMFlag pull-down), the host may send commands to the IMM. Bytes sent to the serial port are stored in the IMM command buffer until the command is terminated. If the IMM command buffer overflows, an error is generated upon termination of the command.

5.2.1 Terminating Commands to IMM

Commands may be terminated with the termination sequence or a low-high transition of the HostFlag signal. The termination sequence for commands from the host to the IMM is carriage return (CR, 0x0D or '\r') followed by line feed (LF, 0x0A or '\n'); the termination sequence cannot be disabled. Enabled/disable HostFlag termination with SetEnableHostFlagTerm=.

Note that the TermFromHost setting does not apply in Host Service mode; it applies only in IM Service when the host responds to data from the IMM.

5.3 IMM Prompts and Echoing

The IMM can send prompts and echo characters to the host. These features are important when using a terminal program to communicate with the IMM. The prompt may also serve as confirmation to the host that the IMM is awake and serving the host. Enable/disable prompts with SetEnablePrompt=; enable/disable echoing with SetEnableEcho=. Note that the prompt applies only to Host Service mode; the prompt is never sent to the host in IM Service mode.

5.4 Backspace

For convenience, the IMM can respond to the backspace character when a user sends commands with a terminal program. Disable the backspace action when transmitting binary data with SetEnableBackspace=0.

5.5 Timeout Parameters in Host Service Mode

The IMM has three timeout settings, in addition to the 2-minute timeout, that apply only in Host Service mode.

5.5.1 TModem2 Maximum Wait for Start of IM Reply

After transmitting an IM command to which a reply is expected, the IMM waits for the first byte of the reply until a timeout occurs¹⁶. This timeout and the setting to control the timeout period are called TModem2. Valid values for TModem2 are from 5 to 3000; units are tens of milliseconds (TModem2=100 means 1 second). TModem2 is analogous to THost2 on the IM receiver.

Note: If the remote device is an IMM, the remote IMM initiates its reply before attempting to wake-up the remote host.

5.5.2 TModem3 Maximum IM Receive Time

After transmitting a command and receiving a start of reply indicator, the IMM listens to the reply until the reply terminates or a timeout occurs. This timeout and the setting to control the time period are called TModem3. TModem3 is analogous to THost3 on the IM receiver. Set TModem3 of the IMM transmitting the command to match or be slightly greater than THost3 of the IMM receiving the command. If the receiver's THost3 is much greater than the transmitter's TModem3, the transmitter might start to send a new command while the receiver is still replying to the first command. Valid values for TModem3 are from 100 to 60000; units are tens of milliseconds (TModem3=100 means 1 second).

In Config Type 1, TModem3 has the same meaning, but different units. TModem3 units in Config Type 1 are seconds, with a maximum value of 600 seconds. If the GetCD response shows TModem3 > 600, the actual delay in Config Type 1 is 600 seconds.

5.5.3 TModem4

TModem4 applies only to Config Type 1 *bit* binary commands (a special case for use exclusively with older SBE 44 and UIM devices). TModem4 is the maximum inter-character delay time before the IMM terminates reception of a reply to a *bit* command. This is analogous to the BinaryGap setting on the SIM. Valid values for TModem4 are 0 to 3000; units are tens of milliseconds (TModem4=3000 means 30 seconds). The TModem4 setting exists in IMM firmware V1.02 and later. Note that inter-character delays cannot terminate IM reception in most communication, because the system may receive noise characters if the IMM is receiving while nothing is transmitting to the line.

5.6 Exiting Host Service Mode

The host should exit Host Service mode with the PwrOff command. The PwrOff command returns the IMM to Sleep mode. If the host removes the IMM's power supply, it should first send the PwrOff command and then wait 100 milliseconds before cutting power.

¹⁶ If no IM carrier is detected, the IMM does not wait for THost2, but terminates immediately with a NO REPLY Error.

6 Sending Commands to Remote Devices

Commands can be sent to remote devices only when the IMM is in Host Service mode.

The Inductive Mooring line allows half-duplex communication, meaning only one device may transmit at a time. If two devices attempt to transmit at the same time, both messages will be corrupted. This limitation prevents the IMM from allowing a standard modem interface, with data transmitted and received simultaneously.

All communication on the Inductive Mooring line takes the form of commands and replies to commands. Only one command may be sent at a time. Not all commands generate replies. If a command requires a reply, the reply begins immediately after the command is received. No additional commands can be sent until the reply is completed. The maximum length of a command is 127 bytes. The maximum recommended size of a reply is 8 Kbytes. The absolute maximum duration of a reply is three minutes, which allows up to 20 Kbytes.

6.1 Capturing IM Line

When the IMM is in Host Service mode, the host may initiate communication on the IM line. Since the IM line is half-duplex (data flows in only one direction at a time), the host should first verify that no other device is using the IM line. The CaptureLine command checks for other devices transmitting on the IM line, and then reserves the IM line by transmitting a carrier signal. This process is called *capturing the line*¹⁷.

The ForceCaptureLine command starts transmitting a carrier signal immediately, without first checking that no other device is using the IM line. This is appropriate for systems where only one device will attempt to capture the line.

6.2 Sending Data

After capturing the IM line, the host may send data to other devices on the line. The IMM supports the old !ii and #ii command formats, as well as new serial number and group command addressing schemes. See *Section 11: IMM Commands* for a complete description. Note that any outgoing IM command may return a NOT ALLOWED Error if the line is not captured, or a FAILED Error and a POWER FAIL Error if the transmit voltage is too low.

6.3 Sending Wake-up Tone

Older IM devices required a wake-up tone before they would power-up and start receiving data from the IM line. The IMM does not wait for a wake-up tone, but starts receiving as soon as a carrier signal is detected. To support legacy devices, the host may send a wake-up tone to the line with the SendWakeUpTone command.

6.4 Releasing IM Line

After completing a session on the IM line, the host may use the ReleaseLine command to close the IM session. ReleaseLine automatically sends PwrOff to the IM line to switch remote IM devices out of IM Receive mode back into Sleep mode. Alternatively, the host may send the PwrOff command to the IMM, which also causes the IMM to automatically send the power off signal to remote devices and release the IM line if it is captured, and then tells the IMM to return to Sleep mode.

Notes:

1. If the IM line is not properly released, all IM devices on the mooring stay awake for up to 2 minutes, causing an unnecessary drain on their batteries.
2. **PwrOff** tells the remote devices to return to Sleep mode, releases the line, and then tells the **IMM** to return to Sleep mode. **SendPwrOff** just tells the **remote devices** to return to Sleep mode. These are different commands

¹⁷ The probability that two devices will collide while capturing the line (i.e., both devices believe they have the line), given that they receive the CaptureLine command at the same time, is approximately 1 in 1000.

7 Receiving Commands from Remote Devices

The IMM receives data only if the Signal Detector is enabled. When another IM device captures the IM line, the IMM automatically enters IM Receive mode, unless the Signal Detector is disabled or the IMM is already in Host Service mode. If properly addressed data arrives when the IMM is in Host Service mode, the IMM responds with a <Busy/> signal. When a properly addressed command arrives from the IM line and the IMM is in IM Receive mode, the IMM enters IM Service mode and executes the command. If the command requires interaction with the IMM's host, as in # commands and certain global commands, the IMM enters IM Service mode and initiates communication with its host.

7.1 Waking IMM from IM line

The IMM has an internal Signal Detector with multiple internal wake-up states in which it detects incoming signals and differentiates between incoming IM transmissions and electrical noise caused by radar, lightning strikes, or other activity. When the IMM identifies an IM carrier on the line (i.e., another device has captured the line or sent a legacy wake-up tone), the IMM enters IM Receive mode.

7.2 Receiving Non-Host Commands

Commands addressing the IMM itself start with an !, except for certain global non-host commands. These non-host commands are executed without disturbing the host. When an IMM processes an incoming ! or non-host global command, it cannot respond to requests from the host.

7.2.1 Receiving Group and Global Non-Host Commands

Group commands and most non-host global commands do not allow a response. The *ID= and ID? commands are exceptions, but these commands may only be used when a single receiving device is coupled to the IM line. Note that any ! command can be made global by addressing it to group 0, as in !G0:Command. Group commands starting with ! are executed in the receiving IMM(s) in the same way as individually addressed commands, but replies to these commands are disabled.

7.3 Detecting Wake-Up Tones

The IMM does not require wake-up tones. The IMM supports wake-up tone detection for compatibility with some older UIM devices that sent a wake-up tone to prompt a SIM to start data upload.

The recommended alternative for new designs is to send a command from one IMM to the other to initiate a data upload, rather than sending a wake-up tone. A false wake-up tone event may be caused by electrical noise or radar signals, but a false incoming command is extremely unlikely.

Incoming wake-up tones are reported to the host only if EnableToneDetect=1. Reporting a wake-up tone has the same effect as a remote device sending '#iWAKE-UP TONE DETECTED'. This means the IMM initiates communication with the host to report a wake-up tone in the manner in which it would report an incoming command.

Enable/disable detection of wake-up tones with SetEnableToneDetect=.

7.4 Receiving Host Commands

When the IMM receives a host command (typically starting with #), the IMM wakes the host (see *Section 8: IMM-Host Interface in IM Service Mode*) and forwards the command to the host. If a reply is required, the IMM relays the host reply to the IM line. There are several optional methods of waking the host, using the serial port or hardware handshake lines. There are also several timeout parameters governing communications with the host. The way in which the command is addressed generally determines whether or not a reply is required.

7.4.1 Device ID and Serial Number Addressed Host Commands

A command addressed with a device ID has the prefix #ii (for example, #01startssampling). A command addressed with a serial number has the prefix #Snnnnnnnn: (for example, #S70000009:startssampling). Commands addressed with these methods always have a reply. The reply may be empty if the host does not provide any data, or in the special case where THost2 is set to 0.¹⁸ The process of waking the host, sending the incoming data, and relaying a reply is described in *Section 8: IMM-Host Interface in IM Service Mode*.

7.4.2 Group Addressed and Global Host Commands

Group and Global host commands never allow an IM reply. The host may send a reply to its IMM, but the IMM will not transmit the data to the IM line. Some Group and Global commands cause the IMM to store the host reply in Sample Data or the GData Reply Buffer. Refer to the detailed command descriptions for these commands. Note that any # command can be made global by addressing it to group 0, as in #G0:Command. This works only if *Command* is understood by all host devices coupled to the IM line. Group commands starting with # are executed in the receiving IMM(s) in the same way as individually addressed commands, but replies to these commands are disabled.

7.5 Disabling Signal Detector

Entering IM Receive mode may be disabled with SetEnableSignalDetector=0. This prevents the IMM from detecting incoming signals (or noise) on the IM line, and is appropriate for applications where all IM communication initiates from a single device, as in most buoy controllers. Disabling the Signal Detector prevents detection of wake-up tones.

Note: EnableSignalDetector cannot be set to 0 if EnableSerialIMMWakeup is set to 0, and vice versa. This prevents the IMM from entering a state from which it cannot wake up.

¹⁸ Setting THost2 to 0 is a method of disabling host replies. See the detailed description for THost2 in *Section 8: IMM-Host Interface in IM Service Mode*.

8 IMM-Host Interface in IM Service Mode

In IM Service mode the IMM will either respond to a local command (e.g., !ii, !G0:) or initiate communications with the Host. The IMM responds to local commands without disturbing the Host. The IMM initiates communications with the host only in three situations:

- The first, and most common situation, is when a # command with matching device ID arrives from the IM line, as would occur if the remote device that captured the IM line received a #ii command.
- The second situation is when a global command or matching group command arrives on the IM line, as would occur if the device that captured the line received SendGData (a global command) or a #G0: group command.
- The third and least common situation is when EnableToneDetect=1 and a wake-up tone is detected, as would occur if the device that captured the line received the SendWakeupTone command.

In IM Service mode, the IMM-Host interface has seven stages:

1. IMM wakes the host,
2. Host confirms wake-up,
3. IMM sends the command to the host,
4. IMM terminates the command,
5. Host sends a reply to the IMM,
6. Host terminates the reply,
7. Host returns to sleep.

Table 7.5.1 Summary of IMM-Host Interface Options in IM Service Mode

| <i>Host Wake-up</i> | <i>Confirm Wake-up</i> | <i>Command Termination</i> | <i>Reply Termination</i> | <i>Host Return To Sleep</i> |
|--|--|--|--|--|
| <ul style="list-style-type: none"> • IMFlag pulled to GND • Serial byte ('\r') • Serial break | <ul style="list-style-type: none"> • Prompt from Host or break • HostFlag low-high transition • Maximum wait for confirm (THost0) | <ul style="list-style-type: none"> • Termination character (TermToHost) • Serial line break • Inter-character delay (50mS)¹⁹ | <ul style="list-style-type: none"> • Termination character (TermFromHost) • HostFlag low-high transition • Inter-character delay (THost4)²⁰ • Maximum wait for start of reply (THost2) • Maximum IM reply transmission time (THost3) | <ul style="list-style-type: none"> • Automatic²¹ • IMFlag release |

8.1 Waking Host with IMM

In most applications, the host is asleep until a command arrives. The IMM uses one of the methods described below to wake up the host before sending the incoming command.

8.1.1 Waking Host with Hardware Handshaking (IMFlag)

Hardware handshaking allows the host to remain in a very low-power state while waiting for incoming commands on the DPSK line, relying on the open collector IMFlag signal to initiate its wakeup. In some applications it may be possible to control the host power supply directly with the IMFlag signal. More often, the IMFlag signal initiates the host power-up, and the host controls its own power-down sequence.

If EnableAutoIMFlag=1, the IMM automatically uses IMFlag to signal the host to wake-up. The IMFlag signal is pulled to ground when the host should wake up. The IMM releases the IMFlag line after an adjustable delay (THost5, described below) when the data transfer process is completed.

Enable/disable automatic switching of IMFlag output with SetEnableAutoIMFlag=. Disabling IMFlag applies to all operating modes.

¹⁹ This delay would be measured by the host. The IMM sends the incoming data to the host in one burst with minimal separation between bytes.

²⁰ This is the inter-character delay specified by the Configuration Data setting THost4.

²¹ The host may go back to sleep immediately after terminating its reply to the incoming command.

8.1.2 Waking Host with a Serial Byte

If EnableHostWakeupCR=1, the IMM sends a '/r' character (carriage return) to the host to generate a wake-up when initiating communication. If EnableHostWakeupBreak=1, the break condition is sent before the '/r' character.

Enable/disable the host wake-up carriage return feature with SetEnableHostWakeupCR=.

8.1.3 Waking Host with a Serial Break Condition

If EnableHostWakeupBreak=1, the IMM sends a serial break condition to the host. The SerialBreakLen setting controls the duration of the break. If EnableHostWakeupCR=1, the break condition is sent before the '/r' character.

Enable/disable the host wake-up break feature with SetEnableHostWakeupBreak=.

8.2 Confirming Host Wake-up

The IMM waits for confirmation that the host is awake before transmitting the command to the host. The THost0 setting provides an adjustable timeout for the confirmation process. If THost0 is set to 0, the host wake-up confirmation stage is disabled and the IMM sends the command to the host after the THost1 delay, without waiting for confirmation.

8.2.1 Confirming Host Wake-up with HostPrompt

If EnableHostPromptConfirm=1, the host may confirm wake-up by sending a string matching the HostPrompt setting. HostPrompt is a string of 2 to 8 characters. Host prompt confirmation operates independently of other host wake-up options, except for the special case of THost0=0, which causes the IMM to ignore the host prompt settings.

Enable/disable the host prompt confirm feature with SetEnableHostPromptConfirm=. Set the host prompt with SetHostPrompt=.

If EnableHostPromptConfirm=1 and the IMM receives the wrong prompt from the host, the IMM logs the event as if it were a THost0 timeout.

8.2.2 Confirming Host Wake-up with HostFlag Input

If EnableHostFlagConfirm=1, the host may confirm wake-up with a low-high transition of the HostFlag input. HostFlag confirmation operates independently of other host wake-up options, except for the special case of THost0=0, which causes the IMM to ignore the HostFlag confirmation setting.

Enable/disable the HostFlag confirm feature with SetEnableHostFlagConfirm=.

8.3 Sending Command to Host

The IMM allows an adjustable delay between receiving confirmation of wake-up from the host and transmission of the command to the host. This delay is set by THost1, described below. The THost1 delay always applies, even in the special case where host wake-up confirmation is disabled by setting THost0=0.

After the THost1 delay, the IMM sends the command to the host. There is no significant inter-character delay in transmission of data to the host.

8.4 Terminating Command to Host

After sending the command to the host, the IMM terminates the command. There are several options for terminating commands. Commands may be terminated with a single termination character, a carriage return followed immediately by a line feed (CRLF), or a serial line break²². The IMFlag signal cannot terminate commands to the host²³. The host may interpret an inter-character delay greater than 50mS as termination.

The IMM has separate settings for the termination sent to the host and the termination received from the host (see SetTermToHost= and SetTermFromHost=). TermToHost applies to termination of the command sent by the IMM to the host. Each setting is recorded in a single byte termination code. The IMM interprets any value less than 250 as a single byte termination character, 253 as a serial line break (applicable for SetTermToHost only), 254 as CRLF, and 255 as no termination. If no termination is selected, the inter-character delay timeout should be shortened to act as a terminator. In CRLF termination, CR or LF may be sent to the device individually; only the CRLF combination acts as termination.

8.5 Host Replies

The host may begin its reply immediately after the IMM terminates the command. No inter-character delay is required in the host reply. The THost2 setting allows an adjustable timeout for the start of the host reply. If the host does not send at least one character (excluding echoed characters) in reply to the command within THost2, an error will be generated and communication cut off. There is an exception when THost2 is 0 - the IMM does not allow the host to reply at all; the IMM ignores the host response and behaves as if the host terminated its reply immediately.

The IMM starts transmitting the host reply to the IM line immediately. If bytes arrive from the host faster than they can be transmitted, they are stored in the Host Reply buffer. This is an 8 Kbyte FIFO buffer. No buffer overflow indication is provided for the host, but if the buffer overflows an error is generated and an event is logged in the Event Counters.

The IMM provides an inter-character-delay timeout (THost4) and a maximum reply time timeout (THost3). THost3 sets a hard limit on the maximum duration of a host reply at 180 seconds. This allows up to 20 Kbytes of data. Sea-Bird recommends limiting replies to 8 Kbytes to avoid problems with the Host Reply buffer and the maximum reply time.

8.5.1 Host Echo

If the host echoes characters of the command sent by the IMM back to the IMM, the echoed command becomes part of the host reply. If EnableStripHostEcho=1, the IMM automatically deletes the host's echo of the command and any termination characters. Note that setting EnableStripHostEcho=1 will cause loss of 1 or more bytes of the host reply if the host does not echo the command.

Enable/disable the strip host echo function with SetEnableStripHostEcho=.

Note: The IMM's echo must be disabled if the host echoes characters! Use SetEnableEcho=0 to disable the IMM's echo.

8.5.2 Termination of Host Reply

Host replies to commands may be terminated with a single termination character, a carriage return followed immediately by a line feed (CRLF), a low-high transition on the HostFlag input, or an inter-character delay timeout. Serial line breaks cannot terminate host replies. The IMM has separate settings for the termination sent to the host and the termination received from the host (see SetTermToHost= and SetTermFromHost=). TermFromHost applies to replies sent by the host to commands received from the IMM. If no termination is selected, the THost4 inter-character delay timeout should be shortened to act as a terminator. In CRLF termination, CR or LF may be sent to the device individually; only the CRLF combination acts as termination.

The TermFromHost setting applies only in IM Service mode, when the host responds to data from the IMM. It does not apply in Host Service mode. Termination characters received from the host in IM Service mode are not included in the reply unless the termination character is CRLF (TermFromHost=254).

²² Break condition termination is allowed only in the termination to the host, not from the host.

²³ Terminating commands to the host with IMFlag would cause problems for devices using IMFlag to control their power supplies.

8.5.3 Termination Character from Host

The TermFromHost setting allows specification of a termination character or use of carriage return followed by line feed (CRLF) as a termination sequence. The setting is recorded in a single byte termination code. The IMM interprets any value less than 250 as a single byte termination character, 254 as CRLF, and 255 as no termination. The remaining values are reserved. This feature operates independently of the other termination options, except for the special case where THost2=0, where the termination character setting is ignored.

Configure the termination character for host replies with SetTermFromHost=. The command requires a decimal number as an argument.

8.5.4 HostFlag Transition

If EnableHostFlagTerm=1, the host may terminate replies to commands with a low-high transition of the HostFlag input. This feature operates independently of the other termination options, except for the special case where THost2=0, where the HostFlag termination setting is ignored.

Enable/disable HostFlag termination with SetEnableHostFlagTerm=.

8.5.5 Inter-character Delay Termination (THost4)

Inter-character delay is the time between arrivals of individual characters in streaming serial data. The IMM allows an adjustable inter-character delay timeout with the THost4 parameter. This timeout forces the end of the host reply. The THost4 timer activates after arrival of two characters. In instruments using hardware handshaking, termination characters, or break condition interfacing, the inter-character delay timeout acts as a safety mechanism, cutting off the host in case of malfunction or error²⁴. If a host instrument cannot use handshaking, termination characters, or break conditions, THost4 may act as the primary reply termination mechanism. Inter-character delays are typically much shorter than maximum transmission lengths, so their use may save considerable line transmission time (and power) over a maximum reply time (THost3) timeout.

8.5.6 Host Return to Sleep

The host may go to sleep immediately after terminating its reply to the command. If the host requires a time delay between terminating its reply and release of the IMFlag output or arrival of the next command, the THost5 setting provides an adjustable delay.

8.6 Timeout Parameters in IM Service Mode

Inductive moorings support only one transmitting device at a time. A single device transmitting continuously might disable an entire mooring. Included in the safety features of the IMM are a variety of timeouts required to protect the mooring from damaged or malfunctioning host instruments.

In addition to the required timeouts, several optional timeouts and delays exist to allow the host to power-up, process commands, and power-down. Configuration of these timeouts and delays may be host-specific. The IMM is shipped with default timeout and delay values, but users should expect to configure these parameters to suit their application.

All timeouts and delays are part of the IMM's Configuration Data. Configuration Data can be updated by the host, with a PC and special cable or over the inductive modem. Configuration Data may be changed in the field.

²⁴ The inter-character delay timeout is not the primary safety mechanism, but it offers additional security through redundancy.

8.6.1 THost0 Maximum Wait for Wake-up Confirmation

Some hosts require time to initialize after wake-up. THost0 allows an adjustable timeout period for the IMM to wait for confirmation that the host is awake before the IMM assumes the host is awake. The delay does not apply to host-initiated communication. Valid values for THost0 are from 0 to 1000; units are tens of milliseconds (THost0=100 means 1 second).

If THost0 is set to a non-zero value, failure of the host to confirm wakeup before the timeout causes the IMM to log a THost0 event in the Event Counters. If the host interface uses the HostPrompt for wake-up confirmation, and the IMM receives the wrong prompt from the host, the IMM logs the event as if it were a THost0 timeout.

8.6.2 THost1 Delay after Confirm

THost1 provides an adjustable delay between the IMM receiving confirmation that the host is awake (or the THost0 timeout expiring) and the IMM sending data to the host. The delay does not apply to host-initiated communication. This delay is useful for slow instruments. THost1 can be set to 0 in most applications. Valid values for THost1 are from 0 to 1000; units are tens of milliseconds (THost1=100 means 1 second).

8.6.3 THost2 Maximum Wait for Start of Reply

After the IMM sends an incoming IM command to a host, it waits for a reply from the host. Some hosts may respond immediately, others may take seconds or longer to reply. THost2 is the maximum time the IMM waits for the first byte of reply from the host before assuming there is no reply. If the THost2 period passes with no reply bytes received from the host, the IMM responds to the IM line with an Error and logs a THost2 event in the Event Counters. Valid values for THost2 are from 100 to 3000, or 0; units are tens of milliseconds (THost2=100 means 1 second).

In the special case of THost2=0, the IMM assumes its host will never send a reply. All incoming # commands receive an automated empty reply after the host wakeup, wake-up confirmation, and transmission of the incoming command to the host is completed. No Error is reported or logged in this case. Setting THost2=0 is useful for simple unidirectional communication.

8.6.4 THost3 Maximum IM Reply Transmission Time

This is the maximum allowed duration of an IM reply to a # command, measured from the arrival of the command in the IMM to the termination of the IMM's transmission of the host's reply. This time limit acts on the 1200 baud reply on the IM line, as well as the higher speed host serial port. If the host fills the IMM's 8 Kbyte reply buffer at 9600 baud, the IMM transmits from that buffer to the IM line at 1200 baud until either the buffer is empty or the THost3 timeout occurs. Any data remaining in the reply buffer after the THost3 timeout is discarded. When a THost3 timeout occurs, the host reply is cut off, an Error is forwarded to the IM line, and a THost3 event is logged in the Event Counters. Valid values for THost3 are from 100 to 18000; units are tens of milliseconds (THost3=100 means 1 second).

8.6.5 THost4 Inter-Character Delay Timeout

Inter-character delay is the time between arrivals of individual characters in streaming serial data. The inter-character delay timeout forces the end of the host's command or reply. This is useful with host devices that do not support termination characters in their replies. The inter-character delay timeout is enabled only after the arrival of the first character from the host. THost4 timeouts are reported as Errors only if the DebugLevel is 6 or higher. Valid values for THost4 are from 5 to 6000; units are tens of milliseconds (THost4=100 means 1 second).

8.6.6 THost5 IMFlag Hold after Reply Termination

Some Hosts require extra time to power down after replying to a command, especially if the IMFlag signal controls the host power supply. THost5 provides an adjustable delay between termination of the host's reply and release of the IMFlag by the IMM. The delay does not apply to host-initiated communication²⁵. Valid values for THost5 are from 0 to 3000; units are tens of milliseconds (THost5=100 means 1 second).

²⁵ When the host initiates communication with the IMM, the IMFlag activates to signal IMM wake-up and deactivates to signal IMM entering Sleep mode or IM Receive mode, if there is an IM signal present at the end of Host Service mode.

9 Handling of Binary Data

The low-level IM transmission protocol reserves some characters to mark the start and end of transmissions, specify addressed instruments, and perform other IM-specific functions. These functions are abstracted from the user as much as possible to reduce system complexity. Unfortunately, the technique used prevents true binary data transmissions. When data sent by a host for transmission over the IM line contains characters reserved by the IM transmission protocol, these characters are automatically replaced by two characters, similar to the '\r' and similar codes used in many programming languages. This transformation increases the number of bytes that must be transmitted, thus increasing the expected transmission time.

A relatively simple rule allows specification of transmission times for binary data:

- The following binary values transmit as single characters -
All values $\geq 0x20$
TAB (0x09)
LF (0x0A)
CR (0x0D)
- All other binary values transmit as two characters and are automatically decoded when received²⁶. The only observable effect for characters $< 0x20$ is an increase in transmission time. There is no observable effect for TAB (0x09), LF (0x0A), CR (0x0D), and characters $\geq 0x20$.

It is the user's responsibility to prevent early termination of their commands from the host to the IMM (with the '\r\n' termination sequence) and in replies from the host to the IMM if termination characters are used. Termination characters less than 0x20 are allowed. Termination characters $\geq 0xFA$ (250 decimal) are NOT allowed.

²⁶ The backspace character (0x08) is interpreted as a command to delete the previously sent character if EnableBackspace=1. Disable the backspace character for binary data transfers with SetEnableBackspace=0.

10 Using Sample Data

The Sample Data structure is a nonvolatile memory space for data buffering between the host and a remote device connected to the inductive mooring. In buoy applications, data buffering allows host instruments to take measurements and report data independently of the update rate of the buoy controller. The IMM buffers data from the host instrument in the Sample Data structure between periodic uploads by the buoy controller (typically once per hour). The buoy controller may then use the Exclusive²⁷ command when uploading the larger data set. This reduces the total active time of the inductive mooring and extends battery life for other IM instruments on the mooring.

The Sample Data structure allows a variable maximum number of logical Samples (up to 40), each containing a variable number of bytes up to the full available memory space (16 Kbyte). Sample Data is read- and write-accessible from both the serial port and the inductive modem. A Sample Data summary is available with length and checksum²⁸ for each sample in memory.

Responses to the GData command may be stored in the Sample Data structure. If EnableGDataToSample=1, the response to GData is diverted from the GData Reply buffer and is added as a new Sample in SampleData. This frees host devices from the 127 byte limit of the GData Reply Buffer.

A **Sample ID** is automatically assigned to each Sample as it is created. The Sample ID is a 32-bit unsigned integer, starting at 1 and incrementing each time a sample is created. There is no way to reset the assignment of Sample IDs in an IMM²⁹.

10.1 Adding and Appending Samples

There are three ways to add or append samples:

- The SampleAdd and SampleAppend commands allow addition of binary data. These commands are accepted only from the host, not from the IM interface.
- The SampleAddLine: and SampleAppendLine: commands are accepted from the host or over the Inductive Modem, and allow addition of up to 100 bytes of data with standard termination. These commands are not intended for binary data. They do not include the CR and LF termination characters in the recorded data.
- Receiving a GData command from the IM line when EnableGDataToSample=1.

10.2 Retrieving List of Samples

The SampleGetSummary and SampleGetList commands provide views of the contents of the Sample Data structure without downloading the contents of particular samples. SampleGetSummary returns a SampleDataSummary tag with the number of samples in memory, total length of data in memory, and available memory. SampleGetList returns a list identifying all samples in memory with the Sample ID, length, and checksum for each sample.

Example:

```
IMM>samplegetsummary
<SampleDataSummary NumSamples='1' TotalLen='2529' FreeMem='13855' />
<Executed />
IMM>samplegetlist
<SampleList>
<Sample ID='0x00000008' Len='79' CRC='0x00000000' />
<Sample ID='0x00000007' Len='2529' CRC='0x00000000' />
</SampleList>
<Executed />
```

²⁷ The Exclusive command is not supported by IM-enabled instruments that do not have an integral IMM (e.g., SBE 16*plus*-IM, 16*plus*-IM V2, 37-IM, 37-IMP [firmware < 4.0], 39-IM, or 44) or by the IMM with firmware < 0.85.

²⁸ Sample Data checksum is implemented in firmware 1.05 and later. The checksum will read as 0x00000000 for firmware versions < 1.05.

²⁹ Sample Data is implemented in EEPROM that has a nominal endurance of 1 million erase/write cycles. The Sample ID allows a rough estimate of the number of cycles the EEPROM had accumulated. If an instrument records 6 samples per hour, each of which uses the entire memory space of the EEPROM, manufacturer's data suggest 99.9 % of EEPROMs will last more than 20 years.

10.3 Reading Samples

Samples are read with the SampleGetData: or SampleGetLast command.

- SampleGetData: allows data retrieval by sample ID, returning the data for a specific sample ID. The argument for SampleGetData: is a hexadecimal sample ID number (format 0x0000000A, A, or 0xA; case independent).
- SampleGetLast retrieves data for the most recently added sample, which has the highest sample ID.

Data is output is enclosed in SampleData tags, including parameters for the sample ID and CRC code.

Example:

```
IMM>samplegetlast
<SampleData ID='0x7' LEN='16' CRC='0x0'>This is the data</SampleData>
IMM>samplegetdata:7
<SampleData ID='0x7' LEN='16' CRC='0x0'>This is the data</SampleData>
```

10.4 Erasing Samples

The Sample Data structure requires that samples be erased in the order in which they are created. This means the lowest Sample ID in memory must be erased first. There are three commands for erasing samples:

- SampleErase: only erases the oldest sample in memory, which has the lowest sample ID. The sample ID of the sample to erase must be specified; this prevents accidental deletion of samples.
- SampleEraseMultiple: erases all samples starting at the specified ID and including all lower IDs.
- SampleEraseAll erases all samples in memory.

10.5 Overwriting Samples and Memory Overflow

If the Sample Data structure does not have enough memory to complete an add or append command, the data for that transaction is truncated. If an add command is received when the maximum number of samples already exist in memory, the oldest sample is automatically deleted. The maximum number of samples³⁰ may be set in Configuration Data with SetMaxNumSamples=xx, where xx is an integer from 1 to 40.

Example:

The Sample Data structure starts empty. A single sample is added that consumes the entire available memory space. Each time an Add command arrives, a new sample is created. All of these new samples are empty because there is no available memory. This continues until the maximum number of samples is reached. After the maximum is reached, the next Add command causes the original large sample to be erased. This immediately frees the memory that was taken by that original large sample, and the in-process add command successfully records data.

³⁰ If the maximum is set to a number smaller than the number of samples currently in memory, each SampleAdd or SampleAddN will delete only one old sample before creating a new sample, leaving the number of samples in memory greater than the specified maximum until a SampleErase:, SampleEraseMultiple:, or SampleEraseAll command removes some of the older samples.

11 IMM Commands

This section describes the IMM commands. Some commands are valid only when received from the host, others are valid from the host or the IM line. Many commands are valid only in a particular Configuration Type.

11.1 Command Addressing

Many IMM commands are executed by the IMM physically connected to the host sending commands. These are called Direct commands. It is important to distinguish between Direct commands and IM commands.

IM commands are transmitted over the IM line and executed in a remote device. IM command addressing is the process of identifying the device or devices on the inductive mooring that should execute a given command. The host addresses commands by prefixing them with either the ! or # character and the Device ID, serial number, or group number the command is intended for. The format for these prefixes is shown in the Command Addressing Types table below. Commands prefixed with an individual ID or individual serial number address a single instrument. Commands prefixed with a group number address a pre-defined group of instruments.

A small number of IMM commands cause transmission of global commands, which address all instruments on the mooring. Global commands do not use the # or ! prefixes.

Table 11.1.1 Command Addressing Types

| Command Type | Prefix for Delivery to remote IMM | Prefix for Delivery to remote Host | Comments |
|--------------------------|--|------------------------------------|--|
| Individual ID | !ii | #ii | ii = ID (ii = 0 – 99) of remote IMM or remote IM-enabled instrument. Note: ID must be two digits (i.e., 00 is correct, 0 is incorrect) |
| Individual Serial Number | !Sx: | #Sx: | x = serial number of remote IMM ($100 < x < 2^{32}$) or remote SBE 39-IM. Note: 39-IM does not have an internal IMM, but accepts serial number addressing. |
| Group | !Gn: | #Gn: | n = group number (0 – 9) of remote IMMs (0 being the group of all IMMs) or remote SBE 16 <i>plus</i> -IM V2 or 39-IM. Notes: 1. G prefix must be upper case for IMM firmware versions < 1.06³¹. 2. Group commands do not allow replies. 3. 16 <i>plus</i> -IM V2 and 39-IM do not have an internal IMM, but accept group number addressing. |
| Global (legacy) | Global commands do not use # or ! Global commands that have replies (such as ID? and *ID) are allowed only if there are exactly 2 IM devices on the mooring (Example: an IMM in the buoy and in 1 instrument). Global commands that have replies support older instruments and should be avoided when possible. | | |
| Direct | Not applicable | Not applicable | Command sent by the IMM's host (e.g., CaptureLine). |

11.2 Sending Commands

IM commands can be sent only when the IM line is captured. An attempt to send a command with a # or ! prefix when the line is not captured results in a NOT ALLOWED Error.

³¹ IMM firmware versions < 1.06 listened for a reply to group commands if the G prefix was lower case. This error was corrected in firmware 1.06.

11.2.1 Interrupting Replies to IM Commands

After sending an IM command, the host can interrupt the reply by sending the escape character (0x1B). This is useful if the host believes the remote device did not respond and the data in the RemoteReply is noise. Using the escape character stops the IMM from listening to the IM signal, but the remote device may still be transmitting. There is no way to interrupt an in-process transmission from a remote device. The host must not attempt to transmit to the IM line until the remote device stops transmitting. If more than one device transmits to the IM line at the same time, system behavior may be unpredictable.

11.3 Group Commands

Group commands allow addressing of pre-defined logical groups of instruments. For example, it may be useful to place all SBE 39-IMs in Group 1, all SBE 37-IMPs in Group 2, etc., allowing you to set parameters in all similar instruments at the same time. A maximum of nine groups may be defined (numbers 1 - 9). Group 0 is pre-defined as the group of all instruments. Individual IMM cannot belong to more than two groups – Group 0 plus one other group.

Since group commands address multiple instruments, they do not allow replies. A command that ordinarily generates a reply does not generate a reply when sent as a group command.

Not all IMM commands are allowed with group commands. For example, all commands that start with *Get* are disabled, because these commands are not useful without a reply; some *Set* commands are also disabled. If a blocked group command is transmitted there is no error message, but the command is not executed in the remote instruments.

Table 11.3.1 Commands Blocked from Group Addressing

| | | |
|-------------------|---------------------------|---|
| SetConfigType= | TXTest | All commands that start with <i>Get</i> |
| SetInterfaceMode= | EchoIM: | All commands that start with <i>Test</i> |
| SetBaudRate= | SetEnableSignalDetector= | All commands that start with <i>HostFileGet</i> |
| SetDeviceID= | SetEnableSerialIMMWakeup= | All commands that start with <i>SampleGet</i> |
| SetSerialType= | SetEnableHostFlagWakeup= | |

11.4 Command Summary Tables

11.4.1 Config Type 1 Commands

Config Type 1 emulates the SIM-coupled and SIM direct modems. Config Type 1 does not allow incoming commands from the IM line (signal detector is disabled). Several Config Type 1 commands either do not exist in Config Type 2, or behave differently from commands with the same name in Config Type 2. These commands are designed to mimic the behavior of the older SIM board.

Table 11.4.1 SIM Commands Described in SIM Documentation and available in Config Type 1

| | | |
|--------|---------|--------|
| PwrOn | HHMMSS= | GData |
| PwrOff | MMDDYY= | Dataii |
| StayOn | DDMMYY= | |

Table 11.4.2 SIM Commands Not Implemented in Config Type 1

| SIM Command | Comments |
|-------------|---|
| Baud= | Use SetBaudRate=. |
| DataNNMax= | No analogous setting for IMM. |
| RelayMax= | Use SetTHost3=, units are tens of milliseconds. |
| EchoOn | Use SetEnableEcho=1. |
| EchoOff | Use SetEnableEcho=0. |
| AutoPwrOn= | No analogous setting for IMM. |
| BinaryGap= | Use SetTHost4=, units are tens of milliseconds. |

11.4.2 Config Type 1 Blocked Commands

All commands listed for Config Type 2 can be used in Config Type 1, except for blocked commands listed below. Blocked Commands generate a NOT ALLOWED error in IMM firmware version 1.02 and higher.

Table 11.4.3 IMM Commands Blocked in Config Type 1

| | | | |
|------------------|-----------------------------|---------------------------|-------------------|
| CaptureLine | SendWakeupTone | SetEnablePrompt= | SetGDataStr= |
| ForceCaptureLine | SetDeviceID= | SetEnableSerialIMMWakeup= | SetInterfaceMode= |
| GetLineStatus | SetEnableHostFlagTerm= | SetEnableSignalDetector= | |
| ReleaseLine | SetEnableHostPromptConfirm= | SetEnableHostFlagConfirm= | |

11.4.3 Config Type 2 Commands

Table 11.4.4 Config Type 2 Command Summary

| Command (Outgoing IM commands are prefixed with !ii, #ii, !Sx:, #Sx:, !Gn:, or #Gn:) | Allowed from Host (in Host Service mode) | Allowed from IM (in IM Service mode) | Allowed as Group (in IM Service mode) | Comments |
|---|--|--|---|--|
| *ID= | Captured Line Only | no | no | Global IM command (legacy). Confirmation required. Returns Warning tag. |
| *Init | yes | no | no | Restores IMM to factory state. Confirmation required. Returns Warning tag. |
| CaptureLine | yes | no | no | |
| Data | no | yes | no | Similar to GetReply (legacy) |
| Disc | yes | no | no | Provisional Command – may change |
| EchoIM: | no | yes | no | |
| Exclusive: | no | yes | no | Use this command when transmitting from a SIM to an IMM. Use SendExclusive: when transmitting from an IMM to other devices. |
| ForceCaptureLine or FCL | yes | no | no | |
| GData | no | yes | yes | Global IM command |
| GetCD | yes | yes | no | |
| GetConfigType | yes | yes | no | |
| GetEC | yes | yes | no | |
| GetHD | yes | yes | no | |
| GetHostFlag | yes | yes | no | |
| GetHostID | yes | yes | no | |
| GetLineStatus | yes | no | no | |
| GetReply | no | yes | no | Same as Data |
| GetSD | yes | yes | no | |
| HostFileAppendLine: | yes | yes | yes | |
| HostFileErase | yes | yes | no | Confirmation required. |
| HostFileGetData | yes | yes | no | |
| HostFileGetSummary | yes | yes | no | |
| HostFileWrite | yes | no | no | |
| ID? | Captured Line only | yes | no | Global IM command |
| MeasureLineSignal or MLS | yes | no | no | |

| Command | Allowed from Host (in Host Service mode) | Allowed from IM (in IM Service mode) | Allowed as Group (in IM Service mode) | Comments |
|---------------------------|---|---|--|---|
| PwrOff | yes ³² | yes | yes | Global IM command to tell all IMMs to go to sleep (Note: This differs from SendPwrOff) |
| ReleaseLine or REL | yes | no | no | |
| ResetEC | yes | yes | yes | |
| ResetEventCounters | yes | yes | yes | Same as ResetEC |
| SampleAdd | yes | no | no | |
| SampleAddLine: | yes | yes | yes | |
| SampleAppend | yes | no | no | |
| SampleAppendLine: | yes | yes | yes | |
| SampleErase: | yes | yes | no | |
| SampleEraseAll | yes | yes | yes | |
| SampleEraseMultiple: | yes | yes | no | |
| SampleGetData: | yes | yes | no | |
| SampleGetLast | yes | yes | no | |
| SampleGetList | yes | yes | no | |
| SampleGetSummary | yes | yes | no | |
| SendBreak | no | yes | yes | |
| SendExclusive: | yes | no | no | Use this command when transmitting from an IMM. Use Exclusive: when transmitting from a SIM to an IMM. |
| SendGData | Captured Line only | no | no | Use this command when transmitting from an IMM to send the global GData command to the IM line. When using a SIM use GData instead. |
| SendPwrOff | Captured Line only | no | no | Tells remote devices to go to sleep, without powering off the IMM. (Note: This differs from PwrOff) |
| SendStayOn | Captured Line only | no | no | Use this command when transmitting from an IMM. Use StayOn when transmitting from a SIM |
| SendWakeupTone or SWT | Captured Line only | no | no | Use this command when transmitting from an IMM. Use PwrOn when transmitting from a SIM |
| SetBaudRate= | yes | no | no | 1200, 2400, 4800, 9600, 19200 ³³ Confirmation required. Returns Warning tag. |
| SetConfigType= | yes | no | no | 1 or 2. Confirmation required. Returns Warning tag. |
| SetDebugLevel= | yes | yes | yes | 0-9 |
| SetDeviceID= | yes | yes | no | 0-99, same as SetID= |
| SetEnableAutoIMFlag= | yes | yes | yes | 0 or 1 |
| SetEnableBackspace= | yes | yes | yes | 0 or 1 |
| SetEnableBinaryData= | yes | yes | yes | 0 or 1 |
| SetEnableEcho= | yes | yes | yes | 0 or 1 |
| SetEnableFullPwrTX= | yes | yes | yes | 0 or 1 |
| SetEnableGDataToSample= | yes | yes | yes | 0 or 1 |
| SetEnableHostFlagConfirm= | yes | yes | no | 0 or 1 |
| SetEnableHostFlagTerm= | yes | yes | no | 0 or 1 |
| SetEnableHostFlagWakeup= | yes | yes | no | 0 or 1 |

³² In Host service, the PwrOff command automatically transmits PwrOff to the IM line if the IM line is captured.

³³ Support for 38400 baud was removed in firmware version 1.05.

| Command | Allowed from Host (in Host Service mode) | Allowed from IM (in IM Service mode) | Allowed as Group (in IM Service mode) | Comments |
|-----------------------------|---|---|--|---|
| SetEnableHostPromptConfirm= | yes | yes | yes | 0 or 1 |
| SetEnableHostServeOnPwrUp= | yes | yes | yes | 0 or 1 |
| SetEnableHostWakeupCR= | yes | yes | yes | 0 or 1 |
| SetEnableHostWakeupBreak= | yes | yes | yes | 0 or 1 |
| SetEnablePrompt= | yes | yes | yes | 0 or 1 |
| SetEnableSerialIMMWakeup= | yes | yes | no | 0 or 1. Confirmation required and returns Warning tag if set to 0. |
| SetEnableSignalDetector= | yes | no | no | 0 or 1. Confirmation required and returns Warning tag if set to 0. |
| SetEnableStripHostEcho= | yes | yes | yes | 0 or 1 |
| SetEnableToneDetect= | yes | yes | yes | 0 or 1 |
| SetGDataStr= | yes | yes | yes | 1-32 character string |
| SetGroupNumber= | yes | yes | yes | 0-9 |
| SetHostID= | yes | yes | yes | 4-64 character string |
| SetHostPrompt= | yes | yes | yes | 1-7 character string |
| SetID= | yes | yes | no | 0-99, same as SetDeviceID= |
| SetIMFlag= | yes | yes | yes | 0 or 1 |
| SetInterfaceMode= | yes | no | no | 1-14. Confirmation required and returns Warning tag. Note: For Versions <1.0, this command was SetConfigMode=. |
| SetMaxNumSamples= | yes | yes | yes | 1-40 |
| SetModemPrompt= | yes | yes | yes | 1-7 character string |
| SetSerialBreakLen= | yes | yes | yes | 1-255 |
| SetSerialType= | yes | yes | no | 0 – 2. Confirmation required and returns Warning tag. |
| SetTermFromHost= | yes | yes | yes | 0-255 |
| SetTermToHost= | yes | yes | yes | 0-255 |
| SetTHost0= | yes | yes | yes | 0-1000 |
| SetTHost1= | yes | yes | yes | 0-300 |
| SetTHost2= | yes | yes | yes | 0, or 100-3000 |
| SetTHost3= | yes | yes | yes | 100-18000 |
| SetTHost4= | yes | yes | yes | 0-1000 |
| SetTHost5= | yes | yes | yes | 5-3000 |
| SetTModem2= | yes | yes | yes | 5-3000 |
| SetTModem3= | yes | yes | yes | 100-18000 |
| SetTModem4= | yes | yes | yes | 0-3000 |
| StayOn | yes | yes | yes | Global IM command |
| Test20TurnCoupler or T20CC | Captured Line only | yes | no | |
| TestCableCoupler or TCC | Captured Line only | yes | no | |
| TestEEPROM | yes | yes | no | |
| TXTest | no | yes | no | |

11.5 Detailed Command Descriptions

Notes:

1. Command descriptions define whether a command is allowed from the IMM's host, from the Inductive Mooring (IM) line, and/or as a Group command.
2. When addressing a single IMM through the Inductive Mooring, the command formats shown are written for addressing the IMM by its ID. Any command that can be sent to a single IMM in this way can also be sent by addressing the IMM with its serial number. For example, the format shown for transmission of GetCD through the IM is !00getcd (transmit command to IMM with ID=00); it is also valid to transmit this command as !S70000019:getcd (transmit command to IMM with serial number = 70000019).

| | | | |
|--|---|--------|-----------|
| *ID= | Sets Device ID of remote device (legacy). Use SetDeviceID= with serial number addressing in new designs. | | |
| Allowed from / Format: | Host? Yes (Captured Line Only). *id=01 | IM? No | Group? No |
| This is a global command that returns data. It must not be used if more than one instrument is coupled to IM line. This command does not use reply tags. | | | |
| Modifies: | Device ID of all instruments on mooring. | | |
| Requires: | There must be exactly one receiving IM device on mooring. | | |
| Effects: | Sets Device ID | | |
| Returns: | WARNING tag | | |
| Returns: | ConfirmationRequired tag | | |
| Example: | <pre> IMM>*id=01 <RemoteReply><WARNING> Repeat Command to set id=01 Do not repeat if more than one IM device is coupled to the line!</WARNING> <ConfirmationRequired/> <Executed/></RemoteReply> <Executed/> IMM>*id=01 <RemoteReply><Executed/></RemoteReply> <Executed/> </pre> | | |

| | | | |
|---------------------------|--|--------|-----------|
| *Init | Resets IMM connected to host to factory state | | |
| Allowed from / Format: | Host? Yes *init | IM? No | Group? No |
| Modifies: | All IMM settable parameters - Serial Type, Baud Rate, Configuration Data, Host File, Sample Data, GData String, Event Counters, etc. | | |
| Requires: | Nothing | | |
| Effects: | Resets IMM to factory default state. | | |
| Returns: | WARNING tag | | |
| Returns: | ConfirmationRequired tag | | |
| Example: | <pre> IMM>*init <WARNING> IMM will revert to factory state next power after confirm up will use RS232 Serial at 9600 baud Contents of Sample Data and HostFile will be erased Device ID will reset to 00 Event Counters will reset </WARNING> <ConfirmationRequired/> <Executed/> IMM>*init <Executed/> </pre> | | |

| | | | |
|------------------------|--|--------|-----------|
| CaptureLine | Checks for other devices transmitting on IM line, and then reserves IM line by transmitting a carrier signal. | | |
| Allowed from / Format: | Host? Yes captureline | IM? No | Group? No |
| Modifies: | DPSK Transceiver State | | |
| Requires: | Nothing | | |
| Effects: | Attempts to start transmitting a carrier on IM line, first checking for other transmitting devices. If line is captured when this command arrives, IMM releases line for 100mS and attempts to capture it again. | | |
| Returns: | FAILED error if another device is transmitting on line, no error if line is captured. | | |
| Returns: | FAILED error and POWER FAIL error if transmit voltage is too low. This condition can be caused by a bad cable coupler. | | |
| Returns: | FAILED error if a DPSK Exclusive is in process. | | |
| Example: | IMM>captureline <Executed/> | | |
| Example: | IMM>captureline <ERROR type='FAILED' msg='LINE BUSY' /> <Executed/> | | |
| Example: | IMM>captureline <ERROR type='FAILED' msg='Low Transmit Voltage - low battery or bad coupler' /> <ERROR type='POWER FAIL' msg='Transmit Voltage Vtx=1.9 ' /> <Executed/> | | |

| | | | |
|------------------------|---|--------------------|-----------|
| Data | Return host reply of last GData command (legacy). Use GetReply in new designs. | | |
| Allowed from / Format: | Host? No | IM? Yes !00data | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | Contents of GData reply buffer, prefixed with instrument ID. | | |
| Example: | !01DATA <RemoteReply>01,this is the gdata reply</RemoteReply> <Executed/> | | |

| | | | |
|---------------------------|--|--------|-----------|
| Disc (provisional) | Execute line discovery process. This is a provisional command; format and/or response may change upon further testing and development. | | |
| Allowed from / Format: | Host? Yes disc | IM? No | Group? No |
| Modifies: | Discovered by list of all IMM's on line | | |
| Requires: | IM line must be captured. | | |
| Effects: | Uses line discovery probes to generate list of all IMM's on mooring (except one executing command). | | |
| Returns: | NOT ALLOWED error if IM line is not captured. | | |
| Example: | IMM>disc <Discovery> <Discovered SN='7000020' /> <Discovered SN='7000004' /> </Discovery> <Executed/> | | |

| | | | |
|------------------------|---|-----------------------------------|-----------|
| EchoIM: | IM communications echo test | | |
| Allowed from / Format: | Host? No | IM? Yes !01echoim:text to echo | Group? No |
| Argument: | String to be echoed to IM line | | |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Echoes string back to IM line | | |
| Example: | !01echoim:this is a test <RemoteReply>echoim:this is a test</RemoteReply> <Executed/> | | |

| | | | |
|------------------------|---|--|-----------|
| Exclusive: | Requests devices other than target to go to sleep for 2 minutes Use this command when transmitting from a SIM to an IMM. | | |
| Allowed from / Format: | Host? no Use SendExclusive: instead | IM? Yes exclusive:S70000019 exclusive:01 | Group? no |
| Argument: | Device ID or serial number (prefaced with S) of target | | |
| Modifies: | DPSK Transceiver state | | |
| Requires: | Nothing | | |
| Effects: | Causes all IMMs on IM line other than addressed device to enter a low power mode for 2 minutes. This saves power in applications with long data transfers. | | |
| Example: | S>exclusive:01 EXCLUSIVE:01 ACCEPTED S> | | |

| | | | |
|--------------------------------|--|--------|-----------|
| ForceCaptureLine or FCL | Reserves IM line by transmitting a carrier signal, without first checking that no other device is using IM line. Support for FCL was added in firmware version 1.05. | | |
| Allowed from / Format: | Host? Yes forcecaptureline | IM? No | Group? No |
| Modifies: | DPSK Transceiver State | | |
| Requires: | Nothing | | |
| Effects: | Starts transmitting a carrier on IM line without first determining if another device is already transmitting. | | |
| Returns: | FAILED Error and POWER FAIL Error if transmit voltage is too low. This condition can be caused by a bad cable coupler. | | |
| Example: | forcecaptureline <Executed/> | | |

| | | | |
|------------------------|---|------------------|-------------------------|
| GData | Global command causes all IMMs to send value of GDataStr to their hosts. (legacy). Use group commands when possible in new designs: !G0:GData instead of GData. | | |
| Allowed from / Format: | Host? No | IM? Yes gdata | Group? Yes #G1:gdata |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | Global or group command – no return allowed. | | |

| | | | |
|---------------------------|---|---------------------|-----------|
| GetCD | Get Configuration Data | | |
| Allowed from / Format: | Host? Yes getcd | IM? Yes !00getcd | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <p>All parameters in Configuration Data in format:</p> <pre><ConfigurationData DeviceType='SBE90554 IMM' SerialNumber='70000047'> <Settings ConfigType='2' DebugLevel='2' BaudRate='9600' HostID='Host ID not set' GdataStr='GDATA' HostPrompt='x' ModemPrompt='imm>' DeviceID='1' EnableHostFlagWakeup='0' EnableHostFlagConfirm='0' EnableHostFlagTerm='0' EnableSerialIMMWakeup='1' EnableHostPromptConfirm='1' EnableHostServeOnPwrup='1' EnableAutoIMFlag='1' EnablePrompt='1' EnableHostWakeupCR='1' EnableHostWakeupBreak='0' EnableEcho='1' EnableSignalDetector='1' EnableToneDetect='0' EnableFullPwrTX='0' EnableBackSpace='1' EnableGDataToSample='0' EnableStripHostEcho='0' EnableBinaryData='1' SerialType='1' TermToHost='254' TermFromHost='254' SerialBreakLen='5' MaxNumSamples='40' GroupNumber='3' THOST0='0' THOST1='5' THOST2='1000' THOST3='12000' THOST4='500' THOST5='5' TMODEM2='500' TMODEM3='18000' TMODEM4='100'/'> </ConfigurationData> <Executed/></pre> | | |

| | | | |
|---------------------------|--|-----------------------------|-----------|
| GetConfigType | Get Configuration Type (1=SIM Compatibility, 2 = Standard IMM) | | |
| Allowed from / Format: | Host? Yes getconfigtype | IM? Yes !00getconfigtype | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <ConfigType T='2'/> | | |
| Returns: | <ConfigType T='1'/> | | |

| | | | |
|---------------------------|--|---------------------|-----------|
| GetEC | Get Event Counter values (nonzero entries only) | | |
| Allowed from / Format: | Host? Yes getec | IM? Yes !00getec | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <pre> IMM>getec <EventSummary numEvents='19' /> <EventList DeviceType='SBE90554 IMM' SerialNumber='70000047'> <Event type='PowerOnReset' Count='18' /> <Event type='THost0' Count='1' /> </EventList> <Executed /> </pre> <p>Definition of events (consult Sea-Bird for any events that are not listed):</p> <ul style="list-style-type: none"> • PowerOnReset – Not an error; IMM has been powered on. • WatchDogReset – IMM has been reset by watch dog timer after becoming unresponsive. • THost0 – Error or timeout when initializing host. • THost2 – Error or timeout while waiting for host to respond. • THost3 – Error or timeout while host was responding (response took too long or was stalled). • IMRxBufferOflow – While attempting to transmit from serial port (up to 38400 baud) to DPSK line (1200baud), buffer was filled and memory pointer was returned to beginning of buffer. Data previously stored in this location was permanently lost. • IMMBusy – IMM was not in idle mode (for example, in host mode) when another node attempted to communicate with it directly via IM line. DPSK section returned busy string and cleared attempted command. Only one interface, host or IM, may be active at one time. • HostCmdOverflow – While attempting to buffer incoming host commands from IM, buffer was filled. Subsequent information is not saved until buffer is emptied. • IMRxOverwrite – While attempting to buffer incoming DPSK characters, buffer was filled. Subsequent information is not saved until buffer is emptied. • IMTxCmdLen – Command length > 121 characters; command is reduced to 121 characters. • DlcmdFormat – Error occurred in discovery process. • DPcmdFormat – Error occurred in discovery process. | | |

| | | | |
|------------------------|--|---------------------|-----------|
| GetHD | Get Hardware Data | | |
| Allowed from / Format: | Host? Yes gethd | IM? Yes !00gethd | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <pre> IMM>gethd <HardwareData DeviceType='SBE90554 IMM' SerialNumber='70000047'> <Manufacturer>Sea-Bird Electronics, Inc</Manufacturer> <HardwareVersion>41420B</HardwareVersion> <HardwareVersion>PCB Type 3, 10345B</HardwareVersion> <MfgDate>May 4 2013</MfgDate> <FirmwareVersion>1.14 Jan 13 2012 16:32:44</FirmwareVersion> <FirmwareLoader>MSP LOADER RS232 57.6K 2007-02-08</FirmwareLoader> </HardwareData> <Executed/> </pre> <p>Definition of lines:</p> <ul style="list-style-type: none"> • SerialNumber - Each IMM and IM-enabled instrument has a unique serial number greater than 100 and less than 2^{32}. Serial numbers are set by Sea-Bird and cannot be changed. In Sea-Bird IM-enabled instruments (which communicate via IM-telemetry, but do not include an integrated IMM), serial number of inductive modem is serial number of instrument. • Manufacturer - Always reads <i>Sea-Bird Electronics, Inc.</i> • HardwareVersion - String identifying assembly number of IMM, or of PCB which holds IM circuit in a Sea-Bird IM-enabled instrument; at least 4 characters. • MfgDate - String identifying manufacture date; at least 6 characters. • FirmwareVersion - String identifying code version, and code date and time.. • FirmwareLoader - String identifying software used to upload firmware to IMM. | | |

| | | | |
|------------------------|---------------------------|---------------------------|-----------|
| GetHostFlag | Get HostFlag Signal value | | |
| Allowed from / Format: | Host? Yes gethostflag | IM? Yes !00gethostflag | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <HostFlag F='0'/> | | |
| Returns: | <HostFlag F='1'/> | | |

| | | | |
|------------------------|----------------------------------|-------------------------|-----------|
| GetHostID | Get HostID String | | |
| Allowed from / Format: | Host? Yes gethostid | IM? Yes !00gethostid | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <HostID>Host ID not set</HostID> | | |

| | | | |
|---------------------------|--|--------|-----------|
| GetLineStatus | Get status of IM line (captured or idle) | | |
| Allowed from / Format: | Host? Yes getlinestatus | IM? No | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <LineStatus S='CAPTURED'/> | | |
| Returns: | <LineStatus S='IDLE'/> | | |

| | | | |
|---------------------------|--|------------------------|-----------|
| GetReply | Get host response to last GData command | | |
| Allowed from / Format: | Host? No | IM? Yes !00getreply | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <GDataReply>the gdata reply</GDataReply> | | |
| Example: | <pre> IMM>!00getreply <RemoteReply><GDataReply>Not Initialized </GDataReply> <Executed/> </RemoteReply> <Executed/>` </pre> | | |

| | | | |
|---------------------------|--|---------------------|-----------|
| GetSD | Get Status Data | | |
| Allowed from / Format: | Host? Yes getsd | IM? Yes !00getsd | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | <pre> IMM>!01getsd <RemoteReply><Executing/> <StatusData DeviceType='SBE90554 IMM' SerialNumber='70000047'> <HostID>Host ID not set</HostID> <EventSummary numEvents='208'/> <Power><TransmitVoltage>7.6</TransmitVoltage></Power> <SampleDataSummary NumSamples='0' TotalLen='0' FreeMem='16384'/> <HostFileSummary Len='0' CRC='0x00000000'/> <DiscoveredBy SN='0'> <LineStatus>CAPTURED</LineStatus> </StatusData> <Executed/> </RemoteReply> <Executed/> </pre> <p>Definition of lines:</p> <ul style="list-style-type: none"> • SerialNumber – IMM serial number (> 100 and < 2³²), programmed at factory. • HostID – User-programmable 4-64 character string. • EventSummary - Number of times events such as power loss, watchdog timer restarts, invalid commands, serial errors, timeouts, firmware failures, etc. have occurred. Generic data contains a vector of pre-defined integer error codes and number of occurrences for each type. • Power - Battery voltage or other indication of state of device power supply. This may not be relevant when IMM is powered through host. • SampleDataSummary – Number of samples in memory, total amount of data in memory, and available memory in Sample Data • HostFileSummary – Total amount of data in memory and checksum for Host Data File • DiscoveredBy - Serial number of device that has identified this IMM in a line discovery process. There may be multiple instances of this tag. Tag may be removed in future firmware revisions. • LineStatus – Captured or idle | | |

| | | | |
|----------------------------|---|---------------------------------------|---|
| HostFileAppendLine: | Append line to Host Data File | | |
| Allowed from / Format: | Host? Yes hostfileappendline:data | IM? Yes !00hostfileappendline:data | Group? Yes !G1:hostfileappendline:data |
| Argument: | Data to be added to Host Data file | | |
| Modifies: | Host Data File | | |
| Requires: | Nothing | | |
| Effects: | Adds data following colon to end of Host Data File. Amount of data stored with this command is limited by length of Command Buffer. | | |
| Returns: | OVERFLOW Error if insufficient memory is available to store data. | | |
| Example: | IMM>hostfileappendline:this is a test <Executed/> | | |

| | | | |
|---------------------------|--|-----------------------------|-----------|
| HostFileErase | Erases contents of Host Data File | | |
| Allowed from / Format: | Host? Yes hostfileerase | IM? Yes !00hostfileerase | Group? No |
| Modifies: | Host Data File | | |
| Requires: | Nothing | | |
| Effects: | Erases data from Host Data File. | | |
| Returns: | <ConfirmationRequired> | | |
| Example: | IMM>hostfileerase <WARNING> <Contents of HostFile will be erased!> </WARNING> <ConfirmationRequired/> <Executed/> IMM>hostfileerase <Executed/> | | |

| | | | |
|---------------------------|---|-------------------------------|-----------|
| HostFileGetData | Retrieves Host Data File | | |
| Allowed from / Format: | Host? Yes hostfilegetdata | IM? Yes !00hostfilegetdata | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | Host Data File, enclosed in a <HostData> tag. | | |
| Example: | IMM>hostfilegetdata <HostData Len='14' CRC='0x00000000'>this is a test</HostData> <Executed/> | | |

| | | | |
|---------------------------|--|----------------------------------|-----------|
| HostFileGetSummary | Returns a summary of Host Data File | | |
| Allowed from / Format: | Host? Yes hostfilegetsummary | IM? Yes !00hostfilegetsummary | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | A <HostFileSummary> tag | | |
| Example: | IMM>hostfilegetsummary <HostFileSummary Len='14' CRC='0x00000000' /> <Executed/> | | |

| | | | |
|---------------------------|--|--------|-----------|
| HostFileWrite | Writes binary data to Host Data file | | |
| Allowed from / Format: | Host? Yes hostfilewrite | IM? No | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Waits up to 3 seconds from <Executing/> tag for first data byte before timing out. Writes data to Host Data file until a 1-second inter-character delay occurs, or Host Data file overflows. | | |
| Returns: | <Executing/> when ready to receive data | | |
| Returns: | NOT ALLOWED Error if Hostfile is not empty. Hostfile must be erased prior to writing data. | | |
| Returns: | TIMEOUT Error if no bytes arrive within 3 seconds of Executing tag. | | |
| Returns: | OVERFLOW Error if more than 4096 bytes arrive. | | |
| Example: | IMM>hostfilewrite <Executing/> this is a test <Executed/> | | |
| Example: | IMM>hostfilewrite <ERROR type='NOT ALLOWED' msg='Erase Hostfile before writing' /> <Executed/> | | |

| | | | |
|--|---|----------------|-----------|
| ID? | Returns Device ID (legacy). Use GetCD or Disc in new designs. | | |
| Allowed from / Format: | Host? Yes ³⁴ (Captured Line only) id? | IM? Yes id? | Group? No |
| This is a global command that returns data. It must not be used if more than one instrument is coupled to IM line. This command does not use reply tags. | | | |
| Modifies: | Nothing | | |
| Requires: | There must be exactly one receiving IM device on mooring. | | |
| Effects: | Returns data | | |
| Returns: | Device ID of IM device on line | | |
| Example: | IMM>id? <RemoteReply>id = 00 </RemoteReply><Executed/> | | |

| | | | |
|---------------------------|---|--------|-----------|
| IMMonitor | Used for debugging, it forwards IM traffic to serial port regardless of origin or destination. Support for IMMonitor was added in IMM firmware version 1.05. | | |
| Allowed from / Format: | Host? Yes immonitor | IM? No | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Nothing | | |
| Returns: | Any characters or noise present on IM loop regardless of origin or destination. If line is not captured, or during turnaround time from transmission to reception, serial port will output random characters. Leave IMMonitor by sending Esc character. | | |

| | | | |
|---------------------------------|---|--------|-----------|
| MeasureLineSignal or MLS | Measures incoming DPSK signal on IM line. Support for MLS was added in IMM firmware version 1.06. | | |
| Allowed from / Format: | Host? Yes measurelinesignal | IM? No | Group? No |
| Modifies: | DPSK Transceiver state | | |
| Requires: | Nothing | | |
| Effects: | Echoes string back to IM line | | |
| Returns: | <LineSignal S='48' /> <Executed/> Note: Typical value is 41-43 when device is transmitting to IM line, 47-49 when a wake-up tone is present. Value <38 or >50 indicates no signal, noise, or weak signal. | | |
| Example: | IMM>measurelinesignal <LineSignal S='42' /> <Executed/> | | |

³⁴ When the Host sends the ID? command to the IMM, the IMM sends the global ID? to the IM line. If an ID is returned, it is the ID of a remote device, not the IMM connected serially to the Host.

| | | | |
|---------------------------------|---|---------------------|-------------------------|
| PwrOff | Tell IMM to go to sleep. | | |
| Allowed from / Format: | Host? Yes pwoff | IM? Yes !00pwoff | Group? Yes !G1:pwoff |
| Modifies: | IMM operating mode | | |
| Requires: | Nothing | | |
| Effects: | Unconditionally switches IMM from Host Service or IM Service to Sleep mode after a 100mS delay. When received from Host, sends global IM command PwrOff to IM line if line is captured before powering off. | | |
| Returns: | <Executed/> | | |
| Example: (line captured) | IMM>pwoff <Executing/> <Executed/> | | |
| Example: (line not captured) | IMM>pwoff <Executed/> | | |

| | | | |
|-------------------------------|---|--------|-----------|
| ReleaseLine or REL | Releases IM line, allowing other devices to use it. Support for REL was added in IMM firmware version 1.06. | | |
| Allowed from / Format: | Host? Yes releaseline | IM? No | Group? No |
| Modifies: | DPSK Transceiver State | | |
| Requires: | Nothing | | |
| Effects: | Stops transmitting IM carrier. Automatically sends PwrOff to IM line before releasing line. | | |
| Returns: | <Executed/> | | |

| | | | |
|--|-----------------------------------|-----------------------|---------------------------|
| ResetEventCounters or ResetEC | Resets all event counters to zero | | |
| Allowed from / Format: | Host? Yes resetec | IM? Yes !00resetec | Group? Yes !G1:resetec |
| Modifies: | Event Counters | | |
| Requires: | Nothing | | |
| Effects: | Resets all event counters to zero | | |
| Returns: | <Executed/> | | |

| | | | |
|---------------------------|---|--------|-----------|
| SampleAdd | Creates a new sample in Sample Data and adds binary data to it. | | |
| Allowed from / Format: | Host? Yes sampleadd | IM? No | Group? No |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Creates a new sample, then sends <Executing/> tag to host. Adds binary data from host to new sample until a 1-second inter-character delay occurs or sample memory overflows. | | |
| Returns: | <Executing/> when ready to receive data | | |
| Returns: | TIMEOUT Error if no bytes arrive within 3 seconds of Executing tag. | | |
| Returns: | OVERFLOW Error if more bytes arrive than can be stored in Sample Data | | |
| Returns: | WARNING tag if Sample data structure contained MaxNumSamples ³⁵ before executing this command. Warning indicates that oldest Sample in memory was deleted to make room for new one. User software should avoid dependencies on contents of Warning tags. | | |
| Example: | IMM>sampleadd <Executing/> this is a test <Executed/> | | |

³⁵ MaxNumSamples is part of Configuration Data, settable to any value from 1 to 40.

| | | | |
|---------------------------|---|----------------------------------|--------------------------------------|
| SampleAddLine: | Creates a new sample in Sample Data and adds data to it | | |
| Allowed from / Format: | Host? Yes sampleaddline:data | IM? Yes !00sampleaddline:data | Group? Yes !G1:sampleaddline:data |
| Argument: | Data to write to new sample (not binary, CRLF termination is not added to sample) | | |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Creates a new sample and adds data following colon to it. Amount of data stored with this command is limited by length of Command Buffer. | | |
| Returns: | OVERFLOW Error if more bytes arrive than can be stored in Sample Data | | |
| Returns: | WARNING tag if Sample data structure contained MaxNumSamples before executing this command. Warning indicates that oldest Sample in memory was deleted to make room for new one. User software should avoid dependencies on contents of Warning tags. | | |
| Example: | IMM>sampleaddline:this is a test <Executed/> | | |

| | | | |
|---------------------------|---|--------|-----------|
| SampleAppend | Appends binary data to current sample in Sample Data | | |
| Allowed from / Format: | Host? Yes sampleappend | IM? No | Group? No |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Creates a new Sample if there are none. Sends <Executing/> tag to host. Appends binary data from host to current Sample until a 1-second inter-character delay occurs or sample memory overflows. | | |
| Returns: | <Executing/> when ready to accept data | | |
| Returns: | TIMEOUT Error if no bytes arrive within 3 seconds of Executing tag. | | |
| Returns: | OVERFLOW Error if more bytes arrive than can be stored in Sample Data | | |
| Returns: | <Executed/> when data entry terminated | | |

| | | | |
|---------------------------|--|-------------------------------------|---|
| SampleAppendLine: | Appends data to current sample in Sample Data | | |
| Allowed from / Format: | Host? Yes sampleappendline:data | IM? Yes !00sampleappendline:data | Group? Yes !G1:sampleappendline:data |
| Argument: | Data to append (not binary, CRLF termination is not added to sample) | | |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Adds data following colon to current sample. Amount of data stored with this command is limited by length of Command Buffer. | | |
| Returns: | OVERFLOW Error if more bytes arrive than can be stored in Sample Data | | |
| Example: | IMM>sampleappendline:this is a test <Executed/> | | |

| | | | |
|---------------------------|--|------------------------------|-----------|
| SampleErase: | Erases a single sample with specified SampleID; sample must be oldest sample in Sample Data | | |
| Allowed from / Format: | Host? Yes sampleerase:FF | IM? Yes !00sampleerase:FF | Group? No |
| Argument: | A hex SampleID must follow colon (up to 10 characters for a 32-bit number; format 0x0000000A, A, or 0xA, case independent). Sample ID of sample to be erased must be specified to avoid accidental deletion of samples. | | |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Erases Sample with specified SampleID if it is oldest sample in Sample Data structure. | | |
| Returns: | INVALID ARGUMENT Error if there is not a properly formatted SampleID after colon. | | |
| Returns: | INVALID ARGUMENT Error if specified SampleID does not exist. | | |
| Returns: | <Executed/> | | |

| | | | |
|---------------------------|---|-------------------------------|-----------------------------------|
| SampleEraseAll | Unconditionally erases all samples in Sample Data | | |
| Allowed from / Format: | Host? Yes sampleerasesall | IM? Yes !00sampleerasesall | Group? Yes !G1:sampleerasesall |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Erases all Samples in Sample memory. | | |
| Returns: | <Executed/> | | |

| | | | |
|-----------------------------|--|--------------------------------------|-----------|
| SampleEraseMultiple: | Erases multiple samples in Sample Data, starting with specified SampleID and including all lower SampleIDs | | |
| Allowed from / Format: | Host? Yes sampleerasemultiple:FF | IM? Yes !00sampleerasemultiple:FF | Group? No |
| Argument: | A hex SampleID must follow colon (up to 10 characters for a 32-bit number; format 0x0000000A, A, or 0xA, case independent) | | |
| Modifies: | Sample Data | | |
| Requires: | Nothing | | |
| Effects: | Erases Sample with specified SampleID, and all Samples in Sample Data with lower Sample IDs. | | |
| Returns: | INVALID ARGUMENT Error if there is not a properly formatted SampleID after colon. | | |
| Returns: | INVALID ARGUMENT Error if specified SampleID does not exist. | | |
| Returns: | <Executed/> | | |

| | | | |
|---------------------------|--|--------------------------------|-----------|
| SampleGetData: | Retrieves data of a specified sample in Sample Data | | |
| Allowed from / Format: | Host? Yes samplegetdata:FF | IM? Yes !00samplegetdata:FF | Group? No |
| Argument: | A hex SampleID must follow colon (up to 10 characters for a 32-bit number; format 0x0000000A, A, or 0xA, case independent) | | |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | INVALID ARGUMENT Error if there is not a properly formatted SampleID after colon. | | |
| Returns: | INVALID ARGUMENT Error if specified SampleID does not exist. | | |
| Returns: | Data for specified Sample in a SampleData tag | | |
| Example: | IMM>samplegetdata:22 <SampleData ID='0x22' LEN='14' CRC='0x0'>this is a test</SampleData> <Executed/> | | |

| | | | |
|---------------------------|--|-----------------------------|-----------|
| SampleGetLast | Retrieves data of most recent sample | | |
| Allowed from / Format: | Host? Yes samplegetlast | IM? Yes !00samplegetlast | Group? No |
| Argument: | None | | |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | INVALID ARGUMENT Error if there are no samples in memory | | |
| Returns: | Data for most recent sample in a SampleData tag | | |
| Example: | IMM>samplegetlast <SampleData ID='0x22' LEN='14' CRC='0x0'>this is a test</SampleData> <Executed/> | | |

| | | | |
|------------------------|--|-----------------------------|-----------|
| SampleGetList | Retrieves a list of samples in Sample Data | | |
| Allowed from / Format: | Host? Yes samplegetlist | IM? Yes !00samplegetlist | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | A SampleList tag with entries (sample ID, sample size, and check sum) for each sample in Sample Data. | | |
| Example: | <pre>IMM>samplegetlist <SampleList> <Sample ID='0x00000022' Len='14' CRC='0x00000000' /> <Sample ID='0x00000021' Len='14' CRC='0x00000000' /> </SampleList> <Executed/></pre> | | |

| | | | |
|-------------------------|---|--------------------------------|-----------|
| SampleGetSummary | Retrieves a summary of data in Sample Data | | |
| Allowed from / Format: | Host? Yes samplegetsummary | IM? Yes !00samplegetsummary | Group? No |
| Modifies: | Nothing | | |
| Requires: | Nothing | | |
| Effects: | Returns data | | |
| Returns: | A summary of data in Sample Data in a SampleDataSummary tag | | |
| Example: | <pre>IMM>samplegetsummary <SampleDataSummary NumSamples='2' TotalLen='28' FreeMem='16356' /> <Executed/></pre> | | |

| | | | |
|------------------------|-------------------------------|-------------------------|-----------------------------|
| SendBreak | Sends a serial break to host. | | |
| Allowed from / Format: | Host? No | IM? Yes !00sendbreak | Group? Yes !G1:sendbreak |
| Modifies: | Host Serial interface | | |
| Requires: | Nothing | | |
| Effects: | Sends a serial break to host | | |
| Returns: | <Executed/> | | |

| | | | |
|------------------------|---|--------|-----------|
| SendExclusive: | Requests devices other than target to go to sleep for 2 minutes (use Exclusive: command when transmitting from a SIM) | | |
| Allowed from / Format: | Host? Yes sendexclusive:01 | IM? no | Group? no |
| Argument: | Device ID or serial number (prefaced with S) of target | | |
| Modifies: | DPSK Transceiver state | | |
| Requires: | nothing | | |
| Effects: | Causes all IMM's on IM line other than addressed device to enter Sleep mode for 2 minutes. This saves power in applications with long data transfers. | | |
| Example: | <pre>IMM>sendexclusive:07 <RemoteReply>EXCLUSIVE:07 ACCEPTED </RemoteReply> <Executed/></pre> | | |
| Example: | <pre>IMM>sendexclusive:S70000047 <RemoteReply>EXCLUSIVE:S70000047 ACCEPTED </RemoteReply> <Executed/></pre> | | |

| | | | |
|------------------------|---|--------|-----------|
| SendGData | Sends global IM command GData to IM line | | |
| Allowed from / Format: | Host? Yes (Captured Line only) sendgdata | IM? No | Group? No |
| Requires: | IM line must be captured | | |
| Modifies: | IM line | | |
| Effects: | Sends global IM command GData to IM line | | |
| Returns: | NOT ALLOWED Error if IM line is not captured. | | |
| Returns: | <Executing/> <Executed/> | | |

| | | | |
|------------------------|---|--------|-----------|
| SendPwrOff | Sends global IM command PwrOff to IM line | | |
| Allowed from / Format: | Host? Yes (Captured Line only) sendpwroff | IM? No | Group? No |
| Requires: | IM line must be captured | | |
| Modifies: | IM line | | |
| Effects: | Sends global IM command PwrOff to IM line; IMM communicating with host remains awake and line remains captured. | | |
| Returns: | NOT ALLOWED Error if IM line is not captured. | | |
| Returns: | <Executing/> <Executed/> | | |

| | | | |
|------------------------|--|--------|-----------|
| SendStayOn | Sends global IM command StayOn to IM line, to reset 2-minute sleep timer for all IMM's on line. IMM communicating with host is not affected. | | |
| Allowed from / Format: | Host? Yes (Captured Line only) sendstayon | IM? No | Group? No |
| Requires: | IM line must be captured | | |
| Modifies: | IM line | | |
| Effects: | Sends global IM command StayOn to IM line | | |
| Returns: | NOT ALLOWED Error if IM line is not captured. | | |
| Returns: | <Executing/> <Executed/> | | |

| | | | |
|------------------------------|--|--------|-----------|
| SendWakeupTone or SWT | Sends a 4-second wake-up tone to IM line. Support for SWT was added in IMM firmware revision 1.05. | | |
| Allowed from / Format: | Host? Yes (Captured Line only) sendwakeuptone | IM? No | Group? No |
| Requires: | IM line must be captured | | |
| Modifies: | IM line | | |
| Effects: | Sends a 4-second wake-up tone to IM line | | |
| Returns: | NOT ALLOWED Error if IM line is not captured. | | |
| Returns: | <Executing/> about once per second, followed by <Executed/> when finished. | | |
| Example: | IMM>sendwakeuptone <Executing/> <Executing/> <Executing/> <Executed/> | | |

| | |
|---------------------|--|
| Set commands | See <i>Config Type 2 Set Commands</i> below. |
|---------------------|--|

| | | | |
|---------------------------|--|----------------------|--------------------------|
| StayOn | Resets IMM's 2-minute sleep timer | | |
| Allowed from / Format: | Host? Yes stayon | IM? Yes !00stayon | Group? Yes !G1:stayon |
| Requires: | Nothing | | |
| Modifies: | 2-minute sleep timer | | |
| Effects: | Resets IMM's 2 minute sleep timer, keeping device awake. | | |

| | | | |
|---------------------------------------|---|---------------------------------|-----------|
| Test20TurnCoupler or T20CC | Tests integrity of IM cable coupler. For 20-turn couplers only. Support for T20CC was added in IMM firmware revision 1.05. | | |
| Allowed from / Format: | Host? Yes (Captured Line only) test20turncoupler | IM? Yes !00test20turncoupler | Group? No |
| Requires: | IM line must be captured if command sent from host | | |
| Requires: | This command requires a 20-turn coupler, or test results will be misleading. | | |
| Modifies: | IM line | | |
| Effects: | Performs a test of IM cable coupler | | |
| NOTE: | When Test20TurnCoupler command is received from IM line, several characters of noise will be transmitted while test is in-process. See example in TestCableCoupler command description below. This noise is an unavoidable result of method of testing coupler. Noise does not occur when command is received in Host Service mode. | | |
| Returns: | NOT ALLOWED Error if IM line is not captured when command is received from host. | | |
| Returns: | FAILED Error and POWER FAIL Error if transmit voltage is low. | | |
| Returns: | FAILED Error if coupler connection is poor or coupler is not connected. | | |
| Returns: | <CableCouplerTest status='GOOD MEDIUM BAD' Z='UINT16'/> if test is performed | | |

| | | | |
|------------------------------------|--|--------------------------------|-----------|
| TestCableCoupler or TCC | Tests integrity of IM cable coupler. For 100-turn couplers only. Support for TCC was added in IMM firmware revision 1.05. | | |
| Allowed from / Format: | Host? Yes (Captured Line only) testcablecoupler | IM? Yes !00testcablecoupler | Group? No |
| Requires: | IM line must be captured if command sent from host | | |
| Requires: | This command requires a 100-turn coupler, or test results will be misleading. | | |
| Modifies: | IM line | | |
| Effects: | Performs a test of IM cable coupler | | |
| NOTE: | When testcablecoupler command is received from IM line, several characters of noise will be transmitted while test is in-process. See example below. This noise is an unavoidable result of method of testing coupler. Noise does not occur when command is received in Host Service mode. | | |
| Returns: | NOT ALLOWED Error if IM line is not captured when command is received from host. | | |
| Returns: | FAILED Error and POWER FAIL Error if transmit voltage is low. | | |
| Returns: | FAILED Error if coupler connection is poor or coupler is not connected. | | |
| Returns: | <CableCouplerTest status='GOOD MEDIUM BAD' Z='UINT16'/> if test is performed | | |
| Example: | IMM>testcablecoupler <CableCouplerTest status='GOOD' Z='1485' /> <Executed/> | | |
| Example: | IMM>testcablecoupler <ERROR type='NOT ALLOWED' msg='IM Line Not Captured' /> <Executed/> | | |
| Example: | IMM>testcablecoupler <ERROR type='FAILED' msg='Low Transmit Voltage - low battery or bad coupler' /> <ERROR type='POWER FAIL' msg='Transmit Voltage Vtx=0.0 ' /> IMM> | | |
| Example | !00TestCableCoupler <RemoteReply><Executing/>Ðÿ!#ÃÿÿÄÿÖ¿ÿ„ÿ (note noise) àÿ<CableCouplerTest status='GOOD' Z='2405' /> (note noise) <Executed/> </RemoteReply> <Executed/> | | |

| | | | |
|------------------------------|---|-------------------------|-----------|
| TestEEPROM | Tests internal EEPROM for errors. | | |
| Allowed from / Format: | Host? Yes testeprom | IM? Yes !00testeprom | Group? No |
| Requires: | Nothing | | |
| Modifies: | Nothing | | |
| Effects: | Tests internal EEPROM for errors. | | |
| Returns: | <Executing/> about once per second. | | |
| Returns: | One or more EEPROM FAILURE Errors if test does not pass | | |
| Returns (when completed): | <Executed/> | | |

| | | | |
|---------------------------|--|----------------------|-----------|
| TXTest | Sends a series of tests of IM communications | | |
| Allowed from / Format: | Host? No | IM? Yes !00txtest | Group? No |
| Requires: | Nothing | | |
| Modifies: | Nothing | | |
| Effects: | Sends 10 seconds of silence, followed by 250 <i>U</i> characters at maximum speed, followed by 250 <i>U</i> characters at pseudo-random intervals. | | |

11.6 Config Type 2 Set Commands

In Config Type 2, all commands that change the Configuration Data begin with the keyword *Set* and end with = followed by some argument. All *Set* command arguments are tested for minimum and maximum values. If the argument is a string, it is tested for minimum and maximum length. If the command contains the keyword *Enable* then the argument must be 1 or 0. If the argument is not valid, the IMM replies with *INVALID ARGUMENT Error*. Some parameters respond with the <ConfirmationRequired/> tag, indicating the command must be repeated to take effect. The IMM always sends the <Executed/> tag, even when the parameter is not changed; the Executed tag indicates the processing of the command is complete – it does not indicate that the command had any effect.

SetBaudRate=x

x= Baud rate for host communication (1200, 2400, 4800, 9600, 19200³⁶). Returns ConfirmationRequired tag and Warning tag. The IMM goes to sleep after this command is accepted. When it is woken up again, it uses the new baud rate.

SetBaudRate= is allowed from the host only.

Factory default is 9600. Baud rate is not affected by a change in Interface mode.

SetConfigType=x

x= 1: Configuration Type 1 (SIM Compatibility).

x= 2: Configuration Type 2 (default).

Returns ConfirmationRequired tag and Warning tag. This command should not be used after deployment.

SetConfigType= is allowed from the host only.

Factory default is 2 (Config Type 2). Config Type is not affected by a change in Interface mode.

SetDebugLevel=x

x= 0: Disable all executed and executing tags, and remote reply tags.

x= 1: Disable all remote reply tags.

x= 2: Normal operation.

x= 3: For debugging with a terminal program. This enables some extra text output <!--in XML comment tags--> in response to many commands. The existence and content of comment tags is not part of the IMM specification - IMM applications should have no dependency on the existence or content of XML comments.

x=4 – 9: Factory use only.

SetDebugLevel= is allowed from the host, the IM, and as a group command.

Factory default is DebugLevel=2 (normal operation). DebugLevel is not affected by a change in Interface mode.

Note: THost4 timeouts are reported as Errors only if DebugLevel is set to 6 or higher.

SetDeviceID=x

x= Device ID (00 – 99), which is the *ii* in !*ii*COMMAND or #*ii*COMMAND.

SetDeviceID= is allowed from the host and the IM.

Factory default is 0. Device ID is not affected by a change in Interface mode. Two digits are required for *ii* in #*ii* or !*ii* commands. However, the SetDeviceID= command does not require two digits (For example, !00GetSD gets the status data of remote device with ID =00 and !0GetSD returns an error. However, SetDeviceID=01 has the same effect as SetDeviceID=1).

SetEnableAutoIMFlag=x

x= 1: IMM automatically controls the IMFlag digital output to wake up the host, confirm the IMM wakeup, and terminate commands and replies.

x= 0: Disable this feature. When set to 0, the only way to change the state of IMFlag is with the SetIMFlag= command.

SetEnableAutoIMFlag= is allowed from the host, the IM, and as a group command.

Default is 1 for all Interface modes.

³⁶ Support for 38400 baud was removed in firmware version 1.05.

SetEnableBackspace=x

x= 1: Interpret the backspace character as a command to delete the previously received character when received from the host. This feature applies only to host commands, not to host replies to IM commands. This feature is extremely useful when the host is a terminal program on a PC.

x= 0: Treat the backspace character as a normal character with no deletion.

SetEnableBackSpace= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1 through 6 and 8 through 13; default is 1 for Interface modes 7 and 14.

SetEnableBinaryData=x

x= 1: Enable the IMM-specific translation of data to the IM line.

x= 0: Disable IMM-specific translation. Use this setting when using the IMM with older IM-enabled instruments that do not support translation, such as the SBE 16*plus*-IM, 16*plus*-IM V2, 37-IM, 37-IMP (firmware < 4.0), 39-IM, 44, and SIM. Older devices will not properly receive < > : \n * @ ! and # if binary data is enabled on the transmitting IMM.

SetEnableBinaryData= is allowed from the host, the IM, and as a group command.

Default is 1 for all Interface modes.

SetEnableEcho=x

x= 1: IMM echoes all characters received from the host back to the host.

x= 0: Disable the echo.

SetEnableEcho= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1 through 6 and 8 through 13; default is 1 for Interface modes 7 and 14.

SetEnableFullPwrTX=x

x= 1: IMM transmits with the lesser of 9V or the input voltage. The higher power transmit option increases signal strength by as much as a factor of 3, and increases power consumption during transmission to 50mW at 10V (with 100-turn coupler).

x= 0: Use low power transmit mode with transmitting power consumption of 10mW at 10V.

SetEnableFullPwrTX= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

SetEnableGDataToSample=x

x= 1: IMM stores host response to GData command as a sample in Sample Data instead of storing the response in the GData Reply Buffer. This removes the 127 byte limitation on the host reply.

x= 0: IMM stores host response to GData command in the 127 byte GData Reply Buffer.

SetEnableGDataToSample= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

SetEnableHostFlagConfirm=x

x= 1: IMM interprets a low-high transition on the HostFlag signal to confirm wake-up of the host.

x= 0: Disable this confirmation.

SetEnableHostFlagConfirm= is allowed from the host and the IM.

Default is 0 for Interface modes 2, 4, 6, 7, 9, 11, 13, and 14; default is 1 for Interface modes 1, 3, 5, 8, 10, and 12.

SetEnableHostFlagTerm=x

x= 1: IMM interprets a high-low transition on the HostFlag signal to terminate a transmission from the host.

x= 0: Disable this termination.

SetEnableHostFlagTerm= is allowed from the host and the IM.

Default is 0 for Interface modes 2, 4, 6, 7, 9, 11, 13, and 14; default is 1 for Interface modes 1, 3, 5, 8, 10, and 12.

SetEnableHostFlagWakeup=x

x= 1: IMM wakes up in response to low-high transitions on the HostFlag digital input. Also, IMM wakes up in response to a static logic high condition on the HostFlag signal when it is about to enter sleep, including after receiving commands from the IM line with non-matching address.

x= 0: Disable this wake up.

SetEnableHostFlagWakeup= is allowed from the host and the IM.

Default is 0 for Interface modes 1, 2, 4, 6, 7, 8, 9, 11, 13, and 14; default is 1 for Interface modes 3, 5, 10, and 12.

SetEnableHostPromptConfirm=x

x= 1: IMM interprets reception of a prompt from the host as confirmation of the host wake-up.

x= 0: Disable this confirmation.

SetEnableHostPromptConfirm= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1 through 6 and 8 through 13; default is 1 for Interface modes 7 and 14.

SetEnableHostServeOnPwrUp=x

x= 1: IMM enters Host Service mode when power is turned on (or when the IMM resets for any other reason).

x= 0: Disable this feature.

SetEnableHostServeOnPwrUp= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1, 2, 3, 5, 6, 7, 8, 9, 10, 12, 13, and 14; default is 1 for Interface modes 4 and 11.

SetEnableHostWakeupBreak=x

x= 1: IMM sends a serial break condition to the host to wake-up the host when a command is received from the IM line. The SerialBreakLen setting controls the duration of the break condition. This feature is independent of IMFlag settings.

x= 0: Disable this feature.

SetEnableHostWakeupBreak= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

SetEnableHostWakeupCR=x

x= 1: IMM sends a character to the host to wake-up the host when a command is received from the IM line. The IMM sends a '\r' to the host to generate the wakeup. This feature is independent of IMFlag settings.

x= 0: Disable this feature.

SetEnableHostWakeupCR= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1 through 5 and 8 through 12; default is 1 for Interface modes 6, 7, 13, and 14.

SetEnablePrompt=x

x= 1: IMM sends the ModemPrompt to the host when it powers up, when it wakes up in response to HostFlag or Host Serial transitions, after processing each host command (except for the PwrOff command), and after confirming host wake-up when relaying IM commands.

x= 0: Disable this prompt.

SetEnablePrompt= is allowed from the host, the IM, and as a group command.

Default is 0 for Interface modes 1 through 6 and 8 through 13; default is 1 for Interface modes 7 and 14.

SetEnableSerialIMMWakeup=x

x= 1: IMM wakes up in response to high-low transitions on the serial input.

x= 0: Disable this wake up.

Returns ConfirmationRequired tag and Warning tag if set to 0.

SetEnableSerialIMMWakeup= is allowed from the host and the IM.

Default is 1 for all Interface modes.

Note: EnableSerialIMMWakeup cannot be set to 0 if EnableSignalDetector is set to 0. This prevents the IMM from reaching a state from which it cannot wake up.

SetEnableSignalDetector=x

x= 1: Enable the IM signal detector. The IMM wakes up in response to commands on the IM line.

x= 0: Disable the signal detector.

Returns ConfirmationRequired tag and Warning tag if set to 0.

SetEnableSignalDetector= is allowed from the host only.

Default is 1 for Interface modes 1, 2, 5, 6, 7, 8, 9, 12, 13, and 14; default is 0 for Interface modes 3, 4, 10, and 11.

Note: EnableSignalDetector cannot be set to 0 if EnableSerialIMMWakeup is set to 0. This prevents the IMM from reaching a state from which it cannot wake up.

SetEnableStripHostEcho=x

x= 1: The IMM in IM Service mode automatically removes the host's echo of the relayed command, if there is an echo.

x= 0: The IMM does not strip any characters from the host reply.

SetEnableStripHostEcho= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

SetEnableToneDetect=x

x= 1: IMM wakes the host when an incoming wake-up tone is detected.

x= 0: Disable IM notification of incoming wake-up tones.

SetEnableToneDetect= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

Example:

Wake-up tone is enabled in IMM1, IMM1 put to sleep.

```
IMM>setenabletonedetect=1
<Executed/>
IMM>pwroff
<Executed/>
```

Another IMM (2) sends a wake-up tone to the IM line:

```
IMM>sendwakeuptone
<Executing/>
<Executing/>
<Executing/>
<Executing/>
<Executed/>
```

While wake-up tone is sent from IMM2, IMM1 sends this to host:

```
IMM>WAKE-UP TONE DETECTED
IMM>WAKE-UP TONE DETECTED
IMM>WAKE-UP TONE DETECTED
[... continues to signal until tone stops...]
IMM>WAKE-UP TONE DETECTED
```

Note: Wake-up tone detection is only usable in configuration types where SetEnableSignalDetector can be set. This command is not a replacement for the SIM Tone Detect board when the IMM is in ConfigType 1.

SetGDataStr=x

x= GDataStr value (1 - 32 characters), which is sent to the host in response to the GData command.

SetGDataStr= is allowed from the host, the IM, and as a group command.

Factory default is GDATA. The GData string is not affected by a change in Interface mode.

SetGroupNumber=x

x= Group membership of the IMM (1 – 9, or 0). Group 0 is pre-defined as the group of all instruments; a 0 entry for this parameter indicates the IMM is not a member of any other group. Individual devices cannot belong to more than two groups – group 0 plus one additional group.

SetGroupNumber= is allowed from the host, the IM, and as a group command.

Factory default is Group 0. Group membership is not affected by a change in Interface mode.

SetHostID=x

x= Host ID string (4 – 64 characters).

SetHostID= is allowed from the host, the IM, and as a group command.

Default is *Host ID not set* for all Interface modes. The Host ID is not affected by a change in Interface mode.

SetHostPrompt=x

x= Prompt string expected from the host to confirm wake-up (1 – 7 characters). Host prompts are not case sensitive. For example, if the hostprompt is set to *s>* and the instrument sends *S>*, the prompt is accepted.

SetHostPrompt= is allowed from the host, the IM, and as a group command.

Default is *S>* for Interface modes 1 through 6 and 8 through 13; default is *x* for Interface modes 7 and 14.

SetID=x

Same as SetDeviceID=.

SetIMFlag=x

x=1: Deactivate the pull-down on the IMFlag output line.

x=0: Activate the pull-down on the IMFlag output line.

SetIMFlag= is allowed from the host, the IM, and as a group command.

NOTE: This is not a setting in Configuration Data - this changes the state of the physical IO line. If EnableAutoIMFlag=1, the IMFlag signal will be released a few milliseconds after parsing the command.

SetInterfaceMode=x

Note: This command changes many parameters in Configuration Data. Check the desired Interface mode carefully before sending this command. Confirmation is required. The IMM goes to sleep after this command is accepted. When it is woken up again, it uses the new Interface mode.

The following parameters are not changed by the SetInterfaceMode command: BaudRate, ConfigType, DebugLevel, DeviceID, GDataStr, GroupNumber, and HostId.

Returns ConfirmationRequired tag and Warning tag.

SetInterfaceMode= is allowed from the host only.

Default Interface mode (from the factory) is 7.

Note: For Versions < 1.0, this command was SetConfigMode=.

Example:

```
IMM>setinterfacemode=7
<WARNING>
IMM will power down
next power up after confirm will use
new Configuration Data settings
with RS232 Serial interface
</WARNING>
<ConfirmationRequired/>
<Executed/>
IMM>setinterfacemode=7
<Executed/>
```

SetMaxNumSamples=x

x= Maximum number of samples allowed in Sample Data (1 – 40). If the maximum is set to a number smaller than the number of samples currently in memory, each SampleAdd or SampleAddLine: deletes only one old sample before creating a new sample, leaving the number of samples in memory greater than the specified maximum.

SetMaxNumSamples= is allowed from the host, the IM, and as a group command.

Default is 40 for all Interface modes.

SetModemPrompt=x

x= Prompt sent by the IMM to the host to cause a wake-up (1 – 7 characters).

SetModemPrompt= is allowed from the host, the IM, and as a group command.

Default is IMM> for all Interface modes.

SetSerialBreakLen=x

x= Length of a serial break in tens of milliseconds (1 – 255, corresponding to 10 milliseconds to 2.55 seconds).

Not affected by the baud rate.

SetSerialBreakLen= is allowed from the host, the IM, and as a group command.

Default is 5 for all Interface modes.

SetSerialType=x

x=0: Logic Level Serial

x=1: RS-232.

Returns ConfirmationRequired tag and Warning tag.

SetSerialType= is allowed from the host and the IM.

Default is 1 (RS-232) for Interface modes 1-7, 0 for 8-14.

SetTermFromHost=x

x= Termination character for host replies to the IMM in IM Service mode (decimal number).

TermFromHost=254 is carriage return line feed ('\r\n') and 255 is no termination character. 251, 252, and 253 are reserved. A serial break condition cannot be used for termination from the host. Note: TermFromHost does not apply in Host Service mode! In Host Service mode the termination sequence is always '\r\n'. Note that termination characters from the host are included in IM replies.

SetTermFromHost= is allowed from the host, the IM, and as a group command.

Default is 254 decimal (ASCII '\r\n') for Interface modes 1, 3-8, and 10-14; default is 62 decimal (ASCII '>') for Interface modes 2 and 9.

SetTermToHost=x

x= Termination character for transmissions from the IMM to the host (decimal number from 0 – 250). TermToHost =253 is a serial line break, 254 is carriage return line feed ('\r\n'), and 255 is no termination character. 251 and 252 are reserved.

SetTermToHost= is allowed from the host, the IM, and as a group command.

Default is 13 decimal (ASCII '\r') for Interface modes 1 through 6 and 8 through 13; default is 254 decimal (ASCII '\r\n') for Interface modes 7 and 14.

SetTHost0=x

x= THost0 maximum wait for wake-up confirmation (0 - 1000; units are tens of milliseconds; THost0=100 means 1 second). Described in detail in *Section 8.6: Timeout Parameters in IM Service Mode*.

SetTHost0= is allowed from the host, the IM, and as a group command.

Default is 0 for all Interface modes.

SetTHost1=x

x= THost1 delay after confirm (0 - 300; units are tens of milliseconds; THost1=100 means 1 second). Described in detail in *Section 8.6 Timeout Parameters in IM Service Mode*.

SetTHost1 is allowed from the host, the IM, and as a group command.

Default is 5 for all Interface modes.

SetTHost2=x

x= THost2 maximum wait for start of reply (0, or 100 - 3000; units are tens of milliseconds; THost2=100 means 1 second). Described in detail in *Section 8.6 Timeout Parameters in IM Service Mode*.

SetTHost2= is allowed from the host, the IM, and as a group command.

Default is 3000 for Interface modes 1, 3, 5, 8, 10, and 12; default is 1000 for Interface modes 2, 4, 6, 7, 9, 11, 13, and 14.

SetTHost3=x

x= THost3 IM reply transmission time (100 - 18000; units are tens of milliseconds; THost3=100 means 1 second).

Described in detail in *Section 8.6 Timeout Parameters in IM Service Mode*.

SetTHost3= is allowed from the host, the IM, and as a group command.

Default is 12000 for all Interface modes.

SetTHost4=x

x= THost4 inter-character delay timeout (5 - 3000); units are tens of milliseconds; THost4=100 means 1 second).

Described in detail in *Section 8.6 Timeout Parameters in IM Service Mode*.

SetTHost4= is allowed from the host, the IM, and as a group command.

Default is 500 for all Interface modes.

SetTHost5=x

x= THost5 IMFlag hold after reply termination. (5 - 3000); units are tens of milliseconds; THost4=100 means 1 second).

Described in detail in *Section 8.6 Timeout Parameters in IM Service Mode*.

SetTHost5= is allowed from the host, the IM, and as a group command.

Default is 5 for all Interface modes.

SetTModem2=x

x= TModem2 maximum wait for start of IM reply (0 - 3000); units are tens of milliseconds; TModem2=100 means 1 second). Described in detail in *Section 5.5 Timeout Parameters in Host Service Mode*.

SetTModem2= is allowed from the host, the IM, and as a group command.

Default is 500 for all Interface modes.

SetTModem3=x

x= TModem3 maximum IM receive time (0 - 60000); units are tens of milliseconds; TModem3=100 means 1 second).

Described in detail in *Section 5.5 Timeout Parameters in Host Service Mode*.

SetTModem3= is allowed from the host, the IM, and as a group command.

Default is 18000 for all Interface modes.

SetTModem4=x

x= TModem4 Configuration Type 1 maximum IM receive inter-character delay before terminating reception of reply to bii command (0 – 3000; units are tens of milliseconds; TModem4=100 means 1 second). Described in detail in *Section 5.5 Timeout Parameters in Host Service Mode*. This setting exists in IMM firmware V1.02 and later.

SetTModem4= is allowed from the host, the IM, and as a group command.

Default is 100 for all Interface modes.

12 Interface Mode Shortcuts

Interface modes act as shortcuts for common settings in Configuration Data. Setting an Interface mode changes the Configuration Data defaults to support a particular style of host interface, as defined below. There are seven basic interface modes, but separate numbers are assigned for the same configuration in RS-232 and in Logic Level Serial (LLS):

| Interface Mode Number | Description |
|-------------------------|---|
| 1 (RS-232) and 8 (LLS) | Hardware handshaking with simple host instrument |
| 2 (RS-232) and 9 (LLS) | Serial handshaking with simple host instrument |
| 3 (RS-232) and 10 (LLS) | Hardware handshaking with simple buoy controller |
| 4 (RS-232) and 11 (LLS) | Serial handshaking with simple buoy controller |
| 5 (RS-232) and 12 (LLS) | Hardware handshaking with symmetric communication |
| 6 (RS-232) and 13 (LLS) | Serial handshaking with symmetric communication |
| 7 (RS-232) and 14 (LLS) | Terminal program with symmetric communication |

Note: Interface mode 7 is the default Interface mode.

The following are **not** affected by Interface mode changes; if you change the Interface mode, they remain set to the previously set value:

| Setting and Default | Setting and Default | Setting and Default | Setting and Default |
|---------------------|---------------------|---------------------|---------------------|
| BaudRate - 9600 | ConfigType - 2 | DebugLevel - 2 | DeviceID - 00 |
| GdataStr - GDATA | GroupNumber - 0 | HostID - not set | |

The settings for the following parameters can be modified by simply changing the Interface mode (**SetInterfaceMode=**):

| Parameter | Interface Mode | | | | | | |
|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|--------------------------|
| | 1 & 8 | 2 & 9 | 3 & 10 | 4 & 11 | 5 & 12 | 6 & 13 | 7 & 14 |
| EnableAutoIMFlag | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EnableBackspace | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| EnableBinaryData | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EnableEcho | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| EnableFullPwrTX | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EnableGDataToSample | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EnableHostFlagConfirm | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| EnableHostFlagTerm | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| EnableHostFlagWakeup | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| EnableHostPromptConfirm | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| EnableHostServeOnPwrUp | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| EnableHostWakeupBreak | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EnableHostWakeupCR | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| EnablePrompt | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| EnableSerialIMMWakeup | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| EnableSignalDetector | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| EnableStripHostEcho | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| EnableToneDetect | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| HostPrompt | S> | S> | S> | S> | S> | S> | x |
| MaxNumSamples | 40 | 40 | 40 | 40 | 40 | 40 | 40 |
| ModemPrompt | IMM> |
| SerialBreakLen | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| SerialType | 1 (RS-232) or 0 (LLS) |
| TermFromHost (decimal) * | 254 | 62 | 254 | 254 | 254 | 254 | 254 |
| TermToHost (decimal) * | 13 | 13 | 13 | 13 | 13 | 13 | 254 |
| THost0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| THost1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| THost2 | 3000 | 1000 | 3000 | 1000 | 3000 | 1000 | 1000 |
| THost3 | 12000 | 12000 | 12000 | 12000 | 12000 | 12000 | 12000 |
| THost4 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |
| THost5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| TModem2 | 500 | 500 | 500 | 500 | 500 | 500 | 500 |
| TModem3 | 18000 | 18000 | 18000 | 18000 | 18000 | 18000 | 18000 |
| TModem4 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |

* 13 decimal = ASCII '\r', 62 decimal = ASCII '>', 254 decimal = ASCII '\r\n'

13 Electrical Specification

13.1 IMM Connector Description

Connector Option 1. Male vertical for PCB to PCB connection (recommended)

Mating connectors: Through hole - Hirose #DF11-10DS-2DSA(01), DigiKey #H2822-ND
Surface mount - Hirose #DF11-10DS-2V(50), DigiKey #H2837-ND

Connector Option 2. Male Vertical for cable connection

Mating connectors: 10-pin header - Hirose #DF11-10DS-2C, DigiKey #H2023-ND
4-pin header - Hirose #DF11-4DS-2C, DigiKey #H2019-ND

Crimp contacts (24 to 28 AWG) - Hirose #DF11-2428SCA, DigiKey #H1505-ND

Crimp tool - Hirose #DF11-TA2428HC/US, Digikey #H9924-ND

Pre-crimped wires - DigiKey #H3BBG-10112-B4-ND

Connector Option 3. Male right angle for cable connection

Mating connectors: same as Connector Option 2.

Table 13.1.1 J1 Pinout RS-232 / Logic Level Serial

| Pin # | Signal | Direction | Description |
|-------|-------------|-----------|--|
| 1 | IM Coil + | -- | Positive side of IM coil (connect to center if using coax) |
| 2 | IM Coil - | -- | Negative side of IM coil (connect to shield if using coax) |
| 3 | HostFlag | In | Hardware Handshake signal. 1.5V to 12V input with internal 4.7K series resistor, 100K pulldown resistor, 10nF load and 12Volt TVS. |
| 4 | IMFlag | Out | Hardware Handshake signal. Pulldown only. 100ohm series resistor and 330pF load. Static protected with 12V TVS. |
| 5 | LLS_Rx | In | Logic Level Serial receive data input |
| 6 | RS-232Rx | In | RS-232 Serial receive data input |
| 7 | LLS_Tx | Out | Logic Level Serial transmit data output. Open collector with 100 ohm series resistor and 12Volt TVS |
| 8 | RS-232_Tx | Out | RS-232 Serial transmit output. |
| 9 | Vin + | -- | Positive Power Input |
| 10 | Vin - / GND | -- | Power Supply Return / Ground |

Table 13.1.2 J2 Pinout

| Pin # | Signal | Description |
|-------|-----------|--|
| 1 | IM Coil + | Positive side of IM coil (connect to center if using coax) |
| 2 | IM Coil - | Negative side of IM coil (connect to shield if using coax) |
| 3 | GND | Power supply return (IM coil shield if using twinax cable) |
| 4 | Vin + | Power supply 6.0 - 30VDC |

13.2 Hardware I/O Line Characteristics

The HostFlag input is triggered on the low-high transition, with one exception. When EnableHostFlagWakeup=1, the IMM tests the state of the HostFlag line just prior to entering Sleep mode. If HostFlag is high, the IMM immediately switches to Host Service instead of going to sleep. If EnableHostFlagWakeup=0, the HostFlag input must be low or floating for the IMM to go to sleep.

Table 13.2.1 Electrical Characteristics

| Parameter | Minimum | Typical | Maximum |
|------------------------------------|-------------|----------|---------|
| HostFlag Input, logic low | NA | | 2.0V |
| HostFlag Input Pulse width | 200uS | | |
| IMFlag Resistance to GND; active | | 100 ohms | |
| IMFlag Resistance to GND; inactive | 10 megaohms | | |

13.3 Absolute Maximum Ratings

Table 13.3.1 Minimum and Maximum Voltages

| Parameter | Minimum | Maximum |
|--|---------|---------|
| Power Supply Input Voltage | -30V | 30 |
| HostFlag Input Voltage to GND | -10V | +30V |
| IMFlag pulldown current | | 30mA |
| IMFlag Output Voltage to GND, inactive | -0.5V | +12V |
| Serial Inputs | -12V | +12V |
| Logic Level Serial input | -12V | +5V |

13.4 Power Supply Input

- **Standard Operation** - The IMM power supply is optimized for direct connection to batteries from 9VDC to 24VDC with potential voltage spikes to 30VDC.
- **High Input Voltage Operation** - A High voltage input adapter board is available for applications with input voltages between 20 VDC and 80 VDC. This board holds both an IMM and a high efficiency power converter.

Continuous operation of the IMM at high input voltages without the adapter board is possible at significantly reduced efficiency with the addition of an external series resistor on the power supply input. The series resistor protects the IMM power system from a worst case operating current of approximately 33mA³⁷. The series resistor value and wattage rating depend on the maximum input voltage.

$R_{series, min} = 31 * (V_{in,max} - 22)$ ohms (for example, 42V in => 620 ohms)

$V_{in} < 27V \Rightarrow 0.125$ Watt resistor rating³⁸

$V_{in} < 35V \Rightarrow 0.25$ Watt

$V_{in} < 42V \Rightarrow 0.5$ Watt

$V_{in} > 42V$ - not recommended, because the efficiency falls below 40%.

³⁷ A linear regulator activates in the IMM when the input voltage rises above 19V. The pass element in this regulator should not dissipate more than 150mW. At 35mA the pass element is safe to 24V. For optimum protection, the series resistor must reduce the input voltage below 24V at an input current of 30 mA.

³⁸ These numbers are for standard axial leaded resistors wrapped in heatshrink, assuming operation in the worst case failure mode for 2 minutes. Continuous operation in the worst case failure mode requires double the series resistor wattage ratings listed.

13.5 Power Consumption Estimates

Power consumption is highly dependent on setup. When connected to a PC with RS-232, the PC's RS-232 load resistors consume more power than the entire IMM. When connected to a low-load RS-232 receiver or with logic-level serial, the device is much more efficient.

Table 13.5.1 Power Consumption Measurements (Firmware Version 0.98)

| Mode | V _{in} =10V |
|---|----------------------|
| Asleep (Signal Detector Enabled) ³⁹ | 25uA |
| IM Receive | 0.65mA |
| Host Service (RS-232, no load) | 0.9 mA |
| Host Service (RS-232 with PC standard load) | 2.15mA |
| Host Service, Captured line (transmitting) (RS-232, no load, low power tx, 100-turn coupler) | 1.42mA |
| IM Service, during reply to # command (RS-232, no load, low power tx, 100-turn coupler) | 1.42mA |

Pessimistic Estimates for an IMM with logic level serial, 12V battery input:

- Total for 1 year, listening to IM line with no activity = 0.3 amp-hours
- Total for 1 year at 1 sample per hour, 2-minute receive per sample = 0.42 amp-hours
- Total for 1 year at 6 samples per hour, 2-minute receive per sample = 1.2 amp-hours

³⁹ Some early hardware versions draw 33uA in Sleep mode. These devices have serial numbers <70000050.

14 Manual Revision History

| Manual Version | Date | Description |
|----------------|-------|---|
| 011 | 07/11 | <ul style="list-style-type: none">• First official release of manual. |
| 012 | 05/13 | <ul style="list-style-type: none">• Add Declaration of Conformity (CE certificate).• Add CE mark on cover.• Update photo on cover.• Add event counter code definitions.• Firmware 1.13 and 1.14 updates:<ul style="list-style-type: none">- Make ERROR all upper case in XML error messages <ERROR type= . . . >.- Commands SampleGetData, SampleErase, and SampleEraseMultiple now accept arguments in following (case independent) formats: 0x0000000A, A, 0xA.- Update GetHD response.- Update firmware revision history (Section 15). |
| 013 | 03/21 | <ul style="list-style-type: none">• Remove references to RS485. No longer offered. |

15 Firmware Revision History

| Firmware Version | Date | Description |
|------------------|-------|---|
| 1.01 | | <ul style="list-style-type: none"> Serial Type not preserved when switching from Config Type 1 to Type 2. Start of reply marker detection disabled for Config Type 1 'B' commands for binary response. This legacy command was used with UIM; UIM does not send start of reply code with 'B' command. |
| 1.02 | | <ul style="list-style-type: none"> Modified timing of IM replies in Config Type 1 'bii' binary commands. This should not cause any performance change. Removed inappropriate TIMEOUT error returned after Config Type 1 'bii' binary commands with normal inter-character delay termination. TModem3 maximum value changed to 600 seconds. Added TModem4 setting for Config Type 1 use. Config Type 1 bii command inter-character delay timeout now activates after reception of two characters instead of one. Fixed inappropriate logging of Error37 in Config Type 1. Config Type 1 blocked commands now return NOT ALLOWED error. Config Type 1 resets timeout timer on CR or LF, instead of just LF. |
| 1.03 | | <ul style="list-style-type: none"> All changes of IMFLAG conditional on EnableAutoIMFlag setting. Added error for invalid argument on SetIMFlag=. Warning added to hostfileerase command. Host service 2 min timeout changed to : <TIMEOUT msg='HostService 2 min timeout'/>. Event ERROR_IM_CMD_MODE removed. Event ERROR_IMM_BUSY added when IMM sends a <busy/> signal. CRLF from host acts as recognized command, returning <Executed/> and resetting host timeout instead of just giving a new prompt. <Executing/> tags removed from Config Type 1. Fixed parsing of !00StayOn command. Config Type 1 host serial decoding matched to SIM. Time commands removed (hhmmss, mmddyy, ddmmyy). Added !Gx:Mark commands=, sending and receiving. Removed ERROR_DPSK_UART_LOCK. Added executing tags to captureline reply. TestCableCoupler commands and GetSD no longer cause tx power level to temporarily switch low. Interface mode 7 now has EnableHostServeOnPwrup=0. PCBtype >3 enables reset pin. Pin must be held low for 1 second on startup. |
| 1.04 | | <ul style="list-style-type: none"> Added filter to transmit voltage measurement. Fixed invalid DPSK transmit state after TestCableCoupler command. Invalid state persisted in Host Service mode only until transmitter was disabled, a ! or # command was transmitted or line was released. Invalid state usually resulted in inappropriate error: FAILED "Low Transmit Voltage - low battery or bad coupler". |
| 1.05 | 12/06 | <ul style="list-style-type: none"> Added checksums and backup storage of Configuration Data and Hardware data to prevent loss of device configuration if power removed during EEPROM write. Fixed error in SampleAdd and SampleAppend that caused early termination of data. Fixed timing error in Host Service mode when IM not transmitting. Error resulted in longer fixed delays and longer timeout periods. Removed support for 38400 baud. Added *NOTIMEOUT: for lab calibration procedures. Added IMMMonitor to log all IM traffic (including noise if no device is transmitting). Added ability to upgrade firmware through RS-232 port. Added shortcuts for common commands: FCL = ForceCaptureLine, TCC = TestCableCoupler, T20CC = Test20TurnCoupler, REL = ReleaseLine, SWT = SendWakeUpTone. |

| | | |
|-------------|-------|---|
| 1.06 | 6/07 | <ul style="list-style-type: none"> Fixed TONE SENT in debuglevel ≥ 3 sendwakeuptone (was not in an XML comment). Added MLS shortcut for MeasureLineSignal command. IMM incorrectly waited for reply to #g00 (was ok for #G00). Added 0.3-sec maximum wait for first @ sign of standard reply - this prevents full TModem2 delay on a quiet line when remote device isn't responding. Updated firmware loader. Discovery timeout period changed from 5 sec to 8 sec. Commands less than 2 characters now generate errors. Empty command now generates executed and clears Host Service mode timer. |
| 1.07 | | <ul style="list-style-type: none"> Fixed sample storage error in SampleAddLine, AppendLine, and Gdata; Sample data was not added correctly. |
| 1.08 | | <ul style="list-style-type: none"> Removed testHF code - this test code was not supposed to be released. Added <PowerOn/> and <PowerOff/> tags. <PowerOn/> is sent when IMM enters Host Service Mode, <PowerOff/> is sent when IMM exits Host Service Mode. Added SVS monitoring (IMM resets immediately if internal supply voltage falls below 2.5V). |
| 1.09 | 06/08 | <ul style="list-style-type: none"> Fixed error in samplegetdata - caused some sample retrieval errors. Added additional data structure protection to IM receive code (should have no visible effect). |
| 1.10 | 12/08 | <ul style="list-style-type: none"> Fixed error in ConfigType=1 (sent wrong character back). |
| 1.11 | 02/09 | <ul style="list-style-type: none"> Fixed spurious characters in Host Service mode. |
| 1.12 | 06/09 | <ul style="list-style-type: none"> Disabled variable gain to make IMM more responsive. |
| 1.13 | 10/11 | <ul style="list-style-type: none"> PCBType re-initialized to 3 instead of 0 so IMM can communicate after a reset hard data event. Prevents bricking of 10345A and 10345B revision of PCB. GetHD now returns date and time to track multiple builds in one day. XML error messages label now all uppercase <ERROR type= . . .>. Typo fixed 'mssg=' changed to msg=. Removed extra r/n from <HardwareData . . .> tag. |
| 1.14 | 01/12 | <ul style="list-style-type: none"> Removed typo which caused GetHD response to return an invalid XML response in version 1.13. Removed extraneous line terminations following <Executed/> tag. Empty 'msg=' attributes now suppressed in Error tags. Sample data formatting re-worked for consistent string formatting. Commands SampleGetData, SampleErase, and SampleEraseMultiple now accept arguments in following (case independent) formats: 0x0000000A, A, 0xA. Removed possibility of a false failed response to an initial remote command after line capture (i.e., receive an '<ERROR type='FAILED' msg='No reply from remote device'/>' response to a successful !01 cmd). |

16 Index

A

Adding samples · 30
 Addressing commands · 32
 Appending samples · 30

B

Backspace · 19
 Binary data · 29

C

Capturing line · 21
 CE certification · 3
 Command prefixes · 32
 Commands · 32
 addressing · 32
 Config Type 1 · 33
 Config Type 2 · 34
 Config Type 2 Set · 53
 detailed descriptions · 37
 group · 33
 receiving · 22, 23
 sending · 32
 Set · 53
 shortcuts · 60
 SIM · 33
 summary tables · 33
 Comment tag · 17
 Conceptual model · 7, 9
 Config Type 1 commands · 33
 Config Type 2 commands · 34
 Config Type 2 Set commands · 53
 ConfigType · 10
 Configuration Data · 10
 Configuration types · 10
 Confirmation required tag · 16
 Confirming Host wake-up · 25
 Connectors · 61

D

Data · 21
 Data reply tag · 17
 Declaration of Conformity · 3
 Detecting wake-up tone · 22
 Device ID · 23
 DPSK Transceiver · 9

E

Echoing · 19
 Electrical specification · 61
 Entering Host Service mode · 18
 Erasing samples · 31
 Error tag · 15
 Event Counters · 11
 Executed tag · 15
 Executing tag · 17
 Exiting Host Service mode · 20

F

Firmware revision history · 65

G

GData Reply Buffer · 11
 GetCD · 10
 GetEC · 11
 GetHD · 10
 GetSD · 11
 Global commands · 23, 32
 Group commands · 33
 Group number · 23
 Group prefix · 32

H

Hardware · 9
 Hardware Data · 10
 Host · 9
 Host Command Buffer · 12
 Host Data File · 11
 Host echo · 26
 Host reply · 26
 Host Service mode · 12, 13, 18
 entering · 18
 exiting · 20
 Timeouts · 20
 HostFlag · 10
 HostFlag input · 25
 HostFlag transition · 27
 HostPrompt · 25

I

ID prefix · 32
 IM Command Buffer · 12
 IM line
 capturing · 21
 releasing · 21
 IM Receive mode · 12, 13
 IM Reply Buffer · 11
 IM Service mode · 12, 13, 24
 timeouts · 27
 IMFlag · 10, 24
 Inductive Cable Coupler · 9
 Interface Mode shortcuts · 60

L

Limited liability statement · 2
 Line capturing · 21

M

Manual revision history · 64
 Mode · 12, 13
 Model · 7, 9

NNonvolatile structures · 10

OOperating mode · 12, 13
Overview · 7
Overwriting samples · 31

PPinouts · 61
Power consumption · 63
Power cycling · 14
Power input · 10
Power supply · 10, 62
Prefix
 group · 32
 ID · 32
 serial number · 32
Prompts · 19

RReading samples · 31
Receiving commands · 22
Receiving host commands · 23
Releasing IM line · 21
Remote devices
 receiving commands from · 22
 sending commands to · 21
Reply tags · 15
ResetEC · 11
Resistance · 62
Retrieving list of samples · 30
Revision history
 firmware · 65
 manual · 64

SSample data · 30
Sample Data · 11
Sample ID · 11
Samples
 adding · 30
 appending · 30
 erasing · 31
 overwriting · 31
 reading · 31
 retrieving list of · 30
Sending command to Host · 25
Sending commands · 32
Sending data · 21
Serial Break · 25
Serial Byte · 25
Serial number · 23
Serial number prefix · 32
Shortcuts
 Interface Modes · 60

Signal detector
 disabling · 23
Signal Detector · 9
SIM commands · 33
Sleep mode · 12, 13
Specification
 electrical · 61
Status Data · 11

TTag
 Comment · 17
 Confirmation required · 16
 Data reply · 17
 Error · 15
 Executed · 15
 Executing · 17
 Timeout · 17
 Warning · 16
Tags · 15
Terminating command to Host · 26
Terminating commands · 19
Termination character from Host · 27
Termination of Host reply · 26
THost0 · 28, 58
THost1 · 28, 58
THost2 · 28, 58
THost3 · 28, 58
THost4 · 27, 28, 58
THost5 · 27, 28, 59
Timeout tag · 17
Timeouts · 20, 27
TModem2 · 20, 59
TModem3 · 20, 59
TModem4 · 20, 59
Tone Detect · 9

VVersions · 64, 65
Volatile structures · 12
Voltage · 62
Voltage input · 10

WWake-up tone · 21, 22
Waking Host · 24
Waking IMM · 22
Warning tag · 16

XXML-like format · 15
