OpenBMIP™ Standard

August 14, 2014

iDIRECT

VT iDirect is a global leader in IP-based satellite communications providing technology and solutions that enable our partners worldwide to optimize their networks, differentiate their services and profitably expand their businesses. Our product portfolio, branded under the name iDirect, sets standards in performance and efficiency to deliver voice, video and data connectivity anywhere in the world. VT iDirect is the world's largest TDMA enterprise VSAT manufacturer and is the leader in key industries including mobility, military/government and cellular backhaul.

VT iDirect
Company Web site: http://www.idirect.net ~ Main Phone: 703.648.8000
TAC Contact Information: Phone: 703.648.8151 ~ Email: tac@idirect.net ~ Web site: http://tac.idirect.net

iDirect Government™, created in 2007, is a wholly owned subsidiary of iDirect and was formed to better serve the U.S. government and defense communities.

iDirect Government™
Company Web site: http://www.idirectgov.com ~ Main Phone: 703.648.8118
TAC Contact Information: Phone: 703.648.8111 ~ Email: tac@idirectgov.com ~ Web site: http://tac.idirectgov.com

Document Name: OpenBMIP_Standard_RevA_08112014.pdf

Document Part Number: T0000683

Contents

Figures

Tables

Revision History

The following table shows all revisions for this document. To determine if this is the latest revision, check the Technical Assistance Center (TAC) Web site. Refer to *Getting Help* on page xvi for TAC access information.

Revision	Date	Updates
1.0-6.0	2011-2012	Internal versions
7.0	Dec 14, 2012	First public release
7.1	February 4, 2013	Minor edits: remove “optional” nature of some messages
7.2	February 13, 2013	Incorporate internal and customer reviews: Major cleanup and consistency check Delete redundant and incorrect message charts; add Table 1 as a clearer format description; add worked example.
7.3	February 19, 2013	Additional error codes Additional command codes to get/set timeouts Added a table of commands in numerical order Clarified Band Select Logic Cleaned up acronyms & Glossary
7.4	February 22, 2013	Clarified terminator character in string data Removed spaces from XML format
7.5	February 25, 2013	(No information)
7.6	March 12, 2013	Revised error codes Corrected 2.2 timeout error code description Deleted get/set timeout commands Clarified CMD_SEQ behavior
7.7	March 26, 2013	Revised error codes Corrected 2.2 timeout error code description Deleted get/set timeout commands Clarified CMD_SEQ behavior Revised Get Band Selection response strings

7.8	May 19, 2013	Added GPOWERAMP and SPOWERAMP commands. Clarified file transfer timeout behavior.
7.9	July 18, 2013	Corrected codes for GPOWERAMP and SPOWERAMP commands.
7.91	Aug 27, 2013	Corrected header in Table 4 Corrected ID items in Section 3.2: BUC Part Number (BPN) , BUC Manufacturer ID (MID) , BUC Serial Number (BSN) , BUC Functional ID (FID) Added descriptions of the XML fields
7.92	Sep 04, 2013	1. 2.2.1 Time outs: BUC vendor to specify timeout if 5 seconds impractical. 2. 2.2.5 RSP_CODE: Change “0x08: File receive fail” to “0x08:File transfer fail” Add “Unimplemented” RSP_CODE. 3. 2.2.6 CMD_VERS: Add “BUC should interoperate with previous versions.” 4. 2.5.1.2.1 Header Deconstruction: Change “RESP_CODE” to “RSP_CODE” 5. 2.7 Command Syntax: Change “14 Set Calibration File” to “14 Get Calibration File” 6. 2.7.3 Get Heartbeat: Change the example of Response DATA to "{BSN=A12345A33,TEMP=+48.8,FSTATUS=lolock,fannormal,outputmute,normaltemp}” 7. 2.7.10 Get Fault Status: Change the example of Response DATA to {lolock,fannormal,outputmute,normaltemp} Separate statuses in the same category by "/" such as “lolock/lounlock,fannormal/fanstanby/fanfailed, ??? 8. 2.7.11 Get Latched Status: Change the example of Response DATA to "{lounlock,fannormal,outputmute,normaltemp}” Define which statuses the BUC should latch.
8	Sep 24, 2013	Change factory calibration procedure and XML file format to add detector reading with power amp disabled.
8.1	Sep 25, 2013	Corrected missing references in 2.7
8.2	Oct 22, 2013	Deleted reference to OpenAMIP in GPOWERAMP command description
8.3	Nov 8, 2013	Added option to compress file using zlib/gzip Added note to select peaks & troughs of gain vs frequency curve
8.4	Nov 11, 2013	Rephrased note to select peaks & troughs of gain vs frequency curve, and endpoints.
8.5	Nov 11, 2013	Added clarification to ADC slope.
8.6	Nov 13, 2013	Added clarification to Kermit description.
8.7	Jan 27, 2013	Added requirement to support both SW upgrade and downgrade to 2.7.15.1.3.1 Software Upgrade Procedure Added requirement to validate the SW package
8.8	Apr 04, 2014	Deleted TFTP from Ethernet implementation

8.9	Apr 11, 2014	Section 3.2 Table 5: Deprecated freq_step_size; steps are permitted to be non-uniform. Section 3.2 Table 5: Clarified detector monotonicity.
9.0	Apr 28, 2014	Section 2.6.2: Describe Ethernet operation: Assign port numbers for OpenBMIP (UDP 6001) and Kermit (TCP 1649) Section 2.7.1: Clarify that the serial number (BSN) needs to be unique, but does not need to follow the example format. Section 3.1: Add note that Calibration frequencies of (band select filter corner frequency) and lower are to be calibrated with the BUC Filter Select input set for low frequency band operation. Section 3.1: Raw ADC hex values are recommended to be linear to BUC power expressed in dB. Section 3.1: Add a comment on use of “extrapolation” if BUC temperature reading is outside of the BUC Cal file temperatures.
9.1	Apr 29, 2014	Section 2.2.5: Removed incorrect code for “invalid FCS”.
9.2	May 29, 2014	Section 2.7.15.2.5: Corrected file transfer diagram (“STARTFILETRANSFER” -> “SENDFILE”)
9.3	Feb 19, 2015	Section 3.2 Table 5: Added “buc_cutoff_frequency_mhz” parameter; increased <num_temps> maximum to 5.
9.4	April 1, 2015	Section 3.2 Table 5: Changed description for <buc_cutoff_frequency_mhz> for consistency and correctness against actual implementation.
A	08/14/2015	First release of the OpenBMIP Standard document in the iDirect Technical Publications template (RevC). Minor changes that do not affect the technical content.

About

Purpose

This document describes the OpenBMIP™ administrative interface between the modem and block up converter (BUC) components of a satellite terminal. The most important function of this interface is to enable calibration of the terminal's transmitted power for regulatory compliance. This enables a terminal integrator to provide a seamless installation process, with respect to transmitter power calibration.

This standard may be tailored to the application by selection of interface type (RS-422 serial or UDP over Ethernet). It is also designed to be extensible for vendor-specific enhancements.

This interface is intended for setup, calibration, and installation purposes. It is not intended for continuous operation while the BUC is transmitting live data. The messages described here are typically exchanged:

- When a BUC is first installed into a terminal,
- When the terminal is first powered on, and
- Before the terminal begins transmission to connect to a satellite network.

Disclaimer

Open BUC Modem Interface Protocol (OpenBMIP™) was invented by iDirect.

The name "OpenBMIP™" is a trademark of iDirect.

To the extent that iDirect has rights to control the protocol itself, iDirect grants rights to implement the protocol to all, without restriction.

Use of the trademark "OpenBMIP" to describe an unmodified implementation of this protocol is unrestricted. Use the term "modified OpenBMIP" to describe a variant of this protocol, is also unrestricted; however the document containing the term "modified OpenBMIP" refers to this document.

While iDirect, Inc. strives to make the information in this document as accurate as possible, iDirect makes no claims, promises, or guarantees about the accuracy, completeness, or adequacy of the contents, and expressly disclaims liability for errors and omissions. No warranty of any kind, whether implied, expressed, or statutory, including but not limited to the warranties of non-infringement of third party rights, title, merchantability, or fitness for a particular purpose, is given with respect to the contents of this document.

iDirect, Inc. reserves the right to change or update this document at any time.

Certification

You may certify your compliance with the test suite yourself. If you do, you are free to use the trademark "OpenBMIP™" freely for any product that you have certified.

Your use of the OpenBMIP™ trademark authorizes any OpenBMIP™ implementer to validate your implementation and publish the results, referring to your product by company and product name, if the implementer finds your implementation to be non-compliant. A finding of non-compliance will not be published until thirty days after the OpenBMIP™ member notifies you of the finding. At your option, the implementer's published finding of non-compliance will include a reference to a statement in rebuttal by you.

Audience

The intended audience for this document is an engineering team responsible for integrating a satellite modem with a Block Up Converter (BUC), or a team designing a compliant BUC or modem.

Contents

This document contains the following major sections:

- *Introduction*

  This chapter gives an introduction about OpenBMIP.

- *Protocol Specification*

  This chapter describes the protocol specifications, message types, and syntax.

- *Calibration*

  This chapter explains how the BUC is calibrated.

- *Acronyms and Abbreviations*

  This list is meant to be generic within this document and may contain acronyms and abbreviations not found in this manual and some terms may not be defined based on industry standards.

- *Glossary*

  This list is meant to be generic within this document and may contain entries not found in this manual and some terms may not be defined based on industry standards.

Standards Reference

- IETF RFC 1055 Serial Line IP
- IETF RFC 1171 Point-to-Point Protocol
- IETF RFC 1700 Assigned Numbers
- IETF RFC 1952 GZIP file format specification version 4.3
- Kermit

Document Conventions

This section illustrates and describes the conventions used throughout this document.

Convention	Description	Example
`Command`	Used when the user is required to type a command at a command line prompt or in a console.	Type the command: `cd /etc/snmp/`

`Terminal Output`	Used when showing resulting output from a command that was entered at a command line or on a console.	`crc report all`
		`8350.3235 : DATA CRC [ 1]`
		`8350.3502 : DATA CRC [5818]`
		`8350.4382 : DATA CRC [ 20]`

Screen Reference	Used when referring to text that appears on the screen on a Graphical User Interface (GUI). Used when specifying names of commands, menus, folders, tabs, dialogs, list boxes, and options.	1. To add a remote to an inroute group, right-click the Inroute Group and select **Add Remote**. The **Remote** dialog box has a number of user-selectable tabs across the top. The **Information** tab is visible when the dialog box opens.
Hyperlink	Used to show all hyperlinked text within a document or external links such as web page URLs.	For instructions on adding a line card to the network tree, see *Adding a Line Card* on page 108.

WARNING: A warning highlights an essential operating or maintenance procedure, practice, condition, or statement which, if not strictly observed, could result in injury, death, or long term health hazards.

CAUTION: A caution highlights an essential operating or maintenance procedure, practice, condition, or statement which, if not strictly observed, could result in damage to, or destruction of, equipment or a condition that adversely affects system operation.

NOTE: A note is a statement or other notification that adds, emphasizes, or clarifies essential information of special importance or interest.

Getting Help

The iDirect Technical Assistance Center (TAC) and the iDirect Government Technical Assistance Center (TAC) are available to provide assistance 24 hours a day, 365 days a year. Software user guides, installation procedures, FAQs, and other documents that support iDirect and iDirect Government products are available on the respective TAC Web site:

- Access the iDirect TAC Web site at http://tac.idirect.net
- Access the iDirect Government TAC Web site at http://tac.idirectgov.com

The iDirect TAC may be contacted by telephone or email:

- Telephone: 703.648.8151

- E-mail: tac@idirect.net

The iDirect Government TAC may be contacted by telephone or email:

- Telephone: 703.648.8111
- Email: tac@idirectgov.com

iDirect and iDirect Government produce documentation that are technically accurate, easy to use, and helpful to our customers. Please assist us in improving this document by providing feedback. Send comments to:

- iDirect: techpubs@idirect.net
- iDirect Government: techpubs@idirectgov.com

For sales or product purchasing information contact iDirect Corporate Sales at the following telephone number or e-mail address:

- Telephone: 703.648.8000
- E-mail: sales@idirect.net

1 Introduction

This chapter contains the following sections:

1.1 Scope

This document describes a monitoring and control interface between the modem and block up converter (BUC) components of a satellite terminal. The most important function of this interface is to enable calibration of the terminal's transmitted power for regulatory compliance.

1.1.1 Tailoring

OpenBMIP™ may be tailored to fit the application. For example, systems with a fixed BUC local oscillator frequency may be implemented without support for the Set Local Oscillator command.

In order to allow different BUC vendors to provide the end user with a richer command set beyond those required by the monitoring and control interface, the protocol allows for the definition of vendor-specific commands (see *File Transfer* on page 21).

1.2 Terminal Overview

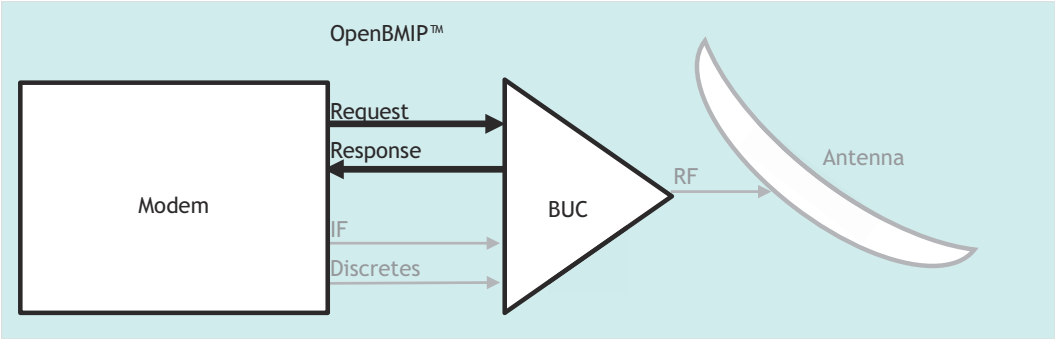
The modem:

- acts as a satellite modem,
- coordinates with the satellite network to provide monitoring and control functions,
- provides IF, reference frequency, and control signals to the BUC,
- reads gain calibration data from the BUC and from other sources, and
- adjusts its IF level to produce the correct RF power level at the BUC output for satellite link operation.

The BUC:

- upconverts the IF from the modem to the required satellite uplink frequency (RF),
- amplifies the upconverted RF power with fixed gain, and
- interacts with the modem to enable calibrated control of the RF output power.

Figure 1-1. BUC and Modem Connections



1.3 BUC Functional Requirements to Support OpenBMIP™

A typical BUC implementation incorporates the following key functions to support calibrated operation using the OpenBMIP™ protocol:

- an RF RMS power detector (typically placed at the input of the final power amp, but may be implemented at another appropriate point in the signal chain to support a fully calibrated power level, from IF input to RF output of the BUC),
- a temperature sensor to support temperature correction of calibrated values,
- a microcontroller with these interfaces and features:
  - one-time-programmable memory to store ID numbers,
  - factory-programmable memory (read-only in the field) to store calibration data,
  - analog to digital converters for the power detector and the temperature sensor, and
  - a physical layer interface (serial or Ethernet) for connection to the modem.

OpenBMIP™ may be implemented as either an RS-422 serial connection, or a UDP connection through the modem's Ethernet connector.

The BUC EEPROM must have enough memory to store the following information:

- In a one-time-programmable (OTP) page (see *Get Product ID* on page 14)
  - BUC Part Number (BPN)
    - 13 bytes; unused trailing bytes filled with "x"
  - BUC Manufacturer ID (MID)
    - 2 bytes
  - BUC Serial Number (BSN) with date code and Revision; per 2.7.1

    - 9 bytes
  - BUC Functional ID (FID) per Figure 2-3
    - 6 bytes; for example: 5W, 29.0-30.0GHz, Standard:: ' 050300
- In an erasable page (with lock feature)
  - Calibration file (XML Format)
  - Check Sum

NOTE: It is anticipated that some applications of this interface will allocate functionality to other system components. For example, the microcontroller and EEPROM might not be co-located with the BUC. Such differences should be implemented so they are transparent from the perspective of the modem.

2 Protocol Specification

This chapter contains the following sections:

2.1 Introduction

The purpose of this protocol is to provide specifications for communication between the satellite router (client) and BUC (server) using a monitoring and control interface.

This protocol is a Client-Server protocol. The Client is the satellite router; the Server is the BUC. Only one router at a time is connected to a BUC. The router always initiates communication with the BUC. The BUC never transmits without receiving a request from the router.

2.2 Frame Description

Table 2-1 shows the structure of an OpenBMIP message.

Table 2-1. MIP Message Format

	Bit							
Byte	7	6	5	4	3	2	1	0
0	TYPE	CMD_CODE						
1	RSP_CODE					CMD_VERS		
2	CMD_SEQ(high byte)							
3	CMD_SEQ(low byte)							
4	DATA_LEN(high byte) DATA_LEN = M							
5	DATA_LEN(low byte)							
6	DATA BYTE 0							
7	DATA BYTE 1							
…	…							
n-4	DATA BYTE M-1 (ASCII NULL character)							
n-3	FCS(high byte) FCS is computed from bytes 0…n-4							
n-2	FCS(low byte)							
n-1	END (0xC0)							

All message fields are binary, with the exception of the DATA field, which is NULL-terminated ASCII, including the representation of decimal or hexadecimal numbers.

2.2.1 Time Outs

If the recipient receives message characters, with more than 2 seconds elapsed between characters, it should time out and assume a connection fault has occurred. In the case of the BUC (server), it should send RSP_CODE 0x02 (see *RSP_CODE* on page 7). There is no concept of a message queue in OpenBMIP; the client (modem) must wait for a response before sending another message to the server (BUC). This provides natural throttling of flow rate. If no response is received within 5 seconds, the client should time out and assume a connection fault. During a file transfer, if the router does not acknowledge a message from the BUC within 5 seconds, the BUC should time out and send a message with RSP_CODE = 0x08. If 5 seconds is impractical, the BUC vendor should specify what the BUC file transfer timeout is. The same rules apply during a file transfer.

2.2.2 Empty Message

If the recipient receives an END character (0xC0), but has not received corresponding prior message characters, it should silently ignore it. This is a standard part of the SLIP protocol (see *IETF RFC-1055 SLIP Protocol* on page 8).

2.2.3 TYPE

One bit is used to identify whether the message is a command or response.

- 0x0: command
- 0x1: response

2.2.4 CMD_CODE

Seven bits are used; there are up to 128 command codes. Codes are assigned as follows:

- 0x00 to 0x5E: assigned by iDirect
- 0x60 to 0x7F: available for third-party use

2.2.5 RSP_CODE

Five bits are used; there are up to 32 response codes. The server responds to every command with either a failure or success. Other error codes can be defined per command as required.

The following values are defined:

- 0x00: Failure (of a type not in the list below)
- 0x01: Success
- 0x02: Message receipt timeout
- 0x03: Invalid command sequence number (future use)
- 0x04: Invalid FCS detected in modem command
- 0x05: Incompatible command version
- 0x06: Invalid third-party command
- 0x07: Data length and data field size mismatch
- 0x08: File transfer fail
- 0x09: BUC Input Buffer Overrun
- 0x0A: Modem Command Data Field Error
- 0x0B: Upgrade fail
- 0x1E: Unimplemented command
- 0x1F: Unrecognized command

2.2.6 CMD_VERS

Three bits are used to identify the version of the protocol. The current protocol is version 2 and subsequent versions will be one up. If the BUC supports at least the same version as the router, it will be compatible; otherwise it should report an error. The BUC should provide backward compatibility for versions 0 through 2.

2.2.7 CMD_SEQ

Sixteen bits are used; this is the command sequence number used to match requests to responses. The command sequence number starts with 0 for the initial command and is incremented by one for each additional command. After the value reaches 0xff 0xff, it wraps to 0x00 0x00. The BUC response should echo the CMD_SEQ value from the command it is responding to. In the case of a timeout, the BUC should use the most recent received CMD_SEQ value. The BUC is not required to detect CMD_SEQ errors; this error code is reserved for future use.

2.2.8 DATA_LEN

Sixteen bits are used; this is the message length in bytes excluding the header. This is the length of the ASCII DATA field, including its NULL (0x00) terminator.

2.2.9 DATA

This is ASCII data; its content and length are command-dependent. It is terminated with an ASCII NULL (0x00) character, and maximum length is 247. Therefore, to send the hexadecimal value 0xABC, one sends four bytes: 65, 66, 67, 0. To send the decimal value 123, one sends four bytes: 49, 50, 51, 0. The DATA field is case insensitive, for hexadecimal values, filenames, or any other content. Only the following characters are allowed in the DATA field (in addition to the NULL terminator at the end):

{0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ,._=+-}

2.2.10 FCS

Sixteen bits are used for the Frame Check Sequence (FCS); this is computed with the algorithm used by PPP for its FCS calculation. Refer to RFC-1171 for a sample implementation (http://tools.ietf.org/html/rfc1171#page-44). The FCS is calculated from all bytes which precede the FCS.

2.3 IETF RFC-1055 SLIP Protocol

The message is packaged and transmitted using escape sequence insertion and checks in accordance with IETF RFC-1055 SLIP protocol. Refer to RFC-1055 for a sample implementation in the C programming language. This sample algorithm explains the steps to send a message:

Optionally, send an initial END (0xC0) byte to flush out any data that may have accumulated in the receiver due to line noise. The initial END byte does not count toward CMD_SEQ, DATA_LEN, or FCS values. The receiver should silently ignore an END with no corresponding message.

- Construct the message, including DATA_LEN and FCS. The DATA_LEN and FCS computed values are unaffected by the substitutions below, but they may themselves be subject to substitution.
- For each byte in the message make the following substitutions:
  - 0xC0 becomes 0xDB 0xDC
  - 0xDB becomes 0xDB 0xDD

- any other value is unchanged

- Send the final END (0xC0) byte. ***The final END byte does not count toward DATA_LEN or FCS values***.
- Apply the above substitutions after constructing the message, so that DATA_LEN and FCS are unaffected.

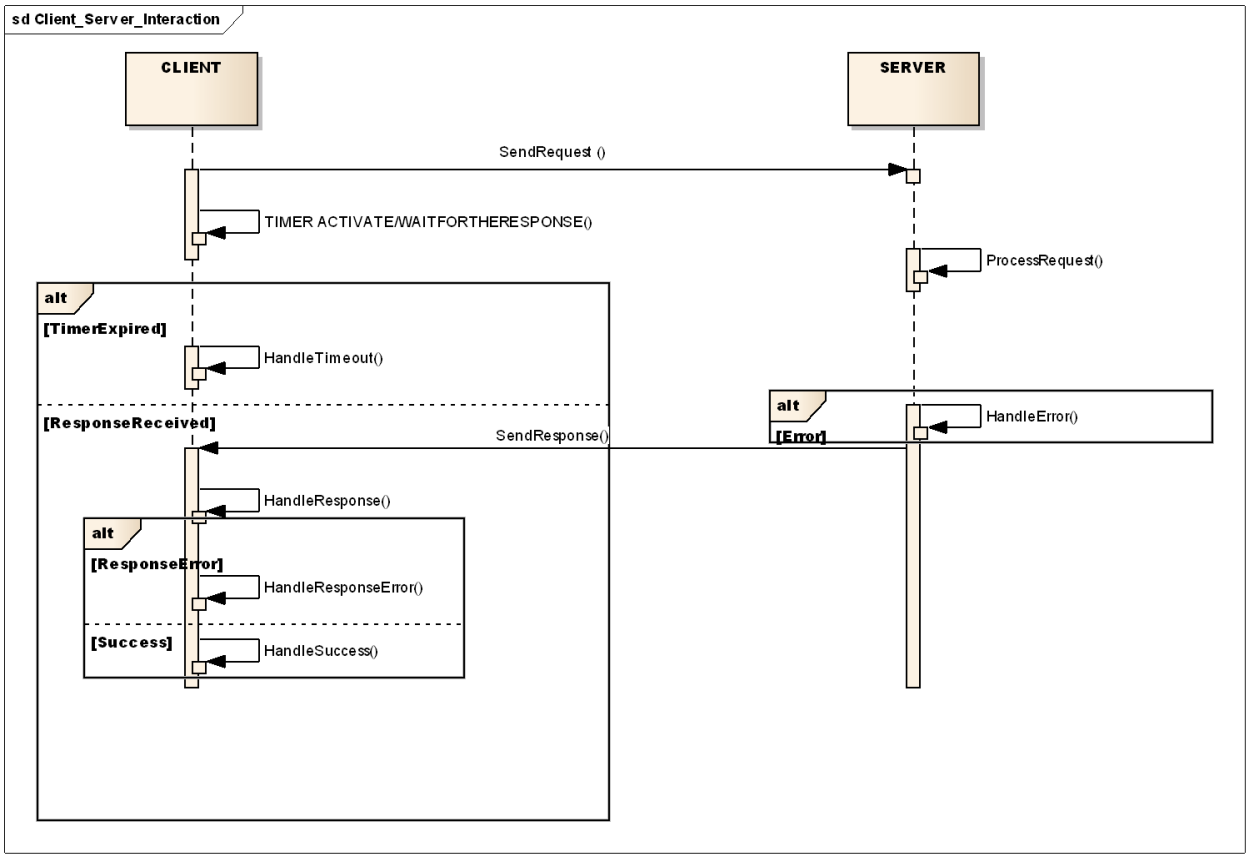
To receive a message:

- Receive the message, including DATA_LEN and FCS. The DATA_LEN and FCS computed values are unaffected by the substitutions below, but they may themselves be subject to substitution.
- For each byte received make the following substitutions:
  - 0xDB 0xDC becomes 0xC0
  - 0xDB 0xDD becomes 0xDB
  - any other value is unchanged
- Receive the final END (0xC0) byte. ***The final END byte does not count toward DATA_LEN or FCS values***.
- Compare the FCS computed, with the FCS received; send an "Invalid FCS" response code if computed FCS does not match received FCS.

2.4 Client/Server Interaction Sequence Diagram

Figure 2-1 shows a sample interaction between the client (CM) and the server (BUC) represented as a sequence diagram. As can be seen from the sequence diagram, it is expected that the CM is always the client while the BUC is always the server.

Figure 2-1. Client-Server Interaction Sequence



2.5 A Worked Example

In this example, the modem commands the BUC to Set band selection, followed by Get band selection.

2.5.1 Construction of Set band selection command

Objective of message: Set band select to low band

2.5.1.1 Command raw byte sequence

Header	Data	FCS	END
0x7 0x0 0x0 0x1 0x0 0x8	0x6c 0x6f 0x77 0x62 0x61 0x6e 0x64 0x0	0x5a 0x73	0xc0

2.5.1.2 Header Deconstruction

2.5.1.2.1 Header Deconstruction by Field

TYPE: 0 => COMMAND

CMD_CODE: 7 => SBANDSEL

RSP_CODE: 0

CMD_VERS: 0

CMD_SEQ: 1

DATA_LEN: 8

2.5.1.2.2 Header Deconstruction by Bit

Byte	Bit							
0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0

2.5.1.3 Data Payload

lowband + NULL => {'l','o','w','b','a','n','d',0}

2.5.1.4 Frame Check Sequence

0x5a73

2.5.1.5 Expected BUC Response

2.5.1.5.1 Response Raw Byte Sequence

Header	Data	FCS	END
0x87 0x8 0x0 0x1 0x0 0x0		0xee 0x00	0xc0

2.5.1.5.2 Header Deconstruction

2.5.1.5.2.1 Header deconstruction by field

TYPE: 1 => RESPONSE

CMD_CODE: 7 => SBANDSEL

RSP_CODE: 0

CMD_VERS: 0

CMD_SEQ: 1

DATA_LEN: 0

2.5.1.5.2.2 Header deconstruction by bit

Byte	Bit							
0	1	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0

2.5.1.5.3 Data Payload

Empty

2.5.1.5.4 Frame Check Sequence

0xee00

2.6 Physical Layer

OpenBMIP™ may be used with either serial RS-422 or Ethernet as the physical layer.

2.6.1 Serial RS-422 Mode

2.6.1.1 Communication Parameters

This mode is used for the data transmission at the physical layer; settings are:

Serial Communication Settings	
Baud Rate	38400
Data Bits	8
Parity	None
Stop Bits	1
Flow Control	None
Handshaking	None

2.6.1.2 Data Ordering

For serial transfers, bits are sent beginning with Bit 7 (MSB) for any given byte. So, in order of transmission, a message is sent over the serial interface as follows:

(Byte 0, Bit 7), (Byte 0, Bit 6), … , (Byte 0, Bit 0), (Byte 1, Bit 7), …, (Byte n-1, Bit 0)

2.6.2 Ethernet Mode

The purpose of this protocol is to provide specification for communication between the modem (client) and BUC Proxy (server) using an Ethernet Interface. The BUC Proxy will communicate with the BUC and provide responses to the modem.

The Modem sends OpenBMIP messages to the BUC's IPv4 address (defined in a separate terminal-specific document), on UDP port 6001. For file transfers, the modem sends Kermit Over TCP/IP commands to the BUC's IPv4 address on TCP port 1649. The BUC accepts a connection from one client at a time. The modem always initiates the connection, similar to the Serial protocol.

The OpenBMIP commands described in *Command Syntax* on page 14 will be sent via UDP. Each UDP packet contains a single command or a response. Figure 2-2 shows the UDP packet structure:

Figure 2-2. UDP Ethernet Packet Format

The OpenBMIP Frame format in Figure 2-2 is specified in *Frame Description* on page 6. The interaction sequence between the client and server has been specified in *Client/Server Interaction Sequence Diagram* on page 9. All commands operate the same as with a serial connection; except for those related to file transfer (see *File Transfer* on page 21).

2.7 Command Syntax

The following sections detail individual commands and their responses. The DATA and Response DATA are shown as example strings enclosed in curly braces: {sample}. Each string should also include a NULL terminator, not shown. If the Command DATA or Response DATA string is blank, the NULL terminator is still included.

Table 2-2. Commands in Numerical Order

CMD_CODE	Section	Function Name
1	*2.7.10 Get Fault Status* on page 19	GFAULTSTATUS
2	*2.7.2 Get PA Power* on page 16	GPAPOWER
3	*2.7.1 Get Product ID* on page 14	GPRODUCTID
4	*2.7.3 Get Heartbeat* on page 16	GHEARTBEAT
5	*2.7.8 Get Local Oscillator* on page 19	GLOSCILLATOR
6	*2.7.4 Get Band Selection* on page 17	GBANDSEL
7	*2.7.5 Set Band Selection* on page 17	SBANDSEL
8	*2.7.9 Set Local Oscillator* on page 19	SLOSCILLATOR
9	*2.7.12 Reset Status Latch* on page 20	RSTATUSLATCH
10	*2.7.15.2.2 Send File* on page 23	SENDFILE
11	*2.7.15.2.4 Kermit File Transfer Status* on page 23	STATUSKERMITFILE
12	*2.7.15.2.3 Receive File* on page 23	RECEIVEFILE
13	*2.7.13 Get Software Version* on page 20	GSOFTWAREVER
14	*2.7.15.1.1 Get Calibration File* on page 21	GCALIBFILE
15	*2.7.15.1.2 Set Calibration File* on page 21	SCALIBFILE
16	*2.7.15.1.3 Prepare For Upgrade* on page 22	PREP_UPGRADE
17	*2.7.14 BUC Reset* on page 21	BUC_RST
18	*2.7.11 Get Latched Fault Status* on page 20	GLFAULTSTATUS
19	*2.7.6 Get Power Amp State* on page 18	GPOWERAMP
20	*2.7.7 Set Power Amp State* on page 19	SPOWERAMP

2.7.1 Get Product ID

This command is used to query the BUC EEPROM contents. The contents of the response from the BUC should follow the format and response information as described below.

Command Description: Echo manufacturer name and BUC model number

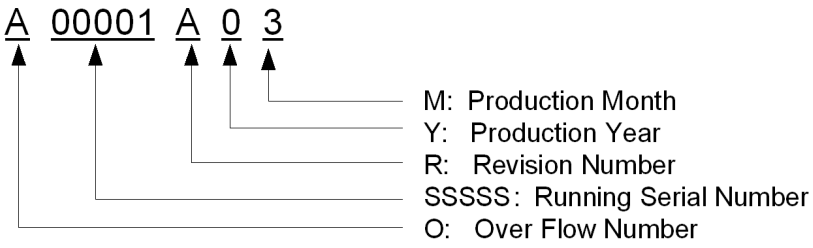
CMD_CODE: 3

Function Name:	GPRODUCTID
Command DATA:	{}
Response DATA:	{BPN=1234567890xxx,MID=mm,BSN=A12345A33,FID=050300}

The BUC identifiers are:

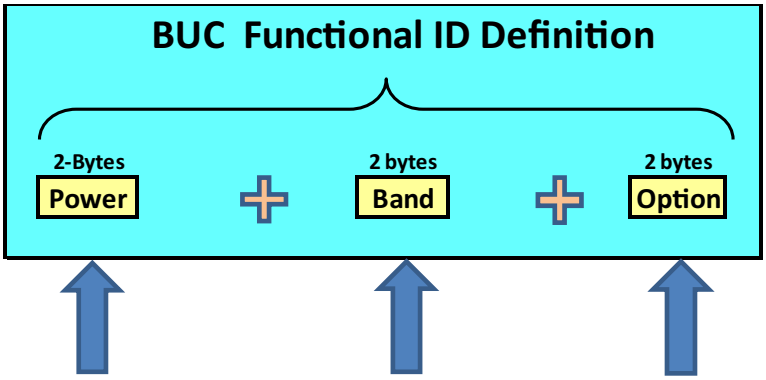
- BUC Part Number (BPN) assigned by iDirect if used with iDirect routers
  - 13 bytes; unused trailing bytes filled with "x"
- BUC Manufacturer ID (MID) assigned by iDirect
  - 2 bytes
- BUC Serial Number (BSN) Must be unique; see suggested example in Figure 2-3
  - 9 bytes
- BUC Functional ID (FID) per Figure 2-4
  - 6 bytes; Example: 5W, 29.0-30.0GHz, Standard:: --> 050300

Figure 2-3. BUC Serial Number with Embedded Date Code and Revision



O: Overflow Number – Alphabet (1 character)
"A" to "Z", e.g.: A99999 → B00001
SSSSS: Running Serial Number – Number (5 digits)
"00001" to "99999"
R: Revision Number – Alphabet (1character)
"A" to "Z"
Y: Production Year – Number (1 digit)
Calendar Number, e.g.: 2011:1, 2012=2, 2013=3....
M: Production Month- AlphaNumeric (1 character)
"1" to "9" and "X" as October, "Y" as November, "Z" as December

Figure 2-4. BUC Functional ID



2W	02
3W	03
4W	04
5W	05
7W	07
8W	08
10W	10

29.0 - 29.5 GHz	01
29.5 - 30.0 GHz	02
29.0 - 30.0 GHz	03
30.0 - 31.0 GHz	04
29.5 - 31.0 GHz	05

Standard	00

2.7.2 Get PA Power

Command Description: Echo raw power in terms of digital voltage with the resolution as specified in the calibration file

CMD_CODE: 2

Function Name: GPAPOWER

Command DATA: {}

Response DATA: {ABC} ASCII representation of hex value from A to D converter

NOTE: This function should operate correctly even if the final power amp stage is disabled by the Keyline discrete or by the Set Power Amp State command.

2.7.3 Get Heartbeat

This command will be used for heartbeat purposes.

Command Description: Echo S/N with embedded MFG date code, temperature and fault status

CMD_CODE: 4

Function Name: GHEARTBEAT

Command DATA: {}

Response DATA: {BSN=A12345A33,TEMP=+48.8,FSTATUS=lolock,fannormal, outputmute,normaltemp}

For FSTATUS definitions, see *Get Fault Status* on page 19.

2.7.4 Get Band Selection

Command Description: Get IF Harmonic Filter Selection

CMD_CODE: 6

Function Name: GBANDSEL

Command DATA: {}

Response DATA: {lowband,gpio_0,serial_0,tone_1}

Response DATA is as shown in Table 2-3.

Explanation of control states:

If the GPIO is left unconnected (floating) GPIO defaults to 0, highband (wideband).

If no serial port command is received, serial defaults to highband (wideband).

If no 27MHz tone is received, tone defaults to highband (wideband).

If any input requires lowband, the BUC operates in lowband.

Table 2-3. Band Select Logic

State No.	Band Select Control Signal			Response DATA
	GPIO	Serial Port	27MHz Tone	
0	0 or float	highband or unspecified	absent	highband,gpio_0,serial_0,tone_0
1	0	highband	present	lowband,gpio_0,serial_0,tone_1
2	0	lowband	absent	lowband,gpio_0,serial_1,tone_0
3	0	lowband	present	lowband,gpio_0,serial_1,tone_1
4	1	highband	absent	lowband,gpio_1,serial_0,tone_0
5	1	highband	present	lowband,gpio_1,serial_0,tone_1
6	1	lowband	absent	lowband,gpio_1,serial_1,tone_0
7	1	lowband	present	lowband,gpio_1,serial_1,tone_1

2.7.5 Set Band Selection

This command is used to select the BUC Band Filter using the monitoring and control interface. It should be noted that this is only one of the methods by which band selection can be commanded. The BUC may also support selection by Tone or using a discrete signal (GPIO).

The selection criterion to be followed by the BUC when one or more of these methods are used is as explained in *Get Band Selection* on page 17.

Command Description: Set IF Harmonic Filter Selection

CMD_CODE: 7

Function Name: SBANDSEL

Command DATA: {highband}

Response DATA: {}

2.7.6 Get Power Amp State

Command Description: Get state of final power amplifier

CMD_CODE: 19

Function Name: GPOWERAMP

Command DATA: {}

Response DATA: {enabled,gpio_0,BMIP_0}

Response DATA is as shown in Table 2-4.

Explanation of control states:

- If the GPIO Keyline discrete is not connected, the Keyline interface defaults to enable the BUC PA.
- If no OpenBMIP command is received to say otherwise, OpenBMIP defaults to enable the BUC PA.
- Once an OpenBMIP command disables the BUC PA, it is disabled until re-enabled by another command - regardless of the state of the discrete.
- If the discrete is in the state which disables the BUC PA, the PA is disabled - regardless of any commands received.
- If the PA is disabled by any inputs, it remains disabled until all inputs re-enable it.
- Hardware discrete GPIO Keyline definition: RS-422 differential signal; PA On:: [Float/Open inputs] or (A-B) > 0.2V; PA Standby:: (A-B) < -0.2V

Table 2-4. Power Amp Logic

State No.	Control Source		Response DATA
	Keyline	OpenBMIP	
0	PA On	enabled or unspecified	enabled,gpio_0,BMIP_0
1	PA On	disabled	disabled,gpio_0,BMIP_1
2	PA Standby	enabled	disabled,gpio_1,BMIP_0
3	PA Standby	disabled	disabled,gpio_1,BMIP_1

2.7.7 Set Power Amp State

This command is used to enable or disable the final power amp in the BUC, using the OpenBMIP interface. It should be noted that this is only one of the methods by which pwer amp state can be commanded. The BUC may also support selection by OpenAMIP (in terminals configured with a common controller for the BUC and the antenna) or using a discrete signal (GPIO) known as “Keyline.” The selection criterion to be followed by the BUC when one or more of these methods are used is as explained in *Get Band Selection* on page 17.

Command Description:	Set state of final power amplifier
CMD_CODE:	20
Function Name:	SPOWERAMP
Command DATA:	{enabled}
Response DATA:	{}

2.7.8 Get Local Oscillator

Command Description:	Get the Local Oscillator Frequency
CMD_CODE:	5
Function Name:	GLOSCILLATOR
Command DATA:	{}
Response DATA:	{28.05GHz}

2.7.9 Set Local Oscillator

This command is not used for a fixed-frequency BUC.

Command Description:	Set the Local Oscillator Frequency
CMD_CODE:	8
Function Name:	SLOSCILLATOR
Command DATA:	{28.05GHz}
Response DATA:	{}

2.7.10 Get Fault Status

Command Description:	Display the fault status within the BUC; display current faults and latched faults (faults that have previously occurred but have not been cleared)
CMD_CODE:	1
Function Name:	GFAULTSTATUS
Command DATA:	{}
Response DATA:	{lolock,fannormal,outputmute,normaltemp}

Any valid comma-separated combination of: lolock/lounlock, fannormal/fanstandby/fanfailed, outputnormal/outputoverdriven/outputmute, normaltemp/overtemp

NOTE: "fanstandby" means the fan(s) are purposely off to self-warm the BUC at low ambient temperature.

2.7.11 Get Latched Fault Status

Command Description:	Request the latched faults within the BUC. Request only latched faults, i.e., faults that have previously occurred but have not been cleared.
CMD_CODE:	18
Function Name:	GLFAULTSTATUS
Command DATA:	{}
Response DATA:	{outputoverdriven}

For status definitions, see *Get Fault Status* on page 19. The following faults should be latched: lounlock, fanfailed, outputoverdriven, overtemp.

2.7.12 Reset Status Latch

This command resets the fault status. Any fault event will be latched and readable, until the BUC receives this command. If the fault condition is persistent, this command will have no effect.

Command Description:	Clear all latched status
CMD_CODE:	9
Function Name:	RSTATUSLATCH
Command DATA:	{}
Response DATA:	{}

2.7.13 Get Software Version

Command Description:	Get the version number
CMD_CODE:	13
Function Name:	GSOFTWAREVER
Command DATA:	{}
Response DATA:	{03.0123}

2.7.14 BUC Reset

Command Description:	Reset the BUC
CMD_CODE:	17
Function Name:	BUC_RST
Command DATA:	{}
Response DATA:	{}

2.7.15 File Transfer

2.7.15.1 Define Transfer Type

There are two file types to be transferred: calibration files in XML format (see *Calibration File Format* on page 28) and software upgrade files in vendor-defined format. A file transfer is always initiated by the modem, using GCALIBFILE (see *Get Calibration File* on page 21), SCALIBFILE (see *Set Calibration File* on page 21), or PREP_UPGRADE (see *Prepare For Upgrade* on page 22). Next, the file transfer is performed (see *Set Calibration File* on page 21).

File names consist only of the characters allowed in the DATA field (see *DATA* on page 8), excluding comma. Filenames do not include any path name. Filenames are case-insensitive.

2.7.15.1.1 Get Calibration File

Command Description:	Prepare BUC to send the calibration file
CMD_CODE:	14
Function Name:	GCALIBFILE
Command DATA:	{}
Response DATA:	{buc.cal,6123} file name and file size (in bytes)

NOTE: After response, perform the file transfer (See *Execute Transfer* on page 22)

2.7.15.1.2 Set Calibration File

Command Description:	Prepare BUC to receive the calibration file
CMD_CODE:	15
Function Name:	SCALIBFILE
Command DATA:	{buc.cal,6123} file name and file size (in bytes)
Response DATA:	{}

NOTE: After response, perform the file transfer (See *Execute Transfer* on page 22)

2.7.15.1.3 Prepare For Upgrade

Command Description:	Prepare BUC to receive a software upgrade
CMD_CODE:	16
Function Name:	PREP_UPGRADE
Command DATA:	{buc.sw,127383} file name and file size (in bytes)
Response DATA:	{}

NOTE: After response, perform the file transfer (See *Execute Transfer* on page 22)

2.7.15.1.3.1 Software Upgrade Procedure

Both upgrades and downgrades must be supported. The BUC software must take measures to ensure that an invalid software image may not be used for a software upgrade. A successful Kermit transfer is necessary but not sufficient. The BUC software must ensure that the payload of the Kermit transfer is valid for the model of BUC being upgraded and that using that image will allow the BUC to start in a valid operating mode.

The serial protocol supports upgrading the software on the BUC. In order to perform a software upgrade of the BUC, the following procedure is performed:

- CM sends the PREP_UPGRADE command that specifies the filename used for upgrade and also its size in bytes.
- The BUC responds (see *RSP_CODE* on page 7) with RSP_CODE = 1 if it is ready to accept the upgrade; 0 if not ready.
- If a RSP_CODE = 0 is received, the satellite router retries the upgrade after a configurable timeout (default 30s).
- If a RSP_CODE = 1 is received, the satellite router starts a file transfer of the file with the name specified in the previous command.
- If the file transfer fails, the CM retries the file transfer after a configurable timeout (default 30s).
- If the file transfer succeeds, the satellite router resets the BUC using the BUC Reset command.
- The BUC is expected to boot with the new software version after the reset.
- The satellite router will restart communications with the BUC after a configurable timeout (default 60s).

2.7.15.2 Execute Transfer

2.7.15.2.1 Kermit

File Transfer uses Kermit, a combined network and serial communication software package offering a consistent, transport-independent, cross-platform approach to connection establishment, terminal sessions, file transfer, file management, character-set translation,

numeric and alphanumeric paging, and automation of file transfer and management, dialogs, and communication tasks through its built-in scripting language.

Not all Kermit implementations in the field behave identically. To ensure compatibility with iDirect terminals, Kermit file transfers over the serial port must work with C-Kermit 9.0, which is the de facto standard implementation of the Kermit file transfer protocol. Source code may be found here:

ftp://ftp.kermitproject.org/kermit/archives/cku302.tar.gz

2.7.15.2.2 Send File

Command Description:	Start the file transfer
CMD_CODE:	10
Function Name:	SENDFILE
Command DATA:	{buc.cal,6123} file name and file size (in bytes)
Response DATA:	{}

2.7.15.2.3 Receive File

Command Description:	Receive a file
CMD_CODE:	12
Function Name:	RECEIVEFILE
Command DATA:	{buc.cal,6123} file name and file size (in bytes)
Response DATA:	{}

2.7.15.2.4 Kermit File Transfer Status

Command Description:	Status of the latest Kermit file transaction
CMD_CODE:	11
Function Name:	STATUSKERMITFILE
Command DATA:	{buc.cal} file name
Response DATA:	{}

2.7.15.2.5 Kermit File Transfer Sequence, Modem to BUC

```
modem -------------- [SENDFILE] ----------->BUC
modem <-------------- [SENDFILE, 1] ----------BUC
modem -------------- [Kermit -s powercal.xml] ----------->BUC
modem -------------- [STATUSKERMITFILE] ----------->BUC
modem <-------------- [STATUSKERMITFILE, 1] ----------BUC
```

2.7.15.2.6 Kermit File Transfer Sequence, BUC to Modem

```
modem -------------- [RECEIVEFILE] ----------->BUC
modem <-------------- [RECEIVEFILE, 1] ----------BUC
modem -------------- [Kermit -r powercal.xml] ----------->BUC
```

```
modem -------------- [STATUSKERMITFILE] ----------->BUC
modem <-------------- [STATUSKERMITFILE, 1] ----------BUC
```

2.7.16 Vendor Specific Commands

In order to allow different BUC vendors to provide the end user with a richer command set beyond those required by the monitoring and control interface, the protocol allows for the definition of vendor-specific commands.

All command codes 0x60 to 0x7E shall be reserved for use by the vendor. The Manufacturer_ID field is the binary representation of the MANUFACTURER_ID assigned to the vendor. This ID is as specified in Section 2.7.1. The vendor is free to use any internal format for the commands in the vendor specific portion after the first two fields, as shown in the command format. It is suggested that these vendor-specific commands be provided as ASCII strings.

Example Client Message:

CMD_CODE (0x60 to 0x7E)	MANUFACTURER_ID	CMD_LEN	Vendor-specific Command (MAX LEN = 255 Bytes) (Preferably an ASCII String)

Example Server Message:

CMD_CODE (0xE0 to 0xFE)	MANUFACTURER_ID	CMD_LEN	Vendor-specific Command (MAX LEN = 255 Bytes) (Preferably an ASCII String)

3 Calibration

This chapter contains the following sections:

- Calibration Algorithm
- Calibration File Format

3.1 Calibration Algorithm

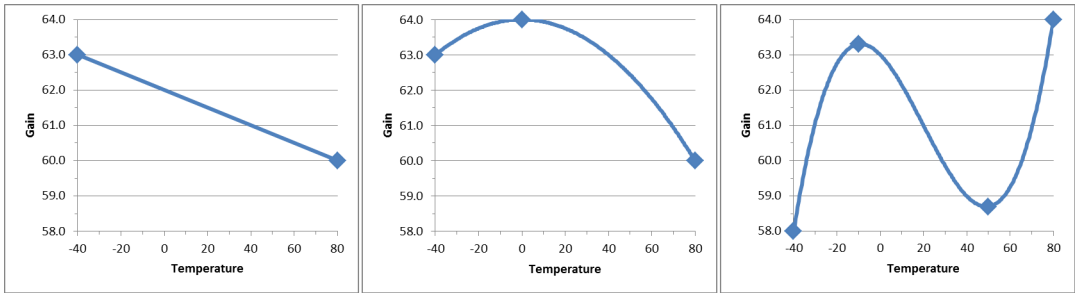
The calibration procedure as indicated below should be run by the BUC manufacturer. The BUC output power is factory-calibrated with IF stimulus at three temperature points: minimum operating temperature, maximum operating temperature, and nominal or average operating temperature. The BUC is calibrated at Maximum Output Power (MOP), [MOP-4dB], and [MOP-8dB] for better linearity interpolation by the modem. For maximum interpolation accuracy, the raw hex bits power reading from the ADC should be linear to BUC output power expressed in dB. The resultant calibration data is stored so the modem can electronically retrieve it via OpenBMIP™ query.

If the BUC supports a band-select filter, the filter should be configured to its low-band setting for all frequencies less than or equal to the transition frequency. The filter should be configured to its high-band (or wide-band) setting for all frequencies greater than the transition frequency. This will ensure consistent operation with the modem.

The BUC Calibration process is shown in Figure 3-1. Measurements are made at each point identified by target power level "p", modulation type "m", frequency "f", and temperature "t". At each measurement point, power is externally measured at the IF input, and at the RF output. Additionally, the BUC's internal detector analog to digital converter (ADC) is read with the BUC power amp enabled, and with the BUC power amp disabled. Finally, the BUC's internal temperature sensor (which should be thermally close to the power amp and power detector) is read. It is desirable to read the power detector and the temperature sensor as rapidly as possible (allowing for detector settling time) to minimize temperature changes between the associated readings.

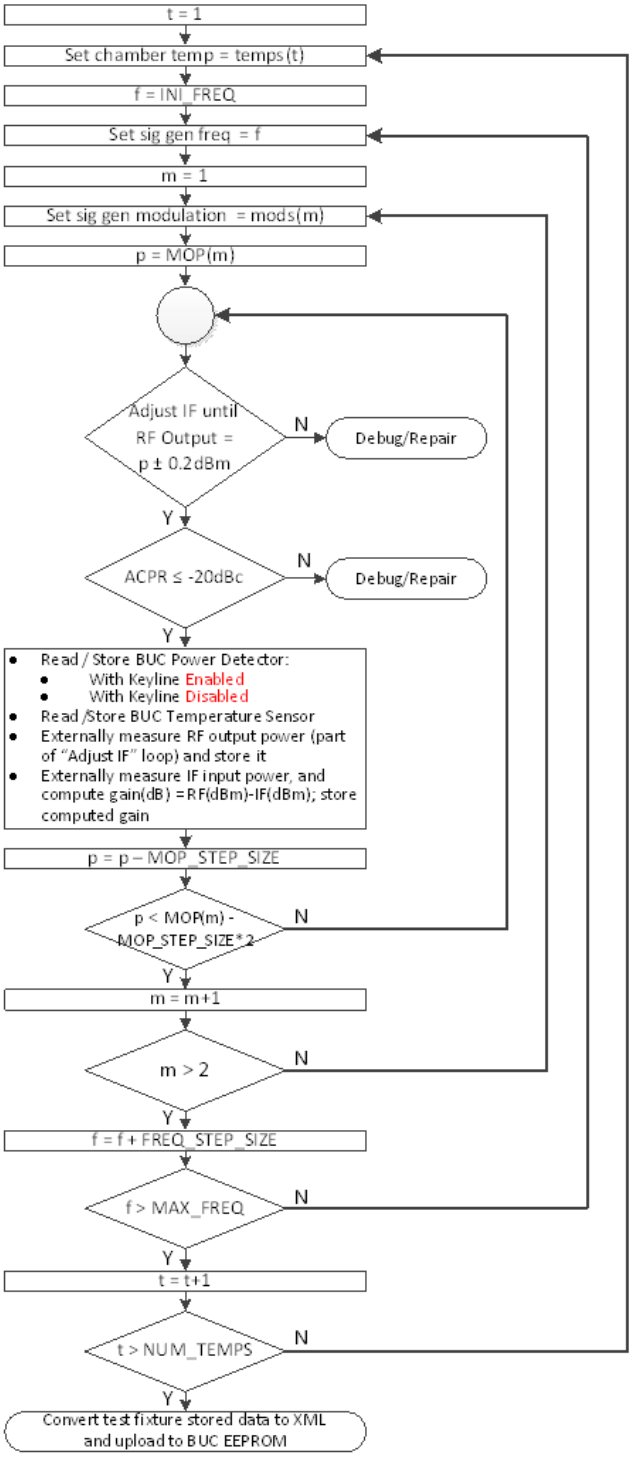
The measurement temperatures should be selected to include any peaks or troughs in the gain versus temperature curve, to minimize error in interpolation. In all cases, the endpoints (minimum and maximum operating temperatures) should be included, but data will be linearly extrapolated if temperatures beyond those in the CAL file are reported by the BUC. The following cases are anticipated:

Figure 3-1. Typical Gain vs Temperature for 2, 3, and 4 Points



NOTE: The temperatures t=1, t=2, and t=3 indicated in Figure 3-2 are for reference; systems may calibrate at the temperatures appropriate for the application. The nominal Maximum Output Power (MOP) indicated in the figure is for reference; systems may implement the MOP appropriate to the application. Other details, such as the frequency calibration points chosen, may be tailored for the application. The file format described in *Calibration File Format* on page 28 must be used in all implementations.

Figure 3-2. BUC Calibration Process



Example (maritime application):
Test chamber ambient temperatures temps(t):
t= 1:: +25 °C
t= 2:: -40 °C
t= 3:: +73 °C
t= 4:: (optional, e.g. -20 °C)
...

f in MHz, for example:
950, 1000, ..1950

mods(m):
m=1: BPSK
m=2: QPSK
m=3: 8PSK (Future)
Note:
8PSK is not required at this time

MOP(m) ≥ +37dBm for Q/8PSK
MOP(m) ≥ +36dBm for BPSK

Note:
BUC Detector Reading (HEX value) slope may be either positive or negative, but must change monotonically with increasing power, over the calibrated range, plus a margin of +0.75/-2.0dB.

3.2 Calibration File Format

The calibration file that will be retrieved from the BUC by the CM should bind to the following xml file format. The BUC manufacturer may optionally choose to compress the calibration file in GZIP format (see IETF RFC 1952, listed in the Standards References). The filename shall end with ".xml" if it is XML and ".gz" if it is a GZIP-compressed file containing the XML file.

In the sample calibration file shown below, only the characters enclosed within the demarcation characters need to be presented as-is. The demarcation characters are "<>" and "</>" respectively. The other information that is presented in this sample calibration file format are all "values" and not string constants. These need to be replaced with the actual measured values of that particular variable. The "values" are indicated as SOME_VARIABLE_NAME in the following format. These can be decimals, floats or strings as required; see Table 3-1 for descriptions.

Table 3-1. Calibration File Values

Name Of Value In Sample Format	Example Value	Units
<vendor_id>	99	BUC Manufacturer ID (MID) unique in network
<model_number>	1234567890xxx	BUC Part Number (BPN)
<serial_number>	A12345A33	BUC Serial Number (BSN)
<functional_id>	050300	BUC Functional ID (FID)
<max_operating_point>	33.0	dBm (for QPSK; nominal)
<back_off_step_size>	4	dB; difference between power steps
<initial_freq>	950	MHz
<freq_step_size>	50	MHz NOTE: freq_step_size is deprecated; the modem will ignore this value. Because each frequency is explicitly listed, the file is permitted to contain non-uniform steps. One typical use for this feature is to add finer steps to accommodate discontinuities in the frequency response, such as at the band select filter breakpoint.
<max_freq>	1950	MHz

Table 3-1. Calibration File Values (continued)

<buc_cutoff_frequency_mhz>	1350.0	MHz; Decision point for the BUC's internal 2nd-harmonic filter. When transmitting an IF frequency **less than or equal to** this value, the remote will set the filter to lowband; frequency **greater than** this value will set the BUC's filter to highband. If this value is absent, the remote will assume: mean(max_freq, min_freq) - 100MHz If the BUC does not have a switchable filter, this value can be set to equal <max_freq>.
<adc_res>	12	Bits; 12 bits is currently supported
<adc_slope>	-1	-1 or +1. "+1" means that a higher power output will produce a higher ADC number. Slope may be mildly nonlinear but must be monotonic, over the calibrated range, plus a margin of +0.75/-2.0dB.
<num_temps>	3	Number of temperature points, maximum **5**.
<mod_type>	1	1 (BPSK), 2 (QPSK), 3 (8PSK), 4(16PSK), 5(32PSK), 6(16QAM), 7(32QAM); 1 & 2 must be supported; 1, 2, and 3 are supported by the remote.
<freq>	950	MHz, IF input
<pout>	33.0	dBm, externally measured output
<pout_det>	7F5	Raw hex bits from ADC, PA enabled
<pout_det_dis>	82E	Raw hex bits from ADC, PA disabled
<gain>	62.4	dB, Externally measured gain
<buc_temp>	57.6	Degrees C near PA & detector

A generic sample BUC Calibration file is shown below.

NOTE: The Vendor_ID, MODEL_NUMBER, SERIAL NUMBER, FUNCT_ID definition/format can be found in *Get Product ID* on page 14. All items in UPPER CASE will be replaced with specific values.

```
<buc_cal_file>
  <vendor_id>MID</vendor_id>
  <model_number>BPN</model_number>
  <serial_number>BSN</serial_number>
```

<functional_id>**FID**</functional_id>

<max_operating_point>**MOP**</max_operating_point>

<back_off_step_size>**MOP_STEP_SIZE**</back_off_step_size>

<initial_freq>**INI_FREQ**</initial_freq>

<freq_step_size>**FREQ_STEP_SIZE**</freq_step_size>

<max_freq>**MAX_FREQ**</max_freq>

<buc_cutoff_frequency_mhz>**CUTOFF**</buc_cutoff_frequency_mhz>

<adc_res>**ADC_RES_IN_BITS**</adc_res>

<adc_slope>**ADC_SLOPE**</adc_slope>

<num_temps>**NUM_TEMPS**</num_temps >

<temperature1>

<modulation_setting>

<mod_type>**1**</mod_type>

<pow_set>

<freq>**INITIAL_FREQ**</freq>

<pout>**MOP**</pout>

<pout_det>**DET_ADC_HEX**</pout_det>

<pout_det_dis>**DET_ADC_HEX**</pout_det_dis>

<gain>**GAIN_IN_DB**</gain>

<pout>**MOP-MOP_STEP_SIZE**</pout>

<pout_det>**DET_ADC_HEX**</pout_det>

<pout_det_dis>**DET_ADC_HEX**</pout_det_dis>

<gain>**GAIN_IN_DB**</gain>

<pout>**MOP-MOP_STEP_SIZE*2**</pout>

<pout_det>**DET_ADC_HEX**</pout_det>

<pout_det_dis>**DET_ADC_HEX**</pout_det_dis>

<gain>**GAIN_IN_DB**</gain>

<buc_temp>**BUC_TEMPERATURE**</buc_temp>

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*2</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*3</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*4</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*6</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*7</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
    <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*9</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*10</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*11</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*12</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
```

```
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
```

```
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
```

```
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
```

```
        <buc_temp>BUC_TEMPERATURE</buc_temp>
      </pow_set>
    </modulation_setting>
    <modulation_setting>
      <mod_type>2</mod_type>
      <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
      </pow_set>
      <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*2</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*3</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*4</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*6</freq>
```

```
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*7</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
```

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*9</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*10</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*11</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*12</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
    <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
```

```
            <pout>MOP-MOP_STEP_SIZE*2</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
          </pow_set>
        </modulation_setting>
      </temperature1>
      <temperature2>
        <modulation_setting>
          <mod_type>1</mod_type>
          <pow_set>
            <freq>INITIAL_FREQ</freq>
            <pout>MOP</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE*2</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
          </pow_set>
          <pow_set>
            <freq>INITIAL_FREQ+FREQ_STEP_SIZE</freq>
```

```
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*2</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
```

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*3</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*4</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*6</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*7</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*9</freq>
      <pout>MOP</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*10</freq>
      <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*11</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*12</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
```

```
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
```

```
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE*2</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
        </pow_set>
        <pow_set>
            <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
            <pout>MOP</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE*2</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
        </pow_set>
    </modulation_setting>
```

```
<modulation_setting>
  <mod_type>2</mod_type>
  <pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
  </pow_set>
  <pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*2</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
```

```
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*3</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*4</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*6</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*7</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
```

```
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*9</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
```

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*10</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*11</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*12</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
      <pout>MOP</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
      <pout>MOP</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
    <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
          </pow_set>
        </modulation_setting>
      </temperature2>
      <temperature3>
        <modulation_setting>
          <mod_type>1</mod_type>
          <pow_set>
            <freq>INITIAL_FREQ</freq>
            <pout>MOP</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <pout>MOP-MOP_STEP_SIZE*2</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
            <gain>GAIN_IN_DB</gain>
            <buc_temp>BUC_TEMPERATURE</buc_temp>
          </pow_set>
          <pow_set>
            <freq>INITIAL_FREQ+FREQ_STEP_SIZE</freq>
            <pout>MOP</pout>
            <pout_det>DET_ADC_HEX</pout_det>
            <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*2</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*3</freq>
```

```
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*4</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
```

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*6</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*7</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*9</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*10</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*11</freq>
      <pout>MOP</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+FREQ_STEP_SIZE*12</freq>
      <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
      </pow_set>
      <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
      </pow_set>
    </modulation_setting>
    <modulation_setting>
      <mod_type>2</mod_type>
      <pow_set>
```

```
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*2</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
```

```
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*3</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*4</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*5</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*6</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*7</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*8</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+2 * FREQ_STEP_SIZE*9</freq>
      <pout>MOP</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <pout>MOP-MOP_STEP_SIZE*2</pout>
      <pout_det>DET_ADC_HEX</pout_det>
      <pout_det_dis>DET_ADC_HEX</pout_det_dis>
      <gain>GAIN_IN_DB</gain>
      <buc_temp>BUC_TEMPERATURE</buc_temp>
   </pow_set>
   <pow_set>
      <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*10</freq>
```

```
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*11</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
```

```
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+ FREQ_STEP_SIZE*12</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
  <freq>INITIAL_FREQ+FREQ_STEP_SIZE*13</freq>
  <pout>MOP</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE</pout>
  <pout_det>DET_ADC_HEX</pout_det>
  <pout_det_dis>DET_ADC_HEX</pout_det_dis>
  <gain>GAIN_IN_DB</gain>
  <pout>MOP-MOP_STEP_SIZE*2</pout>
  <pout_det>DET_ADC_HEX</pout_det>
```

```
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*14</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*15</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
```

```
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*16</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*17</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
```

```
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*18</freq>
    <pout>MOP</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <pout>MOP-MOP_STEP_SIZE*2</pout>
    <pout_det>DET_ADC_HEX</pout_det>
    <pout_det_dis>DET_ADC_HEX</pout_det_dis>
    <gain>GAIN_IN_DB</gain>
    <buc_temp>BUC_TEMPERATURE</buc_temp>
</pow_set>
<pow_set>
    <freq>INITIAL_FREQ+FREQ_STEP_SIZE*19</freq>
    <pout>MOP</pout>
```

```
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
    <pow_set>
        <freq>INITIAL_FREQ+FREQ_STEP_SIZE*20</freq>
        <pout>MOP</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <pout>MOP-MOP_STEP_SIZE*2</pout>
        <pout_det>DET_ADC_HEX</pout_det>
        <pout_det_dis>DET_ADC_HEX</pout_det_dis>
        <gain>GAIN_IN_DB</gain>
        <buc_temp>BUC_TEMPERATURE</buc_temp>
    </pow_set>
```

```
        </modulation_setting>
    </temperature3>
</buc_cal_file>
```

Appendix A Acronyms and Abbreviations

This list is meant to be generic within this document and may contain acronyms and abbreviations not found in this manual and some terms may not be defined based on industry standards.

0...9

16APSK Sixteen Amplitude and Phase Shift Keying

8PSK Eight Phase Shift Keying

A

A-TDMA Adaptive Time Division Multiple Access

ABS Automatic Beam Switching

AC Alternating Current

ACM Adaptive Coding and Modulation

ACS Antenna Control System

AES Advanced Encryption Standard

APSK Amplitude and Phase-shift keying

AZ Azimuth

B

BB BaseBand

BIM Below-Decks Interface Module

BIST Built-In Self-Test

BITE Built-In Test Equipment

BPN BUC Part Number

BPSK Binary Phase Shift Keying

BSN BUC Serial Number

BTP Burst Time Plan

BUC Block Up Converter

C

C/N Carrier to Noise ratio

CBIT Continuous Built In Test

CDR Critical Design Review

CIR Committed Information Rate

CPE Customer Premise Equipment

CPU Central Processing Unit

CRC Cyclic Redundancy Check

CSA Canadian Space Agency

D

DAC Digital to Analog Converter

dB deciBel

dBi deciBel isotropic

dBm deciBel milli-Watt

dBW deciBel Watt

DC Direct Current

DDR Double Data Rate

DHCP Dynamic Host Configuration Protocol

DNS Domain Name Service

DVB-S2 Digital Video Broadcasting over Satellite, Second Generation

E

EIRP Effective Isotropic Radiated Power

Eb/N0 Bit Energy to Noise Power Spectral Density ratio

EEPROM Electrically Erasable Programmable Read-Only Memory

EL Elevation

EMC ElectroMagnetic Compatibility

EMI ElectroMagnetic Interference

ER Embedded Router

ESR Embedded Service Router

ETSI European Telecommunications Standards Institute

F

FCC Federal Communication Commission

FEC Forward Error Correction

FID Functional ID

FMECA Failure Mode Effects Criticality Analysis

FPGA Field Programmable Gate Array

FS Functional Specification

G

G/T Gain over Temperature

GHz GigaHertz

GPIO General-Purpose Input/Output

GPS Global Positioning System

H

HCP High-Capacity Payload

I

IBIT Initiated Built In Test

ICD Interface Control Document

iDX Evolution Software System

IEC International Electrotechnical Commission

IFL Inter-Facility Link

IF Intermediate-frequency

IP Ingress Protection

IP Internet Protocol

IR Infrared

J

K

kbps kilobit per second

kHz kilohertz

KRFU Ku/Ka-band Radio Frequency Unit

ksps kilosymbol per second

L

LAN Local Area Network

LDP CLow-Density Parity Coding

LED Light Emitting Diode

LNB Low Noise Block Converter

LOS Loss of Signal

LRU Line-Replaceable Unit

M

Mbps Megabits per second

Mcps Megachips per second

MES Mobile Earth Station

MF-TDMA Multi-Frequency TDMA

MHz Megahertz

MID Manufacturer ID

MIL-STD US Military Standard

MODCOD Modulation and Coding

Msps Mega Symbols per Second

MTBF Mean Time Between Failures

MTBUR Mean Time Between Unscheduled Removals

N

NAND Not AND

NF Noise Figure

NORNot OR

NMS Network Management System

O

OAE Outside Antenna Equipment

ODU Outdoor Unit

OEM Original Equipment Manufactuer

OMT Orthogonal-Mode Transducer

OpenAMIP Open Antenna-Modem Interface Protocol

OTA Over The Air

OTP One Time Programmable

P

PA Power Amplifier

PAST Person-Activated Self-Test

PCB Printed Circuit Board

PDR Preliminary Design Review

PLL Phased Locked Loop

PSK Phase Shift Keying

PSU Power Supply Unit

Q

QEF Quasi Error Free

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

R

RF Radio Frequency

RGMII Reduced Gigabit Media Independent Interface

RMS Root Mean Square

RoHS Restriction of Hazardous Substances

ROM Read-Only Memory

RSSI Receive Signal Strength Indication

RTP Real-Time Protocol

Rx or RX Receive

S

SAS Satellite Access Station

SCPC Single Channel Per Carrier

SGMII Serial Gigabit Media Independent Interface

SIM Subscriber Identity Module

SNR Signal to Noise Ratio

SRS Systems Requirement Specification

SRU Shop Replaceable Unit

SSB Single Side Band

T

TBD To Be Determined

TBS To Be Supplied

TDMA Time Division Multiple Access

TFI Terminal Functional ID

TMI Terminal Manufacturer ID

TPCFEC Turbo Product Code FEC

TPN Terminal Part Number

TSN Terminal Serial Number

TTC Terminal Transmit Control

Tx or TX Transmit

U

UDP Universal Data Protocol

UL Underwriters Laboratories

UMD Update Manager Daemon

V

VAC Volts Alternating Current

VDC Volts Direct Current

VSAT Very Small Aperature Terminal

W

WFQ Weighted Fair Queuing

WGS Wideband Global SATCOM

X

Y

Z

Appendix B Glossary

2D 16-State Type of Forward Error Correction coding available on iDirect inbound carriers in DVB-S2 networks. 2D 16-State coding can provide better link margins, improved IP throughput and faster acquisition than Turbo Product Coding.

ABS See Automatic Beam Selection (ABS).

ACM See Adaptive Coding and Modulation (ACM).

ACM Gain The ACM Gain represents the increase in performance achieved on a DVB-S2 outbound carrier when the MODCOD used to transmit data is higher than the minimum MODCOD configured for the carrier.

Acquisition The process whereby the satellite router synchronizes its bursts with the upstream TDMA frame timing and joins an iDirect network.

Adaptive Coding and Modulation (ACM) Adaptive Coding and Modulation. A method of applying coding to a data stream in DVB-S2 networks in which every BBFrame can be transmitted on a different MODCOD.

Allocation Fairness Relative to CIR An iDirect Group QoS option which, when enabled, causes satellite bandwidth to be allocated in proportion to the configured CIR of the Group QoS node or virtual remote. When this option is not enabled, bandwidth is allocated equally to competing nodes until available bandwidth is exhausted.

Allocation Fairness Relative to MODCOD Applies only to DVB-S2 outbound carriers using Adaptive Coding and Modulation (ACM). An iDirect Group QoS option which, when enabled, causes satellite bandwidth allocation to be based on information rate rather than raw satellite bandwidth. This favors remotes at lower MODCODs, since their satellite bandwidth allocations must increase to achieve the same information rate as remotes at higher MODCODs.

Alternate Downstream Carrier An iDirect feature that allows iBuilder users to associate a second downstream carrier definition with an iDirect network in order to facilitate moving the network to a new downstream carrier.

Application	In iDirect Group QoS (GQoS), an Application defines a specific type of service (such as Voice over IP or TCP) as defined in a Service Group. An Application is created from an Application Profile. An instance of an Application running on a remote is called a Virtual Remote.
Automatic Beam Selection (ABS)	An iDirect feature that automates the process by which roaming remotes select which network to join and automatically lock on to the associated outbound carrier. Also known as Automatic Beam Switching.
Bandwidth Group	An intermediary iDirect Group QoS node. A Bandwidth Pool contains one or more Bandwidth Groups. Each Bandwidth Group Contains one or more Service Groups.
Bandwidth Pool	The root (or top-level node) of an iDirect Group QoS tree. A Bandwidth Pool can be either an iDirect Network (in which case it defines the QoS properties of the Downstream Carrier) or an Inroute Group (in which case it defines the QoS properties of the Upstream Carrier.)
Blade	Protocol Processor server machine.
Board Support Package (BSP)	Also know as Cumulative Update Package. Support package downloaded to iDirect remotes before loading remote image files.
BSP	See Board Support Package (BSP).
BTP	See Burst Time Plan (BTP).
Burst Time Plan (BTP)	Slot allocation message sent to remote modems to indicate when each remote can transmit on the TDMA upstream carriers.
CA	See Certificate Authority (CA).
CA Foundry	The iDirect Certificate Authority (CA) utility that issues the X.509 public key certificates that allow “hosts” to join an iDirect TRANSEC network.
CCM	See Constant Coding and Modulation (CCM).
Certificate Authority (CA)	An authority in a network that issues and manages security credentials and public keys for message encryption.
Constant Coding and Modulation (CCM)	A method of applying coding in a DVB-S2 data stream in which every BBFrame is transmitted at the same MODCOD.

Chassis Group	A group of iDirect chassis physically linked in a daisy chain allowing a single network to span multiple chassis.
Chassis License	A license purchased from iDirect required to activate slots in an iDirect chassis.
CIR	See Committed Information Rate (CIR).
CNO	See Customer Network Observer (CNO).
Committed Information Rate (CIR)	In iDirect's Group QoS, the guaranteed network bandwidth availability. CIR specifies an amount of bandwidth that is allocated to a node before additional (non-CIR) bandwidth is allocated to that node for traffic with the same priority.
Comms-on-the-MOVE (COTM)	iDirect's mobile remote feature.
COTM	See Comms-on-the-MOVE (COTM).
Cumulative Update Package	See Board Support Package (BSP).
Custom Key	An options file parameter configured on an iBuilder custom tab. Custom keys allow options to be configured on a remote or network that are not available on the GUI.
Customer Network Observer (CNO)	A member of a CNO User Group. A CNO can log on to iMonitor with read-only access to those network elements for which the CNO User Group has permissions. A CNO cannot log on to iBuilder and cannot make any changes to the system.
Daisy Chain	See Chassis Group.
Derived ID (DID)	The unique identifier of an iDirect remote satellite router derived from the model type and serial number.
Deterministic TDMA (DTDMA)	A technique used to prevent collisions of remotes transmitting simultaneously in which synchronized burst time plan provides the network timing.
DFOE	See Dynamic Features and Options Exchange (DFOE).
DID	See Derived ID (DID).
Distributed NMS (DNMS)	A single iDirect Network Management System (NMS) with software processes distributed across two or more server machines.

DNMS	See Distributed NMS (DNMS).
Downstream Carrier	Synonymous with Outbound Carrier. The satellite carrier transmitted from the hub to the remote satellite router.
DTDMA	See Deterministic TDMA (DTDMA)
DVB-S2	A set of open standards for satellite digital broadcasting. DVB-S2 is an extension to the widely-used DVB-S standard and was introduced in March 2005.
Dynamic Features and Options Exchange (DFOE)	A protocol used by the iDirect NMS to allow some remote-side configuration changes to be dynamically applied. Beginning with Release iDX 2.0, hub-side options groups beginning with 'RMT_' are sent from the Protocol Processor to the remote using the DFOE protocol. For these options, users are no longer required to apply remote-side changes to the remote in iBuilder.
EDAS Controller Board	Type of controller board used on older iDirect chassis. See also MIDAS Controller Board.
Eight-Port Switch	Configurable LAN switch available on some iDirect remote satellite router model types.
EIR	See Enhanced Information Rate (EIR).
Enhanced Information Rate (EIR)	In iDirect's Group QoS (GQoS), the EIR option allows you to configure the system to maintain CIR or MIR during rain fade for the physical remote (Remote-Based Group QoS) or critical applications (Application-Based Group QoS). EIR only applies to networks that use DVB-S2 with Adaptive Coding and Modulation (ACM).
Evolution Product Line	iDirect hub line card and remote satellite router hardware capable of processing DVB-S2 downstream carriers.
Fast Fade Margin	For iDirect DVB-S2 outbound carriers, the additional margin added to the SNR thresholds measured at hardware qualification to arrive at the operational threshold during a "fast fade" condition.
Feature License	A license purchased from iDirect allowing NMS operators to configure a license-controlled feature.
Filter Profile	A traffic profile configurable in iBuilder and assigned to remotes to filter out unwanted packets.

Free Slots	Slots left in the dynamic sub-frame after all stream, guaranteed (CIR) and preemptive bandwidth requests are satisfied. Free slots are allocated to all VSATs (up or down), except the master, in a round-robin fashion.
Frequency Hopping	The ability of iDirect remotes to switch between TDMA carriers within an inroute group when transmitting to the hub.
Full-Trigger CIR	Committed Information Rate (CIR) (CIR) that is always fully-allocated even if demand is less than the configured CIR.
Global NMS (GNMS)	An iDirect feature that allows iDirect roaming remotes to move from network to network.
GQoS	See Group QoS (GQoS).
GNMS	See Global NMS (GNMS).
Group QoS (GQoS)	iDirect's Quality of Service (QoS) solution based on a hierarchical tree structure by which satellite bandwidth allocation flows down the tree from the root node to the leaf nodes. GQoS allows advanced network operators a high degree of flexibility in creating subnetworks and groups of remotes with various levels of service tailored to the characteristics of the user applications being supported.
HLC	See Hub Line Card (HLC).
Hub Line Card (HLC)	An iDirect modem deployed at the hub to transmit and/or receive outroutes and inroutes.
Hub Network Operator (HNO)	An NMS operator with privilege to act as an administrator to Virtual Network Operators. An HNO can configure VNO users and networks and set VNO permissions such as visibility and read/write access.
iBuilder	iDirect's Network Management System GUI for configuring iDirect networks.
iMonitor	iDirect's Network Management System GUI for monitoring iDirect networks.
Inbound Carrier	Synonymous with Upstream Carrier. The carrier transmitted from the remote satellite router to the hub.
Indoor Unit (IDU)	The satellite modem and indoor devices (in contrast to Outdoor Unit or ODU).
Information Rate	The rate of transmission of user data over an upstream or downstream carrier including IP headers and iDirect overhead.

Inroute	A TDMA Upstream Carrier.
Inroute Group	A group of inroutes shared by a set of remotes in an iDirect network. Typically, a remote can frequency hop among the TDMA carriers within its inroute group.
iSite	iDirect's Network Management System GUI for commissioning remote modems and setting up and monitoring remote-to-remote connections.
iVantage	iDirect's complete suite of tools for configuring, monitoring, and controlling your iDirect satellite network.
LDCP	Low Density Parity Coding. The error correction coding scheme used in DVB-S2 networks.
LEGS	Lightweight Encapsulation for Generic Streams. An iDirect proprietary protocol for encapsulating data in DVB-S2 networks which maximizes the efficiency of data packing into BBFrames.
Maximum Information Rate (MIR)	In iDirect's Group QoS, MIR specifies the maximum amount of bandwidth that will be allocated to a node, regardless of demand generated by the node. A node with MIR set will never be granted more bandwidth than the configured MIR bit rate.
Maximum MODCOD	Modulation and Coding scheme, used in DVB-S2 networks.
MIDAS Controller Board	Type of controller board used on newer iDirect chassis. See also EDAS Controller Board.
MIN	See Minimum Information Rate (MIN).
Minimum Information Rate (MIN)	In Group QoS, MIR specifies the maximum amount of bandwidth that will be allocated to a node, regardless of demand generated by the node.
MIR	See Maximum Information Rate (MIR).
MODCOD	The combinations of Modulation Types and Error Coding schemes supported on a satellite channel. The higher the modulation the greater the number of bits per symbol (or bits per Hz).
MUSiC Box	iDirect hardware that allows a common antenna/electronics platform to be shared across multiple remotes that are at the same physical location.
Network Accelerator	iDirect hardware that maximizes the speed of encrypted traffic over iDirect secure networks.

Network Management System (NMS)	Software used by network operators to configure and manage their networks. iDirect's NMS consists of iBuilder, iMonitor and iSite and can be enhanced with a number of additional iVantage products.
NMS	See Network Management System (NMS).
Nominal MODCOD	In iDirect's DVB-S2 implementation, the Nominal MODCOD is the Reference Operating Point (ROP) for a remote receiving a downstream DVB-S2 carrier with ACM.
Options File	An iDirect configuration file generated by iBuilder. Options files are used to download configuration settings to protocol processors, hub line cards and remote satellite routers.
Outbound Carrier	See Downstream Carrier.
Outroute	See Outbound Carrier.
Protocol Processor	The iDirect Protocol Processor is a high performing, highly scalable core part of the iDirect hub, providing many critical functions. The protocol processor software is designed to scale and provide load balancing and automatic redundancy.
RCM	The Reference Clock Module on an iDirect chassis. RCMs are required for frequency hopping.
Remote Locking	An iDirect feature that allows individual remotes to be locked to a particular network. Once a remote is locked with a key, it only functions in a network with the same key.
RMA	Return Material Authorization. Issued by iDirect's TAC for equipment that must be returned for repair or replacement.
Roaming Remote	iDirect mobile remotes that use the Global NMS feature to "roam" from network to network around the globe. Roaming remotes are not constrained to a single location or limited to any geographic region.
SCPC	See Single Channel Per Carrier (SCPC).
Single Channel Per Carrier (SCPC)	User data is transmitted to the satellite continuously on a single satellite carrier and can be received by a single location (point-to-point link) or multiple locations (point-to-multipoint link).
SkyCelerator	iDirect's family of network accelerator products. See Network Accelerator.

SkyMonitor	An iDirect digital spectrum analyzer that is fully integrated with the NMS. Operators can use a SkyMonitor unit to view their iDirect inbound and outbound carriers, or to view other carriers present at the hub.
Sleep Mode	An iDirect feature that allows remote modems to conserve power consumption during periods of network inactivity.
Spread Spectrum	A transmission technique in which a pseudo-noise (PN) code is employed as a modulation waveform to "spread" the signal energy over a bandwidth much greater than the signal information bandwidth.
Steady State Margin	In DVB-S2 networks, the margin added to the SNR thresholds measured at hardware qualification to arrive at the operational SNR threshold during steady state operation.
Symbol Rate	The number of symbols that are transmitted in one second. From the symbol rate, calculate the bandwidth (total number of bits per second) by multiplying the bits per symbol by the symbol rate.
TAC	See Technical Assistance Center (TAC).
TDMA	See Time Division Multiple Access (TDMA).
Technical Assistance Center (TAC)	iDirect's customer service and technical support center, at http://tac.idirect.net or 703-648-8151. iDirect Government customer service and technical support center, at http://tac.idirectgov.com
Time Division Multiple Access (TDMA)	A type of over-the-air multiplexing by which two or more channels of information are transmitted simultaneously over the same link by allocating different time slots within TDMA frames for the transmission of each channel.
TRANSEC	Transmission security. The component of communications security that includes the application of measures designed to protect transmissions from interception and exploitation by means other than encryption.
Transmission Rate	A measure of the speed of all over-the-air data. This includes the user data (Information Rate), iDirect overhead, and FEC encoding bits.
Upstream Carrier	Synonymous with Inbound Carrier. The carrier transmitted from the remote satellite router to the hub.
Variable Coding and Modulation (VCM)	A method of applying coding to a DVB-S2 data stream in which MODCODs are assigned according to service type. iDirect does not support VCM.

VCM	See Variable Coding and Modulation (VCM).
Virtual Network Operator (VNO)	A member of a VNO User Group. A VNO User Group restricts visibility and access rights of group members based on the permissions granted to the group by the Hub Network Operator (HNO).
Virtual Remote	In iDirect Group QoS, an instance of a Group QoS Application running on a remote modem. In Application Based QoS mode, a remote has one Virtual Remote for each Application assigned to the remote. In Remote Based QoS mode, all Applications are combined into a single Virtual Remote.
VNO	See Virtual Network Operator (VNO).

iDirect
13861 Sunrise Valley Drive, Suite 300
Herndon, VA 20171-6126
+1 703.648.8000
+1 866.345.0983
www.idirect.net
Advancing a Connected World