# Si468x Programming Guide

## 1. Introduction

This document provides an overview of the programming requirements for the Si468x FM/HD/DAB receiver. The hardware control interface and software commands are detailed along with several examples of the required steps to configure the device for various modes of operation. Table 1 provides a programming guide cross-reference for each Si468x part released by Silicon Labs to date. This programming guide focuses on the (most current) Si468x-A10 release.

**Table 1. Si468x Programming Guide and Firmware Revisions[1,2]**

| Release Name | Release Index | Part | FM/FMHD Radio | DAB Radio | AM/AMHD Radio | Bootloader Full Patch | Mini-Patch | Programming Guide Revision |
|---|---|---|---|---|---|---|---|---|
| 120725 | 1 | A10 | 2.0.12 | | | | | 0.5 |
| 120731 | 2 | A10 | 2.0.12 | 0.0.2 | | | | 0.6 |
| 120914 | 3 | A10 | 2.0.12 | 0.0.6 | | | | 0.7 |
| 121120 | 4 | A10 | 2.0.12 | 1.0.4 | | | | 0.8 |
| 130215 | 5 | A10 | 3.0.11 | 2.0.3 | | ROM0.016 | ROM0.MINI.003 | 0.9 |
| 130524 | 6 | A10 | 3.0.16 | 3.0.5 | | ROM0.016 | ROM0.MINI.003 | 1.0 |
| 130927 | 7 | A10 | 3.0.16 | 3.2.0 | | ROM0.016 | ROM0.MINI.003 | 1.3 |
| 131122 | 8 | A10 | 3.0.17 | 3.2.0 | | ROM0.016 | ROM0.MINI.003 | 1.4 |
| 131209 | 9 | A10 | 3.0.17 | 3.2.1 | | ROM0.016 | ROM0.MINI.003 | 1.5 |
| 131213 | 10 | A10 | 3.0.17 | 3.2.1 | 0.0.6 | ROM0.016 | ROM0.MINI.003 | 1.6 |
| 140210 | 11 | A10 | 3.0.18 | 3.2.1 | 0.0.6 | ROM0.016 | ROM0.MINI.003 | 1.7 |
| 140708 | 12 | A10 | 3.0.19 | 3.2.7 | 1.0.5 | ROM0.016 | ROM0.MINI.003 | 1.8 |

**Notes:**
1. Grayed items are for evaluation or have been retired and should not be used for new designs.
2. If you have been provided a firmware release that is newer than those listed in this table, please see the firmware release notes for any necessary programming-related changes necessary until this guide has been updated (the new value is shown in this table).


Copyright © 2014 by Silicon Laboratories

## 2. Overview

This family of products is programmed using commands and responses. To perform an action, the system controller writes a command byte and associated arguments, which cause the device to execute the given command. The device will, in turn, provide a response depending on the type of command that was sent.

**Table 2. Product Family Function**

| Part Number | Description | Audio Output | FM | RDS | AM | HD Radio | DAB, DAB+ |
|---|---|---|---|---|---|---|---|
| Si4682 | FM/HD Radio Receiver w/ RDS | X | X | X | | X | |
| Si4683 | FM/AM/HD Radio Receiver w/ RDS | X | X | X | X | X | |
| Si4684 | FM/DAB Receiver w/ RDS | X | X | X | | | X |
| Si4688 | FM/HD Radio/DAB Receiver w/ RDS | X | X | X | | X | X |
| Si4689 | FM/AM/HD Radio/DAB Receiver w/ RDS | X | X | X | X | X | X |

# TABLE OF CONTENTS

**Section** **Page**

**1. Introduction . . . . . 1**
**2. Overview . . . . . 2**
**3. Terminology . . . . . 5**
**4. Commands/Properties Summary . . . . . 6**
4.1. Commands Summary—FMHD . . . . . 6
4.2. Properties Summary—FMHD . . . . . 62
4.3. Commands Summary—DAB . . . . . 117
4.4. Properties Summary—DAB . . . . . 170
4.5. Commands Summary—AMHD . . . . . 194
4.6. Properties Summary—AMHD . . . . . 246
**5. Flowcharts . . . . . 297**
5.1. Loading Firmware . . . . . 297
5.2. Data Service Receiver—FM–Band Scan . . . . . 298
5.3. Data Service Receiver—FM–RDS Data Service . . . . . 300
5.4. Data Service Receiver—FMHD–Band Scan . . . . . 301
5.5. Data Service Receiver—FMHD–Data Services . . . . . 303
5.6. Data Service Receiver—DAB–Band Scan . . . . . 306
5.7. Data Service Receiver—DAB–Data Services . . . . . 308
5.8. Updating the Boot Flash . . . . . 311
**6. Managing Firmware . . . . . 313**
6.1. Firmware Components . . . . . 313
6.2. System Architecture Options . . . . . 314
6.3. System Architecture Recommendations . . . . . 317
6.4. Updating NVSPI Flash . . . . . 317
6.5. Managing Firmware Subcommands . . . . . 318
6.6. Flash Pass-Through Properties . . . . . 329
**7. Digital Services User's Guide . . . . . 334**
7.1. Reference Material . . . . . 334
7.2. Definition of Terms . . . . . 334
7.3. Basics of the Si468x Digital Services Interface . . . . . 335
7.4. Basic Operation . . . . . 335
7.5. Finding a Digital Service . . . . . 336
7.6. Starting a Digital Service . . . . . 340
7.7. Getting the Digital Service Data . . . . . 341
7.8. DSRV Error Handling . . . . . 346
7.9. Data Service Formats . . . . . 346
7.10. DAB . . . . . 346
7.11. MOT . . . . . 348
7.12. HD Radio Data Services . . . . . 348
7.13. Audio Related Data Services . . . . . 348

7.14. DAB Program Associated Data (PAD) . . . . . 348
7.15. HD Radio PSD . . . . . 350
7.16. How to Use FMHD Fast Detect in FMHD P2 . . . . . 351
**8. BER Test Setup and Procedure . . . . . 352**
8.1. BER Test Setup . . . . . 352
8.2. DAB BER Test . . . . . 352
8.3. FMHD BER Test . . . . . 353
**9. POWER_UP—How to Determine Crystal-Related Parameters for Your Design . . . . . 354**
9.1. TR_SIZE and IBIAS settings . . . . . 354
9.2. IBIAS Reduction in Steady State . . . . . 359
9.3. CTUN settings . . . . . 360
9.4. External Clock Source . . . . . 361
**Document Change List . . . . . 362**
**Contact Information . . . . . 365**

## 3. Terminology

- **CTS**—Clear to send
- **STC**—Seek/Tune Complete
- **NVM**—Non-volatile internal device memory
- **Device**—Refers to the Receiver/Demodulator
- **System Controller**—Refers to the system microcontroller
- **CMD**—Command byte
- **ARGn**—Argument byte (n = 1 to 7)
- **STATUS**—Status word of 32 bits.
- **RESPn**—Response byte (n = 1 to 15)

# 4. Commands/Properties Summary

## 4.1. Commands Summary—FMHD

**Table 3. Commands Summary—FMHD**

| Cmd | Name | Description |
|---|---|---|
| 0x00 | RD_REPLY | Returns the status byte and data for the last command sent to the device. |
| 0x01 | POWER_UP | Power-up the device and set system settings. |
| 0x04 | HOST_LOAD | Loads an image from HOST over command interface |
| 0x05 | FLASH_LOAD | Loads an image from external FLASH over secondary SPI bus |
| 0x06 | LOAD_INIT | Prepares the bootloader to receive a new image. |
| 0x07 | BOOT | Boots the image currently loaded in RAM. |
| 0x08 | GET_PART_INFO | Reports basic information about the device. |
| 0x09 | GET_SYS_STATE | Reports system state information. |
| 0x0A | GET_POWER_UP_ARGS | Reports basic information about the device such as arguments used during POWER_UP. |
| 0x10 | READ_OFFSET | Reads a portion of response buffer from an offset. |
| 0x12 | GET_FUNC_INFO | Returns the Function revision information of the device. |
| 0x13 | SET_PROPERTY | Sets the value of a property. |
| 0x14 | GET_PROPERTY | Retrieve the value of a property. |
| 0x15 | WRITE_STORAGE | Writes data to the on board storage area at a specified offset. |
| 0x16 | READ_STORAGE | Reads data from the on board storage area from a specified offset. |
| 0x30 | FM_TUNE_FREQ | Tunes the FM receiver to a frequency in 10 kHz steps. |
| 0x31 | FM_SEEK_START | Initiates a seek for a channel that meets the validation criteria for FM. |
| 0x32 | FM_RSQ_STATUS | Returns status information about the received signal quality. |
| 0x33 | FM_ACF_STATUS | Returns status information about automatically controlled features. |
| 0x34 | FM_RDS_STATUS | Queries the status of RDS decoder and Fifo. |
| 0x35 | FM_RDS_BLOCKCOUNT | Queries the block statistic info of RDS decoder. |
| 0x80 | GET_DIGITAL_SERVICE_LIST | Gets a service list of the ensemble. |
| 0x81 | START_DIGITAL_SERVICE | Starts an audio or data service. |
| 0x82 | STOP_DIGITAL_SERVICE | Stops an audio or data service. |
| 0x84 | GET_DIGITAL_SERVICE_DATA | Gets a block of data associated with one of the enabled data components of a digital services. |
| 0x90 | HD_ACQUIRE | Begins the HD demodulator acquisition process. |
| 0x92 | HD_DIGRAD_STATUS | Returns status information about the digital radio and ensemble. |

**Table 3. Commands Summary—FMHD (Continued)**

| Cmd | Name | Description |
|---|---|---|
| 0x93 | HD_GET_EVENT_STATUS | Gets information about the various events related to the HD services. |
| 0x94 | HD_GET_STATION_INFO | Retrieves information about the ensemble broadcaster. |
| 0x95 | HD_GET_PSD_DECODE | Retrieves PSD information |
| 0x96 | HD_GET_ALERT_MSG | Retrieves the FMHD Alert message. |
| 0x97 | HD_PLAY_ALERT_TONE | Plays the HD Alert Tone. |
| 0x98 | HD_TEST_GET_BER_INFO | Reads the current BER information. |
| 0x99 | HD_SET_ENABLED_PORTS | Sets default ports retrieved after acquisition. |
| 0x9A | HD_GET_ENABLED_PORTS | Gets default ports retrieved after acquisition. |
| 0xE5 | TEST_GET_RSSI | Returns the reported RSSI in 8.8 format. |

### 4.1.1. FMHD Commands

**Command 0x00. RD_REPLY**

RD_REPLY command must be called to return the status byte and data for the last command sent to the device. This command is also used to poll the status byte as needed. To poll the status byte, send the RD_REPLY command and read the status byte. This can be done regardless of the state of the CTS bit in the status register. Please refer to individual command descriptions for the format of returned data. RD_REPLY is a hardware command and can be issued while device is powered down. For commands where the size of the response is returned, the user should send the RD_REPLY command to read the SIZE first. Each time the RD_REPLY command is sent, the STAUS bytes will still be returned.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x00 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA_0[7:0] | | | | | | | |
| RESP5 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |
| ERR_CMD | Command Error.<br>0: No error<br>1: Error. The previous command failed. |

| Name | Function |
|---|---|
| DACQINT | Digital radio link change interrupt indicator. Indicates that something in the digital radio ensemble acquisition status has changed.<br>Service via the DAB_DIGRAD_STATUS or the FMHD_DIGRAD_STATUS commands. |
| DSRVINT | Indicates that an enabled data component of one of the digital services requires attention. Service using the GET_DIGITAL_SERVICE_DATA command. |
| RSQINT | Received Signal Quality interrupt indicator.Indicates that a received signal metric is above or below a threshold defined by threshold properties.<br>Service via FM_RSQ_STATUS command. |
| RDSINT | RDS Data Interrupt indicator.<br>Service via [ref FM_RDS_STATUS]. |
| ACFINT | Automatically controlled features interrupt indicator. Indicates the one of the dynamically system modifiers has crossed a programmed threshold. Service via the ACF_STATUS command. |
| STCINT | Seek/Tune complete. Service with RSQ_STATUS command.<br>0: Tune complete has not been triggered. Do not send a new TUNE/SEEK command.<br>1: Tune complete has been triggered. It is safe to send a new TUNE/SEEK command. |
| DEVNTINT | Digital radio event change interrupt indicator. Indicates that a new event related to the digital radio has occurred.<br>Service via the DAB_GET_EVENT_STATUS or FMHD_GET_EVENT_STATUS commands. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| DSPERR | The DSP has encountered a frame overrun. This is a fatal error. |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |
| DATA_0[7:0] | First Data byte. |
| DATA_N[7:0] | Nth Data byte. |

### Command 0x01. POWER_UP

The POWER_UP initiates the boot process to move the device from power down to power up mode. There are two possible boot scenarios: Host image load and FLASH image load. When the host is loading the image the host first executes the POWER_UP command to set the system settings REF_CLK, etc). A LOAD_INIT command then prepares the bootloader to receive a new image. After the LOAD_INIT command, using the HOST_LOAD command loads the image into the device RAM. After the RAM is loaded the host issues the BOOT command. When booting a FLASH image the host issues the POWER_UP command to set the system settings. Then issues the FLASH_LOAD command to select and load the image from FLASH. Once the image is loaded the host sends the BOOT command to boot the application. Power-up is complete when the CTS bit is set. This command may only be sent while the device is powered down. Note: FLASH_LOAD is not supported in A0A or A0B revisions.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x01 | | | | | | | |
| ARG1 | CTSIEN | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ARG2 | 0 | 0 | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| ARG3 | 0 | IBIAS[6:0] | | | | | | |
| ARG4 | XTAL_FREQ[7:0] | | | | | | | |
| ARG5 | XTAL_FREQ[15:8] | | | | | | | |
| ARG6 | XTAL_FREQ[23:16] | | | | | | | |
| ARG7 | XTAL_FREQ[31:24] | | | | | | | |
| ARG8 | 0 | 0 | CTUN[5:0] | | | | | |
| ARG9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | 0 | 0 | | 0 | 0 | | |
| ARG12 | 0 | | | | 0 | 0 | | |
| ARG13 | 0 | IBIAS_RUN[6:0] | | | | | | |
| ARG14 | 0 | | | | | | | |
| ARG15 | 0 | | | | | | | |

| Name | Function |
|---|---|
| CTSIEN | The bootloader will toggle a host interrupt line when CTS is available.<br>0: Disable toggling host interrupt line.<br>1: Enable toggling host interrupt line. |

| Name | Function |
|---|---|
| CLK_MODE[5:4] | Choose clock mode. See refclk spec sheet for more information<br>0: Oscillator and buffer are powered down.<br>1: Reference clock generator is in crystal mode.<br>2: Oscillator is off and circuit acts as single ended buffer.<br>3: Oscillator is off and circuit acts as differential buffer. |
| TR_SIZE[3:0] | XOSC TR_SIZE. See refclk spec sheet for more information.<br>Range: 0-15 |
| IBIAS[6:0] | XTAL IBIAS current at startup. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 0 to 1270 µA.<br>Range: 0-127 |
| XTAL_FREQ[31:0] | XTAL Frequency in Hz. The supported crystal frequencies are: [5.4 MHz–6.6 MHz] [10.8 MHz–13.2 MHz] [16.8 MHz–19.8 MHz] [21.6 MHz–26.4 MHz] [27 MHz–46.2 MHz] |
| CTUN[5:0] | CTUN. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator.<br>Range: 0-63 |
| IBIAS_RUN[6:0] | XTAL IBIAS current at runtime, after the XTAL oscillator has stabilized. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 10 to 1270 µA. If set to 0, will use the same value as IBIAS.<br>Range: 0-127 |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |

| Name | Function |
|---|---|
| ERR_CMD | Error.<br>0: No error<br>1: Error. The previous command failed. If STATUS1 is non-zero, it provides an indication of the cause of the error. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP + command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| RSVD_STAT[5:4] | RFU (Reserved For Future Use). |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |

### Command 0x04. HOST_LOAD

HOST_LOAD loads an image from HOST over command interface. It sends up to 4096 bytes of application image to the bootloader. Note: This command is much more efficient when the image is sent as multiples of 4 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x04 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | IMAGE_DATA0[7:0] | | | | | | | |
| ARG5 | IMAGE_DATA1[7:0] | | | | | | | |
| ARG6 | IMAGE_DATA2[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG7 | IMAGE_DATA3[7:0] | | | | | | | |
| ARG8 | IMAGE_DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| IMAGE_DATA0[7:0] | First byte of data stream from boot_img. |
| IMAGE_DATA1[7:0] | Second byte of data stream from boot_img. |
| IMAGE_DATA2[7:0] | Third byte of data stream from boot_img. |
| IMAGE_DATA3[7:0] | Fourth byte of data stream from boot_img. |
| IMAGE_DATA_N[7:0] | Nth byte of data stream. Maximum of 4096 bytes of data per HOST_LOAD command. For best results, N should be a multiple of 4. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x05. FLASH_LOAD

FLASH_LOAD loads the firmware image from an externally attached SPI flash over the secondary SPI bus. The image must be contiguous on the flash. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: FLASH_LOAD is only supported after patching the bootloader.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x05 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG3 | 0x00 | | | | | | | |
| ARG4 | FLASH_START_ADDR[7:0] | | | | | | | |
| ARG5 | FLASH_START_ADDR[15:8] | | | | | | | |
| ARG6 | FLASH_START_ADDR[23:16] | | | | | | | |
| ARG7 | FLASH_START_ADDR[31:24] | | | | | | | |
| ARG8 | 0 | | | | | | | |
| ARG9 | 0 | | | | | | | |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | | | | | | | |

| Name | Function |
|---|---|
| FLASH_START_ADDR[31:0] | Flash byte starting address of image to load |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x06. LOAD_INIT

LOAD_INIT prepares the bootloader to receive a new image. It will force the bootloader state to waiting for a new LOAD command (HOST_LOAD or FLASH_LOAD.) LOAD_INIT command must always be sent prior to a HOST_LOAD or a FLASH_LOAD command. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x06 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x07. BOOT

BOOT command boots the image currently loaded in RAM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x07 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x08. GET_PART_INFO

GET_PART_INFO reports basic information about the device such as Part Number, Part Version, ROM ID, etc. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_PART_INFO command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x08 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | CHIPREV[7:0] | | | | | | | |
| RESP5 | ROMID[7:0] | | | | | | | |
| RESP6 | X | | | | | | | |
| RESP7 | X | | | | | | | |
| RESP8 | PART[7:0] | | | | | | | |
| RESP9 | PART[15:8] | | | | | | | |
| RESP10 | X | | | | | | | |
| RESP11 | X | | | | | | | |
| RESP12 | X | | | | | | | |
| RESP13 | X | | | | | | | |
| RESP14 | X | | | | | | | |
| RESP15 | X | | | | | | | |
| RESP16 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP17 | X | | | | | | | |
| RESP18 | X | | | | | | | |
| RESP19 | X | | | | | | | |
| RESP20 | X | | | | | | | |
| RESP21 | X | | | | | | | |
| RESP22 | X | | | | | | | |

| Name | Function |
|---|---|
| CHIPREV[7:0] | Chip Mask Revision |
| ROMID[7:0] | ROM Id |
| PART[15:0] | Part Number (decimal) |

### Command 0x09. GET_SYS_STATE

GET_SYS_STATE reports basic system state information such as which mode is active; FM, DAB, etc. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_SYS_STATE command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x09 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | IMAGE[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |

| Name | Function |
|---|---|
| IMAGE[7:0] | This field indicates which firmware image processed this command.<br>0: Bootloader is active<br>1: FMHD is active<br>2: DAB is active<br>3: TDMB or data only DAB image is active<br>4: FMHD Demod is active<br>5: AMHD is active<br>6: AMHD Demod is active |

## Command 0x0A. GET_POWER_UP_ARGS

GET_POWER_UP_ARGS reports basic information about the device such as which parameters were used during power up. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x0A | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | | | | | | | |
| RESP5 | X | X | X | X | X | X | X | X |
| RESP6 | X | X | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| RESP7 | X | IBIAS[6:0] | | | | | | |
| RESP8 | XTAL_FREQ[7:0] | | | | | | | |
| RESP9 | XTAL_FREQ[15:8] | | | | | | | |
| RESP10 | XTAL_FREQ[23:16] | | | | | | | |
| RESP11 | XTAL_FREQ[31:24] | | | | | | | |
| RESP12 | X | X | CTUN[5:0] | | | | | |
| RESP13 | X | X | X | X | X | X | X | X |
| RESP14 | X | | | | | | | |
| RESP15 | X | X | X | | X | X | | |
| RESP16 | X | | | | X | X | | |
| RESP17 | X | IBIAS_RUN[6:0] | | | | | | |

| Name | Function |
|---|---|
| CLK_MODE[5:4] | See POWER_UP command |
| TR_SIZE[3:0] | See POWER_UP command |
| IBIAS[6:0] | See POWER_UP command |
| XTAL_FREQ[31:0] | See POWER_UP command |
| CTUN[5:0] | See POWER_UP command |
| IBIAS_RUN[6:0] | See POWER_UP command |

### Command 0x10. READ_OFFSET

READ_OFFSET is used for applications that cannot read the entire response buffer. This type of application can use this command to read the response buffer in segments. The host must pass in an offset from the beginning of the response buffer to indicate the starting point from which to read. This offset must be modulo 4. The response buffer remains intact as in the READ_REPLY command so that the response can be read again if needed. This function is available for both I2C and SPI mode. This is a software command, therefore it is best to read as much data in each calling as possible. This will reduce the overhead associated with using this command. It is recommended that the minimum reply size be on the order of 512 bytes. This means that for APIs that return less

the 512 bytes the standard READ_REPLY should be used. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x10 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The offset from the beginning of the response buffer from where to begin reading. The OFFSET parameter must be modulo four. An error is returned otherwise. For example to read a 1024 byte response in two chucks the host can call the READ_MORE command twice. The first call would have OFFSET set to 0 and the second call would have OFFSET set to 512. In both cases the host will clock out 516 bytes of data. The first chunk will include 4 bytes for the status word plus 512 bytes of response. The second chunk will include 4 bytes of status word plus the remaining 512 bytes of response. The response sections will be concatenated to form the entire response. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the data chunk. |

### Command 0x12. GET_FUNC_INFO

GET_FUNC_INFO returns the function revision number for currently loaded firmware (FMHD, DAB, AM etc.) as opposed to GET_PART_INFO command that provides the revision number for the combo firmware. For example, GET_PART_INFO would return A0B is the firmware revision while GET_FUNC_INFO would return 1.0.4 for FM function revision if the currently running firmware function is FM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x12 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | REVEXT[7:0] | | | | | | | |
| RESP5 | REVBRANCH[7:0] | | | | | | | |
| RESP6 | REVINT[7:0] | | | | | | | |
| RESP7 | NOSVN | X | LOCATION[1:0] | | X | X | MIXE-DREV | LOCALMOD |
| RESP8 | SVNID[7:0] | | | | | | | |
| RESP9 | SVNID[15:8] | | | | | | | |
| RESP10 | SVNID[23:16] | | | | | | | |
| RESP11 | SVNID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| REVEXT[7:0] | Major revision number (first part of 1.2.3). |

| Name | Function |
|---|---|
| REVBRANCH[7:0] | Minor revision number (second part of 1.2.3). |
| REVINT[7:0] | Build revision number (third part of 1.2.3). |
| NOSVN | If set the build was created with no SVN info. This image cannot be tracked back to the SVN repo. |
| LOCATION[5:4] | The location from which the image was built (Trunk, Branch or Tag).<br>0x0: The image was built from an SVN tag. Revision numbers are valid.<br>0x1: The image was built from an SVN branch. Revision numbers will be 0.<br>0x2: The image was built from the trunk. Revision number will be 0. |
| MIXEDREV | If set, the image was built with mixed revisions. |
| LOCALMOD | If set, the image has local modifications. |
| SVNID[31:0] | SVN ID from which the image was built. |

### Command 0x13. SET_PROPERTY

SET_PROPERTY sets the value of a property. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x13 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |
| ARG4 | DATA0[7:0] | | | | | | | |
| ARG5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| PROPID[15:0] | The property ID of the property to set. |
| DATA0[15:0] | Value for the written property. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x14. GET_PROPERTY

GET_PROPERTY retrieves the value of a property or properties. The host may read as many properties as desired up to the end of a given property group. An attempt to read passed the end of the property group will result in zeros being read. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x14 | | | | | | | |
| ARG1 | COUNT[7:0] | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |

| Name | Function |
|---|---|
| COUNT[7:0] | The number of properties to read. |
| PROPID[15:0] | The id of the property to retrieve. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |
| RESP5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[15:0] | Value of the first property. |

### Command 0x15. WRITE_STORAGE

WRITE_STORAGE writes data to the no board storage area at the specified offset. the largest block of data that can be written at one time is 256 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x15 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |
| ARG4 | LENGTH[7:0] | | | | | | | |
| ARG5 | LENGTH[15:8] | | | | | | | |
| ARG6 | 0x00 | | | | | | | |
| ARG7 | 0x00 | | | | | | | |
| ARG8 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer at which the data will be written. |

| Name | Function |
|---|---|
| LENGTH[15:0] | The number of bytes to write. |
| DATA0[7:0] | The first data byte to write. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x16. READ_STORAGE

READ_STORAGE reads data from the on board storage area from a specified offset. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x16 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer from which the data will be read. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the read data. |

## Command 0x30. FM_TUNE_FREQ

FM_TUNE_FREQ tunes the FM receiver to a frequency in 10 kHz steps. The optional STC interrupt is set when the command completes the tune. Sending this command clears any pending STCINT, RDSINT, and RSQINT bit in STATUS. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x30 | | | | | | | |
| ARG1 | 0 | 0 | OPEN_LOOP | FORCE_WB | TUNE_MODE[1:0] | | INJECTION[1:0] | |
| ARG2 | FREQ[7:0] | | | | | | | |
| ARG3 | FREQ[15:8] | | | | | | | |
| ARG4 | ANTCAP[7:0] | | | | | | | |
| ARG5 | ANTCAP[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OPEN_LOOP | Disables the hardware PLL in favor of a software AFC loop. This feature can be used during analog only broadcasts to reduce power consumption.<br>0: Normal operation.<br>1: Disable the PLL |
| FORCE_WB | Forces the DFE into wideband mode.<br>0: Normal operation.<br>1: Force Wide bandwidth |
| TUNE_MODE[3:2] | Set the desired tuning mode.<br>0: Tune and render analog audio as fast as possible, do not attempt to acquire HD.<br>1: Reserved<br>2: Tune and render analog audio as fast as possible, try to acquire HD and cross-fade to HD if acquired. Always selects the Main Program Service. To select an SPS use the START_DIGITAL_SERVICE command after the HD has been acquired.<br>3: Tune and attempt to acquire HD and render audio if successful. The Main Program Service (MPS) will be rendered. To select an SPS use the START_DIGITAL_-SERVICE command after the HD has been acquired. |
| INJECTION[1:0] | Injection selection<br>0: Automatic injection selection.<br>1: Low-side injection.<br>2: High-side injection. |
| FREQ[15:0] | Frequency in multiples of 10.0 kHz added to a starting frequency of 0 Hz |
| ANTCAP[15:0] | When non-zero this parameter sets the antenna tuning capacitor value to (ANTCAP-1)*250 fF (31.75 pF Max).<br>Range: 0-128<br>0: Automatically determines the cap setting. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

**Command 0x31. FM_SEEK_START**

FM_SEEK_START begins searching for a valid station. The search starts at FM_RSQ_STATUS:READFREQ + FM_SEEK_FREQUENCY_SPACING in the specified direction. In order for a station to be considered valid, each of the following thresholds must be met: FM_VALID_SNR_THRESHOLD, FM_VALID_RSSI_THRESHOLD, and FM_VALID_MAX_TUNE_ERROR. Clears any pending STCINT, RSQINT, or RDSINT interrupt status. Seek can be canceled through setting the CANCEL bit in the FM_RSQ_STATUS command. The optional STC interrupt is set when the command completes. Seek is currently performed in non-HD mode only. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: This command is not implemented in Si46xx-A0A or Si46xx-A0B releases. Note: The OPEN_LOOP, FORCE_WB, TUNE_MODE, INJECTION, and ANTCAP parameters will not be functional until the Si46xx-A0C release.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x31 | | | | | | | |
| ARG1 | 0 | 0 | OPEN_LOOP | FORCE_WB | TUNE_MODE[1:0] | | INJECTION[1:0] | |
| ARG2 | 0 | 0 | 0 | 0 | 0 | 0 | SEEKUP | WRAP |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | ANTCAP[7:0] | | | | | | | |
| ARG5 | ANTCAP[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OPEN_LOOP | Disables the hardware PLL in favor of a software AFC loop. This feature can be used during analog only broadcasts to reduce power consumption.<br>0: Normal operation.<br>1: Disable the PLL |
| FORCE_WB | Forces the DFE into wideband mode.<br>0: Normal operation.<br>1: Force Wide bandwidth |
| TUNE_MODE[3:2] | Set the desired tuning mode.<br>0: Tune and render analog audio as fast as possible, do not attempt to acquire HD.<br>1: Reserved<br>2: Tune and render analog audio as fast as possible, try to acquire HD and cross-fade to HD if acquired. Always selects the Main Program Service.<br>3: Tune and attempt to acquire HD and render audio if successful. After the station has been acquired, use START_DIGITAL_SERVICE to select services. |
| INJECTION[1:0] | Injection selection<br>0: Automatic injection selection.<br>1: Low-side injection.<br>2: High-side injection. |

| Name | Function |
|---|---|
| SEEKUP | determines direction of seek and band limit.<br>0: Seek down. Band limit is FM_SEEK_BAND_BOTTOM.<br>1: Seek up. Band limit is FM_SEEK_BAND_TOP. |
| WRAP | determines seek behavior upon reaching a band limit.<br>0: Halt seek at band limit.<br>1: When band limit is hit, continue seek from opposite band limit. |
| ANTCAP[15:0] | When non-zero this parameter sets the antenna tuning capacitor value to (ANTCAP-1)*250 fF (31.75 pF Max).<br>Range: 0-128<br>0: Automatically determines the cap setting. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

**Command 0x32. FM_RSQ_STATUS**

FM_RSQ_STATUS returns status information about the received signal quality. This command returns the Received Signal Strength Indicator (RSSI), Signal to Noise Ratio (SNR), frequency offset (FREQOFF), and Multipath (MULT) associated with the desired channel. It also indicates valid channel (VALID) and AFC rail status (AFCRL). This command can be used to check if the received signal is above the RSSI high threshold as reported by RSSIHINT or below the RSSI low threshold as reported by RSSILINT. It can also be used to check if the signal is above the SNR high threshold as reported by SNRHINT or below the SNR low threshold as reported by SNRLINT. It can be used to check if the detected multipath is above the Multipath high threshold as reported by MULTHINT or below the Multipath low threshold as reported by MULTLINT. The command clears the RSQINT, BLENDINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT, MULTHINT, and MULTLINT interrupt bits when the RSQACK bit of ARG1 is set. These are sticky meaning they will remain set until RSQACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in FM_RSQ_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Multipath metric and related interrupts/thresholds are not implemented in Si46xx-A0A or Si46xx-A0B releases.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x32 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | RSQACK | ATTUNE | CANCEL | STCACK |

| Name | Function |
|---|---|
| RSQACK | Clears RSQINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT if set. |
| ATTUNE | Return the values as of FM_VALID_SNR_TIME after tune. Only the signal quality metrics RSSI, SNR, ISSI, ASSI, MULT, DEV and the status bits INJECT, AFCL, and VALID are affected by setting this bit.<br>0: Return the current status<br>1: Return the snapshot taken at FM_VALID_SNR_TIME |
| CANCEL | Aborts a seek currently in progress.<br>0: Don't abort<br>1: Abort |
| STCACK | Clears the STC interrupt status indicator if set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | SNRHINT | SNRLINT | RSSI-HINT | RSSILINT |
| RESP5 | BLTF | X | X | X | X | X | AFCRL | VALID |
| RESP6 | READFREQ[7:0] | | | | | | | |
| RESP7 | READFREQ[15:8] | | | | | | | |
| RESP8 | FREQOFF[7:0] | | | | | | | |
| RESP9 | RSSI[7:0] | | | | | | | |
| RESP10 | SNR[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP11 | MULT[7:0] | | | | | | | |
| RESP12 | READANTCAP[7:0] | | | | | | | |
| RESP13 | READANTCAP[15:8] | | | | | | | |
| RESP14 | X | | | | | | | |
| RESP15 | HDLEVEL[7:0] | | | | | | | |
| RESP16 | X | | | | | | | |
| RESP17 | X | | | | | | | |
| RESP18 | X | | | | | | | |
| RESP19 | X | | | | | | | |

| Name | Function |
|---|---|
| SNRHINT | Indicates SNR above FM_RSQ_SNR_HIGH_THRESHOLD. |
| SNRLINT | Indicates SNR below FM_RSQ_SNR_LOW_THRESHOLD. |
| RSSIHINT | Indicates RSSI above FM_RSQ_RSSI_HIGH_THRESHOLD. |
| RSSILINT | Indicates RSSI below FM_RSQ_RSSI_LOW_THRESHOLD. |
| BLTF | Band Limit after FM_SEEK_START. Reports if a seek hit the band limit or wrapped to the original frequency. This does not indicate that the seek failed, only that further seeks from the current location would be unproductive.<br>Tune Failed after FM_TUNE_FREQ tuning with CONDITIONALRETURN bit set, reports '1' if the current station failed to meet the associated criteria and the original frequency was returned to. |
| AFCRL | AFC rail indicator. |
| VALID | Reports if the channel is valid based on the settings of FM_VALID_RSSI_THRESHOLD, FM_VALID_SNR_THRESHOLD, FM_VALID_MAX_TUNE_ERROR |
| READFREQ[15:0] | Returns the currently tuned frequency. If the ATTUNE option is used, returns the frequency of the last completed tune. Applies during a normal tune and during seek. |
| FREQOFF[7:0] | Signed frequency offset in BPPM (2 PPM)<br>Range: -128-127 |
| RSSI[7:0] | Received signal strength indicator in dBµV.<br>Range: -128-127 |
| SNR[7:0] | RF SNR indicator in dB.<br>Range: -128-127 |
| MULT[7:0] | Multipath indicator.<br>Range: 0-255 |

| Name | Function |
|---|---|
| READANTCAP[15:0] | Returns the antenna tuning cap value. |
| HDLEVEL[7:0] | Reports the percentage of the HD OFDM symbols that appear to contain HD information in the upper and lower sidebands when measured over a given number of symbol periods. See FM_RSQ_HD_DETECTION to configure this metric.<br>Range: 0-100 |

### Command 0x33. FM_ACF_STATUS

FM_ACF_STATUS returns status information about automatically controlled features of the device. The automatically controlled features include blend, high cut, and softmute. The bits BLEND_INT, HIGHCUT_INT, and SMUTE_INT are sticky meaning they will remain set until ACFACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire. See the FM_ACF_INTERRUPT_SOURCE property for information on enabling the ACFINT The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x33 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ACFACK |

| Name | Function |
|---|---|
| ACFACK | Clears ACFINT and any ACF interrupt bits if set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | X | BLEND_INT | HIGH-CUT_INT | SMUTE_INT |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP5 | X | BLEND_-CONV | HIGH-CUT_-CONV | SMUTE_CONV | X | BLEND_STATE | HIGH-CUT_STATE | SMUTE_STATE |
| RESP6 | X | X | X | ATTN[4:0] | | | | |
| RESP7 | HIGHCUT[7:0] | | | | | | | |
| RESP8 | PILOT | STBLEND[6:0] | | | | | | |
| RESP9 | X | | | | | | | |
| RESP10 | X | | | | | | | |

| Name | Function |
|---|---|
| BLEND_INT | Indicates that stereo separation has crossed below the blend threshold set by FM_ACF_BLEND_THRESHOLD. |
| HIGHCUT_INT | Indicates that the highcut cutoff frequency has crossed below the highcut threshold as set by FM_ACF_HIGHCUT_THRESHOLD. |
| SMUTE_INT | Indicates that softmute attenuation has increased above the softmute threshold as set by FM_ACF_SOFTMUTE_THRESHOLD. |
| BLEND_CONV | Stereo blend convergence indicator.<br>0: Audio blend is in transition.<br>1: Audio blend is not in transition and is within the tolerance set by FM_ACF_BLEND_TOLERANCE. |
| HIGHCUT_CONV | High Cut convergence indicator.<br>0: The high cut filter is in transition.<br>1: The high cut filter is not in transition and is within the tolerance set by FM_ACF_HIGHCUT_TOLERANCE. |
| SMUTE_CONV | Softmute convergence indicator.<br>0: Softmute is in transition.<br>1: Softmute is not in transition and is within the tolerance set by FM_ACF_SOFT-MUTE_TOLERANCE. |
| BLEND_STATE | Stereo blend indicator.<br>0: Audio is not blended (full stereo).<br>1: Audio is blended. |
| HIGHCUT_STATE | High Cut indicator.<br>0: Audio is not frequency limited<br>1: Audio is frequency limited. |
| SMUTE_STATE | Soft Mute indicator.<br>0: Audio is not soft muted.<br>1: Audio is soft muted. |

| Name | Function |
|---|---|
| ATTN[4:0] | Soft mute attenuation level in dB.<br>Range: 0-31 |
| HIGHCUT[7:0] | Hicut cutoff frequency in units 100Hz.<br>Range: 10-200 |
| PILOT | Stereo pilot indicator<br>0: Stereo pilot is not present.<br>1: Stereo pilot is present. |
| STBLEND[6:0] | Indicates the current stereo separation in dB.<br>STBLEND will only be non-zero if PILOT=1 |

### Command 0x34. FM_RDS_STATUS

FM_RDS_STATUS returns RDS information for current channel and reads an entry from the RDS FIFO. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: FM_RDS_STATUS command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x34 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | STATU-SONLY | MTFIFO | INTACK |

| Name | Function |
|---|---|
| STATUSONLY | Determines if data should be removed from the RDS receive FIFO.<br>0: Data in BLOCKA, BLOCKB, BLOCKC, BLOCKD and BLE contain the oldest data in the RDS FIFO and are removed from the FIFO.<br>1: Data in BLOCKA, BLOCKB, BLOCKC, BLOCKD and BLE will contain the last valid block A data received for the current station. Data in BLOCKB will contain the last valid block B data received for the current station. Data in BLE will describe the bit errors for the data in BLOCKA and BLOCKB. |
| MTFIFO | Clears the FIFO counts of the RDS Receive FIFO to zero if set. The FIFO counts will always be cleared during FM_TUNE_FREQ and FM_SEEK_START. |
| INTACK | Clears the STATUS:RDSINT bit if set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | X | X | X | RDSTPPTYINT | RDSPIINT | X | RDSSYNCINT | RDSFIFOINT |
| RESP5 | X | X | X | TPPTYVALID | PIVALID | X | RDSSYNC | RDSFIFOLOST |
| RESP6 | X | X | TP | PTY[4:0] | | | | |
| RESP7 | X | | | | | | | |
| RESP8 | PI[7:0] | | | | | | | |
| RESP9 | PI[15:8] | | | | | | | |
| RESP10 | RDSFIFOUSED[7:0] | | | | | | | |
| RESP11 | BLEA[1:0] | | BLEB[1:0] | | BLEC[1:0] | | BLED[1:0] | |
| RESP12 | BLOCKA[7:0] | | | | | | | |
| RESP13 | BLOCKA[15:8] | | | | | | | |
| RESP14 | BLOCKB[7:0] | | | | | | | |
| RESP15 | BLOCKB[15:8] | | | | | | | |
| RESP16 | BLOCKC[7:0] | | | | | | | |
| RESP17 | BLOCKC[15:8] | | | | | | | |
| RESP18 | BLOCKD[7:0] | | | | | | | |
| RESP19 | BLOCKD[15:8] | | | | | | | |

| Name | Function |
|---|---|
| RDSTPPTYINT | Traffic Program (TP) flag and Program Type (PTY) code has changed. |
| RDSPIINT | Program Identification (PI) code has changed. |
| RDSSYNCINT | RDS synchronization status changed. |

| Name | Function |
|---|---|
| RDSFIFOINT | RDS Data was received and the RDS receive FIFO is full or has at least FM_RDS_INTERRUPT_FIFO_COUNT entries. |
| TPPTYVALID | Indicates that the TP flag and PTY code are valid. |
| PIVALID | Indicates that the PI code is valid. |
| RDSSYNC | Indicates that RDS is currently synchronized. |
| RDSFIFOLOST | One or more RDS groups have been discarded due to FIFO overrun since the last call to FM_RDS_STATUS. |
| TP | Current channel's TP flag if RDSTPPTY is 1 |
| PTY[4:0] | Current channel's PTY code if RDSTPPTY is 1 |
| PI[15:0] | Current channel's Program Identification if RDSPI is 1 |
| RDSFIFOUSED[7:0] | Indicates number of groups remaining in the RDS FIFO (0 if empty). If this is non-zero, BLOCKA-BLOCKD contain the oldest entry in the FIFO and RDSFIFOUSED will decrement by one on the next call to RDS_FIFO_STATUS (assuming no new RDS Data is received in the interim). |
| BLEA[7:6] | Bit Errors corrected in BLOCKA<br>0: Block received with no bit errors.<br>1: Block had one or two bit errors corrected.<br>2: Block had three, four or five bit errors corrected.<br>3: Block is uncorrectable. |
| BLEB[5:4] | Bit Errors corrected in BLOCKB |
| BLEC[3:2] | Bit Errors corrected in BLOCKC |
| BLED[1:0] | Bit Errors corrected in BLOCKD |
| BLOCKA[15:0] | Block A data from RDS FIFO if STATUSONLY is 0. |
| BLOCKB[15:0] | Block B data from RDS FIFO if STATUSONLY is 0. |
| BLOCKC[15:0] | Block C data from RDS FIFO if STATUSONLY is 0. |
| BLOCKD[15:0] | Block D data from RDS FIFO if STATUSONLY is 0. |

### Command 0x35. FM_RDS_BLOCKCOUNT

FM_RDS_BLOCKCOUNT command queries the block statistic info of RDS decoder. This command returns RDS expected, received and uncorrectable, block statistic information. Information from this command can be reset by setting CLEAR bit or sending FM_TUNE_FREQ command. Once EXPECTED saturates at 65535, all other block count statistics will be frozen until the counts are cleared. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: FM_RDS_BLOCKCOUNT command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x35 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CLEAR |

| Name | Function |
|---|---|
| CLEAR | Clears the block counts if set. The current block counts will be reported before they are cleared. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | EXPECTED[7:0] | | | | | | | |
| RESP5 | EXPECTED[15:8] | | | | | | | |
| RESP6 | RECEIVED[7:0] | | | | | | | |
| RESP7 | RECEIVED[15:8] | | | | | | | |
| RESP8 | UNCORRECTABLE[7:0] | | | | | | | |
| RESP9 | UNCORRECTABLE[15:8] | | | | | | | |

| Name | Function |
|---|---|
| EXPECTED[15:0] | Number of expected RDS blocks. |
| RECEIVED[15:0] | Number of received RDS blocks. Under ideal conditions, EXPECTED and RECEIVED would be identical. The difference between these two numbers is the number of blocks lost. |

| Name | Function |
|---|---|
| UNCORRECT-ABLE[15:0] | Number of uncorrectable RDS blocks. These blocks have been received, but were found to have uncorrectable errors. The block error rate (BLER) is calculated by: BLER=(UNCORRECTABLE+(EXPECTED-RECEIVED))/EXPECTED |

### Command 0x80. GET_DIGITAL_SERVICE_LIST

GET_DIGITAL_SERVICE_LIST gets a service list of the ensemble. This command should be issued each time an audio or data service list is updated as indicated by the SVRLISTINT bit of the DAB_GET_EVENT_STATUS command. This occurs shortly after tune time when a digital radio tuning mode is selected and the ensemble has been acquired. Please refer to the (DAB Service List Information) section of this document for the format of DAB Service List. Please refer to iBiquity document: RX_IDD_2206 Appendix L (Get_All_Data_Services_Info and Get_All_Audio_Services_Info) for the format of the HD Radio Service List. In the case of HD this command also retrieves the audio or data service info when the appropriate service type option is selected. This service info is available whenever the AINFO or DINFO bit(s) are set in the HD_GET_EVENT_STATUS response. This audio and data information provides a quick look at the services in the ensemble and can be used to reduce scan time as this information is ready for parsing well before the service lists. The payload of these responses are defined in Table 5-4 of the RX_IDD_2206 main document. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x80 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | SERTYPE[1:0] | |

| Name | Function |
|---|---|
| SERTYPE[1:0] | Sets the type of service list to retrieve.<br>0: Get the audio(HD) or complete(DAB, DMB) service list.<br>1: Get the data(HD) service list.<br>2: Get the HD audio info.<br>3: Get the HD data info. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | SIZE[7:0] | | | | | | | |
| RESP5 | SIZE[15:8] | | | | | | | |
| RESP6 | DATA_0[7:0] | | | | | | | |
| RESP7 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| SIZE[15:0] | The size of the service list in bytes. |
| DATA_0[7:0] | A max of 2047 bytes of service information. Please see the supplemental Digital Services User's Guide. |
| DATA_N[7:0] | A max of 2047 bytes of service information. Please see the supplemental Digital Services User's Guide. N(max) = SIZE-2. |

### Command 0x81. START_DIGITAL_SERVICE

START_DIGITAL_SERVICE starts an audio or data service. This command is used for both DAB and HD audio and data services. To determine what services exist in an ensemble please use the GET_DIGITAL_SERVICE_LIST command. In HD radio applications the broadcaster does not always transmit this service information. In this case no data services are available but there may be multiple audio programs available. To view which audio services are available use the FMHD_DIGRAD_STATUS or AMHD_DIGRAD_STATUS command's AUDIO_PROG_AVAIL field to see which audio programs can be selected. In addition the SERVICE_ID (service number) is not required when selecting an audio or data service. In this case please set the SERVICE_ID parameter to 0. I the case of starting an audio service, it is not required to stop a currently running audio service/ program before starting a new one. The currently running audio service will be stopped automatically when the new service is requested. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x81 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SER-TYPE |
| ARG2 | 0x00 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service to start for FMHD/AMHD applications. For DAB/DMB applications there is no distinction between audio and data services so this parameter should be written as 0.<br>0: Select an audio service.<br>1: Select a data service. |
| SERVICE_ID[31:0] | The service's Service ID (DAB) or Service Number (FMHD/AMHD). This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>**Note:** For FMHD/AMHD applications the is referred to as the service number and not the global service ID that is provided as part of the component information. |
| COMP_ID[31:0] | The service's component ID (DAB) or Port/Program Number (FMHD/AMHD). This ID is found in the component section of the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

### Command 0x82. STOP_DIGITAL_SERVICE

STOP_DIGITAL_SERVICE stops an audio or data service. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x82 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SER-TYPE |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service list to start.<br>0: Select an audio service.<br>1: Select a data service. |
| SERVICE_ID[31:0] | The service's service ID. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command. |
| COMP_ID[31:0] | The service's Component ID or Port Number of the service to stop. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x84. GET_DIGITAL_SERVICE_DATA

GET_DIGITAL_SERVICE_DATA gets a block of data associated with one of the enabled data components of a digital service. Information about this block of data is found in the data header that is returned at the beginning of the data block. In order to determine the ideal number of PAYLOAD bytes to read, the header information can be read first followed by a second read of the full (header + PAYLOAD) length—it is unnecessary to call GET_DIGITAL_SERVICE_DATA twice to use this method. The data associated with this transaction will be discarded at the receipt of a next GET_DIGITAL_SERVICE_DATA command if STATUS_ONLY = 0. Reading past the end of the buffer will result in zeros for the invalid bytes. Please refer to the (DAB Service List Information) section of this document for the format of DAB Service List. Please refer to iBiquity document: SY_IDD_1019s Rev F (sections 5 and 6) for the format of the HD Radio data service data. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x84 | | | | | | | |
| ARG1 | 0 | 0 | 0 | STA-TUS_ONLY | 0 | 0 | 0 | ACK |

| Name | Function |
|---|---|
| STATUS_ONLY | Returns only the interrupt source and available buffers information. No digital service data is removed from the service queue. This option should be used when polling for available data. |

| Name | Function |
|---|---|
| ACK | Acknowledge the reading of a data buffer or acknowledge one of the error interrupts. Whenever the ACK bit is set the DSRVINT bit in the status register will be cleared before CTS is released unless there are more data blocks to be read. All interrupt source bits associated with this command will also be cleared.<br>0: Don't acknowledge the interrupt<br>1: Acknowledging the interrupt will clear the DSRVINT bit and the interrupt source bits. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | X | X | DSRVOVFLINT | DSRVPCKTINT |
| RESP5 | BUFF_COUNT[7:0] | | | | | | | |
| RESP6 | SRV_STATE[7:0] | | | | | | | |
| RESP7 | DATA_TYPE[7:0] | | | | | | | |
| RESP8 | SERVICE_ID[7:0] | | | | | | | |
| RESP9 | SERVICE_ID[15:8] | | | | | | | |
| RESP10 | SERVICE_ID[23:16] | | | | | | | |
| RESP11 | SERVICE_ID[31:24] | | | | | | | |
| RESP12 | COMP_ID[7:0] | | | | | | | |
| RESP13 | COMP_ID[15:8] | | | | | | | |
| RESP14 | COMP_ID[23:16] | | | | | | | |
| RESP15 | COMP_ID[31:24] | | | | | | | |
| RESP16 | RFU[7:0] | | | | | | | |
| RESP17 | RFU[15:8] | | | | | | | |
| RESP18 | BYTE_COUNT[7:0] | | | | | | | |
| RESP19 | BYTE_COUNT[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP20 | SEG_NUM[7:0] | | | | | | | |
| RESP21 | SEG_NUM[15:8] | | | | | | | |
| RESP22 | NUM_SEGS[7:0] | | | | | | | |
| RESP23 | NUM_SEGS[15:8] | | | | | | | |
| RESP24 | PAYLOAD0[7:0] | | | | | | | |
| RESP25 | PAYLOADN[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DSRVOVFLINT | The data services system has overflowed. This indicates that the host processor is not reading the services data out of the device fast enough. At most 8 outstanding data blocks can be queued in the device. |
| DSRVPCKTINT | Data for an enabled data service is ready for transfer to the host. |
| BUFF_COUNT[7:0] | Indicates the remaining number of buffers in the data service buffer queue. |
| SRV_STATE[7:0] | The status indicator for the associated service component.<br>0: Indicates that the service is playing out normally.<br>1: Indicates that the data service has stopped and that this is the last data block associated with the service.<br>2: Indicates that the system was not able to forward a packet due to a memory overflow. If this status is present the host is not reading the packet data fast enough.<br>3: Indicates the this data packet represents the beginning of a new data object.<br>4: Indicates the this data packet was received with errors. |
| DATA_TYPE[7:0] | RFU. |
| SERVICE_ID[31:0] | The Service ID this data is associated with. |
| COMP_ID[31:0] | The Component ID or Port Number this data is associated with. |
| RFU[15:0] | Reserved for future use. |
| BYTE_COUNT[15:0] | The length of this data block in bytes excluding the DSRV header. |
| SEG_NUM[15:0] | The segment number for this data block. If the data is associated with a stream this value represents a sequence number. Note that segment numbers can be returned out of order. Therefore the host must store and reassemble the data as needed. |
| NUM_SEGS[15:0] | The total number of segments to be returned for this data object. If NUM_SEGS=0 then no object length information is known or the associated service is a stream. |
| PAYLOAD0[7:0] | The first payload byte. |
| PAYLOADN[7:0] | The Nth payload byte where N = BYTE_COUNT-1. |

### Command 0x90. HD_ACQUIRE

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x90 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | FREQ[7:0] | | | | | | | |
| ARG3 | FREQ[15:8] | | | | | | | |

| Name | Function |
|---|---|
| FREQ[15:0] | Frequency in multiples of 10.0 kHz added to a starting frequency of 0 Hz |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x92. HD_DIGRAD_STATUS

HD_DIGRAD_STATUS returns status information about the digital radio and ensemble. The bits AERRHINT, AEERLINT, CDNRHINT, CDNRLINT, and ACQINT are sticky meaning they will remain set until DIGRAD_ACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in HD_DIGRAD_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x92 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DIGRAD_ACK |

| Name | Function |
|---|---|
| DIGRAD_ACK | Clears all pending digital radio interrupts. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | HDLOGOINT | SRCANAINT | SRCDIGINT | X | X | ACQINT | CDNRHINT | CDNRLINT |
| RESP5 | HDLOGO | SRCANA | SRCDIG | X | X | ACQ | CDNRH | CDNRL |
| RESP6 | BCTL[1:0] | | DAAI[5:0] | | | | | |
| RESP7 | CDNR[7:0] | | | | | | | |
| RESP8 | DIG_GAIN[7:0] | | | | | | | |
| RESP9 | AUDIO_PROG_AVAIL[7:0] | | | | | | | |
| RESP10 | AUDIO_PROG_PLAYING[7:0] | | | | | | | |
| RESP11 | AUDIO_CA[7:0] | | | | | | | |
| RESP12 | CORE_AUDIO_ERR[7:0] | | | | | | | |
| RESP13 | CORE_AUDIO_ERR[15:8] | | | | | | | |
| RESP14 | CORE_AUDIO_ERR[23:16] | | | | | | | |
| RESP15 | CORE_AUDIO_ERR[31:24] | | | | | | | |
| RESP16 | ENH_AUDIO_ERR[7:0] | | | | | | | |
| RESP17 | ENH_AUDIO_ERR[15:8] | | | | | | | |
| RESP18 | ENH_AUDIO_ERR[23:16] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP19 | ENH_AUDIO_ERR[31:24] | | | | | | | |
| RESP20 | PTY[7:0] | | | | | | | |
| RESP21 | PS_MODE[7:0] | | | | | | | |
| RESP22 | CODEC_MODE[7:0] | | | | | | | |

| Name | Function |
|---|---|
| HDLOGOINT | Indicates a change in the HD Logo display state. |
| SRCANAINT | Indicates a change in the Analog Audio Source state. Setting of this bit will only occur when the blend state of the analog audio changes. |
| SRCDIGINT | Indicates a change in the Digital Audio Source state. Setting of this bit will only occur when the blend state of the digital audio changes. |
| ACQINT | Indicates a change in the ensembles acquisition state. Setting of this bit will only occur when the acquisition state of the ensemble changes. |
| CDNRHINT | Indicates that CDNR has risen above the CDNR high threshold. This threshold is set in the HD_DIGRAD_CDNR_HIGH_THRESHOLD property. |
| CDNRLINT | Indicates that CDNR has dropped below the CDNR low threshold. This threshold is set in the HD_DIGRAD_CDNR_LOW_THRESHOLD property. |
| HDLOGO | HD Logo display<br>0: Consumer radio should not display HD Logo<br>1: Consumer radio should display HD Logo |
| SRCANA | Audio source is analog. If both SRCANA and SRCDIG are set, the audio is blending.<br>0: Audio source is not analog<br>1: Audio source is analog |
| SRCDIG | Audio source is digital. If both SRCDIG and SRCANA are set, the audio is blending.<br>0: Audio source is not digital<br>1: Audio source is digital |
| ACQ | The ensemble has been acquired. |
| CDNRH | Indicates that CDNR is currently above the CDNR high threshold. This threshold is set in the HD_DIGRAD_CDNR_HIGH_THRESHOLD property. |
| CDNRL | Indicates that CDNR is currently below the CDNR low threshold. This threshold is set in the HD_DIGRAD_CDNR_LOW_THRESHOLD property. |
| BCTL[7:6] | Reflects the state of the TX blend control bits communicated by the broadcast system. |

| Name | Function |
|---|---|
| DAAI[5:0] | Digital Audio quality indicator, Indicates the current estimate of the audio quality of the currently selected audio service. lower numbers indicate poorer quality. The number is unitless and provided in 4.2 format. |
| CDNR[7:0] | Indicates the current estimate of the carrier to noise ratio as seen by the digital radio subsystem. |
| DIG_GAIN[7:0] | Indicates the amount of gain the receiver must apply to the digital audio signal relative to analog audio. |
| AUDIO_PROG_AVAIL[7:0] | Indicates which of the 8 audio programs are available. This field is a bit field where bit 0 represents the MPS and bits 1-7 represent the SPS. For example if this field is set to 0x83 then SPS7, SPS1, and the MPS are available. To select one of these services use the START_DIGITAL_SERVICE command. |
| AUDIO_PROG_PLAYING[7:0] | Indicates which of the audio programs is currently playing. |
| AUDIO_CA[7:0] | Indicates which of the audio programs have CA associated with them. |
| CORE_AUDIO_ERR[31:0] | Indicates the total number of core audio frames received in error. |
| ENH_AUDIO_ERR[31:0] | Indicates the total number of enhanced audio frames received in error. |
| PTY[7:0] | Indicates the current audio program type PTy. |
| PS_MODE[7:0] | Indicates the primary service mode for the currently tuned ensemble. |
| CODEC_MODE[7:0] | Indicates the audio codec configuration for the currently selected audio program.<br>0: FM Hybrid single-stream on P1 Channel<br>2: AM Hybrid/All Digital dual-stream<br>10: FM dual-stream on SPS<br>13: FM Hybrid/All Digital single-stream on P3 Channel |

### Command 0x93. HD_GET_EVENT_STATUS

HD_GET_EVENT_STATUS retrieves the status of HD related events. This includes items such as new alarms available, new PSD, New station info, etc. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x93 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EVENT_ACK |

| Name | Function |
|---|---|
| EVENT_ACK | Clears all pending digital radio event interrupts. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | DINFOINT | AINFOINT | X | ALERTINT | PSDINT | SISINT | DSRVLISTINT | ASRVLISTINT |
| RESP5 | DINFO | AINFO | X | X | PSD | SIS | DSRVLIST | ASRVLIST |
| RESP6 | ASRVLISTVER[7:0] | | | | | | | |
| RESP7 | ASRVLISTVER[15:8] | | | | | | | |
| RESP8 | DSRVLISTVER[7:0] | | | | | | | |
| RESP9 | DSRVLISTVER[15:8] | | | | | | | |
| RESP10 | X | X | X | SIS_LOCATION | X | SIS_LONG_NAME | SIS_SHORT_NAME | SIS_ID |
| RESP11 | X | X | SIS_SLOGAN | SIS_BASIC_SIS | SIS_UNIV_SHORT_NAME | SIS_LEAP_SEC | SIS_TIMEZONE | SIS_MESSAGE |
| RESP12 | X | TEXT | SHORT | LANG | GENRE | ALBUM | ARTIST | TITLE |
| RESP13 | ID | OWNER | DESC | NAME | RECV | URL | VALID | PRICE |
| RESP14 | X | X | X | X | X | ARMSGAVAIL | ARFRM0 | ARFRM |
| RESP15 | ARFRAMECNT[7:0] | | | | | | | |
| RESP16 | ARMSGID[7:0] | | | | | | | |
| RESP17 | ARCRC7[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DINFOINT | New Data Info. Indicates that the data service info has been updated. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| AINFOINT | New Audio Info. Indicates that the audio service info has been updated. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| ALERTINT | New Alert interrupt. Indicates that a new alert has been issued by the broadcaster. Retrieve the event info with the HD_GET_ALERT_MSG command. |
| PSDINT | New Program Service Data interrupt. Indicates that new program service data on the currently playing audio service is available. Retrieve the PSD with the HD_GET_PSD_DECODE command. |
| SISINT | New station information interrupt. Indicates that the station information service data has changed. The updated information is retrieved with the HD_GET_STATION_INFO command. |
| DSRVLISTINT | New data service list interrupt. Indicates that a new digital data service list is available. The new service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. |
| ASRVLISTINT | New audio service list interrupt. Indicates that a new digital audio service list is available. The new service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. |
| DINFO | Data Service Info is available. Indicates data service info is available. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| AINFO | Audio Service Info is available. Indicates audio service info is available. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| PSD | Program Service Data is available. Indicates program service data on the currently playing audio service is available. Retrieve the PSD with the HD_GET_PSD_DECODE command. |
| SIS | Station information is available. Indicates that station information service data is available. The station information service data is retrieved with the HD_GET_STATION_INFO command. |
| DSRVLIST | Data Service list available. Indicates that a digital data service list is available. |
| ASRVLIST | Audio Service list available. Indicates that a digital audio service list is available. |
| ASRVLISTVER[15:0] | Indicates the current version of the audio service list. This field is incremented by 1 each time the audio service list is updated. The host can use this field to help determine if a new audio service list needs to be collected. |
| DSRVLISTVER[15:0] | Indicates the current version of the data service list. This field is incremented by 1 each time the data service list is updated. The host can use this field to help determine if a new data service list needs to be collected. |
| SIS_LOCATION | station location is available. |
| SIS_LONG_NAME | station long name is available. |

| Name | Function |
|---|---|
| SIS_SHORT_NAME | station short name is available. |
| SIS_ID | station ID is available. |
| SIS_SLOGAN | station slogan is available. |
| SIS_BASIC_SIS | the logical OR of bits SIS_LOCATION, SIS_LONG_NAME, SIS_SHORT_NAME and SIS_ID. |
| SIS_UNI-V_SHORT_NAME | station universal short name is available. |
| SIS_LEAP_SEC | station leap seconds is available. |
| SIS_TIMEZONE | station time zone is available. |
| SIS_MESSAGE | station message is available. |
| TEXT | PSD Comment:The actual text is available |
| SHORT | PSD Comment:Short Content Description is available |
| LANG | PSD Comment:Language is available |
| GENRE | PSD Genre is available |
| ALBUM | PSD Album is available |
| ARTIST | PSD Artist is available |
| TITLE | PSD Title is available |
| ID | PSD UFID:Owner ID is available |
| OWNER | PSD UFID:Owner is available |
| DESC | PSD Commercial:Description is available |
| NAME | PSD Commercial:Name of Seller is available |
| RECV | PSD Commercial:Received As is available |
| URL | PSD Commercial:Contact URL is available |
| VALID | PSD Commercial:Valid Until is available |
| PRICE | PSD Commercial:Price is available |
| ARMSGAVAIL | Complete AR MSG is available |
| ARFRM0 | AR Frame 0 has been received |
| ARFRM | AR Frame (any) has been received |
| ARFRAMECNT[7:0] | The Alert message frame counter. |
| ARMSGID[7:0] | A complete alert message is available. The message can be retrieved with the HD_GET_ALERT_MSG command. |
| ARCRC7[7:0] | The alert message CRC7 value extracted from frame 0. |

### Command 0x94. HD_GET_STATION_INFO

HD_GET_STATION_INFO retrieves information about the ensemble broadcaster. The station information is defined in the 2206 standard. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x94 | | | | | | | |
| ARG1 | INFO_SELECT[7:0] | | | | | | | |

| Name | Function |
|---|---|
| INFO_SELECT[7:0] | Selects which information returned by this command.<br>0: RSVD Do not use<br>1: Returns a station message. An arbitrary text message such as telephone number, URL, etc. High-priority messages should take precedence over all other SIS data and PSD. For the station message, a maximum length of 190 bytes is currently supported. The Station Message is only returned when updates have been received OTA. A LENGTH of 0 will be returned if no update has been received since the last request.<br>2: Returns the Local time zone and daylight savings time (DST) information, allowing receivers to automatically calculate and display time of day.<br>3: Returns a leap-second correction factor occasionally adjusts UTC by one second to keep it synchronized with astronomical time. Since GPS time does not apply this correction, the two standards have diverged slightly over the years.<br>4: Provides an alternative to basic short station name, for countries where different text-encoding schemes may be required. The Append Byte indicates whether "-FM" should be appended to the short station name by the HC. The maximum length for the name is 12 bytes. For US markets, the Station Name (short form) should be retrieved using then BASICSIS option.<br>5: Returns basic SIS data. See the 2206 documentation for full details. Basically this includes items such as the station ID and station location. The information returned by this option is controlled by the HD_EVENT_SIS_CONFIG property.<br>6: Returns the station slogan. This field supersedes the long station name returned by the BASIC SIS option 5. For the slogan, a maximum length of 95 bytes is currently supported. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | LENGTH[15:8] | | | | | | | |
| RESP6 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[15:0] | The number of data bytes returned by this command. |
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 SIS documentation for details about then returned data. |

## Command 0x95. HD_GET_PSD_DECODE

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x95 | | | | | | | |
| ARG1 | PROGRAM[7:0] | | | | | | | |
| ARG2 | FIELD[7:0] | | | | | | | |

| Name | Function |
|---|---|
| PROGRAM[7:0] | Program Number<br>0: MPS<br>1: SPS 1<br>2: SPS 2<br>3: SPS 3<br>4: SPS 4<br>5: SPS 5<br>6: SPS 6<br>7: SPS 7<br>0xFF: Currently playing audio service |
| FIELD[7:0] | PSD Field<br>0: Title<br>1: Artist<br>2: Album<br>3: Genre<br>4: Comment: Language<br>5: Comment: Short Content Description<br>6: Comment: The actual text<br>8: Commercial: Price<br>9: Commercial: Valid Until<br>10: Commercial: Contact URL<br>11: Commercial: Received As<br>12: Commercial: Name of Seller<br>13: Commercial: Description<br>14: UFID: Owner<br>15: UFID: Owner ID |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | SEQUENCE[7:0] | | | | | | | |
| RESP5 | SEQUENCE[15:8] | | | | | | | |
| RESP6 | DATATYPE[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP7 | LENGTH[7:0] | | | | | | | |
| RESP8 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| SEQUENCE[15:0] | Sequence number for this group of PSD |
| DATATYPE[7:0] | 0: 8 bit character: ISO/IEC 8859-1:1998<br>1: 16 bit character: ISO/IEC 10646-1:2000 |
| LENGTH[7:0] | The number of data bytes returned by this command. |
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 SIS documentation for details about then returned data. |

### Command 0x96. HD_GET_ALERT_MSG

HD_GET_ALERT_MSG retrieves alert message. Alerts are special messages provided by the broadcaster that may signal important information about emergencies or events. Full details about Alerts can be found in the 2206 standard. This API is used to collect the alert data and is used in response to an alert event. See the HD_GET_EVENT_STATUS command for details on the alert event. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x96 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | LENGTH[15:8] | | | | | | | |
| RESP6 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[15:0] | The number of data bytes returned in the alert payload. The length of the alert message will not exceed 395 bytes. |
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 Alert documentation for details about then returned data. |

### Command 0x97. HD_PLAY_ALERT_TONE

HD_PLAY_ALERT_TONE plays the alert tone. Alerts are special messages provided by the broadcaster that may signal important information about emergencies or events. Full details about Alerts can be found in the 2206 standard. This API is used to play an alert tone at the host's discretion. It is recommended that the host play this tone for each unique alert message it receives. If the host chooses it can also have these tones played automatically on every alert message. See the HD_EVENT_ALERT_CONFIG property for details on playing alert tones automatically. Also see the HD_GET_EVENT_STATUS command for details on the alert event. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x97 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x98. HD_TEST_GET_BER_INFO

HD_TEST_GET_BER_INFO reads the current BER information for the HD digital demod. The information returned by this command is only meaningful if the BER test vector (IB_FMr208c_e1wfc204) is being received. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x98 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | PIDS_BLOCK_ERRORS[7:0] | | | | | | | |
| RESP5 | PIDS_BLOCK_ERRORS[15:8] | | | | | | | |
| RESP6 | PIDS_BLOCK_ERRORS[23:16] | | | | | | | |
| RESP7 | PIDS_BLOCK_ERRORS[31:24] | | | | | | | |
| RESP8 | PIDS_BLOCKS_TESTED[7:0] | | | | | | | |
| RESP9 | PIDS_BLOCKS_TESTED[15:8] | | | | | | | |
| RESP10 | PIDS_BLOCKS_TESTED[23:16] | | | | | | | |
| RESP11 | PIDS_BLOCKS_TESTED[31:24] | | | | | | | |
| RESP12 | PIDS_BIT_ERRORS[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP13 | PIDS_BIT_ERRORS[15:8] | | | | | | | |
| RESP14 | PIDS_BIT_ERRORS[23:16] | | | | | | | |
| RESP15 | PIDS_BIT_ERRORS[31:24] | | | | | | | |
| RESP16 | PIDS_BITS_TESTED[7:0] | | | | | | | |
| RESP17 | PIDS_BITS_TESTED[15:8] | | | | | | | |
| RESP18 | PIDS_BITS_TESTED[23:16] | | | | | | | |
| RESP19 | PIDS_BITS_TESTED[31:24] | | | | | | | |
| RESP20 | P3_BIT_ERRORS[7:0] | | | | | | | |
| RESP21 | P3_BIT_ERRORS[15:8] | | | | | | | |
| RESP22 | P3_BIT_ERRORS[23:16] | | | | | | | |
| RESP23 | P3_BIT_ERRORS[31:24] | | | | | | | |
| RESP24 | P3_BITS_TESTED[7:0] | | | | | | | |
| RESP25 | P3_BITS_TESTED[15:8] | | | | | | | |
| RESP26 | P3_BITS_TESTED[23:16] | | | | | | | |
| RESP27 | P3_BITS_TESTED[31:24] | | | | | | | |
| RESP28 | P2_BIT_ERRORS[7:0] | | | | | | | |
| RESP29 | P2_BIT_ERRORS[15:8] | | | | | | | |
| RESP30 | P2_BIT_ERRORS[23:16] | | | | | | | |
| RESP31 | P2_BIT_ERRORS[31:24] | | | | | | | |
| RESP32 | P2_BITS_TESTED[7:0] | | | | | | | |
| RESP33 | P2_BITS_TESTED[15:8] | | | | | | | |
| RESP34 | P2_BITS_TESTED[23:16] | | | | | | | |
| RESP35 | P2_BITS_TESTED[31:24] | | | | | | | |
| RESP36 | P1_BIT_ERRORS[7:0] | | | | | | | |
| RESP37 | P1_BIT_ERRORS[15:8] | | | | | | | |
| RESP38 | P1_BIT_ERRORS[23:16] | | | | | | | |
| RESP39 | P1_BIT_ERRORS[31:24] | | | | | | | |
| RESP40 | P1_BITS_TESTED[7:0] | | | | | | | |
| RESP41 | P1_BITS_TESTED[15:8] | | | | | | | |
| RESP42 | P1_BITS_TESTED[23:16] | | | | | | | |
| RESP43 | P1_BITS_TESTED[31:24] | | | | | | | |

| Name | Function |
|---|---|
| PIDS_BLOCK_ER-RORS[31:0] | The number of PIDS blocks received with errors. |
| PIDS_BLOCKS_-TESTED[31:0] | The total number of PIDS blocks received. |
| PIDS_BIT_ER-RORS[31:0] | The number of PIDS bits received with errors. |
| PIDS_BITS_-TESTED[31:0] | The total number of PIDS bits received. |
| P3_BIT_ERRORS[31:0] | The number of P3 bits received with errors. |
| P3_BITS_TESTED[31:0] | The total number of P3 bits received. |
| P2_BIT_ERRORS[31:0] | The number of P2 bits received with errors. |
| P2_BITS_TESTED[31:0] | The total number of P2 bits received. |
| P1_BIT_ERRORS[31:0] | The number of P1 bits received with errors. |
| P1_BITS_TESTED[31:0] | The total number of P1 bits received. |

### Command 0x99. HD_SET_ENABLED_PORTS

HD_SET_ENABLED_PORTS sets the default HD ports retrieved/enabled when HD has been acquired. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x99 | | | | | | | |
| ARG1 | LENGTH[7:0] | | | | | | | |
| ARG2 | DATA_0[7:0] | | | | | | | |
| ARG3 | DATA_0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[7:0] | The numbers of ports included in this command. Maximum 64. |
| DATA_0[15:0] | The first port address of LENGTH addresses. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x9A. HD_GET_ENABLED_PORTS

HD_GET_ENABLED_PORTS gets the default HD ports retrieved when HD has been acquired. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x9A | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |
| RESP6 | DATA_0[7:0] | | | | | | | |
| RESP7 | DATA_0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[7:0] | The numbers of ports previously set. Maximum 64. |
| DATA_0[15:0] | If set, the first port byte. |

### Command 0xE5. TEST_GET_RSSI

TEST_GET_RSSI returns the reported RSSI in 8.8 format. This command is used to help calibrate the frontend tracking circuit. It returns the RSSI value in dBµV to 1/256 of a dB. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: TEST_GET_RSSI command is not supported in firmware revision A0A or A0B.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xE5 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | RSSI_HIGHRES[7:0] | | | | | | | |
| RESP5 | RSSI_HIGHRES[15:8] | | | | | | | |

| Name | Function |
|---|---|
| RSSI_HIGHRES[15:0] | The RSSI measurement in dBµV to 1/256 of a dB. |

## 4.2. Properties Summary—FMHD

**Table 4. Properties Summary—FMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x0000 | INT_CTL_ENABLE | Interrupt enable property | 0x0000 |
| 0x0001 | INT_CTL_REPEAT | Interrupt repeat property | 0x0000 |
| 0x0200 | DIGITAL_IO_OUTPUT_SELECT | Selects digital audio Master or Slave. | 0 |
| 0x0201 | DIGITAL_IO_OUTPUT_SAMPLE_RATE | Sets output sample audio rate in units of 1Hz. | 48000 |
| 0x0202 | DIGITAL_IO_OUTPUT_FORMAT | Configure digital output format. | 0x1800 |
| 0x0203 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 | Deviations from the standard framing mode | 0 |
| 0x0204 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 | Deviations from the standard framing mode | 0 |
| 0x0205 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 | Deviations from the standard framing mode | 0 |
| 0x0206 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 | Deviations from the standard framing mode | 0 |
| 0x0300 | AUDIO_ANALOG_VOLUME | Sets the audio analog volume. | 63 |
| 0x0301 | AUDIO_MUTE | AUDIO_MUTE property mutes/unmutes each audio output independently. | 0x0000 |
| 0x0800 | PIN_CONFIG_ENABLE | Pin configuration property | 0x8001 |
| 0x1710 | FM_TUNE_FE_VARM | FM Front End Varactor configuration slope | 0 |
| 0x1711 | FM_TUNE_FE_VARB | FM Front End Varactor configuration intercept | 0 |
| 0x1712 | FM_TUNE_FE_CFG | Additional configuration options for the front end. | 0x0000 |
| 0x3100 | FM_SEEK_BAND_BOTTOM | Sets the lower seek boundary of the FM band in multiples of 10kHz. | 8750 |
| 0x3101 | FM_SEEK_BAND_TOP | Sets the upper seek boundary for the FM band in multiples of 10kHz. | 10790 |
| 0x3102 | FM_SEEK_FREQUENCY_SPACING | Sets the frequency spacing for the FM band in multiples of 10kHz when performing a seek. | 10 |
| 0x3200 | FM_VALID_MAX_TUNE_ERROR | Sets the maximum frequency error allowed before setting the AFCRL indicator. | 114 |
| 0x3201 | FM_VALID_RSSI_TIME | Sets the amount of time in ms to allow the RSSI/ISSI metrics to settle before evaluating. | 15 |

**Table 4. Properties Summary—FMHD (Continued)**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x3202 | FM_VALID_RSSI_THRESHOLD | Sets the RSSI threshold for a valid FM Seek/Tune. | 17 |
| 0x3203 | FM_VALID_SNR_TIME | Sets the amount of time in ms to allow the SNR metric to settle before evaluating. | 40 |
| 0x3204 | FM_VALID_SNR_THRESHOLD | Sets the SNR threshold for a valid FM Seek/Tune. | 10 |
| 0x3300 | FM_RSQ_INTERRUPT_SOURCE | Configures interrupt related to Received Signal Quality metrics. | 0 |
| 0x3301 | FM_RSQ_SNR_HIGH_THRESHOLD | FM_RSQ_SNR_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the SNR is above this threshold. | 127 |
| 0x3302 | FM_RSQ_SNR_LOW_THRESHOLD | FM_RSQ_SNR_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the SNR is below this threshold. | -128 |
| 0x3303 | FM_RSQ_RSSI_HIGH_THRESHOLD | FM_RSQ_RSSI_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the RSSI is above this threshold. | 127 |
| 0x3304 | FM_RSQ_RSSI_LOW_THRESHOLD | Sets low threshold which triggers the RSQ interrupt if the RSSI is below this threshold. | -128 |
| 0x3307 | FM_RSQ_HD_DETECTION | Configures the Fast HD Detection routine. | 0x800d |
| 0x3400 | FM_ACF_INTERRUPT_SOURCE | Enables the ACF interrupt sources. | 0 |
| 0x3401 | FM_ACF_SOFTMUTE_THRESHOLD | Sets the softmute interrupt threshold in dB attenuation. | 31 |
| 0x3402 | FM_ACF_HIGHCUT_THRESHOLD | Sets the high cut interrupt threshold. | 0 |
| 0x3403 | FM_ACF_BLEND_THRESHOLD | Sets the stereo blend interrupt threshold. | 0 |
| 0x3404 | FM_ACF_SOFTMUTE_TOLERANCE | Sets the distance from the final softmute value that triggers softmute convergence flag. | 2 |
| 0x3405 | FM_ACF_HIGHCUT_TOLERANCE | Sets the distance from the final high cut frequency that triggers the high cut convergence flag. | 20 |
| 0x3406 | FM_ACF_BLEND_TOLERANCE | Sets the distance from the final blend state that triggers the blend convergence flag. | 5 |
| 0x3500 | FM_SOFTMUTE_SNR_LIMITS | Sets the SNR limits for soft mute attenuation. | 0x0602 |

**Table 4. Properties Summary—FMHD (Continued)**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x3501 | FM_SOFTMUTE_SNR_ATTENUATION | Sets the SNR based softmute attenuation limits. | 0x0008 |
| 0x3502 | FM_SOFTMUTE_SNR_ATTACK_TIME | Sets the soft mute attack time. | 16 |
| 0x3503 | FM_SOFTMUTE_SNR_RELEASE_TIME | Sets the soft mute release time. | 4000 |
| 0x3600 | FM_HIGHCUT_RSSI_LIMITS | RSSI based high cut limits. | 0x0C06 |
| 0x3601 | FM_HIGHCUT_RSSI_CUTOFF_FREQ | RSSI based high cut cutoff frequencies. | 0xC828 |
| 0x3602 | FM_HIGHCUT_RSSI_ATTACK_TIME | RSSI based high cut attack time. | 16 |
| 0x3603 | FM_HIGHCUT_RSSI_RELEASE_TIME | RSSI based high cut release time. | 4000 |
| 0x3604 | FM_HIGHCUT_SNR_LIMITS | SNR based high cut limits. | 0x0903 |
| 0x3605 | FM_HIGHCUT_SNR_CUTOFF_FREQ | SNR based high cut cutoff frequencies. | 0xc828 |
| 0x3606 | FM_HIGHCUT_SNR_ATTACK_TIME | SNR based high cut attack time. | 16 |
| 0x3607 | FM_HIGHCUT_SNR_RELEASE_TIME | SNR based high cut release time. | 4000 |
| 0x3608 | FM_HIGHCUT_MULTIPATH_LIMITS | Multipath based high cut limits. | 0x2D3C |
| 0x3609 | FM_HIGHCUT_MULTIPATH_CUTOFF_FREQ | Multipath based high cut cutoff frequencies. | 0xc828 |
| 0x360A | FM_HIGHCUT_MULTIPATH_ATTACK_TIME | Multipath based high cut attack time. | 16 |
| 0x360B | FM_HIGHCUT_MULTIPATH_RELEASE_TIME | Multipath based high cut release time. | 4000 |
| 0x3700 | FM_BLEND_RSSI_LIMITS | This property sets the RSSI limits for RSSI controlled stereo separation. | 0x2010 |
| 0x3702 | FM_BLEND_RSSI_ATTACK_TIME | RSSI based stereo to mono attack time in ms. | 16 |
| 0x3703 | FM_BLEND_RSSI_RELEASE_TIME | RSSI based mono to stereo release time in ms. | 4000 |
| 0x3704 | FM_BLEND_SNR_LIMITS | Sets the SNR limits for SNR controlled stereo separation. | 0x180F |
| 0x3706 | FM_BLEND_SNR_ATTACK_TIME | SNR based stereo to mono attack time in ms. | 16 |
| 0x3707 | FM_BLEND_SNR_RELEASE_TIME | SNR based mono to stereo release time in ms. | 4000 |
| 0x3708 | FM_BLEND_MULTIPATH_LIMITS | Sets the multipath limits for multipath controlled stereo separation. | 0x2D3C |
| 0x370A | FM_BLEND_MULTIPATH_ATTACK_TIME | Multi-path based stereo to mono attack time in ms. | 16 |
| 0x370B | FM_BLEND_MULTIPATH_RELEASE_TIME | Multi-path based mono to stereo release time in ms. | 4000 |
| 0x3900 | FM_AUDIO_DE_EMPHASIS | Sets the FM Receive de-emphasis. | 0 |
| 0x3C00 | FM_RDS_INTERRUPT_SOURCE | This property configures interrupt related to RDS. | 0x0000 |

**Table 4. Properties Summary—FMHD (Continued)**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x3C01 | FM_RDS_INTERRUPT_FIFO_COUNT | Configures minimum received data groups in fifo before interrupt. | 0x0000 |
| 0x3C02 | FM_RDS_CONFIG | Enables RDS and configures acceptable block error threshold. | 0x0000 |
| 0x3C03 | FM_RDS_CONFIDENCE | Configures rds block confidence threshold. | 0x1111 |
| 0x8100 | DIGITAL_SERVICE_INT_SOURCE | Configures the interrupt sources for digital services | 0x0000 |
| 0x9101 | HD_BLEND_OPTIONS | This property provides options to control HD/analog audio blend behavior. | 0x0002 |
| 0x9102 | HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME | This property sets the amount of time it takes in ms to blend from analog to HD. This property only applies to primary service channel. | 750 |
| 0x9103 | HD_BLEND_HD_TO_ANALOG_TRANSITION_TIME | This property sets the amount of time it takes in ms to blend from HD to analog. This property only applies to primary service channel. | 100 |
| 0x9106 | HD_BLEND_DYNAMIC_GAIN | This property sets the digital audio dynamic linear scaling factor. | 0x7F |
| 0x9200 | HD_DIGRAD_INTERRUPT_SOURCE | Configures interrupts related to digital receivers HD_DIGRAD_STATUS command. | 0 |
| 0x9201 | HD_DIGRAD_CDNR_LOW_THRESHOLD | Configures CDNR LOW interrupt interrupt threshold. | 0 |
| 0x9202 | HD_DIGRAD_CDNR_HIGH_THRESHOLD | Configures CDNR HIGH interrupt interrupt threshold. | 127 |
| 0x9203 | HD_DIGRAD_AUTO_ACQUIRE | This property sets the radio to automatically acquire digital radio. | 1 |
| 0x9300 | HD_EVENT_INTERRUPT_SOURCE | Configures interrupts related to digital receiver (HD_GET_EVENT_STATUS). | 0 |
| 0x9301 | HD_EVENT_SIS_CONFIG | This property configures which basic SIS information is returned by the HD_GET_STATION_INFO command BASICSIS option. | 0x0017 |
| 0x9302 | HD_EVENT_ALERT_CONFIG | This property configures HD alerts. Alert information is returned by the HD_GET_ALERT_MSG command. | 0x0001 |
| 0x9500 | HD_PSD_ENABLE | Sets which audio services will provide program service data. | 0 |
| 0x9501 | HD_PSD_FIELD_MASK | Sets which PSD fields will be decoded. | 0xFFFF |

**Table 4. Properties Summary—FMHD (Continued)**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x9700 | HD_AUDIO_CTRL_FRAME_DELAY | Controls the value of the delay of decoded digital audio samples relative to the output of the audio quality indicator. | 6 |
| 0x9701 | HD_AUDIO_CTRL_PROGRAM_LOSS_THRESHOLD | Controls the duration before reverting to MPS audio after an SPS audio program is removed or lost. | 0 |
| 0x9702 | HD_AUDIO_CTRL_BALL_GAME_ENABLE | Selects the audio output for hybrid waveforms when the TX Blend Control Status (BCTL) bits are set to 01. | 1 |
| 0x9901 | HD_CODEC_MODE_0_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 0 | 3700 |
| 0x9904 | HD_CODEC_MODE_2_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 2 | 0 |
| 0x9907 | HD_CODEC_MODE_10_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 10 | 0 |
| 0x990A | HD_CODEC_MODE_13_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 13 | 0 |
| 0xE800 | HD_TEST_BER_CONFIG | Enables the HD BER test. | 0 |
| 0xE801 | HD_TEST_DEBUG_AUDIO | Used to put the analog audio output in to special test modes for debug purposes. | 0 |

### 4.2.1. FMHD Properties

**Property 0x0000. INT_CTL_ENABLE**

INT_CTL_ENABLE property enables top-level interrupt sources. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTIEN | Reserved | | | | | CTSIEN | ERR_CMDIEN | DACQIEN | DSRVIEN | RSQIEN | RDSIEN | ACFIEN | STCIEN |
| Default | 0x0 | | 0 | 0x00 | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTIEN | Interrupt when DEVNTINT is set |
| 12:8 | Reserved | Always write to 0x00. |
| 7 | CTSIEN | Interrupt when CTS is set.<br>Note: The default will be 0x1 if POWER_UP:CTSIEN was set. |
| 6 | ERR_CMDIEN | Interrupt when ERR_CMD is set |
| 5 | DACQIEN | Interrupt when DIGINT is set |
| 4 | DSRVIEN | Interrupt when DSRVINT is set |
| 3 | RSQIEN | Interrupt when RSQINT is set |
| 2 | RDSIEN | Interrupt when RDSINT is set |
| 1 | ACFIEN | Interrupt when ACFINT is set |
| 0 | STCIEN | Interrupt when STCINT is set |

**Property 0x0001. INT_CTL_REPEAT**

INT_CTL_REPEAT is used to set repeat interrupt pulses for a given interrupt even if this particular interrupt was previously generated but not acknowledged. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTREP | Reserved | | | | | | | DACQREP | DSRVREP | RSQREP | RDSREP | ACFREP | STCREP |
| Default | 0x0 | | 0 | 0x00 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTREP | Repeat interrupt pulse when DEVNTINT is set, even if a previous interrupt was generated but not acknowledged. |
| 12:6 | Reserved | Always write to 0x00. |
| 5 | DACQREP | Repeat interrupt pulse when DACQINT is set, even if a previous interrupt was generated but not acknowledged. |
| 4 | DSRVREP | Repeat interrupt pulse when DSRVINT is set, even if a previous interrupt was generated but not acknowledged. |
| 3 | RSQREP | Repeat interrupt pulse when DIGINT is set, even if a previous interrupt was generated but not acknowledged. |
| 2 | RDSREP | Repeat interrupt pulse when RDSINT is set, even if a previous interrupt was generated but not acknowledged. |
| 1 | ACFREP | Repeat interrupt pulse when AFCINT is set, even if a previous interrupt was generated but not acknowledged. |
| 0 | STCREP | Repeat interrupt pulse when STCINT is set, even if a previous interrupt was generated but not acknowledged. |

### Property 0x0200. DIGITAL_IO_OUTPUT_SELECT

DIGITAL_IO_OUTPUT_SELECT configures the digital audio output to be I2S Master or Slave. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | MASTER | | | | | | | | | | | | | | | |
| **Default** | 0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | MASTER | 0: Slave mode selected<br>1: Master mode selected |

### Property 0x0201. DIGITAL_IO_OUTPUT_SAMPLE_RATE

DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 48000

Units: Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | OUTPUT_SAMPLE_RATE[15:0] | | | | | | | | | | | | | | | |
| Default | 0xBB80 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | OUTPUT_SAMPLE_RATE[15:0] | DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz.<br>Range: 32000-48000<br>32000: The minimum output sample rate in Hz.<br>48000: The minimum output sample rate in Hz. |

**Property 0x0202. DIGITAL_IO_OUTPUT_FORMAT**

DIGITAL_IO_OUTPUT_FORMAT configures the digital audio output format. This property may only be written before the first tune. Writes after first tune will return error. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x1800

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | SAMPL_SIZE[5:0] | | | | | | SLOT_SIZE[3:0] | | | | FRAMING_FORMAT[3:0] | | | |
| Default | 0x0 | | 0x18 | | | | | | 0x0 | | | | 0x0 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:8 | SAMPL_SIZE[5:0] | Determines the number of bits in a sample. Only the specified number of bits per sample are used. The value of any bits sent over the sample size will be determined by the FILL field of the DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 property. Sample sizes of less then 8 are not supported.<br>Range: 8-24 |
| 7:4 | SLOT_SIZE[3:0] | Determines slot size; must be larger than or equal to SAMPL_SIZE.<br>0x0: Matches Sample Size<br>0x2: 8 bits<br>0x4: 16 bits<br>0x5: 20 bits<br>0x6: 24 bits<br>0x7: 32 bits |
| 3:0 | FRAMING_FORMAT[3:0] | Determines when the data is transmitted relative to frame sync.<br>0x0: I2S mode<br>0x6: DSP mode<br>0x7: Left justified dsp mode<br>0x8: Left justified mode<br>0x9: Right justified mode |

### Property 0x0203. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | FSLATE_EN | FSINV_EN | RJUST_EN | CLK-INV_EN | SWAP_EN | BIT ORDER_EN | Reserved | | FSLATE | FSINV | RJUST | CLK-INV | SWAP | BIT ORDER |
| Default | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | FSLATE_EN | Enable the FSLATE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 12 | FSINV_EN | Enable the FSINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 11 | RJUST_EN | Enable the RJUST override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 10 | CLKINV_EN | Enable the CLKINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 9 | SWAP_EN | Enable the SWAP override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | BITORDER_EN | Enable the BITORDER override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | Reserved | Always write to 0x0. |
| 5 | FSLATE | 0: Sample data on the rising edge of the 2nd clock after DFS<br>1: Sample data on the rising edge of the 1st clock after DFS |
| 4 | FSINV | 0: Use the DFS signal as is<br>1: Invert the DFS signal |
| 3 | RJUST | 0: Left Justified<br>1: Right Justified |
| 2 | CLKINV | Inverts the data clock<br>0: The bit clock is not inverted. DFS will be captured on rising edge of DCLK.<br>1: The bit clock is inverted. DFS will be captured on falling edge of DCLK. |
| 1 | SWAP | Swap position of the left and right channels.<br>0: Transmit the left sample first<br>1: Transmit the right sample first |
| 0 | BITORDER | Determine if the MSB or LSB is transmitted first.<br>0: Transmit MSB first<br>1: Transmit LSB first |

**Property 0x0204. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2**

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | FILL_EN | Reserved | | | | SEQEN_EN | FSEDGE_EN | FILL[1:0] | | Reserved | | | | SEQEN | FSEDGE |
| Default | 0 | 0 | 0x0 | | | | 0 | 0 | 0x0 | | 0x0 | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15 | Reserved | Always write to 0. |
| 14 | FILL_EN | Enable the FILL override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 13:10 | Reserved | Always write to 0x0. |
| 9 | SEQEN_EN | Enable the SEQEN override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | FSEDGE_EN | Enable the FSEDGE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | FILL[1:0] | Defines the fill value for unused data bits.<br>0: The unused bits are filled with 0's<br>1: The unused bits are filled with 1's<br>2: The unused bits are sign extended<br>3: The unused bits are filled with a random sequence |
| 5:2 | Reserved | Always write to 0x0. |
| 1 | SEQEN | 0: Only one mono sample is sent per trigger of DFS<br>1: Both mono samples are sent sequentially after each trigger from DFS |
| 0 | FSEDGE | 0: One mono sample is sent on only one edge of DFS<br>1: One mono sample is sent every edge (rising and falling) of DFS |

### Property 0x0205. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSH_EN | FSH[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSH_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSH[14:0] | Number of samples FS will remain high.<br>Range: 0-32767 |

### Property 0x0206. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSL_EN | FSL[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSL_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSL[14:0] | Number of samples FS will remain low.<br>Range: 0-32767 |

### Property 0x0300. AUDIO_ANALOG_VOLUME

AUDIO_ANALOG_VOLUME sets the analog audio volume. A value of 0 will mute the audio; a value of 1 applies 62 dB of attenuation, and a value of 63 applies no attenuation. Each step accounts for 1 dB of change in the output. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 63

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | VOL[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x3F | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | VOL[5:0] | Controls the analog volume level.<br>Range: 0-63<br>0: The minimum audio level.<br>63: The maximum audio level. |

### Property 0x0301. AUDIO_MUTE

AUDIO_MUTE property mutes/unmutes each audio output independently. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | MUTE[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x0 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | MUTE[1:0] | Controls the audio mute on each audio output<br>0: Do not mute audio outputs<br>1: Mute Left Audio Out.<br>2: Mute Right Audio Out.<br>3: Mute both Left and Right Audio Out |

### Property 0x0800. PIN_CONFIG_ENABLE

PIN_CONFIG is used to enable and disable the various I/O features of the device. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x8001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | INT-BOUTEN | UARTEN | Reserved | | | | | | | | | | | | I2SOUTEN | DACOUTEN |
| Default | 1 | 0 | 0x000 | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15 | INTBOUTEN | When set enables the INTB output. |
| 14 | UARTEN | When set enables the UART I/O. |
| 13:2 | Reserved | Always write to 0x000. |
| 1 | I2SOUTEN | When set enables the I2S the digital audio output. A change to this bit will take effect on the next tune or seek. |
| 0 | DACOUTEN | When set enables the analog audio output. A change to this bit will take effect on the next tune or seek. |

### Property 0x1710. FM_TUNE_FE_VARM

FM_TUNE_FE_VARM FM Front End Varactor configuration slope (x 1000) which has been calculated for a particular board design. Both FM_TUNE_FE_VARB and FM_TUNE_FE_VARM must be configured. This takes effect upon FM_TUNE_FREQ. This is a signed value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FE_VARM[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FE_VARM[15:0] | FM Front End Varactor configuration slope (x 1000) which has been calculated for a particular board design. Both FM_TUNE_FE_VARB and FM_TUNE_FE_VARM must be configured. This takes effect upon FM_TUNE_FREQ. This is a signed value. |

### Property 0x1711. FM_TUNE_FE_VARB

FM_TUNE_FE_VARB FM Front End Varactor configuration intercept which has been calculated for a particular board design. Both FM_TUNE_FE_VARB and FM_TUNE_FE_VARM must be configured. This takes effect upon FM_TUNE_FREQ. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FE_VARB[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FE_VARB[15:0] | FM Front End Varactor configuration intercept which has been calculated for a particular board design. Both FM_TUNE_FE_VARB and FM_TUNE_FE_VARM must be configured. This takes effect upon FM_TUNE_FREQ. |

### Property 0x1712. FM_TUNE_FE_CFG

FM_TUNE_FE_CFG Additional configuration options for the front end. These take effect upon FM_TUNE_FREQ. See FM_TUNE_FE_VARB and FM_TUNE_FE_VARM The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | VHF SW |
| Default | 0x0000 | | | | | | | | | | | | | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | VHFSW | VHFSW sets the open or closed state for the front end switch.<br>0: Switch Open<br>1: Switch Closed |

### Property 0x3100. FM_SEEK_BAND_BOTTOM

FM_SEEK_BAND_BOTTOM sets the lower seek boundary of the FM band in multiples of 10kHz. See FM_SEEK_START. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 8750

Units: 10kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FMSKFREQL[15:0] | | | | | | | | | | | | | | | |
| Default | 0x222E | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FMSKFREQL[15:0] | FM Seek Band Bottom.<br>Range: 7600-10800<br>7600: Japan Range Minimum<br>7800: Wide Range Minimum<br>8750: Standard Range Minimum<br>9000: Japan Range Maximum<br>10790: Standard Range Maximum<br>10800: Wide Range Maximum |

### Property 0x3101. FM_SEEK_BAND_TOP

FM_SEEK_BAND_TOP sets the upper seek boundary of the FM band in multiples of 10kHz. See FM_SEEK_START. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10790

Units: 10kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FMSKFREQH[15:0] | | | | | | | | | | | | | | | |
| Default | 0x2A26 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FMSKFREQH[15:0] | FM Seek Band Top.<br>Range: 7600-10800<br>7600: Japan Range Minimum<br>7800: Wide Range Minimum<br>8750: Standard Range Minimum<br>9000: Japan Range Maximum<br>10790: Standard Range Maximum<br>10800: Wide Range Maximum |

### Property 0x3102. FM_SEEK_FREQUENCY_SPACING

FM_SEEK_FREQUENCY_SPACING sets the frequency spacing for the FM band in multiples of 10kHz when performing a seek. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: 10kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | FMSKSPACE[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x0A | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | FMSKSPACE[4:0] | FM Seek Frequency Spacing.<br>5: 50 kHz<br>10: 100 kHz<br>20: 200 kHz |

### Property 0x3200. FM_VALID_MAX_TUNE_ERROR

FM_VALID_MAX_TUNE_ERROR sets the maximum freq error allowed in units of bppm before setting the AFC rail indicator (AFCRL). This will take effect on the next tune. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 114

Units: bppm

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | FMMAXTUNEERR[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x72 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | FMMAX-TUNEERR[7:0] | Frequency error in bppm (2 ppm).<br>Range: 0-126 |

### Property 0x3201. FM_VALID_RSSI_TIME

FM_VALID_RSSI_TIME sets the amount of time in ms to allow the RSSI/ISSI metrics to settle before evaluating. This property is ignored if the FASTTUNE bit(FM_TUNE_FREQ) is set. The reliability of the valid bit for identifying valid stations relies on this parameter being set properly. NOTE: Setting this property to a value lower then 4 will result in the metric not being measured correctly at tune time. To disable metric as a tune qualifier set the validation time to 0 and set the validation threshold to the minimum value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 15

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | SSIVALTIME[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x0F | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | SSIVALTIME[5:0] | Validation time in ms<br>Range: 0-63—Specified in units ms. |

### Property 0x3202. FM_VALID_RSSI_THRESHOLD

Sets the RSSI threshold for a valid FM Seek/Tune. If the desired channel RSSI is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 17

Units: dBµV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | FMVALRSSI[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x11 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | FMVALRSSI[7:0] | FM Seek/Tune Received Signal Strength Threshold.<br>Range: -128-127—Specified in units of dBµV in 1 dBµV steps.<br>-128: RSSI is not used as a criterion in determining the validity of a station. |

### Property 0x3203. FM_VALID_SNR_TIME

FM_VALID_SNR_TIME sets the amount of time in ms to allow the SNR metric to settle before evaluating. This property is ignored if the FASTTUNE bit(FM_TUNE_FREQ) is set. The reliability of the valid bit for identifying valid stations relies on this parameter being set properly. NOTE: Setting this property to a value lower then 4 will result in the metric not being measured correctly at tune time. To disable metric as a tune qualifier set the validation time to 0 and set the validation threshold to the minimum value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 40

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | SNRVALTIME[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x28 | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | SNRVALTIME[5:0] | Validation time in ms<br>Range: 0-63—Specified in units ms. |

### Property 0x3204. FM_VALID_SNR_THRESHOLD

FM_VALID_SNR_THRESHOLD sets the SNR threshold for a valid FM Seek/Tune. If the desired channel SNR is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | FMVALSNR[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x0A | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | FMVALSNR[7:0] | FM Seek/Tune SNR Threshold.<br>Range: -128-127—Specified in units of dB in 1 dB steps.<br>-128: SNR is not used as a criterion in determining the validity of a station. |

### Property 0x3300. FM_RSQ_INTERRUPT_SOURCE

FM_RSQ_INTERRUPT_SOURCE configures interrupt related to Received Signal Quality metrics. See FM_RSQ_STATUS. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | MULTHINT | MULTLINT | Reserved | | SNRHINT | SNRLINT | RSSIHINT | RSSILINT |
| Default | 0x00 | | | | | | | | 0 | 0 | 0x0 | | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | MULTHINT | Enables the interrupt for multipath High.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if Multipath level goes above FM_RSQ_MULTIPATH_HIGH_THRESHOLD. |
| 6 | MULTLINT | Enables the interrupt for multipath Low.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if Multipath level goes below FM_RSQ_MULTIPATH_LOW_THRESHOLD. |
| 5:4 | Reserved | Always write to 0x0. |
| 3 | SNRHINT | Enables the interrupt for SNR High.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if SNR goes above FM_RSQ_SNR_HIGH_THRESHOLD. |
| 2 | SNRLINT | Enables the interrupt for SNR Low.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if SNR goes below FM_RSQ_SNR_LOW_THRESHOLD. |
| 1 | RSSIHINT | Enables the interrupt for RSSI High.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if RSSI goes above FM_RSQ_RSSI_HIGH_THRESHOLD. |
| 0 | RSSILINT | Enables the interrupt for RSSI Low.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if RSSI goes below FM_RSQ_RSSI_LO_THRESHOLD. |

**Property 0x3301. FM_RSQ_SNR_HIGH_THRESHOLD**

FM_RSQ_SNR_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the SNR is above this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SNRH[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | SNRH[7:0] | FM RSQ SNR High Threshold.<br>Range: -128-127—Specified in units of dB in 1 dB steps. |

### Property 0x3302. FM_RSQ_SNR_LOW_THRESHOLD

FM_RSQ_SNR_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the SNR is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -128

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SNRL[7:0] | | | | | | | |
| Default | 0xFF | | | | | | | | 0x80 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0xFF. |
| 7:0 | SNRL[7:0] | Low threshold for SNR interrupt in dB.<br>Range: -128-127—Specified in units of dB in 1 dB steps. |

### Property 0x3303. FM_RSQ_RSSI_HIGH_THRESHOLD

FM_RSQ_RSSI_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the RSSI is above this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dBµV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | RSSIH[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | RSSIH[7:0] | High threshold for RSSI interrupt in dBµV.<br>Range: -128-127—Specified in units of dBµV in 1 dBµV steps. |

### Property 0x3304. FM_RSQ_RSSI_LOW_THRESHOLD

FM_RSQ_RSSI_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the RSSI is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -128

Units: dBµV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | RSSIL[7:0] | | | | | | | |
| Default | 0xFF | | | | | | | | 0x80 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0xFF. |
| 7:0 | RSSIL[7:0] | Low threshold for RSSI interrupt in dBµV.<br>Range: -128-127—Specified in units of dBµV in 1 dBµV steps. |

### Property 0x3307. FM_RSQ_HD_DETECTION

Default: 0x800d

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SAMPLES[7:0] | | | | | | | |
| Default | 0x80 | | | | | | | | 0x0D | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x80. |
| 7:0 | SAMPLES[7:0] | Sets the number of HD OFDM symbols to look at during HD Detection. Each HD OFDM symbol is ~2.9ms so the detection time is about 2.9 times this number. Enabling this feature will extend the tune time by this amount minus the SNR and RSSI validation times. Setting this field to 0 disables Fast HD Detect. If HD is enabled, the detection period will be larger of the RSSI or SNR validation times or SAMPLES*3ms.<br>Range: 0-100 |

### Property 0x3400. FM_ACF_INTERRUPT_SOURCE

FM_ACF_INTERRUPT_SOURCE Enables the ACF interrupt sources. When one of the interrupts is enabled, the ACFINT bit of the status word will be set when the controlling indicator crosses the threshold set its ACF threshold property. If the interrupt is acknowledged and the condition persists, the interrupt will immediately repeat. If the ACFINT hardware interrupt is enabled by setting the ACFIEN bit of the INT_CTL_ENABLE, the interrupt line will toggle. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | BLEND_INTEN | HIGHCUT_INTEN | SMUTE_INTEN |
| Default | 0x0000 | | | | | | | | | | | | | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2 | BLEND_INTEN | Enables the blend interrupt.<br>0: The blend interrupt is disabled<br>1: The blend interrupt is enabled |
| 1 | HIGHCUT_INTEN | Enables the Hicut Interrupt<br>0: The Hicut interrupt is disabled<br>1: The Hicut interrupt is enabled |
| 0 | SMUTE_INTEN | Enables the softmute interrupt<br>0: The softmute interrupt is disabled<br>1: The softmute interrupt is enabled |

### Property 0x3401. FM_ACF_SOFTMUTE_THRESHOLD

FM_ACF_SOFTMUTE_THRESHOLD sets the softmute interrupt threshold. When softmute attenuation rises above the level set by this property the SMUTE_INT bit of the FM_ACF_STATUS command will be set. If the SMUTE_INTEN bit of the FM_ACF_INTERRUPT_SOURCE property is set the ACFINT interrupt will be asserted in the status word. See FM_ACF_INTERRUPT_SOURCE for details. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 31

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | SMATTN_THRESH[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x1F | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | SMATTN_-<br>THRESH[4:0] | Sets the softmute interrupt threshold in dB.<br>Range: 0-31 |

### Property 0x3402. FM_ACF_HIGHCUT_THRESHOLD

FM_ACF_HIGHCUT_THRESHOLD sets the high cut interrupt threshold. When the cutoff frequency falls below this threshold, the HIGHCUT_INT bit of FM_ACF_STATUS command will be asserted. If the high cut interrupt is enabled, the ACFINT bit in the status word will also be asserted. See FM_ACF_INTERRUPT_SOURCE for details. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | HIGHCUT_THRESH[7:0] | | | | | | | |
| **Default** | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HIGHCUT_-<br>THRESH[7:0] | High cut interrupt threshold in units of 100Hz<br>Range: 0-200 |

### Property 0x3403. FM_ACF_BLEND_THRESHOLD

FM_ACF_BLEND_THRESHOLD sets the Stereo Blend interrupt threshold. When the stereo separation falls below this threshold the BLEND_INT bit of the FM_ACF_STATUS command will be set. If the blend interrupt is enabled, the ACFINT bit of the status word will also be asserted. See the FM_ACF_INTERRUPT_SOURCE property for more details. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | | BLEND_THRESH[6:0] | | | | | | |
| **Default** | 0x000 | | | | | | | | | 0x00 | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:7 | Reserved | Always write to 0x000. |
| 6:0 | BLEND_-THRESH[6:0] | Stereo separation in units of dB.<br>Range: 0-100 |

### Property 0x3404. FM_ACF_SOFTMUTE_TOLERANCE

FM_ACF_SOFTMUTE_TOLERANCE sets the distance from the final softmute value that triggers the softmute convergence flag. Convergence is indicated by setting the SMUTE_CONV flag in the FM_ACF_STATUS command reply. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 2

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | SMUTE_TOL[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x02 | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | SMUTE_TOL[4:0] | When the softmute value is this many dB away from its final value the SMUTE_-CONV flag will be set.<br>Range: 0-31 |

### Property 0x3405. FM_ACF_HIGHCUT_TOLERANCE

FM_ACF_HIGHCUT_TOLERANCE Sets the distance from the final high cut freq that triggers the high cut convergence flag. Convergence is indicated by a setting HIGHCUT_CONV flag of FM_ACF_STATUS command reply. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 20

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HIGHCUT_TOL[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x14 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HIGH-CUT_TOL[7:0] | When the high cut state is N*100Hz away from its final value the HIGHCUT_CONV flag will be set.<br>Range: 0-200 |

### Property 0x3406. FM_ACF_BLEND_TOLERANCE

FM_ACF_BLEND_TOLERANCE sets the distance from the final blend state that triggers the blend convergence flag. Blend convergence is indicated by setting the BLEND_CONV flag of the FM_ACF_STATUS command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 5

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | BLEND_TOL[6:0] | | | | | | |
| Default | 0x000 | | | | | | | | | 0x05 | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:7 | Reserved | Always write to 0x000. |
| 6:0 | BLEND_TOL[6:0] | When the stereo separation is this many dB away from its final value the BLEND_-CONV flag will get set.<br>Range: 0-100 |

### Property 0x3500. FM_SOFTMUTE_SNR_LIMITS

FM_SOFTMUTE_SNR_LIMITS sets the SNR limits for soft mute attenuation. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0602

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x06 | | | | | | | | 0x02 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of SNR in dB which causes the minimum audio attenuation.<br>Range: -20-64 |
| 7:0 | XMIN[7:0] | Value of SNR in dB which causes the maximum audio attenuation.<br>Range: -20-64 |

### Property 0x3501. FM_SOFTMUTE_SNR_ATTENUATION

FM_SOFTMUTE_SNR_ATTENUATION sets the SNR attenuation limits. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0008

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTENMIN[7:0] | | | | | | | | ATTENMAX[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x08 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | ATTENMIN[7:0] | Sets the minimum attenuation amount in dB for SNR based softmute.<br>Range: 0-31 |
| 7:0 | ATTENMAX[7:0] | Sets the maximum attenuation amount in dB for SNR based softmute.<br>Range: 0-31 |

### Property 0x3502. FM_SOFTMUTE_SNR_ATTACK_TIME

FM_SOFTMUTE_SNR_ATTACK_TIME sets the attack time to mute the audio. The attack time is the time it takes the softmute attenuation to go from YMIM to YMAX if the SNR made a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | The softmute attack time in ms for SNR based softmute.<br>Range: 16-65535 |

**Property 0x3503. FM_SOFTMUTE_SNR_RELEASE_TIME**

FM_SOFTMUTE_SNR_RELEASE_TIME Sets the release time to unmute the audio. The release time is the time it takes the softmute attenuation to go from YMAX to YMIN if the SNR made a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | The softmute release time in ms for SNR based softmute.<br>Range: 16-65535 |

**Property 0x3600. FM_HIGHCUT_RSSI_LIMITS**

FM_HIGHCUT_RSSI_LIMITS sets the RSSI limits for RSSI based high cut. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0C06

Units: dBµV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x0C | | | | | | | | 0x06 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of RSSI which causes the maximum cutoff frequency.<br>Range: -20-120 |
| 7:0 | XMIN[7:0] | Value of RSSI which causes the minimum cutoff frequency.<br>Range: -20-120 |

### Property 0x3601. FM_HIGHCUT_RSSI_CUTOFF_FREQ

FM_HIGHCUT_RSSI_CUTOFF_FREQ sets the audio cutoff frequencies for RSSI based high cut. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0xC828

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | YMAX[7:0] | | | | | | | | YMIN[7:0] | | | | | | | |
| Default | 0xC8 | | | | | | | | 0x28 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | YMAX[7:0] | Sets the maximum cutoff frequency for RSSI based high cut.<br>Range: 0-200 |
| 7:0 | YMIN[7:0] | Sets the minimum cutoff frequency for RSSI based high cut.<br>Range: 0-200 |

### Property 0x3602. FM_HIGHCUT_RSSI_ATTACK_TIME

FM_HIGHCUT_RSSI_ATTACK_TIME sets the transition time for which RSSI based high cut lowers the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMAX to YMIN assuming RSSI makes a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 1-65535 |

### Property 0x3603. FM_HIGHCUT_RSSI_RELEASE_TIME

FM_HIGHCUT_RSSI_RELEASE_TIME sets the transition time for which RSSI based high cut increases the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMIN to YMAX assuming RSSI

makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 1-65535 |

### Property 0x3604. FM_HIGHCUT_SNR_LIMITS

FM_HIGHCUT_SNR_LIMITS sets the SNR limits for SNR based high cut. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0903

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x09 | | | | | | | | 0x03 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of SNR which causes the maximum cutoff frequency.<br>Range: -20-64 |
| 7:0 | XMIN[7:0] | Value of SNR which causes the minimum cutoff frequency.<br>Range: -20-64 |

### Property 0x3605. FM_HIGHCUT_SNR_CUTOFF_FREQ

FM_HIGHCUT_SNR_CUTOFF_FREQ sets the audio cutoff frequencies for SNR based high cut. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0xc828

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | YMAX[7:0] | | | | | | | | YMIN[7:0] | | | | | | | |
| Default | 0xC8 | | | | | | | | 0x28 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | YMAX[7:0] | Sets the maximum cutoff frequency for SNR based high cut.<br>Range: 0-200 |
| 7:0 | YMIN[7:0] | Sets the minimum cutoff frequency for SNR based high cut.<br>Range: 0-200 |

### Property 0x3606. FM_HIGHCUT_SNR_ATTACK_TIME

FM_HIGHCUT_SNR_ATTACK_TIME sets the transition time for which SNR based high cut lowers the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMAX to YMIN assuming SNR makes a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 1-65535 |

### Property 0x3607. FM_HIGHCUT_SNR_RELEASE_TIME

FM_HIGHCUT_SNR_RELEASE_TIME sets the transition time for which SNR based high cut increases the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMIN to YMAX assuming SNR makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 1-65535 |

### Property 0x3608. FM_HIGHCUT_MULTIPATH_LIMITS

FM_HIGHCUT_MULTIPATH_LIMITS sets the multipath limits for multipath controlled stereo separation. The limits are in % AM modulation at 1kHz. They are also negated and therefor the max value represents the point at which limiting will begin. The MAX value must be less then the MIN value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x2D3C

Units: %

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x2D | | | | | | | | 0x3C | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of multipath which causes the maximum cutoff frequency.<br>Range: 0-255 |
| 7:0 | XMIN[7:0] | Value of multipath which causes the minimum cutoff frequency.<br>Range: 0-255 |

### Property 0x3609. FM_HIGHCUT_MULTIPATH_CUTOFF_FREQ

FM_HIGHCUT_MULTIPATH_CUTOFF_FREQ sets the audio cutoff frequencies for the multipath based high cut. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0xc828

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | YMAX[7:0] | | | | | | | | YMIN[7:0] | | | | | | | |
| Default | 0xC8 | | | | | | | | 0x28 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | YMAX[7:0] | Sets the maximum cutoff frequency for multipath based high cut.<br>Range: 0-200 |
| 7:0 | YMIN[7:0] | Sets the minimum cutoff frequency for multipath based high cut.<br>Range: 0-200 |

### Property 0x360A. FM_HIGHCUT_MULTIPATH_ATTACK_TIME

FM_HIGHCUT_MULTIPATH_ATTACK_TIME sets the transition time for which multipath based high cut lowers the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMAX to YMIN assuming multipath makes a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 1-65535 |

### Property 0x360B. FM_HIGHCUT_MULTIPATH_RELEASE_TIME

FM_HIGHCUT_MULTIPATH_RELEASE_TIME sets the transition time for which multipath based high cut increases the cutoff frequency. The transition time is the time it will take the cutoff frequency to go from YMIN to YMAX assuming multipath makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 1-65535 |

### Property 0x3700. FM_BLEND_RSSI_LIMITS

FM_BLEND_RSSI_LIMITS sets the RSSI limits for RSSI controlled stereo separation. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x2010

Units: dBµV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x20 | | | | | | | | 0x10 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of RSSI which causes maximum stereo separation.<br>Range: -20-120 |
| 7:0 | XMIN[7:0] | Value of RSSI which causes minimum stereo separation.<br>Range: -20-120 |

### Property 0x3702. FM_BLEND_RSSI_ATTACK_TIME

FM_BLEND_RSSI_ATTACK_TIME ms The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 16-65535 |

### Property 0x3703. FM_BLEND_RSSI_RELEASE_TIME

FM_BLEND_RSSI_RELEASE_TIME sets the mono to stereo release time for RSSI based blend. The release time is the time it will take the stereo separation to go from YMIN to YMAX assuming RSSI makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 16-65535 |

### Property 0x3704. FM_BLEND_SNR_LIMITS

FM_BLEND_SNR_LIMITS sets the SNR limits for SNR controlled stereo separation. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x180F

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x18 | | | | | | | | 0x0F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of SNR which causes maximum stereo separation.<br>Range: -20-64 |
| 7:0 | XMIN[7:0] | Value of SNR which causes minimum stereo separation.<br>Range: -20-64 |

### Property 0x3706. FM_BLEND_SNR_ATTACK_TIME

FM_BLEND_SNR_ATTACK_TIME sets the stereo to mono attack time for SNR based blend. The attack time is the time it will take the stereo separation to go from YMAX to YMIN assuming SNR makes a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 16-65535 |

### Property 0x3707. FM_BLEND_SNR_RELEASE_TIME

FM_BLEND_SNR_RELEASE_TIME sets the mono to stereo release time for SNR based blend. The release time is the time it will take the stereo separation to go from YMIN to YMAX assuming SNR makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 16-65535 |

### Property 0x3708. FM_BLEND_MULTIPATH_LIMITS

FM_BLEND_MULTIPATH_LIMITS sets the multipath limits for multipath controlled stereo separation. The limits are in % AM modulation at 1kHz. They are also negated and therefor the max value represents the point at which limiting will begin. The MAX value must be less then the MIN value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x2D3C

Units: %

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| Default | 0x2D | | | | | | | | 0x3C | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of Multipath which causes maximum stereo separation.<br>Range: 0-255 |
| 7:0 | XMIN[7:0] | Value of Multipath which causes minimum stereo separation.<br>Range: 0-255 |

### Property 0x370A. FM_BLEND_MULTIPATH_ATTACK_TIME

FM_BLEND_MULTIPATH_ATTACK_TIME sets the stereo to mono attack time for multi-path based blend. The attack time is the time it will take the stereo separation to go from YMAX to YMIN assuming multipath makes a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 16

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 16-65535 |

### Property 0x370B. FM_BLEND_MULTIPATH_RELEASE_TIME

FM_BLEND_MULTIPATH_RELEASE_TIME sets the mono to stereo release time for multi-path based blend. The release time is the time it will take the stereo separation to go from YMIN to YMAX assuming multipath makes a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 4000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0FA0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 16-65535 |

### Property 0x3900. FM_AUDIO_DE_EMPHASIS

FM_AUDIO_DE_EMPHASIS property sets the FM Receive de-emphasis to 50 or 75 us. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode. The default is 75 us.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | DE_EMPH[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x0 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | DE_EMPH[1:0] | 0: 75 µs—Standard in USA<br>1: 50 µs—Standard in Europe<br>2: De-emphasis disabled. |

**Property 0x3C00. FM_RDS_INTERRUPT_SOURCE**

FM_RDS_INTERRUPT_SOURCE configures interrupt related to RDS. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode. Note: FM_RDS_STATUS command is not supported in firmware revision A0A.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | RDSTPPTY | RDSPI | Reserved | RDSSYNC | RDSRECV |
| Default | 0x000 | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4 | RDSTPPTY | If set, generates RDS Interrupt when new valid Block B data has been received.<br>0: Disabled<br>1: Enabled |
| 3 | RDSPI | If set, generates RDS Interrupt when new valid Block A data has been received.<br>0: Disabled<br>1: Enabled |
| 2 | Reserved | Always write to 0. |
| 1 | RDSSYNC | If set, generates RDS interrupt when RDS Synchronization changes.<br>0: Disabled<br>1: Enabled |
| 0 | RDSRECV | If set, generate an interrupt whenever the RDS FIFO has at least FM_RDS_INTERRUPT_FIFO_COUNT entries.<br>0: Disabled<br>1: Enabled |

### Property 0x3C01. FM_RDS_INTERRUPT_FIFO_COUNT

FM_RDS_INTERRUPT_FIFO_COUNT sets the minimum number of RDS groups stored in the RDS FIFO before RDSRECV is set. RDSRECV is disabled if set to 0. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | DEPTH[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | DEPTH[7:0] | Sets the minimum number of RDS Groups stored in the RDS FIFO required before RDSRECV is set. RDSRECV is disabled if set to 0.<br>Range: 0-25 |

### Property 0x3C02. FM_RDS_CONFIG

FM_RDS_CONFIG configures RDS settings to enable RDS processing (RDSEN) and set RDS block error thresholds. When a RDS Group is received, all block errors must be less than or equal to the associated block error threshold for the group to be stored in the RDS FIFO. If blocks with errors are permitted into the FIFO, the block error information can be reviewed when the group is read using the FM_RDS_STATUS command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | BLETHB[1:0] | | BLETHCD[1:0] | | Reserved | | | RDSEN |
| Default | 0x00 | | | | | | | | 0x0 | | 0x0 | | 0x0 | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:6 | BLETHB[1:0] | Block Error Threshold For BLOCKB.<br>Recommended value for BLETHB: 0, 1 or 2. Block B is most critical because it tells what C and D contain. Adjust depending on how tolerant the user wants to be on errors, with lower numbers being less tolerant.<br>0: No block errors<br>1: 1-2 bit errors detected and corrected<br>2: 3-5 bit errors detected and corrected<br>3: Uncorrectable |
| 5:4 | BLETHCD[1:0] | Block Error Threshold For BLOCKC and BLOCKD. The error count used to determine if the group is stored in the FIFO is min(BLEC,BLED).<br>Recommended value for BLETHCD: 0, 1 or 2.<br>0: No block errors<br>1: 1-2 bit errors detected and corrected<br>2: 3-5 bit errors detected and corrected<br>3: Uncorrectable |
| 3:1 | Reserved | Always write to 0x0. |
| 0 | RDSEN | Enables RDS Processing.<br>0: RDS Disabled.<br>1: RDS Enabled. |

**Property 0x3C03. FM_RDS_CONFIDENCE**

FM_RDS_CONFIDENCE sets the confidence threshold for deciding if each RDS block is valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x1111

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | CONFIDENCEA[3:0] | | | | CONFIDENCEB[3:0] | | | | CONFIDENCEC[3:0] | | | | CONFIDENCED[3:0] | | | |
| **Default** | 0x1 | | | | 0x1 | | | | 0x1 | | | | 0x1 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:12 | CONFIDEN-CEA[3:0] | Confidence threshold for Block A data.<br>Range: 1-15<br>0: Reserved.<br>1: Lowest confidence required. A valid and usable confidence threshold.<br>15: Highest confidence required. This setting may result in all blocks being marked as uncorrectable. |
| 11:8 | CONFI-DENCEB[3:0] | Confidence threshold for Block B data.<br>Range: 1-15<br>0: Reserved.<br>1: Lowest confidence required. A valid and usable confidence threshold.<br>15: Highest confidence required. This setting may result in all blocks being marked as uncorrectable. |
| 7:4 | CONFI-DENCEC[3:0] | Confidence threshold for Block C data.<br>Range: 1-15<br>0: Reserved.<br>1: Lowest confidence required. A valid and usable confidence threshold.<br>15: Highest confidence required. This setting may result in all blocks being marked as uncorrectable. |
| 3:0 | CONFI-DENCED[3:0] | Confidence threshold for Block D data.<br>Range: 1-15<br>0: Reserved.<br>1: Lowest confidence required. A valid and usable confidence threshold.<br>15: Highest confidence required. This setting may result in all blocks being marked as uncorrectable. |

### Property 0x8100. DIGITAL_SERVICE_INT_SOURCE

DIGITAL_SERVICE_INT_SOURCE configures which digital service events will set the DSRVINT status bit. When one of the bits described below is set, the corresponding event will cause the DSRVINT bit of the status word to be set. To clear the DSRVINT bit the GET_DIGITAL_SERVICE_DATA command must be executed with the ack option. If the DSRVIEN interrupt is enabled, a host interrupt will also be generated. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | DSRVOVFLINT | DSRVPCKTINT |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | DSRVOVFLINT | Configures the data service overflow interrupt. When this interrupt occurs data service data has been lost and indicates that the host is not retrieving the data service data in a timely fashion.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 0 | DSRVPCKTINT | Enables the DSRVPAKTINT interrupt of the GET_DATA_SERVICE_DATA command.<br>0: Interrupt disabled<br>1: Interrupt enabled |

### Property 0x9101. HD_BLEND_OPTIONS

HD_BLEND_OPTIONS provides options to control HD/analog audio blend behavior. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0002

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | ACQ_LOSS[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x2 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | ACQ_LOSS[1:0] | Upon digital acquisition loss, blend to silence or analog.<br>This option only takes effect if using hybrid tune mode.<br>0: Always blend to analog: Acq. loss, blend to analog, reacquire, blend to previously selected service.<br>1: Always blend to silence: Acq. loss, blend to silence, reacquire, blend to previously selected service.<br>2: Service dependent blend: MPS Selected: Acq. loss, blend to analog, reacquire, blend to MPS. SPSn Selected: Acq. loss, blend to silence, reacquire, blend to previous SPSn. |

### Property 0x9102. HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME

HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME sets the amount of time it takes in ms to blend from analog to HD. This property only applies to primary service channel. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 750

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BLEND_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x02EE | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | BLEND_TIME[15:0] | The amount of time in ms it takes blend from analog reception to HD reception. |

**Property 0x9103. HD_BLEND_HD_TO_ANALOG_TRANSITION_TIME**

HD_BLEND_HD_TO_ANALOG_TRANSITION_TIME sets the amount of time it takes in ms to blend from HD to analog. This property only applies to primary service channel. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 100

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BLEND_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0064 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | BLEND_TIME[15:0] | The amount of time in ms it takes blend from analog reception to HD reception. |

**Property 0x9106. HD_BLEND_DYNAMIC_GAIN**

HD_BLEND_DYNAMIC_GAIN sets the digital audio dynamic linear scaling factor. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x7F

Units: Q0.7

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | DGAIN[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | DGAIN[7:0] | Station dependent linear scaling factor (Q0.7) format.<br>Range: 0-0x7f |

**Property 0x9200. HD_DIGRAD_INTERRUPT_SOURCE**

HD_DIGRAD_INTERRUPT_SOURCE configures interrupts related to digital receiver (HD_DIGRAD_STATUS). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HDLOGOINTEN | SRCANAINTEN | SRCDIGINTEN | Reserved | | ACQINTEN | CDNRHINTEN | CDNRLINTEN |
| Default | 0x00 | | | | | | | | 0 | 0 | 0 | 0x0 | | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | HDLOGOINTEN | Enables the HD logo interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 6 | SRCANAINTEN | Enables the analog audio source interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 5 | SRCDIGINTEN | Enables the digital audio source interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 4:3 | Reserved | Always write to 0x0. |
| 2 | ACQINTEN | Ensemble acquisition change interrupt enable.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 1 | CDNRHINTEN | Enables the CDNR high interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 0 | CDNRLINTEN | Enables the CDNR low interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |

### Property 0x9201. HD_DIGRAD_CDNR_LOW_THRESHOLD

HD_DIGRAD_CDNR_LOW_THRESHOLD sets the CDNR level (in dB) below which the CDNRLINT interrupt will occur. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CDNR_LOW_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | CDNR_LOW_-THRESHOLD[15:0] | This property sets the CDNR level (in dB) below which the CDNRLINT interrupt will occur. To make this interrupt set the DACQINT in the status word the CDN-RLINTEN bit must be set in the HD_DIGRAD_INTERRUPT_SOURCE property. |

### Property 0x9202. HD_DIGRAD_CDNR_HIGH_THRESHOLD

HD_DIGRAD_CDNR_HIGH_THRESHOLD sets the CDNR level (in dB) above which the CDNRHINT interrupt will occur. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CDNR_HIGH_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x007F | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | CDNR_HIGH_-THRESHOLD[15:0] | This property sets the CDNR level (in dB) above which the CDNRHINT interrupt will occur. To make this interrupt set the DACQINT in the status word the CDN-RHINTEN bit must be set in the HD_DIGRAD_INTERRUPT_SOURCE property. |

### Property 0x9203. HD_DIGRAD_AUTO_ACQUIRE

HD_DIGRAD_AUTO_ACQUIRE sets the radio to automatically acquire digital radio. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | AUTO_ACQUIRE |
| Default | 0x0000 | | | | | | | | | | | | | | | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | AUTO_ACQUIRE | This option controls automatic acquire.<br>0: Do not acquire digital radio after tune. HD_ACQUIRE must be called to start the acquisition process.<br>1: Automatically acquire digital radio after tune. |

**Property 0x9300. HD_EVENT_INTERRUPT_SOURCE**

HD_EVENT_INTERRUPT_SOURCE property configures interrupts related to HD Events (see DEVENTINT status bit). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | DIN-FO_INTEN | AIN-FO_INTEN | Reserved | ALERT_INTEN | PSD_INTEN | SIS_INTEN | DSRVLIST_INTEN | ASRVLIST_INTEN |
| Default | 0x00 | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | DINFO_INTEN | Enables the Data Info interrupt.<br>0: Disables the Data Info interrupt.<br>1: Enable interrupt to occur whenever a updated data info is available. |
| 6 | AINFO_INTEN | Enables the Audio Info interrupt.<br>0: Disables the Audio Info interrupt.<br>1: Enable interrupt to occur whenever a updated audio info is available. |
| 5 | Reserved | Always write to 0. |
| 4 | ALERT_INTEN | Enables the Alert interrupt.<br>0: Disables the Alert interrupt.<br>1: Enable interrupt to occur whenever a new Alert occurs. |
| 3 | PSD_INTEN | Enables the Program Service Data (PSD) interrupt.<br>0: Disables the PSD interrupt.<br>1: Enable interrupt to occur whenever PSD is available. |
| 2 | SIS_INTEN | Enables the Station Information (SIS) interrupt.<br>0: Disables the SIS interrupt.<br>1: Enable interrupt to occur whenever SIS is available. |
| 1 | DSRVLIST_INTEN | Enables the data service list event interrupt.<br>0: Disables the service list update interrupt.<br>1: Enable interrupt to occur whenever a data service list update occurs. |
| 0 | ASRVLIST_INTEN | Enables the audio service list event interrupt.<br>0: Disables the audio service list update interrupt.<br>1: Enable interrupt to occur whenever an audio service list update occurs. |

### Property 0x9301. HD_EVENT_SIS_CONFIG

HD_EVENT_SIS_CONFIG configures which basic SIS information is returned by the HD_GET_STATION_INFO command BASICSIS option. Takes effect at tune time. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0017

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | LOCATION | RSVD | NAME_LF | NAME_SF | ID |
| Default | 0x000 | | | | | | | | | | | 1 | 0 | 1 | 1 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4 | LOCATION | When set the station location will be returned when BASICSIS option is used in the HD_GET_STATION_INFO command. |
| 3 | RSVD | Reserved |
| 2 | NAME_LF | When set the long form station name will be returned when BASICSIS option is used with the HD_GET_STATION_INFO command. |
| 1 | NAME_SF | When set the short form station name will be returned when BASICSIS option is used with the HD_GET_STATION_INFO command. |
| 0 | ID | When set the station ID will be returned when BASICSIS option is used in the HD_GET_STATION_INFO command. |

**Property 0x9302. HD_EVENT_ALERT_CONFIG**

HD_EVENT_ALERT_CONFIG configures HD alerts. Alert information is returned by the HD_GET_ALERT_MSG command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | PLAY_TONE | ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | PLAY_TONE | When set the FMHD radio will automatically insert a 2 second tone into the audio every time an alert is sent by the broadcaster. This can be quite often so it is recommended that the host use the HD_PLAY_ALERT_TONE command to play a tone each time a unique alert message is received. |
| 0 | ENABLE | When set the Alert system is enabled. Please see the HD_GET_ALERT_MSG command and the HD_GET_EVENT_STATUS commands for more details about alerts. |

**Property 0x9500. HD_PSD_ENABLE**

HD_PSD_ENABLE sets which audio services will provide program service data. The PSD data is forwarded through the data service DSRV interface. Once PSD data is available it is forwarded automatically for each service that is enabled. The PSD data packet is formatted as a collection of ID3 tags that must be decoded by the host. Please refer to the DSRV Users guide and the iBiquity Advanced Applications Services Guide for more information on using the DSRV and PSD information. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ASPS7 | ASPS6 | ASPS5 | ASPS4 | ASPS3 | ASPS2 | ASPS1 | AMPS | SPS7 | SPS6 | SPS5 | SPS4 | SPS3 | SPS2 | SPS1 | MPS |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15 | ASPS7 | Always enable program services data for Secondary Program Service 7. This PSD will be forwarded with a CID of 0x5207. |
| 14 | ASPS6 | Always enable program services data for Secondary Program Service 6. This PSD will be forwarded with a CID of 0x5206. |
| 13 | ASPS5 | Always enable program services data for Secondary Program Service 5. This PSD will be forwarded with a CID of 0x5205. |
| 12 | ASPS4 | Always enable program services data for Secondary Program Service 4. This PSD will be forwarded with a CID of 0x5204. |
| 11 | ASPS3 | Always enable program services data for Secondary Program Service 3. This PSD will be forwarded with a CID of 0x5203. |
| 10 | ASPS2 | Always enable program services data for Secondary Program Service 2. This PSD will be forwarded with a CID of 0x5202. |
| 9 | ASPS1 | Always enable program services data for Secondary Program Service 1. This PSD will be forwarded with a CID of 0x5201. |
| 8 | AMPS | Always enable program services data for Main Program Service. This PSD will be forwarded with a CID of 0x5100. |
| 7 | SPS7 | Enable program services data for Secondary Program Service 7 when tuned to this service. This PSD will be forwarded with a CID of 0x5207. |
| 6 | SPS6 | Enable program services data for Secondary Program Service 6 when tuned to this service. This PSD will be forwarded with a CID of 0x5206. |
| 5 | SPS5 | Enable program services data for Secondary Program Service 5 when tuned to this service. This PSD will be forwarded with a CID of 0x5205. |
| 4 | SPS4 | Enable program services data for Secondary Program Service 4 when tuned to this service. This PSD will be forwarded with a CID of 0x5204. |
| 3 | SPS3 | Enable program services data for Secondary Program Service 3 when tuned to this service. This PSD will be forwarded with a CID of 0x5203. |
| 2 | SPS2 | Enable program services data for Secondary Program Service 2 when tuned to this service. This PSD will be forwarded with a CID of 0x5202. |
| 1 | SPS1 | Enable program services data for Secondary Program Service 1 when tuned to this service. This PSD will be forwarded with a CID of 0x5201. |
| 0 | MPS | Enable program services data for the Main Program Service when tuned to this service. This PSD will be forwarded with a CID of 0x5100. |

**Property 0x9501. HD_PSD_FIELD_MASK**

Default: 0xFFFF

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ID | OWNER | DESC | NAME | RECV | URL | VALID | PRICE | Reserved | TEXT | SHORT | LANG | GENRE | ALBUM | ARTIST | TITLE |
| Default | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15 | ID | UFID: Owner ID |
| 14 | OWNER | UFID: Owner |
| 13 | DESC | Commercial: Description |
| 12 | NAME | Commercial: Name of Seller |
| 11 | RECV | Commercial: Received As |
| 10 | URL | Commercial: Contact URL |
| 9 | VALID | Commercial: Valid Until |
| 8 | PRICE | Commercial: Price |
| 7 | Reserved | Always write to 1. |
| 6 | TEXT | Comment: The actual text |
| 5 | SHORT | Comment: Short Content Description |
| 4 | LANG | Comment: Language |
| 3 | GENRE | Genre |
| 2 | ALBUM | Album |
| 1 | ARTIST | Artist |
| 0 | TITLE | Title |

### Property 0x9700. HD_AUDIO_CTRL_FRAME_DELAY

HD_AUDIO_CTRL_FRAME_DELAY controls the value of the delay of decoded digital audio samples relative to the output of the audio quality indicator. For CODEC modes 0 and 2, the actual delay value is a sum of this parameter and the Digital Audio Delay for a given codec mode, see HD_CODEC properties, The maximum hold-off for FM is 18 audio frames (0x12). Any value exceeding this maximum is limited to 0x12. This parameter is applicable in hybrid service modes only. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 6

Units: frames

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | DELAY[3:0] | | | |
| Default | 0x000 | | | | | | | | | | | | 0x6 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:4 | Reserved | Always write to 0x000. |
| 3:0 | DELAY[3:0] | The number of audio frames to delay the audio.<br>Range: 4-21 |

### Property 0x9701. HD_AUDIO_CTRL_PROGRAM_LOSS_THRESHOLD

HD_AUDIO_CTRL_PROGRAM_LOSS_THRESHOLD controls the duration before reverting to MPS audio after an SPS audio program is removed or lost. The same value applies to all SPS audio programs. The timeout is calculated as follows: Timeout = Value X 1.486 seconds. To hold the SPS audio program indefinitely, a value of 0 should be used. In this case, if the SPS audio program is lost or terminated, the system will not automatically revert to MPS and the SPS audio output will be muted and will continue to dwell on the selected supplemental program until commanded otherwise by the HC/listener or until automatic re-acquisition occurs. The system will perform this automatic re-acquisition when its internal HD acquired parameter is false for 60 seconds. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: frames

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | TRESH[3:0] | | | |
| Default | 0x000 | | | | | | | | | | | | 0x0 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:4 | Reserved | Always write to 0x000. |
| 3:0 | TRESH[3:0] | The delay before reverting to MPS audio.<br>Range: 0-14 |

### Property 0x9702. HD_AUDIO_CTRL_BALL_GAME_ENABLE

HD_AUDIO_CTRL_BALL_GAME_ENABLE selects the audio output for hybrid waveforms when the TX Blend Control Status (BCTL_EN of HD_DIGRAD_STATUS) bits are set to 01 (i.e., ballgame mode). Since analog diversity delay is not applied by the transmitter in this state, the receiver must disable audio blending and force either analog or digital audio to avoid audible discontinuities. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | MODE |
| Default | 0x0000 | | | | | | | | | | | | | | | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | MODE | When 1 selects Ball Game mode.<br>Range: 0-1 |

### Property 0x9901. HD_CODEC_MODE_0_SAMPLES_DELAY

HD_CODEC_MODE_0_SAMPLES_DELAY property is used to perform fine time alignment between the HD digital audio and analog audio to ensure phase aligned blending. Each unit of sample delay represents approximately 22.7us and this delay is applied to the HD audio. Note: Changes take effect upon hd_acquire or tune command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 3700

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| **Default** | 0x0 | | 0x0E74 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x9904. HD_CODEC_MODE_2_SAMPLES_DELAY

HD_CODEC_MODE_2_SAMPLES_DELAY property is used to perform fine time alignment between the HD digital audio and analog audio to ensure phase aligned blending. Each unit of sample delay represents approximately 22.7us and this delay is applied to the HD audio. Note: Changes take effect upon hd_acquire or tune command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| **Default** | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x9907. HD_CODEC_MODE_10_SAMPLES_DELAY

HD_CODEC_MODE_10_SAMPLES_DELAY property is used to perform fine time alignment between the HD digital audio and analog audio to ensure phase aligned blending. Each unit of sample delay represents approximately 22.7us and this delay is applied to the HD audio. Note: Changes take effect upon hd_acquire or tune command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| **Default** | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x990A. HD_CODEC_MODE_13_SAMPLES_DELAY

HD_CODEC_MODE_13_SAMPLES_DELAY property is used to perform fine time alignment between the HD digital audio and analog audio to ensure phase aligned blending. Each unit of sample delay represents approximately 22.7us and this delay is applied to the HD audio. Note: Changes take effect upon hd_acquire or tune command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| **Default** | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0xE800. HD_TEST_BER_CONFIG

HD_TEST_BER_CONFIG Enables the HD BER test. The HD BER test requires the special test vector IB_FMr208c_e1wfc204. To run this test start the vector signal generator with the BER test vector, tune to the frequency indicated on the generator, and enable the test using this property. Once the test is running collect the BER information using the HD_TEST_GET_BER_INFO command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | ENABLE | Enables the BER test. Once enabled the BER information can be collected using the HD_TEST_GET_BER_INFO command. |

### Property 0xE801. HD_TEST_DEBUG_AUDIO

HD_TEST_DEBUG_AUDIO is used to put the DAC audio output in to a special test mode for debug purposes. This is typically used for performing time alignment between the analog audio and the HD audio. The user has the option of selecting either the left or right audio from both the analog demod and the HD demod at the same time. Changes to the system take effect on the first tune following a change to this property. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | TEST-MODE[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x0 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | TESTMODE[1:0] | Selects which FM analog and HD audio channel to render.<br>0: Normal operation<br>1: Left output contains HD audio L, Right output contain FM audio L.<br>2: Left output contains HD audio R, Right output contain FM audio R.<br>3: Reserved, do not use. |

## 4.3. Commands Summary—DAB

**Table 5. Commands Summary—DAB**

| Cmd | Name | Description |
|---|---|---|
| 0x00 | RD_REPLY | Returns the status byte and data for the last command sent to the device. |
| 0x01 | POWER_UP | Power-up the device and set system settings. |
| 0x04 | HOST_LOAD | Loads an image from HOST over command interface |
| 0x05 | FLASH_LOAD | Loads an image from external FLASH over secondary SPI bus |
| 0x06 | LOAD_INIT | Prepares the bootloader to receive a new image. |
| 0x07 | BOOT | Boots the image currently loaded in RAM. |
| 0x08 | GET_PART_INFO | Reports basic information about the device. |
| 0x09 | GET_SYS_STATE | Reports system state information. |
| 0x0A | GET_POWER_UP_ARGS | Reports basic information about the device such as arguments used during POWER_UP. |
| 0x10 | READ_OFFSET | Reads a portion of response buffer from an offset. |
| 0x12 | GET_FUNC_INFO | Returns the Function revision information of the device. |
| 0x13 | SET_PROPERTY | Sets the value of a property. |
| 0x14 | GET_PROPERTY | Retrieve the value of a property. |
| 0x15 | WRITE_STORAGE | Writes data to the on board storage area at a specified offset. |
| 0x16 | READ_STORAGE | Reads data from the on board storage area from a specified offset. |
| 0x80 | GET_DIGITAL_SERVICE_LIST | Gets a service list of the ensemble. |
| 0x81 | START_DIGITAL_SERVICE | Starts an audio or data service. |
| 0x82 | STOP_DIGITAL_SERVICE | Stops an audio or data service. |
| 0x84 | GET_DIGITAL_SERVICE_DATA | Gets a block of data associated with one of the enabled data components of a digital services. |
| 0xB0 | DAB_TUNE_FREQ | Tunes the DAB Receiver to tune to a frequency between 168.16 and 239.20 MHz defined by the frequency table through DAB_SET_FREQ_LIST. |
| 0xB2 | DAB_DIGRAD_STATUS | Returns status information about the digital radio and ensemble. |
| 0xB3 | DAB_GET_EVENT_STATUS | Gets information about the various events related to the DAB radio. |

**Table 5. Commands Summary—DAB (Continued)**

| Cmd | Name | Description |
|---|---|---|
| 0xB4 | DAB_GET_ENSEMBLE_INFO | Gets information about the current ensemble |
| 0xB7 | DAB_GET_SERVICE_LINKING_INFO | Provides service linking info for the passed in service ID. |
| 0xB8 | DAB_SET_FREQ_LIST | Sets the DAB frequency table. The frequencies are in units of kHz. |
| 0xB9 | DAB_GET_FREQ_LIST | Gets the DAB frequency table |
| 0xBB | DAB_GET_COMPONENT_INFO | Gets information about components within the ensemble if available. |
| 0xBC | DAB_GET_TIME | Gets the ensemble time adjusted for the local time offset or the UTC. |
| 0xBD | DAB_GET_AUDIO_INFO | Gets audio service info |
| 0xBE | DAB_GET_SUBCHAN_INFO | Gets sub-channel info |
| 0xBF | DAB_GET_FREQ_INFO | Gets ensemble freq info |
| 0xE5 | TEST_GET_RSSI | Returns the reported RSSI in 8.8 format. |
| 0xE8 | DAB_TEST_GET_BER_INFO | Reads the current BER rate |

### 4.3.1. DAB Commands

**Command 0x00. RD_REPLY**

RD_REPLY command must be called to return the status byte and data for the last command sent to the device. This command is also used to poll the status byte as needed. To poll the status byte, send the RD_REPLY command and read the status byte. This can be done regardless of the state of the CTS bit in the status register. Please refer to individual command descriptions for the format of returned data. RD_REPLY is a hardware command and can be issued while device is powered down. For commands where the size of the response is returned, the user should send the RD_REPLY command to read the SIZE first. Each time the RD_REPLY command is sent, the STAUS bytes will still be returned.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x00 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA_0[7:0] | | | | | | | |
| RESP5 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |
| ERR_CMD | Command Error.<br>0: No error<br>1: Error. The previous command failed. |

| Name | Function |
|---|---|
| DACQINT | Digital radio link change interrupt indicator. Indicates that something in the digital radio ensemble acquisition status has changed.<br>Service via the DAB_DIGRAD_STATUS or the HD_DIGRAD_STATUS commands. |
| DSRVINT | Indicates that an enabled data component of one of the digital services requires attention. Service using the GET_DIGITAL_SERVICE_DATA command. |
| STCINT | Seek/Tune complete.<br>0: Tune complete has not been triggered. Do not send a new TUNE/SEEK command.<br>1: Tune complete has been triggered. It is safe to send a new TUNE/SEEK command. |
| DEVNTINT | Digital radio event change interrupt indicator. Indicates that a new event related to the digital radio has occurred.<br>Service via the DAB_GET_EVENT_STATUS or HD_GET_EVENT_STATUS commands. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| DSPERR | The DSP has encountered a frame overrun. This is a fatal error. |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |
| DATA_0[7:0] | First Data byte. |
| DATA_N[7:0] | Nth Data byte. |

### Command 0x01. POWER_UP

The POWER_UP initiates the boot process to move the device from power down to power up mode. There are two possible boot scenarios: Host image load and FLASH image load. When the host is loading the image the host first executes the POWER_UP command to set the system settings REF_CLK, etc). A LOAD_INIT command then prepares the bootloader to receive a new image. After the LOAD_INIT command, using the HOST_LOAD command loads the image into the device RAM. After the RAM is loaded the host issues the BOOT command. When booting a FLASH image the host issues the POWER_UP command to set the system settings. Then issues the FLASH_LOAD command to select and load the image from FLASH. Once the image is loaded the host sends the BOOT command to boot the application. Power-up is complete when the CTS bit is set. This command may only be sent while the device is powered down. Note: FLASH_LOAD is not supported in A0A or A0B revisions.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x01 | | | | | | | |
| ARG1 | CTSIEN | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ARG2 | 0 | 0 | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| ARG3 | 0 | IBIAS[6:0] | | | | | | |
| ARG4 | XTAL_FREQ[7:0] | | | | | | | |
| ARG5 | XTAL_FREQ[15:8] | | | | | | | |
| ARG6 | XTAL_FREQ[23:16] | | | | | | | |
| ARG7 | XTAL_FREQ[31:24] | | | | | | | |
| ARG8 | 0 | 0 | CTUN[5:0] | | | | | |
| ARG9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | 0 | 0 | | 0 | 0 | | |
| ARG12 | 0 | | | | 0 | 0 | | |
| ARG13 | 0 | IBIAS_RUN[6:0] | | | | | | |
| ARG14 | 0 | | | | | | | |
| ARG15 | 0 | | | | | | | |

| Name | Function |
|---|---|
| CTSIEN | The bootloader will toggle a host interrupt line when CTS is available.<br>0: Disable toggling host interrupt line.<br>1: Enable toggling host interrupt line. |
| CLK_MODE[5:4] | Choose clock mode. See refclk spec sheet for more information<br>0: Oscillator and buffer are powered down.<br>1: Reference clock generator is in crystal mode.<br>2: Oscillator is off and circuit acts as single ended buffer.<br>3: Oscillator is off and circuit acts as differential buffer. |
| TR_SIZE[3:0] | XOSC TR_SIZE. See refclk spec sheet for more information.<br>Range: 0-15 |
| IBIAS[6:0] | XTAL IBIAS current at startup. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 0 to 1270 µA.<br>Range: 0-127 |
| XTAL_FREQ[31:0] | XTAL Frequency in Hz. The supported crystal frequencies are: [5.4 MHz–6.6 MHz] [10.8 MHz–13.2 MHz] [16.8 MHz–19.8 MHz] [21.6 MHz–26.4 MHz] [27 MHz–46.2 MHz] |
| CTUN[5:0] | CTUN. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator.<br>Range: 0-63 |
| IBIAS_RUN[6:0] | XTAL IBIAS current at runtime, after the XTAL oscillator has stabiliized. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 10 to 1270 µA. If set to 0, will use the same value as IBIAS.<br>Range: 0-127 |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |
| ERR_CMD | Error.<br>0: No error<br>1: Error. The previous command failed. If STATUS1 is non-zero, it provides an indication of the cause of the error. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP + command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| RSVD_STAT[5:4] | RFU (Reserved For Future Use). |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |

### Command 0x04. HOST_LOAD

HOST_LOAD loads an image from HOST over command interface. It sends up to 4096 bytes of application image to the bootloader. Note: This command is much more efficient when the image is sent as multiples of 4 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x04 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG3 | 0x00 | | | | | | | |
| ARG4 | IMAGE_DATA0[7:0] | | | | | | | |
| ARG5 | IMAGE_DATA1[7:0] | | | | | | | |
| ARG6 | IMAGE_DATA2[7:0] | | | | | | | |
| ARG7 | IMAGE_DATA3[7:0] | | | | | | | |
| ARG8 | IMAGE_DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| IMAGE_DATA0[7:0] | First byte of data stream from boot_img. |
| IMAGE_DATA1[7:0] | Second byte of data stream from boot_img. |
| IMAGE_DATA2[7:0] | Third byte of data stream from boot_img. |
| IMAGE_DATA3[7:0] | Fourth byte of data stream from boot_img. |
| IMAGE_DATA_N[7:0] | Nth byte of data stream. Maximum of 4096 bytes of data per HOST_LOAD command. For best results, N should be a multiple of 4. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x05. FLASH_LOAD

FLASH_LOAD loads the firmware image from an externally attached SPI flash over the secondary SPI bus. The image must be contiguous on the flash. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: FLASH_LOAD is only supported after patching the bootloader.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x05 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | FLASH_START_ADDR[7:0] | | | | | | | |
| ARG5 | FLASH_START_ADDR[15:8] | | | | | | | |
| ARG6 | FLASH_START_ADDR[23:16] | | | | | | | |
| ARG7 | FLASH_START_ADDR[31:24] | | | | | | | |
| ARG8 | 0 | | | | | | | |
| ARG9 | 0 | | | | | | | |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | | | | | | | |

| Name | Function |
|---|---|
| FLASH_START_ADDR[31:0] | Flash byte starting address of image to load |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPOFERR | CMDOFERR | ARBERR | ERRNR |

### Command 0x06. LOAD_INIT

LOAD_INIT prepares the bootloader to receive a new image. It will force the bootloader state to waiting for a new LOAD command (HOST_LOAD or FLASH_LOAD.) LOAD_INIT command must always be sent prior to a HOST_LOAD or a FLASH_LOAD command. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x06 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x07. BOOT

BOOT command boots the image currently loaded in RAM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x07 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x08. GET_PART_INFO

GET_PART_INFO reports basic information about the device such as Part Number, Part Version, ROM ID, etc. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_PART_INFO command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x08 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | CHIPREV[7:0] | | | | | | | |
| RESP5 | ROMID[7:0] | | | | | | | |
| RESP6 | X | | | | | | | |
| RESP7 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP8 | PART[7:0] | | | | | | | |
| RESP9 | PART[15:8] | | | | | | | |
| RESP10 | X | | | | | | | |
| RESP11 | X | | | | | | | |
| RESP12 | X | | | | | | | |
| RESP13 | X | | | | | | | |
| RESP14 | X | | | | | | | |
| RESP15 | X | | | | | | | |
| RESP16 | X | | | | | | | |
| RESP17 | X | | | | | | | |
| RESP18 | X | | | | | | | |
| RESP19 | X | | | | | | | |
| RESP20 | X | | | | | | | |
| RESP21 | X | | | | | | | |
| RESP22 | X | | | | | | | |

| Name | Function |
|---|---|
| CHIPREV[7:0] | Chip Mask Revision |
| ROMID[7:0] | ROM Id |
| PART[15:0] | Part Number (decimal) |

### Command 0x09. GET_SYS_STATE

GET_SYS_STATE reports basic system state information such as which mode is active; FM, DAB, etc. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_SYS_STATE command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x09 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | IMAGE[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |

| Name | Function |
|---|---|
| IMAGE[7:0] | This field indicates which firmware image processed this command.<br>0: Bootloader is active<br>1: FMHD is active<br>2: DAB is active<br>3: TDMB or data only DAB image is active<br>4: FMHD Demod is active<br>5: AMHD is active<br>6: AMHD Demod is active |

## Command 0x0A. GET_POWER_UP_ARGS

GET_POWER_UP_ARGS reports basic information about the device such as which parameters were used during power up. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x0A | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | | | | | | | |
| RESP5 | X | X | X | X | X | X | X | X |
| RESP6 | X | X | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| RESP7 | X | IBIAS[6:0] | | | | | | |
| RESP8 | XTAL_FREQ[7:0] | | | | | | | |
| RESP9 | XTAL_FREQ[15:8] | | | | | | | |
| RESP10 | XTAL_FREQ[23:16] | | | | | | | |
| RESP11 | XTAL_FREQ[31:24] | | | | | | | |
| RESP12 | X | X | CTUN[5:0] | | | | | |
| RESP13 | X | X | X | X | X | X | X | X |
| RESP14 | X | | | | | | | |
| RESP15 | X | X | X | | X | X | | |
| RESP16 | X | | | | X | X | | |
| RESP17 | X | IBIAS_RUN[6:0] | | | | | | |

| Name | Function |
|---|---|
| CLK_MODE[5:4] | See POWER_UP command |
| TR_SIZE[3:0] | See POWER_UP command |
| IBIAS[6:0] | See POWER_UP command |
| XTAL_FREQ[31:0] | See POWER_UP command |
| CTUN[5:0] | See POWER_UP command |
| IBIAS_RUN[6:0] | See POWER_UP command |

### Command 0x10. READ_OFFSET

READ_OFFSET is used for applications that cannot read the entire response buffer. This type of application can use this command to read the response buffer in segments. The host must pass in an offset from the beginning of the response buffer to indicate the starting point from which to read. This offset must be modulo 4. The response buffer remains intact as in the READ_REPLY command so that the response can be read again if needed. This function is available for both I2C and SPI mode. This is a software command, therefore it is best to read as much data in each calling as possible. This will reduce the overhead associated with using this command. It is recommended that the minimum reply size be on the order of 512 bytes. This means that for APIs that return less the 512 bytes the standard READ_REPLY should be used. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x10 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The offset from the beginning of the response buffer from where to begin reading. The OFFSET parameter must be modulo four. An error is returned otherwise. For example to read a 1024 byte response in two chucks the host can call the READ_MORE command twice. The first call would have OFFSET set to 0 and the second call would have OFFSET set to 512. In both cases the host will clock out 516 bytes of data. The first chunk will include 4 bytes for the status word plus 512 bytes of response. The second chunk will include 4 bytes of status word plus the remaining 512 bytes of response. The response sections will be concatenated to form the entire response. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the data chunk. |

### Command 0x12. GET_FUNC_INFO

GET_FUNC_INFO returns the function revision number for currently loaded firmware (FMHD, DAB, AM etc.) as opposed to GET_PART_INFO command that provides the revision number for the combo firmware. For example, GET_PART_INFO would return A0B is the firmware revision while GET_FUNC_INFO would return 1.0.4 for FM function revision if the currently running firmware function is FM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x12 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | REVEXT[7:0] | | | | | | | |
| RESP5 | REVBRANCH[7:0] | | | | | | | |
| RESP6 | REVINT[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP7 | NOSVN | X | LOCATION[1:0] | | X | X | MIXE-DREV | LOCALMOD |
| RESP8 | SVNID[7:0] | | | | | | | |
| RESP9 | SVNID[15:8] | | | | | | | |
| RESP10 | SVNID[23:16] | | | | | | | |
| RESP11 | SVNID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| REVEXT[7:0] | Major revision number (first part of 1.2.3). |
| REVBRANCH[7:0] | Minor revision number (second part of 1.2.3). |
| REVINT[7:0] | Build revision number (third part of 1.2.3). |
| NOSVN | If set the build was created with no SVN info. This image cannot be tracked back to the SVN repo. |
| LOCATION[5:4] | The location from which the image was built (Trunk, Branch or Tag).<br>0x0: The image was built from an SVN tag. Revision numbers are valid.<br>0x1: The image was built from an SVN branch. Revision numbers will be 0.<br>0x2: The image was built from the trunk. Revision number will be 0. |
| MIXEDREV | If set, the image was built with mixed revisions. |
| LOCALMOD | If set, the image has local modifications. |
| SVNID[31:0] | SVN ID from which the image was built. |

### Command 0x13. SET_PROPERTY

SET_PROPERTY sets the value of a property. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x13 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG4 | DATA0[7:0] | | | | | | | |
| ARG5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| PROPID[15:0] | The property ID of the property to set. |
| DATA0[15:0] | Value for the written property. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

## Command 0x14. GET_PROPERTY

GET_PROPERTY retrieves the value of a property or properties. The host may read as many properties as desired up to the end of a given property group. An attempt to read passed the end of the property group will result in zeros being read. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x14 | | | | | | | |
| ARG1 | COUNT[7:0] | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |

| Name | Function |
|---|---|
| COUNT[7:0] | The number of properties to read. |

| Name | Function |
|---|---|
| PROPID[15:0] | The id of the property to retrieve. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |
| RESP5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[15:0] | Value of the first property. |

## Command 0x15. WRITE_STORAGE

WRITE_STORAGE writes data to the no board storage area at the specified offset. the largest block of data that can be written at one time is 256 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x15 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |
| ARG4 | LENGTH[7:0] | | | | | | | |
| ARG5 | LENGTH[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG6 | 0x00 | | | | | | | |
| ARG7 | 0x00 | | | | | | | |
| ARG8 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer at which the data will be written. |
| LENGTH[15:0] | The number of bytes to write. |
| DATA0[7:0] | The first data byte to write. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x16. READ_STORAGE

READ_STORAGE reads data from the on board storage area from a specified offset. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x16 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer from which the data will be read. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the read data. |

## Command 0x80. GET_DIGITAL_SERVICE_LIST

GET_DIGITAL_SERVICE_LIST gets a service list of the ensemble. This command should be issued each time an audio or data service list is updated as indicated by the SVRLISTINT bit of the DAB_GET_EVENT_STATUS command. This occurs shortly after tune time when a digital radio tuning mode is selected and the ensemble has been acquired. Please refer to the (DAB Service List Information) section of this document for the format of DAB Service List. Please refer to iBiquity document: RX_IDD_2206 Appendix L (Get_All_Data_Services_Info and Get_All_Audio_Services_Info) for the format of the HD Radio Service List. In the case of HD this command also retrieves the audio or data service info when the appropriate service type option is selected. This service info is available whenever the AINFO or DINFO bit(s) are set in the HD_GET_EVENT_STATUS response. This audio and data information provides a quick look at the services in the ensemble and can be used to reduce scan time as this information is ready for parsing well before the service lists. The payload of these responses are defined in Table 5-4 of the RX_IDD_2206 main document. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x80 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | SERTYPE[1:0] | |

| Name | Function |
|---|---|
| SERTYPE[1:0] | Sets the type of service list to retrieve.<br>0: Get the audio(HD) or complete(DAB, DMB) service list.<br>1: Get the data(HD) service list.<br>2: Get the HD audio info.<br>3: Get the HD data info. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | SIZE[7:0] | | | | | | | |
| RESP5 | SIZE[15:8] | | | | | | | |
| RESP6 | DATA_0[7:0] | | | | | | | |
| RESP7 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| SIZE[15:0] | The size of the service list in bytes. |
| DATA_0[7:0] | A max of 2047 bytes of service information. Please see the supplemental TBD documentation. |
| DATA_N[7:0] | A max of 2047 bytes of service information. Please see the supplemental TBD documentation. N(max) = SIZE-2. |

### Command 0x81. START_DIGITAL_SERVICE

START_DIGITAL_SERVICE starts an audio or data service. This command is used for both DAB and HD audio and data services. To determine what services exist in an ensemble please use the GET_DIGITAL_SERVICE_LIST command. In HD radio applications the broadcaster does not always transmit this service information. In this case no data services are available but there may be multiple audio programs available. To view which audio services are available use the HD_DIGRAD_STATUS command's AUDIO_PROG_AVAIL field to see which audio programs can be selected. In addition the SERVICE_ID (service number) is not required when selecting an audio or data service. In this case please set the SERVICE_ID parameter to 0. I the case of starting an audio service, it is not required to stop a currently running audio service/program before starting a new one. The currently running audio service will be stopped automatically when the new service is requested. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x81 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SER-TYPE |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service to start for HD applications. For DAB/DMB applications there is no distinction between audio and data services so this parameter should be written as 0.<br>0: Select an audio service.<br>1: Select a data service. |

| Name | Function |
|---|---|
| SERVICE_ID[31:0] | The service's Service ID (DAB) or Service Number (HD). This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command. NOTE: For HD applications the is refered to as the service number and not the global service ID that is provided as part of the component information. |
| COMP_ID[31:0] | The service's component ID (DAB) or Port/Program Number (HD). This ID is found in the component section of the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

## Command 0x82. STOP_DIGITAL_SERVICE

STOP_DIGITAL_SERVICE stops an audio or data service. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x82 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SERTYPE |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service list to start.<br>0: Select an audio service.<br>1: Select a data service. |
| SERVICE_ID[31:0] | The service's service ID. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command. |
| COMP_ID[31:0] | The service's Component ID or Port Number of the service to stop. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x84. GET_DIGITAL_SERVICE_DATA

GET_DIGITAL_SERVICE_DATA gets a block of data associated with one of the enabled data components of a digital service. Information about this block of data is found in the data header that is returned at the beginning of the data block. In order to determine the ideal number of PAYLOAD bytes to read, the header information can be read first followed by a second read of the full (header + PAYLOAD) length—it is unnecessary to call GET_DIGITAL_SERVICE_DATA twice to use this method. The data associated with this transaction will be discarded at the receipt of a next GET_DIGITAL_SERVICE_DATA command if STATUS_ONLY = 0. Reading past the end of the buffer will result in zeros for the invalid bytes. Please refer to the (DAB Service List Information) section of this document for the format of DAB Service List. Please refer to iBiquity document: SY_IDD_1019s Rev F (sections 5 and 6) for the format of the HD Radio data service data. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x84 | | | | | | | |
| ARG1 | 0 | 0 | 0 | STATUS_ONLY | 0 | 0 | 0 | ACK |

| Name | Function |
|---|---|
| STATUS_ONLY | Returns only the interrupt source and available buffers information. No digital service data is removed from the service queue. This option should be used when polling for available data. |
| ACK | Acknowledge the reading of a data buffer or acknowledge one of the error interrupts. Whenever the ACK bit is set the DSRVINT bit in the status register will be cleared before CTS is released unless there are more data blocks to be read. All interrupt source bits associated with this command will also be cleared.<br>0: Don't acknowledge the interrupt<br>1: Acknowledging the interrupt will clear the DSRVINT bit and the interrupt source bits. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | X | X | DSR-VOV-FLINT | DSRVP-CKTINT |
| RESP5 | BUFF_COUNT[7:0] | | | | | | | |
| RESP6 | SRV_STATE[7:0] | | | | | | | |
| RESP7 | DATA_SRC[1:0] | | DSCTy[5:0] | | | | | |
| RESP8 | SERVICE_ID[7:0] | | | | | | | |
| RESP9 | SERVICE_ID[15:8] | | | | | | | |
| RESP10 | SERVICE_ID[23:16] | | | | | | | |
| RESP11 | SERVICE_ID[31:24] | | | | | | | |
| RESP12 | COMP_ID[7:0] | | | | | | | |
| RESP13 | COMP_ID[15:8] | | | | | | | |
| RESP14 | COMP_ID[23:16] | | | | | | | |
| RESP15 | COMP_ID[31:24] | | | | | | | |
| RESP16 | RFU[7:0] | | | | | | | |
| RESP17 | RFU[15:8] | | | | | | | |
| RESP18 | BYTE_COUNT[7:0] | | | | | | | |
| RESP19 | BYTE_COUNT[15:8] | | | | | | | |
| RESP20 | SEG_NUM[7:0] | | | | | | | |
| RESP21 | SEG_NUM[15:8] | | | | | | | |
| RESP22 | NUM_SEGS[7:0] | | | | | | | |
| RESP23 | NUM_SEGS[15:8] | | | | | | | |
| RESP24 | PAYLOAD0[7:0] | | | | | | | |
| RESP25 | PAYLOADN[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DSRVOVFLINT | The data services system has overflowed. This indicates that the host processor is not reading the services data out of the device fast enough. At most 8 outstanding data blocks can be queued in the device. |

| Name | Function |
|---|---|
| DSRVPCKTINT | Data for an enabled data service is ready for transfer to the host. |
| BUFF_COUNT[7:0] | Indicates the remaining number of buffers in the data service buffer queue. |
| SRV_STATE[7:0] | The status indicator for the associated service component.<br>0: Indicates that the service is playing out normally.<br>1: Indicates that the data service has stopped and that this is the last data block associated with the service.<br>2: Indicates that the system was not able to forward a packet due to a memory overflow. If this status is present the host is not reading the packet data fast enough.<br>3: Indicates the this data packet represents the beginning of a new data object.<br>4: Indicates the this data packet was received with errors. |
| DATA_SRC[7:6] | For DAB indicates the payload source. Not used in HD modes of operation.<br>0: Indicates that the payload is from a standard data service and DATA_TYPE is DSCTy.<br>1: Indicates that the payload is non-DLS PAD and DATA_TYPE is DSCTy.<br>2: Indicates that the payload is DLS PAD and DATA_TYPE is 0.<br>3: Reserved for future use. |
| DSCTy[5:0] | Used for DAB only. Returns 0 or the DSCTy depending on the DATA_SRC field. See DATA_SRC for details. |
| SERVICE_ID[31:0] | The Service ID this data is associated with. |
| COMP_ID[31:0] | The Component ID or Port Number this data is associated with. |
| RFU[15:0] | Reserved for future use. |
| BYTE_COUNT[15:0] | The length of this data block in bytes excluding the DSRV header. |
| SEG_NUM[15:0] | The segment number for this data block. If the data is associated with a stream this value represents a sequence number. Note that segment numbers can be returned out of order. Therefore the host must store and reassemble the data as needed. |
| NUM_SEGS[15:0] | The total number of segments to be returned for this data object. If NUM_SEGS=0 then no object length information is known or the associated service is a stream. |
| PAYLOAD0[7:0] | The first payload byte. |
| PAYLOADN[7:0] | The Nth payload byte where N = BYTE_COUNT-1. |

## Command 0xB0. DAB_TUNE_FREQ

DAB_TUNE_FREQ sets the DAB Receiver to tune to a frequency between 168.16 MHz and 239.20 MHz defined by the table through DAB_SET_FREQ_LIST. The optional STC interrupt is set when the command completes the tune. Sending this command clears any pending STCINT bit in the STATUS. The default list that will be used by the tuner is the European frequency list. To change this list (example: for T-DMB), the user must first call DAB_SET_FREQ_LIST before calling the DAB_TUNE_FREQ command. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB0 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | INJECTION[1:0] | |
| ARG2 | FREQ_INDEX[7:0] | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | ANTCAP[7:0] | | | | | | | |
| ARG5 | ANTCAP[15:8] | | | | | | | |

| Name | Function |
|---|---|
| INJECTION[1:0] | Injection selection<br>0: Automatic injection selection.<br>1: Low-side injection.<br>2: High-side injection. |
| FREQ_INDEX[7:0] | Frequency index for the tuned frequency, see the DAB_SET_FREQ_LIST command that sets the frequency table. |
| ANTCAP[15:0] | Antenna tuning capacitor value in 250 fF units (31.75 pF Max)<br>Range: 0-128<br>0: Automatically determines the cap setting. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN TINT | X | X | X | X | X |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |

## Command 0xB2. DAB_DIGRAD_STATUS

DAB_DIGRAD_STATUS returns status information about the digital radio and ensemble including a change in ensemble acquisition state, current estimates for ensemble's MSC (Main Service Channel) BER (bit error rate), FIC (Fast Information Channel) BER along with number of FIBs (Fast Information Block) that failed a CRC check and number of Reed-Solomon decoder errors (DAB+ and DMB only). The bits RSSILINT, RSSIHINT, ACQINT are sticky meaning they will remain set until DIGRAD_ACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in DAB_DIGRAD_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB2 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | DIGRAD_ACK | ATTUNE | 0 | STC_ACK |

| Name | Function |
|---|---|
| DIGRAD_ACK | Clears all pending digital radio interrupts. |
| ATTUNE | Return the values as of DAB_VALID_RSSI_TIME after tune. Only the signal quality metric RSSI is affected by setting this bit.<br>0: Return the current status<br>1: Return the snapshot taken at DAB_VALID_RSSI_TIME |
| STC_ACK | Clears the STC interrupt status when set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | HARD-MUTEINT | FICER-RINT | ACQINT | RSSI-HINT | RSSILINT |
| RESP5 | X | X | X | HARD-MUTE | FICERR | ACQ | X | VALID |
| RESP6 | RSSI[7:0] | | | | | | | |
| RESP7 | SNR[7:0] | | | | | | | |
| RESP8 | FIC_QUALITY[7:0] | | | | | | | |
| RESP9 | CNR[7:0] | | | | | | | |
| RESP10 | FIB_ERROR_COUNT[7:0] | | | | | | | |
| RESP11 | FIB_ERROR_COUNT[15:8] | | | | | | | |
| RESP12 | TUNE_FREQ[7:0] | | | | | | | |
| RESP13 | TUNE_FREQ[15:8] | | | | | | | |
| RESP14 | TUNE_FREQ[23:16] | | | | | | | |
| RESP15 | TUNE_FREQ[31:24] | | | | | | | |
| RESP16 | TUNE_INDEX[7:0] | | | | | | | |
| RESP17 | FFT_OFFSET[7:0] | | | | | | | |
| RESP18 | READANTCAP[7:0] | | | | | | | |
| RESP19 | READANTCAP[15:8] | | | | | | | |
| RESP20 | CU_LEVEL[7:0] | | | | | | | |
| RESP21 | CU_LEVEL[15:8] | | | | | | | |

| Name | Function |
|---|---|
| HARDMUTEINT | Indicates that the audio had been muted This is likely due to poor signal conditions. |
| FICERRINT | Indicates the FIC decoder has encountered unrecoverable errors. This is likely due to poor signal conditions. |
| ACQINT | Indicates a change in the ensemble acquisition state. |
| RSSIHINT | Indicates RSSI below DAB_DIGRAD_RSSI_LOW_THRESHOLD. |
| RSSILINT | Indicates RSSI above DAB_DIGRAD_RSSI_HIGH_THRESHOLD. |

| Name | Function |
|---|---|
| HARDMUTE | When set to 0 the audio is unmuted. When set to 1 the audio is hard muted. |
| FICERR | When set to 1 the ensemble is experiencing FIC errors. Signal quality has been degraded and acquisition may be lost. |
| ACQ | When set to 1 the ensemble is acquired. |
| VALID | When set to 1, the RSSI is at or above the valid threshold. It is recommended that the valid bit be used as part of tune validation. Once STC is set the valid bit can be checked to verify that then tune has passed both the RSSI valid threshold and that acquisition has been achived. The host should set the RSSI thershold, validation time and acquisition time to achieve solid tune time performance. Doing this helps insure an accurate tune indication and helps to decrease scan times due to quick station disqualification. |
| RSSI[7:0] | Received signal strength indicator.<br>Range: -128-63 |
| SNR[7:0] | Indicates the current estimate of the digital SNR in dB.<br>Range: 0-20 |
| FIC_QUALITY[7:0] | Indicates the current estimate of the ensembles FIC quality. The number is provided is between 0 and 100.<br>Range: 0-100 |
| CNR[7:0] | Indicates the current estimate of the CNR in dB. The CNR is the ratio of the OFDM signal level during the on period and during the off (null) period.<br>Range: 0-54 |
| FIB_ERROR_COUNT[15:0] | Indicates the num of Fast Information Blocks received with errors. |
| TUNE_FREQ[31:0] | indicates the currently tuned frequency in kHz. |
| TUNE_INDEX[7:0] | Indicates the currently tuned frequency index.<br>Range: 0-47 |
| FFT_OFFSET[7:0] | Indicates the frequency offset of the DQPSK tones of the OFDM signal relative to the center of the FFT bins of the digital demod. |
| READANTCAP[15:0] | Returns the antenna tuning cap value. |
| CU_LEVEL[15:0] | Returns the CU usage indicator (number of currently decoded CU's )<br>Range: 0-470 |

### Command 0xB3. DAB_GET_EVENT_STATUS

DAB_GET_EVENT_STATUS gets information about the various events related to the DAB radio. These events include signaling the reception of new PAD (Programme-Associated Data) data, service lists and announcements. The bits SVRLISTINT, ANNOINT, RECFGWRNINT, and RECFGINT are sticky meaning they will remain set until EVENT_ACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in DAB_EVENT_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB3 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EVENT_ACK |

| Name | Function |
|---|---|
| EVENT_ACK | Clears all pending digital radio event interrupts. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | RECFGINT | RECFGWRNINT | X | X | ANNOINT | X | FREQINFOINT | SVRLISTINT |
| RESP5 | X | X | X | X | X | X | FREQ_INFO | SVRLIST |
| RESP6 | SVRLISTVER[7:0] | | | | | | | |
| RESP7 | SVRLISTVER[15:8] | | | | | | | |

| Name | Function |
|---|---|
| RECFGINT | Ensemble reconfiguration event. Indicates that an ensemble reconfiguration has occurred. All changes to the service list that occured after the RECFGWRNINT event will now take effect. If a service that was in operation no longer exists it will be stopped. All other services that did not change should remain active accross the reconfiguration boundary. At this time the host should communicate any relevant changes to the user. |
| RECFGWRNINT | Ensemble reconfiguration warning. Indicates that an ensemble reconfiguration will occur in 6 seconds. From this point on all service list updates will apply to the new ensemble configuration. These changes will not take effect until the RECFGINT is received. At this time the host act upon all changes in the service list. |
| ANNOINT | New announcement info interrupt Indicates that new announcementinformation is available and can be retrieved using the DAB_GET_ANNOUNCEMENT_INFO command. |
| FREQINFOINT | New Frequency Information (FI) interrupt. Indicates that new Frequency Information is available. The Frequency Information list is retrieved with the DAB_GET_FREQ_INFO command. The rate at which frequency information interrupts can occur is defined by the DAB_EVENT_MIN_FREQINFO_PERIOD property. |
| SVRLISTINT | New service list interrupt. Indicates that a new digital service list is available. The new service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. |
| FREQ_INFO | Frequency Information (FI) available. Indicates that Frequency Information (FI) is available. The FI list is retrieved with the DAB_GET_FREQ_INFO command. |
| SVRLIST | Service list available. Indicates that a digital service list is available. The service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. If a service list is not available or it is in transition, this bit will be low. When the service list is in transition, this bit will remain low until the service list debounce timer has expired. See the DAB_EVENT_MIN_SVRLIST_PERIOD property for more details. |
| SVRLISTVER[15:0] | Indicates the current version of the digital service list. This field is incremented by 1 each time the service list is updated. The host can use this field to help determine if a new service list needs to be collected. |

### Command 0xB4. DAB_GET_ENSEMBLE_INFO

DAB_GET_ENSEMBLE_INFO gets information about the current ensemble such as the ensemble ID and label. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB4 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | EID[7:0] | | | | | | | |
| RESP5 | EID[15:8] | | | | | | | |
| RESP6 | LABEL0[7:0] | | | | | | | |
| RESP7 | LABEL1[7:0] | | | | | | | |
| RESP8 | LABEL2[7:0] | | | | | | | |
| RESP9 | LABEL3[7:0] | | | | | | | |
| RESP10 | LABEL4[7:0] | | | | | | | |
| RESP11 | LABEL5[7:0] | | | | | | | |
| RESP12 | LABEL6[7:0] | | | | | | | |
| RESP13 | LABEL7[7:0] | | | | | | | |
| RESP14 | LABEL8[7:0] | | | | | | | |
| RESP15 | LABEL9[7:0] | | | | | | | |
| RESP16 | LABEL10[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP17 | LABEL11[7:0] | | | | | | | |
| RESP18 | LABEL12[7:0] | | | | | | | |
| RESP19 | LABEL13[7:0] | | | | | | | |
| RESP20 | LABEL14[7:0] | | | | | | | |
| RESP21 | LABEL15[7:0] | | | | | | | |

| Name | Function |
|---|---|
| EID[15:0] | The ensemble ID EID. See section 6.4 of ETSI EN 300401. |
| LABEL0[7:0] | First of 16 characters for the ensemble label. |
| LABEL1[7:0] | The second of 16 characters of the component label. |
| LABEL2[7:0] | The third of 16 characters of the component label. |
| LABEL3[7:0] | The fourth of 16 characters of the component label. |
| LABEL4[7:0] | The fifth of 16 characters of the component label. |
| LABEL5[7:0] | The sixth of 16 characters of the component label. |
| LABEL6[7:0] | The seventh of 16 characters of the component label. |
| LABEL7[7:0] | The eighth of 16 characters of the component label. |
| LABEL8[7:0] | The nineth of 16 characters of the component label. |
| LABEL9[7:0] | The tenth of 16 characters of the component label. |
| LABEL10[7:0] | The eleventh of 16 characters of the component label. |
| LABEL11[7:0] | The twelfth of 16 characters of the component label. |
| LABEL12[7:0] | The thirteenth of 16 characters of the component label. |
| LABEL13[7:0] | The fourteenth of 16 characters of the component label. |
| LABEL14[7:0] | The fifteenth of 16 characters of the component label. |
| LABEL15[7:0] | The sixteenth of 16 characters of the component label. |

### Command 0xB7. DAB_GET_SERVICE_LINKING_INFO

DAB_GET_SERVICE_LINKING_INFO provides service linking info for the passed in service ID. Provides information on where to look for the alternate services or supplemental services relating to the passed in service ID. This could include another ensemble, another service within the current ensemble, or an FM broadcast. Please see clause 8.1.15 of ETSI 300-401 for further details. This command can return multiple links for a given service. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB7 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICEID[7:0] | | | | | | | |
| ARG5 | SERVICEID[15:8] | | | | | | | |
| ARG6 | SERVICEID[23:16] | | | | | | | |
| ARG7 | SERVICEID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERVICEID[31:0] | The service ID for which the service linking information will be provided. This service ID is provided by the digital service list. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | NUM_LINKS[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP5 | X | | | | | | | |
| RESP6 | LINK_BYTES[7:0] | | | | | | | |
| RESP7 | LINK_BYTES[15:8] | | | | | | | |
| RESP8 | ACTUATOR_0[7:0] | | | | | | | |
| RESP9 | HARDLINK_0[7:0] | | | | | | | |
| RESP10 | ILS_0[7:0] | | | | | | | |
| RESP11 | LINKTYPE_0[7:0] | | | | | | | |
| RESP12 | LINKID_0[7:0] | | | | | | | |
| RESP13 | LINKID_0[15:8] | | | | | | | |
| RESP14 | LINKID_0[23:16] | | | | | | | |
| RESP15 | LINKID_0[31:24] | | | | | | | |
| RESP16 | NUMALTS_0[7:0] | | | | | | | |
| RESP17 | NUMALTS_0[15:8] | | | | | | | |
| RESP18 | NUMALTS_0[23:16] | | | | | | | |
| RESP19 | NUMALTS_0[31:24] | | | | | | | |
| RESP20 | ALT_0_0[7:0] | | | | | | | |
| RESP21 | ALT_0_0[15:8] | | | | | | | |
| RESP22 | ALT_0_0[23:16] | | | | | | | |
| RESP23 | ALT_0_0[31:24] | | | | | | | |

| Name | Function |
|---|---|
| NUM_LINKS[7:0] | The number of service links available for service. The following 8 bytes are repeated NUM_LINKS times. |
| LINK_BYTES[15:0] | The total number of bytes in the service linking information payload. |
| ACTUATOR_0[7:0] | Indicates whether the link is active or inactive as follows. This field is repeated for each.<br>0: Link is not active.<br>1: Link is active. |
| HARDLINK_0[7:0] | Indicates the link type. This field is repeated for each link.<br>0: Link is a soft link. This type of link carries services that are related to SERVICEID.<br>1: Link is a hard link. This type of link carries the same primary service as SERVICEID. |

| Name | Function |
|---|---|
| ILS_0[7:0] | Indicates if the link is national or international. This field is repeated for each link.<br>0: Link is a national link.<br>1: Link is an international link. |
| LINKTYPE_0[7:0] | Indicates the link type as follows. This field is repeated for each link.<br>0: Link ID is a DAB/DMB service ID.<br>1: Link ID is and RDS PI-code.<br>2: Link ID is an AM or FM service.<br>3: Link ID is a DRM service indentifier. |
| LINKID_0[31:0] | The link ID. This field gets repeated for each link. |
| NUMALTS_0[31:0] | This is the number of alternate sources carrying the linked service. This field is repeated each link. |
| ALT_0_0[31:0] | This is the first alternate for LINK_0. When LINK_TYPE is 0 the field represents the ensemble ID of the ensemble carrying the service. If the LINK_TYPE is either 1 or 2 then this represents the frequency of the station carrying the alternate service. This field is obtained from Frequency Information, which can be retrieved with DAB_GET_FREQ_INFO command. This field may be empty until Frequency Information is available. This field is repeated each alternate as defined by NUMALTS. |

### Command 0xB8. DAB_SET_FREQ_LIST

DAB_SET_FREQ_LIST command sets the DAB frequency table. The frequencies are in units of 1 kHz. The table can be populated with a single entry or a regional list (for example 5 or 6 entries). It is recommended to make the list regional to increase scanning speed. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB8 | | | | | | | |
| ARG1 | NUM_FREQS[7:0] | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | FREQ_0[7:0] | | | | | | | |
| ARG5 | FREQ_0[15:8] | | | | | | | |
| ARG6 | FREQ_0[23:16] | | | | | | | |
| ARG7 | FREQ_0[31:24] | | | | | | | |
| ARG8 | FREQ_1[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG9 | FREQ_1[15:8] | | | | | | | |
| ARG10 | FREQ_1[23:16] | | | | | | | |
| ARG11 | FREQ_1[31:24] | | | | | | | |
| ARG12 | FREQ_N[7:0] | | | | | | | |
| ARG13 | FREQ_N[15:8] | | | | | | | |
| ARG14 | FREQ_N[23:16] | | | | | | | |
| ARG15 | FREQ_N[31:24] | | | | | | | |

| Name | Function |
|---|---|
| NUM_FREQS[7:0] | The number of frequencies in the table.<br>Range: 1-48 |
| FREQ_0[31:0] | The first frequency in the table of NUM_FREQS entries. |
| FREQ_1[31:0] | The second frequency in the table. |
| FREQ_N[31:0] | The last frequency in the table (N=NUM_FREQS-1). |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0xB9. DAB_GET_FREQ_LIST

DAB_GET_FREQ_LIST gets the DAB frequency table. All frequencies are in units of 1 kHz. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xB9 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | NUM_FREQS[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |
| RESP6 | X | | | | | | | |
| RESP7 | X | | | | | | | |
| RESP8 | FREQ_0[7:0] | | | | | | | |
| RESP9 | FREQ_0[15:8] | | | | | | | |
| RESP10 | FREQ_0[23:16] | | | | | | | |
| RESP11 | FREQ_0[31:24] | | | | | | | |

| Name | Function |
|---|---|
| NUM_FREQS[7:0] | The number of frequencies in the table. |
| FREQ_0[31:0] | The first frequency in the table. |

### Command 0xBB. DAB_GET_COMPONENT_INFO

DAB_GET_COMPONENT_INFO gets information about components within the ensemble if available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xBB | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICEID[7:0] | | | | | | | |
| ARG5 | SERVICEID[15:8] | | | | | | | |
| ARG6 | SERVICEID[23:16] | | | | | | | |
| ARG7 | SERVICEID[31:24] | | | | | | | |
| ARG8 | COMPID[7:0] | | | | | | | |
| ARG9 | COMPID[15:8] | | | | | | | |
| ARG10 | COMPID[23:16] | | | | | | | |
| ARG11 | COMPID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERVICEID[31:0] | The service ID |
| COMPID[31:0] | The component ID. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | GLOBAL_ID[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |
| RESP6 | X | X | LANG[5:0] | | | | | |
| RESP7 | X | X | CHARSETID[5:0] | | | | | |
| RESP8 | LABEL0[7:0] | | | | | | | |
| RESP9 | LABEL1[7:0] | | | | | | | |
| RESP10 | LABEL2[7:0] | | | | | | | |
| RESP11 | LABEL3[7:0] | | | | | | | |
| RESP12 | LABEL4[7:0] | | | | | | | |
| RESP13 | LABEL5[7:0] | | | | | | | |
| RESP14 | LABEL6[7:0] | | | | | | | |
| RESP15 | LABEL7[7:0] | | | | | | | |
| RESP16 | LABEL8[7:0] | | | | | | | |
| RESP17 | LABEL9[7:0] | | | | | | | |
| RESP18 | LABEL10[7:0] | | | | | | | |
| RESP19 | LABEL11[7:0] | | | | | | | |
| RESP20 | LABEL12[7:0] | | | | | | | |
| RESP21 | LABEL13[7:0] | | | | | | | |
| RESP22 | LABEL14[7:0] | | | | | | | |
| RESP23 | LABEL15[7:0] | | | | | | | |
| RESP24 | CHAR_ABREV[7:0] | | | | | | | |
| RESP25 | CHAR_ABREV[15:8] | | | | | | | |
| RESP26 | NUMUA[7:0] | | | | | | | |
| RESP27 | LENUA[7:0] | | | | | | | |
| RESP28 | UATYPE[7:0] | | | | | | | |
| RESP29 | UATYPE[15:8] | | | | | | | |
| RESP30 | UADATALEN[7:0] | | | | | | | |
| RESP31 | UADATA0[7:0] | | | | | | | |
| RESP32 | UADATAN[7:0] | | | | | | | |

| Name | Function |
|---|---|
| GLOBAL_ID[7:0] | The global reference for the component. See clause 6.3.5 of 300-401. This field gets concatenated with the Service ID of the service list to form the unique global component ID. |
| LANG[5:0] | The language of the component. See tables 9 and 10 of ETSI TS 101-756. |
| CHARSETID[5:0] | The character set for the component label. See tables 1 of ETSI TS 101-756. |
| LABEL0[7:0] | The first of 16 characters of the component label. |
| LABEL1[7:0] | The second of 16 characters of the component label. |
| LABEL2[7:0] | The third of 16 characters of the component label. |
| LABEL3[7:0] | The fourth of 16 characters of the component label. |
| LABEL4[7:0] | The fifth of 16 characters of the component label. |
| LABEL5[7:0] | The sixth of 16 characters of the component label. |
| LABEL6[7:0] | The seventh of 16 characters of the component label. |
| LABEL7[7:0] | The eighth of 16 characters of the component label. |
| LABEL8[7:0] | The nineth of 16 characters of the component label. |
| LABEL9[7:0] | The tenth of 16 characters of the component label. |
| LABEL10[7:0] | The eleventh of 16 characters of the component label. |
| LABEL11[7:0] | The twelfth of 16 characters of the component label. |
| LABEL12[7:0] | The thirteenth of 16 characters of the component label. |
| LABEL13[7:0] | The fourteenth of 16 characters of the component label. |
| LABEL14[7:0] | The fifteenth of 16 characters of the component label. |
| LABEL15[7:0] | The sixteenth of 16 characters of the component label. |
| CHAR_ABREV[15:0] | The component label abbreviation mask. Used to indicate which characters in the label are use to create the abbreviated label. |
| NUMUA[7:0] | The number of user application types. |
| LENUA[7:0] | The total length (in byte) of the UATYPE, UADATALEN and UADATA fields, including the padding bytes which is described in UADATAN field. |
| UATYPE[15:0] | The user application type. If multiple UA Types exist, all UATTYPE fields will be aligned on a 16-bit (2 byte) boundary. |
| UADATALEN[7:0] | The user application data field length, excluding the padding byte which is described in UADATAN field. |
| UADATA0[7:0] | The first user application data byte. |

| Name | Function |
|---|---|
| UADATAN[7:0] | The last user application data byte. If the user application data field length (UADATALEN) is odd, this byte will be a valid user application data byte. If the user application data field length (UADATALEN) is even, this byte will be a padding byte. The content of the padding byte is 0x00. By adding a padding byte, each UATYPE field will be aligned on a 16-bit (2 byte) boundary. The padding byte will not be counted in the UADATALEN field, but will be counted in the LENUA field. |

### Command 0xBC. DAB_GET_TIME

DAB_GET_TIME gets the ensemble time adjusted for the local time offset. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xBC | | | | | | | |
| ARG1 | TIME_TYPE[7:0] | | | | | | | |

| Name | Function |
|---|---|
| TIME_TYPE[7:0] | "Indicate the type of the time"<br>0: local time<br>1: UTC |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | YEAR[7:0] | | | | | | | |
| RESP5 | YEAR[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP6 | MONTHS[7:0] | | | | | | | |
| RESP7 | DAYS[7:0] | | | | | | | |
| RESP8 | HOURS[7:0] | | | | | | | |
| RESP9 | MINUTES[7:0] | | | | | | | |
| RESP10 | SECONDS[7:0] | | | | | | | |

| Name | Function |
|---|---|
| YEAR[15:0] | The year as reported by the ensemble. |
| MONTHS[7:0] | The month as reported by the ensemble. |
| DAYS[7:0] | The current day reported by the ensemble. |
| HOURS[7:0] | The current hour reported by the ensemble. |
| MINUTES[7:0] | The current minute reported by the ensemble. |
| SECONDS[7:0] | The current second reported by the ensemble. |

### Command 0xBD. DAB_GET_AUDIO_INFO

DAB_GET_AUDIO_INFO gets information about the current audio service (decoder bps, audio mode). The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xBD | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | AUDIO_BIT_RATE[7:0] | | | | | | | |
| RESP5 | AUDIO_BIT_RATE[15:8] | | | | | | | |
| RESP6 | AUDIO_SAMPLE_RATE[7:0] | | | | | | | |
| RESP7 | AUDIO_SAMPLE_RATE[15:8] | | | | | | | |
| RESP8 | X | X | X | X | AUDIO_PS_-FLAG | AUDIO_SBR_-FLAG | AUDIO_MODE[1:0] | |

| Name | Function |
|---|---|
| AUDIO_BIT_RATE[15:0] | Audio bit rate of the current audio service (kbps). |
| AUDIO_SAM-PLE_RATE[15:0] | Sample rate of the current audio service (Hz). |
| AUDIO_PS_FLAG | Audio PS flag. only applicable to DAB+. Set to 0 for DAB<br>0: SBR is not used<br>1: SBR is used |
| AUDIO_SBR_FLAG | Audio SBR flag. only applicable to DAB+. Set to 0 for DAB<br>0: SBR is not used<br>1: SBR is used |
| AUDIO_MODE[1:0] | Audio mode<br>0: dual<br>1: mono<br>2: stereo<br>3: joint stereo |

### Command 0xBE. DAB_GET_SUBCHAN_INFO

DAB_GET_SUBCHAN_INFO gets information about the sub-channel (service mode, protection, subchanel bps). The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xBE | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICEID[7:0] | | | | | | | |
| ARG5 | SERVICEID[15:8] | | | | | | | |
| ARG6 | SERVICEID[23:16] | | | | | | | |
| ARG7 | SERVICEID[31:24] | | | | | | | |
| ARG8 | COMPID[7:0] | | | | | | | |
| ARG9 | COMPID[15:8] | | | | | | | |
| ARG10 | COMPID[23:16] | | | | | | | |
| ARG11 | COMPID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERVICEID[31:0] | The service ID |
| COMPID[31:0] | The component ID. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | SERVICE_MODE[7:0] | | | | | | | |
| RESP5 | PROTECTION_INFO[7:0] | | | | | | | |
| RESP6 | SUBCHAN_BIT_RATE[7:0] | | | | | | | |
| RESP7 | SUBCHAN_BIT_RATE[15:8] | | | | | | | |
| RESP8 | NUM_CU[7:0] | | | | | | | |
| RESP9 | NUM_CU[15:8] | | | | | | | |
| RESP10 | CU_ADDRESS[7:0] | | | | | | | |
| RESP11 | CU_ADDRESS[15:8] | | | | | | | |

| Name | Function |
|---|---|
| SERVICE_MODE[7:0] | Indicates the service mode of the sub-channel.<br>0: AUDIO STREAM SERVICE<br>1: DATA STREAM SERVICE<br>2: FIDC SERVICE<br>3: MSC DATA PACKET SERVICE<br>4: DAB+<br>5: DAB<br>6: FIC SERVICE<br>7: XPAD DATA<br>8: NO MEDIA |
| PROTEC-TION_INFO[7:0] | Indicates the protection profile of the sub-channel. Two error protection profiles can be employed: Unequal Error Protection (UEP) and Equal Error Protection (EEP). For each profile, the protection level indicates the associated convolutional coding rate. Please refer to ETSI EN 300 401 V1.4.1 clause 6.2.1.<br>1: Protection mode is UEP. Protection level is 1<br>2: Protection mode is UEP. Protection level is 2<br>3: Protection mode is UEP. Protection level is 3<br>4: Protection mode is UEP. Protection level is 4<br>5: Protection mode is UEP. Protection level is 5<br>6: Protection mode is EEP. Protection level is A-1<br>7: Protection mode is EEP. Protection level is A-2<br>8: Protection mode is EEP. Protection level is A-3<br>9: Protection mode is EEP. Protection level is A-4<br>10: Protection mode is EEP. Protection level is B-1<br>11: Protection mode is EEP. Protection level is B-2<br>12: Protection mode is EEP. Protection level is B-3<br>13: Protection mode is EEP. Protection level is B-4 |

| Name | Function |
|---|---|
| SUB-CHAN_BIT_RATE[15:0] | Sub-channel bit rate (kpbs). |
| NUM_CU[15:0] | The number of Capacity units assigned to this service component. |
| CU_ADDRESS[15:0] | The CU starting address of this subchannel within the CIF. |

### Command 0xBF. DAB_GET_FREQ_INFO

DAB_GET_FREQ_INFO gets radio Frequency Information (FI) about the ensemble. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xBF | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | LENGTH_FI_LIST[7:0] | | | | | | | |
| RESP5 | LENGTH_FI_LIST[15:8] | | | | | | | |
| RESP6 | LENGTH_FI_LIST[23:16] | | | | | | | |
| RESP7 | LENGTH_FI_LIST[31:24] | | | | | | | |
| RESP8 | FI_ID_0[7:0] | | | | | | | |
| RESP9 | FI_ID_0[15:8] | | | | | | | |
| RESP10 | FI_ID_0[23:16] | | | | | | | |
| RESP11 | FI_ID_0[31:24] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP12 | FI_FREQ_0[7:0] | | | | | | | |
| RESP13 | FI_FREQ_0[15:8] | | | | | | | |
| RESP14 | FI_FREQ_0[23:16] | | | | | | | |
| RESP15 | FI_FREQ_0[31:24] | | | | | | | |
| RESP16 | FI_FREQ_INDEX_0[7:0] | | | | | | | |
| RESP17 | FI_RNM_0[7:0] | | | | | | | |
| RESP18 | FI_CONTINUITY_0[7:0] | | | | | | | |
| RESP19 | FI_CONTROL_0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH_FI_LIST[31:0] | The length of frequency information (FI) list. |
| FI_ID_0[31:0] | This is the first ID in the frequency information list. The FI_ID_0 field depends on the FI_RNM_0 field. Please see clause 8.1.8 of ETSI 300-401 for further details. This field is repeated for each frequency information. |
| FI_FREQ_0[31:0] | This is the first frequency (unit: Hz) in the frequency information list. This field is repeated for each frequency information. |
| FI_FREQ_INDEX_0[7:0] | This is the index of FI_FREQ_0 in the current frequency table defined through DAB_SET_FREQ_LIST. The index starts from 0, and 0xFF represents non-existent frequency. This field is repeated for each frequency information. |
| FI_RNM_0[7:0] | This is the first Range and Modulation (RNM) field in the frequency information list. Please see clause 8.1.8 of ETSI 300-401 for further details. This field is repeated for each frequency information. |
| FI_CONTINUITY_0[7:0] | This is the first Continuity flag field in the frequency information list. This flag depends on the RNM field. Please see clause 8.1.8 of ETSI 300-401 for further details. This field is repeated for each frequency information. |
| FI_CONTROL_0[7:0] | This is the first Control field in the frequency information list. This field only applies when RNM = 0x00 or 0x01 (DAB ensemble). Set to 0 for other cases. Please see clause 8.1.8 of ETSI 300-401 for further details. This field is repeated for each frequency information. |

### Command 0xE5. TEST_GET_RSSI

TEST_GET_RSSI returns the reported RSSI in 8.8 format. This command is used to help calibrate the frontend tracking circuit. It returns the RSSI value in dBuV to 1/256 of a dB. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: TEST_GET_RSSI command is not supported in firmware revision A0A or A0B.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xE5 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | RSSI_HIGHRES[7:0] | | | | | | | |
| RESP5 | RSSI_HIGHRES[15:8] | | | | | | | |

| Name | Function |
|---|---|
| RSSI_HIGHRES[15:0] | The RSSI measurement in dBuV to 1/256 of a dB. |

### Command 0xE8. DAB_TEST_GET_BER_INFO

DAB_TEST_GET_BER_INFO reads the current BER rate using debug information that was sent to the test port. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Please refer to section (BER Test Setup and Procedure) of this document for more details on BER measurement.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xE8 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | X | X | X | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | ERR_BITS[7:0] | | | | | | | |
| RESP5 | ERR_BITS[15:8] | | | | | | | |
| RESP6 | ERR_BITS[23:16] | | | | | | | |
| RESP7 | ERR_BITS[31:24] | | | | | | | |
| RESP8 | TOTAL_BITS[7:0] | | | | | | | |
| RESP9 | TOTAL_BITS[15:8] | | | | | | | |
| RESP10 | TOTAL_BITS[23:16] | | | | | | | |
| RESP11 | TOTAL_BITS[31:24] | | | | | | | |

| Name | Function |
|---|---|
| ERR_BITS[31:0] | The number of bits received with errors. |
| TOTAL_BITS[31:0] | The total number of bits received. |

## 4.4. Properties Summary—DAB

**Table 6. Properties Summary—DAB**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x0000 | INT_CTL_ENABLE | Interrupt enable property | 0x0000 |
| 0x0001 | INT_CTL_REPEAT | Interrupt repeat property | 0x0000 |
| 0x0200 | DIGITAL_IO_OUTPUT_SELECT | Selects digital audio Master or Slave. | 0 |
| 0x0201 | DIGITAL_IO_OUTPUT_SAMPLE_RATE | Sets output sample audio rate in units of 1Hz. | 48000 |
| 0x0202 | DIGITAL_IO_OUTPUT_FORMAT | Configure digital output format. | 0x1800 |
| 0x0203 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 | Deviations from the standard framing mode | 0 |
| 0x0204 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 | Deviations from the standard framing mode | 0 |
| 0x0205 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 | Deviations from the standard framing mode | 0 |
| 0x0206 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 | Deviations from the standard framing mode | 0 |
| 0x0300 | AUDIO_ANALOG_VOLUME | Sets the audio analog volume. | 63 |
| 0x0301 | AUDIO_MUTE | AUDIO_MUTE property mutes/unmutes each audio output independently. | 0x0000 |
| 0x0800 | PIN_CONFIG_ENABLE | Pin configuration property | 0x8001 |
| 0x1710 | DAB_TUNE_FE_VARM | DAB/DMB Front End Varactor configuration slope | 0 |
| 0x1711 | DAB_TUNE_FE_VARB | DAB/DMB Front End Varactor configuration intercept | 0 |
| 0x1712 | DAB_TUNE_FE_CFG | Additional configuration options for the front end. | 0x0001 |
| 0x8100 | DIGITAL_SERVICE_INT_SOURCE | Configures the interrupt sources for digital services | 0x0000 |
| 0x8101 | DIGITAL_SERVICE_RESTART_DELAY | sets the delay time (in miliseconds) to restart digital service when recovering from acquisition loss | 200 |
| 0xB000 | DAB_DIGRAD_INTERRUPT_SOURCE | Configures interrupts related to digital receiver. | 0 |
| 0xB001 | DAB_DIGRAD_RSSI_HIGH_THRESHOLD | DAB_DIGRAD_RSSI_HIGH_-THRESHOLD sets the high threshold, which triggers the DIGRAD interrupt if the RSSI is above this threshold. | 127 |
| 0xB002 | DAB_DIGRAD_RSSI_LOW_THRESHOLD | DAB_DIGRAD_RSSI_LOW_THRESHOLD sets the low threshold, which triggers the DIGRAD interrupt if the RSSI is below this threshold. | –128 |

**Table 6. Properties Summary—DAB (Continued)**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0xB200 | DAB_VALID_RSSI_TIME | Sets the time to allow the RSSI metric to settle before evaluating. | 30 |
| 0xB201 | DAB_VALID_RSSI_THRESHOLD | Sets the RSSI threshold for a valid DAB Seek/Tune | 12 |
| 0xB202 | DAB_VALID_ACQ_TIME | Set the time to wait for acquisition before evaluating | 1200 |
| 0xB203 | DAB_VALID_SYNC_TIME | Sets the time to wait for ensemble synchronization. | 450 |
| 0xB300 | DAB_EVENT_INTERRUPT_SOURCE | Configures which dab events will set the DEVENTINT status bit. | 0 |
| 0xB301 | DAB_EVENT_MIN_SVRLIST_PERIOD | Configures how often service list notifications can occur. | 10 |
| 0xB302 | DAB_EVENT_MIN_SVRLIST_PERIOD_RECONFIG | Configures how often service list notifications can occur during reconfiguration events. | 10 |
| 0xB303 | DAB_EVENT_MIN_FREQINFO_PERIOD | Configures how often frequency information notifications can occur. | 5 |
| 0xB400 | DAB_XPAD_ENABLE | Selects which XPAD data will forwarded to the host. | 1 |
| 0xB500 | DAB_CTRL_DAB_MUTE_ENABLE | enables the feature of hard muting audio when signal level is low | 1 |
| 0xB501 | DAB_CTRL_DAB_MUTE_SIGNAL_LEVEL_THRESHOLD | Set the signal quality threshold, which is evaluated by fic_quality. If fic_quality is below this threshold for a time period (indicated by DAB_MUTE_WIN_THRESHOLD), audio will be hard muted. If fic_quality is above this threshold for a time period (indicated by DAB_UNMUTE_WIN_THRESHOLD), audio will be unmuted. | 98 |
| 0xB502 | DAB_CTRL_DAB_MUTE_WIN_THRESHOLD | Set the time window threshold (in miliseconds) to mute audio | 1000 |
| 0xB503 | DAB_CTRL_DAB_UNMUTE_WIN_THRESHOLD | Set the time window threshold (in miliseconds) to unmute audio | 1500 |
| 0xB504 | DAB_CTRL_DAB_MUTE_SIGLOSS_THRESHOLD | Set the signal RSSI threshold to mute audio. RSSI below this threshold indicates that signal is lost. In this case, audio will be muted. | 6 |
| 0xB505 | DAB_CTRL_DAB_MUTE_SIGLOW_THRESHOLD | Set the signal SNR threshold. The fic_quality based audio mute operation only engages when signal SNR is below this threshold. | 9 |
| 0xE800 | DAB_TEST_BER_CONFIG | Sets up and enables the DAB BER test | 0 |

### 4.4.1. DAB Properties

**Property 0x0000. INT_CTL_ENABLE**

INT_CTL_ENABLE property enables top-level interrupt sources. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTIEN | Reserved | | | | | CTSIEN | ERR_CMDIEN | DACQIEN | DSRVIEN | Reserved | | | STCIEN |
| Default | 0x0 | | 0 | 0x00 | | | | | 0 | 0 | 0 | 0 | 0x0 | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTIEN | Interrupt when DEVNTINT is set |
| 12:8 | Reserved | Always write to 0x00. |
| 7 | CTSIEN | Interrupt when CTS is set.<br>Note: The default will be 0x1 if POWER_UP:CTSIEN was set. |
| 6 | ERR_CMDIEN | Interrupt when ERR_CMD is set |
| 5 | DACQIEN | Interrupt when DIGINT is set |
| 4 | DSRVIEN | Interrupt when DSRVINT is set |
| 3:1 | Reserved | Always write to 0x0. |
| 0 | STCIEN | Interrupt when STCINT is set |

**Property 0x0001. INT_CTL_REPEAT**

INT_CTL_REPEAT is used to set repeat interrupt pulses for a given interrupt even if this particular interrupt was previously generated but not acknowledged. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTREP | Reserved | | | | | | | DACQREP | DSRVREP | Reserved | | | STCREP |
| Default | 0x0 | | 0 | 0x00 | | | | | | | 0 | 0 | 0x0 | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTREP | Repeat interrupt pulse when DEVNTINT is set, even if a previous interrupt was generated but not acknowledged. |
| 12:6 | Reserved | Always write to 0x00. |
| 5 | DACQREP | Repeat interrupt pulse when DACQINT is set, even if a previous interrupt was generated but not acknowledged. |
| 4 | DSRVREP | Repeat interrupt pulse when DSRVINT is set, even if a previous interrupt was generated but not acknowledged. |
| 3:1 | Reserved | Always write to 0x0. |
| 0 | STCREP | Repeat interrupt pulse when STCINT is set, even if a previous interrupt was generated but not acknowledged. |

### Property 0x0200. DIGITAL_IO_OUTPUT_SELECT

DIGITAL_IO_OUTPUT_SELECT configures the digital audio output to be I2S Master or Slave. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MASTER | | | | | | | | | | | | | | | |
| Default | 0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | MASTER | 0: Slave mode selected<br>1: Master mode selected |

### Property 0x0201. DIGITAL_IO_OUTPUT_SAMPLE_RATE

DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 48000

Units: Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | OUTPUT_SAMPLE_RATE[15:0] | | | | | | | | | | | | | | | |
| Default | 0xBB80 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | OUTPUT_SAMPLE_RATE[15:0] | DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz.<br>Range: 32000-48000<br>32000: The minimum output sample rate in Hz.<br>48000: The minimum output sample rate in Hz. |

### Property 0x0202. DIGITAL_IO_OUTPUT_FORMAT

DIGITAL_IO_OUTPUT_FORMAT configures the digital audio output format. This property may only be written before the first tune. Writes after first tune will return error. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x1800

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | SAMPL_SIZE[5:0] | | | | | | SLOT_SIZE[3:0] | | | | FRAMING_FORMAT[3:0] | | | |
| Default | 0x0 | | 0x18 | | | | | | 0x0 | | | | 0x0 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:8 | SAMPL_SIZE[5:0] | Determines the number of bits in a sample. Only the specified number of bits per sample are used. The value of any bits sent over the sample size will be 0.<br>Range: 2-24 |
| 7:4 | SLOT_SIZE[3:0] | Determines slot size; must be larger than or equal to SAMPL_SIZE.<br>0x0: Matches Sample Size<br>0x2: 8 bits<br>0x4: 16 bits<br>0x5: 20 bits<br>0x6: 24 bits<br>0x7: 32 bits |
| 3:0 | FRAMING_FORMAT[3:0] | Determines when the data is transmitted relative to frame sync.<br>0x0: I2S mode<br>0x6: DSP mode<br>0x7: Left justified dsp mode<br>0x8: Left justified mode<br>0x9: Right justified mode |

### Property 0x0203. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | FSLATE_EN | FSINV_EN | RJUST_EN | CLK-INV_EN | SWAP_EN | BITORDER_EN | Reserved | | FSLATE | FSINV | RJUST | CLK-INV | SWAP | BITORDER |
| Default | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | FSLATE_EN | Enable the FSLATE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 12 | FSINV_EN | Enable the FSINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 11 | RJUST_EN | Enable the RJUST override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 10 | CLKINV_EN | Enable the CLKINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 9 | SWAP_EN | Enable the SWAP override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | BITORDER_EN | Enable the BITORDER override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | Reserved | Always write to 0x0. |
| 5 | FSLATE | 0: Sample data on the rising edge of the 2nd clock after DFS<br>1: Sample data on the rising edge of the 1st clock after DFS |
| 4 | FSINV | 0: Use the DFS signal as is<br>1: Invert the DFS signal |
| 3 | RJUST | 0: Left Justified<br>1: Right Justified |

| Bit | Name | Function |
|---|---|---|
| 2 | CLKINV | Inverts the data clock<br>0: The bit clock is not inverted. DFS will be captured on rising edge of DCLK.<br>1: The bit clock is inverted. DFS will be captured on falling edge of DCLK. |
| 1 | SWAP | Swap position of the left and right channels.<br>0: Transmit the left sample first<br>1: Transmit the right sample first |
| 0 | BITORDER | Determine if the MSB or LSB is transmitted first.<br>0: Transmit MSB first<br>1: Transmit LSB first |

### Property 0x0204. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | FILL_EN | Reserved | | | | SEQEN_EN | FSEDGE_EN | FILL[1:0] | | Reserved | | | | SEQEN | FSEDGE |
| Default | 0 | 0 | 0x0 | | | | 0 | 0 | 0x0 | | 0x0 | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15 | Reserved | Always write to 0. |
| 14 | FILL_EN | Enable the FILL override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 13:10 | Reserved | Always write to 0x0. |
| 9 | SEQEN_EN | Enable the SEQEN override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | FSEDGE_EN | Enable the FSEDGE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | FILL[1:0] | Defines the fill value for unused data bits.<br>0: The unused bits are filled with 0's<br>1: The unused bits are filled with 1's<br>2: The unused bits are sign extended<br>3: The unused bits are filled with a random sequence |
| 5:2 | Reserved | Always write to 0x0. |

| Bit | Name | Function |
|---|---|---|
| 1 | SEQEN | 0: Only one mono sample is sent per trigger of DFS<br>1: Both mono samples are sent sequentially after each trigger from DFS |
| 0 | FSEDGE | 0: One mono sample is sent on only one edge of DFS<br>1: One mono sample is sent every edge (rising and falling) of DFS |

### Property 0x0205. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSH_EN | FSH[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSH_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSH[14:0] | Number of samples FS will remain high.<br>Range: 0-32767 |

### Property 0x0206. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSL_EN | FSL[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSL_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSL[14:0] | Number of samples FS will remain low.<br>Range: 0-32767 |

### Property 0x0300. AUDIO_ANALOG_VOLUME

AUDIO_ANALOG_VOLUME sets the analog audio volume. A value of 0 will mute the audio; a value of 1 applies 62 dB of attenuation, and a value of 63 applies no attenuation. Each step accounts for 1 dB of change in the output. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 63

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | VOL[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x3F | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | VOL[5:0] | Controls the analog volume level.<br>Range: 0-63<br>0: The minimum audio level.<br>63: The maximum audio level. |

### Property 0x0301. AUDIO_MUTE

AUDIO_MUTE property mutes/unmutes each audio output independently. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | MUTE[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x0 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | MUTE[1:0] | Controls the audio mute on each audio output<br>0: Do not mute audio outputs<br>1: Mute Left Audio Out.<br>2: Mute Right Audio Out.<br>3: Mute both Left and Right Audio Out |

### Property 0x0800. PIN_CONFIG_ENABLE

PIN_CONFIG is used to enable and disable the various I/O features of the device. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x8001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | INT-BOUTEN | UARTEN | Reserved | | | | | | | | | | | | I2SOUTEN | DACOUTEN |
| Default | 1 | 0 | 0x000 | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15 | INTBOUTEN | When set enables the INTB output. |
| 14 | UARTEN | When set enables the UART I/O. |
| 13:2 | Reserved | Always write to 0x000. |
| 1 | I2SOUTEN | When set enables the I2S the digital audio output. A change to this bit will take effect on the next tune or seek. |
| 0 | DACOUTEN | When set enables the analog audio output. A change to this bit will take effect on the next tune or seek. |

### Property 0x1710. DAB_TUNE_FE_VARM

DAB_TUNE_FE_VARM DAB/DMB Front End Varactor configuration slope (x 1000) which has been calculated for a particular board design. Both DAB_TUNE_FE_VARB and DAB_TUNE_FE_VARM must be configured. This takes effect upon DAB_TUNE_FREQ. This is a signed value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FE_VARM[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FE_VARM[15:0] | DAB/DMB Front End Varactor configuration slope (x 1000) which has been calculated for a particular board design. Both DAB_TUNE_FE_VARB and DAB_TUNE_FE_VARM must be configured. This takes effect upon DAB_TUNE_FREQ. This is a signed value. |

### Property 0x1711. DAB_TUNE_FE_VARB

DAB_TUNE_FE_VARB DAB/DMB Front End Varactor configuration intercept which has been calculated for a particular board design. Both DAB_TUNE_FE_VARB and DAB_TUNE_FE_VARM must be configured. This takes effect upon DAB_TUNE_FREQ. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FE_VARB[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FE_VARB[15:0] | DAB/DMB Front End Varactor configuration intercept which has been calculated for a particular board design. Both DAB_TUNE_FE_VARB and DAB_TUNE_FE_VARM must be configured. This takes effect upon DAB_TUNE_FREQ. |

### Property 0x1712. DAB_TUNE_FE_CFG

Additional configuration options for the front end. These take effect upon DAB_TUNE_FREQ. See DAB_TUNE_FE_VARB and DAB_TUNE_FE_VARM The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | VHF CAPS | VHF SW |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | VHFCAPS | Configure Varactor matching for changes in the AGC's capacitive attenuators<br>0: Varactor will compensate for changes in the CATT<br>1: Varactor is decoupled from the CATT |
| 0 | VHFSW | VHFSW sets the open or closed state for the front end switch.<br>0: Switch Open<br>1: Switch Closed |

**Property 0x8100. DIGITAL_SERVICE_INT_SOURCE**

DIGITAL_SERVICE_INT_SOURCE configures which digital service events will set the DSRVINT status bit. When one of the bits decribed below is set, the corresponding event will cause the DSRVINT bit of the status word to be set. To clear the DSRVINT bit the GET_DIGITAL_SERVICE_DATA command must be executed with the ack option. If the DSRVIEN interrupt is enabled, a host interrupt will also be generated. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | DSRVOVFLINT | DSRVPCKTINT |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | DSRVOVFLINT | Configures the data service overflow interrupt. When this interrupt occurs data service data has been lost and indicates that the host is not retrieving the data service data in a timely fashion.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 0 | DSRVPCKTINT | Enables the DSRVPCKTINT interrupt of the GET_DIGITAL_SERVICE_DATA command.<br>0: Interrupt disabled<br>1: Interrupt enabled |

### Property 0x8101. DIGITAL_SERVICE_RESTART_DELAY

DIGITAL_SERVICE_RESTART_DELAY sets the delay time (in miliseconds) to restart digital service. When the system recovers from an acquisition loss, the service that had previously been started will be restarted after this delay. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 200

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RESTART_DELAY[15:0] | | | | | | | | | | | | | | | |
| Default | 0x00C8 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RESTART_DE-LAY[15:0] | DIGITAL_SERVICE_RESTART_DELAY sets the delay time (in miliseconds) to restart digital service. When the system recovers from an acquisition loss, the service that had previously been started will be restarted after this delay.<br>Range: 100-65535 |

### Property 0xB000. DAB_DIGRAD_INTERRUPT_SOURCE

DAB_DIGRAD_INERRUPT_SOURCE configures interrupts related to digital receiver (DAB_DIGRAD_STATUS). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | HARDMUTEIEN | FICERRIEN | ACQIEN | RSSIHIEN | RSSILIEN |
| Default | 0x000 | | | | | | | | | | | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4 | HARDMUTEIEN | Hard mute state change interrupt enable.<br>0: Disable the hard mute interrupt.<br>1: Enable interrupt to occur whenever the audio is muted. |
| 3 | FICERRIEN | Ensemble FIC error interrupt enable.<br>0: Disable the FIC error interrupt.<br>1: Enable interrupt to occur whenever a new FIC error is detected changes. |
| 2 | ACQIEN | Acquisition state change interrupt enable.<br>0: Disable the acquisition change interrupt.<br>1: Enable interrupt to occur whenever the acquisition state changes. |
| 1 | RSSIHIEN | RSSI above DAB_DIGRAD_RSSI_HIGH_THRESHOLD interrupt enable.<br>0: Disable the RSSI above DAB_DIGRAD_RSSI_HIGH_THRESHOLD interrupt.<br>1: Enable interrupt to occur whenever RSSI is above DAB_DIGRAD_RSSI_HIGH_-THRESHOLD interrupt. |
| 0 | RSSILIEN | RSSI below DAB_DIGRAD_RSSI_LOW_THRESHOLD interrupt enable.<br>0: Disable the RSSI below DAB_DIGRAD_RSSI_LOW_THRESHOLD interrupt.<br>1: Enable interrupt to occur whenever RSSI is below DAB_DIGRAD_RSSI_LOW_-THRESHOLD interrupt. |

**Property 0xB001. DAB_DIGRAD_RSSI_HIGH_THRESHOLD**

DAB_DIGRAD_RSSI_HIGH_THRESHOLD sets the high threshold, which triggers the DIGRAD interrupt if the RSSI is above this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | HIGH_THRESHOLD[7:0] | | | | | | | |
| **Default** | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HIGH_THRESH-OLD[7:0] | Rssi Offset dB.<br>Range: -128-127—sets sets the high threshold, which triggers the RSSIH interrupt if the RSSI is above this threshold. |

### Property 0xB002. DAB_DIGRAD_RSSI_LOW_THRESHOLD

DAB_DIGRAD_RSSI_LOW_THRESHOLD sets the low threshold, which triggers the DIGRAD interrupt if the RSSI is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -128

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | LOW_THRESHOLD[7:0] | | | | | | | |
| Default | 0xFF | | | | | | | | 0x80 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0xFF. |
| 7:0 | LOW_THRESH-OLD[7:0] | Rssi Offset dB.<br>Range: -128-127—sets sets the low threshold, which triggers the RSSIL interrupt if the RSSI is below this threshold. |

### Property 0xB200. DAB_VALID_RSSI_TIME

DAB_VALID_RSSI_TIME sets the time in ms to allow the RSSI metric to settle before evaluating its validity during tune. If RSSI does not exceed DAB_VALID_RSSI_THRESHOLD by this time the tune will be aborted, and STC will be set, and the tune will be flaged as invalid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 30

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | MS[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x1E | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | MS[5:0] | validation time in ms. If set to 0, then RSSI is not considered when determiniting a channel's validity.<br>Range: 0-63 |

### Property 0xB201. DAB_VALID_RSSI_THRESHOLD

DAB_VALID_RSSI_THRESHOLD sets the RSSI threshold for a valid DAB Seek/Tune. If the desired channel RSSI is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 12

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | LEVEL[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x0C | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | LEVEL[7:0] | Range: -128-127 |

### Property 0xB202. DAB_VALID_ACQ_TIME

DAB_VALID_ACQ_TIME sets the time in ms to wait for acquisition before evaluating acquisition validity during tune. If system has not fully acquired by this time the tune will be aborted, STC will be set, and the tune will be flaged as invalid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1200

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | MS[10:0] | | | | | | | | | | |
| Default | 0x00 | | | | | 0x4B0 | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:11 | Reserved | Always write to 0x00. |
| 10:0 | MS[10:0] | validation time in ms. If 0, then acquisition is not considered when determiniting a channel's validity.<br>Range: 0-2047 |

### Property 0xB203. DAB_VALID_SYNC_TIME

DAB_VALID_SYNC_TIME sets the time in ms to wait for synchronization during tune. If the system has not synchronized by this time the tune will be aborted, STC will be set, and the tune will be flaged as invalid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 450

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | MS[10:0] | | | | | | | | | | |
| Default | 0x00 | | | | | 0x1C2 | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:11 | Reserved | Always write to 0x00. |
| 10:0 | MS[10:0] | validation time in ms. If set to 0, then synchronization is not considered when determiniting a channel's validity.<br>Range: 0-2047 |

### Property 0xB300. DAB_EVENT_INTERRUPT_SOURCE

DAB_EVENT_INTERRUPT_SOURCE configures which dab events will set the DEVENTINT status bit. When one of the bits described below is set, the corresponding event will cause the DEVENTINT bit of the status word will be set. To clear the DEVENTINT bit in the status word the DAB_GET_EVENT_STATUS command must be executed with EVENT_ACK set. If the DEVENTINT interrupt is enabled, a host interrupt will also be generated. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | RECFG_INTEN | RECFGWRN_INTEN | Reserved | | | | FREQINFO_INTEN | SRVLIST_INTEN |
| Default | 0x00 | | | | | | | | 0 | 0 | 0x0 | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | RECFG_INTEN | Enables the ensemble reconfiguration event interrupt. This interrupt indicates that the ensemble reconfiguration has occurred.<br>0: Disables the ensemble reconfiguration interrupt.<br>1: Enable the interrupt to occur whenever the ensemble reconfiguration occurs. |
| 6 | RECFG-WRN_INTEN | Enables the ensemble reconfiguration warning event interrupt. This interrupt indicates that an ensemble reconfiguration will occur in 6 seconds. All following service list updates pertain to the new ensemble configuration.<br>0: Disables the ensemble reconfiguration warning interrupt.<br>1: Enable the interrupt to occur whenever the ensemble reconfiguration warning occurs. |
| 5:2 | Reserved | Always write to 0x0. |
| 1 | FREQINFO_INTEN | Enables the frequency information event interrupt.<br>0: Disables the frequency information update interrupt.<br>1: Enable interrupt to occur whenever a frequency information update occurs. |
| 0 | SRVLIST_INTEN | Enables the digital service list event interrupt.<br>0: Disables the service list update interrupt.<br>1: Enable interrupt to occur whenever a service list update occurs. |

**Property 0xB301. DAB_EVENT_MIN_SVRLIST_PERIOD**

DAB_EVENT_MIN_SVRLIST_PERIOD configures how often service list notifications can occur in units of 100ms. This property is used to reduce the number of service list update notifications received at initial tune when the service list is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Note that the service list is a dynamic object and it will be updated periodically. Therefore it is good practice to update the host's service data base whenever tuning to an ensemble. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: 100ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | MIN_SVRLIST_PERIOD[15:0] | | | | | | | | | | | | | | | |
| **Default** | 0x000A | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | MIN_SVRLIST_PERIOD[15:0] | DAB_EVENT_MIN_SVRLIST_PERIOD configures how often service list notifications can occur in units of 100ms. This property is used to reduce the number of service list update notifications received at initial tune when the service list is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Note that the service list is a dynamic object and it will be updated periodically. Therefore it is good practice to update the host's service data base whenever tuning to an ensemble. |

**Property 0xB302. DAB_EVENT_MIN_SVRLIST_PERIOD_RECONFIG**

DAB_EVENT_MIN_SVRLIST_PERIOD_RECONFIG configures how often service list notifications can occur in units of 100ms during reconfiguration. This property is used to reduce the number of service list update notifications received at initial tune when the service list is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Setting this to a smaller period than DAB_EVENT_MIN_SVRLIST_PERIOD will reduce the amount of time required to update the service list during a reconfiguration event. Setting this to a larger period than DAB_EVENT_MIN_SVRLIST_PERIOD will have no effect. Note that the service list is a dynamic object and it will be updated periodically. Therefore it is good practice to update the host's service data base whenever tuning to an ensemble. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: 100ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MIN_SVRLIST_PERIOD_RECONFIG[15:0] | | | | | | | | | | | | | | | |
| Default | 0x000A | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | MIN_SVRLIST_PERIOD_RECONFIG[15:0] | DAB_EVENT_MIN_SVRLIST_PERIOD_RECONFIG configures how often service list notifications can occur in units of 100ms during reconfiguration. This property is used to reduce the number of service list update notifications received at initial tune when the service list is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Setting this to a smaller period than DAB_EVENT_MIN_SVRLIST_PERIOD will reduce the amount of time required to update the service list during a reconfiguration event. Setting this to a larger period than DAB_EVENT_MIN_SVRLIST_PERIOD will have no effect. Note that the service list is a dynamic object and it will be updated periodically. Therefore it is good practice to update the host's service data base whenever tuning to an ensemble. |

### Property 0xB303. DAB_EVENT_MIN_FREQINFO_PERIOD

DAB_EVENT_MIN_FREQINFO_PERIOD configures how often frequency info notifications can occur in units of 100ms. This property is used to reduce the number of frequency info update notifications received at initial tune when frequency info is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Note that frequency info is dynamic and it may be updated periodically. Therefore it is good practice to update the host's frequency info data base whenever tuning to an ensemble. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 5

Units: 100ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MIN_FREQINFO_PERIOD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0005 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | MIN_FREQINFO_PERIOD[15:0] | DAB_EVENT_MIN_FREQINFO_PERIOD configures how often frequency info notifications can occur in units of 100ms. This property is used to reduce the number of frequency info update notifications received at initial tune when frequency info is updated very frequently. With this property a trade off can be made between getting partial information quickly or waiting longer and getting a complete list. Note that frequency info is dynamic and it may be updated periodically. Therefore it is good practice to update the host's frequency info data base whenever tuning to an ensemble. |

### Property 0xB400. DAB_XPAD_ENABLE

DAB_PAD_ENABLE selects which PAD application data will be forwarded to the host when available. When an audio service is playing one of it's audio components this property is used to select which PAD services are forwarded to the host for decoding. Note that not all audio services will carry all XPAD applications so only those that are available and enabled will be forwarded. To determine which PAD services are provided use the DAB_GET_COMPONENT_INFO command. This property can be set at any time. DLS is always provided. The host must have the proper user application(s) and decoders to use this feature such as MOT Slide Show or BWS. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | TDC_ENABLE | MOT_ENABLE | DLS_ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | 0 | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2 | TDC_ENABLE | Enables XPAD delivered TDC data. |
| 1 | MOT_ENABLE | Enables XPAD delivered MOT objects. |
| 0 | DLS_ENABLE | Enables PAD delivered DLS packets. |

### Property 0xB500. DAB_CTRL_DAB_MUTE_ENABLE

DAB_MUTE_ENABLE enables the feature of hard muting audio when signal level is low. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

Units: value

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | MS |
| Default | 0x0000 | | | | | | | | | | | | | | | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | MS | set 1 to enable hard mute feature when signal level is low.<br>Range: 0-1 |

### Property 0xB501. DAB_CTRL_DAB_MUTE_SIGNAL_LEVEL_THRESHOLD

DAB_MUTE_QUALITY_THRESHOLD set the threshold to mute audio when signal level is low. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 98

Units: %

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DAB_MUTE_SIGNAL_LEVEL_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0062 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | DAB_MUTE_SIG-NAL_LEVEL_-THRESHOLD[15:0] | DAB_MUTE_QUALITY_THRESHOLD set the threshold to mute audio when signal level is low.<br>If signal level is lower than this threshold for a accumulated period of time, audio will be muted.<br>Range: 0-100 |

### Property 0xB502. DAB_CTRL_DAB_MUTE_WIN_THRESHOLD

DAB_MUTE_WIN_THRESHOLD set the threshold to mute audio. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1000

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DAB_MUTE_WIN_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x03E8 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | DAB_MUTE_WIN_-THRESHOLD[15:0] | DAB_MUTE_WIN_THRESHOLD set the threshold to mute audio.<br>If signal level is continuously low for a time period longer than this threshold, audio will be hardmuted. |

### Property 0xB503. DAB_CTRL_DAB_UNMUTE_WIN_THRESHOLD

DAB_UNMUTE_WIN_THRESHOLD set the threshold to unmute audio. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1500

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DAB_UNMUTE_WIN_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x05DC | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | DAB_UN-MUTE_WIN_-THRESHOLD[15:0] | DAB_UNMUTE_WIN_THRESHOLD set the threshold to unmute audio.<br>If signal level is continuously high for a time period longer than this threshold, audio will be back. |

### Property 0xB504. DAB_CTRL_DAB_MUTE_SIGLOSS_THRESHOLD

DAB_MUTE_SIGLOSS_THRESHOLD set the threshold to mute audio when signal is loss. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 6

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DAB_MUTE_SIGLOSS_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0006 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | DAB_MUTE_SIGLOSS_THRESHOLD[15:0] | DAB_MUTE_SIGLOSS_THRESHOLD set the threshold to mute audio when signal is loss.<br>If signal is lost, which means RSSI is less than this threshold, audio will be muted. |

### Property 0xB505. DAB_CTRL_DAB_MUTE_SIGLOW_THRESHOLD

DAB_MUTE_SIGLOW_THRESHOLD set the SNR threshold. The fic_quality based audio mute operation only engages when signal SNR is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 9

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | DAB_MUTE_SIGLOW_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0009 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | DAB_MUTE_SIGLOW_THRESHOLD[15:0] | DAB_MUTE_SIGLOW_THRESHOLD set the SNR threshold. The fic_quality based audio mute operation only engages when signal SNR is below this threshold.<br>The operation of fic_quality based audio mute only engages when property DAB_CTRL_DAB_MUTE_ENABLE is 1 AND signal SNR is below this threshold.<br>To allow the audio mute operation at all signal levels, set this property to a value greater than 30. |

### Property 0xE800. DAB_TEST_BER_CONFIG

DAB_TEST_BER_CONFIG sets up and enables the DAB BER test. The test is enabled by transitioning the ENABLE bit from 0 to 1. To restart the test the host must first write the enable bit to 0 and then back to 1. This test requires an ensemble test vector with the test pattern in one of the subchannels. The firmware must be built for a specific mode of this test. The default behavior is that this test will operate on packet mode data. Therefore the test vector must contain the test pattern in a packet mode service. The TMID for this service must be set to 3 and the data group flag must be 0, no data groups. By default subchannels defined for stream mode data, audio or FIDC will not work as the test pattern channel. Special versions of the software will be required if BER testing on one of the mentioned transmission modes is required. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | ENABLE | PATTERN[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | 0 | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:9 | Reserved | Always write to 0x00. |
| 8 | ENABLE | Enables the BER test. Once enabled the BER information can be collected using the DAB_TEST_GET_BER_INFO command. To start or restart the BER test this bit must transition from 0 to 1. |
| 7:0 | PATTERN[7:0] | Sets the BER test pattern character. This must match the character written into the test ETI file. Note that you must enable the audio service that contains this pattern in it. |

## 4.5. Commands Summary—AMHD

**Table 7. Commands Summary—AMHD**

| Cmd | Name | Description |
|---|---|---|
| 0x00 | RD_REPLY | Returns the status byte and data for the last command sent to the device. |
| 0x01 | POWER_UP | Power-up the device and set system settings. |
| 0x04 | HOST_LOAD | Loads an image from HOST over command interface |
| 0x05 | FLASH_LOAD | Loads an image from external FLASH over secondary SPI bus |
| 0x06 | LOAD_INIT | Prepares the bootloader to receive a new image. |
| 0x07 | BOOT | Boots the image currently loaded in RAM. |
| 0x08 | GET_PART_INFO | Reports basic information about the device. |
| 0x09 | GET_SYS_STATE | Reports system state information. |
| 0x0A | GET_POWER_UP_ARGS | Reports basic information about the device such as arguments used during POWER_UP. |
| 0x10 | READ_OFFSET | Reads a portion of response buffer from an offset. |
| 0x12 | GET_FUNC_INFO | Returns the Function revision information of the device. |
| 0x13 | SET_PROPERTY | Sets the value of a property. |
| 0x14 | GET_PROPERTY | Retrieve the value of a property. |
| 0x15 | WRITE_STORAGE | Writes data to the on board storage area at a specified offset. |
| 0x16 | READ_STORAGE | Reads data from the on board storage area from a specified offset. |
| 0x40 | AM_TUNE_FREQ | Tunes the AM receiver to a frequency in 1 kHz steps. |
| 0x41 | AM_SEEK_START | Initiates a seek for a channel that meets the validation criteria for AM. |
| 0x42 | AM_RSQ_STATUS | Returns status information about the received signal quality. |
| 0x43 | AM_ACF_STATUS | Returns status information about automatically controlled features. |
| 0x80 | GET_DIGITAL_SERVICE_LIST | Gets a service list of the ensemble. |
| 0x81 | START_DIGITAL_SERVICE | Starts an audio or data service. |
| 0x82 | STOP_DIGITAL_SERVICE | Stops an audio or data service. |
| 0x84 | GET_DIGITAL_SERVICE_DATA | Gets a block of data associated with one of the enabled data components of a digital services. |
| 0x92 | HD_DIGRAD_STATUS | Returns status information about the digital radio and ensemble. |

**Table 7. Commands Summary—AMHD**

| Cmd | Name | Description |
|---|---|---|
| 0x93 | HD_GET_EVENT_STATUS | Gets information about the various events related to the HD services. |
| 0x94 | HD_GET_STATION_INFO | Retrieves information about the ensemble broadcaster. |
| 0x95 | HD_GET_PSD_DECODE | Retrieves PSD information |
| 0x96 | HD_GET_ALERT_MSG | Retrieves the HD Alert message. |
| 0x97 | HD_PLAY_ALERT_TONE | Plays the HD Alert Tone. |
| 0x98 | HD_TEST_GET_BER_INFO | Reads the current BER information. |
| 0x99 | HD_SET_ENABLED_PORTS | Sets default ports retrieved after acquisition. |
| 0x9A | HD_GET_ENABLED_PORTS | Gets default ports retrieved after acquisition. |
| 0xE5 | TEST_GET_RSSI | Returns the reported RSSI in 8.8 format. |

### 4.5.1. AMHD Commands

**Command 0x00. RD_REPLY**

RD_REPLY command must be called to return the status byte and data for the last command sent to the device. This command is also used to poll the status byte as needed. To poll the status byte, send the RD_REPLY command and read the status byte. This can be done regardless of the state of the CTS bit in the status register. Please refer to individual command descriptions for the format of returned data. RD_REPLY is a hardware command and can be issued while device is powered down. For commands where the size of the response is returned, the user should send the RD_REPLY command to read the SIZE first. Each time the RD_REPLY command is sent, the STAUS bytes will still be returned.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x00 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | DATA_0[7:0] | | | | | | | |
| RESP5 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |
| ERR_CMD | Command Error.<br>0: No error<br>1: Error. The previous command failed. |

| Name | Function |
|---|---|
| DACQINT | Digital radio link change interrupt indicator. Indicates that something in the digital radio ensemble acquisition status has changed.<br>Service via the HD_DIGRAD_STATUS commands. |
| DSRVINT | Indicates that an enabled data component of one of the digital services requires attention. Service using the GET_DIGITAL_SERVICE_DATA command. |
| RSQINT | Received Signal Quality interrupt indicator.Indicates that a received signal metric is above or below a threshold defined by threshold properties.<br>Service via AM_RSQ_STATUS command. |
| ACFINT | Automatically controlled features interrupt indicator. Indicates the one of the dynamically system modifiers has crossed a programmed threshold.<br>Service via AM_ACF_STATUS command. |
| STCINT | Seek/Tune complete.<br>0: Tune complete has not been triggered. Do not send a new TUNE/SEEK command.<br>1: Tune complete has been triggered. It is safe to send a new TUNE/SEEK command. |
| DEVNTINT | Digital radio event change interrupt indicator. Indicates that a new event related to the digital radio has occurred.<br>Service via the HD_DIGRAD_STATUS commands. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| DSPERR | The DSP has encountered a frame overrun. This is a fatal error. |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |
| DATA_0[7:0] | First Data byte. |
| DATA_N[7:0] | Nth Data byte. |

### Command 0x01. POWER_UP

The POWER_UP initiates the boot process to move the device from power down to power up mode. There are two possible boot scenarios: Host image load and FLASH image load. When the host is loading the image the host first executes the POWER_UP command to set the system settings REF_CLK, etc). A LOAD_INIT command then prepares the bootloader to receive a new image. After the LOAD_INIT command, using the HOST_LOAD command loads the image into the device RAM. After the RAM is loaded the host issues the BOOT command. When booting a FLASH image the host issues the POWER_UP command to set the system settings. Then issues the FLASH_LOAD command to select and load the image from FLASH. Once the image is loaded the host sends the BOOT command to boot the application. Power-up is complete when the CTS bit is set. This command may only be sent while the device is powered down. Note: FLASH_LOAD is not supported in A0A or A0B revisions.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x01 | | | | | | | |
| ARG1 | CTSIEN | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ARG2 | 0 | 0 | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| ARG3 | 0 | IBIAS[6:0] | | | | | | |
| ARG4 | XTAL_FREQ[7:0] | | | | | | | |
| ARG5 | XTAL_FREQ[15:8] | | | | | | | |
| ARG6 | XTAL_FREQ[23:16] | | | | | | | |
| ARG7 | XTAL_FREQ[31:24] | | | | | | | |
| ARG8 | 0 | 0 | CTUN[5:0] | | | | | |
| ARG9 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | 0 | 0 | | 0 | 0 | | |
| ARG12 | 0 | | | | 0 | 0 | | |
| ARG13 | 0 | IBIAS_RUN[6:0] | | | | | | |
| ARG14 | 0 | | | | | | | |
| ARG15 | 0 | | | | | | | |

| Name | Function |
|---|---|
| CTSIEN | The bootloader will toggle a host interrupt line when CTS is available.<br>0: Disable toggling host interrupt line.<br>1: Enable toggling host interrupt line. |

| Name | Function |
|---|---|
| CLK_MODE[5:4] | Choose clock mode. See refclk spec sheet for more information<br>0: Oscillator and buffer are powered down.<br>1: Reference clock generator is in crystal mode.<br>2: Oscillator is off and circuit acts as single ended buffer.<br>3: Oscillator is off and circuit acts as differential buffer. |
| TR_SIZE[3:0] | XOSC TR_SIZE. See refclk spec sheet for more information.<br>Range: 0-15 |
| IBIAS[6:0] | XTAL IBIAS current at startup. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 0 to 1270 µA.<br>Range: 0-127 |
| XTAL_FREQ[31:0] | XTAL Frequency in Hz. The supported crystal frequencies are: [5.4 MHz–6.6 MHz] [10.8 MHz–13.2 MHz] [16.8 MHz–19.8 MHz] [21.6 MHz–26.4 MHz] [27 MHz–46.2 MHz] |
| CTUN[5:0] | CTUN. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator.<br>Range: 0-63 |
| IBIAS_RUN[6:0] | XTAL IBIAS current at runtime, after the XTAL oscillator has stabalized. See refclk spec sheet for more information. This parameter is only required if using the crystal oscillator. 10 µA steps, 10 to 1270 µA. If set to 0, will use the same value as IBIAS.<br>Range: 0-127 |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

| Name | Function |
|---|---|
| CTS | Clear to Send.<br>0: Wait before sending next command.<br>1: Clear to send next command. The next command may be sent. |

| Name | Function |
|---|---|
| ERR_CMD | Error.<br>0: No error<br>1: Error. The previous command failed. If STATUS1 is non-zero, it provides an indication of the cause of the error. |
| PUP_STATE[7:6] | Indicates the powerup state of the system.<br>0: The system has been reset but no POWER_UP + command has been issued. The system is currently waiting on the POWER_UP command.<br>1: Reserved<br>2: The bootloader is currently running.<br>3: An application was successfully booted and is currently running. |
| RSVD_STAT[5:4] | RFU (Reserved For Future Use). |
| REPOFERR | When set the control interface has dropped data during a reply read, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the given data arbiter and memory speed. |
| CMDOFERR | When set the control interface has dropped data during a command write, which is a fatal error. This is generally caused by running at a SPI clock rate that is too fast for the data arbiter and memory speed. |
| ARBERR | When set an arbiter error has occurred. |
| ERRNR | When set a non-recoverable error has occurred. The system keep alive timer has expired. |

### Command 0x04. HOST_LOAD

HOST_LOAD loads an image from HOST over command interface. It sends up to 4096 bytes of application image to the bootloader. Note: This command is much more efficient when the image is sent as multiples of 4 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x04 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | IMAGE_DATA0[7:0] | | | | | | | |
| ARG5 | IMAGE_DATA1[7:0] | | | | | | | |
| ARG6 | IMAGE_DATA2[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG7 | IMAGE_DATA3[7:0] | | | | | | | |
| ARG8 | IMAGE_DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| IMAGE_DATA0[7:0] | First byte of data stream from boot_img. |
| IMAGE_DATA1[7:0] | Second byte of data stream from boot_img. |
| IMAGE_DATA2[7:0] | Third byte of data stream from boot_img. |
| IMAGE_DATA3[7:0] | Fourth byte of data stream from boot_img. |
| IMAGE_DATA_N[7:0] | Nth byte of data stream. Maximum of 4096 bytes of data per HOST_LOAD command. For best results, N should be a multiple of 4. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x05. FLASH_LOAD

FLASH_LOAD loads the firmware image from an externally attached SPI flash over the secondary SPI bus. The image must be contiguous on the flash. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: FLASH_LOAD is only supported after patching the bootloader.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x05 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | 0x00 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG3 | 0x00 | | | | | | | |
| ARG4 | FLASH_START_ADDR[7:0] | | | | | | | |
| ARG5 | FLASH_START_ADDR[15:8] | | | | | | | |
| ARG6 | FLASH_START_ADDR[23:16] | | | | | | | |
| ARG7 | FLASH_START_ADDR[31:24] | | | | | | | |
| ARG8 | 0 | | | | | | | |
| ARG9 | 0 | | | | | | | |
| ARG10 | 0 | | | | | | | |
| ARG11 | 0 | | | | | | | |

| Name | Function |
|---|---|
| FLASH_START_ADDR[31:0] | Flash byte starting address of image to load |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x06. LOAD_INIT

LOAD_INIT prepares the bootloader to receive a new image. It will force the bootloader state to waiting for a new LOAD command (HOST_LOAD or FLASH_LOAD.) LOAD_INIT command must always be sent prior to a HOST_LOAD or a FLASH_LOAD command. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x06 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x07. BOOT

BOOT command boots the image currently loaded in RAM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x07 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x08. GET_PART_INFO

GET_PART_INFO reports basic information about the device such as Part Number, Part Version, ROM ID, etc. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_PART_INFO command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x08 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | CHIPREV[7:0] | | | | | | | |
| RESP5 | ROMID[7:0] | | | | | | | |
| RESP6 | X | | | | | | | |
| RESP7 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP8 | PART[7:0] | | | | | | | |
| RESP9 | PART[15:8] | | | | | | | |
| RESP10 | X | | | | | | | |
| RESP11 | X | | | | | | | |
| RESP12 | X | | | | | | | |
| RESP13 | X | | | | | | | |
| RESP14 | X | | | | | | | |
| RESP15 | X | | | | | | | |
| RESP16 | X | | | | | | | |
| RESP17 | X | | | | | | | |
| RESP18 | X | | | | | | | |
| RESP19 | X | | | | | | | |
| RESP20 | X | | | | | | | |
| RESP21 | X | | | | | | | |
| RESP22 | X | | | | | | | |

| Name | Function |
|---|---|
| CHIPREV[7:0] | Chip Mask Revision |
| ROMID[7:0] | ROM Id |
| PART[15:0] | Part Number (decimal) |

## Command 0x09. GET_SYS_STATE

GET_SYS_STATE reports basic system state information such as which mode is active; FM, DAB, etc. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: GET_SYS_STATE command is not supported in firmware revision A0A.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x09 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | IMAGE[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |

| Name | Function |
|---|---|
| IMAGE[7:0] | This field indicates which firmware image processed this command.<br>0: Bootloader is active<br>1: FMHD is active<br>2: DAB is active<br>3: TDMB or data only DAB image is active<br>4: FMHD Demod is active<br>5: AMHD is active<br>6: AMHD Demod is active |

## Command 0x0A. GET_POWER_UP_ARGS

GET_POWER_UP_ARGS reports basic information about the device such as which parameters were used during power up. This command will hold CTS until the reply is available. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x0A | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | X | X | X | X | X | X |
| STATUS1 | X | | | | | | | |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | RSVD_STAT[1:0] | | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | | | | | | | |
| RESP5 | X | X | X | X | X | X | X | X |
| RESP6 | X | X | CLK_MODE[1:0] | | TR_SIZE[3:0] | | | |
| RESP7 | X | IBIAS[6:0] | | | | | | |
| RESP8 | XTAL_FREQ[7:0] | | | | | | | |
| RESP9 | XTAL_FREQ[15:8] | | | | | | | |
| RESP10 | XTAL_FREQ[23:16] | | | | | | | |
| RESP11 | XTAL_FREQ[31:24] | | | | | | | |
| RESP12 | X | X | CTUN[5:0] | | | | | |
| RESP13 | X | X | X | X | X | X | X | X |
| RESP14 | X | | | | | | | |
| RESP15 | X | X | X | | X | X | | |
| RESP16 | X | | | | X | X | | |
| RESP17 | X | IBIAS_RUN[6:0] | | | | | | |

| Name | Function |
|---|---|
| CLK_MODE[5:4] | See POWER_UP command |
| TR_SIZE[3:0] | See POWER_UP command |
| IBIAS[6:0] | See POWER_UP command |
| XTAL_FREQ[31:0] | See POWER_UP command |
| CTUN[5:0] | See POWER_UP command |
| IBIAS_RUN[6:0] | See POWER_UP command |

### Command 0x10. READ_OFFSET

READ_OFFSET is used for applications that cannot read the entire response buffer. This type of application can use this command to read the response buffer in segments. The host must pass in an offset from the beginning of the response buffer to indicate the starting point from which to read. This offset must be modulo 4. The response buffer remains intact as in the READ_REPLY command so that the response can be read again if needed. This function is available for both I2C and SPI mode. This is a software command, therefore it is best to read as much data in each calling as possible. This will reduce the overhead associated with using this command. It is recommended that the minimum reply size be on the order of 512 bytes. This means that for APIs that return less the 512 bytes the standard READ_REPLY should be used. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x10 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The offset from the beginning of the response buffer from where to begin reading. The OFFSET parameter must be modulo four. An error is returned otherwise. For example to read a 1024 byte response in two chucks the host can call the READ_MORE command twice. The first call would have OFFSET set to 0 and the second call would have OFFSET set to 512. In both cases the host will clock out 516 bytes of data. The first chunk will include 4 bytes for the status word plus 512 bytes of response. The second chunk will include 4 bytes of status word plus the remaining 512 bytes of response. The response sections will be concatenated to form the entire response. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the data chunk. |

### Command 0x12. GET_FUNC_INFO

GET_FUNC_INFO returns the function revision number for currently loaded firmware (FMHD, AM etc.) as opposed to GET_PART_INFO command that provides the revision number for the combo firmware. For example, GET_PART_INFO would return A0B is the firmware revision while GET_FUNC_INFO would return 1.0.4 for FM function revision if the currently running firmware function is FM. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x12 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | REVEXT[7:0] | | | | | | | |
| RESP5 | REVBRANCH[7:0] | | | | | | | |
| RESP6 | REVINT[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP7 | NOSVN | X | LOCATION[1:0] | | X | X | MIXE-DREV | LOCALMOD |
| RESP8 | SVNID[7:0] | | | | | | | |
| RESP9 | SVNID[15:8] | | | | | | | |
| RESP10 | SVNID[23:16] | | | | | | | |
| RESP11 | SVNID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| REVEXT[7:0] | Major revision number (first part of 1.2.3). |
| REVBRANCH[7:0] | Minor revision number (second part of 1.2.3). |
| REVINT[7:0] | Build revision number (third part of 1.2.3). |
| NOSVN | If set the build was created with no SVN info. This image cannot be tracked back to the SVN repo. |
| LOCATION[5:4] | The location from which the image was built (Trunk, Branch or Tag).<br>0x0: The image was built from an SVN tag. Revision numbers are valid.<br>0x1: The image was built from an SVN branch. Revision numbers will be 0.<br>0x2: The image was built from the trunk. Revision number will be 0. |
| MIXEDREV | If set, the image was built with mixed revisions. |
| LOCALMOD | If set, the image has local modifications. |
| SVNID[31:0] | SVN ID from which the image was built. |

### Command 0x13. SET_PROPERTY

SET_PROPERTY sets the value of a property. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x13 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG4 | DATA0[7:0] | | | | | | | |
| ARG5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| PROPID[15:0] | The property ID of the property to set. |
| DATA0[15:0] | Value for the written property. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x14. GET_PROPERTY

GET_PROPERTY retrieves the value of a property or properties. The host may read as many properties as desired up to the end of a given property group. An attempt to read passed the end of the property group will result in zeros being read. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x14 | | | | | | | |
| ARG1 | COUNT[7:0] | | | | | | | |
| ARG2 | PROPID[7:0] | | | | | | | |
| ARG3 | PROPID[15:8] | | | | | | | |

| Name | Function |
|---|---|
| COUNT[7:0] | The number of properties to read. |
| PROPID[15:0] | The id of the property to retrieve. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |
| RESP5 | DATA0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[15:0] | Value of the first property. |

## Command 0x15. WRITE_STORAGE

WRITE_STORAGE writes data to the no board storage area at the specified offset. the largest block of data that can be written at one time is 256 bytes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x15 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG4 | LENGTH[7:0] | | | | | | | |
| ARG5 | LENGTH[15:8] | | | | | | | |
| ARG6 | 0x00 | | | | | | | |
| ARG7 | 0x00 | | | | | | | |
| ARG8 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer at which the data will be written. |
| LENGTH[15:0] | The number of bytes to write. |
| DATA0[7:0] | The first data byte to write. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x16. READ_STORAGE

READ_STORAGE reads data from the on board storage area from a specified offset. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x16 | | | | | | | |
| ARG1 | 0x00 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG2 | OFFSET[7:0] | | | | | | | |
| ARG3 | OFFSET[15:8] | | | | | | | |

| Name | Function |
|---|---|
| OFFSET[15:0] | The byte offset within the storage buffer from which the data will be read. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATA0[7:0] | The first byte of the read data. |

## Command 0x40. AM_TUNE_FREQ

AM_TUNE_FREQ tunes the AM receiver to a frequency in 1 kHz steps. The optional STC interrupt is set when the command completes the tune. Sending this command clears any pending STCINT or RSQINT bit in STATUS. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x40 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | TUNE_MODE[1:0] | | INJECTION[1:0] | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG2 | FREQ[7:0] | | | | | | | |
| ARG3 | FREQ[15:8] | | | | | | | |
| ARG4 | ANTCAP[7:0] | | | | | | | |
| ARG5 | ANTCAP[15:8] | | | | | | | |

| Name | Function |
|---|---|
| TUNE_MODE[3:2] | Set the desired tuning mode.<br>0: Tune and render analog audio as fast as possible, do not attempt to acquire HD.<br>1: Reserved<br>2: Tune and render analog audio as fast as possible, try to acquire HD and cross-fade to HD if acquired. Always selects the Main Program Service. To select an SPS use the START_DIGITAL_SERVICE command after the HD has been acquired.<br>3: Tune and attempt to acquire HD and render audio if successful. The Main Program Service (MPS) will be rendered. To select an SPS use the START_DIGITAL_-SERVICE command after the HD has been acquired. |
| INJECTION[1:0] | Injection selection<br>0: Automatic injection selection.<br>1: Low-side injection.<br>2: High-side injection. |
| FREQ[15:0] | Frequency in multiples of 1.0 kHz added to a starting frequency of 0 Hz |
| ANTCAP[15:0] | When non-zero this parameter sets the antenna tuning capacitor value to (ANTCAP-1)*142 fF (581.49 pF Max).<br>Range: 0-4096<br>0: Automatically determines the cap setting. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x41. AM_SEEK_START

AM_SEEK_START begins searching for a valid station. The search starts at AM_RSQ_STATUS:READFREQ + AM_SEEK_FREQUENCY_SPACING in the specified direction. In order for a station to be considered valid, each of the following thresholds must be met: AM_VALID_SNR_THRESHOLD, AM_VALID_RSSI_THRESHOLD, AM_VALID_MAX_TUNE_ERROR, and AM_VALID_HDLEVEL_THRESHOLD (if the value is non-zero, which indicates an HD seek). Clears any pending STCINT or RSQINT interrupt status. Seek can be cancelled through setting the CANCEL bit in the AM_RSQ_STATUS command. The optional STC interrupt is set when the command completes. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: This command is not implemented in Si46xx-A0A or Si46xx-A0B releases. Note: The OPEN_LOOP, FORCE_WB, TUNE_MODE, INJECTION, and ANTCAP parameters will not be functional until the Si46xx-A0C release.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x41 | | | | | | | |
| ARG1 | 0 | 0 | 0 | FORCE_WB | TUNE_MODE[1:0] | | INJECTION[1:0] | |
| ARG2 | 0 | 0 | 0 | 0 | 0 | 0 | SEEKUP | WRAP |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | ANTCAP[7:0] | | | | | | | |
| ARG5 | ANTCAP[15:8] | | | | | | | |

| Name | Function |
|---|---|
| FORCE_WB | Forces the DFE into wideband mode.<br>0: Normal operation.<br>1: Force Wide bandwidth |
| TUNE_MODE[3:2] | Set the desired tuning mode.<br>0: Tune and render analog audio as fast as possible, do not attempt to acquire HD.<br>1: Reserved<br>2: Tune and render analog audio as fast as possible, try to acquire HD and cross-fade to HD if acquired. Always selects the Main Program Service.<br>3: Tune and attempt to acquire HD and render audio if successful. After the station has been acquired, use START_DIGITAL_SERVICE to select services. |
| INJECTION[1:0] | Injection selection<br>0: Automatic injection selection.<br>1: Low-side injection.<br>2: High-side injection. |

| Name | Function |
|---|---|
| SEEKUP | determines direction of seek and band limit.<br>0: Seek down. Band limit is AM_SEEK_BAND_BOTTOM.<br>1: Seek up. Band limit is AM_SEEK_BAND_TOP. |
| WRAP | determines seek behavior upon reaching a band limit.<br>0: Halt seek at band limit.<br>1: When band limit is hit, continue seek from opposite band limit. |
| ANTCAP[15:0] | When non-zero this parameter sets the antenna tuning capacitor value to (ANT-CAP-1)*142 fF (581.49 pF Max).<br>Range: 0-4096<br>0: Automatically determines the cap setting. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

**Command 0x42. AM_RSQ_STATUS**

AM_RSQ_STATUS returns status information about the received signal quality. This command returns the Received Signal Strength Indicator (RSSI), Signal to Noise Ratio (SNR), frequency offset (FREQOFF), and Multipath (MULT) associated with the desired channel. It also indicates valid channel (VALID) and AFC rail status (AFCRL). This command can be used to check if the received signal is above the RSSI high threshold as reported by RSSIHINT or below the RSSI low threshold as reported by RSSILINT. It can also be used to check if the signal is above the SNR high threshold as reported by SNRHINT or below the SNR low threshold as reported by SNRLINT. It can be used to check if the detected multipath is above the Multipath high threshold as reported by MULTHINT or below the Multipath low threshold as reported by MULTLINT. The command clears the RSQINT, BLENDINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT, MULTHINT, and MULTLINT interrupt bits when the RSQACK bit of ARG1 is set. These are sticky meaning they will remain set until RSQACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in AM_RSQ_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Multipath metric and related interrupts/thresholds are not implemented in Si46xx-A0A or Si46xx-A0B releases.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x42 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | RSQACK | ATTUNE | CANCEL | STCACK |

| Name | Function |
|---|---|
| RSQACK | Clears RSQINT, SNRHINT, SNRLINT, RSSIHINT, RSSILINT if set. |
| ATTUNE | Return the values as of AM_VALID_SNR_TIME after tune. Only the signal quality metrics RSSI, SNR, ISSI, ASSI and the status bits INJECT, AFCL, and VALID are affected by setting this bit.<br>0: Return the current status<br>1: Return the snapshot taken at AM_VALID_SNR_TIME |
| CANCEL | Aborts a seek currently in progress.<br>0: Don't abort<br>1: Abort |
| STCACK | Clears the STC interrupt status indicator if set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | SNRHINT | SNRLINT | RSSI-HINT | RSSILINT |
| RESP5 | BLTF | X | X | X | X | X | AFCRL | VALID |
| RESP6 | READFREQ[7:0] | | | | | | | |
| RESP7 | READFREQ[15:8] | | | | | | | |
| RESP8 | FREQOFF[7:0] | | | | | | | |
| RESP9 | RSSI[7:0] | | | | | | | |
| RESP10 | SNR[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP11 | MOD[7:0] | | | | | | | |
| RESP12 | READANTCAP[7:0] | | | | | | | |
| RESP13 | READANTCAP[15:8] | | | | | | | |
| RESP14 | X | | | | | | | |
| RESP15 | HDLEVEL[7:0] | | | | | | | |

| Name | Function |
|---|---|
| SNRHINT | Indicates SNR above AM_RSQ_SNR_HIGH_THRESHOLD. |
| SNRLINT | Indicates SNR below AM_RSQ_SNR_LOW_THRESHOLD. |
| RSSIHINT | Indicates RSSI above AM_RSQ_RSSI_HIGH_THRESHOLD. |
| RSSILINT | Indicates RSSI below AM_RSQ_RSSI_LOW_THRESHOLD. |
| BLTF | Band Limit after AM_SEEK_START. Reports if a seek hit the band limit or wrapped to the original frequency. This does not indicate that the seek failed, only that further seeks from the current location would be unproductive.<br>Tune Failed after AM_TUNE_FREQ tuning with CONDITIONALRETURN bit set, reports '1' if the current station failed to meet the associated criteria and the original frequency was returned to. |
| AFCRL | AFC rail indicator. |
| VALID | Reports if the channel is valid based on the settings of AM_VALID_RSSI_-THRESHOLD, AM_VALID_SNR_THRESHOLD, AM_VALID_MAX_TUNE_ERROR |
| READFREQ[15:0] | Returns the currently tuned frequency. If the ATTUNE option is used, returns the frequency of the last completed tune. Applies during a normal tune and during seek. |
| FREQOFF[7:0] | Signed frequency offset in BPPM (2 PPM)<br>Range: -128-127 |
| RSSI[7:0] | Received signal strength indicator in dBuV.<br>Range: -128-127 |
| SNR[7:0] | RF SNR indicator in dB.<br>Range: -128-127 |
| MOD[7:0] | AM modulation index in percent.<br>Range: 0-100 |
| READANTCAP[15:0] | Returns the antenna tuning cap value. |
| HDLEVEL[7:0] | Reports the percentage of the HD OFDM symbols that appear to contain HD information in the upper and lower sidebands when measured over a given number of symbol periods. See AM_RSQ_HD_DETECTION to configure this metric.<br>Range: 0-100 |

### Command 0x43. AM_ACF_STATUS

AM_ACF_STATUS returns status information about automatically controlled features of the device. The automatically controlled features include blend, high cut, and softmute. The bits BLEND_INT, HIGHCUT_INT, and SMUTE_INT are sticky meaning they will remain set until ACFACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire. See the AM_ACF_INTERRUPT_SOURCE property for information on enabling the ACFINT The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x43 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ACFACK |

| Name | Function |
|---|---|
| ACFACK | Clears ACFINT and any ACF interrupt bits if set. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | X | X | HIGH-CUT_INT | SMUTE_INT |
| RESP5 | X | X | HIGH-CUT_-CONV | SMUTE_-CONV | X | X | HIGH-CUT_STATE | SMUTE_STATE |
| RESP6 | X | X | X | ATTN[4:0] | | | | |
| RESP7 | HIGHCUT[7:0] | | | | | | | |
| RESP8 | LOWCUT[7:0] | | | | | | | |

| Name | Function |
|---|---|
| HIGHCUT_INT | Indicates that the highcut cutoff frequency has crossed below the highcut threshold as set by AM_ACF_HIGHCUT_THRESHOLD. |
| SMUTE_INT | Indicates that softmute attenuation has increased above the softmute threshold as set by AM_ACF_SOFTMUTE_THRESHOLD. |
| HIGHCUT_CONV | High Cut convergence indicator.<br>0: The high cut filter is in transition.<br>1: The high cut filter is not in transition and is within the tolerance set by AM_ACF_HIGHCUT_TOLERANCE. |
| SMUTE_CONV | Softmute convergence indicator.<br>0: Softmute is in transition.<br>1: Softmute is not in transition and is within the tolerance set by AM_ACF_SOFT-MUTE_TOLERANCE. |
| HIGHCUT_STATE | High Cut indicator.<br>0: Audio is not frequency limited<br>1: Audio is frequency limited. |
| SMUTE_STATE | Soft Mute indicator.<br>0: Audio is not soft muted.<br>1: Audio is soft muted. |
| ATTN[4:0] | Soft mute attenuation level in dB.<br>Range: 0-31 |
| HIGHCUT[7:0] | Hicut cutoff frequency in units 100Hz.<br>Range: 10-50 |
| LOWCUT[7:0] | am low cut. |

**Command 0x80. GET_DIGITAL_SERVICE_LIST**

GET_DIGITAL_SERVICE_LIST gets a service list of the ensemble. This command should be issued each time an audio or data service list is updated as indicated by the ASRVLISTINT or DSRVLISTINT bit of the HD_GET_EVENT_STATUS command. This occurs shortly after tune time when a digital radio tuning mode is selected and the ensemble has been acquired. Please refer to iBiquity document: RX_IDD_2206 Appendix L (Get_All_Data_Services_Info and Get_All_Audio_Services_Info) for the format of the HD Radio Service List. In the case of HD this command also retrieves the audio or data service info when the appropriate service type option is selected. This service info is available whenever the AINFO or DINFO bit(s) are set in the HD_GET_EVENT_STATUS response. This audio and data information provides a quick look at the services in the ensemble and can be used to reduce scan time as this information is ready for parsing well before the service lists. The payload of these responses are defined in Table 5-4 of the RX_IDD_2206 main document. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x80 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | SERTYPE[1:0] | |

| Name | Function |
|---|---|
| SERTYPE[1:0] | Sets the type of service list to retrieve.<br>0: Get the audio(HD) service list.<br>1: Get the data(HD) service list.<br>2: Get the HD audio info.<br>3: Get the HD data info. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | SIZE[7:0] | | | | | | | |
| RESP5 | SIZE[15:8] | | | | | | | |
| RESP6 | DATA_0[7:0] | | | | | | | |
| RESP7 | DATA_N[7:0] | | | | | | | |

| Name | Function |
|---|---|
| SIZE[15:0] | The size of the service list in bytes. |
| DATA_0[7:0] | A max of 2047 bytes of service information. Please see the supplemental Digital Services Userís Guide. |
| DATA_N[7:0] | A max of 2047 bytes of service information. Please see the supplemental Digital Services Userís Guide. N(max) = SIZE-2. |

### Command 0x81. START_DIGITAL_SERVICE

START_DIGITAL_SERVICE starts an audio or data service. This command is used for HD audio and data services. To determine what services exist in an ensemble please use the GET_DIGITAL_SERVICE_LIST command. In HD radio applications the broadcaster does not always transmit this service information. In this case no data services are available but there may be multiple audio programs available. To view which audio services are available use the HD_DIGRAD_STATUS command's AUDIO_PROG_AVAIL field to see which audio programs can be selected. In addition the SERVICE_ID (service number) is not required when selecting an audio or data service. In this case please set the SERVICE_ID parameter to 0. I the case of starting an audio service, it is not required to stop a currently running audio service/program before starting a new one. The currently running audio service will be stopped automatically when the new service is requested. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x81 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SER-TYPE |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service to start for HD applications.<br>0: Select an audio service.<br>1: Select a data service. |

| Name | Function |
|---|---|
| SERVICE_ID[31:0] | The service's Service Number (HD). This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command. NOTE: For HD applications the is refered to as the service number and not the global service ID that is provided as part of the component information. |
| COMP_ID[31:0] | The service's Port/Program Number (HD). This ID is found in the component section of the service list returned by the GET_DIGITAL_SERVICE_LIST command. Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

## Command 0x82. STOP_DIGITAL_SERVICE

STOP_DIGITAL_SERVICE stops an audio or data service. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x82 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SER-TYPE |
| ARG2 | 0x00 | | | | | | | |
| ARG3 | 0x00 | | | | | | | |
| ARG4 | SERVICE_ID[7:0] | | | | | | | |
| ARG5 | SERVICE_ID[15:8] | | | | | | | |
| ARG6 | SERVICE_ID[23:16] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG7 | SERVICE_ID[31:24] | | | | | | | |
| ARG8 | COMP_ID[7:0] | | | | | | | |
| ARG9 | COMP_ID[15:8] | | | | | | | |
| ARG10 | COMP_ID[23:16] | | | | | | | |
| ARG11 | COMP_ID[31:24] | | | | | | | |

| Name | Function |
|---|---|
| SERTYPE | Sets the type of service list to start.<br>0: Select an audio service.<br>1: Select a data service. |
| SERVICE_ID[31:0] | The service's service ID. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command. |
| COMP_ID[31:0] | The service's Component ID or Port Number of the service to stop. This ID is found in the service list returned by the GET_DIGITAL_SERVICE_LIST command.<br>Note: COMP_ID will only accept 0 as a value for A0A release. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x84. GET_DIGITAL_SERVICE_DATA

GET_DIGITAL_SERVICE_DATA gets a block of data associated with one of the enabled data components of a digital service. Information about this block of data is found in the data header that is returned at the beginning of the data block. In order to determine the ideal number of PAYLOAD bytes to read, the header information can be read first followed by a second read of the full (header + PAYLOAD) length—it is unnecessary to call GET_DIGITAL_SERVICE_DATA twice to use this method. The data associated with this transaction will be discarded at the receipt of a next GET_DIGITAL_SERVICE_DATA command if STATUS_ONLY = 0. Reading past the end of the buffer will result in zeros for the invalid bytes. Please refer to iBiquity document: SY_IDD_1019s Rev F (sections 5 and 6) for the format of the HD Radio data service data. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x84 | | | | | | | |
| ARG1 | 0 | 0 | 0 | STATUS_ONLY | 0 | 0 | 0 | ACK |

| Name | Function |
|---|---|
| STATUS_ONLY | Returns only the interrupt source and available buffers information. No digital service data is removed from the service queue. This option should be used when polling for available data. |
| ACK | Acknowledge the reading of a data buffer or acknowledge one of the error interrupts. Whenever the ACK bit is set the DSRVINT bit in the status register will be cleared before CTS is released unless there are more data blocks to be read. All interrupt source bits associated with this command will also be cleared.<br>0: Don't acknowledge the interrupt<br>1: Acknowledging the interrupt will clear the DSRVINT bit and the interrupt source bits. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | X | X | X | X | X | DSR-VOV-FLINT | DSRVP-CKTINT |
| RESP5 | BUFF_COUNT[7:0] | | | | | | | |
| RESP6 | SRV_STATE[7:0] | | | | | | | |
| RESP7 | DATA_SRC[1:0] | | DSCTy[5:0] | | | | | |
| RESP8 | SERVICE_ID[7:0] | | | | | | | |
| RESP9 | SERVICE_ID[15:8] | | | | | | | |
| RESP10 | SERVICE_ID[23:16] | | | | | | | |
| RESP11 | SERVICE_ID[31:24] | | | | | | | |
| RESP12 | COMP_ID[7:0] | | | | | | | |
| RESP13 | COMP_ID[15:8] | | | | | | | |
| RESP14 | COMP_ID[23:16] | | | | | | | |
| RESP15 | COMP_ID[31:24] | | | | | | | |
| RESP16 | RFU[7:0] | | | | | | | |
| RESP17 | RFU[15:8] | | | | | | | |
| RESP18 | BYTE_COUNT[7:0] | | | | | | | |
| RESP19 | BYTE_COUNT[15:8] | | | | | | | |
| RESP20 | SEG_NUM[7:0] | | | | | | | |
| RESP21 | SEG_NUM[15:8] | | | | | | | |
| RESP22 | NUM_SEGS[7:0] | | | | | | | |
| RESP23 | NUM_SEGS[15:8] | | | | | | | |
| RESP24 | PAYLOAD0[7:0] | | | | | | | |
| RESP25 | PAYLOADN[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DSRVOVFLINT | The data services system has overflowed. This indicates that the host processor is not reading the services data out of the device fast enough. At most 8 outstanding data blocks can be queued in the device. |
| DSRVPCKTINT | Data for an enabled data service is ready for transfer to the host. |

| Name | Function |
|---|---|
| BUFF_COUNT[7:0] | Indicates the remaining number of buffers in the data service buffer queue. |
| SRV_STATE[7:0] | The status indicator for the associated service component.<br>0: Indicates that the service is playing out normally.<br>1: Indicates that the data service has stopped and that this is the last data block associated with the service.<br>2: Indicates that the system was not able to forward a packet due to a memory overflow. If this status is present the host is not reading the packet data fast enough.<br>3: Indicates the this data packet represents the beginning of a new data object.<br>4: Indicates the this data packet was received with errors. |
| DATA_SRC[7:6] | For DAB indicates the payload source. Not used in HD modes of operation.<br>0: Indicates that the payload is from a standard data service and DATA_TYPE is DSCTy.<br>1: Indicates that the payload is non-DLS PAD and DATA_TYPE is DSCTy.<br>2: Indicates that the payload is DLS PAD and DATA_TYPE is 0.<br>3: Reserved for future use. |
| DSCTy[5:0] | Used for DAB only. Returns 0 or the DSCTy depending on the DATA_SRC field. See DATA_SRC for details. |
| SERVICE_ID[31:0] | The Service ID this data is associated with. |
| COMP_ID[31:0] | The Component ID or Port Number this data is associated with. |
| RFU[15:0] | Reserved for future use. |
| BYTE_COUNT[15:0] | The length of this data block in bytes excluding the DSRV header. |
| SEG_NUM[15:0] | The segment number for this data block. If the data is associated with a stream this value represents a sequence number. Note that segment numbers can be returned out of order. Therefore the host must store and reassemble the data as needed. |
| NUM_SEGS[15:0] | The total number of segments to be returned for this data object. If NUM_SEGS=0 then no object length information is known or the associated service is a stream. |
| PAYLOAD0[7:0] | The first payload byte. |
| PAYLOADN[7:0] | The Nth payload byte where N = BYTE_COUNT-1. |

### Command 0x92. HD_DIGRAD_STATUS

HD_DIGRAD_STATUS returns status information about the digital radio and ensemble. The bits AERRHINT, AEERLINT, CDNRHINT, CDNRLINT, and ACQINT are sticky meaning they will remain set until DIGRAD_ACK is set. If the condition is still true after the interrupt is cleared another interrupt will fire assuming that bit is enabled in HD_DIGRAD_INTERRUPT_SOURCE. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x92 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DIGRAD_ACK |

| Name | Function |
|---|---|
| DIGRAD_ACK | Clears all pending digital radio interrupts. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | HDLO-GOINT | SRCA-NAINT | SRC-DIGINT | X | X | ACQINT | CDN-RHINT | CDN-RLINT |
| RESP5 | HDLOGO | SRCANA | SRCDIG | X | X | ACQ | CDNRH | CDNRL |
| RESP6 | BCTL[1:0] | | DAAI[5:0] | | | | | |
| RESP7 | CDNR[7:0] | | | | | | | |
| RESP8 | X | X | X | TX_GAIN[4:0] | | | | |
| RESP9 | AUDIO_PROG_AVAIL[7:0] | | | | | | | |
| RESP10 | AUDIO_PROG_PLAYING[7:0] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP11 | AUDIO_CA[7:0] | | | | | | | |
| RESP12 | CORE_AUDIO_ERR[7:0] | | | | | | | |
| RESP13 | CORE_AUDIO_ERR[15:8] | | | | | | | |
| RESP14 | CORE_AUDIO_ERR[23:16] | | | | | | | |
| RESP15 | CORE_AUDIO_ERR[31:24] | | | | | | | |
| RESP16 | ENH_AUDIO_ERR[7:0] | | | | | | | |
| RESP17 | ENH_AUDIO_ERR[15:8] | | | | | | | |
| RESP18 | ENH_AUDIO_ERR[23:16] | | | | | | | |
| RESP19 | ENH_AUDIO_ERR[31:24] | | | | | | | |
| RESP20 | PTY[7:0] | | | | | | | |
| RESP21 | PS_MODE[7:0] | | | | | | | |
| RESP22 | CODEC_MODE[7:0] | | | | | | | |

| Name | Function |
|---|---|
| HDLOGOINT | Indicates a change in the HD Logo display state. |
| SRCANAINT | Indicates a change in the Analog Audio Source state. Setting of this bit will only occur when the blend state of the analog audio changes. |
| SRCDIGINT | Indicates a change in the Digital Audio Source state. Setting of this bit will only occur when the blend state of the digital audio changes. |
| ACQINT | Indicates a change in the ensembles acquisition state. Setting of this bit will only occur when the acquisition state of the ensemble changes. |
| CDNRHINT | Indicates that CDNR has risen above the CDNR high threshold. This threshold is set in the HD_DIGRAD_CDNR_HIGH_THRESHOLD property. |
| CDNRLINT | Indicates that CDNR has dropped below the CDNR low threshold. This threshold is set in the HD_DIGRAD_CDNR_LOW_THRESHOLD property. |
| HDLOGO | HD Logo display<br>0: Consumer radio should not display HD Logo<br>1: Consumer radio should display HD Logo |
| SRCANA | Audio source is analog. If both SRCANA and SRCDIG are set, the audio is blending.<br>0: Audio source is not analog<br>1: Audio source is analog |

| Name | Function |
|---|---|
| SRCDIG | Audio source is digital. If both SRCDIG and SRCANA are set, the audio is blending.<br>0: Audio source is not digital<br>1: Audio source is digital |
| ACQ | The ensemble has been acquired. |
| CDNRH | Indicates that CDNR is currently above the CDNR high threshold. This threshold is set in the HD_DIGRAD_CDNR_HIGH_THRESHOLD property. |
| CDNRL | Indicates that CDNR is currently below the CDNR low threshold. This threshold is set in the HD_DIGRAD_CDNR_LOW_THRESHOLD property. |
| BCTL[7:6] | Reflects the state of the TX blend control bits communicated by the broadcast system.. |
| DAAI[5:0] | Digital Audio quality indicator, Indicates the current estimate of the audio quality of the currently selected audio service. lower numbers indicate poorer quality. The number is unitless and provided in 4.2 format. |
| CDNR[7:0] | Indicates the current estimate of the carrier to noise ratio as seen by the digital radio subsystem. |
| TX_GAIN[4:0] | Indicates the amount of gain the receiver must apply to the digital audio signal relative to analog audio. In 5-bit two's complement format. |
| AUDIO_PROG_AVAIL[7:0] | Indicates which of the 8 audio programs are available. This field is a bit field where bit 0 represents the MPS and bits 1-7 represent the SPS. For example if this field is set to 0x83 then SPS7, SPS1, and the MPS are availble. To select one of these services use the START_DIGITAL_SERVICE command. |
| AUDIO_PROG_PLAYING[7:0] | Indicates which of the audio programs is currently playing. |
| AUDIO_CA[7:0] | Indicates which of the audio programs have CA associated with them. |
| CORE_AUDIO_ERR[31:0] | Indicates the total number of core audio frames received in error. |
| ENH_AUDIO_ERR[31:0] | Indicates the total number of enhanced audio frames received in error. |
| PTY[7:0] | Indicates the durrent audio program type PTy. |
| PS_MODE[7:0] | Indicates the primary service mode for the currently tuned ensemble. |
| CODEC_MODE[7:0] | Indicates the audio codec configuration for the currently selected audio program.<br>0: FM Hybrid single-stream on P1 Channel<br>2: AM Hybrid/All Digital dual-stream<br>10: FM dual-stream on SPS<br>13: FM Hybrid/All Digital single-stream on P3 Channel |

### Command 0x93. HD_GET_EVENT_STATUS

HD_GET_EVENT_STATUS retrieves the status of HD related events. This includes items such as new alarms available, new PSD, New station info, etc. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x93 | | | | | | | |
| ARG1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EVENT_ACK |

| Name | Function |
|---|---|
| EVENT_ACK | Clears all pending digital radio event interrupts. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | DIN-FOINT | AIN-FOINT | X | ALERTINT | PSDINT | SISINT | DSRVLISTINT | ASRVLISTINT |
| RESP5 | DINFO | AINFO | X | X | PSD | SIS | DSRVLIST | ASRVLIST |
| RESP6 | ASRVLISTVER[7:0] | | | | | | | |
| RESP7 | ASRVLISTVER[15:8] | | | | | | | |
| RESP8 | DSRVLISTVER[7:0] | | | | | | | |
| RESP9 | DSRVLISTVER[15:8] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP10 | X | X | X | SIS_LO-CATION | X | SIS_LONG_NAME | SIS_SHORT_NAME | SIS_ID |
| RESP11 | X | X | SIS_SLOGAN | SIS_BA-SIC_SIS | SIS_UNI-V_SHORT_NAME | SIS_LEAP_SEC | SIS_TI-MEZONE | SIS_MESSAGE |
| RESP12 | X | TEXT | SHORT | LANG | GENRE | ALBUM | ARTIST | TITLE |
| RESP13 | ID | OWNER | DESC | NAME | RECV | URL | VALID | PRICE |
| RESP14 | X | X | X | X | X | ARMS-GAVAIL | ARFRM0 | ARFRM |
| RESP15 | ARFRAMECNT[7:0] | | | | | | | |
| RESP16 | ARMSGID[7:0] | | | | | | | |
| RESP17 | ARCRC7[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DINFOINT | New Data Info. Indicates that the data service info has been updated. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| AINFOINT | New Audio Info. Indicates that the audio service info has been updated. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| ALERTINT | New Alert interrupt. Indicates that a new alert has been issued by the broadcaster. Retrieve the event info with the HD_GET_ALERT_MSG command. |
| PSDINT | New Program Service Data interrupt. Indicates that new program service data on the currently playing audio service is available. Retrieve the PSD with the HD_GET_PSD_DECODE command. |
| SISINT | New station information interrupt. Indicates that the station information service data has changed. The updated information is retrieved with the HD_GET_STATION_INFO command. |
| DSRVLISTINT | New data service list interrupt. Indicates that a new digital data service list is available. The new service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. |
| ASRVLISTINT | New audio service list interrupt. Indicates that a new digital audio service list is available. The new service list is retrieved with the GET_DIGITAL_SERVICE_LIST command. |
| DINFO | Data Service Info is available. Indicates data service info is available. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |
| AINFO | Audio Service Info is available. Indicates audio service info is available. Retrieve the this data with the GET_DIGITAL_SERVICE_LIST command. |

| Name | Function |
|---|---|
| PSD | Program Service Data is available. Indicates program service data on the currently playing audio service is available. Retrieve the PSD with the HD_GET_PSD_DECODE command. |
| SIS | Station information is available. Indicates that station information service data is available. The station information service data is retrieved with the HD_GET_STATION_INFO command. |
| DSRVLIST | Data Service list available. Indicates that a digital data service list is available. |
| ASRVLIST | Audio Service list available. Indicates that a digital audio service list is available. |
| ASRVLISTVER[15:0] | Indicates the current version of the audio service list. This field is incremented by 1 each time the audio service list is updated. The host can use this field to help determine if a new audio service list needs to be collected. |
| DSRVLISTVER[15:0] | Indicates the current version of the data service list. This field is incremented by 1 each time the data service list is updated. The host can use this field to help determine if a new data service list needs to be collected. |
| SIS_LOCATION | station location is available. |
| SIS_LONG_NAME | station long name is available. |
| SIS_SHORT_NAME | station short name is available. |
| SIS_ID | station ID is available. |
| SIS_SLOGAN | station slogan is available. |
| SIS_BASIC_SIS | the logical OR of bits SIS_LOCATION, SIS_LONG_NAME, SIS_SHORT_NAME and SIS_ID. |
| SIS_UNIV_SHORT_NAME | station universal short name is available. |
| SIS_LEAP_SEC | station leap seconds is available. |
| SIS_TIMEZONE | station time zone is available. |
| SIS_MESSAGE | station message is available. |
| TEXT | PSD Comment:The actual text is available |
| SHORT | PSD Comment:Short Content Description is available |
| LANG | PSD Comment:Language is available |
| GENRE | PSD Genre is available |
| ALBUM | PSD Album is available |
| ARTIST | PSD Artist is available |
| TITLE | PSD Title is available |
| ID | PSD UFID:Owner ID is available |
| OWNER | PSD UFID:Owner is available |

| Name | Function |
|---|---|
| DESC | PSD Commercial:Description is available |
| NAME | PSD Commercial:Name of Seller is available |
| RECV | PSD Commercial:Received As is available |
| URL | PSD Commercial:Contact URL is available |
| VALID | PSD Commercial:Valid Until is available |
| PRICE | PSD Commercial:Price is available |
| ARMSGAVAIL | Complete AR MSG is available |
| ARFRM0 | AR Frame 0 has been received |
| ARFRM | AR Frame (any) has been received |
| ARFRAMECNT[7:0] | The Alert message frame counter. |
| ARMSGID[7:0] | A complete alert message is available. The message can be retreived with the HD_GET_ALERT_MSG command. |
| ARCRC7[7:0] | The alert message CRC7 value extracted from frame 0. |

### Command 0x94. HD_GET_STATION_INFO

HD_GET_STATION_INFO retrieves information about the ensemble broadcaster. The station information is defined in the 2206 standard. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x94 | | | | | | | |
| ARG1 | INFO_SELECT[7:0] | | | | | | | |

| Name | Function |
|---|---|
| INFO_SELECT[7:0] | Selects which information returned by this command.<br>0: RSVD Do not use<br>1: Returns a station message. An arbitrary text message such as telephone number, URL, etc. High-priority messages should take precedence over all other SIS data and PSD. For the station message, a maximum length of 190 bytes is currently supported. The Station Message is only returned when updates have been received OTA. A LENGTH of 0 will be returned if no update has been received since the last request.<br>2: Returns the Local time zone and daylight savings time (DST) information, allowing receivers to automatically calculate and display time of day.<br>3: Returns a leap-second correction factor occasionally adjusts UTC by one second to keep it synchronized with astronomical time. Since GPS time does not apply this correction, the two standards have diverged slightly over the years.<br>4: Provides an alternative to basic short station name, for countries where different text-encoding schemes may be required. The Append Byte indicates whether "-FM" should be appended to the short station name by the HC. The maximum length for the name is 12 bytes. For US markets, the Station Name (short form) should be retrieved using then BASICSIS option.<br>5: Returns basic SIS data. See the 2206 documentation for full details. Basically this includes items such as the station ID and station location. The information returned by this option is controlled by the HD_EVENT_SIS_CONFIG property.<br>6: Returns the station slogan. This field supersedes the long station name returned by the BASIC SIS option 5. For the slogan, a maximum length of 95 bytes is currently supported. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVNTINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | LENGTH[15:8] | | | | | | | |
| RESP6 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[15:0] | The number of data bytes returned by this command. |
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 SIS documentation for details about then returned data. |

### Command 0x95. HD_GET_PSD_DECODE

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x95 | | | | | | | |
| ARG1 | PROGRAM[7:0] | | | | | | | |
| ARG2 | FIELD[7:0] | | | | | | | |

| Name | Function |
|---|---|
| PROGRAM[7:0] | Program Number<br>0: MPS<br>1: SPS 1<br>2: SPS 2<br>3: SPS 3<br>4: SPS 4<br>5: SPS 5<br>6: SPS 6<br>7: SPS 7<br>0xFF: Currently playing audio service |

| Name | Function |
|---|---|
| FIELD[7:0] | PSD Field<br>0: Title<br>1: Artist<br>2: Album<br>3: Genre<br>4: Comment: Language<br>5: Comment: Short Content Description<br>6: Comment: The actual text<br>8: Commercial: Price<br>9: Commercial: Valid Until<br>10: Commercial: Contact URL<br>11: Commercial: Received As<br>12: Commercial: Name of Seller<br>13: Commercial: Description<br>14: UFID: Owner<br>15: UFID: Owner ID |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | X | | | | | | | |
| RESP5 | X | | | | | | | |
| RESP6 | DATATYPE[7:0] | | | | | | | |
| RESP7 | LENGTH[7:0] | | | | | | | |
| RESP8 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| DATATYPE[7:0] | 0: 8 bit character: ISO/IEC 8859-1:1998<br>1: 16 bit character: ISO/IEC 10646-1:2000 |
| LENGTH[7:0] | The number of data bytes returned by this command. |

| Name | Function |
|---|---|
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 SIS documentation for details about then returned data. |

### Command 0x96. HD_GET_ALERT_MSG

HD_GET_ALERT_MSG retrieves alert message. Alerts are special messages provided by the broadcaster that may signal important information about emergencies or events. Full details about Alerts can be found in the 2206 standard. This API is used to collect the alert data and is used in response to an alert event. See the HD_GET_EVENT_STATUS command for details on the alert event. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x96 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | LENGTH[15:8] | | | | | | | |
| RESP6 | DATA0[7:0] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[15:0] | The number of data bytes returned in the alert payload. The length of the alert message will not exceed 395 bytes. |
| DATA0[7:0] | The first of LENGTH data bytes returned by this command. Please see the 2206 Alert documentation for details about then returned data. |

### Command 0x97. HD_PLAY_ALERT_TONE

HD_PLAY_ALERT_TONE plays the alert tone. Alerts are special messages provided by the broadcaster that may signal important information about emergencies or events. Full details about Alerts can be found in the 2206 standard. This API is used to play an alert tone at the host's discretion. It is recommended that the host play this tone for each unique alert message it receives. If the host chooses it can also have these tones played automatically on every alert message. See the HD_EVENT_ALERT_CONFIG property for details on playing alert tones automatically. Also see the HD_GET_EVENT_STATUS command for details on the alert event. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x97 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x98. HD_TEST_GET_BER_INFO

HD_TEST_GET_BER_INFO reads the current BER information for the HD digital demod. The information returned by this command is only meaningful if the BER test vector (IB_FMr208c_e1wfc204 for FMHD, IB_AMr208a_e1awfb00 for AMHD) is being received. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x98 | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | PIDS_BLOCK_ERRORS[7:0] | | | | | | | |
| RESP5 | PIDS_BLOCK_ERRORS[15:8] | | | | | | | |
| RESP6 | PIDS_BLOCK_ERRORS[23:16] | | | | | | | |
| RESP7 | PIDS_BLOCK_ERRORS[31:24] | | | | | | | |
| RESP8 | PIDS_BLOCKS_TESTED[7:0] | | | | | | | |
| RESP9 | PIDS_BLOCKS_TESTED[15:8] | | | | | | | |
| RESP10 | PIDS_BLOCKS_TESTED[23:16] | | | | | | | |
| RESP11 | PIDS_BLOCKS_TESTED[31:24] | | | | | | | |
| RESP12 | PIDS_BIT_ERRORS[7:0] | | | | | | | |
| RESP13 | PIDS_BIT_ERRORS[15:8] | | | | | | | |
| RESP14 | PIDS_BIT_ERRORS[23:16] | | | | | | | |
| RESP15 | PIDS_BIT_ERRORS[31:24] | | | | | | | |
| RESP16 | PIDS_BITS_TESTED[7:0] | | | | | | | |
| RESP17 | PIDS_BITS_TESTED[15:8] | | | | | | | |
| RESP18 | PIDS_BITS_TESTED[23:16] | | | | | | | |
| RESP19 | PIDS_BITS_TESTED[31:24] | | | | | | | |
| RESP20 | P3_BIT_ERRORS[7:0] | | | | | | | |
| RESP21 | P3_BIT_ERRORS[15:8] | | | | | | | |
| RESP22 | P3_BIT_ERRORS[23:16] | | | | | | | |
| RESP23 | P3_BIT_ERRORS[31:24] | | | | | | | |

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| RESP24 | P3_BITS_TESTED[7:0] | | | | | | | |
| RESP25 | P3_BITS_TESTED[15:8] | | | | | | | |
| RESP26 | P3_BITS_TESTED[23:16] | | | | | | | |
| RESP27 | P3_BITS_TESTED[31:24] | | | | | | | |
| RESP28 | P2_BIT_ERRORS[7:0] | | | | | | | |
| RESP29 | P2_BIT_ERRORS[15:8] | | | | | | | |
| RESP30 | P2_BIT_ERRORS[23:16] | | | | | | | |
| RESP31 | P2_BIT_ERRORS[31:24] | | | | | | | |
| RESP32 | P2_BITS_TESTED[7:0] | | | | | | | |
| RESP33 | P2_BITS_TESTED[15:8] | | | | | | | |
| RESP34 | P2_BITS_TESTED[23:16] | | | | | | | |
| RESP35 | P2_BITS_TESTED[31:24] | | | | | | | |
| RESP36 | P1_BIT_ERRORS[7:0] | | | | | | | |
| RESP37 | P1_BIT_ERRORS[15:8] | | | | | | | |
| RESP38 | P1_BIT_ERRORS[23:16] | | | | | | | |
| RESP39 | P1_BIT_ERRORS[31:24] | | | | | | | |
| RESP40 | P1_BITS_TESTED[7:0] | | | | | | | |
| RESP41 | P1_BITS_TESTED[15:8] | | | | | | | |
| RESP42 | P1_BITS_TESTED[23:16] | | | | | | | |
| RESP43 | P1_BITS_TESTED[31:24] | | | | | | | |

| Name | Function |
|---|---|
| PIDS_BLOCK_ERRORS[31:0] | The number of PIDS blocks received with errors. |
| PIDS_BLOCKS_TESTED[31:0] | The total number of PIDS blocks received. |
| PIDS_BIT_ERRORS[31:0] | The number of PIDS bits received with errors. |
| PIDS_BITS_TESTED[31:0] | The total number of PIDS bits received. |
| P3_BIT_ERRORS[31:0] | The number of P3 bits received with errors. |
| P3_BITS_TESTED[31:0] | The total number of P3 bits received. |

| Name | Function |
|---|---|
| P2_BIT_ERRORS[31:0] | The number of P2 bits received with errors. |
| P2_BITS_TESTED[31:0] | The total number of P2 bits received. |
| P1_BIT_ERRORS[31:0] | The number of P1 bits received with errors. |
| P1_BITS_TESTED[31:0] | The total number of P1 bits received. |

### Command 0x99. HD_SET_ENABLED_PORTS

HD_SET_ENABLED_PORTS sets the default HD ports retrieved/enabled when HD has been acquired. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x99 | | | | | | | |
| ARG1 | LENGTH[7:0] | | | | | | | |
| ARG2 | DATA_0[7:0] | | | | | | | |
| ARG3 | DATA_0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[7:0] | The numbers of ports included in this command. Maximum 64. |
| DATA_0[15:0] | The first port address of LENGTH addresses. |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |

### Command 0x9A. HD_GET_ENABLED_PORTS

HD_GET_ENABLED_PORTS gets the default HD ports retrieved when HD has been acquired. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0x9A | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | LENGTH[7:0] | | | | | | | |
| RESP5 | X | | | | | | | |
| RESP6 | DATA_0[7:0] | | | | | | | |
| RESP7 | DATA_0[15:8] | | | | | | | |

| Name | Function |
|---|---|
| LENGTH[7:0] | The numbers of ports previously set. Maximum 64. |
| DATA_0[15:0] | If set, the first port byte. |

### Command 0xE5. TEST_GET_RSSI

TEST_GET_RSSI returns the reported RSSI in 8.8 format. This command is used to help calibrate the frontend tracking circuit. It returns the RSSI value in dBuV to 1/256 of a dB. The command is complete when the CTS bit (and optional interrupt) is set. The ERR bit (and optional interrupt) is set if an invalid argument is sent. Note that only a single interrupt occurs if both the CTS and ERR bits are set. The command may only be sent in powerup mode. Note: TEST_GET_RSSI command is not supported in firmware revision A0A or A0B.

**Command**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| CMD | 0xE5 | | | | | | | |
| ARG1 | 0 | | | | | | | |

**Response**

| Bit | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|
| STATUS0 | CTS | ERR_CMD | DAC-QINT | DSRVINT | RSQINT | X | ACFINT | STCINT |
| STATUS1 | X | X | DEVN-TINT | X | X | X | X | X |
| STATUS2 | X | | | | | | | |
| STATUS3 | PUP_STATE[1:0] | | X | DSPERR | REPO-FERR | CMDO-FERR | ARBERR | ERRNR |
| RESP4 | RSSI_HIGHRES[7:0] | | | | | | | |
| RESP5 | RSSI_HIGHRES[15:8] | | | | | | | |

| Name | Function |
|---|---|
| RSSI_HIGHRES[15:0] | The RSSI measurement in dBuV to 1/256 of a dB. |

## 4.6. Properties Summary—AMHD

**Table 8. Properties Summary—AMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x0000 | INT_CTL_ENABLE | Interrupt enable property | 0x0000 |
| 0x0001 | INT_CTL_REPEAT | Interrupt repeat property | 0x0000 |
| 0x0200 | DIGITAL_IO_OUTPUT_SELECT | Selects digital audio Master or Slave. | 0 |
| 0x0201 | DIGITAL_IO_OUTPUT_SAMPLE_RATE | Sets output sample audio rate in units of 1 Hz. | 48000 |
| 0x0202 | DIGITAL_IO_OUTPUT_FORMAT | Configure digital output format. | 0x1800 |
| 0x0203 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 | Deviations from the standard framing mode | 0 |
| 0x0204 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 | Deviations from the standard framing mode | 0 |
| 0x0205 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 | Deviations from the standard framing mode | 0 |
| 0x0206 | DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 | Deviations from the standard framing mode | 0 |
| 0x0300 | AUDIO_ANALOG_VOLUME | Sets the audio analog volume. | 63 |
| 0x0301 | AUDIO_MUTE | AUDIO_MUTE property mutes/unmutes each audio output independently. | 0x0000 |
| 0x0500 | AM_AVC_MIN_GAIN | Sets the minimum gain for automatic volume control. | -2048 |
| 0x0501 | AM_AVC_MAX_GAIN | Sets the maximum gain for automatic volume control. | 10220 |
| 0x0800 | PIN_CONFIG_ENABLE | Pin configuration property | 0x8001 |
| 0x0900 | WAKE_TONE_ENABLE | Enables the wake tone feature. | 0 |
| 0x0901 | WAKE_TONE_PERIOD | Sets the wake tone duty cycle. | 250 |
| 0x0902 | WAKE_TONE_FREQ | Sets the wake tone frequency. | 750 |
| 0x0903 | WAKE_TONE_AMPLITUDE | Sets the wake tone amplitude. | 8 |
| 0x2200 | AM_CHBW_SQ_LIMITS | The SNR/RSSI threshold for maximum and minimum channel filter bandwidth | 0x1E0F |
| 0x2201 | AM_CHBW_SQ_CHBW | The maximum and minimum channel filter bandwidth. | 0x2314 |
| 0x2202 | AM_CHBW_SQ_WIDENING_TIME | SNR/RSSI based channel filter bandwidth widening time | 2048 |

**Table 8. Properties Summary—AMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x2203 | AM_CHBW_SQ_NARROWING_TIME | SNR/RSSI based channel filter bandwidth narrowing time | 16 |
| 0x2204 | AM_CHBW_OVERRIDE_BW | Overrides the automatically controlled channel filter setting. | 0 |
| 0x4100 | AM_SEEK_BAND_BOTTOM | Sets the lower seek boundary of the AM band in multiples of 1kHz. | 520 |
| 0x4101 | AM_SEEK_BAND_TOP | Sets the upper seek boundary for the AM band in multiples of 1kHz. | 1710 |
| 0x4102 | AM_SEEK_FREQUENCY_SPACING | Sets the frequency spacing for the AM band in multiples of 1kHz when performing a seek. | 10 |
| 0x4200 | AM_VALID_MAX_TUNE_ERROR | Sets the maximum frequency error allowed before setting the AFCRL indicator. | 75 |
| 0x4201 | AM_VALID_RSSI_TIME | Sets the amount of time in ms to allow the RSSI/ISSI metrics to settle before evaluating. | 8 |
| 0x4202 | AM_VALID_RSSI_THRESHOLD | Sets the RSSI threshold for a valid AM Seek/Tune. | 35 |
| 0x4203 | AM_VALID_SNR_TIME | Sets the amount of time in ms to allow the SNR metric to settle before evaluating. | 40 |
| 0x4204 | AM_VALID_SNR_THRESHOLD | Sets the SNR threshold for a valid AM Seek/Tune. | 10 |
| 0x4205 | AM_VALID_HDLEVEL_THRESHOLD | Sets the HDLEVEL threshold for a valid AM Seek/Tune. | 0 |
| 0x4300 | AM_RSQ_INTERRUPT_SOURCE | Configures interrupt related to Received Signal Quality metrics. | 0 |
| 0x4301 | AM_RSQ_SNR_HIGH_THRESHOLD | AM_RSQ_SNR_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the SNR is above this threshold. | 127 |
| 0x4302 | AM_RSQ_SNR_LOW_THRESHOLD | AM_RSQ_SNR_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the SNR is below this threshold. | -128 |
| 0x4303 | AM_RSQ_RSSI_HIGH_THRESHOLD | AM_RSQ_RSSI_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the RSSI is above this threshold. | 127 |

**Table 8. Properties Summary—AMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x4304 | AM_RSQ_RSSI_LOW_THRESHOLD | Sets low threshold which triggers the RSQ interrupt if the RSSI is below this threshold. | -128 |
| 0x4305 | AM_RSQ_HD_DETECTION | Configures the Fast HD Detection routine. | 0x8020 |
| 0x4400 | AM_ACF_INTERRUPT_SOURCE | Enables the ACF interrupt sources. | 0 |
| 0x4401 | AM_ACF_SOFTMUTE_THRESHOLD | Sets the softmute interrupt threshold in dB attenuation. | 31 |
| 0x4402 | AM_ACF_HIGHCUT_THRESHOLD | Sets the high cut interrupt threshold. | 0 |
| 0x4403 | AM_ACF_SOFTMUTE_TOLERANCE | Sets the distance from the final softmute value that triggers softmute convergence flag. | 2 |
| 0x4404 | AM_ACF_HIGHCUT_TOLERANCE | Sets the distance from the final high cut frequency that triggers the high cut convergence flag. | 20 |
| 0x4405 | AM_ACF_CONTROL_SOURCE | Determines what metrics control various ACF features. | 0 |
| 0x4500 | AM_SOFTMUTE_SQ_LIMITS | Sets the SNR limits for soft mute attenuation. | 0x0800 |
| 0x4501 | AM_SOFTMUTE_SQ_ATTENUATION | Sets the SNR based softmute attenuation limits. | 0x000C |
| 0x4502 | AM_SOFTMUTE_SQ_ATTACK_TIME | Sets the soft mute attack time. | 120 |
| 0x4503 | AM_SOFTMUTE_SQ_RELEASE_TIME | Sets the soft mute release time. | 500 |
| 0x4600 | AM_HIGHCUT_SQ_LIMITS | SQ based high cut high threshold. | 0x0a06 |
| 0x4601 | AM_HIGHCUT_SQ_CUTOFF_FREQ | Max and Min audio frequencies | 0x280A |
| 0x4602 | AM_HIGHCUT_SQ_ATTACK_TIME | Hi-cut cutoff frequency attack time in ms. | 16 |
| 0x4603 | AM_HIGHCUT_SQ_RELEASE_TIME | Hi-cut cutoff frequency release time in ms. | 2000 |
| 0x4800 | AM_DEMOD_AFC_RANGE | Specify the AFC range | 0x0000 |
| 0x8100 | DIGITAL_SERVICE_INT_SOURCE | Configures the interrupt sources for digital services | 0x0000 |
| 0x9101 | HD_BLEND_OPTIONS | This property provides options to control HD/analog audio blend behavior. | 0x0002 |
| 0x9102 | HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME | This property sets the amount of time it takes in ms to blend from analog to HD. This property only applies to primary service channel. | 750 |

**Table 8. Properties Summary—AMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x9103 | HD_BLEND_HD_TO_ANALOG_TRANSITION_-TIME | This property sets the amount of time it takes in ms to blend from HD to analog. This property only applies to primary service channel. | 100 |
| 0x9106 | HD_BLEND_DYNAMIC_GAIN | This property sets the digital audio dynamic linear scaling factor. Setting this property to non-zero will override the broadcaster specified digital gain. The property is a signed 8 bit fractional number where the linear gain is equal to DGAIN/128. | 0 |
| 0x9200 | HD_DIGRAD_INTERRUPT_SOURCE | Configures interrupts related to digital receivers HD_DIGRAD_STATUS command. | 0 |
| 0x9201 | HD_DIGRAD_CDNR_LOW_THRESHOLD | Configures CDNR LOW interrupt interrupt threshold. | 0 |
| 0x9202 | HD_DIGRAD_CDNR_HIGH_THRESHOLD | Configures CDNR HIGH interrupt interrupt threshold. | 127 |
| 0x9203 | HD_DIGRAD_AUTO_ACQUIRE | This property sets the radio to automatically acquire digital radio. | 1 |
| 0x9300 | HD_EVENT_INTERRUPT_SOURCE | Configures interrupts related to digital receiver (HD_GET_EVENT_STATUS). | 0 |
| 0x9301 | HD_EVENT_SIS_CONFIG | This property configures which basic SIS information is returned by the HD_GET_STATION_INFO command BASICSIS option. | 0x0017 |
| 0x9302 | HD_EVENT_ALERT_CONFIG | This property configures HD alerts. Alert information is returned by the HD_GET_ALERT_MSG command. | 0x0001 |
| 0x9500 | HD_PSD_ENABLE | Sets which audio services will provide program service data. | 0 |
| 0x9501 | HD_PSD_FIELD_MASK | Sets which PSD fields will be decoded. | 0xFFFF |
| 0x9701 | HD_AUDIO_CTRL_PROGRAM_LOSS_-THRESHOLD | Controls the duration before reverting to MPS audio after an SPS audio program is removed or lost. | 0 |
| 0x9702 | HD_AUDIO_CTRL_BALL_GAME_ENABLE | Selects the audio output for hybrid waveforms when the TX Blend Control Status (BCTL) bits are set to 01. | 1 |
| 0x9900 | HD_CODEC_MODE_0_BLEND_THRESHOLD | Blend threshold for Codec Mode 0 | 3 |
| 0x9901 | HD_CODEC_MODE_0_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 0 | 0 |

**Table 8. Properties Summary—AMHD**

| Prop | Name | Description | Default |
|---|---|---|---|
| 0x9902 | HD_CODEC_MODE_0_BLEND_RATE | Blend Rate for Codec Mode 0 | 1 |
| 0x9903 | HD_CODEC_MODE_2_BLEND_THRESHOLD | Blend threshold for Codec Mode 2 | 3 |
| 0x9904 | HD_CODEC_MODE_2_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 2 | 7398 |
| 0x9905 | HD_CODEC_MODE_2_BLEND_RATE | Blend Rate for Codec Mode 2 | 1 |
| 0x9906 | HD_CODEC_MODE_10_BLEND_THRESHOLD | Blend threshold for Codec Mode 10 | 3 |
| 0x9907 | HD_CODEC_MODE_10_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 10 | 0 |
| 0x9908 | HD_CODEC_MODE_10_BLEND_RATE | Blend Rate for Codec Mode 10 | 1 |
| 0x9909 | HD_CODEC_MODE_13_BLEND_THRESHOLD | Blend threshold for Codec Mode 13 | 3 |
| 0x990A | HD_CODEC_MODE_13_SAMPLES_DELAY | Digital Audio Delay for Codec Mode 13 | 0 |
| 0x990B | HD_CODEC_MODE_13_BLEND_RATE | Blend Rate for Codec Mode 13 | 1 |
| 0xE800 | HD_TEST_BER_CONFIG | Enables the HD BER test. | 0 |

### 4.6.1. AMHD Properties

**Property 0x0000. INT_CTL_ENABLE**

INT_CTL_ENABLE property enables top-level interrupt sources. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTIEN | Reserved | | | | | CTSIEN | ERR_CMDIEN | DACQIEN | DSRVIEN | RSQIEN | Reserved | ACFIEN | STCIEN |
| Default | 0x0 | | 0 | 0x00 | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTIEN | Interrupt when DEVNTINT is set |
| 12:8 | Reserved | Always write to 0x00. |
| 7 | CTSIEN | Interrupt when CTS is set.<br>Note: The default will be 0x1 if POWER_UP:CTSIEN was set. |
| 6 | ERR_CMDIEN | Interrupt when ERR_CMD is set |
| 5 | DACQIEN | Interrupt when DACQINT is set |
| 4 | DSRVIEN | Interrupt when DSRVINT is set |
| 3 | RSQIEN | Interrupt when RSQINT is set |
| 2 | Reserved | Always write to 0. |
| 1 | ACFIEN | Interrupt when ACFINT is set |
| 0 | STCIEN | Interrupt when STCINT is set |

**Property 0x0001. INT_CTL_REPEAT**

INT_CTL_REPEAT is used to set repeat interrupt pulses for a given interrupt even if this particular interrupt was previously generated but not acknowledged. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | DEVNTREP | Reserved | | | | | | | DACQREP | DSRVREP | RSQREP | Reserved | ACFREP | STCREP |
| Default | 0x0 | | 0 | 0x00 | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | DEVNTREP | Repeat interrupt pulse when DEVNTINT is set, even if a previous interrupt was generated but not acknowledged. |
| 12:6 | Reserved | Always write to 0x00. |
| 5 | DACQREP | Repeat interrupt pulse when DACQINT is set, even if a previous interrupt was generated but not acknowledged. |
| 4 | DSRVREP | Repeat interrupt pulse when DSRVINT is set, even if a previous interrupt was generated but not acknowledged. |
| 3 | RSQREP | Repeat interrupt pulse when RSQINT is set, even if a previous interrupt was generated but not acknowledged. |
| 2 | Reserved | Always write to 0. |
| 1 | ACFREP | Repeat interrupt pulse when AFCINT is set, even if a previous interrupt was generated but not acknowledged. |
| 0 | STCREP | Repeat interrupt pulse when STCINT is set, even if a previous interrupt was generated but not acknowledged. |

**Property 0x0200. DIGITAL_IO_OUTPUT_SELECT**

DIGITAL_IO_OUTPUT_SELECT configures the digital audio output to be I2S Master or Slave. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | MASTER | | | | | | | | | | | | | | | |
| **Default** | 0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | MASTER | 0: Slave mode selected<br>1: Master mode selected |

### Property 0x0201. DIGITAL_IO_OUTPUT_SAMPLE_RATE

DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 48000

Units: Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | OUTPUT_SAMPLE_RATE[15:0] | | | | | | | | | | | | | | | |
| Default | 0xBB80 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | OUTPUT_SAMPLE_RATE[15:0] | DIGITAL_IO_OUTPUT_SAMPLE_RATE sets output sample audio rate in units of 1Hz.<br>Range: 32000-48000<br>32000: The minimum output sample rate in Hz.<br>48000: The minimum output sample rate in Hz. |

### Property 0x0202. DIGITAL_IO_OUTPUT_FORMAT

DIGITAL_IO_OUTPUT_FORMAT configures the digital audio output format. This property may only be written before the first tune. Writes after first tune will return error. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x1800

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | SAMPL_SIZE[5:0] | | | | | | SLOT_SIZE[3:0] | | | | FRAMING_FORMAT[3:0] | | | |
| Default | 0x0 | | 0x18 | | | | | | 0x0 | | | | 0x0 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:8 | SAMPL_SIZE[5:0] | Determines the number of bits in a sample. Only the specified number of bits per sample are used. The value of any bits sent over the sample size will be determined by the FILL field of the DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 property. Sample sizes of less then 8 are not supported.<br>Range: 8-24 |
| 7:4 | SLOT_SIZE[3:0] | Determines slot size; must be larger than or equal to SAMPL_SIZE.<br>0x0: Matches Sample Size<br>0x2: 8 bits<br>0x4: 16 bits<br>0x5: 20 bits<br>0x6: 24 bits<br>0x7: 32 bits |
| 3:0 | FRAMING_FORMAT[3:0] | Determines when the data is transmitted relative to frame sync.<br>0x0: I2S mode<br>0x6: DSP mode<br>0x7: Left justified dsp mode<br>0x8: Left justified mode<br>0x9: Right justified mode |

### Property 0x0203. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_1 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | FSLATE_EN | FSINV_EN | RJUST_EN | CLK-INV_EN | SWAP_EN | BIT ORDER_EN | Reserved | | FSLATE | FSINV | RJUST | CLK-INV | SWAP | BIT ORDER |
| Default | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0x0 | | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13 | FSLATE_EN | Enable the FSLATE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 12 | FSINV_EN | Enable the FSINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 11 | RJUST_EN | Enable the RJUST override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 10 | CLKINV_EN | Enable the CLKINV override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 9 | SWAP_EN | Enable the SWAP override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | BITORDER_EN | Enable the BITORDER override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | Reserved | Always write to 0x0. |
| 5 | FSLATE | 0: Sample data on the rising edge of the 2nd clock after DFS<br>1: Sample data on the rising edge of the 1st clock after DFS |
| 4 | FSINV | 0: Use the DFS signal as is<br>1: Invert the DFS signal |
| 3 | RJUST | 0: Left Justified<br>1: Right Justified |
| 2 | CLKINV | Inverts the data clock<br>0: The bit clock is not inverted. DFS will be captured on rising edge of DCLK.<br>1: The bit clock is inverted. DFS will be captured on falling edge of DCLK. |
| 1 | SWAP | Swap position of the left and right channels.<br>0: Transmit the left sample first<br>1: Transmit the right sample first |
| 0 | BITORDER | Determine if the MSB or LSB is transmitted first.<br>0: Transmit MSB first<br>1: Transmit LSB first |

### Property 0x0204. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_2 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | FILL_EN | Reserved | | | | SEQEN_EN | FSEDGE_EN | FILL[1:0] | | Reserved | | | | SEQEN | FSEDGE |
| **Default** | 0 | 0 | 0x0 | | | | 0 | 0 | 0x0 | | 0x0 | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15 | Reserved | Always write to 0. |
| 14 | FILL_EN | Enable the FILL override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 13:10 | Reserved | Always write to 0x0. |
| 9 | SEQEN_EN | Enable the SEQEN override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 8 | FSEDGE_EN | Enable the FSEDGE override value to take effect.<br>0: Use the default value for the specified framing format<br>1: use the value specified in this property |
| 7:6 | FILL[1:0] | Defines the fill value for unused data bits.<br>0: The unused bits are filled with 0's<br>1: The unused bits are filled with 1's<br>2: The unused bits are sign extended<br>3: The unused bits are filled with a random sequence |
| 5:2 | Reserved | Always write to 0x0. |
| 1 | SEQEN | 0: Only one mono sample is sent per trigger of DFS<br>1: Both mono samples are sent sequentially after each trigger from DFS |
| 0 | FSEDGE | 0: One mono sample is sent on only one edge of DFS<br>1: One mono sample is sent every edge (rising and falling) of DFS |

### Property 0x0205. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_3 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSH_EN | FSH[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSH_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSH[14:0] | Number of samples FS will remain high.<br>Range: 0-32767 |

### Property 0x0206. DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4

DIGITAL_IO_OUTPUT_FORMAT_OVERRIDES_4 sets alternate I2S format settings from the standard framing mode. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FSL_EN | FSL[14:0] | | | | | | | | | | | | | | |
| Default | 0 | 0x0000 | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | FSL_EN | Enable the FSH override value to take effect.<br>0: Use the default value for the specified framing format<br>1: Use the value specified in this property |
| 14:0 | FSL[14:0] | Number of samples FS will remain low.<br>Range: 0-32767 |

### Property 0x0300. AUDIO_ANALOG_VOLUME

AUDIO_ANALOG_VOLUME sets the analog audio volume. A value of 0 will mute the audio; a value of 1 applies 62 dB of attenuation, and a value of 63 applies no attenuation. Each step accounts for 1 dB of change in the output. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 63

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | VOL[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x3F | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | VOL[5:0] | Controls the analog volume level.<br>Range: 0-63<br>0: The minimum audio level.<br>63: The maximum audio level. |

### Property 0x0301. AUDIO_MUTE

AUDIO_MUTE property mutes/unmutes each audio output independently. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | MUTE[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | | 0x0 | |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1:0 | MUTE[1:0] | Controls the audio mute on each audio output<br>0: Do not mute audio outputs<br>1: Mute Left Audio Out.<br>2: Mute Right Audio Out.<br>3: Mute both Left and Right Audio Out |

### Property 0x0500. AM_AVC_MIN_GAIN

AM_AVC_MIN_GAIN Sets the minimum gain the AVC can have. The minimum gain value is given by MINGAIN = g * 1024/6.02 or MINGAIN = g * 170 where g is the desired minimum AVC gain in dB. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -2048

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | MINGAIN[15:0] | | | | | | | | | | | | | | | |
| **Default** | 0xF800 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | MINGAIN[15:0] | The minimum gain value for the AVC.<br>Range: -4096-3061—-24dB to +18dB |

### Property 0x0501. AM_AVC_MAX_GAIN

AM_AVC_MAX_GAIN sets the maximum gain the AVC can have. The max gain value is given by MAXGAIN = g * 1024/6.02 or MAXGAIN = g * 170 where g is the desired maximum AVC gain in dB. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10220

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | MAXGAIN[14:0] | | | | | | | | | | | | | | |
| **Default** | 0 | 0x27EC | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15 | Reserved | Always write to 0. |
| 14:0 | MAXGAIN[14:0] | The maximum gain value for the AVC.<br>Range: 0-32767—0 to 193dB |

### Property 0x0800. PIN_CONFIG_ENABLE

PIN_CONFIG is used to enable and disable the various I/O features of the device. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x8001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | INT-BOUTEN | UARTEN | Reserved | | | | | | | | | | | | I2SOUTEN | DACOUTEN |
| Default | 1 | 0 | 0x000 | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15 | INTBOUTEN | When set enables the INTB output. |
| 14 | UARTEN | When set enables the UART I/O. |
| 13:2 | Reserved | Always write to 0x000. |
| 1 | I2SOUTEN | When set enables the I2S the digital audio output. A change to this bit will take effect on the next tune or seek. |
| 0 | DACOUTEN | When set enables the analog audio output. A change to this bit will take effect on the next tune or seek. |

### Property 0x0900. WAKE_TONE_ENABLE

WAKE_TONE_ENABLE is used to enable the wake tone feature. The wake tone feature is a simple alert tone that can be used for various audible alarms such as a wake alarm. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | | | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | ENABLE | Enables the wake tone feature, the wake tone will be played instead of normal audio content. |

### Property 0x0901. WAKE_TONE_PERIOD

WAKE_TONE_PERIOD is used to configure the wake tone feature's on/off period. This property sets the on and off time periods in units of ms. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 250

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | PERIOD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x00FA | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | PERIOD[15:0] | The amount of time the wake tone will be on for each alarm cycle in units of ms. Set to 0 for a continuous tone.<br>Range: 50-2000 |

### Property 0x0902. WAKE_TONE_FREQ

WAKE_TONE_FREQ is the frequency of the wake tone in Hz. The wake tone is a simple square wave whose frequency is defined by this property. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 750

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | FREQ[15:0] | | | | | | | | | | | | | | | |
| Default | 0x02EE | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | FREQ[15:0] | WAKE_TONE_FREQ is the frequency of the wake tone in Hz. The wake tone is a simple square wave whose frequency is defined by this property.<br>Range: 100-4000 |

### Property 0x0903. WAKE_TONE_AMPLITUDE

WAKE_TONE_AMPLITUDE sets the wake tone's output amplitude. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 8

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | AMP[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x08 | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | AMP[4:0] | The linear wake tone amplitude.<br>Range: 0-31 |

### Property 0x2200. AM_CHBW_SQ_LIMITS

Default: 0x1E0F

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | SQ_MAX[7:0] | | | | | | | | SQ_MIN[7:0] | | | | | | | |
| Default | 0x1E | | | | | | | | 0x0F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | SQ_MAX[7:0] | Sets the SNR/RSSI level in dB at which the maximum channel bandwidth will be applied.<br>Range: -128-127 |
| 7:0 | SQ_MIN[7:0] | Sets the SNR/RSSI level in dB at which the minimum channel bandwidth will be applied.<br>Range: -128-127 |

### Property 0x2201. AM_CHBW_SQ_CHBW

Default: 0x2314

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MAX[7:0] | | | | | | | | MIN[7:0] | | | | | | | |
| Default | 0x23 | | | | | | | | 0x14 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | MAX[7:0] | Range: 15-100 |
| 7:0 | MIN[7:0] | Range: 15-100 |

**Property 0x2202. AM_CHBW_SQ_WIDENING_TIME**

Default: 2048

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | WIDENING_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0800 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | WIDENING_-TIME[15:0] | Range: 16-32767 |

**Property 0x2203. AM_CHBW_SQ_NARROWING_TIME**

Default: 16

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | NARROWING_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | NARROWING_-TIME[15:0] | Range: 16-32767 |

**Property 0x2204. AM_CHBW_OVERRIDE_BW**

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | OVERRIDE_BW[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | OVER-RIDE_BW[7:0] | Range: 0-55 |

### Property 0x4100. AM_SEEK_BAND_BOTTOM

AM_SEEK_BAND_BOTTOM sets the lower seek boundary of the AM band in multiples of 1kHz. See AM_SEEK_START. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 520

Units: 1kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | AMSKFREQL[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0208 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | AMSKFREQL[15:0] | AM Seek Band Bottom.<br>Range: 520-1710 - AM |

### Property 0x4101. AM_SEEK_BAND_TOP

AM_SEEK_BAND_TOP sets the upper seek boundary of the AM band in multiples of 1kHz. See AM_SEEK_START. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1710

Units: 1kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | AMSKFREQH[15:0] | | | | | | | | | | | | | | | |
| Default | 0x06AE | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | AMSKFREQH[15:0] | AM Seek Band Top.<br>Range: 520-1710 - AM |

### Property 0x4102. AM_SEEK_FREQUENCY_SPACING

AM_SEEK_FREQUENCY_SPACING sets the frequency spacing for the AM band in multiples of 1kHz when performing a seek. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: 1kHz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | AMSKSPACE[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x0A | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | AMSKSPACE[4:0] | AM Seek Frequency Spacing.<br>Range: 1-31<br>9: AM in Asia (9 kHz)<br>10: AM in US (10 kHz) |

### Property 0x4200. AM_VALID_MAX_TUNE_ERROR

AM_VALID_MAX_TUNE_ERROR sets the maximum freq error allowed in units of bppm before setting the AFC rail indicator (AFCRL). This will take effect on the next tune. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 75

Units: bppm

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | AMMAXTUNEERR[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x4B | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | AMMAX-TUNEERR[7:0] | Frequency error in bppm (2 ppm).<br>Range: 0-126 |

### Property 0x4201. AM_VALID_RSSI_TIME

AM_VALID_RSSI_TIME sets the amount of time in ms to allow the RSSI/ISSI metrics to settle before evaluating. This property is ignored if the FASTTUNE bit(AM_TUNE_FREQ) is set. The reliability of the valid bit for identifying valid stations relies on this parameter being set properly. NOTE: Setting this property to a value lower then 3 will result in the metric not being measured correctly at tune time. To disable metric as a tune qualifier set the validation time to 0 and set the validation threshold to the minimum value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 8

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | SSIVALTIME[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x08 | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | SSIVALTIME[5:0] | Validation time in ms<br>Range: 0-63—Specified in units ms. |

### Property 0x4202. AM_VALID_RSSI_THRESHOLD

Sets the RSSI threshold for a valid AM Seek/Tune. If the desired channel RSSI is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 35

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | AMVALRSSI[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x23 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | AMVALRSSI[7:0] | AM Seek/Tune Received Signal Strength Threshold.<br>Range: -128-127—Specified in units of dBuV in 1 dBuV steps.<br>-128: RSSI is not used as a criterion in determining the validity of a station. |

### Property 0x4203. AM_VALID_SNR_TIME

AM_VALID_SNR_TIME sets the amount of time in ms to allow the SNR metric to settle before evaluating. This property is ignored if the FASTTUNE bit(AM_TUNE_FREQ) is set. The reliability of the valid bit for identifying valid stations relies on this parameter being set properly. NOTE: Setting this property to a value lower then 20 will result in the metric not being measured correctly at tune time. To disable metric as a tune qualifier set the validation time to 0 and set the validation threshold to the minimum value. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 40

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | SNRVALTIME[5:0] | | | | | |
| Default | 0x000 | | | | | | | | | | 0x28 | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:6 | Reserved | Always write to 0x000. |
| 5:0 | SNRVALTIME[5:0] | Validation time in ms<br>Range: 17-63—Specified in units ms. |

### Property 0x4204. AM_VALID_SNR_THRESHOLD

AM_VALID_SNR_THRESHOLD sets the SNR threshold for a valid AM Seek/Tune. If the desired channel SNR is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 10

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | AMVALSNR[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x0A | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | AMVALSNR[7:0] | AM Seek/Tune SNR Threshold.<br>Range: -128-127—Specified in units of dB in 1 dB steps.<br>-128: SNR is not used as a criterion in determining the validity of a station. |

### Property 0x4205. AM_VALID_HDLEVEL_THRESHOLD

Sets the HDLEVEL threshold for AM Seek stop. If the desired channel HDLEVEL threshold is above this threshold, then it is considered valid. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | AMVALHDLEVEL[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | AMVALH-DLEVEL[7:0] | AM Seek/Tune Received Signal Strength Threshold.<br>Range: 0-100—Specified in units of dBuV in 1 dBuV steps. |

### Property 0x4300. AM_RSQ_INTERRUPT_SOURCE

AM_RSQ_INTERRUPT_SOURCE configures interrupt related to Received Signal Quality metrics. See AM_RSQ_STATUS. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | SNRHINT | SNRLINT | RSSIHINT | RSSILINT |
| Default | 0x000 | | | | | | | | | | | | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:4 | Reserved | Always write to 0x000. |
| 3 | SNRHINT | Enables the interrupt for SNR High.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if SNR goes above AM_RSQ_SNR_HIGH_THRESHOLD. |
| 2 | SNRLINT | Enables the interrupt for SNR Low.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if SNR goes below AM_RSQ_SNR_LOW_THRESHOLD. |
| 1 | RSSIHINT | Enables the interrupt for RSSI High.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if RSSI goes above AM_RSQ_RSSI_HIGH_THRESHOLD. |
| 0 | RSSILINT | Enables the interrupt for RSSI Low.<br>0: Disables this interrupt source.<br>1: Enable interrupt to occur if RSSI goes below AM_RSQ_RSSI_LOW_THRESHOLD. |

**Property 0x4301. AM_RSQ_SNR_HIGH_THRESHOLD**

AM_RSQ_SNR_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the SNR is above this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SNRH[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | SNRH[7:0] | AM RSQ SNR High Threshold.<br>Range: -128-127—Specified in units of dB in 1 dB steps. |

### Property 0x4302. AM_RSQ_SNR_LOW_THRESHOLD

AM_RSQ_SNR_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the SNR is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -128

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SNRL[7:0] | | | | | | | |
| Default | 0xFF | | | | | | | | 0x80 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0xFF. |
| 7:0 | SNRL[7:0] | Low threshold for SNR interrupt in dB.<br>Range: -128-127—Specified in units of dB in 1 dB steps. |

### Property 0x4303. AM_RSQ_RSSI_HIGH_THRESHOLD

AM_RSQ_RSSI_HIGH_THRESHOLD sets the high threshold, which triggers the RSQ interrupt if the RSSI is above this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | RSSIH[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x7F | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | RSSIH[7:0] | High threshold for RSSI interrupt in dBuV.<br>Range: -128-127—Specified in units of dBuV in 1 dBuV steps. |

### Property 0x4304. AM_RSQ_RSSI_LOW_THRESHOLD

AM_RSQ_RSSI_LOW_THRESHOLD sets the low threshold, which triggers the RSQ interrupt if the RSSI is below this threshold. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: -128

Units: dBuV

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | RSSIL[7:0] | | | | | | | |
| Default | 0xFF | | | | | | | | 0x80 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0xFF. |
| 7:0 | RSSIL[7:0] | Low threshold for RSSI interrupt in dBuV.<br>Range: -128-127—Specified in units of dBuV in 1 dBuV steps. |

### Property 0x4305. AM_RSQ_HD_DETECTION

Default: 0x8020

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | SAMPLES[7:0] | | | | | | | |
| Default | 0x80 | | | | | | | | 0x20 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x80. |
| 7:0 | SAMPLES[7:0] | Sets the number of HD OFDM symbols to look at during HD Detection. Each HD OFDM symbol is ~5.8ms so the detection time is about 5.8 times this number. Enabling this feature will extend the tune time by this amount minus the SNR and RSSI validation times. Setting this field to 0 disables Fast HD Detect. If HD is enabled, the detection period will be larger of the RSSI or SNR validation times or SAMPLES*6ms.<br>Range: 0-100 |

### Property 0x4400. AM_ACF_INTERRUPT_SOURCE

AM_ACF_INTERRUPT_SOURCE Enables the ACF interrupt sources. When one of the interrupts is enabled, the ACFINT bit of the status word will be set when the controlling indicator crosses the threshold set its ACF threshold property. If the interrupt is acknowlaged and the condition presists, the interrupt will emmideatly repeat. If the ACFINT hardware interrupt is enabled by setting the ACFIEN bit of the INT_CTL_ENABLE, the interrupt line will toggle. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | HIGHCUT_INTEN | SMUTE_INTEN |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | HIGHCUT_INTEN | Enables the Hicut Interrupt<br>0: The Hicut interrupt is disabled<br>1: The Hicut interrupt is enabled |
| 0 | SMUTE_INTEN | Enables the softmute interrupt<br>0: The softmute interrupt is disabled<br>1: The softmute interrupt is enabled |

### Property 0x4401. AM_ACF_SOFTMUTE_THRESHOLD

AM_ACF_SOFTMUTE_THRESHOLD sets the softmute interrupt threshold. When softmute attenuation rises above the level set by this property the SMUTE_INT bit of the AM_ACF_STATUS command will be set. If the SMUTE_INTEN bit of the AM_ACF_INTERRUPT_SOURCE property is set the ACFINT interrupt will be asserted in the status word. See AM_ACF_INTERRUPT_SOURCE for details. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 31

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | SMATTN_THRESH[4:0] | | | | |
| Default | 0x000 | | | | | | | | | | | 0x1F | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | SMATTN_-<br>THRESH[4:0] | Sets the softmute interrupt threshold in dB.<br>Range: 0-31 |

### Property 0x4402. AM_ACF_HIGHCUT_THRESHOLD

AM_ACF_HIGHCUT_THRESHOLD sets the high cut interrupt threshold. When the cutoff frequency falls below this threshold, the HIGHCUT_INT bit of AM_ACF_STATUS command will be asserted. If the high cut interrupt is enabled, the ACFINT bit in the status word will also be asserted. See AM_ACF_INTERRUPT_SOURCE for details. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | HIGHCUT_THRESH[7:0] | | | | | | | |
| **Default** | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HIGHCUT_-<br>THRESH[7:0] | High cut interrupt threshold in units of 100Hz<br>Range: 0-200 |

### Property 0x4403. AM_ACF_SOFTMUTE_TOLERANCE

AM_ACF_SOFTMUTE_TOLERANCE sets the distance from the final softmute value that triggers the softmute convergence flag. Convergence is indicated by setting the SMUTE_CONV flag in the AM_ACF_STATUS command reply. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 2

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | | | | SMUTE_TOL[4:0] | | | | |
| **Default** | 0x000 | | | | | | | | | | | 0x02 | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4:0 | SMUTE_TOL[4:0] | When the softmute value is this many dB away from its final value the SMUTE_-CONV flag will be set.<br>Range: 0-31 |

### Property 0x4404. AM_ACF_HIGHCUT_TOLERANCE

AM_ACF_HIGHCUT_TOLERANCE Sets the distance from the final high cut freq that triggers the high cut convergence flag. Convergence is indicated by a setting HIGHCUT_CONV flag of AM_ACF_STATUS command reply. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 20

Units: 100Hz

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | HIGHCUT_TOL[7:0] | | | | | | | |
| **Default** | 0x00 | | | | | | | | 0x14 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HIGH-CUT_TOL[7:0] | When the high cut state is N*100Hz away from its final value the HIGHCUT_CONV flag will be set.<br>Range: 0-200 |

### Property 0x4405. AM_ACF_CONTROL_SOURCE

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | | | | | | | | | | | AFC_DIS | AFC_SM | Reserved | USE_RSSI |
| **Default** | 0x000 | | | | | | | | | | | | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:4 | Reserved | Always write to 0x000. |
| 3 | AFC_DIS | When set, disables the use of AFC in tune validation.<br>0: Use AFC in the tune validation processes set by AM_VALID_MAX_TUNE_ERROR<br>1: Disable the use of AFC for tune validation |
| 2 | AFC_SM | When set, softmute will be triggered by an AFC rail and force maximum AM_SOFTMUTE_SQ_ATTENUATION.<br>0: Do not use AFC rail to force a softmute.<br>1: Use AFC rail to force a softmute. |
| 1 | Reserved | Always write to 0. |
| 0 | USE_RSSI | When set, RSSI will be used instead of SNR as the controlling metric.<br>0: Use SNR as the controlling metric.<br>1: Use RSSI as the controlling metric. |

**Property 0x4500. AM_SOFTMUTE_SQ_LIMITS**

AM_SOFTMUTE_SQ_LIMITS sets the SNR limits for soft mute attenuation. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0800

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | XMAX[7:0] | | | | | | | | XMIN[7:0] | | | | | | | |
| **Default** | 0x08 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | XMAX[7:0] | Value of SNR in dB which causes the minimum audio attenuation.<br>Range: -20-64 |
| 7:0 | XMIN[7:0] | Value of SNR in dB which causes the maximum audio attenuation.<br>Range: -20-64 |

### Property 0x4501. AM_SOFTMUTE_SQ_ATTENUATION

AM_SOFTMUTE_SQ_ATTENUATION sets the softmute attenuation limits. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x000C

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTENMIN[7:0] | | | | | | | | ATTENMAX[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x0C | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | ATTENMIN[7:0] | Sets the minimum attenuation amount in dB for SNR/RSSI based softmute.<br>Range: 0-31 |
| 7:0 | ATTENMAX[7:0] | Sets the maximum attenuation amount in dB for SNR/RSSI based softmute.<br>Range: 0-31 |

### Property 0x4502. AM_SOFTMUTE_SQ_ATTACK_TIME

AM_SOFTMUTE_SQ_ATTACK_TIME sets the attack time to mute the audio. The attack time is the time it takes the softmute attenuation to go from YMIM to YMAX if the SNR made a step change from XMAX to XMIN. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 120

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0078 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | The softmute attack time in ms for SNR based softmute.<br>Range: 16-65535 |

### Property 0x4503. AM_SOFTMUTE_SQ_RELEASE_TIME

AM_SOFTMUTE_SQ_RELEASE_TIME Sets the release time to unmute the audio. The release time is the time it takes the softmute attenuation to go from YMAX to YMIN if the SNR made a step change from XMIN to XMAX. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 500

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x01F4 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | The softmute release time in ms for SNR based softmute.<br>Range: 16-65535 |

### Property 0x4600. AM_HIGHCUT_SQ_LIMITS

Default: 0x0a06

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | SQ_MAX[7:0] | | | | | | | | SQ_MIN[7:0] | | | | | | | |
| Default | 0x0A | | | | | | | | 0x06 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | SQ_MAX[7:0] | Value of SNR in Hz which causes the minimum audio attenuation.<br>Range: -20-64 |
| 7:0 | SQ_MIN[7:0] | Value of SNR in Hz which causes the maximum audio attenuation.<br>Range: -20-64 |

### Property 0x4601. AM_HIGHCUT_SQ_CUTOFF_FREQ

Default: 0x280A

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | MAX[7:0] | | | | | | | | MIN[7:0] | | | | | | | |
| Default | 0x28 | | | | | | | | 0x0A | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | MAX[7:0] | Range: 10-50 |
| 7:0 | MIN[7:0] | Range: 10-50 |

**Property 0x4602. AM_HIGHCUT_SQ_ATTACK_TIME**

Default: 16

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ATTACK[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0010 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ATTACK[15:0] | Range: 1-32767 |

**Property 0x4603. AM_HIGHCUT_SQ_RELEASE_TIME**

Default: 2000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RELEASE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x07D0 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RELEASE[15:0] | Range: 1-32767 |

**Property 0x4800. AM_DEMOD_AFC_RANGE**

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | RANGE[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | RANGE[15:0] | Range: 0-6000<br>0: The AFC range is determined by AM_VALID_MAX_TUNE_ERROR |

### Property 0x8100. DIGITAL_SERVICE_INT_SOURCE

DIGITAL_SERVICE_INT_SOURCE configures which digital service events will set the DSRVINT status bit. When one of the bits decribed below is set, the corresponding event will cause the DSRVINT bit of the status word to be set. To clear the DSRVINT bit the GET_DIGITAL_SERVICE_DATA command must be executed with the ack option. If the DSRVIEN interrupt is enabled, a host interrupt will also be generated. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0000

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | DSRVOVFLINT | DSRVPCKTINT |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | DSRVOVFLINT | Configures the data service overflow interrupt. When this interrupt occurs data service data has been lost and indicates that the host is not retrieving the data service data in a timely fashion.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 0 | DSRVPCKTINT | Enables the DSRVPCKTINT interrupt of the GET_DIGITAL_SERVICE_DATA command.<br>0: Interrupt disabled<br>1: Interrupt enabled |

### Property 0x9101. HD_BLEND_OPTIONS

HD_BLEND_OPTIONS provides options to control HD/analog audio blend behavior. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0002

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | BLEND_PIN_CTRL | ACQ_LOSS[1:0] | |
| Default | 0x0000 | | | | | | | | | | | | | 0 | 0x2 | |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2 | BLEND_PIN_CTRL | Blend pin control.<br>This option is for host to control the state of the blend pin. Changes take effect upon HD_ACQUIRE.<br>0: Force the blend pin to high (automatically blend mode).<br>1: Force the blend pin to low (force analog blend mode). |
| 1:0 | ACQ_LOSS[1:0] | Upon digital acquisition loss, blend to silence or analog.<br>This option only takes effect if using hybrid tune mode.<br>0: Always blend to analog: Acq. loss, blend to analog, reacquire, blend to previously selected service.<br>1: Always blend to silence: Acq. loss, blend to silence, reacquire, blend to previously selected service.<br>2: Service dependent blend: MPS Selected: Acq. loss, blend to analog, reacquire, blend to MPS. SPSn Selected: Acq. loss, blend to silence, reacquire, blend to previous SPSn. |

### Property 0x9102. HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME

HD_BLEND_ANALOG_TO_HD_TRANSITION_TIME sets the amount of time it takes in ms to blend from analog to HD. This property only applies to primary service channel. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 750

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BLEND_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x02EE | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | BLEND_TIME[15:0] | The amount of time in ms it takes blend from analog reception to HD reception. |

### Property 0x9103. HD_BLEND_HD_TO_ANALOG_TRANSITION_TIME

HD_BLEND_HD_TO_ANALOG_TRANSITION_TIME sets the amount of time it takes in ms to blend from HD to analog. This property only applies to primary service channel. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 100

Units: ms

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | BLEND_TIME[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0064 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | BLEND_TIME[15:0] | The amount of time in ms it takes blend from analog reception to HD reception. |

### Property 0x9106. HD_BLEND_DYNAMIC_GAIN

HD_BLEND_DYNAMIC_GAIN sets the digital audio dynamic linear scaling factor. Setting this property to non-zero will override the broadcaster specified digital gain. The property is a signed 8 bit fractional number where the linear gain is equal to DGAIN/128. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: Q0.7

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | DGAIN[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x00 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | DGAIN[7:0] | HD audio linear scaling factor (Q0.7) format.<br>Range: -128-127 |

### Property 0x9200. HD_DIGRAD_INTERRUPT_SOURCE

HD_DIGRAD_INTERRUPT_SOURCE configures interrupts related to digital receiver (HD_DIGRAD_STATUS). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HDLOGOINTEN | SRCANAINTEN | SRCDIGINTEN | Reserved | | ACQINTEN | CDNRHINTEN | CDNRLINTEN |
| Default | 0x00 | | | | | | | | 0 | 0 | 0 | 0x0 | | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | HDLOGOINTEN | Enables the HD logo interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 6 | SRCANAINTEN | Enables the analog audio source interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 5 | SRCDIGINTEN | Enables the digital audio source interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 4:3 | Reserved | Always write to 0x0. |
| 2 | ACQINTEN | Ensemble acquisition change interrupt enable.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 1 | CDNRHINTEN | Enables the CDNR high interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |
| 0 | CDNRLINTEN | Enables the CDNR low interrupt.<br>0: Interrupt disabled<br>1: Interrupt enabled |

### Property 0x9201. HD_DIGRAD_CDNR_LOW_THRESHOLD

HD_DIGRAD_CDNR_LOW_THRESHOLD sets the CDNR level (in dB) below which the CDNRLINT interrupt will occur. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CDNR_LOW_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x0000 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | CDNR_LOW_-THRESHOLD[15:0] | This property sets the CDNR level (in dB) below which the CDNRLINT interrupt will occur. To make this interrupt set the DACQINT in the status word the CDN-RLINTEN bit must be set in the HD_DIGRAD_INTERRUPT_SOURCE property. |

### Property 0x9202. HD_DIGRAD_CDNR_HIGH_THRESHOLD

HD_DIGRAD_CDNR_HIGH_THRESHOLD sets the CDNR level (in dB) above which the CDNRHINT interrupt will occur. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 127

Units: dB

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | CDNR_HIGH_THRESHOLD[15:0] | | | | | | | | | | | | | | | |
| Default | 0x007F | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | CDNR_HIGH_-THRESHOLD[15:0] | This property sets the CDNR level (in dB) above which the CDNRHINT interrupt will occur. To make this interrupt set the DACQINT in the status word the CDN-RHINTEN bit must be set in the HD_DIGRAD_INTERRUPT_SOURCE property. |

### Property 0x9203. HD_DIGRAD_AUTO_ACQUIRE

HD_DIGRAD_AUTO_ACQUIRE sets the radio to automatically acquire digital radio. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | AUTO_ACQUIRE |
| Default | 0x0000 | | | | | | | | | | | | | | | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | AUTO_ACQUIRE | This option controls automatic acquire.<br>0: Do not acquire digital radio after tune. HD_ACQUIRE must be called to start the acquisition process.<br>1: Automatically acquire digital radio after tune. |

### Property 0x9300. HD_EVENT_INTERRUPT_SOURCE

HD_EVENT_INTERRUPT_SOURCE property configures interrupts related to HD Events (see DEVENTINT status bit). The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | DINFO_INTEN | AINFO_INTEN | Reserved | ALERT_INTEN | PSD_INTEN | SIS_INTEN | DSRVLIST_INTEN | ASRVLIST_INTEN |
| Default | 0x00 | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7 | DINFO_INTEN | Enables the Data Info interrupt.<br>0: Disables the Data Info interrupt.<br>1: Enable interrupt to occur whenever a updated data info is availabe. |
| 6 | AINFO_INTEN | Enables the Audio Info interrupt.<br>0: Disables the Audio Info interrupt.<br>1: Enable interrupt to occur whenever a updated audio info is availabe. |
| 5 | Reserved | Always write to 0. |
| 4 | ALERT_INTEN | Enables the Alert interrupt.<br>0: Disables the Alert interrupt.<br>1: Enable interrupt to occur whenever a new Alert occurs. |
| 3 | PSD_INTEN | Enables the Program Service Data (PSD) interrupt.<br>0: Disables the PSD interrupt.<br>1: Enable interrupt to occur whenever PSD is available. |
| 2 | SIS_INTEN | Enables the Station Information (SIS) interrupt.<br>0: Disables the SIS interrupt.<br>1: Enable interrupt to occur whenever SIS is availabe. |
| 1 | DSRVLIST_INTEN | Enables the data service list event interrupt.<br>0: Disables the service list update interrupt.<br>1: Enable interrupt to occur whenever a data service list update occurs. |
| 0 | ASRVLIST_INTEN | Enables the audio service list event interrupt.<br>0: Disables the audio service list update interrupt.<br>1: Enable interrupt to occur whenever an audio service list update occurs. |

**Property 0x9301. HD_EVENT_SIS_CONFIG**

HD_EVENT_SIS_CONFIG configures which basic SIS information is returned by the HD_GET_STATION_INFO command BASICSIS option. Takes effect at tune time. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0017

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | LOCATION | RSVD | NAME_LF | NAME_SF | ID |
| Default | 0x000 | | | | | | | | | | | 1 | 0 | 1 | 1 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:5 | Reserved | Always write to 0x000. |
| 4 | LOCATION | When set the station location will be returned when BASICSIS option is used in the HD_GET_STATION_INFO command. |
| 3 | RSVD | Reserved |
| 2 | NAME_LF | When set the long form station name will be returned when BASICSIS option is used with the HD_GET_STATION_INFO command. |
| 1 | NAME_SF | When set the short form station name will be returned when BASICSIS option is used with the HD_GET_STATION_INFO command. |
| 0 | ID | When set the station ID will be returned when BASICSIS option is used in the HD_GET_STATION_INFO command. |

### Property 0x9302. HD_EVENT_ALERT_CONFIG

HD_EVENT_ALERT_CONFIG configures HD alerts. Alert information is returned by the HD_GET_ALERT_MSG command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0x0001

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | PLAY_TONE | ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | | 0 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:2 | Reserved | Always write to 0x0000. |
| 1 | PLAY_TONE | When set the HD radio will automattically insert a 2 second tone into the audio every time an alert is sent by the broadcaster. This can be quite often so it is recommended that the host use the HD_PLAY_ALERT_TONE command to play a tone each time a unique alert message is received. |
| 0 | ENABLE | When set the Alert system is enabled. Please see the HD_GET_ALERT_MSG command and the HD_GET_EVENT_STATUS commands for more details about alerts. |

**Property 0x9500. HD_PSD_ENABLE**

HD_PSD_ENABLE sets which audio services will provide program service data. The PSD data is forwarded through the data service DSRV interface. Once PSD data is available it is forwarded automatically for each service that is enabled. The PSD data packet is formatted as a collection of ID3 tags that must be decoded by the host. Please refer to the DSRV Users guide and the iBiquity Advanced Applications Services Guide for more information on using the DSRV and PSD information. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ASPS7 | ASPS6 | ASPS5 | ASPS4 | ASPS3 | ASPS2 | ASPS1 | AMPS | SPS7 | SPS6 | SPS5 | SPS4 | SPS3 | SPS2 | SPS1 | MPS |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| Bit | Name | Function |
|---|---|---|
| 15 | ASPS7 | Always enable program services data for Secondary Program Service 7. This PSD will be forwarded with a CID of 0x5207. |
| 14 | ASPS6 | Always enable program services data for Secondary Program Service 6. This PSD will be forwarded with a CID of 0x5206. |
| 13 | ASPS5 | Always enable program services data for Secondary Program Service 5. This PSD will be forwarded with a CID of 0x5205. |
| 12 | ASPS4 | Always enable program services data for Secondary Program Service 4. This PSD will be forwarded with a CID of 0x5204. |
| 11 | ASPS3 | Always enable program services data for Secondary Program Service 3. This PSD will be forwarded with a CID of 0x5203. |
| 10 | ASPS2 | Always enable program services data for Secondary Program Service 2. This PSD will be forwarded with a CID of 0x5202. |
| 9 | ASPS1 | Always enable program services data for Secondary Program Service 1. This PSD will be forwarded with a CID of 0x5201. |
| 8 | AMPS | Always enable program services data for Main Program Service. This PSD will be forwarded with a CID of 0x5100. |
| 7 | SPS7 | Enable program services data for Secondary Program Service 7 when tuned to this service. This PSD will be forwarded with a CID of 0x5207. |
| 6 | SPS6 | Enable program services data for Secondary Program Service 6 when tuned to this service. This PSD will be forwarded with a CID of 0x5206. |
| 5 | SPS5 | Enable program services data for Secondary Program Service 5 when tuned to this service. This PSD will be forwarded with a CID of 0x5205. |
| 4 | SPS4 | Enable program services data for Secondary Program Service 4 when tuned to this service. This PSD will be forwarded with a CID of 0x5204. |
| 3 | SPS3 | Enable program services data for Secondary Program Service 3 when tuned to this service. This PSD will be forwarded with a CID of 0x5203. |
| 2 | SPS2 | Enable program services data for Secondary Program Service 2 when tuned to this service. This PSD will be forwarded with a CID of 0x5202. |

| Bit | Name | Function |
|---|---|---|
| 1 | SPS1 | Enable program services data for Secondary Program Service 1 when tuned to this service. This PSD will be forwarded with a CID of 0x5201. |
| 0 | MPS | Enable program services data for the Main Program Service when tuned to this service. This PSD will be forwarded with a CID of 0x5100. |

**Property 0x9501. HD_PSD_FIELD_MASK**

Default: 0xFFFF

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | ID | OWNER | DESC | NAME | RECV | URL | VALID | PRICE | Reserved | TEXT | SHORT | LANG | GENRE | ALBUM | ARTIST | TITLE |
| Default | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

| Bit | Name | Function |
|---|---|---|
| 15 | ID | UFID: Owner ID |
| 14 | OWNER | UFID: Owner |
| 13 | DESC | Commercial: Description |
| 12 | NAME | Commercial: Name of Seller |
| 11 | RECV | Commercial: Received As |
| 10 | URL | Commercial: Contact URL |
| 9 | VALID | Commercial: Valid Until |
| 8 | PRICE | Commercial: Price |
| 7 | Reserved | Always write to 1. |
| 6 | TEXT | Comment: The actual text |
| 5 | SHORT | Comment: Short Content Description |
| 4 | LANG | Comment: Language |
| 3 | GENRE | Genre |
| 2 | ALBUM | Album |
| 1 | ARTIST | Artist |
| 0 | TITLE | Title |

### Property 0x9701. HD_AUDIO_CTRL_PROGRAM_LOSS_THRESHOLD

HD_AUDIO_CTRL_PROGRAM_LOSS_THRESHOLD controls the duration before reverting to MPS audio after an SPS audio program is removed or lost. The same value applies to all SPS audio programs. The timeout is calculated as follows: Timeout = Value X 1.486 seconds. To hold the SPS audio program indefinitely, a value of 0 should be used. In this case, if the SPS audio program is lost or terminated, the system will not automatically revert to MPS and the SPS audio output will be muted and will continue to dwell on the selected supplemental program until commanded otherwise by the HC/listener or until automatic re-acquisition occurs. The system will perform this automatic re-acquisition when its internal HD acquired parameter is false for 60 seconds. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: frames

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | TRESH[3:0] | | | |
| Default | 0x000 | | | | | | | | | | | | 0x0 | | | |

| Bit | Name | Function |
|---|---|---|
| 15:4 | Reserved | Always write to 0x000. |
| 3:0 | TRESH[3:0] | The delay before reverting to MPS audio.<br>Range: 0-14 |

### Property 0x9702. HD_AUDIO_CTRL_BALL_GAME_ENABLE

HD_AUDIO_CTRL_BALL_GAME_ENABLE selects the audio output for hybrid waveforms when the TX Blend Control Status (BCTL_EN of HD_DIGRAD_STATUS) bits are set to 01 (i.e., ballgame mode). Since analog diversity delay is not applied by the transmitter in this state, the receiver must disable audio blending and force either analog or digital audio to avoid audible discontinuities. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | MODE |
| Default | 0x0000 | | | | | | | | | | | | | | | 1 |

| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | MODE | When 1 selects Ball Game mode.<br>Range: 0-1 |

### Property 0x9900. HD_CODEC_MODE_0_BLEND_THRESHOLD

HD_CODEC_MODE_0_BLEND_THRESHOLD sets the threshold for determining when to blend between the digital HD stream and the analog stream for codec mode 0. The same threshold applies to all audio programs that utilize codec mode 0. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 3

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | LEVEL[2:0] | | |
| Default | 0x0000 | | | | | | | | | | | | | 0x3 | | |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2:0 | LEVEL[2:0] | 0: No blending. Always output analog audio<br>1: Blend to analog the earliest<br>2: Blend to analog the second earliest<br>3: Blend to analog the third earliest<br>4: Stay in digital the longest<br>7: No blending—always output digital audio |

### Property 0x9901. HD_CODEC_MODE_0_SAMPLES_DELAY

HD_CODEC_MODE_0_SAMPLES_DELAY is used to perform fine time alignment between the digital audio and analog audio to ensure smooth blending. This property should match the expected system delay for the analog demod process. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| Default | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x9902. HD_CODEC_MODE_0_BLEND_RATE

HD_CODEC_MODE_0_BLEND_RATE configures the hysteresis in the blending process. Blend hysteresis has two main components affected by this property; a step size for the analog hold duration, and the digital duration required for state reset. Each time the system blends to analog, the amount of time the system will hold in analog mode in s is incremented by one step (HOLD). The system will remain in analog mode for this analog hold duration regardless of the return of digital demodulation success. For example: with a step size (HOLD) of 5, the analog hold duration steps would be 1.1 s, 6.1 s, 11.1 s, with a maximum hold duration of (1.1 + 5 * HOLD). The analog hold duration will be reset to its lowest value (1.1 s) whenever the system remains in digital mode longer than HOLD s. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

Units: s

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HOLD[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x01 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HOLD[7:0] | Both the step size for the analog hold duration and the digital duration required for state reset, both in s.<br>Range: 1-8 |

### Property 0x9903. HD_CODEC_MODE_2_BLEND_THRESHOLD

HD_CODEC_MODE_2_BLEND_THRESHOLD sets the threshold for determining when to blend between the digital HD stream and the analog stream for codec mode 2. The same threshold applies to all audio programs that utilize codec mode 2. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 3

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | LEVEL[2:0] | | |
| Default | 0x0000 | | | | | | | | | | | | | 0x3 | | |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2:0 | LEVEL[2:0] | 0: No blending. Always output analog audio<br>1: Blend to analog the earliest<br>2: Blend to analog the second earliest<br>3: Blend to analog the third earliest<br>4: Stay in digital the longest<br>7: No blending—always output digital audio |

### Property 0x9904. HD_CODEC_MODE_2_SAMPLES_DELAY

HD_CODEC_MODE_2_SAMPLES_DELAY is used to perform fine time alignment between the digital audio and analog audio to ensure smooth blending. This property should match the expected system delay for the analog demod process. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 7398

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Name** | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| **Default** | 0x0 | | 0x1CE6 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x9905. HD_CODEC_MODE_2_BLEND_RATE

HD_CODEC_MODE_2_BLEND_RATE configures the hysteresis in the blending process. Blend hysteresis has two main components affected by this property; a step size for the analog hold duration, and the digital duration required for state reset. Each time the system blends to analog, the amount of time the system will hold in analog mode in s is incremented by one step (HOLD). The system will remain in analog mode for this analog hold duration regardless of the return of digital demodulation success. For example: with a step size (HOLD) of 5, the analog hold duration steps would be 1.1 s, 6.1 s, 11.1 s, with a maximum hold duration of (1.1 + 5 * HOLD). The analog hold duration will be reset to its lowest value (1.1 s) whenever the system remains in digital mode longer than HOLD s. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

Units: s

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HOLD[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x01 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HOLD[7:0] | Both the step size for the analog hold duration and the digital duration required for state reset, both in s.<br>Range: 1-8 |

### Property 0x9906. HD_CODEC_MODE_10_BLEND_THRESHOLD

HD_CODEC_MODE_10_BLEND_THRESHOLD sets the threshold for determining when to blend between the digital HD stream and the analog stream for codec mode 10. The same threshold applies to all audio programs that utilize codec mode 10. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 3

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | LEVEL[2:0] | | |
| Default | 0x0000 | | | | | | | | | | | | | 0x3 | | |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2:0 | LEVEL[2:0] | 0: No blending. Always output analog audio<br>1: Blend to analog the earliest<br>2: Blend to analog the second earliest<br>3: Blend to analog the third earliest<br>4: Stay in digital the longest<br>7: No blending—always output digital audio |

### Property 0x9907. HD_CODEC_MODE_10_SAMPLES_DELAY

HD_CODEC_MODE_10_SAMPLES_DELAY is used to perform fine time alignment between the digital audio and analog audio to ensure smooth blending. This property should match the expected system delay for the analog demod process. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| Default | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

**Property 0x9908. HD_CODEC_MODE_10_BLEND_RATE**

HD_CODEC_MODE_10_BLEND_RATE configures the hysteresis in the blending process. Blend hysteresis has two main components affected by this property; a step size for the analog hold duration, and the digital duration required for state reset. Each time the system blends to analog, the amount of time the system will hold in analog mode in s is incremented by one step (HOLD). The system will remain in analog mode for this analog hold duration regardless of the return of digital demodulation success. For example: with a step size (HOLD) of 5, the analog hold duration steps would be 1.1 s, 6.1 s, 11.1 s, with a maximum hold duration of (1.1 + 5 * HOLD). The analog hold duration will be reset to its lowest value (1.1 s) whenever the system remains in digital mode longer than HOLD s. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

Units: s

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HOLD[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x01 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HOLD[7:0] | Both the step size for the analog hold duration and the digital duration required for state reset, both in s.<br>Range: 1-8 |

### Property 0x9909. HD_CODEC_MODE_13_BLEND_THRESHOLD

HD_CODEC_MODE_13_BLEND_THRESHOLD sets the threshold for determining when to blend between the digital HD stream and the analog stream for codec mode 13. The same threshold applies to all audio programs that utilize codec mode 13. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 3

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | LEVEL[2:0] | | |
| Default | 0x0000 | | | | | | | | | | | | | 0x3 | | |

| Bit | Name | Function |
|---|---|---|
| 15:3 | Reserved | Always write to 0x0000. |
| 2:0 | LEVEL[2:0] | 0: No blending. Always output analog audio<br>1: Blend to analog the earliest<br>2: Blend to analog the second earliest<br>3: Blend to analog the third earliest<br>4: Stay in digital the longest<br>7: No blending—always output digital audio |

### Property 0x990A. HD_CODEC_MODE_13_SAMPLES_DELAY

HD_CODEC_MODE_13_SAMPLES_DELAY is used to perform fine time alignment between the digital audio and analog audio to ensure smooth blending. This property should match the expected system delay for the analog demod process. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 0

Units: audio samples

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | COUNT[13:0] | | | | | | | | | | | | | |
| Default | 0x0 | | 0x0000 | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:14 | Reserved | Always write to 0x0. |
| 13:0 | COUNT[13:0] | The amount to delay the digital audio stream playback in audio samples. |

### Property 0x990B. HD_CODEC_MODE_13_BLEND_RATE

HD_CODEC_MODE_13_BLEND_RATE configures the hysteresis in the blending process. Blend hysteresis has two main components affected by this property; a step size for the analog hold duration, and the digital duration required for state reset. Each time the system blends to analog, the amount of time the system will hold in analog mode in s is incremented by one step (HOLD). The system will remain in analog mode for this analog hold duration regardless of the return of digital demodulation success. For example: with a step size (HOLD) of 5, the analog hold duration steps would be 1.1 s, 6.1 s, 11.1 s, with a maximum hold duration of (1.1 + 5 * HOLD). The analog hold duration will be reset to its lowest value (1.1 s) whenever the system remains in digital mode longer than HOLD s. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

Default: 1

Units: s

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | HOLD[7:0] | | | | | | | |
| Default | 0x00 | | | | | | | | 0x01 | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:8 | Reserved | Always write to 0x00. |
| 7:0 | HOLD[7:0] | Both the step size for the analog hold duration and the digital duration required for state reset, both in s.<br>Range: 1-8 |

### Property 0xE800. HD_TEST_BER_CONFIG

HD_TEST_BER_CONFIG Enables the HD BER test. The HD BER test requires a special test vector (IB_FMr208c_e1wfc204 for FMHD, IB_AMr208a_e1awfb00 for AMHD). To run this test start the vector signal generator with the BER test vector, tune to the frequency indicated on the generator, and enable the test using this property. Once the test is running collect the BER information using the HD_TEST_GET_BER_INFO command. The CTS bit (and optional interrupt) is set when it is safe to send the next command. This property may only be set or read in powerup mode.

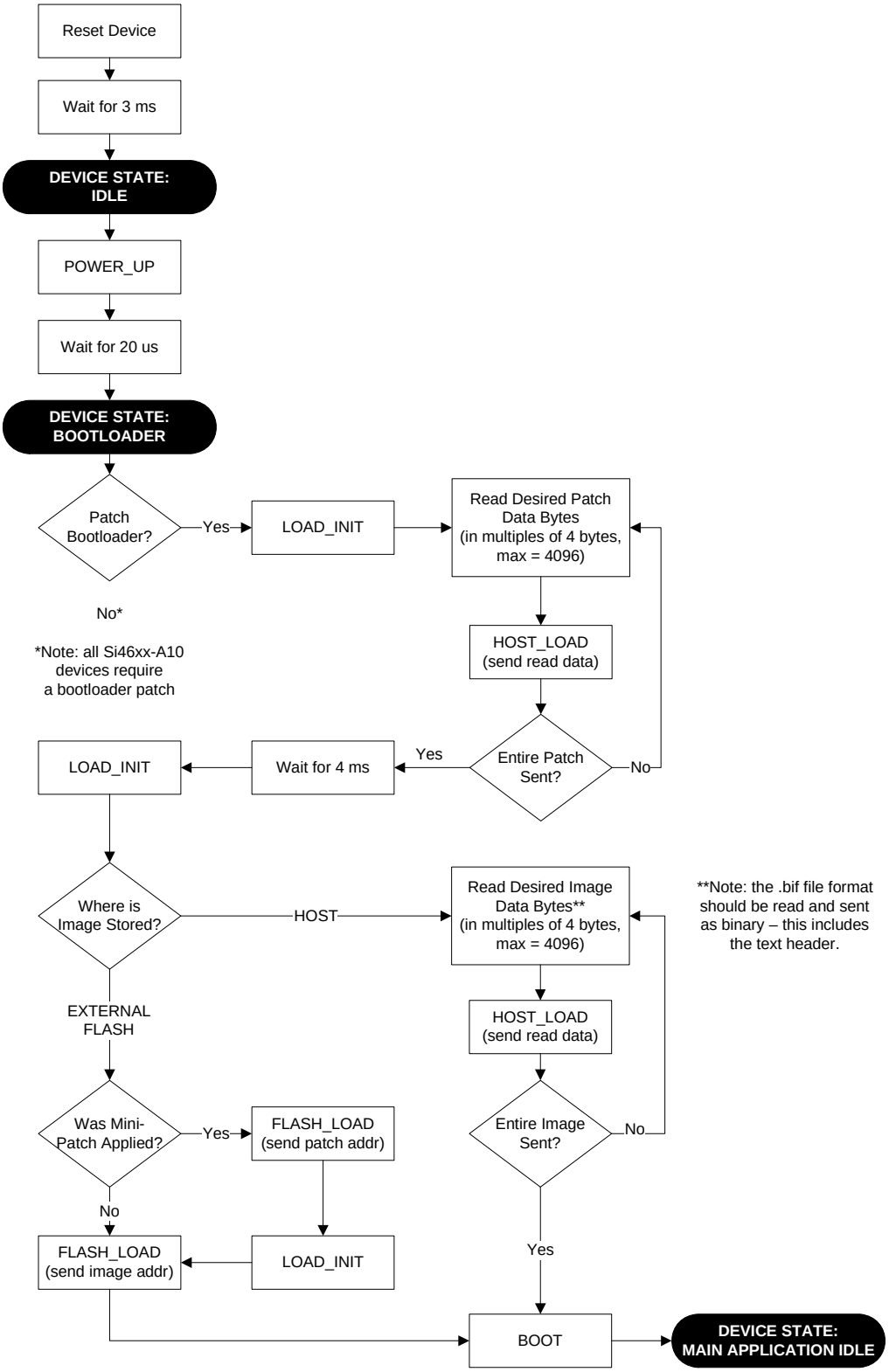
Default: 0

| Bit | D15 | D14 | D13 | D12 | D11 | D10 | D9 | D8 | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | Reserved | | | | | | | | | | | | | | | ENABLE |
| Default | 0x0000 | | | | | | | | | | | | | | | 0 |

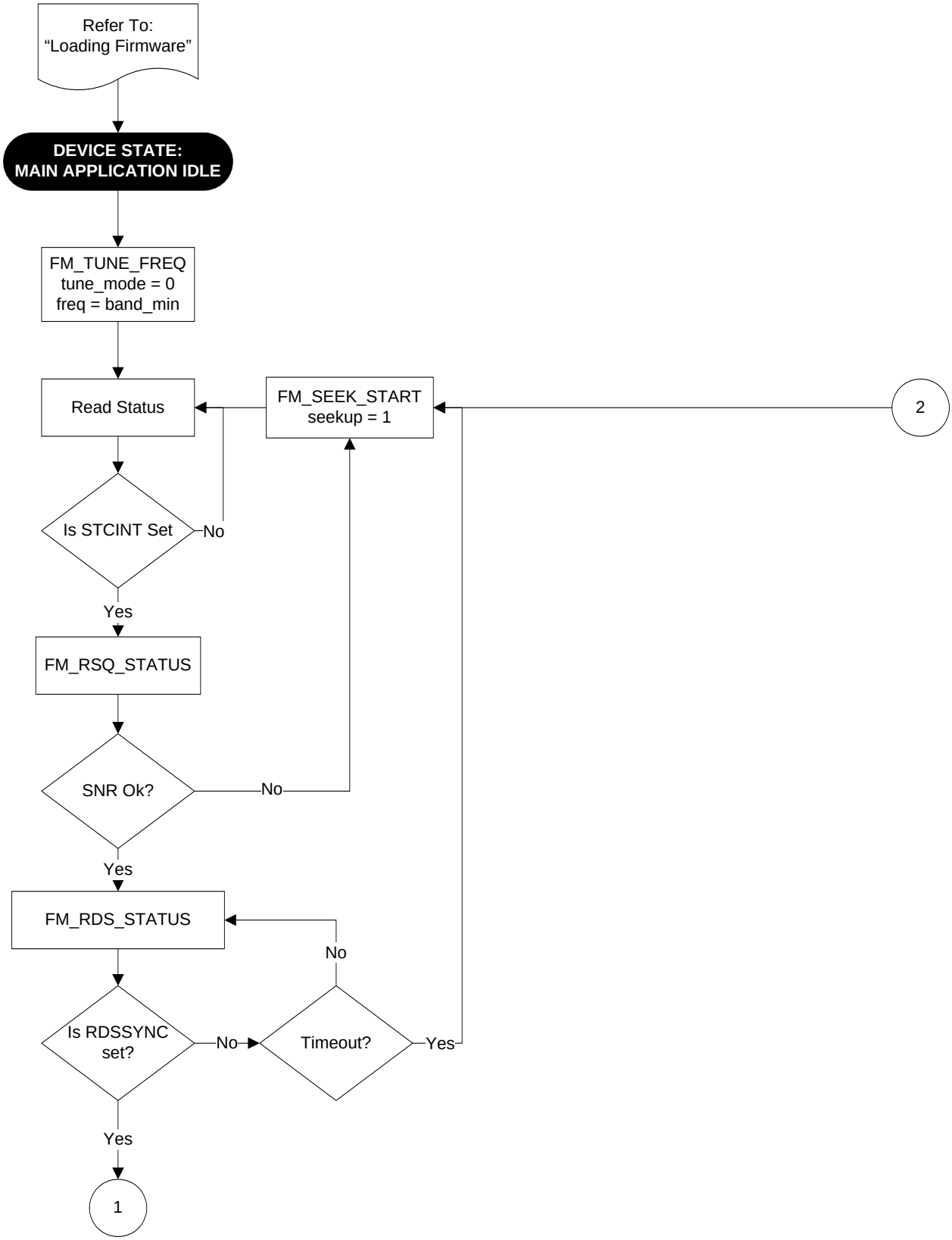
| Bit | Name | Function |
|---|---|---|
| 15:1 | Reserved | Always write to 0x0000. |
| 0 | ENABLE | Enables the BER test. Once enabled the BER information can be collected using the HD_TEST_GET_BER_INFO command. |

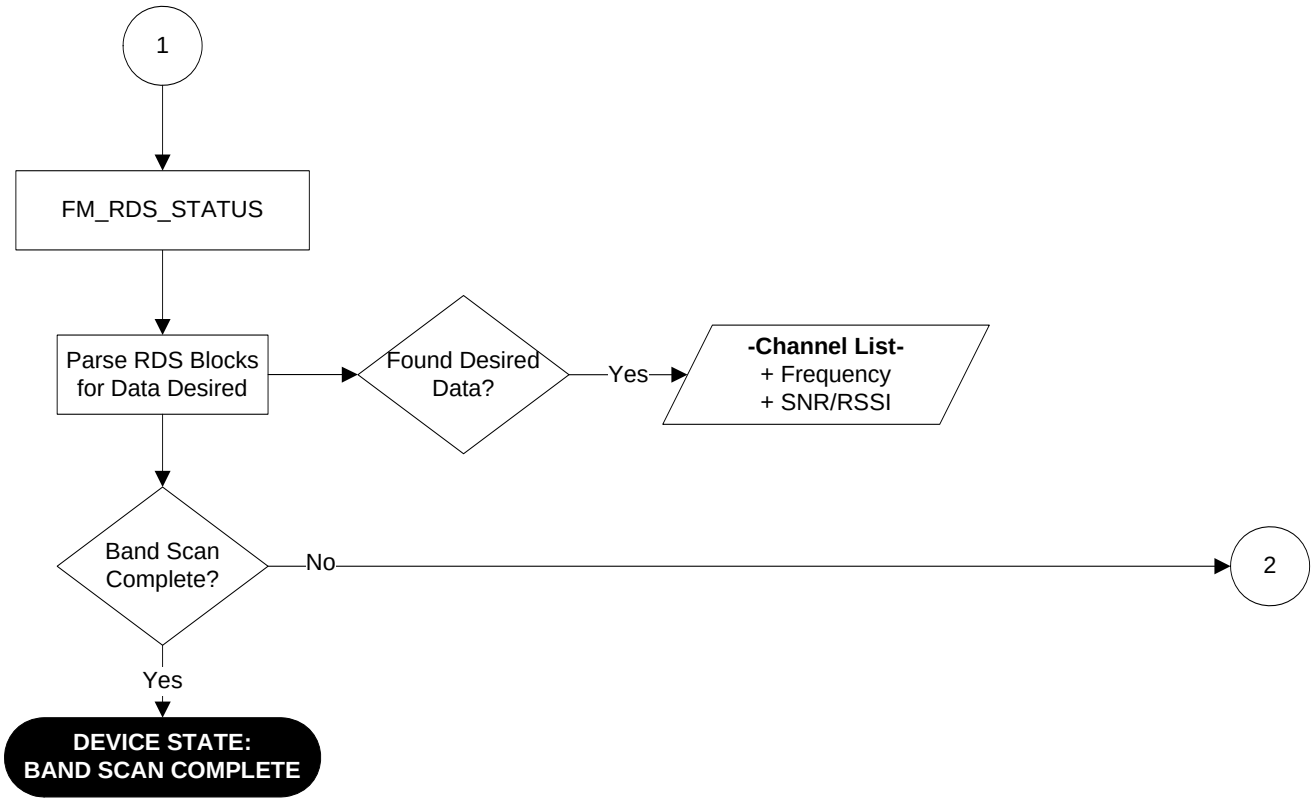
# 5. Flowcharts

## 5.1. Loading Firmware

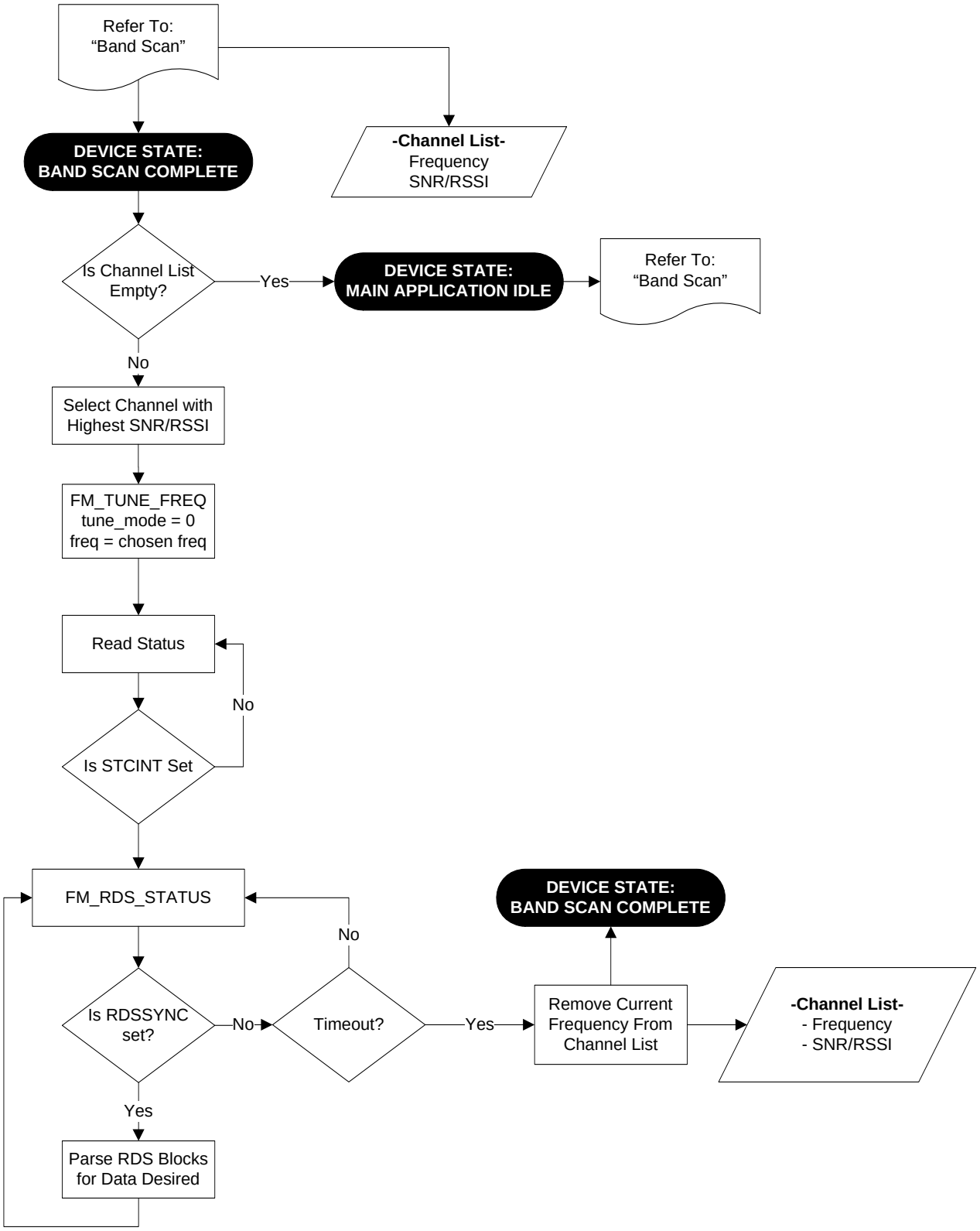


## 5.2. Data Service Receiver—FM–Band Scan

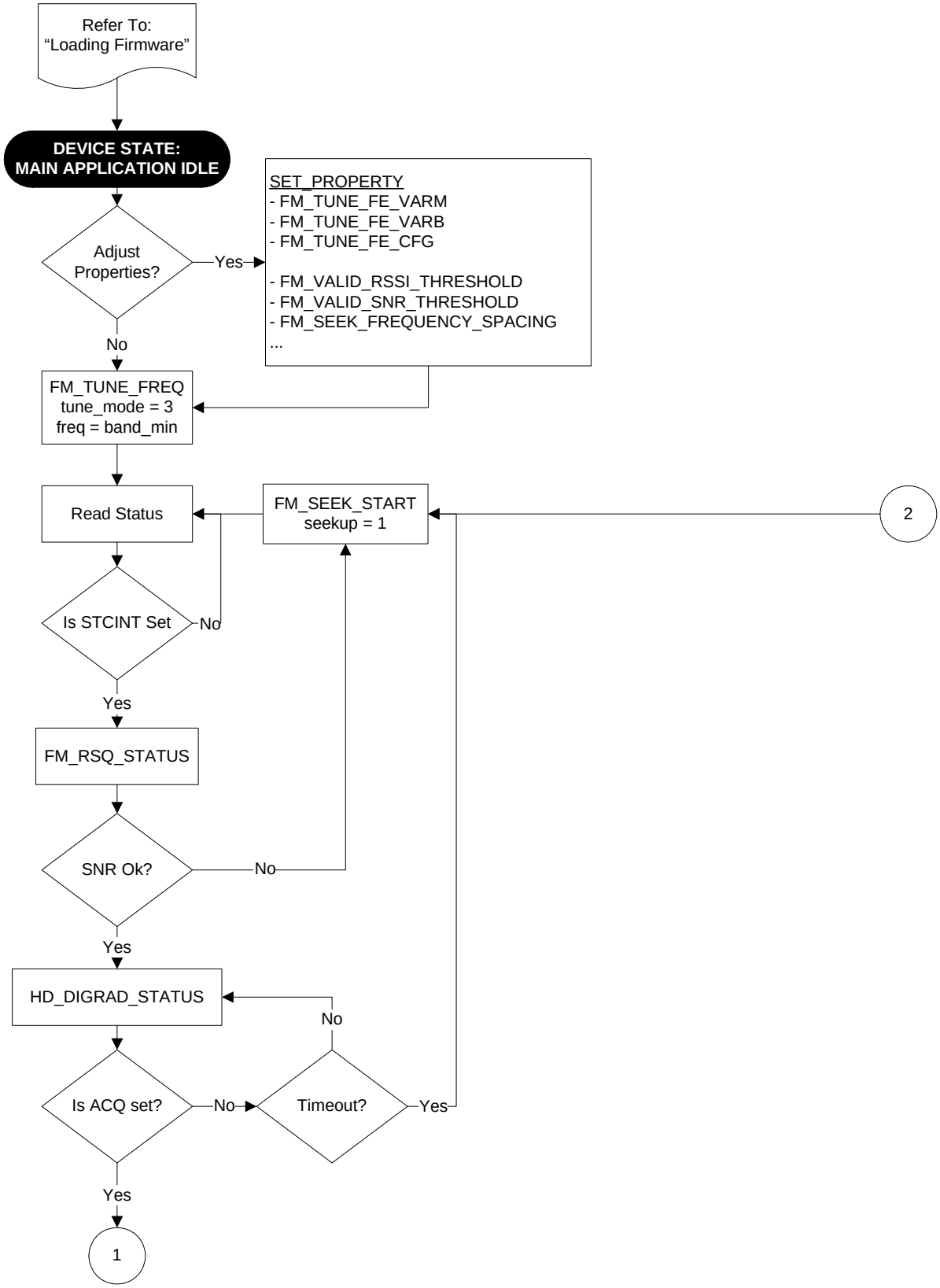


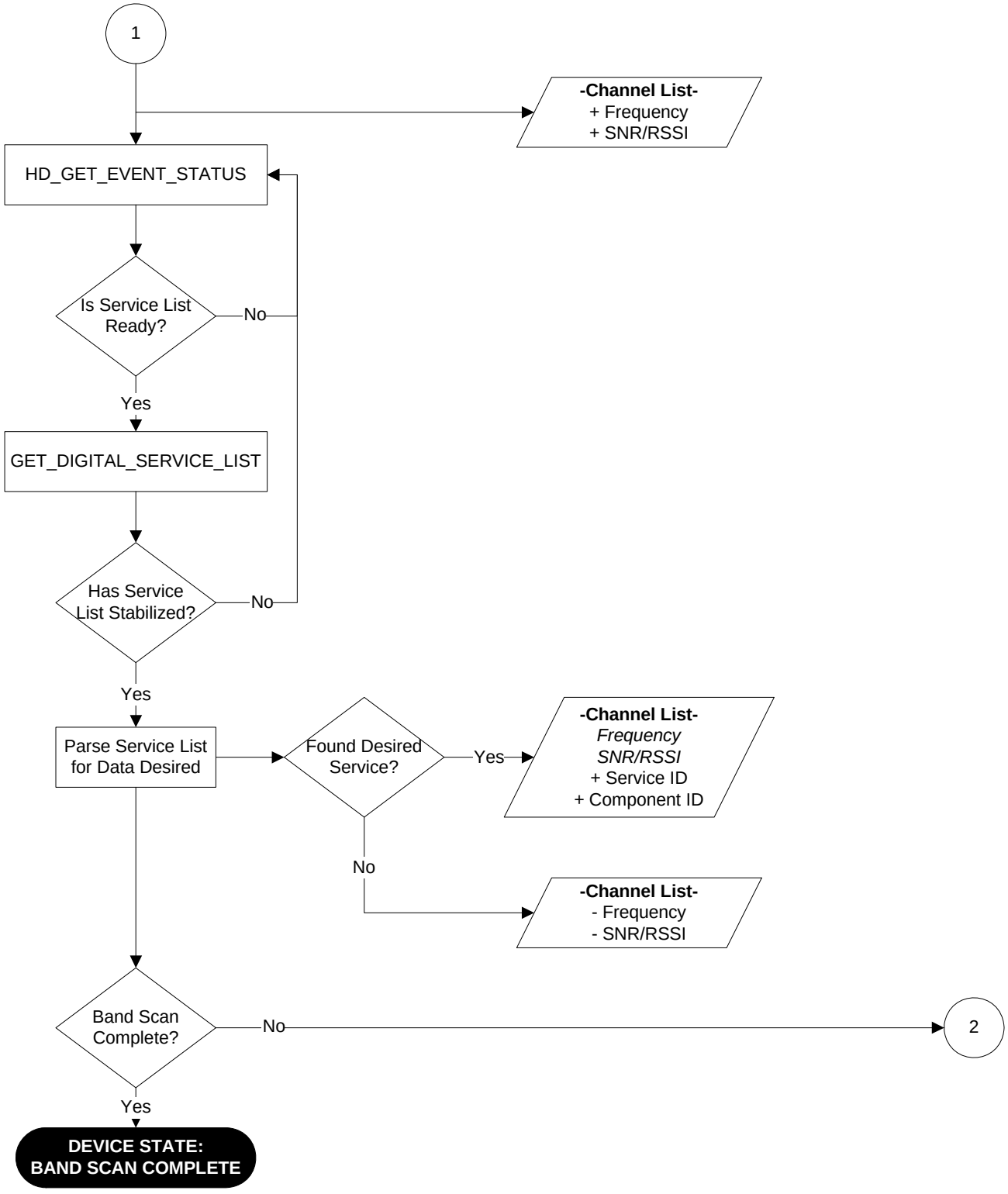

1
FM_RDS_STATUS
Parse RDS Blocks for Data Desired
Found Desired Data?
Yes
-Channel List-
+ Frequency
+ SNR/RSSI
Band Scan Complete?
No
2
Yes
DEVICE STATE:
BAND SCAN COMPLETE

## 5.3. Data Service Receiver—FM–RDS Data Service

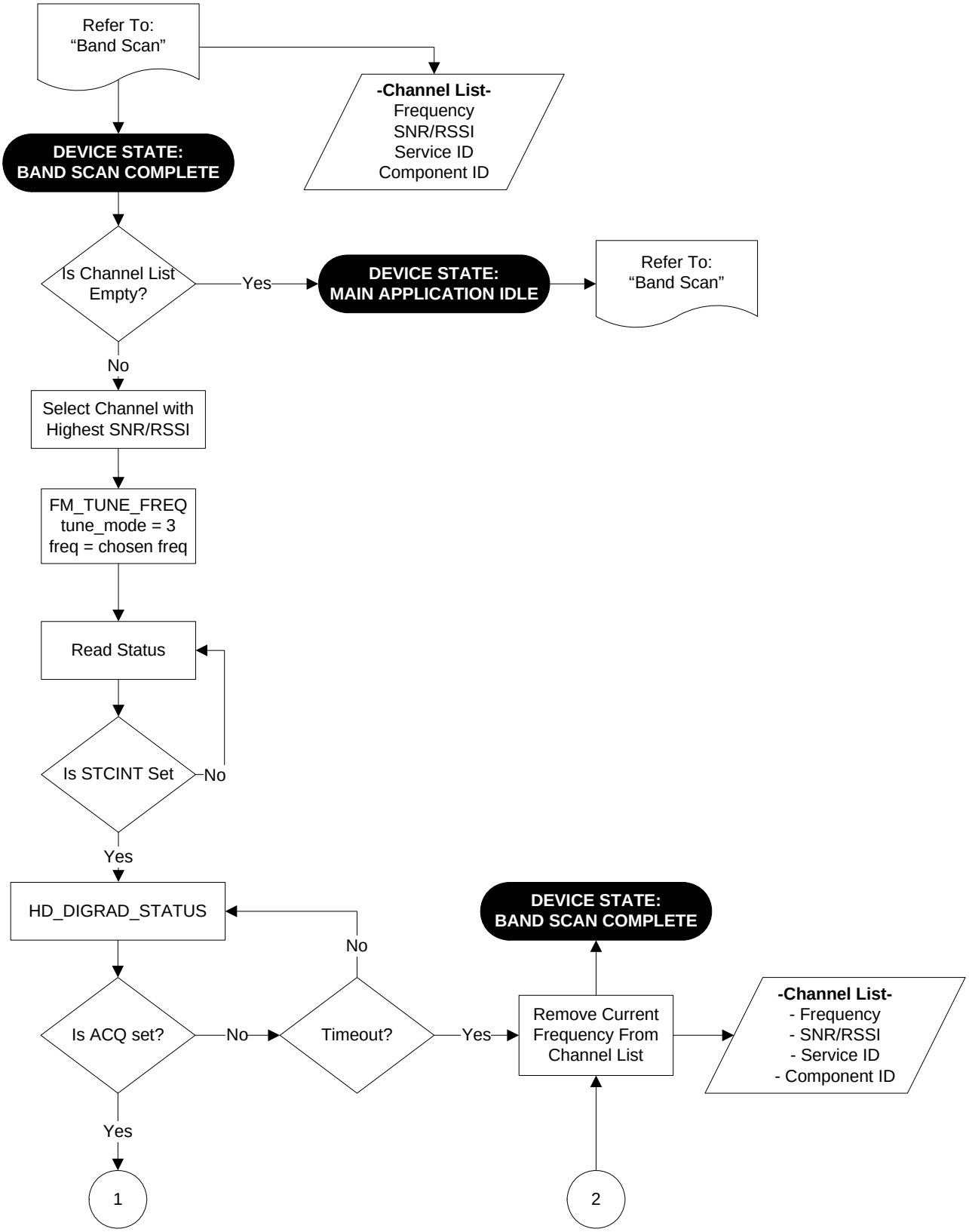


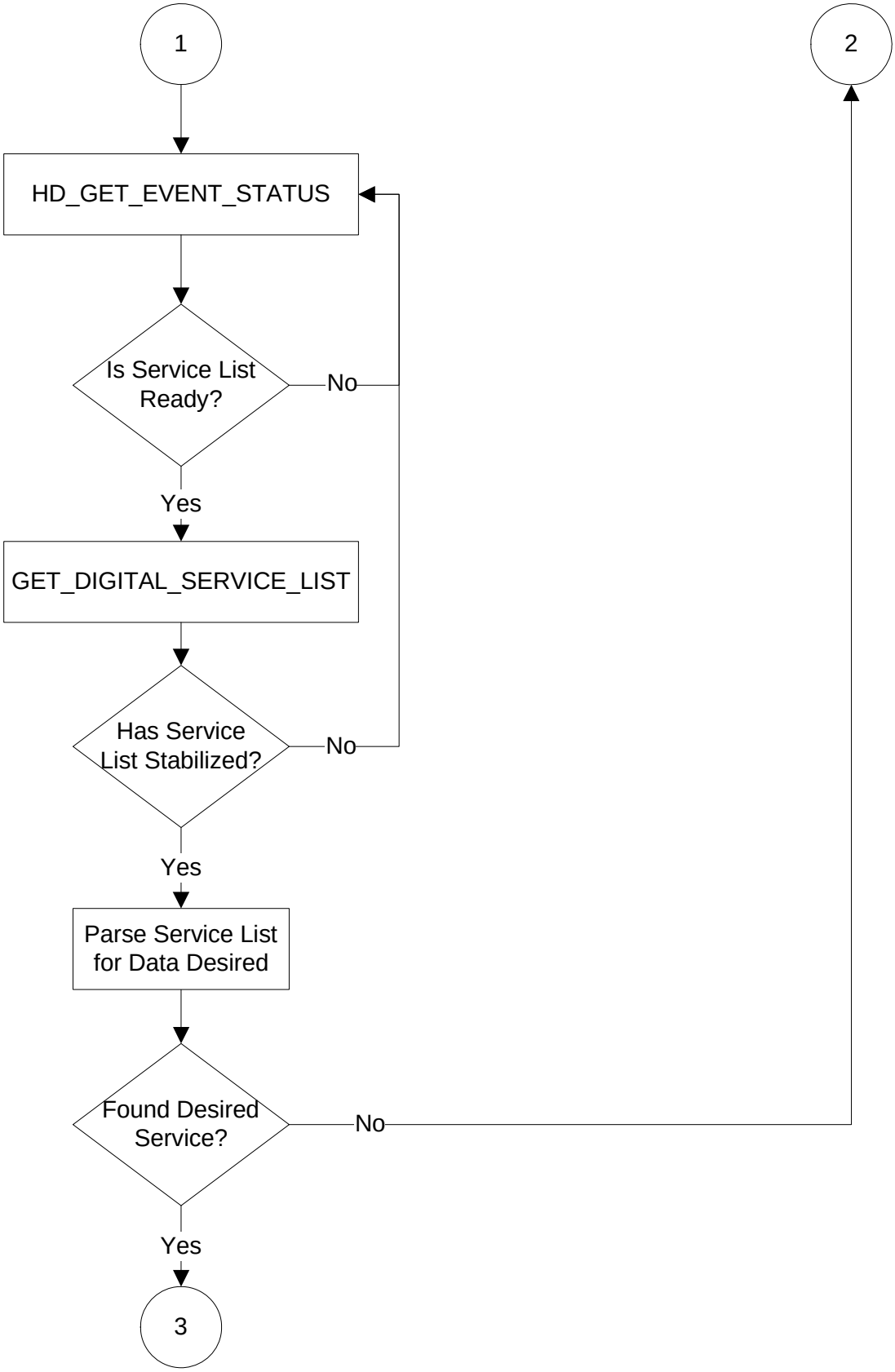
## 5.4. Data Service Receiver—FMHD–Band Scan

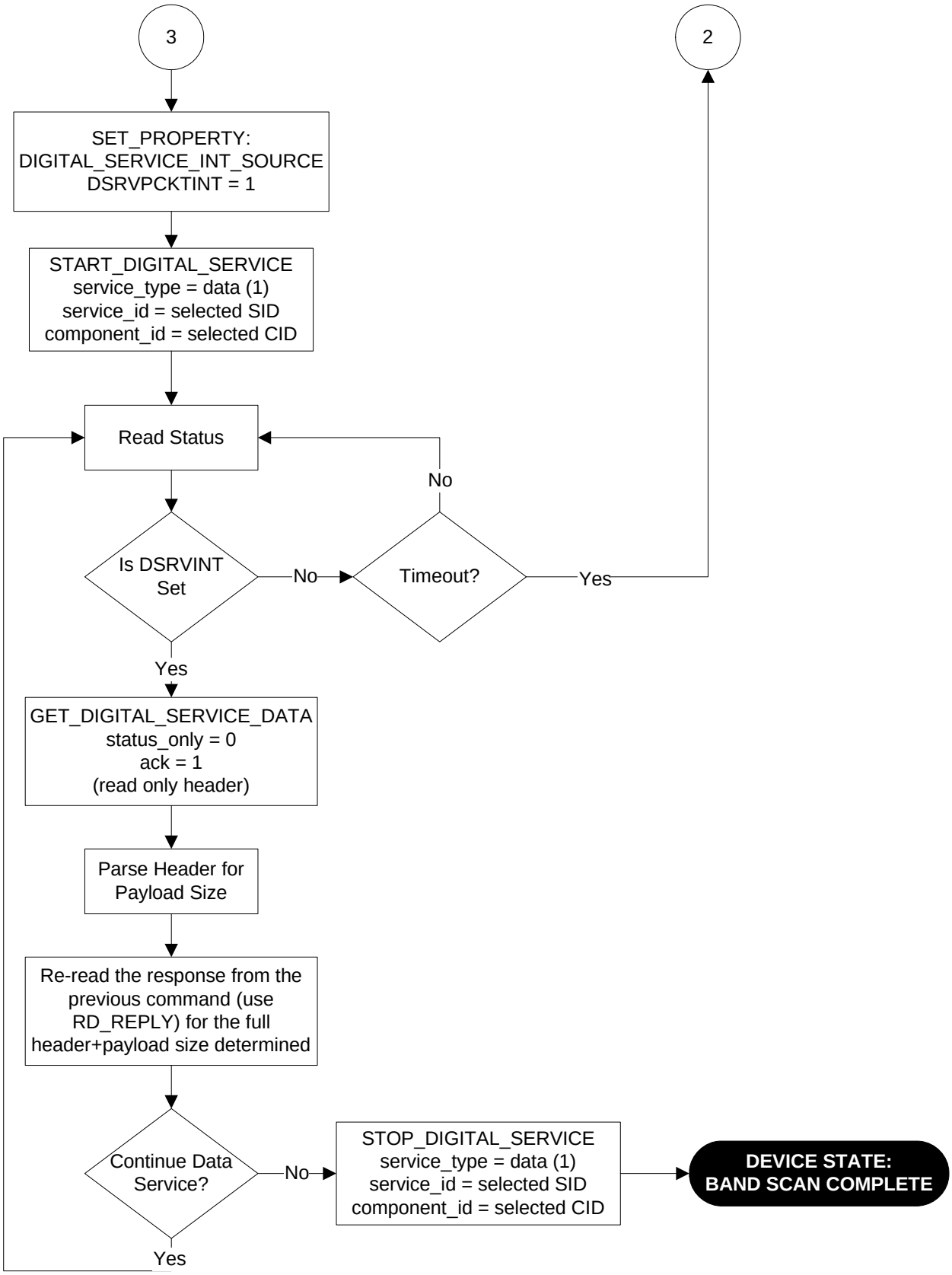



1
-Channel List-
+ Frequency
+ SNR/RSSI
HD_GET_EVENT_STATUS
Is Service List Ready?
No
Yes
GET_DIGITAL_SERVICE_LIST
Has Service List Stabilized?
No
Yes
Parse Service List for Data Desired
Found Desired Service?
Yes
-Channel List-
Frequency
SNR/RSSI
+ Service ID
+ Component ID
No
-Channel List-
- Frequency
- SNR/RSSI
Band Scan Complete?
No
2
Yes
DEVICE STATE:
BAND SCAN COMPLETE

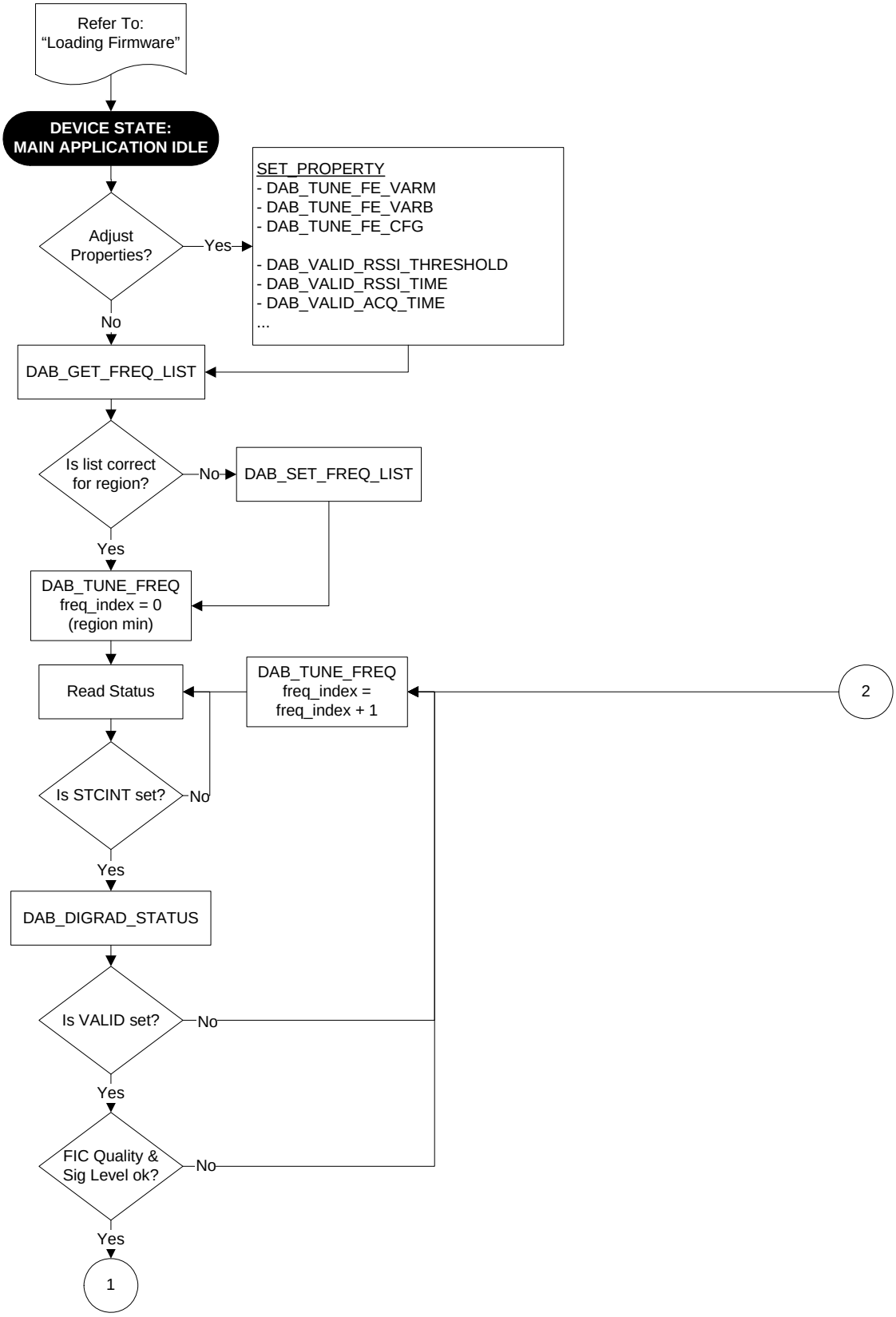
## 5.5. Data Service Receiver—FMHD–Data Services

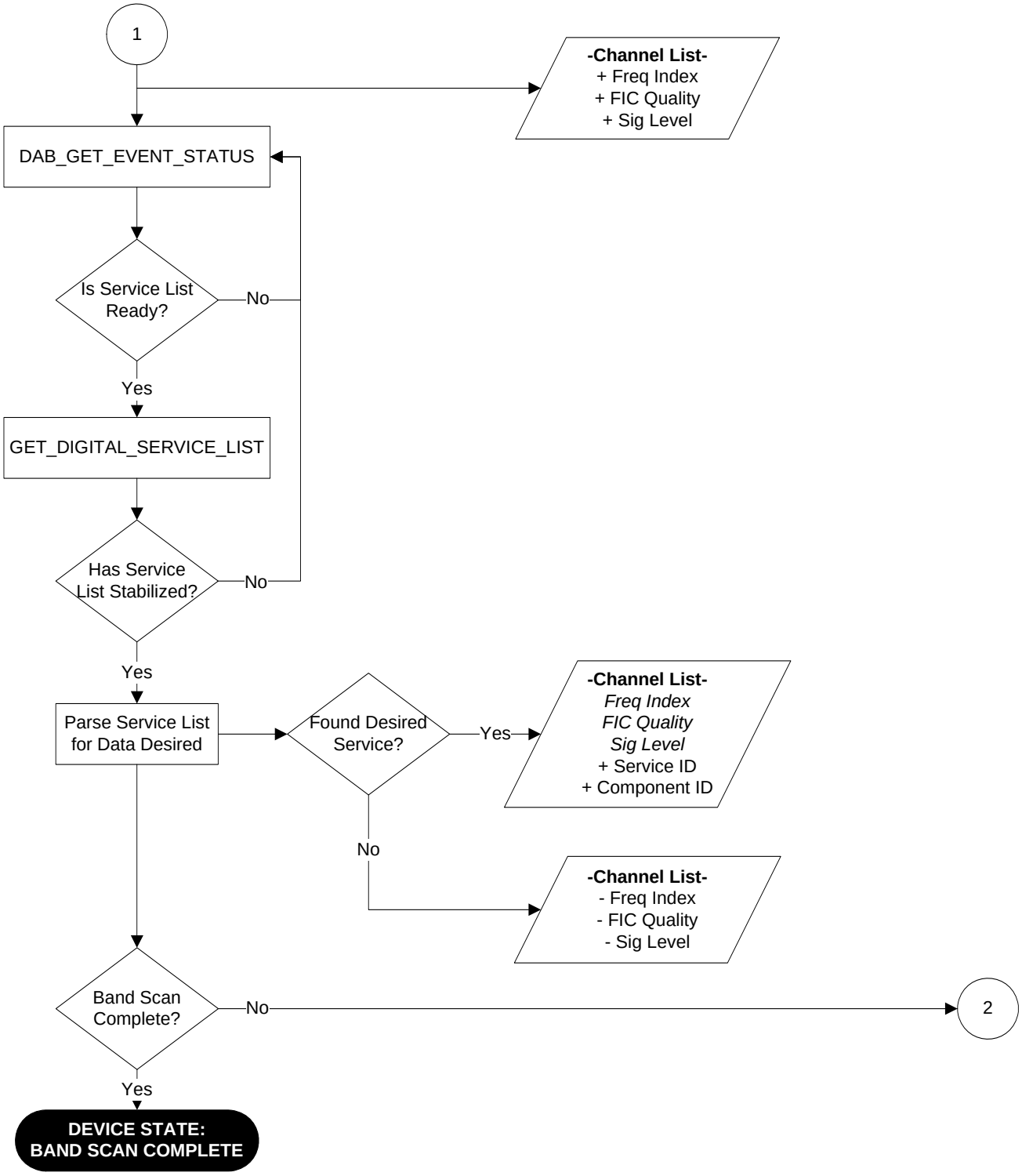



1
2
HD_GET_EVENT_STATUS
Is Service List Ready?
No
Yes
GET_DIGITAL_SERVICE_LIST
Has Service List Stabilized?
No
Yes
Parse Service List for Data Desired
Found Desired Service?
No
Yes
3

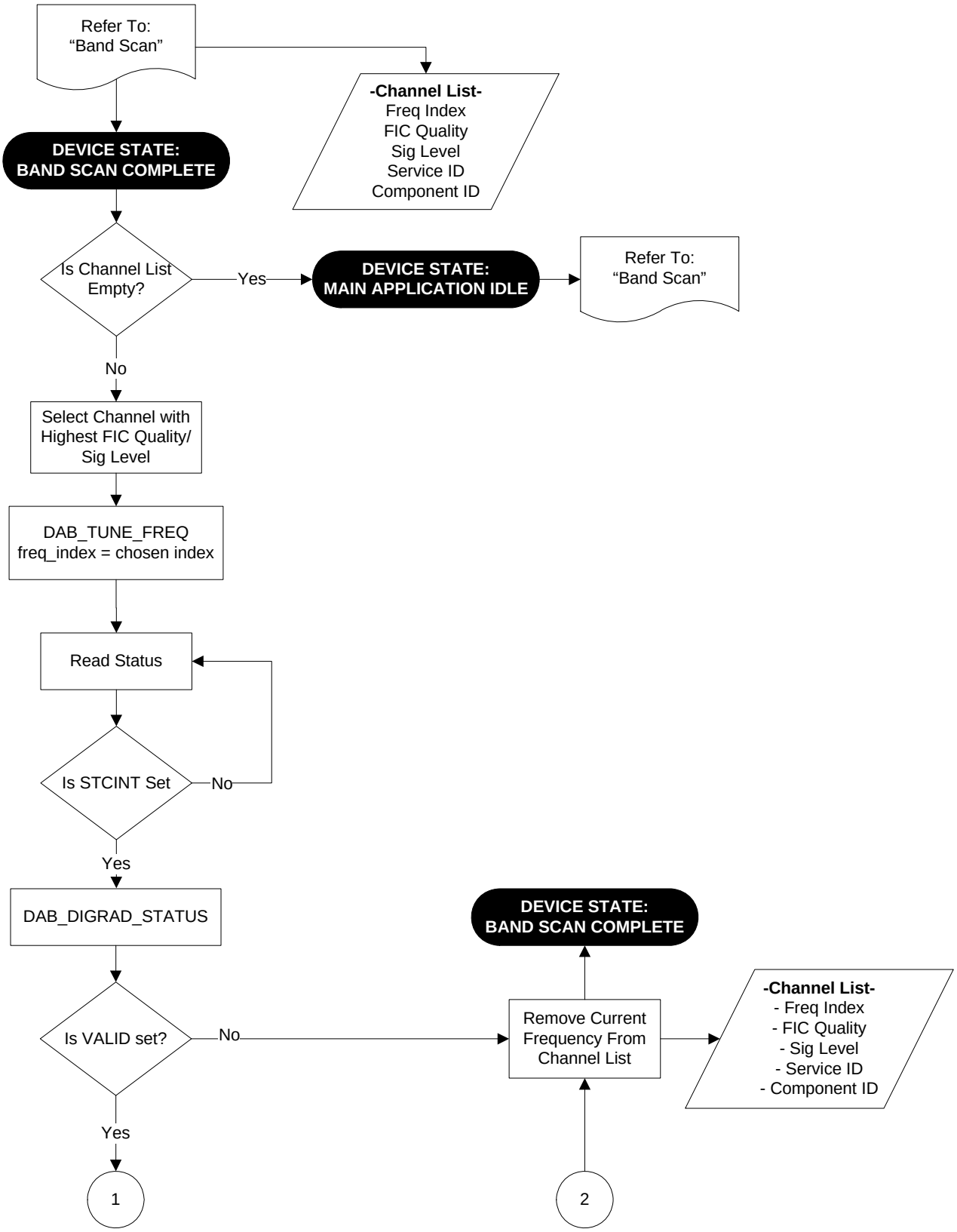

3
SET_PROPERTY:
DIGITAL_SERVICE_INT_SOURCE
DSRVPCKTINT = 1
START_DIGITAL_SERVICE
service_type = data (1)
service_id = selected SID
component_id = selected CID
Read Status
Is DSRVINT Set
No
Timeout?
Yes
2
No
Yes
GET_DIGITAL_SERVICE_DATA
status_only = 0
ack = 1
(read only header)
Parse Header for Payload Size
Re-read the response from the previous command (use RD_REPLY) for the full header+payload size determined
Continue Data Service?
No
STOP_DIGITAL_SERVICE
service_type = data (1)
service_id = selected SID
component_id = selected CID
DEVICE STATE:
BAND SCAN COMPLETE
Yes

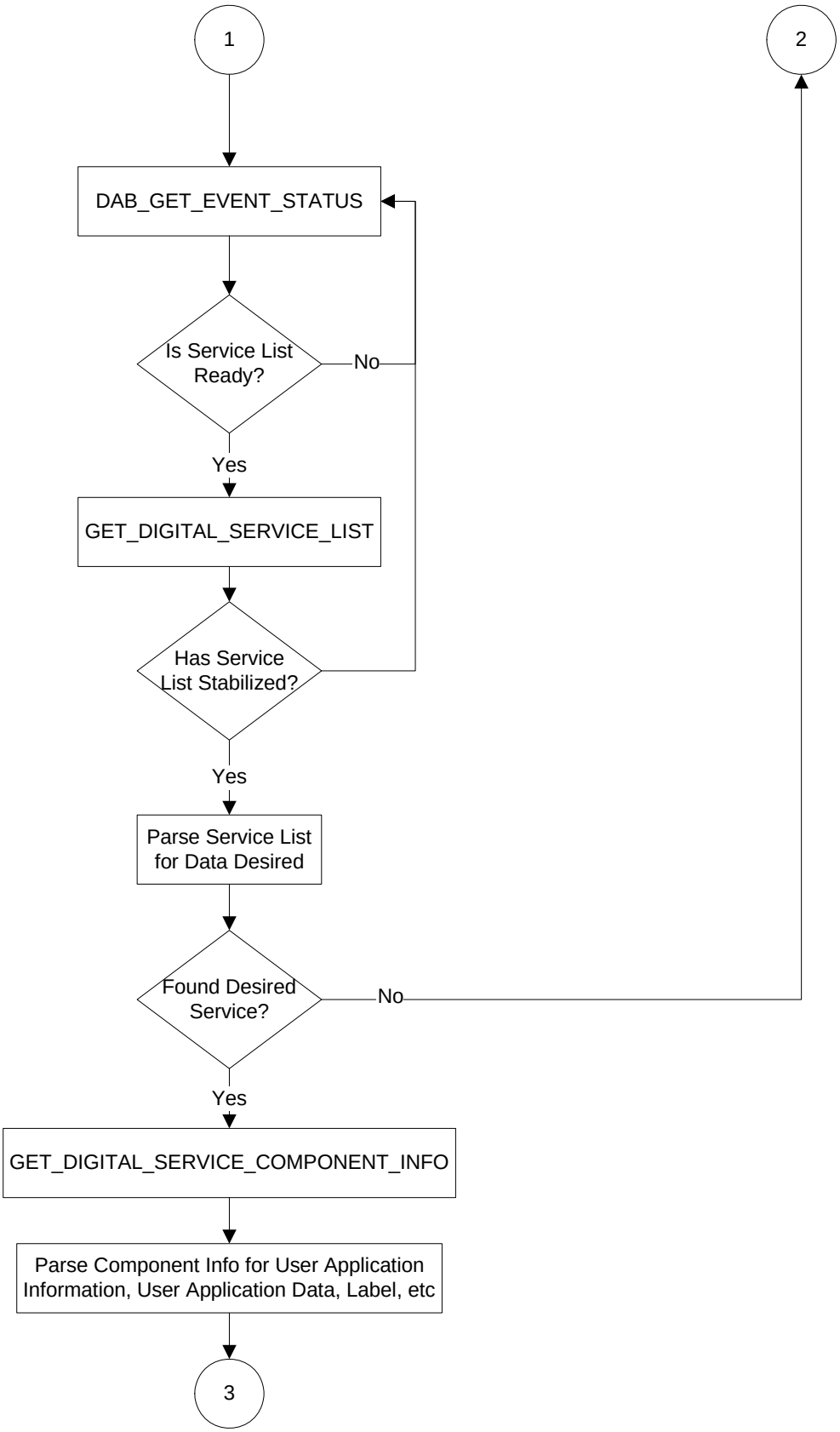
## 5.6. Data Service Receiver—DAB–Band Scan

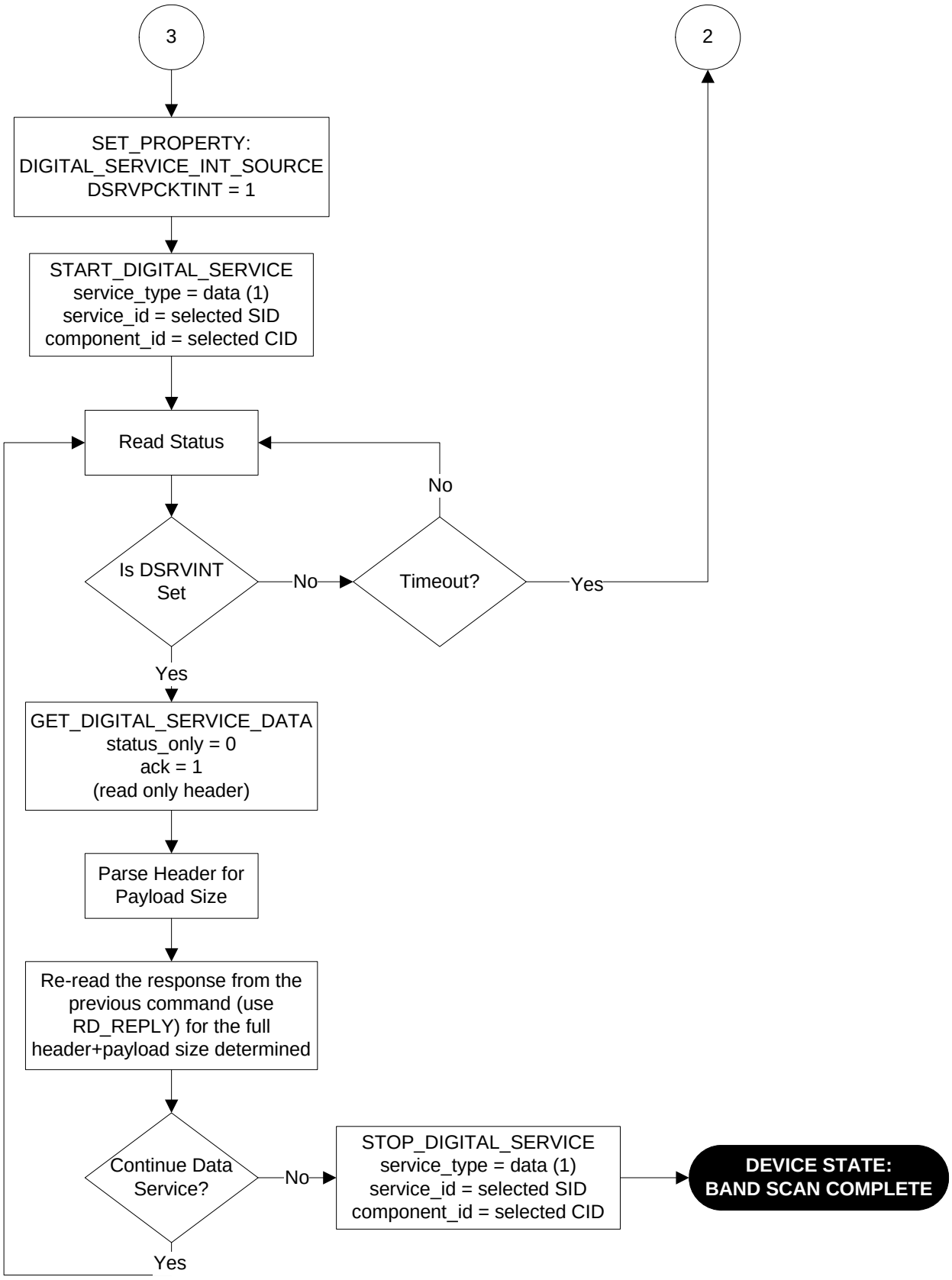



1
-Channel List-
+ Freq Index
+ FIC Quality
+ Sig Level
DAB_GET_EVENT_STATUS
Is Service List Ready?
No
Yes
GET_DIGITAL_SERVICE_LIST
Has Service List Stabilized?
No
Yes
Parse Service List for Data Desired
Found Desired Service?
Yes
-Channel List-
Freq Index
FIC Quality
Sig Level
+ Service ID
+ Component ID
No
-Channel List-
- Freq Index
- FIC Quality
- Sig Level
Band Scan Complete?
No
2
Yes
DEVICE STATE:
BAND SCAN COMPLETE

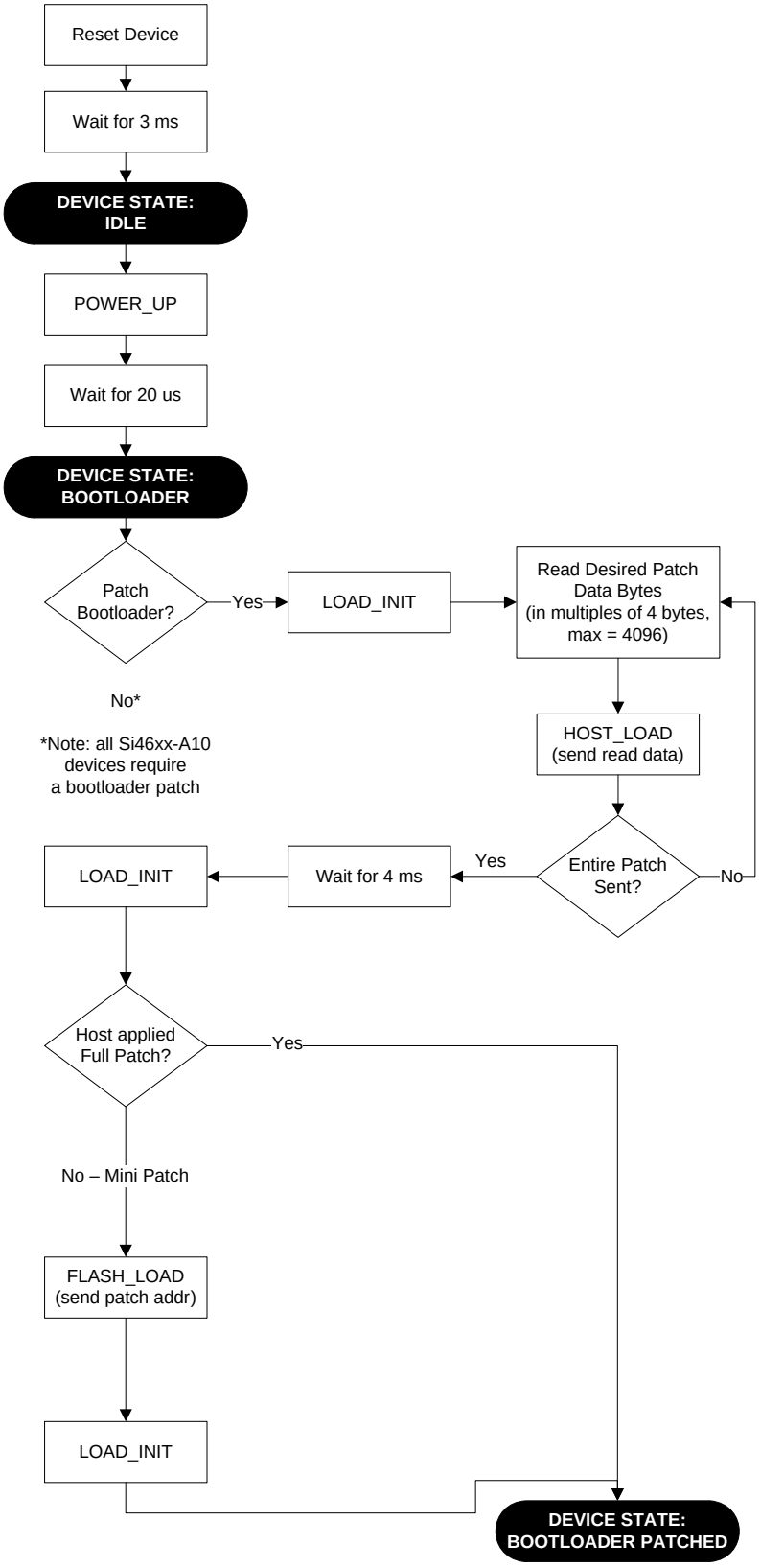
## 5.7. Data Service Receiver—DAB–Data Services

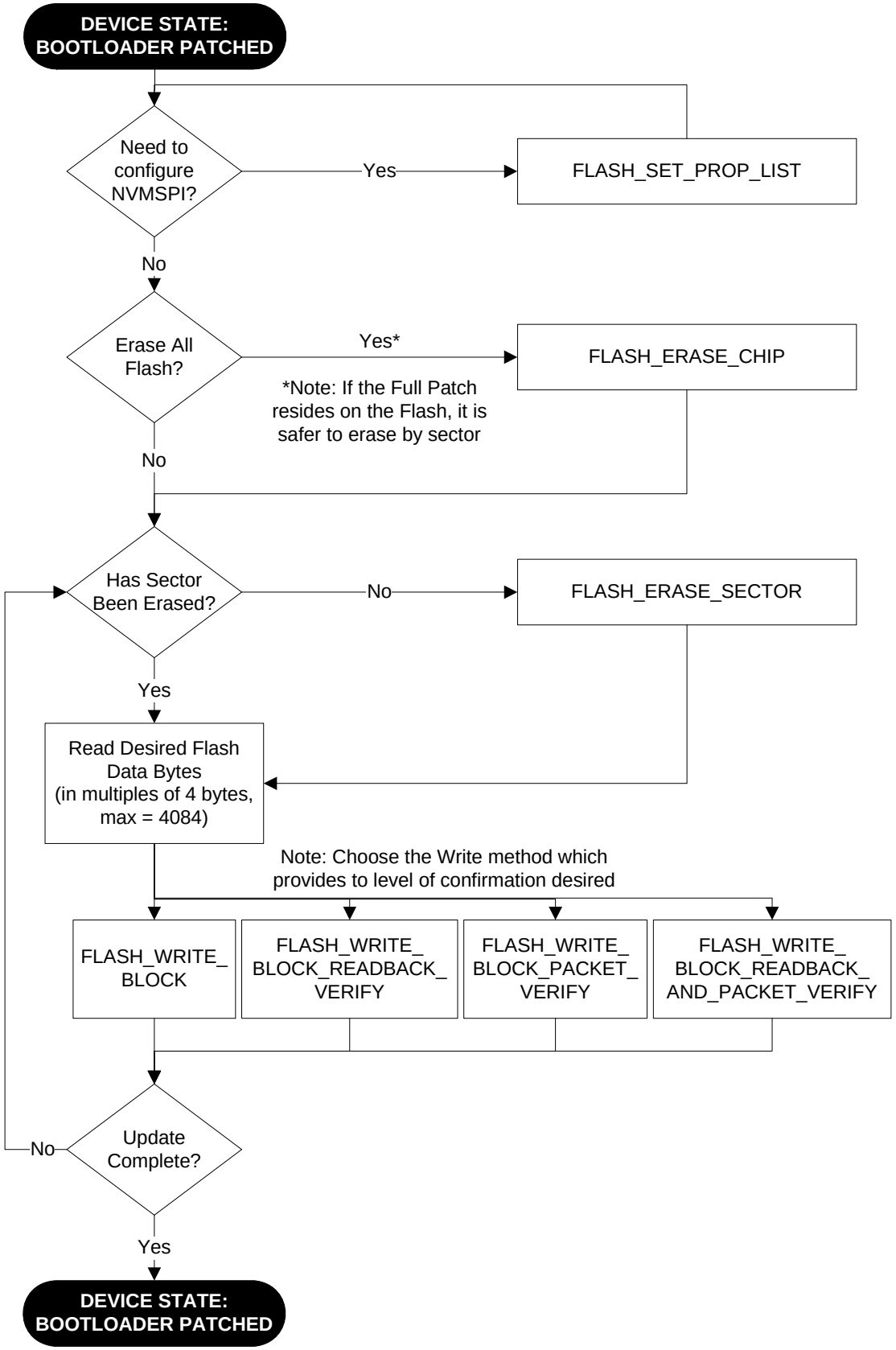



1
2
DAB_GET_EVENT_STATUS
Is Service List Ready?
No
Yes
GET_DIGITAL_SERVICE_LIST
Has Service List Stabilized?
Yes
Parse Service List for Data Desired
Found Desired Service?
No
Yes
GET_DIGITAL_SERVICE_COMPONENT_INFO
Parse Component Info for User Application Information, User Application Data, Label, etc
3


3
2
SET_PROPERTY:
DIGITAL_SERVICE_INT_SOURCE
DSRVPCKTINT = 1
START_DIGITAL_SERVICE
service_type = data (1)
service_id = selected SID
component_id = selected CID
Read Status
No
Is DSRVINT Set
No
Timeout?
Yes
Yes
GET_DIGITAL_SERVICE_DATA
status_only = 0
ack = 1
(read only header)
Parse Header for Payload Size
Re-read the response from the previous command (use RD_REPLY) for the full header+payload size determined
Continue Data Service?
No
STOP_DIGITAL_SERVICE
service_type = data (1)
service_id = selected SID
component_id = selected CID
DEVICE STATE:
BAND SCAN COMPLETE
Yes

## 5.8. Updating the Boot Flash




DEVICE STATE: BOOTLOADER PATCHED
Need to configure NVMSPI?
Yes
FLASH_SET_PROP_LIST
No
Erase All Flash?
Yes*
FLASH_ERASE_CHIP
*Note: If the Full Patch resides on the Flash, it is safer to erase by sector
No
Has Sector Been Erased?
No
FLASH_ERASE_SECTOR
Yes
Read Desired Flash Data Bytes (in multiples of 4 bytes, max = 4084)
Note: Choose the Write method which provides to level of confirmation desired
FLASH_WRITE_ BLOCK
FLASH_WRITE_ BLOCK_READBACK_ VERIFY
FLASH_WRITE_ BLOCK_PACKET_ VERIFY
FLASH_WRITE_ BLOCK_READBACK_ AND_PACKET_VERIFY
Update Complete?
No
Yes
DEVICE STATE: BOOTLOADER PATCHED

# 6. Managing Firmware

The Si468x is RAM based which requires the Host Controller to load any patch or firmware files before us. This section describes the system architecture options surrounding loading the necessary firmware files.

## 6.1. Firmware Components

There are multiple types of firmware which need to be loaded onto the Si468x. In general, it is necessary to first load a small bootloader patch and a matched firmware image which corresponds to the desired mode of operation.

When planning resource allocation for your system, you should expect to carry both a Patch and Firmware Image for each mode. The space requirements are outlined below.

**Table 9. Firmware Component Image Sizes**

| Firmware Component Name | Expected Max Size |
|---|---|
| MiniPatch<br>(Required for Certain Architectures) | 1 kB |
| FullPatch per Mode | 8 kB / mode |
| Firmware Image per Mode | 512 kB / mode |

For example: For a device which supports FM radio and DAB radio modes, the total firmware size would be 8 kB + 8 kB + 512 kB + 512 kB = 1040 kB. If the system architecture uses flash memory, it would be necessary to use a 2 MB flash (16 Mb). If less modes are supported the flash requirements would downsize accordingly.

Below if a sample flash layout for the example of a device which will support two modes of operation. If more modes are desired, it is recommended to follow the trend of grouping FullPatches then grouping Firmware Images. If this general model is followed, the flash spacing will accommodate future image upgrades without having to change start addresses and will provide easy sector boundaries for erasing.

**Table 10. Example Flash Layout for Accommodating Sector Boundaries and Future Updates**

| Component Name | Start Address | Max Size |
|---|---|---|
| Customer Specific Items<br>■ Content Description<br>■ Versioning<br>■ Etc | 0x00000000 | 8 kB |
| FullPatch Mode 1 | 0x00002000 | 8 kB |
| FullPatch Mode 2 | 0x00004000 | 8 kB |
| Firmware Image Mode 1 | 0x00006000 | 512 kB |
| Firmware Image Mode 2 | 0x00086000 | 512 kB |

## 6.2. System Architecture Options

The following tables provide an architectural representation of the Host Controller and Si468x and the associated relation to any external flash. The location for Si468x firmware components are specified for each architecture to aid in understanding the boot process.

**Note:** Components grayed out with diagonal fill are not required for that architecture, but are shown for consistency between the diagrams.

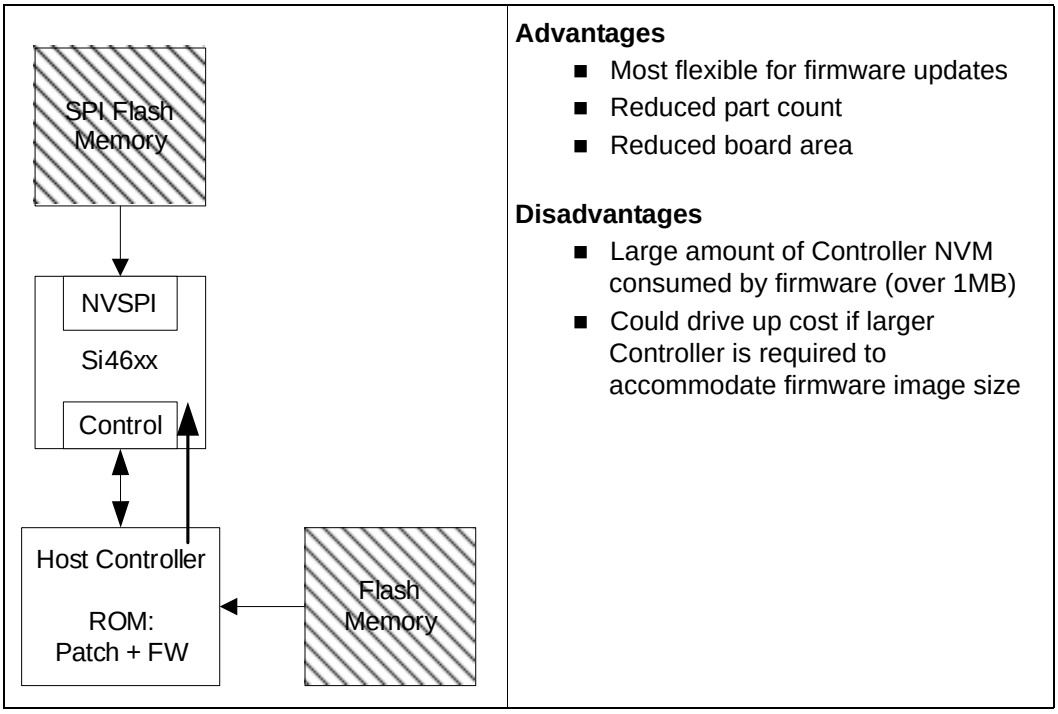
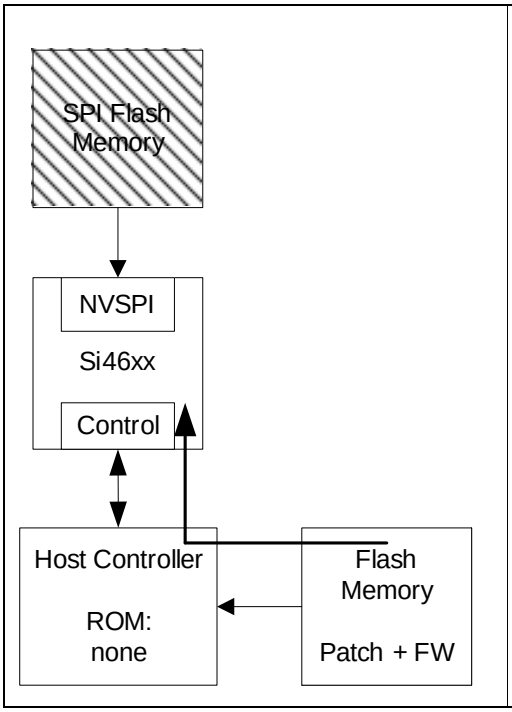



**Figure 1. HostLoad Large Controller**



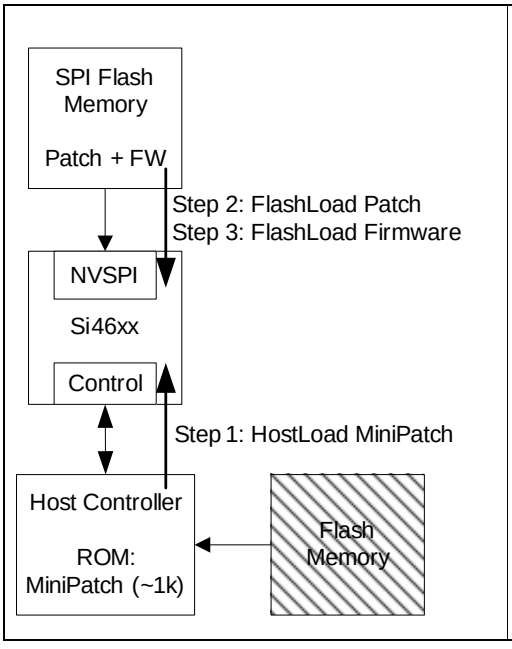

**Advantages**

- Flexible for firmware updates
- Most Flexible for flash vendor and capability choice
- Good for systems which already have flash connected to the controller for other uses

**Disadvantages**

- Longer boot times; significantly longer if Controller is using I2C to communicate with the Si468x

**Figure 2. HostLoad Small Controller**




**Advantages**

- Fastest Boot Time
- Good for systems using more than one Si468x as the NVSPI bus can be shared
- Small NVM requirements for Host Controller
- I2C bus is viable for host control

**Disadvantages**

- More difficult for firmware updates
- Flash selected must adhere to the JEDEC standard programming model

**Figure 3. FlashLoad SmallController Mini Patch**

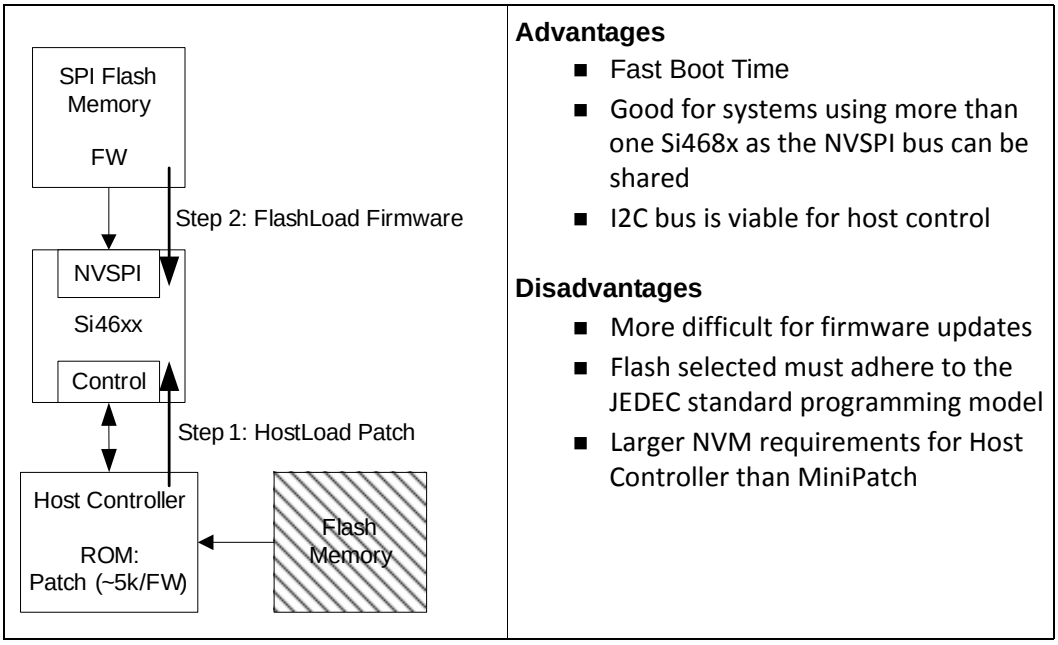



**Figure 4. FlashLoad LargeController Full Patch**

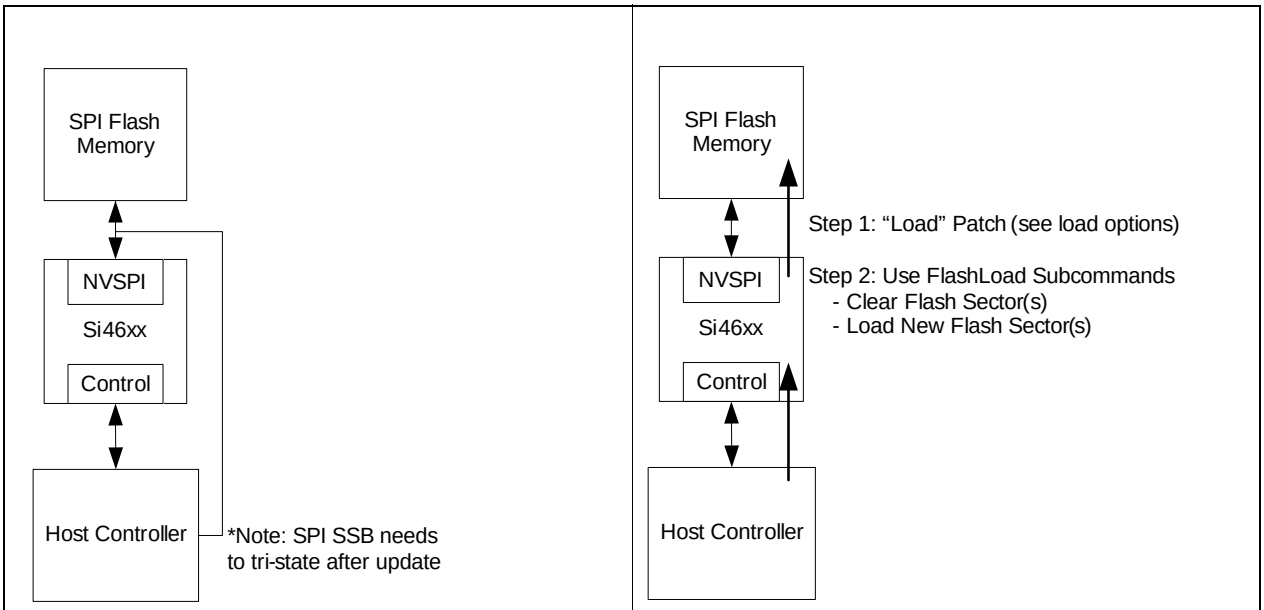
## 6.3. System Architecture Recommendations

Below are some general recommendations based upon certain existing system requirements prior to integrating the Si468x. Note: any recommendations here should be evaluated for your own design to ensure it meets the goals for cost, PCB space, desired boot speed, etc.

| System Requirement | Recommended Architecture |
|---|---|
| Host Control Interface is $I^2C$ OR<br>Fastest boot time is desired | FlashLoad SmallController MiniPatch |
| Host Controller already has flash memory attached AND<br>Host Controller is using a SPI interface to the Si468x | HostLoad SmallController |

## 6.4. Updating NVSPI Flash

When the desired architecture calls for attaching a SPI flash to the Si468x, there are two methods for supporting a "field update" scenario.




| | |
|---|---|
| The Host Controller is directly connected to the NVSPI flash using an additional SPI port (or SSB for chip select).<br><br>**Advantages:**<br>■ Fairly easy update software required as Host Controller manages flash directly<br>■ Potential software reuse from other products which interact with the SPI flash<br><br>**Disadvantages:**<br>■ More complicated hardware design and drivers. | The Host Controller uses the FLASH_LOAD sub-commands to manipulate the NVSPI through the Si468x.<br><br>**Advantages:**<br>■ Less complicated hardware design<br><br>**Disadvantages:**<br>■ Potentially slower update process as commands must be routed through the Si468x |

**Figure 5. Supporting Firmware Update to NVSPI Flash**

## 6.5. Managing Firmware Subcommands

### 6.5.1. NVSPI Flash Pass-Through

The following supplemental commands and properties are provided in order to provide host access to the NVSPI connected flash through the Si468x command interface.

In order to use these commands the A10 bootloader should be patched to ROM0.016. This can be accomplished using HOST_LOAD of the FullPatch ROM0.016 or by using HOST_LOAD of the MiniPatch.003, then loading the FullPatch ROM0.016 from flash using FLASH_LOAD.

#### 6.5.1.1. Flash Pass-Through Commands

**Table 11. Flash Pass-through Command Summary**

| Flash Pass-Through Commands – 0x05 <SubCmd> | | |
|---|---|---|
| **SubCmd** | **Name** | **Summary** |
| 0x02 | FLASH_CHECK_CRC32 | Check CRC32 of a sequence of blocks on flash |
| 0xFF | FLASH_ERASE_CHIP | Erase the entire flash |
| 0xFE | FLASH_ERASE_SECTOR | Erase a single sector of flash |
| 0x11 | FLASH_GET_PROPERTY | Get the value of a single property |
| 0x00 | FLASH_LOAD_IMG | Load a boot_img from flash |
| 0x01 | FLASH_LOAD_IMG_CHECK_CRC32 | Load a boot_img from flash. Check the CRC32 of the boot_img as the boot_img is loaded |
| 0x10 | FLASH_SET_PROP_LIST | Set the value of a list of properties |
| 0xF0 | FLASH_WRITE_BLOCK | Write a block of bytes to the flash |
| 0xF1 | FLASH_WRITE_BLOCK_READBACK_VERIFY | Write a block of bytes to the flash. Reads back the block after write to verify the block was written correctly by comparing the CRC32 |
| 0xF2 | FLASH_WRITE_BLOCK_PACKET_VERIFY | Write a block of data to the flash. Compares the CRC32 of the block of data received by the command. |
| 0xF3 | FLASH_WRITE_BLOCK_READBACK_AND_PACKET_VERIFY | Write a block of data to the flash. Reads back the block after write to verify the block was written correctly by comparing the CRC32. Also compares the CRC32 of the block of data received by the command. |

Unless otherwise specified, the commands provide only the status bits to reflect the commands success. The common status bit definition is outlined below.

**Response**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **STATUS0** | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | x | STCINT |
| **STATUS1** | X | X | DEVENTINT | X | X | X | X | X |
| **STATUS2** | XX | | | | | | | |
| **STATUS3** | PUP_STATE[1:0] | | XX | | REPOFERR | CMDOFERR | ARBERR | ERRNR |

## Command 0x05 0x02 – FLASH_CHECK_CRC32

Check CRC32 of a sequence of blocks on flash. If the CRC32 of the sequence of blocks on flash does not match, sets the status ERR bit. Use LOAD_INIT to clear the status ERR bit.

**Parameters**

- CRC32_CORRECT [31:0]—Correct CRC32 for the binary file.
- FLASH_ADDR [31:0]—Starting byte address of the block on flash.
- SIZE [31:0]—Size of the block in bytes on the flash.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0x02 | | | | | | | |
| **SUBCMD2** | 0x00 | | | | | | | |
| **SUBCMD3** | 0x00 | | | | | | | |
| **ARG0** | CRC32_CORRECT [7:0] | | | | | | | |
| **ARG1** | CRC32_CORRECT [15:8] | | | | | | | |
| **ARG2** | CRC32_CORRECT [23:16] | | | | | | | |
| **ARG3** | CRC32_CORRECT [31:24] | | | | | | | |
| **ARG4** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG5** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG6** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG7** | FLASH_ADDR [31:24] | | | | | | | |
| **ARG8** | SIZE [7:0] | | | | | | | |
| **ARG9** | SIZE [15:8] | | | | | | | |
| **ARG10** | SIZE [23:16] | | | | | | | |
| **ARG11** | SIZE [31:24] | | | | | | | |

### Command 0x05 0xFF – FLASH_ERASE_CHIP

Erase the entire flash.

**Parameters**

None

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xFF | | | | | | | |
| **SUBCMD2** | 0xDE | | | | | | | |
| **SUBCMD3** | 0xC0 | | | | | | | |

### Command 0x05 0xFE – FLASH_ERASE_SECTOR

Erase a single sector of flash.

**Parameters**

- SECTOR_ADDR [31:0]—Starting address on flash of the sector to erase, byte offset from the start of flash. Note: sector_addr[23..0] are used, sector_addr[31..24] are ignored. The least significant bits that would be masked by the sector boundary are ignored.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xFE | | | | | | | |
| **SUBCMD2** | 0xC0 | | | | | | | |
| **SUBCMD3** | 0xDE | | | | | | | |
| **ARG0** | SECTOR_ADDR [7:0] | | | | | | | |
| **ARG1** | SECTOR_ADDR [15:8] | | | | | | | |
| **ARG2** | SECTOR_ADDR [23:16] | | | | | | | |
| **ARG3** | SECTOR_ADDR [31:24] | | | | | | | |

### Command 0x05 0x11 – FLASH_GET_PROPERTY

Get the value of a single property.

**Parameters**

- PROPERTY_ID [15:0]—ID of the property to get.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0x11 | | | | | | | |
| **ARG0** | PROPERTY_ID [7:0] | | | | | | | |
| **ARG1** | PROPERTY_ID [15:8] | | | | | | | |

**Response**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **STATUS0** | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | x | STCINT |
| **STATUS1** | X | X | DEVENTINT | X | X | X | X | X |
| **STATUS2** | XX | | | | | | | |
| **STATUS3** | PUP_STATE[1:0] | | XX | | REPO-FERR | CMDOFERR | ARBERR | ERRNR |
| **RESP0** | PROPERTY_VALUE[7:0] | | | | | | | |
| **RESP1** | PROPERTY_VALUE[15:8] | | | | | | | |

- PROPERTY_VALUE [15:0]—**Property value associated with property ID.**

### Command 0x05 0x00 – FLASH_LOAD_IMG – (FLASH_LOAD)

Load a firmware image or patch from flash. This is the same as the FLASH_LOAD command but is represented here following the flash subcommand format.

**Parameters**

- FLASH_ADDR [31:0]—Starting byte address of the block on flash.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0x00 | | | | | | | |
| **SUBCMD2** | 0x00 | | | | | | | |
| **SUBCMD3** | 0x00 | | | | | | | |
| **ARG0** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG1** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG2** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG3** | FLASH_ADDR [31:24] | | | | | | | |

### Command 0x05 0x01 – FLASH_LOAD_IMG_CHECK_CRC32

Load a firmware image or patch from flash. Check the CRC32 of the boot_img as the boot_img is loaded. If the CRC32 of the boot_img does not match CRC32_CORRECT, the status ERR bit is set.

**Parameters**

- CRC32_CORRECT [31:0]—Correct CRC32 for the binary file.
- FLASH_ADDR [31:0]—Starting byte address of the block on flash.
- SIZE [31:0]—Size of the block in bytes on the flash.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0x01 | | | | | | | |
| **SUBCMD2** | 0x00 | | | | | | | |
| **SUBCMD3** | 0x00 | | | | | | | |
| **ARG0** | CRC32_CORRECT [7:0] | | | | | | | |
| **ARG1** | CRC32_CORRECT [15:8] | | | | | | | |
| **ARG2** | CRC32_CORRECT [23:16] | | | | | | | |
| **ARG3** | CRC32_CORRECT [31:24] | | | | | | | |

| | |
|---|---|
| **ARG4** | FLASH_ADDR [7:0] |
| **ARG5** | FLASH_ADDR [15:8] |
| **ARG6** | FLASH_ADDR [23:16] |
| **ARG7** | FLASH_ADDR [31:24] |
| **ARG8** | SIZE [7:0] |
| **ARG9** | SIZE [15:8] |
| **ARG10** | SIZE [23:16] |
| **ARG11** | SIZE [31:24] |

### Command 0x05 0x10 – FLASH_SET_PROP_LIST

Set the value of a list of properties to configure the parameters used in communicating with the NVSPI flash.

**Parameters**

- PROPERTY_ID [15:0]—ID of the property to get.
- PROPERTY_VALUE [15:0]—Property value associated with property ID.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0x10 | | | | | | | |
| **SUBCMD2** | 0x00 | | | | | | | |
| **SUBCMD3** | 0x00 | | | | | | | |
| **ARG0** | PROPERTY_ID_0 [7:0] | | | | | | | |
| **ARG1** | PROPERTY_ID_0 [15:8] | | | | | | | |
| **ARG2** | PROPERTY_VALUE_0 [7:0] | | | | | | | |
| **ARG3** | PROPERTY_VALUE_0 [15:8] | | | | | | | |
| **ARG4** | PROPERTY_ID_1 [7:0] | | | | | | | |
| **ARG5** | PROPERTY_ID_1 [15:8] | | | | | | | |
| **ARG6** | PROPERTY_VALUE_1 [7:0] | | | | | | | |
| **ARG7** | PROPERTY_VALUE_1 [15:8] | | | | | | | |
| **…** | … | | | | | | | |
| **ARG_N-3** | PROPERTY_ID_N [7:0] | | | | | | | |
| **ARG_N-2** | PROPERTY_ID_N [15:8] | | | | | | | |
| **ARG_N-1** | PROPERTY_VALUE_N [7:0] | | | | | | | |
| **ARG_N** | PROPERTY_VALUE_N [15:8] | | | | | | | |

### Command 0x05 0xF0 – FLASH_WRITE_BLOCK

Write a block of bytes to the flash. All the bytes on flash that are written must have been previously erased to 0xFF with the FLASH_ERASE_CHIP or FLASH_ERASE_SECTOR subcommands.

**Parameters**

FLASH_ADDR [31:0]—Starting address on flash of the block to write, byte offset from the start of flash.

SIZE [31:0]—Size of the block to write in bytes.

BLOCK_DATA [7:0]—Sequence of bytes to write to flash. Maximum number of bytes is 4084.

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xF0 | | | | | | | |
| **SUBCMD2** | 0x0C | | | | | | | |
| **SUBCMD3** | 0xED | | | | | | | |
| **ARG0** | 0x00 | | | | | | | |
| **ARG1** | 0x00 | | | | | | | |
| **ARG2** | 0x00 | | | | | | | |
| **ARG3** | 0x00 | | | | | | | |
| **ARG4** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG5** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG6** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG7** | FLASH_ADDR [31:24] | | | | | | | |
| **ARG8** | SIZE [7:0] | | | | | | | |
| **ARG9** | SIZE [15:8] | | | | | | | |
| **ARG10** | SIZE [23:16] | | | | | | | |
| **ARG11** | SIZE [31:24] | | | | | | | |
| **ARG12** | BLOCK_DATA_0 | | | | | | | |
| **ARG13** | BLOCK_DATA_1 | | | | | | | |
| **…** | … | | | | | | | |
| **ARG_N** | BLOCK_DATA_N | | | | | | | |

### Command 0x05 0xF1 – FLASH_WRITE_BLOCK_READBACK_VERIFY

Write a block of bytes to the flash. All the bytes on flash that are written must have been previously erased to 0xFF with the FLASH_ERASE_CHIP or FLASH_ERASE_SECTOR subcommands. Reads back the block after write to verify the block was written correctly by comparing the CRC32 of the block read back with crc32_correct. If there is not a match, the status ERR bit is set.

**Parameters**

- CRC32_CORRECT [31:0]—CRC32 of the block of data written.
- FLASH_ADDR [31:0]—Starting address on flash of the block to write, byte offset from the start of flash.
- SIZE [31:0]—Size of the block to write in bytes.
- BLOCK_DATA [7:0]—Sequence of bytes to write to flash. Maximum number of bytes is 4084

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xF1 | | | | | | | |
| **SUBCMD2** | 0x0C | | | | | | | |
| **SUBCMD3** | 0xED | | | | | | | |
| **ARG0** | CRC32_CORRECT [7:0] | | | | | | | |
| **ARG1** | CRC32_CORRECT [15:8] | | | | | | | |
| **ARG2** | CRC32_CORRECT [23:16] | | | | | | | |
| **ARG3** | CRC32_CORRECT [31:24] | | | | | | | |
| **ARG4** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG5** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG6** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG7** | FLASH_ADDR [31:24] | | | | | | | |
| **ARG8** | SIZE [7:0] | | | | | | | |
| **ARG9** | SIZE [15:8] | | | | | | | |
| **ARG10** | SIZE [23:16] | | | | | | | |
| **ARG11** | SIZE [31:24] | | | | | | | |
| **ARG12** | BLOCK_DATA_0 | | | | | | | |
| **ARG13** | BLOCK_DATA_1 | | | | | | | |
| **…** | … | | | | | | | |
| **ARG_N** | BLOCK_DATA_N | | | | | | | |

### Command 0x05 0xF2 – FLASH_WRITE_BLOCK_PACKET_VERIFY

Write a block of data to the flash. All the bytes on flash that are written must have been previously erased to 0xFF with the FLASH_ERASE_CHIP or FLASH_ERASE_SECTOR subcommands. Compares the CRC32 of the block of data received by the command with CRC32_CORRECT. If there is not a match, the status ERR bit is set.

**Parameters**

- CRC32_CORRECT [31:0]—CRC32 of the block of data written.
- FLASH_ADDR [31:0]—Starting address on flash of the block to write, byte offset from the start of flash.
- SIZE [31:0]—Size of the block to write in bytes.
- BLOCK_DATA [7:0]—Sequence of bytes to write to flash. Maximum number of bytes is 4084

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xF2 | | | | | | | |
| **SUBCMD2** | 0x0C | | | | | | | |
| **SUBCMD3** | 0xED | | | | | | | |
| **ARG0** | CRC32_CORRECT [7:0] | | | | | | | |
| **ARG1** | CRC32_CORRECT [15:8] | | | | | | | |
| **ARG2** | CRC32_CORRECT [23:16] | | | | | | | |
| **ARG3** | CRC32_CORRECT [31:24] | | | | | | | |
| **ARG4** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG5** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG6** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG7** | FLASH_ADDR [31:24] | | | | | | | |
| **ARG8** | SIZE [7:0] | | | | | | | |
| **ARG9** | SIZE [15:8] | | | | | | | |
| **ARG10** | SIZE [23:16] | | | | | | | |
| **ARG11** | SIZE [31:24] | | | | | | | |
| **ARG12** | BLOCK_DATA_0 | | | | | | | |
| **ARG13** | BLOCK_DATA_1 | | | | | | | |
| **…** | … | | | | | | | |
| **ARG_N** | BLOCK_DATA_N | | | | | | | |

### Command 0x05 0xF3 – FLASH_WRITE_BLOCK_READBACK_AND_PACKET_VERIFY

Write a block of data to the flash. All the bytes on flash that are written must have been previously erased to 0xFF with the FLASH_ERASE_CHIP or FLASH_ERASE_SECTOR subcommands. Reads back the block after write to verify the block was written correctly by comparing the CRC32 of the block read back with crc32_correct. Also compares the CRC32 of the block of data received by the command with CRC32_CORRECT. If either of the compares does not match, the status ERR bit is set.

**Parameters**

- CRC32_CORRECT [31:0]—CRC32 of the block of data written.
- FLASH_ADDR [31:0]—Starting address on flash of the block to write, byte offset from the start of flash.
- SIZE [31:0]—Size of the block to write in bytes.
- BLOCK_DATA [7:0]—Sequence of bytes to write to flash. Maximum number of bytes is 4084

**Command**

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x05 | | | | | | | |
| **SUBCMD1** | 0xF3 | | | | | | | |
| **SUBCMD2** | 0x0C | | | | | | | |
| **SUBCMD3** | 0xED | | | | | | | |
| **ARG0** | CRC32_CORRECT [7:0] | | | | | | | |
| **ARG1** | CRC32_CORRECT [15:8] | | | | | | | |
| **ARG2** | CRC32_CORRECT [23:16] | | | | | | | |
| **ARG3** | CRC32_CORRECT [31:24] | | | | | | | |
| **ARG4** | FLASH_ADDR [7:0] | | | | | | | |
| **ARG5** | FLASH_ADDR [15:8] | | | | | | | |
| **ARG6** | FLASH_ADDR [23:16] | | | | | | | |
| **ARG7** | FLASH_ADDR [31:24] | | | | | | | |
| **ARG8** | SIZE [7:0] | | | | | | | |
| **ARG9** | SIZE [15:8] | | | | | | | |
| **ARG10** | SIZE [23:16] | | | | | | | |
| **ARG11** | SIZE [31:24] | | | | | | | |
| **ARG12** | BLOCK_DATA_0 | | | | | | | |
| **ARG13** | BLOCK_DATA_1 | | | | | | | |
| **…** | … | | | | | | | |
| **ARG_N** | BLOCK_DATA_N | | | | | | | |

## 6.6. Flash Pass-Through Properties

**Table 12. Common Property Summary**

| Common Properties | | | |
|---|---|---|---|
| **Number** | **Name** | **Default** | **Summary** |
| 0x0001 | SPI_CLOCK_FREQ_KHZ | 0x61A8 (25000) | The SPI master clock frequency driving the external SPI flash in kHz |
| 0x0002 | SPI_MODE | 0x03 | SPI master mode |
| 0x0101 | READ_CMD | 0x03 | Flash read command |
| 0x0102 | HIGH_SPEED_READ_CMD | 0x0B | Flash fast or high speed read command |
| 0x0103 | HIGH_SPEED_READ_MAX_FREQ_MHZ | 0x00 | Maximum SPI clock frequency in MHz for the fast or high speed read command |
| 0x0201 | WRITE_CMD | 0x02 | Flash write command |
| 0x0202 | ERASE_SECTOR_CMD | 0x20 | Flash erase sector command |
| 0x0204 | ERASE_CHIP_CMD | 0xC7 | Flash bulk/entire chip erase command |

**Property 0x0001 SPI_CLOCK_FREQ_KHZ**

The SPI master clock frequency driving the external SPI flash.

**Default:** 0x61A8 (25000)

| SPI_CLOCK_FREQ_KHZ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| SPI_CLOCK_FREQ_KHZ [15:0] | | | | | | | | | | | | | | | |
| 0x61A8 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | SPI_CLOCK_FREQ_KHZ [15:0] | The SPI master clock frequency driving the external SPI flash<br>Default: 0x61A8. Range: 3000 to 40000 |

### Property 0x0002 SPI_MODE

SPI master mode.

**Default:** 0x03

| SPI_MODE | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| SPI_MODE [15:0] | | | | | | | | | | | | | | | |
| 0x03 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | spi_mode [15:0] | SPI master mode<br>0 = SPICFG1.CKPOL = 0, SPICFG1.CKPHA = 0.<br>1 = SPICFG1.CKPOL = 0, SPICFG1.CKPHA = 1.<br>2 = SPICFG1.CKPOL = 1, SPICFG1.CKPHA = 0.<br>3 = SPICFG1.CKPOL = 1, SPICFG1.CKPHA = 1.<br>Default: 0x03. Range: 0 to 3 |

### Property 0x0101 READ_CMD

Flash read command.

**Default:** 0x03

| READ_CMD | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| READ_CMD [15:0] | | | | | | | | | | | | | | | |
| 0x03 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | READ_CMD [15:0] | Flash read command<br>Default: 0x03. Range: 0x00 to 0xFF |

### Property 0x0102 HIGH_SPEED_READ_CMD

Flash fast or high speed read command.

**Default:** 0x0B

| high_speed_read_cmd | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| HIGH_SPEED_READ_CMD [15:0] | | | | | | | | | | | | | | | |
| 0X0B | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | HIGH_SPEED_READ_CMD [15:0] | Flash fast or high speed read command.<br>If set to 0x00, high speed read command will not be used.<br>Default: 0x0B. Range: 0x00 to 0xFF |

### Property 0x0103 HIGH_SPEED_READ_MAX_FREQ_MHZ

Maximum SPI clock frequency in MHz for the fast or high speed read command.

**Default:** 0x00

| HIGH_SPEED_READ_MAX_FREQ_MHZ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| HIGH_SPEED_READ_MAX_FREQ_MHZ [15:0] | | | | | | | | | | | | | | | |
| 0x00 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | HIGH_SPEED_READ_MAX_FREQ_MHZ [15:0] | Maximum SPI clock frequency in MHz for the fast or high speed read command.<br>If set to 0x00, command is unavailable<br>Default: 0x00. Range: 0 to 127 |

### Property 0x0201 WRITE_CMD

Flash write command.

Default: 0x02

| WRITE_CMD | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| WRITE_CMD [15:0] | | | | | | | | | | | | | | | |
| 0x02 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | WRITE_CMD [15:0] | Flash write command.<br>If set to 0x00, write command is unavailable<br>Default: 0x02. Range: 0x00 to 0xFF |

### Property 0x0202 ERASE_SECTOR_CMD

Flash erase sector command.

Default: 0x20

| ERASE_SECTOR_CMD | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| ERASE_SECTOR_CMD [15:0] | | | | | | | | | | | | | | | |
| 0x20 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ERASE_SECTOR_CMD [15:0] | Flash erase sector command<br><br>Micron M25P16—0xD8 for 64 kB sector<br>SST25VF032B—0x20 for 4 kB sector<br>SST25VF032B—0x52 for 32 kB sector<br>SST25VF032B—0xD8 for 64 kB sector<br><br>Default: 0x20. Range: 0x00 to 0xFF |

### Property 0x0204 ERASE_CHIP_CMD

Flash bulk/entire chip erase command.

Default**:** 0xC7

| ERASE_CHIP_CMD | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **15** | **14** | **13** | **12** | **11** | **10** | **9** | **8** | **7** | **6** | **5** | **4** | **3** | **2** | **1** | **0** |
| ERASE_CHIP_CMD [15:0] | | | | | | | | | | | | | | | |
| 0xC7 | | | | | | | | | | | | | | | |

| Bit | Name | Function |
|---|---|---|
| 15:0 | ERASE_CHIP_CMD [15:0] | Flash bulk/entire chip erase command.<br>Not available if set to 0x00.<br><br>Micron M25P16—0xC7<br>SSTVF032B—0x60 or 0xC7<br><br>Default: 0xC7. Range: 0x00 to 0xFF |

# 7. Digital Services User's Guide

This section describes how to use the digital services interface of the Si468x device. Both the DAB and HD data services are addressed. The following topics are covered.

- Finding a service in the ensemble
- Starting a digital service
- Using the digital service interface to collect digital service data
- Error handling and recovery
- Formats of the digital service data

## 7.1. Reference Material

The following references are cited throughout this document.

1. ETSI EN 300 401 (1.4.1): "Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers".
2. ETSI TS 101 756 (1.4.1): "Digital Audio Broadcasting (DAB); Registered Tables".
3. ETSI EN 301 234: "Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) protocol".
4. ETSI TS 102 368: "Digital Audio Broadcasting (DAB); DAB-TMC (Traffic Message Channel)".
5. ETSI TS 102 427: "Digital Audio Broadcasting (DAB); Data Broadcasting—MPEG-2 TS streaming".
6. ETSI TS 102 563: "Digital Audio Broadcasting (DAB); Data Broadcasting—Transport of Advanced Audio Coding (AAC) audio".
7. Biquity RX_IDD_2206: "HD Radio Commercial Receiver Baseband Processor Command and Data Interface Definition".
8. Biquity RX_IDD_2206 Appendix K: "HD Radio Commercial Receiver Baseband Processor Command and Data Interface Definition, Appendix K, Advanced Application Services".
9. Biquity RX_IDD_2206 Appendix L: "HD Radio Commercial Receiver Baseband Processor

## 7.2. Definition of Terms

- **Component ID**—A numerical identifier for a digital service component. In the HD Radio context this term is used synonymously with Port Number or Program Number.
- **Digital Service**—A service that is provided by a digital radio ensemble. This could be a data or audio service.
- **Digital Service Data**—The data associated with a digital service. In the context of this document this generally refers to data services.
- **DSCType**—Data Service Component Type. This is defined in [2] Table 2: DSCTy Types.
- **DSRV Interface**—The Digital Service Data Interface of the Si468x.
- **Service ID**—A numerical identifier for a digital service. In the HD Radio context this term is used synonymously with Service Number.
- **STATUS_WORD**—The STATUS_WORD is the first 32 bits of all command responses of the Si468x. It contains the STATUS bits as well as system state and error information. Please see the Host Command API documentation for details.
- **STATUS_BITS**—The first 16 bits (LSBs) of the STATUS_WORD. These bits are used to signal events on the Si468x device.
- **UI**—A generic user interface. Generally associated with an end product.

## 7.3. Basics of the Si468x Digital Services Interface

The Si468x Digital Service (DSRV) data interface has been designed to help make common the method of access to the various data services available in both the DAB and HD Radio broadcasting systems. This interface is generic and is used in both the DAB and HD Radio modes of operation. That is, it is not specific to either the DAB or HD Radio standards. Its purpose is to help ease the task of integrating and managing the various data services available (such as a traffic or news service) in the given standards. This interface adds a DSRV header to each data packet (or payload) returned by the digital demodulator. The intent is not to redefine the payload sections but to provide a common data services management layer for the host. In general the payload sections are unaltered from what has been defined for a given standard. The DSRV header is used by the host to determine the following:

- Identification of the Service and Component IDs of a data packet for routing to the appropriate decoder. For the HD Radio standard, the Service ID and Component ID are the same as the Service Number and Port Number respectively as defined in [9].
- The current status of the data service. The status is used to signal the following types of information.
  - The service has encountered an error or has stopped
  - The service is running normally
- The packet's sequence number, if applicable.
- The packet's service type.
- The size of the packet data.
- The number of data packets remaining in the DSRV queue.

**Note:** The production release of the Si468x firmware may provide the option of removing the DSRV header.

## 7.4. Basic Operation

Both the DAB and HD Radio standards provide a means for a broadcaster to include data based services as part of a broadcast. Before one of these services can be enabled it is necessary for the host application to identify and select a service of interest. It is also necessary for the host application to have the appropriate data decoder to support a given service. Once a service has been identified it is started by the host and the data for that service begins to flow through the DSRV interface.

The DSRV interface is designed to be an event driven interface. Whenever a new block of data is available for an enabled data service the host is notified via a status bit or interrupt. The host may then retrieve the data block through a host command. The DSRV interface can also be used in a polling mode if desired. The various interface methods for using this interface are described below.

The DSRV interface uses a queuing mechanism to hold the data packets until the host is able to collect them. This queue relaxes the timing requirements placed on the host by allowing it to collect all packets in the queue at one time when it is convenient (within reason). The maximum number of packets stored before the packet queue overflows is eight.

There are five steps required to start and render a digital service. Each of these steps is explored in more detail in the sections that follow. Each of the steps is briefly outlined here.

1. Collect and parse the digital service list using the GET_DIGITAL_SERVICE_LIST host command.
2. Determine which services and service components look appealing (Generally this is a user requested selection from a UI).
3. Start a service using the START_DIGITAL_SERVICE host command.
4. Collect the service data using the GET_DIGITAL_SERVICE_DATA host command.
5. Decode and render the service using the appropriate data decoder. Note that, with few exceptions, all service decoding is done by the host.

## 7.5. Finding a Digital Service

Before starting a data service the host must collect information about the services that exist in the ensemble. This is done using the GET_DIGITAL_SERVICE_LIST host command. The service list definitions for both HD Radio and DAB are described in “7.5.1. HD Radio Service List” and “7.5.2. DAB/DMB Radio Service List” . Table 13 shows the format of the GET_DIGITAL_SERVICE_LIST host command.

**Table 13. GET_DIGITAL_SERVICE_LIST Command**

| GET_DIGITAL_SERVICE_LIST Command | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x80 | | | | | | | |
| **ARG1** | 000000 | | | | | | 0 | SERTYPE |

The argument SERTYPE is used to select between audio and data service lists in HD Radio applications. This argument should be set to 0 when running the DAB application.

The service list may be updated periodically by the broadcaster. Therefore the host must keep track of the current service list and update the UI information as appropriate. Notification that a new service list is available is handled by the service list update notifications of the HD_GET_EVENT_STATUS or DAB_GET_EVENT_STATUS host commands. Please review the API documentation of these commands for details regarding the service list update notifications. Each service list has a version associated with it so that host can easily determine if their service list is out of date.

Additionally, in DAB mode at tune time and during reconfiguration the service list gets updated frequently as the ensemble configuration becomes known. To deal with periods when the service list is changing frequently a debounce timer is employed to limit the number of service list update events received by the host. This timer works by blocking service list update events until the service list is stable for a period of time. See the property DAB_EVENT_MIN_SVRLIST_PERIOD for details on setting up the service list event debounce timer.

### 7.5.1. HD Radio Service List

In HD Radio mode there is a separate service list for data services and for audio services. The type of service list returned is determined by the SERTYPE command argument of the GET_DIGITAL_SERVICE_LIST host command. The service lists for HD Radio mode conform to the HD 2206 specification. The definitions of these service lists are found in [9]. The format of the returned service lists is defined by the GET_ALL_AUDIO_SERVICES_INFO and GET_ALL_DATA_SERVICES_INFO functions defined in L-1 of [9]. In both cases Byte 0 as defined in the L-1 is not needed and is omitted.

**Notes:**

1. It is important to note that for the purposes of this document Service ID is synonymous with the HD Radio "Service Number" and Component ID is synonymous with the HD Radio "Program Number" (audio) or "Port Number" (data). This means that when starting a data service the Service Number is used as the Service ID and Port Number is used as the Component ID.
2. For HD Radio the START_DIGITAL_SERVICE and STOP_DIGITAL_SERVICE host commands do not require the Service ID and it may be omitted. In this case a 0 must be used in place of the Service ID.

### 7.5.2. DAB/DMB Radio Service List

Unlike HD Radio the DAB/DMB application has a single service list for both data and audio services. This list is returned by the GET_DIGITAL_SERVICE_LIST host command. The SERTYPE argument is not used and should be set to 0. The DAB/DMB service list is shown in Table 2 below. The list was designed with the FIG formats defined in [1] in mind. Therefore most of the fields map closely to the FIG configurations. Additionally the service list is designed to be small but complete enough for the host to make decisions about the services. This made it necessary to combine certain FIG fields into a single table entry in the table.

To support extensibility and the promote completeness of information, additional Host Command APIs are needed to collect the additional information from the Si468x device.

**Table 14. DAB/DMB Digital Service List Definition**

| Table Entry | | Definition | | | | Size (Bytes) |
|---|---|---|---|---|---|---|
| **List Size** | | Indicates the number of bytes in the digital service list (Max = 2694 Bytes, not including List Size) | | | | 2 |
| **Version** | | Version of this service list. The version number increments by 1 each time the list changes. | | | | 2 |
| **Number of Services** | | Indicates the number of services defined in the list (N; N ≤ 32). | | | | 1 |
| **Align Pad** | | RFU | | | | 3 |
| **Service 1** | | | | | | |
| **Service ID** (See clause 6.3.1 of EN 300 401) | P/D = 0 | RFU[31:16] | CountryID[15:12] | SRV_REF[11:0] | | 4 |
| | P/D = 1 | ECC[31:24] | CountryID[23:20] | SRV_REF[19:0] | | |
| **Service Info 1** | | RFU[7] | SrvLinking Info Flag[6] | Pty[5:1] | P/D Flag[0] | 1 |
| **Service Info 2** (See clause 6.3.1 of EN 300 401 for actual max components) | | LOCAL[7] | | CAId[6:4] | NUM_COMP[3:0] (M ≤ 15) | 1 |
| **Service Info 3** | | RFU[4:8] | | | SlCharset[3:0] (See ETSI TS 101 756, clause 5.3) | 1 |
| Align Pad | | | | | | 1 |
| **Service Label** | | The name of this service | | | | 16 |
| **Service 1 Component 1** | | | | | | |
| **Component ID** (See clause 6.3.1 of EN 300 401) | **TMID = 0** | TMID[15:14] | Rfu[13:6] | | SubChId[5:0] | 2 |
| | **TMID = 1** | TMID[15:14] | Rfu[13:6] | | SubChId[5:0] | |
| | **TMID = 2** | TMID[15:14] | Rfu[13:6] | | FIDCId[5:0] | |
| | **TMID = 3** | TMID[15:14] | DGFlag[13] | Rfu[12] | SCId[11:0] | |
| **Component Info** | | ASCTy[7:2]/ DSCTy[7:2] | | P/S[1] | CA Flag[0] | 1 |
| Valid Flags | | RFU[7:1] | | | µA Info Valid[0] | 1 |
| The component field is repeated M-1 times for each service component in the service. | | | | | | |
| **Service 2** | | | | | | |
| The service field is repeated N-1 times for each service in the ensemble. | | | | | | |

Each of the fields in the service list that are not directly defined in the table above are defined here.

- **Align Pad**—Alignment pads are added to the list to insure that table entries that are larger the 8 bits will be properly aligned when received by the host. In other words 16 bit entries will fall on even address boundaries and so on. To take advantage of this the host must align reply buffers on a 32 bit boundary.
- **RFU**—Entries reserved for future use.
- **Pty**—The service program type. See Annex A. of [2].
- **SrvLinkingInfo Flag**—This bit is set to 1 if service linking info is available for this service.
- **P/D Flag**—This bit is set to 0 for a program (audio) service and 1 for a data service. See clause 5.2.2.1 of [1].
- **LOCAL Flag**—Indicates if the service is available over the entire (0) or part (1) of the ensemble service area. See clause 6.3.1 of [1].
- **CAId**—Identifies the access control used. See clause 6.3.1 of [1].
- **CA Flag**—If set to 1 this flag indicates that access control applies. See 6.3.1 of [1].
- **DGFlag**—Indicates that the data service is transmitted in data groups. See clause 5.3 of [1].
- **ASCTy**—Audio Service Component Type. See clause 6.3.1 of [1].
- **DSCTy**—Data Service Component Type. See clause 6.3.1 of [1].
- **P/S Flag**—Indicates if a component is the primary (0) or a secondary (1) component of a given service.

In general, the host will also want to collect the component information for services that look interesting. This is done with the DAB_GET_COMPONENT_INFO command shown below. This command returns the following information.

- Component Label from Fig 0/5
- Component Language
- Component Global Service Identifier.
- Component User Application Information from Fig 0/1

**Table 15. DAB_GET_COMPONENT_INFO Command**

| DAB_GET_COMPONENT_INFO Command | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0xBB | | | | | | | |
| **ARG1** | 00000000 | | | | | | | |

**Table 16. DAB_GET_COMPONENT_INFO Response**

| Function | Description | | Size (Bytes) |
|---|---|---|---|
| **Global ID (SCIdS)** | The global reference for the component. See clause 6.3.5, Fig 0/8 of [1]. The SCIdS gets concatenated with the Service ID of the service list to form the full global component ID. | | 1 |
| **Align Pad** | | | 1 |
| **Language** | The language the component is in. See tables 9 and 10 of [2] | | 1 |
| **Component Label Character Set** | The character set for the component label.<br>Note: Component Label information may be changed in a future rev to support long component labels. | | 1 |
| **Component Label** | The textual label for the service component. If the first byte of the label is 0, this implies that the label does not exist. In this case the entire label will be 0 and the character set and abbreviation mask will also be zero.<br>**Note:** Component Label information may be changed in a future rev to support long component labels. | | 16 |
| **Component Label Abbreviation Mask** | Each bit set in this 16 bit field represents a character that should be used in the label above when abbreviating the label.<br>**Note:** Component Label information may be changed in a future rev to support long component labels. | | 2 |
| **Num** µA | This number of user applications supported for this component. | | 1 |
| **Align Pad** | | | 1 |
| **User Application Information** | RFU[15:11] | UAType[10:0] | 2 |
| | RFU[7:5] | Length[4:0] | 1 |
| | µA DATA (See Fig 0/13 definition in [1], clause 8.1.20) | | Length |
| **Align Pad** | If needed. Add 1 byte if the value of length is even. | | 1 (If Needed) |
| | The User Application information field is repeated up to 5 times for each user application supported. This number of µA Types is given in the NUM µA field of this table. This is only applicable for application data transported over XPAD. | | |

## 7.6. Starting a Digital Service

Once a data service has been identified in the service list the host starts the service using the START_DIGITAL_SERVICE command. This command is depicted below in Table 17. In the command the host provides the service ID and the component ID as shown. Both the service ID and component ID are returned in the service list described previously. When stating a data service the service type (SERTYPE) field should always be written as 1.

**Note:** The SERTYPE argument is not needed for DAB applications and should be written as 0.

**Table 17. Start Digital Service Command**

| START_DIGITAL_SERVICE Command | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x81 | | | | | | | |
| **ARG1** | 0000000 | | | | | | | SERTYPE |
| **ARG2** | 00000000 | | | | | | | |
| **ARG3** | 00000000 | | | | | | | |
| **ARG4** | SERVICE_ID[7:0] | | | | | | | |
| **ARG5** | SERVICE_ID[15:8] | | | | | | | |
| **ARG6** | SERVICE_ID[23:16] | | | | | | | |
| **ARG7** | SERVICE_ID[31:24] | | | | | | | |
| **ARG8** | COMP_ID[7:0] | | | | | | | |
| **ARG9** | COMP_ID[15:8] | | | | | | | |
| **ARG10** | COMP_ID[23:16] | | | | | | | |
| **ARG11** | COMP_ID[31:24] | | | | | | | |

Once a service is started, data packets for the service will begin appearing at the DSRV interface. If an attempt is made to start a service that does not exist, the error bit (bit 6) of the STATUS_WORD is retuned along with reason code 0x03 (Not available) in the 5th byte of the START_DIGITAL_SERVICE response. Note that in all cases byte 5 only acts as an error code when the ERR_CMD bit is set in the STATUS_WORD.

## 7.7. Getting the Digital Service Data

Data service packets are collected using the GET_DIGITAL_SERVICE_DATA host command. All data service data is returned using this interface. This includes both PAD and PSD data for the DAB and HD radio applications respectively. This command is shown in Table 18.

**Table 18. Get Data Service Data Command**

| GET_DIGITAL_SERVICE_DATA Command | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **CMD** | 0x84 | | | | | | | |
| **ARG1** | 000 | | | STATUS_ONLY | 000 | | | ACK |

When data service packets are ready for consumption they are placed into a holding (DSRV) queue until collected. This queue gives the host some flexibility in terms of when the data is collected but the data must be collected before the queue overflows. When the GET_DATA_SERVICE_DATA command is issued the next packet in the DSRV queue is de-queued and placed in the command interface. This data packet will remain available to the host until the next host command is issued.

The host may issue the READ_REPLY command as many times as needed without losing the packet data. This gives the host the option of reading the header first with one READ_REPLY transaction then reading the entire packet with payload with a second READ_REPLY transaction.

Instead of issuing multiple READ_REPLY commands the host may instead issue the READ_MORE command to first read out the DSRV packet header to determine the size of the payload and then read the remaining bytes of the payload using the READ_REPLY command.

Both of the methods for retrieving the packet data described above require no overhead from the Si468x software and are therefore very efficient. The diagram below depicts the DSRV queuing mechanism.

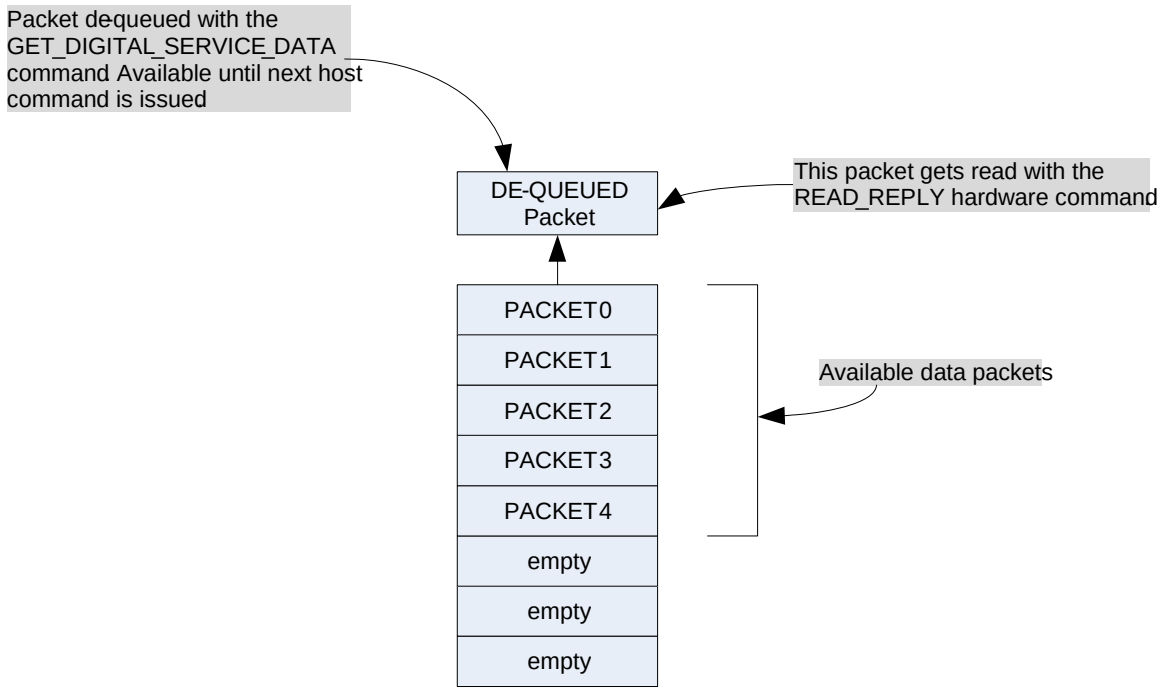



**Figure 6. DSRV Queuing Mechanism**

The following command options are available.

- ACK: When this bit is set all interrupts associated with the GET_DATA_SERVICE_DATA command and the DSRVINT bit of the STATUS_WORD are cleared on the next reading of the service data.
- STATUS_ONLY: This option lets host determine the status of the DSRV interface and also determine the number of packets available for consumption.

The response for this command is shown in Table 19. The fields in this response are discussed throughout the remainder of this section.

**Table 19. Get Data Service Data Response**

| GET_DIGITAL_SERVICE_DATA Reply | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| **STATUS0** | CTS | ERR_CMD | DACQINT | DSRVINT | RSQINT | RDSINT | ACFINT | STCINT |
| **STATUS1** | DBGINT | RSVDINT_14 | DEVNTINT | RSVDINT_12 | RSVDINT_11 | RSVDINT_10 | RSVDINT_9 | RSVDINT_8 |
| **STATUS2** | RSVD_STAT0[7:0] | | | | | | | |
| **STATUS3** | PUP_STATE[1:0] | | RSVD_STAT1[1:0] | | REPOFERR | CMDOFERR | ARBERR | ERRNR |
| **DATA4** | XXXXX | | | | | DSRVER-RINT | DSRVOV-FLINT | DSRVINT |
| **DATA5** | BUFF_COUNT[7:0] | | | | | | | |
| **DATA6** | SRV_STATE[7:0] | | | | | | | |
| **DATA7** | DATA_TYPE[7:0] | | | | | | | |
| **DATA8** | SERVICE_ID[7:0] | | | | | | | |
| **DATA9** | SERVICE_ID[15:8] | | | | | | | |
| **DATA10** | SERVICE_ID[23:16] | | | | | | | |
| **DATA11** | SERVICE_ID[31:24] | | | | | | | |
| **DATA12** | COMP_ID[7:0] | | | | | | | |
| **DATA13** | COMP_ID[15:8] | | | | | | | |
| **DATA14** | COMP_ID[23:16] | | | | | | | |
| **DATA15** | COMP_ID[31:24] | | | | | | | |
| **DATA16** | RFU[7:0] | | | | | | | |
| **DATA17** | RFU[15:8] | | | | | | | |
| **DATA18** | BYTE_COUNT[7:0] | | | | | | | |
| **DATA19** | BYTE_COUNT[15:8] | | | | | | | |
| **DATA20** | SEG_NUM[7:0] | | | | | | | |
| **DATA21** | SEG_NUM[15:8] | | | | | | | |
| **DATA22** | NUM_SEGS[7:0] | | | | | | | |
| **DATA23** | NUM_SEGS[15:8] | | | | | | | |
| **DATA24** | PAYLOAD0[7:0] | | | | | | | |
| **DATA25** | PAYLOADN[7:0] | | | | | | | |

### 7.7.1. GET_DIGITAL_SERVICE_DATA Configuration Parameters

The DSRV interface can be configured to control the DSRVINT bit of the STATUS_WORD and generate hardware interrupts to the host. There are three interrupt sources available for this purpose. These sources are as follows.

1. DRSPCKTVINT: This interrupt source indicates that a new data packet has been forwarded to the DSRV queue by the digital radio module. The host should respond to this interrupt by reading a packet(s) from the DSRV interface with the GET_DIGITAL_SERVICE_DATA command. This bit will remain set as long as there are buffers (packets) available in the DSRV queue.
2. DSRVOVFLINT: This interrupt source indicates that the DSRV queue has overflowed. This indicates that the host is not pulling data service data from the Si468x as fast as it is being produced. The host must take corrective action. The most recent data remains in the queue under this condition. Older is discarded.
3. DSRVERRINT: This interrupt source indicates that there was an error associated with one of the enabled data services at the physical layer (channel/demod). This could be due to a signal fade or some other problem in the channel. The affected service will report an error in the status indicator of the DSRV header. This interrupt is not available in HD Radio mode.

In all cases these indicators will appear in the INTSRC response of the GET_DIGITAL_SERVICE_DATA command if they occur. Use the command's ACK option to clear the indicators described above. If the host wishes to have one or all of these interrupt sources set the DSVRINT bit of the STATUS_WORD and or generate a hardware interrupt, the DIGITAL_SERVICE_INT_SOURCE property (0x8100) is used. Simply set the appropriate enable bit in the property.

### 7.7.2. Interrupt Method (Most Efficient)

Having the host react to interrupts from the Si468x device is generally the most efficient way to deal with data services. This eliminates the need for polling and the overhead associated with it. To generate a DSRV hardware interrupt the host must set the DSRVIEN bit of the INT_CONTROL_ENABLE property (0x0000) in addition to setting the appropriate bits of the DIGITAL_SERVICE_INT_SOURCE property. Please see the descriptions of the DIGITAL_SERVICE_INT_SOURCE and the INT_CONTROL_ENABLE properties in the host command API documentation for more details on enabling these interrupts.

A basic interrupt service routine for dealing with Si468x events is shown in the code snippet below. The interrupt handler simply schedules a Si468x maintenance task and returns; this accomplishes three things.

1. It keeps the Si468x maintenance loop from running when it does not need to.
2. It reduces the overhead that would normally be required on the Si468x to deal with the commands used to poll the device.
3. Helps aligns the timing of the maintenance task to the needs of the Si468x.

```
void Si46xx_interrupt_handler(void)
{
      /*
         On interrupt schedule the Si46xx maintenance task and
         return form the interrupt.
      */
      schedule_dsrv_handler();
      return;
}
```

**Code Snippet 1. Si468x Interrupt Service Routine**

### 7.7.3. Status Polling Method

Status polling is the second most efficient method for getting data service data off of the Si468x. In the status polling method the host must set the appropriate bits of the DIGITAL_SERVICE_INT_SOURCE property and poll the STATUS_WORD at a reasonable rate. When the DSRVINT bit is set, the host then executes a data service maintenance routine to collect and dispatch the DSRV data.

Depending on the rate at which data is produced by the Si468x the host must poll the STATUS_WORD at a given rate. As an example if it is know that data is produced at a fixed 64 kbps rate and that the minimum packet payload size returned by the Si468x is 1024 bytes and the desired maximum queue depth is 4 (the maximum DSRV queue depth on the Si468x is 8 deep), then the poll rate is given by:

$$Rate_{poll} = \frac{Q_{Max} \times N_{min}}{K_{max}}$$

**Equation 2. Poll Rate Assuming a Uniform Distribution of Packet Delivery**

Where $Rate_{Poll}$ is the rate at which the system should be polled. $Q_{Max}$ is the desired queue depth; $N_{min}$ is the expected minimum number of bits per packet, and $K_{max}$ is the expected maximum data rate. For the example cited, this equates to a poll rate of about 0.5 s. When running in HD Radio mode it is recommended that the poll rate be on the order of one HD audio frame which is 46ms.

A typical polling routine might look something like the following.

```
void Si46xx_status_poll(void)
{
      uint32_t status_word;
      uint16_t status_bits;

      status_word = get_Si46xx_status();

      /* Mask off the status bits and handle each as needed. */
      status_bits = status_word & 0x0000FFFF;

      if (status_bits & dsrv_mask) {
            handle_dsrv();
      }

      if (status_bits & other_stuff_mask) {
            handle_other_stuff();
      }
      .
      .
      .
      return;
}
```

**Code Snippet 2. Si468x Polling Function**

The get_Si468x_status() function simply reads 4 bytes from the command interface and returns those four bytes as an unsigned 32 bit word. The function handle_dsrv() should look something like the following.

```
#define DSRV_INTSRC_OFFSET     (4)
#define DSRV_DATA_READY_MASK   (1)
#define DSRV_OVERFLOW_ERR_MASK (2)
#define DSRV_PHYERR_MASK       (4)
void handle_dsrv(void)
{
      uint8_t *p_reply_buffer;
      uint8_t dsrv_int_source;
      uint8_t packets_available;

      /* Read the dsrv data using the GET_DIGITAL_SERVICE_DATA
         command. In this example the reply buffer is pointed
         to by p_reply_buffer. This buffer is assumed to be
         controlled by a buffer management scheme not shown
         but is assumed to remain valid past the scope of this
         function.
      */
      get_dsrv_data(&p_reply_buffer);

      dsrv_int_source   = p_reply_buffer[DSRV_INTSRC_OFFSET];

      /* If DSRV data is ready, dispatch it to the appropriate decoder */
      if (dsrv_int_source & DSRV_DATA_READY_MASK) {
            dispatch_dsrv_data(p_reply_buffer);
      }

      /*
         A DSRV packet overflow has occurred. This indicates that the host
         is not collecting the data packets fast enough. If polling the
         poll rate must be increased, if interrupt driven the queue depth
         property is set to deep.
      /*
      if (dsrv_int_source & DSRV DSRV_OVERFLOW_ERR_MASK) {
            handle_dsrv_overflow();
      }

      /* Phy error, except packets with errors or missing packets. */
      if (dsrv_int_source & DSRV_DATA_READY_MASK) {
            handle_dsrv_phy_err();
      }

      return;
}
```

**Code Snippet 3. Si468x DSRV Handling Function**

#### 7.7.4. DSRV Polling Method (Least Efficient)

The least efficient way to use the DSRV interface is to simply poll the DSRV interface using the GET_DIGITAL_SERVICE_DATA command. If doing this the poll rate defined in equation 1 above should be used.

## 7.8. DSRV Error Handling

In general there is nothing that can be done when errors occur due to signal loss or other impairments. Both the DAB and HD Radio systems employ some form of repetitive transmission of service data to deal with these types of issues. However, if an DSRV overflow condition is detected the host system is not servicing the DSRV queue quickly enough. In this case the host must poll the DSRV interface and collect the data service data at a faster rate. If the overflow condition does occur the data in the queue will always be the most recent data received.

## 7.9. Data Service Formats

This section describes the format in which data is returned from the Si468x device for the HD Radio and DAB modes of operation.

## 7.10. DAB

The service data for data services is returned in three possible modes. The format in which the data is returned depends on the transmission mode of the data. The following modes are possible.

1. Stream Mode: Data is transmitted in a single DMB sub channel. The packetization of this data depends on the DSCType. Data transmitted in stream mode will have a TMId of 1 in the data service list. In general this will include DSCType of TDC and MPEG-II Transport Stream. Please see the details below for these data service component types.
2. Packet Mode/No Data Groups: Data transmitted in Packet Mode without data groups will be passed to the host as packets from a logical frame with CRC. A data service that is transmitted in Packet Mode/No Data Groups will have a TMId of 3 and the DG flag will not be asserted in the service list.
3. Packet Mode/With Data Group: Data is transmitted in Packet Mode with Data Groups. A data service that is transmitted in Packet Mode with Data Groups will have a TMId of 3 and the DG flag will be asserted in the service list. Data that is transmitted in mode will have it's data presented at the data group layer as defined in clause 5.3.3 of [1]. Data group packets will be up to 8191 bytes in length.

Figure 7 shows the level at which data is forwarded to the host for Packet mode based on the state of the data group flag.

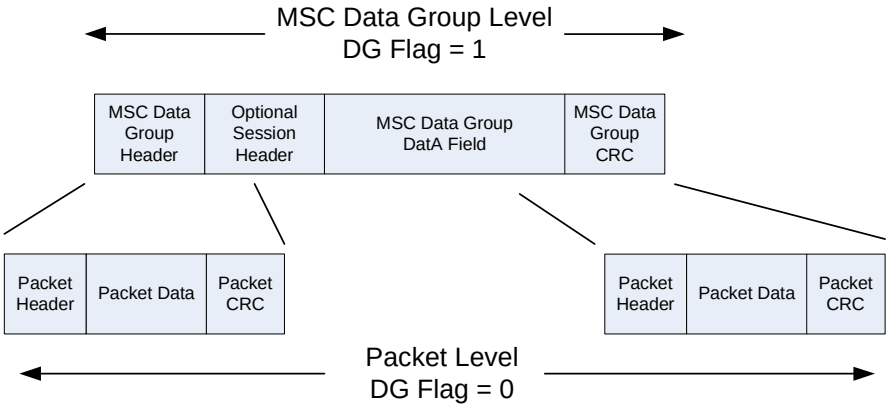



**Figure 7. Packet Mode Payload Definitions**

The following subsections outline the format of the returned data for each data service component type.

#### 7.10.1. Unspecified Data

Since this data format is unspecified the Si468x does not provided any special forwarding capabilities for data services transporting using the data service component type.

### 7.10.2. Traffic Message Channel

DAB transports TMC messages using the Fast Information Data channel (FIDC). TMC-user and TMC-system messages are encoded according to ISO 14819-1 (TTI).

The TMC messages will be passed as one or more 37 bit messages with a maximum payload of 6 messages or 28 bytes. Unused bits located at the end of the payload and will be set to 0.

The TMC system will be passed as one or more 16 bit messages with a maximum payload of 14 messages or 28 bytes. Unused bits located at the end of the payload and will be set to 0.

User Application Information will be passed in the DAB_GET_COMPONENT_INFO command.

Please see reference [4] for full details of TMC for DAB

Note: We are still working out some details of this DSCTy. Complete information will be included in a future revision of this document.

### 7.10.3. Emergency Warning System

**Note:** We are still working out the issues and details of this DSCTy. Complete information will be included in a future revision of this document, but the implementation will be similar to TMC.

### 7.10.4. Interactive Text Transmission

This data service component type appears to be obsolete and is currently not supported.

### 7.10.5. Paging

**Note:** We are still working out the issues and details of this DSCTy. Complete information will be included in a future revision of this document, but the implementation will be similar to TMC.

### 7.10.6. Transparent Data Channel

TDC data can be provided in all data transport mode described above. Please refer to the user application definition for applications transported using the TDC to determine the transport mode used. Based on the transport mode the data will be presented as follows.

- Stream Mode: Presented as a stream of bytes in chunks of reasonable size (about 1k). No special framing or CRC is provided.
- Packet Mode / no Data Groups: Passed to the host as one or more packets from a logical frame with CRC. The format of these packets is entirely defined by the application type. The Si468x firmware performs no repackaging of this data. See Figure 7 for details.
- Packet Mode / with Data Groups: Passed to the host at the MSC data group level as defined in 5.3.3 of [1]. See the specification for specific applications to determine the contents of the data group payload. Please see figure 2 for details.

### 7.10.7. MPEG-2 Transport Stream

The MPEG-2 Transport Format is defined in [5]. This data is transported in steam mode as defined above. Data for this type is DSCType is presented to host as complete TS packets.

The number of packets returned in a single payload can vary but is nominally on the order of 8 packets. Assuming the standard 188 byte TS packet this equates to payload size of 1504 bytes nominally. Figure 8 shows the format of the data returned when the DSCType is MPEG-2 Transport Stream. Also see reference [5] for details on DAB MPEG-2 TS Streaming.

**Figure 8. MPEG-2 Packet Definition**

### 7.10.8. Embedded IP Packets

IP Packets are presented in packet mode with data groups. Therefore the data is presented at the data group level as defined in clause 5.3.3 of [1]. The payload in the data group packet will be a single IP packet. See Figure 7 on page 346 for details.

## 7.11. MOT

MOT is transmitted as MOT segments at the data group level. Therefore the MOT data will be presented as defined in 5.3.3 of [1] as segmented MOT objects. Please also see the definition of MOT transport for DAB in [3].

The Si468x shall forward MOT data groups in the order of arrival. The assembly of MOT data groups into MOT segments shall not be done by the Si468x, and left to the MOT decoder software on the host.

The DAB MSC can carry more than one MOT object at the same time. In order to assemble MOT segments, the host needs to join only MOT applications messages associated with the service identified by the service ID and component ID.

### 7.11.1. Proprietary Service: no DSCTy Signaled

Not well defined. The Si468x does not provide special forwarding of this data service component type.

## 7.12. HD Radio Data Services

In HD Radio mode the format for the data payload returned by the GET_DIGITAL_SERVICE_DATA command is defined in reference [8]. Please refer to this document for the details of the HD Radio payloads. The Silicon Labs GET_DIGITAL_SERVICE_DATA command returns the payload as defined by the GET_PORT_DATA function code that is defined in table K-5 of reference [8]. Only the AAS Data Packet portion of the GET_PORT_DATA response is returned in the GET_DIGITAL_SERVICE_DATA command's payload. The remaining information associated with this command is returned in the DSRV header. Exceptions are the function code and the Num_Pkts_Avail. The Num_Pkts_Avail field is not required because every packet associated with an enabled port is forward to the host through the DSRV interface as soon as it becomes available (i.e., Num_Pkts_Avail should never exceed 1). Specific payload formats for a given service type should be available from the service provider. For example information about traffic data payloads from NavTeq should be available from the Navteq service provider and a Navteq decoder is required on the host before this service can be used. Please review the information of references [7,8], "7.5.1. HD Radio Service List" on page 336, and digital service APIs for complete details on HD Radio Data services.

### 7.12.1. A Note About LOT

LOT is the HD Radio protocol that is used for transferring large files over the air. LOT stands for Large Object Transfer. The LOT standard includes methods for object reassembly and retrieval. The Si4680 does not support the reassembly of large data objects and therefore the host must have a LOT decoder if LOT is to be supported. The Si4680 device does support LOT data services just like any other data service but it is incumbent upon the host to decode and reassemble LOT encoded objects. Services that are transferred using the LOT protocol can be identified by the Processing Field of the component in the service list. Refer to [8] and [9] for more details regarding LOT. Please also contact iBiquity Digital Corporation for information on obtaining LOT a decoder for a host system.

## 7.13. Audio Related Data Services

Both the DAB and HD Radio standards support data services that are closely coupled with the audio content. In DAB this is referred to as Program Associated Data (PAD) and in the HD Radio standard this is referred to as Program Service Data or PSD. The Si4680 device supports the transport of both of these meta data services through the DSRV interface. The following subsections describe how this data is retrieved from the Si4680 device. In both case a host decoder is required to decode the PAD/PSD payloads.

## 7.14. DAB Program Associated Data (PAD)

PAD is closely coupled to the audio content. In fact, PAD is transported along with the audio data in an audio subchannel. Please see [1] Section 7.4 and [6] Section 5.4 for more details regarding the transport of PAD. This data must be extracted from the audio stream before it can be used. The Si4680 device performs the task of extracting the PAD and forwards it automatically to the host when a audio program is playing. The PAD data is forward with a service ID and Component ID that match the audio program.

PAD can support multiple types of data services including DSL, DLS Plus, and MOT. See the [1] for full details on which service types are supported.

### 7.14.1. The GET_DIGITAL_SERVICE_DATA Command DATA_TYPE Field

The data type field will be used in cases where a service component can forward multiple types of data. Right now the only service components that can do this are the DAB/DAB+ DL message, DL+ tags and Audio services PAD. The case of DAB, the DATA_TYPE field is defined in Table 20 below.

**Table 20. Definition of the DATA_TYPE Field of the DSRV Header (DAB Mode**

| Data/PAD/AUDIO[7:6] | TYPE[5:0] |
|---|---|

The Data/PAD/Audio and TYPE sub-fields shown in figure 4 are used to convey the type of data transported in the DSRV payload. The meaning of these fields is shown in the list below.

- Data/PAD/Audio = 00: Standard data channel for data services not related to audio.
  - TYPE = 0, RFU.
- Data/PAD/Audio = 01: Data over PAD , Non-DLS, DLS+
- TYPE: conforms to Table 2: DSCTy types found in [2], selected values are
  - TYPE = 0: unspecified data
  - TYPE = 1: TMC
  - TYPE = 5: TDC/TPEG
  - TYPE =60: MOT
- Data/PAD/Audio = 10: DLS/DL+ over PAD for DLS services
  - For now Type = 0, RFU
- Data/PAD/Audio = 11: Audio (RFU; Audio Currently not transported over the DSRV interface)
  - Type = 0 for MPEG I or II foreground
  - Type = 1 for MPEG I or II Background
  - Type = 2 for multichannel MPEG II

**Note:** This field will always be set to 0 for FMHD (DATA).

### 7.14.2. DLS/DLS+

When Data/PAD/Audio is equal to binary 10 the data packet is PAD-DLS or DLS+. The Si4680 device will completely reconstruct all DLS messages and commands and forward them to the host. In this case the payload will contain an additional payload header that indicates needed information about the payload. The following subsections define this addition payload header and describe how the host should use this information. It is expected that the reader is familiar with the DLS specification as defined in [1] and ETSI TS 102 980.

#### 7.14.2.1. The Prefix (Payload Header) for DLS/DLS+

For DLS and DLS+ a two byte prefix is added to the payload. This prefix conforms to the DL+ spec (ETSI TS 102 980) in conjunction with the XPAD DLS definition as defined in [1] Section 7.4.5.2 and is defined in Table 21 below. Since the Si4680 reconstructs the DLS messages and commands not all elements of the prefix and commands as defined in [1] are required so elements used for message reconstruction are not included in the header. Only those elements of the commands and messages needed for display and message parsing are include a prefix as required by the spec.

**Table 21. Definition of the Payload Header (Prefix) when DLS/DLS+ is transported in the DSRV**

| Toggle[7] | RFU[6:5] | C[4] | (Field 1) C=1, Command[3:0]/ C=0, 0 |
|---|---|---|---|
| (Field 2) C=1, Link[4]/C=0 Charset[7:4] | | | RFU[3:0] |

The first two bytes of the header shall match the definition of the XPAD DLS prefix. Only elements relating to the entire payload are included. Elements use to reconstruct the payload are not needed and thus omitted.

**Note:** There are only two valid values for the command filed. They are 0001 (remove label from display) and 0010 the DL Plus command. In all other cases the command filed should be written as 0000.

#### 7.14.2.2. The Payload for DLS/DLS+

The DLS/DLS+ payload shall consist of either a reassembled message or reassembled command body as defined in [1] Section 7.4.5.2.

When a DL message is indicated by a value of 0 in the C field of the prefix, the messages follows immediately after the prefix. In this case, field 1 is zero and field two contains the character set.

**Table 22. A DLS Message**

| Toggle[7] | RFU[6:5] | C[4] | (Field 1) C=0, 0 |
|---|---|---|---|
| (Field 2) C=0 Charset[7:4] | | | RFU[3:0] |
| Msg (variable length) | | | |

When a DL Plus Command is indicated value of 1 in the C field and by a value of 0010 in the command field of the prefix, the DL Plus command follows the prefix. The structure of the DL Plus command is defined in clause 7.2 of ETSI TS 102 980.

**Table 23. DLS Tags Command with Prefix**

| Toggle[7] | RFU[6:5] | C[4] | (Field 1) C=1, Command[3:0] |
|---|---|---|---|
| (Field 2) C=1, Link[4] | | | RFU[3:0] |
| CID[7:4] (currently only 0000, DL Plus TAGS command) | | | CB (For Tags Command it is IT[3], IR[2], NT[1:0] |
| tags (variable length) | | | |

## 7.15. HD Radio PSD

For HD Radio PSD is supported through the DSRV interface. The PSD payload is transferred to host in the format defined in [7] and an ID3 decoder is required to render the PSD information. Unlike DAB, PSD forwarding is not enabled by default and must be enabled using the HD_PSD_ENABLE property. Additionally PSD can be enabled for any audio program that exists in the ensemble even if it is not being listened to. Please review the HD_ENABLE_PSD property for details.

When PSD data is forward to the host it is identified by unique port numbers (component IDs) in the DSRV header. The MPS is identified by Component ID equal to 0x5100 and SPS1-SPS7 are identified by a component ID of 0x5201-0x5207 respectively.

### 7.15.1. Simple PSD interface

If the host wishes to view only the PSD information for the currently playing audio service and simplified PSD interface is provided. When the simplified interface is used no ID3 parser is required but not all PSD information is available. Please see the HD_GET_PSD_DECODE command and the HD_PSD_FIELD_MASK for full details on using this interface and the limitations of using it.

## 7.16. How to Use FMHD Fast Detect in FMHD P2

Traditional FMHD tuner requires HD decoder to detect the presence of HD station, which can take about 1 sec. The Si468x can detect HD in 96 ms without the HD decoder. This is represented in the HD level metric. The metric is only calculated during a Tune/Seek command and reported from the RSQ status attune command after STC. This document describes Silicon Laboratories recommendations on how to use the HD level metric.

HD metric returns a value ranging from 0 to 100. A reported value greater than 30 indicates that there is HD signal at the present FM station. To enable HD level metric calculation function, the following conditions must be met:

- An HD tune mode is enabled during FM_TUNE_FREQ command (0x08 or 0x0C)
- In property FM_RSQ_HD_DETECTION (0x3307)
  - The Samples[7:0] are not set to zero value
  - The SNR attune metric is same or greater than the SNRThresh[7:0]

Property FM_RSQ_HD_DETECTION (0x3307) controls two parameters:

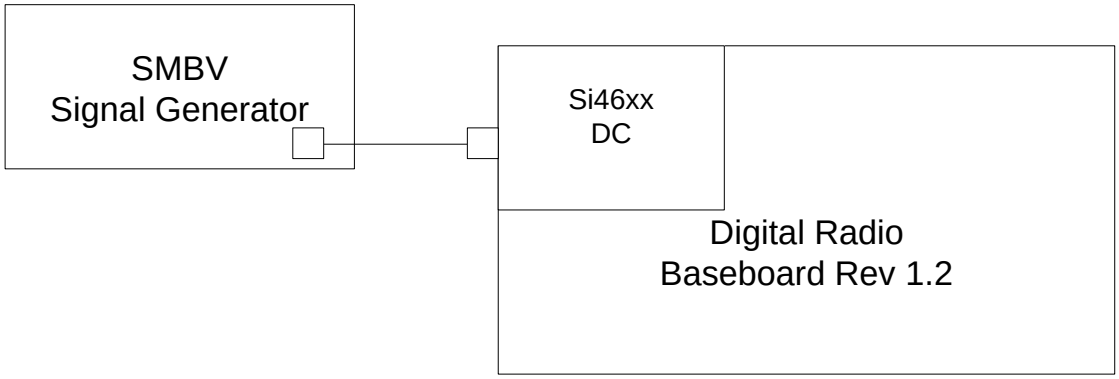
- Samples—Represents the HD detection calculation window in unit of 2.9 ms. Default is 0, which means HD level metric calculation is turned off. It is recommended to set this value to 32 to reliably detect the signal, which is 96ms for Tune/Seek time.
- SNR threshold—Sets the SNR threshold less than which the HD metric calculation is turned off. If the SNR RSQ attune value is below this threshold, and the RSSI RSQ attune value is below the SEEK_VALID_RSSI_THRESHOLD (property 0x3202) then the HD level detection will not run and the HD level metric will report as 0. This is helpful for Tune/Seek by not waiting extra time on an FM Frequency that doesn't have a valid RSSI or SNR, which will produce a normal Tune/Seek time for the given channel. The default value is 12, which is 2dB above the default of SEEK_VALID_SNR_THRESHOLD (property 0x3204). To always enable calculation of the HD level metric, set the HD detection SNR threshold to -128 and the SEEK_VALID_RSSI_THRESHOLD (property 0x3202) to -128.

### 7.16.1. Steps to Use HD Level Metric

1. Set Property FM_RSQ_HD_DETECTION (0x3307) for a valid SNR threshold and number of Samples. Silicon Laboratories recommends SNR threshold of 12 and number of samples or 32 (0x0C20).
2. Set Property SEEK_VALID_RSSI_THRESHOLD (property 0x3202) to a for a valid SNR threshold. Silicon Laboratories recommends SNR threshold of 17.
3. When Tuning, make sure to select a valid tune mode for FMHD (0x08 or 0x0C)
4. Call Tune/Seek command. If the SNR Threshold is met then Si468x part will run the HD Level Calculation for the number of samples. For 64 samples this time is 186mS.
5. At STC, call the FM_RSQ_STATUS Attune command (0x4) to retrieve the HD Level Metric
6. A value greater than 30 is suggested as a good HD station.

# 8. BER Test Setup and Procedure

## 8.1. BER Test Setup




Setup the radio with a signal generator. An example with Si468x EVB is shown in figure above. Signal generator is a SMBV from Rohde&Schwarz in this setup. Connect the RF out from signal generator to J1 (RF in) on the Si468x Daughter card. You can use the GUI to test BER on the EVB setup. To validate another design, follow the steps outlined in the next two sections.

## 8.2. DAB BER Test

**RF generator setup**:

1. Test vector: BER_NullPattern_test2.eti
   The content of this ETI file is shown below:

| ServiceID | BitRate | Protection | Mode | Pattern |
|---|---|---|---|---|
| 0xE1C00000 | 576 kbps | EEP-3A | Packet | 0b00000000 |
| 0xE1C00001 | 384 kbps | EEP-3A | Packet | 0b00000000 |
| 0xE1C00002 | 96 kbps | UEP-5 | Packet | 0b00000000 |
| 0xE1C00003 | 32 kbps | EEP-2A | Packet | 0b00000000 |

2. Setup to desired testing frequency, RF level.

**Si468x Setup**:

1. Power up the part.
2. Tune to the desired frequency.
3. Use GET_DIGITAL_SERVICE_LIST to confirm the vector content.
4. Use START_DIGITAL_SERVICE to start a service
   a. 0xE1C00000 for DMB image
   b. 0xE1C00001 for DAB image
5. Set property DAB_TEST_BER_CONFIG to enable the BER testing with a known pattern (e.g., Pattern 0x00).
6. Use DAB_TEST_GET_BER_INFO to get BER ERR_BITS and TOTAL_BITS info. BER = ERR_BITS/ TOTAL_BITS.

It is good practice to acquire more than 1 million bits for bits accumulation purpose.

## 8.3. FMHD BER Test

**RF generator setup**:

1. Test vector: Use the BER testing vector provided by iBiquity Co.
   Example: IB_FMr208c_e1wfc204.

   The content of this file is shown below:

| Service Mode | Station Short Name | Station Long Name |
|---|---|---|
| MP1 | N/A | N/A |
| Analog Audio Source | Digital Data | Channel Conditions |
| Music | BER Test Pattern | Clean Channel |

2. Setup to desired testing frequency, RF level.

**Si468x setup**:

1. Power up the part.
2. Tune to the desired frequency.
3. Set property FMHD_TEST_BER_CONFIG to enable the BER testing.
4. Use FMHD_TEST_GET_BER_INFO to get BITS_ERRORs and BITS_TESTED info. BER = BITS_ERROR/BITS_TESTED.

It is good practice to acquire more than 1 million bits for bits accumulation purpose.

# 9. POWER_UP—How to Determine Crystal-Related Parameters for Your Design

## 9.1. TR_SIZE and IBIAS settings

The crystal oscillator core gain is controlled by both **TR_SIZE** and **IBIAS** parameters. Each crystal design has different gain requirements to achieve successful startup. The following calculations are needed for each PCB design as they require adjustment for the board capacitance.

Five crystals have been characterized as reference. To achieve the startup requirements, an optimal **TR_SIZE** value is provided for each crystal frequency. These values are reflected in the table below:

| Crystal Frequency (MHz) | TR_SIZE |
|---|---|
| 37.2 | 0xF |
| 27.0 | 0xA |
| 19.2 | 0x7 |
| 12.288 | 0x5 |
| 6.0 | 0x4 |

If choosing a crystal frequency that is not in the table, the corresponding **TR_SIZE** can be extrapolated linearly. This calculation will be demonstrated later in this section.

For the above **TR_SIZE settings**, biasing current ranges (**IBIAS**) are then determined by the following:

- The **startup** ESR (typically, startup ESR should be 5x the specified Resr) values.
- Loading capacitance CL, **which includes the addition of internal tuning load capacitance (using CTUN), external load capacitance (if any) and board and IC equivalent load parasitic capacitance.**

The **IBIAS** selection process will also be demonstrated later in this section.

The following table depicts the biasing current programming. The biasing current is programmed linearly in 10 µA steps starting at 10 µA for IBIAS = 0x01 up to 1270 µA for IBIAS = 0x7F.

| IBIAS [6:0] | Ibias (µA) |
|---|---|
| 0x00 | 0 |
| 0x01 | 10 |
| 0x02 | 20 |
| ... | ... |
| 0x7E | 1260 |
| 0x7F | 1270 |

### 9.1.1. IBIAS and TR_SIZE for a 500 Ω Startup ESR Crystal

To select the appropriate IBIAS value, a set of two curves is determined for each startup ESR. These curves represent the minimum and maximum bias current that may be safely used versus the total load capacitance, given the startup ESR.

The bias current should be chosen between the minimum and maximum. The closer IBIAS is selected to the curves, the less safety margin has the startup conditions.

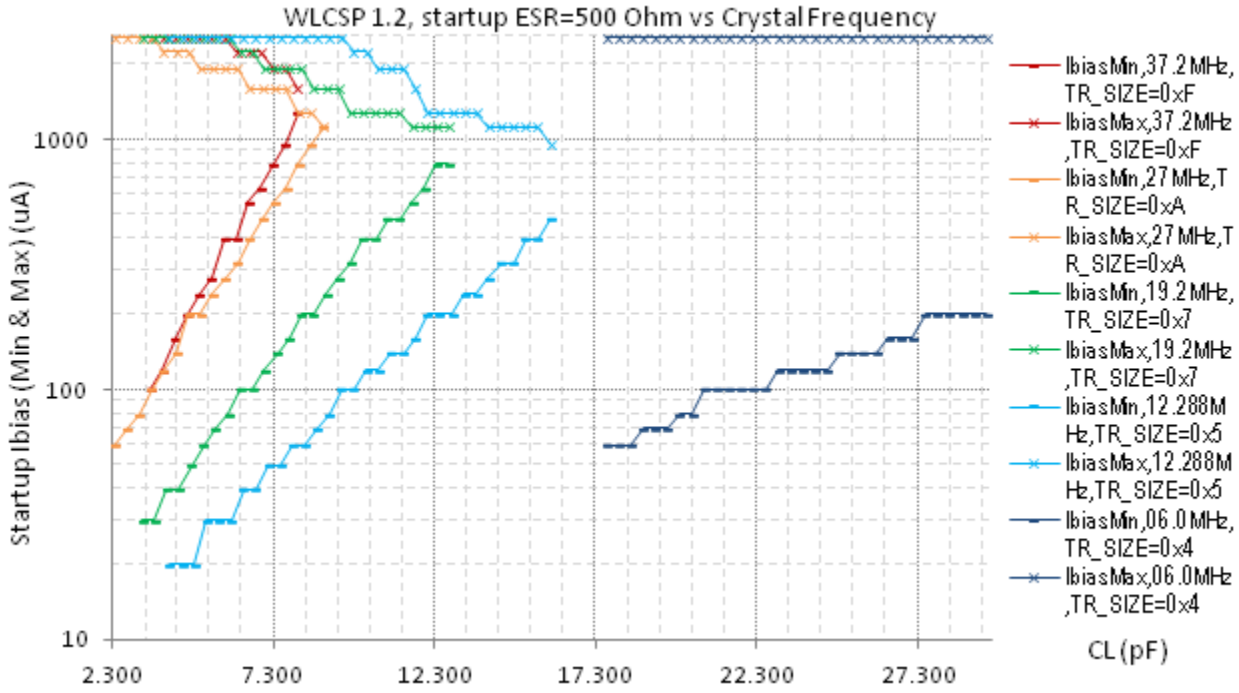



**Figure 9. Safe Range of Operation for a 500 Ω Startup ESR Crystal**

For different crystal frequencies, a linear interpolation from the two closest crystal frequencies may be used to determine TR_SIZE and IBIAS.

For the crystals depicted in the plot, the maximum $C_L$ recommended is the x axis value that still has a safe range of operation.

The maximum recommended $C_L$ for an arbitrary frequency crystal is not to exceed the maximum $C_L$ of the reference crystal with frequency immediately above the desired frequency.

Maximum $C_L$ may be increased using additional external Cx if the Ibias range hasn't disappeared for maximum plotted $C_L$. This is the case for the 6 MHz crystal, and the 19.2 MHz crystal for startup ESR = 250 Ω in the next plot.

### 9.1.2. IBIAS and TR_SIZE for a 250 Ω startup ESR Crystal

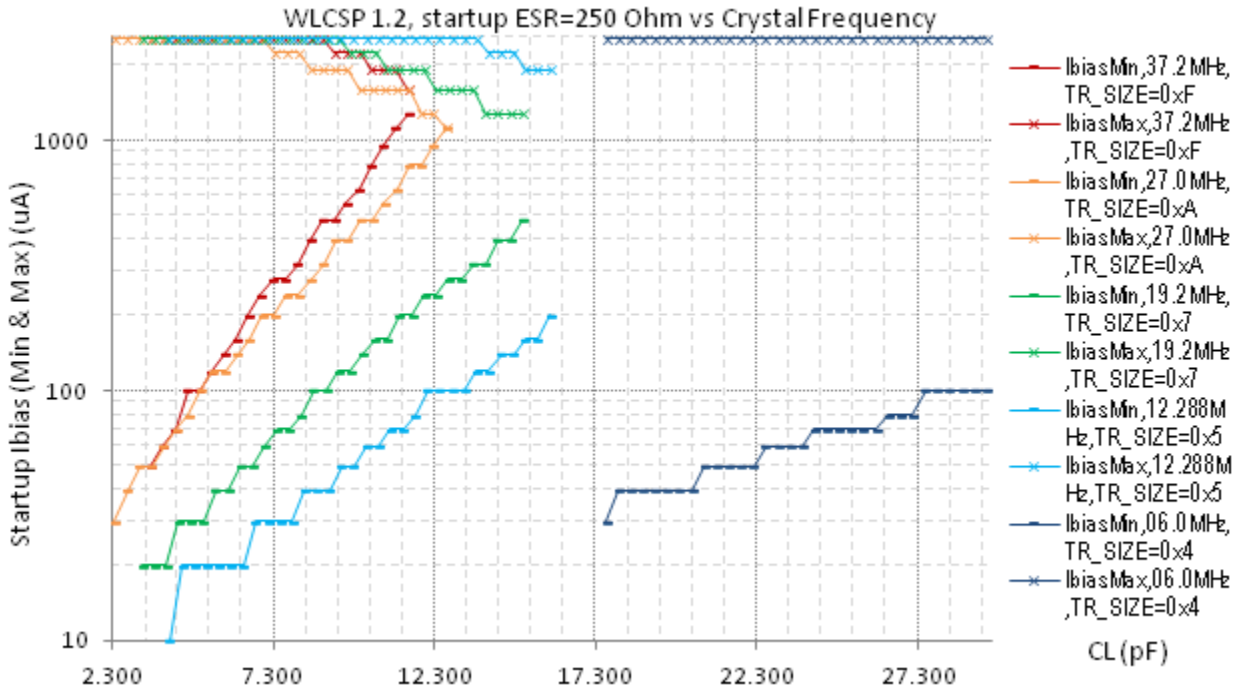



**Figure 10. Safe Range of Operation for a 250 Ω Startup ESR Crystal**

Illustration example for a chosen crystal of 24 MHz with Resr = 50 Ω and $C_L$ = 10 pF:

1. **Maximum $C_L$**: Looking at the safe operation range for the plotted frequency immediately above the desired frequency (27 MHz) and a startup ESR equal to 5x Resr (startup ESR = 250 Ω—Figure 10), one extracts the $C_L$ = 12.5 pF as the maximum $C_L$ with safe operating range. $C_L$ = 10 pF in this example is in the safe operation range.
2. **TR_SIZE**: For the chosen $C_L$ = 10 pF (x axis value), the recommended setting is obtained by linear interpolation between the TR_SIZE of plotted crystals with frequency immediately lower and higher. The calculation results in TR_SIZE = 0x9 (9), from 7 + (10 – 7) x (24 – 19.2)/(27 – 19).
3. **IBIAS**: For the chosen crystal frequency of 24 MHz and startup ESR of 250 Ω (Figure 10), the recommended IBIAS is determined by the linear interpolation between the IBIAS twice above the minimum or centered in the safe Ibias range, whichever lower, for the plotted crystals with frequency immediately lower and higher for the chosen $C_L$ = 10 pF (x axis).

The calculation results in a bias current of 700 µA (IBIAS = 70 = 0x46), which is
300 + (950-300) x (24 – 19.2)/(27 – 19.2).

### 9.1.3. IBIAS and TR_SIZE 37.2 MHz Crystal

The following plots are generated for customer's understanding of how ESR and CL affect IBIAS selection for a given XTAL frequency.

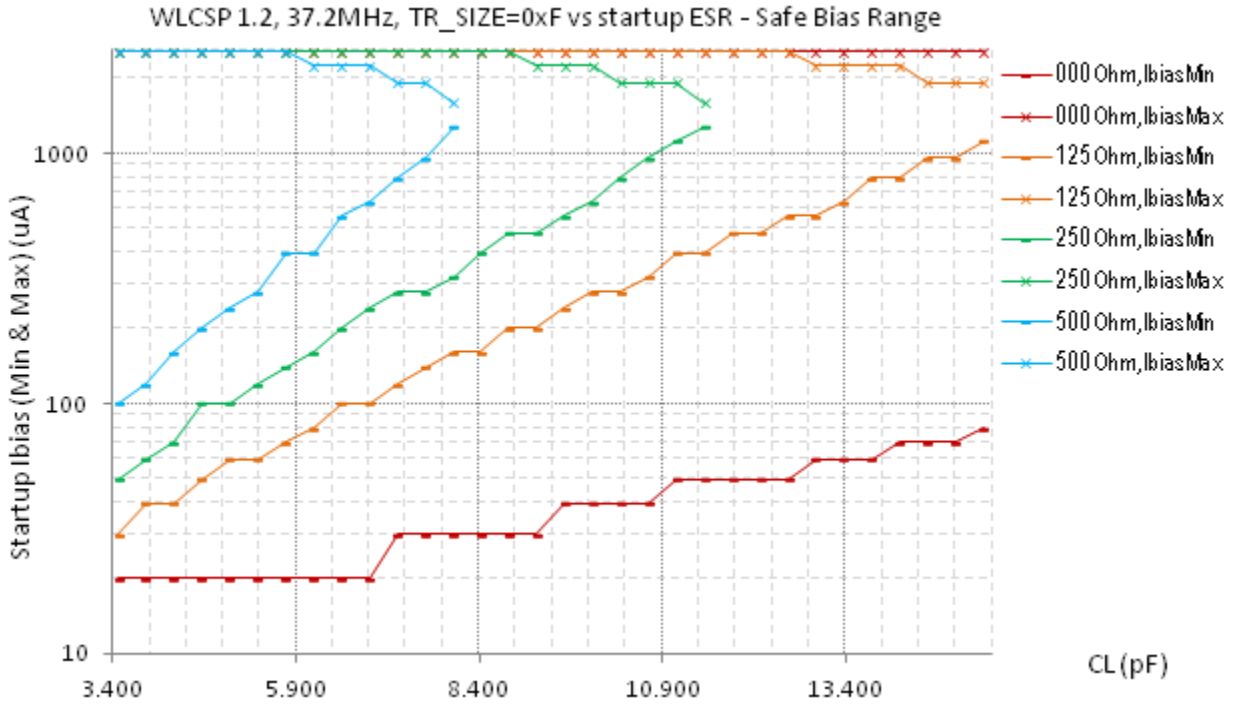



**Figure 11. Safe Range of Operation for a 37.2 MHz Crystal**

### 9.1.4. BIAS and TR_SIZE 27.0 MHz Crystal

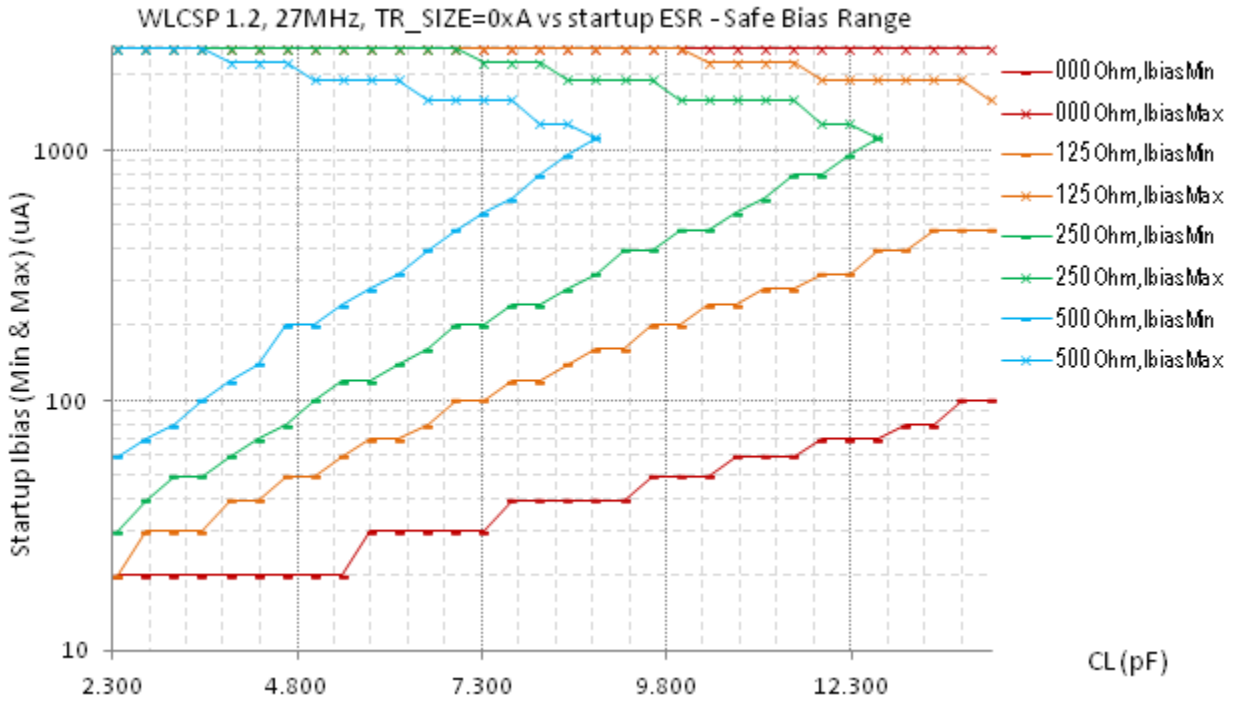



**Figure 12. Safe Range of Operation for a 27.0 MHz Crystal**

### 9.1.5. IBIAS and TR_SIZE 19.2 MHz Crystal

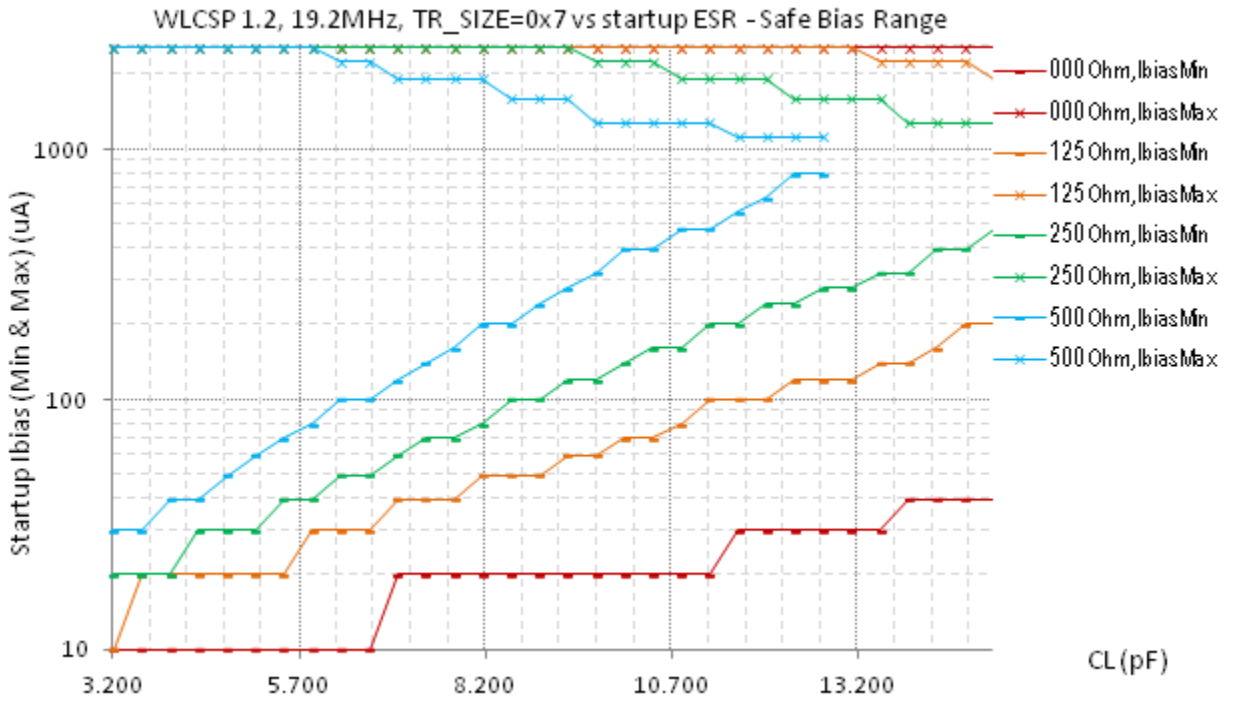



**Figure 13. Safe Range of Operation for a 19.2 MHz Crystal**

### 9.1.6. IBIAS and TR_SIZE 12.288 MHz Crystal

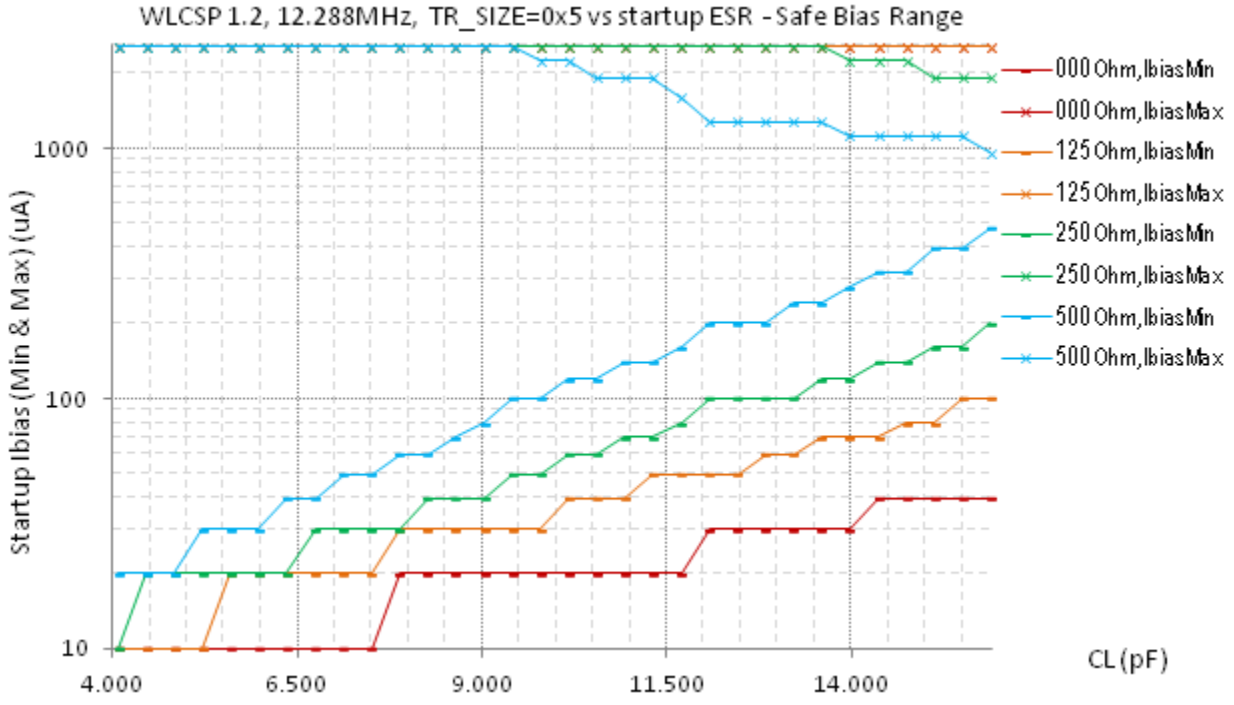



**Figure 14. Safe Range of Operation for a 12.288 MHz Crystal**

### 9.1.7. IBIAS and TR_SIZE 6.0 MHz Crystal

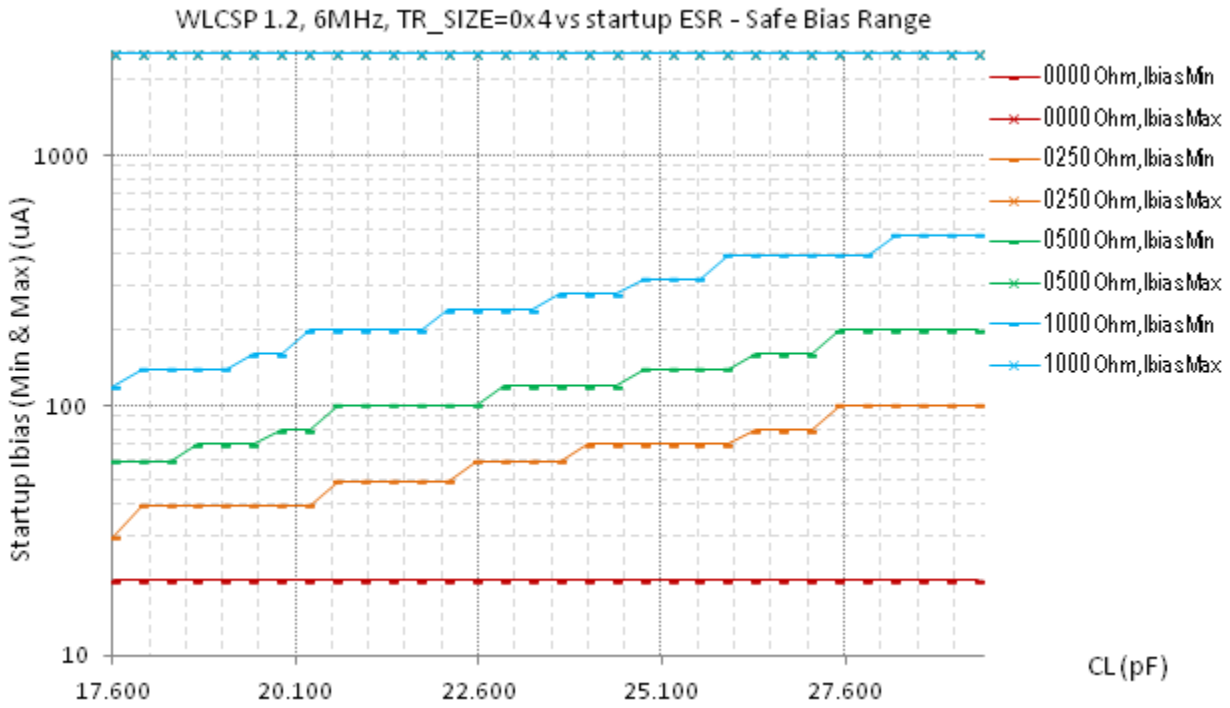



**Figure 15. Safe Range of Operation for a 6.0 MHz Crystal**

## 9.2. IBIAS Reduction in Steady State

Once the crystal oscillator completes the startup phase the model of the crystal resonator changes. The ESR resistor significantly decreases and the startup bias current is larger than needed to maintain safe oscillation conditions.

Furthermore, besides wasting current consumption, the crystal oscillator operation is deeper into nonlinear regime, generating stronger harmonic components which causes the appearance of unwanted spurs in tuner chain for specific tuning frequencies.

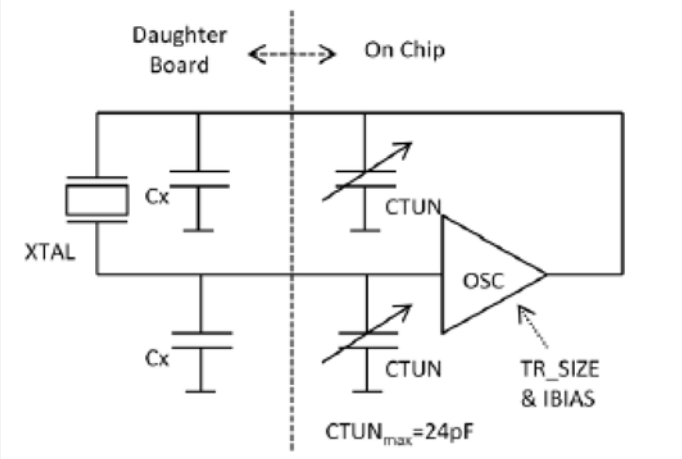
To reduce the impact of these issues, it is recommended to reduce the biasing current by 2x.

The model of the crystal and the safe operation areas may be interpreted by the increase of the safe operation area in the charts above when the startup ESR = 0 Ω, corresponding to the red lines.

## 9.3. CTUN settings

CTUN programs a pair of internal tuning capacitances inside the IC connected between each of the crystal pins and ground.

The equivalent contribution to $C_L$ is half of its value.




The tuning capacitances value is programmed linearly between 0 and 24 pF in 63 steps for CTUN starting at 0x00 up to 0x3F, respectively.

| CTUN [5:0] | CT (pF) | ΔCL (pF) |
|---|---|---|
| 0x00 | 0 | 0 |
| 0x01 | 0.381 | 0.190 |
| 0x02 | 0.762 | 0.381 |
| ... | ... | ... |
| 0x3E | 23.619 | 11.810 |
| 0x3F | 24.000 | 12.000 |

CTUN should be determined by the equation below:

$$CTUN = 2 \cdot (CL - CLpar) - Cx$$

- $C_L$ = XTAL load capacitance from the manufacturer's data sheet.
- CLpar = equivalent parasitic load capacitance in pF.
- Cx= the value of the external discrete capacitances between each of the crystal terminals and ground (if any). Cx is only required for crystals with $C_L$ larger than about 14 pF.

CTUN should be trimmed by measuring the crystal oscillator frequency for typical conditions.

## 9.4. External Clock Source

The Si468x may use an external clock source and bypass its internal crystal oscillator. To use this mode of operation, the crystal and any external load capacitances should be removed from the XTALI and XTALO pins.

The clock source should be connected to the XTALI pin directly in dc coupling mode (not ac coupling), a sine wave is recommended to avoid harmonic interference with the tuner. Signal wave levels should be 1.0 V≤ VH ≤ 1.4 V and 0 ≤ VL ≤ 0.3 V.

The following settings for the POWER_UP command should be used.

- CLK_MODE = 2
- TR_SIZE = 0
- IBIAS = 0
- IBIAS_RUN = 0

# Document Change List

## Revision 0.5 to Revision 0.6

- Updated Si46xx Programming Guide and Firmware Revisions table for release 120731.
- Updated flowchart for initialization to describe FLASH_LOAD process.
- Minor updates to BER Test procedure to highlight differences between DMB and DAB.
- Description clarifications in Command and Property Tables.

## Revision 0.6 to Revision 0.7

- Updated Si46xx Programming Guide and Firmware Revisions table for release 120914.
- Description clarifications in Command and Property tables.
- FMHD_CODEC_MODE family of properties added.
- FMHD_DIGRAD_STATUS: added response—CODEC_MODE.
- DAB_DIGRAD_STATUS: removed command input—CANCEL.
- DAB_DIGRAD_STATUS: added response—CU_LEVEL.
- DAB_GET_SERVICE_LINKING_INFO: added response—LINK_BYTES.
- FMHD_BLEND_OPTIONS: increased field size with new options—ACQ_LOSS.
- FMHD_BLEND_ANALOG_TO_HD_TRANSITION_TIME: property default value changed.
- FMHD_BLEND_HD_TO_ANALOG_TRANSITION_TIME: property default value changed.
- FMHD_PSD_ENABLE: property added to documentation.
- DAB_CTRL_DAB_MUTE_SNR_THRESHOLD: property added.
- DAB_CTRL_DAB_UNMUTE_SNR_THRESHOLD: property added.

## Revision 0.7 to Revision 0.8

- Updated Table 1, “Si468x Programming Guide and Firmware Revisions[1],[2],” on page 1.
- Updated Si46xx Programming Guide and Firmware Revisions table for release 121120.
- Updated initialization flowchart to include mini-patch.
- Separated command/property summaries for the DAB release.
- Added description for how to process PAD/XPAD data streams through the data service interface.
- Added APIs in DAB:
  - write_storage
  - read_storage
  - dab_get_audio_info
  - dab_get_subchan_info

## Revision 0.8 to Revision 0.9

- Updated Si46xx Programming Guide and Firmware Revisions table for release 130215.
- Removed Digital Radio Demodulator parts from this guide—a separate Application Note is available for those parts
- Updated FMHD APIs
  - Added WRITE_STORAGE and READ_STORAGE commands
  - Added HDDET, READANTCAP, HDLEVEL to FM_RSQ_STATUS
  - Added additional options to SERTYPE for GET_DIGITAL_SERVICE_LIST
  - Renamed commands with FMHD_... to HD_...
  - Added CDNRL, CDNRH, CODEC_MODE to HD_DIGRAD_STATUS
  - Added DINFOINT, DINFO, AINFOINT, AINFO, ALERTINT, PSDINT, PSD as well as individual indicators for available SIS and PSD content to HD_GET_EVENT_STATUS
  - Added HD_GET_PSD_DECODE to provide decoded PSD strings, rather than requiring ID3 decoding of PSD received through data service interface.
  - Added HD_GET_ALERT_MSG to provide support for HD Radio alerts.
  - Updated several parameters in DAB_DIGRAD_STATUS
  - Added properties related to HD Detection
  - Added additional interrupt sources to HD_EVENT_INTERRUPT_SOURCE
  - Added HD_TEST_BER_CONFIG
  - Optimized property defaults for better performance
- Updated DAB APIs.
  - Optimized property defaults for better performance
- Updated flowcharts for FMHD to include renamed "HD_" commands.
- Added "6. Managing Firmware" on page 313, which explains the available system architecture options for loading firmware onto the Si46xx.
- Updated "7. Digital Services User’s Guide" on page 334 to include more information about HD Radio

## Revision 0.9 to Revision 1.0

- Updated Si46xx Programming Guide and Firmware Revisions table for release 130524.
- Updated FMHD APIs
  - Added HD_PLAY_ALERT_TONE command
  - Changed FM_RSQ_HD_DETECTION property default and field definition
  - Changed HD_EVENT_ALERT_CONFIG property default
  - Changed HD_AUDIO_CTRL_BALL_GAME_ENABLE property default
  - Changed HD_DIGRAD_STATUS command to provide HDLOGO indication and interrupt bits
- Updated DAB APIs
  - Added DAB_DIGRAD_RSSI_HIGH_THRESHOLD property, added interrupts related to the threshold
  - Added DAB_DIGRAD_RSSI_LOW_THRESHOLD property, added interrupts related to the threshold
  - Added DAB_CTRL_DAB_MUTE_SIGLOW_THRESHOLD property
  - Changed DAB_TUNE_FE_CFG property default, added additional fields to this property
  - Changed the SERTYPE field in GET_DIGITAL_SERVICE_LIST command
  - Changed the DATA_TYPE field in GET_DIGITAL_SERVICE_DATA command—now DATA_SRC and DSCTy
  - Changed some value ranges in DAB_DIGRAD_STATUS command
  - Added DAB_GET_FREQ_INFO command, added event indicators related to this information being updated
  - Added additional info to the DAB_GET_COMPONENT_INFO command
  - Added DAB_EVENT_MIN_SVRLIST_PERIOD_RECONFIG property
  - Changed DIGITAL_IO_OUTPUT_FORMAT property fields—added SLOT_SIZE

## Revision 1.0 to Revision 1.1

- Changed “Si46xx” to “Si468x” throughout.
- Updated Si468x Programming Guide and Firmware Revisions table for release 130927.
- Updated FMHD APIs.
  - Added HD_CODEC_MODE_0_SAMPLES_DELAY property. Note this was added to allow host software to override the default time alignment for the FMHD_RADIO 3.0.16 firmware.
  - Added HD_CODEC_MODE_2_SAMPLES_DELAY property.
  - Added HD_CODEC_MODE_10_SAMPLES_DELAY property.
  - Added HD_CODEC_MODE_13_SAMPLES_DELAY property.
  - Added HD_TEST_DEBUG_AUDIO property.

## Revision 1.1 to Revision 1.2

- Corrected value for ARG9 in POWER_UP command for all firmware modes.

## Revision 1.2 to Revision 1.3

- Corrected single-byte commands to include the necessary pad “0” byte, making them two-byte commands.

## Revision 1.3 to Revision 1.4

- Updated Table 1, “Si468x Programming Guide and Firmware Revisions[1],[2],” on page 1 for release 131209.

## Revision 1.4 to Revision 1.6

- Updated Table 1, “Si468x Programming Guide and Firmware Revisions[1],[2],” on page 1 for Release 131213.
- Updated Table 2, “Product Family Function,” on page 2 for new products supporting AM/AMHD.
- Added AMHD API.

## Revision 1.6 to Revision 1.7

- Updated Si46xx Programming Guide and Firmware Revisions table for release 140210.
- Updated DIGITAL_IO_OUTPUT_FORMAT property in FMHD API section.
  - Added SLOT_SIZE field.
  - Changed range on SAMPL_SIZE field.
- Corrected SUBCMD bytes for the following commands in “6.5.1.1. Flash Pass-Through Commands” :
  - FLASH_ERASE_SECTOR
  - FLASH_WRITE_BLOCK
  - FLASH_WRITE_BLOCK_READBACK_VERIFY
  - FLASH_WRITE_BLOCK_PACKET_VERIFY
  - FLASH_WRITE_BLOCK_READBACK_AND_PACKET _VERIFY

## Revision 1.7 to Revision 1.8

- Updated Si468x Programming Guide(AN649Rev1.7) and Firmware Revisions table for release 140610.
- Added a new property AM_VALID_HDLEVEL_THRESHOLD(0x4205), AM_CHBW_SQ_LIMITS(0x2200)
- Updated the default value of the property AM_RSQ_HD_DETECTION(0x4305), AM_VALID_RSSI_TIME(0x4201),

AM_VALID_RSSI_THRESHOLD(0x4202), AM_VALID_SNR_TIME(0x4203), AM_VALID_SNR_THRESHOLD(0x4204), AM_CHBW_SQ_LIMITS(0x2200), HD_CODEC_MODE_2_SAMPLES_DELAY (0x9904)

- Changed the property name of AM_CHBW_SQ_MIN_MAX(0x2200) to AM_CHBW_SQ_CHBW(0x2201)
- Deleted command HD_ACQUIRE and properties AM_CHBW_SQ_HIGH_THRESHOLD(0x2201), AM_CHBW_SQ_LOW_THRESHOLD(0x2202)
- Added “5.8. Updating the Boot Flash” (flowchart for updating the boot flash using the bootloader commands)

# CONTACT INFORMATION

**Silicon Laboratories Inc.**

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page:
https://www.siliconlabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.

**Patent Notice**

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.

Silicon Laboratories and Silicon Labs are trademarks of Silicon Laboratories Inc.
Other products or brandnames mentioned herein are trademarks or registered trademarks of their respective holders.