Qualcomm Technologies International, Ltd.

Qualcomm GAIA Commands

Reference

80-CF422-1 Rev. AB

November 13, 2017

Confidential and Proprietary – Qualcomm Technologies International, Ltd.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or websites to DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm Technologies International, Ltd. or its affiliated companies without the express approval of Qualcomm Configuration Management.

Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies International, Ltd.

Other Qualcomm products referenced herein are products of Qualcomm Technologies International, Ltd.

Qualcomm is a trademark of Qualcomm Incorporated, registered in the United States and other countries. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited.

Qualcomm Technologies International, Ltd. (formerly known as Cambridge Silicon Radio Limited) is a company registered in England and Wales with a registered office at: Churchill House, Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ, United Kingdom.
Registered Number: 3665875 | VAT number: GB787433096

Revision history

Revision	Date	Description
AA	06 November 2017	Original publication of this document. Alternative document number CS-00406554-DC. Branched from CS-00215201-DC for ADK 4.3 and ADK 6.1
AB	13 November 2017	Title changed to be generic. No change to technical content.

Contents

Revision history . . . 2

1 Wire protocol for GAIA . . . 10

1.1 GAIA command format . . . 10

1.2 GAIA framing . . . 11

1.3 GAIA command and acknowledgement . . . 11

2 GAIA command reference . . . 13

2.1 For users of API versions before v2.2 . . . 13

2.2 GAIA configuration commands . . . 13

2.2.1 GAIA set LED configuration (0x0101) . . . 14

2.2.2 GAIA get LED configuration (0x0181) . . . 16

2.2.3 GAIA set tone configuration (0x0102) . . . 17

2.2.4 GAIA get tone configuration (0x0182) . . . 17

2.2.5 GAIA set user-defined tone configuration (0x010e) . . . 18

2.2.6 GAIA get user-defined tone configuration (0x018e) . . . 19

2.2.7 GAIA set default volume (0x0103) . . . 20

2.2.8 GAIA get default volume (0x0183) . . . 20

2.2.9 GAIA set audio prompt configuration (0x0106) . . . 21

2.2.10 GAIA get audio prompt configuration (0x0186) . . . 21

2.2.11 GAIA set timer configuration (0x0109) . . . 22

2.2.12 GAIA get timer configuration (0x0189) . . . 22

2.2.13 GAIA set audio gain configuration (0x010a) . . . 23

2.2.14 GAIA get audio gain configuration (0x018a) . . . 24

2.2.15 GAIA set volume configuration (0x010b) . . . 24

2.2.16 GAIA get volume configuration (0x018b) . . . 25

2.3 GAIA control commands . . . 25

2.3.1 GAIA change volume (0x0201) . . . 27

2.3.2 GAIA set volume orientation (0x0205) . . . 27

2.3.3 GAIA get volume orientation (0x0285) . . . 28

2.3.4 GAIA set EQ control (0x0214) . . . 28

2.3.5 GAIA get EQ control 28
2.3.6 GAIA switch EQ control (0x0217) 29
2.3.7 GAIA set Bass Boost/Plus control (0x0215) 29
2.3.8 GAIA get Bass Boost/Plus control (0x0295) 29
2.3.9 GAIA toggle Bass Boost/Plus control (0x0218) 30
2.3.10 GAIA set 3D/3D-virtualization control (0x0216) 30
2.3.11 GAIA get 3D/3D-virtualisation Control (0x0296) 30
2.3.12 GAIA toggle 3D/3D-virtualisation control (0x0219) 31
2.3.13 GAIA set user EQ control (0x0220) 31
2.3.14 GAIA get user EQ control (0x02A0) 31
2.3.15 GAIA toggle user EQ control (0x0221) 32
2.3.16 GAIA set user EQ parameter (0x021a) 32
2.3.17 GAIA get user EQ parameter (0x029a) 33
2.3.18 GAIA set user EQ parameter group (0x021b) 33
2.3.19 GAIA get user EQ parameter group (0x029b) 34
2.3.20 GAIA AV remote control (0x021f) 35
2.3.21 GAIA set LED control (0x0207) 36
2.3.22 GAIA get LED control (0x0287) 36
2.3.23 GAIA device reset (0x0202) 36
2.3.24 GAIA set power state (0x0204) 37
2.3.25 GAIA get power state (0x0284) 37
2.3.26 GAIA set audio prompt control (0x020a) 37
2.3.27 GAIA get audio prompt control (0x028a) 38
2.3.28 GAIA change audio prompt language (0x020b) 38
2.3.29 GAIA set audio prompt language (0x0212) 39
2.3.30 GAIA get audio prompt language (0x0292) 39
2.3.31 GAIA set speech recognition control (0x020c) 39
2.3.32 GAIA get speech recognition control (0x028c) 40
2.3.33 GAIA play tone (0x0209) 40
2.3.34 GAIA alert LEDs (0x020d) 41
2.3.35 GAIA alert tone (0x020e) 41
2.3.36 GAIA alert event (0x0210) 42
2.3.37 GAIA alert audio prompt (0x0211) 42
2.3.38 GAIA start speech recognition (0x0213) 43
2.3.39 GAIA enter Bluetooth Pairing mode (0x021d) 43
2.3.40 GAIA set TWS audio routing (0x0224) 44
2.3.41 GAIA get TWS audio routing (0x02a4) 44
2.3.42 GAIA trim TWS volume (0x0226) 45

2.3.43 GAIA set peer link reserved (0x0227) 45
2.3.44 GAIA get peer link reserved (0x02a7) 46
2.3.45 GAIA find my remote (0x022b) 46
2.3.46 GAIA Set Data Endpoint mode (0x022e) 46
2.3.47 GAIA Get Data Endpoint mode (0x02ae) 47
2.4 GAIA polled status commands 47
2.4.1 GAIA get API version (0x0300) 47
2.4.2 GAIA get current RSSI (0x0301) 48
2.4.3 GAIA get current battery level (0x0302) 48
2.4.4 GAIA get module ID (0x0303) 48
2.4.5 GAIA get application version (0x0304) 49
2.5 GAIA partition and file commands 49
2.5.1 GAIA get storage partition status (0x0610) 50
2.5.2 GAIA open storage partition (0x0611) 50
2.5.3 GAIA write storage partition (0x0615) 51
2.5.4 GAIA close storage partition (0x0618) 52
2.5.5 GAIA mount storage partition (0x061a) 52
2.5.6 GAIA open file (0x0621) 53
2.5.7 GAIA read file (0x0624) 53
2.5.8 GAIA close file (0x0628) 54
2.5.9 GAIA get mounted partitions (0x01a0) 54
2.6 GAIA no operation (0x0700) debugging command 55
2.7 GAIA notification commands 55
2.7.1 GAIA register notification (0x4001) 55
2.7.2 GAIA get notification (0x4081) 56
2.7.3 GAIA cancel notification (0x4002) 56
2.7.4 GAIA event notification (0x4003) 57
2.8 GAIA events 57
2.8.1 GAIA RSSI low threshold (Event code 0x01) 57
2.8.2 GAIA RSSI high threshold (Event code 0x02) 58
2.8.3 GAIA battery low threshold (Event code 0x03) 58
2.8.4 GAIA battery high threshold (Event code 0x04) 59
2.8.5 GAIA PIO changed (Event code 0x06) 59
2.8.6 GAIA battery charged (Event code 0x08) 60
2.8.7 GAIA charger connection (Event code 0x09) 60
2.8.8 GAIA user action (Event code 0x0b) 61
2.8.9 GAIA speech recognition (Event code 0x0c) 61

3 Example GAIA exchanges . . . 63
3.1 GAIA no operation example . . . 63
3.2 GAIA get current RSSI example . . . 64
3.3 GAIA alert tone example . . . 65
3.4 GAIA register notification (battery low threshold) example . . . 66
A GAIA commands defined in API v2.5 . . . 69
B GAIA notification event codes . . . 72
C GAIA command status codes . . . 73
D QTIL application tone parameters . . . 74
E GAIA timer numbers . . . 76
F User EQ parameters . . . 78
F.1 Parameter ID format . . . 78
F.2 Filter type parameter . . . 79
F.3 Filter frequency parameter . . . 79
F.4 Gain parameter . . . 80
F.5 Q parameter . . . 80
F.6 Parameter ranges . . . 80
G Volume configuration parameters . . . 81
H GAIA PS Keys . . . 82
Document references . . . 83
Terms and definitions . . . 84

Tables

Table 2-1: Acknowledgement payload 22

Table 3-1: GAIA no operation host example request 63

Table 3-2: Description of no operation device example response 63

Table 3-3: Description of the get current RSSI example request 64

Table 3-4: Description of the get current RSSI example response 64

Table 3-5: Description of the alert tone example request 65

Table 3-6: Description of the alert tone example response 65

Table 3-7: Description of the register notification example request 66

Table 3-8: Description of the register notification example response 66

Table 3-9: Description of the get notification example request 67

Table 3-10: Description of the get notification example response 67

Table 3-11: Description of the event notification example 67

Table 3-12: Description of the event notification acknowledgement example 68

Table A-1: 0x01xx configuration commands 69

Table A-2: 0x02xx control commands 69

Table A-3: 0x03xx polled status commands 71

Table A-4: 0x06xx data transfer commands 71

Table A-5: 0x0700 ‘No Operation’ debugging command 71

Table A-6: 0x40xx notification commands 71

Table B-1: Notification event codes 72

Table C-1: Command status codes 73

Table D-1: Note timbre values 74

Table D-2: Note pitch values 75

Table E-1: Sink Application Timer Numbers 76

Table F-1: Parameter ID format........ 78

Table F-2: User EQ parameter IDs........ 78

Table F-3: User EQ parameterID examples........ 79

Table F-4: User EQ filter types........ 79

Table F-5: User EQ parameter ranges........ 80

Table G-1: Volume configuration parameter numbers........ 81

Figures

Figure 1-1: GAIA command format.. 10

Figure 1-2: QTIL GAIA packet framing.. 11

1 Wire protocol for GAIA

Qualcomm® GAIA (Generic Application Interface Architecture) implements an end-to-end, host-agnostic ecosystem supporting host application access to device functionality.

The underlying protocol data unit is a packet composed of octets with framing to permit transmission over stream-based connections such as SPP and RS-232. Numeric fields longer than 8 bits are packed with the most significant octet first. Textual strings are encoded using UTF-8.

In this document:

- The Host is the controlling party, for example, an application running on a smartphone
- The Device is the controlled party, for example, a Bluetooth Headset

Commands may be sent from the Host to the Device or from the Device to the Host.

Where a connection needs to be established at a lower protocol level, for example, for Bluetooth SPP, the QTIL GAIA protocol does not dictate which party is the initiator, that is, the Host may be initiated as the Client or the Server.

1.1 GAIA command format

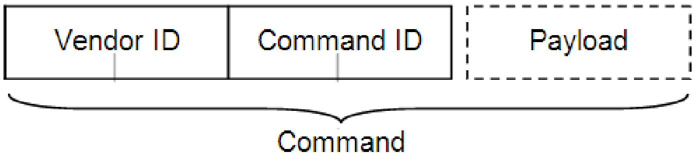

Figure 1-1 GAIA command format

Description of fields:

- Vendor ID: This 16-bit field qualifies the command ID. All commands in this document have the Vendor ID assigned to QTIL by the Bluetooth SIG, that is. `0x000a`.
- Command ID: This 16-bit field identifies the individual command.
- Payload: The payload contains any information required to be passed by a specific command. It consists of zero or more octets depending on the command.
- Check: One octet. If present this field is determined by XORing together the other octets in the packet.

1.2 GAIA framing

For stream-based transports, e.g. connections with Classic Bluetooth devices using SPP..

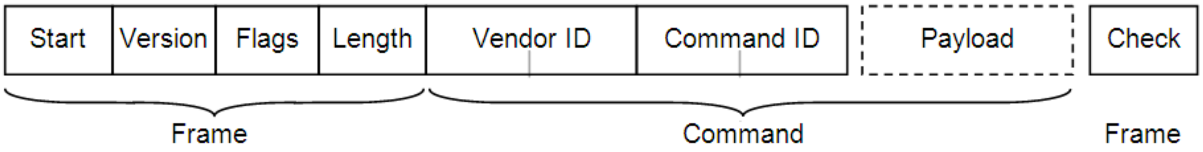

Figure 1-2 QTIL GAIA packet framing

Description of Packet Fields.

- Start: One octet with the fixed value `0xff`.
- Version: One octet. This field indicates the protocol version in use, currently 1.
- Flags: One octet. Bits within this field control protocol options.
  - Bit[0]: If set, a single octet check is present
  - Bit[1:7]: Reserved, must be 0
- Vendor ID: This 16-bit field qualifies the command ID. All commands in this document have the Vendor ID assigned to QTIL by the Bluetooth SIG, that is, `0x000a`.
- Command ID: This 16-bit field identifies the individual command.
- Payload: The payload contains any information required to be passed by a specific command. It consists of zero or more octets depending on the command.
- Check: One octet. If present this field is determined by XORing together the other octets in the packet.

The start, length and check fields together enable the receiver to validate each packet. An acknowledgement (ACK) packet is sent in response to each valid command received. Packets with invalid check fields are silently ignored.

1.3 GAIA command and acknowledgement

Every command is acknowledged. An acknowledgement packet has the same structure as the command packet with the value in the Command ID field being that of the initiating command with the top bit set. For example, the command `0x0001` would be acknowledged with a packet containing `0x8001` in the Command ID field.

The first octet of the payload of an acknowledgement holds a status value, see GAIA command status codes.

With framed (stream-based) transports, the originator may send multiple commands without waiting for each to be acknowledged, subject to implementation and resource constraints. To send a large amount of data you should wait for acknowledgements to avoid exhausting the device memory. Multiple QTIL GAIA packets can be send in a single packet at a lower level (for example, L2CAP, USB), reducing protocol overhead. The QTIL GAIA protocol does not specify that acknowledgements are received in order.

QTIL GAIA commands are organized into categories:

Configuration Commands

These modify the configuration of the Device. It is necessary to reset the Device before these changes take effect. The first octet of each configuration command ID is `0x01`.

NOTE These commands do not automatically cause a reset so that several can be applied before a reset is carried out.

Control Commands

Control command requests are carried out immediately, with no need for a Device reset. The first octet of each control command ID is `0x02`.

Polled Status Commands

These request information about the status of the Device. The first octet of each polled status command ID is `0x03`.

Partition and File Commands

These commands allow read access to files within the file system. On devices that support external file systems, partitions can be updated and mounted. The first octet of each Partition and File command ID is `0x06`.

Debugging Commands

These enable an application developer to verify the operation of the protocol. The first octet of each debugging command ID is `0x07`.

Notification Commands

The QTIL GAIA protocol supports notifications, that is,. asynchronous indications of a change in Device state. This eliminates the need for the Host to continually poll the Device. This reduces radio traffic and increases battery life. The first octet of each notification command ID is `0x40`. See GAIA notification commands for more details.

2 GAIA command reference

2.1 For users of API versions before v2.2

The payload of some commands changed in API version 2.2.

Device events are now 16-bit numbers. The event identifier has been increased from 1 to 2 octets.

The affected commands are

- GAIA set LED configuration (0x0101)
- GAIA get LED configuration (0x0181)
- GAIA set tone configuration (0x0102)
- GAIA get tone configuration (0x0182)
- GAIA alert event (0x0210)

The Long Form of the Set timer configuration and Get timer configuration commands is deprecated in API version 2.5. This document describes only the Short Form of these commands.

2.2 GAIA configuration commands

The following commands can be used to configure an application using GAIA:

- GAIA set LED configuration (0x0101)
- GAIA get LED configuration (0x0181)
- GAIA set tone configuration (0x0102)
- GAIA get tone configuration (0x0182)
- GAIA set user-defined tone configuration (0x010e)
- GAIA get user-defined tone configuration (0x018e)
- GAIA set default volume (0x0103)
- GAIA get default volume (0x0183)
- GAIA set audio prompt configuration (0x0106)
- GAIA get audio prompt configuration (0x0186)
- GAIA set timer configuration (0x0109)
- GAIA get timer configuration (0x0189)
- GAIA set audio gain configuration (0x010a)

- GAIA get audio gain configuration (0x018a)
- GAIA set volume configuration (0x010b)
- GAIA get volume configuration (0x018b)

2.2.1 GAIA set LED configuration (0x0101)

QTIL GAIA devices allow each state/event to be associated with a unique LED flashing pattern. In addition, the software allows filters to be applied to individual LED patterns to change them in response to some event or change in state. For example, it is possible to change the color and/or frequency of a pattern being displayed when the battery is low.

When implementing a filter it is normally necessary to create two filters, one to enable the change when the trigger event or state change occurs and one to disable the change when trigger condition ends.

LED state configuration

Each Device state can be associated with a unique LED flashing pattern.

Command payload

Offset	Value	Comment
0	0x01	LED Configuration type: State
1	…	State ID, high byte
2	. . .	State ID, low byte
3	. . .	LED On time, high byte
4	. . .	LED On time, low byte
5	. . .	LED Off time, high byte
6	. . .	LED Off time, low byte
7	. . .	LED Repeat time, high byte
8	. . .	LED Repeat time, low byte
9	. . .	LED Dim time, high byte
10	. . .	LED Dim time, low byte
11	. . .	Number of flashes
12	. . .	Timeout
13	. . .	LED ‘A’ selector
14	. . .	LED ‘B’ selector
15	. . .	Override Disable
16	. . .	Color

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code The payload is not a valid LED State configuration

LED event configuration

Each system event can be associated with a unique LED flashing pattern. See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for more information.

Command payload

Offset	Value	Comment
0	`0x02`	LED Configuration type: Event
1	...	Event ID, high byte
2	...	Event ID, low byte
3	...	LED On time, high byte
4	...	LED On time, low byte
5	...	LED Off time, high byte
6	...	LED Off time, low byte
7	...	LED Repeat time, high byte
8	...	LED Repeat time, low byte
9	...	LED Dim time, high byte
10	...	LED Dim time, low byte
11	...	Number of flashes
12	...	Timeout
13	...	LED ‘A’ selector
14	...	LED ‘B’ selector
15	...	Override Disable
16	...	Color

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code The payload is not a valid LED Event configuration

LED filter configuration

The command payload defines a filter to modify an existing pattern. The contents depend on the filter action. See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for more information on LED filters.

Command payload

Offset	Value	Comment
0	0x03	LED Configuration type: Filter
1	...	Filter Index
2	...	Event ID, high byte
3	...	Event ID, low byte
4	...	Filter Action
5	...	Override Disable
6	...	Filter Parameter 1
7	...	Filter Parameter 2 if used

The payload includes one or two parameters depending on the filter action.

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success’ status code
	0x05	'Invalid Parameter' status code The event number is not recognized or the parameters are incorrect

2.2.2 GAIA get LED configuration (0x0181)

This command gets the configuration of a LED pattern for a given state, event or filter. The command payload holds three octets giving the configuration type and the configuration index.

Command payload

Offset	Value	Comment
0	0x01	LED State configuration
	0x02	LED Event configuration
	0x03	LED Filter configuration
1	...	Event ID or Filter index, high byte
2	...	Event ID or Filter index, low byte

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The configuration type or index is not recognized
1	0x01	LED State configuration
	0x02	LED Event configuration
	0x03	LED Filter configuration
2	...	Event or Filter ID, high byte
3	...	Event or Filter ID, low byte
4	...	Further octets holding the filter configuration

2.2.3 GAIA set tone configuration (0x0102)

Each event can be associated with a tone to be played when the event occurs. The command payload has three octets, holding the event number and the tone number.

Command payload

Offset	Value	Comment
0	...	Event number, high byte
1	...	Event number, low byte
2	...	Tone number associated with this event

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The event number is not recognized or the tone number is not recognized
1	...	Event number, high byte
2	...	Event number, low byte
3	...	Tone number associated with this event

2.2.4 GAIA get tone configuration (0x0182)

This command gets the tone associated with a given system event. The command payload has two octets holding the event number. The acknowledgement payload holds four octets giving the command status, the event number and the tone number.

Command payload

Offset	Value	Comment
0	. . .	Event number, high byte
1	. . .	Event number, low byte

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The event number is not recognized
1	. . .	Event number, high byte
2	. . .	Event number, low byte
3	. . .	Tone number associated with this event

2.2.5 GAIA set user-defined tone configuration (0x010e)

This command configures a user-defined tone. When configured the tone may be associated with a system event or played in the same way as a built-in tone. Up to eight user-defined tones may be configured.

The command payload holds the tone number and note data for the tone.

NOTE Data consists of tempo, volume, timbre and decay values for the whole sequence followed by pitch and duration values for each note. For details see QTIL application tone parameters.

Command payload

Offset	Value	Comment
0	. . .	Tone number, `0x01` to `0x08`
1	. . .	Tempo of the tone sequence
2	. . .	Volume
3	. . .	Timbre
4	. . .	Decay
5	. . .	Pitch of the first note in the sequence
6	. . .	Duration of the first note in the sequence
7	. . .	Pitch of the second note in the sequence
8	. . .	Duration of the second note in the sequence
9 ...	. . .	Further notes in the sequence

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.2.6 GAIA get user-defined tone configuration (0x018e)

This command gets the configuration of a user-defined tone. The command payload holds the tone number. The acknowledgement payload holds the command status, the tone number and the note data for the tone.

NOTE Data consists of tempo, volume, timbre and decay values for the whole sequence followed by pitch and duration values for each note. For details see QTIL application tone parameters.

Command payload

Offset	Value	Comment
0	...	Tone number, `0x01` to `0x08`
1	...	Tempo of the tone sequence
2	...	Volume
3	...	Timbre
4	...	Decay
5	...	Pitch of the first note in the sequence
6	...	Duration of the first note in the sequence
7	...	Pitch of the second note in the sequence
8	...	Duration of the second note in the sequence
10 ...	...	Further notes in the sequence

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The tone number is not recognized

2.2.7 GAIA set default volume (0x0103)

This command sets the initial volume settings for the Device. The command payload consists of three octets with values representing the gain settings for indication tones, voice calls and streamed music respectively. The tones volume setting is a number in the range `0x00` to `0x16` (0 to 22 decimal). Each number is mapped to a fixed gain. Voice and Music values are numbers in the range `0x00` to `0x0f` (0 to 15 decimal). The mapping of each number to a gain setting is determined by the Audio Gains configuration, see GAIA set audio gain configuration (0x010a). See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for more information.

Command payload

Offset	Value	Comment
0	`...`	Default volume for tones, `0x00` to `0x16`
1	`...`	Default volume for voice, `0x00` to `0x0f`
2	`...`	Default volume for music, `0x00` to `0x0f`

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.2.8 GAIA get default volume (0x0183)

This command gets the initial volume settings for the Device. The acknowledgement payload holds the command status and values representing the gain settings for indication tones, voice calls and streamed music respectively. The tones volume setting is a number in the range `0x00` to `0x16` (0 to 22 decimal). Each number is mapped to a fixed gain. Voice and Music values are numbers in the range `0x00` to `0x0f` (0 to 15 decimal). The mapping of each number to a gain setting is determined by the Audio Gains configuration. See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for more information.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`...`	Default volume for tones, `0x00` to `0x16`
2	`...`	Default volume for voice, `0x00` to `0x0f`
3	`...`	Default volume for music, `0x00` to `0x0f`

2.2.9 GAIA set audio prompt configuration (0x0106)

This command configures an audio prompt to be played when a given event occurs and the Device is in one of a given set of states.

Command payload

Offset	Value	Comment
0	...	Event number, high byte
1	...	Event number, low byte
2	...	Audio prompt ID
3	0x00	Queue the prompt to be played next
	0x01	Cancel all queued prompts and play this one immediately
4	0x00	Play audio prompt even if SCO or Wired audio are routed
	0x01	Do not play prompt if SCO or Wired audio are routed
5	...	State mask, high byte
6	...	State mask, low byte

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success’ status code
	0x05	'Invalid Parameter' status code One or more values in the command payload is invalid

2.2.10 GAIA get audio prompt configuration (0x0186)

This command gets the audio prompt to be played when a given event occurs.

Command payload

Offset	Value	Comment
0	...	Event number, high byte
1	...	Event number, low byte

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success’ status code
	0x05	'Invalid Parameter' status code One or more values in the command payload is invalid
1	...	Event number, high byte
2	...	Event number, low byte
3	...	Audio prompt ID
4	0x00	Queue the prompt to be played next
	0x01	Cancel all queued prompts and play this one immediately

Offset	Value	Comment
5	`0x00`	Play audio prompt even if SCO or Wired audio are routed
	`0x01`	Do not play prompt if SCO or Wired audio are routed
6	...	State mask, high byte
7	...	State mask, low byte

2.2.11 GAIA set timer configuration (0x0109)

This command configures timers and counters on the Device. The command payload holds one octect with the timer number and two octets with the 16-bit value for that timer.

Command payload

Offset	Value	Comment
0	….	Timer Number (See GAIA timer numbers)
1	. . .	Timer Value (high byte)
2	. . .	Timer Value (low byte)

Table 2-1 Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.2.12 GAIA get timer configuration (0x0189)

This command gets the timer and counter configuration from the Device. The acknowledgement payload holds the command status followed by 52 further octets carrying the timer values.

Command Payload

Offset	Value	Comment
0	….	Timer Number (See GAIA timer numbers)

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code. The timer number is invalid

Offset	Value	Comment
1	. . .	Timer Number (See
2	. . .	Timer Value (high byte)

2.2.13 GAIA set audio gain configuration (0x010a)

This command configures the behavior of the Device voice volume controls. The volume control may be set to one of sixteen levels between `0x00` to `0x0f` (0 to 15 decimal). Each level has a reserved field which must be zero.

The configuration of each level determines the following:

- The level to change to when Volume Up is pressed
- The level to change to when Volume Down is pressed
- A tone to play when the level is selected
- The speaker (voice) gain to use at this level

The command payload has 80 octets holding these values. You must supply values for all the fields. To change values, first use the Get Audio Gain configuration command described in GAIA set LED configuration (0x0101) to retrieve the current values.

Command payload

Offset	Value	Comment	Volume Level
0	...	Volume Up level	0
1	...	Volume Down level	
2	...	Indication tone number	
3	0x00	Reserved, must be zero	
4	...	Speaker gain	
“	“	“	“
75	...	Volume Up level	15
76	...	Volume Down level	
77	...	Indication tone number	
78	0x00	Reserved, must be zero	
79	...	Speaker gain	

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.2.14 GAIA get audio gain configuration (0x018a)

This command gets the configuration of the Device voice volume controls. The volume control may be at one of sixteen levels from 0 to 15. The configuration of each level determines

- The level to change to when Volume Up is pressed
- The level to change to when Volume Down is pressed
- A tone to play when the level is selected
- The speaker (voice) gain to use at this level

The acknowledgement payload holds the command status plus 80 octets with these values.

Command Payload

None

Acknowledgement payload

Offset	Value	Comment	Volume Level
0	`0x00`	'Success’ status code	0
1	. . .	Volume Up level for Volume Level 0	
2	. . .	Volume Down level for Volume Level 0	
3	. . .	Indication tone number for Volume Level 0	
4	`0x00`	Reserved	
5	. . .	Speaker gain for Volume Level 0	
...“	. . .‵‵	...“	“
76	. . .	Volume Up level for Volume Level 15	15
77	. . .	Volume Down level for Volume Level 15	
78	. . .	Indication tone number for Volume Level 15	
79	`0x00`	Reserved	
80	. . .	Speaker gain for Volume Level 15	

2.2.15 GAIA set volume configuration (0x010b)

This command sets the configuration of the volume curves for both the main (Primary, Secondary and Sub-woofer) and auxiliary outputs. It also sets the configuration of the TWS device volume.

Decibel values are 16-bit signed numbers scaled by 60. For example a value of -12 dB is represented by 0xfd30.

Command payload

Offset	Value	Comment
0	`0x11`	Parameter number (See Parameter ID format). Example: TWS device trim minimum gain
1	`0xfd`	Parameter value, high byte. Example: -12 dB
2	`0x30`	Parameter value, low byte.

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code

2.2.16 GAIA get volume configuration (0x018b)

This command gets the configuration of the volume curves for both the main (Primary, Secondary and Sub-woofer) and auxiliary outputs. It also gets the configuration of the TWS device volume.

Command payload

Offset	Value	Comment
0	. . .	Parameter number (See Parameter ID format)

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code
1	. . .	Parameter number (See Parameter ID format)
2	. . .	Parameter value, high byte
3	. . .	Parameter value, low byte

2.3 GAIA control commands

The following commands can be used to control an application using GAIA:

- GAIA change volume (0x0201)
- GAIA set volume orientation (0x0205)
- GAIA get volume orientation (0x0285)
- GAIA set EQ control (0x0214)
- GAIA get EQ control
- GAIA switch EQ control (0x0217)

- GAIA set Bass Boost/Plus control (0x0215)
- GAIA get Bass Boost/Plus control (0x0295)
- GAIA toggle Bass Boost/Plus control (0x0218)
- GAIA set 3D/3D-virtualization control (0x0216)
- GAIA get 3D/3D-virtualisation Control (0x0296)
- GAIA toggle 3D/3D-virtualisation control (0x0219)
- GAIA set user EQ control (0x0220)
- GAIA get user EQ control (0x02A0)
- GAIA toggle user EQ control (0x0221)
- GAIA set user EQ parameter (0x021a)
- GAIA get user EQ parameter (0x029a)
- GAIA set user EQ parameter group (0x021b)
- GAIA get user EQ parameter group (0x029b)
- GAIA AV remote control (0x021f)
- GAIA set LED control (0x0207)
- GAIA get LED control (0x0287)
- GAIA device reset (0x0202)
- GAIA set power state (0x0204)
- GAIA get power state (0x0284)
- GAIA set audio prompt control (0x020a)
- GAIA get audio prompt control (0x028a)
- GAIA change audio prompt language (0x020b)
- GAIA set audio prompt language (0x0212)
- GAIA get audio prompt language (0x0292)
- GAIA set speech recognition control (0x020c)
- GAIA get speech recognition control (0x028c)
- GAIA play tone (0x0209)
- GAIA alert LEDs (0x020d)
- GAIA alert tone (0x020e)
- GAIA alert event (0x0210)
- GAIA alert audio prompt (0x0211)
- GAIA start speech recognition (0x0213)
- GAIA enter Bluetooth Pairing mode (0x021d)
- GAIA set TWS audio routing (0x0224)
- GAIA get TWS audio routing (0x02a4)
- GAIA trim TWS volume (0x0226)

- GAIA set peer link reserved (0x0227)
- GAIA get peer link reserved (0x02a7)
- GAIA find my remote (0x022b)
- GAIA Set Data Endpoint mode (0x022e)
- GAIA Get Data Endpoint mode (0x02ae)

2.3.1 GAIA change volume (0x0201)

This command increases or decreases the volume. The command payload is a single octet with a value of `0x00` (volume up) or `0x01` (volume down).

Command payload

Offset	Value	Comment
0	`0x00`	Increase Volume
	`0x01`	Decrease Volume

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The control octet is other than `0` or `1`

2.3.2 GAIA set volume orientation (0x0205)

This command sets the orientation of the Device volume control buttons. The command payload is a single octet with a value of `0x00` (Normal orientation) or `0x01` (Inverted orientation).

Command payload

Offset	Value	Comment
0	`0x00`	Set Normal Volume Orientation
	`0x01`	Set Inverted Volume Orientation

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The control octet is other than `0` or `1`

2.3.3 GAIA get volume orientation (0x0285)

This command gets the orientation of the Device volume control buttons. The acknowledgement payload carries two octets holding the command status `0x00` (indicating Success) and current orientation, `0x00` (Normal) or `0x01` (Inverted).

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x00`	Volume Orientation is Normal
	`0x01`	Volume Orientation is Inverted

2.3.4 GAIA set EQ control (0x0214)

This command selects an audio equalizer bank. The payload carries one octet holding the equalizer bank number, `0x00` to `0x06`.

Command payload

Offset	Value	Comment
0	`...`	Equalizer bank number

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The selected bank is not available.

2.3.5 GAIA get EQ control

This command returns the currently selected equalizer bank.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`...`	Equalizer bank number

2.3.6 GAIA switch EQ control (0x0217)

This command changes the audio equalizer bank. The next available bank is selected in rotation. If the last available bank is selected, the bank selection wraps around to the first.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.3.7 GAIA set Bass Boost/Plus control (0x0215)

This command turns the Bass Boost/Plus function on or off. The command payload is a single octet with a value of `0x00` (Bass Boost/Plus Off) or `0x01` (Bass Boost/Plus On).

Command payload

Offset	Value	Comment
0	`0x00`	Turn off Bass Boost/Plus
	`0x01`	Turn on Bass Boost/Plus

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The control octet is other than `0` or `1`

NOTE The function selection of the MeloD Bass Boost/Bass+/Neither, which are static for the product will be configured via the UFE. Several tuning parameters required are included in the UFE parameters list which is stored in DSP PS Keys. If enabled, the VM GAIA is used to turn the feature on or off.

2.3.8 GAIA get Bass Boost/Plus control (0x0295)

This command gets the state of the Bass Boost/Plus function. The acknowledgement payload carries two octets holding the command status `0x00` (indicating success) and current state, `0x00` (Bass Boost/Plus Off) or `0x01` (Bass Boost/Plus On).

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x00`	Bass Boost/Plus is Off
	`0x01`	Bass Boost/Plus is On

2.3.9 GAIA toggle Bass Boost/Plus control (0x0218)

This command toggles the Bass Boost/Plus function on or off. If Bass Boost/Plus is on, it is turned off. If Bass Boost/Plus is off, it is turned on.

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.3.10 GAIA set 3D/3D-virtualization control (0x0216)

This command turns the 3D/3D-virtualisation function on or off. The command payload is a single octet with a value of `0x00` (3D/3D-virtualisation Off) or `0x01` (3D/3D-virtualisation On).

Command Payload

Offset	Value	Comment
0	`0x00`	Turn off 3D/3D-virtualisation
	`0x01`	Turn on 3D/3D-virtualisation

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The control octet is other than `0` or `1`

2.3.11 GAIA get 3D/3D-virtualisation Control (0x0296)

This command gets the state of the 3D/3D-virtualisation function. The acknowledgement payload carries two octets holding the command status `0x00` (indicating Success) and current state, `0x00` (3D/3D-virtualisation Off) or `0x01` (3D /3D-virtualisation On).

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x00`	3D/3D-virtualisation is Off
	`0x01`	3D/3D-virtualisation is On

2.3.12 GAIA toggle 3D/3D-virtualisation control (0x0219)

This command toggles the 3D/3D-virtualisation function on or off. If 3D/3D-virtualisation is on, it is turned off. If 3D/3D-virtualisation is off, it is turned on.

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.3.13 GAIA set user EQ control (0x0220)

This command turns the User EQ function on or off. The command payload is a single octet with a value of `0x00` (User EQ Off) or `0x01` (User EQ On).

Command Payload

Offset	Value	Comment
0	`0x00`	Turn off User EQ
	`0x01`	Turn on User EQ

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The control octet is other than `0` or `1`

2.3.14 GAIA get user EQ control (0x02A0)

This command gets the state of the User EQ function. The acknowledgement payload carries two octets holding the command status `0x00` (indicating Success) and current state, `0x00` (User EQ Off) or `0x01` (User EQ On).

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x00`	User EQ is Off
	`0x01`	User EQ is On

2.3.15 GAIA toggle user EQ control (0x0221)

This command toggles the User EQ function on or off. If User EQ is on, it is turned off. If User EQ is turned off, it is turned on.

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.3.16 GAIA set user EQ parameter (0x021a)

This command sets an EQ parameter. If byte 4 is non zero, then the filter coefficients are recalculated. Setting the coefficients to be recalculated should only be done if the filter being updated is part of the currently selected EQ bank. See User EQ parametersfor details of the User EQ parameters.

Command Payload

Offset	Value	Comment
0	. . .	Parameter ID, high byte
1	. . .	Parameter ID, low byte
2	. . .	Value, high byte
3	. . .	Value, low byte
4	`0x00`	Do not recalculate coefficients of filter
	`0x01`	Recalculate coefficients of filter

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.3.17 GAIA get user EQ parameter (0x029a)

This command gets an EQ parameter.

Command payload

Offset	Value	Comment
0	. . .	Parameter ID, high byte
1	. . .	Parameter ID, low byte

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The Parameter ID is invalid
1	. . .	Parameter ID, high byte
2	. . .	Parameter ID, low byte
3	. . .	Value, high byte
4	. . .	Value, low byte

2.3.18 GAIA set user EQ parameter group (0x021b)

This command sets a group of User EQ parameters. There is no option to recalculate the coefficients when setting a group of parameters. See Parameter ID format for details of the User EQ parameters.

Command Payload

Offset	Value	Comment
0	. . .	Number of parameters, high byte
1	. . .	Number of parameters, low byte
2	. . .	Parameter 1 ID, high byte
3	. . .	Parameter 1 ID, low byte
4	. . .	Value 1, high byte
5	. . .	Value 1, low byte
"	"	"

Offset	Value	Comment
"	"	Parameter ‘n’ ID, high byte
"	"	Parameter ‘n’ ID, low byte
"	"	Value ‘n’, high byte
"	"	Value ‘n’, low byte
"	`0x00`	Reserved, must be zero
"	`0x00`	Reserved, must be zero

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid

2.3.19 GAIA get user EQ parameter group (0x029b)

This command gets a group of User EQ parameters. See Parameter ID format for details of the User EQ parameters.

Command Payload

Offset	Value	Comment
0	`...`	Number of parameters, high byte
1	`...`	Number of parameters, low byte
2	`...`	Parameter 1 ID, high byte
3	`...`	Parameter 1 ID, low byte
4	`0x00`	Must be zero
5	`0x00`	Must be zero
"	"	"
"	"	Parameter ‘n’ ID, high byte
"	"	Parameter ‘n’ ID, low byte
"	`0x00`	Must be zero
"	`0x00`	Must be zero

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code One or more values in the command payload is invalid.
1	`...`	Number of parameters, high byte

Offset	Value	Comment
2	. . .	Number of parameters, low byte
3	. . .	Parameter 1 ID, high byte
4	. . .	Parameter 1 ID, low byte
5	. . .	Value 1, high byte
6	. . .	Value 1, low byte
"	"	"
"	"	Parameter ‘n’ ID, high byte
"	"	Parameter ‘n’ ID, low byte
"	"	Value 'n', high byte
"	"	Value 'n', low byte

2.3.20 GAIA AV remote control (0x021f)

This command controls AV functions of the device. The command payload is a single octet with the function to perform.

NOTE If the Device power is on, the Power command powers it off. If the power is off and the Device is configured to remain connected, the Power command powers it on.

Command payload

Offset	Value	Comment
0	`0x40`	Power
	`0x41`	Volume Up
	`0x42`	Volume Down
	`0x43`	Mute
	`0x44`	Play
	`0x45`	Stop
	`0x46`	Pause
	`0x4b`	Skip Forward
	`0x4c`	Skip Backward

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code The control octet is not a valid AV command code

2.3.21 GAIA set LED control (0x0207)

This command enables LED indicators on the Device. The command payload is a single octet with a value of `0x00` (disable) or `0x01` (enable).

Command payload

Offset	Value	Comment
0	`0x00`	Disable LEDs
	`0x01`	Enable LEDs

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code The control octet is other than 0 or 1

2.3.22 GAIA get LED control (0x0287)

This command gets the current setting of the LED indicator control. The acknowledgement payload holds one octet with the command status and one octet with the control setting.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
1	`0x00`	LEDs are disabled
	`0x01`	LEDs are enabled

2.3.23 GAIA device reset (0x0202)

This command causes the Device to warm reset and load any changes made by the configuration commands. The Device sends the acknowledgement then resets after a delay of approximately one second.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code

2.3.24 GAIA set power state (0x0204)

This command causes the Device to power on or off. If powering off, the Device sends the acknowledgement then powers off after a delay of approximately one second. The Device must be configured to remain connected for powering on to work.

For backward compatibility, the command with no payload powers off the Device.

Command payload

Offset	Value	Comment
0	0x00	Power Off
	0x01	Power On

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code

2.3.25 GAIA get power state (0x0284)

This command causes gets the current power state of the Device.

Command payload

None

Acknowledgement Payload

Offset	Value	Comment
0	0x00	'Success' status code
1	0x00	The Device is powered Off.
	0x01	The Device is powered On.

2.3.26 GAIA set audio prompt control (0x020a)

This command enables audio prompts on the Device. The command payload is a single octet with a value of 0 (disable) or 1 (enable).

Command Payload

Offset	Value	Comment
0	0x00	Disable Audio Prompts
	0x01	Enable Audio Prompts

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The control octet is other than 0 or 1

2.3.27 GAIA get audio prompt control (0x028a)

This command gets the current setting of the Audio Prompt feature. The acknowledgement payload holds one octet with the command status and one octet with the control setting.

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	0x00	'Success' status code
1	0x00	Audio Prompts are disabled
	0x01	Audio Prompts are enabled

2.3.28 GAIA change audio prompt language (0x020b)

This command changes the language used for audio prompts. The next available language is selected in rotation. For example, if the Device supports English, French, Italian, German and, Spanish, and if German is currently in use, this command changes the language to Spanish. If Spanish is in use, the language selection wraps around to English.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code

2.3.29 GAIA set audio prompt language (0x0212)

This command selects a language to use for audio prompts. The command payload is a single octet holding a number representing the language to be used. The languages available and the numbers which represent them depend on the implementation.

Command payload

Offset	Value	Comment
0	...	A number representing the language

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code No language is associated with the command payload number

2.3.30 GAIA get audio prompt language (0x0292)

This command gets the language in use for audio prompts. The acknowledgement payload contains one octet holding the command status and one octet holding a number representing the language. The languages available and the numbers which represent them depend on the implementation.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
1	...	A number representing the currently-selected language

2.3.31 GAIA set speech recognition control (0x020c)

This command enables the speech recognition feature on the Device. The command payload is a single octet with a value of 0 (disable) or 1 (enable). This controls the use of speech recognition by the Device to answer or reject incoming calls. It does not disable the use of the QTIL GAIA Start Speech Recognition command.

Command payload

Offset	Value	Comment
0	`0x00`	Disable Speech Recognition
	`0x01`	Enable Speech Recognition

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The control octet is other than 0 or 1

2.3.32 GAIA get speech recognition control (0x028c)

This command gets the current setting of the Speech Recognition feature. The acknowledgement payload holds one octet with the command status and one octet with the control setting.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
1	0x00	Speech Recognition is disabled
	0x01	Speech Recognition is enabled

2.3.33 GAIA play tone (0x0209)

This command causes the Device to play the indicated tone. This may be a built-in tone or one configured using the Set User Tone Configuration command, see Section GAIA set user-defined tone configuration (0x010e). The command payload indicates the tone number to play.

Command payload

Offset	Value	Comment
0	. . .	A number representing the tone to play

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code No tone is associated with the command payload number.

2.3.34 GAIA alert LEDs (0x020d)

This command displays a pattern on the Device LED indicators. The command payload of 13 octets holds the nine fields which define the pattern. See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for a description of these fields.

Command Payload

Offset	Value	Comment
0	. . .	LED On time, high byte
1	. . .	LED On time, low byte
2	. . .	LED Off time, high byte
3	. . .	LED Off time, low byte
4	. . .	LED Repeat time, high byte
5	. . .	LED Repeat time, low byte
6	. . .	LED Dim time, high byte
7	. . .	LED Dim time, low byte
8	. . .	Number of flashes
9	. . .	Timeout
10	. . .	LED ‘A’ selector
11	. . .	LED ‘B’ selector
12	. . .	Color

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
	`0x05`	'Invalid Parameter' status code One or more fields in the pattern is invalid

2.3.35 GAIA alert tone (0x020e)

This command plays a tone sequence on the Device. The command payload holds values for the tempo, volume, timbre and decay settings for the whole sequence followed by pitch and duration values for each note in the sequence. See Tones in QTIL application tone parameters.

Command payload

Offset	Value	Comment
0	. . .	Tempo of the tone sequence
1	. . .	Volume
2	. . .	Timbre

Offset	Value	Comment
3	. . .	Decay
4	. . .	Pitch of the first note in the sequence
5	. . .	Duration of the first note in the sequence
6 ...	. . .	Further notes in the sequence

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code One or more fields in the tone sequence is invalid

2.3.36 GAIA alert event (0x0210)

This command causes the Device to alert the user to an event. The method used to alert the user depends on the Device and its configuration. The Device can be configured to display an LED pattern, play a tone or issue an audio prompt for each event ID. See the *Audio Sink Application Configuration User Guide* or the *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for more information.

Command payload

Offset	Value	Comment
0	. . .	Event ID, high byte
1	. . .	Event ID, low byte

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The event ID is not recognized by the Device.

2.3.37 GAIA alert audio prompt (0x0211)

This command causes the Device to alert the user with an audio prompt. The command payload holds two octets indicating the 16-bit number of the audio prompt to play. The meaning of each number depends on the audio prompts loaded on the Device.

Command payload

Offset	Value	Comment
0	. . .	Audio prompt number, high byte
1	. . .	Audio prompt number, low byte

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The audio prompt ID is not recognized by the Device.

2.3.38 GAIA start speech recognition (0x0213)

This command starts the Simple Speech Recognition engine. The acknowledgement indicates that the engine was started. When a spoken response is detected a notification will be sent, if this has been registered, see Events in GAIA events.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.3.39 GAIA enter Bluetooth Pairing mode (0x021d)

This command puts the Device into Pairing mode.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x06`	'Incorrect State' status code The Device is not ready to be put into pairing mode.

2.3.40 GAIA set TWS audio routing (0x0224)

This command controls the audio routing to a Device in a Qualcomm® TrueWireless™ Stereo system. See the *Audio Sink Application Peer Device User Guide* or the *CSRA64xxx A11 TWS User Guide* for more information.

Command payload

Offset	Value	Comment
0	`0x00`	Select Master Device
	`0x01`	Select Slave Device
1	`0x00`	Route Both Stereo Channels to Selected Device
	`0x01`	Route Left Channel to Selected Device
	`0x02`	Route Right Channel to Selected Device
	`0x03`	Route Mono Audio to Selected Device

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code. The Device selector or the routing control octet is incorrect.

2.3.41 GAIA get TWS audio routing (0x02a4)

This command gets the current audio routing for a Device in a TrueWireless Stereo system. See the *Audio Sink Application Peer Device User Guide* or the *CSRA64xxx A11 TWS User Guide* for more information.

Command payload

Offset	Value	Comment
0	`0x00`	Select Master Device
	`0x01`	Select Slave Device

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code. The Device selection octet is incorrect.
1	`0x00`	Master Device is selected
	`0x01`	Slave Device is selected
2	`0x00`	Route Both Stereo Channels to Selected Device
	`0x01`	Route Left Channel to Selected Device

Offset	Value	Comment
	`0x02`	Route Right Channel to Selected Device
	`0x03`	Route Mono Audio to Selected Device

2.3.42 GAIA trim TWS volume (0x0226)

This command trims the volume of a Device in a TrueWireless Stereo system. See the *Audio Sink Application Peer Device User Guide* or *CSRA64xxx A11 TWS User Guide* for more information.

Command payload

Offset	Value	Comment
0	`0x00`	Select Master Device
	`0x01`	Select Slave Device
1	`0x00`	Volume Up
	`0x01`	Volume Down

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code. The Selected Device or the Volume Direction are incorrect.

2.3.43 GAIA set peer link reserved (0x0227)

This command reserves one Multipoint link for use by a TrueWireless device. See the *Audio Sink Application Peer Device User Guide* or the *CSRA64xxx A11 TWS User Guide* for more information.

Command payload

Offset	Value	Comment
0	`0x00`	Do not reserve a link
	`0x01`	Reserve a link

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code. The control octet was other than 0 or 1.

2.3.44 GAIA get peer link reserved (0x02a7)

This command gets the status of the Reserved Peer Link.

Command Payload

None

Acknowledgement Payload

Offset	Value	Comment
0	0x00	'Success' status code
1	0x00	Peer Link is not reserved
	0x01	Peer Link is reserved

2.3.45 GAIA find my remote (0x022b)

This command controls the Find Me alert function on a Bluetooth low energy HID Remote connected to the Device.

Command payload

Offset	Value	Comment
0	0x00	“No Alert”: no alerting is done on the HID Remote
	0x01	“Mild Alert”: the HID Remote alerts
	0x02	“High Alert”: the HID Remote alerts in the strongest possible way.

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code.

2.3.46 GAIA Set Data Endpoint mode (0x022e)

This command sets the transfer mode to use for the Data Endpoint handle.

Command payload

Offset	Value	Comment
0	0x00	Disable data endpoint
	0x01	RWCP (Reliable Write Command Protocol)

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code.

2.3.47 GAIA Get Data Endpoint mode (0x02ae)

This command gets the transfer mode currently set for the Data Endpoint.

Command payload

None.

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x00`	None
	`0x01`	RWCP

2.4 GAIA polled status commands

The following commands can be used to poll an application's status using GAIA:

- GAIA get API version (0x0300)
- GAIA get current RSSI (0x0301)
- GAIA get current battery level (0x0302)
- GAIA get module ID (0x0303)
- GAIA get application version (0x0304)

2.4.1 GAIA get API version (0x0300)

This command gets the highest supported QTIL GAIA protocol version and the QTIL GAIA API version running on the Device. This document describes protocol version 1 and API version 2.5.

Command payload

None

Acknowledgement payload (Example, API v2.5)

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x01`	Highest supported QTIL GAIA protocol version

Offset	Value	Comment
2	`0x02`	API major version number
3	`0x05`	API minor version number

2.4.2 GAIA get current RSSI (0x0301)

This command gets the current Reported Signal Strength Indication from the Device. The acknowledgement payload holds one octet with the command status and one octet giving the RSSI. The RSSI is returned in decibel-milliwatts in twos-complement form.

Command payload

None

Acknowledgement payload (Example, RSSI -30 dBm)

Offset	Value	Comment
0	`0x00`	'Success’ status code
1	`0xe2`	-30

2.4.3 GAIA get current battery level (0x0302)

This command gets the battery level from the Device. The acknowledgement payload holds one octet with the command status and two octets giving the battery level as a 16-bit number of millivolts.

Command payload

None

Acknowledgement payload (Example: 0x0e74 = 3700 mV)

Offset	Value	Comment
0	`0x00`	'Success’ status code
1	`0x0e`	3700, high byte
2	`0x74`	3700, low byte

2.4.4 GAIA get module ID (0x0303)

This command gets module identification information from the Device. The acknowledgement payload contains three values, the 16-bit Hardware ID (from hardware), the 16-bit Design ID and the 32-bit Module ID (from PS Keys).

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	. . .	Hardware ID Bits[8:15]
2	. . .	Hardware ID Bits[0:7]
3	. . .	Design ID Bits[815]
4	. . .	Design ID Bits[0:7]
5	. . .	Module ID Bits[24:31]
6	. . .	Module ID Bits[16:23]
7	. . .	Module ID Bits[8:15]
8	. . .	Module ID Bits[0:7]

2.4.5 GAIA get application version (0x0304)

This command gets application version information from the Device. The acknowledgement payload contains the command status and 8 further octets holding Device Identification information according to the Bluetooth Device Identification Profile.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	. . .	Vendor ID Source, high byte
2	. . .	Vendor ID Source, low byte
3	. . .	Vendor ID, high byte
4	. . .	Vendor ID, low byte
5	. . .	Product ID, high byte
6	. . .	Product ID, low byte
7	. . .	Product Version, high byte
8	. . .	Product Version, low byte

2.5 GAIA partition and file commands

The following commands provide Read access to files within the file system and allow partitions to be updated and mounted on devices that support external file systems:

- GAIA get storage partition status (0x0610)
- GAIA open storage partition (0x0611)

- GAIA write storage partition (0x0615)
- GAIA close storage partition (0x0618)
- GAIA mount storage partition (0x061a)
- GAIA open file (0x0621)
- GAIA read file (0x0624)
- GAIA close file (0x0628)
- GAIA get mounted partitions (0x01a0)

2.5.1 GAIA get storage partition status (0x0610)

This command finds the partition status. The acknowledgement holds the type, size and mounted state of the partition.

Command payload

Offset	Value	Comment
0	`0x00`	Memory type, internal flash
	`0x01`	Memory type, external SQIF
1	`...`	The index of the partition to be queried

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success’ status code
1	`...`	Memory type
2	`...`	Partition index
3	`...`	Type of partition
4	`...`	Partition size bits 24 to 31
5	`...`	Partition size bits 16 to 23
6	`...`	Partition size bits 8 to 15
7	`...`	Partition size bits 0 to 7
8	`...`	Partition mounted state

2.5.2 GAIA open storage partition (0x0611)

Opens a SQIF partition for writing with one or two possible command payload types. If no CRC check is required, the command payload holds the memory type identifier and the index of the partition to be opened. If a CRC check is required, the command payload also includes the 32-bit CRC.

The acknowledgement payload contains two octets, one octet providing the command status, the other the number of the opened stream.

Command payload with no CRC

Offset	Value	Comment
0	0x01	Memory type, external SQIF
1	. . .	The index of the partition to be opened

Command payload with 32-bit CRC

Offset	Value	Comment
0	0x01	Memory type, external SQIF
1	. . .	The index of the partition to be opened
2	. . .	CRC Bits[24: 31]
3	. . .	CRC Bits[16:23]
4	. . .	CRC Bits[8:15]
5	. . .	CRC Bits[0:7]

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
1	. . .	GAIA stream number

2.5.3 GAIA write storage partition (0x0615)

This command writes data to the previously opened partition. The command payload contains the opened stream GAIA stream number that is incremented by one, each time a packet is written, and the data to be written. The first packet written must have a sequence number of 0.

To avoid exhausting device memory, wait for acknowledgements when attempting to send large quantities of data.

Command payload

Offset	Value	Comment
0	. . .	GAIA stream number to be written
1	. . .	Sequence number bits 24 to 31
2	. . .	Sequence number bits 16 to 23
3	. . .	Sequence number bits 8 to 15
4	. . .	Sequence number bits 0 to 7
5 ...	. . .	Data to be written, 1 to 58 octets

2.5.4 GAIA close storage partition (0x0618)

This command closes a previously opened storage partition. The acknowledgement also indicates a successful CRC check, if appropriate.

Command payload

Offset	Value	Comment
0	. . .	GAIA stream number to be closed

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The GAIA stream number is not an open stream or the CRC check (if used) is not correct
1	. . .	GAIA stream number

2.5.5 GAIA mount storage partition (0x061a)

This command mounts the specified partition into the union file system.

Command payload

Offset	Value	Comment
0	0x00	Memory type, internal flash
	0x01	Memory type, external SQIF
1	. . .	The index of the partition to be mounted
2	0x00	Higher priority, mount before existing partitions
	0x01	Lower priority, mount after existing partitions

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
	0x05	'Invalid Parameter' status code The partition does not exist or the priority is not recognized.

2.5.6 GAIA open file (0x0621)

This command opens a file in the file system for Read operations. The command payload holds the file open mode (only 'read' mode is supported), and the name of the file. The acknowledgement payload contains two octets, one indicating the command status and the other giving the number of the opened stream.

Command payload (example, file name test.txt)

Offset	Value	Comment
0	`0x00`	'Read' mode
1	`0x74`	`'t'`
2	`0x65`	`'e'`
3	`0x73`	`'s'`
4	`0x74`	`'t'`
5	`0x2e`	`'.'`
6	`0x74`	`'t'`
7	`0x78`	`'x'`
8	`0x74`	`'t'`

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The file does not exist or the mode is not valid.

2.5.7 GAIA read file (0x0624)

This command reads data from a previously opened file. The command payload holds the stream number to read from and a sequence number. The first packet read must have a sequence number of 0. The sequence number must increase by one each time a packet is read.

Command payload

Offset	Value	Comment
0	...	GAIA stream number to be read
1	...	Sequence number bit [24:31]
2	...	Sequence number bits [16:23]
3	...	Sequence number bits [8:15]
4	...	Sequence number bits [0:7]

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	. . .	GAIA stream number
2	. . .	Sequence number bit [24:31]
3	. . .	Sequence number bits [16:23]
4	. . .	Sequence number bits [8:15]
5	. . .	Sequence number bits [0:7]
6 …	. . .	Data from file

2.5.8 GAIA close file (0x0628)

This command closes a file that was previously open.

Command payload

Offset	Value	Comment
0	. . .	GAIA stream number to be closed

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The GAIA stream number is not an open stream
1	. . .	GAIA stream number

2.5.9 GAIA get mounted partitions (0x01a0)

The Device maintains a bitmap of partitions it has currently mounted in order to play audio prompts. This can be queried using the Get Mounted Partitions command. This saves time compared to querying the status of multiple partitions individually. The acknowledgement payload holds one octet with the command status and one octet holding the bitmap.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	. . .	Partitions bitmap

2.6 GAIA no operation (0x0700) debugging command

This command has no effect other than to produce an acknowledgement.

Command payload

None

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code

2.7 GAIA notification commands

The following commands can be used to control notifications within an application using GAIA.

- GAIA register notification (0x4001)
- GAIA get notification (0x4081)
- GAIA cancel notification (0x4002)
- GAIA event notification (0x4003)

2.7.1 GAIA register notification (0x4001)

This command is used to inform the Host, when a specified event occurs on the Device. The command payload may contain extra data for some events. See Events in GAIA events.

Command payload

Offset	Value	Comment
0	...	An Event Type code from GAIA events
1 ...	...	Event-specific data, see the individual events

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The Event ID is not recognized or the event-specific data is incorrect
1	...	An Event Type code from GAIA events

2.7.2 GAIA get notification (0x4081)

This command gets details of the registered notifications of a given type. The acknowledgement payload holds the command status, the event type code and the number of events registered. There may be more octets depending on the event type. See the Events in GAIA events for details.

For simple 'enable' or 'disable' notifications, the number of events is 1 (Enabled) or 0 (Disabled).

Command payload

Offset	Value	Comment
0	. . .	An Event Type code from GAIA events

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The Event Type code is not recognized
1	. . .	An Event Type code from GAIA events
2	. . .	The number of registered events of this type
3 ...	. . .	Event-specific data; see the individual events

2.7.3 GAIA cancel notification (0x4002)

This command stops notifications of the given type.

Command payload

Offset	Value	Comment
0	. . .	An Event Type code from GAIA events

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
	`0x05`	'Invalid Parameter' status code The Event Type code is not recognized
1	. . .	An Event Type code from GAIA events

2.7.4 GAIA event notification (0x4003)

An Event Notification command is sent from the Device to the Host whenever a registered event occurs. The payload holds the Event Type and any event-specific data. An acknowledgement must be sent from the Host to the Device. See GAIA events. for details.

Notification payload (Example,Battery Level = 3380 mV, 0x0d34)

Offset	Value	Comment
0	`0x03`	Event Type, Battery Low Threshold
1	`0x0d`	Event-specific data, battery level, 3380 mV (high byte)
2	`0x34`	Event-specific data, battery level, 3380 mV (low byte)

Notification acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x03`	Event Type, Battery Low Threshold

2.8 GAIA events

To reduce radio traffic and increases battery life, the QTIL GAIA protocol supports Event Notifications of, for example, the asynchronous indication of a change in Device state. Each supported event is given a numeric code. The Host always acknowledges receipt, when notified by the Device.

The payloads of these commands are described in:

- GAIA RSSI low threshold (Event code 0x01)
- GAIA RSSI high threshold (Event code 0x02)
- GAIA battery low threshold (Event code 0x03)
- GAIA battery high threshold (Event code 0x04)
- GAIA battery charged (Event code 0x08)
- GAIA PIO changed (Event code 0x06)
- GAIA charger connection (Event code 0x09)
- GAIA user action (Event code 0x0b)
- GAIA speech recognition (Event code 0x0c)

2.8.1 GAIA RSSI low threshold (Event code 0x01)

This event shows that the signal strength has fallen to or below a given value. The Register Notification command payload includes two or three octets, used to set one or two thresholds.

Register notification command payload (Example: notification at -40 dBm and -50 dBm)

Offset	Value	Comment
0	`0x01`	Event Type, RSSI Low Threshold
1	`0xd8`	Event-specific data, -40 dBm
2	`0xce`	Event-specific data, -50 dBm

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x01`	Event Type, RSSI Low Threshold

2.8.2 GAIA RSSI high threshold (Event code 0x02)

This event shows that signal strength has risen to or above a given value. The command sets one or two thresholds.

Register notification command payload (Example: one threshold at -30 dBm)

Offset	Value	Comment
0	`0x02`	Event Type, RSSI High Threshold
1	`0xe2`	Event-specific data, -30 dBm

Acknowledgement Payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x02`	Event Type, RSSI High Threshold

2.8.3 GAIA battery low threshold (Event code 0x03)

This event shows that the battery voltage has fallen to or below a given value. The command payload has three or five octets to set one or two thresholds.

Register notification command payload (Example: notification at 3400 mV, 0x0d48)

Offset	Value	Comment
0	`0x03`	Event Type, Battery Low Threshold
1	`0x0d`	Event-specific data, battery level, 3400 mV (high byte)
2	`0x48`	Event-specific data, battery level, 3400 mV (low byte)

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x03`	Event Type, Battery Low Threshold

2.8.4 GAIA battery high threshold (Event code 0x04)

This event shows that the battery voltage has risen to or above a given value. The command payload has three or five octets to set one or two thresholds.

Register notification command payload (Example: notification at 3600 mV and 3700 mV)

Offset	Value	Comment
0	`0x04`	Event Type, Battery High Threshold
1	`0x0e`	Event-specific data, battery level, 3600 mV (high byte)
2	`0x10`	Event-specific data, battery level, 3600 mV (low byte)
3	`0x0e`	Event-specific data, battery level, 3700 mV (high byte)
4	`0x74`	Event-specific data, battery level, 3700 mV (high byte)

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x04`	Event Type, Battery High Threshold

2.8.5 GAIA PIO changed (Event code 0x06)

This event shows that the state of PIOs connected to a device has changed. The notification payload holds one octet with the event type, four octets with a timestamp and four octets with the values of up to 32 PIOs.

Register notification command payload

Offset	Value	Comment
0	`0x06`	Event Type, PIO Changed

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x06`	Event Type, PIO Changed

Notification payload

Offset	Value	Comment
0	0x06	Event Type, PIO Changed
1	. . .	Time stamp, Bits[24:31]
2	. . .	Time stamp, Bits[16:23]
3	. . .	Time stamp, Bits [8:15]
4	. . .	Time stamp, Bits[0:7]
5	. . .	PIO state, Bits[24:31]
6	. . .	PIO state, Bits[16:23]
7	. . .	PIO state, Bits[8:15]
8	. . .	PIO state, Bits[0:7]

2.8.6 GAIA battery charged (Event code 0x08)

This event shows that the battery has become fully charged. There is no event-specific data.

Register notification command payload

Offset	Value	Comment
0	0x08	Event Type, Battery Charged

Acknowledgement payload

Offset	Value	Comment
0	0x00	'Success' status code
1	0x08	Event Type, Battery Charged

Notification payload

Offset	Value	Comment
0	0x08	Event Type, Battery Charged

2.8.7 GAIA charger connection (Event code 0x09)

This event shows that the battery charger has been either connected or disconnected. The notification payload holds the Event Type code (0x09) and one further octet indicating Connection (0x01) or Disconnection (0x00).

Register notification command payload

Offset	Value	Comment
0	0x09	Event Type, Charger Connection

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x09`	Event Type, Charger Connection

Notification payload

Offset	Value	Comment
0	`0x09`	Event Type, Charger Connection
1	`0x00`	Charger Disconnected
	`0x01`	Charger Connected

2.8.8 GAIA user action (Event code 0x0b)

This event shows that the Device user has operated a control. The notification payload holds the event type (`0x0c`) and two further octets holding the 16-bit action code. See the *Audio Sink Application Configuration User Guide* or *CSRA64xxx A11 ROM Series Configuration Bit Field Application Note* for a list of user events.

Register notification command payload

Offset	Value	Comment
0	`0x0b`	Event Type, User Action

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x0b`	Event Type, User Action

Notification payload

Offset	Value	Comment
0	`0x0b`	Event Type, User Action
1	`...`	Action Code, high byte
2	`...`	Action Code, low byte

2.8.9 GAIA speech recognition (Event code 0x0c)

This event shows that the Simple Speech Recognition engine has detected an input. The notification payload holds the Event Type code (`0x0c`) and one further octet indicating the detected response.

Register notification command payload

Offset	Value	Comment
0	`0x0c`	Event Type, Speech Recognition

Acknowledgement payload

Offset	Value	Comment
0	`0x00`	'Success' status code
1	`0x0c`	Event Type, Speech Recognition

Notification payload

Offset	Value	Comment
0	`0x0c`	Event Type, Speech Recognition
1	`0x00`	An unrecognized response was detected
	`0x01`	The word “No” was recognized
	`0x02`	The word “Yes” was recognized

3 Example GAIA exchanges

The examples in this section show QTIL GAIA packets in hexadecimal with colons separating the header, payload and checksum components.

3.1 GAIA no operation example

This is the simplest complete QTIL GAIA protocol exchange. The command has no payload and the response is a simple acknowledgement of success. To request a no operation the Host sends:

```
ff 01 01 00 00 0a 07 00 : : f2
```

Table 3-1 describes the request.

Table 3-1 GAIA no operation host example request

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x01`	Flags: `GAIA_FLAG_CHECK`
Header	`0x00`	Payload length (0)
Header	`0x000a`	Vendor ID
Header	`0x0700`	Command ID (No Operation)
		This request has no payload.
Checksum	`0xf2`	Checksum (`0xff` ⊕ `0x01` ⊕ `0x01` ⊕ `0x00` ⊕ `0x00` ⊕ `0x0a` ⊕ `0x07` ⊕ `0x00`)

The expected response is:

```
ff 01 01 01 00 0a 87 00 : 00 : 73
```

Table 3-2 describes the response.

Table 3-2 Description of no operation device example response

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x01`	Flags: `GAIA_FLAG_CHECK`
Header	`0x01`	Payload length (1)
Header	`0x000a`	Vendor ID (CSR)

Table 3-2 Description of no operation device example response (cont.)

Type	Data	Comment
Header	`0x8700`	Acknowledged Command ID (No Operation)
Payload	`0x00`	Status (success)
Checksum	`0x73`	Checksum (`0xff` ⊕ `0x01` ⊕ `0x01` ⊕ `0x01` ⊕ `0x00` ⊕ `0x0a` ⊕ `0x87` ⊕ `0x00` ⊕ `0x00`)

This example shows the use of the optional GAIA packet checksum.

3.2 GAIA get current RSSI example

To get the received signal strength indication from the Device, the Host application sends:

```
ff 01 00 00 00 0a 03 01 : :
```

Table 3-3 describes the request.

Table 3-3 Description of the get current RSSI example request

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x00`	Flags (none)
Header	`0x00`	Payload length (0)
Header	`0x000a`	Vendor ID
Header	`0x0301`	Command ID (Get Current RSSI)
		This request does not have a payload.

A typical response would be:

```
ff 01 00 02 00 0a 83 01 : 00 ec :
```

Table 3-4 describes the response.

Table 3-4 Description of the get current RSSI example response

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x00`	Flags (none)
Header	`0x02`	Payload length (2)
Header	`0x000a`	Vendor ID
Header	`0x8301`	Acknowledged Command ID (Get Current RSSI)
Payload	`0x00`	Status (success)
Payload	`0xec`	RSSI value (-20 dBm)

3.3 GAIA alert tone example

The Host sends:

```
ff 01 00 10 00 0a 02 0e : 14 1f 00 20 2e 08 30 08 2c 08 20 08 27 02 7F
08 :
```

Table 3-5 describes the request.

Table 3-5 Description of the alert tone example request

Type	Data	Comment
Header	0xff	Start of Frame
Header	0x01	QTIL GAIA Protocol version 1
Header	0x00	Flags (none)
Header	0x0e	Payload length (14)
Header	0x000a	Vendor ID
Header	0x020e	Command ID (Alert Tone)
Payload	0x14	The tone tempo (20 × 4 = 80 crotchets per minute)
Payload	0x1f	The tone volume (31 = maximum volume)
Payload	0x00	The timbre (0 = sine wave)
Payload	0x20	The decay value (32)
Payload	0x2e08	The first note (46 = B♭3, 8 = quaver)
Payload	0x3008	The second note (48 = C4, 8 = quaver)
Payload	0x2c08	The third note (44 = A♭3, 8 = quaver)
Payload	0x2008	The fourth note (32 = A♭2, 8 = quaver)
Payload	0x2702	The fifth note (39 = E♭3, 2 = minim)
Payload	0x7f08	The final note (127 = rest, 8 = quaver)

NOTE To avoid a click at the end of the tone, the sequence ends with a rest.

The Device acknowledges the command:

```
ff 01 00 01 00 0a 82 0e : 00 :
```

Table 3-6 describes the response.

Table 3-6 Description of the alert tone example response

Type	Data	Comment
Header	0xff	Start of Frame
Header	0x01	QTIL GAIA Protocol version 1
Header	0x00	Flags (none)
Header	0x01	Payload length [(1)]
Header	0x000a	Vendor ID

Table 3-6 Description of the alert tone example response (cont.)

Type	Data	Comment
Header	0x820e	Acknowledged Command ID (Alert Tone)
Payload	0x00	Status (success)

3.4 GAIA register notification (battery low threshold) example

To receive a notification when the battery voltage falls to 3500 mV, the Host sends:

```
ff 01 00 03 00 0a 40 01 : 03 0d ac :
```

Table 3-7 describes the request.

Table 3-7 Description of the register notification example request

Type	Data	Comment
Header	0xff	Start of Frame
Header	0x01	QTIL GAIA Protocol version 1
Header	0x00	Flags (none)
Header	0x03	Payload length (3)
Header	0x000a	Vendor ID
Header	0x4001	Command ID (Register Notification)
Payload	0x03	Notification Event Type (Battery Low Threshold)
Payload	0x0dac	Threshold Voltage (3500 mV)

The Device acknowledges the command by sending:

```
ff 01 00 02 00 0a c0 01 : 00 03 :
```

Table 3-8 describes the response.

Table 3-8 Description of the register notification example response

Type	Data	Comment
Header	0xff	Start of Frame
Header	0x01	QTIL GAIA Protocol version 1
Header	0x00	Flags (none)
Header	0x02	Payload length (2)
Header	0x000a	Vendor ID
Header	0xc001	Acknowledged Command ID (Register Notification)
Payload	0x00	Status (success)
Payload	0x03	Notification Event Type (Battery Low Threshold)

If the Host now sends a Get Notification command the details of the registered events will be returned.

The Host sends:

```
ff 01 00 03 00 0a 40 81 : 03 :
```

Table 3-9 describes the request.

Table 3-9 Description of the get notification example request

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x00`	Flags (none)
Header	`0x01`	Payload length (1)
Header	`0x000a`	Vendor ID
Header	`0x4081`	Command ID (Get Notification)
Payload	`0x03`	Notification Event Type (Battery Low Threshold)

The Device responds with:

```
ff 01 00 05 00 0a c0 81 : 00 03 01 0d ac :
```

Table 3-10 describes the response.

Table 3-10 Description of the get notification example response

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x00`	Flags (none)
Header	`0x05`	Payload length (5)
Header	`0x000a`	Vendor ID
Header	`0xc081`	Acknowledged Command ID (Get Notification)
Payload	`0x00`	Status (success)
Payload	`0x03`	Notification Event Type (Battery Low Threshold)
Payload	`0x01`	Number of Registered Notifications (1)
Payload	`0x0dac`	Threshold Voltage (3500 mV)

If at any time later the Device battery voltage is measured at 3480 mV for example, then the Device sends:

```
ff 01 00 03 00 0a 40 03 : 03 0d 98 :
```

Table 3-11 describes the event notification.

Table 3-11 Description of the event notification example

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1

Table 3-11 Description of the event notification example (cont.)

Type	Data	Comment
Header	`0x00`	Flags (none)
Header	`0x03`	Payload length (3)
Header	`0x000a`	Vendor ID
Header	`0x4003`	Command ID (Event Notification)
Payload	`0x03`	Notification Event Type (Battery Low Threshold)
Payload	`0x0d98`	Battery Voltage (3480 mV)

In this case the command is sent from the Device to the Host. The Host must acknowledge the command. The Host sends:

```
ff 01 00 02 00 0a c0 03 : 00 03 :
```

Table 3-12 describes the acknowledgement.

Table 3-12 Description of the event notification acknowledgement example

Type	Data	Comment
Header	`0xff`	Start of Frame
Header	`0x01`	QTIL GAIA Protocol version 1
Header	`0x00`	Flags (none)
Header	`0x02`	Payload length (2)
Header	`0x000a`	Vendor ID
Header	`0xc003`	Acknowledged Command ID (Event Notification)
Payload	`0x00`	Status (success)
Payload	`0x03`	Notification Event Type (Battery Low Threshold)

A GAIA commands defined in API v2.5

The tables list the Commands supported by API v2.5

Table A-1 0x01xx configuration commands

CommandID	IDCommand
0x0101	Set LED Configuration
0x0102	Set Tone Configuration
0x0103	Set Default Volume
0x0103	Set Default Volume
0x0106	Set Audio Prompt Configuration
0x0109	Set Timer Configuration
0x010a	Set Audio Gain Configuration
0x010b	Set Volume Configuration
0x010e	Set User-Defined Tone Configuration
0x0181	Get LED Configuration
0x0182	Get Tone Configuration
0x0183	Get Default Volume
0x0186	Get Audio Prompt Configuration
0x0189	Get Timer Configuration
0x018a	Get Audio Gain Configuration
0x018b	Get Volume Configuration

Table A-2 0x02xx control commands

ID	Command
0x0201	Change Volume
0x0202	Device Reset
0x0204	Power Off
0x0205	Set Volume Orientation
0x0207	Set LED Control
0x0209	Play Tone
0x020a	Set Audio Prompt Control

Table A-2 0x02xx control commands (cont.)

ID	Command
`0x020b`	Change Audio Prompt Language
`0x020c`	Set Speech Recognition Control
`0x020d`	Alert LEDs
`0x020e`	Alert Tone
`0x0210`	Alert Event
`0x0211`	Alert Audio Prompt
`0x0212`	Set Audio Prompt Language
`0x0213`	Start Speech Recognition
`0x0214`	Set EQ Control
`0x0215`	Set Bass Boost/Plus Control
`0x0216`	Set 3D/3D-virtualisation Control
`0x0217`	Switch EQ Control
`0x0218`	Toggle Bass Boost/Plus Control
`0x0219`	Toggle 3D/3D-virtualisation Control
`0x021a`	Set User EQ Parameter
`0x021b`	Set User EQ Group Parameter
`0x021d`	Enter Bluetooth Pairing Mode
`0x021f`	AV Remote Control
`0x0220`	Set User EQ Control
`0x0221`	Toggle User EQ
`0x0224`	Set TWS Audio Routing
`0x0226`	Trim TWS Volume
`0x0227`	Set Peer Link Reserved
`0x022b`	Find My Remote
`0x0285`	Get Volume Orientation
`0x0287`	Get LED Control
`0x028a`	Get Audio Prompt Control
`0x028c`	Get Speech Recognition Control
`0x0292`	Get Audio Prompt Language
`0x0294`	Get EQ Control
`0x0295`	Get Bass Boost/Plus Control
`0x0296`	Get 3D/3D-virtualisation Control
`0x029a`	Get User EQ Parameter
`0x029b`	Get User EQ Group Parameter
`0x02a0`	Get User EQ Control

Table A-2 0x02xx control commands (cont.)

ID	Command
`0x02a4`	Get TWS Audio Routing
`0x02a7`	Get Peer Link Reserved

Table A-3 0x03xx polled status commands

ID	Command
`0x0300`	Get API Version
`0x0301`	Get Current RSSI
`0x0302`	Get Current Battery Level
`0x0303`	Get Module ID

Table A-4 0x06xx data transfer commands

ID	Command
`0x0610`	Get Storage Partition Status
`0x0611`	Open Storage Partition
`0x0615`	Write Storage Partition
`0x0618`	Close Storage Partition
`0x061a`	Mount Storage Partition
`0x0621`	Open File
`0x0624`	Read File

Table A-5 0x0700 ‘No Operation’ debugging command

ID	Command
`0x0700`	No Operation

Table A-6 0x40xx notification commands

ID	Response
`0x4001`	Register Notification
`0x4002`	Cancel Notification
`0x4003`	Event Notification
`0x4081`	Get Notification

B GAIA notification event codes

Table B-1 lists the notifications defined in API v2.5.

Table B-1 Notification event codes

Event	Code	Description
RSSI Low Threshold	`0x01`	An **RSSI Low Threshold** event occurs whenever the Received Signal Strength Indication falls to or below a preset threshold. One or two thresholds may be set, each expressed as a single octet representing a signal level in decibel milliwatts in twos-complement form.
RSSI High Threshold	`0x02`	An **RSSI High Threshold** event occurs whenever the Received Signal Strength Indication rises to or above a preset threshold. One or two thresholds may be set, each expressed as a single octet representing a signal level in decibel milliwatts in twos-complement form.
Battery Low Threshold	`0x03`	A **Battery Low Threshold** event occurs when the measured battery voltage falls to or below a preset threshold. One or two thresholds may be set, each represented by an unsigned 16-bit number of millivolts.
Battery High Threshold	`0x04`	A **Battery High Threshold** occurs when the measured battery voltage rises to or above a preset threshold. One or two thresholds may be set, each represented by an unsigned 16-bit number of millivolts.
PIO Changed	`0x06`	A **PIO Changed** event occurs when the logical state of an input PIO changes.
Battery Charged	`0x08`	A **Battery Charged** event occurs when the Device hardware detects that the battery charging is complete.
Charger Connection	`0x09`	A **Charger Connection** event occurs when the Device hardware detects that the battery charger has become connected or disconnected.
User Action	`0x0a`	A **User Action** event occurs when the Device user changes a control setting.
Speech Recognition	`0x0b`	A **Speech Recognition** event occurs when the Simple Speech Recognition engine detects an input.

C GAIA command status codes

By convention, the first octet in an acknowledgement (ACK) packet is a Status code used to indicate the successful receipt or the reason for the failure of a command. Table C-1 lists the Status codes.

Table C-1 Command status codes

Status	Code	Description
Success	`0x00`	The command completed successfully.
Failed: Command Not Supported	`0x01`	An invalid Command ID was specified.
Failed: Insufficient Resources	`0x03`	The command is valid but the Device could not complete it successfully.
Failed: Invalid Parameter	`0x05`	An invalid parameter was used in the command.
Failed: Incorrect State	`0x06`	The Device is not in the correct state to process the command.
In Progress	`0x07`	The command is in progress [1].
[1] `GAIA_STATUS_IN_PROGRESS` (`0x07`) is the value of the Command Status characteristic until the operation completes, for use by Bluetooth low energy hosts that do not support GATT Notifications but need to check that a command has succeeded.		

D QTIL application tone parameters

Tones are defined by values for tempo, volume, timbre and, decay values for the whole sequence followed by pitch and duration value for each individual note.

A rest following a note causes the note to fade out and avoids a click at the end of a sequence.

NOTE Rest length should be adjusted to avoid a long silence after the sequence.

Tempo

The `Tempo` parameter sets the speed at which the tone sequence will be played. The value is scaled by 4 to give the tempo in crotchets per minute, so a value of 30 gives a tempo of 120 crotchets per minute.

Volume

The `Volume` parameter sets the level at which the tone sequence will be played with 0 being silent and 31 being the loudest.

Timbre

The `Timbre` parameter selects the waveform of the tone generator and the timbre of the notes, as defined by the values in Table D-1 shows the valid values.

Table D-1 Note timbre values

Value	Waveform
0	Sine wave (a pure tone)
1	Square wave (a mellow timbre)
2	Sawtooth wave (a rich timbre)
3	Symmetrical triangle wave
4	Asymmetrical triangle wave
5	Clipped sine wave
6	Plucked string sound

Decay

The `Decay` parameter sets the rate at which each note dies away. A higher value gives a slower decay. A value of `0x08` causes each note to reach zero volume at half of its duration, to give a staccato feel. A value of `0x10` causes each note to reach zero volume just as the next note starts, giving a more legato result. A value of `0x20` causes each note to reach half its initial volume when the next note starts.

Pitch

This parameter sets the pitch of a note to be played. The available pitches are numbered from 0 (C0) to 119 (B9) with 48 representing C4 or Middle C. A value of 127 represents a rest (silence). See Table D-2 fo precise values.

Table D-2 Note pitch values

Pitch	8ve 0	8ve 1	8ve 2	8ve 3	8ve 4	8ve 5	8ve 6	8ve7	8ve 8	8ve 9
C	0	12	24	36	48	60	72	84	96	108
C♯ D♭	1	13	25	37	49	61	73	85	97	109
D	2	14	26	38	50	62	74	86	98	110
D♯ E♭	3	15	27	39	51	63	75	87	99	111
E	4	16	28	40	52	64	76	88	100	112
F	5	17	29	41	53	65	77	89	101	113
F♯ G♭	6	18	30	42	54	66	78	90	102	114
G	7	19	31	43	55	67	79	91	103	115
G♯ A♭	8	20	32	44	56	68	80	92	104	116
A	9	21	33	45	57	69	81	93	105	117
A♯ B♭	10	22	34	46	58	70	82	94	106	118
B	11	23	35	47	59	71	83	95	107	119

Duration

The `Duration` parameter sets the length of a note to be played. A value of 1 represents a semibreve (four crotchet beats). 64 represents a hemi-demi-semiquaver (1/64th of a crotchet beat).

E GAIA timer numbers

Table E-1 Sink Application Timer Numbers

Number	Timer	Unit
0	Automatic switch off time	Seconds
1	Limbo timeout	
2	Network Service Indicator repeat time	
3	Disable Power Off after Power On time	
4	Pairing timeout	
5	Mute Reminder time	
6	Connectable timeout	
7	Pairing mode timeout if PDL is empty	
8	Reconnection attempts	Attempts
9	Encryption refresh timeout	Minutes
10	Inquiry timeout	Seconds
11	Second AG connect delay	
12	Missed call indicator time	
13	Missed call indicator attempts	Attempts
14	A2DP link loss reconnection time	Seconds
15	Audio prompt set confirm time	
16	Speech recognition repeat time	ms
17	Store current sink volume and audio source time	Seconds
18	Wired audio connected power off timeout	
19	Store current PEQ settings time	
20	Defragmentation check time	
21	Audio amp power down in limbo time	
22	Immediate Alert timer	
23	Immediate Alert stop timeout	
24	Link Loss Alert timer	
25	Link Loss Alert stop timeout	
26	Audio amp unmute time	ms
27	Audio amp mute time	

Table E-1 Sink Application Timer Numbers (cont.)

Number	Timer	Unit
28	Party Mode music timeout	Seconds
29	Party Mode stream resume timeout	
30	BR/EDR Authenticated Payload timeout	
31	Bluetooth low energy Authenticated Payload timeout (seconds)	

F User EQ parameters

User EQ parameters are described in:

- Parameter ID format
- Filter type parameter
- Filter frequency parameter
- Gain parameter
- Q parameter
- Parameter ranges

F.1 Parameter ID format

The Parameter ID is a 16-bit number that represents the Parameter Number and Filter Bank and Filter Band being modified or accessed.:

Table F-1 Parameter ID format

15			12	11			8	7			4	3			0
0	0	0	0	X	X	X	X	Y	Y	Y	Y	Z	Z	Z	Z

Where:

XXXX: Is the bank of the filter being modified or accessed.

YYYY: Is the band of the filter being modified or accessed.

ZZZZ: Is the parameter number of the parameter being modified or accessed

Table F-2 User EQ parameter IDs

Bank	Band	Parameter	Description
0	0	0	Number of active banks (see note)
Non-zero	0	0	Number of bands in the specified bank (see note)
Non-zero	0	1	Master gain of the specified bank
Non-zero	Non-zero	0	Filter Type of the specified filter
Non-zero	Non-zero	1	Frequency of the specified filter

Table F-2 User EQ parameter IDs (cont.)

Bank	Band	Parameter	Description
Non-zero	Non-zero	2	Gain of the specified filter
Non-zero	Non-zero	3	Q of the specified filter

NOTE GAIA Version 2.5 only reconfigures Bank 1. The number of active Banks or Bands cannot be changed but setting the Filter Type to 0 (Bypass) will effectively remove that Band from the Bank.

Table F-3 User EQ parameterID examples

Example	Description
`0x0120`	Filter type of band 2 of filter bank 1
`0x0123`	Q of band 2 of filter bank 1

F.2 Filter type parameter

Table F-4 User EQ filter types

Value	Filter Type
0	Bypass
1	First order low pass
2	First order high pass
3	First order all pass
4	First order low shelf
5	First order high shelf
6	First order tilt
7	Second order low pass
8	Second order high pass
9	Second order all pass
10	Second order low shelf
11	Second order high shelf
12	Second order tilt
13	Parametric equalizer

F.3 Filter frequency parameter

The filter frequency is a 16-bit unsigned number. The parameter value is 3 times the filter frequency. For example a frequency of 20 kHz is represented by a parameter value of 60000 (`0xea60`).

F.4 Gain parameter

The gain is a 16-bit signed number. The parameter value is 60 times the gain in decibels. For example a gain of -9.6 dB is represented by a parameter value of -576 (`0xfdc0`).

F.5 Q parameter

The filter Q is a 16-bit unsigned number. The parameter value is 4096 times the Q. For example a Q of 4.0 is represented by a parameter value of 16384 (`0x4000`). A parameter value of 2896 (`0x0b50`) represents a Q of 0.70703125 (the nearest value to $\frac{1}{\sqrt{2}}$).

F.6 Parameter ranges

User EQ parameters must be within the range shown in Table F-5.

Table F-5 User EQ parameter ranges

Filter Type	Frequency	Gain	Q
Master Gain	N/A	-36 dB to +12 dB	N/A
First order high pass, all pass and low pass	0.3 Hz to 20 kHz	N/A	N/A
Second order high pass, all pass and low pass	40 Hz to 20 kHz	N/A	0.25 to 2.0
First order low shelf, high shelf and tilt	20 Hz to 20 kHz	-12 dB to +12 dB	N/A
Second order low shelf, high shelf and tilt	40 Hz to 20 kHz	-12 dB to +12 dB	0.25 to 2.0
Parametric Equalizer	20 Hz to 20 kHz	-36 dB to +12 dB	0.25 to 8.0

G Volume configuration parameters

Table G-1 Volume configuration parameter numbers

Parameter number	Parameter name	Description
0	Number of steps main	Number of steps of volume change permitted on main channels
1	Volume knee 1 main	First volume point at which curve of dB conversion changes on main channels
2	Volume knee 2 main	Second volume point at which curve of dB conversion changes on main channels
3	dB knee 1 main	First dB value for point at which curve of dB conversion changes on main channels
4	dB knee 2 main	Second dB value for point at which curve of dB conversion changes on main channels
5	dB max main	dB value at maximum volume level on main channels
6	dB min main	dB value at minimum volume level on main channels
7	Number of steps aux	Number of steps of volume change permitted on auxiliary channels
8	Volume knee 1 aux	First volume point at which curve of dB conversion changes on auxiliary channels
9	Volume knee 2 aux	Second volume point at which curve of dB conversion changes on auxiliary channels
10	dB knee 1 aux	First dB value for point at which curve of dB conversion changes on auxiliary channels
11	dB knee 2 aux	Second dB value for point at which curve of dB conversion changes on auxiliary channels
12	dB max aux	dB value at maximum volume level on auxiliary channels
13	dB min aux	dB value at minimum volume level on auxiliary channels
14	Device trim master	TWS master device trim gain in dB
15	Device trim slave	TWS slave device trim gain in dB
16	Device trim change	TWS device trim stepsize in dB
17	Device trim min	TWS device trim minimum gain in dB
18	Device trim max	TWS device trim maximum gain in dB

H GAIA PS Keys

The PS Keys that control the operation of QTIL GAIAthe ADK Sink implementation are.

PSKEY_MODULE_DESIGN (0x025a) and PSKEY_MODULE_ID (0x0259)

The Module ID key is sent in response to a Get Module ID command.

PSKEY_MOD_MANUF0 (0x025e)

Module Manufacturer key 0 holds an arbitrary Service Discovery Protocol record. If this record is valid, GAIA registers it. The first word holds the length of the SDP record. Further words hold the packed octets of the record.

Document references

Document	Reference
Company Identifiers page	www.bluetooth.org
Audio Sink Application Peer Device User Guide	80-CT414-1/CS-00316086-UG
CSRA64xxx A11 TWS User Guide	80-CT469-1/CS-00323830-UG
Audio Sink Application Configuration User Guide	80-CT451-1/CS-00334708-UG
CSRA64xxx A11 ROM Series Configuration Bit Field Application Note	80-CT479-1/CS-00334710-AN

Terms and definitions

Term	Definition
ACK	Acknowledgement
ADK	Audio or Application Development Kit
API	Application Programming Interface
Bluetooth	Set of technologies providing audio and data transfer over short-range radio connections
Bluetooth SIG	The Bluetooth Special Interest Group oversees the development of Bluetooth standards and the licensing of Bluetooth technologies and trademarks to manufacturers.
GAIA	Generic Application Interface Architecture
ID	Identifier
LED	Light-Emitting Diode
PEQ	Parametric Equalizer
PIO	Programmable Input/Output
PS	Persistent Store
QTIL	Qualcomm Technologies International, Ltd.
RSSI	Received Signal Strength Indication
RWCP	Reliable Write Command Protocol
SDP	Service Discovery Protocol
SIG	Special Interest Group
SPP	Serial Port Profile
SQIF	Serial Quad I/O Flash, a non-volatile memory technology
TWS	TrueWireless Stereo
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USB-IF	The USB Implementers' Forum is responsible for issuing USB vendor IDs to product manufacturers.