Melody Audio 7.x

User Guide

SIERRA WIRELESS®

41112426
Rev 3

Important Notice

Due to the nature of wireless communications, transmission and reception of data can never be guaranteed. Data may be delayed, corrupted (i.e., have errors) or be totally lost. Although significant delays or losses of data are rare when wireless devices such as the Sierra Wireless modem are used in a normal manner with a well-constructed network, the Sierra Wireless modem should not be used in situations where failure to transmit or receive data could result in damage of any kind to the user or any other party, including but not limited to personal injury, death, or loss of property. Sierra Wireless accepts no responsibility for damages of any kind resulting from delays or errors in data transmitted or received using the Sierra Wireless modem, or for failure of the Sierra Wireless modem to transmit or receive such data.

Safety and Hazards

Do not operate the Sierra Wireless modem in areas where blasting is in progress, where explosive atmospheres may be present, near medical equipment, near life support equipment, or any equipment which may be susceptible to any form of radio interference. In such areas, the Sierra Wireless modem **MUST BE POWERED OFF**. The Sierra Wireless modem can transmit signals that could interfere with this equipment.

Do not operate the Sierra Wireless modem in any aircraft, whether the aircraft is on the ground or in flight. In aircraft, the Sierra Wireless modem **MUST BE POWERED OFF**. When operating, the Sierra Wireless modem can transmit signals that could interfere with various onboard systems.

Note: Some airlines may permit the use of cellular phones while the aircraft is on the ground and the door is open. Sierra Wireless modems may be used at this time.

The driver or operator of any vehicle should not operate the Sierra Wireless modem while in control of a vehicle. Doing so will detract from the driver or operator's control and operation of that vehicle. In some states and provinces, operating such communications devices while in control of a vehicle is an offence.

Limitation of Liability

The information in this manual is subject to change without notice and does not represent a commitment on the part of Sierra Wireless. SIERRA WIRELESS AND ITS AFFILIATES SPECIFICALLY DISCLAIM LIABILITY FOR ANY AND ALL DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES INCLUDING, BUT NOT LIMITED TO, LOSS OF PROFITS OR REVENUE OR ANTICIPATED PROFITS OR REVENUE ARISING OUT OF THE USE OR INABILITY TO USE ANY SIERRA WIRELESS PRODUCT, EVEN IF SIERRA WIRELESS AND/OR ITS AFFILIATES HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

Notwithstanding the foregoing, in no event shall Sierra Wireless and/or its affiliates aggregate liability arising under or in connection with the Sierra Wireless product, regardless of the number of events, occurrences, or claims giving rise to liability, be in excess of the price paid by the purchaser for the Sierra Wireless product.

Patents

This product may contain technology developed by or for Sierra Wireless Inc. This product includes technology licensed from QUALCOMM®. This product is manufactured or sold by Sierra Wireless Inc. or its affiliates under one or more patents licensed from InterDigital Group and MMP Portfolio Licensing.

Copyright

Trademarks

Sierra Wireless®, AirPrime®, AirLink®, AirVantage® and the Sierra Wireless logo are registered trademarks of Sierra Wireless.

Windows® and Windows Vista® are registered trademarks of Microsoft Corporation.

Macintosh® and Mac OS X® are registered trademarks of Apple Inc., registered in the U.S. and other countries.

QUALCOMM® is a registered trademark of QUALCOMM Incorporated. Used under license.

Other trademarks are the property of their respective owners.

Contact Information

Sales information and technical support, including warranty and returns	Web: sierrawireless.com/company/contact-us/ Global toll-free number: 1-877-687-7795 6:00 am to 5:00 pm PST
Corporate and product information	Web: sierrawireless.com

Revision History

Revision number	Release date	Changes
1	June 2018	Draft release

Revision number	Release date	Changes
2	August 2018	Changed IAP references to iAP Corrected AVRCP_META_DATA example Corrected example outputs for several commands Added detailed parameter descriptions for TX_POWER Added notes for AUDIO_DIGITAL Updated Deep Sleep Mode description Added Examples appendix
3	August 2018	Update to Melody 7.2 • Updated Melody Commands—ASSOCIATION, BLE_GET_CHAR (added <properties>), BLE_GET_SERV (replaced <handle> with start/end handles), BROADCAST (clarified response notifications received), INQUIRY (response units corrected to dBm from dB), SCAN (added <type> to SCAN_RAW response, updated examples), STATUS (removed <assoc> value ’2’) • Updated Melody Configurations—AUDIO_DIGITAL (PCM notes), BA_CONFIG (removed broadcaster product id and version id parameters) • Updated Melody Unsolicited Notifications—Removed ASSOCIATION • Expanded Broadcast Audio description • Updated Broadcast Audio examples

Contents

1: Introduction

Melody is an embedded firmware solution running on the BC127 module offering both Classic Bluetooth and Bluetooth Low Energy (BLE) functionality, allowing Bluetooth product development without detailed knowledge of the Bluetooth standard.

Melody provides a simple high level UART-based command interface and all functionality needed to design a wireless audio device that can connect seamlessly to smartphone applications and deliver the best audio quality and user experience.

This document describes how to use and update Melody firmware, and is current to Melody 7.2.

Key Specifications

- Best-in-Class Configurable Bluetooth 5.0 + BLE and Wireless Audio
- Embedded Software and Stack that provides an abstraction layer of the Bluetooth Link
- Controlled via UART, USB
- BR/EDR Profiles: A2DP, AVRCP, DID, HFP, HID, HSP, iAP, MAP, PBAP, SPP
- BLE Profiles: GAP, GATT, BC Smart profile and generic GATT support
- Multi-point support
- Supports SBC, AAC, aptX, aptX Low Latency and aptX HD codecs
- True Wireless Stereo (TWS) support
- Analog and digital audio inputs and outputs (I2S, PCM or SPDIF)
- Supports High Speed data transfer SPP and iAP
- Remote commands over BLE or SPP
- GPIO control, LED indications
- Future proof—Uses the latest hardware and supports firmware upgrade (DFU)

Releases

Melody 7.x
- Standard release available from source.sierrawireless.com.

Melody 7.x MFI
- Available to Apple MFI licensees.
- Includes iAP support.
- Contact your Sierra Wireless Account Manager to obtain this version.

Melody 7.x HD
- Available for BC127-HD modules only.
- Includes aptX HD codec and supports 24 bit audio resolution.
- aptX Low Latency support disabled
- Available from source.sierrawireless.com

Melody 7.x BA
- Includes Broadcast Audio support
- aptX Low Latency support disabled
- Contact your Sierra Wireless Account Manager to obtain this version.

Applications

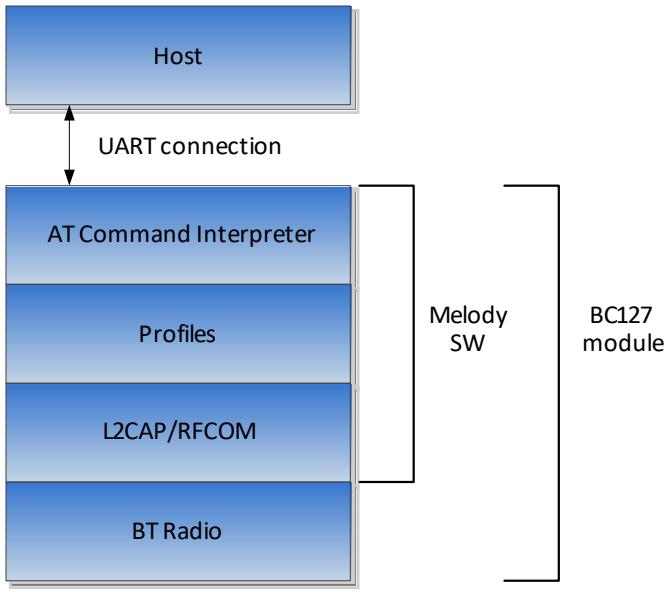


Figure 1-1: Component Block

- Wireless speakers, docks and headsets
- Smartphone controller audio system
- Automotive infotainment systems
- Medical devices
- High-quality audio streaming
- Gaming accessories and MP3 players

2: Quick Start

Setting Up

To start using Melody you need:

- Development board (BC127-DISKIT)
- Computer running a serial terminal (e.g. HyperTerminal, PuTTY for Windows, or an equivalent program) to communicate over the COM interface

Connect the BC127-DISKIT (USB<>UART connector) to the computer. The port enumerates as a USB Serial Port.

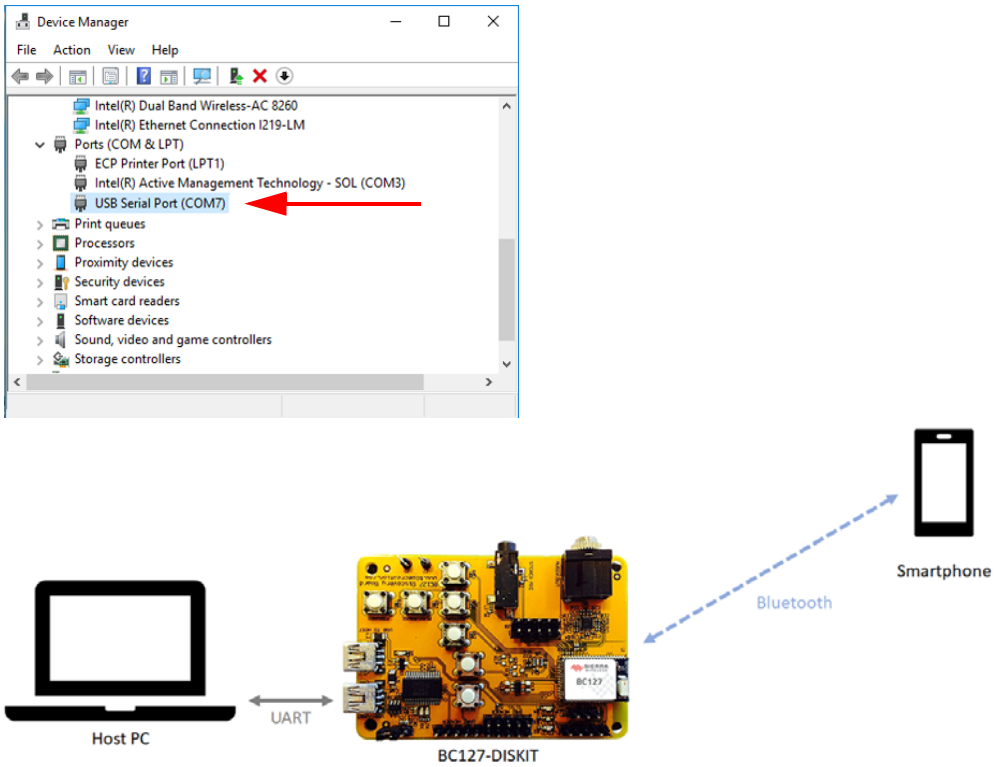


Figure 2-1: Example Standard Configuration

Table 2-1: Default UART Configuration

Baud Rate	9600 bps
Data Bits	8
Stop Bits	1
Parity Bit	None
HW Flow Control	Disabled

Note: The end of line (EOL) character used by Melody is the Carriage Return (CR).

Once you have configured your serial terminal and opened the COM port, press the DISKIT's RESET button. A prompt should appear on the terminal indicating the module is ready to operate.

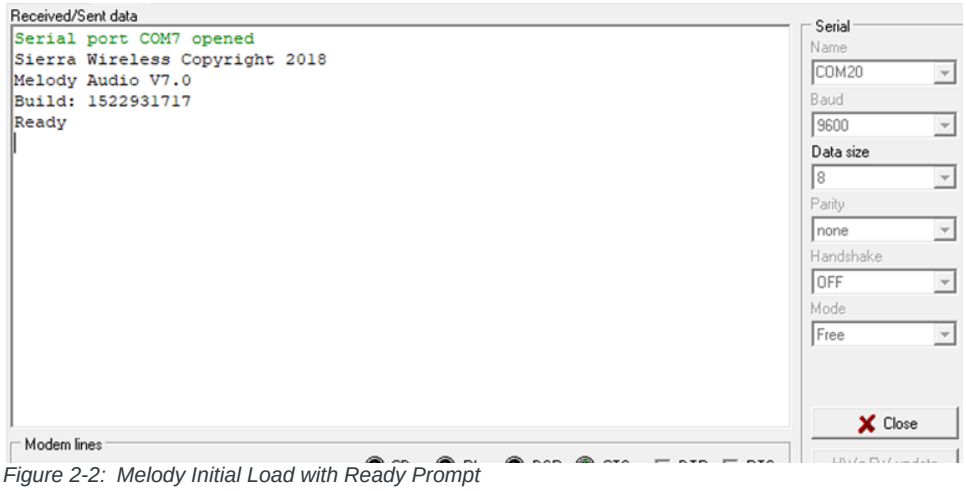


Figure 2-2: Melody Initial Load with Ready Prompt

Connect to the BC127 Module

By default the module is not in a connectable state.

To connect to the module:

1. Press the PAIRING button to enter pairing mode (make the module discoverable and connectable). The blue and green LEDs will start blinking.
2. On your smartphone, go to the Bluetooth settings application and scan for devices in the area. Your device should appear in the list with its default name—'BC-' and the last 6 characters of the Bluetooth address. (For example, "BC-04A3F1".)
3. Connect to the device. You can now start streaming music—the audio is automatically routed to the analog output (AUDIO OUT) of the BC127-DISKIT.

3: Device Firmware Upgrade

Melody supports firmware upgrade over the UART interface.

Sierra Wireless provides firmware files and the Melody Device Firmware Upgrade Tool on the BC127 device page on source.sierrawireless.com.

Sierra Wireless recommends upgrading to the latest version to benefit from new features and bug fixes.

Note: STANDARD, MFI, and BA releases are compatible with BC127 modules only. HD releases are compatible with BC127-HD modules only.

4: Melody API

General

Link ID Management

Melody uses Link IDs as an easy way for the user to interact with devices and profiles.

A Link ID is an 8-bit (two digit) hexadecimal value that represents a device ID (first digit) and a profile ID (second digit), as described in the tables below.

For example, Link ID 24 refers to a BLE connection on device 2:

- Device ID (2)—Device 2
- Profile ID (4)—BLE connection

Table 4-1: Device ID Values

Device ID	Description
1	Device 1
2	Device 2
3	Device 3
4	Device 4
5	Device 5

Table 4-2: Profile ID Values

Profile ID	Description
0	A2DP (Advanced Audio Distribution Profile)
1	AVRCP (Audio/Video Remote Control Profile)
2	AGHFP (HFP/HSP Audio Gateway)
3	HFP (HFP Hands-Free unit or HSP Headset)
4	BLE (Bluetooth Low Energy)
5	SPP (Serial Port Profile)
6	PBAP (Phone Book Access Profile)
7	HID (Human Interface Device Profile)
8	MAP (Manage Access Profile)
9	iAP (iPod Accessory Protocol)
A	TWS (A2DP True Wireless Stereo)

Operating Modes

Melody has two operating modes that define how data coming from the host (UART or USB) is processed.

Command Mode

Command mode is the default operation mode. Data received from the host is parsed as a command (see Commands).

Note: Commands always end with a carriage return (CR).

Data Mode

Data mode is used to transfer data between the host and a remote device without using commands.

To switch to data mode on an established connection, use the ENTER_DATA_MODE command with the BLE, SPP, or iAP link ID.

In data mode, data is transferred seamlessly over the specified link between the host and the remote device. The host interface is exclusively dedicated to transferring data—the only command Melody will parse is the escape sequence ($$$$) which switches Melody back to Command mode.

Additional notes:

- UART flow control should be enabled when using Data mode, to avoid any data loss.
- Melody returns automatically to Command mode if a disconnection occurs with the Data mode link.
- PIO 5 can be used to enter or exit Data mode (see GPIO Functionality).
- Remote commands can be sent (over BLE or SPP) to control the module when it is in Data mode.
- PIO 4 can be configured to automatically be raised upon reception of specific events (see GPIO Functionality).

Table 4-3: Data Mode Throughput by Connection Type

Connection Type	Throughput (kbps)
BLE (server to client, MTU=120)	100
BLE (client to server, MTU=120)	8
iAP (accessory to iOS device)	70
iAP High Speed (accessory to iOS device)	500
SPP	100
SPP High Speed	600

Switch Between Data/Command Modes

To switch between Data and Command modes, use the following commands:

Table 4-4: Operating Mode Commands

Command	Switch From
$$$$	Data Mode → Command Mode
ENTER_DATA_MODE <link_ID>	Command Mode → Data Mode

Commands

In Command mode (the default operation mode), the module accepts commands with the following format:

```
<command_name> <param_1> .. <param_n> [<option-
al_param>]<CR>
```

Command format details:

- Parameters are separated by spaces.
- Parameters between square brackets ('[...]') are optional.
- Commands must end with a carriage return (<CR>, \r, 0x0D).
- Maximum command length=150 characters. Error 0x0018 is returned if command is too long.

Commands are usually sent by the host over UART (or USB), but can also be sent remotely over BLE or SPP.

The following tables describe available Melody API commands:

- Table 4-5—Summary list of available commands linked to detailed command descriptions
- Table 4-6—Detailed descriptions for all available commands

Table 4-5: Melody Commands Summary

Command	Description	Page
$$$$	Exit Data Mode	18
ADVERTISING	Start/Stop/Configure BLE Advertising	18
AFH_MAP	Read/Set AFH Channel Classification	19
ASSOCIATION	Start/Stop Broadcast Audio Association	20
AT	Send AT Command/Response over HFP	23
AVRCP_META_DATA	Set/Request AVRCP Metadata	24
BATTERY_STATUS	Read Battery Status	25
BC_SMART_COMMAND	Send Remote Command Over BLE	25
BC_SMART_NOTIF	Enable/Disable BC Smart Notifications	26
BLE_GET_CHAR	GATT Characteristic Discovery	27

Table 4-5: Melody Commands Summary (Continued)

Command	Description	Page
BLE_GET_SERV	GATT Service Discovery	29
BLE_INDICATION	GATT Indication Request	30
BLE_NOTIFICATION	GATT Notification Request	31
BLE_READ	GATT Read Request	32
BLE_READ_RES	GATT Read Response	33
BLE_SECURITY	Start BLE Connection Security	34
BLE_SET_DB	Set Custom GATT Database	35
BLE_WRITE	GATT Write Request	36
BROADCAST	Start/Stop Broadcast Audio	37
BT_STATE	Set Bluetooth Classic State	38
CALL	Manage HFP/AGHFP Call Status	39
CLOSE	Send Disconnection Request	40
CONFIG	Get All Configurations	41
CVC_CFG	Read/Write cVc Configuration	42
DFU	Reboot in Device Firmware Update (DFU) mode	43
ENTER_DATA_MODE	Enter Data mode	43
GET	Get Single Configuration	44
HELP	List All Commands	44
HID_DESC	Set HID USB Descriptor	45
HID_READ	Read HID USB Descriptor	46
IAP	Get/Set iAP Identification Parameters	47
IAP_APP_REQ	iOS App Launch Request	48
INQUIRY	Search For Bluetooth Classic Devices	49
LICENSE	Read/Write aptX and cVc License Keys	51
LINK_POLICY	Set Link Policy Power Table	52
LIST	Show Paired Device List (PDL)	53
MAP_GET_MSG	Send request to retrieve message	54
MM_CFG	Read/Write Music Manager Configuration	56
MUSIC	AVRCP Music Playback Control	57
NAME	Get Remote Device Name	57
OPEN	Bluetooth Connection Request	58

Table 4-5: Melody Commands Summary (Continued)

Command	Description	Page
PAIR	Pairing Request	59
PASSKEY	Pairing User Confirmation	59
PB_ABORT	Abort Phonebook Download	60
PB_PULL	Download phonebook	61
PIO	Set GPIO State	63
POWER	Turn Bluetooth On/Off	64
REMOTE_VOLUME	Set Remote Hands-Free Unit Volume	64
RESET	Reset Module	65
RESTORE	Restore module to factory default configuration	65
ROLE	Read/change HCI role	66
ROUTE	Set/Get Audio Routing	67
RSSI	Get Signal Strength	68
SCAN	Search for BLE Devices	69
SEND	Send Data	70
SEND_RAW	Send Raw Data	71
SET	Set Single Configuration Value	72
SPEECH_REC	Activate/Deactivate Speech Recognition	72
SSRD	Set Scan Response Data	73
STATUS	Return Device Connection Status	74
TOGGLE_VR	Start/Stop Voice Recognition	76
TONE	Play Tone	77
TX_POWER	Get/Set Transmitter Power	79
UNPAIR	Unpair Devices	80
VERSION	Display Module Firmware Version and Bluetooth Address	80
VOLUME	Get/Set Volume	81
WRITE	Save Current Melody Configuration	82

Table 4-6: Melody Command Details

Command	Description
$$$$	**Exit Data Mode**
Details: Escape sequence to exit Data mode (return to Command mode). **Usage:** • Format: **$$$$** Response: OK Purpose: Escape sequence to exit Data mode (return to Command mode). **Parameters:** None **Notes:** A delay before and after the escape sequence is required (see CMD_TO configuration). **Example(s):** • Exit data mode # 20 x CMD_TO ms delay $$$$ # 20 x CMD_TO ms delay OK	
ADVERTISING	**Start/Stop/Configure BLE Advertising**
Details: Start/stop BLE advertising, or set the advertising data. When setting the advertising data, PENDING is received. Enter the raw data and OK is returned when the specified length of data has been sent. **Usage:** • Format: **ADVERTISING <action>** Response: OK or PENDING *(Enter <data>)*OK Purpose: Start/stop BLE advertising, or set the advertising data length and send the data. **Parameters:** <action> (BLE advertising action to take) • ON—Start BLE advertising • OFF—Stop BLE advertising • 1..31—Set size of the advertising data (max 31 bytes) <data> (Advertising data) • Raw data • e.g. {02}{01}{06}... indicates bytes with values 0x02, 0x01, 0x06, etc., are sent **Example(s):** • Start advertising ADVERTISING ON OK • Stop advertising ADVERTISING OFF OK • Set advertising data ADVERTISING 30 PENDING {02}{01}{06}{11}{06}{f0}{28}{e3}{68}{62}{d6}{34}{90}{51}{43}{ef}{aa}{c6}{4c}{2f}{bc}{08}{08}{42}{43}{30}{30}{31}{38}{31} OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
AFH_MAP	**Read/Set AFH Channel Classification**

Details: Read or set the AFH (Adaptive Frequency Hopping) map classifications for all 79 Bluetooth channels.

The AFH_MAP 'Set' format is used to initialize the AFH map (identifying all 79 Bluetooth channels) by classifying each channel as BAD (channels with interference) or UNKNOWN—the BC127 avoids BAD channels to create Asynchronous Connection-Less (ACL) connections.

Adaptive Frequency Hopping (AFH) is supported using internal HCI commands as described in the Bluetooth Core Protocol (version 5.0).

This command takes effect immediately, but does not persist after a reset.

Usage:

- Format (Read):
  AFH_MAP <conn_handle> <bd_addr>
  Response: AFH_MAP <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx>
  OK
  Purpose: Read the AFH channel map for a specific ACL connection.
- Format (Set):
  AFH_MAP <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx> <xx>
  Response: OK
  Purpose: Initialize the AFH channel map using specified classifications (BAD/UNKNOWN).

Parameters:

<conn_handle> (Connection handle of ACL connection)

- Two-digit hexadecimal format (e.g. B7)

<bd_addr> (Bluetooth address of remote device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

<xx> ... <xx> (AFH channel map)

- 10 byte channel map. Each byte represents 8 channels.
- Hexadecimal format (e.g. 00 00 00 00 00 1E F0 0F C0 3F)
- Bit values:
  - 0—BAD channel
  - 1—UNKNOWN channel
- First <xx> parameter is the MSB (most-significant byte)
- By default, channels 0–18 (2.402–2.420 MHz) are classified as BAD and will be put into a blacklist to avoid overlap with Wi-Fi.
- Channel 79 is reserved and must be set to BAD (bit set to 0)
- At least 20 channels must be set to UNKNOWN (bits set to 1)
- The first 40 channels are set to BAD. Only the last 39 channels are to be used.

Notes:Example(s):

- Read AFH map
  AFH_MAP 27 20FABB112233
  AFH_MAP 00 00 00 00 00 1E F0 0F C0 3F
  OK
- Set AFH map
  AFH_MAP 0 0 0 0 0 FF FF FF FF 7F
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
ASSOCIATION	**Start/Stop Broadcast Audio Association**

Details: (Note: This command applies to BA releases only.)
Start or stop association between a Broadcaster device and Receiver device(s). An association is a temporary GATT connection used to exchange security information; when the exchange is complete, the connection stops. If the association is successful, the Broadcaster and Receivers must start the Broadcast Audio feature (BROADCAST ON) to start transmitting/receiving audio.

A Broadcaster device uses ASSOCIATION ON to enable associations. While enabled, any Receiver devices can then associate to the Broadcaster. For each of these Receivers, a temporary BLE connection is automatically established and an ASSOCIATION_IN_PROGRESS notification is received.
The Broadcaster can disable the associations at any time using ASSOCIATION OFF. When disabled, no additional Receivers can associate to this Broadcaster, but Receivers that are already associated will continue to be able to receive audio.

Receiver devices use ASSOCIATION ON to start an association with a local Broadcaster device. A temporary BLE connection is automatically established and an ASSOCIATION_IN_PROGRESS notification is received, followed by an ASSOCIATION notification to indicate the result of the association with the Broadcaster.
While associating, the Receiver device can cancel the association using ASSOCIATION OFF.

Versions: Available for Melody 7.1 and later

Notes:
• The association between Broadcaster and Receiver(s) needs to be done only once.
• A Receiver can be associated with only one Broadcaster at a time. The association is required the first time and persists across Receiver power cycles, until a new association (with a different Broadcaster) is established.
• The address of the associated Broadcaster can be retrieved using the LIST command.
• Restoring the module with the RESTORE command removes the association.
• The association status can be retrieved from the STATUS command result.

Usage:

- Format (Broadcaster):
  ASSOCIATION <state>
  Response: OK
  (If <state>=ON, receive OPEN_OK, ASSOCIATION_IN_PROGRESS, BLE_PAIR_OK and CLOSE_OK notifications)
  Purpose: Enable or disable association with multiple receivers (ASSOCATION_IN_PROGRESS notifications appear only if association is enabled (i.e. ASSOCATION ON)).
- Format (Receiver):
  ASSOCIATION <state>
  Response (Melody 7.1):
  ASSOCIATION
  PENDING
  (Receive ASSOCIATION_IN_PROGRESS notification)
  (Receive ASSOCIATION notification)
  OK
  Response (Melody 7.2):
  OK
  (Receive OPEN_OK and ASSOCIATION_IN_PROGRESS notifications)
  (Receive BLE_PAIR_OK notification)
  ASSOCIATION <status> <bd_addr>
  (Receive CLOSE_OK notification)
  Purpose: Start or cancel association with a broadcaster.

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
ASSOCIATION (continued)	

Parameters:

<state> (Start/stop Broadcast Audio association)

- ON—Enable/Start association
- OFF—Disable/Cancel association

<status> (Association status)

- Applies to Melody 7.2 and later
- 0—Association successful
- 1—Security failed
- 2—Read characteristic failed

<bd_addr> (Bluetooth address of associated Broadcaster)

- Applies to Melody 7.2 and later
- 12-digit hexadecimal format (e.g. 20FABB000180)

Example(s):

- (Melody 7.1)
  Broadcaster (20FABB000160) enables association with multiple receivers; Receivers (20FABB000162 and 20FABB000127 shown in example) enable associations with Broadcaster:

Broadcaster	Receiver #1	Receiver #2
	ASSOCIATION ON	ASSOCIATION ON
ASSOCIATION ON	PENDING	PENDING
OK		
--- Receivers associating ---		
ASSOCIATION_IN_PROGRESS 20FABB000162	ASSOCIATION_IN_PROGRESS 20FABB000160	
	ASSOCIATION 0 0A02 0304	
ASSOCIATION_IN_PROGRESS 20FABB000127	OK	ASSOCIATION_IN_PROGRESS 20FABB000160
		ASSOCIATION 0 0A02 0304
		OK
...		
ASSOCIATION_IN_PROGRESS 20FABB000184		
ASSOCIATION OFF		
OK		

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
ASSOCIATION (continued)	

Example(s):

- (Melody 7.2 and later)
  Broadcaster (20FABB000180) enables association with multiple receivers; Receivers (20FABB000181 and 20FABB000182 shown in example) enable associations with Broadcaster:

Broadcaster	Receiver #1	Receiver #2
	ASSOCIATION ON	ASSOCIATION ON
ASSOCIATION ON	OK	OK
OK		
--- Receiver #1 associating ---		
OPEN_OK 14 BLE 583E24408AA9	OPEN_OK 14 BLE 6EF029B7D0A6	
ASSOCIATION_IN_PROGRESS 583E24408AA9	ASSOCIATION_IN_PROGRESS 6EF029B7D0A6	
BLE_PAIR_OK 583E24408AA9 20FABB000181	BLE_PAIR_OK 6EF029B7D0A6 20FABB000180	
	ASSOCIATION 0 20FABB000180	
CLOSE_OK 14 BLE 583E24408AA9	CLOSE_OK 14 BLE 6EF029B7D0A6	
--- Receiver #2 associating ---		
OPEN_OK 14 BLE 7840E4BAB0A9		OPEN_OK 14 BLE 6EF029B7D0A6
ASSOCIATION_IN_PROGRESS 7840E4BAB0A9		ASSOCIATION_IN_PROGRESS 6EF029B7D0A6
BLE_PAIR_OK 7840E4BAB0A9 20FABB000182		BLE_PAIR_OK 6EF029B7D0A6 20FABB000180
		ASSOCIATION 0 20FABB000180
CLOSE_OK 14 BLE 7840E4BAB0A9		CLOSE_OK 14 BLE 6EF029B7D0A6
---Broadcaster disables association ---		
ASSOCIATION OFF		
OK	*--- Receiver associated ---*	*--- Receiver associated ---*
	LIST	LIST
	BA 20FABB000180	BA 20FABB000180
	OK	OK

Table 4-6: Melody Command Details (Continued)

Command	Description
AT	**Send AT Command/Response over HFP**

Details: Send an AT command or response over a specified HFP/AGHFP link.

AT notifications are received for AT commands/responses.

Notes:
- HFP_CONFIG must be used to enable AT commands before AT can be used.
- AT command (HF→AG)—<CR> is automatically added after the command.
- AT response (AG→HF)—<CR><LF> is automatically added before and after the response.

Usage:

- Format: **AT <link_ID> <command>**
  Response: OK, CALL_REDIAL
  Purpose: Send an AT command over the specified HFP/AGHFP link (identified by <link_ID>).

Parameters::

<link_ID> (Link identifier (HFP or AGHFP))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<command> (AT command string)
- Valid AT command required (e.g. AT+VTS=1)

Example(s)::
- Send DTMF tones to device 1:
  AT 13 AT+VTS=1
  OK
  AT 13 AT+VTS=#
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
AVRCP_META_DATA	**Set/Request AVRCP Metadata**

Details: Set or request AVRCP metadata.

Usage:

- Format: **AVRCP_META_DATA <link_ID> [<type> <data>]**

  Response (Set):
  PENDING
  OK

  Response (Request):
  OK

  Purpose: Set a metadata value for the specified <link_ID> (all parameters required), or request all the metadata (use <link_ID> only).

Parameters:

<link_ID> (Link identifier (AVRCP))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<type> (Meta data type)

- 1—Title
- 2—Artist
- 3—Album
- 4—Number
- 5—Total number
- 6—Genre
- 7—Playing time

<data> (Meta data type)

- ASCII string

Example(s):

- Set AVRCP meta data for device 1:
  AVRCP_META_DATA 11 1 Get Lucky
  PENDING
  OK
- Request AVRCP meta data:
  AVRCP_META_DATA 11
  OK
  (AVRCP meta data received ...)
  AVRCP_MEDIA TITLE: Get Lucky
  AVRCP_MEDIA ARTIST: Daft Punk
  ...

Table 4-6: Melody Command Details (Continued)

Command	Description
BATTERY_STATUS	**Read Battery Status**

Details: Display battery status information.

Usage:

- Format: **BATTERY_STATUS**
  Response : <battery status notifications>
  ...
  OK
  Purpose: Display battery status details.

Parameters: None

Example(s):

- Display all available battery status details (this example shows charging in progress, voltage at 3154 mV):
  BATTERY_STATUS
  BATTERY_STATUS CHARGER IN PROGRESS
  BATTERY_STATUS VOLTAGE 3154
  OK

BC_SMART_COMMAND	**Send Remote Command Over BLE**

Details: Send a remote command over BLE to a connected device (GATT server).

Notes: This command is only allowed when the BC127 is connected to a device that supports the BC Smart profile—e.g. BC127, BC188, BX300x.

Usage:

- Format: **BC_SMART_COMMAND <link_ID> <command>**
  Response : OK
  Purpose: Send a remote command over BLE to a connected device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<command> (Remote command)

- ASCII string

Example(s):

- Send VERSION command to device 1 over BLE:
  BC_SMART_COMMAND 14 VERSION
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BC_SMART_NOTIF	**Enable/Disable BC Smart Notifications**

Details: Enable or disable BC Smart notifications on a connected device (GATT server)

Notes: This command is only allowed when the BC127 is connected to a device that supports the BC Smart profile—e.g. BC127, BC188, BX300x.

Usage:

- Format: **BC_SMART_NOTIF <link_ID> <enable_data> <enable_command_resp>**
  Response : OK
  Purpose: Enable or disable BC Smart notifications on a connected device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<enable_data> (Enable BC Smart Data notifications)

- ON—Enable notifications
- OFF—Disable notifications

<enable_command_resp> (Enable BC Smart Command response notifications)

- ON—Enable notifications
- OFF—Disable notifications

Example(s):

- Enable BC Smart Data notifications and BC Smart Command response notifications on device 1:
  BC_SMART_NOTIF 14 ON ON
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_GET_CHAR	**GATT Characteristic Discovery**

Details: Discover the characteristics of a remote GATT Server. A BLE_CHAR notification is received for each characteristic found.

Notes: To narrow the search, use the optional start and end handles.

Usage:

- Format: **BLE_GET_CHAR <link_ID> [<handle_start> <end_handle>]**

  Response (Melody 7.0/7.1):

  PENDING
  BLE_CHAR <link_ID> <type> <uuid> <handle>
  OK

  Response (Melody 7.2 and later):

  PENDING
  BLE_CHAR <link_ID> <type> <uuid> <handle> <properties>
  OK

  Purpose: Discover the characteristic of a remote GATT Server device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<start_handle> (Start discovery handle)
- 4-digit Hexadecimal format
- e.g. 2A05

<end_handle> (End discovery handle)
- 4-digit Hexadecimal format
- e.g. 2A05

<type> (UUID type)
- U16
- U32
- U128

<uuid> (Characteristic UUID)
- Hexadecimal format
- e.g. 2A05 (U16), 2A05-79AD (U32), AAD1E537-79AD-4A71-8FAA-373789F7D93C (U128)

<handle> (Characteristic handle)
- 4-digit Hexadecimal format
- e.g. 000C

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_GET_CHAR (continued)	

<properties> (Characteristic properties)
• Applies to Melody 7.2 and later
• 2-digit Hexadecimal format
• Bit values:
  • 0x01—Broadcast
  • 0x02—Read
  • 0x04—Write without response
  • 0x08—Write
  • 0x10—Notify
  • 0x20—Indicate
  • 0x40—Authenticated signed writes
  • 0x80—Extended properties
Example(s):
• Get the characteristics of device 1 (a GATT server):
  • Melody 7.0/7.1:
  BLE_GET_CHAR 14
  PENDING
  BLE_CHAR 14 U16 2A05 0003
  BLE_CHAR 14 U16 2A00 0007
  BLE_CHAR 14 U16 2A01 0009
  BLE_CHAR 14 U128 06D1E5E7-79AD-4A71-8FAA-373789F7D93C 000C
  OK
  • Melody 7.2 and later:
  BLE_GET_CHAR 14
  PENDING
  BLE_CHAR 14 U16 2A05 0003 20
  BLE_CHAR 14 U16 2A00 0007 02
  BLE_CHAR 14 U16 2A01 0009 02
  BLE_CHAR 14 U128 06D1E5E7-79AD-4A71-8FAA-373789F7D93C 000C 18
  OK
• Get the characteristics of device 1 (a GATT server), for handles in the range 0008-000B:
  • Melody 7.0/7.1:
  BLE_GET_CHAR 14 0008 000B
  PENDING
  BLE_CHAR 14 U16 2A01 0009
  OK
  • Melody 7.2 and later:
  BLE_GET_CHAR 14 0008 000B
  PENDING
  BLE_CHAR 14 U16 2A01 0009 02
  OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_GET_SERV	**GATT Service Discovery**

Details: Discover the services of a remote GATT Server. A BLE_SERV notification is received for each service found.

Usage:

- Format: **BLE_GET_SERV <link_ID>**

  Response (Melody 7.0/7.1):

  PENDING
  BLE_SERV <link_ID> <type> <uuid> <handle>
  OK

  Response (Melody 7.2 and later):

  PENDING
  BLE_SERV <link_ID> <type> <uuid> <start_handle> <end_handle>
  OK

  Purpose: Discover the services of a remote GATT Server device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<type> (UUID type)
- U16
- U32
- U128

<uuid> (Service UUID)
- Hexadecimal format
- e.g. 1801 (U16), 1801-79AD (U32), 002F4CC6-AAEF-4351-9034-D66268E328F0 (U128)

<handle> (Service handle)
- Applies to Melody 7.0/7.1 only
- 4-digit Hexadecimal format
- e.g. 000A

<start_handle> (Service start handle)
- Applies to Melody 7.2 and later
- 4-digit Hexadecimal format
- e.g. 000A

<end_handle> (Service end handle)
- Applies to Melody 7.2 and later
- 4-digit Hexadecimal format
- e.g. 000A

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_GET_SERV (continued)	

Example(s):
• Get the services of device 1 (a GATT server):
  • Melody 7.0/7.1:
  BLE_GET_SERV 14
  PENDING
  BLE_SERV 14 U16 1801 0001
  BLE_SERV 14 U16 1800 0006
  BLE_SERV 14 U128 002F4CC6-AAEF-4351-9034-D66268E328F0 000A
  OK
  • Melody 7.2 and later:
  BLE_GET_SERV 14
  PENDING
  BLE_SERV 14 U16 1801 0001 0005
  BLE_SERV 14 U16 1800 0006 0009
  BLE_SERV 14 U128 002F4CC6-AAEF-4351-9034-D66268E328F0 000A FFFF
  OK	

BLE_INDICATION	**GATT Indication Request**
Details: Send GATT indication request over BLE. When PENDING response is received, enter the characteristic <data> and then the BC127 sends the <data> as a GATT indication and returns OK. **Usage:** • Format: **BLE_INDICATION <link_ID> <handle> <size>** Response : PENDING *(Enter <data>)*OK Purpose: Send GATT indication request to specified link, then indicate the characteristic <data>. “OK” is returned when the expected number of bytes (<size>) has been received. **Parameters:** <link_ID> (Link identifier (BLE)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details. <handle> (Characteristic handle) • 4-digit Hexadecimal format • e.g. 000A <size> (Size of raw data, in bytes) • Max size=max(MTU) - 3 <data> (Characteristic data) • Raw data • e.g. {31}{32}{33}{34}{35} indicates bytes with values 0x31–0x35 (“12345”) are sent **Example(s):** • Send indication request and then indicate characteristic value “12345”: BLE_INDICATION 14 000A 5 PENDING {31}{32}{33}{34}{35}OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_NOTIFICATION	**GATT Notification Request**

Details: Send GATT notification request over BLE. When PENDING response is received, enter the characteristic <data> and then the BC127 sends the <data> as a GATT notification and returns OK.

Usage:

- Format: **BLE_NOTIFICATION <link_ID> <handle> <size>**
  Response : PENDING
  *(Enter <data>)*OK
  Purpose: Send GATT notification request to specified link, then notify the characteristic <data>. "OK" is returned when the expected number of bytes (<size>) has been received.

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<handle> (Characteristic handle)

- 4-digit Hexadecimal format
- e.g. 000A

<size> (Size of raw data, in bytes)

- Max size=max(MTU) - 3

<data> (Characteristic data)

- Raw data
- e.g. {31}{32}{33}{34}{35} indicates bytes with values 0x31–0x35 ("12345") are sent

Example(s):

- Send notification request and then notify characteristic value "12345":
  BLE_NOTIFICATION 14 000A 5
  PENDING
  {31}{32}{33}{34}{35}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_READ	**GATT Read Request**

Details: Get (read) the value of a specific characteristic of a remote GATT server device.

Usage:

- Format: **BLE_READ <link_ID> <handle>**
  Response : PENDING
  BLE_READ_RES <link_ID> <handle> <size> <data>
  OK
  Purpose: Read the specified characteristic <handle> value of a remote GATT Server device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<handle> (Characteristic handle)

- 4-digit Hexadecimal format
- e.g. 000A

<size> (Size of characteristic value, in bytes)

- Max size=max(MTU) - 3

<data> (Characteristic value)

- Hexadecimal format ASCII string
- e.g. “3132333435”=0x31 0x33 0x33 0x34 0x35=“12345”

Example(s):

- Read the value of characteristic 000A on remote GATT Server (identified by <link_ID>):
  BLE_READ 14 000A
  PENDING
  BLE_READ_RES 14 000A 5 3132333435
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_READ_RES	**GATT Read Response**

Details: Send a requested characteristic value to a remote GATT device, in response to a BLE_READ notification.

Usage:

- Format: **BLE_READ_RES <link_ID> <handle> <size>**
  Response : PENDING
  *(Send <size> bytes of <data>)*OK
  Purpose: Send data to the specified link in response to a BLE_READ notification. After “PENDING” is received, send <size> bytes of response data.

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<handle> (Characteristic handle)

- 4-digit Hexadecimal format
- e.g. 000A

<size> (Size of characteristic value, in bytes)

- Max size=max(MTU) - 3

<data> (Raw response data)

- ASCII
- e.g. {31}{32}{33}{34}{35} indicates bytes with values 0x31–0x35 (“12345”) are sent

Example(s):

- Send 5 bytes of data in response to a BLE_READ request from device 1:
  BLE_READ_RES 14 000A 5
  PENDING
  {31}{32}{33}{34}{35}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_SECURITY	**Start BLE Connection Security**

Details: Start security on a BLE connection with a remote GATT device.

Notes: If the devices are not paired, the authentication procedure takes place based on the device I/O capabilities (see SSP_CAPS configuration).

Usage:

- Format: **BLE_SECURITY <link_ID>**
  Response : PENDING
  BLE_SECURITY <link_ID> <status>
  Purpose: Start security on the connection to a remote GATT device (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<status> (Result of security request)

- 0—Success
- 1—Failure

Example(s):

- Successfully start security on connection to remote GATT device (identified by <link_ID>):
  BLE_SECURITY 14
  PENDING
  BLE_PAIR_OK 49E959ACF211 20FABB000151 *(BLE_PAIR_OK is received only if devices are not already paired.)*
  BLE_SECURITY 14 0
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_SET_DB	**Set Custom GATT Database**

Details: Set up a custom GATT database by specifying the database size, then sending the data.
This command returns a PENDING response. Enter data until the specified number of bytes has been sent, and an OK response will be received. (The data being entered can be entered in batches separated by carriage returns (<CR>). PENDING responses will be received after each carriage return until all the data has been sent as shown in the example below.

Notes:
- Database is not persistent after a reboot.
- To generate the values for a custom database, use the BC127 Melody BLE database generation tool (db_gen_tool) available at source.sierrawireless.com.

Usage:

- Format: **BLE_SET_DB <size>**
  Response : PENDING
  (Enter <data>)
  PENDING
  (Enter <data>)
  ...
  OK
  Purpose: Set up a custom GATT database.
  "OK" is returned when the expected number of bytes (<size>) has been received.

Parameters:

<size> (Database size, in words)
- Hexadecimal value
- e.g. 44=68 words

<data> (Database data)
- Hexadecimal values
- e.g. 0002 0118 3005 2003 0005 2A00 D000 6400 0002 0018

Example(s):
- Set up database, size=68 words (sent in six 10-word batches and a final 6-word batch, each separated by <CR>):
  BLE_SET_DB 44
  PENDING
  0002 0118 3005 2003 0005 2A00 D000 6400 0002 0018
  PENDING
  3005 0207 0000 2A00 D400 3005 0209 0001 2A00 D400
  PENDING
  0010 F028 E368 62D6 3490 5143 EFAA C64C 2F00 3013
  PENDING
  180C 003C D9F7 8937 37AA 8F71 4AAD 79E7 E5D1 AA00
  PENDING
  CC01 0000 3053 180E 003C D9F7 8937 37AA 8F71 4AAD
  PENDING
  79E7 E5D1 BB00 CD01 0000 3093 1810 003C D9F7 8937
  PENDING
  37AA 8F71 4AAD 79E7 E5D1 CC00 CE01 0000
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BLE_WRITE	**GATT Write Request**

Details: Write (set) the value of a specific characteristic of a remote GATT server device.

Usage:

- Format: **BLE_WRITE <link_ID> <handle> <size>**
  Response : PENDING
  *(Enter <data>)*OK
  Purpose: Write data (<size> bytes) to the specified characteristic <handle> of a remote GATT Server device (identified by <link_ID>).
  "OK" is returned when the expected number of bytes (<size>) has been received.

Parameters:

<link_ID> (Link identifier (BLE))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<handle> (Characteristic handle)

- 4-digit Hexadecimal format
- e.g. 000A

<size> (Size of characteristic value, in bytes)

- Max size=max(MTU) - 3

<data> (Raw data)

- ASCII
- e.g. {31}{32}{33}{34}{35} indicates bytes with values 0x31–0x35 ("12345") are sent

Example(s):

- Write a 5-byte value for characteristic 000A on remote GATT Server (identified by <link_ID>):
  BLE_READ 14 000A 5
  PENDING
  {31}{32}{33}{34}{35}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
BROADCAST	**Start/Stop Broadcast Audio**

Details: (Note: This command applies to BA releases only.)
Start or stop Broadcast Audio.
When Broadcast Audio is started for a Broadcaster, the audio from an A2DP Source device (e.g. a smartphone) or from the analog input (ROUTE 1) is automatically broadcasted.

Versions: Available for Melody 7.1 and later

Notes:
- A receiver must be associated with a broadcaster before it can receive audio.
- The module can be configured to automatically start broadcasting/receiving when it powers on using the BA_CONFIG configuration.
- The broadcast status can be retrieved from the STATUS command result.

Usage:

- Format: **BROADCAST <state>**
  Response : OK
  (Broadcaster receives BA_BROADCASTER_START or BA_BROADCASTER_STOP notification, or Receiver receives BA_RECEIVER_START or BA_RECEIVER_STOP notification.)
  Purpose: Broadcaster uses command to start or stop broadcasting audio; receiver uses command to start or stop receiving broadcast audio.

Parameters:

<state> (Broadcast Audio state)
- ON—Start Broadcast Audio
- OFF—Stop Broadcast Audio

Example(s):

- Broadcaster starts/stops broadcasting audio:
  A2DP_STREAM_START 10
  AVRCP_PLAY 11
  BROADCAST ON
  OK
  BA_BROADCASTER_START *(Receivers will now start receiving audio)*
  BROADCAST OFF
  OK
  BA_BROADCASTER_STOP
- Receiver starts/stops receiving broadcast audio:
  BROADCAST ON *(Receiver ready to receive audio)*
  OK
  BA_RECEIVER_START *(Receiver starts receiving audio)*
  BROADCAST OFF *(Receiver stops receiving audio)*
  OK
  BA_RECEIVER_STOP

Table 4-6: Melody Command Details (Continued)

Command	Description
BT_STATE	**Set Bluetooth Classic State**

Details: Set the device's connectable and discoverable states. To set the state automatically at power-on, see BT_STATE_CONFIG.

Notes: To display the current states, see the STATUS command and LED Indications.

Usage:

- Format: **BT_STATE <connectable> <discoverable>**
  Response : OK
  Purpose: Configure the devices connectable and discoverable states.

Parameters:

<connectable> (Device's connectable state)

- ON—Connectable
- OFF—Not connectable

<discoverable> (Device's discoverable state)

- ON—Discoverable
- OFF—Not discoverable

Example(s):

- Make the device connectable and discoverable:
  BT_STATE ON ON
  OK
- Make the device connectable but not discoverable:
  BT_STATE ON OFF
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
CALL	**Manage HFP/AGHFP Call Status**

Details: Manage an HFP/AGHFP call (e.g. make a call, answer or reject an incoming call, end a call, etc.).

Notes: Call notifications are received whenever a call state changes. Notifications include CALL_ACTIVE, CALL_DIAL, CALL_END, CALL_INCOMING, CALL_MEMORY, CALL_OUTGOING, and CALL_REDIAL.

Usage:

- Format: **CALL <link_ID> <action> [<value>]**
  Response : OK
  Purpose: Manage the call status of the specified HFP/AGHFP link (identified by <link_ID>).

Parameters:

<link_ID> (Link identifier (HFP or AGHFP))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<action> (Call action)

- HFP link:
  - REDIAL—Call “redial last number” request
  - MEMORY—Call “memory” request; <value> parameter required.
  - OUTGOING—Call “outgoing” request; <value> parameter required.
  - ANSWER—Accept an incoming call.
  - REJECT—Reject an incoming call.
  - TWC—Three-way calling; <value> parameter required.
  - TRANSFER—Perform an audio transfer. <value> parameter (optional) can be used to specify the direction.
  - END—Terminate a call.
- AGHFP link:
  - ANSWER—Notify the call has been answered
  - TRANSFER—Audio connection transfer
  - INCOMING—Notify an incoming call. <value> parameter required.
  - OUTGOING—Notify an outgoing call. <value> parameter required.
  - TRANSFER—Perform an audio transfer. <value> parameter (optional) can be used to specify the direction.
  - END—Terminate a call.

<value> (<action>-dependent value)

- For <action>=MEMORY—Required. Combination of alphanumeric characters
- For <action>=OUTGOING—Required. Number (any combination of alphanumeric characters)
- For <action>=TWC—Required:
  - 0—Reject
  - 1—Hold & Accept/Swap
  - 2—End & Accept
  - 3—Merge calls (multiparty)
  - 4—Hang up calls (multiparty)
- For <action>=TRANSFER—Optional:
  - ON_HF—Transfer to HF
  - ON_AG—Transfer to AG

Example(s):

- See HFP Functionality on page 166

Table 4-6: Melody Command Details (Continued)

Command	Description
CLOSE	**Send Disconnection Request**

Details: Send a disconnection request to a link, a specific device (and all its profiles), or all devices (and all their profiles).

Notes: CLOSE_OK notifications are received when each disconnection occurs.

Usage:

- Format: **CLOSE < <link_ID> | <device_ID> | ALL >**
  Response : OK
  Purpose: Issue a request to disconnect from a specific link, a specific device (and all its profiles), or all devices (and all their profiles).

Parameters:

<link_ID> (Link identifier)

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<device_ID> (Device identifier)

- 4-bit (one digit) hexadecimal value

Example(s):

- Disconnect link 14
  CLOSE 14
  OK
  CLOSE_OK 14 BLE 20FABB000150
- Disconnect device 1 and all its profiles
  CLOSE 1
  OK
  CLOSE_OK 10 A2DP 20FABB000150
  CLOSE_OK 11 AVRCP 20FABB000150
  CLOSE_OK 14 BLE 20FABB000150
- Disconnect all devices and profiles
  CLOSE ALL
  OK
  CLOSE_OK 10 A2DP 20FABB000150
  CLOSE_OK 11 AVRCP 20FABB000150
  CLOSE_OK 14 BLE 20FABB000150
  CLOSE_OK 20 A2DP 20FABB000167
  CLOSE_OK 21 AVRCP 20FABB000167
  CLOSE_OK 23 BLE 20FABB000167

Table 4-6: Melody Command Details (Continued)

Command	Description
CONFIG	**Get All Configurations**

Details: Display all Melody configurations.

Notes:
- For configuration details, refer to Configurations.
- To get a single configuration, see GET.

Usage:

- Format: **CONFIG**
  Response : <config_name>=<config_value>
  ...
  OK
  Purpose: Retrieve all Melody configurations.

Parameters:

<config_name> (Configuration name)
- ASCII string
- See Configurations for configuration names

<config_value> (Configuration value)
- ASCII string
- See Configurations for configuration formats

Example(s):
- Show all configurations:
  CONFIG
  AUDIO=0 0
  AUDIO_ANALOG=15 15 1 OFF
  ...
  VREG_ROLE=1
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
CVC_CFG	**Read/Write cVc Configuration**

Details: Read or write cVc (Clear Voice Capture) configuration values.

Usage:

- Format: **CVC_CFG <type> [<key> <length>]**

  Response (Read):

  CVC_CFG <type> <config_value>
  ...
  OK

  Response (Write):

  PENDING
  (Data entered)
  OK

  Purpose: Read or write cVc configuration values.

Parameters:

<type> (Configuration type)
- NB—Narrow band configuration
- WB—Wide band configuration

<key> (Key index)
- Valid range: 0–3

<length> (Key length, in words)
- Valid range: 0–64

<config_value> (Configuration value)
- ASCII string
- e.g. 0000 0001 0002 0003 0004

Example(s):

- Write cVc WB configuration

  CVC_CFG WB 0 5
  PENDING
  0000 0001 0002 0003 0004
  OK

- Read cVc WB configuration

  CVC_CFG WB
  CVC_CFG WB 0 0000 0001 0002 0003 0004
  CVC_CFG WB 1
  CVC_CFG WB 2
  CVC_CFG WB 3
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
DFU	**Reboot in Device Firmware Update (DFU) mode**
Details: Reboot device in DFU (device firmware update) mode. **Usage:** • Format: **DFU** Response: none Purpose: Reboot device to receive firmware update. **Parameters:** none **Example(s):** • Reboot device to prepare to receive firmware update: DFU	
ENTER_DATA_MODE	**Enter Data mode**
Details: Enters Data mode on a BLE, SPP, or iAP link. (To exit data mode, enter the escape sequence $$$$.) **Notes:** For iAP links, this command is allowed only if an EA (External Accessory) session is already opened. (IAP_OPEN_SESSION and IAP_CLOSE_SESSION notifications indicate when the EA session opens/closes.) The IAP_OPEN_SESSION notification must be received before this command can be used. **Usage:** • Format: **ENTER_DATA_MODE <link_ID>** Response : OK Purpose: Enter Data mode. **Parameters:** <link_ID> (Link identifier (BLE, SPP, or iAP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details. **Example(s):** • Enter Data mode on device 1 (a BLE link): ENTER_DATA_MODE 14 OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
GET	**Get Single Configuration**
Details: Get a specific Melody configuration. (e.g. Get the AUDIO configuration, which has two parameters associated with it.) **Notes:** To get all configurations, see CONFIG. **Usage:** • Format: **GET <config_name>** Response : <config_name>=<config_value> OK Purpose: Retrieve a specific Melody configuration. **Parameters:** <config_name> (Configuration name) • ASCII string • See Configurations for configuration types <config_value> (Configuration value) • ASCII string • See Configurations for configuration formats **Example(s):** • Get the AUDIO configuration: GET AUDIO AUDIO=0 0 OK	
HELP	**List All Commands**
Details: List all Melody audio commands in alphabetical order. **Usage:** • Format: **HELP** Response: <command> ... OK Purpose: List all available Melody audio commands. **Parameters:** none **Example(s):** • List all commands: HELP ADVERTISING AT AVRCP_META_DATA ... VOLUME WRITE OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
HID_DESC	**Set HID USB Descriptor**

Details: Set the USB descriptor to use when acting as a HID device.
This command returns a PENDING response. Enter the descriptor data until the specified number of bytes has been sent, and an OK response will be received.

Notes:
- By default, a USB descriptor for a simple keyboard is loaded.
- When using this command, any previously loaded USB descriptor is deleted.
- The USB descriptor loaded with this command is not persistent across a reset.

Usage:

- Format: **HID_DESC <size>**
  Response: PENDING
  *(Enter <data>)*OK
  Purpose: Set the USB descriptor to <data>.

Parameters:

<size> (USB descriptor size, in bytes)
- Integer value
- Valid range: 1–255

<data> (Descriptor data)
- Raw data
- e.g. {05}{01}{09}... indicates bytes with values 0x05, 0x01, 0x09, etc., are sent

Example(s):
- Set the USB descriptor (length = 71 bytes):
  HID_DESC 71
  PENDING
  {05}{01}{09}{02}{A1}{01}{09}{01}{A1}{00}{05}{09}{19}{01}{29}{08}{15}{00}{25}{01}{95}{08}{75}{01}{81}{02}{95}{00}{81}{03}{05}{01}{09}{30}{09}{31}{16}{01}{F8}{26}{FF}{07}{75}{0C}{95}{02}{81}{06}{09}{38}{15}{81}{25}{7F}{75}{08}{95}{01}{81}{06}{05}{0C}{0A}{38}{02}{95}{01}{81}{06}{C0}{C0}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
HID_READ	**Read HID USB Descriptor**

Details: Read the USB descriptor of a remote HID device from the remote device's SDP record.

Usage:

- Format: **HID_READ <bd_addr>**
  Response: PENDING
  HID_READ <size> <data>
  OK
  Purpose: Read the HID USB descriptor from a remote device at specified Bluetooth address.

Parameters:

<bd_addr> (Bluetooth address of remote HID device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

<size> (HID USB descriptor size, in bytes)

- Integer value
- Range is remote device-dependent

<data> (HID USB descriptor)

- Raw data
- e.g. {09}{02}{06}... indicates bytes with values 0x09, 0x02, 0x06, etc., are received

Example(s):

- Read the HID USB descriptor from remote HID device at specified Bluetooth address:
  HID_READ 20FABB000155
  PENDING
  HID_READ 335
  {09}{02}{06}{36}{01}{49}{36}{01}{46}{08}{22}{26}{01}{41}{05}{01}{09}{06}{A1}{01}{85}{01}{75}{01}{95}{08}{05}{07}{19}{E0}{29}{E7}{15}{00}{25}{01}{81}{02}{95}{01}{75}{08}{81}{03}{95}{05}{75}{01}{05}{08}{19}{01}{29}{05}{91}{02}{95}{01}{75}{03}{91}{03}{95}{06}{75}{08}{15}{00}{26}{FF}{00}{05}{07}{19}{00}{29}{FF}{81}{00}{C0}{05}{0C}{09}{01}{A1}{01}{85}{02}{15}{00}{25}{01}{75}{01}{95}{12}{0A}{23}{02}{0A}{AE}{01}{0A}{B1}{01}{0A}{94}{01}{0A}{8A}{01}{0A}{82}{01}{0A}{21}{02}{09}{B6}{09}{CD}{09}{B5}{09}{E2}{09}{EA}{09}{E9}{09}{30}{09}{40}{0A}{96}{01}{0A}{9D}{01}{0A}{C1}{01}{81}{02}{95}{01}{75}{06}{81}{03}{C0}{05}{0C}{09}{01}{A1}{01}{85}{03}{05}{01}{09}{06}{A1}{02}{05}{06}{09}{20}{15}{00}{26}{FF}{00}{75}{08}{95}{01}{81}{02}{C0}{C0}{05}{01}{09}{80}{A1}{01}{85}{04}{15}{00}{25}{01}{75}{01}{95}{01}{09}{82}{81}{02}{95}{01}{75}{07}{81}{03}{C0}{05}{0C}{09}{01}{A1}{01}{85}{05}{05}{01}{09}{06}{A1}{02}{06}{00}{FF}{25}{01}{75}{01}{95}{02}{0A}{03}{FE}{0A}{04}{FE}{81}{02}{95}{06}{81}{03}{C0}{C0}{05}{01}{09}{02}{A1}{01}{85}{08}{09}{01}{A1}{00}{05}{09}{19}{01}{29}{05}{15}{00}{25}{01}{75}{01}{95}{05}{81}{02}{75}{03}{95}{01}{81}{01}{05}{01}{09}{30}{09}{31}{15}{81}{25}{7F}{75}{08}{95}{02}{81}{06}{09}{38}{15}{81}{25}{7F}{75}{08}{95}{01}{81}{06}{C0}{C0}{05}{0C}{09}{01}{A1}{01}{85}{FF}{05}{06}{95}{01}{75}{02}{19}{24}{29}{26}{81}{02}{75}{06}{81}{01}{C0}{4F}{4B}{0A}
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
IAP	**Get/Set iAP Identification Parameters**

Details: Get or set iAP identification parameters.
After setting the parameters, use WRITE to save the new iAP configuration, and then reboot the device for the parameters to take effect.

Usage:

- Format: **IAP [<parameter>[=<value>]]**

  Response (Set):
  OK

  Response (Get):
  <parameter>=<value>
  ...
  OK

  Purpose: Get a specified iAP identification parameter or all identification parameters, or set a specific identification parameter.

Parameters:

<parameter> (iAP identification parameter)

- ACCESSORY_NAME—Name of accessory. Max length=40 characters
- MANUFACTURER_NAME—Name of manufacturer. Max length=20 characters
- MODEL_NAME—Name of model. Max length=16 characters
- SERIAL_NO—Serial number. Max length=20 characters
- SEED_ID—Seed ID. Max length=12 characters
- PROTOCOL_STRING—Default protocol string. Max length=36 characters
- HARDWARE_VER—Hardware version. Max length=3 characters
- FIRMWARE_VER—Firmware version. Max length=3 characters

<value> (Value of <parameter>)

- ASCII string.
- Max length—see <parameter> descriptions

Example(s):

- Get all iAP identification parameters
  IAP
  ACCESSORY_NAME=BC127
  MANUFACTURER_NAME=Sierra Wireless
  MODEL_NAME=Melody 7
  SERIAL_NO=0000000000000000
  SEED_ID=1234567890
  PROTOCOL_STRING=com.sierrawireless.Melody
  HARDWARE_VER=3 2 1
  FIRMWARE_VER=1 2 3
  OK
- Set iAP model name
  IAP MODEL_NAME=My Model Name
  OK
- Get iAP model name
  IAP MODEL_NAME
  MODEL_NAME=Melody 7
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
IAP_APP_REQ	**iOS App Launch Request**

Details: Send a request to the iOS device to open the app specified by the protocol (optional).

Notes: If the protocol is not specified, the protocol in the iAP configuration will be used instead.

Usage:

- Format: **IAP_APP_REQ [<protocol>]**
  Response: OK
  Purpose: Send request to iOS device

Parameters:

<protocol> (Protocol string)

- ASCII string

Example(s):

- Send request to iOS device to open the app indicated in the iAP configuration:
  IAP_APP_REQ
  OK
- Send request to iOS device to open the app specified by the indicated protocol (Note: com.sierrawireless.Melody is the default protocol string.):
  IAP_APP_REQ com.sierrawireless.Melody
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
INQUIRY	**Search For Bluetooth Classic Devices**

Details: Search for all (or a subset of all) Bluetooth Classic devices in the area.

Notes:
- The same device may appear multiple times.
- To scan for BLE devices, use SCAN.

Usage:

- Format: **INQUIRY [<timeout> [<filter_type> <filter_data> [<max_results>]]]**

  Response :
  PENDING
  INQUIRY <bd_addr> “<name>” <cod> <rssi>
  ...
  INQUIRY <bd_addr> “<name>” <cod> <rssi>
  INQU_OK

  Purpose: Search for Bluetooth Classic devices—Use no parameters (to search for all) or optional parameters to narrow the search.

Parameters:

<timeout> (Search time limit, in 1.28 sec increments)
- Valid range: 1–48 (1.28 sec–61.44 sec)
- Default: 5 (6.4 sec)

<filter_type> (Filter type to narrow search results)
- 0—No filter; <filter_data> is ignored (a junk value must be entered for <filter_data>)
- 1—Filter by name (<filter_data> format—string). Display result if name starts with <filter_data>.
- 2—Filter by Class of Device (COD); <filter_data> format—valid COD. Display result if COD matches.
- 3—Filter by RSSI (<filter_data> format—minimum signal strength in dBm). Display result if signal strength is at least the minimum.
- 4—TWS devices only (<filter_data> format— ON or OFF). Display only TWS devices.

<filter_data> (Filter type-dependent value)
- See <filter_type> option descriptions for <filter_type> value format.

<max_results> (Maximum number of results to display)
- Integer value
- 0—(Default) Unlimited
- Valid range: 1–255—Max number to display

<bd_addr> (Device Bluetooth address)
- 12-digit hexadecimal format (e.g. 20FABB112233)

<name> (Device name)
- ASCII string

<cod> (Device class)
- Hexadecimal value
- Class of Device (per Bluetooth specification)

<rssi> (Signal strength, in dBm)
- ASCII string
- e.g. “-49dBm”

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
INQUIRY (continued)	
Example(s): *Note: Melody 7.0/7.1 INQUIRY notifications incorrectly indicate signal strength units as "dB" instead of "dBm".* • Search for devices, default parameters, no filter: INQUIRY PENDING INQUIRY 20FABB200D46 “BM31xx-200D46” 2C0424 -75dB ... INQUIRY 20FABB000150 “BC-000150” 2C0404 -49dBm INQU_OK • Search for devices, 12.8 sec timeout, filter by name (“BC”): INQUIRY 10 1 BC PENDING INQUIRY 20FABB000150 “BC-000150” 240404 -49dBm INQU_OK • Search for devices, 12.8 sec timeout, filter by RSSI (min -70dBm): INQUIRY 10 3 70 PENDING INQUIRY 20FABB200D46 “BM31xx-200D46” 2C0414 -66dBm ... INQUIRY 20FABB000150 “BC-000150” 240404 -49dBm INQU_OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
LICENSE	**Read/Write aptX and cVc License Keys**

Details: Read or write aptx and cVc license keys.

Usage:

- Format: **LICENSE [<type>[=<key>]]**

  Response (Read):

  <type>=<key>
  [<type>=<key>]
  OK

  Response (Write):

  OK

  Purpose: Read one or both keys, or write the aptX or cVc license key.

Parameters:

<type> (License type)

- CVC—cVc license key
- APTX—aptX licnse key

<key> (License key)

- 5 words
- e.g. 0001 0002 0003 0004 0005

Example(s):

- Write aptX license key:
  LICENSE APTX=0001 0002 0003 0004 0005
  OK
- Read aptX license key:
  LICENSE APTX
  APTX=0001 0002 0003 0004 0005
  OK
- Read both license keys:
  LICENSE
  APTX=0001 0002 0003 0004 0005
  CVC=0000 0000 0000 0000 0000
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
LINK_POLICY	**Set Link Policy Power Table**

Details: Set the link policy power table for a specific connection. The link policy power table allows the device to switch between different power modes.
By default, Melody uses its own default values. This command is used to temporarily overwrite the default values—the change does not persist across device reset.
The command returns PENDING after each entry is received; each entry is separated by <CR>.

Usage:

- Format: **LINK_POLICY <device_ID> <nb_entries>**
  Response: PENDING
  (Enter data: <state> <min_interval> <max_interval> <attempt> <timeout> <time>)
  PENDING
  (Enter data: <state> <min_interval> <max_interval> <attempt> <timeout> <time>)
  ...
  OK
  Purpose: Set the entries for the link policy power table for a specific connection (<device_id>).
  Note: After each PENDING response, enter the next table entry followed by a <CR>.

Parameters:

<device_ID> (Device identifier)
- 4-bit (one digit) hexadecimal value

<nb_entries> (Number of entries in power table)
- 1–8

<state> (Power mode)
- 00—Active mode
- 01—Sniff mode
- FF—Passive mode

<min_interval> (Sniff minimum interval, in 0.625 sec increments)
- Hexadecimal value
- Values (Sniff mode): (Even values only) 0x0002–0xFFFE (1.25–40958.75 sec)
- Value (Active/Passive mode): 0

<max_interval> (Sniff maximum interval, in 0.625 sec increments)
- Hexadecimal value
- Values (Sniff mode): (Even values only) 0x0002–0xFFFE (1.25–40958.75 sec)
- Value (Active/Passive mode): 0

<attempt> (Number of baseband receive slots for sniff attempt, in 1.25 ms increments)
- Hexadecimal value
- Values (Sniff mode): 0x0001–0x7FFF (1.25 ms–40.96 sec)
- Value (Active/Passive mode): 0

<timeout> (Number of baseband receive slots for sniff timeout, in 1.25 ms increments)
- Hexadecimal value
- Values (Sniff mode): 0x0001–0x7FFF (1.25 ms–40.96 sec)
- Value (Active/Passive mode): 0

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
LINK_POLICY (continued)	
<time> (Time spent in <state>, in seconds) • Hexadecimal value • Value must be 0 for last entry in the power table **Example(s):** • Populate the power table for device 1 with two entries (30 second Passive mode, and 500 ms Sniff mode): LINK_POLICY 1 2 PENDING FF 0 0 0 0 1E<CR> #Passive mode (30 seconds) PENDING 1 320 320 2 1 0<CR> #Enter sniff mode (500 ms) OK	
LIST	**Show Paired Device List (PDL)**
Details: Display all entries in the paired device list. **Notes:** To remove devices from the paired device list, use the UNPAIR command. **Usage:** • Format: **LIST** Response: LIST <bd_addr> [<profile_1> ... <profile_n>] ... OK Purpose: Display the entire paired device list. **Parameters:** <bd_addr> (Bluetooth address of paired device) • 12-digit hexadecimal format (e.g. 20FABB112233) <profile_1> ... <profile_n> (Previously connected profiles) • ASCII string • Valid profile type (e.g. A2DP, AVRCP, BLE, etc.). See Table 4-2, Profile ID Values, on page 13 for profile types. **Example(s):** • Show the paired device list: LIST LIST 20FABB000155 A2DP AVRCP ... LIST 20FABB000188 BLE OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
MAP_GET_MSG	**Send request to retrieve message**

Details: After receiving a MAP_NEW_MSG notification, use this command to send a request to retrieve the message from the link handle specified in the notification.
The command returns PENDING, followed by one or more MAP_GET_MSG notifications containing the message data.

Notes: Message downloading requires a baud rate of 115200 or above.Lower baud rates can cause the UART to stall and lose message data.

Usage:

- Format: **MAP_GET_MSG <link_ID> <handle>**
  Response : PENDING
  MAP_GET_MSG <link_ID> <size> <data>
  ...
  OK
  Purpose: Get the message <data> from the specified link handle.

Parameters:

<link_ID> (Link identifier (MAP))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<handle> (Message handle included in MAP_NEW_MSG notification)
- 4-digit Hexadecimal format
- e.g. 000A

<size> (Message size, in bytes)
- Decimal format

<data> (Message content)
- ASCII string

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
MAP_GET_MSG (continued)	
Example(s): • Get message from device 1 (after receiving new message notification): *# New message received* MAP_NEW_MSG 18 166 <MAP-event-report version = “1.0”> <event type = "NewMessage" handle = "0123456789000002" folder = "TELECOM/MSG/INBOX" msg_type = "SMS_CDMA" /> </MAP-event-report> *# Get message* MAP_GET_MSG 18 0123456789000002 PENDING MAP_GET_MSG 18 666 BEGIN:BMSG VERSION:1.0 STATUS:UNREAD TYPE:SMS_GSM FOLDER:TELECOM/MSG/INBOX BEGIN:VCARD VERSION:2.1 N:IUT TEL:1234567891 END:VCARD BEGIN:BENV BEGIN:VCARD VERSION:2.1 N:PTS MAP_GET_MSG 18 405 TEL:14256913524 END:VCARD BEGIN:BBODY ENCODING:G-7BIT CHARSET:native LENGTH:290 BEGIN:MSG 0681000000000004048117600000218013219101C97FC8F71D340FD3D37373BA4C06DDCBF23228FFAE83EE693A1A44479741F3B2DC9E1E974170F9DB9E2697C920711E249487DDE4B71BF4AECB41F2323D9C6683A46538A8057ABB4161D07C1C66974166F9BB0D8AC140A8F1BB0D6797E965763E05A2BF413110CAFDA683C2745098CD4EFD00 END:MSG END:BBODY END:BENV END:BMSGOK	

Table 4-6: Melody Command Details (Continued)

Command	Description
MM_CFG	**Read/Write Music Manager Configuration**

Details: Read or write Music Manager (MM) configuration.

Usage:

- Format: **MM_CFG <key> [<length>]**

  Response (Read):
  MM_CFG <key> <config_value>
  OK

  Response (Write):
  PENDING
  (Enter <config_value>)
  OK

  Purpose: Read (use <key> only) or write (use <key> and <length>) MM configuration values.

Parameters:

<key> (Key index)
- Valid range: 24–38

<length> (Key length, in words)
- Valid range: 0–64

<config_value> (Configuration value)
- Series of hexadecimal words (4-digits)
- e.g. If <length>=2, <config_value> is two 4-digit values, such as 2AF1 0347.

Example(s):

- Write MM configuration value (0x0000, 0x1111, 0x2222, 0x3333, 0x4444):
  MM_CFG 24 5
  PENDING
  0000 1111 2222 3333 4444
  OK
- Read MM configuration
  MM_CFG 24
  MM_CFG 24 0000 1111 2222 3333 4444
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
MUSIC	**AVRCP Music Playback Control**

Details: Perform an AVRCP music playback action.

Usage:

- Format: **MUSIC <link_ID> <action>**
  Response : OK
  Purpose: Perform the <action> on the AVRCP/A2DP link.

Parameters:

<link_ID> (Link identifier (AVRCP or A2DP))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<action> (Music command)

- PLAY
- PAUSE
- STOP
- FORWARD
- BACKWARD
- FF_PRESS
- FF_RELEASE
- REW_PRESS
- REW_RELEASE

Example(s):

- Play music on the AVRCP link on device 1:
  MUSIC 11 PLAY
  OK

NAME	Get Remote Device Name

Details: Get the name of the remote device.

Usage:

- Format: **NAME <bd_addr>**
  Response: PENDING
  NAME <bd_addr> “<name>”
  Purpose: Get the name of the remote device at the specified Bluetooth address.

Parameters:

<bd_addr> (Bluetooth address of remote device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

<name> (Name of remote device)

- ASCII string

Example(s):

- Get the name of the device at Bluetooth address 20FABB00155:
  NAME 20FABB00155
  PENDING
  NAME 20FABB000150 “BC-000155”

Table 4-6: Melody Command Details (Continued)

Command	Description
OPEN	**Bluetooth Connection Request**

Details: Request a connection to a remote Bluetooth device.

Notes: Both devices (module and remote device) must be connectable to establish a connection. This command automatically makes the module connectable (if it is not already), but you must make sure the remote device is also connectable.
An OPEN_OK or OPEN_ERROR notification is received to indicate the outcome of the connection request.

Usage:

- Format: **OPEN <bd_addr> <profile> [<address_type>]**
  Response: PENDING
  (OPEN_OK notification, or OPEN_ERROR notification)
  Purpose: Attempt to open a <profile> connection to the device at <bd_addr>. The result of the attempt is indicated by an OPEN_OK or OPEN_ERROR notification.

Parameters:

<bd_addr> (Bluetooth address of remote device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

<profile> (Profile type to open)

- A2DP
- AGHFP
- AVRCP
- BLE
- HFP
- HID
- IAP
- MAP
- PBAP
- SPP
- TWS

<address_type> (Address type)

- 0—(Default) Public
- 1—Private (for BLE private address only)

Example(s):

- Request an SPP connection to a remote device, and successfully connect:
  OPEN 20FABB00155 SPP
  PENDING
  OPEN_OK 15 SPP 20FABB000150
- Request an SPP connection to a remote device, and fail to connect:
  OPEN 20FABB00155 SPP
  PENDING
  OPEN_ERROR SPP

Table 4-6: Melody Command Details (Continued)

Command	Description
PAIR	**Pairing Request**

Details: Request to pair with a remote Bluetooth device.

Notes: If desired, the module I/O capabilities can be configured. Refer to SSP_CAPS for details.

Usage:

- Format: **PAIR <bd_addr>**
  Response: PAIR_PENDING
  (PAIR_OK notification, or PAIR_ERROR notification)
  Purpose: Attempt to pair with the device at <bd_addr>. The result of the pairing attempt is indicated by a PAIR_OK or PAIR_ERROR notification.

Parameters:

<bd_addr> (Bluetooth address of remote device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

Example(s):

- Request to pair with a remote device:
  PAIR 20FABB00155
  PAIR_PENDING
  PAIR_OK 20FABB000150

PASSKEY	Pairing User Confirmation

Details: After receiving a PAIR_PASSKEY pairing request notification, use this command to accept or reject the pairing request.

Usage:

- Format: **PASSKEY <type> <value>**
  Response: OK
  (PAIR_OK notification, or PAIR_ERROR notification)
  Purpose: Accept or reject a pairing request that was received via a PAIR_PASSKEY notification. The result of the pairing acceptance/rejection is indicated by a PAIR_OK or PAIR_ERROR notification.

Parameters:

<type> (Pairing response type)

- 0—Confirmation type. <value>=1 (accept) or 0 (reject)
- 1—Passkey type. <value>=6-digit passkey

<value> (<type>-dependent response value)

- ASCII string
- See <type> descriptions

Example(s):

- Accept a pairing request:
  PAIR_PENDING
  PAIR_PASSKEY 20FABB00155 1
  PASSKEY 0 1
  OK
  PAIR_OK 20 FABB000150

Table 4-6: Melody Command Details (Continued)

Command	Description
PB_ABORT	**Abort Phonebook Download**

Details: Abort downloading of the phonebook from a specified link.

Notes: This command automatically aborts an active phonebook download (PB_PULL).
The PBAP profile cannot be closed if an active download is in progress. This command must be used before the PBAP profile can be closed via CLOSE.

Usage:

- Format: **PB_ABORT <link_ID>**
  Response : OK
  Purpose: Stop downloading the phonebook from the specified link.

Parameters:

<link_ID> (Link identifier (PBAP))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

Example(s):

- Stop downloading the phonebook from device 1:
  PBAP_ABORT 16
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
PB_PULL	**Download phonebook**

Details: Download the phonebook from a specified link.
The command returns PENDING, followed by one or more PB_PULL notifications containing phonebook entries. After the last entry is received, OK is returned.

Notes:
• Phonebook download requires a baud rate of 115200 or above. Lower baud rates can cause the UART to stall and lose phonebook data.
• Mandatory attributes that should be enabled all the time are: VERSION, N, and TEL.

Usage:

- Format: **PB_PULL <link_ID>[<repository>[<phonebook>[<maxlist>[<start_index>[<filter>]]]]]**
  Response : PENDING
  PB_PULL <link_ID> <size> <data>
  OK
  Purpose: Download the phonebook from the specified link.

Parameters:

<link_ID> (Link identifier (PBAP))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<repository> (Location of phonebook to be retrieved)
- 0—Current repository
- 1—(Default) Local repository. Phonebook will be retrieved from the phone's local memory.
- 2—SIM repository. Phonebook will be retrieved from the SIM card.
- 3—Any. The phone selects which repository to use.

<phonebook> (Phonebook entry)
- 1—(Default) Main phonebook folder
- 2—Incoming calls history
- 3—Outgoing calls history
- 4—Missed calls history
- 5—Combined call history

<maxlist> (Maximum number of phonebook entries to download from remote device)
- Default—0x1000
- If remote device has fewer than <maxlist> entries, then download finishes.

<start_index> (Phonebook index entry to start downloading from)
- Default—0x0000
- To download entire phonebook, set value to 0.

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
PB_PULL (continued)	
<filter> (Information types to download) • 32-bit value; each bit represents one phonebook entry attribute • Default—0x0087 (Bits 7,2,1,0). These bits must always be enabled. • Bit[0]—VERSION (vCard Version) • Bit[1]—FN (Formatted Name) • Bit[2]—N (Structured Presentation of Name) • Bit[3]—PHOTO (Associated Image or Photo) • Bit[4]—DAY (Birthday) • Bit[5]—ADR (Delivery Address) • Bit[6]—LABEL (Delivery) • Bit[7]—TEL (Telephone Number) • Bit[8]—EMAIL (Electronic Mail Address) • Bit[9]—MAILER (Electronic Mail) • Bit[10]—TZ (Time Zone) • Bit[11]—GEO (Geographic Position) • Bit[12]—TITLE (Job) • Bit[13]—ROLE (Role within the Organization) • Bit[14]—LOGO (Organization Logo) • Bit[15]—AGENT (vCard of Person Representing) • Bit[16]—ORG (Name of Organization) • Bit[17]—NOTE (Comments) • Bit[18]—REV (Revision) • Bit[19]—SOUND (Pronunciation of Name) • Bit[20]—URL (Uniform Resource Locator) • Bit[21]—UID (Unique ID) • Bit[22]—KEY (Public Encryption Key) • Bit[23]—NICKNAME (Nickname) • Bit[24]—CATEGORIES (Categories) • Bit[25]—PROID (Product ID) • Bit[26]—CLASS (Class information) • Bit[27]—SORT-STRING (String used for sorting operations) • Bit[28]—X-IRMC-CALL-DATETIME (vCard) • Bits[29]–[32]—Reserved <size> (<data> length) • Decimal format <data> (Phonebook entry containing selected <filter> information) • ASCII strings separated by <CR> (Continued on next page)	

Table 4-6: Melody Command Details (Continued)

Command	Description
PB_PULL (continued)	
Example(s): • Download (from local memory) the last 2 numbers dialed from device 1 (Total <size>=184 bytes): PB_PULL 16 1 3 2 0 85 PENDING PB_PULL 16 184 BEGIN:VCARD VERSION:2.1 FN;CHARSET=UTF-8:My Number N;CHARSET=UTF-8:My Number TEL;TYPE=CELL:+447446110144 X-IRMC-CALL-DATETIME;DIALED:20151110T133324 END:VCARD ... OK	
PIO	**Set GPIO State**
Details: Set general purpose I/O (GPIO) state. **Notes:** GPIO control must be disabled (via GPIO_CONFIG) before this command can be used. **Usage:** • Format: **PIO <pio> <state>** Response: OK Purpose: Set the specified <pio> ON or OFF. **Parameters:** <pio> (GPIO index) • Valid range: 0–5 <state> (GPIO state) • ON—High • OFF—Low **Example(s):** • Set GPIO 4 ON: PIO 4 ON OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
POWER	**Turn Bluetooth On/Off**
Details: Turn Bluetooth Classic functionality on or off. **Notes:** • OFF disconnects all active connections and puts the device in limbo mode, where it is not connectable or discoverable. • ON returns the device to an active state. • Power off is disabled for 3 seconds after power on. **Usage:** • Format: **POWER <state>** Response : OK Purpose: Turn Bluetooth Classic on or off. **Parameters:** <state> (Bluetooth Classic state) • ON—Active • OFF—Inactive **Example(s):** • Turn Bluetooth off: POWER OFF OK • Turn Bluetooth on: POWER ON OK	
REMOTE_VOLUME	**Set Remote Hands-Free Unit Volume**
Details: Set the volume of a remote hands-free unit. **Notes:** • REMOTE_VOLUME notifications are received when the HF unit informs of its new volume. • To set the volume of the BC127, use the VOLUME command for the same AGHFP link identifier. **Usage:** • Format: **REMOTE_VOLUME <link_ID> <volume>** Response : OK Purpose: Set the volume of the specified hands-free unit (<link_ID>). **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details. <volume> (HF speaker volume, in steps between 0 and max volume (0x0F)) • 1-digit Hexadecimal value • Range: 0–F (e.g 0=0%, F=100%) • This volume does not have a Melody API configuration. **Example(s):** • Set volume of <link_ID> to 2/3 of max volume (step 10 of 15): REMOTE_VOLUME 12 A OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
RESET	**Reset Module**
Details: Reset the BC127 module. **Usage:** • Format: **RESET** Response: *(module begins reset)* <module boot messages> Purpose: Reset the module. **Parameters:** None **Example(s):** • Reset the module: RESET Sierra Wireless Copyright 2018 Melody Audio V7.0 Build: 1522931717 Ready	
RESTORE	**Restore module to factory default configuration**
Details: Reset and restore the BC127 module to factory default configuration. **Usage:** • Format: **RESTORE** Response: *(module begins reset)* <module boot messages> Purpose: Restore the module to factory default configuration. **Parameters:** None **Example(s):** • Restore module to factory default configuration: RESTORE Sierra Wireless Copyright 2018 Melody Audio V7.0 Build: 1522931717 Ready	

Table 4-6: Melody Command Details (Continued)

Command	Description
ROLE	**Read/change HCI role**

Details: Read or set (change) the HCI role for a specified link.

Notes:
- A ROLE notification (ROLE_OK or ROLE_NOT_ALLOWED) is received to indicate the result of the command.
- If the module is a Slave connected to multiple devices, it automatically attempts to switch role to Master.
- Some Bluetooth devices do not allow the role to be switched.
- Role switch is not allowed when a SCO is open.

Usage:

- Format: **ROLE <link_ID>[<role>]**

  Response (Read):
  OK
  <notification>

  Response (Set):
  PENDING
  <notification>

  Purpose: Set the HCI role of the <link_id>.

Parameters:

<link_ID> (Link identifier (A2DP, AVRCP, HFP, AGHFP, iAP, or SPP))
- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<role> (HCI role)
- M—Master
- S—Slave

<notification> (ROLE notification)
- ROLE_OK
- ROLE_NOT_ALLOWED

Example(s):
- Get role of device 1:
  ROLE 15
  OK
  ROLE_OK 20FABB000150 S
- Set device 1 role to Master:
  ROLE 15 M
  PENDING
  ROLE_OK 20FABB000150 M
- Fail to switch role to Master:
  ROLE 15 M
  PENDING
  ROLE_NOT_ALLOWED 20FABB000150 S

Table 4-6: Melody Command Details (Continued)

Command	Description
ROUTE	**Set/Get Audio Routing**

Details: Set or get audio routing.

Notes:
• Automatic routing is based on priorities—An HFP link has priority over an A2DP link. If more than one device is connected, the first device has the highest priority. Audio received from a broadcaster has the lowest priority.
• Music Manager and tones features are not supported with <link_ID> special cases 2 and 3.
• With <link_ID> special case 1 or with an A2DP Source link, if a TWS connection is established the audio is automatically relayed.

Usage:

- Format: **ROUTE [<link_ID_1_special> [<link_ID_2>]]**

  Response (Get):
  ROUTE <link_ID_1_special>
  OK

  Response (Set):
  OK

  Purpose: Get (use no parameters) or set (use one or two parameters) the audio routing.

Parameters:

<link_ID_1_special> (Link identifier (A2DP/HFP/AGHFP), or special case)

- Format:
  - 8-bit (two digit) hexadecimal value
    or
  - Special case:
    - 0—(Default) Automatic routing
    - 1—Analog input to analog output
    - 2—Digital interface to analog interface (bidirectional)
    - 3—Digital input to digital output
    - 4—Digital input to analog output
    - 7—Broadcast Audio (receiver mode on BA releases only)
    - FF—Audio disabled
- See Link ID Management on page 13 for details.

<link_ID_2> (Second Link identifier (A2DP))

- 8-bit (two digit) hexadecimal value
- Applies only to A2DP Dual Stream feature
- See Link ID Management on page 13 for details.

Example(s):

- Route A2DP to/from device 1:
  ROUTE 10
  OK
- Route digital input to digital output:
  ROUTE 4
  OK
- Get current audio routing:
  ROUTE
  ROUTE 4
  OK
- Set Automatic routing:
  ROUTE 0
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
RSSI	**Get Signal Strength**

Details: Get the signal strength (in dBm).

Notes: Signal quality examples—-70 dBm is a good link; -80 dBm is a poor link

Usage:

- Format: **RSSI <link_ID>**
  Response: <rssi> dBm
  OK
  Purpose: Get the current signal strength.

Parameters:

<link_ID> (Link identifier (A2DP, AVRCP, HFP, AGHFP, SPP, or iAP))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<rssi> (Current signal strength in dBm (RSSI—Received Signal Strength Indicator))

- Signal strength
- e.g. -62

Example(s):

- Get signal strength from device 1:
  RSSI 15
  -62 dBm
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
SCAN	**Search for BLE Devices**

Details: Search for all (or a subset of all) BLE devices in the area
Scan results are returned by SCAN or SCAN_RAW notifications (for each result), depending on the selected <format>.

Notes:
- The same device might appear multiple times.
- To scan for Bluetooth Classic devices, use INQUIRY.

Usage:

- Format: **SCAN <timeout> [<format>[<filter>]]**

  Response (Melody 7.0/7.1):
  PENDING
  SCAN <bd_addr> <type> <name> <flags> <rssi> *or* SCAN_RAW <bd_addr> <rssi> <size> <data>
  ...
  SCAN_OK

  Response (Melody 7.2 and later):
  PENDING
  SCAN <bd_addr> <type> <name> <flags> <rssi> *or* SCAN_RAW <bd_addr> <type> <rssi> <size> <data>
  ...
  SCAN_OK

  Purpose: Search for BLE devices.

Parameters:

<timeout> (Scan duration (search time limit), in seconds)
- Valid range: 0–255
- 0—No timeout

<format> (Scan results format)
- ON—SCAN_RAW notifications
- OFF—(Default) SCAN notifications

<filter> (Show only devices supporting BC Smart Profile)
- ON—Only BC Smart devices
- OFF—No filter

<bd_addr> (Device Bluetooth address)
- 12-digit hexadecimal format (e.g. 20FABB112233)

<type> (Address type)
- 0—Public
- 1—Private

<name> (Device name)
- ASCII string, device name includes “< >”
- e.g. <BC00150>, <UNKNOWN>

<flags> (Advertising flags)
- Hexadecimal value, as defined in Bluetooth specification

<rssi> (Signal strength, in dBm)
- ASCII string
- e.g. -49dBm

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
SCAN (continued)	
Example(s): *Note: Melody 7.0/7.1 INQUIRY notifications incorrectly indicate signal strength units as "dB" instead of "dBm".* • Scan for all BLE devices (no filter, default parameters): SCAN 5 PENDING SCAN 20FABB000150 0 <BC00150> 1A -56dBm ... SCAN 28F0765A885F <UNKNOWN> 06 -91dBm SCAN_OK • Scan for all BLE devices (SCAN_RAW, no filter, default parameters): SCAN 2 ON PENDING SCAN_RAW 1891D15160A6 1 -97dBm 31 1E FF 06 00 01 09 20 00 7B 85 AE 92 76 02 D6 6F 26 9C AB 06 45 CC A4 C1 19 29 EA 26 93 D0 07 SCAN_RAW 20FABB000100 0 -66dBm 31 02 01 06 09 09 42 43 30 30 30 31 30 30 11 06 F0 28 E3 68 62 D6 34 90 51 43 EF AA C6 4C 2F BC ... SCAN_RAW 3321B77632C0 1 -65dBm 31 1E FF 06 00 01 09 20 00 00 55 EA F3 0F EC AE 55 29 5C D2 96 2E 88 87 37 2F 41 A7 0D 6D 5A AA SCAN_OK	
SEND	**Send Data**

Details: Send data to a BLE, SPP, or iAP link.
 	
Notes: For iAP links, the command is allowed only if an EA session is opened (i.e. the IAP_OPEN_SESSION notification must be received first.
Usage:
• Format: **SEND <link_ID> <string>**
Response: OK
Purpose: Send the data <string> to the specified <link_ID>.
Parameters:
<link_ID> (Link identifier (BLE, SPP, or iAP))
• 8-bit (two digit) hexadecimal value
• See Link ID Management on page 13 for details.
<string> (Data string to send)
• ASCII string, no quotation marks required—everything after the <link_ID> is part of the <string>
• String length:
  • BLE link—maximum length=MTU-3; remainder of string is truncated
  • SPP/iAP link—unlimited (entire string up to <CR> is used)
• e.g. Hello world!
Example(s):
• Send “Hello world!” over SPP on device 1:
SEND 15 Hello world!
OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
SEND_RAW	**Send Raw Data**

Details: Send raw data to a BLE, SPP, iAP, or HID link.
After sending the command, a PENDING response is received. The data received after that is used to set the raw data to send. OK is returned when the expected number of bytes is received.

Notes: For iAP links, the command is allowed only if an EA session is opened (i.e. the IA_OPEN_SESSION notification must be received first.

Usage:

- Format: **SEND_RAW <link_ID> <size>**
  Response: PENDING
  *(Send <data>)*OK
  Purpose: Send <size> bytes of raw data to the specified <link_ID>.

Parameters:

<link_ID> (Link identifier (BLE, SPP, iAP, or HID))

- 8-bit (two digit) hexadecimal value
- See Link ID Management on page 13 for details.

<size> (Number of bytes to send)

- Decimal format
- Valid range: 1–255

<data> (Data being sent)

- Raw data
- e.g. {68}{65}{6c}{6c}{0a} indicates five bytes with values 0x68, 0x65, 0x6c, 0x6c, 0x0a are sent (the hexadecimal values of the ASCII characters in “Hello”)

Example(s):

- Send “Hello” (5 bytes of data) over SPP on device 1:
  SEND_RAW 15 5
  PENDING
  {68}{65}{6c}{6c}{0a}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
SET	**Set Single Configuration Value**
Details: Set a single Melody configuration. (e.g. Set the AUDIO configuration, which has two parameters associated with it.) **Notes:** Some configuration changes only take effect after a reboot. To make those changes take effect, save them using the WRITE command, and then reboot. **Usage:** • Format: **SET <config_name>=<param_1> ... <param_n>** Response: OK Purpose: Set the configuration (<config_name>) item to the specified parameter values. **Parameters:** <config_name> (Melody configuration name) • ASCII string • See Configurations for available configurations. <param_1> ... <param_n> (Configuration parameters) • See the <config_name> description in Configurations for details. **Example(s):** • Configure audio input/output: SET AUDIO=1 1 OK	
SPEECH_REC	**Activate/Deactivate Speech Recognition**
Details: Activate or deactivate speech recognition. **Notes:** • A valid cVc license is required to support speech recognition. • SR (Speech Recognition) notifications are received when a word is recognized. **Usage:** • Format: **SPEECH_REC <sr_state>** Response: OK Purpose: Enable/disable speech recognition. **Parameters:** <sr_state> (Speech recognition state) • ON—Speech recognition enabled • OFF—Speech recognition disabled **Example(s):** • Enable speech recognition: SPEECH_REC ON OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
SSRD	**Set Scan Response Data**

Details: Set scan response data.
After sending the command, a PENDING response is received. The data received after that is used to set the scan response data. OK is returned when the expected number of bytes is received.

Usage:

- Format: **SSRD <size>**
  Response: PENDING
  *(Enter <data>)*OK
  Purpose: Set <size> bytes of scan response data.

Parameters:

<size> (Number of bytes to set)

- 1–31

<data> (Data being sent)

- Raw data
- e.g. {07}{09}{4D}... indicates bytes with values 0x07, 0x09, 0x4D, etc., are sent

Example(s):

- Set 8 bytes of scan response data:
  SSRD 8
  PENDING
  {07}{09}{4D}{79}{43}{61}{6D}{65}OK

Table 4-6: Melody Command Details (Continued)

Command	Description
STATUS	**Return Device Connection Status**

Details: Return the connection status of the device.

Usage:

- Format: **STATUS [<link_ID>|<device_ID>|<profile>]**

  Response : STATE CONNECTED[<nb_dev>] CONNECTABLE[<connectable>] DISCOVERABLE[<discoverable>] BLE[<ble_state>]
  BA MODE[<mode>] ASSOC[<assoc>] BROADCAST[<broad>]
  LINK <link_ID> <link_state> <profile>[<bd_addr>[<additional_info>]]
  ...
  OK

  Purpose: Return the BC127 module's current state, and link information for each connected profile. (Note: The BA MODE... portion of the response appears only on BA releases.)

Parameters:

<link_ID> (Link identifier)
- 8-bit (two digit) hexadecimal value; all profile types supported
- See Link ID Management on page 13 for details.

<device_ID>
- First digit of the <link_ID>

<profile> (Profile type to open)
- A2DP
- AGHFP
- AVRCP
- BLE
- HFP
- HID
- IAP
- MAP
- PBAP
- SPP
- TWS

<nb_dev> (Number of devices connected)
- Valid range: 0–5

<connectable> (Module's connectable state)
- ON—Connectable
- OFF—Not connectable

<discoverable> (Module's discoverable state)
- ON—Discoverable
- OFF—Not discoverable

<ble_state> (Module's BLE state)
- OFF
- IDLE
- ADVERTISING
- CONNECTED

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
STATUS (continued)	

<mode> (Current Broadcast Audio mode)
• 0—Broadcast Audio disabled
• 1—Broadcaster mode
• 2—Receiver mode
<assoc> (Association state)
• 0—Association disabled
• 1—Association enabled/in progress
• (Melody 7.0/7.1 only) 2—Associated (receiver mode only)
<broad> (Broadcast Audio state)
• 0—Not broadcasting/receiving audio
• 1—Broadcasting/receiving audio
<link_state> (Current state of <link_ID>)
• CONNECTED
• LINK_LOSS
• DISCONNECTED
<bd_addr> (Bluetooth address of remote device)
• 12-digit hexadecimal format (e.g. 20FABB112233)
<additional_info> (Profile-specific information)
• For A2DP/TWS:
  • Format: <streaming_state> <codec> <role> <sample rate>
  • <streaming_state>:
  • SUSPENDED
  • STREAMING
  • <codec>:
  • SBC
  • AAC
  • APTX
  • APTX_LL
  • APTX_HD
  • <role>:
  • SNK (A2DP Sink)
  • SRC (A2DP Source)
  • <sample_rate>:
  • A2DP sample rate, in Hz
• For AVRCP:
  • Format: <avrcp_status>
  • <avrcp_status>:
  • STOPPED
  • PLAYING
  • PAUSED
(Continued on next page)	

Table 4-6: Melody Command Details (Continued)

Command	Description
STATUS (continued)	

• For BLE:
  • Format: <mtu>
  • <mtu>:
  • MTU negotiated with the remote device
• For HFP/AGHFP:
  • Format: <call_state> <codec>
  • <call_state>:
  • IDLE
  • OUTGOING
  • INCOMING
  • ACTIVE
  • <codec>:
  • NB (narrow band)
  • WB (wide band)
• For iAP:
  • Format: <EA_session>
  • <EA_session>:
  • 0—EA session not opened
  • 1—EA session is opened
Example(s):
• Display the status of all links:
STATUS
STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE]
BA MODE[1] ASSOC[0] BROADCAST[1]
LINK 10 CONNECTED A2DP 20FABB000161 SUSPENDED SBC SRC 44100
LINK 11 CONNECTED AVRCP 20FABB000161 STOPPED
LINK 13 CONNECTED HFP 20FABB000161 IDLE NB
LINK 19 CONNECTED IAP 20FABB000161 0
OK	

TOGGLE_VR	**Start/Stop Voice Recognition**
Details: Start or stop voice recognition over HFP. **Usage:** • Format: **TOGGLE_VR <link_ID>** Response: OK Purpose: Start or stop voice recognition over HFP on the specified <link_ID>. **Parameters:** <link_ID> (Link identifier (HFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details. **Example(s):** • Toggle voice recognition on device 1: TOGGLE_VR 13 OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
TONE	**Play Tone**

Details: Play a tone (a sequence of notes).
Tones are defined as a sequence of notes described with several characteristics—pitch (N or TN), length (L), tempo (TE), volume (V), timbre (TI) and decay (D).

Notes:
- All characteristics can be individually set for each note, changed at any place in the tone string, or omitted (all except length) to use default values.
- A tone must have at least one note, and each note must have at least a length characteristic set (e.g. “N AS7 L 1”—note pitch A sharp, octave 7, whole note).

Usage:

- Format: **TONE <flag> <value> [... <flag> <value>]**
  Response: OK
  Purpose: Play a tone (note sequence).

Parameters:

<flag> (Tone characteristic)

- TE—Tempo in quarter notes per minute.
  <value> range—0–4095; default 120
- TI—Timbre
  <value>:
  - 0—Sine wave
  - 1—Square wave
  - 2—Saw-tooth wave
  - 3—Triangle wave
  - 4—Triangle wave (asymmetric)
  - 5—Clipped sine wave
  - 6—Simulates a plucked instrument
- V—Volume.
  <value> range—0–255; default TBD
- D—Decay
  <value>:
  - Two-digit hexadecimal value, range 00–FF
    Interpreted as fixed point decimal number in format <digit_1>.<digit_2>.
    e.g. 15=1.5
  - As each tone is played, its volume decreases with variable rate. Low values for this parameter cause notes to decay very quickly; high values cause notes to continue with an almost-constant volume.
    For example:
    05 (0.5) causes each note to reach zero halfway through its duration, giving a staccato feel.
    20 (2.0) causes each note to reach half its initial value when the next note starts.
  - Default: 20. This allows notes of the same length to be tied together with TN.
- N—Note pitch
  <value> format—<note>[<flat_sharp>]<octave>
  - <note>: A, B, C, D, E, F, G, or R (Rest/Pause)
  - <flat_sharp>: F (Flat) or S (Sharp)
  - <octave>: 0–9. (Note: Must be specified for all <note> values, including ‘R”, for consistency—e.g. R0))
- TN—Tied note. Note is “tied” to the next note (the pitch of the note appears to change as opposed to a new note starting), and the volume continues to decay from the previous note.
  Note: An appropriate decay must be set for this to work.

(Continued on next page)

Table 4-6: Melody Command Details (Continued)

Command	Description
TONE (continued)	
• L—Length of note <value>: • 1—Whole note • 2—Half note • 3—Half note triplet • 4—Quarter note • 6—Quarter note triplet • 8—Eighth note • 12—Eighth note triplet • 16—Sixteenth note • 24—Sixteenth note triplet • 32—Thirty-second note • 48—Thirty-second note triplet • 64—Sixty-fourth note • 96—Sixty-fourth note triplet <value> (<flag>-dependent value) • See <flag> parameter descriptions for supported values. **Example(s):** • Sample tone: TONE TE 400 V 64 TI 0 N C5 L 8 N R0 L 32 N E5 L 8 N R0 L 32 N G5 L 8 N R0 L 32 N B5 L 4 N R0 L 1 N C6 L 2 TN C6 L 8 OK • Variable volume single note: TONE V 64 N C6 L 4 V 128 N C6 L 4 V 255 N C6 L 4 V 128 N C6 L 4 V 64 N C6 L 4 OK • Musical scale starting from C4 and omitting flat and sharp tones: TONE V 128 TI 0 N C4 L 8 N D4 L 8 N E4 L 8 N F4 L 8 N G4 L 8 N A4 L 8 N B4 L 8 N C5 L 8 OK	

Table 4-6: Melody Command Details (Continued)

Command	Description
TX_POWER	**Get/Set Transmitter Power**

Details: Get or set the default and maximum transmit power values.

Usage:

- Format: **TX_POWER [<default_tx_power> <max_tx_power>]**

  Response (Get):

  TX_POWER <default_tx_power> <max_tx_power>
  OK

  Response (Set):

  OK

  Purpose: Get (no parameters) or set (both parameters) the default and maximum transmitter power.

Parameters:

<default_tx_power> (Default transmit power, in dBm)

- Value in 'Set' format is rounded down to the next available value in the power table (-20, -16, -12, -8, -4, 0, 4, 8). Therefore, the value returned using 'Get' format may be lower than the value entered in the 'Set' format.
- Used for paging, inquiry, and their responses. Also used as initial power for new ACL (Asynchronous Connection-Less) links.

<max_tx_power> (Maximum transmit power, in dBm)

- Value is referenced when increasing transmit power. If transmit power on a link is already above this level, the new value does not take effect until an attempt is made to increase the power.
- Value in 'Set' format is rounded down to the next available value in the power table (-20, -16, -12, -8, -4, 0, 4, 8). Therefore, the value returned using 'Get' format may be lower than the value entered in the 'Set' format.

Example(s):

- Get Tx power (values returned are 4 dBm default, and 8 dBm max):
  TX_POWER
  TX_POWER 4 8
  OK
- Set Tx power to 0 dBm default and 4 dBm max:
  TX_POWER 0 4
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
UNPAIR	**Unpair Devices**

Details: Remove one or all Bluetooth devices from Paired Device List (PDL) and delete authentication information.

Notes: To display the devices in the PDL, use the LIST command.

Usage:

- Format: **UNPAIR [<bd_addr>]**
  Response: OK
  Purpose: Remove a specific device from the PDL (using the <bd_addr>) or remove all devices from the PDL (no parameter).

Parameters:

<bd_addr> (Bluetooth address of remote device)

- 12-digit hexadecimal format (e.g. 20FABB112233)

Example(s):

- Unpair a single device:
  UNPAIR 20FABB00161
  OK
- Unpair all devices:
  UNPAIR
  OK

Command	Description
VERSION	**Display Module Firmware Version and Bluetooth Address**

Details: Display the BC127 module's version information, Bluetooth addresses (public and private), and other information.

Usage:

- Format: **VERSION**
  Response: <version_information>
  OK
  Purpose: Display module information.

Parameters:

<version_information> (Details may vary)

- Copyright information—e.g. "Sierra Wireless Copyright 2018"
- Firmware information—e.g. "Melody Audio V7.0", "Build: 1522931717"
- Bluetooth addresses—e.g. "20FABB000160 7191D978637D"
  The first address is the public address. The second address is the BLE address, which is different when the module is configured to use a BLE private randomly-generated address (see BLE_CONFIG).
- Profiles—e.g. "A2DP AV4CP HFP BLE SPP PBAP MAP TWS"
- Codecs—e.g. "SBC"

Example(s):

- Display information:
  VERSION
  Sierra Wireless Copyright 2018
  Melody Audio V7.0
  Build: 1522931717
  Bluetooth addresses: 20FABB000160 7191D978637D
  Profiles: A2DP AVRCP HFP BLE SPP PBAP MAP TWS
  Codecs: SBC
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
VOLUME	**Get/Set Volume**

Details: Get or set volume.

Notes:
- Number of steps for <value> is configurable for A2DP (see BT_VOL_CONFIG).
- To set the volume of a remote device, use the REMOTE_VOLUME command.

Usage:

- Format: **VOLUME [<link_ID>[<value>]]**
  Response : OK
  Purpose: Get (no parameters) or set (1–2 parameters) the volume on <link_ID>.

Parameters:

<link_ID> (Link identifier (A2DP, HFP AGHFP), or 01 for the analog interface)
- Format:
  - 8-bit (two digit) hexadecimal value
    or
  - 01—Analog interface
- See Link ID Management on page 13 for details.

<value> (Volume direction, or volume in number of steps between 0 and max volume)
- Number of steps:
  - For HFP—16 steps
  - For A2DP—Default=16, configurable via BT_VOL_CONFIG.
- Valid values:
  - UP—Increase volume one step
  - DOWN—Decrease volume one step
  - Hexadecimal value. Valid range: 0–(<nb_steps>-1)

Example(s):
- Get volume for all A2DP, HFP, AGHFP links, and analog interface:
  VOLUME
  01 ANLG A
  10 A2DP A
  13 HFP A
  OK
- Increase volume on the A2DP link (device 1):
  VOLUME 10 UP
  OK
- Set volume to 2 on the A2DP link (device 1):
  VOLUME 10 2
  OK

Table 4-6: Melody Command Details (Continued)

Command	Description
WRITE	**Save Current Melody Configuration**
Details: Save the current Melody configuration (all values) to persistent memory. **Notes:** For Melody configuration value details, refer to Configurations. **Usage:** • Format: **WRITE** Response : OK Purpose: Save Melody configuration. **Parameters:** none **Example(s):** • Save Melody configuration: WRITE OK	

Configurations

Melody configuration values (listed in Table 4-8) are managed using the commands listed in Table 4-7.

Note: When using SET (to modify configuration values) or RESTORE (to reset all values to default), some changes take effect immediately, and others take effect after a reboot. Before rebooting the module, make sure to use WRITE to save the configuration values.

Table 4-7: Melody Configuration Commands

Command	Description	Page
CONFIG	Get All Configurations	41
GET	Get Single Configuration	44
RESTORE	Restore module to factory default configuration	65
SET	Set Single Configuration Value	72
WRITE	Save Current Melody Configuration	82

Table 4-8: Melody Configurations Summary

Configuration	Description	Reboot Required	Page
AUDIO	Audio Interface	No	85
AUDIO_ANALOG	Audio Analog Configuration	No	85
AUDIO_DIGITAL	Audio Digital Configuration	No	86
AUTOCONN	Auto-connection at Power On	Yes	88
AUTO_DATA	Automatically Enter Data Mode	Yes	89
BA_CONFIG	Broadcast Audio Mode	Yes	90
BALANCE	Left–Right Audio Balance	No	90
BATT_CONFIG	Battery Configuration	Yes	91
BC_SMART_CONFIG	BC Smart Profile Configuration	Yes	91
BEACON_DATA	BLE Beacon Configuration	Yes	92
BLE_CONFIG	BLE Configuration	Yes[a]	93
BLE_CONN_PARAMS	BLE Advertising, Connection, and Scanning Parameters	Yes	94
BT_STATE_CONFIG	Bluetooth Classic State Configuration	Yes	95
BT_VOL_CONFIG	Bluetooth Classic Volume Configuration	Yes	96
CLASS_1	Enable/Disable GPIO Control of Class 1 Device Using External PA	Yes	96

Table 4-8: Melody Configurations Summary (Continued)

Configuration	Description	Reboot Required	Page
CMD_TO	Guard Time for Escape Sequence ($$$$)	No	97
COD	Class of Device	No	97
CODEC	A2DP Optional Codecs	Yes	98
DEEP_SLEEP	Enable/Disable Deep Sleep	No	98
DEVICE_ID	Device ID Profile Configuration	Yes	99
ENABLE_BATT_IND	Enable/Disable iOS Battery Indication	No	99
ENABLE_LED	Enable/Disable LEDs	No	100
ENABLE_SPP_SNIFF	SPP Sniff Mode Configuration	No	101
GPIO_CONFIG	GPIO Configuration	Yes	102
HFP_CONFIG	HFP/AGHFP Profile Configuration	Yes	103
HIGH_SPEED	High Speed Configuration	No	104
IAP_PARAMS	IAP Link Parameters	Yes	105
LOCAL_ADDR	Bluetooth addresses	N/A	106
MAX_REC	Maximum Reconnection Attempts	Yes	106
MM	Music Manager Enhancements	No	107
MUSIC_META_DATA	Enable/Disable AVRCP Metadata	No	108
MUSIC_OLD_AVRCP	Switch to AVRCP v1.0	Yes	108
NAME	Device Name	Yes	108
NAME_SHORT	Device Short Name	Yes	109
PIN	PIN Code	No	109
PROFILES	Bluetooth Profiles	Yes	110
REMOTE_ADDR	Auto-connection Remote Address	Yes	111
SPP_UUID	SPP Profile UUID	Yes	111
SSP_CAPS	Secure Simple Pairing I/O Capabilities	No	111
TWS_CONFIG	True Wireless Stereo Configuration	No	112
UART_CONFIG	UART Configuration	Yes[b]	113
USB_HOST	Enable USB Host Interface	Yes	113
VREG_ROLE	VREG Button Configuration	No	114

a. BLE_CONFIG—Reboot required for <adv_mode> and <private_address> changes only.
b. UART_CONFIG—Reboot required for <enable_flow_control> changes only.

Table 4-9: Melody Configurations Details

Configuration	Description
AUDIO	**Audio Interface**
Details: Configure the audio interface input and output signals as analog or digital. **Usage:** • Format: **AUDIO=<input>[<output>]** Purpose: Configure audio input and output. If only one value (<input>) is set, it applies to both input and output. **Default:** AUDIO=0 0 (Analog input and output) **Reboot required:** No **Parameters:** <input> (Audio input) • 0—Analog input • 1—Digital input <output> (Audio output) • 0—Analog output • 1—Digital output	
AUDIO_ANALOG	**Audio Analog Configuration**
Details: Configure analog audio parameters. These configuration parameters are taken into account when routing the audio. **Notes:** • In designs using a microphone, enabling pre-amp is recommended. • For HFP or A2DP, or with ROUTE 1 and 4, output gain is not relevant—it is controlled by the DSP based on the current volume. The VOLUME command must be used to change the output gain. • For HFP, when cVc is active, input gain is controlled by the DSP—the cVc configuration can be changed using the CVC_CFG command. **Usage:** • Format: **AUDIO_ANALOG=<input_gain> <output_gain> <mic_bias> <enable_preamp>** Purpose: Configure analog gain, mic bias, and microphone pre-amp parameters. **Default:** AUDIO_ANALOG=15 15 1 OFF (Codec input gain (15 dB), output gain (15 dB), Mic bias on only when there is audio, and microphone pre-amp disabled) **Reboot required:** No **Parameters:** <input_gain> (Codec input gain) • Decimal format • Valid range: 0–22 <output_gain> (Codec output gain) • Decimal format • Valid range: 0–15 <mic_bias> (Mic bias configuration) • 0—Disabled • 1—Turned on only when there is audio • 2—Enabled alway <enable_preamp> (Enable microphone pre-amp) • ON—Enabled. Results in additional 20 dB of input gain on the channel. Recommended in designs using a microphone. • OFF—Disabled	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
AUDIO_DIGITAL	**Audio Digital Configuration**

Details: Configure digital audio parameters. These configuration parameters are taken into account when routing the audio.

Notes:
- Re-sampling:
  - Automatically done by the DSP when the digital rate differs from the A2DP or HFP audio sampling rate, using specified <rate>.
  - Can be disabled by setting <enable_auto_rate> to ON. Digital interface rate will match audio sampling rate.
  - HFP/AGHFP—Audio sampling rate: 8 kHz (Narrow Band (NB)), 16 kHz (Wide Band (WB))
    HFP/AGHFP—Supported digital rates (<rate>): 8000, 16000, 32000, 44100, 48000
  - A2DP—Audio sampling rate: 44.1 kHz or 48 kHz, may vary according to the codec used.
    A2DP—Supported digital rates (<rate>): 44100, 48000
- I2S:
  - Bit clock (BCLK) = Word Clock (WCLK)×Bit clock scaling factor, where WCLK=<rate>, and Bit clock scaling factor=<param_1>
  - Default BCLK=44100×64=2.822 MHz
  - Bit clock (BCLK) should be:
    (2×bps×WCLK) ≤ BCLK ≤ (256×WCLK)
  - If the frame is fully packed, there are no idle BCLK cycles (ie BCLK = 2×bps×WCLK), crop (<param_2> Bit[2]) should be enabled.
- PCM:
  - When configured as PCM interface Master
    - Module generates PCM_CLK and PCM_SYNC.
      PCM_CLK—128/256/512/1536/2400 kHz
      PCM_SYNC—8/16/32/48 kHz
  - When configured as PCM interface Slave, module supports:
      PCM_CLK—Up to 2400 kHz
      PCM_SYNC—8/16/32/48 kHz
- For configuration examples, refer to Digital Audio Configuration on page 157.

Usage:

- Format: **AUDIO_DIGITAL=<format> <rate> <param_1> <param_2>[<enable_auto_rate>]**
  Purpose: Specifiy digital audio format and configure format-specific parameters.

Default: AUDIO_DIGITAL=0 44100 64 100A00 OFF
(I2S Audio, WCLK=44100, BCLK scaling factor=64, 100A00 (I2S master, 16 bps, left-justified with 1 bit delay), auto-adjust digital rate to audio sampling rate disabled)

Reboot required: No

Parameters:

<format> (Digital format)
- 0—I2S
- 1—PCM
- 2—SDPIF

(Continued on next page)

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
AUDIO_DIGITAL (continued)	

<rate> (Digital rate, <format>-dependent)
• For I2S/PCM—Word clock (WCLK)
• For SDPIF—Output rate, in Hz
• For A2DP—Supported rates: 44100, 48000
• For HFP/AGHFP—Supported rates: 8000, 16000, 32000, 44100, 48000
 	
<param_1> (<format>-dependent content)
• For I2S—Bit clock (BCLK) scaling factor
• For PCM—Master clock, in kHz
• For SDPIF—Not used
 	
<param_2> (<format>-dependent content)
• Four-byte bitmap
• For I2S:
  • Bit[24:31]—Not used
  • Bit[16:23]—Bits per sample (bps).
  • Valid values: 16, 20, 24
  • Bit[12:15]—Audio attenuation, in 6 dB steps. Valid range: 0–15
  • Bit[11]—Mode (0—Slave; 1—Master). In Master mode, clock and sync will be generated by the I2S hardware.
  • Bit[10]—Justify format. 0 (left-justified), 1 (right-justified)
  • Bit[9]—Left justify delay:
  • 0—MSB of SD data occurs in the first SCLK period following WS transition.
  • 1—MSB of SD data occurs in the second SCLK period.
  • Bit[8]—Channel polarity. 0 (SD data is left channel when WS is high), 1 (SD data is right channel when WS is high)
  • Bit[7]—Audio attenuation enable. 0 (17-bit SD data is rounded down to 16 bits); 1 (Audio attenuation defined in Audio attenuation (Bits 12:15) is applied over 24 and 20 bits of incoming data with saturated rounding. Requires crop enable (Bit 2) to be 0.
  • Bit[5:6]—Not used
  • Bit[3:4]—Justify resolution (resolution of data on SD_IN), required for right-justified format and with left-justified LSB first.
  Values: 00 (16-bit); 01 (20-bit); 10 (24-bit); 11 (Reserved)
  • Bit[2]—Crop enable. 0 (17-bit SD_IN data is rounded down to 16 bits; 1 (only the most significant 16 bits of data are received)
  • Bit[1]—Start Tx sampling. 0 (During low WCLK phase); 1 (During high WCLK phase)
  • Bit[0]—Start Rx sampling. 0 (During low WCLK phase); 1 (During high WCLK phase)
• For PCM:
  • Bit[26:31]—Not used
  • Bit[25]—Enable PCM master mode (0—Slave; 1—Master). In master mode, clock and sync will be generated by the PCM hardware.
  • Bit[22:24]—PCM slot count. Valid range: 0–4. If 0, slot count will be derived from master clock and synchronization rate.
  • Bit[21]—Enable PCM Manchester encoding mode.
  • Bit[20]—Enable PCM short synchronization. In short frame sync, falling edge indicates start of frame, and in long frame sync, rising edge indicates start of frame.
  • Bit[19]—Enable PCM Manchester slave mode. Force transmit frames to follow receive frames with constant delay. Requires extended features to be enabled.
  • Bit[18]—Enable PCM sign extend. Sign extend 13/8 bit sequence to 16 bit sequence, otherwise pad with the Audio gain (Bits[0:2]) for 13 bits or zeros (0) for 8 bits.
 	
(Continued on next page)	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
AUDIO_DIGITAL (continued)	
• Bit[17]—Enable PCM LSB first. Transmit data LSB first. • Bit[16]—Enable PCM Tx tristate. 0 (Drive PCM_OUT continuously); 1 (Tri-state PCM_OUT immediately after falling edge of PCM_CLK in the last bit of an active slot, assuming the next slot is not active.) • Bit[15]—Enable PCM Tx tristate rising edge. 0 (Tri-state PCM_OUT immediately after falling edge of PCM_CLK in last bit of an active slot, assuming the next slot is not active.); 1 (Tri-state PCM_OUT after rising edge of PCM_CLK.) • Bit[14]—Enable PCM synchronization suppress. Suppress PCM_SYNC while generating PCM_CLK (in master mode). Some CODECs (connected to the PCM interface) use this to enter a low power state. • Bit[13]—Enable PCM GCI mode. • Bit[12]—Mute PCM_DATA output. • Bit[11]—Enable PCM long length sync. Set PCM_SYNC to 8 or 16 PCM_CLK cycles. • Bit[10]—Enable PCM sample rising edge. Sample PCM_DATA on rising edge of PCM_CLK. • Bit[7:9]—Rx rate delay. Selects the number of clocks to wait before receive DDS update rate is changed to match the new internal clock frequency. Valid range: 0–7 • Bit[5:6]—Sample format. 0 (13 bits in 16 cycle slot duration); 1 (16 bits in 16 cycle slot duration), 2 (8 bits in 16 cycle slot duration), 3 (8 bits in 8 cycle slot duration) • Bit[3:4]—Manchester receive offset. When in Manchester mode, selects the delay between receiving the start bit and sampling the first significant bit from the voice sample. Valid range: 0–3. • Bit[0:2]—Audio gain. Used to pad the end 3 bits of a 13 bit PCM sample. Used by some CODECs (connected to the PCM interface) to allow their gain to be controlled. Valid range: 0–7. • For SPDIF: • Bit[4:31]—Not used • Bit[3]—Set the reporting mode for the SPDIF Rx channel Status. • Bit[2]—Set the SPDIF Tx channel B status equal to SPDIF Tx channel A status. • Bit[1]—Set the SPDIF Tx channel status word value. • Bit[0]—Set the SPDIF Rx in auto rate detect mode. <enable_auto_rate> (Auto-adjust digital rate to audio sampling rate) • ON—Enable. Digital <rate> is automatically set to the current Bluetooth audio sampling rate. • OFF—Disable	
AUTOCONN	**Auto-connection at Power On**
Details: Enable/disable automatic connection at device power-on. Automatic connection can be enabled for either all devices in the Paired Device List (PDL) or a specific remote address defined by the REMOTE_ADDR configuration. **Notes:** If automatic connection is enabled, the maximum number of connection attempts is specified in the MAX_REC configuration. **Usage:** • Format: **AUTOCONN=<type>** Purpose: Disable auto-connection, or enable auto-connection to all devices in PDL or a specific remote address. **Default:** AUTO_CONN=0 (Auto-connection disabled) **Reboot required:** Yes **Parameters:** <type> (Auto-connection type) • 0—(Default) OFF. Auto-connection disabled. • 1—Auto-connect to all devices in the Paired Device List (PDL) • 2—Auto-connect to specific remote address specified in REMOTE_ADDR configuration.	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
AUTO_DATA	**Automatically Enter Data Mode**

Details: Configure module to automatically enter Data mode when BLE and/or SPP connections are established.

Notes: If the module is already in Data Mode when the connection is established, this configuration has no effect.

Usage:

- Format: **AUTO_DATA=<ble> <spp>**
  Purpose: Automatically enter data mode (if currently in command mode).

Default: AUTO_DATA=0 0
(Auto-entry to Data Mode disabled for both BLE and SPP connections.)

Reboot required: No

Parameters:

<ble> (Automatically enter Data Mode when BLE connection is established)
- 0—(Default) OFF
- 1—ON

<spp> (Automatically enter Data Mode when SPP connection is established)
- 0—(Default) OFF
- 1—ON

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BA_CONFIG	**Broadcast Audio Mode**

Details: (Note: This configuration applies to BA releases only.)
Configure device as Broadcaster or Receiver, or disable Broadcast Audio feature.

Versions: Available for Melody 7.1 and later

Notes: (Melody 7.1) BLE is disabled when Broadcast Audio is enabled (device is in Broadcaster or Receiver mode.

Usage:

- Format (Melody 7.1):
  BA_CONFIG=<mode> <enable_auto_broadcast> <broadcaster_product_id> <broadcaster_version_id>
  Purpose: Broadcast Audio configuration—Set Broadcast Audio mode and (in broadcaster mode) configure broadcaster settings.
- Format (Melody 7.2 and later):
  BA_CONFIG=<mode> <enable_auto_broadcast>
  Purpose: Broadcast Audio configuration—Set Broadcast Audio mode and enable/disable auto-broadcast.

Default: (Melody 7.1)
BA_CONFIG=0 ON 0A02 0304
(Broadcast Audio enabled, auto-broadcast enabled, product id=0A02, version id = 0304)

(Melody 7.2 and later)
BA_CONFIG=0 OFF
(Broadcast Audio enabled, auto-broadcast disabled)

Reboot required: Yes

Parameters:

<mode> (Broadcast Audio mode)
- 0—(Default) Normal (Broadcast Audio disabled)
- 1—Broadcaster
- 2—Receiver

<enable_auto_broadcast> (Enable/disable auto-broadcast at power on or after a successful association)
- Value applies to Broadcaster mode only
- ON—(Default Melody 7.1) Auto-broadcast enabled
- OFF—(Default Melody 7.2 and later) Auto-broadcast disabled

BALANCE	Left–Right Audio Balance

Details: Specify left channel/right channel volume percentages.

Usage:

- Format: **BALANCE=<left_channel> <right_channel>**
  Purpose: Set the volume for left and right channels.

Default: BALANCE=100 100
(Left and right channels at full volume (100%))

Reboot required: No

Parameters:

<left_channel> (Left channel volume as percentage)
- Valid range: 0–100 (integer values only)

<right_channel> (Right channel volume as percentage)
- Valid range: 0–100 (integer values only)

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BATT_CONFIG	**Battery Configuration**

Details: Configure the battery behavior.

Usage:

- Format: **BATT_CONFIG=<enable_charge> <critical_level> <charge_level> <temp_level> <current>**
  Purpose: Configure the battery.

Default: BATT_CONFIG=OFF 145 4250 1500 150
(Charging disabled, critical battery level=2900 mV, charge level=4250 mV, thermistor voltage limit=1500 mV, charging current=150 mA)

Reboot required: Yes

Parameters:

<enable_charge> (Enable/disable battery charging)
- OFF—(Default) Charging disabled—Should always be OFF when not using the battery.
- ON—Charging enabled

<critical_level> (Critical (minimum) battery level, in 20 mV increments)
- Default: 145 (2900 mV)
- Module remains off until charged above <critical_level>. (During use, module shuts off if charge drops below <critical_level>.)

<charge_level> (Maximum charging level, in mV)
- Default: 42500
- Charging stops when battery reaches <charge_level>

<temp_level> (Thermistor voltage limit, in mV)
- Default: 1500
- Charging stops if thermistor voltage reaches <temp_level>

<current> (Charging current, in mA)
- Default: 150
- Valid range: 0–200 (0—Disabled; >0—Charging current)

BC_SMART_CONFIG	BC Smart Profile Configuration

Details: Configure BC Smart Profile service and characteristic UUIDs, and enable/disable remote commands.

Usage:

- Format: **BC_SMART_CONFIG=<service_uuid> <characteristic_uuid> <enable_command>**
  Purpose: Configure BC Smart profile UUIDs and enable/disable remote commands.

Default: BC_SMART_CONFIG=68E3 28F0 89F7 D93C ON
(Service UUID=68E3 28F0; Characteristic UUID=89F7 D93C; remote commands enabled)

Reboot required: Yes

Parameters:

<service_uuid> (BC Smart service UUID (last 2 words))
- Hexadecimal format
- e.g. 68E3 28F0

<characteristic_uuid> (BC Smart Data characteristic UUID (last 2 words))
- Hexadecimal format
- e.g. 89F7 D93C

<enable_command> (Enable/disable remote commands)
- OFF—Remote commands disabled
- ON—(Default) Remote commands enabled

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BEACON_DATA	**BLE Beacon Configuration**

Details: Configure BLE beacon type and associated parameters.
BEACON_DATA is used by the module to generate default advertising data when BLE_CONFIG's <adv_mode> parameter is set to Beacon mode (non-connectable advertising).

Usage:

- Format: **BEACON_DATA=<type> <data>**
  Purpose: Configure the BLE beacon.

Default: BEACON_DATA=0 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 04 D2 16 2E 33
(iBeacon; Proximity UUID=00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF; Major=04 D2; Minor=16 2E; Tx Power=EE)

Reboot required: Yes

Parameters:

<type> (Beacon type)
- 0—iBeacon
- 1—Eddystone UID
- 2—Eddystone URL

<data> (Beacon data (<type>-dependent). See iBeacon/Eddystone specifications for details)
- Length: 21 bytes (maximum. Actual length is <type>-dependent.)
- For <type>=0 (iBeacon)
  <proximity_uuid> <major> <minor> <tx_power>
  - <proximity_uuid>
    - Hexadecimal format, 16 bytes
  - <major>
    - Hexadecimal format, 2 bytes
  - <minor>
    - Hexadecimal format, 2 bytes
  - <tx_power>
    - Hexadecimal format, 1 byte
- For <type>=1 (Eddystone UID)
  <namespace> <instance> <tx_power>
  - <namespace>
    - 10 bytes
  - <instance>
    - 6 bytes
  - <tx_power>
    - Hexadecimal format, 1 byte
  - <unused>
    - 4 bytes (any values)
    - Must be entered to complete the <data> packet (21 bytes)
- For <type>=2 (Eddystone URL)
  <tx_power> <prefix> <encoded_URL>
  - <tx_power>
    - Hexadecimal format, 1 byte
  - <prefix>
    - Hexadecimal format, 1 byte
  - <encoded_URL>
    - Hexadecimal format, 17 bytes

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BLE_CONFIG	**BLE Configuration**

Details: Configure BLE parameters.

Usage:

- Format: **BLE_CONFIG=<adv_mode> <enable_auto_adv> <local_mtu> <enable_private_address>**
  Purpose: Configure BLE parameters.

Default: BLE_CONFIG=0 OFF 80 ON
(Normal advertising mode; auto-advertising disabled; MTU=80, BLE private address enabled)

Reboot required: Yes, for <adv_mode> and <enable_private_address> values). Changes to <enable_auto_adv> and <local_mtu> take effect immediately.

Parameters:

<adv_mode> (Advertising mode)
- 0—Normal (connectable undirected)
- 1—Beacon (non-connectable)

<enable_auto_adv> (Enable/disable auto-advertising)
- OFF—(Default) Disabled
- ON—Enabled. Module generates a random address (different from the public permanent address) at power-on for BLE connections
- Note: To view the public and private addresses, use the VERSION command or the LOCAL_ADDR configuration.

<local_mtu> (Maximum transmission unit used for GATT MTU Exchange request/response)
- Valid range: 23–158
- Note: To view the negotiated MTU value resulting from the GATT MTU Exchange, use the STATUS command.

<enable_private_address> (Enable/disable BLE private address)
- OFF—Disabled
- ON—(Default) Enabled

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BLE_CONN_PARAMS	**BLE Advertising, Connection, and Scanning Parameters**

Details: Configure BLE advertising, connection, and scanning parameters. (To enable/disable BLE advertising, use the BLE_CONFIG configuration.)

Notes: All parameter values are decimal format.

Usage:

- Format: **BLE_CONN_PARAMS=<scan_interval> <scan_window> <conn_interval_min> <conn_interval_max> <conn_latency> <conn_supervision_timeout> <conn_attempt_timeout> <adv_interval_min> <adv_interval_max> <conn_latency_max> <supervision_timeout_min> <supervision_timeout_max>**

  Purpose: Configure BLE advertising, connection, and scanning parameters.

Default: BLE_CONN_PARAMS= 128 12 24 40 0 400 100 400 400 64 400 400
(Scan interval=80 ms; scan window=7.5 ms; connection interval=30–50 ms; connection latency=0 events; connection supervision timeout = 400 ms; connection attempt timeout=100 ms; min advertising interval=400 ms; max advertising interval=400 ms; connection latency max=64 events, supervision timeout min and max=400 ms)

Reboot required: Yes

Parameters:

<scan_interval> (Interval between beginning of one LE scan and beginning of next LE scan, in 0.625 ms increments)
- Valid range: 4–16384 (2.5–10240 ms)
- Default: 128 (80 ms)

<scan_window> (LE scan duration)
- Valid range: 4–16384 (2.5–10240 ms)
- Must be ≤ <scan_interval>
- Default: 12 (7.5 ms)

<conn_interval_min> (Minimum connection event interval, in 1.25 ms increments)
- Valid range: 6–3200 (7.5–4000 ms)
- Default: 24 (30 ms)
- Must be ≤ <conn_interval_max>

<conn_interval_max> (Maximum connection event interval, in 1.25 ms increments)
- Valid range: 6–3200 (7.5–4000 ms)
- Default: 24 (30 ms)
- Must be ≥ <conn_interval_min>

<conn_latency> (Connection slave latency, in number of connection events)
- Valid range: 0–500
- Default: 40

<conn_supervision_timeout> (Timeout before disconnecting when no communication is present on lower layers, in 10 ms increments)
- Valid range: 10–3200 (100 ms–32 s)
- Default: 400 (4 s)

<conn_attempt_timeout> (Time to wait for connection to fully establish, in 0.625 ms increments)
- Valid range: 1–65535 (0.625 ms – 40959 s)
- Default: 100 (TBD s)
- Note: Changing this value can reduce the number of successful connections.

(Continued on next page)

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BLE_CONN_PARAMS (continued)	
<adv_interval_min> (Minimum advertising interval for non-directed advertising) • Valid range: 32–16384 • Must be ≤ <adv_interval_max> • Note: Melody does not support directed advertisements. <adv_interval_max> (Maximum advertising interval for non-directed advertising) • Valid range: 32–16384 • Must be ≥ <adv_interval_min> <conn_latency_max> (Maximum allowed slave latency that is accepted if slave requests connection parameter update once connected,) • Valid range: 0–500 <supervision_timeout_min> (Minimum allowed supervision timeout that is accepted if slave requests connection parameter update once connected) • Valid range: 10–3200 (100 ms to 32 s) • Must be ≤ <supervision_timeout_max> <supervision_timeout_max> (Maximum allowed supervision timeout that is accepted if slave requests connection parameter update once connected) • Valid range: 10–3200 (100 ms to 32 s) • Must be ≥ <supervision_timeout_min>	
BT_STATE_CONFIG	**Bluetooth Classic State Configuration**
Details: Configure Bluetooth Classic state at power-on. **Notes:** • The configuration (connectable, discoverable) can be immediately overwritten (no reboot required) using the BT_STATE command. • A connectable/discoverable module becomes automatically not connectable/not discoverable when a connection is established unless it is configured to be always connectable/discoverable. **Usage:** • Format: **BT_STATE_CONFIG=<connectable_mode> <discoverable_mode>** Purpose: Configure Bluetooth Classic power-on state. **Default:** BT_STATE_CONFIG=0 0 (Not connectable or discoverable at power-on) **Reboot required:** Yes **Parameters:** <connectable_mode> • 0—(Default) Not connectable at power-on • 1—Always connectable • 2—Connectable at power-on <discoverable_mode> • 0—(Default) Not discoverable at power-on • 1—Always discoverable • 2—Discoverable at power-on	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
BT_VOL_CONFIG	**Bluetooth Classic Volume Configuration**
Details: Configure default volume settings and related parameters. **Usage:** • Format: **BT_VOL_CONFIG=<default_hfp_volume> <default_a2dp_volume> <a2dp_volume_steps> <volume_scaling_method>** Purpose: Configure default volume settings and related parameters. **Default:** BT_VOL_CONFIG=A 60 10 1 (Default HFP volume=A (10); default A2DP volume=60%; # of A2DP volume steps=10; DSP volume scaling) **Reboot required:** Yes **Parameters:** <default_hfp_volume> (Default HFP (hands-free unit) volume, in range 0–F) • Hexadecimal format • Valid range: 0–F • Default: A <default_a2dp_volume> (Default A2DP (Advanced Audio Distribution Profile) volume, as a decimal percentage) • Integer value • Valid range: 0–100 • Default: 60 <a2dp volume steps> (Number of steps for A2DP volume) • Hexadecimal format • Valid range: 1–255 • Default: 10 <volume_scaling_method> (Volume scaling method) • Value should be set to match the audio type (analog/digital) • 0—Hardware. Volume scaling applied at the DAC for analog output. • 1—(Default) DSP. Volume scaling applied in the digital domain by the DSP.	
CLASS_1	**Enable/Disable GPIO Control of Class 1 Device Using External PA**
Details: Enable/disable use of GPIOs to control external power amplifier (PA) of Class 1 device. **Usage:** • Format: **CLASS_1=<enable>** Purpose: Enable/disable GPIO control of PA. **Default:** CLASS_1=OFF (GPIOs not used to control external PA) **Reboot required:** Yes **Parameters:** <enable> (Enable/disable PIO 0 and PIO 1) • OFF—(Default) Disabled • ON—Enabled. PIO 0 and PIO 1 are used to control an external PA for a Class 1 device	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
CMD_TO	**Guard Time for Escape Sequence ($$$$)**
Details: Set the escape sequence guard time. The guard time is the 'blank' time between the last character and the escape sequence '$$$$'. The guard time also applies after the escape sequence for the command to be recognized. **Usage:** • Format: **CMD_TO=<value>** Purpose: Set the escape sequence guard time. **Default:** CLASS_1=20 (Guard time—400 ms (<value> × 20 ms)) **Reboot required:** No **Parameters:** <value> (Guard time, in 20 ms increments) • Valid range: 0–255 (integer values only) • Default: 20 (400 ms)	
COD	**Class of Device**
Details: Set the device class. **Usage:** • Format: **COD=<value>** Purpose: Set the device class. **Default:** COD=240404 (Wearable headset device) **Reboot required:** No **Parameters:** <value> (Device class) • Hexadecimal value • Common examples: • 200404—Audio headphones • 240404—Wearable headset device	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
CODEC	**A2DP Optional Codecs**
Details: Enable/disable optional A2DP codecs and A2DP Talkback feature. (The SBC codec is mandatory and always enabled, regardless of any optional codecs.) **Usage:** • Format: **CODEC=<codec_bitmask>[<enable_talkback>]** Purpose: Enable/disable A2DP codecs, and optionally enable/disable A2DP Talkback feature. **Default:** CODEC=0 OFF (No optional codecs enabled, and A2DP Talkback feature disabled.) Note: For HD builds (BC127–HD), default is CODEC=8 OFF. **Reboot required:** Yes **Parameters:** <codec_bitmask> (Optional codec bitmask) • Decimal value • Bit 0—AAC • Bit 1—aptX • Bit 2—aptX Low Latency. Not available on HD builds. • Bit 3—aptX HD. Available only on HD builds (for BC127–HD) • Example: 3=AAC and aptX enabled • Note: A valid aptX license key is required to use aptX, aptX Low Latency, or aptX HD codecs. <enable_talkback> (Enable/disable A2DP Talkback feature) • OFF—(Default) Disabled • ON—Enabled • Note: Valid aptX and cVc license keys are required to use the A2DP Talkback feature.	
DEEP_SLEEP	**Enable/Disable Deep Sleep**
Details: Enable/Disable Deep Sleep. **Usage:** • Format: **DEEP_SLEEP=<enable>** Purpose: Enable/disable Deep Sleep. **Default:** DEEP_SLEEP=OFF (Deep Sleep disabled) **Reboot required:** No **Parameters:** <enable> (Enable/disable Deep Sleep) • OFF—(Default) Disabled • ON—Enabled	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
DEVICE_ID	**Device ID Profile Configuration**
Details: Configure Device ID profile. **Notes:** For more information about the parameters for this command, refer to the Device ID profile specification on the Bluetooth adopted specification page—https://www.bluetooth.com/specifications/profiles-overview. **Usage:** • Format: **DEVICE_ID=<vendor_source_id> <vendor_id> <product_id> <bcd_version> <software_version>** Purpose: Configure device ID profile details. **Default:** DEVICE_ID=0001 0002 0003 0004 0005 0006 0007 0008 **Reboot required:** Yes **Parameters:** <vendor_source_id> (Vendor ID source) • 2 bytes <vendor_id> (Vendor ID) • 2 bytes <product_id> (Product ID) • 2 bytes <bcd_version> (BCD version) • 2 bytes <software_version> (Software version) • 8 bytes	
ENABLE_BATT_IND	**Enable/Disable iOS Battery Indication**
Details: Enable/disable iOS battery indications, which are sent over HFP. **Notes:** This value must be set before pairing with the iOS device. **Usage:** • Format: **ENABLE_BATT_IND=<enable>** Purpose: Enable/disable iOS battery indications. **Default:** ENABLE_BATT_IND=ON (iOS battery indications enabled) **Reboot required:** No **Parameters:** <enable> (Enable/disable iOS battery indications) • OFF—Disabled • ON—(Default) Enabled	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
ENABLE_LED	**Enable/Disable LEDs**
Details: Enable/disable the module's LEDs. **Usage:** • Format: **ENABLE_LED=<enable>** Purpose: Enable/disable the module's LEDs. **Default:** ENABLE_LED=ON (LEDs enabled) **Reboot required:** No **Parameters:** <enable> (Enable/disable LEDs) • OFF—Disabled • ON—(Default) Enabled	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
ENABLE_SPP_SNIFF	**SPP Sniff Mode Configuration**

Details: Enable/disable SPP sniff mode, and configure parameters. SPP sniff mode allows the module to use low power modes.

Notes: This configuration takes effect if SPP is the only connected profile.

Usage:

- Format: **ENABLE_SPP_SNIFF=<state> <min_interval> <max_interval> <attempt> <timeout> <duration>**
  Purpose: Enable/disable SPP sniff mode, and configure the feature.

Default: ENABLE_SPP_SNIFF=OFF 0 0 0 0 0
(SPP sniff mode disabled, all parameters set to 0)

Reboot required: No

Parameters:

<state> (Enable/disable SPP sniff mode)
- OFF—(Default) Disabled
- ON—Enabled

<min_interval> (Minimum sniff period interval, 0.625 ms increments)
- Decimal value
- Valid range: 2–65534 (1.25–40.9 ms)
  Note: Even numbers only. (e.g. 2, 4, ..., 65532, 65534)
- Must be ≤ <max_interval>

<max_interval> (Maximum sniff period interval, 0.625 ms increments)
- Decimal value
- Valid range: 2–65534 (1.25–40.9 ms)
  Note: Even numbers only. (e.g. 2, 4, ..., 65532, 65534)
- Must be ≥ <min_interval>

<attempt> (Number of baseband receive slots for sniff attempt, 1.25 ms increments)
- Decimal value
- Valid range: 1–32767 (1.25–40.9 ms)

<timeout> (Number of baseband receive slots for sniff timeout, 1.25 ms increments)
- Decimal value
- Valid range: 1–32767 (1.25–40.9 ms)

<duration> (Length of time to stay in sniff mode, in seconds)
- Decimal value
- Valid range: 0–65535

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
GPIO_CONFIG	**GPIO Configuration**

Details: Enable/disable GPIO control, and indicate reportable events.
For GPIO details, refer to GPIO Functionality.

Notes: If an event is selected in the <pio4_event_bitmask>, PIO 4 is raised when the event occurs (only if GPIO control is disabled).
To clear the event, use the PIO command (“PIO 4 OFF”).
This can be useful when the module is in Data mode and the notifications are not received on the UART.

Usage:

- Format: **GPIO_CONFIG=<gpio_control> <pio4_event_bitmask>**
  Purpose: Enable/disable PIO 4 control, and its associated reportable events.

Default: GPIO_CONFIG=ON 0
(PIO 4 enabled; no events selected)

Reboot required: Yes

Parameters:

<gpio_control> (Enable/disable GPIO control)
- OFF—Disabled
- ON—(Default) Enabled

<pio4_event_bitmask> (PIO 4 event bitmask)
- Hexadecimal value
- Bit 0—A2DP_STREAM_START notification received
- Bit 1—A2DP_STREAM_SUSPEND notification received
- Bit 2—AVRCP_PLAY notification received
- Bit 3—AVRCP_PAUSE or AVRCP_STOP notification received
- Bit 4—CALL_INCOMING notification received
- Bit 5—CALL_OUTGOING notification received
- Bit 6—LINK_LOSS received
- Bit 7—Data mode exited
- Bit 8—PIO 4 high if connected, low if not connected
- Bit 9—AVRCP_MEDIA notification received
- Bit 10—CALL_END or CALL_ACTIVE notification received

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
HFP_CONFIG	**HFP/AGHFP Profile Configuration**

Details: Configure HFP/AGHFP profile parameters. This configuration applies to HFP and AGHFP roles.

Usage:

- Format: **HFP_CONFIG=<enable_cvc> <enable_wbs> <enable_inband> <enable_nrec> <enable_at_command> <ag_uses_hsp>**

  Purpose: Configure all HFP/AGHFP profile parameters.

Default: HFP_CONFIG=OFF ON OFF OFF OFF OFF
(cVc disabled, Wide Band Speech enabled, in-band ringing disabled, noise reduction disabled, unparsed AT commands disabled, Audio gateway uses HFP)

Reboot required: Yes

Parameters:

<enable_cvc> (Enable/disable clear Voice capture)
- OFF—(Default) Disabled
- ON—Enabled

<enable_wbs> (Enable/disable Wide Band Speech)
- OFF—Disabled
- ON—(Enabled) Enabled
  Note: Whether Wide Band Speech is used or not depends on the results of the codec negotiation between the hands-free device and the audio gateway device.

<enable_inband> (Enable/disable in-band ringing for incoming calls)
- OFF—(Default) Disabled
- ON—Enabled

<enable_nrec> (Set/unset NREC flag to tell AG to not use/use internal Noise Reduction/Echo Canceler algorithm))
- OFF—(Default) Disabled. AG should use internal Noise Reduction/Echo Canceler algorithm.
- ON—Enabled. AG should not use internal Noise Reduction/Echo Canceler algorithm.
  This option is recommended if using cVc or an external noise reduction platform.

<enable_at_command> (Enable/disable unparsed AT commands)
- OFF—(Default) Disabled
- ON—Enabled

<ag_uses_hsp> (Profile used by Audio Gateway)
- OFF—(Default) HFP
- ON—HSP

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
HIGH_SPEED	**High Speed Configuration**

Details: Enable/disable SPP and/or iAP2 High Speed. High Speed allows faster data transfer in Data mode. Note that when high speed is being used:

- Audio is disabled because the DSP is used to speed up the transfer.
- Escape sequence to exit Data mode is disabled. Use PIO 5 (if GPIO control is disabled using GPIO_CONFIG—see GPIO Functionality for details) to exit Data mode, or close the connection (from the remote device).

Usage:

- Format: **HIGH_SPEED=<enable_spp> <enable_iap>**
  Purpose: Enable or disable SPP and/or iAP2 High Speed.

Default: HIGH_SPEED=OFF OFF
(SPP high speed disabled; iAP2 high speed disabled)

Reboot required: No

Parameters:

<enable_spp> (Enable/disable SPP High Speed)

- OFF—(Default) Disabled
- ON—Enabled

<enable_iap> (Enable/disable iAP2 High Speed)

- OFF—(Default) Disabled
- ON—Enabled. Different iAP parameters should be used for iAP2 High Speed. See IAP_PARAMS.

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
IAP_PARAMS	**IAP Link Parameters**

Details: Configure iAP link parameters.

Notes: iAP1 support can be enabled/disabled using the <disable_iap1> parameter. iAP2 is always supported.

Usage:

- Format: **IAP_PARAMS=<max_nb_pkt> <max_pkt_len> <disable_iap1> <iap2_features_bitmask>**
  Purpose: Configure iAP link parameters.

Default: IAP_PARAMS=5 64 OFF 3
(Max packets=5; max packet length=64 bytes; iAP1 support enabled; iAP2 App Launch and HID Headset Remote enabled)

Recommended settings for iAP2 High Speed:
IAP_PARAMS=4 750 OFF 3
(Max packets=5; max packet length=64 bytes; iAP1 support enabled; iAP2 App Launch and HID Headset Remote enabled)

Reboot required: Yes

Parameters:

<max_nb_pkt> (Maximum number of packets that may be sent without receiving an acknowledgement)
- Decimal value
- Valid range: 1–16
- Default: 5

<max_pkt_len> (Maximum packet length, in bytes)
- Decimal value
- Valid range: 24–768
- Default: 64

Note: Modifying the maximum number of packets (<max_nb_pkt>) or packet length (<max_pkt_len>) may improve data throughput, but could also lead to data loss or the module freezing if not enough memory is available.

<disable_iap1> (Enable/disable iAP1 support)
- OFF—(Default) iAP1 support enabled
- ON—iAP1 support disabled

<iap2_features_bitmask> (Bitmap—Supported iAP2 features)
- Decimal value
- Valid range: 0–3
- 1-byte bitmap:
  - Bit[0]—App Launch
  - Bit[1]—HID Headset Remote

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
LOCAL_ADDR	**Bluetooth addresses**
Details: Contains the local Bluetooth/BLE addresses of the device. **Notes:** • Read only. GET will read the addresses, but SET cannot be used to change them. • The BLE address is different from the public Bluetooth address if the module is configured to use a randomly-generated BLE private address (see BLE_CONFIG). **Usage:** • Format: **LOCAL_ADDR=<public_bdaddr> <ble_bdaddr>** **Default:** n/a. **Reboot required:** n/a **Parameters:** <public_bdaddr> (Local Bluetooth address (permanent)) • 12-digit hexadecimal format (e.g. 20FABB112233) <ble_bdaddr> (Local BLE address) • 12-digit hexadecimal format (e.g. 5B0E550FF8FC) • By default, BLE private random address is enabled (see BLE_CONFIG). If BLE random address id disabled, <ble_bdaddr> is the same as <public_bdaddr>.	
MAX_REC	**Maximum Reconnection Attempts**
Details: Set the maximum number of reconnection attempts (number of times that the module tries to reconnect to a device in PDL during auto-connection). **Usage:** • Format: **MAX_REC=<value>** Purpose: Set the maximum number of reconnection attempts. **Default:** MAX_REC=2 **Reboot required:** Yes **Parameters:** <value> (Max number of reconnection attempts) • Decimal value • Valid range: 0–255 • Default: 2	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
MM	**Music Manager Enhancements**

Details: Enable, disable, and configure Music Manager Enhancements.

Notes: The Music Manager configuration can also be modified using the MM_CFG command. Please contact QTIL for more information on the Music Manager configuration.

Usage:

- Format: **MM=<enable_mm>[<enable_usr_eq> <bank> <enable_bass_boost> <enable_3d_enhance> <enable_compander> <enable_dither> <enable_speaker_eq>**

  Purpose: Enable/disable Music Manager enhancements, and configure the enhancements.

Default: MM=OFF OFF 0 OFF OFF OFF OFF OFF
(All enhancements off)

Reboot required: No

Parameters:

<enable_mm> (Enable/disable use of Music Manager enhancements.)
- OFF—(Default) Disabled
- ON—Enabled. Individual enhancements are enabled/disabled using the remaining configuration parameters.

<enable_usr_eq> (Enable/disable user equalizer block)
- OFF—(Default) Disabled
- ON—Enabled

<bank> (Equalizer bank
- 0—(Default) No EQ (flat frequency response)
- 1—Bass boost
- 2—Treble Boost
- 3—Rock
- 4—Jazz

<enable_bass_boost> (Enable/disable bass boost block)
- OFF—(Default) Disabled
- ON—Enabled

<enable_3d_enhance> (Enable/disable 3D enhancement block)
- OFF—(Default) Disabled
- ON—Enabled

<enable_compander> (Enable/disable compander block)
- OFF—(Default) Disabled
- ON—Enabled

<enable_dither> (Enable/disable post-processing/dither block)
- OFF—(Default) Disabled
- ON—Enabled

<enable_speaker_eq> (Enable/disable speaker equalization block)
- OFF—(Default) Disabled
- ON—Enabled

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
MUSIC_META_DATA	**Enable/Disable AVRCP Metadata**
Details: Enable/disable AVRCP metadata. **Usage:** • Format: **MUSIC_META_DATA=<enable_meta_data>** Purpose: Enable/disable AVRCP metadata. **Default:** MUSIC_META_DATA=OFF **Reboot required:** No **Parameters:** <enable_meta_data> (Enable/disable AVRCP meta data) • OFF—(Default) Disabled • ON—Enabled	
MUSIC_OLD_AVRCP	**Switch to AVRCP v1.0**
Details: Switch back and forth between AVRCP v1.0 and v1.6. **Notes:** AVRCP Meta Data and Absolute Volume features are not supported by v1.0. **Usage:** • Format: **MUSIC_OLD_AVRCP=<enable_old_avrcp>** Purpose: Enable/disable AVRCP meta data. **Default:** MUSIC_META_DATA=OFF (Use AVRCP v1.6) **Reboot required:** Yes **Parameters:** <enable_old_avrcp> (Select AVRCP version) • OFF—(Default) AVRCP 1.6 • ON—AVRCP 1.0	
NAME	**Device Name**
Details: Set the long name used for the device. To set the short name, use the NAME_SHORT configuration. **Usage:** • Format: **NAME=<name>** Purpose: Set the device's long name. **Default:** NAME=BC-000180 **Reboot required:** Yes **Parameters:** <name> (Long name of device) • ASCII format • Length: 1–32 characters • Note: Last 6 characters are the LAP portion of the device's Bluetooth address. For example, address 20FABB000ABC has LAP=0x000ABC.	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
NAME_SHORT	**Device Short Name**
Details: Set the short name used for the device. To set the long name, use the NAME configuration. **Usage:** • Format: **NAME_SHORT=<short_name>** Purpose: Set the device's short name. **Default:** NAME_SHORT=BC000180 **Reboot required:** Yes **Parameters:** <short_name> (Short name of device) • ASCII format • Length: 1–8 characters • Note: Last 6 characters are the LAP portion of the device's Bluetooth address.	
PIN	**PIN Code**
Details: Set the module's PIN code. **Usage:** • Format: **PIN=<code>** Purpose: Set the module's PIN code. **Default:** PIN=0000 **Reboot required:** Yes **Parameters:** <name> (PIN code for module) • ASCII format • Length: 4 characters	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
PROFILES	**Bluetooth Profiles**

Details: Configure the maximum number of connections for each profile type.

Usage:

- Format: **PROFILES=<max_hfp> <max_aghfp> <max_a2dp_snk> <max_a2dp_src> <max_avrcp> <max_ble> <max_spp> <max_pbap> <max_hid_dev> <max_hid_host> <max_map> <max_iap>**

  Purpose: Set the maximum number of connections for each profile type.

Default: PROFILES=2 0 2 0 2 3 2 1 0 0 1 0
(Note: On MFI builds, <max_iap> default is 1.)

Reboot required: Yes

Parameters:

<max_hfp> (Maximum number of HFP connections)
- 0–3

<max_aghfp> (Maximum number of AGHFP connections)
- 0–3

<max_a2dp_snk> (Maximum number of A2DP Sink/TWS connections)
- 0–3
- Note: Only one of <max_a2dp_snk> and <max_a2dp_src> can be enabled—the other must be disabled (0).

<max_a2dp_src> (Maximum number of A2DP Source connections)
- 0–3
- Note: Only one of <max_a2dp_snk> and <max_a2dp_src> can be enabled—the other must be disabled (0).

<max_avrcp> (Maximum number of AVRCP connections)
- 0–3

<max_ble> (Maximum number of BLE connections)
- 0–3

<max_spp> (Maximum number of SPP connections)
- 0–3

<max_pbap> (Maximum number of PBAP connections)
- 0–2

<max_hid_dev> (Maximum number of HID Device connections)
- 0–1
- Note: Only one of <max_hid_dev> and <max_hid_host> can be enabled, the other must be disabled (0).

<max_hid_host> (Maximum number of HID Host connections)
- 0–1
- Note: Only one of <max_hid_dev> and <max_hid_host> can be enabled, the other must be disabled (0).

<max_map> (Maximum number of MAP connections)
- 0–2

<max_iap> (Maximum number of iAP connections)
- 0–2 on MFI build
- Supported on MFI build only; default=1

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
REMOTE_ADDR	**Auto-connection Remote Address**
Details: Set the auto-connection remote address, which is related to the AUTOCONN and MAX_REC configurations. **Usage:** • Format: **REMOTE_ADDR=<bdaddr>** Purpose: Set the auto-connection remote address. **Default:** REMOTE_ADDR=000000000000 **Reboot required:** Yes **Parameters:** <bdaddr> (Auto-connection remote address) • 12-digit hexadecimal format • If not all zeroes (000000000000), must be the Bluetooth address of an already paired device.	
SPP_UUID	**SPP Profile UUID**
Details: Set the SPP profile's UUID (Universally Unique Identifier). **Usage:** • Format: **SPP_UUID=<uuid>** Purpose: Set the SPP profile's UUID. **Default:** SPP_UUID=00 00 11 01 00 00 10 00 80 00 00 80 5F 9B 34 FB **Reboot required:** Yes **Parameters:** <uuid> (SPP UUID) • Hexadecimal format, 16 bytes	
SSP_CAPS	**Secure Simple Pairing I/O Capabilities**
Details: Configure 'Secure Simple Pairing' (SSP) I/O capabilities. **Usage:** • Format: **SSP_CAPS=<io_caps>** Purpose: Configure the SSP I/O capabilities. **Default:** SSP_CAPS=3 (Just works (no display, no keyboard)) **Reboot required:** No **Parameters:** <io_caps> (SSP I/O capabilities) • 0—Display only • 1—Display, Yes/No • 2—Keyboard only • 3—No display, no keyboard ("Just works") • 4—Display and keyboard (Note: Not supported by standard) • 5—Reject any SSP request • Note: MITM (Man In The Middle) is enabled when <io_caps> is 0, 1, 2, or 4. In this case, PAIR_PASSKEY notifications are received and the user must accept or reject pairing using the PASSKEY command.	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
TWS_CONFIG	**True Wireless Stereo Configuration**

Details: Enable/disable True Wireless Stereo (TWS) and configure TWS routing parameters.

Notes:
- The routing configuration of the TWS device with the highest Bluetooth address is applied to both devices.
- Changing the audio routing mode will have no effect on TWS devices that have already been paired. This routing choice is only stored on initial pairing.
- Permitted routing mode combinations (<master_routing> and <slave_routing>):
  - Master stereo, Slave stereo
  - Master left, Slave right
  - Master right, Slave left
  - Master downmix, Slave downmix

Usage:

- Format: **TWS_CONFIG=<enable_autoconn> <master_routing> <slave_routing>**
  Purpose: Configure the TWS parameters.

Default: TWS_CONFIG=OFF 1 2
(TWS auto-connect disabled; master routing through left channel; slave routing through right channel)

Reboot required: No

Parameters:

<enable_autoconn> (Enable/disable TWS auto-connect when the A2DP source connects)
- OFF—(Default) Disabled
- ON—Enabled

<master_routing> (Audio routing of the device with the highest Bluetooth address)
- 0—Stereo
- 1—(Default) Left channel
- 2—Right channel
- 3—Downmix

<slave_routing> (Audio routing of the device with the lowest Bluetooth address)
- 0—Stereo
- 1—Left channel
- 2—(Default) Right channel
- 3—Downmix

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
UART_CONFIG	**UART Configuration**
Details: Configure UART interface parameters. **Usage:** • Format: **UART_CONFIG=<baudrate> <enable_flow_control> <parity>** Purpose: Configure the UART interface's parameters. **Default:** UART_CONFIG=9600 OFF 0 (9600 baud; Hardware flow control disabled; No parity) **Reboot required:** Yes (for <enable_flow_control> change only) **Parameters:** <baudrate> (UART baudrate) • 9600 (Default) • 19200 • 38400 • 57600 • 115200 • 230400 • 460800 • 961200 <enable_flow_control> (Enable/disable hardware flow control) • OFF—(Default) Disabled • ON—Enabled <parity> (UART parity) • 0—None • 1—Odd • 2—Even	
USB_HOST	**Enable USB Host Interface**
Details: Select the module's control interface—enable/disable the USB host interface. When USB host is enabled, UART is disabled; when USB host is disabled, UART is enabled. **Notes:** The BC127 USB Demo app is available on source.sierrawireless.com. The application gives you the ability to control the module over USB. **Usage:** • Format: **USB_HOST=<enable>** Purpose: Enable/disable the USB host interface. **Default:** USB_HOST=OFF (UART interface enabled, USB host interface disabled) **Reboot required:** Yes **Parameters:** <enable> (Enable/disable USB Host interface) • 0—Enable USB host, disable UART • 1—(Default) Disable USB host, enable UART	

Table 4-9: Melody Configurations Details (Continued)

Configuration	Description
VREG_ROLE	**VREG Button Configuration**
Details: Configure the behavior of the VREG button. **Usage:** • Format: VREG_ROLE=<action> Purpose: Configure the VREG button behavior. **Default:** VREG_ROLE=1 (Button is used to power on/off) **Reboot required:** No **Parameters:** <action> (Action initiated by VREG single press for 1 second) • 0—None • 1—Power on/off	

Notifications

Melody uses unsolicited notification messages to notify the host of events in the Bluetooth link. Notifications are received in Command mode only.

The syntax used is:

<event> [<link_ID>] <parameters>

Note: In Data mode, it is possible to configure PIO 4 to be raised when certain events occur (see GPIO_CONFIG), since notifications are not received.

The following table lists all Melody notifications:

Table 4-10: Melody Unsolicited Notifications Summary

Notification	Description	Page
A2DP_STREAM_START	Audio Interface	85
A2DP_STREAM_SUSPEND	A2DP Streaming Suspended	117
ABS_VOL	AVRCP Absolute Volume	117
ASSOCIATION	(Melody 7.0/7.1) Broadcast Audio Association Status	118
ASSOCIATION_IN_PROGRESS	Broadcast Audio Association In Progress	118
AT	AT Command	118
AVRCP_BACKWARD	AVRCP Backward	119
AVRCP_FORWARD	AVRCP Forward	119
AVRCP_MEDIA	AVRCP Metadata	119
AVRCP_PAUSE	AVRCP Pause	119
AVRCP_PLAY	AVRCP Play	120
AVRCP_STOP	AVRCP Stop	120
BA_BROADCASTER_START	Broadcast Audio Started (Broadcaster)	120
BA_BROADCASTER_STOP	Broadcast Audio Stopped (Broadcaster)	120
BA_RECEIVER_START	Broadcast Audio Started (Receiver)	120
BA_RECEIVER_STOP	Broadcast Audio Stopped (Receiver)	120
BC_SMART_CMD	BC Smart Command Received	121
BC_SMART_CMD_RESP	BC Smart Command Response Received	121
BLE_INDICATION	GATT Indication Received	121
BLE_NOTIFICATION	GATT Notification Received	122
BLE_PAIR_ERROR	BLE Pairing Failed	122
BLE_PAIR_OK	BLE Pairing Successful	122

Table 4-10: Melody Unsolicited Notifications Summary (Continued)

Notification	Description	Page
BLE_READ	GATT Read Request	122
BLE_WRITE	GATT Write Received	123
CALL_ACTIVE	HFP/AGHFP Active Call Notification	123
CALL_DIAL	Dial Number Request From Remote HF	123
CALL_END	Call Termination Notification	123
CALL_INCOMING	Incoming Call Notification	124
CALL_MEMORY	Memory Dial Request From HF	124
CALL_OUTGOING	Outgoing Call Notification	124
CALL_REDIAL	Redial Request From HF	124
CALLER_NUMBER	Incoming Call Number	124
CHARGING IN PROGRESS	Charger Currently Charging Battery	125
CHARGING COMPLETE	Battery Charged and Charger in Standby Mode	125
CHARGER DISCONNECTED	Charger Disconnected	125
CLOSE_OK	Disconnection Indication	125
DTMF	DTMF Code Received	125
ERROR	Error Occurred	126
IAP_CLOSE_SESSION	iAP Data Session Closed	126
IAP_OPEN_SESSION	iAP Data Session Opened	126
INBAND_RING	Ring Indicator	126
LINK_LOSS	Link Loss Indication	126
MAP_NEW_MSG	Notification—New Message From MSE	127
OPEN_OK	Connection Indication	127
PAIR_ERROR	Pairing Failed	127
PAIR_OK	Pairing Successful	128
PAIR_PASSKEY	User Action Required to Complete Pairing	128
PAIR_PENDING	Pairing In Progress	128
RECV	Data Received	128
REMOTE_VOLUME	Remote HF Unit Volume Notification	129
ROLE	HCI Role Notification	129
ROLE_OK	HCI Role Switch Successful	129
ROLE_NOT_ALLOWED	HCI Role Switch Not Allowed	129

Table 4-10: Melody Unsolicited Notifications Summary (Continued)

Notification	Description	Page
SCO_OPEN	SCO Link Open	130
SCO_CLOSE	SCO Link Closed	129
SR	Speech Recognition Notification	130

Table 4-11: Melody Unsolicited Notifications Details

Notification	Description
A2DP_STREAM_START	**A2DP Streaming Started**
Details: A2DP media stream has opened. One or more of these notifications may be received when the remote or local side has requested the A2DP media channel state to change. Note: Even though the steam is open, it may not be active. **Format:** **A2DP_STREAM_START <link_ID>** **Parameters:** <link_ID> (Link identifier (A2DP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
A2DP_STREAM_SUSPEND	**A2DP Streaming Suspended**
Details: A2DP media stream has been suspended. One or more of these notifications may be received when the remote or local side has requested the A2DP media channel state to change. Note: The media stream suspend notification may come some time after the active music stream has ended or, depending on the phone implementation, may not come at all. **Format:** **A2DP_STREAM_SUSPEND <link_ID>** **Parameters:** <link_ID> (Link identifier (A2DP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
ABS_VOL	**AVRCP Absolute Volume**
Details: Current absolute volume. **Format:** **ABS_VOL <link_ID> <volume>** **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <volume> (Absolute volume) • Valid range: 0–127 (represents 0–100%)	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
ASSOCIATION	**Broadcast Audio Association Status**
Details: (Note: This notification applies to Melody 7.1 BA releases only.) Broadcast Audio Association notification received from Broadcaster device. **Versions:** Available for Melody 7.1 only **Format:** ASSOCIATION <status> <broadcaster_product_id> <broadcaster_version_id> **Parameters:** <status> (Association status) • 0—Success • 1—Failure <broadcaster_product_id> (Vendor-specific Broadcaster product identifier) • 4-digit hexadecimal value (e.g. 0A02) • Value will be 0000 if association failed <broadcaster_version_id> (Vendor-specific Broadcaster version identifier) • 4-digit hexadecimal value (e.g. 0304) • Value will be 0000 if association failed	
ASSOCIATION_IN_PROGRESS	**Broadcast Audio Association In Progress**
Details: (Note: This notification applies to BA releases only.) Broadcast Audio Association In Progress notification received from remote broadcaster or receiver. **Versions:** Available for Melody 7.1 and later **Format:** ASSOCIATION_IN_PROGRESS <bd_addr> **Parameters:** <bd_addr> (Bluetooth address of remote broadcaster or receiver) • 12-digit hexadecimal format (e.g. 20FABB112233)	
AT	**AT Command**
Details: Remote device sent an AT command or AT reply. **Format:** AT <link_ID> <size> <command> **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <size> (Length of <command>) • Decimal value • Valid range: 1–65535 <command> (AT command) • ASCII string—AT command sent by remote device	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
AVRCP_BACKWARD	**AVRCP Backward**
Details: AVRCP Backward event received from remote device. **Format:** **AVRCP_BACKWARD <link_ID>** **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
AVRCP_FORWARD	**AVRCP Forward**
Details: AVRCP Forward event received from remote device. **Format:** **AVRCP_FORWARD <link_ID>** **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
AVRCP_MEDIA	**AVRCP Metadata**
Details: AVRCP metadata received from remote device. **Format:** **AVRCP_MEDIA <link_ID> <type>:<data>** **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <type> (Metadata type) • “TITLE“ • “ARTIST“ • “ALBUM“ • “NUMBER“ • “TOTAL_NUMBER“ • “GENRE“ • “PLAYING_TIME(MS)“ <data> (Metadata for corresponding <type>) • ASCII string **Example(s):** • AVRCP_MEDIA 11 TITLE:Get Lucky	
AVRCP_PAUSE	**AVRCP Pause**
Details: AVRCP pause event received from remote device. **Format:** **AVRCP_PAUSE <link_ID>** **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
AVRCP_PLAY	**AVRCP Play**
Details: AVRCP play event received from remote device. **Format:** AVRCP_PLAY <link_ID> **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
AVRCP_STOP	**AVRCP Stop**
Details: AVRCP stop event received from remote device. **Format:** AVRCP_STOP <link_ID> **Parameters:** <link_ID> (Link identifier (AVRCP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
BA_BROADCASTER_START	**Broadcast Audio Started (Broadcaster)**
Details: (Note: This notification applies to BA releases only.) Broadcast Audio Started notification received by Broadcaster in response to BROADCAST ON. **Versions:** Available for Melody 7.1 and later **Format:** BA_BROADCASTER_START **Parameters:** None	
BA_BROADCASTER_STOP	**Broadcast Audio Stopped (Broadcaster)**
Details: (Note: This notification applies to BA releases only.) Broadcast Audio Stopped notification received by Broadcaster in response to BROADCAST OFF. **Versions:** Available for Melody 7.1 and later **Format:** BA_BROADCASTER_STOP **Parameters:** None	
BA_RECEIVER_START	**Broadcast Audio Started (Receiver)**
Details: (Note: This notification applies to BA releases only.) Broadcast Audio Started notification received by Receiver. **Versions:** Available for Melody 7.1 and later **Format:** BA_RECEIVER_START **Parameters:** None	
BA_RECEIVER_STOP	**Broadcast Audio Stopped (Receiver)**
Details: (Note: This notification applies to BA releases only.) Broadcast Audio Stopped notification received by Broadcaster. **Versions:** Available for Melody 7.1 and later **Format:** BA_RECEIVER_STOP **Parameters:** None	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
BC_SMART_CMD	**BC Smart Command Received**
Details: BC Smart Command received from remote device. **Format:** **BC_SMART_CMD <link_ID> <size> <cmd>** **Parameters:** <link_ID> (Link identifier (BLE)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <size> (Length of <cmd>) • Decimal value • Valid range: 1–65535 <command> (BC Smart command) • ASCII string	
BC_SMART_CMD_RESP	**BC Smart Command Response Received**
Details: BC Smart Command response received from remote device. **Format:** **BC_SMART_CMD_RESP <link_ID> <size> <cmd_resp>** **Parameters:** <link_ID> (Link identifier (BLE)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <size> (Length of <cmd_resp>) • Decimal value • Valid range: 1–65535 <cmd_resp> (BC Smart command response) • ASCII string	
BLE_INDICATION	**GATT Indication Received**
Details: GATT indication received from peripheral device. **Format:** **BLE_INDICATION <link_ID> <handle> <size> <data>** **Parameters:** <link_ID> (Link identifier (BLE) of peripheral device) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <handle> (Characteristic handle) • Hexadecimal <size> (Length of <data>, in bytes) • Hexadecimal value • Valid range: 1–FFFF <data> (Characteristic value) • Hexadecimal	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
BLE_NOTIFICATION	**GATT Notification Received**
Details: GATT notification received from peripheral device. **Format:** **BLE_NOTIFICATION <link_ID> <handle> <size> <data>** **Parameters:** <link_ID> (Link identifier (BLE) of peripheral device) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <handle> (Characteristic handle) • Hexadecimal <size> (Length of <data>, in bytes) • Hexadecimal value • Valid range: 1–FFFF <data> (Characteristic value) • Hexadecimal	
BLE_PAIR_ERROR	**BLE Pairing Failed**
Details: BLE pairing attempt failed with device at <current_bdaddr>. **Format:** **BLE_PAIR_ERROR <current_bdaddr>** **Parameters:** <current_bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233)	
BLE_PAIR_OK	**BLE Pairing Successful**
Details: BLE pairing attempt succeeded with device at <current_bdaddr>. **Format:** **BLE_PAIR_OK <current_bdaddr>[<public_bdaddr>]** **Parameters:** <current_bdaddr> (BLE current address) • 12-digit hexadecimal format (e.g. 20FABB112233) <public_bdaddr> (Permanent Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 270FCB143233)	
BLE_READ	**GATT Read Request**
Details: GATT read request received from a central device. Use BLE_READ_RES to respond to this request. **Format:** **BLE_READ <link_ID> <handle>** **Parameters:** <link_ID> (Link identifier (BLE) of the central device) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <handle> (Characteristic handle) • Hexadecimal	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
BLE_WRITE	**GATT Write Received**
Details: GATT write indication received from central device. **Format:** **BLE_WRITE <link_ID> <handle> <size> <data>** **Parameters:** <link_ID> (Link identifier (BLE) of the central device) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <handle> (Characteristic handle) • Hexadecimal <size> (Length of <data>, in bytes) • Hexadecimal value • Valid range: 1–FFFF <data> (Characteristic value) • Hexadecimal values **Example(s):** • BLE_WRITE 14 A 5 3032217409 with <data>=0x30, 0x32, 0x21, 0x74, 0x09	
CALL_ACTIVE	**HFP/AGHFP Active Call Notification**
Details: Notification of active call on specified HFP/AGHFP link. **Format:** **CALL_ACTIVE <link_ID>** **Parameters:** <link_ID> (Link identifier (HFP or AGHFP) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
CALL_DIAL	**Dial Number Request From Remote HF**
Details: Notification that the HFP connected with the AGHFP on the specified link wants to establish an outgoing call to the specified number. **Format:** **CALL_DIAL <link_ID> <number>** **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <number> (Requested number to dial) • ASCII string. • e.g. +3234324	
CALL_END	**Call Termination Notification**
Details: Call on the specified HFP or AGHFP link has terminated. **Format:** **CALL_END <link_ID>** **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
CALL_INCOMING	**Incoming Call Notification**
Details: Notification that there is an incoming call on the specified HFP/AGHFP link. **Format:** CALL_INCOMING <link_ID> **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
CALL_MEMORY	**Memory Dial Request From HF**
Details: Notification that the HF device connected to the specified AGHFP link wants to establish an outgoing call using memory dialing with the specified memory string. Note: The memory string is AG-specific. **Format:** CALL_MEMORY <link_ID> **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details\	
CALL_OUTGOING	**Outgoing Call Notification**
Details: Notification of an outgoing call from the Audio Gateway (AGHFP). **Format:** CALL_OUTGOING <link_ID> **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
CALL_REDIAL	**Redial Request From HF**
Details: Notification that the HF connected to the specified AGHFP link wants to establish a call to the last number dialed. **Format:** CALL_REDIAL <link_ID> **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
CALLER_NUMBER	**Incoming Call Number**
Details: Notification containing the number of the caller on an incoming call. Notification is received after a CALL_INCOMING notification. **Format:** CALLER_NUMBER <link_ID> <number> **Parameters:** <link_ID> (Link identifier (HFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <number> (Caller number) • TBD	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
CHARGING IN PROGRESS	**Charger Currently Charging Battery**
Details: Notification that the charger is currently charging the battery. **Format:** CHARGING IN PROGRESS **Parameters:** None	
CHARGING COMPLETE	**Battery Charged and Charger in Standby Mode**
Details: Notification that the battery is charged and the charger is in standby mode. **Format:** CHARGING COMPLETE **Parameters:** None	
CHARGER DISCONNECTED	**Charger Disconnected**
Details: Notification that the battery charger has been disconnected. **Format:** CHARGER DISCONNECTED **Parameters:** None	
CLOSE_OK	**Disconnection Indication**
Details: Connection on the specified link has closed. **Format:** CLOSE_OK <link_ID> <profile> **Parameters:** <link_ID> (Link identifier) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <profile> (Profile type) • A2DP • AGHFP • AVRCP • BLE • HFP • HID • IAP • MAP • PBAP • SPP • TWS	
DTMF	**DTMF Code Received**
Details: Notification that the HF connected to the specified AGHFP link requests that the AGHFP send the specified DTMF code to the network. **Format:** DTMF <link_ID> <code> **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <code> (DTMF code) • One ASCII character	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
ERROR	**Error Occurred**
Details: An error occurred. **Format:** **ERROR <error_code>** **Parameters:** <error_code> (Error code) • Refer to Error Codes for possible values.	
IAP_CLOSE_SESSION	**iAP Data Session Closed**
Details: The iAP data session on the specified link has closed. **Format:** **IAP_CLOSE_SESSION <link_ID>** **Parameters:** <link_ID> (Link identifier (iAP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
IAP_OPEN_SESSION	**iAP Data Session Opened**
Details: An iAP data session has opened on the specified link. **Format:** **IAP_OPEN_SESSION <link_ID>** **Parameters:** <link_ID> (Link identifier (iAP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
INBAND_RING	**Ring Indicator**
Details: Ring indicator received periodically when there is an incoming call. **Format:** **INBAND_RING <link_ID>** **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
LINK_LOSS	**Link Loss Indication**
Details: Link loss notification for the specified link. **Format:** **LINK_LOSS <link_ID> <status>** **Parameters:** <link_ID> (Link identifier (HFP or A2DP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <status> (Link loss status) • 0—Recovered • 1—Link loss. Melody will attempt to recover the link and, if it fails, the connection is closed (a CLOSE_OK notification will be received).	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
MAP_NEW_MSG	**Notification—New Message From MSE**
Details: Notification that a new message from MSE has been received on the specified link. **Format:** **MAP_NEW_MSG <link_ID> <size_report> <report>** **Parameters:** <link_ID> (Link identifier (MAP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <size_report> (Size of <report> parameter) • Decimal value • Valid range: 1–65535 <report> (Message content) • ASCII string	
OPEN_OK	**Connection Indication**
Details: A connection has been established for the specified profile. **Format:** **OPEN_OK <link_ID> <profile> <bdaddr>** **Parameters:** <link_ID> (Link identifier) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <profile> (Profile type) • A2DP • AGHFP • AVRCP • BLE • HFP • HID • IAP • MAP • PBAP • SPP • TWS <bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233)	
PAIR_ERROR	**Pairing Failed**
Details: Pairing attempt with remote device at <bdaddr> has failed. **Format:** **PAIR_ERROR <bdaddr>** **Parameters:** <bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233)	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
PAIR_OK	**Pairing Successful**
Details: Pairing attempt with remote device at <bdaddr> succeeded. **Format:** **PAIR_OK <bdaddr>** **Parameters:** <bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233)	
PAIR_PASSKEY	**User Action Required to Complete Pairing**
Details: SSP passkey request has been received. Use the PASSKEY command to reply to this notification. **Format:** **PAIR_PASSKEY <bdaddr> <type>[<passkey]** **Parameters:** <bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233) <type> (Pairing Indication Type) • 0—Passkey requested • 1—Confirmation Yes/No requested • 2—Display passkey <passkey> (Passkey to display) • Parameter is not supplied for type 0 (passkey requested)	
PAIR_PENDING	**Pairing In Progress**
Details: Module has started to pair with a remote device. **Format:** **PAIR_PENDING** **Parameters:** None	
RECV	**Data Received**
Details: Data received from a BLE, HID, iAP, or SPP link. **Format:** **RECV <link_ID> <size> <data>** **Parameters:** <link_ID> (Link identifier (BLE, HID, iAP, or SPP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <size> (Length of <data>) • Decimal value • Valid range: 1–65535 <data> (Received data) • ASCII string	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
REMOTE_VOLUME	**Remote HF Unit Volume Notification**
Details: The remote hands-free (HF) unit indicates its new volume. **Format:** **REMOTE_VOLUME <link_ID> <volume>** **Parameters:** <link_ID> (Link identifier (AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details <volume> (Remote HF unit volume) • One-digit hexadecimal format • Valid range: 0–F	
ROLE	**HCI Role Notification**
Details: Notification of the current HCI role (following a role switch by either the module or the remote device). **Format:** **ROLE <bdaddr> <role>** **Parameters:** <bdaddr> (Bluetooth address of remote device) • 12-digit hexadecimal format (e.g. 20FABB112233) <role> (HCI role) • M—Master • S—Slave	
ROLE_OK	**HCI Role Switch Successful**
Details: HCI role switch was successful. **Format:** **ROLE_OK <role>** **Parameters:** <role> (HCI role) • M—Master • S—Slave	
ROLE_NOT_ALLOWED	**HCI Role Switch Not Allowed**
Details: HCI role change failed (remote device refused role change requested by the host). **Format:** **ROLE_NOT_ALLOWED <role>** **Parameters:** <role> (HCI role) • M—Master • S—Slave	
SCO_CLOSE	**SCO Link Closed**
Details: SCO connection has closed and audio is no longer being routed. **Format:** **SCO_CLOSE <link_ID>** **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	

Table 4-11: Melody Unsolicited Notifications Details (Continued)

Notification	Description
SCO_OPEN	**SCO Link Open**
Details: SCO connection has opened and audio is routed. **Format:** **SCO_OPEN <link_ID>** **Parameters:** <link_ID> (Link identifier (HFP or AGHFP)) • 8-bit (two digit) hexadecimal value • See Link ID Management on page 13 for details	
SR	**Speech Recognition Notification**
Details: Notification that speech recognition has detected a word. **Format:** **SR :<type>** **Parameters:** <type> (Word recognized) • YES • NO • Unrecognized word	

5: Bluetooth Classic (BR/EDR)

Melody 7.x supports several Bluetooth Classic profiles, as listed in the following table.

This chapter describes each of these profiles and provides summaries of the Melody commands, configurations and notifications that apply to them.

Table 5-1: Supported Bluetooth Classic Profiles

Profile	Description	Version	Roles Supported
A2DP	Advanced Audio Distribution Profile (A2DP)	1.3	Sink (SNK) and Source (SRC)
AVRCP	Audio/Video Remote Control Profile (AVRCP)	1.6	Target (TG) and Controller (CT)
DI	Device ID Profile (DI)	1.2	N/A
HFP	Hands-Free Profile (HFP)	1.7	Hands-Free unit (HF) and Audio Gateway (AG)
HID	Human Interface Device Profile (HID)	1.1	HID Host and HID device
HSP	Headset Profile (HSP)	1.2	Headset (HS) and Audio Gateway (AG) *Note: AG supported by Melody 7.1 and later.*
IAP	iPod Accessory Protocol (iAP)	R19	Accessory
MAP	Message Access Profile (MAP)	1.1	Messaging Client Equipment (MCE)
PBAP	Phone Book Access Profile (PBAP)	1.1.1	Phone Book Client Equipment (PCE)
SPP	Serial Port Profile (SPP)	1.2	Device A and Device B

Advanced Audio Distribution Profile (A2DP)

Overview

A2DP defines the manner in which high-quality stereo audio can be streamed from one device to another over a Bluetooth connection. For example, A2DP can define the method for streaming music from a mobile phone to wireless headphones.

A2DP supports two roles—A2DP Sink (receiver) and A2DP Source (transmitter). Only one role may be enabled at any time.

Melody provides the best audio quality and supports several codecs:

- SBC—Low-complexity subband codec
- AAC—Advanced Audio Coding
- aptX
- aptX Low Latency
- aptX HD (for BC127 HD only)

Note: aptX codecs require an aptX license key.

Additional A2DP features include True Wireless Stereo (TWS), Dual Stream, and Talkback.

The following table describes the Melody commands, notifications, and configurations available for A2DP profiles:

Table 5-2: A2DP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	MUSIC	AVRCP Music Playback Control	57
	ROUTE	Set/Get Audio Routing	67
	VOLUME	Get/Set Volume	81
Notifications	A2DP_STREAM_START	A2DP Streaming Started	117
	A2DP_STREAM_SUSPEND	A2DP Streaming Suspended	117
Configurations	BALANCE	Left–Right Audio Balance	90
	BT_VOL_CONFIG	Bluetooth Classic Volume Configuration	96
	CODEC	A2DP Optional Codecs	98
	MM	Music Manager Enhancements	107
	PROFILES	Bluetooth Profiles	110
	TWS_CONFIG	True Wireless Stereo Configuration	112

True Wireless Stereo (TWS)

Melody 7.x includes True Wireless Stereo (TWS) support with SBC or aptX codecs.

TWS allows two modules to connect to each other and share audio that is streamed to one of them (A2DP Source, or analog input). Each module can be configured for Left, Right, or Stereo channels using the TWS_CONFIG configuration. The True Wireless Stereo feature allows the use case for TWS speakers over Bluetooth.

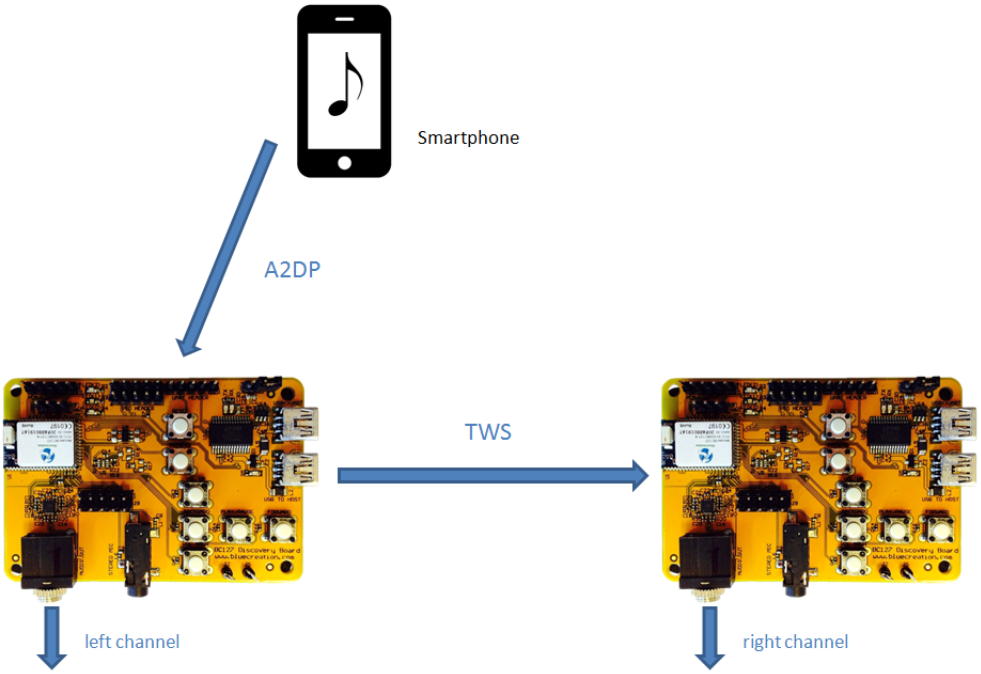


Figure 5-1: True Wireless Stereo

Dual Stream

Melody 7.x includes the Dual Stream feature, which allows audio to be streamed to two A2DP sink devices simultaneously (e.g. most commonly, to two headsets).

To start streaming to two devices, use the ROUTE command with the link IDs of the two A2DP sink devices as parameters (e.g. “ROUTE 10 20”).

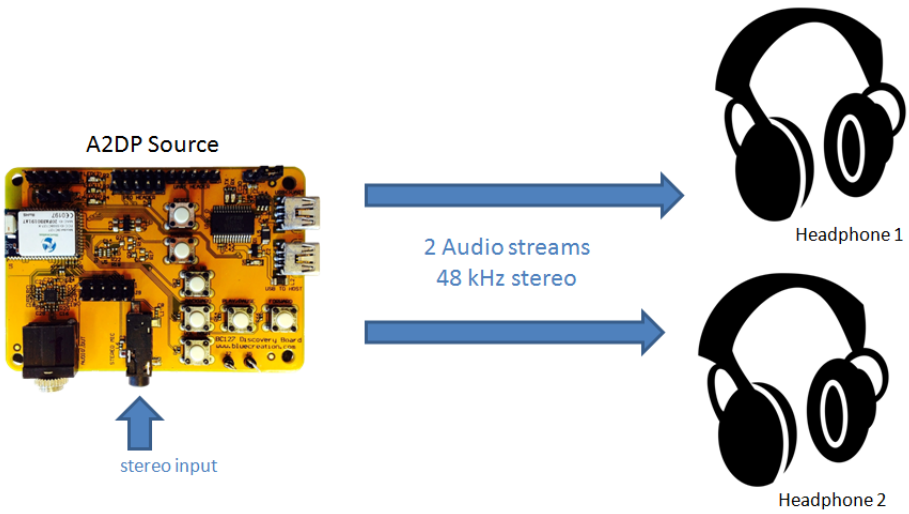


Figure 5-2: Dual Stream

Talkback

The aptX Low Latency codec can provide additional cVc wideband back channel support when enabled in the CODEC configuration.

The back channel is a 16 kHz mono audio stream that is taken from the left analog input of the A2DP Sink device and routed to the left analog output of the A2DP Source device.

Note: This feature requires valid aptX and cVc license keys.

Figure 5-3: Talkback

Audio/Video Remote Control Profile (AVRCP)

AVRCP is designed to provide a standard interface to control TVs, stereo equipment, etc. It is generally used in concert with A2DP to control the music playback status and volume, and to retrieve metadata (e.g. track title, artist name, duration, etc.).

The following table describes the Melody commands, notifications, and configurations available for AVRCP profiles:

Table 5-3: AVRCP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	AVRCP_META_DATA	Set/Request AVRCP Metadata	24
	MUSIC	AVRCP Music Playback Control	57
	VOLUME	Get/Set Volume	81

Table 5-3: AVRCP Commands, Notifications, and Configurations (Continued)

	Name	Description	Page
Notifications	ABS_VOL	AVRCP Absolute Volume	117
	AVRCP_BACKWARD	AVRCP Backward	119
	AVRCP_FORWARD	AVRCP Forward	119
	AVRCP_PAUSE	AVRCP Pause	119
	AVRCP_PLAY	AVRCP Play	120
	AVRCP_STOP	AVRCP Stop	120
Configurations	MUSIC_META_DATA	Enable/Disable AVRCP Metadata	108
	MUSIC_OLD_AVRCP	Switch to AVRCP v1.0	108
	PROFILES	Bluetooth Profiles	110

Device ID Profile (DI)

The Device ID profile allows a device to be identified by specifying the manufacturer, product ID, product version, and version of the Device ID specification.

The following table describes the Melody configuration available for DI profiles:

Table 5-4: DI Configurations

	Name	Description	Page
Configurations	DEVICE_ID	Device ID Profile Configuration	99

Hands-Free Profile (HFP)

HFP is commonly used to allow car hands-free kits to communicate with mobile phones in the vehicle.

HFP supports the following roles:

- Hands-Free unit (HF)
- Audio Gateway (AG)

Melody 7.x supports four codecs for HFP:

- NB—Narrow Band (8 kHz)
- WB—Wide Band (16 kHz)
- cVc NB—Clear Voice Capture Handsfree Narrow Band (8 kHz)
- cVc WB—Clear Voice Capture Handsfree Wide Band (16 kHz)

Each of these codecs uses one mic (left channel).

Note: Clear Voice Capture (cVc) uses an algorithm provided by Qualcomm for echo and noise cancellation, and requires a valid cVc license key.

The following table describes the Melody commands, notifications, and configurations available for HFP profiles:

Table 5-5: HFP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	AT	Send AT Command/Response over HFP	23
	CALL	Manage HFP/AGHFP Call Status	39
	REMOTE_VOLUME	Set Remote Hands-Free Unit Volume	64
	ROUTE	Set/Get Audio Routing	67
	VOLUME	Get/Set Volume	81
Notifications	CALL_ACTIVE	HFP/AGHFP Active Call Notification	123
	CALL_END	Call Termination Notification	123
	CALL_INCOMING	Incoming Call Notification	124
	CALL_MEMORY	Memory Dial Request From HF	124
	CALL_OUTGOING	Outgoing Call Notification	124
	CALL_REDIAL	Redial Request From HF	124
	CALLER_NUMBER	Incoming Call Number	124
	DTMF	DTMF Code Received	125
	SCO_CLOSE	SCO Link Closed	129
	SCO_OPEN	SCO Link Open	130
Configurations	BALANCE	Left–Right Audio Balance	90
	BT_VOL_CONFIG	Bluetooth Classic Volume Configuration	96
	HFP_CONFIG	HFP/AGHFP Profile Configuration	103
	PROFILES	Bluetooth Profiles	110

Human Interface Device Profile (HID)

HID is a lightweight wrapper of the human interface device protocol defined for USB.

HID supports the following roles:

- HID Host
- HID Device

By default, when HID Device is enabled, an HID Keyboard descriptor is loaded. This descriptor can be changed with HID_DESC on page 45.

The following table describes the Melody commands, notifications, and configurations available for HID profiles:

Table 5-6: HID Commands, Notifications, and Configurations

	Name	Description	Page
Commands	HID_DESC	Set HID USB Descriptor	45
	HID_READ	Read HID USB Descriptor	46
	SEND_RAW	Send Raw Data	71
Notifications	RECV	Data Received	128
Configurations	PROFILES	Bluetooth Profiles	110

Headset Profile (HSP)

HSP provides the basic functionality needed for communication between an Audio Gateway (computer, mobile phone, etc.) and a Headset.

HSP supports the following roles:

- Headset (HS)
- Audio Gateway (AG)—AG supported by Melody 7.1 and later.

HSP uses the same profile IDs as HFP.

The following table describes the Melody commands, notifications, and configurations available for HSP profiles:

Table 5-7: HSP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	CALL	Manage HFP/AGHFP Call Status	39
	REMOTE_VOLUME	Set Remote Hands-Free Unit Volume	64
	ROUTE	Set/Get Audio Routing	67
	VOLUME	Get/Set Volume	81
Notifications	CALL_ACTIVE	HFP/AGHFP Active Call Notification	123
	CALL_END	Call Termination Notification	123
	CALL_INCOMING	Incoming Call Notification	124
	CALL_OUTGOING	Outgoing Call Notification	124
	SCO_CLOSE	SCO Link Closed	129
	SCO_OPEN	SCO Link Open	130

Table 5-7: HSP Commands, Notifications, and Configurations (Continued)

	Name	Description	Page
Configurations	BALANCE	Left–Right Audio Balance	90
	BT_VOL_CONFIG	Bluetooth Classic Volume Configuration	96
	HFP_CONFIG	HFP/AGHFP Profile Configuration	103
	PROFILES	Bluetooth Profiles	110

iPod Accessory Protocol (iAP)

Melody supports the iAP2 version of the iAP2 protocol, and has limited support for iAP1 as a fallback option.

Supported iAP features include:

- Multiple EA sessions (based on <max_iap> in the PROFILES configuration)
- App Launch (enable/disable via IAP_PARAMS)
- HID Headset Remote (enable/disable via IAP_PARAMS)

Two EA sessions can be opened simultaneously and data can be transferred in the following modes:

- Command mode—Using SEND or SEND_RAW commands
- Data mode—In this mode, the High Speed feature enables higher transfer speeds (up to 500 kbps from the accessory to the iOS device).
  Note: When High Speed is active, the audio is disabled.

Note: iAP is available on MFI builds only (MFI license). Please contact Sierra Wireless for more information.

The following table describes the Melody commands, notifications, and configurations available for iAP:

Table 5-8: iAP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	$$$$	Exit Data Mode	18
	ENTER_DATA_MODE	Enter Data mode	43
	IAP	Get/Set iAP Identification Parameters	47
	IAP_APP_REQ	iOS App Launch Request	48
	SEND	Send Data	70
	SEND_RAW	Send Raw Data	71
Notifications	IAP_CLOSE_SESSION	iAP Data Session Closed	126
	IAP_OPEN_SESSION	iAP Data Session Opened	126
	RECV	Data Received	128

Table 5-8: iAP Commands, Notifications, and Configurations (Continued)

	Name	Description	Page
Configurations	HIGH_SPEED	High Speed Configuration	104
	PROFILES	Bluetooth Profiles	110

Message Access Profile (MAP)

MAP allows text messages (SMS) to be exchanged between devices.

Melody 7.x supports the Messaging Client Equipment (MC) role.

When MAP is connected, a notification service is registered and the phone notifies Melody when a new message arrives.

The following table describes the Melody commands and notifications available for MAP:

Table 5-9: MAP Commands and Notifications

	Name	Description	Page
Commands	MAP_GET_MSG	Send request to retrieve message	54
Notifications	MAP_NEW_MSG	Notification—New Message From MSE	127
Configurations	PROFILES	Bluetooth Profiles	110

Phone Book Access Profile (PBAP)

PBAP allows a client to access the connected server's contacts list. It is typically used between a car kit and a mobile phone.

Melody 7.x supports the Phone Book Client Equipment (PCE) role.

The following table describes the Melody commands and notifications available for PBAP:

Table 5-10: PBAP Commands and Notifications

	Name	Description	Page
Commands	PB_ABORT	Abort Phonebook Download	60
	PB_PULL	Download phonebook	61
Configurations	PROFILES	Bluetooth Profiles	110

Serial Port Profile (SPP)

SPP emulates a serial cable to provide a simple substitute for existing RS232 serial cable.

SPP enables easy data transfer and sending of remote commands:

- Command mode—Using SEND or SEND_RAW commands and RECV notification.
- Data mode—Data is transferred seamlessly at up to 600 kbps with the High Speed feature enabled (HIGH_SPEED) or 100 kbps if High Speed is disabled.
- Remote commands—Received data ending with a carriage return (<CR>) is processed as a command. A command response is then sent back to the remote device over SPP.

The following table describes the Melody commands, notifications, and configurations available for SPP:

Table 5-11: SPP Commands, Notifications, and Configurations

	Name	Description	Page
Commands	$$$$	Exit Data Mode	18
	ENTER_DATA_MODE	Enter Data mode	43
	SEND	Send Data	70
	SEND_RAW	Send Raw Data	71
Notifications	RECV	Data Received	128
Configurations	ENABLE_SPP_SNIFF	SPP Sniff Mode Configuration	101
	HIGH_SPEED	High Speed Configuration	104
	PROFILES	Bluetooth Profiles	110
	SPP_UUID	SPP Profile UUID	111

6: Bluetooth Low Energy (BLE)

Melody 7.x supports three BLE profiles—GAP, GATT, and BC Smart Profile.

This chapter describes each of these profiles and provides summaries of the Melody commands, configurations and notifications that apply to them.

Generic Access Profile (GAP)

GAP supports the following features:

- Two roles—GAP Peripheral and GAP Central. This allows the module to scan and advertise at the same time.
- Up to 3 simultaneous BLE connections
- Non-connectable advertising for beacons implementations (e.g. iBeacon, Eddystone, etc.)
- BLE bonding with unauthenticated encryption and authenticated encryption

The following table describes the Melody commands and configurations available for GAP profiles:

Table 6-1: GAP Commands and Configurations

	Name	Description	Page
Commands	ADVERTISING	Start/Stop/Configure BLE Advertising	18
	BLE_SECURITY	Start BLE Connection Security	34
	CLOSE	Send Disconnection Request	40
	OPEN	Bluetooth Connection Request	58
	SCAN	Search for BLE Devices	69
	SSRD	Set Scan Response Data	73
	UNPAIR	Unpair Devices	80
Configurations	BEACON_DATA	BLE Beacon Configuration	92
	BLE_CONFIG	BLE Configuration	93
	BLE_CONN_PARAMS	BLE Advertising, Connection, and Scanning Parameters	94
	NAME_SHORT	Device Short Name	109

Generic Attribute Profile (GATT)

GATT supports the following features:

- Two roles—GATT Client and GATT Server
- GAP, GATT, and BC Smart profiles
- Custom profiles implemented using the generic GATT commands

The following table describes the Melody commands, notifications, and configurations available for GATT profiles:

Table 6-2: GATT Commands, Notifications and Configurations

	Name	Description	Page
Commands	BLE_GET_CHAR	GATT Characteristic Discovery	27
	BLE_GET_SERV	GATT Service Discovery	29
	BLE_INDICATION	GATT Indication Request	30
	BLE_NOTIFICATION	GATT Notification Request	31
	BLE_READ	GATT Read Request	32
	BLE_READ_RES	GATT Read Response	33
	BLE_SET_DB	Set Custom GATT Database	35
	BLE_WRITE	GATT Write Request	36
Notifications	BLE_INDICATION	GATT Indication Received	121
	BLE_NOTIFICATION	GATT Notification Received	122
	BLE_READ	GATT Read Request	122
	BLE_WRITE	GATT Write Received	123
Configurations	BLE_CONFIG	BLE Configuration	93

BC Smart Profile

BC Smart Profile is a Sierra Wireless profile that allows the use to send data or remote commands, and is compatible with all Sierra Wireless products (BC118, BX300x).

In Data mode, max throughput is approximately 100 kbps server-to-client, and up to 8 kbps client-to-server (with MTU=120).

Melody Smart (the Android and iOS example app) is also available and supports the BC Smart profile.

The following table describes the Melody commands, notifications, and configurations available for BC Smart Profile:

Table 6-3: BC Smart Profile Commands, Notifications and Configurations

	Name	Description	Page
Commands	$$$$	Exit Data Mode	18
	BC_SMART_COMMAND	Send Remote Command Over BLE	25
	BC_SMART_NOTIF	Enable/Disable BC Smart Notifications	26
	ENTER_DATA_MODE	Enter Data mode	43
	SEND	Send Data	70
	SEND_RAW	Send Raw Data	71
Notifications	BC_SMART_CMD	BC Smart Command Received	121
	BC_SMART_CMD_RESP	BC Smart Command Response Received	121
	RECV	Data Received	128
Configurations	BC_SMART_CONFIG	BC Smart Profile Configuration	91

7: GPIO Functionality

GPIO functionality depends on the values of GPIO_CONFIG parameters.

By default, GPIO control is enabled, allowing the module to work autonomously without the need of a host processor (for example, as in a wireless speaker).

The following tables list the functionality associated with each GPIO.

Table 7-1: GPIO Functionality with GPIO Control Enabled

GPIO	Type	Button timing[a]	Description
VREG	Input	Short	Enter pairing mode (connectable and discoverable)
		Long	Refer to VREG_ROLE.
PIO 0	Input	Short	Volume up
PIO 1	Input	Short	Volume down
PIO 2	Input	Short	• HFP call answer if there is an incoming call, or • HFP call end if there is an outgoing call, or • AVRCP Play/Pause *Note: Not available on MFI builds.*
		Long	HFP call last number redial *Note: Not available on MFI builds.*
PIO 3	Output	N/A	• High when audio is active (e.g. during a call or when streaming music) • Low otherwise *Note: Not available on MFI builds.*
PIO 4	Input	Short	• AVRCP Backward or • Flash (Apple Center)—Only on MFI builds with HID Headset Remote enabled in IAP_PARAMS.
PIO 5	Input	Short	• AVRCP Forward

a. Short—GPIO held and released after 50 ms;
Long—GPIO held and released after 1 s

Table 7-2: GPIO Functionality with GPIO Control Disabled

GPIO	Type	Button timing	Description
PIO 0	Output	N/A	• High if there is at least one connection established (BT Classic or BLE), or • Low otherwise
PIO 4	Output	N/A	Refer to <pio4_event_bitmask> in GPIO_CONFIG.
PIO 5	Input	Rising	Enter Data Mode with the first BLE, iAP, or SPP link available.
		Falling	Exit Data Mode

Restoring the Default Configuration

The default configuration can be restored using either of the following equivalent methods:

- Maintain PIO 0 high (press the VOL UP button on a BC127-DISKIT) while resetting the module
- Use RESTORE.

8: LED Indications

LEDs provide two indication types:

- Event patterns—Pattern plays once only
- State patterns—Patterns repeat.

LED indications are enabled by default (see ENABLE_LED on page 100).

Events

Events are indicated by all three LEDs (LED 0, LED 1, LED 2), as described in the following table:

Table 8-1: Event patterns

Event	LED pattern
Power on	LED 0, LED 1, and LED 2 turned on for 1000 ms
Power off	LED 0, LED 1, and LED 2 quickly flash 3 times

States

Bluetooth Classic State

The Bluetooth Classic state is indicated by LED 0 and LED 1, as described in the following table:

Table 8-2: Bluetooth Classic State Patterns

Classic state	LED pattern
DISCOVERABLE	LED 0 and LED 1 alternate quick flashes (every 200 ms)
CONNECTABLE	LED 1 flashes slowly (every 2400 ms); LED 0 is off
CONNECTED	LED 0 flashes slowly (every 2400 ms); LED 1 is off
IDLE	LED 0 and LED 1 are both off

Note: The Classic state can be simultaneously DISCOVERABLE, CONNECTABLE, and CONNECTED. The LED pattern is shown based on the following priority: DISCOVERABLE > CONNECTABLE > CONNECTED. (e.g. If device is discoverable and connected, the DISCOVERABLE pattern is shown.)

Bluetooth Low Energy (BLE) State

The BLE state is indicated by LED2, as described in the following table:

Table 8-3: BLE State Patterns

BLE state	LED pattern
OFF or IDLE	LED 2 is off
ADVERTISING	LED 2 flashes quickly (every 400 ms)
CONNECTED	LED 2 flashes slowly (every 2400 ms)

Note: The BLE state can be simultaneously ADVERTISING and CONNECTED. The LED pattern is shown based on the following priority: ADVERTISING > CONNECTED.

9: Power Management

This chapter describes Melody's power management-related commands and configurations for battery operation and for reducing power consumption.

Battery Operation

The following battery operation power management features are available:

- Battery configuration—Charging and thresholds are configured using BATT_CONFIG.
- Battery status can be checked using BATTERY_STATUS.
- Battery indications can be sent (when enabled) to iOS devices. See ENABLE_BATT_IND.

Deep Sleep Mode

Deep Sleep is a very low power consumption mode.

To enable or disable deep sleep mode, modify the DEEP_SLEEP configuration.

When deep sleep is enabled:

- If no activity (UART, GPIO, Bluetooth) for 2000 ms, module goes to deep sleep.
- Any data on the UART wakes the module from deep sleep, and the chip wakes up approximately 5 ms later.

  However, the transmission that wakes the module is lost.

  Therefore, it is recommended to send dummy data followed by a Carriage Return (<CR>), which will return ERROR 0x0012, before sending an actual command. This will wake the chip and clear the command buffer.

Link Policy

When the module is connected, it is set up to reduce power consumption while maintaining the connection. It automatically establishes a link policy to allow it to reduce power consumption while still being able to receive link updates.

Note: The link policy can be overwritten using the LINK_POLICY command.

10: Speech Recognition

Melody's Speech Recognition (SR) feature allows the detection of "Yes" or "No" on the mic input.

SR can be activated or deactivated with the SPEECH_REC command. When the feature is activated, SR notifications are received when a word is detected. This allows the user to trigger actions such as answering or rejecting a phone call for example.

Note: Speech Recognition has the highest audio priority. During a call or when music is streaming, the audio won't come out while Speech Recognition is active.

A cVc license key is required to use this feature.

11: Tones

Tones of any pitch can be played using the TONE command. If audio is already active (HFP call or A2DP streaming), the tones are mixed with the audio.

12: Broadcast Audio

The Broadcast Audio feature supports audio broadcasting and receiving between the module and numerous devices within Bluetooth range.

Note: This feature is available only on Melody BA releases (7.1 and later), which do not support aptX Low Latency.

The module can be configured to act as a Broadcaster or as a Receiver.

A Receiver must be associated with a Broadcaster before it can receive audio. The association procedure is required only once.

The audio source to broadcast can be:

- Bluetooth A2DP connection with a device such as a smartphone
- Module's analog input

The Broadcast Audio feature uses the GATT Broadcast Audio service.

BLE support while Broadcast Audio is enabled is version-dependent:

- Melody 7.1—BLE is not supported when Broadcast Audio is enabled.
- Melody 7.2 and later—BLE is supported when Broadcast Audio is enabled, with the following limitations:
  - During the Association procedure (see ASSOCIATION command):
    - A Receiver advertises the GATT Broadcast Audio service, waiting for a Broadcaster to connect.
    - The Broadcaster scans for the advertising Receiver and, when it finds it, a temporary connection is established between the Broadcaster and the Receiver to complete the association.

    Note: Receivers should not modify the advertising data or GATT database. New BLE connections are not allowed during the Association (except for one with a Broadcaster).

    Note: Broadcasters cannot scan or establish new BLE connections during the Association (except for one with a Receiver).

  - When Broadcast Audio is active (see BROADCAST command):
    - The Broadcaster automatically advertises the Broadcast Audio service.
    - Receivers automatically scan for the Broadcaster and, once found, can start receiving audio (BA_RECEIVER_START notification).

    Note: Receivers cannot scan while searching for a Broadcaster (i.e. when Broadcast Audio is on but the BA_RECEIVER_START notification has not been received).

    Note: Broadcasters should not modify the advertising data. The scan response data can still be set.

The following table describes the Melody commands, notifications, and configurations available for Broadcast Audio:

Table 12-1: Broadcast Audio Commands, Notifications, and Configurations

	Name	Description	Page
Commands	ASSOCIATION	Start/Stop Broadcast Audio Association	20
	BROADCAST	Start/Stop Broadcast Audio	37
	ROUTE	Set/Get Audio Routing	67
	STATUS	Return Device Connection Status	74
Notifications	ASSOCIATION_IN_PROGRESS	Broadcast Audio Association In Progress	118
	BA_BROADCASTER_START	Broadcast Audio Started (Broadcaster)	120
	BA_BROADCASTER_STOP	Broadcast Audio Stopped (Broadcaster)	120
	BA_RECEIVER_START	Broadcast Audio Started (Receiver)	120
	BA_RECEIVER_STOP	Broadcast Audio Stopped (Receiver)	120
Configurations	BA_CONFIG	Broadcast Audio Mode	90

A: Error Codes

Table 1-1: Command Errors

Error Code	Description
0x0003	Unknown Error
0x0011	Command not allowed with the current configuration
0x0012	Command not found
0x0013	Wrong parameter
0x0014	Wrong number of parameters
0x0015	Command not allowed in the current state
0x0016	Device already connected
0x0017	Device not connected
0x0018	Command is too long
0x0019	Name not found
0x001A	Configuration not found
0x0100	Failed to read battery voltage
0x1002	Failed to communicate with the Apple MFI Co-processor
0x1004	Failed to register/unregister device
0x1005	BLE request failed
0x1006	Insufficient encryption
0x1007	Insufficient authentication
0x1008	Operation not permitted
0x1009	Invalid handle
0xF00x	Critical Error
0xFF01	Melody license key is missing
0xFF02	Melody license key is invalid

B: Examples

BT Connection and Secure Simple Pairing

The following examples illustrate the secure simple pairing procedure and connection with a Bluetooth Classic profile.

Table 2-1: Bluetooth Classic Inquiry

Description: Search for local Bluetooth devices		
Inquiry	Device A (BC127; BT Addr = 20FABB000160)	Device B (BC127; BT Addr = 20FABB000161)
	--- Search for Bluetooth devices in the area --- INQUIRY	*--- Make B connectable and discoverable ---* BT_STATE ON ON
	PENDING	OK
	INQUIRY 20FABB000161 "BC-000161" 240404 -65db	
	INQUIRY 20FABB000161 "BC-000161" 240404 -64db	
	INQUIRY B88A6054DABB "UNKNOWN" 02010C -62db	
	INQUIRY 00226986D5E9 "UNKNOWN" 140680 -87db	
	INQUIRY 00226986D5E9 "UNKNOWN" 140680 -79db	
	INQUIRY B88A6054DABB "DESKTOP" 02010C -87db	
	....	
	INQUIRY 20FABB000161 "BC-000161" 240404 -70db	
	INQU_OK	

Table 2-2: Connection and Secure Simple Pairing — Just Works (default)

Description: Simply pair and connect a profile (SPP). *Note: The OPEN command initiates pairing if the devices are not already paired.*		
Pairing and connection	Device A (BC127; BT Addr = 20FABB000160)	Device B (BC127; BT Addr = 20FABB000161)
		--- Make B connectable and discoverable --- BT_STATE ON ON
	--- Pairing and SPP connection request --- OPEN 20FABB000161 SPP	OK
	PENDING	
	--- Pairing automatically accepted, (default SSP_CAPS3, Just Works) --- PAIR_PENDING	*--- Pairing automatically accepted, (default SSP_CAPS3, Just Works) ---* PAIR_PENDING
	--- SPP connection with 20FABB000161 established --- PAIR_OK 20FABB000161	*--- SPP connection with 20FABB000160 established ---* PAIR_OK 20FABB000160
	OPEN_OK 15 SPP 20FABB000161	OPEN_OK 15 SPP 20FABB000160
Status	*--- Verify device status ---* STATUS	*--- Verify device status ---* STATUS
	STATE CONNECTED[1] CONNECTABLE[OFF] DICOVERABLE[OFF] [IDLE]	STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE]
	LINK 15 CONNECTED SPP 20FABB000161	LINK 15 CONNECTED SPP 20FABB000160
	OK	OK
	--- Verify device present in Paired Device List --- LIST	*--- Verify device present in Paired Device List ---* LIST
	LIST 20FABB000161 SPP	LIST 20FABB000160 SPP
	OK	OK

Table 2-3: Connection and Secure Simple Pairing — MITM

Description: Simply pair and connect a profile (SPP) with MITM (Man In The Middle) protection. *Note: The OPEN command initiates pairing if the devices are not already paired.*		
I/O capabilities configuration	Device A (BC127; BT Addr = 20FABB000160) *--- I/O caps Display, yes/no ---* SET SSP_CAPS=1 OK	Device B (BC127; BT Addr = 20FABB000161) *--- I/O caps Display, yes/no ---* SET SSP_CAPS=1 OK
Pairing and connection	 *--- Pairing and SPP connection request ---* OPEN 20FABB000161 SPP PENDING *--- User confirmation required—make sure same passkey displayed on both devices ---* PAIR_PENDING PAIR_PASSKEY 20FABB000160 1 960630 *--- Accept pairing (yes) ---* PASSKEY 0 1 OK *--- SPP connection with 20FABB000161 established ---* PAIR_OK 20FABB000161 OPEN_OK 15 SPP 20FABB000161	*--- Make B connectable and discoverable ---* BT_STATE ON ON OK *--- Pair request received. User confirmation required—make sure same passkey displayed on both devices ---* PAIR_PENDING PAIR_PASSKEY 20FABB000161 1 960630 *--- Accept pairing (yes) ---* PASSKEY 0 1 OK *--- SPP connection with 20FABB000160 established ---* PAIR_OK 20FABB000160 OPEN_OK 15 SPP 20FABB000160
Status	*--- Verify device status ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE] LINK 15 CONNECTED SPP 20FABB000161 OK *--- Verify device present in Paired Device List ---* LIST LIST 20FABB000161 SPP OK	*--- Verify device status ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE] LINK 15 CONNECTED SPP 20FABB000160 OK *--- Verify device present in Paired Device List ---* LIST LIST 20FABB000160 SPP OK

Table 2-4: Remote device from Paired Device List (PDL)

Description: Remove a device (Device B, BT Addr = 20FABB000161) from the Paired Device List.	
Unpair	Device A (BC127; BT Addr = 20FABB000160) *--- Show Paired Device List ---* LIST ... LIST 20FABB000161 SPP ... OK *--- Remove device from PDL ---* UNPAIR 20FABB000161 OK

Digital Audio Configuration

The following examples illustrate how to configure digital audio.

Table 2-5: Digital Audio Configuration

Description: Configure Device A as I2S master (default configuration) and Device B as I2S slave. • Bit clock—2.822 MHz (64 × 44.1 kHz) • 16 bits per sample • Left-justified, 1 bit delay		
I2S	Device A (BC127; BT Addr = 20FABB000160) *--- I2S Master (default configuration) ---* SET AUDIO_DIGITAL=0 44100 64 100A00 OK	Device B (BC127; BT Addr = 20FABB000161) *--- I2S Slave ---* SET AUDIO_DIGITAL=0 44100 64 100200 OK
Description: Configure Device A as PCM master and Device B as PCM slave. • Bit clock—1.536 MHz • Word clock—48 kHz		
PCM	Device A (BC127; BT Addr = 20FABB000160) *--- PCM Master ---* SET AUDIO_DIGITAL=1 48000 1536 2100420 OK	Device B (BC127; BT Addr = 20FABB000161) *--- PCM Slave ---* SET AUDIO_DIGITAL=1 48000 1536 100420 OK
Description: Configure Device A and Device B as SPDIF. • Output rate—48 kHz		
SPDIF	Device A (BC127; BT Addr = 20FABB000160) *--- Set output rate ---* SET AUDIO_DIGITAL=2 48000 0 9 OK	Device B (BC127; BT Addr = 20FABB000161) *--- Set output rate ---* SET AUDIO_DIGITAL=2 48000 0 9 OK

A2DP Sink/Source Functionality

The following examples illustrate basic A2DP Sink and Source functionalities.

Table 2-6: A2DP Sink/Source Functionality

Description: Configure Device A as A2DP Sink and Device B as A2DP Source.		
	Device A (BC127; BT Addr = 20FABB000160)	Device B (BC127; BT Addr = 20FABB000161)
Initial configuration	*--- Enable A2DP Sink and AVRCP ---* SET PROFILES=0 0 1 0 1 0 0 0 0 0 0 0 OK WRITE OK RESET Sierra Wireless Copyright 2018 Melody Audio V7.0.0 Build: 1519847376 Ready	*--- Enable A2DP Source and AVRCP ---* SET PROFILES=0 0 0 1 1 0 0 0 0 0 0 0 OK WRITE OK RESET Sierra Wireless Copyright 2018 Melody Audio V7.0.0 Build: 1519847376 Ready
Establish connection	*--- Make Device A connectable and discoverable ---* BT_STATE ON ON OK *--- Connection accepted ---* OPEN_OK 10 A2DP 20FABB000166	 *--- Connection request ---* OPEN 20FABB000161 A2DP PENDING OPEN_OK 10 A2DP 20FABB000162
Status	*--- Check A2DP and AVRCP status ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[OFF] LINK 10 CONNECTED A2DP 20FABB000166 SUSPENDED SBC SNK 48000 LINK 11 CONNECTED AVRCP 20FABB000166 STOPPED OK	*--- Check A2DP and AVRCP status ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[OFF] LINK 10 CONNECTED A2DP 20FABB000162 SUSPENDED SBC SRC 48000 LINK 11 CONNECTED AVRCP 20FABB000162 STOPPED OK

Table 2-6: A2DP Sink/Source Functionality (Continued)

Description: Control music playback.		
Playback control	*--- Play music ---* MUSIC 10 PLAY OK A2DP_STREAM_START 10 AVRCP_PLAY 11 AVRCP_PAUSE 11 A2DP_STREAM_SUSPEND 10	 A2DP_STREAM_START 10 AVRCP_PLAY 11 *--- Pause music ---* MUSIC 10 PAUSE OK AVRCP_PAUSE 11 A2DP_STREAM_SUSPEND 10
Volume control	*--- Raise volume one step ---* VOLUME 10 UP OK ABS_VOL 11 93 ABS_VOL 11 16	 ABS_VOL 11 93 *--- Set volume to step 2 ---* VOLUME 10 2 OK ABS_VOL 11 16
AVRCP meta data	*--- Enable AVRCP metadata ---* SET MUSIC_META_DATA=ON OK *--- Track changed notification received, request metadata and print it ---* AVRCP_MEDIA 11 TITLE: Get Lucky	 *--- Set AVRCP metadata (Title: Get Lucky) ---* AVRCP_META_DATA 11 1 Get Lucky PENDING *--- Notifies that track has changed and wait for remote remote device to request metadata.---* OK

BLE Functionality

The following examples illustrate all basic Bluetooth Low Energy (BLE) functionalities.

Table 2-7: BLE Connection Establishment

Description: Establish a BLE connection.		
	Device A (BC127; BD Addr = 5B6142CC75AB)	Device B (BC127; BD Addr = 636285DC6816)
Scan/ advertise		*--- Start advertising ---* ADVERTISING ON OK
	--- Search for Bluetooth devices in the area --- SCAN 5 PENDING SCAN 28F0765A885F 0 <UNKNOWN> 06 -95db SCAN 636285DC6816 1 <BC000161> 06 -38db SCAN 2C7649C28609 1 <UNKNOWN> 00 -84 db ... SCAN 762A3B9EA219 1 <UNKNOWN> 1A -96dB SCAN_OK	
BLE connection	*--- BLE connection request ---* OPEN 636285DC6816 BLE 1 PENDING OPEN_OK 14 BLE 636285DC6816 *--- BLE connection with 636285DC6816 established ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 636285DC6816 80 OK	*--- Connection accepted ---* OPEN_OK 14 BLE 5B6142CC75AB *--- BLE connection with 5B6142CC75AB established ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 5B6142CC75AB 80 OK

Table 2-8: BC Smart Profile

Description: Send data and remote commands over BLE with the BC Smart profile.		
	Device A (BC127; BD Addr = 5B6142CC75AB)	Device B (BC127; BD Addr = 636285DC6816)
Initial state	*--- Display status (Current state is Connected)---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 63285DC6816 80 OK	*--- Display status (Current state is Connected) ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 5B6142CC75AB 80 OK
Send data in Command mode	*--- Send data to Device B ---* SEND 14 Hello! OK	*--- Receive data from Device A ---* RECV 14 6 Hello!
Send data in Data mode	*--- Enter Data mode, send data to Device B, then exit Data mode ---* ENTER_DATA_MODE 14 OK Everything is sent over BLE now$$$$OK	*--- Receive data from Device A ---* RECV 14 20 Everything is sent o RECV 14 11 ver BLE now RECV 14 1 $ RECV 14 1 $ RECV 14 1 $ RECV 14 1 $
Remote command	*--- Send remote command 'ROUTE 0' to Device B ---* BC_SMART_COMMAND 14 ROUTE 0 OK *--- Command response received 'OK\r' ---* BC_SMART_CMD_RESP 14 3 OK<CR>	*--- Remote command received from Device A ---* BC_SMART_CMD 14 7 ROUTE 0 OK

Table 2-9: BLE Generic Commands and Custom Profile

Description: Set up a custom BLE profile and use the generic BLE commands.		
	Device A (BC127; BD Addr = 43BFD1BD82C8)	Device B (BC127; BD Addr = 554A60A73750)
Custom BLE profile setup		*--- Set customer GATT database ---* BLE_SET_DB 44 PENDING 0002 0118 3005 2003 0005 2A00 D000 6400 0002 0018 PENDING 3005 0207 0000 2A00 D400 3005 0209 0001 2A00 D400 PENDING 0010 F028 E368 62D6 3490 5143 EFAA C64C 2F00 3013 PENDING 180C 003C D9F7 8937 37AA 8F71 4AAD 79E7 E5D1 AA00 PENDING CC01 0000 3053 180E 003C D9F7 8937 37AA 8F71 4AAD PENDING 79E7 E5D1 BB00 CD01 0000 3093 1810 003C D9F7 8937 PENDING 37AA 8F71 4AAD 79E7 E5D1 CC00 CE01 0000 OK
Establish BLE connection		*--- Advertise and accept BLE connection ---* ADVERTISING ON OK
	--- Establish BLE connection --- OPEN 554A60A73750 BLE 1 PENDING OPEN_OK 14 554A60A73750	OPEN_OK 14 43BFD1BD82C8 000A

Table 2-9: BLE Generic Commands and Custom Profile (Continued)

GATT Service and characteristic discovery	*--- Service discovery ---* BLE_GET_SERV 14 PENDING BLE_SERV 14 U16 1801 0001 BLE_SERV 14 U16 1800 0005 BLE_SERV 14 U128 002F4CC6-AAEF-4351-9034-D66268E328F0 000A OK *--- Characteristic discovery ---* BLE_GET_CHAR 14 PENDING BLE_CHAR 14 U16 2A05 0003 BLE_CHAR 14 U16 2A00 0007 BLE_CHAR 14 U16 2A01 0009 BLE_CHAR 14 U128 AAD1E537-79AD-4A71-8FAA-373789F7D93C 000C BLE_CHAR 14 U128 BBD1E5E7-79AD-4A71-8FAA-373789F7D93C 000E BLE_CHAR 14 U128 CCD1E5E7-79AD-4A71-8FAA-373789F7D93C 0010 OK	
GATT Write characteristic value request	*--- Write characteristic value (handle: 0x000E, size: 0xA, value: 0123456789 ---* BLE_WRITE 14 000E A PENDING {30}{30}{30}{30}{30}{30}{30}{30}{30}{30}OK	*--- Write request received & automatically accepted ---* BLE_WRITE 14 000C A 30313233343536373839 OK

Table 2-10: BLE Security (Bonding)

Description: Bond two BLE devices (SSP_CAPS=3 Just Works)		
	Device A (BC127; BD Addr = 55C42978DEAC)	Device B (BC127; BD Addr = 4DD675EE1D26)
Initial state	*--- Display permanent address and BLE private address (randomly generated) ---* GET LOCAL_ADDR LOCAL_ADDR=20FABB000160 55C42978DEAC OK *--- BLE connection is established ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 4DD675EE1D26 80 OK *--- Devices are no longer paired ---* LIST OK	*--- Display permanent address and BLE private address (randomly generated) ---* GET LOCAL_ADDR LOCAL_ADDR=20FABB000161 4DD675EE1D26 OK *--- BLE connection is established ---* STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[CONNECTED] LINK 14 CONNECTED BLE 55C42978DEAC 80 OK *--- Devices are no longer paired ---* LIST OK
BLE bonding	*--- Start encryption request and pairing ---* BLE_SECURITY 14 PENDING PAIR_PENDING PAIR_OK 20FABB000161 BLE_PAIR_OK 4DD675EE1D26 20FABB000161 BLE_SECURITY 14 0 *--- Devices are now paired and BLE link is encrypted ---* LIST LIST 20FABB000161 BLE OK	*--- Accept BLE pairing request ---* PAIR_PENDING PAIR_OK 20FABB000160 *--- Devices are now paired and BLE link is encrypted ---* LIST LIST 20FABB000160 BLE OK

Table 2-11: Beacon Setup

Description: Set up a BLE beacon and switch advertising mode to Beacon mode.	
Set Beacon data	*--- Set iBeacon data:* *• Proximity UUID: 0x00112233-4455-6677-8899-AABBCCDDEEFF* *• Major: 0x04D2 (1234)* *• Minor: 0x162E (5678)* *• Tx Power: 0xEE (-18 dBm) ---* SET BEACON_DATA=0 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 04 D2 16 2E EE OK
	--- Set Eddystone UID: *• Namespace: 0x00112233445566778899* *• Instance: 0xAABBCCDDEEFF* *• Tx Power: 0xEE (-18 dBm)* *• Note: Final four bytes are not used. ---* SET BEACON_DATA=1 00 11 22 33 44 55 66 77 99 AA BB CC DD EE FF EE 00 00 00 00 OK
	--- Set Eddystone URL: *• Tx Power: 0xEE (-18 dBm)* *• Prefix: 0x02 (http://)* *• Encoded URL: 0x676F6F2E676C2F495A304F5141 (goo.gl/IZ0OQA)* *• Note: Final six bytes are not used. ---* SET BEACON_DATA=2 EE 02 67 6F 6F 2E 67 6C 2F 49 5A 30 4F 51 41 00 00 00 00 00 00 OK
Advertising	*--- Set advertising mode to Beacon mode ---* SET BLE_CONFIG=1 OFF 80 OFF OK *--- Store the configuration ---* WRITE OK RESET Sierra Wireless Copyright 2018 Melody Audio V7.0.0 Build: 1519847376 *--- Start advertising ---* ADVERTISING ON OK

HFP Functionality

The following examples illustrate all basic HFP functionalities.

Table 2-12: HFP—Incoming Call

Description: Make an incoming call.		
	BC127 (HF)	BC127 (AG)
Notify incoming call		*--- Notify incoming call to HF ---* CALL 12 INCOMING "123456"
		OK
	CALL_INCOMING 13	CALL_INCOMING 12
	CALLER_NUMBER 13 123456	
	CALLER_NUMBER 13 123456	
	CALLER_NUMBER 13 123456	
	...	
HF answers call	CALL 13 ANSWER	
	OK	
	CALL_ACTIVE 13	CALL_ACTIVE 12
	SCO_OPEN 13	SCO_OPEN 12
HF ends call	CALL 13 END	
	OK	
	SCO_CLOSE 13	SCO_CLOSE 12
	CALL_END 13	CALL_END 12

Table 2-13: HFP—Outgoing Call

Description: Make an outgoing call from AG following a redial request from the HF.		
HF requests redial	BC127 (HF) CALL 13 REDIAL OK	BC127 (AG) CALL_REDIAL 12
AG notifies outgoing call to HF	 CALL_OUTGOING 13 SCO_OPEN 13	CALL 12 OUTGOING 123456 OK SCO_OPEN 12 CALL_OUTGOING 12
AG notifies call answered	 CALL_ACTIVE HFP 13	CALL 12 ANSWER OK CALL_ACTIVE 12
Call termination	 SCO_CLOSE 13 CALL_END 13	CALL 12 END OK SCO_CLOSE 12 CALL_END 12

Table 2-14: HFP—Call Transfer

Description: Transfer audio to the HF or AG during a call.		
Call transfer to AG	BC127 (HF) SCO_CLOSE 13	BC127 (AG) CALL 12 TRANSFER OK SCO_CLOSE 12
Call transfer to HF	 SCO_OPEN 13	CALL 12 TRANSFER OK SCO_OPEN 12

Table 2-15: HFP—Remote Volume

Description: Set HF speaker gain from the AG.		
HF sets speaker gain and notifies AG	BC127 (HF) VOLUME 13 5 OK	BC127 (AG) REMOTE_VOLUME 12 5
AG sets HF speaker gain	VOLUME 13 13 HFP 7 OK	REMOTE_VOLUME 12 7 OK

Table 2-16: HFP—Custom AT Commands

Description: Send custom AT commands over HFP.		
Enable AT commands	BC127 (HF) SET HFP_CONFIG=OFF ON OFF OFF ON OFF OK	BC127 (AG) SET HFP_CONFIG=OFF ON OFF OFF ON OFF OK
Execute AT command	AT 13 AT+ABCD=1,2,3,4 OK	AT 12 15 AT+ABCD=1,2,3,4
AT command response	AT 13 2 OK	AT 12 OK OK

HID Functionality

The following examples illustrate basic HID functionalities with keyboard (default) and mouse descriptors.

Table 2-17: HID Keyboard Functionality

Description: Configure a keyboard, connect to the host, and send HID reports when keys are activated.		
Initial configuration	BC127 (HID Host) *--- Enable HID Host ---* SET PROFILES=0 0 0 0 0 0 0 0 0 1 0 0 OK WRITE OK RESET OK	BC127 (HID Device) *--- Enable HID Device ---* SET PROFILES=0 0 0 0 0 0 0 0 1 0 0 0 OK SET COD=000540 OK WRITE OK RESET OK
Read remote HID descriptor	*--- Read remote HID descriptor (default) ---* HID_READ 20FABB000161 PENDING HID_READ 74 {09}{02}{06}{35}{45}{35}{43}{08}{22}{25}{3f}{05}{01}{09}{06}{a1}{01}{05}{07}{19}{e0}{29}{e7}{15}{00}{25}{01}{75}{01}{95}{08}{81}{02}{95}{01}{75}{08}{81}{03}{95}{05}{75}{01}{05}{08}{19}{01}){05}'{02}{95}{01}{75}{03}{91}{03}{95}{06}{75}{08}{15}{00}{25}{65}{05}{07}{19}{00}{29}{65}{81}{00}{c0} OK	*--- Set device state connectable and discoverable ---* BT_STATE ON ON OK

Table 2-17: HID Keyboard Functionality (Continued)

Establish connection	*--- Issue connection request ---* OPEN 20FABB000161 HID PENDING OPEN_OK 17 HID 20FABB000161	*--- Connection request accepted ---* OPEN_OK 17 HID 20FABB000160
Send HID report	*--- HID reports received ---* RECV 17 8 {00}{00}{04}{00}{00}{00}{00}{00} RECV 17 8 {00}{00}{04}{00}{00}{00}{00}{00}	*--- Send HID report ('a' key pressed) ---* SEND_RAW 17 8 PENDING {00}{00}{04}{00}{00}{00}{00}{00}OK *--- Send HID report ('a' key pressed) ---* SEND_RAW 17 8 PENDING {00}{00}{04}{00}{00}{00}{00}{00}OK

Table 2-18: HID Mouse Functionality

Description: Configure a mouse, connect to the host, and send HID reports when mouse actions are performed.		
Initial configuration	BC127 (HID Host) *--- Enable HID Host ---* SET PROFILES=0 0 0 0 0 0 0 0 0 1 0 0 OK WRITE OK RESET OK	BC127 (HID Device) *--- Enable HID Device ---* SET PROFILES=0 0 0 0 0 0 0 0 1 0 0 0 OK SET COD=000580 OK WRITE OK RESET OK

Table 2-18: HID Mouse Functionality (Continued)

Set HID descriptor	*--- Read remote HID descriptor (default) ---* HID_READ 20FABB000161 PENDING HID_READ 82 {09}{02}{06}{35}{4d}{35}{4b}{08}{22}{25}{47}{05}{01}{09}{02}{A1}{01}{09}{01}{A1}{00}{05}{09}{19}{01}{29}{08}{15}{00}{25}{01}{95}{08}{75}{01}{81}{02}{95}{00}{81}{03}{05}{01}{09}{30}{09}{31}{16}{01}{F8}{26}{FF}{07}{75}{0C}{95}{02}{81}{06}{09}{38}{15}{81}{25}{7F}{75}{08}{95}{01}{81}{06}{05}{0C}{0A}{38}{02}{95}{01}{81}{06}{C0}{C0} OK	*--- Set example mouse descriptor ---* HID_DESC 71 PENDING {05}{01}{09}{02}{A1}{01}{09}{01}{A1}{00}{05}{09}{19}{01}{29}{08}{15}{00}{25}{01}{95}{08}{75}{01}{81}{02}{95}{00}{81}{03}{05}{01}{09}{30}{09}{31}{16}{01}{F8}{26}{FF}{07}{75}{0C}{95}{02}{81}{06}{09}{38}{15}{81}{25}{7F}{75}{08}{95}{01}{81}{06}{05}{0C}{0A}{38}{02}{95}{01}{81}{06}{C0}{C0}OK *--- Set state connectable and discoverable ---* BT_STATE ON ON OK
Establish connection	*--- Issue connection request ---* OPEN 20FABB000161 HID PENDING OPEN_OK 17 HID 20FABB000161	*--- Connection request accepted ---* OPEN_OK 17 HID 20FABB000160
Send HID report	*--- HID reports received ---* RECV 17 6 {00}{00}{30}{00}{00}{00} RECV 17 6 {01}{00}{00}{00}{00}{00} RECV 17 6 {00}{00}{00}{00}{00}{00}	*--- Send HID report (mouse moved vertically) ---* SEND_RAW 17 6 PENDING {00}{00}{30}{00}{00}{00}OK *--- Send HID report (left button pressed) ---* SEND_RAW 17 6 PENDING {01}{00}{00}{00}{00}{00}OK *--- Send HID report (left button released) ---* SEND_RAW 17 6 PENDING {00}{00}{00}{00}{00}{00}OK

SPP—Send/Receive Data

The following examples demonstrate how to send and receive data with the Melody Classic app.

Table 2-19: SPP Send/Receive

Description: Connect a smartphone and send/receive data.	
Connection	BC127 (Name = BC-000170) *---Set state connectable and discoverable ---* BT_STATE ON ON OK *--- A2DP, AVRCP and HCP are opened by the phone (from the Bluetooth settings) ---* OPEN_OK 13 HFP 1C56FEBB9C9C OPEN_OK 11 AVRCP 1C56FEBB9C9C OPEN_OK 10 A2DP 1C56FEBB9C9C *--- SPP is opened by the phone (from Melody Classic app) ---* OPEN_OK 10 A2DP 1C56FEBB9C9C

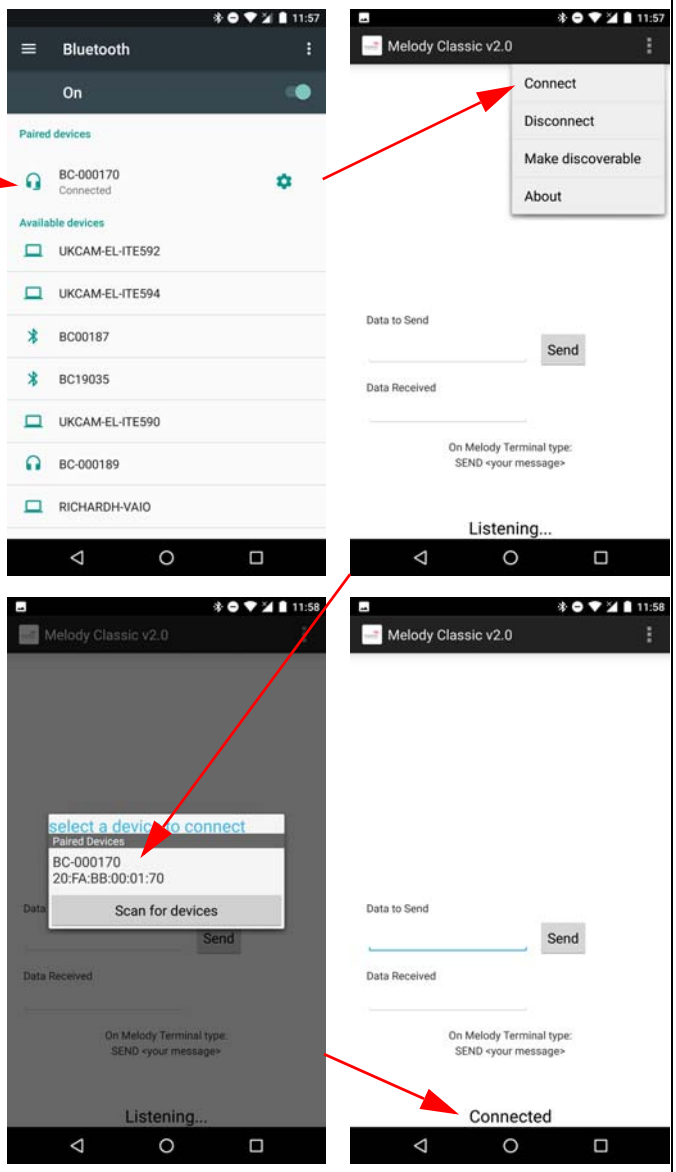


Table 2-19: SPP Send / Receive (Continued)

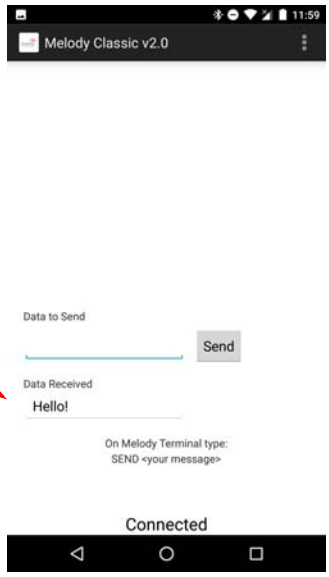
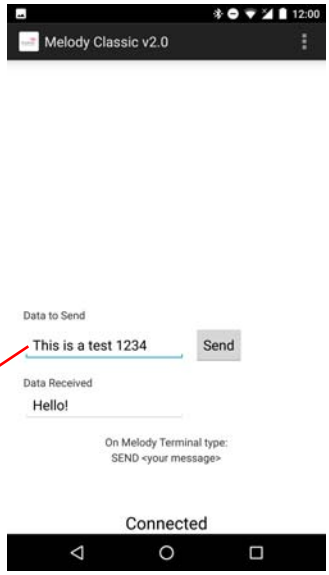
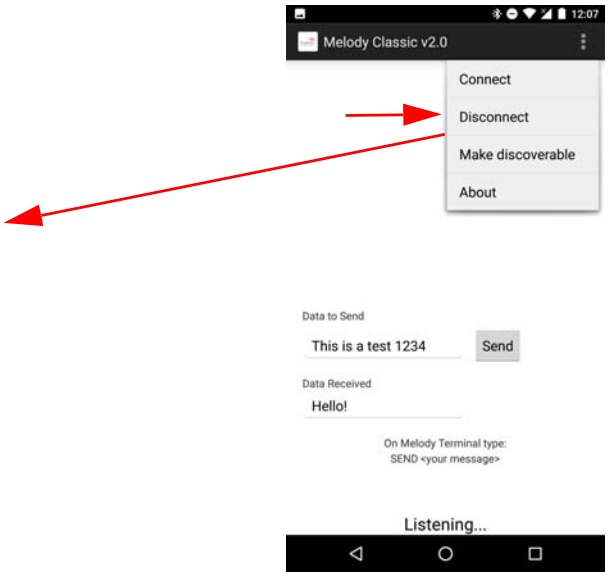
Send data	SEND 15 Hello! OK 
Receive data	 RECV 15 19 This is a test 1234

Table 2-19: SPP Send/Receive (Continued)

Disconnect	CLOSE_OK 15 SPP 

TWS Functionality

The following example describes how to set up a TWS connection between a phone playing music and two BC127 devices. Both BC127 modules are in A2DP Sink mode (default).

Table 2-20: TWS Functionality

Description: Connect BC127 devices to a smartphone (BD addr = 1C56FEBB9C9C) that is playing music.		
Connection with phone	BC127 (Device A; BD Addr = 20FABB000170)	BC127 (Device B; BD Addr = 20FABB000172)
	BT_STATE ON ON	
	OK	
	PAIR_PENDING	
	PAIR_OK 1C56FEBB9C9C	
	OPEN_OK 13 HFP 1C56FEBB9C9C	
	OPEN_OK 10 A2DP 1C56FEBB9C9C	
	OPEN_OK 11 AVRCP 1C56FEBB9C9C	

Table 2-20: TWS Functionality (Continued)

TWS connection		*(Set device state connectable and discoverable)* BT_STATE ON ON OK
	--- Search for TWS devices --- INQUIRY 5 4 ON PENDING INQUIRY 20FABB000172 "BC-000172" 240404 -47db ... INQUIRY 20FABB000172 "BC-000172" 240404 -44db INQU_OK	
	--- TWS connection request --- OPEN 20FABB000172 TWS PENDING	
	PAIR_PENDING	PAIR_PENDING
	PAIR_OK 20FABB000172	PAIR_OK 20FABB000170
	OPEN_OK 2A TWS 20FABB000172	OPEN_OK 1A TWS 20FABB000170
	OPEN_OK 21 AVRCP 20FABB000172	OPEN_OK 11 AVRCP 20FABB000170
Check status	STATUS STATE CONNECTED[2] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE] LINK 10 CONNECTED A2DP 1C56FEBB9C9C SUSPENDED SBC SNK 44100 LINK 11 CONNECTED AVRCP 1C56FEBB9C9C PAUSED LINK 13 CONNECTED HFP 1C56FEBB9C9C IDLE NB LINK 21 CONNECTED AVRCP 20FABB000172 STOPPED LINK 2A CONNECTED TWS 20FABB000172 SUSPENDED SBC SRC 0 OK	STATUS STATE CONNECTED[1] CONNECTABLE[OFF] DISCOVERABLE[OFF] BLE[IDLE] LINK 11 CONNECTED AVRCP 20FABB000170 STOPPED LINK 1A CONNECTED TWS 20FABB000170 SUSPENDED SBC SRC 0 OK
Start streaming music	AVRCP_PLAY 11 AVRCP_PLAY 21 A2DP_STREAM_START 10 A2DP_STREAM_START 2A	AVRCP_PLAY 11 A2DP_STREAM_START 1A

Broadcast Audio

The following examples describe how to configure devices as Broadcaster or Receiver, associate a Receiver to a Broadcaster, and start/stop Broadcast Audio.

Table 2-21: Broadcast Audio—Initial Configuration

Description: Configure one BC127 as a Broadcaster and another BC127 as a Receiver		
Broadcast Audio configuration	Melody 7.1	
	BC127 (Broadcaster)	BC127 (Receiver)
	SET BA_CONFIG=1 OFF 0A02 0304	SET BA_CONFIG=2 OFF 0A02 0304
	OK	OK
	WRITE	WRITE
	OK	OK
	RESET	RESET
	Sierra Wireless Copyright 2018	Sierra Wireless Copyright 2018
	Melody Audio V7.1 BA	Melody Audio V7.1 BA
	Build: 155366443	Build: 1526923344
	Ready	Ready
	Melody 7.2 and later	
	BC127 (Broadcaster)	BC127 (Receiver)
	SET BA_CONFIG=1 OFF	SET BA_CONFIG=2 OFF
	OK	OK
	WRITE	WRITE
	OK	OK
	RESET	RESET
	Sierra Wireless Copyright 2018	Sierra Wireless Copyright 2018
	Melody Audio V7.2 BA	Melody Audio V7.2 BA
	Build: 155366443	Build: 1526923344
	Ready	Ready

Table 2-22: Broadcast Audio — Association

Description: Associate a Receiver to a Broadcaster.		
Enable Broadcaster association	BC127 (Broadcaster) ASSOCIATION ON OK	BC127 (Receiver)
Associate Receiver	Melody 7.1	
	ASSOCIATION_IN_PROGRESS 20FABB000162	ASSOCIATION ON ASSOCIATION_IN_PROGRESS 20FABB000160 ASSOCIATION 0 0A02 0304 OK
	Melody 7.2	
	OPEN_OK 14 BLE 583E24408AA9 ASSOCIATION_IN_PROGRESS 20FABB000162 BLE_PAIR_OK 583E24408AA9 20FABB000181 CLOSE_OK 14 BLE 583E24408AA9	ASSOCIATION ON OK OPEN_OK 14 BLE 6EF029B7D0A6 ASSOCIATION_IN_PROGRESS 6EF029B7D0A6 BLE_PAIR_OK 6EF029B7D0A6 20FABB000180 ASSOCIATION 0 0A02 0304 CLOSE_OK 14 BLE 6EF029B7D0A6 LIST BA 20FABB000180 OK
Disable Broadcaster association	ASSOCIATION OFF OK	

Table 2-23: Broadcast Audio — Broadcast

Description: After associating a receiver with a broadcaster, start and stop Broadcast Audio.		
Start broadcast	BC127 (Broadcaster) BROADCAST ON OK BA_BROADCASTER_START	BC127 (Receiver) BROADCAST ON OK BA_RECEIVER_START
Stop broadcast	BROADCAST OFF OK BA_BROADCASTER_STOP	BROADCAST OFF OK BA_RECEIVER_STOP

C: Terms and Definitions

Table 3-1: Terms and Definitions

Abbreviation	Description
A2DP	Advanced Audio Distribution Profile
AG	Audio Gateway
AVRCP	Audio/Video Remote Control Profile
BLE	Bluetooth Low Energy
cVc	Clear Voice Capture
DFU	Device Firmware Upgrade
EA	External Accessory
HF	Hands-Free Unit
HFP	Hands-Free Profile
HID	Human Interface Device Profile
HSP	Headset Profile
iAP	iPod Accessory Protocol
MAP	Message Access Profile
Multipoint	When more than one device is connected
PBAP	Phone Book Access Profile
PDL	Paired Device List
QTIL	Qualcomm Technologies International, Ltd.
SCO	Synchronized Connection Oriented
SDP	Service Discovery Profile
SPP	Serial Port Profile
TWS	True Wireless Speaker
UUID	Universally Unique Identifier
WBS	Wide Band Speech