Melody Audio 6

User Guide

SIERRA WIRELESS®

41110747
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 6:00 pm PST
Corporate and product information	Web: sierrawireless.com

Revision History

Revision number	Release date	Changes
1	March 20, 2017	Initial Draft in SWI Template.
2	May 26, 2017	Minor text color updates in chapters 5 & 6.
3	July 31, 2019	Updated Change Log up to Melody 6.1.8 RC3

Contents

1: Introduction

Melody is an embedded firmware solution running on the BC127 module offering both Classic Bluetooth and Bluetooth Smart (BLE) functionalities. It allows you to develop a Bluetooth product without detailed knowledge of the Bluetooth standard.

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

Key Specifications

- Best-in-Class Configurable Bluetooth 4.2 + BLE and Wireless Audio
- Embedded Software and Stack that provides an abstraction layer of the Bluetooth Link
- Controlled via UART, USB, BLE or SPP interfaces
- BR/EDR Profiles: HFP (AG and HF), A2DP (Sink and Source), AVRCP, PBAP, MAP, HID, SPP
- BLE Profiles: GAP, GATT and BC Smart services and generic functionalities for any other profile.
- Multipoint Support
- True Wireless Stereo (TWS) support
- Supports SBC, AAC, aptX and MP3(optional) codecs
- Analogue and digital (I2S, PCM, SPDIF) audio inputs and outputs
- Supports data transfer over BLE, SPP, iAP1, and iAP2[1] protocols
- GPIO control, LED indications, capacitive touch sensor
- Future proof - uses the latest hardware and supports firmware upgrade (DFU)

Releases

Melody 6
- Standard release available on our website.

Melody 6 MFI
- Available to Apple MFI licensees.
- Include IAP support.

Melody 6 HD
- For BC127-HD only.

1. Only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees.

- Include aptX-HD codec and support 24 bit audio resolution.
- aptX Low Latency not available.

Please upgrade to the latest version of Melody 6 available on our website. The BC127 Upgrade Tool is necessary to perform the firmware update and can be downloaded from our website as well.

http://source.sierrawireless.com/

Applications

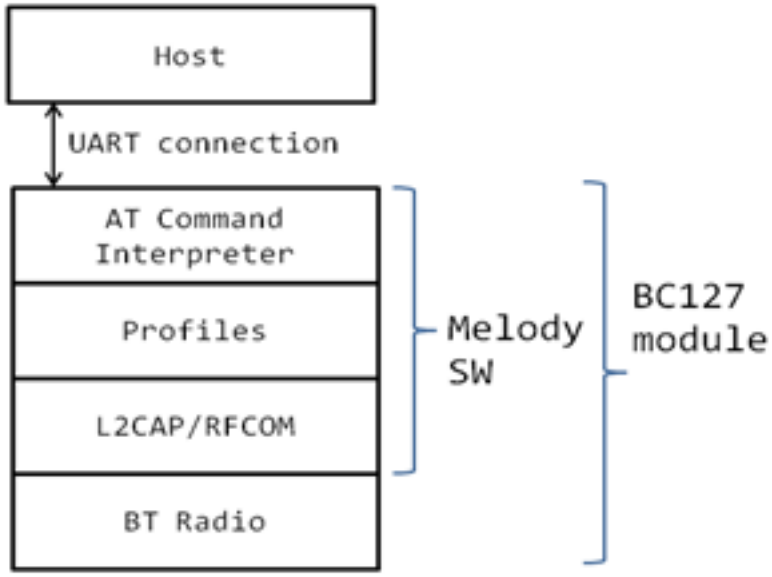

Figure 1-1: Component Block

- Wireless Speakers, Docks and Headsets
- SmartPhone Controlled Audio Systems
- Automotive Infotainment Systems
- Medical Devices
- High Quality Audio Streaming
- Gaming Accessories and MP3 Players

2: Quick Start

Please make sure you are running the latest version of Melody 6 available on our website: http://source.sierrawireless.com/.

Setting Up

To start using Melody you will need to have:

- A Bluetooth Development board such as BC127-DEVKIT.
- A computer with a serial port running a serial terminal, such as HyperTerminal or PuTTY for Windows or an equivalent program, to communicate over the COM interface.

Connect your PC to your Bluetooth module using the serial port. By default, Melody uses the following UART settings:

Table 2-1: Default UART Settings

Baud Rate	9600bps
Data Bits	8
Stop Bits	1
Parity Bit	No Parity
HW Flow Control	Disabled

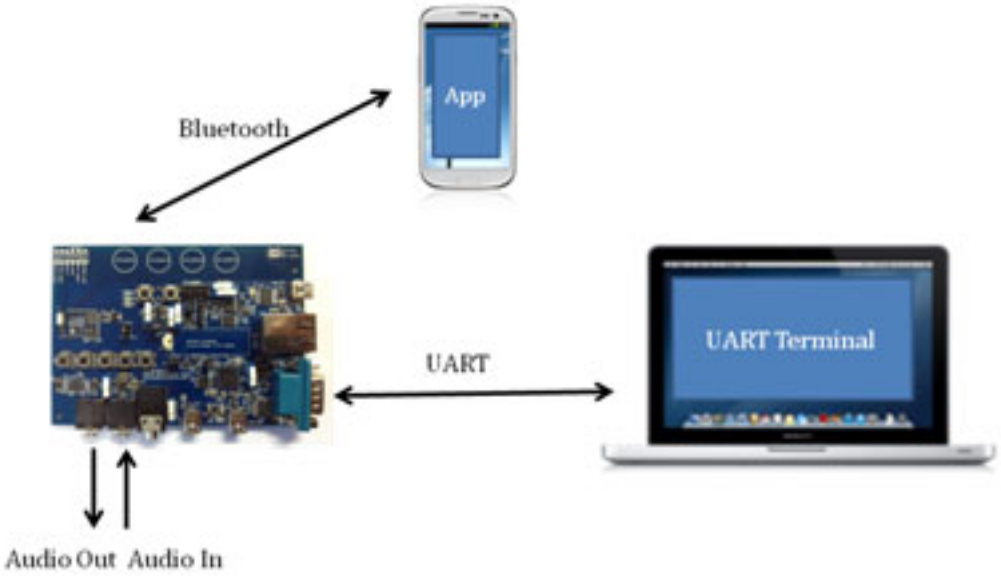

Figure 2-1: Example Standard Configuration

Once you have configured your serial terminal and opened the COM port, power up the development kit. You should see a prompt appear on the screen of the terminal. If you see a prompt and a “Ready”, the module is ready to operate. Note that end of line character used by Melody is a Carriage Return(‘\r’ or 0x0D).

You are now ready to control the module!

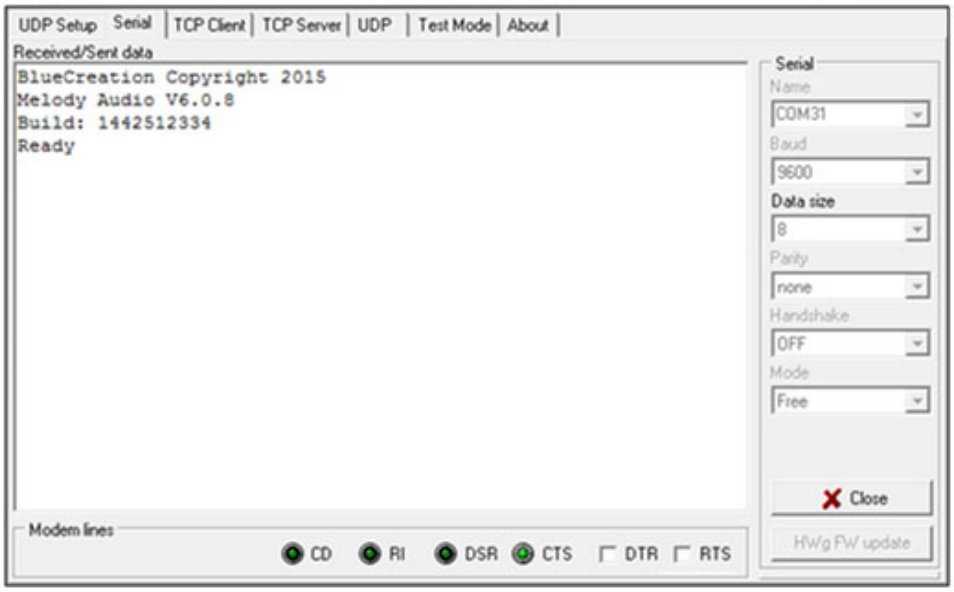

Figure 2-2: Melody Initial Load with Ready Prompt

If you do not see the prompt and “Ready” appear, please check that:

- the module is powered ON and receiving power.
- the UART settings on your serial terminal are the same as above.
- you have opened the right COM port.
- the module is preloaded with Melody firmware.

Default Behaviour

When Melody boots with factory default settings, it is discoverable. Any Bluetooth enabled phone or other Bluetooth device can see it when scanning (discovering) other Bluetooth devices. Melody will appear as ‘BCXXXXXX’[1] and once connected, it will transparently transfer any audio stream to the speaker output.

1. Where “XXXXXX” represents the last six hexadecimal digits of the device’s Bluetooth address.

3: LINK ID Management

Melody provides an easy way for the user to interact with devices, profiles and other interfaces through Link IDs.

A Link ID is an 8-bit hexadecimal value *0xAB*. In most cases, the first digit identifies a device and the second digit refers to a profile as described in the tables below:

Table 3-1: Device Field Values

Device A	Description
0	Reserved Values
1	Device 1
2	Device 2
3	Device 3
..	...

Table 3-2: Profile Field Values

Profile (B)	Description
0	Advanced Audio Distribution Profile (A2DP)
1	Audio/Video Remote Control Profile (AVRCP)
2	Hands-Free Profile (AGHFP), audio gateway
3	Hands-Free Profile (HFP), hand-free device
4	Bluetooth Low Energy (BLE)
5	Serial Port Profile (SPP)
6	Phone Book Access Profile (PBAP)
7	Human Interface Device Profile (HID)
8	Manage Access Profile (MAP)
9	iPod Accessory Protocol (IAP)
A	True Wireless Stereo (TWS)

For example the link ‘24’ refers to the BLE connection with the second device.

In case the device field has a value of zero, the value of the profile field indicates any of the following interfaces:

Table 3-3: Reserved Values for Profile Field

Profile B	Description
1	Analog Interface
2-F	Reserved for Future Use

4: Operating Modes

Melody has two operating mode which define the way the data coming from UART (or USB) is processed. By default Melody is in *Command* mode.

Command Mode

In this mode, Melody parses the commands (see Melody Commands) received over UART and executes them. This allows controlling and configuring Melody.

Data Mode

If one or more BLE, IAP or SPP link are established, the user can enter Data Mode for any of these links to be able to transfer data without having to use a command. The raw data received through the connection is seamlessly transferred to the UART interface and vice versa. The UART interface is hence exclusively dedicated to this mode and Melody won't parse any command expect the special sequence "$$$$" to exit Data mode.

UART flow control should be enabled to use this feature. Note that it is still possible to interact with Melody by using another link (BLE or SPP) to send commands. If Melody is in Data mode and the connection is closed, it returns automatically to Command mode. To switch between Data and Command mode the following commands can be used:

Table 4-1: Operating Mode Commands

Command	Switch From	Switch To
$$$$	Data Mode	Command Mode
ENTER_DATA_MODE [link_id]	Command Mode	Data Mode

Alternatively, if GPIO control is disabled, PIO_5 can also be used to enter or exit Data mode (see PIO Functionalities).

Please note that events are not notified over UART in Data mode but it is possible to configure PIO_4 to be raised on specific events.

5: Melody Commands

When in Command mode, the module accepts commands from the host via the UART interface. The generic syntax for commands is:

```
COMMAND (parameter_1) (parameter_2) . . . (parameter_n)\r
```

with a space between each parameter and a Carriage Return (‘r’ or 0x0D) at the end of each command.

The different commands to control the Bluetooth link are listed in alphabetical order below. Mandatory parameters are listed in “( )” optional parameters are listed in “[ ]”.

The maximum length for a command is 150 characters, if a command larger than this is provided, Melody will return an error.

Table 5-1: UART Commands

UART Command	Description
$$$$	Exit Data mode. Must be sent through the UART interface.
ADVERTISING (value)	BLE Advertising. This command is allowed in Peripheral or Beacon mode. **value**: ON/OFF: to start/stop advertising. Returns OK if success or size where size is the size of the advertising data (0 to 31 bytes). Returns PENDING. The data received after this command is used to set the advertising data. Returns OK when the data is fully retrieved.
AT (link_ID) (command to send)	Send a command over the HFP link to the remote device. AT commands must be enabled in HFP_CONFIG.
AVRCP_META_DATA (link_ID)[(type)(data)]	This command can be used to get or set the AVRCP Meta Data depending on the A2DP role of Melody: In sink mode, if ‘type’ and ‘data’ are not provided, Melody will request the AVRCP Meta Data specified in the remote device and reply OK. The attributes received are notified with the AVRCP_MEDIA events. Note that AVRCP Meta Data must be enabled in MUSIC_META_DATA). In source mode, if ‘type’ and ‘data’ are provided, Melody will send a notification (track changed) to inform that the AVRCP Meta Data has changed. The result will be PENDING until the remote device sends a request to get the new attribute. Melody will respond with the data provided and reply OK. **link_id**: AVRCP link **type: attribute type** 1. Title 2. Artist 3. Album 4. Number 5. Total number 6. Genre 7. Playing time **data**: attribute data (string)

Table 5-1: UART Commands

UART Command	Description
BATTERY_STATUS	Returns the battery status of the battery attached to the module by detecting the voltage level.
BC_SMART_COMMAND (link)(command)	Send a command to a peripheral device over BLE.
BC_SMART_NOTIF (link)(data)(cmd)	Enable/disable notifications for the BC smart characteristics of a peripheral device.
BLE_GET_CHAR (link)[handle_start][handle_end]	Discover the characteristics of a peripheral device between handle_start and handle_end in. Lists all the characteristics if handles are not provided. BLE_CHAR notifications are received for each characteristics discovered and OK is returned when the discovery is completed.
BLE_GET_SERV (link)	Discover all the primary services of a peripheral device. BLE_SERV notifications are received for each services discovered and OK is returned when the discovery is completed.
BLE_INDICATION (link)(handle)(size_value)	Sends an indication to the central device for the handle. **link_id**: link of the central device **handle**: handle of the characteristic **size_value**: length of the characteristic value to be indicated in bytes Returns PENDING. The characteristic value is filled with size_value octets received after this command. Returns OK when fully retrieved.
BLE_NOTIFICATION (link)(handle)(size_value)	Sends a notification to the central device for the handle. **link_id**: link of the central device **handle**: handle of the characteristic **size_value**: length of the characteristic value to be notified in bytes. Returns PENDING. The characteristic value is filled with size_value octets received after this command. Returns OK when fully retrieved.
BLE_READ (link)(handle)	Send a request to read a characteristic value from a peripheral device for the handle. **link**: link of the peripheral device **handle**: handle of the characteristic Returns PENDING. A BLE_READ_RES notification with the characteristic value is received when the remote device reply.

Table 5-1: UART Commands

UART Command	Description
BLE_READ_RES (link)(handle)(size_value)	Respond to a read request from a central device (BLE_READ notification). **link**: link of the central device **handle**: handle of the characteristic **size_value**: length of the characteristic value in bytes Returns PENDING. The characteristic value is filled with size_value octets received after this command. Returns OK when fully retrieved.
BLE_SET_DB (size_db)	Change the GATT database for a peripheral device in BLE IDLE state. **size_db**: size of the GATT database in hex. Returns PENDING. The database is filled with size_db words received after this command. Returns OK when fully retrieved or an error if not enough memory to store the database.
BLE_WRITE (link)(handle)(size_value)	Write a characteristic value to a peripheral device for the handle. **link**: link of the peripheral device **handle**: handle of the characteristic. Note that in Melody the handle of a client characteristic config is equal to the handle of the characteristic + 1. **size_value**: length of the characteristic value to be written in bytes Returns PENDING. The characteristic value is filled with size_value octets received after this command. Returns OK when fully retrieved.

Table 5-1: UART Commands

UART Command	Description
CALL (link) (action) [param]	Call interactions with the specified HFP or AGHFP link. For an HFP link, action can be any of the following: • REDIAL: Notifies the AG that the HF wants to establish an outgoing call with the last number dialed. • MEMORY: Notifies the AG that the HF wants to establish an outgoing call using memory dialing. In this case, param is a combination of alphanumeric characters supplied to the AG by the HF. The memory string is AG specific. • OUTGOING: Notifies the AG that the HF wants to establish a call with the number specified in param. • ANSWER: Answer an incoming call from the HF. • REJECT: Reject an incoming call from the HF. • TRANSFER: Performs an audio connection transfer towards the HF (when value is ON_HF) or an audio connection transfer towards the AG (when value is ON_AG). If param is not provided, Melody will decide depending on where the audio is currently routed. • TWC: Three way calling. Send a request to the AG to perform an action based on param. param can be: – 0 Reject – 1 Hold & Accept / Swap – 2 End & Accept – 3 Merge calls (multiparty) – 4 Hang up calls (multiparty) • END: Terminate an outgoing or active call from the HF. For an AGHFP link, action can be any of the following: • ANSWER: For an incoming call, answer the call from the AG. For an outgoing call, notify the AG that the remote side has answered. • END: Terminate an incoming, outgoing or active call. • TRANSFER: Performs an audio connection transfer towards the HF (when value is ON_HF) or an audio connection transfer towards the AG (when value is ON_AG). If param is not provided, Melody will decide depending on where the audio is currently routed. • INCOMING: Setups the AG in incoming call mode for the number specified in param. Number can by any combination of alphanumeric characters. • OUTGOING: Setups the AG in outgoing call mode for the number specified in param. Number can be any combination of alphanumeric characters.
CLOSE (link/ device)	Terminates the Bluetooth Profile connection. link is a number that defines the connection ID. all closes all links. It is also possible to give the device number as parameter to close all the connections for with this device.
CONFIG	Shows all configuration registers.

Table 5-1: UART Commands

UART Command	Description
CONNECTABLE (mode)	ON puts the module in connectable mode. OFF makes the device non connectable.
CVC_CFG (type) [(key) (length)]	Read or write the CVC configuration stored on the module. *type* can be WB or NB. If only type parameter is supplied all four config keys for this type will be displayed. If the *key* and *length* parameters are also present, the key indicated will be written with a value of size length. key can be 0 - 3. *length* is in 16 bits word and can be 0 - 64. 0 will delete the key. The CVC_CFG command will reply with PENDING. The value of the pskey is expected, each word in hexadecimal space separated. A carriage return has to be supplied at the end. OK will be returned if the operation is a success and error code otherwise. Note that you can send the data into chunks. After each, if Melody is still expecting data, it will reply PENDING. When all data has been received, it replies OK.
DFU	Causes a reset to make the device boot in DFU[1] mode.
DISCOVERABLE (mode)	ON puts the module in discoverable mode. OFF makes the device non discoverable.
ENTER_DATA_MODE (link)	Configure the behavior of the specified link to be in Data mode. The parameter link can be any active BLE, iAP or SPP link. In Data mode, Melody sends the data received through the UART interface directly to the remote device and prints the data received without handling it. Note that hardware flow control on the UART interface should be enabled in Data mode.
HID_DESC (size)	Sets the USB descriptor to use when acting as a HID device. Size is the length of binary data following (similar to Send when using raw data). By default a USB descriptor for a simple keyboard is loaded. When using this command any previously loaded USB descriptor is deleted. Any USB descriptor loaded with this command is not persistent across a reset.
HID_READ (BT_addr)	Reads USB descriptor from SDP of the remote HID device with address BT_addr.
HELP	Returns available list of commands.
GET (config_name)	Reads the value of a configuration parameter.

Table 5-1: UART Commands

UART Command	Description
IAP [Parameter=value]	Set iAP parameters. These are the parameters that identify the device for iOS and iOS application. Maximum sizes of configurations in bytes/chars: ACCESSORY_NAME 48 characters MANUFACTURER_NAME 48 characters MODEL_NAME 48 characters SERIAL_NO 16 characters SEED_ID 10 characters PROTOCOL_STRING 48 characters HARDWARE_VER 3 characters FIRMWARE_VER 3 characters If no parameter is provided, all parameters are displayed. Refer to the Application Note on iAP for more information.
IAP_APP_REQ [protocol]	This command will send a request to the iOs device to open an application with the specified protocol (works only with iAP2)s. If no protocol is specified, the default protocol set with the IAP command will be used.
INQUIRY [timeout] [(filter) (value)] [max_results]	Searches Bluetooth devices in the area. The command will return PENDING and an INQUIRY notification will be received for each device found. All parameters are optional. If not provided, default values are used. **timeout**: Inquiry stops after timeout x1.28s. (value between 1 and 48. 10 by default) **filter and value**: Filter inquiry notifications. No filter by default. • 0 - No filter. value is ignored. • 1 - Filter the results by name. value is a string, for example, if the string is “BC-“, only devices with a name starting with “BC-“ will be shown. Spaces are not allowed. • 2 - Filter the results by class of device. value is the COD. • 3 - Filter the results by RSSI. For example 72 to filter the device with an RSSI value inferior to -72dB. • 4 - Show only TWS devices. value is the ON or OFF. **max_results**: Maximum number of inquiry results. 16 by default. Set to zero to indicate unlimited number of devices.
LICENSE [type] [=value]	Sets or returns the APTX or CVC license. Type can be any of the following: • CVC: Sets or gets the CVC license. • APTX: Sets or gets the APTX license. If parameter value is provided along with parameter type, the command updates the specified license to the one provided by the user in the value parameter. A valid license has the following format: XXXX XXXX XXXX XXXX XXXX Where X is any hexadecimal digit. If no parameters are provided, the current CVC and APTX licenses are returned.

Table 5-1: UART Commands

UART Command	Description
LINK_POLICY (device) (nb_entries)	Sets a link policy power table for a connection. By default Melody uses its own default values. But these can be changed with this command. The link policy power table allows the device to switch between different power modes. **device**: index of the remote device. **nb_entries**: Number of entries in the power table (max. 8). This command will return PENDING and expect the entries separated by <CR>. Each entry shall have the following format in hex: (state) (min_interval) (max_interval) (attempt) (timeout) (time). **state**: The power mode • 00 : active mode • 01 : sniff mode • FF : passive mode **min_interval**: Sniff minimum interval Time = N x 0.625. Range 0x0002 to 0xFFFE. Only even values are valid. Only used if state is sniff mode. **max_interval**: Sniff maximum interval Time = N x 0.625. Range 0x0002 to 0xFFFE. Only even values are valid. Only used if state is sniff mode. **attempt**: Number of baseband receive slots for sniff attempt. Length = N * 1.25 msec. Range 0x0001 to 0x7FFF. Only used if state is sniff mode. **timeout**: Number of baseband receive slots for sniff timeout. Length = N * 1.25 msec. Range 0x0001 to 0x7FFF. Only used if state is sniff mode. **time**: The time spent in this state of the power table, in seconds. This must be 0 for the last entry in the table. The command returns OK when all the entries have been successfully retrieved (see Link Policy example).
LIST	Lists paired devices in the format LIST (BT addr) (Space separated list of supported profiles).

Table 5-1: UART Commands

UART Command	Description
MM_CFG (key) [(length)]	Read or write the Music Manager configuration stored on the module. **key** is the number of the DSP PSKey. It can be between 24 and 38. If only *key* is supplied the value of the key is displayed. If *length* is also present, the key indicated will be written with a value of size length. *length* is in 16 bits word and can be 0 - 64. 0 will delete the key. The *MM_CFG* command will reply with *PENDING*. The value of the pskey is expected, each word in hexadecimal space separated. A carriage return has to be supplied at the end. *OK* will be returned if the operation is a success and error code otherwise. Note that you can send the data into chunks. After each, if Melody is still expecting data, it will reply PENDING. When all data has been received, it replies OK.
MUSIC (link_ID)(instruction)	Controls the music stream state and sends AVRCP instructions (PLAY, PAUSE, STOP, FORWARD, BACKWARD, FF_PRESS, FF_RELEASE, REW_PRESS and REW_RELEASE).
NAME (BT_addr)	Returns the friendly name of device with the provided Bluetooth address.
OPEN (BT_addr) (profile) [type]	Establishes a connection with a given Bluetooth address for a certain profile. **BT_addr**: Bluetooth address formatted as 12 Hexadecimal digits with no separators (e.g. 3859F9CCB893). **profile**: Bluetooth profile. Can be: HFP, AGHFP, A2DP, AVRCP, BLE, SPP, MAP, HFP, PBAP[2] , HID or IAP. If profile is left blank, SPP will be assumed. **type** (optional): Type of address • 0 – public address • 1 – random address Note that type is only used for BLE connections. It corresponds to the type in the scan results. Automatically put Melody in connectable mode. Melody will become also discoverable if attempting to connect to an unpaired device.
PAIR (BT_addr)	Enter connectable and discoverable mode and attempt to pair with the device that has the Bluetooth address given as parameter (without opening any profile).
PASSKEY (type) (key)	This function is used to respond to a SSP request (when SSP_CAPS is not equal to 3). Use type = 0 for a confirmation type and key = 0 to reject or key = 1 to accept the pairing. Use type = 1 and followed by 6-digit passkey to confirm the pairing with your key.

Table 5-1: UART Commands

UART Command	Description
PIO (PIOx) (state)	Sets the specified PIO pin to the specified state. • PIO: Output PIO pin index. • State: ON to set (level high), OFF to reset (level low). Before using this command, GPIO control must be disabled.
POWER (status) [status_ble]	Turn Bluetooth functionalities ON or OFF. If one parameter is provided it will be applied to both Classic and BLE functionalities. If two parameters are given, the first one is for Classic and the second one is for BLE. Note that the second parameter is optional and has no effect if BLE is not enabled. **OFF** Disconnects all active connections and puts the device in limbo mode, where it is not connectable, or discoverable. As a consequence, UART commands that cannot be executed are rejected. **ON** Returns the device to a connectable state.
PB_PULL (link_id)<repository><phonebook><maxlist><start index><filter>[3]	Downloads the phonebook from the connected device. The default parameters are: • **Repository**: 1 = local to the device (uint8) • **Phonebook**: 1 = phonebook (uint8) • **Max list**: 0x1000 entries (uint16) • **Start index**: 0x0000 (uint16) • **Filter**: 0x0087 = version, name, full name and phone number (uint16) This command will send the downloaded phonebook in raw format to the host. Received data will be surrounded by: PB_PULL_START <link_id> and PB_PULL_END <link_id>. When the download is finished or aborted the event PB_PULL_OK <link_id> will be sent. For the full description of the parameters, see the PBAP section. Note that the phonebook download requires a baud rate of 115200 or above. Lower baud rates can cause the UART to stall and lose phonebook data.
PB_ABORT (link_id)	Aborts an active phonebook download. By default, the PBAP profile will not be closed if an active download is in process. Aborting the operation is required before closing the pbap profile. Link_id is the PBAP link id in status.
RESET	Resets the device.
RESTORE	Resets and restores the configuration parameters to default factory settings.

Table 5-1: UART Commands

UART Command	Description
ROLE (link_id)[role]	Changes the current role for a particular classic connection. If role is not provided, a notification will be sent with the current role of the local device. link id: A2DP, AVRCP, HFP, SPP or iAP link. role (optional): • M- Change the role of the local device to be the master of the link. • S - Change the role of the local device to be the slave of the link. A notification will be sent if the new role is different from the current role.
ROUTE (link_id/ value) [link_2]	Select which audio routing to apply. When an audio routing is applied, automatic-routing is disabled. You can re-enable the automatic-routing using the value 0 as parameter. The parameter(s) can be either a link_id (A2DP or AG/HFP) or two A2DP source link_id (Dual Stream) or one of the following special values to route the audio stream: • 0 – 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 • 5 – USB to Analogue out • 6 – USB to Digital out Note: Music Manager and TONES features are not supported for the special case 2 and 3. Note 2: If a TWS connection is established, the audio will be automatically forwarded with ROUTE 1 or any A2DP link in Sink mode.
RSSI (link_id)	Returns the receiver signal Strength of the link. - 70dBm is a good link, -80dBm is a poor link.
SCAN (timeout) [raw_data] [melody_filter]	Searches BLE-enabled devices in the area for maximum period of time (given in seconds). *raw_data* and *melody_filter* are optional parameters. They are set to OFF if they are not provided. *raw_data* can be used to retrieve the raw advertising packets. melody_filter can be used to look only for Melody devices. Following this command, a SCAN notification is received when an advertising device is found. Please note that this command is only available when the device is configured in central mode.

Table 5-1: UART Commands

UART Command	Description
SEND (link) (string)	Sends a string to a device through the specified profile. The parameter link can be any active SPP, BLE link or iAP link.
SEND_RAW (link) (nb_bytes)	Define the number of bytes (up to 255) that will be sent to the device through the specified profile. The parameter link can be any active BLE, SPP, iAP or HID link. After this command, the next nb_bytes received over UART are sent to the remote device.
SET (config)=value	Sets a new value to a configuration parameter.
SPEECH_REC (value)	The value can be ON or OFF to respectively activate and deactivate the Speech Recognition.
SSRD (size)	Set the scan response data (BLE Peripheral mode only). **size**: 0 to 31 bytes. 0 will delete the scan response data otherwise the returns PENDING and the data received after this command is used to overwrite the scan response data. Returns OK when the data is fully retrieved.
STATUS [param]	Returns the state of the device: *STATE CONNECTED[n] CONNECTABLE[ON/OFF] DISCOVERABLE[ON/OFF] BLE[ble_state]* **Classic Bluetooth**: Where n is the number of devices connected. The CONNECTABLE and DISCOVERABLE states can be ON or OFF. **BLE**: ble_state can be OFF, IDLE, ADVERTISING or CONNECTED. And a link status for each established links in the format: *LINK (link_id) (state) (profile) (BT_Addr) (Additional info)* where state can be: CONNECTED, DISCONNECTED or LINK_LOSS. Additional information depends on the profile: Streaming status, codec and sample rate for A2DP/TWS. Call state and codec for AG/HFP. Playing status for AVRCP. MTU size for BLE. *param* can be any of the following: • A link. • A profile identifier. (SPP, HFP, A2DP. . .) • A device identifier (RES, DEV0, DEV1, DEV2) When param is supplied, the status command returns the relevant information associated to the value of param
TOGGLE_VR (link)	Start/Stop Voice call command on the phone. The link provided must be an HFP link.

Table 5-1: UART Commands

UART Command	Description
TONE (flag) (value) (flag) (value)	Plays a tone based on the input. A tone must have at least one note. A note must have a length parameter. Please refer to TONE PARAMETERS AND FLAGS for full information regarding, flags, accepted values and descriptions. Flags: • Tempo TE 0 – 4095 • Timbre TI 0 – 7 • Volume V 0 – 255 • Decay D 00 – FF (enter value in hex[4]) • Note N A – F + octave 0 – 9 eg AF7, A7, AS7 • Length L 1,2,4. . . 64 or 3,6,12. . . 96 triplets Note that this command will return an error if you have used the ROUTE command to route the analog interface to the digital interface (ROUTE 2).
UNPAIR [addr]	If the Bluetooth address provided currently exists in the paired device list it is removed, otherwise the user is notified that the Bluetooth address provided could not be found in the PDL. If no parameters are provided, clears all of the Bluetooth addresses from the paired devices list.
VOLUME [link] [command]	Get or set the volume. When the VOLUME command is used without parameters, it lists the volumes of the A2DP, HFP and AGHFP links currently connected. If a link is provided, it will display the volume of the link. Note that link can also be equal to 1 for the volume when using the ROUTE command (ROUTE 1/4). The volume of a link can be changed when a command is also provided. The parameter command can be any of the following: • UP to increase the volume on the specified link by one. • DOWN to decrease the volume on the specified link by one. • An hexadecimal value between 0 and the maximum number of steps defined for the profile. The number of volume steps can be configured for the A2DP profile in BT_VOL_CONFIG. Otherwise it is 15 steps.
VERSION	Returns information on the firmware version number and the Bluetooth address of the device.
WRITE	Store Configurations.

1. DFU allows downloading a new firmware upgrades onto the Bluetooth module via the UART interface and allows users to upgrade melody to new releases.
2. PBAB requires an active HFP connection.
3. The phonebook download requires a baud rate of 115200 or above. Lower baud rates can cause the UART to stall and lose phonebook data.
4. Parsed as fixed point of the following format in binary bbbb.bbbb or hexadecimal X.X.

6: Melody Configuration

The user can configure general parameters for the module. These parameters are stored in the RAM memory. If required, the parameters can be stored to Flash memory. When the module reboots, it will boot with the parameters that are saved to Flash memory.

There are four main commands to configure parameters. The commands to modify configuration parameters are described below. Mandatory parameters are listed in “( )” optional parameters are listed in “[ ]”.

Table 6-1: General Configuration Commands

Command	Description
CONFIG	Shows all parameters.
GET (config_name)	Reads the value of a configuration parameter.
RESTORE	Resets and restores the configuration parameters to default factory settings.
SET (config_name)=param_1 param_2 . . . param_n	Sets a new value to a configuration parameter.
WRITE	Write configuration changes to Flash memory. This command writes all the current configuration parameters to non-volatile memory. These are then read at boot-up time.

Note: if the module boots up with PIO_0 high, the factory default configurations are reset. This allows reverting to a known and working configuration state if severe problems are encountered.

The different configuration parameters are described in alphabetical order in Table 6-2 below. Once modified, many configuration parameters will not take effect before a reboot. Therefore they need to be stored to Flash before rebooting.

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
AUDIO=(input)[output]	Select the audio interface. The first parameter is for the input, second for the output. If only one parameter is provided, it will be applied to both. Values can be: • 0 - Analog • 1 - Digital • 2 - USB	0 0	No
AUDIO_ANALOG=(input_gain)(output_gain)(micbias)(preamp)	Configuration of the audio analog interface(AUDIO=0): • input_gain: Codec input gain (0 – 22) • output_gain: Codec output gain (0 – 15) • micbias: Enable mic bias (2.6V) 1. Disable the microphone bias. 2. Turn the microphone bias on only when there is audio. 3. Enable the microphone bias all the time. • preamp: Enable microphone pre-amplifier	15 15 1 OFF	No
AUDIO_DIGITAL=(format)(rate)(param1)(param2)	Configuration of the digital audio interface(AUDIO=1): **format**: digital output format 0. I2S 1. PCM 2. SPDIF Check the digital interface section in the Audio Configuration chapter for more information about the rest of the parameters.	0 44100 64 100A00	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
AUTOCONN=(value)	0. Disabled 1. Paired Device List. Melody attempts to connect to previously paired devices at power on (see LIST command). 2. Remote address. Melody attempts to connect to the device with the Bluetooth address specified by the REMOTE_ADDR parameter at power on. When autoconn is enabled and Melody is establishing a connection, it automatically connects all the other profiles that have been connected before. This also applies when using the OPEN command. Note: Melody can store up to eight paired devices in its persistent memory. When the pairing memory is full, Melody erase the oldest paired device and keep the eight most recently paired devices in memory.	0	Yes
AUTO_DATA=(ble)(iap)(spp)	Enter Data Mode immediately on BLE/iAP or SPP connection if not already in Data mode.	OFF OFF OFF	No
BALANCE=(left_channel)(right_channel)	Adjust the left/right volume. Values between 0 to 100.	100 100	No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
BATT_CONFIG=(status)(critLvl)(chrgLvl) (vthm_max)(current)	Configures the battery charger and other settings related to the battery: • **status**: ON to enable charging. Should be OFF when not using a battery. • **critLvl**: Critical battery level. Device will shut off until charged above this level (in units of 20mV). • **chrgLvl**: Charge voltage limit threshold. Charging will stop once the battery reaches this voltage (in mV). • **vthm_max**: Thermistor voltage limit threshold. Charging will stop once the thermistor reaches this voltage (in mV). • **current**: Set the current for charging the battery in mA. Values should be in range 0-200 mA. If set to zero, charging is disabled.	OFF 145 4250 1500 150	Yes
BC_SMART_CONFIG=(srv_uuid)(char_uuid) (enable_cmd)	Change the default value for Melody Smart primary service UUID and Melody Smart Data Characteristic UUID. The last parameter can be used to disable commands over BLE (BC_SMART_COMMANDS). It is recommended to disable if high BLE data rate is needed.	68E3 28F0 89F7 D93C ON	Yes
BEACON_DATA=(data)	Configure Beacon. Note that this parameter is used only when the BLE role is Beacon. The first byte define the type of Beacon: 1. iBeacon 2. Eddystone UID 3. Eddystone URL The following 21 bytes defines the Beacon parameters. Refer to the BLE Beacon section for more info.	0 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 04 D2 16 2E EE	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
BLE_CONFIG=(role)[auto_adv][indication][mtu] [random_addr]	**role**: Set the BLE role of the device 1. Peripheral 2. Central 3. Beacon **auto_adv**: Enable/Disable BLE constant advertisements when the device is not connected (only when BLE role is Peripheral). **mtu**: Maximum transmission unit (ATT_MTU). This value is used when negotiating the MTU with the remote device. The actual MTU value can be retrieved from the STATUS command's result. It is important to notice that the maximum packet length that can be sent is MTU-3. Value between 23 and 150. **random_addr**: Generate a random address for BLE connections	0 OFF 40 ON	No
BLE_CONN_PARAMS	Sets the BLE advertising, scanning and connection parameters. Refer to BLE Connection Parameters for more info.	128 12 6 40 0 400 50 400 400 64 400 400	Yes
BT_VOL_CONFIG=(def_hfp)(def_a2dp)(a2dp_steps)(volume_scaling_method)	Bluetooth volume configuration: **def_hfp**: Default HFP volume level in the range 0 to F. **def_a2dp**: Default A2DP volume level in the range 0 to a2dp_steps-1 **a2dp_steps**: Number of steps in hex between minimum and maximum volume (for A2DP only). **volume_scaling_method**: 1. Hardware: volume scaling is applied at the DAC in the case of analog output. 2. DSP: volume scaling is applied in the digital domain by the DSP.	A A 10 1	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
CLASS_1=(value)	Class 1 operation. If set to ON, PIO_0 and PIO_1 functionalities are disabled and can be used to control an external power amplifier (PA).	OFF	Yes
CMD_TO=(value)	Guard time for detecting a $$$$ command to switch from data mode to command mode. There should be 20ms x CMD_TO between the last character and the $$$$. And there should be 20ms x CMD_TO after the $$$$ for the command to be recognised.	20	No
COD=(value)	Class of Device. Some common examples of COD values are the following: • 240404: Wearable Headset Device • 200404: Audio headphones	240404	No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
CODEC=(value) [enable_ talkback]	The value specifies which A2DP optional codecs are enabled. • bit[0]. AAC • bit[1]. aptX • bit[2]. aptX Low Latency (not available on Melody HD releases) • bit[3]. aptX • HD (available on Melody HD releases only) Example: CODEC=3 (0b011) means aptX and AAC are enabled, aptX LL is disabled. Note: SBC is always enabled. When optional codecs are enabled, the order of selection when establishing a stream is: aptX HD > aptX LL > aptX > AAC > SBC. Note: aptX is required for aptX Low Latency and aptX HD. enable_talkback(optional): Enables or disables A2DP TalkBack feature. The TalkBack feature works only with aptX Low Latency and requires a cVc and aptX license key.	0 OFF	Yes
CONN_TO=(value)	Time Melody stays connectable (in seconds) after a connection has been established in Multipoint. Values accepted: 0 – 65534. 0 – never stop being connectable;	0	No
DEEP_SLEEP=(val ue)	Enable and Disable Deep Sleep	OFF	No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
DEVICE_ID=(word_1) . . . (word_8)	The Device ID sets the configuration parameters for the Device ID Profile. This profile sets the following as an SDP record so that remote devices can get extra information about the product: • Vendor Id source (2 bytes) • Vendor Id(2 bytes) • Product Id(2 bytes) • BCD version (2 bytes) • Software version (8 bytes) For more information about these settings, refer to Device ID profile specification on the Bluetooth.org adopted specification page	0000 0000 0000 0000 0000 0000 0000 0000	Yes
DISCOVERABLE=(value)(timeout)	Configure when Melody becomes discoverable. Note that you can also use the DISCOVERABLE command. value: 0: Not discoverable after power on 1: Always discoverable. 2. Discoverable at power on. Not discoverable after a successful connection. Timeout: Discoverable timeout in seconds. Zero to disable. Does not apply if always discoverable.	2 0	Yes
ENABLE_BATT_IND=(value)	Enables and disables the iOS battery indication.[a]	On	No
ENABLE_CAPSENSE=(value)	Enables and disables cap senses. When it is enabled, CAPSENSE notification are received when the state of a pad changes.	Off	Yes
ENABLE_LED=(value)	Enables and disables LED support in Melody	On	No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
ENABLE_SPP_SNIFF=(state)(min_interval)(max_interval)(attempt)(timeout)[duration]	Enables SPP only connections to go into sniff mode which allows Melody to use low power modes. These values take effect if SPP is the only connected profile • state: **OFF** active **ON** -passive (if all parameters are 0) -sniff with passive (with all parameters supplied) Parameters: • **min_interval**: the minimum interval of the sniff period • **max_interval**: the maximum interval of the sniff period • **attempt**: Determines for how many slots the slave shall listen when the slave isnot treating this as a scatternet link. • **Timeout**: Determines for how many additional slots the slave shall listen whenthe slave is not treating this as a scatternet link. • **Duration**: the time to stay in this sniff state.	Off 0 0 0 0 0	No[b]

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
GPIO_CONFIG=(ctrl) [pio4_cfg] [analog]	GPIO configuration. **ctrl**: Enables/disables GPIO control. **pio4_cfg**: Bitmask to select PIO_4 functionality (only if GPIO control disabled): If an event is selected, PIO4 is raised when the event occurs. You can clear it with the PIO command (“PIO 4 OFF”). • Bit0: A2DP_STREAM_START (event) • Bit1: A2DP_STREAM_-SUSPEND (event) • Bit2: AVRCP_PLAY (event) • Bit3: AVRCP_PAUSE or AVRCP_STOP (event) • Bit4: CALL_INCOMING (event) • Bit5: CALL_OUTGOING (event) • Bit6: LINK_LOSS (event) • Bit7: When data mode is exited (event) • Bit8: PIO4 high if connected, low if not connected(state) • Bit9: AVRCP_MEDIA (event) • Bit10: CALL_IDLE or CALL_ACTIVE (event) Value expected in hexadecimal. (ex: ‘003’ to raise PIO_4 on all the A2DP events). **analog**: Configures the analog input detection pin. Value can be any of the following: • 0 to 31: Pin 0 to 31. • 254: Disabled. • 255: Assume always connected. The analog input detection pin is active low.	ON 0 254	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
HFP_CONFIG=(cvc)(wbs) (inband)(nrec)(at_ cmd)	Configuration of the HFP and AGHFP profiles: **cvc**: Enables or disables CVC (Clear Voice Capture). **wbs**: Enables or disables Wide-Band Speech support. **inband**: Enables or disables in-band ringing for incoming calls. **nrec**: Sets the HFP NREC flag, which tells the AG not to use its internal Noise Reduction/Echo Canceller algorithm. We suggest you set this to ON if using CVC or an external noise reduction platform. **at_cmd**: Enables or disables AT commands. Note: Whether Wide Band Speech is used or not depends on the result of the codec negotiation between the hands free device and the audio gateway device.	OFF ON OFF OFF OFF	Yes
HIGH_SPEED=(spp)(iap2)	Enable/Disable High Speed for SPP and iAP2. This feature allows faster data transfer in Data mode. When active, the DSP is used to speed up the transfer and therefore the audio is disabled. Note that when using this mode, the escape sequence to exit Data mode is disabled. You can however use GPIO 5 to exit from Data mode (if GPIO control is disabled) or close the connection.	OFF OFF	No
IAP_PARAMS=(ma x_nb_pkt) (max_pkt_len) (disable_iap1)	iAP link parameters. **max_nb_pkt**: Maximum number of outstanding packets. **max_pkt_len**: Maximum packet length in bytes. **disable_iap1**: ON to disable iAP1 support. OFF to keep iAP1 support.	5 64 OFF	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
LOCAL_ADDR=(value)[c]	Contains the local Bluetooth address of the device.	-	-
MAX_REC=(value)	Maximum times to try to reconnect to a device in .PDL	2	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
MM (enable) [(user_eq_en) (bank) (bass_boost_en) (3d_enhance_en) (compander_ en) (dither_en) (speaker_eq_en)]	Configure the Music Manager audio enhancements for A2DP music streams. All configuration parameters following MUSIC_ MANAGER are ON or OFF except bank which is between 0 and 6. **enable** – mandatory, enable or disable MM. All subsequent parameters are optional but must be provided in order. i.e. if you want to modify (bass boost) you have to supply (user_eq_en) and (bank). **user_eq_en** – enable/disable the user equaliser block **bank** – set the equaliser bank to use when user equaliser is enabled. Available banks are 0 to 6: • 0: None (flat frequency repsponse) • 1: None • 2: Bass boost • 3: Treble boost • 4: Rock • 5: Jazz • 6: None **bass_boost_en** – enable/disable the bass boost block **3d_enhance_en** – enable/disable the 3D enhancement block **compander_en** – enable/disable the compander block **dither_en** – enable/disable the post processing/dither block **speaker_eq_en** – enable/disable the speaker equaliser block Please note that these configurations only enable/disable music manager blocks. Those need to be configured use UFE, as specified by CSR.	OFF OFF 0 OFF OFF OFF OFF OFF	No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
MUSIC_META_DATA=(value)	Enables and disables AVRCP Meta Data.	OFF	No
MUSIC_OLD_AVRCP=(value)	Switch AVRCP version to 1.0. This will disable Meta Data and Absolute volume support.	OFF	Yes
NAME=(value)	Name of the device (32 characters max). The last 5 symbols (“XXXXX”) are the last 6 symbols of the Bluetooth address.	BC-XXXXXX	Yes
NAME_SHORT=(value)	Short name (7 characters max) used for Melody Smart Advertisements. The last 5 symbols (“XXXXX”) are the last 5 symbols of the Bluetooth address.	BCXXXXX	Yes
PROFILES=(hfp)(aghfp)(a2dp_snk)(a2dp_src)(avrcp)(ble)(spp)(pbap)(hid_device)(hid_host)(map)(iAP)	Configuration of the Bluetooth profiles. The value for each profile indicates the maximum number of connections (up to 3). Notes: • It is not possible to have A2DP Sink and A2DP Source at the same time. It means that one of the values must be 0. • BLE is limited to a single connection in Peripheral mode and up to 3 in Central mode. • It is not possible to have HID host and HID device in the same device. • SPP and iAP[d] are limited to two connections.	2 0 2 0 2 1 2 1 0 0 1 0	Yes
PIN=(value)	Pin Code	0000	No
REMOTE_ADDR=(value)	Address of device to auto-connect to on reset. Works only if devices are paired. Will attempt MAX_REC times and then go discoverable.	0	Yes
SPP_UUID=(value)	Change the UUID for the SPP profile.	00 00 11 01 00 00 10 00 80 00 00 80 5F 9B 34 FB	Yes

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
SSP_CAPS=(value)	Set the Secure Simple Pairing capabilities for this device. Accepted values are: • 0 - display only • 1 - display yes/no • 2 - keyboard only • 3 - no display, no keyboard (i.e.JustWorks) • 4 - display and keyboard // not • supported by standard • 5 - reject SSP request If SSP_CAPS is not equal to 3, the PASSKEY command might have to be used to pair with a device.	3	No
TWS_CONFIG=(autoconn) (master) (slave)	True Wireless Stereo configuration: **autoconn**: if ON, connect automatically TWS after an A2DP connection provided that there is there is a TWS present in the paired list and no other TWS device is already connected. **master/slave**: Audio routing for a Master/Slave TWS device relaying/ receiving audio in a TWS session. • 0 - stereo • 1 - left channel • 2 - right channel • 3 - downmix Note that the audio routing applied on both devices is the one configured on the master device. By default, the master device is left and the slave is right.	OFF 1 2	No
UART_CONFIG=(baud) (flow_ctrl)(parity)	UART configuration: **baud**: UART Baud rate (9600, 19200, 38400, 57600, 115200, 230400, 460800, 921600). **flow_ctrl**: Enable or disable flow control. **parity**: UART Parity (0=none, 1=odd, 2=even).	9600 OFF 0	No Yes No

Table 6-2: Configuration Parameters

COMMAND	Description	Default	Requires Reboot
USB_HOST=(value)	Determine the host interface: value is ON for USB and OFF for UART.	OFF	Yes
VREG_ROLE=(valu e)	Changes behaviour of VREG button when held (3s press) • Nothing • Power On/Off	1	No

a. Due to iOS behaviour, this needs to be changed before initial Pairing to iOS device. Changing after that will not remove indicator on iOS.
b. Changing the sniff parameters require the connection to restart.
c. Please note that value is a read only.
d. Only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees.

Audio Configuration

Melody has two types of audio interfaces, analog and digital(I2S/PCM/SPDIF). Both are configurable with the *AUDIO_ANALOG* and *AUDIO_DIGITAL* parameters. The AUDIO parameter allows the selection of the audio input and output separately for the A2DP and HFP profiles.

Analog interface

The analog configuration is stored in the parameter AUDIO_ANALOG:

AUDIO_ANALOG = (input_gain) (output_gain) (micbias) (preamp)

- **input_gain**: Codec input gain (0-22).
- **output_gain**: Codec output gain (0-15).
- **micbias**: Configures mic bias (2.6 V). 0: Disable the microphone bias, 1: Turn the microphone bias on only when there is audio, 2: Enable the microphone bias all the time.
- **preamp**: ON to enables the microphone pre-amp. This results in an additional 20dB of input gain on that channel. In designs that use a microphone, enabling this functionality is recommended.

Note that thoses parameters are taken into account when connecting the audio. For HFP/A2DP, the output gain is adjusted by the DSP based on the current volume. In that case the value is not relevant and the VOLUME command shall be used instead. When cVc is active, the input gain is also set by the DSP. The cVc configuration can be changed using *CVC_CFG* command.

Digital Interface

The digital configuration of the digital audio interface is stored in the parameter AUDIO_DIGITAL:

AUDIO_DIGITAL = (format) (rate) (param1) (param2)

- **format**: 0 - I2S 1 - PCM 2 - SPDIF

- **rate**: Word clock (WCLK) in hertz for I2S/ PCM or output rate for SPDIF. Please note that I2S/PCM support re-sampling for HFP and A2DP, the supported rates are detailed in the Resampling section.
- **param1**: Bit clock scaling factor for I2S. Master clock in kHz for PCM. Not used for SPDIFF.
- **param2**: refer to the tables below. Depends on the format.

By default, I2S is selected and configured in Master mode, Left Justified with 1-bit delay, 16 bps with a Word clock (WCLK) at 44100Hz and a Bit clock (BCLK) at 2.822Mhz (64*44100Hz)

*Note: Bit clock (BCLK) should be equal or superior to 2*bps*WCLK and up to 256*WCLK. If the frame is fully packed, there are no idle BCLK cycles (ie BCLK = 2*bps*WCLK), crop should be enabled.*

The following tables describe param2 of the digital interface for each format:

Table 6-3: Param2 of the Digital Interface for I2S Format

I2S	
[24:31]	Not used.
[16:23]	Bits per sample (bps). If larger than the internal format used by Melody, the additional bits will be output as zeros in the least significant bits.
[12:15]	Audio attenuation in 6 dB steps. Valid range: 0 to 15 inclusive.
[11]	Master mode - clock and sync will be generated by the I2S hardware.
[10]	Justify format - 0: left justified, 1: right justified.
[9]	Left justify delay - 0: left justified formats: 0 is MSB of SD data occurs in the first SCLK period following WS transition. 1: MSB of SD data occurs in the second SCLK period.
[8]	Channel polarity Valid values: 0 (SD data is left channel when WS is high), 1 (SD data is right channel when WS is high).
[7]	Audio attenuation enable - 0: 17 bit SD data is rounded down to 16 bits. 1: the audio attenuation defined in Audio attenuation is applied over 24 and 20 bits of incoming data with saturated rounding. Requires crop enable to be 0.
[5:6]	Not used.
[3:4]	Justify resolution - Resolution of data on SD_IN, 00=16 bit, 01=20 bit, 10=24 bit, 11=Reserved. This is required for right justified format and with left justified LSB first.
[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.
[1]	Start Tx sampling 0: during low wclk phase. 1: during high wclk phase.
[0]	Start Rx sampling 0: during low wclk phase. 1: during high wclk phase.

Table 6-4: Param2 of the Digital Interface for I2S Format

PCM	
[26:31]	Not Used.
[25]	Enable PCM master mode - clock and sync will be generated by the PCM hardware.
[22:24]	PCM slot count - Valid range 0 to 4 inclusive. If 0, slot count will be derived from master clock and synchronisation rate.
[21]	Enable PCM Manchester encoding mode.
[20]	Enable PCM short synchronisation - Short frame sync (falling edge indicates start of frame), rising edge indicates start of fame in long sync mode.
[19]	Enable PCM Manchester slave mode - Force transmit frames to follow receive frames with constant delay. Requires extended features to be enabled.
[18]	Enable PCM sign extend - Sign extend 13/8 bit sequence to 16 bit sequence, else pad with the STREAM_PCM_AUDIO_GAIN for 13-bits or 0s for 8 bits.
[17]	Enable PCM LSB first - Transmit data LSB first.
[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.
[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 also not active. 1: tri-state PCM_OUT after rising edge of PCM_CLK.
[14]	Enable PCM synchronisation 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.
[13]	Enable PCM GCI mode.
[12]	Mute PCM_DATA output.
[11]	Enable PCM long length sync - Set PCM_SYNC to 8 or 16 PCM_CLK cycles.
[10]	Enable PCM sample rising edge - Sample PCM_DATA on rising edge of PCM_CLK.
[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 to 7 inclusive.
[5:6]	Sample format - Valid values: 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).
[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 to 3 inclusive.
[0:2]	Audio gain - Valid range: 0 to 7. Used to pad the end 3 bits of a 13 bit PCM sample. It is used by some CODECs (connected to the PCM interface) to allow their gain to be controlled.

Table 6-5: Param2 of the Digital Interface for SPDIF Format

SPDIF	
[4:31]	Not used.
[3]	Set the reporting mode for the SPDIF Rx channel Status.
[2]	Set the SPDIF Tx channel B status same as that of channel A.
[1]	Set the SPDIF Tx channel status word value.
[0]	Set SPDIF RX in auto rate detect mode.

Resampling

Re-sampling is supported for the digital audio input and output. Here are the list of supported rates per profiles:

- HFP-AGHFP: 8000, 16000, 32000, 44100, 48000, 96000
- A2DP: 44100, 48000

7: Melody Notifications

Melody uses ‘notifications’, or text prompts, to notify the host of events in the Bluetooth link (for example, completed commands or incoming connections), to provide information, or to require action. Notifications are generated only when Melody is in command or remote configuration mode.

The syntax used is: **NOTIFICATION [link_ID](Parameter)**.

Table 7-1: Events and Notifications

Event	Description
A2DP_STEAM_START [link_ID]	Indicates that the A2DP media stream has been opened. One or more of these messages may be displayed when the remote or local side has requested the A2DP media channel state to change. Please note that an open media stream does not guarantee that there is an active music stream.
A2DP_STEAM_SUSPEND [link_ID]	Indicates that the A2DP media stream has been suspended. One or more of these messages may be displayed when the remote or local side has requested the A2DP media channel state to change. Please note that an a media stream suspend may come sometime after the active music stream has ended, or depending on phone implementation, may not come at all.
ABS_VOL [link_ID](value)	The current absolute volume set as an integer in the range 0 – 127 (representing 0 – 100% as per spec).
AT [link_ID] (length) (data)	Indicates that the remote device sent an AT command or AT reply. The reply is sent over UART as is without modification. The length indicates the length of the data.
AVRCP_MEDIA [link_ID](property: value)	ARTIST: string TITLE: string ALBUM: string NUMBER: integer TOTAL_NUMBER: integer PLAYING_TIME(MS): integer
AVRCP_PLAY [link_ID]	When an AVRCP play event is received from the remote device.
AVRCP_STOP [link_ID]	When an AVRCP stop event is received from the remote device.
AVRCP_PAUSE [link_ID]	When an AVRCP pause event is received from the remote device.
AVRCP_FORWARD [link_ID]	When an AVRCP forward event is received from the remote device.
AVRCP_BACKWARD [link_ID]	When an AVRCP backward event is received from the remote device.

Table 7-1: Events and Notifications

Event	Description
BLE_CHAR [link_ID] type uuid handle	Characteristic discovered: • **link_id**: link of the peripheral device • **type**: U16(16bits UUID) or U128(128bits UUID) • **uuid**: Characteristic UUID (XXXX or XXXXXXXXXXXX- • XXXX-XXXX-XXXXXXXXXXXX) • **handle**: Handle of the characteristic in hex
BLE_INDICATION [link_ID] handle size data	Indication received from a peripheral device. • **link_id**: link of the peripheral device • **handle**: Handle of the characteristic in hex • **size**: Length of the characteristic value in bytes • **data**: Characteristic value (hex)
BLE_NOTIFICATION [link_ID] handle size data	Notification received from a peripheral device. • **link_id**: link of the peripheral device • **handle**: Handle of the characteristic in hex • **size**: Length of the characteristic value in bytes • **data**: Characteristic value (hex)
BLE_READ [link_ID] handle	Read request from a central device. Must use BLE_READ_RES to respond. • **link_id**: link of the central device • **handle**: Handle of the characteristic in hex
BLE_READ_RES [link_ID] handle size data	Read response from a peripheral device. • **link_id**: link of the peripheral device • **handle**: Handle of the characteristic in hex • **size**: Length of the characteristic value in bytes • **data**: Characteristic value (hex)
BLE_SERV [link_ID] type uuid handle	Primary service discovered: • **link_id**: link of the peripheral device • **type**: U16(16bits UUID) or U128(128bits UUID) • **uuid**: Service UUID (XXXX or XXXXXXXXXXXX- • XXXX-XXXX-XXXXXXXXXXXX) • **handle**: Handle of the service in hex

Table 7-1: Events and Notifications

Event	Description
BLE_WRITE [link_ID] handle size data	Write indication. • **link_id**: link of the central device • **handle**: Handle of the characteristic written in hex • **size**: Length of the characteristic value in bytes • **data**: Characteristic value (hex)
CALL_ACTIVE AGHFP [link_ID]	Notifies that there is an active call on the specified AGHFP link.
CALL_ACTIVE HFP [link_ID]	Notifies an active call on the specified HFP link.
CALL_DIAL AGHFP [link_ID] (number)	Notifies that the HFP connected with the AGHFP on the link specified wants to establish an outgoing call to the number specified.
CALL_END AGHFP [link_ID]	Notifies that a call termination on the AGHFP link specified.
CALL_IDLE HFP [link_ID]	Notifies that there are no outgoing, incoming or active calls on the HFP link specified.
CALL_INCOMING AGHFP [link_ID]	Notifies that there is an incoming call procedure on the AGHFP link specified.
CALL_INCOMING HFP [link_ID]	Notifies that there is an incoming call procedure on the HFP link specified.
CALL_MEMORY AGHFP [link_ID] (mem_string)	Notifies that the HF connected to the AGHFP on the link specified wants to establish an outgoing call using memory dialling with the memory string specified. The memory string is AG specific.
CALL_OUTGOING AGHFP [link_ID]	Notifies that there is an outgoing call procedure on the AGHFP link specified.
CALL_OUTGOING HFP [link_ID]	Notifies that there is an outgoing call procedure on the HFP link specified.
CALL_REDIAL AGHFP [link_ID]	Notifies that the HF connected to the AGHFP link specified wants to establish a call to the last number dialed.
CALLER_NUMBER [lind_ID] [number]	Notifies the number of the incoming call.
CAPSENSE (PAD) (DIRECTION)	Notifies when a cap sense pad has changed. PAD can be 0-5 and DIRECTION is UP when pressing the pad or DOWN when releasing it.
CHARGING IN PROGRESS	The charger is currently charging the battery.
CHARGING COMPLETE	The battery is charged and the charger is in standby mode.
CHARGER DISCONNECTED	The charger has been disconnected.

Table 7-1: Events and Notifications

Event	Description
CLOSE_OK [link_ID] (profile)	A connection has been closed for the link indicated.
CLOSE_ERROR [link_ID] (profile)	A connection could not be closed for the link indicated.
CVC_CFG (TYPE)(KEY)[value]	TYPE – WB or NB for wideband and narrowband CVC keys. KEY is the key id 0 – 3; Value – is the key value in ASCII representation of 16bit hexadecimal words. Value will not be present if the key is empty.
DMTF AGHFP [link_ID] (code)	The HF connected to the AGHFP on the link specified requests the AGHFP to send the following DTMF code to the network.
ERROR 0xXXXX	An error with a code is returned when a command has not been executed or the parameters are not correct. You can refer to Error Codes to find out more about the issue.
HID_READ size value	Read HID descriptor. size is the number of bytes. value is the descriptor in hex.
HID_READ_ERROR	Failed to read HID descriptor.
IAP_CLOSE_SESSION [link_ID]	iAP data session is closed.
IAP_OPEN_SESSION [link_ID]	iAP data session is opened.
INBAND_RING AGHFP[link_ID]	Ring indicator received periodically when there is an incoming call.
INQU_OK	Indicates the end of the inquiry.
INQUIRY(BDADDR)(NAME)(COD) (RSSI)	Returned after an INQUIRY command is used if there are discoverable devices nearby. Returns Bluetooth address, name, class of device and RSSI.
LINK_LOSS [link_ID] (status)	Link Loss notification for a specific link (HFP or A2DP). status=1 indicates a link loss. Melody will try to reconnect. status=0 indicates that the link loss has been recovered.
MAP_NEW_SMS [link_ID]	Notifies that a new SMS has been received on the link.
MAP_MSG_BEGIN [link_ID]	Returned when a message data is going to be sent to the host.
MAP_MSG_END [link_ID]	Returned when a message data was completely sent to the host.
NAME [addr] [remote_name]	Returned the BT address and the name of a remote device in response to a NAME command.
OPEN_ERROR [link_ID] (profile)	A connection has failed or there was a link loss for the profile indicated.

Table 7-1: Events and Notifications

Event	Description
OPEN_OK [link_ID] (profile) (Bluetooth address)	A connection has been successfully established for the profile indicated.
PAIR_ERROR (Bluetooth address)	When the module fails to pair with the remote device with the address specified.
PAIR_OK (Bluetooth address)	When the module successfully pairs with the remote device with the address specified.
PAIR_PASSKEY (BDADDRESS) (TYPE)[PASSKEY]	An SSP passkey request is being made. Depending on type these may require user action. TYPE: 0 – Passkey request. Enter passkey using the PASSKEY command 1 – Confirm passkey. Display passkey and have user confirm passkey using the PASSKEY command 2 – Display passkey. Display passkey for user. Note: the PASSKEY parameter will not be supplied for type 0.
PAIR_PENDING	When the module is starting to pair with a remote device.
PB_PULL_START [link_ID]	Returned when a phonebook data is going to be sent to the host.
PB_PULL_END [link_ID]	Returned when a phonebook data was completely sent to the host.
PB_PULL_OK	Returned when the phonebook download has been completed or aborted.
PXP IMM (level)	Proximity Profile, Immediate Alert Service alert level set to: 0 – No Alert; 1- Mild Alert, 2 – High Alert
PXP LLA (level)	Proximity Profile, Link Loss Service alert level set to: 0 – No Alert; 1- Mild Alert, 2 – High Alert
READY	Melody is ready to take commands.
RECV [link_ID] (size) (report data)	Data received from a BLE, HID, IAP or SPP link.
ROLE (BT ADDRESS)<role>	When the remove device changes the classic role, an indication event will be sent to the host with the remote device request a role change.
ROLE_OK <role>	Successful change of the classic role when the ROLE command was sent by the host.
ROLE_NOT_ALLOWED <role>	Failure to change the classic role. This is reached when the remote device refuses the role change requested by the host.

Table 7-1: Events and Notifications

Event	Description
SCAN (BT ADDRESS)(TYPE)(<BT SHORT NAME>) (ADVERTISING FLAGS) (RSSI)	Returned after calling SCAN and contains information about advertising devices. Type is 0 when the address is public and 1 when it is a randomly generated address. If the advertising data does not contain a short name, UNKNOWN will be displayed. The Advertising flags indicate device capabilities: 0x0A indicated this is a General Discoverable Dual Mode device (such as Melody), 0x02 indicates this is a General Discoverable Single mode device (such as Melody Smart).
SCAN_OK	Indicates the end of the scan.
SCAN_RAW (BT ADDRESS) (RSSI) (size) (data)	Returned after calling SCAN with the parameter raw_data ON. size is the length of the advertising data in decimal. data is the advertising data, in hexadecimal.
SCO_OPEN (link_ID)	A SCO connection has been opened and audio is routed.
SCO_CLOSE (link_ID)	A SCO connection has been closed and audio is no longer routed.
SR: YES	“YES” has been detected with Speech Recognition
SR: NO	“NO” has been detected with Speech Recognition
SR: Unrecognised word	An unrecognised word has been detected with Speech Recognition.

8: Pairing and Connection

Secure Simple Pairing capabilities

When discoverable, Melody will accept any pairing request from devices in range. By default Melody will use secure simple pairing (Bluetooth 2.1 and above). If the devices trying to pair are older (Bluetooth 2.0 and below), Melody will accept the older pairing procedure.

For Bluetooth 2.1 devices and newer Melody support different type of pairing (see SSP_CAPS configuration). By default it will accept any pairing request using the 'JustWorks' Bluetooth 2.1 method. This means that the module will accept any connection from Smartphones and other devices. The user of the device will not be required to enter a PIN code.

If SSP_CAPS is no set to 3, you might have to use the PASSKEY command to respond to a pairing request.

For Bluetooth 2.0 devices and older Melody will require a PIN code to accept a connection. The PIN code is set to 0000 by default, but can be reconfigured. This means that the phone user will be required to enter the pin code in order to pair successfully.

Connection

Melody can be put in discoverable mode using the DISCOVERABLE command. It means it will be visible to other devices. To scan for discoverable devices, you can use the INQUIRY command.

When you have the Bluetooth address of the device you want to connect to, use the OPEN command to connect and open a Bluetooth profile. If you are not paired, Melody will automatically pair before connecting. Note that there are a limited number of connections allowed per profile. This can be configured by setting up the PROFILES parameter. When a connection is established, an OPEN notification is received (see Event Notifications). All connected devices are listed with their respective profiles when using the STATUS command. The CLOSE command can be used to disconnect the connected profiles.

Link loss

Link loss is supported for the HFP or A2DP profile. It means that if the connection is lost, Melody will attempt to reconnect to the remote device for 10 minutes. A LINK_LOSS notification is sent with status=1 (e.g. "LINK_LOSS 13 1") every 10 seconds if the profile is HFP or just once if it is A2DP.

If the connection is recovered, a LINK_LOSS notification with status=0 is sent (e.g. "LINK_LOSS 13 0"). Melody will automatically reconnect all the profiles that have been closed due to the connection lost.

If Melody cannot reconnect after the 10 minutes, the link is closed and a CLOSE notification is sent (e.g. "CLOSE_OK 13 HFP").

Note that you can check the connection status of a link (CONNECTED, LINK_LOSS or DISCONNECTED) with the STATUS command.

9: Bluetooth Profiles

A2DP / AVRCP

A2DP(v1.3) and AVRCP(v1.6) profiles can be enabled by setting up their maximum number of connections (up to 3) in the parameter PROFILES. Please note that for A2DP, there are 2 values, one for the A2DP Sink role and one for the A2DP Source role. Only one A2DP role can be enabled at the time (i.e. Sink or Source).

The MUSIC command allows you to control independently the audio streams and the ROUTE command to select the one to route when there multiple devices connected. Finally the VOLUME command can be used to adjust the volume for each link.

SBC is the codec used by default, additional codecs (AAC, aptX, aptX low latency) can be enabled with the CODEC configuration. aptX HD is available with Melody HD.

Extra information related to a song (title, artist, album, genre...) can be displayed on the UART when metadata are enabled (MUSIC_META_DATA).

True Wireless Stereo

Melody 6 includes True Wireless Stereo (TWS) support with SBC or aptX. TWS allows two Melody boards to connect to each other and share audio that is being streamed to one of them. The Melody boards can be configured to be Left, Right or Stereo channels (cf. TWS_CONFIG configuration). This allows the use case for TrueWireless Stereo speakers over Bluetooth.

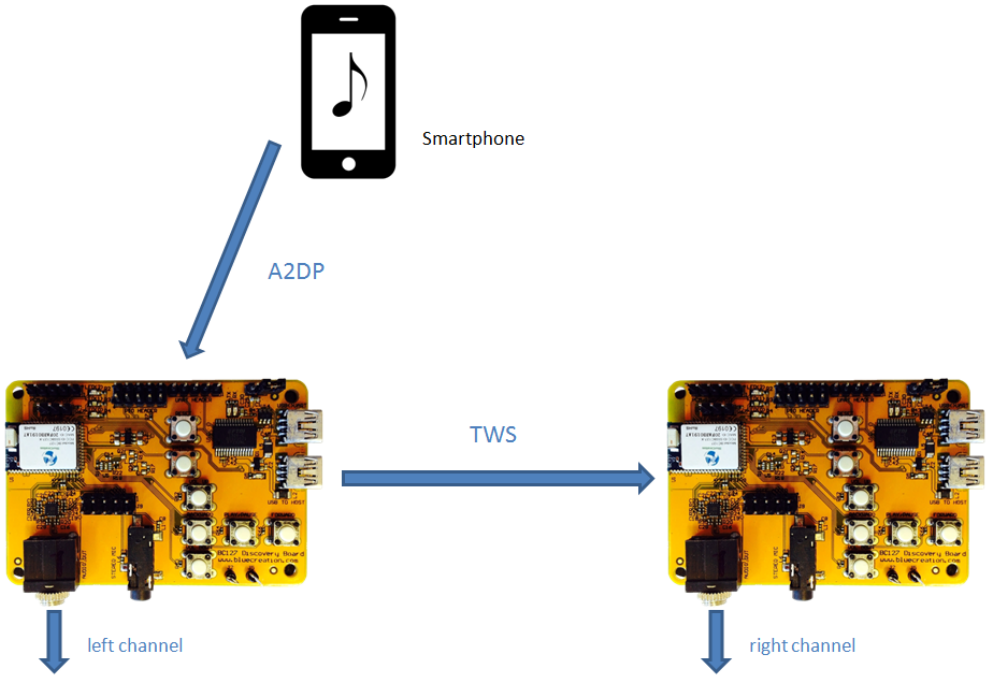

Figure 9-1: True Wireless Stereo with Melody

You can find TWS devices using the INQUIRY command with the TWS filter enabled and use the OPEN command to open the TWS connection (see True Wireless Stereo (TWS) example).

Dual Stream

Melody 6 includes the Dual Stream feature. It allows to stream audio to two A2DP sink devices simultaneously. Most commonly to two A2DP headsets. To start streaming to 2 A2DP sink, you can use the ROUTE command with the 2 A2DP links as parameters (ex: “ROUTE 10 20”).

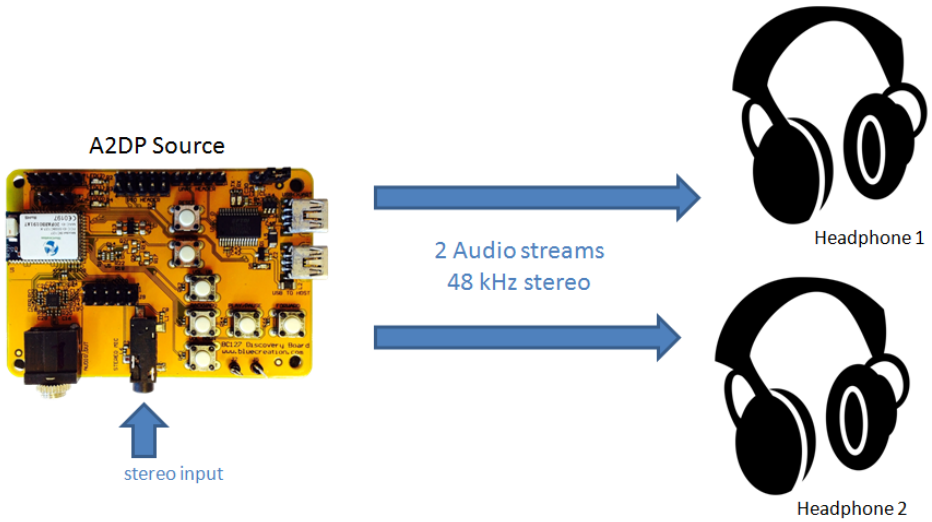

Figure 9-2: Dual Strem with Melody

TalkBack

The aptX Low Latency codec can provide additional cVc wideband back channel support. It can be enabled in the CODEC config. The back channel is an 16k mono audio stream taken from the left analog input of the A2DP Sink device and is routed to the left analog output of the A2DP Source device. Please note that this feature requires a valid aptX and cVc license key.

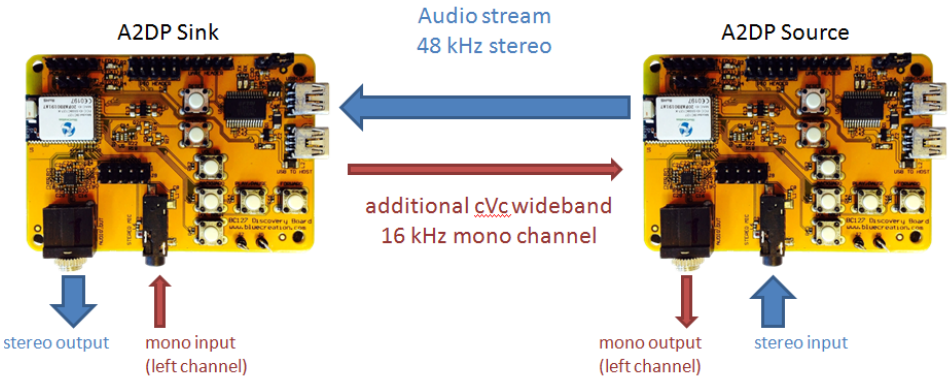

Figure 9-3: TalkBack with Melody

BLE

This chapter explains the BLE (Bluetooth Low Energy) functionalities of Melody 6.

General BLE functionalities

Enable / Disable BLE

BLE is enabled by default. The number of connection supported is specified in the PROFILES config. Only one connection is possible in Peripheral mode and up to three in Central mode. When set to 0 BLE is disabled.

BLE states

There are four BLE states. The current state can be known using the STATUS command:

- OFF - BLE disabled
- IDLE - wait for user action (advertising, connect..)
- ADVERTISING – Melody is advertising (Peripheral or Beacon only)
- CONNECTED – at least one BLE connection is established

BLE roles

Melody supports three BLE roles:

- Peripheral (server)
- Central (client)
- Beacon (iBeacon / Eddystone)

The BLE BLE role is set in BLE_CONFIG. When the role is changed, all BLE connections are closed.

Connection, advertising and scanning parameters

The BLE connections parameters can be modified by changing the values in the parameter BLE_CONN_PARAMS. Theses parameters are used by Melody to set the following:

- Advertising parameters. This will control how Melody advertises (fast advertisements or slow advertisements). While fast advertisements allow remote devices to see and connect to melody faster (melody will be consuming more power), slow advertisements will reduce the power consumption and make the scanning and connection slower.
- Scanning parameters. This will control melody scanning intervals. As in advertising, fast scanning will find more devices with higher power consumption, and slow scanning will require more time to find the required device, but with lower power consumption.
- Connection parameters. Once a BLE link is established, the connection parameters control how often and quick the data transfer is. For example to achieve a higher data rate or lower the power consumption.

The parameters are the following:

- **scan_interval**: The time interval from when BC-127 started its last LE scan until it begins the subsequent LE scan. Scan interval in units of 0.625 ms. The allowed range is between 4 (2.5 ms) and 16384 (10240 ms).
- **scan_window**: Amount of time for the duration of the LE scan. LE ScanWindow shall be less than or equal to LE Scan Interval. Scan window in units of 0.625 ms. The allowed range is between 4 (2.5 ms) and 16384 (10.240 s).
- **conn_interval_min**: Minimum value for the connection event interval. This shall be less than or equal to Connection Interval Max. Connection interval in units of 1.25 ms. The allowed range is between 6 (7.5 ms) and 3200 (4 s).
- **conn_interval_max**: Maximum value for the connection event interval. This shall be greater than or equal to Connection Interval Min. Connection interval in units of 1.25 ms. The allowed range is between 6 (7.5 ms) and 3200 (4 s).
- **conn_latency**: Slave latency for the connection in number of connection events. The allowed range is between 0 and 500.
- **supervision_timeout**: The timeout before disconnecting when no communication is present on the lower layers. Supervision timeout in units of 10 ms. The allowed range is between 10 (100 ms) and 3200 (32 s).
- **conn_attempt_timeout**: Time to wait for connection to be fully established. Changing this value can reduce the number of successful connection.
- **adv_interval_min**: Minimum advertising interval for non-directed advertising. Melody does not support directed advertisements. Range: 32 - 16384.
- **adv_interval_max**: Maximum advertising interval for non-directed advertising. Range: 32 - 16384.
- **conn_latency_max**: Maximum allowed slave latency that is accepted if slave requests connection parameter update once connected.
- **supervision_timeout_min**: Minimum allowed supervision timeout. The minimum allowed supervision timeout that is accepted if slave requests connection parameter update once connected.
- **supervision_timeout_max**: Maximum allowed supervision timeout. The maximum allowed supervision timeout that is accepted if slave requests connection parameter update once connected.

Advertising data

The ADVERTISING command can be used start/stop advertising or to modify the advertising data. If the advertising data is not modified by the user, default values are used. The SSRD command allows the user to set the scan response data.

Random address

By default Melody use a random BLE address. This is necessary to make it work with android. It can be disabled in BLE_CONFIG.

MTU

The maximum MTU (Maximum transmission unit) value supported is set in BLE_CONFIG. When the connection is established the MTU is negotiated with the remote device. The value negotiated is displayed in the STATUS command's result.

BLE Data Mode

When using the BC Smart service (enabled by default), it possible to enter in Data mode to transfer data up to 7.2kbps.

BC Smart Service

The BC Smart service can be used to send seamlessly data and commands over BLE (see BC Smart Service example). The UUIDs of the BC Smart service and its Data characteristic are store in the parameter BC_SMART_CONFIG.

Peripheral Role

1. Set Advertising data and scan response (optional)

   The default advertising data includes NAME_SHORT and the BC Smart service UUID. It can be changed by the user with the ADVERTISING command. The scan response data can be set using the SSRD command.

2. Set GATT database (optional)

   Melody uses a default GATT database including the GAP, GATT and BC Smart services. It can be changed by the user using BLE_SET_DB command in order to support other BLE profiles. The values expected by this command can be generated with the db_gen library which is available on our website in the Tool sections (Tools/Melody BLE db gen). The modified database is not persistent and should be set every time after a reset.

3. Start advertising

   Use the ADVERTISING command to start or stop advertising.

4. Wait for Central device to connect

   When Melody is advertising, a Central device can see it and initiate a connection.

5. Send and receive data

Melody has generic commands and notifications in order to be able to work any BLE profile:

Commands:
- BLE_NOTIFICATION: Send a notification to the remote device.
- BLE_INDICATION: Send an indication to the remote device.
- BLE_READ_RES: Send a read response to the remote device.

Notifications:
- BLE_WRITE: Write indication received from remote device.
- BLE_READ: Read request received from remote device. Must reply with BLE_READ_RES command.

  If the database is not changed by the user, the functionalities relative to the BC Smart services can be used:
- Sending and receiving data with SEND, SEND_RAW commands and RECV notification or in Data mode
- Handling commands over BLE and sending command responses

6. Close connection

   The BLE connection can be closed by the remote device or using the CLOSE command.

Central role

1. Scan

   The SCAN command is used to see the advertising devices in the area. Melody sends a SCAN notification for each device found with its Bluetooth address and the type of address (public or random).

2. Initiate connection with Peripheral

   With the informations retrieved from the SCAN notification (address and type) you can initiate a connection with the OPEN command.

3. Service / Characteristic discovery

   Once connected, the BLE_GET_SERV and BLE_GET_CHAR commands can be used to discover the services and characteristics of the remote device. It gives the handles of the services and characteristics supported by the remote server.

4. Send/Receive data

   The following generic commands and notifications can be used communicate with the remote device (See BLE central example.):

   Commands:
   - BLE_READ: Request to read a characteristic value
   - BLE_WRITE: Write a characteristic value

   Notifications:
   - BLE_READ_RES: Read response received from remote device
   - BLE_NOTIFICATION: Notification received from remote device
   - BLE_INDICATION: Indication received from remote device

   If the remote device supports the BC Smart service, the following commands and notifications can also be used:

   BC Smart commands:

- BC_SMART_NOTIF: Enable/Disable BC Smart notifications
- BC_SMART_COMMAND: Send a command over BLE
- SEND/SEND_RAW/ENTER_DATA_MODE: Send data

BC Smart notification:

- RECV: Data or command response received

5. Close connection

   The BLE connection can be closed by the remote device or using the CLOSE command.

Beacon role

Three types of Beacons are supported: iBeacon, Eddystone UID and Eddystone URL. The type of Beacon and its parameters are set in BEACON_DATA (see Beacons example). Just like with the Peripheral role, the ADVERTISING command can be used to start and stop advertising.

Table 9-1:

	Beacon Data				
iBeacon	0	ProximityUUID (16 bytes)	Major (2 bytes)	Minor (2 bytes)	Tx Power (1 byte)
Eddystone-UID	1	Namespace (10 bytes)	Instance (6 bytes)	Tx Power (1 byte)	unused (4 bytes)
Eddystone-URL	2	Tx Power (1 byte)	Prefix (1 byte)	encoded URL (17 bytes)	

HFP

Melody supports the HFP profile (v1.7). The two HFP roles, Hands-Free unit (HF) and Audio Gateway (AG), are supported and can be enabled or disabled in the PROFILES config. Melody allows up to 3 HF or AG connections. The configuration for this profile is stored in HFP_CONFIG.

Melody 6 supports 4 codecs: * Narrow Band (8kHz) *Wide Band Speech (16kHz) * cVc Handsfree Narrow Band (8kHz) * cVc Handsfree Wide Band (16kHz).

Please note that the use of Clear Voice Capture (cVc), which is an algorithm provided by Cambridge Silicon Radio (CSR) for echo and noise cancellation, requires a license key. Please contact info@blue-creation.com or your distributor for more information about cVc.

Use the *CALL* command to initiate, answer, reject or transfer calls on the AG or HF (see HFP example).

When using the digital interface with I2S, Melody re-samples the audio if the required digital rate (WCLK) is superior or equal to 44100 Hz. Otherwise you need to match the sample rate of the codec (ie. 8kHz or 16 kHz).

Since the audio is a mono signal, the input is always taken on the left channel (Analog or Digital). The output is stereo (same signal copied on both left and right channel).

HID

The HID profile is a generic implementation of HID, this gives better control of the HID capabilities and the supported hid descriptors.

HID host will connect to any HID device, and to identify the HID device, it is possible to use HID_READ {BD ADDR} command to read the HID descriptor of the remote device.

In HID device mode, Melody starts with the default descriptor. The default descriptor is for a simple keyboard without any extra keys (e.g. media keys).

To modify the descriptor, the HID_DESC {size} command should be used.

```
>DESC {size}
>{binary data}
```

Size is the number of octets of binary data following. Make sure you set the descriptor before opening the HID connection.

When HID reports are received over hid, an event will be received:

```
>RECV HID {size} {data}
```

Size is the number of octets of binary data following. Melody simply outputs the received data without processing.

It is the responsibility of the host to handle it.

To send data:

```
>SEND {link id} {size}
>{binary data}
```

“link id” is the id of the HID connection (can be retrieved by sending STATUS command). “size” is the length of binary data to send. Similarly to receive data, any data sent is not parsed or processed by Melody, only sent to the remote device.

iAP

General functionalities

This profile is only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees.

iAP1 support can be disabled in IAP_PARAMS.

Please refer to the iAP Application Note to configure iAP.

Once configured it is possible to connect to two apple devices and send data using the SEND and SEND_RAW commands. It is also possible to send data in Data mode (see Operating Modes).

The IAP_APP_REQ command can be used to send a request to open a application.

iAP link parameters

Two parameters can be adjusted in IAP_PARAMS. It applies only on iAP2 links:

- maximum packets length: The largest possible packet length in bytes (default 64 bytes).
- maximum number of packets: The maximum number of packets that may be sent without receiving an acknowledgement (default 5 packets).

It is recommended to keep the default value. Modifying thoses values may have an impact on the data throughput but could also lead to data loss if not enough memory is available.

iAP Data mode

In Data mode, a 100ms delay should be added after the last character sent over UART before exiting Data Mode to ensure that it is sent to the remote device. With the default configuration, the average throughput is approximatively 70kpbs.

iAP2 High Speed

Data transfer can be speeded up to 500kbps with the iAP2 High Speed feature. This feature, supported only with iAP2, can be enabled in the HIGH_SPEED config. Different link parameters can be used to increase the speed.

We recommend to use the following values: IAP_PARAMS=4 750 OFF. The audio is disabled when using this feature. Please note that it takes approx 500ms to load the feature. Data received over UART during this time is handled after that the feature is loaded. A 100ms delay should be added after the last character sent over UART before exiting Data mode to ensure that it is sent to the remote device.

MAP

Melody 6 includes a generic access to receiving messages over MAP(v1.1) profile. When MAP is connected, a notification service is registered and the phone notifies Melody when a new message is arrived.

When notified, Melody notifies the host with the following event:

MAP_NEW_SMS [link id]

After receiving the notification the full message will be received and passed on to the host without any modification. The full message is surrounded with

MAP_MSG_BEGIN [link id]

and

MAP_MSG_ENG [link Id]

Note that in cases where a message is big, it can arrive inside multiple MAP_MSG_BEGIN and MSP_MSG_END events, and in this case, different events might come between the different MAP messages.

When using MAP with an iOS device, iOS requires the user to enable notification for the paired device from the Bluetooth menu.

PBAP

Melody 6 includes a generic access to downloading the phonebook with PBAP(v1.1.1). To download the phonebook the following command can be used.

PB_PULL (link_id) [repository] [phonebook] [max_list] [start] [filter]

PB_ABORT can be used to stop downloading the phonebook.

LINK ID

Melody PBAP link id, this will be 16, 26, 36, etc., and it can be retrieved from the STATUS command.

REPOSITORY

Table 9-2: Values for repository Parameter

Value	Description
0	Current repository.
1	Local repository: the phonebook will be retrieved from the phones local memory.
2	Sim repository: the phonebook will be retrieved from the SIM card.
3	Any: the phone will select which repository to select.

PHONEBOOK

Table 9-3: Values for phonebook Parameter

Value	Description
0	Not used, will default to 1
1	Main Phonebook folder
2	Incoming calls history
3	Outgoing calls history
4	Missed calls history
5	Combined call history

MAX LIST

The maximum number of entries to download from the remote device. If the remote device number of entries is lower than MAX LIST, then the download will finish.

START

The index of the entry to start download from. To download the whole phonebook, this should be set to 0.

FILTER

The filter defines what information to download. The filter is a 32 bit value where each bit represents a phonebook entry attribute (bits 29-31 are ignored).

Table 9-4: Values for filter Parameter

Bit	Name	Description
0	VERSION	vCard Version
1	FN	Formatted Name
2	N	Structured Presentation of Name
3	PHOTO	Associated Image or Photo
4	DAY	Birthday
5	ADR	Delivery Address
6	LABEL	Delivery
7	TEL	Telephone Number
8	EMAIL	Electronic Mail Address
9	MAILER	Electronic Mail
10	TZ	Time Zone
11	GEO	Geographic Position
12	TITLE	Job
13	ROLE	Role within the Organization
14	LOGO	Organization Logo
15	AGENT	vCard of Person Representing
16	ORG	Name of Organization
17	NOTE	Comments
18	REV	Revision
19	SOUND	Pronunciation of Name
20	URL	Uniform Resource Locator
21	UID	Unique ID
22	KEY	Public Encryption Key

Table 9-4: Values for filter Parameter

Bit	Name	Description
23	NICKNAME	Nickname
24	CATEGORIES	Categories
25	PROID	Product ID
26	CLASS	Class information
27	SORT-STRING	String used for sorting operations
28	X-IRMC-CALL-DATETIME	vCard

Mandatory attributes that should be enabled all the time are:

VERSION, N, and TEL.

Melody defaults are:

VERSION, N, FN, TEL which from the table are bits 0, 1, 2, and 7. This turns into 135 decimal, 0x87 hex or 10000111 binary.

Note that the phonebook download requires a baud rate of 115200 or above. Lower baud rates can cause the UART to stall and lose phonebook data.

SPP

General functionalities

The SPP(v1.1) profile emulates a serial cable to provide a simple substitute for existing RS-232.

It is possible to open up to two SPP connections.

Once connected, you can send data using the SEND or SEND_RAW commands or you can enter in Data mode and everything that will be received over UART will be transferred seamlessly (see Operating Modes).

Data received over SPP are showed with the RECV notification.

Command over SPP

It is possible to send commands over SPP, you need to add a Carriage Return character add the end of your string (SPP example).

SPP High Speed

Data transfer can be speeded up to 700kbps with the High Speed feature. This feature is enabled in the HIGH_SPEED config. The audio is disabled when using this feature. Please note that it takes approx 500ms to load the feature. Data

received over UART during this time is handled after that the feature is loaded. A one second delay should be added after the last character sent over UART before exiting Data mode to ensure that it is sent to the remote device.

10: PIO Functionalities

PIOs have a different behaviour if GPIO control is enabled or disabled in the parameter GPIO_CONFIG.

GPIO Control Enabled

By default, GPIO control is enabled. It allows the module to work autonomously without the need of a host processor as in a wireless speaker for example. The table below lists the functionalities that are attributed to the PIOs. Please note that when in Multipoint, PIO controls will take effect on the active link.

Table 10-1: Command Mode General PIO Bluetooth Commands

GPIO[a]	UART Equivalent	Description
VREGEN single	DISCOVERABLE ON	Melody will become discoverable.
VREGEN long	if VREG_ROLE=0 nothing if VREG_ROLE=1 POWER ON/OFF	Depends on the configuration of VREG_ROLE.
PIO_0 single	VOLUME UP	Increases volume of the current link (HFP or A2DP).
PIO_1 single	VOLUME DOWN	Decreases volume of the current link (HFP and A2DP).
PIO_2 single[b]	MUSIC PLAY/PAUSE or CALL ANSWER/END	Starts playing music or stop playing music for A2DP Source and Sink. If there is an incoming HFP call it will answer the call. During an ongoing call it will end the call.
PIO_2 long[b]	CALL	If HFP is enabled, it will initiate a CALL (last number redial).
PIO_4 single	MUSIC BACKWARD	Sends AVRCP instruction BACKWARD
PIO_5 single	MUSIC FORWARD	Sends AVRCP instruction FORWARD (Next track)

a. Refer to BC127 Datasheet for location of PIO on the module pinout.
b. Only on non-MFI builds.

Note: 'single' is defined as any press and release that happens within 50ms. 'long' is defined as holding the PIO high for 1000ms. A repeated 'long' press will be detected if the PIO is held high for multiples of 1000ms.

Audio Enable PIO

On non-MFI builds, PIO_3 will go high during a call (active SCO) or when streaming music (A2DP). This can be used for example as an audio enable to an amplifier. This feature is not configurable.

GPIO Control Disabled

If PIO control is disabled PIO_0, PIO_4 and PIO_5 are used as the table below:

Table 10-2: PIO Control and Indication

PIO	Type	Description
PIO_0	Output	High if Melody is connected (at least one profile opened, BLE or Classic). Low if Melody is not connected.
PIO_4	Output	Refers to PIO_4 config in GPIO_CONFIG.
PIO_5	Input	On a rising edge, BLE, IAP or SPP connection enter Data mode On a falling edge, exit Data mode.

Restoring the Default Configuration

If the module boots up with PIO_0 high, the factory default configurations are restored. This allows reverting to a known and working configuration state if severe problems are encountered. It is the equivalent of using the RESTORE command.

Capacitive Touch Sensors

This features can be enabled or disabled with the parameter ENABLE_CAPSENSE (OFF by default). If enabled, CAPSENSE notifications are received to indicate when a change (press or release) occurs on any of the 6 pads.

Class 1 device

If CLASS_1 is enabled, PIO_0 and PIO_1 are used to control an external power amplifier (PA). The functionalities for theses PIOs are then disabled. PIO_4 can be used instead of PIO_0 to restore the default configuration.

Note that some PSKEYs need to be updated in this case.

11: LED Indications

LEDs can be enabled or disabled easily by changing the parameter ENABLE_LED. They are enabled by default and the indications are described in the following tables:

Table 11-1: LED Patterns Classic

Classic	LED Pattern (repeating)
Discoverable	Fast Flash Toggle LED 0 and LED1
Connected	LED 1 flashes once approx 2.5s
Connectable	LED 0 flashes once approx 2.5s

Table 11-2: LED Patterns Smart

Smart State	LED Pattern (repeating)
Idle/Off	LED 2 off
Advertising	LED 2 blinks quickly
Connected	LED 2 flashes once approx 2.5s

Table 11-3: Other Indications

Other States	LED Pattern (once)
Powering On	All LEDs ON for 1s
Powering Off	All LEDs quick blink 3 times
Off or DFU	All LEDs OFF

12: Power Management

Melody is configured to reduce the power consumption in both connected and not connected state.

Not Connected

By default, if Melody is not conected, it will be connectable and stays discoverable for a certain amount of time (configurable). If connection is lost, Melody will attempt to reconnect the number of times specified by the MAX_REC parameter. After that, and if Deep Sleep is enabled, Melody will go into Deep sleep mode (a very low power consumption mode). In this mode Melody will not be discoverable or able to accept any connection.

This allows reducing current consumption to a minimum when the device is not connected and not operational. In order for Melody to get out of this mode, a random character has to be sent through the UART. After a 5ms wake-up time the module will be operational again. If GPIO commands are enabled, any GPIO change will also get Melody out Deep Sleep.

Note: Waking Melody up will not disable Deep Sleep and the module will go back into Deep Sleep after the command is executed. It is recommended if Deep Sleep is enabled, to always send a carriage return before a command. This will not return anything if Melody wakes up from Deep Sleep and will return "ERROR 0x0012" if Melody is already awake. After sending the carriage return, wait for 5ms (the chip wake-up time) and then send the command. The chip will stay awake for 1s after the last byte is received on the UART.

Connected

When Melody is connected, it is set-up to reduce power consumption while maintaining the connection. In this state, Melody will automatically establish a link policy to allow it to reduce power consumption while still being able to receive link updates. Note that if Deep Sleep is enabled, Melody will go into deep sleep after 1000ms of no activity.

Battery configuration

The battery configuration is in BATT_CONFIG. The value of the charging current is configurable (between 0 and 200 mA) and three levels define the battery behavior.

- **Vbat Battery Voltage:**

  When the voltage drops to the critical voltage (145*20=2900mV by default), Melody automatically shuts down to preserve remaining power.
- **Vchg Threshold Voltage:**

  Melody will stop charging if the value is superior to this limit (4250mV by default).
- **Vthm Thermistor Threshold Voltage:**

  The Vthm is on AIO1[1] input. Melody will not charge if the value is superior to this limit (1350mV by default).

1. AIO readings saturate above 1.35V

13: Features

Device Firmware Upgrade

Melody support firmware upgrade. It is recommended to upgrade Melody with the latest version on our website as it includes new features and bug fixes.

Please note that there are two different versions, on for Melody and one for Melody HD.

Firmware upgrade can be done over UART from your PC using the latest BC127 Upgrade Tool available on our website.

The DFU command can also be used to enter DFU mode.

Speech recognition

Speech Recognition can be activated / deactivated with the SPEECH_REC command.

When it is ON, notifications are sent when a “YES” or a “NO” is detected on the mic input. You can use theses notifications to trigger actions such as answering or rejecting a phone call.

Note that the Speech Recognition has a highest audio priority. It means that during a call or when music is streaming the audio won’t come out while the Speech Recognition is active.

Speech recognition requires a cVc license key.

Tone generation

Melody allows for the playing of tones of any pitch with the TONE command. The parameters for the TONE command are detailed in TONE PARAMETERS AND FLAGS.

A: Examples

These are some typical use case example.

In every one of them we assume that the pairing list is empty and the configuration is the default one.

```
UNPAIR
>OK
RESTORE
>BlueCreation Copyright 2016
>Melody Audio V6.0.46
>Build: 1470411720
>Ready
```

Pairing and Connection with Melody

Discovering Melody from a phone and initiating paring:

Use the STATUS command to make sure that Melody is discoverable, if it is not the case you can use the DISCOVERABLE command:

```
DISCOVERABLE ON
>OK
STATUS
>STATE CONNECTABLE DISCOVERABLE IDLE
```

Now look for devices from your mobile phone. Your device should appear with the configured name (see NAME parameter). Select and initiate pairing with Melody. This should automatically connect the phone to Melody with all available profiles.

Melody will respond with a notification OPEN_OK or OPEN_ERROR for each profiles.

Discovering a phone from Melody and connect to it:

Make sure Bluetooth is ON and that your phone is discoverable if you want to it to be visible to Melody.

Use the INQUIRY command to look for discoverable devices. Melody will return a list of discoverable device:

When you find the correct device, take note of its Bluetooth address and use the OPEN command to connect.

Melody will respond with a notification OPEN_OK or OPEN_ERROR for each profiles.

```
INQUIRY 10
>PENDING
>INQUIRY {BT ADDRESS}{NAME}{DEVICE CLASS}{RSSI}
...
```

```
>INQU_OK
OPEN {BT ADDRESS} {PROFILE}
>PAIR_PENDING
>PAIR_OK {BT ADDRESS}
>OPEN_OK 10 A2DP {BT ADDRESS}
```

Please note that if you want to connect Melody to a new device, Melody will go automatically in discoverable mode to be able to pair with the new device. Once paired, the device will be added to paired device list (use LIST command to see it) and Melody will be able to connect directly to it.

Link Policy

In this example we assume that Melody is connected to a device (index 1). Here is how to set a power table with two entries:

```
LINK_POLICY 1 2
>PENDING
// Passive mode 30 seconds */
FF 0 0 0 0 1E
>PENDING
// Enter sniff mode (500mS)*/
1 320 320 2 1 0
>OK
```

Digital Audio Configuration

Table 1-1: Digital Audio Configuration Commands

Melody Board 1	Melody Board 2
//initial state SET AUDIO=1 >OK	//initial state SET AUDIO=1 >OK

// I2S master (default configuration)
// Bit clock: 2.822Mhz (64*44100Hz)
// Word clock: 44100Hz
// 16 bits per sample
// Left Justified with 1 Bit delay
SET AUDIO_DIGITAL=0 44100 64 100A00
>OK
	// I2S slave
	SET AUDIO_DIGITAL=0 44100 64 100200
	>OK

// PCM master
// Bit clock: 1.536Mhz
// Word clock: 48000Hz
SET AUDIO_DIGITAL=1 48000 1536 2100420
>OK
	// PCM slave
	SET AUDIO_DIGITAL=1 48000 1536 100420
	>OK

// SPDIF
// Output rate: 48000Hz
SET AUDIO_DIGITAL=2 48000 0 9
>OK
	// SPDIF
	SET AUDIO_DIGITAL=2 48000 0 9
	>OK

A2DP

Multi-connection and Audio Routing

To be able to connect to multiple devices to Melody you have to set the maximum number of connection per profile.

This can be done by setting up the PROFILES parameter as in the following example:

```
// Enable 3 A2DP Sink and 3 AVRCP connections

SET PROFILES=0 0 3 0 3 0 0 0 0 0 0 0

>OK

WRITE

>OK
```

```
RESET
```

In the following, three devices are connected with the A2DP and AVRCP profiles.

The MUSIC command is used to start streaming music and the ROUTE command allows you to select which stream to listen to:

```
>STATE CONNECTED[3] CONNECTABLE[ON] DISCOVERABLE[OFF]
BLE[IDLE]
>LINK 10 CONNECTED A2DP {BT ADDRESS} SBC SRC 44100
>LINK 11 CONNECTED AVRCP {BTADDRESS} STOPPED
>LINK 20 CONNECTED A2DP {BTADDRESS} SBC SRC 44100
>LINK 21 CONNECTED AVRCP {BTADDRESS} STOPPED
>LINK 30 CONNECTED A2DP {BT ADDRESS} SBC SRC 44100
>LINK 31 CONNECTED AVRCP {BT ADDRESS} STOPPED

MUSIC 11 PLAY
>OK
>A2DP_STREAM_START 10
>AVRCP_PLAY 11

MUSIC 21 PLAY
>OK
>A2DP_STREAM_START 20
>AVRCP_PLAY 21

MUSIC 31 PLAY
>OK
>A2DP_STREAM_START 30
>AVRCP_PLAY 31

// Force audio routing to select device 3
ROUTE 30
>OK
// Revert to automatic routing (if more than one device is
streaming, select to
the one that starts playing first, device 1 in this case)
ROUTE 0
>OK
```

Music Source

To be able to stream music from Melody to a speaker or headset you have to enable A2DP Source. This can be done by setting up the PROFILES parameter as in the following example:

```
// Enable A2DP Source and AVRCP (A2DP Sink is disabled)
SET PROFILES=0 0 0 1 1 0 0 0 0 0 0 0
>OK
WRITE
>OK
RESET
```

Melody is now ready to stream some music. Once connected to a device you can use the MUSIC and VOLUME commands:

```
>STATE CONNECTED[1] CONNECTABLE[ON] DISCOVERABLE[OFF]
BLE[IDLE]
>LINK 10 CONNECTED A2DP {BD ADDRESS} SBC SRC 44100
>LINK 11 CONNECTED AVRCP {BD ADDRESS} STOPPED
MUSIC 11 PLAY
>OK
>A2DP_STREAM_START 10
>AVRCP_PLAY 11

VOLUME 11 UP
>OK
>ABS_VOL 11 87
```

It is possible send meta data using the AVRCP_META_DATA command:

```
//notify a track changed and store the meta data
AVRCP_META_DATA 11 2 BlueCreation //ARTIST: BlueCreation
>PENDING
//wait for the remote device to send a request to get the
new meta data
...
//Melody respond to the remote device with the data
previously stored in memory
>OK [#f2]_
```

True Wireless Stereo (TWS)

Here is an example that shows how to stream music from a phone to two Melody devices connected together with TWS.

Table 1-2: Streaming Music from a Phone to two Melody Devices Connected with TWS

Melody Board 1	Melody Board 2
//initial state STATUS >STATE CONNECTED[0] CONNECTABLE[ON] ..DISCOVERABLE[ON] BLE[IDLE] >OK	//initial state STATUS >STATE CONNECTED[0] CONNECTABLE[ON] ..DISCOVERABLE[ON] BLE[IDLE] >OK
//search and connect to the TWS device (melody board 2) INQUIRY 10 4 ON >PENDING >INQUIRY {BT ADDRESS}" melody board 2" 240418 -75db ... >INQU_OK	
OPEN {BT ADDRESS}TWS >PENDING >PAIR_PENDING >PAIR_OK {BT ADDRESS} >OPEN_OK 1A TWS {BT ADDRESS} >OPEN_OK 11 AVRCP {BT ADDRESS}	//TWS connection >PAIR_PENDING >PAIR_OK {BT ADDRESS} >OPEN_OK 1A TWS {BT ADDRESS} >OPEN_OK 11 AVRCP {BT ADDRESS}
//device discoverable, phone pairs and connects DISCOVERABLE ON >OK	
>PAIR_PENDING >PAIR_OK {BT ADDRESS 2} >OPEN_OK 23 HFP {BT ADDRESS} >OPEN_OK 20 A2DP {BT ADDRESS} >OPEN_OK 21 AVRCP {BT ADDRESS}	
//Phone start streaming music >AVRCP_PLAY 11 >A2DP_STREAM_START 10 >AVRCP_PLAY 21	//Phone start streaming music >AVRCP_PLAY 11

TalkBack

```
//Enable aptX Low Latency and TalkBack feature
SET CODEC=6 ON
>OK
WRITE
>OK
RESET
```

HFP

In these examples, we assume that we have two boards running Melody. One configured is configured as the AG (AGHFP enabled in PROFILES) and one as the HF (HFP enabled in PROFILES). The connection is established using the OPEN command from the AG or HF.

Incoming Call

Table 1-3: Incoming Call

ACH	HF
CALL 12 INCOMING {NUMBER} >OK >CALL_INCOMING AGHFP 12 >INBAND_RING AGHFP [#f1]_	>CALL_INCOMING HFP 13

Outgoing call

An outgoing call can be requested from the HF with CALL OUTGOING, REDIAL or MEMORY. On the AG, CALL OUTGOING is used to inform the HF of the outgoing state.

Table 1-4: Outgoing Call

AG	HF
>CALL_DIAL AGHFP 12 {NUMBER}	CALL 13 OUTGOING {NUMBER} >OK
>CALL_REDIAL AGHFP 12	CALL 13 REDIAL >OK

Table 1-4: Outgoing Call

AG	HF
>CALL_MEMORY AGHFP 12 {MEMORY STRING}	CALL 13 MEMORY {MEMORY STRING} >OK
CALL 12 OUTGOING {NUMBER} >OK >CALL_OUTGOING AGHFP 12	>CALL_OUTGOING HFP 13

Call Answer/Reject

When there is an incoming or outgoing call, CALL ANSWER or REJECT shall be used from the HF or the AG.

Table 1-5: Call Answer/Reject

AG	HF
>CALL_ACTIVE AGHFP 12	CALL 13 ANSWER >OK >CALL_ACTIVE HFP 13
>CALL_END AGHFP 12	CALL 13 REJECT >OK >CALL_IDLE HFP 13
CALL 12 ANSWER >OK >CALL_ACTIVE AGHFP 12	>CALL_ACTIVE HFP 13
CALL 12 REJECT >OK >CALL_END AGHFP 12	>CALL_IDLE HFP 13

Call Transfer

When a call is active, the audio is transferred through the eSCO/SCO connection. The CALL TRANSFER command allows switching the audio between the AG and HF.

Table 1-6: Call Transfer

AG	HF

CALL 12 TRANSFER
>OK
CALL 12 TRANSFER
>OK
	// transfer from HF to AG
	>SCO_CLOSE 13
	// transfer from AG to HF
	>SCO_OPEN 13

End call

An incoming/outgoing or active call can be terminated from the AG with the CALL END command. It is also possible to end an outgoing or active call from the HF.

Table 1-7: End Call

AG	HF
CALL 12 END >OK >CALL_END AGHFP 12	>CALL_IDLE HFP 13
>CALL_END AGHFP 12	CALL 13 END >OK >CALL_IDLE HFP 13

Three Way Calling

When there is an active call and another call incoming, the CALL TWC command shall be used instead of CALL ANSWER Tor reject. It can be used for instance to accept the incoming call and put the other call on hold or to merge the calls.

Table 1-8: Three Way Calling

AG	HF

>CALL_INCOMING AGHFP 12	>CALL_INCOMING HFP 13
	
	// accept incoming call and put other
	call on hold
	CALL 12 TWC 1
	>OK
	
	>SCO_CLOSE 13
	// transfer from AG to HF
	>SCO_OPEN 13
	
	// swap calls
	CALL 12 TWC 1
	>OK
	
	// merge calls
	CALL 12 TWC 3
	>OK

AT Commands

```
// Enable AT commands
SET HFP_CONFIG=OFF ON OFF OFF ON
>OK
// open HFP link 13, active call
...
LINK 13 CONNECTED HFP {BT ADDRESS} ACTIVE NB
// send DTMF tones
AT 13 AT+VTS=1
>OK
AT 13 AT+VTS=#
>OK
```

PBAB

```
// Enable PBAP in the profiles
SET PROFILES=1 0 0 0 0 0 0 1 0 0 0 0
// Open PBAP connection (make sure PBAP is enabled in the
PROFILES parameter)
OPEN {BT ADDRESS} PBAP
>PENDING
>OPEN_OK 16 PBAP {BT ADDRESS}

// Download phonebook
PB_PULL 16
>PENDING
>PB_PULL_START 16
>BEGIN:VCARD
>VERSION:2.1
>FN;CHARSET=UTF-8:My Number
>N;CHARSET=UTF-8:My Number
>TEL;TYPE=CELL:+447446110144
>END:VCARD
>BEGIN:VCARD
>VERSION:2.1
>FN;CHARSET=UTF-8:Acavbsxns
>N;CHAPB_PULL_END 16
>PB_PULL_START 16
>RSET=UTF-8:;Acavbsxns
>TEL;TYPE=CELL:1234 567890
>END:VCARD
>PB_PULL_OK 16
```

```
// Download last 16 numbers dialled
PB_PULL 16 1 3 10 0 85
>PENDING
>PB_PULL_START 16
>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
>...
>PB_PULL_OK 16
```

Map

```
// Enable MAP in the profiles
SET PROFILES=1 0 0 0 0 0 0 0 0 0 1 0
// Open MAP connection (make sure MAP is enabled in the
PROFILES parameter)
OPEN {BT ADDRESS} MAP
>PENDING
>OPEN_OK 18 MAP {BT ADDRESS}

// Receive a notification and message
>MAP_NEW_SMS 18
>MAP_MSG_BEGIN 18
>BEGIN:BMSG
>VERSION:1.0
>STATUS:UNREAD
>TYPE:SMS_GSM
>FOLDER:telecom/msg/inbox
>NOTIFICATION:1
>BEGIN:VCARD
>VERSION:2.1
>FN;CHARSET=UTF-8:BlueCreation Test
>N;CHARSET=UTF-8:BlueCreation Test
>TEL:
>END:VCARD
```

```
>BEGIN:BENV
>BEGIN:BBODY
>CHARSET:UTF-8
>LANGUAGE:UNKNOWN
>LENGTH:45
>BEGIN:MSG
>Hello from bluecreation
>END:MSG
>END:BBODY
>END:BENV
>END:BMSG
>MAP_MSG_END 18
```

BLE

BC Smart Service

Table 1-9: BC Smart Service

Melody (Central Mode)	Melody (Peripheral Mode)

// configure device in Central mode
SET BLE_CONFIG=1
>OK
// scan for peripheral devices
SCAN 5
>SCAN {BT ADDRESS}1
<BC00187> 12 -84db
.....
>SCAN_OK
// central connects to peripheral
OPEN {BT ADDRESS} BLE 1
[#f3]_
>PENDING
>OPEN_OK 14 BLE {BT ADDRESS}
	// configure device in Peripheral mode
	SET BLE_CONFIG=0
	>OK
	// start advertising
	ADVERTISING ON
	>OK
	// connection established
	>OPEN_OK 14 BLE {BT ADDRESS}

// write message SEND 14 Hello [#f4]_ >OK	// receive message >RECV 14 Hello

// enable notification (BC_SMART Data characteristic)
BC_SMART_NOTIF 14 ON OFF
>OK
// receive notification
>RECV 14 Hi!!
	// send notification
	SEND 14 Hi!!
	>OK

// send command BC_SMART_COMMAND 14 GET AUDIO >OK >RECV 14 AUDIO=0 >RECV 14 OK	// handle command // response over BLE (if notif enabled)

Generic commands

In this example, Melody is connected to another board with Melody configured in

Peripheral mode.

```
// Get list of services
BLE_GET_SERV 14
>PENDING
>BLE_SERV 14 U16 1801 0001 0004
>BLE_SERV 14 U16 1800 0005 0009
>BLE_SERV 14 U16 180A 000A 001C
>BLE_SERV 14 U16 1804 001D 001F
>BLE_SERV 14 U16 1802 0020 0022
>BLE_SERV 14 U16 1803 0023 0025
>BLE_SERV 14 U128 BC2F4CC6-AAEF-4351-9034-D66268E328F0 0026
002C
>BLE_SERV 14 U128 67D13B00-89B8-11E3-9DE5-0002A5D5C51B 002D
FFFF
>OK

// Get list of characteristics (Device Information service)
BLE_GET_CHAR 14 000A 001C
>PENDING
>BLE_CHAR 14 U16 2A29 000C
>BLE_CHAR 14 U16 2A24 000E
>...
>OK

// Get list of characteristics (BC Smart service)
BLE_GET_CHAR 14 0026 002C
>PENDING
>BLE_CHAR 14 U128 06D1E5E7-79AD-4A71-8FAA-373789F7D93C 0028
//BC Smart Data
>BLE_CHAR 14 U128 818AE306-9C5B-448D-B51A-7ADD6A5D314D 002B
//BC Smart Command
>OK

// Read characteristic - Manufacturer Name (Device
Information service)
BLE_READ 14 000C
>PENDING
>BLE_READ_RES 14 000C 12426C75654372656174696F6E //
BlueCreation
>OK

// Write characteristic - BC Smart Data (BC Smart service)
BLE_WRITE 14 0028 5
>PENDING
{48}{45}{4C}{4C}{4F} // values sent in hex format, 5 bytes
>OK
```

Beacons

This example shows you how to set your beacons parameters. Make sure that BLE is enabled in PROFILES. To advertise, you can use the ADVERTISING command or enable the auto-advertising feature in BLE_CONFIG.

```
SET BLE_CONFIG=2
>OK
// iBeacon
UUID: 0x00112233-4455-6677-8899-AABBCCDDEEFF
Major: 0x04D2 (1234)
Minor: 0x162E (5678)
Tx Power: -18dBm (0xEE)
>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

// Eddystone UID
Namespace: 0x00112233445566778899
Instance: 0xAABBCCDDEEFF
Tx Power: 0xEE
(the last 4 bytes are not used)
>SET BEACON_DATA=1 00 11 22 33 44 55 66 77 88 99 AA BB CC DD
EE FF EE 00 00 00 00
>OK

// Eddystone URL
Tx Power: 0xEE
Prefix: 0x02 (http://)
Encoded URL: 0x676F6F2E676C2F495A304F5141 (goo.gl/IZ0OQA)
[#f6]_
(the last 6 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

WRITE
>OK


RESET
```

Set Advertising Data

```
// Set custom advertising data (flags: 0x06 - UUID128:
bc2f4cc6-aaef-4351-9034-
‹→d66268e328f0 - Short name: BC00181)
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
// Start advertising
ADVERTISING ON
>OK
```

Set Scan Response Data

```
// Set custom scan response data (set the Complete Local
Name(0x09) to "MyName")
SSRD 8
>PENDING
{07}{09}{4d}{79}{4e}{61}{6d}{65}OK
```

HID

HID Host: note that all HID data is in HEX.

```
// Enable HID host profile
SET PROFILES=0 0 0 0 0 0 0 0 0 1 0 0
>OK
WRITE
>OK
RESET
// Read remote device descriptor
HID_READ {BD ADDRESS}
>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
// Initiate a connect to a HID device
OPEN {BD ADDRESS} HID
>PENDING
>OPEN_OK 17 HID {BT ADDRESS}
// Keyboard
>RECV 17 9 {01}{00}{00}{04}{00}{00}{00}{00}{00}{0A}
>RECV 17 9 {01}{00}{00}{00}{00}{00}{00}{00}{00}{0A}
>RECV 17 9 {01}{02}{00}{00}{00}{00}{00}{00}{00}{0A}
>RECV 17 9 {01}{02}{00}{04}{00}{00}{00}{00}{00}{0A}
>RECV 17 9 {01}{02}{00}{00}{00}{00}{00}{00}{00}{0A}
>RECV 17 9 {01}{00}{00}{00}{00}{00}{00}{00}{00}{0A}
// Mouse
>RECV 17 5 {08}{00}{02}{FF}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
>RECV 17 5 {08}{00}{00}{00}{00}{0A}
```

HID Device.

```
// Enable HID device profile and use default keyboard
descriptor
SET PROFILES=0 0 0 0 0 0 0 0 1 0 0 0
>OK
SET COD=000540
>OK
WRITE
>OK
RESET
// Initiate a connect to a HID host
OPEN {BD ADDRESS} HID
>PENDING
```

```
>OPEN_OK 17 HID {BT ADDRESS}
// Press ‘a’
SEND_RAW 17 8
>PENDING
{00}{00}{04}{00}{00}{00}{00}{00}
// Release ‘a’
SEND_RAW 17 8
>PENDING
{00}{00}{00}{00}{00}{00}{00}{00}
// Enable HID device profile
SET PROFILES=0 0 0 0 0 0 0 0 1 0 0 0
>OK
SET COD=000580
>OK
WRITE
>OK
RESET
// Update HID descriptor to be a mouse
// example descriptor:
05010902A1010901A1000509190129081500250195087501810295008103
0501093009311601F826F
F07750C9502810609381581257F75089501810605 0C0A380295018106C0C
0
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}{0
5}{0C}{0A}{38}{02}{95}{01}{81}{06}{C0}{C0}
/ Initiate a connect to a HID host
OPEN {BD ADDRESS} HID
>PENDING
>OPEN_OK 17 HID
// Move the mouse vertically
SEND_RAW 17 6
>PENDING
{00}{00}{30}{00}{00}{00}
// Press left button
SEND_RAW 17 6
>PENDING
{01}{00}{00}{00}{00}{00}
// Release left button
SEND_RAW 17 6
>PENDING
```

{00}{00}{00}{00}{00}{00}

SPP

The following section describes a typical use case for an SPP connection. In this example, we create a SPP connection between two Melody devices to show some of the capabilities of the SPP profile from the transmitter and receiver side.

First you must assure that the SPP profile is enabled on both devices:

Table 1-10: Ensuring the SPP Profile is Enabled on both Devices

Melody SPP Device One	Melody SPP Device Two
// Enable SPP >SET PROFILES=0 0 0 0 0 0 1 0 0 0 0 0 >OK >WRITE >OK >RESET >BlueCreation Copyright 2015 >Melody Audio V6.0.10 >Build: 1443202002 >Ready	// Enable SPP >SET PROFILES=0 0 0 0 0 0 1 0 0 0 0 0 >OK >WRITE >OK >RESET >BlueCreation Copyright 2015 >Melody Audio V6.0.10 >Build: 1443202002 >Ready

The next step is to create an SPP connection, in this example, the OPEN command is used:

Table 1-11:

Melody SPP Device One	Melody SPP Device Two
//Start connection OPEN {BT ADDRESS} SPP >OPEN_OK 15 SPP {BT ADDRESS}	//Accepts connection >OPEN_OK 15 SPP {BT ADDRESS}

We can type status on both devices to see the details of the SPP connection:

Table 1-12: Creating an SPP Connection

Melody SPP Device One	Melody SPP Device Two
//Get connection details STATUS >LINK 15 CONNECTED SPP {BT ADDRESS} COMMAND	//Get connection details STATUS >LINK 15 CONNECTED SPP {BT ADDRESS} COMMAND

So, in both devices, we are in command mode. In this mode, we can use the SEND command as follows to send a string from device one to device two:

Table 1-13: Sending a String from Device One to Device Two

Melody SPP Device One	Melody SPP Device Two
//Sends a string to device two. SEND 15 Hi, I'm device one. >OK	//Device two prints a notification with the SPP link, the length of the received data and the data. >RECV 15 19 Hi, I'm device one.

or we can use the SEND_RAW command to transfer a specified number of bytes:

Table 1-14: Transferring a Specified Number of Bytes

Melody SPP Device One	Melody SPP Device Two

// Use the SEND command to send 20
bytes of data through the SPP
connection.
SEND_RAW 15 20
>PENDING
// The next 20 bytes we send through the UART interface, will be sent through the SPP connection to device
two.
01234567890123456789
>OK
// Device one has already received 20 characters, so send to the data to device two.
	// Do nothing in device two.
	// Do nothing in device two.
	// Device two receives the 20 bytes.
	>RECV 15 20
	01234567890123456789

If device two is in command mode, we can also execute commands in device two from device one, by sending the command we want to execute through the SPP connection. This is currently not supported for all commands.

Table 1-15: Sending a Command through the SPP Connection

Melody SPP Device One	Melody SPP Device Two
// Execute the status command in device two and obtain the results. SEND_RAW 15 7 >PENDING	// Do nothing in device two.
// Send the status command, <CR> is the carriage return character. STATUS<CR> >OK	//Device two, which is in command mode, receives the status command and executes it, then it sends the results through the SPP connection.
// Device one prints the results from executing the status command in device two. >RECV 15 15 STATE CONNECTED >RECV 15 4 OFF >RECV 15 54 LINK 15 CONNECTED {BT_ADDRESS} COMMAND STRING	// Do nothing in device two.

Now we will use data mode in device one, but first the user must assure that hardware flow control in the UART interface is enabled.

Table 1-16: Determining if UART is Enabled

Melody SPP Device One	Melody SPP Device Two
// Get the UART configuration, which shows that HW flow control is enabled. GET UART_CONFIG >UART_CONFIG=9600 ON 0	// Do nothing in device two.

When data mode is enabled, data received from the UART interface is sent through the SPP interface with no additional parsing, and, similarly, data received from the SPP connection is sent through the UART interface.

Table 1-17: Data Received from the UART Interface in Data Mode

Melody SPP Device One	Melody SPP Device Two

// Enable data mode.
ENTER_DATA_MODE 15
>OK
 	
//Data received from the UART interface is immediately sent through the SPP connection.
 	
>Hi I'm device one.
// Data received from the SPP
connection is immediately sent through the UART.
>Hi, I'm device two.
	// Do nothing in device two.
	
	// Device two receives the data sent from device one.
	>RECV 15 1 H
	>RECV 15 18 I, I'm device one.
	
	// Send something from device two to device one.
	SEND 15 Hi, I'm device two.
	>OK

To exit from data mode, you can use the escape sequence:

Table 1-18: Existing Data Mode

Melody SPP Device One	Melody SPP Device Two

// Follow the procedure to exit from data mode using the escape sequence.
$$$$
>OK
// Device two is no longer in data mode.
	// Do nothing in device two.
	
	// Do nothing in device two.

Finally, you can use the CLOSE command to terminate an SPP connection, in this example, we will use the close command in device one:

Table 1-19: Using the CLOSE Command to Terminate an SPP Connection

Melody SPP Device One	Melody SPP Device Two
// Follow the procedure to exit from data mode using the escape sequence. CLOSE 15 >CLOSE_OK 15 SPP	//Device two confirms that the SPP has closed successfully. >CLOSE_OK 15 SPP

Tones

The TONE command is used to play back tones on the Melody audio output. Tones are mixed with any ongoing audio playback. See Appendix A for more details about the tone parameters.

Here are some examples that demonstrate the TONE command capabilities:

A 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 any 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
```

Notes

- If in-band ringing is enabled in HFP_CONFIG
- If the remote device is Melody, don't forget to enable meta data to receive the track information (MUSIC_META_DATA).
- By default Melody uses random address as we can see in the scan result (type = 1)
- To send data over BLE, it is also possible to enter in Data mode.
- Note that with BC Smart Service, it is simpler to use the SEND or the BC_SMART commands
- BlueCreation URL encoded with Google URL shortener at https://goo.gl

B: TONE PARAMETERS AND FLAGS

Tones are defined as a sequence of notes described with pitch (N or TN) described through length (L), tempo (TE),

loudness (V), timbre (TI) and decay (D). All of these parameters can be individually set for each note, changed at any place in the tone string or omitted (all except length) to use default values. Below you can find tables describing the different parameters and the flags used to set them and the values accepted.

Table 2-1: TONE parameter detailed description

Parameter	Flag	Accepted Value and Meaning
Tempo	TE	0 – 4095 Given in quarter notes (crotches) per minute. If no tempo is specified, default is 120.
Timbre	TI	0 – 7 Defines timber of following notes. Please see Table 9 for all available timbres. Default is 0 (Sine).
Volume	V	0 – 255 Sets the volume for the tone. Default is 255 (max).
Decay	D	00 – FF Takes a hexadecimal values that is interpreted as a fixed point decimal number according to this format 0000.0000. e.g. 15 = 1.5. As each tone is played, its volume decreases with a variable rate. Low values for this parameter cause notes to decay very quickly, whereas high values cause the notes to continue with an almost constant volume. A value of 005 (meaning 0.5) will cause each note to reach zero halfway through its duration giving a staccato feel. A value of 20 (meaning 2.0) will cause each note to reach half its initial volume when the next note starts. The default value is 20. This allows notes of the same length to be tied together with TN.

Table 2-1: TONE parameter detailed description

Parameter	Flag	Accepted Value and Meaning
Note	N	This describes a note pitch in the format: Note{F/S}[octave] Note is any note in A – G or R. R is used for a rest or pause. It has to be described for consistency with an octave (e.g. R0). The optional F or S following that indicates if the note is Flat or Sharp. Note, not all notes have flat and sharp variants. The mandatory octave parameter gives the note octave. It takes values between 0 – 9. A note is describes as G4 or AS7 for example. Each note must have a length parameter associated with it and following immediately after: G4 L 1
Tied Note	TN	Ties note to next one. This means that 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. An appropriate decay must be set for this to work. Tied Notes are described in the same manner as Notes.
Length	L	Ringtone Duration. Please refer Table 2-2 to below for details.

Table 2-2: Note Length Values and Description

Length Values	UK Notation	American Notation
1	Semibreve	Whole note
2	Minim Half	note
4	Crotchet	Quarter note
8	Quaver	Eighth note
16	Semiquaver	Sixteenth note
32	Demisemiquaver	Thirty-second note
64	Hemidemisemiquaver	Sixty-forth note
3	Minim triplet	Half note triplet
6	Crotchet triplet	Quarter note triplet
12	Quaver triplet	Eighth note triplet
24	Semiquaver triplet	Sixteenth note triplet

Table 2-2: Note Length Values and Description

Length Values	UK Notation	American Notation
48	Demisemiquaver triplet	Thirty-second note triplet
96	Hemidemisemiquaver triplet	Sixty-forth note triplet

C: Error Codes

Table 3-1: Generic Errors

Error Code	Description
0x0003	Unknown Error

Table 3-2: Command Errors

Error Code	Description
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

Table 3-3: Warnings

Error Code	Description
0x0100	Fail to read battery voltage
0x1001	Fail to initiate IAP due to a wrong configuration
0x1002	Fail to communicate with the Apple MFI Co-processor
0x1003	Fail to configure cap sense
0x1004	Fail to register/unregister device
0x1005	BLE request failed
0xFF01	License key is missing
0xFF02	License key is invalid

Table 3-4: Critical Errors

Error Code	Description
0xF00X	Critical Error
0xF004	Wrong Config

D: Terms and Definitions

Table 4-1: 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
HF	Hands-Free Unit
HFP	Hand-Free Profile
HID	Human Interface Device 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
SPP	Serial Port Profile
TWS	True Wireless Speaker
WBS	Wide Band Speech

E: Change Log

- **Melody 6.1.8 RC3**
  - BLE: Update default advertising flag (dual mode host and controller if random access is disabled)
  - BLE: update BC Smart support (Remote command compatibility with BX310x module and GATT Indications support)
  - BLE: Fix 'CLOSE ALL' command makes the device stop advertising
  - BLE: Fix BC127 stays in CONNECTING mode if remote BLE device not present
  - IAP: Add iap2_feature parameter to IAP_PARAMS to select the iAP2 supported features
  - IAP: Remove automatic App Launch request after connection
  - IAP: Fix incorrectly stored IAP config and limit the maximum number of characters
  - IAP: Fix ERROR 0x1002 on MFI build when VPADS is 3.3V
  - HFP: Fix audio drift with multiconnections
  - Print READY prompt only after all profiles init
  - Fix pairing issue when HFP is disabled
  - Fix inquiry reports Bluetooth name as "unknown"
  - Fix autoconnection does not stop after max unsuccessful attempts is reached (see AUTOCONN and MAX_REC config).
  - Fix OPEN command does not return an error if an A2DP connection is unsuccessful.
  - Fix PROFILES config (3 connections) corrupting the unit
  - Fix Deep sleep on MFI build

- **Melody 6.1.5**
  - Fix High Speed data mode
  - Fix SSR command error mode
  - Add TX Power commands (GET_TX_POWER and SET_TX_POWER)
  - Add disable resampling option (auto_rate param in AUDIO_DIGITAL)
  - Fix AGHFP audio parameters negotiation

- **Melody 6.1.4**
  - Add HID over iAP2 support
  - Update GPIO control for MFI build (Apple Center/Flash button)
  - Fix commands over USB (USB_HOST config)
  - Fix Class of device (COD config) when HFP is disabled
  - Fix POWER OFF when HFP is disabled
  - Update BLE (Add BLE Connecting state. Cannot open a BLE connection in this state)
  - Fix PIO_3 Audio Enable with HFP
  - Fix VOLUME UP with A2DP Source

- **Melody 6.1.3**
  - Add volume scaling method configuration in BT_VOL_CONFIG

- Fix TWS (init, autoconnection and link loss)
- Fix HFP audio priority
- Fix Capsense calibration (pads 1-4 only)
- Fix UNPAIR command (close properly all links)

- **Melody 6.1.2**
  - Add INQU_OK and SCAN_OK notification instead of OK when the inquiry or scan is over
  - Add Bluetooth address to OPEN and CLOSE notifications
  - Fix AGHFP codec negotiation
  - Update 2nd SCO parameters (HFP)
  - Fix ERROR 0x0013 after BLE connection established in Central Mode

- **Melody 6.1.1**
  - HD release available
  - Fix HFP audio
  - Fix ADVERTISING command (in Beacon mode)
  - Add TWS to LIST command profiles
  - Fix TWS with line in (ROUTE 1)
  - Fix A2DP automatically resume streaming
  - Fix aptX audio
  - Fix IAP (App launch after connection and remove version sent after data session opened)

- **Melody 6.1.0**
  - AGHFP 1.7 and HFP 1.7 support
  - Max number of connections increased to 3 for AGHFP and A2DP source in PROFILES
  - SSRD command updated
  - Support iAP multiconnection (2 max)
  - IAP_OPEN_SESSION / IAP_CLOSE_SESSION notifications added for the iAP data session
  - iAP warnings fixed

- **Melody 6.0.55**
  - Fix a crash issue in iAP

- **Melody 6.0.54**
  - Add audio transfer for HFP (with CALL command)
  - Fix TONES command (sharp/flat notes)
  - Fix PIO_3 Audio Enable and A2DP_STREAM notification
  - Fix A2DP_STREAM notifications for TWS
  - Fix AGHFP (panic when SCO open from HF)
  - Fix AUDIO_ANALOG config
  - Fix iAP2 ACK timeout warnings

- **Melody 6.0.53**

- Add AUTO_DATA config to enter automatically in data mode for BLE/SPP/iAP links
- Remove UART flow control check when entering data mode (not recommended to use data mode without flow control).
- Fix VOLUME command when using ROUTE 4 and SPDIF audio input
- Update ADVERTISING command. Stop advertising not allowed if auto advertising is enabled.
- Allow max BLE MTU value up to 150.

- **Melody 6.0.52**
  - DFU over USB support for special builds.

- **Melody 6.0.51**
  - Add CONNECTABLE command.
  - Update POWER command with the possibility to turn on/off the Classic Bluetooth and BLE functionalities independently.
  - Update STATUS command to return the CONNECTED, CONNECTABLE, DISCOVERABLE and BLE state of Melody.
  - Add parameter to disable iAP1 support (in IAP_PARAMS).
  - iAP update: reduce UART flush timer to 100ms in data mode.
  - BLE fixes (ADVERTISING OFF command and sending data in Central mode).

- **Melody 6.0.50**
  - Remove BLE database size limitation (BLE_SET_DB).Will return an error if not enough memory is available.
  - Fix iAP, last packets not handled.
  - iAP data mode: reduce UART flush timer to 500ms.

- **Melody 6.0.49**
  - Add IAP_PARAMS to set the maximum packet length and maximum number of packet for iAP (MFI build only).
  - Fix iAP data transfer
  - Update test app for iAP (Melody Classic v2.0)
  - Fix UART (issue when buffer is full and when switching between operating modes)
  - Fix playback issue with Spotify and iPhone

- **Melody 6.0.48**
  - Add BLE role switching (possibility to switch the BLE role without reset)
  - Fix BLE data mode for central devices
  - Fix A2DP Source multi-connection
  - Fix enter/exit data mode with GPIO_5
  - Fix wait time after entering data mode with iAP High Speed
  - Fix iAP notifications in command mode
  - Fix PB_PULL command with multi-connection

- **Melody 6.0.47**
  - Add generic BLE functionality in Peripheral mode to support all BLE profiles.
  - Update BLE generic commands and notifications (Central and Peripheral mode).
  - Add TalkBack feature (A2DP)
  - Update DEEP_SLEEP config (can be changed without reset)
  - Add re-sampling 8k to 16k and 16k to 8k for HFP

- **Melody 6.0.46**
  - Update parser to support strings with spaces (SEND command, NAME...)
  - Add AVRCP Fast Forward and Rewind
  - Fix iAP config
  - Update VOLUME command for ROUTE 1/4
  - Fix Volume +/- with ROUTE 1/4
  - Update AUDIO_ANALOG codec output gain

- **Melody 6.0.45**
  - Fix SPDIF output
  - Fix SEND_RAW command
  - Fix AUDIO_ANALOG input gain
  - Fix Speech recognition
  - Fix PB_ABORT command
  - Fix BC_SMART_COMMAND
  - Update HID_READ notification
  - Update HFP codec status after connection
  - Update IAP_APP_REQ to use default protocol not provided as parameter

- **Melody 6.0.44**
  - Add Bluetooth volume configuration BT_VOL_CONFIG (default volume and number of steps)
  - Add balance control config BALANCE (adjust volume left/right)
  - Add special cases to ROUTE command and support for Music Manager and tones.
  - Update AUDIO_ANALOG configuration (input/output gain range and mic bias)
  - Fix PAIR notification for legacy pairing
  - Fix INQUIRY command (RSSI and max results parameters)
  - Fix APTX Low Latency codec
  - Fix BLE (panic when sending data and streaming music, default MTU and BLE/Classic state)

- **Melody 6.0.43**
  - Fix INQUIRY command
  - Fix PASSKEY command
  - Fix TONE command

- **Melody 6.0.42**
  - Fix DFU

- **Melody 6.0.41**
  - Update INQUIRY command, add maximum number of results
  - Enable CALLER_NUMBER notification from AGHFP
  - Possibility to change AUDIO config without reset
  - Fix SSP_CAPS

- **Melody 6.0.40**
  - Add Dual A2DP
  - Fix AGHFP audio parameters
  - Fix multiconnection issue
  - Fix SEND_RAW with SPP
  - Add AT commands with AGHFP
  - Disable AT commands by default
  - Fix BLE connection parameters

- **Melody 6.0.39**
  - Fix APTX with TWS
  - Update Inquiry command
  - Update BLE Advertising Flag
  - Add support for AT commands with AGHFP
  - Fix HFP/AGHFP multiconnection

- **Melody 6.0.38**
  - Add LINK_POLICY command (configure the power mode)
  - Add support for AGHFP and HFP concurrently
  - Add CALLER_NUMBER notification for incoming calls
  - Fix ROUTE command with AGHFP
  - Enable WBS for AGHFP
  - Update auto-connection
  - Update COD config (accept a 0 value)
  - Fix audio dropouts with Macbook Pro 10.10.5
  - Fix CLASS1 config
  - Fix AVRCP Meta data
  - Fix TWS disconnection

- **Melody 6.0.37**
  - Update AUDIO: possibility to select the interface for the input and output
  - Update AUDIO_ANALOG: remove rate parameter
  - Update I2S audio atten parameter
  - Update DISCOVERABLE config
  - Fix GPIO (PIO_0 / PIO_4 connection status)

- **Melody 6.0.36**

- Add support for Three Way Calling (CALL TWC)
- Fix MAP connection
- Fix MM_CFG / CVC_CFG commands
- Fix PIO4 config
- Fix iAP High Speed disconnection

- **Melody 6.0.35**
  - Fix for TWS issue

- **Melody 6.0.34**
  - Add CLASS_1 config
  - Update PIO_4 configuration
  - Update LINK_LOSS
  - Update AVRCP_META_DATA (possibility to request AVRCP Meta Data)
  - Fix HID issue
  - Fix BLE (BC Smart commands)
  - Fix iAP Data mode (disconnection)

- **Melody 6.0.33**
  - Update PIO4 config in GPIO_CONFIG
  - Fix SPP disconnection with High Speed feature
  - Disable AT commands in Data mode
  - Fix iAP Data mode
  - Fix SEND_RAW command when the link close before sending the data
  - Update DAC gain table with WB codec (fix volume issue)

- **Melody 6.0.32**
  - Add A2DP streaming state, AG/HFP call state and BLE MTU size to STATUS
  - Merge IAP_HIGH_SPEED and SPP_HIGH_SPEED in HIGH_SPEED (no need to reset Melody)
  - Merge ENABLE_GPIO_CONTROL and GPIO_ANALOG in GPIO_CONFIG
  - Add configuration to raise PIO4 on specific events (in GPIO_CONFIG)
  - Change BC_SMART_UUIDS to BC_SMART_CONFIG and add a parameter to disablecommands over BLE
  - Fix HID disconnection issue in multipoint and add OPEN_ERROR notification when fail to connect
  - Update SEND_RAW command to be able to receive notification while the status is PENDING
  - Fix issue when transmitting data over BLE and streaming music at the same time (reduce BLE throughput)
  - Fix I2S disconnection with A2DP Source

- **Melody 6.0.31**
  - Exit automatically Data mode when SPP/BLE/iAP connection is closed
  - Disable all notifications in Data mode (ROLE...)
  - Fix in I2S configuration (Unused values)

- **Melody 6.0.30**
  - Fix volume issue with HFP Narrow band
  - Fix HFP with Narrow band (Stereo output, I2S)

- **Melody 6.0.28**
  - Fix digital interface with HFP (no CVC / no WBS)
  - Small fix (Flow control check, parser issue)

- **Melody 6.0.27**
  - Add new filter to INQUIRY command
  - Add BLE to SEND_RAW command
  - Bug fixes (MUSIC_OLD_AVRCP, WBS with Siri, HID descriptor, BLE data mode)

- **Melody 6.0.26**
  - Bug fixes (AAC codec, TONES command with I2S, HFP volume, AGHFP terminate call)
  - Update VOLUME command to be able to change the volumes anytime (even if audio is not streaming for an A2DP link, or if there is no active call for an HFP/AGHGP link...)

- **Melody 6.0.25**
  - Fix MM_CFG command
  - Add support for TWS with analog input (ROUTE 1)

- • **Melody 6.0.24**
  - Update battery configuration, fix battery charging
  - Update TWS and Music Manager: allow slave to update audio enhancements/EQ
  - Add command IAP_APP_REQ (launch iOS app)
  - Fix minor bugs (BLE auto-advertising)
  - Update I2S configuration

- • **Melody 6.0.23**
  - Add Eddystone support
  - Add Cap Sense support
  - Fix small bugs (PIO 0 high when connected, ROUTE command, Music Manager)
  - Update TWS routing (no need to reset anymore)
  - Update PAIR command
  - Change mic bias (2.6V)
  - Change CVC plugin (handsfree)

- **Melody 6.0.22**
  - Add AVRCP_META_DATA command
  - Add SPP_HIGH_SPEED feature

- Add Beacon role (BLE_CONFIG) and BEACON_DATA configuration (only iBeacon for now)
- Small fixes (ROUTE command, exit iAp data mode, number of PBAP connections)
- Update BLE notifications
- Fix TWS audio drop-outs
- Update AVRCP playback status notifications

- **Melody 6.0.21**
  - Fix Tones
  - Fix AUDIO_ANALOG input gain
  - Update I2S re-sampling (with A2DP source only)

- **Melody 6.0.20**
  - Fix several BLE issues
  - Update BLE generic commands
  - Update SCAN notifications
  - Update LICENSE command
  - Update CODEC selection
  - Fix other minor issues

- **Melody 6.0.19**
  - Change EOF character (r instead of n)
  - Fix small issues (volume, SPP disconnection, notifications in data mode...)
  - Memory optimizations

- **Melody 6.0.18**
  - Auto connect AVRCP after A2DP
  - Add new event notifications
  - Fix Audio Enable PIO on non-MFI build
  - Add possibility to get the raw data with SCAN command (full advertising packet)
  - Add mic bias feature with A2DP
  - Use Analog default volume when using the ROUTE command
  - Melody becomes discoverable automatically when connecting to an unpaired device

- **Melody 6.0.17**
  - Fix several IAP/BLE/HFP bugs
  - Fix USB host
  - Modify default config (PROFILES and VREG_ROLE)
  - Auto reset after RESTORE

- **Melody 6.0.16**
  - Add IAP profile Enable mic bias by default (AUDIO_ANALOG)
  - Fix SSRD command
  - Fix AT command

  - Fix Name issue
  - Fix audio interface bugs

- **Melody 6.0.15**
  - Fix Enable Audio PIO (PIO_3)
  - Enable mic bias by default (AUDIO_ANALOG)
  - Update CLOSE command (possibility to use a device number as parameter)
  - Update the battery configuration (see BATT_CONFIG)