



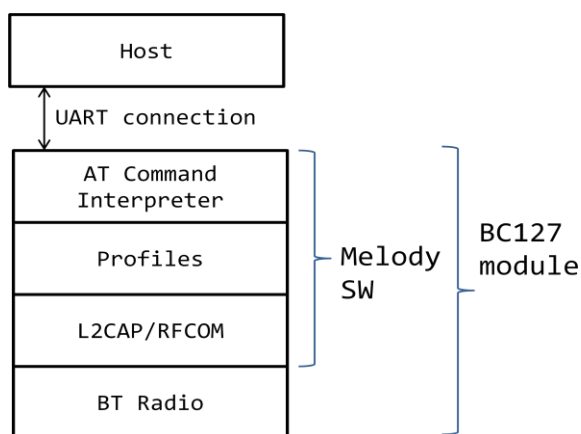
Key Specifications

- Best-in-Class Configurable Bluetooth 4.0 + 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: BC Smart, BC Proximity, DIS, PXP
- Multipoint Support
- True Wireless Stereo (TWS) support
- Supports data transfer over BLE, SPP, iAP1, and iAP2¹ protocols
- Supports SBC, AAC, aptX and MP3(optional) codecs
- Supports digital audio input and output I2S, PCM, SPDIF
- Supports analogue audio input and output from module ADC and DAC
- GPIO control, LED indications, capacitive touch sensor
- Future proof - uses the latest hardware and supports firmware upgrade (DFU)



Description

Melody is an embedded firmware running on the BC127 module. It offers both Classic Bluetooth and Bluetooth Low Energy functionality. It provides all functions needed for designing a wireless Audio device that can connect seamlessly to Smartphone's applications and deliver the best audio quality and user experience



Applications

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

¹ Only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees. Please, contact BlueCreation for more information.



Table of Contents

Key Specifications	1
Description	1
Applications.....	1
Introduction	5
Getting started	5
1. Setting up - Equipment	5
2. Default behavior.....	6
Link ID management.....	7
Operating Modes	8
Melody Commands	9
Melody Configuration	21
Power management.....	29
1. Battery configuration	29
PIO functionalities	30
1. GPIO control enabled	30
2. GPIO control disabled.....	31
3. Audio Enable PIO	31
4. Restoring the default configuration	31
5. Capacitive touch sensors.....	31
6. Class 1 device.....	31
LED Indications.....	32
Pairing and connection	33
1. Secure Simple Pairing capabilities	33
2. Pairing commands	33
3. Connection.....	33
4. Link Loss.....	34
Audio configuration	35
1. Analog interface	35
2. Digital interface	35
Bluetooth profiles	39
1. A2DP / AVRCP	39



Melody 6.0.39

2.	True Wireless Stereo	39
2.	HFP	40
3.	BLE	41
3.	General BLE functionalities	41
4.	Connection parameters	42
5.	BC Smart service	42
4.	HID	43
5.	iAP	44
6.	MAP	44
7.	PBAP	45
8.	SPP	47
Features		48
1.	Speech Recognition	48
2.	Device Firmware Upgrade	48
Appendix A: Event Notifications		49
Appendix B: Tone parameters and flags		52
Appendix C: Examples		55
1.	Pairing and connection with Melody	56
2.	SPP	57
3.	HFP	61
1.	Incoming call	61
2.	Outgoing call	61
3.	Call answer/reject	61
4.	Call transfer	62
5.	End call	62
6.	Three way calling	62
4.	PBAP	64
5.	MAP	65
6.	Music source (A2DP & AVRCP)	66
7.	HID	67
8.	Multi-connection and audio routing	70
9.	True Wireless Stereo (TWS)	71
10.	BLE (BC Smart Service)	72



Melody 6.0.39

11.	BLE central (Generic commands).....	73
12.	Beacons	74
13.	Tones	75
14.	Link Policy.....	76
Appendix D: Error codes.....		77
Terms and definitions		78
General Notes		79
Change Log		80



Introduction

Melody is an embedded firmware solution running on the BC127 module. It includes an application, the Bluetooth profiles and the Bluetooth protocol stack both for Classic Bluetooth and Bluetooth Low Energy. It therefore allows using the BC127 module to develop a Bluetooth device without detailed knowledge of the Bluetooth standard. Melody provides a simple high level UART-based command interface. So the Bluetooth module can be controlled from a host processor.

Getting started

1. Setting up - Equipment

To start you need to have:

- A Bluetooth Development board such as BC127-DEVKIT. Please contact sales@bluecreation.com for more information.
- 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:

- Baud rate : 9600bps
- Data bits : 8
- Stop bits : 1
- Parity bit : No parity
- HW Flow Control : Disabled

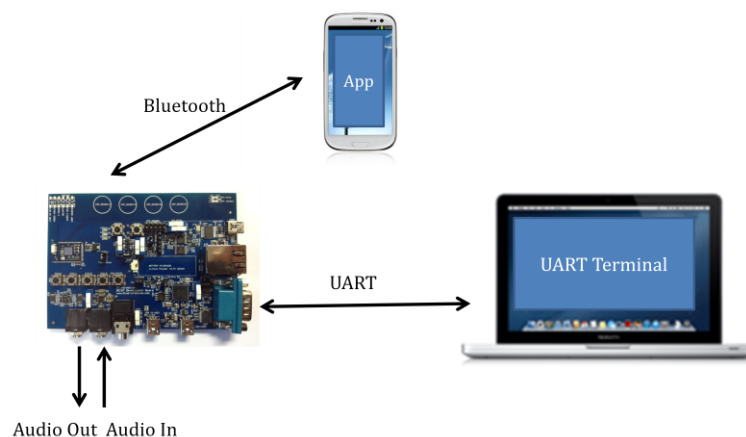


Figure 1: Example configuration



Melody 6.0.39

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 expected by Melody is a Carriage Return ('\r' or 0x0D).

You are now ready to control the module!

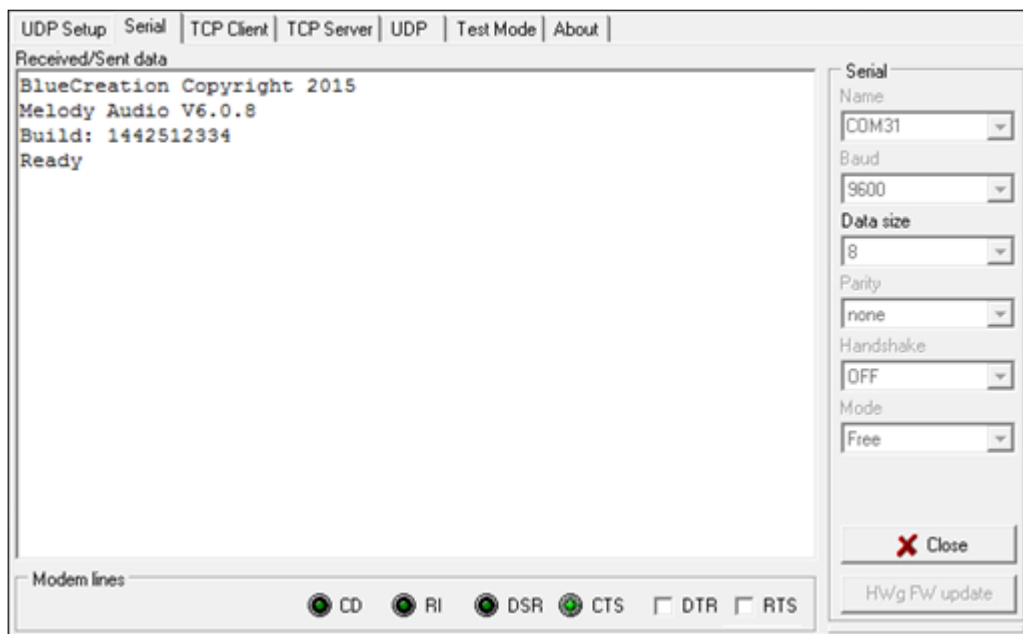


Figure 2: Melody initial load

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
- The module is preloaded with Melody firmware

2. Default behavior

When Melody boots with factory default settings, it is discoverable. Any Bluetooth enabled phone or other Bluetooth device can see it when scanning (discovering) Bluetooth devices.

Melody will appear as 'BC-XXXXXX'² and once connected, it will transparently transfer any audio stream to the audio interface (DAC, I2S, PCM, SPDIF or USB).

² XXXXXX will be six hexadecimal digits that correspond to the last four hexadecimal digits of the device Bluetooth address. If you are using the BC127-DEVKIT, the LED will be flashing alternatively to indicate It is Discoverable.



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:

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

Table 1: Device 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)
B-F	Reserved for Future Use

Table 2: Profile Field values

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:

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

Table 3: Reserved values for Profile Field



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](#) 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. Note that it is still possible to interact with Melody by using another link (BLE or SPP) to send commands.

Data transfer can be speeded up with the high speed feature for SPP (~700kbps) or iAP(~500kbps). For BLE connections, the MTU size negotiated between Melody and the remote device has an influence on the rate (~75kpbs from a peripheral to a central device with a MTU of 100).

To switch between Melody operating modes the following commands can used. PIO_5 can also be used if GPIO control is disabled (see [PIO functionalities](#)).

Command	Switch from:	Switch to:
\$\$\$\$	Data Mode	Command Mode
ENTER_DATA_MODE [link_id]	Command Mode	Data Mode

Table 4: Operating mode commands



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.

UART Command	Description
\$\$\$\$	Exit Data mode. Must be sent through the UART interface.
ADVERTISING (mode)	ON starts BLE advertising and OFF stops BLE advertising. Please note that this command is only available when the device is configured as peripheral.
AT (link_ID) (command to send)	Send a command over the HFP link to the remote device.
AVRCP_META_DATA (link_ID)	Request the AVRCP Meta Data specified in a remote device.
AVRCP_META_DATA (link_ID) [(type) (data)]	<p>This command can be used to get or set the AVRCP Meta Data:</p> <p>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.</p> <p>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.</p> <p>link_id: AVRCP link</p> <p>type: attribute type</p> <ul style="list-style-type: none"> 1 – Title 2 – Artist 3 – Album 4 – Number 5 – Total number 6 – Genre 7 – Playing time <p>data: attribute data (string)</p>



Melody 6.0.39

BATTERY_STATUS	Returns the battery status of the battery attached to the module by detecting the voltage level
BC_SMART_COMMAND (link) (command)	<p>Send a command over BLE.</p> <p>This command is allowed only when there is a BLE connection opened and Melody role is in central.</p>
BC_SMART_NOTIF (link) (data) (cmd)	<p>Enable/disable notifications on the Data and Command characteristics of the BC Smart service.</p> <p>This command is allowed only when there is a BLE connection opened and Melody role is in central.</p>
BLE_GET_SERV (link)	<p>List all the primary services in the following format: BLE LINK TYPE UUID [HANDLE_START HANDLE_END]</p> <p>TYPE: U16(16bits UUID) or U128(128bits UUID) HANDLE: value in hex between 0x0001 and 0xFFFF</p> <p>This command is allowed only when there is a BLE connection opened and Melody role is in central.</p>
BLE_GET_CHAR (link) [handle_start] [handle_end]	<p>List the characteristics between handle_start and handle_end in the following format: BLE LINK TYPE UUID [HANDLE]</p> <p>TYPE: U16(16bits UUID) or U128(128bits UUID) HANDLE: value in hex between 0x0001 and 0xFFFF</p> <p>If there is no parameter, lists all the characteristics.</p> <p>This command is allowed only when there is a BLE connection and Melody is in central mode.</p>
BLE_NOTIF (link) (handle) (value)	<p>Enable notifications of the characteristic associated to the handle when value=ON. Disable notifications when value=OFF.</p> <p>Melody will return OK in case of success or an error code if the operation fails (invalid handle).</p> <p>This command is allowed only when there is a BLE connection and Melody is in central mode.</p>
BLE_READ (link) (handle)	<p>Return the value of the characteristic associated to the handle in the following format: BLE LINK [HANDLE] SIZE DATA</p> <p>HANDLE: value in hex between 0x0001 and 0xFFFF SIZE: nb bytes of data DATA: in hex</p> <p>Melody will return an error if the operation fails (invalid handle, read no permitted...).</p> <p>This command is allowed only when there is a BLE connection</p>



Melody 6.0.39

	and Melody is in central mode.
BLE_WRITE (link) (handle) (size) (data)	<p>Write command on a characteristic.</p> <p>Size is the number of bytes of the data; each byte in data is separated by a space.</p> <p>Please note that the maximum size (MTU-3) depends on the MTU value negotiated when establishing the connection with the remote device.</p> <p>Melody will return OK in case of success or an error code if the operation fails (invalid handle, write not permitted, invalid length..).</p> <p>This command is allowed only when there is a BLE connection and Melody is in central mode.</p>
CALL (link) (action) [param]	<p>Call interactions with the specified HFP or AGHFP link.</p> <p>For an HFP link, <i>action</i> can be any of the following:</p> <ul style="list-style-type: none">- REDIAL: Notifies the AG that the HF wants to establish an outgoing call with the last number dialled.- MEMORY: Notifies the AG that the HF wants to establish an outgoing call using memory dialling. In this case, <i>param</i> 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 <i>param</i>- ANSWER: Answer an incoming call from the HF.- REJECT: Reject an incoming call from the HF.- TWC: Three way calling. Send a request to the AG to perform an action based on param. param can be:<ul style="list-style-type: none">▪ 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. <p>For an AGHFP link, <i>action</i> can be any of the following:</p> <ul style="list-style-type: none">- 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 <i>param</i> is not provided, Melody will decide depending on where the



Melody 6.0.39

	<p>audio is currently routed.</p> <ul style="list-style-type: none"> - INCOMING: Setups the AG in incoming call mode for the number specified in <i>param</i>. Number can be any combination of alphanumeric characters. - OUTGOING: Setups the AG in outgoing call mode for the number specified in <i>param</i>. 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
CVC_CFG (type) [(key) (length)]	<p>Read or write the CVC configuration stored on the module.</p> <p>type can be WB or NB. If only type parameter is supplied all 4 config keys for this type will be displayed.</p> <p>If 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.</p> <p>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.</p> <p>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.</p>
DFU	Causes a reset to make the device boot in DFU ³ mode
DISCOVERABLE (mode)	ON puts the module in discoverable mode, OFF makes the device non discoverable.
ENTER_DATA_MODE (link)	<p>Configure the behaviour of the specified link to be in Data mode.</p> <p>The parameter <i>link</i> can be any active BLE or SPP link.</p> <p>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.</p> <p>Note that to use Data mode, you must enable hardware flow control on the UART interface, otherwise Melody will return an</p>

³DFU allows downloading a new firmware upgrades onto the Bluetooth module via the UART interface and allows users to upgrade melody to new releases.

Melody 6.0.39

	error.																
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.																
IAP [Parameter=value]	<p>Set iAP parameters. These are the parameters that identify the device for iOS and iOS application. Maximum sizes of configurations in bytes/chars:</p> <table> <tr> <td>ACCESSORY_NAME</td><td>48 characters</td></tr> <tr> <td>MANUFACTURER_NAME</td><td>48 characters</td></tr> <tr> <td>MODEL_NAME</td><td>48 characters</td></tr> <tr> <td>SERIAL_NO</td><td>16 characters</td></tr> <tr> <td>SEED_ID</td><td>10 characters</td></tr> <tr> <td>PROTOCOL_STRING</td><td>48 characters</td></tr> <tr> <td>HARDWARE_VER</td><td>3 characters</td></tr> <tr> <td>FIRMWARE_VER</td><td>3 characters</td></tr> </table> <p>If no parameter is provided, all parameters are displayed.</p> <p>Refer to the Application Note on iAP for more information.</p>	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
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																
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) [type filter]	<p>Searches Bluetooth devices in the area for maximum period of time (equal to 1.28s * timeout). Timeout takes a value between 1 and 48. If the command succeed, it will return PENDING and a maximum of 16 inquiry results will be displayed as inquiry notifications.</p> <p>Results can be filtered by providing the optional parameters type and filter:</p> <ul style="list-style-type: none"> - type=0 Filter the results by name. filter is a string, for example, if the string is "BC-", only devices with a name starting with "BC-" will be shown. - type=1 Filter the results by class of device. filter is the COD. - type=2 Filter the results with an RSSI value in dB, for example -72. filter is the min value. 																



Melody 6.0.39

	<ul style="list-style-type: none">- type=3 Show only TWS devices. filter is the ON or OFF.
LICENSE [type] [=value]	<p>Sets or returns the APTX or CVC license.</p> <p>Type can be any of the following:</p> <ul style="list-style-type: none">- CVC: Sets or gets the CVC license.- APTX: Sets or gets the APTX license. <p>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.</p> <p>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.</p>
LINK_POLICY (device) (nb_entries)	<p>Sets a link policy power table for a connection. By default Melody uses its own default values. But theses can be changed with this command. The link policy power table allows the device to switch between different power modes.</p> <p>device: index of the remote device nb_entries: Number of entries in the power table (max. 8)</p> <p>This command will return PENDING and expect the entries separated by <CR>.</p> <p>Each entry shall have the following format in hex: (state) (min_interval) (max_interval) (attempt) (timeout) (time)</p> <p>state: The power mode 00 : active mode 01: sniff mode FF : passive mode</p> <p>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.</p> <p>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.</p> <p>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.</p>



Melody 6.0.39

	<p>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.</p> <p>time: The time spent in this state of the power table, in seconds. This must be 0 for the last entry in the table.</p> <p>The command returns OK when all the entries have been successfully retrieved (see example).</p>
LIST	Lists paired devices in the format LIST (BT_addr) (Space separated list of supported profiles).
MM_CFG (key) [(length)]	<p>Read or write the Music Manager configuration stored on the module.</p> <p>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.</p> <p>If length is also present, the key indicated will be written with a value of size length.</p> <p>length is in 16 bits word and can be 0 - 64. 0 will delete the key.</p> <p>The MM_CFG command will reply with PENDING.</p> <p>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.</p> <p>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.</p>
MUSIC (link_ID) (instruction)	Controls the music stream state and sends AVRCP instructions (PLAY, PAUSE, STOP, FORWARD, BACKWARD)
NAME (BT_addr)	Returns the friendly name of device with the provided Bluetooth address
OPEN (BT_addr) (profile) [type]	<p>Establishes a connection with a given Bluetooth address for a certain profile.</p> <p>BT_addr: Bluetooth address formatted as 12 Hexadecimal digits with no separators (e.g.: 3859F9CCB893).</p> <p>profile: Bluetooth profile. Can be: HFP, AGHFP, A2DP, AVRCP, BLE, SPP, MAP⁴, HFP, PBAP⁵, HID or IAP. If profile is left blank, SPP will be assumed.</p> <p>type (optional): Type of address</p>

⁵ PBAB requires an active HFP connection.

Melody 6.0.39

	0 – public address 1 – random address Note that type is only used for BLE connections. It corresponds to the type in the scan results.
PAIR (BT_addr)	Enter 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. Use type = 1 for a confirmation type and key = 1 to accept or key = 0 to reject the pairing. Use type = 0 and followed by 6 digit passkey to confirm the pairing with your key.
PIO (PIOx) (state)	Sets the specified PIO pin to the specified state. <ul style="list-style-type: none"> - 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 (mode)	OFF disconnects all active connections and puts the device in limbo mode, where it is not connectable, or discoverable. As a consequence, the UART commands that cannot be executed are rejected. ON returns the device to a connectable state.
PB_PULL (link_id)<repository><phonebook><max list><start index><filter>⁶	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, look at the PBAP section.
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

⁶The phonebook download requires a baud rate of 115200 or above. Lower baud rates can cause the UART to stall and lose phonebook data



Melody 6.0.39

RESET	Resets the device
RESTORE	Resets and restores the configuration parameters to default factory settings.
ROLE (link_id) [role]	<p>Changes the current role for a particular connection. If role is not provided, a notification will be sent with the current role of the local device.</p> <p>link id: A2DP, AVRCP, HFP, SPP or iAP link.</p> <p>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.</p>
ROUTE (value/link_id)	<p>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.</p> <p>You can use either use a link_id (A2DP or HFP only) or the following values to route the audio stream: 0 – Enable automatic routing 1 – Analogue input to Analogue output 2 –Digital interface to Analogue in/out 3 – Digital Interface to Digital Interface 4– USB to Analogue in/out 5 – USB to Digital Interface</p> <p>Note1: Routing an A2DP link is possible only if music is streaming. Routing an HFP link is only possible during a call. If the routing is not possible or already done an error is returned. Note2: Upscaling/downscaling is not supported when routing the Digital interface to the Analogue interface. They both need to work at the same frequency.</p>
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]	<p>Searches BLE-enabled devices in the area for maximum period of time (given in seconds).</p> <p>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.</p> <p>Following this command, a SCAN notification is received when an advertising device is found.</p>

Melody 6.0.39

	Please note that this command is only available when the device is configured in central mode.
SEND (link) (string)	Sends a string to a device through the specified profile. The parameter <i>link</i> can be any active SPP, BLE link or iAP link.
SEND_RAW (link) (nb_bytes)	Define the number of bytes that will be sent to the device through the specified profile. The parameter <i>link</i> 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) (raw_data)	<p>Set the scan response data. This is a manufacturer specific data blob. The first 2 bytes need to be Bluetooth SIG manufacturer ID to be SIG compliant.</p> <p>Size – the size of the data in decimal (including company id). Size must be <= 29. Size 0 with no additional arguments deletes the current Scan response data</p> <p>Raw_data – SIZE number of space separated ASCII represented unsigned octet values.</p> <p>Structure is: [company_id LSB] [company_ID MSB] [SIZE-2 data bytes in manufacturer determined order]</p> <p>Eg. SSRD 8 01 23 45 67 89 AB CD EF</p> <p>Please note that this function will return error if Melody is not configured in BLE Peripheral mode; for wrong size, or if the function cannot allocate size number of bytes when called.</p> <p>If function successfully performs internal actions “PENDING” result will be returned, followed by an asynchronous “OK” or “ERROR” status once the scan data registration request completes.</p> <p>Please note that this function may be called at any time, and will overwrite any pre-existing registered scan response data.</p>
STATUS (param)	<p>Lists the device status for Classic and BLE:</p> <p>STATUS (Classic status) (BLE status)</p> <p>Where Classic status can be: OFF, CONNECTABLE, DISCOVERABLE, CONNECTABLE or CONNECTED. BLE status can be: OFF, IDLE, ADVERTISING or CONNECTED.</p>



Melody 6.0.39

	<p>And Link status for established links in the format:</p> <pre>LINK (link_id) (state) (profile) (BT_Addr) (Additional info)</pre> <p>Where state can be: CONNECTED, DISCONNECTED or LINK_LOSS.</p> <p>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.</p> <p>Param can be any of the following:</p> <ul style="list-style-type: none"> - A link. - A profile identifier. (SPP, HFP, A2DP...) - A device identifier (RES, DEV0, DEV1, DEV2) <p>When param is supplied, the status command returns the relevant information associated to the value of param.</p>
TOGGLE_VR (link)	<p>Start/Stop Voice call command on the phone. The link provided must be an HFP link.</p>
TONE⁷ (flag) (value) (flag) (value) ...	<p>Plays a tone based on the input. A tone must have at least 1 note. A note must have a length parameter.</p> <p>Please refer to Appendix B for full information regarding, flags, accepted values and descriptions.</p> <p>Flags :</p> <p>Tempo TE 0 - 4095 Timbre TI 0 - 7 Volume V 0 - 255 Decay D 00 - FF(enter value in hex⁸) Note N A - F + octave 0 - 9eg AF7, A7, AS7 Length L 1,2,4...64 or 3,6,12...96 triplets</p> <p>Please note that this command will return an error if you have used the ROUTE command to route the analog, digital or USB interface (ROUTE 1, 2, 3, 4 or 5).</p>
UNPAIR [addr]	<p>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.</p> <p>If no parameters are provided, clears all of the Bluetooth</p>

⁷ Currently not supported over digital audio output.

⁸ Parsed as fixed point of the following format in binary bbbb.bbbb or hexadecimal X.X



Melody 6.0.39

	addresses from the paired devices list.
VOLUME [<i>link</i>] [<i>command</i>]	<p>When the VOLUME command is used with no parameters, it will list the volumes of the ANALOG and A2DP, HFP and AGHFP profiles currently connected.</p> <p>The parameter <i>link</i> can be any ANALOG, A2DP, HFP or AGHFP link that is currently routed. When a <i>link</i> is provided, a <i>command</i> must be provided as well.</p> <p>The parameter <i>command</i> can be any of the following:</p> <ul style="list-style-type: none">- UP, increases the volume on the specified link by one.- DOWN, decreases the volume on the specified link by one.- An hexadecimal value between 0x00 and 0x0F, sets the volume on the specified link to the value specified. <p>If when increasing the volume the resulting volume is greater than the maximum volume allowed on the channel, the volume is fixed to the maximum level allowed. Similarly, when decreasing the volume, if the resulting volume is lower than the minimum volume allowed for the channel, the volume is fixed to the minimum volume allowed. These same rules apply when setting the volume to a specified value.</p>
VERSION	Returns information on the firmware version number and the Bluetooth address of the device.
WRITE	Store configurations.

Table 5: Command Mode Bluetooth Commands



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 4 main commands to configure parameters. The commands to modify configuration parameters are described below. Mandatory parameters are listed in "()" optional parameters are listed in "[]".

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.

Table 6: General Configuration Commands

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 the table below. Once modified, many configuration parameters will not take effect before a reboot. Therefore they need to be stored to Flash before rebooting.

Configuration Parameter	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	Yes
AUDIO_ANALOG=(gain) (def_vol) (micbias) (preamp)	Configuration of the analog audio interface(AUDIO=0): gain: Input gain (0 – 31)	15 10 ON OFF	Yes



Melody 6.0.39

)	def_vol: Default volume level (0 – 15) micbias: Enable mic bias (2.6V) pream: Enable microphone preamplifier		
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
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. <i>Note:</i> Melody can store up to 8 paired devices in its persistent memory. When the pairing memory is full, Melody erase the oldest paired device and keep the 8 most recently paired devices in memory.	0	Yes
BATT_CONFIG=(status) (critLvl) (chrgLvl) (vthm_max) (current)	Configures the battery charger and other settings related to the battery: <ul style="list-style-type: none"> - 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



Melody 6.0.39

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: 0- iBeacon 1- Eddystone UID 2- 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
BLE_CONFIG=(role) [auto_adv] [indication] [mtu] [random_addr]	role: Set the BLE role of the device 0- Peripheral 1- Central 2- Beacon auto_adv: Enable/Disable BLE constant advertisements when the device is not connected (only when BLE role is Peripheral) indication: If ON, Melody Smart will use indications instead of notifications when configured as server (only when BLE role is Peripheral) mtu: Maximum transmission unit (effectively maximum packet length) random_addr: Generate a random address for BLE connections	0 OFF OFF 40 ON	Yes
BLE_CONN_PARAM	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
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.	240404	No



Melody 6.0.39

	Some common examples of COD values are the following: <ul style="list-style-type: none">- 240404: Wearable Headset Device- 200404: Audio headphones		
CODEC= (value)	The value specifies which A2DP optional codecs are enabled. bit[0] – AAC bit[1] – aptX bit[2] – aptX Low Latency 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 LL > aptX > AAC > SBC	0	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= (value)	Enable and Disable Deep Sleep	OFF	No
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. Mode: 0 – Not discoverable at power on 1 – Always Discoverable 2 – Discoverable at power on Timeout: Pairing timeout in seconds. Zero to disable.	2 0	Yes
ENABLE_BATT_IND= (value)	Enables and disables the iOS battery indication ⁹	ON	No
ENABLE_CAPSENSE= (value)	Enables and disables cap senses. When it is enabled,	OFF	Yes

⁹ Due to iOS behaviour, this needs to be changed before initial Pairing to iOS device. Changing after that will not remove indicator on iOS.



Melody 6.0.39

)	CAPSENSE notification are received when the state of a pad changes.		
ENABLE_LED=(value)	Enables and disables LED support in Melody	ON	No
ENABLE_SPP_SNIFF=(state) (min_interval) (max_interval) (attempt) (timeout) [duration]	<p>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</p> <p>state:</p> <p>OFF – active</p> <p>ON – passive (if all parameters are 0)</p> <p>– sniff with passive (with all parameters supplied)</p> <p>Parameters:</p> <p>min_interval: the minimum interval of the sniff period</p> <p>max_interval: the maximum interval of the sniff period</p> <p>attempt: Determines for how many slots the slave shall listen when the slave is not treating this as a scatternet link.</p> <p>Timeout: Determines for how many additional slots the slave shall listen when the slave is not treating this as a scatternet link.</p> <p>Duration: the time to stay in this sniff state.</p>	OFF 0 0 0 0	No ¹⁰
GPIO_CONFIG=(ctrl) [pio4_cfg] [analog]	<p>GPIO configuration.</p> <p>ctrl: Enables/disables GPIO control.</p> <p>pio4_cfg: Bitmask to select PIO_4 functionality (only if GPIO control disabled):</p> <p>If an event is selected, PIO4 is raised when the event occurs. You can clear it with the PIO command (“PIO 4 OFF”).</p> <ul style="list-style-type: none"> - 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 (equivalent to PIO_0) (state) - Bit9: AVRCP_MEDIA (event) - Bit10: CALL_IDLE or CALL_ACTIVE (event) <p>Value expected in hexadecimal. (ex: ‘003’ to raise PIO_4 on all the A2DP events).</p>	ON 0 254	Yes

¹⁰ Changing the sniff parameters require the connection to restart.



Melody 6.0.39

	analog: Configures the analog input detection pin. Value can be any of the following: <ul style="list-style-type: none"> - 0 to 31: Pin 0 to 31. - 254: Disabled. - 255: Assume always connected. The analog input detection pin is active low.		
HFP_CONFIG=(cvc) (wbs) (inband) (nrec) (comm)	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. comm: Enables or disables AT commands communications from the host. 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 ON	Yes
HIGH_SPEED=(spp) (iap)	Enable/Disable High Speed for SPP and iAP. 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
LOCAL_ADDR=(value¹¹)	Contains the local Bluetooth address of the device.	-	-
MAX_REC=(value)	Maximum times to try to reconnect to a device in PDL	2	Yes
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)	OFF OFF 0 OFF OFFOFFOFFFO FF	No

¹¹ Please note that value is a read only.



Melody 6.0.39

	<p>you have to supply (user_eq_en) and (bank).</p> <p>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. 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</p> <p>Please note that these configurations only enable/disable music manager blocks. Those need to be configured use UFE, as specified by CSR.</p>		
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) (avr_cp) (ble) (spp) (pbap) (hid_device) (hid_host) (map) (iap)	<p>Configuration of the Bluetooth profiles. The value for each profile indicates the maximum number of connections (up to 3).</p> <p>Notes:</p> <ul style="list-style-type: none"> - 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 AGHFP are limited to 2 connections. - iAP is limited to a single connection.¹² 	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

¹² Only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees. Please, contact BlueCreation for more information.



Melody 6.0.39

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
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 5 – reject SSP request	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, 921200). flow_ctrl : Enable or disable flow control. parity : UART Parity (0=none, 1=odd, 2=even).	9600OFF 0	No Yes No
USB_HOST= (value)	Determine the host interface: value is ON for USB and OFF for UART.	OFF	Yes
VREG_ROLE= (value)	Changes behaviour of VREG button when held (3s press) 0 – Nothing 1 – Power On/Off	1	No

Table 7: Configuration Parameters and values



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 connected, 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: If Deep Sleep is enabled, Melody will go into deep sleep after 1000ms of no activity.

1. 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 ($145 \times 20 = 2900\text{mV}$ 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 AIO1¹³ input. Melody will not charge if the value is superior to this limit (1350mV by default).

¹³ Note that AIO readings saturate above 1.35V

PIO functionalities

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

1. 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.

GPIO ¹⁴	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 ¹⁵	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 ¹⁵	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)

Table 8: Command Mode General PIO Bluetooth Commands

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.

¹⁴ Refer to BC127 Datasheet for location of PIO on the module pinout.

¹⁵ Only on non-MFI builds.



Melody 6.0.39

2. GPIO control disabled

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

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

Table 9: PIO control and indication

3. 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.

4. 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.

5. 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.

6. 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. Contact BlueCreation for more informations.



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:

Classic State	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 10: LED Patterns Classic

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: LED Patterns Smart

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

Table 12: Other indications



Pairing and connection

1. 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 'Just Works' 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.

2. Pairing commands

You can use the PAIR command to send a pairing request. A notification will be received (PAIR_OK/PENDING/ERROR or PAIR_PASSKEY). When you pair with a device it is automatically added to the Paired Device List (PDL) that can be seen using the LIST command. Finally you can remove any device from your pairing list using the UNPAIR command.

3. 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.



4. 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 (ex: "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 (ex: "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 (ex: "CLOSE_OK 13 HFP").

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



Audio configuration

Melody has different 2 types of audio interfaces that are configurable. You can select the audio interfaces (Analog or Digital) for the input and output separately in the **AUDIO** parameter.

1. Analog interface

The analog configuration is stored in the parameter **AUDIO_ANALOG**:

AUDIO_ANALOG = *gain def_vol micbias preamp*

- *gain*: Input gain (0-31).
- *def_vol*: Default volume level (0-15). *def_vol* is the volume used when establishing an A2DP or HFP connection for the first time with a device or when selecting the analogue interface with the ROUTE command. The volume can then be changed using the VOLUME command. When disconnecting and reconnecting, the current volume value is saved and restored.

Note that with devices using Absolute volume, this value is not used.

- *micbias*: ON to activate the mic bias (2.6 V).
- *preamp*: ON to enable the preamp.

2. 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 Narrow Band (8KHz) / Wide Band (16kHz) and for A2DP when the rate is superior or equal to 44100 Hz.
- *param1*: Bit clock scaling factor for I2S. Master clock in kHz for PCM. Not used for SPDIF.
- *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).

Please note:

In Slave mode, due to hardware limitations, a Bit clock (BCLK) of at least 4*bps*WCLK and up to 256*WCLK must be supplied.



Melody 6.0.39

In Master mode we also recommend to generate a Bit clock of at least $4 \times \text{bps} \times \text{WCLK}$. The Bit clock scaling factor should be superior or equal to $4 \times \text{bps}$.

The following tables describe param2 of the digital interface for each 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.



Melody 6.0.39

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 frame 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



Melody 6.0.39

some CODECs (connected to the PCM interface) to allow their gain to be controlled.

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.



Bluetooth profiles

All the Bluetooth profiles can be enabled or disabled by changing the PROFILES configuration.

1. A2DP / AVRCP

A2DP(v1.3) and AVRCP(v1.6) profiles can be enabled by setting up their maximum number of connections in the parameter PROFILES. They can both have up to 3 connections. 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). Two commands allow you to control the audio streams and select the one to route when there multiple devices connected. The VOLUME command can be used to adjust the volume for each link independently.

SBC is the codec used by default, additional codecs can be enabled with the CODEC configuration (note that aptX requires a license key).

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

2. True Wireless Stereo

Melody 6 includes True Wireless Stereo (TWS) support. 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 True Wireless Stereo speakers over Bluetooth.

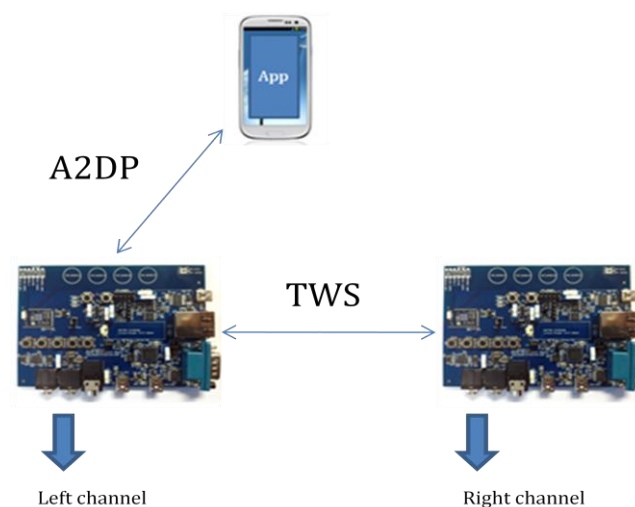


Figure 3- TWS connection between two melody boards

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



2. HFP

Melody supports the HFP profile (v1.6). 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 connections and up to 2 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).



Melody 6.0.39

3. BLE

3. General BLE functionalities

Melody allows BLE connections. It supports three BLE roles: peripheral, central and beacon (see BLE_CONFIG).

The profiles available are BC Smart, BC Proximity and PXP profile.

Peripheral:

When Melody is in Peripheral mode (role=0), the ADVERTISING command can be used to start / stop advertising. The advertisement data include a name (NAME_SHORT) and the UUID of the BC Smart service (BC_SMART_CONFIG). Note that if Melody uses a random address it will not advertise as a Dual Mode device. The scan response can also be set using the SSRD command. Melody can connect to only one central device.

Central :

In Central mode, you can use the SCAN command to look for advertising devices and the OPEN command to establish a connection. Melody can connect up to 3 peripheral devices. Note that when you try to connect to a device that uses a random address (see "type" in the scan result) you need to provide the optional parameter of the command to indicate the type of the address (by default it is public address). A set of generic BLE commands (starting by "BLE_") gives the possibility once connected to a peripheral device to retrieve the list of all its services and characteristics and to read or write their values. See [BLE central example](#).

Beacon:

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

	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 bytes)	encoded URL (17 bytes)	



Melody 6.0.39

4. Connection parameters

The BLE connections parameters can be modified by changing the values in the parameter BLE_CONN_PARAM. These 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_window:** Amount of time for the duration of the LE scan. LE Scan Window shall be less than or equal to LE Scan Interval.
- **conn_interval_min:** Minimum value for the connection event interval. This shall be less than or equal to Connection Interval Max.
- **conn_interval_max:** Maximum value for the connection event interval. This shall be greater than or equal to Connection Interval Min.
- **conn_latency:** Slave latency for the connection in number of connection events.
- **supervision_timeout:** The timeout before disconnecting when no communication is present on the lower layers.
- **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)
- **adv_interval_max:** Maximum advertising interval for non-directed advertising.
- **conn_latency_max:** Maximum allowed slave latency
- **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.

5. BC Smart service

The BC Smart service can be used to send easily data and commands over BLE. See [BC Smart example](#).

The UUIDs of the BC Smart service and its Data characteristic are store in the parameter BC_SMART_CONFIG.



Melody 6.0.39

4. 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.

The

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.



Melody 6.0.39

5. iAP

This profile is only on MFI builds. iAP1 and iAP2 protocols are available to Apple MFI Licensees. Contact BlueCreation for more information.

Please refer to the iAP Application Note to configure iAP.

Once configured it is possible to open the iAP profile and send data using the SEND command. It is also possible to send data in Data mode (see [Operating Modes](#)). Note that in Data mode, transfer can be significantly faster with iAP2 when using the High Speed feature (IAP_HIGH_SPEED config).

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

6. 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.



Melody 6.0.39

7. 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]

LINK ID

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

REPOSITORY

Value	Description
0	Not used, will default to 1
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

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.

Melody 6.0.39

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).

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



Melody 6.0.39

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	Time stamp

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.

8. SPP

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](#)).

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

Data received over SPP are showed with the RECV notification.



Features

1. 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 these 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 being streamed the audio won't come out while the Speech Recognition is active.

Speech recognition is not available with cVc.

2. Device Firmware Upgrade

Melody support firmware upgrade. This can be done over UART from your PC using the Melody Device Firmware Upgrade Tool available on our website. Please contact sales@bluecreation.com for more details.

You can use DFU to upgrade to a newer version of Melody and get additional features.

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



Appendix A: Event 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)**.

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
BC_SMART PROX (value)	BC Smart Proximity Service indicates that this is the closest (1) or not (0) device. If an indication of 0 is received the remote side may disconnect.
BLE_NOTIF link_ID [handle] size data	Notification received on the specified link. Size is a decimal value. The data received and the handle are in hexadecimal.
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.

Melody 6.0.39

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 dialled.
CALLER_NUMBER [link_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.
CLOSE_ERROR (profile) [link_ID]	A connection could not be closed for the profile 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.
DTMF 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 the error codes in the appendix to find out more about the issue.
INBAND_RING AGHFP [link_ID]	Ring indicator received periodically when there is an incoming call.
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 (profile) [link_ID]	A connection has failed or there was a link loss for the profile indicated
OPEN_OK (profile) [link_ID]	A connection has been successfully established for the profile indicated
PAIR_ERROR (Bluetooth)	When the module fails to pair with the remote device with the address

Melody 6.0.39

address)	specified
PAIR_OK (Bluetooth address)	When the module successfully pairs with the remote device with the address specified
PAIR_PASSKEY (BDADDRESS) (TYPE) [PASSKEY]	A 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
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
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.
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_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.
---------------------	---

Table 13: Melody Notifications

Appendix 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.

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. Eg 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.



Melody 6.0.39

Note	N	<p>This describes a note pitch in the format:</p> <p>Note{F/S} [octave]</p> <p>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 eg R0.</p> <p>The optional F or S following that indicates if the note is Flat or Sharp. Note, not all notes have flat and sharp variants.</p> <p>The mandatory octave parameter gives the note octave. It takes values between 0 – 9.</p> <p>A note is describes as G4 or AS7 for example.</p> <p>Each note must have a length parameter associated with it and following immediately after:</p> <p>G4 L 1</p>
Tied Note	TN	<p>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.</p> <p>Tied Notes are described in the same manner as Notes.</p>
Length	L	<p>Ringtone duration.</p> <p>Please look at Table 10 for details.</p>

Table 14: TONE parameter detailed description

Table 15: Timbre value 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



Melody 6.0.39

64	Hemidemisemiquaver	Sixty-fourth 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
48	Demisemiquaver triplet	Thirty-second note triplet
96	Hemidemisemiquaver triplet	Sixty-fourth note triplet

Table 16: Note length values and description



Appendix C: 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 2015  
>Melody Audio V6.0.10  
>Build: 1443202002  
>Ready
```



Melody 6.0.39

1. Pairing and connection with Melody

Discovering Melody from a phone and initiating pairing:

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

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.



Melody 6.0.39

2. 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:

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

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

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

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

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



Melody 6.0.39

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:

Melody SPP Device One	Melody SPP Device Two
<pre>//Sends a string to device two. >SEND 15 Hi, I'm device one. >OK</pre>	<pre>//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.</pre>

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

Melody SPP Device One	Melody SPP Device Two
<pre>// Use the SEND command to send 20 bytes of data through the SPP connection. >SEND_RAW 15 20 >OK // The next 20 bytes we send through the UART interface, will be sent through the SPP connection to device two. >012345678901234567890 // Device one has already received 20 characters, so send to the data to device two.</pre>	<pre>// Do nothing in device two. // Do nothing in device two. // Device two receives the 20 bytes. >RECV 15 20 01234567890123456789</pre>

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.**

Melody SPP Device One	Melody SPP Device Two
<pre>// Execute the status command in device two and obtain the results. >SEND_RAW 15 7 >OK // Send the status command, <CR> is the carriage return character.</pre>	<pre>// Do nothing in device two. //Device two, which is in command mode, receives the status command and</pre>



Melody 6.0.39

<pre>>STATUS<CR> // 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</pre>	<pre>executes it, then it sends the results through the SPP connection. // Do nothing in device two.</pre>
--	---

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.

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

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.

Melody SPP Device One	Melody SPP Device Two
<pre>// 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.</pre>	<pre>// 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</pre>



Melody 6.0.39

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

Melody SPP Device One	Melody SPP Device Two
<pre>// Follow the procedure to exit from data mode using the escape sequence. >\$\$\$\$ >OK // Device two is no longer in data mode.</pre>	<pre>// Do nothing in device two. // Do nothing in device two.</pre>

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

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



Melody 6.0.39

3. 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.

1. Incoming call

AG	HF
<pre>>CALL 12 INCOMING {NUMBER} >OK >CALL_INCOMING AGHFP 12 >INBAND RING AGHFP¹⁶</pre>	<pre>>CALL_INCOMING HFP 13</pre>

2. 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.

AG	HF
<pre>>CALL DIAL AGHFP 12 {NUMBER}</pre>	<pre>>CALL 13 OUTGOING {NUMBER} >OK</pre>
<pre>>CALL REDIAL AGHFP 12</pre>	<pre>>CALL 13 REDIAL >OK</pre>
<pre>>CALL MEMORY AGHFP 12 {MEMORY STRING}</pre>	<pre>>CALL 13 MEMORY {MEMORY STRING} >OK</pre>
<pre>>CALL 12 OUTGOING {NUMBER} >OK >CALL OUTGOING AGHFP 12</pre>	<pre>>CALL OUTGOING HFP 13</pre>

3. Call answer/reject

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

AG	HF
<pre>>CALL ACTIVE AGHFP 12</pre>	<pre>>CALL 13 ANSWER >OK >CALL ACTIVE HFP 13</pre>
<pre>>CALL END AGHFP 12</pre>	<pre>>CALL 13 REJECT >OK >CALL IDLE HFP 13</pre>
<pre>>CALL 12 ANSWER</pre>	

¹⁶ If in-band ringing is enabled in HFP_CONFIG



Melody 6.0.39

>OK >CALL ACTIVE AGHFP 12	>CALL ACTIVE HFP 13
>CALL 12 REJECT >OK >CALL END AGHFP 12	>CALL IDLE HFP 13

4. 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.

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

5. 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.

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

6. Three way calling

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

AG	HF
>CALL_INCOMING AGHFP 12	>CALL_INCOMING HFP 13 // accept incoming call and put other call on hold >CALL 12 TWC 1 >OK



Melody 6.0.39

```
// swap calls  
>CALL 12 TWC 1  
>OK
```

```
// merge calls  
>CALL 12 TWC 3  
>OK
```



Melody 6.0.39

4. PBAP

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

// 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 10 numbers dialled
>PB_PULL 16 0 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
```




Melody 6.0.39

5. MAP

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

// 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
```



Melody 6.0.39

6. Music source (A2DP & AVRCP)

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=X X 0 1 1 X X X X X X X
>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 CONNECTABLE DISCOVERABLE OFF
>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
>OK17
```

¹⁷ If the remote device is Melody, don't forget to enable meta data to receive the track information (MUSIC_META_DATA).



Melody 6.0.39

7. HID

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

```
// Enable HID host profile
>SET PROFILES=X X X X X X X X 0 1 X X
>OK
>WRITE
>OK
>RESET

// Read remote device descriptor
>HID_READ {BD ADDRESS}
>PENDING
>HID_READ 0x014F
>{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}{9
1}{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}{0
1}{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}{0
2}{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}{0
8}{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}

// Initiate a connect to a HID device
>OPEN {BD ADDRESS} HID
>PENDING
>OPEN_OK 17 HID

// 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}
```



Melody 6.0.39

```
// 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=X X X X X X X X 1 0 X X
>SET COD=000540
>OK
>WRITE
>OK
>RESET

// Initiate a connect to a HID host
>OPEN {BD ADDRESS} HID
>PENDING
>OPEN_OK 17 HID

// Press 'a'
>SEND_RAW 17 8
>OK
>{00}{00}{04}{00}{00}{00}{00}{00}
// Release 'a'
>SEND_RAW 17 8
>OK
>{00}{00}{00}{00}{00}{00}{00}{00}
```

```
// Enable HID device profile
>SET PROFILES=X X X X X X X X 1 0 X X
>SET COD=000580
>OK
>WRITE
>OK
>RESET

// Update HID descriptor to be a mouse
// example descriptor:
05010902A1010901A10005091901290815002501950875018102950081030501093009311601F826F
F07750C9502810609381581257F750895018106050C0A380295018106C0C0
```



Melody 6.0.39

```
>HID_DESC 71
>{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
>OK
>{00}{00}{30}{00}{00}{00}{00}{00}
// Press left button
>SEND_RAW 17 6
>OK
>{01}{00}{00}{00}{00}{00}{00}{00}
// Release left button
>SEND_RAW 17 6
>OK
>{00}{00}{00}{00}{00}{00}{00}{00}
```



Melody 6.0.39

8. 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=X X 3 0 3 X X X X X X X
>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 CONNECTABLE DISCOVERABLE OFF
>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
```



9. 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.

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



Melody 6.0.39

10. BLE (BC Smart Service)

Melody (Central mode)	Melody(Peripheral mode)
<pre>// configure device in Central mode >SET BLE_CONFIG=1 >OK >WRITE >OK >RESET >BlueCreation Copyright 2015 >Melody Audio V6.0.8 >Build: 1442512334 >Ready // scan for peripheral devices >SCAN 5 >SCAN {BT ADDRESS}1 <BC00187> 12 -84db >OK // central connects to peripheral >OPEN {BT ADDRESS} BLE 1¹⁸ >PENDING >OPEN_OK 14 BLE</pre>	<pre>// configure device in Peripheral mode >SET BLE_CONFIG=0 >OK >WRITE >OK >RESET >BlueCreation Copyright 2015 >Melody Audio V6.0.8 >Build: 1442512334 >Ready // start advertising >ADVERTISING ON >OK // accept connection >OPEN_OK 14 BLE</pre>
<pre>// write message >SEND 14 Hello¹⁹ >OK</pre>	<pre>// receive message >RECV 14 Hello</pre>
<pre>// enable notification (BC_SMART Data characteristic) >BC_SMART_NOTIF 14 ON OFF >OK // receive notification >RECV 14 Hi!!</pre>	<pre>// send notification >SEND 14 Hi!! >OK</pre>
<pre>// send command >BC_SMART_COMMAND 14 GET AUDIO >OK >RECV 14 AUDIO=0 >RECV 14 OK</pre>	<pre>// handle command // response over BLE (if notif enabled)</pre>

¹⁸ By default Melody uses random address as we can see in the scan result (type = 1)

¹⁹ To send data over BLE, it is possible to enter in Data mode.



Melody 6.0.39

11. BLE central (Generic commands)

In this example, Melody is connected to another board with Melody configured in Peripheral mode (link 14).

Melody (Central mode)

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

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

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

// Read characteristic - Manufacturer Name (Device Information service)
>BLE_READ 14 000C
>BLE 14 [000C] 12 42 6C 75 65 43 72 65 61 74 69 6F 6E //BlueCreation
>OK

// Listen notifications - BC Smart Data (BC Smart service)20
>BLE_NOTIF 14 0028 ON
>OK

// Write characteristic - BC Smart Data (BC Smart service)20
>BLE_WRITE 14 0028 5 48 45 4C 4C 4F //HELLO
>OK
```

²⁰ Note that with BC Smart Service, it is simpler to use the SEND or the BC_SMART commands



Melody 6.0.39

12. 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: -12dBm (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)21
    (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
```

²¹ BlueCreation URL encoded with Google URL shortener at <https://goo.gl/IZ0OQA>



Melody 6.0.39

13. 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
```



Melody 6.0.39

14. 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 2 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
```



Appendix D: Error codes

Error code	Description
0x0003	Unknown error
Command Errors	
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
Warnings	
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
Critical errors	
0xF00X	Critical error
0xF004	Wrong config



Terms and definitions

A2DP	Advanced Audio Distribution Profile
AG	Audio Gateway
AVRCP	Audio/Video Remote Control Profile
cVc	Clear Voice Capture
BLE	Bluetooth Low Energy
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
PDL	Paired Device List
PBAP	Phone Book Access Profile
SPP	Serial Port Profile
TWS	True Wireless Speaker
WBS	Wide Band Speech



General Notes

- **Text** - Features that have not been tested yet and will be integrated in the next releases.
- **Text** - Features that are meant to be improved in the next releases are in orange.
- BlueCreation's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. BlueCreation will not warrant the use of its devices in such applications.
- While every care has been taken to ensure the accuracy of the contents of this document, BlueCreation cannot accept responsibility for any errors. BlueCreation reserves the right to make modifications, corrections and any other changes to its products at any time. Customers should obtain the latest information before placing orders.
- BlueCreation, other products, services and names used in this document may have been trademarked by their respective owners. The publication of this information does not imply that any license is granted under any patent or other rights owned by BlueCreation.
- Refer to <http://www.bluecreation.com/> for more information. BlueCreation® is a trading name for Cambridge Executive Limited.



Melody 6.0.39

Change Log

- 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 (see [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



Melody 6.0.39

- 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 disable commands 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
- Melody 6.0.29
 - Add support for Melody Audio Proximity
 - 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 6.0.39

- 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 6.0.39

- 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)