JBL Portable Protocol Specification

Version 2.39

Apr 8, 2024

Table of Revisions

Revision	Description	Date	Author	Reviewer
0.1	Init version.	2014-08-30	Eason Wang	Software team
0.2	Add ReqRoleCheck command.	2016-07-09	Eason Wang	Software team
0.3	Add color and MID details.	2016-10-20	Eason Wang	Software team
0.4	Add PID, MID, and Pusle3 LED lighting control.	2017-01-17	Eason Wang	Software team
0.5	Add Pulse3 brightness control.	2017-03-02	Eason Wang	Software team
0.6	Update Flip4 MID.	2017-03-14	Eason Wang	Software team
0.7	Add EQ, apps tone, and HFP switch.	2017-04-18	Eason Wang	Software team
0.8	Add adopt device list description for Charge3.	2017-06-01	Eason Wang	Software team
0.9	Add BoomBox MID, and update SetHFPMode command.	2017-05-06	Eason Wang	Software team
1.0	Update for Boombox, Filps4 and Pulse3.	2017-06-06	Eason Wang	Software team
1.1	Update Connectable, Non-Connectable bits.	2017-06-08	Eason Wang	Software team
1.2	Add Xtreme2 MID.	2017-12-13	Vito Gai	Software team
1.3	Add new color MID 0x08 for Boombox.	2018-01-05	Vito Gai	Software team
1.4	Add AppACK for RetDevInfo command.	2018-02-02	Ken Liu	Software team
1.5	Update bass bar range to 1~21.	2018-06-21	Ken Liu	Software team
1.6	Update color naming and MID for Charge4.	2018-08-01	Ken Liu	Software team
1.7	Add Flip5, Pulse4 support.	2019-08-10	Ken Liu	Software team
1.8	Add cancel OTA command.	2020-03-01	Ken Liu	Software team
1.9	Add enter Partyboost command, channel info in broadcast, and lock command.	2020-03-17	Ken Liu	Software team
2.0	Add Boombox2, Xtreme3, and Xtreme2 GM support.	2020-05-25	Ken Liu	Software team
2.1	Add Charge5 support.	2020-07-07	Ken Liu	Software team
2.2	Add new color for Charge35 new color white, and GATT indicator in broadcast.	2020-06-11	Ken Liu	Software team
2.3	Add command for GATT support.	2020-12-11	Ken Liu	Software team
2.4	Add general EQ settings.	2021-01-06	Gavin Gan	Software team
2.5	Add new color 0x12 for Charge 5.	2021-03-08	Gavin Gan	Software team

2.6	Add CRC16 of BT address in BLE adverting Add CRC16 of BT address in Device Info	2021-04-20	Gavin Gan	Software team
2.7	Added Analytics Protocol Add Packets for Sharing Analytics with JBL App Add Payload structure for Analytics Data Payload (Merged from v1.3.1 of old spec)	2021-04-24	Gavin Gan	Software team
2.8	Add command to detect water in charging port	2021-04-27	Gavin Gan	Software team
2.9	Add new color ID for Flip 6 Remove BR/EDR flag in BLE adverting Add customized SDP UUID	2021-05-12	Gavin Gan	Software team
2.10	Add command to detect whether there was water in charging port and whether charging port was overheating	2021-06-22	Gavin Gan	Software team
2.11	Update reqLinkModeCmdId to setLinkMode	2021-07-07	Gavin Gan	Software team
2.12	Add specification for JBL Boombox3	2021-09-28	Gavin Gan	Software team
2.13	Add specification for unsupported command	2021-11-04	Gavin Gan	Software team
2.14	Add specification for LED of Pulse 5	2022-01-17	Gavin Gan	Software team
2.16	Add switch LED package command	2022-02-23	Gavin Gan	Software team
2.17	Update LED pattern table	2022-03-04	Gavin Gan	Software team
2.18	Add snap show command for patterns	2022-03-16	Gavin Gan	Software team
2.19	Remove position of CANVAS pattern	2022-03-29	Gavin Gan	Software team
2.20	Add cocktail patterns	2022-04-07	Gavin Gan	Software team
2.21	Add new analytics command	2022-04-14	Gavin Gan	Software team
2.23	Add PID/MID for pulse 5	2022-05-17	Gavin Gan	Software team
2.24	Add command for serial number	2022-08-18	Gavin Gan	Software team
2.25	Add command for play data analytic and spec of OneTouch music button	2022-12-26	Gavin Gan	Software team
2.26	Add EQ command for NTI command	2023-02-07	Gavin Gan	Software team
2.27	Add color ID definition for Xtreme 4 and Go 4	2023-06-06	Gavin Gan	Software team
2.28	Add new commands for battery health information	2023-06-28	Gavin Gan	Software team
2.29	Removed the Auracast stereo specification	2023-07-14	Gavin Gan	Software team
2.30	Set the preset EQ payload length as a dynamic value	2023-07-20	Gavin Gan	Software team
2.31	Add new color ID for JBL Go 4	2023-08-07	Gavin Gan	Software team
2.32	Add new command for player information	2023-08-16	Gavin Gan	Software team
2.33	Add new color ID for JBL Clip 5	2023-08-23	Gavin Gan	Software team
2.34	Add flag for Auracast in return device info	2023-10-27	Gavin Gan	Software team
2.35	Update parameter for analytics command	2023-12-08	Gavin Gan	Software team
2.36	Add analytics for lost package of LC3 audio	2023-12-27	Gavin Gan	Software team
2.37	Request vendor information for Flip 6	2024-02-01	Gavin Gan	Software team
2.38	Update vendor information commands for Flip 6	2024-03-01	Gavin Gan	Software team
2.39	Add PID/Color ID for JBL Flip 7 and JBL Charge 6	2024-04-08	Gavin Gan	Software team

CONTENTS

1. Scope

This document describes the App-device interface between the JBL link App and JBL portable device.

2. Adopt Device List

The protocol is support by JBL portable speakers listed as below.

Product Name	PID	MID	Description	Role	Description
JBL Flip4	0x1ED1	0x01	Black	0x00	Device is in Normal Mode
		0x02	Red	0x01	Device is in Broadcaster Mode
		0x03	Blue	0x02	Device is in Receiver Mode
		0x04	Teal	0x03 – 0x7F	Reserved
		0x05	Gray		
		0x06	White		
		0x07	Malta		
		0x08	Squad		
		0x09	Zap		
		0x0A	Trio		
		0x0B	Mosaic		
		0x10	Trio		
		0x11	Mosaic		
JBL Charge3	0x1EBC	0x01	Black		
		0x02	Red		
		0x03	Teal		
		0x04	Blue		
		0x05	Gray		
		0x06	Mosaic		
		0x07	Squad		
		0x08	Zap		
		0x09	Malta		
JBL Pulse3	0x1ED2	0x01	Black		
		0x02	White		
JBL BoomBox	0x1EE7	0x01	Black		
		0x02	Forest Green		
		0x08	Squad		
Onyx Studio4	0x1EE5	0x01	Black		
		0x02	White		
		0x03	Deep sea blue		
		0x04	Metal gray		
JBL Xtreme2	0x1EFC	0x01	Black (Used)		
		0x02	Red		
		0x03	Blue (Used)		
		0x04	Teal (Used)		
		0x05	Gray		
		0x06	White		
		0x07	Malta		
		0x08	Squad		
		0x09	Zap		
		0x0A	Trio		
		0x0B	Mosaic		
		0x10	Trio		
		0x11	Mosaic		

JBL Xtreme2 GM	0x2038	0x12	Gun metal		
JBL Charge4	0x1F17	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x05	Forest Green		
		0x06	Mustard Yellow		
		0x07	Grey		
		0x08	Sand		
		0x09	Teal		
		0x10	Pink		
		0x11	Squad		
		0x12	Magenta		
		0x13	Black Camouflage		
JBL Flip5	0x1F31	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x05	Forest Green		
		0x06	Mustard Yellow		
		0x07	Grey		
		0x08	Sand		
		0x09	Teal		
		0x10	Pink		
		0x11	Black star		
		0x12	Tomorrowland		
		0x13	Squad		
		0x14	ECOBLU		
		0x15	ECOGRN		
		0x16	Black Camouflage		
JBL Pulse4	0x1F56	0x01	Black		
		0x02	Red		
JBL BoomBox2	0x1F53	0x01	Black		
		0x02	Red		
JBL Xtreme3	0x202F	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
JBL Charge5	0x2040	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x05	Forest Green		
		0x07	Grey		
		0x09	Teal		
		0x10	Pink		
		0x12	Tomorrowland		
		0x13	Black Camouflage		
JB Flip 6	0x204F	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x05	Forest Green		
		0x07	Grey		
		0x09	Teal		
		0x10	Pink		
		0x12	Tomorrowland		
		0x13	Camouflage		
		0x14	Martin Garrix		
JBL Boombox3	0x206D	0x01	Black		
		0x13	Camouflage		
JBL Pulse 5	0x2050	0x01	Black		
JBL Xtreme 4	0x20DC	0x01	Black		
		0x03	Blue		
		0x13	Camouflage		
JBL GO 4	0x20E4	0x01	Black		

		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x10	Pink		
		0x13	Camouflage		
		0x14	Purple		
		0x15	Funky Black		
		0x16	Wimbledon Green		
JBL Clip 5	0x20F5	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x10	Pink		
		0x13	Camouflage		
		0x14	Purple		
		0x15	Funky		
		0x16	Wimbledon Green		
JBL Flip 7	0x2107	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x08	Sand		
		0x10	Pink		
		0x13	Camouflage		
		0x14	Purple		
		0x15	Funky Black		
JBL Charge 6	0x20e3	0x01	Black		
		0x02	Red		
		0x03	Blue		
		0x04	White		
		0x08	Sand		
		0x10	Pink		
		0x13	Camouflage		
		0x14	Purple		
		0x15	Funky Black		

Table 2-1 JBL Portable Product ID, Model ID, Role Definitions

Model IDs were defined per product and indicate different HW versions and appearances.

3. Transport

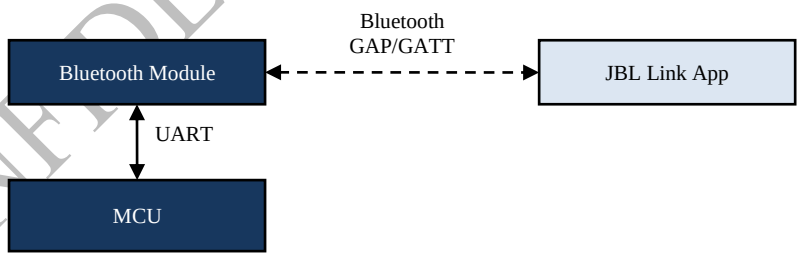


Figure 3-1 App-Dev Transport

App will discovery device Bluetooth GAP profile, see **Section 3.1** for details. There will be 3 different data transports to run this protocol:

- GATT (BLE)

The JBL portable speakers adopt this documents shall supports BLE services list in **Table 3-1**.

Primary Service	Service UUID
GAP	0x1800
GATT	0x1801
BLE_RX_TX	0x6578 6365 6c70 6f69 6e74 2e63 6f6d 0000

Table 3-1 Bluetooth LE Services

GAP service is used for device discovery. `BLE_RX_TX` service is used for transfer data packets between App and device.

3.1 Device Discovery

App shall try to discovery device via GAP profile.

When App discovered a peripheral with the GAP profile, it can read device characteristics in **Table 3-2**.

Data Type	Description
Manufacturer Specific Data	A unique ID to identify device and its role.
Broadcaster/Receiver Data	**4-6bytes of data is only for the devices not intended for APP.** Name type can be changed by device depending on this data

Table 3-2 GAP characteristics

Data Type	Size	Description
VID	2 bytes	Harman Vendor ID, VID of old products is 0x00 0x57, new VID should be 0x0E 0xCB
PID	2 bytes	Product ID, Refer to **Table 2-1**.
MID	1 byte	Model ID, Refer to **Table 2-1**.
Role and Connectable /Non-connectable bit	1 byte	Bit 7 = Connectable status (0 = Connectable, 1 = Non-connectable) Bit 5-6 = Stereo channel mode (0 = old firmware,1 = Party,2 = ChannelLeft,3 = ChannelRigh Bit 2-4 = Battery (BATT_CRITICAL: 0, BATT_LOW: 001, BATT_LEVEL0: 010, BATT_LEVEL1: 011, BATT_LEVEL2: 100, BATT_LEVEL3: 101, BATT_LEVEL4: 110) Bit 0-1 = Role (0 = normal, 1 = primary, 2 = receive), E.g. This field should be changed to 1 or 2 if users turn on Partyboost or Auracast. 0 means Partyboost or Auracast is off. Primary role means DUT is streaming music to other DUTs. Other details, Refer **Table 2-1**.
SrcName	2 bytes	A CRC16 value of source device name.

CRC16 of BT device address	2 bytes	CRC16 of BT device address (6 bytes)
Extra info of Auracast, Low Energy Mode and Spotify Button Actions	1 byte	Bit 0 = Support Auracast (1 = Support Auracast, 0 = Not support Auracast) Bit 1 = Auracast Party status (1 = Auracast Party on, 0 = Auracast Party off), default value is 0. Bit 2 = Spotify Quick Access button is triggered (1 = triggered), default value is 0. Once users triggered quick access button, device should broadcast status 5 seconds. Bit 3 ~ 7, reserved.
Auracast Stereo Group ID (Reserved)	1 byte	If the user created an Auracast stereo group for 2 devices. The App should generate a group ID (4 bytes) for the grouped devices. And the Auracast stereo grouped devices should broadcast the 1st byte of group ID in BLE ADV data. 0x00: device does not work in Auracast stereo mode.

Table 3-3 Manufacturer Data Format

The App shall firstly check if the `SrcName` (CRC16 value) is same as it-self. Only if they are the same, App can connect target device. The same CRC16 algorithm in Linux Kernel is used for calculating this field. The poly is 0x8005 (x^16 + x^15 + x^2 + 1). Refer to:

https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/include/linux/crc16.h

https://git.kernel.org/cgit/linux/kernel/git/stable/linux-stable.git/tree/lib/crc16.c

3.2 Data Transfer

App and device use `BLE_RX_TX` service to talk with each other by private protocols defined in follow sections. There are 2 characteristics in the service that used for sending and receiving data packet.

Characteristic	UUID	Access	Size
RX_CHAR	0x6578 6365 6c70 6f69 6e74 2e63 6f6d 0001	Read/Notify	Max 60 bytes
TX_CHAR	0x6578 6365 6c70 6f69 6e74 2e63 6f6d 0002	Write	Max 60 bytes

Table 3-4 GATT characteristics for packet transfer

Device will send notification to App with packet bytes in `RX_CHAR` characteristic. One notification always carries only one data packet. App shall firstly register the notification after connection established.

App can also write packets to `TX_CHAR` characteristic. A packet shall write to the characteristic by one write operation.

4. Packet Format

The general packet format defines in **Table 4-1**.

Section	Field	Size	Description
Header	Identifier	1 byte	Always 0xAA.
	CmdID	4 bits (MSB)	Command ID.
	SubCmdID	4 bits (LSB)	Sub command ID.
	PayloadLen	1 byte / 2 bytes	Payload length (0 ~ 0x3C), Big Endian
Payload	Actually data packet.		

Table 4-1 Package format

The field `PayloadLen` defines packet length without headers. So the total package length equals `PayloadLen` + 4.

4.1 Command Summary

Category	CmdID	Name	Sub-CmdID	R/T	ACK	Description
General	0x0	DevACK	0x0	RX	No	Some commands need ACK.
		AppACK	0x1	TX	No	Some commands need App ACK.
		DevByeBye	0x2	RX	AppACK	Device want disconnect.
		AppByeBye	0x3	TX	DevACK	App want disconnect.
Device Info	0x1	ReqDevInfo	0x1	TX	RetDevInfo	App request device information.
		RetDevInfo	0x2	RX	AppACK	Device info notify to App
		ReqDevInfoToken	0x3	TX	RetDevInfo / DevACK	App request device to update all its info.
		SetDevInfo	0x5	TX	DevACK	App change device properties.
		RetRoleInfo	0x6	RX	No	Provides the Information about Device Mode (0-Norrmal, 1-Broadcaster 2-Receiver)
		ReqRoleCheck	0x07	TX	RetRoleInfo	App requests device role info.
Remote Control	0x3	IdentDev	0x01	TX	No	App request to trigger some sound/LED light indication in device.
		SetMFBStatus	0x02	TX	DevACK	App change MFB status.
		ReqMFBStatus	0x03	TX	RetMFBStatus/ DevACK	App request MFB status.

		RetMFBStatus	0x04	RX	No	Device return MFB status.
OTA DFU	0x4	ReqVer	0x1	TX	RetVer	App request device version
		RetVer	0x2	RX	No	Device report version to App
		ReqDfuStart	0x3	TX	DevACK	App request to start firmware upgrade.
		NotifyDfuStatusChange	0x5	RX	No	Response data from upgrade engine.
		SetDfuData	0x4	TX	DevACK	App sends DFU data to device.
		NotifySecStart	0x6	TX	DevACK	App indicate a new section started.
		NotifyDfuCancel	0x7	TX	DevACK	App notifies device DFU is cancelled.
		ReqDfuResume	0x8	TX	RetDfuResume	App request to resume DFU.
		RetDfuResume	0x9	RX	No	Device return if the DFU can be start.
LED Pattern Control	0x5	ReqLedPatternInfo	0x1	TX	RetLedPatternInfo	App request device led pattern information.
		RetLedPatternInfo	0x2	RX	No	Device report led pattern settings to app.
		SetLedPattern	0x3	TX	DevACK	App change device led pattern information.
		NotifyInsLelChange	0x4	RX	No	Device report Intensity level change to app.
Speaker Settings	0x6	ReqEQMode	0x01	TX	RetEQMode	App request device EQ mode.
		RetEQMode	0x02	RX	No	Device returns EQ mode.
		SetEQMode	0x03	TX	DevACK	App request to change EQ mode. Below data will be send in payload to identify which mode to set. 0: Indoor 1: Outdoor.
		NotifyEQChangeMode	0x04	RX	No	Device sent the notification that EQ mode is changed on device. Below data in the received packet will identify the EQ mode. 0: Indoor. 1: Outdoor.

		ReqFeedbackToneStatus	0x05	TX	RetToneStatus	App request speaker feedback tone status.
		RetFeedbackToneStatus	0x06	RX	No	Device returns Tone status.
		SetFeedbackTone	0x07	TX	DevACK	App request to set the tone status. Below data is sent in payload to identify to set tone on or off 0:off 1: on.
		ReqHFPStatus	0x08	TX	RetHFPStatus	App request HFP status.
		RetHFPStatus (0x69)	0x09	RX	No	Device returns HFP Status.
		SetHFPMode (0x70)	0x10	TX	DevACK	App request to set the HFP Status. Below data will be send in payload to identify which mode to set. 0: Off 1: On.
		ReqLinkModeCmdId	0xa	TX	RetLinkModeCmdId	App set link mode
		RetLinkModeCmdId	0xb	RX	No	Device returns link mode status
		ReqEQ	0xC	TX	RetEQ	Return RetEQ
		RetEQ	0xD	RX	No	
		SetCurrentEQ	0xE	TX	RetEQ	Return RetEQ
Open API	0x7	ReqLightStatus	0x01	TX	RetLightStatus	App request device light status.
		RetLightStatus	0x02	RX	No	Device returns light status.
		SetLightStatus	0x03	TX	DevACK	App request to set the light status. Below data is sent in payload to identify to set light on or off 0:off 1: on.
		ReqImageStart	0x04	TX	DevACK	App request to start sending image.
		SetImageData	0x05	TX	DevACK	App sends image data to device.
		SetBassVolume	0x06	TX	DevACK	App requrest to set bass volume
		ReqBassVolume	0x07	TX	RetBassVolume	App requrest bass volume
		RetBassVolume	0x08	RX	No	Device returns Bass volume.

Table 4-2 Command summary

As the Air traffic is very busy when audio playing and App communication at same time, it is necessary to reduce data packets. Developers shall try to pack as many requests as possible to one packet, especially the `ReqDevInfo` and `RetDevInfo` commands. See command details for reference.

4.2 ACK

4.2.1 DevACK

Some commands need a special `DevACK` from device, shows in **Table 4-3**.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x00, PayloadLen: 0x02)			
Payload	ACKCmdID	1 byte	The command ID need to acknowledge.
	StatusCode	1 byte	Response status.

Table 4-3 DevACK packet format

Usually a `StatusCode` 0 indicates success and other non 0 codes indicate failure or error occurred.

4.2.2 AppACK

App also need `AppACK` to device for some commands.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x01, PayloadLen: 0x02)			
Payload	ACKCmdID	1 byte	The command ID need to acknowledge.
	StatusCode	1 byte	Response status.

Table 4-4 DevACK packet format

Usually a `StatusCode` 0 indicates success and other none 0 codes indicate failure or error occur.

4.2.3 Response for unsupported command

Device should reply App as follow format once device received unsupported command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: EE, PayloadLen: 0x01)			
Payload	Received command	1 byte	

4.3 Device Info

Category	CmdID	Name	Sub-CmdID	R/T	ACK	Description
Device Info	0x1	ReqDevInfo	0x1	TX	RetDevInfo	App request device information.

		RetDevInfo	0x2	RX	No	Device report information to App.
		ReqDevInfoToken	0x3	TX	RetDevInfo / DevACK	App request device to update all its info.
		SetDevInfo	0x5	TX	DevACK	App change device properties.
		RetRoleInfo	0x6	RX	No	Provides the Information about Device Mode (0- Norrmal, 1- Broadcaster, 2- Receiver)
		ReqRoleCheck	0x7	TX	RetRoleInfo	App requests device role info.

Table 4-5 Device Info commands

4.3.1 ReqDevInfo

App can query device information by `ReqDevInfo` command (**Table 4-6**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x11, PayloadLen: 0x00)			
Payload	NA.		

Table 4-6 ReqDevInfo packet format

4.3.2 RetDevInfo

If device received this command, it will reply a `ReqDevInfo` commands to report all its information to App including how many devices are linked and its information. See **Table 4-7** for details.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x12, PayloadLen: 0xnn)			
Payload	DevIdxToken	1 byte	Device index. Can be ignored if it is 0x00
	Info Token	n bytes	Information token for device 0.
	Info Token	n bytes	Information token for device 0.
	DevIdxToken	1 byte	Device index.
	InfoToken	n bytes	Information token for device 1.
	InfoToken	n bytes	Information token for device 1.
	…		

Table 4-7 RetDevInfo packet format

There may be several tokens in one packet. Each of them presents a kind of device information item. All these tokens are separated by device index token (**Table 4-8)**. All the tokens following are describe the information for this device until another device index token is reached.

Field	Size	Description
TokenType	2 bits (MSB)	Always 0 for device index token.
DevIdx	6 bits (LSB)	Device index in link system.”0” always means the device connected to App.

Table 4-8 Device index token format

The device index shall be ordered from 0. The first `DevIdxToken` can be ignored if it is 0x00 (present device 0).

The information token have 2 kinds of format: fixed length format and dynamic length format, shows in **Table 4-9** and **Table 4-10**.

Field	Size	Description
TokenType	2 bits(MSB)	Always 1 for fixed length information token.
TokenID	6 bits(LSB)	Total information token count.
TokenValue	n bytes	Information value

Table 4-9 Fixed length information token format

Field	Size	Description
TokenType	2 bit(MSB)	Always 3 for dynamic length information token.
TokenID	6 bits(LSB)	Total information token count.
TokenLen	1bytes	Token length (0-0x3C).
TokenValue	TokenLen	Information value.

Table 4-10 Dynamic length information token format

The `TokenID` defines in **Table 4-11**.

Name	ID	R/W	Length	Value
Device name	0xC1	RW	dynamic	UTF-8, max 16 bytes. It is writable only. App can get the device name by BLE name.
Product ID	0x42	RW	2 bytes	Product ID, Refer to Table 2-1. No need for the device 0. Only test/tool App can write this token.
Model ID	0x43	RW	1 byte	Model ID, Refer to Table 2-1. No need for the device 0. Only test/tool App can write this token.
Battery status	0x44	R	1 byte	1 bit (MSB): charging status, 1 means battery charging. 7 bit (LSB), 0-100 present 0% - 100%.
Active channel	0x46	RW	1 byte	0x0: Stereo 0x1: Left 0x2: Right
Audio source	0x47	R	1 byte	0x0: no audio playing 0x1: Bluetooth A2DP 0x2: Aux-in 0x3: Receiver Playing
MAC Address	0x48	R	6 bytes	Bluetooth MAC address.
Stereo Lock	0x49	RW	1 byte	Get or set stereo lock status 0x00 not locked 0x01 locked
CRC16 of BT device address	0x4a	R	2 bytes	CRC16 of BT device address (6 bytes)
Water in charging port	0x4b	R	1 byte	0x00 there is no water in charging port 0x01 there is water in charging port

Support Auracast	0x4c	R	1 byte	0x00: doesn't support Auracast 0x01: support Auracast

Table 4-11 Information token ID

Device may separate the tokens to 2 or more packet if the overall data length is larger than maximum packet length.

When firstly App connected to device, the device shall report all current information to App actively without App query. If other devices are linked, the device shall also report their information including device name, product ID, Model ID…

Device will also send `RetDevInfo` packets to App actively to notify information changes.

4.3.3 ReqDevInfoToken

App can query a specified device token by `ReqDevInfoToken` command. In the command, App assigned the device index (**Table 4-8**) and the target token ID (**Table 4-11**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x13, PayloadLen: 0x02)			
Payload	DevIdx	1 byte	Total information token count.
	TokenID	1 byte	Target token ID.

Table 4-12 ReqDevInfoToken packet format

Device will reply a `RetDevInfo` packet to App with single token or a `DevACK` with error code if any errors occur. E.g. target information is not available in device.

4.3.4 SetDevInfo

App can change some of the device information that is writable with `SetDevInfo` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x15, PayloadLen: 0xnn)			
Payload	DevIdxToken	1 bytes	Device index.
	Info Token	n bytes	Information to be set for device 0.
	Info Token	n bytes	Information to be set for device 0.
	DevIdxToken	1 bytes	Device index.
	InfoToken	n bytes	Information to be set for device 1.
	InfoToken	n bytes	Information to be set for device 1.
	…		

Table 4-13 SetDevInfo packet format

Device shall replay `DevAck` to indicate if success or not.

4.4 Remote Control

App has to identify the device if more than 2 speakers linked. When `IdentDev` (**Table 4-14**) command is received, the target device shall have some LED and Sound indication to let user know who they are looking for.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x31, PayloadLen: 0x01)			
Payload	DevIndex	1 byte	Device index to be identified.

Table 4-14 IdentifyDev packet format

4.4.1 SetMFBStatus

App can change the "Multi-Function Button" to different behavior. When device received `SetMFBStatus:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x32, PayloadLen: 0x01)			
Payload	MFBStatus	1 byte	0x00: MFB act as play/pause. 0x01: MFB act as voice trigger.

4.4.2 ReqMFBStatus

Device will set Multi Function Button accordingly. App need to query status of Multi Function Button once connected with Device by `ReqMFBStatus:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x33, PayloadLen: 0x00)			
Payload	NA		

4.4.3 RetMFBStatus

And device will return Multi Function Button status to App by `RetMFBStatus:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x34, PayloadLen: 0x01)			
Payload	MFBStatus	1 byte	0x00: MFB act as play/pause. 0x01: MFB act as voice trigger.

4.5 OTA DFU

To confirm current SW version, the app shall query device version before upgrade by `ReqVer` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x41, PayloadLen: 0x00)			
Payload	NA		

Table 4-15 ReqVer packet format

4.5.1 RepVer

Device will reply a `RepVer` command to App.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x42, PayloadLen: 0x01)			
Payload	SwVersion	3 bytes	SW version. E.g. 030201 means v3.2.1
	HwVersion	1 byte	HW version.

Table 4-16 RepVer packet format

App shall use the version to check with Internet server if upgrade is needed.

4.5.2 ReqDfuStart

When user trigger upgrade in App, App shall send a `ReqDfuStart` command to device.

For CSR or QCC product:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x43, PayloadLen: 0xnn)			
Payload	DfuCrc	4 bytes	CRC32 value of the section.
	DfuSecIdx	1 byte	DFU section index.
	DfuSize	3 bytes	DFU data size in bytes. Big Endian
	DfuCrc	4 bytes	CRC32 value of next section.
	DfuSecIdx	1 byte	DFU section index.
	DfuSize	3 bytes	DFU data size in bytes. Big Endian
	…		

For Vimicro product:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x43, PayloadLen: 0xD)			
Payload	DfuCrc	4 bytes	CRC32 value of the section.
	DfuSecIdx	1 byte	DFU section index (default 0x00).
	DfuSize	3 bytes	DFU data size in bytes. Big Endian.
	DfuVersion	3 bytes	DFU version, E.g. 030201 means v3.2.1.
	BPType	1 byte	OTA Breakpoint resume type: 0x01: normal case (need breakpoint). 0x02: reset breakpoint. 0x11: silent normal case (need breakpoint). 0x12: silent reset breakpoint.

Table 4-17 ReqDfuStart packet format

The command can send the information of several DFU sections. Each section has a 4 bytes CRC value for data verification, 1-byte section index and 3 bytes section data size.

The section index is not needed to start from 0. Only the necessary upgrade section shall be listed in the command.

Device will report `DevAck` to App that it already received the request and start for DFU preparing. During the preparation, other link such as A2DP, HFP, AVRCP, etc. will be shut down. Only the communication link with App (GATT) is open.

4.5.3 NotifyDfuStatusChange

App shall not start sending data until a `NotifyDfuStatusChange` command is received with DFU engine status is `Ready (0x1)`.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x45, PayloadLen: 0x01)			
Payload	Status	1 byte	DFU engine status: 0x01: DFU ready. 0x02: DFU downloading. 0x03: DFU uploading. 0x04: DFU cancel. 0x00: DFU error.

Table 4-18 NotifyDfuStatusChange packet format

The figure below shows how DFU start.

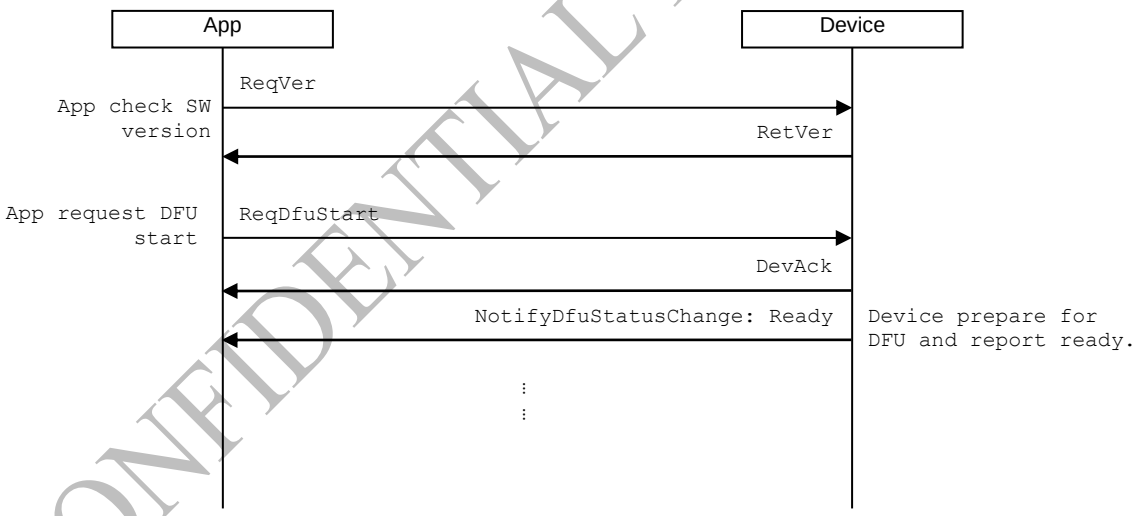


Figure 4-1 DFU Start

4.5.4 SetDfuData

After device ready for downloading, App shall send the DFU data to device via `SetDfuData` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x44, PayloadLen: 0xnn)			

Payload	DfuData	n bytes	DFU content, limit to 56 bytes.

Table 4-19 SetDfuData packet format

For Vimicro product (OTA over BLE connection):

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x44, PayloadLen: 0xnn, 1byte)			
Payload	Sequence number	1 byte	Sequence number to indicate packet index
	DfuData	n byte	DFU content, limit to MTU size

For Vimicro product (OTA over BR/EDR connection):

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x44, PayloadLen: 0xnn, 2bytes Big Endian)			
Payload	Sequence number	1 byte	Sequence number to indicate packet index
	DfuData	n byte	DFU content, limit to MTU size

4.5.5 NotifySectStart

If there are more than 1 sections, before each section, App shall send the `NotifySectStart` command to mark a new section of the follow data. The command of first section can be ignored if the first section index is 0.

For CSR or QCC product:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x45, PayloadLen: 0xnn)			
Payload	DfuCrc	4 bytes	CRC32 value of next section.
	DfuSecIdx	1 byte	DFU section index.
	DfuSize	3 bytes	DFU data size in bytes.

For Vimicro product:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x45, PayloadLen: 0x06)			
Payload	OTA Status	1 byte	DFU engine status: 0x01: DFU ready. 0x02: DFU downloading. 0x03: DFU uploading. 0x00: DFU error.
	Offset	4 byte	The value on dfu engine status is ready
	Offset status	1 byte	Offset status： 0x01: empty 0x02: refresh 0x03: fresh resume

Table 4-20 SetDfuData packet format

After all data received, device shall verify the DFU content. If no error found, device will send a `NotifyDfuStatusChange` command with `Upgrading` status to indicate upgrading started. During upgrading, the device cannot communicate with App anymore.

4.5.6 DFU Data Transfer

Figure 4-2 shows the DFU data transfer flow. The section index can be defined per product.

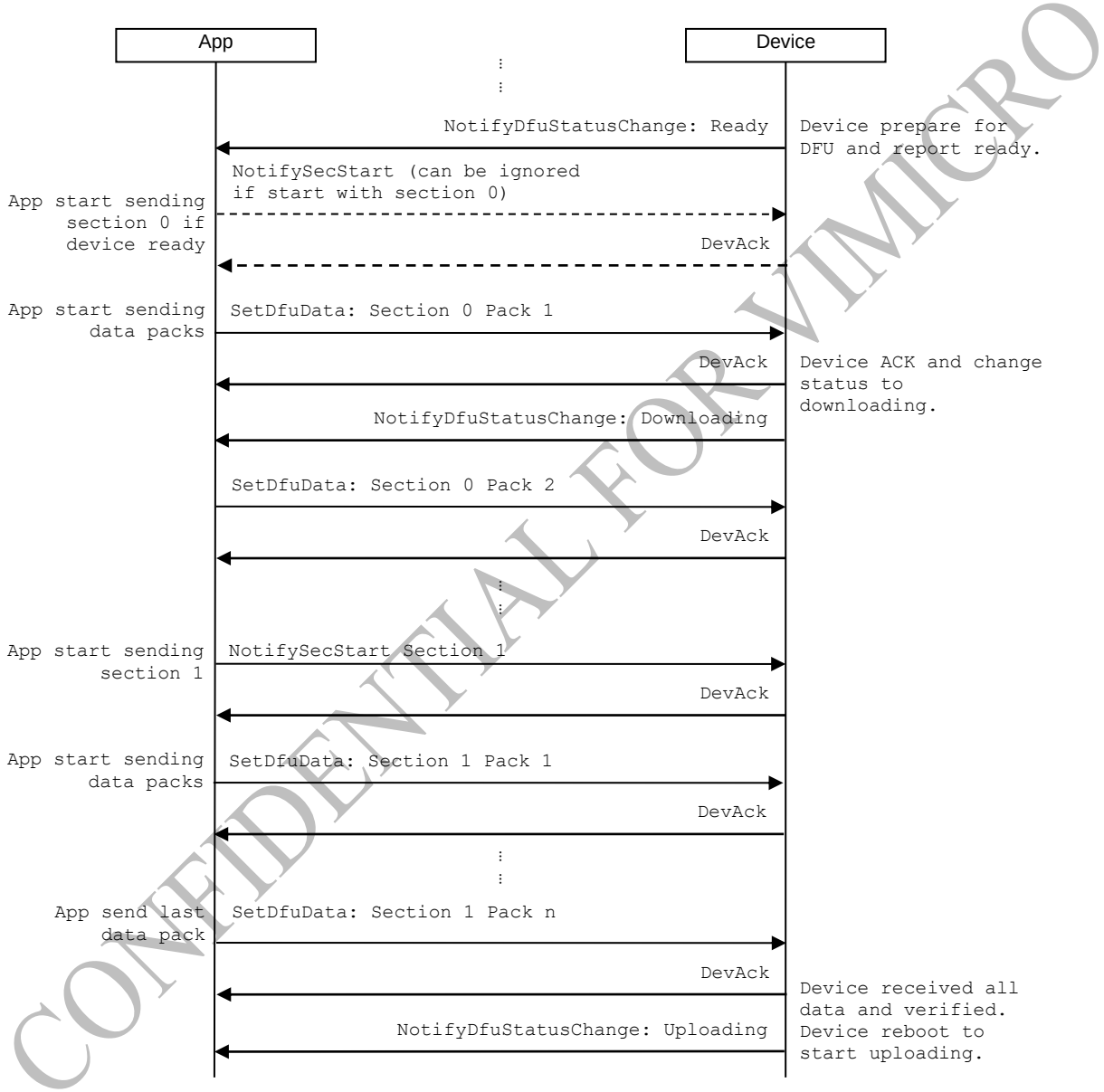


Figure 4-2 DFU Data Transfer

4.5.7 NotifyDfuCancel

User can cancel the upgrade process in the App, while app shall send a `NotifyDfuCancel` command to device.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x47, PayloadLen: 0x1)			
Payload	MsgCode	1 byte	Cancel reason: 0x0: Unknown 0x1: User cancelled.

Table 4-21 NotifyDfuCancel packet format

Device shall stop the DFU process and reboot to last version after DFU cancelled or any other error occurred such as network drop.

4.5.8 ReqDfuResume

If DFU process stopped unexceptional, device shall wait for DFU resume for 5 minutes. When App resumed the connection, it shall firstly send the resume request command: `ReqDfuResume`.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x48, PayloadLen: 0x0)			
Payload	NA		

Table 4-22 ReqDfuResume packet format

4.5.9 RetDfuResume

Device will reply: `RetDfuResume` to App.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x49, PayloadLen: 0x4)			
Payload	ResumeSec	1 byte	0xFF: Device is not in the DFU mode. Otherwise: the section index to be resumed.
	ResumePoint	3 bytes	0xFFFFFF: Device is not in the DFU mode, DFU cancelled. Otherwise: the bytes device already received.

Table 4-23 RetDfuResume packet format

Added breakpoint resume for Vimicro product:

To confirm current device DFU version, the app shall query device version before upgrade by `ReqdfuVer` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x4A, PayloadLen: 0x0)			
Payload	NA		

Table 4-24 ReqDfuResume packet format

Device will reply: `RetDfuVersion` to App.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x4B, PayloadLen: 0x3)			
Payload	dfuversion	3 bytes	dfu version. E.g. 030201 means v3.2.1

Device send command when device image CRC check is completed.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x4C, PayloadLen: 0x1)			
Payload	Status	1 byte	OTA status: 0x1: Ready (The App can trigger upgrade) 0x2: CRC error 0x3: unknown error

Table 4-26

App trigger upgrade will be sent command to device, the device reset after received the command

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x4D, PayloadLen: 0x0)			
Payload	NA		

4.5.10 DFU Resume Flow

The figure below shows the DFU resume flow.

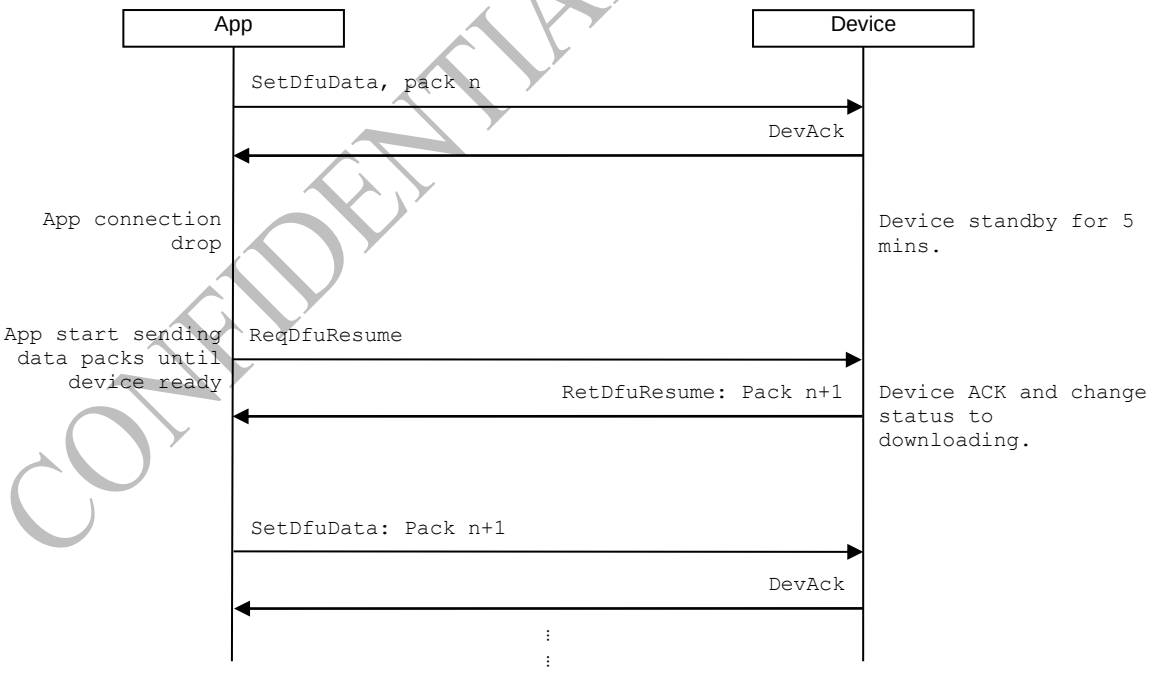


Figure 4-3 DFU Resume Flow

Any time if `NotifyDfuStatusChange` command received with `Error` status from device, App shall stop upgrading and indicate error to end user. The Error may send before `Ready`, in `Downloading` or `Upgrading`.

4.6 Pulse2 LED pattern Control

4.6.1 ReqLedPatternInfo

When the app connected with the JBL pulse 2, app can query device led pattern information by `ReqLedPatternInfo` command (**Table 4-23**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x51, PayloadLen: 0x00)			
Payload	NA		

Table 4-23 ReqLedPatternInfo packet format

4.6.2 RetLedPatternInfo

If device received this command, it would reply a `RetLedPatternInfo` commands to report all its information to App including how show the animation in app and its information.

See table below for details.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x52, PayloadLen: 0x64)			
Payload	PatternID	1 byte	Device LED pattern to be identified, refer to Table 4-25.
	DefaultStatus	99 byte	Default status of LED pattern, refer to **Table 4-25**.

Table 4-24 RetLedPatternInfo packet format

4.6.3 PatternID

The `PatternID` defines in **Table 4-25**.

Name	ID	DefaultPatternStatus
Firework	0x0	Default: Normal 0x0.
Traffic	0x1	No
Star	0x2	No
Wave	0x3	No
Firefly	0x4	No
Rain	0x5	Default: Small 0x0.
Fire	0x6	Default: Level1 0x0.
Canvas	0x7	No
Hourglass	0x8	Default: 1min 0x0.

Ripple	0x9	No

Table 4-25 PatternID definitions

4.6.4 SetLedPattern

App can change device Led Pattern information with `SetLedPattern` command (**Table 4-26**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x53, PayloadLen: 0x64)			
Payload	PatternID	1 byte	LED pattern ID, refer to **Table 4-25**.
	PatternStatus	99 bytes	LED pattern status, refer to **Table 4-27**.

Table 4-26 SetLedPattern packet format.

4.6.5 Led Pattern Status

The Led Pattern Status are defined as below.

Name	ID	R/W	Length	Value
Firework	0x00	W	1 byte	0x0: normal 0x1: Special (Heart)
Traffic	0x01	W	1 byte	a point.(0x1).
Star	0x02	W	1 byte	a point.(0x1).
Wave	0x03	W	1 byte	a point.(0x1).
Firefly	0x04	W	99 bytes	a multiple points (max 0x1 to 0x99).
Rain	0x05	W	1 byte	11 Level. For example, 0x0 to 0x10.
Fire	0x06	W	1 byte	0x0: small 0x1: middle 0x2: high
`Canvas	0x07	W	99 bytes	a multiple points (max 0x1 to 0x99).
Hourglass	0x08	W	1 byte	0x0: 1min 0x1: 5mins 0x2: 10mins 0x3:15mins
Ripple	0x09	W	1 byte	a point (0x1 to 0x99).

Table 4-27 Led pattern status.

4.6.6 NotifyInsLelChange

Device will report intensity level change to App by `NotifyInsLelChange` command (**Table 4-28**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x54, PayloadLen: 0x04)			
Payload	IntensityLevel	4 bytes	Can get different intensity level form device.for detail refer to **Table 4-29**.

Table 4-28 NotifyLedPatternInfo packet format.

The Intensity level defines in (**Table 4-29**).

ID	Leve	Description
0x0	Waveform	99 steps from 1Hz~10000Hz
0x1	High pitch	99 steps from 1~100Hz
0x2	Alto	99 steps from 100Hz~1000Hz
0x3	Bass	99 steps from 1000Hz~10000Hz

Table 4-29 Intensity level.

When device led pattern change will report by `NotifyLedPattem` command:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x55, PayloadLen: 0x64)			
Payload	PatternID	1 byte	Device LED pattern to be identified, refer to Table 4-25.
	DefaultStatus	99 bytes	Default status of LED pattern, refer to **Table 4-25 and 4-27.**

Table 4-30 NotifyLedPattem packet format.

4.6.7 SetBrightness

App can change device led brightness by `SetBrightness` command (**Table 4-31**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x56, PayloadLen: 0x01)			
Payload	BrightnessLevel	1 byte	App change device led brightness.

Table 4-31 SetBrightness packet format.

When device color sensor captures the color send to app by `RetColorSniffer` command (**Table 4-32**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x57, PayloadLen: 0x01)			
Payload	ColorValue	1 byte	Device report Color code to app.

Table 4-32 RetColorSniffer packet format.

4.7 Pulse3 LED pattern Control

4.7.1 ReqLedPatternInfo

When the app connected with the JBL pulse 3, app can query device led pattern information by `ReqLedPatternInfo` command. See **Table 4-33** for details.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x51, Payload Len: 0x00)			
Payload	NA		

Table 4-33 ReqLedPatternInfo packet format

4.7.2 RetLedPatternInfo

If device received this command, it will reply a `RetLedPatternInfo` command to report all pattern information to App. See **Table 4-34** for details. It will also replay a `NotifyLedPattem` command to notify App the current using pattern information.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x52, Payload Len: 0x28)			
Payload	Pattern ID	1 byte	Device LED pattern, refer to **Table 4-35**.
	Current Color or Effects	3 bytes	Current color for Pattern ID 1-7: 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue Current effect for Pattern ID 8: 1st Byte: Value for Low freq Light Effect number N 2nd Byte: Value for High freq Light Effect number N

			3rd Byte: Value for Mid freq Light Effect number N Value of N is from 1 to 9 when selected and 0 is for no effect selected.
	Default status	1 byte	Whether is it default color or effect: 1,Yes 0, No
	…	…	Up to 8 Pattern ID and its Stored color or Effect. Refer to Table 4-35 to get the pattern ID.

Table 4-34 RetLedPatternInfo packet format

The `Pattern ID` defines in **Table 4-35**.

Pattern Name	ID	Color/ Effects
Wave	0x1	R: G: B:
Jet	0x2	R: G: B:
Explosion	0x3	R: G: B:
Equalizer	0x4	R: G: B:
Rave	0x5	R: G: B:
Cross	0x6	R: G: B:
Fire	0x7	R: G: B:
Customize	0x8	Light Effect High: Light Effect Mid: Light Effect Low:

Table 4-35 Pattern ID definitions

4.7.3 SetLedPattern

App can change Color or effects of device Led Pattern with `SetLedPattern` command (**Table 4-36**). If the user clicks Reset button or swipe to a pattern which have default color or effects, the Reset to default region should be set to 01. Otherwise it should be set to 00. If the device received `SetLedPattern` command which the Reset to default region is 01, it should automatically ignore the Current Color or Effects region value.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x53, Payload Len: 0x04)			
Payload	Pattern ID	1 byte	LED pattern ID, refer to **Table 4-35**.
	Current Color or Effects	3 bytes	Current color for Pattern ID 1-7: 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue

			Current effect for PatternID 8: 1st Byte: Value for Low freq Light Effect number N 2nd Byte: Value for High freq Light Effect number N 3rd Byte: Value for Mid freq Light Effect number N Value of N is from 1 to 9 when selected and 0 is for no effect selected.
	Reset to default	1 byte	Whether to reset to default color or effect: 1,Yes 0, No

Table 4-36 SetLedPattern packet format.

Device will notify its current pattern with its information just after App send out `ReqLedpatternInfo` or `SetLedPattern` command, or when device led pattern was changed by shake or by Light Key. It is `NotifyLedPattern` command (**Table 4-37**).

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x54, Payload Len: 0x04)			
Payload	Pattern ID	1 byte	LED pattern ID, refer to **Table 4-35**.
	Current Color or Effects	3 bytes	Current color for Pattern ID 1-7: 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue Current effect for Pattern ID 8: 1st Byte: Value for Low freq Light Effect number N 2nd Byte: Value for High freq Light Effect number N 3rd Byte: Value for Mid freq Light Effect number N Value of N is from 1 to 9 when selected and 0 is for no effect selected.
	Default status	1 byte	Whether is it default color or effect: 1, Yes 0, No

Table 4-37 NotifyLedPattern packet format.

4.7.4 SetPatternBrightness

App can change the global pattern brightness. When device received `SetPatternBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x55, PayloadLen: 0x01)			
Payload	Brightness value	1 byte	0x00~0xFF

Table 4-38 `SetPatternBrightness` packet format

4.7.5 ReqPatternBrightness

Device will set global pattern brightness accordingly. App need to query value of global pattern brightness once connected with Device by `ReqPatternBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x56, PayloadLen: 0x00)			
Payload	NA		

Table 4-39 `ReqPatternBrightness` packet format

4.7.6 RetPatternBrightness

And device will return global pattern brightness value to App by `RetPatternBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x57, PayloadLen: 0x01)			
Payload	Brightness value	1 byte	0x00~0xFF

Table 4-40 `RetPatternBrightness` packet format

4.7.7 NotifyPatternBrightness

Device will notify global pattern brightness value to App by `NotifyPatternBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x58, PayloadLen: 0x01)			
Payload	Brightness value	1 byte	0x00~0xFF

Table 4-41 `NotifyPatternBrightness` packet format

4.8 Pulse5 LED Package Control

Package Name	Pattern Name	Value
Nature	Campfire	0x01
	Northern Lights	0x02
	Sea Wave	0x03
	Universe	0x04
Party	Strobe	0x05
	Equalizer	0x06
	Geometry	0x07
	Spin	0x08
	Rainbow	0x09
Spiritual	Dynamic Wave	0x0A
	Lava	0x0B
	Focus	0x0C
Weather	Sky(sunny)	0x0D
	Rain	0x0E
	Snow	0x0F
	Storm	0x10
	Cloud	0x11
	Thunder	0x12
Cocktail	Fruit Gin	0x13
	Mojito	0x14
	Takila Sunrise	0x15
	Cherry Margarita	0x16

Pattern Table

4.8.1 ReqLedPackageInfo

When the app connected with the JBL pulse 5, app can query device led package information by `ReqLedPackageInfo` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x83, Payload Len: 0x00)			
Payload	NA		

4.8.2 RetLedPackageInfo

If device received command `ReqLedPackageInfo`, it would reply a `RetLedPackageInfo` command to report all package information to App.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x84, Payload Len: 0xnn)			
Payload	Package Count	1 byte	Package Count
	Active Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL

				0x04: COCKTAIL 0x05: WEATHER 0xC1: CANVAS
	Package 1	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER
		Active Pattern Count	1 byte	Active Pattern Count.
		Pattern Count	1 byte	Full Pattern Count of this package. (Include active patterns and inactive patterns), Default value is 5.
		Patterns	N bytes	Full patterns with active patterns and inactive patterns. (e. g. active pattern count is 3 and pattern count is 5, then the first 3 patterns are active, the last 2 patterns are inactive)
		Color Effect	1 byte	0: Static color 1: Colorful (loop color)
		Static Color	3 bytes	If color effect is **Colorful**, this field should be previous static color 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue
	…	…		
	Package CANVAS (if has this package)	Package ID	1 byte	0xC1: CANVAS
		Active Pattern Count	1 byte	Pattern Count, max value is 5
		Pattern 1 Pattern ID	2 bytes	Pattern ID
		Pattern 2 Pattern ID	2 bytes	Pattern ID
		… …	…	…

4.8.3 SwitchLedPackage

App can switch Led Package with `SwitchLedPackage` command. Device reply `NotifyLedPackageInfo` to App.

ction	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x90, Payload Len: 0x01)			
Payload	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER 0xC1 CANVAS

4.8.4 SetLedPackage

App can change Color or effects of device Led Package with `SetLedPackage` command. Device reply `NotifyLedPackageInfo` to App.

Set normal package as below format:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x85, Payload Len: 0xnn, 0x09~0x0B)			
Payload	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER
	Active Pattern Count	1 byte	Active Pattern Count
	Pattern Count	1 byte	Full Pattern count of this package. (Include active patterns and inactive patterns), Default value is 5.
	Patterns	N bytes	Full patterns with active patterns and inactive patterns. (e. g. Active pattern count is 3 and pattern count is 5, then the first 3 patterns are active, the last 2 patterns are inactive)
	Color Effect	1 byte	0: Static Color 1: Colorful (loop color)
	Custom Color	3 bytes	If color effect is **Colorful**, this field should be previous static color 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue

4.8.5 SetLedCanvasPackage

App can change effects of device Led Package with `SetLedCanvasPackage` command. Device reply `NotifyLedPackageInfo` to App.

Set Canvas Package as below format

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x86, Payload Len: 0x01D2, 2 bytes Big Endian)			
Payload	Package ID	1 byte	0xC1: CANVAS
	Enable Status	1 byte	0x00: Disable/Remove this Canvas Pattern 0x01: Enable/add this Canvas Pattern
	Pattern ID	2 bytes	Pattern ID (scope: 0x0000 ~ 0x0004)
	Patterns	462 bytes	LED info (This field can be ignored in “Disable/Remove” command)

4.8.6 PreviewPattern

App can request a preview for pattern by command `previewPattern`. And this command will not change pattern “Enable Status”. Device reply `notifyLedPatternInfo` to App.

Preview pattern as below format

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x91, Payload Len: 0x02)			
Payload	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER
	Pattern ID	1 byte	Pattern ID

4.8.7 PreviewCanvasPattern

App can request a preview for canvas pattern by command `previewCanvasPattern`. And this command will not change pattern “Enable Status”. Device reply `notifyLedPatternInfo` to App.

Preview canvas pattern as below format

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x92, Payload Len: 0x01D1, 2 bytes Big Endian)			
Payload	Package ID	1 byte	0xC1: CANVAS
	Pattern ID	2 bytes	Pattern ID Pattern ID (scope: 0x0000 ~ 0x0004)
	Patterns	462 bytes	LED info

4.8.8 NotifyLedPackageInfo

Once device LED package changed by app or device, device should notify app current package info by command `NotifyLedPackageInfo`.

Notify app while LED info is normal package:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x87, Payload Len: 0xnn, 0x09 ~ 0x0B)			
Payload	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER
	Active Pattern Count	1 byte	Active Pattern Count.

	Pattern Count	1 byte	Full Pattern Count of this package. (Include active patterns and inactive patterns), Default value is 5.
	Patterns	N bytes	Full patterns with active patterns and inactive patterns. (e. g. Active pattern count is 3, Pattern count is 5, then first 3 patterns are active, the last 2 patterns are inactive)
	Color Effect	1 byte	0: Static Color 1: Colorful (loop color)
	Custom Color	3 bytes	If color effect is **Colorful**, this field should be previous static color 1st Byte: Value for Red 2nd Byte: Value for Green 3rd Byte: Value for Blue

Notify app while LED info is CANVAS package:

Section	Field		Size	Description
Header (Identifier: 0xAA, Command ID: 0x87, Payload Len: 0xnn)				
Payload	Package ID		1 byte	0xC1: CANVAS
	Pattern Count		1 byte	Pattern number, max value is 5
	Pattern 1	Pattern ID	2 bytes	
	Pattern 2	Pattern ID	2 bytes	
	…			

4.8.9 NotifyLedPatternInfo

Once playing pattern changed, device should notify app current playing pattern with command `NotifyLedPackageInfo`. Command can be configured by command `EnableNotifyLedPatternInfo`.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x88, Payload Len: 0x02)			
Payload	Package ID	1 byte	0x01: NATURE 0x02: PARTY 0x03: SPIRITUAL 0x04: COCKTAIL 0x05: WEATHER
	Current Pattern ID	1 byte	Refer to pattern table

Notify pattern to app while LED info is CANVAS package:

Section	Field	Size	Description

Header (Identifier: 0xAA, Command ID: 0x88, Payload Len: 0x03)			
Payload	Package ID	1 byte	0xC1: CANVAS
	Pattern ID	2 bytes	

4.8.10 EnableNotifyLedPatternInfo

App can enable / disable NotifyLedPatternInfo.

Set normal package as below format:

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x89, Payload Len: 0x01)			
Payload	Enable/Disable	1 byte	0x00: Disable NotifyLedPatternInfo 0x01: Enable NotifyLedPatternInfo

4.8.11 SetLedBrightness

App can change the global package brightness. When device received `SetLedBrightness` device should reply RetBrightness.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8A, PayloadLen: 0x03)			
Payload	Brightness value	1 byte	0x00~0xFF
	Bright status	1 byte	0x00: off 0x01: on
	Project status	1 byte	0x00: off 0x01: on

4.8.12 ReqLedBrightness

Device will set global pattern brightness accordingly. APP need to query value of global brightness once connected with Device by `ReqLedBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8B, PayloadLen: 0x00)			
Payload	NA		

4.8.13 RetLedBrightness

And device will return global pattern brightness value to App by `RetLedPackageBrightness:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8C, PayloadLen: 0x03)			

Payload	Brightness value	1 byte	0x00~0xFF
	Bright status	1 byte	0x00: off 0x01: on
	Project status	1 byte	0x00: off 0x01: on

4.8.14 SetLedMovementSpeed

App can change the LED movement speed. When device received `SetLedDynamicMovement` device should reply `RetLedDynamicMovement`.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8D, PayloadLen: 0x01)			
Payload	Dynamic Movement	1 byte	0x01: Low 0x02: Mid 0x03: High

4.8.15 ReqLedMovementSpeed

APP query value of global LED movement speed once connected with Device by `ReqLedDynamicMovement:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8E, PayloadLen: 0x00)			
Payload	NA		

4.8.16 RetLedMovementSpeed

And device will return global LED movement speed value to App by `RetLedDynamicMovement:`

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x8F, PayloadLen: 0x01)			
Payload	Dynamic Movement	1 byte	0x01: Low 0x02: Mid 0x03: High

4.9 Speaker Settings

4.9.1 EQ Settings (BoomBox):

4.9.1.1 ReqEQMode(BoomBox):

Application request the device EQ status with command ReqEQMode.

Below is the Packet format for ReqEQMode.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x61, PayloadLen: 0x00)			

4.9.1.2 RetEQMode(BoomBox):

Once the Device receives the ReqEQMode it will reply the EQ status to the app with RetEQMode.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x62, PayloadLen: 0x01)			
Payload	EQ value	1 byte	0x00~0xFF

4.9.1.3 SetEQMode(BoomBox):

App request to change EQ mode. Below data will be send in payload to identify which mode to set. 0 for Indoor. 1 for Outdoor.
Device will send DevACK to the application for the status of the operation.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x63, PayloadLen: 0x01)			
Payload	EQ value	1 byte	0x00~0xFF

4.9.1.4 NotifyEQChangeMode(BoomBox):

Device will send this message whenever there is change in EQMode.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x64, PayloadLen: 0x01)			
Payload	EQ value	1 byte	0x00~0xFF

4.9.2 General EQ Settings:

EQ Name	Value (Category ID)
Default	0x01
…	…
Custom EQ 1	0xC1
…	…

Table 4-8-2-1 General EQ

EQ Parameter Name	Value
Bass	0x01
Mid	0x02
Treble	0x03
…	…

Table 4-8-2-2 General EQ Parameter

EQ Parameter Scope	Value
[-1, 1]	0x01
[-2, 2]	0x02
…	…

Table 4-8-2-3 General EQ Parameter Scope

4.9.2.1 ReqEQ:

Application request the device EQ status with command ReqEQ.

Below is the Packet format for ReqEQ.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x6C, Payload Len: 0x00)			

4.9.2.2 RetEQ:

Once the Device receives the ReqEQ, device will reply to app the EQ status by RetEQ. Preset EQ's parameter can be ignored.

Section	Field			Size	Description
Header (Identifier: 0xAA, Command ID: 0x6D, PayloadLen: 0xnn)					
Payload	EQ parameter scope			1 byte	Refer to column **value** of Table 4-8-2-3 General EQ parameter Scope
	Current Active EQ ID			1 byte	Active EQ ID, refer to Table **4-8-2-1** General EQ
	EQ 1	ID		1 byte	Refer to Table **4-8-2-1** General EQ
		Parameter count		1 byte	EQ parameter count
		Parameter 1	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
			Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1
		Parameter 2	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
			Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1

		…	…		
	EQ 2	ID		1 byte	Refer to Table **4-8-2-1** General EQ
		Parameter count		1 byte	EQ parameter count
		Parameter 1	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
			Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1
		Parameter 2	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
			Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1
		…	…		
	EQ 3 …				

4.9.2.3 SetCurrentEQ:

App request to change EQ mode. Device should send DevAck to the application for the status of the operation. Preset EQ's parameter can be ignored.

Section	Field		Size	Description
Header (Identifier: 0xAA, Command ID: 0x6E, Payload Len: 0xnn)				
Payload	EQ ID		1 byte	Refer to Table **4-8-2-1** General EQ
	EQ Parameter count		1 byte	EQ parameter count
	Parameter 1	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
		Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1
	Parameter 2	ID	1 byte	Refer to Table 4-8-2-2 General EQ Parameter
		Value	1 byte	Refer to Column **EQ Parameter Scope** of Table 4-8-2-3 e.g. -1, 0, 1
	…	…		

4.9.3 EQ Settings (3.0)

New EQ settings include preset and user custom 2 parts.

4.9.3.1 Preset EQ Band Type:

BAND Type	Value
IIR_BIQUARD_LOWSHELF (Low Shelf Filter)	0x00
IIR_BIQUARD_PEAKINGEQ (Peaking EQ)	0x01
IIR_BIQUARD_HIGHSHELF (High Shelf Filter)	0x02
IIR_BIQUARD_LPF (Low Pass Filter)	0x03
IIR_BIQUARD_HPF (High Pass Filter)	0x04

4.9.3.2 App Preset EQ:

Payload	EQ Category			1 byte	Preset EQs: 0x03: VOCAL 0x06: JBL SIGNATURE 0x07: RELAXING 0x08: ENERGETIC 0x09: EXTREME 0x21: RESERVED_1 0x22: RESERVED_2 0x23: RESERVED_3 0x24: RESERVED_4 0x25: RESERVED_5 0x26: RESERVED_6 0x27: RESERVED_7 0x28: RESERVED_8 0x29: RESERVED_9 0x2a: RESERVED_10 0x2b: RESERVED_11
	Band Count			1 byte	5, Number of the band count.
	Sample Rate			4 bytes	
	IIR param	Band0 …	Type	1 byte	EQ Band type
			Gain	4 bytes	Gain value
			Fc	4 bytes	Frequency
			Q	4 bytes	Q value
		Band (count – 1)	…		…

4.9.3.3 Custom EQ Band:

EQ Parameter Name	Value
Band 1	0x01
Band 2	0x02
Band 3	0x03
Band 4	0x04
Band 5	0x05

4.9.3.4 Custom EQ Level Scope:

EQ Level Scope	Value
[-1, 1]	0x01
[-2, 2]	0x02
…	…

4.9.3.5 Custom EQ Payload:

Payload	EQ Category			1 byte	0xC1: **CUSTOM_EQ**
	EQ Level Scope			1 byte	Custom EQ Level cope
	Band Count			1 byte	5, Number of the band count.
	IIR param	Band0 …	Band ID	1 byte	Band ID Value
			Level	1 byte	
		Band (count – 1)	…		…

4.9.3.6 Req/Set/Ret NTI EQ

App can set current EQ by command setEQs

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: **0x97**, Payload Len: 0xnnnn, 2 bytes in Big Endian)			
Payload	Active EQ category	1 byte	**Preset EQ**s: 0x03: VOCAL 0x06: JBL SIGNATURE 0x07: RELAXING 0x08: ENERGETIC 0x09: EXTREME 0x21: RESERVED_1 0x22: RESERVED_2 0x23: RESERVED_3 0x24: RESERVED_4 0x25: RESERVED_5 0x26: RESERVED_6 0x27: RESERVED_7 0x28: RESERVED_8 0x29: RESERVED_9 0x2a: RESERVED_10 0x2b: RESERVED_11 0xC1: **CUSTOM_EQ**
	Custom EQ (This field should be ignored if active EQ category is Preset EQ)	N bytes	**Custom EQ payload**
	Preset EQ payload (This field can be ignored if active EQ category is CUSTOM_EQ)	N bytes	**DSP team provide formula, which can be used to transfer App preset EQ to this payload for firmware.**

App can query current running EQ setting via `reqEQs`.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: **0x98**, Payload Len: 0x0000)			

Command `retEQs` is response for `reqEQs` and `setEQs` from device to app.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: **0x99**, Payload Len: 0xnnnn, 2 bytes in Big Endian)			
Payload	Active EQ category EQ category	1 byte	**Preset EQ**s: 0x03: VOCAL 0x06: JBL SIGNATURE 0x07: RELAXING 0x08: ENERGETIC 0x09: EXTREME 0xC1: **CUSTOM_EQ**
	Custom EQ	N bytes	Custom EQ payload **Content should be previous CUSTOM_EQ if active EQ category is preset EQ.**

4.9.4 Speaker Feedback Tone:

4.9.4.1 ReqFeedbackToneStatus:

Application will request the speaker feedback status with ReqFeedbackToneStatus.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x65, Payload Len: 0x00)			

4.9.4.2 RetFeedbackToneStatus:

Device will return the feedback status with RetFeedbackToneStatus.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x66, Payload Len: 0x01)			
Payload	Tone Value	1 byte	0x00~0xFF

4.9.4.3 SetFeedbackTone:

App request to set the tone status. Below data is sent in payload to identify to set tone on or off.
0: off, 1: on.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x67, Payload Len: 0x01)			
Payload	Tone Value	1 byte	0x00~0xFF

4.9.5 HFP ENABLE/DISABLE

4.9.5.1 ReqHFPStatus:

Application request the HFP status with this command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x68, PayloadLen: 0x00)			

4.9.5.2 RetHFPStatus:

Device returns the HFP status with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x69, PayloadLen: 0x01)			
Payload	HFP Value	1 byte	0x00~0xFF

4.9.5.3 SetHFPMode:

Application will change the HFP mode with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x70, PayloadLen: 0x01)			
Payload	HFP Value	1 byte	0x00~0xFF

4.9.5.4 RetLinkMode:

Device returns the link status with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x6b, PayloadLen: 0x01)			
Payload	Value	1 byte	0x00: Off 0x01: Partyboost 0x02: Auracast

4.9.5.5 SetLinkMode:

Application will change the link mode with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x6a, PayloadLen: 0x01)			
Payload	Value	1 byte	0x00: Off 0x01: Partyboost 0x02: Auracast

4.9.6 Serial Number

4.9.6.1 App get serial number by `ReqSerialNumber` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7a, Payload Len: 0x00)			

4.9.6.2 Device reply to app with serial number by `RetSerialNumber` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7b, Payload Len: 0xnn)			
Payload	Device serial number	Max 64 bytes	ASCII, max 64 bytes.

4.9.7 One Touch Music Button

4.9.7.1 User set One Touch Music button on App by `setOneTouchMusicButton` command, and device should reply to App with button settings by `retOneTouchMusicButton` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7c, Payload Len: 0x02)			
Payload	Action / Gesture	1 byte	0x01: Long press 0x02: Double click
	Button ID	1 byte	0x01: Lighting button 0x02: Play button

4.9.7.2 Request One Touch Music button settings by command `reqOneTouchMusicButton`, and device should reply to App with button settings by `retOneTouchMusicButton` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7d, Payload Len: 0x00)			

4.9.7.3 Response of set / req One Touch Music button settings, device reply to App by `retOneTouchMusicButton` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7e, Payload Len: 0x02)			
Payload	Action / Gesture	1 byte	0x01: Long press 0x02: Double click
	Button ID	1 byte	0x01: Lighting button 0x02: Play button

4.9.7.4 Once user pressed One Touch Music button on speaker, device should reply to app with button pressed status by `retOneTouchMusicTriggered` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x7f, Payload Len: 0x00)			

4.9.8 Standby Mode

4.9.8.1 User set Standby mode on/off App by `setStandbyMode` command, and device should reply to App with standby status by `retStandbyMode` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x9a, Payload Len: 0x01)			
Payload	Standby mode	1 byte	0x00: off 0x01: on

4.9.8.2 Request Standby mode by command `reqStandbyMode`, and device should reply to App with standby status by `retStandbyMode` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x9b, Payload Len: 0x00)			

4.9.8.3 Response of set / req standaby mode, device reply to App by `retStandbyMode` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x9c, Payload Len: 0x01)			
Payload	Standby mode status	1 byte	0x00: off 0x01: on

4.9.9 LE Audio Status

4.9.9.1 The users can turn on/off LE Audio feature by `setLeAudioStatus` command, and device should reply to App with new LE Audio status by `retLeAudioStatus` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0xA3, Payload Len: 0x01)			
Payload	LE Audio Status	1 byte	0x00: turn off LE Audio 0x01: turn on LE Audio

4.9.9.2 Request LE Audio status by command `reqLeAudioStatus`, and device should reply to App with current LE Audio status by `retLeAudioStatus` command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0xA4, Payload Len: 0x00)			

4.9.9.3 Response of set / req LE Audio status, device reply to App by `retLeAudioStatus` command. And if user switch A2DP/LE-Audio in mobile system settings, device also should push new LE Audio status to App by this command.

Section	Field	Size	Description

Header (Identifier: 0xAA, Command ID: 0xA5, Payload Len: 0x01)			
Payload	LE Audio Status	1 byte	0x00: LE Audio off 0x01: LE Audio on

4.9.10 ReqLightStatus

The APP should request the light status first.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x71, PayloadLen: 0x00)			
Payload	NA		

Table 4-58 ReqLightStatus packet format

4.9.11 RetLightStatus

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x72, PayloadLen: 0x01)			
Payload	Light status	1 byte	0: off 1: on.

Table 4-59 RetLightStatus packet format

4.9.12 SetLightStatus

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x73, PayloadLen: 0x01)			
Payload	Light status	1 byte	0: off 1: on.

Table 4-60 SetLightStatus packet format

4.9.13 ReqImageStart

To show a Picture on LED, APP should send ReqImageStart to device.

Figure 4-3 shows the image flow.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x74, PayloadLen: 0x00)			
Payload	NA		

Table 4-61 ReqImageStart packet format

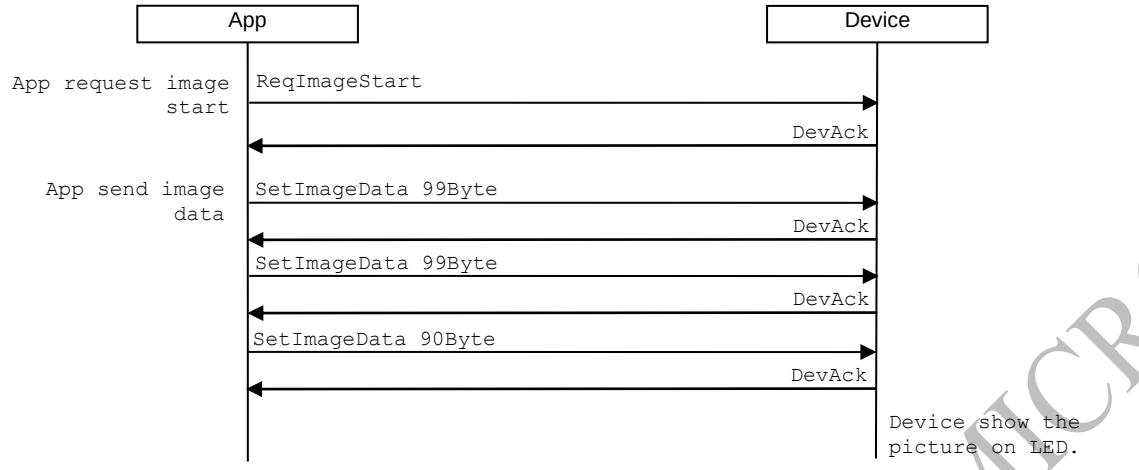


Figure 4-3 Image flow

4.9.14 SetImageData

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x75, PayloadLen: 0xnn)			
Payload	Image Data	n bytes	Image content.

Table 4-62 SetImagedata packet format

Payload: <flag><size><Image data>;
flag = 0x54;
size = Image data size + 1.

The LED array on Pulse3 is 12 columns wide and 8 rows high making 12 x 8 as the Resolution. Each LED illuminates based on RGB Value. App should consider the Image Resolution and the format of the Image before sending data to device. An Image Color Map should contain 8x12x3 = 288 Bytes of RGB Data, and it is suggested to divide the data into 3 parts, containing 99 Bytes, 99 Bytes, and 90 Bytes separately (99 + 99 + 90 = 288).

The RGB Data should be in Raster scan format, starting from the Top Left corner of the Image Color Map.

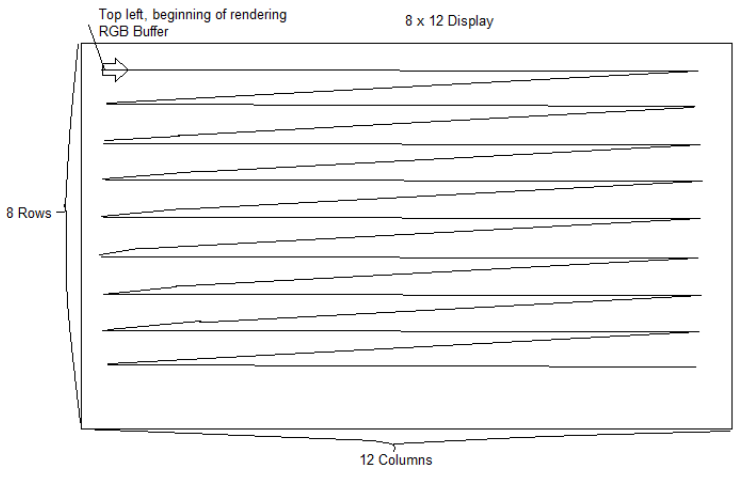


Figure 4-3 Raster scan format

4.9.15 Bass volume adjustment:

4.9.15.1 SetBassVolume

Application set the bass volume with this command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x76, PayloadLen: 0x01)			
Payload	Bass Value	1 byte	1~21

4.9.15.2 ReqBassVolume

Application get the bass volume with this command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x77, PayloadLen: 0x00)			

4.9.15.3 RetBassVolume

The device returns the bass volume with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x78, PayloadLen: 0x01)			
Payload	Bass Value	1 byte	1~21

4.9.16 Auracast Group Information:

Auracast Group Info	Key ID	Value Length	Value
Group Action	0x01	1 byte	0x01: Create Group 0x02: Destroy Group 0x03: Prepare to Create Stereo Group, App would send this Action to primary device while user try to create Auracast Stereo Group, primary device should play tone till user set channel info to the primary device.
Group Type	0x02	1 byte	0x01: Stereo 0x02: Party
Role (Reserved)	0x03	1 byte	0x01: Primary 0x02: Secondary
Stereo Channel	0x04	1 byte	0x00: Full Channel 0x01: Left Channel 0x02: Right Channel
Stereo Group ID	0x05	4 bytes	ASCII. Generated in App. First 4 bytes of MD5 value of current timestamp and random value in [1, 10000]. Pseudo Code like: md5ForString(currentTimeMillis.toString()

			+ randomValue.toString(), UTF-8).substring(0, 8)
Stereo Group Name (Reserved)	0x06	Max to 30 bytes	UTF-8. The default group name will be generated in the primary device. And the default name should like “My Pulse 5 Stereo Group”
Partner’s MAC address (Reserved)	0x07	6 bytes	Partner’s BT MAC Address

4.9.16.1 setAuracastGroup

Application set the Auracast group with this command.

Section	Field		Size	Description
Header (Identifier: 0xAA, Command ID: 0xA0, Payload Len: 0xnn)				
Payload	Group Info 1	Feature Key	1 byte	Refer to Key ID of Table Group Features
		Feature Value Length	1 byte	Length of below value
		Value	N bytes	Refer to column Value of Table Group Features
	Group Info N	Feature Key	1 byte	Refer to Key ID of Table Group Features
		Feature Value Length	1 byte	Length of below value
		Value	N bytes	Refer to column Value of Table Group Features

4.9.16.2 reqAuracastGroup

Application retrieves the Auracast info with this command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0xA1, Payload Len: 0x00)			

4.9.16.3 retAuracastGroup

The device returns the Auracast info with this message for command setAuracastGroup and reqAuracastGroup

Section	Field		Size	Description
Header (Identifier: 0xAA, Command ID: 0xA2, Payload Len: 0xnn)				
Payload	Group Info 1	Feature Key	1 byte	Refer to Key ID of Table Group Features
		Feature Value Length	1 byte	Length of below value
		Value	N bytes	Refer to column Value of Table Group Features
	Group Info N	Feature Key	1 byte	Refer to Key ID of Table Group Features
		Feature Value Length	1 byte	Length of below value
		Value	N bytes	Refer to column Value of Table Group Features

4.9.17 Water in charging port and overheating detection:

4.9.17.1 ReqWaterOverHeating

The Application get the device log of water / overheating in/with charging port with this command.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x79, PayloadLen: 0x00)			

4.9.17.2 RetWaterOverHeating

Device returns the water / overheighting in/with charging port with this message.

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x80, PayloadLen: 0x01)			
Payload	Water in charging port Overheating with charging port	1 byte	Bit 0: indicate whether there was water in charging port Bit 1: indicate whether charging port was overheating 0 means no, 1 means yes

4.9.18 Device information (Battery Status & Playback time duration):

4.9.18.1 Device information and Battery Status Feature Table

Feature Type	Key ID	Value Length	Value
All	0x00	None	This field is defined to fetch value of all features.
Battery ID	0x01	Max to 16 bytes	ASCII.
Remaining playtime	0x02	2 bytes, big endian	0 ~ 65535 minutes
Temperature Max	0x03	2 bytes, big endian	Celsius degree X 10
Remaining Capacity	0x04	2 bytes, big endian	0 ~ 65536 mAh
Full Charge Capacity	0x05	2 bytes, big endian	0 ~ 65536 mAh
Design Capacity	0x06	2 bytes, big endian	0 ~ 65536 mAh
Cycle Count	0x07	2 bytes, big endian	0 ~ 65536 cycles
State of Health	0x08	1 byte	0 ~ 100 %
Charging Status	0x09	1 byte	0x01: AC charging. 0x02: DC. 0x03: full charged. 0x04: full depleted.
Battery health notification	0x0A	1 byte	0x00: Health 0x01: Low Battery Health
Total power on duration	0x0B	4 bytes, big endian	Total power on duration in minutes.
Total playback time duration	0x0C	4 bytes, big endian	Total playback duration in minutes.
…			…

4.9.18.2 ReqDeviceBatteryInformation

Application can retrieve the device battery status by this command. Device response command is RetDeviceBatteryInformation

Section	Field		Size	Description
Header	Identifier		1 byte	0xAA
	Command ID		1 byte	0x9D
	Payload Len		1 byte	Payload length
Payload	Feature 1	Key ID	1 byte	Refer to column Key ID in Battery Status Feature Table
	Feature N	Key ID	1 byte	Refer to column Key ID in Battery Status Feature Table

4.9.18.3 RetDeviceBatteryInformation

Device replies to App with battery status once device received command ReqBatteryStatus. And device need to push battery status to App while device charging status/health status has been changed or full charging/low battery event has been triggered.

Section	Field	Size	Description
Header	Identifier	1 byte	0xAA

	Command ID		1 byte	0x9E
	Payload Len		1 byte	Payload length
Payload	Feature 1	Key ID	1 byte	Refer to column Key ID in Battery Status Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Battery Status Feature Table
	Feature N	Key ID	1 byte	Refer to column Key ID in Battery Status Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Battery Status Feature Table

4.10 Player Settings

Player Feature	Key ID	Value Length	Value
Play/Pause	0x01	1 byte	0x00: Pause. 0x01: Play.
Prev/Next	0x02	1 byte	0x00: Prev 0x01: Next
Volume level	0x03	1 byte	Value Scope: 0~32
…			
Below fields for debugging only			
Clear LC3 lost audio analytics	0xD0	0 byte	Clear LC3 lost audio analytics
Total package count	0xD1	4 bytes, big endian	Total lost package count
Total lost package count	0xD2	4 bytes, big endian	Total lost package count
Total lost left channel package count	0xD3	4 bytes, big endian	Total lost left channel package count
Total lost right channel package count	0xD4	4 bytes, big endian	Total lost right channel package count
Total lost left and right channel package count	0xD5	4 bytes, big endian	Total lost left and right channel package count

Player Feature Table

Application can retrieve the device play status by this command. Device response command is RetPlayerInfo

Section	Field		Size	Description
Header	Identifier		1 byte	0xAA
	Command ID		1 byte	0xA8, request play status
	Payload Len		1 byte	Payload length
Payload	Feature 1	Key ID	1 byte	Refer to column Key ID in Player Feature Table
	Feature N	Key ID	1 byte	Refer to column Key ID in Player Feature Table

Application can set the device play status by setPlayerInfo command. Device response command is RetPlayerInfo

Section	Field		Size	Description
Header	Identifier		1 byte	0xAA
	Command ID		1 byte	0xA7
	Payload Len		1 byte	Payload length
Payload	Feature 1	Key ID	1 byte	Refer to column Key ID in Player Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Player Feature Table
	Feature N	Key ID	1 byte	Refer to column Key ID in Player Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Player Feature Table

Device respond App by command RetPlayerInfo

Section	Field		Size	Description
Header	Identifier		1 byte	0xAA
	Command ID		1 byte	0xA6, Response/notify of play status
	Payload Len		1 byte	Payload length
Payload	Feature 1	Key ID	1 byte	Refer to column Key ID in Player Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Player Feature Table
	Feature N	Key ID	1 byte	Refer to column Key ID in Player Feature Table
		Value Len	1 byte	Value length
		Value	N bytes	Refer to Column Value in Player Feature Table

4.11 Analytics

~~4.11.1 ReqAnalyticsData (deprecated)~~

~~Section~~	~~Field~~	~~Size~~	~~Description~~
~~Header (Identifier: 0xAA, Command ID: 0x81, PayloadLen: 0x00)~~			
~~Payload~~	~~NA~~	~~NA~~	~~NA~~

~~App will request for Analytics data after BLE Connection is done and Basic Settings are read~~

~~Analytics Data will be returned by **"RetAnalyticsCmd"** packet by Device~~

~~4.11.2 RetAnalyticsCmd (deprecated)~~

~~Section~~	~~Field~~	~~Size~~	~~Description~~
~~Header (Identifier: 0xAA, Command ID: 0x82, PayloadLen: <Number of Analytics Params> * 2)~~			
~~Payload~~	~~Analytics Data of all Params.~~	~~<Number of Analytics Params> * 2(i.e., 20 Bytes)~~	~~Analytics Data of all Params. Each Param is 2 Bytes. App should Parse byte array and read each Param~~

~~In Current version 10 Analytics parameters are implemented.~~

~~All Analytics parameters values are placed in Payload field sequentially (2 Bytes each).~~

~~As below~~

~~S. No~~	~~Parameter~~	~~Description~~
~~1~~	~~JBL Connect~~	~~No of times the DUT acts as Broadcaster/Receiver~~
~~2~~	~~Duration of JBL Connect~~	~~Time duration for which the DUT acts as Broadcaster/Receiver~~
~~3~~	~~Critical Temperature~~	~~No of times the DUT Battery reached critical temperature~~
~~4~~	~~Power bank~~	~~No. of times the DUT used as Power bank~~
~~5~~	~~Music Playtime~~	~~Time duration for which the music was played on DUT.~~
~~6~~	~~Music Playtime in Battery Mode~~	~~Time duration for which the DUT played the music in Battery mode~~
~~7~~	~~Charging Time~~	~~Time duration for which the Power adapter was connected to the DUT~~
~~8~~	~~Power ON Count~~	~~No. of times the DUT was Powered ON from Power OFF state~~

~~9~~	~~Duration Between Power ON and OFF~~	~~Time duration for which DUT was in Power ON state~~
~~10~~	~~Speakerphone~~	~~No. of times the DUT was used as Speakerphone~~
~~11~~	~~PD charging time~~	~~Charging time duration with USB-PD activated (in mins)~~
~~12~~	~~The most popular volume level~~	~~The most popular volume level~~
~~13~~	~~Usage time of the most popular volume level~~	~~Usage time in minutes of the most popular volume level~~

4.11.3 ReqAnalyticsData

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x93, PayloadLen: 0x00)			
Payload	NA	NA	NA

App will request for Analytics data after BLE/BR/EDR Connection is done and Basic Settings are ready.

Analytics Data will be returned by **"RetAnalyticsCmd"** packet by Device

4.11.4 RetAnalyticsCmd

Package category:

Package	ID
Universal Package	0x01
Light Pattern Package	0x02
Canvas Package	0x03
…	…

Analytics Feature of Universal Package:

Feature	ID
Number of times using PartyBoost/Auracast	0x01
Duration of using PartyBoost/Auracast	0x02
Music playback - total time (in mins).	0x03
Music playback - with power plug in (in mins)	0x04
Total charging time duration (in mins)	0x05
Total time duration of music playing in Aux-in	0x06
Number of times of DUT power on	0x07
Duration between power on and power off	0x08
Number of times that user presses Play/Pause	0x09
Number of times that user uses DUT as power bank	0x0A
Number of times that DUT reaches temperature threshold and auto shut down	0x0B
Number of times that smart control is triggered	0x0C
Number of times that USB waterproof alarm triggered	0x0D
Number of times that users adjust volume through physical buttons on device	0x0E

Number of times that users adjust volume through AVRCP	0x0F
The most popular volume level	0x10
Time duration of the most popular volume being used	0x11
Duration of both body light and projection light on	0x12
Duration of both body light and projection light off	0x13
Duration of body light on	0x14
Duration of projection light on	0x15
Projection light temperature protection activated since bootup (in minutes)	0x16
Duration of using Auracast Stereo	0x17
Duration of using Auracast Party	0x18

RetAnalyticsCmd

Section	Field			Size	Description
Header (Identifier: 0xAA, Command ID: 0x94, Payload Len: 0xnnnn, 2bytes Big Endian)					
Payload	Package Count			1 byte	Package Count
	Package 1	Package ID		1 byte	0x01: Universal Package
		Feature Count		1 byte	Features Count
		Feature 1	Feature ID	1 byte	Refer to package ID in **Analytics Feature Package**
			Value	2 bytes	Analytics Data in 2 bytes
		Feature 2	Feature ID	1 byte	Refer to package ID in **Analytics Feature Package**
			Value	2 bytes	Analytics Data in 2 bytes
		…	…		
	Package 2 (Supported in Pulse 5…)	Package ID		1 byte	0x02: Pattern light
		Pattern Count		1 byte	
		Pattern 1	Pattern ID	1 byte	

			Patterns playing with music duration time (in minutes)	2 bytes	
			Patterns playing duration time (in minutes)	2 bytes	
		Pattern 2	Pattern ID	1 byte	
			Patterns playing with music duration time (in minutes)	2 bytes	
			Patterns playing duration time (in minutes)	2 bytes	
		…	…		
	Package 3 (Supported in Pulse 5…)	Package ID		1 byte	0x03: Canvas
		Canvas Count		1 byte	
		Canvas 1	Pattern ID	2 bytes	
			Patterns playing with music duration time (in minutes)	2 bytes	
			Patterns playing duration time (in minutes)	2 bytes	
		Canvas 2	Pattern ID	2 bytes	
			Patterns playing with music duration time (in minutes)	2 bytes	
			Patterns playing duration time (in minutes)	2 bytes	
		…	…		

App will send **"AppAck"** with command Id **"RetAnalyticsCmd"** with status Success (0) or Fail (1)

If **"AppAck"** with status success then all Analytics data except play analytics on device's persistent memory will be reset

4.11.5 ReqPlayAnalyticsData

Section	Field	Size	Description
Header (Identifier: 0xAA, Command ID: 0x95, PayloadLen: 0x00)			
Payload	NA	NA	NA

App will request for play status analytics data after BLE/BR/EDR Connection is done and Basic Settings are ready.

Analytics Data will be returned by **"RetPlayAnalyticsCmd"** packet by Device

4.11.6 RetPlayAnalyticsCmd

Table of Play Analytics Features

Feature	ID	Value Scope
Play Number	0x01	Play number in this uploading content
Charging Status	0x02	0: DC playing 1: AC Charging playing

		2: Non-Playing charging
Volume Level	0x03	
Partyboost Status	0x04	0x00: Normal 0x01: PB Party Broadcasting 0x02: PB Party Receiver 0x03: PB Stereo
Auracast Status	0x05	0x00: Normal 0x01: Auracast Party Broadcasting 0x02: Auracast Party Receiver 0x03: Auracast Stereo
Audio In Status	0x06	0x01: A2DP 0x02: LE Audio
EQ Category ID	0x10	Refer to **NTI EQ Category**
…	…	
Play duration	0x30	Playing duration in minutes

RetPlayAnalyticsCmd
Each data set will store only when the speaker is in playback status and no parameter changes in 10 seconds. And play duration max value is 241 minutes, if duration up to 241, device should store current record and launch another new record.
[Note]: Speaker should record the "Non-Playing Charging status", and charging status value should be 0x02, other fields value should be 0.
Condition 1: "Non-playing charging status" record should only be stored if charging status = 0 at previous record (including playing and non-playing record)
Condition 2: A new "Playing charging status" record should override the previous "Non-playing charging status" record.

Section	Field			Size	Description
Header (Identifier: 0xAA, Command ID: 0x96, Payload Len: 0xnnnn, 2bytes Big Endian)					
Payload	Recorded Play Times			1 byte	Recorded play times. Case received play times less than recorded play times, which means data split into multi-packages, app should wait all data packages.
	Current Play Count			1 byte	Play count in current received data analytics packet.
	1st Play	Feature 1	ID	1 byte	Refer to column ID of **Above Play Analytics Feature Table**.
			Value	1 / 2 bytes	If feature ID is 0x30(play duration), value length should be 2 bytes, otherwise value length should be 1 byte.
		Feature N	…	…	Let play duration as the last feature.
	Nth Play	…		…	…

App will send **"AppAck"** with command Id **"RetPlayAnalyticsCmd"** with status Success (0) or Fail (1)

If **"AppAck"** with status success then all play analytics data on device's persistent memory will be reset

4.12 Vendor Codes and disabled feature (only support by Flip 6)

4.12.1 ReqVendorCodesAndFeatures

App can request vendor information from device by command ReqVendorCodesAndFeatures as below command, response should be command RetVendorCodesAndFeatures:

Section	Field	Size	Description
Header	Identifier	1 byte	0xAA
	Command ID	1 byte	0xAB
Payload Len		1 byte	The Payload Length, 0x00

4.12.2 ControlDeviceFeatures

App can disable device feature by command ControlDeviceFeature. And device should notify App with command RetVendorCodesAndFeatures:

Section	Field	Size	Description
Header	Identifier	1 byte	0xAA
	Command ID	1 byte	0xAC
Payload Len		1 byte	The Payload Length
Payload	Feature codes	N bytes	0x00: Recover all disabled features 0x01: Recover the disabled music playback feature. 0x02: Recover the disabled EQ feature. 0x03: Recover the disabled PartyBoost/Auracast feature. 0x11: Disable the music playback feature. 0x12: Disable EQ feature. 0x13: Disable the PartyBoost/Auracast feature.

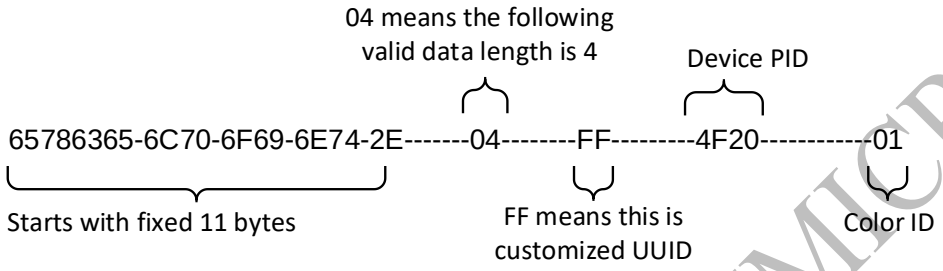
4.12.3 RetVendorCodesAndFeatures

The device notifies vendor information codes to App by command RetVendorCodesAndFeatures

Section	Field		Size	Description
Header	Identifier		1 byte	0xAA
	Command ID		1 byte	0xAD
Payload Len			1 byte	The Payload Length
Payload	Vendor Code	Type	1 byte	0x01: Vendor code
		Length	1 byte	
		Value	N bytes	Vendor information codes
	Disabled Features	Type	1 byte	0x02: Disabled features
		Length	1 byte	
		Value	N bytes	0x00: No disabled feature 0x11: Disable the music playback feature. 0x12: Disable EQ feature. 0x13: Disable the PartyBoost/Auracast feature.ssssss

4.13 Customized UUID format

Since Flip 6, we applied GATT over BR/EDR to connect/control device. For quick recognizing device by UUIDs within A2DP profile. We defined a special UUID in device, format as below: 65786365-6C70-6F69-6E74-2E-XX-XX-XX-XX-XX. Firmware team need to add an ATT service record and add the customized UUID in service description.
Example of black Flip 6.



5. Error Handling

For the ACK required commands, if no reply from device after 300ms, App shall resend the commands for 3 times. If all retry failed or any other exceptions happened, App and device should close the BLE connections. App will go back to discovery status. Device go back to wait for connection status.