



MYND

Software Specification for App Compatibility

Version 2025-05-23

Teufel

Table of Contents

1. MIT License	1
2. Bluetooth protocol requirements	1
2.1. Bluetooth Low Energy (BLE) Support	1
2.1.1. General Requirements	1
2.1.2. BLE Characteristics	1
2.1.3. BLE Advertisement of manufacturer data	2
2.1.4. BLE GATT Command Packet Framing	2
2.2. RFComm / Bluetooth Classic Support	3
2.2.1. General Requirements	3
2.2.2. RFComm Packet Framing	3
2.3. Acknowledgement (ACK)	3
3. Commands	4
3.1. Commands I	4
3.1.1. Battery	4
Get Current Battery Level	4
Get Power Adapter Connection Status	5
3.1.2. Auto-Off Timer	5
Get Auto-Off Timer Value	6
Set Auto-Off Timer Value	6
3.1.3. Device Color	6
Get Device Color	6
3.1.4. ECO-Mode	7
Enable ECO-Mode	7
Disable ECO-Mode	7
Get ECO-Mode Status	8
3.1.5. Equalizer	8
Get EQ Parameter Gain	8
Set EQ Parameter Gain	9
3.1.6. Master Volume	9
Get Master Volume	9
Set Master Volume	9
3.1.7. Master Mute	10
Master Mute Requirements	10
Get Master Mute Status	10
Set Master Mute Status	11
3.1.8. MCU Firmware	11
Get MCU Firmware Version	11
3.1.9. PartyLink Broadcast	12

Start PartyLink Broadcast	12
Stop PartyLink Broadcast	12
Get PartyLink Broadcast Status	12
3.1.10. Sound Icons	13
Disable Sound Icons	13
Enable Sound Icons	13
Get Sound Icons Status	14
3.1.11. Source Identifiers	14
3.1.12. Bluetooth Firmware	14
Get Bluetooth Firmware Version	14
3.2. Commands II	15
3.2.1. Battery Capacity	15
Get Battery Capacity	15
3.2.2. Battery Friendly Charging	15
Enable Battery Friendly Charging	15
Disable Battery Friendly Charging	16
Get Battery Friendly Charging Status	16
Get Battery Capacity	16
3.2.3. Multipoint	17
Enable Multipoint	17
Disable Multipoint	17
Get Multipoint Status	18
3.2.4. Source Selection	18
Get Source	18
Set Source	18
Get Connected Sources	19
3.3. Commands III	20
3.3.1. LED Brightness	20
Get LED Brightness	20
Set LED Brightness	20
4. Notifications	20
4.1. Register notification	21
4.2. Cancel notification	21
4.3. Event notification	21
4.4. Notifications list	22

1. MIT License

Copyright (c) 2025 Lautsprecher Teufel GmbH.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

2. Bluetooth protocol requirements

2.1. Bluetooth Low Energy (BLE) Support

2.1.1. General Requirements

- BLE advertising is active at all times. Note: This is required because only when a unit advertises BLE other devices can connect to it via BLE.
- The advertised BLE device name shall be "TMYND_BLE". The device name for BT classic remains "MYND", as specified in the general product specification.
- When the speaker is in pairing mode, it advertises the **Teufel Pairing Service** - a BLE service with the following UUID: **ce1eb45c-1bd2-45be-8163-acc88be94cb2**. This BLE service offers no characteristics.
- When the speaker is **not** in pairing mode, it **is not** advertise the Teufel Pairing Service.
- When the speaker is **not** in pairing mode, it **is not** possible for other devices to pair with it.

2.1.2. BLE Characteristics

- The App Protocol interface is usable via GATT over Bluetooth Low Energy.
- The Command service UUID for GATT over BLE is: **00001100-d102-11e1-9b23-00025b00a5a5**
 - Characteristics inside this service is:

- `00001101-d102-11e1-9b23-00025b00a5a5` to write commands
- `00001102-d102-11e1-9b23-00025b00a5a5` to read responses and notifications
- The App Protocol interface is only usable by a mobile device that has been paired with the device.

2.1.3. BLE Advertisement of manufacturer data

- The BLE advertisement has to advertised our specific manufacturer data:

Name	Offset	Format	Value
Manufacturer ID	0	UINT16	<code>0x0ad2</code>
Product ID	2	UINT16	<code>0x003f</code>
Color-ID (see Device Color Table)	4	UINT8	<code>0x..</code>
Device Capabilities	5	UINT8	<code>0x00</code>
Power state	6	UINT8	<code>0x00</code>

For example the complete BLE advertisement data will look like this (with a `0x00` Color ID and no capabilities or defined power state):

Length	Data Type	Company ID	Manufacturer Specific Data
<code>0x08</code>	<code>0xff</code>	<code>0x0ad2</code>	<code>0x003f000000</code>

Table 1. Device Color Table

Color ID	Decimal value	Device Color
<code>0x13</code>	19	Warm Black
<code>0x14</code>	20	Warm White
<code>0x15</code>	21	Light Mint
<code>0x16</code>	22	Wild Berry

2.1.4. BLE GATT Command Packet Framing

```
|| Vendor Id | Command Id | Payload ...||
```

- **Vendor Id:** This 16-bit field qualifies the command ID. Only commands with the Teufel Vendor Id `0x2cc2` are accepted.
- **Command Id:** This 16-bit field identifies the individual command.
- **Payload:** The payload contains any information required to be passed by a specific command. It consists of zero or more octets depending on the command.

2.2. RFComm / Bluetooth Classic Support

2.2.1. General Requirements

- The App Protocol interface is usable via RFComm over Bluetooth Classic (BR/EDR) (This is used by Android, iOS uses BLE).
- The App Protocol interface on RFComm is usable via the default SPP service UUID: `00001101-0000-1000-8000-00805f9b34fb`

2.2.2. RFComm Packet Framing

```
|| Start | Version | Flags | Length | Vendor Id | Command Id | Payload ... | Check ||
```

- **Start:** One octet with the fixed value `0xff`.
- **Version:** One octet. This field indicates the protocol version in use, currently `1`.
- **Flags:** One octet. Bits within this field control protocol options.
 - Bit[0]: If set, a single octet check is present
 - Bit[1:7]: Reserved, must be `0`
- **Vendor Id:** This 16-bit field qualifies the command ID. Only commands with the Teufel Vendor Id `0x2cc2`
- **Command Id:** This 16-bit field identifies the individual command.
- **Payload:** The payload contains any information required to be passed by a specific command. It consists of zero or more octets depending on the command.
- **Check:** One octet. If present this field is determined by XORing together the other octets in the packet.

2.3. Acknowledgement (ACK)

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

Every command must be acknowledged.

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

The first octet of the payload of an acknowledgement holds a status value. This value indicates the success or failure of the command. The remaining octets of the payload are command-specific.

The Success byte should come between the command ID and the data as shown in the tables above, see sections [RFComm Packet Framing](#) and [BLE GATT Command Packet Framing](#).

The byte should use the following codes to indicate the status of the packet.

Status	Cod e	Description
Success	0x00	The command completed successfully.
Failed: Command Not Supported	0x01	An invalid Command ID was specified.
Failed: Invalid Parameter	0x05	An invalid parameter was used in the command.
Failed: Incorrect/Error State	0x06	Device is not in the correct state to process the command or the command produced an error

Table 2. Example Success ACK response without payload (only success byte), typically a response for a set request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8828	0x00	0xb0

Table 3. Example Success ACK response with payload, typically a response for a get request

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x03	0x2cc2	0x8189	0x00	0x0384	0x9d

Table 4. Example Failure ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8189	0x05	0xa1

3. Commands

3.1. Commands I

3.1.1. Battery

Get Current Battery Level

- Command ID: 0x0302

Response: a two Bytes representing the battery level in percent:

- Response: 0x0053

Where 0053 is 83%.

Table 5. Example request (RFCOMM packet framing)

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0302	-	0x4e

Table 6. Example Success ACK response (RFCOMM packet framing)

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x03	0x2cc2	0x8302	0x00	0x0038	0xab

Table 7. Example Success ACK response (BLE packet framing)

Vendor Id	Command Id	Success Byte	Payload
0x2cc2	0x8302	0x00	0x0053

Note: This command must work when the battery is currently charging.

Get Power Adapter Connection Status

This command detects whether a power adapter is physically connected to the device.

It should **not** report if the battery is charging, but if a power adapter is connected or not.

- Command ID 0x0857

Returns a single Byte:

- 0x00: No power adapter connected
- 0x01: Power adapter connected

Table 8. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0857	-	0x4e

Table 9. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8857	0x00	0x00	0xcc

3.1.2. Auto-Off Timer

The Auto-Off Timer is a feature that allows the device to turn off automatically after a certain period of inactivity. Inactivity is defined as the time between the last user interaction or music playback and the present moment.

Please note that receiving commands from the companion app also counts as user interaction. Therefore, any command - regardless of whether it was sent via RFCOMM or BLE - should reset the

Auto-Off timeout.

Get Auto-Off Timer Value

Gets the active "auto-off" time.

- Command ID: **0x0189**

Returns 'auto-off time': 0 - 65535

Table 10. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0189	-	0x99

Table 11. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x03	0x2cc2	0x8189	0x00	0x0384	0x9d

Set Auto-Off Timer Value

- Command ID: **0x0109**

Sets the active "auto-off" time.

- Parameter: 'auto-off time': 0 - 65535

Must return the 'Invalid Parameter' status code **0x05** when the step is outside the configured number of steps.

Table 12. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x0109	0x0384	0x9c

Table 13. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8109	0x00	0x98

3.1.3. Device Color

Get Device Color

- Command ID **0x0900**

Returns a single Byte:

Table 14. Device Color Table

Color ID	Decimal value	Device Color
0x13	19	Warm Black
0x14	20	Warm White
0x15	21	Light Mint
0x16	22	Wild Berry

Table 15. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0900	-	0x18

Table 16. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8900	0x00	0x13	0x89

3.1.4. ECO-Mode

Enable ECO-Mode

- Command ID: 0x0830

Enables the "Eco-Mode" feature. No additional parameters and return values.

Table 17. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0830	-	0x29

Table 18. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8830	0x00	0xa8

Disable ECO-Mode

- Command ID: 0x0831

Disables the "Eco-Mode" feature. No additional parameters and return values.

Table 19. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0831	-	0x28

Table 20. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
-------	---------	-------	--------	-----------	------------	---------	-------

0xff	0x01	0x01	0x01	0x2cc2	0x8831	0x00	0xa9
------	------	------	------	--------	--------	------	------

Get ECO-Mode Status

- Command ID: 0x0832

Returns the current state of the "Eco-Mode" feature.

Returns 'state':

- 0x00: DISABLED
- 0x01: ENABLED

Table 21. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0832	-	0x2b

Table 22. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8832	0x00	0x01	0x9a

3.1.5. Equalizer

Get EQ Parameter Gain

- Command ID: 0x090d

Parameter: a single byte representing which parameter must be retrieved.

Returns the gain value multiplied by 10 as a single byte signed int. Example: -2dB = -20 = 0xec

The value range is -6dB....+6dB in 1dB steps, intermediate steps are not supported.

- 0x00: Bass
- 0x01: Treble

Example:

A payload of 0x00 means Get Bass Gain

Response: 0xec means -20 -2dB (2's complement)

Table 23. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x090d	0x01	0x15

Table 24. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
-------	---------	-------	--------	-----------	------------	--------------	---------	-------

0xff	0x01	0x01	0x02	0x2cc2	0x890d	0x00	0x03	0x94
------	------	------	------	--------	--------	------	------	------

Set EQ Parameter Gain

- Command ID: **0x090f**

Two parameters:

- Parameter 1: a single byte representing which band parameter should be set. See [Get EQ Parameter Gain](#)
- Parameter 2: one byte signed int that represent a gain value multiplied by 10.

Example: A payload of **0x001e** means Set Bass Gain to 3dB (30=**0x1e**)

Table 25. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x090f	0x011e	0x0b

Table 26. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x890f	0x00	0x96

3.1.6. Master Volume

Get Master Volume

- Command ID: **0x0866**

Returns the current master volume as a single byte. Volume values of 0%...100% are mapped to byte values of 0x00...0x64. Example: **0x4b** decodes to 75%.

Table 27. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0866	-	0x7f

Table 28. Example Failure ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8866	0x01	0xff

Set Master Volume

- Command ID: **0x0867**

Parameter a single byte where volume values of 0%...100% are mapped to byte values of 0x00...0x64. Example: 75% would be encoded as **0x4b**.

Table 29. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x0867	0x49	0x36

Table 30. Example Failure ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8867	0x01	0xfe

3.1.7. Master Mute

Master Mute Requirements

- The device must save the current volume level when being muted.
- The device must restore the previously saved volume level when being unmuted.

Pressing a hardware volume button in the muted state must work like that:

Current state

- The device is muted
- A hardware volume button (+ or -) at the device is pressed

Expected behavior

- The device is unmuted
- The previous volume level is restored
- Because the (+) or (-) button was pressed, the increase / decrease of the volume is applied
- A volume change notification (0x85) is sent to the app

Please note:

- The mute state and volume level are independent from each other.
- An unmuted state and a volume level of 0% are valid at the same time.

Get Master Mute Status

- Command ID: 0x0876

Returns the mute status of the device as a single byte.

- 0x00: not muted
- 0x01: muted

Table 31. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Check
-------	---------	-------	--------	-----------	------------	-------

0xff	0x01	0x01	0x01	0x2cc2	0x0876	0x??
------	------	------	------	--------	--------	------

Table 32. Example ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8876	0x00	0x01	0x??

The response **0x01** indicates that the device is currently muted.

Set Master Mute Status

- Command ID: **0x0877**

Parameter is a single byte:

- **0x00**: not muted
- **0x01**: muted

Table 33. Example mute request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x0877	0x01	0x??

Table 34. Example mute ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8877	0x00	0x??

3.1.8. MCU Firmware

Get MCU Firmware Version

- Command ID: **0x0856**

Parameter 1: 'target' (1 byte). Where target is one of

- **0x00**: DSP
- **0x01**: MCU

Returns 3 bytes representing the firmware version for the specified target as [semantic version](#) with the mapping:

Offset	Description
0	Major
1	Minor
2	Patch

Press the power & Bluetooth buttons to announce the firmware version. The music stops and a

voice reads out the current installed firmware version number.

Table 35. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x0856	0x01	0x4f

Table 36. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x04	0x2cc2	0x8856	0x00	0x001500	0xde

3.1.9. PartyLink Broadcast

Start PartyLink Broadcast

- Command ID: 0x0901

Starts the PartyLink broadcast.

Table 37. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0901	-	0x19

Table 38. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8901	0x00	0x98

Stop PartyLink Broadcast

- Command ID: 0x0902

Stops the PartyLink broadcast.

Table 39. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0902	-	0x1a

Table 40. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8902	0x00	0x9b

Get PartyLink Broadcast Status

- Command ID: 0x0903

Returns the current state of the Partylink broadcast as single byte.

Returns:

- **0x00**: INACTIVE
- **0x01**: ACTIVE

Table 41. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0903	-	0x1b

Table 42. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8903	0x00	0x01	0x98

3.1.10. Sound Icons

Disable Sound Icons

- Command ID: **0x0828**

Disable system sounds. No additional parameters and return values.

Table 43. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0828	-	0x31

Table 44. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8828	0x00	0xb0

Enable Sound Icons

- Command ID: **0x0829**

Enable system sounds. No additional parameters and return values.

Table 45. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0829	-	0x30

Table 46. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8829	0x00	0xb1

Get Sound Icons Status

- Command ID: **0x082a**

Returns the current state of the Sound Icons feature Returns 'state':

- **0x00**: DISABLED - Fewer or No Sound Icons Heard
- **0x01**: ENABLED - All Sound Icons Heard

Table 47. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x082a	-	0x33

Table 48. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x882a	0x00	0x00	0xb1

3.1.11. Source Identifiers

Each source has a unique identifier that is used in multiple commands.

Table 49. MYND
source identifiers

ID	Source
0x00	Bluetooth
0x01	AUX
0x02	USB

3.1.12. Bluetooth Firmware

Get Bluetooth Firmware Version

- Command ID: **0x0304**

Response: an ASCII string byte array with the version as [semantic version](#). The firmware version must follow the semantic version schema: **MAJOR.MINOR.PATCH**.

Example: The byte string **0x312E322E34** resolves to **1.2.4**.

Press the power & Bluetooth buttons to announce the firmware version. The music stops and a voice reads out the current installed firmware version number.

Table 50. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0304	-	0x16

Table 51. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8304	0x00	0x312E322E34	0xa7

3.2. Commands II

3.2.1. Battery Capacity

Get Battery Capacity

- Command ID: 0x0913

Returns the two unsigned shorts concatenated:

- first two bytes is the current capacity in mAh
- second two bytes is the maximal capacity in mAh (of a new battery)

Example Response Payload: 0x0BB81388

- 0x0BB8: 3000 mAh
- 0x1388: 5000 mAh

Table 52. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0913	-	0x??

Table 53. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x03	0x2cc2	0x8913	0x00	0x0bb8	0x??

3.2.2. Battery Friendly Charging

Enable Battery Friendly Charging

- Command ID: 0x0910

Enables the "Battery Friendly Charging" feature. No additional parameters and return values.

When the device is being charged and the power button is double pressed (pressed twice within 500ms), it will trigger "Battery Friendly Charging". The power LED will blink twice.

Table 54. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0910	-	0x08

Table 55. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8910	0x00	0x89

Disable Battery Friendly Charging

- Command ID: 0x0911

Disables the "Battery Friendly Charging" feature. No additional parameters and return values.

Table 56. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0911	-	0x09

Table 57. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8911	0x00	0x88

Get Battery Friendly Charging Status

- Command ID: 0x0912

Returns the current state of the "Battery Friendly Charging" feature as a single byte.

Returns 'state':

- 0x00: DISABLED
- 0x01: ENABLED

Table 58. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0912	-	0x0a

Table 59. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8912	0x00	0x01	0x8a

Get Battery Capacity

- Command ID: 0x0913

Returns the two unsigned shorts concatenated:

- first two bytes is the current capacity in mAh
- second two bytes is the maximal capacity in mAh (of a new battery)

Example Response Payload: 0x0BB81388

- **0x0BB8**: 3000 mAh
- **0x1388**: 5000 mAh

Table 60. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0913	-	0x0b

Table 61. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x05	0x2cc2	0x8913	0x00	0x13241324	0x8e

3.2.3. Multipoint

Enable Multipoint

- Command ID: **0x084c**

Enables the "Multipoint" feature. No additional parameters and return values.

Table 62. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x084c	-	0x55

Table 63. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x884c	0x00	0xd4

Disable Multipoint

- Command ID: **0x084d**

Disables the "Multipoint" feature. No additional parameters and return values.

Table 64. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x084d	-	0x54

Table 65. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
-------	---------	-------	--------	-----------	------------	---------	-------

0xff	0x01	0x01	0x01	0x2cc2	0x884d	0x00	0xd5
------	------	------	------	--------	--------	------	------

Get Multipoint Status

- Command ID: 0x084e

Returns the current state of the "Multipoint" feature.

Returns 'state':

- 0x00: DISABLED
- 0x01: ENABLED

Table 66. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x084e	-	0x57

Table 67. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x884e	0x00	0x01	0xd4

3.2.4. Source Selection

Get Source

- Command ID: 0x0880

Parameter 1: 'channel' (1 byte):

- 0x00: channel 1 (default for single channel devices)

Returns the [Source Identifiers](#) as a single Byte.

Table 68. Example request

Vendor Id	Command Id	Payload
0x2cc2	0x0880	0x00

Table 69. Example Success ACK response

Vendor Id	Command Id	Success Byte	Payload
0x2cc2	0x8880	0x00	0x01

In this example the source with ID 0x01 is currently selected (see [Source Identifiers](#)).

Set Source

- Command ID: 0x0881

The source can be switched amongst Bluetooth, USB-C and AUX by pressing the Bluetooth button. The source selection will cycle through but USB-C and AUX will be skipped if no cable nor audio signal can be detected.

Parameter 1: 'channel' (1 byte):

- **0x00**: channel 1 (default for single channel devices)

Parameter 2: 'source' (1 byte). For mapping see [Get Source](#).

Table 70. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x03	0x2cc2	0x0881	0x0001	0x9b

Table 71. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8881	0x0001	0x1b

Get Connected Sources

- Command ID: **0x0882**

Returns a single byte representing a bitmask with the following mapping:

- **00000001** - **0x01**: Bluetooth
- **00000010** - **0x02**: USB
- **00000100** - **0x04**: AUX

If multiple sources are connected, then individual source values must be combined through the use of a logical OR operation, i.e.:

- **00000000** - **0x00**: None
- **00000011** - **0x03**: Bluetooth + USB
- **00000101** - **0x05**: Bluetooth + AUX
- **00000110** - **0x06**: USB + AUX
- **00000111** - **0x07**: Bluetooth + AUX + USB

Table 72. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0882	-	0x9b

Table 73. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8882	0x00	0x03	0x1a

3.3. Commands III

3.3.1. LED Brightness

Get LED Brightness

- Command ID: **0x0840**

Gets the brightness of the device's LEDs.

Returns a byte:

- **0x00...0x64**: which is mapped to the range 0..100%

Example: **0x64** = 100 = 100%, **0x4b** = 75 = 75%

Table 74. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x00	0x2cc2	0x0840	-	0x59

Table 75. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Success Byte	Payload	Check
0xff	0x01	0x01	0x02	0x2cc2	0x8840	0x00	0x4b	0x90

Set LED Brightness

- Command ID: **0x0841**

Sets the brightness of the device's LEDs.

Parameter: brightness value (1 Byte), see [Get LED Brightness](#).

Table 76. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x0841	0x4b	0x12

Table 77. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x8841	0x00	0xd9

4. Notifications

The Device can send notifications to the Host. To reduce radio traffic (caused by frequent polling) and increases battery life, we supports event notifications.

Each supported event is given a numeric code. The Host always acknowledges receipt, when

notified by the Device.

The notifications should be sent only if the Host registered to receive them.

4.1. Register notification

This command **0x4001** is used to register for notifications that inform the Host, when a specified event occurs on the Device. The command payload defines which notification to register to using an Id. See: [Notifications list](#).

Offset	Value	Description
0	...	Notification Id

Table 78. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x4001	0x09	0x58

Table 79. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0xc001	0x00	0xd1

4.2. Cancel notification

This command **0x4002** stops notifications of the given type. See: [Notifications list](#).

Offset	Value	Description
0	...	Notification Id

4.3. Event notification

Notifications are sent from the device with the command id **0x4003**. Notifications should not be sent during OTA update. The payload contains the notification Id and optional additional data depending on the notification.

Offset	Value	Description
0	...	Notification Id
1	...	Notification-specific data, see the individual notification (OPTIONAL)

Table 80. Example request

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0x4003	0x09	0x5b

Table 81. Example Success ACK response

Start	Version	Flags	Length	Vendor Id	Command Id	Payload	Check
0xff	0x01	0x01	0x01	0x2cc2	0xc003	0x00	0xd2

4.4. Notifications list

Notification ID	Response Payload	Description
0x03	Battery Low Threshold	It is sent when battery reaches the low threshold.
0x09	Power Adapter Connection Status 0x01: power adapter connected 0x00: No power adapter connected	It is sent when the power adapter is (un)plugged. See Get Power Adapter Connection Status .
0x85	See Get Master Volume command	Volume change (master)
0x86	Mute status: 0x00: not muted 0x01: muted	
0x87	0x00: disconnected 0x01: connected	USB connection status. This notification is deprecated. It should be replaced with 0x97 notification.
0x89	0x00: Eco mode off 0x01: Eco mode on	Eco mode status
0x8a	Battery level as unsigned byte with value from 0 to 100	it is sent when the value changes
0x92	Party link Broadcast status 0x00: inactive 0x01: active	It is sent when the party link broadcast status changes
0x97	Returns a 2 bytes bitmask. Every 1 in the bitmask means that the source is active. Every mapping is device dependent, the mapping is the same as used in Get Connected Sources Example: 0x0005 → 00000000 00000101 → The first and the third source (bit's 1 and 3) are connected, the second source (bit 2) is not connected.	Plug detection (line, instrument ect).
0x98	0x00: Battery Friendly Charging off 0x01: Battery Friendly Charging on	It is sent when Battery Friendly Charging status changes