Produciton Test Design Document

|  |  |
| --- | --- |
| Owner: | SW Engineering |
| Scope: | Software Component Design Document |
| Originator: | Daniel Qin |
| Status: | Draft |
| Version: | V1.0 |
| Document ID: |  |
| Location: | http://flow.tymphany.com/redmine/projects/portable-1/dmsf?folder\_id=21363 |

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Handled by** | **Comments** |
| 0.1 | 16-Jan-2018 | Benjamin.Tan | Initial version |
| 0.2 | 16-Jan-2018 | Benjamin.Tan | Add bt cmd |
| 1.0 | 13-Apr-2018 | Benjamin.Tan | Added DSP cmd  Added Factory Reset cmd |
| 1.1 | 11-May-2018 | Benjamin.Tan | Added Enter POS mode cmd |
| 1.2 | 14-May-2018 | Benjamin.Tan | Changed Factory Reset Cmd |
| 1.3 | 16-July-2018 | Benjamin.Tan | Add Reset battery display cmd |
| 1.4 | 27-Nov-2018 | Benjamin.Tan | Add ntc test method |
|  |  |  |  |

Contents

[1. Component Overview & Summary 4](#_Toc531096850)

[2. test command 5](#_Toc531096851)

[2.1 Version Read 5](#_Toc531096852)

[2.2 Read Volume&Treble&Bass Knob Value 6](#_Toc531096853)

[2.3 LED Verification 7](#_Toc531096854)

[2.4 Set Volume & Treble & Bass 8](#_Toc531096855)

[2.5 Set Audio Channel 9](#_Toc531096856)

[2.6 Set Audio EQ Bypass 9](#_Toc531096857)

[2.7 Set AMP MUTE/UNMUTE 10](#_Toc531096858)

[2.8 Set DSP MUTE/UNMUTE 11](#_Toc531096859)

[2.9 Get Battery Capacity 12](#_Toc531096860)

[2.10 Get DC in status 13](#_Toc531096861)

[2.11 Set DUT to Standby 13](#_Toc531096862)

[2.12 Set DUT to Active 14](#_Toc531096863)

[2.13 Set DUT to Sleep 14](#_Toc531096864)

[2.14 Set DUT to FactoryReset 14](#_Toc531096865)

[2.15 Set DUT to Shutdown 14](#_Toc531096866)

[2.16 Send BT Command 15](#_Toc531096867)

[2.17 Set to POS & LS mode 16](#_Toc531096868)

[2.18 Set Displayed battery capacity to actual battery capacity 17](#_Toc531096869)

[2.19 NTC test commands 17](#_Toc531096870)

# Component Overview & Summary

This is a document that describes production test cases, tools and methods on how to perform them effectively.

Please be noted that test command should sent at System Active Status.

# test command

## Version Read

1. To read **Software Version** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_SW\_VER**

Offset: **0**

Size: **8**

Press “**Send**”

MSG\_EXAMPLE:

[17:22:39:880]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xA 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0x86 0x5

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:22:39:891]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0xA 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0x32 0x2E 0x30 0x2E 0x30 0x2E 0x30 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x3F 0x73

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_SW\_VER ; offset : 0x0 ; size : 0x8 ; data :

String: 2.0.0.0\_ ;

Raw bstream(hex): 0x32-0x2E-0x30-0x2E-0x30-0x2E-0x30-0x0 ;

Raw bstream(dec): 50-46-48-46-48-46-48-0 ;

Software version: **2.0.0.0**

1. To read **BT Software Version** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_BT\_VER\_STR**

Offset: **0**

Size: **5**

Press “**Send**”

MSG\_EXAMPLE:

[18:33:25:115]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x6D 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x11 0x0 0x0 0x0 0x0 0x0 0x5 0x0 0x78 0x25

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[18:33:25:143]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x31 0x0 0x57 0x0 0x31 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x11 0x0 0x0 0x0 0x0 0x0 0x5 0x0 0x35 0x2E 0x30 0x2E 0x30 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x7C 0xAE

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_BT\_VER\_STR ; offset : 0x0 ; size : 0x5 ; data :

String: 5.0.0 ;

Raw bstream(hex): 0x35-0x2E-0x30-0x2E-0x30 ;

Raw bstream(dec): 53-46-48-46-48 ;

)

1. To read **Hardware Version** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_HW\_VER**

Offset: **0**

Size: **4**

Press “**Send**”

MSG\_EXAMPLE:

[17:24:36:333]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xB 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x38 0x7

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:24:36:340]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0xB 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x45 0x56 0x54 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x68 0xF5

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_HW\_VER ; offset : 0x0 ; size : 0x4 ; data :

String: EVT\_ ;

Raw bstream(hex): 0x45-0x56-0x54-0x0 ;

Raw bstream(dec): 69-86-84-0 ;

Hardware version: **EVT**

1. To read **DSP Version** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_DSP\_VER**

Offset: **0**

Size: **4**

Press “**Send**”

MSG\_EXAMPLE:

[17:25:18:917]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x15 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x9D 0xA4

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:25:18:925]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x15 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x30 0x2E 0x33 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x6C 0x76

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_DSP\_VER ; offset : 0x0 ; size : 0x4 ; data :

String: 0.3\_ ;

Raw bstream(hex): 0x30-0x2E-0x33-0x0 ;

Raw bstream(dec): 48-46-51-0 ;

DSP version: **0.3**

## Read Volume&Treble&Bass Knob Value

1. To read **Volume** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_VOLUME**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[17:41:19:757]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x62 0x2E

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:41:56:502]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xA 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xF5 0x53

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_VOLUME ; offset : 0x0 ; size : 0x1 ; data :

String:

Raw bstream(hex): 0xA ;

Raw bstream(dec): 10

Volume: **10**

1. To read **Volume** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_TREBLE**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[17:43:02:945]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xB1 0x69

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:43:02:954]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x2 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x73 0x75

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_TREBLE ; offset : 0x0 ; size : 0x1 ; data :

String: ;

Raw bstream(hex): 0x2 ;

Raw bstream(dec): 2 ;

Treble: **2**

1. To read **Volume** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_Bass**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[17:44:21:121]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xC4 0xA1

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[17:44:21:128]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x4 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xCD 0xEC

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_BASS ; offset : 0x0 ; size : 0x1 ; data :

String: ;

Raw bstream(hex): 0x4 ;

Raw bstream(dec): 4 ;

Bass: **4**

## LED Verification

1. To set all **LED ON,** use **LED\_REQ\_SIG,** set

Led\_cmd: **LED\_PAT\_ON\_CMD**

Led\_mask: **MASK\_ALL\_LED**

Led\_pattery:LED\_ON

Press “**Send**”

MSG\_EXAMPLE:

[17:52:07:939]: sent>>>>>>>> LED\_REQ\_SIG:0xAA 0x37 0x8 0x1B 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0xFF 0x1F 0x0 0x0 0x1 0x0 0x0 0x0 0xEB 0x3E

There is no response except all led lighted up.

1. To set all **LED OFF,** use **LED\_REQ\_SIG,** set

Led\_cmd: **LED\_PAT\_ON\_CMD**

Led\_mask: **MASK\_ALL\_LED**

Led\_pattery:LED\_OFF

Press “**Send**”

MSG\_EXAMPLE:

[17:52:59:846]: sent>>>>>>>> LED\_REQ\_SIG:0xAA 0x37 0x8 0x1B 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0xFF 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x5F 0x48

## Set Volume & Treble & Bass

1. To set all **Volume,** use **MAINAPP\_SET\_AUDIO\_SIG,** set

Set\_ID: **DSP\_VOLUME\_SETT\_ID**

Enable/Disable: **TRUE**

Value: Volume

Not\_used: 0

Press “**Send**”

MSG\_EXAMPLE: set volume to 32

[15:15:38:586]: sent>>>>>>>> MAINAPP\_SET\_AUDIO\_SIG:0xAA 0x8E 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x20 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xFD 0x2C

Volume range : 0 – 32

1. To set all **Volume,** use **AUDIO\_SET\_SIG,** set

Set\_ID: AUDIO\_**DSP\_TREBLE\_SETT\_ID**

Enable/Disable: **TRUE**

Value: Treble

Not\_used: 0

Press “**Send**”

MSG\_EXAMPLE: set volume to 10

[15:13:50:969]: sent>>>>>>>> AUDIO\_SET\_SIG:0xAA 0x58 0x7 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xE 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xA 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x86 0x84

Treble Range: 0 - 10

1. To set all **Volume,** use **AUDIO\_SET\_SIG,** set

Set\_ID: AUDIO\_**DSP\_BASS\_SETT\_ID**

Enable/Disable: **TRUE**

Value: Bass

Not\_used: 0

Press “**Send**”

MSG\_EXAMPLE: set volume to 10

[15:14:55:466]: sent>>>>>>>> AUDIO\_SET\_SIG:0xAA 0x58 0x7 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xD 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xA 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xCB 0x6C

Bass range : 0 – 10

## Set Audio Channel

1. To set all **Volume,** use **MAINAPP\_SWITCH\_CHANNEL\_SIG,** set

Set\_ID: **DSP\_VOLUME\_SETT\_ID**

Enable/Disable: **TRUE**

Value: Volume

Not\_used: 0

Press “**Send**”

MSG\_EXAMPLE: set channel to bt

[08:39:34:501]: sent>>>>>>>> AUDIO\_SET\_SIG:0xAA 0x56 0x7 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x6 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xF6 0xAB

MSG\_EXAMPLE: set channel to aux

[08:47:42:741]: sent>>>>>>>> AUDIO\_SWITCH\_CHANNEL\_SIG:0xAA 0x55 0x7 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x0 0x0 0x38 0x14

There is no response except volume of DUT changed.

## Set Audio EQ Bypass

1. To set all **Bypass EQ ,** use **MAINAPP\_SET\_AUDIO\_SIG,** set

Set\_ID: **DSP\_PASSTHROUGH\_SETT\_ID**

Enable/Disable: **TRUE or FALSE**

Value: 0

Not\_used: 0

Press “**Send**”

MSG\_EXAMPLE: enable eq bypass:

[16:55:47:888]: sent>>>>>>>> MAINAPP\_SET\_AUDIO\_SIG:0xAA 0x88 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x13 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xAA 0x47

As response **MAINAPP\_SET\_AUDIO\_SIG** will be received

[16:55:47:895]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x1A 0x0 0x0 0x0 0x6 0x2 0x15 0xA 0x76 0x69

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

MSG\_EXAMPLE: disable eq bypass

[16:57:09:254]: sent>>>>>>>> MAINAPP\_SET\_AUDIO\_SIG:0xAA 0x88 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xDF 0x44

As response **MAINAPP\_SET\_AUDIO\_SIG** will be received

[16:57:09:263]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x1A 0x0 0x0 0x0 0x6 0x2 0x15 0xA 0x76 0x69

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## Set AMP MUTE/UNMUTE

1. To set amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_AMP\_SOFT\_MUTE/AUDIO\_AMP\_MUTE**

Press “**Send**”

MSG\_EXAMPLE: set amp mute

[08:44:46:634]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x54 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x4 0x0 0x0 0x0 0x9 0xFA

MSG\_EXAMPLE: set amp unmute

[08:48:32:131]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x54 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x0 0x0 0xDA 0xBD

As response **AUDIO\_MUTE\_SIG** will be received

[16:57:54:021]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xF4 0x12 0x0 0x20 0x5 0x0 0x5B 0x7 0xF1 0x46

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

1. To set tweeter amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_AMP\_SOFT\_MUTE\_TW\_1/ AUDIO\_AMP\_SOFT\_MUTE\_TW\_2**

Press “**Send**”

MSG\_EXAMPLE: set amp mute

[14:14:26:882]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x54 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x6 0x0 0x0 0x0 0x61 0x17

As response **AUDIO\_MUTE\_SIG** will be received

[14:15:05:257]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xF4 0x12 0x0 0x20 0x6 0x0 0x0 0x0 0x8F 0x7F

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

1. To set tweeter amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_AMP\_SOFT\_MUTE\_WF**

Press “**Send**”

MSG\_EXAMPLE: set amp mute

[14:16:00:488]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x54 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x8 0x0 0x0 0x0 0x3B 0xB5

As response **AUDIO\_MUTE\_SIG** will be received

[14:15:05:257]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xF4 0x12 0x0 0x20 0x6 0x0 0x0 0x0 0x8F 0x7F

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## Set DSP MUTE/UNMUTE

1. To set amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_SOURCE\_MUTE**

Press “**Send**”

MSG\_EXAMPLE: set dsp mute

[14:09:31:920]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x9E 0x5B

MSG\_EXAMPLE: set dsp unmute

[14:09:34:787]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x4D 0x1C

As response **AUDIO\_MUTE\_SIG** will be received

[[14:10:16:167]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x74 0x13 0x0 0x20 0x0 0x0 0x0 0x0 0xF6 0x1D

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

1. To set tweeter amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_DSP\_MUTE\_TW\_1/ AUDIO\_AMP\_MUTE\_TW\_2**

Press “**Send**”

MSG\_EXAMPLE: set dsp tw mute

[14:12:40:789]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x11 0x0 0x0 0x0 0x8D 0x36

MSG\_EXAMPLE: set dsp tw unmute

[14:15:06:532]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x12 0x0 0x0 0x0 0x82 0xEA

As response **AUDIO\_MUTE\_SIG** will be received

[14:12:40:798]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x74 0x13 0x0 0x20 0x11 0x0 0x0 0x0 0xE5 0x70

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

1. To set woofer amp MUTE/UNMUTE **,** use **AUDIO\_MUTE\_SIG,** set

mute: **MUTE/UNMTE**

audioMuteType:**AUDIO\_DSP\_MUTE\_WF**

Press “**Send**”

MSG\_EXAMPLE: set dsp wf mute

[14:11:04:074]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xF 0x0 0x0 0x0 0x70 0x8F

MSG\_EXAMPLE: set dsp wf unmute

[14:11:25:744]: sent>>>>>>>> AUDIO\_MUTE\_SIG:0xAA 0x56 0x7 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xF 0x0 0x0 0x0 0xA3 0xC8

As response **AUDIO\_MUTE\_SIG** will be received

[14:11:25:752]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x74 0x13 0x0 0x20 0xF 0x0 0x0 0x0 0x18 0xC9

-> AUDIO\_MUTE\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## Get Battery Capacity

1. To read display **battery capacity** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_BATTERY\_CAPACITY**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[09:03:47:032]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x6 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xA9 0xAE

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[09:04:28:189]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x6 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x64 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xFD 0xFA

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_BATTERY\_CAPACITY ; offset : 0x0 ; size : 0x1 ; data :

String: d ;

Raw bstream(hex): 0x64 ;

Raw bstream(dec): 100 ;

1. To read actual **battery capacity** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_ADC\_CAPACITY**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[15:29:41:641]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x6A 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x3 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xB7 0xD3

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[15:30:15:021]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2E 0x0 0x27 0x0 0x2E 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x3 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x3D 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x78 0x97

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_ADC\_CAPACITY ; offset : 0x0 ; size : 0x1 ; data :

String: = ;

Raw bstream(hex): 0x3D ;

Raw bstream(dec): 61 ;

Number(hex): 0x3D ;

Number(dec): 61 ;

)

## Get DC in status

1. To read **Volume** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_IS\_DC\_PLUG\_IN**

Offset: **0**

Size: **1**

Press “**Send**”

MSG\_EXAMPLE:

[09:05:58:021]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x11 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x5 0x54

As response SETTING\_READ\_OFFSET\_RESP\_SIG will be received

Example:

[09:05:58:030]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x27 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x11 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x6D 0xDD

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_IS\_DC\_PLUG\_IN ; offset : 0x0 ; size : 0x1 ; data :

String: ;

Raw bstream(hex): 0x1 ;

Raw bstream(dec): 1 ;

## Set DUT to Standby

1. To set DUT **Standby** use **MAINAPP\_STANDBY\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[09:08:05:297]: sent>>>>>>>> MAINAPP\_STANDBY\_SIG:0xAA 0x81 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0xB4 0x33

As response **MAINAPP\_STANDBY\_SIG** will be received

Example:

[14:16:51:101]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xF4 0x12 0x0 0x20 0x6 0x0 0x0 0x0 0xDE 0xEB

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## Set DUT to Active

1. To set DUT **Active** use **MAINAPP\_ACTIVE\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[09:11:29:146]: sent>>>>>>>> MAINAPP\_ACTIVE\_SIG:0xAA 0x80 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0x67 0x74

As response **MAINAPP\_STANDBY\_SIG** will be received

Example:

[14:17:48:020]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0xB6 0xE 0x5A 0xF2

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## Set DUT to Sleep

1. To set DUT **Sleep** use **MAINAPP\_SLEEP\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[09:12:16:598]: sent>>>>>>>> MAINAPP\_SLEEP\_SIG:0xAA 0x7F 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0x28 0xE0

press bt button & plug dc in & turn volume knob to on can wake system up

## Set DUT to FactoryReset

1. If DUT is not stay at playback mode then can skip first step.
2. To set DUT **FactoryReset** use **BT\_REQ\_SIG** First, set

Signal name : BT\_REQ\_SIG;

4 byte ENUM: BT\_ENTER\_PAIRING\_CMD

Press “**Send**”

MSG\_EXAMPLE:

[14:03:50:041]: sent>>>>>>>> BT\_REQ\_SIG:0xAA 0x5F 0xB 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xB 0x0 0x0 0x0 0xCD 0x2D

BT will disconnect all linked connection. THEN delay 500ms later send second command.

1. To set DUT **FactoryReset** use **MAINAPP\_FACTORY\_RESET\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[14:17:58:482]: sent>>>>>>>> MAINAPP\_FACTORY\_RESET\_SIG:0xAA 0x8C 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0xD0 0x65

BT led will shown purple fading pattern for about 5s then DUT will restart.

## Set DUT to Shutdown

1. To set DUT **Shutdown** use **MAINAPP\_SHUTDOWN\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[09:13:57:646]: sent>>>>>>>> MAINAPP\_SHUTDOWN\_SIG:0xAA 0x84 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0xA 0x7B

But if power key stay at on status, DUT will turn on again.

## Send BT Command

1. To set **BT Module** into Pairing Mode use **BT\_REQ\_SIG** set

Setting\_id: **BT\_ENTER\_PAIRING\_CMD**

Press “**Send**”

MSG\_EXAMPLE:

[13:23:46:492]: sent>>>>>>>> BT\_REQ\_SIG:0xAA 0x5D 0xB 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xBB 0x73

As response **BT\_STATE\_SIG** will be received

Example:

[13:30:17:274]: recv<<<<<<<< BT\_STATE\_SIG:0xAA 0x16 0x0 0x17 0x0 0x16 0x0 0x2 0x1 0x1 0x0 0x0 0x0 0x24 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xEB 0xCA

-> BT\_STATE\_SIG ( btStatus : BT\_DISCOVERABLE\_STA ; )

1. To set **BT Module** into Connectable Mode use **BT\_REQ\_SIG** set

Setting\_id: **BT\_ENTER\_CONNECTABLE\_CMD**

Press “**Send**”

MSG\_EXAMPLE:

[13:30:53:474]: sent>>>>>>>> BT\_REQ\_SIG:0xAA 0x5D 0xB 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xF 0x5

As response **BT\_STATE\_SIG** will be received

Example:

[13:32:49:598]: recv<<<<<<<< BT\_STATE\_SIG:0xAA 0x16 0x0 0x17 0x0 0x16 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x24 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x9E 0xC9

-> BT\_STATE\_SIG ( btStatus : BT\_CONNECTABLE\_STA ; )

1. To set **BT Module** to Play test Cue use **BT\_REQ\_SIG** set

Setting\_id: **BT\_TEST\_CUE\_CMD**

Press “**Send**”

MSG\_EXAMPLE:

[14:21:46:817]: sent>>>>>>>> BT\_REQ\_SIG:0xAA 0x5D 0xB 0x13 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1B 0x0 0x0 0x0 0x3 0x76

As response **BT\_STATE\_SIG** will be received & BT channel will comes out 1khz sine tone

Example:

[14:21:51:379]: recv<<<<<<<< BT\_STATE\_SIG:0xAA 0x16 0x0 0x17 0x0 0x16 0x0 0x2 0x1 0x10 0x0 0x0 0x0 0x24 0x0 0x0 0x0 0x1 0x0 0xB6 0xE 0xAB 0xAB

-> BT\_STATE\_SIG ( btStatus : )

## Set to POS & LS mode

1. To set **Mode** use **SETTING\_WRITE\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_SHOP\_MODE\_WORD**

Offset: **0**

Size: **4**

Data: 0x5a,0x15,0x1e,0x5a (LS mode)

Data: 0x1e,0x5a,0x1e,0x5a (POS)

Press “**Send**”

MSG\_EXAMPLE:

[17:16:18:799]: sent>>>>>>>> SETTING\_WRITE\_OFFSET\_REQ\_SIG:0xAA 0x69 0xC 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x13 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x5A 0x15 0x1E 0x5A 0x0 0x0 0x0 0x0 0xB4 0x77

1. To active **Mode** use **MAINAPP\_FACTORY\_RESET\_SIG** set

Press “**Send**”

MSG\_EXAMPLE:

[17:19:35:215]: sent>>>>>>>> MAINAPP\_FACTORY\_RESET\_SIG:0xAA 0x8C 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0xD0 0x65

After system power up ,DUT will enter target mode

As response SETTING\_WRITE\_OFFSET\_RESP\_SIG will be received

Example:

[17:19:35:223]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x78 0xC 0x0 0x20 0x0 0x0 0x4 0x0 0x46 0xE9

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

1. To confirm **Mode** use **SETTING\_READ\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_SHOP\_MODE\_WORD**

Offset: **0**

Size: **4**

Press “**Send**”

MSG\_EXAMPLE:

[17:18:28:951]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x6A 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x13 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0xF6 0x11

As response SETTING\_WRITE\_OFFSET\_RESP\_SIG will be received

Example:

[17:18:28:959]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2E 0x0 0x27 0x0 0x2E 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x13 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x5A 0x15 0x1E 0x5A 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xD9 0xE5

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_SHOP\_MODE\_WORD ; offset : 0x0 ; size : 0x4 ; data :

String: Z‑Z ;

Raw bstream(hex): 0x5A-0x15-0x1E-0x5A ; (LS mode)

Raw bstream(dec): 90-21-30-90 ;

)

## Set Displayed battery capacity to actual battery capacity

1. Let DUT recalculate battery capacity by voltage value use **POWER\_SET\_SIG** set

Setting\_id: **POWER\_SET\_ID\_RESET\_CAPACITY**

Enable/Disable: **TRUE**

Press “**Send**”

MSG\_EXAMPLE:

[09:12:16:598]: [16:25:25:818]: sent>>>>>>>> POWER\_SET\_SIG:0xAA 0x4C 0x2 0x11 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x7 0x1 0xBE 0xA9

As response **POWER\_SET\_SIG** will be received

Example:

[16:26:07:867]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x7C 0xC 0x0 0x20 0x5 0x2 0x7C 0x7 0x19 0xC5

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

## NTC test commands

1. Let DUT start to NTC testing, using **SETTING\_WRITE\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_NTC\_TEST\_VALUE**

Offset: **0**

Size: **1**

Data: **0**

Press “**Send**”

MSG\_EXAMPLE:

[15:39:56:419]: sent>>>>>>>> SETTING\_WRITE\_OFFSET\_REQ\_SIG:0xAA 0x69 0xC 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1D 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x85 0xB9

As response **SETTING\_WRITE\_OFFSET\_REQ\_SIG** will be received

Example:

[16:26:07:867]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x7C 0xC 0x0 0x20 0x5 0x2 0x7C 0x7 0x19 0xC5

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

// Now the NTC value of DUT were set to 1600 as default value

1. Let NTC value of DUT decrease, using **SETTING\_WRITE\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_NTC\_TEST\_VALUE**

Offset: **0**

Size: **1**

Data: **1**

Press “**Send**”

MSG\_EXAMPLE:

[15:39:56:419]: sent>>>>>>>> SETTING\_WRITE\_OFFSET\_REQ\_SIG:0xAA 0x69 0xC 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1D 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x85 0xB9

As response **SETTING\_WRITE\_OFFSET\_REQ\_SIG** will be received

Example:

[16:26:07:867]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x7C 0xC 0x0 0x20 0x5 0x2 0x7C 0x7 0x19 0xC5

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

// NTC value will decrease 50 once the cmd received.

1. Let NTC value of DUT decrease, using **SETTING\_WRITE\_OFFSET\_REQ\_SIG** set

Setting\_id: **SETID\_NTC\_TEST\_VALUE**

Offset: **0**

Size: **1**

Data: **2**

Press “**Send**”

MSG\_EXAMPLE:

[15:39:56:419]: sent>>>>>>>> SETTING\_WRITE\_OFFSET\_REQ\_SIG:0xAA 0x69 0xC 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1D 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x85 0xB9

As response **SETTING\_WRITE\_OFFSET\_REQ\_SIG** will be received

Example:

[16:26:07:867]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x7C 0xC 0x0 0x20 0x5 0x2 0x7C 0x7 0x19 0xC5

-> DEBUG\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; )

// NTC value will increase 50 once the cmd received.

//Notice: once the DUT have the cold reset or Hardware reset or using Part a with data 0xFF to DUT, the testing method will be reset, so you have to send part a again to start NTC testing.