Produciton Test Design Document

|  |  |
| --- | --- |
| Owner: | SW Engineering |
| Scope: | Software Component Design Document |
| Originator: | Daniel Qin |
| Status: | Draft |
| Version: | V0.4 |
| Document ID: |  |
| Location: |  |

**Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Issue** | **Date** | **Handled by** | **Comments** |
| 0.1 | 05-Feb-2018 | Benjamin.Tan | Initial version |
| 0.2 | 26-Feb-2018 | Benjamin.Tan | Added bypass eq command |
| 0.3 | 1-Mar-2018 | Benjamin.Tan | 1. Added MAINAPP\_PTE\_TEST\_CMD\_SIG 2. Added volume & source setting cmd 3. Added Mute outpu channel SIG 4. Added BT address get cmd 5. Added Test Cue Cmd |
| 0.4 | 12-April-2018 | Daniel Qin | [*2.10*]Added command for switch system work mode with below parameter:  0: Normal Mode  1: Shop(POS) Mode |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1. Component Overview & Summary 4](#_Toc511313018)

[2. test command 5](#_Toc511313019)

[2.1 Version Read 5](#_Toc511313020)

[2.2 LED Verification 7](#_Toc511313021)

[2.3 Set Source 8](#_Toc511313022)

[2.4 Set Volume 9](#_Toc511313023)

[2.5 Mute DSP Output Channel 9](#_Toc511313024)

[2.6 Set DUT to Standby 10](#_Toc511313025)

[2.7 Set Factory Reset 10](#_Toc511313026)

[2.8 Set DSP Bypass EQ 10](#_Toc511313027)

[2.9 Play 1khz Test Cue 10](#_Toc511313028)

[2.10 Set System Work Mode 11](#_Toc511313029)

# 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\_STR**

Offset: **0**

Size: **8**

Press “**Send**”

MSG\_EXAMPLE:

[14:25:15:910]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xC 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0x4D 0x85

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

Example:

[14:25:15:924]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x57 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0xC 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0x31 0x2E 0x30 0x2E 0x33 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 0x59 0xC5

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

String: 1.0.3.0\_ ;

Raw bstream(hex): 0x31-0x2E-0x30-0x2E-0x33-0x2E-0x30-0x0 ;

Raw bstream(dec): 49-46-48-46-51-46-48-0 ;

)

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

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

Offset: **0**

Size: **8**

Press “**Send**”

MSG\_EXAMPLE:

[14:27:13:435]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x9 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0xF3 0xCD

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

Example:

[14:27:30:220]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x57 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x9 0x0 0x0 0x0 0x0 0x0 0x8 0x0 0x4A 0x6F 0x70 0x6C 0x69 0x6E 0x20 0x53 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 0x4E 0x8C

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

String: Joplin S ;

Raw bstream(hex): 0x4A-0x6F-0x70-0x6C-0x69-0x6E-0x20-0x53 ;

Raw bstream(dec): 74-111-112-108-105-110-32-83 ;

)

Notice: you may need to set offset value to get next part of HW version string.

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

Offset: **8**

Size: **8**

Press “**Send**”

MSG\_EXAMPLE:

[14:29:02:716]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x9 0x0 0x0 0x0 0x8 0x0 0x8 0x0 0x30 0x48

[14:29:02:729]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x57 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x9 0x0 0x0 0x0 0x8 0x0 0x8 0x0 0x2F 0x4D 0x20 0x45 0x53 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 0xF9 0x23

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

String: /M ES\_\_\_ ;

Raw bstream(hex): 0x2F-0x4D-0x20-0x45-0x53-0x0-0x0-0x0 ;

Raw bstream(dec): 47-77-32-69-83-0-0-0 ;

)

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:

[14:29:39:937]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xD 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0xF3 0x87

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

Example:

[14:29:39:951]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x57 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0xD 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x31 0x2E 0x31 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 0x0 0x0 0xEA 0xD6

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

String: 1.1\_ ;

Raw bstream(hex): 0x31-0x2E-0x31-0x0 ;

Raw bstream(dec): 49-46-49-0 ;

)

DSP version: **1.1**

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

Setting\_id: **SETID\_BT\_ADDR**

Offset: **0**

Size: **4**

Press “**Send**”

MSG\_EXAMPLE:

[09:03:03:355]: sent>>>>>>>> SETTING\_READ\_OFFSET\_REQ\_SIG:0xAA 0x65 0xC 0x17 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x10 0x0 0x0 0x0 0x0 0x0 0x4 0x0 0x23 0xEC

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

Example:

[10:19:59:154]: recv<<<<<<<< SETTING\_READ\_OFFSET\_RESP\_SIG:0xAA 0x2D 0x0 0x57 0x0 0x2D 0x0 0x3 0x1 0x0 0x0 0x0 0x0 0x10 0x0 0x0 0x0 0x0 0x0 0x6 0x0 0x0 0x2 0x5B 0x0 0xA5 0xB3 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 0xFF 0x6

-> SETTING\_READ\_OFFSET\_RESP\_SIG ( evtReturn : RET\_SUCCESS ; setting\_id : SETID\_BT\_ADDR ; offset : 0x0 ; size : 0x6 ; data :

String: \_[\_�� ;

Raw bstream(hex): 0x0-0x2-0x5B-0x0-0xA5-0xB3 ;

Raw bstream(dec): 0-2-91-0-165-179 ;

)

BT Address: **0002 5b 00a5b3**

## 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\_mask: **MASK\_ALL\_LED**

Led\_pattery:LED\_ON

Press “**Send**”

MSG\_EXAMPLE:

[14:33:03:832]: sent>>>>>>>> LED\_REQ\_SIG:0xAA 0x37 0x8 0x23 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0xFF 0xFF 0xFF 0xFF 0x1F 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0xD3 0x5C

As response **LED\_REQ\_SIG** will be received

Example:

[14:33:03:842]: recv<<<<<<<< LED\_RESP\_SIG:0xAA 0x1C 0x0 0xF 0x0 0x1C 0x0 0x1 0x1 0x0 0x0 0x0 0x0 0x3B 0xAD

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

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

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

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

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

Led\_pattery:LED\_OFF

Press “**Send**”

MSG\_EXAMPLE:

[14:34:14:515]: sent>>>>>>>> LED\_REQ\_SIG:0xAA 0x37 0x8 0x23 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0xFF 0xFF 0xFF 0xFF 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x67 0x2A

As response **LED\_REQ\_SIG** will be received

Example:

[14:33:03:842]: recv<<<<<<<< LED\_RESP\_SIG:0xAA 0x1C 0x0 0xF 0x0 0x1C 0x0 0x1 0x1 0x0 0x0 0x0 0x0 0x3B 0xAD

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

1. Notice:

Second led mask is stand for last 5 leds around treble knob.

## Set Source

1. To set **Source,** use **KEY\_STATE\_SIG,** set

Key\_ID: **INPUT\_KEY**

Key\_event: **KEY\_EVT\_SHORT\_PRESS**

Index:0

Press “**Send**”

MSG\_EXAMPLE:

[14:42:44:820]: sent>>>>>>>> KEY\_STATE\_SIG:0xAA 0x7 0x0 0x14 0x0 0x0 0x0 0x0 0x0 0x7 0x0 0x0 0x0 0x7 0x0 0x0 0x0 0x0 0x5 0x62

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

Example:

[14:42:44:849]: recv<<<<<<<< KEY\_STATE\_SIG:0xAA 0x7 0x0 0x17 0x0 0x7 0x0 0x2 0x1 0x7 0x0 0x0 0x0 0x7 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xB2 0xD1

-> KEY\_STATE\_SIG ( keyId : INPUT\_KEY ; keyEvent : KEY\_EVT\_SHORT\_PRESS ; INDEX : 0x0 ; )

Notice: audio source will switch to next one when you sent this meesage to DUT.

1. To set **Source,** use **MAINAPP\_PTE\_TEST\_CMD\_SIG,** set

EPteTestCmdIds: **PTE\_TEST\_CMD\_ID\_SOURCE\_SWITCH**

Enable/Disable: TRUE

Param1: **0-BT 1-AUX 2-RCA**

Param2: **0**

Press “**Send**”

MSG\_EXAMPLE:

[11:25:53:041]: sent>>>>>>>> MAINAPP\_PTE\_TEST\_CMD\_SIG:0xAA 0x80 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x2 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0xCA 0x37

As response **MAINAPP\_PTE\_TEST\_CMD\_SIG** will be received

Example:

[11:19:47:946]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xC0 0x0 0x0 0x20 0x0 0x0 0x0 0x0 0x6E 0xD2

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

## Set Volume

1. To set DUT **Volume,** use **MAINAPP\_PTE\_TEST\_CMD\_SIG,** set

EPteTestCmdIds: **PTE\_TEST\_CMD\_ID\_VOLUME**

Enable/Disable: **True**

Param1: volume:0-32

Param2:0

Press “**Send**”

MSG\_EXAMPLE:

[11:19:27:685]: sent>>>>>>>> MAINAPP\_PTE\_TEST\_CMD\_SIG:0xAA 0x80 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x9B 0x57

As response **MAINAPP\_PTE\_TEST\_CMD\_SIG** will be received

Example:

[11:19:47:946]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xC0 0x0 0x0 0x20 0x0 0x0 0x0 0x0 0x6E 0xD2

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

## Mute DSP Output Channel

1. To set mute/unmute **DSP Output Channel,** use **AUDIO\_MUTE\_SIG,** set

Mute: **Mute/Unmute**

audioMuteType: **AUDIO\_DSP\_MUTE\_WF / AUDIO\_DSP\_MUTE\_TW1 / AUDIO\_DSP\_MUTE\_TW2**

Press “**Send**”

MSG\_EXAMPLE:

[14:42:44:820]: sent>>>>>>>> KEY\_STATE\_SIG:0xAA 0x7 0x0 0x14 0x0 0x0 0x0 0x0 0x0 0x7 0x0 0x0 0x0 0x7 0x0 0x0 0x0 0x0 0x5 0x62

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

Example:

[10:38:07:205]: recv<<<<<<<< AUDIO\_MUTE\_RESP\_SIG:0xAA 0x22 0x0 0x17 0x0 0x22 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x10 0x14 0x0 0x20 0x0 0x4 0x0 0x0 0xA7 0x5A

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

[10:38:07:203]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xC0 0x0 0x0 0x20 0x0 0x0 0x0 0x0 0x6E 0xD2

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

## Set DUT to Standby

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

Press “**Send**”

MSG\_EXAMPLE:

[14:45:08:446]: sent>>>>>>>> MAINAPP\_STANDBY\_SIG:0xAA 0x72 0x1F 0xB 0x0 0x0 0x0 0x0 0x0 0x4C 0xB6

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

Example:

[14:45:08:503]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0x1 0xE 0x0 0x20 0x4 0x0 0x0 0x0 0x95 0x66

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

## Set Factory Reset

1. To set factory reset**,** use **KEY\_STATE\_SIG,** set

Key\_ID: **Factory reset comb**

Key\_event: **comb\_key\_evt**

Index:0

Press “**Send**”

MSG\_EXAMPLE:

[14:48:04:455]: sent>>>>>>>> KEY\_STATE\_SIG:0xAA 0x7 0x0 0x14 0x0 0x0 0x0 0x0 0x0 0x52 0x0 0x0 0x0 0x5 0x0 0x0 0x0 0x0 0x6D 0x5D

[14:48:04:517]: recv<<<<<<<< KEY\_STATE\_SIG:0xAA 0x7 0x0 0x17 0x0 0x7 0x0 0x2 0x2 0x52 0x0 0x0 0x0 0x5 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x75

-> KEY\_STATE\_SIG ( keyId : FACTORY\_RESET\_COMB ; keyEvent : COMB\_KEY\_EVT ; INDEX : 0x0 ; )

## Set DSP Bypass EQ

1. To set DSP Bypass EQ **,** use **AUDIO\_SET\_SIG,** set

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

Enable/Disable: **TRUE/FALSE**

value:0

not\_used:0

Press “**Send**”

MSG\_EXAMPLE:

[17:40:09:466]: sent>>>>>>>> AUDIO\_SET\_SIG:0xAA 0x56 0x7 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 0x7A 0xE0

## Play 1khz Test Cue

1. To Play Test Cue **,** use **BT\_REQ\_SIG,** set

Set\_ID: **BT\_TEST\_CUE\_CM**

Press “**Send**”

[14:04:54:536]: 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\_REQ\_SIG** will be received

Example:

[14:04:54:543]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xC0 0x0 0x0 0x20 0x0 0x0 0x0 0x0 0x6E 0xD2

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

## Set System Work Mode

1. To set **Work mode,** use **MAINAPP\_PTE\_TEST\_CMD\_SIG,** set

EPteTestCmdIds: **PTE\_TEST\_CMD\_ID\_SET\_WORK\_MODE**

Enable/Disable: TRUE

Param1: **0-Normal Mode 1-Shop(POS) Mode**

Param2: **0**

Press “**Send**”

MSG\_EXAMPLE:

[16:12:53:176]: sent>>>>>>>> MAINAPP\_PTE\_TEST\_CMD\_SIG:0xAA 0x86 0x1F 0x1F 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x3 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x1 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x8A 0xCC

As response **MAINAPP\_PTE\_TEST\_CMD\_SIG** will be received

Example:

[16:13:48:235]: recv<<<<<<<< DEBUG\_RESP\_SIG:0xAA 0x1D 0x0 0x17 0x0 0x1D 0x0 0x2 0x1 0x0 0x0 0x0 0x0 0xC0 0x0 0x0 0x20 0x0 0x4 0x0 0x0 0xAE 0xE

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