



AB156x Series Mass Production RACE Application Note

Version: 1.10
Release date: 26 November 2021

© 2020 Airoha Technology Corp.

This document contains information that is proprietary to Airoha Technology Corp. ("Airoha") and/or its licensor(s). Airoha cannot grant you permission for any material that is owned by third parties. You may only use or reproduce this document if you have agreed to and been bound by the applicable license agreement with Airoha ("License Agreement") and been granted explicit permission within the License Agreement ("Permitted User"). If you are not a Permitted User, please cease any access or use of this document immediately. Any unauthorized use, reproduction or disclosure of this document in whole or in part is strictly prohibited. THIS DOCUMENT IS PROVIDED ON AN "AS-IS" BASIS ONLY. AIROHA EXPRESSLY DISCLAIMS ANY AND ALL WARRANTIES OF ANY KIND AND SHALL IN NO EVENT BE LIABLE FOR ANY CLAIMS RELATING TO OR ARISING OUT OF THIS DOCUMENT OR ANY USE OR INABILITY TO USE THEREOF. Specifications contained herein are subject to change without notice.

Document revision history

Revision	Date	Description
1.0	17 April 2020	Initial release
1.1	04 May 2020	Added 2-mic test function. Renamed document from AB156x_Series_ANC_Calibration_Application_Note to AB156x_Series_Audio_Calibration_Application_Note
1.2	15 June 2020	Added 3-mic test supported function. Updated ANC gain index mapping.
1.3	15 September 2020	Added Mic Swap parameter for support 6-mic test
1.4	17 September 2020	Added command to enter mp mode and leave mp mode
1.5	16 November 2020	Renamed document from AB156x_Series_Audio_Calibration_Application_Note to AB156x_Mass_Production_RACE_Application_Note
1.6	25 November 2020	Added MCSync key for AB1565/1568 writing MCSync information
1.7	27 November 2020	Added UART software flow control
1.8	05 July 2021	Enhanced ANC RACE commands to support speaker gains Corrected RACE ID of get version and get BD address commands
1.9	01 November 2021	Added command for factory reset Added information: USB-HID dongle & 1wire uart MCSync pairing/un pairing USB-HID LE dongle & 1wire uart MCSync pairing/un pairing USB-HID dongle & Headset pairing/un pairing Added analog gain calibration
1.10	26 November 2021	Modify MCSync information for AB1562

Table of contents

1.	Introduction.....	1
2.	RACE command packet.....	2
2.1.	RCMD Packet Format.....	2
2.2.	Type List.....	2
3.	UART flow control	3
4.	ANC RACE command.....	4
4.1.	ANC On	5
4.2.	ANC Off	6
4.3.	Set ANC Gain.....	6
4.4.	Read ANC gain from NvKey	7
4.5.	Write ANC gain to NvKey	8
4.6.	Get ANC hybrid capability.....	8
4.7.	Enter ANC Mp Mode.....	9
4.8.	Leave ANC Mp Mode	9
5.	Relay RACE command.....	10
5.1.	Get Available Destination	10
5.2.	Relay command to partner.....	10
6.	ANC calibration flow.....	11
6.1.	156x Series Hybrid ANC diagram	11
6.2.	FB/FF ANC gains.....	11
6.3.	ANC calibration flow chart.....	11
7.	Sub Function RACE command	12
7.1.	Read NV key.....	12
7.2.	Write NV key.....	12
7.3.	Get version (AB1565/1568)	13
7.4.	Get version (AB1562).....	13
7.5.	Set PEQ index.....	14
7.6.	Power OFF	14
7.7.	Get battery level	14
7.8.	Get BD address	14
7.9.	Write MCSync information (AB1565/1568).....	15
7.10.	Write MCSync information (AB1562)	15
7.11.	Read/Write device name.....	16
7.12.	Get model name	16
7.13.	Get audio channel.....	16
7.14.	Enable/Disable DUT mode.....	16
7.15.	Factory Reset	16
7.16.	Write USB-HID dongle & headset pairing information (AB1565/1568)	17
7.17.	Write USB-HID dongle & MCSync pairing information (AB1565/1568)	17
7.18.	Write USB-HID LE dongle & MCSync SIRQ key (AB1565/1568).....	17
7.19.	Un Pairing: Write MCSync information (AB1565/1568)	18
7.20.	Un Pairing: Write USB-HID dongle & headset information (AB1565/1568)	18
7.21.	Un Pairing: Write USB-HID dongle & MCSync information (AB1565/1568)	18
7.22.	Un Pairing: Write USB-HID LE dongle & MCSync SIRQ key (AB1565/1568)	19
8.	Mic test RACE Command	20
8.1.	2-mic BIST	20
8.2.	MIC Swap	20

8.3.	AECNR on/off	20
9.	Analog Gain calibration flow	22
9.1.	Read/Write Analog Gain	22
9.2.	DSP Suspend RACE command	23
9.3.	DSP Resume RACE command	23

Lists of tables and figures

Table 2-1.RACE command format.....	2
Table 2-2.RACE receive format	2
Table 2-3.RACE type list	2
Table 3-1. UART flow control encoding table	3
Table 3-2. UART flow control decoding table	3
Table 4-1.ANC RACE payload ID	4
Table 4-2.ANC gain index mapping	4
Figure 6-1. 156x Series Hybrid ANC diagram	11
Figure 6-2. ANC calibration flow chart	11
Figure 9-1. Analog gain calibration flow chart	22

1. Introduction

This application note describes AB156x series Mass Production RACE related information. The following topics are included to help users to establish Audio Mass Production environment.

1. AB156x RACE definition
2. UART software flow control
3. ANC calibration flow
4. ANC RACE commands
5. Relay RACE commands
6. Sub-function RACE commands
7. Mic test RACE commands

ANC RACE commands are used to calibrate ANC.

Relay RACE commands are used to send RACE to partner for MCSync ANC calibration.

Sub-function RACE commands is to support version check, model name check...etc.

Mic test RACE commands are used to test mic functionality.

2. RACE command packet

The Race Command (RCMD) packet is used to send commands to AB156x device from the Host (external MCU or PC tool) or receive events (indications or responses) from AB156x device. Any AB156x device is able to accept RCMD with **up to 1000 bytes** of data excluding the RCMD header and length field. Each RCMD command is assigned two types of transported used to uniquely identify different format of commands. These two fields called the Transported by H4 and Transported by H5.

2.1. RCMD Packet Format

RCMD Command Format (send to AB156x)

Command				
Channel	Type	Length	ID	Payload
1 byte	1 byte	2 bytes	2 bytes	Varied
0x05	0x5A or 0x5C	#1	RACE Command ID	#2

Table 2-1.RACE command format

#1 ID + Payload

#2 Command parameters

#3 Little Endian used for multi-bytes area

RCMD Receive Format (receive from AB156x)

Response				
Channel	Type	Length	ID	Payload
1 byte	1 byte	2 bytes	2 bytes	Varied
0x05	0x5B or 0x5D	#1	RACE Command ID	#2

Table 2-2.RACE receive format

RCMD Commands are sent from host to AB156x via UART. AB156x would response with individual 'ID' which represents the ID of the responding command.

2.2. Type List

Type ID	Description
0x5A	Command Needs Response
0x5B	Response
0x5C	Command Doesn't Need Response
0x5D	Notification

Table 2-3.RACE type list

3. UART flow control

AB1562/AB1565/AB1568 implements UART software flow control which uses 0x11 and 0x13 as control bytes. If sending RACE via UART, please encode/decode data according to the following tables.

Sending raw data	Encoded data
0x11	0x77 0xEE
0x13	0x77 0xEC
0x77	0x77 0x88

Table 3-1. UART flow control encoding table

Receiving raw data	Decoded data
0x77 0xEE	0x11
0x77 0xEC	0x13
0x77 0x88	0x77

Table 3-2. UART flow control decoding table

4. ANC RACE command

Payload ID

All ANC command use RACE command ID **0x0E06**, and defined Payload ID in payload:

Payload ID	
Description	ID
ANC On	0x0A
ANC Off	0x0B
Set ANC Gain	0x0C
Read ANC gain from NvKey	0x0D
Write ANC gain to NvKey	0x0E
Get ANC hybrid capability	0x16

Table 4-1.ANC RACE payload ID

ANC gain index mapping

Gain Index	Gain Value (dB)
0x0258(600)	6
...	..
0x0000	0
0xFFFF(-1)	-0.01
0xFFFE(-2)	-0.02
...	Gain value = Gain index/100
0xFF9C(-100)	-1
...	...
0xFA24(-1500)	-15
...	...
0xDCD8(-9000)	-90

Table 4-2.ANC gain index mapping

4.1. ANC On

Command (0x055A)								
Length		ID		Payload				
2 bytes		2 bytes		5 bytes				
0x07	0x00	0x06	0x0E	Status	ID	Filter coefficient index	ANC mode	Sync mode
				0x00	0x0A	XX	00:Hybrid 01:FF only 02:FB only 04:AiroThru	00: Turn on agent ANC only 01: Turn on both agent and partner ANC

Response (0x055B)									
Length		ID		Payload					
2 bytes		2 bytes		6 bytes					
0x08	0x00	0x06	0x0E	Status	ID	Filter coefficient index	ANC mode	Sync mode	reserved
				0x00: success Else: fail	0x0A	XX	00:Hybrid 01:FF only 02:FB only 04:AiroThru	00 or 01	XX

Filter coefficient index: ANC: 0x01 ~ 0x04
Airo through: 0x09 ~ 0x0B

4.2. ANC Off

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		3 bytes		
0x05	0x00	0x06	0x0E	Status	ID	Sync mode
				0x00	0x0B	00: Turn off agent ANC only 01: Turn off both agent and partner ANC

Response (0x055B)									
Length		ID		Payload					
2 bytes		2 bytes		6 bytes					
0x08	0x00	0x06	0x0E	Status	ID	Sync mode	Reserved		
				0x00: success Else: fail	0x0B	00 or 01	XX	XX	XX

4.3. Set ANC Gain

Command (0x055A)													
Length		ID		Payload									
2 bytes		2 bytes		10 bytes									
0x0C	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R	
				0x00	0x0C	XX	XX	XX	XX	XX	XX	XX	XX

Response (0x055B)													
Length		ID		Payload									
2 bytes		2 bytes		10 bytes									
0x0C	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R	
				0x00:success Else: fail	0x0C	XX	XX	XX	XX	XX	XX	XX	XX

AB1562 SDK 1.4.0 and AB1562E SDK 2.0.0 support speaker gain settings. The command is extended as below.

Command (0x055A)																	
Length		ID		Payload													
2 bytes		2 bytes		14 bytes													
0x10	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R		Gain SPK L		Gain SPK R	
				0x00	0x0C	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

Command (0x055B)																	
Length		ID		Payload													
2 bytes		2 bytes		14 bytes													
0x10	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R		Gain SPK L		Gain SPK R	
				0x00: success Else: fail	0x0C	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

4.4. Read ANC gain from NvKey

Command (0x055A)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00	0x0D

Response (0x055B)													
Length		ID		Payload									
2 bytes		2 bytes		10 bytes									
0x0C	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R	
				0x00: success Else: fail	0x0D	xx	xx	xx	xx	xx	xx	xx	xx

AB1562 SDK 1.4.0 and AB1562E SDK 2.0.0 support speaker gain settings. The command is extended as below.

Command (0x055A)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00	0x0D

Command (0x055B)																	
Length		ID		Payload													
2 bytes		2 bytes		14 bytes													
0x10	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R		Gain SPK L		Gain SPK R	
				0x00: success Else: fail	0x0D	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

4.5. Write ANC gain to NvKey

Command (0x055A)													
Length		ID		Payload									
2 bytes		2 bytes		10 bytes									
0x0C	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R	
				0x00	0x0E	XX	XX	XX	XX	XX	XX	XX	XX

Response (0x055B)														
Length		ID		Payload										
2 bytes		2 bytes		10 bytes										
0x0C	0x00	0x06	0x0E	Status		ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R	
				0x00: success Else: fail		0x0E	XX	XX	XX	XX	XX	XX	XX	XX

AB1562 SDK 1.4.0 and AB1562E SDK 2.0.0 support speaker gain settings. The command is extended as below.

Command (0x055A)																	
Length		ID		Payload													
2 bytes		2 bytes		14 bytes													
0x10	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R		Gain SPK L		Gain SPK R	
				0x00	0x0E	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

Command (0x055B)																	
Length		ID		Payload													
2 bytes		2 bytes		14 bytes													
0x10	0x00	0x06	0x0E	Status	ID	Gain FF L		Gain FB L		Gain FF R		Gain FB R		Gain SPK L		Gain SPK R	
				0x00: success Else: fail	0x0E	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX	XX

4.6. Get ANC hybrid capability

Command (0x055A)													
Length		ID		Payload									
2 bytes		2 bytes		2 bytes									
0x04	0x00	0x06	0x0E	Status						ID			
				0x00						0x16			

Response (0x055B)													
Length		ID		Payload									
2 bytes		2 bytes		3 bytes									
0x05	0x00	0x06	0x0E	Status		ID	Hybrid capability						
				0x00: success Else: fail		0x16	0x01: support hybrid						

4.7. Enter ANC Mp Mode

Command (0x055A)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00	0x10

Response (0x055B)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00: success Else: fail	0x10

4.8. Leave ANC Mp Mode

Command (0x055A)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00	0x11

Response (0x055B)					
Length		ID		Payload	
2 bytes		2 bytes		2 bytes	
0x04	0x00	0x06	0x0E	Status	ID
				0x00: success Else: fail	0x11

5. Relay RACE command

5.1. Get Available Destination

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		N/A
0x02	0x00	0x00	0x0D	N/A

Response (0x055B)				
Length		ID		Payload
2 bytes		2 bytes		N bytes
XX	0x00	0x00	0x0D	Destination list
				Pairs of [dst type:1 byte][dst id: 1 byte] e.g. 0x01020506 (type USB and type AWS peer)

* dst type: 0 uart, 1 usb, 2 airapp, 5 AWS peer

5.2. Relay command to partner

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
XX	XX	0x01	0x0D	Dst type	Dst ID	Data to partner
				0x05	%AWS_peer_ID	

Response (0x055C)							
Length		ID		Payload			
2 bytes		2 bytes		N bytes			
XX	XX	0x01	0x0D	Status	Dst type	Dst ID	Data from partner
				0x00: success Else: fail	0x05	%AWS_peer_ID	

* %AWS_peer_ID is queried by Get Available Destination command. Type is 0x05 (AWS_peer).

Relay example:

Step 1: get AWS peer destination ID

055A0200000D

055B0600000D03040506 => get AWS peer ID: 06

Step 2: using AWS peer destination ID to send ANC OFF command to partner

Relay ANC OFF command to partner

055A0D00010D0506 055A0500060E000B00

Get ANC OFF Response from partner

055D1000010D0506 055B0800060E000B00000000

6. ANC calibration flow

6.1. 156x Series Hybrid ANC diagram

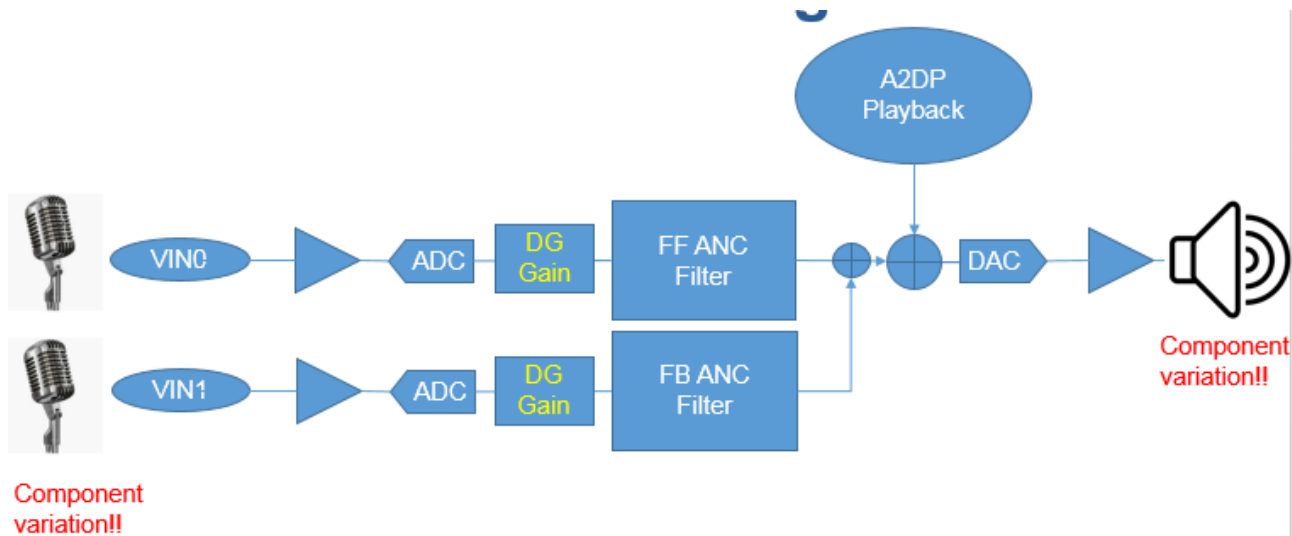


Figure 6-1. 156x Series Hybrid ANC diagram

6.2. FB/FF ANC gains

There are 2 gains to be calibrated during ANC test.

- FB DG gain: The FB gain should be calibrated in the first stage under ANC FB mode.
- FF DG gain: After FB gain is calibrated, the FF gain has to be calibrated under ANC hybrid mode.

6.3. ANC calibration flow chart

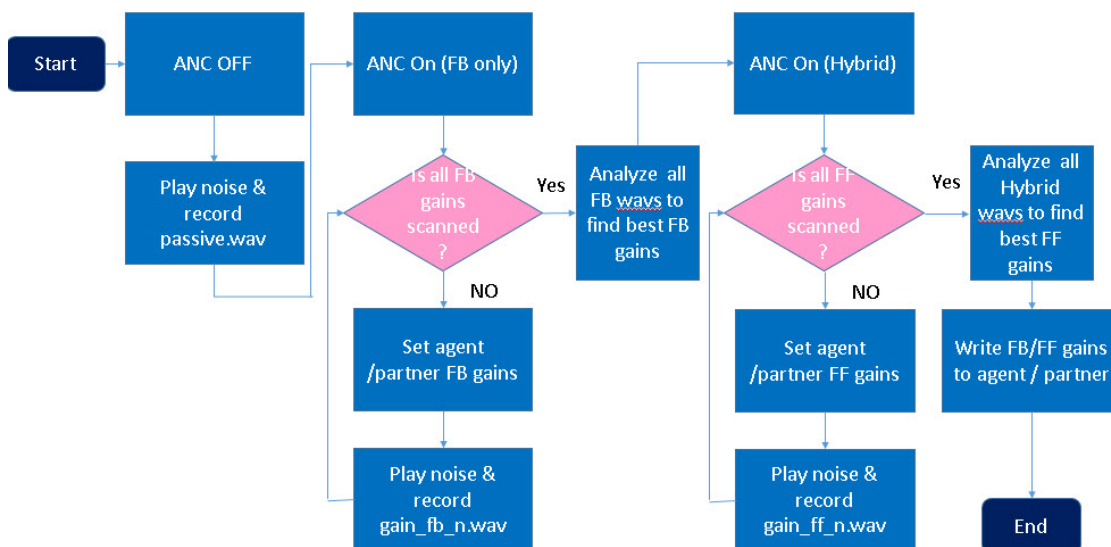


Figure 6-2. ANC calibration flow chart

7. Sub Function RACE command

7.1. Read NV key

Command (0x055A)							
Length		ID		Payload			
2 bytes		2 bytes		4 bytes			
0x06	0x00	0x00	0x0A	NV_ID_B0	NV_ID_B1	Length_B0	Length_B1
				XX	XX	XX	XX

Command (0x055B)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
XX	XX	0x00	0x0A	Length_B0	Length_B1	NV value (N-2 bytes)
				XX	XX	XX

e.g.

Read NV ID = 0xF500, Length = 0x0028

055A060000A00F52800

Response, Length = 0x0028, NV value = 0x5941595500

055B2C0000A280**594159550000000000000000000000006D6F64656C3100000000
00000000000000000000**

7.2. Write NV key

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
0xXX	0xXX	0x01	0x0A	NV_ID_B0	NV_ID_B1	Payload
				XX	XX	NV values

Command (0x055B)					
Length		ID		Payload	
2 bytes		2 bytes		1 byte	
03	00	0x01	0x0A	Status	
				0x00: success Else: fail	

e.g.

Write NV ID = **0x3A00**, Value = **0x00**

055A0500010A003A00

Response, Status = 00

055B0300010A00

7.3. Get version (AB1565/1568)

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		Role: 1 byte
0x03	0x00	0x07	0x1C	Agent: 0x00

Notification (0x055D)							
Length		ID		Payload			
2 bytes		2 bytes		N bytes			
XX	XX	0x07	0x1C	Status	Role (1 byte)	Length (1 byte)	version (N-3 bytes) in ASCII
				0x00: success Else: fail	0x00: agent	XX	XX

e.g.

055A0300071C00

Notification, Length = 0x06, NV value = 0x76312E302E30

055D0B00071C00000676312E302E30

0x76312E302E30 in ASCII is "v1.0.0".

7.4. Get version (AB1562)

Command (0x055A)					
Length		ID		Payload	
2 bytes		2 bytes		Recipient Count (1 byte)	Role (1 byte)
0x04	0x00	0x07	0x1C	0x01	DUT: 0xFF

Notification (0x055D)								
Length		ID		Payload				
2 bytes		2 bytes		N bytes				
XX	XX	0x07	0x1C	Status	Recipient Count (1 byte)	Role (1 byte)	Length (1 byte)	version (N-4 bytes) in ASCII
				0x00: success Else: fail	0x01	DUT: 0xFF	XX	XX

e.g.

055A0400071C01FF

Notification, Length = 0x07, NV value = 0x312E312E312E31

055D0D00071C0001FF07312E312E312E31

0x312E312E312E31 in ASCII is "1.0.0.0".

7.5. Set PEQ index

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		Module (2 bytes)	PEQ index (1 byte)	
0x05	0x00	0x00	0x09	0x00	0x00	index

Notification (0x055D)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
0x05	0x00	0x00	0x09	Module (2 bytes)		Status
				0x00	0x00	0x00: success Else: fail

7.6. Power OFF

Command (0x055A)						
Length		ID				
2 bytes		2 bytes				
0x02	0x00	0x09			0x02	

7.7. Get battery level

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		Role (1 byte)		
0x03	0x00	0xD6	0x0C	Agent: 0x00		

Notification (0x055D)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
0x05	0x00	0xD6	0x0C	Status	Role (1 byte)	Battery level (1 byte)
				0x00: success Else: fail	0x00: agent	Unit: percentage

e.g.

055A0300D60C00

055D0500D60C000050

Battery level is 80%.

7.8. Get BD address

Command (0x055A)						
Length		ID		Payload		
2 bytes		2 bytes		Role: 1 byte		
0x03	0x00	0xD5	0x0C	Agent: 0x00		

Response (0x055B)						
Length		ID		Payload		
2 bytes		2 bytes		N bytes		
0x05	0x00	0xD5	0x0C	Status	Role (1 byte)	BD address (6 bytes)
				0x00: success Else: fail	0x00: agent	

e.g.

055A0300D50C00

055B0A00D50C0000665544332211 BD address is 0x112233445566.

7.9. Write MCSync information (AB1565/1568)

MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

e.g.

Agent BD address: 0x**112233445566**

Partner BD address: 0x**998877665544**

MCSync key: 0x**01020304050607080910111213141516**

Write to agent (0x40)

055A3400010A B0F2 **40** 0000 **445566778899** FF **665544332211**

01**01020304050607080910111213141516**00000000000000000000000000000000

Write to partner (0x20)

055A3400010A B0F2 **20** 0000 **665544332211** FF **445566778899**

01**01020304050607080910111213141516**00000000000000000000000000000000

***If you want to keep values of other fields, please read NV back, replace agent BDA, partner BDA, role, MCSync key then write it back.**

***The agent and partner must have the same MCSync key in one group, but the different agent partner group should use a different MCSync key.**

7.10. Write MCSync information (AB1562)

MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

e.g.

Agent BD address: 0x**112233445566**

Partner BD address: 0x**998877665544**

Write to agent (0x40)

055A1A00010A B0F2 **40** 0000 **665544332211** FF **445566778899** 000000000000

Write to partner (0x30)

055A1A00010A B0F2 **30** 0000 **665544332211** FF **445566778899** 000000000000

***If you want to keep values of other fields, please read NV back, replace agent BDA, partner BDA, role then write it back.**

7.11. Read/Write device name

Device name is saved in the NV key 0xF203 in ASCII format. Please use Read/Write NV RACE commands to access it.

7.12. Get model name

Model name is saved in the 21st to 40th bytes of NV key 0xF500 in ASCII format. Please use Read NV RACE command to get it.

7.13. Get audio channel

Audio channel setting is saved in the 2nd byte of NV key 0xF2B5. Please use Read NV RACE command to get it.

Value = {
 1: Left channel
 2: Right channel
 }

e.g.
 055A0600000AB5F2E803
 Response, Length = 0x0009, NV value = 0x0001010214, Left channel
 055B0900000A05000001010214

7.14. Enable/Disable DUT mode

DUT mode control is saved in the NV key 0xF2C2. Please use Write NV RACE command to enable/disable it.

e.g.
 Write NV ID = 0xF2C2, Value = 0x01 (0x00 for disable, 0x01 for enable)
 055A0500010AC2F201
 Response, Status = 00
 055B0300010A00

7.15. Factory Reset

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		2 byte
0x04	0x00	0x01	0x11	0x9500

Response (0x055B)		
Length	ID	Payload
2 bytes	2 bytes	1 byte

0x03	0x00	0x01	0x11	0x00: success Else: fail
------	------	------	------	-----------------------------

7.16. Write USB-HID dongle & headset pairing information (AB1565/1568)

USB-HID dongle & headset pairing is saved by NV key 0x1815. Please use the Write NV RACE command to write it.

e.g.

Dongle BD address: 0x**112233445566**

Headset BD address: 0x**998877665544**

Write to Dongle

055A0A00010A 1518 **445566778899**

Write to Headset

055A0A00010A 1518 **665544332211**

7.17. Write USB-HID dongle & MCSync pairing information (AB1565/1568)

Flow:

1. MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

Please refer to 7.9Write MCSync information (AB1565/1568)

2. USB-HID dongle & MCSync pairing is saved by NV key 0x1815. Please use the Write NV RACE command to write it.

e.g.

Dongle BD address: 0x**112233445566**

MCSync Agent BD address: 0x**998877665544**

Write to Dongle

055A0A00010A 1518 **445566778899**

Write to MCSync (earbuds)

055A0A00010A 1518 **665544332211**

7.18. Write USB-HID LE dongle & MCSync SIRK key (AB1565/1568)

Flow:

1. MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

Please refer to 7.9Write MCSync information (AB1565/1568)

2. USB-HID LE dongle & MCSync SIRK key is saved by NV key 0x1900. Please use the Write NV RACE command to write it.

e.g.

LE Dongle:

Write NV ID = **0x1900**, Value1 = **0x01020304050607080910111213141516** (SIRK: 16bytes random num.)

055A1600010A0019010203040506070809101112131415160000

MCSync (earbuds):

Write NV ID = **0x1900**, Value1 = **0x01020304050607080910111213141516** (SIRK: 16bytes random num.)

Value2 = 0x02 (Size: earbuds num., default = 0x02) , Value3 = 0x01(Rank)

055A1600010A0019010203040506070809101112131415160201

***If you want to keep values of other fields, please read NV back, replace SIRQ key then write it back.**

***The LE dongle and earbuds must have the same SIRQ key in one group, but the different LE dongle and earbuds group should use a different SIRQ key.**

7.19. Un Pairing: Write MCSync information (AB1565/1568)

Flow:

1. Un Pairing MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

e.g.

Write to agent and partner

055A3400010A B0F2 40 0000 000000000000 FF 000000000000

[illegible]

2. Please use command: Factory Reset, refer to 7.15Factory Reset

7.20. Un Pairing: Write USB-HID dongle & headset information (AB1565/1568)

Flow:

1. Un Pairing USB-HID dongle & headset is saved by NV key 0x1815. Please use the Write NV RACE command to write it.

e.g.

Write to Dongle

055A0A00010A 1518 000000000000

Write to Headset

055A0A00010A 1518 000000000000

2. Please use command: Factory Reset, refer to 7.15Factory Reset

7.21. Un Pairing: WriteUSB-HID dongle & MCSync information (AB1565/1568)

Flow:

1. Un Pairing MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.

Please refer to 7.19 Un Pairing: Write MCSync information (AB1565/1568)

2. Un Pairing USB-HID dongle & MCSync is saved by NV key 0x1815. Please use the Write NV RACE command to write it.

e.g.

Write to Dongle

055A0A00010A 1518 000000000000

Write to MCSync (earbuds)

055A0A00010A 1518 000000000000

3. Please use command: Factory Reset, refer to 7.15Factory Reset

7.22. Un Pairing: Write USB-HID LE dongle & MCSync SIRQ key (AB1565/1568)

Flow:

1. Un Pairing MCSync setting is saved by NV key 0xF2B0. Please use the Write NV RACE command to write it.
Please refer to 7.19 Un Pairing: Write MCSync information (AB1565/1568)

2. Un Pairing USB-HID LE dongle & MCSync, SIRQ key is saved by NV key 0x1900. Please use the Write NV RACE command to write it.

e.g.

Original SIRQ Key: 0x01010101010101010101010101010101

MCSync (earbuds1):

Write NV ID = 0x1900, Value1 = 0x02020202020202020202020202020202

055A1600010A0019020202020202020202020202020202020000

MCSync (earbuds2):

Write NV ID = 0x1900, Value1 = 0x03030303030303030303030303030303

055A1600010A0019030303030303030303030303030303030201

LE Dongle:

Write NV ID = 0x1900, Value1 = 0x04040404040404040404040404040404

055A1600010A0019040404040404040404040404040404040201

***Let the LE dongle and earbuds have a different SIRQ key.**

3. Please use command: Factory Reset, refer to 7.15Factory Reset

8. Mic test RACE Command

8.1. 2-mic BIST

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x00	0x10	Test mode
				01: silence mode 02: ADDA left mode 03: ADDA right mode

Command (0x055B)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
03	00	0x00	0x10	Status
				00: success Else: fail

This command is supported only for AB1562.

8.2. MIC Swap

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x0C	0x0E	MIC0 (0x00) MIC1 (0x01) MIC2 (0x02) MIC3 (0x03) MIC4 (0x04) MIC5 (0x05)

Response (0x055B)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x0C	0x0E	0x00: success Else: fail

For AB1562, this command is supported from MP v1.2.0.

8.3. AECNR on/off

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x0D	0x0E	0x00 (Off) 0x01 (On)

Response (0x055B)				
Length		ID		Payload



AB156x Series Mass Production RACE Application Note

2 bytes		2 bytes		1 byte
0x03	0x00	0x0D	0x0E	0x00: success Else: fail

For AB1562, this command is supported from MP v1.2.0.

9. Analog Gain calibration flow

Analog gain calibration is supported in AB1565/AB1568 only.

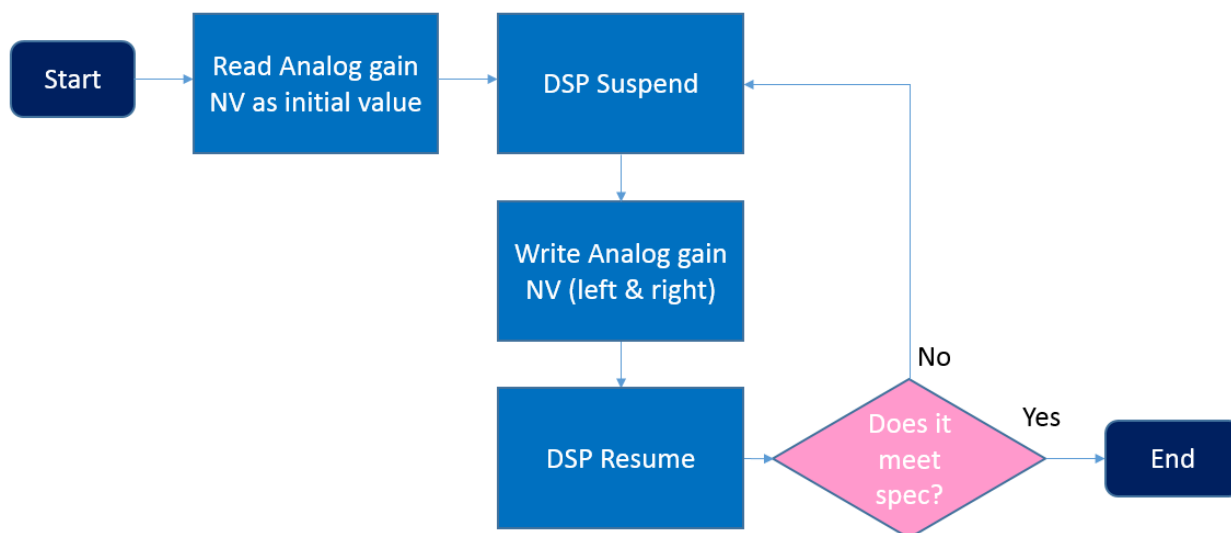


Figure 9-1. Analog gain calibration flow chart

9.1. Read/Write Analog Gain

Analog gain setting is saved by NV key 0xE00A. Please use the Read/Write NV RACE command to access it.

In NV key 0xE00A, the 3rd and 4th bytes composes left analog gain and the 7th and 8th bytes composes right analog gain in unit of 0.01 db.

e.g.

Analog gain left: 0x0190 (400 in decimal. i.e. 4db)

Analog gain right: 0xFF38 (-200 in decimal. i.e. -2db)

Write NV ID = **0xE00A**, Value = 0x00009001.....

055A3C00010A0AE00009001900138FF08070807B004B00408070807000000000000000000
00000807080708070807080708070807080708070807

Response, Status = 00

055B0300010A00

9.2. DSP Suspend RACE command

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		0 byte
0x02	0x00	0x01	0x0E	

Command (0x055B)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x01	0x0E	Status
				00: success
				Else: fail

e.g.

055A0200010E

055B0300010E00

9.3. DSP Resume RACE command

Command (0x055A)				
Length		ID		Payload
2 bytes		2 bytes		0 byte
0x02	0x00	0x02	0x0E	

Command (0x055B)				
Length		ID		Payload
2 bytes		2 bytes		1 byte
0x03	0x00	0x02	0x0E	Status
				00: success
				Else: fail

e.g.

055A0200020E

055B0300020E00