AG35 Series QuecOpen Audio Volume Adjustment API Reference Manual

Automotive Module Series

Version: 1.0

Date: 2020-10-12

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
Building 5, Shanghai Business Park Phase III (Area B), No.1016 Tianlin Road, Minhang District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm.

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to support@quectel.com.

General Notes

Quectel offers the information as a service to its customers. The information provided is based upon customers' requirements. Quectel makes every effort to ensure the quality of the information it makes available. Quectel does not make any warranty as to the information contained herein, and does not accept any liability for any injury, loss or damage of any kind incurred by use of or reliance upon the information. All information supplied herein is subject to change without prior notice.

Disclaimer

While Quectel has made efforts to ensure that the functions and features under development are free from errors, it is possible that these functions and features could contain errors, inaccuracies and omissions. Unless otherwise provided by valid agreement, Quectel makes no warranties of any kind, implied or express, with respect to the use of features and functions under development. To the maximum extent permitted by law, Quectel excludes all liability for any loss or damage suffered in connection with the use of the functions and features under development, regardless of whether such loss or damage may have been foreseeable.

Duty of Confidentiality

The Receiving Party shall keep confidential all documentation and information provided by Quectel, except when the specific permission has been granted by Quectel. The Receiving Party shall not access or use Quectel's documentation and information for any purpose except as expressly provided herein. Furthermore, the Receiving Party shall not disclose any of the Quectel's documentation and information to any third party without the prior written consent by Quectel. For any noncompliance to the above requirements, unauthorized use, or other illegal or malicious use of the documentation and information, Quectel will reserve the right to take legal action.

Copyright

About the Document

Revision History

Version	Date	Author	Description
-	2020-08-14	Ewen LI	Creation of the document
1.0	2020-10-12	Ewen LI	First official release

Contents

Table Index

1 Introduction

Quectel AG35 series supports QuecOpen® solution. QuecOpen® is an open-source embedded development platform based on Linux system, which is intended to simplify the design and development of IoT applications. For more information on QuecOpen®, see ***document [1]***.

This document introduces the APIs through which you can adjust the audio volume of AG35 series in QuecOpen® solution.

2 Audio Volume Adjustment APIs

2.1. Header File Path

The header file *ql_acdb.h* of audio volume adjustment APIs is located in the QuecOpen SDK directory of *ql-ol-sdk/ql-ol-crosstool/sysroots/armv7ahf-neon-oe-linux-gnueabi/usr/include/*.

2.2. Example Path

The use examples, which demonstrate how the APIs are best used, are located in the QuecOpen SDK directory of *ql-ol-sdk/ql-ok-extsdk/example/acdb*.

2.3. API Overview

Table 1: API Overview

Function	Description
ql_audplay_gain_write()	Sets audio playback volume
ql_audplay_gain_read()	Reads audio playback volume
ql_audrd_gain_write()	Sets audio recording volume
ql_audrd_gain_read()	Reads audio recording volume
ql_clvl_write()	Sets downlink call volume level
ql_clvl_read()	Reads downlink call volume level
ql_mic_gain_write()	Sets uplink call gain
ql_mic_gain_read()	Reads uplink call gain

ql_spk_gain_write()	Sets downlink call gain
ql_spk_gain_read()	Reads downlink call gain
ql_audloop_write()	Enables/disables loopback
ql_audloop_read()	Reads the current state of loopback
ql_sidet_write()	Sets sidetone gain
ql_sidet_read()	Reads sidetone gain

NOTE

Unless otherwise specified, all above APIs do not support concurrent calls, and do not call them in any callback function.

2.4. API Description

2.4.1. ql_audplay_gain_write

This function sets the audio playback volume of the module.

- **Prototype**

```
int ql_audplay_gain_write(unsigned short gain);
```

- **Parameter**

gain:
[In] Audio playback volume. Range: 0–65535.

- **Return Value**

0	Set the audio playback volume successfully.
Other values	Failed to set the audio playback volume.

NOTE

A set of examples illustrating the use of the API is available in *example_audplay_gain.c*. Call this API before audio playback, otherwise it cannot take effect immediately.

2.4.2. ql_audplay_gain_read

This function reads audio playback volume of the module.

- **Prototype**

```
int ql_audplay_gain_read(void);
```

- **Parameter**

None.

- **Return Value**

-1	Failed to read the audio playback volume.
Other values	The audio playback volume.

NOTE

A set of examples illustrating the use of the API is available in *example_audplay_gain.c*.

2.4.3. ql_audrd_gain_write

This function sets audio recording volume of the module.

- **Prototype**

```
int ql_audrd_gain_write (unsigned short gain);
```

- **Parameter**

gain:
[In] Audio recording volume. Range: 0–65535.

- **Return Value**

0	Set the audio recording volume successfully.
Other values	Failed to set the audio recording volume.

NOTE

A set of examples illustrating the use of the API is available in *example_audrd_gain.c*. Call this API before audio recording, otherwise it cannot take effect immediately.

2.4.4. ql_audrd_gain_read

This function reads audio recording volume of the module.

- **Prototype**

```
int ql_audplay_gain_read(void);
```

- **Parameter**

None.

- **Return Value**

-1	Failed to read the audio recording volume.
Other values	The audio recording volume.

NOTE

A set of examples illustrating the use of the API is available in *example_audrd_gain.c*.

2.4.5. ql_clvl_write

This function sets the downlink call volume level of the module.

- **Prototype**

```
int ql_clvl_write(unsigned int value);
```

- **Parameter**

value:
[In] The downlink call volume level. Range: 0–5.

- **Return Value**

1	Set the downlink call volume level successfully.
0	Failed to set the downlink call volume level.

NOTE

A set of examples illustrating the use of the API is available in *example_clvl.c*.

2.4.6. ql_clvl_read

This function reads the downlink call volume level of the module.

- **Prototype**

```
int ql_clvl_read(void);
```

- **Parameter**

None.

- **Return Value**

The downlink call volume level of the module.

NOTE

A set of examples illustrating the use of the API is available in *example_clvl.c*.

2.4.7. ql_mic_gain_write

This function sets the uplink call gain to change the uplink call volume of the module.

- **Prototype**

```
int ql_mic_gain_write(unsigned short vol, unsigned short mic_gain);
```

- **Parameter**

vol:
[In] Rear uplink call gains. Range: 0–65535.

mic_gain:
[In] Front uplink call gains. Range: 0–65535.

- **Return Value**

1	Set the uplink call gain successfully.
Other values	Failed to set the uplink call gain.

NOTE

A set of examples illustrating the use of the API is available in *example_qmic.c*. Call this API before a call, otherwise it cannot take effect immediately.

2.4.8. ql_mic_gain_read

This function reads the uplink call gain.

- **Prototype**

```
int ql_mic_gain_read(unsigned int *vol, unsigned int *mic_gain)
```

- **Parameter**

vol:
[Out] Rear uplink call gains. Range: 0–65535.

mic_gain:
[Out] Front uplink call gains. Range: 0–65535.

- **Return Value**

1 Read the uplink call gain successfully.
0 Failed to read the uplink call gain.

NOTE

A set of examples illustrating the use of the API is available in *example_qmic.c*.

2.4.9. ql_spk_gain_write

This function sets the downlink call gain to change the downlink call volume of the module.

- **Prototype**

```
int ql_spk_gain_write(unsigned short spk_gain);
```

- **Parameter**

spk_gain:
[In] The downlink call gain. Range: 0–65535.

- **Return Value**

1	Set the downlink call gain successfully.
Other values	Failed to set the downlink call gain.

NOTE

A set of examples illustrating the use of the API is available in *example_qspk.c*. Call this API before audio recording, otherwise it cannot take effect immediately.

2.4.10. ql_spk_gain_read

This function reads the downlink call gain of the module.

- **Prototype**

```
int ql_spk_gain_read(unsigned int *spk_gain);
```

- **Parameter**

spk_gain:
[Out] The downlink call gain. Range: 0–65535.

- **Return Value**

1	Read the downlink call gain successfully.
0	Failed to read the downlink call gain.

NOTE

A set of examples illustrating the use of the API is available in *example_qspk.c*.

2.4.11. ql_audloop_write

This function enables/disables the loopback.

- **Prototype**

```
int ql_audloop_write(unsigned int value);
```

- **Parameter**

value:
[In] Enables/disables loopback.
0 Disable
1 Enable

- **Return Value**

1 Enabled/disabled loopback successfully.
0 Failed to enable/disable loopback.

NOTE

A set of examples illustrating the use of the API is available in *example_audloop.c*.

2.4.12. ql_audloop_read

This function reads the current state of loopback.

- **Prototype**

```
int ql_audloop_read(void);
```

- **Parameter**

None.

- **Return Value**

1 Loopback enabled.
0 Loopback disabled.

NOTE

A set of examples illustrating the use of the API is available in *example_audloop.c*.

2.4.13. ql_sidet_write

This function sets the sidetone gain.

- **Prototype**

```
int ql_sidet_write(unsigned int value);
```

- **Parameter**

value:
[In] Sidetone gain. Range: 0–65535.

- **Return Value**

1	Set the sidetone gain successfully.
Other values	Failed to set the sidetone gain.

NOTE

A set of examples illustrating the use of the API is available in *example_sidet.c*.

2.4.14. ql_sidet_read

This function reads the sidetone gain.

- **Prototype**

```
int ql_sidet_read(void);
```

- **Parameter**

None.

- **Return Value**

The sidetone gain.

NOTE

A set of examples illustrating the use of the API is available in *example_sidet.c*.

2.5. API Use Examples

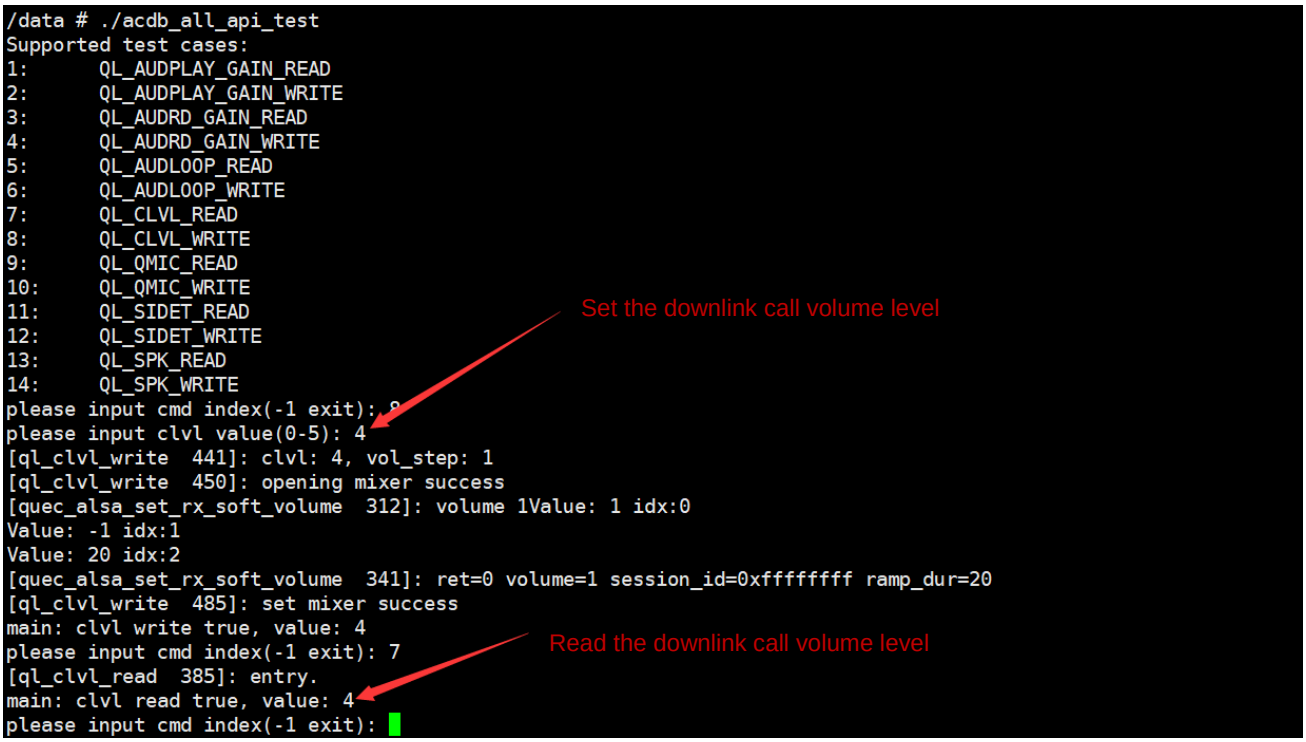
2.5.1. Set/Read Audio Playback Volume

```
/data # ./acdb_all_api_test
Supported test cases:
1:      QL_AUDPLAY_GAIN_READ
2:      QL_AUDPLAY_GAIN_WRITE
3:      QL_AUDRD_GAIN_READ
4:      QL_AUDRD_GAIN_WRITE
5:      QL_AUDLOOP_READ
6:      QL_AUDLOOP_WRITE
7:      QL_CLVL_READ
8:      QL_CLVL_WRITE
9:      QL_QMIC_READ
10:     QL_QMIC_WRITE
11:     QL_SIDET_READ
12:     QL_SIDET_WRITE
13:     QL_SPK_READ
14:     QL_SPK_WRITE
please input cmd index(-1 exit): 2
please input audplay_gain value(0-65535): 65500 <-- Set the audio playback volume
main: audplay gain write true, value: 65500
please input cmd index(-1 exit): 1
main: audplay gain read true, value: 65500 <-- Read the audio playback volume
please input cmd index(-1 exit):
```

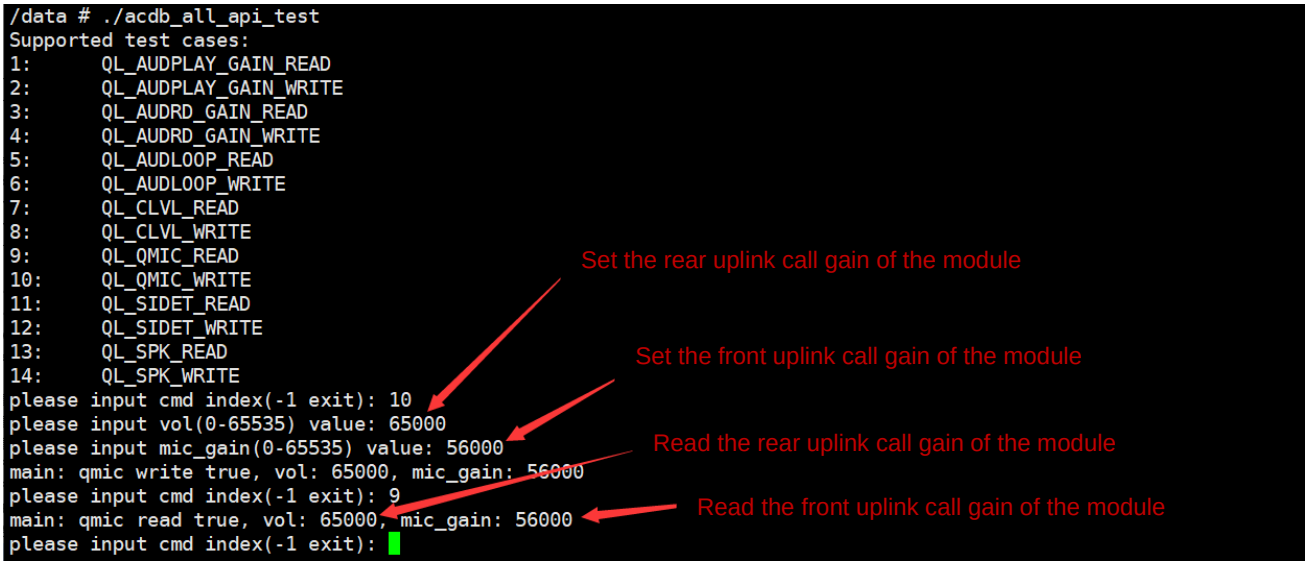
2.5.2. Set/Read Audio Recording Volume

```
/data # ./acdb_all_api_test
Supported test cases:
1:      QL_AUDPLAY_GAIN_READ
2:      QL_AUDPLAY_GAIN_WRITE
3:      QL_AUDRD_GAIN_READ
4:      QL_AUDRD_GAIN_WRITE
5:      QL_AUDLOOP_READ
6:      QL_AUDLOOP_WRITE
7:      QL_CLVL_READ
8:      QL_CLVL_WRITE
9:      QL_QMIC_READ
10:     QL_QMIC_WRITE
11:     QL_SIDET_READ
12:     QL_SIDET_WRITE
13:     QL_SPK_READ
14:     QL_SPK_WRITE
please input cmd index(-1 exit): 4
please input audrd_gain value(0-65535): 55660 <-- Set the audio recording volume
main: audrd gain write true, value: 55660
please input cmd index(-1 exit): 3
main: audrd gain read true, value: 55660 <-- Read the audio recording volume
please input cmd index(-1 exit):
```

2.5.3. Set/Read Call Volume Level



2.5.4. Set/Read Uplink Call Gains



2.5.5. Set/Read Downlink Call Gains

```
/data # ./acdb_all_api_test
Supported test cases:
1:      QL_AUDPLAY_GAIN_READ
2:      QL_AUDPLAY_GAIN_WRITE
3:      QL_AUDRD_GAIN_READ
4:      QL_AUDRD_GAIN_WRITE
5:      QL_AUDLOOP_READ
6:      QL_AUDLOOP_WRITE
7:      QL_CLVL_READ
8:      QL_CLVL_WRITE
9:      QL_QMIC_READ
10:     QL_QMIC_WRITE
11:     QL_SIDET_READ
12:     QL_SIDET_WRITE
13:     QL_SPK_READ
14:     QL_SPK_WRITE
please input cmd index(-1 exit): 14
please input spk gain value (0-65535): 55000
main: qspk write true, spk_gain: 55000
please input cmd index(-1 exit): 13
main: qspk read true, spk_gain: 55000
please input cmd index(-1 exit):
```

Set the downlink call gain of the module

Read the downlink call gain of the module

2.5.6. Enable Loopback

```
/data # ./acdb_all_api_test
Supported test cases:
1:      QL_AUDPLAY_GAIN_READ
2:      QL_AUDPLAY_GAIN_WRITE
3:      QL_AUDRD_GAIN_READ
4:      QL_AUDRD_GAIN_WRITE
5:      QL_AUDLOOP_READ
6:      QL_AUDLOOP_WRITE
7:      QL_CLVL_READ
8:      QL_CLVL_WRITE
9:      QL_QMIC_READ
10:     QL_QMIC_WRITE
11:     QL_SIDET_READ
12:     QL_SIDET_WRITE
13:     QL_SPK_READ
14:     QL_SPK_WRITE
please input cmd index(-1 exit): 6
please input audloop value(0-1): 1
[ql_audloop_write  270]: ql_audloop_write: == QAUDLOOP = 1
main: audloop enable
please input cmd index(-1 exit): 5
main: audloop enable
please input cmd index(-1 exit):
```

Enable the loopback

The current loopback status: enabled

2.5.7. Set/Read Sidetone Gain

```
/data # ./acdb_all_api_test
Supported test cases:
1:      QL_AUDPLAY_GAIN_READ
2:      QL_AUDPLAY_GAIN_WRITE
3:      QL_AUDRD_GAIN_READ
4:      QL_AUDRD_GAIN_WRITE
5:      QL_AUDLOOP_READ
6:      QL_AUDLOOP_WRITE
7:      QL_CLVL_READ
8:      QL_CLVL_WRITE
9:      QL_QMIC_READ
10:     QL_QMIC_WRITE
11:     QL_SIDET_READ
12:     QL_SIDET_WRITE
13:     QL_SPK_READ
14:     QL_SPK_WRITE
please input cmd index(-1 exit): 12
please input sidet value (0-65535): 65535 <-- Set the sidetone gain
quectel_clt_set_mixer_value, device: SEC_AUXPCM_RX Port Mixer SEC_AUX_PCM_UL_TX, value: 1
quectel_clt_set_mixer_value, set mixer: SEC_AUXPCM_RX Port Mixer SEC_AUX_PCM_UL_TX sucess
main: set sidetone: 65535
please input cmd index(-1 exit): 11
main: sidet value: 65535 <-- The sidetone gain is 65535
please input cmd index(-1 exit):
```

3 Appendix A References

Table 1: Related Document

SN	Document Name	Remark
[1]	Quectel_AG35_Series_QuecOpen_Quick_Start_Guide	Quick start guide for QuecOpen solution of AG35 series

Table 2: Terms and Abbreviations

Abbreviation	Description
API	Application Programming Interface
IoT	Internet of Things
SDK	Software Development Kit