WCDMA&LTE Audio Design Note

UMTS/HSPA/LTE Module Series

Rev. WCDMA&LTE_Audio_Design_Note_V1.2

Date: 2017-07-23

www.quectel.com

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

Quectel Wireless Solutions Co., Ltd.
Office 501, Building 13, No.99, Tianzhou Road, Shanghai, China, 200233
Tel: +86 21 5108 6236
Email: info@quectel.com

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

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/techsupport.aspx
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.

About the Document

History

Revision	Date	Author	Description
1.0	2016-12-02	Yeoman CHEN/ Jun WU	Initial
1.1	2016-12-29	Jun WU	1. Added some descriptions, notes and an example to explain the differences in audio setting and audio AT commands between EC2x and UCxx modules (Chapter 5 and Chapter 6) 2. Modified the descriptions of AT commands AT+QAUDCFG="digital/dlgain" and AT+QAUDCFG="innercodec/dlgain" (Chapter 6.10 and Chapter 6.11)
1.2	2017-07-23	Jerry YOU	3. Added more configuration about AT+QAUDCFG and others related audio AT commands.

Contents

Table Index

Figure Index

1 Introduction

The document is intended for engineers and customers who are using the audio interface (PCM interface) of Quectel modules. It introduces how to design and use the PCM interface in details in the following chapters.

This document is applicable to Quectel UCxx and EC2x modules. In this document, UCxx includes UC15 and UC20, and EC2x includes EC20, EC21, EC25 and EC20 R2.0.

For the audio design of Quectel UGxx (UG95 & UG96) modules, please refer to ***document [4]***.

2 PCM Characteristics

The PCM interfaces of Quectel UCxx and EC2x modules support the following working modes:

- Primary mode (short frame synchronization)
- Auxiliary mode (long frame synchronization)

These modules support 8-bit a-law and μ-law, and also 16-bit linear data formats. They can work as either the master or the slave in primary mode, and only the master in auxiliary mode.

The following tables show the pin definition and electrical characteristics of PCM interface.

Table 1: Pin Definition

Pin Name	I/O	Description
PCM_IN	DI	PCM data input
PCM_OUT	DO	PCM data output
PCM_SYNC	IO	PCM data frame sync signal, output as master and input as slave
PCM_CLK	IO	PCM data bit clock, output as master and input as slave
I2C_SCL	OD	I2C serial clock
I2C_SDA	OD	I2C serial data

Table 2: I/O Characteristics

Parameter	Description	Min	Max	Unit
V_{IL}	Low-level input voltage.	-0.3	0.35*VDD_EXT	V
V_{IH}	High-level input voltage.	0.65*VDD_EXT	VDD_EXT+0.3	V
V_{OL}	Low-level output voltage.	0	0.45	V
V_{OH}	High-level output voltage.	VDD_EXT-0.45	VDD_EXT	V

NOTE

VDD_EXT is the power source for modules' GPIO group.

In primary mode, the data is sampled on the falling edge of the PCM_CLK and transmitted on the rising edge. The PCM_SYNC falling edge represents the MSB. In this mode, PCM_CLK supports 128, 256, 512, 1024 and 2048KHz for different speech codecs, but UC15 supports 2048KHz only in primary mode.

The figure below shows the timing relationship of PCM interface with 8KHz PCM_SYNC and 2048KHz PCM_CLK in primary mode.

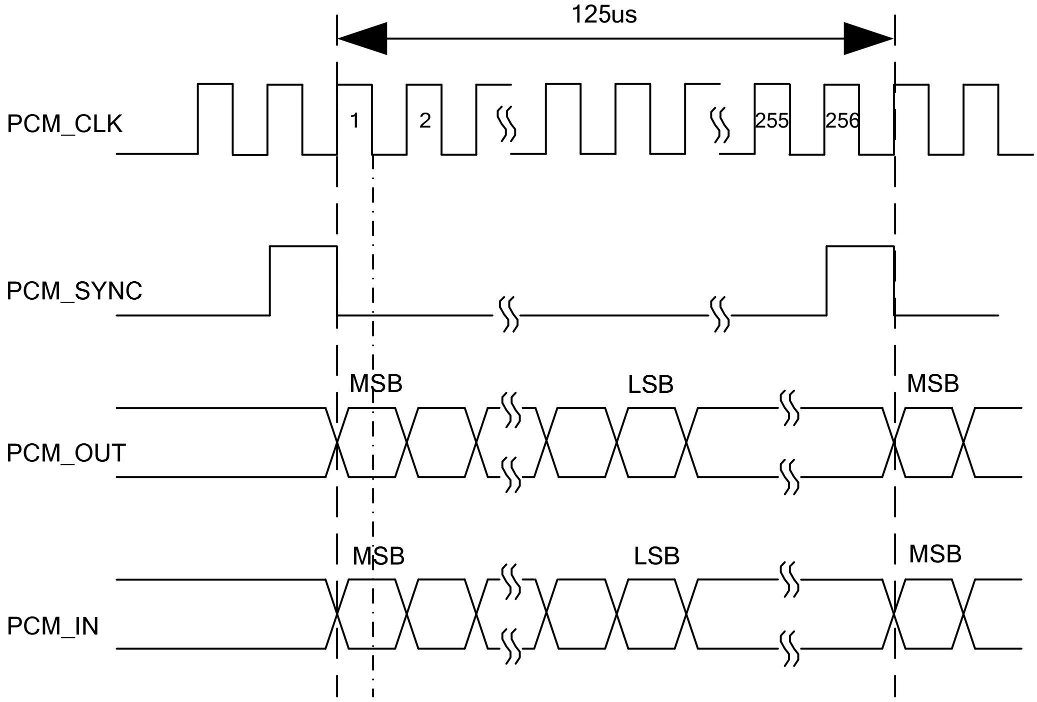

Figure 1: PCM Interface Timing in Primary Mode

In auxiliary mode, the data is also sampled on the falling edge of PCM_CLK and transmitted on the rising edge. But the PCM_SYNC rising edge represents the MSB. In this mode, the PCM interface only acts as the master with a 128KHz PCM_CLK and an 8KHz, 50% duty cycle PCM_SYNC.

The following figure shows the timing relationship of PCM interface in auxiliary mode.

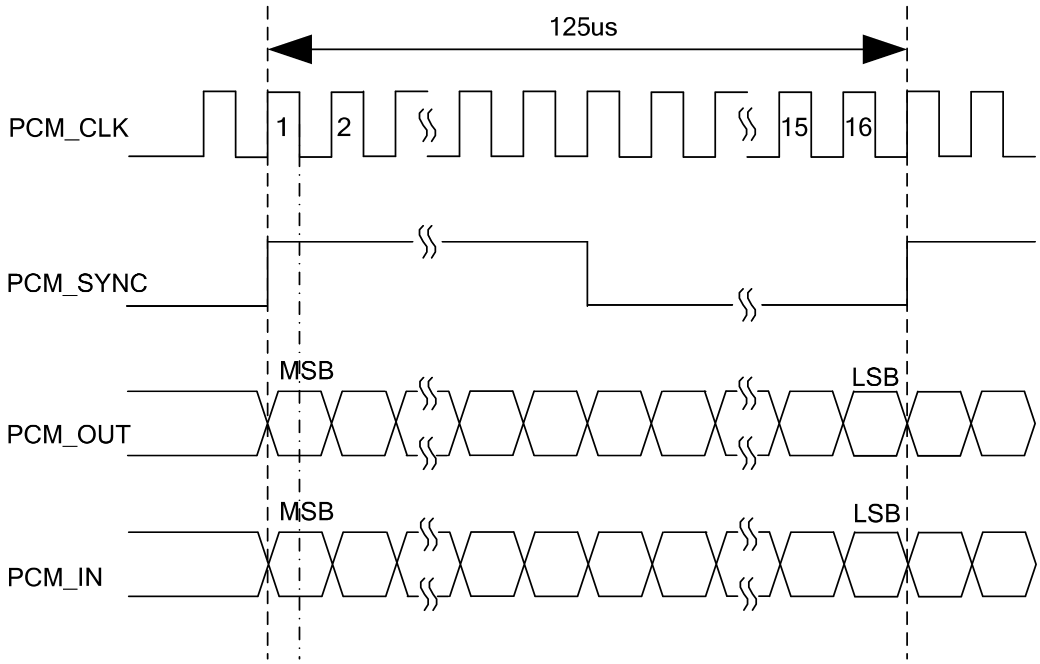

Figure 2: PCM Interface Timing in Auxiliary Mode

Both the clock and mode can be configured by AT commands, and the default configuration is the master mode using short frame synchronization format with 2048KHz PCM_CLK and 8KHz PCM_SYNC. Additionally, the module firmware has integrated the configuration on some popular audio codecs (such as NAU8814, ALC5616, etc.) which can be realized through the I2C interface. For more details, please refer to ***document [3]*** for the command **AT+QDAI**.

For peripherals, the module works in PCM master mode, and the following figure illustrates the connection among the peripheral, module, and the codec.

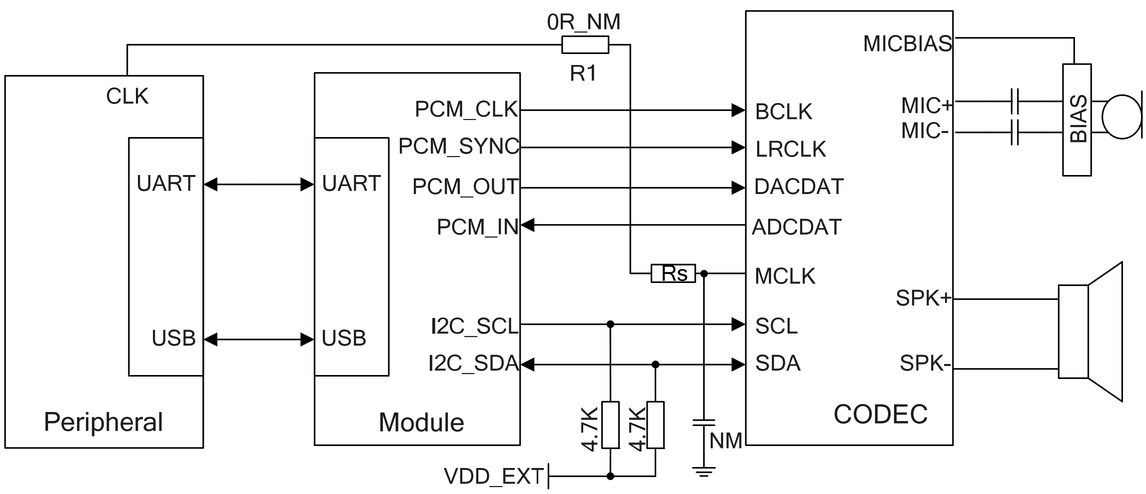

Figure 3: Peripheral, Module and Codec Connection Diagram

NOTES

1. The communication between module and peripheral can be realized through the UART or USB interface.
2. The MCLK of codec can be powered by the peripheral or an external XTAL, if it is needed in the audio codec.
3. It is recommended to reserve an RC (R=22Ω, C=22pF) circuit on the PCM lines, especially for PCM_CLK.
4. EC20's 8-bit a-law and μ-law is still under development.

3 Audio Circuit Design

Audio codecs NAU8814, ALC5616, MAX9860 and TLV320AIC3104 have been approved for PCM application. The following sub-chapters will show some audio circuit designs with these codecs.

3.1. PCM Design with NAU8814

The following figure shows the PCM application with NAU8814 from *Nuvoton* (http://www.nuvoton.com), and an I2C interface is available in the design for codec configuration.

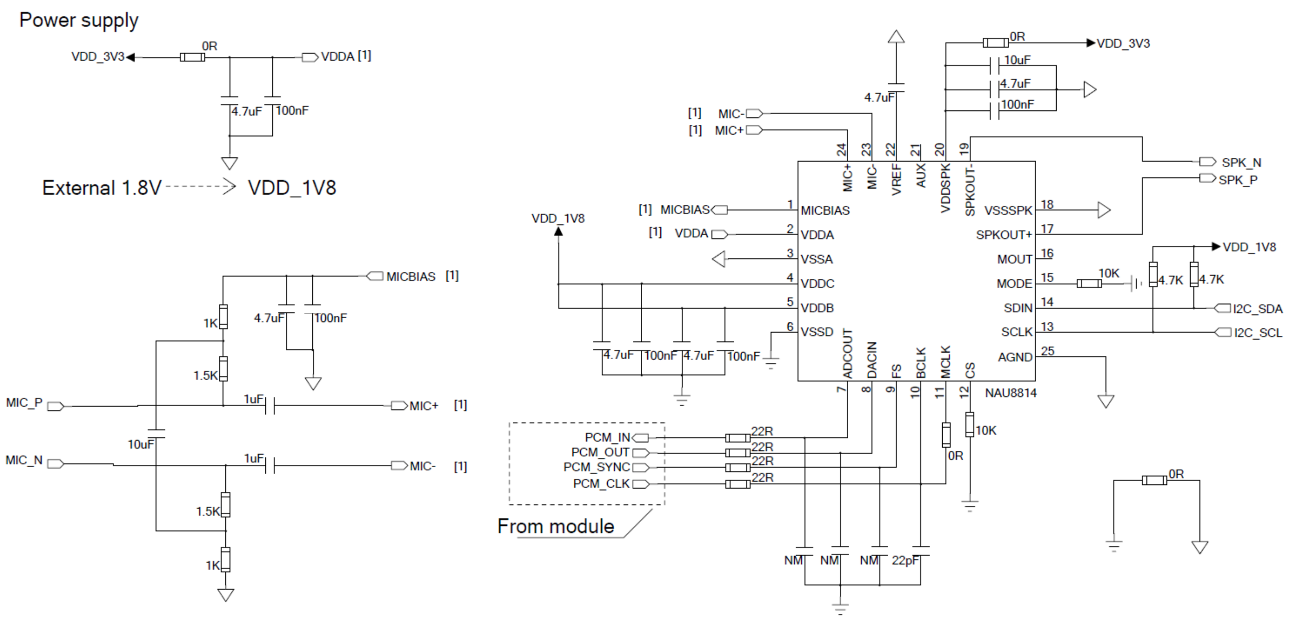

Figure 4: PCM Application with NAU8814

NOTES

1. The RC filter circuit (R=22 ohm, C=22pF) needs to be installed on PCM_CLK line.
2. VDD_3V3 needs to be powered externally.
3. Set **AT+QDAI=2** to choose NAU8814. For more details please refer to ***document [3]***.

3.2. PCM Design with ALC5616

The following figure shows the PCM application with ALC5616 from *Realtek* (http://www.realtek.com.tw), and an I2C interface is available in the design for codec configuration.

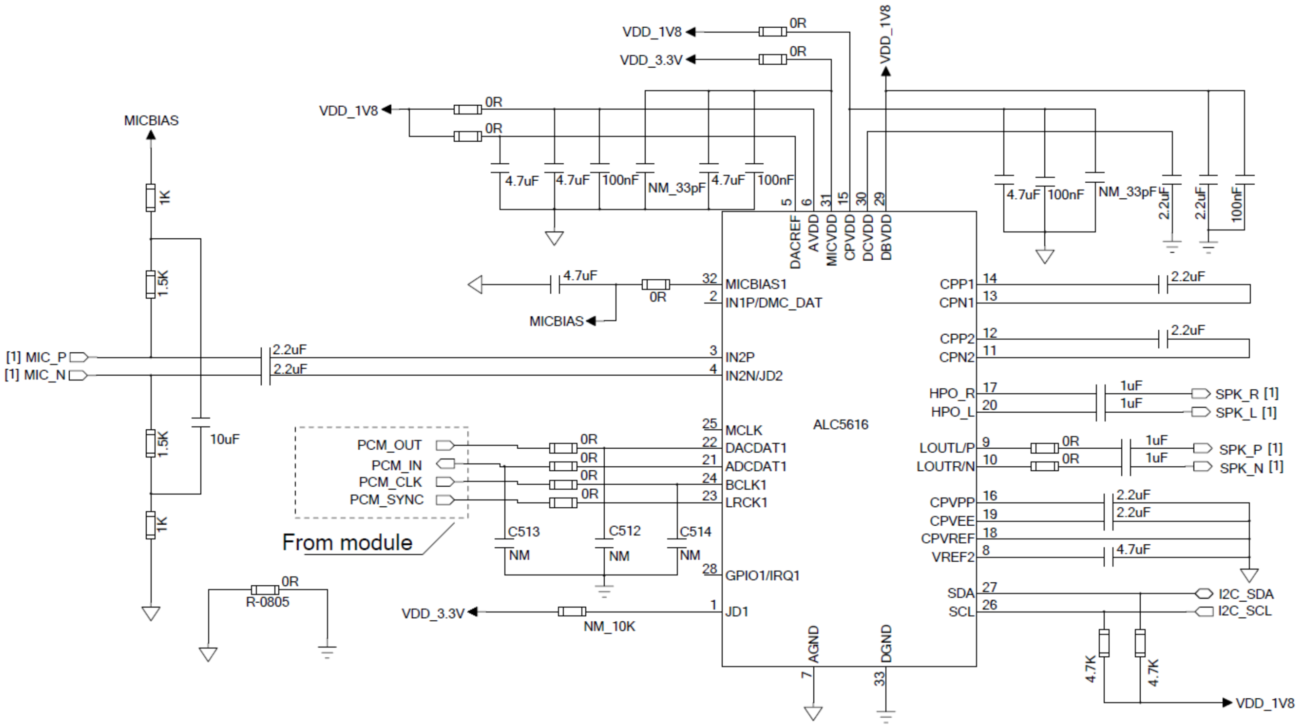

Figure 5: PCM Application with ALC5616

NOTES

1. VDD_3V3 and VDD_1V8 need to be powered externally. Please pay attention to the power-on sequence of ALC5616. For more details, please refer to its datasheet.
2. The RC filter circuit (R=22 ohm, C=22pF) should be installed on PCM_CLK line.
3. Set **AT+QDAI=3** to choose ALC5616. For more details please refer to ***document [3]***.

3.3. PCM Design with MAX9860

The following figure shows the PCM application with MAX9860 from *Maxim* (http://www.maxim-ic.com), and an I2C interface is available in the design for codec configuration.

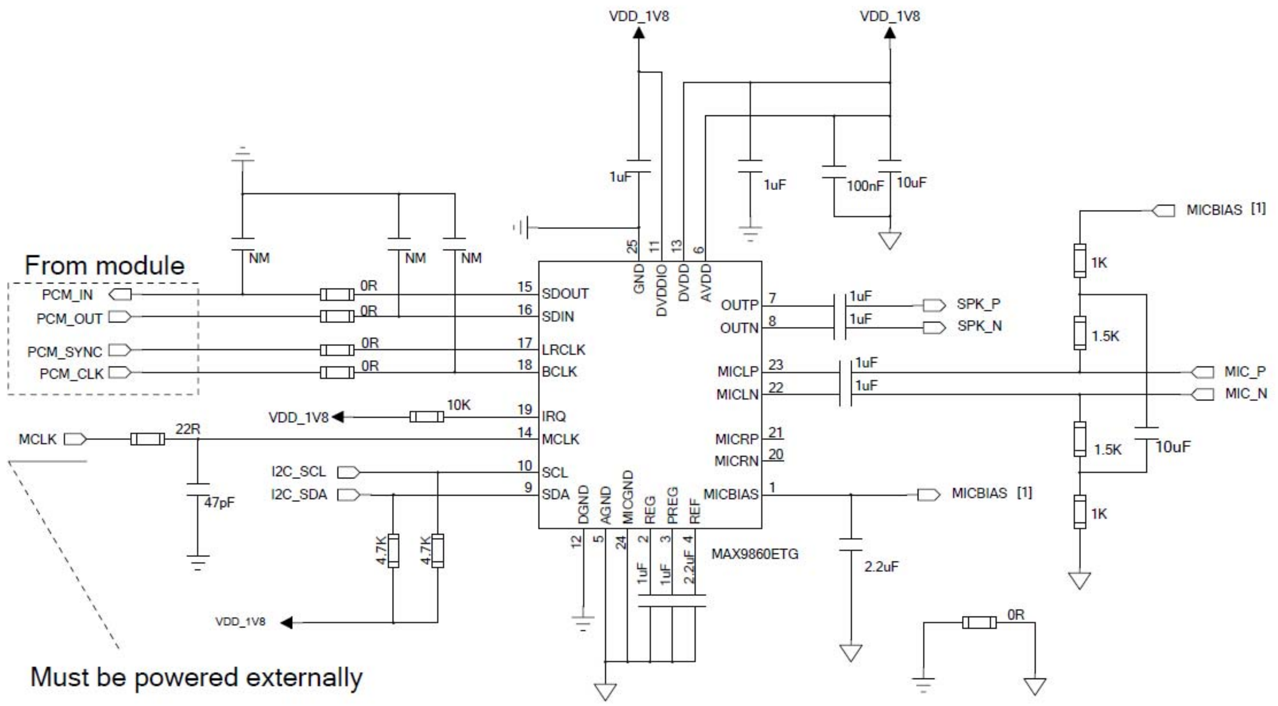

Figure 6: PCM Application with MAX9860

NOTES

1. The RC filter circuit (R=22 ohm, C=22pF) should be installed on PCM_CLK line.
2. Set **AT+QDAI=4** to choose MAX9860. For more details please refer to ***document [3]***. The firmware of EC2x modules has not integrated the configuration on MAX9860 by now.
3. MCLK should be powered externally.

3.4. PCM Design with TLV320AIC3104

The following figure shows the PCM application with TLV320AIC3104 from *Texas Instrument* (http://www.ti.com.cn), and an I2C interface is available in the design for codec configuration.

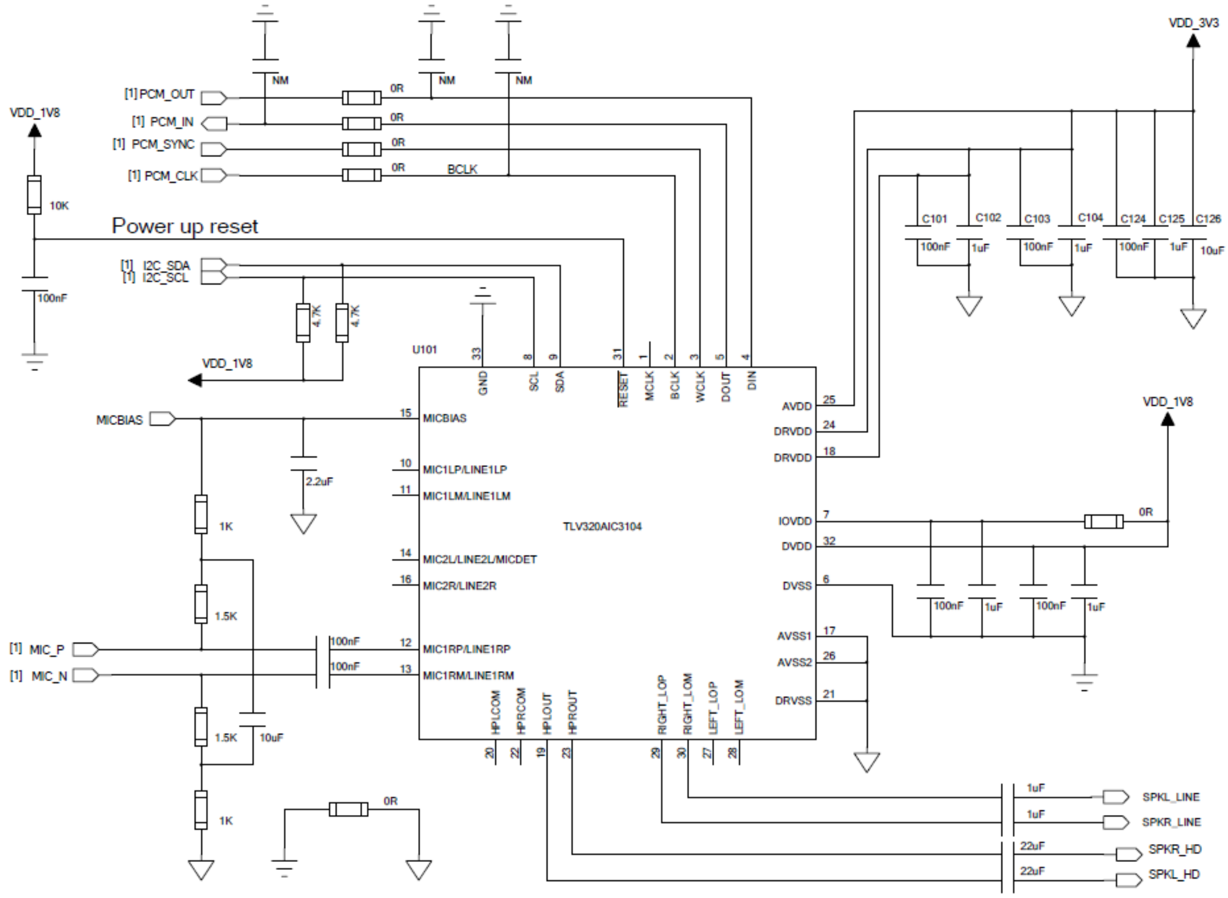

Figure 7: PCM Application with TLV320AIC3104

NOTES

1. The RC filter circuit (R=22ohm, C=22pF) should be installed on PCM_CLK line.
2. VDD_3V3 and VDD_1V8 need to be powered externally. Please pay attention to the power-on sequence of TLV320AIC3104. For more details, please refer to its datasheet.
3. Set **AT+QDAI=5** to choose TLV320AIC3104. For more details please refer to ***document [3]***.

3.5. PCM Design with MCU

UCxx and EC2x modules can work in either PCM master mode or slave mode to communicate with MCU.

- Primary mode (short frame synchronization, works as both master and slave)
- Auxiliary mode (long frame synchronization, works as master only)

They provide a 1.8V PCM interface, but 2.6V PCM interface for UC15. A level translator should be used if customers' application is equipped with a 3.3V UART interface. The level translator TXS0104 provided by *Texas Instrument* is recommended.

3.5.1. Module Works in PCM Master Mode

In PCM master mode, UCxx and EC2x modules support both short and long frame synchronization. The data is sampled on the falling edge of the PCM_CLK and transmitted on the rising edge. The PCM_SYNC falling edge represents the MSB. PCM_CLK supports 128, 256, 512, 1024 and 2048KHz, and PCM_SYNC supports 8KHz.

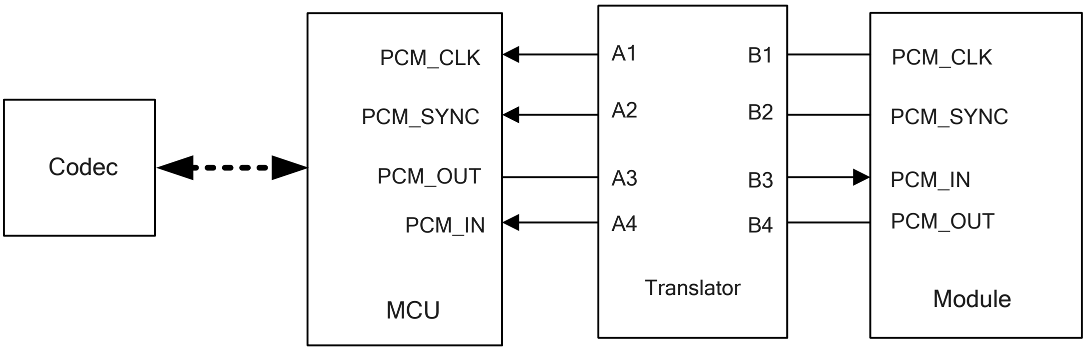

Figure 8: Reference Design When Module Works in PCM Master Mode

The modules support 8-bit a-law and μ-law, and also 16-bit linear data formats. The PCM interface can be configured via **AT+QDAI** command, and an example is shown below.

Example

AT+QDAI=1,0,0,4,0	//PCM master mode, short frame synchronization, PCM_CLK=2.048MHz,
.	16bit data
OK	

3.5.2. Module Works in PCM Slave Mode

When the module works in PCM slave mode, the data is sampled on the falling edge of the PCM_CLK and transmitted on the rising edge. The PCM_SYNC falling edge represents the MSB. PCM_CLK supports 128, 256, 512, 1024 and 2048KHz, and PCM_SYNC supports 8KHz.

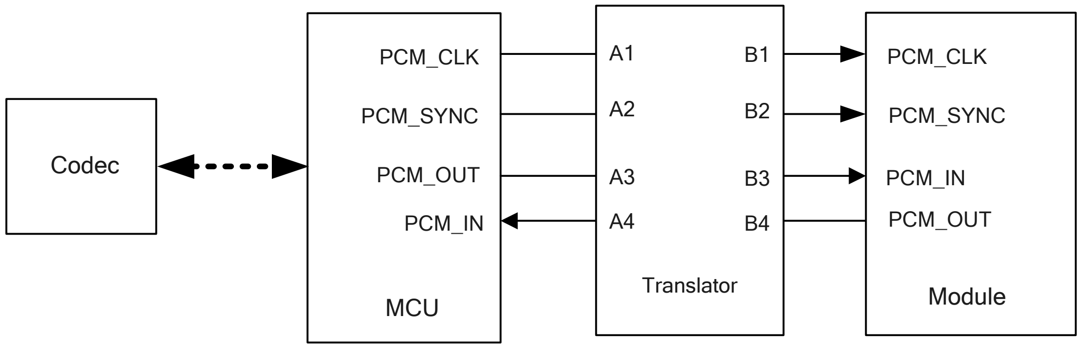

Figure 9: Reference Design When Module Works in PCM Slave Mode

The modules support 16-bit linear data formats in PCM slave mode. The PCM interface can be configured via **AT+QDAI** command, and an example is shown below.

Example

```
AT+QDAI=1,1,0,4,0            //PCM slave mode, short frame synchronization, PCM_CLK=2.048MHz,
                              16bit data.
OK
```

NOTES

1. It is recommended to reserve an RC (R=22 ohm, C=22pF) circuit on the PCM lines, especially for PCM_CLK.
2. UC15 and EC20 do not support PCM slave mode presently.

3.6. Microphone Interface Design

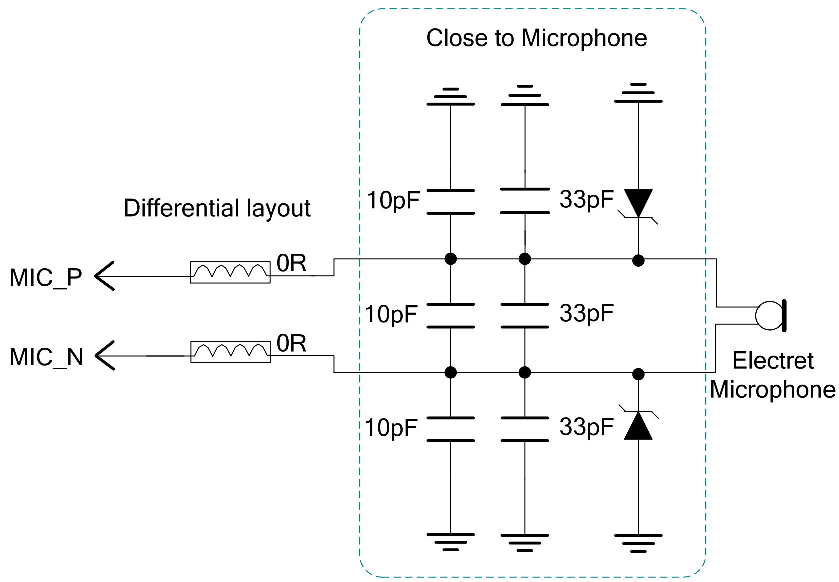

Figure 10: Microphone Interface Design

3.7. Receiver Interface Design

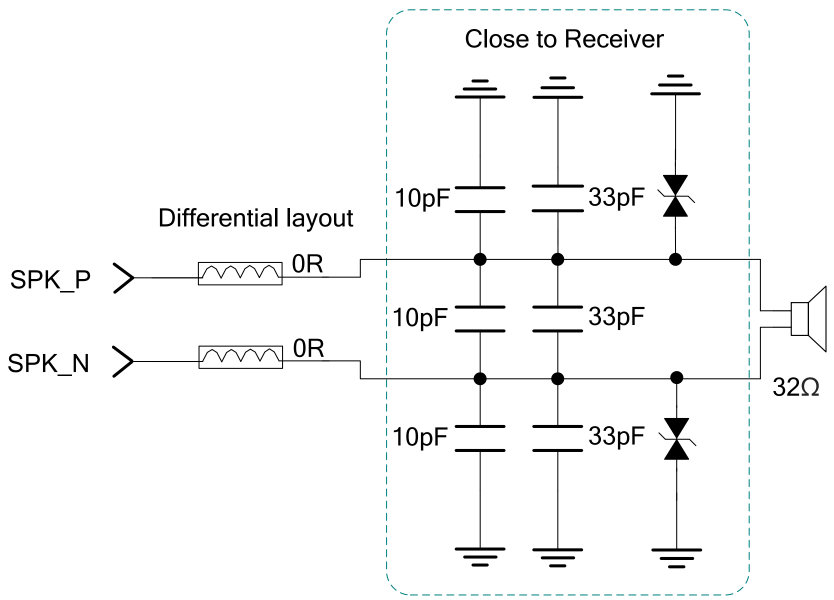

Figure 11: Receiver Interface Design

3.8. Earphone Interface Design

The following figure shows the single-ended application for earphone interface design.

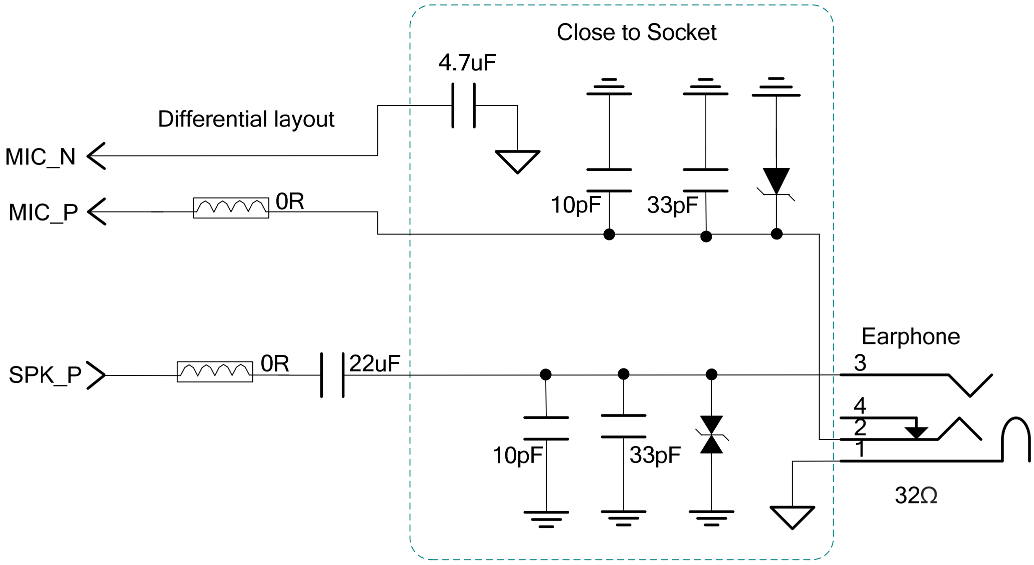

Figure 12: Earphone Interface Design

3.9. Speaker Interface Design

If an 8Ω speaker is applied, it is recommended to add an audio amplifier.

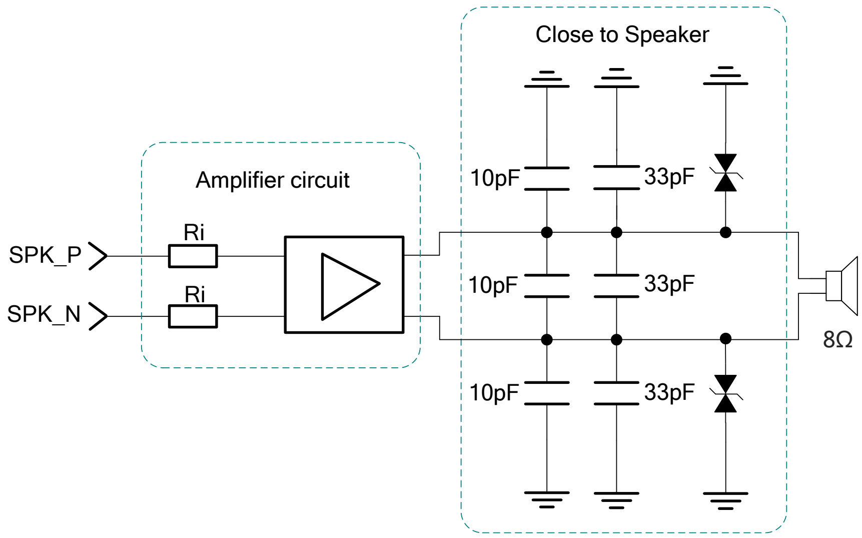

Figure 13: Speaker Interface Design (with Amplifier Circuit)

4 Design Considerations

4.1. Power Supply for PCM Codecs

Different PCM codecs have different supply voltages. Hence, it is recommended to power a PCM codec with a dedicated LDO, and do not share this power supply with other circuits.

4.2. Suggestions for Audio Circuit Layout

Power supply ripple, unbalanced ground and RF burst have negative effects to audio circuit layout. In order to avoid them, the layout of MICP_PCM/MICN_PCM and SPKP_PCM/SPKN_PCM must meet the rule of differential signal. Moreover, these two pairs of signals should be separated from each other through ground shielding on not only upper and lower layers but also right and left sides, to avoid echo from SPK signal to MIC signal. The following figure shows an example.

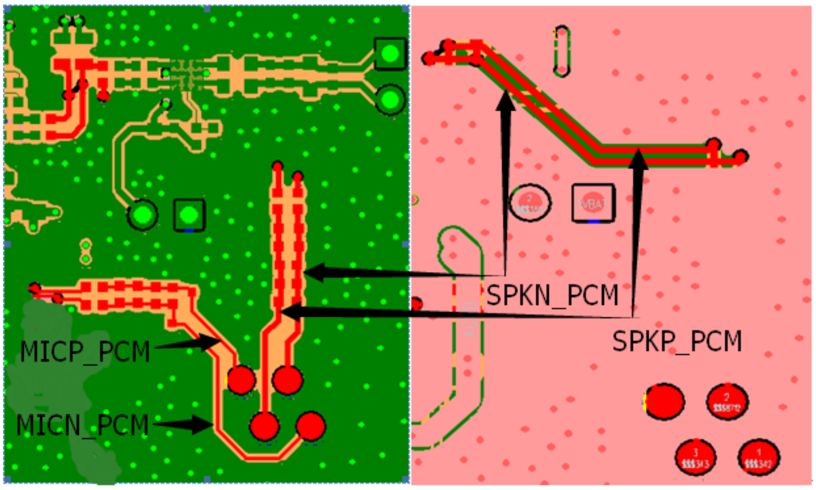

Figure 14: Audio Channel Layout Example

4.3. TDD Noise Solution

It is important to avoid or reduce TDD noise in audio circuit design and layout. This sub-chapter provides some suggestions for TDD noise reduction or elimination.

1. Different capacitors have their own self-resonant frequencies due to different fabrication processes and materials. Generally, a 33pF capacitor (0603 package) is recommended to be used for filtering GSM900 interference and 10pF capacitor (0603 package) for filtering DCS1800 interference on the power and analog audio signal lines. It is strongly recommended to add the two capacitors (10pF and 33pF) near the electret-microphone in handset and hands-free applications. These two capacitors could largely suppress coupling TDD noise from RF interference.

2. The capacitors should be placed close to audio components or audio interface, and the layout must be short.

3. Ground shielding area should be as large as possible to reduce the ground impedance and improve grounding performance.

4. Reduce power supply voltage ripple, especially the power supply in audio circuits. This can be achieved through using a wide wire for the layout between power source (like adapter interface, battery connector, or LDO output pin) and audio power supply. Good antenna matching is also important for reducing power ripple.

5. The filtering capacitors and ESD protection devices should be connected to the main digital ground. Separate the analog ground from the digital ground while routing, and then connect them at a single point on the PCB with a 0 ohm resistor, so as to reduce digital interference and background noise.

6. The antenna must be stay away from audio components and the layout of audio circuit. Keep a distance of at least 5cm between the antenna and the microphone.

7. The layout of power supply must be stay away from the audio components and layout, and cannot be parallel.

4.4. Suggestions for Mechanical Design

It is important to consider how to suppress echo in the equipment with hands-free function or in an application where the microphone and the speaker are very close to each other.

The mechanical structure design has significant impact on echo issue. If it is not properly designed, the echo suppressing arithmetic in software will not be able to make up the echo issue caused by bad mechanical structure, and even force to redesign.

The echo issue could be generated from several paths as shown in the figure below.

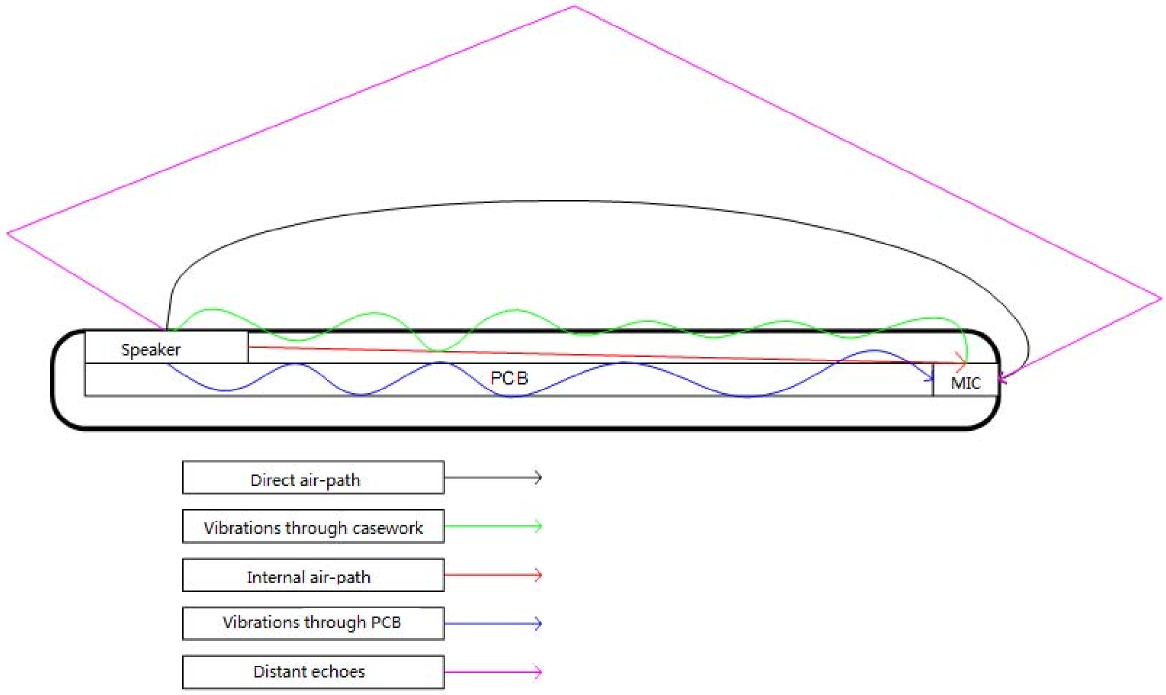

Figure 15: Five Echo Paths

In these five paths, the internal air-path and direct air-path are the primary influential factors. Other three factors (vibrations through casework, vibrations through PCB, distant echoes) are the secondary ones.

How to deal with the echo generated from internal air-path?

Separating microphone from internal space of chassis by foam or rubber ring can effectively suppress the inner echo interference. The figure below shows a recommended design for microphone socket.

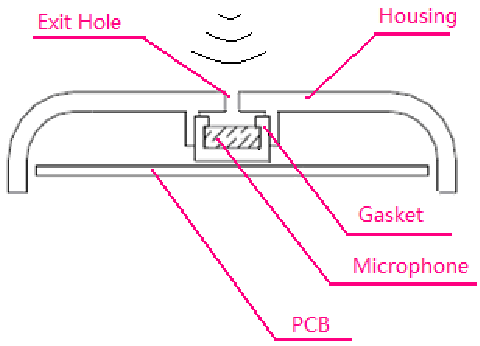

Figure 16: Recommended Microphone Socket Design

NOTE

The best installation way of microphone socket is to encase microphone by silicone cover except for the front cavity, and design a cylindrical hole whose center is the exit hole of the chassis. Also, make sure the microphone with silicone cover just fit the cylindrical hole, so as to only let voice enter into microphone from the exit hole rather than the leak of chassis interior. Certain air space room should be reserved in the front cavity of microphone as it is necessary for good microphone performance.

The following figure shows a recommended design for speaker socket.

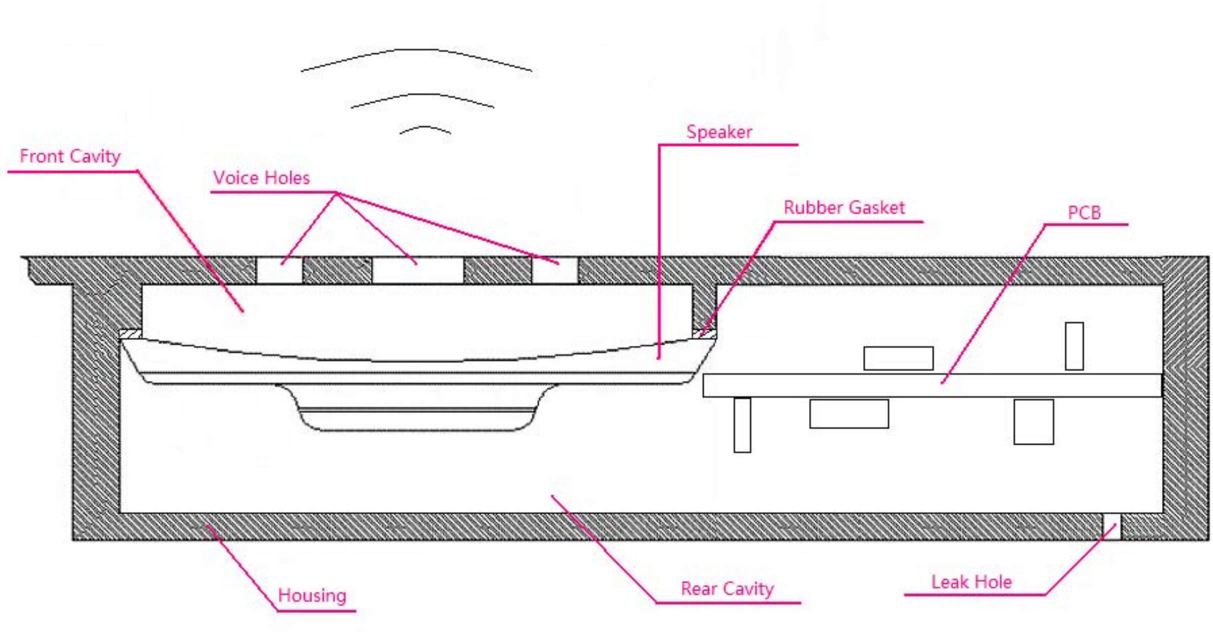

Figure 17: Speaker Socket Design

A good way to suppress the internal echo path is to seal the rear cavity of receiver, which is usually expensive. The rear cavity of receiver and speaker is important for good voice quality. A sealed rear cavity with sufficient space could produce a good voice output. An 8Ω speaker is often big and thus it is difficult to give an independently sealed rear cavity for it. However, sealing microphone socket in chassis is always useful for internal echo suppressing. The microphone socket and speaker socket should be designed as far as possible. If there is any unavoidable leak hole, keep it far away from the microphone. If the leak hole is close to the microphone, the voice coming from the hole could be picked up by microphone, and then leads to echo at the far end. If the leak hole is close to the speaker, the output voice quality could be aggravated at a certain extent.

4.5. Components of Speaker

Speakers and receivers with higher sensitivity, flatter frequency response, less THD and impedance of 32Ω (receiver), 16Ω (receiver) or 8Ω (speaker) are recommended. These technical data are often shown in the datasheet of the speaker and receiver. For a speaker, its frequency response and THD performance can be tested by a speaker test system. The speaker frequency response and THD are shown in the following two figures.

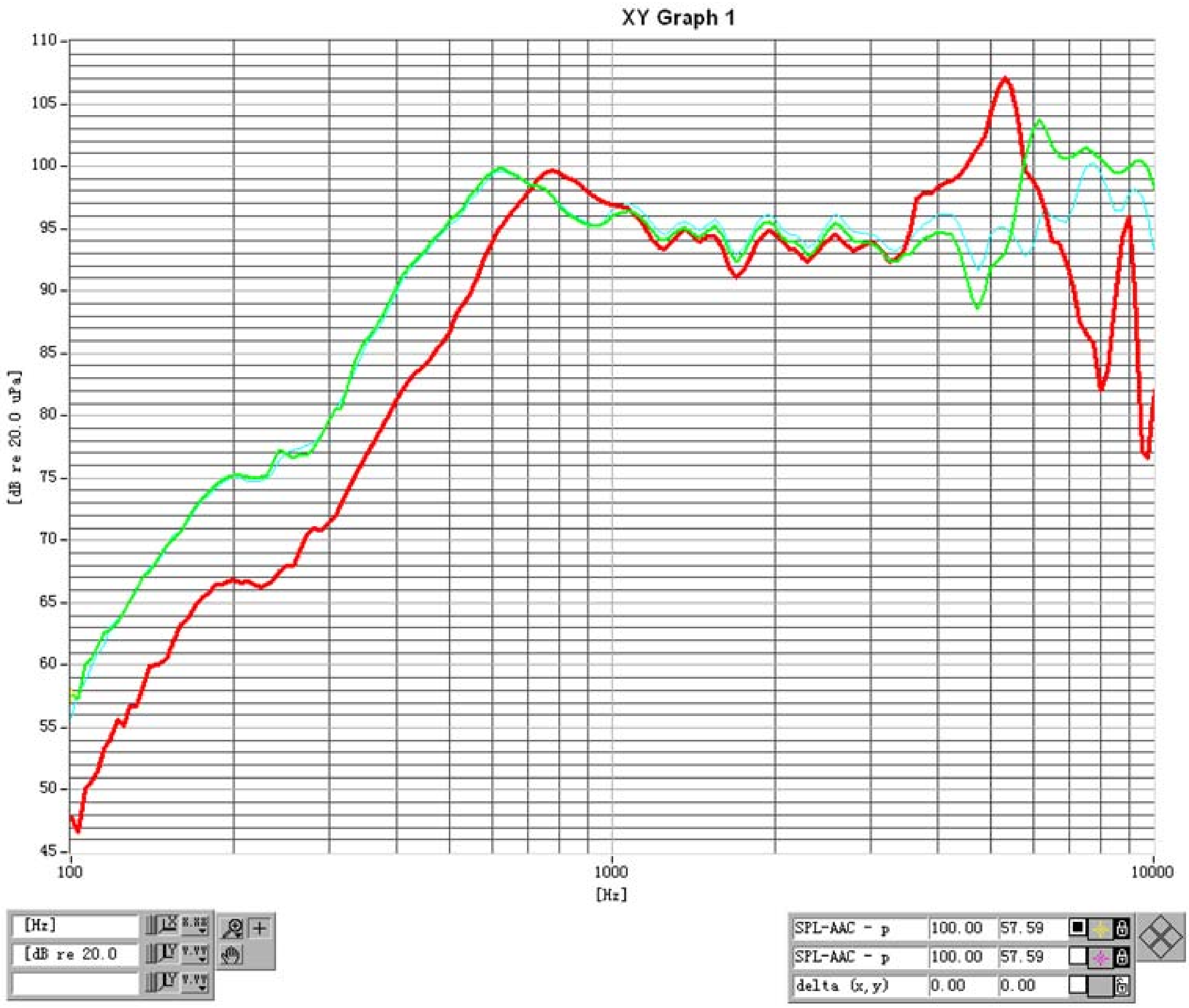

Figure 18: Speaker Frequency Response

NOTE

1, Horizontal axis: frequency
2, Longitudinal axis: loudness (dB)

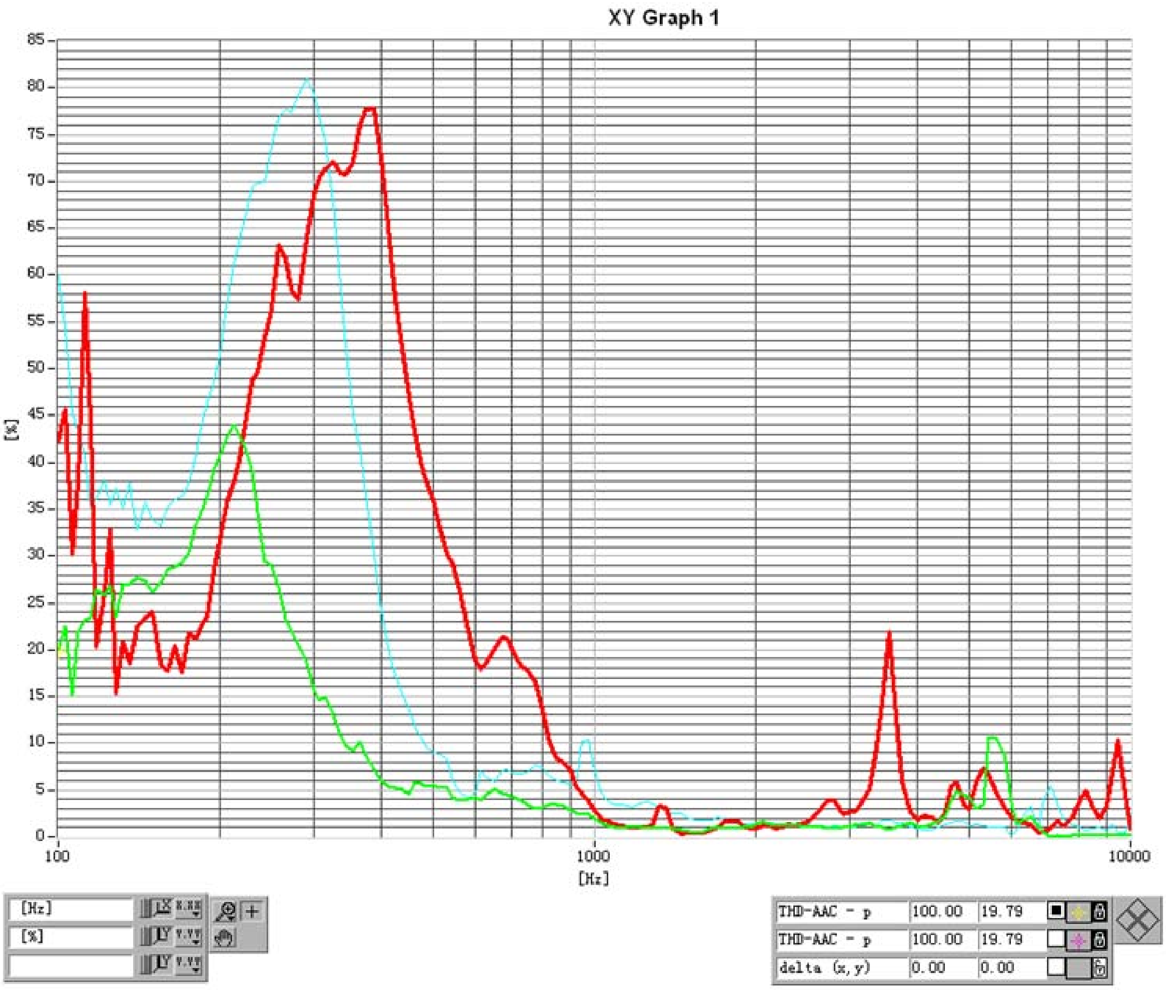

Figure 19: Speaker THD

NOTES

1. Horizontal axis: frequency
   Longitudinal axis: distortion (%)
2. In the above two figures, the three colors represent products of three different vendors. Through comprehensive comparison, the green one performs the best, the blue one is the second, and the red one is the worst.

4.6. Components of Microphone

It is recommended to use an electret microphone with a sensitivity of -42±3dB/Pa @2V (not less than -44±3dB) and impedance of 2.2Kohm. If RF TDD noise is detected at the microphone, please contact the microphone vendor for products with better RF suppression capability. Furthermore, the microphone channel circuit can be optimized to decrease TDD noise at microphone side.

5 Audio Modes

There are three kinds of common audio modes: handset, headset and handsfree. **AT+QAUDMOD** is used to choose audio modes. For each mode, there are some default settings, such as **AT+CLVL**, **AT+QSIDET**, and **AT+QMIC**.

If the audio performance is not good and **AT+QAUDMOD** command has been set, the following steps can be applied to tune the audio settings:

Step 1: Set **AT+CLVL** to tune downlink volume.
Step 2: Set **AT+QMIC** to tune uplink volume.
Step 3: If the downlink volume is a little low or high, and **AT+CLVL** has been set, please use **AT+QRXGAIN** (EC2x modules)/**AT+QAUDCFG="digital/dlgain"** (UCxx modules) to tune it. For uplink volume, **AT+QMIC** can be used.
Step 4: If the recommended codec is used, customers can use **AT+QAUDCFG** to tune the volume of recommended codec.
Step 5: Set **AT+QSIDET** to tune side tone.
Step 6: If there is echo or noise left, please use **AT+QEEC** to tune EEC.

NOTES

1. Please do not set the volume too high, otherwise it will influence EEC and NR.
2. **AT+QAUDMOD** must be set before using other commands.

The following sub-chapters provide some examples which show how to tune the audio settings in different audio modes.

5.1. Speech Call in Handset Mode

The default mode is handset mode.

```
AT+QAUDMOD=0                      //<mode>=0 means handset mode is selected.

OK
AT+CLVL=3                         //Tuning downlink volume

OK
```

```
AT+QMIC= 25905,14567            //Tuning uplink volume

OK
AT+QSIDET=1298                  //Tuning side tone

OK
```

5.2. Speech Call in Headset Mode

Set **AT+QAUDMOD=1** to select headset mode.

```
AT+QAUDMOD=1                    //<mode>=1 means headset mode is selected.

OK
AT+CLVL=2                       //Tuning downlink volume

OK
AT+QMIC=25905,14567             //Tuning uplink volume

OK
AT+QSIDET=0                     //Tuning side tone

OK
```

5.3. Speech Call in Hands free Mode

Set **AT+QAUDMOD=2** to select hands free mode. This mode is always used for vehicle-mounted devices.

```
AT+QAUDMOD=2                    //<mode>=2 means hands free mode is selected.

OK
AT+CLVL=4                       //Tuning downlink volume

OK
AT+QMIC=25905,14567             //Tuning uplink volume

OK
AT+QSIDET=0                     //Tuning side tone

OK
```

6 Audio AT Commands

This chapter will introduce the common audio AT commands. For more other AT commands details, refer to ***document [3]***.

6.1. AT+QDAI Digital Audio Interface Configuration

The command is used to configure the digital audio interface. When **<io>**=1, customers can define the PCM mode (master/slave mode) by themselves. When **<io>**=2 and the external codec chip linked with PCM interface is the NAU8814 model which is configurable through the I2C, the module can be used directly and set by the default configurations. When **<io>**=3 and the external codec chip linked with PCM interface is the ALC5616 model which is configurable through the I2C, the module can also be used directly and set by the default configurations.

AT+QDAI Digital Audio Interface Configuration	
Test Command **AT+QDAI=?**	Response **+QDAI: (1-3),(0,1),(0,1),(0-5),(0-2)** **OK**
Read Command **AT+QDAI?**	Response **+QDAI: <io>[,<mode>,<fsync>,<clock>,<format>]** **OK**
Write Command **AT+QDAI=<io>[,<mode>,<fsync>,<clock>[,<format>]]**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<io>	1	Digital PCM output (customer-defined)
	2	Analog output (for the default audio codec NAU8814).
	3	Analog output (for the default audio codec ALC5616).
<mode>	0	Master mode

	1	Slave mode
<fsync>	0	Primary mode (short-frame synchronization)
	1	Auxiliary mode (long-frame synchronization)
<clock>	0	128K
	1	256K
	2	512K
	3	1024K
	4	2048K
	5	4096K
<format>	0	16-bit linear
	1	8-bit a-law
	2	8-bit μ-law

NOTES

1. Configuration of **<io>** will be saved to NV immediately by default.
2. The PCM interface supports both master and slave mode under short frame synchronization, and only master mode under long frame synchronization.
3. When short frame synchronization and master mode are selected, PCM_CLK supports 256K~4096K clock frequency. If long frame synchronization and master mode are selected, only 128K clock frequency is supported.
4. If slave mode is selected, then PCM_CLK and PCM_SYNC must be provided for modules.
5. When NAU8814 or ALC5616 is selected, please do not input any other parameters.
6. A-law & μ-law are not supported by EC2x modules.

Example

```
AT+QDAI=?               //Query the range.
+QDAI: (1-3),(0,1),(0,1),(0-5),(0-2)

OK
AT+QDAI?                //Query the current interface configuration.
+QDAI: 1,0,0,4,0

OK
AT+QDAI=1,1,0,4,1       //Set PCM interface to slave short-frame synchronization mode, PCM format 8-bit
                          a-law.
OK
AT+QDAI=2               //Select NAU8814.
OK
AT+QDAI=3               //Select ALC5616.
OK
```

6.2. AT+QIIC I2C Read and Write

The command is used to configure the codec via I2C interface.

AT+QIIC I2C Read and Write	
Test command **AT+QIIC=?**	Response **+QIIC: (0-1),(0-FF),<0-FF>,<1-2>[,<0-FFFF>]** **OK**
Read command **AT+QIIC?**	Response **+QIIC:** **OK**
Execute command **AT+QIIC**	Response **ERROR**
Write command **AT+QIIC=<rw>,<device>,<addr>,<bytes>[,<value>]**	Response **OK** Response **+QIIC: <value>** **OK**

Parameter

<rw>	0	Write command
	1	Read command
<device>	0-0xFF	Device address
<addr>	0-0xFF	Register address
<bytes>	1-2	Read/write bytes
<value>	0-0xFFFF	Data value

NOTES

1. Only 7bit device address is supported presently.
2. The parameters are hexadecimal, and there are some differences between UCxx and EC2x modules in command format:
   - The device address should be shifted one bit to the left for UCxx modules, and this is not required for EC2x modules;
   - There is a need to add a prefix “0x” to the parameters of EC2x modules.

Example

AT+QIIC=0,0x18,0x00,1,0x00	//This command is used for EC2x modules, and the module writes 1byte data 0x00 to device. The device address is 0x18, and the register address is 0x00.
OK	
AT+QIIC=0,30,00,1,00	//This command is used for UCxx modules, and the module writes 1byte data 0x00 to device. The device address is 0x18, and the register address is 0x00.
OK	

6.3. AT+CLVL Loudspeaker Volume Level Selection

The command is used to select the volume level of the internal loudspeaker of Quectel UCxx and EC2x modules.

AT+CLVL Loudspeaker Volume Level Selection	
Test Command **AT+CLVL=?**	Response **+CLVL:** (list of supported **<level>**s) **OK**
Read Command **AT+CLVL?**	Response **+CLVL: <level>** **OK**
Write Command **AT+CLVL=<level>**	Response **OK** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms
Reference 3GPP TS 27.007	

Parameter

<level>	For UCxx modules: Integer type, value (0-3-7) with manufacturer specific range (smallest value represents the lowest sound level).
<level>	For EC2x modules:

	Integer type, value (0-3-5) with manufacturer specific range (smallest value represents the lowest sound level).

NOTE

1, The parameter can't be saved after powered down the module.

6.4. AT+CRSL Set Ring Tone Volume

AT+CRSL can be used to set the volume of ring tone.

AT+CRSL Set Ring Tone Volume	
Test Command **AT+CRSL=?**	Response **+CRSL:** (list of supported **<volume>**s) **OK**
Read Command **AT+CRSL?**	Response **+CRSL: <volume>** **OK**
Write Command **AT+CRSL=<level>**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<volume>	Numeric type, indicates the configured volume of ring tone Range: 0-7, default: 3

NOTE

1, These parameters can be saved after powered down the module.

6.5. AT+QMIC Set Uplink Gains of MIC

The command is used to set the uplink gains of microphone.

AT+QMIC Set Uplink Gains of MIC	
Test Command **AT+QMIC=?**	Response **+QMIC: (1),(0-65535),(0-65535)** **OK**
Read Command **AT+QMIC?**	Response **+QMIC: <txgain1>,<txgain2>,<txdgain>** **OK**
Write Command **AT+QMIC=<txgain1>,<txgain2>[,<txdgain>]**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<txgain1>	Numeric type, reserved, set to 1. This parameter is invalid for EC2x modules.
<txgain2>	Numeric type, indicates uplink codec gain, range: 0-65535. Default value might be different in different audio modes. This parameter is invalid for UC20 module.
<txdgain>	Numeric type, indicates uplink digital gain, range: 0-65535. Default value might be different in different audio modes.

NOTE

1, These parameters can't be saved after powered down.

6.6. AT+QRXGAIN Set Downlink (RX) Gain

This command is used to set RX gains to change downlink volume.

AT+QRXAGIN Set Downlink (RX) Gains	
Test Command **AT+QRXGAIN=?**	Response **+QRXGAIN: (0-65535)** **OK**
Read Command **AT+QRXGAIN?**	Response **+QRXGAIN: <rxgain>** **OK**
Write Command **AT+QRXGAIN=<rxgain>**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<rxgain>	Numeric type, which indicates downlink codec gain. The range is 0-65535. The default value might be different in different audio modes**.**

NOTE

1, This parameter can't be saved after powered down the module.

Example

AT+QRXGAIN=?	//Test Command.
+QRXGAIN: (0-65535)	
OK	
AT+QRXGAIN?	// Get gain values, the default value might be different in different audio modes.
+QRXGAIN: 36864	
OK	

AT+QRXGAIN=8192	//Set codec gain to 8192.
OK	
AT+QRXGAIN?	//Get gain values.
+QRXGAIN: 8192 OK	

6.7. AT+QTTS Play text

AT+QTTS is used to play text.

AT+QTTS Play text	
Test Command **AT+QTTS=?**	Response **+QTTS: (0-2),<text>** **OK**
Read Command **AT+QTTS?**	Response **+QTTS: <status>** **OK**
Write Command **AT+QTTS=<mode>,<text>**	Response **OK** **+CME ERROR: <err>** Play completed report: **+QTTS: 0**

Parameter

<mode>	Numeric type, range: 0-2, start/stop play, and indicates <text> format. 0 Stop play, **<text>**can be ignored. 1 Start play, **<text>** is UCS2 encoding. 2 Start play, **<text>** is character(s), normal is ASCII, chinese is GBK encoding.
<text>	String type, text need play, according to different **<mode>** to adopt different encoding, max length is 548 bytes.
<status>	Numeric type, the status of the TTS player 0 Idle 1 busy

NOTE

1. This parameter can't be saved after powered down.
2. Supports for playing TTS with AT+QTTS or AT+QWTTS during a non-call process..
3. The TTS will be terminated when calling, but the caller will not terminate.
4. Supports both TTS and audplay playback, but does not support replacement playback.

Example

```
AT+QTTS=?                                   //query ranges.
+QTTS: (0-2),<text>

OK
AT+QTTS=1,"6B228FCE4F7F752879FB8FDC6A215757"//Play a UCS2 string
OK

+QTTS: 0                                    //when play completed, reporting
AT+QTTS=2,"hello world,你好"                // play a ASCII string.
OK

+QTTS: 0                                    //when play completed, reporting
AT+QTTS=0                                   //when playing, use this command to stop it
OK
```

6.8. AT+QTTSETUP Set Parameters for TTS

AT+QTTSETUP is used to set the TTS speed and adjust the volume.

AT+QTTSETUP Set Parameters for TTS	
Test Command **AT+QTTSETUP=?**	Response **+QTTSETUP: (1,2),(1,2),(-32768~32767)** **OK**
Read Command **AT+QTTSETUP?**	Response **OK**
Write Command **AT+QTTSETUP=<mode>,<ID>[,<value>]**	Response **OK** **ERROR**

	If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

<mode>	Set or read the value of parameters 1 Write 2 Read
<ID>	ID 1 Set the speed 2 Set the volume
<value>	Parameter value. TTS speed range is -32768~32767, the normal speed is 0, the default is 0. TTS volume range is -32768~32767, the default is -30000. If it is in read mode, please omit the value.

NOTE

1, This parameter can’t be saved after powered down.

Example

```
AT+QTTSETUP=?                    //Query the range.
+QTTSETUP: (1,2),(1,2),( -32768-32767)

OK
AT+QTTSETUP=1,2,0        //Set the volume.
OK
```

6.9. AT+QWTTS Play Text Or Send To Far-end

AT+QWTTS is used to play text or play text to far-end.

AT+QWTTS Play Text Or Send To Far-end	
Test Command **AT+QWTTS=?**	Response **+QWTTS: (0,1),(0,1),(0-2),<text>** **OK**
Read Command **AT+QWTTS?**	Response **+QWTTS: <status>** **OK**
Write Command **AT+QWTTS=<ulmute>,<dlmute>,<mode>,<text>**	Response **OK** **+CME ERROR: <err>** Playing completed, report: **+QWTTS: 0**

Parameter

<ulmute>	Numeric type, mute uplink or not 0 mute 1 not mute
<dlmute>	Numeric type, mute downlink or not 0 mute 1 not mute
<mode>	Numeric type, range: 0-2, start/stop playing, indicates **<text>** format 0 Stop playing, **<text>** can be ignored 1 Start playing, **<text>** is UCS2 encoding 2 Start playing, **<text>** is character(s), normal is ASCII, Chinese is GBK encoding
<text>	String type, text to be played, according to different **<mode>** to adopt different encoding, max length is 543 bytes
<status>	Numeric type, the status of the TTS player 0 Idle 1 Busy
<err>	901 Audio unknown error 902 Audio invalid parameters 903 Audio operation not supported 904 Audio device busy

NOTE

1. This parameter can't be saved after powered down the module.
2. +QWTTS:4111 means TTS is interrupted by call event.
3. In the non-call state uplink volume or uplink and downlink the volume is muted, playing tts will be reported **+CME : 903.**
4. In the call state, set the uplink and downlink volume are muted, play tts will report **+CME: 903.**
5. When setting up and down mute parameters out of range or Modem parameters do not match the text format or Modem parameters are out of range, will report **+CME: 902.**
6. Support for playing txt characters The maximum number of bytes is 543 bytes.
7. When playing empty characters, it will be reported **+ CME: 902.**

Example

```
AT+QWTTS=?                                          //Query ranges.
+QWTTS: (0,1),(0,1), (0-2),<text>

OK
AT+QWTTS=1,1,1,"6B228FCE4F7F752879FB8FDC6A215757"   //Play an UCS2 string and send to
                                                    far-end.
OK

+QWTTS: 0                                           //When playing completed.


AT+QWTTS=1,0,2,"hello world,你好"                   //Play an ASCII string to far-end.
OK

+QWTTS: 0                                           //When playing completed.

AT+QWTTS=1,0,0                                      //When playing, use this command to stop it.
OK
```

6.10. AT+QAUDRD Record Media File

AT+QAUDRD is used to record the uplink or downlink speech during voice call or record sound from local microphone in idle state and save it to files.

AT+QAUDRD Record Media File	
Test Command **AT+QAUDRD=?**	Response **+QAUDRD: (0,1),<file_name>,(13),(0,1)** **OK**
Read Command **AT+QAUDRD?**	Response **+QAUDRD: <state>** **OK**
Write Command **AT+QAUDRD=<control>[,<file_name>[,<format>[,<dlink>]]]**	Response **OK** **+CME ERROR: <err>**

Parameter

<state>	0 Module is not in recording 1 Module is in recording
<control>	0 Stop the recording 1 Start to record
<file_name>	Name of the file to record.
<format>	Format of the file. 13 WAV_PCM16
<dlink>	Record down-link sound. <u>0</u> record up-link sound 1 record down-link sound
<err>	901 Audio unknown error 902 Audio invalid parameters 903 Audio operation not supported 904 Audio device busy

NOTES

1. <file_name> parameter is the path to save the recording file, the default saved in the "/ data / ufs" directory.
2. Recording format support "13", suffix wav audio files.
3. If the recording file's name and format is same with an existed file or an unknown error occur, module will report **+QAUDRIND: 0,1**.
4. If current recording is interrupted by other audio task, Module will report URC **+QAUDRIND: 0,6.**
5. If there is no space to record, module will report URC **+QAUDRIND: 0,3**.
6. Supports recording of uplink and downlink audio data, but don't support simultaneous recording.
7. The format is incorrectly reported error, the file name format is incorrect, not yet processed.

Table 1: +QAUDRIND Code

<code>	Meaning
0	Reserved
1	Unknown error
3	No space to record
6	Interrupted by other audio task

Example

```
AT+QAUDRD=1,"A.wav",13,0      // Record the uplink sound, format wav, store it in UFS.
OK
AT+QAUDRD=0                   //stop recording.
OK

AT+QAUDRD=1,"B.wav",13,1      // Record the downlink sound, format wav, store it in UFS.
OK

AT+QAUDRD=0                   //Stop recording.
OK
```

6.11. AT+QAUDPLAY Play Media File

AT+QAUDPLAY is used to play local media file.

AT+QAUDPLAY Play Media File	
Test Command **AT+QAUDPLAY=?**	Response **+QAUDPLAY: <file_name>,(0,1)** **OK**
Read Command **AT+QAUDPLAY?**	Response **+QAUDPLAY: <state>** **OK**
Write Command **AT+QAUDPLAY=<file_name>,<repeat>**	Response **OK** **+CME ERROR: <err>** Play completed report: **+QAUDPLAY: 0**

Parameter

<state>	0 Module is not in playing 1 Module is in playing
<file_name>	Name of the file to play.
<repeat>	Repeating play or not 0 play only once 1 repeat
<err>	901 Audio unknown error 902 Audio invalid parameters 903 Audio operation not supported 904 Audio device busy

NOTES

1. <file_name> includes file path, file name and file suffix. The default play path is the "/ data/ufs" directory.
2. If there is an unknown error occurred, module will report URC **+QAUDPIND: 0,1.**
3. If the playback is terminated by **AT + QAUDSTOP** or incoming call is terminated, reported **+**

QAUDPIND: 0,6.

4. Only support wav format playback, without wav header file can not play, while reported **+ CME: 902**.
5. The call will stop playing, and if the ringing tone is on, the ringtone will be played,does not support the replacement play
6. Parameter type out of range, error **+ CME: 902**, playback does not exist file error **+ CME: 903.**

Table 1: +QAUDPIND Code

<code>	Meaning
0	Reserved
1	Unknown error
6	Interrupted by other audio task

Example

AT+QAUDPLAY=”A.mp3”,0 //Playing a mp3 file which is stored in UFS only once.
OK

+QAUDPLAY: 0

AT+QAUDPLAY=”A.mp3”,1 //Playing a mp3 file which is stored in UFS and repeat it.
OK

+QAUDPLAY: 0

6.12. AT+QAUDSTOP Stop Media File Play

AT+QAUDSTOP is used to stop a playing media file.

AT+QAUDSTOP Stop Media File Play	
Test Command **AT+QAUDSTOP=?**	Response **OK**
Write Command **AT+QAUDSTOP**	Response **OK** **+CME ERROR: <err>**

Parameter

<err>	901	Audio unknown error
	902	Audio invalid parameters
	903	Audio operation not supported
	904	Audio device busy

6.13. AT+QTONEDET Enable Or Disable DTMF Detection

AT+QTONEDET is used to enable or disable DTMF detection. When you enable this function, DTMF tones sent by other side will be detected, and report DTMF tones on the serial port which you assigned.

AT+ QTONEDET Enable Or Disable DTMF Detection	
Test Command **AT+QTONEDET=?**	Response **+QTONEDET: (0,1)** **OK**
Read Command **AT+QTONEDET?**	Response **+QTONEDET: <enable>** **OK**
Write Command **AT+QTONEDET=<enable>**	Response **OK** **ERROR**
Maximum Response Time	300ms
Reference	

Parameter

<enable>	Enable/disable DTMF detection <u>0</u> Disable 1 Enable

NOTES

1. Settings will take effect immediately. When you reset the module, settings will revert to the default values.
2. DTMF characters - ASCII:

DTMF	ASCII	DTMF	ASCII
0	48	8	56
1	49	9	57
2	50	A	65
3	51	B	66
4	52	C	67
5	53	D	68
6	54	*	42
7	55	#	35

6.14. AT+QPCMV Enable to Transfer PCM Data via USB Or UART Port

AT+QPCMV is used to transfer PCM data through USB/UART port. When you dial a call, the sound from the opposite side will be decoded as PCM data and output from PCM port. When you input PCM data to the port, the data will be transmitted to the opposite side through the network. Once the call is over, the port will stop outputting, and the input data will be invalid. PCM data format is 8K, 16bit linear.

AT+ QPCMV Enable to Transfer PCM Data through USB/UART Port	
Test Command **AT+QPCMV=?**	Response **+QPCMV: (0,1),(0,1)** **OK**
Read Command **AT+QPCMV?**	Response **+QPCMV: <enable>,<port>** **OK**
Write Command **AT+QPCMV=<enable>[,<port>]**	Response **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms
URC	**+QPCMV: 0,0** **+QPCMV: 1,0**
Reference	

Parameter

<enable>	If this function is enabled when you dial a call, the sound from the opposite side sound will be output from PCM port. When you input PCM data to the port, the data will be transmitted to the opposite side. 0 Disable 1 Enable
<port>	PCM port 0 USB NMEA port

NOTES

1. Ensure the current port is not used for other functions when configuring <port>. In addition, when using USB NMEA port, you should configure the port via AT+QGPSCFG="outport","none" to prevent it from being used by GNSS.
2. The settings will take effect immediately. Parameters will not be saved and they will be recovered to default value after rebooting the module.
3. If the modem output the URC +QPCMV: 0,0, it means that the modem is busy, and can't receive more PCM data, please stop send PCM data to the modem. If the modem out the URC +QPCMV: 1,0, it means that the modem is ready to receive more PCM data.
4. Currently only supports 8K, 16Bit Liner data, does not support 16k liner data.
5. Does not support turn on or turn off PCMV during calls, only supports non-call to turn on or off PCMV.

Example

AT+QPCMV=1,0 //Enable to transfer pcm data via USB NMEA port..

OK

6.15. AT+QPSND Play Wave File

AT+QPSND is used to play local wave file.

AT+QPSND Play Wave File	
Test Command **AT+QPSND=?**	Response **+QPSND: (0,1),<file_name>,(0,1),(0,1),(0,1)** **OK**
Read Command	Response

AT+QPSND?	**+QPSND: <state>** **OK**
Write Command **AT+QPSND=<control>,<file_name>,<repeat>[,<ulmute>[,<dlmute>]]**	Response **OK** **+CME ERROR: <err>** Playing completed report: **+QPSND: 0**

Parameter

<state>	0	Module is not playing
	1	Module is playing
<control>	0	Stop playing
	1	Start playing
<file_name>	Name of the file to be played	
<repeat>	Repeat play or not	
	0	Play only once
	1	Repeat play
<ulmute>	Numeric type, mute uplink or not	
	0	Mute
	1	Not mute
<dlmute>	Numeric type, mute downlink or not	
	0	Mute
	1	Not mute
<err>	901	Audio unknown error
	902	Audio invalid parameters
	903	Audio operation not supported
	904	Audio device busy

NOTES

1. **<file_name>** includes file path, file name and file suffix. The default play path is the "/ data/ufs" directory.
2. We only support 8K liner, mono wave format.

Example

```
AT+QPSND=1,“A.wav”,0                  //Play a wave file which is stored in UFS.
OK
```

```
+QPSND: 0

AT+QPSND=1,"A.wav",0,1          //Play a wave file to far-end when a call is ongoing.
OK

+QPSND: 0
```

6.16. AT+QLDTMF Play Local DTMF

AT+QLDTMF is used to play a DTMF string, maximum length is 20 characters. You can use AT+QLDTMF to stop it.

AT+QLDTMF Play local DTMF	
Test Command **AT+QLDTMF=?**	Response **+QLDTMF: (1-1000),(0-9,*,#,A-G)** **OK**
Write Command **AT+QLDTMF=<n>,<DTMF_string>[,<y>]**	Response **OK** **+CME ERROR: <err>** After play completed report: **+QLDTMF: 5**
Execute Command **AT+QLDTMF**	Response **OK**
Maximum Response Time	300ms

Parameter

<n>	Numeric type, indicates every DTMF's on time and mute time. Ranges from 0 to 1000, the unit is 1/100 second when <y> is set to 1,or 1/10 second when <y> is not set.
<DTMF_string>	String type, max 20 DTMFs, separated by comma. DTMF: 0-9,*,#,A-G.
<err>	901 Audio unknown error 902 Audio invalid parameters 903 Audio operation not supported 904 Audio device busy

NOTE

1, This parameter can’t be saved after powered down the module.

Example

```
AT+QLDTMF=?                  //Query range.
+QLDTMF: (1-1000),(0-9,*,#,A-G)

OK
AT+QLDTMF=2,“A,B,1,2,#”      //Play A,B,1,2,#, on time & mute time is 200ms.

OK
AT+QLDTMF                    //Stop playing.

OK
```

6.17. AT+QLTONE Play a Local Customized Tone

AT+QLTONE is used to play a customized tone, use **<period_on>** to indicate play time and **<period_off>** to indicate mute time, and **<duration>** to indicate total time.

AT+QLTONE Play a Local Customized Tone	
Test Command **AT+QLTONE=?**	Response **+QLTONE: (0,1),(100-4000),(0-1000),(0-1000),(0-15300000)** **OK**
Write Command **AT+QLTONE=<mode>[,<frequency>,<period_on>,<period_off>,<duration>]**	Response **OK** **+CME ERROR: <err>** After playing completed report: **+QLTONE: 0**
Maximum Response Time	300ms

Parameter

<mode>	0	Stop play
	1	Start play
<frequency>	Tone’s frequency, unit: Hz, ranges: 100-4000Hz	

<period_on>	tone's on time, unit ms, ranges: 0-1000ms
<period_off>	Tone's mute time, unit: ms, ranges: 0-1000ms
<duration>	Tone's total time, unit: ms, ranges: 0-15300000ms
<err>	901 Audio unknown error 902 Audio invalid parameters 903 Audio operation not supported 904 Audio device busy

NOTES

1, This parameter can't be saved after powered down the module.

Example

```
AT+QLTONE=?                         //Query range.
+QLTONE: (0,1),(100-4000),(0-1000),(0-1000),(0-15300000)

OK
AT+QLTONE=1,1000,200,300,3000       //Play a 1000Hz tone, on time is 200ms, mute time is 300ms,
                                    //total time is 3000ms.
OK

+QLTONE:0
AT+QLTONE=0                         //Stop playing.
OK
```

6.18. AT+QRXIIR Set The Parameters Of the Rx-side IIR Filter

AT+QRXIIR is designed to implement the RX Infrared Impulse Response (IIR) filter function, primarily for voice and audio to meet the frequency response requirements of the handset and to improve voice / audio quality.

AT+ QRXIIR	**Set The Parameters Of The Rx-side IIR Filter**
Test Command **AT+QRXIIR=?**	Response **+QRXIIR:(0,1),(1-5),<pregain>,<filtercoeffs>,<numshiftfactor>** **OK**

Read Command **AT+QRXIIR?**	Response **+QRXIIR:<enable>,<N>,<pregain>,<filtercoeffs>,<numshiftfactor>** **OK**
Write Command **AT+QRXIIR=<enable>,<N>,<pregain>,<filtercoeffs>,<numshiftfactor>**	Response **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms
Reference	

Parameter

<enable> This parameter is used to enable or disable the rx-side filter function.
0 Disable
1 Enable
< N > 1-5 Number of filter stages.
< pregain > 0-0xFFFF Sets the gain before the filter.
< filtercoeffs > The coefficient of the biquad filter (a1 a2 b0 b1 b2).
< numshiftfactor > Number of shift factors.

NOTES

1. For narrowband voice signals, a sampling frequency of fs = 8 kHz and N = 3.
1. For wideband voice signals, a sampling frequency of fs = 16 kHz and N = 5.
2. For audio signals, a sampling frequency of fs = 48 kHz and N = 4.
3. <pregain> parameter fixed byte is 2, < filtercoeffs > parameter fixed byte is 4, < numshiftfactor > parameter fixed byte is 2, such as the parameters are set to 1, need the following settings: 0x0001,0x00000001,0x0001.

Example

```
AT+QRXIIR=?                                                  // query range.
+QRXIIR: (0,1),(1-5),<pregain>,<filtercoeffs>,<numshiftfactor>

OK
```

```
AT+QRXIIR=1,2,0x1234,"0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678","0x1234,0x1234"      //set N=2.

OK
```

6.19. AT+QTXIIR Set The Parameters Of the Tx-side IIR Filter

AT+QTXIIR is designed to implement the TX Infrared Impulse Response (IIR) filter function, primarily for voice and audio to meet the frequency response requirements of the handset and to improve voice / audio quality.

AT+ QTXIIR	Set The Parameters Of The Tx-side IIR Filter
Test Command **AT+QTXIIR=?**	Response **+QTXIIR:(0,1),(1-5),<pregain>,<filtercoeffs>,<numshiftfactor>** **OK**
Read Command **AT+QTXIIR?**	Response **+QTXIIR:<enable>,<N>,<pregain>,<filtercoeffs>,<numshiftfactor>** **OK**
Write Command **AT+QTXIIR=<enable>,<N>,<pregain>,<filtercoeffs>,<numshiftfactor>**	Response **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms
Reference	

Parameter

<enable> This parameter is used to enable or disable the tx-side filter function.
  0 Disable
  1 Enable
< N > 1-5 Number of filter stages.
< pregain > 0-0xFFFF Sets the gain before the filter.
< filtercoeffs > The coefficient of the biquad filter (a1 a2 b0 b1 b2).
< numshiftfactor > Number of shift factors.

NOTES

1. For narrowband voice signals, a sampling frequency of fs = 8 kHz and N = 3.
2. For wideband voice signals, a sampling frequency of fs = 16 kHz and N = 5.
3. For audio signals, a sampling frequency of fs = 48 kHz and N = 4.
4. **<pregain>** parameter fixed byte is 2, **< filtercoeffs >** parameter fixed byte is 4, **< numshiftfactor >** parameter fixed byte is 2, such as the parameters are set to 1, need the following settings: 0x0001,0x00000001,0x0001.

Example

```
AT+QTXIIR=?                                                    // query range.
+QTXIIR: (0,1),(1-5),<pregain>,<filtercoeffs>,<numshiftfactor>

OK

AT+QTXIIR=1,2,0x1234,"0x12345678,0x12345678,0x12345678,0x12345678,0x12345678,0x12345678
,0x12345678,0x12345678,0x12345678,0x12345678","0x1234,0x1234"      //set N=2.

OK
```

6.20. AT+QCFG="pcmclk" PCM Clock Signal Configuration

AT+QCFG="pcmclk" is used to enable or disable PCM clock output when there is no calling and audio play. The configuration will be stored into NV automatically.

AT+QCFG="pcmclk" PCM CLK Signal Configuration	
Write Command **AT+QCFG="pcmclk"[,<PCM_clkout>]**	Response If configuration parameters are omitted (**+QCFG="pcmclk"**), return current configuration: **+QCFG: "pcmclk",<PCM_clkout>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality:

	+CME ERROR: <err>
Maximum Response Time	300ms

Parameter

< PCM_clkout>	Enable/disable PCM clock output 0 Disable PCM clock output 1 Enable PCM clock output

NOTE

1. Configuration will be saved to NV immediately by default.
2. The output frequency is depend on the **<clock>** parameter of **AT+QDAI**

6.21. AT+QAUDCFG Audio Tuning Process

AT+QAUDCFG is used to query and configure various audio settings.

AT+QAUDCFG Audio Tuning Process	
Test Command **AT+QAUDCFG=?**	Response **+QAUDCFG: "alc5616/dlgain",<level>** (list of supported**<level>**s) **+QAUDCFG: "alc5616/ulgain",<level>** (list of supported**<level>**s) **+QAUDCFG: "tonevolume",<level>** (list of supported**<level>**s) **+QAUDCFG: "alc5616/pwrctr",<level>** (list of supported**<level>**s) **OK**

6.22. AT+QAUDCFG="alc5616/dlgain" Extension Configuration

AT+QAUDCFG="alc5616/dlgain" is used to set down link gain level for codec alc5616.

AT+QAUDCFG Extension Configuration	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG: "alc5616/ dlgain" , <level>** …… **OK**
Write Command **AT+QAUDCFG="alc5616/dlgain" [,<level>]**	Response If configuration parameters are omitted (**+QAUDCFG="alc5616/dlgain"**), return current configuration: **+QAUDCFG: "alc5616/dlgain",<level>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

<Level>	Numeric type, indicates the downlink gain of alc5616, Range: 0-100, default: 100

NOTES

1. This parameter can't be saved after powered down the module.

Example

```
AT+QAUDCFG=”alc5616/dlgain",85          //Set downlink gain to 85

OK

AT+QAUDCFG="alc5616/dlgain"             //Query the current downlink gain
+QAUDCFG: 85

OK
```

6.23. AT+QAUDCFG=”alc5616/ulgain” Extension Configuration

AT+QAUDCFG=“alc5616/ulgain” is used to set uplink gain level for codec alc5616.

AT+QAUDCFG Extension Configuration	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG: "alc5616/ ulgain" , <level>** …… **OK**
Write Command **AT+QAUDCFG="alc5616/ulgain" [,<level>]**	Response If configuration parameters are omitted (**+QAUDCFG="alc5616/ulgain"**), return current configuration: **+QAUDCFG: "alc5616/ulgain",<level>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

<Level>	Numeric type, indicates the uplink gain of alc5616, Range: 0-100, default: 100

NOTES

1. This parameter can't be saved after powered down the module.

Example

AT+QAUDCFG="alc5616/ulgain",85	//Set uplink gain to 85
OK	
AT+QAUDCFG="alc5616/ulgain"	//Query the current uplink gain
+QAUDCFG: 85	
OK	

6.24. AT+QAUDCFG=”tonevolume” Extension Configuration

AT+QAUDCFG=”tonevolume” is used to set the tone volume, use **<tonevolume>** to indicate tone volume.

AT+QAUDCFG Extension Configuration	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG:”tonevolume”,(0,100)** …… **OK**
Write Command **AT+QAUDCFG=”tonevolume”[,<0,100>]**	Response **OK** **+CME ERROR: <err>**

Parameter

< tonevolume>	Tone’s volume value, ranges: 0-100

NOTES

1, This parameter can't be saved after powered down the module.

Example

```
AT+QAUDCFG=?                     //Query range.
+QAUDCFG: "tonevolume",(0,100)

OK
AT+QAUDCFG="tonevolume",10       //Set the tone of volume is 10.
OK

AT+ QAUDCFG="tonevolume"         //Query the current volume
+QAUDCFG: 10

OK
```

6.25. AT+QAUDCFG="alc5616/pwrctr" Extension Configuration

AT+QAUDCFG="alc5616/pwrctr" is used to reset the MX-66h register power management function when the codec is powered up.

AT+QAUDCFG="alc5616/pwrctr"	Extension Configuration
Test Command **AT+QAUDCFG=?**	Response **+QAUDCFG: " alc5616/pwrctr ",(0-1)** **OK**
Write Command **AT+QAUDCFG="alc5616/pwrctr",<enable>**	Response **OK** **ERROR**
Read Command **AT+QAUDCFG="alc5616/pwrctr"**	Response **+QCFG: "alc5616/pwrctr", <enable>** **OK**
Maximum Response Time	300ms

Parameter

<enable>	Enable/disable when the codec power is reset to the MX-66h register. 0 Disable 1 Enable

NOTE

1, These parameters can be saved to NV immediately by default.
2, This command just supports Codec ALC5616, if enable this function that can be improved output gain about 10dB.

Example

```
AT+QAUDCFG=?                                   // query command
+QAUDCFG: "alc5616/pwrctr",(0-1)

OK

AT+QAUDCFG="alc5616/pwrctr",1                  //write command

OK

AT+QAUDCFG="alc5616/pwrctr"                    //read command
+QCFG: "alc5616/pwrctr", 1

OK
```

6.26. AT+QAUDCFG=”nau8814/dlgain” Extension Configuration

AT+QAUDCFG=“nau8814/dlgain” is used to set down link gain level for codec nau8814.

AT+QAUDCFG Extension Configuration	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG: "nau8814/ dlgain" , <level>** …… **OK**
Write Command **AT+QAUDCFG="nau8814/dlgain" [,<level>]**	Response If configuration parameters are omitted (**+QAUDCFG="nau8814/dlgain"**),

	return current configuration: **+QAUDCFG: "nau8814/dlgain",<level>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

<Level>	Numeric type, indicates the downlink gain of nau8814, Range: 0-100, default: 100

NOTE

1, This parameter can't be saved after power down.

Example

```
AT+QAUDCFG=”nau8814/dlgain",85          //Set downlink gain to 85

OK

AT+QAUDCFG="nau8814/dlgain"             //Query the current downlink gain
+QAUDCFG: 85

OK
```

6.27. AT+CMUT Mute Control

The command is used to enable and disable the uplink voice muting during a voice call.

AT+CMUT Mute Control	
Test Command **AT+CMUT=?**	Response **+CMUT:** (list of supported**<n>**s) **OK**
Read Command **AT+CMUT?**	Response **+CMUT:<n>** **OK**
Write Command **AT+CMUT=<n>**	Response **OK** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms
Reference 3GPP TS 27.007	

Parameter

<n>	0	Mute OFF
	1	Mute ON

NOTES

1. This command is valid only during the call.
2. This parameter can't be saved.

6.28. AT+QAUDMOD Set Audio Mode

The command is used to set the audio mode required for the connected device.

AT+QAUDMOD Set Audio Mode	
Test Command	Response

AT+QAUDMOD=?	For UC20 & EC2x modules: **+QAUDMOD: (0-2)** For UC15 module: **+QAUDMOD: (0-8)** **OK**
Read command **AT+QAUDMOD?**	Response **+QAUDMOD: <mode>** **OK**
Write Command **AT+QAUDMOD=<mode>**	Response **OK** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	300ms

Parameter

<mode>	For UC20 & EC2x modules: Numeric type, indicates the current configured audio mode. <u>0</u> Echo canceller, noise suppressor, digital gain and calibration parameter for Handset 1 Echo canceller, noise suppressor, digital gain and calibration parameter for Headset 2 Echo canceller, noise suppressor, digital gain and calibration parameter for Speaker
<mode>	For UC15 module: Numeric type, indicates the current configured audio mode. <u>0</u> Handset mode. 1 Headset mode. 2 Hands free kit mode. 3 Analog hands free kit mode. 4 Loudspeaker mode. 5 AUX PCM Handset mode. 6 AUX PCM Headset mode. 7 AUX PCM Loudspeaker mode. 8 Bluetooth headset mode.

NOTE

The parameter can’t be saved.

6.29. AT+QSIDET Set the Side Tone Gain in Current Mode

The command is used to set the side tone gain value of the current mode.

AT+QSIDET Set the Side Tone Gain in Current Mode	
Test Command **AT+QSIDET=?**	Response **+QSIDET: (0-65535)** **OK**
Read Command **AT+QSIDET?**	Response **+QSIDET: <stgain>** **OK**
Write Command **AT+QSIDET=<stgain>**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<stgain>	Numeric type, indicates the configured side tone gain in current mode. Range: 0-65535. Default value might be different in different audio modes.

NOTE

1, The parameter can't be saved after power down the module.
2, The parameter should be set to 0 at hands free field.

6.30. AT+QAUDLOOP Enable Audio Loop Test

The command is used to enable audio loop test.

AT+QAUDLOOP Enable Audio Loop Test	
Test Command **AT+QAUDLOOP=?**	Response **+QAUDLOOP: (0,1),(0-2)**

	OK
Read Command **AT+QAUDLOOP?**	Response **+QAUDLOOP: <enable>,<path>** **OK**
Write Command **AT+QAUDLOOP=<enable>[,<path>]**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<enable>	Numeric type, to enable or disable audio loop test. 0 Disable audio loop test 1 Enable audio loop test
<path>	Numeric type, indicates the test path. This parameter is intended for UC15 module only. 0 MIC1 & SPEAKER1 1 MIC2 & SPEAKER2 2 AUX PCM

NOTE

1, These parameters can’t be saved after powered down the module.

6.31. AT+QAUDPATH Set Audio Output Path

The command is used to set the current audio output path which can be MIC&SPEAKER or AUX PCM.

AT+QAUDPATH Set Audio Output Path	
Test Command **AT+QAUDPATH=?**	Response **+QAUDPATH: (0-2)** **OK**
Read Command **AT+QAUDPATH?**	Response **+QAUDPATH: <path>** **OK**

Write Command **AT+QAUDPATH=<path>**	Response **OK** **ERROR**
Maximum Response Time	300ms

Parameter

<path>	Numeric type, indicates the configured output path. <u>0</u> MIC1&SPEAKER1 1 MIC2&SPEAKER2 2 AUX PCM

NOTES

1. The parameter can't be saved after powered down the module.
2. This command is intended for UCxx modules only.

6.32. AT+VTS DTMF and Tone Generation

AT+VTS is used to send ASCII characters which causes MSC to transmit DTMF tones to a remote subscriber. This command can only be operated in voice call.

AT+VTS DTMF and Tone Generation	
Test Command **AT+VTS=?**	Response **+VTS:** (list of supported **<dtmf_string>**s),(list of supported **<duration>**s) **OK**
Write Command **AT+VTS=<dtmf_string>[,<duration>]**	Response **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**
Maximum Response Time	Depends on the length of **<dtmf_string>** and **<duration>**.
Reference 3GPP TS 27.007	

Parameter

<dtmf_string>	ASCII characters in the set 0...9,#,*, A, B, C, D. The string should be enclosed in quotation marks ("...") When sending multiple tones at a time, the time interval of two tones **<interval>** can be specified by **+VTD**. The maximal length of the string is 31.
<duration>	The duration of each tone in 1/10 seconds with tolerance The value ranges from 0 to 255, and the default is 0 If the duration is less than the minimum time specified by the network, the actual duration will be the network specified time. If this parameter is omitted, **<duration>** can be specified by **+VTD**.

Example

```
ATD12345678900;                //Dial
OK

<Call connect>

AT+VTS="1"                     //The remote caller can hear the DTMF tone
OK
AT+VTS="1234567890A"           //Send multiple tones at a time
OK
```

6.33. AT+VTD Set Tone Duration

AT+VTD is used to set the duration of DTMF tones. This command can also set time interval of two tones when sending multiple tones at a time.

AT+VTD Set Tone Duration	
Test Command **AT+VTD=?**	Response **+VTD: (0-255),(0-255)** **OK**
Read Command **AT+VTD?**	Response **+VTD: <duration>,<interval>** **OK**
Write Command **AT+VTD=<duration>[,<interval>]**	Response **OK** **ERROR** If error is related to ME functionality:

	+CME ERROR: <err>
Maximum Response Time	300ms
Reference 3GPP TS 27.007	

Parameter

<duration>	The duration tone in 1/10 seconds with tolerance. The value ranges from 0 to 255, and the default is 3. If the duration is less than the minimum time specified by the network, the actual duration will be network specified time.
<interval>	The time interval of two tones when sending multiple tones at a time by **+VTS**. The value ranges from 0 to 255, and the default is 0.

NOTE

1, These parameters can't be saved after powered down the module.

6.34. AT+QAUDCFG="digital/dlgain" Set Downlink Digital Gain

The command is used to set downlink digital gain level.

AT+QAUDCFG="digital/dlgain" Set Downlink Digital Gain	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG: "digital/dlgain",<gain>** …… **OK**
Write Command **AT+QAUDCFG="digital/dlgain"[,<gain>]**	Response If configuration parameters are omitted (**+QAUDCFG="digital/dlgain"**), return the current configuration: **+QAUDCFG: "digital/dlgain",<gain>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality:

	+CME ERROR: <err>

Parameter

<gain>	Numeric type, indicates the downlink digital gain level. Range: 0-10000. Default value might be different in different audio modes

NOTES

1. The parameter will not be saved.
2. This command is intended for UCxx modules only.

Example

```
AT+QAUDCFG="digital/dlgain",8000        //Set downlink digital gain to 8000.

OK
```

6.35. AT+QAUDCFG="innercodec/dlgain" Set Downlink Digital Gain for Internal Codec

The command is used to set the downlink digital gain level for internal codec.

AT+QAUDCFG="innercodec/dlgain" Set Downlink Digital Gain for Internal Codec	
Test Command **AT+QAUDCFG=?**	Response …… **+QAUDCFG: "innercodec/dlgain",<gain>** …… **OK**
Write Command **AT+QAUDCFG="innercodec/dlgain"[,<gain>]**	Response If configuration parameters are omitted (**+QAUDCFG="innercodec/dlgain"**), return the current configuration: **+QAUDCFG: "innercodec/dlgain",<gain>** **OK** If configuration parameters are entered: **OK** **ERROR** If error is related to ME functionality: **+CME ERROR: <err>**

Parameter

<gain>	Numeric type, indicates the downlink digital gain level for internal codec. Range: 0-65535. Default value might be different in different audio modes

NOTES

1. The parameter can't be saved after powered down the module.
2. This command is intended for UCxx modules only.

Example

```
AT+QAUDCFG="innercodec/dlgain",8000       //Set downlink digital gain to 8000.

OK
```

6.36. AT+ QEEC Set Echo Cancellation parameters

AT+QEEC is used to set Echo Cancellation parameters.

AT+QEEC Set Echo Cancellation parameters	
Test Command **AT+QEEC=?**	Response **+QEEC: (0-49),(0-65535)** **OK**
Read Command **AT+QEEC?**	Response **+QEEC: <index>,<value>** **……** **+QEEC: <index>,<value>** **OK**
Write Command **AT+QEEC=<index>,<value>**	Response **OK** **ERROR**

Parameter

<index>	Numeric types, indicates the parameter's index. Range:0-49
<value>	Numeric types, indicates the parameter's value. Range: 0-65535

NOTE

1. This parameter can't be saved after powered down the module.
2. For the detailed information about EEC, please refer to AT+EEC_Manual_V1.0.

Example

```
AT+QEEC=?            // query range.
+QEEC: (0-49), (0-65535)
OK

AT+QEEC=6,1234       //set index 6 value to 1234.
OK
```

7 Appendix A References

Table 3: Related Documents

SN	Document name	Remark
[1]	Quectel_UCxx/EC2x_Hardware_Design	Hardware design for UC15, UC20, EC20, EC21, EC25 and EC20 R2.0
[2]	Quectel_UCxx/EC2x_Reference_Design	Reference design for UC15, UC20, EC20, EC21, EC25 and EC20 R2.0
[3]	Quectel_UCxx/EC2x_AT_Commands_Manual	AT commands manual for UC15, UC20, EC20, EC21, EC25 and EC20 R2.0
[4]	Quectel_WCDMA_UGxx_Audio_Design_Note	Audio design note for Quectel WCDMA UGxx modules

Table 4: Terms and Abbreviations

Abbreviation	Description
EEC	Enhanced Echo Canceller
ESD	Electrostatic Discharge
I2C	Inter Integrated Circuit
LDO	Low Dropout Regulator
LSB	Least Significant Bit
MCU	Microcontroller Unit
ME	Mobile Equipment
MSB	Most Significant Bit
NR	Noise Reducer
NV	Non-Volatile Memory
PCB	Printed Circuit Board

PCM	Pulse-code Modulation
RC	Resistance Capacitance
RF	Radio Frequency
THD	Total Harmonic Distortion
TDD	Time Division Duplexing
UART	Universal Asynchronous Receiver/Transmitter
XTAL	Crystal