EC2X&AG35-QuecOpen VOICE API MANUAL

LTE Module Series

Rev. EC2X&AG35-QuecOpen_VOICE_guide_manual_V1.0

Date: 2018-03-17

Status: Temporary

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.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

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

For technical support, or to report documentation errors, please visit:
http://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.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-3-17	Laurence	Initial

Contents

About the Document **3**

Contents **4**

1 **Introduction** **5**

2 **VOICE API** **6**

2.1. QL_Voice_Call_Client_Init 6
2.2. QL_Voice_Call_Client_Deinit 6
2.3. QL_Voice_Call_AddStateHandler 6
2.4. QL_Voice_Call_RemoveStateHandler 7
2.5. QL_Voice_Call_Start 7
2.6. QL_Voice_Call_End 7
2.7. QL_Voice_Call_Answer 8

3 **Procedure Steps Of The Demo** **9**

4 **VOICE Daemon Demo** **10**

4.1. Execute the command 10
4.2. Originating call 10
4.3. Answer and Hand up 11

5 **Compilation Introduction** **12**

1 Introduction

This document mainly introduces how to use the voice function of Quectel Open module. Voice function is only supported by the special software version.

The mobile phone service is a telephone service in which a user uses a mobile phone to communicate with a domestic and foreign phone user via a base station and automatic switching equipment, or through a public telephone network.

The calling party is the mobile phone holder actively establishes communication. The called party, ie the mobile phone holder, passively answers the phone. Hang up the phone, that is, the party on the phone actively disconnects the communication between the parties, this can be done by either party. Refused to answer the call, that is, the call was not fully established, the called party refused the communication.

VOICE API

2 QL_Voice_Call_Client_Init

(1) Function prototype：

2.1.

```
int QL_Voice_Call_Client_Init(voice_client_handle_type    *ph_voice);
```

(2) Parameter description：

1) ph_voice：OUT   the pointer of voice handle

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Init Voice function handle.

2.2. QL_Voice_Call_Client_Deinit

(1) Function prototype：

```
int QL_Voice_Call_Client_Deinit (voice_client_handle_type    h_voice);
```

(2) Parameter description：

1) h_voice：IN   voice handle

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Destroy related Voice feature resources

2.3.

QL_Voice_Call_AddStateHandler

(1) Function prototype：

```
int QL_Voice_Call_AddStateHandler(voice_client_handle_type                h_voice,
                                  QL_VoiceCall_StateHandlerFunc_t    handlerPtr,
                                  void*                                     contextPtr)
```

(2) Parameter description：

1) h_voice：IN voice handle

2) handlerPtr IN   voice callback function

3) contextPtr IN （the content of incoming call_id）

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Register the callback function to receive the incoming voice;

QL_Voice_Call_RemoveStateHandler

(1) Function prototype：

```
int QL_Voice_Call_RemoveStateHandler(voice_client_handle_type          h_voice)
```

2.4. (2) Parameter description：

1) h_voice：IN voice handle

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Destroy the registered the callback function;

QL_Voice_Call_Start

2.5. (1) Function prototype：

```
int QL_Voice_Call_Start(voice_client_handle_type      h_voice,
                        E_QL_VCALL_ID_T                   simId,
                        char*                           phone_number,
                        int                             *call_id)
```

(2) Parameter description：

1) h_voice：IN voice handle

2) simId：IN slot ID（dumped params）

3) phone_number: IN The called number

4) call_id: OUT call ID

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

2.6. **Originating-caller make a call**；

QL_Voice_Call_End

(1) Function prototype：

```
int QL_Voice_Call_End(  voice_client_handle_type      h_voice,
                        int                             call_id)
```

(2) Parameter description：

1) h_voice：IN voice handle

2) call_id: IN call ID

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Hang up or refuse a call；

2.7. QL_Voice_Call_Answer

(1) Function prototype：

```
int QL_Voice_Call_Answer(voice_client_handle_type     h_voice,
                         int                          call_id )
```

(2) Parameter description：

1) h_voice：IN voice handle

2) call_id: IN Call ID of current communication

(3) Return description：int,0-SUCCESS, Greater than 0- partial SUCCESS,Less than 0- FAILURE

(4) Functional description：

Answer the call；

3 Procedure Steps Of The Demo

Please refer to **example/voice/example_voice.c**

Description：

step1：QL_Voice_Call_Client_Init------ register voice client

step2：QL_Voice_Call_AddStateHandler----- register callback

step3：communication

step4：QL_Voice_Call_RemoveStateHandler----destroy callback

step5：QL_Voice_Call_Client_Deinit----- destroy client

VOICE Daemon Demo

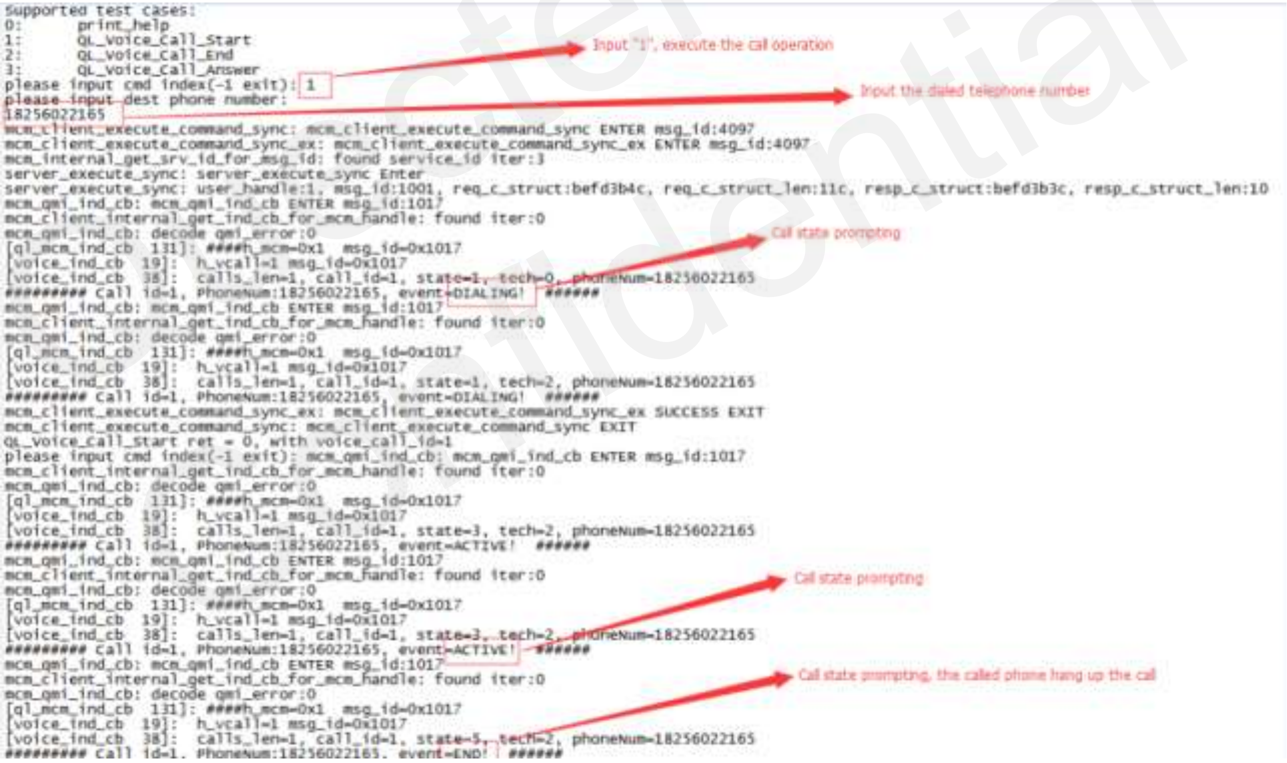
4 Execute the command

Please execute the command **root@mdm9607-perf:~# ./example_voice**

4.1.

Originating call

4.2.



```
Supported test cases:
0:      print_help
1:      QL_Voice_Call_Start
2:      QL_Voice_Call_End
3:      QL_Voice_Call_Answer
please input cmd index(-1 exit): 1
please input dest phone number:
18256022165
mcm_client_execute_command_sync: mcm_client_execute_command_sync ENTER msg_id:4097
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex ENTER msg_id:4097
mcm_internal_get_srv_id_for_msg_id: found service_id iter:3
server_execute_sync: server_execute_sync Enter
server_execute_sync: user_handle:1, msg_id:1001, req_c_struct:befd3b4c, req_c_struct_len:11c, resp_c_struct:befd3b3c, resp_c_struct_len:10
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=1, tech=0, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=DIALING!  ######
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=1, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=DIALING!  ######
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex SUCCESS EXIT
mcm_client_execute_command_sync: mcm_client_execute_command_sync EXIT
QL_Voice_Call_Start ret = 0, with voice_call_id=1
please input cmd index(-1 exit): mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=3, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=ACTIVE!  ######
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=3, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=ACTIVE!  ######
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=5, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=END!  ######
```

Input "1", execute the call operation

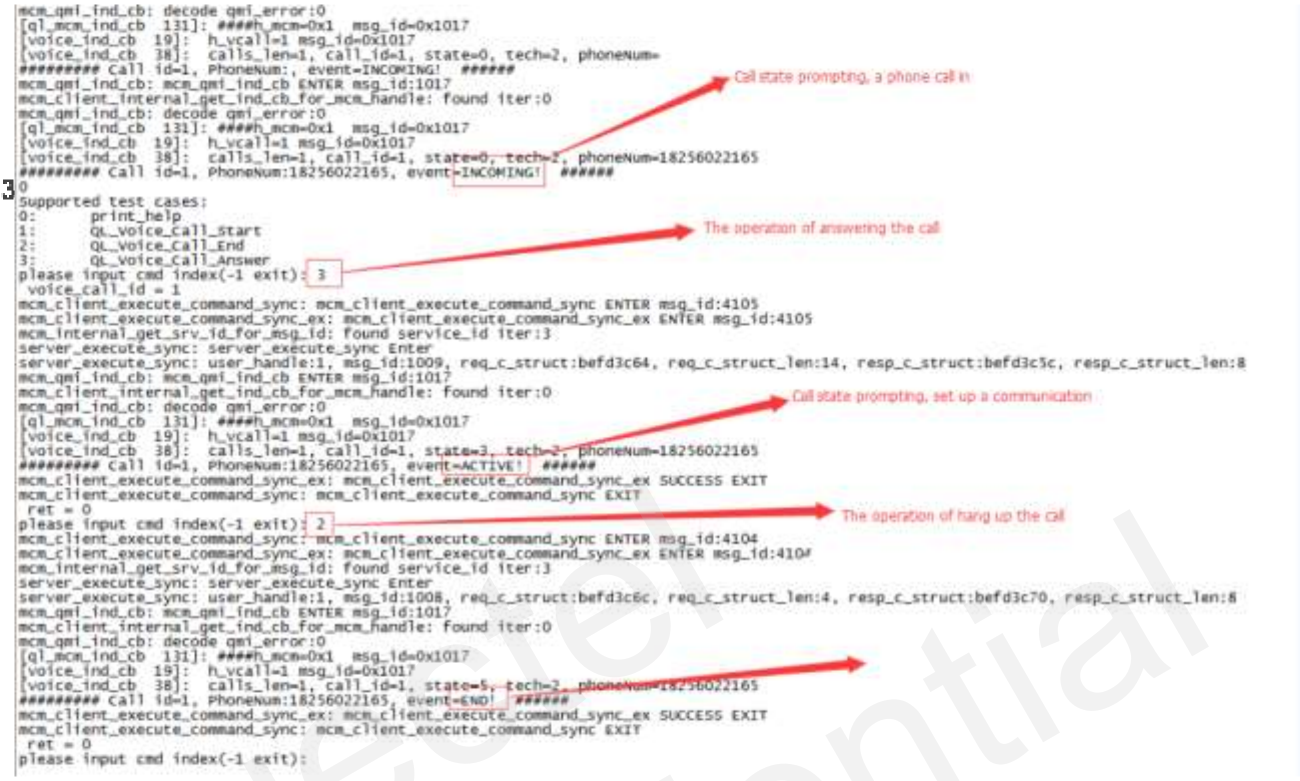
Input the dialed telephone number

Call state prompting

Call state prompting

Call state prompting, the called phone hang up the call

4.3 Answer and Hand up



```
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=0, tech=2, phoneNum=
######### Call id=1, PhoneNum:, event=INCOMING!  ######
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=0, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=INCOMING!  ######
0
Supported test cases:
0:      print_help
1:      QL_Voice_Call_Start
2:      QL_Voice_Call_End
3:      QL_Voice_Call_Answer
please input cmd index(-1 exit): 3
 voice_call_id = 1
mcm_client_execute_command_sync: mcm_client_execute_command_sync ENTER msg_id:4105
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex ENTER msg_id:4105
mcm_internal_get_srv_id_for_msg_id: found service_id iter:3
server_execute_sync: server_execute_sync Enter
server_execute_sync: user_handle:1, msg_id:1009, req_c_struct:befd3c64, req_c_struct_len:14, resp_c_struct:befd3c5c, resp_c_struct_len:8
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=3, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=ACTIVE!  ######
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex SUCCESS EXIT
mcm_client_execute_command_sync: mcm_client_execute_command_sync EXIT
 ret = 0
please input cmd index(-1 exit): 2
mcm_client_execute_command_sync: mcm_client_execute_command_sync ENTER msg_id:4104
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex ENTER msg_id:4104
mcm_internal_get_srv_id_for_msg_id: found service_id iter:3
server_execute_sync: server_execute_sync Enter
server_execute_sync: user_handle:1, msg_id:1008, req_c_struct:befd3c6c, req_c_struct_len:4, resp_c_struct:befd3c70, resp_c_struct_len:8
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:1017
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb  131]: ####h_mcm=0x1  msg_id=0x1017
[voice_ind_cb  19]:  h_vcall=1 msg_id=0x1017
[voice_ind_cb  38]:  calls_len=1, call_id=1, state=5, tech=2, phoneNum=18256022165
######### Call id=1, PhoneNum:18256022165, event=END!  ######
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex SUCCESS EXIT
mcm_client_execute_command_sync: mcm_client_execute_command_sync EXIT
 ret = 0
please input cmd index(-1 exit):
```

Call state prompting, a phone call in

The operation of answering the call

Call state prompting, set up a communication

The operation of hang up the call

5 Compilation Introduction

Description：

(1) Unzip file ql-ol-sdk.tar.bz2：tar -jxvf ql-ol-sdk.tar.bz2

(2) Enter the ql-ol-sdk directory：cd ql-ol-sdk

(3) source ql-ol-crosstool/ql-ol-crosstool-env-init (**Ensure that the SDK version is consistent with the module version, or there may be an error**)

(4) Execute the command：cd ql-ol-extsdk/example/voice

(5) Execute the command：make clean;make;