

# **EC2X&AG35-Quecpen**

# **SMS API MANUAL**

**LTE Module Series**

Rev. EC2X&AG35-QuecOpen\_SMS\_Guide\_Manual\_V1.4

Date: 2018-06-28

Status: Temporary



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

**Quectel Wireless Solutions Co., Ltd.**

7<sup>th</sup> Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: [info@quectel.com](mailto: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](mailto: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**

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

***Copyright © Quectel Wireless Solutions Co., Ltd. 2018. All rights reserved.***

# About the Document

## History

Revision	Date	Author	Description
1.0	2017-12-27	Vicent GAO	Initial
1.1	2018-03-17	Laurence YIN	Modification
1.2	2018-03-30	Laurence YIN	Add sending PDU API
1.3	2018-04-23	Laurence YIN	Add Get Setting SMS Center Address Interface
1.4	2018-06-28	Laurence YIN	Add E_QL_SMS_FORMAT_T type

Quectel  
Confidential

## Contents

	About the Document.....	3
	Contents .....	4
	Introduction .....	5
	SMS API .....	6
	QL_SMS_Client_Init .....	6
	QL_SMS_Client_Deinit .....	6
1	QL_SMS_AddRxMsgHandler .....	6
2	QL_SMS_Send_Sms .....	7
2.1.	Struct ql_sms_info_t .....	7
2.2.	2.5.1. E_QL_SMS_FORMAT_T .....	7
2.3.	QL_SMS_Send_SmsPdu .....	7
2.4.	QL_SMS_SetSmsCenterAddress .....	8
2.5.	2.7. QL_SMS_GetSmsCenterAddress .....	8
2.6.		
2.7.		
2.8.	Execution Step Of The Demo .....	9
3	Initialization (Interface:EXSMS_slNit) .....	9
3.1.	Send Message (Interface:EXSMS_sSend) .....	9
3.2.	Receive SMS (Interface:EXSMS_sRecvCb) .....	9
3.3.	Program Exit (Interface:EXSMS_sExit, press Ctrl+C) .....	9
3.4.		
4	Execution of the demo .....	10
4.1.	Execute the command .....	10
4.2.	Send English Characters Message .....	10
4.3.	Send Chinese Message .....	11
4.4.	Recv English Characters Message .....	11
4.5.	Receive Chinese Message .....	11
4.6.		
4.7.		
5	Send GW PDU .....	12
	Get SCA Number .....	12
	SMS Build instructions.....	13

# Introduction

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

SMS(Short Message Service), is that the text or digital information which user send or receive directly through mobile phones or other telecommunications terminal. The number of characters the user can send or receive each time, is 160 in English or numeric characters, or 70 Chinese characters.

Quectel  
Confidential

# SMS API

## 2 QL\_SMS\_Client\_Init

(1) Function prototype:

2.1. `int QL_SMS_Client_Init(sms_client_handle_type *ph_sms);`

(2) Parameter description:

1) `ph_sms:[OUT]` the pointer of sms handle

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

(4) Functional description:

init SMS function handle.

## 2.2. QL\_SMS\_Client\_Deinit

(1) Function prototype:

`int QL_SMS_Client_Deinit(sms_client_handle_type h_sms);`

(2) Parameter description:

1) `h_sms:[IN]` sms handle

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

(4) Functional description:

2.3.

## QL\_SMS\_AddRxMsgHandler

(1) Function prototype:

`int QL_SMS_AddRxMsgHandler(QL_SMS_RxMsgHandlerFunc_t handlerPtr, void* contextPtr);`

(2) Parameter description:

1) `handlerPtr [IN]` sms callback function

2) `contextPtr [IN]` (the content of incoming text messages.)

(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 new text message;

## QL\_SMS\_Send\_Sms

(1) Function prototype:

```
int QL_SMS_Send_Sms(sms_client_handle_type h_sms, ql_sms_info_t *pt_sms_info);
```

(2) Parameter description:

1) h\_sms:[IN] sms handle

2.4. 2) pt\_sms\_info:[IN] Text message content and destination number.

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

(4) Functional description:

Send messages;

## Struct ql\_sms\_info\_t

### 2.5.2.5.1. E\_QL\_SMS\_FORMAT\_T

表示发送和接收上报的短信的编码格式。

E\_QL\_SMS\_FORMAT\_GSM\_7BIT = 0,

//Indicate is GSM-7coding when networking is GW (for example CU or CMCC), GSM 03.38  
version 7.2.0 Release 1998

// Indicate is 7bit ASCII when networking is CDMA 2G (for example CT)

E\_QL\_SMS\_FORMAT\_BINARY\_DATA = 1,

// Binary coding

E\_QL\_SMS\_FORMAT\_UCS2 = 2,

//Unicode coding

E\_QL\_SMS\_FORMAT\_IRA = 3,

// Internal test use

2.6.

## QL\_SMS\_Send\_SmsPdu

(1) Function prototype:

```
int QL_SMS_Send_SmsPdu( sms_client_handle_type h_sms,  
                        ql_wms_send_raw_message_data_t raw_message_data,  
                        ql_wms_raw_send_resp_t *rawresp);
```

(2) Parameter description:

1) h\_sms:[IN] sms handle (no work)

2) raw\_message\_data:[IN] PDU content,PDU len,PDU format.

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

(4) Functional description:

Send messages (sca can be not code into PDU) ;

## **QL\_SMS\_SetSmsCenterAddress**

(1) Function prototype:

```
int QL_SMS_SetSmsCenterAddress( sms_client_handle_type    h_sms,  
                               ql_sms_service_center_cfg_t *set_sca_cfg)
```

**2.7.**(2) Parameter description:

- 1) h\_sms: [IN] sms handle
- 2) get\_sca\_cfg: [IN] SMS center number and type

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

(4) Functional description:

Set SMS center number and type.

## **2.7. QL\_SMS\_GetSmsCenterAddress**

**2.8.**(1) Function prototype:

```
int QL_SMS_GetSmsCenterAddress( sms_client_handle_type    h_sms,  
                               ql_sms_service_center_cfg_t *get_sca_cfg)
```

(2) Parameter description:

- 1) h\_sms: [IN] sms handel
- 2) get\_sca\_cfg:[IN] SMS center number and type

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

(4) Functional description:

Get SMS center number and type.



# Execution Step Of The Demo

The whole eCall procedure contains Pull Mode, Push Mode and Call Release. The following figures show the detailed procedure of each mode.

**3**

## Initialization (Interface:EXSMS\_sInit)

- 3.1**(1) Register the necessary signal processing functions. (SIGINT)  
(2) Call interface QL\_SMS\_Client\_Init, register SMS client  
(3) Call interface QL\_SMS\_AddRxMsgHandler, register callback function

## **3.2.** Send Message (Interface:EXSMS\_sSend)

- (1) Obtain target number, message encoding and text message from standard input.  
(2) Call QL\_SMS\_Send\_Sms, and send messages.

**3.3.**

## Receive SMS (Interface:EXSMS\_sRecvCb)

- (1) If the new message is not stored, it will display the message directly (destination number, message encoding, text message content).  
**3.4**(2) If the new message has been stored, display the SMS memory and its index.

## Program Exit (Interface:EXSMS\_sExit, press Ctrl+C)

- (1) Call interface QL\_SMS\_Client\_Deinit, log out the sms client.

# Execution of the demo

## 4 Execute the command

Please execute following command.

```
root@mdm9607-perf:~# ./example_sms
```

4.1.

### Send English Characters Message

4.2.

```
[sms_ind_cb 283]: n_sms=0x2 msg_id=2023406815, ind_c_struct=0x0, ind_len=0
[sms_ind_cb 375]:
===== SMS service Ready ! =====
Supported test cases:
61: Send SMS
please input item(-1 exit): 61
Enter EXSMS_sisRegisterTonw(101) SMS_NOTE: eRat:7,eState:1,iRssi:-31,iBer:0
please input dest phone number:
18256022165
please input sms encoding type(0:GSM-7, 1:Binary, 2:UCS2, 3:IRA):
0
please input GSM data:
hello,world!
[QL_SMS_SConvCharSet 521]: SMS_WARNING! format:0 ignore.[QL_SMS_Send_Sms 563]: MCM_CLIENT_EXECUTE_COMMAND
mcm_client_execute_command_async: mcm_client_execute_command_async ENTER msg_id:1794
mcm_internal_get_srv_id_for_msg_id: found service_id iter:2
mcm_client_internal_update_async_cb_info: mcm_client_internal_update_async_cb_info: found slot 0 with mcm ha
server_execute_async: server_execute_async Enter
mcm_client_execute_command_async: mcm_client_execute_command_async SUCCESS EXIT
[QL_SMS_Send_Sms 570]: ret_val=0, resp_code= 0, error_code=0
#QL_SMS_Send_Sms succ
Supported test cases:
61: Send SMS
please input item(-1 exit): mcm_qmi_async_cb: mcm_qmi_async_cb ENTER msg_id:1794 transp_err:0
mcm_qmi_async_cb: token_id:11
mcm_client_internal_get_async_cb_for_mcm_handle: found iter:0
[sms_async_cb 396]: h_sms=0x2 msg_id=0x702
[sms_async_cb 406]:
== Send message successfully! ==
```

1th step: choose item 61

2th step: enter phone numbers

3th step: choose encode type

4th step: enter sms content(GSM-7)

send sms status

## Send Chinese Message

4.3

```

== Send message successfully! ==
61
Enter EXSMS_SISRegisterToNW(101),SMS_NOTE: eRat:7,eState:1,iRssi:-37,iBer:0
please input dest phone number:
18256022165
please input sms encoding type(0:GSM-7, 1:Binary, 2:UCS2, 3:IRA):
2
please input UCS2 data:
62116211
[QL_SMS_SConvCharSet 521]: SMS_WARNING! format:2 ignore.[QL_SMS_Send_Sms 563]: MCM_CLIENT_EXECUTE_COMMAND
mcm_client_execute_command_async: mcm_client_execute_command_async ENTER msg_id:1794
mcm_internal_get_srv_id_for_msg_id: found service_id iter:2
mcm_client_internal_update_async_cb_info: mcm_client_internal_update_async_cb_info: found slot 0 with mcm
server_execute_async: server_execute_async Enter
mcm_client_execute_command_async: mcm_client_execute_command_async SUCCESS EXIT
[QL_SMS_Send_Sms 570]: ret_val=0, resp_code= 0, error_code=0
#QL_SMS_Send_Sms succ
Supported test cases:
61: send SMS
please input item(-1 exit): mcm_qmi_async_cb: mcm_qmi_async_cb ENTER msg_id:1794 transp_err:0
mcm_qmi_async_cb: token_id:12
mcm_client_internal_get_async_cb_for_mcm_handle: found iter:0
[sms_async_cb 396]: h_sms=0x2 msg_id=0x702
[sms_async_cb 406]:
== Send message successfully! ==

```

1th step: choose item 61

2th step: enter phone numbers

3th step: enter encode type

4th step: enter unicode sms content

## Recieve English Characters Message

4.4.

```

[wms_client_ind_cb_fcn 37]: msg_id=0x1
[wms_client_ind_cb_fcn 55]: New msg got! decode ret=0, stored to 0 index=0
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:707
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb 133]: ###h_mcm=0x2 msg_id=0x707
[sms_ind_cb 283]: h_sms=0x2 msg_id=1799, ind_c_struct=0xB3C0A7A8, ind_len=440
[sms_ind_cb 299]: ###New msg arrived! SMS format=GSM-7, MsgId=4294967295, SrcNum=18119602794, content=Hello world
[EXSMS_SRecvCbDisplayGsm 367]: ###You've got one new GSM-7 msg from 18119602794 at content=Hello world
48656C6C6F20776F726C64

```

one incoming sms  
encode:GSM-7

4.5.

## Receive Chinese Message

```

[wms_client_ind_cb_fcn 37]: msg_id=0x1
[wms_client_ind_cb_fcn 55]: New msg got! decode ret=0, stored to 0 index=0
mcm_qmi_ind_cb: mcm_qmi_ind_cb ENTER msg_id:707
mcm_client_internal_get_ind_cb_for_mcm_handle: found iter:0
mcm_qmi_ind_cb: decode qmi_error:0
[ql_mcm_ind_cb 133]: ###h_mcm=0x2 msg_id=0x707
[sms_ind_cb 283]: h_sms=0x2 msg_id=1799, ind_c_struct=0xB3C0A7A8, ind_len=440
[sms_ind_cb 299]: ###New msg arrived! SMS format=UCS2, MsgId=4294967295, SrcNum=18119602794, content=0
###You've got one new UCS2 msg from 18119602794 at , len=8, content=
Received UCS raw data:4F60597D4E2D56FD

```

one incoming sms  
encode type:UCS2

## Send GW PDU

```
Supported test cases:
61: Send SMS
62: QL_SMS_SendSmsPdu
please input item(-1 exit): 62
please input PDU content:
0001000b818152062261f500000679743e269b01
4.6 sms pdu_len = 20 , pdu = 00 01 00 0b 81 81 52 06 22 61 f5 00 00 06 79 74 3e 26 9b 01
please input PDU FORMAT(0-CAMD_PP; 6-GW_PP):
6
please FORMAT = 6
message_id:54
Supported test cases:
61: Send SMS
62: QL_SMS_SendSmsPdu
```

## Get SCA Number

```
4.7 ===== SMS service Ready ! =====
Supported test cases:
61: Send SMS
62: QL_SMS_Send_SmsPdu
63: QL_SMS_GetSmsCenterAddress
64: QL_SMS_SetSmsCenterAddress
please input item(-1 exit): 63
mcm_client_execute_command_sync: mcm_client_execute_command_sync ENTER msg_id:1793
mcm_client_execute_command_sync_ex: mcm_client_execute_command_sync_ex ENTER msg_id:1793
mcm_internal_get_srv_id_for_msg_id: found service_id iter:2
server_execute_sync: server_execute_sync Enter
server_execute_sync: user_handle:3, msg_id:701, req_c_struct:befc9ddc, req_c_struct_len:1, resp_c.
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_sms_getSmsCenterAddress_1007]: ql_sms_getSmsCenterAddress fail ret = 0
service_center_addr:+316540942001 service_center_addr_type:145
Supported test cases:
61: Send SMS
62: QL_SMS_Send_SmsPdu
63: QL_SMS_GetSmsCenterAddress
64: QL_SMS_SetSmsCenterAddress
please input item(-1 exit): █
```

## SMS Build instructions

- (1) Unpack ql-ol-sdk.tar.bz2:tar -jxvf ql-ol-sdk.tar.bz2
- 5** (2) Enter ql-ol-sdk directory:cd ql-ol-sdk
- (3) source ql-ol-crostoool/ql-ol-crostoool-env-init **(Make sure the SDK version is consistent with the module version, otherwise you might have an error)**
- (4) Execute command:cd ql-ol-extsdk/example/sms
- (5) Execute command:make clean;make;

Quectel  
Confidential