AG35-QuecOpen

ECALL API MANUAL

LTE Module Series

Rev. AG35-QuecOpen_ECALL_API_guide_manual_V1.7

Date: 2018-09-20

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	2018-02-28	Laurence YIN	Initial
1.1	2018-03-17	Laurence YIN	Modification
1.2	2018-03-27	Laurence YIN	Updated
1.3	2018-04-04	Laurence YIN	Updated
1.4	2018-05-31	Laurence YIN	Updated Push API
1.5	2018-08-08	Laurence YIN	Update QL_Voice_Call_Ecall struct
1.6	2018-09-13	Laurence YIN	Add Test Item Introduction
1.7	2018-09-20	Laurence YIN	Modify MSD PUSH Interface

Contents

Figure Index

1 Introduction

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

eCall is defined as a manually or automatically initiated emergency call from a vehicle, supplemented with a minimum set of emergency related data (MSD), as defined under the EU Commission's eSafety initiative.

It can be depicted by the figure below.

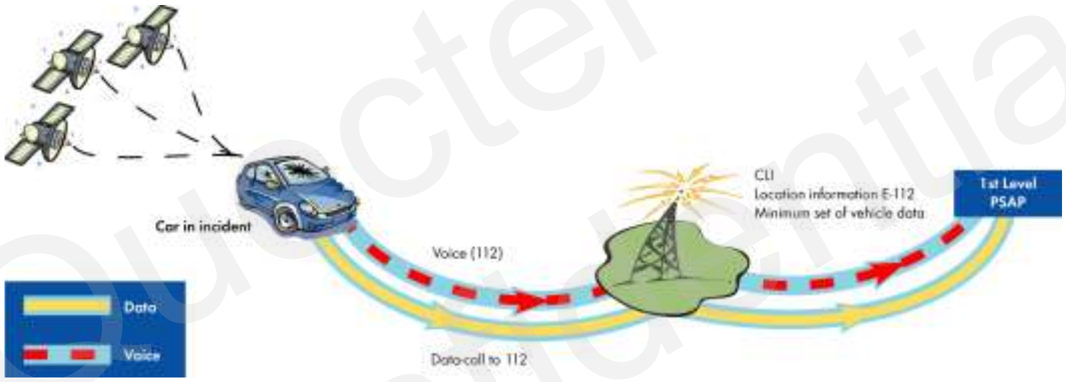


Figure 1: eCall System Overview

The architecture of eCall system is described in Figure 2. In Quectel test system, the module has the ability to act as IVS and also to simulate the PSAP. Thus, eCall testing can be easily performed by preparing two Quectel modules in the circumstance without access to a real PSAP. It will be described in the following chapters. Obviously, if a real PSAP can be accessed, testing in the real environment is preferred.

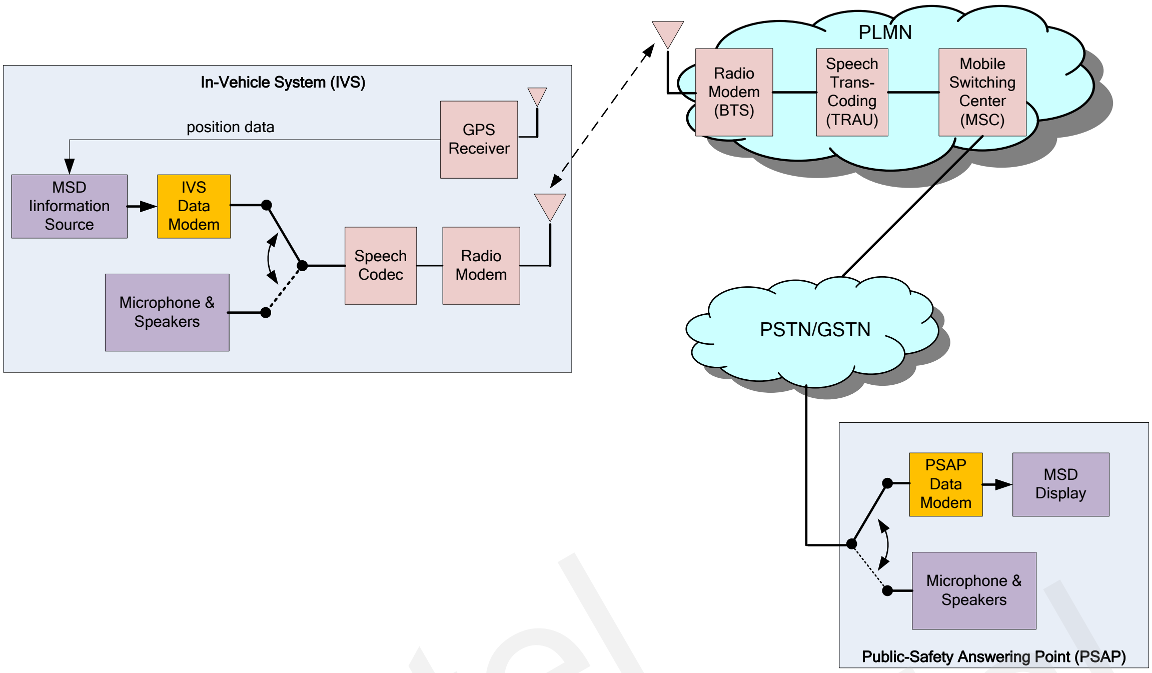


Figure 2: eCall System Architecture

2 ECALL API

2.1. QL_Voice_Call_Client_Init

(1) Function prototype:

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, 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, 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, Less than 0- FAILURE

(4) Functional description:

Register the callback function to receive the incoming voice;

2.4. QL_Voice_Call_RemoveStateHandler

(1) Function prototype:

int QL_Voice_Call_RemoveStateHandler(voice_client_handle_type h_voice)

(2) Parameter description:

1) h_voice: IN voice handle

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

(4) Functional description:

Destroy the registered the callback function;

2.5. QL_Voice_Call_Ecall

(1) Function prototype:

```
int QL_Voice_Call_Ecall(voice_client_handle_type      h_voice,
                        E_QL_VCALL_ID_T               simId,
                        char*                         phone_number,
                        ql_mcm_ecall_info             ecall_info,
                        int                           *call_id);
```

(2) Parameter description:

1) h_voice: IN voice handle
2) simId: INslot ID（dumped params）
3) phone_number: IN The called number
4) ecall_info IN msd info
5) call_id: OUT call ID

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

(4) Functional description:

Originating-caller make a ecall;

2.6. QL_Voice_Call_Ecall_HangUp

(1) Function prototype:

```
int QL_Voice_Call_Ecall_HangUp (   voice_client_handle_type      h_voice)
```

(2) Parameter description:

1) h_voice: IN   voice handle

(3) Return description: int, 0-SUCCESS, nonzero - FAILURE.

(4) Functional description:

Hang up;

2.7. QL_Voice_Call_AddCommonStateHandler

(1) Function prototype:

```
int QL_Voice_Call_AddCommonStateHandler(voice_client_handle_type h_voice,
                                        QL_VoiceCall_CommonStateHandlerFunc_t     handlerPtr);
```

(2) Parameter description:

1) h_voice: IN   voice handle

2) handlerPtr IN

(3) Return description: int, nonzero - FAILURE

(4) Functional description:

Register the callback function to receive the ind info;

2.8. QL_Voice_Call_RemoveCommonStateHandler

(1) Function prototype:

```
int QL_Voice_Call_RemoveCommonStateHandler(voice_client_handle_type h_voice);
```

(2) Parameter description:

1) h_voice: IN   voice handle

(3) Return description: int, nonzero - FAILURE

(4) Functional description:

Destroy the registered the callback function;

2.9. QL_Voice_Call_Ecall_MsdPush

(1) Function prototype:

```
int QL_Voice_Call_Ecall_MsdPush
(
    voice_client_handle_type      h_voice,
    E_QL_MCM_ECALL_STATE_T *ecall_state
)
```

2.10. QL_Voice_Call_Ecall_UpdateMsd

(1) Function prototype:

```
int QL_Voice_Call_Ecall_UpdateMsd
(
    voice_client_handle_type      h_voice,
    const char *hex_msd
);
```

(2) Parameter description:

1) h_voice: IN voice handle
2) hex_msd:IN msd hex string

(3) Return description: int, nonzero - FAILURE

(4) Functional description:

Update IVS msd;

3 Program Steps Of The Demo

Please refer to **example/ecall/example_ecall.c**

Description:

step1:QL_Voice_Call_Client_Init------ register voice client
step2:QL_Voice_Call_AddCommonStateHandler ----- register callback
step3: Communication
step4:QL_Voice_Call_RemoveCommonStateHandler ----destroy callback
step5:QL_Voice_Call_Client_Deinit----- destroy client

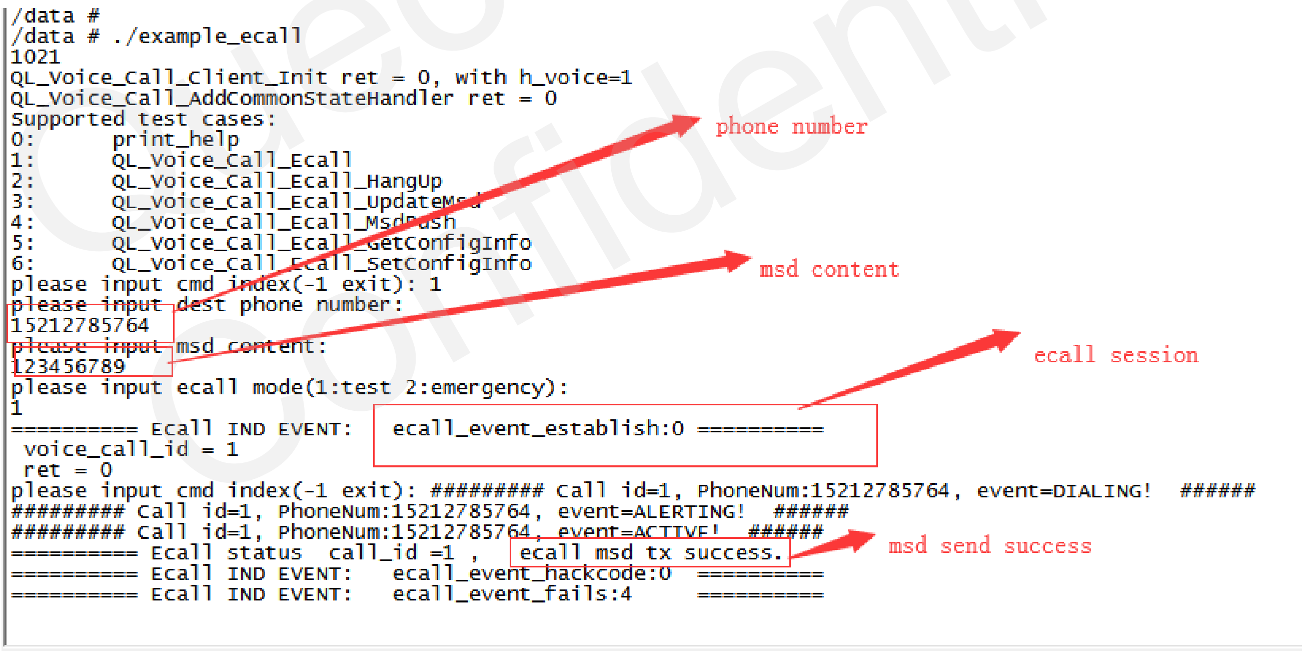
Execution of the demo

4 Execute the command

4.1.

```
/usrdata # ./example_ecall
```

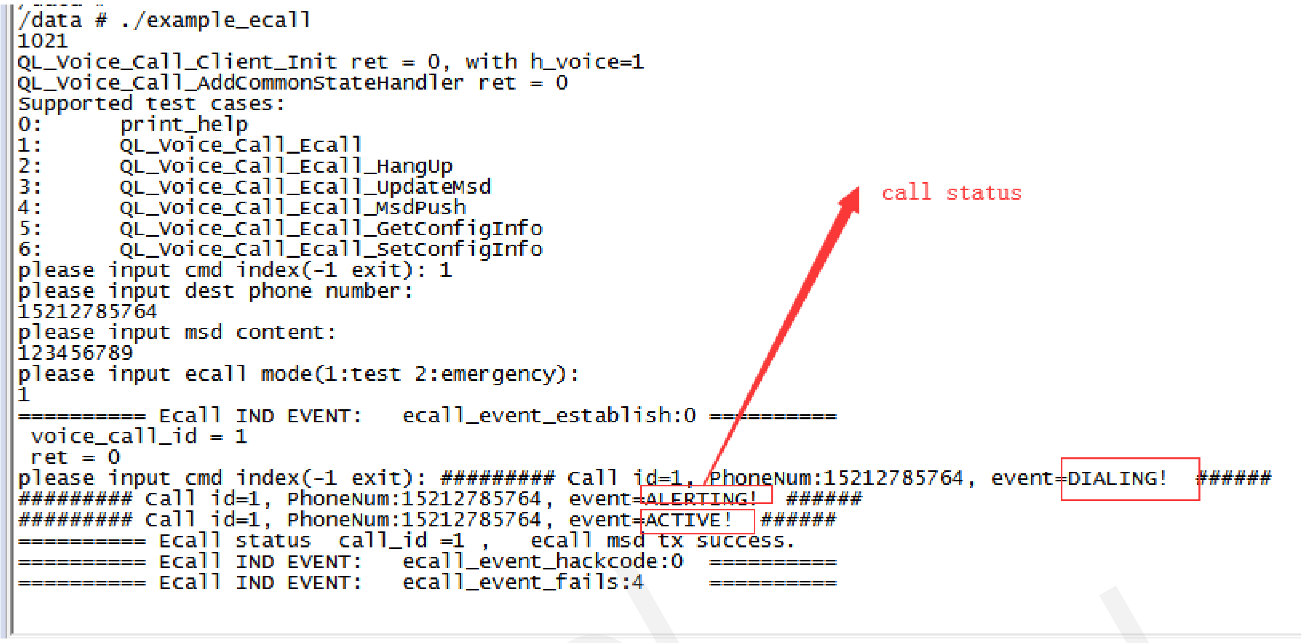
4.2. MO ECALL（E_QL_MCM_ECALL_TEST 模式）



```
/data #
/data # ./example_ecall
1021
QL_Voice_Call_Client_Init ret = 0, with h_voice=1
QL_Voice_Call_AddCommonStateHandler ret = 0
Supported test cases:
0:       print_help
1:       QL_Voice_Call_Ecall
2:       QL_Voice_Call_Ecall_HangUp
3:       QL_Voice_Call_Ecall_UpdateMsd
4:       QL_Voice_Call_Ecall_MsdPush
5:       QL_Voice_Call_Ecall_GetConfigInfo
6:       QL_Voice_Call_Ecall_SetConfigInfo
please input cmd index(-1 exit): 1
please input dest phone number:
15212785764
please input msd content:
123456789
please input ecall mode(1:test 2:emergency):
1
========== Ecall IND EVENT:   ecall_event_establish:0 ==========
 voice_call_id = 1
 ret = 0
please input cmd index(-1 exit): ######### Call id=1, PhoneNum:15212785764, event=DIALING!  ######
######### Call id=1, PhoneNum:15212785764, event=ALERTING!  ######
######### Call id=1, PhoneNum:15212785764, event=ACTIVE!  ######
========== Ecall status  call_id =1 ,  ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

Figure 3: IVS MO ecall

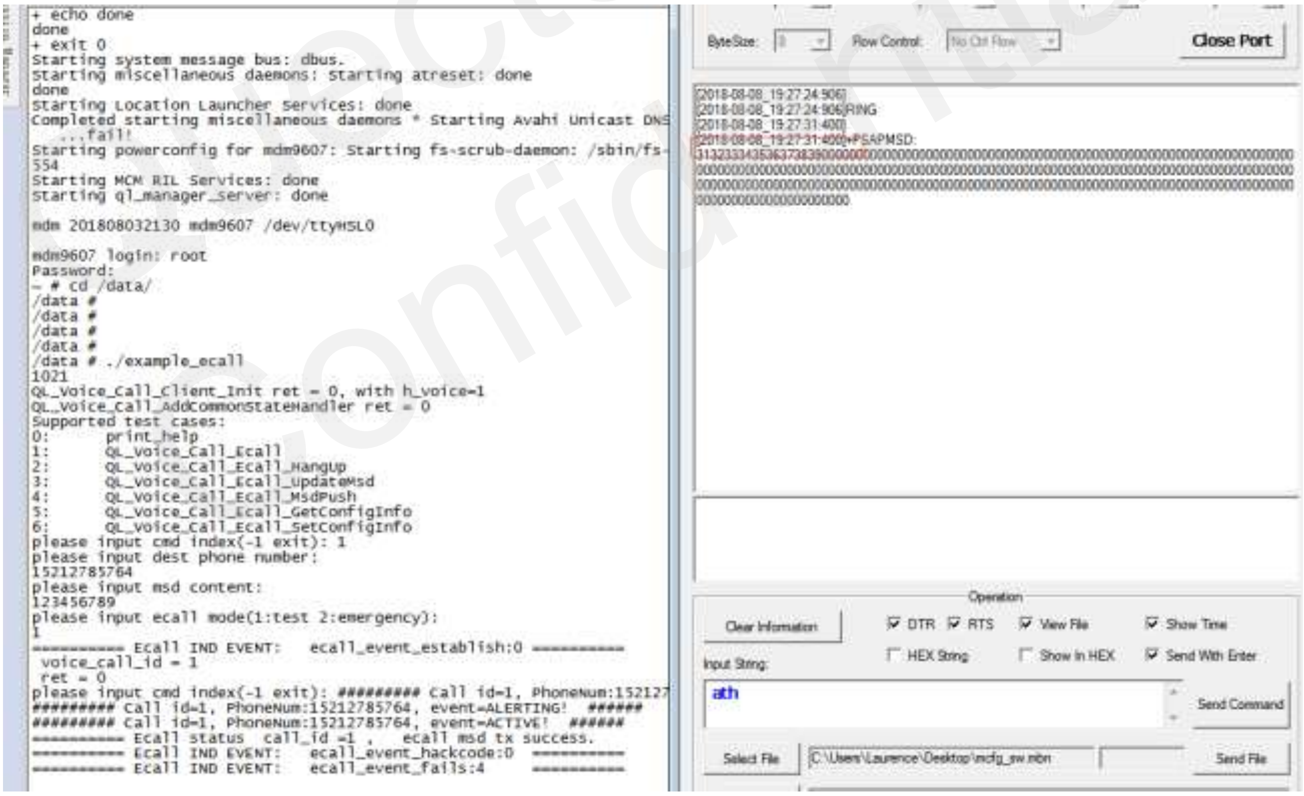
The IVS terminal initiated ECALL, and when the PSAP terminal answered, the IVS sent MSD automatically. After the MSD was successfully received by PSAP, the event was reported to the IVS terminal.



```
/data # ./example_ecall
1021
QL_Voice_Call_Client_Init ret = 0, with h_voice=1
QL_Voice_Call_AddCommonStateHandler ret = 0
Supported test cases:
0:      print_help
1:      QL_Voice_Call_Ecall
2:      QL_Voice_Call_Ecall_HangUp
3:      QL_Voice_Call_Ecall_UpdateMsd
4:      QL_Voice_Call_Ecall_MsdPush
5:      QL_Voice_Call_Ecall_GetConfigInfo
6:      QL_Voice_Call_Ecall_SetConfigInfo
please input cmd index(-1 exit): 1
please input dest phone number:
15212785764
please input msd content:
123456789
please input ecall mode(1:test 2:emergency):
1
========== Ecall IND EVENT:   ecall_event_establish:0 ==========
 voice_call_id = 1
 ret = 0
please input cmd index(-1 exit): ######### Call id=1, PhoneNum:15212785764, event=DIALING!   ######
######### Call id=1, PhoneNum:15212785764, event=ALERTING!   ######
######### Call id=1, PhoneNum:15212785764, event=ACTIVE!   ######
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

call status

Figure 4: IVS Received Event



```
+ echo done
done
+ exit 0
Starting system message bus: dbus.
Starting miscellaneous daemons: Starting atreset: done
done
Starting Location Launcher Services: done
Completed starting miscellaneous daemons * Starting Avahi Unicast DNS
    ...fail!
Starting powerconfig for mdm9607: Starting fs-scrub-daemon: /sbin/fs-
554
Starting MCM RIL Services: done
Starting ql_manager_server: done

mdm 201808032130 mdm9607 /dev/ttyHSL0

mdm9607 login: root
Password:
~ # cd /data/
/data #
/data #
/data #
/data #
/data # ./example_ecall
1021
QL_Voice_Call_Client_Init ret = 0, with h_voice=1
QL_Voice_Call_AddCommonStateHandler ret = 0
Supported test cases:
0:      print_help
1:      QL_Voice_Call_Ecall
2:      QL_Voice_Call_Ecall_HangUp
3:      QL_Voice_Call_Ecall_UpdateMsd
4:      QL_Voice_Call_Ecall_MsdPush
5:      QL_Voice_Call_Ecall_GetConfigInfo
6:      QL_Voice_Call_Ecall_SetConfigInfo
please input cmd index(-1 exit): 1
please input dest phone number:
15212785764
please input msd content:
123456789
please input ecall mode(1:test 2:emergency):
1
========== Ecall IND EVENT:   ecall_event_establish:0 ==========
 voice_call_id = 1
 ret = 0
please input cmd index(-1 exit): ######### Call id=1, PhoneNum:152127
######### Call id=1, PhoneNum:15212785764, event=ALERTING!   ######
######### Call id=1, PhoneNum:15212785764, event=ACTIVE!   ######
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

```
[2018-08-08_19:27:24:906]
[2018-08-08_19:27:24:906]RING
[2018-08-08_19:27:31:400]
[2018-08-08_19:27:31:400]+PSAPMSD:
31323334353637383900000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000
```

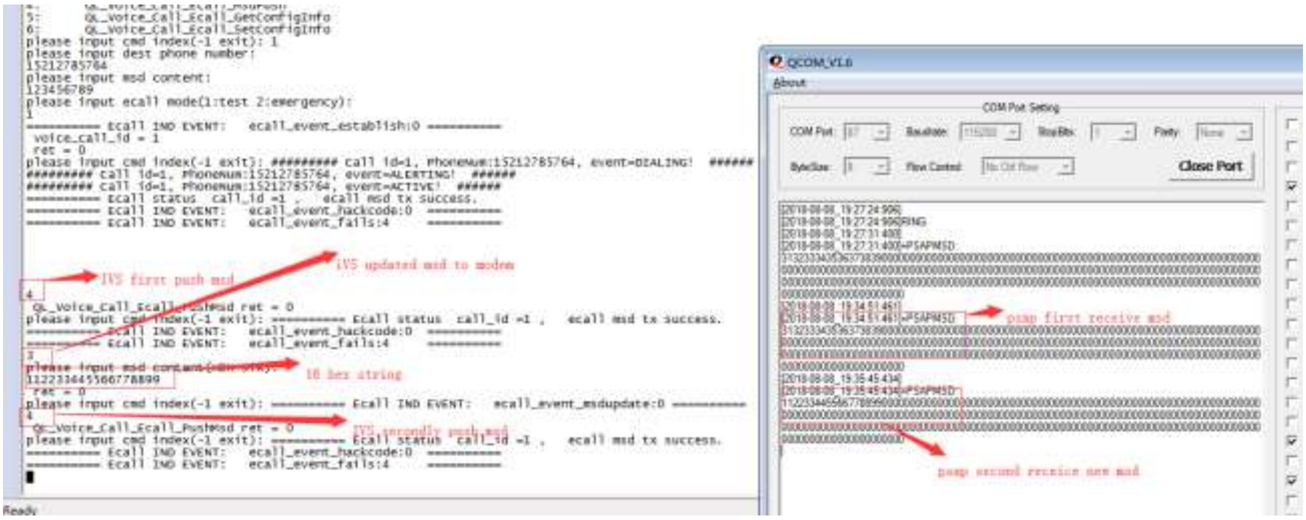
ByteSize: 8
Flow Control: No Ctrl Flow

Operation
☑ DTR ☑ RTS ☑ View File ☑ Show Time
□ HEX String □ Show In HEX ☑ Send With Enter

Input String: ath

C:\Users\Laurence\Desktop\mcfg_sw.mbn

Figure 5: PSAP dispay MSD



```
5:      QL_Voice_Call_Ecall_GetConfigInfo
6:      QL_Voice_Call_Ecall_SetConfigInfo
please input cmd index(-1 exit): 1
please input dest phone number:
15212785764
please input msd content:
123456789
please input ecall mode(1:test 2:emergency):
1
========== Ecall IND EVENT:   ecall_event_establish:0 ==========
 voice_call_id = 1
 ret = 0
please input cmd index(-1 exit): ######### call id=1, PhoneNum:15212785764, event=DIALING!  ######
######### call id=1, PhoneNum:15212785764, event=ALERTING!  ######
######### call id=1, PhoneNum:15212785764, event=ACTIVE!  ######
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========


4
 QL_Voice_Call_Ecall_PushMsd ret = 0
please input cmd index(-1 exit): ========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
3
please input msd content(HEX STR):
11223344556677889 9
 ret = 0
please input cmd index(-1 exit): ========== Ecall IND EVENT:   ecall_event_msdupdate:0 ==========
4
 QL_Voice_Call_Ecall_PushMsd ret = 0
please input cmd index(-1 exit): ========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

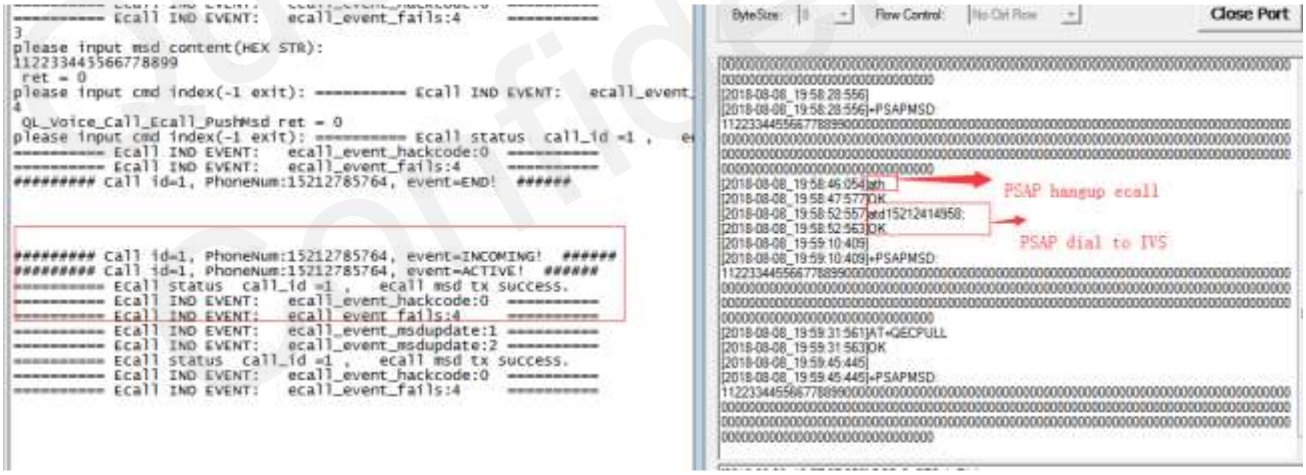
```
[2018-08-08_19:27:24:906]
[2018-08-08_19:27:24:906]RING
[2018-08-08_19:27:31:400]
[2018-08-08_19:27:31:400]+PSAPMSD:
31323334353637383900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000
[2018-08-08_19:34:51:461]
[2018-08-08_19:34:51:461]+PSAPMSD:
31323334353637383900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000
[2018-08-08_19:35:45:434]
[2018-08-08_19:35:45:434]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000
```

Figure 6: IVS PUSH MODE

When IVS and PSAP make a voice call, they can use the PUSH command to send a new MSD again.

4.3. MT ECALL

When eCall has established and PSAP terminates the eCall, MT eCall will be valid in 12 hours. For MT eCall, when eCall is established, MSD can be transferred in push mode or pull mode.



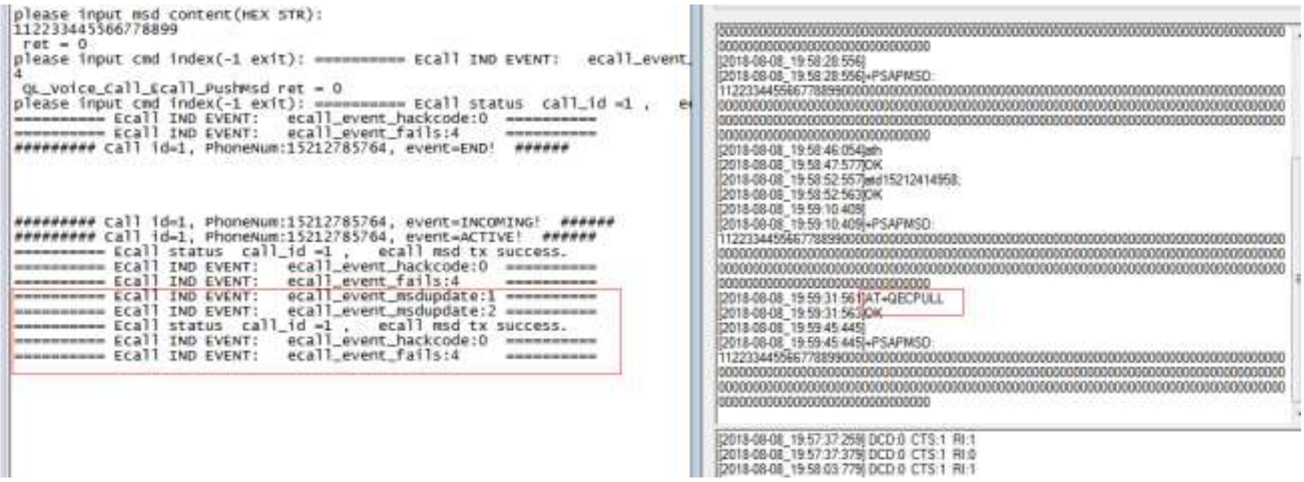
```
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
3
please input msd content(HEX STR):
11223344556677889 9
 ret = 0
please input cmd index(-1 exit): ========== Ecall IND EVENT:   ecall_event_
4
 QL_Voice_Call_Ecall_PushMsd ret = 0
please input cmd index(-1 exit): ========== Ecall status  call_id =1 ,   e
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
######### call id=1, PhoneNum:15212785764, event=END!  ######


######### call id=1, PhoneNum:15212785764, event=INCOMING!  ######
######### call id=1, PhoneNum:15212785764, event=ACTIVE!  ######
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
========== Ecall IND EVENT:   ecall_event_msdupdate:1 ==========
========== Ecall IND EVENT:   ecall_event_msdupdate:2 ==========
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:58:28:556]
[2018-08-08_19:58:28:556]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:58:46:054]ath
[2018-08-08_19:58:47:577]OK
[2018-08-08_19:58:52:557]atd15212414958;
[2018-08-08_19:58:52:563]OK
[2018-08-08_19:59:10:409]
[2018-08-08_19:59:10:409]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:59:31:561]AT+QECPULL
[2018-08-08_19:59:31:563]OK
[2018-08-08_19:59:45:445]
[2018-08-08_19:59:45:445]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
```

Figure 7: PSAP ORI CALL

Figure 7. The PSAP initiate a call to IVS, and IVS began to send MSD after answer the call automatically. After successful transmission, IVS can make voice call with PSAP.



```
please input msd content(HEX STR):
112233445566778899
 ret = 0
please input cmd index(-1 exit): ========== Ecall IND EVENT:   ecall_event_
4
 QL_Voice_Call_Ecall_PushMsd ret = 0
please input cmd index(-1 exit): ========== Ecall status  call_id =1 ,   e
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
######### call id=1, PhoneNum:15212785764, event=END!  ######



######### call id=1, PhoneNum:15212785764, event=INCOMING!  ######
######### call id=1, PhoneNum:15212785764, event=ACTIVE!  ######
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
========== Ecall IND EVENT:   ecall_event_msdupdate:1 ==========
========== Ecall IND EVENT:   ecall_event_msdupdate:2 ==========
========== Ecall status  call_id =1 ,   ecall msd tx success.
========== Ecall IND EVENT:   ecall_event_hackcode:0  ==========
========== Ecall IND EVENT:   ecall_event_fails:4     ==========
```

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:58:28:556]
[2018-08-08_19:58:28:556]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:58:46:054]ath
[2018-08-08_19:58:47:577]OK
[2018-08-08_19:58:52:557]atd15212414958;
[2018-08-08_19:58:52:563]OK
[2018-08-08_19:59:10:409]
[2018-08-08_19:59:10:409]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000
[2018-08-08_19:59:31:561]AT+QECPULL
[2018-08-08_19:59:31:563]OK
[2018-08-08_19:59:45:445]
[2018-08-08_19:59:45:445]+PSAPMSD:
11223344556677889900000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000

[2018-08-08_19:57:37:259] DCD:0  CTS:1  RI:1
[2018-08-08_19:57:37:379] DCD:0  CTS:1  RI:0
[2018-08-08_19:58:03:779] DCD:0  CTS:1  RI:1
```

Figure 8: PSAP PULL Mode

Figure 8 PSAP PULL mode, PSAP can send the PULL command to request the IVS to send MSD content.

5 eCall Compiling Introduction

This chapter is the introductions of compiling single example_ecall.c.

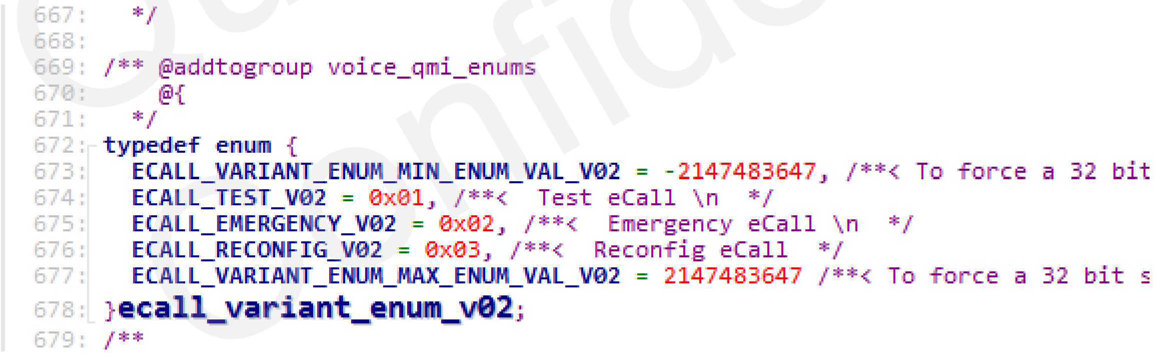
1. Unzip tar -jxvf ql-ol-sdk.tar.bz2: tar -jxvf ql-ol-sdk.tar.bz2
2. Enter ql-ol-sdk: cd cd ql-ol-sdk
3. EnsureSDK version is same as firmware version: source ql-ol-crosstool/ql-ol-crosstool-env-init
4. Execute: cd ql-ol-extsdk/example/ecall
5. Execute: make clean;make

6 ECALL Test Introduction

eCall is an eSafety initiative proposed by the European Commission. It can also be seen from Figure 1 that the real environment consists of the Accident Vehicle (IVS) and the Call for Rescue Center (PSAP). Because the domestic does not have such a testing environment, currently Quectel uses the simulation PSAP for testing. The PSAP is compiled from the Qualcomm code and used to do in-band modem testing. For the PSAP, only ordinary sim cards are used for simulation. Therefore, ECALL test only can be done in the TEST mode.

The following is a brief introduction to the function testing which Quectel current supports and the environment required for the testing.

6.1. Test Mode



```
  */

/** @addtogroup voice_qmi_enums
    @{
  */
typedef enum {
  ECALL_VARIANT_ENUM_MIN_ENUM_VAL_V02 = -2147483647, /**< To force a 32 bit
  ECALL_TEST_V02 = 0x01, /**<  Test eCall \n  */
  ECALL_EMERGENCY_V02 = 0x02, /**<  Emergency eCall \n  */
  ECALL_RECONFIG_V02 = 0x03, /**<  Reconfig eCall  */
  ECALL_VARIANT_ENUM_MAX_ENUM_VAL_V02 = 2147483647 /**< To force a 32 bit s
}ecall_variant_enum_v02;
/**
```

Figure 9: ECALL-VARIANT Introduction

When the parameter configuration of ecall is TEST mode, the in-band modem test of IVS and analog PSAP can be performed. At this time, the numbers of PSAP can be ordinary (except CDMA). When in a real environment, this value must be configured as ECALL_EMERGENCY mode so that the called number is 112.

The following is mainly about the functions testing that Quectel currently support and the environment required for testing.

PSAP

Currently, AG35 ecall can take version UC20AQAR02A02M1024_BETA1202_PSAP or EC20CEFAG_PSAP. Please pay attention that MIC voice channel of this version is off by default, use AT + QECVOC = 0 to open (the IVS PUSH command will temporarily failure).

6.2.

NOTES

1. The PSAP is in the automatic answer mode. For external calls, the PSAP will automatically answer the call by default.
2. Ensure that the CODEC exists and confirm that the voice call functioning is normal before testing.
3. Testing ECALL under CDMA network is not supported.
4. Use ordinary SIM cards.

IVS

6.3.

NOTES

1. Ensure that the CODEC exists and confirm that the voice call functioning is normal before testing.
2. Testing ECALL under CDMA network is not supported.
3. Use ordinary SIM cards.

6.4.

Ecall Function Testing Items

6.4.1 Whether MSD was Sent Successfully

Use event E_QL_MCM_VOICE_ECALL_STATUE_IND to check whether IVS sent MSD successfully or not.

6.4.2 Whether PSAP Receive Correct MSD

To Verify whether PSAP receive correct MSD.

6.4.3 Whether Voice is normal

Verify that after receiving MSD, whether the IVS will automatically switch the voice (ie, normal voice call).

ECALL Testing

After SIM cards insert to IVS and PSAP, in-band modem testing under TEST mode can be performed.

6.5 6.5.1 Calling Ecall Testing

The calling test is initiated from the IVS (AG35) to the PSAP, please refer to Chapter 4.2.

6.5.2 Called Ecall Testing

The called party is that the IVS (AG35) to answer the call from the PSAP. The test condition is the previous calling ecall test is actively hanged up by the PSAP, and then the PSAP initiates the ECALL to the IVS, please refer to Chapter 4.3.

6.5.3 MSD Introduction

Quectel does not pay attention to MSD content and format, its content and format are filled by customers.

6.6. Other Introduction

The ECALL protocol does not support CDMA. Because the ECALL function test cannot be performed using the CSIM card, it is recommended that the IVS and PSAP terminals use the same operators' SIM card.

7 ECALL Callback Event

ECALL event is reported by E_QL_MCM_VOICE_ECALL_URC_IND, E_QL_MCM_VOICE_ECALL_STATUE_IND and E_QL_MCM_VOICE_CALL_IND.

eCALL event	
E_QL_MCM_VOICE_CALL_IND	Report voice status.
E_QL_MCM_VOICE_ECALL_STATUE_IND	Report the sending statue of MSD,
E_QL_MCM_VOICE_ECALL_URC_IND	Report URC event (No longer recommended and maintained, please refer to Chapter 7.1-7.3.)
E_QL_MCM_VOICE_ECALL_EVENT_IND	Report ECALL event, its function is same as URC event (Recommended, please refer to Chapter 7.4)

7.1. URC of Failed eCALL

If the eCall setup fails or MSD sends timeout, the event will be reported. When IVS cannot get the synchronous frame, the IVS will reset.

URC Failed of eCall	
+QECURC: 0,<errorcode>	<errorcode> indicates error causes

Parameters

<errorcode>	Error code 0 Establishing eCall fails 1 Wait for START timeout 2 Wait for HACK timeout 3 MSD transmission timeout 4 IVS reset

7.2 URC of Successful eCall MSD Transmission

When the MSD sends successfully, this event will be reported. At this time PUSH API can be called to send the MSD again.

URC of Successful eCall MSD Transmission	
+QECURC: 1,0	Establish eCall successfully.
+QECURC: 1,1,<hackcode>	When IVS Link Layer receives 2 HACK messages, this URC will be reported to IVS APP. <hackcode> indicates HACK code which is transferred by PSAP.
+QECURC: 1,2,<pullstatus>	<pullstatus> indicates the status of MSD request.

Parameters

<hackcode> HACK code which is transferred by PSAP. For details, please refer to EN 16062

<pullstatus> URC of updating eCall MSD

0 Indicate timeout of updating MSD and module starts to transfer the old MSD

1 Indicate to update MSD in 5 seconds

7.3. URC of eCall Process Information

URC of eCall Process Information	
+QECURC: 2,<processinfo>	<processinfo> indicates eCall process information

Parameters

<processinfo> URC of eCall process information

1 When IVS Link Layer receives START message and starts to send MSD, this URC will be reported

2 When IVS Link Layer receives NACK message, this URC will be reported

3 When IVS Link Layer receives the first LACK message, this URC will be reported

4 When IVS Link Layer receives the second LACK message, this URC will be reported

5 When IVS Link Layer receives first HACK message, this URC will be reported

6 Indicates MSD has been updated

7.4 EVENT of eCall

The first 3 subsections are no longer recommended, and the ECALL API event has been redesigned and will replace the URC event. Please refer to ql_mcm_voice.c

ql_mcm_voice_ecall_event_ind	
E_QL_MCM_VOICE_ECALL_EVENT_FAIL_T	event of eCall Failed.
E_QL_MCM_VOICE_ECALL_EVENT_PROCESS_T	event of eCall process.
E_QL_MCM_VOICE_ECALL_EVENT_MSDUPDATE_T	event of ecall msd update.
E_QL_MCM_VOICE_ECALL_EVENT_ESTABLISH_T	event of eCall establish.
ql_mcm_voice_ecall_event_hackcode_t	event of eCall hack code.
ql_mcm_voice_ecall_event_originate_fail_and_redial_t	event of originate fail and redial.
ql_mcm_voice_ecall_event_drop_and_redial_t	event of drop and redial.