



VoIP AT Command Interface

Application Note

80000NT12322A Rev.0
2026-09-11
Released
Confidential



Contents

1	Applicability Table.....	4
2	Introduction	5
2.1	Scope	5
2.2	Audience	5
2.3	Contact Information, Support	5
2.4	Conventions	5
2.5	Terms and Conditions	6
2.6	Disclaimer	6
3	Operational Overview	7
3.1	Logic Introduction	7
3.2	Recommended Startup Sequence	7
3.3	Lifecycle Behavior	7
4	AT Commands Reference.....	8
4.1	AT#VOIPDVI – Configure Audio and DVI codec	8
4.2	AT#VOIPSTART - Start the VoIP engine	9
4.3	AT#VOIPSTOP - Stop the VoIP engine	10
4.4	AT#VOIPSTATUS - Read the PJSUA state	11
4.5	AT#VOIPACCOUNT - Configure a SIP account	12
4.6	AT#VOIPACCOUNTDEL - Delete a SIP account	14
4.7	AT#VOIPREG - Control SIP registration	15
4.8	AT#VOIPBUDDY - Manage buddies	16
4.9	AT#VOIPDIAL - Originate a call	18
4.10	AT#VOIPANS - Answer or reject an incoming call	20
4.11	AT#VOIPHUP - Terminate a call	22
4.12	AT#VOIPDTMF - Send DTMF digits	24
4.13	AT#VOIPCODEC - List or modify codecs	25
4.14	AT#VOIPSETCFG - Set a runtime configuration value	27
4.15	AT#VOIPGETCFG - Read a runtime configuration value	29
4.16	AT#VOIPONLINESTAT - Read or toggle basic presence	31
4.17	AT#VOIPONLINESTATEXT - Set extended presence activity	33
4.18	AT#VOIPPURC - Enable or disable notifications	35
5	Runtime Configuration Parameters	36
6	Unsolicited Result Codes.....	37
7	Usage Examples	38
7.1	Configure, start, and place a call	38
7.2	Incoming call and answer	38
7.3	Send DTMF and hang up	38
8	VoIP Application Architecture	39
8.1	Two-binary applications design	39
8.1.1	AT client responsibilities	39
8.1.2	VoIP server responsibilities	39
8.1.3	Client/server communication overview	40
8.2	Applications execution and logging	40
9	Acronyms and Abbreviations.....	42
10	Related Documents.....	43

11	Document History	44
----	------------------------	----



1 Applicability Table

Table 1: Applicability Table

Products
LE910Cx Linux family, firmware version 25.21.xx5 and above

2 Introduction

2.1 Scope

This application note describes the custom AT command interface used to configure and operate a Voice over IP (VoIP) service built on the PJSUA engine. It focuses on the external AT command behavior: syntax, parameters, responses, configuration values, unsolicited result codes, and representative usage sequences.

2.2 Audience

This document is intended for system integrators, firmware developers, validation engineers, and technical support personnel integrating or testing the custom VoIP AT interface.

2.3 Contact Information, Support

For technical support and general questions, e-mail:

- TS-EMEA@telit.com
- TS-AMERICAS@telit.com
- TS-APAC@telit.com
- TS-SRD@telit.com
- TS-ONEEDGE@telit.com
- TS-LOCATION@telit.com

Alternatively, use: <https://www.telit.com/contact-us/>

For Product information and technical documents, visit: <https://www.telit.com>

2.4 Conventions

Note: Provides advice and suggestions that may be useful when integrating the module.

Danger: This information MUST be followed, or catastrophic equipment failure or personal injury may occur.

Warning: Alerts the user on important steps about the module integration.

All dates are in ISO 8601 format, that is YYYY-MM-DD.

2.5 Terms and Conditions

Refer to <https://www.telit.com/terms-conditions/>.

2.6 Disclaimer

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY. TELIT CINTERION RESERVES THE RIGHT TO MAKE CHANGES TO THE PRODUCTS DESCRIBED HEREIN. THE SPECIFICATIONS IN THIS DOCUMENT ARE SUBJECT TO CHANGE AT THE DISCRETION OF TELIT CINTERION WITHOUT PRIOR NOTICE. THIS DOCUMENT IS PROVIDED ON "AS IS" BASIS ONLY AND MAY CONTAIN DEFICIENCIES OR INADEQUACIES. TELIT CINTERION DOES NOT ASSUME ANY LIABILITY FOR INFORMATION PROVIDED IN THE DOCUMENT OR ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT DESCRIBED HEREIN.

TELIT CINTERION GRANTS A NON-EXCLUSIVE RIGHT TO USE THE DOCUMENT. THE RECIPIENT SHALL NOT COPY, MODIFY, DISCLOSE, OR REPRODUCE THE DOCUMENT EXCEPT AS SPECIFICALLY AUTHORIZED BY TELIT CINTERION.

TELIT CINTERION AND THE TELIT CINTERION LOGO, ARE TRADEMARKS OF TELIT CINTERION AND ARE REGISTERED IN CERTAIN COUNTRIES. ALL OTHER REGISTERED TRADEMARKS OR TRADEMARKS MENTIONED IN THIS DOCUMENT ARE THE PROPERTY OF THEIR RESPECTIVE OWNERS AND ARE EXPRESSLY RESERVED BY TELIT CINTERION (AND ITS LICENSORS).

3 Operational Overview

3.1 Logic Introduction

The implementation is split into two executable binaries. The *AT client* receives and validates custom AT commands, translates them into internal requests, and presents synchronous responses and asynchronous indications. The *server* receives those requests through POSIX message queues and manages the whole logic, relying on <https://github.com/pjsip/pjproject/> source code for VoIP engine initialization, SIP accounts, registration, calls, media, presence, buddies, codecs, and notifications. In the current implementation, **pjproject** release 2.17 is used.

See chapter [Two-binary applications design](#) and following for further details.

3.2 Recommended Startup Sequence

Configuration can be provided before the VoIP engine is started. The server keeps a local account, buddy, and runtime configuration cache. **AT#VOIPSTART** initializes PJSUA from this cache, starts the engine, and applies post-start settings.

```
AT#VOIPDVI
AT#VOIPURC=1
AT#VOIPSETCFG="LOG_LEVEL",4
AT#VOIPSETCFG="MIC_VOL",100
AT#VOIPSETCFG="SPK_VOL",100
AT#VOIPACCOUNT="<registrar>","<sip_uri>","<realm>","<username>","<password>"
AT#VOIPSTART
```

3.3 Lifecycle Behavior

AT#VOIPSTART performs initialization and starts PJSUA when it is not already running.

AT#VOIPSTOP destroys the PJSUA subsystem.

In the supplied implementation, local parameters are retained across stop/start cycles.

4 AT Commands Reference

The following sections describe each supported AT command, its parameters, usage and examples.

4.1 AT#VOIPDVI – Configure Audio and DVI codec

Configures the DVI audio codec and sets the audio channels for further usage. The current implementation supports only EVB2's Maxim9867 codec.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPDVI[=<id>]

Execution command sets the DVI corresponding to the provided ID. If called without parameters, the default Maxim9867 is set.

Parameters:

Name	Type	Default	Description
<id>	Integer	-	0 = Maxim9867



AT#VOIPDVI=?

Test command reports the accepted parameter forms.

Note:

On PL801 in EVB2, ensure that the jumpers connect the codec's I2C lines to B2B endpoints.

4.2 AT#VOIPSTART - Start the VoIP engine

Initializes and starts the PJSUA subsystem. Stored account, buddy, and runtime configuration values are loaded during initialization.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPSTART

Execution command creates, initializes and starts the VoIP engine.



AT#VOIPSTART=?

Test command returns the command availability

Note:

A working PDP Context must be in place for the internal engine to communicate with the VoIP server. It can be enabled with AT#SGACT command, for example.



4.3 AT#VOIPSTOP - Stop the VoIP engine

Stops the PJSUA subsystem and releases runtime resources.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPSTOP

Execution command stops the VoIP engine.



AT#VOIPSTOP=?

Test command returns the command availability

4.4 AT#VOIPSTATUS - Read the PJSUA state

Returns the numeric PJSUA operational state.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPSTATUS

Execution command returns the current PJSUA engine state.



AT#VOIPSTATUS?

Read command returns the current PJSUA engine state

#VOIPSTATUS: <state>

OK

Note: The returned value is the numeric result of pjsua_get_state().

<state> values:

0	:	Library not initialized
1	:	Internal structures created
2	:	Internal structured initialized
3	:	Engine starting
4	:	Engine running
5	:	Engine stopping

4.5 AT#VOIPACCOUNT - Configure a SIP account

Stores the SIP registrar, local SIP identity, realm, authentication username, and password. If PJSUA engine is already running, the corresponding account is created or modified immediately.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPACCOUNT=<registrar>,<sip_uri>,<realm>,<username>,<password>

Execution command sets the account parameters.

Parameters:

Name	Type	Default	Description
<registrar>	string	-	SIP registrar URI. (maximum length 127 characters)
<sip_uri>	string	-	Local SIP identity URI. (maximum length 127 characters)
<realm>	string	-	Digest authentication realm; "*" may be used where supported (maximum length 127 characters)
<username>	string	-	Authentication username. (maximum length 127 characters)
<password>	string	-	Authentication password. (maximum length 127 characters)



AT#VOIPACCOUNT?

Read command returns the set parameters

#VOIPACCOUT: "REGISTRAR": "sip:proxy.example.com"

#VOIPACCOUNT: "SIP_URI": "sip:1234@server.com"

...

OK

Note: If PJSUA is already running (that is, AT#VOIPSTART was already executed) the query command will also report as first row the internal account id number.

Warning: The read implementation returns account data, including the password. Protect access to the AT interface accordingly.



AT#VOIPACCOUNT=?

Test command reports the command parameters ranges

</> Example of command execution:

AT#VOIPSTART

OK

AT#VOIPACCOUNT="sip:proxy.example.com","sip:alice@example.com","*",
"alice","password"

#VOIPACCOUNT: <account_id>

OK

Note: The setting command returns an account id (which can be used for later configuration) only if AT#VOIPSTART was already executed. If called before AT#VOIPSTART, no account id will be provided. It can be retrieved later using the query command.

4.6 AT#VOIPACCOUNTDEL - Delete a SIP account

Deletes an account from PJSUA when running and clears the locally stored account configuration.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPACCOUNTDEL=<account_id>

Execution command deletes the selected account

Parameters:

Name	Type	Default	Description
<account_id>	Integer	-	Account identifier, range 2 to 65535. Special value 0 selects the current account.



AT#VOIPACCOUNTDEL=?

Test command returns the parameter ranges

4.7 AT#VOIPREG - Control SIP registration

Registers or unregisters the current default SIP account.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPREG=<mode>

Execution command controls the registration state

Parameters:

Name	Type	Default	Description
<mode>	Integer	-	The mode to be set. 0: unregister; 1: register/re-register



AT#VOIPREG?

Read command returns the registration state, the last SIP response code, and reason text.

AT#VOIPREG?

#VOIPREG: <state>,<code>,"<reason>"

OK



AT#VOIPREG=?

Test command reports the command parameter ranges

4.8 AT#VOIPBUDDY - Manage buddies

Adds a SIP URI to the buddy list, removes an entry, or returns the configured list. Public buddy identifiers are one-based.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPBUDDY=<cmd>,<parameter>

Execution command allows to add or remove buddies

Parameters:

Name	Type	Default	Description
<cmd>	String	-	The command to be executed. "ADD": add a new buddy SIP URI. "REM": remove the provided buddy ID.
<parameter>	-		The additional parameter. If <cmd> is "ADD", it is a string representing the buddy SIP URI (max length: 127 characters). If <cmd> is "REM", it represent the buddy ID number (returned by the "ADD" command), starting from 1.



AT#VOIPBUDDY?

Read command returns the buddies list

AT#VOIPBUDDY?

#VOIPBUDDY: <id>,"<sip_uri>"

...

OK

Note: If PJSUA is already running (that is, AT#VOIPSTART was already executed) the query command will report the whole list of buddies already registered. When PJSUA is stopped, a successfully added URI is stored locally and returned by the read form; it is loaded when the engine starts.



AT#VOIPBUDDY=?

Test command reports accepted operations and parameter limits.



Example of commands execution:

```
AT#VOIPBUDDY="ADD","sip:bob@example.com"
```

```
#VOIPBUDDY: 1
```

```
OK
```

```
AT#VOIPBUDDY?
```

```
#VOIPBUDDY: 1:"sip:bob@example.com"
```

```
OK
```

4.9 AT#VOIPDIAL - Originate a call

Starts an outgoing SIP call using either a destination SIP URI or a configured buddy identifier. On success the server returns the assigned call identifier.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPDIAL=<remote>

Execution command allows to call a SIP URI or a configured buddy. Upon success, the call ID number (starting from 0) is returned.

Parameters:

Name	Type	Default	Description
<remote>	String	-	Can be both a SIP URI (max length: 127) or an existing configured buddy ID number (starting from 1)



AT#VOIPDIAL=?

Test command reports the accepted parameter forms.

Note: The PJSUA engine must be running when calling this command



Example of commands execution:

```
AT#VOIPDIAL="sip:bob@example.com"
#VOIPDIAL: <call_id>
OK
AT#VOIPDIAL=2
```

```
.....  
.....  
#VOIPDIAL: <call_id>
```

```
OK  
.....  
.....
```



4.10 AT#VOIPANS - Answer or reject an incoming call

Sends a SIP response for the current pending incoming call. If only the call identifier is supplied, the handler default response code is 200 (OK).

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPANS=<call_id>[,<code>[,<reason>]]

Execution command allows to respond to a pending incoming call

Parameters:

Name	Type	Default	Description
<call_id>	Integer	-	The call ID (received from the URC message if URCs are active).
<code >	Integer	200	Optional SIP response code. 100 to 699
<reason>	String	-	Optional SIP reason text, max 127 characters



AT#VOIPANS=?

Test command reports the accepted parameter ranges.

Note: Code values can be in the ranges:

100-199: Provisional Keep call setup in progress

200-299: Success Accept the call

300-399: Redirection Redirect the caller

400-499: Client error Reject due to caller-side problem

500-599: Server error Reject due to server-side problem

600-699: Global failure Reject everywhere



Example of commands execution:

```
AT#VOIPANS=0,200
```

```
OK
```

```
AT#VOIPANS=0,486,"Busy Here"
```

```
OK
```

4.11 AT#VOIPHUP - Terminate a call

Terminates the selected call. If no valid identifier is supplied internally, the current call is used.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPHUP[=<call_id>][,<code>[,<reason>]]

Execution command allows to terminate a running call

Parameters:

Name	Type	Default	Description
<call_id>	Integer	-	The call ID (received from the URC message if URCs are active). If no parameter is provided, the current call is terminated
<code>	Integer	0	Optional SIP response code. 100 to 699. 0 indicates a normal termination.
<reason>	String	-	Optional SIP reason text, max 127 characters



AT#VOIPHUP=?

Test command reports the accepted parameter ranges.



Example of commands execution:

```
AT#VOIPHUP=0
OK
```

.....
.....

AT#VOIPHUP

OK

.....
.....



4.12 AT#VOIPDTMF - Send DTMF digits

Queues DTMF digits for transmission on the current active call.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPDTMF=<digits>

Execution command sends DTMF digits on the active call

Parameters:

Name	Type	Default	Description
<digits>	String	-	DTMF digit sequence. Max length: 127



AT#VOIPDTMF=?

Test command reports the accepted parameter ranges.

Note:

The command fails when there is no active call or when media is not established.



Example of commands execution:

```
AT#VOIPDTMF="123#"
OK
```

4.13 AT#VOIPCODEC - List or modify codecs

Returns available audio codecs or changes the priority of a codec identified by name and clock rate.

PJSUA must be running (that is, AT#VOIPSTART was already executed) or the command will fail.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPCODEC=<priority>,<name>,<clock_rate>

Execution command changes the priority of a codec.

Parameters:

Name	Type	Default	Description
<priority>	Integer	-	Codec priority; 0 disables a codec. Range: 0-255. The higher the value, the higher the priority.
<name>	String	-	Codec name. Max length: 127
<clock_rate>	Integer	-	Codec clock rate in Hz. Ranges 8000-44100



AT#VOIPCODEC?

Read command returns the available codecs list.

The VoIP engine must be running. The list response includes priority, name, clock rate, and channel count.

AT#VOIPCODEC?

#VOIPCODEC: <priority>,<name>,<clock_rate>,<channels>

...

OK



AT#VOIPCODEC=?

Test command reports the accepted parameter range.



Example of commands execution:

AT#VOIPCODEC=100,"speex",8000

OK

Sets "speex" codec, at 8kHz, to priority 100.

4.14 AT#VOIPSETCFG - Set a runtime configuration value

Stores and, where supported, immediately applies a named configuration parameter.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPSETCFG=<tag>,<value>

Execution command sets the option to the requested value. Values are locally stored and applied when PJSUA engine is started.

Parameters:

Name	Type	Default	Description
<tag>	String	-	Option Tag. "LOG_LEVEL": internal log level "LOG_COLOR": enable colored logs "LOG_FILE": enable logs to file "MIC_VOL": microphone volume percentage. "SPK_VOL": speaker volume percentage. "EC_OPT": Echo canceller option. "EC_TAIL": Echo canceller tail length, in milliseconds
<value>	String	-	Value accepted by the selected tag, typically a numeric value "LOG_LEVEL": 0-6 "LOG_COLOR": 0-1

"MIC_VOL": 0-300.
"SPK_VOL": 0-300.
"EC_OPT" :
- 0 : default
- 1 : speex
- 2 : simple echo
 suppressor
- 3: WebRTC
- 4: WebRTC AEC3
"EC_TAIL": 0-1000



AT#VOIPSETCFG=?

Test command lists the accepted tags and their value ranges.

Note: The command will automatically apply the selected option to PJSUA if it is running.



Example of commands execution:

```
AT#VOIPSETCFG="LOG_LEVEL",4
OK
AT#VOIPSETCFG="SPK_VOL",150
OK
```

4.15 AT#VOIPGETCFG - Read a runtime configuration value

Returns the current value associated with a named configuration tag.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPGETCFG=<tag>

Execution command returns the specified tag value

Parameters:

Name	Type	Default	Description
<tag>	String	-	Option Tag. "LOG_LEVEL": internal log level "LOG_COLOR": enable colored logs "LOG_FILE": enable logs to file "MIC_VOL": microphone volume percentage. "SPK_VOL": speaker volume percentage. "EC_OPT": Echo canceller option. "EC_TAIL": Echo canceller tail length, in milliseconds



AT#VOIPGETCFG?

Read command returns the available tags list and their currently set value.

AT#VOIPGETCFG?

#VOIPGETCFG: "LOG_LEVEL": "0"

...

OK



AT#VOIPGETCFG=?

Test command lists the accepted tags.

Note:

The command will automatically apply the selected option to PJSUA if it is running.



Example of commands execution:

```
AT#VOIPGETCFG="MIC_VOL"
```

```
#VOIPGETCFG: "MIC_VOL","120"
```

```
OK
```

```
AT#VOIPGETCFG="LOG_COLOR"
```

```
#VOIPGETCFG: "LOG_COLOR","1"
```

```
OK
```

4.16 AT#VOIPONLINESTAT - Read or toggle basic presence

Reads the current online/offline presence state, or set /toggle its value

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPONLINESTAT[=<status>]

Execution command sets the required online

Parameters:

Name	Type	Default	Description
<status>	Integer	-	0: offline 1: online

Note: If no parameter is passed, the current state will be toggled to the opposite value



AT#VOIPONLINESTAT?

Read command returns the current online state

AT#VOIPONLINESTAT?

#VOIPONLINESTAT: 1

OK



AT#VOIPONLINESTAT=?

Test command lists the accepted values range.

Note: The command will fail if PJSUA is not running.



Example of commands execution:

```
AT#VOIPONLINESTAT=1
OK
```

```
AT#VOIPONLINESTAT?
#VOIPONLINESTAT: 1
OK
```

```
AT#VOIPONLINESTAT
OK
```

```
AT#VOIPONLINESTAT?
#VOIPONLINESTAT: 0
OK
```

4.17 AT#VOIPONLINESTATETEXT - Set extended presence activity

Sets or reads the RPID activity of the current account.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPONLINESTATETEXT=<activity>

Execution command sets the required online activity state

Parameters:

Name	Type	Default	Description
<activity>	Integer	-	Activity state as an integer: 0 : available 1 : busy 2 : on the phone 3 : idle 4 : away 5 : be right back 6 : offline



AT#VOIPONLINESTATETEXT?

Read command returns the current online RPID state

AT#VOIPONLINESTATETEXT?

#VOIPONLINESTATETEXT: 0

OK



AT#VOIPONLINESTATETEXT=?

Test command lists the accepted values range.

Note: The command will fail if PJSUA is not running.



Example of commands execution:

```
AT#VOIPONLINESTATEXT=1
```

```
OK
```

```
AT#VOIPONLINESTATEXT?
```

```
#VOIPONLINESTATEXT: 1
```

```
OK
```

4.18 AT#VOIPURC - Enable or disable notifications

Controls forwarding of normal VoIP URCs and returns the current notification state.

SIM Presence	Setting saved	Can be aborted
Not required	Other	No



AT#VOIPURC=<mode>

Execution command enables/disables the URC notification.

Parameters:

Name	Type	Default	Description
<mode>	Integer	0	URC state 0 : disabled 1 : enabled



AT#VOIPURC?

Read command returns the current online URCs configuration state

AT#VOIPURC?

#VOIPURC: 0

OK



AT#VOIPURC=?

Test command lists the accepted values range.

5 Runtime Configuration Parameters

Tag	Range	Meaning	Runtime behavior
LOG_LEVEL	0 to 6	PJSUA logging verbosity.	Reconfigures logging when running.
LOG_COLOR	0 or 1	Disable or enable colored PJSUA log decoration. It works only when the prints are on stdout.	Reconfigures logging when running.
LOG_FILE	256 characters	If passed, pjsua logs will also be saved in a file with LOG_FILE path	Reconfigures logging when running.
MIC_VOL	0 to 300	Microphone's receive-level percentage. Default 100.	Applied to conference port 0; values above 300 are clamped.
SPK_VOL	0 to 300	Speaker transmit-level percentage. Default 100.	Applied to conference port 0; values above 300 are clamped.
EC_OPT	0 to 4	Echo-canceller option selector.	Non-zero values add the software echo flag; applied with pjsua_set_ec().
EC_TAIL	0 to 1000	Echo-canceller tail length.	Applied with pjsua_set_ec().

6 Unsolicited Result Codes

When AT#VOIPURC=1, event callbacks can generate #VOIPURC indications. The AT client formats the event name, object identifier, event data, and optional extra data received on the URC queue.

Event	Format	Meaning
INCOMING	#VOIPURC: "INCOMING",<call_id>,<remote>"	Incoming call. Remote data is built from the display name, when present, and SIP user.
CALLSTATE	#VOIPURC: "CALLSTATE",<call_id>,<state>"[,<sip_code>]"	Call state change. DISCONNECTED includes the last SIP status code.
REG	#VOIPURC: "REG",<account_id>,<reason>,<sip_code>"	Registration-state callback result.
DTMF	#VOIPURC: "DTMF",<call_id>,<digit>[duration]"	Incoming DTMF. SIP INFO can include duration text.

7 Usage Examples

7.1 Configure, start, and place a call

```
AT#VOIPDVI
OK

AT#VOIPURC=1
OK

AT#VOIPSETCFG="LOG_LEVEL",4
OK

AT#VOIPACCOUNT="sip:proxy.example.com","sip:alice@example.com","*", "alice", "password"
#VOIPACCOUNT: <account_id>
OK

AT#VOIPSTART
OK

AT#VOIPDIAL="sip:bob@example.com"
#VOIPDIAL: <call_id>
OK
```

7.2 Incoming call and answer

```
#VOIPURC: "INCOMING",0,"Bob <sip:bob>"

AT#VOIPANS=0,200
OK

#VOIPURC: "CALLSTATE",0,"CONFIRMED"
```

7.3 Send DTMF and hang up

```
AT#VOIPDTMF="123#"
OK

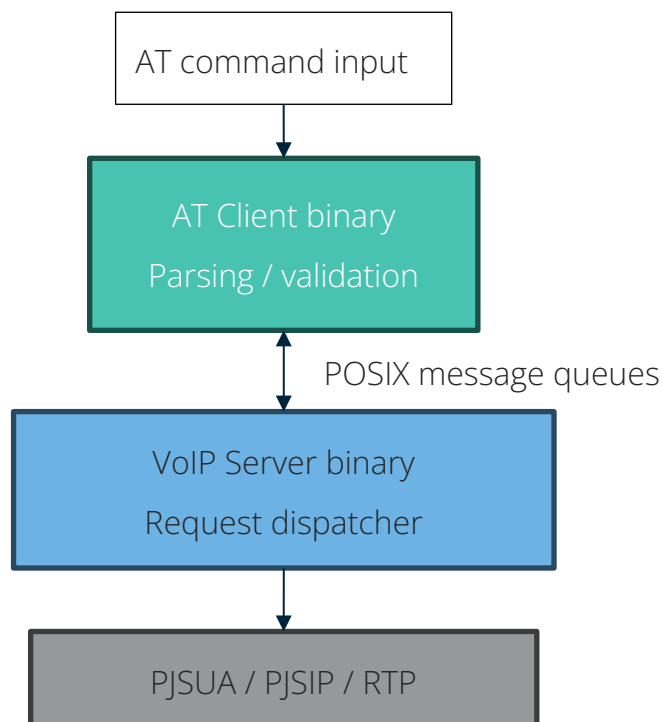
AT#VOIPHUP=0
OK

#VOIPURC: "CALLSTATE",0,"DISCONNECTED",200
```

8 VoIP Application Architecture

8.1 Two-binary applications design

The application consists of an AT client binary and a VoIP server binary. The AT client registers the AT#VOIP... handlers, parses each command form, validates command parameters, constructs a fixed-size message, and waits for the matching response. The server owns the PJSUA runtime and dispatches each received opcode to a do_* handler function.



8.1.1 AT client responsibilities

The AT client implements the custom command callbacks, validates execution, read, and test forms, sends operation messages to the server, converts returned status values into AT responses, and forwards server notifications as #VOIPURC indications. It also configures the DVI codec and audio channels based on the request received with AT#VOIPDVI.

8.1.2 VoIP server responsibilities

The server creates the message queues, runs a command-processing thread, maps request opcodes to server functions, and owns the PJSUA objects and callbacks. In the current implementation, **pjproject** release 2.17 is used.

It also emits internal SRV_UP and SRV_DOWN notifications during server start and shutdown.

8.1.3 Client/server communication overview

Communication uses a common packed message_t structure containing an opcode, a result code, and an operation-specific union payload. A request is sent on the client-to-server queue. The server changes the opcode to the corresponding reply opcode by adding the reply offset and sends the response on the server-to-client queue.

Asynchronous indications use a dedicated server-to-client URC queue.

Queue	Direction	Purpose
/pjsua_client_to_server	AT-client to server	Command requests
/pjsua_server_to_client	Server to AT-client	Synchronous command replies
/pjsua_server_urc_to_client	Server to AT-client	Asynchronous URCs

Note: The message queue protocol is an internal implementation detail. Applications are supposed to use the AT commands documented below rather than access the queues directly.

8.2 Applications execution and logging

If the client and server apps are manually built, their binary files are named **at_client** and **server**, and can be placed in any writable partition of the modem filesystem (e.g. /data, or /customapps for double memory devices). When built with default options, the output logs are available on **stdout**.

They can be launched manually as any binary file:

```
chmod +x server at_client
./server
On a different shell (as both server and client lock the interface
while running):
./at_client
```

When they are built using Yocto (e.g. through the OpenLinux SDK) with the recipes provided, the generated firmware image will have them built so that:

- the binary files are placed in **/data/voip**
- **systemd** services to start them are available:

```
systemd start voip-server voip-atclient
```

- the logs are moved to syslog, thus making them available with journalctl:

```
journalctl -f -t voip-atclient -t voip-server | grep -viE  
"ipc|p_rr|rm_dispatcher"
```

Note: | **grep -viE "ipc|p_rr|rm_dispatcher"** allows to filter out internal calls to modem APIs that are present in the atclient output.

9 Acronyms and Abbreviations

Table 2: Acronyms and Abbreviations

Acronym	Definition
AT	Attention command
CME	Mobile equipment error
DTMF	Dual-tone multi-frequency
IPC	Inter-process communication
MQ	Message queue
PJSIP	Open-source SIP communications library
PJSUA	High-level PJSIP user-agent API
RPID	Rich Presence Information Data
RTP	Real-time Transport Protocol
SIP	Session Initiation Protocol
URC	Unsolicited Result Code
VoIP	Voice over Internet Protocol

10 Related Documents

For current documentation and downloads, refer to <https://dz.telit.com/>.

Table 3: Related Documents

S.no	Doc Code	Document Title
1	80586ST11192A	LE910Cx Linux AT Commands Reference Guide

11 Document History

Table 4: Document History

Revision	Date	Changes
0	2026-09-11	Initial release

© Telit Cinterion. All rights reserved.

