Grandstream Networks, Inc.

GXV3175 IP Multimedia Phone

GMI HTML SDK and API Guide

Table of Contents

GMI API Guide V2.0

1. GMI API V2.0 Overview

GMI (Grandstream Manager Interface) is a management API developed by Grandstream Networks. Designed for our IP Multimedia phones, it allows partners to develop customized applications on the phone.

GMI supports standard HTML/CSS/Javascript, users can use these dynamic web page development languages to develop their customized application. GMI will display the application on the phone based on the web pages generated.

Additionally, GMI provides several basic API functions (packaged in Javascript) to facilitate users calling the existing applications on the multimedia phone or to obtain the phone status etc. This allows users with basic web application programming skills to develop their customized applications on the multimedia phone, without the effort of learning a new programming language. All that is required for the user is to understand how the GMI works and how to use these simple APIs to interact with the phone.

GXV3175 supports GMI Version 2.0. GMI V2.0 has two types of JavaScript interface:

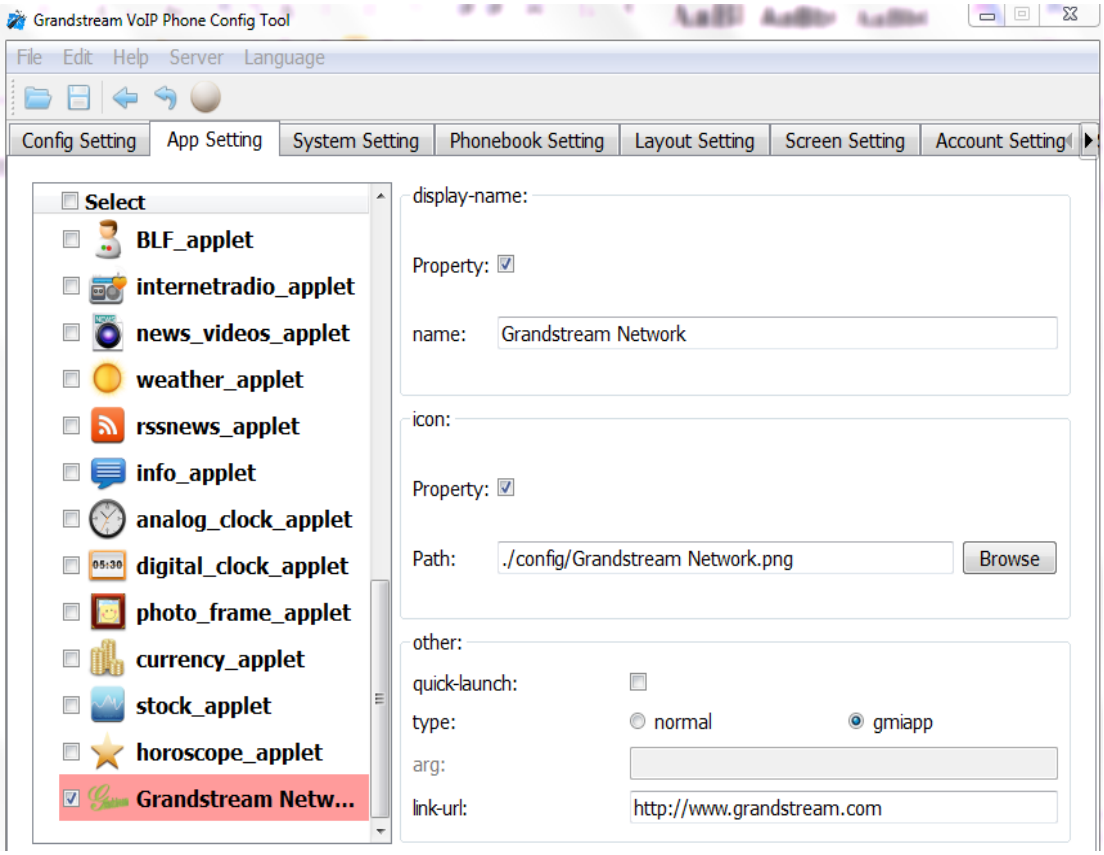
- simpleGMI: Provides a few simple and fast interfaces. It is highly efficient but has no return value. Users could not obtain execution result nor interface for specific data.
- GMIEngine: Provides return value(s) and enables users to obtain interface for specific data and environment variables.

All functions in SimpleGMI interface are also included in GMIEngine interface. We recommend you use the GMIEngine interface as GMIEngine will be updated in newer software versions while SimpleGMI will not.

The number of interfaces is growing with increasing customer demand. Some interfaces in this guide are introduced after certain GMI version. To avoid compatibility issue, before using these interfaces, please check GMIEngine.version to determine the GMI version needed for that interface. For interfaces that do not have GMI version requirements indicate that they exist in all GMI versions and there is no need to identify GMI version.

2. How to Add a GMI Application

To add a GMI application to the multimedia phone, users need to use the GUI Customization Tool (GXV3175 GUI Customization Tool). In the figure below, for example, under “App Setting”, check “Grandstream Networks” option on the left panel, users can edit the following fields on the right side:



display-name	GMI application's name to be displayed on the phone's LCD
Icon/Path	GMI application's icon to be displayed on the phone's LCD
quick-launch	Add the GMI application in the LCD workbench menu
type	Select "gmiapp" for GMI application
link-URL	HTTP URL for the GMI application or the location of a local GMI application, i.e. file://sda1/example/index.html

Users will then need to generate a custom file "gxv3170cust" and upload it to the phone via the firmware upgrade process. For more information regarding GXV3175 GUI Customization, please refer to the following document: GXV3175 GUI Customization Guide.

Local GMI application

Users may prefer local GMI application when the network speed is a concern for remote access. Please follow the procedures below to add a local GMI application to GXV3175.

- Download the following GMI example package from Grandstream web site: Grandstream GMI Example 1

- Unzip and copy it to GXV3175 with a storage device. The main file *index.html* will be located under */sda1/Hotel* directory. (Depending on the storage device type, the folder name in the File Manager could be sda1, udisk, sda2 etc).
- In GXV3175 GUI Customization Tool, under "App Setting", add a GMI application as explained in the above section. Enter the following content for the link-URL:

  file://sda1/Hotel/index.html
- Generate a "*gxv3170cust"* file with the GXV3175 GUI Customization Tool.
- Load the file "*gxv3170cust* "onto GXV3175 via firmware upgrade process.
- Reboot the GXV3175, a local GMI application will be added to GXV3175.

3. simpleGMI Interface

All simpleGMI functions are also included in GMIEngine. For example, the simpleGMI.refresh( ) function can also be called using GMIEngine.refresh( ). Duplicate functions will not be explained twice.

3.1 simpleGMI.refresh()

Function	simpleGMI.refresh()
Description	Refresh the current page. The phone will obtain the current page from the web server and reload the page on the phone.
Parameters and Return value	**Parameter:** N/A **Return value:** N/A
Note	This interface is mainly for debugging. It is recommended to remove this function in your program before official release so that the program will run as smoothly as the built-in applications on the phone, providing users with better user experience.

3.2 simpleGMI.historypage()

Function	simpleGMI.historypage(num)
Description	Go to the page visited in history, as specified in parameter num.

Parameters and Return value	**Parameter:** num - Any natural number. • Negative number represents the number of page records to go backward; • 0 represents the current page; • Positive number represents the number of page records to go forward. For example, the user visited these web pages in the following order: www.google.com www.yahoo.com www.baidu.com When the user is browsing the webpage www.baidu.com, the function call to: simpleGMI.historypage(-1) will allow the user to return to the last page visited (www.yahoo.com). At this point, the last page visited will be www.google.com. If the function: simpleGMI.historypage(-1) is called again, the user will return to www.google.com and the next page visited is www.baidu.com. If the function: simpleGMI.historypage(1) is called, the user will return to www.yahoo.com. If num is a non-existent record, nothing will take effect. **Return Value:** N/A
Note	

3.3 simpleGMI.gotoURL()

Function	simpleGMI.gotoURL(url)
Description	Go to the URL address specified in the url parameter.
Parameters and Return value	**Parameter:** url - any url

	Return Value: N/A
Note	This function is used to implement a hyperlink.

3.4 simpleGMI.dial()

Function	simpleGMI.dial(acct, isVideo, isDialPlan, number, headers, hideCall)
Description	Call a specified number from a specified account.
Parameters and Return value	**Parameter:** acct - the account to be used starting from 0 isVideo - video call or not. 1-Yes, 0-No isDialPlan - check dial plan or not. 1-Yes, 0-No number - number to be dialed headers - add in SIP header. For instance, when "headers" is: myheader1=myvalue1&myheader2=myvalue2 When Call is initiated, the following two headers will be added in INVITE: myheader1: myvalue1 myheader2: myvalue2 hideCall – Whether to hide the phone interface or not. 1- Yes, 0-No. This augment is only effective for audio calls. Phone interfaces can not be hidden for video calls. **Return Value:** N/A
Note	N/A

3.5 simpleGMI. changeVolume ()

Function	simpleGMI.changeVolume(isAdd)
Description	Adjust volume during a call
Parameter and Return Value	**Parameter:**

	isAdd – Whether to increase volume. 1-Yes, 0-No **Return Value:** N/A
Note	N/A

3.6 simpleGMI. keyPadInput ()

Function	simpleGMI..keyPadInput (digit, isDown, dialedStr)
Description	Send DTMF during a call
Parameter and Return Value	**Parameter:** digit - DTMF number (0 to 9) isDown – the button is pressed or released. 1 – Pressed, 0 – Released. One pressing button event consists of one pressing operation and one releasing operation. dialedStr - Reserved **Return Value:** N/A
Note	N/A

3.7 simpleGMI.transfer()

Function	simpleGMI.transfer()
Description	Go to Transfer status.
Parameter and Return Value	**Parameter:** N/A **Return Value:** N/A
Note	Valid only when the current line is connected in a call.

3.8 simpleGMI.transferTo()

Function	simpleGMI.transferTo(destnum)

Description	In transfer status, transfer the current call to destination number.
Parameter and Return Value	**Parameter:** destnum - the destination number to be transferred to **Return Value:** N/A
Note	Valid only when the current call is in transfer status. If users would like to use one-key-transfer, users could enter the number first then call simpleGMI.transfer() and simpleGMI.simpleGMI(destnum) in the return function to dial out.

3.9 simpleGMI.hangup()

Function	SimpleGMI.hangup()
Description	Hang up the current call.
Parameter and Return Value	**Parameter:** N/A **Return Value:** N/A
Note	N/A

3.10 simpleGMI.launchService()

Function	simpleGMI.launchService(program)
Description	Launch the specified program.
Parameter and Return Value	**Parameter:** program - specify the program to be launched The program (displayed as the menu item name on the phone) to be launched and the corresponding code which could be sent to launch interface as parameter are as follows: Contacts - “Phonebook” Call History - “CallHistory” Message - “Message”

	Browser - “Browser”
	AlarmColck - “AlarmClock”
	BunnyHunt - “BunnyHunt”
	Picture Matching - “Buzzy”
	Calculator - “Calculator”
	Calendar - “Calendar”
	Color Code - “colorcode”
	Facebook - “Facebook”
	File Manager - “FileManager”
	Flickr - “Flickr”
	Gobang - “Gobang”
	Google Voice - “GoogleVoice”
	Tetris - “gottet”
	Internet Radio - “InternetRadio”
	IP to Location - “IP2Location”
	Last.FM - “LastFM”
	Media Player - “MediaPlayer”
	Movie Trailer - “MovieTrailer”
	News Videos - “NewsVideos”
	Peg - “peg”
	Phanfare - “Phanfare”
	Photobucket - “Photobucket”
	Camara - “PIP”
	Checker - “Qchecker”
	Soduku - “simsu”
	Slide Show - “SlideShowApp”
	Solitare - “Solitaire”
	System Info - “SystemInfo”
	System Setting - “SystemSetting”
	Tudou - “Tudou”
	Twitter - “Twitter”
	Youtube - “Youtube”

	Return Value: N/A
Note	LaunchService program must exist in the menu which has already removed the unused items. Otherwise, it could not be launched. For instance, if Youtube has already been removed from *menu.xml*, launchService("Youtube") could not be launched.

3.11 simpleGMI.closeService()

Function	simpleGMI. closeService (program)
Description	Close the specified program.
Parameter and Return Value	**Parameter:** program - specify the program to be closed Please refer to simpleGMI.launchService(program) for a complete list of program names. **Return Value:** N/A
Note	N/A

3.12 simpleGMI.play()

Function	simpleGMI.play(path, mode, barShow, cb_play)
Description	Play the audio/video as specified in path
Parameter and Return Value	**Parameter:** path - specify the relative path of the audio/video to be played (relative to the file that calls this function). Use ";" to separate multiple files mode - play mode. 0 - Play once; 1 – Repeat barShow – Whether to display the controller when playing a video (output device, play, pause, end and change volume). Valid only if a video is played. cb_play – function cb_play (data), the call back function. It is called once per second. Argument data is the playing information formatted as {time:7, filename:recording.avi,

	state:2}. Argument time is the amount of time that the video has already been played for (in second). Argument filename is the name of the file being played. Argument state is status. Status code: 0=IDLE, 1=STARTING, 2=PLAYING, 3=PAUSED, 4=STOPPED, 5=END, 6=STOPPING **Return Value:** N/A
Note	For continuous play, please set playback interval to more than 5 seconds.

3.13 simpleGMI.stopPlay()

Function	simpleGMI.stopPlay()
Description	Stop playing
Parameter and Return Value	**Parameter:** N/A **Return Value:** N/A
Note	N/A

3.14 simpleGMI.udp()

Function	simpleGMI.udp(host, port, data, cb_udp)
Description	Send data to specific port of specific host
Parameter and Return Value	**Parameter:** host - destination host or IP address port - destination udp port data - send data contents (format: string) cb_udp – the call back function after udp data sent, prototype of the call back function: function cb_udp(data) Argument data is the response of the destination host. **Return Value:**

	N/A
Note	N/A

3.15 simpleGMI.post()

Function	simpleGMI. post(url, data, cb_ post)
Description	Send http “POST” request to specific host
Parameter and Return Value	**Parameter:** url - Destination url data - Parameters sent with POST cb_post - The return function for the response to the POST request For example: function cb_post(data) The data parameter is for the response to the POST request. (If the data in the response is in xml format, GMI interface will transform it to JSON format. The data in other format in the response will remain the same). **Return Value:** N/A
Note	N/A

3.16 simpleGMI.exit()

Function	simpleGMI.exit()
Description	Exit from GMI. The application programs from GMI will be ended and the resource will be released.
Parameter and Return Value	**Parameter:** N/A **Return Value:** N/A
Note	N/A

3.17 simpleGMI.fullScreen()

Function	simpleGMI.fullScreen()
Description	Display in full screen. The control bar on the right hand side will not be displayed.
Parameter and Return Value	**Parameter:** N/A **Return Value:** N/A
Note	N/A

3.18 simpleGMI.normalScreen()

Function	simpleGMI.normalScreen()
Description	Normal display mode. The control bar on the right hand side will be displayed.
Parameter and Return Value	**Parameter:** N/A **Retuen Value:** N/A
Note	It is the default screen display mode.

3.19 simpleGMI.dialWithDomain()

Function	simpleGMI.dial(domain, isVideo, isDialPlan, number, headers)
Description	Call a number on a specific SIP server
Parameter and Return Value	**Parameter:** Domain – URI of the SIP server called isVideo – is a video call or not. 1 – Yes; 0 – No isDialPlan – whether to check the dial plan. 1 – Yes; 0 – No number – the number specified to be called headers – a list of SIP headers to be added. For example, if 'headers' is "myheader1=myvalue1&myheader2=myvalue2", two headers will be added in the INVITE message when a call is initiated:

	myheader1: myvalue1 myheader2: myvalue2 **Return Value:** N/A
Note	The phone will use an account registered on the same SIP server to call out. If no such account exists, a call cannot be initiated.

3.20 simpleGMI.setAudioVolume(vol)

Function	simpleGMI.setAudioVolume (vol)
Description	Set the audio or video volume
Parameter and Return Value	**Parameter:** vol – the value of the volume to set. Integer between 0 and 100. **Return Value:** N/A
Note	N/A

4 GMIEngine Interface

Note: All functions in SimpleGMI interface are also included in GMIEngine interface. The following interfaces are those solely supported by GMIEngine.

4.1 GMIEngine.getNetWorkInfo()

Function	GMIEngine.getNetWorkInfo()
Description	Obtain network information
Parameter and Return Value	**Parameter:** N/A **Return Value:** Return strings in JSON format. It has to be transformed to JSON format first. For instance: {“mac”: “00-0b-82-27-ea-ed”, “ip”: “192.168.1.118”, “mask”: “255.255.255.0”, “type”: “DHCP”, “gateway”: “192.168.1.1”, “dns”: “192.168.1.253”, “nat”: “Port Restricted Cone NAT (STUN)”} Strings in red will be displayed as the actual value in the phone.
Note	N/A

4.2 GMIEngine.getCurrentLanguage()

Function	GMIEngine.getCurrentLanguage()
Description	Obtain the current language
Parameter and Return Value	**Parameter:** N/A **Return value:** Return string in JSON format. It has to be transformed to JSON format first. For instance: {“lan”: “en”}

	Strings in red will be displayed as the actual value in the phone.
Note	N/A

4.3 GMIEngine.getCountry()

Function	GMIEngine.getCountry()
Description	Obtain the country code
Parameter and Return Value	**Parameter:** N/A **Return Value:** Return string in JSON format. It has to be transformed to JSON format first. For instance: {“country”: “CN”} Strings in red will be displayed as the actual value in the phone.
Note	N/A

4.4 GMIEngine.getAccountInfo()

Function	GMIEngine. getAccountInfo()
Description	Obtain the account info
Parameter and Return Value	**Parameter:** N/A **Return Value:** Return string in JSON format. It has to be transformed to JSON format first. For instance: {“info”: [{“index”: “0”, “enable”: “1”, “registerFlag”: “1”, “acctName”: “IPVideoTalk”, “server”: “sip.ipvideotalk.com:48879”, “userID”: “8109060”, “authID”: “8109060”, “callIDName”: “8109060”}, {“index”: “1”, “enable”: “1”, “registerFlag”: “1”, “acctName”: “8089”, “server”: “192.168.1.20”, “userID”: “8089”, “authID”: “8089”, “callIDName”: “8089”}, {“index”: “2”, “enable”: “0”, “registerFlag”: “0”, “acctName”: “8011”, “server”:

	"192.168.1.20", "userID": "8011", "authID": "8011", "callIDName": "8011"}] } Strings in red will be displayed as the actual value in the phone.
Note	N/A

4.5 GMIEngine.put()

Function	GMIEngine. put(family, valuelist)
Description	Store the data specified by the parameter
Parameter and Return Value	**Parameter:** family – The family where the data belongs to valuelist – The data list to be set. The values should be separated by "&". **Return Value:** N/A Example: The variables and function are as below: var ip = "192.168.1.220"; var name = "admin"; var password = "admin"; GMIEngine.put("HoneyWell", "ip=" + ip + "&name=" + name + "&password=" + password); Then ip/name/password will be stored and set under "HoneyWell" family.
Note	This interface can be used with get() so the application data can be read and written in GMI.

4.6 GMIEngine.get()

Function	GMIEngine.get(family, keylist)
Description	Get the data specified by the parameter
Parameter and Return Value	**Parameter:** family - The family where the data belongs to keylist – The data list to be retrieved. The values should be

	separated by “&”. **Return Value:** Return data in JSON format. The format should be transformed into JSON format first. Example: var valueList = GMIEngine.get(“HoneyWell”, “name&ip&password”); If the variables have been set as below: var ip = “192.168.1.220”; var name = “admin”; var password = “admin”; GMIEngine.put(“HoneyWell”, “ip=” + ip + “&name=” + name + “&password=” + password); Then the following expression var valueList = GMIEngine.get(“HoneyWell”, “name&ip&password”) will return {“name”: “admin”, “ip”: ”192.168.1.220”, “password” : “admin”}
Note	This interface can be used with put() so the application data can be read and written in GMI.

4.7 GMIEngine.getGroup()

Function	GMIEngine.getGroup (gpID)
Description	Obtain information for the group with group ID “gpID”. If “gpID” is empty, returns information for all phone book groups.
Parameter and Return Value	**Parameter:** gpID – group ID **Return Value:** {"res": "success", "msg": [{“groupID":"0","groupName":"default"}, {“groupID":"100","groupName":"blacklist"}]}
Note	N/A

4.8 GMIEngine.getContact()

Function	GMIEngine.getContact (ctID, gpID, ctName)
Description	Obtain contact information from phone book 1、If ctID is not empty but ctName is empty, return contact information for ctID; 2、If cfName is not empty, get all contacts information whose name contains cfName; 3、If both ctID and ctName are empty but gpID is not, get all contacts information within group gpID; 4、If ctID, ctName and gpID are all empty, get all contacts information.
Parameter and Return Value	**Parameter:** ctID – contact ID gpID – group ID ctName – contact name **Return Value:** 1.Return Successful： "{"res": "success", "msg": [{“contactID":"0","contactName":"xxx","groupID":"0","work":"8819","home":"", "mobile":"","fax":"","photo":"","email":""}, {“contactID":"1","contactName":"grandstream","groupID":"0","work":"3587","home":"“, "mobile":"","fax":"","photo":"","email":""}]}" 2.Failed to get information： "{"res": "error", "msg" : "can't get contacts information"}" 3.Contact is empty： "{"res": "success", "msg" : "0"}"
Note	N/A

4.9 GMIEngine.getGroupCount()

Function	GMIEngine.getGroupCount( )

Description	Obtain the number of groups in the phonebook
Parameter and Return Value	**Parameter:** N/A **Return Value:** Return a json formatted string. The format should be transformed into JSON format first. For example: Var valueList = GMIEngine.getGroupCount (); If corresponding data has been stored, the function will return {“phoneGroupAccount”: “3”}
Note	N/A

4.10 GMIEngine.getContactCount()

Function	GMIEngine.getContactCount()
Description	Obtain the number of all contacts in the phonebook
Parameter and Return Value	**Parameter:** N/A **Return Value:** Return a json formatted string. The format should be transformed into JSON format first. For example: Var valueList = GMIEngine.getContactCount(); If corresponding data has been stored, return {“phoneContactAccount”: “9”}
Note	N/A

4.11 GMIEngine.setContact()

Function	GMIEngine.setContact (phbkContact)
Description	（1）Set up phonebook contacts information (including edit and add contacts information) （2）Character array of contact information，the length of array

	is 8，they are -- ctID，ctName，gpID，work，home，mobile，fax，mail Example：var a = new Array("","testName","0","111","222","333","444","mail@test.com"); GMIEngine.setContact (a) This will add a contact with name “testName”，work phone number 111in the system default group。 ctID – contact ID ctName – contact Name gpID – group ID work – work phone number home – home phone number mobile – mobile phone number fax – fax number mail – mailing address （3）If ctID is empty, add contacts. Otherwise, edit contacts.
Parameter and Return Value	**Parameter:** PhbkContact – Array of contact information **Return Value:** －1 – edit failed 1 or other – ctID of the added contact
Note	N/A

4.12 GMIEngine.setGroup()

Function	GMIEngine. setGroup (gpName, gpID)
Description	（1）Set up phonebook group information（including edit and add group information） （2） If gpID is empty，add contact group；otherwise edit contact group based on gpID
Parameter and Return Value	**Parameter:** gpID – group ID，string gpName – group name, string **Return Value:** 0 – edit successful

	−1 – edit failed 1 or other – gpID of the added contact group
Note	N/A

4.13 GMIEngine.moveToDefault()

Function	GMIEngine. moveToDefault (ctID)
Description	Move a contact to system default group based on ctID.
Parameter and Return Value	**Parameter:** ctID – Contact ID **Return Value:** 1.Move Successful： true 2.Move Failed： false
Note	N/A

4.14 GMIEngine.removeContact()

Function	GMIEngine. removeContact (ctID)
Description	Delete contact based on ctID. If ctID is empty, delete all contacts.
Parameter and Return Value	**Parameter:** ctID – Contact ID **Return Value:** 1.Delete Successful： true 2.Delete Failed： false
Note	N/A

4.15 GMIEngine.clearGroup()

Function	GMIEngine. clearGroup (gpID)
Description	Clear contacts in group based on group ID. All cleared contacts will be put in system default group.
Parameter and Return Value	**Parameter:** gpID – group ID **Return Value:** 1.Clear Successful: true 2.Clear Failed: false
Note	N/A

4.16 GMIEngine.removeGroup()

Function	GMIEngine. removeGroup (gpID)
Description	Delete a phonebook group based on its group ID. If gpID is empty, delete all groups.
Parameter and Return Value	**Parameter:** gpID – group ID **Return Value:** 1.Delete Successful: true 2.Delete Failed: false
Note	System default group and black list group cannot be deleted. Their IDs are 0 and 100, respectively.

4.17 GMIEngine.getAudioVolume()

Function	GMIEngine.getAudioVolume( )
Description	Obtain audio and video volume
Parameter and Return Value	**Parameter:** N/A **Return Value:** An integer between 0 and 100 for volume. 0 means mute.
Note	N/A

4.18 GMIEngine.getLineStatus()

Function	GMIEngine. getLineStatus (line)
Description	Get phone line status（Version GMI8 and up）
Parameter and Return Value	**Parameter:** line – line index，value 0－2, corresponds to line 1－3 **Return Value:** Line status value，an integer between -1 and 13. Each value represents the following： -1， Failed to get line status CALL_IDLE = 0, Idle, Can make calls CALL_DIALING=1, Dialing CALL_RINGING=2, Ringing CALL_CALLING=3, Calling CALL_CONNECTED=4, Call is connected CALL_ONHOLD=5, Call is on hold CALL_TRANSFERED=6, Call is transferred CALL_ENDING=7, Call is ending CALL_FAILED=8, Call failed CALL_TRANSFER=9, Call is being transferred CALL_CONFERENCE=10, In conference CALL_PAGING=11, Paging CALL_RINGBACK=12, Ringback

	CALL_IPCALL＝13　Direct IP Call
Note	N/A

4.19 GMIEngine.openSoftKeyboard()

Function	GMIEngine.openSoftKeyboard(show)
Description	Display or hide the soft keyboard
Parameter and Return Value	**Parameter:** show: 1- Display the soft keyboard; 0 – Hide the soft keyboard **Return Value:** N/A
Note	N/A

5. GMIEngine Environment Variables

GMI provides global variables for users to obtain the device information from them. The variables are

listed below with their descriptions.

Variable Name	Description
GMIEngine.version	GMIEngine version, currently 6. Incremented by 1 for a new version
GMIEngine.ip	Phone's network IP address
GMIEngine.mask	Phone's subnet mask
GMIEngine.gateway	Phone's gateway IP
GMIEngine.dns	Phone's DNS IP
GMIEngine.mac	Phone's MAC address
GMIEngine.adressType	Three types to get the IP address: DHCP/static/PPPoE
GMIEngine.natType	NAT type
GMIEngine.accountActive	Account active or not. Array: GMIEngine.accountActive[0] GMIEngine.accountActive[1] GMIEngine.accountActive[2] are corresponding to the status of the three accounts
GMIEngine.accountName	Account name. Array: GMIEngine. accountName[0] GMIEngine. accountName[1] GMIEngine. accountName[2] are corresponding to the name of the three accounts
GMIEngine.accountServer	SIP server of the account. Array: GMIEngine. accountServer[0] GMIEngine. accountServer[1] GMIEngine. accountServer[2] are corresponding to the sip server of the three accounts
GMIEngine.accountUserID	UserID of the account Array: GMIEngine. accountUserID[0]

	GMIEngine. accountUserID[1] GMIEngine. accountUserID[2] are corresponding to the userID of the three accounts
GMIEngine.accountAuthID	AuthID of the account Array: GMIEngine. accountAuthID[0] GMIEngine. accountAuthID[1] GMIEngine. accountAuthID[2] are corresponding to the AuthID of the three accounts
GMIEngine.accountCallerID	CallerID of the account Array: GMIEngine. accountCallerID[0] GMIEngine. accountCallerID[1] GMIEngine. accountCallerID[2] are corresponding to CallerID of the three accounts

Users could use these variables directly in Javascript, for instance:

```
var account1Uri = GMIEngine.accountUserID[0] + "@" + GMIEngine.accountServer[0];
var account2Uri = GMIEngine.accountUserID[1] + "@" + GMIEngine.accountServer[1];
var account3Uri = GMIEngine.accountUserID[2] + "@" + GMIEngine.accountServer[2];
```