



Grandstream Networks, Inc.

GXV3140 IP Multimedia Phone

GMI HTML SDK and API Guide

1 What is GMI?

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

GMI supports standard HTML/CSS/Javascript code, so that 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 pages generated.

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

2 GMI Working Model

The GMI working model is a cross-domain model; it must interact with servers in two domains to implement the functions needed.

- **Interact with the user's web server:** Send page request, obtain page content and remodel it for display as an application on the multimedia phone.
- **Interact with the GMI server on the phone:** Control the phone behavior (e.g. initiate a call, launch the phonebook application etc...) by calling the Javascript packaged GMI interface.

Therefore, by interacting with the user's web server, it obtains the user-defined display content, style and data for the customized application and redraw it for display on the phone (at this point, it is similar to a browser). When any action from the phone is needed (e.g. making a call, launching some applications on the phone etc...), it is then required to control the phone behavior through calling the GMI API functions.

Refer to the GMI Working Model below:

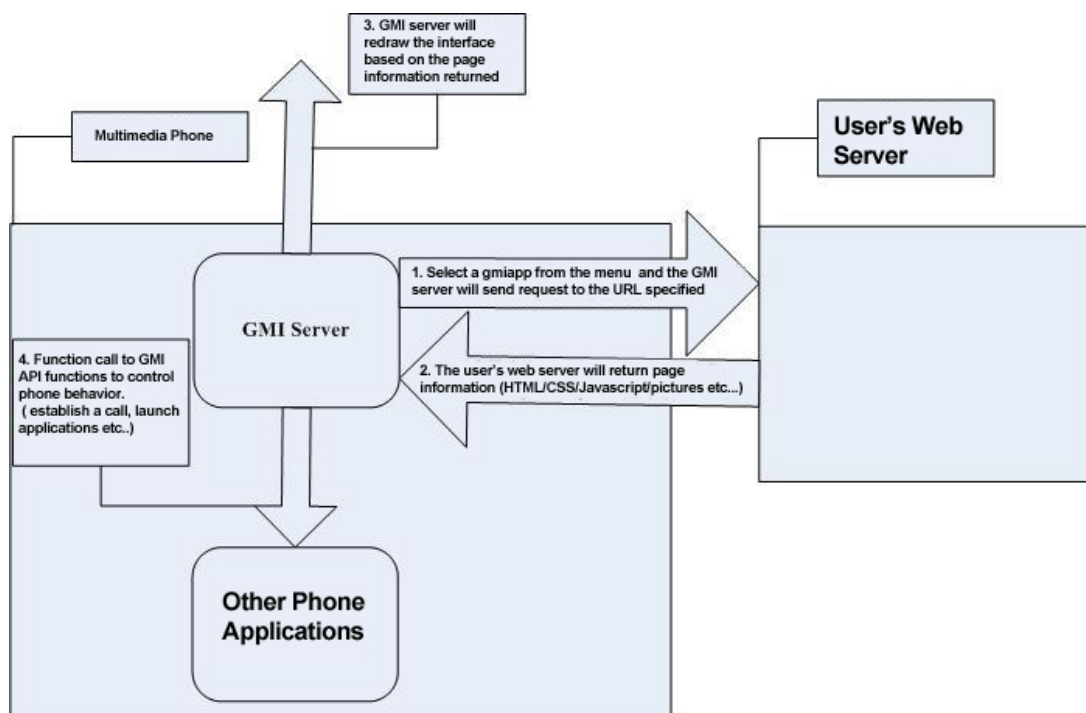


Figure 1: GMI Working Model

3 How do I add a GMI application?

To use the GMI application, users need to add the following in menu.xml:

```
<link icon="icon/Grandstream.png" dynamic-icon="icon/Grandstream.gif" display-name="GMI" type="gmiapp">
  <link-url>
  www.grandstream.com/support/gxv\_series\_phone/gxv3140/resources/gxv3140\_gmi\_example
  </link-url>
</link>
```

This line of code will add a menu selection in your phone menu called “GMI Example”. The parameter “Icon” specifies the icon displayed and the parameter “dynamic-icon” specifies the icon displayed when it is selected. The parameter func-name="gmiapp" specifies that this is a GMI application, and the parameter “url” specifies the URL address at which this application is stored. In this example, the phone will point to www.grandstream.com/support/gxv_series_phone/gxv3140/resources/gxv3140_gmi_example to get the corresponding application to display on the phone. The parameter “display-name” specifies the text displayed in the phone menu and the parameter “ver” specifies the menutree version. Users can modify these four parameters - icon/dynamic-icon/url/display-name to suit their needs.

For more details regarding menu.xml, please refer to the following document: [《GXV31XX XML Based GUI Customization Guide》](#).

The example files are available for [download here](#).

4 GMI API Description

In order to facilitate the ease of controlling the phone behavior within the html code, we have developed and provided several basic API functions (packaged in Javascript). This chapter provides introduction to all the API functions available.

4.1 GMIEngine.refresh()

Function:	GMIEngine.refresh()
Purpose:	Used to refresh the current page. The phone will obtain the current page from the web server and reload the page on the phone.
Parameter and Return Value:	Parameter: None Return Value: None
Additional Description:	This API function is mainly used for testing during development. It is advised to remove this function call 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.

4.2 GMIEngine.historypage()

Function:	GMIEngine.historypage(num)
Purpose:	Go to the pages visited in history, as specified by num.
Parameter and Return Value:	Parameter: num - Any natural number. Negative numbers represents the number of page records to go backward. 0 represents the current page. Positive numbers represents the number of page records to

	<p>go forward.</p> <p>For example, the user visited these web pages in this order: www.google.com, www.yahoo.com and www.baidu.com. When the user is browsing the webpage www.baidu.com, the function call to GMIEngine.historypage(-1) will allow the user to return to the last page visited (returning to www.yahoo.com). At this point, the last page visited will be www.google.com, if the function call GMIEngine.historypage(-1) is called again, the user will return to www.google.com. If the num specified is a non-existent record, nothing will take effect.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	This API function can be used to implement the “back” function for the softkey, allowing users to re-visit the upper level menus.

4.3 GMIEngine.gotourl()

Function:	GMIEngine.gotourl(url)
Purpose:	Go to the URL address specified in the url parameter.
Parameter and Return Value:	<p>Parameter:</p> <p>url – Can be any URL address.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	This API function is used to implement a hyperlink.

4.4 GMIEngine.call()

Function:	GMIEngine.call(acct, number, cb_call)
Purpose:	Used to call a specified number from a specified account.
Parameter and Return Value:	Parameter:

Value:	<p>acct – Specifies the account used to dial out. (Starting from 0).</p> <p>number – Specifies the number to call to.</p> <p>cb_call - Callback function after the call request is sent. The function prototype is function cb_call(result), the parameter “result” is the response to the call request</p> <p>result is a text string in json format. For example, {"Response": "Success", "Message": "Call Originate" } or {"Response": "ERROR", "Message": "User can't be null"};</p> <p>“Response” returns the result. There are two possibilities- “Success” or “ERROR”, “Message” returns the specific error message.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.5 GMIEngine.getURI()

Function:	GMIEngine. getURI (acct, cb_get)
Purpose:	Used to obtain the URI of the specified account
Parameter and Return Value:	<p>Parameter:</p> <p>acct – Specifies the account to be used. (Starting from 0)</p> <p>cb_get—Callback function after the getURI request is sent. The function prototype is function cb_get(result), the parameter “result” is the response to the function call.</p> <p>result is a text string in json format. For example,</p> <p>{"Response": "Success", "Message": "8100000@sip.ipvideotalk.com:48879"} or {"Response": "ERROR", "Message": "Acct can't be null"};</p> <p>“Response” returns the result. There are two possibilities- “Success” or “ERROR”, “Message” returns the URI. When the specified account is configured, the URI returned will contain “@”.</p> <p>Return Value:</p> <p>None</p>

Additional Description:	None
--------------------------------	------

4.6 GMEngine. hangup()

Function:	GMEngine. hangup(cb_hangup)
Purpose:	Used to hang up the current call
Parameter and Return Value:	<p>Parameter:</p> <p>cb_hangup— Callback function after the hangup request is sent. The function prototype is function cb_hangup (result), "result" is the response to the hangup function call.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.7 GMEngine. getstate()

Function:	GMEngine. getstate (acct, cb_get)
Purpose:	Used to obtain the state of the specified account
Parameter and Return Value:	<p>Parameter:</p> <p>acct—Specifies the account (starting from 0)</p> <p>cb_get—Callback function after the gestate request is sent. The function prototype is function cb_get (result), "result" is the response to the getstate function call.</p> <p>result is a text string in json format. For example, if the function call is:</p> <pre>GMEngine.getstate(0, cb_get)</pre> <p>If account 1 is currently in a call on line 1 with 8100100, it will return the following:</p> <pre>{"Response":"Success","registered":"1","line":"1","line-0" : "8100100"};</pre> <p>This specifies that account 1 is registered and using 1 line, it is currently in a call with 8100100 on line 1.</p> <p>If account 1 is idle, the response will be as follows:</p> <pre>{"Response":"Success","registered": "1", "line": "0"};</pre>

	<p>This specifies that account 1 is registered and is not using any line currently.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.8 GMIEngine.launch()

Function:	GMIEngine.launch(program, cb_launch)
Purpose:	Used to launch the program specified.
Parameter and Return Value:	<p>Parameter:</p> <p>program – Specifies the program to be launched using the program code.</p> <p>See the program (program name as displayed on the phone) and the corresponding program code below:</p> <p>Phone Book--"phonebook"</p> <p>Call History--"callhistory"</p> <p>Messages--"messages"</p> <p>Social Networks --"socialnetwork"</p> <p>IM--"IM"</p> <p>Twitter--"twitter"</p> <p>Facebook--"facebook"</p> <p>Google Voice--"googlevoice"</p> <p>Skype--"skype"</p> <p>Web Browser--"webbrowser"</p>

	<p>Multimedia--"multimedia"</p> <p>Gstris--"gstris"</p> <p>Photo Viewer--"gphoto"</p> <p>Photo Album--"webalbum"</p> <p>Media Player--"player"</p> <p>Internet Radio--"internetradio"</p> <p>Online Music--"onlinemusic"</p> <p>News Videos--"mulnews"</p> <p>Online Video--"onlinevideo"</p> <p>World Photos--"panoramio"</p> <p>Slide Show--"slide_show"</p> <p>Applications--"officetools"/"application"</p> <p>Calendar--"calendar"</p> <p>Alarm Clock--"alarmclock"</p> <p>Calculator--"calculator"</p> <p>Weather--"weatherupdate"</p> <p>Direct IP Call--"ipcall"</p> <p>File Manager--"filemanager"</p> <p>Stock--"stock"</p> <p>Currencies--"currencies"</p> <p>IP2Location--"ip2location"</p> <p>Today--"today"</p> <p>Settings--"settings"</p> <p>Display--"display"</p> <p>Time--"time"</p> <p>Accounts--"account"</p> <p>Network--"connection"</p> <p>Maintenance--"maintenance"</p> <p>Call Features--"callfeature"</p> <p>Camera--"camera"</p>
--	--

	<p>Video Setting--"video"</p> <p>FXO Setting--"fxo" (This is only applicable to phones to FXO support)</p> <p>Personalize--"personalize"</p> <p>Softkeys--"softkey_def"</p> <p>Screen Saver--"screen_display"</p> <p>RSS News--"rssnews"</p> <p>Screen Layout--"screen_layout"</p> <p>Language--"language"</p> <p>My Profile--"myprofile"</p> <p>Tones--"tones"</p> <p>Horoscopes--"horoscope"</p> <p>Info--"status"</p> <p>cb_launch— Callback function after the launch request is sent. The function prototype is function cb_launch(result), the parameter "result" is the response to the launch request</p> <p>result is a text string in json format. For example, {"Response": "Success", ", "Message": " Launch Originate" } or {"Response": "ERROR", ", "Message": " Program can't be null"};</p> <p>"Response" returns the result. There are two possibilities- "Success" or "ERROR", "Message" returns the specific error message.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.9 GMIEngine.message()

Function:	GMIEngine.message(acct, user, msg, cb_message)
------------------	--

Purpose:	Send message to a specified number from a specified account.
Parameter and Return Value:	<p>Parameter:</p> <p>acct- Specifies the account used to send the message. (Starting from 0).</p> <p>user- Specifies the number to send the message to.</p> <p>msg- Message Body</p> <p>cb_message - Callback function for the send message request. The function prototype is function cb_message(result), the parameter “result” is the response to the message request</p> <p>result is a text string in json format. For example, {"Response": "Success", "Message": "Message Send"} or {"Response": "ERROR", "Message": "MSG can't be null"}.</p> <p>“Response” returns the result. There are two possibilities- “Success” or “ERROR”, “Message” returns the specific error message.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.10 GMIEngine.exit()

Function:	GMIEngine.exit()
Purpose:	Exit the GMI. Calling this function will exit the customized application remodeled on the phone by the GMI and will release the resources.
Parameter and Return Value:	<p>Parameter:</p> <p>None</p> <p>Return Value:</p> <p>None</p>
Additional Description:	This API function can be used to implement the “Exit” softkey for the application implemented using GMI.

4.11 GMIEngine.backspace()

Function:	GMIEngine.backspace()
Purpose:	Simulates the backspace function for a keyboard.
Parameter and Return Value:	Parameter: None Return Value: None
Additional Description:	This API function is used to delete the characters in the input box.

4.12 GMIEngine.mapleftright2tab ()

Function:	GMIEngine.mapleftright2tab ()
Purpose:	Map the left/right button functions to tab. Pressing the Left button will be equivalent of pressing the alt+tab button; the previous control component will be highlighted. Pressing the Right button will be equivalent of pressing the tab button; the next control component will be highlighted.
Parameter and Return Value:	Parameter: None Return Value: None
Additional Description:	This API function is used to change the functions of the left/right button, so that the different control components can be selected using the left/right button.

4.13 GMIEngine.mapupdown2tab()

Function:	GMIEngine.mapupdown2tab()
------------------	---------------------------

Purpose:	Map the up/down button functions to tab. Pressing the Up button will be equivalent of pressing the alt+tab button; the previous control component will be highlighted. Pressing the Down button will be equivalent of pressing the tab button; the next control component will be highlighted.
Parameter and Return Value:	Parameter: None Return Value: None
Additional Description:	This API function is used to change the functions of the up/down button, so that the different control components can be selected using the up/down button.

4.14 GMIEngine.revokemap()

Function:	GMIEngine.revokemap()
Purpose:	Revoke the mapping between the up/down or left/right button with the tab button.
Parameter and Return Value:	Parameter: None Return Value: None
Additional Description:	None

4.15 GMIEngine. play ()

Purpose:	Play the designated audio/video file from the URL
Parameter and Return Value:	Parameter: url – Specifies the URL of the audio/video file. If there are several files, use “;” to separate them. mode – Play Mode. Currently there are two modes, 0: Play once; 1: Repeat. Callback function for the GMIEngine.play function. The function

	<p>prototype is function <code>cb_play(result)</code>, the parameter “result” is the response to the <code>GMIEngine.play</code> request.</p> <p>result is a text string in json format. For example, <code>{"Response": "Success", "Message": " Play Originate" }</code>; “Response” returns the result. There are two possibilities- “Success” or “ERROR”, “Message” returns the response message.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.16 GMIEngine. udp ()

Function:	<code>GMIEngine.udp (host, port, data, timeout, cb_udp)</code>
Purpose:	Send packets to the specified UDP port of the host/server
Parameter and Return Value:	<p>Parameter:</p> <p>host -- Destination server URL or IP address</p> <p>port -- Destination UDP port</p> <p>data -- Data to be sent (String)</p> <p>timeout – The timeout time for waiting for the reponse from the server. The unit is in seconds.</p> <p>cb_udp – Callback function for the UDP function. The function prototype is function <code>cb_udp(result)</code>, the parameter “result” is the response to the send UDP request.</p> <p>result is a text string in json format. For example, <code>{"Response": "Success", "Message": "Light Off"}</code>; “Response” returns the result. There are two possibilities- “Success” or “ERROR”, “Message” returns the response message returned by the destination server.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.17 GMIEngine.transfer()

Function:	<code>GMIEngine.transfer ()</code>
------------------	------------------------------------

Purpose:	<p>Activate transfer mode</p> <p>Note: Activate transfer mode. This will only work when current line is connected (You can use the function GMIEngine.getstate() to get the current state of the lines). Additionally, the transfer mode will also be activated when the “Transfer” button on the phone is pressed when the phone is running the GMI application.</p>
Parameter and Return Value:	<p>Parameter:</p> <p>None</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.18 GMIEngine.transfer2num()

Function:	GMIEngine.transfer2num (destnum)
Purpose:	<p>Transfer the call to the destination number (when transfer mode is activated)</p> <p>Note: Transfer current line to destnum. Need to call transfer() function to active transfer mode first. If the user wishes to implement a one-button transfer, they can implement a textbox to enter the number and call the two functions GMIEngine.transfer () and GMIEngine.transfer2num (destnum) consecutively when the user presses the button to submit.</p>
Parameter and Return Value:	<p>Parameter:</p> <p>destnum –The number to transfer to</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

4.19 GMIEngine. change_ime ()

Function:	GMIEngine. change_ime(id, cb_input_changed)
Purpose:	Change the current input method.
Parameter and Return Value:	<p>Parameter:</p> <p>id—id value for the display text type, used to display the current input method.</p> <p>cb_input_changed— Callback function for the change input method function. The function prototype is function cb_input_changed(result), the parameter “result” is the response to the change input method request.</p> <p>result is a text string in json format. For example, {"Response": "Success", "InputMethod": "abc" };</p> <p>“Response” returns the result. There are two possibilities- “Success” or “ERROR”, “InputMethod” returns the current input method.</p> <p>Return Value:</p> <p>None</p>
Additional Description:	None

5 GMI Applet Application

The GMI Applet is closely related to the GMI application, but there are some small differences.

The similarities between GMI Applet and GMI include:

- Both will model the application interface based on the obtained html page.
- Both supports cross-domain interaction (between the user-defined webserver and the built-in GMI server on the phone) and supports the same GMI API functions.

The differences are:

- GMI Applet is a desktop plugin application while GMI application will need to launch a separate application.
- Since GMI Applet is a desktop plugin application and the softkeys/navigation buttons are being used by the desktop application; therefore, the Backspace and the Shift keys are used to replace the F3 and F4 keys. When writing the GMI Applet Application source code, the Backspace key event should be mapped to the F4 key event and the Shift key event should be mapped to the F3 event.

Also, no modification in the menu.xml file is necessary for the GMI Applet. By default, the user can locate the GMI applet under the phone menu->Personalize->Screen Layout and configure the URL for the phone to obtain the page.

For more details and examples, please refer to the two source files in the applet folder (ajax.html and timer.html)

```
File: applet/ajax.html
function softkey_process(keyCode)
{
    document.title = "Applet-Ajax: in softkey_process, keyCode = " + keyCode;

    switch (keyCode)
    {
        case 8:
            //Process F3 event in Backspace keyevent
            //Backspace
            GMIEngine.refresh();
            break;

        case 16:
            //Process F4 event in Shift keyevent
            //Shift
            var orig = document.location;
            var dst = orig.href.replace(/ajax/i, "timer");
            GMIEngine.gotourl(dst);
            break;

        case 112:
            //F1 process here;
            break;

        case 13:
            //Enter process here
            break;

        case 113:
            //F2 process here;
            break;

        case 114:
            //F3 process here;
            break;

        case 115:
```

```

//F4 process here
break;

case 37:
//Left process here
break;

case 38:
//Up process here
case 40:
//Down process here
break;

case 39:
//Right process here
break;

default:
break;
}
}

```

6 GMI Demo Program

6.1 Menu Example

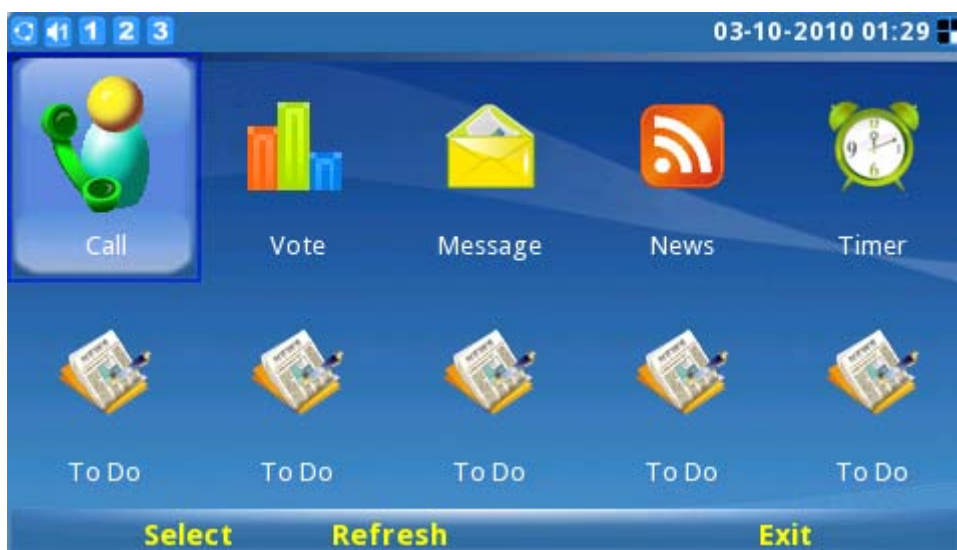


Figure 2: Menu Example illustration

Refer to the code and the comments below for the implementation of the Menu Example:

```

File: index.html

/*****

* After the page loads, the first item will be highlighted. At the same time, all the
keypad events are handled by softkey_process(keyCode).

o softkey_process has mapped F1/Enter button to "Select" to load the url of the
selected item. F2 is mapped to the API function call GMIEngine.refresh() to refresh
the current page, F4 is mapped to the API function call GMIEngine.exit() to exit the
application.

* The GMI App will process the up/down button keystroke and locate the item
selected.(Here, up/down/left/right button is not mapped to tab, the items are located
through an array )

*****/

<body onload="init();" onkeydown="softkey_process(event.keyCode);" id="body">
<table id="main">
  <tr width="100%">
    <td width="20%">

/*****

* Defines a menu item and change the button background when
mouseover/hover/focus.

*****/

    <button class="menu" id="b_call_0" onmouseover="chang_to_gif(document.
getElementById('i_call_1'));" onfocus="chang_to_gif(document.getElementById('i_
call_1'));" onmouseout="chang_to_png(document.getElementById('i_call_1'))" onb
lur="chang_to_png(document.getElementById('i_call_1'));">
      <table class="item_tab">
        <tr class="item_tr">
          <td align="center"></td>
        </tr>

        <tr><td align="center"><font class="item_label">Call</font></td></tr>
      </table>
    </button>
  </td>

  <td width="20%">
    <button class="menu" id="b_call_1" onmouseover="chang_to_gif(document.
getElementById('i_ticket_1'));" onfocus="chang_to_gif(document.getElementById('
i_ticket_1'));" onmouseout="chang_to_png(document.getElementById('i_ticket_1'))"

```

```

" onblur="chang_to_png(document.getElementById('i_ticket_1'))">
  <table class="item_tab">
    <tr class="item_fr">
      <td align="center"></td>
    </tr>

    <tr><td align="center"><font class="item_label">Vote</font></td></tr>
  </table>
</button>
</td>

<td width="20%">
  <button class="menu" id="b_call_2" onmouseover="chang_to_gif(document.
getElementById('i_ticket_2'));" onfocus="chang_to_gif(document.getElementById('
i_ticket_2'));" onmouseout="chang_to_png(document.getElementById('i_ticket_2'))"
  " onblur="chang_to_png(document.getElementById('i_ticket_2'))">
    <table class="item_tab">
      <tr class="item_fr">
        <td align="center"></td>
      </tr>

      <tr><td align="center"><font class="item_label">Message</font></td></tr>
    </table>
  </button>
</td>

..... //Omit the code for the other menu items

</tr>
</table>

/*****
* Draw/Simulate the softkey using HTML. Softkey should also support mouse click
events, therefore the API functions GMIEngine.refresh();GMIEngine.exit() are also
mapped in onclick.
*****/

<table id="softkey">
  <tr width="100%">
    <td width="40px">
    </td>

```



```
// After the page loads, call the API function GMIEngine.mapleftright2tab(); to map
the left/right button to tab/alt+tab

function bind_tab()
{
    GMIEngine.mapleftright2tab();
}

//When the page unloads, revoke the key mapping
function unbind_tab()
{
    GMIEngine.revokemap();
}

.....

/******
* After the page loads, call the API function GMIEngine.mapleftright2tab(); to map
the left/right button to tab/alt+tab. When the page unloads, revoke this mapping so
that it would not influence other pages.
* At the same time, all the key events are handled by the softkey_process function.
*****/

<body onload="bind_tab();" onunload="unbind_tab();" onkeydown="softkey_proce
ss(event.keyCode);" id="call-body">

/******
* Use an iframe to display the contact list, call button and procedure call button.
Users can make use of iframe to generate scroll windows.
*****/

<iframe src="call-iframe.html" id="iframe-call">
</iframe>

/******
* Draw softkey in html
*****/

<table id="softkey">
  <tr width="100%">
    <td width="40px">
    </td>
    <td width="100px" align="center">
      <a><font class="softkey">Backspace</font></a>

```

```

</td>

<td onclick="GMIEngine.refresh();" width="100px" align="center">
  <a><font class="softkey">Refresh</font></a>
</td>

<td width="100px" align="center">
  <a><font class="softkey">Call</font></a>
</td>

<td onclick="GMIEngine.historypage(-1);" width="100px" align="center" >
  <a><font class="softkey">Back</font></a>
</td>
<td width="40px">
</td>
</tr>
</table>
</body>

```

File: call-iframe.html

```

.....

function cb_call(result)           //Callback function for initiating call
{
  //Process call result here
  //alert(result["Response"]);
}

function call(user)
{
  GMIEngine.call(0, user, cb_call); //Initiate Call
}

function cb_launch(result)        //Callback function for launching application
{
  //Process launch result here
  //alert(result["Response"]);
}

function launch(program)
{
  GMIEngine.launch(program, cb_launch); //launch application
}

```



```

<h4>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Or Hear <font color='yellow' size="5">The
Music</font> they like</h4>
<div>
<input id="playsound" type="button" value="音 乐"
" onclick="mplay('sounds/2.mp3;sounds/hey oh.mp3;sounds/lige.wma;')"/>
//Multiple files can be separated by “;”
</div>

```

6.4 UDP Example

This example is a part of the “Call” example. Please scroll down to the end of the page to view this.

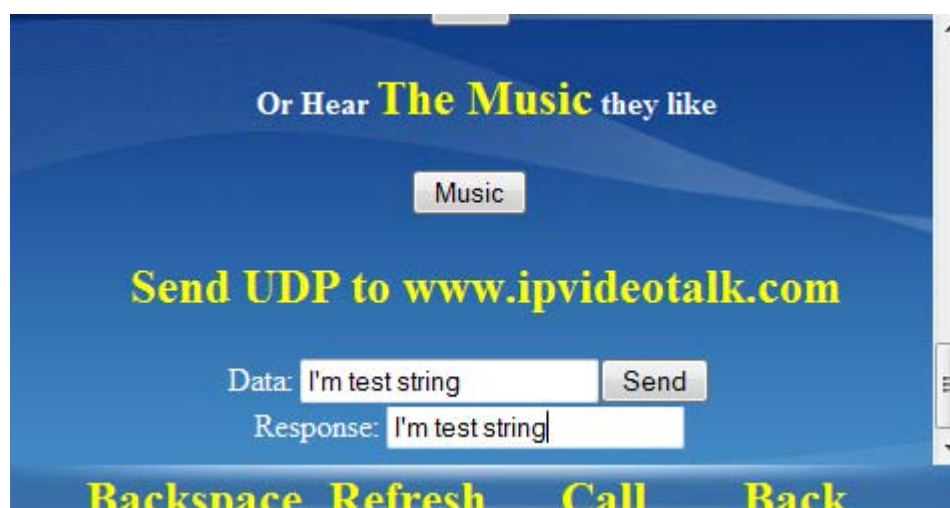


Figure: Send UDP packets to destination host

```

File: call-iframe.html

function cb_udp(result)
{
  //Obtain the response message sent from the destination host
  document.getElementById("udp_response").value = result['Message']
  //Process play result here
  //alert(result['Response']);
}

function udp(data)
{
  //Send some string to the specified port of the destination host via UDP. The timeout
time for the response from the server is 5 seconds.
  GMIEngine.udp("www.ipvideotalk.com", 1500, data, 5, cb_udp)
}

```



```

</td>
</tr>

<tr>
<td align="center">
<fieldset style="border:1px solid blue;margin-left:15px;">
<legend>What's your favorite application in GXV3140?</legend>
//When results are submitted, a function call to myvote.php will be invoked . The
//code in myvote.php includes storing of the vote result and displaying of the poll
//statistics.

<form method="POST" action="myvote.php">
<table>
<tr>
<td>
<input type="radio" name="myvote" value="0" id="myvoteval" checked><font
class="voteitem">IM</font>
</td>
<td>
<input type="radio" name="myvote" value="1"><font class="voteitem">Twitter</font>
</td>
</tr>
<tr>
<td>
<input type="radio" name="myvote" value="2"><font class="voteitem">Facebook</font>
</td>
<td>
<input type="radio" name="myvote" value="3"><font class="voteitem">Skype</font>
</td>
</tr>
<tr>
<td>
<input type="radio" name="myvote" value="4"><font class="voteitem">Web Browser</font>
</td>
<td>
<input type="radio" name="myvote" value="5"><font class="voteitem">Internet Radio</font>
</td>
</tr>

```

```

        <tr>
        <td>
        <input type="radio" name="myvote" value="6"><font class="voteitem">Onli
ne Music</font>
        </td>
        <td>
        <input type="radio" name="myvote" value="7"><font class="voteitem">Onli
ne Video</font>
        </td>
        </tr>
        <p align="center"><input type="submit" value="Submit" name="vote" id="v
otesub" style="display:none"></p>
        </table>
    </form>
</fieldset>
</td>
</tr>
</table>
/*****
* Draw the softkey in html
*****/
<table id="softkey">
    <tr width="100%">
        <td width="40px">
        </td>
        <td width="100px" align="center">
            <a href="myvote.php?vote=1" style="text-decoration:none;"><font class="soft
key">View Result</font></a>
        </td>
        <td onclick="GMIEngine.refresh();" width="100px" align="center">
            <a><font class="softkey">Refresh</font></a>
        </td>
        <td width="100px" align="center" onclick="document.getElementById('votesub')
.click();">
            <a><font class="softkey">Submit</font></a>
        </td>
        <td onclick=" GMIEngine.historypage(-1);" width="100px" align="center" >
            <a><font class="softkey">Back</font></a>
        </td>
    </tr>

```



Figure 5: Vote Example illustration 2

```

文件: myvote.php
<?
//Database configuration
$mysqlserver = "localhost"; // hostname of mysql server
$mysqldb = "test"; // database containing table
$mysqluser = "root"; // user with priveledges to add/modify table
$mysqlpass = "s1ptest"; // password for above user
$mysqltable = "myvote"; // table containing data

//Categories
$option[0]='IM';
$option[1]='Twitter';
$option[2]='Facebook';
$option[3]='Skype';
$option[4]='Web Browser';
$option[5]='Internet Radio';
$option[6]='Online Music';
$option[7]='Online Video';

//Background color for the category statistics bar
$bgcolor[0]='#04F9FA';
$bgcolor[1]='#E76F48';

```

```

$bgcolor[2]='#3AC23A';
$bgcolor[3]='#D1D238';
$bgcolor[4]='#EE0EF2';
$bgcolor[5]='#ff0000';
$bgcolor[6]='#3EF812';
$bgcolor[7]='#F9E4C5';

// for basic functions, you won't need to change anything after this, though you
// might want to. It's pretty ugly as it stands.

?>
<html><head><title>Vote</title>
<link href='../css/main.css' type='text/css' rel='stylesheet'>
<script language="javascript" src="../js/GMI.js"></script>
<script language="javascript">

//This function process es key events. Here, only F4 key stroke is processed, and
// returns to the last page visited.
function softkey_process(keyCode)
{
    parent.document.title = "myvote-iframe, in softkey_process, keyCode = " +
    keyCode;

    switch (keyCode)
    {
        case 112:
            //F1 process here;
        case 13:
            //Enter process here
            break;

        case 113:
            //F2 process here;
            break;

        case 114:
            //F3 process here;
            break;

        case 115:
            //F4 process here
            GMIEngine.historypage(-1);
            break;
    }
}

```



```

case 37:
    //Left process here
    break;

case 38:
    //Up process here
case 40:
    //Down process here
    break;

case 39:
    //Right process here
    break;

default:
    break;
}
}

</script>
</head>
<body onkeydown="softkey_process(event.keyCode);">
<table id='vote_result' align="center"><tr><td>
/******
* Query the database and output the result as html
*****/
<?php
    $conn = mysql_pconnect($mysqlserver,$mysqluser,$mysqlpass) or die("Sorry
,cannot connect");
    //if(mysql_num_rows(mysql_query("SHOW          TABLES          LIKE
''.$mysqltable.'''))==1) {}
    $result = mysql_list_tables($mysqladb);
    $tableexist = false;
    for ($i = 0; $i < mysql_num_rows($result); $i++){
        if(mysql_tablename($result, $i) == $mysqltable){
            $tableexist = true;
            break;
        }
    }
    if(!$tableexist){
        printf ("Table is not exist,creating...\n");
        $sql = 'CREATE TABLE `myvote` (
            `name` VARCHAR( 30 ) NOT NULL,

```

```

        `votes` INT( 4 ),
        PRIMARY KEY ( `name` )
    );
    mysql_query( $sql, $conn );
}
mysql_free_result($result);
?>

<?php
    $selvalue = $_POST["myvote"];
    $getval = $_GET["vote"];
    $name = mt_rand();
    if($selvalue != "" || $getval == 1){
        mysql_pconnect($mysqlserver,$mysqluser,$mysqlpass);
        if(!mysql_db_query($mysqldb,"insert into $mysqltable values
($name','$selvalue')")) {
            print mysql_errno().": ".mysql_error()."<BR>";
        }

        $sumresult=mysql_db_query($mysqldb,"select votes from $mysqltable");
        $sum = (int) mysql_num_rows($sumresult);
        if(!$getval){
            print "<p>Thanks for your vote.</p>\n";
            $result=mysql_db_query($mysqldb,"select votes from
$mysqltable where votes=$selvalue");
            if($result){
                $num = (int) mysql_num_rows($result);
                $per = round($num/$sum,3);
                $percent = $per * 100;
                print "<p>The total number of votes: $sum. Your vote percent:
$percent%</p>\n";
            }
        }else{
            print "<p>The total number of votes: $sum.</p>\n";
        }

        print "<table width='100%' border='0' cellpadding='0' cellspacing='0'
style='margin-top:10px;'>";
        print "<tr><td width='450'>";
        print "<table width='420' border='0' cellpadding='0' cellspacing='0'>";
        for($i=0;$i<8;$i++){
            $per=mysql_db_query($mysqldb,"select votes from
$mysqltable where votes=$i");
            $tempnum = (int) mysql_num_rows($per);

```

```

        $percent = round($tempnum/$sum,3);
        $a = $percent * 100;
        print "<tr><td
style='height:20px;vertical-align:left;width:400px;padding-left:5px;'>";
        print "<div                                class='vote_title'
style='float:left;width:120px;'>$option[$i]:</div>";
        print "<div style='background:#3D97DD;float:left;width:200px;'><div
style='float:left;height:15px;background:$bgcolor[$i];width:$a%;'></div></div>";
        print "<div                                class='vote_title'
style='float:left;padding-left:10px;width:50px;'>$tempnum($a%)</div></td></tr>";
    }
    print "</table>";
    print "</td></tr></table>";
}
else{
    print "<p style='color:red;text-align:center;'>Please select a option
first.</p>\n";
}
?>

</td>
</tr>
</table>
//Draw the softkey in html
<table id="softkey">
    <tr width="100%">
        <td width="40px">
        </td>
        <td width="100px" align="center">
            <a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a>
        </td>

        <td width="100px" align="center">
            <a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a>
        </td>

        <td width="100px" align="center">
            <a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a>
        </td>

        <td onclick=" GMIEngine.historypage(-1);" width="120px" align="center" >
            <a><font class="softkey">Back</font></a>
        </td>
        <td width="40px">
        </td>

```

```

</tr>
</table>

</body>
</html>

```

6.6 Message Example

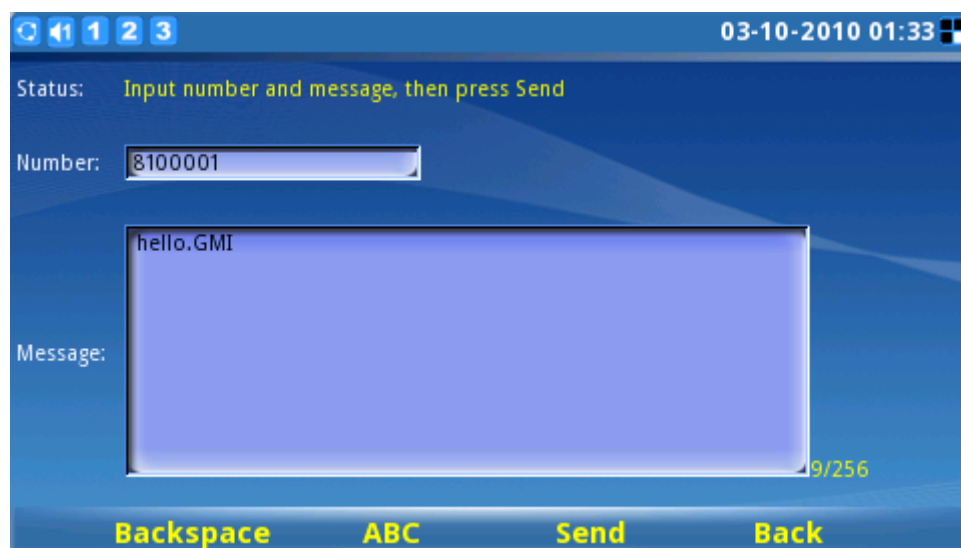


Figure 6: Message Example illustration

Refer to the code and the comments below for the implementation of Message Example:

```

File: message.html
//After the page loads, map the up/down button to tab/alt+tab
function bind_tab()
{
    GMIEngine.mapupdown2tab();
}

//When the page unloads, revoke the mapping between tab/alt+tab
function unbind_tab()
{
    GMIEngine.revokemap();
}

/*****

* After the page loads, map the up/down button to tab button. When the page
unloads, revoke this mapping so that it would not influence other pages.

* * At the same time, all the key events are handled by the softkey_process function.

*****/

```

```

<body onload="bind_tab();number_focus();" onunload="unbind_tab();" onkeydown
="softkey_process(event.keyCode);" class="normal-body">
<table id="myvote-table">

  <tr>
  <td>
    Status:
  </td>
  <td>
    <font id="status" color="yellow">Input number and message, then press
Send</font>
  </td>
  </tr>

  <tr>
  <td>
    Number:
  </td>
  <td>
    <input type="text" id="number">
  </td>
  </tr>

  <tr>
  <td>
    Message:
  </td>
  <td>
    //When there are keypad events in the textarea, obtain the length of textarea and
display it in msgstatus.
    <textarea cols="45" rows="10" id="msg" onkeydown="message_body();"></te
xtarea><font id="msgstatus" color="yellow">0/256</font>
  </td>
  </tr>
</table>

<table id="softkey">
  <tr width="100%">
  <td width="40px">
  </td>
  <td width="100px" align="center">
    <a><font class="softkey">Backspace</font></a>

```

```

</td>
/*****
* Change the input method by calling API function GMIEngine. change_ime. At the
same time, use "chinput" to show the users what input method is being used after
the change.
*****/

<td onclick="GMIEngine.chinput_method('chinput',cb_input_changed);" width="
100px" align="center">
<a><font class="softkey" id="chinput">123</font></a>
</td>

//Send message by calling the API function GMIEngine.message (acct, number,
msg, cb_message)
<td width="100px" align="center" onclick="message(document.getElementById(
'number').value, document.getElementById('msg').value);">
<a><font class="softkey">Send</font></a>
</td>

<td onclick="GMIEngine.historypage(-1);" width="100px" align="center" >
<a><font class="softkey">Back</font></a>
</td>
<td width="40px">
</td>
</tr>
</table>
</body>

```

6.7 News Example



Figure 7: News Example illustration1

Refer to the code and the comments below for the implementation of News Example:

```
File: news.html

/*****
* After the page loads, map the up/down button to tab button. When the page
unloads, revoke this mapping so that it would not influence other pages.
* At the same time, all the key events are handled by the softkey_process function.
*****/

<body onload="bind_tab();" onunload="unbind_tab();" onkeydown="softkey_process(event.keyCode);" id="news-body">
<iframe src="iframe-news.html" id="iframe-news">
</iframe>
<table id="softkey">
  <tr width="100%">
    <td width="40px">
    </td>
    <td width="100px" align="center">
      <a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</a>
    </td>
    <td onclick="GMIEngine.refresh();" width="100px" align="center">
      <a><font class="softkey">Refresh</font></a>
    </td>
  </tr>
</table>
```

```

<td width="120px" align="center">
  <a>&nbsp;&nbsp;&nbsp;</a>
</td>

//Function call to GMIEngine.historypage(-1) to return to the last visited page.
<td onclick="GMIEngine.historypage(-1);" width="100px" align="center" >
  <a><font class="softkey">Back</font></a>
</td>
<td width="40px">
</td>
</tr>
</table>
</body>

```

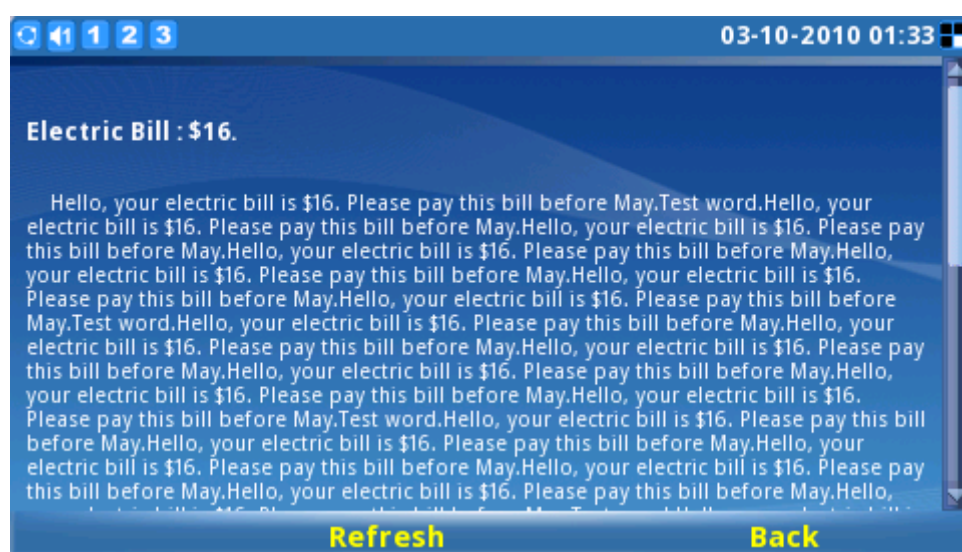


Figure 8: News Example illustration2

```

File: content1.html
<body onload="init()" onkeydown="softkey_process(event.keyCode);" id="news-b
ody">
<iframe src="news1.html" id="iframe-news">
</iframe>
<table id="softkey">
  <tr width="100%">
    <td width="40px">
    </td>
    <td width="100px" align="center">
      <a>&nbsp;&nbsp;&nbsp;</a>
    </td>

    <td onclick="GMIEngine.refresh();" width="100px" align="center">
      <a><font class="softkey">Refresh</font></a>

```



```

</td>

<td width="100px" align="center">
  <a>&nbsp;&nbsp;&nbsp;</a>
</td>

<td onclick="GMIEngine.historypage(-1);" width="100px" align="center" >
  <a><font class="softkey">Back</font></a>
</td>
<td width="40px">
</td>
</tr>
</table>
</body>

```

6.8 Timer Example



Figure 9: Timer Example illustration

Refer to the code and the comments below for the implementation of Timer Example:

```

File: timer.html
.....
/******
*After the page loads, it is executed immediately and the timer is activated. The
timer will display the current time in innerHTML every second.
*****/
function clockon() {
  thistime = new Date();
  hours = thistime.getHours();

```

```

minutes = thistime.getMinutes();
seconds = thistime.getSeconds();
if (eval(hours) < 10) {hours = "0" + hours};
if (eval(minutes) < 10) {minutes = "0" + minutes};
if (seconds < 10) {seconds = "0" + seconds};
thistime = hours+":" + minutes + ":" + seconds;
if (document.all) {
    for (i = 0;i <= clockid.length-1; i++) {
        var thisclock = eval(clockid[i]);
        thisclock.innerHTML = thistime;
    }
}
if (document.getElementById) {
    for (i = 0; i <= clockid.length - 1; i++) {
        document.getElementById(clockid[i]).innerHTML = thistime;
    }
}
var timer = setTimeout("clockon()",1000);
}
}
.....
/*****
* After the page loads, start the timer and display the time.
* At the same time, all the key events are handled by the softkey_process function.
*****/
<body onLoad="clockon()" onkeydown="softkey_process(event.keyCode);" id="myvote-body">
<table id="myvote-table">
<tr>
<td align="center">
    <form id="clock">
        <font color='#FF6699' face='tahoma,Arial,Helvetica,sans-serif' style='font-size:20px;'>
            <div id="layer2" style="z-index:2;width:57px;height:12px;position:absolute;top:95px;left:230px;">
                <script>writeclock()</script>
            </div>
            <table width="4%" cellspacing="0" cellpadding="0" border="0">
                <tr>
                    <td></a>
                </td>
            </tr>
        </table>
    </form>
</td>
</tr>
</table>

```

