



App Note

VoxStack GSM Gateway API

第一部分：OpenVox GSM Gateway AMI 接口的介绍：

1.1: 接口简介

OpenVox 的 GSM 网关 AMI 接口允许客户端应用程序连接网关的 asterisk，并且允许客户端通过 TCP/IP 流去发送命令和读取事件。这是非常有用的，例如：你可以通过这个接口去发送/读取 SMS。

1.2: 协议特点：

- a: 你需要在发送你的命令之前创建一个连接。
- b: 数据包可以在任何时间双向传递。
- c: 如果数据包的第一行是“Action”，那么表明这个数据包是从客户端传输到服务器（即从应用程序传送到 OpenVox 的 asterisk）。
- d: 如果第一行是“event”,或者” Response “，则表明是服务器的回复消息（即网关的 asterisk 对应用程序的响应）。

第二部分：OpenVox GSM 网关的 AMI 配置：

2.1: 概述

网关和应用程序之间的链接使用的是 5038 端口。因此我们需要创建一个用户，客户端将使用这个用户名和密码来链接 OpenVox 网关的 asterisk。

2.2: 在网关上为链接创建一个用户（WEB 界面方式）

用你的用户名和密码登陆网关（出厂为 admin,admin）。找到 AMI 配置的地方，路径如下(ADVANCED → Asterisk API).配置示例：

| General | |
|---------|---|
| Enable: | <input checked="" type="checkbox"/> ON <input type="checkbox"/> |
| Port: | 5038 |

| Manager | |
|-----------------|--|
| Manager Name: | <input type="text" value="admin"/> |
| Manager secret: | <input type="text" value="admin"/> |
| Deny: | <input type="text" value="0.0.0.0/0.0.0.0"/> |
| Permit: | <input type="text" value="172.16.8.44/255.255.0.0"/> |

说明：**enable**:使能（禁用）AMI 协议，如果你需要使用 AMI 接口，请设置为 on.
Manager Name: 用于登陆的用户名
Manager secret: 用于登陆的密码
Deny: 不允许登陆的 IP 地址和子网掩码。0.0.0.0/0.0.0.0 表示禁止所有客户端的 IP 地址和子网掩码链接。例如你的应用程序运行在 192.168.99.1.那么将不会被允许链接。
Permit: 允许链接的 IP 地址和子网掩码。以上配置说明。除了来自于 172.16.88.44 并且子网掩码为 255.255.0.0 的链接，其他的链接都不会被允许链接。

2.2: 配置用户（直接在配置文件中配置）

请后台登陆网关，找到文件 /etc/asterisk/manage.conf

命令：`vi /etc/asterisk/manage.conf`

```
[general]
bindaddr=0.0.0.0
enabled=yes
port=5038
[admin]
secret=admin
permit=0.0.0.0/0.0.0.0
read=system,call,log,verbose,agent,user,config,dtmf,reporting,cdr,dialplan
write=system,call,log,verbose,command,agent,user,config,reporting,originate
```

你可以编辑这个文件，保存之后重新加载配置即可。

命令：`asterisk -r`
`Core reload`

2.3 :链接 OpenVox 网关的服务器

2.3.1: 使用 telnet 链接

首先你必须发送一个“Login “的动作去登陆你的 OpenVox 网关的 asterisk 服务器，然后输入你的用户名和密码，如下图：

```
root@Openvox-Wireless-Gateway: # telnet 172.16.127.127 5038
Asterisk Call Manager/1.1
action: Login
username: admin
secret: admin

Response: Success
Message: Authentication accepted

Event: FullyBooted
Privilege: system,all
Status: Fully Booted
```

注意：我的网关的 IP 地址是 172.16.127.127，你登陆的时候注意改为你的 IP 地址。

2.3.2: 使用程序登陆，示例如下：

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<unistd.h>

//Login
void login_fun(int sock_fd)
{
    char username[20];
    char secret[20];
    int login_len=0;
    int secret_len=0;
    int res = 0;
    char receive_buf[4096];
    char login_buf[40];

    memset(username,'\0',20);
    memset(secret,'\0',20);
    printf("please input your username\n");
    scanf("%s",username);
```

```

fflush(stdin);
printf("please input your secret\n");
scanf("%s",secret);
fflush(stdin);

memset(login_buf,'\0',40);
sprintf(login_buf,"action:Login\r\nusername:%s\r\nsecret:%s\r\n\r\n",username,secret);

login_len = strlen(login_buf);

if(res = write(sock_fd,login_buf,login_len) == login_len)
{
    sleep(1);
    memset(receive_buf,'\0',4096);
    if(res = read(sock_fd,receive_buf,sizeof(receive_buf)<0)
    {
        perror("login failed\n");
        return;
    }
    printf("%s\n",receive_buf);
    if(NULL != strstr(receive_buf,"Authentication accepted"))
    {
        printf("login success\n");
    }
}

}

int main(void)
{
    int client_socket;
    struct sockaddr_in client_addr;
    char ServerIp[20];

    memset(ServerIp,'\0',20);
    printf("Please input your server ip address\n");
    scanf("%s",ServerIp);
    fflush(stdin);

```

```

client_socket = socket(AF_INET,SOCK_STREAM,0);
if(client_socket < 0)
{
    perror("create socket error\n");
    return -1;
}

client_addr.sin_family = AF_INET;
client_addr.sin_port    = htons(5038);
client_addr.sin_addr.s_addr = inet_addr(ServerIp);
//connect
if(connect(client_socket,(struct sockaddr *)&client_addr,sizeof(client_addr))<0)
{
    perror("connect error\n");
    return -1;
}
else
{
    printf("connect to %s success\n",ServerIp);
}
login_fun(client_socket);
return 0;
}

```

第三部分：发送或者接受 SMS

概述：OpenVox 网关扩展了 AMI 命令和 AMI 事件，允许你能通过 AMI 接口去发送或者接受 SMS，开发人员可以通过 AMI 接口去集成自己的应用程序用来发送或者接受 SMS。接下来我们将演示如何通过 OpenVox 网关的 AMI 接口来发送或者接受 SMS。

命令介绍：

GSM send sms <destination> <message> <timeout> [id]

GSM send ussd <destination> <message> <time out> [id]

参数说明：
span: 发送短信所使用的通道，通俗的说就是第几张 SIM 卡
Destination:目的地，及需要发送到哪儿；
Message:短信内容
Timeout:超时时间
Id: 命令的区别身份，可选参数

示例：

第一步：登陆网关

第二步：运行命令：asterisk -r

```
root@Openvox-Wireless-Gateway:~# asterisk -r
Cannot read termcap database;
using dumb terminal settings.
Openvox-Wireless-Gateway*CLI> █
```

第三步：使用命令发送 SMS

我们使用第一个通道发送 “hello world”到手机 13632919026

```
root@Openvox-Wireless-Gateway:~# asterisk -r
Cannot read termcap database;
using dumb terminal settings.
Openvox-Wireless-Gateway*CLI> gsm send sms 1 13632919026 "hello world"
Send SMS to 13632919026 on span 1 at 01:56:09
Openvox-Wireless-Gateway*CLI> █
```

3.2: 使用 AMI 接口发送短信

Action: command

说明： 执行一条 CLI 命令

示例:

第一步：登陆 AMI 服务器

第二步：使用 AMI 动作发送 SMS

```
root@Openvox-Wireless-Gateway:~# telnet 172.16.127.127 5038
Asterisk Call Manager/1.1
action: Login
username: admin → login the AMI server
secret: admin
events: off

Response: Success
Message: Authentication accepted → login success!

action: Command
command: gsm send sms 2 13632919026 "hello world"

Response: Follows
Privilege: Command → send SMS to 13632919026 use span 2
--END COMMAND-- → send success!
```

3.3: 使用 AMI 接口接受短信

说明： 如果网关接受到短信，网关的 asterisk 将会向连结到该网关的客户端报告事件。注意：如果你需要知道发送的状态（例如：发送成功或者失败，请使用参数 sync）

示例:

```
gsm send sync sms <span> <destination> <message> <timeout> [id]
```

接受短信示例:

```
action: command
command: gsm send sync sms 1 13632919026 "hello" 30

Response: Follows successfully
Privilege: Command
SPAN:1 SEND SMS TO PHONE:13632919026 SUCCESSFULLY
--END COMMAND--

action: command
command: gsm send sync sms 1 13632919026 "hello" 30

Response: Follows I without inset SIM card,failed
Privilege: Command
SPAN:1 SEND SMS TO PHONE:13632919026 TIMEOUT
--END COMMAND--
```

```
Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: SMSSRC
Value: +8613632919026
Uniqueid: 1377584220.2

Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: SMSTXT
Value: test
Uniqueid: 1377584220.2

Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: SMSPPDU
Value: 0891683108705505F0040D91683136929120F600003180724161932304F4F29C0E
Uniqueid: 1377584220.2

Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: SMSTIME
Value: 2013/08/27 14:16:39
Uniqueid: 1377584220.2

Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: SMSTZ
Value: GMT+8
Uniqueid: 1377584220.2

Event: VarSet
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Variable: DIALSTATUS
Value: SMS_END
Uniqueid: 1377584220.2
```

```
Event: Newexten
Privilege: dialplan,all
Channel: EXTRA-SMS/3-1
Context: gsm-2
Extension: sms
Priority: 1
Application: System
AppData: /my_tools/process_sms "2" "+8613632919026" "2013/08/27 14:16:39" "test"
Uniqueid: 1377584220.2
```

重要的是画框的部分；

2: 表示接收到短信的通道，
+86....:表示的是发送者的号码
后面的是日期和短信内容。

3.3: 使用程序接受和发送短信

```
void sendsms_fun(int sock_fd)
{
    char send_buf[4096];
    char span_num[3];
    char destination[12];
    char message[2048];
    int res;
    int send_len;
    char receive_buf[4096];

    memset(send_buf,'\0',4096);
    memset(destination,'\0',11);
    memset(message,'\0',2048);
    memset(span_num,'\0',3);

    printf("please input the span you want used\n");
    scanf("%s",span_num);
    fflush(stdin);
    printf("please input the destination num you want send\n");
    scanf("%s",destination);
    fflush(stdin);
    printf("Please input the message you want send\n");
    scanf("%s",message);
    fflush(stdin);

    sprintf(send_buf,"action:Command\r\ncommand:GSM send
```



```

sms %s %s %s\r\n\r\n",span_num,destination,message);
send_len = strlen(send_buf);

memset(receive_buf,'\0',4096);
printf("%s\n",send_buf);
if(res = write(sock_fd,send_buf,send_len) == send_len)
{
    sleep(1);
    if(res = read(sock_fd,receive_buf,sizeof(receive_buf))<0)
    {
        perror("send sms error\n");
        return;
    }
    printf("%s\n",receive_buf);
    if(NULL != strstr(receive_buf,"Response: Follows"))
    {
        printf("send SMS success\n");
    }
}
}

```

```

int main(void)
{
    int client_socket;
    struct sockaddr_in client_addr;
    char ServerIp[20];

    memset(ServerIp,'\0',20);
    printf("Please input your server ip address\n");
    scanf("%s",ServerIp);
    fflush(stdin);

    client_socket = socket(AF_INET,SOCK_STREAM,0);
    if(client_socket < 0)
    {
        perror("create socket error\n");
        return -1;
    }
}

```

```

client_addr.sin_family = AF_INET;
client_addr.sin_port    = htons(5038);
client_addr.sin_addr.s_addr = inet_addr(ServerIp);

if(connect(client_socket,(struct sockaddr *)&client_addr,sizeof(client_addr))<0)
{
    perror("connect error\n");
    return -1;
}
else
{
    printf("connect to %s success\n",ServerIp);
}
//login
login_fun(client_socket);
//send SMS
sendsms_fun(client_socket);
return 0;
}

```

Now use the program receive SMS.

This program base on our first program.

Compile command: gcc send_sms.c -o send_sms

```

void readsms_fun(int sock_fd)
{
    struct Message
    {
        char sender_num[12];
        char span[3];
        char message_buf[4096];
        char receive_time[30];
    };
    char receive_buf[4096];
    char *receive_AppData = NULL;
    struct Message SMS_buf;
    int res;
    int i = 0;
}

```

```

char *temp = NULL;
char temp_buf[1024];

while(1)
{
    sleep(1);
    memset(receive_buf,'\0',4096);
    memset(SMS_buf.sender_num,'\0',12);
    memset(SMS_buf.span,'\0',3);
    memset(SMS_buf.message_buf,'\0',4096);
    memset(SMS_buf.receive_time,'\0',30);
    memset(temp_buf,'\0',1024);

    if(res = read(sock_fd,receive_buf,sizeof(receive_buf))<0)
    {
        perror("read failed\n");
    }
    else
    {
        if((receive_AppData = strstr(receive_buf,"AppData"))!=NULL)
        {
            printf("%s\n",receive_AppData);
            strcpy(temp_buf,receive_AppData);
        }
        if((temp=strstr(temp_buf,"process_sms"))!=NULL)
        {
            for(i = 0;i<13;i++)
            {
                temp++;
            }
            strncpy(SMS_buf.span,temp,1);
            for(i=0;i<7;i++)
            {
                temp++;
            }
            strncpy(SMS_buf.sender_num,temp,11);
            for(i =0;i<14;i++)
            {

```

```

        temp++;
    }
    strncpy(SMS_buf.receive_time,temp,19);
    for(i = 0;i<22;i++)
    {
        temp++;
    }
    strcpy(SMS_buf.message_buf,temp);

    printf("span = %s\n",SMS_buf.span);
    printf("num=%s\n",SMS_buf.sender_num);
    printf("time = %s\n",SMS_buf.receive_time);
    printf("message = %s\n",SMS_buf.message_buf);
}
//printf("span = %s\n",SMS_buf.span);
}
}
}

```

```

int main(void)
{
    int client_socket;
    struct sockaddr_in client_addr;
    char ServerIp[20];

    memset(ServerIp,'\0',20);
    printf("Please input your server ip address\n");
    scanf("%s",ServerIp);
    fflush(stdin);
    client_socket = socket(AF_INET,SOCK_STREAM,0);
    if(client_socket < 0)
    {
        perror("create socket error\n");
        return -1;
    }
    client_addr.sin_family = AF_INET;
    client_addr.sin_port = htons(5038);
    client_addr.sin_addr.s_addr = inet_addr(ServerIp);
}

```

```
if(connect(client_socket,(struct sockaddr *)&client_addr,sizeof(client_addr))<0)
{
    perror("connect error\n");
    return -1;
}
else
{
    printf("connect to %s success\n",ServerIp);
}

login_fun(client_socket);
//sendsms_fun(client_socket);
readsms_fun(client_socket);
return 0;
}
```