自动化测试 Action URI 说明

V3.0

目录

1.引言

本文主要介绍了如何通过 URI 指令修改配置或操作话机。

本文还介绍了如何通过脚本实现自动化测试及验证。

2.URI 指令

2.1 修改 P 值

请使用如下 URI 来修改 P 值。

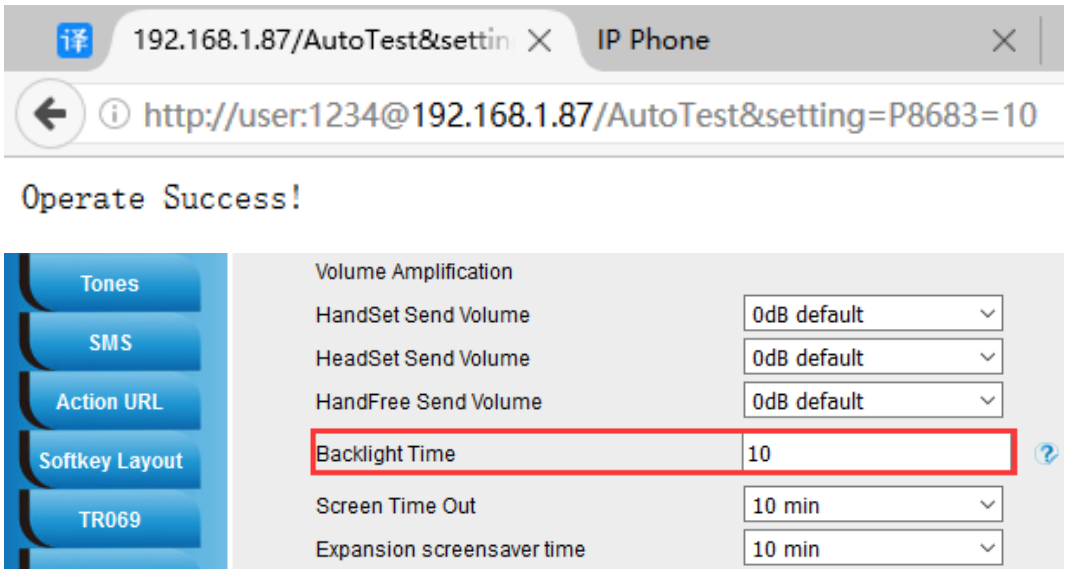
http://user:password@IP_Address/AutoTest&setting=%setting_value%

例:

使用话机 IP 为 192.168.1.87，修改 P 值：P8683。

Action URI 为:

http://user:1234@192.168.1.87/AutoTest&setting=P8683=10

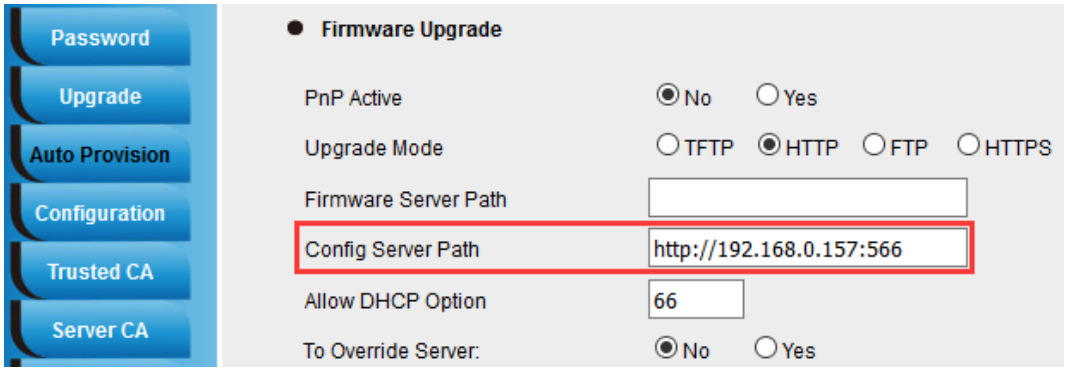


Notes:

1. By default the password for user is 1234.

2.2 检查自动指派

当登录话机网页并填入配置服务器路径 “Config Server Path”后。



请使用如下 URI 来检查自动指派。

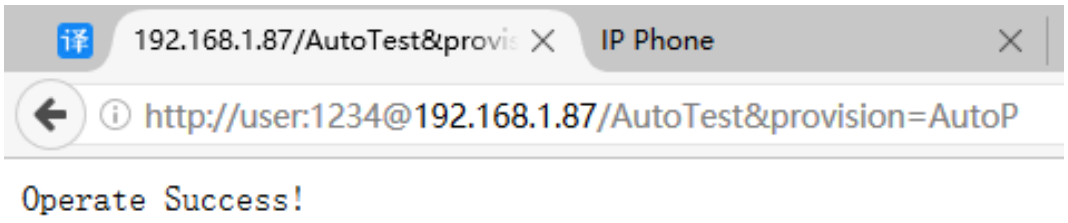
http://user:password@IP_Address/AutoTest&provision=%provision_value%

例:

使用话机 IP 为 192.168.1.87。

Action URI 为:

http://user:1234@192.168.1.87/AutoTest&provision=AutoP



2.3 按键

请使用如下 URI 来操作键盘。

http://user:password@IP_Address/AutoTest&keyboard=%keyboard_value%

2.3.1 当前支持的按键

To answer the call: keyboard=OK/ keyboard=ENTER

To turn on speaker mode: keyboard=SPEAKER

Press transfer button: keyboard=F_TRANSFER

Increasing the volume: keyboard=VOLUME_UP

Reduce the volume: keyboard=VOLUME_DOWN

To mute the call: keyboard=MUTE

To hold the call: keyboard=F_HOLD

To end the call: keyboard=X

To enter the DTMF number (include Numeric, * or # keys): keyboard=0-9/ */ POUND

Press a line key: keyboard=L1-L4

Press a DSS key: keyboard=D1-D10

Press Conference button: keyboard=F_CONFERENCE

Press a soft key: keyboard=F1-F4

Press Message button: keyboard=MSG

Press Headset button: keyboard=HEADSET

Press RD button: keyboard=RD

Press navigation key: keyboard=UP/ DOWN/ LEFT/ RIGHT

To reboot the phone: keyboard=Reboot

To enable DND: keyboard=DNDOn

To disable DND: keyboard=DNDOff

Notes:

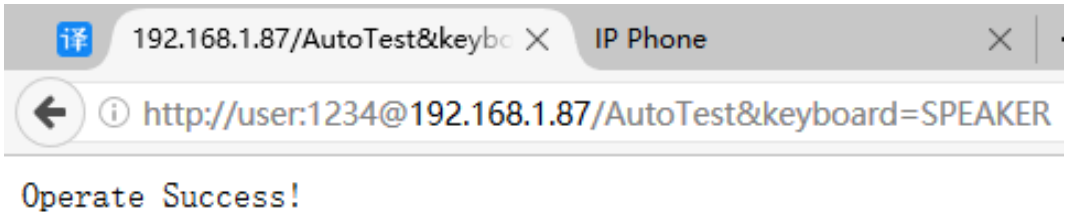
1. **The URI is case sensitive.**
2. **In IDLE status, if you input the URI with keyboard=OK or keyboard=ENTER, the effect is the same as you press "ok" button in the idle status (access to the Status page).**

例:

使用“Action URI”来打开免提模式。

Action URI 为:

http://user:1234@192.168.1.87/AutoTest&keyboard=SPEAKER



2.3.2 免按键拨号

➢ 免按键拨号

例:

使用“Action URI”来实现直接拨号（具体见 3.1.2 Auto_Test_Function.sh 相关配置子章节 New Dial 调用）

相关 Action URI 为:

http://user:1234@192.168.1.87/AutoTest&keyboard=BASICCALL:NUMBER=708

2.4 BSP 相关

请使用如下 URI 来操作话机。

http://user:password@IP_Address/AutoTest&bsp=%bsp_value%

2.4.1 当前支持的功能

- Mute All: bsp=MuteAll
- 清空 hlcfg 文件夹： bsp=RMhlcfg
- 清空 hlfs 文件夹： bsp=RMhlfs

例:

使用“Action URI”来清空 hlcfg 文件夹。

Action URI 为:

http://user:1234@192.168.1.87/AutoTest&bsp=RMhlcfg

- 验证话机 IMG 和 ROM 版本信息

例:

使用“Action URI”来验证话机 IMG 和 ROM 版本信息。（具体见 3.1.2 Auto_Test_Function.sh 相关配置子章节 Check_VersionInfo 调用）

（验证 IMG--1.0.4.36(2018-05-07)16:20:00）

相关 Action URI 为:

http://user:1234@192.168.1.87/AutoTest&bsp=CHECKIMGINFO:IMG--1.0.4.36(2018-05-07)16:20:00

（验证 ROM--1.0.4.36(2018-05-07)16:20:00）

相关 Action URI 为:

http://user:1234@192.168.1.87/AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36(2018-05-07)16:20:00

2.5 GUI 相关

请使用如下 URI 来操作话机。

http://user:password@IP_Address/AutoTest&gui=%gui_value%

2.5.1 当前支持的功能

- 验证 GUI 窗口是否预定窗口

例:

使用“Action URI”来验证 GUI 窗口是否预定窗口。（具体见 3.1.2 Auto_Test_Function.sh 相关配置子章节 Check_GUIStatus 调用）

（验证是否在 Menu→Settings→Basic 中第二图标）

相关 Action URI 为:

http://user:1234@192.168.1.87/AutoTest&gui= LCD_BasicSetting_Win:1

2.6 DRD 特色业务相关

请使用如下 URI 来操作话机。

http://user:password@IP_Address/AutoTest&drd=%drd_value%

2.6.1 当前支持的功能

- Return Idle
- Guest In/Out
  - Guest In
  - Guest Out
- ACD
  - Log In
  - Log Out
  - Available
  - Unavailable
  - Wrap Up
  - Disposition Code
  - CheckACDStatus
- Call Pickup
  - DPick
  - GPick
- Call Park
  - DPark
  - GPark

1） Return Idle

1. Return Idle 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=RETURNIDLE

2） Guest In/Out

1. Guest In 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=GUESTIN:aid=0&number=0028&password=8606

2. Guest Out 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=GUESTOUT

3） ACD

1. Log In 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:LOGIN:aid=0

2. Log Out 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:LOGOUT:aid=0

3. Available 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:AVAILABLE:aid=0

4. Unavailable 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:UNAVAILABLE

5. Wrap Up 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:WRAPUP:aid=0

6. Disposition Code 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:DISPCODE

7. CheckACDStatus 调用

   验证 ACD 状态是否正确（具体见 3.1.2 Auto_Test_Function.sh 相关配置子章节 CheckACDStatus 调用）

   例:

   使用“Action URI”来验证 ACD 状态是否为 LOGOUT。

   相关 Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=ACD:CHECK:IFLOGOUT

4） Call Pickup

1. DPick 调用

   注：调用此 URI 前，需先对 DUT 执行按 Speaker 操作。

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=CALLPICKUP:DPICK

2. GPick 调用

   注：调用此 URI 前，需先对 DUT 执行按 Speaker 操作。

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=CALLPICKUP:GPICK

5） Call Park

1. DPark 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=CALLPARK:DPARK

2. GPark 调用

   Action URI 为:

   http://user:1234@192.168.1.87/AutoTest&drd=CALLPARK:GPARK

3.自动化测试脚本

在 linux 终端，我们可以使用如下指令来实现 Action URI 功能。

wget --http-user=user --http-password=1234 "http://192.168.1.87/AutoTest&keyboard=2"

3.1 脚本模板

目前脚本模块化，具体分为：

- Auto_Test_Main.sh 主脚本
- Auto_Test_Function.sh 功能函数脚本
- Auto_Test_Config.sh 参数配置脚本
- Auto_Test_Call.sh 具体测试话机 Call 功能脚本
- Auto_Test_Transfer.sh 具体测试话机 Transfer 功能脚本
- Auto_Test_XXXXXX.sh XXXXXX 后续可依次添加

3.1.1 Auto_Test_Main.sh 相关配置

1） 修改环境变量设置路径

- CONFIG_PATH（所有测试脚本所在路径）
- Auto_Test_log_file（测试日志文件所在路径）

```
#环境变量设置
export CONFIG_PATH="/home/Work/simon/work/H1-uClinux-uc8xx-v1.0.3/user/Auto_Test"
export Auto_Test_log_file="/home/Work/simon/work/H1-uClinux-uc8xx-v1.0.3/user/Auto_Test/Backup/Auto_Test_Syslog.xml"
```

2） 添加具体测试话机功能脚本函数

后续添加了其他功能脚本需在 Auto_Test_Main.sh 主脚本中添加。

```
Auto_Test_Main.sh | Auto_Test_Function.sh | Auto_Test_Config.sh | Auto_Test_Call.sh
7   ######数据定义区引用
8   source $CONFIG_PATH/Auto_Test_Config.sh          #定义自动化测试的全局变量
9   source $CONFIG_PATH/Auto_Test_Function.sh        #定义自动化测试的工具函数
10
11  ######测试功能引用
12  source $CONFIG_PATH/Auto_Test_Call.sh            #定义自动化测试Call脚本函数
13
14  ## 判断备份文件夹是否存在，不存在则创建
15  if [ ! -d "$CONFIG_PATH/Backup" ]; then
```

3） 执行功能段设置

执行 Auto_Test_Main.sh 时，带入参来选择执行测试功能段。

例：

./ Auto_Test_Main.sh CALL　　#执行 Call_function

./ Auto_Test_Main.sh TRANSFER　#执行 Transfer_function

```
43  #执行功能段设置
44  if [ $1 = "CALL" ]; then
45      echo "Auto Test Call Function!!!"
46      Call_Function;
47  elif [ $1 = "TRANSFER" ]; then
48      echo "Auto Test Transfer Function!!!"
49      Transfer_Function;
50  elif [ $1 = "FORWARD" ]; then
51      echo "Auto Test Forward Function!!!"
52      Forward_Function;
53  elif [ $1 = "ALL" ]; then
54      echo "Auto Test Forward Function!!!"
55      Call_Function;
56      Transfer_Function;
57      Forward_Function;
58  else
59      echo "Please make sure the positon varial
60  fi
```

3.1.2 Auto_Test_Function.sh 相关配置

功能函数脚本调用：

- CheckState_Main 验证状态主函数
- Screenshot 截图功能函数
- Dial 拨号函数
- Get_MemoryFree 获取内存信息函数
- Check_PValue 验证 P 值生效函数
- Check_GUIStatus 验证 GUI 窗口函数
- Check_ACDStatus 验证 ACD 状态函数
- Check_VersionInfo 验证 IMG 和 ROM 信息
- New Dial 直接拨号函数

1） CheckState_Main 调用

测试脚本中，关键状态点需添加此函数。

例：

CheckState_Main $URL1 $CALLS_STATE_IDLE

如上，共有 2 个入参：

a) 入参$URL1 为需验证的话机（在 Auto_Test_Config.sh 中设置）

b) 入参$CALLS_STATE_IDLE 为需验话机的状态（在 Auto_Test_Config.sh 中设置）

2） Screenshot 调用

注：调用 Screenshot 之前，需在测试功能函数首尾 2 处添加命令。

A. ScreenInitialValue=`expr $ScreenInitialValue + 1`

B. ScreenInitialValue=0

添加位置如下：

```
#基本呼叫
function Call_Function()
{
    for((j=0;j<1000;j++))
    do
        #每执行一次，ScreenInitialValue值加1
        ScreenInitialValue=`expr $ScreenInitialValue + 1`
        #验证状态
        CheckState_Main $URL1 $CALLS_STATE_IDLE

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=SPEAKER"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OFFHOOK
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OFFHOOK" 20

        #拨号705
        Dial $Num2

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=F1"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OUT_GOING
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OUT_GOING" 30

        wget --http-user=$USER --http-password=$PASS $URL3"keyboard=OK"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_TALK

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=X"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_IDLE
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_IDLE" 50
    done
    ScreenInitialValue=0
}
```

Screenshot 需与 CheckState_Main 一起调用，方便查看调用入参。

例：

CheckState_Main $URL1 $CALLS_STATE_IDLE

Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_IDLE" 10

如上，共有 4 个入参：

a) 入参$FUNCNAME 为当前执行的函数名，用于更改截图名称，直接调用即可，无需修改。

b) 入参$SCREENURL1 为需截图的话机（在 Auto_Test_Config.sh 中设置）

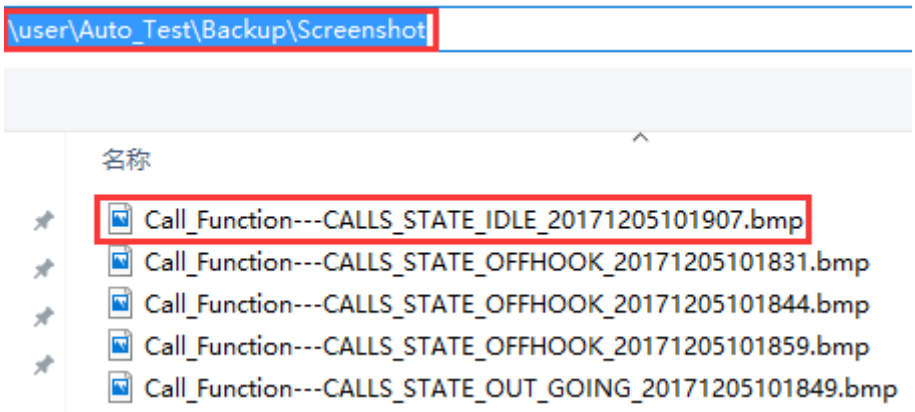
c) 入参"CALLS_STATE_IDLE"，用于更改截图名称，需修改为调用的 CheckState_Main 第二入参名称，方便截图查看。

d) 入参 10 为执行截图条件，当脚本运行 10 次时截图 1 次，可修改为任意整数。

成功调用 Screenshot 后，会截图保存，并命名为：

测试项目---当前状态_截图时间.bmp

存放目录为\Auto_Test\Backup\Screenshot



3） Dial 调用

拨号时调用此函数。

例：

Dial $URL1 $Num2

如上，共有 2 个入参：

a) 入参$URL1 为主叫话机（在 Auto_Test_Config.sh 中设置）

b) 入参$Num2 为被叫号码（在 Auto_Test_Config.sh 中设置）

4） Get_MemoryFree 调用

获取内存时调用此函数。

例：

Get_MemoryFree $URL1

如上，共有 1 个入参：

a) 入参$URL1 为需查询内存信息话机（在 Auto_Test_Config.sh 中设置）

调用成功后，可在 Auto_Test_Syslog.xml 日志文件中查看内存信息：


```
#基本呼叫
function Call_Function()
{

    for((j=0;j<1000;j++))
    do
        #每执行一次，ScreenInitialValue值加1
        ScreenInitialValue=`expr $ScreenInitialValue + 1`
        #验证状态
        CheckState_Main $URL1 $CALLS_STATE_IDLE
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_IDLE" 10

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=SPEAKER"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OFFHOOK
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OFFHOOK" 10

        #拨号705
        Dial $URL1 $Num2

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=F1"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OUT_GOING
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OUT_GOING" 20

        wget --http-user=$USER --http-password=$PASS $URL2"keyboard=OK"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_TALK
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_TALK" 30
    done
    ScreenInitialValue=0
    #获取内存信息
    Get_MemoryFree $URL1
}
```

```
************************************************************ AutoTest Start ************************************************************
Auto Test Call Function!!!
--2017-12-06 09:10:58-- http://192.168.1.103/AutoTest&autoverify=MEMORYFREE
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&autoverify=MEMORYFREE"

    0K                                                      20.7M=0s

2017-12-06 09:10:58 (20.7 MB/s) - "AutoTest&autoverify=MEMORYFREE" saved [68]

************************************************************ Total free memory: 33024 KB ************************************************************
```

5） Check_PValue 调用

验证 P 值是否生效时添加此函数。

例：

Check_PValue $URL1 P35=705

如上，共有 2 个入参：

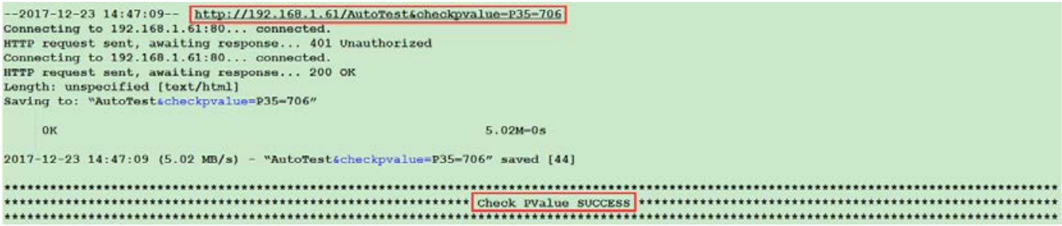
a) 入参$URL1 为需验证的话机（在 Auto_Test_Config.sh 中设置）

b) 入参 P35=705 为需验证的 P 值

调用成功后，可在 Auto_Test_Syslog.xml 日志文件中查看 P 值是否更改成功：

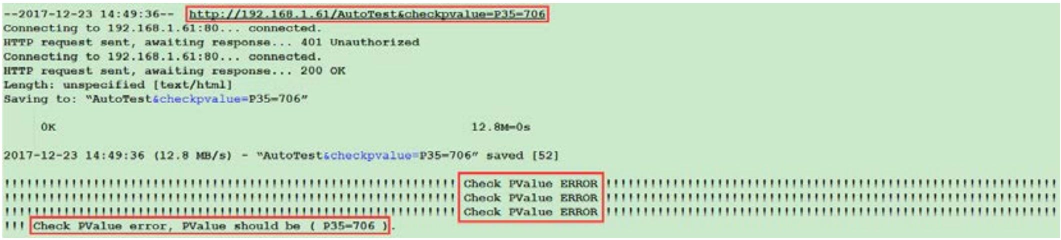
```
4   function Test_Function()
5   {
6
7       for((j=0;j<1;j++))
8       do
9           CheckState_Main ${array_URL[j]} $CALLS_STATE_IDLE
10          Check_PValue $URL1 P35=705
11      done
12  }
```

P 值修改成功：



```
--2017-12-23 14:47:09--  http://192.168.1.61/AutoTest&checkpvalue=P35=706
Connecting to 192.168.1.61:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.61:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: “AutoTest&checkpvalue=P35=706”

    OK                                                        5.02M=0s

2017-12-23 14:47:09 (5.02 MB/s) - “AutoTest&checkpvalue=P35=706” saved [44]

*******************************************************************************************************************
***************************************************************  Check PValue SUCCESS  *****************************
*******************************************************************************************************************
```

P 值修改失败：



```
--2017-12-23 14:49:36--  http://192.168.1.61/AutoTest&checkpvalue=P35=706
Connecting to 192.168.1.61:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.61:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: “AutoTest&checkpvalue=P35=706”

    OK                                                      12.8M=0s

2017-12-23 14:49:36 (12.8 MB/s) - “AutoTest&checkpvalue=P35=706” saved [52]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check PValue ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check PValue ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check PValue ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Check PValue error, PValue should be ( P35=706 ).
```

6） Check_GUIStatus 调用

验证 GUI 窗口是否在预定窗口时添加此函数。

例：

进入 Menu→Settings→Basic，再按一次向右键，当前图标选定为 Time & Date：

查看此时 GUI 窗口状态需输入 URI 如下：

Check_GUIStatus $URL1 LCD_BasicSetting_Win:1

如上，共有 2 个入参：

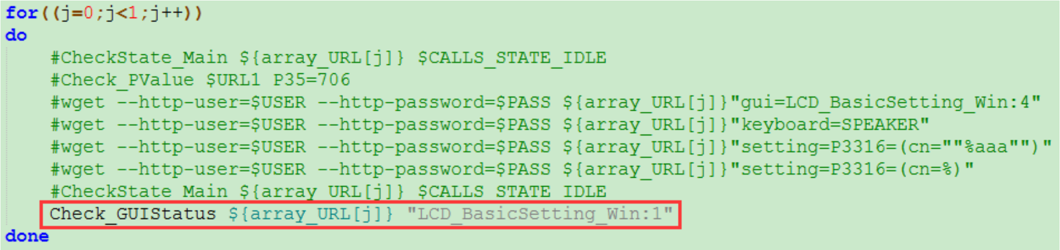
a) 入参$URL1 为需验证的话机（在 Auto_Test_Config.sh 中设置）

b) 入参 LCD_BasicSetting_Win:1 为需验证的窗口。其中：

“:”前 LCD_BasicSetting_Win 为当前所在大窗口标题；

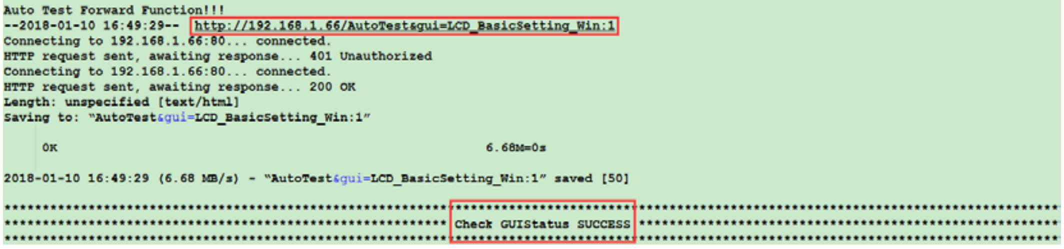
"：" 后 1 为当前高亮小窗口位置（从左至右从上至下依次为 0、1、2、3、4、5、6）

调用成功后，可在 Auto_Test_Syslog.xml 日志文件中查看 GUI 窗口是否正确：



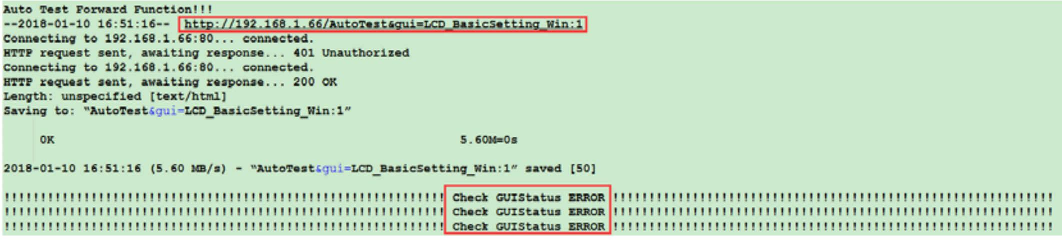
```
for((j=0;j<1;j++))
do
    #CheckState_Main ${array_URL[j]} $CALLS_STATE_IDLE
    #Check_PValue $URL1 P35=706
    #wget --http-user=$USER --http-password=$PASS ${array_URL[j]}"gui=LCD_BasicSetting_Win:4"
    #wget --http-user=$USER --http-password=$PASS ${array_URL[j]}"keyboard=SPEAKER"
    #wget --http-user=$USER --http-password=$PASS ${array_URL[j]}"setting=P3316=(cn=""%aaa"")"
    #wget --http-user=$USER --http-password=$PASS ${array_URL[j]}"setting=P3316=(cn=%)"
    #CheckState_Main ${array_URL[j]} $CALLS_STATE_IDLE
    Check_GUIStatus ${array_URL[j]} "LCD_BasicSetting_Win:1"
done
```

GUI 窗口正确：



```
Auto Test Forward Function!!!
--2018-01-10 16:49:29--  http://192.168.1.66/AutoTest&gui=LCD_BasicSetting_Win:1
Connecting to 192.168.1.66:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.66:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&gui=LCD_BasicSetting_Win:1"

     OK                                                    6.68M=0s

2018-01-10 16:49:29 (6.68 MB/s) - "AutoTest&gui=LCD_BasicSetting_Win:1" saved [50]

******************************************************************************************************************************************
********************************************************** Check GUIStatus SUCCESS *******************************************************
******************************************************************************************************************************************
```

GUI 窗口错误：



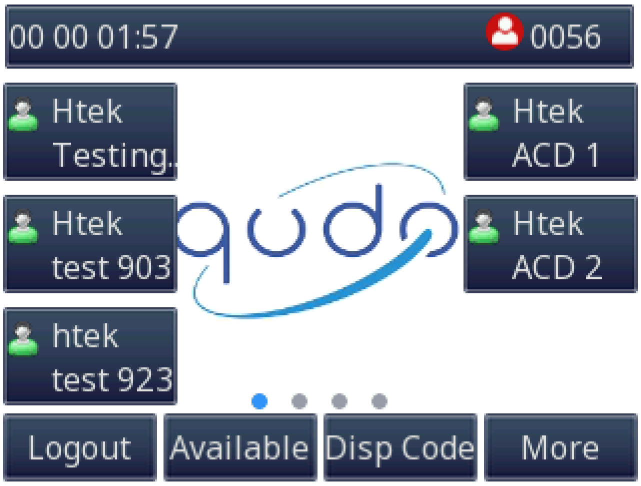
```
Auto Test Forward Function!!!
--2018-01-10 16:51:16--  http://192.168.1.66/AutoTest&gui=LCD_BasicSetting_Win:1
Connecting to 192.168.1.66:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.66:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&gui=LCD_BasicSetting_Win:1"

     OK                                                 5.60M=0s

2018-01-10 16:51:16 (5.60 MB/s) - "AutoTest&gui=LCD_BasicSetting_Win:1" saved [50]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check GUIStatus ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check GUIStatus ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check GUIStatus ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

7） CheckACDStatus 调用

验证 ACD 状态是否正确时添加此函数。

例：

DUT 此时已经 Log In，且处于 Unavailable 状态：



查看此时 ACD 状态需输入 URI 如下：

CheckACDStatus_Main $URL1 "IFUN_AVAILABLE" $SCREENSHORT $FUNCNAME

如上，共有 4 个入参：

a) 入参$URL1 为需验证的话机（在 Auto_Test_Config.sh 中设置）

b) 入参"IFUN_AVAILABLE"为需验证的 ACD 状态：UNAVAILABLE

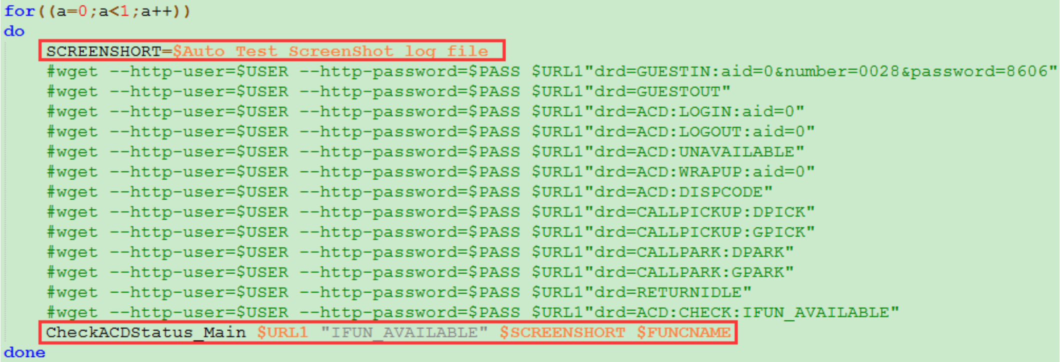
注：当前共支持四个状态

1. "IFLOGOUT"
2. "IFUN_AVAILABLE"
3. "IFAVAILABLE"
4. "IFWRAPUP"

c) 入参$SCREENSHORT 为对应的日志文件

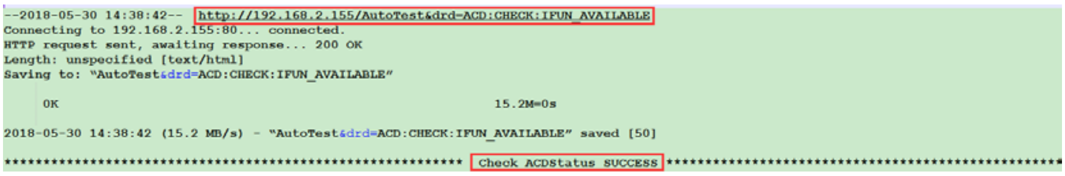
d) 入参$FUNCNAME 为取当前脚本函数名函数

调用成功后，可在对应的日志文件中查看 ACD 状态是否正确：



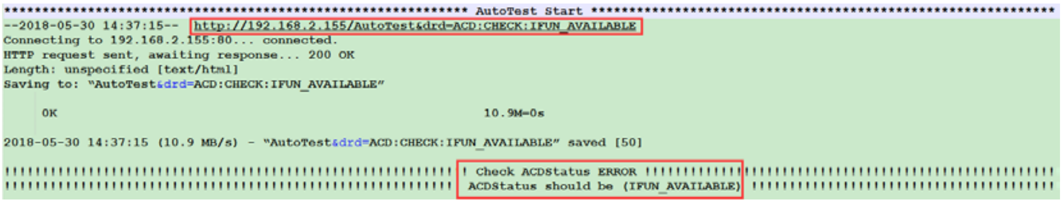
```
for((a=0;a<1;a++))
do
    SCREENSHORT=$Auto_Test_ScreenShot_log_file
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=GUESTIN:aid=0&number=0028&password=8606"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=GUESTOUT"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:LOGIN:aid=0"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:LOGOUT:aid=0"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:UNAVAILABLE"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:WRAPUP:aid=0"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:DISPCODE"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=CALLPICKUP:DPICK"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=CALLPICKUP:GPICK"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=CALLPARK:DPARK"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=CALLPARK:GPARK"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=RETURNIDLE"
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:CHECK:IFUN_AVAILABLE"
    CheckACDStatus_Main $URL1 "IFUN_AVAILABLE" $SCREENSHORT $FUNCNAME
done
```

ACD 状态正确：



```
--2018-05-30 14:38:42--  http://192.168.2.155/AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE
Connecting to 192.168.2.155:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE"

     OK                                                     15.2M=0s

2018-05-30 14:38:42 (15.2 MB/s) - "AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE" saved [50]

************************************************************ Check ACDStatus SUCCESS ****************************************************
```

ACD 状态错误：



```
************************************************************* AutoTest Start *************************************************************
--2018-05-30 14:37:15--  http://192.168.2.155/AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE
Connecting to 192.168.2.155:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE"

     OK                                                     10.9M=0s

2018-05-30 14:37:15 (10.9 MB/s) - "AutoTest&drd=ACD:CHECK:IFUN_AVAILABLE" saved [50]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! ! Check ACDStatus ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  ACDStatus should be (IFUN_AVAILABLE) !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

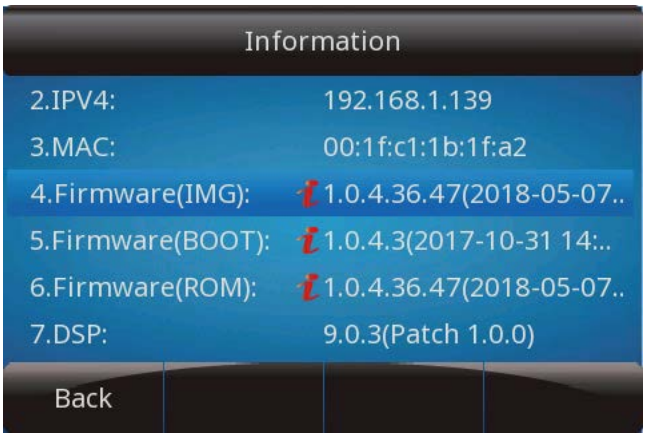
8） Check_VersionInfo 调用

验证话机 IMG 和 ROM 版本信息。

例：

IMG 信息：IMG--1.0.4.36.47(2018-05-07)16:20:00

ROM 信息：ROM--1.0.4.36.47(2018-05-07)16:20:00



查看此时 IMG 状态需输入 URI 如下：

Check_VersionInfo $URL1 "IMGINFO:IMG--1.0.4.36.47(2018-05-07)16:20:00"

查看此时 ROM 状态需输入 URI 如下：

Check_VersionInfo $URL1 "ROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00"

如上，共有 2 个入参：

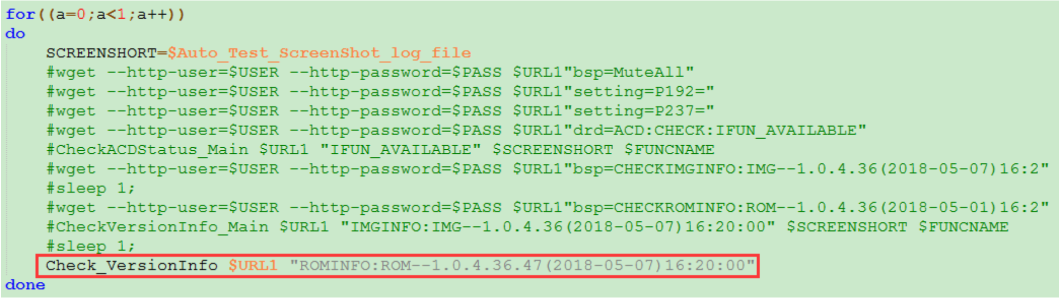
a) 入参$URL1 为需验证的话机（在 Auto_Test_Config.sh 中设置）

b) 入参"IMGINFO:IMG--1.0.4.36.47(2018-05-07)16:20:00"为需验证的 IMG 信息

入参 "ROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00" 为需验证的 ROM 信息

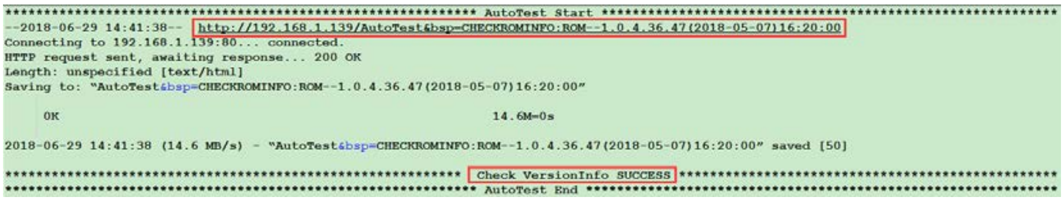
调用成功后，可在对应的日志文件中查看 IMG/ROM 信息是否正确。

以验证 ROM 信息为例：



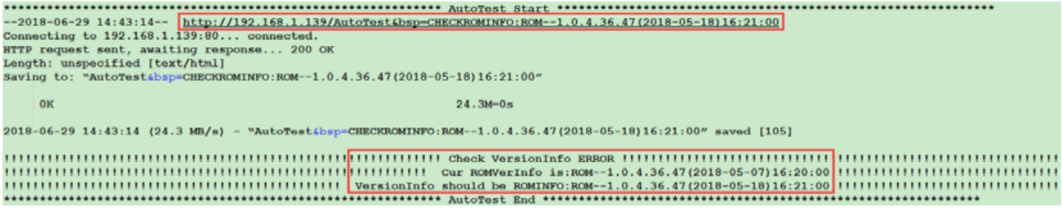
```
for((a=0;a<1;a++))
do
    SCREENSHORT=$Auto_Test_ScreenShot_log_file
    #wget --http-user=$USER --http-password=$PASS $URL1"bsp=MuteAll"
    #wget --http-user=$USER --http-password=$PASS $URL1"setting=P192="
    #wget --http-user=$USER --http-password=$PASS $URL1"setting=P237="
    #wget --http-user=$USER --http-password=$PASS $URL1"drd=ACD:CHECK:IFUN_AVAILABLE"
    #CheckACDStatus_Main $URL1 "IFUN_AVAILABLE" $SCREENSHORT $FUNCNAME
    #wget --http-user=$USER --http-password=$PASS $URL1"bsp=CHECKIMGINFO:IMG--1.0.4.36(2018-05-07)16:2"
    #sleep 1;
    #wget --http-user=$USER --http-password=$PASS $URL1"bsp=CHECKROMINFO:ROM--1.0.4.36(2018-05-01)16:2"
    #CheckVersionInfo_Main $URL1 "IMGINFO:IMG--1.0.4.36(2018-05-07)16:20:00" $SCREENSHORT $FUNCNAME
    #sleep 1;
    Check_VersionInfo $URL1 "ROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00"
done
```

ROM 信息正确：



```
************************************************************ AutoTest Start ************************************************************
--2018-06-29 14:41:38--  http://192.168.1.139/AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00
Connecting to 192.168.1.139:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: “AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00”

     0K                                                          14.6M=0s

2018-06-29 14:41:38 (14.6 MB/s) - “AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-07)16:20:00” saved [50]

*********************************************************** Check VersionInfo SUCCESS ***********************************************************
************************************************************ AutoTest End ************************************************************
```

ROM 信息错误：



```
************************************************************ AutoTest Start ************************************************************
--2018-06-29 14:43:14--  http://192.168.1.139/AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-18)16:21:00
Connecting to 192.168.1.139:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: “AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-18)16:21:00”

     0K                                                          24.3M=0s

2018-06-29 14:43:14 (24.3 MB/s) - “AutoTest&bsp=CHECKROMINFO:ROM--1.0.4.36.47(2018-05-18)16:21:00” saved [105]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Check VersionInfo ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  Cur ROMVerInfo is:ROM--1.0.4.36.47(2018-05-07)16:20:00 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!  VersionInfo should be ROMINFO:ROM--1.0.4.36.47(2018-05-18)16:21:00 !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
************************************************************ AutoTest End ************************************************************
```

9） New Dial 调用

直接拨号功能，与 Dial 调用类似。

1. 修改 Auto_Test_Function.sh 中 Dial 功能函数。修改如下：

```
function Dial {

    #Number=$2
    #取Number中3位号码，若有多位号码，可在此处依次添加
    #number1=`echo ${Number:0:1}`
    #number2=`echo ${Number:1:1}`
    #number3=`echo ${Number:2:1}`
    #number4=`echo ${Number:3:1}`

    wget --http-user=$USER --http-password=$PASS $1"keyboard=BASICCALL:NUMBER="$2
    usleep $STIME5
    #拨号，若有多位号码，可在此处依次添加
    #wget --http-user=$USER --http-password=$PASS $1"keyboard=$number1"
    #if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; continue; fi
    #usleep $STIME5
    #wget --http-user=$USER --http-password=$PASS $1"keyboard=$number2"
    #if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; continue; fi
    #usleep $STIME5
    #wget --http-user=$USER --http-password=$PASS $1"keyboard=$number3"
    #if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; continue; fi
    #usleep $STIME5
    #wget --http-user=$USER --http-password=$PASS $1"keyboard=$number4"
    #if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; continue; fi
    #usleep $STIME5

}
```

2. 隐掉所有小脚本中 Dial 函数之后的 Send 操作。修改如下：

```
    #拨号
    sleep $STIME1
    Dial $1 $4

    #echo "****************************************** 话机A按键send，把号码拨出去*****************************************************"
    #wget --http-user=$USER --http-password=$PASS $1"keyboard=F1"
    #if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; continue; fi

    echo "****************************************** 判断话机A是否处于呼出态*****************************************************"
    CheckState_Main $1 $CALLS_STATE_OUT_GOING $SCREENSHORT $FUNCNAME
    #Screenshot $FUNCNAME $5 "CALLS_STATE_OUT_GOING" 1 $SCREENSHORT $FUNCNAME
    sleep $STIME6
```

注：Blind Transfer 业务无法使用此功能函数。

3.1.3 Auto_Test_Config.sh 相关配置

此脚本用于设置：

- 话机 IP

```
#设置用于测试的话机IP
#DUT1
HOST1="http://192.168.1.103/"
#DUT2
HOST2="http://192.168.1.220/"
#DUT3
HOST3="http://192.168.1.226/"
```

- 测试 URL

```
#测试URL设置
URL1=$HOST1$COMMPATH
URL2=$HOST2$COMMPATH
URL3=$HOST3$COMMPATH
```

- 话机 Number

```
#设置用于测试的话机Number
#DUT1
Num1=706
#DUT2
Num2=705
#DUT3
Num3=707
```

- 截图 URL

```
#截图URL设置
SCREENURL1=$HOST1$SCREENSHOT
SCREENURL2=$HOST2$SCREENSHOT
SCREENURL3=$HOST3$SCREENSHOT
```

- 常用变量（如 sleep 时间）

```
#常用变量设置
STIMEVerify=1     #验证等待时间s
STIME1=2          #按其他键间隔s
STIME2=5          #通话持续时间s
STIME6=5          #接通前等待间隔s
#usleep时间设置
STIME5=500000     #数字键拨号间隔us
```

- 话机状态

```
#话机状态
CALLS_STATE_IDLE=0                #FXSState=0x80 CallCtlState=0x60 LCMState=-1  /* 空闲态 */
CALLS_STATE_OFFHOOK=1             #FXSState=0x82 CallCtlState=0x61 LCMState=4   /* Sperker摘机态 */
CALLS_STATE_OUT_GOING=2           #FXSState=0x82 CallCtlState=0x62 LCMState=5   /* Sperker呼出态 */
CALLS_STATE_TALK=3                #FXSState=0x82 CallCtlState=0x64 LCMState=6   /* Sperker基本通话态 */
CALLS_STATE_RING=4                #FXSState=0x80 CallCtlState=0x63 LCMState=3   /* 振铃态 */
CALLS_STATE_HOLD=5                #FXSState=0x82 CallCtlState=0x87 LCMState=7   /* 普通IPPhone Hold对方 */
CALLS_STATE_NS_INITIATE=6         #FXSState=0x82 CallCtlState=0x81 LCMState=11  /* 新业务等待用户拨号态 */
CALLS_STATE_NS_TALK=7             #FXSState=0x82 CallCtlState=0x82 LCMState=6   /* 新业务通话态（和其中一方） */
CALLS_STATE_CONFERENCE=8          #FXSState=0x82 CallCtlState=0x88 LCMState=9   /* 会议状态 */
CALLS_STATE_CONF_HOLD=9           #FXSState=0x82 CallCtlState=0x8d LCMState=9   /* 会议HOLD状态 */
```

CALLS_STATE_IDLE=0 /* 空闲态 */

CALLS_STATE_OFFHOOK=1 /* Sperker 摘机态 */

CALLS_STATE_OUT_GOING=2 /* Sperker 呼出态 */

CALLS_STATE_TALK=3 /* Sperker 基本通话态 */

CALLS_STATE_RING=4 /* 振铃态 */

CALLS_STATE_HOLD=5 /* 普通 IPPhone Hold 对方 */

CALLS_STATE_NS_INITIATE=6 /* 新业务等待用户拨号态 */

CALLS_STATE_NS_TALK=7 /* 新业务通话态（和其中一方） */

CALLS_STATE_CONFERENCE=8 /* 会议状态 */

CALLS_STATE_CONF_HOLD=9 /* 会议 HOLD 状态 */

- 话机 GUI 状态

```
#话机GUI状态
    LCD_Idle_Win,
    LCD_Menu_Win,
    LCD_Status_Win,
    LCD_Information_Win,
    LCD_Network_Win,
    LCD_Account_Win,
    LCD_Feature_Win,
    LCD_CallForword_Win,
    LCD_AlwaysForword_Win,
    LCD_BusyForword_Win,
    LCD_NoAnswer_Win,
    LCD_FunctionKey_Win,
    LCD_LineasFunction_Win,
    LCD_KeyasSend_Win,
```

LCD_Idle_Win,

LCD_Menu_Win,

LCD_Status_Win,

LCD_Information_Win,

LCD_Network_Win,

LCD_Account_Win,

LCD_Feature_Win,

LCD_CallForword_Win,

LCD_AlwaysForword_Win,

LCD_BusyForword_Win,

LCD_NoAnswer_Win,

LCD_FunctionKey_Win,

LCD_LineasFunction_Win,

LCD_KeyasSend_Win,

LCD_HotLine_Win,

LCD_AnonymousCall_Win,

LCD_AnonymousSec_Win,

LCD_DND_Win,

LCD_RecordHistory_Win,

LCD_Diretory_Win,

LCD_AllContacts_Win,

LCD_LDAPList_Win,

LCD_XMLList_Win,

LCD_BlackList_Win,

LCD_APList_Win,

LCD_MenuHistory_Win,

LCD_LocalHistory_Win,

LCD_AllCalls_Win,

LCD_MissedCalls_Win,

LCD_ReceiveCalls_Win,

LCD_DialCalls_Win,

LCD_ForWordCalls_Win,

LCD_CallLog_Win,

LCD_Message_Win,

LCD_VoiceMail_Win,

LCD_ViewVoiceMail_Win,

LCD_SetVoiceMail_Win,

LCD_SMS_Win,

LCD_ViewSMS_Win,

LCD_SetSMS_Win,

LCD_Settings_Win,

LCD_BasicSetting_Win,

LCD_Language_Win,

LCD_TimeDate_Win,

LCD_TimeFormat_Win,

LCD_DhcpTime_Win,

LCD_RingTone_Win,

LCD_HeadSet_Win,

LCD_Bluetooth_Win,

LCD_AdvancedSetting_Win,

LCD_AdvancedSetPass_Win,

LCD_AccountPass_Win,

LCD_AdvanedNetwork_Win,

LCD_WANPort_Win,

LCD_PCPort_Win,

LCD_Vlan_Win,

LCD_WebserverType_Win,

LCD_8021x_Win,

LCD_VPN_Win,

LCD_DhcpVlan_Win,

LCD_LLDP_Win,

LCD_PhoneSetting_Win,

LCD_Lock_Win,

LCD_SetPassword_Win,

LCD_AutoProvision_Win,

LCD_Display_Win,

LCD_DisplayMode_Win,

LCD_Wallpaper_Win,

LCD_Screensaver_Win,

LCD_OthersPass_Win,

LCD_OthersAPP_Win,

LCD_Factory_Win,

LCD_Restart_Win,

LCD_Reboot_Win,

LCD_PcapFeature_Win,

3.1.4 Auto_Test_Call.sh 相关配置

后续新建测试脚本需写在函数内，方便在 Auto_Test_Main.sh 主脚本中调用。格式可参考此脚本。

在关键状态点添加验证状态函数：

```
#基本呼叫
function Call_Function()
{

    for((j=0;j<1;j++))
    do
        #每执行一次，ScreenInitialValue值加1
        ScreenInitialValue=`expr $ScreenInitialValue + 1`
        #验证状态
        CheckState_Main $URL1 $CALLS_STATE_IDLE
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_IDLE" 10

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=SPEAKER"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OFFHOOK
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OFFHOOK" 10

        #拨号705
        Dial $URL1 $Num2

        wget --http-user=$USER --http-password=$PASS $URL1"keyboard=F1"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_OUT_GOING
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OUT_GOING" 20

        wget --http-user=$USER --http-password=$PASS $URL2"keyboard=OK"
        if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

        CheckState_Main $URL1 $CALLS_STATE_TALK
        Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_TALK" 30
    done
    ScreenInitialValue=0
    #获取内存信息
    Get_MemoryFree $URL1
}
```

后续脚本中，每条 wget 命令后，跟一条验证返回状态码的命令（以验证 wget 命令是否成功执行）：

```
wget --http-user=$USER --http-password=$PASS $URL1"keyboard=F1"
if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi

CheckState_Main $URL1 $CALLS_STATE_OUT_GOING
Screenshot $FUNCNAME $SCREENURL1 "CALLS_STATE_OUT_GOING" 2

wget --http-user=$USER --http-password=$PASS $URL3"keyboard=OK"
if [[ $? -ne 0 ]]; then echo "***************** Command Fail, Please Check.*****************"; exit 1; fi
```

3.2 Backup 文件夹

- Auto_Test_Syslog.xml

脚本执行完后，可查看 Backup 文件夹内的 Auto_Test_Syslog.xml，查询每条命令的执行状态。（主要查看 Verify 状态）

例：终端运行主脚本并带入参 CALL

./ Auto_Test_Main.sh CALL #执行 Call_function

若执行 1000 次呼叫，执行完后，查看 Auto_Test_Syslog.xml。可直接在日志文件中搜索关键字“Verify ERROR”，快速查看是否有验证状态错误。

若验证状态正确，会打印 Verify SUCCESS。

例：（Verify SUCCESS）

若验证状态有错误，会打印 Verify ERROR，且会打印出错误状态。再由日志中打印的前后执行命令，查看话机何时状态错误。

例：（Verify ERROR）

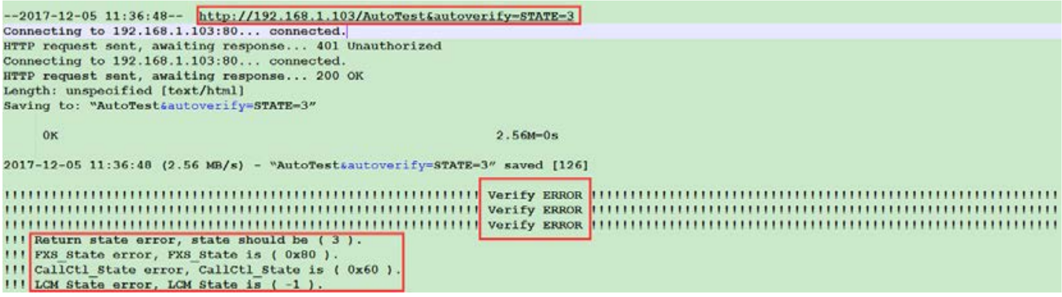
如下图，查看打印的 Verify ERROR 上一条命令为：

http:// 192.168.1.103/AutoTest&autoverify=STATE=3

对比 Auto_Test_Config.sh 中话机状态定义，可知 STATE=3 对应话机状态为：

CALLS_STATE_TALK=3　/* Sperker 基本通话态　*/

由下图打印可知，话机此时为空闲态，非通话态，状态错误。



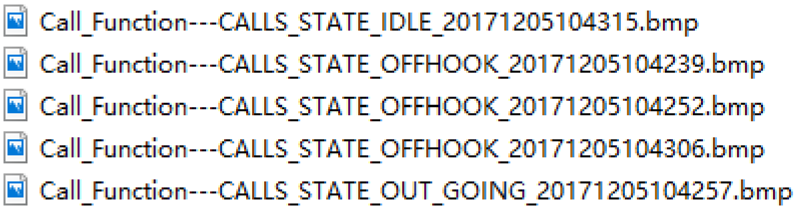
```
--2017-12-05 11:36:48--  http://192.168.1.103/AutoTest&autoverify=STATE=3
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 401 Unauthorized
Connecting to 192.168.1.103:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: "AutoTest&autoverify=STATE=3"

    OK                                                   2.56M=0s

2017-12-05 11:36:48 (2.56 MB/s) - "AutoTest&autoverify=STATE=3" saved [126]

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Verify ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Verify ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! Verify ERROR !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Return state error, state should be ( 3 ).
!!! FXS_State error, FXS_State is ( 0x80 ).
!!! CallCtl_State error, CallCtl_State is ( 0x60 ).
!!! LCM State error, LCM State is ( -1 ).
```

◆ 截图 bmp 文件

若 Auto_Test_Syslog.xml 文件正确，则查看截图文件，以验证话机屏幕显示状态是否正确。

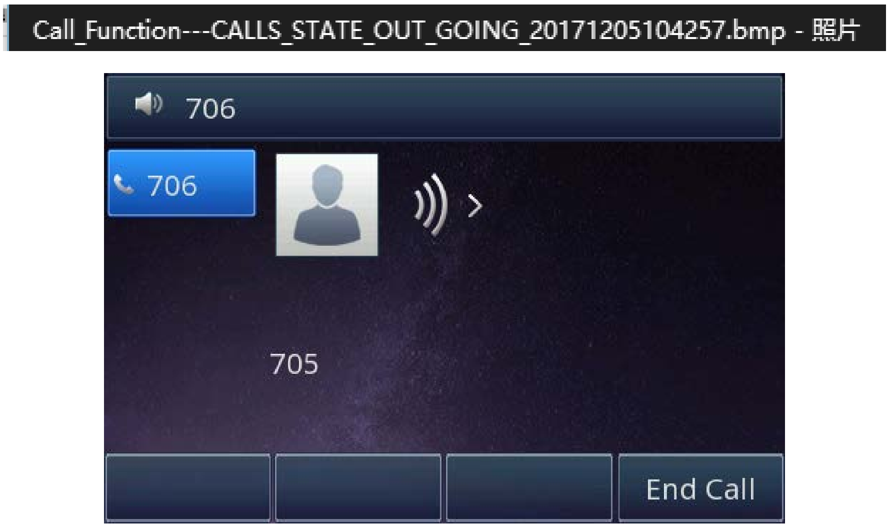
截图 bmp 文件存放目录为：\Auto_Test\Backup\Screenshot

命名规则为：测试项目---当前状态_截图时间.bmp



Call_Function---CALLS_STATE_IDLE_20171205104315.bmp
Call_Function---CALLS_STATE_OFFHOOK_20171205104239.bmp
Call_Function---CALLS_STATE_OFFHOOK_20171205104252.bmp
Call_Function---CALLS_STATE_OFFHOOK_20171205104306.bmp
Call_Function---CALLS_STATE_OUT_GOING_20171205104257.bmp

将当前状态与截图对比查看是否正确，如下为呼出态：



Call_Function---CALLS_STATE_OUT_GOING_20171205104257.bmp - 照片

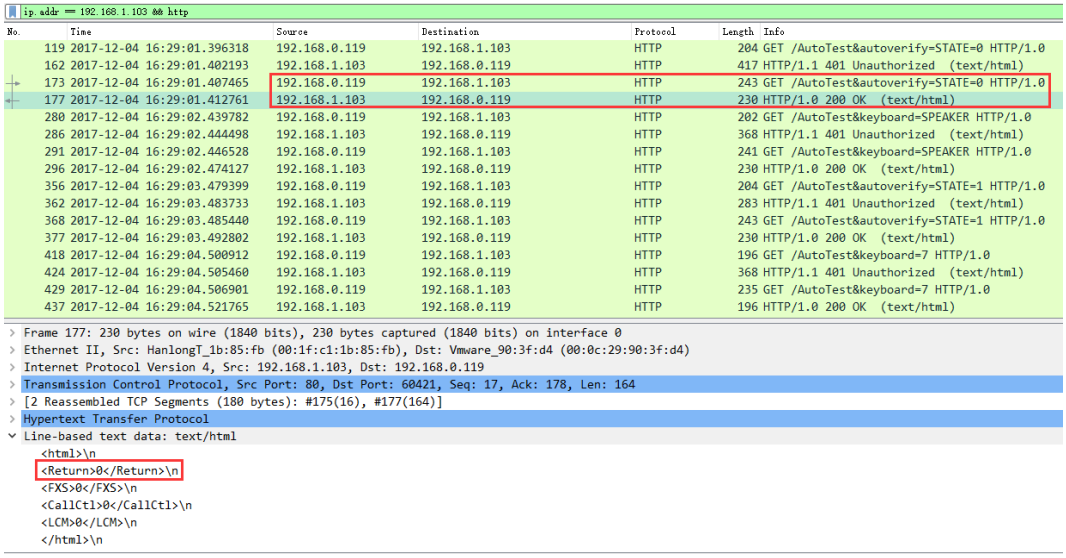
4.WireShark 抓包验证

抓取 http 包，也可查看脚本流程及验证状态信息。

查看话机对 STATE 回复的 200 OK。

- Return = 0

  验证话机状态 FXS、CallCtl、LCM 均正确。



- Return = 1

  话机状态 FXS、CallCtl、LCM 中有错误状态。

  若状态错误，则对应返回值为 1，且会返回此时话机状态。