**脚本回放工具**

**环境说明**

1. jre1.7以上
2. adb环境
3. monkeyrunner环境

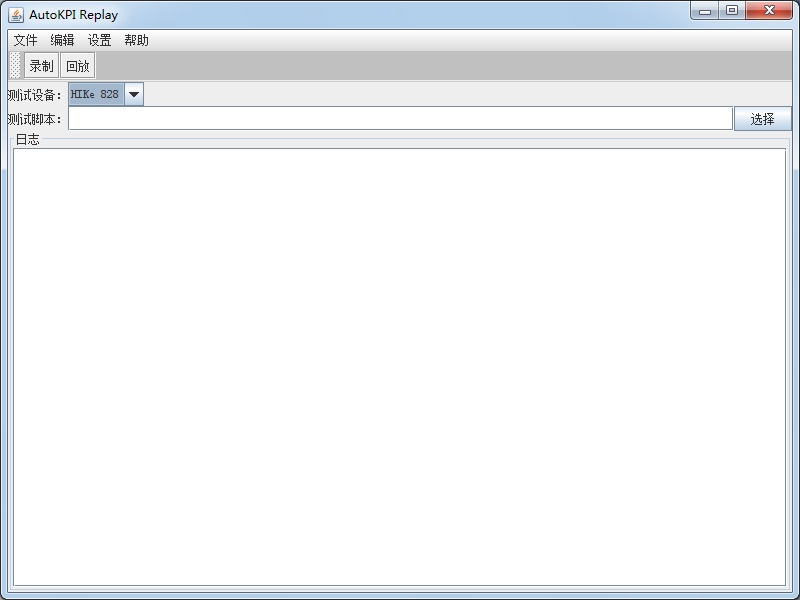
**安装说明**

1. 下载工具jar包存放在指定目录:

**使用说明**

**打开AutoKPI Replay**

通过java打开AutoKPI Replay

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Replay_001.jpg)

**录制脚本**

点击工具栏中的"录制"打开脚本录制工具

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Recorder_001.jpg)

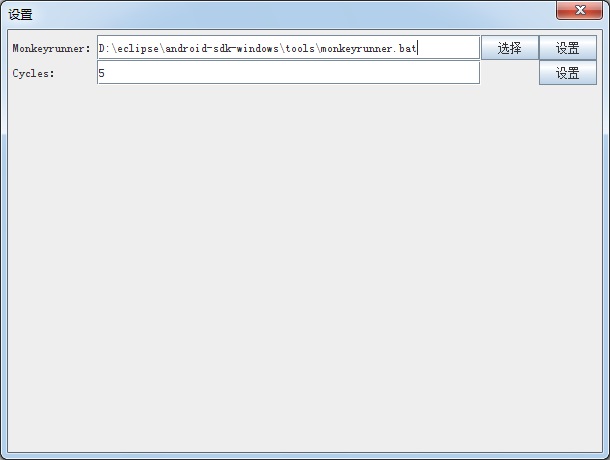
脚本录制工具的使用方法请参考脚本录制工具使用说明

注意：

1、脚本录制工具必须和脚本回放工具放在相同目录。

2、录制脚本时，PC有且只能连接一台手机。

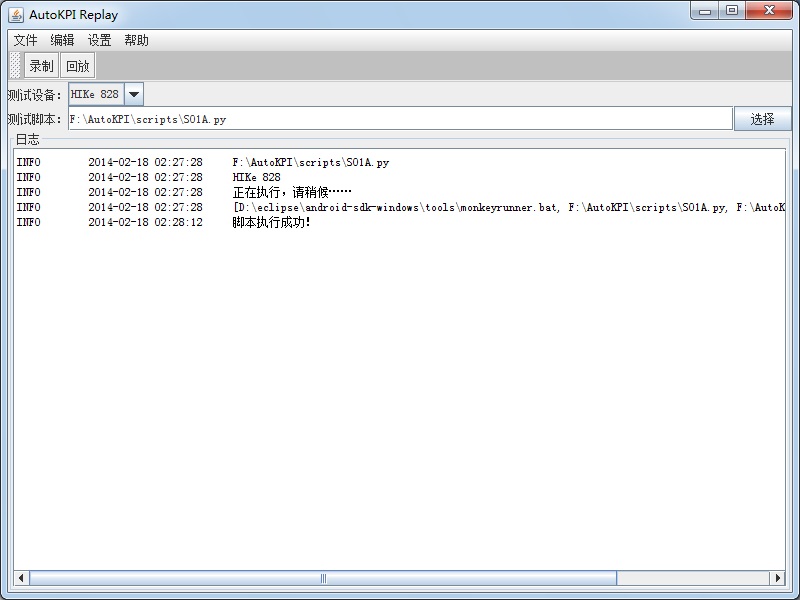
**设置**

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Replay_setting_001.jpg)

工具会通过"ANDROID\_HOME"环境变量自动搜索Monkeyrunner.bak并设置，如果没有，请手动设置。

设置完成后，设置信息将保持在当前目录config文件夹下的setting.Properties文件中

**脚本回放**

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Replay_002.jpg)

选择录制的脚本后，点击工具栏中的"回放"

回放完成后，日志将显示在日志栏中。

回放完成后，mark标记的时间戳存放在当前目录下result\测试用例名\循环次数\timestamp.txt文件中

**清空日志**

点击"编辑"->"清空日志"，清除日志栏中的日志信息

## 视频录制工具

### 环境说明

1. jre1.7以上

### 安装说明

1. 下载 AutoKPI Video.jar

### 使用说明

#### 打开AutoKPI Video

通过java打开AutoKPI Video

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_001.jpg)

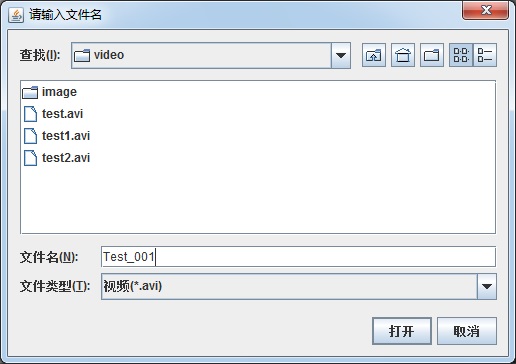
注意：摄像头需要在打开工具之前连接好

#### 手动录制视频

##### 开始录制

1、点击录制按钮

2、选择视频存放路径，默认视频存放在“video”目录下。

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_002.jpg)

##### 停止录制

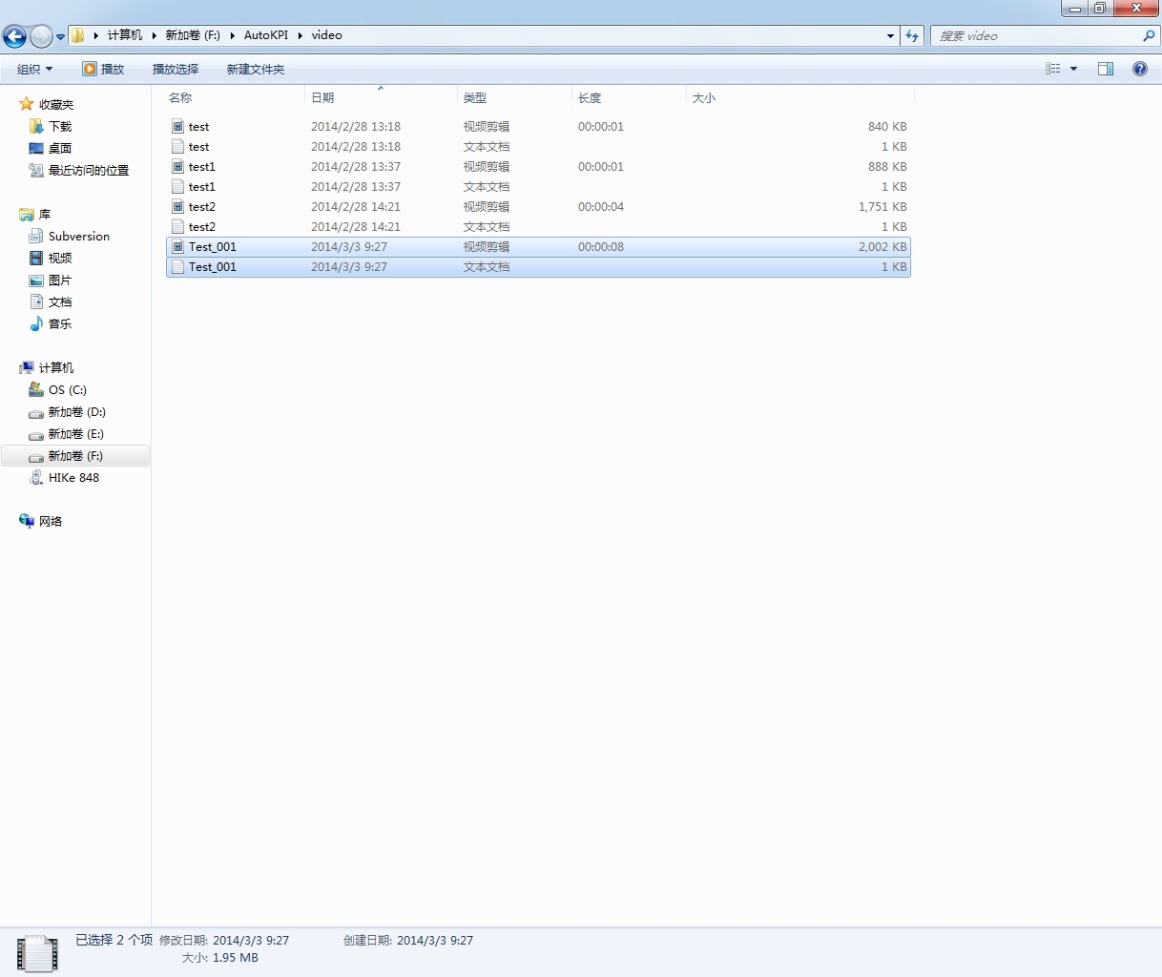
1、点击停止录制。

2、录制完成后，界面上显示录制开始时间（纳秒）、结束时间（纳秒）、视频时长。

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_003.jpg)

3、在视频存放文件夹下，生成与视频同名的txt文件。

4、在txt文件中保存视频录制开始时间（纳秒）、结束时间（纳秒）、总帧数、视频时长（毫秒）

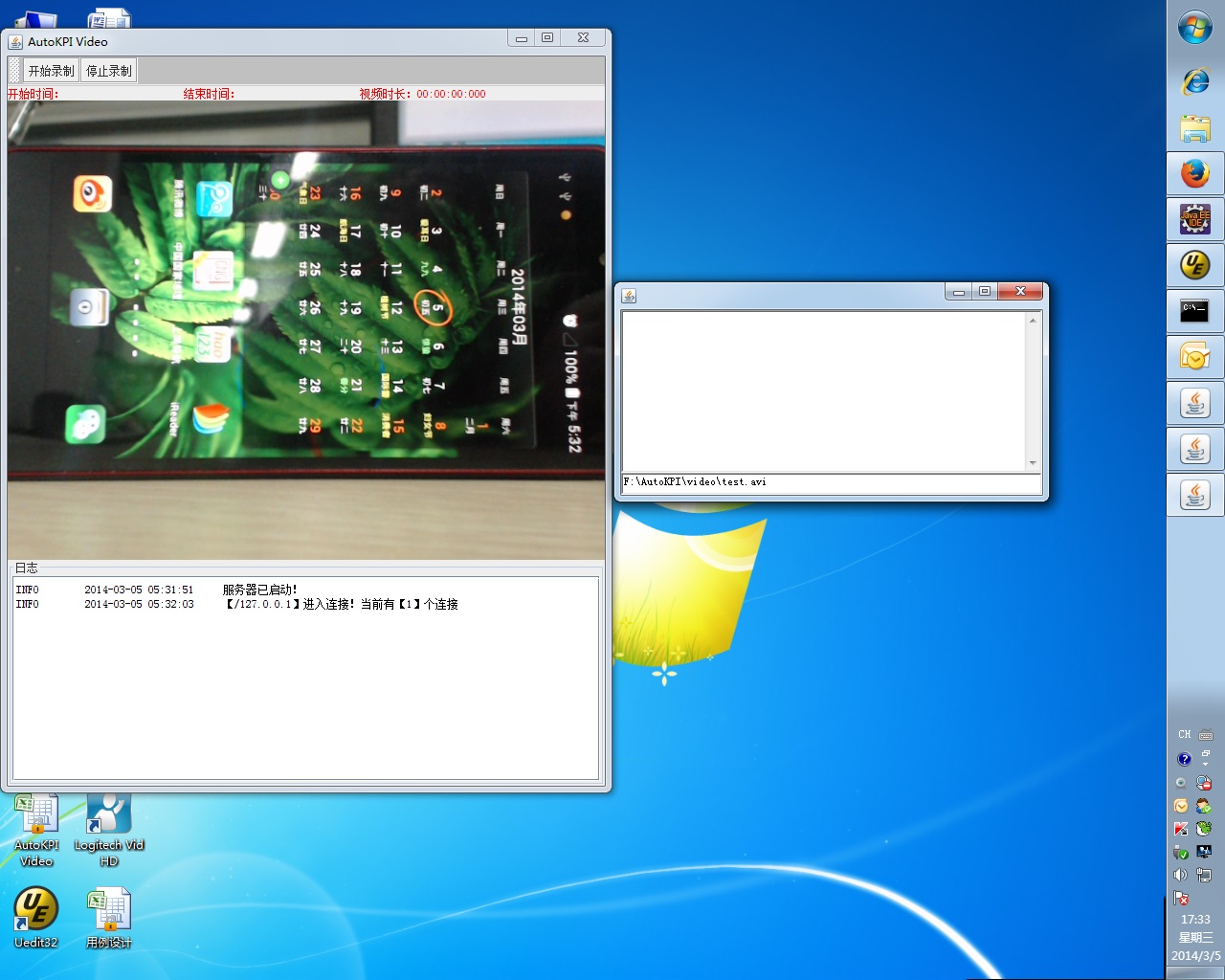
[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_004.jpg)

#### 远程控制录制视频

##### 客户端控制录制视频

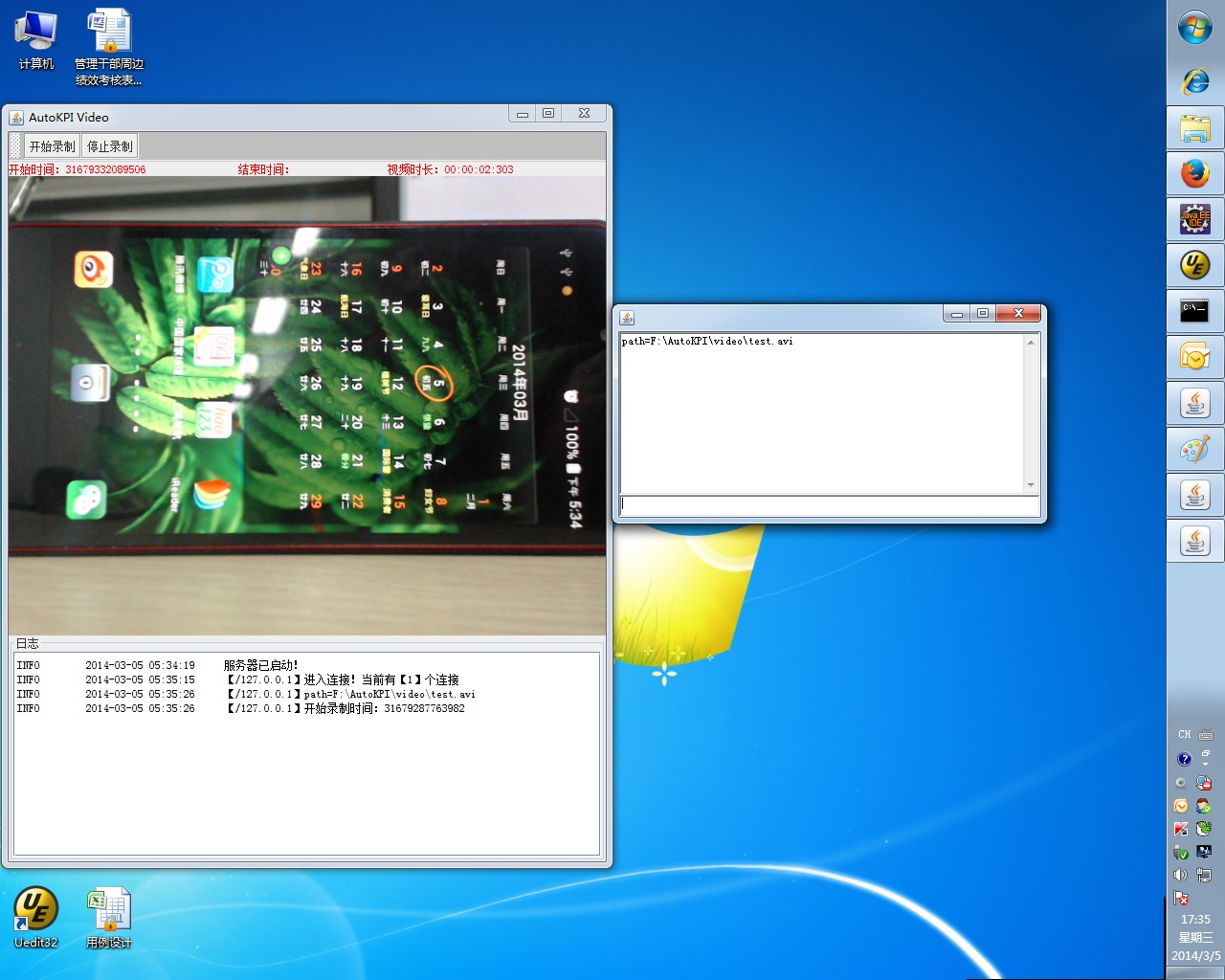
1、下载 AutoKPI Video Client.jar

2、打开视频录制工具和控制工具

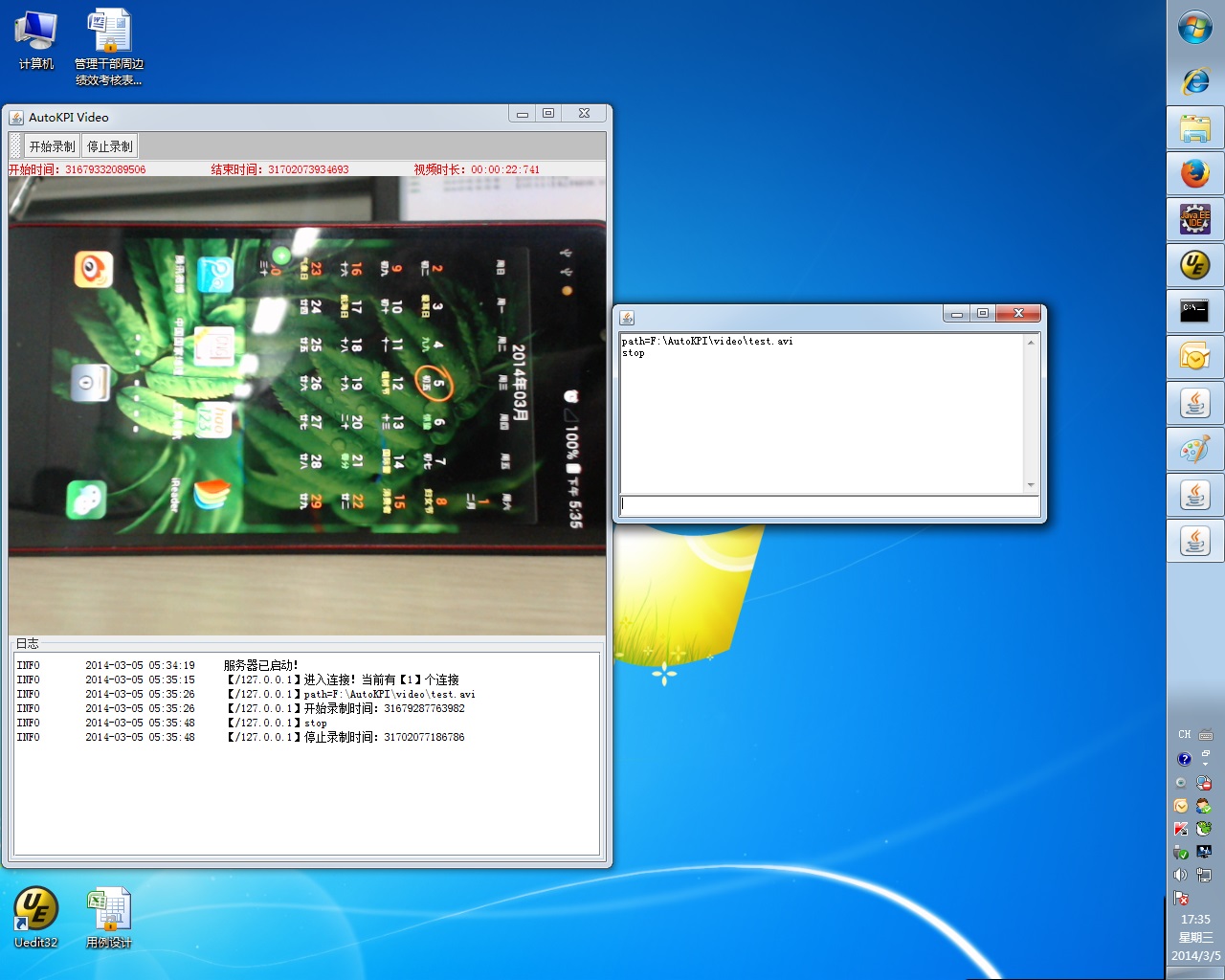
[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_005.jpg)

3、远程控制客户端发送"path=视频录制保存路径"，工具开始录制。

例如：发送"path=F:\AutoKPI\video\test.avi"

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_006.jpg)

4、远程控制客户端发送“stop”，工具停止录制。

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_007.jpg)

##### MR脚本控制录制视频

1、下载 AutoKPI Video Client.jar

2、将下载的远程录制视频包存放到%ANDROID\_HOME%\tools\lib目录下

3、在MR脚本中引入：

from com.ckt.autokpi.video.socket import VideoClient as vc

4、获得视频录制对象(默认端口为9999)：

sc = vc.getSocketClient("127.0.0.1",9999);

5、发送开始录制视频命令：

sc.write("path=F:\\AutoKPI\\video\\test.avi")

6、发送停止录制视频命令：

sc.write("stop")

7、关闭视频录制对象：

sc.close()

实例：

#!/usr/bin/python

#coding=UTF-8

import sys

import os

import time

from com.android.monkeyrunner.easy import EasyMonkeyDevice

from com.android.monkeyrunner.easy import By

from com.android.monkeyrunner import MonkeyRunner as mr

from com.android.monkeyrunner import MonkeyDevice as md

from com.android.monkeyrunner import MonkeyImage as mi

from com.ckt.autokpi.video.socket import VideoClient as vc

from java.lang import System

projectPath = str(sys.argv[1])

deviceName = str(sys.argv[2])

count = str(sys.argv[3])

# python script name

def getscname():

osname = os.path.basename(\_\_file\_\_)

n = len(osname)

return osname[0:n-3]

def mark():

mname = str(getscname())

fname = 'timestamp.txt'

cpath = projectPath+'\\result\\'+mname+'\\'+str(count)

if not os.path.exists(cpath):

os.makedirs(cpath)

pname = cpath+'\\'+fname

if os.path.exists(pname):

file\_object = open(pname, 'w')

else:

file\_object = file(pname, 'w')

tmp = str(System.nanoTime())

file\_object.write(tmp)

file\_object.close()

device = mr.waitForConnection(3,deviceName)

sc = vc.getSocketClient("127.0.0.1",9999);

sc.write("path=F:\\AutoKPI\\video\\test.avi")

mark()

device.touch(550,1748,md.DOWN\_AND\_UP)

device.press('KEYCODE\_BACK', md.DOWN\_AND\_UP)

mr.sleep(5.0)

sc.write("stop")

sc.close()

#### 摄像头重连

摄像头未连接、摄像头异常断开时，点击重新连接，重新连接摄像头。

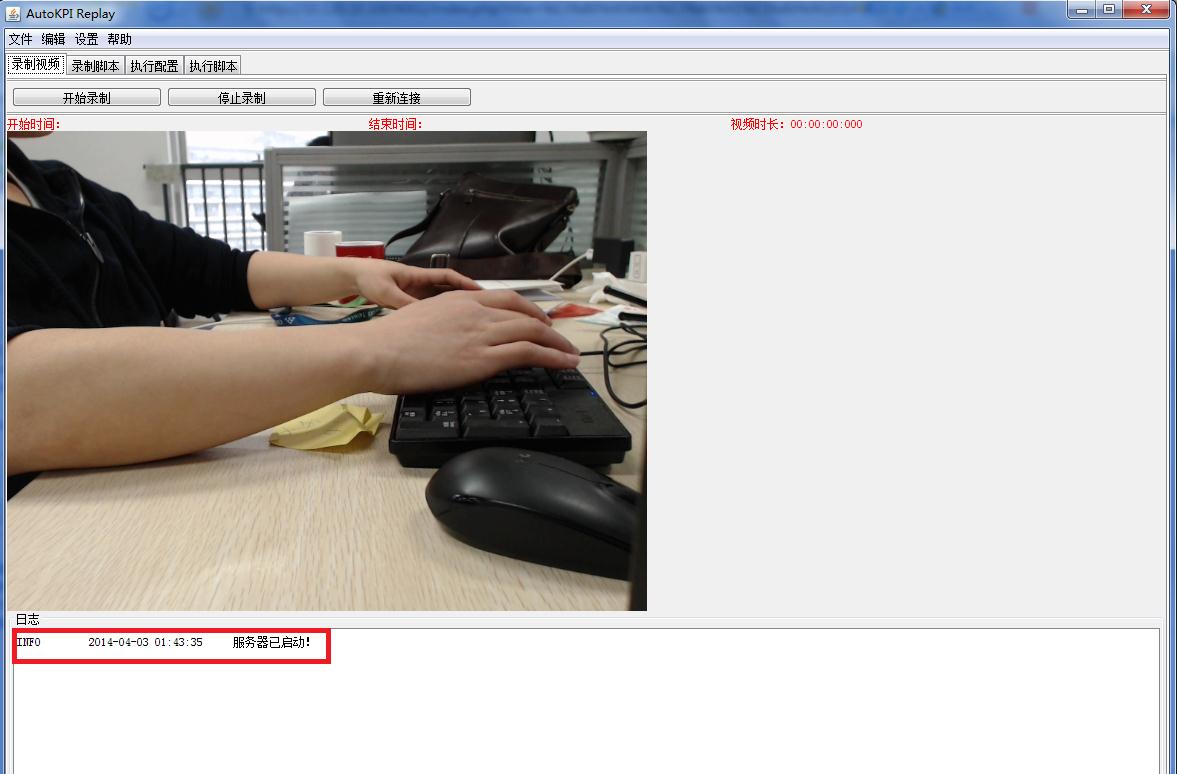
[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:AutoKPI_Video_008.jpg)

# 环境配置

1：安装jdk1.7

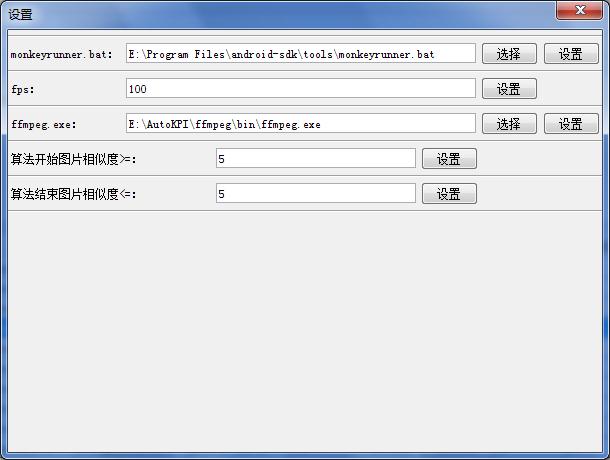
2:连接摄像头，安装摄像头驱动

3:执行AutoKPI.jar,出现图中红色字体表明摄像头连接成功！

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T001.jpg)

4:设置monkeyrunner路径

5:设置ffmpeg.exe路径

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T005.jpg)

# 脚本录制

连接手机,启动monkeyrecorder 录制脚本.

这里需要注意：

1：录制某一个动作之前需要点击VIDEO\_START

2：录制完成之后需要点击VIDEO\_STOP

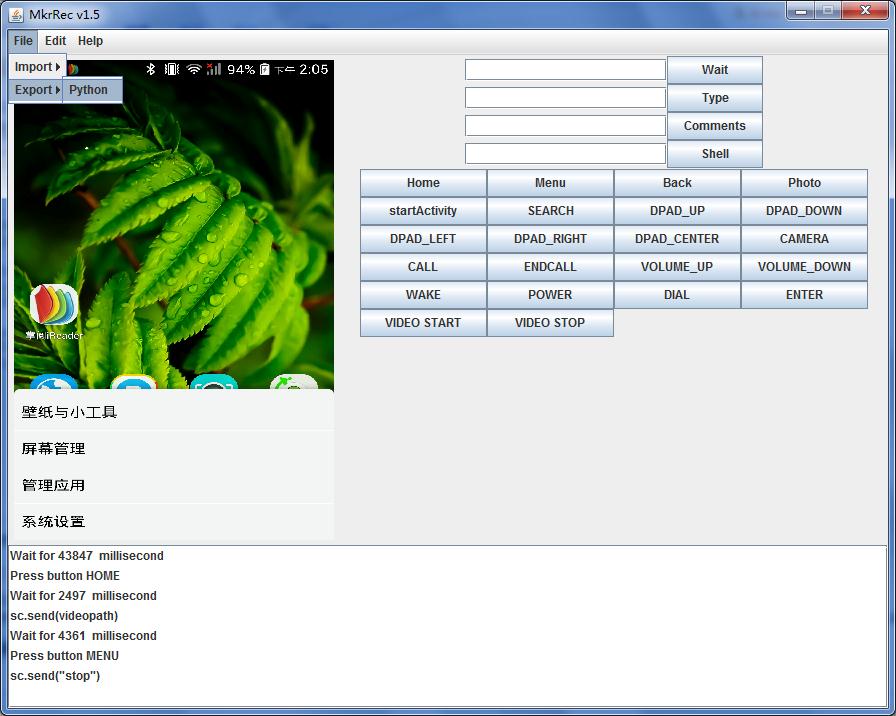
这样录制的视频就是所需响应时间的视频文件

比如我们要测试MENU 键的响应时间,如图所示

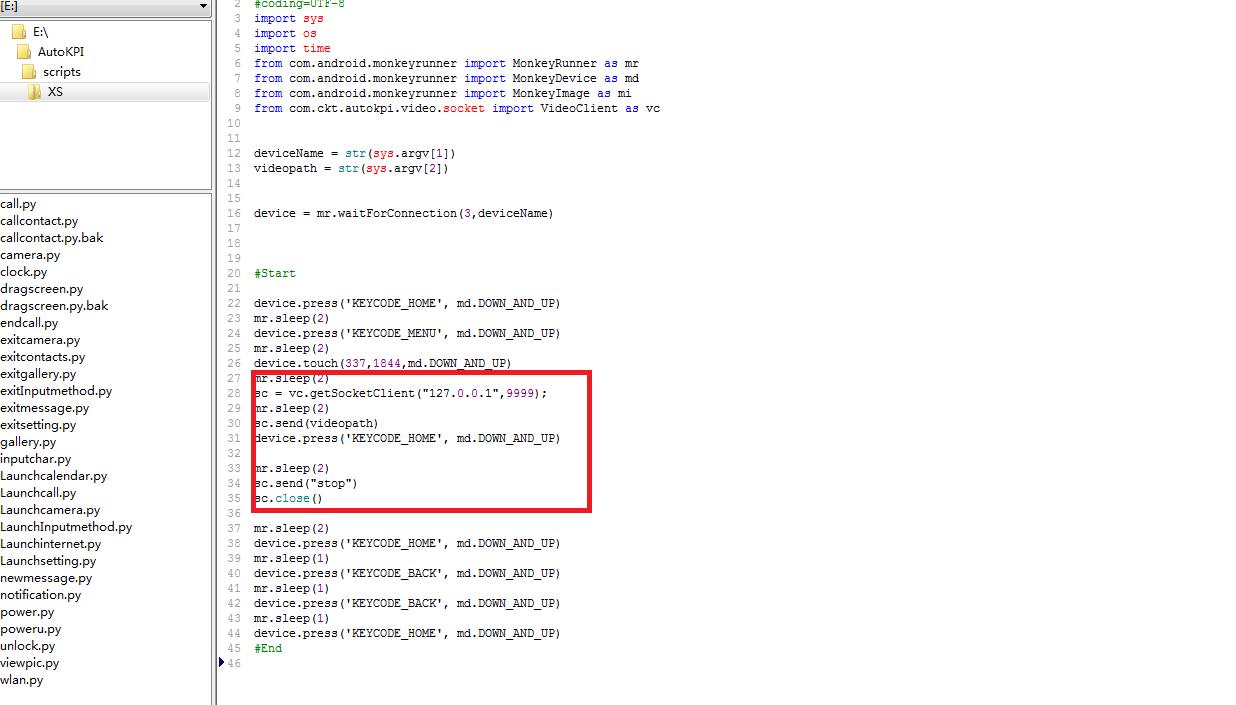
点击录制脚本->

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T002.jpg)

导出为python文件

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T003.jpg)

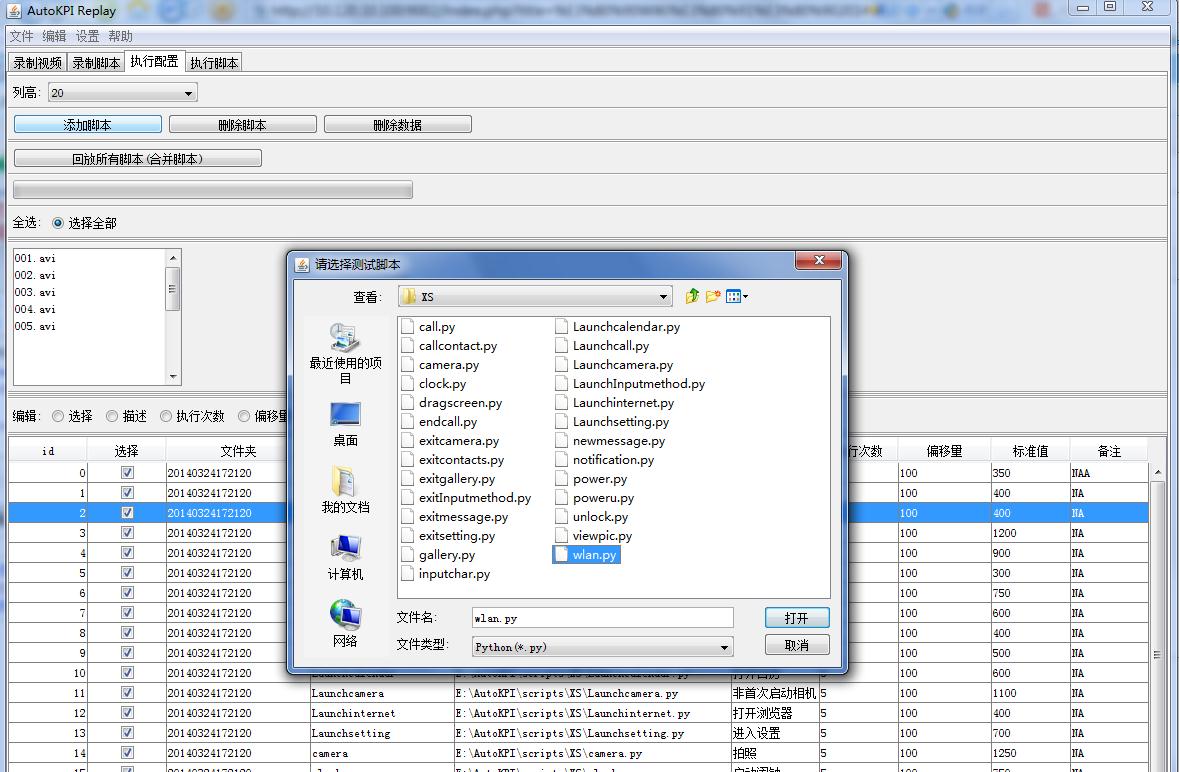
手动修改脚本里边特别大的sleep时间，红色方框里边的代码不能修改

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T004.jpg)

# 添加脚本以及配置脚本

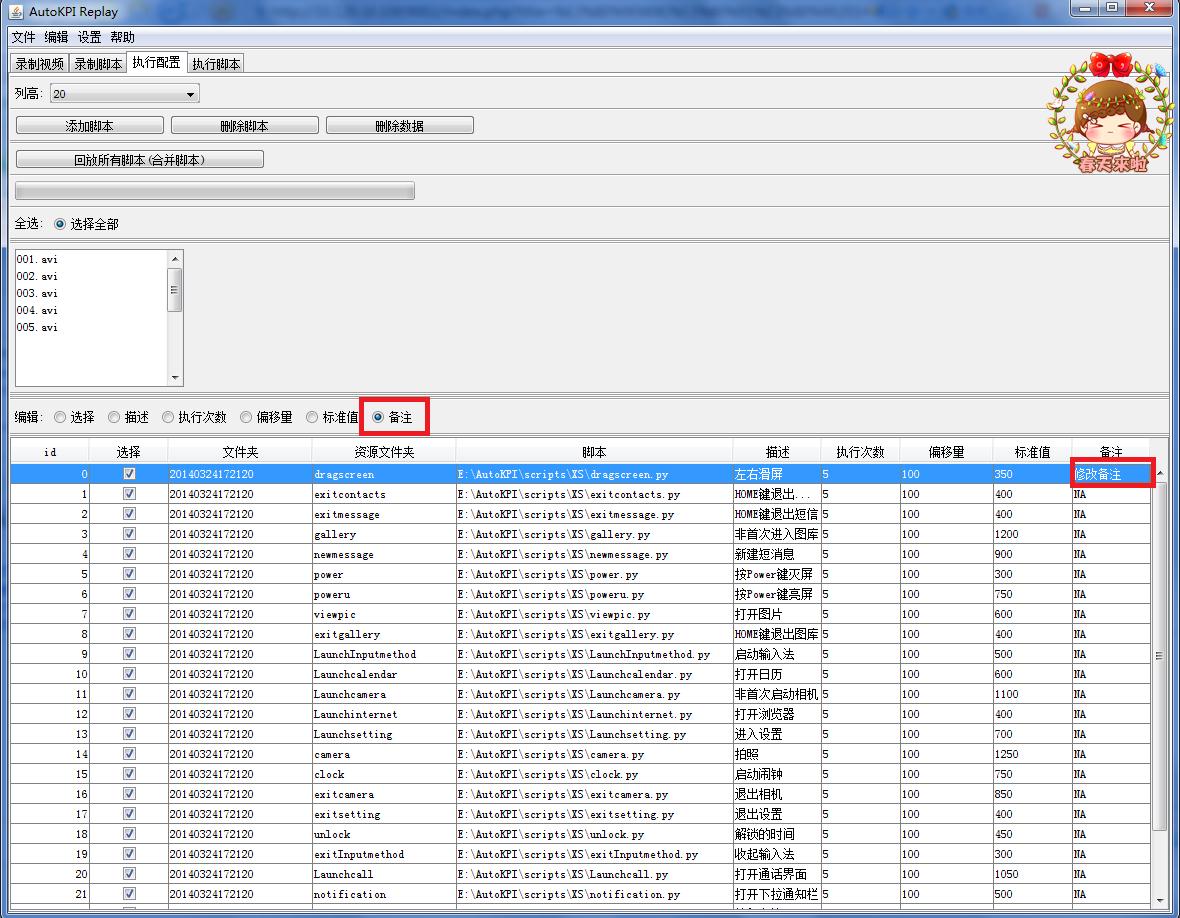
1:点击执行配置->添加脚本

选择录制的python脚本

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T006.jpg)

2:配置参数修改

enable jcheckbox 双击table中的单元格修改相应的属性值，然后回车保存

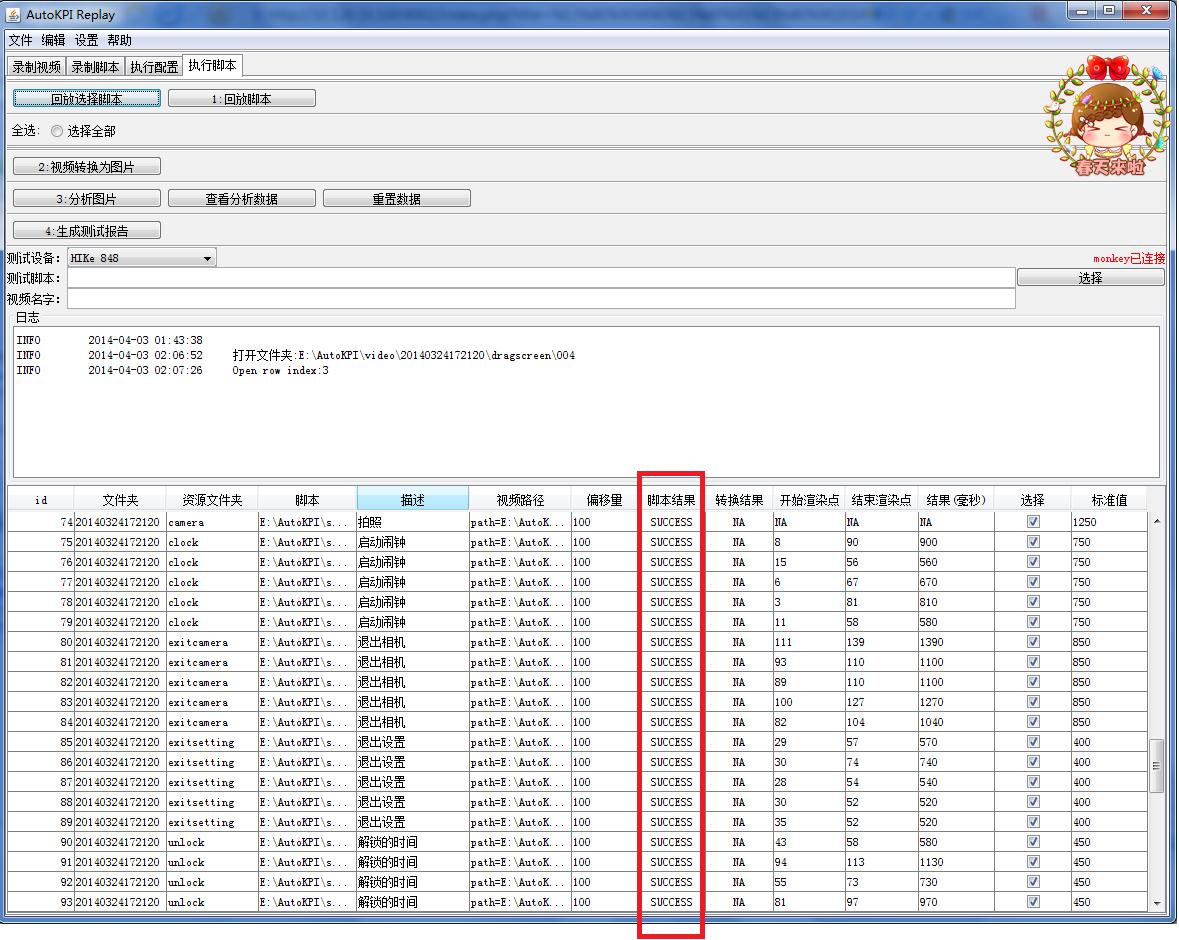
[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T007.jpg)

# 执行脚本

1:点击 执行配置->回放所有脚本

2:切换到 执行脚本,可以看见执行情况，执行结果为SUCCESS 表明执行成功

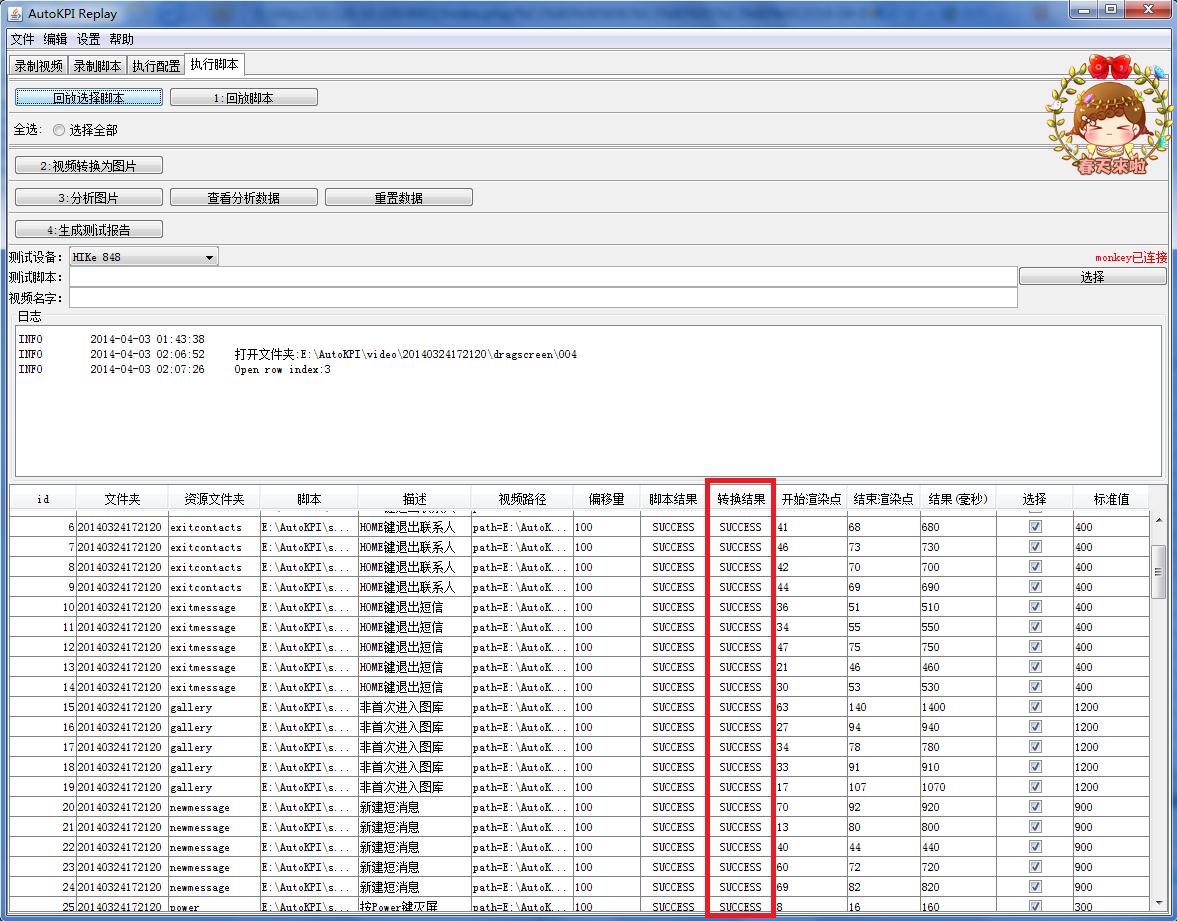
执行完成之后会有提示对话框

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T008.jpg)

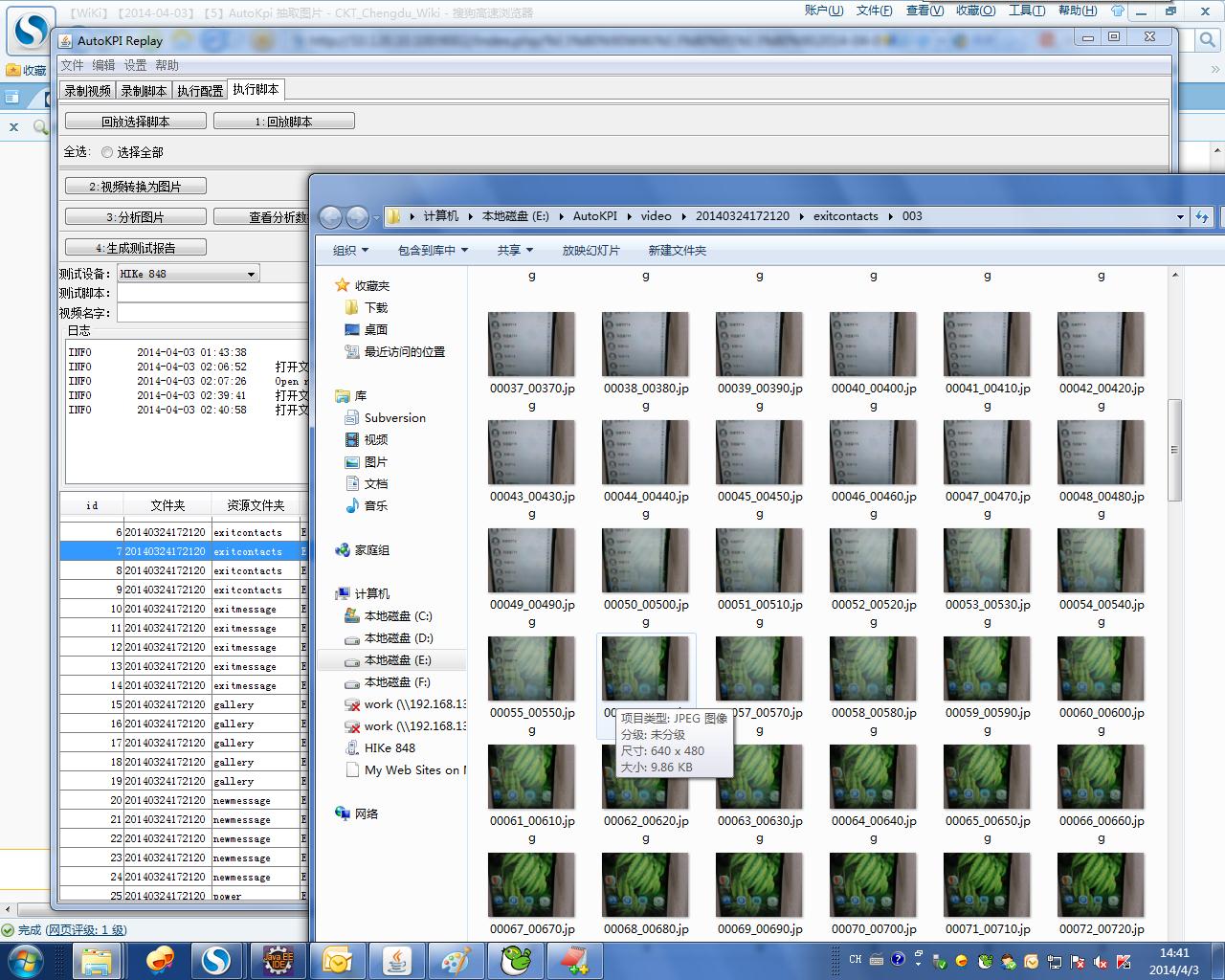
# 抽取图片

切换到 执行脚本->2视频转换为图片,可以看见执行情况，执行结果为SUCCESS 表明转换成功

采取一秒100张图片,及间隔10ms抽取一张图片

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T009.jpg)

双击某一行进入抽取的图片文件夹

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T011.jpg)

# 分析脚本

1：切换到 执行脚本->3分析图片,耐心等待分析一个脚本大约需要3分钟！

2：图片分析原理介绍

下面是一个最简单的实现：

第一步，缩小尺寸。

将图片缩小到8x8的尺寸，总共64个像素。这一步的作用是去除图片的细节，只保留结构、明暗等基本信息，摒弃不同尺寸、比例带来的图片差异。

用汉明距离进行图片相似度检测的Java实现 用汉明距离进行图片相似度检测的Java实现

第二步，简化色彩。

将缩小后的图片，转为64级灰度。也就是说，所有像素点总共只有64种颜色。

第三步，计算平均值。

计算所有64个像素的灰度平均值。

第四步，比较像素的灰度。

将每个像素的灰度，与平均值进行比较。大于或等于平均值，记为1；小于平均值，记为0。

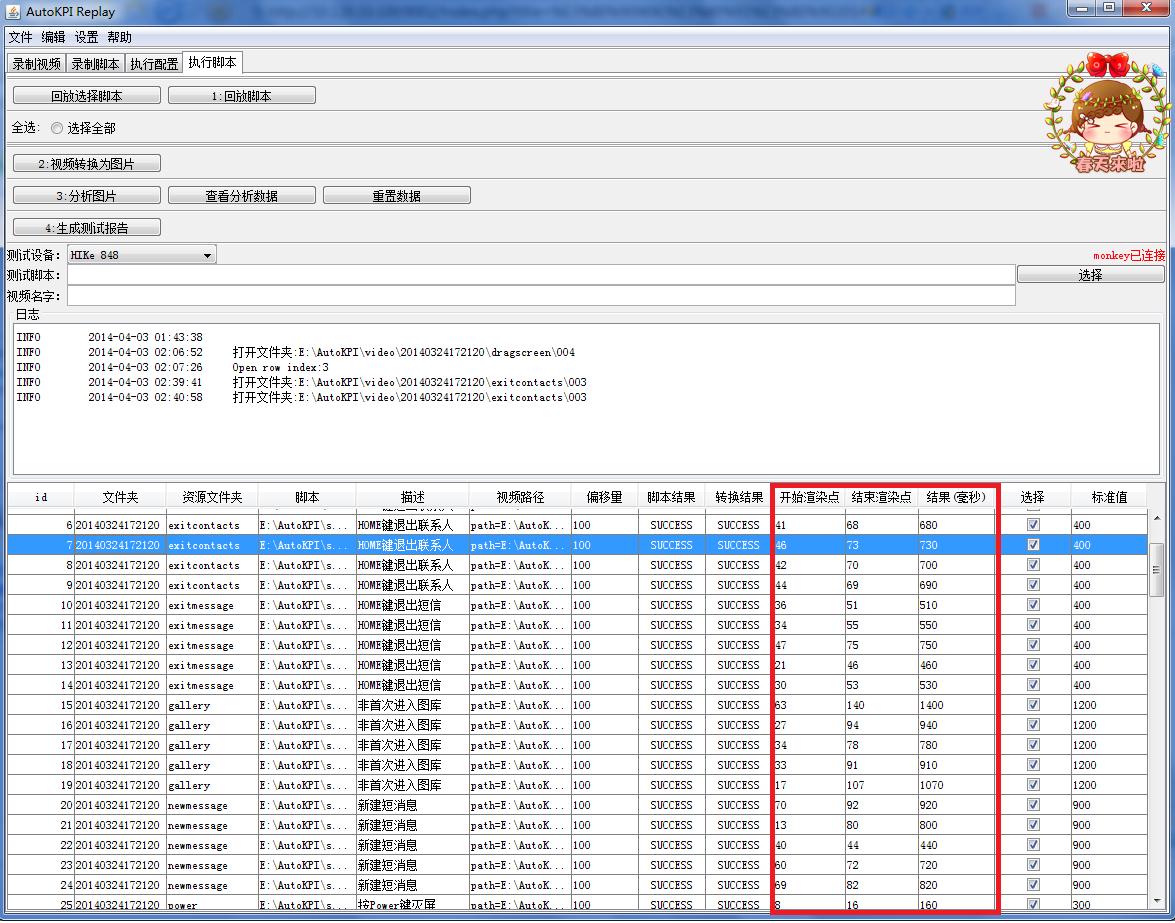
第五步，计算哈希值。

将上一步的比较结果，组合在一起，就构成了一个64位的整数，这就是这张图片的指纹。组合的次序并不重要，只要保证所有图片都采用同样次序就行了。

用汉明距离进行图片相似度检测的Java实现 = 用汉明距离进行图片相似度检测的Java实现 = 8f373714acfcf4d0

得到指纹以后，就可以对比不同的图片，看看64位中有多少位是不一样的。在理论上，这等同于计算"汉明距离"（Hamming distance）。

如果不相同的数据位不超过5，就说明两张图片很相似；如果大于10，就说明这是两张不同的图片。

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T012.jpg)

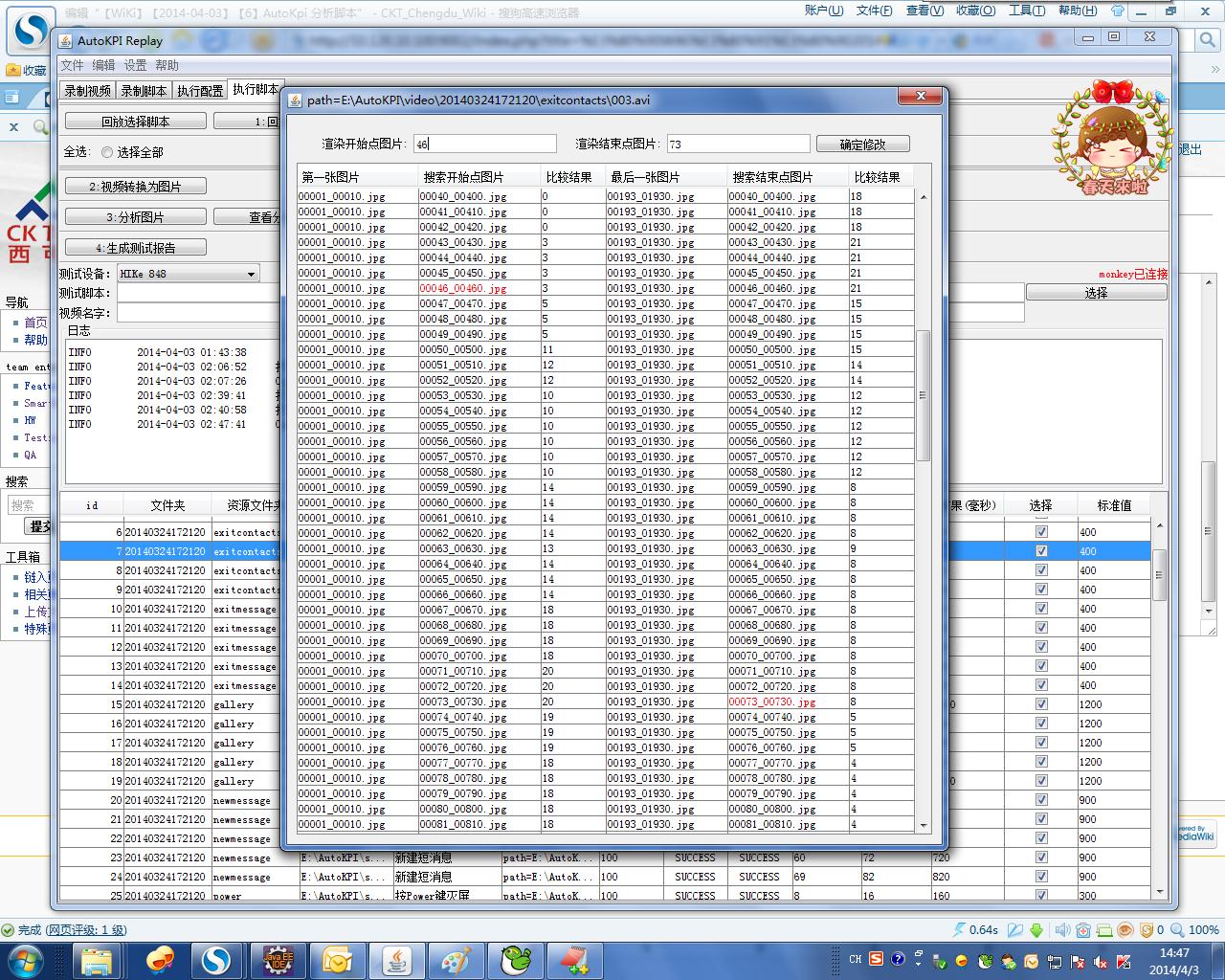
3:查看分析的数据，对比的HASH值

选中一行->按下空格键

开始渲染点：界面开始变化的时刻

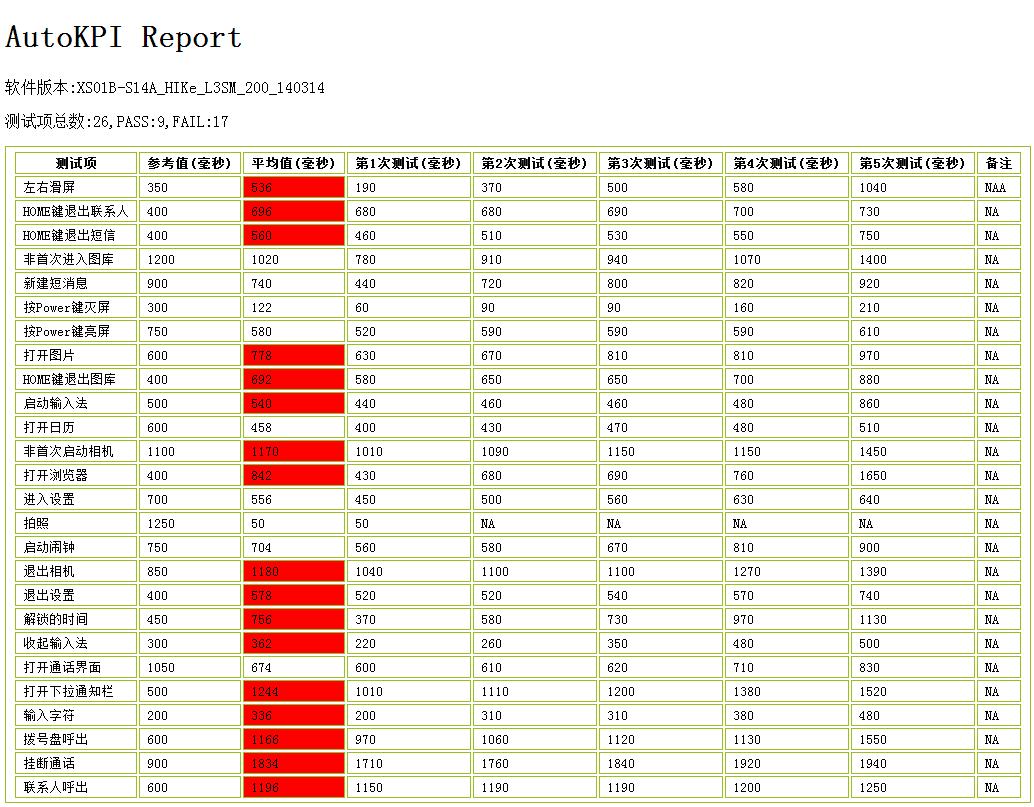
结束渲染点：界面开始结束的时刻

响应时间等于结束渲染点图片对应的那个时刻

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T013.jpg)

# 测试结果

点击 4生成测试报告

[](http://10.120.10.100:9002/%E6%96%87%E4%BB%B6:T014.jpg)