# Automation scripts

# Automated playback of utterances

semi_manual_test.sh

```
#!/bin/bash
adb devices -l
cycle=0

echo "starting barge in"

echo "input serial number"
read DSN
echo "Input volume"
read volume
echo "Input device type"
read device_type

sleep 2
adb -s $DSN logcat -d > device_log-$NOW.log & #collect device logs in background


for iteration in {1..100} #100 iterations
do
  cycle=$((cycle + 1))
  NOW=$(TZ="UTC" date +%F_%H:%M:%S)
  echo "Time: " $NOW
  echo "playing cycle: " $cycle
  echo ""
  echo `play /home/sanitized_username/Desktop/US/Utterance_A.wav` #UTTERANCE to start playback
  echo $NOW
  sleep 15
  echo `play /home/sanitized_username/Desktop/US/Utterance_B.wav` #Test utterance
  echo $NOW
  sleep 10
done

echo `play /home/deviceqa/Desktop/US/STOP_Utterance.wav` #stop playback

echo "Finished test"
```

# Automated rotation of flashing devices

group_flashing.sh

```
#!/bin/bash

declare -a device_type_A=("G0914XXXXXXXNH" "G091XXXXXX0EV" "G09XXXXXX3008L") #sanitized device serial numbers of devices
declare -a device_type_B=("G09XXXXXXXXXX97006K" "G091XXXXX006J")


adb devices -l
echo "Type of of the device types"
read device_type
echo “Please enter url: ” #download link from the build repository website
read url
echo "Going to ~/tmp/" #temporary folder
cd ~/tmp/
rm -rf ~/tmp/* # purging old files
echo `wget --auth-no-challenge --user=username --ask-password -c $url`
echo `tar xvzf *.tgz`
cd release-*
if [ $device_type == 'device_type_A' ] #determine device type
    then
        devices="${device_type_A[@]}" #this syntax formatting allows to return all elements of the array and assign to the variable "devices"
elif [ $device_type = 'device_type_B' ]
    then
        devices="${device_type_B[@]}"
else
    echo "incorrect device type"
fi


for DSN in $devices
  do
    sudo ./flashimage.py --aserial "$DSN" --fserial "$DSN" --toolsdir $(dirname $(which adb))
    echo "Flashing done for "$DSN""
  done
```