

中国·珠海 全志科技股份有限公司 All Winner Tech. CO. LTD	文档编号		SW1112TMP000				
	文档名称		Adb 使用说明				
	版    本	V0.1	密    级	1	共 1 页		
<div>Melis v459 Adb 使用说明</div>							
( 内部公开)							
文档作者	wuhuating			创建日期	2020-1-17		
拟    制	wuhuating			日    期	2020-1-17		
审    核				日    期			
批    准				日    期			
分发部门							
[ ]SW	[ ]SD	[ ]AL	[ ]CS	[ ]TST	[ ]MKT		



All Winner Technology

CopyRight©2020 All Winner Technology, All Right Reserved



## 版本历史

	修改人	时间	备注
V1.0	吴华庭	2020-1-17	建立初始版本





## 1. 概述

### 1.1. 编写目的

介绍 adb 的使用。

### 1.2. 适用范围

Melis v459 方案。

## 2. Adb 的使用

### 2.1. 简介

adb(Android Debug Bridge)是 Android SDK 中一个通用的调试工具。我们可以通过这个工具管理移动设备的状态。adb 其实是一个客户端-服务器端程序，其中客户端是你用来操作的电脑，服务器端是 android 设备(小机端)。adb 分为三部分:一部分是 adb server ，一部分是 adb command，这两部分运行在电脑主机上;另外一部分是 adbd(adbd daemon)，运行在设备端。

Adb 可以进行的操作包括：

1. 运行设备的 shell（命令行）。
2. push 文件到小机端或 pull 小机端的文件到 PC 端。

### 2.2. 使用步骤

开启小机端的 adb 服务之前，先确保 PC 端是否安装了 adb 工具。对于 Ubuntu 系统，可以在终端执行 adb 查看是否安装了 adb，若没安装则执行 `sudo adb-get install adb` 来安装。

Windows 系统的安装方法请移步百度。

```
wuhuating@PCwuhuating:~$ adb
Android Debug Bridge version 1.0.32
Revision debian
```

连接串口，在串口控制台执行 adbd 命令开启 adbd，然后才能使用 adb。

```
[DBG]: [awos_init_thread:0195]: System bringup finished! ByeBye!
msh />adbd
adbd version:AW-V1.1.0, compiled on: Jan 17 2020 10:07:43
adbd service init successful
msh />
CTRL-A Z for help | 115200 8N1 | NOR | Minicon 2.7 | VT102 | Online 1:27 | ttyUSB0
```

### 3. Adb 常用命令

adbd 启动之后就可以在 PC 终端使用 adb 命令了。

#### 3.1. adb devices

adb devices 用于查看当前所连接的设备号。

如果没连接小机端，执行 adb devices 则不会发现设备。

```
wuhuating@PCwuhuating:~$ adb devices
List of devices attached
```

通过 usb 线连接小机端，再执行 adb devices，就可以查看当前所连接的设备了（前提是小机端的 adbd 已开启，2.2 章节中有 adbd 启动说明）。

```
wuhuating@PCwuhuating:~$ adb devices
List of devices attached
c22fffffffff00000000    device
wuhuating@PCwuhuating:~$
```

#### 3.2. adb push

通过 adb push，可以把 pc 端的文件推送到小机端。

```
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push hehe.c /data
1 KB/s (437 bytes in 0.283s)
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push one.c /data
2 KB/s (786 bytes in 0.331s)
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push rand.c /data
2 KB/s (697 bytes in 0.300s)
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push urandom.c /data
2 KB/s (728 bytes in 0.292s)
```

可以看到，文件已 push 到小机端。

```
msh /data>ls
.gitignore apps hehe.c one.c rand.c
res urandom.c
msh /data>
CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Online 2:57 | ttyL
```

### 3.3. adb pull

通过 adb pull，可以把小机端的文件拉取到 PC 端。

```
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ ll
总用量 8
drwxrwxr-x 2 wuhuating wuhuating 4096 1月 17 11:49 ./
drwxr-xr-x 9 wuhuating wuhuating 4096 1月 17 11:49 ../
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ adb pull /data/hehe.c
4 KB/s (437 bytes in 0.088s)
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ adb pull /data/one.c
8 KB/s (786 bytes in 0.089s)
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ adb pull /data/rand.c
7 KB/s (697 bytes in 0.090s)
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ adb pull /data/urandom.c
7 KB/s (728 bytes in 0.091s)
wuhuating@PCwuhuating:~/Desktop/adb_testpull$ ls
hehe.c one.c rand.c urandom.c
```

### 3.4. adb shell

adb shell 用于开启终端进行调试。退出终端可以执行 exit 。

```
wuhuating@PCwuhuating:~$ adb shell
msh />
msh />exit
msh />wuhuating@PCwuhuating:~$
```

adb shell <command> 用于执行小机端的命令，可以使用 adb shell help 列出小机端支持的所有命令。

```
wuhuating@PCwuhuating:~$ adb shell hello
hello rt-thread!
wuhuating@PCwuhuating:~$ adb shell list_thread
-----
      pri  status      sp      stack size max used left tick error
-----
adbd-shell-ser-6      21  ready  0x00000048  0x00002000      00%  0x00000005  000
list_thread          21  ready  0x00000130  0x00002000     16%  0x00000004  000
adbd-output          20  suspend 0x00000148  0x00002000     11%  0x00000005  000
adbd-input           20  suspend 0x00000128  0x00002000     12%  0x00000004  000
tshell               20  suspend 0x000001e0  0x00001000     26%  0x0000000a  000
ksrv_task             6  suspend 0x000000f0  0x00001000     05%  0x0000000a  000
mnt_task              7  suspend 0x000000c8  0x00004000     01%  0x0000000a  000
tidle                 31  ready  0x00000060  0x00000400     16%  0x00000001  000
timer                  8  suspend 0x00000084  0x00002000     01%  0x0000000a  000
wuhuating@PCwuhuating:~$
```



## 4. 常见问题

### 4.1. 设备权限不足

adb push 或 adb shell 出现 error: insufficient permissions for device,而且输入 adb devices 显示:

```
List of devices attached
???????????? no permissions
```

```
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push hehe.c /data
error: insufficient permissions for device
wuhuating@PCwuhuating:~/Desktop/exercise$
wuhuating@PCwuhuating:~/Desktop/exercise$ adb devices
List of devices attached
???????????? no permissions
```

解决方法:

首先在终端查看 usb 的 ID, 输入 lsusb, 可以查看到刚插入的 usb 的 ID 号, 如:

```
wuhuating@PCwuhuating:~/Desktop/exercise$ lsusb
Bus 002 Device 002: ID 8087:8000 Intel Corp.
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 002: ID 8087:8008 Intel Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 061: ID 18d1:d002 Google Inc.
Bus 003 Device 003: ID 1c4f:0065 SiGma Micro
Bus 003 Device 002: ID 258a:002a
Bus 003 Device 005: ID 067b:2303 Prolific Technology, Inc. PL2303 Serial Port
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

最后一行是刚插入的 usb ID 号。接着我们进入到 cd /etc/udev/rules.d/ 下, 新建一个

51-android.rules 文件 ( sudo vim 51-android.rules ), 在文件中写入 SUBSYSTEM=="usb", ATTRS{idVendor}=="1d6b", ATTRS{idProduct}=="0002",MODE="0666"

保存退出, 然后执行 sudo chmod a+x 51-android.rules 。拔掉 usb 再重新插上, 执行 adb devices

就可以识别了。

```
wuhuating@PCwuhuating:~$ adb devices
List of devices attached
c22ffffffffff00000000 device
```

### 4.2. adb push 之后, pc 终端无反应

经测试, 目前一次 push 到小机端文件的大小超过 5M, push 的速度大约为 31KB/s, 一次推

送太大的文件, adb 传输比较慢, 只要等待一会就好。

```
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push fivebox.test /data
```

```
wuhuating@PCwuhuating:~/Desktop/exercise$  
wuhuating@PCwuhuating:~/Desktop/exercise$  
wuhuating@PCwuhuating:~/Desktop/exercise$ adb push fivebox.test /data  
32 KB/s (5242880 bytes in 159.812s)
```

### 4.3 adb 使用注意事项

一、PC 端执行 adb 相关命令后, 如果使用 CTRL+C 退出, 需要注意下面几点:

1.adb shell + command

- \* 如果该 command 是不会主动退出的, 那么 PC 端执行 CTRL+C 退出后, 该 command 仍然会继续运行, 并且打印从 uart 输出

- \* 如果该 command 是运行一定时间后自动退出的, 那么如果在 timeout 之前 PC 端执行 CTRL+C 退出, 该 command 仍然会继续运行直到自行退出, 并且打印从 uart 输出

- \* 上述两种情况下, command 对应的部分任务资源不会被释放(如信号量), 会造成泄露

- \* 如果出现过 PC 端执行 CTRL+C 退出的情况, adbd 任务会记录下来(最多 32 个记录),

并且会在每次使用 adb 命令时从串口打印提示, 如:

```
adbd shell command maybe alive[position=0]:  
=====Start=====  
thread name:play /data/16000.wav  
=====End=====
```

说明之前有执行过 adb shell play /data/16000.wav, 并且被 CTRL+C 中断退出了

2.adb push

- \* 如果 adb push 命令结束前, PC 端执行 CTRL+C 退出, 实际上仍然会隐式的传输该文件, 直到传输结束, 但此时仍然可以同时运行其他 adb 命令(Linux 上也是如此)



## 二、adb shell 进入交互模式后需要注意：

\* 如果执行的 command 命令会创建 Task，并且后面执行了 exit 命令退出 adb shell,那么该 Task 仍然会继续运行，并且打印从 uart 输出