



## Linux 系统操作指南

### Revision History

Draft Date	Revision No.	Description	Author
2024/05/22	V1.0	1.初始版本。	invoker_li
2024/06/22	V1.1	1.更新 HDMI 驱动说明 2.更新设备树镜像编译	
2024/07/09	V1.2	提取 LinuxSDK 部分 单独成立一份文档	



## 目录

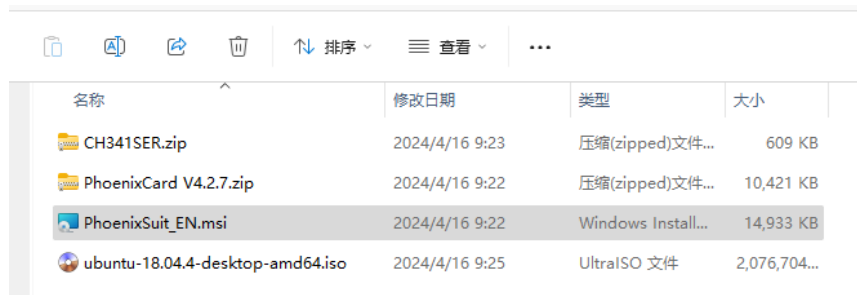
1 通过 USB 固化.....	1
2 U-Boot 命令和环境变量说明.....	4
3 系统信息查询.....	9
4 内存分配说明.....	12
5 程序开机自启动说明.....	14
6 Linux 设备驱动说明.....	15

## 1 通过 USB 固化

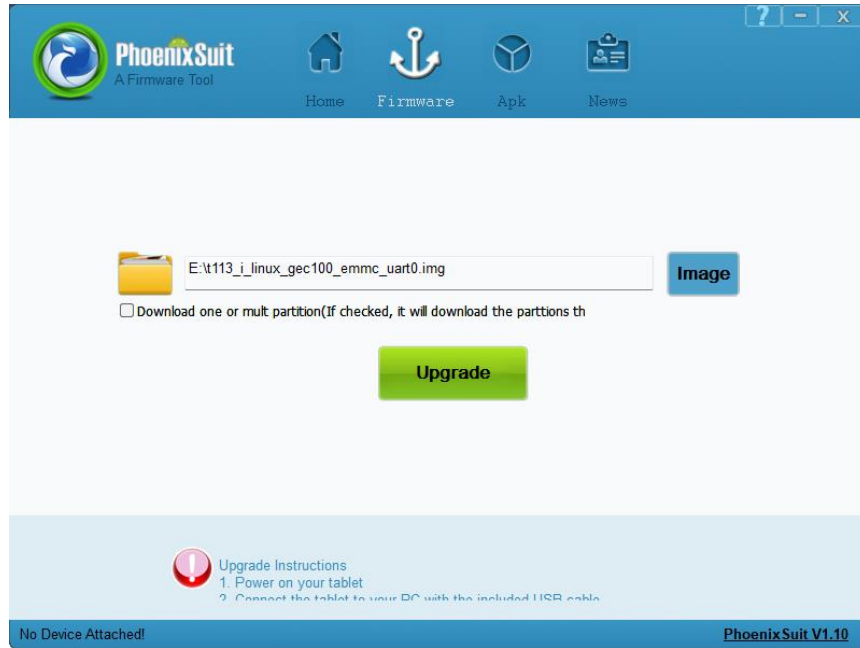
- (1) 将评估板断电，使用 Type-C 线将评估板 OTG 接口连接至 PC 机 USB 接口。



- (2) 双击“软件资料\Tools\Windows\”目录下的“PhoenixSuit\_EN.msi”进行安装，默认安装选项即可，安装完成将在桌面生成快捷方式。PhoenixSuit 工具可在上位机将 Linux 系统镜像通过 USB 接口固化至板载 eMMC。

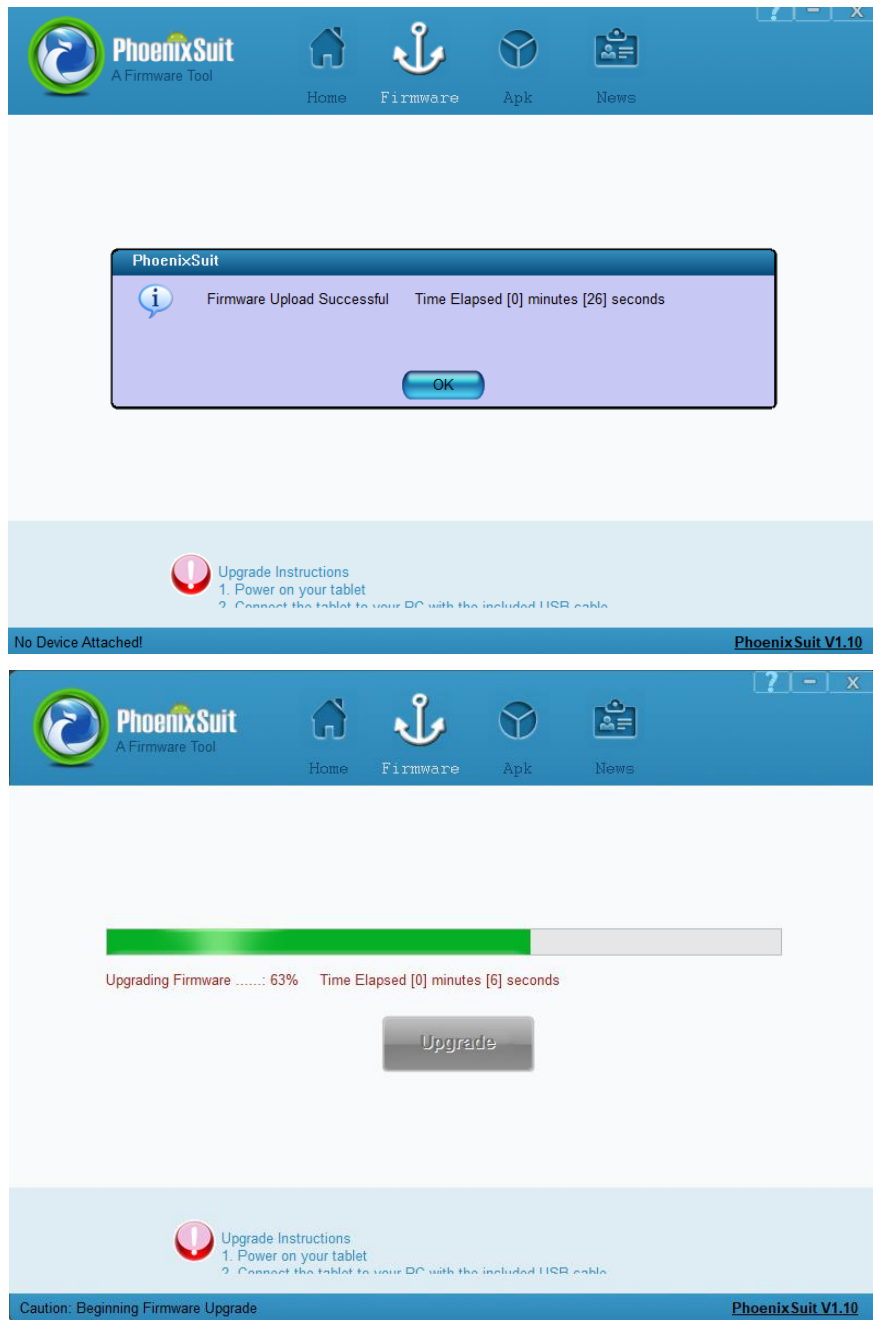


- (3) 双击打开 PhoenixSuit 工具，点击 Firmware，然后选择 Linux 系统镜像文件。

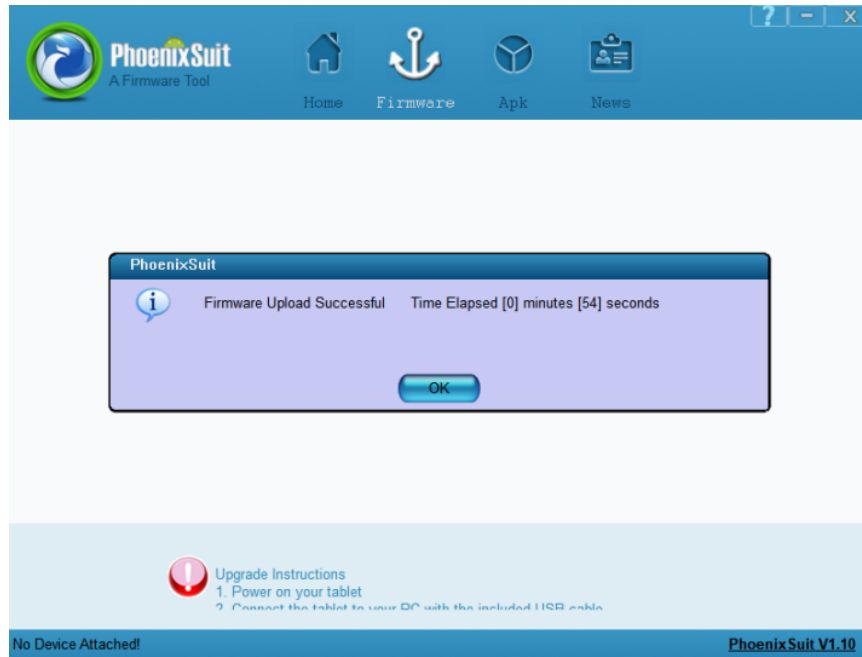


- (4) 确保评估板已断电，使用 Type-C 线连接评估板的 OTG 接口至 PC 机,再长按 DOWNLOAD 按键，将评估板上电启动。当 PhoenixSuit 工具 出现如下界面后，松开 DOWN LOAD 按键。在 PhoenixSuit 工具界面选择"Yes",





- (5) 格式化完成后, PhoenixSuit 工具将自动固化 Linux 系统镜像文件至 eMMC。直至出现如下界面, 表示系统固化成功。



- (6) 请将评估板断电，然后重新上电后，评估板将从 eMMC 启动系统。

## 2 U-Boot 命令和环境变量说明

开发板上电启动后，在 U-Boot 倒计时结束之前按下空格键进入 U-Boot 命令行模式，执行 help 或者?命令，可查看当前 U-Boot 所支持的命令。



Command 1

```
=> help
? - alias for 'help'
base - print or set address offset
bdinfo - print Board Info structure
boot - boot default, i.e., run 'bootcmd'
bootd - boot default, i.e., run 'bootcmd'
bootm - boot application image from memory
bootp - boot image via network using BOOTP/TFTP protocol
bootr - boot application image from memory
chpart - change active partition
cmp - memory compare
colorbar - show colorbar
coninfo - print console devices and information
cp - memory copy
crc32 - checksum calculation
dhcp - boot image via network using DHCP/TFTP protocol
disp - show display status
dns - lookup the IP of a hostname
echo - echo args to console
editenv - edit environment variable
efex - run to efex
env - environment handling commands
erase - erase FLASH memory
exit - exit script
false - do nothing, unsuccessfully
fastboot - fastboot - enter USB Fastboot protocol
fatinfo - print information about filesystem
fatload - load binary file from a dos filesystem
fatls - list files in a directory (default /)
fatsize - determine a file's size
fatwrite - write file into a dos filesystem
fdt - flattened device tree utility commands
flinfo - print FLASH memory information
gpt - GUID Partition Table
help - print command description/usage
i2c - I2C sub-system
itest - return true/false on integer compare
loadb - load binary file over serial line (kermit mode)
loads - load S-Record file over serial line
loadx - load binary file over serial line (xmodem mode)
loady - load binary file over serial line (ymodem mode)
logo - show default logo
loop - infinite loop on address range
md - memory display
memtester - start application at address 'addr'
mm - memory modify (auto-incrementing address)
mmc - MMC sub system
mmcinfo - display MMC info
mtdparts - define flash/nand partitions
mw - memory write (fill)
nfs - boot image via network using NFS protocol
nm - memory modify (constant address)
part - disk partition related commands
ping - send ICMP ECHO_REQUEST to network host
printenv - print environment variables
protect - enable or disable FLASH write protection
```



```

help - print command description/usage
i2c - I2C sub-system
itest - return true/false on integer compare
loadb - load binary file over serial line (kermit mode)
loads - load S-Record file over serial line
loadx - load binary file over serial line (xmodem mode)
loady - load binary file over serial line (ymodem mode)
logo - show default logo
loop - infinite loop on address range
md - memory display
memtester- start application at address 'addr'
mm - memory modify (auto-incrementing address)
mmc - MMC sub system
mmcinfo - display MMC info
mtdparts- define flash/nand partitions
mw - memory write (fill)
nfs - boot image via network using NFS protocol
nm - memory modify (constant address)
part - disk partition related commands
ping - send ICMP ECHO_REQUEST to network host
printenv- print environment variables
protect - enable or disable FLASH write protection
reset - Perform RESET of the CPU
run - run commands in an environment variable
saveenv - save environment variables to persistent storage
screen_char- show default screen chars
set_working_fdt- set_working_fdt fdt_addr
setenv - set environment variables
setexpr - set environment variable as the result of eval expression
showvar - print local hushshell variables
sleep - delay execution for some time
source - run script from memory
sprite_test- do a sprite test
sunxi_bmp_info- manipulate BMP image data
sunxi_bmp_show- manipulate BMP image data
sunxi_card0_probe- probe sunxi card0 device
sunxi_dma- do dma test
sunxi_flash- sunxi_flash sub-system
test - minimal test like /bin/sh
tftpboot- boot image via network using TFTP protocol
tftppt - TFTP put command, for uploading files to a server
tftpsrv - act as a TFTP server and boot the first received file
timer_test- do a timer and int test
timer_test1- do a timer and int test
true - do nothing, successfully
ubi - ubi commands
ubifsload- load file from an UBIFS filesystem
ubifsls - list files in a directory
ubifsmount- mount UBIFS volume
ubifsumount- unmount UBIFS volume
version - print monitor, compiler and linker version
=>

```

命令	解析
<b>setenv</b>	设置或者修改环境变量的值
<b>saveenv</b>	用于保存修改后的环境变量
<b>env default -f -a</b>	恢复默认环境变量
<b>printenv</b>	输出当前 U-Boot 环境变量信息
<b>boot</b>	读取环境变量 bootcmd (U-Boot 启动的命令集合) 来启动 Linux 系统
<b>reset</b>	重新上电获取启动介质参数，传参给环境变量
<b>help 或?</b>	查看当前 U-Boot 支持的命令



执行命令 `setenv` 或 "`env default -f -a`" 修改的是运行空间中的环境变量值，须使用 `saveenv` 命令将修改后的环境变量保存起来。否则 U-Boot 重启后，将会使用修改前的环境变量值。环境变量修改完成后，执行 `reset` 命令，即可使用修改后的 U-Boot 环境变量启

```
U-Boot# env default -f -a
U-Boot# saveenv
U-Boot# reset
```

Command 2

```
=> env default -f -a
## Resetting to default environment
=> saveenv
Saving Environment to SUNXI_FLASH... Writing to redundant env... OK
=> reset
```

在 U-Boot 命令行执行 `printenv` 命令可查看环境变量，以下提供评估板从 NAND FLASH 启动时主要 U-Boot 环境变量的解析说明。不同启动方式（如 Micro SD、eMMC 启动）、不同版本的 U-Boot，环境变量会有所不同，内容仅供参考。

`/*dsp0 启动参数，sunxi_flash 会根据当前启动介质（如 eMMC、Micro SD 等）读取对应的文件。例如，GEC100 从 eMMC 启动，boot_dsp0 环境变量指定读取 eMMC dsp0 分区的数据至 DDR 的 43000000*/`

```
boot_dsp0=sunxi_flash read 43000000 ${dsp0_partition};bootr 4300000000
```

`/*网络启动参数，板卡从 TFTP 服务端获取 boot.fex 文件。至 DDR 的 43000000*/`

```
boot_net=tftpboot 43000000 boot.fex;run setargs_netboot;bootm 43000000
```

`/*fastboot 模式*/`

```
boot_fastboot=fastboot
```

`/*normal 模式启动参数，sunxi_flash 会根据当前启动介质（如 eMMC、Micro SD 等）读取对应的文件。例如，GEC100 从 eMMC 启动，boot_normal 环境变量指定读取`

`eMMC boot 分区的数据至 DDR 的 0x43000000*/`

```
boot_normal=sunxi_flash read 43000000 boot;bootm 43000000
```



```
/*recovery 模式启动参数，sunxi_flash 会根据当前启动介质（如 eMMC、Micro SD  
等）读取对应的文件。例如，GEC100 从 eMMC 启动，boot_recovery 环境变量指定读  
取 eMMC recovery 分区的数据至 DDR 的 0x43000000*/  
  
boot_recovery=sunxi_flash read 43000000 recovery;bootm 43000000  
  
/*启动命令*/  
  
bootcmd=run boottestcount;run setargs_nand_ubi boot_normal  
  
/*启动次数*/  
  
bootcount=0  
  
/*是否启用计数功能*/  
  
boottest=0  
  
/*启动延时为 1 秒*/  
  
bootdelay=1  
  
/*调试串口为 UART0，波特率为 115200*/  
  
console=ttyAS0,115200  
  
/*dsp0 模式启动镜像分区名*/  
  
dsp0_partition=dsp0  
  
/*初始化串口控制台之前，使用 uart8250,mmio32,0x02500000 打印启动信息*/  
  
earlycon=uart8250,mmio32,0x02500000  
  
/*分区表参数*/  
  
partitions=mbr@ubi0_0:boot-resource@ubi0_1:env@ubi0_2:env-redund@ubi0_3:boot@ubi0_4:  
rootfs@ubi0_5:dsp0@ubi0_6:private@ubi0_7:UDISK@ubi0_8:rootfstype=ubifs,rw  
  
/*网络启动参数*/
```



```
setargs_netboot=setenvbootargsclk_ignore_unusedinitcall_debug=${initcall_debug}console=
${console}loglevel=${loglevel}root=/dev/nfsrootwait ip=${ipaddr}:${serverip}:${gatewayip}:
${netmask}::eth0:off nfsroot=${serverip}:${nfsroot},nolock,v3,tcp init=${init} partitions=${par
titions} cma=${cma} snum=${snum}mac_addr=${mac}wifi_mac=${wifi_mac}bt_mac=${bt_ma
c}specialstr=${specialstr} gpt=1
```

### 3 系统信息查询

系统启动后，可参照如下方法查询系统相关信息。

- (1) 执行如下命令查看操作系统登录欢迎信息。

**Target#** cat /etc/issue

Command 3

```
root@T113-GINGKO:~# cat /etc/issue
welcome to Allwinner KunoOS Platform
root@T113-GINGKO:~# █
```

操作系统登录欢迎信息记录在"/etc/issue"文件，直接修改此文件内容改变系统登录信息。

- (2) 执行如下命令查看计算机名字。

**Target#** hostname

Command 4

```
root@T113-GINGKO:~# hostname
T113-GINGKO
root@T113-GINGKO:~#
```

主机名字记录在"/etc/hostname"里面，可修改此文件内容改变主机名字。

- (3) 执行如下命令可查看当前 Linux 内核版本，详细信息解析如下表。

**Target#** cat /proc/version

Command 5

```
root@T113-GINGKO:~# cat /proc/version
Linux version 5.4.61 (zhubuntu) (arm-linux-gnueabi-gcc (Linaro GCC 5.3-2016.05) 5.3.1 20160412, GNU ld (Linaro_Binutils-2016.05)
2.25.0 Linaro 2016_02) #2 SMP PREEMPT Fri Apr 12 01:15:40 PDT 2024
root@T113-GINGKO:~#
```

内容	说明
5.4.61	内核版本
arm-linux-gnueabi-gcc (Linaro GCC 5.3 -2016.05)	交叉编译工具版本
#1/#2	清理内核后编译次数
SMP PREEMPT/SMP PREEMPT_RT	SMP PREEMPT: Linux 内核 SMP PREEMPT_RT: Linux-RT 内核
Fri Apr 12 01:15:40 PDT 2024	内核镜像编译时间

(4) 执行如下命令可查看当前文件系统版本信息。

**Target#** `cat /etc/os-release`

Command 6

```
root@T113-GINGKO:/home# cat /etc/os-release
NAME=Buildroot
VERSION=2019.02.1 Fri Apr 12 02:01:20 PDT 2024
ID=buildroot
VERSION_ID=2019.02.1
PRETTY_NAME="Buildroot 2019.02.1"
root@T113-GINGKO:/home#
```

(5) 执行如下命令查看当前 CPU 使用率。

**Target#** `top`

Command 7

```
top - 09:17:06 up 15 min, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 69 total, 1 running, 68 sleeping, 0 stopped, 0 zombie
%Cpu0  :  0.0/0.0  0 [
%cpu1  :  0.0/0.7  1 [
GiB Mem : 11.2/0.5  [
GiB Swap:  0.0/0.0  [

  PID USER  PR  NI  VIRT  RES  %CPU  %MEM  TIME+ S COMMAND
    1 root    20   0   2.4m  0.4m  0.0   0.1   0:02.44 S init
  1243 root    20   0   2.4m  0.4m  0.0   0.1   0:00.02 S - /sbin/syslog+
  1247 root    20   0   2.4m  0.4m  0.0   0.1   0:00.01 S - /sbin/klogd+
  1250 root    20   0   2.5m  1.6m  0.0   0.3   0:00.29 S - /sbin/udev +
  1355 root    20   0   2.3m  1.4m  0.0   0.3   0:00.00 S - dbus-daemon +
  1359 root    20   0   1.8m  1.3m  0.0   0.3   0:00.01 S - /usr/bin/rpc+
  1391 root    20   0   7.3m  2.2m  0.0   0.5   0:00.22 S - /usr/sbin/nt+
  1395 mosquit+ 20   0   5.1m  2.1m  0.0   0.4   0:00.29 S - /usr/sbin/mo+
  1404 root    20   0   4.9m  1.6m  0.0   0.3   0:00.00 S - /usr/sbin/ss+
  1409 root    20   0   2.4m  0.4m  0.0   0.1   0:00.00 S - /usr/sbin/te+
  1413 root    20   0  887.9m 43.9m  0.0   9.0   0:01.97 S - /usr/bin/doc+
  1543 root    20   0  870.2m 30.3m  0.0   6.2   0:02.77 S - container+
  1420 root    20   0   2.2m  1.6m  0.0   0.3   0:00.00 S - rpc.statd
  1432 root    20   0   2.1m  0.4m  0.0   0.1   0:00.00 S - rpc.mountd
  1437 root    20   0   4.3m  1.6m  0.0   0.3   0:00.00 S - vsftpd
  1536 root    20   0   4.7m  2.6m  0.0   0.5   0:00.06 S - -sh
```

(6) 执行如下命令查看当前内存使用情况。

Target# `cat /proc/meminfo`

Command 8

```
root@T113-GINGKO:/home# cat /proc/meminfo
MemTotal: 497540 kB
MemFree: 352284 kB
MemAvailable: 441956 kB
Buffers: 4312 kB
Cached: 90880 kB
SwapCached: 0 kB
Active: 28996 kB
Inactive: 84788 kB
Active(anon): 18656 kB
Inactive(anon): 136 kB
Active(file): 10340 kB
Inactive(file): 84652 kB
Unevictable: 0 kB
Mlocked: 0 kB
HighTotal: 0 kB
HighFree: 0 kB
LowTotal: 497540 kB
LowFree: 352284 kB
SwapTotal: 0 kB
SwapFree: 0 kB
Dirty: 0 kB
writeback: 0 kB
AnonPages: 18588 kB
Mapped: 67152 kB
Shmem: 204 kB
KReclaimable: 5212 kB
Slab: 16276 kB
SReclaimable: 5212 kB
SUnreclaim: 11064 kB
KernelStack: 680 kB
PageTables: 596 kB
NFS_Unstable: 0 kB
Bounce: 0 kB
writebackTmp: 0 kB
CommitLimit: 248768 kB
Committed_AS: 159700 kB
VmallocTotal: 507904 kB
VmallocUsed: 2348 kB
VmallocChunk: 0 kB
PerCpu: 168 kB
CmaTotal: 16384 kB
CmaFree: 436 kB
root@T113-GINGKO:/home#
```

(7) 执行如下命令查看系统环境变量信息。



Target# env

Command 9

```
root@T113-GINGKO:/home# env
TSLIB_CALIBFILE=/etc/pointerca1
LD_LIBRARY_PATH=/usr/lib/arm-qt/lib:/usr/lib/arm-qt/plugins/platforms:/usr/lib/arm-qt/lib
QTDIR=/usr/lib/arm-qt
EDITOR=/bin/vi
QT_QPA_FONTDIR=/usr/lib/arm-qt/lib/fonts
TSLIB_PLUGINDIR=/usr/lib/ts
TSLIB_CONFIGFILE=/etc/ts.conf
USER=root
PAGER=/bin/more
QT_QPA_PLATFORM_PLUGIN_PATH=/usr/lib/arm-qt/plugins
PWD=/home
HOME=/root
LC_CTYPE=zh_CN.UTF-8
QT_QPA_GENERIC_PLUGINS=tslib:/dev/input/event2
TSLIB_TSDEVICE=/dev/input/event2
QT_PLUGIN_PATH=/usr/lib/arm-qt/plugins
SHELL=/bin/sh
TERM=vt102
SHLVL=1
TSLIB_FBDEVICE=/dev/fb0
QT_QPA_PLATFORM=linuxfb:tty=/dev/fb0
LOGNAME=root
XDG_RUNTIME_DIR=/dev/shm
TSLIB_CONSOLEDEVICE=none
PATH=/bin:/sbin:/usr/bin:/usr/sbin
QT_BOOT=/usr/lib/arm-qt
PS1=\u@h:\w\$
_=/usr/bin/env
OLDPWD=/root
root@T113-GINGKO:/home#
```

(8) 执行如下命令可查看文件系统支持库的存放目录。

Target# ls /usr/lib/

Command 10

```
root@T113-GINGKO:/home# ls /usr/lib
```

```
libxkbcommon.so
libxkbcommon.so.0
libxkbcommon.so.0.0.0
libxmetadataretriever.so
libxml2.so
libxml2.so.2
libxml2.so.2.9.9
libxplayer.so
libxtables.so
libxtables.so.12
libxtables.so.12.2.0
libz.so
libz.so.1
libz.so.1.2.11
locale
```

## 4 内存分配说明

核心板 DDR 内存使用分配说明如下:



DDR 容量	用途	地址范围	大小
128MByte	Linux 系统	0x40000000~0x47FFFFFF	128MByte
256MByte	Linux 系统	0x40000000~0x4FFFFFFF	256MByte
512MByte	Linux 系统	0x40000000~0x5FFFFFFF	512MByte

(1) 执行如下命令查看内存空间。本次操作为 512MByte DDR 配置核心板的查询结果。

**Target#** mount -t debugfs none /sys/kernel/debug

**Target#** cat /sys/kernel/debug/memblock/memory

Command 11

```
root@T113-GINGKO:/home# mount -t debugfs none /sys/kernel/debug/
root@T113-GINGKO:/home# cat /sys/kernel/debug/memblock/memory
0: 0x40000000..0x41ffffff
1: 0x42100000..0x5ffffff
root@T113-GINGKO:/home# █
```

(2) 执行如下命令查看内核已经分配的内存。

**Target#** cat /sys/kernel/debug/memblock/reserved

Command 12

```
root@T113-GINGKO:/home# cat /sys/kernel/debug/memblock/reserved
0: 0x40004000..0x40007fff
1: 0x40100000..0x40ec5c7f
2: 0x41800000..0x4181ffff
3: 0x41900000..0x41ffffff
4: 0x5eac99c0..0x5eb29cc3
5: 0x5eb29d00..0x5eb47fff
6: 0x5eb4a000..0x5eb4a0bf
7: 0x5eb4a0f0..0x5effefff
8: 0x5efff040..0x5efff098
9: 0x5efff0c0..0x5efff11f
10: 0x5efff140..0x5efff2c3
11: 0x5efff300..0x5efff4d8
12: 0x5efff500..0x5efff577
13: 0x5efff580..0x5efff587
14: 0x5efff5c0..0x5efff5c7
15: 0x5efff600..0x5efff603
16: 0x5efff640..0x5efff830
17: 0x5efff840..0x5efffa30
18: 0x5efffa40..0x5efffc30
19: 0x5efffc34..0x5efffc9a
20: 0x5efffc9c..0x5efffc2
21: 0x5efffc2b4..0x5efffe4e
22: 0x5efffe50..0x5efffe6a
23: 0x5efffe6c..0x5efffe86
24: 0x5efffe88..0x5efffea2
25: 0x5efffea4..0x5efffebe
26: 0x5efffec0..0x5effff83
27: 0x5effff94..0x5effffae
28: 0x5effffb0..0x5ffffff
root@T113-GINGKO:/home#
```

编号 28 为 CMA (连续内存区管理) 空间, 地址范围一般由内核自动分配 CMA 的预留内存是分配给驱动使用, 应用层不可使用。

(3) Linux 内核"t113-i\_v1.0/kernel/linux-5.4/arch/arm/boot/dts/sun8iw20p1.dtsi"文件中默认分配了 64KByte 的 share space0 内存空间, 用于 DSP、ARM 数据存储以及 DSP Log 信息存储

```
241 share_space0: share_space@0x42100000 {
242     #address-cells = <1>;
243     #size-cells = <1>;
244     device_type = "share_space0";
245     /* dsp write addr and len */
246     /* arm write addr and len */
247     /* dsp log addr and len */
248     /* reserve */
249     reg = < 0x42100000 0x00001000
250           0x42101000 0x00002000
251           0x42102000 0x0000E000
252           0x00000000 0x00000000>;
253 };
```

## 5 程序开机自启动说明

通过命令行修改自启动脚本, 实现程序开机自启动。请将产品资料“软件资料\Demo\base-demos\led\_flash\bin\”目录下的可执行文件拷贝至评估板文件系统 root 目录下。进入文件系统, 执行如下命令修改"/etc/init.d/"目录下的 rcS 配置文件。

Target# vi /etc/init.d/rcS

Command 13

```
root@T113-GINGKO:~# ls /etc/init.d/rcs
/etc/init.d/rcs
root@T113-GINGKO:~# vi /etc/init.d/rcs
```

在配置文件文末修改为如下内容:

```
#. /etc/qtenv.sh && Launcher //注释默认 Qt 开机程序
/root/led_flash -n 1 & //指定自启动的程序路径
```

```
#
for i in /etc/init.d/s??* ;do
    # Ignore dangling symlinks (if any).
    [ ! -f "$i" ] && continue
    case "$i" in
        *.sh)
            # Source shell script for speed.
            (
                trap - INT QUIT TSTP
                set start
                . $i
            )
            ;;
        *)
            # No sh extension, so fork subprocess.
            $i start
            ;;
    esac
done
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
#/etc/adb_conf.sh start &
tinymix set "ADC2 Input MIC2 Boost Switch" 1 > /dev/null
tinymix set "ADC3 Input MIC3 Boost Switch" 1 > /dev/null
tinymix set "Hpspeaker Switch" 1 > /dev/null
tinymix set "Headphone Switch" 1 > /dev/null
if [ ! -f "/usr/lib/libcrypto.so.1.0.0" ]; then
ln -s /usr/lib/libcrypto.so.1.1 /usr/lib/libcrypto.so.1.0.0
fi
if [ ! -f "/usr/lib/libssl.so.1.0.0" ]; then
ln -s /usr/lib/libssl.so.1.1 /usr/lib/libssl.so.1.0.0
fi
# ./etc/qtenv.sh && Launcher &
/root/led_flash -n 1 &
~
~
~
I /etc/init.d/rcS [Modified] 47/46 102%
```

修改完成后，保存并退出。请将评估板断电重启，Linux 系统将会自动运行程序，调试串口终端将打印如下类似信息，同时可观察到评估底板 LED 进行闪烁。

```
system leds :
/sys/class/leds/heartbeat
/sys/class/leds/user-beep
/sys/class/leds/user-led0

Flashing leds :
/sys/class/leds/user-led0
welcome to Allwinner KunOS Platform
```

如需取消程序开机自启动，请修改 rcS 配置文件，删除相应命令。

## 6 Linux 设备驱动说明

如下为主要的 Linux 设备驱动说明：

设备名	Linux 内核驱动源码路径	Linux 设备文件或头文件
-----	----------------	----------------



LED	drivers/leds/leds-gpio.c	/sys/class/leds/user-ledX
KEY(LRADC)	drivers/input/keyboard/sunxi-keyboard.c	/dev/input/eventX
USB TO UART0	drivers/tty/serial/sunxi-uart.c	/dev/ttyAS0
Micro SD/eMMC	drivers/mmc/host/sunxi-mmc.c	/dev/mmcblkX /run/media/mmcblkX
I2C	drivers/i2c/busses/i2c-sunxi.c	/dev/i2c-X
SPI	drivers/spi/spi-sunxi.c	/sys/bus/spi
SPI NAND FLASH	drivers/mtd/awnand/spinand/	/dev/mtdX /dev/mtdblockX
HP OUT/MIC IN	sound/soc/sunxi/sun8iw20-codec.c sound/soc/sunxi/sun8iw20-sndcodec.c sound/soc/sunxi/sunxi-dummy-cpudai.c sound/soc/codecs/dmic.c sound/soc/sunxi/sunxi-simple-card.c	/dev/snd/controlC0 /dev/snd/pcmC0D0c /dev/snd/pcmC0D0p
Display Frame Driver (HDMI)	drivers/video/fbdev/sunxi/disp2/	/dev/fbX /dev/disp
HDMI OUT	drivers/gpu/drm/bridge/sii902x.c	/dev/fbX /dev/disp
USB HOST	drivers/usb/host/	/sys/bus/usb/
USB DRD	drivers/usb/sunxi_usb/manager/ drivers/usb/sunxi_usb/udc/	/sys/bus/usb/ /sys/devices/platform/soc\@3000000/soc\@3000000\usb0\@0/
ETH0	drivers/net/phy/icplus.c	/sys/class/net/ethX

备注：“X”表示可变化的数值。