



ARK1668ED 动画功能实现

1、动画分区的大小设置:

Nandflash 启动模式:

```
Architecture select (ARM architecture) --->
Skipping low level initialization functions --->
ARM architecture --->
Functionality shared between NXP SoCs --->
General setup --->
[ ] Enable U-Boot API
Boot options --->
Console --->
Logging --->
Init options --->
Security support --->
Update support --->
Blob list --->
[*] Enable SPL
    SPL configuration options --->
[ ] Enable VPL
[ ] FDT tools for simplefb support
[ ] Enable bmp image display
[ ] Command line interface --->
    Partition types --->
    Device Tree Control --->
    Environment --->
[*] Networking support --->
(4) Number of receive packet buffers
Device Drivers --->
File systems --->
Library routines --->
FWU Multi Bank Updates ----
[ ] Power On Self Test support
Unit tests --->
Tools options --->
```

```
[*] Support U-Boot commands
[*] Use hush shell
[*] Enable command line editing
[ ] Enable support for changing the command prompt string at run-time
[*] Enable auto complete using TAB
[*] Enable long help messages
(=> ) Shell prompt
(> ) Hush shell secondary prompt
(16) Maximum number arguments accepted by commands
(512) Console input buffer size
(532) Buffer size for console output
[*] Command execution tracer
*** Commands ***
Info commands --->
Boot commands --->
Environment commands --->
Memory commands --->
Compression commands --->
Device access commands --->
Shell scripting commands --->
Android support commands ----
[*] Network commands --->
[ ] ethsw
[ ] pxe
[ ] wol
Misc commands --->
TI specific command line interface --->
Power commands ----
Security commands --->
Firmware commands ----
[ ] filesystem commands --->
Debug commands ----
[*] Enable UBI - Unsorted block images commands
v(+)
```

<select> <Exit> <Help> <Save> <Load>



```
[ ] enable the 'btsubvol' command
[ ] EROFS command support
[ ] ext2 command support
[ ] ext4 command support
[*] FAT command support
[ ] SquashFS command support
[ ] filesystem commands
[ ] fsuuid command
[ ] jffs2 command
[*] MTD partition support
[ ] Padd partition size to take account of bad blocks
[ ] Show net size (w/o bad blocks) of partitions
(nand0=ark-nand) Default MTD IDs
(mtdparts=ark-nand:256k(bootstrap),1m(bootloader),1m(bootloader_bak),512k(bootloaderenv),512k(bootloaderenv_bak),512k(fdt),7m
[ ] Reiser - Access to Reiserfs filesystems
[ ] zfs - Access of ZFS filesystem
```

在红线的位置进行修改;

EMMC 启动模式:

```
menu. <Enter> selects submenus --- (or empty submenus ---). Highlighted letters are notkeys. Pressing <t> includes, <n> excludes, <m> modularizes
< > to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module capable

*** Compiler: arm-none-linux-gnueabi-gcc (Arm GNU Toolchain 13.2.rel1 (Build arm-13.7)) 13.2.1 20231009 ***
Architecture select (ARM architecture) --->
Skipping low level initialization functions --->
[ ] ARM architecture --->
Functionality shared between NXP SoCs --->
General setup --->
[ ] Enable U-Boot API
Boot options --->
Console --->
Logging --->
Init options --->
Security support --->
Update support --->
Blob list --->
[*] Enable SPL
[*] SPL configuration options --->
[ ] Enable VPL
[ ] FDT tools for simplefb support
[ ] Enable bmp image display
Command line interface --->
Partition Types --->
Device Tree Control --->
Environment --->
[*] Networking support --->
(4) Number of receive packet buffers
Device Drivers --->
File systems --->
Library routines --->
Firmware Multi Bank Updates --->
[ ] Power On Self Test support
Unit tests --->
Tools options --->
```

```
[ ] ARM PL310 L2 cache controller in SPL
[ ] L2cache off
[ ] prepare BOOT0 header
[*] Use an assembly optimized implementation of memcpy
[ ] Use an assembly optimized implementation of memcpy for SPL
[*] Use an assembly optimized implementation of memset
[ ] Use an assembly optimized implementation of memset for SPL
Target select (arkmicro soc) --->
[ ] Support pre-devicetree ATAG-based booting
[ ] Statically define the Machine ID number
[ ] Enable DIPS detection for CHIP board
[ ] Enable support for a hardware secure memory area
[ ] Support the 'dek blob' command
[ ] Support the DEK blob encapsulation with CAAM U-Boot driver
[ ] Support the DEK blob encapsulation with OP-TEE
[ ] Support the DEK blob encapsulation with SEC0
[ ] Support the DEK blob encapsulation with ELE
[ ] Support the 'hdmidet' command
(0x00910000) DCD Blocks location on the image
[ ] Enable SPL loading U-Boot as a i.MX Container image
( ) i.MX Container config file
[*] Arkmicro board select (ark1668ed_devb_emmc board) --->
(0x50900000) GPIO Controller Base
(0x50b00000) PWM Controller Base
(24000000) PWM Source Clock Freq
(101) SPI CS0 GPIO
(1) sd dev part
(0) emmc dev part
(mmcblk0:512k(bootstrap),1m(bootloader),1m(bootloader_bak),2m(bootloaderenv),5m(bootanimation),4m(reversingtrack),30m(userdat)
[ ] usb-support
(2) mcu serial port
(115200) mcu serial baud rate
ARM debug --->
```

在红线的位置进行 emmc 添加动画分区;

2、动画内存地址的设置:

在 内核中, 需要分配相应的动画数据空间: 主要在设备树文件, 其路径为:

ark1668ed-bsp/linux/arch/arm/boot/dts/arkmicro 下:



```
ark1668ed.dtsi ark1668ed_devb_emmc.dts ai
21  struct path = ser_tst0-119200n0 ;
22  };
23
24  cpus {
25      #address-cells = <1>;
26      #size-cells = <0>;
27      enable-method = "arkmicro,arke-smp";
28
29      cpu0: cpu@0 {
30          compatible = "arm,cortex-a7";
31          device_type = "cpu";
32          reg = <0>;
33          clock-frequency = <800000000>;
34          next-level-cache = <&L2_CA7>;
35      };
36
37      cpu1: cpu@1 {
38          compatible = "arm,cortex-a7";
39          device_type = "cpu";
40          reg = <1>;
41          clock-frequency = <800000000>;
42          next-level-cache = <&L2_CA7>;
43      };
44
45      L2_CA7: cache-controller-0 {
46          compatible = "cache";
47          cache-unified;
48          cache-level = <2>;
49      };
50  };
51
52  memory {
53      reg = <0x60000000 0x1c000000>;
54  };
55
56  reserved-memory {
57      #address-cells = <1>;
58      #size-cells = <1>;
59      ranges;
60
61      /* global autoconfigured region for contiguous allocations */
```

memory 中 0x1c000000 为内存的大小，0x60000000 为内存的起始位置。Ark1668ed 内存起始地址是固定的，内存大小要根据 ddr 的大小以及显示的空间来确定分配大小。目前是在 512M 的 ddr 的基础上，进行分配的。分配空间的大小和位置如图所示（DDR256M 的可根据相应的偏移进行修改）：



```
ark1668ed.dtsi ark1668ed_devb_emmc.dts ark1668ed-pinctrl.dtsi
924         status = "disabled";
925     };
926 };
927 #endif
928 gpu0: gpu@40600000 {
929     compatible = "arkmicro,ark1668ed_gc555";
930     reg = <0x40600000 0x1000>;
931     interrupts = <GIC_SPI 4 IRQ_TYPE_LEVEL_HIGH>;
932 };
933
934 vdec0: vdec@40700000 {
935     compatible = "on2,ark-vdec";
936     reg = <0x40700000 0x1000
937           0x7c000000 0x500000>;
938     interrupts = <GIC_SPI 1 IRQ_TYPE_LEVEL_HIGH>;
939     clocks = <&mfccclk>;
940     clock-names = "vdec_clk";
941     //status = "disabled";
942 };
943
944 axi_scale: axi-scale@41000000 {
945     compatible = "arkmicro,ark1668ed-axi-scale";
946     reg = <0x41000000 0x1000
947           0x50000000 0x1000>;
948     interrupts = <GIC_SPI 8 IRQ_TYPE_LEVEL_HIGH>;
949     clocks = <&scalclk>;
950     clock-names = "scale_clk";
951     /*
952     softreset-reg = <0x74>;
953     softreset-offset = <28>;
954 */
955 };
956
957 gpu_2d: gpu-2d@40500000 {
958     compatible = "arkmicro,gpu-2d";
959 }
```

分配的动画的空间是 0x500000 (5 M) 空间。(客户可根据实际情况进行分配不能超过 8M)。

在 uboot 中需要进行相应的数据的地址设置：目前按照 ark1668ed_devb 分支进行说明：

在/ark1668ed-bsp/u-boot/include/configs / ark1668ed_devb.h

Nandflash 需要修改：

```
62
63
64 #ifdef CONFIG_NAND_BOOT
65
66 #define CFG_EXTRA_ENV_SETTINGS \
67     "need_update=yes\0" \
68     "update_dev_type=usb\0" \
69     "update_dev_part=0\0" \
70     "sd_dev_part="CONFIG_SD_DEV_PART"\0" \
71     "loadaddr=0x64000000\0" \
72     "cmploadaddr=0x66000000\0" \
73     "fdtaddr=0x62000000\0" \
74     "kerneladdr=0x62100000\0" \
75     "bootanimationaddr=0x7c000000\0" \
76     "bootanimationsize=0\0" \
77     "reversingtrackaddr=0x6ea00000\0" \
78     "reversingtracksize=0\0" \
79     "ubootreset=0\0" \
80     "boardfdt="CONFIG_DEFAULT_FDT_FILE"\0" \
81     NANDARGS
82
83 #define CONFIG_BOOTCOMMAND \
84     "run nandboot;" \
85
```

emmc 需要修改：



```
ark1668ed_devb_emmc.h
1100  static bootloadersize_t fsize;
1101  emmc_erase_part_bootloader_bak;
1102  emmc_write_s(loadaddr) bootloadersize fsize;
1103  emmcargs=bootloadersize console=115200 " \
1104  earlyprintk loglevel=3 clk_ignore_unused 10-240000 " \
1105  $emmcargs;
1106  root=$emmcroot;
1107  rootfstype=$(emmcrootfstype) \
1108  "emmcroot=/dev/mmcblk0p10 rw \
1109  "emmcrootfstype=$(emmcrootfstype) \
1110  "emmcboot=emmc booting from emmc ...; " \
1111  run emmcargs;
1112  emmc_dev $emmc_dev_part;
1113  emmc_read $fdtaddr $fdt_part $fdtsize;
1114  emmc_read $kerneladdr $kernel_part $kernel_size;
1115  emmc_read $bootanimaddr $bootanim_size $bootanim_size;
1116  bootz $kerneladdr - $fdtaddr \
1117
1118
1119 #define CFG_EXTRA_ENV_SETTINGS \
1120 "mtd_update=1" \
1121 "update_dev_type=usb" \
1122 "update_dev_part=0" \
1123 "sd_dev_part="CONFIG_SD_DEV_PART" \
1124 "emmc_dev_part="CONFIG_EMMC_DEV_PART" \
1125 "loadaddr="CONFIG_LOADADDR" \
1126 "kerneladdr="CONFIG_KERNELADDR" \
1127 "bootanimaddr="CONFIG_BOOTANIMADDR" \
1128 "reversingtrackaddr="CONFIG_REVERSINGTRACKADDR" \
1129 "reversingtracksize="CONFIG_REVERSINGTRACKSIZE" \
1130 "ipaddr="IPADDR" \
1131 "update_from_atmega" \
1132 "update_from_part=0" \
1133 "update_status=none" \
1134 "kernel_part=kernel" \
1135 "fdt_part=fdt" \
1136 "rootfs_part=rootfs" \
1137 "bootargs=" \
1138 "boardfdt="CONFIG_DEFAULT_FDT_FILE" \
1139 EMMCARGS
1140
1141 #define CONFIG_BOOTCOMMAND \
1142 "run emmcboot;"
1143
1144 #endif
```