嵌入式系统设计 实验报告

 学生姓名
 许强

 学生学号
 SA18225428

实验日期___2018.11.6____

实 验 报 告

一、 实验名称: LCD 驱动移植

__\

二、实验学时: 4学时

三、实验内容和目的:

- 1. 熟悉 LCD 的接口电路
- 2. 进行 LCD 驱动的移植

目的:熟悉 Linux 操作系统上 LCD 驱动的移植过程。

三、 实验原理:

Linux 内核是 Linux 操作系统的核心,也是整个 Linux 功能体现。它是用 C 语言编写,符合 POSIX 标准。Linux 是一个一体化内核(Monolithic Kernel)系统。设备驱动程序可以完全访问硬件。Linux 内的设备驱动程序可以方便地以模块化(Modularize)的形式设置,并在系统运行期间可直接装载或卸载。

Linux 内核主要功能包括: 进程管理、内存管理、文件管理、设备管理、网络管理等。

Linux 内核源代码非常庞大,随着版本的发展不断增加。它使用目录树结构,并且使用 Makefile 组织配置编译。初次接触 Linux 内核,最好仔细阅读顶层目录的 readme 文件,它是 Linux 内核的概述和编译命令说明。readme 的说明更加针对 X86 等通用的平台,对于某些特殊的体系结构,可能有些特殊的地方。顶层目录的 Makefile 是整个内核配置编译的核心文件,负责组织目录树中子目录的编译管理,还可以设置体系结构和版本号等。

Exynos4412 内部集成了一个显示控制器 FIMD。该控制器支持三种接口,分别是 RGB 接口、indirect-i80 接口和 YUV 接口。在 FS4412 开发板上使用的是 RGB 接口连接外部的 LCD 屏。本次使用的 LCD 为 24 位 RGB 模式,由 PWM1 的输出来控制 LCD 屏的背光。

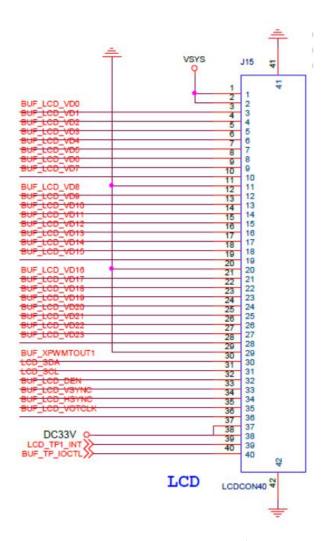


图 1 FS4412LCD 接口电路图(LCD 侧)

LCD_HSYNC	AG17 AG18 AF17 AH16	XvHSYNC/GPF0_0 XvVSYNC/GPF0_1
LCD_VSYNC	AF17	
LCD_DEN <		MYVOIIIO/OI I U_I
	AH16	XvVDEN/GPF0 2
LCD VOTCLK <<		XvVCLK/GPF0 3
LCD VD0	AG19	ATTOLITOT TO_5
ICD VD0	AB14	XvVD0/GPF0 4
LCD VD1	AD14 AD18	XvVD1/GPF0_5
ICD VD3	AE14	XvVD2/GPF0 6
LOD YDO	AF19	XVVD3/GPF0_7
LCD VD4	AB15	XvVD4/GPF1_0
LCD VD5	AD15	XvVD5/GPF1_1
LCD VD5	AB13	XvVD6/GPF1_2
LCD VDI	AD16	XvVD7/GPF1 3
ICD VD9	AG20	XvVD8/GPF1_4
TCD VD40	AE19	XvVD9/GPF1 5
LCD VDIO	AG16	XvVD10/GPF1 6
TCD VD11	AF15	XvVD11/GPF1 7
LCD_VD12	AC15	XvVD12/GPF2 0
LCD VD13	AF18	XvVD13/GPF2 1
LCD VD14	AF16	XVVD14/GPF2 2
LCD VD15	AC13	XvVD15/GPF2 3
LCD VD16	AF13	XVVD16/GPF2 4
LCD VD17	AD15	XvVD17/GPF2 5
LCD VD18	AF18	XvVD18/GPF2 6
ICD VD19	AE17	XvVD19/GPF2 7
LCD VD20	AE16	XvVD20/GPF3 0
LCD VD21	AG12	XvVD21/GPF3 1
LCD VD23	AF14	XvVD22/GPF3 2
LOD_YDZ3	AC16	XvVD23/GPF3 ⁻³
XvVSYNC_LDI >>	AC14	XVVSYNC LDI/GPF3 4
XvSYS_OE >>	AC14	XvSYS OE/GPF3 5

图 2 FS4412 LCD 接口电路图 (CPU 侧)

五、实验步骤:

(一) LCD 驱动移植

(1) 使用 menuconfig 打开配置页面,进行相关配置。

```
Device Drivers --->
Graphics support --->
<*> Direct Rendering Manager (XFree86 4.1.0 and higher DRI support) ----
<*> DRM Support for Samsung SoC EXYNOS Series
[*] Exynos DRM FIMD
```

图 5-1 makeconfig 配置

(2) 修改设备树源文件中关于 FIMD 设备节点和显示时序。将 display-timings 节点移动到 fimd 节点中。添加行、场同步信号,数据使能信号和像素时钟信号的阳极属性。

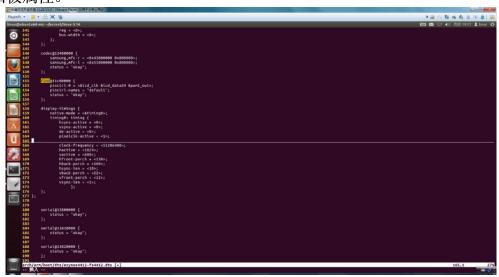


图 5-2 修改设备树文件

(3) 添加两个文件,对 LCD 时钟相关寄存器进行配置,以提高刷新频率。首先设置 LCD 的时钟源为 SCLKVPLL,再设置分频值和时钟使能位,最后选择 FIMD。将两个文件拷贝到 divers/gpu/drm/exynos/目录下,并修改该目录下的 Makefile 文件,添加 exynos_drm_fbclk.o。注意添加相关的头文件。

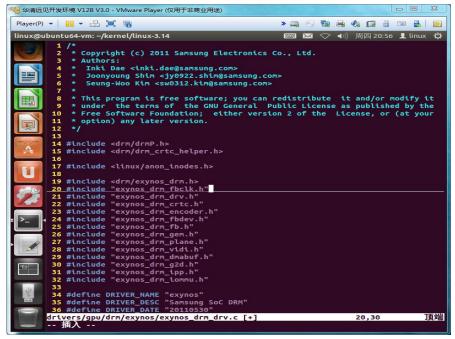


图 5-3 添加相关头文件

(4) 重新编译内核,将编译好的内核镜像和设备树文件拷贝到/tftpboot/目录下, 打开 putty 并重启开发板。

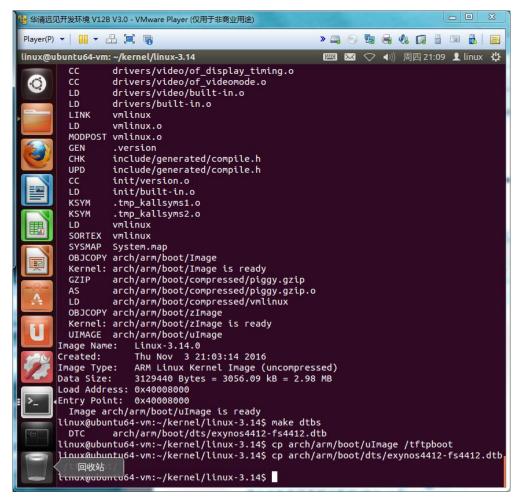
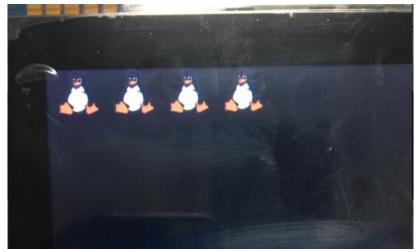


图 5-4 重新编译内核并拷贝至/tftpboot/目录下

```
_ D X
COM4 - PuTTY
     1.565000] mmc1: new high speed DDR MMC card at address 0001
     1.565000] mmcblk0: mmc1:0001 AWPD3R 14.5 GiB
     1.565000] mmcblkOboot0: mmc1:0001 AWPD3R partition 1 4.00 MiB
     1.565000] mmcblk0boot1: mmc1:0001 AWPD3R partition 2 4.00 MiB
     1.565000] mmcblk0rpmb: mmc1:0001 AWPD3R partition 3 4.00 MiB
     1.615000] mmcblk0: p1 p2 p3 p4
     1.625000] mmcblk0boot1: unknown partition table 1.630000] mmcblk0boot0: unknown partition table
      1.635000] dm9000 5000000.ethernet eth0: link down
     1.900000] IP-Config: Guessing netmask 255.255.25.0
1.900000] dm9000 50000000.ethernet eth0: link up, 100Mbps, full-duplex, lpa
      1.910000] IP-Config: Complete:
     1.910000]
                     device=eth0, hwaddr=00:0a:2d:a6:55:a2, ipaddr=192.168.100.19
   mask=255.255.255.0, gw=255.255.255.255
     1.925000] host=192.168.100.191, domain=, nis-domain=(none)
                      bootserver=255.255.255.255, rootserver=192.168.100.192, root
     1.935000] clk: Not disabling unused clocks
     2.260000] VFS: Mounted root (nfs filesystem) on device 0:10. 2.265000] devtmpfs: mounted
      2.265000] Freeing unused kernel memory: 228K (c0568000 - c05a1000)
Please press Enter to activate this console.
```

图 5-5 重启开发板

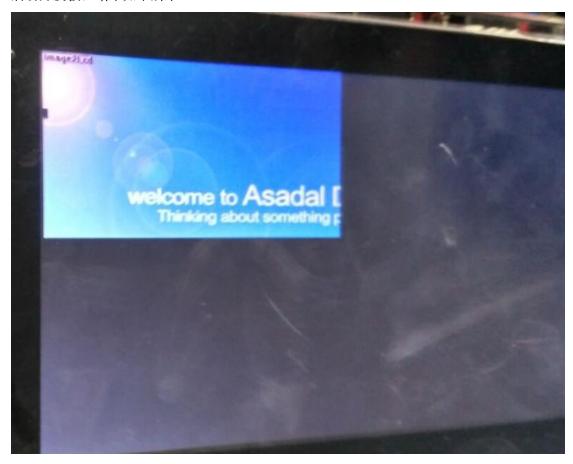


(5) 运行 LCD 测试程序,使用 NFS 挂载的方法进行测试查看 LCD 驱动是否成功。

```
COM4 - PuTTY
     1.860000] dm9000 5000000.ethernet eth0: link up, 100Mbps, full-duplex, lpa
0xCDE1
    1.880000]
                    host=192.168.100.191, domain=, nis-domain=(none)
    1.885000]
                    bootserver=255.255.255.255, rootserver=192.168.100.192, root
path=
     1.890000] clk: Not disabling unused clocks
     3.055000] VFS: Mounted root (nfs filesystem) on device 0:10.
     3.060000] devtmpfs: mounted
    3.065000] Freeing unused kernel memory: 228K (c0568000 - c05a1000)
Please press Enter to activate this console.
[root@farsight]#
[root@farsight]#
[root@farsight]#
[root@farsight]#ls
                linuxrc proc
                                             testfb usr
[root@farsight] #./testfb
The framebuffer device was opened successfully.
1024x600, 32bpp
The framebuffer device was mapped to memory successfully.
The framebuffer device was munmapped to memory successfully.
                                                                                 E
The framebuffer device was closed successfully.
[root@farsight]#
```

图 5-7 驱动移植成功

启动开发板,结果如下所示:



六. 实验总结

添加代码注意格式和字符,注意格式,参考所修改文件格式,如非特别说明不使用 root 权限执行命令,注意执行命令所在的路径(根目录和当前用户目录不一样)。

本次实验时最后一次实验,每次实验都在同一台机器上进行实验并且每次实验都成功做出,因此说明整个环境的配置时正常的,但是在实验的过程中提示找不到编译器,说明环境已将变化,因此在别人机器上完成了本次实验五的实验,完成了 LCD 的显示。

七. 实验结论、问题及改进建议:

关于 LCD 控制寄存器的一些总结:

- 1. LCD 控制器的外部接口信号
- 1) VFRAME:

LCD 控制器和 LCD 驱动器之间的帧同步信号。该信号告诉 LCD 屏新的一帧 开始了。LCD 控制器在一个完整帧显示完成后立即插入一个 VFRAME 信号,开 始新一帧的显示;该信号与 LCD 模块的 YD 信号相对应。

2) VLINE:

LCD 控制器和 LCD 驱动器之间的线同步脉冲信号,该信号用于 LCD 驱动器将水平线(行)移位寄存器的内容传送给 LCD 屏显示。LCD 控制器在整个水平线(整行)数据移入 LCD 驱动器后,插入一个 VLINE 信号;该信号与 LCD 模块的 LP 信号相对应。

- 3) VCLK: LCD 控制器和 LCD 驱动器之间的像素时钟信号,由 LCD 控制器送出的数据在 VCLK 的上升沿处送出,在 VCLK 的下降沿处被 LCD 驱动器采样;该信号与 LCD 模块的 XCK 信号相对应。
 - 2. LCD 的几个控制寄存器
 - 1)LCDCON1

设置 VCLK 频率: VCLK(Hz) = HCLK / [(CLKVAL + 1) * 2]

选择 LCD 类型: TFT LCD

2)LCDCON2/3/4

设置控制信号时间参数 设置分辨率(行数及列数)

3)LCDSADDR1

设置 LCDBANK、LCDBASEU

4)LCDSADDR2

设置 LCDBASEL: 帧缓冲区结束地址 A【21: 1】

5)LCDSADDR3

设置虚拟屏的偏移量