

文章编号: 1005-8451 (2010) 06-0044-05

嵌入式 OLED 驱动及其在列车故障诊断中的应用

孙惠琴

(中国铁道科学研究院 机车车辆研究所, 北京 100081)

摘要: 本文在 AT91SAM9261 硬件平台和嵌入式 Linux 软件平台下, 进行了 OLED 驱动程序的设计和实现。驱动程序起到了硬件与软件应用程序之间的接口作用, 屏蔽了底层硬件的不同, 方便了应用软件开发。本文分析 AT91SAM9261 与 OLED 连接的硬件电路, 进行驱动程序的设计, 详细介绍驱动程序开发的过程, 并阐述其在列车故障诊断系统中的应用前景。

关键词: 嵌入式系统; OLED; 故障诊断

中图分类号: TP39 **文献标识码:** A

Application of embedded OLED device driver to fault diagnosis of train

SUN Hui-qin

(Locomotive & Car Research Institute, China Academy of Railway Sciences, Beijing 100081, China)

Abstract: This paper designed and implemented the OLED device driver based on AT91SAM9261 and Linux. The device driver was the interface of hardware and application software, shielded the difference of hardware and convenient to the application development. This paper analyzed the hardware circuit of AT91SAM9261 connectiong OLED and designed the device driver of OLED. The device driver development was introduced in detail. The brilliant future of OLED device driver was introduced in train.

Key words: Embedded System; OLED; fault diagnosis

在列车的运行过程中, 需要进行故障诊断, 及时报告各个车厢中转向架的运行情况, 包括构架横向的稳定性、空簧的运行状态、列车的平稳性指标、纵向冲动、轴温等, 并在车厢的 OLED 显示屏上显示, 以便及时提醒列车上的工作人员进行处理。因为列车运行很快, 需要高性能的 CPU。本系统的 CPU 选用了工业 ARM 芯片 AT91SAM9261, 主频为 200 MHz, 可以满足列车的要求。显示器采用 OLED128*128, 在此硬件基础上, 编写了嵌入式 linux2.6 的 OLED 驱动程序及应用程序, 成功地进行了故障的显示和报警、车号及车顺号的修改等功能。

1 系统的硬件结构与分析

下面具体阐述 AT91SAM9261 与 OLED 芯片的种类、连接方案。

本系统的 CPU 采用 ARM9 芯片 AT91SAM9261, 它是一种工业级别的 ARM 芯片, 针对功耗低、数

据吞吐量大的应用场合, 主频为 200 MHz, 支持 Linux 2.6 操作系统。

本系统采用 LT1618 升压型 DC/DC 转换器, 它不但有提升电压的功能, 还有使通过 OLED 的电流恒定的功能。LT1618 转换器将一个电压反馈环路和一个电流反馈环路组合在一起, 以便起恒定电流和电压的作用。

本系统的 OLED 显示屏选择 LD50T5128M (SSD1355) 控制器和 OLED 显示面板, LD50T5128M 控制器属于高度集成设备, 支持 128*128 65 K 真彩色的 OLED 图形显示。它以点阵的方式显示汉字、数字和图形, 最大支持 128 行和 128 列点阵, 每个点采用 16 bit 表示, 其中红色占 5 bit、绿色占 6 bit、蓝色占 5 bit。OLED 显示面板大小为 1.5 in, 能够展现逼真的图像色彩。

图 1 为 AT91SAM9261 与 OLED 的连接示意图。

其中, LT1618 的 /SHDN 引脚为停机引脚, 把该引脚与高于 1V 的电压相连则接通 LT1618; 而与低于 0.3 V 的电压相连则关断 LT1618。在图 1 中, /SHDN 与 AT91SAM9261 的 PA20 相连, 在控

收稿日期: 2009-12-23

作者简介: 孙惠琴, 副研究员。

制的时候, 将 PA20 设置为高电平, 从而启动 LT1618 工作, 控制 LT1618 输出一个 13 V 左右的高电平 (VEE), 为 OLED 驱动器供电。这需要在驱动程序中用软件的方法控制实现。

LT1618 的 FB 引脚为反馈引脚。通过选择 R_{33} 和 R_{34} 的阻值来设定输出电压, 公式如下:

$$R_{34} = R_{33} \left(\frac{V_{EE}}{1.263 V} - 1 \right)$$

选择 R_{34} 为 93 k Ω , R_{33} 为 10 k Ω , 因此输出的 V_{EE} 大概为 13 V 左右, 从而使 OLED 驱动器工作。

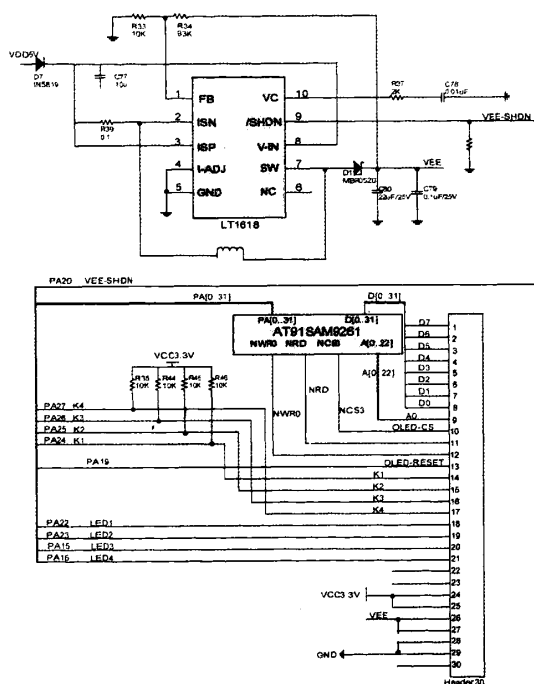


图1 AT91SAM9261 与 OLED 的连接示意图

AT91SAM9261 的 CS3 片选信号选择 OLED, 在 AT91SAM9261 中, CS3 的起始地址为 0X40000000。AT91SAM9261 的数据线 D0~D7, 连接到 OLED 的 1~8 脚, 采用数据和命令复用的机制, 在 D0~D7 上分时传送数据和命令。使用 AT91SAM9261 的地址线 A0 控制的方法进行复用, 如果 A0 为 0, 则 AT91SAM9261 向 OLED 中发送命令信息; 如果 A0 为 1, 则 AT91SAM9261 向 OLED 中发送数据信息。即 OLED 控制器的命令地址起始于 0x40000000, OLED 控制器的数据地址起始于 0x40000001。

AT91SAM9261 的 PA19 控制 OLED 驱动器的复位操作, 在控制的时候, 首先需要将其设置为低电平, 然后延时 1ms 后, 再设置其为高电平, 即为 OLED 驱动器产生一个复位信号, 这需要在驱动程序中用软件的方法控制实现。

2 驱动程序的设计与实现

驱动程序是底层硬件和上层应用程序之间的中间层, 驱动程序屏蔽了底层硬件的不同, 使得上层的应用程序设计不必考虑硬件的不同, 从而使设计更为容易。驱动程序工作在内核空间, 本质是实现逻辑设备到物理设备的转换。要编写驱动程序, 首先要了解硬件及其工作原理; 其次要熟悉所用的操作系统, 本系统使用 Linux2.6。

Linux 支持 3 类硬件设备: 字符设备、块设备和网络设备。因为 OLED 为字符设备, 驱动程序应该按照字符设备驱动流程开发。具体流程包括: 注册与初始化过程, 与建立 OLED 驱动程序与应用程序的接口。

2.1 注册与初始化过程

注册与初始化过程包括: 通过 Probe 函数对驱动程序进行注册, 在 Probe 中设置 PA19、PA20 的配置, 控制 OLED 驱动器的复位和供电操作, 以便启动 OLED 控制器; 在 Probe 中设置总线的宽度、OLED 的命令和数据映射地址, 以便通过总线读写 OLED 的数据和命令。

(1) 通过 Probe 函数对驱动程序进行注册

首先通过 alloc_chrdev_region 动态分配 OLED 设备的编号; 其次, 通过 cdev 结构及其函数在 Linux2.6 操作系统中初始化和注册 oled 字符设备。在 Linux2.6 内核中, 一个字符设备用 cdev 结构来描述。定义如下:

```
struct cdev {
    struct kobject kobj;
    struct module *owner; // 所属模块
    const struct file_operations *ops; // 文件操作结构
    struct list_head list;
    dev_t dev; // 设备号, int 类型, 高 12 bit 为主设备号, 低 20 bit 为次设备号
    unsigned int count;
```

```

};

static int __init led_probe(void)
{
    .....
    if (led_major) {
        devid = MKDEV(led_major, led_minor);
        ret = register_chrdev_region(devid, 0,
"OLED");
    } else {
        ret = alloc_chrdev_region(&devid,
led_minor, 0, "OLED");
        led_major = MAJOR(devid);
    }
    led_cdev = cdev_alloc();
    led_cdev->owner = THIS_MODULE;
    led_cdev->ops = &led_fops;
    cdev_init(led_cdev, &led_fops);
    ret = cdev_add(led_cdev, devid, 1);
    .....
}

```

(2) 在 Probe 中设置 PA19、PA20 的配置, 初始化 OLED

下述程序设置 AT91SAM9261 的 GPIO 的一些特性。首先设置 PA19、PA20 为输出模式、上拉电阻方式; 其次设置 PA19 为低电平, 延迟 1 ms 后, 设置 PA19 为高电平, 最后设置 PA20 为高电平。又因为 OLED 驱动器的数据和命令由 8 bit 总线控制, 所以在 CS3 的控制寄存器 SMC_CTRL3 设置总线的宽度为 8 bit。

其中 AT91C_VA_BASE_PIOA 在 Linux2.6 内核的 `\include\asm-arm\arch-at91sam9261\at91sam9261.h` 中定义的 I/O 端口的虚拟地址, 它通过同一文件下定义的 AT91_IO_P2V 宏来完成物理 I/O 地址到虚拟地址 I/O 的映射, 即:

```

#define AT91C_IO_PHYS_BASE
0xFFFFF000
#define AT91C_IO_VIRT_BASE
VMALLOC_END
#define AT91_IO_P2V(x) ((x)-AT91C_IO_
PHYS_BASE+AT91C_IO_VIRT_BASE)
#define AT91C_VA_BASE_PIOA

```

AT91_IO_P2V(AT91C_BASE_PIOA)

```

    at91_gpio_configure(AT91C_VA_BASE_PIOA, 19, OUTPUT_MODE, PULLED_UP, NO_
GLITCH_FLT);
    at91_gpio_configure(AT91C_VA_BASE_PIOA, 20, OUTPUT_MODE, PULLED_UP,
NO_GLITCH_FLT);
    at91_gpio_set_level(AT91C_VA_BASE_PIOA, 19, 0);
    udelay(1000);
    at91_gpio_set_level(AT91C_VA_BASE_PIOA, 19, 1);
    at91_gpio_set_level(AT91C_VA_BASE_PIOA, 20, 1);
    writel( (SMC_READMODE | SMC_WRITE-
MODE | SMC_DBW_WI-DTH_EIGHTH_BITS),
AT91C_VA_BASE_EBI+SMC_CTRL3);

```

(3) 在 Probe 中设置 OLED 驱动器的命令和数据映射地址, 以便内核可以向总线发送数据和命令。

在 ARM 体系下, 外设的 I/O 内存资源的物理地址是已知的, 比如, 由前面的分析可知, OLED 控制器的命令地址空间起始于 0x40000000, OLED 控制器的数据地址空间起始于 0x40000001。但是, AT91SAM9261 并没有为这些外设的 I/O 内存资源预定义虚拟地址范围, 驱动程序不能直接访问这些物理地址, 必须将它们映射到核心虚拟地址空间内 (通过 `ioremap` 函数), 然后才能够访问核心虚拟地址空间。

2.2 建立 OLED 驱动程序与应用程序接口

建立 OLED 驱动程序与应用程序的接口, 以方便应用程序的开发, 这主要实现了以下几方面: 通过 `led_ioctl` 设置与应用程序的参数接口, 方便应用程序向驱动程序中传递这些参数; 通过 `led_write` 实现应用程序向驱动程序传送数据。

在设计驱动程序的时候, 必须要考虑应用程序需要向驱动程序传递什么参数, 并将这些参数考虑进去, 这样才能设计出好的驱动程序。本系统应用程序需要向驱动程序传递参数为: 显示点阵在屏幕中位置、颜色以及点阵的大小等。

Linux内核是通过一个file_operations的结构体来组织对字符设备操作的具体实现函数。

```
static const struct file_operations led_fops =
{
    .owner = THIS_MODULE,
    .read = led_read,
    .write = led_write,
    .ioctl = led_ioctl,
    .open = led_open,
};
```

(1) 通过led_ioctl设置与应用程序的参数接口,以方便应用程序向驱动程序中传递这些参数。

应用程序运行于用户地址空间,驱动程序运行于内核地址空间。为了将应用程序设置的参数拷贝到驱动程序中,需要使用函数copy_from_user,它可以将用户地址空间的数据拷贝到内核地址空间。以下程序中,通过将应用程序中设置的参数(位置信息、颜色、点阵的大小)拷贝到结构ShowPos,实现了从应用程序到驱动程序的参数传递,这些信息在OLED显示中用到。见下面的led_write中的处理。

```
struct ShowPos
{
    unsigned short xbegin; unsigned short xend;
    unsigned short ybegin; unsigned short yend;
    unsigned short color;   unsigned short num;};
struct ShowPos *pshowpos;
static int led_ioctl(struct inode *inode, struct
file *file,
                    unsigned int cmd, unsigned long
arg)
{
    int ret = 0;
    copy_from_user(pshowpos, (void __user
*)arg, sizeof(struct ShowPos));
    .....
```

(2) 通过led_write实现应用程序向驱动程序传送数据。

应用程序需要将汉字或者图形的点阵数据传送到驱动程序,由驱动程序显示点阵,这些点阵数据放在应用程序的用户地址空间buf缓冲区中,通过驱动程序的copy_from_user,将这些数据拷贝

到驱动程序的内核地址空间bufto缓冲区中,驱动程序再对bufto进行处理,显示汉字或者图形。最后,要记得释放bufto缓冲区。

3 应用实例

驱动程序屏蔽了底层硬件,使得应用程序的设计更为方便,简洁。本文在驱动程序的基础上,设计和实现了基于Linux2.6的OLED显示应用程序,使用C语言编写,数字和汉字的显示等可以通过调用驱动程序的接口函数(led_ioctl、led_write等)实现,非常方便,可以很好地应用在列车转向架故障诊断中,监测列车转向架故障和轴温,实现报警信息的显示和存储的功能。

3.1 报警信息的显示

报警信息的显示界面如图2。



图2 OLED在列车故障诊断中的应用

其中,系统默认界面显示列车的车号以及车顺号,操作按键共4个,依次为[上]、[下]、[进入]、[退出]。在系统默认界面按[进入]键,进入菜单选择界面,可以使用[上][下]键选择菜单,用[进入]键进入相应的界面,用[退出]键返回主界面(系统默认界面),其中,选择的菜单使用高亮背景。在设置车号界面,可以使用[上][下]键设置车顺号加1或者减1,直到符合要求;在查看温度界面,可以查看1 bit~8 bit轴温、1 bit~4 bit齿轮箱温度和车外温;在查看报警界面,可以用[上][下]键翻页查看本车厢的报警信息;在这些子界面(设置车号界面、查看温度界面、查看报警界面),可以使用[退出]键返回上级菜单(菜单选择界面)。

3.2 报警数据的存储

文章编号: 1005-8451 (2010) 06-0048-04

Simulink 环境下的扩频码序列仿真实验研究

张重阳

(西安铁路职业技术学院 电子信息系, 西安 710014)

摘要: 本文应用 Simulink 建立了常用扩频码序列 m 序列和 Gold 序列仿真模型, 模拟仿真数据与理论分析高度吻合, 证明应用 Simulink 进行常用扩频码序列仿真实验的可行性。Simulink 仿真实验环境可以作为 m 序列和 Gold 序列仿真实验的一种可靠的计算机模拟途径, 可以为 m 序列和 Gold 序列的设计与研究提供有效帮助, 节省人力、物力与实验周期, 且实验结果可靠, 实现了扩频码序列的实时精度仿真。

关键词: Matlab/Simulink; m 序列; Gold 序列; 仿真

中图分类号: TN929.533 **文献标识码:** A

Simulink environment simulation of spreading code sequences

ZHANG Chong-yang

(The Department of Electronic Information, Xi'an Railway Vocational Technical Institute, Xi'an 710014, China)

Abstract: This paper used Simulink to establish a sequence of spreading code sequences and Gold sequences m simulation model, simulation data, a high degree of agreement with the theoretical analysis, proved to be popular Simulink spreading code sequence of the feasibility of simulation. Simulink simulation environment could be used as m sequences and Gold sequences simulation of a reliable means of computer simulation, for m sequences and Gold sequences of design and research to provide effective help to save human, material and experimental period, and the experimental results and reliable achieved accuracy of spreading code sequences, real-time simulation.

Key words: Matlab/Simulink; m sequence; Gold sequence; Simulation

MATLAB 是当今国际上科学界最具影响力, 应用最广泛的也是最有活力的科学计算软件, 它提供了强大的数值运算和符号运算功能, 并提供了面

向用户的可视化操作功能, 而其最吸引用户的功能是它的动态仿真集成环境 Simulink。Simulink 在 MATLAB 强大的数值计算功能基础上, 提供了一个建立模型方块图的图形用户接口 (GUI), Simulink 中包含许多不同功能的模块库, 如 Sources (输

收稿日期: 2009-04-17

作者简介: 张重阳, 副教授。

因为 CF 卡存储容量大, 现在 4G 的 CF 卡已经被普遍使用, 所以本系统使用 CF 卡保存车厢级的信息, 包括车顺号、报警信息等。可以手工修改车顺号, 并且保存在 CF 卡上, 在设置车号界面, 如果用[进入]键, 即可将设置的车顺号保存在 CF 卡上。通过读取 CF 卡上的车顺号信息, 自动修改 OLED 显示屏上的车顺号; 报警信息也可以保存在 CF 卡上, 方便列车运行过程乘务员的观察以及列车到站后的查询。

4 结束语

本文介绍了 AT91SAM9261 与 OLED 显示模

块连接电路的设计, 以及在嵌入式 Linux2.6 操作系统下 OLED 驱动程序的编写, 最后介绍了 OLED 驱动程序在列车故障诊断中的应用。本系统硬件选用工业级的芯片, 设计高效、可靠、稳定; 软件选用的 Linux2.6 操作系统又具有源码公开、多用户、多任务、实时等优良性能, 因此在车载控制系统中具有广阔的应用前景。

参考文献:

- [1] 严隽堃. 车辆工程[M]. 北京: 中国铁道出版社, 2004.
- [2] 刘 森. 嵌入式系统接口设计与 linux 驱动程序开发[M]. 北京: 航空航天大学出版社, 2006.