

基于 OMAP-L138 平台 SPI 总线的嵌入式应用*

于森 郑楠 崔洋 李佩玥

(中国科学院长春光学精密机械与物理研究所应用光学国家重点实验室超精密光学工程研究中心 长春 130033)

摘要: 数据的采集和控制是嵌入式应用领域的重点课题,随着数字转模拟器件精度的扩展和转换速率的提升,需要更加高效的运算平台和方法。PCM1702 是一款高达 20 位分辨率的精密数字模拟转换器,其还具备超低失真、低噪声和快速电流输出等特性。SPI 总线接口是嵌入式 Linux 系统中的典型应用,在高性能达芬奇平台 OMAP-L138 处理器 ARM+DSP 双核环境下,通过 GPIO 实现 SPI 总线驱动,在高精度 DA 芯片 PCM1702 控制中得到应用。实验结果表明该驱动实现方法高效、稳定,具备针对精密器件的高速处理能力。

关键词: OMAP-L138; SPI; PCM1702; 嵌入式 Linux; 驱动

中图分类号: TN721.1 **文献标识码:** A **国家标准学科分类代码:** 510.1010

Embedded application of SPI bus based on OMAP-L138 platform

Yu Miao Zheng Nan Cui Yang Li Peiyue

(Engineering Research Center of Extreme Precision Optics, State Key Laboratory of Applied Optics, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, Changchun, 130033, China)

Abstract: Data acquisition and control is the key issue for embedded applications. With the expansion of the precision and the promotion of the conversion speed used in the digital to analog devices, we need more efficient computing platforms and methods. The PCM1702 is a precision 20-bit digital-to-analog converter. It also features a very low noise, ultra-low distortion and fast settling current output. SPI bus interface is the typical application of embedded Linux system. In this paper, SPI bus driver is achieved through the general purpose I/O pins based on DSP and ARM dual core processor on high-performance DaVinci OMAP-L138 platform, which is applied in high-precision DA chip PCM1702. The experimental results show that the method is efficient and stable, which is aimed at the high processing capacity of precision components.

Keywords: OMAP-L138; SPI; PCM1702; embedded Linux system; driver

1 引言

嵌入式 Linux 是以 Linux 内核为基础,经剪裁、定制适用于特定嵌入式应用场合的专用系统,由于其高效性、移植性、稳定性等特点,已经广泛应用于工业控制、仪器设备、安防通信、消费电子等产业。随着数字信号技术领域的蓬勃发展,嵌入式 Linux 与 DSP 相关平台结合成为研究热点。

OMAP-L138^[1]是一款基于 TI 公司达芬奇(DaVinci)平台^[2]的低功耗嵌入式应用处理器,其最大特点在于使用 DSP 与 ARM 结合的非对称多核处理架构,兼具 DSP 的高性能数字信号处理能力与 ARM 精简指令集下的丰富接口

扩展。

SPI 总线^[3]是嵌入式 Linux 系统中串行总线设备的典型应用,全双工主从结构的设备接口,由串行数据输入(MISO)、串行数据输出(MOSI)、串行移位时钟(SCK)、从使能信号(SS)组成标准的收发时序。

目前,国内外基于 SPI 总线的串行通信应用案例主要使用传统的单片机或是 ARM 平台^[4],随着嵌入式应用场合的复杂化,主控平台的性能需要更新提高,例如高主频、低功耗的达芬奇双核平台。文献[5]提出针对 WinCE 系统平台下 SPI 总线驱动的设计方法,并且在喷头控制中得以成功使用。文献[6-7]分别提出 SPI 总线在 OLED 和 flash 控

收稿日期:2015-05

* 基金项目:国家重大科技专项 02 专题(2009ZX02205)项目

制方面的实现方法,而针对 SPI 总线串行 A/D、D/A 转换芯片控制则不具备丰富的案例。结合 OMAP-L138 异构双核硬件平台,搭建以 Ubuntu 10.04 为上位机系统,利用 arm-none-linux-gnueabi 为编译工具的交叉开发环境,在嵌入式 Linux 中实现针对于高精度 DA 控制器 PCM1702 的 SPI 总线驱动及应用。

2 平台简介

2.1 OMAP-L138

OMAP-L138 采用 300 MHz ARM926EJ-S 32 位 CPU 和 300 MHz C6748 VLIW DSP 核结构,拥有丰富的片上外设资源,支持最多 144 pin 的 GPIO 复用接口以及两路标准 SPI 接口。

尽管 OMAP-L138 平台提供两个标准 SPI 总线接口,但是由于在许多应用场合下,例如 LCD Controller、SPI Flash、EEPROM、RTC 等器件的必要占用,以及众多 SPI 接口芯片的自定义时序要求,因此采用嵌入式 Linux 系统下 GPIO 引脚产生 SPI 总线驱动信号的方法,可以提高移植性。

2.2 DA 芯片

PCM1702^[8]是 TI 公司的一款 20 bit 分辨率的数字-模拟转换器,因其器件精密性、超低噪声和快速稳定输出等特点而广泛应用于高端消费类和专业的数字音频领域。

由于双极性输出特性,PCM1702 采用二进制补码

(BTC)的数据输入形式进行编码。起始于数据最高有效位(MSB),以最低有效位(LSB)为终止。

如图 1 所示,在稳定连续的时钟 CLOCK 条件下,由 OMAP-L138 的 GPIO 引脚给出串行比特流做为数字输入。当锁存使能信号(LATCH ENABLE)出现下降沿开始,触发时钟的每个上升沿采集 1 个比特位,根据芯片特性,累计采集 20 个比特位做为有效数据传入并行寄存器完成 D/A 转化。直至时钟延续至少第 24 个比特位的上升沿,模拟信号才可以完成上一次的 D/A 转化输出结果至芯片引脚。锁存使能信号必须保持低电平,直至下一个 20 位有效数据的第一个时钟周期。

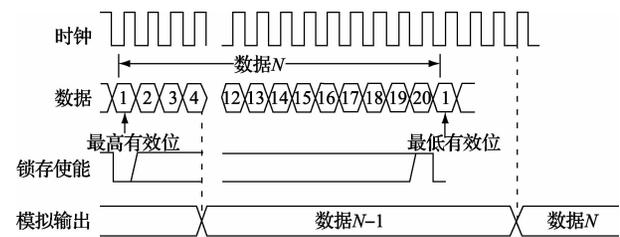


图 1 PCM1702 控制时序

2.3 硬件连接

PCM1702^[9]的引脚主要包括双极性供电以及公共端以外,是通过时钟(CLOCK)、数据(DATA)、锁存使能(LATCH ENABLE)三线引脚以约定的工业标准串行输入时序格式配合产生模拟输出(Iout)信号。如图 2 所示,OMAP-L138 复用 3 个 GPIO 与器件连接。

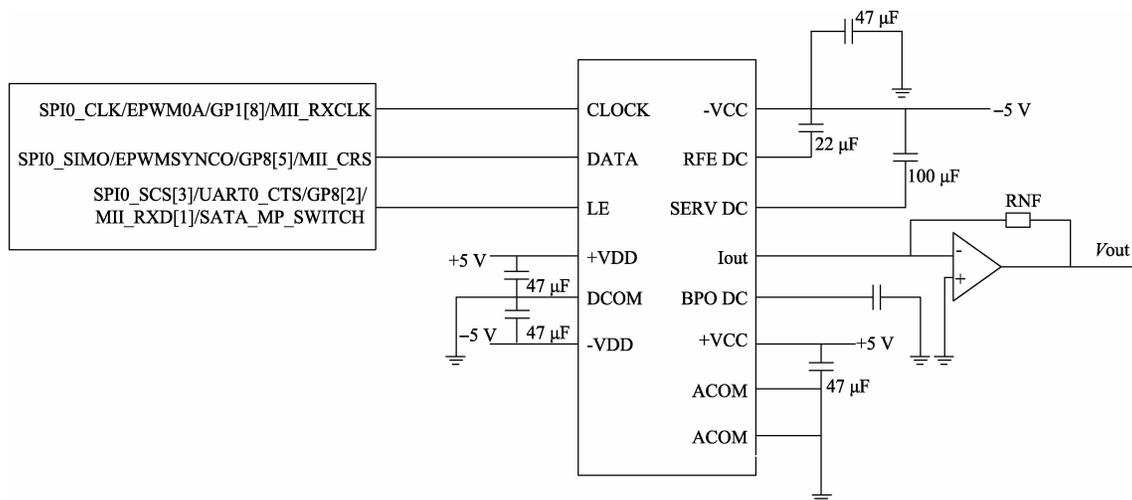


图 2 OMAP-L138 与 PCM1702 原理

3 嵌入式 Linux 设备 SPI 总线驱动

Linux 系统根据外设的特点,包含字符设备、块设备和网络设备 3 种抽象类型。其中字符设备是指面向字节数据流,读取数据需要顺序执行的设备的抽象,例如 UART 串口、SPI、鼠标、LED 等^[10]。

嵌入式 Linux 设备驱动程序是应用程序与设备硬件的接口。通过设备驱动程序,既可以使硬件设备被操作系统有效识别,也可以满足用户通过应用程序控制底层。由于具体嵌入式 Linux 应用芯片的差异化,需要定制不同的驱动程序^[11]。

Linux 操作系统将所有硬件设备抽象为设备文件^[12],

应用程序在用户空间利用文件操作调用内核空间的驱动影响底层硬件。

为了良好的一致性和可移植性, Linux 驱动结构框架中需要定义 file_operation 结构体, 其中定义了驱动操作的重要函数。

```

/* 文件操作结构体 */
static const struct file_operations SPIGPIO_fops =
{
    .owner = THIS_MODULE,
    .ioctl = SPIGPIO_ioctl, /* 控制函数 */
    .read = SPIGPIO_read, /* 读设备函数 */
    .write = SPIGPIO_write, /* 写设备函数 */
    .open = SPIGPIO_open, /* 打开设备函数 */
    .release = SPIGPIO_release, /* 释放设备函数 */
};

```

3.1 SPI 设备加载和卸载函数

根据 Linux 驱动框架结构, 加载函数主要实现设备号的申请和字符结构体的注册, 以及 PCM1702 的初始化。卸载函数则相反。如下段代码示例, 设备号静态申请后, 完成设备节点创建与自动系统挂载, 避免 mknod、insmod 等命令的重复操作。

```

dev_t devno = MKDEV(SPIGPIO_major, 0); /*
构建设备号 */
if (SPIGPIO_major) /* 以静态申请为例, 申请设备号 */
    result = register_chrdev_region(devno, 1, DEV_NAME);
    spigpio_class = class_create(THIS_MODULE, DEV_NAME);
    /* 创建 GPIO 这个类 */
    device_create(spigpio_class, NULL, MKDEV(SPIGPIO_major, 0), NULL, DEV_NAME);
    /* 创建 dev/DEV_NAME 设备节点 */
    cdev_init(&dev->cdev, &SPIGPIO_fops);
    /* 初始化 cdev 设备 */
    err = cdev_add(&dev->cdev, devno, 1); /* 将 cdev 注册到系统中 */

```

如图 3 所示, OMAP-L138 在操作 GPIO 引脚时首先需要识别 IO 所在系统配置寄存器 SYSCFG 地址与偏移量, 根据具体的 PINMUX 复用规则定义引脚功能与方向特性。

```

// 初始化设备实例
PINMUX_REG = ioread32(IO_ADDRESS(SYSCFG_BASE)+CLOCK_PINMUX_OFFSET);
PINMUX_REG = (PINMUX_REG&0xf0ff0ff0) | 0x04004004;

```

//配置 3 个引脚为 GPIO 引脚

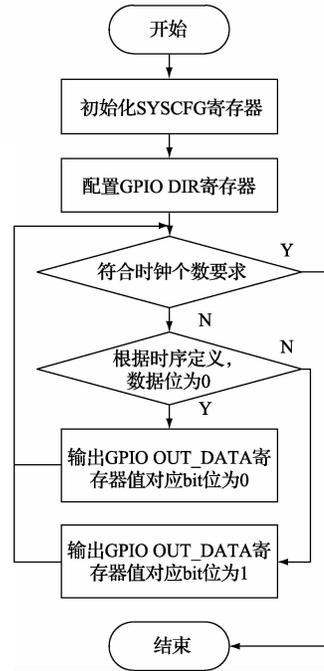


图 3 OMAP-L138 GPIO 控制流程

```

iowrite32(PINMUX_REG, IO_ADDRESS(SYSCFG_BASE)+CLOCK_PINMUX_OFFSET);

```

// 以 CLOCK 引脚为例, 配置引脚操作方向为输出, 初始化为高电平有效

```

CLOCK_REG = ioread32(IO_ADDRESS(CLOCK_GPIO_DIRECT));
// 读取当前 IO 寄存器值
iowrite32(CLOCK_REG&~CLOCK_GPIO_MASK, IO_ADDRESS(CLOCK_GPIO_DIRECT));
CLOCK_REG = ioread32(IO_ADDRESS(CLOCK_GPIO_DATA));
iowrite32(CLOCK_REG | ~CLOCK_GPIO_MASK, IO_ADDRESS(CLOCK_GPIO_DATA));

```

3.2 SPI 设备读写函数

PCM1702 作为数字-模拟转换器, 主要应用为用户对设备的写操作。因此 SPI 设备驱动写函数首先从用户空间将输入数据拷贝至内核空间, 按照芯片约定比特位模拟产生输入时序。

SPIGPIO_write 代码示例如下:

```

unsigned int spidata = 0;
char spibuf[4];
if (copy_from_user(spibuf, buf, 4))
// 用户空间 4 byte 数据拷贝至内核空间
ret = -EFAULT;
else

```

```

{
    for(i = 3;i>0;i--) //调整字节序,保证前 20 bit
    数据有效
    {
        spidata |= spibuf[i];
        spidata = spidata <<8;
    }
    spidata |= spibuf[0];
    if(write_spi(spidata))
    ret = - EFAULT;
}

```

3.3 SPI 时序生成

如图 1、图 3 所示,通过时序定义传入驱动输入数据,配置 CLOCK 和 LE 的起止电平,然后根据时钟个数要求,输出 DATA 对应 GPIO 数据位的高低信号,当数据位为 1 时,GPIOOUT_DATA 寄存器对应位赋值为高,否则寄存器对应位为低。伪代码如下所示,

```

static int ret,tmp; // 返回值
static ssize_t write_spi(int data)
{
    int spi_bit_count = MAX_SPI_BIT_COUNT;
    //根据时序必须大于 24 bit
    tmp = data;
    set_pin_value(0, IO_ADDRESS(LE_GPIO_DATA)); //LE 引脚为低
    while(spi_bit_count)
    {
        set_pin_value(0, IO_ADDRESS(CLOCK_GPIO_DATA)); //CLOCK 引脚为低 ndelay(5); //根据
        时序延时
        if((tmp & 0x80000000) == 0x80000000)
        { set_pin_value(1, IO_ADDRESS(DATA_GPIO_DATA)); //DATA 引脚为高 ndelay(15);
        }
        else{ set_pin_value(0, IO_ADDRESS(DATA_GPIO_DATA)); //DATA 引脚为低 ndelay(15);
        }
        set_pin_value(1, IO_ADDRESS(CLOCK_GPIO_DATA)); //CLOCK 引脚为高 ndelay(20);
        tmp = tmp<<1;
        --spi_bit_count;
    }
    set_pin_value(1, IO_ADDRESS(LE_GPIO_DATA)); //LE 引脚为高
    return ret;
}

```

3.4 应用测试

将上述驱动在上位宿主主机 Ubuntu10.04 通过 arm-

none-linux-gnueabi 交叉工具链进行编译成功后,需要将编译生成文件加入内核,重新进行 make menuconfig 形成新的 uImage 镜像。成功烧写镜像文件至 nandflash 后,利用硬件电路通电启动才可以在文件系统/dev 目录下看到自动挂载的设备文件节点^[13],该节点与前文所述 DEV_NAME 同名。

利用嵌入式 Linux 文件操作方法,首先通过 open 操作打开 DEV_NAME 设备文件节点,成功后利用 write(fd,spidata,4)操作将数据 spidata 写入 SPI 设备,观察输出效果,最后根据 PCM1702 时序设计形成完善的测试逻辑。

如图 4 所示,示波器采集 D/A 输出信号时序,1 通道(底部信号)为 LE 信号,2 通道(中部信号)为 CLOCK 信号,4 通道(顶部信号)为 DATA 信号。通过与图 1 对比,示波器采集波形结果与 PCM1702 器件手册定义时序完全一致,可说明完整实现器件功能,并以典型 20 bit 测试输入数据 0x55555 为例,图 4 中 LE 信号的下降沿作为采集过程起始信号,上升沿作为采集过程完成信号,CLOCK 信号每个时钟上升沿作为触发信号,测试数据具备间隔出现高低信号的明显特点,可以准确采集到输入数据 DATA 信号进行 DAC 转换。经测定,该驱动程序在自定义 CLOCK 信号 1.25 MHz 高速时钟条件下,稳定将输入数据进行模拟量输出。

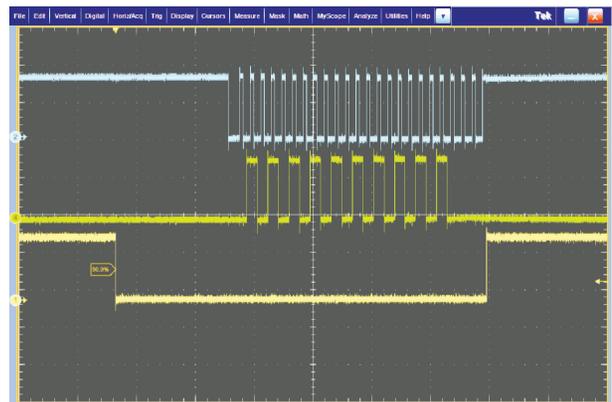


图 4 PCM1702 示波器测试效果

如图 5 所示,PCM1702 的输出电流转换成±10 V 电压后的正弦波形,可以满足在全量程内的输出控制。经过调试最小输出间隔分析结果得知,PCM1702 器件在±10 V 电压输出的理论分辨率最高可达 19 μV,然而本次实验测试电路通过示波器测量,在稳定环境下,符合电压输出最高峰值的最低值大于最低峰值的最高值判定标准,最高达到 100 μV 的分辨率效果,该结果适用于大多数精密输出控制场合。

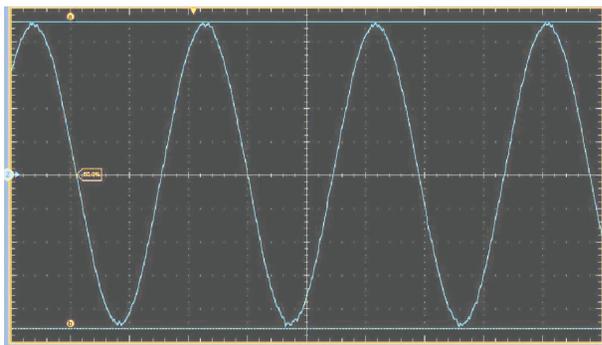


图5 PCM1702 电压正弦输出

4 结论

由于串行 SPI 总线在嵌入式环境下的广泛应用需求,在低功耗 OMAP-L138 异构双核 ARM+DSP 平台中移植嵌入式 Linux 操作系统,基于编程实现 GPIO 模拟 SPI 总线操作的驱动程序,并成功测试在 20 bit 分辨率高精度 DA 芯片 PCM1702 应用场合。结果表明该方法可以高速、稳定的完成用户对器件的操作,更加有效地发挥出器件的特性,对类似的串行驱动开发提出通用的思路。

参考文献

[1] Texas Instruments. OMAP-L138 DSP + ARM processor technical reference manual (Rev. A)[EB/OL]. (2011-11), <http://www.ti.com.cn/cn/lit/ug/spruh77a/spruh77a.pdf>.

[2] 宋执环,杜往泽,李斌,等.基于图像检测的除尘风机嵌入式控制系统[J].仪器仪表学报,2014,35(5):1192-1200.

[3] 付兴武,张军,王洋.基于 SPI 总线协议的字符设备驱动程序[J].计算机系统应用,2013,22(2):146-150.

[4] 杨斌,满峰,姜秀杰,等.基于 ARM-Linux 和单片机的数据通信系统设计[J].电子设计工程,2013,21(12):126-129.

[5] 王卫民,赵天环,曹思源.基于 ARM9 的高解析喷码机系统设计与实现[J].电子测量技术,2013,36(1):70-74.

[6] 邢艳芳,张延冬.基于 Zynq 的 OLED 驱动设计[J].液晶与显示,2014,29(2):224-228.

[7] 杨立身,张安伟,王磊,等.基于 STM32 的 μ C/GUI 外置 spiflas 字库研究与实现[J].液晶与显示,2015,30(2):290-295.

[8] BiCMOS Advanced Sign Magnitude 20-Bit DIGITAL-TO-ANALOG CONVERTER. <http://www.ti.com.cn/cn/lit/ds/symlink/pcm1702.pdf>.

[9] 周静雷,贾保军,张宏艳.基于以太网的声频数据实时传输[J].电子测量技术,2011,34(5):107-110.

[10] 怯肇乾,陈永超. ARM-Linux 下 SPI 设备的添加与驱动实现[J].单片机与嵌入式系统应用,2012,12(4):80-81.

[11] 李彦,李镛.基于嵌入式 Linux 系统的双网卡大数据传输[J].电子测量与仪器学报,2014,28(9):1027-1032.

[12] 黄帅,程良伦.基于 OMAP-L138 平台多通道手持频谱分析仪研究[J].电子设计工程,2014,22(11):130-133.

[13] 徐立松,李佩玥,葛川,等.高精度 D/A 电路的设计与实现[J].国外电子测量技术,2014,33(12):21-25.

作者简介

于淼,1985 年出生,工学硕士,助理研究员。主要研究方向为嵌入式系统开发、精密控制等。
E-mail:miao_miaoyu@sina.com