

高速公路视频监控下基于 8250 串口的摄像头控制系统^①

廖建尚

(广东交通职业技术学院 计算机工程学院, 广州 510650)

摘要: 分析了 8250 串口的工作原理和通信协议以及 Linux 的串口驱动程序, 构建了高速公路视频监控系统下基于串口控制的摄像头控制系统, 完成了基于 Linux 的 8250 串口的驱动程序设计、串口的应用程序设计和摄像头控制应用程序设计, 实现了摄像头的控制, 并应用于高速公路视频监控嵌入式系统中, 具有良好的效果和较高的社会应用价值.

关键词: 视频监控; 8250; 串口; 摄像头; 驱动

Camera Control System Based on 8250 under the Freeway Video Surveillance

LIAO Jian-Shang

(Computer Engineering, Guangdong Communication Polytechnic, Guangzhou 510650, China)

Abstract: This paper analyzes the working principle and communication protocol of 8250 serial and driver of 8250 in Linux. Then it builds the camera control system which is based on serial under the video surveillance system on freeway. It completes serial driver design in the Linux, serial application design and application design of control on camera. It implements the control of camera, which is used in video surveillance system on freeway. So, it has good results and high social value.

Key words: video surveillance; 8250; serial; camera; driver

1 引言

近年来, 随着国内视频技术发展, 网络环境稳定性的提高, 视频监控业务需求越来越大, 其应用领域也越来越广泛, 社会的发展, 汽车增量不断增多, 道路上各种违章的车辆也逐渐增多, 为了构建一个平安交通和智能交通, 视频监控系统在交通行业的应用也越来越多, 目前高速公路存在以下现状^[1].

目前有多种方式用于摄像头控制, 可以用 I2C 或 I2S 等总线控制, 由于用于摄像头的控制的数据量非常少, 本设计采用了串口的通信方式, 8250 串口数据传输速度不快, 但是足够使用, 也实现了设计的成本最小化.

本文将研究详细分析串口的工作原理和通信协议和 Linux 的串口以及 TTY 的驱动程序, 设计一个应用在高速公路视频监控的基于串口的摄像头控制系统.

2 编码系统硬件设计

根据高速公路视频监控的需求, 高清视频 H264 编码硬件平台设计选用 DM368, 硬件平台主要由 ARM9 处理器、视频输入输出处理子系统 VPSS、视频硬件编码器 HDVICP 以及 DSP 组成, 采用 512M 的 DDR2, 人机交互硬件处理器是 ARM9, 相关控制硬件有云台、时钟、摄像头调焦等硬件构成如图 1 所示^[2,3].

因此硬件平台设计中需要的外围接口有串口、SPI、GPIO、I2C 和网络等外围接口, 其中串口设计开发时进行终端调试和摄像头的焦距调节, 完成 SPI 与 Cortex M3 的通信设计, Cortex M3 处理器设计为环境物理信息监测及报警子系统的控制单元, GPIO 实现看门狗的驱动, 网络实现所有音视频数据的传输和相关控制命令的传输, 人机交互设备控制全部由 ARM9 和其运行的操作系统 Linux 完成.

① 基金项目:2013 年广东省科技计划项目;科技部 2014 星火计划项目(国科发计[2014]303 号);2014 年广东交通职业技术学院科研课题

收稿时间:2015-03-02;收到修改稿时间:2015-04-15

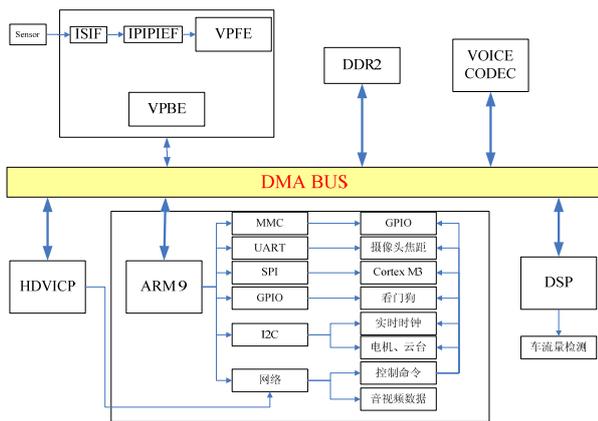


图 1 硬件设计图

3 串口 8250 协议以及移植

3.1 串口 8250 的通信协议^[4,5,7]

8250 是专用于串行通信的可编程串行接口芯片，具有较强的串行通信能力和灵活的可编程性能，在微机中应用广泛。

(1)发送数据。处理器将要发送的数据写到 8250 的发送数据寄存器 THR，当发送移位寄存器 TSR 中的数据全部移出变空时，存于 THR 中待发送的数据将会自动发送并行送到 TSR，TSR 在发送时钟的控制下，按照设定的传送方式，加上起始位，奇偶校验位和停止位，再以设定的波特率由发送端发送出去。发送完毕后，建立已发送数据的状态位，利用该状态触发的中断即可实现数据的连续发送。

(2)接收数据。接收时钟的作用下，通过输入端逐位进入接收移位寄存器 RSR。RSR 根据初始化时定义的数据位数确定接收到了一个完整的数据后会立即将数据传送到接收数据缓冲寄存器 RBR，RBR 收到 RSR 的数据后，建立已接收数据的状态，利用该状态触发的中断即可实现数据的连续接收。

8250 串口可以实现目前广泛应用的 RS-232 通信协议，标准的 RS232 接口已成为计算机、外设、交换机和许多通讯设备的标准接口，同时 RS232 标准广泛应用于微型计算机系统和大型系统中，RS-232 标准有连线简单、通讯距离长等优点。本课题采用的 FCB-EX1010 摄像头的通信协议 VISCA 协议也是在 RS-232 的基础上实现的一种通信协议。

3.2 Linux 系统中 TTY 的结构^[5]

3.2.1 TTY 的构成

视频监控运行 Linux 操作系统，其中系统通

过串口控制摄像头。在 Linux 系统中，终端是一种字符型设备，它包含控制台、串口、伪终端 3 类设备终端，通常使用 TTY 来简称各种类型的终端设备。TTY 是 Teletype 的缩写，Teletype 是 Teletype 公司生产最早出现的一种终端设备。下面介绍串行终端和控制台终端：

(1)串行终端。串行端口终端是使用处理器串行端口连接的终端设备。处理器把每个串行端口都看作是一个字符设备。这些串行端口所对应的设备名称是 /dev/ttyS0、/dev/ttyS1 等等。在命令行上把标准输出重定向到端口对应的设备文件名上就可以通过该端口发送数据，或者采用文件 IO 读写也可以进行数据发送。

(2)控制台终端。控制台终端是当前进程的控制终端的设备特殊文件。并且有一些设备特殊文件与之相关联：tty0、tty1、tty2 等。当用户在控制台上登录时，使用的是 tty1，可以切换到 tty2、tty3 等。在 Linux 中，可以在系统启动命令行里指定当前的输出终端，用户可以在内核命令行中同时设定多个终端，这样输出将会在所有的终端上显示。

3.2.2 Linux 驱动中串口数据在 TTY 中传送

TTY 在 Linux 的中由 TTY 核心、线程控制程序构成，串口在 TTY 中的数据传送有串口数据发送和串口数据接收构成，发送数据流程如图 2 所示。

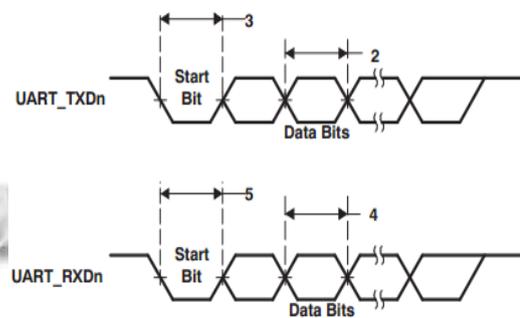


图 2 串口的数据帧格式

(1)串口发送数据的流程为：TTY 核心从一个用户获取将要发送给一个 TTY 设备的数据，TTY 核心将数据传递给 TTY 线路规程驱动，接着数据被传递到 TTY 驱动，TTY 驱动将数据转换为可以发送给硬件的格式。

(2)串口接收数据的流程为：从 TTY 硬件接收到的数据传送给 TTY 驱动，进入 TTY 线路规程驱动，再进入 TTY 核心，大多数时候 TTY 核心和 TTY 之间的数据传输会通过 TTY 线路规程的转换，但是 TTY 驱动与 TTY 核心之间也可以直接传输数据。

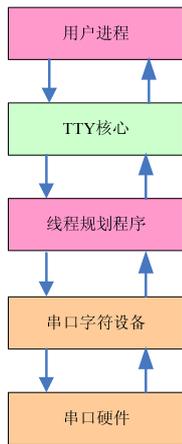


图3 TTY的数据发送流程图

```

<-> 8250/16550 and compatible serial support
[*] Console on 8250/16550 and compatible serial port
(2) Maximum number of 8250/16550 serial ports
(2) Number of 8250/16550 serial ports to register at runtime
[*] Extended 8250/16550 serial driver options
[*] Support more than 4 legacy serial ports
[*] Support for sharing serial interrupts
[*] Autodetect IRQ on standard ports (unsafe)
[ ] Support RSA serial ports
*** Non-8250 serial port support ***
<-> MAX3100 support
  
```

图4 8250 驱动移植图

3.3 8250 串口驱动移植^[5,9]

在 DM368 中采用了 8250 的串口终端，在 Linux2.6.32 中，有两种方式可以移植 8250，其中一种是通过图形界面添加的方式，在 Linux 源代码上采用 make menuconfig，在 X 界面上直接选择。

在图4的界面中，选择进入“Serial driver”，里面就是 8250 的串口配置，将 8250 驱动的相应选项选择上重新编译 Linux 内核，其中 DM368 支持 UART0 和 UART1, UART1;

驱动移植后，就可以和操作字符设备那样在用户空间利用 read 和 write 访问串口了。

4 FCB-EX1010摄像头工作原理

4.1 FCB-EX1010 摄像头功能^[8]

FCB-EX1020 彩色一体化摄像组件，支持逐行扫描技术，配备数字输出接口，支持增强型数字降噪功能，支持全新技术的宽动态功能：自动宽动态、隔行模式宽动态和逐行模式宽动态，支持图像防抖动功能高分辨率，支持高级白平衡模式，支持温度读取功能，支持慢速 AE 响应功能，镜头 36 倍光学变焦，数字 12

倍变焦，支持白平衡，慢速自动曝光响应，支持宽动态范围，有背光补偿和区域遮蔽等功能，支持视频输出 YCbCr(4:2:2)(ITU-R BT656 标准)，最高 38.4Kb/s 高速数字串行端口，使用 VISCA 协议控制。

4.2 VISCA 协议分析^[8]

VISCA 协议是 SONY 公司的专用摄像机控制协议，在串口 RS232 的通信协议上实现的一种控制协议，遵循 RS-232 的串口标准。主要实现对摄像头的焦点控制、焦距控制、光圈调节、图像防抖动、背光补偿以及区域遮挡报警等等。

VISCA 协议主要是定义了一系列的控制命令来控制摄像头，VISCA 通讯的基本单位称为数据帧。图2为数据帧格式，其中一个数据帧可以是3到10个字节，出去报头和结束标志，信息位可以使1到14个字节，数据帧的第一个字节被称为报头，并且包括发送者和接收者的地址。例如，发送到从控制器(地址0)的FCB分配相机地址为1中的数据包的报头是0x81H，第二个控制摄像机地址是82H。在命令列表中，标题为8XH，输入摄像机在X上的地址，从所分配的相机地址的应答数据包的第一个地址为90H。从相机包分配地址2A0H。VISCA支持广播协议，如果广播，报头为88H，数据帧的结束标志为FFH，它表示数据帧的结束。

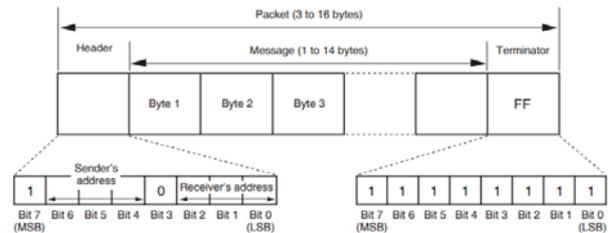


图5 VISCA 数据帧结构图

本系统主要实现对摄像头的焦点控制、焦距控制、光圈调节、图像防抖动、背光补偿以及区域遮挡报警等等。下面举一个焦点控制详细介绍 VISCA 的控制原理。

焦点控制有以下模式，自动对焦模式，正常自动对焦模式，间隔自动对焦模式，缩放触发模式，手动对焦模式，单键触发模式，无穷远模式，接近极限模式，所有这些都可以使用 VISCA 命令来设置，在手动模式中，可以实现变级调节，共 8 级，分别是 0-7，命令中除了报头 8x 和结束标志 FF 外，其他都是命令消

息,不同的命令有不同消息,具体如表 1 所示.

表 1 Focus 的调节命令功能表

功能	命令
停止	8x 01 04 08 00 FF
调远(标准)	8x 01 04 08 02 FF
调近(标准)	8x 01 04 08 03 FF
调远(可变)	8x 01 04 08 2p FF
调近(可变)	8x 01 04 08 3p FF
自动模式	8x 01 04 38 02 FF
手动模式	8x 01 04 38 03 FF

命令利用 VISCA 的协议,通过上位机发送命令给摄像头,就可以实现不同的功能,其中焦点 8 级可调是主要调节 P 参数.

5 软件设计^[9-12]

云台控制其中有一部分是摄像头的调节,本课题研究采用图像界面实现人机交互,其中界面设计采用 QT 图形界面软件来设计,可以在界面操作实现焦点控制、焦距控制、光圈调节、图像防抖动、背光补偿以及区域遮挡报警,如图 4 所示.重点围绕 VISCA 协议,实现软件设计,其中部分函数接口如下:

- (1)自动调节焦点,函数接口为 FocusAuto();
- (2)手动调节焦点,函数接口为 FocusManual();
- (3)停止调节焦点,函数接口为 FocusStop();
- (4)焦点变级调近,函数接口为 FocusNear();
- (5)焦点变级调远,函数接口为 FocusFar();

下面通过对焦点调节的 8 级可变调节来说明软件的实现过程,首先用数组完成 VISCA 控制命令的定义,在函数中分别用传入参数代码控制的级别,通过调用 WriteUart 函数完成命令的传送和控制,详细代码如下所示:

```
static char CAM_Far_0 []
= {0x81,0x01,0x04,0x08,0x20,0xFF};
static char CAM_Far_1 []
= {0x81,0x01,0x04,0x08,0x21,0xFF};
```

```
FocusFar(int ptr) {
    unsigned int ret = -1;
    switch(ptr) {
        case 0:ret = WriteUart(CAM_Far_0,
sizeof(CAM_Far_0));break;
        case 1:ret = WriteUart(CAM_Far_1,
```

```
sizeof(CAM_Far_1));break;
        case 2:ret = WriteUart(CAM_Far_2,
sizeof(CAM_Far_2));break;
        case 3:ret = WriteUart(CAM_Far_3,
sizeof(CAM_Far_3));break;
        case 4:ret = WriteUart(CAM_Far_4,
sizeof(CAM_Far_4));break;
        case 5:ret = WriteUart(CAM_Far_5,
sizeof(CAM_Far_5));break;
        case 6:ret = WriteUart(CAM_Far_6,
sizeof(CAM_Far_6));break;
        case 7:ret = WriteUart(CAM_Far_7,
sizeof(CAM_Far_7));break;
        default:ret = -1;break;
    }
    return ret;
}
```

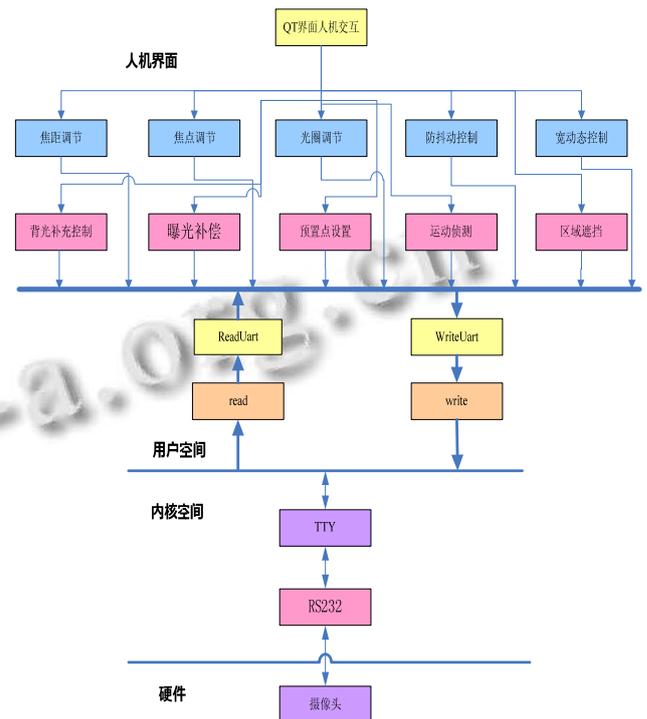


图 6 软件设计图

其中 WriteUart 实现了对写 8250 串口的函数封装,通过 read 和 write 文件 IO 操作发送 VISCA 协议的命令,达到控制摄像头的目的,代码如下:

```
WriteUart(char *buf, int n){
```

```
int ret = -1;
unsigned char str_ret[16];
write(fd_8250,buf,nbytes);
msleep(100);
memset(str_ret, 0, sizeof(str_ret));
ret = ReadUart(fd_8250 sizeof(str_ret));
return ret;
}
```

6 测试结果

通过实现应用层和底层软件设计,实现了 VISCA 协议的封装和应用,能够正常控制摄像头的焦点、焦距、光圈调节、图像防抖动、背光补偿等功能,达到了预期的目的。

7 结语

本系统设计实现了一款基于 TMS320DM368 的高清视频监控系统中的云台电机控制设计,完成了 8250 串口驱动程序的分析以及移植,并且完成了串口应用程序的控制设计和基于 VISCA 协议的摄像头控制应用程序设计,达到了应用目的,取得了良好效果。

参考文献

- 1 张志.高速公路高清视频监控系统的构建.中国交通信息化,2011,4:99-102.
- 2 邓旻熙.基于 DM365 的智能视频分析系统的设计与实现[硕士学位论文].成都:电子科技大学,2011.

- 3 Texas Instruments Incorporated. TMS320DM368 digital media system-on-chip(DMSoC). <http://www.ti.com/lit/ds/symlink/tms320dm368.pdf>. 2014-03.
- 4 Kevin R. Bulgrien RS-232 Support for the IBM PC[Thesis]. LeTourneau College Microcomputer Services, 1989.
- 5 耿杰恒,王竹林,贾春宁.基于 ARM9 和嵌入式 Linux 的串口驱动开发.科学技术与工程,2008,3:786-789.
- 6 TMS320DM36x Digital Media System-on-Chip (DMSoC) Universal Asynchronous Receiver/Transmitter (UART) User's Guide.
- 7 葛磊蛟,毛一之,等.基于 C 语言的 RS232 串行接口通信实现.河北工业大学学报,2008,6:11-16.
- 8 FCB-EX1020 Color Camera Module Technical Manual. SONY. 2012.
- 9 linux-2.6.32. 17 http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/dvSDK/DVSDK_4_00/latest/index_FDS.html. 2014-10
- 10 ARM9: <http://www.arm.com/zh/products/processors/classic/arm9/index.php>.
- 11 TMS320DM36x Digital Media System-on-Chip(DMSoC) ARM Subsystem User's Guide. http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/dvSDK/DVSDK_4_00/latest/index_FDS.html. 2014-03.
- 12 Corbet J. Linux Device Drivers.北京:中国电力出版社,2006.