

摘要: 介绍了一种基于 USB2.0 协议的高分辨率数字摄像头, 并给出了利用 CMOS 图像传感器 OV9620 和 USB 控制器 CY7C68013 设计高分辨率数字摄像头的基本硬件电路以及软件设计方法。

关键词: USB2.0; I2C; 摄像头; OV9620

1 概述

目前, 能够传输高品质视频信号的两种总线接口 (USB 和 IEEE 1394 接口) 都支持即插即用, 且具有易扩展、使用方便等特点。但是 IEEE 1394 接口的价格相对较高, 因此, 当前采用更多的是 USB 接口。

USB 的数据传输有 4 种模式: 块传输 (Bulk Transfers)、中断传输 (Interrupt Transfers)、同步传输 (Isochronous Transfers)、控制传输 (Control Transfers)。当需要快速传输大批量的准确数据时, 一般采用块传输模式; 而当传输实时性较强的数据时, 则应采用中断传输模式。现在的 USB 2.0 规范在原有的低速模式 (low speed) 1.5 Mb/s、全速模式 (full speed) 12 Mb/s 的基础上又支持高达 480 Mb/s 的高速模式 (high speed), 从而使 USB 总线真正成为能满足高速应用场合的标准连接总线。在 USB 2.0 中, 每毫秒可发送 8~10 微帧, 这比 USB 1.1 的传输速率有了极大的提高, 同时对缓存的要求也大大降低了。USB 2.0 向下完全兼容, 协议规定, 在全速模式和低速模式下, 主机控制器和 hub 之间也可以进行高速传输, 但是 hub 和外设之间仍然保持原有传输模式, 这样可以使设备在分享带宽时冲突最小。

2 硬件电路

本系统选用 Omnicision Technologies Inc. 公司的 OV9620 作为视频信号的采集芯片, 用 EZ-USB FX2 作为 USB 控制芯片并为 OV9620 提供工作时钟, EEPROM 使用 24LC65, 通过 I2C 总线连接所有芯片。硬件连接框图见图 1 所示。

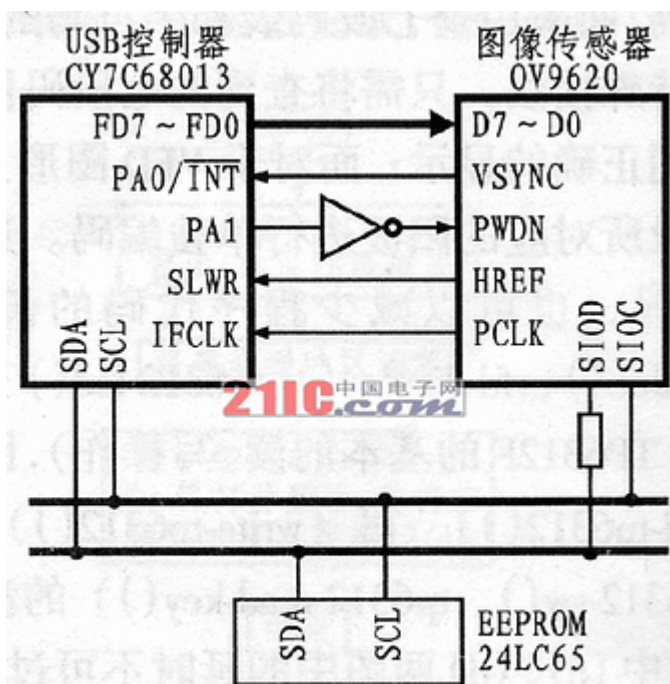


图 1 硬件电路主要部件连接框图

2.1 OV9620 的主要特性

OV9620 是美国 OmniVision 公司为数字照相机和摄像机产品开发的高性能 130 万像素彩色 CMOS 数字图像传感器芯片。与传统的 CCD 传感器相比, CMOS 最明显的优势是集成度高、功耗小、生产成本低,容易与其他芯片整合。该款芯片的图像总区域为 $6.82 \times 5.39 \text{ mm}^2$, 成像区域为 $6.66 \times 5.32 \text{ mm}^2$, 非常有利于产品的小型化。

该芯片支持 SXGA (1280×1024 分辨率) 和 VGA (640×480 分辨率) 两种模式和 SCCB (Serial Camera Control Bus) 接口, 使用 24 MHz 晶振, 内置 10 位 A/D 转换器, 能提供 10 位数字视频信号。另外该芯片还可输出像素时钟以及行、场同步信号, 并可以直接与专用集成电路系统连接, 其后端系统可以通过 SCCB 接口来控制。可利用 USB 2.0 控制器, 在 P4 1.5 G 以上系统中提供 15 fps 的 SXGA 或 30 fps 的 VGA 动态影像。该芯片采用 Bayer 滤光阵列把输入光信号分解成 RGB 三原色, 像素阵列有 $1312 \times 1036 = 1359232$ 个, 成像像素有 $1280 \times 1024 = 1310720$ 个。不成像像素可作为光学黑体像素用作黑值校准以补偿由光线引起的温度变化和暴光变化。

芯片中使用的 SCCB 接口是 OmniVision 公司制定的一种用于控制图像传感器的三线连续总线 (SCS、SIO1、SIO0), 其中 SCS 用于实现片选 (低电平有效), 当芯片被选中后, SIO1、SIO0 在功能及使用上和 I2C 总线兼容。

另外, OV9620 还提供了一个控制信号输入端 (PWDN), 可用于实现掉电模式的开关。在本设计中, 因为 OV9620 与其他芯片同时工作, 因此, 没有采用片选而是由 PWDN 来实现控制。

2.2 EZ-USB FX2 的主要特点

EZ-USB FX2 (即 CY7C68013) 是 Cypress 公司为高速传输外设设计的 USB 2.0 控制芯片。该芯片内含 1 个增强型 8051 处理器、1 个串行接口引擎 (SIE)、1 个 USB 收发器、8.5 kB 片上 RAM、4 kB FIFO 存储器以及 1 个通用可编程接口 (GPIO)。

FX2 采用量子 FIFO 处理构架, 其 USB 接口和应用环境可以直接共享 FIFO, 而微控制器则可不参与数据传输但允许以 FIFO 或 RAM 方式访问这些共享 FIFO, 这样, 就较好地解决了 USB 高速模式的带宽问题。

FX2 有一个 I2C 兼容端口, 当 FX2 作为主控制器, SCL 时钟频率为 100 / 400 kbps 时, 该端口兼容 I2C 总线。一般情况下, 该端口由两个内部控制器驱动: 一个在程序加载时自动读取 VID / PID / DID 和配置信息; 另一个是芯片中的 8051, 一旦开始运行, 8051 利用芯片中的 I2CTL 和 I2DAT 寄存器控制挂接在 I2C 总线上的外围设备。具体来说, 系统上电时, FX2 会通过内部逻辑监测 I2C 端口, 如果发现有地址为 0xC0 或者是 0xC2 的 EEPROM, 便在 EEPROM 内部存储空间 0xC0 使用 V

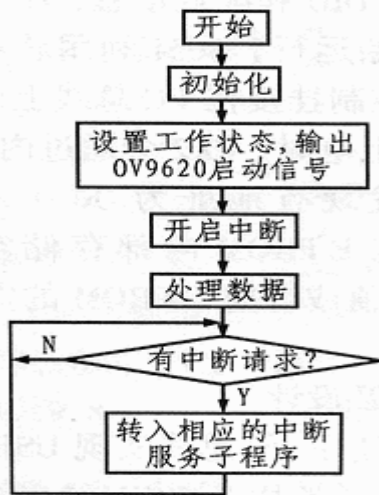


图3 主程序流程图

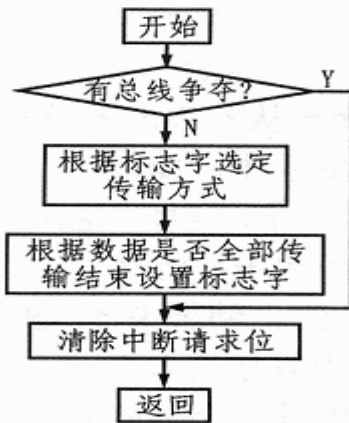


图4 I²C INT中断服务子程序

USB设备驱动程序的设计是基于WDM(驱动程序模型)并采用分层模型来进行。应用程序通过调用Windows API函数来进行的对WIN32子系统进行WIN32调用,并通过I/O管理器产生IRP(I/O请求包),然后传递给USB驱动程序,接着再由驱动程序执行相应的操作,并将结果返还给I/O管理器。用DriverEntry可设置整个系统的回调例程,当添加一个新设备和IRP需要发送到驱动程序时,通过内核可调用这些例程。下面是DriverEntry例程的简要内容:

```

DriverObject->DriverExtension->AddDevice=VcameraAddDevice;

```

```

DriverObject->DriverUnload=VcameraUnload;

```

```

DriverObject->MajorFunction[IRP_MJ_CREATE]=VcameraCreate;

```

```

DriverObject->MajorFunction[IRP_MJ_CLOSE]=VcameraClose;

```

```

DriverObject->MajorFunction[IRP_MJ_READ]=VcameraRead;

```

```

DriverObject->MajorFunction[IRP_MJ_WRITE]=VcameraWrite;

```

实际上,USB客户驱动程序中包含大量的例程,也对Windows SDK工具下开发USB客户驱动程序的方法和过程作了详细的介绍。

4 小结

目前,笔者已经开发出样品,并且在一些应用程序上进行过测试,获得了很好的效果,其分辨率高达1280×1024。