

基于 EZ-USB FX2 实现的高速数据采集系统

焦斌亮, 韩志学

(燕山大学信息科学与工程学院, 河北 秦皇岛 066004)

摘要:设计了一种基于 EZ-USB FX2 实现的高速数据采集系统, 系统采用 GPIF 控制模式, 数据通过 FIFO 缓存后打包送给 USB 总线, 打包传送时 A/D 仍持续工作, 保证数据的高速不间断采集。给出了硬件设计方案, 并对固件程序、驱动程序、应用程序编写过程中的关键性问题作了阐述。FX2 是一个全面集成的解决方案, 简化了电路的设计, 提高了系统的可靠性, 并实现了数据的高速采集与传输, 非常适合应用于各种便携式测量仪器。

关键词: CY7C68013; 通用可编程接口; 固件; 驱动程序

中图分类号: TP274.2 **文献标识码:** B **文章编号:** 1002-1841(2005)07-0021-02

High-speed Data Acquisition and Processing System Based on EZ-USB FX2

JIAO Bin-liang, HAN Zhi-xue

(College of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China)

Abstract: A high speed data acquisition system based on EZ-USB FX2 was designed. The system works in GPIF mode, after arriving in FIFO, the data was packed and sent to USB bus. In the process of packing, the A/D converter works simultaneously, which can assure high speed and uninterrupted transmission. The hardware design and some critical problems in the programming of the firmware, device driver and client software were described in details. FX2 is a good integrated solution for the system to simplifies the design, improve the reliability and implement the high speed, so the system can be used for various apparatus.

Key Words: CY7C68013; GPIF; Firmware; Device Driver

1 数据采集系统的工作原理

该采集系统总体框架为主机(能支持 USB2.0 协议的计算机)、内部包含 CPU 及 FIFO 高速缓存的 USB 接口控制芯片 FX2 (CY7C68013)^[1]和高速模数转换器 (MAX1421), 如图 1 所示。其数据传输为控制信号传输和采集数据传输。控制信号方向为由主机到外设, 由外设 CPU 控制, 数据量较小; 采集到的数据由外设到主机, 数据量较大。为了保证较高的传输速度, 使用不经过 CPU 的 GPIF 控制工作模式。系统基本操作过程为: 主机给外设一个采样控制信号, CY7C68013 根据该信号向 A/D 转换器送出时钟控制信号 CLK; A/D 转换完成的数据直接存入 CY7C68013 的内部 FIFO。当 FIFO 容量达到指定程度后, 自动将数据打包送给 USB 总线, 期间所有操作不需要 CPU 的干预, 提高了采集速度。采样过程中接口控制逻辑依次取走批量数据, 在打包传送时 A/D 仍持续转换, 内部 FIFO 也持续写入转换结果。只要内部 FIFO 写指针和读指针位置相差达到指定的值就立即取走数据, 从而保证了连续高速采集的可靠性。

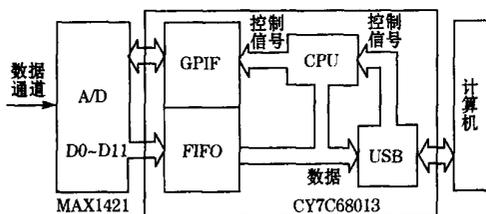


图 1 系统框图

2 硬件电路设计

系统采用 MAXIM 公司的 MAX1421, 采用单端输入和内部参考电压工作方式。图 2 为系统硬件原理

图, 其中 MAX4108 是模数转换前端放大器, 单位增益带宽可达 400 MHz, 充分满足系统要求。

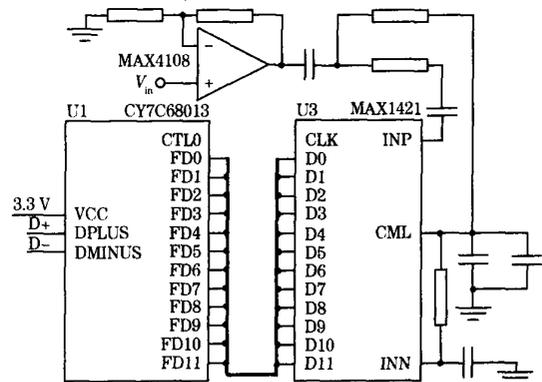


图 2 硬件连接

CY7C68013 与 MAX1421 之间的连接是通过 GPIF 和 FIFO 实现的。GPIF 控制模数转换器 MAX1421 工作, 当转换开始后, 数据从 MAX1421 的 D0~D11 脚输出到 CY7C68013 的 FD0~FD11 脚, 也就是直接将数据输入到 CY7C68013 内部的 FIFO。GPIF 的核心是一个可编程状态机, 它最多可产生 6 个“control”输出和 9 个“address”输出以及接受 6 个外部和 2 个内部的“ready”输入。可以由用户定义波形描述符去控制状态机, 产生系统运行所需的控制信号去控制外围设备的工作。Cypress 公司提供了一个名为 GPIFtool 的波形生成工具, 可以方便的定义所需波形, 不必考虑描述符的细节。该数据采集系统就是使用 GPIFtool 工具对 GPIF 编程, 利用 GPIF 的一个“control”输出 CTLO 作为 MAX1421 的时钟输入的。这样不仅简化了电路, 而且可以通过编程改变 CTLO 的频率来改变 ADC 采样频率, 通过启动和停止 GPIF 来控制 ADC 采样的启停。图 3 为 GPIF 运行时序图, IFCLK 是 GPIF 的时钟信号, 可设置为内部提供或外

部输入。CTL0 在图例中被编程为 IFCLK 的 1/4, 根据实际需要还可以被编成其他频率。如果 IFCLK 被设置为 48 MHz, 这时的采样频率为 12 MHz。

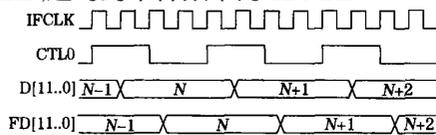


图3 GPIF 运行时序图

3 软件设计

3.1 固件程序

固件程序是指运行在设备 CPU 中的程序, 只有在该程序运行时, 外设才称为具有给定功能的外围设备。Cypress 公司为 FX2 固件开发提供了一个固件库和固件框架, 都是在 KeilC51 集成开发环境下开发的, 固件库提供了一些常量、数据结构、宏、函数来简化用户对芯片的使用。固件框架实现了初始化芯片、处理 USB 标准设备请求以及挂起状态下的电源管理等功能。该框架不添加任何代码, 编译后生成的 HEX 文件载入芯片就能和主机进行基本的 USB 通信。用户主要的工作就是选择合适的传输方式, 添加需要使用的端点, 在框架预留的地方 (如 `IDInit()`, `IDPoll` 等函数中) 添加初始化代码和完成特定功能的代码。固件程序流程图见图 4。

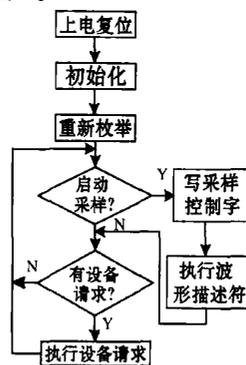


图4 固件程序流程图

3.2 驱动程序

该系统需要通用驱动程序和下载固件的驱动程序。通用驱动完成与外设和用户程序的通信及控制; 而下载固件的驱动则只负责在外设连接 USB 总线后把特定的固件程序下载到 FX2 的 RAM 中, 使 FX2 的 CPU 重启, 模拟断开与 USB 总线连接, 完成对外设的重新设置。主机根据新的设置安装通用驱动程序, 重新枚举外设为一个新的 USB 外设。通用驱动程序一般不需要重新编写, 用 cypress 公司已经编好的驱动 `ezusb.sys`; 而下载固件程序的驱动则必须定做^[3]。

3.3 用户程序

用户程序编写的关键是如何实现从 USB 设备读取指定数量的数据以及向其发送指定数量的数据。另外, 还有向 USB 设备发送 USB 标准设备请求和特定的命令, 包括厂商定义的命令和用户自定义的命令。

对用户来说, 所有的应用程序都是通过 I/O 控制调用来访问 `ezusb.sys` 通用驱动程序的。用户程序首先通过调用 Win32 函数 `CreateFile()` 得到设备驱动程序的句柄, 然后使用 Win32 函数 `DeviceIoControl()` 来

提交 I/O 控制码, 并且为 `CreateFile()` 函数返回的设备句柄设置 I/O 缓冲区。EZ-USB 通用驱动程序支持的 IOCTLs 和它们相应的输入输出结构都定义在开发包提供的头文件 `EZUSBSYS.H` 中, 用户在编程时需要包含这个文件。下面就是利用块传输方式从 USB 设备读取 2 048 字节数据的程序, 用 VC++ .NET 编写。

定义变量及初始化

```
int nBytes = 0;
UCHAR buff[2048];
BOOLEAN bResult = FALSE;
HANDLE hDevice = FALSE;
BULK_TRANSFER_CONTROL bulkControl;
bulkControl.pipeNum = 0; 传输使用的管道号, 与具体应用有关
```

打开通用驱动程序

```
hDevice = CreateFile(\\.\Ezusb,
GENERIC_READ|GENERIC_WRITE, 存取方式
FILE_SHARE_WRITE|FILE_SHARE_READ, NULL, OPEN_EXISTING, 0, NULL);
```

```
If (hDevice == INVALID_HANDLE_VALUE)
```

```
AfxMessageBox(" 打开驱动程序失败 ");
```

执行块传输读操作

```
bResult = DeviceIoControl(hDevice, 已打开的设备句柄
IOCTL_EZUSB_BULK_READ, I/O 操作控制代码
&bulkControl, 输入数据缓冲区指针
sizeof(BULK_TRANSFER_CONTROL), 输入缓冲区大小
&buff[0], 输出缓冲区指针
sizeof(buff), 输出缓冲区大小
(unsigned long *)&nBytes, 实际返回的字节数
NULL);
```

```
...
```

```
CloseHandle(hDevice);
```

执行程序, 如果 `bResult` 为 `TRUE`, 则 `buff` 中存放的是从 USB 设备读回的 2 048 字节的数据。

在系统测试中, 以 100 kHz、幅值 1.5 V 的正弦波作为输入进行数据采集与处理, 在测试程序显示的波形如图 5 所示。

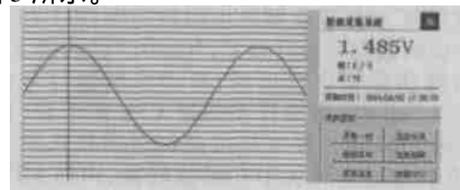


图5 测试波形

4 结束语

EZ-USB FX2 符合 USB 2.0 协议规范^[4], 理论最大速率达 480 M bit/s。在应用中去除其他时间消耗外, 数据传输速率仍可达 416 M bit/s。另外, USB 总线所具有的即插即用、热插拔等特性给采集系统提供了很大的便利。将该数据采集系统用于各种便携式测量仪器可以实现较传统示波器更方便、更精确、更有效的信号显示。

参考文献

- [1] EZ-USB FX2 technical reference manual version 2.0. Cypress Semiconductor Corp data book, 2001.
- [2] 马忠梅, 籍顺心, 张凯, 等. 单片机的 C 语言应用程序设计. 北京: 北京航空航天大学出版社, 2003: 285 - 300.
- [3] 颜荣江, 余志强, 张进, 等. EZ-USB 2100 系列单片机原理、编程及应用. 北京: 北京航空航天大学出版社, 2002: 259 - 263.
- [4] 肖踞雄, 翁铁成, 宋中庆. USB 技术与应用技术. 北京: 清华大学出版社, 2003: 280 - 288.