触摸屏控制芯片 ADS7843 中文资料

ADS7843 是一个内置 12 位模数转换、低导通电阻模拟开关的串行接口芯片。供电电压 2.7~5 V，参考电压 VREF 为 1 V~+VCC，转换电压的输入范围为 0~ VREF，最高转换速率为 125 kHz。

ADS7843 引脚图及引脚功能说明了:

ADS7843 的引脚配置如图 3 所示。表 1 为引脚功能说明，图 4 为典型应用。

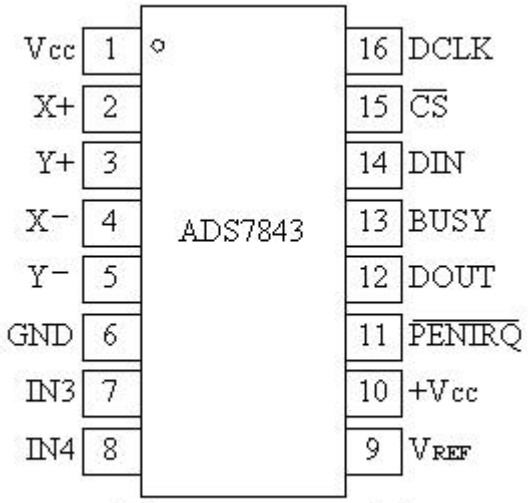

图 3　ADS7843 引脚

aDS7843 引脚说明

表 1　引脚功能说明

引脚号	引脚名	功 能 描 述
1,10	$+V_{cc}$	供电电源 2.7~5 V
2,3	X+,Y+	接触摸屏正电极，内部 A/D 通道
4,5	X−,Y−	接触摸屏负电极
6	GND	电源地
7,8	IN3,IN4	两个附属 A/D 输入通道
9	V_{REF}	A/D 参考电压输入
11	$\overline{PENIRQ}$	中断输出，须接外拉电阻（10 kΩ或 100 kΩ）
12,14,16	DOUT,DIN,DCLK	串行接口引脚，在时钟下降沿数据移出，上升沿移进
13	BUSY	忙指示，低电平有效
15	$\overline{CS}$	片选

ADS7843 典型应用电路

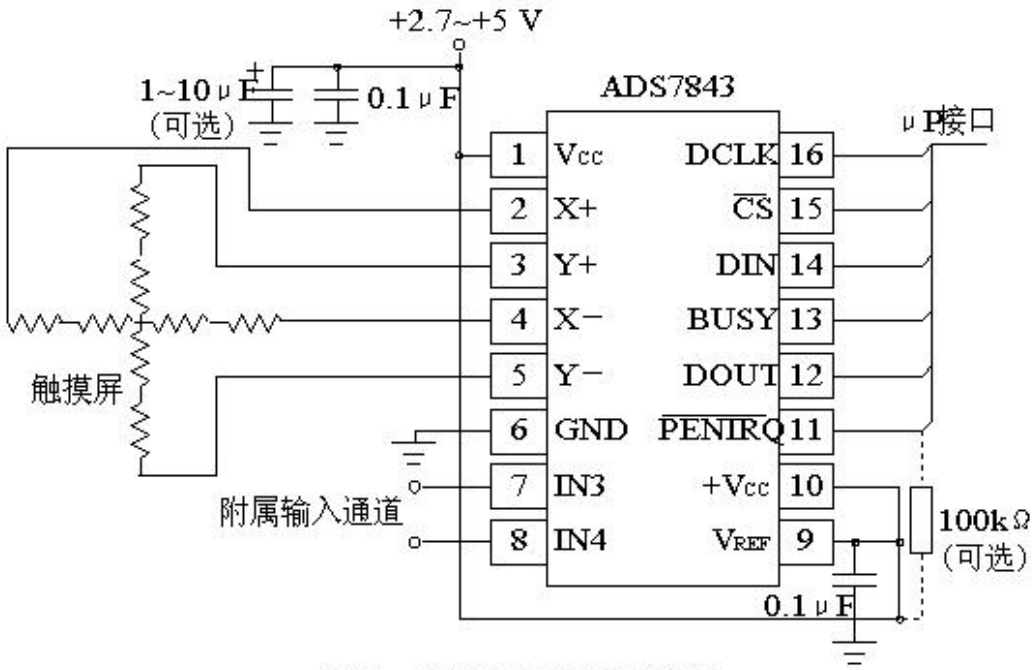

图 4　ADS7843 的典型应用

2.2 ADS7843 的内部结构及参考电压模式选择

ADS7843 之所以能实现对触摸屏的控制，是因为其内部结构很容易实现电极电压的切换，并能进行快速 A/D 转换。图 5 所示为其内部结构，A2~A0 和 SER/为控制寄存器中的控制位，用来进行开关切换和参考电压的选择。

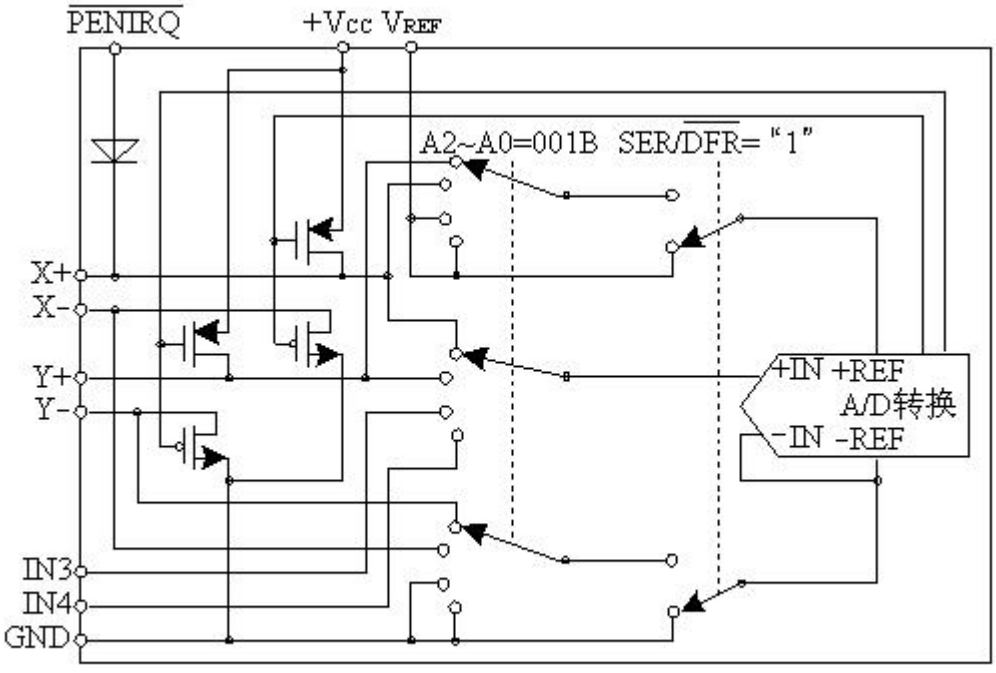

图 5　ADS7843 内部结构

ADS7843 支持两种参考电压输入模式：一种是参考电压固定为 VREF，另一种采取差动模式，参考电压来自驱动电极。这两种模式分别如图 6(a)、(b)所示。采用图 6(b)的差动模式可以消除开关导通压降带来的影响。表 2 和表 3 为两种参考电压输入模式所对应的内部开关状况。

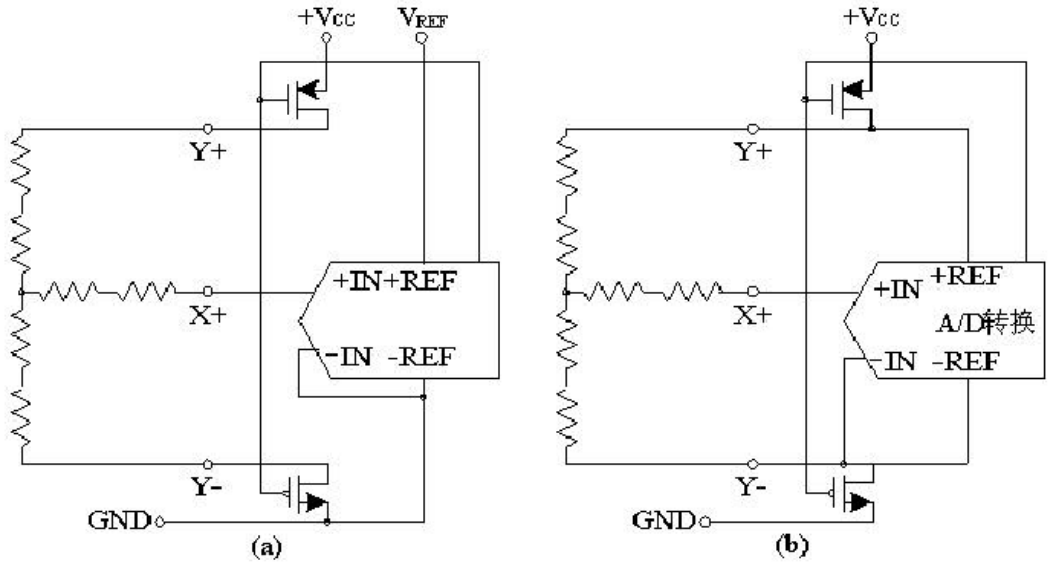

图 6　参考电压输入模式

表 3　参考电压差动输入模式（SER/$\overline{DFR}$ = “0”）

A2	A1	A0	X+	Y+	IN3	IN4	-IN	X 开关	Y 开关	+REF	-REF
0	0	1	+IN				-Y	OFF	ON	+Y	-Y
1	0	1		+IN			-X	ON	OFF	+X	-X
0	1	0			+IN		GND	OFF	OFF	$+V_{REF}$	GND
1	1	0				+IN	GND	OFF	OFF	$+V_{REF}$	GND

表 4　ADS7843 的控制字

bit7(MSB)	bit6	bit5	bit4	bit3	bit2	bit1	bit0
S	A2	A1	A0	MODE	SER/$\overline{DFR}$	PD1	PD0

2.3 ADS7843 的控制字及数据传输格式

ADS7843 的控制字如表 4 所列，其中 S 为数据传输起始标志位，该位必为"1"。A2~A0 进行通道选择(见表 2 和 3)。

MODE 用来选择 A/D 转换的精度，"1"选择 8 位，"0"选择 12 位。

SER/选择参考电压的输入模式(见表 2 和 3)。PD1、PD0 选择省电模式：

"00"省电模式允许，在两次 A/D 转换之间掉电，且中断允许；

"01"同"00"，只是不允许中断；

"10"保留；

"11"禁止省电模式。

为了完成一次电极电压切换和 A/D 转换，需要先通过串口往 ADS7843 发送控制字，转换完成后再通过串口读出电压转换值。标准的一次转换需要 24 个时钟周期，如图 7 所示。由于串口支持双向同时进行传送，并且在一次读数与下一次发控制字之间可以重叠，所以转换速率可以提高到每次 16 个时钟周期，如图 8 所示。如果条件允许，CPU 可以产生 15 个 CLK 的话(比如 FPGAs 和 ASICs),转换速率还可以提高到每次 15 个时钟周期，如图 9 所示。

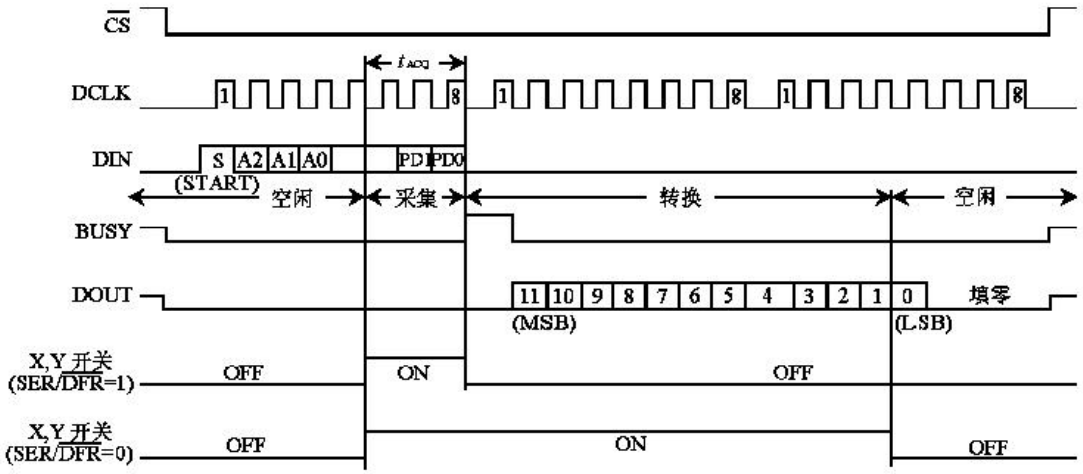

图 7　A/D 转换时序（每次转换需 24 个时钟周期）

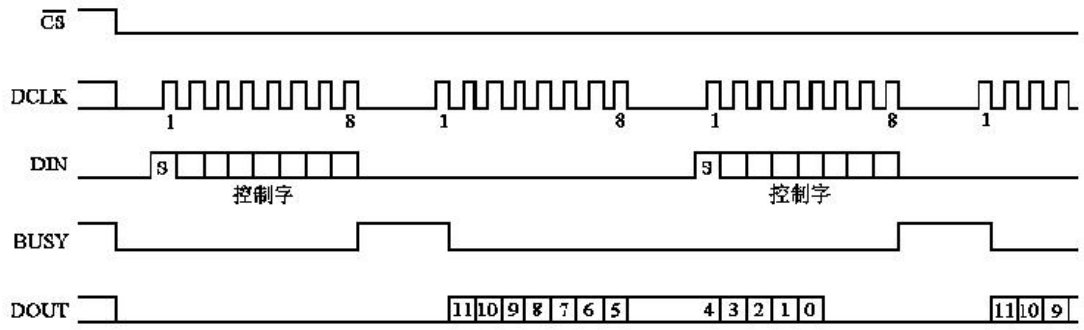

图 3　A/D 转换时序（每次转换需 16 个时钟周期）

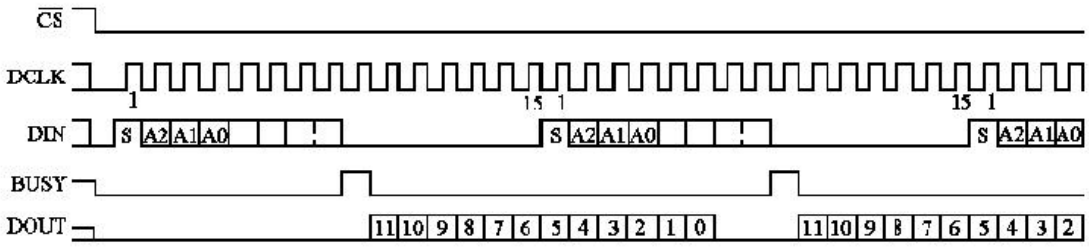

图 9　A/D 转换时序（每次转换需 15 个时钟周期）

2.4 A/D 转换时序的程序设计

ADS7843 的典型应用如图 4 所示。假设 μP 接口与 51 单片机的 P1.3~P1.7 相连，现以一次转换需 24 个时钟周期为例，介绍 A/D 转换时序的程序设计。

; A/D 接口控制线

DCLK BIT P1.3

CS BIT P1.4

DIN BIT P1.5

BUSY BIT P1.6

DOUT BIT P1.7

; A/D 通道选择命令字和工作寄存器

CHX EQU 094H ;通道 X+的选择控制字

CHY EQU 0D4H;通道 Y+的选择控制字

CH3 EQU 0A4H

CH4 EQU 0E4H

AD_CH EQU 35H ;通道选择寄存器

AD_RESULTH EQU 36H ;存放 12 bit A/D 值

AD_RESULTL EQU 37H

; 存放通道 CHX+的 A/D 值

```
CHXAdResultH EQU 38H

CHXAdResultL EQU 39H

; 存放通道 CHY+的 A/D 值

CHYAdResultH EQU 3AH

CHYAdResultL EQU 3BH

; 采集通道 CHX+的程序段(CHXAD)

CHXAD: MOV AD_CH,#CHX

LCALL AD_RUN

MOV CHXAdResultH,AD_RESULTH

MOV CHXAdResultL,AD_RESULTL

RET

; 采集通道 CHY+的程序段(CHYAD)

CHYAD: MOV AD_CH,#CHY

LCALL AD_RUN

MOV CHYAdResultH,AD_RESULTH

MOV CHYAdResultL,AD_RESULTL

RET

; A/D 转换子程序(AD_RUN)

; 输入: AD_CH-模式和通道选择命令字

; 输出: AD_RESULTH,L ;12 bit 的 A/D 转换值

; 使用: R2 ;辅助工作寄存器

AD_RUN:
```

```
CLR CS ;  芯片允许

CLR DCLK

MOV R2,#8 ;先写 8 bit 命令字

MOV A,AD_CH

AD_LOOP:

MOV C, ACC.7

MOV DIN,C ;时钟上升沿锁存 DIN

SETB DCLK ;开始发送命令字

CLR DCLK ;时钟脉冲，一共 24 个

RL A

DJNZ R2,AD_LOOP

NOP

NOP

NOP

NOP

ADW0: JNB BUSY,AD_WAIT ;等待转换完成

SJMP ADW1

AD_WAIT:

LCALL WATCHDOG

NOP

SJMP ADW0

CLR DIN
```

```
ADW1: MOV R2,#12 ;开始读取 12bit 结果
SETB DCLK
CLR DCLK
AD_READ:
SETB DCLK
CLR DCLK ;用时钟的下降沿读取
MOV A,AD_RESULTL
MOV C,DOUT
RLC A
MOV AD_RESULTL,A
MOV A,AD_RESULTH
RLC A
MOV AD_RESULTH,A
DJNZ R2,AD_READ
MOV R2,#4 ;最后是没用的 4 个时钟
IGNORE:
SETB DCLK
CLR DCLK
DJNZ R2,IGNORE
SETB CS ;禁止芯片
ANL AD_RESULTH,#0FH ;屏蔽高 4 bit
RET
```

2.5 A/D 转换结果的数据格式

ADS7843 转换结果为二进制格式。需要说明的是，在进行公式计算时，参考电压在两种输入模式中是不一样的。而且，如果选取 8 位的转换精度，1LSB=VREF/256，一次转换完成时间可以提前 4 个时钟周期，此时串口时钟速率也可以提高一倍。

本文来自: DZ3W.COM 原文网址：http://www.dz3w.com/info/commonIC/0086705.html

基于 S3C2410 的 ADS7843 触摸屏驱动程序设计

触摸屏介绍

随着多媒体信息查询的与日俱增，人们越来越多地谈到触摸屏，因为触摸屏不仅适用于中国多媒体信息查询的国情，而且触摸屏具有坚固耐用、反应速度快、节省空间、易于交流等许多优点。利用这种技术，我们用户只要用手指轻轻地碰计算机显示屏上的图符或文字就能实现对主机操作，从而使人机交互更为直截了当，这种技术大大方便了那些不懂电脑操作的用户。

一、触摸屏的工作原理

为了操作上的方便，人们用触摸屏来代替鼠标或键盘。工作时，我们必须首先用手指或其它物体触摸安装在显示器前端的触摸屏，然后系统根据手指触摸的图标或菜单位置来定位选择信息输入。触摸屏由触摸检测部件和触摸屏控制器组成；触摸检测部件安装在显示器屏幕前面，用于检测用户触摸位置，接受后送触摸屏控制器；而触摸屏控制器的主要作用是从触摸点检测装置上接收触摸信息，并将它转换成触点坐标，再送给 CPU，它同时能接收 CPU 发来的命令并加以执行。

二、触摸屏的主要类型

从技术原理来区别触摸屏，可分为五个基本种类：矢量压力传感技术触摸屏、电阻技术触摸屏、电容技术触摸屏、红外线技术触摸屏、表面声波技术触摸屏。

本文主要介绍在三星 S3C2410X 微处理器的硬件平台上进行基于嵌入式 Linux 的触摸屏驱动程序设计。

1．硬件设计

SPI 接口是 Motorola 推出的一种同步串行接口，采用全双工、四线通信系统，S3C2410X 是三星推出的自带触摸屏">触摸屏接口的 ARM920T 内核芯片，ADS7843 为 Burr-Brown 生产的一款性能优异的触摸屏">触摸屏控制器。本文采用 SPI 接口的触摸屏">触摸屏控制器 ADS7843 外接四线电阻式触摸屏">触摸屏，这种方式最显著的特点是响应速度更快、灵敏度更高，微处理器与触摸屏">触摸屏控制器间的通讯时间大大减少，提高了微处理器的效率。ADS7843 与 S3C2410 的硬件连接如图 1 所示，鉴于 ADS7843 差分工作模式的优点，在硬件电路中将其配置为差分模式。

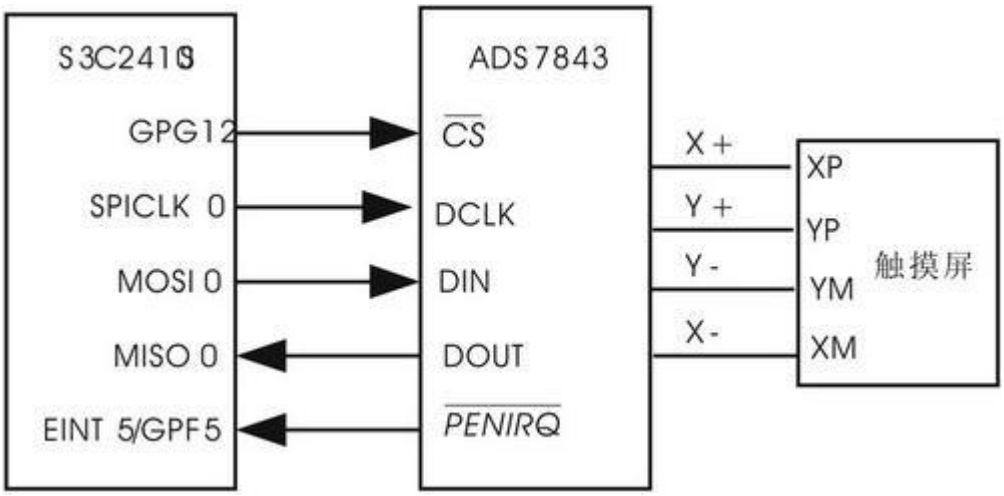

图 1 触摸屏输入系统示意图

2．嵌入式 Linux 系统下的驱动程序

设备驱动程序是 Linux 内核的重要组成部分，控制了操作系统和硬件设备之间的交互。Linux 的设备管理是和文件系统紧密结合的，各种设备都以文件的形式存放在/dev 目录下，成为设备文件。应用程序可以打开、关闭、读写这些设备文件，对设备的操作就像操作普通的数据文件一样简便。为开发便利、提高效率，本设计采用可安装模块方式开发调试触摸屏驱动程序。

设备驱动在加载时首先需要调用入口函数 init_module()，该函数完成设备驱动的初始化工作。其中最重要的工作就是向内核注册该设备，对于字符设备调用 register_chrdev()完成注册，对于块设备需要调用 register_blkdev()完成注册。注册成功后，该设备获得了系统分配的主设备号、自定义的次设备号，并建立起与文件系统的关联。字符设备驱动程序向 Linux 内核注册登记时，在字符设备向量表 chrdevs 中增加一个 device_struct 数据结构条目，这个设备的主设备标识符用作这个向量表的索引。向量表中的每一个条目，即一个 device_struct 数据结构包括两个元素：一个登记的设备驱动程序的名称的指针和一个指向一组文件操作的指针。这块文件操作本身位于这个设备的字符设备驱动程序中，每一个都处理特定的文件操作，比如打开、读写和关闭。所谓登记，就是将由模块提供的 file_operations 结构指针填入 device_struct 数据结构数组的某个表项。登记以后，位于上层的模块(内核)可以“看见”这个模块了。但是，应用程序却还不能“看见”它，因而还不能通过系统调用它。要使应用程序能“看见”这个模块或者它所驱动的设备，就要在文件系统中为其创建一个代表它的节点。通过系统调用 mknod()创建代表此项设备的文件节点——设备入口点，就可使一项设备在系统中可见，成为应用程序可以访问的设备。另外，设备驱动在卸载时需要回收相应的资源，令设备的相应寄存器值复位并从系统中注销该设备。

Linux 操作系统通过系统调用和硬件中断完成从用户空间到内核空间的控制转移。设备驱动模块的功能就是扩展内核的功能，主要完成两部分任务：一个是系统调用，另一个是处理中断。图 2 是一个设备驱动模块动态挂接、卸载和系统调用的全过程。系统调用部分则是对设备的操作过程，比如 open，read，write，ioctl 等操作，设备驱动程序所提供的这组入口点由几个结构向系统进行说明，分别是 file_operations 数据结构、inode 数据结构和 file 数据结构。内核内部通过 file 结构识别设备，通过 file_operations 数据结构提供文件系统的入口点函数，也就是访问设备驱动的函数，结构中的每一个成员都对应着一个系统调用。在嵌入式系统的开发中，我们一般仅仅实现其中几个接口函数：read、write、open、ioctl 及 release 就可以完成应用系统需要的功能。写驱动程序的任务之一就是完成 file_operations 中的函数指针。

3．触摸屏驱动程序设计

触摸屏驱动程序中重要数据结构

```
typedef struct {
  unsigned short pressure;
```

```
  unsigned short x;
  unsigned short y;
  unsigned short pad;
} TS_RET;
typedef struct {
 unsigned int PenStatus;
 TS_RET buf[MAX_TS_BUF];
 unsigned int head, tail;
 wait_queue_head_t wq;
 spinlock_t lock;
} TS_DEV;
static struct file_operations s3c2410_fops = {
  owner: THIS_MODULE,
  open: s3c2410_ts_open,
  read: s3c2410_ts_read,    release: s3c2410_ts_release,
poll: s3c2410_ts_poll,    };
```

在程序中有三个重要的数据结构：用于表示笔触点数据信息的结构 TS_RET，表示 ADS7843 中有关触摸屏控制器信息的结构 TS_DEV，以及驱动程序与应用程序的接口 file_operations 结构的 s3c2410_fops。

TS_RET 结构体中的信息就是驱动程序提供给上层应用程序使用的信息，用来存储触摸屏的返回值。上层应用程序通过读接口，从底层驱动中读取信息，并根据得到的值进行其他方面的操作。

TS_DEV 结构用于记录触摸屏运行的各种状态，PenStatus 包括 PEN_UP、PEN_DOWN 和 PEN_FLEETING。buf[MAX_TS_BUF]是用来存放数据信息的事件队列，head、tail 分别指向事件队列的头和尾。程序中的笔事件队列是一个环形结构，当有事件加入时，队列头加一，当有事件被取走时，队列尾加一，当头尾位置指针一致时读取笔事件的信息，进程会被安排进入睡眠。wq 等待队列，包含一个锁变量和一个正在睡眠进程链表。当有好几个进程都在等待某件事时，Linux 会把这些进程记录到这个等待队列。它的作用是当没有笔触事件发生时，阻塞上层的读操作，直到有笔触事件发生。lock 使用自旋锁，自旋锁是基于共享变量来工作的，函数可以通过给某个变量设置一个特殊值来获得锁。而其他需要锁的函数则会循环查询锁是否可用。MAX_TS_BUF 的值为 16，即在没有被读取之前，系统缓冲区中最多可以存放 16 个笔触数据信息。

s3c2410_fops 就是内核对驱动的调用接口，完成了将驱动函数映射为标准接口。上面的这种特殊表示方法不是标准 C 的语法，而是 GNU 编译器的一种特殊扩展，它使用名字进行结构字段的初始化，它的好处体现在结构清晰，易于理解，并且避免了结构发生变化带来的许多问题。

init_module 函数

这是模块的入口函数。在函数内部通过 s3c2410_ts_init( )实现模块的初始化工作。在本设计中设备与系统之间以中断方式进行数据交换。整个触摸屏">触摸屏的驱动程序处理比较复杂，而且耗时较长，因而触摸屏">触摸屏驱动程序不可能在中断服务程序中完成。在 Linux 操作系统中一般把中断处理切为两个部分或两半。中断处理程序是上半部——接收到一个中断，它就立即开始执行，但只做有严格时限的工作，例如对接收的中断进行应答或复位硬件。这些工作都是在所有中断被禁止的情况下完成的，能够被允许稍后完成的工作会推迟到下半部去。在 Linux 中下半部的实现有多种机制。按触摸屏">触摸屏时，从 ADS7843 输出的数值有一个抖动过程，即从 ADS7846 输出的数值有一个不稳定时期，这个过程大约为 10ms。所以中断处理程序的下半部处理函数采用内核定时器机制，使下半部在中断发生 50ms 后再作处理。这样有效地避开了 ADS7843 输出值的不稳定时期，使中断服务程序和中断处理任务串行化，达到了处理时间较长的触摸屏">触摸屏事件的目的。驱动程序通过 request_irq 函数注册并激活一个中断处理程序，以便处理中断。

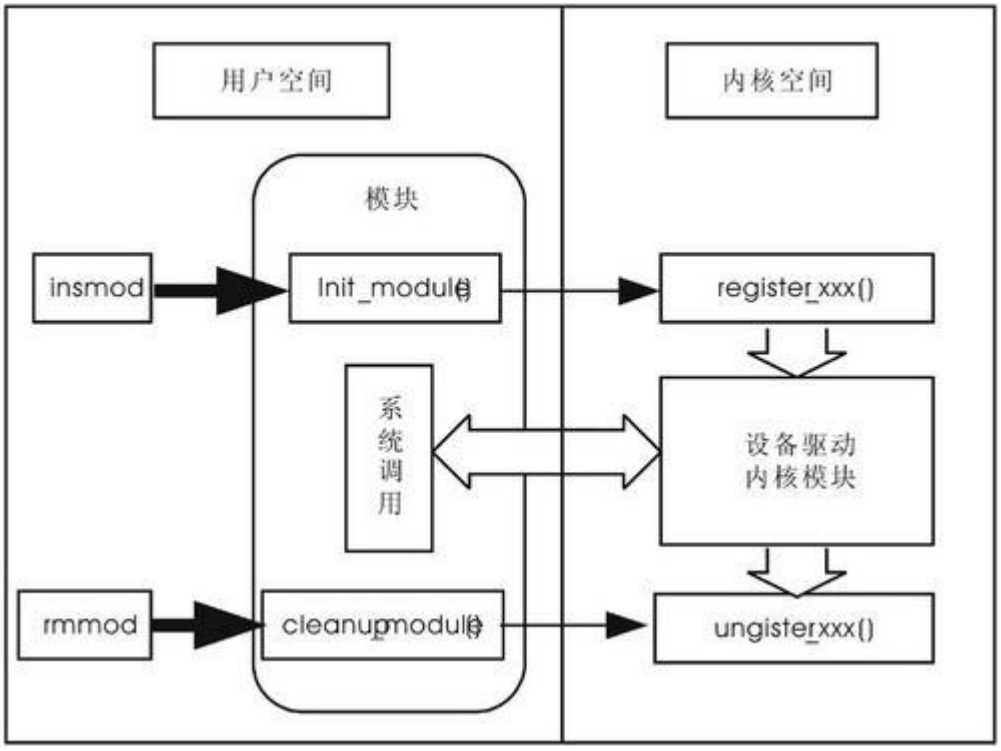

图 2 设备驱动在内核中的挂接、卸载和系统调用过程

int reguest_irq(unsigned int irq, void(*handler)(int, void *, struct pt_regs *), unsigned long irq_flags, const char *dev_name, void *dev_id)

参数 irq 表示所要申请的中断号；handler 为向系统登记的中断处理子程序，中断产生时由系统来调用；dev_name 为设备名；dev_id 为申请时告诉系统的设备标识符；irq_flags 是申请时的选项，它决定中断处理程序的一些特性，其中最重要的是中断处理程序是快速处理程序还是慢速处理程序。

本设计中触摸屏">触摸屏控制器 ADS7843 的中断输出通过外部中断 5 接在中断控制器上，当触摸屏">触摸屏上有触摸事件发生时，会引发中断号为 IRQ_EINT5 的中断服务程序 s3c2410_isr_tc()。图 3 所示为该中断处理程序的流程图。

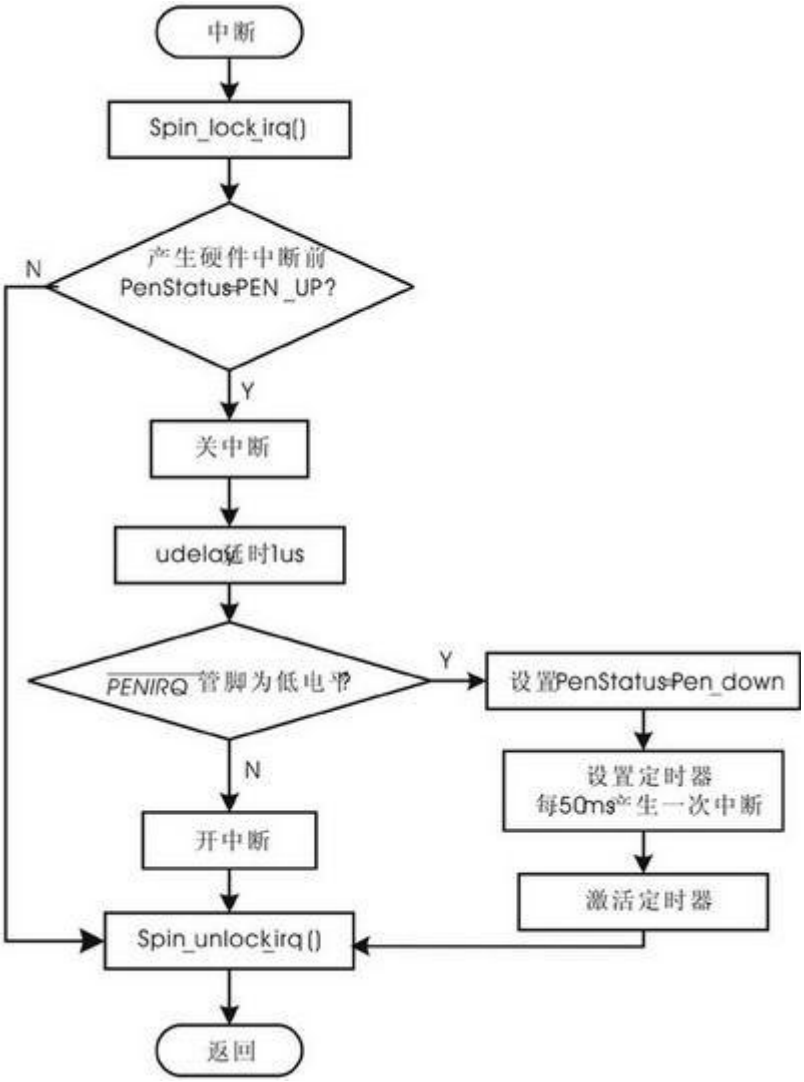

图 3 触摸屏硬件中断处理程序流程图

在 s3c2410_isr_tc()中设定了定时器的定时时间为 50ms，并立即激活。因此有触摸屏">触摸屏硬件中断的情况下 50ms 后就会引发定时中断，中断服务程序为 ts_timer_handler()，这个程序实现了触摸屏">触摸屏中断的下半部，即在过了抖动时间之后如果触摸屏">触摸屏确实有有效事件发生则采集触摸屏">触摸屏坐标，并将定时器的时间重新设为 100ms 并重新激活，这样做的目的是如果触摸笔是拖动的情况，以后每 100ms 采集一次坐标值，并存入缓冲区，如果不是拖动在采集一次坐标值之后，在第二次进入 ts_timer_handler()时，查询管脚的状态值，则变为高电平，就将触摸屏">触摸屏状态 tsdev.PenStatus 设为 PEN_UP，并释放定时器，为下次触摸屏">触摸屏事件做好准备，定时中断服务程序流程图如图 4 所示。

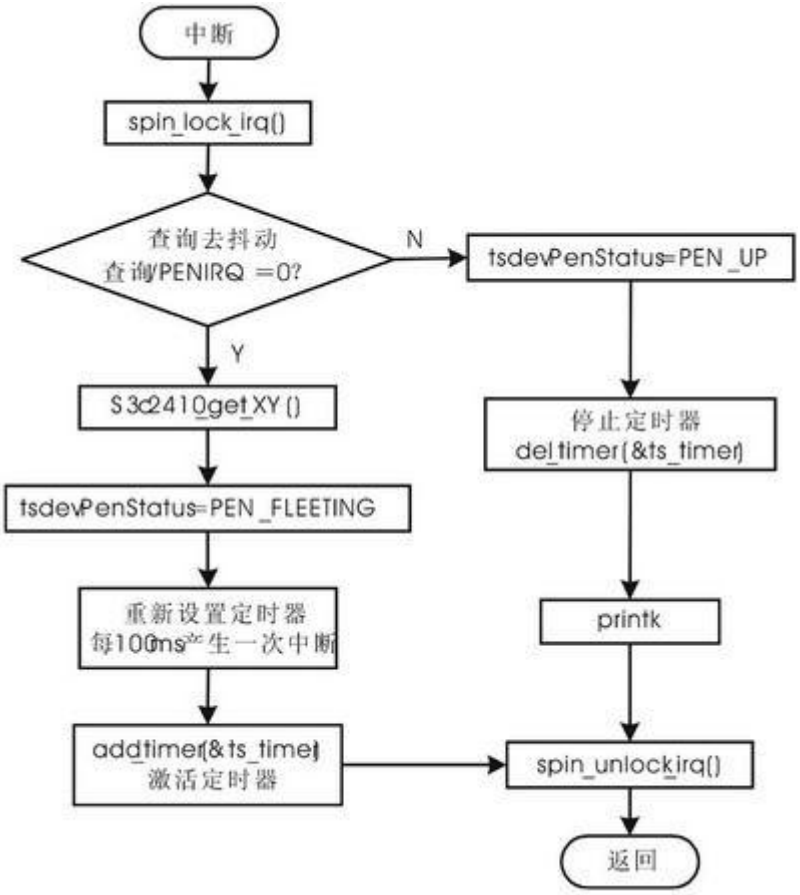

图 4 定时中断服务程序流程图

在 s3c2410_ts_init()中的另一个重要任务是执行接口函数 s3c2410_ts_open()，在这个函数中初始化缓冲区的头尾指针、触摸屏状态变量及触摸屏事件等待队列。

module_exit()

该函数调用 s3c2410_ts_exit()，主要任务是撤销驱动程序向内核的登记以及释放申请的中断资源。

接口函数 s3c2410_ts_read( )

这个函数实现的任务是将事件队列从设备缓存中读到用户空间的数据缓存中。实现的过程主要是通过一个循环，只有在事件队列的头、尾指针不重合时，才能成功的从 tsdev.tail 指向的队列尾部读取到一组触摸信息数据，并退出循环。否则调用读取函数的进程就要进入睡眠。

坐标读取函数 s3c2410_get_XY()

在定时器中断处理程序中，当查询到与相连的 EINT5/GPF5 为低电平时，即表示有有效事件，应该调用 s3c2410_get_XY()函数采集笔触信息。

ADS7843 有多种转换时序，时序规定了芯片与设备及 CPU 间是如何配合工作的。设计中采用 16 个时钟周期启动一次转换的坐标转换方式。ADS7843 的操作时序如图 5 所示。坐标的读取是通过多次采集取平均值的方法，。循环过程中的每一步都在 8 个时钟周期内完成，数据的处理严格按照时序进行，Y 坐标的采集与 X 坐标类似。

结束语：

触摸屏作为一种最新的电脑输入设备，它是目前最简单、方便、自然的一种人机交互方式。它赋予了多媒体以崭新的面貌，是极富吸引力的全新多媒体交互设备。触摸屏在我国的应用范围非常广阔，主要是公共信息的查询；如电信局、税务局、银行、电力等部门的业务查询；城市街头的信息查询；此外应用于领导办公、工业控制、军事指挥、电子游戏、点歌点菜、多媒体教学、房地产预售等。将来，触摸屏还要走入家庭。