触摸屏在S3C2410上的应用实例※

■ 国防科技大学　宋成　孙广富

摘　要　给出S3C2410上触摸屏的实现原理、硬件结构和软件程序；对软件进行优化，改进软件滤波的实现方法。其算法使用C语言实现，可移植到任何操作系统的触摸屏驱动程序中。

关键词　触摸屏　S3C2410　滤波

引　言

随着个人数字助理(PDA)、瘦容户机等的普及，触摸屏作为终端与用户交互的媒介，在我们的生活中使用得越来普遍。触摸屏分为电阻式、电容式、声表面波式和红外线扫描式等类别，使用得最多的是4线电阻式触摸屏。

本文以三星公司ARM9内核芯片S3C2410触摸屏接口为基础，通过外接4线电阻式触摸屏构成硬件基础。在此基础上，开发了触摸屏画图板程序。

1　触摸屏原理

S3C2410接4线电阻式触摸屏的电路原理如图1所示。整个触摸屏由横向电阻丝和纵向电阻丝组成，由nYPON、YMON、nXPON、XMON四个控制信号控制4个MOS管(S1、S2、S3、S4)的通断。S3C2410有8个模拟输入通道。其中，通道7作为触摸屏接口的X坐标输入(图1的AIN[7])，通道5作为触摸屏接口的Y坐标输入(图1的AIN[5])。电路如图2所示。在接入S3C2410触摸屏接口前，它们都通过一个阻容式低通滤波器滤除坐标信号噪声。这里的滤波十分重要，如果传递给S3C2410模拟输入接口的信号中干扰过大，不利于后续的软件处理。在采样过程中，软件只用给特殊寄存器置位，S3C2410的触摸屏控制器就会自动控制触摸屏接口打开或关闭各MOS管，按顺序完成X坐标点采集和Y坐标点采集。

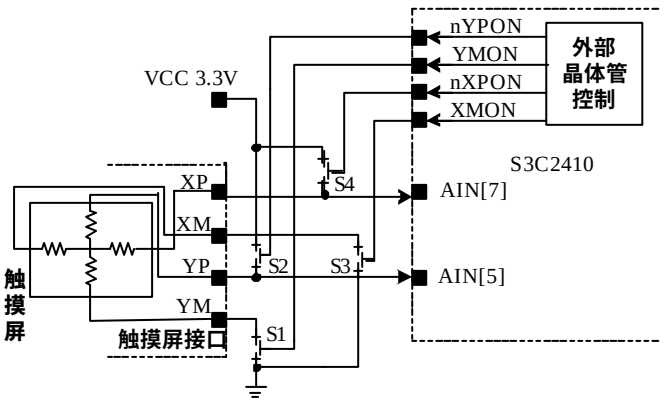

图1　S3C2410接4线电阻式触摸屏电路原理

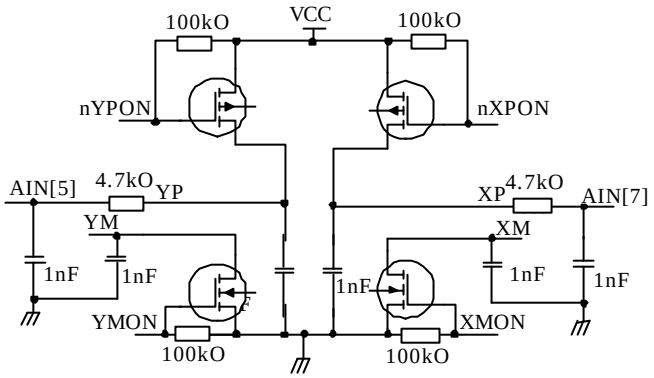

图2　S3C2410触摸屏接口电路

2　S3C2410触摸屏控制器

S3C2410触摸屏控制器有2种处理模式：

① X/Y位置分别转换模式。触摸屏控制器包括两个控制阶段，X坐标转换阶段和Y坐标转换阶段。

② X/Y位置自动转换模式。触摸屏控制器将自动转换X和Y坐标。

本文使用X/Y位置自动转换模式。

3　S3C2410触摸屏编程

由于触摸屏程序中参数的选取优化需要多次试验，而加入操作系统试验参数，每次编译下载耗费时间过多，不易于试验的进行，因而我们直接编写裸机触摸屏程序。三星公司开放了S3C2410测试程序2410test(可在三星网站下载)，提供了触摸屏接口自动转换模式的程序范例ts_auto.c，见本刊网站www.dpj.com.cn。本文在此范例的基础上编写了触摸屏画图板程序——在显示屏

上画出触摸笔的流走痕迹。

针对坐标点采样时产生的噪声，本文采用噪声滤波算法，编写了相应的噪声滤波程序，滤除干扰采样点。整个触摸屏画图板程序的处理流程如图3所示。

3.1 程序初始化

初始化触摸屏控制器为自动转换模式。其中寄存器ADCDLY的值需要根据具体的试验选取，可运行本文提供的程序看画线的效果来选取具体的参数。触摸屏中断处理程序Adc_or_TsAuto是判断触摸屏是否被按下了。触摸屏被按下，给全局变量Flag_Touch赋值为Touch_Down，否则赋值为Touch_Up。

初始化脉宽调制计时器(PWM TIMER)，选择计时器4为时钟，定义10ms中断1次，提供触摸屏采样时间基准，即10ms触摸屏采样1次。计时器中断处理程序Timer4Intr中判断Flag_Touch被赋值为Touch_Down，则给全局变量gTouchStartSample置位，以控制触摸屏采样。

之后清除触摸屏中断和计时器中断屏蔽位，接受中断响应，同时计时器开始计时。

3.2 触摸屏采样程序

如果gTouchStartSample为TRUE，触摸屏接口开始对坐标X和Y的模拟量进行采样，根据试验选取适合的采集次数。本文中使用9次采集，分别记入到ptx[TouchSample]和pty[TouchSample]数组中，TouchSample为采集次数。

为了减少运算量，将ptx[]和pty[]分别分三组取平均值，存储在px[3]和py[3]中。这里以处理 X 坐标为例：

```
px[0]=(ptx[0]+ptx[1]+ptx[2])/3;
px[1]=(ptx[3]+ptx[4]+ptx[5])/3;
px[2]=(ptx[6]+ptx[7]+ptx[8])/3;
```

计算以上三组数据的差值：

```
dlXDiff0 = px[ 0 ] - px[ 1 ];
dlXDiff1 = px[ 1 ] - px[ 2 ];
dlXDiff2 = px[ 2 ] - px[ 0 ];
```

然后对上述差值取绝对值，所得结果简称绝对差值：

```
dlXDiff0 = dlXDiff0 > 0 ? dlXDiff0 : -dlXDiff0;
dlXDiff1 = dlXDiff1 > 0 ? dlXDiff1 : -dlXDiff1;
dlXDiff2 = dlXDiff2 > 0 ? dlXDiff2 : -dlXDiff2;
```

判断上述计算的绝对差值是否都超过差值门限，如果这3个绝对差值都超过门限值，判定这次采样点为野点，抛弃采样点，程序返回等待下次采样。其中的差值门限需要根据试验测试得到，本文取值为2。

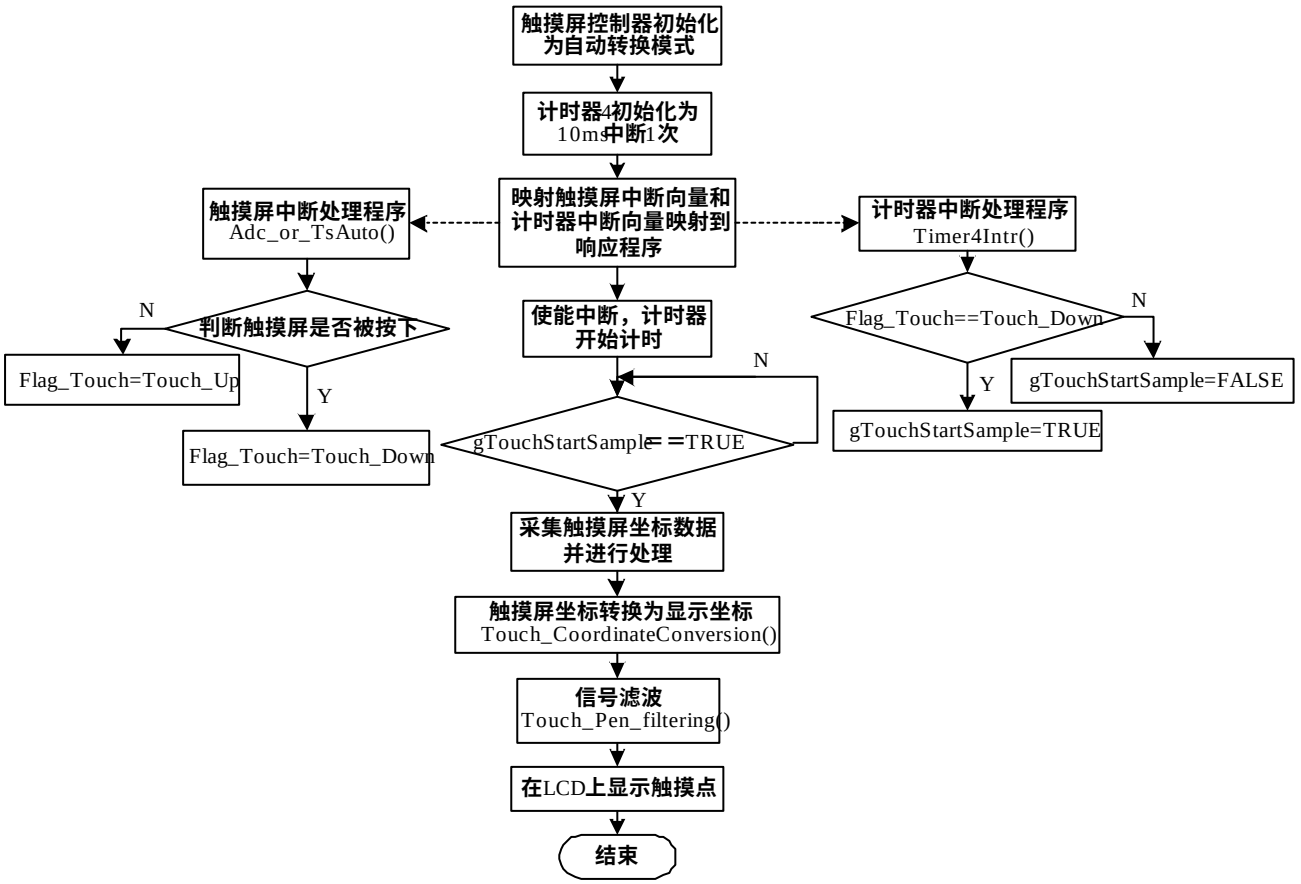

图3 程序框架

找出其中绝对差值最小的2组数据，再将它们作平均，同时赋值给tmx：

```
if ( dlXDiff0 < dlXDiff1 ){
    if ( dlXDiff2 < dlXDiff0 ){
        tmx = ( ( px[ 0 ] + px[ 2 ] ) >> 1 ) ;
    }
    else{
        tmx = ( ( px[ 0 ] + px[ 1 ] ) >> 1 );
    }
}
else if ( dlXDiff2 < dlXDiff1 ){
    tmx = ( ( px[ 0 ] + px[ 2 ] ) >> 1 ) ;
}
else{
    tmx = ( ( px[ 1 ] + px[ 2 ] ) >> 1 ) ;
}
```

函数Touch_Coordinate Conversion完成触摸屏采样值转换成显示坐标，根据不同的硬件有不同的转换方法。本触摸屏采样坐标及显示坐标如图4、图5所示。其中TOUCH_MAX_X和TOUCH_MIN_X是触摸屏*X*坐标采样值的最大和最小值；*Y*坐标同理。可以运行本文程序，同时使用触摸笔在触摸屏的4个角取得最大最小采样值。这里使用的是320×240的TFT屏，所以TOUCH_X值为320。下面是X坐标的转换程序：

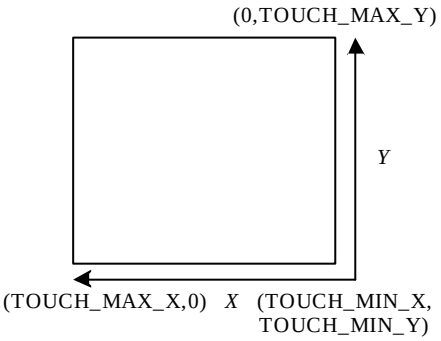

图4 触摸屏坐标

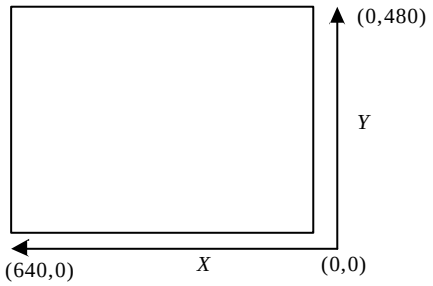

图5 显示坐标

```
Touch_CoordinateConversio(int *px){
    TmpX = (tmx >= TOUCH_MAX_X) ? (TOUCH_MAX_X) :
*px;
    TmpX - = TOUCH_MIN_X;
    TmpX = (TmpX) ? TmpX : 0;
    *px= (TmpX*  TOUCH_X)/(TOUCH_MAX_X-
TOUCH_MIN_X);
}
```

3.3 坐标滤波程序

坐标滤波程序Touch_Pen_filtering，考虑人机界面中对触摸屏的操作有3种：

◆ 触摸笔在触摸屏上的位置不变；

◆ 触摸笔在触摸屏上连续滑过；

◆ 触摸笔在触摸屏上有大幅度的跳跃。

假设三次连续采样时刻为T1、T2、T3(T3>T2>T1)，采样间隔为10ms。由于采样间隔远小于人的反应时间，所以在前两种操作模式下，如果采样点有效，将T1和T3时刻的采样值作平均。其平均值和T2时刻的采样值比较一般不会大于某个门限，否则判定此次采样点为野点。而对于第三种模式下，采样点数据会有很大的跳变。跳变过程中的数据是不稳定的，虽然记入了数据，但被判定成无效的采样点，所以需要在程序中定义一个静态数组x[2]记录相邻的两次采样数据。只有当前后数据持续稳定一段时间，才认为这时的采样点有效。程序中使用的间隔门限FILTER_LIMIT是需要经过试验来选取的。这里只给出*X*坐标的滤波过程：

```
//*px为T3时刻的采样值，count是记录连续有效采样点次数的
//静态变量，标志当前数据持续稳定时间，一旦发现大于
//FILTER_LIMIT，count的值又要从0开始计数。
int Touch_Pen_filtering(int *px){
    BOOL retVal;
    static int count = 0;
    static int x[2];
    count++;
//如果连续有效采样点次数大于2次，开始进行滤波算法
    if (count > 2) {
        count = 2;
        //将T3时刻采样值和T1时刻采样值作平均
        TmpX = (x[0] + *px) / 2;
//计算平均值和T2时刻采样值的差值
        dx = (x[1] > TmpX) ? (x[1] - TmpX) : (TmpX - x[1]);
//如果差值大于门限值，说明T3的采样值无效，判为野点返回
//值为FALSE。为了避免过大的跳跃，认为触摸笔坐标没变，使
//用T2时刻采样值来代替本次采样点，同时静态变量x[]中的数
//据不变，count重新开始记录连续有效采样点次数
    if ((dx > FILTER_LIMIT)) {
        *px = x[1];
        retVal = FALSE;
        count = 0;
    }
//否则采样点有效返回值为TRUE，将T3的采样点记入到x[1]中，
//T2的采样点移到x[0]中
    else {
        x[0] = x[1];
```

```
        x[1] = *px;
        retVal = TRUE;
    }
}
    else {
//连续有效采样次数小于2，将T3的采样值记入到x[1]，T2的采
//样值移动到x[0]，并不进行滤波处理
        x[0] = x[1];
        x[1] = *px;
        retVal = FALSE;
        }
    return retVal;
}
```

3.4 后续处理

经过上述的筛选和滤波，如果被判定采样值有效，则将其滤波值送给操作系统进行后续处理，否则程序返回，等待下一次采样。在2410 test程序中，可以结合LCD的画点函数，将有效的采样点在LCD上画出，以此检验参数设置是否合理。

结 语

本文以三星公司ARM9内核芯片S3C2410和4线电阻式触摸屏为硬件基础。基于此硬件结构，开发了触摸屏画图板程序。通过软件滤波，提高了系统性能，得到了很好的处理效果，有很强的实用性。用此算法实现的WindowsCE触摸屏驱动，在汉王手写输入软件的测试下得到了很高的识别率。软件使用C语言实现，可以方便地移植到任何操作系统上。

参考文献

1 杜春雷. ARM 体系结构与编程[M]. 北京: 清华大学出版社, 2003

2 Samsung Corp. User's Manual S3C2410A[EB/OL]. http://www.samsung.com

（收稿日期：2004-08-09）