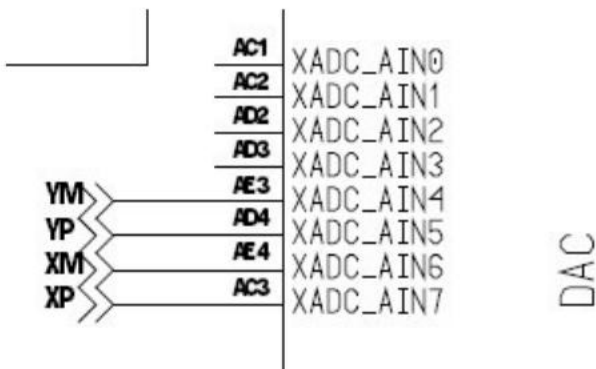


电阻屏驱动.....	2
1、基于 6410 的电阻屏驱动.....	2
1.1 硬件连接图.....	2
1.2 注册触屏设备.....	2
1.3 触屏驱动.....	4
2. 电阻式触摸屏的工作原理.....	5
2.1 四线式电阻式触摸屏原理.....	5

电阻屏驱动

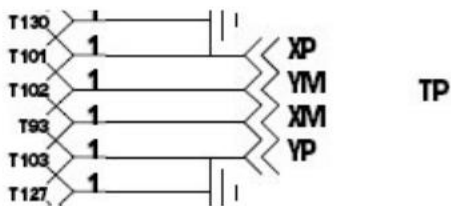
1、基于 6410 的电阻屏驱动

1.1 硬件连接图



XP、XM、YP、YM 对应了 ADC 采用的 4，5，6，7 输入接口。也就是说，XP、XM、YP、YM 是模拟信号，这就

1. 要求这四个接口的位置能触屏硬件的输出，连线不能太远，否则模拟信号可能会有干扰或者损失，造成触屏不准，这方面我们有过教训。
2. 还有一个问题是，可能有电池的 adc 采样存在，电池的 adc 采样一般是 8 位，而触屏的 adc 采样用的是 12 位，要防止电池采样修改触屏采样的这一个参数。
3. 我在 wince 系统，调试触屏时，发现有中断丢失的现象，原因是在 PENDWON 中断未处理完时，定时器中断又来了，这时会丢一个中断，调试触屏时要关注这个问题。



XP、XM、YP、YM 是电阻屏引出的四个接口，可以测量它们的电压，当触笔在不同点点击时，XP、YP 会发生电压的变化，依据此点可以判断触屏是否正确工作。

1.2 注册触屏设备

参见[\[mach-s3c64xx/Dev-ts.c\]](#)文件中

```
/* Touch screen */
```

```
static struct resource s3c_ts_resource[] = {
    [0] = {
        .start = SAMSUNG_PA_ADC,
        .end    = SAMSUNG_PA_ADC + SZ_256 - 1,
        .flags = IORESOURCE_MEM,
    },
    [1] = {
        .start = IRQ_PENDN,
        .end    = IRQ_PENDN,
        .flags = IORESOURCE_IRQ,
    },
    [2] = {
        .start = IRQ_ADC,
        .end    = IRQ_ADC,
        .flags = IORESOURCE_IRQ,
    }
};
```

Touch 关注两个中断，其一是触笔按下 **IRQ_PENDN**，其二是 **IRQ_ADC**

```
struct platform_device s3c_device_ts = {
    .name          = "s3c-ts",
    .id            = -1,
    .num_resources = ARRAY_SIZE(s3c_ts_resource),
    .resource       = s3c_ts_resource,
};
```

```
Static struct platform_device *smdk6410_devices[] __initdata = {
    ...
    & samsung_device_ts,
    ...
}
```

```
void __init s3c_ts_set_platdata(struct s3c_ts_mach_info *pd)
{
    struct s3c_ts_mach_info *npd;
    .....
    npd = kmemdup(pd, sizeof(struct s3c_ts_mach_info), GFP_KERNEL);
    s3c_device_ts.dev.platform_data = npd;
}
```

[Mach-smdk6410.c]

Smdk6410_machine_init

S3c_ts_set_platdata(&s3c_ts_platform)

在这里， **s3c_device_ts** 关联了它的私有数据 **s3c_ts_platform**

```
static struct s3c_ts_mach_info s3c_ts_platform __initdata = {
    .delay          = 10000,
    .presc           = 49,
    .oversampling_shift= 2, //取数之前丢掉的采样点数, (1<<2)
    .resol_bit       = 12, //adc 采样 12 位
    .s3c_adc_con     = ADC_TYPE_2,
};
```

1.3 触屏驱动

Kernel 模块选中

Device Drivers ->

Input device support ->

Touchscreens -> //选为内置

S3c touchscreen driver //选中

查看 Help, Symbol: TOUCHSCREEN_S3C [=y]

参考 Kconfig 和 Makefile 知道,

S3c-ts.c 被编进内核, 是触屏的驱动文件。

S3c_ts_driver 定义了触屏的函数指针操作。最主要的函数是 s3c_ts_probe [s3c_ts_probe]看看初始化工作, 都做了些什么。

1. 首先是 LCD 的大小, SCREEN_X=800, SCREEN_Y=480
2. 内存和 IO 资源的分配, ts_mem, ts_base
3. 时钟

Ts_clock = clk_get("adc");

Clk_enable(ts_clock); 【1】

4. 寄存器相关的初始化直接在 probe 中进行。

具体如何配置参考文档 (s3c6410_rev12.pdf)。

【1】adc 的时钟也来自 PCLK, 开关位: S3C_CLKCON_PCLK_TSADC

6410 触屏工作原理

1. 用户点击触屏, 产生 IRQ_PENDN 的中断。
2. IRQ_PENDN 中断处理函数

Stylus_updown

如果触笔压下, 立即进入定时器处理 touch_timer_fire(0)

清除中断, 不管触笔是抬起还是压下。

3. 定时器处理函数 touch_timer_fire

- a. 如果触笔压下,

将原始电压值通过校屏参数转换为屏幕点

通过 input_reput_xx, 向 input 系统上报一些事件。

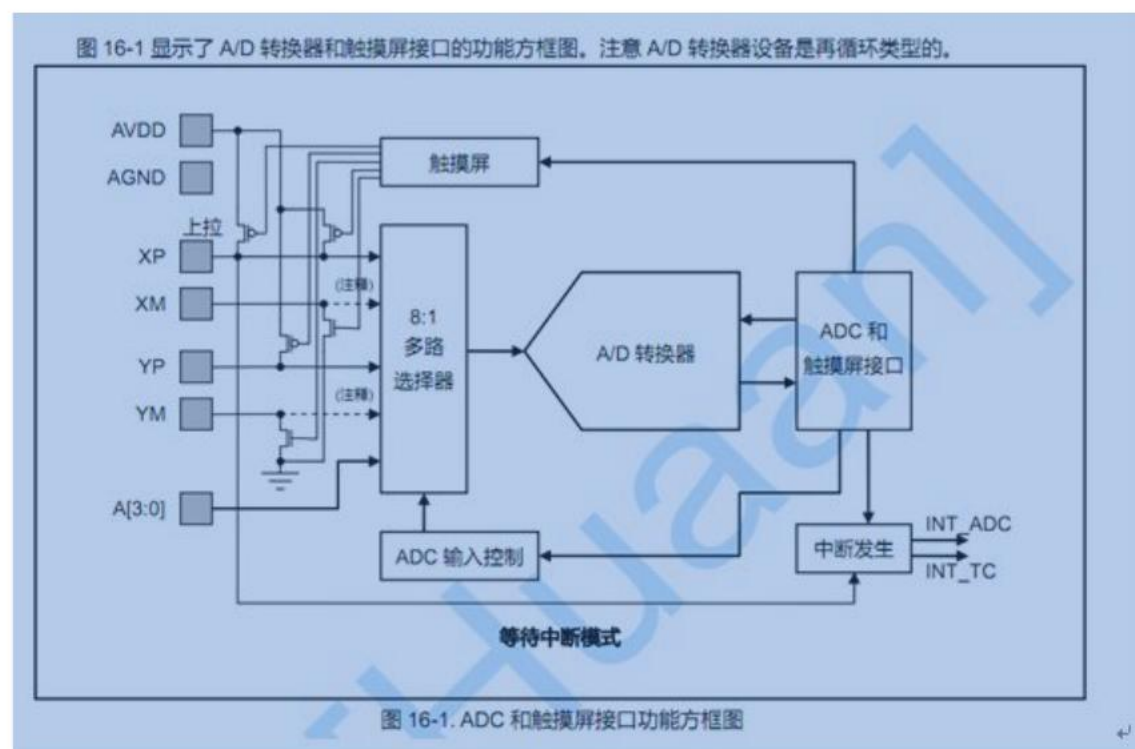
开始 adc 采样, 等待采样完成, adc 中断 IRQ_ADC 被触发。

- b. 如果触笔是抬起状态, 则停止 adc 采样

4. IRQ_ADC 中断处理中断 `stylus_action`
 - a. 采样完成，读取采样数据。
 - b. 看看采样点数是否够，如果不够，再次开始 adc 采样，等待采样完成，adc 中断 IRQ_ADC 被触发。
 - c. 依据 `ts->shift`，丢掉 $(1 < ts->shift)$ 个点，只取第 $(1 < ts->shift)$ 个点为有效点。
 - d. 启动定时器 `mod_timer(&touch_timer, jiffies+1)`
5. 等待定时器函数 `touch_timer_fire` 被触发

2. 电阻式触摸屏的工作原理

2.1 四线式电阻式触摸屏原理



S3C2440 而言：在直接试用触摸屏的是时，引脚 XP、XM、XP 和 YM 被用于和触摸屏直接相连。只剩下 AIN[3:0] 共 4 个引脚用于一般的 ADC 输入；当不使用的时候，XP、XM、XP 和 YM 也可用于一般的 ADC 输入。

S3C2440 的触摸屏接口可以驱动成四线电阻触摸屏，四线触摸屏的等效电路如图：14.4 所示。图中粗线表示相互绝缘的两层导电层，当按下时，他们触点处相连；不同的触点在 x，y 方向电压值不一样，将这两个值经过 A/D 转换后即可得到 X,Y 的坐标。

触摸屏工作过程：

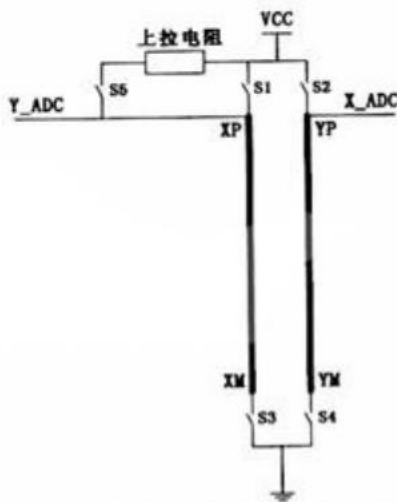


图 14.4 四线电阻触摸屏的等效电路

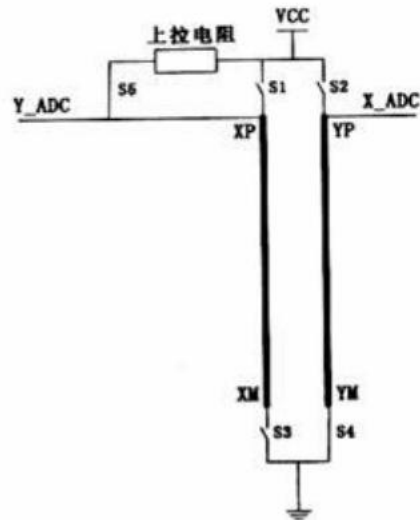


图 14.5 触摸屏处于“等待中断模式”时的等效电路

1. 触摸屏没有被按下时，等效电路如 14.5 所示

S4, S5 闭合, S1, S2, S3 断开, 即 YM 接地、XP 上拉、XP 作为模拟输入 (对 CPU 而言), YP 作为模拟输入 (对 CPU 而言), XM 高阻。

触摸屏没有按下的时候, Y_ADC 因为上拉, 为高电平, 当 X 轴和 Y 轴受挤压而接触导通后, Y_ADC 的电压由于连通到地变为低电平, 此低电平可作为中断触发信号来通知 CPU 发生“Pen Down”事件, 称为等待中断模式。

Pen Down/Up: 就是触摸屏按下 (松开), 都产生 INT_TC 中断信号, 在 ADCTSC 寄存器的位【8】为 0 或者 1 时, 表示等待 Pen Down/Up 中断。

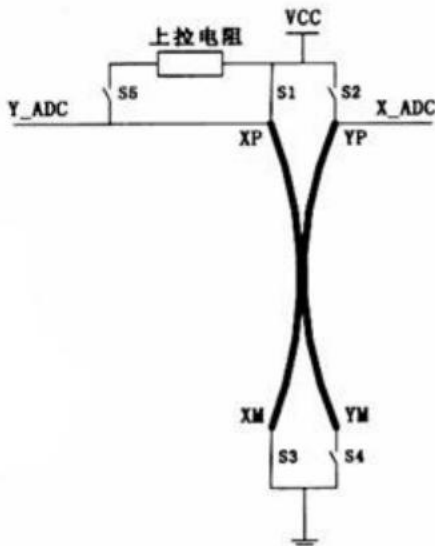


图 14.6 读取 x 坐标时的等效电路

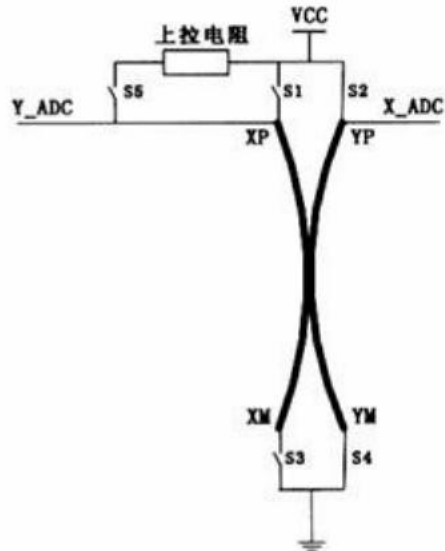


图 14.7 读取 y 坐标时的等效电路

2. 采集 X_ADC 电压, 得到 X 坐标, 等效电路如 14.6 所示。

S1, S3 闭合, S2, S4, S5 断开, 即 XP 接上电源, XM 接地, YP 作为输入 (对 CPU 而言)、YM 高阻、XP 禁止上拉。这时, YP 即 X_ADC 就是 X 轴的分压点, 进行 AD 转换后就

到 X 坐标。

3. 采集 Y_ADC 电压，得到 Y 坐标，等效电路如 14.7 所示。

S2、S4 闭合，S1，S3，S5 断开，即 YP 接上电源、YM 接地、XP 作为模拟输入（对 CPU 而言），XM 高阻，XP 禁止上拉，这时，XP 即 Y_ADC 就是 y 轴的分压点，进行 A/D 转换后就得到 y 坐标。