

### 1、触摸屏设备:

像玻璃一样的透明设备。与显示器吻合，放置在显示器表面的坐标输入设备



### 2、功能：

触摸屏是一个绝对坐标定位系统, 输入位置坐标信息。

派生功能：功能选择、图像放大与缩小、浏览等



### 3、应用：

#### (1)、位置输入

触笔点击触摸屏，即可输入触笔点击、位置在屏幕的坐标

#### (2)、功能选择事件

触笔点击触摸屏持续一段时间，即可输入功能选择事件。等

### 4、相似设备： 鼠标

### 5、触摸屏分类(按触摸点识别技术):

(1)、单点触摸屏: 单点识别

(2)、多点触摸屏: 多点识别

两只手, 多个手指, 甚至多个人, 同时操作屏幕的内容, 称为多点触摸屏, 此技术也叫多点触控技术(多点触摸技术)

### 6、触摸屏分类(按传感器技术):

- (1)、电阻式触摸屏
- (2)、表面声波式触摸屏
- (3)、红外式触摸屏
- (4)、电容式触摸屏

### (1)、电阻式触摸屏

**(A)功能:**

矩形区域中触摸点 (X, Y) 的物理位置表示电压坐标:

$X(0..V_{ref})$

$Y(0..V_{ref})$

**(B)感应方式:压力感应**

### (1)、电阻式触摸屏

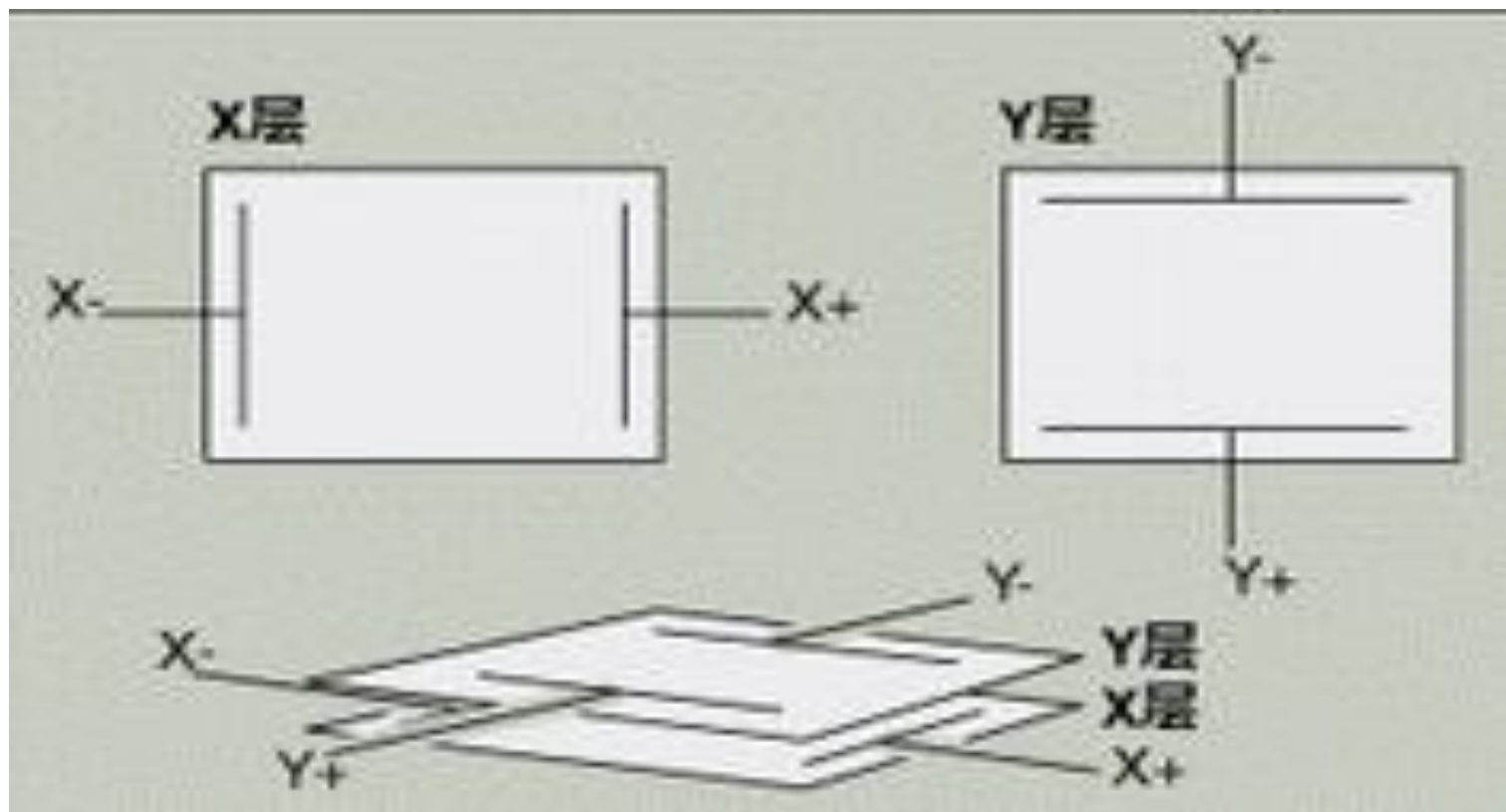
(C) 结构：一块电阻薄膜屏

ITO: Indium tin oxide 氧化  
铟锡（透明导电薄膜）



### (1)、电阻式触摸屏

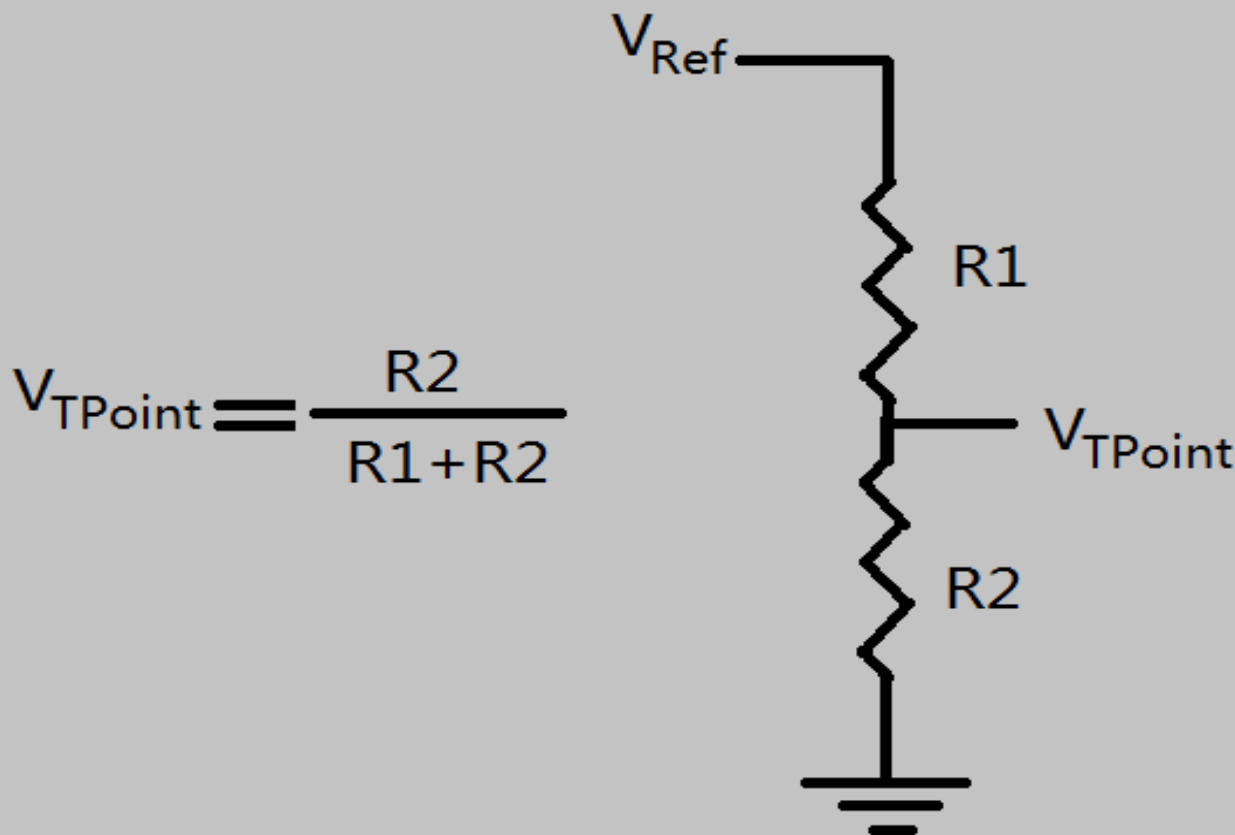
#### (D)四线触摸屏





### (1)、电阻式触摸屏

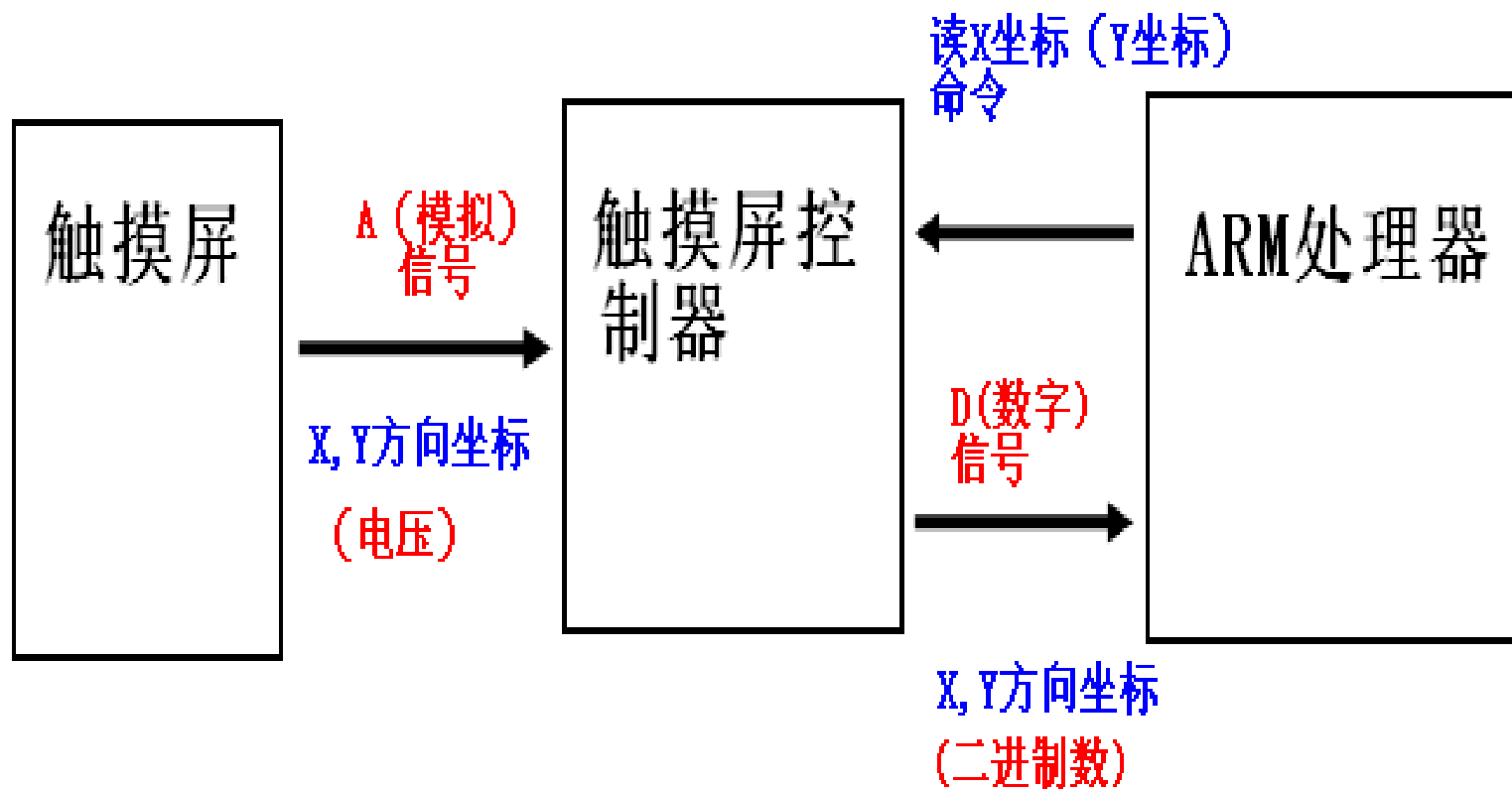
### (E)触摸点坐标计算公式



### 7、触摸屏特性：

- 透明度
- 色彩失真度
- 反光性
- 清晰度

## 8、触摸屏系统



# 触摸屏

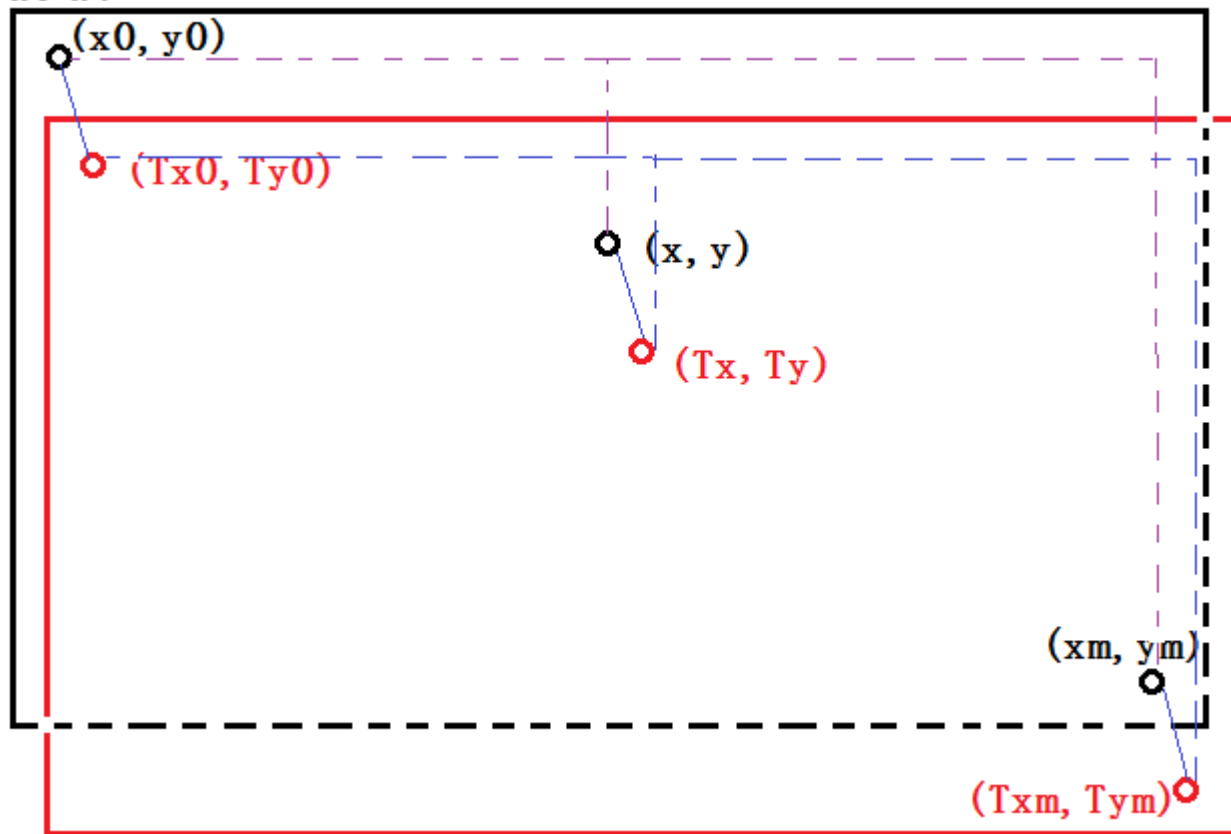
## 一、简介

$$(x-x_0) / (x_m-x_0) = (T_x-T_{x0}) / (T_{xm}-T_{x0})$$

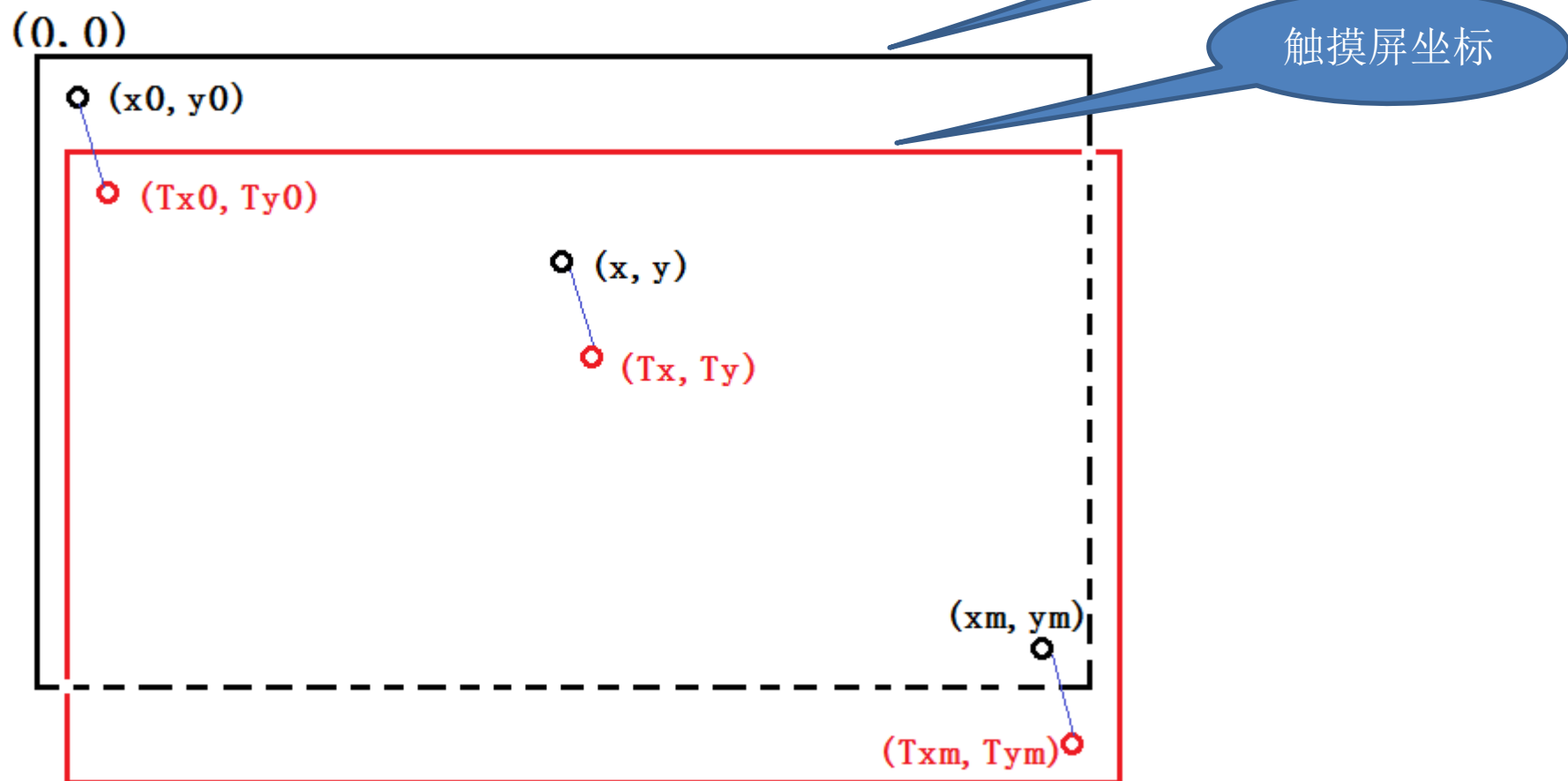
$$x = x_0 + (x_m - x_0) * (T_x - T_{x0}) / (T_{xm} - T_{x0})$$

$$y = y_0 + (y_m - y_0) * (T_y - T_{y0}) / (T_{ym} - T_{y0})$$

(0, 0)



### 9、触摸屏坐标与显示器坐标



### 1、接口电路功能

获取触摸屏X方向、Y方向坐标

已知 A/D 转换采用8位

X坐标 (00000000B...11111111B)

Y坐标 (00000000B...11111111B)

### 2、触摸屏与触摸屏控制器

#### (1) 触摸屏

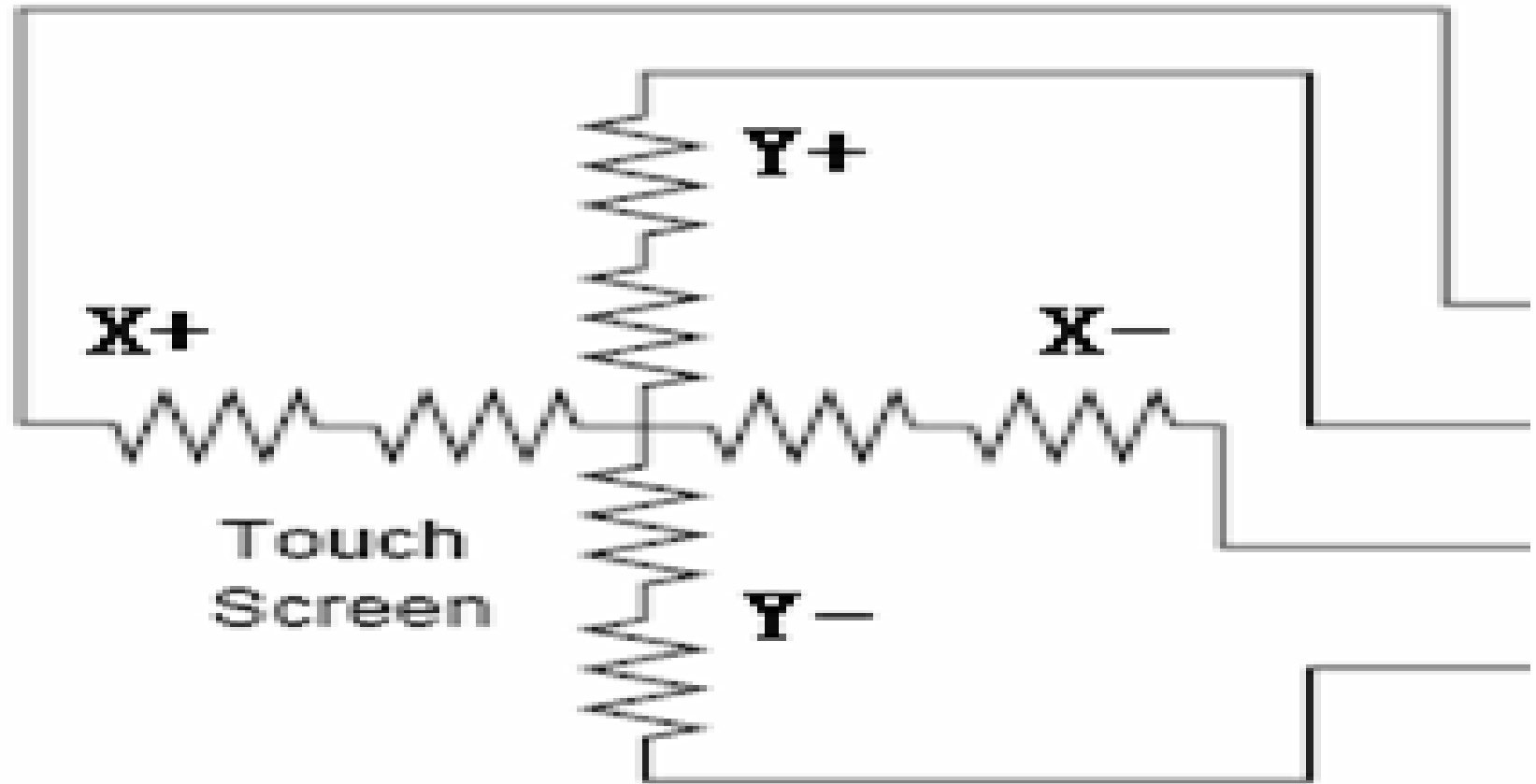
- 接收触摸操作
- 触摸点坐标

$X(0..V_{ref})$

$Y(0..V_{ref})$

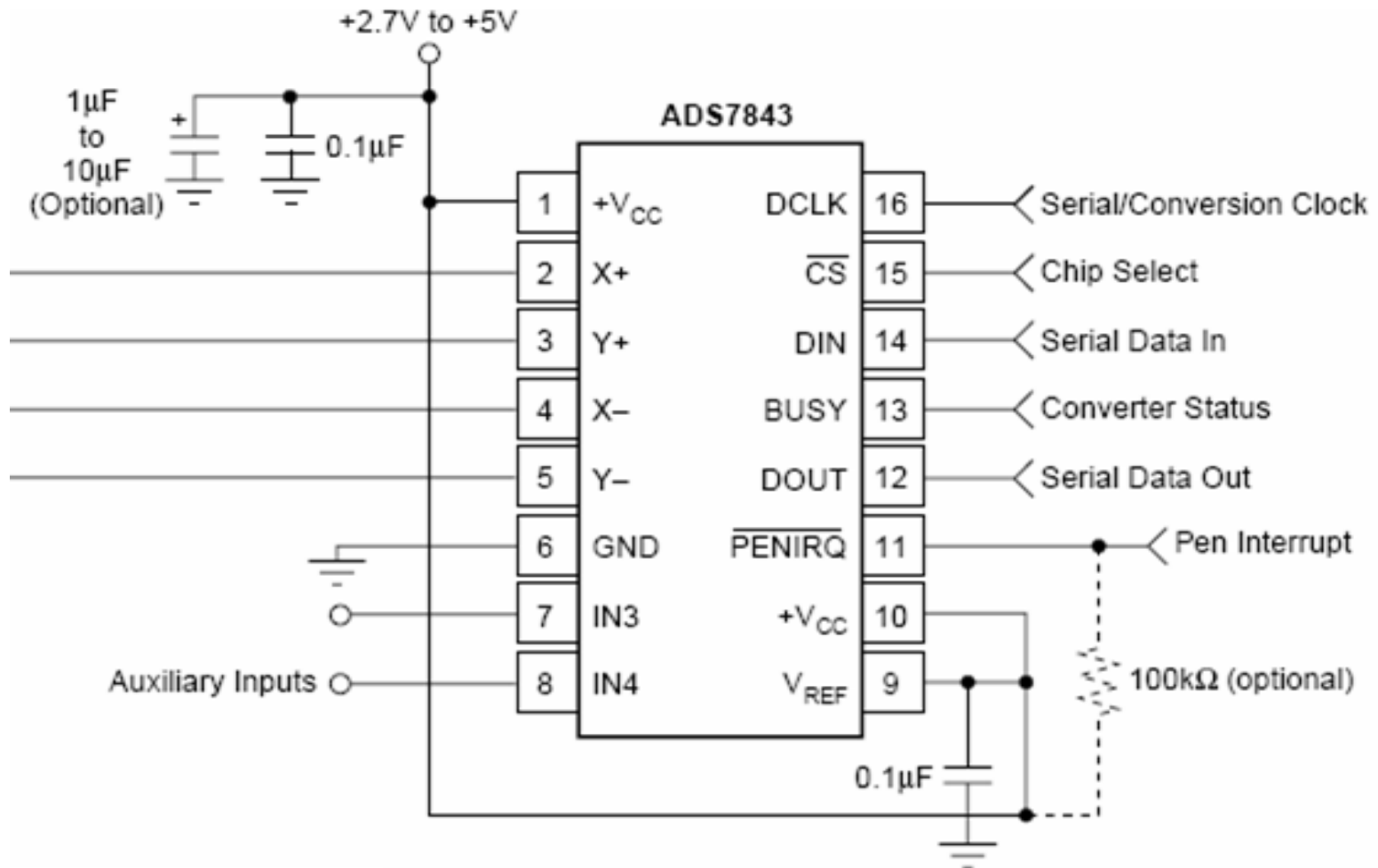
- 接口(四线):

- 接口(四线)(坐标输出):





### (2) 触摸屏控制器ADS7843



### (A) 与触摸屏连接信号

- X+、X-用于与触摸屏 (X+、X-) 或 (Y+、Y-) 连接，作为ADS7843的A/D转换输入信号，即X方向坐标信号
- Y+、Y-用于与触摸屏 (X+、X-) 或 (Y+、Y-) 连接，作为ADS7843的A/D转换输入信号，即Y方向坐标信号

### (B) 与S3C2440连接信号

- DIN用于接收来自S3C2440的读取触摸屏X(或Y)方向坐标命令,是一组由高、低电平组成的信号序列.

### 引脚特性:GPIO输入

- DOUT用于将A/D转换结果传送至来自S3C2440,是一组由高、低电平组成的信号序列

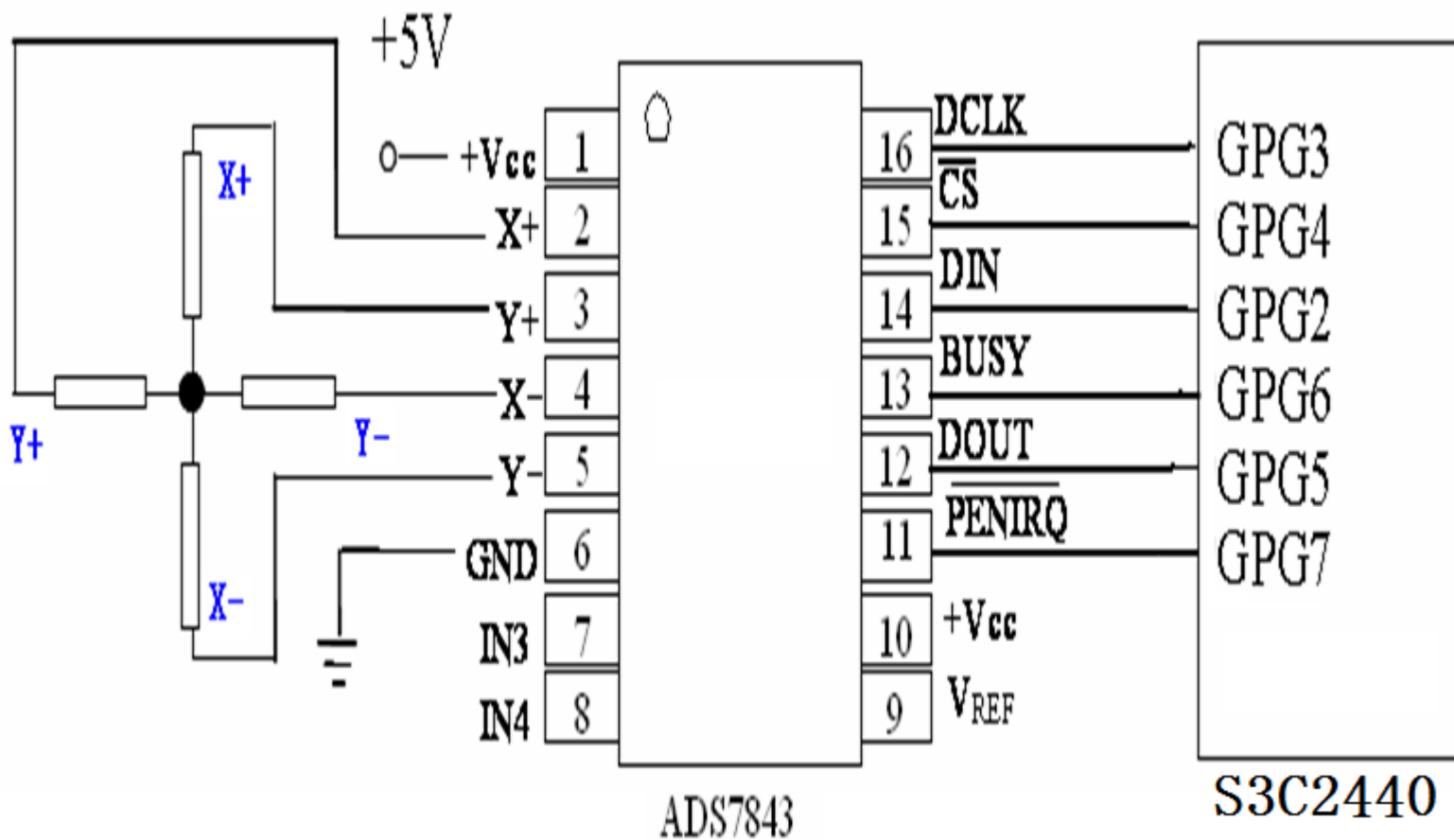
### 引脚特性: GPIO输出

- $\overline{CS}$  片选信号, 低电平有效  
引脚特性: GPIO输入
- DCLK, 时钟信号  
引脚特性: GPIO输入
- BUSY, ADS7843示忙  
引脚特性: GPIO输出
- $\overline{PENIRQ}$  , 中断信号, 低电平有效  
引脚特性: GPIO输出

### 3、S3C2440引脚需求

|               |             |        |
|---------------|-------------|--------|
| DIN           | GPI0输入→GPG2 | GPI0输出 |
| DOUT          | GPI0输出→GPG5 | GPI0输入 |
| DCLK          | GPI0输入→GPG3 | GPI0输出 |
| BUSY          | GPI0输出→GPG6 | GPI0输入 |
| <u>PENIRQ</u> | GPI0输出→GPG7 | GPI0输入 |
| <u>CS</u>     | GPI0输入→GPG4 | GPI0输出 |

### 4、接口电路



## 5、S3C2440处理器引脚初始化

| Register | Address    | R/W | Description                         | Reset |
|----------|------------|-----|-------------------------------------|-------|
| GPGCON   | 0x56000060 | R/W | Configures the pins of port G       | 0     |
| GPGDAT   | 0x56000064 | R/W | The data register for port G        | U     |
| GPGUP    | 0x56000068 | R/W | Pull-up disable register for port G | 0x    |

### 5、S3C2440处理器引脚初始化

| GPGCON | Bit     | Description                 |                               |
|--------|---------|-----------------------------|-------------------------------|
| GPG7   | [15:14] | 00 = Input<br>10 = EINT[15] | 01 = Output<br>11 = SPICLK1   |
| GPG6   | [13:12] | 00 = Input<br>10 = EINT[14] | 01 = Output<br>11 = SPIMOSI1  |
| GPG5   | [11:10] | 00 = Input<br>10 = EINT[13] | 01 = Output<br>11 = SPIMISO1  |
| GPG4   | [9:8]   | 00 = Input<br>10 = EINT[12] | 01 = Output<br>11 = LCD_PWRDN |
| GPG3   | [7:6]   | 00 = Input<br>10 = EINT[11] | 01 = Output<br>11 = nSS1      |
| GPG2   | [5:4]   | 00 = Input<br>10 = EINT[10] | 01 = Output<br>11 = nSS0      |
| GPG1   | [3:2]   | 00 = Input<br>10 = EINT[9]  | 01 = Output<br>11 = Reserved  |
| GPG0   | [1:0]   | 00 = Input<br>10 = EINT[8]  | 01 = Output<br>11 = Reserved  |



### 5、S3C2440处理器引脚初始化

| GPGDAT    | Bit    | Description   |
|-----------|--------|---|
| GPG[15:0] | [15:0] | <p>When the port is configured as an input port, the corresponding bit shows the pin state. When the port is configured as an output port, the pin state is the same as the corresponding bit.</p> <p>When the port is configured as functional pin, the undefined value is read.</p> |

| GPGUP     | Bit    | Description   |
|-----------|--------|---|
| GPG[15:0] | [15:0] | <p>0: The pull up function attached to the corresponding port pin is enabled.</p> <p>1: The pull up function is disabled.</p> |

### (1)、寄存器变量定义

```
#define rPGCON (*((volatile unsigned char *) 0x56000060))  
#define rGPDAT (*((volatile unsigned char *) 0x56000064))  
#define rGPUP (*((volatile unsigned char *) 0x56000068))
```

### (2)、配置内部上拉电阻电路

控制字=00000000B=0x00

配置语句:

```
rGPGPU=0x00;
```

(3)、 GPGCON=xxxx xxxx xxxx xxxx 0000 0001 0101 xxxx

|      |        |            |        |
|------|--------|------------|--------|
| GPG7 | GPI0输入 | [15:14]=00 | Input  |
|      | 或      | [15:14]=00 | EINT   |
| GPG6 | GPI0输入 | [13:12]=00 | Input  |
| GPG5 | GPI0输入 | [11:10]=00 | Input  |
| GPG4 | GPI0输出 | [9:8] =01  | Output |
| GPG3 | GPI0输出 | [7:6] =01  | Output |
| GPG2 | GPI0输出 | [5:4] =01  | Output |

控制字=0x00000150

配置语句:

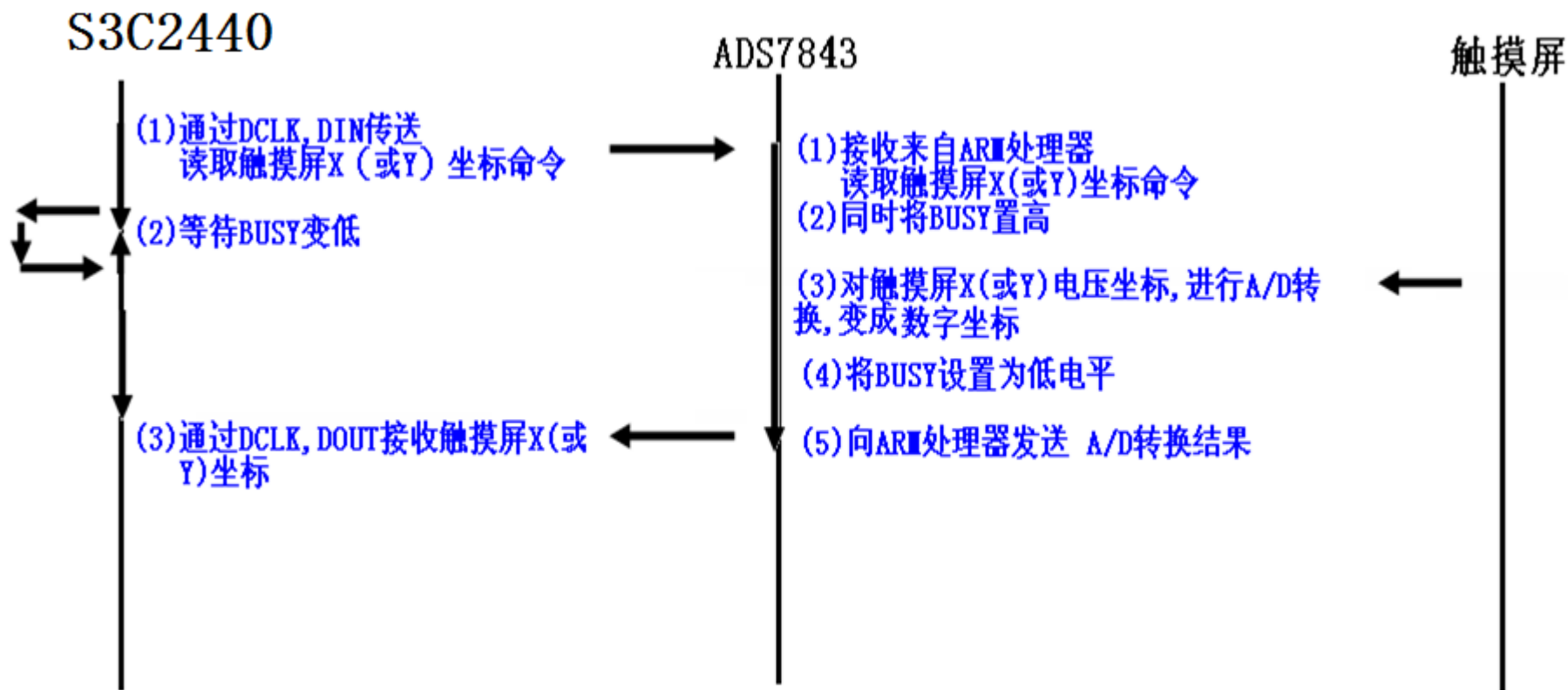
```
rGPGCON=0x000000150;
```

### (4) G口初始化函数

```
void PortG_Init(void)
{
    rGPGCON=0x00000150;
    rGPGUP=0x0000;
}
```

### 6、编程

#### (1)、S3C2440与触摸屏控制交互过程



### (2)、S3C2440向触摸屏发送读取X, Y坐标命令

#### (A) 命令字格式

| BIT | NAME    | DESCRIPTION  |
|-----|---------|--|
| 7   | S       | Start Bit. Control byte starts with first HIGH bit on DIN. A new control byte can start every 16th clock cycle in 12-bit conversion mode or every 12th clock cycle in 8-bit conversion mode. |
| 6-4 | A2-A0   | Channel Select Bits. Along with the SER/DFR bit, these bits control the setting of the multiplexer input, switches, and reference inputs, see Tables I and II.                               |
| 3   | MODE    | 12-Bit/8-Bit Conversion Select Bit. This bit controls the number of bits for the following conversion: 12 bits (LOW) or 8 bits (HIGH).   |
| 2   | SER/DFR | Single-Ended/Differential Reference Select Bit. Along with bits A2-A0, this bit controls the setting of the multiplexer input, switches, and reference inputs, see Tables I and II.          |
| 1-0 | PD1-PD0 | Power-Down Mode Select Bits. See Table V for details.  |

### (B) 组合读取X方向坐标命令

- D7: S=1
- D6 D5 D4: A2 A1 A0=101 选择X坐标
- D3: 转换精度 MODE=1 8位模式
- D2: SER/DFR=1 单端模式
- D1 D0: PD1 PD0=00 省电模式
- 控制字=1 101 1 1 00=0xDC
- #define TOUCH\_MSR\_X 0xDC



### (c) 组合读取y方向坐标命令

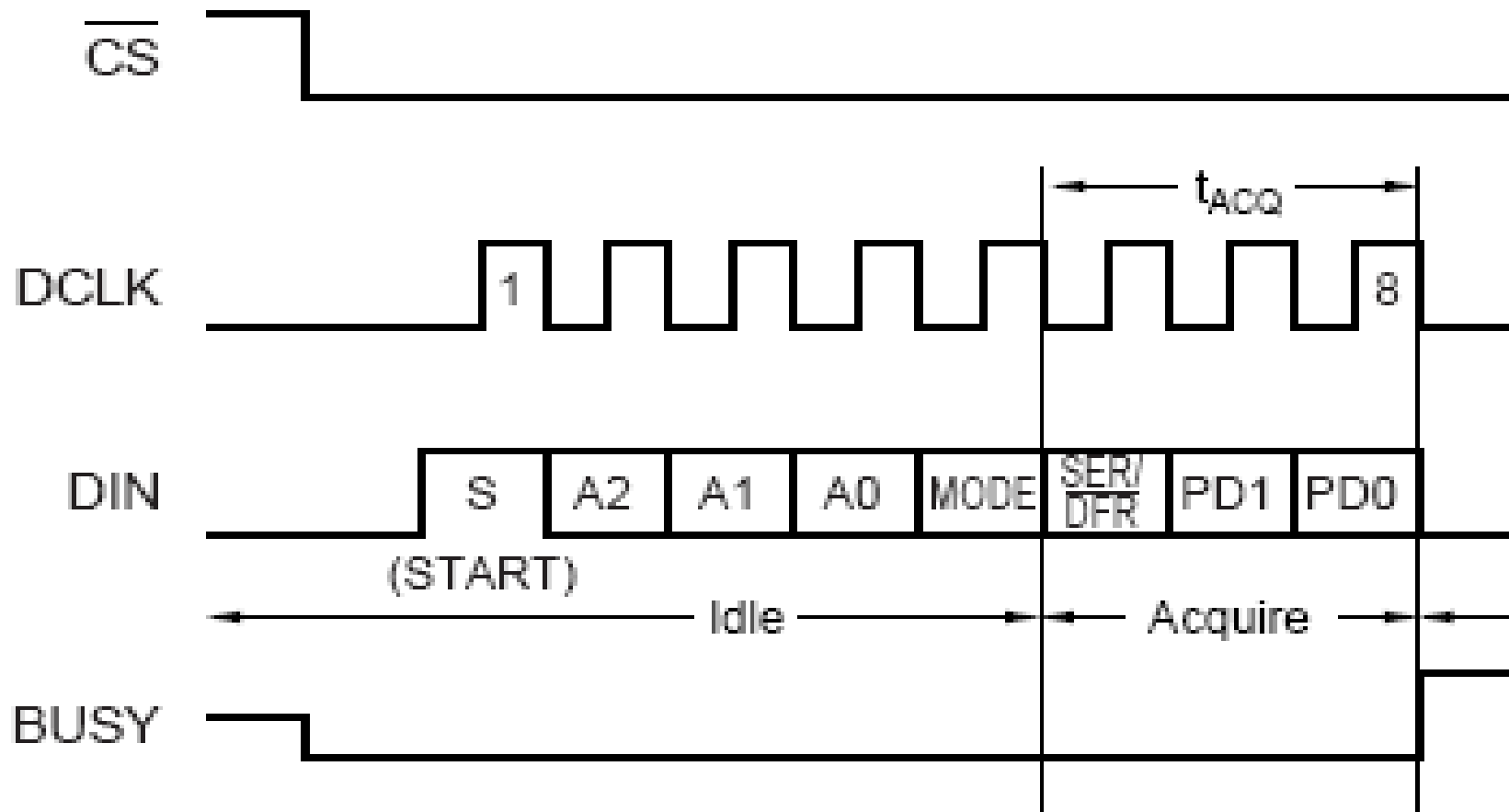
- D7: S=1
- D6 D5 D4: A2 A1 A0=001 选择X坐标
- D3: 转换精度 MODE=1 8位模式
- D2: SER/DFR=1 单端模式
- D1 D0: PD1 PD0=00 省电模式
- 控制字=1 001 1 1 00=0x9C
- #define TOUCH\_MSR\_Y 0x9C

# 触摸屏

## 二、触摸屏接口电路

| A2 | A1 | A0 | X+  | Y+  | IN3 | IN4 | -IN | X开关 | Y开关 | +REF              | -REF |
|----|----|----|-----|-----|-----|-----|-----|-----|-----|-------------------|------|
| 0  | 0  | 1  | +IN |     |     |     | GND | OFF | ON  | +V <sub>REF</sub> | GND  |
| 1  | 0  | 1  |     | +IN |     |     | GND | ON  | OFF | +V <sub>REF</sub> | GND  |
| 0  | 1  | 0  |     |     | +IN |     | GND | OFF | OFF | +V <sub>REF</sub> | GND  |
| 1  | 1  | 0  |     |     |     | +IN | GND | OFF | OFF | +V <sub>REF</sub> | GND  |

### (D) S3C2440向ADS7843发送命令时序图



### (E) S3C2440向ADS7843发送命令编程

#### 要点

- 判断当前触摸屏是否有触摸动作PENIRQ

```
if ((rGPGDAT & 0x80) == 0)
```

```
{
```

```
//判断ADS7843是否忙
```

```
//不忙, 发送读取X, Y 坐标命令
```

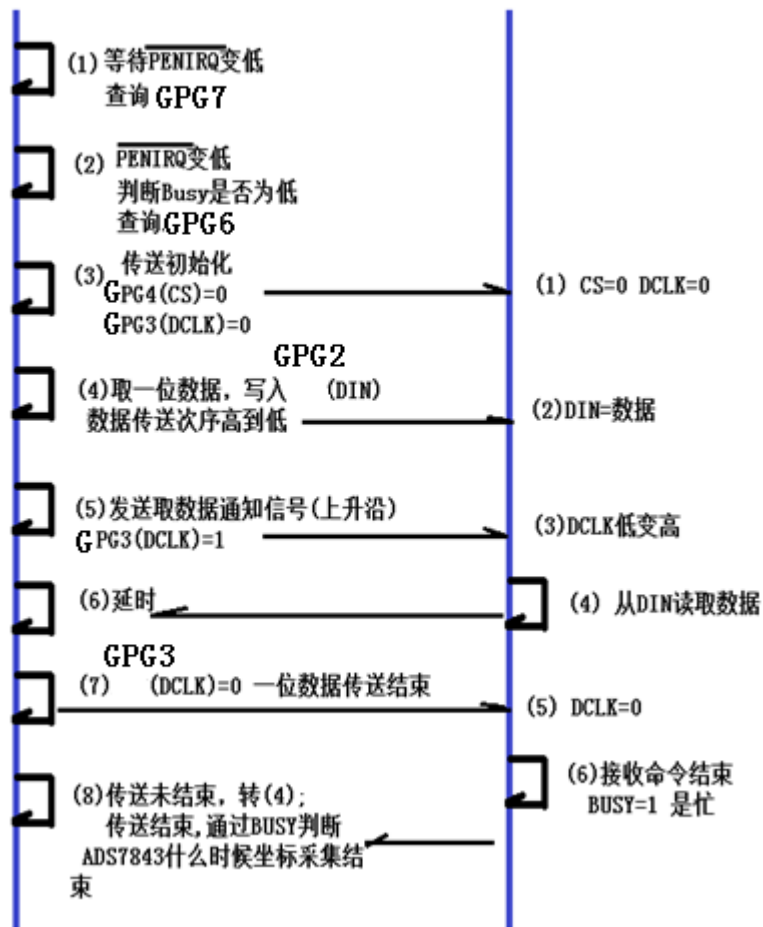
```
}
```

```
//GPG7, PENIRQ 电平为低有触摸动作
```

- 命令传送过程

S3C2440

ADS7843



# 触摸屏

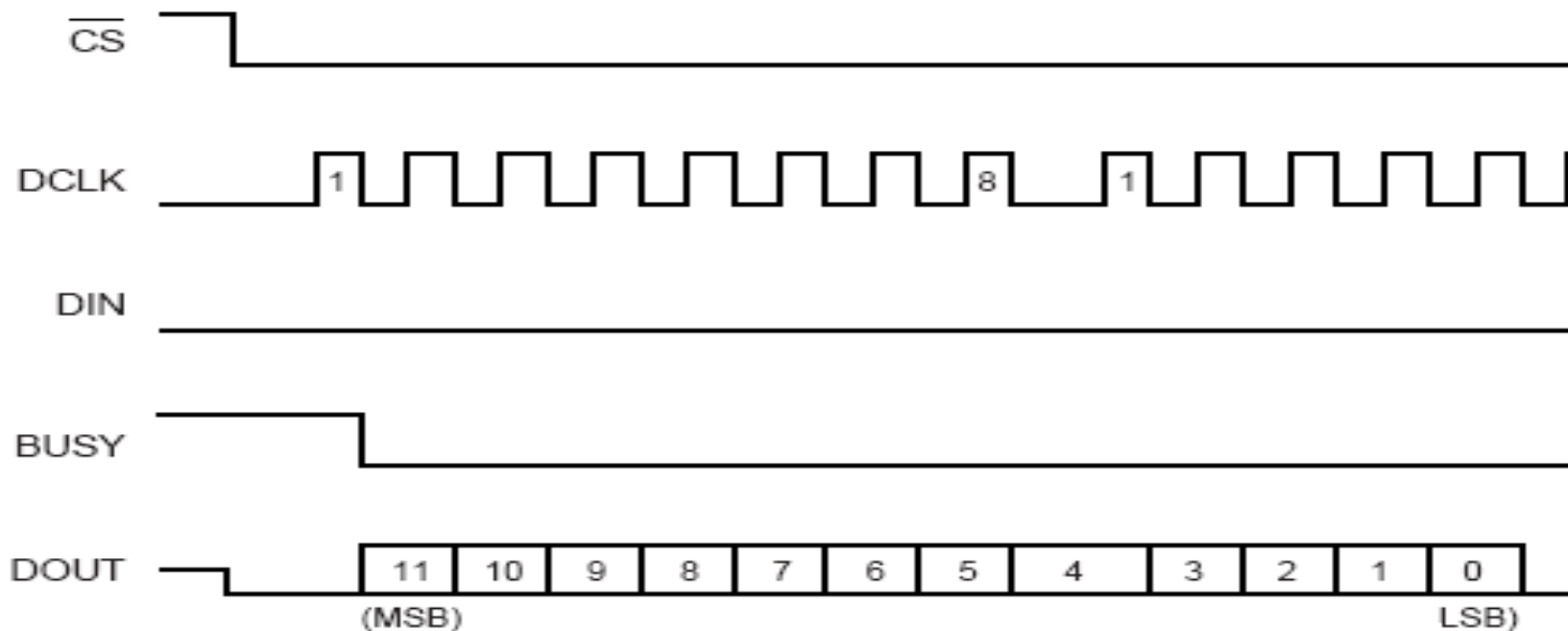
## 二、触摸屏接口电路

- 命令传送程序段

```
rGPGDAT &= 0xe7;    //CS置低;DCLK置低, GPG4
temp =0x80;          //设置要传送的位
for(i=0; i<8; i++)    //发送命令字节
{
    if(command&temp))  rGPGDAT|=0x04;//DIN=1
    else    rGPGDAT&=0xfb; //DIN=0, GPG2
    rGPGDAT |= 0x08;      //DCLK置高上升沿, GPG3
    delay(2);
    rGPGDAT&= 0xf7;      //清除DCLK, 1位送出, GPG3
    delay(2);
    temp =temp>>1;      //右移1位
}
```

(3)、S3C2440接收触摸屏X, Y坐标

(A)、S3C2440从ADS7843接收数据时序图



(B) 判断ADS7843是否忙 BUSY

```
if ((rGPGDAT & 0x40) == 0)
```

```
{
```

```
    //读取触摸屏坐标代码
```

```
}
```

```
    //GPG6, BUSY电平为高时, ADS7843忙
```



### (C) S3C2440从ADS7843接收数据过程

1) 条件  $GPG6 (BUSY) = 0$

2) 初始化, 准备接收数据

$GPG5 (DOUT) = 0$        $GPG3 (DCLK) = 0$

3) 通知ADS7843向DOUT写数据

$GPG3 (DCLK) = 1$

4) 延时2个单位

5)  $GPG3 (DCLK) = 0$

6) 延时2个单位

7) S3C2440准备读数据 DCLK=1, 通知

ADS7843向DOUT写数据

8) 从GPG5 读取数据

9) 延时2个单位

10) GPG3 (DCLK)=0

11) 延时2个单位

12) 是否是接收结束, 未结束转7)

13) GPG4 (CS)=1, ADS7843无效

14) 保存接收数据

### (D) S3C2440从ADS7843接收数据程序段

```
rGPGDAT|=0x08;          //DCLK 置高平, GPG3=1
delay(2);
rGPGDAT&=0xf7; //DCLK 低电平 , 下降沿 GPG3=0
delay(2);
ack=0;
for(i=0;i<8;i++) //取得8位坐标数据
{
    rGPGDAT|=0x08;      //DCLK置高, GPG3
    ack=ack<<1;
    if (rGPGDAT & 0x20)  ack+=1;
    delay(2);
    rGPGDAT&=0xf7; //DCLK 低电平 , GPG3=0
    delay(2);
}
rGPGDAT|=0x10;          //CS=1
```

### (4)、读取触摸屏坐标函数

- 入口参数：读取X(或Y)坐标命令
- 返回值：触摸屏X(或Y)坐标
- 出口参数：无
- 功能：读取触摸屏X(或Y)坐标
- 函数声明

```
Int ReadTouch(unsigned char CMD)
```

# 触摸屏

## 二、触摸屏接口电路

```
Int ReadTouch(unsigned char CMD)
{
    unsigned char temp, i, j, k, ack=0;
    rGPGDAT &= 0xe7;    //CS置低;DCLK置低, GPG4
    temp =0x80;         //设置要传送的位
    for(i=0; i<8; i++)    //发送命令字节
    {
        if(command&temp)    rGPGDAT|=0x04; //DIN=1
        else    rGPGDAT&=0xfb; //DIN=0, GPG2
        rGPGDAT |= 0x08;    //DCLK置高上升沿, GPG3
        delay(2);
        rGPGDAT&= 0xf7;    //清除DCLK, 1位送出, GPG3
        delay(2);
        temp =temp>>1;    //右移1位
    }
}
```

# 触摸屏

## 二、触摸屏接口电路

```
While((temp=(rGPGDAT&0x40))==0);  
rGPGDAT|=0x08;          //DCLK 置高平, GPG3=1  
delay(2);  
rGPGDAT&=0xf7; //DCLK 低电平, 下降沿 GPG3=0  
delay(2);  
ack=0;  
for(i=0;i<8;i++) //接收8位坐标数据  
{   rGPGDAT|=0x08;    //DCLK置高, GPG3  
    ack=ack<<1;  
    if (rGPGDAT & 0x20)  ack+=1;  
    delay(2);  
    rGPGDAT&=0xf7; //DCLK 低电平, GPG3=0  
    delay(2);      }  
rPDATG|=0x10;        //CS=1
```

### 7、关注问题：

- (1)、触摸屏与触摸屏控制器之间的连线
- (2)、触摸屏控制器与ARM处理器之间的连线
- (3)、触摸屏坐标系与显示器坐标系，以及转换
- (4)、从触摸屏控制器读取的坐标数据与模拟量之间的关系
- (5)、向触摸屏控制器发送读取触摸屏X（Y）坐标命令程序

信号控制指令

发送数据形成

- (6)、从触摸屏控制器接收触摸屏X（Y）坐标数据

信号控制指令

接收数据形成