# mini2440裸机试炼之—计算器(LCD显示,触摸屏中断)

**Pait\_Bmp(x0, y0, x, y, BMPaddr);函数**

**像素图在屏幕左上角为（0,0），Pait\_Bmp中的x0，y0分别像素点初始位置，x，y为BMP图片的X，Y的大小，BMPaddr为BMP图片转换的数组地址**

**基于T35 TFT LCD屏实现功能**：

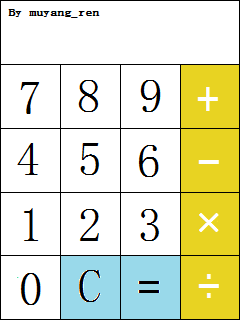
1. 在屏幕上画一个计算器界面，包括0-9，+，-，\*，/；

2. 实现触摸选择界面上的数字和运算符，

3. 并计算出结果显示在显示框内。

注意，只实现整数功能，没涉及小数，只实现一次运算，不保存上次运算结果。

**实现界面:**



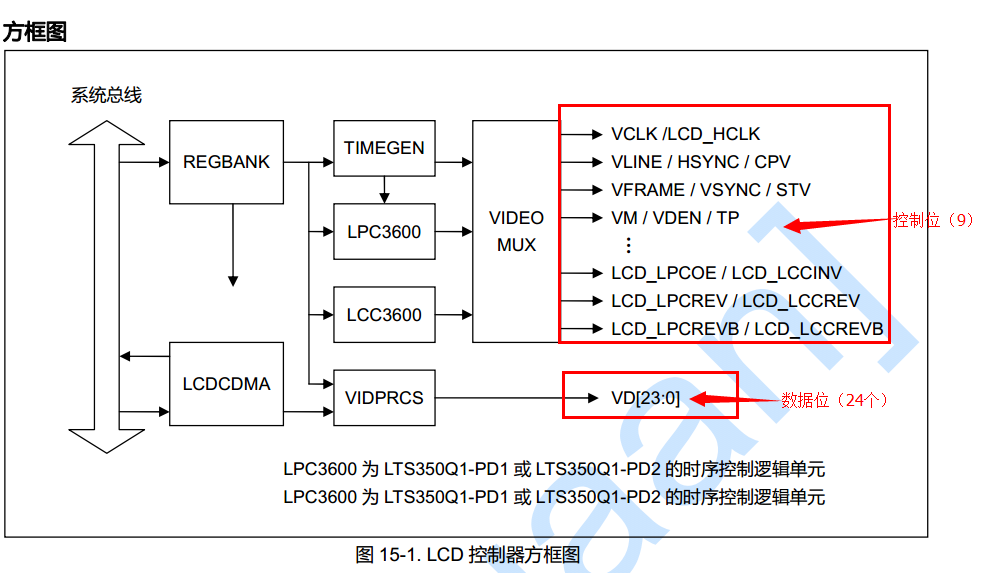
**实现过程:**

开启LCD电源,初始化LCD >>>LCD填充界面图片>>>开启触摸屏中断>>>获取键值>>>区分数字和功能来调用数字图片显示。

**数字图片**：



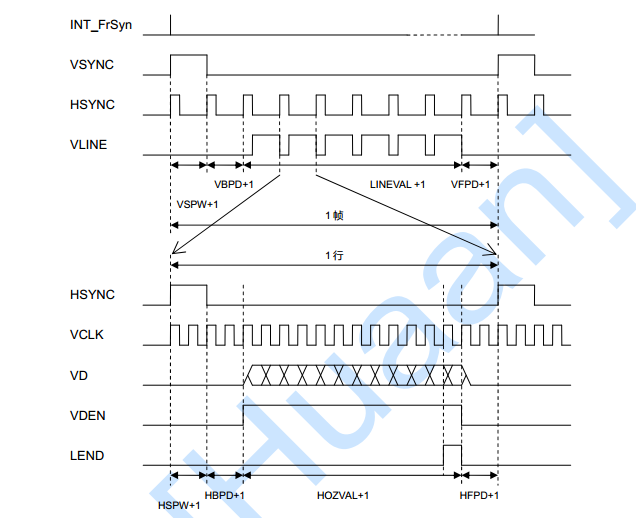
## 开启LCD电源,初始化LCD



其中：时序发生器（TIMEGEN）

TIMEGEN产生LCD驱动器的控制信号，如VFRAME、VLINE、VCLK和VM。这些控制信号与REGBANK中的LCDCON1/2/3/4/5寄存器配制有着紧密关系。基于这些可编程的REGBANK中LCD控制寄存器的配制，TIMEGEN可以产生合适的可编程控制信号来支持多种不同类型的LCD驱动器.

TFT屏的工作时序：



VSYNC为帧同步信号，每发出一个脉冲表示新的一屏图像数据开始传输。

VSF＝HSF÷[(VSPW＋1)＋(VBPD＋1)＋(VFPD＋1)＋(LINEVAL＋1)]

HSYNC为行同步信号，每发出一个脉冲表示新的一行图像数据开始传输。

HSF＝VCLK÷[(HSPW＋1)＋(HSPD＋1)＋(HFPD＋1)＋(HOZVAL＋1)]

VCLK为像素同步信号，每发出一个脉冲表示新的一个点图像数据开始传输。

VCLK＝HCLK÷[(CLKVAL＋1)×2]

LCD显示是一帧一帧（一个画面）的，每一帧里显示又是从上到下一行一行的，每一行显示又是从左到右一个点一个点的。而VSYNC，HSYNC，VCLK这些决定了他们的显示速度。

图中的VSPW,HSPW等决定相应脉冲的宽度，VBPD,HBPD决定了延时时间。

这些参数的设置都是由LCDCONn决定的。

初始化LCD：(设置TIMEGEN)

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD功能模块初始化//此函数没开启屏蔽功能  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void LCD\_Init**(**void**)**  **{**  #define M5D(n) ((n)&0x1fffff)  #define LCD\_ADDR ((U32)LCD\_BUFFER)  //con1[6:5]显示模式选择为 TFT LCD面板 模式 [4:1]选择为 TFT的16bpp 像素模式    rLCDCON1 **=** **(**LCD\_PIXCLOCK **<<** 8**)** **|** **(**3 **<<** 5**)** **|** **(**12 **<<** 1**);**  //con2[23:14]设置高度320 VSPW[5:0]决定相应脉冲的宽度; VFPD[13:6]、VBPD[31:24]决定了延时时间; 选择option.h文件中T35定义参数  rLCDCON2 **=** **(**LCD\_UPPER\_MARGIN **<<** 24**)** **|** **((**LCD\_HEIGHT **-** 1**)** **<<** 14**)** **|** **(**LCD\_LOWER\_MARGIN **<<** 6**)** **|** **(**LCD\_VSYNC\_LEN **<<** 0**);**  //con3[18:8]设置高度240 HFPD、HBPD决定了延时时间; 选择option.h文件中T35定义参数  rLCDCON3 **=** **(**LCD\_RIGHT\_MARGIN **<<** 19**)** **|** **((**LCD\_WIDTH **-** 1**)** **<<** 8**)** **|** **(**LCD\_LEFT\_MARGIN **<<** 0**);**  //con4 HSPW[8:0]决定相应脉冲的宽度; 选择option.h文件中T35定义参数  rLCDCON4 **=** **(**13 **<<** 8**)** **|** **(**LCD\_HSYNC\_LEN **<<** 0**);**  #if !defined(LCD\_CON5)  # define LCD\_CON5 ((1<<11) | (1 << 9) | (1 << 8) | (1 << 3) | (1 << 0))  #endif  rLCDCON5 **=** LCD\_CON5**;**  rLCDSADDR1 **=** **((**LCD\_ADDR **>>** 22**)** **<<** 21**)** **|** **((**M5D**(**LCD\_ADDR **>>** 1**))** **<<** 0**);**  rLCDSADDR2 **=** M5D**((**LCD\_ADDR **+** LCD\_WIDTH **\*** LCD\_HEIGHT **\*** 2**)** **>>** 1**);**  rLCDSADDR3 **=** LCD\_WIDTH**;**  rLCDINTMSK **|=** 3**;** //中断屏蔽  rTCONSEL **&=** **(~**7**);**  rTPAL **=** 0x0**;**  rTCONSEL **&=** **~((**1**<<**4**)** **|** 1**);**    **}** |

使能LCD

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  LCD视频和控制信号输出或者停止，1开启视频输出  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Lcd\_EnvidOnOff**(**int onoff**)**  **{**  **if(**onoff**==**1**)**  rLCDCON1**|=**1**;** // ENVID=ON  **else**  rLCDCON1 **=**rLCDCON1 **&** 0x3fffe**;** // ENVID Off  **}** |

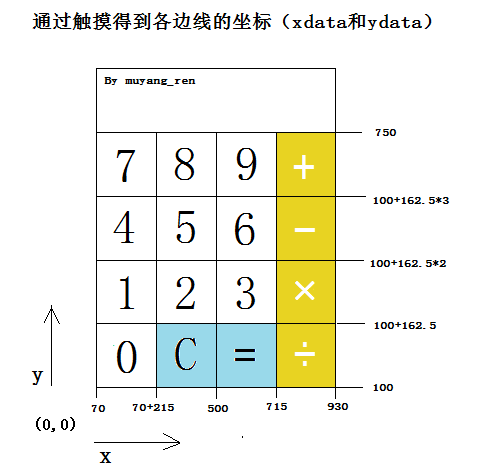
开启LCD电源

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD 电源控制引脚使能  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Lcd\_PowerEnable**(**int invpwren**,**int pwren**)**  **{**  //GPG4 is setted as LCD\_PWREN  rGPGUP **=** rGPGUP**|(**1**<<**4**);** // Pull-up disable  rGPGCON **=** rGPGCON**|(**3**<<**8**);** //GPG4=LCD\_PWREN    //Enable LCD POWER ENABLE Function  rLCDCON5 **=** rLCDCON5**&(~(**1**<<**3**))|(**pwren**<<**3**);** // PWREN  rLCDCON5 **=** rLCDCON5**&(~(**1**<<**5**))|(**invpwren**<<**5**);** // INVPWREN  **}** |

## 开启触摸屏中断

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD \*触摸屏中断函数\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/    void \_\_irq Adc\_Tc\_Handler**(**void**)**  **{**    rADCTSC**|=(**1**<<**3**)|(**1**<<**2**);** //XP上拉电阻无效, 自动连续测量X坐标和Y坐标.  rADCCON**|=(**1**<<**0**);**//ADC转换开始    **while(**rADCCON**&(**1**<<**0**));**//检测ADC转换是否开始且ADCCON[0]自动清0  **while(!(**rADCCON**&(**0x1**<<**15**)));** //检测ADCCON[15]是否为1,ADC转换是否结束,(必须)  **while(!(**rINTPND**&((**U32**)**0x1**<<**31**)));**//检测ADC中断是否已请求    xdata**=**rADCDAT0**&**0x3ff**;**//读x坐标 >>xdata并不是像素点，而是模拟信号 0-1000  ydata**=**rADCDAT1**&**0x3ff**;**//读y坐标    Uart\_Printf**(**"\n Xdata=%04d, Ydata=%04d\n"**,** xdata**,** ydata**);**    rSUBSRCPND**|=(**0x1**<<**9**);** //清除中断  rSRCPND**|=((**U32**)**0x1**<<**31**);**  rINTPND**|=((**U32**)**0x1**<<**31**);**    rADCTSC **=**0xd3**;** //ADC等待中断模式  rADCTSC**|=(**0x1**<<**8**);** //ADCTSC[8]=1,设置抬起中断信号    **while(!(**rSUBSRCPND**&(**0x1**<<**9**)));** //检测触屏抬起中断是否已请求    rADCTSC **&=~(**0x1**<<**8**);**//ADCTSC[8]=0光标按下中断信号    // 由于下面这段代码和上面这段代码是看到大神们都会加上去的，调试得到  // 结论：上面的清除中断实现触屏中断，而下面这段则是加快中断后的响应  // (我的实验结果是：没下面的时候，触屏后的数值显示时间变长，蜂鸣器的  // 响声也变长，有种慢一拍的感觉)    rSUBSRCPND**|=(**0x1**<<**9**);**  rSRCPND**|=((**U32**)**0x1**<<**31**);**  rINTPND**|=((**U32**)**0x1**<<**31**);**  **}**    void Touch\_Init**(**void**)**  **{**  rADCCON**=((**1**<<**14**)|(**9**<<**6**));** //A/D分频时钟有效，其值为9  rADCTSC**=**0xd3**;** //光标按下中断信号,YM有效，YP无效，XM有效，XP无效，XP上拉电阻，普通ADC转换，等待中断模式  rADCDLY**=**50000**;** //正常转换模式转换延时大约为(1/3.6864M)\*50000=13.56ms    rINTSUBMSK **&=~(**0x1**<<**9**);**//TC中断使能  rINTMSK **&=~((**U32**)**0x1**<<**31**);**//ADC总中断使能    pISR\_ADC**=(**U32**)**Adc\_Tc\_Handler**;**//指向中断向量表    **}** |

## 获取键值



并定义变量botten1存储键值，flag1按键标记，flag2运算标记

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------获取键值--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void get\_key**(**void**){**  //数字块  **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //0  **{** Beep**(**2000**,** 100**);**botten1 **=** 0**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //1  **{** Beep**(**2000**,** 100**);**botten1 **=** 1**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //2  **{** Beep**(**2000**,** 100**);**botten1 **=** 2**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //3  **{** Beep**(**2000**,** 100**);**botten1 **=** 3**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //4  **{** Beep**(**2000**,** 100**);**botten1 **=** 4**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //5  **{** Beep**(**2000**,** 100**);**botten1 **=** 5**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //6  **{** Beep**(**2000**,** 100**);**botten1 **=** 6**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //7  **{** Beep**(**2000**,** 100**);**botten1 **=** 7**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //8  **{** Beep**(**2000**,** 100**);**botten1 **=** 8**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //9  **{** Beep**(**2000**,** 100**);**botten1 **=** 9**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else**  //功能运算块  **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //除  **{** Beep**(**2000**,** 100**);**botten1 **=** '/'**;**flag1**=**1**;**flag2**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //乘以  **{** Beep**(**2000**,** 100**);**botten1 **=** '\*'**;**flag1**=**1**;**flag2**=**2**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //减法  **{** Beep**(**2000**,** 100**);**botten1 **=** '-'**;**flag1**=**1**;**flag2**=**3**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //加法  **{** Beep**(**2000**,** 100**);**botten1 **=** '+'**;**flag1**=**1**;**flag2**=**4**;**xdata **=** ydata **=** 0**;** **}**  **else**  **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //等于  **{** Beep**(**2000**,** 100**);**botten1 **=** '='**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else**  **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** // 清空  **{** Beep**(**2000**,** 100**);**botten1 **=** 'c'**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **}** |

## 区分键值来调用数字图片显示

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------处理键值--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void resout**(){**    **if(**flag2**==**4**)** **{**sum1**=**sum1**+**sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}** //根据标志flag2 进行相应的 sum1=sum1 ？ sum2 的运算,运算完sum1 sum2置零  **else** **if(**flag2**==**3**)** **{**sum1**=**sum1**-**sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}**  **else** **if(**flag2**==**2**){**sum1**=**sum1**\***sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}**  **else** **if(**flag2**==**1**){**  **if(**sum2**==**0**)** **{**sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** Pait\_Bmp**(**1**,** 34**,** 236**,** 30**,** error**);**Delay**(**500**);**show\_picturu**(**0**);}** //对除数为0处理。  **else** **{**sum1**=**sum1**/**sum2**;**show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}}** //注意:除法运算结果会舍去小数部分    **}**  void cleaning**(){**  sum1**=**0**,**sum2**=**0**;**flag2**=**0**;**show\_picturu**(**0**);**  **}**  void show\_num**(**void**){**  **while(**flag1**){**  flag1**=**0**;** //标志置零  **switch(**botten1**){**  **case** 0**:**  **case** 1**:**  **case** 2**:**  **case** 3**:**  **case** 4**:**  **case** 5**:**  **case** 6**:**  **case** 7**:**  **case** 8**:**  **case** 9**:{**sum2**=**sum2**\***10**+**botten1**;**  **if(**sum2**>**999999999**){**  sum2**=**0**;**Pait\_Bmp**(**2**,** 34**,** 236**,** 30**,** numerror**);**  Delay**(**500**);**show\_picturu**(**sum2**);**  **}**  **else** show\_picturu**(**sum2**);**  **}** **break;** //按键值存于 sum2    **case** '+'**:**  **case** '-'**:**  **case** '\*'**:**  **case** '/'**:** sum1**=**sum2**,**sum2**=**0**;** **break;** //sum2值转存于sum1，sum2置零    **case** '='**:** resout**();** **break;**    //之前把下面的这段放在这里，观察变量发现sum1和和sum2到这里就置零了(按下=后永远等于0)，无奈之下试试把判断放进一个函数 resout(); 里看看。结果成功了！！！  /\* if(flag2==4) {show\_picturu(sum1+sum2); sum2=0;flag2=0;} //根据标志flag2 进行相应的 sum1=sum1 ？ sum2 的运算  else if(flag2==3) {show\_picturu(sum1-sum2); sum2=0;flag2=0; }  else if(flag2==2){show\_picturu(sum1\*sum2); sum2=0;flag2=0; }  else if(flag2==1){  if(sum2==0) {sum1=0;sum2=0;flag2=0; Pait\_Bmp(1, 34, 236, 30, error);Delay(500);show\_picturu(0);} //对除数为0处理。  else {show\_picturu(sum1/sum2); sum2=0;flag2=0; }} //注意:除法运算结果会舍去小数部分  \*/    **case** 'c'**:** cleaning**();** **break;**  **}**    **}**  **}** |

**调用图片**

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------调用图片函数--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void show\_picturu**(**int sum**){**  int k**=**0**,**j**=**0**;**  Pait\_Bmp**(**1**,** 34**,** 238**,** 30**,** clean**);**  **if(**sum**>**0**){** //数值大于0时  **for(;**sum**>**0**;){**  k**=**sum**%**10**;** //k用来取余数  sum**=**sum**/**10**;** //num舍去最后一位  **++**j**;** //j用来标志k的余数在原数值中的实际位数  numTOpic**(**k**,**j**);**  **}**  **}**  **else** **if(**sum**==**0**){** //数值等于0时  numTOpic**(**0**,**1**);**  **}**  **else** **if(**sum**<**0**){** //数值小于0时  sum**=-**sum**;**  **for(;**sum**>**0**;){**  k**=**sum**%**10**;** //k用来取余数  sum**=**sum**/**10**;** //num舍去最后一位  **++**j**;** //j用来标志k的余数在原数值中的实际位数  numTOpic**(**k**,**j**);**  **}**  Pait\_Bmp**(**238**-**13**\*(**j**+**1**),** 34**,** 13**,** 30**,** num\_**);**  **}**  **}**  void numTOpic**(**int m **,**int n**){**  **switch(**m**){**  **case** 0**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num0**);break;**  **case** 1**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num1**);break;**  **case** 2**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num2**);break;**  **case** 3**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num3**);break;**  **case** 4**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num4**);break;**  **case** 5**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num5**);break;**  **case** 6**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num6**);break;**  **case** 7**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num7**);break;**  **case** 8**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num8**);break;**  **case** 9**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num9**);break;**  **}**  **}** |

## 附：

**硬件环境：J-link v8、mini2440、J-link转接板、串口转USB线、T35 TFT LCD屏**

**软件环境：windows7（32位）、开发板uboot（NandFlash）、J-link驱动（J-Link ARM V4.10i）、SecureCRT、ADS1.2**

完整代码：

Main.c文件

|  |
| --- |
| #define GLOBAL\_CLK 1  #include "def.h"  #include "option.h"  #include "2440addr.h"  #include "2440lib.h" //函数声明  #include "2440slib.h"  extern void RTC\_Display\_TICK\_ALM**(**void**);**  extern void RTC\_time**(**void**);**  extern void ADC\_read\_W1**(**void**);**  extern void LCD\_Calculator**(**void**);**    void Main**(**void**)**  **{**  U32 mpll\_val **=** 0**,**consoleNum**;**  Port\_Init**();**    mpll\_val **=** **(**92**<<**12**)|(**1**<<**4**)|(**1**);**    //init FCLK=400M,  ChangeMPllValue**((**mpll\_val**>>**12**)&**0xff**,** **(**mpll\_val**>>**4**)&**0x3f**,** mpll\_val**&**3**);**  ChangeClockDivider**(**14**,** 12**);** //the result of rCLKDIVN [0:1:0:1] 3-0 bit  cal\_cpu\_bus\_clk**();** //HCLK=100M PCLK=50M    consoleNum **=** 0**;** // Uart 1 select for debug.  Uart\_Init**(** 0**,**115200 **);**  Uart\_Select**(** consoleNum **);**    Beep**(**2000**,** 100**);**  LCD\_Calculator**();**  **}** |

option.h文件主要修改（定义屏幕尺寸）

|  |
| --- |
| //--by Customer--  //#define LCD\_N35  //#define LCD\_L80  #define LCD\_T35 //统宝LCD  //#define LCD\_A70  //#define LCD\_VGA1024768  //--end of by Customer--  其中 LCD\_T35 定义  #elif defined(LCD\_T35)  #define LCD\_WIDTH 240  #define LCD\_HEIGHT 320  #define LCD\_PIXCLOCK 4  #define LCD\_RIGHT\_MARGIN 25  #define LCD\_LEFT\_MARGIN 0  #define LCD\_HSYNC\_LEN 4  #define LCD\_UPPER\_MARGIN 1  #define LCD\_LOWER\_MARGIN 4  #define LCD\_VSYNC\_LEN 1 |

Lcd\_init.c文件（LCD初始化与画图函数）

|  |
| --- |
| #include "option.h"  #include "def.h"  #include "2440addr.h"  #include "2440lib.h"  #define LCD\_XSIZE LCD\_WIDTH  #define LCD\_YSIZE LCD\_HEIGHT  #define SCR\_XSIZE LCD\_WIDTH  #define SCR\_YSIZE LCD\_HEIGHT  volatile static unsigned short LCD\_BUFFER**[**SCR\_YSIZE**][**SCR\_XSIZE**];** //定义320行，240列的数组，用于存放显示数据  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD \*在屏幕上画图\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Pait\_Bmp**(**int x0**,**int y0**,**int h**,**int l**,**const unsigned char **\***bmp**)**  **{**  int x**,**y**;**  U32 c**;**  int p **=** 0**;**    **for(** y **=** 0 **;** y **<** l **;** y**++** **)**  **{**  **for(** x **=** 0 **;** x **<** h **;** x**++** **)**  **{**  c **=** bmp**[**p**+**1**]** **|** **(**bmp**[**p**]<<**8**)** **;**  **if** **(** **(** **(**x0**+**x**)** **<** SCR\_XSIZE**)** **&&** **(** **(**y0**+**y**)** **<** SCR\_YSIZE**)** **)**  LCD\_BUFFER**[**y0**+**y**][**x0**+**x**]** **=** c **;**  p **=** p **+** 2 **;**  **}**  **}**  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD \*端口初始化\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Lcd\_Port\_Init**(** void **)**  **{**  rGPCUP**=**0xffffffff**;** // Disable Pull-up register  rGPCCON**=**0xaaaa02a8**;** //Initialize VD[7:0],VM,VFRAME,VLINE,VCLK  rGPDUP**=**0xffffffff**;** // Disable Pull-up register  rGPDCON**=**0xaaaaaaaa**;** //Initialize VD[15:8]  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD功能模块初始化//此函数没开启屏蔽功能  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void LCD\_Init**(**void**)**  **{**  #define M5D(n) ((n)&0x1fffff)  #define LCD\_ADDR ((U32)LCD\_BUFFER)  //con1[6:5]显示模式选择为 TFT LCD面板 模式 [4:1]选择为 TFT的16bpp 像素模式  rLCDCON1 **=** **(**LCD\_PIXCLOCK **<<** 8**)** **|** **(**3 **<<** 5**)** **|** **(**12 **<<** 1**);**  //con2[23:14]设置高度320 VSPW[5:0]决定相应脉冲的宽度; VFPD[13:6]、VBPD[31:24]决定了延时时间; 选择option.h文件中T35定义参数  rLCDCON2 **=** **(**LCD\_UPPER\_MARGIN **<<** 24**)** **|** **((**LCD\_HEIGHT **-** 1**)** **<<** 14**)** **|** **(**LCD\_LOWER\_MARGIN **<<** 6**)** **|** **(**LCD\_VSYNC\_LEN **<<** 0**);**  //con3[18:8]设置高度240 HFPD、HBPD决定了延时时间; 选择option.h文件中T35定义参数  rLCDCON3 **=** **(**LCD\_RIGHT\_MARGIN **<<** 19**)** **|** **((**LCD\_WIDTH **-** 1**)** **<<** 8**)** **|** **(**LCD\_LEFT\_MARGIN **<<** 0**);**  //con4 HSPW[8:0]决定相应脉冲的宽度; 选择option.h文件中T35定义参数  rLCDCON4 **=** **(**13 **<<** 8**)** **|** **(**LCD\_HSYNC\_LEN **<<** 0**);**  #if !defined(LCD\_CON5)  # define LCD\_CON5 ((1<<11) | (1 << 9) | (1 << 8) | (1 << 3) | (1 << 0))  #endif  rLCDCON5 **=** LCD\_CON5**;**  rLCDSADDR1 **=** **((**LCD\_ADDR **>>** 22**)** **<<** 21**)** **|** **((**M5D**(**LCD\_ADDR **>>** 1**))** **<<** 0**);**  rLCDSADDR2 **=** M5D**((**LCD\_ADDR **+** LCD\_WIDTH **\*** LCD\_HEIGHT **\*** 2**)** **>>** 1**);**  rLCDSADDR3 **=** LCD\_WIDTH**;**  rLCDINTMSK **|=** 3**;** //中断屏蔽  rTCONSEL **&=** **(~**7**);**    rTPAL **=** 0x0**;**  rTCONSEL **&=** **~((**1**<<**4**)** **|** 1**);**    **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  LCD视频和控制信号输出或者停止，1开启视频输出  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Lcd\_EnvidOnOff**(**int onoff**)**  **{**  **if(**onoff**==**1**)**  rLCDCON1**|=**1**;** // ENVID=ON  **else**  rLCDCON1 **=**rLCDCON1 **&** 0x3fffe**;** // ENVID Off  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD 电源控制引脚使能  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void Lcd\_PowerEnable**(**int invpwren**,**int pwren**)**  **{**  //GPG4 is setted as LCD\_PWREN  rGPGUP **=** rGPGUP**|(**1**<<**4**);** // Pull-up disable  rGPGCON **=** rGPGCON**|(**3**<<**8**);** //GPG4=LCD\_PWREN    //Enable LCD POWER ENABLE Function  rLCDCON5 **=** rLCDCON5**&(~(**1**<<**3**))|(**pwren**<<**3**);** // PWREN  rLCDCON5 **=** rLCDCON5**&(~(**1**<<**5**))|(**invpwren**<<**5**);** // INVPWREN  **}** |

LCD\_Calculator.c文件（按键处理与调用图片显示）

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  实现功能 LCD计算器  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  #include "2440lib.h"  #include "2440slib.h"  #include "LCD\_init.h"  #include "2440addr.h"  #include "mmu.h"  extern void show\_picturu**(**int num**);**  void numTOpic**(**int m **,**int n**);**  extern unsigned char my1\_240x320**[];**  extern unsigned char Calculator**[];**  extern unsigned char clean**[];**  extern unsigned char error**[];**  extern unsigned char numerror**[];**  extern unsigned char num\_**[];** //负号  extern unsigned char num0**[];**  extern unsigned char num1**[];**  extern unsigned char num2**[];**  extern unsigned char num3**[];**  extern unsigned char num4**[];**  extern unsigned char num5**[];**  extern unsigned char num6**[];**  extern unsigned char num7**[];**  extern unsigned char num8**[];**  extern unsigned char num9**[];**  volatile int xdata**,** ydata**;** //volatile的作用： 作为指令关键字，确保本条指令不会因编译器的优化而省略，且要求每次直接读值.  volatile int flag1**=**0 **,** flag2**=**0 **;**  volatile static long long int sum1**=**0 **,** sum2**=**0**;**  volatile int botten1**;**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD \*触摸屏中断函数\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/    void \_\_irq Adc\_Tc\_Handler**(**void**)**  **{**    rADCTSC**|=(**1**<<**3**)|(**1**<<**2**);** //XP上拉电阻无效, 自动连续测量X坐标和Y坐标.  rADCCON**|=(**1**<<**0**);**//ADC转换开始    **while(**rADCCON**&(**1**<<**0**));**//检测ADC转换是否开始且ADCCON[0]自动清0  **while(!(**rADCCON**&(**0x1**<<**15**)));** //检测ADCCON[15]是否为1,ADC转换是否结束,(必须)  **while(!(**rINTPND**&((**U32**)**0x1**<<**31**)));**//检测ADC中断是否已请求    xdata**=**rADCDAT0**&**0x3ff**;**//读x坐标 >>xdata并不是像素点，而是模拟信号 0-1000  ydata**=**rADCDAT1**&**0x3ff**;**//读y坐标    Uart\_Printf**(**"\n Xdata=%04d, Ydata=%04d\n"**,** xdata**,** ydata**);**    rSUBSRCPND**|=(**0x1**<<**9**);**  rSRCPND**|=((**U32**)**0x1**<<**31**);**  rINTPND**|=((**U32**)**0x1**<<**31**);**    rADCTSC **=**0xd3**;** //ADC等待中断模式  rADCTSC**|=(**0x1**<<**8**);** //ADCTSC[8]=1,设置抬起中断信号    **while(!(**rSUBSRCPND**&(**0x1**<<**9**)));** //检测触屏抬起中断是否已请求    rADCTSC **&=~(**0x1**<<**8**);**//ADCTSC[8]=0光标按下中断信号  rSUBSRCPND**|=(**0x1**<<**9**);**  rSRCPND**|=((**U32**)**0x1**<<**31**);**  rINTPND**|=((**U32**)**0x1**<<**31**);**  **}**    void Touch\_Init**(**void**)**  **{**  rADCCON**=((**1**<<**14**)|(**9**<<**6**));** //A/D分频时钟有效，其值为9  rADCTSC**=**0xd3**;** //光标按下中断信号,YM有效，YP无效，XM有效，XP无效，XP上拉电阻，普通ADC转换，等待中断模式  rADCDLY**=**50000**;** //正常转换模式转换延时大约为(1/3.6864M)\*50000=13.56ms    rINTSUBMSK **&=~(**0x1**<<**9**);**//TC中断使能  rINTMSK **&=~((**U32**)**0x1**<<**31**);**//ADC总中断使能    pISR\_ADC**=(**U32**)**Adc\_Tc\_Handler**;**//指向中断向量表    **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------获取键值--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void get\_key**(**void**){**  //数字块  **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //0  **{** Beep**(**2000**,** 100**);**botten1 **=** 0**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //1  **{** Beep**(**2000**,** 100**);**botten1 **=** 1**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //2  **{** Beep**(**2000**,** 100**);**botten1 **=** 2**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //3  **{** Beep**(**2000**,** 100**);**botten1 **=** 3**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //4  **{** Beep**(**2000**,** 100**);**botten1 **=** 4**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //5  **{** Beep**(**2000**,** 100**);**botten1 **=** 5**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //6  **{** Beep**(**2000**,** 100**);**botten1 **=** 6**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 70 **&&** xdata **<=** 285 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //7  **{** Beep**(**2000**,** 100**);**botten1 **=** 7**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //8  **{** Beep**(**2000**,** 100**);**botten1 **=** 8**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //9  **{** Beep**(**2000**,** 100**);**botten1 **=** 9**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else**  //功能运算块  **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //除  **{** Beep**(**2000**,** 100**);**botten1 **=** '/'**;**flag1**=**1**;**flag2**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 263 **&&** ydata **<** 425 **)** //乘以  **{** Beep**(**2000**,** 100**);**botten1 **=** '\*'**;**flag1**=**1**;**flag2**=**2**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 425 **&&** ydata **<** 587 **)** //减法  **{** Beep**(**2000**,** 100**);**botten1 **=** '-'**;**flag1**=**1**;**flag2**=**3**;**xdata **=** ydata **=** 0**;** **}**  **else** **if(** xdata **>=** 715 **&&** xdata **<=** 930 **&&** ydata **>=** 587 **&&** ydata **<** 750 **)** //加法  **{** Beep**(**2000**,** 100**);**botten1 **=** '+'**;**flag1**=**1**;**flag2**=**4**;**xdata **=** ydata **=** 0**;** **}**  **else**  **if(** xdata **>=** 500 **&&** xdata **<=** 715 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** //等于  **{** Beep**(**2000**,** 100**);**botten1 **=** '='**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **else**  **if(** xdata **>=** 285 **&&** xdata **<=** 500 **&&** ydata **>=** 100 **&&** ydata **<** 262 **)** // 清空  **{** Beep**(**2000**,** 100**);**botten1 **=** 'c'**;**flag1**=**1**;**xdata **=** ydata **=** 0**;** **}**  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------处理键值--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void resout**(){**    **if(**flag2**==**4**)** **{**sum1**=**sum1**+**sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}** //根据标志flag2 进行相应的 sum1=sum1 ？ sum2 的运算,运算完sum1 sum2置零  **else** **if(**flag2**==**3**)** **{**sum1**=**sum1**-**sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}**  **else** **if(**flag2**==**2**){**sum1**=**sum1**\***sum2**;** show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}**  **else** **if(**flag2**==**1**){**  **if(**sum2**==**0**)** **{**sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** Pait\_Bmp**(**1**,** 34**,** 236**,** 30**,** error**);**Delay**(**500**);**show\_picturu**(**0**);}** //对除数为0处理。  **else** **{**sum1**=**sum1**/**sum2**;**show\_picturu**(**sum1**);** sum1**=**0**;**sum2**=**0**;**flag2**=**0**;** **}}** //注意:除法运算结果会舍去小数部分    **}**  void cleaning**(){**  sum1**=**0**,**sum2**=**0**;**flag2**=**0**;**show\_picturu**(**0**);**  **}**  void show\_num**(**void**){**  **while(**flag1**){**  flag1**=**0**;** //标志置零  **switch(**botten1**){**  **case** 0**:**  **case** 1**:**  **case** 2**:**  **case** 3**:**  **case** 4**:**  **case** 5**:**  **case** 6**:**  **case** 7**:**  **case** 8**:**  **case** 9**:{**sum2**=**sum2**\***10**+**botten1**;**  **if(**sum2**>**999999999**){**  sum2**=**0**;**Pait\_Bmp**(**2**,** 34**,** 236**,** 30**,** numerror**);**  Delay**(**500**);**show\_picturu**(**sum2**);**  **}**  **else** show\_picturu**(**sum2**);**  **}** **break;** //按键值存于 sum2    **case** '+'**:**  **case** '-'**:**  **case** '\*'**:**  **case** '/'**:** sum1**=**sum2**,**sum2**=**0**;** **break;** //sum2值转存于sum1，sum2置零    **case** '='**:** resout**();** **break;**    //之前把下面的这段放在这里，观察变量发现sum1和和sum2到这里就置零了(按下=后永远等于0)，无奈之下试试把判断放进一个函数 resout(); 里看看。结果成功了！！！  /\* if(flag2==4) {show\_picturu(sum1+sum2); sum2=0;flag2=0;} //根据标志flag2 进行相应的 sum1=sum1 ？ sum2 的运算  else if(flag2==3) {show\_picturu(sum1-sum2); sum2=0;flag2=0; }  else if(flag2==2){show\_picturu(sum1\*sum2); sum2=0;flag2=0; }  else if(flag2==1){  if(sum2==0) {sum1=0;sum2=0;flag2=0; Pait\_Bmp(1, 34, 236, 30, error);Delay(500);show\_picturu(0);} //对除数为0处理。  else {show\_picturu(sum1/sum2); sum2=0;flag2=0; }} //注意:除法运算结果会舍去小数部分  \*/    **case** 'c'**:** cleaning**();** **break;**  **}**    **}**  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  ------------调用图片函数--------------  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  void show\_picturu**(**int sum**){**  int k**=**0**,**j**=**0**;**  Pait\_Bmp**(**1**,** 34**,** 238**,** 30**,** clean**);**  **if(**sum**>**0**){** //数值大于0时  **for(;**sum**>**0**;){**  k**=**sum**%**10**;** //k用来取余数  sum**=**sum**/**10**;** //num舍去最后一位  **++**j**;** //j用来标志k的余数在原数值中的实际位数  numTOpic**(**k**,**j**);**  **}**  **}**  **else** **if(**sum**==**0**){** //数值等于0时  numTOpic**(**0**,**1**);**  **}**  **else** **if(**sum**<**0**){** //数值小于0时  sum**=-**sum**;**  **for(;**sum**>**0**;){**  k**=**sum**%**10**;** //k用来取余数  sum**=**sum**/**10**;** //num舍去最后一位  **++**j**;** //j用来标志k的余数在原数值中的实际位数  numTOpic**(**k**,**j**);**  **}**  Pait\_Bmp**(**238**-**13**\*(**j**+**1**),** 34**,** 13**,** 30**,** num\_**);**  **}**  **}**  void numTOpic**(**int m **,**int n**){**  **switch(**m**){**  **case** 0**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num0**);break;**  **case** 1**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num1**);break;**  **case** 2**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num2**);break;**  **case** 3**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num3**);break;**  **case** 4**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num4**);break;**  **case** 5**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num5**);break;**  **case** 6**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num6**);break;**  **case** 7**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num7**);break;**  **case** 8**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num8**);break;**  **case** 9**:**Pait\_Bmp**(**238**-**13**\***n**,** 34**,** 13**,** 30**,** num9**);break;**  **}**  **}**  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  TFT LCD \*子main函数\*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/    void LCD\_Calculator**(**void**)**  **{**    Lcd\_Port\_Init**();** //端口初始化  LCD\_Init**();** //TFT LCD功能模块初始化  Lcd\_PowerEnable**(**0**,** 1**);** // TFT LCD 电源控制引脚使能  Lcd\_EnvidOnOff**(**1**);** //LCD视频和控制信号输出或者停止，1开启视频输出        /\*红(255:0:0);绿(0:255:0);蓝(0:0:255);黑(0:0:0);白(255,255,255)\*/    /\*在屏幕上显示三基色\*/    Lcd\_ClearScr**((**0x1f**<<**11**)** **|** **(**0x00**<<**5**)** **|** **(**0x00**));** //red  Delay**(**500**);**  Lcd\_ClearScr**((**0x00**<<**11**)** **|** **(**0x3f**<<**5**)** **|** **(**0x00**));** //green  Delay**(**500**);**  Lcd\_ClearScr**((**0x00**<<**11**)** **|** **(**0x00**<<**5**)** **|** **(**0x1f**));** //blue  Delay**(**500**);**  Lcd\_ClearScr**(** **(**0x1f**<<**11**)** **|** **(**0x3f**<<**5**)** **|** **(**0x1f**)** **)** **;** //clear screen white  Delay**(**500**);**    /\*显示一副图片在屏幕上\*/  Pait\_Bmp**(**0**,** 0**,** 240**,** 320**,** my1\_240x320**);**  Delay**(**1000**);**  Pait\_Bmp**(**0**,** 0**,** 240**,** 320**,** Calculator**); //计算器图形界面**  Delay**(**1000**);**  Pait\_Bmp**(**225**,** 34**,** 13**,** 30**,** num0**);**    //MMU\_EnableICache();这一点至关重要， 不要关闭mmu，否则中断不能正常使用  MMU\_Init**();**  Touch\_Init**();** //开触摸屏中断    **while(**1**){**  get\_key**(); //获取键值**  show\_num**(); //处理键值并调用图片**  **}**    **}** |