

NH070-ARM 彩色液晶显示控制模块使用说明

一、NH070-ARM 型显示控制模块的特点。

- A: 安装方便, 板尺寸为 168mm×116mm, 安装孔 Φ 3 mm, 四个安装孔位置为 162mm×110mm, 高度为 14mm。
- B: 微处理器可以随时读写显示存储器, 而不影响显示效果, 即显示不会出现“雪花”。
- C: 与微处理器接口连接简单, 接口的读、写操作兼容 8 0 3 1 总线时序。
- D: 有两页显示缓存, 可以任意设定显示页和操作页。

二、物理特性:

NO.	Item	Specification	Remark
1	Display Resolution(dot)	480 (W) *234 (H)	
2	Active area(mm)	155.7 (W)*88.2 (H)	
3	Screen size(inch)	7.0 (Diagonal)	
4	Dot pitch(mm)	0.320 (W)*0.370 (H)	
5	Color configuration	R. G. B. stripe	
6	Overall dimension(mm)	168*116*14	
7	Weight(g)		

三、电器参数:

Parameter	Symbol	Min	Max	Unit	Comment
Power Supply for Voltage	VDD	4.8	5.2	V	
Power Supply for Current	Icc	110	150	mA	NOTE1
Input Voltage (H) for logic	Vih	2.4	VDD+0.3	V	
Input Voltage (L) for logic	Vil	VSS-0.3V	0.8	V	
Input Current for logic	Li	0.02	4	mA	
Output Voltage (H) for	Voh	3.0	VDD+0.3	V	

logic					
Output Voltage (L) for logic	Vol	Vss-0.3	0.4	V	
Power Supply for Lamp	Vlamp	11	13	V	
	Ilamp	320	380	mA	

NOTE1:最小电流的条件是电源 4.8V，不对模块操作，液晶屏幕显示全黑。
最大电流的条件是电源 5.2V，对控制卡进行高速逻辑写操作，屏幕显示全白。

使用环境参数：

Item	OPERATION		STORAGE		COMMENT
	MIN	MAX	MIN	MAX	
Ambient Temperature	-20℃	70℃	-25℃	80℃	
Relative Humidity	0%	85%	0%	95%	

四、MCU 接口定义

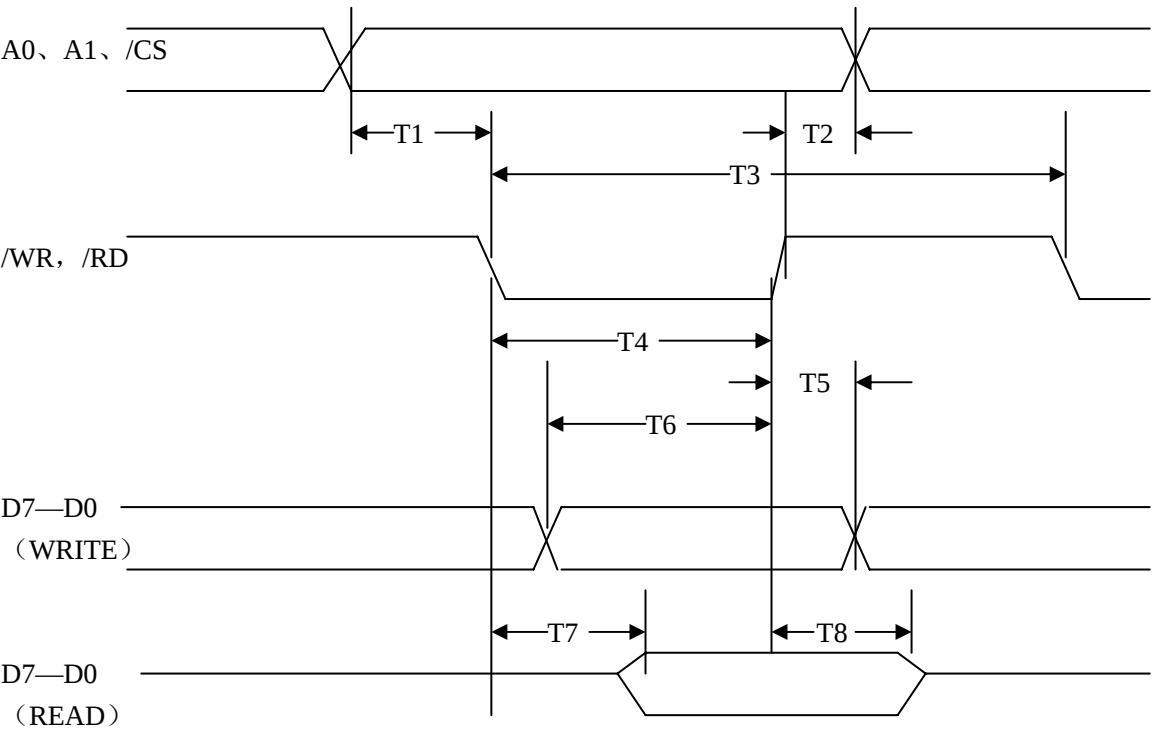
用户板和 NH070-ARM 控制模块的连接（J1）采用 21 线 2.54mm 间距单排插针。对液晶的操作采用标准 intel 总线时序（8031 总线时序）。管脚图如下：（顶视图）

序号	符号	状态	功能
J1-1	A	--	背光电源+12V
J1-2	K	--	背光电源地
J1-3	GND	--	逻辑电源地
J1-4	VDD	--	逻辑电源+5V
J1-5	WR\	输入	写操作信号，低电位有效
J1-6	RD\	输入	读操作信号，低电位有效
J1-7	CS\	输入	使能信号，低电位有效
J1-8	A0	输入	地址信号 0
J1-9	A1	输入	地址信号 1
J1-10	D0	三态	数据总线
J1-11	D1	三态	数据总线
J1-12	D2	三态	数据总线
J1-13	D3	三态	数据总线
J1-14	D4	三态	数据总线
J1-15	D5	三态	数据总线
J1-16	D6	三态	数据总线
J1-17	D7	三态	数据总线
J1-18	VDD	--	逻辑电源+5V
J1-19	GND	--	逻辑电源地
J1-20	GND	--	逻辑电源地
J1-21	NC	--	

五、NH070-ARM 液晶控制板与微处理器的接口时序

接口时序与 8031 的总线时序相同

时序:



时序特性参数

项目	符号	参数说明	Vdd=4.8 to 5.2V		单位	测试条件
			最小	最大		
A0、A1、 /CS	T2	地址保持时间	10	-	ns	CL=10 0pF
	T1	地址建立时间	0	-	ns	
/WR,/R D	T3	读写周期	350	-	ns	
	T4	读写脉冲速度	120	-	ns	
D0—D7	T5	写数据保持时间	120	-	ns	
	T6	写数据建立时间	80	-	ns	
	T7	读数据建立时间	-	50	ns	
	T8	读数据保持时间	10	50	ns	

六、指令介绍及编程方法:

CS	WR	RD	A1	A0	D0-D7
H	任意	任意	任意	任意	操作无效
L	L	H	0	0	写显示数据到 LCD
L	H	L	0	0	从 LCD 读数据
L	L	H	0	1	写显示数据前用：设定行地址，地址 0-233 对应液晶屏从上到下 1 到 234 行
L	L	H	1	0	设定列地址（0-479）
L	L	H	1	1	设定显示页和操作页

1. 写指令和数据

- a) 设定显示页和操作页地址。（A1=A0=1，CS=0，WR=0）

开机时先执行此操作再执行其他操作，工作中不需要换页时

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1/0	0	0	0	1/0

D0 位设定显示页。

D4 位设定操作页。

- b) 写行地址。（A1=0,A0=1）
c) 写列地址高字节。（A1=1,A0=0）
d) 写列地址低字节。（A1=1,A0=0）
e) 写 1 个字节数据。对应一个像素点。（A1=0,A0=0）

D7	D6	D5	D4	D3	D2	D1	D0
R2	R1	R0	G2	G1	G0	B1	B0

R2、R1、R0 表示红色灰度，从 000 到 111，灰度有 8 级 R2 为高灰度位，R0 为低灰度位。

G2、G1、G0 表示绿色灰度，从 000 到 111，灰度有 8 级 G2 为高灰度位，G0 为低灰度位。

B1、B0 表示蓝色灰度，从 00 到 11，灰度有 4 级 B1 为高灰度位，B0 为低灰度位。

如在同一行内连续读、写数据不用每次重写行坐标和列坐标，每一次“读、写数据”操作后列地址自动+1。

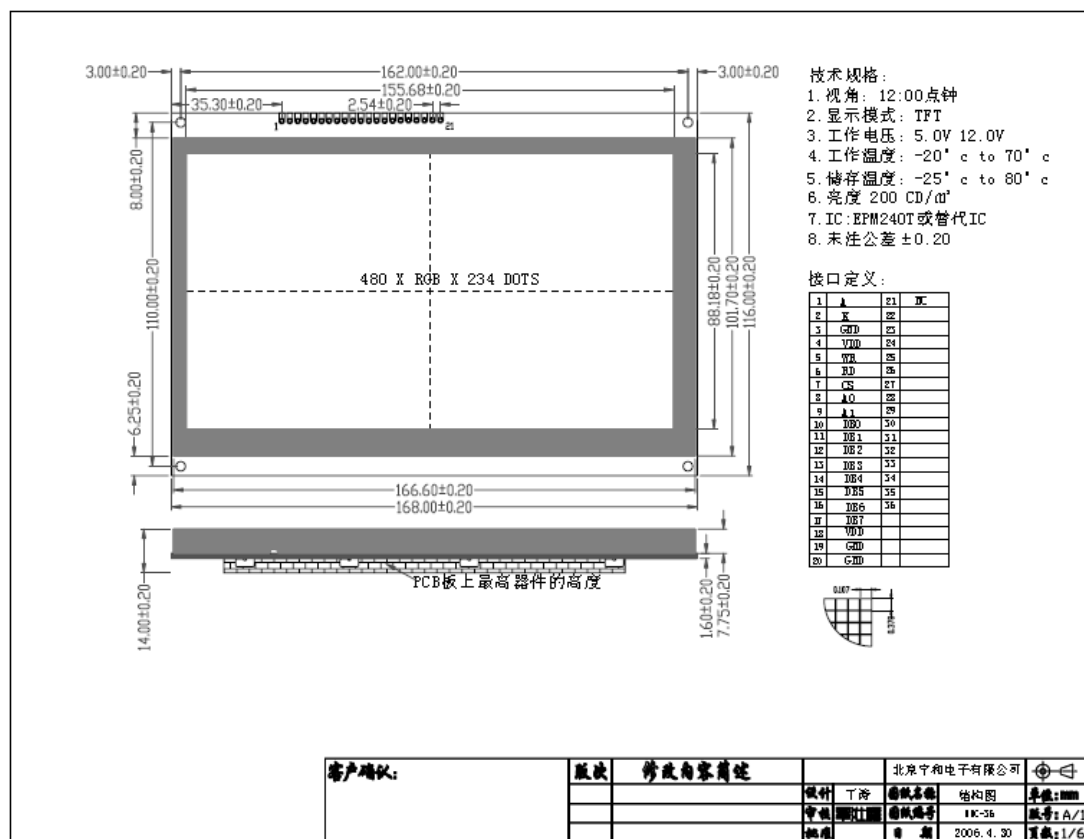
2、操作说明：

- a) 在开机后，延时一段时间（如 100 毫秒），让液晶模块充分上电。
b) 软件先写显示页和操作页指令，工作中如果不需要换页就不用重复使用该指令。
c) 设定行列地址时，先写行地址再写两字节列地址。
d) 写显示内容，每写一字节为一像素点，写后列地址自动加 1，如果同一行从左到右写数据可以连续写不用重新写行列地址。

七、注意事项：

1. 控制板在安装时禁止受力挤压，更不要变形弯曲，以免造成工作不正常或控制板损坏。
2. 控制板的安装和与其连接线的安装要尽量远离强电、高压（交流 220V、逆变器）、大功率接触器、继电器、变压器等对外有干扰的器件。控制板电源要做好抗干扰滤波，这样会避免控制模块受到干扰，造成显示不正常的现象。
3. 使用时禁止用手直接触摸模块电路，以避免静电对板上器件造成损害。
4. 如果第18管脚“RD\”不用时，接上拉电阻

八、模块尺寸



Nh070-ARM 液晶模块例程:

ASM51 格式:

;立即数

```
black      equ      0          ;颜色字节 0,1 位是蓝色 4 级灰度;2,3,4 位是绿色 8 级灰度
blue       equ      03h
green      equ      1ch        ;5,6,7 位是红色 8 级灰度
blueness   equ      1fh        ;三位表示灰度由低到高分别为 000, 001, 010,011,100,101,110,111
red        equ      0e0h
pink       equ      0e3h
yellow     equ      0fch
white      equ      0ffh
```

;以上表示 8 种颜色的最高灰度 89C51 的 P0 口为 8 位数据口

;P2.0, P2.1 为 A0, A1; P2.7 为 CS\; P3.6 为 WR\; P3.7 为 RD\

```
addr_wda   equ      0000h      ;写数据到地址
addr_wcol   equ      0200h      ;写列号地址
addr_wrow   equ      0100h      ;写行号地址
```

addr_wop equ 0300h ;写操作码地址,选择显示页和操作页

;标志位

flag_cir equ 01h ;画圆时用位标志区第二个字节

radw_0 equ 08h

radw_1 equ 09h

radw_2 equ 0ah

radw_3 equ 0bh

radw_4 equ 0ch

radw_5 equ 0dh

radw_6 equ 0eh

radw_7 equ 0fh

;寄存器

bit_byte equ 22h ;操作象素点时的标志位

xstal equ 24h ;x 起始坐标低字节

xstah equ 25h ;x 起始坐标高字节

ystal equ 26h ;y 起始坐标低字节

ystah equ 27h ;y 起始坐标高字节

xendl equ 28h ;x 终止坐标低字节

xendh equ 29h ;x 终止坐标高字节

yendl equ 2ah ;y 终止坐标低字节

yendh equ 2bh ;y 终止坐标高字节

d_xl equ 2ch ;x 差值低字节

d_xh equ 2dh ;x 差值高字节

d_yl equ 2eh ;y 差值低字节

d_yh equ 2fh ;y 差值高字节

;errl equ 30h ;误差值低字节

;errh equ 31h ;误差值高字节

cirxl equ 30h

cirxh equ 31h

ciryl equ 32h

ciryh equ 33h

rad equ 34h

radw equ 21h

radf equ 20h

arcx1 equ 35h

arcy1 equ 36h

arcx2 equ 37h

arcy2 equ 38h

errl equ 39h

errh equ 3ah

```

color          equ      3bh    ;暂存颜色值
char_byte equ 3ch ;写汉字时暂存要写入的单字节字模
char_num equ 3dh ;一个汉字要写入 char_byte32 次(16*16)
char_num1 equ 3eh
bak_char_num equ      3fh
num_byte equ 46h ;一次操作 LCD 的数据 12 字节计数器

```

1. 设定显示页为第 0 页，操作页为第一页。

```

Mov      dptr,#addr_wop
Mov      a,#10h
Movx     @dptr,a

```

2. 用某颜色清全屏

```

clr_lcd:      mov      r6,#0          ;入口: color=颜色, r6=行数
clr_lcd1:     mov      dptr,#addr_wrow ;用到 r6,r7,dptr,psw,a
              mov      a,r6
              movx     @dptr,a
              mov      dptr,#addr_wcol
              mov      a,#0h          ;设为 0 列
              movx     @dptr,a
              movx     @dptr,a
              mov      r7,#240
              mov      a,color
              mov      dptr,#addr_wda
clr_lcd2:     movx     @dptr,a
              movx     @dptr,a        ;一个循环写两个点, 共 240 个循环
              djnz     r7,clr_lcd2    ;共写了 480 个点
              inc      r6
              mov      a,r6
              cjne     a,#234,clr_lcd1
              ret

```

例如: 用红色清全屏

```

Mov      color,#red
Call     clr_lcd

```

3. 在坐标 xsta,ysta 位置画一点。

```

dotw:      mov      dptr,#addr_wrow ;入口 ysta=y 坐标地址,xsta=x 坐标地址, color=颜色
              mov a,ystal           ;资源: dptr,a,psw
              movx     @dptr,a
              mov      dptr,#addr_wcol
              mov a,xstah
              movx     @dptr,a
              mov a,xstal

```

```

movx    @dptr,a
mov     dptr,#addr_wda
mov     a,color
movx    @dptr,a
ret

```

5. 从起始坐标到中止坐标，以某颜色画线。

```

line:      clr      c
           mov      a,#233
           subb     a,ystal
           mov      ystal,a
           clr      c
           mov      a,#233
           subb     a,yendl
           mov      yendl,a

line01:    mov     errl,#0      ;入口：起始坐标双字节(xsta,ysta)，终止坐标双字节(xend,yend)
           mov     errh,#0      ;color=颜色
           mov     a,yendl      ;资源：xstah,xstal 起始 X 坐标
           clr     c            ;      ystah,ystal 起始 Y 坐标
           subb    a,ystal      ;      xendh,xendl 中止 X 坐标
           mov     d_yl,a       ;      yendh,yendl 中止 Y 坐标
           mov     a,yendl      ;      d_xh,d_xl   x 差值
           subb    a,ystah      ;      d_yh,d_yl   y 差值
           mov     d_yh,a       ;      errh,errl   误差值
           jnc     line1        ;      a,psw

line0:     mov     a,yendl
           xch     a,ystal
           mov     yendl,a
           mov     a,yendh
           xch     a,ystah
           mov     yendh,a
           mov     a,xendl
           xch     a,xstal
           mov     xendl,a
           mov     a,xendh
           xch     a,xstah
           mov     xendh,a
           sjmpl   line01

line1:     call    dotw        ;画一点
           mov     a,xendl
           clr     c
           subb    a,xstal
           mov     d_xl,a
           mov     a,xendh
           subb    a,xstah

```

```

        mov d_xh,a
        jc  line3
        mov a,d_yh
        cjne a,d_xh,line2
        mov a,d_yl
        cjne a,d_xl,line2
        sjmprate1
line2:   jnc  line21
        sjmprate1
line21:  jmp  rate2
line3:   mov a,xstal
        clr  c
        subb a,xendl
        mov d_xl,a
        mov a,xstah
        subb a,xendh
        mov d_xh,a
        cjne a,d_yh,line4
        mov a,d_xl
        cjne a,d_yl,line4
line4:   jc  line5
        jmp  rate3
line5:   jmp  rate4
;*****k=0-1 即 d_y<=d_x 时
rate1:   mov a,ystah
        cjne a,yendh,rate10
        mov a,ystal
        cjne a,yendl,rate10
        sjmprate11
rate10:  mov a,errh
        jb  acc.7,rate12
        mov a,ystal
        add a,#1
        mov ystal,a
        mov a,ystah
        addc a,#0
        mov ystah,a
rate11:  mov a,xstal
        add a,#1
        mov xstal,a
        mov a,xstah
        addc a,#0
        mov xstah,a
        mov a,errl

```

```

        add a,d_yl
        mov errl,a
        mov a,errh
        addc a,d_yh
        mov errh,a
        clr c
        mov a,errl
        subb a,d_xl
        mov errl,a
        mov a,errh
        subb a,d_xh
        mov errh,a
        sjmp rate13
rate12:  mov a,xstal
        add a,#1
        mov xstal,a
        mov a,xstah
        addc a,#0
        mov xstah,a
        mov a,errl
        add a,d_yl
        mov errl,a
        mov a,errh
        addc a,d_yh
        mov errh,a
rate13:  call dotw
        mov a,xstal
        cjne a,xendl,rate10
        mov a,xstah
        cjne a,xendh,rate10
        ret

```

,*****k>1 即 d_y>d_x 时

```

rate2:  mov a,errh
        jb acc.7,rate21
        mov a,ystal
        add a,#1
        mov ystal,a
        mov a,ystah
        addc a,#0
        mov ystah,a
        clr c
        mov a,errl
        subb a,d_xl
        mov errl,a

```

```

        mov a,errh
        subb a,d_xh
        mov errh,a
        sjmp rate22
rate21:  mov a,ystal
        add a,#1
        mov ystal,a
        mov a,ystah
        addc a,#0
        mov ystah,a
        mov a,xstal
        add a,#1
        mov xstal,a
        mov a,xstah
        addc a,#0
        mov xstah,a
        mov a,errl
        add a,d_yl
        mov errl,a
        mov a,errh
        addc a,d_yh
        mov errh,a
        clr c
        mov a,errl
        subb a,d_xl
        mov errl,a
        mov a,errh
        subb a,d_xh
        mov errh,a
rate22:  call dotw
        mov a,ystal
        cjne a,yendl,rate2
        mov a,ystah
        cjne a,yendh,rate2
        ret
,*****k=-1--0 即 d_x>=d_y 时
rate3:   mov a,ystal
        cjne a,yendl,rate30
        mov a,ystah
        cjne a,yendh,rate30
        sjmp rate31
rate30:  mov a,errh
        jb acc.7,rate32
        mov a,ystal

```

```

    add a,#1
    mov ystal,a
    mov a,ystah
    addc a,#0
    mov ystah,a
rate31:  mov a,xstal
        subb a,#1
        mov xstal,a
        mov a,xstah
        subb a,#0
        mov xstah,a
        mov a,errl
        add a,d_yl
        mov errl,a
        mov a,errh
        addc a,d_yh
        mov errh,a
        clr c
        mov a,errl
        subb a,d_xl
        mov errl,a
        mov a,errh
        subb a,d_xh
        mov errh,a
        sjmprate33
rate32:  mov a,xstal
        clr c
        subb a,#1
        mov xstal,a
        mov a,xstah
        subb a,#0
        mov xstah,a
        mov a,errl
        add a,d_yl
        mov errl,a
        mov a,errh
        addc a,d_yh
        mov errh,a
rate33:  call dotw
        mov a,xstal
        cjne a,xendl,rate30
        mov a,xstah
        cjne a,xendh,rate30
        ret

```

*****k<-1 即 d_x>d_y

```
rate4:    mov a,errh
          jb  acc.7,rate41
          mov a,ystal
          add a,#1
          mov ystal,a
          mov a,ystah
          addc a,#0
          mov ystah,a
          clr  c
          mov a,errl
          subb a,d_xl
          mov errl,a
          mov a,errh
          subb a,d_xh
          mov errh,a
          sjmprate42
```

```
rate41:   clr  c
          mov a,xstal
          subb a,#1
          mov xstal,a
          mov a,xstah
          subb a,#0
          mov xstah,a
          mov a,ystal
          add a,#1
          mov ystal,a
          mov a,ystah
          addc a,#0
          mov ystah,a
          mov a,errl
          add a,d_yl
          mov errl,a
          mov a,errh
          addc a,d_yh
          mov errh,a
          clr  c
          mov a,errl
          subb a,d_xl
          mov errl,a
          mov a,errh
          subb a,d_xh
          mov errh,a
```

```
rate42:   call dotw
```

```

mov a,ystal
cjne a,yendl,rate4
mov a,ystah
cjne a,yendh,rate4
ret

```

例如：从坐标（16，16）到坐标（200，150）画一跟黄线。

```

Mov    xstah,#0
mov    xstal,#10h
mov    ystah,#0
mov    ystal,#10h
mov    xendh,#0
mov    xendl,#200
mov    yendh,#0
mov    yendl,#150
mov    color,#yellow
call   line

```

5. 以某坐标（cirx,ciry）为圆心，以 rad 为半径，以 color 为颜色画圆。

```

circle:  mov    a,#234      ;入口: cirxh,cirxl=圆心 X 坐标
          subb   a,ciryl    ;      ciryh,ciryl=圆心 Y 坐标
          mov    ciryl,a    ;      rad=半径, color=颜色
          clr    a          ;使用资源: cirxh,cirxl,ciryh,ciryl,a,psw,flag_cir
          clr    flag_cir   ;arcx1,arcy1,arcx2,arcy2,errl,errh,color,dptr
          mov    arcx1,a    ;xstah,xstal,ystahystal
          mov    arcy2,a
          mov    a,rad
          mov    arcy1,a
          mov    arcx2,a
cir1:     call   amp
          call   squ
          mov    a,arcx1
          cjne   a,arcx2,cir3
cir2:     call   amp
          ret
cir3:     xch    a,arcx2
          mov    arcx1,a
          mov    a,arcy1
          xch    a,arcy2
          mov    arcy1,a
          cpl    flag_cir
          call   amp
          mov    a,arcx1
          xch    a,arcy1

```

```

        mov arcx1,a
        call squ
        mov a,arcx1
        xch a,arcy1
        mov arcx1,a
        cpl flag_cir
        cjne a,arcx2,cir4
        sjmp cir2
cir4:    xch a,arcx2
        mov arcx1,a
        mov a,arcy1
        xch a,arcy2
        mov arcy1,a
        sjmp cir1
;*****方 差 计 算 程 序
;*****@=(a+1)^2+b^2-r^2-b    a=arcx1 b=arcy1
squ:    mov a,arcx1
        inc a
        mov b,a
        mul ab
        mov errl,a
        mov errh,b
        mov a,arcy1
        mov b,a
        mul ab
        add a,errl
        mov errl,a
        mov a,b
        addc a,errh
        mov errh,a
        mov a,rad
        mov b,a
        mul ab
        clr c
        xch a,errl
        subb a,errl
        mov errl,a
        mov a,b
        xch a,errh
        subb a,errh
        mov errh,a
        jc squ1
        mov a,errl
        subb a,arcy1

```

```

        mov errl,a
        mov a,errh
        subb a,#0
        mov errh,a
        jc  squ1
        dec arcy1
squ1:    inc arcx1
        ret
;*****圆弧扩展子程序
;*****输入: 坐标值 x=arcx1,y=arcy1,圆心坐标值(cirx,ciry)
;*****x 为 x1 和 x2 之一, y 为 y1 和 y2 之一
amp:     mov a,cirxl
        add a,arcx1
        mov xstal,a
        mov a,cirxh
        addc a,#0
        mov xstah,a
        mov a,ciryl
        add a,arcy1
        mov ystal,a
        mov a,ciryh
        addc a,#0
        mov ystah,a
        jb  flag_cir,amp1
        jb  radw_6,amp2
        sjmp amp3
amp1:    jb  radw_7,amp2
        sjmp amp3
amp2:    call dotw
amp3:    mov a,cirxl
        clr  c
        subb a,arcx1
        mov xstal,a
        mov a,cirxh
        subb a,#0
        mov xstah,a
        jb  flag_cir,amp4
        jb  radw_5,amp5
        sjmp amp6
amp4:    jb  radw_4,amp5
        sjmp amp6
amp5:    call dotw
amp6:    mov a,ciryl
        clr  c

```

```

        subb a,arcy1
        mov ystal,a
        mov a,ciryh
        subb a,#0
        mov ystah,a
        jb  flag_cir,amp7
        jb  radw_2,amp8
        sjmp amp9
amp7:   jb  radw_3,amp8
        sjmp amp9
amp8:   call dotw
amp9:   mov a,cirxl
        add a,arcx1
        mov xstal,a
        mov a,cirxh
        addc a,#0
        mov xstah,a
        jb  flag_cir,ampa
        jb  radw_1,ampb
        sjmp ampc
ampa:   jb  radw_0,ampb
        sjmp ampc
ampb:   call dotw
ampc:   ret

```

例如：在坐标（160，117）为圆心，以 110 位半径画一白色圆。

```

        mov color,#white
        mov cirxl,#160
        mov cirxh,#0
        mov ciryl,#117
        mov ciryh,#0
        mov rad,#110
        mov radw,#0ffh
        call circle

```

5. 显示一行汉字子程序

```

disp_hanzi:   mov     a,bak_char_num    ;显示汉字子程序
              mov     char_num,a       ;xsta 和 ysta 是显示汉字的左上角起始位置(以像素点为单
              ;位)
disp_hanzi1:  clr     a                 ;dptr 是指向字模表首地址的指针,字模表写在程序中
              movc    a,@a+dptr        ;char_num1 显示汉字需要的行数,bak_char_num 显示汉字
              ;一行需要的像素点除以 8
              inc     dptr              ;color 是颜色
              mov     char_byte,a
              mov     a,dph             ;暂存 dptr

```

```

mov    ciryh,a
mov    a,dpl
mov    ciryh,a
mov    r7,#8
disp_hanzi2:  mov    a,char_byte
               rlc    a
               mov    char_byte,a
               jnc    disp_hanzi3
               push   07h
               call   dotw
               pop    07h
disp_hanzi3:  mov    a,xstal
               add    a,#1
               mov    xstal,a
               mov    a,xstah
               addc   a,#0
               mov    xstah,a
               djnz   r7,disp_hanzi2
               ;暂存 dptr
               mov    a,ciryh
               mov    dph,a
               mov    a,ciryl
               mov    dpl,a
               djnz   char_num,disp_hanzi1
               mov    a,bak_char_num
               mov    b,#8
               mul    ab
               mov    r0,a
               mov    a,xstal
               clr    c
               subb   a,r0
               mov    xstal,a
               mov    a,xstah
               subb   a,b
               mov    xstah,a
               inc    ystal
               djnz   char_num1,disp_hanzi
               ret

```

例如：在坐标(80,177)位置用绿色显示一行汉字

```

mov xstah,#0
mov xstal,#80
mov ystal,#177
mov color,#green
mov dptr,#bairi6

```

```

mov char_num1,#16
mov bak_char_num,#20
; xsta 和 ysta 是显示汉字的左上角起始位置; color 是颜色
call disp_hanzi
;-- 白日依山尽黄河入海流 -- ** 楷体_GB2312, 12 ** 下划线小四
; 当前所选字体下一个汉字对应的点阵为: 宽度 x 高度=160x16。
bairi6:
DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H
DB 000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,000H,004H
DB 000H,000H,000H,000H,000H,008H,000H,004H,000H,030H,000H,000H,001H,00CH,000H,030H
DB 000H,0FEH,000H,044H,001H,080H,000H,000H,000H,018H,000H,004H,000H,020H,000H,03EH
DB 001H,000H,000H,030H,000H,088H,000H,04EH,000H,081H,0C0H,000H,003H,01EH,003H,003H
DB 000H,040H,001H,0C4H,002H,003H,000H,020H,001H,010H,001H,0F8H,000H,0FEH,000H,080H
DB 001H,060H,001H,03CH,001H,09EH,001H,004H,004H,0FCH,000H,020H,001H,0E0H,000H,091H
DB 088H,002H,000H,040H,000H,084H,000H,010H,007H,0E2H,001H,008H,004H,030H,000H,020H
DB 002H,040H,001H,0FEH,004H,012H,000H,040H,008H,07AH,008H,024H,004H,006H,003H,0E8H
DB 01CH,066H,000H,041H,002H,020H,01EH,040H,000H,0F4H,000H,0A0H,000H,054H,008H,078H
DB 004H,024H,002H,008H,024H,0A8H,008H,043H,004H,010H,003H,0FCH,002H,0A4H,001H,020H
DB 004H,09FH,082H,088H,00FH,0C4H,002H,010H,009H,090H,008H,042H,009H,01CH,002H,0A8H
DB 004H,0C4H,002H,020H,00BH,064H,005H,028H,008H,008H,004H,010H,00EH,088H,008H,082H
DB 011H,08FH,085H,0D0H,008H,004H,004H,010H,009H,028H,009H,050H,008H,008H,007H,0D0H
DB 009H,00CH,008H,0FCH,020H,000H,004H,0F0H,008H,008H,008H,018H,012H,008H,009H,051H
DB 00FH,0F0H,000H,020H,011H,047H,01FH,004H,000H,000H,007H,000H,030H,008H,030H,01CH
DB 027H,0F4H,032H,051H,000H,010H,000H,020H,011H,080H,000H,008H,003H,000H,002H,030H
DB 000H,038H,0C0H,00FH,020H,010H,024H,00FH,000H,020H,000H,000H,002H,000H,000H,000H
DB 001H,000H,00CH,010H,000H,010H,000H,000H,000H,060H,008H,000H,0FFH,0FFH,0FFH,0FFH
DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH

```

C51 格式:

```

uchar *p,*p1,*p2; /*建立数组指针*/
uchar data l1;
uchar data l2;
uchar data page; /*行地址*/
uchar data col1; /*列地址高字节*/
uchar data col=0x00; /*列地址低字节*/
uchar data yanse1; /*背景色*/
uchar data yanse2; /*前景色—显示字符颜色*/
#define dlcd XBYTE[0x0000] /*定义送数据的地址*/
#define clcd1 XBYTE[0x0100] /*定义送指令的地址*/
#define clcd2 XBYTE[0x0200] /*定义送指令的地址*/
#define clcd3 XBYTE[0x0300] /*定义送指令的地址*/
1. 用某颜色清屏
void Clearlcd() /*入口: yanse=清屏颜色*/
{

```

```

uchar data i,j;
uchar data k=0x00;
clcd3=0x00;    /*设定显示页和操作页*/
for(i=0;i<234;i++)
{
    clcd1=k;    /*设定行地址*/
    clcd2=0x00;
    clcd2=0x00;    /*设定列地址*/
    for(j=0;j<160;j++) ;写入一行颜色
    {
        dlcd=yanse1; /*写显示数据*/
        dlcd=yanse1;
        dlcd=yanse1;
        dlcd=yanse1;
    }
    k=k+0x01;
}
}

```

2. 显示一行汉字子程序,汉字字模表在程序中

```

void Hanz()    /*入口: *p=汉字字模 16 位首地址*/
{
    /*page=汉字显示位置左上角坐标行地址*/
    uchar data i,j,col2;    /*col,col1=汉字显示位置左上角坐标列地址*/
    for(i=0;i<11;i++)    /*yanse1=汉字显示背景色,yanse=汉字显示颜色*/
    {
        /*l1=字模表中显示一行字节数*/
        clcd3=0x00;    /*l2=显示汉字行数*/
        clcd1=page;
        clcd2=col;
        clcd2=col1;
        page=page+0x01;
        for(j=0;j<12;j++)
        {
            Hanz1();
        }
        col2=col1;
        col2=col2+0x01;
        if((col2|0x00)!=0x00)
        {
        }
        else
        {
            col=0x01;
        }
    }
}

```

```
}
```

```
void Hanz1()
```

```
{
```

```
    uchar data i,wbyte,wbyte1;
```

```
    wbyte=*p;
```

```
    for(i=0;i<8;i++)
```

```
    {
```

```
        wbyte1=wbyte;
```

```
        if((wbyte1&0x80)!=0x80)
```

```
        {
```

```
            wbyte=wbyte<<1;
```

```
            dlcd=yanse1;
```

```
        }
```

```
        else
```

```
        {
```

```
            wbyte=wbyte<<1;
```

```
            dlcd=yanse2;
```

```
        }
```

```
    }
```

```
    p=p+1;
```

```
}
```

例如： 在第 16 行 16 列显示一行兰底黄字

```
l1=33;
```

```
l2=25
```

```
page=0x10;
```

```
col1=0x00;
```

```
col=0x10;
```

```
yanse1=0x03;    /*蓝色*/
```

```
yanse2=0xfc;    /*黄色*/
```

```
p=hanzi;
```

```
hanz();
```

```
char code hanzi[25][33]=
```

```
{
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x03,0xE0,0x00,0x00,0x00,0xE0,0x00,0x00,0x00,
```

```
    0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xE0,0x00,0x00,
```

```
    0x01,0xF0,0x00,0x00,0x00,0xF0,0x00,0x00,0x00,0x60,0x00,0x00,0x00,0xF0,0x00,0x00,
```

```
    0x00,0x00,0xE0,0x00,0x00,0x00,0x70,0x00,0x00,0x00,0xF0,0x00,0x00,0x00,0x78,0x00,
```



```
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00  
};
```