



LED 显示 开发指南

REV1.0

通讯地址：深圳市南山区蛇口南海大道 1079 号花园城数码大厦 A 座 9 楼

邮政编码：518067

公司电话：+(86 755)86169257

传 真：+(86 755)86169057

公司网站：www.chipsea.com

微 信 号：芯海科技

微信二维码：



版本历史

版本	记录	时间
V1.0	起草	2016/8/5

目 录

版本历史	2
目 录	3
1 LED 驱动程序原理	4
1.1 定时器自动扫描	4
2 应用层的映射关系	8
2.1 IO 口接线和 LED 编号	8
2.2 编程时逻辑对应关系	9
2.2.1 简化编程的考虑	9
3 总结	16

1 LED 驱动程序原理

1.1 定时器自动扫描

LED 初始化代码:

```
Sub_LED_init:
    movlw    240
    movwf    TM1IN
    bsf      INTE,TM1IE
    movlw    10110000b
    movwf    TM1CON

    bsf      inte,gie

    movlw    00111000b
    movwf    LEDCON1
    movlw    00000000b
    movwf    LEDCON2
    movlw    00000000b
    ;movlw   00010111b
    movwf    CHPCON

    movlf    26H,7DH
    movlf    ffH,PT2EN

    return
```

这里关闭 LED 自动扫描功能，用定时器模拟 IO 口翻转，主程序系统指令周期时钟设置为 1000K,所以定时器扫描频率为 $1000K/241/8=500HZ$ 左右

定时中断代码:

采用了定时中断 1 来自动扫描，每 8 次一个循环

```
inter_timer1:
    bcf      intf,tm1if

    incf     R_LED_COUNT,f      ;计数器计数
    movlw    8
    subwf    R_LED_COUNT,w
    btfsc    status,c
    clrf     R_LED_COUNT

    movlw    ffh
    movwf    pt2
    movwf    PT2CON
    movwf    7FH
    movwf    R_LED_LIGHT

    clrf     R_LED_DUTY
```

```

movfw    R_LED_COUNT
addpcw
goto     INTERRUPT_LED_P20
goto     INTERRUPT_LED_P21
goto     INTERRUPT_LED_P22
goto     INTERRUPT_LED_P23
goto     INTERRUPT_LED_P24
goto     INTERRUPT_LED_P25
goto     INTERRUPT_LED_P26
goto     INTERRUPT_LED_P27

```

每次扫描控制一个 IO 口为低，另外 IO 口的状态根据寄存器 LED1-LED7 的内容控制如下面 PT2.0 输出低的情况：

```

□ INTERRUPT_LED_P20:
    ;PT2.0
    btfsc    LED1,0
    incf     R_LED_DUTY,f
    btfsc    LED2,1
    incf     R_LED_DUTY,f
    btfsc    LED1,6
    incf     R_LED_DUTY,f
    btfsc    LED2,2
    incf     R_LED_DUTY,f
    btfsc    LED4,6
    incf     R_LED_DUTY,f
    btfsc    LED2,3
    incf     R_LED_DUTY,f
    btfsc    LED7,0
    incf     R_LED_DUTY,F

    btfsc    LED1,0
    bcf      R_LED_LIGHT,1
    btfsc    LED2,1
    bcf      R_LED_LIGHT,2
    btfsc    LED1,6
    bcf      R_LED_LIGHT,3
    btfsc    LED2,2
    bcf      R_LED_LIGHT,4
    btfsc    LED4,6
    bcf      R_LED_LIGHT,5
    btfsc    LED2,3
    bcf      R_LED_LIGHT,6
    btfsc    LED7,0
    bcf      R_LED_LIGHT,7

```

这里 R_LED_DUTY 用于记录点灯的个数，R_LED_LIGHT 用于记录点灯的 IO 口位

根据 R_LED_LIGHT 控制 IO 口的状态，并跳转到 INTERRUPT_LED_END

```

movlw    11111111b
xorwf    R_LED_LIGHT,w
btfsc    status,z
goto     INTERRUPT_LED_END

bcf      R_LED_LIGHT,0
movfw    R_LED_LIGHT
movwf    PT2CON
movwf    7FH

movlw    11111110b
movwf    PT2

goto     INTERRUPT_LED_END

```

在 INTERRUPT_LED_END 根据 R_LED_DUTY 灯的个数调整点灯的电流，让亮度均匀

```

□ INTERRUPT_LED_END:
    bcf      status,c
    rlf      R_LED_DUTY,f
    ;根据点灯个数做电流调整
    ;111 110 10ma
    ;101    15ma
    ;100    20ma
    ;011    25ma
    ;010    30ma
    ;001    40ma
    ;000    50ma
    movfw    R_LED_DUTY
    addpcw
    nop
    nop
    movlw    10111000b
    goto     INTERRUPT_LED_END1
    movlw    01111000b
    goto     INTERRUPT_LED_END1
    movlw    00111000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
    movlw    00011000b
    goto     INTERRUPT_LED_END1
□ INTERRUPT_LED_END1:
    movwf    LEDCON1

```

为了使亮度更均匀还要根据点灯的个数，调整灯亮的周期

```

;根据点灯个数做时间调整
;1、不做电流调整时候的配置 80 100 120 160 200 240
movfw    R_LED_DUTY
addpcw
nop
nop
movlw    115
goto     INTERRUPT_LED_END2
movlw    139
goto     INTERRUPT_LED_END2
movlw    124
goto     INTERRUPT_LED_END2
movlw    130
goto     INTERRUPT_LED_END2
movlw    170
goto     INTERRUPT_LED_END2
movlw    210
goto     INTERRUPT_LED_END2
movlw    255
goto     INTERRUPT_LED_END2
INTERRUPT_LED_END2:
movwf    TM1IN

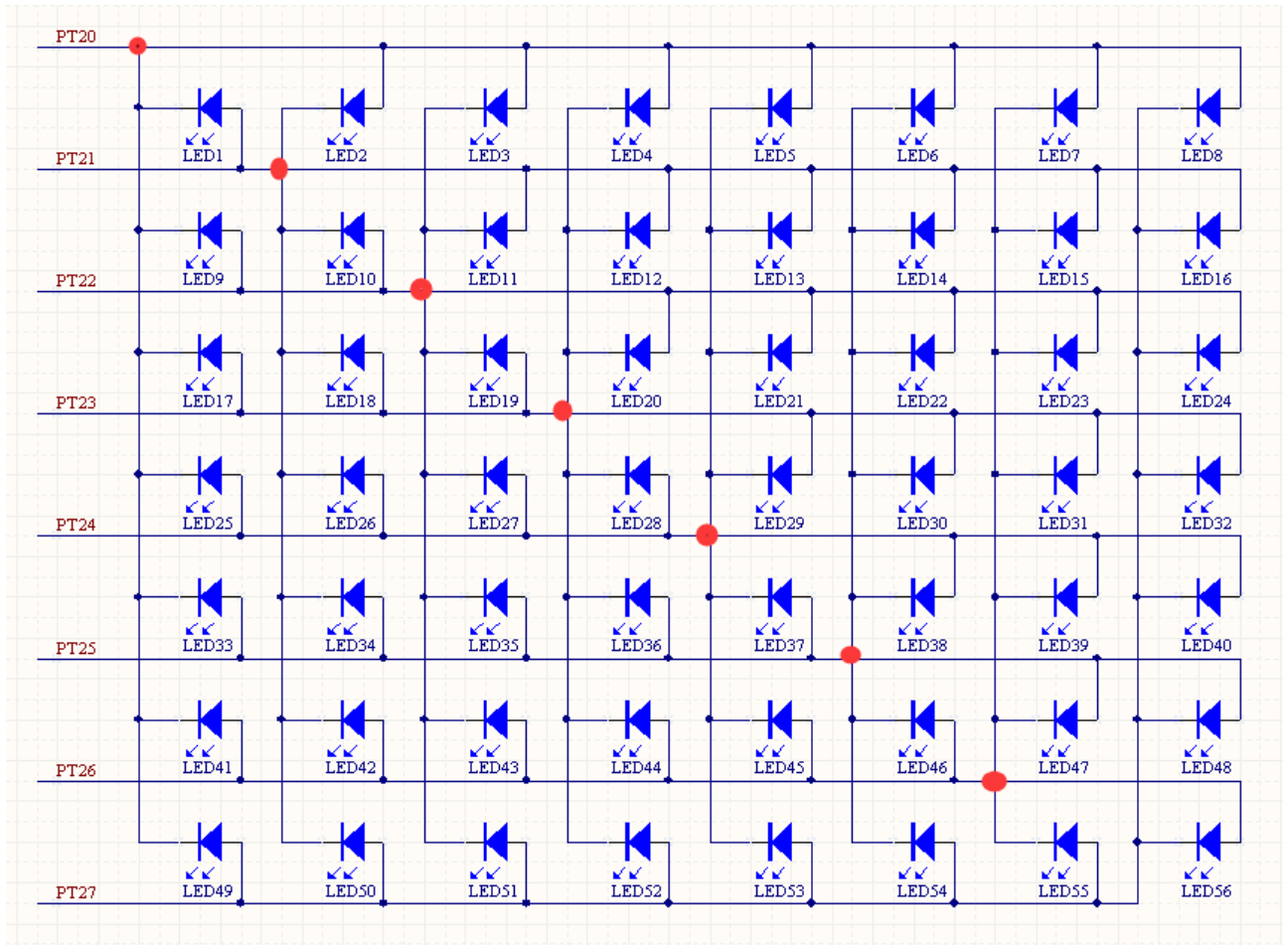
goto     interrupt_end
    
```

到这里，添加 LED 初始化的代码，并添加中断代码，以及定义 R_LED_COUNT、R_LED_DUTY、R_LED_LIGHT 几个寄存器的定义。LED 自动扫描功能就可以正常工作了。

正常使用的时候只需要对 LED1-LED7 寄存器赋相应的值，就可以点亮对应 IO 口上的 LED。

2 应用层的映射关系

2.1 IO 口接线和 LED 编号



IO 接法如上，并且定义 LED 的编号如上图所示。例如要点亮 LED34，要 PT25 输出高，PT21 输出低其它 IO 口高阻态。对应用户手册 LED3 的 bit[1]。如下图所示。

LED1 寄存器（地址为 3BH）

~

LED7 寄存器（地址为 41H）

位地址	标识符		功能	
7:0	LED1~LED7		LED 驱动器控制信号	

LED1	LED1[0]	LED1	P2.1 高 P2.0 低	LED4	LED4[0]	LED43	P2.6 高 P2.2 低	LED7	LED7[0]	LED49	P2.7 高 P2.0 低
	LED1[1]	LED18	P2.3 高 P2.1 低		LED4[1]	LED35	P2.5 高 P2.2 低		LED7[1]	LED50	P2.7 高 P2.1 低
	LED1[2]	LED44	P2.6 高 P2.3 低		LED4[2]	LED38	P2.4 高 P2.5 低		LED7[2]	LED51	P2.7 高 P2.2 低
	LED1[3]	LED7	P2.0 高 P2.6 低		LED4[3]	LED45	P2.6 高 P2.4 低		LED7[3]	LED52	P2.7 高 P2.3 低
	LED1[4]	LED31	P2.3 高 P2.6 低		LED4[4]	LED37	P2.5 高 P2.4 低		LED7[4]	LED53	P2.7 高 P2.4 低
	LED1[5]	LED12	P2.1 高 P2.3 低		LED4[5]	LED22	P2.2 高 P2.5 低		LED7[5]	LED54	P2.7 高 P2.5 低
	LED1[6]	LED17	P2.3 高 P2.0 低		LED4[6]	LED33	P2.5 高 P2.0 低		LED7[6]	LED55	P2.7 高 P2.6 低
	LED1[7]	LED30	P2.3 高 P2.5 低		LED4[7]	LED21	P2.2 高 P2.4 低		LED7[7]	LED42	P2.6 高 P2.1 低
LED2	LED2[0]	LED2	P2.0 高 P2.1 低	LED5	LED5[0]	LED10	P2.2 高 P2.1 低				
	LED2[1]	LED9	P2.2 高 P2.0 低		LED5[1]	LED20	P2.2 高 P2.3 低				
	LED2[2]	LED25	P2.4 高 P2.0 低		LED5[2]	LED29	P2.3 高 P2.4 低				
	LED2[3]	LED41	P2.6 高 P2.0 低		LED5[3]	LED36	P2.5 高 P2.3 低				
	LED2[4]	LED5	P2.0 高 P2.4 低		LED5[4]	LED28	P2.4 高 P2.3 低				
	LED2[5]	LED3	P2.0 高 P2.2 低		LED5[5]	LED19	P2.3 高 P2.2 低				
	LED2[6]	LED4	P2.0 高 P2.3 低		LED5[6]	LED13	P2.1 高 P2.4 低				
	LED2[7]	LED11	P2.1 高 P2.2 低		LED5[7]	LED27	P2.4 高 P2.2 低				
LED3	LED3[0]	LED23	P2.2 高 P2.6 低	LED6	LED6[0]	LED8	P2.0 高 P2.7 低				
	LED3[1]	LED34	P2.5 高 P2.1 低		LED6[1]	LED16	P2.1 高 P2.7 低				
	LED3[2]	LED47	P2.5 高 P2.6 低		LED6[2]	LED24	P2.2 高 P2.7 低				
	LED3[3]	LED39	P2.4 高 P2.6 低		LED6[3]	LED32	P2.3 高 P2.7 低				
	LED3[4]	LED46	P2.6 高 P2.5 低		LED6[4]	LED40	P2.4 高 P2.7 低				
	LED3[5]	LED14	P2.1 高 P2.5 低		LED6[5]	LED48	P2.5 高 P2.7 低				
	LED3[6]	LED6	P2.0 高 P2.5 低		LED6[6]	LED56	P2.6 高 P2.7 低				
	LED3[7]	LED26	P2.4 高 P2.1 低		LED6[7]	LED15	P2.1 高 P2.6 低				

2.2 编程时逻辑对应关系

2.2.1 简化编程的考虑

实际编程的时候，假设显示的数据放到 R_LED_D1 ~ R_LED_D7 这 7 个寄存器里面，我们需要把这些数据映射到 LED1~LED7 这 7 个寄存器上面去，即假设 R_LED_D1 存放了第一个数码管的数据，它对应着实际硬件电路的编号为 1、3、5、7、9、11、13、15 这些 LED，我们需要根据这些灯的编号，找到要置起的 LED1~LED7 要置起的位分别为：

LED1	bit0	bit3		
LED2	bit5	bit4	bit1	bit7
LED3				
LED4				
LED5	bit6			
LED6	bit7			
LED7				

即完整对应过去，需要依次对这些位进行映射。这个过程时间比较长，可能会影响到显示的效果，为此，定义 7 个转换缓存寄存器。R_LED_TEMP1~R_LED_TEMP7 用于转换的中间过程，等转换完了再把 R_LED_TEMP1~R_LED_TEMP7 赋值给 LED1~LED7。

这时候可能编写出下面的代码：

```

LED_YINSHE_TIAOZHENG5:
    clr    R_LED_TEMP1
    clr    R_LED_TEMP2
    clr    R_LED_TEMP3
    clr    R_LED_TEMP4
    clr    R_LED_TEMP5
    clr    R_LED_TEMP6
    clr    R_LED_TEMP7

;---LED1映射转换
    btfsc  R_LED_D1,0
    bsf    R_LED_TEMP3,0
    btfsc  R_LED_D1,1
    bsf    R_LED_TEMP1,4
    btfsc  R_LED_D1,2
    bsf    R_LED_TEMP6,7
    btfsc  R_LED_D1,3
    bsf    R_LED_TEMP1,3
    btfsc  R_LED_D1,4
    bsf    R_LED_TEMP6,6
    btfsc  R_LED_D1,5
    bsf    R_LED_TEMP6,0

。。。。。LED2~LED6 映射（略）

;---LED7映射转换
    btfsc  R_LED_D7,0      ;KG
    bsf    R_LED_TEMP3,2
    btfsc  R_LED_D7,1      ;小数点
    bsf    R_LED_TEMP3,3
    btfsc  R_LED_D7,2
    bsf    R_LED_TEMP3,4    ;BMI
    btfsc  R_LED_D7,3
    bsf    R_LED_TEMP6,1    ;.C
    btfsc  R_LED_D7,4
    bsf    R_LED_TEMP6,2    ;偏瘦
    btfsc  R_LED_D7,5
    bsf    R_LED_TEMP6,3    ;正常
    btfsc  R_LED_D7,6
    bsf    R_LED_TEMP6,4    ;偏胖
    btfsc  R_LED_D7,7
    bsf    R_LED_TEMP6,5    ;肥胖
    
```

```

movfw    R_LED_TEMP1
movwf    LED1
movfw    R_LED_TEMP2
movwf    LED2
movfw    R_LED_TEMP3
movwf    LED3
movfw    R_LED_TEMP4
movwf    LED4
movfw    R_LED_TEMP5
movwf    LED5
movfw    R_LED_TEMP6
movwf    LED6
movfw    R_LED_TEMP7
movwf    LED7

return

```

到这里，逻辑关系已经对应好了。但实际编写这个转换函数，你会发现，你要先看数码管的某一段对应的是哪个编号的 LED，再查用户手册如下表，找到对应的是哪个 LED 寄存器的哪一位，再返回来编写代码，这样来来回回 40~50 次（最多 56 次）才能把这个函数写完。这么繁琐的过程导致很容易出错的同时，PCB 改版，或者灯的位置有变动要修改，都要修改这个函数。重复劳动导致效率低下而且容易出错！！

LED1	LED1[0]	LED1	P2.1 高 P2.0 低	LED4	LED4[0]	LED43	P2.6 高 P2.2 低	LED7	LED7[0]	LED49	P2.7 高 P2.0 低
	LED1[1]	LED18	P2.3 高 P2.1 低		LED4[1]	LED35	P2.5 高 P2.2 低		LED7[1]	LED50	P2.7 高 P2.1 低
	LED1[2]	LED44	P2.6 高 P2.3 低		LED4[2]	LED38	P2.4 高 P2.5 低		LED7[2]	LED51	P2.7 高 P2.2 低
	LED1[3]	LED7	P2.0 高 P2.6 低		LED4[3]	LED45	P2.6 高 P2.4 低		LED7[3]	LED52	P2.7 高 P2.3 低
	LED1[4]	LED31	P2.3 高 P2.6 低		LED4[4]	LED37	P2.5 高 P2.4 低		LED7[4]	LED53	P2.7 高 P2.4 低
	LED1[5]	LED12	P2.1 高 P2.3 低		LED4[5]	LED22	P2.2 高 P2.5 低		LED7[5]	LED54	P2.7 高 P2.5 低
	LED1[6]	LED17	P2.3 高 P2.0 低		LED4[6]	LED33	P2.5 高 P2.0 低		LED7[6]	LED55	P2.7 高 P2.6 低
	LED1[7]	LED30	P2.3 高 P2.5 低		LED4[7]	LED21	P2.2 高 P2.4 低		LED7[7]	LED42	P2.6 高 P2.1 低
LED2	LED2[0]	LED2	P2.0 高 P2.1 低	LED5	LED5[0]	LED10	P2.2 高 P2.1 低				
	LED2[1]	LED9	P2.2 高 P2.0 低		LED5[1]	LED20	P2.2 高 P2.3 低				
	LED2[2]	LED25	P2.4 高 P2.0 低		LED5[2]	LED29	P2.3 高 P2.4 低				
	LED2[3]	LED41	P2.6 高 P2.0 低		LED5[3]	LED36	P2.5 高 P2.3 低				
	LED2[4]	LED5	P2.0 高 P2.4 低		LED5[4]	LED28	P2.4 高 P2.3 低				
	LED2[5]	LED3	P2.0 高 P2.2 低		LED5[5]	LED19	P2.3 高 P2.2 低				
	LED2[6]	LED4	P2.0 高 P2.3 低		LED5[6]	LED13	P2.1 高 P2.4 低				
	LED2[7]	LED11	P2.1 高 P2.2 低		LED5[7]	LED27	P2.4 高 P2.2 低				
LED3	LED3[0]	LED23	P2.2 高 P2.6 低	LED6	LED6[0]	LED8	P2.0 高 P2.7 低				
	LED3[1]	LED34	P2.5 高 P2.1 低		LED6[1]	LED16	P2.1 高 P2.7 低				
	LED3[2]	LED47	P2.5 高 P2.6 低		LED6[2]	LED24	P2.2 高 P2.7 低				
	LED3[3]	LED39	P2.4 高 P2.6 低		LED6[3]	LED32	P2.3 高 P2.7 低				
	LED3[4]	LED46	P2.6 高 P2.5 低		LED6[4]	LED40	P2.4 高 P2.7 低				
	LED3[5]	LED14	P2.1 高 P2.5 低		LED6[5]	LED48	P2.5 高 P2.7 低				
	LED3[6]	LED6	P2.0 高 P2.5 低		LED6[6]	LED56	P2.6 高 P2.7 低				
	LED3[7]	LED26	P2.4 高 P2.1 低		LED6[7]	LED15	P2.1 高 P2.6 低				

为此整理了下表的定义：

```

DEFINE L_LED1 "R_LED_TEMP1,0"
DEFINE L_LED2 "R_LED_TEMP2,0"
DEFINE L_LED3 "R_LED_TEMP2,5"
DEFINE L_LED4 "R_LED_TEMP2,6"
DEFINE L_LED5 "R_LED_TEMP2,4"
DEFINE L_LED6 "R_LED_TEMP3,6"
DEFINE L_LED7 "R_LED_TEMP1,3"
DEFINE L_LED8 "R_LED_TEMP6,0"

DEFINE L_LED9 "R_LED_TEMP2,1"
DEFINE L_LED10 "R_LED_TEMP5,0"
DEFINE L_LED11 "R_LED_TEMP2,7"
DEFINE L_LED12 "R_LED_TEMP1,5"
DEFINE L_LED13 "R_LED_TEMP5,6"
DEFINE L_LED14 "R_LED_TEMP3,5"
DEFINE L_LED15 "R_LED_TEMP6,7"
DEFINE L_LED16 "R_LED_TEMP6,1"

DEFINE L_LED17 "R_LED_TEMP1,6"
DEFINE L_LED18 "R_LED_TEMP1,1"
DEFINE L_LED19 "R_LED_TEMP5,5"
DEFINE L_LED20 "R_LED_TEMP5,1"
DEFINE L_LED21 "R_LED_TEMP4,7"
DEFINE L_LED22 "R_LED_TEMP4,5"
DEFINE L_LED23 "R_LED_TEMP3,0"
DEFINE L_LED24 "R_LED_TEMP6,2"

DEFINE L_LED49 "R_LED_TEMP7,0"
DEFINE L_LED50 "R_LED_TEMP7,1"
DEFINE L_LED51 "R_LED_TEMP7,2"
DEFINE L_LED52 "R_LED_TEMP7,3"
DEFINE L_LED53 "R_LED_TEMP7,4"
DEFINE L_LED54 "R_LED_TEMP7,5"
DEFINE L_LED55 "R_LED_TEMP7,6"
DEFINE L_LED56 "R_LED_TEMP6,6"

DEFINE L_LED25 "R_LED_TEMP2,2"
DEFINE L_LED26 "R_LED_TEMP3,7"
DEFINE L_LED27 "R_LED_TEMP5,7"
DEFINE L_LED28 "R_LED_TEMP5,4"
DEFINE L_LED29 "R_LED_TEMP5,2"
DEFINE L_LED30 "R_LED_TEMP1,7"
DEFINE L_LED31 "R_LED_TEMP1,4"
DEFINE L_LED32 "R_LED_TEMP6,3"

DEFINE L_LED33 "R_LED_TEMP4,6"
DEFINE L_LED34 "R_LED_TEMP3,1"
DEFINE L_LED35 "R_LED_TEMP4,1"
DEFINE L_LED36 "R_LED_TEMP5,3"
DEFINE L_LED37 "R_LED_TEMP4,4"
DEFINE L_LED38 "R_LED_TEMP4,2"
DEFINE L_LED39 "R_LED_TEMP3,3"
DEFINE L_LED40 "R_LED_TEMP6,4"

DEFINE L_LED41 "R_LED_TEMP2,3"
DEFINE L_LED42 "R_LED_TEMP7,7"
DEFINE L_LED43 "R_LED_TEMP4,0"
DEFINE L_LED44 "R_LED_TEMP1,2"
DEFINE L_LED45 "R_LED_TEMP4,3"
DEFINE L_LED46 "R_LED_TEMP3,4"
DEFINE L_LED47 "R_LED_TEMP3,2"
DEFINE L_LED48 "R_LED_TEMP6,5"

```

可以看到这张表已经把 LED 的编号和对应的 LED 寄存器位置对应起来。如下图所示：

DEFINE L_LED1 "R_LED_TEMP1,0"	LED1[6]	LED17	P2.3 高 P2.0 低	
DEFINE L_LED2 "R_LED_TEMP2,0"	LED1[7]	LED30	P2.3 高 P2.5 低	
DEFINE L_LED3 "R_LED_TEMP2,5"	LED2[0]	LED2	P2.0 高 P2.1 低	
DEFINE L_LED4 "R_LED_TEMP2,6"	LED2[1]	LED9	P2.2 高 P2.0 低	
DEFINE L_LED5 "R_LED_TEMP2,4"	LED2[2]	LED25	P2.4 高 P2.0 低	
DEFINE L_LED6 "R_LED_TEMP3,6"	LED2[3]	LED41	P2.6 高 P2.0 低	
DEFINE L_LED7 "R_LED_TEMP1,3"	LED2[4]	LED5	P2.0 高 P2.4 低	LED5
DEFINE L_LED8 "R_LED_TEMP6,0"	LED2[5]	LED3	P2.0 高 P2.2 低	
	LED2[6]	LED4	P2.0 高 P2.3 低	
	LED2[7]	LED11	P2.1 高 P2.2 低	
	LED3[0]	LED23	P2.2 高 P2.6 低	

这时候只需要编写，实际数码管每一段跟 LED 的对应关系这张表：

```

;-----
; LED Character table
;   _   A   点   4   : 5-6
; | | | F B   kg   3
;   _   G   斤   2
; | | | E C   BLE 1
;   _   D   BAT 0
;-----
; LED1 LED2 LED3 LED4

DEFINE LED1_1U "L_LED43"
DEFINE LED1_1D "L_LED44"
DEFINE LED1_1U "L_LED41"
DEFINE LED1_DD "L_LED42"
DEFINE LED1_MM "L_LED56"
DEFINE LED1_CM "L_LED49"
DEFINE LED1_G "L_LED8"

DEFINE LED2_A "L_LED1"
DEFINE LED2_B "L_LED2"
DEFINE LED2_C "L_LED3"
DEFINE LED2_D "L_LED4"
DEFINE LED2_E "L_LED5"
DEFINE LED2_F "L_LED6"
DEFINE LED2_G "L_LED7"

DEFINE LED6_A "L_LED33"
DEFINE LED6_B "L_LED34"
DEFINE LED6_C "L_LED35"
DEFINE LED6_D "L_LED36"
DEFINE LED6_E "L_LED37"
DEFINE LED6_F "L_LED38"
DEFINE LED6_G "L_LED39"

DEFINE LED7_KG "L_LED46"
DEFINE LED7_DOT "L_LED45"
DEFINE LED7_BMI "L_LED47"
DEFINE LED7_C "L_LED50"
DEFINE LED7_S "L_LED51"
DEFINE LED7_N "L_LED52"
DEFINE LED7_F "L_LED53"
DEFINE LED7_FAT "L_LED54"

DEFINE LED3_A "L_LED9"
DEFINE LED3_B "L_LED10"
DEFINE LED3_C "L_LED11"
DEFINE LED3_D "L_LED12"
DEFINE LED3_E "L_LED13"
DEFINE LED3_F "L_LED14"
DEFINE LED3_G "L_LED15"

DEFINE LED4_A "L_LED17"
DEFINE LED4_B "L_LED18"
DEFINE LED4_C "L_LED19"
DEFINE LED4_D "L_LED20"
DEFINE LED4_E "L_LED21"
DEFINE LED4_F "L_LED22"
DEFINE LED4_G "L_LED23"

DEFINE LED5_A "L_LED25"
DEFINE LED5_B "L_LED26"
DEFINE LED5_C "L_LED27"
DEFINE LED5_D "L_LED28"
DEFINE LED5_E "L_LED29"
DEFINE LED5_F "L_LED30"
DEFINE LED5_G "L_LED31"

```

以及把映射转换关系函数改写成下面这样，即可转换映射关系。这样子不容易出错，后续 PCB 不修改时，LED 的位置变换，只需要修改上面的表格。

```

LED_YINSHE_TIAOZHENG:
    ;---LED1映射转换
    btfsc    R_LED_D1,0
    bsf      LED1_1U
    btfsc    R_LED_D1,1
    bsf      LED1_1D
    btfsc    R_LED_D1,2
    bsf      LED1_UU
    btfsc    R_LED_D1,3
    bsf      LED1_DD
    btfsc    R_LED_D1,4
    bsf      LED1_MM
    btfsc    R_LED_D1,5
    bsf      LED1_CM
    btfsc    R_LED_D1,6
    bsf      LED1_G
    . . . . . LED2~LED6 映射转换 (略)
    ;---LED7映射转换
    btfsc    R_LED_D7,0
    bsf      LED7_KG      ;KG
    btfsc    R_LED_D7,1
    bsf      LED7_DOT     ;小数点
    btfsc    R_LED_D7,2
    bsf      LED7_BMI     ;BMI
    btfsc    R_LED_D7,3
    bsf      LED7_C       ;.C
    btfsc    R_LED_D7,4
    bsf      LED7_S       ;偏瘦
    btfsc    R_LED_D7,5
    bsf      LED7_N       ;正常
    btfsc    R_LED_D7,6
    bsf      LED7_F       ;偏胖
    btfsc    R_LED_D7,7
    bsf      LED7_FAT     ;肥胖
    return
    
```

可以看到，转换函数里仅仅描述了，R_LED_D1~R_LED_D7 里面的位跟实际 LED 的对应关系。这里 LED7_FAT、LED2_A 等都只是自己定义的实际 LED 别名。最终的对应关系是在上面的表里完成的如下图所示。

```

DEFINE LED7_KG      "L_LED46"
DEFINE LED7_DOT     "L_LED45"
DEFINE LED7_BMI     "L_LED47"
DEFINE LED7_C       "L_LED50"
DEFINE LED7_S       "L_LED51"
DEFINE LED7_N       "L_LED52"
DEFINE LED7_F       "L_LED53"
DEFINE LED7_FAT     "L_LED54"
    
```

即现在完成 LED 映射关系要做的工作是：

1、添加下面对应表

```

DEFINE L_LED1 "R_LED_TEMP1,0"
DEFINE L_LED2 "R_LED_TEMP2,0"
DEFINE L_LED3 "R_LED_TEMP2,5"
DEFINE L_LED4 "R_LED_TEMP2,6"
DEFINE L_LED5 "R_LED_TEMP2,4"
DEFINE L_LED6 "R_LED_TEMP3,6"
DEFINE L_LED7 "R_LED_TEMP1,3"
DEFINE L_LED8 "R_LED_TEMP6,0"

DEFINE L_LED9 "R_LED_TEMP2,1"
DEFINE L_LED10 "R_LED_TEMP5,0"
DEFINE L_LED11 "R_LED_TEMP2,7"
DEFINE L_LED12 "R_LED_TEMP1,5"
DEFINE L_LED13 "R_LED_TEMP5,6"
DEFINE L_LED14 "R_LED_TEMP3,5"
DEFINE L_LED15 "R_LED_TEMP6,7"
DEFINE L_LED16 "R_LED_TEMP6,1"

DEFINE L_LED17 "R_LED_TEMP1,6"
DEFINE L_LED18 "R_LED_TEMP1,1"
DEFINE L_LED19 "R_LED_TEMP5,5"
DEFINE L_LED20 "R_LED_TEMP5,1"
DEFINE L_LED21 "R_LED_TEMP4,7"
DEFINE L_LED22 "R_LED_TEMP4,5"
DEFINE L_LED23 "R_LED_TEMP3,0"
DEFINE L_LED24 "R_LED_TEMP6,2"

DEFINE L_LED25 "R_LED_TEMP2,2"
DEFINE L_LED26 "R_LED_TEMP3,7"
DEFINE L_LED27 "R_LED_TEMP5,7"
DEFINE L_LED28 "R_LED_TEMP5,4"
DEFINE L_LED29 "R_LED_TEMP5,2"
DEFINE L_LED30 "R_LED_TEMP1,7"
DEFINE L_LED31 "R_LED_TEMP1,4"
DEFINE L_LED32 "R_LED_TEMP6,3"

DEFINE L_LED33 "R_LED_TEMP4,6"
DEFINE L_LED34 "R_LED_TEMP3,1"
DEFINE L_LED35 "R_LED_TEMP4,1"
DEFINE L_LED36 "R_LED_TEMP5,3"
DEFINE L_LED37 "R_LED_TEMP4,4"
DEFINE L_LED38 "R_LED_TEMP4,2"
DEFINE L_LED39 "R_LED_TEMP3,3"
DEFINE L_LED40 "R_LED_TEMP6,4"

DEFINE L_LED41 "R_LED_TEMP2,3"
DEFINE L_LED42 "R_LED_TEMP7,7"
DEFINE L_LED43 "R_LED_TEMP4,0"
DEFINE L_LED44 "R_LED_TEMP1,2"
DEFINE L_LED45 "R_LED_TEMP4,3"
DEFINE L_LED46 "R_LED_TEMP3,4"
DEFINE L_LED47 "R_LED_TEMP3,2"
DEFINE L_LED48 "R_LED_TEMP6,5"

DEFINE L_LED49 "R_LED_TEMP7,0"
DEFINE L_LED50 "R_LED_TEMP7,1"
DEFINE L_LED51 "R_LED_TEMP7,2"
DEFINE L_LED52 "R_LED_TEMP7,3"
DEFINE L_LED53 "R_LED_TEMP7,4"
DEFINE L_LED54 "R_LED_TEMP7,5"
DEFINE L_LED55 "R_LED_TEMP7,6"
DEFINE L_LED56 "R_LED_TEMP6,6"

```

2、给每颗 LED 命名，并编写映射关系函数，把寄存器的位跟要点的灯对应起来。

```

LED_YINSHE_TIAOZHENG:
;---LED1映射转换
btfsc R_LED_D1,0
bsf LED1_1U
btfsc R_LED_D1,1
bsf LED1_1D
btfsc R_LED_D1,2
bsf LED1_UU
btfsc R_LED_D1,3
bsf LED1_DD
btfsc R_LED_D1,4
bsf LED1_MM
btfsc R_LED_D1,5
bsf LED1_CM

```

3、把上面命名的 LED 名字跟实际的 LED 编号对应起来。

```

;-----
; LED Character table
;   A      点  4    : 5-6
; | |      F  B      kg 3
;   G      斤  2
; | |      E  C      BLE 1
;   D      BAT 0
;-----
; LED1 LED2 LED3 LED4

DEFINE LED1_1U "L_LED43"
DEFINE LED1_1D "L_LED44"
DEFINE LED1_UU "L_LED41"
DEFINE LED1_DD "L_LED42"
DEFINE LED1_MM "L_LED56"
DEFINE LED1_CM "L_LED49"
DEFINE LED1_G "L_LED8"

```


3 总结

使用 LED 程序的步骤：

- 1、添加 LED 初始化程序
- 2、添加 LED 启用的定时器 1 的代码
- 3、添加 LED 编号和寄存器对应关系的映射表
- 4、添加 LED 映射函数
- 5、添加 LED 段名跟 LED 编号的映射表

参考程序详见附件：