

```

LPO:      MOV     second, #59

LP1:MOV     R5, #20          ;开始计时
LP2:MOV     TH0, #4CH
      MOV     TL0, #0B0H
      JNB     B_STOP,COUNT_SET0;运行中, count清零

LP3:CALL    DISP
      LCALL    KEY_SCAN      ;;键盘扫描
      LCALL    DO_KEY        ;;操作
      MOV     R4,#5

TIMER:
      JNB     TF0, LP3
      CLR     TF0
      DJNZ    R5, LP2
      JB      B_STOP,STOP
      JNB     STATE,COUNT_UP ;正计时

COUNT_DOWN:
      MOV     A,second
      DEC     second
      JNZ     LP1            ;second未到0跳转LP1
      MOV     second,#0
      MOV     A,minute
      JZ      BEEP           ;second到0, minute到0, 跳转BEEP
      MOV     R6,minute
      DEC     minute
      CJNE    R6,#0,LP0
      SJMP    STOP

COUNT_UP:
      INC     second
      MOV     A,second
      CLR     C
      SUBB    A,#60
      JC      LP1
      INC     minute
      SJMP    LP1

STOP:
      SETB    B_STOP
      LCALL    KEY_SCAN      ;;键盘扫描
      LCALL    DO_KEY        ;;根据键码值运行相应操作
      LCALL    DISP
      LJMP    LP1
      RET

BEEP:
      LCALL    KEY_SCAN
      MOV     A,KEYCODE
      CJNE    A,#0FFH,STOP

LOOP:
      CPL     P3.3           ;取反小喇叭 (p3.3)
      lcall    delay         ;调用延时
      lcall    DISP
      ajmp    BEEP           ;反复循环
      delay: mov     R6,#01h ;延时子程序

LLA:

```

按键处理：按键扫描后查表得到 KEYCODE，KEYCODE=0FFH 跳回，KEYCODE=0FH 时 B_STOP 翻转，KEYCODE=1 加 10min，KEYCODE=2 加 1min，KEYCODE=3 加 10s，当 LED_STATE=1 熄屏时，KEYCODE!=0FFH 就将 LED_STATE 置零。

三、调试过程

1. MOV A, #0H

```
MOVC A, @A+DPTR
MOV P0, A
SETB P2.5
LCALL DELAY_LED
CLR P2.5
```

```
MOV A, second
MOV B, #10
DIV AB
MOVC A, @A+DPTR
MOV P0, A
CLR P2.6
LCALL DELAY_LED
SETB P2.6
MOV A, B
MOVC A, @A+DPTR
MOV P0, A
CLR P2.7
LCALL DELAY_LED
SETB P2.7
```

虽然显示后面的 minute 也会加延迟，但是实验证明这种方式显示带不动 3 位以上的 LED，调整 LED 亮灯设置，把 P2 赋成 0FFH，如右图所示先 SETB 后 CLR，能够正常显示。

2. 开启后 LED 不亮，因为暂停的跳转指令放到了 L3 里的 CALL DISP 之后，注意暂停跳转的位置，LED 灯持续扫描，需让计时器持续进工作。将指令放在 R5 判断后解决问题。

```
LP1:MOV R5, #20 ;开始计时
LP2:MOV TH0, #4CH
MOV TL0, #0B0H

LP3:CALL DISP
JNB P1.0, K1
JNB P1.1, K2
JNB P1.2, K3
JNB TF0, LP3
CLR TF0
DJNZ R5, LP2
JB B_STOP, STOP |
DJNZ second, LP1 ;second未到0跳转LP1
DJNZ minute, LP0 ;second到0, minute未到0, 跳转LP4
JMP BEGIN
```

3. 由于每次按键时间稍长就会被判定按键多次，加入 R4 作为单次标志，每次定时器到一定时间 R4 复位。

```
KEY_1:
JNB B_STOP, KEY_RET
CLR C ;C=0
CJNE A, #4, ADD_TIME
CJNE R4, #4, SET_0 ;R4!=4 A=4 清零
LJMP KEY_RET
```

4. 在设置倒计时时原代码如下：

```
DJNZ second, LP1 ;second未到0跳转LP1
MOV A, minute
JZ STOP
DJNZ minute, LP0 ;second到0, minute未到0, 跳转LP0
```

测试时发现时间为 01'01 时下一秒会跳到 00'00，故用在减一前先赋值给 R6，用未减一的值 R6 进行比较跳转；MINUTE 原值为 0，如果用 DJNZ 语句，先自减为 0FFH，跳不到 STOP，故用 A 先判断，MINUTE 不为 0 再减 1，代码如下。

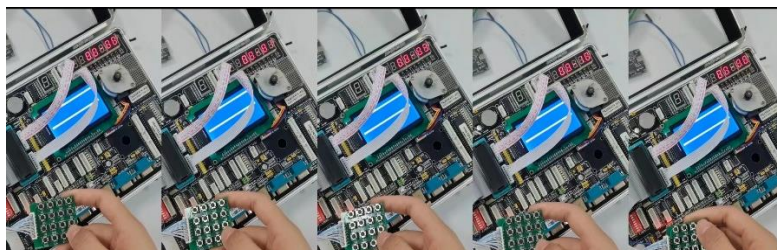
```

MOV    A,minute
JZ     BEEP      ;second到0, minute到0, 跳转BEEP
MOV    R6,minute
DEC    minute
CJNE   R6,#0,LPO
SJMP   STOP

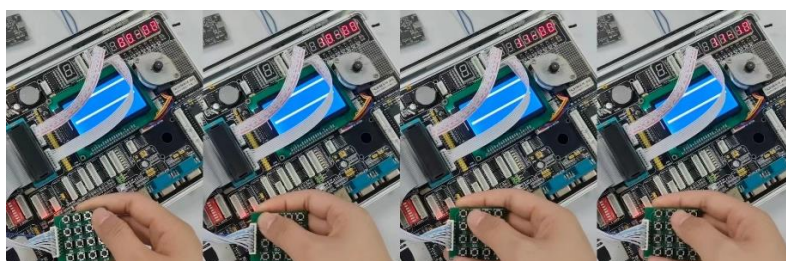
```

四、实验结果

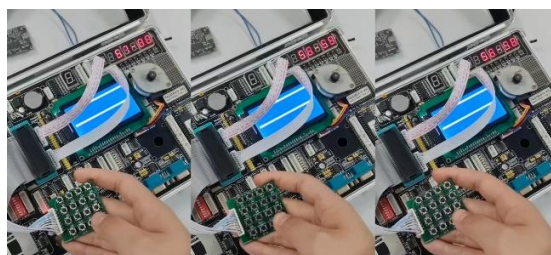
正计时功能：



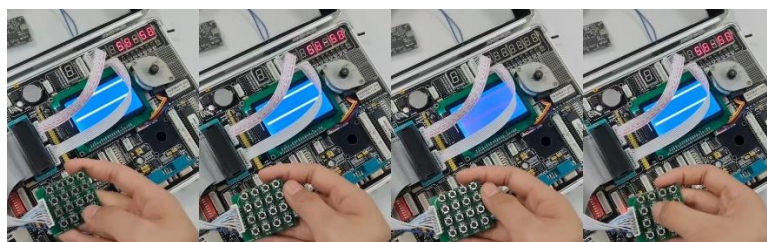
按键功能：



倒计时功能：



熄屏功能：



五、心得

细节处理很重要。注意使计时器始终运行，注意调用与返回时各变量的值，若定时器里用到 CY，在调用跳转指令时尽量不用 C 判断，若定时器里用到 PUSH 和 POP，在其他地方用将容易出错。功能一点一点加，方便调试，若有大段修改复制到记事本里，因为返回上一步有步数限制，进行较多修改后可能会丢失前面的信息。

实验注重方法的设计与优化，关注细节处理，不仅提升了我对单片机定时器的理解，使我熟练运用数码管显示、按键扫描等方法，还锻炼了我的系统设计与调试能力。