```
;=========================================================================
;   2021.09.26
;   Fully annotated & fixed  by @ElioYang
;   Source code comes from https://zhuanlan.zhihu.com/p/266066452
;=========================================================================
ORG 0000H                               ;复位起始地址
    LJMP START
ORG 000BH                               ;中间地址保留给中断向量表
                                        ;定时器的中断部分，更改R7
    LJMP INTERRUPT_T0                   ;定时器0中断程序入口地址
ORG 0040H                               ;程序实际起始地址
START:                                  ;初始化
    P4          EQU 0C0H                ;单片机P4口地址，PPT中给出
    P4SW        EQU 0BBH                ;P4 方式控制字地址
    MOV         P4SW,#70H               ;将P4口设置为普通的IO口，则P4SW=0X70
    CLK         EQU P4.4                ;时钟线
    DAT         EQU P4.5                ;数据线
    SW          EQU P3.6                ;该单片机上电复位后须设置P4SW寄存器
    MOV         DPTR,#DIGIT_TABLE       ;将断码表首地址传给数据指针寄存器
INIT:
    MOV R3,#0                           ;计数 数码管显示 个位
    MOV R4,#0                           ;计数 数码管显示 十位
    MOV R5,#0                           ;计数 数码管显示 百位
SETTING:
;=========================================================================
                        ;TMOD 方式寄存器(设置方式参考C51手册)
    MOV TMOD,#01H       ; TMOD是方式寄存器0000 0001B 定时方式，不受外部控制
    MOV IE,#82H         ;允许中断,T0中断允许 中断控制字;
                        ;直接对中断寄存器IE 和优先级寄存器 IP设置
    ORL IP,#2H          ;逻辑或,T0中断的优先级高
;=========================================================================

    SETB P1.1       ;CE1 置1
    SETB P1.4       ;CE2 置1
                    ;只要将CE1和CE2分别置为高，然后IN1和IN2按照预定的脉冲输出
                    ;即 01->11->10->00->01
                    ;这个循环构成一个方向旋转的输出脉冲
                    ;将此序列翻转，就是相反方向的输出脉冲。
                    ;P3.7---S2,
                    ;P3.6---S1,
                    ;R1---循环次数
                    ;R0---存20H---脉冲时序,
                    ;R2---快慢速
;=========================================================================
GET_TIME_ORDER:
                            ;判断S2，当按下S2开关时，按照顺时针旋转
                            ;当松开时，按照逆时针旋转
                            ;JB 为1跳转
                            ;若按下则为0 ---> 不跳转 顺时针
                            ;松开则为1   ---> 跳转  逆时针
    JB P3.7,REVERSE         ;若P3.7=1,s2松开跳转到OPP
    MOV R0, #01111000B      ;按下s2为顺时针   R0=78   01-->11-->10-->00
    MOV 20H,R0              ;将步进电机的脉冲时序排序存储到20H地址中
    LJMP JUDGE_SPEED
REVERSE:
```

```
    MOV R0, #00101101B      ;松开为逆时针     R0=2D    00-->10-->11-->01
    MOV 20H,R0              ;将步进电机的脉冲时序排序存储到20H地址中
;=========================================================================
JUDGE_SPEED:
                            ;当按下S1开关时，进行快速旋转，速度为60转/分。
                            ;当松开开关时，进行慢速旋转，速度为10转/分。
                            ;JB 为1跳转
    JB P3.6,SLOW_SPEED      ;若P3.6=1,s1松开---->慢速 跳转;
                            ; P3.6=0，s1按下---->快速
    MOV R2,#0H              ;快速,(5D3E #DIGIT_TABLE)
    LJMP STEP_BY_ORDER
SLOW_SPEED:
    MOV R2,#1H              ;慢速
STEP_BY_ORDER:
    MOV R1,#4               ;相位四次变换，将对应的循环次数4保存到R1中
    MOV R0,20H              ;取出步进电机的脉冲时序
                            ;只要将CE1和CE2分别置为高，然后IN1和IN2按照预定的脉冲输出
                            ;即 01->11->10->00->01
                            ;这个循环构成一个方向旋转的输出脉冲
                            ;将此序列翻转，就是相反方向的输出脉冲。
STEPPING:
                            ; A 即 ACC 累加器
                            ; C 即为 CY 进位标志 同时也是布尔处理机的累加器C
                            ; PSW 字节地址D0H 位寻址地址为D0-D7
                            ; RLC A    将A和进位标志一起向左循环移动一位
                            ; 位7移入CY，CY移入位0
    MOV A,R0                ; A=R0 存放脉冲时序
                            ; R0=78    0(In1)1(In2)-->11-->10-->00
                            ; R0=2D    00-->10-->11-->01
    RLC A                   ; 累加器A 循环左移
    MOV P3.2,C              ;IN1 脉冲高一位送至INT1
    RLC A                   ; 再次左移一位
    MOV P1.0,C              ; IN2 低一位送至INT2
    MOV R0,A                ; 将累加器A循环左移两位之后的结果保存到R0中(即新的时序)
;=========================================================================
    LCALL SHOW_DIGITS       ;LED显示器显示步进电机的已转动的次数
    LCALL CLOCKING          ;定时器
    DJNZ  R1,STEPPING       ;R1=R1-1，结果不为0继续循环，循环次数4(R1=4)
    LJMP GET_TIME_ORDER     ;重新判断开关是否按下，死循环
;=========================================================================
                            ;定时器子程序
                            ;R2 = 1H 慢速
                            ;R2 = 0H 快速
                            ;CJNE 指令比较两个操作数是否相等，如果不相等则转移。
                            ;如果第一操作数小于第二个操作数，则置位CY，否则CY 清零。
CLOCKING:
    CJNE R2,#1,FAST_COUTING ;跳转说明R2=0H 快速
    MOV R6,#6               ;慢速六次计时
INIT_COUNT:
                            ;得到的计数器初值s需要分成高8位和低8位分别放入计数器TH0和TL0
;=========================================================================
; <关于计算部分如下>
;系统的频率为 12MHz
;转动一周需要 24步
;快速转动时，转速为:  n=60(r/min)=1(r/s)
;定时周期为: T=1/(1MHz)= 1e-6 (s)
;每一步的用时为: t=1/24(s)
;计时器位数: b=16
```

```
;因此设初始的计数为s，则有
;(2^b - s)*T=t
;解得 s=23870 ---> 5D3EH
;因此 TH0=5DH TL0=3EH
;================================================================================
    MOV TH0,#5DH            ;初值5D3E
    MOV TL0,#3EH
    SETB TR0                ;计时器0启动，如果CPU响应定时器中断则此位为1时
                            ;发生定时器中断，在中断响应时由硬件清零。
    MOV R7,#0H              ;R7为中断判断标志，置0
SLOW_COUNTING:              ;Slower
                            ;R7=0H
    CJNE R7,#1H,SLOW_COUNTING
                            ;空循环，等待定时器中断产生,如果中断，R7=1H
                            ;等于0则顺次执行
                            ;这组指令把源操作数减 1，结果回送到源操作数中
                            ;如果结果不为 0 则转移。不改变标志位。
    DJNZ R6,INIT_COUNT      ;R6-1，结果不为0继续循环(慢速继续)
                            ;慢速为快速的1/6
    LJMP OUT
FAST_COUTING:
    MOV TH0,#5DH            ;定时器0启动,快速状态60转/分
    MOV TL0,#3EH
    SETB TR0                ;定时器0启动，如果CPU响应定时器中断则此位为1时
                            ;发生定时器中断，在中断响应时由硬件清零。
    MOV R7,#0H              ;R7为中断判断标志，置0
DO_FAST:                    ;Faster
    CJNE R7,#1H,DO_FAST     ;若等于0则顺次执行即直接跳出
                            ;慢速的需要R6变成0才跳出(循环6次)。
OUT:
RET
;================================================================================
INTERRUPT_T0:               ;中断程序
    MOV R7,#1               ;中断标志置1
RETI
;================================================================================
SHOW_DIGITS:                ;调用LED显示器的子程序，显示步进电机已转动的次数
DISPLAY:
                            ;R3个位
                            ;R4十位
                            ;R5百位
    MOV A,R3
    CALL TO_TUBE            ;显示个位
    MOV A,R4
    CALL TO_TUBE            ;显示十位
    MOV A,R5
    CALL TO_TUBE            ;显示百位
                            ;已转动的步数加1,达到999时归零
                            ;R3!=9时跳转 自增返回
    CJNE R3,#9,S1           ;个位
    MOV R3,#0               ;进位
    CJNE R4,#9,S2           ;十位
    MOV R4,#0
    CJNE R5,#9,S3           ;百位
    MOV R5,#0
S1:
    INC R3
    LJMP DONE
```

```
S2:
    INC R4
    LJMP DONE
S3:
    INC R5
    LJMP DONE
DONE:
    RET
;================================================================================
;<关于数码管显示的说明>
;三个74HC164进行级联，作为数码管
;使用单片机P4.5=DAT 作为模拟串口数据，使用P4.4=CLK模拟串口时钟
;CLR 端接高电平
;使用上一个74HC164的Q7(第7位)作为下一个74HC164的输入端
;74HC164是8位边沿触发式移位寄存器，串行输入数据，然后并行输出。
;数据通过两个输入端（A或B）之一串行输入；
;任一输入端可以用作高电平使能端，控制另一输入端的数据输入。
;两个输入端或者连接在一起，或者把不用的输入端接高电平，一定不要悬空。
;时钟 (CLK) 每次由低变高时，数据右移一位，输入到Q0，Q0 是两个数据
;输入端（A 和B）的逻辑与，它将上升时钟沿之前保持一个建立时间的长度。
;主复位(CLR)输入端上的一个低电平将使其它所有输入端都无效，同时非同步
;地清除寄存器，强制所有的输出为低电平。
;================================================================================
TO_TUBE:                    ;显示数码管
                            ;R0 存在21H处
    MOV 21H,R0              ;压栈，保存之前R0的值
                            ; A=R3 即位数字，段码表中按数字递增存放
                            ;故 @A+DPTR 就是R3中对应数字的段码
    MOVC A,@A+DPTR          ;将累加器与数据指针寄存器的值相加存到A中
    MOV R0,#8               ;循环8次，（由高到低8个bit）

TRIGGER:
    CLR CLK                 ;P4.4 通过CLR清0 低电平
    RLC A                   ;累加器A左移一位，最高位移到C中
    MOV DAT,C               ;8位数据按位输出
    SETB CLK                ;P4.4  时钟线高电平，产生上升沿
    DJNZ R0,TRIGGER         ;不为0 跳转
    MOV R0,21H              ;弹栈，恢复R0
    RET                     ;返回主程序
;================================================================================
DIGIT_TABLE:                ; 段码表
                            ; 0---> 0C0H
                            ; 9---> 90H
    DB 0C0H,0F9H,0A4H,0B0H,99H,92H,82H,0F8H,80H,90H
END
```