jmp main

org 0x000B

jmp ISR\_T0

org 0x001B

jmp ISR\_T1

directionFlag bit 0x00 ;(1):for increasing duty cycle, (0) for decreasing

DutyFlag bit 0x01 ;(1):positive portion of duty cycle, (0):negative

DutyCycleByte equ r0 ;duty cycle time byte valuefor timer 0

org 0x30

main:

mov TMOD, #0x11 ;config Timer0 mode1 and Timer1 mode1

mov p1, #0x00 ;configure p1 as output

setb ET0 ;enable Timer0 interrupt

setb ET1 ;enable Timer1 interrupt

setb EA ;enable global interrupt

;load timer registers with 0x0000

mov TH0, 0x00

mov TL0, 0x00

mov TH1, 0x00

mov TL1, 0x00

setb TR0 ;start Timer0

setb TR1 ;start Timer1

mainloop:

sjmp $

ISR\_T1:

mov TH1, #0xC0

jb directionFlag, increase

;decreasing section

cjne DutyCycleByte, #0x60, not0x60

cpl directionFlag ;directionFlag = ~directionFlag

inc DutyCycleByte ;DutyCycleByte++

reti ;return from interrupt service route

not0x60:

dec DutyCycleByte

reti ;return from interrupt service route

increase:

cjne DutyCycleByte, #0xFF, not0xFF

cpl directionFlag ;directionFlag = ~directionFlag

dec DutyCycleByte ;DutyCycleByte--

reti ;return from interrupt service route

not0xFF:

inc DutyCycleByte ;DutyCycleByte++

reti ;return from interrupt service route

ISR\_T0:

mov TH0, #0xFF

cpl DutyFlag ;DutyFlag = ~DutyFlag

jnb DutyFlag, off

mov TL0, DutyCycleByte

mov p1, #0xFF ;turn on led

reti ;return from interrupt service route

off:

mov a, #0xFF

clr cy

subb a, DutyCycleByte

mov TL0, a ;TL0 =- DutyCycleByte

mov p1, #0x00 ;turn off led

reti ;return from interrupt service route

end