JMP Main

charIndex EQU 0xE0 ;alias variable for Accumulator

DutyByte EQU 0x30

DutyFlag BIT 0x00

ORG 0x0B

JMP T0\_ISR

ORG 0x30

Main:

MOV P1, #0x00 ;Configure port P1 as output

MOV TMOD, #0x21 ;set Timer 1 mode 2 and Timer 0 mode 1

MOV TH1, #0xFD ;set Timer 1 to operate on 9600 baud

MOV SCON, #0x50 ;configure Serial communication

SETB TR1 ;Start Timer 1

MOV DPTR, #Welcome

CALL WriteString

MOV DPTR, #DutyValues

MOV TH0, #0xFF

MOV TL0, #0x60

SETB ET0

SETB EA

SETB TR0

Main\_Loop:

JB RI, not0RI

;RI is equal to 0

jmp Main\_Loop

T0\_ISR:

mov TH0, #0xFF ;load timer high 0 with the value 0xff

cpl DutyFlag ;complement duty flag

jb DutyFlag, T\_ON ;check to see if dutyflag is set to on or off

;T\_OFF Delay Cycle

mov a, #0xFF ;move into a the value 0xff

clr cy ;clear the carry flag

subb a, DutyByte ;complements DutyCycleByte

mov TL0, a ;move into the timer low 0 the value in a

mov p1, #0x00 ;turn off all led lights

reti ;return from interrupt service routine

GetChar:

MOV a, SBUF

CLR RI

RET

WriteString:

MOV charIndex, #0x00

again:

PUSH charIndex

MOVC a, @a + DPTR

JNZ not0

;a is equal to 0

POP charIndex

RET

WriteChar:

MOV SBUF, a

JNB TI, $

CLR TI

RET

T\_ON:

mov TL0, DutyByte ;mov the hex value #0xFF to DutyCycleByte

mov p1, #0xFF ;turn on all the led lights

reti ;return from interrupt

not0:

CALL WriteChar

POP charIndex

INC charIndex

JMP again

not0RI:

CALL GetChar

ANL a, #0x0F

MOV b, a

MOV a, #9

CLR CY

SUBB a, b

JC Main\_Loop

;CY is not equal to 1

MOV a, b

MOVC a, @a + DPTR

MOV DutyByte, a

jmp Main\_Loop

ORG 0x200

Welcome: DB "Enter a value 0 through 9 for LED brightness control: ",0

DutyValues: DB 0x60, 0x70, 0x80, 0x90, 0xA0, 0xB0, 0xC0, 0xD0, 0xE0, 0xF0

END