

```

'*****
' *
' *-----LEDB-----*
' * Task : Sets the various bits in the BIOS keyboard *
' * flag, causing the LEDs on the AT keyboard to *
' * flash. *
' * QuickBASIC and the QB.LIB must be loaded using *
' * QB /L QB *
' * before loading and running this file. *
' *-----*
' * Author : Michael Tischer *
' * Developed on : 06/08/91 *
' * Last update : 01/20/92 *
'*****
'
DECLARE SUB SetFlag (Flag AS INTEGER)
DECLARE SUB Delay (Pause AS INTEGER)
DECLARE SUB ClrFlag (Flag AS INTEGER)

'$INCLUDE: 'QB.BI' 'Include register declarations

CONST SCRL = 16 'SCROLL LOCK bit
CONST NUML = 32 'NUM LOCK bit
CONST CAPL = 64 'CAPS LOCK bit

DIM Counter AS INTEGER 'Loop counter

CLS 'Clear screen
PRINT " LEDB - (c) 1988, 92 by Michael Tischer"
PRINT
PRINT " Look at the LEDs on your keyboard"

FOR Counter = 1 TO 10 'Run through the loop 10 times
    SetFlag (CAPL) 'Turn CAPS on
    Delay (100) 'Wait 100 milliseconds
    ClrFlag (CAPL) 'Turn CAPS off again
    SetFlag (NUML) 'Turn NUM LOCK on
    Delay (100) 'Wait 100 milliseconds
    ClrFlag (NUML) 'NUM LOCK off again
    SetFlag (SCRL) 'Turn SCROLL LOCK on
    Delay (100) 'Wait 100 milliseconds
    ClrFlag (SCRL) 'Turn SCROLL LOCK off again
NEXT

FOR Counter = 1 TO 10 'Run through the loop 10 times
    SetFlag (CAPL OR SCRL OR NUML) 'Turn all three flags on
    Delay (500) 'Wait 500 milliseconds
    ClrFlag (CAPL OR SCRL OR NUML) 'Turn all three flags off
    Delay (500) 'Wait 500 milliseconds
NEXT
END

'*****
' * ClrFLAG : Clears a flag in the BIOS status byte. *
' * Input : The flag to be cleared (see constants) *
' * Output : None *
'*****
'
SUB ClrFlag (Flag AS INTEGER)

DIM Register AS RegType 'Processor registers

DEF SEG = &H40 'Segment address of BIOS keyboard status byte
POKE &H17, (PEEK(&H17) AND (NOT Flag)) 'Clear keyboard status byte
Register.ax = 1 * 256 'AH = Function number: Character ready?
CALL INTERRUPT(&H16, Register, Register) 'Call BIOS keyboard interrupt

END SUB

'*****
' * Delay : Wait a certain length of time in milliseconds. *
' * Input : PAUSE = The number of milliseconds to wait. *
' * Output : None *
'*****
'
SUB Delay (Pause AS INTEGER)

```

```

DIM Register AS RegType          'Processor registers for interrupt
DIM Time AS LONG                  'Get the target time value

Register.ax = 0                   'Function no.: Read time counter
CALL INTERRUPT(&H1A, Register, Register) 'Call BIOS timer interrupt
Time = Register.dx + (Register.cx * 32768) 'Compute target
Time = Time + (Pause * 18 + ((Pause * 2) / 10)) / 1000 'time value
DO
    CALL INTERRUPT(&H1A, Register, Register) 'Call timer interrupt until
LOOP WHILE (Register.dx + (Register.cx * 32768)) <= Time 'time elapses

END SUB

'*****
'*  SETFLAG : Sets a flag in the BIOS status byte. *
'*  Input   : The flag to be set (see constants). *
'*  Output  : None *
'*****
'
SUB SetFlag (Flag AS INTEGER)

DIM Register AS RegType          'Processor registers for interrupt call

DEF SEG = &H40                   'Segment address of BIOS keyboard status byte
POKE &H17, (PEEK(&H17) OR Flag) 'OR status byte flag
Register.ax = 1 * 256            'AH = Function number: Character ready?
CALL INTERRUPT(&H16, Register, Register) 'Call BIOS keyboard interrupt

END SUB

```