

BASIC listing: MF2B.BAS

```
*****
' *                                     M F 2 B                                     *
' -----
' *      Task:           Demonstrates keyboard reading MF-II keyboards.      *
' *                    QuickBASIC and the QB.LIB must be loaded using      *
' *                    QB /L QB                                             *
' *                    before loading and running this file.                *
' -----
' *      Author       : Michael Tischer                                     *
' *      Developed on  : 01/01/92                                           *
' *      Last update   : 04/07/95                                           *
' *****
'
'$INCLUDE: 'QB.BI'                  'Include file contains register declarations

DECLARE FUNCTION MakeWord! (WNum AS INTEGER)
DECLARE FUNCTION HexByte$ (bval AS INTEGER)
DECLARE FUNCTION GetMFKey% ()
DECLARE FUNCTION TestMF% ()

CONST TRUE = -1                      'Define the truth
CONST FALSE = NOT TRUE

'-- Main program -----
DIM pdkey AS INTEGER
DIM CR AS STRING

CLS
CR = CHR$(13)
PRINT "MF2B - (c) 1992 by Michael Tischer"; CR
```

```

IF TestMF THEN
    PRINT "BIOS functions implemented for MF-II keyboards."
    PRINT CR + CR + "Press any key or combination to display ";
    PRINT "key codes." + CR + CR
    PRINT "Press <Esc> to end the program." + CR

    DO
        pdkey = GetMFKey                                'Input loop
                                                    'Get key
        PRINT "Scan : "; HexByte(MakeWord(pdkey) / 256); " ";
        PRINT "ASCII: "; HexByte(pdkey AND 255);
        IF ((pdkey AND 255) = &HE0) AND ((pdkey / 256) <> 0) THEN
            PRINT " <---- MF-II key"
        ELSE
            PRINT
        END IF
        LOOP UNTIL (pdkey = &H1B)                    'Repeat until user presses <ESC>
        PRINT CR
    ELSE
        PRINT "No BIOS extensions available for MF-II keyboards!"
    END IF
END

' *****
' * GetMFKey   : Reads a key using extended keyboard function 10H.      *
' * Input      : None                                                    *
' * Output     : The returned keycode                                    *
' *****
'
FUNCTION GetMFKey%

DIM reg AS RegType                                'Processor registers for interrupt call

```

```

reg.ax = &H1000                'Extended read function for MF-II keyboards
CALL INTERRUPT(&H16, reg, reg)  'Call BIOS keyboard interrupt
GetMFKey% = reg.ax              'Return keycode

```

END FUNCTION

```

' *****
' * HexByte : Changes a byte into a two-digit hex string.      *
' * Input   : BVAL = Byte to be converted                      *
' * Output  : Two-digit hex string                             *
' *****
'

```

FUNCTION HexByte\$ (bval AS INTEGER)

```

IF bval < 16 THEN                'One digit?
    HexByte$ = "0" + HEX$(bval)  'Yes --> First digit = "0"
ELSE                             'No --> Make two digits
    HexByte$ = HEX$(bval)
END IF
END FUNCTION

```

```

' *****
' * Makeword : Makes a long number from an integer, to avoid getting *
' *           a negative result during bit manipulations performed      *
' *           through integer division.                                  *
' * Input    : Integer number                                           *
' * Output   : Bit pattern compatible long number                      *
' *****
'

```

FUNCTION MakeWord! (WNum AS INTEGER)

```

IF WNum < 0 THEN

```

```

    MakeWord = 65536! + WNum
ELSE
    MakeWord = WNum
END IF

```

```

END FUNCTION

```

```

! *****
! * TestMF: Tests whether the extended BIOS functions for reading the *
! *           MF-II keyboard are available. *
! * Input    : None *
! * Output   : TRUE if the functions are available, otherwise FALSE *
! *****
!

```

```

FUNCTION TestMF%

```

```

    DIM reg AS RegType                'Processor registers for interrupt call

```

```

    reg.ax = &H1200                    'Extended status function for MF-II keyboards
    CALL INTERRUPT(&H16, reg, reg)     'Call BIOS keyboard interrupt
    PRINT HEX$(reg.ax)
    TestMF% = (reg.ax <> &H1200)       'AX =1200H : Function absent

```

```

END FUNCTION

```