

```

/*****
/*
/*----- M F 2 C -----*/
/* Task: : Demonstrates key read from MF-II keyboards. */
/*-----*/
/* Author : Michael Tischer */
/* Developed on : 01/01/92 */
/* Last update : 04/07/95 */
/*****

/*== Add include files =====*/

#include <stdio.h>
#include <dos.h>

/*== Type definitions =====*/

typedef unsigned char BYTE; /* Create a byte */
typedef unsigned int WORD;

/*== Constants =====*/

#define TRUE ( 0 == 0 ) /* Constants make reading */
#define FALSE ( 0 == 1 ) /* program code easier */

/*== Screen routines (Microsoft C) =====*/

#ifdef __TURBOC__ /* Microsoft C? */

/*****
/* Gotoxy : Places cursor. */
/* Input : Cursor coordinates */
/* Output : None */
/*****

void gotoxy( int x, int y )
{
    union REGS regs; /* Register variables for interrupt call */

    regs.h.ah = 0x02; /* Function number for interrupt call */
    regs.h.bh = 0; /* Color */
    regs.h.dh = y - 1;
    regs.h.dl = x - 1;
    int86( 0x10, &regs, &regs ); /* Interrupt call */
}

/*****
/* clrscr : Clears the screen. */
/* Input : None */
/* Output : None */
/*****

void clrscr( void )
{
    union REGS regs; /* Register variables for interrupt call */

    regs.h.ah = 0x07; /* Function number for interrupt call */
    regs.h.al = 0x00;
    regs.h.ch = 0;
    regs.h.cl = 0;
    regs.h.dh = 24;
    regs.h.dl = 79;
    int86( 0x10, &regs, &regs ); /* Interrupt call */
    gotoxy( 1, 1 ); /* Set cursor */
}

#endif

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

char *HexByte( BYTE bval )
{

```

```

char HexDigits[16] = "0123456789ABCDEF";
static char dummy[3] = "00";

dummy[0] = HexDigits[ bval >> 4 ];          /* Convert both */
dummy[1] = HexDigits[ bval & 0x0F ];         /* nibbles to hex */
return dummy;
}

/*****
/* 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  */
*****/

int TestMF( void )
{
    union REGS regs;          /* Register variables for interrupt call */

    regs.x.ax = 0x1200; /* Extended status function for MF-II keyboards */
    int86( 0x16, &regs, &regs );
    return ( regs.x.ax != 0x1200 ); /* AX=0x1200 : Function absent */
}

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

WORD GetMFKey( void )
{
    union REGS regs;          /* Register variables for interrupt call */

    regs.h.ah = 0x10; /* Extended read function for MF-II keyboards */
    int86( 0x16, &regs, &regs );
    return regs.x.ax;          /* Return keycode */
}

/*****
/*           M A I N   P R O G R A M                               */
*****/

void main( void )
{
    WORD pdkey;

    clrscr();
    printf( "MF2C - (c) 1992 by Michael Tischer\n\n" );
    if ( TestMF() )
    {
        printf( "BIOS functions implemented for MF-II keyboards.\n\n"
                "Press any key or combination to display key codes.\n\n"
                "Press <Esc> to end the program.\n\n" );

        do
        {
            pdkey = GetMFKey();          /* Get key */
            printf( "Scan : %s ", HexByte((BYTE) (pdkey >> 8)) );
            printf( "ASCII: %s", HexByte((BYTE) (pdkey & 255)) );
            if ( ((pdkey & 255) == 0xe0) && ((pdkey & 65280) != 0) )
                printf( " <---- MF-II key" );
            printf( "\n" );
        }
        while ( pdkey != 0x011b ); /* Repeat until user presses <ESC> */
        printf( "\n\n" );
    }
    else
        printf( "No BIOS extensions available for MF-II keyboards!");
}

```