

C listing: HMAC.C

```

/*****
*
*                               H M A C . C
*
**-----**
* Description      : Demonstration of direct access to the HMA without
*                   the assistance of any special drivers.
*
**-----**
* Author          : MICHAEL TISCHER
* Developed on    : 07/27/1990
* Last update on  : 04/07/1995
**-----**
* (MICROSOFT C)
* Creation        : CL /AS /Zp hmac.c hmaca
* Call            : hmac
**-----**
* (BORLAND TURBO C)
* Creation        : create a project file with the following contents
*                   hmac.c
*                   hmaca.obj
*****/

/*-- Include files -----*/

#include <dos.h>                                /* for interrupt call */

#ifdef __TURBOC__
    #include <alloc.h>
#else
    #include <malloc.h>
#endif

/*-- Constants -----*/
```

```

#define TRUE  ( 0 == 0 )
#define FALSE ( 0 == 1 )

/*-- Macros -----*/

#ifndef MK_FP
    #define MK_FP(seg,ofs) \
        ((void far *) (((unsigned long)(seg) << 16) | (unsigned)(ofs)))
#endif

#define Hi(x) (*(BYTE *) &x+1))          /* Hi-Byte one ints */
#define Lo(x) (*(BYTE *) &x))            /* Lo-Byte one ints */

/*-- Type declarations -----*/

typedef unsigned char BYTE;
typedef BYTE BOOL;
typedef unsigned WORD;

/*-- extern declarations -----*/

extern BOOL HMAAvail( void ); /* HMA available? */
extern BOOL GateA20( BOOL free ); /* A20 locked/free */
extern BOOL IsA20On( void ); /* A20 available? */

/*****
* HMA Test : Demonstration of accessing the HMA
*-----*
* Input   : none
*-----*/

```

```

void HMAtest( void)

{
    BYTE far * hmap;                                /* Pointer to the HMA */
    WORD i,                                           /* loop counter */
        err;                                         /* Number of the error for HMA access */

    if ( IsA20On() )
        printf( "The address line A20 is already switched on!\n" );
    else
        if ( GateA20( TRUE ) == FALSE || IsA20On() == FALSE )
        {
            printf( "Note! The address line A20 can not be " \
                "be made available." );
            return;
        }
    else
        printf( "The access to the HMA is switched on.\n" );

    hmap = MK_FP( 0xFFFF, 0x0010 );                /* Pointer to HMA */
    err = 0;                                         /* start with no errors */
    for ( i = 1; i < 65520; ++i, ++hmap )
    {
        /* test the memory locations */
        printf( "\rMemory location: %u", i );
        *hmap = i % 256;                          /* memory location description */
        if ( *hmap != i % 256 )                   /* and return selection */
        {
            printf( " ERROR!\n" );                /* ERROR! */
            ++err;
        }
    }
}

```

```

printf( "\n" );
if ( err == 0 )                                /* Output test results */
    printf( "HMA ok, no defective memory locations.\n" );
else
    printf( "ATTENTION: %d defective memory locations in the HMA " \
            "discovered!\n", err );
GateA20( FALSE ); /* Address line switched off */
}

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

void main( void )
{
    int i; /* loop counter */

    for ( i = 1; i < 25; ++i )                    /* clear screen */
        printf ( "\n" );

    printf("HMAC - HMA Demo program by MICHAEL TISCHER\n\n" );
    if ( HMAAvail() )
    {
        HMAtest();                                /* HMA test */
        printf( "\n" );
    }
    else
        printf( "No access to HMA possible.\n" );
}

```