

```

;*****
;*                               M O U S E C A                               *
;*-----*
;* Task      : Mouse driver event handler intended for      *
;*            linking to a C program compiled as a SMALL    *
;*            memory model.                                  *
;*-----*
;* Author     : MICHAEL TISCHER                               *
;* Developed on : 04/20/1989                                   *
;* Last update  : 06/14/1989                                   *
;*-----*
;* assembly   : MASM /MX MOUSECA; or TASM -MX MOUSECA        *
;*            ... link to program MOUSEC                     *
;*****

;== Segment declarations for the C program =====

IGROUP group _text          ;Program segment
DGROUP group _bss, _data    ;Data segment
      assume CS:IGROUP, DS:DGROUP, ES:DGROUP, SS:DGROUP

_BSS   segment word public 'BSS' ;This segment includes all un-
_BSS   ends                     ;initialized static variables

_DATA  segment word public 'DATA' ;This segment includes all initialized
_DATA  ends                     ;global and static variables

;== Program =====

_TEXT  segment byte public 'CODE' ;Program segment

public  _AssmHand              ;Gives the C program the ability to
                                ;access assembler handler addresses

extrn   _MouEventHandler : near ;Event handler to be called

active  db  0                  ;Indicates whether a call is under
                                ;execution

;-----
;-- _AssmHand: The event handler called by the mouse driver, then
;--             called by the MouEventHandler() function
;-- Call from C: not allowed!

_AssmHand  proc far

      ;-- Place all processor registers on the stack ---

      cmp  active,0            ;Call still not finished?
      jne  ende                ;NO --> Do not exit call

      mov  active,1            ;No more calls

      push ax
      push bx
      push cx
      push dx
      push di
      push si
      push bp
      push es
      push ds

      ;-- Place all arguments for calling C_FCT on the stack ---
      ;-- Call: MouEventHandler( int EvFlags, int ButStatus,
      ;--                               int x,          int y );

      mov  di,cx                ;Place horizontal coordinate in DI
      mov  cl,3                 ;Counter for coordinate number
      shr  dx,cl                 ;Divide DX (vertical coord.) by 8
      push dx                   ;and place on the stack

      shr  di,cl                 ;Divide DI (horizontal coord.) by 8
      push di                   ;and place on the stack

```

```

push bx          ;Push mouse button status onto stack
push ax          ;Push event flag onto stack

mov ax,DGROUP    ;Move segment address of DGROUP to AX
mov ds,ax        ;Move AX to DS register

call _MouEventHandler ;C function call

add sp,8          ;Get arguments from stack

;-- Pop register contents off of stack -----

pop ds
pop es
pop bp
pop si
pop di
pop dx
pop cx
pop bx
pop ax

mov active,0      ;Re-enable call

ende:            ret          ;Return to mouse driver

_AssmHand endp

;-----

_text            ends          ;End of code segment
end              end          ;End of program

```