MSX BANZAI!

DEVELOPMENT | COMPUTERS | MAGAZINES | CURIOSITIES
GFX9000/VIDEO9000 | SOFTWARE | FAQ

How do I boot a cartridge from BASIC? *by Albert Beevendorp*

```
DEFUSR=&H7D75:A=USR(0)
```

The DiskROM doesn't like to be initialised twice, so booting cartridges that use the diskdrive will crash.

How do I (re)boot DOS in assembly, without going to BASIC? *by Roberto Vargas*

```
Call_System:
        ld      hl,Start         ; relocate to page 3 and execute
        ld      de,0C000h
        push    de
        ld      bc,End-start
        ldir
        ret

Start:  ld      a,(0FCC1h)
        push    af
        ld      h,00h
        call    0024h
        pop     af
        ld      h,40h
        call    0024h
        xor     a
        ld      hl,0F41Fh
        ld      (0F860h),hl
        ld      hl,0F423h
        ld      (0F41Fh),hl
        ld      (hl),a
        ld      hl,0F52Ch
        ld      (0F421h),hl
        ld      (hl),a
        ld      hl,0F42Ch
        ld      (0F862h),hl
        ld      hl,0C000h+System-Start
        jp      4601h
System: db      3Ah,0CAh,"SYSTEM",0
End:
```

DOS2 only: It's possible to directly load a different program or batchfile using: SYSTEM("progname params"), but in order to avoid an error on exit, you must set the "PROG" environment variable to "$SHELL" using function 6Ch.

How do I convert a binary number to hexadecimal in assembly? *by Marcel Delorme*

```
; HL = pointer to hex output buffer (upper case ASCII)
;  A = binary number to convert
Bin2Hex:
        ld      b,a
        rrca
        rrca
        rrca
        rrca
        and     0Fh             ; These instructions are the important ones
        cp      10              ;
        sbc     a,69h           ;
        daa                     ;
        ld      (hl),a
        inc     hl
        ld      a,b
        and     0Fh
        cp      10
        sbc     a,69h
        daa
        ld      (hl),a
        ret
```

How do I quickly divide by 10 without using loops? *by Patriek Lesparre*

```
; In: A = dividend
; Out: B = quotient, A = rest
; Chg: AF,AF',BC
Div10:  ld      c,a
        sub     10
        ex      af,af'          ; rest = quotient - 10
        ld      a,c
        srl     c
        srl     c
        sub     c               ; quotient = quotient - quotient >> 2
        srl     c
        srl     c
        srl     c
        add     a,c             ; quotient = quotient + quotient >> 5
        rrca
        rrca
        rrca
        and     00011111b       ; quotient = quotient >> 3
        ld      b,a
        add     a,a
        add     a,a
        add     a,b             ; temp = quotient + quotient << 2
        add     a,a
        ld      c,a             ; temp = temp << 1
        ex      af,af'
        sub     c               ; rest = rest - temp
        jp      p,.plus         ; IF rest < 0 THEN
        add     a,10            ;   rest = rest + 10
        ret
.plus:                          ; ELSE
        inc     b               ;   quotient = quotient + 1
        ret                     ; END IF
```

How do I call the MSX2 SubROM from MSX-DOS? *by Alex Wulms*

```
; CALSUB
;
; In: IX = addr of routine in MSX2 subrom
;     AF, HL, DE, BC = parameters for the routine
```

```
;
; Out: AF, HL, DE, BC = depending on the routine
;
; Changes: IX, IY, AF', BC', DE', HL'
;
; Call MSX2 subrom from MSXDOS. Should work with all versions of MSXDOS.
;
; Notice: NMI hook will be changed. This should pose no problem as NMI is
; not supported on the MSX at all.
;
CALLSLT equ 001ch
NMI     equ 066h
EXTROM  equ 015fh
EXPTBL  equ 0fcc1h
H_NMI   equ 0fdd6h
;
CALSUB:  exx
         ex   af,af'       ; store all registers
         ld   hl,EXTROM
         push hl
         ld   hl,0c300h
         push hl           ; push NOP ; JP EXTROM
         push ix
         ld   hl,021ddh
         push hl           ; push LD IX,
         ld   hl,03333h
         push hl           ; push INC SP; INC SP
         ld   hl,0
         add  hl,sp        ; HL = offset of routine
         ld   a,0c3h
         ld   (H_NMI),a
         ld   (H_NMI+1),hl ; JP  in NMI hook
         ex   af,af'
         exx               ; restore all registers
         ld   ix,NMI
         ld   iy,(EXPTBL-1)
         call CALLSLT      ; call NMI-hook via NMI entry in ROMBIOS
                           ; NMI-hook will call SUBROM
         exx
         ex   af,af'       ; store all returned registers
         ld   hl,10
         add  hl,sp
         ld   sp,hl        ; remove routine from stack
         ex   af,af'
         exx               ; restore all returned registers
         ret
```

How do I create a 48K ROM? *by Armando Pérez Abad*

Here is a tutorial with example source code.

How do I use the diskdrive in a ROM? *by Armando Pérez Abad*

Here is a tutorial with example source code.

How do I use the keyboard and joystick? *by Armando Pérez Abad*

Here is a tutorial with example source code.

How do I read the mouse on all MSX models? *by Armando Pérez Abad*

Here is a tutorial with example source code.