

B_ANZ2. BI B

```

; B_ANZ2. BI B -----
BS_POS MACRO ZEILE, SPALTE
; --- Cursor an ZEILE, SPALTE stellen -----
    push    ax
    push    bx
    push    dx
    mov     ah, 02h
    mov     bh, 0
    mov     dh, ZEILE
    mov     dl, SPALTE
    int     10h
    pop     dx
    pop     bx
    pop     ax
ENDM

```

```

CCOUT MACRO ZEICH, COLOR, ANZ
; --- ANZ*ZEICH in Farbe COLOR ausgeben -----
    push    ax
    push    bx
    push    cx
    mov     ah, 09h
    mov     bh, 0
    mov     al, ZEICH
    mov     bl, COLOR
    mov     cx, ANZ
    int     10h
    pop     cx
    pop     bx
    pop     ax
ENDM

```

```

TEXT_ANZEIGE MACRO TEXT
; --- Textfolge TEXT ausgeben -----
    push    ax
    push    dx
    mov     dx, offset TEXT
    mov     ah, 09h
    int     21h
    pop     dx
    pop     ax
ENDM

```

```

BYTE_ANZ MACRO WERT, ZEI, SPA
; --- Anzeige eines Bytes als LED -----
    jmp     ausg

```

```

line1 DB 'AAAAAAAAAAAAAAAAAA;$'
line2 DB 'AAAAAAAAAAAAAAAAAAU$'
zuweis DB ' => $'
spalte DB '?'
zeile DB '?'
zh DB '?'
pos DB '?'

```

```

ausg:  nop
       push    ax
       push    bx
       push    cx
       push    dx

       mov     ax, cs
       mov     ds, ax

       mov     al, ZEI
       mov     zeile, al
       add     al, 1
       mov     zh, al

```

B_ANZ2. BI B

```

mov     al, SPA
mov     spal te, al

;..... ASCII-Zeichen ausgeben ....
mov     al, spal te
mov     pos, al
BS_POS  zh, pos
mov     dl, WERT
mov     ah, 02h
int     21h
add     pos, 1
BS_POS  zh, pos
TEXT_ANZEIGE  zuwei s
add     spal te, 5

BS_POS  zeile, spal te

TEXT_ANZEIGE  line1

;..... BYTE darstellen .....
mov     al, spal te
mov     pos, al

mov     bh, 10000000b
mov     bl, WERT
mov     cx, 8

anf:    nop
BS_POS  zh, pos
mov     dl, '3'
mov     ah, 02h
int     21h
add     pos, 1
BS_POS  zh, pos

mov     al, bl
and     al, bh
jz      null

ei ns:  nop
CCOUT  219, 12, 1
jmp     wei ter

null:   nop
CCOUT  ' ', 12, 1

wei ter:  nop
add     pos, 1
shr     bh, 1
loop   anf

;.....
BS_POS  zh, pos
mov     dl, '3'
mov     ah, 02h
int     21h

mov     al, zeile
add     al, 2
BS_POS  al, spal te

TEXT_ANZEIGE  line2

add     zh, 6
BS_POS  zh, 1

pop     dx
pop     cx
pop     bx
pop     ax
ENDM

```