```
Copyright = Venotec Ug
Www = Http : / / Www.venotec.de
Email = Info@venotec.de

Libversion = 1.00
Date = 6 Januar 2010
Statement = Source Code Is Property Of Venotec Ug
History = Startversion

;Library fuer Displaytech module umgearbeitet
;uC Speed 16MHz


;Rs = PortB.3
;RW = PortB.2
; E = PortB.1
;E2 = PortB.0        optional fuer 2 chip LCD
;Db4 = PortB.4       high nibble
;Db5 = PortB.5
;Db6 = PortB.6
;Db7 = PortB.7

; Im Programm:
;$lib "lcd4busy.lib"
;Const _lcdport = Porta
;Const _lcdddr = Ddra
;Const _lcdin = Pina
;Const _lcd_e = 1
;Const _lcd_rw = 2
;Const _lcd_rs = 3


;40ms warten nach Power on reset

    Ldi R30, 64
    Ldi R31, 156
    Rcall _lcd_delay                    ; 40 ms warten


;Displayinitialisierung:
[_init_lcd]
_init_lcd:
    clr R24                              ;alle pins displayport low setzen
  * Out _lcdport , R24
    ser r24                              ;port als output setzen
  * Out _lcdddr , R24
    Ldi R30, 64
    Ldi R31, 156
    Rcall _lcd_delay                     ;40 ms warten

    ldi r24, &B00110000                  ;Funktion Set: 8-bit interface
  * Out _lcdport , R24

  * Sbi _lcdport , _lcd_e                ;Set E
    rcall _delay1us                      ;1us warten
  * Cbi _lcdport , _lcd_e                ;clear E
    Ldi R30, 132
    Ldi R31, 19
    rcall _lcd_delay                     ;5 ms warten

  * Sbi _lcdport , _lcd_e                ;Set E
    rcall _delay1us                      ;1us warten
  * Cbi _lcdport , _lcd_e                ;clear E
    Ldi R30, 132
    Ldi R31, 19
    rcall _lcd_delay                     ;5 ms warten

  * Sbi _lcdport , _lcd_e                ;Set E
    rcall _delay1us                      ;1us warten
  * Cbi _lcdport , _lcd_e                ;clear E
    Ldi R30, 132
    Ldi R31, 19
    rcall _lcd_delay                     ;5 ms warten

    ldi r24, &B00100000                  ;Funktion Set: 4-bit interface high nibble
  * Out _lcdport , R24

  * Sbi _lcdport , _lcd_e                ;Set E
    rcall _delay1us                      ;1us warten
  * Cbi _lcdport , _lcd_e                ;clear E
    Ldi R30, 132
    Ldi R31, 19
    rcall _lcd_delay                     ;5 ms warten

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B10000000                  ;Function Set: 2-line mode, 5x7 dots
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B00000000                  ;Befehl Display on/off control: high nibble
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B11110000                  ;Befehl Display on/off control: low nibble, display on, cursor on, blink on
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B00000000                  ;Befehl clear display: high nibble
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B00010000                  ;Befehl clear display: low nibble
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    Ldi R30, 132
    Ldi R31, 19
    rcall _lcd_delay                     ;5 ms warten

    ldi r24, &B00000000                  ;Befehl entry mode set: high nibble
  * Out _lcdport , R24

    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us

    ldi r24, &B11110000                  ;Befehl entry mode set: low nibble, increment mode, entire shift off (cursor rechts, text still)
  * Out _lcdport , R24
    rcall _etoggledelay                  ;Set E, 1us, clear E, 100us
RET

_etoggledelay:
  * Sbi _lcdport , _lcd_e                ;Set E
    rcall _delay1us                      ;1us warten
  * Cbi _lcdport , _lcd_e                ;clear E
    Ldi R30, 96                          ;100 us warten
    Ldi R31, 0
    rcall _lcd_delay
RET

_lcd_delay: ; 1us pro loop
   Sbiw R30,1                           ;2 cycles
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Brne _lcd_delay                      ;2 cycles
Ret

_delay1us: ; 1us
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
   Nop                                  ;1 Cycle
```

```
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
   Nop                                ;1 Cycle
Ret

[end]

;high nibble must be written first
;rw must be low for a write
;rs must be low for control, high for data
;data passed in r24

[_write_lcd]
_write_lcd:
  rcall _LCD_Busy                     ; Wait for LCD to be ready
  ldi r25, &b11110000
  and r25, r24                        ; Get upper nibble into upper half port
  * Out _lcdport , R25 ; Send Data To Lcd

  * Sbi _lcdport , _lcd_rs ; Set Lcd To Data Mode
  rcall _write_lcd2                   ; saving code

  ldi r25, &B00001111
  and r25, r24                        ; Get lower nibble into upper half port
  Swap R25
  * Out _lcdport , R25                ; Send Data To Lcd
  * Sbi _lcdport , _lcd_rs            ; Set Lcd To Data Mode
_write_lcd2:
  * Sbi _lcdport , _lcd_e             ; Toggle E For Lcd
    rcall _delay1us                   ;1us warten
  * Cbi _lcdport , _lcd_e
  ret

_lcd_control:
  rcall _LCD_Busy                     ; Wait for LCD to be ready
  ldi r25, &B11110000
  and r25,r24                         ; Get upper nibble into upper half port
  * Out _lcdport , R25 ; Send Data To Lcd
  rcall _write_lcd2
  Swap R24
  ldi r25, &b11110000
  and r25, r24                        ; Get lower nibble into upper half port
  * Out _lcdport , R25 ; Send Data To Lcd
rjmp _write_lcd2                      ; finish there to save code

_lcd_busy:
  ldi r25, &b00001111                 ; Set high nibble to input
  * Out _lcdddr , R25
  clr r25                             ; remove pull-ups on data nibble
  * Out _lcdport , R25
  * Sbi _lcdport , _lcd_rw            ; Setup To Read Busy Flag
  * Sbi _lcdport , _lcd_e             ; Set E High
    rcall _delay1us                   ;1us warten
  * In R25 , _lcdin                   ; Read Upper Nibble Busy Flag , Ddram Address
  * Cbi _lcdport , _lcd_e             ; Set E Low
    rcall _delay1us                   ;1us warten
  * Sbi _lcdport , _lcd_e             ; Set E High To Get Rid Of Lower Nibble
    rcall _delay1us                   ;1us warten
  * Cbi _lcdport , _lcd_e             ; Set E Low
  sbrc r25, 7                         ; Check busy flag, high = busy
  rjmp _LCD_Busy                      ; if high check again
  * Cbi _lcdport , _lcd_rw            ; Back To Write Mode
  ser r25
  * Out _lcdddr , R25                 ; Set For Output
ret
[end]

[_cls]
$external _write_lcd , _init_lcd
_cls:
  Ldi R24,1                           ; Clear display
  Rcall _Lcd_control
  Ldi R30, 180
  Ldi R31, 11
  rcall _lcd_delay
  Ldi R24,128                         ; Select line 1
  Rjmp _Lcd_control                   ; finish
[end]
```