```
;-------------------------------------------------------------------------------
; Title       :   4WIRE_DOGM163_3_3V
;             :
; Functions   :   SPI with 256 kHz
;             :   64 kHz at PB1 (Pin 15)
;             ;   Display of 64 different Contrast Settings
;             ;   For temporary Stop: PC0 (Pin 23) = Low
;-------------------------------------------------------------------------------
; CPU         :   ATMEGA8A-PU
; Frequency   :   4.096 MHz
; Language    :   Assembler Studio 4
; Date        :   06.06.2015
; Version     :   1.0
; Author      :   Klaus
;-------------------------------------------------------------------------------
        .include   "m8def.inc"
;-------------------------------------------------------------------------------
; Reset and Interrupt Vector          Description

Begin: rjmp    Main                   ;  1 POWER ON RESET
       reti                           ;  2 Int0-Interrupt
       reti                           ;  3 Int1-Interrupt
       reti                           ;  4 TC2 Compare Match
       reti                           ;  5 TC2 Overflow
       reti                           ;  6 TC1 Capture
       reti                           ;  7 TC1 Compare Match A
       reti                           ;  8 TC1 Compare Match B
       reti                           ;  9 TC1 Overflow
       reti                           ; 10 TC0 Overflow
       reti                           ; 11 SPI, STC Serial Transfer Complete
       reti                           ; 12 UART Rx complete
       reti                           ; 13 UART Data Register Empty
       reti                           ; 14 UART Tx complete
       reti                           ; 15 ADC Conversion Complete
       reti                           ; 16 EEPROM Ready
       reti                           ; 17 Analog Comparator
       reti                           ; 18 TWI (I²C) Serial Interface
       reti                           ; 19 Store Program Memory Redy


;-------------------------------------------------------------------------------
; Start, Power ON, Reset, Stackpointer, Ports, Power Up Waiting Time

Main:
       ldi    R16    ,   LOW (RAMEND)  ; Stackpointer Initialization
       out    SPL    ,   R16           ;
       ldi    R16    ,   HIGH(RAMEND)  ;
       out    SPH    ,   R16           ;

       ldi    R16    ,   0b00111111    ; PORTB Bits 1 until 5 = Output
       out    DDRB   ,   R16           ; PORTB Bits 6 and   7 = Quartz
       ldi    R16    ,   0b00000000    ; PORTB Bits 0 until 7 = 0
       out    PORTB  ,   R16           ;

       ldi    R16    ,   0b00000000    ; PORTC Bits 0 until 7 = Input
       out    DDRC   ,   R16           ;
       ldi    R16    ,   0b00111111    ; PORTC Bits 0 until 5 = 1 (Pullup)
       out    PORTC  ,   R16           ;

       ldi    R16    ,   0b11111111    ; PORTD Bits 0 until 7 = Output
       out    DDRD   ,   R16           ;
       ldi    R16    ,   0b00000000    ; PORTD Bits 0 until 7 = 0
       out    PORTD  ,   R16           ;

       ldi    R16    ,   5             ; 50 ms Power Up Waiting Time
       rcall  Wait

;-------------------------------------------------------------------------------
; Timer/Counter1: 64 kHz at PB1 Pin 15

       ldi    R18    ,   (1<<COM1A0)
       out    TCCR1A ,   R18             ; COM1A0 = Toggle PB1 (Pin15)

       ldi    R18    ,   (0<<CS10) | (1<<CS11) | (1<<WGM12)
       out    TCCR1B ,   R18             ; CS10   = 0 Prescaling 8
```

```
                                        ; CS11   = 1
                                        ; WGM12  = CTC (Clear Timer on Compare Match)

        ldi     R19     ,   0b00000000
        ldi     R18     ,   0b00000011      ; 3 + 1 = 4
        out     OCR1AH  ,   R19
        out     OCR1AL  ,   R18             ; 4096 / (8 * 4) = 128 kHz (PB0 = 64 kHz)

        ldi     R18     ,   (1<<OCIE1A)     ; OCIE1A = Output Compare A Match Interrupt
        out     TIMSK   ,   R18             ;          Enable

        sei                                 ; Global Interrups Enabled

;-------------------------------------------------------------------------------------------
; CGRAM Initialization

        ldi     R20     ,   112
        ldi     R21     ,   87
        rcall   LCD_init
        rcall   CGRAM_Init
        rcall   Start
        rjmp    Loop1


;-------------------------------------------------------------------------------------------
; Initialization of the LCD DOGM162W-A 5V with 64 different Settings of the Contrast

Start:
        rcall   LCD_line1
        ldi     R16     ,   1               ; 10 ms
        rcall   Wait
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'f'
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   'r'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   'h'
        rcall   LCD_data
        ldi     R16     ,   'i'
        rcall   LCD_data
        ldi     R16     ,   's'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'p'
        rcall   LCD_data
        ldi     R16     ,   'l'
        rcall   LCD_data
        ldi     R16     ,   's'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data


        rcall   LCD_line2
        ldi     R16     ,   1               ; 10 ms
        rcall   Wait
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'w'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'i'
        rcall   LCD_data
```

```
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'p'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   'i'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   'l'
        rcall   LCD_data
        ldi     R16     ,   'y'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data


        ldi     R16     ,   250             ; 2500 ms
        rcall   Wait


        ret


;----------------------------------------------------------------------------------------
; Counting up the Contrast Values C5 C4 C3 C2 C1 C0

Loop1:
        ldi     R20     ,   112             ;        C3 C2 C1 C0 | R20 = 112...127
                                            ;                    | R20 = 0b01110000...0b01111111
        ldi     R21     ,   84              ; C5 C4              | R21 =  84... 87
                                            ;                    | R21 = 0b01010100...0b01010111
                                            ;                    | Ion: Icon Display Off
                                            ;                    | Bon: Booster Circuit On
                                            ;                    | Bon: Charge Pump (3.3V -> 5V)
Loop2:
        sbis    PINC    ,   0               ; Stop Button
        rjmp    Loop2
        rcall   LCD_Init
        ldi     R16     ,   1               ;  10 ms
        rcall   Wait
        rcall   LCD_Text
        ldi     R16     ,   50              ; 500 ms
        rcall   Wait
        inc     R20                         ; R20 = R20 + 1
        cpi     R20     ,   128             ; R20 = 128 ?
        breq    Loop3                       ; if R20  = 128
        rjmp    Loop2                       ; if R20 <= 127
Loop3:
        inc     R21                         ; R21 = R21 + 1
        cpi     R21     ,   88              ; R21 = 88 ? (possible only if  R20 = 128
                                            ;                            and R21 =  88
        breq    Loop1                       ; if R20  = 128 and R21 = 88
        ldi     R20     ,   112             ; R20 = 112
        rjmp    Loop2                       ; if R21 <= 83


;----------------------------------------------------------------------------------------
; Subroutine for the creation of Text

LCD_Text:
        ldi     ZL      ,   LOW(Text00*2)  ; " CONTRAST 543210"
        ldi     ZH      ,   HIGH(Text00*2)
        rcall   TextOut10
        ldi     ZL      ,   LOW(Text10*2)  ; "SETTING C "
        ldi     ZH      ,   HIGH(Text10*2)
        rcall   TextOut20


        cpi     R21     ,   84
```

```
                breq    T11
                cpi     R21     ,    85
                breq    T12
                cpi     R21     ,    86
                breq    T13
                cpi     R21     ,    87
                breq    T14
T11:
                ldi     ZL      ,    LOW(Text11*2)  ; "00    "
                ldi     ZH      ,    HIGH(Text11*2)
                rcall   TextOut30
                rjmp    LCD_Text01
T12:
                ldi     ZL      ,    LOW(Text12*2)  ; "01    "
                ldi     ZH      ,    HIGH(Text12*2)
                rcall   TextOut30
                rjmp    LCD_Text01
T13:
                ldi     ZL      ,    LOW(Text13*2)  ; "10    "
                ldi     ZH      ,    HIGH(Text13*2)
                rcall   TextOut30
                rjmp    LCD_Text01
T14:
                ldi     ZL      ,    LOW(Text14*2)  ; "11    "
                ldi     ZH      ,    HIGH(Text14*2)
                rcall   TextOut30
                rjmp    LCD_Text01


LCD_Text01:
                cpi     R20     ,    112
                breq    T21
                cpi     R20     ,    113
                breq    T22
                cpi     R20     ,    114
                breq    T23
                cpi     R20     ,    115
                breq    T24
                cpi     R20     ,    116
                breq    T25
                cpi     R20     ,    117
                breq    T26
                cpi     R20     ,    118
                breq    T27
                cpi     R20     ,    119
                breq    T28
                cpi     R20     ,    120
                breq    T29
                cpi     R20     ,    121
                breq    T30
                cpi     R20     ,    122
                breq    T31
                cpi     R20     ,    123
                breq    T32
                cpi     R20     ,    124
                breq    T33
                cpi     R20     ,    125
                breq    T34
                cpi     R20     ,    126
                breq    T35
                cpi     R20     ,    127
                breq    T36


T21:    rjmp    LCD_Text21
T22:    rjmp    LCD_Text22
T23:    rjmp    LCD_Text23
T24:    rjmp    LCD_Text24
T25:    rjmp    LCD_Text25
T26:    rjmp    LCD_Text26
T27:    rjmp    LCD_Text27
T28:    rjmp    LCD_Text28
T29:    rjmp    LCD_Text29
T30:    rjmp    LCD_Text30
T31:    rjmp    LCD_Text31
T32:    rjmp    LCD_Text32
```

```
T33:    rjmp    LCD_Text33
T34:    rjmp    LCD_Text34
T35:    rjmp    LCD_Text35
T36:    rjmp    LCD_Text36


LCD_Text21:
        ldi     ZL      ,   LOW(Text21*2)   ; "0000"
        ldi     ZH      ,   HIGH(Text21*2)
        rcall   TextOut40
        ldi     R22     ,   1
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text22:
        ldi     ZL      ,   LOW(Text22*2)   ; "0001"
        ldi     ZH      ,   HIGH(Text22*2)
        rcall   TextOut40
        ldi     R22     ,   2
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text23:
        ldi     ZL      ,   LOW(Text23*2)   ; "0010"
        ldi     ZH      ,   HIGH(Text23*2)
        rcall   TextOut40
        ldi     R22     ,   3
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text24:
        ldi     ZL      ,   LOW(Text24*2)   ; "0011"
        ldi     ZH      ,   HIGH(Text24*2)
        rcall   TextOut40
        ldi     R22     ,   4
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text25:
        ldi     ZL      ,   LOW(Text25*2)   ; "0100"
        ldi     ZH      ,   HIGH(Text25*2)
        rcall   TextOut40
        ldi     R22     ,   5
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text26:
        ldi     ZL      ,   LOW(Text26*2)   ; "0101"
        ldi     ZH      ,   HIGH(Text26*2)
        rcall   TextOut40
        ldi     R22     ,   6
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text27:
        ldi     ZL      ,   LOW(Text27*2)   ; "0110"
        ldi     ZH      ,   HIGH(Text27*2)
        rcall   TextOut40
        ldi     R22     ,   7
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text28:
        ldi     ZL      ,   LOW(Text28*2)   ; "0111"
        ldi     ZH      ,   HIGH(Text28*2)
        rcall   TextOut40
        ldi     R22     ,   8
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text29:
        ldi     ZL      ,   LOW(Text29*2)   ; "1000"
        ldi     ZH      ,   HIGH(Text29*2)
        rcall   TextOut40
        ldi     R22     ,   9
        rcall   LCD_Text50
        rjmp    LCD_Text40
LCD_Text30:
        ldi     ZL      ,   LOW(Text30*2)   ; "1001"
        ldi     ZH      ,   HIGH(Text30*2)
        rcall   TextOut40
        ldi     R22     ,   10
        rcall   LCD_Text50
```

```
                rjmp    LCD_Text40
LCD_Text31:
                ldi     ZL      ,   LOW(Text31*2)   ; "1010"
                ldi     ZH      ,   HIGH(Text31*2)
                rcall   TextOut40
                ldi     R22     ,   11
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text32:
                ldi     ZL      ,   LOW(Text32*2)   ; "1011"
                ldi     ZH      ,   HIGH(Text32*2)
                rcall   TextOut40
                ldi     R22     ,   12
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text33:
                ldi     ZL      ,   LOW(Text33*2)   ; "1100"
                ldi     ZH      ,   HIGH(Text33*2)
                rcall   TextOut40
                ldi     R22     ,   13
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text34:
                ldi     ZL      ,   LOW(Text34*2)   ; "1101"
                ldi     ZH      ,   HIGH(Text34*2)
                rcall   TextOut40
                ldi     R22     ,   14
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text35:
                ldi     ZL      ,   LOW(Text35*2)   ; "1110"
                ldi     ZH      ,   HIGH(Text35*2)
                rcall   TextOut40
                ldi     R22     ,   15
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text36:
                ldi     ZL      ,   LOW(Text36*2)   ; "1111"
                ldi     ZH      ,   HIGH(Text36*2)
                rcall   TextOut40
                ldi     R22     ,   16
                rcall   LCD_Text50
                rjmp    LCD_Text40
LCD_Text40:
                ret


LCD_Text50:                                 ; Bar Graph in LCD_line3
                push    R17
                ldi     R17     ,   16
                ldi     R16     ,   32
                rcall   LCD_goto
LCD_Text51:
                ldi     R16     ,   32
                rcall   LCD_data
                dec     R17
                cpi     R17     ,   0
                breq    LCD_Text52
                rjmp    LCD_Text51
LCD_Text52:
                ldi     R16     ,   32
                rcall   LCD_goto
LCD_Text53:
                ldi     R16     ,   5
                rcall   LCD_data
                dec     R22
                cpi     R22     ,   0
                breq    LCD_Text54
                rjmp    LCD_Text53
LCD_Text54:
                pop     R17
                ret


;-------------------------------------------------------------------------------------
; Subroutibes for the Output of Text to the LCD
```

```
TextOut10:
        push    R16
        rcall   LCD_line1
TextOut11:
        lpm     R16     ,   Z+
        cpi     R16     ,   255
        breq    TextOut12
        rcall   LCD_data
        rjmp    TextOut11
TextOut12:
        pop     R16
        ret


TextOut20:
        push    R16
        rcall   LCD_line2
TextOut21:
        lpm     R16     ,   Z+
        cpi     R16     ,   255
        breq    TextOut22
        rcall   LCD_data
        rjmp    TextOut21
TextOut22:
        pop     R16
        ret


TextOut30:
        push    R16
        ldi     R16     ,   26
        rcall   LCD_goto
TextOut31:
        lpm     R16     ,   Z+
        cpi     R16     ,   255
        breq    TextOut32
        rcall   LCD_data
        rjmp    TextOut31
TextOut32:
        pop     R16
        ret


TextOut40:
        push    R16
        ldi     R16     ,   28
        rcall   LCD_goto
TextOut41:
        lpm     R16     ,   Z+
        cpi     R16     ,   255
        breq    TextOut42
        rcall   LCD_data
        rjmp    TextOut41
TextOut42:
        pop     R16
        ret


; ********************************************************************************************
; Begin Initializaton of LCD DOGM162W-A / 5.0 V
; ********************************************************************************************
LCD_init:

;-------------------------------------------------------------------------------------------
; Function Set 01: 3 Mal den 8-Bit Mode ausgeben

        ldi     R16     ,   0b00110000      ; 8 Bit Mode
        swap    R16                         ; 0b00000011 wegen 4 Bit Verdrahtung
                                            ; RS  = Low = Command
                                            ; R/W = Low = Read
        out     PORTD   ,   R16             ; must be set 3 times, why?
;-------------------------------------------------------------------------------------------
; Function Set 01: Das 1. Mal

        rcall   LCD_enable                  ; Impuls von E = MPU D.6 Pin 12

        ldi     R16     ,   1
```

```
        rcall   Wait

;--------------------------------------------------------------------------------
; Function Set 02: Das 2. Mal

        rcall   LCD_enable              ; Impuls von E = MPU D.6 Pin 12

        ldi     R16     ,   1
        rcall   Wait

;--------------------------------------------------------------------------------
; Function Set 03: Das 3. Mal

        rcall   LCD_enable              ; Impuls von E = MPU D.6 Pin 12

        ldi     R16     ,   1
        rcall   Wait

;--------------------------------------------------------------------------------
; Function Set 04: Den 4-Bit Mode einschalten

        ldi     R16     ,   0b00100000  ; 4 Bit Mode, D.4 = Low = Command
        swap    R16                     ; 0b00000010 wegen 4 Bit Verdrahtung
        out     PORTD   ,   R16         ;
        rcall   LCD_enable              ; Impuls von E = MPU D.6 Pin 12

        ldi     R16     ,   1
        rcall   Wait

;--------------------------------------------------------------------------------
; Function Set 05: 4-Bit Mode, 2 Lines, 5x8 Dots, Instruction table 1, CGRam not available

        ldi     R16     ,   0b00101001
        rcall   LCD_cmd

;--------------------------------------------------------------------------------
; Function Set 06: Bias, 2 Lines

        ldi     R16     ,   0b00010100  ; BS: 1/5, 2 Lines
        rcall   LCD_cmd                 ;

;--------------------------------------------------------------------------------
; Function Set 07: Contrast Set (Low Byte)

;       ldi     R16     ,   0b01111000  ; Contrast Set
;       rcall   LCD_cmd

; Contrast Setting by Program
        mov     R16     ,   R20         ; Contrast: Content of R20
        rcall   LCD_cmd

;--------------------------------------------------------------------------------
; Function Set 08: ICON/Power/Contrast (High Byte)

;       ldi     R16     ,   0b01010110  ; Ion: ICON Off
;       rcall   LCD_cmd                 ; Bon: Charge Pump On
                                        ; C5:  Contrast Value On
                                        ; C4:  Contrast Value Off

; Contrast Setting by Program
        mov     R16     ,   R21         ; Contrast: Content of R21
        rcall   LCD_cmd

;--------------------------------------------------------------------------------
; Function Set 09: Follower Control

        ldi     R16     ,   0b01101110  ; Follower CCircuit On
        rcall   LCD_cmd                 ; Follower Amplified Ratio:
                                        ; Rab2=1 Rab1=1 Rab0=0

; Remarks: Fon=0: No Contrast, Fon must be set to 1
;          Rab2=0 Rab1=0 Rab0=0 Contrast: None
;          Rab2=1 Rab1=1 Rab0=1 Contrast: Very High
```

```
;-------------------------------------------------------------------------------------
; Function Set 10: 4-Bit Mode, 2 Lines, 5x8 Dots, Normal Instructions, CGRam available

        ldi     R16     ,   0b00101000
        rcall   LCD_cmd


;-------------------------------------------------------------------------------------
; Function Set 11: Display On

        ldi     R16     ,   0b00001100      ; Display On
        rcall   LCD_cmd                     ;


;-------------------------------------------------------------------------------------
; Function Set 12: Clear Display

        ldi     R16     ,   0b00001101      ; Clear Display
        rcall   LCD_cmd                     ;


;-------------------------------------------------------------------------------------
; Functin Set 13: Entry Mode Set

        ldi     R16     ,   0b00001100      ; Entry Mode Set
        rcall   LCD_cmd


;-------------------------------------------------------------------------------------
; The End

        ret


; ************************************************************************************
; End Initialization of LCD DOGM162W-A / 5.0 V
; ************************************************************************************


;-------------------------------------------------------------------------------------
; CGRAM Initialisation

CGRAM_Init:
        ldi     R16     ,   0b01000000      ; Set CGRAM address in DDRAM
        rcall   LCD_cmd
CGRAM_Init1:
        ldi     ZL      ,   LOW(Symbols*2)
        ldi     ZH      ,   HIGH(Symbols*2)
CGRAM_Init2:
        lpm     R16     ,   Z+
        cpi     R16     ,   255
        breq    CGRAM_Init3
        rcall   LCD_data
        rjmp    CGRAM_Init2
CGRAM_Init3:
        ret


;-------------------------------------------------------------------------------------
; Ausgabe von Daten zur Anzeige im LCD, sendet ein Datenbyte

LCD_data:
        push    R17
        mov     R17     ,   R16             ; "Sicherungskopie" für das 2. Nibble
        swap    R16                         ; Nibbles vertauschen
        andi    R16     ,   0b00001111      ; obereses Nibble auf Null setzen
        sbr     R16     ,   0b00010000      ; RS auf 1
        out     PORTD   ,   R16             ; 1. Nibble plus RS ausgeben
        rcall   LCD_enable                  ; 1. Nibble plus RS übernehmen
                                            ; 2. Nibble kein swap, da es in R17 schon an
                                            ; der richtigen Stelle steht
        andi    R17     ,   0b00001111      ; oberes Nibble auf Null setzen
        sbr     R17     ,   0b00010000      ; RS auf 1
        out     PORTD   ,   R17             ; 2. Nibble plus RS ausgeben
        rcall   LCD_enable                  ; 2. Nibble plus RS übernehmen
        rcall   LCD_busy                    ; Busy Flag prüfen
        pop     R17
        ret


;-------------------------------------------------------------------------------------
; Ausgabe von Kommandos ans LCD, wie LCD_data aber RS = 0
```

```
LCD_cmd:
        push    R17
        mov     R17     ,   R16             ; "Sicherungskopie" für das 2. Nibble
        swap    R16                         ; Nibbles vertauschen
        andi    R16     ,   0b00001111      ; obereses Nibble auf Null setzen
        sbr     R16     ,   0b00000000      ; RS auf 0 (bzw. nicht auf 1)
        out     PORTD   ,   R16             ; 1. Nibble ausgeben
        rcall   LCD_enable                  ; 1. Nibble übernehmen
                                            ; 2. Nibble kein swap, da es in R17 schon an
                                            ; der richtigen Stelle steht
        andi    R17     ,   0b00001111      ; oberes Nibble auf Null setzen
        sbr     R17     ,   0b00000000      ; RS auf 0 (bzw. nicht auf 1)
        out     PORTD   ,   R17             ; 2. Nibble ausgeben
        rcall   LCD_enable                  ; 2. Nibble übernehmen
        rcall   LCD_busy                    ; Busy Flag prüfen
        pop     R17
        ret

;------------------------------------------------------------------------------------
LCD_enable:
        sbi     PORTD   ,   6               ; Enable High
        nop                                 ; kurz warten
        nop
        nop
        cbi     PORTD   ,   6               ; Enable Low
        ret

;------------------------------------------------------------------------------------
; Busy Flag prüfen

LCD_busy:
        push    R16
        ldi     R16     ,   0b11110000      ; Disable Data Bit Outputs
        out     DDRD    ,   R16
        ldi     R16     ,   0b00000000      ; Clear all outputs
        out     PORTD   ,   R16
LCD_busy1:
        ldi     R16     ,   0b00100000      ; Enable only read bit
        out     PORTD   ,   R16
        sbi     PORTD   ,   6               ; Raise the Enable signal
        nop                                 ; kurz warten
        nop
        in      R16     ,   PIND            ; Read the current values
        cbi     PORTD   ,   6               ; Disable the Enable signal
        rcall   LCD_enable                  ; Puls the Enable (the second nibble is discarded)
        swap    R16                         ; Busy flag von R16.3 nach R16.7
        sbrc    R16     ,   7               ; Check busy flag
        rjmp    LCD_busy1                   ;
        ldi     R16     ,   0b11111111      ; Enable all outputs
        out     DDRD    ,   R16
        pop     R16
        ret

;------------------------------------------------------------------------------------
LCD_clear:
        push    R16
        ldi     R16     ,   0b00000001      ; Display Clear
        rcall   LCD_cmd
        pop     R16
        ret

;------------------------------------------------------------------------------------
LCD_home:
        push    R16
        ldi     R16     ,   0b00000010      ; Display Cursor HOME
        rcall   LCD_cmd
        pop     R16
        ret

;------------------------------------------------------------------------------------
LCD_off:
        push    R16
        ldi     R16     ,   0b00001000      ; LCD OFF
```

```asm
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_on:
        push    R16
        ldi     R16     ,   0b00001100      ; LCD On, Cursor Off, Blink Off
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_CurOn:
        push    R16
        ldi     R16     ,   0b00001110      ; LCD On, Cursor On, Blink Off
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_BlkOn:
        push    R16
        ldi     R16     ,   0b00001101      ; LCD On, Cursor Off, Blink On
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_CurBlkOn:
        push    R16
        ldi     R16     ,   0b00001111      ; LCD On, Cursor On, Blink On
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
; LCD_line1:  0
; LCD_line2: 16
; LCD_line3: 32

LCD_line1:
        push    R16
        ldi     R16     ,   0b10000000      ; DDRAM Address 0x00 = 0
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_line2:
        push    R16
        ldi     R16     ,   0b10010000      ; DDRAM Address 0x10 = 16
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
LCD_line3:
        push    R16
        ldi     R16     ,   0b10100000      ; DDRAM Address 0x20 = 32
        rcall   LCD_cmd
        pop     R16
        ret

;--------------------------------------------------------------------------------------------
; Goto R16 = Addresses in LCD_line1:  0 ... 15
;                          LCD_line2: 16 ... 31
;                          LCD_line3: 32 ... 47

LCD_goto:
        push    R16
        ori     R16     ,   0b10000000      ; Goto DDRAM Address R16
        rcall   LCD_cmd
        pop     R16
```

```
                  ret

;--------------------------------------------------------------------------------------------
LCD_CUL:
          push    R16
          ldi     R16     ,   0b00010000      ; Cursor one position left
          rcall   LCD_cmd
          pop     R16
          ret

;--------------------------------------------------------------------------------------------
LCD_CUR:
          push    R16
          ldi     R16     ,   0b00010100      ; Cursor one position right
          rcall   LCD_cmd
          pop     R16
          ret

;--------------------------------------------------------------------------------------------
; Waiting Time: R16 = 1 ==> 0.01 s, R16 = 255 ==> 2.55 s at 3.686 MHz

Wait:
          push    R16
          push    R17
          push    R18

          cpi     R16     ,   0
          breq    WLoop0
WLoop1:
          ldi     R17     ,   0b01101110
WLoop2:
          ldi     R18     ,   0b01101110
WLoop3:
          dec     R18
          brne    WLoop3
          nop
          nop
          dec     R17
          brne    WLoop2
          dec     R16
          brne    WLoop1
WLoop0:
          pop     R18
          pop     R17
          pop     R16
          ret

;--------------------------------------------------------------------------------------------
; Text

Text00: .db     " CONTRAST 543210",255,0

Text10: .db     "Setting C ",255,0

Text11: .db     "00      ",255,0
Text12: .db     "01      ",255,0
Text13: .db     "10      ",255,0
Text14: .db     "11      ",255,0

Text21: .db     "0000",255,0
Text22: .db     "0001",255,0
Text23: .db     "0010",255,0
Text24: .db     "0011",255,0
Text25: .db     "0100",255,0
Text26: .db     "0101",255,0
Text27: .db     "0110",255,0
Text28: .db     "0111",255,0
Text29: .db     "1000",255,0
Text30: .db     "1001",255,0
Text31: .db     "1010",255,0
Text32: .db     "1011",255,0
Text33: .db     "1100",255,0
Text34: .db     "1101",255,0
Text35: .db     "1110",255,0
```

```
Text36: .db    "1111",255,0


;-------------------------------------------------------------------------------
; Symbols for the CGRAM

Symbols:
        .db     0,0,0,0,0,0,0,0
        .db     0,0,0,16,16,16,0,0
        .db     0,0,0,16,16,16,0,0
        .db     0,0,0,20,20,20,0,0
        .db     0,0,0,20,20,20,0,0
        .db     0,0,0,21,21,21,0,0,255,0


;-------------------------------------------------------------------------------
```