```
;-------------------------------------------------------------------------------
; Titel         :   DOGM162W-A
;               :
; Functions     :   Line 1:
;               :   LCD-Output of a Voltage in 4 Measuring Ranges, each with 10 Bits
;               :     0.00000 until   0.25550 {V}   Resolution: 0.00025 [V] DC or AC
;               :     0.0000  until   2.5550  (V)   Resolution: 0.0025  [V] DC or AC
;               :     0.000   until  25.550   (V)   Resolution: 0.025   [V] DC or AC
;               :     0.00    until 255.50    (V)   Resolution: 0.25    [V] DC or AC
;               :
;               :   To be selected with the white push button:
;               :   Line 2:, after one Second   Line 2
;               :    BAR GRAPH TYPE 0           Empty
;               ;    BAR GRAPH TYPE 1           "Quasianalogue" with 0 until 80 Bars
;               ;    BAR GRAPH TYPE 2           "Quasianalogue" with 0 until 48 Bars
;               :    CALIBRATION MODE           " CALIBR. R24=123"
;               :
;               :   To be selected with the green push button:
;               :    DC Range: AUTO     automatically after "POWER ON"
;               :    DC Range:   0.25 V
;               :    DC Range:   2.5 V
;               :    DC Range:  25 V
;               :    DC Range: 250 V
;               :    AC Range: AUTO
;               :    AC Range:   0.25 V
;               :    AC Range:   2.5 V
;               :    AC Range:  25 V
;               :    AC Range: 250 V
;               :
; Diagrams      :   Entire Diagram
;               :   Upper Circuit Board
;               :   Lower Circuit Board
;               :
;-------------------------------------------------------------------------------
; Register      :   No      used        local       | Function 1    | Function 2        | Function 3
;               :           Y   N       Y   N
;               :
;               :   R0      X               X       | ADC Low       | Multiplication    | Calculation a
;               :   R1      X               X       | ADC High      | of 2 Values,      | Decimal Number
;               :   R2      X               X                       | each 16 Bit       | from a 16 Bit
;               :   R3      X               X                       |                   | Binary Number
;               :   R4      X               X                       |                   |
;               :   R5      X               X                       |                   |
;               :   R6      X               X                                         |
;               :   R7      X               X       | Measuring Range (1)
;               :   R8      X               X       | Measuring Range (2)
;               :   R9      X           X                           | Multiplier
;               :   R10     X           X                           | Multiplier
;               :   R11     X           X           | R11=1: Battery Low
;               :   R12         X
;               :   R13         X
;               :   R14         X
;               :   R15         X
;               :   R16     X               X       | diverse
;               :   R17     X           X           | diverse       | ADC Value Correction
;               :   R18     X           X           | diverse       | ADC Value Correction
;               :   R19     X           X           | diverse       | Fields in LCD_line2
;               :   R20     X           X                           | Lines  in LCD_line2
;               :   R21     X           X                           | Blanks in LCD_line2
;               :   R22     X               X       | ADC Low
;               :   R23     X               X       | ADC High
;               :   R24     X               X       | 0...80 Bars in LCD_line2
;               :   R25     X               X       | ADC Value Correction Low
;               :   R26     X               X       | ADC Value Correction High
;               :   R27     X               X       | Mode of Display in LCD_line2
;               :   R28         X
;               :   R29         X
;               :   R30     X                       | ZL
;               :   R31     X                       | ZH
;-------------------------------------------------------------------------------
; Processor     :   ATmega8A-PU
; Takt (Quarz)  :   4.096 MHz
; Language      :   Assembler Studio 4
; Date          :   20.08.2014
; Version       :   1.0
; Autor         :   KlausD
;-------------------------------------------------------------------------------
        .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
        rjmp    onTMR0                  ; 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
        rjmp    onADC                   ; 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, Ports, Interrupt

Main:
        ldi     R16     ,   LOW (RAMEND)    ; for  Stackpointer LOW
        out     SPL     ,   R16             ; INIT Stackpointer LOW

        ldi     R16     ,   HIGH(RAMEND)    ; for  Stackpointer HIGH
        out     SPH     ,   R16             ; INIT Stackpointer HIGH

        ldi     R16     ,   0b00111111      ; PORTB Bits 0 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     ,   0b00001111      ; PORTC Bits 0 until 3 = 1 (Pullup)
        out     PORTC   ,   R16             ; PORTC Bit  4 = ADC Input for DC
                                            ; PORTC Bit  5 = ADC Input for AC
                                            ; Pullup would adulerate the Measuring Value !

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

;-------------------------------------------------------------------------------
; Timer/Counter2
; B.3 shall deliver a 64 kHz rectangle signal 1:1. This serves as an input signal for a
; charge pump which creates a voltage of -5 [VDC]. The ICs LM324 and CD4066 need the
; supply voltages +5 [VDC] and -5 [VDC]. With the charge pump, the DVM can work with
; a single voltage, i.e. an accumulator.
;
;       TCCR2 - Timer/Counter2 Control Register ..... ATMEGA8A Data Sheet P. 112
;       +-----------------------------------------------------------------------+
;       |   FOC2  |  WGM20  |  COM21  |  COM20  |  WGM21  |   CS22  |   CS21  |   CS20  |
;       +-----------------------------------------------------------------------+
        ; FOC2  = 0 (for future use)
        ; WGM21 WGM20
        ;   0     0     Normal, TOV2 Flag Set MAX
        ;   0     1     PWM, Phase Correct
        ;   1     0     CTC, Top OCR2
        ;   1     1     Fast PWM, Top 0xFF
        ;
        ; COM21 COM20   Non  PWM Mode
        ;   0     0     Normal Port Operation
        ;   0     1     Toggle B.3 Pin17 On Compare Match
        ;   1     0     Clear  B.3 Pin17 On Compare Match
        ;   1     1     Set    B.3 Pin17 On Compare Match
        ;
        ;                 Fast PWM Mode
        ;   0     0     Normal Port Operation
        ;   0     1     Reserved
        ;   1     0     Clear  B.3 Pin17 On Compare Match
        ;   1     1     Set    B.3 Pin17 On Compare Match
        ;
        ; CS22 CS21 CS20
        ;   0    0    0 Timer Stopped
        ;   0    0    1 Prescaling    1
        ;   0    1    0 Prescaling    8
        ;   0    1    1 Prescaling   32
        ;   1    0    0 Prescaling   64
        ;   1    0    1 Prescaling  128
        ;   1    1    0 Prescaling 1024

        ldi     R16     ,   (1<<COM20) | (1<<WGM21) | (1<<CS20)
        out     TCCR2   ,   R16             ; COM20 = Toggle B.3
                                            ; WGM21 = CTC, TOP OCR2
                                            ; CS20  = Prescaling 1

;       TCNT2 - Timer/Counter2 Register ..... ATMEGA8A Data Sheet P. 113
;       +-----------------------------------------------------------------------+
```

```
;       |        |        |        |        |        |        |        |
;       +-----------------------------------------------------------------+
;       The Content is the actual counting Status


;       OCR2  - Output Compare Register ..................... ATMEGA8A Data Sheet P. 114
;       +-----------------------------------------------------------------+
;       |        |        |        |        |        |        |        |
;       +-----------------------------------------------------------------+
;       The content is continuously compared with the content of the TCNT2
        ; The content can be used to generate an
        ; Output Compare Interrupt
        ; or to generate a waveform output on B.3 Pin17

        ldi     R16     ,   0b00011111      ; 4096000 / 32 = 128000 Hz (B.3 = 64000 Hz)
        out     OCR2    ,   R16


;       ASSR  - Asynchronus Status Register ................. ATMEGA8A Data Sheet P. 114
;       +-----------------------------------------------------------------+
;       |        |        |        |   AS2  | TCN2UB | OCR2UB | TCR2UB |
;       +-----------------------------------------------------------------+
        ; AS2    - Asynchronus Timer/Counter2
        ; TCN2UB - Timer/CounterUpdate Busy
        ; OCR2UB - Output Compare Register2 Update Busy
        ; TCR2UB - Timer/Counter Control Register2 Update Busy


;       TIMSK - Timer/Counter Interrupt Mask Register ........ ATMEGA8A Data Sheet P. 115
;       +-----------------------------------------------------------------+
;       | OCIE2  |  TOIE2  |  TOIE1  | OCIE1A | OCIE1B |  TOIE1  |        |  TOIE0  |
;       +-----------------------------------------------------------------+
        ; OCIE2  - Timer/counter2 Output Compare Match Interrupt Enable
        ; TOIE2  - Timer/Counter2 Overflow Interrupt Enable

        ldi     R16     ,   (1<<OCIE2)      ; Timer/counter2 Output Compare
        out     TIMSK   ,   R16             ; Match Interrupt Enable


;       TIFR  - Timer/Counter Interrupt Flag Register ........ ATMEGA8A Data Sheet P. 115
;       +-----------------------------------------------------------------+
;       |   OCF2  |   TOV2  |   ICF1  |  OCF1A  |  OCF1B  |   TOV1  |        |   TOV0  |
;       +-----------------------------------------------------------------+
        ; OCF2   - Output Compare Flag 2
        ; TOV2   - Timer/Counter2 Overflow Flag 2


;       SFIOR  - Special Function IO Register ................ ATMEGA8A Data Sheet P. 115
;       +-----------------------------------------------------------------+
;       |        |        |        |   ACME  |   PUD   |   PSR2  |  PSR10  |
;       +-----------------------------------------------------------------+
        ; PSR2   - Prescaler Reset Timer/Counter2

;-------------------------------------------------------------------------------
; ADC
;
;       ADMUX - ADC Multiplexer Selection Register ........... ATMEGA8A Data Sheet P. 190
;       +-----------------------------------------------------------------+
;       |  REFS1  |  REFS0  |  ADLAR  |        |   MUX3  |   MUX2  |   MUX1  |   MUX0  |
;       +-----------------------------------------------------------------+
        ; REFS1 REFS0
        ;   0     0   Reference External
        ;   0     1   Reference Internal 5 V (AVCC)
        ;   1     0   Reserved
        ;   1     1   Reference Internal 2.56 V
        ;
        ; ADLAR
        ;   0         Result Right Adjusted
        ;   1         Result Left  Adjusted

;       ADLAR = 0
;       +-----------------------------------------------------------------+
;       |        |        |        |        |        |   ADC9  |   ADC8  | ADCH
;       +-----------------------------------------------------------------+
;       |   ADC7  |   ADC6  |   ADC5  |   ADC4  |   ADC3  |   ADC2  |   ADC1  |   ADC0  | ADCL
;       +-----------------------------------------------------------------+

;       ADLAR = 1
;       +-----------------------------------------------------------------+
;       |   ADC7  |   ADC6  |   ADC7  |   ADC6  |   ADC5  |   ADC4  |   ADC3  |   ADC2  | ADCH
;       +-----------------------------------------------------------------+
;       |   ADC1  |   ADC0  |        |        |        |        |        |        | ADCL
;       +-----------------------------------------------------------------+

        ; MUX3  MUX2  MUX1  MUX0
        ;   0     0     0     0  Channel 0, PC0 Pin 23
        ;   0     0     0     1  Channel 1, PC1 Pin 24
```

```
        ;   0     0     1     0  Channel 2, PC2 Pin 25
        ;   0     0     1     1  Channel 3, PC3 Pin 26
        ;   0     1     0     0  Channel 4, PC4 Pin 27
        ;   0     1     0     1  Channel 5, PC5 Pin 28
        ;   0     1     1     0  Channel 6  (not in PDIP case)
        ;   0     1     1     1  Channel 7  (not in PDIP case)

;       ldi     R16     ,   (1<<MUX2) | (1<<MUX0)
;       out     ADMUX   ,   R16             ; Ref. extern
                                            ; Result rightbound
                                            ; Channel C.5


;       ADCSRA - ADC Control And Status Register A ........... ATMEGA8A Data Sheet P. 191
;       +-------------------------------------------------------------------------+
;       |   ADEN  |   ADSC  |   ADFR  |   ADIF  |   ADIE  |  ADPS2  |  ADPS1  |  ADPS0  |
;       +-------------------------------------------------------------------------+
        ; ADEN = ADC Enable
        ; ADSC = ADC Start Conversion
        ; ADFR = ADC Free Running
        ; ADIF = ADC Interrupt Flag
        ; ADTE = ADC Interrupt Enable
        ;
        ; ADPS2 ADPS1 ADPS0 Prescale
        ;   0     0     0       2
        ;   0     0     1       2
        ;   0     1     0       4
        ;   0     1     1       8
        ;   1     0     0      16
        ;   1     0     1      32
        ;   1     1     0      64
        ;   1     1     1     128

        ldi     R16     ,   (1<<ADEN) | (1<<ADIF) | (1<<ADIE) | (1<<ADPS2) | (1<<ADPS0)
        out     ADCSRA  ,   R16             ; ADC Enable
                                            ; ADC Interrupt Flag
                                            ; ADC Interrupt Enable
                                            ; ADC Prescale 32

;-------------------------------------------------------------------------------------
; 8-Bit Timer/Counter0

;       TCCR0 - Timer/Counter0 Control Register ............... ATMEGA8A Data Sheet P. 69
;       +-------------------------------------------------------------------------+
;       |         |         |         |         |         |   CS02  |   CS01  |   CS00  |
;       +-------------------------------------------------------------------------+
        ; CSnn = Clock Select
        ;
        ; CS02  CS01  CS00  Description
        ;   0     0     0   Timer/Counter stopped
        ;   0     0     1   Prescale      1
        ;   0     1     0   Prescale      8
        ;   0     1     1   Precsale     64
        ;   1     0     0   Prescale    256
        ;   1     0     1   Prescale   1024
        ;   1     1     0   External Clock at T0 Pin = D.4 = Pin 6, falling edge
        ;   1     1     1   External Clock at T0 Pin = D.4 = Pin 6, rising  edge

        ldi     R16     ,   (1<<CS02) | (1<<CS00)
        out     TCCR0   ,   R16             ; Prescale 1024


;       TCNT0 - Timer/Counter0 Register ....................... ATMEGA8A Data Sheet P. 69
;       +-------------------------------------------------------------------------+
;       |         |         |         |         |         |         |         |         |
;       +-------------------------------------------------------------------------+
        ; Actual Content of the Timer/Counter0 8-Bit Counter

        ldi     R16     ,   131             ; count upwords 131 bis 256 = Overflow
        out     TCNT0   ,   R16             ; these are 125 Increments

        ;       f_Quarz                                         4096000
        ; f = ------------------------------------------ = --------------------- = 32
        ;       Prescaler * (2 ^ Bit - TCNT0_Initial Value)  1024 * (2 ^ 8 - 131)
        ;
        ; Control of B.5 Input 32 times per second wheather the level changed


;       TIMSK - Timer/Counter Interrupt Mask Register ........ ATMEGA8A Data Sheet P. 69
;       +-------------------------------------------------------------------------+
;       |  OCIE2  |  TOIE2  |  TICIE1 |  OCIE1A |  OCIE1B |  TOIE1  |         |  TOIE0  |
;       +-------------------------------------------------------------------------+
        ; TOIE0 = Timer/Counter0 Overflow Interrupt Enable
        ;
        ; TOIE0 = 0: Overflow Interrupt Disabled
        ; TOIE0 = 1: Overflow Interrupt Enabled
```

```
;       ldi     R16     ,   0b00000001
;       out     TIMSK   ,   R16             ; Overflow Interrupt Enabled

        ldi     R16     ,   (1<<OCIE2) | (1<<TOIE0)
        out     TIMSK   ,   R16             ; OCIE2 = Timer/Counter2 Output Compare
                                            ;         Match Interrupt Enable
                                            ; TOIE0 = Timer/Counter0 Overflow Interrupt
                                            ;         Enable


;       TIFR - Timer/Counter0 Interrupt Flag Register ......... ATMEGA8A Data Sheet P. 70
;       +-------------------------------------------------------------------------------+
;       |   OCF2  |   TOV2  |   ICF1  |  OCF1A  |  OCF1B  |   TOV1  |         |   TOV0  |
;       +-------------------------------------------------------------------------------+
        ; TOV0 = Timer/Counter0 Overflow Flag
        ;
        ; TOV0 = 1 after the overflow
        ; TOV0 = 0 when the interrupt handler is being executed


;       SFIOR - Special Function IO Register .................. ATMEGA8A Data Sheet P. 72
;       +-------------------------------------------------------------------------------+
;       |         |         |         |         |   ACME  |    PUD  |   PSR2  |  PSR10  |
;       +-------------------------------------------------------------------------------+
        ; PSR10 = Prescaler Reset (Timer/Counter0 and Timer/Counter1)
        ;
        ; PSR10 = 1 (by Program)  Prescaler Reset
        ; PSR10 = 0 (by Hardware) after the operation has been performed

;-------------------------------------------------------------------------------------

        ldi     R16     ,   0b00010000
        mov     R8      ,   R16
        out     PORTB   ,   R16             ; Swith on Measuring Range 0.25 V
                                            ; B.0 =  1 = 0b00000001: MR 250.00    V
                                            ; B.1 =  2 = 0b00000010: MR  25.000   V
                                            ; B.2 =  4 = 0b00000100: MR   2.5000  V
                                            ; B.4 = 16 = 0b00010000: MR   0.25000 V
                                            ; B.3 = 64 KHz for the charge pump
                                            ;          whitch creates - 5 VDC
                                            ; B.5 = Push Button

        ldi     R16     ,   1               ; MR = DC AUTO  (selection with Push Button)
        mov     R7      ,   R16
        ldi     R27     ,   1               ; R27 = 1: Bar Graph Type 1 (all bars)
                                            ; R27 = 2: Bar Graph Type 2 (every second bar)
                                            ; R27 = 3: Calibration Mode
        rcall   LCD_init
        rcall   LCD_clear
        rcall   CGRam
        rjmp    Loop010                     ; First check of the Measuring Range

;-------------------------------------------------------------------------------------
; Main Loop

Loop:
        sei                                 ; Allow Interrupt

        mov     R16     ,   R7              ; Selection of the ADC Channel
        cpi     R16     ,   1               ; R7 = 1, 2, 3, 4,  5 Channel 5 Pin 28
        breq    A1                          ; R7 = 6, 7, 8, 9, 10 Channel 4 Pin 27
        cpi     R16     ,   2
        breq    A1                          ;  R7
        cpi     R16     ,   3               ;
        breq    A1                          ;   1   DC AUTO
        cpi     R16     ,   4               ;   2   DC 0.25 V
        breq    A1                          ;   3   DC 2.50 V
        cpi     R16     ,   5               ;   4   DC 25.0 V
        breq    A1                          ;   5   DC 250  V
        cpi     R16     ,   6               ;
        breq    A2                          ;   6   AC AUTO
        cpi     R16     ,   7               ;   7   AC 0.25 V
        breq    A2                          ;   8   AC 2.50 V
        cpi     R16     ,   8               ;   9   AC 25.0 V
        breq    A2                          ;  10   AC 250  V
        cpi     R16     ,   9
        breq    A2
        cpi     R16     ,   10
        breq    A2
A1:                                         ; ADC Channel C.4 Pin 27
        ldi     R16     ,   (1<<MUX2)
        out     ADMUX   ,   R16
        rjmp    A3
A2:                                         ; ADC Channel C.5 Pin 28
        ldi     R16     ,   (1<<MUX2) | (1<<MUX0)
        out     ADMUX   ,   R16
```

```
        rjmp    A3
A3:
        sbi     ADCSRA  ,   ADSC           ; next ADC action
waitADC:
        sbic    ADCSRA  ,   ADSC           ; wait until ADC action ready
        rjmp    waitADC

;-------------------------------------------------------------------------------------
; Determing Measuring Range in AUTO Mode

        push    R16
        mov     R16     ,   R7
        cpi     R16     ,   1              ; Range = DC AUTO
        breq    Loop201
        cpi     R16     ,   2              ; Range = DC 0.25 V
        breq    Loop202
        cpi     R16     ,   3              ; Range = DC 2.55 V
        breq    Loop203
        cpi     R16     ,   4              ; Range = DC 25.5 V
        breq    Loop204
        cpi     R16     ,   5              ; Range = DC 255  V
        breq    Loop205
        cpi     R16     ,   6              ; Range = AC AUTO
        breq    Loop206
        cpi     R16     ,   7              ; Range = AC 0.25 V
        breq    Loop207
        cpi     R16     ,   8              ; Range = AC 2.55 V
        breq    Loop208
        cpi     R16     ,   9              ; Range = AC 25.5 V
        breq    Loop209
        cpi     R16     ,   10             ; Range = AC 255  V
        breq    Loop210
Loop201:
        rjmp    Loop101                    ; MR AUTO
Loop202:
        ldi     R16     ,   16             ; MR 0.25 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop203:
        ldi     R16     ,   4              ; MR 2.55 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop204:
        ldi     R16     ,   2              ; MR 25.5 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop205:
        ldi     R16     ,   1              ; MR 255 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop206:
        rjmp    Loop101                    ; MR AUTO
Loop207:
        ldi     R16     ,   16             ; MR 0.25 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop208:
        ldi     R16     ,   4              ; MR 2.55 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop209:
        ldi     R16     ,   2              ; MR 25.5 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop210:
        ldi     R16     ,   1              ; MR 255 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        pop     R16
        rjmp    Loop100
Loop101:
        mov     R16     ,   R8             ;
        cpi     R16     ,   1              ;  MR 250 V already on
```

```
        breq    Loop104
        cpi     R23     ,   0b00000011      ; R23 = ADCH
        breq    Loop102
        rjmp    Loop104
Loop102:
        cpi     R22     ,   0b11111111      ; R22 = ADCL
        breq    Loop010
        rjmp    Loop104
Loop104:
        mov     R16     ,   R8              ;
        cpi     R16     ,   16              ; MR 0.25 V already on
        breq    Loop107                 ;
        cpi     R23     ,   0b00000000
        breq    Loop105
        rjmp    Loop107
Loop105:
        cpi     R22     ,   0b01000000
        brlo    Loop010
        rjmp    Loop107
Loop107:
        pop     R16
        rjmp    Loop100

; Maximum in MR 0.25 V ?
Loop010:
        ldi     R16     ,   16              ; MR 0.25 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        ldi     R16     ,   30
        rcall   wait
        sbi     ADCSRA  ,   ADSC            ; next ADC conversion
Loop011:
        sbic    ADCSRA  ,   ADSC            ; wait until done
        rjmp    Loop011
Loop012:
        cpi     R23     ,   0b00000011
        breq    Loop013
        rjmp    Loop050
Loop013:
        cpi     R22     ,   0b11111111
        breq    Loop020
        rjmp    Loop050

; Maximum in MR 2.5 V ?
Loop020:
        ldi     R16     ,   4               ; MR 2.5 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        ldi     R16     ,   30
        rcall   wait
        sbi     ADCSRA  ,   ADSC            ; next ADC conversion
Loop021:
        sbic    ADCSRA  ,   ADSC            ; wait until done
        rjmp    Loop021
Loop022:
        cpi     R23     ,   0b00000011
        breq    Loop023
        rjmp    Loop050
Loop023:
        cpi     R22     ,   0b11111111
        breq    Loop030
        rjmp    Loop050

; Maximum in MR 25 V ?
Loop030:
        ldi     R16     ,   2               ; MR 25 V
        mov     R8      ,   R16
        out     PORTB   ,   R16
        ldi     R16     ,   30
        rcall   wait
        sbi     ADCSRA  ,   ADSC            ; next ADC conversion
Loop031:
        sbic    ADCSRA  ,   ADSC            ; wait until done
        rjmp    Loop031
Loop032:
        cpi     R23     ,   0b00000011
        breq    Loop033
        rjmp    Loop050
Loop033:
        cpi     R22     ,   0b11111111
        breq    Loop040
        rjmp    Loop050

; switch on MR 250 V because the lower MRs show maximum
Loop040:
        ldi     R16     ,   1               ; MR 250 V
        mov     R8      ,   R16
```

```
        out     PORTB   ,   R16
        ldi     R16     ,   30
        rcall   wait
        sbi     ADCSRA  ,   ADSC            ; next ADC conversion
Loop041:
        sbic    ADCSRA  ,   ADSC            ; wait until done
        rjmp    Loop041

; finish the search for the Maximum
Loop050:
        rjmp    Loop

;-----------------------------------------------------------------
; Scanning frequency of the analogue signal
; Influence of different values for Wait in LCD_data
; 'wait5ms' : Scanning Frequency =  4.76 Hz
; 'wait01ms': Scanning Frequency = 60.36 Hz
; 'LDC_busy': Scanning Frequency ~ 62.4  Hz
; B.5 = Scanning Frequency / 2

Loop100:
        mov     R16     ,   R1
        cpi     R16     ,   0
        breq    Loop0
        cpi     R16     ,   1
        breq    Loop1
Loop0:
        ldi     R16     ,   1
        mov     R1      ,   R16
        sbi     PORTB   ,   5
        rjmp    Loop10
Loop1:
        ldi     R16     ,   0
        mov     R1      ,   R16
        cbi     PORTB   ,   5
        rjmp    Loop10

;-------------------------------------------------------------------------------------
; R24 - Code
; R23 and R22 come from onADC. R23 = High, R22 = Low of the 10-Bit ADC
; R23 values range from 0 until   3
; R22 values range from 0 until 255
; R22 is being devided into 20 steps with similar distances as good as possible
;
; Step      from    until   Diff            from [VDC]                  until [VDC]
;
;    1        1       12      11*     0.0025 + R23 * 0.6400      0.0300 + R23 * 0.6400
;    2       13       25      12      0.0325 + R23 * 0.6400      0.0625 + R23 * 0.6400
;    3       26       38      12      0.0650 + R23 * 0.6400      0.0950 + R23 * 0.6400
;    4       39       51      12      0.0975 + R23 * 0.6400      0.1275 + R23 * 0.6400
;    5       52       63      11*     0.1300 + R23 * 0.6400      0.1575 + R23 * 0.6400
;    6       64       76      12      0.1600 + R23 * 0.6400      0.1900 + R23 * 0.6400
;    7       77       89      12      0.1925 + R23 * 0.6400      0.2225 + R23 * 0.6400
;    8       90      102      12      0.2250 + R23 * 0.6400      0.2550 + R23 * 0.6400
;    9      103      115      12      0.2575 + R23 * 0.6400      0.2875 + R23 * 0.6400
;   10      116      127      11*     0.2900 + R23 * 0.6400      0.3175 + R23 * 0.6400
;   11      128      140      12      0.3200 + R23 * 0.6400      0.3500 + R23 * 0.6400
;   12      141      153      12      0.3525 + R23 * 0.6400      0.3825 + R23 * 0.6400
;   13      154      166      12      0.3850 + R23 * 0.6400      0.4150 + R23 * 0.6400
;   14      167      179      12      0.4175 + R23 * 0.6400      0.4475 + R23 * 0.6400
;   15      180      191      11*     0.4500 + R23 * 0.6400      0.4775 + R23 * 0.6400
;   16      192      204      12      0.4800 + R23 * 0.6400      0.5100 + R23 * 0.6400
;   17      205      217      12      0.5125 + R23 * 0.6400      0.5425 + R23 * 0.6400
;   18      218      230      12      0.5450 + R23 * 0.6400      0.5750 + R23 * 0.6400
;   19      231      243      12      0.5775 + R23 * 0.6400      0.6075 + R23 * 0.6400
;   20      244      255      11*     0.6100 + R23 * 0.6400      0.6375 + R23 * 0.6400
;
;   R24 = Step + 20 * R23
;
;   R24 get the values 1 until 80 for 0 until 80 bars in LCD_line2
;
;-------------------------------------------------------------------------------------
Loop10:                                 ; R22 sort out into the steps 1 until 20
        cpi     R22     ,   244         ; R22 from onADC
        brsh    Loop30
        cpi     R22     ,   231
        brsh    Loop29
        cpi     R22     ,   218
        brsh    Loop28
        cpi     R22     ,   205
        brsh    Loop27
        cpi     R22     ,   192
        brsh    Loop26
        cpi     R22     ,   180
        brsh    Loop25
        cpi     R22     ,   167
        brsh    Loop24
```

```
        cpi     R22     ,   154
        brsh    Loop23
        cpi     R22     ,   141
        brsh    Loop22
        cpi     R22     ,   128
        brsh    Loop21
        cpi     R22     ,   116
        brsh    Loop20
        cpi     R22     ,   103
        brsh    Loop19
        cpi     R22     ,   90
        brsh    Loop18
        cpi     R22     ,   77
        brsh    Loop17
        cpi     R22     ,   64
        brsh    Loop16
        cpi     R22     ,   52
        brsh    Loop15
        cpi     R22     ,   39
        brsh    Loop14
        cpi     R22     ,   26
        brsh    Loop13
        cpi     R22     ,   13
        brsh    Loop12
        cpi     R22     ,   0
        brsh    Loop11
Loop30:
        ldi     R24     ,   20
        rjmp    Loop40
Loop29:
        ldi     R24     ,   19
        rjmp    Loop40
Loop28:
        ldi     R24     ,   18
        rjmp    Loop40
Loop27:
        ldi     R24     ,   17
        rjmp    Loop40
Loop26:
        ldi     R24     ,   16
        rjmp    Loop40
Loop25:
        ldi     R24     ,   15
        rjmp    Loop40
Loop24:
        ldi     R24     ,   14
        rjmp    Loop40
Loop23:
        ldi     R24     ,   13
        rjmp    Loop40
Loop22:
        ldi     R24     ,   12
        rjmp    Loop40
Loop21:
        ldi     R24     ,   11
        rjmp    Loop40
Loop20:
        ldi     R24     ,   10
        rjmp    Loop40
Loop19:
        ldi     R24     ,   9
        rjmp    Loop40
Loop18:
        ldi     R24     ,   8
        rjmp    Loop40
Loop17:
        ldi     R24     ,   7
        rjmp    Loop40
Loop16:
        ldi     R24     ,   6
        rjmp    Loop40
Loop15:
        ldi     R24     ,   5
        rjmp    Loop40
Loop14:
        ldi     R24     ,   4
        rjmp    Loop40
Loop13:
        ldi     R24     ,   3
        rjmp    Loop40
Loop12:
        ldi     R24     ,   2
        rjmp    Loop40
Loop11:
        ldi     R24     ,   1
        rjmp    Loop40
```

```
Loop40:                                     ; R24 = 2 * R23 + R24
        push    R16                         ; is being popped in Loop41, Loop42, Loop43 and Loop44
        cpi     R23     ,   0               ; R23 from onADC
        breq    Loop41
        cpi     R23     ,   1
        breq    Loop42
        cpi     R23     ,   2
        breq    Loop43
        cpi     R23     ,   3
        breq    Loop44
Loop41:
        ldi     R16     ,   0
        add     R24     ,   R16
        pop     R16
        rjmp    Loop60
Loop42:
        ldi     R16     ,   20
        add     R24     ,   R16
        pop     R16
        rjmp    Loop60
Loop43:
        ldi     R16     ,   40
        add     R24     ,   R16
        pop     R16
        rjmp    Loop60
Loop44:
        ldi     R16     ,   60
        add     R24     ,   R16
        pop     R16
        rjmp    Loop60

;-----------------------------------------------------------------------------------------
; LCD Line 1: Voltage in [V]

Loop60:
        mov     R16     ,   R8
        cpi     R16     ,   16
        breq    Loop610
        cpi     R16     ,   4
        breq    Loop620
        cpi     R16     ,   2
        breq    Loop630
        cpi     R16     ,   1
        breq    Loop640
Loop610:
        rjmp    Loop61                      ; MR   0.25000 V
Loop620:
        rjmp    Loop62                      ; MR   2.5000  V
Loop630:
        rjmp    Loop63                      ; MR   25.000  V
Loop640:
        rjmp    Loop64                      ; MR   250.00  V
Loop61:                                     ; MR   0.25000 VDC
        push    R17                         ; is being popped in Loop609
        ldi     R17     ,   48
        rcall   LCD_line1
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '='
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '0'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        mov     R16     ,   R6
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R5
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R4
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R3
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R2
        add     R16     ,   R17
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
```

```
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop609
Loop62:                                 ; MR   2.5000  VDC
        push    R17                     ; is being popped in Loop609
        ldi     R17     ,   48
        rcall   LCD_line1
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '='
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        mov     R16     ,   R6          ; R2, R3, R4, R5 and R6:
        add     R16     ,   R17         ; 1. Mul1616,    Multiplication of the ADC - value with 25
        rcall   LCD_data                ; 2. Bin16inDez, change into decimal numbers
        ldi     R16     ,   '.'         ;
        rcall   LCD_data                ; 1.:
        mov     R16     ,   R5          ; the ADC deliveres 10-Bit values from 0 until 1023
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R4
        add     R16     ,   R17
        rcall   LCD_data                ; 2.:
        mov     R16     ,   R3          ; the LCD shows ASCII signs
        add     R16     ,   R17         ; the ASCII signs are being created by the addition
        rcall   LCD_data                ; of the Dezimalnumbers plus 48
        mov     R16     ,   R2
        add     R16     ,   R17         ; Example: ASCII '5' = 5 + 48
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop609
Loop63:                                 ; MR  25.000  VDC
        push    R17                     ; is being popped in Loop609
        ldi     R17     ,   48
        rcall   LCD_line1
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '='
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        mov     R16     ,   R6          ; fade out the leading 0
        cpi     R16     ,   0
        breq    Loop631
        rjmp    Loop632
Loop631:
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop633
Loop632:
        mov     R16     ,   R6
        add     R16     ,   R17
        rcall   LCD_data
Loop633:
        mov     R16     ,   R5
        add     R16     ,   R17
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        mov     R16     ,   R4
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R3
        add     R16     ,   R17
        rcall   LCD_data
```

```
        mov     R16     ,   R2
        add     R16     ,   R17
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop609
Loop64:                                 ; MR 255.50  VDC
        push    R17                     ; is being popped in Loop609
        ldi     R17     ,   48
        rcall   LCD_line1
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '='
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        mov     R16     ,   R6          ; fade out the first leading 0
        cpi     R16     ,   0
        brne    Loop641
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop642
Loop641:
        mov     R16     ,   R6
        add     R16     ,   R17
        rcall   LCD_data
Loop642:
        mov     R16     ,   R6          ; if the first leading number > 0, then
        cpi     R16     ,   0           ; do not fade out the second leading number
        brne    Loop643
        mov     R16     ,   R5          ; fade out the second leading number
        cpi     R16     ,   0
        brne    Loop643
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop644
Loop643:
        mov     R16     ,   R5
        add     R16     ,   R17
        rcall   LCD_data
Loop644:
        mov     R16     ,   R4
        add     R16     ,   R17
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        mov     R16     ,   R3
        add     R16     ,   R17
        rcall   LCD_data
        mov     R16     ,   R2
        add     R16     ,   R17
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        rjmp    Loop609
Loop609:
        pop     R17                     ; have been pushed in Loop61, Loop62, Loop63 and Loop64

        mov     R16     ,   R11         ; R11 = 1: LOW BATTERY
        cpi     R16     ,   1
        breq    Loop607

        mov     R16     ,   R7
        cpi     R16     ,   1
        breq    Loop691
        cpi     R16     ,   2
        breq    Loop691
```

```
        cpi     R16     ,   3
        breq    Loop691
        cpi     R16     ,   4
        breq    Loop691
        cpi     R16     ,   5
        breq    Loop691
        cpi     R16     ,   6
        breq    Loop692
        cpi     R16     ,   7
        breq    Loop692
        cpi     R16     ,   8
        breq    Loop692
        cpi     R16     ,   9
        breq    Loop692
        cpi     R16     ,   10
        breq    Loop692
Loop691:
        ldi     R16     ,   14
        rcall   LCD_goto
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        rjmp    Loop70
Loop692:
        ldi     R16     ,   14
        rcall   LCD_goto
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        rjmp    Loop70
Loop607:
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'L'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'W'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'B'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'Y'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   10
        rcall   Wait
        rjmp    Loop

;-----------------------------------------------------------------------------------------
; Display of the fields, bars and blanks in LCD_line2
; from here only comments in German language unless somebody wants me to translate them
;
; Ausgabe der Felder, Balken und Blanks in der Zeile 2 des LCD
;
; Das verwendete LCD Display hat 2 Zeilen zu je 16 Zeichen. Jedes Zeichen besteht
; aus 5 horizontalen mal 7 vertikalen Punkten.
; In der 2. (unteren) Zeile sollen 0 bis 80 "Balken" als Quasi-Analog-Anzeige
; erscheinen. Hierzu dienen 6 Sonderzeichen, welche vom Programm ins CGRAM der
; LCD Anzeige geschrieben werden. 8 Sonderzeichen sind max. möglich.
;
;      Feld     0 Balken    1 Balken    2 Balken    3 Balken    4 Balken    5 Balken
;   Zeichen 5   Zeichen 0   Zeichen 1   Zeichen 2   Zeichen 3   Zeichen 4   Zeichen 5
;
;   00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0
;   00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0
```

```
;   00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0
;   11111 = 31  00000 =  0  10000 = 16  11000 = 24  11100 = 28  11110 = 30  11111 = 31
;   11111 = 31  00000 =  0  10000 = 16  11000 = 24  11100 = 28  11110 = 30  11111 = 31
;   11111 = 31  00000 =  0  10000 = 16  11000 = 24  11100 = 28  11110 = 30  11111 = 31
;   00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0
;   00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0  00000 =  0
;
;   Zeichen für Feld  = Zeichen für 5 Balken
;   Zeichen für Blank = Zeichen für 0 Balken
;
; Beispiel: Gemessen wird 1.1875 [VDC]
;
; 1.1875 - 0.6400 = 0.5475
;
; R23 = 1, Stufe 18, R24 = 18 + 20 * 1 = 38
;
; Die Ausgabe der Sonderzeichen in der 2. Zeile des LCD muß also so geschehen, dass
; 38 "Balken" erscheinen:
;
; ||||| ||||| ||||| ||||| ||||| ||||| ||||| |||
; ||||| ||||| ||||| ||||| ||||| ||||| ||||| |||
; ||||| ||||| ||||| ||||| ||||| ||||| ||||| |||
;
; Auszugeben sind 7 mal das Zeichen 5 und 1 mal das Zeichen 3 sowie 8 Blanks.
;
Loop70:
        push    R19                     ; R19: Anzahl der Felder in LCD_line2
        push    R20                     ; R20: Anzahl der Balken in LCD_line2
        push    R21                     ; R21: Anzahl der Blanks in LCD_line2
;-------------------------------------------------------------------------------
Loop701:                                ; R24 =  1, 0.0025 bis 0.0300 VDC
        cpi     R24     ,   1
        brne    Loop702
        ldi     R19     ,   0           ;  0 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   15          ; 15 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop702:                                ; R24 =  2, 0.0325 bis 0.0625 VDC
        cpi     R24     ,   2
        brne    Loop703
        ldi     R19     ,   0           ;  0 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   15          ; 15 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop703:                                ; R24 =  3, 0.0625 bis 0.0950 VDC
        cpi     R24     ,   3
        brne    Loop704
        ldi     R19     ,   0           ;  0 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   15          ; 15 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop704:                                ; R24 =  4, 0.0957 bis 0.1275 VDC
        cpi     R24     ,   4
        brne    Loop705
        ldi     R19     ,   0           ;  0 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   15          ; 15 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop705:                                ; R24 =  5, 0.1300 bis 0.1575 VDC
        cpi     R24     ,   5
        brne    Loop706
        ldi     R19     ,   0           ;  0 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   15          ; 15 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop706:                                ; R24 =  6, 0.1600 bis 0.1900 VDC
        cpi     R24     ,   6
        brne    Loop707
        ldi     R19     ,   1           ;  1 Feld
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   14          ; 14 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop707:                                ; R24 =  7, 0.1925 bis 0.2225 VDC
        cpi     R24     ,   7
        brne    Loop708
        ldi     R19     ,   1           ;  1 Feld
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   14          ; 14 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop708:                                ; R24 =  8, 0.2250 bis 0.2550 VDC
```

```
        cpi     R24     ,   8
        brne    Loop709
        ldi     R19     ,   1               ;  1 Feld
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   14              ; 14 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop709:                                    ; R24 =  9, 0.2575 bis 0.2875 VDC
        cpi     R24     ,   9
        brne    Loop710
        ldi     R19     ,   1               ;  1 Feld
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   14              ; 14 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop710:                                    ; R24 = 10, 0.2900 bis 0.3175 VDC
        cpi     R24     ,   10
        brne    Loop711
        ldi     R19     ,   1               ;  1 Feld
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   14              ; 14 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop711:                                    ; R24 = 11, 0.3200 bis 0.3500 VDC
        cpi     R24     ,   11
        brne    Loop712
        ldi     R19     ,   2               ;  2 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   13              ; 13 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop712:                                    ; R24 = 12, 0.3525 bis 0.3825 VDC
        cpi     R24     ,   12
        brne    Loop713
        ldi     R19     ,   2               ;  2 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   13              ; 13 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop713:                                    ; R24 = 13, 0.3850 bis 0.4150 VDC
        cpi     R24     ,   13
        brne    Loop714
        ldi     R19     ,   2               ;  2 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   13              ; 13 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop714:                                    ; R24 = 14, 0.4175 bis 0.4475 VDC
        cpi     R24     ,   14
        brne    Loop715
        ldi     R19     ,   2               ;  2 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   13              ; 13 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop715:                                    ; R24 = 15, 0.4500 bis 0.4775 VDC
        cpi     R24     ,   15
        brne    Loop716
        ldi     R19     ,   2               ;  2 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   13              ; 13 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop716:                                    ; R24 = 16, 0.4800 bis 0.5100 VDC
        cpi     R24     ,   16
        brne    Loop717
        ldi     R19     ,   3               ;  3 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   12              ; 12 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop717:                                    ; R24 = 17, 0.5125 bis 0.5425 VDC
        cpi     R24     ,   17
        brne    Loop718
        ldi     R19     ,   3               ;  3 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   12              ; 12 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop718:                                    ; R24 = 18, 0.5450 bis 0.5750 VDC
        cpi     R24     ,   18
        brne    Loop719
        ldi     R19     ,   3               ;  3 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   12              ; 12 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
```

```
Loop719:                                ; R24 = 19, 0.5775 bis 0.6075 VDC
        cpi     R24     ,   19
        brne    Loop720
        ldi     R19     ,   3           ;  3 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   12          ; 12 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop720:                                ; R24 = 20, 0.6100 bis 0.6375 VDC
        cpi     R24     ,   20
        brne    Loop721
        ldi     R19     ,   3           ;  3 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   12          ; 12 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop721:                                ; R24 = 21, 0.6400 bis 0.6700 VDC
        cpi     R24     ,   21
        brne    Loop722
        ldi     R19     ,   4           ;  4 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   11          ; 11 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop722:                                ; R24 = 22, 0.6725 bis 0.7025 VDC
        cpi     R24     ,   22
        brne    Loop723
        ldi     R19     ,   4           ;  4 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   11          ; 11 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop723:                                ; R24 = 23, 0.7050 bis 0.7350 VDC
        cpi     R24     ,   23
        brne    Loop724
        ldi     R19     ,   4           ;  4 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   11          ; 11 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop724:                                ; R24 = 24, 0.7375 bis 0.7675 VDC
        cpi     R24     ,   24
        brne    Loop725
        ldi     R19     ,   4           ;  4 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   11          ; 11 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop725:                                ; R24 = 25, 0.7700 bis 0.7975 VDC
        cpi     R24     ,   25
        brne    Loop726
        ldi     R19     ,   4           ;  4 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   11          ; 11 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop726:                                ; R24 = 26, 0.8000 bis 0.8300 VDC
        cpi     R24     ,   26
        brne    Loop727
        ldi     R19     ,   5           ;  5 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   10          ; 10 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop727:                                ; R24 = 27, 0.8325 bis 0.8625 VDC
        cpi     R24     ,   27
        brne    Loop728
        ldi     R19     ,   5           ;  5 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   10          ; 10 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop728:                                ; R24 = 28, 0.8650 bis 0.8950 VDC
        cpi     R24     ,   28
        brne    Loop729
        ldi     R19     ,   5           ;  5 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   10          ; 10 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop729:                                ; R24 = 29, 0.8975 bis 0.9275 VDC
        cpi     R24     ,   29
        brne    Loop730
        ldi     R19     ,   5           ;  5 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   10          ; 10 Blanks
        rjmp    Loop80
```

```
;-------------------------------------------------------------------------------
Loop730:                                ; R24 = 30, 0.9300 bis 0.9575 VDC
        cpi     R24     ,   30
        brne    Loop731
        ldi     R19     ,   5           ;  5 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   10          ; 10 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop731:                                ; R24 = 31, 0.9600 bis 0.9900 VDC
        cpi     R24     ,   31
        brne    Loop732
        ldi     R19     ,   6           ;  6 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   9           ;  9 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop732:                                ; R24 = 32, 0.9925 bis 1.0225 VDC
        cpi     R24     ,   32
        brne    Loop733
        ldi     R19     ,   6           ;  6 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   9           ;  9 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop733:                                ; R24 = 33, 1.0250 bis 1.0550 VDC
        cpi     R24     ,   33
        brne    Loop734
        ldi     R19     ,   6           ;  6 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   9           ;  9 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop734:                                ; R24 = 34, 1.0575 bis 1.0875 VDC
        cpi     R24     ,   34
        brne    Loop735
        ldi     R19     ,   6           ;  6 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   9           ;  9 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop735:                                ; R24 = 35, 1.0900 bis 1.1175 VDC
        cpi     R24     ,   35
        brne    Loop736
        ldi     R19     ,   6           ;  6 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   9           ;  9 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop736:                                ; R24 = 36, 1.1200 bis 1.1500 VDC
        cpi     R24     ,   36
        brne    Loop737
        ldi     R19     ,   7           ;  7 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   8           ;  8 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop737:                                ; R24 = 37, 1.1525 bis 1.1825 VDC
        cpi     R24     ,   37
        brne    Loop738
        ldi     R19     ,   7           ;  7 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   8           ;  8 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop738:                                ; R24 = 38, 1.1850 bis 1.2150 VDC
        cpi     R24     ,   38
        brne    Loop739
        ldi     R19     ,   7           ;  7 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   8           ;  8 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop739:                                ; R24 = 39, 1.2175 bis 1.2475 VDC
        cpi     R24     ,   39
        brne    Loop740
        ldi     R19     ,   7           ;  7 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   8           ;  8 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop740:                                ; R24 = 40, 1.2500 bis 1.2775 VDC
        cpi     R24     ,   40
        brne    Loop741
        ldi     R19     ,   7           ;  7 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   8           ;  8 Blanks
```

```
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop741:                                ; R24 = 41, 1.2800 bis 1.3100 VDC
        cpi     R24     ,   41
        brne    Loop742
        ldi     R19     ,   8           ;  8 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   7           ;  7 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop742:                                ; R24 = 42, 1.3125 bis 1.3425 VDC
        cpi     R24     ,   42
        brne    Loop743
        ldi     R19     ,   8           ;  8 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   7           ;  7 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop743:                                ; R24 = 43, 1.3450 bis 1.3750 VDC
        cpi     R24     ,   43
        brne    Loop744
        ldi     R19     ,   8           ;  8 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   7           ;  7 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop744:                                ; R24 = 44, 1.3775 bis 1.4075 VDC
        cpi     R24     ,   44
        brne    Loop745
        ldi     R19     ,   8           ;  8 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   7           ;  7 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop745:                                ; R24 = 45, 1.4100 bis 1.4375 VDC
        cpi     R24     ,   45
        brne    Loop746
        ldi     R19     ,   8           ;  8 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   7           ;  7 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop746:                                ; R24 = 46, 1.4400 bis 1.4700 VDC
        cpi     R24     ,   46
        brne    Loop747
        ldi     R19     ,   9           ;  9 Felder
        ldi     R20     ,   1           ;  1 Balken
        ldi     R21     ,   6           ;  6 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop747:                                ; R24 = 47, 1.4725 bis 1.5025 VDC
        cpi     R24     ,   47
        brne    Loop748
        ldi     R19     ,   9           ;  9 Felder
        ldi     R20     ,   2           ;  2 Balken
        ldi     R21     ,   6           ;  6 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop748:                                ; R24 = 48, 1.5050 bis 1.5350 VDC
        cpi     R24     ,   48
        brne    Loop749
        ldi     R19     ,   9           ;  9 Felder
        ldi     R20     ,   3           ;  3 Balken
        ldi     R21     ,   6           ;  6 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop749:                                ; R24 = 49, 1.5375 bis 1.5675 VDC
        cpi     R24     ,   49
        brne    Loop750
        ldi     R19     ,   9           ;  9 Felder
        ldi     R20     ,   4           ;  4 Balken
        ldi     R21     ,   6           ;  6 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop750:                                ; R24 = 50, 1.5700 bis 1.5975 VDC
        cpi     R24     ,   50
        brne    Loop751
        ldi     R19     ,   9           ;  9 Felder
        ldi     R20     ,   5           ;  5 Balken
        ldi     R21     ,   6           ;  6 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop751:                                ; R24 = 51, 1.6000 bis 1.6300 VDC
        cpi     R24     ,   51
        brne    Loop752
        ldi     R19     ,   10          ; 10 Felder
        ldi     R20     ,   1           ;  1 Balken
```

```
        ldi     R21     ,   5               ;  5 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop752:                                    ; R24 = 52, 1.5325 bis 1.6625 VDC
        cpi     R24     ,   52
        brne    Loop753
        ldi     R19     ,   10              ; 10 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   5               ;  5 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop753:                                    ; R24 = 53, 1.6650 bis 1.6950 VDC
        cpi     R24     ,   53
        brne    Loop754
        ldi     R19     ,   10              ; 10 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   5               ;  5 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop754:                                    ; R24 = 54, 1.6975 bis 1.7275 VDC
        cpi     R24     ,   54
        brne    Loop755
        ldi     R19     ,   10              ; 10 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   5               ;  5 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop755:                                    ; R24 = 55, 1.7300 bis 1.7575 VDC
        cpi     R24     ,   55
        brne    Loop756
        ldi     R19     ,   10              ; 10 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   5               ;  5 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop756:                                    ; R24 = 56, 1.7600 bis 1.7900 VDC
        cpi     R24     ,   56
        brne    Loop757
        ldi     R19     ,   11              ; 11 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   4               ;  4 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop757:                                    ; R24 = 57, 1.7925 bis 1.8225 VDC
        cpi     R24     ,   57
        brne    Loop758
        ldi     R19     ,   11              ; 11 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   4               ;  4 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop758:                                    ; R24 = 58, 1.8250 bis 1.8550 VDC
        cpi     R24     ,   58
        brne    Loop759
        ldi     R19     ,   11              ; 11 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   4               ;  4 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop759:                                    ; R24 = 59, 1.8575 bis 1.8875 VDC
        cpi     R24     ,   59
        brne    Loop760
        ldi     R19     ,   11              ; 11 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   4               ;  4 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop760:                                    ; R24 = 60, 1.8900 bis 1.9175 VDC
        cpi     R24     ,   60
        brne    Loop761
        ldi     R19     ,   11              ; 11 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   4               ;  4 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop761:                                    ; R24 = 61, 1.9200 bis 1.9500 VDC
        cpi     R24     ,   61
        brne    Loop762
        ldi     R19     ,   12              ; 12 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   3               ;  3 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop762:                                    ; R24 = 62, 1.9525 bis 1.9825 VDC
        cpi     R24     ,   62
        brne    Loop763
        ldi     R19     ,   12              ; 12 Felder
```

```
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   3               ;  3 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop763:                                    ; R24 = 63, 1.9850 bis 2.0150 VDC
        cpi     R24     ,   63
        brne    Loop764
        ldi     R19     ,   12              ; 12 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   3               ;  3 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop764:                                    ; R24 = 64, 2.0175 bis 2.0475 VDC
        cpi     R24     ,   64
        brne    Loop765
        ldi     R19     ,   12              ; 12 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   3               ;  3 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop765:                                    ; R24 = 65, 2.0500 bis 2.0775 VDC
        cpi     R24     ,   65
        brne    Loop766
        ldi     R19     ,   12              ; 12 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   3               ;  3 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop766:                                    ; R24 = 66, 2.0800 bis 2.1100 VDC
        cpi     R24     ,   66
        brne    Loop767
        ldi     R19     ,   13              ; 13 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   2               ;  2 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop767:                                    ; R24 = 67, 2.1125 bis 2.1425 VDC
        cpi     R24     ,   67
        brne    Loop768
        ldi     R19     ,   13              ; 13 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   2               ;  2 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop768:                                    ; R24 = 68, 2.1450 bis 2.1750 VDC
        cpi     R24     ,   68
        brne    Loop769
        ldi     R19     ,   13              ; 13 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   2               ;  2 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop769:                                    ; R24 = 69, 2.1775 bis 2.2075 VDC
        cpi     R24     ,   69
        brne    Loop770
        ldi     R19     ,   13              ; 13 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   2               ;  2 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop770:                                    ; R24 = 70, 2.2100 bis 2.2375 VDC
        cpi     R24     ,   70
        brne    Loop771
        ldi     R19     ,   13              ; 13 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   2               ;  2 Blanks
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop771:                                    ; R24 = 71, 2.2400 bis 2.2700 VDC
        cpi     R24     ,   71
        brne    Loop772
        ldi     R19     ,   14              ; 14 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   1               ;  1 Blank
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop772:                                    ; R24 = 72, 2.2725 bis 2.3025 VDC
        cpi     R24     ,   72
        brne    Loop773
        ldi     R19     ,   14              ; 14 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   1               ;  1 Blank
        rjmp    Loop80
;-------------------------------------------------------------------------------
Loop773:                                    ; R24 = 73, 2.3050 bis 2.3350 VDC
        cpi     R24     ,   73
        brne    Loop774
```

```
        ldi     R19     ,   14              ; 14 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   1               ;  1 Blank
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop774:                                    ; R24 = 74, 2.3375 bis 2.3675 VDC
        cpi     R24     ,   74
        brne    Loop775
        ldi     R19     ,   14              ; 14 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   1               ;  1 Blank
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop775:                                    ; R24 = 75, 2.3700 bis 2.3975 VDC
        cpi     R24     ,   75
        brne    Loop776
        ldi     R19     ,   14              ; 14 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   1               ;  1 Blank
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop776:                                    ; R24 = 76, 2.4000 bis 2.4300 VDC
        cpi     R24     ,   76
        brne    Loop777
        ldi     R19     ,   15              ; 15 Felder
        ldi     R20     ,   1               ;  1 Balken
        ldi     R21     ,   0               ;  0 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop777:                                    ; R24 = 77, 2.4325 bis 2.4625 VDC
        cpi     R24     ,   77
        brne    Loop778
        ldi     R19     ,   15              ; 15 Felder
        ldi     R20     ,   2               ;  2 Balken
        ldi     R21     ,   0               ;  0 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop778:                                    ; R24 = 78, 2.4650 bis 2.4950 VDC
        cpi     R24     ,   78
        brne    Loop779
        ldi     R19     ,   15              ; 15 Felder
        ldi     R20     ,   3               ;  3 Balken
        ldi     R21     ,   0               ;  0 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop779:                                    ; R24 = 79, 2.4975 bis 2.5275 VDC
        cpi     R24     ,   79
        brne    Loop780
        ldi     R19     ,   15              ; 15 Felder
        ldi     R20     ,   4               ;  4 Balken
        ldi     R21     ,   0               ;  0 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
Loop780:                                    ; R24 = 80, 2.5300 bis 2.5575 VDC
        ldi     R19     ,   15              ; 15 Felder
        ldi     R20     ,   5               ;  5 Balken
        ldi     R21     ,   0               ;  0 Blanks
        rjmp    Loop80
;-----------------------------------------------------------------------------------
; Ausgabe LCD Zeile 2: Löschen der LCD Zeile 2 durch Ausgabe von Blanks wenn R22 = 0 und R23 = 0

Loop80:
        tst     R23                         ;
        brne    Loop81                      ; R23 > 0 oder/und R22 > 0 ?
        tst     R22                         ; dann ist U_Mess > 0
        brne    Loop81                      ; und Balken werden in der LCD Zeile 2 ausgegeben
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
```

```
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data

        pop     R21
        pop     R20
        pop     R19

Loop81:
        cbr     R19     ,   0b11000000

        cbi     PORTD   ,   4               ; RS
        cbi     PORTD   ,   5               ; R/W
        ldi     R16     ,   0b01000000
        rcall   LCD_cmd
        sbi     PORTD   ,   4               ; RS
        sbi     PORTD   ,   5               ; R/W

        rcall   LCD_line2
Loop82:                                     ; Felder, 0 ... 15
        cpi     R19     ,   0
        breq    Loop83
        ldi     R16     ,   5
        rcall   LCD_data
        dec     R19
        tst     R19
        brne    Loop82
Loop83:                                     ; Balken, 1 ... 5
        cpi     R20     ,   1
        breq    Loop831
        cpi     R20     ,   2
        breq    Loop832
        cpi     R20     ,   3
        breq    Loop833
        cpi     R20     ,   4
        breq    Loop834
        cpi     R20     ,   5
        breq    Loop835
Loop831:                                    ; 1 Balken
        ldi     R16     ,   1
        ldi     R16     ,   0b00000001
        rcall   LCD_data
        rjmp    Loop84
Loop832:                                    ; 2 Balken
        ldi     R16     ,   2
        ldi     R16     ,   0b00000010
        rcall   LCD_data
        rjmp    Loop84
Loop833:                                    ; 3 Balken
        ldi     R16     ,   3
        ldi     R16     ,   0b00000011
        rcall   LCD_data
        rjmp    Loop84
Loop834:                                    ; 4 Balken
        ldi     R16     ,   4
        ldi     R16     ,   0b00000100
        rcall   LCD_data
        rjmp    Loop84
Loop835:                                    ; 5 Balken
        ldi     R16     ,   5
        ldi     R16     ,   0b00000101
        rcall   LCD_data
        rjmp    Loop84
Loop84:                                     ; Blanks, 15 ... 0
        cpi     R21     ,   0
        breq    Loop85
        ldi     R16     ,   0
        rcall   LCD_data
        dec     R21
        tst     R21
        brne    Loop84
Loop85:
        pop     R21
        pop     R20
        pop     R19

        ldi     R16     ,   1
        rcall   Wait
```

```
        rjmp    Loop

;-------------------------------------------------------------------------------------
; Wartezeit 0.1 ms bei 4 MHz
wait01ms:
        push    R16
        push    R17
        ldi     R17     ,   1               ; für 0.1 ms = 1
w01ms1:
        ldi     R16     ,   123
w01ms2:
        dec     R16
        brne    w01ms2
        dec     R17
        brne    w01ms1
        pop     R17
        pop     R16
        ret

;-------------------------------------------------------------------------------------
; Wartezeit 5 ms bei 4 MHz
wait5ms:
        push    R16
        push    R17
        ldi     R17     ,   50              ; für 1 ms = 10
w5ms1:
        ldi     R16     ,   123
w5ms2:
        dec     R16
        brne    w5ms2
        dec     R17
        brne    w5ms1
        pop     R17
        pop     R16
        ret

;-------------------------------------------------------------------------------------
; Wartezeit: R16 = 1 ==> 0.01 s, R16 = 255 ==> 2.55 s bei = 3.686 MHz

Wait:
        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
        ret

;-------------------------------------------------------------------------------------
;************************************************************* Anfang Initialisierung
; Initialisierung des LCD DOGM162W-A mit ST7036 Dot Matrix LCD Controller/Driver
;
; DAS LCD DOGM162W-A MIT DEM ST7036 KANN OHNE DIE KONTROLLE DES BUSY FLAG NICHT
; INITIALISIERT WERDEN !!!
;
; ST7036 Datenblatt Seite 45:
; "The busy flag must be checked (one instruction) after the 4-bit data has been
; transferred twice. Two more 4-bit operations then transfer the busy flag and address
; counter data."
;
; Dies bedeutet auch, dass das R/W Signal zwingend an einen Port des MPU angeschlossen
; sein muß!

LCD_init:
        push    R18
        ldi     R18     ,   50
PowerUpWait:
        rcall   wait5ms                     ; Power Up Wait Time
        dec     R18                         ; 250 ms
        brne    PowerUpWait                 ;
        pop     R18
```

```
;-------------------------------------------------------------------------------
; Function Set 01: 3 Mal den 8-Bit Mode ausgeben
;
; 0b00110000 lt. ST7036 Datenblatt S. 41 und 42
;
; Definitionen auf S. 31    DL  (Bit 4) = 1 (8-Bit Mode)
;                           N   (Bit 3) = 0 (1-Line Mode)
;                           DH  (Bit 2) = 0 (Double Height Mode)
; Definitionen auf S. 32    IS2 (Bit 1) = 0 (Instruction Table 0, S.28)
;                           IS1 (Bit 0) = 0 (Instruction Table 0, S.28)
; das gilt für eine 8-Bit Verdrahtung

;   Pin     Function                      Set
;
;   35      DB0 Data Bit 0                 0
;   34      DB1 Data Bit 1                 0
;   33      DB2 Data Bit 2                 0
;   32      DB3 Data Bit 3                 0
;   31      DB4 Data Bit 4                 1
;   30      DB5 Data Bit 5                 1
;   29      DB6 Data Bit 6                 0
;   28      DB7 Data Bit 7                 0
;   36      E   Falling Edge: Enable       0
;   37      R/W Low: Read, High: Write     0
;   39      RS  Low: Command, High: Data   0

; bei der 4-Bit Verdrahtung ergibt sich folgendes Bild:

;   Pin     Function                      Set
;
;   31      DB4 Data Bit 4                 1
;   30      DB5 Data Bit 5                 1
;   29      DB6 Data Bit 6                 0
;   28      DB7 Data Bit 7                 0
;   36      E   Falling Edge: Enable       0
;   37      R/W Low: Read, High: Write     0
;   39      RS  Low: Command, High: Data   0

; somit ist auszugeben: 0b00000011
; dieser Zusammenhang wird im AVR-Tutorial, Kapitel 7: LCD 7.41 erklärt, ist aber
; auch ohne diese Erklärung logisch

        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
        rcall   wait5ms                     ; Wait 5ms
;-------------------------------------------------------------------------------
; Function Set 02: Das 2. Mal
        rcall   LCD_enable                  ; Impuls von E = MPU D.6 Pin 12
        rcall   wait5ms                     ;
;-------------------------------------------------------------------------------
; Function Set 03: Das 3. Mal
        rcall   LCD_enable                  ; Impuls von E = MPU D.6 Pin 12
        rcall   wait5ms                     ;

;-------------------------------------------------------------------------------
; Function Set 04: Den 4-Bit Mode einschalten
;
; 0b00100000 lt. ST7036 Datenblatt S. 41 und 42
;
; weil hier der St7036 noch von einer 8-Bit Verdrahtung ausgeht gilt das
; unter Function Set 01 gesagte
; Auszugeben ist also nicht 0b00100000 sondern 0b00000010
; DL (Bit 4) = 0: 4-Bit Mode einschalten, S. 31
;
        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
        rcall   wait5ms                     ;

; Ab hier sind die Daten in jeweils 2 Nibbles auszugeben.
; Das geschieht mittels der SUBs LCD_cmd für Commands und LCD_data für Daten

;-------------------------------------------------------------------------------
; Function Set 05: 4-Bit Mode, 2 Lines, 5x8 Dots, Extension Instructions, GCGRAM not available
;
; 0b00101001 lt. ST7036 Datenblatt S. 41 und 42
;
; DL  = 0 (Bit 4): 4-Bit Mode               ; ST7036 S. 31
; N   = 1 (Bit 3): 2-Line Display Mode      ; ST7036 S. 31
; DH  = 0 (Bit 2): 5x8 Dot Font             ; ST7036 S. 31
```

```
; IS2 = 0 (Bit 1): Extension Instruction    ; ST7036 S. 28: Instruction Table 1
; IS1 = 1 (Bit 0): Extension Instruction    ; ST7036 S. 28: Instruction Table 1
;                                           ; ST7036 S. 28: Set CGRAM not available
;
       ldi     R16     ,   0b00101001
       rcall   LCD_cmd

;-------------------------------------------------------------------------------
; Function Set 06: Bias, 2 Lines
;
; 0b00010100 lt. ST7036 Datenblatt S. 41 und 42
;
; BS  = 1 (Bit 2): Bias = 1/4               ; ST7036 S. 36
; FX  = 0 (Bit 0): 2 Lines                  ; ST7036 S. 36
;
; Unklar: Wie läßt sich fesstellen, ob sich in der Schaltung externe Widerstände
; befinden? Weil es dazu keine Pins gibt kann man weder selbst welche einlöten noch
; mit einem Ohmmeter messen, ob welche eingelötet sind.
; Wie lassen sich die Zustände von OPF1 und OPF2 feststellen?
; Im ST7036 S.1 steht für den Typ (product name) ST7036-0A OPR1 = 1 und OPR2 = 1
; Ob dieser Typ aber im DOGM162W-A Verwendung findet, geht nicht aus dessen
; Datenblatt hervor. Davon wird aber im Folgenden ausgegangen.
; Unter welchen Umständen soll Bias = 1/4 bzw. 1/5 sein? Was genau ist damit
; gemeint? Es fehlen Schaltpläne.
;
       ldi     R16     ,   0b00010100      ; BIAS
       rcall   LCD_cmd                     ;

;-------------------------------------------------------------------------------
; Function Set 07: Contrast Set (Low Byte)
;
; 0b01111000 lt. ST7036 Datenblatt S. 41 und 42
;
; C3  = 1 (Bit 3): Contrast Set Low Byte    ; ST7036 S. 37
; C2  = 0 (Bit 2): Contrast Set Low Byte    ; ST7036 S. 37
; C1  = 0 (Bit 1): Contrast Set Low Byte    ; ST7036 S. 37
; C0  = 0 (Bit 0): Contrast Set Low Byte    ; ST7036 S. 37
;
       ldi     R16     ,   0b01111000      ; Contrast Set
       rcall   LCD_cmd

;-------------------------------------------------------------------------------
; Function Set 08: ICON/Power/Contrast (High Byte)
;
; 0b01011110 lt. ST7036 Datenblatt S. 41 und 42
;
; Ion = 1 (Bit 3): ICON Display On          ; ST7036 S. 36
; Bon = 1 (Bit 2): Booster On               ; ST7036 S. 36
; C5  = 1 (Bit 1): Contrast Set High Byte   ; ST7036 S. 36
; C4  = 0 (Bit 0): Contrast Set High Byte   ; ST7036 S. 36
;
       ldi     R16     ,   0b01011110      ; ICON/Power/Contrast
       rcall   LCD_cmd                     ;

;-------------------------------------------------------------------------------
; Function Set 09: Follower Control
;
; Ein Voltage Follower ist ein Operationsverstärker, dessen Ausgang mit dem
; invertierenden Eingang verbunden ist. Angesteuert wird er am nicht invertierenden
; Eingang. Dieser ist sehr hochohmig, so dass die Spannungsquelle kaum belastet
; wird. Am Ausgang steht dieselbe Spannung an wie am Eingang. Hier kann aber
; ein relativ grosser Strom entnommen werden, ohne dass die Spannung einbricht.
; Ob mit Internal Follower eine solche Schaltung gemeint ist, wird nicht klar
; beschrieben. Sie hätte dann einen Sinn, wenn mit dem an anderer Stelle
; "Booster" genannter Schaltungsteil eine sog. Ladungspumpe (Charge Pump)
; gemeint ist. Auch hierzu fehlen Schaltpläne.
;
; 0b01101010 lt. ST7036 Datenblatt S. 41 und 42
;
; Fon = 1 (Bit 3): Internal Follower On     ; ST7036 S. 37
; Rab2= 0 (Bit 2): V0 Generator             ; ST7036 S. 37
; Rab1= 1 (Bit 1): V0 Generator             ; ST7036 S. 37
; Rab0= 0 (Bit 0): V0 Generator             ; ST7036 S. 37
;
       ldi     R16     ,   0b01101010      ; Follower Control
       rcall   LCD_cmd                     ;

;-------------------------------------------------------------------------------
; Function Set 10: 4-Bit Mode, 2 Lines, 5x8 Dots, Normal Instructions, GCGRAM available
;
; 0b00101000 lt. ST7036 Datenblatt S. 41 und 42 nicht drin, aber nötig!
;
; DL  = 0 (Bit 4): 4-Bit Mode               ; ST7036 S. 31
; N   = 1 (Bit 3): 2-Line Display Mode      ; ST7036 S. 31
; DH  = 0 (Bit 2): 5x8 Dot Font             ; ST7036 S. 31
; IS2 = 0 (Bit 0): Extension Instruction    ; ST7036 S. 28: Instruction Table 0
; IS1 = 1 (Bit 0): Extension Instruction    ; ST7036 S. 28: Instruction Table 0
```

```
;                                       ; ST7036 S. 28: Set CGRAM available
;
        ldi     R16     ,   0b00101000
        rcall   LCD_cmd
;-------------------------------------------------------------------------------
; Function Set 11: Display On
;
; 0b00001100 lt. ST7036 Datenblatt S. 41 und 42
;
; D   = 1 (Bit 2): Display On           ; ST7036 S. 30
; C   = 0 (Bit 1): Cursor Off           ; ST7036 S. 30
; B   = 0 (Bit 0): Cursor Blink Off     ; ST7036 S. 30
;
        ldi     R16     ,   0b00001100  ; Display On
        rcall   LCD_cmd                 ;

;-------------------------------------------------------------------------------
; Function Set 12: Clear Display
;
; 0b00001101 lt. ST7036 Datenblatt S. 41 und 42
;
        ldi     R16     ,   0b00001101  ; Clear Display
        rcall   LCD_cmd                 ;

;-------------------------------------------------------------------------------
; Functin Set 13: Entry Mode Set
;
; 0b00001100 lt. ST7036 Datenblatt S. 41 und 42
;
; I/D = 1 (Bit 1): Cursor Moves To Right
; S   = 0 (Bit 0): No Shift Of The Entire Display
;
        ldi     R16     ,   0b00001100  ; Entry Mode Set
        rcall   LCD_cmd

; The End
        ret

;****************************************************************** Ende Initialisierung
;-------------------------------------------------------------------------------
; 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:
        ldi     R16     ,   0b00000001      ; Display löschen
        rcall   LCD_cmd
        ret

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

;-----------------------------------------------------------------------------
LCD_off:
        ldi     R16     ,   0b00001000
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
LCD_on:
        ldi     R16     ,   0b00001110
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
LCD_line1:
        ldi     R16     ,   0b10000000      ; DRAM auf Adresse 0x00
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
LCD_line2:
        ldi     R16     ,   0b11000000      ; DRAM auf Adresse 0x40
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
; Goto R16 = Adresse (Zeile 1 = 0...15, Zeile 2 = 64...79
LCD_goto:
        ori     R16     ,   0b10000000      ; Goto DRAM auf Adresse R16
                                            ; ORI - Logical OR with Immediate
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
LCD_CUL:
        ldi     R16     ,   0b00010000      ; Cursor um eine Poation nach links
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
LCD_CUR:
        ldi     R16     ,   0b00010100      ; Cursor um eine Postion nach rechts
        rcall   LCD_cmd
        ret

;-----------------------------------------------------------------------------
; R16 numerisch augeben, d.h. als Ziffern 0 bis 255, dezimal
```

```
LCD_number:
        push    R16
        push    R17
        mov     R17     ,   R16
        ldi     R16     ,   '0'-1
LCD_number1:
        inc     R16
        subi    R17     ,   100
        brcc    LCD_number1
        subi    R17     ,   -100
        rcall   LCD_data
        ldi     R16     ,   '0'-1
LCD_number2:
        inc     R16
        subi    R17     ,   10
        brcc    LCD_number2
        subi    R17     ,   -10
        rcall   LCD_data
LCD_number3:
        ldi     R16     ,   '0'
        add     R16     ,   R17
        rcall   LCD_data
        pop     R17
        pop     R16
        ret

;-----------------------------------------------------------------------------------
onADC:
        push    R0
        push    R1
        push    R9
        push    R10
        push    R16
        push    R17
        push    R18
        push    R19
        push    R20
        push    R21

        in      R16     ,   ADCL            ; 10 Bit ADC-Daten holen
        in      R17     ,   ADCH
onADC1:                                     ;*****************************************
        mov     R18     ,   R16
        mov     R19     ,   R17
        mov     R22     ,   R16             ; für Quasi-Analoganzeige in Loop, R22 = Low
        mov     R23     ,   R17             ; für Quasi-Analoganzeige in Loop, R23 = High

        rol     R17                         ;
        rol     R17                         ; R17.0 --> R17.2, R17.1 -> R17.3 kopieren
        cbr     R17     ,   0b11110000      ; R17.4 bis R17.7 löschen

        sbrc    R16     ,   6               ;
        sbr     R17     ,   0b00000001      ; R16.6 -> R17.0 kopieren
        sbrc    R16     ,   7               ; R16.7 -> R17.1 kopieren
        sbr     R17     ,   0b00000010      ;
        cbr     R16     ,   0b11000000      ; R16.6 und R16.7 löschen

        mov     R23     ,   R19             ; Multiplikant,  für M1
        mov     R22     ,   R18             ; 0 bis 1023

        ldi     R21     ,   0b00000000      ; Multiplikator
        ldi     R20     ,   0b00011001      ; 25
        mov     R10     ,   R21             ; Multiplikator, für M1
        mov     R9      ,   R20             ; Multiplikator, für M1

        mov     R0      ,   R18             ; 16-Bit Zahl Low  für D1
        mov     R1      ,   R19             ; 16-Bit Zahl High für D1

        rcall   M1
        cpi     R27     ,   0b00000001      ; Kalibriermodus Aus
        breq    onADC2
        cpi     R27     ,   0b00000010      ; Kalibriermodus Aus
        breq    onADC2
        cpi     R27     ,   0b00000011      ; Kalibriermodus Ein
        breq    onADC3
onADC2:
        rcall   K1                          ; Korekturberechnungen Ein
        rjmp    onADC4                      ; keine Bereichsanzeige
onADC3:
        rcall   K500                        ; Korekturberechnungen Aus
        rjmp    onADC4                      ; mit Bereichsanzeige
onADC4:
        rcall   D1

        pop     R21
        pop     R20
        pop     R19
```

```
        pop     R18
        pop     R17
        pop     R16
        pop     R10
        pop     R9
        pop     R1
        pop     R0

        reti

;-----------------------------------------------------------------------------------------
; Multiplikation von 2 Binärzahlen zu je 16 Bit
;
M1:
        mul     R22     ,   R9
        movw    R3:R2   ,   R1:R0
        mul     R23     ,   R10
        movw    R5:R4   ,   R1:R0
        mul     R22     ,   R10
        add     R3      ,   R0
        adc     R4      ,   R1
        clr     R0
        adc     R5      ,   R0
        mul     R23     ,   R9
        add     R3      ,   R0
        adc     R4      ,   R1
        clr     R0
        adc     R5      ,   R0
        mov     R18     ,   R3
        mov     R17     ,   R2
        ret

;-----------------------------------------------------------------------------------------
; Korrektur-Berechnungen für die 4 AC-Meßbereiche
;
K1:
        mov     R16     ,   R7
        cpi     R16     ,   7               ; AC 0,25 V
        breq    K11
        cpi     R16     ,   8               ; AC 2,5  V
        breq    K12
        cpi     R16     ,   9               ; AC 25   V
        breq    K13
        cpi     R16     ,   10              ; AC 250  V
        breq    K14
        ret
K11:
        rcall   K100
        ret
K12:
        rcall   K200
        ret
K13:
        rcall   K300
        ret
K14:
        rcall   K400
        ret

;-----------------------------------------------------------------------------------------
; Korrektur-Berechnung des Meßbereichs 0,25 VAC
;
K100:
        cpi     R24     ,   1
        brsh    K101
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K101:
        cpi     R24     ,   2
        brsh    K102
        ldi     R25     ,   0b01001011      ; - 3 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K102:
        cpi     R24     ,   4
        brsh    K103
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K103:
```

```
        cpi     R24     ,   7
        brsh    K104
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K104:
        cpi     R24     ,   10
        brsh    K105
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K105:
        cpi     R24     ,   13
        brsh    K106
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K106:
        cpi     R24     ,   16
        brsh    K107
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K107:
        cpi     R24     ,   19
        brsh    K108
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K108:
        cpi     R24     ,   23
        brsh    K109
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K109:
        cpi     R24     ,   26
        brsh    K110
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K110:
        cpi     R24     ,   29
        brsh    K111
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K111:
        cpi     R24     ,   32
        brsh    K112
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K112:
        cpi     R24     ,   35
        brsh    K113
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K113:
        cpi     R24     ,   38
        brsh    K114
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
```

```
K114:
        cpi     R24     ,   42
        brsh    K115
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K115:
        cpi     R24     ,   45
        brsh    K116
        ldi     R25     ,   0b11100001      ; - 9 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K116:
        cpi     R24     ,   48
        brsh    K117
        ldi     R25     ,   0b11001000      ; - 8 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K117:
        cpi     R24     ,   51
        brsh    K118
        ldi     R25     ,   0b10101111      ; - 7 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K118:
        cpi     R24     ,   54
        brsh    K119
        ldi     R25     ,   0b10101111      ; - 7 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K119:
        cpi     R24     ,   57
        brsh    K120
        ldi     R25     ,   0b10010110      ; - 6 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K120:
        cpi     R24     ,   60
        brsh    K121
        ldi     R25     ,   0b10010110      ; - 6 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K121:
        cpi     R24     ,   64
        brsh    K122
        ldi     R25     ,   0b01111101      ; - 5 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K122:
        cpi     R24     ,   68
        brsh    K123
        ldi     R25     ,   0b01100100      ; - 4 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K123:
        cpi     R24     ,   70
        brsh    K124
        ldi     R25     ,   0b01001011      ; - 3 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K124:
        cpi     R24     ,   74
        brsh    K125
        ldi     R25     ,   0b01001011      ; - 3 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
```

```
        rjmp    K127
K125:
        cpi     R24     ,   77
        brsh    K126
        ldi     R25     ,   0b00110010      ; - 2 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K126:
        ldi     R25     ,   0b00110010      ; - 2 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K127
K127:
        ret

;-------------------------------------------------------------------------------
; Korrektur-Berechnung des Meßbereichs 2,5 VAC
;
K200:
        cpi     R24     ,   1
        brsh    K201
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K201:
        cpi     R24     ,   2
        brsh    K202
        ldi     R25     ,   0b00110010      ; - 9 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K202:
        cpi     R24     ,   6
        brsh    K203
        ldi     R25     ,   0b01011110      ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K203:
        cpi     R24     ,   9
        brsh    K204
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K204:
        cpi     R24     ,   12
        brsh    K205
        ldi     R25     ,   0b11000010      ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K205:
        cpi     R24     ,   25
        brsh    K206
        ldi     R25     ,   0b11000010      ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K206:
        cpi     R24     ,   18
        brsh    K207
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K207:
        cpi     R24     ,   21
        brsh    K208
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K208:
        cpi     R24     ,   24
```

```
        brsh    K209
        ldi     R25     ,   0b11011011     ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K209:
        cpi     R24     ,   27
        brsh    K210
        ldi     R25     ,   0b11011011     ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K210:
        cpi     R24     ,   31
        brsh    K211
        ldi     R25     ,   0b11011011     ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K211:
        cpi     R24     ,   34
        brsh    K212
        ldi     R25     ,   0b11000010     ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K212:
        cpi     R24     ,   37
        brsh    K213
        ldi     R25     ,   0b10101001     ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K213:
        cpi     R24     ,   40
        brsh    K214
        ldi     R25     ,   0b10010000     ; -16 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K214:
        cpi     R24     ,   43
        brsh    K215
        ldi     R25     ,   0b10010000     ; -16 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K215:
        cpi     R24     ,   46
        brsh    K216
        ldi     R25     ,   0b01110111     ; -15 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K216:
        cpi     R24     ,   50
        brsh    K217
        ldi     R25     ,   0b01011110     ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K217:
        cpi     R24     ,   53
        brsh    K218
        ldi     R25     ,   0b01011110     ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K218:
        cpi     R24     ,   56
        brsh    K219
        ldi     R25     ,   0b01000101     ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K219:
```

```
        cpi     R24     ,   59
        brsh    K220
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K220:
        cpi     R24     ,   62
        brsh    K221
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K221:
        cpi     R24     ,   65
        brsh    K222
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K222:
        cpi     R24     ,   69
        brsh    K223
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K223:
        cpi     R24     ,   72
        brsh    K224
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K224:
        cpi     R24     ,   75
        brsh    K225
        ldi     R25     ,   0b11100001      ; - 9 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K225:
        cpi     R24     ,   78
        brsh    K226
        ldi     R25     ,   0b11001000      ; - 8 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K226:
        ldi     R25     ,   0b10101111      ; - 7 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K227
K227:
        ret

;-------------------------------------------------------------------------------
; Korrektur-Berechnung des Meßbereichs 25 VAC
;
K300:
        cpi     R24     ,   1
        brsh    K301
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K301:
        cpi     R24     ,   2
        brsh    K302
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K302:
        cpi     R24     ,   6
        brsh    K303
        ldi     R25     ,   0b01110111      ; -15 Digits
```

```
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K303:
        cpi     R24     ,   9
        brsh    K304
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K304:
        cpi     R24     ,   12
        brsh    K305
        ldi     R25     ,   0b11000010      ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K305:
        cpi     R24     ,   15
        brsh    K306
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K306:
        cpi     R24     ,   18
        brsh    K307
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K307:
        cpi     R24     ,   21
        brsh    K308
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K308:
        cpi     R24     ,   24
        brsh    K309
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K309:
        cpi     R24     ,   27
        brsh    K310
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K310:
        cpi     R24     ,   31
        brsh    K311
        ldi     R25     ,   0b11000010      ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K311:
        cpi     R24     ,   34
        brsh    K312
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K312:
        cpi     R24     ,   37
        brsh    K313
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K313:
        cpi     R24     ,   40
        brsh    K314
```

```
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K314:
        cpi     R24     ,   43
        brsh    K315
        ldi     R25     ,   0b10010000      ; -16 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K315:
        cpi     R24     ,   46
        brsh    K316
        ldi     R25     ,   0b01110111      ; -15 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K316:
        cpi     R24     ,   50
        brsh    K317
        ldi     R25     ,   0b01110111      ; -15 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K317:
        cpi     R24     ,   53
        brsh    K318
        ldi     R25     ,   0b01011110      ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K318:
        cpi     R24     ,   56
        brsh    K319
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K319:
        cpi     R24     ,   59
        brsh    K320
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K320:
        cpi     R24     ,   62
        brsh    K321
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K321:
        cpi     R24     ,   65
        brsh    K322
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K322:
        cpi     R24     ,   69
        brsh    K323
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K323:
        cpi     R24     ,   72
        brsh    K324
        ldi     R25     ,   0b00010011      ; -11 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K324:
        cpi     R24     ,   75
```

```
        brsh    K325
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K325:
        cpi     R24     ,   78
        brsh    K326
        ldi     R25     ,   0b11001000      ; - 8 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K326:
        ldi     R25     ,   0b10101111      ; - 7 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K327
K327:
        ret

;-----------------------------------------------------------------------------------------
; Korrektur-Berechnung des Meßbereichs 250 VAC
;
K400:
        cpi     R24     ,   1
        brsh    K401
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K401:
        cpi     R24     ,   2
        brsh    K402
        ldi     R25     ,   0b00000000      ;   0 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K402:
        cpi     R24     ,   5
        brsh    K403
        ldi     R25     ,   0b00001101      ; -21 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K403:
        cpi     R24     ,   8
        brsh    K404
        ldi     R25     ,   0b01011000      ; -24 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K404:
        cpi     R24     ,   11
        brsh    K405
        ldi     R25     ,   0b01011000      ; -24 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K405:
        cpi     R24     ,   14
        brsh    K406
        ldi     R25     ,   0b01011000      ; -24 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K406:
        cpi     R24     ,   18
        brsh    K407
        ldi     R25     ,   0b01011000      ; -24 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K407:
        cpi     R24     ,   21
        brsh    K408
        ldi     R25     ,   0b00111111      ; -23 Digits
        ldi     R26     ,   0b00000010
```

```
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K408:
        cpi     R24     ,   24
        brsh    K409
        ldi     R25     ,   0b00100110      ; -22 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K409:
        cpi     R24     ,   27
        brsh    K410
        ldi     R25     ,   0b00100110      ; -22 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K410:
        cpi     R24     ,   30
        brsh    K411
        ldi     R25     ,   0b00001101      ; -21 Digits
        ldi     R26     ,   0b00000010
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K411:
        cpi     R24     ,   33
        brsh    K412
        ldi     R25     ,   0b11110100      ; -20 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K412:
        cpi     R24     ,   37
        brsh    K413
        ldi     R25     ,   0b11110100      ; -20 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K413:
        cpi     R24     ,   40
        brsh    K414
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K414:
        cpi     R24     ,   43
        brsh    K415
        ldi     R25     ,   0b11011011      ; -19 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K415:
        cpi     R24     ,   46
        brsh    K416
        ldi     R25     ,   0b11000010      ; -18 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K416:
        cpi     R24     ,   49
        brsh    K417
        ldi     R25     ,   0b10101001      ; -17 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K417:
        cpi     R24     ,   53
        brsh    K418
        ldi     R25     ,   0b10010000      ; -16 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K418:
        cpi     R24     ,   56
        brsh    K419
        ldi     R25     ,   0b01110111      ; -15 Digits
```

```
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K419:
        cpi     R24     ,   59
        brsh    K420
        ldi     R25     ,   0b01011110      ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K420:
        cpi     R24     ,   62
        brsh    K421
        ldi     R25     ,   0b01011110      ; -14 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K421:
        cpi     R24     ,   65
        brsh    K422
        ldi     R25     ,   0b01000101      ; -13 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K422:
        cpi     R24     ,   69
        brsh    K423
        ldi     R25     ,   0b00101100      ; -12 Digits
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K423:
        cpi     R24     ,   72
        brsh    K424
        ldi     R25     ,   0b00010011      ; -11 Digit
        ldi     R26     ,   0b00000001
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K424:
        cpi     R24     ,   75
        brsh    K425
        ldi     R25     ,   0b11111010      ; -10 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K425:
        cpi     R24     ,   78
        brsh    K426
        ldi     R25     ,   0b11100001      ; - 9 Digits
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K426:
        ldi     R25     ,   0b11001000      ; - 8 Digit
        ldi     R26     ,   0b00000000
        add     R17     ,   R25
        adc     R18     ,   R26
        rjmp    K427
K427:
        ret

;-----------------------------------------------------------------------------------
; ASCII-Wert von R24 in LCD_line2 anzeigen
;
K500:
        push    R17
        ldi     R17     ,   48
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'L'
        rcall   LCD_data
        ldi     R16     ,   'I'
        rcall   LCD_data
        ldi     R16     ,   'B'
```

```
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '4'
        rcall   LCD_data
        ldi     R16     ,   '='
        rcall   LCD_data
        ldi     R16     ,   77
        rcall   LCD_goto
        mov     R16     ,   R24
        rcall   LCD_number
        ldi     R16     ,   20
        rcall   Wait
        pop     R17
        ret

;-----------------------------------------------------------------------------------------
; Dezimalzahl aus einer Binärzahl mit 16 Bit ermitteln
;
D1:
        mov     R1      ,   R18
        mov     R0      ,   R17

        clr     R2
        clr     R3
        clr     R4
        clr     R5
        sub     R6      ,   R6
        tst     R0
        cpc     R1      ,   R2
        breq    D11
        ldi     R16     ,   9
        ldi     R17     ,   17
D12:
        dec     R17
        lsl     R0
        rol     R1
        brcc    D12
D13:
        adc     R2      ,   R2
        cp      R16     ,   R2
        brcc    D14
        sbc     R2      ,   R16
        sec
D14:
        adc     R3      ,   R3
        cp      R16     ,   R3
        brcc    D15
        sbc     R3      ,   R16
        sec
D15:
        adc     R4      ,   R4
        cp      R16     ,   R4
        brcc    D16
        sbc     R4      ,   R16
        sec
D16:
        adc     R5      ,   R5
        cp      R16     ,   R5
        brcc    D17
        sbc     R5      ,   R16
        sec
D17:
        adc     R6      ,   R6
        lsl     R0
        rol     R1
        dec     R17
        brne    D13
D11:
        push    R16
        mov     R16     ,   R6          ;
        cpi     R16     ,   2           ; "OVERFLOW" in LCD_line2
        breq    D20                     ; in case of 256xx or higher or
        rjmp    D26                     ; in case of 25575
D20:                                    ; (2 ^ 10 - 1) * 25 = 25575
        mov     R16     ,   R5          ;
        cpi     R16     ,   5           ;
        breq    D21                     ;
        rjmp    D26                     ;
```

```
D21:                                            ;
        mov     R16     ,   R4                  ;
        cpi     R16     ,   6                   ; kann nur bei den berechneten
        brsh    D25                             ; Werten vorkommen, also nur bei
        rjmp    D22                             ; AC-Messungen
D22:                                            ;
        mov     R16     ,   R4                  ;
        cpi     R16     ,   5
        breq    D23
        rjmp    D26
D23:
        mov     R16     ,   R3
        cpi     R16     ,   7
        breq    D24
        rjmp    D26
D24:
        mov     R16     ,   R2
        cpi     R16     ,   5
        breq    D25
        rjmp    D26
D25:
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'F'
        rcall   LCD_data
        ldi     R16     ,   'L'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'W'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   10
        rcall   Wait
        pop     R16
        ret
D26:
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
```

```
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        pop     R16
        ret

;-----------------------------------------------------------------------------------
; Zeichen ins CGRAM des LCD schreiben
;
CGRam:
        cpi     R27     ,   0           ; BAR GRAPH: NONE
        breq    CGRam0
        cpi     R27     ,   1           ; BAR GRAPH TYPE 1 "||||| ||||| ||||| |||"
        breq    CGRam1
        cpi     R27     ,   2           ; BAR GRAPH TYPE 2 "| | | | | | | | | |"
        breq    CGRam2
CGRam0:
        ldi     ZL      ,   LOW(Symbole0*2)
        ldi     ZH      ,   HIGH(Symbole0*2)
        rjmp    CGRam3
CGRam1:
        ldi     ZL      ,   LOW(Symbole1*2)
        ldi     ZH      ,   HIGH(Symbole1*2)
        rjmp    CGRam3
CGRam2:
        ldi     ZL      ,   LOW(Symbole2*2)
        ldi     ZH      ,   HIGH(Symbole2*2)
        rjmp    CGRam3
CGRam3:
        ldi     R16     ,   64
        rcall   LCD_cmd
CGRam4:
        lpm     R16     ,   Z+
        cpi     R16     ,   253
        breq    CGRam5
        rcall   LCD_data
        rjmp    CGRam4
CGRam5:
        ret

;-----------------------------------------------------------------------------------
; Zeichen fürs CGRAM des LCD, Bargraph alle Zeichen leer

Symbole0:
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,0,0,0,0,0,253,0

;-----------------------------------------------------------------------------------
; Zeichen fürs CGRAM des LCD, Bargraph mit Leerstellen zwischen den Zeichen

Symbole1:
        .db 0,0,0,0,0,0,0,0
        .db 0,0,0,16,16,16,0,0
        .db 0,0,0,24,24,24,0,0
        .db 0,0,0,28,28,28,0,0
        .db 0,0,0,30,30,30,0,0
        .db 0,0,0,31,31,31,0,0,253,0

;-----------------------------------------------------------------------------------
; Zeichen fürs CGRAM des LCD, Bargraph mit Leerstallen von Bar zu Bar

Symbole2:
        .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 0,0,0,21,21,21,0,0,253,0

;-----------------------------------------------------------------------------------
; Timer/Counter0 Kontrolle

onTMR0:
        sbis    PINC    ,   0
        rjmp    Taste
        sbis    PINC    ,   2
        rcall   Taste5
        reti

;-----------------------------------------------------------------------------------
; Mode Of The Display Line 2
```

```
                                        ; R27 = 0: CLCD_line2 ist leer
                                        ; R27 = 1: Bar Graph Type 1
                                        ; R27 = 2: Bar Graph Type 2
                                        ; R27 = 3: Calibration Mode
Taste5:
        ldi     R27     ,   0
        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
Taste510:
        rcall   LCD_line2
        ldi     R16     ,   'B'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'G'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'P'
        rcall   LCD_data
        ldi     R16     ,   'H'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'N'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'N'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   100             ; entprellen und anzeigen
        rcall   Wait
        ldi     R27     ,   0
        sbis    PINC    ,   2
        rjmp    Taste511
        rcall   CGRam
        reti
Taste511:
        rcall   LCD_line2
        ldi     R16     ,   'B'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'R'
```

```
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'G'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'P'
        rcall   LCD_data
        ldi     R16     ,   'H'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'Y'
        rcall   LCD_data
        ldi     R16     ,   'P'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '1'
        rcall   LCD_data
        ldi     R16     ,   100             ; entprellen und anzeigen
        rcall   Wait
        ldi     R27     ,   1
        sbis    PINC    ,   2
        rjmp    Taste512
        rcall   CGRam
        reti
Taste512:
        rcall   LCD_line2
        ldi     R16     ,   'B'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'G'
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'P'
        rcall   LCD_data
        ldi     R16     ,   'H'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'Y'
        rcall   LCD_data
        ldi     R16     ,   'P'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   100             ; entprellen und anzeigen
        rcall   Wait
        ldi     R27     ,   2
        sbis    PINC    ,   2
        rjmp    Taste513
        rcall   CGRam
        reti
Taste513:
        rcall   LCD_line2
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'L'
        rcall   LCD_data
        ldi     R16     ,   'I'
        rcall   LCD_data
        ldi     R16     ,   'B'
        rcall   LCD_data
```

```
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'I'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'N'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'M'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'E'
        rcall   LCD_data
        ldi     R16     ,   100             ; entprellen und anzeigen
        rcall   Wait
        ldi     R27     ,   3
        sbis    PINC    ,   2
        rjmp    Taste510
        reti

;-------------------------------------------------------------------------------------
; Funktionen mittels der grünen Taste aufrufen

; R7 =  1: DC Range AUTO
; R7 =  2: DC Range 0.25 V
; R7 =  3: DC Range 2.50 V
; R7 =  4: DC Range 25.0 V
; R7 =  5: DC Range 250  V
; R7 =  6: AC Range AUTO
; R7 =  7: AC Range 0.25 V
; R7 =  8: AC Range 2.50 V
; R7 =  9: AC Range 25.0 V
; R7 = 10: AC Range 250  V

Taste:
        cli
        push    R16
        push    R17
        push    R18
        push    R19
        ldi     R16     ,   0
        ldi     R17     ,   0
        ldi     R18     ,   0
        ldi     R19     ,   0
        rcall   wait5ms                     ; entprellen
        rcall   wait5ms
        rcall   LCD_line1
        ldi     R16     ,   'S'             ; ASKCII 83
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   'l'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   'c'
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   't'
        rcall   LCD_data
        ldi     R16     ,   'h'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
```

```
        rcall   LCD_data


        rcall   LCD_line2
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   50              ; 500 ms warten damit die Taste
        rcall   Wait                        ; wieder losgelassen wurde
Taste30:
        inc     R18                         ; wenn innerhalb von 2 s kein
        ldi     R16     ,   1               ; Tastendruck erfolgt, dann wird
        rcall   Wait                        ; die zuletzt gewählte Funktion
        cpi     R18     ,   200             ; übernommen
        breq    Taste31
        sbic    PINC    ,   0               ; warten auf Tastendruck
        rjmp    Taste30
        rjmp    Taste00
Taste31:
        pop     R19
        pop     R18
        pop     R17
        pop     R16
        sei
        rjmp    Loop
Taste00:
        inc     R19
        cpi     R19     ,   1
        breq    Taste01
        cpi     R19     ,   2
        breq    Taste02
        cpi     R19     ,   3
        breq    Taste03
        cpi     R19     ,   4
        breq    Taste04
        cpi     R19     ,   5
        breq    Taste05
        cpi     R19     ,   6
        breq    Taste06
        cpi     R19     ,   7
        breq    Taste07
        cpi     R19     ,   8
        breq    Taste08
        cpi     R19     ,   9
        breq    Taste09
        cpi     R19     ,   10
        breq    Taste10
Taste01:
        rjmp    Taste11
Taste02:
        rjmp    Taste12
Taste03:
        rjmp    Taste13
Taste04:
        rjmp    Taste14
Taste05:
        rjmp    Taste15
Taste06:
```

```
        rjmp    Taste21
Taste07:
        rjmp    Taste22
Taste08:
        rjmp    Taste23
Taste09:
        rjmp    Taste24
Taste10:
        rjmp    Taste25
Taste11:
        rcall   LCD_line2
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   1
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste12:
        rcall   LCD_line2
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '0'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   2
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
```

```
Taste13:
        rcall   LCD_line2
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   3
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste14:
        rcall   LCD_line2
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   4
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste15:
        rcall   LCD_line2
        ldi     R16     ,   'D'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
```

```
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   '0'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   5
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste21:
        rcall   LCD_line2
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'U'
        rcall   LCD_data
        ldi     R16     ,   'T'
        rcall   LCD_data
        ldi     R16     ,   'O'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   6
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste22:
        rcall   LCD_line2
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
```

```
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '0'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   7
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste23:
        rcall   LCD_line2
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '.'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   8
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste24:
        rcall   LCD_line2
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   ' '
```

```
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   9
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
Taste25:
        rcall   LCD_line2
        ldi     R16     ,   'A'
        rcall   LCD_data
        ldi     R16     ,   'C'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'R'
        rcall   LCD_data
        ldi     R16     ,   'a'
        rcall   LCD_data
        ldi     R16     ,   'n'
        rcall   LCD_data
        ldi     R16     ,   'g'
        rcall   LCD_data
        ldi     R16     ,   'e'
        rcall   LCD_data
        ldi     R16     ,   ':'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   '2'
        rcall   LCD_data
        ldi     R16     ,   '5'
        rcall   LCD_data
        ldi     R16     ,   '0'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   'V'
        rcall   LCD_data
        ldi     R16     ,   ' '
        rcall   LCD_data
        ldi     R16     ,   10
        mov     R7      ,   R16
        ldi     R16     ,   50
        rcall   wait
        rjmp    Taste30
```