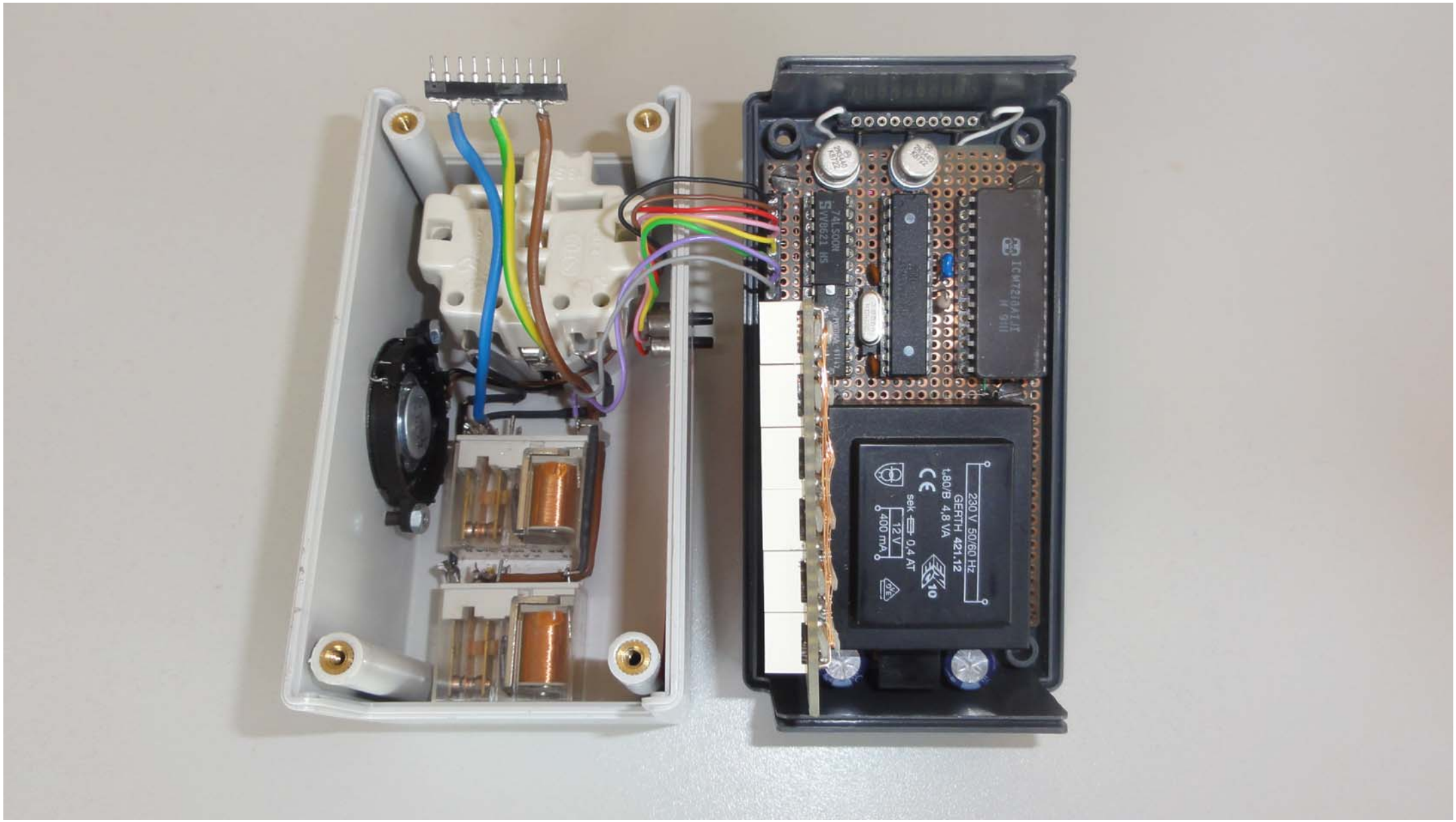
CountDown, Gerät in Funktion

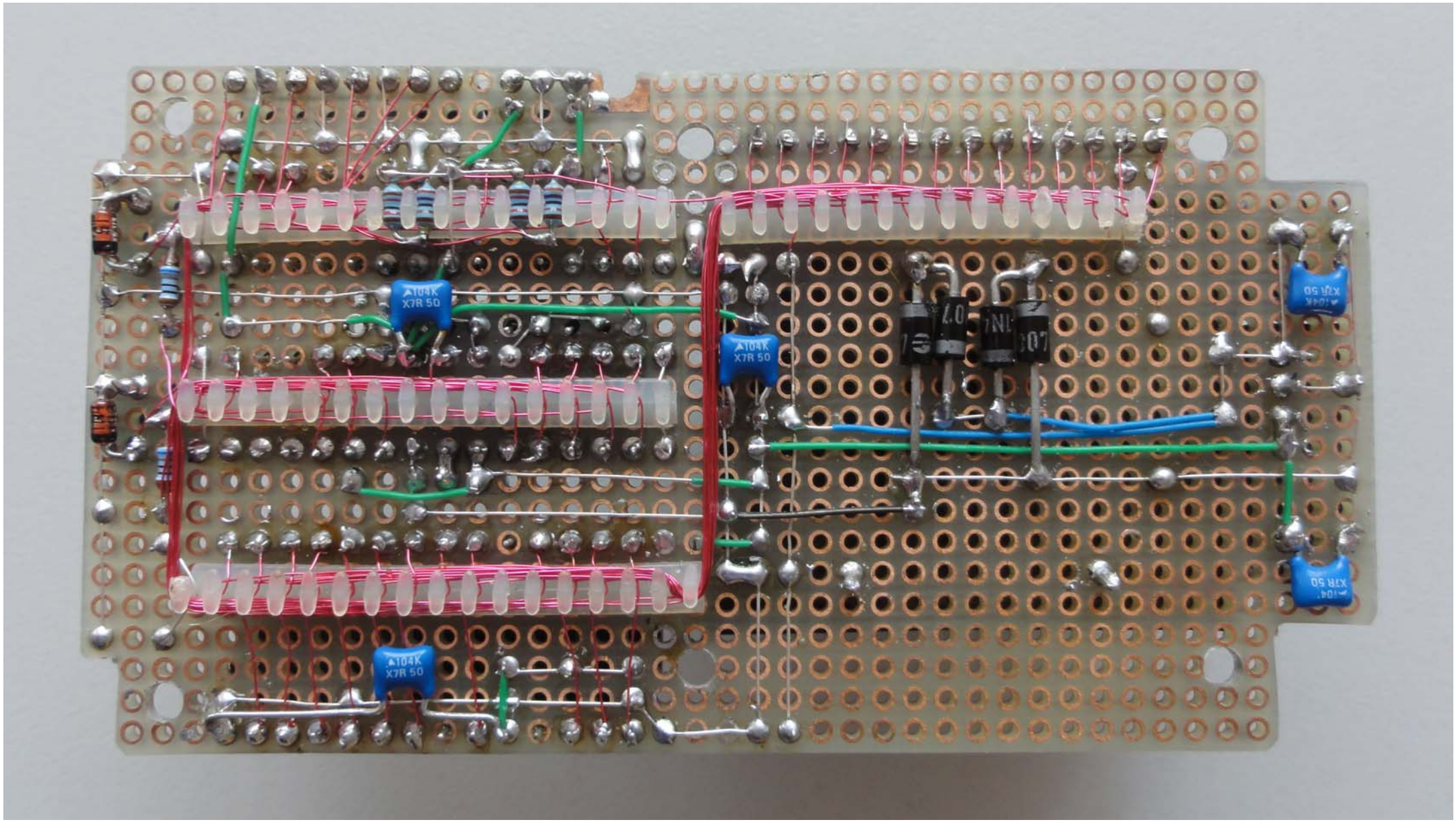
DSC21521.JPG

CountDown, Stecker- und Steckdosenteil



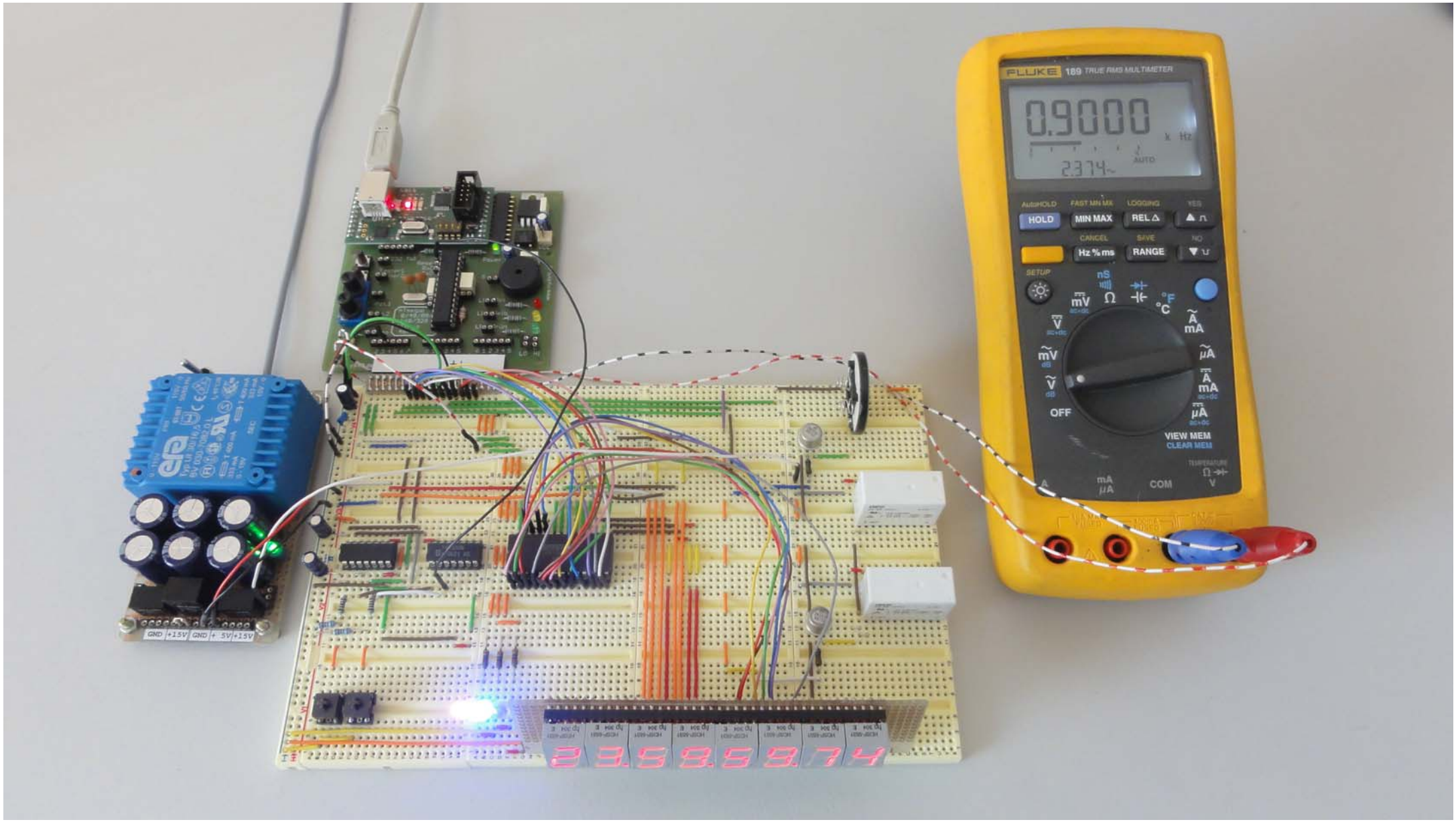
DSC21600.JPG

CountDown, Verdrahtung



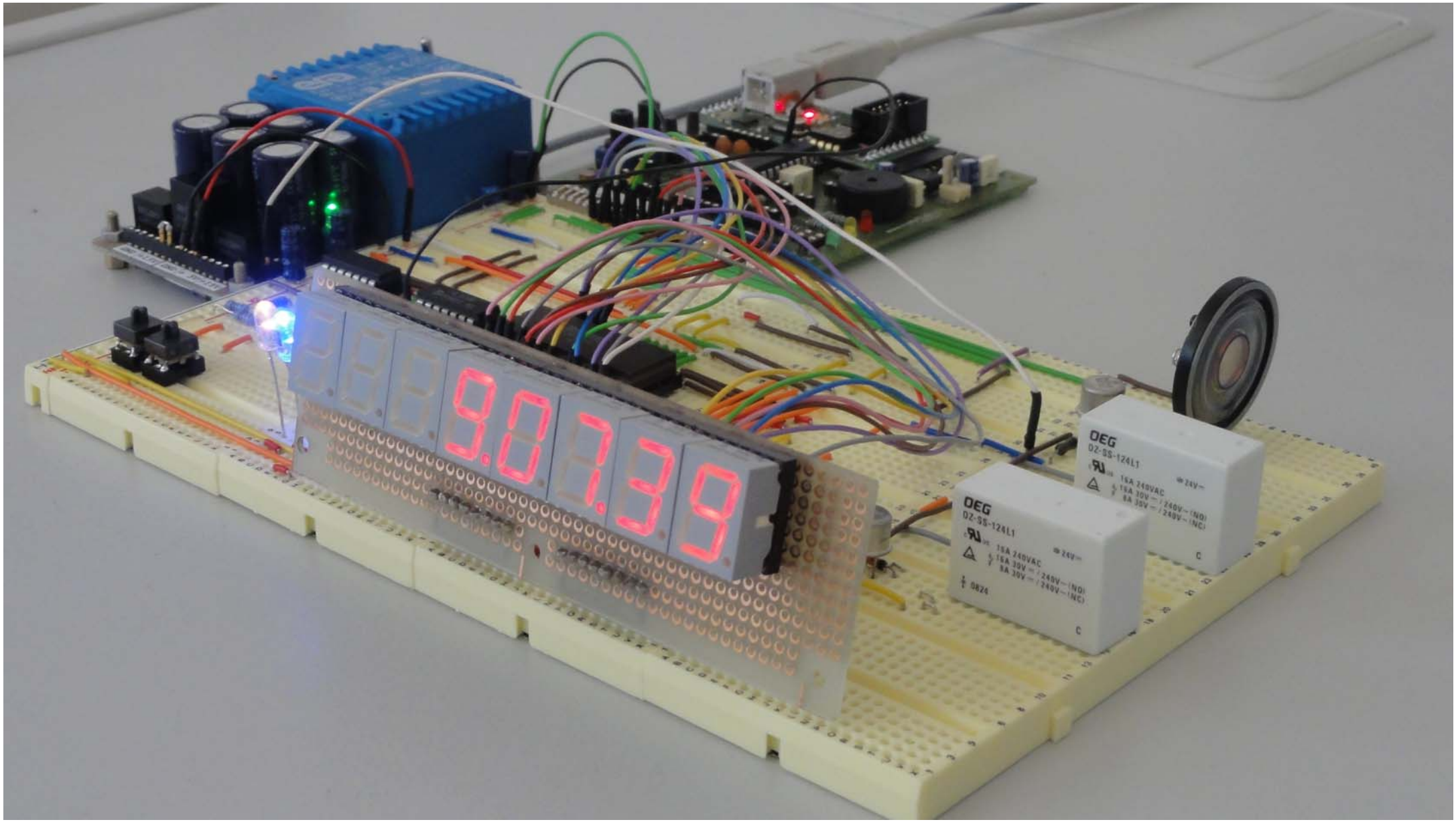
DSC21530.JPG

CountDown, Versuchsaufbau 1

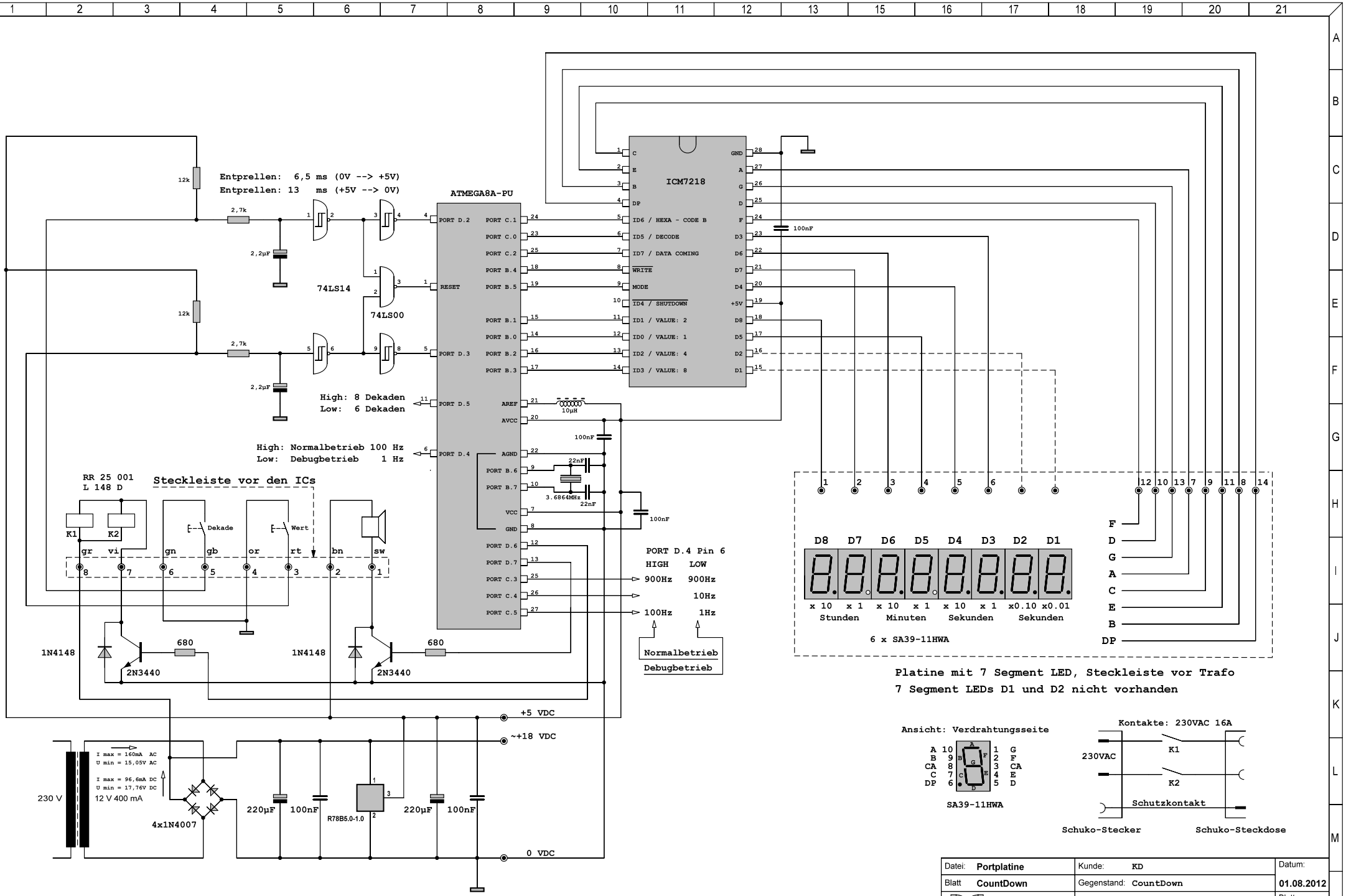


DSC21561.JPG

CountDown, Versuchsaufbau 2



DSC21598.JPG


Entprellen: 6,5 ms (0V --> +5V)
Entprellen: 13 ms (+5V --> 0V)
ATMEGA8A-PU
ICM7218
74LS14
74LS00
High: 8 Dekaden
Low: 6 Dekaden
High: Normalbetrieb 100 Hz
Low: Debugbetrieb 1 Hz
RR 25 001
L 148 D
Steckleiste vor den ICs
Dekade
Wert
K1
K2
gr vi gn gb or rt bn sw
1N4148
2N3440
680
PORT D.4 Pin 6
HIGH LOW
900Hz 900Hz
10Hz
100Hz 1Hz
Normalbetrieb
Debugbetrieb
D8 D7 D6 D5 D4 D3 D2 D1
x 10 x 1 x 10 x 1 x 10 x 1 x0.10 x0.01
Stunden Minuten Sekunden Sekunden
6 x SA39-11HWA
F D G A C E B DP
Platine mit 7 Segment LED, Steckleiste vor Trafo
7 Segment LEDs D1 und D2 nicht vorhanden
Ansicht: Verdrahtungsseite
SA39-11HWA
Kontakte: 230VAC 16A
230VAC
K1
K2
Schutzkontakt
Schuko-Stecker
Schuko-Steckdose
230 V
12 V 400 mA
4x1N4007
220µF
100nF
R78B5.0-1.0
+5 VDC
~+18 VDC
0 VDC
Datei: Portplatine
Blatt CountDown
Kunde: KD
Gegenstand: CountDown
Maschine:
Zeichn.-Nr.: Portplatine
Datum: 01.08.2012
Blatt: 9
von: 9
rodo38

```
;-----------------------------------------------------------------------------------------------
; Title         :   CountDown\1_Main.asm
;-----------------------------------------------------------------------------------------------
; Funktion      :   Das Gerät verbindet einen Verbraucher für eine einstellbare Zeit mit dem 230 VAC Netz.
;               :   Geschaltet werden die Leitungen L und N mit 2 24 VDC Relais mit 16 A 230 VAC Kontakten.
;               :   Bedienung:
;               :   Beide Tasten drücken:  Die Dekade "Stunden x 10" beginnt zu blinken. Wert: 0
;               :   Taste 1:               Auf die nächste Dekade umschalten, welche dann blinkt.
;               :   Taste 2:               Wert in der blinkenden Dekade ändern.
;               :   Betrieb mit 8 Dekaden: Maximal eingebbarer Wert: 23:59:59.99  [HH:MM:SS.ss]
;               :   Betrieb mit 6 Dekaden: Maximal eingebbarer Wert: 23:59:59     [HH:MM:SS]
;               :   Taste 1:               Nach der Eingabe des Wertes in die letzte Dekade:
;               :                          Die Relais werden eingeschaltet.
;               :                          Die gewählte Zeit wird mit 100 Hz zurückgezählt.
;               :                          Weil diese Frequenz aus der Quarzfrequenz gewonnen wird,
;               :                          wird die Zeit sehr präzise eingehalten.
;               :                          Wenn 00:00:00.00 erreicht ist, werden die Relais ausgeschaltet
;               :                          und ein kleiner Lautsprecher summt für ca. 2 Sekunden.
;               :   Während des Rückzähl-Vorgangs werden führende Nullen ausgeblendet.
;               :   Das gilt nicht für die Dekaden "Sekunden x 1", "Sekunden x 0.1" und "Sekunden x 0.01"
;               :   Weil die 7 Segment Anzeigen keinen Doppelpunkt haben, wird der Dezimalpunkt angezeigt.
;               :   Dezimalpunkt jeweils bei Dekade:   "Stunden  x 1"
;               :                                      "Minuten  x 1"
;               :                                      "Sekunden x 1" (nur bei Betrieb mit 8 Dekaden)
;               :   Wenn eine dieser Dekaden ausgeblendet wird, weil es sich um eine führende Null handelt,
;               :   dann wird auch der Dezimalpunkt dieser Dekade ausgeblendet.
;               :
; Anzeige       :   IC ICM7218A. Das ist ein Treiber für 7 Segment + DP LED-Anzeigen mit gemeinsamer Anode.
;               :   In dieser Version werden dafür 9 Output Pins des Mikrocontrollers benötigt.
;               :   Hinweis:
;               :   Das Datenblatt des ICM7218 fordert Pausen- bzw. Signal-Haltezeiten.
;               :   Diese werden offenbar eingehalten, ohne dass sie programmiert werden müssen.
;               :
; Schaltung     :   Schaltplan "CountDown.spl7"
;               :
; Ports und     :   Ports          ATMEGA8A       ICM7218A
; ICM7218A      :
;               :   B.0 Output      Pin 14        Pin 12 = ID 0  HEX 0 = 1
;               :   B.1 Output      Pin 15        Pin 11 = ID 1  HEX 1 = 2
;               :   B.2 Output      Pin 16        Pin 13 = ID 2  HEX 2 = 4
;               :   B.3 Output      Pin 17        Pin 14 = ID 3  HEX 3 = 8
;               :
;               :   B.4 Output      Pin 18        Pin 08 = High: Input Not Loaded
;               :                                 Pin 08 = Low : Input    Loaded
;               :
;               :   B.5 Output      Pin 19        Pin 09 = High: Load Control Data on /WRITE LOW Pulse
;               :                                 Pin 09 = Low : Load Input   Data on /WRITE LOW Pulse
;               :
;               :   C.0 Output      Pin 23        Pin 05 = High: Hexadecimal Decoding
;               :                                 Pin 05 = Low : Code B     Decoding
;               :
;               :   C.1 Output      Pin 24        Pin 06 = High: No Decode
;               :                                 Pin 06 = Low :    Decode
;               :
;               :   C.2 Output      Pin 25        Pin 07 = High:   Data Coming
;               :                                 Pin 07 = Low : No Data Coming + Decimal Point
;               :
;               :   unbenutzt                     Pin 10 = High: NO SHUTDOWN
;               :                                 Pin 10 = Low :    SHUTDOWN
;               :
;               :
; Ports         :                                 Normalbetrieb          Debugbetrieb
;               :
;               :   C.3 Output      Pin 26        900 Hz                 900 Hz (LED Anzeige)
;               :   C.4 Output      Pin 27                                10 Hz (LED Anzeige)
;               :   C.5 Output      Pin 28        100 Hz                   1 Hz (LED Anzeige)
;               :
;               :
;               :   D.2 Input       Pin  4        IC1 (74LS14) "onInt0" (Taste 2)
;               :   D.3 Input       Pin  5        IC1 (74LS14) "onInt1" (Taste 1)
;               :
;               :   D.4 Input       Pin  6        High: 100 Hz, Low: 1 Hz (zum Debuggen)
;               :
;               :   D.5 Input       Pin 11        High: Betrieb mit 8 Dekaden
;               :                                 Dezimalpunkte hinter          Stunden  x 1
;               :                                                               Minuten  x 1
;               :                                                               Sekunden x 1
;               :
;               :                                 Low:  Betrieb mit 6 Dekaden, ohne  Sekunden x 0.10
;               :                                                              und ohne  Sekunden x 0.01
;               :
;               :                                       Dezimalpunkte hinter          Stunden  x 1
;               :                                                                     Minuten  x 1
;               :
;               :   D.6 Output      Pin 12        Relais
```

```
;               :   D.7 Output      Pin 13        Lautsprecher
;               :
;-------------------------------------------------------------------------------------------------------
; Prozessor     :   ATmega8
; Takt          :   3,6864 MHz
; Sprache       :   Assembler
; Datum         :   15.08.2012
; Version       :   1.0
; Autor         :   rodo38
;-------------------------------------------------------------------------------------------------------
        .include "m8def.inc"
;-------------------------------------------------------------------------------------------------------
; Variable definieren

        .def    mCoun1  =    R0
        .def    mCoun2  =    R1
        .def    mCoun3  =    R2
        .def    mCoun4  =    R3
        .def    mCoun5  =    R4
        .def    mCoun6  =    R5
        .def    mCoun7  =    R6
        .def    mCoun8  =    R7
        .def    mTimr1  =    R8
        .def    mTimr2  =    R9
        .def    mTimr3  =   R10
        .def    mTimr4  =   R11
        .def    mTimr5  =   R12

        .def    Temp    =   R16
        .def    Time    =   R17
        .def    Contr0  =   R18
        .def    Contr1  =   R19
        .def    Number  =   R20
        .def    Help00  =   R21
        .def    Freq00  =   R22
        .def    Temp01  =   R23
        .def    Help01  =   R24

;-------------------------------------------------------------------------------------------------------
; Reset and Interrupt Vector              Beschreibung

Begin:  rjmp    Main                    ;  1 POWER ON RESET
        rjmp    onInt0                  ;  2 Int0-Interrupt
        rjmp    onInt1                  ;  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    onTc0                   ; 10 TC0 Overflow
        reti                            ; 11 SPI, STC Serial Transfer Complete
        reti                            ; 12 UART Rx complete
        reti                            ; 13 UART Data Register Empty
        reti                            ; 14 UART Tx complete
        reti                            ; 15 ADC Conversion Complete
        reti                            ; 16 EEPROM Ready
        reti                            ; 17 Analog Comparator
        reti                            ; 18 TWI (I²C) Serial Interface
        reti                            ; 19 Store Program Memory Redy

;-------------------------------------------------------------------------------------------------------
; Start, Power ON, Reset

Main:
        ldi     Temp    ,   LOW (RAMEND)    ; für  Stackpointer LOW
        out     SPL     ,   Temp            ; INIT Stackpointer LOW

        ldi     Temp    ,   HIGH(RAMEND)    ; für  Stackpointer HIGH
        out     SPH     ,   Temp            ; INIT Stackpointer HIGH

        ldi     Temp    ,   0b00001111      ; Bit 2 = 1, Bit 3 = 1: steigende Flanke erzeugt Interrupt
        out     MCUCR   ,   Temp            ;

        ldi     Temp    ,   0b11000000      ; Bit 6 = Int0 aktiv
        out     GICR    ,   Temp            ; Bit 7 = Int1 aktiv

        ldi     Temp    ,   0b00000010      ; Bit 1 = 1: Prescaler 8
        out     TCCR0   ,   Temp            ;

        ldi     Temp    ,   0b00000000      ; Bit 0 = 1: Timer0 Overflow
        out     TIMSK   ,   Temp            ; Bit 0 = 0: Timer nicht aktiv

        ldi     Temp    ,   0b00111111      ; B.0 bis B.5 = Output
        out     DDRB    ,   Temp            ;
```

```
        ldi     Temp    ,   0b00111111      ; C.0 bis C.5 = Output
        out     DDRC    ,   Temp            ;
        ldi     Temp    ,   0b11000000      ; D.2 bis D.5 = Input
        out     DDRD    ,   Temp            ; D.6 und D.7 = Output
        ldi     Temp    ,   0b00111100      ; D.2 bis D.5 = Pullup
        out     PORTD   ,   Temp            ;
        ldi     Temp    ,   0b00000000      ; PORTB und PORTC: Alle Bits = 0
        out     PORTB   ,   Temp            ;
        out     PORTC   ,   Temp            ;

;-----------------------------------------------------------------------------------------------------
; Variable initialisieren

        ldi     Temp    ,   0b00000000
        ldi     Time    ,   0b00000000
        ldi     Contr0  ,   0b00000000
        ldi     Contr1  ,   0b00000000
        ldi     Number  ,   0b00000000
        ldi     Help00  ,   0b00000000
        ldi     Freq00  ,   0b00000000
        ldi     Temp01  ,   0b00000000
        ldi     Help01  ,   0b00000000

        mov     mCoun1  ,   Temp
        mov     mCoun2  ,   Temp
        mov     mCoun3  ,   Temp
        mov     mCoun4  ,   Temp
        mov     mCoun5  ,   Temp
        mov     mCoun6  ,   Temp
        mov     mCoun7  ,   Temp
        mov     mCoun8  ,   Temp

        mov     mTimr1  ,   Temp
        mov     mTimr2  ,   Temp
        mov     mTimr3  ,   Temp
        mov     mTimr4  ,   Temp
        mov     mTimr5  ,   Temp

;-----------------------------------------------------------------------------------------------------
; Interrupt erlauben

        sei

;-----------------------------------------------------------------------------------------------------
; Files includet

        .include "2_Select.asm"
        .include "3_Decades.asm"
        .include "4_Digits.asm"
        .include "5_Timer.asm"
        .include "6_Down.asm"

;---------1---------2---------3---------4---------5---------6---------7---------8---------9---------0------
```

```
;-----------------------------------------------------------------------------------------------------
; CountDown\2_Select.asm
;-----------------------------------------------------------------------------------------------------
; Hauptprogramm

MLoop:
        rjmp    outDek

;-----------------------------------------------------------------------------------------------------
; Eingabe von Werten in die Dekaden
; Interrupt Auslösung durch Drücken und Loslassen der Taste 2, d.h. durch die Flanke 0 V --> + 5 V

onInt0:
        ldi     Contr1  ,   0b00000001
        reti

;-----------------------------------------------------------------------------------------------------
; Weiterschaltung zur nächsten Dekade, nach der letzten Dekade Start des Runterzählens
; Interrupt Auslösung durch Drücken und Loslassen der Taste 1, d.h. durch die Flanke 0 V --> + 5 V

onInt1:
        rjmp    outDek
        reti

;-----------------------------------------------------------------------------------------------------
; Dekade wählen

outDek:
        ldi     Temp    ,   0b11000000      ; Bit 6 = 1: Int0 aktiv
        out     GICR    ,   Temp            ; Bit 7 = 1: Int1 aktiv
Dek_0:
        inc     Contr0
        cpi     Contr0  ,   0b00000001      ; Stunden  x 10
        breq    Dek_1
        cpi     Contr0  ,   0b00000010      ; Stunden  x  1
        breq    Dek_2
        cpi     Contr0  ,   0b00000011      ; Minuten  x 10
        breq    Dek_3
        cpi     Contr0  ,   0b00000100      ; Minuten  x  1
        breq    Dek_4
        cpi     Contr0  ,   0b00000101      ; Sekunden x 10
        breq    Dek_5
        cpi     Contr0  ,   0b00000110      ; Sekunden x  1
        breq    Dek_6
        sbis    PIND    ,   5               ; D.5 = Low:  Betrieb mit 6 Dekaden: [HH:MM:SS]
                                            ; D.5 = High: Betrieb mit 8 Dekaden: [HH:MM:SS.ss]
        rjmp    Dek_9
        cpi     Contr0  ,   0b00000111      ; Sekunden x  0.10
        breq    Dek_7
        cpi     Contr0  ,   0b00001000      ; Sekunden x  0.01
        breq    Dek_8
        cpi     Contr0  ,   0b00001001      ; Neueingabe oder Count Down Beginn
        breq    Dek_9
Dek_1:
        rjmp    Dek01
Dek_2:
        rjmp    Dek02
Dek_3:
        rjmp    Dek03
Dek_4:
        rjmp    Dek04
Dek_5:
        rjmp    Dek05
Dek_6:
        rjmp    Dek06
Dek_7:
        rjmp    Dek07
Dek_8:
        rjmp    Dek08
Dek_9:
;       sbi     PORTD   ,   6               ; Relais: Wischer 10 ms
;       ldi     Time    ,   0b00000001      ; dient zum Starten der Stoppuhr.
;       rcall   Wait                        ; Aktivierung:
;       cbi     PORTD   ,   6               ; durch Beseitigung der Semikolons am Anfang der Zeilen.

        sei                                 ; Interrupt Ein
        ldi     Temp    ,   0b00000000      ; Bit 6 = 0: Int0 nicht aktiv
        out     GICR    ,   Temp            ; Bit 7 = 0: Int1 nicht aktiv
        ldi     Temp    ,   0b00000010      ; Bit 1 = 1: Prescaler 8
        out     TCCR0   ,   Temp            ;
        ldi     Temp    ,   0b00000001      ; Bit 0 = 1: Timer0 Overflow
        out     TIMSK   ,   Temp            ; Timer0 aktiv
Dek_:
        rjmp    Dek_
```

```
;-----------------------------------------------------------------------------------------------------
; Stunden x 10
; gewählt werden können 0, 10 oder 20 Stunden

Dek01:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek10
        brne    Dek14
Dek10:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun1
        cpi     Temp    ,   0b00000000
        breq    Dek11
        cpi     Temp    ,   0b00000001
        breq    Dek12
        cpi     Temp    ,   0b00000010
        breq    Dek13
        rjmp    Dek14
Dek11:
        ldi     Temp    ,   0b00000001
        mov     mCoun1  ,   Temp
        rjmp    Dek14
Dek12:
        ldi     Temp    ,   0b00000010
        mov     mCoun1  ,   Temp
        rjmp    Dek14
Dek13:
        ldi     Temp    ,   0b00000000
        mov     mCoun1  ,   Temp
        rjmp    Dek14
Dek14:
        rcall   Dunk01
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell01
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek01

;-----------------------------------------------------------------------------------------------------
; Stunden x 1
; wenn bei "Stunden x 10" 0 oder 10 Stunden gewählt wurde, können 0 bis 9        Stunden gewählt werden
; wenn bei "Stunden x 10"        20 Stunden gewählt wurde, können 0, 1, 2 oder 3 Stunden gewählt werden
; die längste wählbare Zeit soll 23:59:59.99 [HH:MM:SS.ss] sein, jedoch muß z.B. 19:59:59.99 möglich sein

Dek02:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek200
        brne    Dek211
Dek200:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun2
        cpi     Temp    ,   0b00000000
        breq    Dek201
        cpi     Temp    ,   0b00000001
        breq    Dek202
        cpi     Temp    ,   0b00000010
        breq    Dek203

        push    Temp
        mov     Temp    ,   mCoun1
        cpi     Temp    ,   0b00000010
        breq    Dek210
        pop     Temp

        cpi     Temp    ,   0b00000011
        breq    Dek204
        cpi     Temp    ,   0b00000100
        breq    Dek205
        cpi     Temp    ,   0b00000101
        breq    Dek206
        cpi     Temp    ,   0b00000110
        breq    Dek207
        cpi     Temp    ,   0b00000111
        breq    Dek208
        cpi     Temp    ,   0b00001000
        breq    Dek209
        cpi     Temp    ,   0b00001001
        breq    Dek210
        rjmp    Dek211
Dek201:
        ldi     Temp    ,   0b00000001
        mov     mCoun2  ,   Temp
```

```
        rjmp    Dek211
Dek202:
        ldi     Temp    ,   0b00000010
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek203:
        ldi     Temp    ,   0b00000011
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek204:
        ldi     Temp    ,   0b00000100
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek205:
        ldi     Temp    ,   0b00000101
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek206:
        ldi     Temp    ,   0b00000110
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek207:
        ldi     Temp    ,   0b00000111
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek208:
        ldi     Temp    ,   0b00001000
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek209:
        ldi     Temp    ,   0b00001001
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek210:
        ldi     Temp    ,   0b00000000
        mov     mCoun2  ,   Temp
        rjmp    Dek211
Dek211:
        rcall   Dunk02
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell02
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek02

;-----------------------------------------------------------------------------------------------------
; Minuten x 10
; gewählt werden können 0, 10, 20, 30, 40 oder 50 Minuten

Dek03:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek30
        brne    Dek37
Dek30:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun3
        cpi     Temp    ,   0b00000000
        breq    Dek31
        cpi     Temp    ,   0b00000001
        breq    Dek32
        cpi     Temp    ,   0b00000010
        breq    Dek33
        cpi     Temp    ,   0b00000011
        breq    Dek34
        cpi     Temp    ,   0b00000100
        breq    Dek35
        cpi     Temp    ,   0b00000101
        breq    Dek36
        rjmp    Dek37
Dek31:
        ldi     Temp    ,   0b00000001
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek32:
        ldi     Temp    ,   0b00000010
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek33:
        ldi     Temp    ,   0b00000011
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek34:
        ldi     Temp    ,   0b00000100
```

```
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek35:
        ldi     Temp    ,   0b00000101
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek36:
        ldi     Temp    ,   0b00000000
        mov     mCoun3  ,   Temp
        rjmp    Dek37
Dek37:
        rcall   Dunk03
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell03
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek03

;-----------------------------------------------------------------------------------------------------------
; Minuten x 1
; gewählt werden können 0, 1, 2 usw. bis 9 Minuten

Dek04:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek400
        brne    Dek411
Dek400:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun4
        cpi     Temp    ,   0b00000000
        breq    Dek401
        cpi     Temp    ,   0b00000001
        breq    Dek402
        cpi     Temp    ,   0b00000010
        breq    Dek403
        cpi     Temp    ,   0b00000011
        breq    Dek404
        cpi     Temp    ,   0b00000100
        breq    Dek405
        cpi     Temp    ,   0b00000101
        breq    Dek406
        cpi     Temp    ,   0b00000110
        breq    Dek407
        cpi     Temp    ,   0b00000111
        breq    Dek408
        cpi     Temp    ,   0b00001000
        breq    Dek409
        cpi     Temp    ,   0b00001001
        breq    Dek410
        rjmp    Dek411
Dek401:
        ldi     Temp    ,   0b00000001
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek402:
        ldi     Temp    ,   0b00000010
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek403:
        ldi     Temp    ,   0b00000011
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek404:
        ldi     Temp    ,   0b00000100
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek405:
        ldi     Temp    ,   0b00000101
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek406:
        ldi     Temp    ,   0b00000110
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek407:
        ldi     Temp    ,   0b00000111
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek408:
        ldi     Temp    ,   0b00001000
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek409:
```

```
        ldi     Temp    ,   0b00001001
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek410:
        ldi     Temp    ,   0b00000000
        mov     mCoun4  ,   Temp
        rjmp    Dek411
Dek411:
        rcall   Dunk04
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell04
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek04

;-----------------------------------------------------------------------------------------------------
; Sekunden x 10
; gewählt werden können 0, 10, 20, 30, 40 oder 50 Sekunden

Dek05:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek50
        brne    Dek57
Dek50:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun5
        cpi     Temp    ,   0b00000000
        breq    Dek51
        cpi     Temp    ,   0b00000001
        breq    Dek52
        cpi     Temp    ,   0b00000010
        breq    Dek53
        cpi     Temp    ,   0b00000011
        breq    Dek54
        cpi     Temp    ,   0b00000100
        breq    Dek55
        cpi     Temp    ,   0b00000101
        breq    Dek56
        rjmp    Dek57
Dek51:
        ldi     Temp    ,   0b00000001
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek52:
        ldi     Temp    ,   0b00000010
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek53:
        ldi     Temp    ,   0b00000011
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek54:
        ldi     Temp    ,   0b00000100
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek55:
        ldi     Temp    ,   0b00000101
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek56:
        ldi     Temp    ,   0b00000000
        mov     mCoun5  ,   Temp
        rjmp    Dek57
Dek57:
        rcall   Dunk05
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell05
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek05

;-----------------------------------------------------------------------------------------------------
; Sekunden x 1
; gewählt werden können 0, 1, 2 usw. bis 9 Sekunden

Dek06:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek600
        brne    Dek611
Dek600:
        ldi     Contr1  ,   0b00000000
```

```
        mov     Temp    ,   mCoun6
        cpi     Temp    ,   0b00000000
        breq    Dek601
        cpi     Temp    ,   0b00000001
        breq    Dek602
        cpi     Temp    ,   0b00000010
        breq    Dek603
        cpi     Temp    ,   0b00000011
        breq    Dek604
        cpi     Temp    ,   0b00000100
        breq    Dek605
        cpi     Temp    ,   0b00000101
        breq    Dek606
        cpi     Temp    ,   0b00000110
        breq    Dek607
        cpi     Temp    ,   0b00000111
        breq    Dek608
        cpi     Temp    ,   0b00001000
        breq    Dek609
        cpi     Temp    ,   0b00001001
        breq    Dek610
        rjmp    Dek611
Dek601:
        ldi     Temp    ,   0b00000001
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek602:
        ldi     Temp    ,   0b00000010
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek603:
        ldi     Temp    ,   0b00000011
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek604:
        ldi     Temp    ,   0b00000100
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek605:
        ldi     Temp    ,   0b00000101
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek606:
        ldi     Temp    ,   0b00000110
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek607:
        ldi     Temp    ,   0b00000111
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek608:
        ldi     Temp    ,   0b00001000
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek609:
        ldi     Temp    ,   0b00001001
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek610:
        ldi     Temp    ,   0b00000000
        mov     mCoun6  ,   Temp
        rjmp    Dek611
Dek611:
        rcall   Dunk06
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell06
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek06

;-----------------------------------------------------------------------------------------------------
; Sekunden x 0.10
; gewählt werden können 0, 0.1, 0.2 usw. bis 0.9 Sekunden

Dek07:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek700
        brne    Dek711
Dek700:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun7
        cpi     Temp    ,   0b00000000
        breq    Dek701
```

```
        cpi     Temp    ,   0b00000001
        breq    Dek702
        cpi     Temp    ,   0b00000010
        breq    Dek703
        cpi     Temp    ,   0b00000011
        breq    Dek704
        cpi     Temp    ,   0b00000100
        breq    Dek705
        cpi     Temp    ,   0b00000101
        breq    Dek706
        cpi     Temp    ,   0b00000110
        breq    Dek707
        cpi     Temp    ,   0b00000111
        breq    Dek708
        cpi     Temp    ,   0b00001000
        breq    Dek709
        cpi     Temp    ,   0b00001001
        breq    Dek710
        rjmp    Dek711
Dek701:
        ldi     Temp    ,   0b00000001
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek702:
        ldi     Temp    ,   0b00000010
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek703:
        ldi     Temp    ,   0b00000011
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek704:
        ldi     Temp    ,   0b00000100
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek705:
        ldi     Temp    ,   0b00000101
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek706:
        ldi     Temp    ,   0b00000110
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek707:
        ldi     Temp    ,   0b00000111
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek708:
        ldi     Temp    ,   0b00001000
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek709:
        ldi     Temp    ,   0b00001001
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek710:
        ldi     Temp    ,   0b00000000
        mov     mCoun7  ,   Temp
        rjmp    Dek711
Dek711:
        rcall   Dunk07
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell07
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek07

;-----------------------------------------------------------------------------------------------------
; Sekunden x 0.01
; gewählt werden können 0, 0.01, 0.02 usw. bis 0.09 Sekunden

Dek08:
        sei
        cpi     Contr1  ,   0b00000001
        breq    Dek800
        brne    Dek811
Dek800:
        ldi     Contr1  ,   0b00000000
        mov     Temp    ,   mCoun8
        cpi     Temp    ,   0b00000000
        breq    Dek801
        cpi     Temp    ,   0b00000001
        breq    Dek802
        cpi     Temp    ,   0b00000010
```

```
        breq    Dek803
        cpi     Temp    ,   0b00000011
        breq    Dek804
        cpi     Temp    ,   0b00000100
        breq    Dek805
        cpi     Temp    ,   0b00000101
        breq    Dek806
        cpi     Temp    ,   0b00000110
        breq    Dek807
        cpi     Temp    ,   0b00000111
        breq    Dek808
        cpi     Temp    ,   0b00001000
        breq    Dek809
        cpi     Temp    ,   0b00001001
        breq    Dek810
        rjmp    Dek811
Dek801:
        ldi     Temp    ,   0b00000001
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek802:
        ldi     Temp    ,   0b00000010
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek803:
        ldi     Temp    ,   0b00000011
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek804:
        ldi     Temp    ,   0b00000100
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek805:
        ldi     Temp    ,   0b00000101
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek806:
        ldi     Temp    ,   0b00000110
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek807:
        ldi     Temp    ,   0b00000111
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek808:
        ldi     Temp    ,   0b00001000
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek809:
        ldi     Temp    ,   0b00001001
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek810:
        ldi     Temp    ,   0b00000000
        mov     mCoun8  ,   Temp
        rjmp    Dek811
Dek811:
        rcall   Dunk08
        ldi     Time    ,   0b00001100
        rcall   Wait
        rcall   Hell08
        ldi     Time    ,   0b00001100
        rcall   Wait
        rjmp    Dek08

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

```
;-----------------------------------------------------------------------------------------------------
; CountDown\3_Decades.asm
;-----------------------------------------------------------------------------------------------------
; Dieser Programmteil wird dazu benutzt, die mittels Taste 1 gewählte Dekade blinken zu lassen.
; Die zuvor schon gewählten Dekaden leuchten konstant mit den jeweils mit Taste 2 eingestellten Werten.
;-----------------------------------------------------------------------------------------------------
; Stunden x 10

Dunk01:
        cli
        rcall   outCon                  ; No Shutdown, Data Coming, Decode, Code B
        ldi     Number  ,   0b00001111  ; Blank
        rcall   outDig                  ; Dekade 1 (ganz rechts)
        rcall   outDig                  ; Dekade 2
        rcall   outDig                  ; Dekade 3
        rcall   outDig                  ; Dekade 4
        rcall   outDig                  ; Dekade 5
        rcall   outDig                  ; Dekade 6
        rcall   outDig                  ; Dekade 7
        rcall   outDig                  ; Dekade 8 (ganz links)
        ldi     Number  ,   0b00001111  ; Blank

        ret

;-----------------------------------------------------------------------------------------------------
; Stunden x 10

Hell01:
        cli
        rcall   outCon                  ; No Shutdown, Data Coming, Decode, Code B
        ldi     Number  ,   0b00001111  ; Blank
        rcall   outDig                  ; Dekade 1
        rcall   outDig                  ; Dekade 2
        rcall   outDig                  ; Dekade 3
        rcall   outDig                  ; Dekade 4
        rcall   outDig                  ; Dekade 5
        rcall   outDig                  ; Dekade 6
        rcall   outDig                  ; Dekade 7
        mov     Number  ,   mCoun1      ; '0', '1' oder '2'
        rcall   outDig                  ; Dekade 8
        ldi     Number  ,   0b00001111  ; Blank
        ret

;-----------------------------------------------------------------------------------------------------
; Stunden x 1

Dunk02:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   DP1                     ; Dezimalpunkt für Dekade 7 = Stunden x 1, Signal Ein
        rcall   outDig
        rcall   DP2                     ; Dezimalpunkt für Dekade 7 = Stunden x 1, Signal Aus
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111
        ret

;-----------------------------------------------------------------------------------------------------
; Stunden x 1

Hell02:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111
```

```
        ret

;-----------------------------------------------------------------------------------------------------
; Minuten x 10

Dunk03:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Minuten x 10

Hell03:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Minuten x 1

Dunk04:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   DP1
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret
;-----------------------------------------------------------------------------------------------------
; Minuten x 1

Hell04:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
```

```
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 10

Dunk05:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 10

Hell05:
        cli
        rcall   outCon
        rcall   outDig
        rcall   outDig
        ldi     Number  ,   0b00001111
        rcall   outDig
        mov     Number  ,   mCoun5
        rcall   outDig
        cpi     Number  ,   0b00000000
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 1

Dunk06:
        cli
        rcall   outCon
        rcall   outDig
        rcall   outDig
        ldi     Number  ,   0b00001111
        rcall   outDig
        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
```

```
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 1

Hell06:
        cli
        rcall   outCon
        rcall   outDig
        rcall   outDig

        sbic    PIND    ,   5               ; bei 6 Dekaden Betrieb Dezimalpunkt
        rcall   DP1                         ; hinter Minuten x 1 ausblenden
        mov     Number  ,   mCoun6          ;
        rcall   outDig                      ;
        sbic    PIND    ,   5               ;
        rcall   DP2                         ;

        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 0.10

Dunk07:
        cli
        rcall   outCon
        rcall   outDig
        ldi     Number  ,   0b00001111
        rcall   outDig

        sbic    PIND    ,   5               ; bei 6 Dekaden Betrieb Dezimalpunkt
        rcall   DP1                         ; hinter Minuten x 1 ausblenden
        mov     Number  ,   mCoun6          ;
        rcall   outDig                      ;
        sbic    PIND    ,   5               ;
        rcall   DP2                         ;

        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

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

```
; Sekunden x 0.10

Hell07:
        cli
        cbi     PORTC   ,   0b00000100
        rcall   outCon
        rcall   outDig
        mov     Number  ,   mCoun7
        rcall   outDig

        sbic    PIND    ,   5               ; bei 6 Dekaden Betrieb Dezimalpunkt
        rcall   DP1                         ; hinter Minuten x 1 ausblenden
        mov     Number  ,   mCoun6          ;
        rcall   outDig                      ;
        sbic    PIND    ,   5               ;
        rcall   DP2                         ;

        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 0.01

Dunk08:
        cli
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        mov     Number  ,   mCoun7
        rcall   outDig

        sbic    PIND    ,   5               ; bei 6 Dekaden Betrieb Dezimalpunkt
        rcall   DP1                         ; hinter Minuten x 1 ausblenden
        mov     Number  ,   mCoun6          ;
        rcall   outDig                      ;
        sbic    PIND    ,   5               ;
        rcall   DP2                         ;

        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------
; Sekunden x 0.01

Hell08:
        cli
        rcall   outCon
        mov     Number  ,   mCoun8
        rcall   outDig
        mov     Number  ,   mCoun7
        rcall   outDig

        sbic    PIND    ,   5               ; bei 6 Dekaden Betrieb Dezimalpunkt
        rcall   DP1                         ; hinter Minuten x 1 ausblenden
        mov     Number  ,   mCoun6          ;
```

```
        rcall   outDig                  ;
        sbic    PIND    ,   5           ;
        rcall   DP2                     ;

        mov     Number  ,   mCoun5
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun4
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun3
        rcall   outDig
        rcall   DP1
        mov     Number  ,   mCoun2
        rcall   outDig
        rcall   DP2
        mov     Number  ,   mCoun1
        rcall   outDig
        ldi     Number  ,   0b00001111

        ret

;-----------------------------------------------------------------------------------------------------------
; Dezimalpunkt

DP1:
        cbi     PORTC   ,   2
        ret

DP2:
        sbi     PORTC   ,   2
        ret

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

```
;-----------------------------------------------------------------------------------------------------
; CountDown\4_Digits.asm
;-----------------------------------------------------------------------------------------------------
; Das CONTROL WORD des ICM7218 wird ausgegeben.

outCon:
        push    Contr0
        ldi     Contr0  ,   0b00000100      ; Data Coming, Decode, Code B
        out     PORTC   ,   Contr0          ; (Shutdown not wired)
        sbr     Contr0  ,   0b00110000      ; MODE   = High: Load Control Bits on /WRITE LOW Pulse
        out     PORTB   ,   Contr0          ; /WRITE = High: Input Not Loaded
        ldi     Contr0  ,   0b00100000      ; MODE   = High: Load Control Bits on /WRITE LOW Pulse
        out     PORTB   ,   Contr0          ; /WRITE = Low : Input Loaded       = /WRITE LOW Pulse ON
        ldi     Contr0  ,   0b00110000      ; MODE   = High: Load Control Bits on /WRITE LOW Pulse
        out     PORTB   ,   Contr0          ; /WRITE = High: Input Not Loaded   = /WRITE LOW Pulse OFF
        pop     Contr0
        ret

;-----------------------------------------------------------------------------------------------------
; Die HEX-Codes für die 7-Segment - Anzeigen werden ausgegeben

outDig:
        push    Contr1
        mov     Contr1  ,   Number          ; MODE   = Low : Load Input Data
        sbr     Contr1  ,   0b00010000      ; /WRITE = High: Input Not Loaded
        out     PORTB   ,   Contr1          ;
        mov     Contr1  ,   Number          ; MODE   = Low : Load Input Data
        cbr     Contr1  ,   0b00010000      ; /WRITE = Low : Input Loaded
        out     PORTB   ,   Contr1          ; /WRITE = Low : /WRITE LOW Pulse ON
        mov     Contr1  ,   Number          ; MODE   = Low : Load Input Data
        sbr     Contr1  ,   0b00010000      ; /WRITE = High: Input Not Loaded
        out     PORTB   ,   Contr1          ; /WRITE = High: /WRITE LOW Pulse OFF
        pop     Contr1
        ret

;-----------------------------------------------------------------------------------------------------
; Wartezeit: Time = 1 ==> 0.01 s, Time = 255 ==> 2.55 s.

Wait:
        push    Contr0
        push    Contr1
        cpi     Time    ,   0
        breq    WLoop0
WLoop1:
        ldi     Contr0  ,   0b01101110
WLoop2:
        ldi     Contr1  ,   0b01101110
WLoop3:
        dec     Contr1
        brne    WLoop3
        nop
        nop
        dec     Contr0
        brne    WLoop2
        dec     Time
        brne    WLoop1
WLoop0:
        pop     Contr1
        pop     Contr0
        ret

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

```
;-----------------------------------------------------------------------------------------------------
; CountDown\5_Timer.asm
;-----------------------------------------------------------------------------------------------------
; Timer:    Normalbetrieb:  (PORT D.4 = High) jede Sekunde wird "LTim00" 100 mal aufgerufen.
;           Debug-Betrieb:  (PORT D.4 = Low)  jede Sekunde wird "LTim00"   1 mal aufgerufen.
;
; Die nachfolgende Rechnung gilt, wenn D.4 = Low ist (Debug-Betrieb)
;
; Prozessortakt : 3.686.100 Hz      : 3.686.400 /   8   / 256   =   1.800   Hz
; Prescaler     : 8                 :           1.800   /   2   =     900   Hz  mTimr1 (C.3) Ein-Aus
; Zählstufen    : 256               :           1.800   /  90   =      20   Hz  mTimr2       gerechnet
; C.3           : 900,0000  Hz      :               20  /   2   =      10   Hz  mTimr2 (C.4) Ein-Aus
; C.4           :  10,0000  Hz      :               20  /  10   =       2   Hz  mTimr3       gerechnet
; C.5           :   1,0000  Hz      :                2  /   2   =       1   Hz  mTimr3 (C.5) Ein-Aus
;-----------------------------------------------------------------------------------------------------

onTc0:
        rcall   LTim00
reti

;-----------------------------------------------------------------------------------------------------
; Relais Ein und Aus, Sound
; Die Relais-Funktion kann zum Wischer geändert werden, zum Stoppen der Stoppuhr.
; Dafür auch die Wischer-Funktion in 2_Select.asm aktivieren und die Relais Ein Funktion hier deaktivieren.

LTim00:
        sbi     PORTD   ,   6               ; Relais Ein
        mov     Temp    ,   mCoun8          ; Bedingung für das Abschalten der Relais:
        cpi     Temp    ,   0b00000000      ; wenn alle Dekaden auf 0 stehen.
        brne    LTim01                      ; Wischfunktion des Relais für die Stoppuhr Start:
        mov     Temp    ,   mCoun7          ; s. in 2_Select.asm.
        cpi     Temp    ,   0b00000000      ; Dafür muß hier die 1. Zeile unter LTime00
        brne    LTim01                      ; auskommentiert werden.
        mov     Temp    ,   mCoun6          ; Das Relais kann hier nicht den Soppuhr Start auslösen,
        cpi     Temp    ,   0b00000000      ; weil LTimm 100 Mal pro Sekunse aufgerufen wird.
        brne    LTim01
        mov     Temp    ,   mCoun5
        cpi     Temp    ,   0b00000000
        brne    LTim01
        mov     Temp    ,   mCoun4
        cpi     Temp    ,   0b00000000
        brne    LTim01
        mov     Temp    ,   mCoun3
        cpi     Temp    ,   0b00000000
        brne    LTim01
        mov     Temp    ,   mCoun2
        cpi     Temp    ,   0b00000000
        brne    LTim01
        mov     Temp    ,   mCoun1
        cpi     Temp    ,   0b00000000
        brne    Ltim01
;       sbi     PORTD   ,   6               ; | Relais Ein für Wischer 10 ms für die  Stoppuhr Stopp
;       ldi     Time    ,   0b00000001      ; | Zur Aktivierung der Funktion Stoppuhr Stopp:
;       rcall   Wait                        ; | diese drei Zeilen Entfernen der Semikolons vorne.
        cbi     PORTD   ,   6               ; Relais Aus
        rjmp    Sound

;-----------------------------------------------------------------------------------------------------
; 1. Teil: 900 Hz ausgeben an C.3

LTim01:
        mov     Freq00  ,   mTimr1
        cpi     Freq00  ,   0
        breq    LTim02
        mov     Freq00  ,   mTimr1
        cpi     Freq00  ,   1
        breq    LTim03
LTim02:
        sbi     PORTC   ,   3               ; LED für die Anzeige von 900 Hz Ein
        ldi     Freq00  ,   1
        mov     mTimr1  ,   Freq00
        rjmp    LTim04
LTim03:
        cbi     PORTC   ,   3               ; LED für die Anzeige von 900 Hz Aus
        ldi     Freq00  ,   0
        mov     mTimr1  ,   Freq00
LTim04:
        in      Temp    ,   PIND            ; D.4 = High: 900 Hz / 9  = 100 Hz
        sbrs    Temp    ,   4               ; D.4 = Low:  900 Hz / 90 =  10 Hz
        rjmp    LTim05
        mov     Freq00  ,   mTimr4
        inc     Freq00
        mov     mTimr4  ,   Freq00
        cpi     Freq00  ,   9               ; Divisor für 100 Hz
        breq    LTim10
```

```
        rjmp    TimeEnd
LTim05:
        mov     Freq00  ,   mTimr4
        inc     Freq00
        mov     mTimr4  ,   Freq00
        cpi     Freq00  ,   90              ; Divisor für   1 Hz
        breq    LTim10
        rjmp    TimeEnd

;-----------------------------------------------------------------------------------------------
; 2. Teil:  Wenn D.4 = High, 100 Hz, keine Ausgabe an C.4
;           wenn D.4 = Low,   10 Hz,       Ausgabe an C.4 (Debug-Betrieb)

LTim10:
        mov     mTimr4  ,   Freq00
        mov     Freq00  ,   mTimr2
        cpi     Freq00  ,   0
        breq    LTim11
        mov     Freq00  ,   mTimr2
        cpi     Freq00  ,   1
        breq    LTim12
LTim11:
        sbi     PORTC   ,   4               ; LED für die Anzeige von 10 Hz Ein (nur im Debug-Betrieb)
        ldi     Freq00  ,   1
        mov     mTimr2  ,   Freq00
        rjmp    LTim13
LTim12:
        cbi     PORTC   ,   4               ; LED für die Anzeige von 10 Hz Aus (nur im Debug-Betrieb)
        ldi     Freq00  ,   0
        mov     mTimr2  ,   Freq00
LTim13:
        ldi     Freq00  ,   0
        mov     mTimr4  ,   Freq00
LTim14:
        in      Temp    ,   PIND            ; D.4 = High: 100 Hz / 1  = 100 Hz
        sbrs    Temp    ,   4               ; D.4 = Low:   10 Hz / 10 =   1 Hz
        rjmp    LTim15
        mov     Freq00  ,   mTimr5
        inc     Freq00
        mov     mTimr5  ,   Freq00
        cpi     Freq00  ,   1               ; Divisor für 100 Hz
        breq    LTim20
        rjmp    TimeEnd
LTim15:
        mov     Freq00  ,   mTimr5
        inc     Freq00
        mov     mTimr5  ,   Freq00
        cpi     Freq00  ,   10              ; Divisor für   1 Hz
        breq    LTim20
        rjmp    TimeEnd

;-----------------------------------------------------------------------------------------------
; 3. Teil:  Wenn D.4 = High, 100 Hz ausgeben an C.5
;           wenn D.4 = Low,    1 Hz ausgeben an C.5, dient zum Debuggen

LTim20:
        mov     mTimr5  ,   Freq00
        mov     Freq00  ,   mTimr3
        cpi     Freq00  ,   0
        breq    LTim21
        mov     Freq00  ,   mTimr3
        cpi     Freq00  ,   1
        breq    LTim22
LTim21:
        rcall   CountDown               ; Runterzählen
        sbi     PORTC   ,   5           ; LED für die Anzeige von 100 Hz bzw. 1 Hz Ein
        ldi     Freq00  ,   1
        mov     mTimr3  ,   Freq00
        rjmp    LTim23
LTim22:
        cbi     PORTC   ,   5           ; LED für die Anzeige von 100 Hz bzw. 1 Hz Aus
        ldi     Freq00  ,   0
        mov     mTimr3  ,   Freq00
LTim23:
        ldi     Freq00  ,   0
        mov     mTimr5  ,   Freq00

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

TimeEnd:
        ret

;-----------------------------------------------------------------------------------------------
; Sound  Programmende mit Summton ca. 460 Hertz, ca. 2 Sekunden und Ausgabe von Blanks an alle Dekaden.
;        Neustart durch gleichzeitiges Drücken beider Tasten, das löst einen Hardware-Reset aus.
```

```
Sound:
        cli
Sound1:
        ldi     Temp    ,   0b00000010
        ldi     Time    ,   0b11111111
        ldi     Contr0  ,   0b00000000
        ldi     Contr1  ,   0b00000000
        ldi     Number  ,   0b00000000
        ldi     Help00  ,   0b00000000
        ldi     Freq00  ,   0b00000000
        ldi     Temp01  ,   0b00000000
Sound3:
        cpi     Temp01  ,   0b00000001
        breq    Sound1
        ldi     Temp01  ,   0b00000001
        in      Freq00  ,   PIND
        sbrs    Freq00  ,   7
        sbi     PORTD   ,   7
        in      Freq00  ,   PIND
        sbrc    Freq00  ,   7
        cbi     PORTD   ,   7
Sound4:
        ldi     Contr0  ,   0b00111111
Sound5:
        ldi     Contr1  ,   0b00111111
Sound6:
        dec     Contr1
        brne    Sound6
        dec     Contr0
        brne    Sound5
Sound7:
        ldi     Temp01  ,   0b00000000
        cbi     PORTD   ,   7
        rcall   outCon
        ldi     Number  ,   0b00001111
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        rcall   outDig
        dec     Time
        brne    Sound3
Sound8:
        dec     Temp
        brne    Sound3
Sound9:
        rjmp    Sound9

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

```
;-----------------------------------------------------------------------------------------------------
; CountDown\6_Down.asm
;-----------------------------------------------------------------------------------------------------
; Runterzählen; Vergabe der Anfangswerte für jede Dekade

; mCoun1: 0 bis 2      Stunden  x 10
; mCoun2: 0 bis 3      Stunden  x  1 (bei der Eingabe)
; mCoun2: 0 bis 9      Stunden  x  1 (beim Runterzählen)
; mCoun3: 0 bis 5      Minuten  x 10
; mCoun4: 0 bis 9      Minuten  x  1
; mCoun5: 0 bis 5      Sekunden x 10
; Mcoun6: 0 bis 9      Sekunden x  1
; mCoun7; 0 bis 9      Sekunden x  0.1
; mCoun8: 0 bis 9      Sekunden x  0.01

CountDown:                              ; wird vom Timer   100 mal pro Sekunde aufgerufen
                                        ; im Debug-Betrieb   1 Mal pro Sekunde

Down1:
        mov     Contr0  ,   mCoun8      ; Sekunden x 0.01
        cpi     Contr0  ,   0
        breq    Down2
        dec     Contr0
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00      ; Sekunden x 0.01
        mov     mCoun8  ,   Contr0      ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down2:
        mov     Contr0  ,   mCoun7      ; Sekunden x 0.10
        cpi     Contr0  ,   0
        breq    Down3
        dec     Contr0
        mov     mCoun7  ,   Contr0      ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00      ; Sekunden x 1
        rcall   Anzeig
        ret
Down3:
        mov     Contr0  ,   mCoun6      ; Sekunden x 1
        cpi     Contr0  ,   0
        breq    Down4
        dec     Contr0
        mov     mCoun6  ,   Contr0      ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00      ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00      ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down4:
        mov     Contr0  ,   mCoun5      ; Sekunden x 10
        cpi     Contr0  ,   0
        breq    Down5
        dec     Contr0
        mov     mCoun5  ,   Contr0      ; Sekunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun6  ,   Help00      ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00      ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00      ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down5:
        mov     Contr0  ,   mCoun4      ; Minuten x 1
        cpi     Contr0  ,   0
        breq    Down6
        dec     Contr0
        mov     mCoun4  ,   Contr0      ; Minuten x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun5  ,   Help00      ; Sekunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun6  ,   Help00      ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00      ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00      ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down6:
        mov     Contr0  ,   mCoun3      ; Minuten x 10
        cpi     Contr0  ,   0
        breq    Down7
        dec     Contr0
```

```
        mov     mCoun3  ,   Contr0          ; Minuten x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun4  ,   Help00          ; Minuten x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun5  ,   Help00          ; Sekunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun6  ,   Help00          ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00          ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00          ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down7:
        mov     Contr0  ,   mCoun2          ; Stunden x 1
        cpi     Contr0  ,   0
        breq    Down8
        dec     Contr0
        mov     mCoun2  ,   Contr0          ; Stunden x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun3  ,   Help00          ; Minuten x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun4  ,   Help00          ; Minuten x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun5  ,   Help00          ; Sekunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun6  ,   Help00          ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00          ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00          ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down8:
        mov     Contr0  ,   mCoun1          ; Stunden x 10
        cpi     Contr0  ,   0
        breq    Down9
        dec     Contr0
        mov     mCoun1  ,   Contr0          ; Stunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun2  ,   Help00          ; Stunden x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun3  ,   Help00          ; Minuten x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun4  ,   Help00          ; Minuten x 1
        ldi     Help00  ,   0b00000101
        mov     mCoun5  ,   Help00          ; Sekunden x 10
        ldi     Help00  ,   0b00001001
        mov     mCoun6  ,   Help00          ; Sekunden x 1
        ldi     Help00  ,   0b00001001
        mov     mCoun7  ,   Help00          ; Sekunden x 0.10
        ldi     Help00  ,   0b00001001
        mov     mCoun8  ,   Help00          ; Sekunden x 0.01
        rcall   Anzeig
        ret
Down9:
        rjmp    Down9
;-----------------------------------------------------------------------------------------------------
; Ausgabe der Daten an die 7-Segment-Anzeigen; Ausblenden der führenden Nullen

; mCoun1: 0 bis 2       Stunden  x 10
; mCoun2: 0 bis 3       Stunden  x  1 (Eingabe)
; mCoun2: 0 bis 9       Stunden  x  1 (Runterzählen)
; mCoun3: 0 bis 5       Minuten  x 10
; mCoun4: 0 bis 9       Minuten  x  1
; mCoun5: 0 bis 5       Sekunden x 10
; Mcoun6: 0 bis 9       Sekunden x  1
; mCoun7; 0 bis 9       Sekunden x  0.1
; mCoun8: 0 bis 9       Sekunden x  0.01

Anzeig:
        cli
        rcall   outCon
        mov     Number  ,   mCoun8
        rcall   outDig
        mov     Number  ,   mCoun7
        rcall   outDig
        sbic    PIND    ,   5               ; bei 6-Dekaden-Betrieb Dezimalpunkt hinter
                                            ; Minuten x 1 ausblenden.
        rcall   DP1
        mov     Number  ,   mCoun6
        rcall   outDig
        sbic    PIND    ,   5
        rcall   DP2
        ldi     Number  ,   0b00001111      ; Blank
```

```
        mov     Help01  ,   mCoun1
        cpi     Help01  ,   0b00000000      ; mCoun1, mCoun2, mCoun3, mCoun4 und mCoun5
        brne    Anzg41                      ; werden geprüft, ob sie den Wert "0" besitzen.
        mov     Help01  ,   mCoun2          ; Wenn das für alle stimmt, dann wird ein Blank
        cpi     Help01  ,   0b00000000      ; an die Dekade "Sekunden x 10" ausgegeben.
        brne    Anzg41
        mov     Help01  ,   mCoun3
        cpi     Help01  ,   0b00000000
        brne    Anzg41
        mov     Help01  ,   mCoun4
        cpi     Help01  ,   0b00000000
        brne    Anzg41
        mov     Help01  ,   mCoun5
        cpi     Help01  ,   0b00000000
        brne    Anzg41
        rjmp    Anzg42
Anzg41:
        mov     Number  ,   mCoun5
Anzg42:
        rcall   outDig

        ldi     Number  ,   0b00001111
        mov     Help01  ,   mCoun1
        cpi     Help01  ,   0b00000000      ; mCoun1, mCoun2, mCoun3 und mCoun4
        brne    Anzg51                      ; werden geprüft, ob sie den Wert "0" besitzen.
        mov     Help01  ,   mCoun2          ; Wenn das für alle stimmt, dann wird ein Blank
        cpi     Help01  ,   0b00000000      ; an die Dekade "Minuten" ausgegeben.
        brne    Anzg51
        mov     Help01  ,   mCoun3
        cpi     Help01  ,   0b00000000
        brne    Anzg51
        mov     Help01  ,   mCoun4
        cpi     Help01  ,   0b00000000
        brne    Anzg51
        rjmp    Anzg52
Anzg51:
        rcall   DP1
        mov     Number  ,   mCoun4
Anzg52:
        rcall   outDig
        rcall   DP2

        ldi     Number  ,   0b00001111
        mov     Help01  ,   mCoun1
        cpi     Help01  ,   0b00000000      ; mCoun1, mCoun2 und mCoun3
        brne    Anzg61                      ; werden geprüft, ob sie den Wert "0" besitzen.
        mov     Help01  ,   mCoun2          ; Wenn das für alle stimmt, dann wird ein Blank
        cpi     Help01  ,   0b00000000      ; an die Dekade "Minuten x 10" ausgegeben.
        brne    Anzg61
        mov     Help01  ,   mCoun3
        cpi     Help01  ,   0b00000000
        brne    Anzg61
        rjmp    Anzg62
Anzg61:
        mov     Number  ,   mCoun3
Anzg62:
        rcall   outDig

        ldi     Number  ,   0b00001111
        mov     Help01  ,   mCoun1
        cpi     Help01  ,   0b00000000      ; mCoun1 und mCoun2
        brne    Anzg71                      ; werden geprüft, ob sie den Wert "0" besitzen.
        mov     Help01  ,   mCoun2          ; Wenn das für alle stimmt, dann wird ein Blank
        cpi     Help01  ,   0b00000000      ; an die Dekade "Stunden x 1" ausgegeben.
        brne    Anzg71
        rjmp    Anzg72
Anzg71:
        rcall   DP1
        mov     Number  ,   mCoun2
Anzg72:
        rcall   outDig
        rcall   DP2

        ldi     Number  ,   0b00001111
        mov     Help01  ,   mCoun1
        cpi     Help01  ,   0b00000000      ; mCoun1
        brne    Anzg81                      ; wird geprüft, ob es den Wert "0" besitzt.
        rjmp    Anzg82                      ; Wenn das stimmt, dann wird ein Blank
Anzg81:                                     ; an die Dekade "Stunden x 10" ausgegeben.
        mov     Number  ,   mCoun1
Anzg82:
        rcall   outDig
        ldi     Number  ,   0b00001111
        ret
;-----------------------------------------------------------------------------------------------
```