

DCF77-FUNKUHR BAUSTEIN in AVR Assembler

```
; Aufruf im 10 ms Zyklus (z.B. über Timer-Interrupt).

; Verwendung finden die Register R20-R23 sowie R30/31 als Z-Pointer.

; Der statische Datenbereich belegt 12 Bytes für den DCF77-Decoder.
; Vor Verwendung ist zu dessen Initialisierung dcf77_init aufzurufen.
; Weitere 8 Bytes enthalten die aktuelle Zeitinformation im BCD-Format
; und können mit zusätzlich installierter Soft-RTC betrieben werden.
; Diese sollte dem Decoder vorangestellt werden und nach Möglichkeit
; auch 1/100 Sekunden zählen (zur Vermeidung von Sek.-Sprüngen beim
; Minutenwechsel).

; Bitte die 20 Variablen wie definiert beieinander lassen.
; Kurzbeschreibung/Initialisierungswerte siehe unten.

; Das DCF77-Signal wird mit Aktiv "Low" an Input PB0 erwartet
; (kann frei nach Bedarf angepasst werden). Die Impulslängenmessung
; erfolgt über die Bestimmung der Signal-Pausenzeiten (in x * 1/100 Sek.).
```

```
dcf77_start: lds    r20,DCF77CT           ;alle 10 ms aufzurufen
             lds    r21,DCF77CHECK
             sbis   PINB,0              ;Input-Pin
             rjmp   dcf77_5            ;Aktiv = "Low" unbedingter Sprung!
                                     ; Keine Sprungmarke vorhanden
             inc    r20                 ; * Pulsdauer zählen
             brne   dcf77_3            ;*wenn kein Überlauf, speichern und fertig
dcf77_1:     sbr    r21,$80             ;Fehler-Flag setzen *Überlaufsituation 2,56s
             sts    DCF77CHECK,r21
dcf77_2:     clr    r20                 ; * Pulsdauer wieder auf „0“ setzen
dcf77_3:     sts    DCF77CT,r20        ; * Pulsdauer speichern
dcf77_4:     ret

dcf77_5:     tst    r20                 ;Auswertung fertig?
             breq   dcf77_4            ;Ja!
             lds    r22,DCF77STATE     ;Nein!
             lds    r23,DCF77SHIFT
             lds    ZL,DCF77TAL
             lds    ZH,DCF77TAH

             ; * Pausenlängen testen
             cpi    r20,196            ;LastBit 0 Obere Zeitgrenze+1 x
             brsh   dcf77_1            ;überschritten -> Fehler
             cpi    r20,187            ;LastBit-Scheide x
             brsh   dcf77_11           ;187-195 = LastBit0 * und Minutenende
             cpi    r20,178            ;LastBit 1 Untere Zeitgrenze x
             brsh   dcf77_10           ;178-186 = Lastbit1 * und Minutenende

             sbrc   r21,7              ;Wenn inzwischen Fehler...
             rjmp   dcf77_2            ;dann Abbruch unbedingter Sprung!
                                     ;Sprungmarke fehlt, hier kommen wir nie an
             cpi    r22,$15            ;STATE &lt; 21: * Alle Bits < 21 verwerfen
             brlo   dcf77_8            ;Inc STATE

             cpi    r20,6              ;Störspikegrenze x
             brlo   dcf77_2            ;Störspike wird ignoriert
             cpi    r20,96             ;Datenbit obere Zeitgrenze+1 x
             brsh   dcf77_1            ;überschritten -> Fehler
             cpi    r20,78             ;Datenbit untere Zeitgrenze x
             brlo   dcf77_1            ;unterschritten -> Fehler
             cpi    r20,87             ;Bitscheide x
             brsh   dcf77_6            ;87-95 = Bit0
             inc    r21                ;78-86 = Bit1
             ; * r21 = Zwischenspeicher für Paritätscheck

             sts    DCF77CHECK,r21
             cpi    r22,$1c            ; * Position des Minutenprüfbits im Telegramm
             breq   dcf77_6            ;Prüfbit Minute!
             cpi    r22,$23            ; * Position des Stundenprüfbits im Telegramm
             breq   dcf77_6            ;Prüfbit Stunde!
```

```

        ld      r20,Z          ;R20 = (aktuelle NEWTIME) Zelle
        add    r20,r23        ;addiere SHIFTER
        st     Z,r20         ;zurückspeichern

dcf77_6:  lsl      r23          ;Bit0/1: SHIFTER um 1 nach links
        sts    DCF77SHIFT,r23
        cpi    r22,$1c
        brlo  dcf77_8        ;Minuten-Zyklus
        breq  dcf77_9        ;Minute Prüfbit
        cpi    r22,$23
        brlo  dcf77_8        ;Stunden-Zyklus
        breq  dcf77_9        ;Stunden-Prüfbit
        cpi    r22,$29
        brlo  dcf77_8
        breq  dcf77_7        ;LastBit TAG
        cpi    r22,$2c
        brlo  dcf77_8
        breq  dcf77_7        ;LastBit WOCHENTAG
        cpi    r22,$31
        brne  dcf77_8

dcf77_7:  ; letztes Bit eines Datums (Stunde, Tag etc.)
        ldi    r23,1         ;LastBit MONAT
        adiw   ZH:ZL,1
        sts    DCF77SHIFT,r23
        sts    DCF77TAL,ZL
        sts    DCF77TAH,ZH

dcf77_8:  ; * beliebiges Bit verarbeitet
        inc    r22           ;Inc STATE * Bitnummer erhöhen
        sts    DCF77STATE,r22 ; * Bitnummer speichern
        rjmp   dcf77_2       ; * Pulsdauer auf 0 und fertig

dcf77_9:  sbrcl  r21,0        ;Prüfbit MIN/STU Auswertung
        rjmp   dcf77_1       ;Prüfbit Fehler unbedingter Sprung!
        rjmp   dcf77_7       ;Prüfbit OK Sprungmarke fehlt, dieser Sprung
        ; wird nie ausgeführt

dcf77_10: inc    r21          ;LastBit = 1: Inc CHECK
dcf77_11: cpi    r22,$3a      ;58 DCF77-Bits erfasst?
        brne  dcf77_init     ;unvollständig -&gt; dcf77_init
        andi  r21,$81        ; * Parity für Datumsfeld prüfen
        brne  dcf77_init     ;Fehlerstatus -&gt; dcf77_init

        ;Erfolgreicher Empfang:
        ldi    r20,6         ; 6 Datenbyte wegspeichern
        ld     r21,Z         ;BCD-Zeitinformation
dcf77_12: std    Z+11,r21     ;nach MIN...JHR sichern
        ld     r21,-Z
        dec   r20
        brne  dcf77_12      ; Schleife über 6 Byte

dcf77_13: ; * Sprungmarke wird nicht genutzt
        std    Z+11,r20     ;Sek. = 0 setzen
        std    Z+10,r20     ;1/100 Sek. = 0 setzen

dcf77_init: ldi    ZL,LOW(DCF77CT) ;DCF77-VARIABLEN RESET * initialisieren
        ldi    ZH,HIGH(DCF77CT)
        ldi    r21,LOW(NEWTIME) ; * Adresse nach r21/r22
        ldi    r22,HIGH(NEWTIME)
        ldi    r20,$09       ; * insgesamt 9 Byte löschen
        clr   r23           ; * r23 = 0
dcf77_15: st     Z+,r23      ; * Speicherstelle löschen
        dec   r20
        brne  dcf77_15     ; * Schleife zum Löschen des Speichers
        inc   r20          ; * r20 = 1
        st    Z+,r20       ; * DCF77Shift initialisieren
        st    Z+,r21       ; * Pointer r21/r22 anlegen
        st    Z,r22
        ret

```

;SRAM Variable	Definition	Init.Wert	Beschreibung
DCF77CT:	.BYTE 1	;0	DCF77 Impuls-Pausenzeit-Zähler
DCF77CHECK:	.BYTE 1	;0	DCF77 Fehler-Flag + 1-Zähler
DCF77STATE:	.BYTE 1	;0	DCF77 Aktuelle Bitnummer
NEWTIME:	.BYTE 6	;0,0,0,0,0,0	DCF77 Zeit-Zwischenspeicher
DCF77SHIFT:	.BYTE 1	;1	DCF77 Shifter
DCF77TAL:	.BYTE 1	;LOW(NEWTIME)	DCF77 Pointer auf NEWTIME Low
DCF77TAH:	.BYTE 1	;HIGH(NEWTIME)	DCF77 Pointer auf NEWTIME High
HSEK:	.BYTE 1	;RTC verwaltet	BCD 1/100 Sekunde
SEK:	.BYTE 1	;RTC verwaltet	BCD Sekunde
MIN:	.BYTE 1	;RTC verwaltet	BCD Minute
STU:	.BYTE 1	;RTC verwaltet	BCD Stunde
TAG:	.BYTE 1	;RTC verwaltet	BCD Tag
WTAG:	.BYTE 1	;RTC verwaltet	BCD Wochentag
MON:	.BYTE 1	;RTC verwaltet	BCD Monat
JHR:	.BYTE 1	;RTC verwaltet	BCD Jahr