

## Berechnung des Karfreitages für die Jahre 2000 bis 2099

Mit der Berechnung des Karfreitages können Geräte, die öffentliche Musikdarbietungen (z.B. Glockenspiele) erzeugen, vollautomatisch am Karfreitag abgeschaltet werden. Die Auflage des Gesetzgebers für diesen Feiertag werden hierdurch frei von menschlichen Fehlern und ohne manuelles Eingreifen zuverlässig erfüllt.

Für alle Berechnungen ist vorteilhafterweise nur 1-Byte Mathematik erforderlich. Bei einem Quarz mit 10 MHz dauert die Errechnung des Karfreitagsdatum lediglich 180 Mikrosekunden (ca. 450 Zyklen).

Mit 83 Bytes beträgt der benötigte Programm-Platzbedarf nur 30 % des Platzes, den eine Tabelle mit 261 Bytes zur Abspeicherung einer Karfreitagstabelle mit allen Daten der Jahre 2013 bis 2099 einnimmt.

Datum in packed BCD – Format: JJ MM TT; (J=Jahr, M=Monat, T=Tag).

Außerdem entfällt das lästige Erstellen der Daten für alle Jahre dieser großen Tabelle, was nicht ganz unproblematisch ist, da die Internet-Lösungen zum Teil mit erheblichen Fehlern belastet sind.

Sollte dispSec bis dispWtag im Hex-Format vorhanden sein, dann kann die Subroutine „inHEX“ und das Programmteil BCD-Umwandlung entfallen. Dann beträgt der Programmplatz nur noch 47 Byte und die Laufzeit verkürzt sich um 37 Zyklen.

Die Formeln wurden im April 2013 von Manfred Wilzeck entwickelt:

<http://manfred.wilzeck.de/>

Das Assembler-Programm für den PIC16F628A wurde von Michael Buschmann geschrieben:

<http://www.pneuphoniker.de/>

Die Subroutine zum Umwandeln einer packed BCD Zahl in Hex (inHEX) stammt von Scott Dattalo.

Formeln für Jahr 2000 bis 2099 (00 bis 99):

A = year MOD 19 ; year = Jahreszahl, 2-stellig.  
LookTAB Ostermond, A, B ; A gibt die Stelle der Tabelle Ostermond  
; an, die in B eingelesen werden muss.  
C = (B + year + year / 4) MOD 7 ; Wochentag des Ostervollmonds  
E = B – C + 111 ; Osterkennzahl  
D = B – C + 109 ; Karfreitagkennzahl

TABLE Ostermond ; Position 0 bis 18  
37 27 16 35 24 13 32 21 10 29 18 36 26 15 33 22 11 30 19  
TABEND

Tag == 1 + D MOD 31 then Karfreitag  
Monat == D / 31 then Karfreitag

Beispiel: year = 13 (= 2013)  
A = year MOD 19 = 13  
Look TAB Ostermond, A, B: B = 15  
C = (B + year + year / 4) MOD 7 = 3  
D = B – C + 109 = 121  
1 + D MOD 31 = 29  
D / 31 = 3  
Also ist Karfreitag am 29.3.2013

```

-----
Microcontroller definieren
;-----
#define __16F628A
;-----
;RAM-Variablen festlegen
;-----
cblock0x20

; aktuelle Zeit (in packed BCD!!!)
    dispSec, dispMin, dispStd, dispTag, dispMon, dispJahr, dispWtag, dispStat

;Für Berechnung Karfreitag
    AA, BB, CC, DD, H1, H2          ;Konstante A, B, C, E, Hilfsregister H1 u. H2

                                ; Für AA gilt nach Durchführung der Berechnung:
                                ; Wenn AA=0 dann kein Karfreitag,
                                ; wenn AA<>0 dann Karfreitag

; Für Division (Dividend / Divisor = Quotient + Rest)
    divid, divis, quo, rest
; Für Umwandlung BCD in Hex
    JBin, MBin, TBin
endc

;-----
; Tabellen in Page 0:
;-----
;Tabelle für Errechnung des Karfreitages
Karfr clrf   PCLATH
      addwf PCL,f
; Tabelle mit 19 Werten an Position 0 - 18:
;37 27 16 35 24 13 32 21 10 29 18 36 26 15 33 22 11 30 19
    retlw d'37'
    retlw d'27'
    retlw d'16'
    retlw d'35'
    retlw d'24'
    retlw d'13'
    retlw d'32'
    retlw d'21'
    retlw d'10'
    retlw d'29'
    retlw d'18'
    retlw d'36'
    retlw d'26'
    retlw d'15'
    retlw d'33'
    retlw d'22'
    retlw d'11'
    retlw d'30'
    retlw d'19'

```

```

;-----
; Hauptprogramm :
;-----
;
;-----
;Jahreszahl auf Karfreitag prüfen (Dauer: 181 Mikrosekunden)
;-----
Kaf          clrwdt          ;Watchdog (normalerweise: 18 mS) löschen
;-----
;kann entfallen, wenn dispJahr – dispTag im Hex-Format vorliegt.
;Jahr von packed BCD in Hex umwandeln, damit gerechnet werden kann
    movfw     dispJahr
    call      inHEX
    movwf     JBin
;Monat von packed BCD in Hex umwandeln, damit später verglichen werden kann
    movfw     dispMon
    call      inHEX
    movwf     MBin
;Tag von packed BCD in Hex umwandeln, damit später verglichen werden kann
    movfw     dispTag
    call      inHEX
    movwf     TBin
;-----

;    Konstante A Berechnen: Jahr MOD 19
    movfw     JBin          ;Jahr nach divid
    movwf     divid
;MOD 19
    movlw     d'19'
    movwf     divis
    call      DIVIS
    movfw     rest
    movwf     AA          ;A wurde errechnet

;    Tabelle lesen und somit B ermitteln
    call      Karfr        ;AA ist noch in w
    movwf     BB          ;B wurde berechnet

;    Konstante C berechnen
    movfw     JBin          ;w = Jahr
    movwf     H1          ;H1=Jahr
;Division durch 4
    bcf       STATUS,C    ;H1=H1/2
    rrf       H1,1
    bcf       STATUS,C    ;H1=H1/2
    rrf       H1,1
; + Jahr
    movfw     JBin          ;w=Jahr

```

```

    addwf    H1,1      ;H1=Jahr+Jahr/4
    movfw   BB
    addwf    H1,1      ;H1=B+Jahr+Jahr/4
; MOD 7
    movfw   H1
    movwf   divid
    movlw   d'7'
    movwf   divis
    call    DIVIS
    movfw   rest
    movwf   CC        ;C wurde errechnet

; Konstante D berechnen
    movlw   d'109'
    movwf   H1
    movfw   BB
    addwf    H1,1      ;H1=109+B
    movfw   CC
    subwf   H1,1      ;H1=109+B-C
    movfw   H1
    movwf   DD        ;D wurde berechnet ab hier sind die RAM-
                    ;Zellen AA,BB,CC, H1 u. H2 frei

; D/31=Monat Karfreitag und DMOD31 = Tag Karfreitag berechnen
    movfw   DD
    movwf   divid
    movlw   d'31'
    movwf   divis
    call    DIVIS     ;Monat: quo = D / 31 und Tag: rest = D MOD 31
    incf    rest,1    ;rest = D MOD 31 + 1

;aktueller Tag und Monat mit Karfreitagsdatum vergleichen:
    movfw   TBin
    subwf   rest,0    ;w=rest-dispTag
    btfss   STATUS,Z  ;Tag Karfreitag = dispTag?
    goto    G1Sp      ;nein, kein Karfreitag!!!
    movfw   MBin     ;ja, also Monat prüfen
    subwf   quo,0     ;w=quo-dispMon
    btfss   STATUS,Z  ;Monat Karfreitag = dispMon?
    goto    G1Sp      ;nein, also kein Karfreitag!!!
    goto    sz6       ;ja, also KARFREITAG!!!

```

```

;-----
;Subroutine DIVIS
;Division für Karfreitagsberechnung durchführen
;      divid / divis = quo + rest
;-----
DIVIS      clrf      quo
           Movfw    divid
           Movwf    rest
           Movfw    divis
DIVIS1     subwf    rest,1      ;rest=rest-divis
           btfss   STATUS,C    ;C=1?
           goto    DIVIS2     ;C=0, also Überlauf erreicht
           incf    quo,1      ;C=1, also war divis kleiner als divid: Also
                               ;wiederholen bis Übertrag entsteht
           goto    DIVIS1
DIVIS2     addwf    rest,1      ;rest=rest+divis
           return   ;Ende Division
;-----
;Subroutine inHEX
;      packed BCD in Hex umwandeln für die Berechnung Karfreitag
;      Formel Hex = 10*(BCD / 16) + BCD MOD 16
;      Ergebnis steht in w
;      Input: w mit umzuwandelnder packed BCD Zahl laden
;      Benötigte Register: H1, H2
;-----
inHEX     movwf    H1
           rrf     H1,w
           andlw   b'01111000'
           movwf    H2
           clrc
           rrf     H2,F
           rrf     H2,F
           subwf   H1,w
           addwf   H2,w
           return
;-----

GlSp      clrf      AA          ;AA=0 weil kein Karfreitag ist
           goto    sz6

sz6       bsf      AA,1        ; AA<>0 weil Karfreitag ist
           return

end

```