

```

*****
' * ----- 2 0 0 4 0 8 2 2 . B A S ----- *
' * 6 STELLIGE UHRZEITAUSGABE DCF-GESTEUERT MIT STUNDENSIGNAL *
' * TREIBER UHR57SM3.S19 FUER 7-SEGMENTANZEIGE "LTS 23805 HRB" *
' * PROGRAMM FUER 4MHZ M-UNIT GESCHRIEBEN / VERSION 22.08.2004 *
*****

'== ADRESSEN =====
DEFINE LEDZAHL      &H101      'MEHRSTELLIGE ZAHLEN AUSGEBEN
DEFINE LEDZEICHEN   &H192      'ASCII ZEICHEN AUSGEBEN
DEFINE DCFSTATUS    &H1C0      'PRUEFEN OB DCF SIGNAL EMPFANGEN
DEFINE DEFINIERT     &H1CB      'PORT-A AUS AUSGANG DEFINIEREN
'== EIN/AUSGAENGE =====
DEFINE DCFLED       PORT[1]     'LED      DCF-STATUS
DEFINE LSHCLK        PORT[6]     'CLOCK    SCHIEBEREGISTER 74HC595
DEFINE LEDCLK        PORT[7]     'CLOCK    DATENREGISTER 74HC595
DEFINE LSHDAT        PORT[8]     'DATEN    SCHIEBEREGISTER 74HC595
DEFINE POT1          AD[1]       'POT1     GRUNDEINSTELLUNG HELBIGKEITSREGELUNG
DEFINE LDR           AD[2]       'LDR      AUTOMATISCHE HELBIGKEITSREGELUNG
DEFINE LED           DA[1]       'LED      PWM-AUSGANG HELBIGKEITSREGELUNG
'== VARIABLEN =====
DEFINE ZDCF          WORD        'ZAEHLER  DCF SIGNAL ANZEIGEN
DEFINE ZTON          BYTE        'ZAEHLER  STUNDENSIGNAL
DEFINE HELL          BYTE        'MERKERBYTE HELBIGKEITSREGELUNG
DEFINE NULL1         BYTE        'FUEHRENDE NULL UNTERDRUECKUNG UEBERLISTEN
DEFINE NULL2         BYTE        'FUEHRENDE NULL UNTERDRUECKUNG UEBERLISTEN
DEFINE HMZEIT        BIT[190]    'MERKER  ANZEIGE AKTUALISIERT
DEFINE HMTON         BIT[191]    'MERKER  STUNDENSIGNAL
DEFINE HMDCF         BIT[192]    'MERKER  DCF SIGNAL EMPFANGEN
'== GRUNDEINSTELLUNG =====
HOUR      = 12
LEDCLK    = 0
LSHCLK    = 0
LSHDAT    = 0
HMZEIT    = 0
HMTON     = 0
HMDCF     = 0
ZTON      = 0
ZDCF      = 0
LED       = 255
HELL      = 255
SYS DEFINIERT          'PORT-A ALS AUSGANG DEFINIEREN (ESD)
'== ANFANG HAUPTROUTINE =====
#LOOP
GOSUB LUMEN              'HELLIGKEITSREGELUNG
GOSUB ZEIT              'UHRZEIT AUSGEBEN
GOSUB KUCKUCK           'STUNDENSIGNAL AUSGEBEN
SYS DCFSTATUS           'DCF EMPFANG PRUEFEN (MIN 2X PRO SEK.)
DCFLED = HMDCF          'DCF LED AUF PORT-1 AUSGEBEN
PRINT ZDCF              'ZAEHLWERT DCF ANZEIGEN
GOTO LOOP              'ERNEUTER PROGRAMMDURCHLAUF
'== PROGRAMMENDE =====
END

*****
' * STUNDEN UND MINUTEN AUSGEBEN *
*****
#ZEIT
' ** ANZEIGEN OB DIE ANGEZEIGTE UHRZEIT DCF SYNCRON IST *****
IF NOT HMDCF THEN ZDCF = ZDCF +1      'KEIN DCF EMPFANG = AUFWAERTSZAEBLEN
IF HMDCF THEN ZDCF = ZDCF -1          ' DCF EMPFANG = ABWAERTSZAEBLEN
IF ZDCF > 32500 THEN ZDCF = 32500     'UEBERLAUF ZAEHLER VERHINDERN
IF ZDCF < 5 THEN ZDCF = 5             'UEBERLAUF ZAEHLER VERHINDERN
' ** ZEIT NUR 1 MAL PRO SEKUNDE ANZEIGEN *****
IF NULL1 = SECOND + 100 THEN GOTO ZEITAUS
' ** MINUTEN UND SEKUNDEN MIT FUEHRENDEN NULLEN ANZEIGEN *****
NULL1 = SECOND +100                  'LED-TREIBER UEBERLISTEN
NULL2 = MINUTE +100                  'LED-TREIBER UEBERLISTEN
'DIESES PROGRAMM LAEUFT MIT 18,6 ZYKLEN PRO SEKUNDE D.H. ERST WENN 28 MINUTEN
'KEIN DCF SIGNAL EMPFANGEN WIRD, SO WIRD EIN SEKUNDENWERT VON 99 ANGEZEIGT.
'OHNE DEN ZAEHLER WUERDE, NACH EINEM KURZZEITIGEN AUSFALL DES EMPFANGS,
'DIE ANZEIGE DER SEKUNDEN FUER MINDESTENS 3 MINUTEN 99 ANZEIGEN.
'ANMERKUNG: NACH 28 MINUTEN SOLLTE DIE UHR DER C-CONTROL NOCH RICHTIG GEHEN.
IF ZDCF > 32000 THEN NULL1 = 199      'WENN ZEIT NICHT SYNCRON, DANN SEKUNDEN = 99
' ** BLINKENDE DEZIMALPUNKTE ERZEUGEN UND ZEIT AUSGEBEN *****
#ZEITDPAUS
' ** ZEIT OHNE DOPPELPUNKT AUSGEBEN *****
IF HMZEIT = -1 THEN GOTO ZEITDPEIN    'AUSGABE NUR 1MAL PRO SEKUNDE
SYS LEDZAHL NULL1,0,2                 'SEKUNDEN 2-STELLIG AUSGEBEN
SYS LEDZAHL NULL2,2,2                 'MINUTEN 2-STELLIG AUSGEBEN
SYS LEDZAHL HOUR,1,2                 'STUNDEN 1 ODER 2-STELLIG AUSGEBEN
PULSE LEDCLK                         'DATEN UEBERNEHMEN
HMZEIT = -1                          'AUSGABE IST ERFOLGT
GOTO ZEITAUS                         'ENDE DER UNTERROUTINE
#ZEITDPEIN
' ** ZEIT MIT DOPPELPUNKT AUSGEBEN *****

```

```

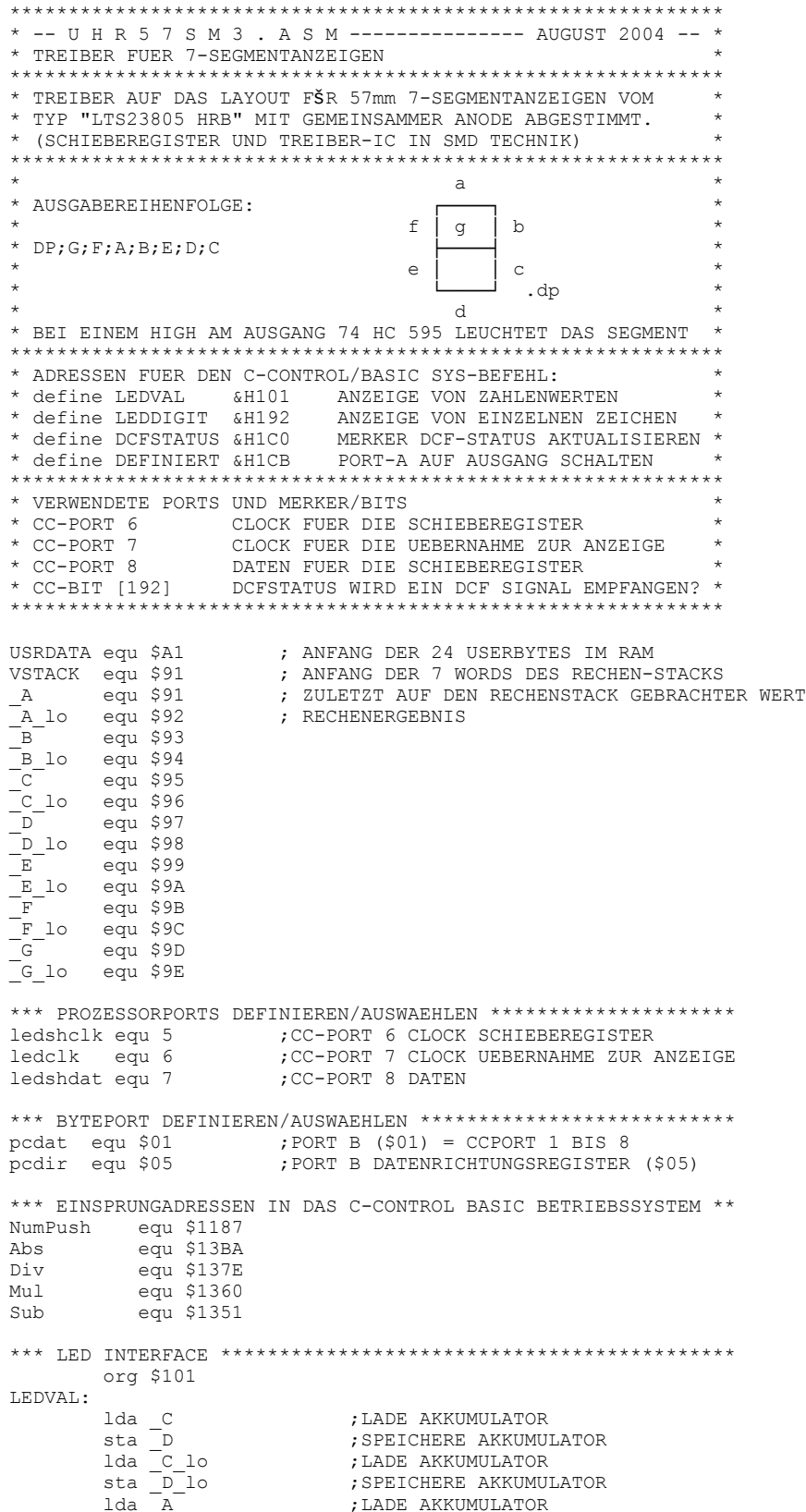
IF HMZEIT = 0 THEN GOTO ZEITAUS      'AUSGABE NUR 1MAL PRO SEKUNDE
SYS LEDZAHL NULL1,2,2                'SEKUNDEN 2-STELLIG AUSGEBEN
SYS LEDZAHL NULL2,1,2                'MINUTEN 2-STELLIG AUSGEBEN
SYS LEDZAHL HOUR,0,2                 'STUNDEN 1 ODER 2-STELLIG AUSGEBEN
PULSE LEDCLK
HMZEIT = 0                           'AUSGABE IST ERFOLGT
GOTO ZEITAUS                         'ENDE DER UNTERROUTINE
'== ENDE UNTERPROGRAMM =====
#ZEITAUS
RETURN

'*****
'*  DISPLAY-HELLIGKEIT AENDERN      *
'*****
#LUMEN
'** ANZEIGE DUNKEL ZWISCHEN 23 UHR UND 6 UHR *****
IF HOUR > 22 OR HOUR < 6 THEN GOTO NACHT ELSE GOTO TAG
#TAG
'255 = ANZEIGE DUNKEL / 0 = ANZEIGE HELL
HELL = LDR                           'WERT VOM LDR SENSOR UEBERNEHMEN
IF LDR < POTI THEN HELL = POTI        'MAXIMALE HELLIGKEIT UND LIMIT
'UM 22:08:08 UHR LEUCHTEN DIE MEISTEN LED-SEGMENTE
IF HELL < 75 THEN HELL = 75          'STROMLIMIT DES NETZTEILS 420mA
LED = HELL                           'HELLIGKEIT ZUM DA-AUSGANG SENDEN
GOTO LUMENAU
#NACHT
LED = 255                            'NACHTS ENERGIESPAREN
#LUMENAU
RETURN

'*****
'*  STUNDENSIGNAL AUSGEBEN          *
'*****
#KUCKUCK
'** WENN NEUE STUNDE ANGEFANGEN, DANN STUNDENSIGNAL *****
IF MINUTE = 0 THEN GOTO TONSTART      'MINUTE = 0 ?
HMTON = 0                            'SIGNAL AUSGEGEREN
ZTON = 0                             'ZAEHLER ZURUECKSETZEN
GOTO TONAU
#TONSTART
'** WENN STUNDENSIGNAL AUSGEGEREN, DANN BEENDEN *****
IF HMTON = -1 THEN GOTO TONAU
'** GRUNDEINSTELLUNGEN *****
IF ZTON = 0 THEN ZTON = HOUR          'ZAEHLER STUNDENSIGNAL LADEN
IF ZTON = 0 THEN GOTO TONAU          'KEIN SIGNAL
IF HOUR > 22 OR HOUR < 6 THEN GOTO TONAU 'KEIN SIGNAL VON 22 - 6 UHR
IF ZTON > 19 THEN GOTO TON20          '2-LANGE TONE AB 20 UHR
IF ZTON > 9 THEN GOTO TON10           '1-LANGER TON AB 10 UHR
#TONNEXT
'** 0 BIS 9 UHR *****
BEEP 400,5,5                         'KURZEN TON AUSGEBEN
GOTO TONCOUNT                       'ZAEHLER UM 1 VERMINDERN
'** 10 BIS 19 UHR *****
#TON10
BEEP 500,30,20                       'LANGEN TON AUSGEBEN
ZTON = ZTON - 9                      'ZAEHLER UM 9 VERMINDERN
GOTO TONCOUNT                       'ZAEHLER UM 1 VERMINDERN
'** 20 BIS 23 UHR *****
#TON20
BEEP 500,30,20                       'LANGEN TON AUSGEBEN
GOSUB ZEIT                           'NUR UHRZEIT AKTUALISIEREN
BEEP 500,30,20                       'LANGEN TON AUSGEBEN
ZTON = ZTON - 19                     'ZAEHLER UM 19 VERMINDERN
#TONCOUNT
GOSUB ZEIT                           'NUR UHRZEIT AKTUALISIEREN
ZTON = ZTON - 1                      'ZAEHLER UM 1 VERMINDERN
IF ZTON > 0 THEN GOTO TONNEXT        'NAECHSTEN TON AUSGEBEN
HMTON = -1
'== ENDE UNTERPROGRAMM =====
#TONAU
RETURN

'** HINWEIS ZUR TREIBERDATEI *****
'-> SOLANGE DIE TREIBERDATEI NICHT VERAENDERT WIRD
'-> BRAUCHT DIESE NUR EINMAL GELADEN ZU WERDEN.
'SYSCODE "UHR57SM1.S19"             'LCD TREIBER
'SYSCODE "UHR57SM2.S19"             'LCD TREIBER & DCF-MERKER
'SYSCODE "UHR57SM3.S19"             'LCD TREIBER & DCF-MERKER & PORT-A AUF GND

```



```

    sta _C                ;SPEICHERE AKKUMULATOR
    lda _A_lo             ;LADE AKKUMULATOR
    sta _C_lo             ;SPEICHERE AKKUMULATOR
    lda _D                ;LADE AKKUMULATOR
    sta _A                ;SPEICHERE AKKUMULATOR
    lda _D_lo             ;LADE AKKUMULATOR
    sta _A_lo             ;SPEICHERE AKKUMULATOR
    brclr 7,_A,nextdigit  ;SPRINGE WENN BIT 7,_A GELOESCHT ZU ..
    bset 0,_B             ;SETZE BIT 0 , _B
    jsr Abs               ;SPRINGE ZUR UNTERROUTINE ...
nextdigit:
    jsr NumPush           ;SPRINGE ZUR UNTERROUTINE ...
    jsr NumPush           ;SPRINGE ZUR UNTERROUTINE ...
    lda _C                ;LADE AKKUMULATOR      sta _A                ;SPEICHERE AKKUMULATOR
    sta _B                ;SPEICHERE AKKUMULATOR
    lda _C_lo             ;LADE AKKUMULATOR
    lda _A_lo             ;LADE AKKUMULATOR
    lda _B_lo             ;LADE AKKUMULATOR
    jsr NumPush           ;SPRINGE ZUR UNTERROUTINE ...
    jsr load10            ;SPRINGE ZUR UNTERROUTINE ...
    jsr Div               ;SPRINGE ZUR UNTERROUTINE ...
    jsr NumPush           ;SPRINGE ZUR UNTERROUTINE ...
    jsr load10            ;SPRINGE ZUR UNTERROUTINE ...
    jsr Mul               ;SPRINGE ZUR UNTERROUTINE ...
    jsr Sub               ;SPRINGE ZUR UNTERROUTINE ...
    ldx _A_lo             ;LADE INDEX REGISTER
    lda ledcodes,x        ;ZEICHEN AUS VORRAT LADEN
    ldx _C_lo             ;LADE INDEX REGISTER
    beq nopoint           ;DEZIMALPUNKT NICHT VERWENDET
    dec _C_lo             ;REGISTER C LOW UM 1 VERMINDERN
    bne nopoint           ;DEZIMALPUNKT VERWENDET, ABER ANDERE STELLE
    bra point             ;DEZIMALPUNKT VERWENDET, DIESE STELLE
nopoint:
    and #$7F             ;DEZIMALPUNKT ABSCHALTEN (BIT 7)
point:
    jsr digit             ;SPRINGE ZUR UNTERROUTINE ...
    jsr load10            ;SPRINGE ZUR UNTERROUTINE ...
    jsr Div               ;SPRINGE ZUR UNTERROUTINE ...
    lda _B_lo             ;LADE AKKUMULATOR
    bne no0check          ;SPRINGE WENN NICHT GLEICH ...
    lda _A                ;LADE AKKUMULATOR
    ora _A_lo             ;LOGISCHE ODER VERKNUEPFUNG
    beq sign              ;SPRINGE, WENN NULL ...
no0check:
    dec _C_lo             ;VERMINDERE REGISTER UM 1
    bne nextdigit        ;SPRINGE, WENN NICHT GLEICH ...
    rts                  ;UNTERROUTINE BEENDEN
sign:
    dec _C_lo             ;VERMINDERE REGISTER UM 1
    beq exit              ;SPRINGE, WENN NULL ...
    brclr 0,_B,space      ;SPRINGE, WENN BIT 0,_B = LOW
    brclr 1,_B,minus      ;SPRINGE, WENN BIT 1,_B = LOW
    lda #$BF              ;LADE AKKU MIT CODE FUEER "0"
    jsr digit             ;SPRINGE ZUR UNTERROUTINE ...
    jmp nextspace         ;SPRINGE NACH NEXTSPACE
minus:
    lda #$40              ;BITMUSTER FUEER DAS MINUSZEICHEN
    jsr digit             ;SPRINGE ZUR UNTERROUTINE ...
nextspace:
    dec _C_lo             ;VERMINDERE REGISTER UM 1
    beq exit              ;SPRINGE, WENN NULL
space:
    lda #0                ;ALLE SEGMENTE AUS = LEERZEICHEN
    jsr digit             ;SPRINGE ZUR UNTERROUTINE ...
    bra nextspace         ;SPRINGE IMMER ...
exit:
    rts                  ;UNTERROUTINE BEENDEN
LEDDIGIT:
    lda _A_lo             ;LADE AKKUMULATOR
digit:
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    jsr shift             ;SPRINGE ZUR UNTERROUTINE ...
    rts                  ;UNTERROUTINE BEENDEN
shift:
    bclr ledshdat,pcdat    ;PORT DATA LOW SCHALTEN
    lsra                  ;SHIFT NACH LINKS
    bcc shftex            ;SPRINGE, WENN CARRY GELOESCHT ...
    bset ledshdat,pcdat    ;PORT DATA HIGH SCHALTEN
shftex:
    bset ledshclk,pcdat    ;PORT CLOCK HIGH SCHALTEN

```

```
        bclr ledshclk,pcdat ;PORT CLOCK LOW SCHALTEN
        rts                ;UNTERROUTINE BEENDEN

load10:
        clr _A              ;LOESCHE AKKUMULATOR
        lda #10             ;LADE TEILUNGSFAKTOR HIER DEZIMAL
        sta _A_lo           ;SPEICHERE AKKUMULATOR
        rts                ;UNTERROUTINE BEENDEN

* ABFRAGE OB DCF SIGNAL EMPFANGEN WIRD *****
* WIRD KEIN SIGNAL EMPFANGEN SO IST MERKER [192] = LOW
* WIRD EIN SIGNAL EMPFANGEN SO IST MERKER [192] AM BLINKEN 1-SEK TAKT
* IST DAS DCF SIGNAL ERKANNT. D.H. DIE INNERE UHR IST DCF SYNCRON,
* DANN IST DER MERKER [192] = HIGH.
DCFSTATUS:
        brset 4,$00,dcf1     ;WENN BIT-5 PORT-A GESETZT SPRINGE NACH DCF1
        bset 7,$B8           ;SETZE CC-MERKER = BIT [192]
        jmp dcfend           ;SPRINGE NACH DCFEND

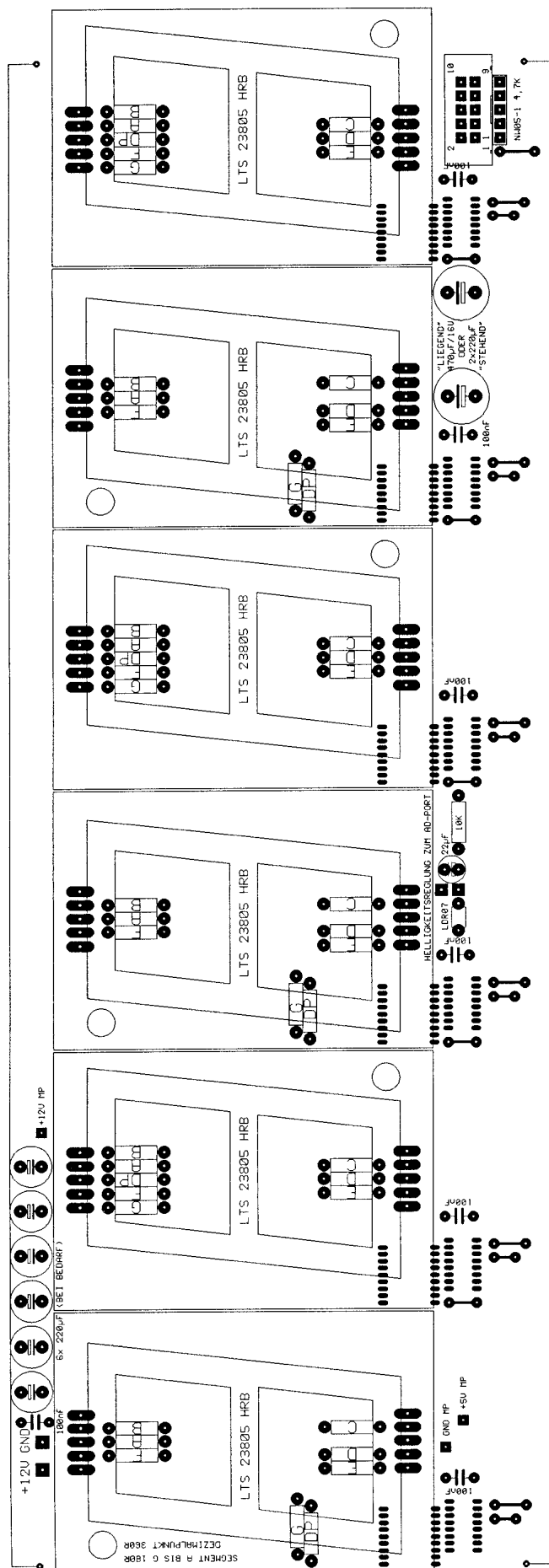
dcf1:
        bclr 7,$B8           ;LOESCHE CC-MERKER = BIT [192]
dcfend:
        rts                ;UNTERROUTINE BEENDEN

* ALLE EINGAENGE VON PORT-A AUF AUSGANG SCHALTEN: *****
* WENN DIE EINGAENGE ALS AUSGANG DEFINIERT SIND, SO SIND DIESE
* NIEDEROHMIG UND LIEGEN AUF GND POTENTIAL
* D.H. SIE SIND WENIGER EMPFINDLICH GEGEN STATISCHE ENTLADUNGEN
DEFINIERT:
        lda $04              ;DDRA HOLEN (SOLLTE %01011111 SEIN)
        ora #%11011110       ;CTS ZUSAETZLICH ALS AUSGANG SCHALTEN
        sta $04              ;AKKU AUF ADRESSE .. SPEICHERN
        lda $01              ;PORT A HOLEN
        and #%01111111        ;CTS AUF 0-VOLT SETZEN (GND)
        sta $01              ;AKKU AUF ADRESSE .. SPEICHERN
        rts                ;UNTERROUTINE BEENDEN

* ZEICHENTABELLE FUER DIE ZEICHEN 0,1,2,3,4 *****
* ..... 5,6,7,8,9
* DIESE TABELLE MUSS IMMER AM ENDE DES ASSEMBLERPROGRAMMS STEHEN!
ledcodes:
        fcb $BF, $89, $DE, $DB, $E9
        fcb $F3, $F7, $99, $FF, $FB
```

6-STELLIGE LED ANZEIGE

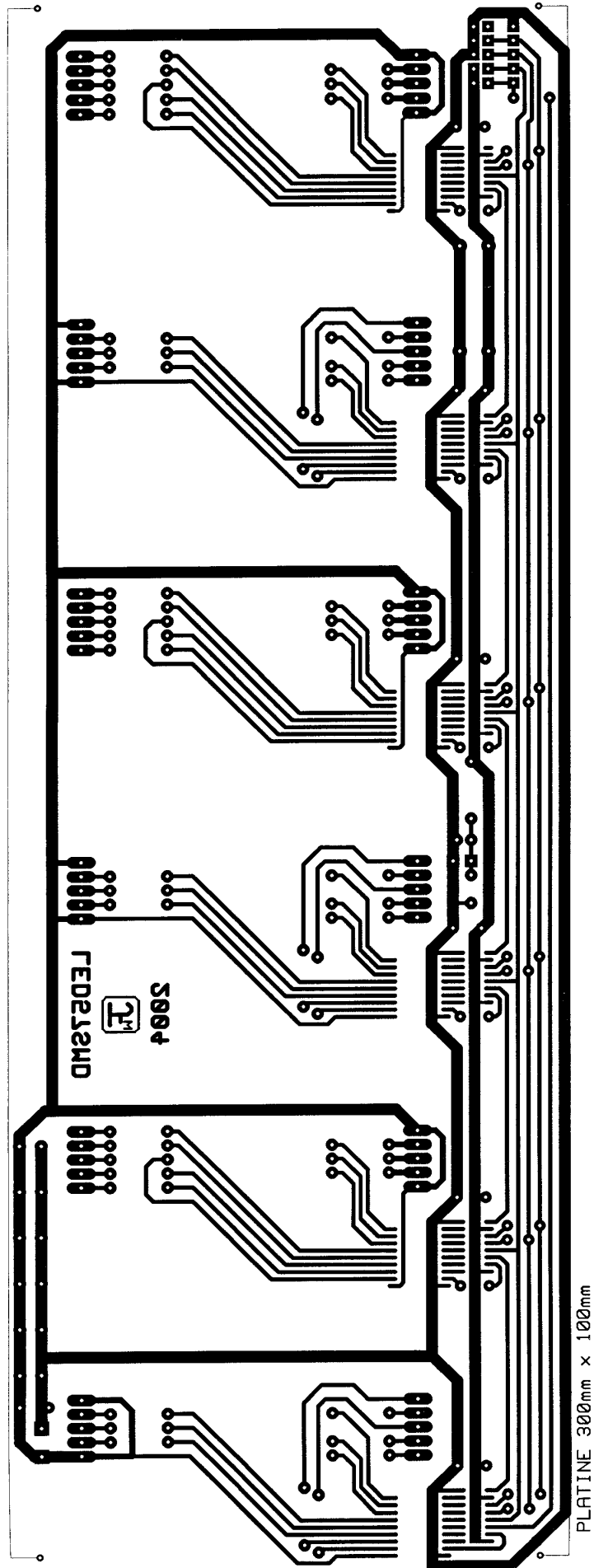
DIE ANZEIGE EIGNET SICH FÜR DIE ANZEIGE VON ZAHLENWERTEN ODER DER UHRZEIT
 DIE ANSTEUERUNG ERFOLGT SERIELL IN EIN 48-BIT SCHIEBEREGISTER

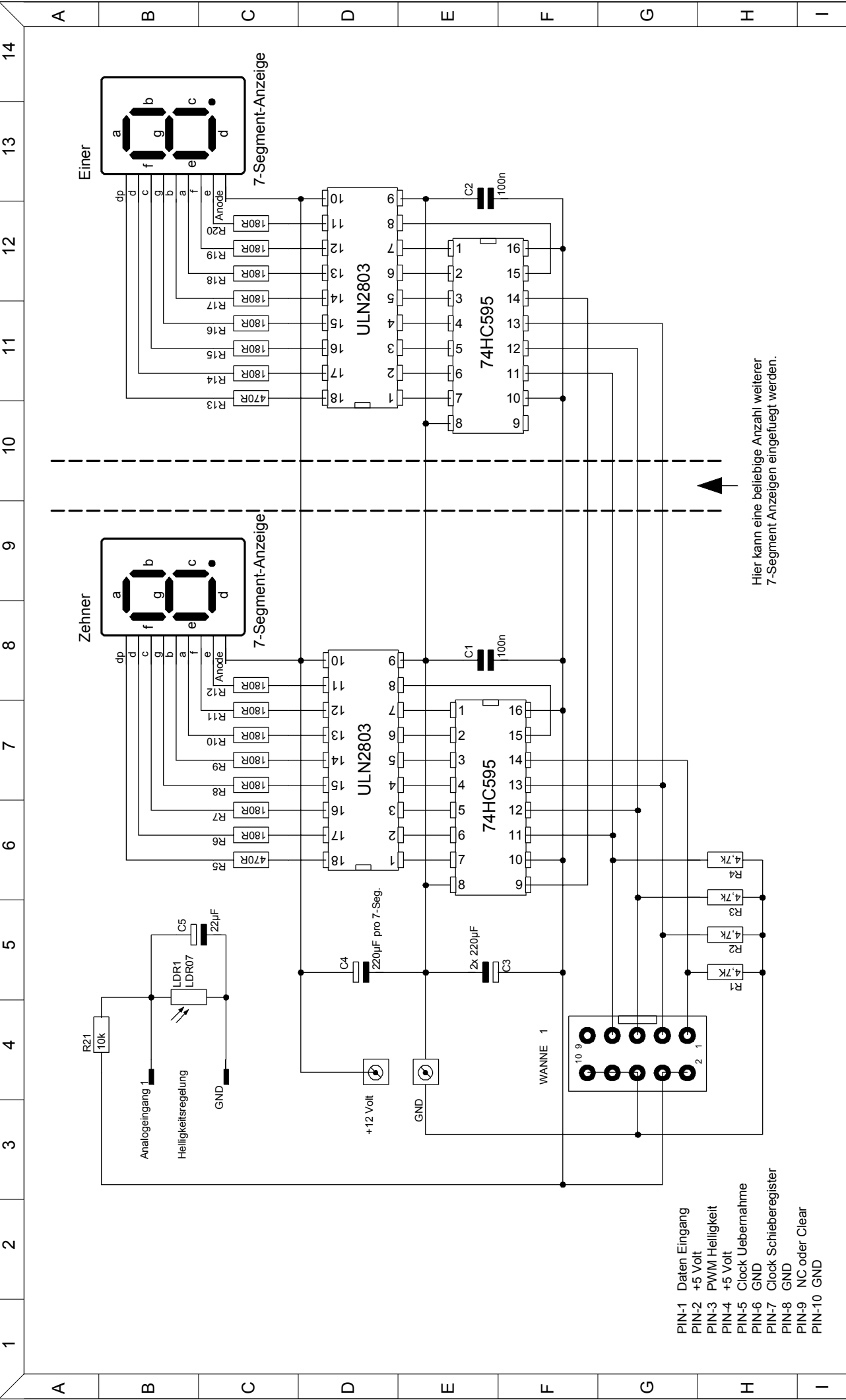


PLATINE 300mm x 100mm

6-STELLIGE LED ANZEIGE

DIE ANZEIGE EIGNET SICH FÜR DIE ANZEIGE VON ZAHLENWERTEN ODER DER UHRZEIT
DIE ANSTEUERUNG ERFOLGT SERIELL IN EIN 48-BIT SCHIEBEREGISTER





Datum		Name		Bezeichnung	Blatt
August 2004		J. Fisch			
gez.:					
gepr.:					
Norm:					
				LED-ANZEIGE fuer C-CONTROL	
				Datenuebertragung Seriell	
				Segmentreihenfolge auf Diskrete IC's abgestimmt	
				von	
				1	